



APACHE HADOOP VE DAĞITIK SİSTEMLER ÜZERİNDEKİ ROLÜ

Gürcan YAVUZ, Sevcan AYTEKİN, Muammer AKÇAY

Dumlupınar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Kütahya,
gurcanyavuz@dpu.edu.tr, saytekin@dpu.edu.tr, makcay@dpu.edu.tr

Geliş Tarihi: 22.11.2010

Kabul Tarihi: 11.10.2011

ÖZET

Dağıtık sistemler, kaynakları paylaşmaya duyulan gereklilik neticesinde ortaya çıkmıştır. Kaynak paylaşımı donanımsal bileşenler ve yazılımsal varlıklar olabilir. Özellikle büyük çaplı verilerin güvenli, hızlı, tutarlı ve ölçeklenebilir bir şekilde işlenebilmesi için dağıtık sistemlerin kullanılabilir olduğu görülmüştür. Bu amaçla Apache Hadoop, dağıtık sistemlerde özellikle son yıllarda çokça kullanılmaya başlayan bir yazılım haline gelmiştir. Amacı Google Dosya Sistemi teknolojisi olan MapReduce algoritmasını birçok bilgisayara dağıtarak uygulamayı sağlamaktır. Google dosya sisteminin benzerini kullanarak çok büyük karmaşık verileri ölçeklendirebilme imkânı sağlar. Bu imkânla Hadoop'un kullanılabilirliği giderek artmaktadır. Bu çalışmada, mapreduce açık kaynak gerçekleştirilmiş hali hadoop çatısı incelenmiştir. Üç makineye kurulumu yapıлып, küçük bir küme oluşturulmuştur. Hadoop çatısı, küme üzerinde çalışması incelenmiştir.

Anahtar Kelimeler: *Dağıtık Sistemler, Hadoop, Paralel, Bulut, MapReduce*

APACHE HADOOP AND ROLE ON DISTRIBUTED SYSTEMS

ABSTRACT

Distributed systems have emerged because of the need for sharing of sources. Sharing of sources can be hardware component and software properties. Especially distributed systems can be useful for safe, fast and scalable process for a large amount of data. For this reason, Apache Hadoop software is being started to be mostly used especially in distributed systems recent years. Its purpose is distributing MapReduce algorithm which refers to Google File System technology to a lot of computer. It allows scaling a large amount of complicated data with Google File System similar. For this possibility availability of Hadoop is increasing. In this work, one of the MapReduce open sourced implementation hadoop is examined which is installed to three machine in the small cluster.

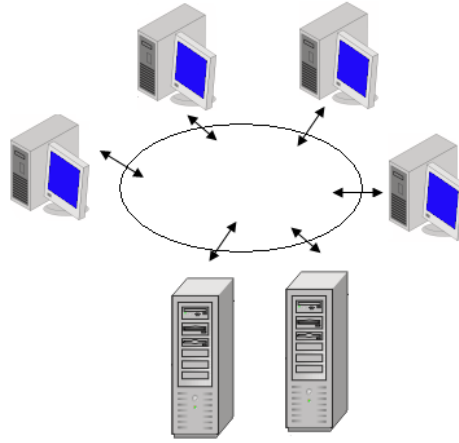
Keywords: *Distributed Systems, Hadoop, Parallel, Cloud, MapReduce*

1. GİRİŞ

Birçok farklı alanda kullanılan verilerin sayısal ortamda işlenmesi ile çeşitli ihtiyaçların ortaya çıkması söz konusu olmuştur. Günümüzde birçok alanda (mühendislik, tıp, bilişim, ...) kullanılan verilerin güvenilir, etkin, hızlı işlenebilmesi için sağlıklı olarak saklanabileceği çeşitli kaynaklara ihtiyaç vardır. Tüm kullanıcıların ortak talebi verilerin güvenle saklanıp etkin bir şekilde işlenmesidir. Zamana bağlı olarak verilerin de hızlı artış göstermesi kaynak ihtiyacını yani verilerin tutulacağı alan gereksinimini ortaya çıkarmaktadır. Bu ihtiyacı gidermenin ortak yollarından biri kaynak sayısını veya veri alanının büyüklüğünü arttırmaktır. Bu amaçla verilerin bütünlüğünü bozmadan birden fazla kaynak üzerinde paylaşılması sağlanarak işlenebilirliği ortaya çıkmıştır. Böylelikle kaynak paylaşımının gerekliliği sonucu Dağıtık Sistemler ortaya çıkmıştır. Bu çalışmada Dağıtık Sistemlerin Apache Hadoop [1] yazılımı ile farklı düğümler üzerinden birbiriyle haberleşmesi incelenmiştir. Apache Hadoop yazılımı ve alt sistemleri (Hbase [2], Pig [3], Hive [4], ZooKeeper [5]) ile Dağıtık Sistemler üzerindeki etkinliği ve performansı tek makine ve birden fazla makinede test edilmeye çalışılmıştır.

2. DAĞITIK SİSTEMLER

Bir Dağıtık sistem, hafıza ve saati paylaşmayan işlemciler kümesidir. Her işlemci kendisine ait yerel hafızaya sahiptir. İşlemciler birbirleriyle iletişim ağları ile haberleşirler [6]. Dağıtık sistemlerde bilgi işleme, tek bir makinede sınırlanmamış birkaç bilgisayar üzerine dağıtılmıştır. Büyük bilgisayar tabanlı sistemlerin çoğu dağıtık sistemlere geçmiştir. Dağıtık sistemlerde; donanım ve yazılım kaynaklarının paylaşımı yapılabilir, bir hata oluştuğundan sonra işleme devam edilebilir, eş zamanlı işleme sayesinde performans artırılabilir, farklı sağlayıcılardan gelen yazılım ve donanımlar kullanılabilir. Tüm bu faydaları dışında dağıtık sistemler genelde merkezi sistemlere göre daha karmaşıktır ve sistem yönetimi için daha fazla uğraş gerekir. Çoklu işlemci mimarileri en basit dağıtık sistem modelidir [7].



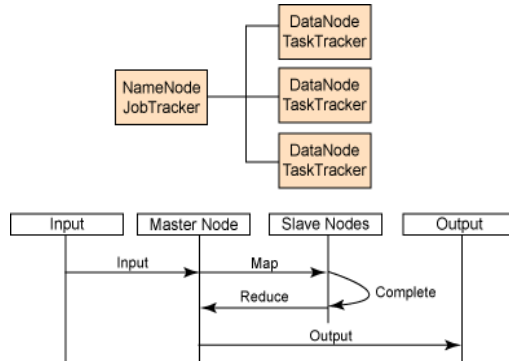
Şekil 1: Dağıtık Sistem Modeli

Dağıtık sistem, Şekil 1’de görüldüğü gibi bir ağ üzerine kuruludur. Yazılımla, ağdaki bileşenler arası işlevsel uyumluluk ve kullanıcı açısından saydamlık sağlanır. Bir ağdan farklı olarak ağ üzerinde gerçekleşen işlemler kullanıcıya bir görünmektedir yani kullanıcı karşısında, tek ve bütünlüklü bir yapı görülmektedir (Transparency). Kullanıcı, bir işlemci ve bir ara yüz olduğunu düşünse de bu sistemler, değişik bilgisayar sistemleri üzerindeki verileri ve işlemleri bir bütün olarak işleyebilir ve çalıştırabilirler. Dağıtık sistemlere, sunucu-istemci, peer-to-peer (noktadan noktaya) modelleri örnek gösterilebilir [8].

3. HADOOP

MapReduce, Google tarafından geliştirilmiş bir programlama modelidir [9]. Küme boyunca dağılmış çok büyük veri kümelerini işlemeye kullanılır. MapReduce uygulaması en az üç ana parçadan oluşmaktadır: Map fonksiyonu, Reduce Fonksiyonu ve Ana Fonksiyon. Map işleminde Master düğüm veri girişlerini alır, bunları küçük alt görevlere böler ve işçi düğümlere dağıtır. İşçi düğümleri bu görevi yerine getirir ve yaptığı işlemin cevabını Master düğüme geri gönderir. Reduce işleminde Master düğüm, alt görevlerden gelen cevapları alır ve sonucu almak için bu görevleri birleştirir (Şekil 2).

Çeşitli gruplar, MapReduce ortaya çıktığından buyana kendi MapReduce çatısını gerçekleştirmişlerdir [10][11]. Hadoop, Java [12] programlama dili ile yazılmış, Apache tarafından devam ettirilen açık kaynak MapReduce çatısıdır. Üzerinde tutarlı, ölçeklenebilir ve dağıtık çalışan projeler geliştirmeye imkân sağlayacak bir çatıdır. Çok büyük verileri dağıtık olarak işlemeye izin veren bir yazılım mimarisidir. Proje, Google’ın MapReduce[9] ve GFS [13] (Google File System) makalelerinden esinlenmiş ve birçok dev firma tarafından da kullanılmaktadır. Genellikle endüstriyel ve akademik alanda olmak üzere başta Yahoo, IBM, Facebook, Adobe gibi yaklaşık 70’in üzerinde firma Hadoop yazılımını kullanmaktadır [14].



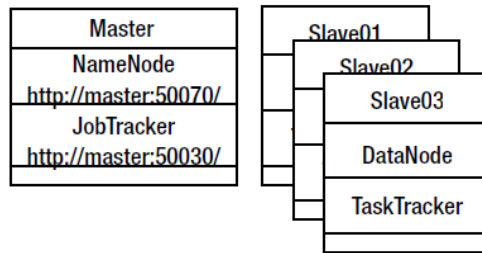
Şekil 2: Hadoop Küme ve MapReduce Modeli [15]

3.1. Hadoop Mimarisi

Hadoop birçok özelliği bakımından verilerin güvenli, etkin ve ölçeklenebilir olarak işlenmesini sağlar. Hesaplanan elemanların ve depolamanın başarısız olabileceğini varsayarak verilerin yedek kopyası ile çalışmaktadır[16]. Eğer bir düğümde bir hata meydana gelirse, başka bir düğümde bulunan sağlam bir kopyasını tekrar kopyalama yapmaya başlar, böylece verilerin güvenli bir şekilde tutulmasını sağlar. Hadoop paralellime prensipleriyle çalışır, verilerin paralel işlenmesini, böylece etkinliğini korumaktadır. Ölçeklenebilir, petabyte boyutlarındaki verileri işlemeye izin verir.

Hadoop mimarisinin en alt katmanında HDFS(Hadoop Distributed File System) [16] dosya sistemi bulunmaktadır. HDFS dosya sistemi hadoop kümesinde verileri düğümler boyunca kaydetmektedir. Dışarıdan bir istemciye HDFS; geleneksel hiyerarşik dosya sistemi gibi görünmektedir. CRUD [17] işlemler yapılabilir. Yani dosyalar oluşturulabilir, silinebilir bilinen dosya işlemleri yapılabilir. HDFS mimarisi özel düğümlerden oluşmaktadır. Bunlar;

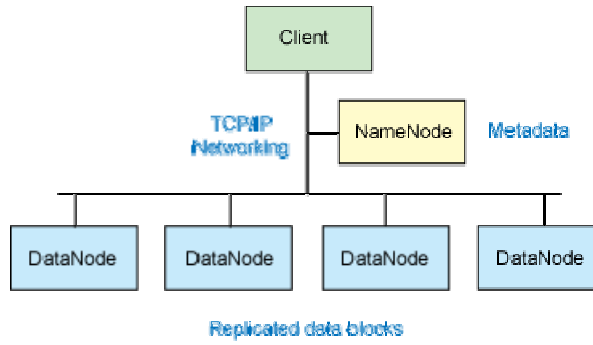
1. NameNode: HDFS içinde metadata hizmeti vermektedir.
2. DataNode: HDFS için depolama blokları sunar.



Şekil 3: Bir Master ve üç Slave Düğüm Modeli[18]

HDFS'nin üst katmanında MapReduce engine (motor) bulunmaktadır. Bu engine JobTracker'lardan ve TaskTracker'lardan oluşmaktadır (Şekil 3).

HDFS içinde dosyalar bloklara bölünürler. Bu bloklar çoklu olarak bilgisayarlara kopyalanır(replicate). Blok boyutları genellikle 64 MB'dir. Blok boyutu ve verilerin kopya sayısı kullanıcı tarafından belirlenebilir. Bütün dosya işlemleri NameNode ile yönetilmektedir. HDFS içindeki tüm iletişimler TCP/IP protokolü ile gerçekleştirilmektedir (Şekil 4).



Şekil 4:Hadoop Kümesi'nin basitleştirilmiş görünümü[19]

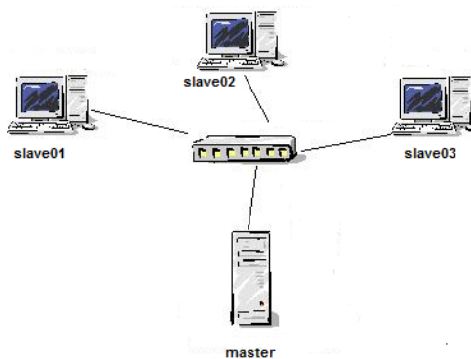
Hadoop içinde HBase, Hive, Pig gibi birçok bileşeni kapsar (HBase, Hive, Pig, Mahout, Hama, ZooKeeper,...). Bu bileşenlerden biri olan HBase, Hadoop'un veri tabanıdır. Çok sayıda verinin (milyarlarca satırlar halindeki verilerin) depolandığı okunup yazılabilen bir tablo olarak düşünülebilir. Google ait BigTable [20] projesinin javada yazılmış açık kaynak karşılığıdır.

3.2. Hadoop Kurulumu

Hadoop'un çeşitli modlarda kurulumunu yapmak mümkündür. Hadoop üç tip kurulum moduna sahiptir. Bunlar; Standalone Mod, Pseudo Distributed Mod, Fully Distributed Mod olmak üzeredir.

1. **Standalone (Local) Mod** : Tek düğümde çalışmazdır. Hadoop kurulduğunda bu mod öntanımlı olarak gelir. Debug amaçlı kullanılır.
2. **Pseudo-Distributed Mod** : Tek bir düğümde, her bir hadoop daemon, ayrı java süreci olarak çalışmaktadır.
3. **Fully-Distributed Mode** : Hadoop, küme üzerinde çalıştırma modudur. Birden fazla düğümden oluşmaktadır.

Bu çalışmada Fully Distributed Mod kurulum özellikleri üzerinde durulmuştur. Bu kurulum birden fazla makine üzerine Hadoop'un kurulum ve ayar işlemlerini kapsar.



Şekil 5: Anahtar üzerinden Master-Slave düğümlerin bağlantısı

Bu çalışmada donanım özellikleri farklı üç tane düğüme sahip Hadoop kümesi kurulumu anlatılmaktadır (Şekil 5). Her makinede Debian 5.03 [21] sürümü mevcuttur.

Makine adları ve özellikleri aşağıda belirtilmiştir.

Makine adları:

```
yavuz  
worker1  
worker2
```

- yavuz: AMD Athlon(tm) Dual Core Processor 5400B
- worker1: Intel(R) Pentium(R) 4 CPU 3.00GHz
- worker2: Intel(R) Core(TM)2 Duo CPU E8500 @ 3.16GHz

- Hadoop'un ihtiyaç duyduğu gerekli paketlerin kurulumu apt-get aracı ile gerçekleştirilir.

```
# apt-get install ssh  
# apt-get install rsync
```

- Hadoop ait özel bir kullanıcı oluşturulur.

```
yavuz:/home/gurcan# adduser hadoop
```

- hadoop kullanıcısı olarak giriş yapılır.

```
gurcan@yavuz:~$ su hadoop  
Parola:
```

- hadoop kullanıcısına ait olan ev dizini içerisine \$HOME/tmp adlı bir dizin oluşturulur.

```
hadoop@yavuz:~$ mkdir tmp
```

- hadoop kullanıcısının ev dizinine hadoop programının en son sürümünü indirilir.

```
hadoop@yavuz:~$ wget  
http://ftp.itu.edu.tr/Mirror/Apache/hadoop/core/hadoop-0.20.2/hadoop-  
0.20.2.tar.gz
```

- İndirilmiş olan hadoop arşivi /opt dizini altına kopyalanır.

```
yavuz:/home/hadoop# cp hadoop-0.20.2.tar.gz /opt
```

- /opt altında sıkıştırılmış hadoop paketi açılır.

```
yavuz:/home/hadoop# cd /opt/  
yavuz:/opt# tar -xvf hadoop-0.20.2.tar.gz
```

- Sıkıştırılmış dosya açılınca bir dizin oluşur. Buna kolay erişim için bir sembolik link oluşturulur.

```
yavuz:/opt# ln -s hadoop-0.20.2 hadoop
```

- Oluşan dizin ve sembolik linkin sahipliğini root kullanıcılarından hadoop kullanıcısı olarak değiştirilir.

```
yavuz:/opt# chown -R hadoop:hadoop hadoop-0.20.2
yavuz:/opt# chown -R hadoop:hadoop hadoop
```

- Kümede kullanılacak makinelerin her birine statik ip tanımlaması yapılır.

- Master olacak makinede:

```
yavuz:/opt# vim /etc/network/interfaces
```

Aşağıdaki satırlar eklenir:

```
auto lo
iface lo inet loopback

auto eth0
#
## Static IP atama işlemi yapılıyor.
##
iface eth0 inet static
    address 10.1.29.165
    netmask 255.255.248.0
    gateway 10.1.24.1
```

- Worker1 olacak makinede:

```
worker1:/home/hadoop# vim /etc/network/interfaces
```

Aşağıdaki satırlar eklenir:

```
auto lo
iface lo inet loopback

auto eth0

# Elle ip atama
iface eth0 inet static
    address 10.1.29.139
    netmask 255.255.248.0
```

- Worker2 olacak makinede:

```
worker2:/home/hadoop# vim /etc/network/interfaces
```

Aşağıdaki satırlar eklenir:

```
auto lo
iface lo inet loopback

#iface eth0 inet dhcp
auto eth0

iface eth0 inet static
    address 10.1.27.142
    netmask 255.255.248.0
    gateway 10.1.24.1
```

- Tüm makinelerde /etc/hosts dosya içeriğinin tanımlanması yapılır.

```
yavuz:/home/hadoop# vim /etc/hosts
worker1:/home/hadoop# vim /etc/hosts
worker2:/home/hadoop# vim /etc/hosts
```

Aşağıdaki satırlar eklenir:

```
127.0.0.1    localhost

# host isimleri
10.1.29.165 yavuz
10.1.29.139 worker1
10.1.27.142 worker2
```

- (MASTER) hadoop kümede kesintisiz iletişimi sağlamak için şifresiz ssh ayarının yapılması gerekir. Bunun için, public key oluşturup authorized_keys dosyasına eklenir.

```
hadoop@yavuz:~$ ssh-keygen -t rsa -f ~/.ssh/id_rsa
hadoop@yavuz:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

- Şifresiz bağlantının test edilmesi yapılır, eğer şifre istemez ise ayarlamalar doğrudur.

```
hadoop@yavuz:~$ ssh localhost
```

- Yukarıda oluşturulan public key kümede bulunacak tüm Slave makinelerin authorized_keys dosyasına eklenir. Bunun için: Master yani yavuz adlı makinesinden public key dosyası slave makinelerin tümüne kopyalanıp, authorized_keys dosyasının sonuna bu dosya içeriği eklenir.

```
hadoop@yavuz:~$ scp .ssh/id_rsa.pub worker1:/home/hadoop/
hadoop@worker1:~$ cat id_rsa.pub >> .ssh/authorized_keys
```

```
hadoop@yavuz:~$ scp .ssh/id_rsa.pub worker2:/home/hadoop/
hadoop@worker2:~$ cat id_rsa.pub >> .ssh/authorized_keys
```

- Yukarıdaki işlemler yapıldıktan sonra artık yavuz makinesinden worker1 ve worker2 bilgisayarlarına şifresiz bağlantı yapılabilir.

```
hadoop@yavuz:~$ ssh worker1  
hadoop@yavuz:~$ ssh worker2
```

3.3. Ayarlamalar

- Hadoop, yol tanımlamaları \$HOME/.bashrc dosyasının en alt satırına yapılmaktadır. Bu java ve hadoop kurulu olduğu dizinleri göstermektedir. Bunlar PATH değişkenine eklenir.

```
hadoop@yavuz:~$ vim .bashrc
```

Aşağıdaki satırlar dosyanın sonuna eklenir:

```
export JAVA_HOME=/usr/lib/jdk  
export HADOOP_INSTALL=/opt/hadoop  
export PATH=$PATH:$HADOOP_INSTALL/bin:$JAVA_HOME/bin:
```

Bu dosya tüm düğümlere gönderilir.

```
hadoop@yavuz:~$ scp .bashrc worker1:/home/hadoop/  
hadoop@yavuz:~$ scp .bashrc worker2:/home/hadoop/
```

- core-site.xml dosyasının açılır. Aşağıdaki satırlar eklenir. **fs.default.name** özelliği ile NameNode makinesi adı ve port numarası tanımlaması yapılır.

```
hadoop@yavuz:/opt/hadoop/conf$ vim core-site.xml
```

```
<?xml version="1.0"?>  
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  
<configuration>  
  <property>  
    <name>hadoop.tmp.dir</name>  
    <value>/home/hadoop/tmp</value>  
  </property>  
  <property>  
    <name>fs.default.name</name>  
    <value>hdfs://yavuz:9000</value>  
  </property>  
</configuration>
```

Bu dosya tüm kümedeki bütün düğümlere gönderilir.

```
hadoop@yavuz:/opt/hadoop/conf$ scp core-site.xml worker1:/opt/hadoop/conf/  
hadoop@yavuz:/opt/hadoop/conf$ scp core-site.xml worker2:/opt/hadoop/conf/
```

- mapred-site.xml dosyası açılarak aşağıdaki satırlar eklenir. Bu dosya ile JobTracker makinesinin adı ve port numarası tanımlaması yapılır.

```
hadoop@yavuz:/opt/hadoop/conf$ vim mapred-site.xml
```

```
<?xml version="1.0"?>  
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
```



```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>yavuz:9001</value>
  </property>
</configuration>
```

Bu dosya tüm düğümlere gönderilir.

```
hadoop@yavuz:/opt/hadoop/conf$ scp mapred-site.xml worker1:/opt/hadoop/conf/
hadoop@yavuz:/opt/hadoop/conf$ scp mapred-site.xml worker2:/opt/hadoop/conf/
```

- `hdfs-site.xml` dosyası açılarak aşağıdaki satırlar eklenir. **dfs.replication** değişkeni değiştirilir. Bu değişken ile verilerin kopya sayısı belirlenir. Böylelikle verilerin güvenliği sağlanmaktadır.

```
hadoop@yavuz:/opt/hadoop/conf$ vim hdfs-site.xml
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
</configuration>
```

Bu dosya tüm düğümlere gönderilir.

```
hadoop@yavuz:/opt/hadoop/conf$ scp hdfs-site.xml
```

```
worker1:/opt/hadoop/conf/
```

```
hadoop@yavuz:/opt/hadoop/conf$ scp hdfs-site.xml worker2:/opt/hadoop/conf/
```

- `hadoop-env.sh` dosyası açılarak aşağıdaki satırlar eklenir. Java'nın nerede olduğu belirtilmektedir.

```
#export JAVA_HOME=/usr/lib/j2sdk1.5-sun
```

(java'nın /usr/lib/jdk altına kurulu olduğu kabul edilmiştir.)

```
export JAVA_HOME=/usr/lib/jdk
```

Bu dosya tüm düğümlere gönderilir.

```
hadoop@yavuz:/opt/hadoop/conf$ scp hadoop-env.sh worker1:/opt/hadoop/conf/
hadoop@yavuz:/opt/hadoop/conf$ scp hadoop-env.sh worker2:/opt/hadoop/conf/
```

- `masters` dosyası açılarak master olan makinenin ismi eklenir. (sadece master olan makinede bu işlem yapılır)

```
hadoop@yavuz:/opt/hadoop/conf$ vim masters yavuz
```

- `slaves` dosyası açılarak tüm makinelerin ismi eklenir. Bu dosyada slave olarak çalıştırılmak istenen makinelerin adları buraya yazılır. (sadece master olan makinede bu işlem yapılır) :

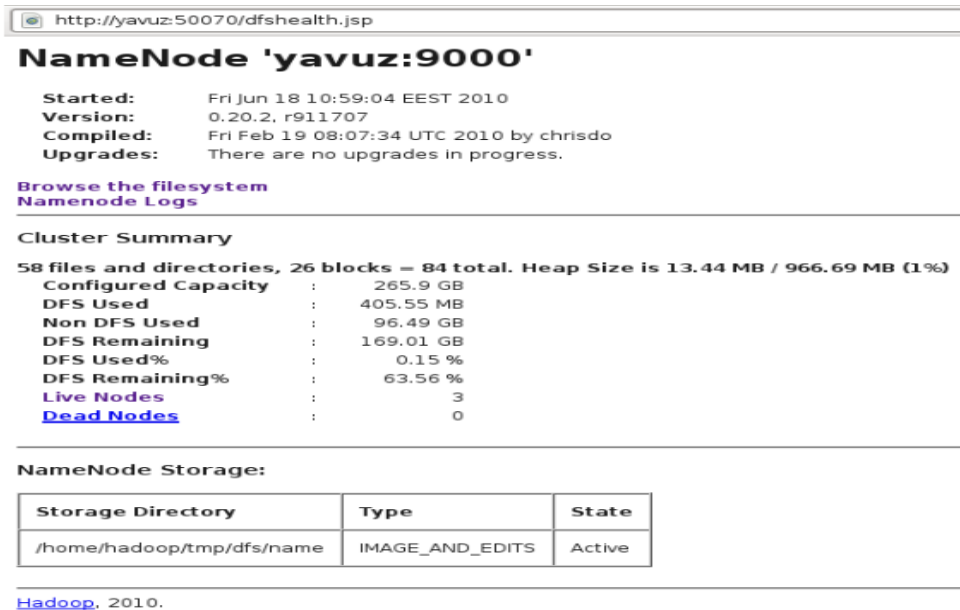
```
hadoop@yavuz:/opt/hadoop/conf$ vim slaves
yavuz
worker1
worker2
```

- Kurulum adımları tamamlandıktan sonra küme üzerinde kullanılacak hdfs dosya sistemi, formatlanır.

```
hadoop@yavuz:~$ hadoop NameNode -format
```

- Formatlama işlemi tamamlandığında, servisler başlatılır.

```
hadoop@yavuz:~$ start-dfs.sh
hadoop@yavuz:~$ start-mapred.sh
```



The screenshot shows the Hadoop NameNode web interface for the node 'yavuz:9000'. It displays the following information:

- NameNode 'yavuz:9000'**
- Started:** Fri Jun 18 10:59:04 EEST 2010
- Version:** 0.20.2, r911707
- Compiled:** Fri Feb 19 08:07:34 UTC 2010 by chrisdo
- Upgrades:** There are no upgrades in progress.

Below this, there are links for "Browse the filesystem" and "Namenode Logs".

Cluster Summary

58 files and directories, 26 blocks = 84 total. Heap Size is 13.44 MB / 966.69 MB (1%)

Configured Capacity	: 265.9 GB
DFS Used	: 405.55 MB
Non DFS Used	: 96.49 GB
DFS Remaining	: 169.01 GB
DFS Used%	: 0.15 %
DFS Remaining%	: 63.56 %
Live Nodes	: 3
Dead Nodes	: 0

NameNode Storage:

Storage Directory	Type	State
/home/hadoop/tmp/dfs/name	IMAGE_AND_EDITS	Active

At the bottom, there is a link to "Hadoop, 2010."

Şekil 6. Kümeye ait özet bilgileri

Hadoop, küme ile ilgili genel bilgileri elde etmek için çeşitli araçlar ile gelmektedir. Hadoop, web ara yüz aracı sayesinde tarayıcı yardımıyla kümeye ait bilgilere ulaşılabilir. Tutulan loglar, hdfs dosya sistemi gezilebilmektedir. Şekil 6 da `http://yavuz:50070` adresinden elde edilen ekran görüntüsü yer almaktadır. Kümeye ait çalışan, ölü düğüm, dosya sistemi kullanımı gibi özet bilgiler görülmektedir.

The screenshot shows the NameNode 'yavuz:9000' web interface. It displays the following information:

- NameNode 'yavuz:9000'**
- Started:** Fri Jun 18 10:59:04 EEST 2010
- Version:** 0.20.2, r911707
- Compiled:** Fri Feb 19 08:07:34 UTC 2010 by chrisdo
- Upgrades:** There are no upgrades in progress.
- Links:** [Browse the filesystem](#), [Namenode Logs](#), [Go back to DFS home](#)
- Live Datanodes : 3**

Node	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks
worker1	1	In Service	12.85	0.14	1.1	11.61	1.05		90.36	16
worker2	1	In Service	38.45	0.13	4.47	33.84	0.35		88.02	15
yavuz	0	In Service	214.6	0.13	90.98	123.5	0.06		57.55	21

[Hadoop](#), 2010.

Şekil 7. Çalışan Düğümlerin Listesi

Şekil 7 de çalışan düğümlere ait bilgiler gözükmemekte. Burada, bu düğümlerdeki dosya sistemi kullanma miktarları yer almaktadır. Şekil 8 de daha önce icra edilmiş işler listelenmektedir. İş yapan kullanıcının adı, işin öncelik durumu gibi bilgilere ulaşılmaktadır.

Completed Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information
job_201006181059_0001	NORMAL	hadoop	job1343085795327896384.jar		2	2		0	0	NA
job_201006181059_0002	NORMAL	hadoop	job3960696931891320404.jar		2	2		1	1	NA
job_201006181059_0003	NORMAL	hadoop	job1664444738780756024.jar		2	2		1	1	NA
job_201006181059_0004	NORMAL	hadoop	job5142024569833639523.jar		2	2		1	1	NA
job_201006181059_0006	NORMAL	hadoop	job1524481052958790085.jar		2	2		1	1	NA
job_201006181059_0008	NORMAL	hadoop	job8692431736604921079.jar		2	2		1	1	NA

Şekil 8. Tamamlanmış işlerin listesi

- Servislerin kapatılması için stop-all adlı bir bash betiği çalıştırılmaktadır

```
hadoop@yavuz:~$ stop-all.sh
```

SONUÇLAR

Bu çalışmada, aynı işletim sistemi ve yazılımların kurulu olduğu farklı donanım özelliklerine sahip bilgisayarların Hadoop çatısı üzerinden haberleşmesi sağlanmıştır. Bunun için Hadoop'un ihtiyaç duyduğu gerekli paketlerin kurulumu ve ayarlamaları ile xml dosyalarının düzenlenmesi anlatılmıştır. Tek makine ve birden fazla düğüm üzerindeki değişiklikler incelenmiştir.

Çalışmanın bundan sonraki kısmında küme içindeki düğüm sayıları artırılarak kümenin genişletilmesi amaçlanmaktadır. Hbase, Pig, Zookeeper projeleri küme üzerine kurularak, Hadoop ile çalışması sağlanacaktır. Büyük verilerin işlenmesinde oluşturulan kümenin kullanılması düşünülmektedir.

KAYNAKLAR

- [1] Apache Hadoop, <http://hadoop.apache.org/>
- [2] Apache HBase, <http://hbase.apache.org/>
- [3] Apache Pig, <http://hadoop.apache.org/pig/>
- [4] Apache Hive, <http://hadoop.apache.org/hive/>
- [5] Apache Zookeeper, <http://hadoop.apache.org/zookeeper/>
- [6] A. Silberschatz, P. B. Galvin, Greg Gagne, “Operating System Concepts 7th Ed”, John Wiley&Sons,611,(2004)
- [7] J. WU, “Distributed System Design”, CRC-Press, (1998)
- [8] İnternet: İşletim Sistemleri Dağıtık Sistemler (Çevrimiçi – “www.pinarsahin.net/dosyalar/isli1.ppt”)
- [9] J. Dean , S. Ghemawat, “MapReduce: simplified data processing on large clusters”, Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation, p.10-10, San Francisco, (2004)
- [10] İnternet: <http://en.wikipedia.org/wiki/MapReduce>, (2010)
- [11] J. A. Stuart, C. K. Chen, K. L. Ma, J. D. Owens, “Multi-GPU volume rendering using MapReduce”, High Performance Distributed Computing Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, Chicago, Illinois, USA (2010)
- [12] İnternet: java, <http://www.java.com>, (2010)
- [13] S. Ghemawat , H. Gobioff , S. Leung, “The Google file system”, Proceedings of the nineteenth ACM symposium on Operating systems principles, Bolton Landing, NY, USA (2003)
- [14] İnternet: <http://wiki.apache.org/hadoop/PoweredBy>(2010)
- [15] İnternet: http://www.ibm.com/developerworks/aix/library/au-cloud_apache/(2010)
- [16] K. Shvachko, K. Hairong, S. Radia, R. Chansle, “The Hadoop Distributed File System”, Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium, Incline Village, NV, USA (2010)
- [17] İnternet: Create, read, update and delete http://en.wikipedia.org/wiki/Create,_read,_update_and_delete (2010)
- [18] J. Venner, “Pro Hadoop”, Apress, 1st ed. (2009)
- [19] İnternet: Ken Mann, M. Tim Jones Distributed computing with Linux and Hadoop, <http://www.ibm.com/developerworks/linux/library/l-hadoop/> (2010)
- [20] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, R. E. Gruber, “Bigtable: A Distributed Storage System for Structured Data”, OSDI'06: Seventh Symposium on Operating System Design and Implementation, Seattle, WA, (2006)
- [21] İnternet: Debian GNU/Linux <http://www.debian.org/>(2010)