



**UZAKTAN ALGILAMA SİSTEMLERİ İÇİN PERFORMANS BİLİNÇLİ
BÜYÜK VERİ YÖNETİMİ**

Mustafa Kemal PEKTÜRK

**DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

TEMMUZ 2023

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Mustafa Kemal PEKTÜRK

27/07/2023

UZAKTAN ALGILAMA SİSTEMLERİ İÇİN PERFORMANS BİLİNÇLİ BÜYÜK VERİ YÖNETİMİ

(Doktora Tezi)

Mustafa Kemal PEKTÜRK

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Temmuz 2023

ÖZET

Günümüzde yaşanan teknolojik gelişmeler sonucunda birçok organizasyon tarafından günlük olarak üretilen veri miktarı aşırı olarak artmıştır. Yeni teknolojiler sayesinde hem uzamsal hem de spektral çözünürlüklerinin artmasıyla boyutları üstel olarak artan ve büyük veriye dönüşen uzaktan algılayıcı verileri depolama, taşıma ve işleme konusunda büyük zorluklara neden olmaktadır. İlgili büyük verilerin gerçek/yakın gerçek zamanlı analizinin gerektiği zaman kritik uygulamalarda, yetersiz kalan geleneksel yöntemler yerine dağıtık yaklaşımların uygulanması bir zorunluluk haline gelmiştir. Bu tez kapsamında, uzaktan algılama sistemlerinde gerçek/yakın gerçek zamanlı büyük veri analizini mümkün kılabilmek amacıyla ağ topolojisinin avantajlarından yararlanan ve kaynak kullanımını optimize eden, coğrafi dağıtılmış bulut üzerinde çalışacak yeni bir büyük veri yönetimi sistemi önerilmiştir. Önerilen iki fazlı PAA metodu, algılayıcılar ve veri merkezleri arasındaki gecikme, bant genişliği ve sıçrama sayısı metriklerini; diğer yandan veri merkezlerindeki işlemci, depolama, bellek ve IOPS miktarlarını gözeterek en uygun atamayı yapar. Önerilen yaklaşımın başarısını ölçebilmek için bir küme bilgisayarı üzerinde sanal makinelerden oluşan sanal bir ağ kurulmuştur. Bunun yanında önerilen PAA metodunu kıyaslayabilmek adına RA, HA ve HDFS metotları geliştirilmiştir. Deneysel sonuçlara göre PAA metodu, daha hızlı şekilde veri depolamakla birlikte veri merkezlerinde oluşabilecek aşırı kullanımı engelleyerek kaynak kullanımını iyileştirmiştir.

Bilim Kodu : 92414

Anahtar Kelimeler : Büyük veri analitiği, yüksek performanslı hesaplama, gerçek zamanlı analiz, uzaktan algılama, coğrafi dağıtık bulut, Lagrange gevşemesi, optimizasyon, doğrusal programlama

Sayfa Adedi : 124

Danışman : Prof. Dr. Hadi GÖKÇEN

İkinci Danışman : Öğr. Gör. Dr. Muhammet ÜNAL

PERFORMANCE-AWARE BIG DATA MANAGEMENT FOR REMOTE SENSING
SYSTEMS

(Ph. D. Thesis)

Mustafa Kemal PEKTÜRK

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

July 2023

ABSTRACT

As a result of today's technological advancements, many organizations produce an excessive amount of data on a daily basis. Remote sensing data whose spatial and spectral resolutions increase with technology turns into big data causes great difficulties in storing, transferring, and processing. Therefore, when real-time/near real-time analysis of big data is required, it becomes a necessity to apply distributed approaches instead of traditional ones. In this thesis, a novel big data management system is proposed to enable near real-time processing of big data emerging in remote sensing applications on a geographically distributed cloud. In this new system, the advantages of network topology are utilized, and workloads are balanced by taking into account the usages of processor, storage, and memory resources of data centers. Thus, the data to be taken from the sensors and stored in the data center will be sent to the data center with the most appropriate resource and network speed, and this will ensure faster transfer, storage, and processing of the relevant data. The proposed two-phase PAA method makes the most appropriate assignment by considering the metrics of latency, bandwidth, and hops between sensors and data centers; on the other hand, the amount of processor, storage, memory, and IOPS in the data centers. In order to measure the success of the proposed approach, a virtual network consisting of virtual machines was established on a cluster computer. In addition, RA, HA, and HDFS methods have been developed to compare the proposed PAA method. According to the experimental results, the PAA method not only stores data faster but also improves resource usage by preventing excessive use that may occur in data centers.

Science Code : 92414

Key Words : Big data analytics, high-performance computing, real-time analytics, remote sensing, geo-distributed cloud, Lagrangian relaxation, optimization, linear programming

Page Number : 124

Supervisor : Prof. Dr. Hadi GÖKÇEN

Co-Supervisor : Dr. Muhammet ÜNAL

TEŞEKKÜR

Bu uzun yolculukta beni yalnız bırakmayan ve her türlü desteklerini benden esirgemeyen kıymetli danışmanlarım Prof. Dr. Hadi GÖKÇEN ve Öğr. Gör. Dr. Muhammet ÜNAL hocalarıma; akademik bilgi birikimleriyle çalışmalarına katkı sağlayan, değerli tez izleme komitesi üyesi hocalarım Prof. Dr. Hasan Şakir BİLGE ve Doç. Dr. İbrahim Alper DOĞRU'ya sonsuz teşekkür ve saygılarımı sunarım.

Tezimi, doktora çalışmalarım boyunca sabrını ve desteğini benden esirgemeyen eşim Şeyma PEKTÜRK'e, çocuklarım Murat Yunus PEKTÜRK ve Asrın Yusuf PEKTÜRK'e ithaf ediyorum.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ	x
ŞEKİLLERİN LİSTESİ	xi
SİMGELER VE KISALTMALAR	xiv
1. GİRİŞ.....	1
2. LİTERATÜRDEKİ ÇALIŞMALAR	9
2.1. Yerleşik Yaklaşımlar.....	9
2.2. Dağıtık Yaklaşımlar	12
2.2.1. Küme yaklaşımı	14
2.2.2. Bulut yaklaşımı	17
3. UZAKTAN ALGILAMA BÜYÜK VERİSİ VE MİMARİ YAKLAŞIMLAR.....	33
3.1. Uzaktan Algılama Verileri	33
3.2. Uzaktan Algılama Veri İşleme Adımları	35
3.3. Uzaktan Algılama Algoritmalarındaki Veri Erişim Desenleri.....	36
3.4. Gerçek Zamanlı Büyük Veri Mimarisi	38
3.4.1. Dağıtık dosya sistemi.....	40
3.4.2. Dağıtık veri tabanı.....	43
3.4.3. Kullanıcı arayüzü, görselleştirme ve izleme.....	45
3.4.4. Yüksek performanslı hesaplama	46
3.5. Değerlendirme.....	57
3.6. Motivasyon.....	59

4. PERFORMANS BİLİNÇLİ ATAMA TABANLI OPTİMİZASYON MODELİ.....	61
4.1. Coğrafi Dağıtık Bulut.....	62
4.2. Performans Bilinçli Atama Metodu	62
4.2.1. Lagrange gevşemesinin probleme uygulanması	67
4.2.2. Alt eğri yöntemi	69
4.2.3. Uygulanabilir başlangıç çözümü.....	70
4.3. Veri Parçalama	71
5. YAZILIM GERÇEKLEŞTİRME	73
5.1. Sistemin Yapısal Tasarımı	73
5.1.1. Kamera bilgisayar	73
5.1.2. Merkezi bilgisayar	75
5.1.3. Veri merkezi.....	77
5.2. Sistemin Davranışsal Tasarımı.....	78
5.3. Karşılaştırma için Gerçekleştirilen Yöntemler.....	80
5.3.1. Rastgele atama	80
5.3.2. Macar metodu	80
5.3.3. HDFS kullanımı	82
6. DENEYSEL ÇALIŞMALAR	83
6.1. Simülasyon Ağının Kurulması.....	83
6.1.1. Quagga ile ağ konfigürasyonu	86
6.1.2. SNMP ile bilgi yönetimi	88
6.2. Kullanılan Veri Kümesi	89
6.3. Deneysel Sonuçlar.....	92
6.3.1. Kullanılabilir depolama miktarlarının zamana göre değişimi	94
6.3.2. İşlemci kullanımlarının zamana göre değişimi	98

	Sayfa
6.3.3. Bellek kullanımlarının zamana göre değişimi	102
6.3.4. IOPS miktarlarının zamana göre değişimi	105
6.3.5. Veri merkezlerinin kaynak durumlarının birlikte değerlendirilmesi	108
6.4. Bilinen Kısıtlar	111
7. SONUÇ	113
KAYNAKLAR	117



ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Uzaktan algılama alanındaki yerleşik ve dağıtık yaklaşımlar	23
Çizelge 3.1. Uydu veri merkezleri veri hız ve hacimleri	34
Çizelge 4.1. Ağırlık matrisi başarı yüzdesi	67
Çizelge 6.1. Sanal ağda kurulan bilgisayarların özellikleri	84
Çizelge 6.2. Sanal ağ üzerinde gerçekleşen metrikler	85
Çizelge 6.3. AVIRIS veri portalından alınan görüntülerin özellikleri	91
Çizelge 6.4. Kullanılan yöntemler için işlem süreleri	92
Çizelge 6.5. Kullanılan yöntemler için toplam depolanan veri miktarları	94

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. Uydu üzerindeki uzak algılayıcıdan ışıma bilgisinin elde edilerek hedef tanıma işleminin yapılması.....	4
Şekil 1.2. Spektral veri küpü yapısı, tek piksel için spektra ve tek dalga boyundaki görüntü.....	5
Şekil 1.3. Büyük veri karakteristikleri	6
Şekil 3.1. Uzaktan algılama veri işleme adımları	35
Şekil 3.2. Dikdörtgen blok erişim deseni.....	37
Şekil 3.3. Çapraz dosya erişimi deseni	37
Şekil 3.4. Düzensiz erişim deseni	38
Şekil 3.5. Gerçek zamanlı büyük veri mimarisi.....	39
Şekil 3.6. Hadoop dağıtık dosya sistemi.....	41
Şekil 3.7. Çok çekirdekli işlemciler.....	47
Şekil 3.8. Grafik işlemci birimi.....	47
Şekil 3.9. Programlanabilir mantık blokları.....	48
Şekil 3.10. Bilgisayar kümesi	49
Şekil 3.11. Geleneksel yöntem ve sanallaştırma.....	50
Şekil 3.12. Hadoop ekosistemi.....	54
Şekil 4.1. Coğrafi dağıtık bulut hiyerarşisi ve yeryüzü görünümü	63
Şekil 5.1. Bileşen diyagramı	73
Şekil 5.2. Metrik bulucu bileşeni sözde kodu	74
Şekil 5.3. Dinleyici bileşeni sözde kodu.....	75
Şekil 5.4. Gönderici bileşeni sözde kodu.....	75
Şekil 5.5. Merkezi hesaplama bileşeni sözde kodu.....	76
Şekil 5.6. GAMS çağırıcı bileşeni sözde kodu	77
Şekil 5.7. Önerilen model için aktivite diyagramı	79

Şekil	Sayfa
Şekil 5.8. Macar metodu için örnek maliyet matrisi.....	81
Şekil 5.9. Birinci modifiye matrisin elde edilmesi	81
Şekil 5.10. İkinci modifiye matris ve çözümün elde edilmesi.....	82
Şekil 6.1. Önerilen modelin simülasyonu için kurulan heterojen ağ	84
Şekil 6.2. Önerilen modelin simülasyonu için kurulan homojen ağ.....	85
Şekil 6.3. Yönlendirici bilgisayarda Quagga konfigürasyonu	87
Şekil 6.4. Yönlendirici bilgisayarda IP forwarding özelliğinin etkinleştirilmesi	87
Şekil 6.5. Yönlendirici bilgisayarda OSPF konfigürasyonu.....	87
Şekil 6.6. Bilgisayarların ağ konfigürasyonları	88
Şekil 6.7. SNMP komutları ile metriklerin alınması.....	90
Şekil 6.8. Web uygulaması olarak erişilebilen AVIRIS veri portalı.....	91
Şekil 6.9. Heterojen ağda toplam kullanılabilir depolama alanının zamana göre değişimi	93
Şekil 6.10. Homojen ağda toplam kullanılabilir depolama alanının zamana göre değişimi.....	94
Şekil 6.11. RA yöntemi için veri merkezlerindeki kullanılabilir depolama miktarının zamana göre değişimi.....	95
Şekil 6.12. HA yöntemi için veri merkezlerindeki kullanılabilir depolama miktarının zamana göre değişimi.....	96
Şekil 6.13. PAA yöntemi için veri merkezlerindeki kullanılabilir depolama miktarının zamana göre değişimi.....	97
Şekil 6.14. HDFS yöntemi için veri merkezlerindeki kullanılabilir depolama miktarının zamana göre değişimi.....	98
Şekil 6.15. RA yöntemi için veri merkezlerindeki işlemci kullanımları	99
Şekil 6.16. HA yöntemi için veri merkezlerindeki işlemci kullanımları	100
Şekil 6.17. PAA yöntemi için veri merkezlerindeki işlemci kullanımları	101
Şekil 6.18. HDFS yöntemi için veri merkezlerindeki işlemci kullanımları.....	101
Şekil 6.19. RA yöntemi için veri merkezlerindeki bellek kullanımları	102
Şekil 6.20. HA yöntemi için veri merkezlerindeki bellek kullanımları.....	103

Şekil	Sayfa
Şekil 6.21. PAA yöntemi için veri merkezlerindeki bellek kullanımları.....	104
Şekil 6.22. HDFS yöntemi için veri merkezlerindeki bellek kullanımları	104
Şekil 6.23. RA yöntemi için veri merkezlerindeki IOPS miktarları	105
Şekil 6.24. HA yöntemi için veri merkezlerindeki IOPS miktarları.....	106
Şekil 6.25. PAA yöntemi için veri merkezlerindeki IOPS miktarları.....	107
Şekil 6.26. HDFS yöntemi için veri merkezlerindeki IOPS miktarları	107
Şekil 6.27. RA için veri merkezlerindeki kaynak durumlarının toplu gösterimi.....	109
Şekil 6.28. HA için veri merkezlerindeki kaynak durumlarının toplu gösterimi.....	110
Şekil 6.29. PAA için veri merkezlerindeki kaynak durumlarının toplu gösterimi	110
Şekil 6.30. HDFS için veri merkezlerindeki kaynak durumlarının toplu gösterimi	111

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

Açıklamalar

GB

Gigabyte

PB

Petabyte

TB

Terabyte

min

Minute

ms

Millisecond

dk

Dakika

Kısaltmalar

Açıklamalar

API

Application Programming Interface

BGP

Border Gateway Protocol

BT

Bilişim Teknolojisi

COTS

Commercial Off-The-Shelf

CPU

Central Process Unit

DAS

Direct Attached Storage

DPGrid

Digital Photogrammetric Grid System

DSP

Digital Signal Processors

EDW

Enterprise Data Warehouse

FFT

Fast Fourier Transform

FPGA

Field Programmable Gate Array

GAMS

General Algebraic Modeling System

GFS

Google File System

GPFS

General Parallel File System

GPU

Graphics Processing Unit

HDD

Hard Disk Drive

HDFS

Hadoop Distributed File System

Kısaltmalar**Açıklamalar**

HPC	High Performance Computing
I/O	Input/Output
IOPS	Input/Output Operations Per Second
IoT	Internet of Things
MIPS	Million Instructions per Second
MPI	Message Passing Interface
MPP	Massively Parallel Processing
NAS	Network Attached Storage
NDPP	N-Dimensional Parallel Partitioning Module
NDVI	Normalized Difference Vegetation Index
NUMA	Non-Uniform Memory Access
OGC	Open Geospatial Consortium
OpenMP	Open Multi-Processing
PCI	Peripheral Component Interconnect
PCM	Phase Change Memory
PLC	Programmable Logic Controller
RDMA	Remote Direct Memory Access
RIP	Routing Information Protocol
RS	Remote Sensing
S3	Amazon Simple Storage
SAN	Storage Area Network
SSD	Solid State Drive
SVM	Support Vector Machine
WCPS	Web Coverage Processing Service
WCS	Web Coverage Service
WFS	Web Feature Service
WPS	Web Processing Service

1. GİRİŞ

Son teknolojik gelişmelerle birlikte ortaya çıkan veri hacmindeki inanılmaz artış, geleneksel yaklaşımları kullanan analiz süreçlerini çeşitli alanlardaki birçok uygulama için daha zor hale getirmiştir. Özellikle uzaktan algılayıcıların yeni teknolojiler sayesinde hem uzamsal hem de spektral çözünürlüklerinin artmasıyla boyutları üstel olarak artan ve büyük veriye dönüşen uzaktan algılayıcı verileri depolama, taşıma ve işleme konusunda büyük zorluklara neden olmaktadır (Toma, Panica, Zaharie ve Petcu, 2012; Ma ve diğerleri, 2015). İlgili büyük verilerin gerçek/yakın gerçek zamanlı analizinin gerektiği zaman kritik uygulamalarda yetersiz kalmakta olan geleneksel yöntemler yerine paralel yaklaşımların uygulanması bir zorunluluk haline gelmiştir.

Hem fazla sayıda piksel içeren hem de çok miktarda spektral banttı (multispektral/hiperspektral) oluştı büyük görüntü dosyaları, okunma ve bellekte tutulma konusunda büyük güçlüklerı neden olmaktadır. Uydu ve uzaktan algılama verilerinden bilgi çıkartımı yapabilmek için zaman ihtiyacı olan teknik ve işlemlerin kullanılması, veri işlemlerinin iyileştirilmesini ve hızlı cevaplama süresini önemli ölçüde etkilemektedir. Bu özellikleriyle uzaktan algılama uygulamaları, hem veri-yoğun (data-intensive) (Ma, Wang, Liu ve Ranjan, 2015) hem de hesaplama-yoğun (compute-intensive) (Aji ve diğerleri, 2013) olan uygulamalardır.

Büyük veri analizcileri, en önemli teknik güçlüklerin çok büyük veri yapılarının bellekte tutulmaya çalışılmasında ve makul sürede sonuca ulaşılmasında yaşandığını belirtmektedirler (Zhong, Fang ve Zhao, 2013). Birçok algoritmanın hesaplama karmaşıklığı, örnek sayısına ve örneğin boyutuna süper lineer bir şekilde bağılıdır. Bu yüzden bu iki değişkende artış yaşanırken sadece kodun optimizasyonu daha iyi performans için yeterli olmayacaktır.

Büyük veri analizi söz konusu olduğunda bilgisayar mimarisinde bulunan önde gelen güçlükler CPU'nun yoğun ve I/O işlemlerinin yavaş olmasıdır. Moore Yasasına göre her 18 ayda CPU ve sürücü performansı iki katına çıkmaktadır. Diğer yandan I/O arayüzlerindeki eğilimler incelendiğinde ağ hızına yakın iyileşme olsa da hala geride kalınmaktadır (Oliveira, Furlinger ve Kranzlmüller, 2012). I/O arayüzleri, işlemci veri yoluyla aynı hızda

alışmasına rağmen evresel donanım kartlarının düşük hızda alışmasından dolayı geride kalmaktadır. I/O performansının artırılması iin PCI kartları kullanılmaya başlanmıştır. I/O performansı yıllık olarak yaklaşık 33% artmaktadır. Geleneksel yaklaşımlardaki bu iyileştirmelere karşın toplanan bilgilerin üstel olarak artması fakat bilgi işleme tekniklerinin göreceli olarak yavaş kalması özellikle gerçek zamanlı analizlerin yapılmasını güçleştirmektedir (Chen ve Zhang, 2014). Bu nedenle analiz işlemlerinin performansını belirleyen en önemli etmen, donanım kaynaklarının yapısı gereği limitli olmasıdır (Fernández ve diğerleri, 2015). Bundan dolayı bilgi ıkarımının karmaşık ve zaman alıcı olduğu büyük hacimli veriler üzerinde analiz işlemlerinin gerçekleştirilebilmesi iin multi-core programlama, GPU programlama, FPGA, küme bilgisayar, bulut bilişim gibi paralel işleme yeteneğine sahip modern teknolojilere ihtiyaç duyulmaktadır (Fernández ve diğerleri, 2015; Lin ve diğerleri, 2013; Krämer ve Senner, 2015).

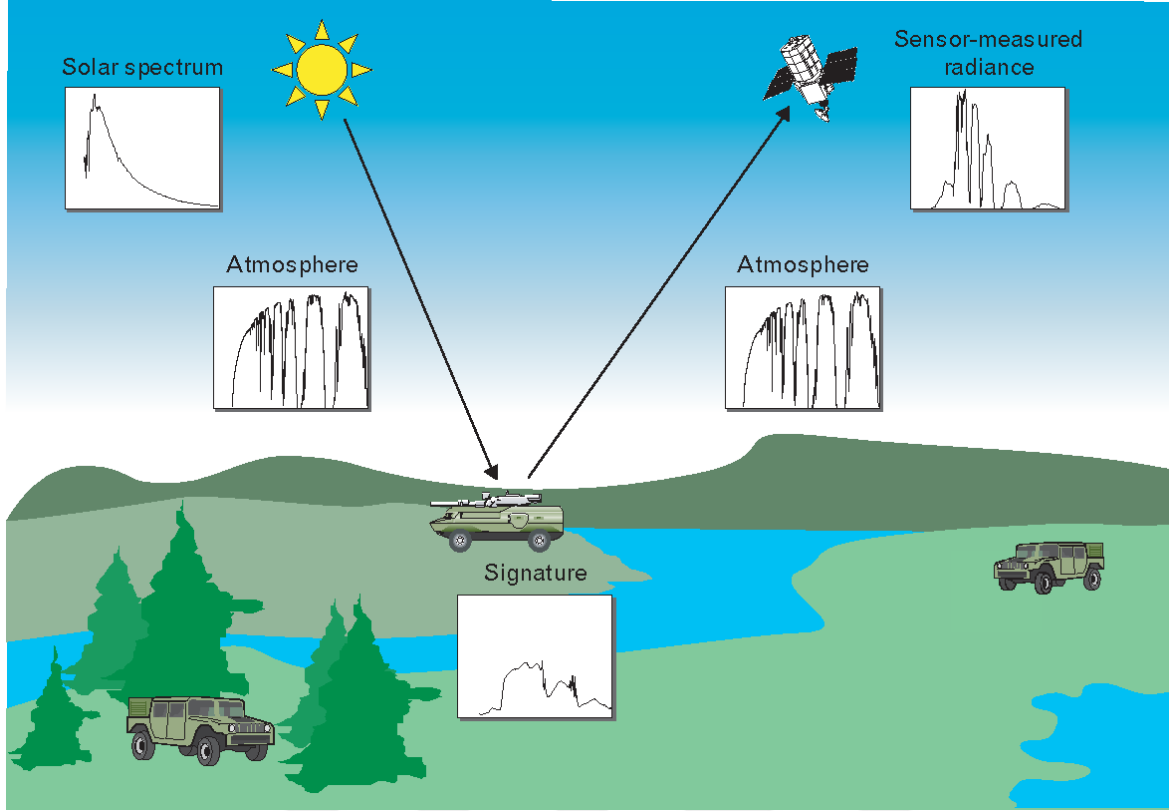
Büyük veri doğal olarak bir bilgisayarın belleğine yerleştirilmeye uygun değildir ve dağıtık olarak oklu bilgisayar sistemleri üzerinde dağıtık yaklaşımlarla tutulması bir zorunluluk haline gelmiştir (Plaza ve diğerleri, 2006). Bu noktada veri yönetimi ve analiz işlemleri iin oklu bilgisayarlardan oluşan bir altyapı sunan bulut kavramının kullanılması, yüksek ölçeklenebilirlik ve kullanılabilirlik, hataya toleranslılık, performanslılık anlamında büyük kolaylıklar sağlamaktadır. Özellikle yeni nesil uzaktan algılayıcıların yakın gerçek zamanlı üretebildiği verilerden bilgi ıkarımı yapılması gereken kritik uygulamalar düşünüldüğünde bulut bilişim teknolojilerinin yüksek performanslı hesaplama amacıyla kullanılması büyük önem kazanmıştır. Ayrıca bulut bilişim altyapısı, analiz işlemlerinin performanslı yapılmasının yanında büyük verinin depolanması iin de verimli bir platform oluşturma yeteneğine sahiptir. Böylece bu teknolojinin kullanılmasıyla birlikte küme sistemleri gibi bakımı pahalı hesaplama donanımları, ayrılmış alan ve yazılım ihtiyaçları ortadan kaldırılabilmektedir (Qu ve diğerleri, 2012).

Bu alışmada incelenen araştırmalara göre, mevcut tekniklerin ve sistemlerin uzaktan algılamada gerçek/yakın gerçek zamanlı büyük veri analizini ele alırken mevcut sorunları kapsayan bir özüm bulamamış oldukları görülmüştür. Hadoop ve benzeri teknolojiler, büyük jeo-uzamsal verileri işlemek iin jeoloji toplumlarında giderek daha fazla ilgi çekmeye başlamıştır. Büyük jeo-uzamsal verilerin işlenmesi iin bu teknolojilerin benimsenmesi amacıyla birçok araştırma yapılmıştır, ancak dinamik jeo-işleme iş yükünü verimli bir şekilde işlemek iin bilgisayar kaynaklarının optimize edilmesi hususunda ok

az çalışma bulunmaktadır. Bu noktada tez kapsamında uzaktan algılama büyük verilerini uygun bir bulut mevkisinde depolamak ve bunları coğrafi dağıtık bir bulut üzerinde kaynak tahsisini optimize ederek işlemeye olanak sağlamak için yenilikçi bir yaklaşım önerilmiştir. Önerilen model, iki aşamalı yeni bir optimizasyon problemi çözme tekniği kullanır ve gerektiği durumlarda çözümsüzlüğü engellemek için Lagrange gevşemesi (Lagrange relaxation) metodunu kullanarak yaklaşık bir çözüm sunar. Bunu yaparken sensörler ile veri merkezleri arasındaki veri aktarım süresini ve hata oranını azaltmak için ağ metriği olarak bant genişliği, gecikme ve sıçrama sayısını kullanır. Diğer yandan büyük veri analizinin verimli bir şekilde yapılabilmesi için veri merkezlerindeki işlemci, depolama, bellek kaynakları ve iş yükünü hesaba katarak kaynak paylaşımını optimize eder. Bu noktada çalışmamızın literatürdeki çalışmalardan farkları veri aktarım süresini azaltmak, depolama alanı en uygun veri merkezini seçmek ve veri merkezlerinin iş yükünü dikkate almaktır. Öte yandan, mümkün olduğunca RS verilerini veri merkezinde bir bütün olarak tutarak, sıklıkla bitişik piksellere ihtiyaç duyan RS uygulamaları için analiz süreçlerinde veri okuma zamanını azaltması modelin bir başka önemli avantajıdır. Önerilen modelin ilk aşamasında veriler bir bütün olarak veri merkezlerine atanırken, kaynak eksikliği olursa ya da kısıtlar karşılanamazsa ikinci aşama başlatılır. Modelin ikinci aşamasında ise karşılanamayan kısıtlar, Lagrange gevşemesi ile gevşetilir ve ilk aşamada atanamayan verilerin atanması sağlanır. İkinci aşama, kısıtları karşılamak için verilerin birden fazla veri merkezine bölünüp bölünmeyeceğini de kararlaştırır.

Uzaktan algılama

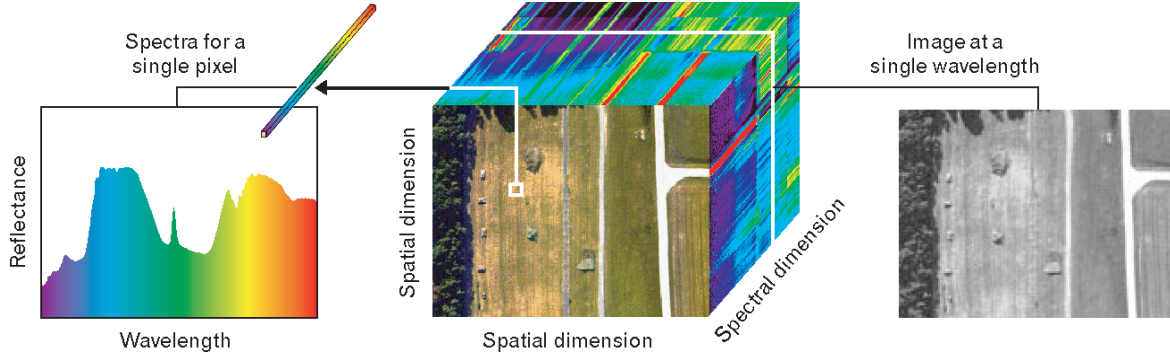
Uzaktan algılama (Remote Sensing - RS), genellikle bir mesafeden bir nesnenin veya yüzeyin özelliklerini ölçme teknolojisi olarak tanımlanır. Yerinde gözlemin aksine, fiziksel bir temas olmaksızın bir nesne veya olgu hakkında bilgi elde etme tekniğidir. Bu genellikle algılayıcıların veya uzak algılayıcıların kullanılarak nesnelerin elektromanyetik ışıma (radiance) ve yansıma (reflectance) karakterlerinin tanınmasını ifade eder (Şekil 1.1).



Şekil 1.1. Uydu üzerindeki uzak algılayıcıdan ışıma bilgisinin elde edilerek hedef tanıma işleminin yapılması (Manolakis, Marden ve Shaw, 2003)

Uzaktan algılama verisi

Son teknolojik gelişmelerle uzak algılayıcılar, yeryüzü üzerinde belirli bir alana ait yüzlerce ve hatta binlerce farklı dalga boyu için ölçüm yapabilir hale gelmiştir. Bu tarz algılayıcılardan sağlanan görüntüler veri küpü olarak ifade edilmekle birlikte veri boyutu oldukça geniştir (Şekil 1.2). NASA'nın AVIRIS veri tabanında bulunan görüntü verileri incelendiğinde 20-30 GB büyüklüğüne kadar 224 banttan oluşan görüntü verileri bulunmaktadır.



Şekil 1.2. Spektral veri küpü yapısı, tek piksel için spektra (solda) ve tek dalga boyundaki görüntü (Manolakis, Marden ve Shaw, 2003)

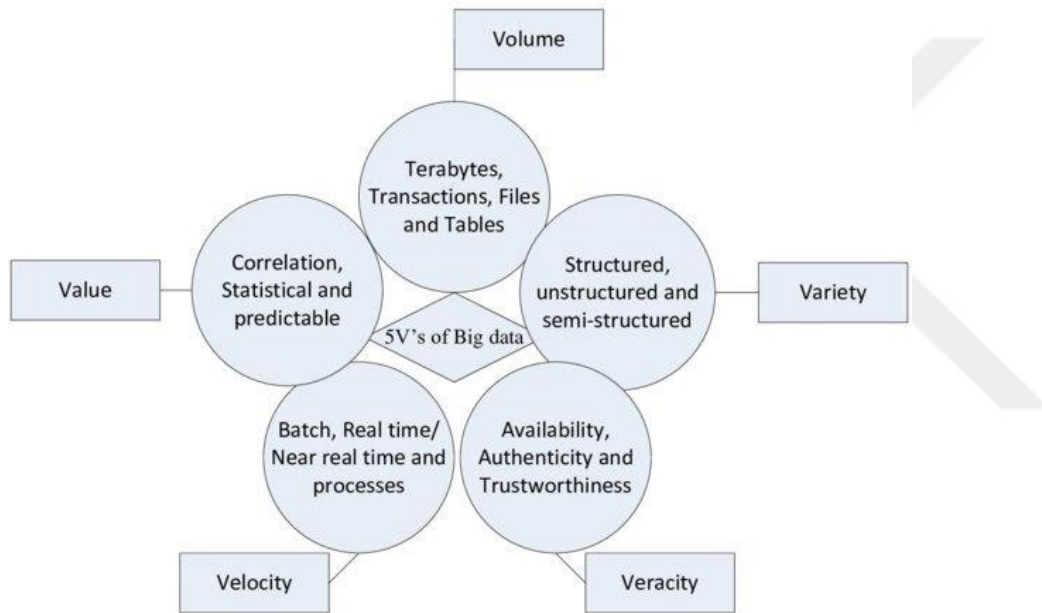
Büyük veri

Uzaktan algılama teknolojilerinin hızla gelişmesi neticesinde aşırı büyüyen uzaktan algılama verilerinin elde edilmesi büyük veri alanında incelenmeye başlanmıştır. Büyük veri, ağ bağlantılı, sayısallaştırılmış, sensör yüklü, bilgi odaklı dünyadaki büyük miktardaki veriyi tanımlamak için kullanılan bir terimdir. Verilerin büyümesi, veri analitiğindeki bilimsel ve teknolojik gelişmeleri geride bırakmaktadır. Bu yüzden verilerin hacmini, hızını ve çeşitliliğini ele almak için yeni ölçeklenebilir mimarilerin kullanılması kaçınılmaz hale gelmiştir. Uzaktan algılama verileri aşağıda belirtilen büyük veri karakteristiklerini göstermektedir (Şekil 1.3):

- **Hacim (volume):** Uygulamaların yönettiği ve analiz ettiği büyük verilerin boyutu ve miktarlarını ifade eder. İstatistiklere göre, Dünya Gözlem Sistemi Veri ve Bilgi Sistemi (EOSDIS) üzerinde arşivlenen toplam veri miktarı 2017 yılı civarında 23,8 petabyte (PB) olmakla birlikte arşiv büyüme hızı günlük ortalama 15,3 TB/gün büyüme hızına ulaşmıştır (Wang, Yan ve Ma, 2019).
- **Hız (velocity):** Bu terim, verilerin oluşturulduğu hızı ifade eder. Buradaki hız, aynı zamanda verinin ne kadar hızlı işleneceği ile de ilgilidir. Günümüzde oluşturulan uydu takımları sayesinde uydu tekrar ziyaret süresi günlerden dakika seviyelerine kadar düşürülmüştür. Bu sayede veri üretme hızı üssel olarak artmıştır.
- **Çeşitlilik (variety):** En önemli Büyük Veri özelliklerinden bir diğeri de çeşitliliğidir. Farklı veri kaynaklarına ait verilerin doğası gereği farklı yapılarda olmasını ifade etmektedir. Verilerin çeşitliliği, depolanması ve analizi için çok önemli bir husustur. RS verileri çevresel izleme, yer örtüsü, atmosferik, hidroloji ve oşinografi içerikli bir çok yer biliminde kullanılmaktadır. Bu kadar farklı ve geniş aralıktaki uygulama alanlarından

dolayı RS verilerinin çeşitliliği çok fazla artmıştır. Bilindiği kadarıyla NASA arşivlerinde yaklaşık 7000 farklı tipte RS veri kümeleri bulunmaktadır (Ma ve diğerleri, 2015).

- Değer (value): Elde edilen verilerin işlenerek değerli bir bilgiye dönüştürülmesini ifade etmektedir. RS verileri tarım, askeri ve çevresel izleme gibi birçok alanda kıymetlendirme ve hedef tanıma amacıyla kullanılmaktadır.
- Doğruluk (veracity): Verilerin güvenilirlik derecesini tanımlar. Karşılaştığınız verilerin çoğu yapılandırılmamış olduğundan, gereksiz bilgileri filtrelemek ve gerisini işlemek için kullanmak önemlidir.



Şekil 1.3. Büyük veri karakteristikleri (Nda ve Tasmin, 2019)

Bulut bilişim

Bulut ortamı bir çok kullanıcının bulunduğu, Bilişim Teknolojisi (BT) kaynaklarının geniş bir ağ üzerinden isteğe bağlı kullanılabilirdiği, bu kaynakların gerektiğinde kullanımının kısıtlanabilirdiği, çok hızlı ve genellikle müdahale olmadan kendini otomatize etmiş ve tercihen sanallaştırılmış bir ortamdır.

Coğrafi dağıtık bulut

Coğrafi dağıtık bulut, verileri ve uygulamaları birbirine bağlamak için farklı coğrafi bölgelere dağıtılmış birden fazla bulut sitesi içeren bulut bilişim teknolojilerinin uygulanmasıdır.

Dağıtık dosya sistemi

Dağıtık dosya sistemi, farklı merkezlerdeki veri düğümlerinin heterojenliğinin soyutlanmasını sağlamak üzere kurulmuştur. Böylece dağıtık dosya sistemi, uygulamaların farklı düğümlerde bireysel olarak farklı işletim sistemleri ve farklı dosya sistemleri kullanan heterojen düğümler üzerindeki verilere erişebilmesi için ortak arayüz sağlar.

Dağıtık veri tabanı

Dağıtılmış bir veri tabanı, temelde tek bir sistemle sınırlı olmayan, farklı sitelere, yani birden fazla bilgisayara veya bir bilgisayar ağı üzerinden yayılan bir veri tabanıdır. Fiziksel bileşenleri paylaşmayan çeşitli sitelerde dağıtılmış bir veri tabanı sistemi bulunur. Bu, belirli bir veri tabanına küresel olarak çeşitli kullanıcılar tarafından erişilmesi gerektiğinde gerekli olabilir. Kullanıcılar için tek bir veri tabanı gibi görünecek şekilde yönetilmesi gerekir.

Tez organizasyonu

Tezin organizasyonu şu şekildedir: 2. bölümde RS alanında ve büyük veri konusunda kullanılmakta olan yerleşik ve dağıtık yaklaşımlar incelenerek bulunan açık noktalar irdelenmektedir. 3. bölümde RS büyük verisi ve bu veriyi yönetebilmek için sunulan mimari yapılar incelenmektedir. 4. bölümde veriyi yönetmek için önerilen iki aşamalı optimizasyon modeli detaylı bir şekilde açıklanmıştır. 5. bölümde önerilen sistemin tasarımı ve gerçekleştirmenin nasıl yapıldığı anlatılmaktadır. 6. bölümde yapılan deneysel çalışmalar ile elde edilen sonuçlar verilmekte ve değerlendirmeler yapılmaktadır. 7. bölümde sonuç açıklanırken gelecekte yapılması planlanan çalışmalar verilmektedir.



2. LİTERATÜRDEKİ ÇALIŞMALAR

Bir önceki bölümde de bahsedildiği üzere uzaktan algılayıcı teknolojilerinde yaşanan son gelişmelerle birlikte uzaktan algılayıcı verilerinin boyutlarında inanılmaz büyük bir artış yaşanmıştır (Oliveira, Furlinger ve Kranzlmüller, 2012). Tek bir uydu veri merkezinde toplanan RS veri miktarı dramatik olarak artmış ve sadece bir uydunun ürettiği günlük veri miktarı birkaç terabaytlık değerlere ulaşmıştır (Chen ve Zhang, 2014). Bunun nedeni, algılayıcıların yeni teknolojiler sayesinde yüksek çözünürlüğe ve fazla miktarda banda sahip olmasıdır. Böylelikle yüksek boyutlara ulaşan RS verileri, “Büyük Veri” olarak tanımlanmaya başlamıştır. Bu kadar fazla miktara ulaşan verilerin depolanması, yönetilmesi, işlenmesi ve taşınması konusunda çözülmesi gereken önemli zorluklar ortaya çıkmaktadır. Özellikle gerçek zamanlı/yakın gerçek zamanlı uygulamalarda geleneksel yöntemler, yenilikçi paralel işleme yaklaşımlarıyla değiştirilerek analiz işlemlerinin makul sürelerde yapılması bir gereksinim haline gelmiştir.

2.1. Yerleşik Yaklaşımlar

Yerleşik (onboard) uzaktan algılama veri işleme senaryolarında, özellikle hava araçlarının kullanıldığı sistemlerde düşük ağırlıklı ve düşük güçle çalışabilen bileşenlere ihtiyaç olabilmektedir. Bu noktada ilk akla gelen çözümlerden biri olan FPGA yaklaşımı faydalı yükü azaltmasının yanında düşük güçle çalışabilen donanım tabanlı bir mimari sunmaktadır. Plaza ve diğerleri (2006) tarafından yapılan çalışmada yerleşik hiperspektral veri sıkıştırma işlemi için kıymetlendirme tabanlı bir algoritma Xilinx Virtex-II isimli FPGA üzerinde paralel olarak geliştirilmiştir. Shan, X. S. Wang ve Z. S. Wang (2010) ise bulut tespiti için kullanılan, karmaşık mimarisi nedeniyle büyük bir bellek ve mantık birimi ihtiyacına sahip olan spektrum ve doku analizi kombinasyonu yaklaşımını XC4VFX140 isimli FPGA üzerinde gerçekleştirmiştir. Bu yaklaşım iki farklı teknikten meydana gelir: spektrum eşiği kıyaslama ve doku analizi. Spektrum eşiği kıyaslama donanım üzerinde gerçek zamanlı gerçekleştirilmek için basitken; fraktal boyut ve açı ikinci moment tekniklerinden oluşan doku analizi hem veri yoğun hem de hesaplama yoğun olduğundan donanım üzerinde gerçekleştirilmesi zordur. Shan ve diğerleri (2010), spektrum ve doku analizi kombinasyonu yaklaşımıyla gerçek zamanlı otomatik bulut tespit sistemi için iyileştirilmiş bir mimari önermiştir. Fraktal boyut için yaygın olarak kullanılan kutu sayma boyutu metodu

kullanılırken 4 aşamalı olarak minimum ve maksimum gri değerler bulunmaktadır. 64*64 piksellik bloklar alınarak 2 aşamada sağa doğru sütunlardaki 4 pikselin minimum ve maksimum değerleri hesaplanırken diğer 2 aşamada aynı hesap aşağı doğru satırlar kullanılarak yapılmaktadır. Her fazda daha az bellek kullanımı elde edilirken aşamalar zamana bağlı olarak kesiştiğinden hesaplama süresi de azalmaktadır. Açık ikinci moment ise piksel çiftleri için tüm matris elemanları güncellendiği hem fazla bellek hem de donanımda geliştirilmesi zor matris işlemleri gerektirir. Açık ikinci momentte kullanılan gri seviyesi eşdizimlilik matrisinin diyagonale göre simetrik olduğunun keşfedilmesiyle matris işlemleri ve gerekli bellek miktarı %50 azaltılmıştır. Alınan sonuçlara göre iyileştirilmiş spektrum ve doku analizi kombinasyonu gerçekleştirmesi klasik yaklaşıma göre 3 kat hızlanma sağlamakla birlikte donanım kaynağı kullanımını %75 azaltmıştır.

Hafif entegre bileşen olmasına rağmen tasarım ve programlama açısından önemli bir efor gerektiren FPGA'ye alternatif olarak günümüzde birçok bilgisayarda mevcut bulunan Multi-core işlemciler ve GPU kartları düşük maliyetleri ve yüksek hesaplama potansiyelinden ötürü tercih edilmektedir. Bernabé ve diğerleri (2013) tarafından yapılan çalışmada bu iki alternatifi kullanarak tam hiperspektral ayrıştırma (unmixing) zincirinin gerçekleştirilmesi sağlanmakla birlikte bu algoritma zinciri daha önceki çalışmada GPU için tartışılırken ilk kez çok çekirdekli işlemci üzerinde geliştirilerek tartışılmıştır. İki farklı görüntü üzerinde yapılan deneylerde iki farklı GPU modeli için ortalama 13 ve 27 kat hızlanmayla gerçek zamanlı işleme sağlanırken iki farklı çok çekirdekli işlemci üzerindeki çalışmalarda ortalama 3,5 ve 4 kat hızlanma sağlanmasına karşın sadece birisinde gerçek zamanlı işleme mümkün olmuştur. Qu ve arkadaşları ise spektral açı eşleştiricisi algoritmasının GPU ve dağıtık küme üzerinde gerçekleştirilmesini sağlayarak verinin büyük miktarda olması durumunda dağıtık hesaplamanın ilk aday olacağı, algoritmanın yapısından dolayı hesaplama miktarının artmasına bağlı olarak GPU ile hızlandırılmasının daha iyi bir performans sunacağı kanısına varmıştır (Qu, Zhang, Y. Chen, H. Chen ve Lin, 2012; Qu, Zhang, Lin ve Chen, 2013). Ayrıca GPU üzerinde geliştirmelerde CUDA kullanımının MATLAB'ta GPU kullanımına göre daha iyi olduğu ve bundan dolayı GPU kullanımında CUDA'nın tercih edilmesinin gerekliliği belirtilmiştir. Setoain, Prieto, Tenllado ve Tirado (2008) tarafından yapılan bir çalışmada ise biçimsel son üye çıkarımı algoritması GPU tabanlı gerçekleştirilmiş ve yerleşik işleme senaryoları için pahalı ve zor olan küme yaklaşımına karşın uygun maliyetli bir alternatif olan GPU yaklaşımı önerilmiştir. Bernabe, López, Plaza ve Sarmiento (2012) tarafından yapılan başka bir çalışmada ise otomatik hedef

bulma ve sınıflandırma algoritması GPU mimarisi üzerinde gerçek zamanlı olarak ilk kez gerçekleştirilmekle birlikte düşük bellek transfer zamanı için paylaşılan bellekteki ve GPU lokalindeki veri düzeni dikkate alınmıştır. Torti ve diğerleri (2014), boyut indirgeme açısından önemli bir algoritma olan hiperspektral altuzay tanıma algoritmasının CPU, GPU ve DSP üzerinde gerçekleştirilmesini sağlamışlardır. Elde edilen deneysel sonuçlara göre veri alma oranına göre gerçek zamanlı kısıtlamaları sağlayan GPU ve CPU gerçekleştirmelerinin yanında DSP yaklaşımının “double precision floating point” ve matematiksel kütüphane desteği olmamasına rağmen düşük güç ve yüksek performans arasında iyi bir ödünleşme sunduğu belirtilmiştir. X. Wu ve diğerleri (2013), son üye çıkarımı için yaygın olarak kullanılan piksel saflığı indeksi algoritmasını GPU ile paralelleştirerek ENVI yazılımında hazır bulunan bu algoritmanın mevcut yazılıma göre 769 kat daha hızlı çalıştırılmasını sağlamışlardır. Z. Wu ve diğerleri (2015) tarafından yapılan çalışmada aralıklı temsili sınıflandırıcıların hiperspektral görüntü analizi amacıyla GPU üzerinde gerçekleştirilmesi sağlanarak 130 kat hızlı sonuç üretilmesi sağlanmıştır. Chang ve diğerleri (2014), hiperspektral görüntü sınıflandırma için en yakın özellik uzayı algoritmasının yenilikçi bir türevini GPU mimarisi kullanarak yer örtüsü sınıflandırmada gerçekleştirmiş ve önemli ölçüde bir hızlanma sağlamışlardır. Liu ve diğerleri (2011) yaptıkları çalışmada büyük uzaktan algılama verilerinin işlenmesi için GPU ve POSIX thread teknolojilerini temel alan dual-paralel bir işleme mekanizması önermişlerdir. Bu yaklaşımda POSIX thread’leri iki gruba ayrılarak organize edilmiştir; birinci grup görüntü verisi ve sistem belleği arasındaki I/O görevi için ikinci grup ise tampon bellekten GPU belleğine veriyi taşıyıp işleme için sorumludur. Böylece GPU ile verilerin işlenmesi %30 daha performanslı yapılabilmektedir.

Bahsi geçen çalışmalardaki Multi-core, GPU ve FPGA donanımları üzerinde programlama yaklaşımında işleme gücü dikey ölçeklendirilebilir, yani bilgisayar üzerinde daha çok işlemci, daha gelişmiş grafik kartı ve daha fazla bellek daha iyi performans anlamına gelmektedir. İlgili teknolojilerin kullanıldığı çalışmalar büyümekte olan veri için bir çözümden ziyade kullanılan algoritmaların bu donanımlar üzerinde gerçekleştirilmesinden ibarettir. Bu konuda yapılan çalışmalar sadece analizin hızlandırılmasını sağlamakla birlikte günlük terabyte’lara varan büyük verinin yönetimi için çaresiz kalmaktadır.

2.2. Dağıtık Yaklaşımlar

RS verilerinin tek bir bilgisayarda ele alınamayacağı ve ölçeklenebilir bir sistemin gerekliliğinin ortaya çıkmasıyla verinin dağıtık olarak çoklu bilgisayar kümesi üzerinde tutulmasını sağlayan dağıtık veri tabanı mimarileri ve dağıtık veri tabanı yönetimi sistemleri çözüm olarak sunulmuştur. Dağıtık veri tabanı, bir bilgisayar ağı ile bağlı çoklu bilgisayar kümesinde her bir düğümde/sunucuda bulunan birbirinden ayrı veri tabanlarından meydana gelir. Dağıtık veri tabanı yönetimi sistemi ise bahsi geçen sunucular üzerinde verilerin dağıtımını ve yönetilmesini sağlar. Karantzas, Bliziotis ve Karmas (2015), çevrim içi olarak büyük uzaktan algılama verilerinin alınarak dağıtık bir veri tabanı üzerinde tutulması ve yakın gerçek zamanlı işlenmesi konusundaki mevcut güçlükleri adresleyerek arazi örtüsünün sınıflandırılması için bir web servisi geliştirmişlerdir. Geliştirilen platformda Landsat 8 uydu verilerinin tutulması için büyük çok boyutlu dizilerin tutulmasını destekleyen dağıtık bir veri tabanı olan Rasdaman veri tabanı kullanılmıştır. Veri sorgulama için ise Açık Mekansal Konsorsiyum'un açık web kapsama işleme servisi (Web Coverage Processing Service - WCPS) kullanılmıştır. Arazi örtüsünün sınıflandırılması için Orfeo Toolbox ve OpenCV gibi uzaktan algılama ve bilgisayarla görme kütüphaneleriyle entegre, istatistiksel ve çoklu sınıf karar destek makinesi tabanlı iki aşamalı otomatik sınıflandırıcı geliştirilmiştir. İlk aşamada sınıflandırıcı 15 metre çözünürlükteki keskinleştirilmiş resimler ile 31 spektral sınıf kullanılarak eğitilmiş, ikinci aşamada ise geç sınıflandırma (post-classification) ve birleştirme prosedürü çalıştırılarak 10 yer örtüsü sınıfı ile çok zamanlı yer örtüsü haritası çıkarılmıştır. Yapılan çalışmada veri yönetimi Rasdaman veri tabanına, işleme ise WCPS sistemine bırakıldığından bu çalışma daha çok farklı teknolojilerin entegre edildiği bir yaklaşımdan öteye gidememiştir.

Büyüyen verilerin sadece tutulması değil aynı zamanda işlenmesi için de dağıtık yaklaşımlara ihtiyaç olmaktadır. Bu noktada geliştirilmekte olan birçok yeni model arasında sıvıran bir model olan MapReduce modeli, Apache Hadoop çatısının açık kaynak bir gerçekleştirmesidir. Bu model ile büyük veri kümelerinin çoklu bilgisayarlar (küme ya da bulut bilişim) üzerinde dağıtık olarak eş zamanlı işlenmesi sağlanır. Uzaktan algılama verilerinin algılayıcı teknolojilerindeki son gelişmeler dahilinde üstel büyümesi sonucunda analiz işlemlerinin performansını arttırmak amacıyla MapReduce modeli kullanan uzaktan algılayıcı uygulamaları araştırma konusu olmaya başlamıştır. Giachetta (2015) tarafından yapılan çalışmada coğrafi (geospatial) veri yönetimi ve işleme aracı sunan AEGIS çatısı

(framework) üzerine geliştirilen uzamsal ve uzaktan algılama verilerinin MapReduce görevleri olarak dağıtık sistemlerde yönetimi ve işlenmesini mümkün kılan bir çatı önerilmiştir. Krämer ve Senner (2015) tarafından büyük coğrafi veri kümelerinin bulut ortamında işlenmesi için çok modlu bir yazılım mimarisi önerilmiştir. Önerilen sistem modüler ve esnek bir yapıya sahip olmakla birlikte MapReduce, bellekte hesaplama ve etmen tabanlı programlama gibi tasarım modlarını destekler. Web tabanlı arayüzü vasıtasıyla alana özel dil (domain-specific language) kullanılarak yüksek seviye işleme için iş akışı tanımlanabilmektedir. Bulut üzerindeki işleme servisleri verilere dağıtık dosya sistemi (distributed file system - DFS) üzerinden ulaşır. Veri erişimini azaltmak için verinin bulunduğu merkezdeki bilgisayarlarda işlenmesi makul görülmüştür. Lin ve diğerleri (2013), Açık Mekansal Konsorsiyum'dan WCS, WMS, WPS gibi standart arayüzleri, HDFS depolama teknolojisini ve MapReduce'tan görüntü işleme işlemlerini kullanan bir çatı önermiştir. Bulut bilişim ortamı ise dağıtık işleme için temel alt yapıyı sağlamaktadır. Farklı veri kümeleriyle yapılan testlerde HDFS ile geleneksel yöntemler okuma/yazma performanslarına göre değerlendirilmiş ve büyük dosyalar (8-10 MB) için HDFS'in daha iyi performans gösterdiği belirtilmiştir. MapReduce ile alakalı çalışmalar Küme ve Bulut başlıkları altında daha detaylı bir şekilde sunulmuştur.

Ma, Wang, Liu ve Ranjan (2015), yüksek boyutlu veriler üzerinde gerçek zamanlı işleme yeteneği ihtiyacından doğan veri-yoğun hesaplama güçlüklerini inceleyerek uzaktan algılama sistemleri için verinin büyüklüğü dışındaki faktörler ele alınmıştır. Makalede önerilen deneysel model ile RS uygulamaları için ana faktörler belirlenip ölçülerek veri-yoğun endeksi oluşturulmuştur. Sonuç olarak ana faktörler ve veri-yoğun endeks arasında ilişkiler tanımlanarak veri-yoğun model formüle edilmiştir.

Dünya üzerinde tarımsal kuraklık verilerini istenildiği anda ve zamanında kullanıma hazır bulundurmak birçok zorluğa sahip acil bir ihtiyaç olarak görülmektedir. Deng ve diğerleri (2012) bu problemin çözümü için web servis teknolojisi, mekansal birlikte işlerlik, siber altyapı ve yakın gerçek zamanlı uzaktan algılama verilerini bir araya getirerek umut vadeden çözümler sunmaktadır. Yazarlar tarafından GADMFS olarak adlandırılan sistem, ileri mekansal web servis teknolojileriyle (WCS, WFS, vb.) küresel kuraklık izleme, tahmin ve analiz işlemlerini mevcut limitleri aşarak iyileştirmiştir. Diğer kuraklık sistemlerine göre farkı, daha iyi uzamsal ve zamansal çözünürlük ile erişim ve izleme sağlanmasıyla birlikte, ilgi alanı için çevrim içi ve talep üzerine kişiselleştirilmiş analizlerin yapılabilmesidir.

Coğrafi bilimler veri ve hesaplama yoğun olmasının yanında eş zamanlı erişim yoğunluğuna ve mekan-zamansal yoğunluğa sahiptir. Eş zamanlı erişim yoğunluğu, coğrafi sistemlerin aynı anda fazla sayıda kullanıcı tarafından eş zamanlı kullanılması sonucu oluşmaktadır. Mekan-zamansal yoğunluk ise coğrafi bilgilerin zamana bağlı olarak kayıt altına alınması anlamına gelmektedir. Yang ve diğerleri (2011) tarafından sunulan çalışmada elastik ve isteğe bağlı çalışan bulut bilişimin, coğrafi sistemlerin sahip olduğu yoğunluk problemlerine potansiyel bir çözüm olduğu belirtilmiştir. Bu bağlamda bulut bilişimin coğrafi bilimleri nasıl mümkün kılacağı ve mekan-zamansal temel kuralların bulut bilişime nasıl fayda sağlayacağı tartışılmıştır. Mekan-zamansal prensipler, kaynak kullanımını azaltarak veriyi ve hesaplama optimize etmede önemli rol oynar. Veriyi ve işleme birimlerini en iyi şekilde dizimler, veri taşımayı minimize eder, veri işleme için kaynak kapasitesini en iyi şekilde planlar ve sonuçları yaymak için optimize yaklaşımlar kullanır. Hesaplama alt yapılarının ve uygulamaların mekan-zamansal davranışlarının daha iyi anlaşılması için geniş kapsamlı araştırmalara ihtiyaç duyulmakla birlikte hesaplama kaynaklarının ve uygulamaların optimize planlaması kritik önem taşımaktadır.

2.2.1. Küme yaklaşımı

Donanım çözümlü geliştirmeler düşünüldüğünde en çok kullanılan yaklaşımlardan birisi de kullanıma hazır ticari (commercial off-the-shelf - COTS) bilgisayar ekipmanı tabanlı çözümlerdir. Bu yaklaşımda takım gibi birlikte çalışmak üzere bir miktar bilgisayardan bir küme (cluster) oluşturulur (Plaza, Du, Chang ve King, 2011). Çok sayıda CPU ile kurulan bu paralel sistemler, hem uzaktan algılayıcı hem de veri akıları kullanan gerçek/yakın gerçek zamanlı uygulamalar için iyi sonuçlar verse de bu sistemler hem oldukça pahalı (bakım maliyetleri düşünüldüğünde) hem de ölçeklenebilirliği belli bir kapasitenin üzerine çıkamamaktadır. Buna örnek olarak Cavallaro ve diğerleri (2015), çalışmalarında somut büyük veri problemi olarak görüntü tabanlı veri kümesi üzerinden yer örtüsü türlerinin sınıflandırılmasını ele almıştır. Çalışma kapsamında sınıflandırma için LibSVM'yi temel alan bir gerçekleştirme olan PiSVM kullanılmıştır. PiSVM paralel olarak kullanılırken birden fazla düğüm üzerinde iletişim için MPI kullanılmıştır. Paralel analiz için Almanya'daki Juelich Süper Hesaplama Merkezi'nde bulunan JUDGE kümesi kullanılmıştır. Paralel çalışan PiSVM'deki eğitim süresi, başarımlı düşmeksizin paralel Matlab çalıştırılmasına oranla önemli miktarda azalmıştır. İlgili çalışmada veri işleme için MPI ve PiSVM kütüphaneleri kullanılmakla birlikte veriyi yöneten bir yaklaşım

sunulmamıştır.

Büyük hiperspektral görüntüler, bellekte tutulabilmesi ve yüksek performanslı hesaplama sistemlerinde paralel bir şekilde işlenebilmesi için düzgün bir şekilde parçalanmaya ihtiyaç duyarlar. Toma, Panica, Zaharie ve Petcu (2012) tarafından yapılan çalışmada büyük görüntülerin dağıtılıp işlenmesi için birkaç farklı yaklaşım kullanılmıştır. Bu yaklaşımlar: (1) usta (master) işlemci görüntüyü okur ve parçaları köle (slave) işlemcilere dağıtır (scatter); (2) her işlemci ilgilendiği görüntü parçasını okur; (3) Bir düğüm bütün görüntüyü alarak paylaşılan belleğe koyar ve diğer düğümler ilgili parçayı buradan okuyarak RDMA vasıtasıyla lokale alır. Paralel gerçekleştirim için ise Fuzzy c-Mean ve otomatik morfolojik son üye çıkarma algoritmaları seçilerek BlueGene/P süper bilgisayar üzerindeki 1024 işlemci ile denenmiştir. Farklı sayıdaki işlemci ile farklı genişlik ve yükseklikteki parçalarla yapılan çalıştırmalarda elde edilen sonuca göre görüntünün parçalara ve işlemci sayısına göre iletişim süresinin değiştiği görülmüştür. Yapılan çalışmada görüntü verisi işlemci sayısına göre düz bir şekilde parçalanmakla beraber ağ topolojisinin farkındalığıyla daha sonra çalışma yapılacağı belirtilmiştir.

Sierra-Pajuelo, Paz-Gallardo ve Plaza (2015) tarafından yapılan çalışmada AVIRIS'ten alınan hiperspektral görüntüler üzerinde çok döngülü bir algoritma olan (3 döngülü) otomatik hedef üretme işlemi algoritması geliştirilerek denenmiştir. Makale özelinde ise ilgili algoritmanın OpenMP ve MPI kullanılan paralel versiyonları geliştirilerek sunulmuştur. İşlemciler arası iletişimi minimum seviyede tutmak için görüntüler uzamsal parçalanmıştır. OpenMP gerçekleştirmesi, Xeon 64 çekirdekli işlemcili, 1 TB NUMA paylaşılan bellekli platformda (s6030) ve Xeon 6 çekirdekli işlemcili, 24 GB bellekli düğüm üzerinde koşturulmuştur. İki farklı platformda 2, 4, 8 çekirdekli (çekirdek başına tek thread) koşturmalar s6030 üzerinde daha iyi sonuçlar elde edilmiştir. S6030 üzerindeki 16, 32, 64 çekirdekli koşturmalar ise 64 çekirdekli ile daha iyi sonuç alınsa da aşırı bir iyileşme gözlemlenmemiştir. MPI kullanılan deneylerde ise Xeon 6 çekirdekli işlemcili, 24 GB bellekli 16 düğümden oluşan platform kullanılmıştır. 2, 4, 8 ve 10 düğüm üzerinde tek thread ile testler gerçekleştirilmiştir. Daha fazla düğümde küme üzerinde performans kötüleştiği tespit edildiği için daha fazla düğüm denenmemiştir. 4, 8, 10 düğüm ile yapılan koşturmalar çok yakın sonuçlar elde edilmiştir ve bunun en büyük nedeni algoritmanın döngülü olması ile daha büyük verilerin kullanılmamış olmasıdır. Tüm koşturmalar sonucunda en iyi performansla sahip olan ise tek düğüm ve 10 thread ile gerçekleştirilen

MPI'dır. Bu çalışmada temel yaklaşım verinin yönetiminden ziyade ilgili algoritmaların OpenMP ve MPI kütüphaneleri kullanılarak paralelleştirilmesidir.

Valencia, Plaza, Martínez ve Plaza (2005), gerçek zamanlı analize ihtiyaç duyan uygulamalar için hiperspektral görüntüler üzerinde kullanılan birkaç algoritmanın (spektral ayrıştırma, morfolojik metot ve son üye çıkarımı) paralel olarak gerçekleştirilmesini sağlamıştır. Uzamsal olarak parçalanmış görüntüler işlemciler arası iletişimi engellemek amacıyla sınırdaki veriler tekrarlanarak işlemcilerle dağıtılmıştır. Fazlalık bilgi oransal olarak genel bilginin sadece 2,5% olduğundan iletişime oranla daha verimlidir. Önerilen sistemde n-boyutlu paralel bölümlendirme modülü (NDPP), I/O işlemlerinden sorumlu ana düğümdür ve aldığı görüntüyü diğer işlemcilerle dağıtır. Her işlemci ilgili parçayı lokalinde işler ve sonucu NDPP'ye döner. NDPP aldığı sonuçları bir araya getirerek final son üyeleri seçer. Geliştirilen paralel yaklaşım, NUMA paylaşılan bellek erişimine sahip SGI Origin 2000 kümesi (64 MIPS işlemci, her işlemci 4 Mb cache ve 12 GB belleğe sahip, 1,2 GBPS ağ) üzerinde ve Beowulf türü Thunderhead isimli küme (256 düğüm, her düğümde 2 x 2,4 GHz Xeon işlemci, 1 GB bellek, 2 GBPS ağ) üzerinde denenmiştir. Algoritmadaki iterasyon sayısı 1, 3, 5, 7 alınarak koşurmalar yapılmış ve iterasyon 7'de %90 doğruluk elde edilmekle birlikte 7'den fazla iterasyonda doğruluk açısından önemli bir iyileşme olmamıştır. Amdahl yasasına göre hesaplanan hızlanma 7. iterasyonda en iyidir. İşlemci sayısının daha da fazla artmasına rağmen paralel algoritmanın hızlanması artmaya devam etmemektedir. Bunun nedeni algoritmanın paralel olmayan kısmının, işlemci sayısına oranla daha baskın olmasıdır ve bu durum aşırı sayıda işlemci kullanıldığında paralelleştirmenin performansını düşürür. Yapılan çalışma sadece belirtilen kümeler üzerinde paralel algoritma gerçekleştirmesinden ibaret olmakla birlikte veri ve kaynağın yönetimine ilişkin bir çalışma yapılmamıştır.

J. Chen, Zheng ve H. Chen (2013) yaptıkları çalışmada uzaktan algılama verileri üzerinde yer örtüsü sınıflandırma işlemlerini hadoop kümesi üzerinde MapReduce tabanlı yaklaşımla hızlandıran bir yenilik önermiştir. Önerilen yaklaşımda ilk fazda ekstrem öğrenme makinesi birlik algoritması kararlı ve doğru şekilde map prosedüründe paralel olarak çalıştırılmıştır. İkinci fazda ise MapReduce kullanılarak daha verimli ve ölçeklendirilebilir bir ortam sağlanmıştır. Yapılan çalışma, literatürdeki MapReduce üzerinde geliştirilen ilk ekstrem öğrenme makinesi olsa da veri yönetimi içermemekle beraber veri işleme kısmı veri yoğun problemlerde yavaşlayan MapReduce'ün kabiliyetine bırakılmıştır.

2.2.2. Bulut yaklaşımı

Büyük coğrafi veriler söz konusu olduğunda bulut bilişim en kuvvetli büyük veri tekniklerinden biri olarak öne çıkmaktadır (Gao ve diğerleri, 2017). Sanallaştırma teknolojilerinin hesaplama altyapılarını fiziksel bir bilgisayarmışçasına sanallaştırarak esnek işleme, bellek ve sürücü sağlayabilmesi süper hesaplama kavramını daha erişilebilir ve uygun fiyatlara alınabilir hale getirmiştir (O'Driscoll, Daugelaite ve Sleator, 2013). Veri yönetimi ve analiz işlemleri için çoklu bilgisayarlardan oluşan bir altyapı sunan bulut kavramının kullanılması yüksek ölçeklenebilirlik ve kullanılabilirlik, hataya toleranslılık, performanslılık anlamında büyük kolaylıklar sağlamaktadır. Özellikle yeni nesil uzaktan algılayıcıların yakın gerçek zamanlı üretebildiği verilerden bilgi çıkarımı yapılması gereken kritik uygulamalar düşünüldüğünde bulut bilişim teknolojilerinin yüksek performanslı hesaplama amacıyla kullanılması büyük önem kazanmıştır (Qu, Zhang, Lin ve Chen, 2013). Ayrıca bulut bilişim altyapısı, analiz işlemlerinin performanslı yapılmasının yanında büyük verinin depolanması için de verimli bir platform oluşturma yeteneğine sahiptir. Böylece bu teknolojinin kullanılmasıyla birlikte küme sistemleri gibi bakımı pahalı hesaplama donanımları, ayrılmış alan ve yazılım ihtiyaçları ortadan kaldırılabilmektedir (Kambatla, Kollias, Kumar ve Grama, 2014).

Coğrafi mekansal uygulamalarda önemli rolü olan nokta bulutu verileri, farklı analiz türleri için zengin bir bilgi içeriğine sahiptir. Makine öğrenmesi ile farklı kategorilere ayrılarak anlamlandırılan nokta verilerinin hacim olarak hızlı artışı sonucu özellikle işlenmesinde büyük güçlükler meydana gelmektedir. Liu ve Boehm (2015) tarafından yapılan çalışmada nokta verilerinin sınıflandırılması büyük veri olarak ele alınıp Apache Spark kullanılarak işlenmiştir. Paralel programlama arayüzü kolay olduğu için Apache Spark'ın sağladığı Python API kullanılmış ve kodlanan metotların tamamı Python ile yazılmıştır. Geliştirilen uygulama, Amazon EC2 servisinden alınan bir usta ve 10 köle düğüm ile kurulan Spark kümesi üzerinde denenmiştir. Deneysel sonuçlar büyük nokta verilerinin işlenmesi için umut verici olarak değerlendirilse de çalışmada verinin yönetimi ve kaynakların optimizasyonu için bir çalışma yapılmamıştır.

NASA'ya ait EO-1 uydusundan hiperspektral teknolojisiyle alınan yeryüzü görüntüleri üzerinde spektral ayırma yaklaşımı kullanılarak veriler analiz edilir (Cappelaere ve diğerleri, 2013). Yüksek spektral boyutluluğa bağlı olarak bu yaklaşım oldukça zaman alan bir

işlemdir. Cappelaere ve diğerleri (2013) tarafından geliştirilen bulut uygulamasında boyut indirgeme, otomatik son üye tanılama ve miktar tahminleme adımlarından oluşan hiperspektral ayrıştırma zinciri gerçekleştirilmiştir. Geliştirilen ayırma zincirinin daha sonra NASA SensorWeb süitinde bulut üzerinde çalışan bir görüntü işleme çatısı ile canlıya alınacağı belirtilmiştir. Önerilen sistem sadece adımlara ayrılan algoritmanın zincir olarak bulut üzerinde koşan hazır bir görüntü işleme çatısında çalıştırılmasından ibarettir.

Cui, Wu ve Zhang (2010) tarafından yazılan makalede büyük miktardaki uzamsal verilerin işlenmesini sağlayacak bir bulut bilişim modeli anlatılmıştır. Bu modelde bulut istasyonu görevlerin dağıtımı ve yönetiminden sorumludur. Bulut işleme sunucusu ise dağıtılan görevi yapmak amacıyla depolama sisteminden ilgili görüntüleri alıp işler. Bulut temelli uzamsal veri işleme ise ön işleme, bilgi çıkarımı/tematik uygulama ve sentezleme adımları olarak ele alınmıştır. Önerilen modelin geliştirilerek denenmesi için uzun bir zamana ihtiyaç olduğundan deneyler dijital fotogrametrik örgü sistemi (DPGrid) üzerinde gerçekleştirilmiştir. Orta büyüklükteki bir şehre ait 6000 görüntü ile yapılan mozaikleme, geleneksel yöntemlerle 10 personelin bir yıldan fazla çalışmasıyla yapılabilmekteyken 8 sunuculu DPGrid ile 15 gün sürmektedir. Bu çalışma buluta göre daha farklı bir teknoloji olan örgü sistemi üzerinde geliştirildiğinden ancak bir dağıtık işleme simülasyonu seviyesinde kalmıştır.

MapReduce modeli, sadece küme yaklaşımında değil aynı zamanda bulut üzerinde yapılan uzaktan algılama uygulamalarında da dikkat çekmeye başlamıştır. Wu ve diğerleri (2012) tarafından yapılan çalışmada görüntü işleme uygulamalarında MapReduce üzerinde çalışacak yenilikçi bir planlayıcı önerilmiştir. Önerilen sistemde Reduce aşamasında boşa geçen zaman içinde dinamik olarak bir sonraki göreve geçilir. Önerilen planlayıcı, 2D-3D uygulamalarda denenmiş ve geleneksel MapReduce'e göre iyileştirme sağladığı gözlemlenmiştir.

Guo ve diğerleri (2010), uzaktan algılama görüntülerinin işlenmesi için MapReduce tabanlı, OpenRS-Cloud isimli, bulut teknolojisini temel alan bir platform geliştirmiştir. İlgili çalışmada geliştirme desenleri, hesaplama modeli, veri yönetimi ve servis modeli konuları üzerinde durulmuştur. OpenRS-Cloud, algoritma geliştirmelerinin basit bir şekilde yapılabilmesi için programlama soyutluğunu arttırmak ve programlama karmaşıklığını azaltmak amacıyla arayüz-yönelimli ve nitelik-yönelimli eklenti algoritma sistemine

sahiptir. Hesaplama modeli olarak MapReduce modelini temel alan ayrıştır-birleştir isimli bir model önerilmiştir. Bu model ile MapReduce’ta değer olarak görüntünün bir alanının birleştirilmesiyle ortaya çıkan verimlilik probleminin çözülmesi sağlanmaktadır. Bu yeni model görev parçalama, alt görev işleme ve alt görev birleştirme adımlarından oluşur. Görev parçalama, MapReduce’taki ayrıştırma işlemine karşılık gelen görevi alt görevlere ayırma işlemidir. Alt görev işleme ise Map fazına karşılık gelmekle birlikte değer algoritma girdi/çıkıtlı niteliklerini taşıyan xml verisidir (anahtar geliştirici yerine platform tarafından belirlenmeye devam ediyor). Alt görev birleştirmede ise Reduce fazındaki gibi birleştirme yapılıp fakat burada aynı anahtara sahip alt görevleri birleştiren sadece bir Reduce görevi vardır. İlgili çalışmada büyük veriye dönüşen uzaktan algılama verilerinin bulut üzerinde nasıl tutulacağı ve yönetimine değinilmemekle birlikte MapReduce modelinin kabiliyetine bırakılmıştır.

Başka bir çalışmada ise Guo, She ve Zhu (2014) OpenRS’yi kullanan ve vakitliliğe dayanan acil durum uygulamaları için uzaktan algılama görüntülerinin isteğe bağlı hesaplanmasını sağlayacak bir şema sunmuştur. Bu bağlamda uydu görüntülerinin özel bulut bilişim platformu üzerinde işlenebilmesi için bir zincir modeli önerilmiştir. Önerilen zincir görüntü parçalarından ve işleme operasyonlarından oluşan bir sıralamadır. Zincir oluşturulurken dosya okuma, algoritma ve görüntüleme düğümleri kullanılmaktadır. Bir görüntü parçası tanımlanan düğümlere girdi ve çıktı olabilmektedir. Oluşturulan zincir sıralı bir şekilde işlenirken her bir düğümün çıktısı bir sonrakine girdi olmaktadır. Önerilen sistem OpenRS’nin eklenti modelini temel alarak geliştirilmiştir. Burada önerilen sistem, veri yönetimini dağıtık veri tabanı ile dağıtık dosya sistemine bıraktığından ve bu sistemlerin RS algoritmalarının karmaşıklığını göz ardı etmesi nedeniyle buluttaki mevcut kaynağı verimli kullanamayacaktır.

Gowda ve Hulyal (2015) tarafından önerilen yaklaşımda MapReduce’u baz alan bir uzaktan algılama görüntü işleme sistemi bulut üzerinde kullanılmıştır. Önerilen yapıda verinin depolanması için HDFS, bulut üzerindeki verilerin analizi için MapReduce ve düğümlerde ise GPU ile POSIX iplikleri kullanılmıştır. POSIX bir grup görüntü verisi ve sistem belleği arasındaki I/O işlemleri için kullanılırken, diğer bir grup tamponu GPU belleğine taşımaktan ve tampon havuzundan alınanları işlemekten sorumludur. İplikler arasında senkronizasyon için kilitleme mekanizmasıyla semaforlar kullanılır. Böylece I/O işlemleri ve analiz işlemleri eş zamanlı yürütülmüş olur. Önerilen sistem bulut üzerinde GPU kullanılması açısından

umut vaad edici olsa da MapReduce üzerinde tekrar eden I/O işlemlerini engelleyen bir mekanizması olmadığından istenilen yüksek performansı sağlayamayacaktır.

P. Wang, J. Wang, Chen ve Ni (2013) yaptıkları çalışmada bulut bilişimin sadece RS verilerinin yönetilmesi için değil aynı zamanda işlenmesi için de verimli bir yaklaşım olduğunu göstermiştir. Özellikle meteorolojik izleme ve doğal afet uyarımı gibi hızlı işleme gerektiren uzaktan algılama uygulamaları için önerilen mimari, büyük verinin yönetimini ve dağıtık olarak paralel işlemeyi bir araya getirmektedir. Bu mimari, görev planlama, iş konuşlandırma ve Hadoop modüllerinden oluşmaktadır. Görev planlama, kullanıcıdan istekleri alan ve bir kuyruğa atarak yöneten modüldür. İş konuşlandırma, kuyruktan gelen işin konuşlandırılmasını sağlamakla birlikte GDAL kütüphanesi ile görüntü bilgilerinin alınarak buluttaki veri düğümlerine bilgi dosyalarının yollanmasından sorumludur. Son olarak Hadoop modülü, HDFS'teki dosyaların MapReduce kütüphanesi ile işlenmesini sağlar. Modelin denenmesi için maksimum olabilirlik sınıflandırması algoritması seçilerek standart bilgisayar, MPI kullanılan dağıtımli hesaplama ve MapReduce kullanılan bulut üzerinde koşturulmuştur. IO işlemlerinin önemsenmediği ölçümlerde 4 düğümlü MPI'nın, 4 düğümlü MapReduce'tan biraz daha fazla performanslı olmasına karşın veri parçalama gibi konularda MapReduce'un otomatik olması bulut yapısının diğer hesaplama modellerine göre daha avantajlı olabileceğini göstermektedir. Yapılan çalışmada veri yönetimine değinilmesine karşın yönetim işlemini MapReduce'e bırakmaktan öteye bir çalışma gerçekleştirilmemiştir.

Lin ve diğerleri (2013), Açık Mekansal Konsorsiyum'dan WCS, WMS, WPS gibi standart arayüzleri, HDFS depolama teknolojisini ve MapReduce'tan görüntü işleme işlemlerini kullanan bir çatı önermiştir. Uzaktan algılayıcı görüntüleri için makalede önerilen bulut bilişim çatısı, 4 ana bileşenden oluşur. Kullanıcı, UAV Görüntü Yönetimi vasıtasıyla hava görüntülerini alabilir ve yükleyebilir. WebGIS hava görüntülerini web sayfasındaki dinamik katmana koyarak gösterebilmeyi sağlar. Veri Analizi modülü görüntü verilerinin tutulması ve işlenmesinden sorumludur. Bulut bilişim ortamı ise dağıtık işleme için temel alt yapıyı sağlamaktadır. Önerilen mimaride görüntüler HDFS üzerindeki veri ambarında tutulmakla birlikte MapReduce tekniği kullanılmaktadır. Görüntü verilerinin sınıflandırılması için Hadoop uzantısı olan Mahout projesi kullanılmıştır. Görüntü işleme modülü MapServer ile entegre olarak çalışır. WebGIS görselleştirme amacıyla OpenLayers vasıtasıyla WMS ve WCS arayüzleriyle MapServer'a bağlanarak görüntüleri alırken WPS arayüzü ile görüntü

analizi modülünden sonuçları alarak görselleştirir. Önerilen mimari birçok teknolojiyi bir araya getirmesine rağmen, büyük görüntü verilerini işlemesindeki performansı ile alakalı deneysel bir sonuç sunulmamıştır.

Rathore, Ahmad, Paul ve Wu (2016) uzaktan algılamalı uydu uygulamaları için 3 ana birimden oluşan (veri alma, veri işleme ve analiz-karar birimi) gerçek zamanlı büyük veri analitiği mimarisi önermişlerdir. Önerilen mimari, bilinen uydu verileri üzerinde yer ve deniz bulunması için hazırlanan algoritmalar ile Hadoop ekosistemi üzerinde HDFS ve MapReduce kullanılarak denenmiştir. Sonuçlara göre düz java üzerinde çalışan sistemin performansı, Hadoop'a göre 200 MB'tan büyük veriler için önemli ölçüde düşmeye başlamaktadır. Büyük görüntü verilerinin parçalanarak tutulması, işlenmesi ve sonuçların birleştirilmesi tamamen Hadoop tarafından gerçekleştirilmiştir. Bu durum metin tabanlı işlemlere özel geliştirilen HDFS, büyük görüntüler ile birlikte düşünüldüğünde komşu parçalara ihtiyaç duyulduğunda ortaya çıkacak iletişim maliyetinden dolayı performansı aşağı çekecektir.

Üretilen verilerdeki üstel artışla birlikte, çoğu büyük çevrim içi sistem coğrafi olarak dağıtılmış bulutlar üzerine geçirilmeye başlandı (Narayanan, Kansal, Sivasubramaniam, Urgaonkar ve Govindan, 2014). Coğrafi olarak dağıtılmış bulutlar, dağıtık olmasından dolayı en kritik neden olan gecikmeyi azaltmak için müşteriye hizmet ederken en yakın bulut sitesini kullanır. L. Wang ve diğerleri (2018) pipsCloud'u RS büyük verileri için yüksek performanslı bir bulut ortamı olarak sundu ve önerilen sistem Çin'deki bağlı birkaç ulusal uydu veri merkezinden oluşan çok veri merkezli işbirlikçi süreç sisteminde kullanıldı. Bu sistem bulut bilişim modelleriyle, karmaşık dağıtılmış algoritma zincirlerini işlemek için uyarlanabilir bir uzaktan algılama iş akışı yönetim sistemi sağlamanın yanı sıra, yüksek boyutlu RS verilerini yönetmek için paralel dosya sistemleri kullanır ve düzensiz RS verilerine erişmek için saniye başına giriş/çıkış işlemleri (IOPS) performansını optimize etmek için arabirimler kullanır. Veri işleme için kullanıcı tarafından özelleştirilmiş zamanlama kuralları ile hizmet kalitesi (QoS) tabanlı optimum kaynak zamanlama modelinden yararlanılmıştır. Ayrıca, RS verilerinin dağıtılmış depolanması için uygulamaya duyarlı bir veri düzeni kullanılmıştır. Bununla birlikte, çalışma, farklı algoritma erişim modeli türleri için verimli veri erişimine rağmen, veri merkezlerinde veri elde etmek için kaynak optimizasyonunu ele almamıştır.

RS verilerinin özelliklerini tam olarak karşılamak amacıyla Xu ve diğerleri (2021), uzaktan algılama ve büyük veri teknolojisi arasındaki boşluğu kapatmak için büyük veri teknolojilerini birleştiren bilimsel bir çerçeve önerdi. Önerilen çerçeve, hem uzaktan algılama verilerini analiz sonuçlarına dönüştürme sürecinin tamamını kapsamaktadır. Spark, Hadoop, HBase, Zookeeper, PostgreSQL, Thrift ve Python teknolojilerini bir araya getiren çerçeve umut verici bir çözüm olsa da bazı basit algoritmalar ile çalışan bir prototiptir. Modüler yapısı sayesinde önerilen modeldeki yapıların gelecekte geliştirileceği belirtilmiştir.

Lunga ve diğerleri (2020) makine öğrenimi sürecini büyük miktarlarda RS görüntüsü ile daha hızlı hale getirmek için yeni bir uzaktan algılama veri akışı (RESFlow) uygulamıştır. Bu çalışmanın birincil katkısı, basit modellere uyacak şekilde büyük miktarda veriyi tek tip dağılımlara bölmektir. RESFlow, kapsamlı uzaktan algılama görüntülerinde derin öğrenme çıkarımını hızlandırmak için Apache Spark ve GPU'yu kullanır. Önerilen çerçeve ayrıca, tek bir çalışana atanan birden çok yürütücü arasında kaynak kullanımını en üst düzeye çıkarmak için bir stratejiyi de bütünleştirir. İş akışı, piksel etiketleme iş yüklerinde derin öğrenme çıkarımı için işlem hızlarını önemli ölçüde artıran NVIDIA DGX platformları kullanılarak gösterilmiştir. 21 028 TB görüntü verisinin işleme süresi, 28 günlük iş yükünden yalnızca 21 saate düşürülmüştür.

Özellikle geniş alan ağı üzerinde kullanılan Nesnelerin interneti (Internet of Things - IoT), uç bilişim (edge computing), veri akışı (data streaming) ve benzeri teknolojilerin yaygınlaşmasıyla birlikte, düşük gecikme süresi ve yüksek verim elde etmenin önemi, sensörler ve veri merkezleri arasındaki değişken ağ durumları nedeniyle artmaktadır (Jonathan, Chandra ve Weissman, 2020; Stanford, 2020; Sharma ve Chandra, 2020; Bergui, Najah ve Nikolov, 2021; Abdullah, Peng ve Tak, 2021). Apache çözümleri olan Storm, Kafka ve Flink gibi çerçeveleri kullanan çalışmalar, anlık olarak işlenmesi gereken akış verileri için daha uygundur. Uzaktan algılama algoritmaları doğası gereği görüntü piksellerindeki komşuluklara ihtiyaç duymakla birlikte, bu çerçevelerdeki parçalama stratejileri çok büyük görüntülerin kullanıldığı uzaktan algılama uygulamaları için pek uygun değildir. Bu tez kapsamında önerilen sistem, coğrafi olarak dağıtılmış bir bulutta RS büyük verilerini yönetmek için iki aşamalı bir optimizasyon modeliyle her bir görüntüyü bir veri merkezine atamaya çalışır. Ağ ve kaynak metriklerinin bazı kısıtları karşılayamaması durumunda, kısıtları karşılayabilmek için bazı görüntülerin parçalanmasına ve birden fazla

veri merkezine atanmasına izin verir. Böylece sistem, daha yüksek performans sağlamak için işlemci, depolama, bellek ve ağ kaynaklarını uçtan uca optimize edecek şekilde anlık uygulama gereksinimlerini karşılayacaktır (Tso, Jouet ve Pezaros, 2016). Ayrıca RS verilerini yeterli kaynağa sahip daha uygun bulut sitesine yönlendirmek, analiz işlemlerinin işleme süresi üzerinde azalan bir etki yaratacaktır.

Çizelge 2.1. Uzaktan algılama alanındaki yerleşik ve dağıtık yaklaşımlar

Kaynak	Yayın Tarihi	Kullanılan Teknolojiler	Açıklama	Eksiklikler
Valencia, Plaza, Martínez ve Plaza (2005)	2005	Küme	İki farklı küme üzerinde spektral ayrıştırma, morfolojik metot ve son üye çıkarımı paralel olarak gerçekleştirilmiştir. İşlemciler arası iletişimi engellemek amacıyla sınırdaki veriler tekrarlanarak işlemcilerle dağıtılmıştır.	Çalışma sadece belirtilen kümeler üzerinde paralel algoritma gerçekleştirmesinden ibaret olmakla birlikte veri ve kaynağın yönetimine ilişkin bir çalışma yapılmamıştır.
Plaza ve diğerleri (2006)	2006	FPGA, Küme, Dağıtık Hesaplama	Hiperspektral görüntü işleme algoritmaları 3 farklı teknoloji ile paralel olarak geliştirilmiş ve işleme süresi azaltılmıştır.	Görüntü verileri büyük veri olarak ele alınmamıştır.
Setoain, Prieto, Tenllado ve Tirado (2008)	2008	GPU	Son üye çıkarımı algoritması GPU tabanlı gerçekleştirilmiştir.	Tek bir algoritmaya özel çalışılmıştır. Görüntü verileri çok büyük değildir (1-512 MB).
Shan, X. S. Wang ve Z. S. Wang (2010)	2010	FPGA	FPGA üzerinde bellek kullanımı %75 azaltılmıştır. İşleme hızı 3 kat arttırılmıştır.	Önerilen yaklaşım sadece yerleşik kullanım durumlarını içermektedir.

Çizelge 2.1. (Devam) Uzaktan algılama alanındaki yerleşik ve dağıtık yaklaşımlar

Kaynak	Yayın Tarihi	Kullanılan Teknolojiler	Açıklama	Eksiklikler
Cui, Wu ve Zhang (2010)	2010	Bulut	Önerilen bulut modelinde bulut istasyonu görevlerin dağıtımı ve yönetiminden sorumludur; bulut işleme sunucusu ise dağıtılan görevi yapmak amacıyla depolama sisteminden ilgili görüntüleri alıp işler. Bulut temelli uzamsal veri işleme ise ön işleme, bilgi çıkarımı/tematik uygulama ve sentezleme adımları olarak ele alınmıştır.	Önerilen modelin geliştirilerek denenmesi için uzun bir zamana ihtiyaç olduğundan deneyler dijital fotogrametrik örgü sistemi (DPGrid) üzerinde gerçekleştirilmiştir. Bu yüzden geliştirilen sistem ancak bir dağıtık işleme simülasyonu seviyesinde kalmıştır.
Guo ve diğerleri (2010)	2010	Bulut	MapReduce tabanlı, OpenRS-Cloud isimli, bulut teknolojisini temel alan bir platform geliştirmiştir. İlgili çalışmada geliştirme desenleri, hesaplama modeli, veri yönetimi ve servis modeli konuları üzerinde durulmuştur.	Veri yönetiminde veri yerelliği dikkate alınmamıştır.
Liu ve diğerleri (2011)	2011	GPU, POSIX Thread	GPU ve POSIX thread teknolojilerini temel alan dual-paralel bir işleme mekanizması geliştirilmiştir. POSIX thread ile verinin GPU belleğine taşınma süresi düşürülmüştür.	
Qu, Zhang, Y. Chen, H. Chen ve Lin (2012)	2012	GPU, Küme	SAM algoritması CUDA ve MATLAB kullanılarak GPU üzerinde çalıştırılmıştır.	SAM algoritması piksel bazlı bir algoritma olduğundan "embarrassingly parallel" bir paralelleştirme yapılmıştır.

Çizelge 2.1. (Devam) Uzaktan algılama alanındaki yerleşik ve dağıtık yaklaşımlar

Kaynak	Yayın Tarihi	Kullanılan Teknolojiler	Açıklama	Eksiklikler
Bernabe, López, Plaza ve Sarmiento (2012)	2012	GPU	Otomatik hedef bulma ve sınıflandırma algoritması GPU üzerinde gerçek zamanlı olarak ilk kez gerçekleştirilmiştir. Düşük bellek transfer zamanı için veri düzeni dikkate alınmıştır.	Kullanılan görüntü verileri küçüktür (40-120 MB).
Deng ve diğerleri (2012)	2012	Dağıtık Hesaplama	İlgi alanı için çevrim içi ve talep üzerine kişiselleştirilmiş analizlerin yapılabilmesini sağlayan bir kuraklık izleme uygulaması önerilmiştir.	İlgili sistem için performans konusunda bir sonuç sunulmamıştır.
Toma, Panica, Zaharie ve Petcu (2012)	2012	Küme	Büyük görüntülerin dağıtık olarak okunup işlenmesi için 3 farklı yaklaşım kullanılmıştır. Fuzzy c-Mean ve otomatik morfolojik son üye çıkarma algoritmaları BlueGene/P süper bilgisayar üzerinde paralel olarak çalıştırılmıştır.	Çalışmada ağ topolojisinin farkındalığıyla daha iyi sonuç elde edilebileceğine değinilmiş fakat bu konuda çalışma yapılmamıştır.
Wu ve diğerleri (2012)	2012	Bulut	Görüntü işleme uygulamalarında MapReduce ile bulut üzerinde çalışacak yenilikçi bir planlayıcı önerilmiştir. Önerilen sistemde Reduce aşamasında boşa geçen zaman içinde dinamik olarak bir sonraki göreve geçilerek iyileşme sağlanmıştır.	Verinin yönetilmesi HDFS'in becerisine bırakılmıştır.
Bernabé ve diğerleri (2013)	2013	GPU, Çok Çekirdekli İşlemci	Hiperspektral ayrıştırma zinciri 2 farklı GPU ile 13 ve 27 kat hızlandırılmıştır. Ayrıca 2 farklı çok çekirdekli işlemci ile 3,5 ve 4 kat hızlandırılmıştır.	Sadece ayrıştırma zinciri ele alınarak bir çalışma yapılmıştır.

Çizelge 2.1. (Devam) Uzaktan algılama alanındaki yerleşik ve dağıtık yaklaşımlar

Kaynak	Yayın Tarihi	Kullanılan Teknolojiler	Açıklama	Eksiklikler
X. Wu ve diğerleri (2013)	2013	GPU	Piksel saflığı indeksi algoritması GPU ile ENVI yazılımındaki oranla 769 kat hızlandırılmıştır.	Tek bir algoritmaya özel çalışılmıştır.
Lin ve diğerleri (2013)	2013	Bulut	WCS, WMS, WPS gibi standart arayüzleri, HDFS depolama teknolojisini ve MapReduce'u kullanan bir çatı önerilmiştir.	Kullanılan görüntü verileri çok küçüktür (8-10 MB). Büyük görüntü verilerinin işlenmesindeki performansıyla alakalı deneysel bir sonuç sunulmamıştır.
J. Chen, Zheng ve H. Chen (2013)	2013	Küme	Yer örtüsü sınıflandırma işlemleri hadoop kümesi üzerinde MapReduce tabanlı yaklaşımla hızlandırılmıştır. İlk fazda ekstrem öğrenme makinesi birlik algoritması map prosedüründe paralel olarak çalıştırılmıştır. İkinci fazda ise MapReduce kullanılarak daha verimli ve ölçeklendirilebilir bir ortam sağlanmıştır.	Çalışma veri yönetimi içermemekle beraber veri işleme kısmı veri yoğun problemlerde yavaşlayan MapReduce'un kabiliyetine bırakılmıştır.
Cappelaere ve diğerleri (2013)	2013	Bulut	Bulut uygulamasında hiperspektral ayrıştırma zinciri gerçekleştirilmiştir.	Önerilen sistem sadece adımlara ayrılan algoritmanın zincir olarak bulut üzerinde koşan hazır bir görüntü işleme çatısında çalıştırılmasından ibarettir.

Çizelge 2.1. (Devam) Uzaktan algılama alanındaki yerleşik ve dağıtık yaklaşımlar

Kaynak	Yayın Tarihi	Kullanılan Teknolojiler	Açıklama	Eksiklikler
P. Wang, J. Wang, Chen ve Ni (2013)	2013	Bulut	Özellikle meteorolojik izleme ve doğal afet uyarımı gibi hızlı işleme gerektiren uzaktan algılama uygulamaları için önerilen mimari, büyük verinin yönetimini ve dağıtık olarak paralel işlemeyi bir araya getirmektedir. 4 düğümle testlerde MPI'ın MapReduce'tan daha iyi olmasına karşın veri parçalama gibi konularda MapReduce'un otomatik olması bulut yapısının diğer hesaplama modellerine göre daha avantajlı olabileceğini göstermektedir.	Yapılan çalışmada veri yönetimine değinilmesine karşın yönetim işlemini MapReduce'e bırakmaktan öteye bir çalışma gerçekleştirilmemiştir.
Torti ve diğerleri (2014)	2014	GPU, Çok Çekirdekli İşlemci, DSP	Boyut indirgeme açısından önemli bir algoritma olan hiperspektral alt uzay tanıma algoritması CPU, GPU ve DSP üzerinde gerçekleştirilmiştir.	DSP teknolojisi görüntü işleme için uygun bir teknoloji değildir. Kullanılan görüntü verileri küçüktür (128 MB).
Chang ve diğerleri (2014)	2014	GPU	En yakın özellik uzayı algoritmasının yenilikçi bir türevini GPU mimarisi kullanarak hızlandırılmıştır.	Tek bir algoritmaya özel çalışılmıştır.
Guo, She ve Zhu (2014)	2014	Bulut	Acil durum uygulamalarında uydu görüntülerinin özel bulut bilişim platformu üzerinde işlenebilmesi için bir zincir modeli önerilmiştir. Önerilen sistem OpenRS'in eklenti modelini temel alarak geliştirilmiştir.	Dağıtık dosya sisteminin RS algoritmalarının karmaşıklığını göz ardı etmesi nedeniyle buluttaki mevcut kaynak verimli kullanılamamıştır.

Çizelge 2.1. (Devam) Uzaktan algılama alanındaki yerleşik ve dağıtık yaklaşımlar

Kaynak	Yayın Tarihi	Kullanılan Teknolojiler	Açıklama	Eksiklikler
Z. Wu ve diğerleri (2015)	2015	GPU	Aralıklı temsili sınıflandırıcıların GPU üzerinde gerçekleştirilmesiyle 130 kat hızlı sonuç üretilmiştir.	Tek bir algoritmaya özel çalışılmıştır.
Karantzas, Bliziotis ve Karmas (2015)	2015	Dağıtık Veri Tabanı	Çevrim içi olarak arazi örtüsünün sınıflandırılması için bir web servisi geliştirilmiştir. Orfeo Toolbox ve OpenCV kütüphanelerine entegredir.	Sadece arazi örtüsü için sınıflandırma ele alınmıştır. Önerilen yaklaşımın performansı Rasdaman dağıtık veri tabanının becerisine bağlıdır. Performans konusunda bir çalışma yapılmamıştır.
Giachetta (2015)	2015	Dağıtık Hesaplama	AEGIS çatısı üzerinde RS verilerinin MapReduce görevleri olarak dağıtık sistemlerde yönetimi ve işlenmesini mümkün kılan bir çatı önerilmiştir.	Verilerin işlenmesi MapReduce çatısının becerisine bırakılmıştır. Performans konusunda bir çalışma yapılmamıştır.
Krämer ve Senner (2015)	2015	Bulut	Büyük coğrafi verilerin bulut ortamında işlenmesi için çok modlu bir yazılım mimarisi önerilmiştir. MapReduce, bellekte hesaplama ve etmen tabanlı programlama gibi tasarım modları desteklenmiştir. Web tabanlı arayüzü vasıtasıyla iş akışı tanımlanabilir. Bulut üzerindeki işleme servisleri verilere dağıtık dosya sistemi üzerinden ulaşır.	Önerilen mimari için herhangi bir gerçekleştirme yapılmamıştır.

Çizelge 2.1. (Devam) Uzaktan algılama alanındaki yerleşik ve dağıtık yaklaşımlar

Kaynak	Yayın Tarihi	Kullanılan Teknolojiler	Açıklama	Eksiklikler
Cavallaro ve diğerleri (2015)	2015	Küme	JUDGE kümesi üzerinde yer örtüsü türlerinin sınıflandırılması MPI vasıtasıyla paralelleştirilmiştir. Eğitim süresi, başarımlar düşmeksizin paralel Matlab çalıştırılmasına oranla önemli miktarda azalmıştır.	Büyük verinin yönetilmesi konusunda bir yaklaşım sunulmamıştır.
Sierra-Pajuelo, Paz-Gallardo ve Plaza (2015)	2015	Küme	Küme üzerinde 3 döngülü otomatik hedef üretme işlemi algoritması OpenMP ve MPI kullanılarak geliştirilmiştir. Çalışmada en iyi sonuç algoritmanın çok döngülü olmasından dolayı 1 düğüm 10 thread ile çalıştırılan MPI ile alınmıştır.	Çalışmanın temel yaklaşımı verinin yönetiminden ziyade ilgili algoritmaların OpenMP ve MPI kütüphaneleri kullanılarak paralelleştirilmesidir.
Liu ve Boehm (2015)	2015	Bulut	Coğrafi uygulamalar için nokta verilerinin sınıflandırılması büyük veri olarak ele alınıp Apache Spark kullanılarak işlenmiştir. Apache Spark'ın sağladığı Python API kullanılmıştır. Amazon EC2 servisinden alınan bir usta ve 10 köle düğüm ile kurulan Spark kümesi üzerinde denenmiştir.	Çalışmada verinin yönetimi ve kaynakların optimizasyonu için bir çalışma yapılmamıştır.
Gowda ve Hulyal (2015)	2015	Bulut, GPU	Önerilen yapıda RS verilerinin depolanması için HDFS, bulut üzerindeki verilerin analizi için MapReduce ve düğümlerde ise GPU ile POSIX iplikleri kullanılmıştır.	Önerilen sistem bulut üzerinde GPU kullanılması açısından umut vaat edici olsa da MapReduce'un tekrar eden I/O işlemleri performansı düşürecektir. Önerilen sistem için performans çalışması sunulmamıştır.

Çizelge 2.1. (Devam) Uzaktan algılama alanındaki yerleşik ve dağıtık yaklaşımlar

Kaynak	Yayın Tarihi	Kullanılan Teknolojiler	Açıklama	Eksiklikler
Rathore, Ahmad, Paul ve Wu (2016)	2016	Bulut	Önerilen mimari, bilinen uydu verileri üzerinde yer ve deniz bulunması için hazırlanan algoritmalar ile HDFS ve MapReduce kullanılarak denenmiştir. Sonuçlara göre düz java üzerinde çalışan sistemin performansı, Hadoop'a göre 200 MB'tan büyük veriler için önemli ölçüde düşmeye başlamaktadır.	Metin tabanlı işlemlere özel geliştirilen HDFS, büyük görüntüler ile birlikte düşünüldüğünde komşu parçalara ihtiyaç duyulduğunda ortaya çıkacak iletişim maliyetinden dolayı performansı aşağı çekecektir.
L. Wang ve diğerleri (2018)	2018	Bulut, Dağıtık Bulut	PipsCloud RS büyük verileri için yüksek performanslı bir bulut ortamı olarak sunuldu ve Çin'de birkaç ulusal uydu veri merkezinden oluşan coğrafi dağıtık sistemde kullanıldı. Bu sistem iş akışı yönetimi, paralel dosya sistemi, düzensiz veri erişimi optimizasyonu ve hizmet kalitesi tabanlı optimizasyon gibi konuları bir araya getirmiştir.	Bu çalışma, bir çok tekniği bir araya getirmesi ve farklı algoritma erişim desenlerini desteklemesine rağmen, veri merkezlerinde veri elde etmek için kaynak optimizasyonunu ele almamıştır.
Lunga ve diğerleri (2020)	2020	Bulut, GPU	Bu çalışmada makine öğrenimi sürecini büyük miktarlarda RS görüntüsü ile daha hızlı hale getirmek için yeni bir uzaktan algılama veri akışı (RESFlow) uygulanmıştır. RESFlow, kapsamlı uzaktan algılama görüntülerinde derin öğrenme çıkarımını hızlandırmak için Apache Spark ve GPU'yu kullanır.	Önerilen sistemde büyük verinin elde edilmesi ve yönetilmesi konularında bir bilgi sunulmamıştır.

Çizelge 2.1. (Devam) Uzaktan algılama alanındaki yerleşik ve dağıtık yaklaşımlar

Kaynak	Yayın Tarihi	Kullanılan Teknolojiler	Açıklama	Eksiklikler
Xu ve diğerleri (2021)	2021	Bulut	Bu çalışmada uzaktan algılama ve büyük veri teknolojisi arasındaki boşluğu kapatmak için büyük veri teknolojilerini birleştiren bilimsel bir çerçeve önerilmiştir.	Spark, Hadoop, HBase, Zookeeper, PostgreSQL, Thrift ve Python teknolojilerini bir araya getiren çerçeve umut verici bir çözüm olsa da bazı basit algoritmalar ile çalışan bir prototiptir.



3. UZAKTAN ALGILAMA BÜYÜK VERİSİ VE MİMARİ YAKLAŞIMLAR

Büyük veri, birçok çalışma tarafından 3 V ile nitelendirilmektedir: hacim (volume), hız (velocity) ve çeşitlilik (variety) (Yang ve diğerleri, 2014; Douglas, 2014). Hacim, veri kümesinin büyüklüğünü ifade eden en önemli büyük veri niteliğidir. Hız, büyük verinin üretilme oranını belirtmekle birlikte üretilme oranındaki artışlar verinin daha hızlı işlenmesi ihtiyacını da ortaya çıkarmaktadır. Çeşitlilik, verinin toplandığı farklı kaynakların çeşitliliğini belirtmektedir. Yapılan bazı çalışmalarda bu niteliklere ek olarak doğruluk (veracity) ve değer (value) nitelikleri de eklenmiştir. Doğruluk, verinin doğru kaynaktan güvenli bir şekilde sağlanması ve yetkili kişiler tarafından erişilmesi olarak tanımlanmaktadır. Değer ise büyük veri kullanılarak elde edilen bilgiyi ifade etmektedir.

Verinin çeşitliliğine bakıldığında elde edilen verilerin çoğunluğu yapılandırılmamış (unstructured) veya yarı yapılandırılmış (structured) olmaktadır (Gao ve diğerleri, 2017). Veri hızı düşünüldüğünde uygulama alanlarına göre değişen hız gereksinimleri görülmektedir. Genel olarak hız, belirli bir zaman aralığında işleme gereksinimi yığın işleme (batch processing), dar bir zaman aralığı ihtiyacı yakın gerçek zaman, devamlı girdi-çıkı ihtiyacı gerçek zaman ve veri akısı ihtiyacı akı işleme (stream processing) başlıkları altında ele alınmaktadır. Bazı uygulamalar, verinin gelişinin ve analiz işlemlerinin yapılmasını yığın halinde isterken kritik ve canlı analize dayalı uygulamalar devamlı ve gerçek zamanlı analizlere ihtiyaç duymakla birlikte özellikle kritik uygulamalar gelen veri akışlarının analiz işlemlerine bağlı olarak anında müdahale gerektirmektedir. Akıllarla gelen verinin analiz edilmesine ve geçmiş bilgilerle karşılaştırılmasına ihtiyaç vardır. Farklı veri kaynaklarının kendine özgü veri formatlarına sahip olabilmesi, analiz yapabilmek amacıyla çoklu kaynaklardan alınan verilerin entegrasyonunu zorlaştırmaktadır. Bundan dolayı bazı konsorsiyumlar tarafından tanımlanan standart formatlar ve arayüzler kullanmak hayati önem taşımaktadır.

3.1. Uzaktan Algılama Verileri

Uzaktan algılama, uzak bir mesafeden bir yüzey veya nesnenin niteliğini ölçme teknolojisi olarak tanımlanmaktadır (Ma ve diğerleri, 2015). RS verileri uydulardan, uçak ve insansız hava araçlarındaki çeşitli veri alma teknolojilerinden (lidar, hiperspektral kamera vb.) elde

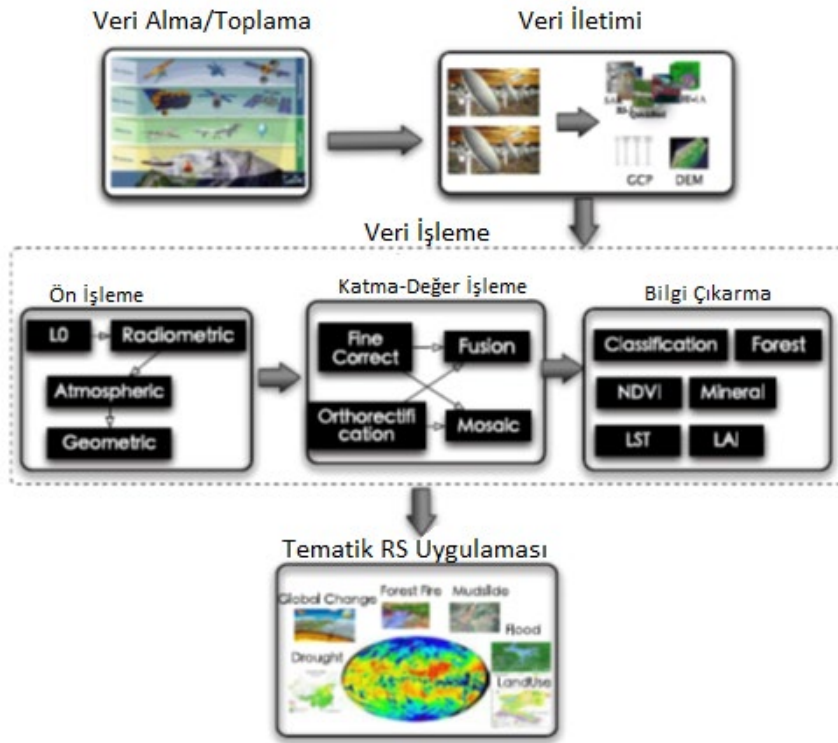
edilmektedir. Uzaktan algılayıcı teknolojilerinde yaşanan son gelişmelerle birlikte uzaktan algılayıcı verilerinin miktarı, üretilme oranı ve çeşitliliği üstel olarak artmıştır (bkz. Çizelge 3.1). Böylece uzaktan algılayıcılardan alınan veriler büyük veri olarak ele alınmaya başlamıştır.

Çizelge 3.1. Uydu veri merkezleri veri hız ve hacimleri

Uydular	Hız (Mbps)	Günlük Hacim (GB/Day)	Yıllık Hacim (TB/Year)
HJ-1B	60	57	20
HJ-1A	120	114	41
ZY-03	900	498	176
HJ-1C	320	188	67
ZY-02C	320	176	63
SPOT-4	50	11	4
LANDSAT5	85	28	10
RADASAT-2	105	58	21
RADASAT-1	105	58	21
SPOT-5	100	55	20
ENVISAT	100	33	12
IRS-P6	210	46	16
LANDSAT8	440	242	86
TOPLAM	2915	1563	556

Çeşitlilik ve çok boyutluluk, RS büyük verilerinin karmaşıklığındaki en büyük etmenlerdir. RS verileri çevresel izleme, yer örtüsü, atmosferik, hidroloji ve oşinografi içerikli bir çok yer biliminde kullanılmaktadır. Bu kadar farklı ve geniş aralıktaki uygulama alanlarından dolayı RS verilerinin çeşitliliği çok fazla artmıştır. Bilindiği kadarıyla NASA arşivlerinde yaklaşık 7000 farklı tipte RS veri kümeleri bulunmaktadır (Ma ve diğerleri, 2015).

Daha yüksek uzamsal çözünürlük, zamansal çözünürlük ve hatta spektral çözünürlük neticesinde farklı çözünürlüklere sahip çok sayıda uydu ve algılayıcı ortaya çıkmıştır (Ma ve diğerleri, 2015). HDF, netCDF, GeoTIFF ve FAST gibi çok sayıda dosya yapısının yanında farklı veri organizasyonları, üstveriler ve bu veriler için farklı fiziksel yapılar mevcuttur. Uzaktan algılama verilerinin artmaya ve karmaşılaşmaya devam etmesiyle var olan algoritmaların ve veri yönetiminin yeni bir mimariye geçirilmesi bir zorunluluk haline gelmiştir (Karantzas, Bliziotis ve Karmas, 2015).



Şekil 3.1. Uzaktan algılama veri işleme adımları

3.2. Uzaktan Algılama Veri İşleme Adımları

RS verilerinin işlenmesi yeryüzü algılama (earth observation) sistemlerinde önemli bir role sahiptir. En uzun RS iş akışı, veri toplamayla başlar ve tematik uygulama (bilgi çıkarımı) ile sonlanır. İş akışı içerisinde her bir adımda birbirini bekleyen ya da eş zamanlı işlenebilecek işlemler bulunabilmektedir. Bir RS uygulaması tipik olarak sırasıyla şu aşamalardan oluşmaktadır (Ma, Wang, Liu ve Ranjan, 2015):

- **Veri Alma/Toplama:** Uydu ya da hiperspektral kameralardan elde edilen görüntüler uydu indirme yolu (downlink) üzerinden alınır. Yüksek uzamsal ve zamansal çözünürlükle birlikte veri hacmi artmaktadır.
- **Veri İletimi:** Elde edilen veri akışı eş zamanlı olarak uydu merkezine gönderilir. Verinin taşınmasındaki bant genişliğinin yüksek olması gerçek zamanlı işleme gerektiren uygulamalar için kritiktir. Gerçekte bakıldığında uydu ve kamera teknolojilerine bağlı olarak gerçekleşen veri artışının veri iletim oranına göre daha hızlı olması gerçek zamanlı veri iletimini imkansız kılmaktadır.
- **Önişleme:** Ayrıştırma, radyometrik doğrulama, geometrik doğrulama gibi görüntü kalitesi ve coğrafi doğruluğu ile alakalı ön işlemler yapılmaktadır.

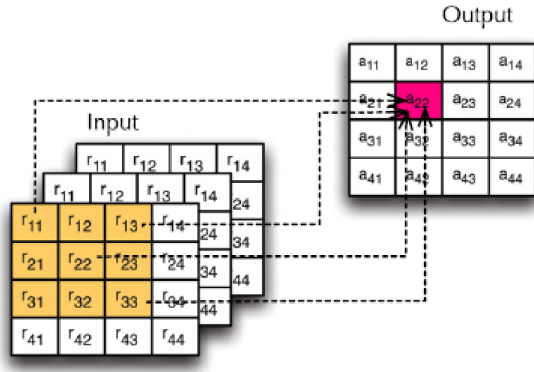
- Katma-değer İşlemler: ince düzeltme, ortorektifikasyon, füzyon ve mozaikleme gibi katma değer işlemler yapılmaktadır. Mozaikleme işleminin diğerlerine göre daha uzun bir işlem süresine sahip olduğunu söylemek mümkündür.
- Bilgi Çıkarımı/Tematik Uygulama: Bu adımda önışleme ve katma değer işlemlere tabii tutulan görüntülerle sınıflandırma, öznetelik çıkarımı ve nicel çıkarımlar elde edilir. Örneğin RS görüntüleri üzerinde yaprak alan indeksi, yeryüzü sıcaklığı, karlı alan gibi bilgilerin çıkarılması.

3.3. Uzaktan Algılama Algoritmalarındaki Veri Erişim Desenleri

Uzaktan Algılama algoritmalarındaki veri erişim deseni algoritmaların karakteristiğine göre değişmektedir. RS verileri için 4 farklı erişim deseni ön plana çıkmaktadır. RS verilerinin büyüklüğüne bağlı olarak kaçınılmaz I/O yükü ve düzensiz veri erişim desenleri, geleneksel küme tabanlı paralel I/O sistemlerini uygulanamaz kılmıştır (Wang, Ma, Zomaya, Ranjan ve Chen, 2014).

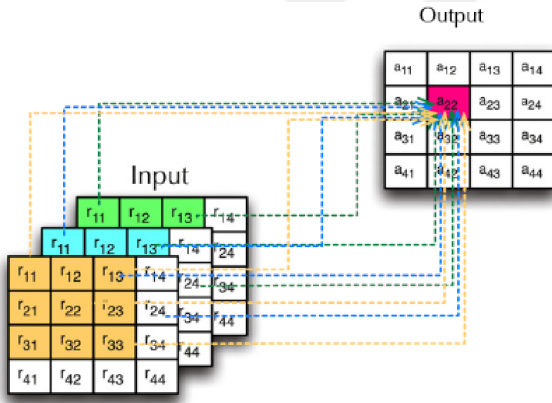
“Ardıl satır erişim deseni” piksel tabanlı işlemeye dayanan radyometrik doğrulama, SVM sınıflandırıcı gibi algoritmalar tarafından kullanılır. Algoritmalar paralel olarak gerçekleştirildiğinde her işlemci, mantıksal olarak çoklu ardıl görüntü satırlarına ihtiyaç duyar.

“Dikdörtgen blok erişim deseni” komşuya dayalı işleme gerektiren konvolüsyon filtre ve yeniden örnekleme (resample) gibi algoritmalar tarafından kullanılır. Bu algoritmalarda bitişik olmayan I/O desenleri görülmektedir ve bu yüzden normal paralel dosya sistemleri tarafından verimli bir şekilde desteklenmemektedir.



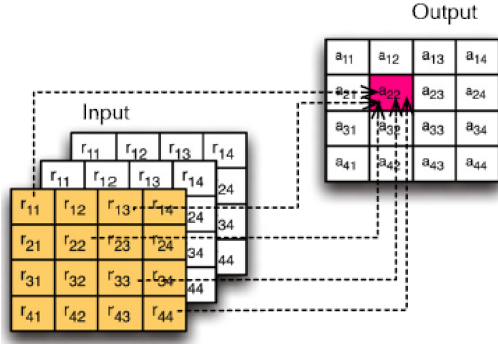
Şekil 3.2. Dikdörtgen blok erişim deseni

“Çapraz dosya erişimi deseni” bantlar arası hesaplamalara ihtiyaç duyan füzyon ve NDVI gibi algoritmalar tarafından kullanılır. Buradaki veriler, yüzlerce görüntü dosyasındaki küçük ve bitişik olmayan parçalardan oluşmaktadır. Böylece bu erişim tipinde çok sayıda okuma/yazma işlemi gerçekleşmekte ve bu da zaman alıcı bir işleme dönüşmektedir.



Şekil 3.3. Çapraz dosya erişimi deseni

“Düzensiz erişim deseni” dağınık şekilli erişime veya tüm görüntüye ihtiyaç duyan FFT, görüntü çarpıtma ve bilgi çıkarımı gibi algoritmalar tarafından kullanılır. Bu algoritmalar, düzensiz erişim olarak diyagonal ve poligonal erişim desenleri kullanırlar. Bu desenlerde veriler küçük ve bitişik olmayan veri parçalarında olmakla beraber farklı büyüklükteki parçalar farklı düğümlerde olabilmektedir. I/O güçlüklerine ek olarak düzensiz veri bölgesinin tanımlanması problemi de oluşmaktadır.



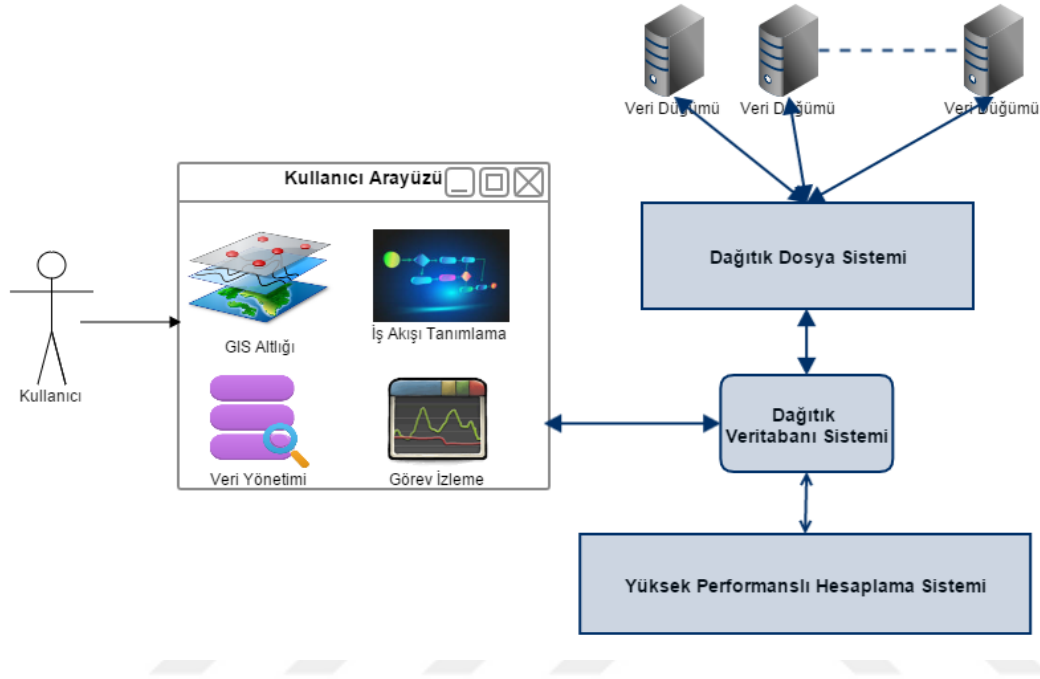
Şekil 3.4. Düzensiz erişim deseni

3.4. Gerçek Zamanlı Büyük Veri Mimarisi

Büyük veri mimarileri, gerek veri depolama gerekse veri analizi için dağıtık bir sistem yapısına ihtiyaç duymaktadır. Tehranian ve arkadaşları (2006), uydu verilerinin yer istasyonlarında dağıtık olarak gerçek zamanlı işlenmesi için aday donanım ve yazılımlardan oluşan, performans, güvenilirlik ve ölçeklenebilirlik sağlayan bir mimari model önermiştir. Ortaya çıkarılan prototip sistem açık kaynaklı uyarlanabilir iletişim ortamı çatısı vasıtasıyla C++ kullanılarak gerçekleştirilip küme üzerinde denenmiş. Yapılan çalışmada gerçek zamanlı sistemlerde ihtiyaç olan güvenilirlik ve yüksek kullanılabilirliği düşürmeden kayda değer bir performans elde edilmiştir. RS verilerinin dağıtık bir şekilde analiz edilebilmesi için paralel programlamaya yönelik yapı ve mekanizmalara ihtiyaç duyulmaktadır. Bu bağlamda Y. Ma ve arkadaşları (2012) tarafından yüksek performanslı kümeler üzerindeki RS uygulamaları için jenerik paralel programlama iskeleti önerilmiştir. Önerilen mekanizma, RS algoritmaları için hem dağıtık veriyi hem de jenerik paralel programlama iskeletini sağlayan programlama şablonlarına sahiptir. Çok boyutlu verinin depolanması, analizi ve görselleştirilmesi için önerilen magHD, kümeler üzerinde kullanılmak üzere Hadoop ve MapReduce teknolojilerini çeşitli indeksleme teknikleriyle bir araya getirmiştir (Angleraud, 2014). Rathore, Ahmad, Paul ve Wu (2016) tarafından yapılan çalışmadaki gibi tasarlanan bazı sistemler toplama, filtreleme, yük dengeleme, işleme, birleştirme ve yorumlama gibi farklı adımların tamamını içermektedir. İlgili çalışmada uydu görüntülerinden ırmak, kara ve demiryolu bulmayı hedefleyen devamlı öznitelik çıkarımı ve tespiti için gerçek zamanlı bir yaklaşım önerilerek Hadoop ekosistemi ve MapReduce kullanılarak gerçekleştirilmiştir.

Gerçek zamanlı analizleri gerçekleştirebilmek amacıyla oluşturulacak bir büyük veri mimarisinin sahip olması gereken bileşenler asgari olarak şunlardır: kullanıcı arayüzü,

dağıtık dosya sistemi, dağıtık veri tabanı, yüksek performanslı hesaplama (Şekil 3.5). Bu bileşenlerin altında ekstra performans ve gereksinim ihtiyacı dahilinde yeni alt bileşenlere de ihtiyaç duyulabilir.



Şekil 3.5. Gerçek zamanlı büyük veri mimarisi

Kullanıcı arayüzü, temel olarak kullanıcının sunucu tabanlı uygulamaya görsel bir arayüz vasıtasıyla ulaşabildiği, verileri sorgulama, yükleme, silme, güncelleme, analiz isteğinde bulunma, iş akışı tanımlama gibi yetenekleri kullanabildiği bileşendir.

Dağıtık dosya sistemi, yüklenmesi istenen verilerin çoklu bilgisayar kümesi üzerinde dağıtık olarak tutulmasını sağlayan sanal dosya sistemidir. Heterojen düğümlerden oluşabilecek kümelerde uygulamanın işletim ve dosya sistemlerinden izole bir şekilde veriye erişebilmesi için ortak arayüz sağlamaktadır.

Dağıtık veri tabanı, bir bilgisayar ağı ile bağlı çoklu bilgisayar kümesinde her bir düğümde/sunucuda bulunan birbirinden ayrı veri tabanlarından meydana gelir. Dağıtık veri tabanı yönetimi sistemi ise bahsi geçen sunucular üzerinde verilerin dağıtımını ve yönetilmesini sağlar.

Yüksek performanslı hesaplama sistemleri, büyük verinin paralel olarak analiz edilmesi için yeterince hızlı bir hesaplama ortamı sunan altyapıyı sağlayan teknolojilerdir. Sistemin kullanıcıya makul süreler içerisinde cevap dönebilmesi için hayati bir önem taşımaktadır.

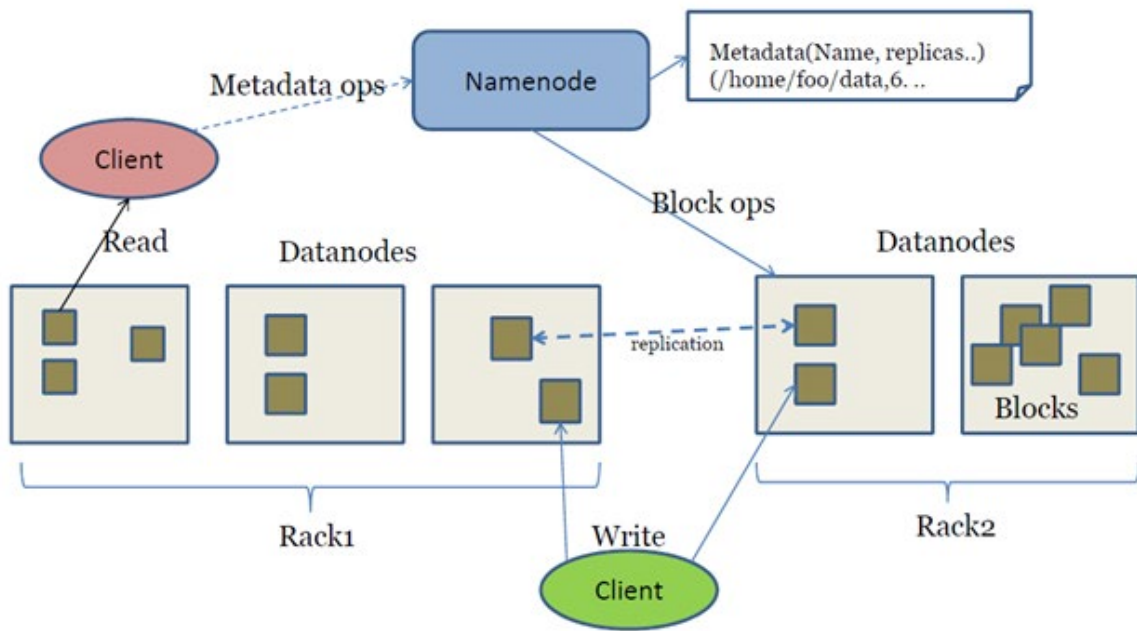
3.4.1. Dağıtık dosya sistemi

Veri depolama olarak HDD yerine kullanılmaya başlanan SSD ve PCM cihazları büyük veri için gereken I/O performansından uzak kalmaktadır. DAS, NAS ve SAN gibi kurumsal depolama mimarileri ise dağıtık sistemler olarak kullanıldığında sert engellere (drawback) ve limitlere sahip olmaktadır (Chen ve diğerleri, 2014). Bu noktada ortaya çıkan dağıtık dosya sistemleri (Distributed File System – DFS), bulut üzerinde çoklu düğümler üzerine yayılan sanal dosya sistemleridir (Lin ve diğerleri, 2013). Bu sistem, farklı merkezlerdeki veri düğümlerinin heterojenliğinin soyutlanmasını sağlamak üzere kurulmuştur. Böylece dağıtık dosya sistemi, uygulamaların farklı düğümlerde bireysel olarak farklı işletim sistemleri ve farklı dosya sistemleri kullanan heterojen düğümler üzerindeki verilere erişebilmesi için ortak arayüz sağlar.

Dağıtık dosya sisteminin genel olarak sağlaması gereken bazı yetenekler vardır. Konum şeffaflığı (location transparency), uygulamanın verinin gerçekte nerede tutulduğuna gerek duymadan lokalde tutuluyormuş gibi verilere erişebilmesidir. Erişim şeffaflığı (access transparency), işletim sistemi ve dosya sisteminden bağımsız verilere erişim için ortak arayüzün sağlanmasıdır. Hata toleransı (fault tolerance), birden fazla düğümden verinin kopyasının tutulması ve böylece hata durumunda verinin kopyayı tutan düğümlerde korunmasıdır. Ölçeklenebilirlik (scalability), dosya sisteminin çalıştığı düğüm sayısının ihtiyaç halinde istenilen miktarda (hatta sistem kesintiye uğratılmadan) artırılabilmesidir.

Hadoop Dağıtık Dosya Sistemi (Hadoop Distributed File System – HDFS), Apache lisansı ile dağıtılan açık kaynak bir dağıtık dosya sistemidir. HDFS, bahsi geçen genel yeteneklerin dışında özellikle büyük veri kümesi ve yüksek kullanılabilirlik için tasarlanmıştır. Ayrıca Java üzerinde gerçekleştirildiği için platformdan bağımsızdır. Uygulamalar, herhangi bir dosyalama sistemiyle tutulan dosyalara HDFS API'si vasıtasıyla erişir. Böylece dosyalara erişim, lokal dosya sistemlerinden izole edilmiş olunur. HDFS, ana/bağımlı bir mimariye sahiptir (bkz. Şekil 3.6). Bir HDFS kümesi, dosya sistemi ad alanını yöneten ve istemciler tarafından dosyalara erişimi düzenleyen bir ana sunucu olan tek bir NameNode'dan oluşur.

Ayrıca, üzerinde çalıştıkları düğümlere bağlı depolamayı yöneten, genellikle kümedeki düğüm başına bir tane olmak üzere bir dizi DataNode vardır. Diğer dağıtık dosya sistemleriyle (IRODS, Lustre) karşılaştırıldığında performanstan ziyade tasarım olarak farklı olduğu ve HDFS'in otomatik yük dengesine sahip tek DFS olduğu belirtilmektedir (Lin ve diğerleri, 2013). Aynı zamanda platformdan bağımsız olması ve MapReduce desteğinin bulunması, birçok sistemde kullanım kolaylığı sağladığından tercih nedeni olarak öne çıkmaktadır.



Şekil 3.6. Hadoop dağıtık dosya sistemi (HDFS)

Google File System (GFS), Google tarafından kendi kullanımı için geliştirilen özel bir dağıtık dosya sistemidir (Schmuck ve Haskin, 2002). Geliştirilme nedeni, büyük veri-yoğun uygulamalarda ortaya çıkan ölçeklenebilir dağıtık dosya sistemi ihtiyacıdır. Her biri binlerce sürücü ve makineden oluşan çok sayıda kümeden meydana gelen ve eş zamanlı olarak çok sayıda kullanıcının eriştiği sistemde verilerin güvenilir, verimli ve hataya toleranslı bir şekilde kullanılmasını sağlayacak biçimde tasarlanmıştır.

Bulut sistemlerinde kullanılmakta olan Amazon Simple Storage (S3), Nirvanix Cloud Storage, OpenStack Swift ve Windows Azure Blob gibi depolama sistemleri bulut uygulamalarının ölçeklenebilirlik ve replikasyon ihtiyaçlarını karşılama da analiz

uygulamalarının eşzamanlılık ve performans gereksinimlerini tam anlamıyla sağlayamamaktadır (Gao ve diğerleri, 2017).

General Parallel File System (GPFS), IBM tarafından geliştirilen yüksek performanslı bir kümelendirilmiş dosya sistemidir. GPFS, paylaşılan sürücü veya paylaşımsız dağıtık paralel düğümler üzerine kurulabilmektedir. POSIX tabanlı dosya sistemini tamamen desteklediği için diğer depolama sistemlerinin getirdiği yeni API kümesinin öğrenilmesi ihtiyacını ortadan kaldırmaktadır. Diğer taraftan HDFS ve GFS, tamamen POSIX uyumlu değildir ve bulut üzerinde analiz çözümleri sunmak için yeni API tanımlarına ihtiyaç duyarlar. Schmuck ve Haskin (2002) tarafından yapılan çalışmada meta blok kavramıyla GPFS'nin HDFS'e dosya okuma performansı bakımından yetiştirdiği belirtilmektedir. Meta blok, aynı disk üzerinde yer alan ardışık veri bloğu kümeleridir. Önerilen yaklaşımda farklı blok büyüklükleri arasında bir ödünleşme oluşturulmaya çalışılmıştır.

Parallel Virtual File System (PVFS)'in HDFS ile karşılaştırıldığı çalışmada edinilen bilgiye göre PVFS tamamlanma zamanı ve yapılan iş (throughput) bakımından kayda değer bir gelişme gösterememiştir (Tantisiri ve diğerleri, 2011).

Rasdaman veri tabanı, geleneksel veri tabanlarının tutamadığı çok boyutlu büyük dizileri destekleyen ve büyük uzaktan algılama verilerini doğası gereği tutabilen bir veri tabanıdır (Karantzas ve diğerleri, 2015). Rasdaman'ın mimarisi "tiling" denilen dizi parçalama işlemini esas almaktadır. Rasdaman paralel sunucu mimarisi özelliği sayesinde, fazla sayıda eş zamanlı kullanıcı isteğinin verimli olarak işlenmesi için ölçeklenebilir ve dağıtık bir ortam sunar. Böylece web üzerinden dağıtık veri kümelerini sunabilmektedir. Rasdaman'dan veri kümesini almak ve işlemek için OGC'nin WCPS standartları tarafından tanımlanan sorgulama dilinde yazılan veri alma (data retrieval) sorguları çalıştırılmalıdır. Bu noktada kullanılan Java Servlet olarak geliştirilen PetaScope bileşeni, OGC standart arayüzlerini gerçekleyerek çok boyutlu verinin aranması, alınması, filtrelenmesi ve işlenmesi için sorguları sağlar. Ayrıca coğrafi ve zamansal koordinat sistemi desteğini ekler.

Uzaktan algılama uygulamalarındaki RS verilerinin büyüklüğüne bağlı olarak kaçınılmaz bir hale gelen I/O yükü ve düzensiz veri erişim desenleri, geleneksel küme tabanlı paralel I/O sistemlerini uygulanamaz kılmıştır. L. Wang ve diğerleri (2014) tarafından yapılan çalışmada uzaktan algılama uygulamaları için nesne tabanlı paralel dosya sistemi önerilmiş

ve OrangeFS dosya sistemi kullanılarak gerçekleştirilmiştir. Uygulamaya özel veri yerleşimi politikası sağlayarak farklı veri erişimi desenleri için verim sağlanmıştır. Önerilen sistemin performansındaki iyileşme ortalama %20 olarak görülmektedir.

3.4.2. Dağıtık veri tabanı

Yapılandırılmış verilerin yönetilmesinde kullanılan klasik yaklaşımlar, veri depolama için bir şema ve veri alma içinse ilişkisel veri tabanına sahiptir. Mevcut veri tabanı yönetimi araçları, hızla büyüyen ve karmaşıklaşan büyük verinin işlenmesi konusunda yetersiz kalmıştır. Fazla miktarda yapılandırılmış veri bulunduran sistemlerde ise veri ambarı ve veri pazarı yaklaşımları popülerlik kazanmıştır (Chen ve Zhang, 2014). Bu yaklaşımlardan biri olan veri ambarı, eskiyen yapılandırılmış verilerin depolanması, analizi ve kullanıcıya sonuçların raporlanması için kullanılır. Veri pazarı (mart) yaklaşımı ise veri ambarını temel olarak veri erişim ve analizini iyileştiren bir yaklaşımdır. Büyük kurumların rağbet gösterdiği kurumsal veri ambarı (Enterprise Data Warehouse - EDW), veri işleme ve analiz yeteneğinin çok büyük ve birleşmiş kurumsal veri tabanı üzerinde kullanılabilmesine olanak tanımaktadır (Chen ve Zhang, 2014). Bazı bulut sağlayıcılar EDW ile bir petabyte veri ve daha fazlasına ölçeklendirme çözümü sunabilmektedir. Örneğin Amazon Redshift, sütun yönelimli depolama ve veri sıkıştırma sunmakla birlikte yüksek performanslı sorgulama amacıyla fazla sayıda işlemciden oluşan masif paralel işleme (massively parallel processing - MPP) mimarisini kullanmaktadır. Ayrıca yerel ekli depolama ve bölge haritalama (zone maps) kullanılarak sorgular tarafından ihtiyaç duyulan I/O miktarı indirgenmiştir.

Bulut bilişimde bilgi depolama ve alma için tercih edilen NoSQL veri tabanlarının kullanımı artarak günümüzde belirgin bir eğilim haline gelmiştir (Chen ve Zhang, 2014). NoSQL veri depolama için ilişkisel olmayan bir model kullanır. Anahtar-depolama (key-store), sütun yönelimli ve doküman tabanlı depolama gibi modellerle daha fazla dikkat çeken NoSQL kavramına olan ilginin artma nedenleri, daha iyi performans, yapılandırılmamış veri tutma yeteneği ve dağıtık ortamlar için uygun olmasıdır.

Yapılandırılmamış veya ilişkisel olmayan verilerin depolanması ve yönetilmesi için NoSQL yaklaşımı veri depolama ve yönetimi olmak üzere iki bağımsız parçaya bölünür (Chen ve Zhang, 2014). Depolama kısmında anahtar-değer depolama modeliyle NoSQL'in odak noktası, veri depolamanın ölçeklenebilir ve yüksek performanslı yapılmasıdır. Yönetim

kısımında ise sağladığı alt seviye erişim mekanizması sayesinde veri yönetimi görevleri uygulama katmanında gerçekleştirilebilir. NoSQL veri tabanının en önemli özellikleri, şema serbestliği sağlayarak veri yapısı değişikliğinin hızlı yapılabilmesi ve verilerin tekrar yazılmaya ihtiyaç duymaması, esneklik sağlayarak yapılandırılmış verilerin heterojen olarak depolanabilmesidir. En popüler NoSQL veri tabanı, ilk kez Facebook tarafından kullanılan ve 2008’de açık kaynak olarak yayınlanan Cassandra veri tabanıdır (Chen ve Zhang, 2014). SimpleDB, Google BigTable, MongoDB, Voldemort gibi NoSQL gerçekleştirmeleri de mevcuttur. Twitter, LinkedIn, NetFlix gibi sosyal ağ uygulamaları/firmaları da NoSQL yeteneklerinden faydalanmıştır.

Geleneksel yöntemlerin yenik düştüğü uzaktan algılama verilerinin yönetimi problemi için L. Wang ve arkadaşları tarafından önerilen metoda göre görüntü verileri GeoSOT küresel ayrık grid sistemini temel alan bloklara ayrılır ve veri blokları HBase’de depolanır (Wang, Cheng, S. Wu, F. Wu ve Teng, 2015). Bu metotta veriler ilk olarak MetaDataInfo tablosuna kayıt edilir. Satır kimliği olarak uydu-sensör-alma zamanı kullanılmaktadır. DataGridBlock tablosunda ise satır kimliği, coğrafi koordinat ile birlikte MetaDataInfo satır kimliğinden oluşur. HBase tabloları, satır sayılarının artan sırada sıralanması ile coğrafi olarak yakın olan blokların tablodaki komşu satırlarda tutulacağını garanti eder. Uzamsal bir sorgulama geldiğinde öncelikle GeoSOT kodları hesaplanır ve DataGridBlock tablosu bu kodlara göre filtrelenerek taranır. Ayrıca görüntü veri blokları ile başa çıkmak adına MapReduce modelini kullanan bir dağıtık işleme metodu da tasarlanmıştır. MapReduce işi başlatırken tabloyu bölgelerin sınırları olarak böler ve her bölge görüntü veri bloğu kümesini içerir. Daha sonra map fonksiyonu bölgeden her bir veri bloğunu işler ve üretilen sonuçlar parçalama fonksiyonuna gönderilir. Parçalama fonksiyonu parçalama kuralına göre sonuçları belirli reducer’lara tayin eder ve sıralama fonksiyonu bunları sıralar. Parçalama kuralı GeoSOT kodu aynı olanlar aynı reducer’da olacak şekilde tasarlanmıştır. Reducer’lar işlemleri tamamladıklarında iş sonucu üretilmiş olur.

Geleneksel veri tabanlarının büyük dijital verilerin depolanması ve analizi için verimsiz olması nedeniyle ultra büyük veri tabanlarının analizi birçok araştırmacının ilgisini çekmiştir. Apache HBase, HDFS üzerine geliştirilen noSQL dağıtık veri tabanı da bu araştırmaların sonuçlarından biridir. Vora (2011) tarafından yapılan çalışmada HDFS’in metin olmayan görüntü gibi verileri tuttuğu ve HBase’in bu verilerin yerlerini tuttuğu hibrit bir yaklaşım değerlendirilmiştir. Bu hibrit mimari, daha hızlı arama ve veri alma işlemini

mümkün kılmıştır. Yapılan deneylerde HBase'e rastgele veri depolama, yer bilgisini yazma-okuma ve HDFS'e veri depolama ile HDFS'ten veri alma performansları değerlendirilmiştir. Ayrıca HBase-HDFS ve MySQL-HDFS mimarileri arasında karşılaştırma çalışması yapılmıştır.

3.4.3. Kullanıcı arayüzü, görselleştirme ve izleme

Artan veri miktarları ve analiz ihtiyaçlarının karşılanmasıyla birlikte iyi bir görselleştirme aracının kullanılması hayati önem taşır (Assunção, Calheiros, Bianchi, Netto ve Buyya, 2015). Bu araçların veri kalitesi ve navigasyon imkanlarını dikkate alması gerekmektedir. Görselleştirme araçları doğası gereği herhangi bir alana özel olarak tanımlı olmak durumundadır.

Bulut üzerinde kullanıcı etkileşimi ve görselleştirme düşünüldüğünde önemli nokta, ağına hala bir darboğaz oluşturmamasıdır (Assunção, 2015). Büyük veri analizi için istek gönderen kullanıcı, özellikle gerçek zamanlı kritik sistemlerde hızlı bir şekilde cevap almayı beklemektedir. Bu yüzden gerçek zamanlı görselleştirmeyi mümkün kılmak için daha verimli veri işleme tekniklerine ihtiyaç duyulmaktadır.

Özellikle uzamsal bilgilerin kullanıldığı sistemlerin kullanıcı arayüzlerinde GIS altlığının kullanımı, görsel olarak sonuçların daha anlaşılır bir şekilde kullanıcıya sunulmasını sağlar. Song Gao ve diğerleri (2017), sosyal ağ üzerinden coğrafya sözlüğü bilgilerini toplamak amacıyla Hadoop tabanlı dağıtık bir platform geliştirmiştir. Geliştirilen sistemde Flickr'dan alınan coğrafi etiketler kullanılmış ve uzamsal etkin Hadoop kümesinde çalıştırılan MapReduce tabanlı model vasıtasıyla coğrafya sözlüğü oluşturulmuştur. Sistemde kullanılan GIS altlığı, Hadoop kümesinden alınan MapReduce işlemlerinin sonuçlarını coğrafi olarak görselleştirme desteği sağlar. HDFS doğrudan standart GIS veri formatlarını (Esri shapefiles gibi) sağlamadığından uzamsal verilerin depolanması için basit geometrik şekiller (nokta, poligon, vb.) GeoJSON ile kodlanmaktadır. Aynı çalışmada sağlanan Cloudera Manager Web User Interface (CMWebUI) isimli kullanıcı arayüzü, Hadoop ekosistemi için hazırlanmış endüstri standartlarındaki yönetim paketini içeren arayüzdür. Bu kullanıcı arayüzü vasıtasıyla küme üzerindeki düğümler gerçek zamanlı görüntülenir, çalışan servisler izlenir ve konfigürasyon değişikliklerine olanak tanınır.

Bazı analiz algoritmaları birbirleriyle girdi-çıkı ilişkisi içerisinde çalıştırılmayı gerektirirler. Bunu sağlamak amacıyla iş akışı tanımlama arayüzü ile kullanıcıya esnek görevler oluşturma yeteneği kazandırılması gerekir. İşlemler dizisinden oluşan iş akışının çalıştırılması boyunca birkaç ara sonucun oluşturulmasına ihtiyaç olabilir (Giachetta, 2015). Oluşan ara sonuçların dağıtık sistem üzerinde koşturmakta olan işlemlere taşınması gerekebilir. Bu noktada ara sonuçların dosya sistemine atılması ve üstverilerin işlemlere dağıtılarak ihtiyaç duydukları ara sonuçlara ulaşmaları sağlanabilir. Kullanılması gereken altyapının da birden fazla işlemi tek bir işlem gibi çalıştırıp sorgulanmasına imkan vermesi gerekir.

Hadoop ekosisteminde çoklu MapReduce işlemler mantıksal olarak tek bir işlem gibi çalıştırılmaktadır (Giachetta, 2015). Verilerin işlemler arası koordinasyonu da Hadoop tarafından sağlanır. Apache tarafından yayınlanan ve Hadoop ekosistemine entegre olarak çalışan Oozie iş akışı düzenleyici (scheduler), Hadoop görevlerinin yönetilmesini sağlayan bir sistemdir. Oozie iş akışı görevleri, yönlendirilmiş çevrimsiz çizge (Directed Acyclical Graphs - DAG) olarak ifade edilebilen Hadoop üzerinde sıralanan aksiyonlar (MapReduce görevleri, Pig görevleri gibi) topluluğudur (İslam ve diğerleri, 2012).

3.4.4. Yüksek performanslı hesaplama

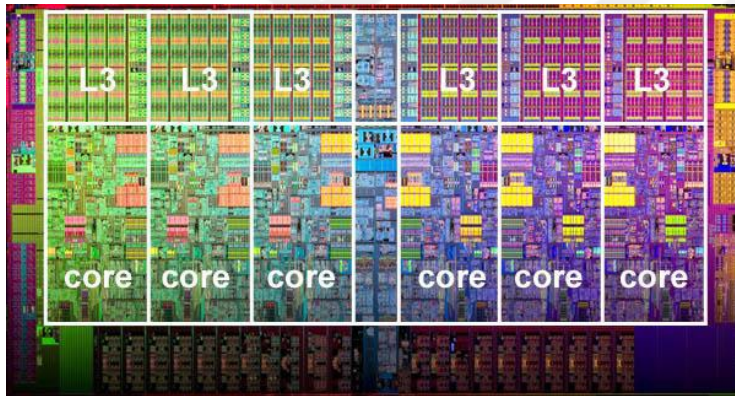
Veri analizi ele alındığında verinin hacmine bağlı olarak en önemli güçlük ölçeklenebilirliktir. Son yıllarda araştırmacılar daha çok analiz algoritmalarını hızlandırmaya odaklanmışlardır. Oysaki veri miktarı CPU hızına göre çok daha hızlı artmaktadır. Bu durum işlemcilerin çok çekirdekli olarak paralel hesaplamayı destekleyecek konuma gelmesine neden olmuştur. Birçok alandaki büyük veri kullanan gerçek zamanlı uygulamalar için vakitlilik (timeliness) ilk sırada gelmektedir. Bundan dolayı ortaya çıkan birçok güçlük, sadece donanım gelişimini değil aynı zamanda yazılım mimarilerinin de geliştirme yönünü değiştirmektedir. Bu noktadaki en önemli eğilim bulut bilişim teknolojisinin kullanılarak dağıtık hesaplama geliştirmelerinin yapılmasıdır.

Uzaktan algılama uygulamalarında kullanılan teknolojiler verinin indirilmesi, işlenmesi ve zamanında cevaplama konularında güçlükler ortaya çıkarmaktadır. Tumwizere ve Karpjoo (2012) tarafından yapılan araştırmada uzaktan algılama verileri üzerinde uygulanan web teknolojileri, grid hesaplama, veri madenciliği ve paralel hesaplama konuları taranmıştır. Veri hacminin büyüklüğü, veri formatları ve indirme zamanı genel güçlüklerdir. Bahsi geçen

teknolojilerin kombinasyonu ile bazı uygulamalardaki işleme zamanı, özellikle yardım görevlilerinin zamanında karar verebileceği kadar kısa sürelerle indirgenebilir. Uzaktan algılama verilerinin en kısa zamanda işlenmesi makul görülmekle birlikte otomatik olarak gerçek zamanlı işleme mümkün görünmemektedir.

Bütünleşik (onboard) yaklaşımlar

Multi-core/GPU programlama, performansın artırılması amacıyla tek bir bilgisayar üzerinde bulunan çok çekirdekli işlemci veya grafik kartı kullanılarak algoritmaların gerçekleştirilmesidir (bkz. Şekil 3.7, Şekil 3.8). Bu yaklaşımda işleme gücü dikey ölçeklendirilebilir, yani bilgisayar üzerinde daha çok işlemci, daha gelişmiş grafik kartı ve daha fazla bellek daha iyi performans anlamına gelmektedir. Buna karşın bulut yaklaşımında olduğu gibi çoklu bilgisayarlar üzerinde yatay ölçeklenebilme yeteneği yoktur.

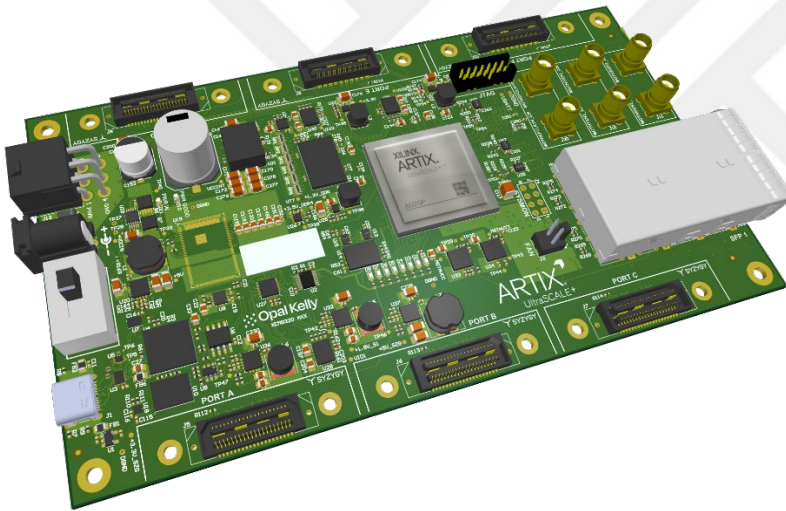


Şekil 3.7. Çok çekirdekli işlemciler (LLNL, 2021)



Şekil 3.8. Grafik işlemci birimi (NVIDIA, 2023)

Bazı yerleşik uzaktan algılama veri işleme senaryolarında, özellikle hava araçlarının kullanıldığı sistemlerde düşük ağırlıklı ve düşük güçle çalışabilen bileşenlere ihtiyaç olabilmektedir. Bu bileşenler faydalı yük miktarını düşürürken algılayıcıdan veri elde edilirken aynı zamanda gerçek/yakın gerçek zamanlı analiz sonuçları üretebilir. Bu amaca uygun olarak FPGA gibi programlanabilir donanım cihazları kullanılabilir (bkz. Şekil 3.9). Fakat FPGA'ler için programlama ve yeni bir API kümesini öğrenme ihtiyacı ortaya çıkmaktadır. Diğer bir yaklaşım ise bazı grafik kartlarında bulunan GPU'lar olarak ön plana çıkmaktadır. En popüler GPU programlama mimarisi, NVIDIA firması tarafından önerilen CUDA paralel hesaplama mimarisidir (Liu ve diğerleri, 2011). CUDA yazılım ortamı, geliştiricilerin üst seviye programlama dili olarak C ile program geliştirebilmelerine olanak sağlamaktadır. Ayrıca FORTRAN, C++, OpenCL ve DirectX hesaplama gibi diğer dil ve uygulama programlama arayüzlerini de desteklemektedir.



Şekil 3.9. Programlanabilir mantık blokları (XILINX, 2023)

Bahsi geçen FPGA ve GPU çözümleri çoklu işlemcili sistemlerden daha iyi sonuçlar üretse de bu iyileşme, algılayıcı teknolojileriyle büyüyen veri miktarının üstel artması neticesinde yetersiz kalmaktadır. Bunun yanında programlanabilir donanım cihazlarının algılayıcıdan veri alınırken yapılabilecek bazı ön işlemleri (atmosferik doğrulama, coğrafi koordinatlandırma gibi) verileri daha alırken yapması, analizler öncesi gerekli işlemlerin bir kısmının yapılmış olmasıyla performansa katkı sağlayabilir. Diğer yandan herhangi bir ön işleme veya analiz algoritmasının FPGA ya da GPU üzerinde çalıştırılması için bu donanımlara özel olarak tekrar yazılmaya ihtiyacı vardır.

Küme yaklaşımı

Donanım çözümlü geliştirmeler düşünüldüğünde en çok kullanılan yaklaşımlardan birisi de hazır ticari bilgisayar ekipmanı tabanlı çözümlerdir (bkz. Şekil 3.10). Bu yaklaşımda takım gibi birlikte çalışmak üzere bir miktar bilgisayardan bir küme oluşturulur (Plaza ve diğerleri, 2011). Çok sayıda CPU ile kurulan bu paralel sistemler, hem uzaktan algılayıcı hem de veri akıları kullanan gerçek/yakın gerçek zamanlı uygulamalar için iyi sonuçlar verse de bu sistemler hem bakım maliyeti bakımından oldukça pahalı hem de ölçeklenebilirliği belirli bir kapasitenin üzerine çıkamamaktadır.

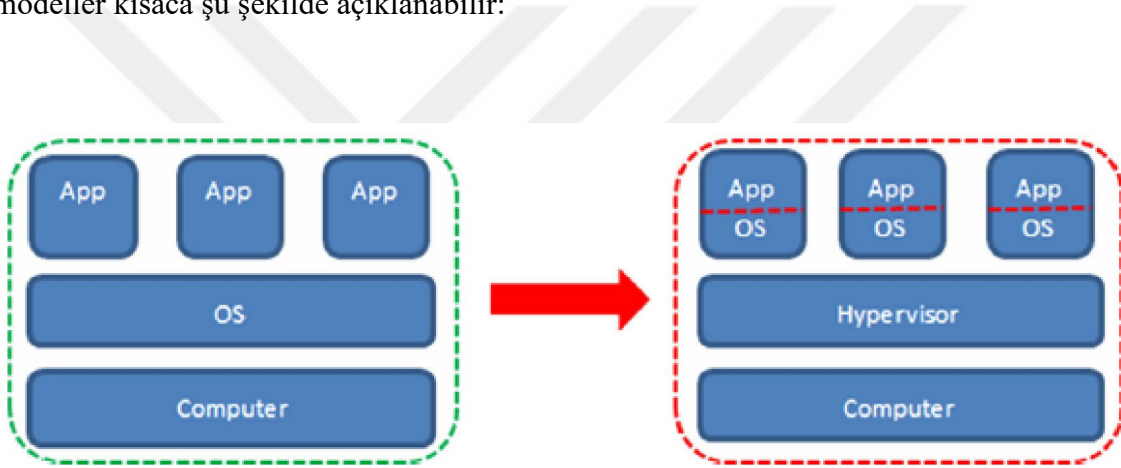


Şekil 3.10. Bilgisayar kümesi (LLNL, 2021)

Büyük hiperspektral görüntüler, bellekte tutulabilmesi ve yüksek performanslı hesaplama sistemlerinde paralel bir şekilde işlenebilmesi için düzgün bir şekilde parçalanmaya ihtiyaç duyarlar. Toma ve diğerleri (2012) tarafından yapılan çalışmada büyük görüntülerin dağıtılıp işlenmesi için birkaç farklı yaklaşım kullanılmıştır. Bu yaklaşımlar: (1) usta işlemci görüntüyü okur ve parçaları köle işlemcilere dağıtır; (2) her işlemci ilgilendiği görüntü parçasını okur; (3) Bir düğüm bütün görüntüyü alarak paylaşılan belleğe koyar ve diğer düğümler ilgili parçayı buradan okuyarak RDMA vasıtasıyla lokale alır. Paralel gerçekleştirim için ise Fuzzy c-Mean ve otomatik morfolojik son üye çıkarma algoritmaları seçilerek BlueGene/P süper bilgisayarı üzerindeki 1024 işlemci ile denenmiştir. Farklı sayıdaki işlemci ile farklı genişlik ve yükseklikteki parçalarla yapılan çalıştırmalarda elde edilen sonuca göre görüntünün parçalara ve işlemci sayısına göre iletişim süresinin değiştiği görülmüştür.

Bulut yaklaşımı

Bulut ortamı birçok kullanıcının bulunduğu, Bilişim Teknolojisi (BT) kaynaklarının geniş bir ağ üzerinden isteğe bağlı kullanılabilirdiği, bu kaynakların gerektiğinde kullanımının kısıtlanabildiği, çok hızlı, genellikle müdahale olmadan kendini otomatize etmiş ve tercihen sanallaştırılmış bir ortamdır (Ma ve diğerleri, 2015) (bkz. Şekil 3.11). Sanallaştırma teknolojisi yukarıda verilen yeteneklerin daha basit ve kullanışlı yoldan gerçekleştirilmesine olanak sağlar. Bulut teknolojisi çeşitli konuşlandırma (deployment) yöntemleri ile kullanıma açılabilir. Bunlar yöntemler arasında özel, umumi, belli bir topluluğa açık veya bunların bir kaçının beraber kullanıldığı melez modeller vardır. Bu modeller kısaca şu şekilde açıklanabilir:



Şekil 3.11. Geleneksel yöntem ve sanallaştırma

Özel (Private) Bulut: Bu modelde, bulut altyapısı tamamıyla belirli bir kurum için kullanılır, bulut altyapısı fiziksel olarak kurumun içinde veya dışında bulunabilir. Bulut ortamı kurum tarafından yönetilebildiği gibi, üçüncü bir sağlayıcıdan destek de alınabilir.

Umumi (Public) Bulut: Bu modelde bulut servisi genel kullanıma veya endüstrinin geniş bir grubuna açıktır. Bulut servisi sadece sağlayıcısı tarafından yönetilir.

Topluluk (Community) Bulutu: Bu modelde bulut birden çok kurum/organizasyon tarafından paylaşılır. Sadece ortak amaçları olan belli bir topluluk tarafından desteklenir. Yönetimi ise kurum tarafından gerçekleştirilebildiği gibi üçüncü bir sağlayıcıdan destek alınabilir. Fiziksel olarak kurum içinde veya dışında bulunabilir.

Melez (Hybrid) Bulut: bu model de yukarıda bahsedilen modellerden ikisini veya daha fazlasını bir arada kullanmak mümkündür. Bu model verinin veya uygulamaların bir modelden diğerine taşınmasına olanak sağlar.

Yukarıda bahsedildiği gibi farklı modeller olmasına rağmen bir bulut sisteminin ortak özelliklerini aşağıda ki gibi sayabiliriz:

- Ne zaman istenirse kullanılabilen servisler
- Geniş ağ erişimi
- Kaynak biriktirmesi
- Kolay ve hızlı olarak yeni bir kaynak eklenmesi ve genişlemesi
- Ölçülebilen servis kullanımı
- Kaynaklarının birden fazla kullanıcı tarafından ortak kullanımı

Bir bulut ortamında üç farklı servis modeli vardır. Bu servis modelleri:

Altyapı (Infrastructure) Hizmeti: Bu bulut servis modeli, altyapı kaynaklarını son kullanıcılara servis olarak sunar. Bu modelde kullanıcılar işlem, depolama kapasitesi, ağ ve diğer yazılımları koşturacak işletim sistemi ve diğer uygulama kaynaklarını önceden ayarlayabilir. Sağlayıcı bulut altyapısının kontrolü ve yönetiminden sorumludur. Kullanıcı kendisine sağlanan kaynakları (işletim sistemi, depolama ve yüklü uygulamalar) yönetebilir. Kullanıcı aynı zamanda ağ kaynakları üzerinde de belirli bazı kontrollere sahip olabilir.

Platform Hizmeti: Bu modelde kullanıcı bulut altyapısı tarafından sağlanan bir platformu kullanır. Sağlayıcı ağ, işletim sistemi, depolama gibi kaynakları yönetir. Son kullanıcının uygulamalar üzerinde kontrol hakkı vardır. Son kullanıcı aynı zamanda uygulamaların çalışma ortamını da kontrol edebilir.

Yazılım Hizmeti: Bu katman kullanıcılara yazılım hizmeti sunar. Örneğin; son kullanıcılara sağlanan çevrim içi hesaplama hizmeti gibi. Son kullanıcılar bu hizmetlere basit bir web tarayıcısı ile erişebilir. Son kullanıcıların ağ, işletim sistemi, depolama, sunucu gibi kaynakları yönetme hakkı yoktur. Uygulama seviyesinde yönetme hakkı da olmayabilir. Sadece uygulama konfigürasyonunu değiştirmede bazı haklar tanınabilir.

“Altyapı Hizmeti” ile vurgulanmak istenen, sağlayıcının yukarı katmanlardaki hizmetleri vermesine olanak sağlayacak hesaplama donanımları (işlemci), ağ donanımları (ağ kartı) ve depolama donanımlarıdır (hard disk). Bu kaynaklar kullanıcılara otomatik kaynak sağlama katmanı ile entegre olmuş bir istek yönetim sistemi tarafından sunulur. Bir bulut sisteminin aynı zamanda bir kaynak kullanım ölçümü yapacak ve kullanıma göre bir fatura çıkaracak ayrı bir sisteme ihtiyacı vardır.

Sanallaştırma, sağlayıcıya, birden çok kullanıcının bulunduğu bir ortamda kaynakların bir arada toplanmasına imkan sağlar. Genel olarak sanallaştırmanın getirdiği faydalar kaynak havuzu oluşturmaya, modern küme altyapısıyla birleştirmesi, kaynakların etkin kullanımı, çeviklik artırması, çoklu kullanıcı modelini desteklemesi, ölçeklendirme ve esneklik sağlamasıdır.

Bulut bilişim en kuvvetli büyük veri tekniklerinden biri olarak öne çıkmaktadır (Gao ve diğerleri, 2017). Sanallaştırma teknolojilerinin hesaplama altyapılarını fiziksel bir bilgisayar mimarisine sanallaştırarak esnek işleme, bellek ve sürücü sağlayabilmesi süper hesaplama kavramını daha erişilebilir ve uygun fiyatlara alınabilir hale getirmiştir (O’Driscoll ve diğerleri, 2013).

Büyük veri analizinin sağlayıcıdan elde edilen kaynaklar üzerinde çalıştırılmasıyla ekstra donanım almadan ihtiyaç olan hesaplama birimi kadar ödeme yapılarak tasarruflu bir şekilde gerekli işleme kapasitesi sağlanmış olmaktadır. Amazon Web Service, Amazon EC2, Google AppEngine ve Microsoft Azure önde gelen bulut sağlayıcılarıdır. Enrique Moguel ve arkadaşları, rüzgardan enerji elde eden Eolic Park üzerinde büyük miktardaki verinin yönetimini algılama tekniklerini de kullanarak sağlamaya ve Amazon’un bulut altyapısı üzerinde yüksek performanslı büyük veri analizi yapmaya yönelik çok katmanlı bir mimari önermiştir (Moguel, Preciado, Sánchez-Figueroa, Preciado ve Hernández, 2015). Bu çok katmanlı donanım/yazılım mimarisi, Eolic park büyük verisinin yönetimi için bulut bilişim tekniklerini kullanmaktadır. Bu mimari aynı zamanda büyük, dağıtık ve heterojen verilerin uzaktan algılama bağlamında işlenmesini de sağlamaktadır. Veri yönetimi bu çalışmada 3 farklı seviyede ele alınmaktadır: rüzgar tribünü, ara istasyonlar ve kontrol merkezi. Her katmanda farklı stratejiler kullanılarak bir araya getirilmesi önerilen sistemin bilimsel katkısı olarak öne çıkmaktadır. Rüzgar tribünü katmanı, algılayıcıları ve alarm/uyarıcıları yöneten kural tabanlı bir sistemdir. Eski sistemdeki PLC’lerin yerini Raspberry Pi bilgisayarlar

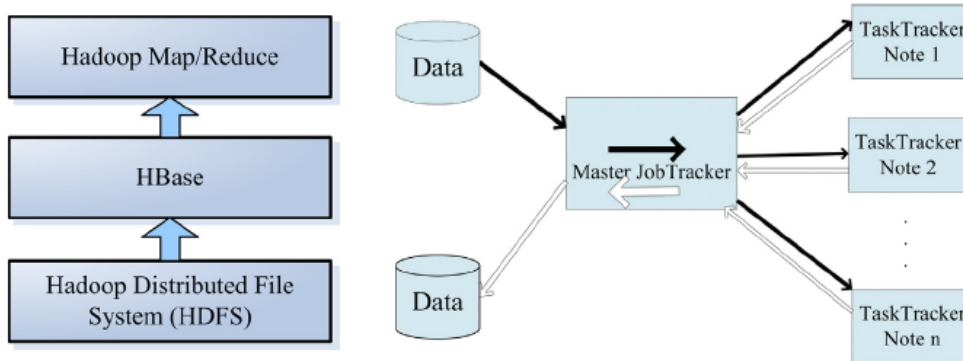
almıştır. Raspberry Pi, rüzgar tribünü katmanına işlem kabiliyeti, dahili MySQL veri tabanı ile veri kaydetme ve çevrimdışı çalışabilme, çift taraflı iletişim gibi kritik yetenekler katmıştır. Aynı zamanda Raspberry Pi PLC'den daha ucuz ve kurulumu daha basittir. Bu katmanda her rüzgar tribününden günlük 20 GB veri toplanmaktadır (çok büyük kısmı kameralardan alınan verilerdir). Önerilen sistem sayesinde veriler ön işlemden geçirilerek ara istasyona taşınan veri miktarı %20'ye düşürülmektedir. Ara istasyon katmanında tribünlerden alınan veriler depolamak ve görselleştirmek amacıyla işlenir. Kural tabanlı olarak otomatik uyarı/alarm çalıştırma ve verileri kontrol merkezine taşıma işlemlerini gerçekleştirir. Günlük gelen veri miktarı 600 GB olmakla birlikte merkeze gönderilen veri bu katmanda 105 GB'a kadar düşürülür. Kontrol merkezi katmanı ara istasyonlardan verileri alarak analitik, görselleştirme ve otomatik olarak kural tabanlı uyarı/alarm görevlerini gerçekleştirir. Günlük olarak 65 ara istasyondan toplam 6.69 TB veri alınmaktadır. Bu katman bulut ortamında yüksek performanslı veri işleme, görselleştirme ve etkileşim servisi, büyük veri ve hızlı sorgulama için 3 alt katmana ayrılır. En önemli alt katman olan yüksek performanslı veri işlemede büyük veri analitiği için AWS altyapısı üzerinde çalıştırılan WCPS çatısı kullanılmıştır. Sistemin prototipi 6 farklı ara istasyondaki toplam 756 tribün üzerinde test edilmiştir. Mevcut çözümle karşılaştırıldığında veri trafiğinde yaklaşık %65 azalma, veri işleme hızında ise %40 iyileşme elde edilmiştir.

Hadoop ekosistemi ve mapreduce

Hadoop Ekosistemi, bulut ve büyük veri analizi için en başarılı altyapılardan birisi olarak karşımıza çıkmaktadır. Birçok aracın çeşitli amaçlar için bir araya getirildiği platformda iki önemli servis bulunmaktadır: bir dağıtık dosya sistemi olan HDFS ve yüksek performanslı paralel veri işleme motoru MapReduce. Geliştirilmekte olan birçok yeni model arasında sıvrılan bir model olan MapReduce modeli, Apache Hadoop çatısının açık kaynak bir gerçekleştirmesidir. Aslında MapReduce modelinin gerçek amacı basit metin dokümanlarının işlenmesi olduğundan karmaşık algoritmaların gerçekleştirilmesi ve heterojen veri yapılarının yönetilmesi konusunda zorluklar ortaya çıkabilmektedir (Yang ve diğerleri, 2014). Bu zorlukların üstesinden gelebilmek için Hadoop platformu üzerinde gerçekleştirilen bazı uzantılar ve araçlar ortaya konularak geniş veri yönetimi ve analiz yeteneği sağlanmıştır. Büyük miktarda verinin bilgisayar kümeleri üzerinde dağıtık olarak işlenmesini sağlayan Hadoop platformu birkaç bulut sağlayıcısı tarafından da etkin olarak kullanılmaktadır (Gao ve diğerleri, 2017). Örneğin Amazon EMR, Amazon Elastic Compute

Cloud (EC2) ve diğer Amazon Web Services araçlarını kullanarak büyük miktardaki verileri işlemek isteyen müşterilerine Hadoop kümelerinin ilklendirilmesini şart koşturmaktadır.

Hadoop, Google'ın MapReduce framework'ünün açık kaynaklı bir gerçekleştirmesidir (Karun ve Chitharanjan, 2013). Büyük bir işi küçük görevlere böler ve büyük veriyi daha küçük parçalara bölerek her görevin farklı parçalarda paralel olarak çalışmasını sağlar. Hadoop, Google File System (GFS)'in açık bir gerçekleştirmesi olan HDFS'i kullanarak verileri depolamaktadır. HDFS, düğümlerden oluşan çok büyük dağıtık bir dosya sistemidir. HDFS dosyaları seri bloklar olarak depolamakla birlikte hata toleransı için replikasyonunu da tutar. Stratejik veri parçalama, işleme, yerleştirme, replikasyon ve veri bloklarının yerleştirilmesi performansı arttırmaktadır. Hadoop ile ilişkili araştırma konularının çoğu bu alanlarda yapılmaktadır. K. Karun ve arkadaşları tarafından yazılan makalede Hadoop ile veri depolama, işleme ve yerleştirme konusunda yapılan önemli iyileştirme önerileri incelenmiştir.



Şekil 3.12. Hadoop ekosistemi

MapReduce modeli, verinin bulunduğu bölgenin (locality) avantajını kullanmaya dayanır, yani veri olduğu yerde veya verinin taşınma süresinin az olacağı yakın bir düğümde işlenir. MapReduce modeli, iki ana adımdan oluşmaktadır. Map adımı ana düğüm girdiyi alır, daha ufak alt problemlere ayırır ve bu alt problemleri işçi düğümlere dağıtır. İşçi düğümler, aldıkları iş parçalarını işler ve ana düğüme cevap döner. Reduce adımı ise ana düğüm tüm alt problemler için cevapları alır ve sonuç çıktısını oluşturmak üzere cevapları birleştirir. MapReduce modeli ile 1 petabaytlık verinin sadece bir saatte sıralanabildiği belirtilmektedir. Ayrıca paralel yaklaşım, analiz operasyonları boyunca oluşabilecek depolama veya kısmi düğüm hatalarının kurtarılabilmesine olanak tanımaktadır. Mapper ya da reducer

fonksiyonlarından birisi başarısızlıkla sonuçlanırsa ilgili iş orijinal veriyle tekrar zamanlanır (Chen ve diğerleri, 2013).

Mevcut HDFS gerçekleştirmesi Java soket arayüzünü kullanarak iletişim sağladığından gecikme ve ortaya çıkan iş anlamında yetersiz performans sağlanmaktadır. Veri yoğun uygulamalar düşünüldüğünde ağ performansı kilit elemanlardan biri olmaktadır. N.S. Islam ve diğerleri (2012) tarafından yapılan çalışmada InfiniBand üzerinden JNI vasıtasıyla uzaktan doğrudan bellek erişimi (RDMA) kullanan yenilikçi bir HDFS tasarımı önerilmiştir. Deneyler göstermiştir ki 5 GB'lık bir HDFS dosyasının yazılması işleminde yeni tasarım iletişim zamanını 87% ve 30% oranlarında düşürmüştür. HBase içinse veri yerleştirme işlemi performansı 26% iyileştirilmiştir. Bilindiği kadarıyla önerilen tasarım InfiniBand ağı üzerinden çalışan ilk HDFS tasarımıdır.

Wu ve diğerleri (2012) tarafından yapılan çalışmada MapReduce üzerinde çalışacak yenilikçi bir planlama işlemi önerilmiştir. Önerilen sistemde Reduce aşamasında boşa geçen zaman içinde dinamik olarak bir sonraki göreve geçilir. Önerilen yaklaşım, 2D-3D uygulamalarda denenmiş ve geleneksel MapReduce'e göre iyileştirme sağladığı gözlemlenmiştir.

Makine öğrenmesi, örüntü tanıma ve yapay sinir ağları gibi teknikler, veriye dayalı birçok alanda analiz yapmak amacıyla yaygın bir şekilde kullanılmaktadır. Artan veri miktarlarına bağlı olarak mevcut modeller daha ölçeklenebilir ve daha verimli modellere dönüştürülmelidir. Chen ve diğerleri (2013) yaptıkları çalışmada uzaktan algılama verileri üzerinde yer örtüsü sınıflandırmada kullanılan ekstrem öğrenme makinesi tekniğini MapReduce tabanlı yaklaşımla hızlandıran bir yenilik önermiştir. Önerilen çatıda ilk fazda çoklu makine ağları, farklı gizli düğüm sayıları ve aktivasyon fonksiyonları map fonksiyonuna gönderilerek eğitilir. İlgili model için gizli düğümlerin biasları, girdi ve çıktı ağırlıkları map fonksiyonunun çıktıları oluşturur. İkinci fazda ise büyük test verisi, paralel olarak map prosedüründe alt veri kümeleri halinde tahmin işlemine girer. Reduce prosedürü her örnek için tahmin edilen sınıfı oylama mekanizmasını kullanarak birleştirir. Gerçek RS verileri kullanılarak yapılan deneyler önerilen sistemin avantaj sağladığını göstermiştir.

Bruce (2013), temel oyun teorisi kavramlarını açıklayarak uzaktan algılama ve yer bilimlerini alanında büyük veri analitiği ve karar verme yaklaşımları için dağıtık sınıflandırıcılarda oyun

teorisi modellerinin uygulanmasını ele almıştır. Önerilen yaklaşımda spektral bant gruplama problemi için stratejik ve rekabetçi oyun teorisi kullanılmıştır. Problemin çözümünde Nash dengesi kullanılarak denge durumuna sahip çözüm bulunur. Yapılan deneylerde önerilen oyun tabanlı hiperspektral bant gruplama modeli, otomatik yer örtüsü sınıflandırma işlemi için çoklu sınıflandırıcı karar birleştirme sisteminde (multi-classifier decision fusion - MCDF) bir bileşen olarak kullanılmıştır. Bu sistemde hiperspektral imza N adet imza grubuna ayrılır. Her bir grup N adet SVM ile sınıflandırılır ve N adet çıktı elde edilir. Karar birleştirme adımı N adet çıktıdan tek bir çıktı elde edilir. Oyun teorisine dayanan sistem, geleneksel bant gruplama yöntemlerine göre daha iyi sonuçlar vermiştir. Bu çalışmada önerilen oyun tabanlı dağıtık sınıflandırıcı sistemi, MapReduce modelinde geliştirilmesi oldukça uygun ve umut verici bir sistem olarak görülmektedir.

Lin ve diğerleri (2013), Açık Mekansal Konsorsiyum'dan WCS, WMS ve WPS gibi standart arayüzleri, HDFS depolama teknolojisini ve MapReduce'tan görüntü işleme işlemlerini kullanan bir çatı önermiştir. Uzaktan algılayıcı görüntüleri için makalede önerilen bulut bilişim çatısı, 4 ana bileşenden oluşur. Kullanıcı, UAV Görüntü Yönetimi vasıtasıyla hava görüntülerini alabilir ve yükleyebilir. WebGIS hava görüntülerini web sayfasındaki dinamik katmana koyarak gösterebilmeyi sağlar. Veri Analizi modülü görüntü verilerinin tutulması ve işlenmesinden sorumludur. Bulut bilişim ortamı ise dağıtık işleme için temel alt yapıyı sağlamaktadır. Önerilen mimaride görüntüler HDFS üzerindeki veri ambarında tutulmakla birlikte MapReduce tekniği kullanılmaktadır. Görüntü verilerinin sınıflandırılması için Hadoop uzantısı olan Mahout projesi kullanılmıştır. Görüntü işleme modülü MapServer ile entegre olarak çalışır. WebGIS görselleştirme amacıyla OpenLayers kullanarak WMS ve WCS arayüzleriyle MapServer'a bağlanarak görüntüleri alırken WPS arayüzü ile görüntü analizi modülünden sonuçları alarak görselleştirir. Farklı veri kümeleriyle yapılan testlerde HDFS ile geleneksel yöntemler okuma/yazma performanslarına göre değerlendirilmiş ve büyük dosyalar (8-10 MB) için HDFS'in daha iyi performans gösterdiği belirtilmiştir.

Guo ve diğerleri (2010), uzaktan algılama görüntülerinin işlenmesi için OpenRS-Cloud isimli, bulut teknolojisini temel alan bir platform geliştirmiştir. İlgili çalışmada geliştirme desenleri, hesaplama modeli, veri yönetimi ve servis modeli konuları üzerinde durulmuştur. OpenRS-Cloud, algoritma geliştirmelerinin basit bir şekilde yapılabilmesi için programlama soyutluğunu arttırmak ve programlama karmaşıklığını azaltmak amacıyla arayüz-yönelimli ve nitelik-yönelimli eklenti algoritma sistemine sahiptir. Hesaplama modeli olarak

MapReduce modelini temel alan ayrıştır-birleştir isimli bir model önerilmiştir. Bu model ile MapReduce’ta değer olarak görüntünün bir alanının birleştirilmesiyle ortaya çıkan verimlilik probleminin çözülmesi sağlanmaktadır. Bu yeni model görev parçalama, alt görev işleme ve alt görev birleştirme adımlarından oluşur. Görev parçalama, MapReduce’taki ayırma işlemine karşılık gelen görevi alt görevlere ayırma işlemidir. Alt görev işleme ise Map fazına karşılık gelmekle birlikte değer algoritma girdi/çıkıtlı niteliklerini taşıyan xml verisidir. Alt görev birleştirmede ise Reduce fazındaki gibi birleştirme yapılır fakat burada aynı anahtara sahip alt görevleri birleştiren sadece bir “Reduce” görevi vardır.

3.5. Değerlendirme

Zaman ve mekan, uzaktan algılama uygulamalarda önemli iki kavram olarak karşımıza çıkmaktadır. Uzaktan algılayıcı uygulamalarında değişiklik analizi gibi önemli analiz işlemleri hem coğrafi koordinat hem de zaman bilgisine sahip görüntüler üzerinde çalışmaktadır. Bu yüzden dağıtık olarak tutulan verilerin yönetilmesinde dağıtık veri tabanı sistemlerinin, verilerin zaman-mekansal (spatiotemporal) bilgilerine göre de sorgulanabilmesi yeteneğini sağlaması gerekir.

Hadoop ekosistemi coğrafi konumların görselleştirilmesi konusunda eksik kalmaktadır. “Geoprocessing Tool for Hadoop” gibi GIS ortamı ile Hadoop sistemi arasında bağlantı kuran araç kitlerine ihtiyaç duyulmaktadır. Hadoop-GIS isimli umut verici bir sistem, Hive üzerinde uzamsal veri ambarı çözümü ve veriyi dağıtmak için uzamsal parçalama sağlasa da github üzerindeki projede yeni bir aktivite görülmemektedir (Aji ve diğerleri, 2013). Ayrıca görselleştirmelerin istenildiği gibi yapılabilmesi için algoritmaların kullandığı vektör ve raster verilerini destekleyen sistemlere ihtiyaç vardır.

İşlemler dizisinden oluşan iş akışının çalıştırılması boyunca birkaç ara sonucun oluşturulmasına ihtiyaç olabilir. Oluşan ara sonuçların dağıtık sistem üzerinde koşturma işlemlere taşınması gerekebilir. Uzaktan algılama verileri ya da veri akıları gibi büyük veri kümesinde çalışan bir işlemin ara sonuçlarının da en azından girdi veri kadar olacağı düşünülürse, bu denli büyük bir ara sonucun taşınması da ekstra bir maliyet oluşturacaktır. Bu noktada ara sonuçların dosya sistemine atılması ve üstverilerin işlemlere dağıtılarak ihtiyaç duydukları ara sonuçlara ulaşmaları sağlanabilir. Kullanılması gereken altyapının da birden fazla işlemin tek bir işlem gibi çalıştırılıp sorgulanmasına imkan vermesi gerekir.

Veri görselleştirme, mevcut araçlarla ancak düşük performanslı olarak yapılabilir. Bilgi görselleştirme için tutulması gereken geçmiş veriler de veri-yoğun işlemlere ihtiyaç duyduğundan verimli yaklaşımlara ihtiyaç duyulmaktadır.

Görüntülerin dağıtık olarak yönetiminin nasıl sağlanacağı, bir görüntünün analizini verimli kılacak parçalama (tiling) stratejilerinin belirlenmesi, parçalanma sonrası dağıtık olarak nasıl tutulacağı ve nasıl işlenebileceği araştırılmalıdır. Verilerin parçalanmasında dikkat edilmesi gereken iki konu mevcuttur. Verilerin yoğun parçalardan oluşmasından kaçınılması gerekir. Böyle bir durum olduğunda eş zamanlı çalışan işçiler arasındaki yük dengesi sağlanamayacaktır. Diğer dikkat edilmesi gereken husus ise parçaların kesiştiği sınır bölgelerdeki nesnelerdir. Dağıtık depolama ve işlemenin daha verimli yapılması için topolojiden haberdar bir parçalama stratejisinin ve kaynak yönetimi yapısının oluşturulması kaynakların daha verimli bir şekilde kullanılabilmesini sağlayabilir.

Uzaktan algılama uygulamalarında analiz amacıyla kullanılmakta olan birçok algoritma yoğun ve düzensiz veri erişimi deseni kullanır. Bu erişim deseni dağıtık dosya sistemleri tarafından nadiren desteklenmektedir. Bu yüzden dağıtık olarak tutulan görüntü dosyalarının düzensiz erişim ihtiyacı bulunan analiz algoritmaları için ayrıca değerlendirilerek dağıtık dosya sistemlerinin bu ihtiyacı karşılayabilecek biçimde ele alınması gerekir.

Uzaktan Algılama uygulamalarında yakın gerçek zamanlı analiz için kullanılan yüksek performanslı hesaplama (HPC) yaklaşımları bireysel başarılarının dışında tamamlayıcı bir şekilde kullanılması eksik bir noktadır. Her ne kadar HPC sistemlerinin rolü ağırlıklı olarak uygulama alanına bağlı düşünülse de küme tabanlı yaklaşımlar büyük veri arşivlerinden bilgi çıkarımı için uygun görünürken yerleşik teknolojilere ihtiyacı olan zaman kritik uygulamalar GPU ve FPGA gibi donanımlar üzerinde geliştirilmeye uygundur. Bunların dışında kalan bir eksik nokta ise tüm bu hesaplama kaynaklarının isteğe bağlı olarak hazır olduğu bulut gibi bir alt yapıdır.

MapReduce çatısını MPI ve OpenMP ile kıyaslayan çalışmalar göstermiştir ki MapReduce büyük verinin yinelemeli işlemlere fazla ihtiyaç duymadığı durumlarda iyi sonuçlar vermektedir (Kang, S. Y. Lee ve K. M. Lee, 2015). Yinelemeli işlemlerin fazla olduğu işlem-yoğun problemlerde ise MPI ve OpenMP'ye göre daha az performans göstermiştir. MapReduce'un işlem yoğun problemlerde yavaş çalışmasından dolayı bir alternatif olarak

öne çıkan Apache Spark, “in memory” hesaplama yaparak yinelemeli işlemlerde MapReduce’tan 10-100 kat daha iyi sonuç üretmektedir (Zaharia ve diğerleri, 2012). Bu anlamda Spark’ın gücünü GPU, FPGA ya da Xeon Phi gibi bir çözüm ile birleştirmek ve iş akışı üzerinde iki seviyeli bir paralelleştirme yapmak muazzam bir hesaplama gücü ortaya çıkarır. Bu noktada görev planlamanın veri mevkisinden ve topolojiden haberdar bir şekilde yapılması ekstra performans sağlayacaktır.

3.6. Motivasyon

Tek bir uydu veri merkezinde toplanan RS veri miktarı dramatik olarak artmış ve sadece bir uydunun ürettiği günlük veri miktarı birkaç terabaytlık değerlere ulaşmıştır (Chen ve Zhang, 2014). Bu kadar fazla miktara ulaşan verilerin depolanması, yönetilmesi, işlenmesi ve taşınması konusunda çözülmesi gereken önemli zorluklar ortaya çıkmaktadır. Bu zorlukların tamamının çözülmesi ve yıllık petabyte seviyesindeki veriyle başa çıkılabilmesi için coğrafi dağıtık bulut kullanmak bir zorunluluk haline gelmiştir.

Literatürdeki çalışmalarda RS verilerinin depolanması ve işlenmesi için büyük veri çerçeveleri ve dağıtılmış dosya sistemleri kullanılarak uzaktan algılama algoritmaları paralel olarak geliştirilmiştir. Veri yönetimini, genellikle metin tabanlı bir yaklaşımla geliştirilen dağıtılmış dosya sistemlerine emanet etmek, uydulardan veya İHA'lardan günlük terabaytlarca veri toplayan veri merkezleri için verimli değildir, çünkü verilerin doğası çok büyük tek dosyalardan oluştuğu için kaba tanelidir (coarse-grained). Bununla birlikte, literatürde sıklıkla kullanılan HDFS’in performansının, verilerin fiziksel olarak uzak konumlara dağıtıldığı heterojen ağlarda ciddi derecede düştüğü kanıtlanmıştır (Di Modica ve Tomarchio, 2022). Bu düşüşün en önemli nedeni ise HDFS’in homojen ağ yapısına uygun olarak tasarlanmış olmasıdır. Bu çalışmalar ayrıca, işlenmeden önce bir ağda veri elde etmede birincil sorun olması gereken veri aktarımı sorununu da göz ardı etmektedir. Bu nedenlerden dolayı, önerilen çalışma, RS alanında çok fazla dikkate alınmayan, ağ ve kaynak kullanımını optimize eden bir model kullanarak RS verilerini elde etmeyi amaçlamaktadır.

Bu tez çalışmasında yapılan değerlendirme neticesinde, verinin kameralardan alınıp bir veri merkezine transfer edilmesi için bir kaynak yönetimi ile en uygun işlemci, depolama, bellek ve ağ kaynaklarının atanması anlık uygulama gereksinimlerinin karşılanmasını sağlayabilecektir. Böylece, önerilen sistem, coğrafi olarak dağıtılmış veri merkezlerine

uzaktan algılama büyük verileri elde etmek için üst seviye bir bulut yöneticisi olarak hizmet verebilir. Üst seviye veri dağıtımı tamamlandıktan sonra, her veri merkezi Hadoop ve benzeri büyük veri çerçevelerini kullanarak kendi içerisinde yönetilebilir.



4. PERFORMANS BİLİNÇLİ ATAMA TABANLI OPTİMİZASYON MODELİ

Büyük RS verilerinin işlenmesi, doğal afetler, iklim değişiklikleri ve askeri/ulusal güvenlik sistemleri gibi zaman açısından kritik uygulamalar için hayati önem taşımaktadır. Verilerin işlenmesindeki güçlükler yalnızca büyük veri hacmiyle değil, aynı zamanda mekansal veri kümelerinin içsel karmaşıklığı ve yüksek boyutlarıyla da ilgilidir (Yang ve diğerleri, 2014). Hadoop ve benzeri teknolojiler, büyük verileri işlemek için jeoloji toplumlarında giderek daha fazla ilgi çekmektedir. Büyük verilerin işlenmesi için bu teknolojilerin benimsenmesi adına birçok araştırma yapılmıştır, ancak dinamik jeo-işleme iş yükünü verimli bir şekilde işlemek için bilgisayar kaynaklarının optimize edilmesi konusunda çok az çalışma bulunmaktadır.

Mevcut yazılım sistemlerinde, hesaplama en pahalı kısım olarak görülmektedir. Günlük toplanan veri miktarının son teknolojilerle birlikte katlanarak artmasıyla birlikte, veri taşımanın performans üzerinde ucuz ve büyük ölçüde paralel olan hesaplama daha büyük bir etkisi vardır (Chen ve Zhang, 2014). Bu noktada, yeni yüksek performanslı sistemlerin veri merkezli paradigmaya uyum sağlamak için kendilerini güncellemeleri gerekmektedir. Yeni sistemler bu uyumu başarmak amacıyla veri lokasyonunu kullanmalıdır. Mevcut sistemler, iletişimin gerçekleşen maliyetini görmezden gelir ve veri taşımayı sanallaştırmak için donanım önbelleği tutarlılığına (cache coherency) güvenir. Artan miktarda veri, işleme elemanları arasında daha fazla iletişim ortaya çıkarmaktadır ve bu durum veri yerelliği (locality) ile ilişkisini (affinity) desteklemeyi gerektirmektedir. Ortaya çıkacak yeni model ile veri döşeme, veri düzeni, dizi görünümleri, görev ve iş parçacığı benzeşimi/ilişkisi ve topoloji bilinçli iletişim kütüphaneleri içeren son teknolojilerle veri yerelliği sağlanmalıdır. Bu teknolojilerin en iyi kombinasyonu, yüksek performanslı hesaplama sistemindeki veri lokasyonunu yönetmek için kapsamlı bir model geliştirilmesine yardımcı olabilir (Karun ve Chitharanjan, 2013).

Bu tez çalışması kapsamında önerilen sistem ile bilgisayar kaynaklarının optimize edilmesi ve ağ topolojisinden yararlanılması sağlanarak coğrafi dağıtık bir bulut üzerinde çalışacak yenilikçi yüksek performanslı hesaplama sistemleri için üst seviye bir veri yönetimi gerçekleştirilmesi hedeflenmiştir.

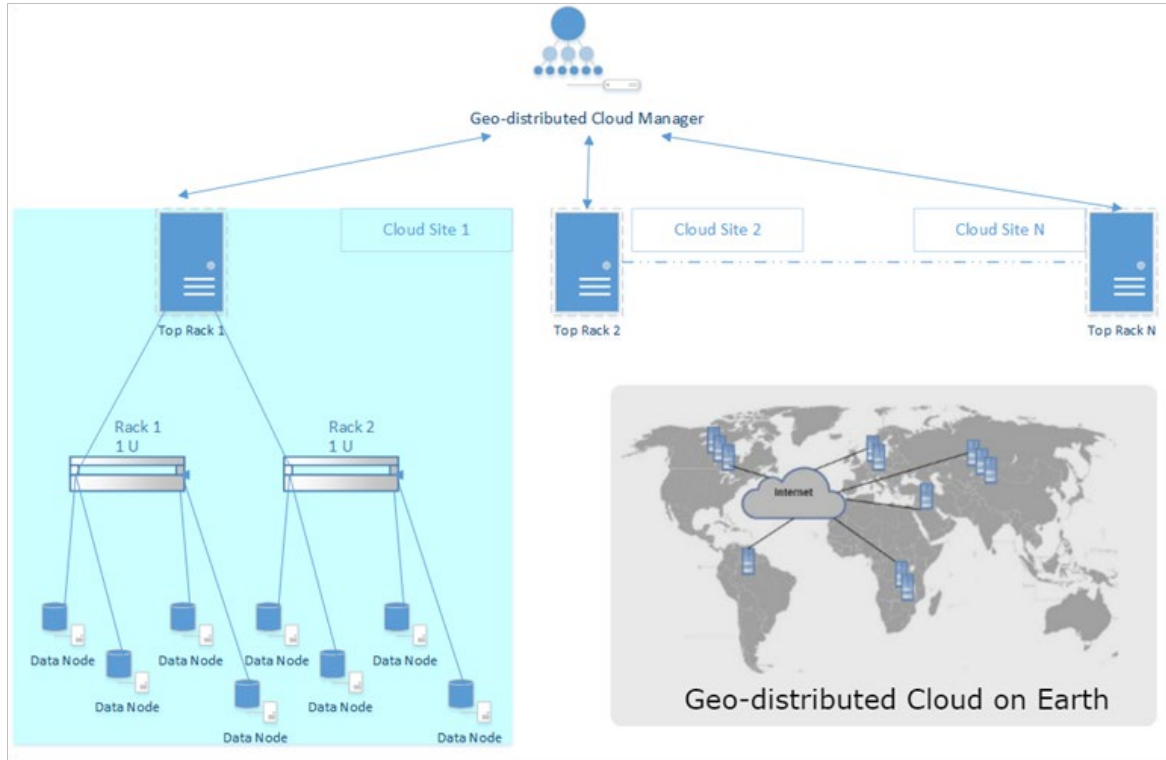
4.1. Coğrafi Dağıtık Bulut

Coğrafi dağıtık bulut, verileri ve uygulamaları birbirine bağlamak için farklı coğrafi bölgelere dağıtılmış birden fazla bulut sitesi içeren bulut bilişim teknolojilerinin uygulanmasıdır (Narayanan, Kansal, Sivasubramaniam, Urgaonkar ve Govindan, 2014). Bulut sağlayıcılar, bulut tüketicileri için daha düşük gecikme süresi ve daha iyi performans sağlamak amacıyla dağıtılmış bulut sistemlerini tercih etmektedir. Son zamanlarda, büyük çevrim içi hizmetlerin çoğu, verilerin katlanarak artmasıyla birlikte coğrafi olarak dağıtılmıştır (Y. Wu ve diğerleri, 2014). Hizmetlerin coğrafi olarak dağıtılmasının en önemli nedeni gecikmedir (latency). Coğrafi dağıtılmış bulutlar, gecikmeleri azaltmak için müşterilere yakın bir yer sağlar. Bu bağlamda, kaynak yönetimi en uygun CPU, depolama, bellek ve ağ kaynakları ile anlık uygulama gereksinimlerini karşılamalıdır (Tso, Jouet ve Pezaros, 2016).

Bu tez kapsamında, coğrafi olarak dağıtılmış bir bulutta RS büyük verilerini yönetmek için yeni bir kaynak paylaşırma (allocation) tekniği önerilmektedir (bkz. Şekil 4.1). Önerilen model, iki aşamalı yeni bir optimizasyon çözme tekniği kullanır ve gerektiği durumlarda çözümsüzlüğü engellemek için Lagrange gevşemesi metodunu kullanarak yaklaşık bir çözüm sunar. Bunu yaparken sensörler ve veri merkezleri arasında veri aktarım süresini ve hata oranını azaltmak için ağ metriği olarak bant genişliği, gecikme ve sıçrama sayısını kullanır. Diğer yandan büyük veri analizinin verimli bir şekilde yapılabilmesi için veri merkezlerindeki işlemci, depolama, bellek kaynakları ve iş yükünü hesaba katarak kaynak paylaşımını optimize eder. Bu noktada çalışmamızın literatürdeki çalışmalardan farkı veri aktarım süresini azaltmak, depolama alanı en uygun veri merkezini seçmek ve veri merkezinin iş yükünü dikkate almaktır.

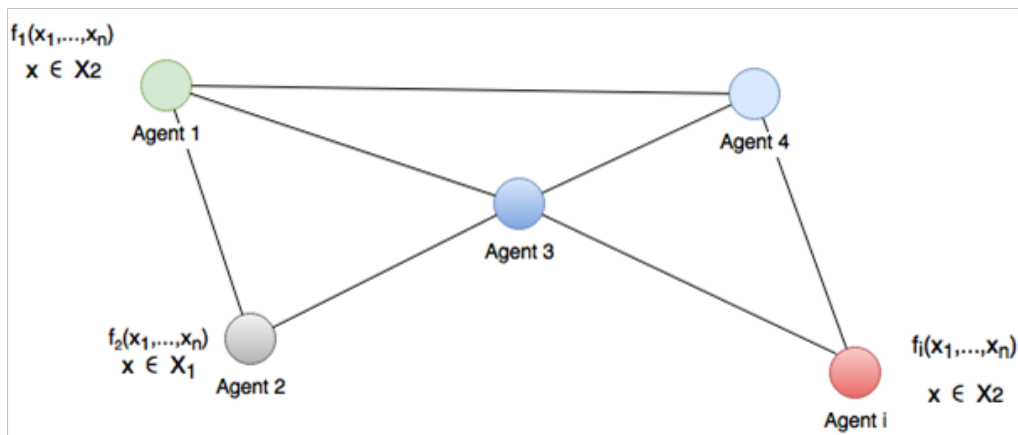
4.2. Performans Bilinçli Atama Metodu

Büyük ölçekli ağlar ve uygulamaları, bilgiye erişimde merkezi bir yapının eksikliğini duymaktadır etkilidir (Nedic ve Ozdaglar, 2010). Buluttaki RS büyük verisini büyük ölçekli bir ağ olarak yorumladığımızda, kaynak tahsisinin optimizasyonu her bir düğümün yerel gözlemlerine ve bilgilerine bağlıdır. Optimum kaynak tahsisini bulmak için kontrol ve optimizasyon algoritmaları aracılığıyla ağ üzerindeki düğümlerden yerel gözlemlerin toplanması gerekir.



Şekil 4.1. Coğrafi dağıtık bulut hiyerarşisi ve yeryüzü görünümü

Sistemin optimizasyonunun başarımı için, ajan olarak çağrılan her bir düğüm yerel olarak sahip olduğu metrikleri merkezi yapıya ileterek optimizasyon algoritmasının çalıştırılmasını sağlar. Bir ağ üzerinden bağlı çoklu ajandan oluşan bu sistemde (bkz. Şekil 4.2), her bir ajan, sadece bu ajan tarafından bilinen bir yerel amaç fonksiyonuna ve yerel kısıtlama kümesine sahiptir. Ajanlar, küresel bir karar vektörüne kendi amaç fonksiyonları ve kısıtlamalarına göre iş birliği yaparak karar vermeye çalışırlar.



Şekil 4.2. Ağ üzerinde kaynak tahsisi için çok ajanlı optimizasyon problemi

$f_i(x)$: Yerel amaç fonksiyonu, $f: \mathbb{R}^n \rightarrow \mathbb{R}$

X_i : Yerel kısıtlama kümesi, $X_i \subset \mathbb{R}^n$

x : Ajanların birlikte karar vermeye çalıştığı küresel karar vektörü, $x \in \mathbb{R}^n$

Ajanlar, yerel amaç fonksiyonlarının bir kombinasyonu olan $f(x)$ ile gösterilen küresel amaç fonksiyonunu kooperatif olarak optimize ederler:

$$f(x) = T(f_1(x), \dots, f_i(x)) \quad (4.1)$$

$f: \mathbb{R}^n \rightarrow \mathbb{R}$ ve karar vektörü x , bir kısıtlama kümesi ile sınırlandırılmıştır, burada $x \in C$, ağ yapısının empoze edebileceği yerel kısıtlamalardan ve küresel kısıtlamalardan oluşur:

$$C = \left(\bigcap_{i=1}^n X_i \right) \cap C_g \quad (4.2)$$

burada C_g küresel sınırlamaları ifade eder. Bu model aşağıdaki optimizasyon problemine öncülük eder:

$$\begin{aligned} & \text{minimize } f(x) \\ & \text{subject to } x \in C \end{aligned} \quad (4.3)$$

burada $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ve C kümesi kısıtlamaları içeren kümedir. Denklem (4.3) içindeki karar vektörü bileşeni, yerel bilgi kullanan ağdaki düğümler tarafından tahmin edilen küresel karar vektörü veya her bir düğüme tahsis edilen kaynaklara karşılık gelen kaynak vektörü olarak düşünülebilir.

Bu tez kapsamında yapılan çalışmada Performans Bilinçli Atama (Performance Aware Assignment - PAA) olarak adlandırılan iki aşamalı yeni bir optimizasyon modeli önerilmiştir. PAA'nın ilk aşamasında, sensörler ve veri merkezleri arasında gecikme, bant genişliği ve sıçrama sayısından oluşan ağ ölçümleri ve veri merkezlerindeki CPU, depolama, bellek ve I/O iş yüklerinden oluşan kaynak miktarları gözetilerek çok kısıtlı (multi-constrained) bir atama problemi çözülmeye çalışılır. Model, bu fazda uygulanabilir olmayan çözümü (infeasible solution) önlemek için sınırsız kaynağa sahip olan sahte bir veri merkezi kullanmaktadır, buna karşın sensörlerin fazladan atanmasını önlemek için bir ceza terimi de

uygulanmaktadır. PAA'nın ikinci aşamasında ise yapay veri merkezine atanmış herhangi bir sensör varsa, Lagrange gevşemesi kullanılarak ilgili düğümler için bireysel çözüm bulunur.

Faz 1'de aşağıdaki 0-1 tamsayı problemi olarak sunulan modelin çözülmesi gerekir:

$$z = \min \sum_{i=1}^n \sum_{j=1}^m (P_{ij} + p_j) x_{ij} + \sum_{j=dummy}^m (P_{ij} + p_j) x_{ij} \quad (4.4)$$

$$\sum_{j=1}^m x_{ij} = 1 \quad \forall i \quad (4.5)$$

$$\sum_{i=1}^n x_{ij} l_{size}^i \leq A_{mem}^j \quad \forall j \quad (4.6)$$

$$\text{subject to} \quad \sum_{i=1}^n x_{ij} l_{size}^i \leq A_{stor}^j \quad \forall j \quad (4.7)$$

$$\sum_{i=1}^n x_{ij} B_{ij} \leq A_{band}^j \quad \forall j \quad (4.8)$$

$$B_{min} \leq B_{ij} \quad (4.9)$$

$$H_{ij} \leq Hop_{max} \quad (4.10)$$

$$IOPS_j \leq IOPS_{max} \quad (4.11)$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \quad (4.12)$$

Burada P_{ij} kamera bilgisayarı ve veri merkezi arasındaki ağ maliyeti, p_j ise veri merkezinin kullanılması durumunda ortaya çıkan kaynak maliyetidir. Bu değerler aşağıdaki gibi formüle edilir:

$$P_{ij} = w_1 L_{ij} + w_2 \frac{1}{B_{ij}} + w_3 H_{ij} \quad (4.13)$$

$$p_j = w_4 W_{CPU}^j + w_5 W_{STR}^j + w_6 W_{MEM}^j + w_7 W_{IOPS}^j \quad (4.14)$$

Formüldeki parametre ve değişkenler aşağıdaki gibi açıklanabilir:

L_{ij} : Kamera bilgisayarı i ile veri merkezi j arasındaki gecikmedir.

B_{ij} : Kamera bilgisayarı i ile veri merkezi j arasındaki bant genişliğidir.

H_{ij} : Kamera bilgisayarı i ile veri merkezi j arasındaki ağ sıçraması sayısıdır.

W_{CPU}^i : Veri merkezi j'nin CPU yüküdür.

W_{STR}^i : Veri merkezi j'nin depolama yüküdür.

W_{MEM}^i : Veri merkezi j'nin bellek yüküdür.

W_{IOPS}^i : Veri merkezi j'nin IOPS yüküdür.

I_{size}^i : Kamera bilgisayarı i'den gelecek olan görüntünün büyüklüğüdür.

x_{ij} : Kamera bilgisayarı i'nin veri merkezi j'ye atanmasını ifade eden karar vektörüdür.

Amaç fonksiyonundaki kısıtlar şu şekilde açıklanabilir:

- Denklem (4.5): her bir görüntünün çözüm matrisinde kesinlikle bir veri merkezine atanmasını sağlar.
- Denklem (4.6): her bir veri merkezine atanan görüntülerin büyüklüklerinin toplamının veri merkezindeki mevcut bellekten küçük olmasını sağlar.
- Denklem (4.7): her bir veri merkezine atanan görüntülerin büyüklüklerinin toplamının veri merkezindeki mevcut depolamadan küçük olmasını sağlar.
- Denklem (4.8): her bir veri merkezine atanan kamera bilgisayarlarının o veri merkezi ile aralarındaki bant genişliği toplamalarının veri merkezindeki mevcut bant genişliğinden küçük olmasını sağlar.
- Denklem (4.9), (4.10) ve (4.11): (4.9) kamera bilgisayarı ile veri merkezi arasındaki bant genişliğinin minimum bant genişliğinden büyük olmasını, (4.10) aradaki sıçrama sayısının maksimum bir değerden küçük olmasını, (4.11) ise veri merkezlerinin belirli bir IOPS'un altında olmasını sağlar.
- Denklem (4.12): Çözüm matrisindeki karar değişkenlerinin 0 ya da 1 olarak atanmasını sağlar.

Formül (4.13) ve (4.14)'te verilen maliyet fonksiyonları ağ ve iş yükü değerlerinin ağırlıklı toplamlarıdır. Burada, ağırlık matrisi $(w_1, \dots, w_7)$, yapılan simülasyonlar vasıtasıyla deneysel bir şekilde sırasıyla 20, 1, 0.5, 6, 5, 4 ve 3 olarak bulunmuştur (bkz. Çizelge 4.1). Bu ağırlık matrisiyle yapılan bir saatlik simülasyonda veri merkezlerindeki depolama alanlarının toplamının yaklaşık %75'i doldurulmuştur. Diğer yandan optimal ağırlıkları bulma konusunda devam eden başka bir çalışma daha var. Optimal ağırlıklar olmasa da mevcut deneysel ağırlık matrisi diğer yöntemlere göre umut verici sonuçlar vermiştir.

Çizelge 4.1. Ağırlık matrisi başarı yüzdesi (depolama 0-1 aralığında ölçeklendirilmiştir)

w1	w2	w3	w4	w5	w6	w7	Toplam Depolama
20	1	0,5	6	5	4	3	0,743
20	1,5	0,2	6	5	4	3	0,735
20	1,5	0,7	6	5	4	3	0,730
20	2	0,5	6	5	4	3	0,726
20	1,5	0,5	18	5	4	3	0,710
20	1,5	0,5	12	5	4	3	0,685
120	0,1	0,2	4	2	4	1	0,670
30	0,1	0,2	1	2	2	4	0,662
30	0,25	0,1	4	4	2	2	0,614
60	0,5	0,1	2	4	4	1	0,607

İkinci aşamada, sahte veri merkezine atanan kamera bilgisayarları için bireysel çözümler bulunur. Bu bireysel çözümlerde RS verileri, kısıtlara bağlı olarak gerekirse farklı veri merkezlerine parçalanarak depolanır. Yani buradaki amaç sorunu alt problemlere ayrıştırarak bireysel bir çözüm elde etmektir. Bu noktada Lagrange gevşemesi ve ayrıştırma tekniği, ilgili bireysel çözümlerin elde edilmesi için kullanılmıştır.

4.2.1. Lagrange gevşemesinin probleme uygulanması

Amaç fonksiyonu, kısıtlardan ötürü çözülemiyorsa Lagrange gevşemesi tekniği uygulanarak ayrıştırılır. Lagrange gevşemesi, karma tam sayı problemlerin kısıtlarıyla çözülmesi için sezgisel bir metottur ve literatürde birçok alanda optimizasyon için kullanılmaktadır. Metodun fikri, problemi karmaşıklaştıran kısıtların Lagrange çarpanı olarak adlandırılan ilişkili vektör ile çarpılarak nesnel fonksiyona eklenmesiyle ayrıştırılmasıdır (Hooker, 2006). Bu yaklaşımı problemimize uyguladıktan sonra, verimli bir çözüm için karmaşık kısıtları Lagrange çarpanı ile çarpıp ekleyerek denklemin ikili (dual) problemini elde ederiz (Denklemler (4.15)).

$$\begin{aligned} \min L(x, \lambda, \gamma) = & \sum_{i=1}^n \sum_{j=1}^m (P_{ij} + p_j) x_{ij} + \sum_{j=1}^m \lambda_j \left(A_{mem}^j - \sum_{i=1}^n x_{ij} I_{size}^i \right) \\ & + \sum_{j=1}^m \gamma_j \left(A_{band}^j - \sum_{i=1}^n x_{ij} B_{ij} \right) \end{aligned} \quad (4.15)$$

$$\text{subject to } \begin{cases} \sum_{j=1}^m x_{ij} = 1 \quad \forall i \\ \sum_{i=1}^n x_{ij} I_{size}^i \leq A_{stor}^j \quad \forall j \\ B_{min} \leq B_{ij} \\ H_{ij} \leq Hop_{max} \\ IOPS_j \leq IOPS_{max} \\ x_{ij} \in [0,1] \quad \forall i, j \end{cases} \quad (4.16)$$

$\lambda_j \geq 0$ ve $\gamma_j \geq 0 \quad \forall j$ ikili değişkenlerdir. İkili problem gerekli sadeleştirme işlemlerinden sonra aşağıdaki biçimi alır (bkz. Denklem (4.17)):

$$\begin{aligned} \min L(x, \mu, \lambda, \gamma) = & \sum_{i=1}^n \sum_{j=1}^m (P_{ij} + p_j) x_{ij} - \sum_{j=1}^m \sum_{i=1}^n \lambda_j x_{ij} I_{size}^i \\ & - \sum_{j=1}^m \sum_{i=1}^n \gamma_j x_{ij} B_{ij} + \sum_{j=1}^m \lambda_j A_{mem}^j + \sum_{j=1}^m \gamma_j A_{band}^j \end{aligned} \quad (4.17)$$

Bazı parantezleme ve sadeleştirme işlemlerinden sonra Denklem (4.18) elde edilir:

$$\begin{aligned} \min L(x, \lambda, \gamma) = & \sum_{i=1}^n \left\{ \sum_{j=1}^m (P_{ij} + p_j - \lambda_j I_{size}^i - \gamma_j B_{ij}) x_{ij} \right\} \\ & + \sum_{j=1}^m \lambda_j A_{mem}^j + \sum_{j=1}^m \gamma_j A_{band}^j \end{aligned} \quad (4.18)$$

Yukarıdaki optimizasyon problemi elde edildikten sonra, bu problem her bir düğüm i için n adet alt probleme ayrıştırılabilir. Böylece her düğüm için tek boyutlu optimizasyon probleminin $[\sum_{j=1}^m (P_{ij} + p_j - \lambda_j I_{size}^i - \gamma_j B_{ij}) x_{ij}]$ çözülmesi gerekir. Bu optimizasyon problemi, düğüm i için kendi faydalanma fonksiyonu ve Lagrange çarpanlarından oluşur. Genel olarak, yineleme başına hesaplamamanın basitliğinden dolayı, elde edilen ikili problemi çözmek için

alt-eğri (subgradient) yöntemi kullanılır. Alt-eğri gibi birinci dereceden yöntemler yüksek doğruluk için daha yavaş yakınsama oranına sahiptir, ancak amacın optimale yakın çözümler bulmak olduğu geniş ölçekli optimizasyon problemlerinde çok etkilidir (Nedic ve Ozdaglar, 2010).

N adet alt problem çözülürken şu amaç fonksiyonu kullanılır:

$$\min \sum_{j=1}^m U_i x_{ij} \quad (4.19)$$

$$\text{subject to} \begin{cases} \sum_{j=1}^m x_{ij} = 1 \quad \forall i \\ \sum_{i=1}^n x_{ij} I_{size}^i \leq A_{stor}^j \quad \forall j \\ B_{min} \leq B_{ij} \\ H_{ij} \leq Hop_{max} \\ IOPS_j \leq IOPS_{max} \\ x_{ij} \in [0,1] \quad \forall i,j \end{cases} \quad (4.20)$$

Burada Lagrange gevşemesinden gelen U_i değeri aşağıdaki gibi formüle edilir:

$$U_i = (P_{ij} + p_j - \lambda_j I_{size}^i - \gamma_j B_{ij}) \quad (4.21)$$

4.2.2. Alt eğri yöntemi

Optimizasyon problemlerinin çözümü için verimli yöntemlerin geliştirilmesi, Lagrange yönteminin sağladığı dualite ilişkilerine bağlıdır. İkili (dual) problem tipik olarak düzgün olmayan dışbükey (nonsmooth convex) bir problemdir ve bu nedenle düzgün olmayan minimizasyon teknikleri onu çözmek için kullanılabilir (Burachik, Gasimov, Ismayilova ve Kaya, 2006). Bu tekniklerin başında alt-eğri yöntemi ve uzantıları gelmektedir. Alt-eğri yöntemi, dışbükey minimizasyon problemlerini çözmek için kullanılan yinelemeli bir yöntemdir. Bir Lagrange problemi $Z(\lambda)$ verildiğinde, gevşetilmiş kısıtlamalar kümesini temsil eden bir A matrisi, buna karşılık gelen bir sağ taraftaki vektör b ve bir başlangıç çarpan vektörü λ_0 , aşağıdaki kural kullanılarak bir çarpan dizisi oluşturulur:

$$\lambda_{k+1} = \lambda_k + \alpha_k (AX_k - b) \quad (4.22)$$

burada X_p , $Z(\lambda_k)$ için optimal bir çözümdür ve α_k , pozitif bir skaler adım boyutudur. Eğer $\alpha_k \rightarrow 0$ ve $\sum_{k=0}^{\infty} \alpha_k \rightarrow 0$ ise “lim sup $Z(\lambda_k)$ ”nin $Z(\lambda^*)$ 'a yakınsadığı gösterilmiştir. Genelde sıkça kullanılan adım boyutu:

$$\alpha_k = \delta_k \frac{(Z^* - Z(\lambda_k))}{\|AX_k - b\|^2} \quad (4.23)$$

burada Z^* , uygun bir çözüm (feasible solution) değeridir ve δ_k , $0 < \delta_k < 2$ 'yi sağlayan bir skalerdir. Bu skaler, genellikle, algoritmanın başlangıcında 2'ye ayarlanır ve sınır, belirli bir sayıda yinelemede gelişmediğinde yarıya bölünür. Algoritma, belirli bir noktaya ulaşmadan önce optimal bir çözüm bulunmadıkça, önceden belirlenmiş sayıda yinelemeden sonra sonlandırılır.

4.2.3. Uygulanabilir başlangıç çözümü

Sezgisel (Heuristic) yöntemler, birçok optimizasyon probleminde önemli rol oynar. Atama problemi gibi tam sayı programlama problemlerinin çoğu NP-tam problem olduğundan kabul edilebilir seviyedeki bir hesaplama süresinde ilgili probleme optimal bir çözüm bulunamaması doğal bir durumdur. Optimal çözüm bulunamaması durumunda sezgisel yöntemler kullanılarak uygulanabilir çözümler bulmak mümkündür. Sezgisel yöntemlerle oluşturulan çözümler problemin büyüklüğünü indirgemede ve uygulanabilir başlangıç çözümü sağlamada kullanılabilir. Yukarıda anlatılan alt-eğri yönteminde başlangıç çözümü sağlamak amacıyla bu tez çalışmasına özel olarak geliştirilen sezgisel yaklaşımda, gevşemeye uğrayan kısıtlar veri merkezi bazında normalize edilir ve ortalamaları alınır. Faz ikide çözüm bekleyen her bir kamera bilgisayarı için bu ortalamalar amaç fonksiyonu ile çarpılır ve daha sonra hepsi toplanarak uygulanabilir başlangıç çözümü değeri bulunmuş olur.

$$b_j^k = \frac{b_j - (b)}{(b) - \min(b)} \quad (4.24)$$

Burada b_j^k , veri merkezi j 'de kaynak miktarı k 'nın 0-1 aralığında ölçeklendirilmiş halidir. b_j , veri merkezlerindeki mevcut kaynak miktarlarını temsil eder. Lagrange gevşemesi yönteminde bellek miktarı ve bant genişliği gevşetildiği için sadece bu iki kaynak türü için ölçeklendirme yapılır.

$$\beta_j^k = b_j^k / \sum_{j=1}^m b_j^k \quad \forall k \quad (4.25)$$

Daha sonra her bir kaynak miktarı k için veri merkezi j 'nin normalleştirilmiş kaynak miktarının toplam normalleştirilmiş kaynak miktarına oranlanmasıyla β_j^k vektörü elde edilir.

$$z_{feas} = \sum_{i=1}^t \sum_{j=1}^m \frac{\sum_{k=1}^2 \beta_j^k}{2} (P_{ij} + p_j) \quad (4.26)$$

Uygulanabilir çözüm z_{feas} , β_j^k vektörünün veri merkezi bazında ortalaması ile ikinci fazda atanması gereken t adet kamera bilgisayarının ilgili veri merkezlerine atanması durumundaki maliyetinin çarpımlarının toplamı ile bulunur.

4.3. Veri Parçalama

Alan ayrıştırma istihbarat, meteoroloji, tarım, şehircilik, arama ve kurtarma gibi alanlardaki yüksek performanslı modelleme ve sayısal simülasyonlar için önemli bir konudur (Gui ve diğerleri, 2016). Büyük bir hesaplama görevini daha küçük parçalara bölerek ve bunları farklı hesaplama kaynaklarına dağıtarak paralel hesaplamayı mümkün kılar. Yüksek performanslı hesaplama, performansı arttırmak için tüm problemi birden çok alt alana bölerek her birini paralel bir şekilde farklı hesaplama düğümlerine dağıtmada yaygın olarak kullanılır. Kaynakların uygun olmayan şekilde tahsisi, dengesiz görev yüklerini ve hesaplama düğümleri arasında gereksiz iletişimi teşvik eder. Böylelikle, kaynak tahsisi paralel sürecin verimliliği üzerinde derin bir etkisi olan hayati bir parçadır. Kaynak ayırma algoritması, iletişim ve hesaplama maliyetini göz önünde bulundurarak toplam yürütme süresini en aza indirmeli ve tüm işlemler için toplam maliyeti de azaltmalıdır. Bu bölümde bahsi geçen iki aşamalı modelde eğer ikinci aşama işletilirse ikinci aşamada çözüm sağlanan kamera bilgisayarları için veriler gerektiği durumda parçalanabilecektir. Bu yeni veri

parçalama stratejisi ile darboğaz yaşanılabilecek olan kaynaklara daha az veri parçası atanarak oluşabilecek kaynak tükenmesi problemi en aza indirilmiştir.

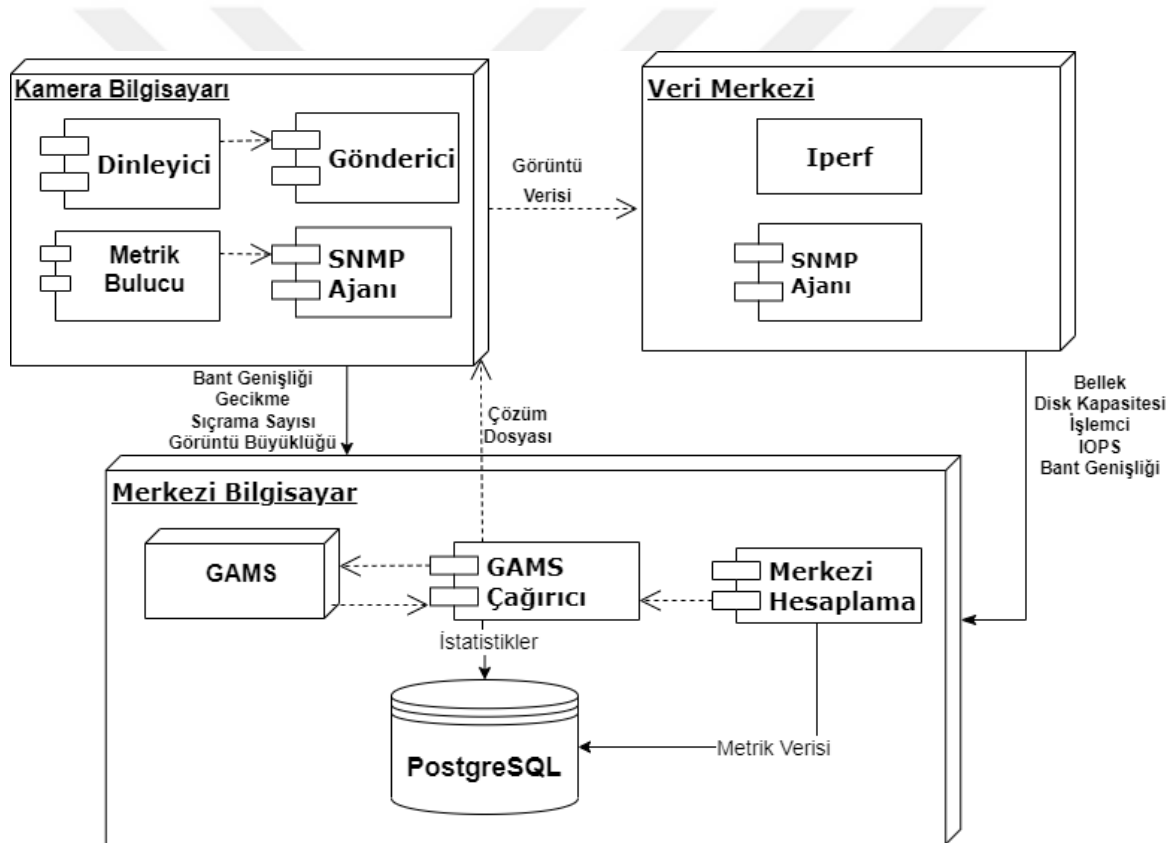


5. YAZILIM GERÇEKLEŞTİRME

Bu bölümde sistemin yapısal ve davranışsal tasarımları detaylı bir şekilde ele alınmaktadır. Bunun yanında önerilen sistemin karşılaştırılmasında kullanılan rastgele atama, macar metodu ve HDFS yaklaşımları anlatılmaktadır.

5.1. Sistemin Yapısal Tasarımı

Şekil 5.1’de görüldüğü üzere sistem 3 farklı ana bileşenden oluşmaktadır: Kamera Bilgisayarı, Veri Merkezi ve Merkezi Bilgisayar.



Şekil 5.1. Bileşen diyagramı

5.1.1. Kamera bilgisayarı

Kamera bilgisayarı, uzak algılayıcıdan gelen görüntüleri alan, uyduya ya da insansız hava aracına bütünlük bilgisi ifade eder. Yazılımsal olarak 3 adet bileşenden oluşur: Metrik Bulucu (Metric Finder), Dinleyici (Listener) ve Gönderici (Sender). Bunlara ek olarak bir

de SNMP Ajanı (SNMP Agent) modülü, ağ üzerinden merkezi bilgisayara metriklerin gönderilmesi ve sonuçların alınması için kullanılmıştır.

“Metrik Bulucu” bileşeni vasıtasıyla kamera bilgisayarları periyodik olarak veri merkezleriyle aralarındaki gecikme, bant genişliği ve sıçrama metriklerini hesaplayarak bir dosyaya yazar. Metrik dosyası merkezi bilgisayar tarafından “SNMP Ajanı” bileşeni vasıtasıyla toplanır.

Algorithm 1: Metric Finder algorithm
Result: Network metric file Input: Data center list initialization; for <i>All data center in list</i> do measure latency between camera pc and data center; measure bandwidth between camera pc and data center; get hop count between camera pc and data center; end generate network metric file; if <i>Camera pc status WAITING</i> then put image size info to network metric file; end

Şekil 5.2. Metrik bulucu bileşeni sözde kodu

“Dinleyici” bileşeni merkezi bilgisayarın çözüm dosyasının kamera bilgisayarına gelip gelmediğini kontrol eder, çözüm dosyası gelmişse “Gönderici” bileşenini çağırarak verinin gönderilmesini tetikler.

Algorithm 2: Listener algorithm
Result: Input: Solution file initialization; status = WAITING; while <i>True</i> do instructions; if <i>Solution file exist</i> then if <i>status = WAITING</i> then status = SENDING; call Sender algorithm for sending data to assigned data center; end status = WAITING; end wait for 1 second; end

Şekil 5.3. Dinleyici bileşeni sözde kodu

“Gönderici” bileşeni, çözüm dosyasında kamera bilgisayarının tayin edildiği veri merkezi ya da merkezlerine verinin gönderilmesinden sorumludur.

Algorithm 3: Sender algorithm
Result: Data sent Input: Assignment vector initialization; parse assignment vector; for <i>All data partition in assignment vector</i> do create subprocess for sending data; send partition to assigned data center; end wait for finishing all subprocesses;

Şekil 5.4. Gönderici bileşeni sözde kodu

5.1.2. Merkezi bilgisayar

Merkezi bilgisayar, veri merkezlerinden ve kamera bilgisayarlarından metrikleri toplayan, bu metrikler dahilinde atama probleminin çözüldüğü ve sonuçların kamera bilgisayarına gönderildiği bilgisayardır. Yazılımsal olarak 2 adet bileşenden oluşur: “Merkezi Hesaplama” ve “GAMS Çağırıcı”. Bunlarla birlikte merkezi bilgisayar üzerinde üçüncü parti yazılım olarak optimizasyon modelinin çözülmesi için kullanılan GAMS (General Algebraic

Modeling System) yazılımı (GAMS, 2021), alınan metrikler ile sonuçların tutulduğu PostgreSQL veri tabanı ve ağ üzerinden metriklerin toplandığı SNMP yönetici yazılımı mevcuttur.

“Merkezi Hesaplama” bileşeni optimizasyona başlamadan önce SNMP protokolünü kullanarak her kamera bilgisayarından veri merkezleriyle arasındaki ağ metriklerine ait dosyalardaki değerleri alır. Diğer yandan yine SNMP protokolü ile veri merkezlerindeki iş yüklerini (işlemci, I/O, bellek, depolama) almak bu bileşenin görevidir. Alınan tüm metrikler GAMS’de çalıştırılacak girdi dosyasına uygun olarak dizelenir. Girdi dosyası oluşturulduktan sonra “GAMS Çağırıcı” bileşeni tetiklenir.

Algorithm 4: Central Compute algorithm

```

Result: GAMS model file
Input: network metrics, resource metrics
initialization;
for All camera pc do
    | get network metrics;
    | if new image taken then
    | | put camera pc to valid pc list;
    | end
end
for All data center do
    | get resource metrics;
    | if not exhausted and reachable then
    | | put center to data center list;
    | end
end
for All camera pc in valid pc list do
    | for All data center in data center list do
    | | put network metrics to GAMS model file
    | | end
    | put image size to GAMS model file
end
for All data center in data center list do
    | put resource metrics to GAMS model file
end
call GAMS Caller;
write data to database;

```

Şekil 5.5. Merkezi hesaplama bileşeni sözde kodu

“GAMS Çağırıcı” bileşeni, mevcut girdi dosya ile GAMS çözücüyü çağırarak modelin çözümlenmesini sağlar. Modelin çözümü yoksa önerilen yaklaşımın ilk fazında GAMS üzerinde sahte veri merkezi ile modeli çözer. İkinci aşamada çözümde sahte veri merkezine atanan kamera bilgisayarları için bireysel olarak Lagrange çözümleri oluşturulur ve gerekiyorsa ilgili veriler için parçalama gerçekleştirilir. Tüm bu işlemlerden sonra GAMS’in ilk fazdaki çıktı dosyası parse edilerek sonuç matrisi bulunur ve bu matrise bireysel çözümler dahil edilerek kamera bilgisayarlarına yollanır. Bu işlemlere ek olarak kaç bilgisayar için çözüm üretildiği, kaçında bireysel çözüm uygulandığı, gelen veri miktarı, atanan veri miktarı ve GAMS çözücünün işlem süreleri istatistiksel veri olarak veri tabanına kaydedilir. Veri tabanında tutulan veriler ile modelin analizinin kolaylaştırılması hedeflenmiştir.

Algorithm 5: GAMS Caller algorithm
Result: Solution file Input: GAMS model file initialization; run pure GAMS with GAMS model file; if <i>feasible solution found</i> then generate solution file with pure GAMS solution; else run pure GAMS dummy with GAMS model file; get camera pc assigned to dummy data center in solution; for <i>All camera pc assigned to dummy data center</i> do generate 1-d GAMS model for camera pc; run GAMS with Lagrange Relaxation; if <i>particular solution found</i> then put particular solution to solution file; end end end write statistical information to database; for <i>All camera pc in solution file</i> do send solution file to camera pc; end

Şekil 5.6. GAMS çağırıcı bileşeni sözde kodu

5.1.3. Veri merkezi

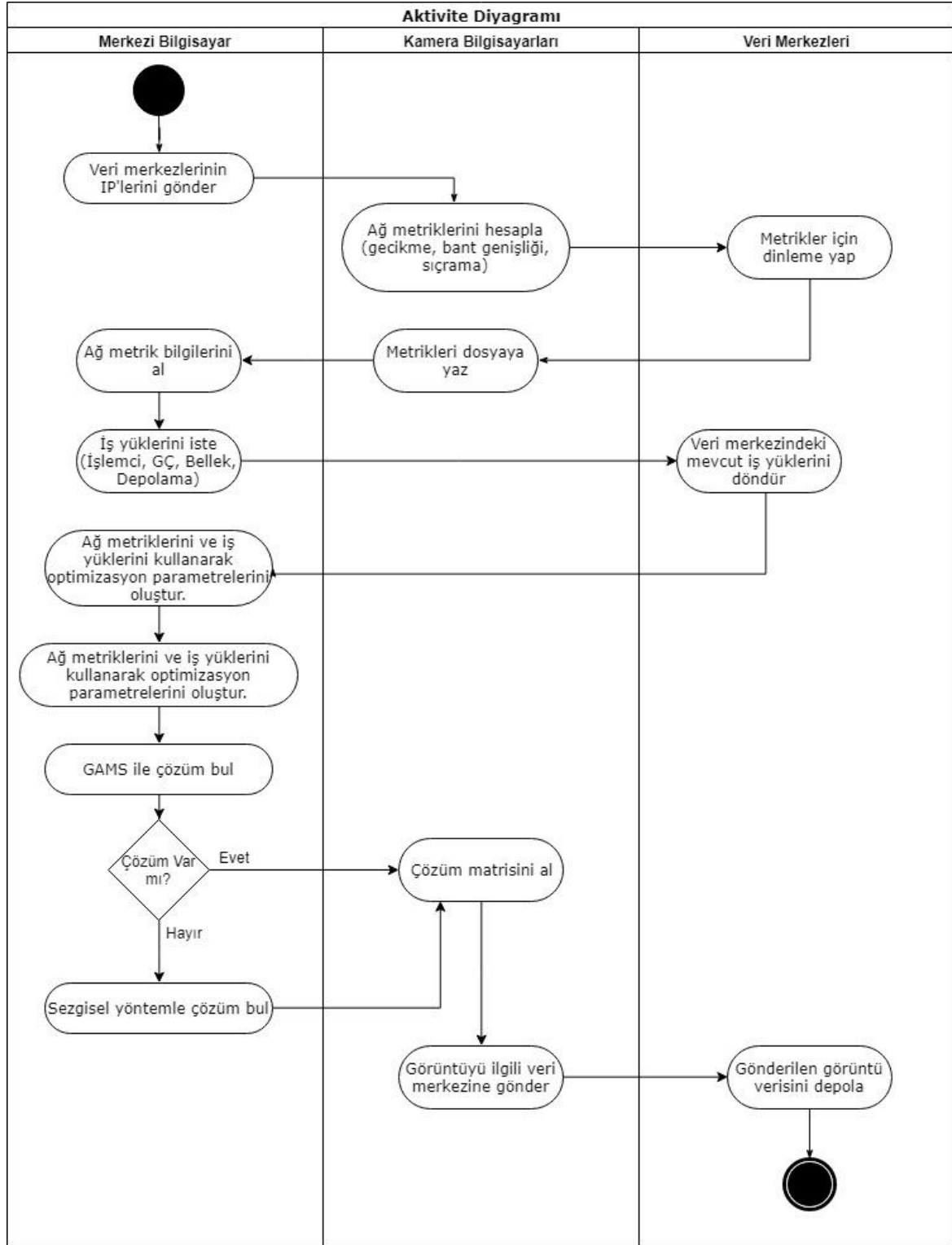
Veri merkezi, kamera bilgisayarlarının aldığı uzaktan algılama verilerinin yeryüzüne inidilerek depolandığı ve işlendiği merkezdir. “SNMP Ajanı” ve “iPerf” (iPerf, 2016) üçüncü parti yazılımlarından meydana gelir.

“SNMP Ajanı” bileşeni veri merkezindeki iş yüklerinin (İşlemci, I/O, Bellek, Depolama) ve sahip olduğu kaynakların bilgisini merkezi bilgisayara yollamasından sorumludur.

“iPerf” bileşeni kamera bilgisayarlarının veri merkezi ile arasındaki bant genişliğini anlık olarak ölçebilmesini sağlayan bir servistir.

5.2. Sistemin Davranışsal Tasarımı

Merkezi bilgisayar, veri merkezleri ve kamera bilgisayarlarının koordinasyonunu sağlayarak çekilen bir görüntünün mevcut metrikler dahilinde en sağlıklı veri merkezinde tutulmasından sorumludur. Şekil 5.7’de görüldüğü üzere ilk olarak kamera bilgisayarlarına belirli periyotlarla veri merkezi IP’lerini gönderir. Kamera bilgisayarları aldığı IP listesindeki veri merkezleriyle çeşitli komutlar kullanarak iletişim kurar ve veri merkezleriyle arasındaki gecikme, bant genişliği ve sıçrama metriklerini hesaplayarak bir dosyaya yazar. Merkezi bilgisayar optimizasyona başlamadan önce SNMP protokolü kullanarak her kamera bilgisayarından veri merkezleriyle arasındaki bu metrik dosyasındaki değerleri alır. Metrikleri aldıktan sonra veri merkezlerinden SNMP kullanılarak İş yükleri (İşlemci, GÇ, Bellek, Depolama) alınır. Ağ metrikleri ve veri merkezi metrikleri bir dosyaya konulur. Oluşturulan dosya merkezi bilgisayarda GAMS’te geliştirilmiş Lagrange modeline girdi olarak yollanır. GAMS’in çıktı dosyası parse edilerek sonuç matrisi bulunur ve bu matris çözüm dosyası olarak kamera bilgisayarlarına yollanır. Kamera bilgisayarı çözüm dosyasında hangi veri merkezine verisini göndereceğini görür ve görüntü ilgili veri merkezine aktarılmaya başlar.



Şekil 5.7. Önerilen model için aktivite diyagramı

5.3. Karşılaştırma için Gerçekleştirilen Yöntemler

Tez kapsamında önerdiğimiz sistemde bulunan atama modelimiz PAA'nın farklı atama yöntemleriyle karşılaştırılması adına rastgele atama, Macar metodu ve HDFS yöntemleri gerçekleştirilmiştir.

5.3.1. Rastgele atama

Rastgele atama (random assignment - RA), bir deney ya da eşleştirmede katılımcıları farklı gruplara atayan deneysel bir tekniktir (Witte, R. S. ve Witte, J. S., 2017). Bu teknikte atama yapılırken bir şans prosedürü (yazı tura atmak vb.) veya rastgele sayı üretici (random number generator) gibi bir rastgeleleştirme kullanılır. Bu durum herhangi bir katılımcının herhangi bir gruba atanmasında eşit şansa sahip olmasını sağlar. Bu noktada yapılan gerçekleştirmede her bir kamera bilgisayarı (katılımcı), depolama alanı müsait olan veri merkezlerinden (grup) birine stokastik olarak atanır.

5.3.2. Macar metodu

Macar metodu (Hungarian algorithm/method - HA), atama problemleri için en sık kullanılan çözüm yöntemlerinden biridir (Mills-Tettey, Stentz ve Dias, 2007; Durmaz ve Koyuncu, 2019; Amalia, Saikhu ve Soelaiman, 2021). Kombinatorial bir optimizasyon algoritması olan bu yöntemde atama problemi polinomiyal zamanda çözümlenir. Çözüme ulaşmak için her adımda problem 0-1 matrisine dönüştürülerek azaltılmış matrisler elde edilmeye çalışılır. Kamera bilgisayarı ile veri merkezi arasındaki atamada kullanılan maliyet değeri ($cost_{i,j}$) aşağıdaki formül ile elde edilir:

$$cost_{i,j} = (P_{ij} + p_j)x_{ij} \quad (5.1)$$

Burada P_{ij} kamera bilgisayarı i ve veri merkezi j arasındaki ağ maliyeti, p_j ise veri merkezinin kullanılması durumunda ortaya çıkan kaynak maliyetidir. Bu değerler Formül (4.13) ve (4.14)'te verildiği gibi kullanılmaktadır. Örnek olarak, bir Yöneylem Araştırması kitabından iyi bilinen bir maliyet matrisi Şekil 5.8'de verilmiştir (Gupta, 1992: 391).

	a	b	c	d	e
A	160	130	175	190	200
B	135	120	130	160	175
C	140	110	155	170	185
D	50	50	80	80	110
E	55	35	70	80	105

Şekil 5.8. Macar metodu için örnek maliyet matrisi

Her satırdaki en küçük elemanı o satırdaki her elemandan çıkartarak, satır azaltılmış matris (row reduced matrix) elde edilir (bkz. Şekil 5.9). Satır azaltılmış matrisin her sütunundaki en küçük öğeyi, o sütundaki her öğeden çıkartarak birinci modifiye matris (I modified matrix) elde edilir (bkz. Şekil 5.9). Sıfır olan satır ve sütunların toplamı 3 olduğundan ve 5 olmadığından dolayı tüm matris kapsanmaz.

Row Reduced Matrix					I Modified Matrix				
30	0	45	60	70	30	0	35	30	15
15	0	10	40	55	15	0	0	10	0
30	0	45	60	75	30	0	35	30	20
0	0	30	30	60	0	0	20	0	5
20	0	35	45	70	20	0	25	15	15

Şekil 5.9. Birinci modifiye matrisin elde edilmesi

Birinci modifiye matriste kapsanmayan kısımdaki minimum değeri (15) tüm kapsanmayan elemanlardan çıkarıp aynı elemanı yatay kesişme noktasına ekleyerek ikinci modifiye matris elde edilir (bkz. Şekil 5.10). Sıfır olan satır ve sütunların toplamı birbirine eşit olduğundan azaltma işlemi sonlandırılır. Elde edilen son matriste tek 0'a sahip olan sütunlar (a ve c) kare içine alınarak atanır (bkz. Şekil 5.10). Geriye kalanlarda ataması yapılanların üstü çizilerek kalan 0 değerler ile tüm matrisin kapsanması sağlanır. Böylece çözüm matrisi elde edilmiş olur.

II Modified Matrix											
15	0	20	15	0			a	b	c	d	e
15	15	0	10	0		A	15	∅	20	15	0
15	0	20	15	5		B	15	15	0	10	∅
0	15	20	0	5		C	15	0	20	15	5
5	0	10	0	0		D	0	15	20	∅	5
						E	5	∅	10	0	∅

Şekil 5.10. İkinci modifiye matris ve çözümün elde edilmesi

5.3.3. HDFS kullanımı

Bölüm 3.4.1'de anlatılan ve birçok büyük veri sisteminde dağıtık dosya sistemi olarak kullanılan HDFS teknolojisi, kıyaslama amacıyla RS verilerinin tutulması için kullanılmıştır. Bu işlemi gerçekleştirmek için merkezi bilgisayar NameNode olarak ve dört veri merkezi DataNode olarak kullanıldı. Bu sistemde çoklu DataNode'lar NameNode'a bağlanır ve NameNode görüntü verilerini bloklara bölerek DataNode'lar üzerine yönlendirir. Yönlendirme işlemi yapılırken DataNode'lar üzerinde depolanan veri miktarları baz alınarak dengeleme işlemi yapılır. HDFS ile yapılan çalıştırmalardaki konfigürasyon dosyasında blok boyutu 128 MB ve 1 replika (replica) olarak kullanılmıştır.

6. DENEYSEL ÇALIŞMALAR

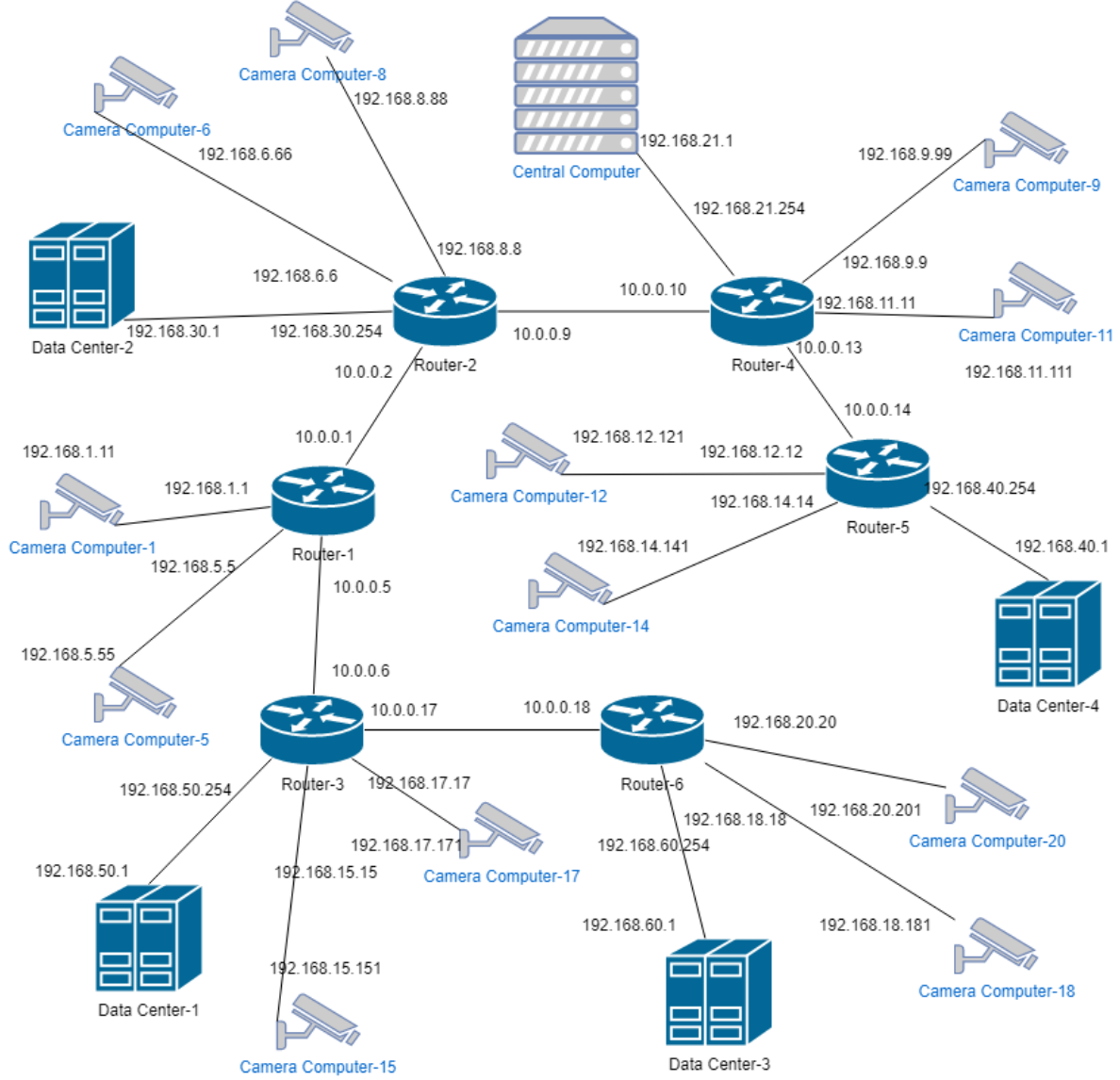
6.1. Simülasyon Ağının Kurulması

Önerilen modelin test edilebilmesi için simülasyon ortamı olarak Gazi Üniversitesi bünyesinde bulunan küme bilgisayarlar üzerinde 2 farklı ağ topolojisi kurulmuştur. Birincisi heterojen ağ, ikincisi ise farklı ağ topolojileri üzerinde karşılaştırılacak yöntemleri analiz etmek için homojen ağ olarak tasarlanmıştır. Sanal makinelerin oluşturulması için VMware ESXI 7.0.3 hipervizörü kullanılmıştır. Hipervizörün modeli ProLiant DL380p Gen8'dir ve 2 adet 24 mantıksal işlemciye sahip Intel(R) Xeon(R) CPU E5-2620 2,00 GHz işlemci ile 256 GB bellekten oluşmaktadır.

Heterojen ağ üzerinde sanal makine olarak bir adet 4 GB RAM, 4 CPU, 40 GB depolama özelliklerine sahip merkezi bilgisayar; 4 adet 20 GB RAM, 8 CPU, 120 GB depolama özelliklerine sahip veri merkezi; 20 adet 4 GB RAM, 2 CPU, 120 GB depolama özelliklerine sahip kamera bilgisayarları konuşlandırılmıştır (bkz. Şekil 6.1). Bunlara ek olarak ağ üzerinde sıçrama sayısını da benzetebilmek için 6 adet 1 GB RAM, 1 CPU, 20 GB depolama özelliklerine sahip yönlendirici (router) sanal makine kurulmuştur. Tüm sanal makinelerde işletim sistemi olarak Ubuntu 18.04 kuruludur. Ağın alt yapısında yönlendiriciler üzerinde Quagga yazılımı (Quagga, 2018) kurulmakla birlikte yapılan konfigürasyonlarla ilk açık yöne öncelik (open shortest path first - OSPF) protokolü işletilmiştir. Homojen ağ ise veri merkezleri ve kamera bilgisayarları için heterojen ağ ile aynı miktar ve yapılandırmaya sahiptir, ancak aynı yapılandırmaya sahip yalnızca 5 yönlendirici bilgisayar kullanılmıştır (bkz. Şekil 6.2). Her yönlendirici beş kamera bilgisayarına bağlıdır ve yönlendirici 5, bunları veri merkezlerine bağlamak için ağ geçidi görevi görür. Homojen bir ağda, her kamera bilgisayarını veri merkezinden eşit mesafede bulunur. Buna karşılık, heterojen bir ağ, kamera bilgisayarları ve veri merkezleri arasında değişken mesafelere sahiptir.

Simülasyon ortamında tüm kamera bilgisayarları, eş zamanlı olarak büyüklükleri 0,41 GB ile 7,32 GB aralığında değişen uzaktan algılama görüntü verilerini, merkezi bilgisayarda koşan optimizasyon metotlarının ürettiği atama sonuçlarına göre uygun veri merkezlerine gönderirler. Simülasyon çalışması kapsamında atama metodu olarak önerdiğimiz PAA

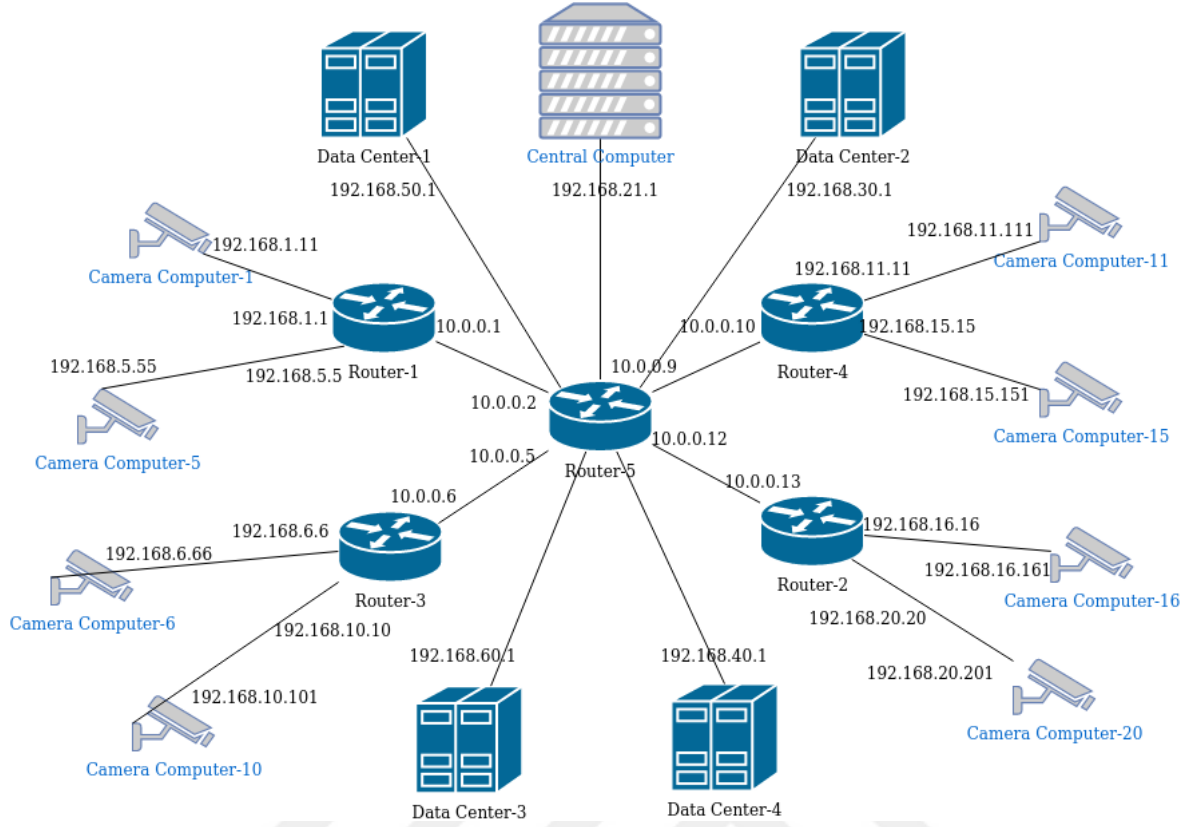
metodu ile HA, RA ve HDFS metotları ayrı ayrı koşturularak kaynak miktarlarındaki değişim analiz edilmiştir.



Şekil 6.1. Önerilen modelin simülasyonu için kurulan heterojen ağ

Çizelge 6.1. Sanal ağda kurulan bilgisayarların özellikleri

Cihaz Türü	Adet	Bellek	Depolama	İşlemci Sayısı
Merkezi Bilgisayar	1	4 GB	40 GB	4
Veri Merkezi	4	20 GB	120 GB	8
Kamera Bilgisayarı	20	4 GB	120 GB	2
Yönlendirici (Heterojen Ağ)	6	1 GB	20 GB	1
Yönlendirici (Homojen Ağ)	5	1 GB	20 GB	1



Şekil 6.2. Önerilen modelin simülasyonu için kurulan homojen ağ

Kurulan sanal ağ üzerinde ağ metrikleri araştırıldığında (bkz. Çizelge 6.2), kamera bilgisayarları ve veri merkezleri arasındaki anlık bant genişliği ortalaması 20 ile 200 MBPS aralığındadır. Öte yandan, ağdaki ortalama gecikme miktarı 0,5 ile 10 saniye arasında değişir. Şekil 6.1’de görüldüğü gibi, heterojen ağdaki atlama sayıları 2-7 aralığındadır. Homojen ağdaki sıçrama sayısı ise Şekil 6.2’de görüldüğü üzere 3 olmaktadır. Veri merkezlerindeki depolama, bellek, işlemci ve IOPS yükleri, 0-1 aralığında normalleştirilmiş. Mevcut depolama miktarları veri merkezlerinde 0-100 GB aralığında iken, bellek miktarları ise 5-19 GB aralığında değişmektedir.

Çizelge 6.2. Sanal ağ üzerinde gerçekleşen metrikler

Metrik	Minimum	Maksimum
Bant Genişliği (MBPS)	20	200
Gecikme (s)	0,5	10
Sıçrama (Heterojen Ağ)	2	7
Sıçrama (Homojen Ağ)	3	3
İş Yüğü	0	1
Bellek (GB)	5	19
Depolama (GB)	10	100

6.1.1. Quagga ile ağ konfigürasyonu

Quagga, ağ için dinamik yönlendirmeyi yöneten bir pakettir (Quagga, 2018). Bu paket içerisinde tümü zebra arka plan programı (daemon) tarafından yönetilen OSPF, RIP, BGP, vb. yönlendirme protokolleri mevcuttur.

OSPF, bir iç ağ geçidi protokolüdür; LAN'lar ve internet üzerinden bağlı LAN'lar içindir. Ağdaki her OSPF yönlendiricisi, tüm ağ için topolojiyi içerir ve ağ üzerinden en iyi yolları hesaplar. OSPF, algıladığı tüm ağ değişikliklerini otomatik olarak çok noktaya yayınlar. Yönlendirme tablolarını yönetilebilir tutmak için ağı alanlara bölebilir; her alandaki yönlendiricilerin, ağ için tüm yönlendirme tablosundan ziyade yalnızca alanlarından bir sonraki sıçramayı bilmesi gerekir.

RIP, yönlendirme bilgi protokolü, daha eski bir protokoldür. RIP yönlendiricileri, OSPF'nin yaptığı gibi yalnızca değişiklikler yerine, tüm yönlendirme tablolarını periyodik olarak çok noktaya yayınlar. RIP, rotaları atlama noktalarına göre ölçer ve 15 atlamanın üzerindeki herhangi bir hedefi ulaşılamaz olarak görür. RIP'in kurulumu kolaydır, ancak OSPF hız, verimlilik ve ölçeklenebilirlik açısından daha iyi bir seçimdir.

BGP, sınır ağ geçidi protokolü, internet trafiğini yönlendirmek için bir dış ağ geçidi protokolüdür (EGP). İnternet servis sağlayıcısı olunmadığı sürece BGP'yi kullanmaya gerek yoktur. Bu sebeplerden dolayı kurulan ağda Quagga OSPF protokolü kullanılmıştır.

Yönlendirici ayarları

Yönlendiricilere kurulan Quagga arka plan uygulamalarını yapılandırmak için “/etc/quagga/daemons” dosyası düzenlenerek zebra ve ospfd aktifleştirilir. Böylece protokol olarak OSPF'nin ağ üzerinde çalışması sağlanır. Kurulan sanal ağ üzerine konuşlandırılan 6 adet yönlendirici bilgisayarda “/etc/quagga/daemons” dosyasının konfigürasyonu Şekil 6.3'teki gibi yapılmıştır.

```
zebra=yes
bgpd=no
ospfd=yes
ospf6d=no
ripd=no
ripngd=no
```

Şekil 6.3. Yönlendirici bilgisayarda Quagga konfigürasyonu

Yönlendiricilerde IP forwarding, ilgili değişkene “1” atanarak aşağıdaki gibi etkinleştirilir:

```
# enable ip forwarding
echo "1" > /proc/sys/net/ipv4/ip_forward
```

Şekil 6.4. Yönlendirici bilgisayarda IP forwarding özelliğinin etkinleştirilmesi

Quagga’yı OSPF protokolü ile yapılandırmak için “/etc/quagga/ospfd.conf” isimli dosya oluşturularak içerisine aşağıdaki ağ konfigürasyon bilgileri eklenir. Burada örnek olarak yönlendirici 1 için 2 yönlendirici ve bir kamera bilgisayarına ait ağ adresi tanımlanmıştır. Arayüz olarak da bağdaştırıcı isimleri yine aynı dosyada belirtilmelidir. Ayarlar bittikten sonra “systemctl start ospfd” komutu kullanılarak Quagga yazılımı OSPF protokolü ile çalıştırılmış olmalıdır.

```
hostname router1
router ospf
    network 10.0.0.0/30 area 0
    network 10.0.0.4/30 area 0
    network 192.168.1.0/24 area 0
interface ens192
interface ens224
interface ens34
```

Şekil 6.5. Yönlendirici bilgisayarda OSPF konfigürasyonu

Diğer bilgisayarların ayarlanması

Merkezi bilgisayar, veri merkezleri ve kamera bilgisayarları, “/etc/network/interfaces” dosyasında bulunan konfigürasyonlar ile yönlendiriciler üzerinden bağlantı kurarlar. Aşağıdaki gibi bir bağdaştırıcı (ens192) için ağ tanımları yapılmakla birlikte bu bağdaştırıcı için ayrıca rota tanımlamaları eklenir. Böylece bilgisayar alacağı veya göndereceği verileri ayarlanan ağ konfigürasyonu üzerinden yönlendiriciye iletir. Yönlendiriciler vasıtasıyla ilgili veri uçtan uca en kısa yoldan taşınmış olur.

```
auto ens192
iface ens192 inet static
    address 192.168.1.11
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255

    post-up route add -net 192.168.0.0 netmask
255.255.0.0 gw 192.168.1.1 dev ens192

    pre-down route del -net 192.168.0.0 netmask
255.255.0.0 gw 192.168.1.1 dev ens192
```

Şekil 6.6. Bilgisayarların ağ konfigürasyonları

6.1.2. SNMP ile bilgi yönetimi

SNMP protokolü, büyüyen bilgisayar ağları üzerindeki çeşitli birimleri denetlemek ve yönetmek için tasarlanmış bir ağ protokolüdür. Cihazların üzerindeki işlemci kullanımı, bellek miktarı, disk miktarı, saniyedeki I/O miktarı (IOPS), bağlantı hızı, sıcaklık, bağlı kullanıcılar vb. bilgiler SNMP’de bulunan ağaç yapısında tutulmaktadır. SNMP’yi oluşturan bileşenler şunlardır: Ajan uygulama, yönetici uygulama, ağ yönetim sistemi ve yönetim bilgi birimleri (management information base - MIB).

Ajan uygulama

Ağ üzerindeki cihazlarda istenilen bilgileri tutan ve yönetici uygulamaya aktaran uygulamadır. Diğer yandan yönetici uygulamadan gelen cihaz değişikliklerini alarak uygulayan bileşendir.

Yönetici uygulama

Ağ yönetici tarafında kullanılan bu uygulama, ajan uygulamayı kullanan bir cihazdan SNMP aracılığıyla bilgilerin alınmasını sağlamaktadır. Aynı zamanda cihaza gönderilecek bilgileri ya da uygulanacak konfigürasyonları göndermekten sorumludur. Bu uygulamanın çalıştığı bilgisayar aynı zamanda ağ yönetim sistemini de çalıştırmaktadır.

Ağ yönetim sistemi

Yönetici birimde çalışmakta olan, ağdaki bütün cihazların izlenebilmesine ve yönetilebilmesine olanak sağlayan uygulamadır.

Yönetim bilgi birimleri (MIB)

MIB, Ajan ile yönetici uygulama arasında iletilecek bilgilere ait yapıların hiyerarşik olarak tutulduğu koleksiyonlara verilen isimdir. SNMP ile belirli bir değişkene ait değerin alınabilmesi için evrensel olarak tanımlı bu koleksiyonun ilgili biriminin kullanılması gerekmektedir. Merkezi bilgisayar tarafından kullanılan SNMP komutları, MIB tanımları ve açıklamaları Şekil 6.7’de verilmiştir.

6.2. Kullanılan Veri Kümesi

AVIRIS, Yeryüzü Uzaktan Algılama alanında kanıtlanmış bir araçtır. 400 ila 2500 nanometre dalga boyuna sahip 224 ardışık spektral kanalda (bantlarda) yükselen spektral ışımalarının kalibre edilmiş görüntülerini sunan özgün bir optik sensördür (AVIRIS, 2021). AVIRIS dört uçak platformunda uçmaktadır: NASA'nın ER-2 jeti, Twin Otter International'ın turboprop, Scaled Composites'in Proteus ve NASA'nın WB-57 platformları. ER-2, deniz seviyesinden yaklaşık 20 km yükseklikte, yaklaşık 730 km/sa hızla uçar. Twin

Otter uçağı, 130 km/sa hızla yer seviyesinden 4 km yükseklikte uçar. AVIRIS, Kuzey Amerika, Avrupa, Güney Amerika'nın bazı bölümleri ve Arjantin'de uçarak veri toplamaktadır.

```
#Kamera bilgisayarlarından file isimli dosyada bulunan ağ
metrikleri ve veri büyüklüğü bilgileri alınır:
snmpget -v2c -c testing $camera_id 'NET-SNMP-EXTEND MIB::nsExtendOutputFull."file"'

#Veri merkezlerinden işlemci yükleri alınır:
snmpwalk -v2c -c testing ${center_ids[i]} HOST-RESOURCES-MIB::hrProcessorLoad.196608

#Veri merkezlerinden depolama bilgileri alınır:
snmpget -v2c -c testing ${center_ids[i]} UCD-SNMP-MIB::diskAvail.1

#Veri merkezlerinden bellek bilgileri alınır:
snmpget -v2c -c testing ${center_ids[i]} UCD-SNMP-MIB::memAvailReal.0

#Veri merkezlerinden IO bilgileri alınır:
snmpwalk -v2c -c testing ${center_ids[i]} UCD-DISKIO-MIB::diskIOLAt.9

#Veri merkezlerinden bant genişliği bilgileri alınır:
snmpwalk -v 2c -c testing ${center_ids[i]} IF-MIB::ifHighSpeed.3
```

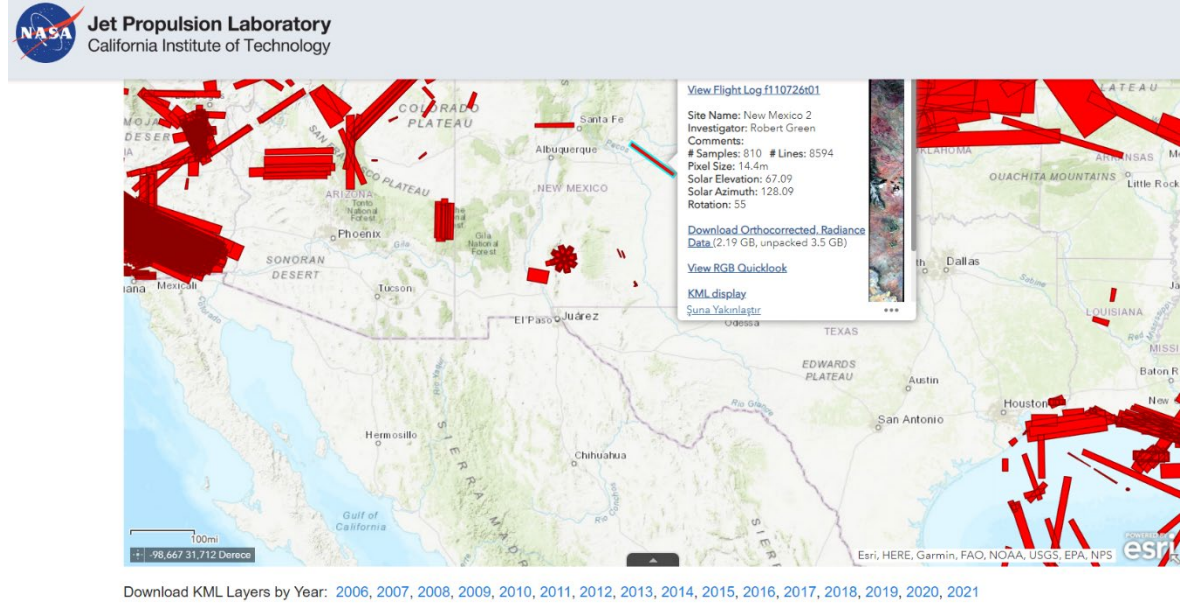
Şekil 6.7. SNMP komutları ile metriklerin alınması

AVIRIS projesinin temel amacı, moleküler absorpsiyon ve partikül saçılma imzalarına dayalı olarak Dünya yüzeyinin ve atmosferinin bileşenlerini belirlemek, ölçmek ve izlemektir. AVIRIS verileriyle yapılan araştırmalar, ağırlıklı olarak küresel çevre ve iklim değişikliği ile ilgili süreçleri anlamaya odaklanır.

AVIRIS sensörü, geometrik olarak uyumlu spektrometrik ölçümlerden Dünya yüzeyinin ve atmosferinin karakterizasyonu için kullanılabilecek verileri toplar. Bu veriler oşinografi, çevre bilimi, kar hidrolojisi, jeoloji, volkanoloji, toprak ve arazi yönetimi, atmosferik ve aerosol çalışmaları, tarım ve limnoloji gibi alanlardaki çalışmalara uygulanabilir. Geliştirme aşamasındaki uygulamalar arasında toksik atık, petrol sızıntıları ve kara/hava/su kirliliği gibi çevresel tehlikelerin değerlendirilmesi ve izlenmesi yer almaktadır. Atmosferik etkiler için uygun kalibrasyon ve düzeltme ile ölçümler, daha sonra yüzey özelliklerinin nicel karakterizasyonu için kullanılabilecek zemin yansıma verilerine dönüştürülebilmektedir. Toplanan tüm veriler, web tarayıcıdan girilebilen AVIRIS veri portalı üzerinden erişilebilmektedir (bkz. Şekil 6.8).

AVIRIS veri portalından büyüklükleri 0,41 GB ile 7,32 GB aralığında değişen ve Çizelge 6.3'te detayları verilen 10 adet görüntü verisi alınarak kamera bilgisayarlarına yüklenmiştir.

Simülasyonun başlamasıyla birlikte, her bir kamera bilgisayarı farklı sıradan başlamak kaydıyla sırayla görüntüleri uygun bir veri merkezine göndermek için merkezi bilgisayara ilgili görüntünün büyüklüğü ile istek gönderir. Simülasyonun sonuna kadar, tüm görüntüler gönderildiyse başlanan görüntü indeksinden görüntüler tekrar gönderilmeye devam eder.



Şekil 6.8. Web uygulaması olarak erişilebilen AVIRIS veri portalı

Çizelge 6.3. AVIRIS veri portalından alınan görüntülerin özellikleri

Name	Site Name	Pixel Size (m)	Number of Lines	Number of Samples	File Size (GB)	Gzip Size (GB)
f060925t01p00r06	Yellowstone NP 1B, WY	14,3	11273	792	5,03	2,87
f061002t01p00r09	Minnesota Spruce Budworm 1, MN	16,8	6969	765	2,73	1,66
f100521t01p00r05	Senator Beck Basin E-W, CO	11,9	13489	1062	7,32	3,51
f100825t01p00r06	Dewar Refill	15,6	3186	693	1,12	0,71
f100829t01p00r08	Texas Coast	17,2	1079	739	0,41	0,22
f110512t01p00r04	Colorado Plateau 6	3,4	1901	797	0,76	0,45
f110512t01p00r09	Colorado Plateau 13	3	3822	985	1,86	1
f110512t01p00r11	Colorado Plateau 9	3,1	8450	885	3,74	2,2
f110523t01p00r16	WSMR Cloverleaf 1-2	16	2941	1829	2,56	0,85
f180820t01p00r07	Open Ocean, off southern Oregon	17,9	5783	1071	3,08	1,61

6.3. Deneysel Sonuçlar

Önerilen model (PAA) geliştirilirken kıyaslama yapabilmek adına 5.3'te anlatılan RA, HA ve HDFS yöntemleri de gerçekleştirilmiştir. Çizelge 6.4'te gerçekleştirilen yöntemlere ait işlem süreleri verilmiştir. RA ve HA yöntemleri, verilen ağlar için 0-100 milisaniye aralığında sonuç üretebilirken, önerilen PAA yöntemi ilk aşamada sonuçlanması durumunda 100-200 milisaniye aralığında sonuç üretmektedir. Eğer kısıtların karşılanamamasından dolayı PAA'nın ikinci aşaması çalışırsa, toplamda 500-700 milisaniye aralığında sonuç üretilir. Böylece önerilen iki aşamalı model, çözümü 1 saniyeden daha kısa bir sürede bulurken aynı zamanda kısıtları da göz önünde bulundurarak kaynak optimizasyonu yapabilmektedir. Diğer yandan, HDFS her dakika çalıştırdığı senkronizasyon algoritması için 1500-2700 milisaniye harcamaktadır.

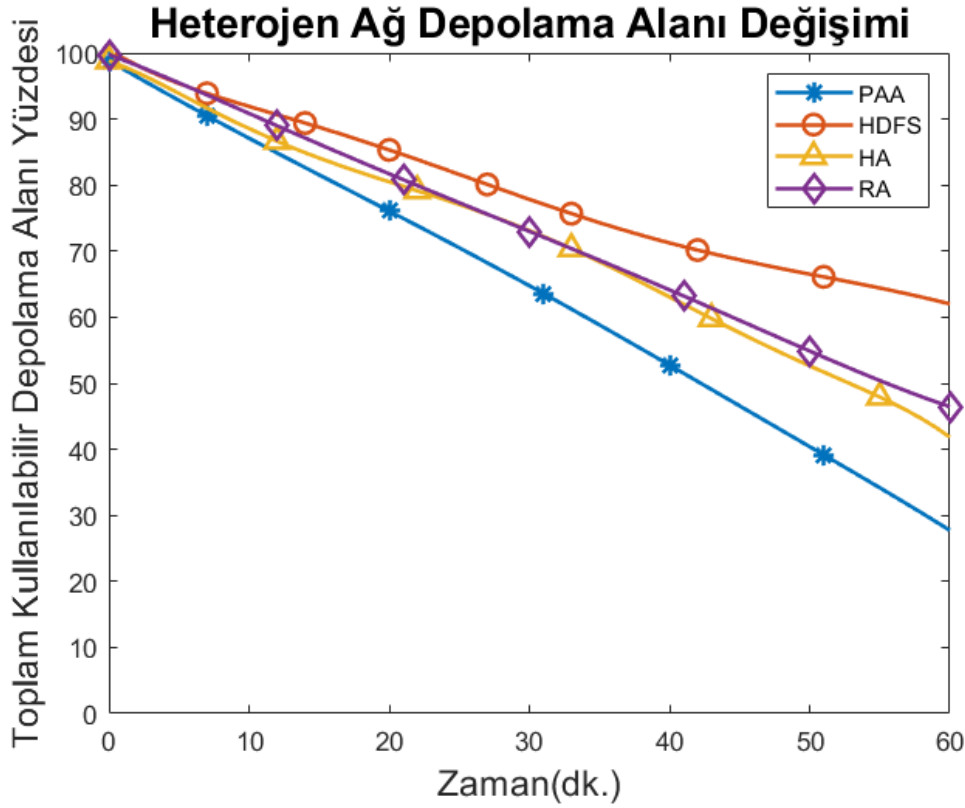
Çizelge 6.4. Kullanılan yöntemler için işlem süreleri

Yöntem	İşlem Süresi (ms)
RA	0-10
HA	10-100
PAA (Faz 1)	100-200
PAA (Faz 1+Faz 2)	500-700
HDFS	1500-2700

Bu bölümde PAA, RA, HA ve HDFS yöntemleri için kaynak kullanımlarının zamana göre değişimleri, veri merkezi bazında verilmiştir. Burada simülasyonların ilk bir saati ve PAA yönteminin yanında diğer yöntemlerin kaynak tüketimi sunulmuştur. İlk bir saatin kullanılmasının nedeni, bazı veri merkezlerinde bir saatin sonunda 10 GB'ın altında kullanılabilir depolama alanı kalması ve işletim sisteminin 10 GB'dan daha az depolama alanıyla iyi çalışmıyor olmasıdır. Elde edilen sonuçlardaki anlık salınımları (oscillation) yumuşatmak amacıyla grafiklerde 5 dakikalık hareketli ortalama kullanılmıştır. Böylece ortalama kaynak kullanımları grafiklerde yüzde 80 ve 90 arasında görüldüğünde, gerçekleşen anlık kullanım değerleri yüzde 100 değerine yaklaşmaktadır. Bu durumun görüldüğü veri merkezinde ilgili kaynak için verilen zaman aralığında darboğaz oluşmuş demektir.

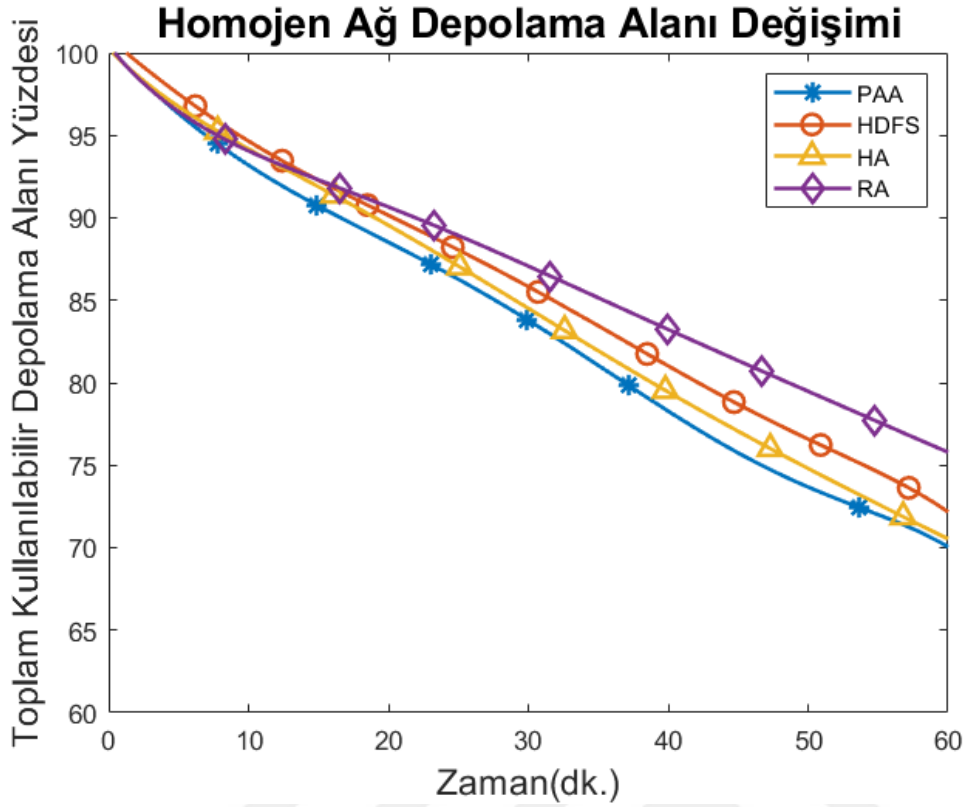
Şekil 6.9 RA, HA, HDFS ve PAA için heterojen ağdaki toplam depolama değişikliklerini gösterir. PAA genel olarak daha iyi depolama performansı üretir; bir saatlik simülasyonda

346 GB depolama alanını doldurmayı başarır ve bu da sırasıyla RA, HA ve HDFS'den 36%, 26% ve 71% daha iyi performans anlamına gelir (bkz. Çizelge 6.5).



Şekil 6.9. Heterojen ağda toplam kullanılabilir depolama alanının zamana göre değişimi

Şekil 6.10 RA, HA, HDFS ve PAA için homojen ağdaki toplam depolama değişikliklerini gösterir. Heterojen ağ ile karşılaştırıldığında, PAA sırasıyla RA, HA ve HDFS'den 25%, 1% ve 12% daha iyi performansa sahip olmasına rağmen tüm yöntemler yakın performans üretir (bkz. Çizelge 6.5). Ağı homojen hale getirmek için, veri merkezleri yönlendirici 5'e bağlanır. Böylece, kamera bilgisayarları ve veri merkezleri arasındaki bant genişlikleri ve gecikme süreleri, heterojen ağa göre daha yakındır. Bu durum farklı yöntemlerde daha yakın depolanan veri miktarına neden olur. Öte yandan, yönlendirici 5'te bir darboğaz olasılığı artar ve bu da homojen ağ için daha az veri depolanmasına neden olur. Bu karşılaştırmanın bir diğer değerli konusu, HDFS'in homojen ağdaki performansının heterojen ağdan çok daha iyi olmasıdır. Bunun nedeni, HDFS uygulamasının homojen bir ağ yapısı için tasarlanmış olmasıdır. Kamera bilgisayarları ve veri merkezleri arasındaki ağ yapısı doğal olarak değişken olduğundan, tezin geri kalanında analizler heterojen ağ üzerinde detaylandırılmıştır.



Şekil 6.10. Homojen ağda toplam kullanılabilir depolama alanının zamana göre değişimi

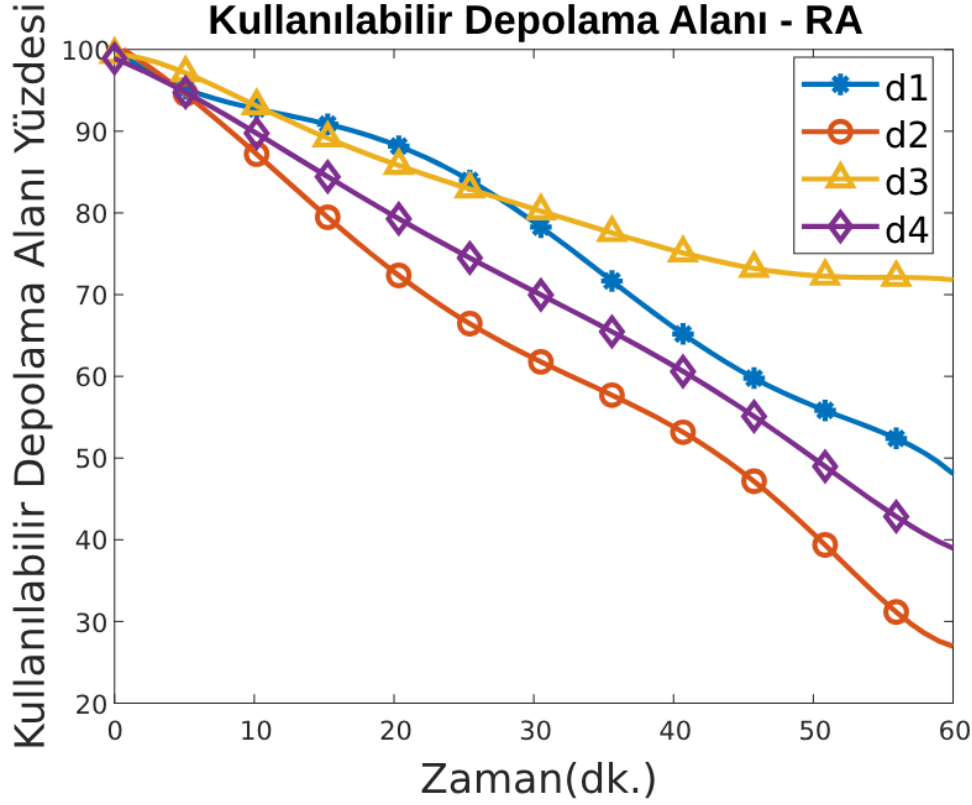
Çizelge 6.5. Kullanılan yöntemler için toplam depolanan veri miktarları

Yöntem	Heterojen Ağ (GB)	Homojen Ağ (GB)
PAA	346	134
HDFS	202	119
HA	274	132
RA	254	107

6.3.1. Kullanılabilir depolama miktarlarının zamana göre değişimi

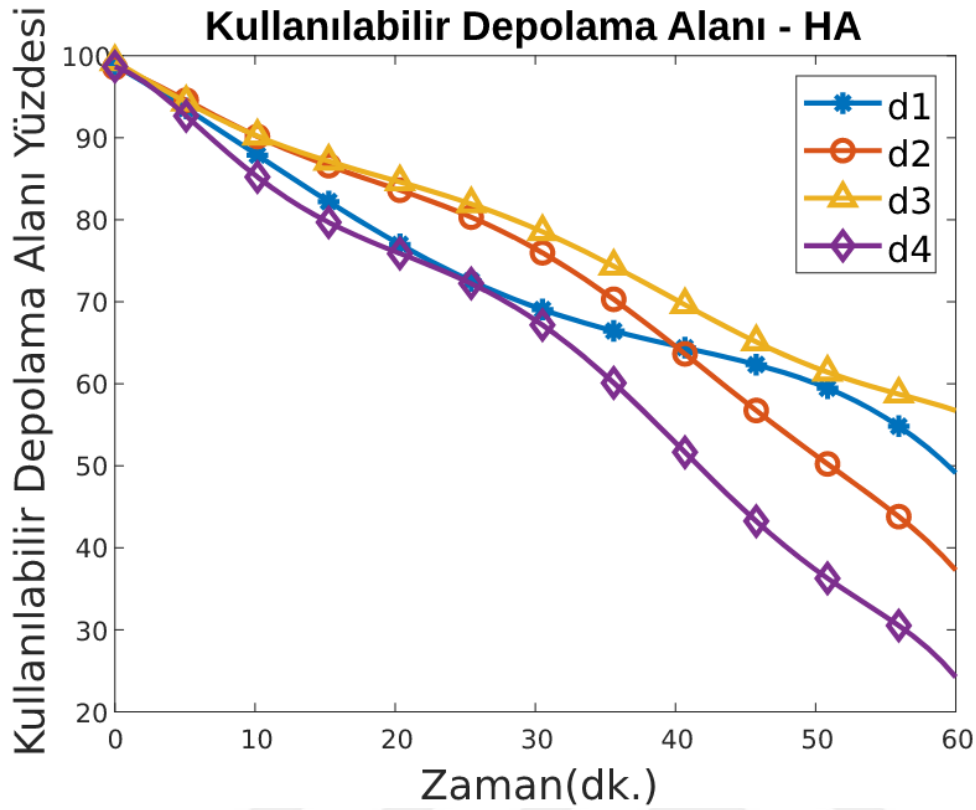
RA yöntemi için veri merkezlerindeki kullanılabilir depolama miktarının zamana göre değişimi Şekil 6.11’de verilmiştir. Burada görüldüğü üzere veri merkezi d3, 30. dakikadan sonra diğer veri merkezlerine oranla daha düşük bir veri depolama miktarına sahiptir. Veri merkezi d3 için diğer kaynak kullanımı miktarları incelendiğinde, bellek kullanımının (bkz. Şekil 6.19) diğer veri merkezlerinden daha yüksek olduğu görülmekle birlikte 30-60 dakika aralığında tepe (peak) yaptığı görülmektedir. Bunun nedenlerinden biri, topolojik olarak veri merkezi d3 ve d4’e olan ortalama kamera bilgisayarları uzaklıklarının d1 ve d2’ye göre daha fazla olmasıdır. Diğerleri ise rastgele atama yapıldığı için d3’e uzak olan kamera

bilgisayarlarının atanması sonucu oluşan paket hataları ve ağ gecikmeleri neticesinde 30. dakikadan sonra veri transferlerinin gecikerek bellekte tükenmeye yol açmasıdır.



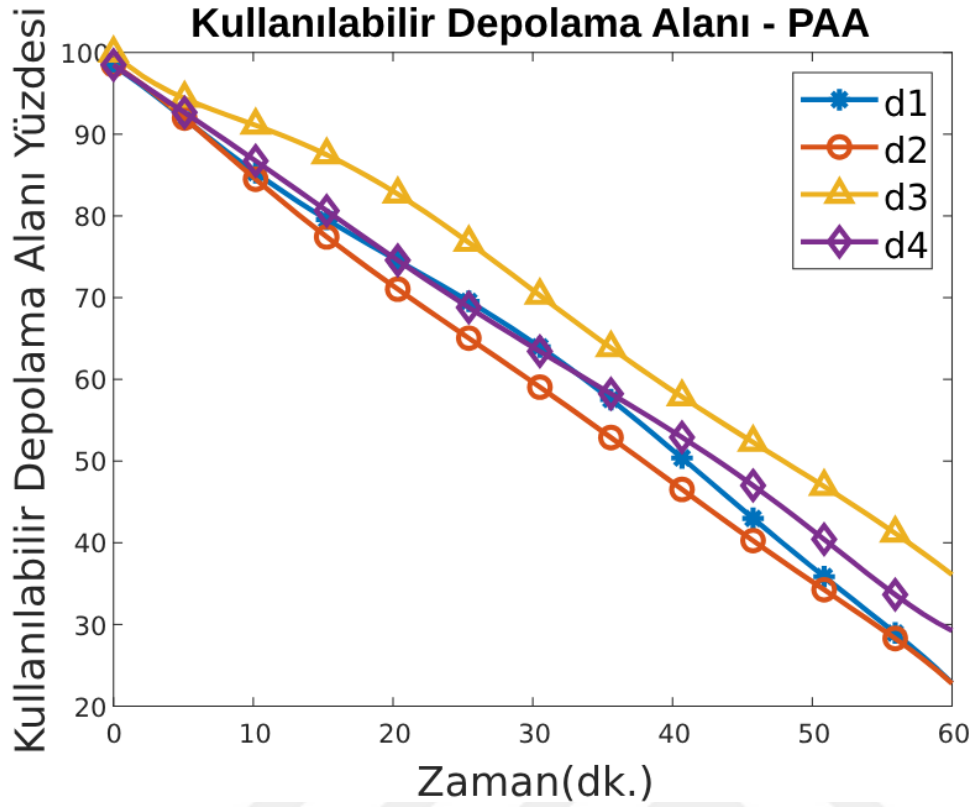
Şekil 6.11. RA yöntemi için veri merkezlerindeki kullanılabilir depolama miktarının zamana göre değişimi

HA yöntemi için veri merkezlerindeki kullanılabilir depolama miktarının zamana göre değişimi Şekil 6.12’de verilmiştir. Burada görüldüğü üzere veri merkezi d3, yine diğer veri merkezlerine oranla daha düşük bir veri depolama miktarına sahiptir; fakat RA’dakinden daha fazla veri depolayabilmiştir. Veri merkezi d3 için bellek kullanımı (bkz. Şekil 6.20) ve IOPS miktar (bkz. Şekil 6.24), diğer veri merkezlerine oranla daha yüksek seyretmiştir. Veri merkezi d1 ise, 30. dakikadan sonra veri depolama miktarında bir düşüş yaşamıştır. Bu durum, 30. dakikaya kadar yüksek seyretmekte olan işlemci kullanımı ve IOPS miktarının düşmesiyle birlikte bellek kullanımının artması neticesinde oluşmuştur. Bunun ana nedeni ise, ne kadar kaynakların mevcut maliyetini düşürmeye çalışsa da HA yönteminin veri merkezlerinin kapasite ve kısıtlarından habersiz olmasıdır.



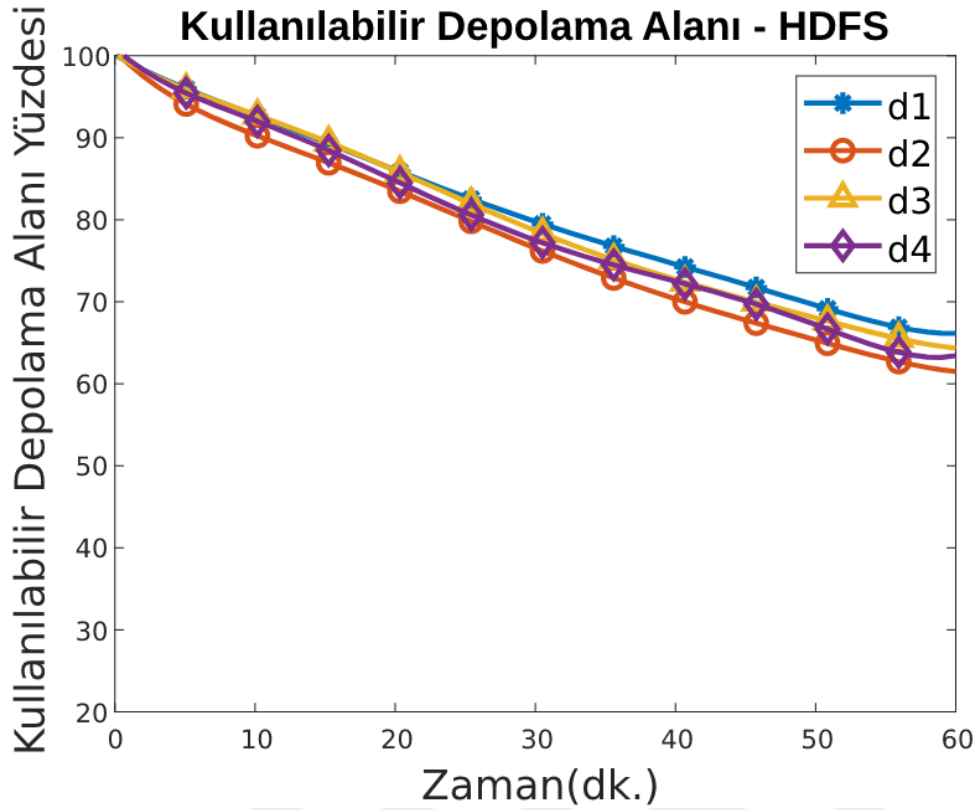
Şekil 6.12. HA yöntemi için veri merkezlerindeki kullanılabilir depolama miktarının zamana göre değişimi

PAA yöntemi için veri merkezlerindeki kullanılabilir depolama miktarının zamana göre değişimi Şekil 6.13'te verilmiştir. Diğer yöntemlere benzer bir şekilde veri merkezi d3, 10. dakikada daha az veri depolamaya başlamıştır. Bunun nedeni 10. dakikaya doğru d3'teki bellek kullanımının (bkz. Şekil 6.21) diğer veri merkezlerine oranla artmasıdır. PAA, kaynak kullanımını dikkate alarak atama yaptığından veri merkezi d3'teki azalan veri depolama miktarı artarak 20. dakikadan itibaren diğer veri merkezlerine yaklaşmıştır. Diğer yandan veri merkezi d1 ve d2'de diğerlerine oranla daha fazla veri depolanmasının nedeni, bu merkezlerin topolojik olarak ağın daha ortalarında olmasından dolayı kamera bilgisayarlarına olan ortalama uzaklıklarının daha düşük olmasıdır. Ayrıca PAA, genel toplamda RA, HA ve HDFS'e göre daha iyi bir depolama performansı ortaya koymuştur. Bir saatlik simülasyonda 346 GB veriyi depolamayı başarmakla birlikte bu, RA, HA ve HDFS'ten sırasıyla %36, %26 ve %71 daha iyi performans anlamına gelmektedir.



Şekil 6.13. PAA yöntemi için veri merkezlerindeki kullanılabilir depolama miktarının zamana göre değişimi

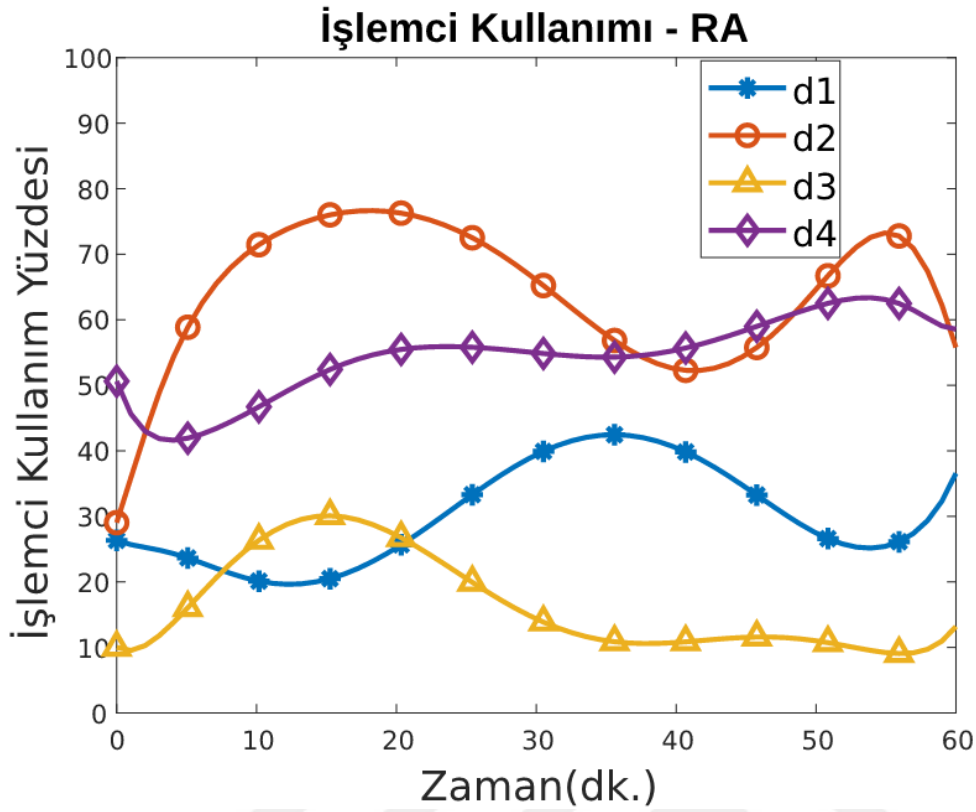
HDFS yöntemi için veri merkezlerindeki kullanılabilir depolama miktarının zamana göre değişimi Şekil 6.14’te verilmiştir. Diğer yöntemlerden farklı bir şekilde veri merkezlerinin tümünde veri depolama miktarları birbirlerine çok yakın olmuştur. Bunun nedeni HDFS’in bünyesinde bulundurduğu dengeleyici (balancer) olarak adlandırılan araçtır. Bu araç ile birlikte Hadoop, tüm DataNode’ların kabaca eşit miktarda veri depolaması için bir kümenin blok dağıtımını yeniden dengelemeye olanak tanır. NameNode DataNode’lara veri bloklarını ayırırken, hangi DataNode’ların yeni blokları alacağını belirlemek için kümenin DataNode’ları arasındaki veri dağılımının tekdüze dağılımlı olmasını dikkate alır. Bu noktada dağılım için kullanılan tek bilgi depolama kapasiteleri olmakla birlikte yeniden dengeleme şeması, bir DataNode üzerindeki boş depolama alanı belirli bir eşiğin altına düşerse verileri otomatik olarak bir DataNode’dan diğerine taşıyabilir. HDFS’in heterojen bir ağda çalıştırılması ve dengelemenin oluşturduğu fazladan veri taşıma maliyeti, HDFS’in diğer yöntemlere göre daha az veri depolama miktarına sahip olmasının nedenleridir.



Şekil 6.14. HDFS yöntemi için veri merkezlerindeki kullanılabilir depolama miktarının zamana göre değişimi

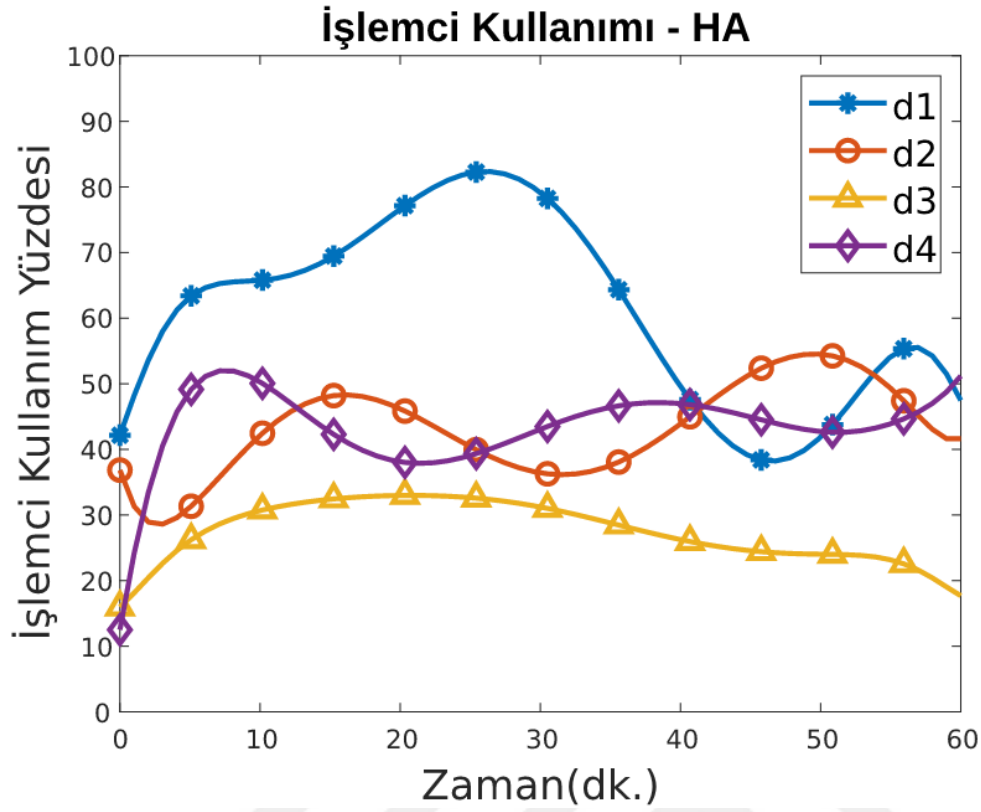
6.3.2. İşlemci kullanımlarının zamana göre değişimi

RA yöntemi için veri merkezlerindeki işlemci kullanımlarının zamana göre değişimi Şekil 6.15’de verilmiştir. Veri merkezi d2, işlemci kullanımı bakımından en yoğun veri merkezi olmakla birlikte en çok veri depolama miktarına da sahiptir. Diğer yandan d2 için bellek miktarı 30%’u aşmamakla birlikte IOPS miktarı işlemci kullanımı ile doğru orantılı bir şekilde artmıştır. İkinci en çok veri depolama miktarına sahip olan veri merkezi d4 için de simülasyon boyunca kaynak kullanım miktarları benzer şekilde seyretmiştir. En az veri depolama miktarına sahip olan veri merkezi d3’te 10. dakikaya kadar işlemci kullanımı ve IOPS miktarı artmıştır fakat sonrasında bellek kullanımının artmasıyla işlemci kullanımı 10%’a gerilemiştir. Son olarak veri merkezi d1 için işlemci kullanımı, 20% ile 40% aralığında seyrederken kullanılan bellek miktarının 30. dakikaya doğru düşmesiyle IOPS miktarında artış gerçekleşmiştir.



Şekil 6.15. RA yöntemi için veri merkezlerindeki işlemci kullanımları

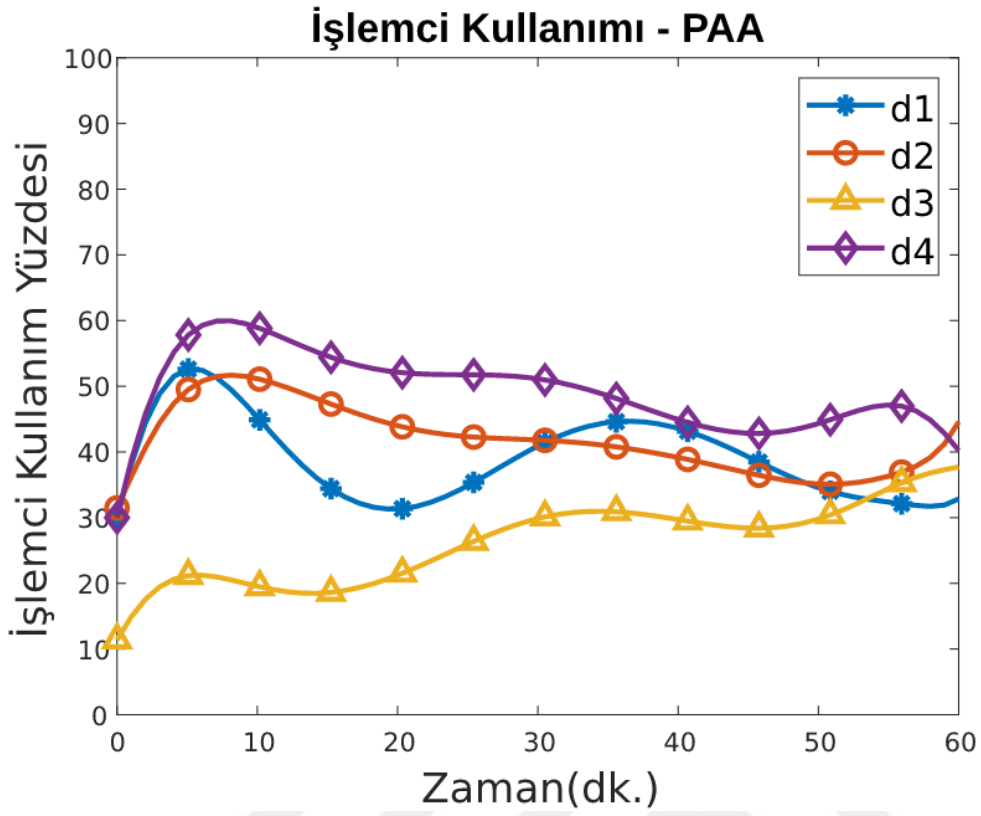
HA yöntemi için veri merkezlerindeki işlemci kullanımlarının zamana göre değişimi Şekil 6.16'da verilmiştir. Veri merkezi d1, işlemci kullanımı bakımından en yoğun veri merkezi olmakla birlikte 30. dakikadan sonra bellek kullanımının artmasıyla hem işlemci kullanımı hem de IOPS miktarında düşüş gerçekleşmiştir. Böylece d1 için veri depolama miktarında düşüş yaşanmıştır. Veri merkezi d2 ve d4 için işlemci kullanımları 30% ile 50% aralığında değişmiştir ve d4 en fazla veri depolama miktarına sahiptir. Veri merkezi d3 için işlemci kullanımı 30% üzerine hiç çıkmamakla birlikte bunun nedeni bellek miktarının 50% ve üzerinde seyretniş olmasıdır. Bu durum d3 için veri depolama miktarının diğerlerine oranla düşük kalmasına neden olmuştur.



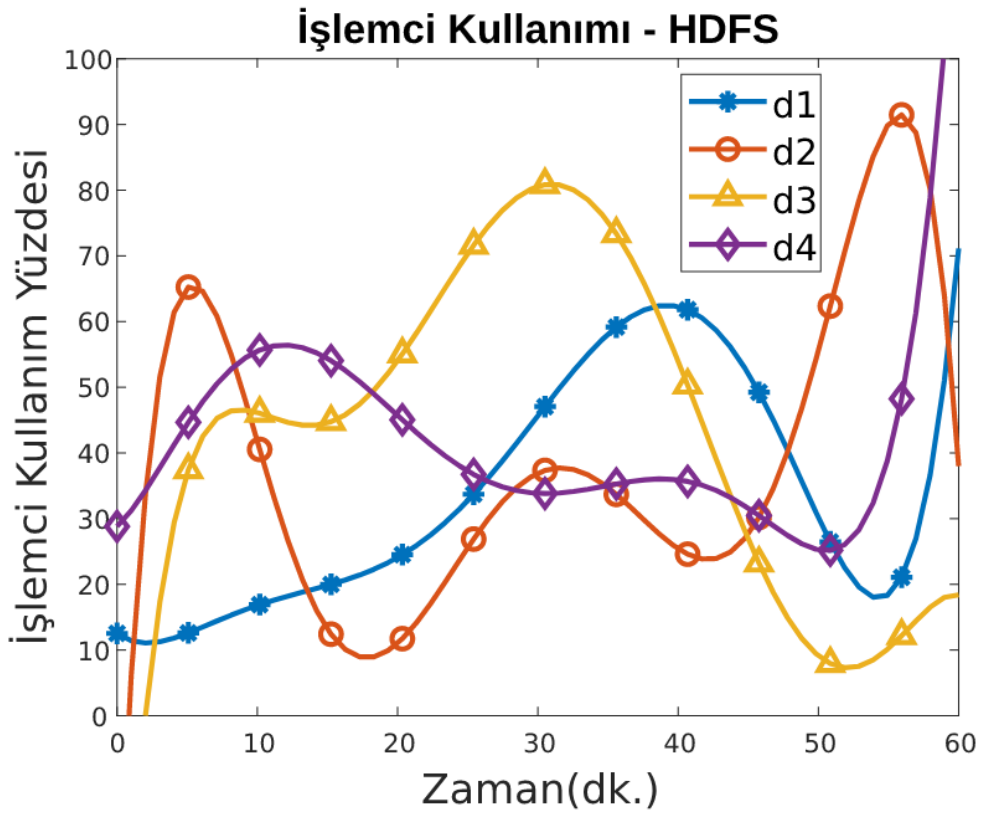
Şekil 6.16. HA yöntemi için veri merkezlerindeki işlemci kullanımları

PAA yöntemi için veri merkezlerindeki işlemci kullanımlarının zamana göre değişimi Şekil 6.17’de verilmiştir. Veri merkezi d1, d2 ve d4 için işlemci kullanımları 10. dakikaya kadar 50%-60% seviyelerine çıkmakla birlikte bu dakikadan sonra işlemci kullanımları 30% civarına düşmektedir. Veri merkezi d3 için ise işlemci kullanımı 10% ile başlayıp ilerleyen dakikalarda 20%-30% seviyelerine çıkarılmıştır. Böylece önerilen PAA yöntemi, veri merkezlerinin işlemci kullanımlarını dengeleyerek herhangi birinde oluşabilecek yoğun kullanımı engellemiştir.

HDFS yöntemi için veri merkezlerindeki işlemci kullanımlarının zamana göre değişimi Şekil 6.18’de verilmiştir. Veri merkezi d2 için işlemci kullanımları 10. ile 50. Dakika aralığında 10%-35% seviyelerinde olmakla birlikte diğer veri merkezlerinde bu değer 20%-80% aralığında gerçekleşmiştir. Aralarında az farklar olsa da veri merkezi d2’de daha fazla veri toplanmasının nedeni işlemci yükünün daha az olmasıdır. Diğer yandan bu yöntemde işlemci yükü için alınan anlık salınımlar diğer yöntemlere göre fazla olmuştur.



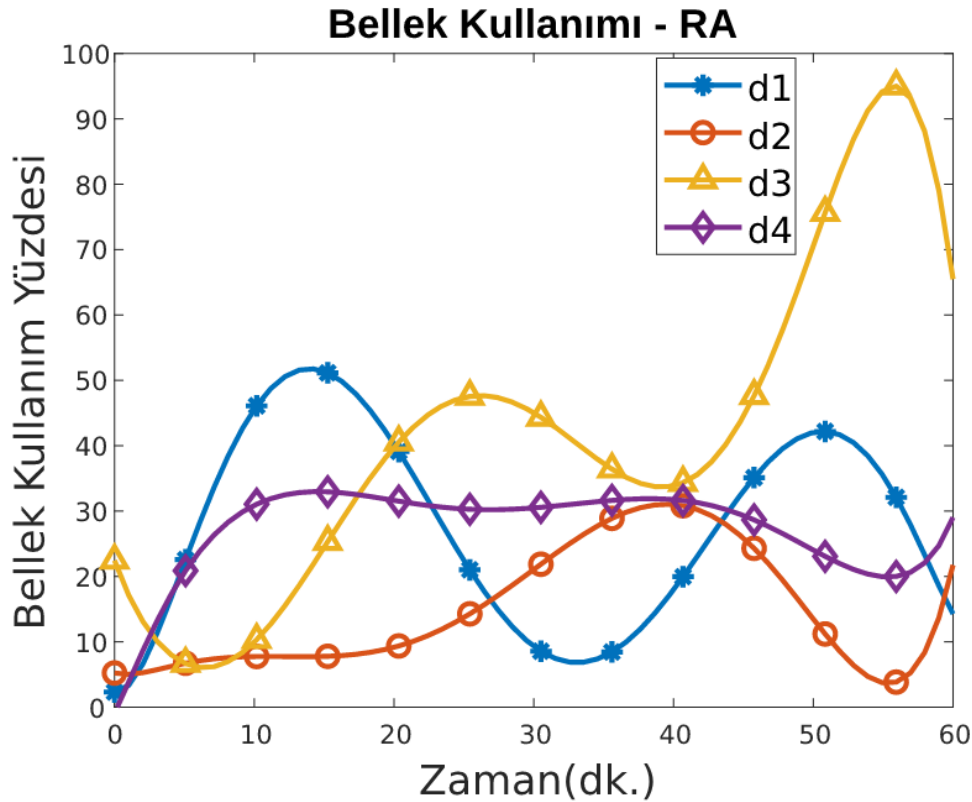
Şekil 6.17. PAA yöntemi için veri merkezlerindeki işlemci kullanımları



Şekil 6.18. HDFS yöntemi için veri merkezlerindeki işlemci kullanımları

6.3.3. Bellek kullanımlarının zamana göre değişimi

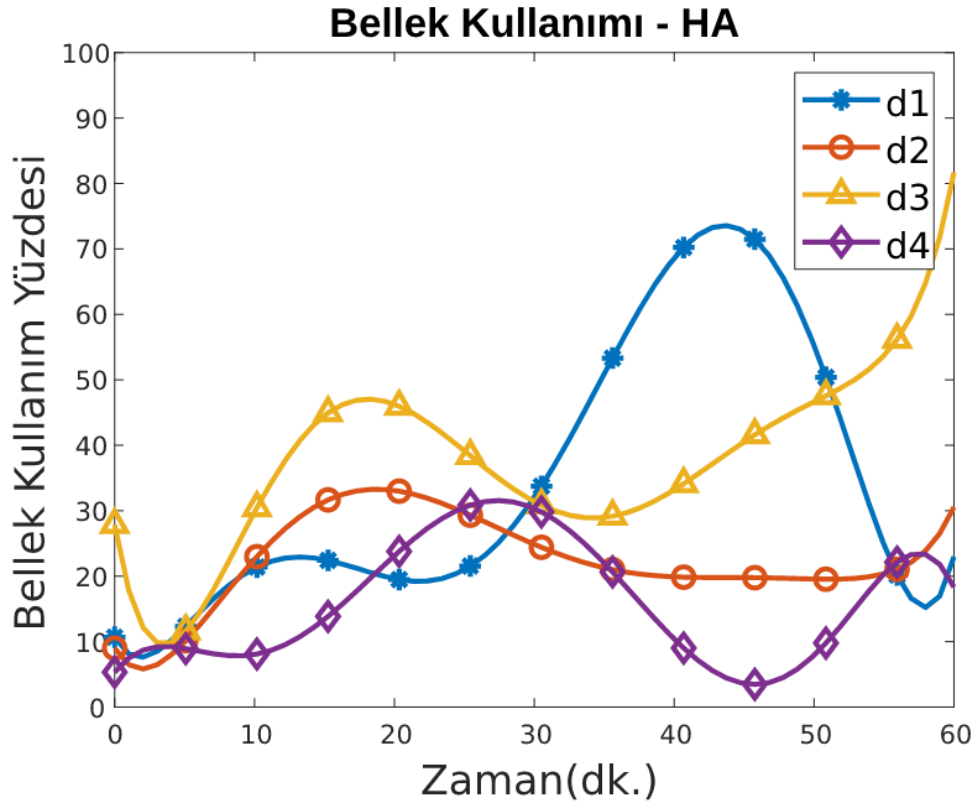
RA yöntemi için veri merkezlerindeki bellek kullanımlarının zamana göre değişimi Şekil 6.19'da verilmiştir. Veri merkezi d2 ve d4 için bellek kullanımları 10%-30% aralığında değişmektedir. En az bellek kullanımı d2'de gerçekleşmekle beraber işlemci kullanımı ve IOPS miktarı bu veri merkezi için diğerlerine göre daha yüksektir. Böylece veri depolama miktarı en fazla d2'de olmuştur. Veri merkezi d1 ve d3 için bellek kullanımları 10%-50% aralığında değişirken 40. dakikadan sonra d3'te bellek kullanımı 90%'lara varmıştır. Bu dakikadan sonra veri merkezi d3'te bellek tükenme noktasına geldiği için veri depolama miktarı diğer veri merkezlerine göre düşük kalmıştır.



Şekil 6.19. RA yöntemi için veri merkezlerindeki bellek kullanımları

HA yöntemi için veri merkezlerindeki bellek kullanımlarının zamana göre değişimi Şekil 6.20'de verilmiştir. Veri merkezi d2 ve d4 için bellek kullanımları, RA'da olduğu gibi 10%-30% aralığında değişmektedir. En az bellek kullanımı d4'de gerçekleşmekle beraber veri depolama miktarı en fazla olan veri merkezi de d4 olmuştur. Diğer yandan RA'da olduğu gibi en fazla depolamaya sahip veri merkezi için işlemci kullanımı ve IOPS en fazla

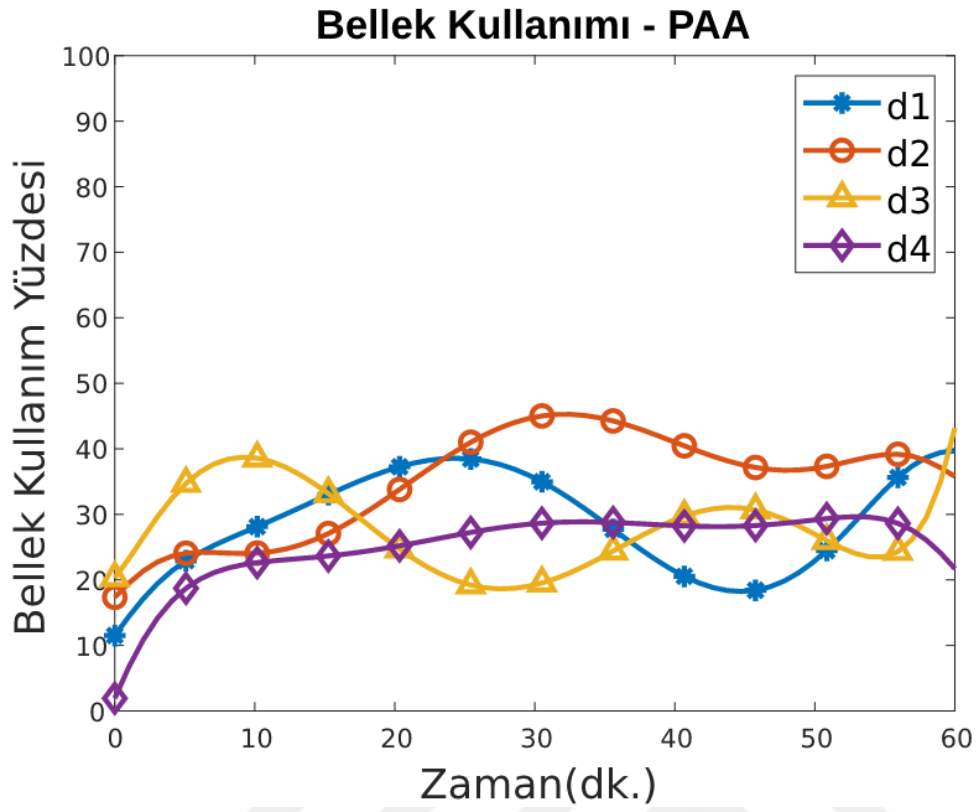
olmamıştır. Veri merkezi d1 ve d3 için bellek kullanımları, 10%-50% aralığında değişirken 35. dakikadan sonra 60%-80% aralığını görmüştür. Bu dakikadan sonra veri merkezi d1’de ani bir bellek kullanımı artışı neticesinde veri depolama oranı diğer veri merkezlerine göre düşük kalmıştır. Genel toplamda ise veri merkezi d3 en az veri depolama miktarına sahip olmuştur.



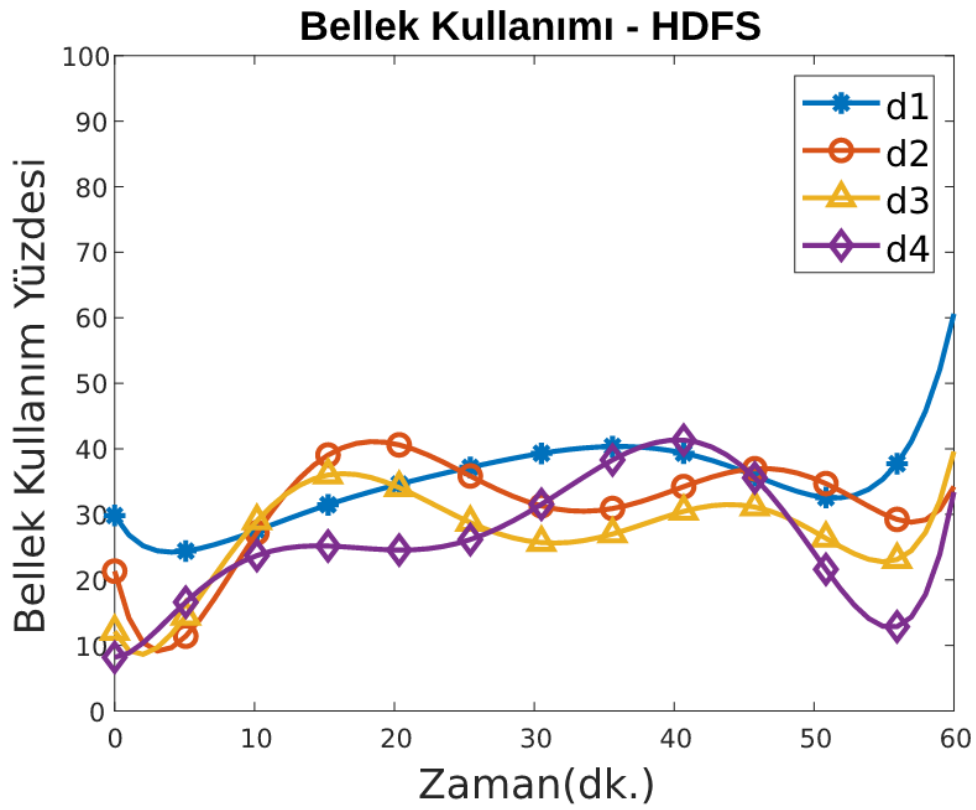
Şekil 6.20. HA yöntemi için veri merkezlerindeki bellek kullanımları

PAA yöntemi için veri merkezlerindeki bellek kullanımlarının zamana göre değişimi Şekil 6.21’de verilmiştir. Tüm veri merkezleri için bellek kullanımları 15%-45% aralığında gerçekleşmiştir. Önerilen yöntem bellek kullanımını dengelediği için diğer yöntemlerde ortaya çıkan ani bellek kullanımı artışı ve buna bağlı olarak veri depolama oranındaki ani düşüşler ortaya çıkmamıştır.

HDFS yöntemi için veri merkezlerindeki bellek kullanımlarının zamana göre değişimi Şekil 6.22’de verilmiştir. Tüm veri merkezleri için bellek kullanımları 10%-40% aralığında gerçekleşmekle birlikte kötü performansın nedeni olmadığı görülmüştür.



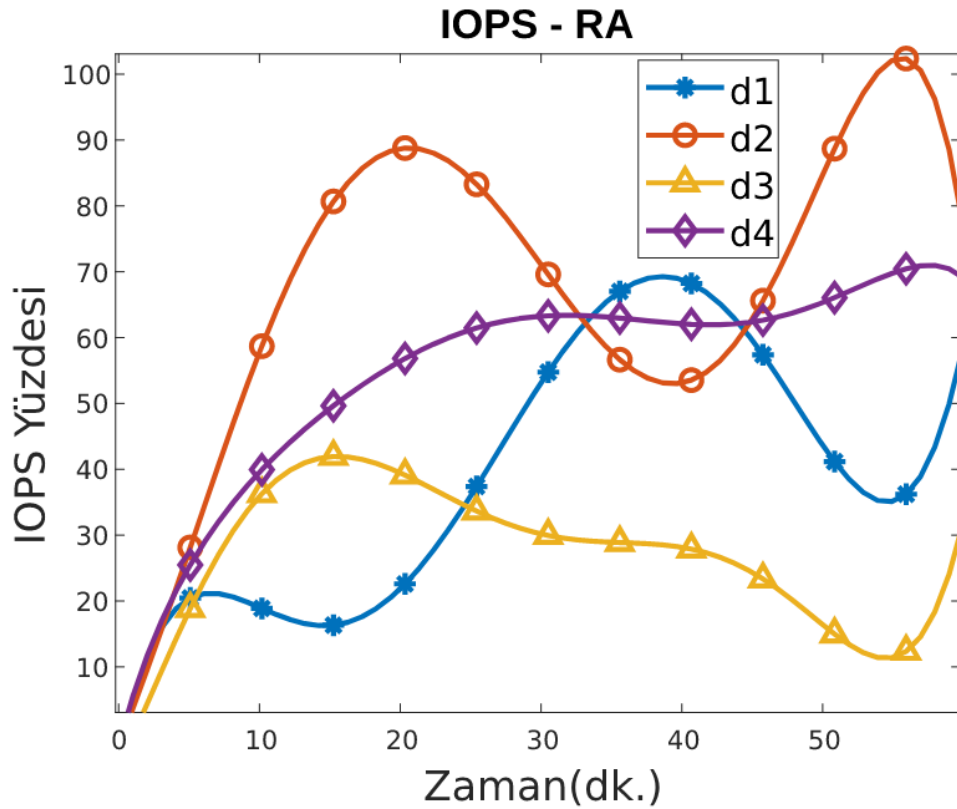
Şekil 6.21. PAA yöntemi için veri merkezlerindeki bellek kullanımları



Şekil 6.22. HDFS yöntemi için veri merkezlerindeki bellek kullanımları

6.3.4. IOPS miktarlarının zamana göre değişimi

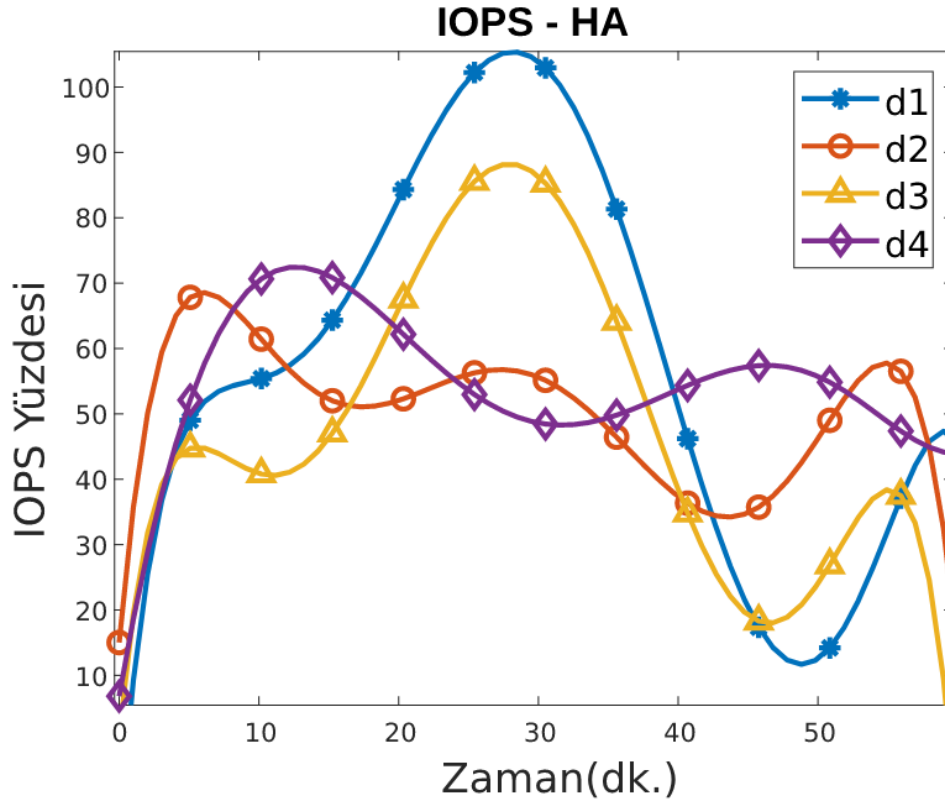
RA yöntemi için veri merkezlerindeki IOPS miktarlarının zamana göre değişimi Şekil 6.23'de verilmiştir. Veri merkezi d2 için IOPS miktarı 10. dakikadan sonra 50%-100% aralığında değişmektedir. En az bellek kullanımı d2'de gerçekleşmekle beraber işlemci kullanımı IOPS miktarıyla doğru orantılı olarak artış göstermiştir. Böylece veri depolama miktarı en fazla d2'de olmuştur. Veri merkezi d1 ve d4 için IOPS miktarları 40%-60% aralığında seyretmiştir. Veri merkezi d3'te IOPS miktarı 15. dakikadan sonra düşmeye başlamıştır. Bunun nedeni bellek kullanımının IOPS miktarı ile ters orantılı olarak artmasıdır ve böylece veri depolama miktarı d3'te diğer veri merkezlerine göre düşük kalmıştır.



Şekil 6.23. RA yöntemi için veri merkezlerindeki IOPS miktarları

HA yöntemi için veri merkezlerindeki IOPS miktarlarının zamana göre değişimi Şekil 6.24'te verilmiştir. Veri merkezi d2 ve d4 için IOPS miktarları 40%-70% aralığında değişmektedir. Veri merkezi d1 ve d3 içinse 35. dakikaya kadar IOPS miktarları, 60%-100% aralığında değişirken simülasyonun kalan bölümünde bellek kullanımının artmasıyla 10%-50% aralığına gerilemiştir. Bu durumda bu veri merkezlerinde yaşanan ani IOPS miktarı

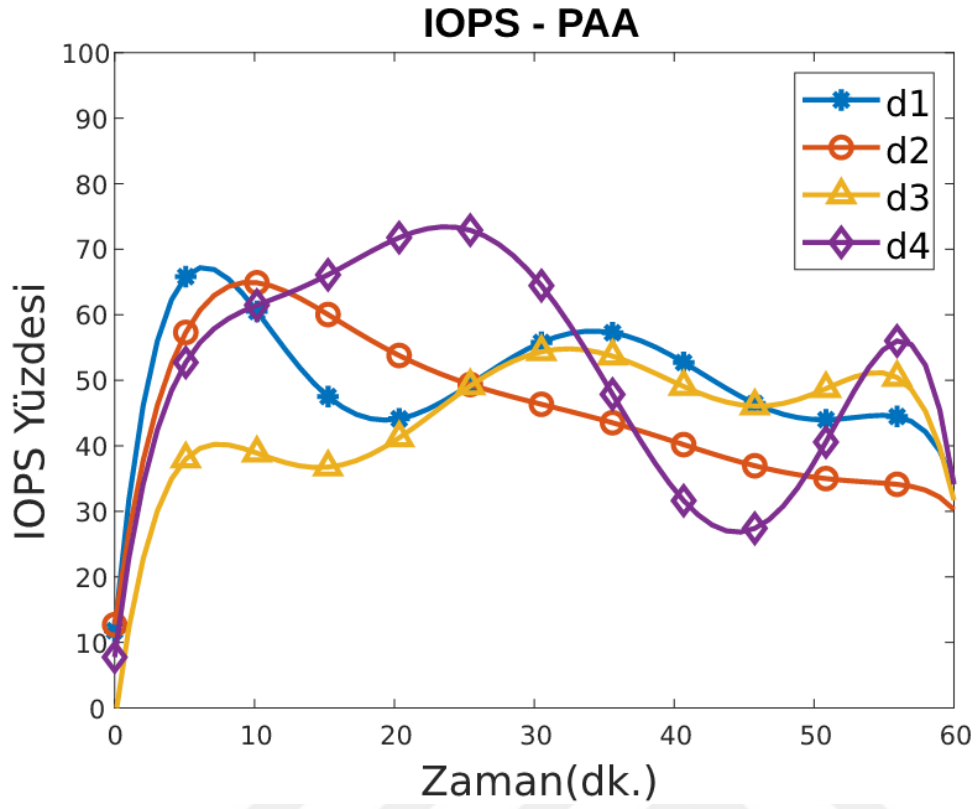
düşüşleri neticesinde depolanan veri miktarlarında ciddi oranda azalma olmuştur. IOPS miktarlarında yavaş azalmaların görüldüğü veri merkezi d2 ve d4 ise diğerlerine göre daha fazla veri depolamayı başarmıştır.



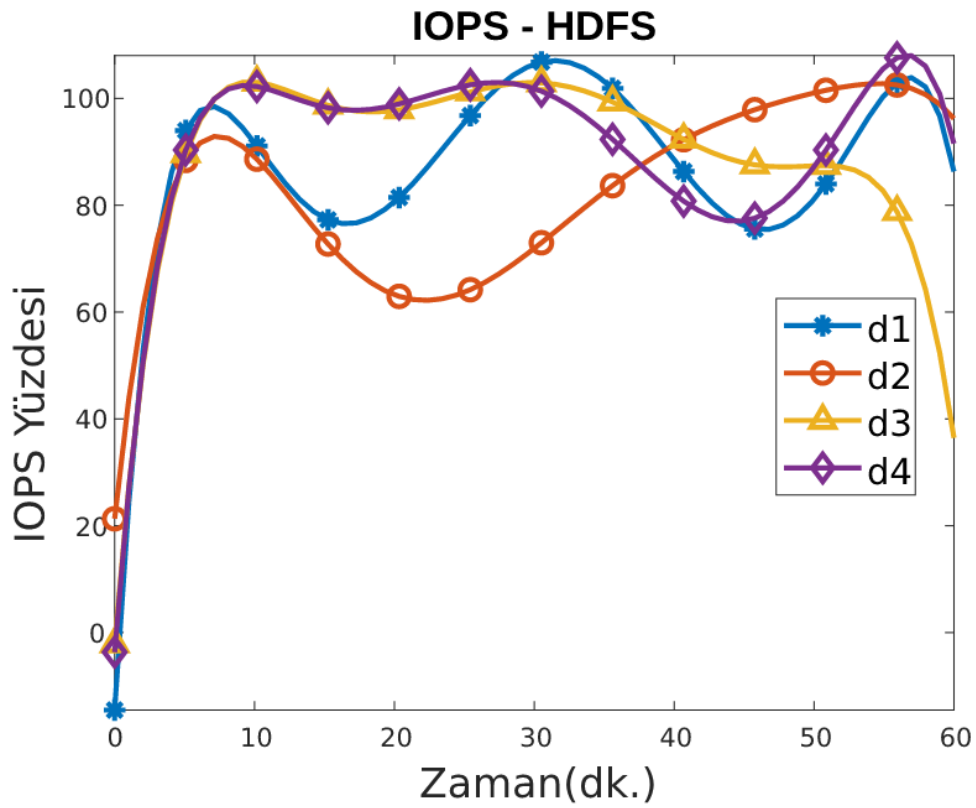
Şekil 6.24. HA yöntemi için veri merkezlerindeki IOPS miktarları

PAA yöntemi için veri merkezlerindeki IOPS miktarlarının zamana göre değişimi Şekil 6.25'te verilmiştir. Tüm veri merkezleri için IOPS miktarları 30%-70% aralığında gerçekleşmiştir. Önerilen yöntem IOPS yükünü dengelediği için diğer yöntemlerde ortaya çıkan ani IOPS miktarı artışı veya düşüşüne bağlı olarak veri depolama oranındaki ani düşüşler engellenmiştir.

HDFS yöntemi için veri merkezlerindeki IOPS miktarlarının zamana göre değişimi Şekil 6.26'da verilmiştir. Tüm veri merkezleri için IOPS miktarları 60%-100% aralığında gerçekleşmiştir. Bu yöntemde IOPS miktarı diğer yöntemlere oranla daha yüksek seviyelerde gerçekleşmiştir. RA ve HA yöntemlerinde olduğu gibi HDFS'te de IOPS yükünün 100% oranına yaklaştığı durumlarda veri merkezinde depolanan veri miktarlarında düşüş olduğu gözlemlenmiştir.



Şekil 6.25. PAA yöntemi için veri merkezlerindeki IOPS miktarları



Şekil 6.26. HDFS yöntemi için veri merkezlerindeki IOPS miktarları

6.3.5. Veri merkezlerinin kaynak durumlarının birlikte değerlendirilmesi

Veri merkezlerindeki kaynakların zaman içindeki değişimi RA, HA, HDFS ve PAA için ayrı ayrı olarak Şekil 6.27, Şekil 6.28, Şekil 6.29 ve Şekil 6.30'da gösterilmektedir. Üç yönteme veri merkezlerini baz alarak baktığımızda, özellikle üçüncü veri merkezi (d3), RA ve HA'da daha az veri aktarım hızına sahipken PAA ile daha iyi bir performans ortaya koymuştur. Ayrıca PAA genel olarak daha iyi depolama performansı sağlar; bir saatlik simülasyonda 346 GB depolama miktarına erişmiştir ve bu, sırasıyla RA, HA ve HDFS'ten %36, %26 ve %71 daha iyi performans anlamına gelir.

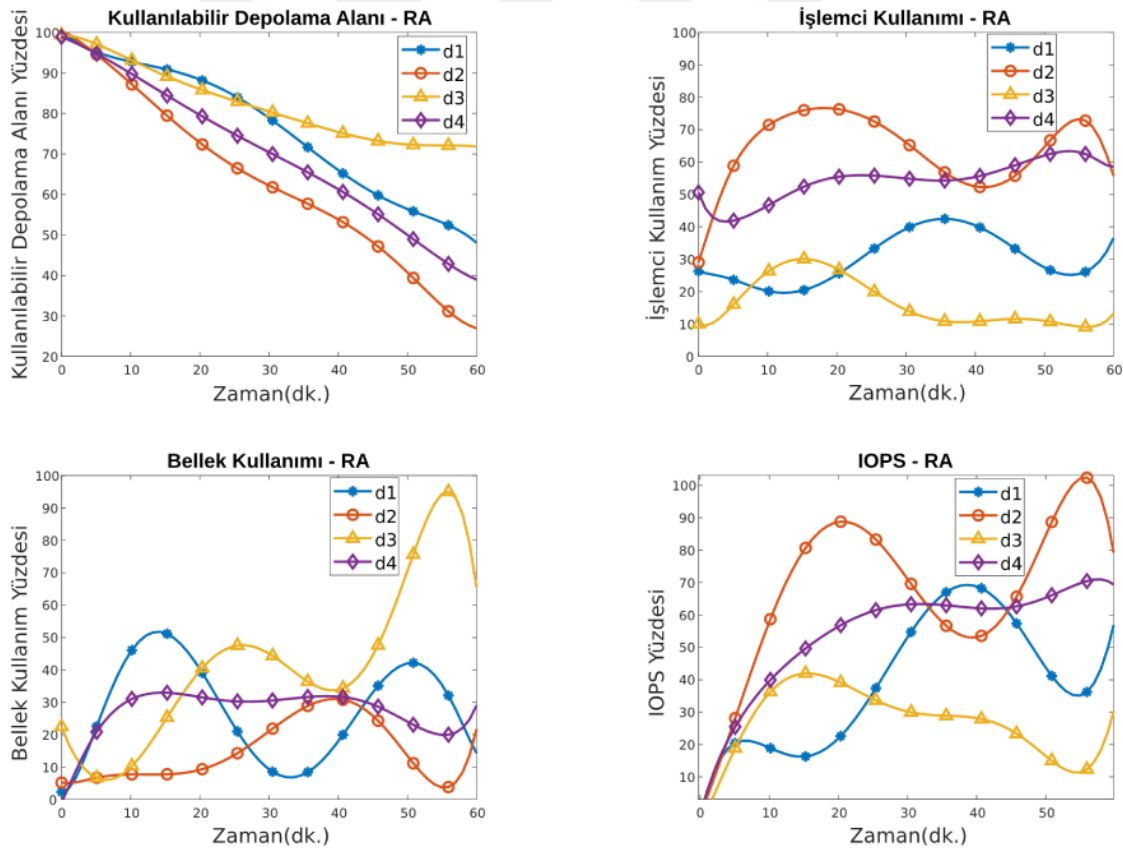
RA yönteminde veri depolama miktarları, çoktan aza olacak şekilde sırayla d2, d4, d1 ve d3 olarak gerçekleşmiştir (bkz. Şekil 6.27). RA için tüm kaynak kullanımları her bir veri merkezi için ele alındığında veri depolama miktarının işlemci kullanımı ve IOPS miktarı ile doğru orantılı, bellek kullanımı ile ters orantılı arttığı görülmektedir.

HA yönteminde veri depolama miktarları, çoktan aza olacak şekilde sırayla d4, d2, d1 ve d3 olarak gerçekleşmiştir (bkz. Şekil 6.28). İşlemci kullanımlarına bakıldığında en yoğun olan veri merkezi d1'dir fakat 25. dakikadan sonra bellek kullanımındaki üssel artış ile işlemci kullanımı ve IOPS miktarı ciddi oranda düşmüştür. Kalan zamanda düşüşün devam etmesi neticesinde d1'deki veri depolama miktarı ciddi oranda düşmüştür. Veri merkezi d3 için işlemci kullanımı 30%'u geçmemiştir, bellek kullanımında gerçekleşen üssel artış ile veri depolama miktarı en az seviyede olan veri merkezi olmuştur. Veri merkezi d2 ve d4 için işlemci kullanımları 30%-50% aralığında, bellek kullanımları 10%-30% aralığında ve IOPS miktarları 40%-70% aralığında gerçekleşmiştir. Veri merkezi d4'te bellek kullanımı daha az olduğu için veri depolama miktarı d2'den daha fazla olmuştur.

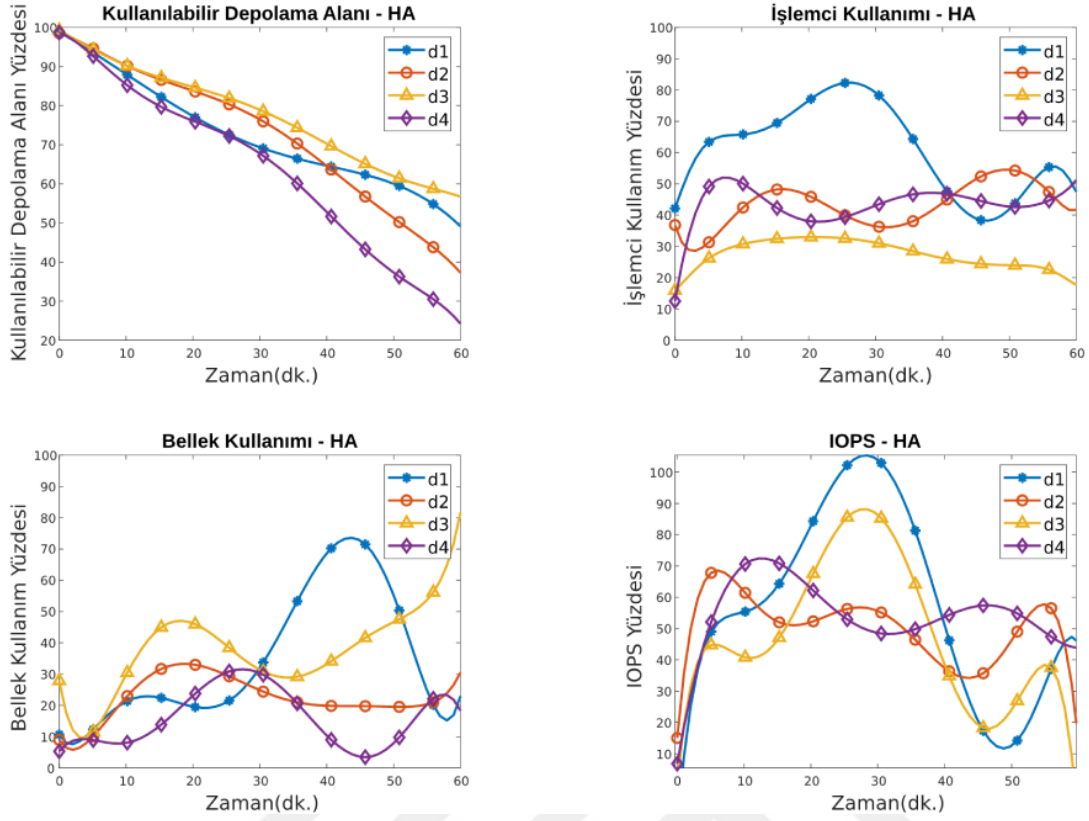
PAA yönteminde veri depolama miktarları, çoktan aza olacak şekilde sırayla d1=d2, d4 ve d3 olarak gerçekleşmiştir (bkz. Şekil 6.29). İşlemci kullanımı en fazla veri merkezi d4'te olmakla birlikte d1, d2 ve d4 için 30%-60% aralığında, d3 için ise 10%-35% aralığında gerçekleşmiştir. Bellek kullanımları tüm veri merkezlerinde 15%-45% aralığında değişmektedir. IOPS miktarları ise tüm veri merkezlerinde 30%-70% aralığında değişmektedir. PAA yöntemi kaynak durumlarını dengelediği için grafiklerde tepe oluşumu gözlemlenmemiştir. Böylece topolojik olarak ağır daha ortalarında bulunmakta olan veri

merkezlerine (d1, d2) daha fazla veri depolanmakla birlikte daha dışarda kalan veri merkezleri (d3, d4) de diğer yöntemlere oranla daha fazla veri depolamayı başarmıştır.

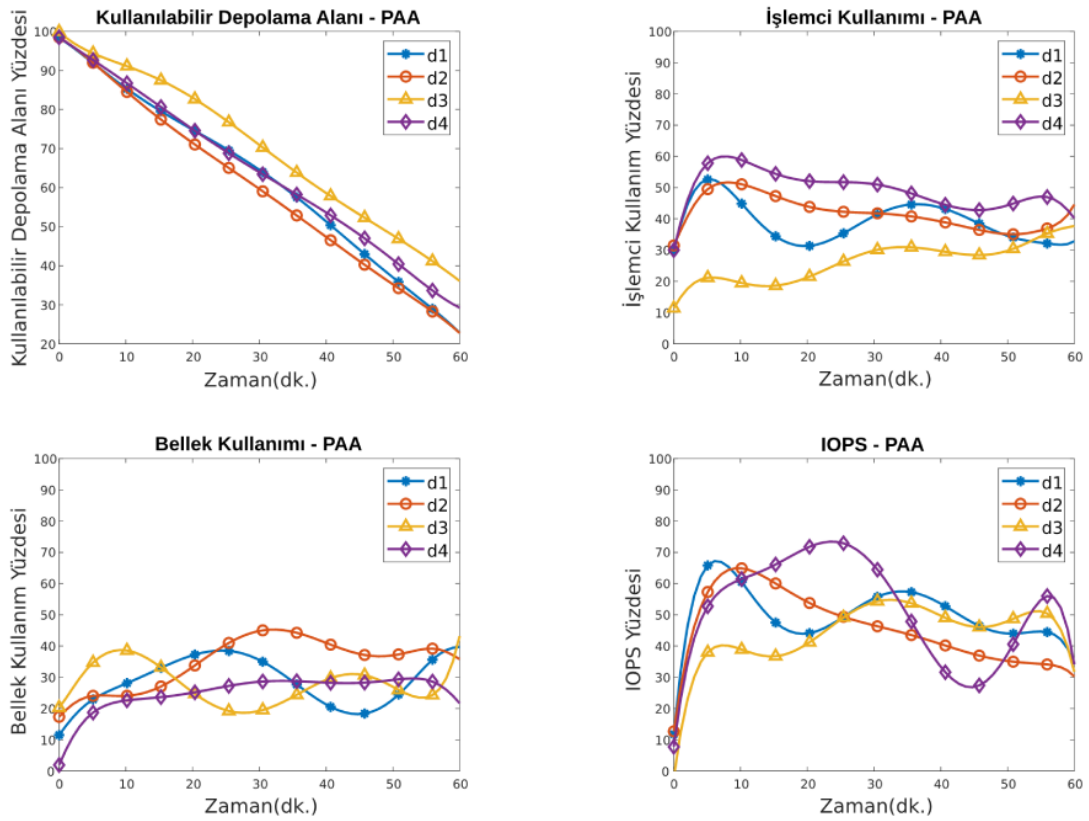
HDFS yönteminde tüm veri merkezlerinde yakın miktarda veri depolanmıştır (bkz. Şekil 6.30). Tüm veri merkezlerinde bellek kullanımları 10%-40% aralığında gerçekleşmiştir. İşlemci kullanımlarında ise diğer yöntemlere oranla büyük salınımlar meydana gelmiştir. Bunun nedeni veri merkezlerinin DataNode olarak HDFS'e özel algoritmalar çalıştırmasıdır. Diğer yandan IOPS miktarı da 60%-100% aralığında gerçekleşmiştir ve bu miktarlar diğer yöntemlere oranla daha fazladır. Replika konfigürasyonu 1 olmasına rağmen IOPS değerinin bu kadar fazla olmasının nedeni veri merkezlerinde gerçekleşen tekrar dengeleme işlemleridir. Böylece, işlemcideki salınım ve IOPS miktarındaki artışlar HDFS'in daha az veri depolamasına neden olmuştur.



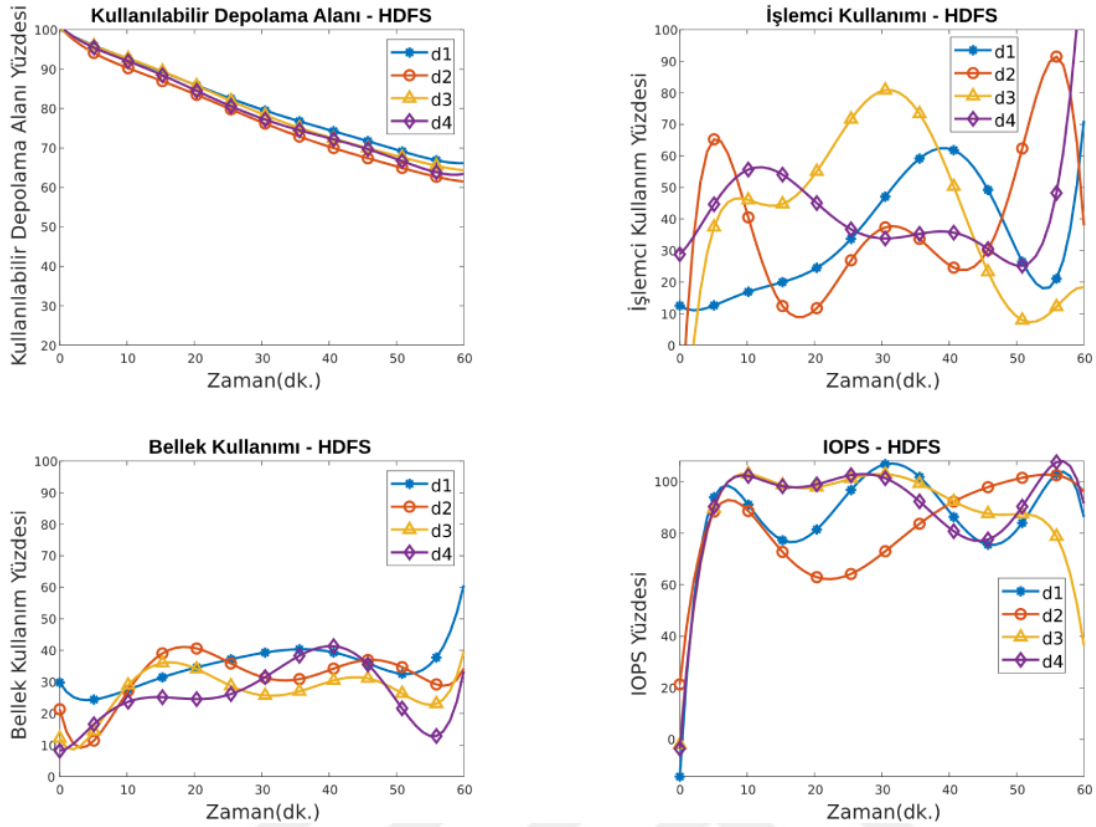
Şekil 6.27. RA için veri merkezlerindeki kaynak durumlarının toplu gösterimi



Şekil 6.28. HA için veri merkezlerindeki kaynak durumlarının toplu gösterimi



Şekil 6.29. PAA için veri merkezlerindeki kaynak durumlarının toplu gösterimi



Şekil 6.30. HDFS için veri merkezlerindeki kaynak durumlarının toplu gösterimi

6.4. Bilinen Kısıtlar

Önerilen sistemin limitleri aşağıdaki gibi sıralanabilir:

- Önerilen model, simülasyon boyunca bir çözüm matrisi üretirken, belirli durumlarda bazı veri merkezlerinden kaynak yükü bilgilerini alamayabilir. Örneğin, bir veri merkezi büyük miktarda veriyle yoğunlaştıysa veya ağ sorunları yaşıyorsa (daha yüksek gecikme süresi, daha az bant genişliği vb.) merkezi bilgisayara anlık yük bilgilerini gönderemeyebilir ve bu da meşgul olarak kabul edilmesine neden olabilir. Bunun sonucunda, kamera bilgisayarı ilgili veri merkezine atanamaz. Tüm veri merkezleri meşgul duruma düşerse, önerilen sistem veri merkezlerinden biri kullanılabilir hale gelene kadar atama yapamaz.
- Lagrangian gevşemesinin ana dezavantajı, kısıtlamanın gevşetilmesi nedeniyle optimal bir çözüm bulma garantisi sağlamamasıdır. Önerilen yöntem, modeli ilk aşamada doğrusal programlama ile çözer, böylece kısıtlamaları sağlayan kamera bilgisayarları veri merkezlerine atanmış olur. Tüm kamera bilgisayarları ilk aşamada atanmışsa, ikinci aşama gerekli değildir. Ancak, bazı kamera bilgisayarları atanamıyorsa, ikinci aşama

kamera bilgisayarlarının küçük bir alt kümesi için yerel bir optimum çözüm bulmak üzere Lagrange gevşemesi tekniğini kullanır. Böylece, dezavantaj az sayıda kısıtlama ihlali ile minimum bir değere düşürülür. İkinci aşamada çok fazla kamera bilgisayarı atamak, sistem içindeki genel kaynak miktarlarının artırılmasını gerektirebilecek kaynak kısıtlamalarına yol açabilir.

- RS uygulamaları genellikle ulusal yönetimler ve savunma sanayii için hayati önem taşımaktadır. Bu nedenle, simülasyonlarda PAA, yazılım kaynaklarının yalnızca tek bir müşteriye tahsis edildiği ve yalnızca tek bir müşteri tarafından erişilebildiği coğrafi olarak dağıtılmış bir özel bulut üzerinde çalıştırılmıştır. Genel bulut üzerinde çalıştırılmak isteniyorsa, önerilen modele bir güvenlik katmanı eklenmelidir.



7. SONUÇ

Teknolojik gelişmelerin bir sonucu olarak, birçok kuruluşun günlük bazda ürettiği veri miktarı terabayt seviyesine yükseldi. Uzamsal ve spektral olarak çoğaltılan ve farklı algılama teknikleriyle heterojen olan uzaktan algılama verileri, geleneksel yöntemlerle depolama, aktarma ve analiz etme konusunda büyük zorluklara neden olmaktadır. İlgili büyük verinin gerçek / yakın gerçek zamanlı analizine ihtiyaç duyulduğunda, kritik uygulamalarda yetersiz kalan geleneksel yöntemlerin yerine dağıtık yaklaşımların uygulanması bir zorunluluk haline gelmiştir. Mevcut dağıtık dosya sistemleri, veri tabanları ve yüksek performanslı hesaplama sistemleri, uzaktan algılama ve veri akışlarının uzamsal büyük verilerinin depolanması, alınması ve analizi için iş akışlarını optimize etmede zorluklar yaşamaktadır.

Bu çalışmada incelenen araştırmalara göre, mevcut tekniklerin ve sistemlerin uzaktan algılamada gerçek / yakın gerçek zamanlı büyük veri analizini mümkün kılmak için mevcut sorunları kapsayan bir çözüm bulamamış oldukları görülmüştür. Hadoop ve benzeri teknolojiler, büyük jeo-uzamsal verileri işlemek için jeoloji toplumlarında giderek daha fazla ilgi çekmeye başlamıştır. Büyük jeo-uzamsal verilerin işlenmesi için bu teknolojilerin benimsenmesine yönelik birçok araştırma yapılmıştır, ancak dinamik jeo-işleme iş yükünü verimli bir şekilde işlemek amacıyla bilgisayar kaynaklarının optimize edilmesi için çok az çalışma bulunmaktadır.

Bu tez çalışmasında dağıtılmış veri merkezlerinden oluşan coğrafi dağıtık bulut üzerinde çalıştırılacak olan RS uygulamaları için yeni bir büyük veri yönetimi sistemi önerilmiştir. Bu sistem, hangi kamera bilgisayarının RS verilerini hangi veri merkezine göndereceğine karar vermek için çok kısıtlı bir atama problemini çözmektedir. Sistem, ağ ölçümlerini ve kaynak kapasitelerini topladıktan sonra, atama problemini iki aşamalı bir optimizasyon modeli olarak çözer. İlk aşamada model, sınırsız kaynakları sayesinde geçerli olmayan çözümü engelleyen sahte bir veri merkezini de sisteme dahil ederek 0-1 tamsayı problemini çözer. İkinci aşamada ise ilk aşamada sahte veri merkezine atanan kamera bilgisayarları varsa, bunlar Lagrange yöntemi kullanılarak yaklaşık bir çözümle gerçek veri merkezlerine atanır. Bunu yaparken sensörler ve veri merkezleri arasında veri aktarım süresini ve hata oranını azaltmak için ağ metriği olarak bant genişliği, gecikme ve sıçrama sayısını kullanır. Diğer yandan büyük veri analizinin verimli bir şekilde yapılabilmesi için veri

merkezlerindeki işlemci, depolama, bellek kaynakları ve iş yükünü hesaba katarak kaynak paylaşımını optimize eder. Önerilen PAA yöntemini, sanal test ağında çalıştırarak karşılaştırmak için RA, HA ve HDFS yöntemleri geliştirildi. Deneysel sonuçlara göre önerilen PAA yöntemi, bir saatlik simülasyonda diğer yöntemlerden daha başarılı olarak RA, HA ve HDFS'ten sırasıyla %36, %26 ve %71 daha fazla/hızlı depolama gerçekleştirmiştir. Ayrıca, kaynak kullanımlarını belirli bir seviyede tutarak tüm kaynak kullanımlarında salınımı engellemiştir. Böylece önerilen yöntem, kaynak ve ağ kullanımını optimize ederek bir merkezi yönetim maliyeti dahilinde veri merkezlerinde daha verimli bir veri analizi işlemine imkan sağlayabilecektir.

RS ve büyük veri yönetimi alanında araştırmanın literatüre katkıları aşağıdaki gibi özetlenebilir:

- Burada önerilen sistemde kaynakları optimize ederek RS verilerinin depolanması ve yönetilmesi için kamera bilgisayarları ve veri merkezleri arasındaki eşleştirme literatürde ilk kez çok kısıtlı bir atama problemi olarak ele alınmıştır.
- Önerilen sistemde ağ topolojisinin avantajlarından yararlanılmış, veri merkezlerinin işlemci, depolama ve bellek kaynaklarının kullanımları dikkate alınarak iş yükleri dengelenmiştir. Böylece sensörlerden alınacak ve veri merkezinde saklanacak veriler en uygun kaynak ve ağ hızı ile veri merkezine gönderilecek, bu da ilgili verilerin daha hızlı aktarılmasını, depolanmasını ve işlenmesini sağlayacaktır.
- RS verilerinin veri merkezinde bir bütün olarak tutulmasının sıklıkla bitişik piksellere ihtiyaç duyan RS uygulamaları için analiz süreçlerinde veri okuma süresini azaltacak olması önerilen modelimizin bir diğer önemli avantajıdır.
- Önerilen modelimizin ilk aşamasında veriler bir bütün olarak veri merkezlerine atanırken, kaynak yetersizliği durumunda ikinci aşamaya geçilmektedir. İkinci aşamada, Lagrange gevşemesi ile karşılanamayan kısıtlar gevşetilir ve ilk aşamada atanamayan veriler atanır. İkinci aşama, kısıtlamaları karşılamak için verilerin parçalanıp birden fazla veri merkezine atanıp atanamayacağını belirler.
- Model, kaynak doluluğunu bir aralık içinde korur ve kaynakları dengeleyerek salınımı önler. Böylece daha kararlı ve güvenilir bir bulut ortamı oluşturur.

Gelecekte yapılması planlanan çalışmalar şu şekildedir:

- Önerilen PAA sistemi, simülasyonlar yoluyla elde edilen veriler üzerinde yapay zeka tekniklerini kullanarak, gecikme, bant genişliği ve kaynak kullanılabilirliğindeki gerçek zamanlı değişikliklere dayanarak ağırlık matrisini adaptif bir şekilde belirleyecektir.
- Simülasyon verilerini ve elde edilen Lagrange çözümlerini temel alarak atama problemini öğrenme algoritmaları vasıtasıyla çözen bir model geliştirilmesi planlanmaktadır.
- Sistemin boşta olduğu zaman aralıklarında veri merkezlerine indirilen verilerin istemci kullanımı için yeniden düzenlenmesi planlanmaktadır. Bu nedenle, PAA, veri işleme sunucularını ve istemci isteklerinin ağdaki konumunu dikkate alarak kaynak tüketimini optimize edecektir.
- Son olarak, önerilen sistemde bir güvenlik katmanı geliştirilerek genel bulut üzerinde çalıştırılabilecek bir yapıya geçilmesi hedeflenmektedir.



KAYNAKLAR

- Abdullah, F., Peng, L., and Tak, B. (2021). A survey of iot stream query execution latency optimization within edge and cloud. *Wireless Communications and Mobile Computing*, 1-16.
- Aji, A., Wang, F., Vo, H., Lee, R., Liu, Q., Zhang, X., and Saltz, J. (2013, August). *Hadoop-GIS: A high performance spatial data warehousing system over MapReduce*. In Proceedings of the VLDB Endowment International Conference on Very Large Data Bases, 6(11). NIH Public Access.
- Amalia, I. Z., Saikhu, A., and Soelaiman, R. (2021). A fast dynamic assignment algorithm for solving resource allocation problems. *Jurnal Online Informatika*, 6(1), 118-127.
- Angleraud, C. (2014, June). *magHD: A new approach to multi-dimensional data storage, analysis, display and exploitation*. In IOP Conference Series: Earth and Environmental Science, 20(1), p. 012035. IOP Publishing.
- Assunção, M. D., Calheiros, R. N., Bianchi, S., Netto, M. A., and Buyya, R. (2015). Big data computing and clouds: Trends and future directions. *Journal Of Parallel And Distributed Computing*, 79, 3-15.
- Bergui, M., Najah, S., and Nikolov, N. S. (2021). A survey on bandwidth-aware geo-distributed frameworks for big-data analytics. *Journal of Big Data*, 8(1), 1-26.
- Bernabe, S., López, S., Plaza, A., and Sarmiento, R. (2012). GPU implementation of an automatic target detection and classification algorithm for hyperspectral image analysis. *IEEE Geoscience and Remote Sensing Letters*, 10(2), 221-225.
- Bernabe, S., Sanchez, S., Plaza, A., López, S., Benediktsson, J. A., and Sarmiento, R. (2013). Hyperspectral unmixing on GPUs and multi-core processors: A comparison. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(3), 1386-1398.
- Bruce, L. M. (2013, July). *Game theory applied to big data analytics in geosciences and remote sensing*. In 2013 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 4094-4097.
- Burachik, R. S., Gasimov, R. N., Ismayilova, N. A., and Kaya, C. Y. (2006). On a modified subgradient algorithm for dual problems via sharp augmented Lagrangian. *Journal of Global Optimization*, 34(1), 55-78.
- Cappelaere, P., Sánchez, S., Bernabé, S., Scuri, A., Mandl, D., and Plaza, A. (2013). Cloud implementation of a full hyperspectral unmixing chain within the NASA web coverage processing service for EO-1. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(2), 408-418.

- Cavallaro, G., Riedel, M., Bodenstein, C., Glock, P., Richerzhagen, M., Götz, M., and Benediktsson, J. A. (2015, July). *Scalable developments for big data analytics in remote sensing*. In 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 1366-1369.
- Chang, Y. L., Chao, H. T., Huang, M. Y., Chang, L., Fang, J. P., and Hsieh, T. J. (2014, December). *Incenter-based nearest feature space method for hyperspectral image classification using GPU*. In 2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS), 931-936.
- Chen, C. P. and Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, 275, 314-347.
- Chen, J., Zheng, G., and Chen, H. (2013, June). *ELM-MapReduce: MapReduce accelerated extreme learning machine for big spatial data analysis*. In 2013 10th IEEE International Conference on Control and Automation (ICCA), 400-405.
- Cui, D., Wu, Y., and Zhang, Q. (2010, May). *Massive spatial data processing model based on cloud computing model*. In 2010 Third International Joint Conference on Computational Science and Optimization, 2, 347-350.
- Deng, M., Di, L., Yu, G., Yagci, A., Peng, C., Zhang, B., and Shen, D. (2012, July). *Building an on-demand web service system for global agricultural drought monitoring and forecasting*. In 2012 IEEE International Geoscience and Remote Sensing Symposium, 958-961.
- Di Modica, G. and Tomarchio, O. (2022). A hierarchical Hadoop framework to process geodistributed big data. *Big Data and Cognitive Computing*, 6(1), 5.
- Douglas, C. C. (2014). An open framework for dynamic big-data-driven application systems (DBDDAS) development. *Procedia Computer Science*, 29, 1246-1255.
- Durmaz, H. and Koyuncu, M. (2019). Optimization of assignment problems in production lines with different skilled labor levels. *International Advanced Researches and Engineering Journal*, 3(2), 123-136.
- Fernández, A., Gómez, Á., Lecumberry, F., Pardo, Á., and Ramírez, I. (2015). Pattern recognition in Latin America in the Big Data era. *Pattern Recognition*, 48(4), 1185-1196.
- Gao, S., Li, L., Li, W., Janowicz, K., and Zhang, Y. (2017). Constructing gazetteers from volunteered big geo-data based on Hadoop. *Computers, Environment and Urban Systems*, 61, 172-186.
- Giachetta, R. (2015). A framework for processing large scale geospatial and remote sensing data in MapReduce environment. *Computers and Graphics*, 49, 37-46.
- Gowda, M. S. and Hulyal, V. G. (2015, April). *Parallel image processing from cloud using cuda and hadoop architecture: a novel approach*. In Int'l Conference on Computational Systems for Health and Sustainability (CSFHS), 2020-2023.

- Gui, Z., Yu, M., Yang, C., Jiang, Y., Chen, S., Xia, J., and Jin, B. (2016). Developing subdomain allocation algorithms based on spatial and communicational constraints to accelerate dust storm simulation. *Plos One*, 11(4), p. 0152250.
- Guo, W., Gong, J., Jiang, W., Liu, Y., and She, B. (2010). OpenRS-Cloud: A remote sensing image processing platform based on cloud computing environment. *Science China Technological Sciences*, 53(1), 221-230.
- Guo, W., She, B., and Zhu, X. (2014). Remote sensing image on-demand computing schema for the china zy-3 satellite private cloud-computing platform. *Transactions in GIS*, 18, 53-75.
- Gupta, R. K. (1992). *Operations research*. India: Krishna Prakashan Media.
- Hooker, J. N. (2006). Operations research methods in constraint programming. *Foundations of Artificial Intelligence*, 2, 527-570.
- Islam, M., Huang, A. K., Battisha, M., Chiang, M., Srinivasan, S., Peters, C., and Abdelnur, A. (2012, May). *Oozie: Towards a scalable workflow management system for hadoop*. In Proceedings of the 1st ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies, 1-10.
- Islam, N. S., Rahman, M. W., Jose, J., Rajachandrasekar, R., Wang, H., Subramoni, H., and Panda, D. K. (2012, November). *High performance RDMA-based design of HDFS over InfiniBand*. In SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, 1-12.
- İnternet: AVIRIS (2021). Airborne Visible InfraRed Imaging Spectrometer, NASA JPL. URL: <https://aviris.jpl.nasa.gov/>, Son Erişim Tarihi: 01.06.2023.
- İnternet: GAMS (2021). The General Algebraic Modeling System (GAMS). URL: <https://gams.com/>, Son Erişim Tarihi: 01.06.2023.
- İnternet: iPerf (2016). The ultimate speed test tool for TCP, UDP and SCTP. URL: <https://iperf.fr/>, Son Erişim Tarihi: 01.06.2023.
- İnternet: Lawrence Livermore National Laboratory (LLNL) (2021) , Introduction to Parallel Computing Tutorial. URL: <https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial>, Son Erişim Tarihi: 01.06.2023.
- İnternet: NVIDIA (2023), GeForce RTX 40 Series Graphics Cards Details. URL: <https://www.nvidia.com/en-in/geforce/news/rtx-40-series-graphics-cards-announcements/>, Son Erişim Tarihi: 01.06.2023.
- İnternet: Quagga (2018). Quagga Routing Software Suite. URL: <https://www.nongnu.org/quagga/>, Son Erişim Tarihi: 01.06.2023.
- İnternet: XILINX (2023), AMD Artix 7 Product Details. URL: <https://www.xilinx.com/products/silicon-devices/fpga/artix-7.html>, Son Erişim Tarihi: 01.06.2023.

- Jonathan, A., Chandra, A., and Weissman, J. (2020, December). *WASP: Wide-area adaptive stream processing*. In Proceedings of the 21st International Middleware Conference, 221-235.
- Kambatla, K., Kollias, G., Kumar, V., and Grama, A. (2014). Trends in big data analytics. *Journal of Parallel and Distributed Computing*, 74(7), 2561-2573.
- Kang, S. J., Lee, S. Y., and Lee, K. M. (2015). Performance comparison of OpenMP, MPI, and MapReduce in practical problems. *Advances in Multimedia*, 2015, 7-15.
- Karantzalos, K., Bliziotis, D., and Karmas, A. (2015). A scalable geospatial web service for near real-time, high-resolution land cover mapping. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(10), 4665-4674.
- Karun, A. K. and Chitharanjan, K. (2013, April). *A review on Hadoop - HDFS infrastructure extensions*. In 2013 IEEE Conference On Information and Communication Technologies, 132-137.
- Krämer, M. and Senner, I. (2015). A modular software architecture for processing of big geospatial data in the cloud. *Computers and Graphics*, 49, 69-81.
- Lin, F. C., Chung, L. K., Ku, W. Y., Chu, L. R., and Chou, T. Y. (2013, March). *The framework of cloud computing platform for massive remote sensing images*. In 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (Aina), 621-628.
- Lin, F. C., Chung, L. K., Wang, C. J., Ku, W. Y., and Chou, T. Y. (2013). Storage and processing of massive remote sensing images using a novel cloud computing platform. *GIScience and Remote Sensing*, 50(3), 322-336.
- Liu, K. and Boehm, J. (2015). Classification of big point cloud data using cloud computing. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40, 553-557.
- Liu, Y., Chen, B., Yu, H., Zhao, Y., Huang, Z., and Fang, Y. (2011, June). *Applying GPU and POSIX thread technologies in massive remote sensing image data processing*. In 2011 19th International Conference on Geoinformatics, 1-6.
- Lunga, D., Gerrand, J., Yang, L., Layton, C., and Stewart, R. (2020). Apache spark accelerated deep learning inference for large scale satellite image analytics. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 271-283.
- Ma, Y., Wang, L., Liu, D., Liu, P., Wang, J., and Tao, J. (2012, September). *Generic parallel programming for massive remote sensing data processing*. In 2012 IEEE International Conference on Cluster Computing, 420-428.
- Ma, Y., Wang, L., Liu, P., and Ranjan, R. (2015). Towards building a data-intensive index for big data computing: A case study of remote sensing data processing. *Information Sciences*, 319, 171-188.

- Ma, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A., and Jie, W. (2015). Remote sensing big data computing: Challenges and opportunities. *Future Generation Computer Systems*, 51, 47-60.
- Manolakis, D., Marden, D., and Shaw, G. A. (2003). Hyperspectral image processing for automatic target detection applications. *Lincoln Laboratory Journal*, 14(1), 79-116.
- Mills-Tettey, G. A., Stentz, A., and Dias, M. B. (2007, July). *The dynamic hungarian algorithm for the assignment problem with changing costs*. Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-07-27.
- Moguel, E., Preciado, J. C., Sánchez-Figueroa, F., Preciado, M. A., and Hernández, J. (2015). Multilayer big data architecture for remote sensing in Eolic parks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(10), 4714-4719.
- Narayanan, I., Kansal, A., Sivasubramaniam, A., Urgaonkar, B., and Govindan, S. (2014, June). *Towards a leaner geo-distributed cloud infrastructure*. In 6th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 14).
- Nda, R. M. and Tasmin, R. B. (2019). Big data management in education sector: An overview. *Path of Science*, 5(6), 5009-5014.
- Nedic, A. and Ozdaglar, A. (2010). Cooperative distributed multi-agent optimization. *Convex Optimization in Signal Processing and Communications*, 340-386.
- O'Driscoll, A., Daugelaite, J., and Sleator, R. D. (2013). Big data, Hadoop and cloud computing in genomics. *Journal of Biomedical Informatics*, 46(5), 774-781.
- Oliveira, S. F., Förlinger, K., and Kranzlmüller, D. (2012, June). *Trends in computation, communication and storage and the consequences for data-intensive science*. In 2012 IEEE 14th International Conference on High Performance Computing and Communication, 572-579.
- Plaza, A., Du, Q., Chang, Y. L., and King, R. L. (2011). High performance computing for hyperspectral remote sensing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 4(3), 528-544.
- Plaza, A., Valencia, D., Plaza, J., Sánchez-Testal, J., Munoz, S., and Blázquez, S. (2006, July). *Parallel implementation of hyperspectral image processing algorithms*. In 2006 IEEE International Symposium on Geoscience and Remote Sensing, 940-943.
- Qu, H., Zhang, J., Chen, Y., Chen, H., and Lin, Z. (2012, July). *Parallel implementation for SAM algorithm based on GPU and distributed computing*. In 2012 IEEE International Geoscience and Remote Sensing Symposium, 4074-4077.
- Qu, H., Zhang, J., Lin, Z., and Chen, H. (2013). Parallel acceleration of SAM algorithm and performance analysis. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(3), 1172-1178.

- Rathore, M. M. U., Ahmad, A., Paul, A., and Wu, J. (2016). Real-time continuous feature extraction in large size satellite images. *Journal of Systems Architecture*, 64, 122-132.
- Schmuck, F. and Haskin, R. (2002). GPFS: *A shared-disk file system for large computing clusters*. In Conference on File and Storage Technologies (FAST 02), 231-244.
- Setoain, J., Prieto, M., Tenllado, C., and Tirado, F. (2008). GPU for parallel on-board hyperspectral image processing. *The International Journal of High Performance Computing Applications*, 22(4), 424-437.
- Shan, N., Wang, X. S., and Wang, Z. S. (2010, September). *Efficient FPGA implementation of cloud detection for real-time remote sensing image processing*. In 2010 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), 190-193.
- Sharma, R. A. and Chandra, R. (2020, February). *DeepEdge: A network edge for deep learning workloads*. In: 16th USENIX Symposium on Networked Systems Design and Implementation (2019).
- Sierra-Pajuelo, F., Paz-Gallardo, A., and Plaza, A. (2015, March). *Performance optimizations for an automatic target generation process in hyperspectral analysis*. In ARCS 2015-The 28th International Conference on Architecture of Computing Systems, 1-6.
- Stanford, C. (2020). *Geo-distributed stream processing*. Survey for the UPenn WPE II.
- Tantisiriroj, W., Son, S. W., Patil, S., Lang, S. J., Gibson, G., and Ross, R. B. (2011, November). *On the duality of data-intensive file system design: Reconciling HDFS and PVFS*. In Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, 1-12.
- Tehrani, S., Zhao, Y., Harvey, T., Swaroop, A., and McKenzie, K. (2006). A robust framework for real-time distributed processing of satellite data. *Journal of Parallel and Distributed Computing*, 66(3), 403-418.
- Toma, A. C., Panica, S., Zaharie, D., and Petcu, D. (2012, October). *Computational challenges in processing large hyperspectral images*. In 2012 5th Romania Tier 2 Federation Grid, Cloud and High Performance Computing Science (RQLCG), 111-114.
- Torti, E., Acquistapace, M., Danese, G., Leporati, F., and Plaza, A. (2014). Real-time identification of hyperspectral subspaces. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6), 2680-2687.
- Tso, F. P., Jouet, S., and Pezaros, D. P. (2016). Network and server resource management strategies for data centre infrastructures: A survey. *Computer Networks*, 106, 209-225.
- Tumwizere, R. P., and Karpjoo, J. (2012, August). *A survey on computing technology applications in remote sensing*. In 2012 8th International Conference on Computing and Networking Technology (INC, ICCIS and ICMIC), 77-80.

- Valencia, D., Plaza, A., Martínez, P., and Plaza, J. (2005, June). *On the use of cluster computing architectures for implementation of hyperspectral image analysis algorithms*. In 10th IEEE Symposium on Computers and Communications (ISCC'05), 995-1000.
- Vora, M. N. (2011, December). *Hadoop-HBase for large-scale data*. In Proceedings of 2011 International Conference on Computer Science and Network Technology, 601-605.
- Wang, L., Cheng, C., Wu, S., Wu, F., and Teng, W. (2015, July). *Massive remote sensing image data management based on HBase and GeoSOT*. In 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 4558-4561.
- Wang, L., Yan, J., and Ma, Y. (2019). *Cloud computing in remote sensing*. Boca Raton: CRC Press.
- Wang, L., Ma, Y., Yan, J., Chang, V., and Zomaya, A. Y. (2018). pipsCloud: High performance cloud computing for remote sensing big data management and processing. *Future Generation Computer Systems*, 78, 353-368.
- Wang, P., Wang, J., Chen, Y., and Ni, G. (2013). Rapid processing of remote sensing images based on cloud computing. *Future Generation Computer Systems*, 29(8), 1963-1968.
- Witte, R. S. and Witte, J. S. (2017). *Statistics*. Hoboken: John Wiley and Sons.
- Wu, T. Y., Chen, C. Y., Kuo, L. S., Lee, W. T., and Chao, H. C. (2012). Cloud-based image processing system with priority-based data distribution mechanism. *Computer Communications*, 35(15), 1809-1818.
- Wu, X., Huang, B., Plaza, A., Li, Y., and Wu, C. (2013). Real-time implementation of the pixel purity index algorithm for endmember identification on GPUs. *IEEE Geoscience and Remote Sensing Letters*, 11(5), 955-959.
- Wu, Y., Wu, C., Li, B., Zhang, L., Li, Z., and Lau, F. C. (2014). Scaling social media applications into geo-distributed clouds. *IEEE/ACM Transactions On Networking*, 23(3), 689-702.
- Wu, Z., Wang, Q., Plaza, A., Li, J., Liu, J., and Wei, Z. (2015). Parallel implementation of sparse representation classifiers for hyperspectral imagery on GPUs. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6), 2912-2925.
- Xu, C., Du, X., Fan, X., Yan, Z., Kang, X., Zhu, J., and Hu, Z. (2021). A modular remote sensing big data framework. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1-11.
- Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., and Fay, D. (2011). Spatial cloud computing: How can the geospatial sciences use and help shape cloud computing. *International Journal of Digital Earth*, 4(4), 305-329.

- Yang, C., Zhang, X., Zhong, C., Liu, C., Pei, J., Ramamohanarao, K., and Chen, J. (2014). A spatiotemporal compression based approach for efficient big data processing on cloud. *Journal of Computer and System Sciences*, 80(8), 1563-1583.
- Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauly, M., and Stoica, I. (2012, April). *Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing*. In 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12), 15-28.
- Zhong, Y., Fang, J., and Zhao, X. (2013, July). *VegaIndexer: A distributed composite index scheme for big spatio-temporal sensor data on cloud*. In 2013 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 1713-1716.





Gazili olmak ayrıcalıktır...