

**TÜRK HAVA KURUMU ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**TEK KART BİLGİSAYAR KÜMELERİ ÜZERİNDE BULUT**

**YÜKSEK LİSANS TEZİ**

**Levent AYSAN**

**Bilişim Teknolojileri Anabilim Dalı**

**Yüksek Lisans Programı**

**EKİM 2015**

**TÜRK HAVA KURUMU ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**TEK KART BİLGİSAYAR KÜMELERİ ÜZERİNDE BULUT**

**YÜKSEK LİSANS TEZİ**

**Levent AYSAN**

**1303657019**

**Bilişim Teknolojileri Anabilim Dalı**

**Yüksek Lisans Programı**

**Tez Danışmanı: Doç. Dr. İzzet Gökhan ÖZBİLGİN**

Türk Hava Kurumu Üniversitesi Fen Bilimleri Enstitüsü'nün 1303657019 numaralı Yüksek Lisans öğrencisi “Levent AYSAN“ ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “Tek Kart Bilgisayar Kümeleri Üzerinde Bulut” başlıklı tezini, aşağıda imzaları bulunan jüri önünde başarı ile sunmuştur.

**Tez Danışmanı : Doç. Dr. İzzet Gökhan ÖZBİLGİN**  
**Türk Hava Kurumu Üniversitesi**



**Jüri Üyeleri : Doç. Dr. İzzet Gökhan ÖZBİLGİN**  
**Türk Hava Kurumu Üniversitesi**



**: Yrd. Doç. Dr. İbrahim Alper DOĞRU**  
**Gazi Üniversitesi**



**: Yrd. Doç. Dr. Meltem İMAMOĞLU**  
**Türk Hava Kurumu Üniversitesi**

.....

**Tez Savunma Tarihi: 12.10.2015**

**TÜRK HAVA KURUMU ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE**

Yüksek lisans tezi olarak sunduğum “Tek Kart Bilgisayar Kümeleri Üzerinde Bulut” adlı çalışmanın tarafımdan, akademik etik ve kurallara aykırı düşecek bir yardıma başvurmaksızın yazıldığını ve yararlandığım kaynakların kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yaparak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

12/10/2015

Levent AYSAN



## ÖNSÖZ

Topluluk içinde işlerin gönüllü ve el birliği içinde yapılmasına imece adı verilirken para toplanmasına salma adı verilmektedir. İmece ve salma köylerde yazılı olmayan hukuka dayalı, herkes tarafından kabul gören bir dayanışma örgütüdür. Günümüz bilişim teknolojilerinde de aynı yaklaşımları görebilmekteyiz. Yapmak ve ya yaptırmak mümkün olmadığı zaman, ilan edip birilerinin işe talip olmasını bekleyip onlara yaptırmak kitle kaynak olarak adlandırılmaktadır. Vikipedi ağ sitesi ve Linux işletim sistemi buna örnektir.

Açık kaynak modeli, geliştirmenin merkezi modellerinin tersine üretimde eş zamanlı, farklı gündemleri ve yaklaşımları içerir. Geliştirmenin merkezli olan modelleri, öncelikle kâr amacı güden yazılım şirketleri tarafından kullanılır. Açık kaynak ise kullanıcıya çalıştırma, kopyalama, dağıtma, inceleme, değiştirme ve geliştirme özgürlüklerini tanımaktadır.

Gerçekleştirilen çalışmada açık kaynak donanım olan tek kart bilgisayarları bir araya getirerek, açık kaynak işletim sistemi, açık kaynak sanallaştırma programı ve açık kaynak veri eşleme indirgeme programları kullanımı ile bulut oluşturulması sağlanmaktadır. Bu teknolojilerin toplam sahip olma maliyetleri düşük, enerji tüketimleri ve karbon salınımları düşük ancak işlem güçleri veri işleme uygulamalarında doğru kullanımda oldukça etkindir. Bulut kabiliyetleri yeterlilikleri ile bir çok farklı alanda bilişim teknolojisi gereksinimlerini karşılayabilecek durumdadırlar.

Ekim 2015

Levent AYSAN

## İÇİNDEKİLER

ÖNSÖZ .....	iv
TABLO LİSTESİ.....	vii
ŞEKİL LİSTESİ.....	viii
KISALTMALAR .....	viii
ÖZET.....	x
ABSTRACT.....	xi
GİRİŞ .....	1
<b>BİRİNCİ BÖLÜM.....</b>	<b>3</b>
<b>1. LİTERATÜR VE KAVRAMLAR.....</b>	<b>3</b>
1.1 Bulut Bilişim ve Bilgisayar Kümeleri .....	3
1.2 Paralel ve Dağıtık Hesaplama .....	3
1.3 Sanallaştırma .....	4
<b>İKİNCİ BÖLÜM .....</b>	<b>5</b>
<b>2. UYGULAMA BİLEŞENLERİ.....</b>	<b>5</b>
2.1 Tek Kart Bilgisayar Donanımları .....	5
2.1.1 Adaptea Parallella Tek Kart Bilgisayar .....	6
2.1.2 Parallella Hedef Uygulamaları.....	10
2.2 Linux İşletim Sistemi .....	11
2.3 İşletim Sistemi Seviyesinde Sanallaştırma .....	12
2.4 MapReduce.....	13
2.4.1 Hadoop (hadoop.apache.org).....	15
2.4.2 Hadoop HDFS.....	16
2.4.3 Hadoop YARN .....	17
2.4.4 Hadoop MapReduce .....	19
<b>ÜÇÜNCÜ BÖLÜM .....</b>	<b>21</b>
<b>3. TEK KART BİLGİSAYAR İLE BULUT OLUŞTURMA.....</b>	<b>21</b>
3.1 Tek Kart Bilgisayarları Kümeleme .....	23
3.2 İşlem Adımları .....	26
3.2.1 Bulut Üzerinde Çalışma.....	26
3.2.1.1 WordCount.....	26
3.2.1.2 WordMean.....	27
3.2.1.3 TeraGen.....	28
3.2.1.4 TeraSort.....	28
3.2.1.5 TeraValidate .....	28
3.2.1.6 TestDFSIO -read .....	28
3.2.1.7 TestDFSIO -write.....	29
3.2.1.8 $\pi$ (pi) sayısı hesaplama .....	29
3.3 Çalışma Çıktıları.....	30
3.4 Değerlendirme ve Tartışma .....	39
<b>DÖRDÜNCÜ BÖLÜM .....</b>	<b>42</b>
<b>4. SONUÇ VE ÖNERİLER.....</b>	<b>42</b>

<b>KAYNAKÇA</b> .....	45
<b>EKLER</b> .....	50
Ek-A: Linaro Kurulumu .....	51
Ek-B: LXC Kurulumu.....	55
Ek-C: Hadoop Kurulumu .....	56
<b>ÖZGEÇMİŞ</b> .....	59

## TABLO LİSTESİ

<b>Tablo 2.1</b>	: Parallela özellikleri.....	6
<b>Tablo 2.2</b>	: TIANHE-2 TH-IVB-FEP küme yapısı.....	12
<b>Tablo 3.1</b>	: İşlemci ve Yonga üzeri sistem karşılaştırmas üsttetableda yer almaktadır.15 kat fark görülmektedir.....	21
<b>Tablo 3.2</b>	: ARM ve Intel örnek işlemci kıyaslaması.....	22
<b>Tablo 3.3</b>	: Referans sistem.....	22
<b>Tablo 3.4</b>	: Zynq 7000 özellikleri.....	25
<b>Tablo 3.5</b>	: Çeşitli hesaplama senaryoları için farklı sanal makine yapılandırılmalı bulut üzerine çalışma sonuçları.....	30
<b>Tablo 3.6</b>	: 1 Pi Sayısı hesaplama için farklı sanal makine yapılandırılmalı bulut üzerine çalışma sonuçları.....	31
<b>Tablo 3.7</b>	: DataNode bilgileri.....	32
<b>Tablo 3.8</b>	: 4 adet tek kart bilgisayar üzerinde 4 LXC çalışırken her biri üzerindeki yük dağılımı.....	33
<b>Tablo 3.9</b>	: 4 adet tek kart bilgisayar üzerinde 4 LXC çalışırken her biri üzerindeki bellek durumu.....	34
<b>Tablo 3.10</b>	: 4 adet tek kart bilgisayar üzerinde 4 LXC çalışırken her biri üzerindeki işlemci dağılımı.....	35
<b>Tablo 3.11</b>	: 4 adet tek kart bilgisayar üzerinde 4 LXC çalışırken her biri üzerindeki ağ durumu.....	36
<b>Tablo 3.12</b>	: Tek kart bilgisayar üzerinde 2 LXC ve 1 LXC çalışırken bellek işlemci durumları.....	37
<b>Tablo 3.13</b>	: Tek kart bilgisayar üzerinde 2 LXC ve 1 LXC çalışırken bellek işlem durumları.....	38



## ŞEKİL LİSTESİ

Şekil 2.1	: Parallella tek kart bilgisayar üst seviye mimarisi.....	8
Şekil 2.2	: Parallella tek kart bilgisayar üst görünüm.....	9
Şekil 2.3	: Parallella tek kart bilgisayar alt görünüm. ....	10
Şekil 2.4	: MapReduce fiziksel işleyişi. ....	14
Şekil 2.5	: MapReduce mantıksal işleyişi.....	15
Şekil 2.6	: Hdfs Mimarisi .....	16
Şekil 2.7	: Hdfs replikasyon. ....	17
Şekil 2.8	: Küme kaynak yönetimi sağlayan yarn. ....	18
Şekil 2.9	: JobTracker ve TaskTracker yaklaşımı. ....	18
Şekil 2.10	: Yarn mimarisi Resource Manager, Node Manager, Application Master.....	19
Şekil 3.1	: Dört adet Parallella oluşturulan küme görülmektedir .....	23
Şekil 3.2	: Xilinx Zynq-7000 ve adapteva yongası board üzerinde görülmektedir. ....	25
Şekil 3.3	: MapReduce ile WordCount işleminin veri akışı.....	27

## KISALTMALAR

<b>RISC</b>	:	Reduced Instruction Set Computing
<b>CISC</b>	:	Complex Instruction Set Computer
<b>ARM</b>	:	Acorn RISC Machine
<b>CPU</b>	:	(Central Processing Unit) Merkezi İşlem Birimi
<b>IO</b>	:	(Input Output) Girdi/Çıktı
<b>LXC</b>	:	(Linux Containers) Kap temelli Sanallaştırma Uygulaması
<b>RAM</b>	:	(Random Access Memory) Bellek
<b>ROM</b>	:	(Read Only Memory) Yalnızca Okunabilir Bellek
<b>SBC</b>	:	(Single Board Computer) Tek Kart Bilgisayar
<b>CAD</b>	:	(Computer Aided Design) Bilgisayar Destekli Tasarım
<b>PCB</b>	:	(Printed Circuit Board) Baskı Devre Kartı
<b>ANSI</b>	:	(American National Standards Institute)
<b>SDK</b>	:	(Software Development Kit) Yazılım Geliştirme Kiti
<b>USB</b>	:	(Universal Serial Bus) Evrensel Seri Veriyolu
<b>HDFS</b>	:	(Hadoop Distributed File System) Hadoop Dağıtık Dosya Sistemi
<b>RAID</b>	:	(Redundant Array of Inexpensive Disks) Ucuz Sıgalar Artıklıklı Dizisi
<b>FPGA</b>	:	(Field Programmable Gate Array) Alanda Programlanabilir Kapı Dizileri
<b>PC</b>	:	(Personal Computer) Kişisel Bilgisayar
<b>DFSIO</b>	:	(Distributed File System Input Output) Dağıtık Dosya Sistemi Girdi Çıktı
<b>SD</b>	:	(Secure Digital) Güvenli Sayısal
<b>Ref</b>	:	Referans
<b>vb.</b>	:	ve benzeri

## ÖZET

### TEK KART BİLGİSAYAR KÜMELERİ ÜZERİNDE BULUT

AYSAN, Levent

Yüksek Lisans, Bilişim Teknolojileri Anabilim Dalı

Tez Danışmanı: Doç. Dr. İzzet Gökhan ÖZBİLGİN

Ekim 2015, 59 sayfa

Günümüzde bilişim sistemlerinde geçmişe oranla çok daha büyük veriler oluşmaktadır. Bu verilerin depolanması ve analizinde önemli kaynak sorunları yaşanmaktadır. Büyük Verinin depolanması, işlenmesi ve analiz edilmesi için ihtiyaç duyulan sistemlerin, güncel sistemlerden daha hızlı çalışması ve daha az enerji tüketmesi gerekmektedir. Aksi takdirde çok büyük maliyet ve veri analiz süreleri önümüze çıkmaktadır. Bu çalışmada tek kart mini kişisel bilgisayarlar ile küme oluşturup ve üzerinde kap tabanlı sanallaştırma sağlayıp büyük veri algoritmaları denemeleri yapılmıştır. Bu kapsamda oluşturulan büyük veri sistemlerinin temelini oluşturan Map Reduce işlemlerinin özel olarak tasarlanmış Acorn RISC Machine (ARM) işlemci kümeleri üzerinde yürütülmesini ve etkinliğinin test edilmesi araştırılmıştır. ARM işlemcili tek kart mini bilgisayarların maliyeti ucuz, enerji tüketimi düşük, karbon salınımı düşüktür. Tek kart bilgisayar kümelerinin; bulut bilişim, çoklu işlem, paralel işlem ve büyük veri uygulamalarına da uygunluğu da görülmüştür. Tek kart bilgisayar donanımı üzerinde kap temelli sanallaştırma kullanımını denememiş bir yaklaşımdır. MapReduce uygulamasında işçi düğüm olarak işlem tecritlenmesi kullanılması da yeni bir uygulamadır. Bu çalışma ile yapılanlar kap temelli sanallaştırma ve işçi düğümün sanal kullanımının tek kart bilgisayarlar ile sağlanmasını tam olarak karşılamaktadır.

**Anahtar Kelimeler:** tek kart bilgisayar kümesi, sanallaştırma, büyük veri, MapReduce, Hadoop

## **ABSTRACT**

### **CLOUD ON SINGLE BOARD COMPUTER CLUSTERS**

AYSAN, Levent

Master, Department of Information Technology

Thesis Supervisor: Assist Professor İzzet Gökhan ÖZBİLGİN

October 2015, 59 pages

Nowadays information systems have much larger data than in the past. The storage and analysis of this data has a huge lack of resource. Saving, processing and analyzing of the big data needs systems that work faster and consume less energy than current systems. Otherwise much greater costs and times of data analysis will be faced. In this study, a cluster of single board computers is created and succeeded to run process isolation operating system level virtualization for experiencing on big data algorithms. In this context created our work, the Map Reduce transactions, which are the basics of big data systems, were executed on specifically designed ARM architecture mini supercomputer clusters. ARM processed single board computers have effective costs, less energy consumptions and less carbon emissions. Clustering, cloud computing, multiprocessing, parallel processing and big data applications compliance has been also observed. Container virtualization on single board computer is an untested approach to use. Using process isolation for MapReduce WorkerNode is yet another new practice. In this study, the use of container virtualization and virtual nodes usage as MapReduce WorkerNode with single board computers fully meets.

**Keywords:** single board computer cluster, virtualization, big data, MapReduce, Hadoop

## GİRİŞ

Bulut bilişim ilgili cihazlar arasında bağ ve etkileşim ile ortak bir bilgi altyapısı sunma olarak tanımlanabilmektedir. Bulut bilişimin temelini oluşturan büyük veri sistemlerine ait işlemler yüksek güç tüketimine sahip Complex Instruction Set Computer (CISC) tabanlı işlemcilerle sahip donanım kümeleri üzerinde yürütülmektedir. Temelde büyük veri işlemleri göz önünde bulundurularak tasarlanmamış bu donanımlar, bu ve birçok farklı nedenden bu işlemlerdeki başarımları kısıtlamakta ve maliyetleri artırmaktadır. Bulut bilişim üzerinde veri işleminin zaman ve maliyetini düşürüp kaliteyi artırma gereksinimi artmıştır. Bu soruna çözüm için halihazırda mevcut bulunan donanımlar ve yazılımların farklı yapılandırmalar ile biraraya getirilip işlevliliğinin sağlanması gerekmektedir. Bulut bilişim yaygınlaştıkça kurulum maliyeti, toplam sahip olma maliyeti ve işletim maliyeti enerji fiyatları açısından bir sorun oluşturmaktadır. Bulut işlemlerinin önemli bir kısmı analiz işlemleri oluşturduğu göz önüne alındığında maliyeti düşük ve bunun yanı sıra enerji tüketimi, karbon salınımı düşük cihazların kullanılması gereksinimi ortaya çıkmaktadır. ARM işlemcili yüksek hızlı bellek ve veri yoluna sahip tek kart bilgisayarların kullanımının bu gereksinimi çözüm olabileceği görülmektedir. Bu çalışma temel olarak büyük veri işlemlerini düşük biçim katsayısına sahip donanımlar üzerinde işleterek güncel sistemlerden çok daha az enerji tüketerek hızlı bir şekilde gerçekleştirilmesine olanak sağlanmasını amaçlanmaktadır. Enerji tüketimi başına düşen işlem sayısının arttırılması yönüyle başarımların sağlanması hedeflenmektedir. Tek kart bilgisayarlar değişik yapılandırma ve özelliklere sahip olabilmektedir. Kısıtlı bir kısmında yardımcı işlemci ve alanda programlanabilir kapı dizileri ile de yapılandırılmış olabilmektedir. Bu nitelikleri ile tek kart bilgisayarların bulut bilişimde kullanılabilirliği sağlanırsa bir çözüm olabileceği varsayılmaktadır.

MapReduce büyük hacimli verilerin incelenmesini sağlayan bir sistemdir. Veriler işlenirken indirgeme ve eşleme olarak temel iki işlem yapılır. Eşleme için

veriler alınıp parçalanıp işlenir bittikçe sonuçlar toplanır. İndirgeme safhasında biten işler birleştirilerek özet sonuç elde edilmektedir. Eşleme de veri içerisinden istenen ayıklanıp, indirgeme safhasında çekilen veri üzerinde inceleme yapılmaktadır [1].

Ürünün hafif, küçük ve düşük güç tüketimli olması, havacılık alanında insansız hava araçlarının uydu haberleşme, sinyal istihbaratı gibi görev yetilerinin geliştirilmesini olanaklı hale getirmekte ve kolaylaştırmaktadır. Güncel olarak insansız hava aygıtlarının, topladığı ham verinin işlenmesi için yer istasyonuna aktarılması gerekmektedir. Veri iletimi için harcanacak enerji tüketimi aygıtın menziline kısaltmaktadır. Bu nedenle aygıtın görev sonunda yer istasyonuna indikten sonra üzerindeki veriler sunuculara aktarılarak işlenmektedir. Veri iletim hızının düşük kalması, araç üzerinde kapsamlı veri analizini gerekli ve cazip hale getirmektedir. Tek kart bilgisayarların hedeflenen ağırlık, boyut, güç gereksinimi ve performansı göz önünde bulundurulduğunda, bu amaç için kullanımı da uygun olmaktadır.

Bulut yapılmaya elverişliliği, işletim sistemi desteği, açık kaynak kod desteği ve düşük enerji tüketimleri ile kullanımı artan bir eğilim göstermektedir. Paralel dağıtık işlem yapabilmesi nedeniyle sanallaştırma ve bilişim bulut yönünden incelenmektedir. Sunulan tez çalışması dört bölümden oluşmaktadır. Çalışmanın birinci bölümünde bulut bilişim ve bunun temelinde yer alan dağıtık hesaplama ile sanallaştırma kavramlarından bahsedilmektedir. İkinci bölümde uygulamanın bileşeni olan tek kart bilgisayarlar ve Adapteva Parallella modelinden, işletim sistemi olarak linux ve lxc kullanımı anlatılmaktadır. Tüm bu donanım, işletim sistemi ve kap sistemi üzerinde çalışan MapReduce işlerini yapan Hadoop ile iş dağıtımı, dağıtık dosya sistemini yöneten yan Hadoop bileşenleri aynı bölümde yer almaktadır. Tüm bu sistemin ayağa kaldırıp bulut üzerinde yapılan çalışmalar ve çalıştırılan programlar ile görüntülenen çıktılar üçüncü bölümde yer almaktadır. Elde edilen sonuçların değerlendirilmesi ve önerilen çalışmalar ise dördüncü olarak son bölümde yer almaktadır.

## BİRİNCİ BÖLÜM

### LİTERATÜR VE KAVRAMLAR

#### 1.1 Bulut Bilişim ve Bilgisayar Kümeleri

Bulut Bilişim, ağ üzerinden etkileşim ile kolaylıkla yönetilebilen çeşitli bilişim kaynakları sunulması modelidir [2]. Bilgisayar Kümeleri, tek bilgisayar gibi işlem yapabilen bilgisayar topluluklarıdır. Amaç, kullanıcıların ya da çalışan işlemlerin kümenin iç yapısından bağımsız olması ve kümenin her durumda istenilen işi en verimli şekilde çalıştırmaktır.[3] Bir bilgisayar kümesi içindeki bilgisayarlar, çoğunlukla birbirine hızlı bir ağ bağlantısı ile bağlanmaktadır. Bilgisayarların üzerinde çalıştırılan özel yazılımlar ile istenen işin paylaşılması hedeflenmektedir.

Kümeleme işlemi verimlilik, yüksek hız, ağ bağlantısı kesintileri, veri kaybını önleme gibi nedenlerle tercih edilmektedir [3]. Değişik amaçlar için değişik özelliklerde bilgisayar kümeleri oluşturulabilmektedir [4-5]. Bir çok süper bilgisayar kümeleme ile kullanılmaktadır. (TOP500 Cluster list) Japonya'daki Hesaplamalı Bilim için Gelişmiş Bilgisayar Bilimi Enstitüsü, K bilgisayar, Fujitsu SPARC64 VIIIfx 2.0GHz kümeleme kullanmaktadır [6].

#### 1.2 Paralel ve Dağıtık Hesaplama

Paralel hesaplama, bir işlemin sonuçlarını daha kısa sürede bulmak için çoklu işlemcilerde eş zamanlı olarak çalıştırılması mantığı olup çözümün kolaylaştırılması ve zaman kazanılması için daha küçük işlem parçalarına ayrılması ve eş zamanlı olarak sonuçlandırılması fikrine dayanmaktadır [7]. Paralel hesaplama ile performans artar, büyük sorunlar daha az sürede çözülür [8]. Bilimdeki gelişmeler paralel hesaplama gereksinim duymaktadır [9].

Paralel programlama modelleri sayesinde programın bağımsız parçalara ayrılabilen her bir kısmı farklı işlemci üzerinde eş zamanlı olarak işletilir [10].

Çoklu işlemcilerin tek başına bir işlemciden daha hızlı olması beklenir. İşlemci tasarımında oluşan zorluklar çoklu işlemcileri zorunlu kılmaktadır. Grace Hopper bu konu ile ilgili olarak "Eğer bir öküz işi yapamıyorsa, öküzü büyütme yerine iki öküz kullandılar" demiştir [11]. Bir araya getirilen işlemciler arasında iletişim ve fiziksel bağlar vardır.

### **1.3 Sanallaştırma**

Sanallaştırma bilgisayar kaynaklarının kullanıcılardan soyutlanmasıdır. Sanallaştırma birçok faydası olması nedeniyle hızlı yayılmıştır.

1. Kolay yönetilebilirlik
2. Uyum sorununun ortadan kalkması
3. Hata tespiti ve izolasyonunda diğerlerinin etkilenmemesi
4. Arttırılmış güvenlik ile erişim kontrolü sağlanması
5. Kaynakları etkin kullanmayı sağlanması
6. Verilerin taşınabilirliğinin kolaylaştırılması
7. Sorunsuz testler ve test ortamları sağlanması
8. Hızlı dağıtım kolaylığı
9. Maliyetlerin azalması ve uygulamaları ayırmak

vb. faydaları nedeniyle kullanımı artmakta ve yeni teknolojiler ile de daha da güçlü hale gelmektedir [12]. Uygulama, masaüstü, sunucu, depolama ve sunum sanallaştırmaları başarıyla sağlanmaktadır. Donanım bağımsızlığı sağlanması, tek merkezden tamamını yönetebilme, fiziki sunucu sayısını azaltma gibi avantajları nedeniyle veri merkezlerinde tercih edilir olmaktadır. Bir donanımın tüm özelliklerini sanal hale getirmek yerine işlemleri işletim sistemi seviyesinde tecritleyerek bulut bilişimde yeni bir akımdır. Microsoft windows ortamlarında kum havuzu (sandbox), unix tarafında ise kap tabanlı (container) sanallaştırma adı verilmektedir. Uygulama ve bağımlılıklarının paketlenip kap tabanlı sanallaştırılması açık kaynak projesi Docker.com bu konuda ilgi çeken bir çalışmadır[13].



## İKİNCİ BÖLÜM

### UYGULAMA BİLEŞENLERİ

Uygulamamızda donanım olarak tek kart bilgisayar seçilmesinde düşük enerji tüketimi ve kümelemeye uygunluk göz önüne alınmıştır. Maliyet ve küçük boyut ise tek kart bilgisayarların tamamının haiz olduğu niteliklerdir. Alanda programlanabilir mantık devreleri içermesi ve epiphany yardımcı işlemci içeren donanım olması nedeniyle Adapteva Parallella öne çıkmıştır. Nesnelerin interneti alanında toplanacak verilerin işlenmesi konusunda gelişmeye açık bir donanım olması akademik çalışmalara yatkınlığı bir diğer husustur.

#### 2.1 Tek Kart Bilgisayar Donanımları

Tek kart bilgisayar özünde bir devre kartı üzerine mikroişlemci, bellek, giriş ile çıkış birimleri ve bir bilgisayar için gerekli diğer işlevsel özellikleri içermekte olan bilgisayardır. Tek kart bilgisayarlar eğitim sistemleri, gösteri öğretim ve geliştirme sistemleri ve ya sanayi çözümleri için gömülü bilgisayar sistemleri olarak tasarlanmaya başlanmıştır. Ev bilgisayarı veya taşınabilir bilgisayarın birçok türlerine benzemekte olup tek bir baskılı devre kartı üzerine tüm fonksiyonlarını karşılayacak donanımlar entegre edilmiştir.

Bir masaüstü kişisel bilgisayarın aksine tek kartlı bilgisayarlar genellikle çevresel işlevleri veya genişleme yuvaları yönünden kısıtlı sayıda donanıma haizdir. Tek kartlı bilgisayarlar geniş bir mikroişlemci yelpazesini kullanarak inşa edilmiştir. Bilgisayar ile ilgilenen çeşitli topluluklar tarafından açık kaynak donanım tasarımları ile basit devreler ile hazırlanılmakta olmalarına rağmen büyük sunucu bilgisayarların tüm bellek ve işlemci performansına yakın değerler sağlayabilen kümeler kullanılmaya yeni başlanılmaktadır. Taşınabilir boyutlu yerden tasarruf sağlayan

düşük maliyetli olmalarına yanı sıra harcadıkları güce nazaran işlemci saat vuruş sayıları ile tercih nedeni olmaktadır.

İngiltere Raspberry Pi Vakfı tarafından temel öğretimde bilgisayar bilimini okullarda teşvik etmek amacı ile Raspberry Pi adında kredi kartı ölçüsünde tek kartlı bilgisayarların üretilmesiyle bu tarz ürünler daha da ün kazanmıştır [14].

Mikro denetleyicilerin aksine Beaglebone, Beageboard, Raspberry pi ve Intel Galileo gibi gömülü sistem kartları işletim sistemi ile kullanılmaktadır. Mikro denetleyicinin gücü ve yapabilecekleri sınırlı iken tek kart bilgisayarlar ve benzeri gömülü sistemler için bu durum daha farklılaşmaktadır. Örneğin; mikro denetleyicilerde bellek olarak kendi iç yapısında ROM ve Flash kullanılırken tek kart bilgisayarlarda durağan disk ve ARM Cortex 'in ROM belleğini kullanılabilir.

Yüksek performansa sahip ucuz ve hafif donanımlar ile çalışmak isteyenler için benzeri bir çok kart piyasada yer almaktadır. Örneğin; Chip marka 9\$ tek kart pc gibi çok daha ulaşılabilir ürünler piyasaya girmektedir [15].

### 2.1.1 Adapteva Parallela Tek Kart Bilgisayar

Parallela tek kart bilgisayarlar açık kaynak donanım olarak Adapteva firması tarafından üretilmektedir. Mimari olarak Xilinx Firmasının Zebdboard ürününü temel almaktadır. Tablo 2.1'de özellikleri gösterilmektedir.

**Tablo 2.1:** Parallela özellikleri.

Entegre Üzeri Sistem:	Zynq 7010 / 7020
İşlemci:	800 MHz ÇİFT ARM® Cortex
Ağ Tümeleşik:	10/ 100/ 1000 kablolu RJ45 ağ
FPGA genişleme:	Farklı sinyal standartlarının bir dizi desteklemek için Zynq cihaz içinde yapılandırılabilir 48 çift yönlü sinyaller içermektedir. Sinyaller olarak yapılandırıldığında, her diferansiyel sinyal çifti 950Mbps maksimum bant genişliği sağlar. IO bant genişliği 22Gbps sağlayabilir.
Güç kaynağı:	5 V (DC) 2A
Boyutu:	90mmx 55mmx 18mm
Ağırlığı:	36 gr

Tabloda görüldüğü üzere çok çekirdekli yongaların yüksek performansını, kredi kartı büyüklüğünde bir boyut ile kullanıcıya sağlamaktadır. Parallella bağımsız bir bilgisayar, gömülü bir şekilde ya da ölçeklenebilir paralel sunucu kümesinin bir bileşeni olarak da kullanılabilir. Kümeleme yapmaya elverişli olması da tercih nedeni olmaktadır.

Parallella tek kart bilgisayar açık kaynak donanım olup devre tasarım dosyaları Creative Commons Attribution-ShareAlike 3.0 lisansı altında yayınlanmaktadır[16]. Parallella tek kart bilgisayar açık kaynak olması ve temel özellikleri ile göz önüne alındığında paralel hesaplama alanı için ideal bir donanım olmaktadır.

Açık kaynak donanım olması nedeniyle tasarım bilgileri ve parallella projesi incelenebilmesi, değiştirilebilmesi ve tasarıma katkıda bulunulabilmesi için gerekli tüm dosyaları sağlamaktadır [17].

Sağlanan tasarım kaynakları şunlardır:

1. Tüm referans el kitabı
2. OrCAD biçiminde şematik kaynakları
3. PCB düzeni kaynakları, üretim dosyaları yaygın dosya biçimlerindedir.
4. Montajlı kurulu üç boyutlu CAD modeli
5. Malzeme fiyatları ile üretim maliyet tahminini kolaylaştırır

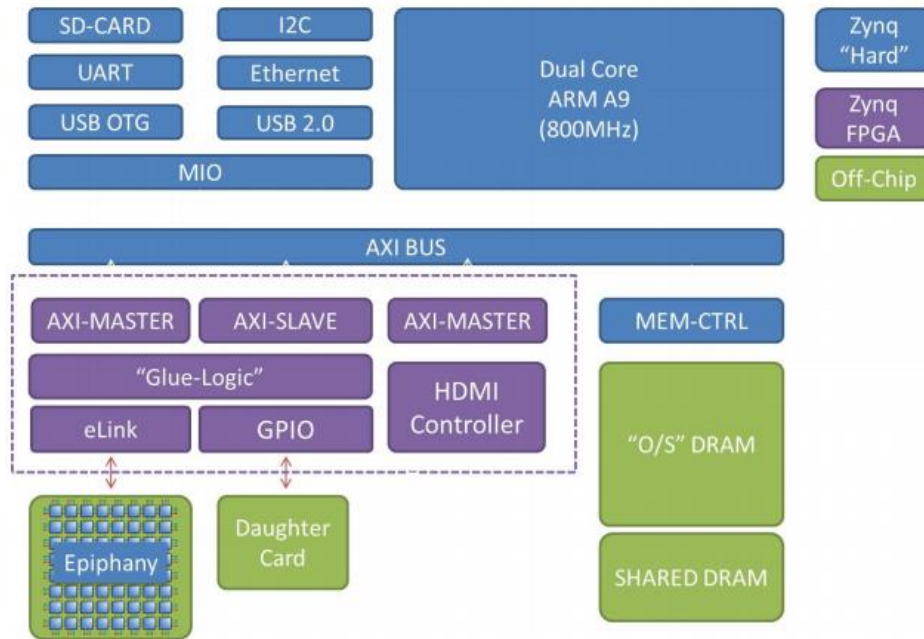
Tüm bu tasarım dosyaları güncel haliyle internet ortamında serbest paylaşımına sunulmaktadır [18]. Parallella projesi için açık kaynak tasarım Arduino, Beaglebone, Zedboard, Raspberry Pi projelerinden faydalanılmıştır [19].

Temel Özellikleri şu şekildedir

1. Açık Kaynak donanım ve yazılım ile istisnasız herkesin erişilebileceği çözüm sağlanmaktadır.
2. Basit ve düşük maliyetli entegrasyon
3. Esnek kümeleme ve kaynak kullanılabilirliği
4. Düşük maliyetli, düşük güç çekirdekli platformu
5. Kredi kartı boyutu bilgisayar
6. Çift çekirdek ARM ve 16/64 tabaka yardımcı işlemciler
7. C / C ++ ve OpenCL programlanabilir
8. 90 GFLOPS kadar işleme yeteneği ki bu özelliği ile öne çıkmaktadır
9. Zynq-Z7010/Z7020 Çift çekirdekli ARM A9 işlemci
10. 4 genel amaçlı genişleşme bağlantı yuvaları

Parallella bilgisayarların ilk yatırım maliyetleri de Kickstarter kitle fonlama kampanyaları ile açık kaynak olarak finanse edilmektedir [20]. Kickstarter kampanyası Ekim 2012 tarihinde dünyadan birçok destekçileri aracılığıyla 898.921 \$ toplanarak üretime geçmeyi başarmıştır. Herkes için süper bilgisayar gücünü 99 \$ fiyatına sağlamak ana fikriyle hayata geçirilmiş bir projedir. Açık Erişim, Kaynak ve Ekonomik olma hedefleri vardır. Açıkça web üzerinde yayınlanan tüm mimari ve SDK belgeleri - özel izin ve erişim yetkisi istememektedir. Ücretsiz ve açık kaynak yazılım geliştirme araçları ve kütüphaneler dayalı yazılım platformu desteklenmektedir. Parallella bilgisayar açık kaynak donanım lisansı altında sağlanmaktadır ve tüm tasarım dosyaları da açıktır. Amaç herkes için uygun bir platform olması için 100 \$ altında bir donanım maliyeti getirmektir.

Parallella en fazla 5w güç tüketirken, çift çekirdekli ARM A9 işlemci içeren ve Ubuntu, Debian, Linaro, OpenSuse, Archlinux ARM gibi Linux dağıtımları ile de çalışabilmektedir. Xilinx firmasının açık kaynak dünyasına teknoloji desteği sayesinde, teknolojik yenilikler alınabilmektedir [21]. İlerleyen süreçlerde altyapı olarak yeni nesil Zynq yonga üzeri sistem kullanmak için yükseltilmiş olacaktır. Parallella merkezi işlemci ARM çift çekirdekli Cortex işlemci ürünü Xilinx 28nm programlanabilir mantık ile işleme sistemli Zynq yonga üzeri sistem kümesini içeriği Şekil 2.1’de gösterilmektedir.

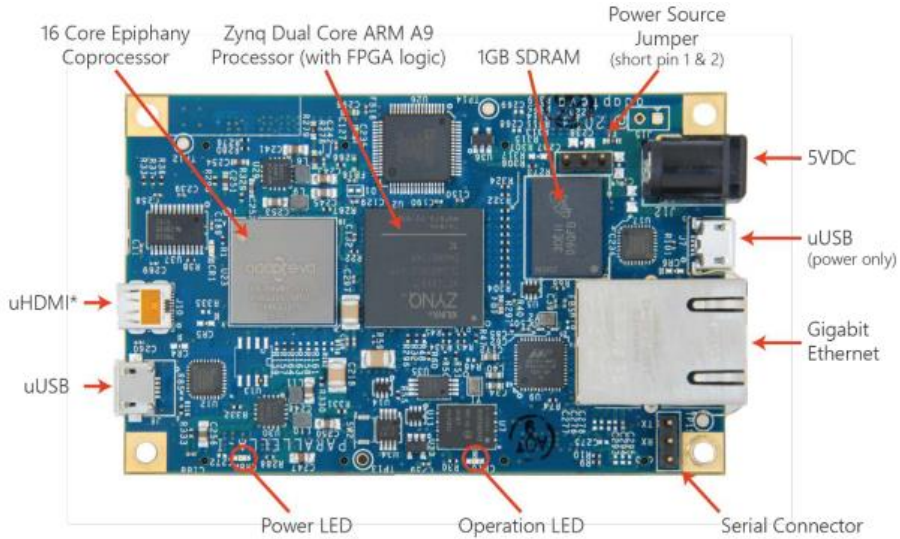


Şekil 2.1: Parallella tek kart bilgisayar üst seviye mimarisi [21].

Şekilden görüldüğü üzere belleğin bir kısmını kendi işletimi için kullanırken bir kısmını paylaşmaktadır. Bu yapının temel özellikleri şunlardır:

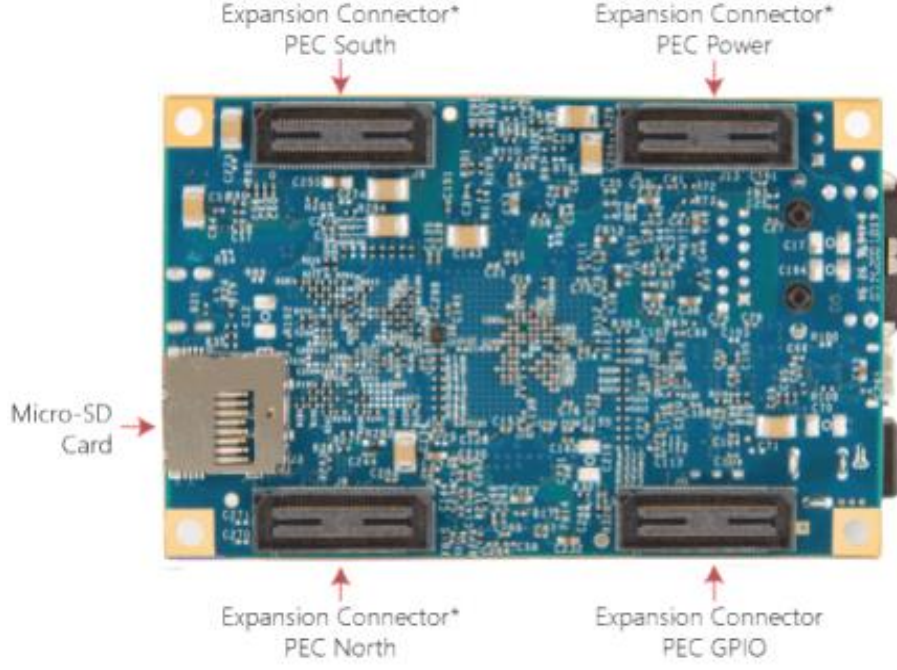
1. Çift çekirdekli Cortex ARM™ -A9 İşlemci:
2. Çoklu işlemci desteği
3. 32 KB Seviye 1 4 yollu set-ilişkisel talimat / (her işlemci için bağımsız) veri önbelleklerini ve 512 KB 8-yollu set-ilişkisel işlemciler arasında paylaşılan Düzey 2 önbellek
4. Tek ve çift hassasiyetli Vektör Kayan Nokta Birimi
5. Üç Watchdog zamanlayıcı, tek küresel zamanlayıcı, iki üçlü zamanlayıcı sayaçları

Epiphany yardımcı işlemci yongaları C / C++ veya OpenCL, MPI ve OpenMP gibi paralel programlama için basit bir kullanım sağlamaktadır. RISC işlemciler ile programlanabilir ölçeklenebilir dizi oluşmaktadır. Bağımsız Epiphany çekirdek örgü dağıtılmış paylaşılan bellek mimarisi içinde kart iç ağında hızlı bir şekilde birbirine bağlanır. Şekil 2.2 ile Parallella üst görünümünde ve sol tarafta Epiphany yardımcı işlemci ile yanında yer alan Arm A9 işlemci yer almaktadır.



Şekil 2.2: Parallella tek kart bilgisayar üst görünüm.

Şekilde görüldüğü üzere ses, görüntü, ağ, güç girişleri ile kartın giriş çıkış bağlantı yerleri dış kısımlarda yer almaktadır. Şekil 2.3'de Parallella alt görünümünde dört adet genişleme bağlantı yuvası görülmektedir.



Şekil 2.3: Parallella tek kart bilgisayar alt görünüm.

Şekilde görüldüğü üzere bellek girişi de altta kızımda yer almaktadır. Bu bağlantılar ile girdi çıktı cihazları eklenebileceği gibi mimariye uygun kullanıcıların kendi harici donanımlarını tasarlayarak birleştirilmesinin yolu açılmaktadır. Değişik hedef uygulamalarına göre birleştirme esnekliği bu yolla sağlanmaktadır.

### 2.1.2 Parallella Hedef Uygulamaları

Adapteva tarafından nitelikleri ile Parallella şu alanlarda kullanımı önerilmektedir [22].

Tüketici:

1. Akıllı telefonlar ve tablet uygulaması ivme
2. Yüksek ses
3. Hesaplamalı fotoğrafçılık
4. Konuşma Tanıma
5. Yüz algılama / tanıma

Askeri ve Havacılık:

1. Radar / Sonar
2. Aşırı Büyük Algılayıcı Görüntüleme
3. Hiperspektral Görüntüleme
4. İletişim Karıştırma

5. Askeri Telsizler
6. Cephane / Rehberlik

Tıp:

1. Ultrason
2. Bilgisayarlı Tomografi

İletişim:

1. Haberleşme test yatağı
2. Yazılım tanımlı radyo
3. Uyarlanabilir Ön frekans

Endüstri / Aletler:

1. Yapay Görme
2. Özerk Robotlar
3. Otomotiv Güvenlik
4. Yüksek Hızlı Veri Toplama / Üretimi

Diğer:

1. Sıkıştırma
2. Güvenlik Kameraları
3. Video İşlemleri

Aslen bu alanlarda öne çıkan ortak özelliğin işlemci gücü, analiz yeteneği, yüksek hız gerektiren sinyal ve veri işleme yetenekleri olduğu ve tüm bu yeteneklerin bulut tarafında da gerek olduğu ortadadır. Yoğun veri analizi, yazılım tanımlı ağ, veri merkezi aleti ve yüksek frekans işleme aygıtı olarak kullanıma uygunluğu donanım olarak tercih nedeni olmaktadır.

## **2.2 Linux İşletim Sistemi**

Linux açık kaynak koduna sahip bir işletim sistemidir. Unix'e fikirselsel ve teknik anlamda atıfta bulunarak geliştirilmiş açık kaynak kodlu, özgür ve ücretsiz (destek hariç) bir işletim sistemi çekirdeğidir [23]. Çok sayıda ve çeşitte donanıma destek verir. Linux sunucu işletim sistemlerinde kullanım oranı bakımından ilk sırada tercih edilmekte ve dünyanın 10 en hızlı süper bilgisayarında da linux işletim sistemi kullanılmaktadır [24]. Kasım 2010 tarihinden beri en iyi 500 sistem arasında 485'i (%97) Linux dağıtımını kullanmaktadır. Ayrıca Tablo 2.2'de özellikleri görülmekte

olan dünyanın en güçlü süper bilgisayarı olan Guangzhou Ulusal Süper Bilgisayar Merkezi için de işletim sistemi olarak kullanılmaktadır [24].

**Tablo 2.2:** TIANHE-2 TH-IVB-FEP küme yapısı.

Site: Guangzhou Ulusal Süper Bilgisayar Merkezi	
Üretici:	NUDT
Çekirdekler:	3120000
Linpack Performansı (Rmax)	33, 862.7 TFlop / sn
Teorik Tepe (Rpeak)	54, 902.4 TFlop / sn
Güç:	17, 808.00 kW
Bellek:	1.024.000 GB
İşlemci:	Intel Xeon E5-2692v2 12C 2.2GHz
İşletim Sistemi:	Kylin Linux
MPI: Intel MKL-11.0.0 Matematik Kütüphanesi:	Özelleştirilmiş bir GLEX kanalı ile mpich2

Linaro ARM tabanlı işlemci ve kart üreten/pazarlayan firmaların bir araya gelip ürettikleri donanımların üstünde, işletim sistemi, işletim sistemi çekirdeği ve yan yazılım ihtiyaçları için ortak çözüm oluşturmayı amaçlayan bir organizasyon olup kar amacı gütmemektedir [25]. Android ve Ubuntu Linux üzerinde değişiklik ve uygunluk çalışmaları yaparak bunları sağlamaktadırlar. Yeni çıkan ARM kart üzeri işlemcileri destekleyen linux çekirdek ve bootloaderlar oluşturmak, çoklu ortam, güç yönetimi, grafik ve güvenlik desteği ve bu tarz yazılım parçalarını oluşturmak için gerekli uygulama parçalarını da bizzat derlemek, hazırlamak, uyum testlerini yapmak gibi çalışmaları mevcuttur. ARM komut setlerinde 32 bit ve 64 bit çalışan Cortex işlemci ailelerine destekleri geliştirilmiştir [25].

### 2.3 İşletim Sistemi Seviyesinde Sanallaştırma

İşletim sistemi çekirdeğinin aralarındaki izolasyonu sağlayarak birden fazla kullanıcı alanı yaratabilme yeteneğine işletim sistemi seviyesinde sanallaştırma adı verilmektedir [26]. İşletim sistemi seviyesi sanallaştırma, bir bilgisayar üzerinde



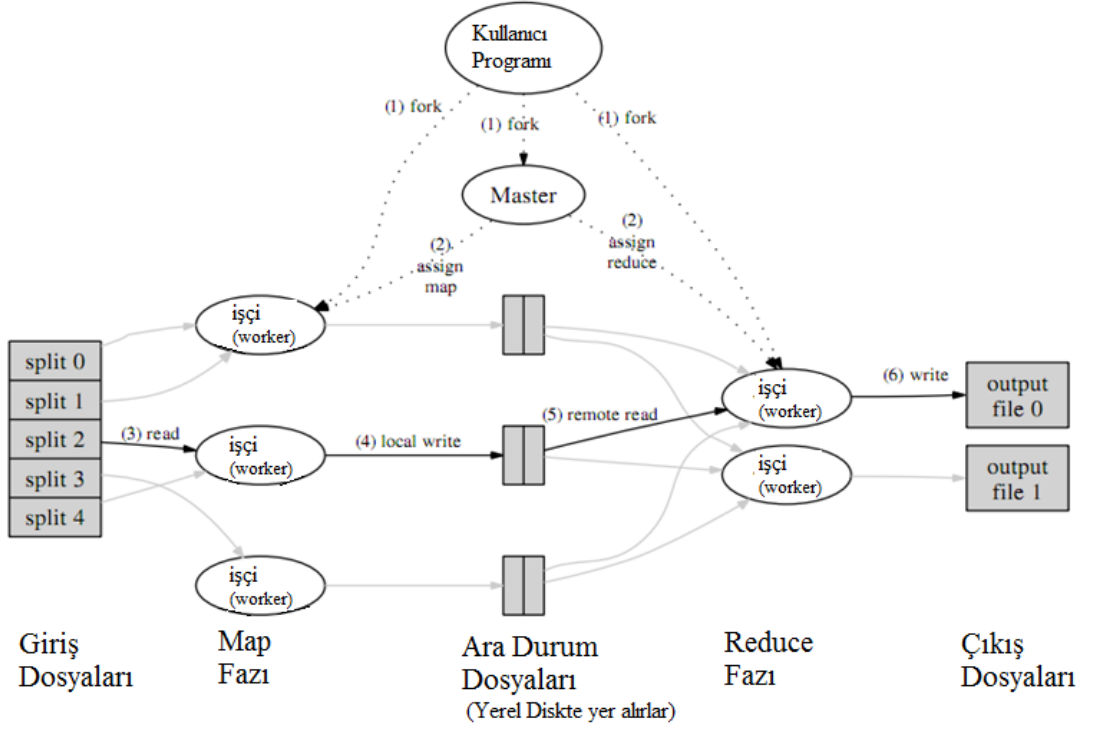
çalışmakta olan birçok uygulamanın, aynı fiziksel donanım üzerinde çalışıyor olmalarına rağmen, birbirleri ile hiçbir şekilde etkileşemeyecek şekilde yalıtılmış olmaları ile sağlanmaktadır. En yaygın olarak kullanılanı LXC'dir (Linux Container). Her bir sanal sistem yani kap, altyapıda aynı işletim sistemini kullanan farklı uygulamalardan oluşmaktadır.

LXC kolay kurulabilen, sistem üzerindeki yükü hafif, kaynak paylaşımına ve kotalar oluşturulmasına olanak veren, farklı Linux dağıtımlarının birlikte çalıştırılmasını sağlayan, yönetimi için web tabanlı bir kullanıcı arayüzü sunan, çok aktif olarak geliştirilen bir sanallaştırma teknolojisidir. Benzer şekilde diğer sanallaştırma teknolojileri ile karşılaştırıldığında kurulumu ve kullanımı açısından daha basit ve sistem üzerindeki ek yükü çok az olmaktadır. LXC kontrol grupları ile kaynak yönetimi ve çekirdek isim alanları ile kaynak işlem kaplandırılması sağlamaktadır [27]. Sistemde yürütülen işlemlerin bir bölümünün diğerlerinden tamamen yalıtılması suretiyle çalışmaktadır.

LXC adında geçmekte olan kap kavramı bir işlem grubunun, aynı sistem üzerinde birlikte çalıştıkları diğer işlemlerden ayrılmasını sağlayan yöntem verileri isimdir. Her bir kap diğerlerinden tamamen bağımsız bir yapıdır ve bu halleri ile ayrı birer fiziksel sunucu gibi kullanılabilirler. Her biri ayrı bir dosya sistemi yapısına, ayrı kullanıcılara ve kullanıcı gruplarına, ayrı sanal ağ cihazlarına ve en önemlisi ayrı bir işlem ağacına sahiptir. Ayrıca her bir kap ayrı bir disk kullanım limitine, işlemci kullanım yüzdesine sahip olabilmektedir.

## **2.4 MapReduce**

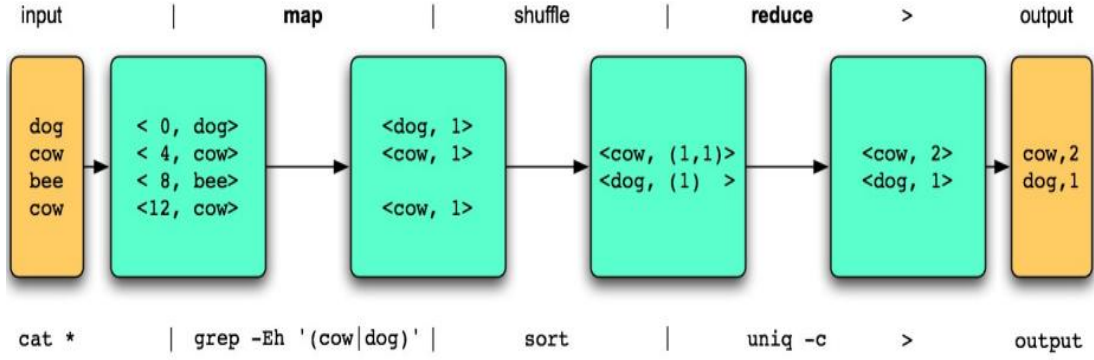
MapReduce büyük hacimli verilerin incelenmesini sağlayan bir sistemdir. Veriler işlenirken indirgeme ve eşleme olarak temel iki işlem yapılır. Eşleme için veriler alınıp parçalanıp işlenir bittikçe sonuçlar toplanır. Reduce aşamasında ise tamamlanan işler birleştirilerek sonuç elde edilmektedir. Eşleme de veri içerisinden istenen ayıklanıp, indirgeme safhasında çekilen veri üzerinde inceleme yapılmaktadır [28]. 2004 yılında Google tarafından duyurulan bu sistem aslen 1960'lı yıllarda geliştirilen fonksiyonel programlamadaki map ve reduce fonksiyonlarından esinlenmiştir [29]. Veriler işlenirken map ve reduce işlemleri yapan iki fonksiyon kullanılmaktadır.



Şekil 2.4: MapReduce fiziksel işleyişi.

Map aşamasında ana düğüm verileri alıp daha ufak parçalara ayırarak işçi düğümlere dağıtmaktadır. Şekil 2.4 ile MapReduce işleminin fiziki olarak işleyişi görülmektedir. İşçi düğümler bu işleri tamamladıkça sonucunu ana düğüme geri göndermektedir. Reduce aşamasında ise tamamlanan işler işin mantığına göre birleştirilerek sonuç elde edilmektedir. Map aşamasında analiz edilen veri içerisinden almak istenen veriler çekilmekte, Reduce aşamasında ise bu çekilen veri üzerinde istenilen analiz gerçekleştirilmektedir [30].

Büyük veriyi işleyebilmek için çok yüksek donanımlı sunucular kullanmak yerine sıradan sunuculardan oluşan bir küme üzerinde MapReduce yardımıyla aynı işlem çok daha etkin bir şekilde gerçekleştirilmektedir. Sıradan veritabanı sistemleri çok büyük verilerin işlenebilmesi yüksek fiyatlı donanım ve yazılım ile mümkün iken, MapReduce ise buna düşük fiyatlı çözümlere kapı açmaktadır.



Şekil 2.5: MapReduce mantıksal işleyişi.

Şekil 2.5 ile görüldüğü üzere MapReduce sisteminin mantıksal işleyişinde girişte yer alan dosyanın eşlenip indirgenme aşamaları düğümlere iş dağıtımı ile sağlanmaktadır. Girişte alınan dosya eşlenip indirgenerek çıktı verilmektedir.

### 2.4.1 Hadoop

Hadoop, küme üzerinde büyük verileri işlemek amaçlı uygulamaları çalıştıran ve Hadoop dağıtık dosya sistemi ile Hadoop MapReduce özelliklerini bir araya getiren, Java ile geliştirilmiş açık kaynak kodlu bir yazılım kütüphanesidir [31].

Hadoop, arka plandaki kümelenmiş bilgisayarların üstüne kurguladığı uygulama seviyesindeki çözümüyle farklı, ucuz bir alternatif olmayı hedeflemektedir. Hadoop Projesi aşağıda belirtildiği üzere dört modülden oluşmaktadır [32].

Hadoop Common: Hadoop ortak modülleridir.

Hadoop Hadoop Distributed File System (HDFS): Dağıtık Dosya Yönetim Sistemidir. Farklı cihazlardaki dosya sistemlerini tek bir dosya sistemi gibi gözükmesini sağlar.

Hadoop YARN: İş zamanlayıcı (job scheduler) ve kaynak yönetimini yapan bir dizi kütüphanedir [33].

Hadoop MapReduce: YARN temelli, büyük miktarda veriyi paralel olarak işlemeye bilgisayar düğüm noktalarına bu iş yükünü atama imkanı sağlamaktadır.

Hadoop MapReduce Özellikleri

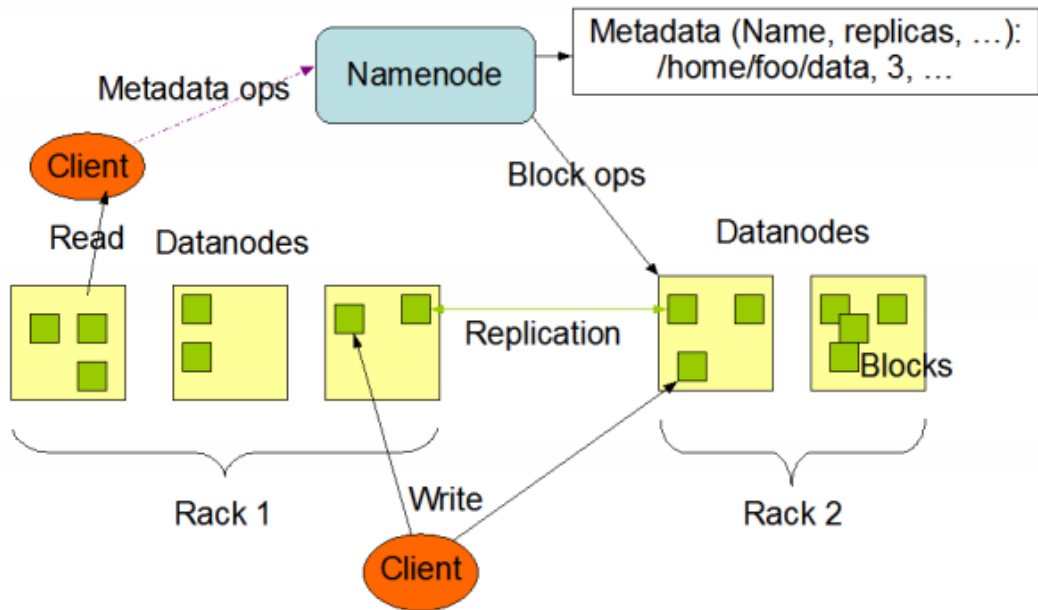
1. Büyük miktarda veriyi parçalar halinde paralel işleyebilmektedir.
2. Basitleştirilmiş anahtar/değer arayüzü ile herkesin dağıtık algoritma yazabilmesini sağlamaktadır.
3. Toplu işlemi canlı akan bilgi işlemlerinde yapabilmektedir.

4. Hata toleranslıdır.
5. Yüksek bilgi yerelleştirmesi sağlamaktadır.
6. Anahtar/değer tabanlı Map/Reduce modeli veri madenciliği / makine öğrenmesi/ web gezinimli bilgi toplama gibi pek çok alana uygundur.

Hadoop, özellikle büyük hacimli verinin işlenmesinde, ölçeklenebilirlik, süreklilik, hatadan kurtarma, paralel işlem yetenekleri, fiyat/performans, esnek tasarım gibi parametrelerin hepsinin aynı anda gerçekleşmesini sağlayabilir [34-35].

### 2.4.2 Hadoop HDFS

HDFS kullanılarak sıradan sunucuların diskleri tek bir disk gibi davranırlar. Çok büyük boyutlu dosyaların barındırılmasında etkindir. Değişik yapılandırma ayarları ile RAID tarzında yedekleme diskler üzerinde kopya barındırılarak sağlanır. Böylece veri kaybının önüne geçilir. NameNode ve DataNode gibi düğümlerin çalışması ile gerçekleştirilen işlemlerin mimarisi Şekil 2.6'da görülmektedir.

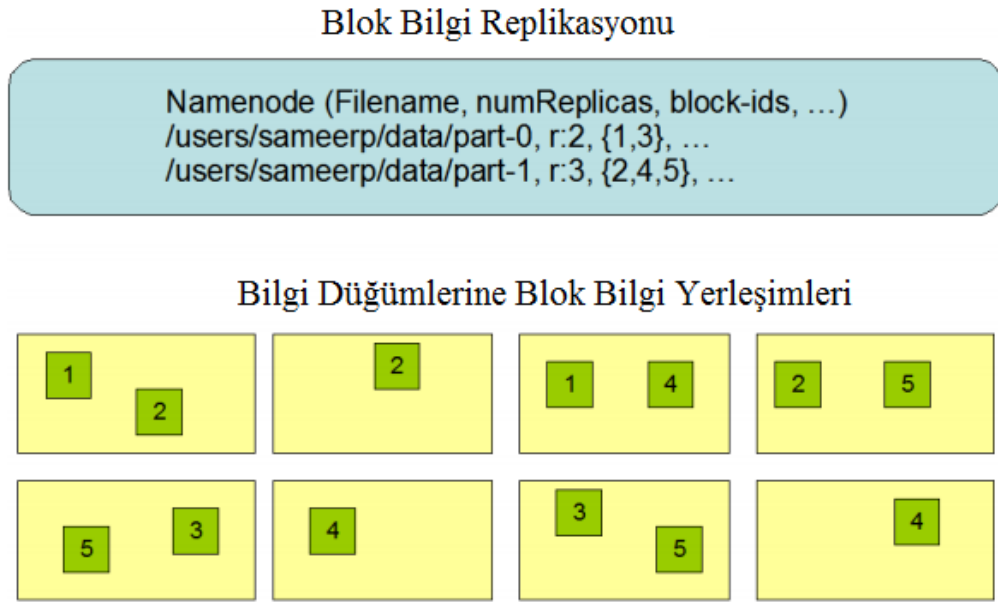


Şekil 2.6: Hdfs mimarisi [36].

Şekil 2.6'da görüldüğü üzere blokların sunucular üzerindeki silinmesinden ve yeniden oluşturulmasından NameNode sorumludur. Bir blokta sorun meydana geldiğinde yeniden oluşturulmasından ve her türlü dosya erişiminden sorumludur.

Kısacası HDFS dosyalarının üst bilgileri saklanması ve yönetilmesinden NameNode sorumludur. NameNode her bir kümede bir tane vardır.

DataNode işçi süreç olarak blokları oluşturur. Her DataNode ait olduğu öz diskindeki veriler ile ilgilenir. Ancak farklı DataNode verilerinin yedeklerini de görev olarak saklar. Her bir kümede bir çok DataNode mevcuttur. Şekil 2.7 Blok bilgilerinin yedeklenmesi ve DataNode bilgi düğümlerinde yerleştirmeyi göstermektedir.

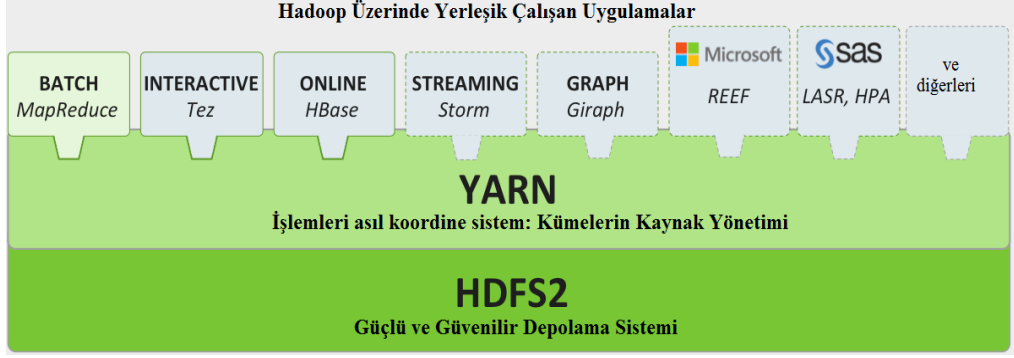


Şekil 2.7: Hdfs replikasyon.

Hadoop dağıtık dosya sisteminde yedeklenme ve tekrarlanma parçalı dağıtık olarak bilgi düğümlerinde yapıldığı her düğümde farklı bir tekrarlama olabileceği Şekil 2.7’den de rakamların düğümlere dağılımı olarak temsili olarak gösterilmiştir.

### 2.4.3 Hadoop YARN

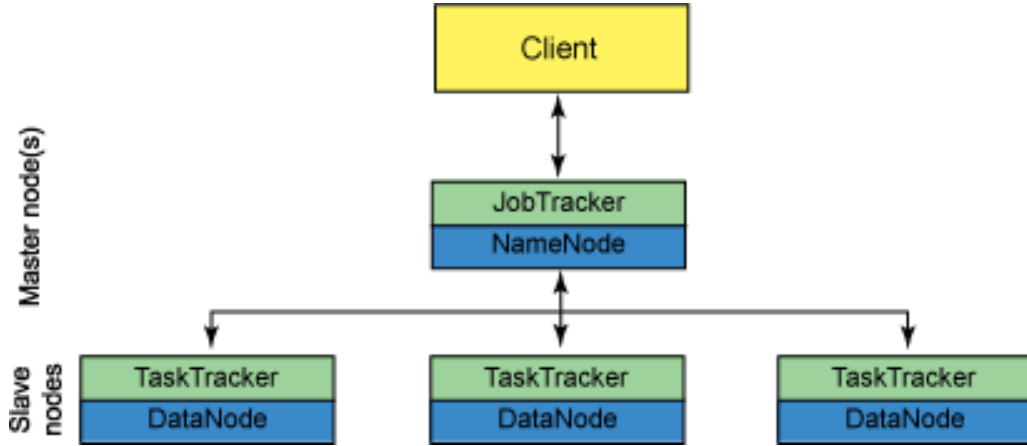
Hadoop işlerini zamanlamak ve küme kaynaklarını yönetmek için kullanılan kütüphanedir. Hadoop 2.0 ile birlikte gelmiştir. Hadoop mimarisinin merkezinde bulunan YARN, tüm iş yükleri için tek bir erişim noktası tanımlayarak her türlü veri işleme framework’leri için tutarlı paylaşım servisleri ve kaynakları tanımlamaya yardımcı olmaktadır [37].



Şekil 2.8: Küme kaynak yönetimi sağlayan yarn.

Hadoop YARN dosya yönetim sistemi üzerinde yer alarak dağıtık uygulamaları çalıştırmak için kullanılan kaynakların yönetimini sağlamaktadır. Şekil 2.8’de belirtildiği üzere küme kaynak yönetimi sağlayan Yarn HDFS ile diğer yerleşik çalışan uygulamaların arasında asıl koordinasyonu sağlamaktadır.

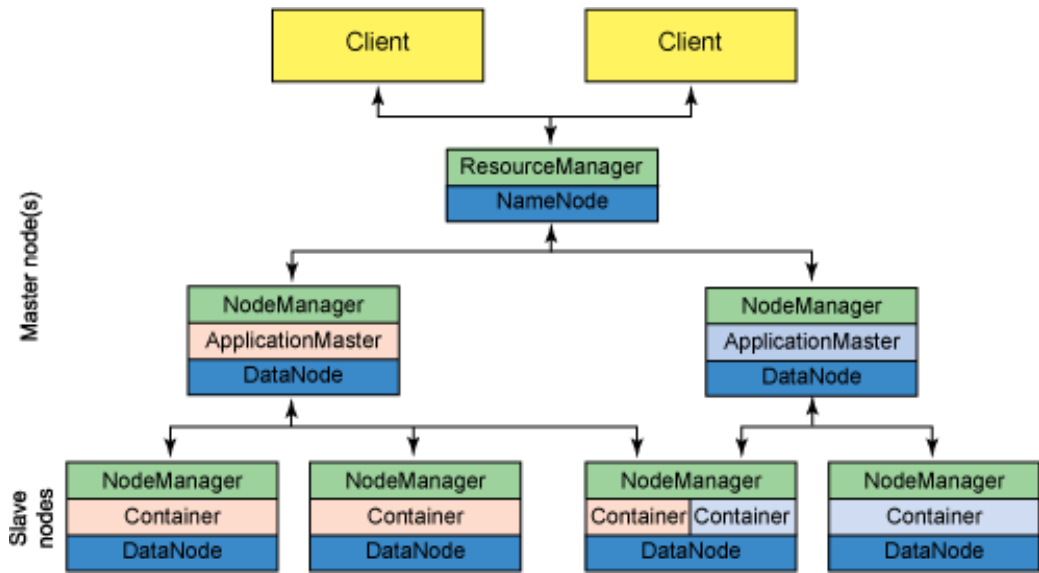
Yarn temel fikri, JobTracker ve TaskTracker iki sorumluluğu ayrı görevler haline getirme isteği ile başlamıştır. Dikeyde üstten alınan istekleri JobTracker karşılarken bunları TaskTrackerlara yönlendirerek çalışmasını sağlamaktadır. Şekil 2.9’da JobTracker efendi olarak TaskTracker yapısının ise köle olarak konumlanması görülebilmektedir.



Şekil 2.9: JobTracker ve TaskTracker yaklaşımı [38].

Hadoop'un ilk versiyonlarında yer alan, Şekil 2.9’da gösterim üzere görev alan JobTracker ve TaskTracker yaklaşımları aşağıda açıklanan artık üç bileşen ile ikame edilmiştir:

1. ResourceManager: Koşan uygulamalar için kümede mevcut kaynakları ayıran bir zamanlayıcı.
2. NodeManager: Kümedeki her düğüm üzerinde çalışır ve ResourceManager yönlendirmesiyle görev alır. Her tek bir düğüm üzerindeki mevcut kaynakları yönetiminden sorumludur.
3. ApplicationMaster: Hadoop altyapısına özel bir kütüphane olup, bir ApplicationMaster Yarn işleminin işleyişine özel görevi çalıştırmakla görevlidir. Resource Manager kaynakların durumu için müzakere ederken ve aynı anda konteynerleri yürütmek ve izlemek için NodeManager ile birlikte çalışmaktan sorumludur. Yarn mimarisi ile Resource Manager, Node Manager, Application Master Şekil 2.10'da görüldüğü halde efendi ve köle (master, slave) olarak iş yükünü almışlardır.



Şekil 2.10: Yarn mimarisi Resource Manager, Node Manager, Application Master [38].

Yarn mimarisi ile Resource Manager, Node Manager, Application Master Şekil 2.10 da görüldüğü halde efendi ve köle (master, slave) olarak iş yükünü almışlardır.

#### 2.4.4 Hadoop MapReduce

Google, LinkedIn, Yahoo, Twitter, Amazon, Microsoft, eBay, Facebook, IBM benzeri birçok lider firma ve üniversitelerde büyük boyuttaki verileri incelemek ve

analiz etmek amacıyla Hortonworks, Cloudera, MapR gibi firmalarda eğitim ve danışmanlık maksatlı ürünlerinin merkezinde Hadoop kullanmaktadır [39].

Çalışmamızda açık kaynak kodlu olması, gelişime açık olması, bir çok işletim sistemi ve donanım üzerinde sorunsuz çalışması nedeniyle MapReduce işlemleri için Hadoop ürün ailesi kullanılmaktadır.



## ÜÇÜNCÜ BÖLÜM

### TEK KART BİLGİSAYAR İLE BULUT OLUŞTURMA

Güncel sistemlerle, tek kart bilgisayarlı sistemlerin donanım öğeleri olarak birebir kıyaslanması mimari farklılıklar nedeniyle uygun görülmemektedir. Ancak ARM işlemci ile yaygın olarak kullanılan Intel Xeon işlemcilerin başarımları, güç tüketimi, ısı karakteristikleri, maliyet karşılaştırmaları yapılması daha anlamlı bir kıyaslama olacaktır. Bu konuda gerek akademik, gerekse ticari birimlerin yapmış olduğu birçok ve çeşitli karşılaştırma bulunmaktadır[40]. Tablo 3.1’de görülen karşılaştırmalar incelendiğinde, ARM işlemcilerin hem işlem / güç ölçütünde hem de işlem sığası / maliyet ölçütünde Xeon işlemcileri 15 kat geride bıraktığı gözlenmektedir [41].

**Tablo 3.1:** İşlemci ve yonga üzeri sistem karşılaştırması üstte tabloda yer almaktadır. 15 kat fark görülmektedir [42].

	Calxeda Energy Core ECX-1000	Intel Xeon E3-1240
Çekirdek Frekansı	1, 1 GHz 3, 3 GHz	
Toplam İstek	1.000.000	1.000.000
Saniye başına İstek	5500	6950
Gecikme (Ort.)	9 ms.	7 ms.
Güç (Ort.)	5, 26 W	102 W
Başarımlar/W üstünlüğü	15X	

Arm ve Intel örnek işlemci kıyaslanması Tablo 3.2’de görüldüğü üzere güç tüketiminde bariz farklar vardır. Xilinx Zynq-7000 serisi Yonga üzeri Sistem gibi kartlar ile x86-64 işlemci mimarisine sahip sistemler karşılaştırıldığında da benzer sonuçlar elde edilmektedir.

**Tablo 3.2:** ARM ve Intel örnek işlemci kıyaslaması.

	ARM Cortex A9 (40nm)	Intel I7 960 (45nm)
Çekirdek Frekansı	2 GHz	3.2 GHz
Çekirdek Sayısı	4	4 Çekirdek (8 HT Thread)
L1 önbellek	32 KB	32 KB
L2 önbellek	-	1 MB + 8 MB L3
Besleme Voltajı	1, 05 V	0, 8-1, 375 V
Güç tüketimi	1, 9 W	130 W

Her ne kadar güncel benzeri sistemlerin başarımlarını testleri ve karşılaştırmaları, yeterli olmasa da, tek kart kümelenmesinin enerji başarımlarını göstermede güzel bir örnek teşkil etmektedir [43]. Accenture şirketinin büyük veri sistemleri karşılaştırma referansı olarak kullandığı yapılandırma Tablo 3.3'te yer almaktadır [44].

**Tablo 3.3:** Referans sistem.

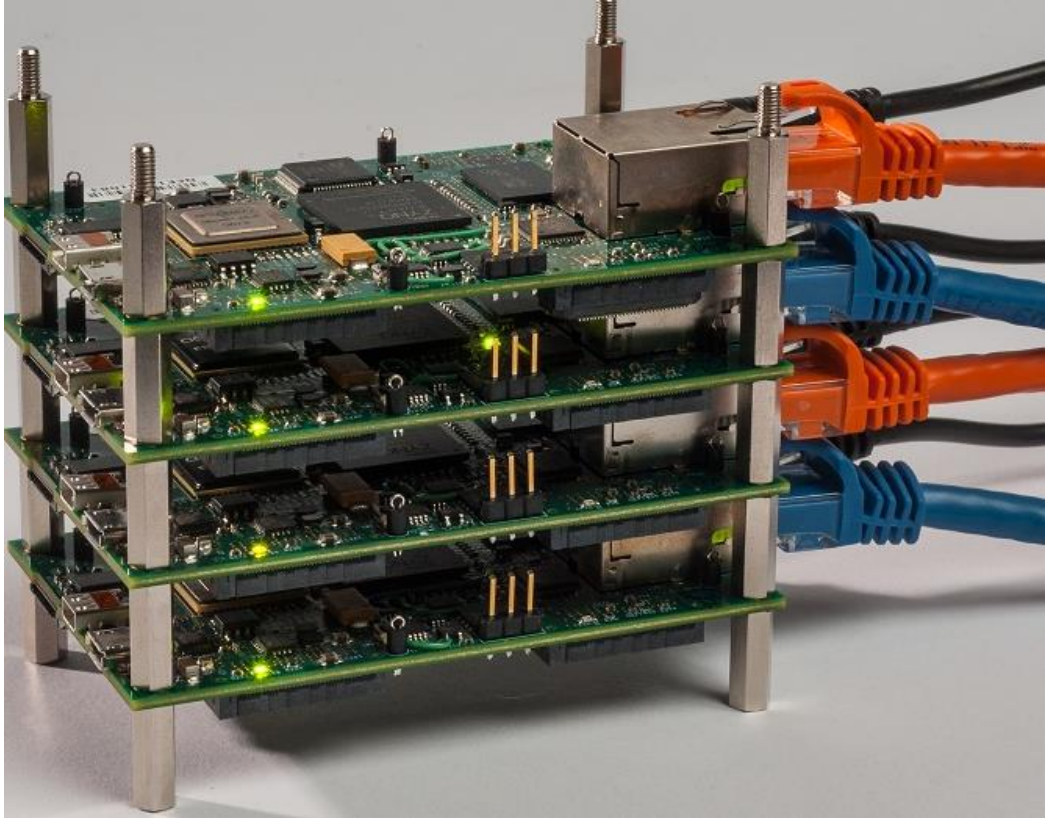
Çalışan/Veri Düğümü Özeti	
Model	Dell R415
İşlemci Tipi	Opteron 4180
Çekirdek Sayısı	12
Çekirdek Hızı (GHz)	2, 6
Bellek (GB)	24
Bellek veri yolu hızı	1333 MHz
Disk sayısı	4
Her diskin sığası (TB)	2
Toplam Sığa (TB)	8
Toplam Küme	
İstemci Düğümleri	1
Birincil Düğümler	3
Çalışan/Veri Düğümleri	22
Çekirdek Sayısı	264
Bellek (GB)	528
Ham TB	176
Kullanılabilir TB 148, 7 (3 kopya ile birlikte) Müsait HDFS TB	49, 6

Verilen 24-düğüm 50 TB sığalı Apache Hadoop kümesi yapılandırmasının, enerji, bakım, soğutma gibi kalemlere sahip toplam sahip olma maliyeti aylık 2914,

58\$ olarak verilmiştir [45]. Bu hesaplamalar sunucuların çalışma süresince harcadıkları enerji yanında soğutma maliyetlerini de içermektedir. Çalışma süresi, ortam sıcaklığı, konumun iklim özellikleri, ülkelere göre değişkenlik gösteren enerji fiyatları, bölgelere bağlı tarife ve vergiler toplam sahip olma maliyetine etki etmektedir.

### 3.1 Tek Kart Bilgisayarları Kümeleme

Gerçekleştirilen çalışmada düğüm donanımları için piyasada halihazırda bulunan Zynq-7020 temelli yonga üzeri sistemler içeren Parallella kullanılmıştır. Dört adet Parallella ile yapılan küme Şekil 3.1’de görülebilmektedir.



Şekil 3.1: Dört adet Parallella oluşturulan küme görülmektedir [46].

Şekilde görüldüğü üzere tek kart bilgisayar donanımlarının kümelenmesi için bir çok hazır birleştirme malzemesi mevcut olduğu gibi, çalışma durumlarını gösteren ışık dizimleri dış kısımda yer almaktadır. Giriş çıkış birimleri ise birbiriyle fiziki olarak temas etmeyecek mesafe de tasarlanmıştır. Tüm bu fiziki özellikleri yanı

sıra hızlı veri yolu ile birbirlerine bağlanabilirlikleri kümeleme için uygunluk oluşturmaktadır.

Düğümler arası ve dış ortamla olan iletişimin, bellek erişiminin özelleştirilmiş donanım üzerinde gerçekleştirilmesi, örüntü eşleme işlemleri özelleştirilmiş giriş/çıkış ara birimlerinin varlığı, ilgili işlevlerin işletim sistemi düzeyinde gerçekleştirilmesini gerektirmektedir. Bu nedenle ARM mimarisine göre derlenmiş ve belirtilen özellikleri haiz bir çekirdeğe sahip, Unix tabanlı bir işletim sistemi kullanılmalıdır. Uygulama için yaygın kullanımı, paket yönetim sisteminin uygunluğu, halihazırda var olan sistemlere uyarlama kolaylığı gibi kıstaslar göz önünde bulundurularak Debian türevi bir işletim sistemi tercih edilmiştir. Seçimi yaparken kullanılacak bu öncül ölçütlerle birlikte, seçilen çekirdeğin ARM mimarisindeki başarımı, üretilen donanıma ve işletim sisteminin bu donanım üzerindeki küme davranışına yatkınlığı gibi ölçütler de değerlendirilmiştir. Standart dağıtımların barındırdığı birçok paket ve çekirdek modülü, ürünün potansiyel kullanım amacı ve kapsamı göz önünde bulundurularak kullanılması belirlenen dağıtımdan çıkarılmıştır. Bu sayede geliştirilen işletim sisteminin boyutunun düşürülmesi, çalışacağı her bir düğümün bellek ve işlemci kullanımı, yönetilebilirliği gibi ölçütlerde başarımının artırılması hedeflenmektedir.

RapidIO, Aurora ve Interlaken yüksek hızlı, paket anahtarlama (packet-switching) bağlantı protokoller olup bu protokollerin Xilinx ve Altera gibi üreticilerin yonga üzeri sistem ürünleri üzerinde 6, 25 Gbps hızdan başlayarak 150 Gbps gibi hızlara erişen çeşitli güncel uygulamaları mevcuttur [47]. Fiziksel katmanda LAN ve WAN bağlantıları için yaygın olarak kullanılan Ethernet ve Fiber optik sistemlerin ortalama hızları göz önünde bulundurularak yalnız bu veri dahi dikkate alındığında, tek bir düğümün bile sahip olacağı işlem sığasının önemi anlaşılmaktadır [48].



Şekil 3.2: Xilinx Zynq-7000 ve adapteva yongası board üzerinde görülmektedir.

Zynq ve Epiphany işlemcileri Şekil 3.2’de yan yana görülebilmektedir. Performans bir map-reduce işi çalıştırıldığında giren-çıkan veri miktarının azaltılması ve kullanılan işlemci miktarının düşürülmesidir. Bu mimarının performansını görebilmek için hız testi yaparken olağan Hadoop dağıtımı kullanılmıştır. Özellikle giriş/çıkış performansı ve işlemci zaman kullanımına önem verilmiştir. Oluşturulan küme bir adet yönetim maksatlı bilgisayar ve dört adet Parallella içermektedir (7000, RAM 1GB) içermektedir. Zynq özellikleri Tablo 3.4’te yer almaktadır.

Tablo 3.4: Zynq 7000 özellikleri.

İşlemci Çekirdek	Çift ARM A9
L1	32 KB komut, 32 KB her işlemci için veri
L2	512 KB
Çip üstü Bellek	256 KB
Mantık Hücre	85K
Blok RAM	560 KB
Sayısal sinyal işleme parçalama	220

Kullanılan bilgisayar özellikleri şu şekildedir: x64, Intel Core i7-4700MQ, İşlemci 2.40GHz × 8, RAM 32 GB.

Tüm nodelar için Apache Hadoop 2.6 kurulumu yapılmıştır. NameNode ve JobTracker PC üzerinde çalışmaktadır, Parallella boardlar DataNode ve Tasktracker olarak ayarlanmıştır. Depolama alanı olarak micro-sd card kullanılmaktadır.

Sürücülerinin IO performansına olumsuz etkisi düşünülerek USB Flash bellek kullanılmamıştır. Boardlar üzerinde ve Pc üzerinde sıra dışı bir tuning ve ayarlama yapılmamıştır.

### **3.2 İşlem Adımları**

Parallella tek kart bilgisayar üzerinde linaro işletim sistemi olarak kullanılmıştır. [www.releases.linaro.org/14.06/ubuntu/trusty-images/developer](http://www.releases.linaro.org/14.06/ubuntu/trusty-images/developer) adresinden indirilen linaro-trusty-developer-20140623-671.tar.gz sd kartlara boot edilebilir halde yazılarak kurulum yapılmıştır.Linaro Kurulumu Ek-A içinde yer almaktadır.

Her bir Parallella için iki ve ya üç adet LXC kurulumu başarılı olmuştur. Bellek yetersizliği nedeniyle ikiden daha fazlasını kurmak bulut için performansı düşürmektedir. LXC Kurulumu Ek-B içinde detaylı olarak yer almaktadır.

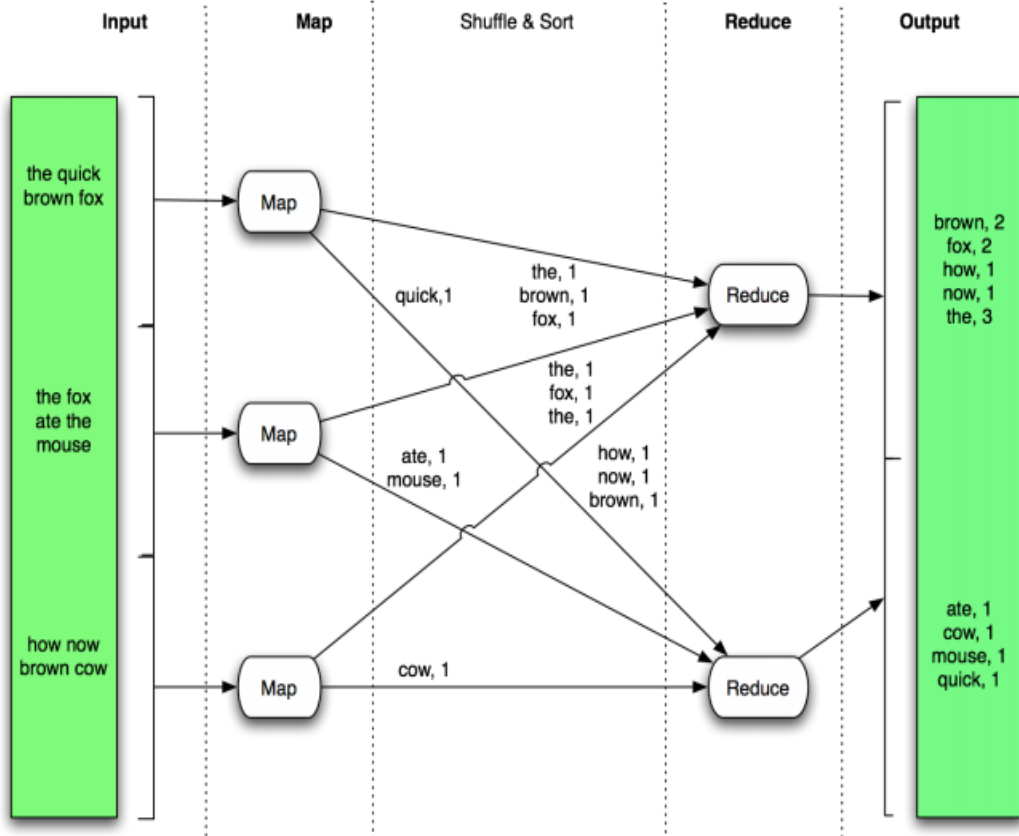
Parallella Tek kart bilgisayar üzerinde Hadoop kurulumu deb paket ve shell script ile yapılmıştır. Hadoop kurulumu ve yapılandırılması Ek-C içinde yer almaktadır.

#### **3.2.1 Bulut Üzerinde Çalışma**

Bulut üzerinde işletim sistemleri, sanallaştırma ve Hadoop kurulumu ile çeşitli deneme programı çalıştırılmıştır. Kullanılanlar WordCount, WordMean, DFSIO, TeraGen, TeraSort, TeraValidate ve pi sayısı üretimidir.

##### **3.2.1.1 WordCount**

Giriş dosyası olan metin içinde geçen her bir tekil kelimeleri ve sayısını MapReduce üzerinden hesaplamaktadır. Şekil 3.3 ile WordCount işleminin veri akışı görülebilmektedir.



Şekil 3.3: MapReduce ile WordCount işleminin veri akışı.

Girişte alınan dosya içinde çeşitli kelimeler yer almaktadır. Map aşamasında tüm bu kelimeler eşlenmektedir. Ardından her biri karıştırılıp sıralanmaktadır. Bu sırada İndirgeme işlemi olarak Reduce yapılmakta ve her bir kelime için kaç tane olduğu sayılmaktadır. Çıktı olarak ise her bir kelime ve kaç tane olduğu verilmektedir. Tüm bunlar eş zamanlı olarak farklı düğümlere dağıtılarak paralel işlem yeteneği ile hızlı bir şekilde başarılmaktadır.

Komut:

```
hduser@hadoop3:~$ yarn jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar wordcount /Turan /out_sh
```

### 3.2.1.2 WordMean

Giriş dosyası olan metin içinde geçen her kelimelerin uzunluklarının ortalamasını MapReduce üzerinden hesaplamaktadır. Her bir işlem eş zamanlı olarak farklı düğümlere dağıtılarak sonuçları toplanılarak hızlı şekilde başarılmaktadır.

Komut:

```
hduser@hadoop3:~$ yarn jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar WordMean /Turan /out_mean
```

### **3.2.1.3 TeraGen**

MapReduce için veri oluşturur. TeraGen rastgele veriler oluşturmaktadır. Bu görevleri istenilen satır sayısına istenilen sayıda böler ve her map rastgele satır atar [49-50].

Komut:

```
hduser@hadoop3:~$ yarn jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar teragen 100 /out_tera
```

### **3.2.1.4 TeraSort**

TeraSort işleminin amacı kümenin Bellek/İşlemci gücünü test etmektir. Olası en kısa sürede sıralaması ve mevcut küme kaynaklarına bağlı olarak değişmesi beklenmelidir. Giriş verisini alarak MapReduce yöntemi ile parçalayıp sıralayıp tamamını sıralı halde çıkarı verir [49-50].

Komut:

```
hduser@hadoop3:~$ yarn jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar terasort /out_tera /out_tera_sort
```

### **3.2.1.5 TeraValidate**

TeraValidate giriş dosyası olarak TeraSort çıktısını alıp sıralamanın doğru yapıldığının sağlamasını yapmaktadır.

Komut:

```
hduser@hadoop3:~$ yarn jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar teravalidate /out_tera_sort /teravalidate_output
```

### **3.2.1.6 TestDFSIO -read**

HDFS için okuma yazma testi amacıyla kullanılan test program parçasıdır. Hadoop dosya sistemi yük testi ve ağdaki gecikmeleri tespit etmede kullanılır.



İşletim sistemi ve kümeleme bilgisayarlarının giriş çıkış yeteneklerinin bir bakışta görülmesini sağlar. Giriş çıkış performans bilgisini her map için bir dosya oluşturarak değişik boyut ve sayıdaki dosyalarla test edilmesi şeklinde kullanılmaktadır.

Komut:

```
hduser@hadoop3:~$ yarn jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.6.0.jar TestDFSIO -read -nrFiles 10 -fileSize 1
```

### 3.2.1.7 TestDFSIO -write

Komut:

```
hduser@hadoop3:~$ yarn jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-jobclient-2.6.0.jar TestDFSIO -write -nrFiles 10 -fileSize 1
```

### 3.2.1.8 $\pi$ (pi) sayısı hesaplama

Matematik sabiti  $\pi$  sayısının ( $\pi=3.1415926535..$ ) değerini Monte-Carlo yöntemi (Arndt ve Haenel, 2006:36-43) kullanarak MapReduce üzerinden bulmaya çalışmaktadır. Rastgele sayıların üretilmesi prensibine dayanan bu yöntem ile işlemci ve bellek işlemlerinin yoğun kullanımı sağlanılmaktadır. IO gecikmelerinin olmadığı sadece hesaplamaların performansı için kullanılmaktadır. Monte-Carlo metodu ile pi sayısı hesaplama diğer hesaplamalara göre çok daha yavaş ve kesin olmayan sonuçlar üretmektedir [51-52].

Komut:

```
hduser@hadoop3:~$ yarn jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar pi 3 3
```

Tüm bu test kodları çalıştırıldığında çıktıları loglanmıştır. Tablo 3.5’de çeşitli hesaplama senaryoları için farklı sanal makine yapılandırılmalı bulut üzerine çalışma sonuçları listelenmiştir.

Pi sayısı hesaplama için farklı sanal makine yapılandırılmalı bulut üzerine çalışma sonuçları Tablo 3.6’da gösterilmiştir. Bu oluşturulan programlar ile tek kart bilgisayarların bellek yoğun, işlemci yoğun, okuma yazma işlemleri yoğun vb. değişik senaryolarda bulut davranışını görmemizi sağlamıştır.

### 3.3 Çalışma Çıktıları

**Tablo 3.5:** Çeşitli hesaplama senaryoları için farklı sanal makine yapılandırılmalı bulut üzerine çalışma sonuçları.

yarn Mapreduce opsiyonları	2 SBC	3 SBC	4 SBC	4 SBC	4 SBC	4 SBC	Bilgisayar	
	2 node	3 node	4 node	6 node	7 node	8 node	2node Ref	
WordCount /input.txt /output_count	8470	8420	8880	8510	9640	8750	2920	CPU time spent (ms)
WordMean /input.txt /output_count	6990	7030	6590	7430	8030	8060	1860	
TeraGen 100	4820	4680	5760	5000	5730	5530	730	
TeraSort	8310	8230	7530	8930	8880	7530	1100	
TeraValidate	5560	5410	5080	5190	6010	5200	850	
TestDFSIO -read -nrFiles 10 -fileSize 1 tümü	-	-	-	-	-	189.287	112.863	
TestDFSIO -read -nrFiles 10 -fileSize 1	-	-	-	-	-	35390	6290	
TestDFSIO -write -nrFiles 10 -fileSize 1 tümü	-	-	-	-	-	283.731	111.845	
TestDFSIO -write -nrFiles 10 -fileSize 1	-	-	-	-	-	45140	8820	

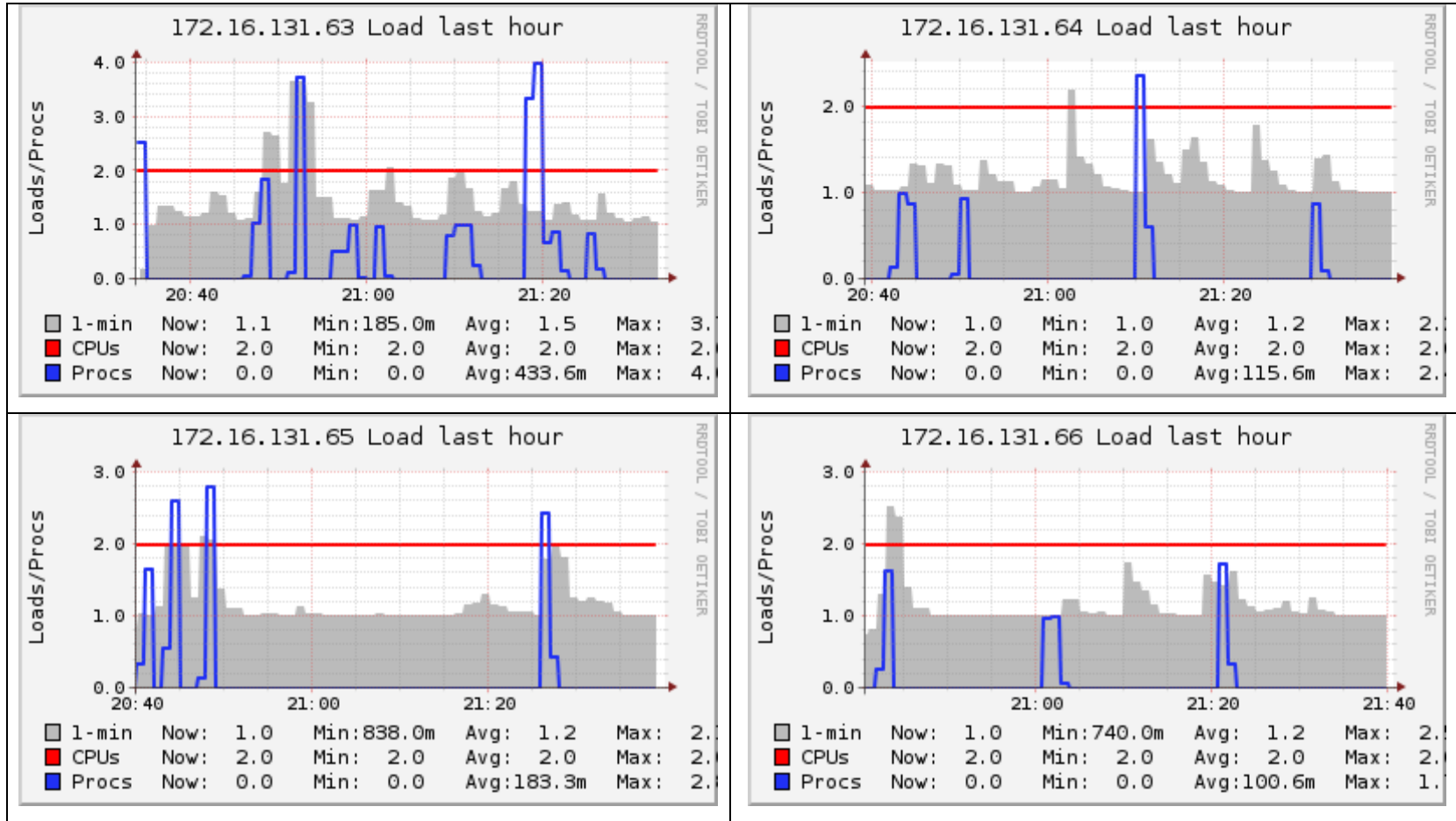
**Tablo 3.6:** 1 Pi Sayısı hesaplama için farklı sanal makine yapılandırılmalı bulut üzerine çalışma sonuçları.

yarn MapReduce	2 SBC	3 SBC	4 SBC	4 SBC	4 SBC	4 SBC	Bilgisayar	CPU time spent (ms)	Hesaplanan Pi Sayısı Değeri
	2 node	3 node	4 node	6 node	7 node	8 node	2node Ref		
pi 3 3	11200	11180	9950	11260	12230	10590	2000		Pi 3.55555555555555555556
pi 3 3 overall	133445	115314	144511	19064	181193	250857	109939		
pi 3 100	11630	12120	11520	12750	12370	11230	1660		Pi 3.16
pi 3 1000	11350	11150	12720	10770	12250	10720	1540		Pi 3.14133333333333333333
pi 3 1000 tümü	108416	87495	162471	169589	122493	106556	106369		
pi 3 10000	-	-	-	11670	10980	10940	3330		Pi 3.14146666666666666667
pi 3 100000	-	-	-	-	12320	12740	3350		Pi 3.14168
pi 4 1000	14270	14750	13580	-	14140	14650	1940		Pi 3.14
pi 4 10000	-	-	12460	-	15310	15930	4280		Pi 3.1414
pi 4 100000	-	-	-	-	15340	14130	4180		Pi 3.14161
pi 5 1000	-	-	-	-	17380	17190	4920		Pi 3.1416
pi 5 10000	-	-	-	-	17760	17840	4450		Pi 3.14248
pi 5 100000	-	-	-	-	19010	19800	5120		Pi 3.141592
pi 6 100	-	-	-	-	başarısız	22310 bellek yetersiz hatası	5590		Pi 3.16666666666666666667
pi 16 100000	-	-	-	-	-	bellek yetersiz	14100		Pi 3.141575

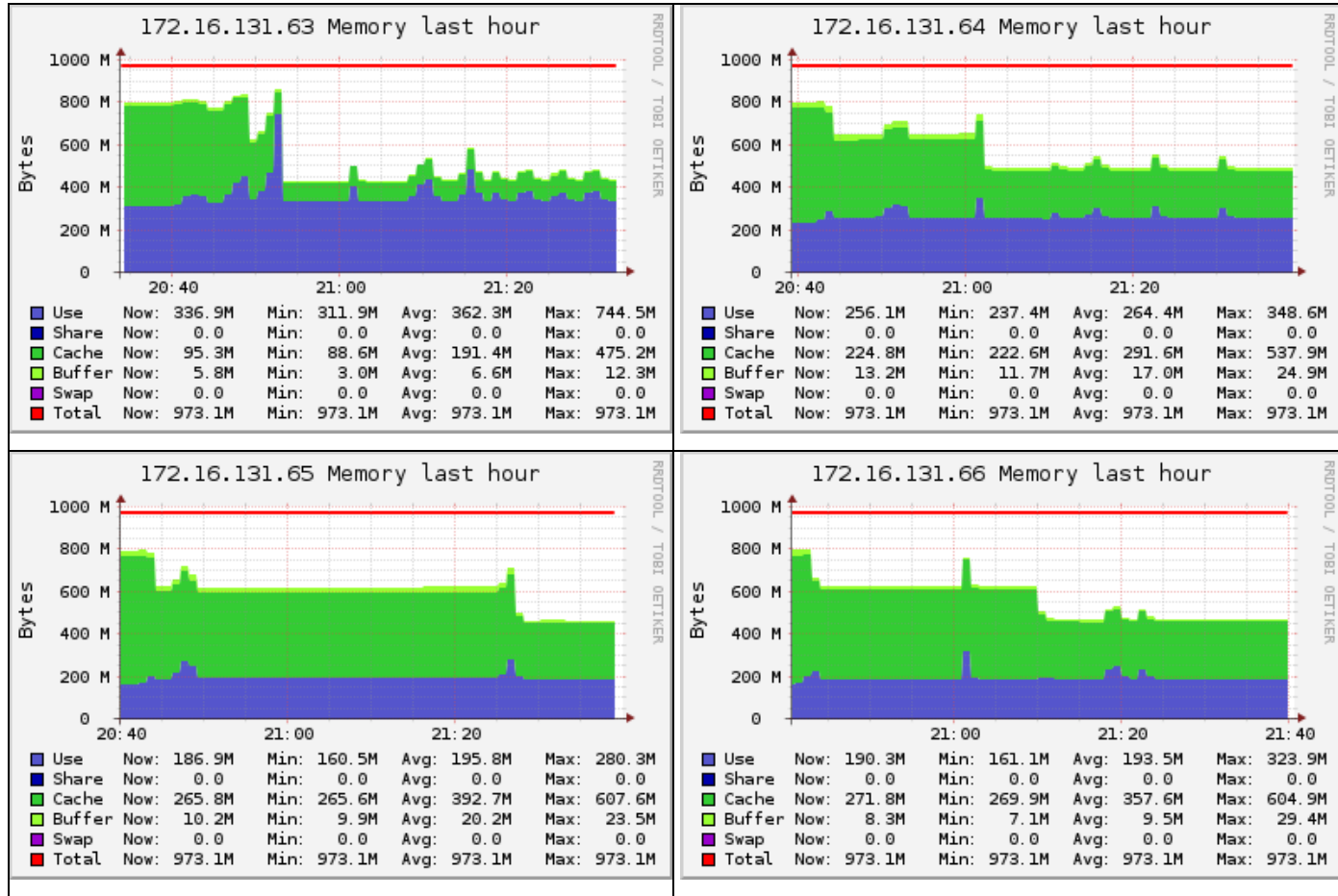
**Tablo 3.7:** DataNode bilgileri.

Düğüm	Durum	Toplam	Kullanılan	DFS dışı Used	Kalan	Blok	Hata
Hadoop9 (10.0.33.108:50010)	In Service	14.29 GB	289.26 KB	3.73 GB	10.56 GB	2	0
Hadoop8 (10.0.33.107:50010)	In Service	14.29 GB	289.51 KB	3.72 GB	10.56 GB	4	0
Hadoop7 (10.0.33.106:50010)	In Service	13.9 GB	289.26 KB	3.7 GB	10.2 GB	2	0
Hadoop6 (10.0.33.105:50010)	In Service	13.9 GB	289.26 KB	3.7 GB	10.2 GB	2	0
Hadoop5 (10.0.33.104:50010)	In Service	13.9 GB	289.26 KB	3.7 GB	10.2 GB	2	0
Hadoop3 (10.0.33.102:50010)	In Service	14.29 GB	375.95 KB	3.73 GB	10.56 GB	7	0
Hadoop4 (10.0.33.103:50010)	In Service	13.9 GB	375.62 KB	3.7 GB	10.2 GB	4	0
Hadoop2 (10.0.33.101:50010)	In Service	14.29 GB	289.34 KB	3.72 GB	10.56 GB	3	0

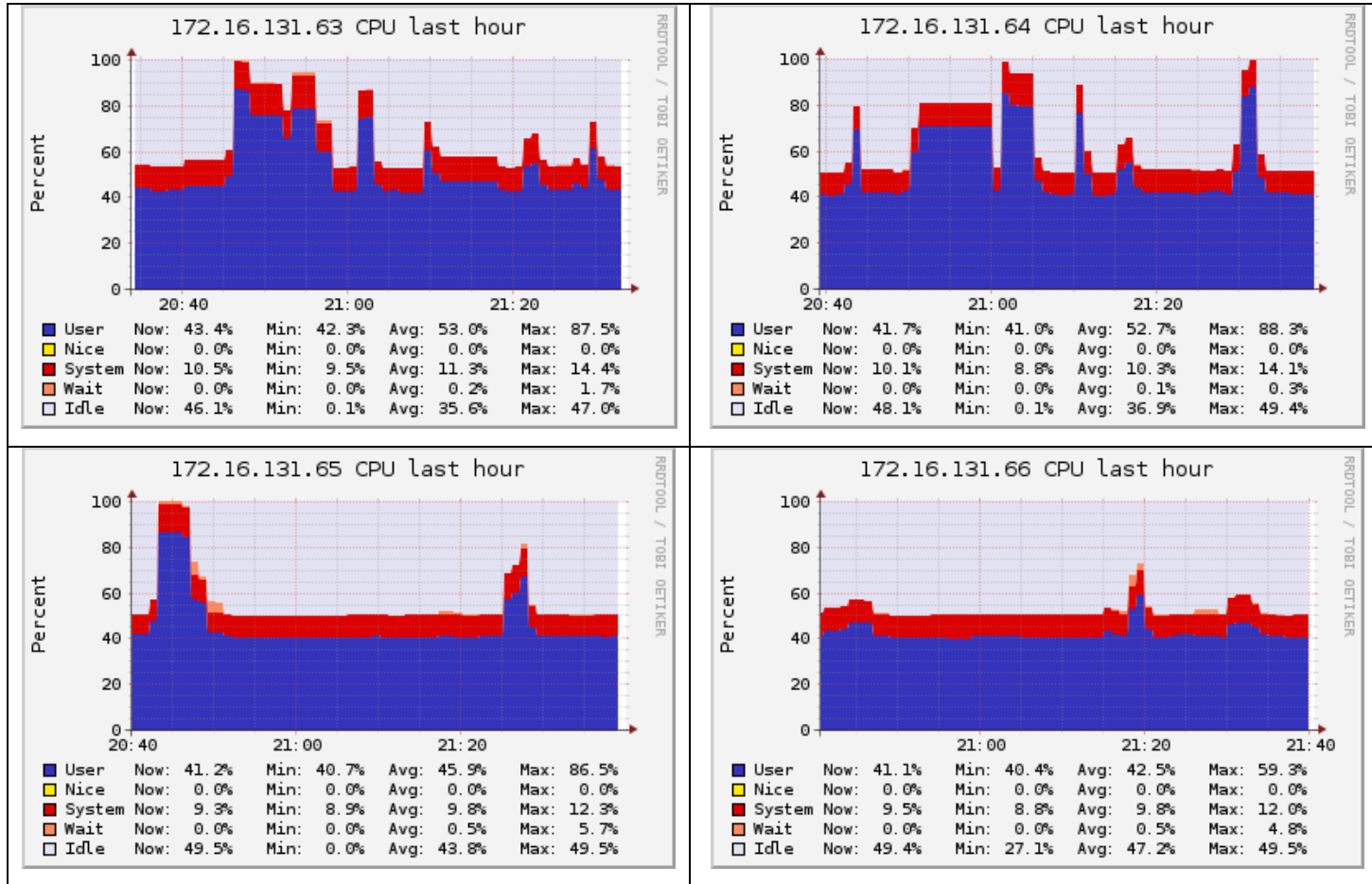
**Tablo 3.8:** 4 adet tek kart bilgisayar üzerinde 4 LXC çalışırken her biri üzerindeki yük dağılımı.



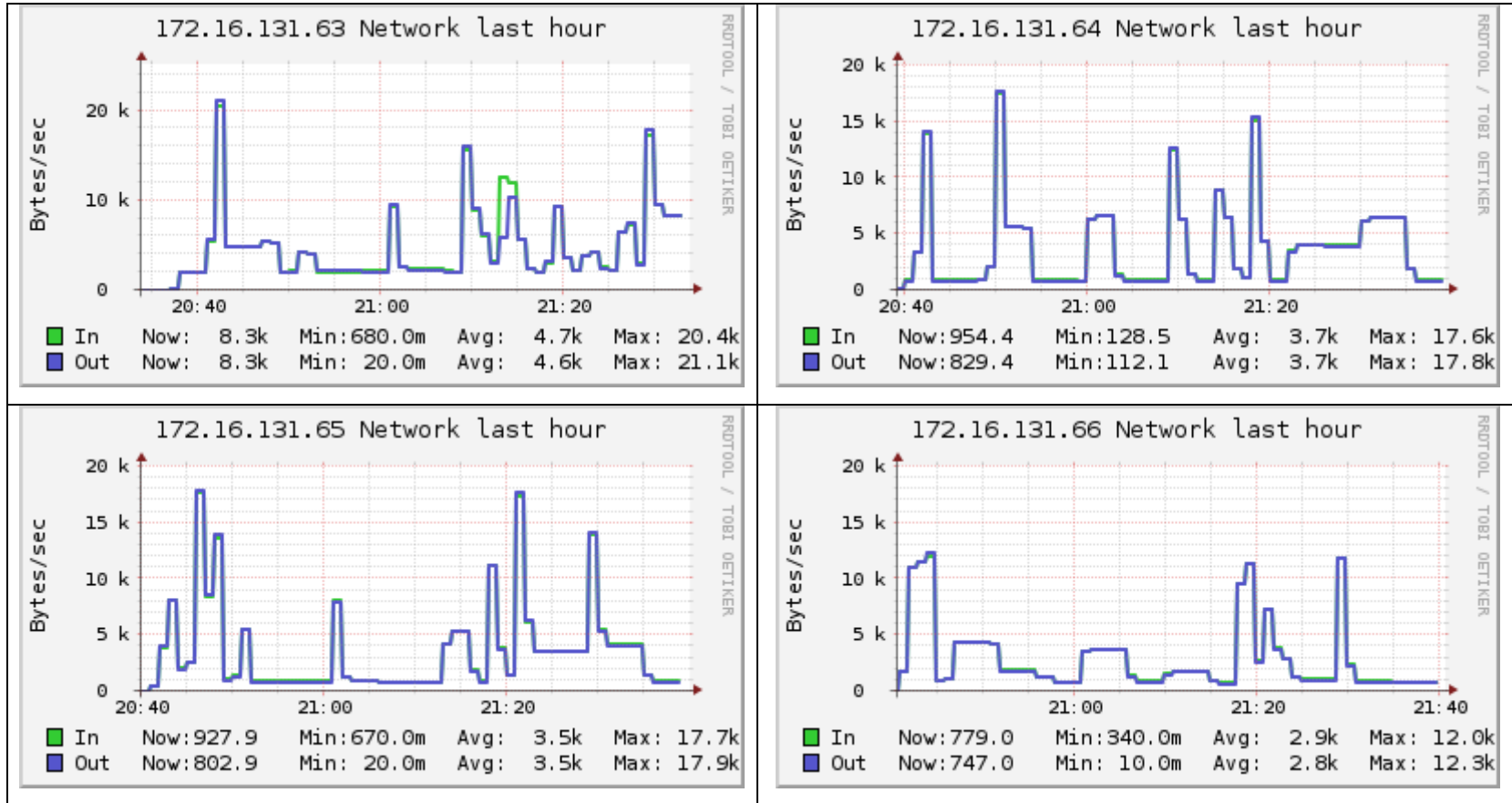
**Tablo 3.9:** 4 adet tek kart bilgisayar üzerinde 4 LXC çalışırken her biri üzerindeki bellek durumu.



**Tablo 3.10:** 4 adet tek kart bilgisayar üzerinde 4 LXC çalışırken her biri üzerindeki işlemci dağılımı.

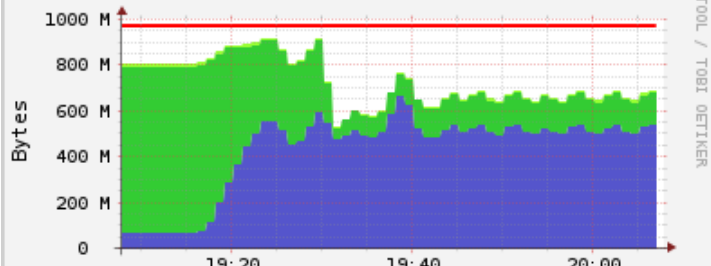
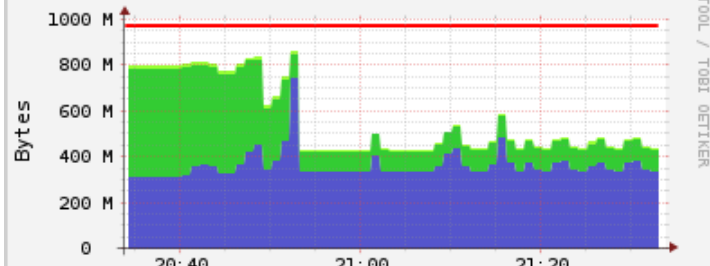
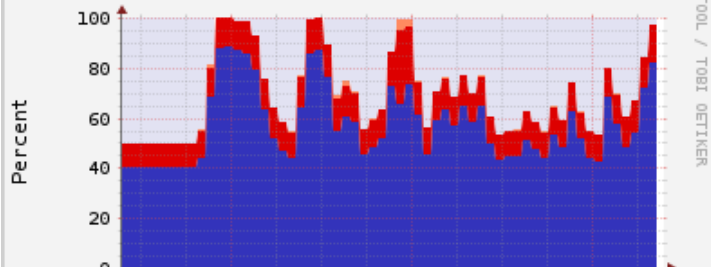
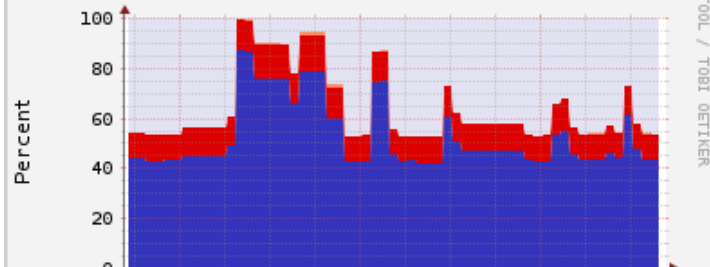


**Tablo 3.11:** 4 adet tek kart bilgisayar üzerinde 4 LXC çalışırken her biri üzerindeki ağ durumu.





**Tablo 3.12:** Tek kart bilgisayar üzerinde 2 LXC ve 1 LXC çalışırken bellek işlemci durumları.

Her Parallella üzerinde 2 LXC	Her Parallella üzerinde 1 LXC	Sonuç																																																												
<p>172.16.131.63 Memory last hour</p>  <table border="1" data-bbox="315 630 1023 758"> <tr> <td>Use</td> <td>Now: 543.5M</td> <td>Min: 68.0M</td> <td>Avg: 428.5M</td> <td>Max: 668.1M</td> </tr> <tr> <td>Share</td> <td>Now: 0.0</td> <td>Min: 0.0</td> <td>Avg: 0.0</td> <td>Max: 0.0</td> </tr> <tr> <td>Cache</td> <td>Now: 139.8M</td> <td>Min: 53.3M</td> <td>Avg: 292.7M</td> <td>Max: 726.9M</td> </tr> <tr> <td>Buffer</td> <td>Now: 6.4M</td> <td>Min: 1.6M</td> <td>Avg: 6.4M</td> <td>Max: 11.1M</td> </tr> <tr> <td>Swap</td> <td>Now: 0.0</td> <td>Min: 0.0</td> <td>Avg: 0.0</td> <td>Max: 0.0</td> </tr> <tr> <td>Total</td> <td>Now: 973.1M</td> <td>Min: 973.1M</td> <td>Avg: 973.1M</td> <td>Max: 973.1M</td> </tr> </table>	Use	Now: 543.5M	Min: 68.0M	Avg: 428.5M	Max: 668.1M	Share	Now: 0.0	Min: 0.0	Avg: 0.0	Max: 0.0	Cache	Now: 139.8M	Min: 53.3M	Avg: 292.7M	Max: 726.9M	Buffer	Now: 6.4M	Min: 1.6M	Avg: 6.4M	Max: 11.1M	Swap	Now: 0.0	Min: 0.0	Avg: 0.0	Max: 0.0	Total	Now: 973.1M	Min: 973.1M	Avg: 973.1M	Max: 973.1M	<p>172.16.131.63 Memory last hour</p>  <table border="1" data-bbox="1055 630 1762 758"> <tr> <td>Use</td> <td>Now: 336.9M</td> <td>Min: 311.9M</td> <td>Avg: 362.3M</td> <td>Max: 744.5M</td> </tr> <tr> <td>Share</td> <td>Now: 0.0</td> <td>Min: 0.0</td> <td>Avg: 0.0</td> <td>Max: 0.0</td> </tr> <tr> <td>Cache</td> <td>Now: 95.3M</td> <td>Min: 88.6M</td> <td>Avg: 191.4M</td> <td>Max: 475.2M</td> </tr> <tr> <td>Buffer</td> <td>Now: 5.8M</td> <td>Min: 3.0M</td> <td>Avg: 6.6M</td> <td>Max: 12.3M</td> </tr> <tr> <td>Swap</td> <td>Now: 0.0</td> <td>Min: 0.0</td> <td>Avg: 0.0</td> <td>Max: 0.0</td> </tr> <tr> <td>Total</td> <td>Now: 973.1M</td> <td>Min: 973.1M</td> <td>Avg: 973.1M</td> <td>Max: 973.1M</td> </tr> </table>	Use	Now: 336.9M	Min: 311.9M	Avg: 362.3M	Max: 744.5M	Share	Now: 0.0	Min: 0.0	Avg: 0.0	Max: 0.0	Cache	Now: 95.3M	Min: 88.6M	Avg: 191.4M	Max: 475.2M	Buffer	Now: 5.8M	Min: 3.0M	Avg: 6.6M	Max: 12.3M	Swap	Now: 0.0	Min: 0.0	Avg: 0.0	Max: 0.0	Total	Now: 973.1M	Min: 973.1M	Avg: 973.1M	Max: 973.1M	<p>1LXC kurulu olan yapılandırmada bellek bariz olarak daha az kullanıldığı gözlemlenilmektedir.</p>
Use	Now: 543.5M	Min: 68.0M	Avg: 428.5M	Max: 668.1M																																																										
Share	Now: 0.0	Min: 0.0	Avg: 0.0	Max: 0.0																																																										
Cache	Now: 139.8M	Min: 53.3M	Avg: 292.7M	Max: 726.9M																																																										
Buffer	Now: 6.4M	Min: 1.6M	Avg: 6.4M	Max: 11.1M																																																										
Swap	Now: 0.0	Min: 0.0	Avg: 0.0	Max: 0.0																																																										
Total	Now: 973.1M	Min: 973.1M	Avg: 973.1M	Max: 973.1M																																																										
Use	Now: 336.9M	Min: 311.9M	Avg: 362.3M	Max: 744.5M																																																										
Share	Now: 0.0	Min: 0.0	Avg: 0.0	Max: 0.0																																																										
Cache	Now: 95.3M	Min: 88.6M	Avg: 191.4M	Max: 475.2M																																																										
Buffer	Now: 5.8M	Min: 3.0M	Avg: 6.6M	Max: 12.3M																																																										
Swap	Now: 0.0	Min: 0.0	Avg: 0.0	Max: 0.0																																																										
Total	Now: 973.1M	Min: 973.1M	Avg: 973.1M	Max: 973.1M																																																										
<p>172.16.131.63 CPU last hour</p>  <table border="1" data-bbox="315 1098 1023 1225"> <tr> <td>User</td> <td>Now: 82.4%</td> <td>Min: 40.5%</td> <td>Avg: 57.7%</td> <td>Max: 88.7%</td> </tr> <tr> <td>Nice</td> <td>Now: 0.0%</td> <td>Min: 0.0%</td> <td>Avg: 0.0%</td> <td>Max: 0.0%</td> </tr> <tr> <td>System</td> <td>Now: 15.3%</td> <td>Min: 9.0%</td> <td>Avg: 11.9%</td> <td>Max: 29.1%</td> </tr> <tr> <td>Wait</td> <td>Now: 0.0%</td> <td>Min: 0.0%</td> <td>Avg: 0.3%</td> <td>Max: 4.6%</td> </tr> <tr> <td>Idle</td> <td>Now: 2.3%</td> <td>Min: 0.0%</td> <td>Avg: 30.1%</td> <td>Max: 50.0%</td> </tr> </table>	User	Now: 82.4%	Min: 40.5%	Avg: 57.7%	Max: 88.7%	Nice	Now: 0.0%	Min: 0.0%	Avg: 0.0%	Max: 0.0%	System	Now: 15.3%	Min: 9.0%	Avg: 11.9%	Max: 29.1%	Wait	Now: 0.0%	Min: 0.0%	Avg: 0.3%	Max: 4.6%	Idle	Now: 2.3%	Min: 0.0%	Avg: 30.1%	Max: 50.0%	<p>172.16.131.63 CPU last hour</p>  <table border="1" data-bbox="1055 1098 1762 1225"> <tr> <td>User</td> <td>Now: 43.4%</td> <td>Min: 42.3%</td> <td>Avg: 53.0%</td> <td>Max: 87.5%</td> </tr> <tr> <td>Nice</td> <td>Now: 0.0%</td> <td>Min: 0.0%</td> <td>Avg: 0.0%</td> <td>Max: 0.0%</td> </tr> <tr> <td>System</td> <td>Now: 10.5%</td> <td>Min: 9.5%</td> <td>Avg: 11.3%</td> <td>Max: 14.4%</td> </tr> <tr> <td>Wait</td> <td>Now: 0.0%</td> <td>Min: 0.0%</td> <td>Avg: 0.2%</td> <td>Max: 1.7%</td> </tr> <tr> <td>Idle</td> <td>Now: 46.1%</td> <td>Min: 0.1%</td> <td>Avg: 35.6%</td> <td>Max: 47.0%</td> </tr> </table>	User	Now: 43.4%	Min: 42.3%	Avg: 53.0%	Max: 87.5%	Nice	Now: 0.0%	Min: 0.0%	Avg: 0.0%	Max: 0.0%	System	Now: 10.5%	Min: 9.5%	Avg: 11.3%	Max: 14.4%	Wait	Now: 0.0%	Min: 0.0%	Avg: 0.2%	Max: 1.7%	Idle	Now: 46.1%	Min: 0.1%	Avg: 35.6%	Max: 47.0%	<p>2 LXC kurulu olan yapılandırmada işlemci çok daha yoğun ve sıklıkla kullanılmış ve sistem tarafından işlemci kullanımı kırmızı bölgelerde daha tavan yaptığı görülmektedir.</p>										
User	Now: 82.4%	Min: 40.5%	Avg: 57.7%	Max: 88.7%																																																										
Nice	Now: 0.0%	Min: 0.0%	Avg: 0.0%	Max: 0.0%																																																										
System	Now: 15.3%	Min: 9.0%	Avg: 11.9%	Max: 29.1%																																																										
Wait	Now: 0.0%	Min: 0.0%	Avg: 0.3%	Max: 4.6%																																																										
Idle	Now: 2.3%	Min: 0.0%	Avg: 30.1%	Max: 50.0%																																																										
User	Now: 43.4%	Min: 42.3%	Avg: 53.0%	Max: 87.5%																																																										
Nice	Now: 0.0%	Min: 0.0%	Avg: 0.0%	Max: 0.0%																																																										
System	Now: 10.5%	Min: 9.5%	Avg: 11.3%	Max: 14.4%																																																										
Wait	Now: 0.0%	Min: 0.0%	Avg: 0.2%	Max: 1.7%																																																										
Idle	Now: 46.1%	Min: 0.1%	Avg: 35.6%	Max: 47.0%																																																										

**Tablo 3.13:** Tek kart bilgisayar üzerinde 2 LXC ve 1 LXC çalışırken bellek işlem durumları.

4SBC + Her biri üzerinde 2 LXC	4SBC + Her biri üzerinde 1 LXC	Sonuç
		1LXC kurulu olan yapılandırmada bellek daha az kullanılmıştır.
		1LXC kurulu olan yapılandırmada işlemci daha az kullanılmıştır.
		2LXC kurulu olan yapılandırmada toplam işlem daha fazla olmuştur kullanılmıştır.
TeraSort 7530ms hızlı hesaplanmıştır	TeraSort 8930ms	Düğüm sayısının artması ile sıralama işlemi daha hızlı yapılmıştır.
pi (3 3) 10590ms hızlı hesaplanmıştır	pi (3 3) 11200ms	Düğüm sayısının artması ile işlemci yoğun pi sayısı işlemi daha hızlı yapılmıştır.
TeraValidate 5200	TeraValidate 5080	Düğüm sayısı az olunca disk i/o hızlı olmuştur

### 3.4 Değerlendirme ve Tartışma

Yapılan uygulamada 2 tekkart bilgisayar, 3 tekkart bilgisayar, 4 tekkart bilgisayar, ardından 10 tekkart bilgisayara kadar kümeler oluşturulmuştur. Birbirlerini ağ bağdaştırıcısı üzerinden görür hale getirilmiştir. Üzerine ubuntu, debian ve linaro işletim sistemleri kurulmuştur. Ubuntu ve Debian'a nazaran Linaro'nun daha az kaynak sarfıyatlı işletim sistemi olduğu gözlemlenmiştir. 2 parallella üzerine 2 LXC, 3 parallella üzerine 3LXC, 4 parallella üzerine 4LXC, 4 parallella üzerine 6LXC, 4 parallella üzerine 7LXC, 4 parallella üzerine 8LXC kurulu olan yapılandırmalar ile kümelerin üzerinde sanallaştırma yapılmıştır. Her durumun yer aldığı

1. Tablo 3.5 incelendiğinde çeşitli hesaplama senaryoları için farklı sanal makine yapılandırılmalı bulut üzerine çalışma sonuçları
2. Tablo 3.6 incelendiğinde Pi sayısı hesaplama için farklı sanal makine yapılandırılmalı bulut üzerine çalışma sonuçları
3. Tablo 3.7 incelendiğinde DataNode bilgileri
4. Tablo 3.8 incelendiğinde 4 adet tek kart bilgisayar üzerinde 4 LXC çalışırken her biri üzerindeki yük dağılımı
5. Tablo 3.9 incelendiğinde 4 adet tek kart bilgisayar üzerinde 4 LXC çalışırken her biri üzerindeki bellek durumu
6. Tablo 3.10 incelendiğinde 4 adet tek kart bilgisayar üzerinde 4 LXC çalışırken her biri üzerindeki işlemci dağılımı
7. Tablo 3.11 incelendiğinde 4 adet tek kart bilgisayar üzerinde 4 LXC çalışırken her biri üzerindeki ağ durumu
8. Tablo 3.12 incelendiğinde tek kart bilgisayar üzerinde 2 LXC ve 1 LXC çalışırken bellek işlemci durumları
9. Tablo 3.13 incelendiğinde tek kart bilgisayar üzerinde 2 LXC ve 1 LXC çalışırken bellek işlem durumları

görülebilmektedir, tüm bunlar incelendiğinde mikroişlemci, bellek, network ve işlem yoğunluğu açısından incelendiğinde tek kart bilgisayarlar ile yapılan bulut işlemci gücü açısından yetersizlik gözlemlenmemiştir. Sistem gözleme araçlarından işlemci meşguliyeti değerlerine bakıldığında %70 üzerine çıktığı gözlemlenmemiştir. İşlemci değerlerinde bekleme, askıda kalma, cevap verememe olmamakta ancak bellek açısından yoğunluk yaşandığı gözlemlenmektedir [53].

Bellek açısından bakıldığında tek kart bilgisayar başına kap temelli sanallaştırma sayısı az olduğunda bellek daha yeterli olabilmektedir. Ancak 2GB olan Ram'in sistem kaynakları için 1 GB'ının sistemin kendi kullanımına adanmış olması aleyhte bir durum olarak görülmektedir. Kümenin toplam belleğinin arttırılması için donanım yani düğüm sayısının arttırılması geçici bir çözüm olabilmektedir.

Küme oluşturulmasında bir tek kart bilgisayar üzerinde bir adet sanallaştırma yapılırken diğerinde birden fazla sanallaştırma yapılır ise ve bu farklı sanallaştırma içerenlerden kümeleme yapılarak melez bir yaklaşım yapıldığında performanstan ziyade bekleminin arttığı tespit edilmiştir. Farklı sayıda sanallaştırma içeren düğümler küme içinde işlemci ve bellek açısından yetersiz kalmaktadır. Bunun nedeni donanımın üzerindeki kaynakları göz önüne alarak yük dağıtımının yapılamamış olmasıdır. Sanal makine başına düşen bellek ve işlemci gücü göz önüne alarak yük dağılımı sağlayan yazılım eklentisine ihtiyaç vardır.

Ağ iletişimi açısından bir sorun görülmemektedir. Hatta bu donanımların ağ donanımları ile akan bilgi işlemede kullanımının yerel disk kullanımına göre daha uygun olduğu görülmektedir. Dosya giriş çıkış işlemleri ve dosya okuma yazma işlemlerinde yoğun bir gecikme olduğu gözlemlenmiştir.

Her bir donanıma bir LXC kurulu olan yapılandırmada işlemci daha az kullanılmıştır. Her bir donanıma iki LXC kurulu olan yapılandırmada toplam işlem daha fazla olmuştur.

Düğüm sayısının artması ile sıralama işlemi daha hızlı yapılmıştır. Düğüm sayısının artması ile işlemci yoğun pi sayısı işlemi daha hızlı yapılmıştır. Düğüm sayısı az olunca disk girdi çıktı işlemleri hızlı olmuştur. Değişik yapılandırmaya sahip düğümler ile düğüm sayısı arttırmak hızlanma yerine gecikme ile sonuçlanmaktadır. Bu yapılarda homojen kümelenme yapılması daha uygun olmaktadır. Amaca göre yüksek erişilebilirlik kümeleri, kullanım şekline göre adanmış kümeler ve üye düzenine göre homojen kümeler için tek kart bilgisayar kullanımı daha uygun olduğu gözlemlenmektedir. Bu ürünlerin en temel özelliği ise yapılan işlem başına harcanan elektrik enerjisinin düşük olmasıdır.

Tek kart bilgisayar kümelendirilmesinin dosya barındırma ve paylaşım için, yazılım test otomasyonu araçları için, dinamik içerikli web siteleri sunuculuğu için ve orta katman sunucuları için kullanılması uygun görülmektedir.

Kap temelli sanallaştırma sunucuları için sistem kaynakları kısıtlı kalmakta beklenen performansı gösterememekte ancak geç dahi olsa sonuç döndürebilmektedir.

Uygulama için kullanılan tek kart bilgisayarlar açık kaynak olduğundan sürekli dünya üzerinde ilgi duyanlar tarafından katkı ile güncellenmektedir. Doğru kullanım ve ayar için kullanılan işletim sistemi, sürücü, programların bir çoğuna hakim olmak ayrı bir uzmanlık bilgisi gerektirmektedir.

ARM aslen mimarinin adı olup bir çok farklı üretici tarafından ARM mimarili işlemci üretilebilmekte ve bunlar arasında da bir performans kalite değişimi oluşmaktadır. Her küme için ürün seçiminde aynı üretici kullanımı en iyi çözüm gözükmektedir. Bu düğümler arası bekleme zamanını azaltmakta ancak bu da üreticiden bağımsızlığın önüne engel getirmektedir.

## DÖRDÜNCÜ BÖLÜM

### SONUÇ VE ÖNERİLER

Kümeleme yapılmaya elverişliliği, işletim sistemi desteği, açık kaynak kod desteği ve düşük enerji tüketimleri ile gittikçe kullanımı yaygınlaşan tek kart bilgisayarların kümeleme, sanallaştırma ve bulut işlemleri için de kullanılabilmesi sonucuna ulaşılmıştır [54-55]. Fiziksel boyutlarının yanında paralel işlem ve paralel hesaplama için fiyat/performans açısından tercih edilebileceği gibi enerji maliyeti açısından da etkin olduğu anlaşılmıştır. Pazarlama açısından mini süper bilgisayar olarak lanse edilen bu tek kart bilgisayarların aslında birkaç yıl öncesine göre bu şekilde adlandırılması makul görülürken çoklu sayıda küme içinde kullanımının daha uygun olacağı eğitim, sağlık, savunma sanayi gibi sinyal işleme, görüntü işleme gibi çoklu işlem yapılmasının daha uygun olduğu alanlarda tercih edilmesinin uygun olduğu tez çalışması çıktılarında gözlemlenmiştir. Merkezi işlem birimi kullanımı yoğun olan analitik işlemlerdeki hız bu uygunluğu ispatlamaktadır.

Büyük veri, bulut bilişim, siber güvenlik, akıllı sistemler gibi alanlarda kullanılan ve bu alanlardaki birçok uygulamanın temelini oluşturan örüntü eşleme ve örüntü temelli veri dönüşümü işlemleri günümüzde yazılımsal olarak gerçekleştirilmektedir.[55] Bilgisayarların bulunduğu basit bir sunucu odasından, veri merkezlerine kadar olan ölçekte soğutma, güç ve bakım gibi birçok ek maliyet ve yatırım gereksinimi ortaya çıkmaktadır. Google gibi büyük veri merkezlerine ihtiyaç duyan şirketlerin bazı veri merkezlerini güç santrallerinin yakınına kurduğu, enerji kıtlığı ve küresel ısınma gibi dünyada her geçen gün önemi artan konular, enerjinin pahalı olduğu ülkeler göz önünde bulundurulduğunda düşük güç tüketimi ve düşük soğutma gereksinimi olan sistemlere olan ihtiyaç ortaya çıkmaktadır. Çalışma sonucunda işlemlerin ARM gibi oldukça düşük enerji gereksinimi ve ısı kaybı bulunan işlemcilerle dayanması, düğüm başına 5-10 watt aralığında bir güç

tüketimini mümkün hale getirmektedir [56-57]. Dolayısıyla tek kart bilgisayarlar, eşdeğer işlem yetisine sahip çağdaşlarından daha az güç gereksinimine ve soğutma giderlerine sahip olmakta; böylelikle eşdeğer bir sistem için hem yatırım tutarlarının hem de toplam sahip olma maliyetlerinin oldukça düşmesi mümkün olmaktadır [57].

Tek kart bilgisayarların düşük güç tüketimi ve düşük soğutma maliyetinin yanında, temel aldığı donanımların diğer avantajlarından dolayı fiziksel ağırlığı ve boyutları, eşdeğer çağdaşlarına göre çok daha düşük olabilmektedir. Dolayısıyla bu kısıtların hayati olduğu alanlarda ve özellikle iş sürekliliği gereksinimlerinde ürünün kullanımı kolaylaşmakta, ürünün varlığı daha önce olanaklı olmayan yeni uygulamaların yapılabilmesini mümkün kılmaktadır [58]. Örneğin uzay, uydu ve havacılık alanlarındaki en büyük sorunlardan biri, gözlem amacıyla toplanan her türlü verinin gerçek zamanlı olarak araç üzerinde işlenmesidir. Güncel sistemler gerek fiziksel ağırlık ve boyut, gerekse güç tüketimi nedeniyle verinin araç üzerinde işlenmesini imkansız kılmaktadır. Bu nedenle yaygın olarak ya toplanan veri yeryüzündeki istasyonlara gönderilmekte ve burada işlenmekte, yahut araç üzerine yerleştirilen görece düşük güç tüketimli, sabit bir görevi olan entegre devreler aracılığıyla kısıtlı işlemler yapılmaktadır. Bu nedenlerle araçlardan elde edilebilecek işlevsel verim ve etkinlik düşmektedir.

Bu tarz tek kart bilgisayarlar hafif ve düşük güç gereksinimine sahip nitelikte olması ve hedeflenen yüksek başarımlı, veri toplama ve denetim amacıyla yerel noktalara kurulumunu kolaylaştırmakta; böylelikle akıllı şehirler, şehir güvenliği, akıllı trafik yönetim sistemleri gibi akıllı sistemlerdeki uygulamaları hızlandırıcı ve maliyetleri düşürücü nitelikte olmaktadır. Ülkemizde bulut bilişim ilerledikçe kamu ve özel sektörde bir çok kullanım alanına sahip olabilecektir.[59]

Apache Hadoop, Apache Cassandra, MongoDB gibi açık kaynaklı Büyük Veri yazılımları, SQLite, MariaDB, MySQL, PostgreSQL vb. veritabanları tek kart bilgisayar mimarisinde çalışacak şekilde yeniden derlenmiş / uyarlanmış ve geçerli şekilde çalıştıkları gözlenmiştir. Owncloud web üzerinde çalışan, açık kaynaklı bir bulut dosya barındırma ve depolama yazılım sistemi olup bir taneden çokluya kadar ölçeklenebilir bir şekilde tek kart bilgisayar kümemizde başarıyla çalışmaktadır.

Yaptığımız uygulamada katı hal sığa kullanımı ile de yapmak istediğimizde sürücü desteği sıkıntısı yaşanmıştır. Bu konuda uyarlanmış sığa sürücülerini ihtiyacımızı giderilememiştir.

Tek kart bilgisayarların bir kısmında alanda programlanabilir kapı dizileri tümleşik olmasına rağmen veri yolu üzerinde yapılması gereken işlemler ile ilgili bir çok geliştirme henüz gerçekleşmemiştir. Gelişmeye açık bir alan olarak ilgi duyanların katkısını beklemektedir. Ancak bunlar yapıldığında bir çok sinyal işleme işlemleri çok hızlandırılmış olacaktır [60].

64 bit tek kart bilgisayarlar henüz yaygınlaşmamıştır, bu da bir çok kısıtı beraberinde getirmektedir. Yapılan çalışmada sistematik paralel programlama çözümü olarak MapReduce için Apache Hadoop yerine Sector/Sphere kullanılabilir [61]. Sanallaştırma için LXC yerine başka çözümler kullanılabilceği gibi baştan da yazılabilir[62]. Konu ile ilgilenen araştırmacılara fpga üzerinde MapReduce işlemlerinin gerçekleştirilmesi ise en önerilen çalışma konusudur. Büyük veri işlemlerinde yüksek performans sağlanması için üzerinde çalışılması en elzemdir. ARM yanında Intel ve AMD firmalarının yeni nesil işlemcilerinde fpga kullanımına yatırım yapması bunu destekleyen en önemli etkenlerdendir.

Tek kart bilgisayar üzeri Hadoop MapReduce işlemlerinin LXC ile kapı tabanlı sanallaştırması tarafımızdan yapılmıştır. Ancak bir çok farklı bilgi alanı uygulamalarını içeren hazır sığa yansılarının oluşturulması ve farklı donanımlarda çapraz kullanım sağlayabilen sığa yansısı üretimi bu konuda çalışanlara önerilen bir alandır.



## KAYNAKÇA

1. Lämmel, R., (2008). Google's MapReduce programming model — Science of Computer Programming
2. NIST Cloud Computing, <http://csrc.nist.gov/groups/SNS/cloud-computing>  
Erişim: 23.01.2015
3. Bader, David; Robert Pennington, *Cluster Computing: Applications*, 1996
4. Zubaroglu Alaettin, Bilgisayar Kümeleri, e-bergi.com *Elektronik Dergi*.  
Erişim: 21.09.2015
5. Cluster Computing, Mississippi State University, <http://hpc.msstate.edu/>  
Erişim: 21.03.2015
6. Yokokawa, (2011). M K Computer, "Low Power Electronics and Design" (*ISLPED*)
7. Skillicorn, David B., Talia Domenico. *Models and languages for parallel computation ACM Computing Surveys*, 1998:123 s.38-50
8. Thomas Rauber, Gudula Rünge. 2010. *Parallel Programming: For Multicore and Cluster Systems*, ISBN 3-642-04817-X s. 94–95
9. Alfred V. Aho, Edward K. Blum (2011) *Computer Science: The Hardware, Software and Heart of It* ISBN 1-4614-1167-X s.156-166
10. Rauber, Thomas, Rünge Gudula. (2010) *Parallel Programming: For Multicore and Cluster Systems* ISBN 3-642-04817-X s.94–95
11. Kay A. R., Robbins Steven (2003), *UNIX Systems Programming: Communication, Concurrency, and Threads* ISBN-13: 007-6092012177
12. Menken Ivanka, (2010) *Cloud Computing Virtualization Specialist Complete Certification Kit* ISBN:1742444539 9781742444536 s.201

13. Docker, <http://docker.com/open-source> Erişim: 29.08.2015
14. Raspberry Pi, <http://raspberrypi.org/help/what-is-a-raspberry-pi/> Erişim: 20.06.2015
15. Kickstarter <http://kickstarter.com/projects/1598272670/chip-the-worlds-first-9-computer> Erişim: 20.09.2014
16. Parallella Licence, <http://parallella.org/licenses/> Erişim: 16.02.2015
17. Parallella GitHub, <http://github.com/parallella/> Erişim: 16.02.2015
18. Parallella Supercomputing <http://github.com/parallella/parallella-hw> Erişim: 16.02.2015
19. Adapteva Parallella Manual, [http://parallella.org/docs/parallella\\_manual .pdf](http://parallella.org/docs/parallella_manual.pdf) Erişim: 20.01.2015
20. Parallella: A Supercomputer For Everyone <https://kickstarter.com/projects/adapteva/parallella-a-supercomputer-for-everyone> Erişim: 20.09.2014
21. Parallella Platform Reference Design, <http://adapteva.com/white-papers/parallella-platform-reference-design/> Erişim: 20.09.2014
22. Adapteva Parallella Board <http://adapteva.com/parallella-board/> Erişim: 20.09.2014
23. Hargrove, William W., M. Hoffman (1999). Cluster Computing: Linux Taken to the Extreme. Linux magazine s.1-3
24. Statistics TOP500 Cluster, TOP500 Supercomputer Sites <http://top500.org/statistics/sublist> Erişim: 11.09.2014
25. Linaro Limited "Linaro announces Actions Technology as a founding member of the Linaro Community Boards Group". Linaro.org Erişim: 20.01.2015
26. Üç farklı sanallaştırma tekniğine giriş [http://wiki.openvz.org/Introduction\\_to\\_virtualization](http://wiki.openvz.org/Introduction_to_virtualization) Erişim: 31.12.2014
27. Rami Rosen, Resource Management <http://www.cs.ucsb.edu/~rich/class/cs290-cloud/papers/lxc-namespace.pdf> Erişim: 20.01.2015

28. Lammell, R., (2008). Google's MapReduce programming model — Revisited. Science of Computer Programming s.1–30
29. Dean, Jeffrey & Ghemawat, Sanjay (2004). "MapReduce: Simplified Data Processing on Large Clusters" <http://static.googleusercontent.com/media/research.google.com/tr//archive/mapreduce-osdi04.pdf> Erişim: 23.01.2015
30. MapReduce-MPI MapReduce-MPI Library <http://mapreduce.sandia.gov/> Erişim: 20.01.2015
31. Awadallah Amr, Ders Sunumu, Introducing Apache Hadoop The Modern Data Operating System <http://web.stanford.edu/class/ee380/Abstracts/111116-slides.pdf> Erişim: 20.03.2015
32. Hadoop Map/Reduce <http://wiki.apache.org/hadoop/ProjectDescription> Erişim: 01.03.2015
33. Hadoop The Architectural Center of Enterprise Hadoop <http://hortonworks.com/hadoop/yarn/> Erişim: 02.03.2015
34. Lam, Chuck (2010). *Hadoop in Action*. ISBN 1-935182-19-6 s.325
35. White, Tom (2009). *Hadoop: The Definitive Guide O'Reilly Media*. 524. ISBN 0-596-52197-9. s.524
36. Hadoop HDFS Architecture <http://hadoop.apache.org/docs/r2.7.0/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html> Erişim: 15.02.2015
37. Hadoop Yarn <http://hadoop.apache.org/docs/r2.7.0/hadoop-yarn/hadoop-yarn-site/YARN.html> Erişim: 07.03.2015
38. Jeffrey Jeste,(2012) MapReduce Job Processing <http://www.cs.utah.edu/~lifeifei/cs6931/MRJob.pdf> Erişim: 20.12.2014
39. 39\_Hadoop/PoweredBy <http://wiki.apache.org/hadoop/PoweredBy> Erişim: 10.03.2015
40. CoreMark Scores <http://eembc.org> Erişim: 26.03.2015
41. Redefining Datacenter Efficiency <http://sc12.supercomputing.org/sites/default/files/SC12PM3.pdf> Erişim: 26.03.2015

42. Armservers <http://armservers.com/2012/06/18/apache-benchmarks-for-calxeda-5-watt-web-server> Erişim: 01.01.2015
43. SpecPower [http://spec.org/power\\_ssj2008/results/res2011q3/power\\_ssj2008-20110806-00392.html](http://spec.org/power_ssj2008/results/res2011q3/power_ssj2008-20110806-00392.html) Erişim: 7.01.2015
44. Accenture, Hadoop Deployment Comparison Study, <http://www.accenture.com/sitecollectiondocuments/pdf/accenture-hadoop-deployment-comparison-study.pdf> Erişim: 20.20.2014
45. Accenture, Cloud-based Hadoop Deployments <http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Cloud-Based-Hadoop-Deployments-Benefits-and-Considerations.pdf> Erişim: 20.10.2014
46. A Parallella Hardware Update <https://parallella.org/2013/08/21/parallella-hardware-update-s.47> Erişim: 16.02.2015
47. High Speed Serial, <http://xilinx.com/products/technology/high-speed-serial.html> Erişim: 02.04.2015
48. Multi-Gigabit Transceiver, [http://en.wikipedia.org/wiki/Multi-gigabit\\_transceiver](http://en.wikipedia.org/wiki/Multi-gigabit_transceiver) Erişim: 20.01.2015
49. Hadoop TeraSort <http://hadoop.apache.org/docs/current/api/org/apache/hadoop/examples/terasort/package-summary.html> Erişim: 17.04.2015
50. O'Malley Owen (2008) TeraByte Sort on Apache Hadoop <http://sortbenchmark.org/YahooHadoop.pdf> Erişim: 20.05.2015
51. Posamentier, Alfred S.; Lehmann, Ingmar (2004). *Pi: A Biography of the World's Most Mysterious Number*, ISBN 978-1-59102-200-8. s.105-108
52. Arndt, Jörg; Haenel, Christoph (2006). *Pi Unleashed*. ISBN 978-3-540-66572-4
53. Yong Zhao, Raicu I., Lu S. (2008); "Cloud Computing and Grid Computing 360-Degree Compared"; Grid Computing Environments Workshop.
54. Coulouris, George.Jean Dollimore; Tim Kindberg; Gordon Blair. (2011) *Distributed Systems: Concepts and Design* Boston: Addison-Wesley. ISBN-13: 978-0132143011

55. Sidd Satija, Soni, Aditya What is Cloud Computing Technology? Open Cloud Manifesto <http://opencloudmanifesto.org> Eriřim: 20.01.2015
56. Blem Emily, Jaikrishnan M., Karthikeyan S.(2013); “Power Struggles: Revisiting the RISC vs. CISC Debate on Contemporary ARM and x86 Architectures”; IEEE Int.Sym.on High Performance Computer Architecture (HPCA 2013) <http://research.cs.wisc.edu/vertical/papers/2013/hpca13-isa-power-struggles.pdf> Eriřim: 21.03.2015
57. Seal, David. *ARM Architecture Reference Manual*, (2001) ISBN-13: 078-5342737196
58. Kaplan, Yasin, Bulut Biliřim ve İř Sreklilięi, *Telepati Telekomnikasyon*, 2010: 183.
59. Korkmaz, Yakup (2008). Bulut Biliřim: Trkiye İin Fırsatlar TBİTAK – UEKAE
60. Xilinx Forum, <http://forums.xilinx.com/t5/General-Technical-Discussion/bd-p/GenDis> Eriřim: 19.08.2015
61. Yu Liu, Zhenjiang Hu, Kiminori Matsuzaki.Towards Systematic Parallel Programming over MapReduce <http://research.nii.ac.jp/~hu/pub/europar11.pdf> Eriřim: 25.05.2015
62. Erdem Bayer, Kap Temelli zgr Sanallařtırma zmleri Sunumu Akademik Biliřim Konferansı, <http://seminer.linux.org.tr> Eriřim: 23.01.2015

## **EKLER**

<b>1. Ek-A</b> : Linaro Kurulumu.....	51
<b>2. Ek-B</b> : LXC Kurulumu.....	55
<b>3. Ek-C</b> : Hadoop Kurulumu.....	56

## Ek-A: Linaro Kurulumu

Parallella Tek kart bilgisayar üzerinde linaro işletim sistemi olarak kullanılmıştır. [releases.linaro.org/14.06/ubuntu/trusty-images/developer](https://releases.linaro.org/14.06/ubuntu/trusty-images/developer) adresinden indirilen `linaro-trusty-developer-20140623-671.tar.gz` sd kartlara boot edilebilir halde yazılarak kurulum yapılmıştır.

```
wget releases.linaro.org/14.04/ubuntu/saucy-images/nano/linaro-saucy-nano-20140410-652.tar.gz
```

Sd kartlarda şu işlemler gerçekleştirilmiştir.

```
#!/bin/bash
parted -s /dev/mmcbk0 mklabel msdos
#) mklabel msdos
parted -s /dev/mmcbk0 mkpart p fat32 2048s 256MB
parted -s /dev/mmcbk0 mkpart p ext2 256MB -- -2MB

sudo mkfs.vfat -n BOOT /dev/mmcbk0p1
sudo mkfs.ext4 -L rootfs /dev/mmcbk0p2
md5sum linaro-saucy-nano-20140410-652.tar.gz
cded44187fb762ba7ea7b1ee092a0998 linaro-saucy-nano-20140410-652.tar.gz
sudo parted -l
```

2GB SD Card olanlar için  
Model: SD/MM Reader  
Disk /dev/sdX: 2042MB  
Sector size (logical/physical): 512B/512B  
Partition Table: msdos

```
Number Start End Size Type File system Flags
1 1049kB 49.3MB 48.2MB primary fat16
2 49.3MB 2042MB 1992MB primary ext4
"/dev/sdX" olan SD Card adı
sudo parted -a optimal /dev/sdX
(parted) print
```

```
Number Start End Size Type File system Flags
1 1049kB 99.6MB 98.6MB primary fat16 lba
2 99.6MB 2042MB 1942MB primary
```

(parted) mklable msdos

Warning: The existing disk label on /dev/sdX will be destroyed and all data on this disk will be lost. Do you want to continue?

Yes/No? yes

yeni partition oluşmak için

(parted) mkpart p fat32 2048s 256MB

(parted) mkpart p ext2 256MB -2MB

(parted) q

partition doğrulaması için

sudo fdisk -l /dev/sdX

Device Boot Start End Blocks Id System

/dev/sdX1 2048 499711 248832 c W95 FAT32 (LBA)

/dev/sdX2 499712 3887103 1693696 83 Linux

Biçimlendirmek için

sudo mkfs.vfat -n BOOT /dev/sdX1

sudo mkfs.ext4 -L rootfs /dev/sdX2

rootfs extract ile açmak için

sudo mkdir /tmp/mnt1 /tmp/mnt2

sudo mount /dev/sdX2 /tmp/mnt2

sudo tar --strip-components=1 -C /tmp/mnt2 -xzf linaro-saucy-nano-20140410-652.tar.gz

sudo sync

sudo umount /tmp/mnt2

bootfs extract ile açmak için

sudo mount -o umask=000 -t vfat /dev/sdX1 /tmp/mnt1

sudo tar --no-same-owner -C /tmp/mnt1 -xzf [kernelad]

sudo cp parallella.bit.bin /tmp/mnt1/

sudo sync

sudo umount /tmp/mnt1

Gereksiz paketleri kurmamak için

sudo mount /dev/sdX2 /tmp/mnt2

sudo nano /tmp/mnt2/etc/apt/apt.conf.d/00InstallRecommends

bu kısmı ekleyip saklama gerekli

APT::Install-Recommends "false";

Eğer router IP adres 192.168.0.1 ve Parallella IP address 192.168.0.2 ise

sudo nano /tmp/mnt2/etc/network/interfaces

ekleyip saklamalı

source-directory /etc/network/interfaces.d

auto lo



```
iface lo inet loopback
auto eth0
# static network settings
iface eth0 inet static
address 192.168.0.2
netmask 255.255.255.0
gateway 192.168.0.1
up sleep 3; mii-tool -F 1000baseT-FD
sudo nano /tmp/mnt2/etc/resolv.conf
nameserver 192.168.0.1
sudo sync
sudo umount /tmp/mnt2
```

SD kartı Parallellaya yerleşirip, güç verip, 30 saniye bekleyip, root terminal gelmesini bekliyoruz.

```
openssh-server kurulumu için Parallella tarafında
apt-get update
apt-get install openssh-server
sync
```

```
pc'den bağlanılabilir durumda
ssh linaro@192.168.0.2
password: linaro
```

```
cihaz bağlantı noktaları için
sudo mknod -m 600 /dev/ttyPS0 c 251 0
sudo chown root:tty /dev/ttyPS0
```

```
free
total used free shared buffers cached
Mem: 992452 38356 954096 0 1816 13076
-/+ buffers/cache: 23464 968988
Swap: 0 0 0
```

```
bellek harcaması 23mb
df
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/root 30348232 311108 28472452 2% /
none 99248 68 99180 1% /run
none 5120 0 5120 0% /run/lock
none 496224 0 496224 0% /run/shm
none 102400 0 102400 0% /run/user
```

toplam disk harcaması 304mb

```
sudo apt-get install vim-tiny nano fake-hwclock wget less unzip
sudo apt-get install build-essential man-db manpages-dev git libmpc-dev libmpc2
libgmp3-dev libmpfr-dev ca-certificates
wget http://ftp.parallella.org/esdk/esdk.5.13.09.10_linux_armv7l.tgz
```

```
md5sum esdk.5.13.09.10_linux_armv7l.tgz
sudo mkdir -p /opt/adapteva/
sudo tar xzf esdk.5.13.09.10_linux_armv7l.tgz -C /opt/adapteva/
sudo ln -sTf /opt/adapteva/esdk.5.13.09.10 /opt/adapteva/esdk
```

```
touch ~/.nano_history
nano ~/.bashrc
```

```
sudo mknod -m 660 /dev/mmcblk0 b 179 0
sudo mknod -m 660 /dev/mmcblk0p1 b 179 1
sudo mknod -m 660 /dev/mmcblk0p2 b 179 2
sudo apt-get install device-tree-compiler nano
sudo mount /dev/mmcblk0p1 /mnt
cd /mnt
sudo cp devicetree.dtb devicetree.dtb.bak
sudo dtc -I dtb -O dts -o devicetree.dts devicetree.dtb
sudo nano devicetree.dts
bootargs = "root=/dev/mmcblk0p2 rw rootfstype=ext4 rootwait";
sudo dtc -I dts -O dtb -o devicetree.dtb devicetree.dts
cd /
sync
sudo reboot
```

LXDE kurulumu için

```
sudo apt-get install alsa-base alsa-utils libasound2-plugins lxde x11-xserver-utils
xserver-xorg xserver-xorg-video-fbdev xserver-xorg-video-modesetting feh xinit
hızlı görüntüleme için
```

```
sudo nano /etc/xdg/lxsession/LXDE/autostart
```

normal kullanıcı olarak giriş boot ettikten sonra Ctrl+Alt+F2

login: linaro

Password: linaro

su linaro

cd

Start X environment

startx

## Ek-B: LXC Kurulumu

sudo apt-get install lxc cgroup-lite libvirt-bin libcap-dev

Her LXC ayarlanırken her biri NodeName ve ikincisi SecondaryNodeName, ve her ikisi de DataNode olarak ayarlanmıştır. Shell script ile şu şekilde yapılmıştır:

```
#!/bin/bash
#-set-Variables
LXC_NAME="LXC303"
userN="linaro"
password="linaro"
#--

#-Get IP Address of LXC
if subutai clone master $LXC_NAME | grep Successfully;then
echo "Successfully created $LXC_NAME"
else
echo "ERROR in creation"
exit;
fi
IP_LXC=$(lxc-ls -f | grep $LXC_NAME | awk -F" " '{print $3}')
if [ $IP_LXC == "-" ]; then
echo "EMPTY IP ADDRESS. Let's try once more";
echo "IP: $IP_LXC"
# exit;
fi
sleep 5;
IP_LXC=$(lxc-ls -f | grep $LXC_NAME | awk -F" " '{print $3}')
if [ $IP_LXC == "-" ]; then
echo "EMPTY IP ADDRESS";
echo "IP: $IP_LXC"
exit;
fi
#-
ssh-keyscan -H $IP_LXC >> ~/.ssh/known_hosts
pass="ubuntu"
./ssh-set.sh $userN $pass $IP_LXC
./send_pub_keys.sh $userN $IP_LXC./Owner_id_rsa.pub
#-
```

Tüm yapılandırmaların doğruluğunu görmek için

sudo lxc-checkconfig

Başlatmak için

sudo lxc-start -n mycontainer

Kullanmak üzere bağlanmak istenildiğinde

sudo lxc-attach -n mycontainer ve ya sudo lxc-console -n mycontainer

## Ek-C: Hadoop Kurulumu

Parallella Tek kart bilgisayar üzerinde Hadoop kurulumu deb paket ve shell script ile yapılmıştır.

1. HDFS okuma yazma hızı dfs.replication parametresi ile değişkenlik göstermektedir. dfs.replication=2 olarak kullanılmıştır.
2. Parallella üzerindeki /etc/hosts dosyası açılarak hostnames ve küme içindeki diğer düğümlerin ipleri girilmiştir. Her bir node için aynı işlemler tekrar edilmiştir. Tablo 3.5 yapılandırması uygulanmıştır.
3. Tüm düğümlere kullanıcı ve şifre olarak 'subutay' ' subutay' ile erişilebilir duruma getirilmiştir, superuser \$ sudo su ile superusera geçilmiştir.
4. Her Parallella için public key public key ~/.ssh/authorized\_keys dosyasına yazılmıştır. Public herhangi bir kullanıcının public key ~/.ssh/id\_\_\_\_.pub olarakta yazılmıştır.
5. Kümenin NameNode ayarlarında slave olan Parallella ayarları yazılmıştır. \$hadoop-master-slave.sh içinde slave hostnameler belirtildi. Tüm bunları yapınca herhangi bir node üzerinden tüm kümeyi başlatıp sonlandırabilir hale gelmiştir.
6. Parallella üzerindeki hadoop ayarlarını değiştirmek için, Küme içindeki NameNode ve JobTracker ayarları core-site.xml ve mapred-site.xml içine yazıldı. Portlar 8020 ve 9000olarak bırakılmıştır.
7. NameNode başlatmak için: \$ service hadoop-dfs start  
JobTracker başlatmak için: \$ service hadoop-mapred start

veya basit olarak yönetim arayüzünde da yer alan Start DataNode/JobTracker butonu da kullanılabilir.

1. Test görev otomasyon aracı ile otomatik olarak başlatıldı. iş başlatıldığında aşağıdaki Hadoop test sonuçları hem olağan çıktı hemde HDFS kısmına yazılmıştır.

Hadoop yapılandırması şu şekildedir

Parallella 1	NameNode(master, slave)
Parallella 2	SecondaryNameNode(slave)
Parallella 3	DataNode(slave), DataNode(slave)
Parallella 4	DataNode(slave), DataNode(slave)

```
<cluster>
<clustertype>hadoop21</clustertype>
  <node>
    <rh>linaro1</rh>
    <hostname>hadoop2</hostname>
    <login>daralbaev</login>
    <password>ubuntu</password>
    <arch>amd64</arch>
    <device>mac2</device>
    <nodetype>SecondaryNameNode</nodetype>
    <ip>10.0.33.101</ip>
    <mask>255.255.255.0</mask>
    <gw>10.0.33.1</gw>
  </node>
  <node>
    <rh>linaro1</rh>
    <hostname>hadoop8</hostname>
    <login>daralbaev</login>
    <password>ubuntu</password>
    <arch>amd64</arch>
    <device>mac2</device>
    <nodetype>DataNode</nodetype>
    <ip>10.0.33.107</ip>
    <mask>255.255.255.0</mask>
    <gw>10.0.33.1</gw>
  </node>

  <node>
    <rh>linaro2</rh>
    <hostname>hadoop3</hostname>
    <login>daralbaev</login>
    <password>ubuntu</password>
    <arch>amd64</arch>
    <device>mac2</device>
    <nodetype>NameNode</nodetype>
    <ip>10.0.33.102</ip>
    <mask>255.255.255.0</mask>
    <gw>10.0.33.1</gw>
  </node>
  <node>
    <rh>linaro2</rh>
    <hostname>hadoop9</hostname>
    <login>daralbaev</login>
    <password>ubuntu</password>
    <arch>amd64</arch>
```

```

<device>mac2</device>
<nodetype>DataNode</nodetype>
<ip>10.0.33.108</ip>
<mask>255.255.255.0</mask>
<gw>10.0.33.1</gw>
</node>

<node>
<rh>linaro3</rh>
<hostname>hadoop4</hostname>
<login>daralbaev</login>
<password>ubuntu</password>
<arch>amd64</arch>
<device>mac2</device>
<nodetype>DataNode</nodetype>
<ip>10.0.33.103</ip>
<mask>255.255.255.0</mask>
<gw>10.0.33.1</gw>
</node>
<node>
<rh>linaro4</rh>
<hostname>hadoop5</hostname>
<login>daralbaev</login>
<password>ubuntu</password>
<arch>amd64</arch>
<device>mac2</device>
<nodetype>DataNode</nodetype>
<ip>10.0.33.104</ip>
<mask>255.255.255.0</mask>
<gw>10.0.33.1</gw>
</node>

<rh>linaro3</rh>
<hostname>hadoop6</hostname>
<login>daralbaev</login>
<password>ubuntu</password>
<arch>amd64</arch>
<device>mac2</device>
<nodetype>DataNode</nodetype>
<ip>10.0.33.105</ip>
<mask>255.255.255.0</mask>
<gw>10.0.33.1</gw>
</node>
<node>
<rh>linaro4</rh>
<hostname>hadoop7</hostname>
<login>daralbaev</login>
<password>ubuntu</password>
<arch>amd64</arch>
<device>mac2</device>
<nodetype>DataNode</nodetype>
<ip>10.0.33.106</ip>
<mask>255.255.255.0</mask>
<gw>10.0.33.1</gw>
</node>
</cluster>

```

## ÖZGEÇMİŞ

### KİŞİSEL BİLGİLER

Ad Soyad : Levent Aysan  
Uyruk : TC  
Doğum Tarihi Yer : 1977 Bursa  
İletişim : leventaysan@hotmail.com

### EĞİTİM

2001 Yıldız Teknik Üniversitesi, Bilgisayar Mühendisliği  
2015 Türk Hava Kurumu Üniversitesi, Bilişim Teknolojileri

### MESLEKİ DENEYİM

1998-2011 Teknik Danışman  
2011-2015 Yönetim Danışmanlığı

### YABANCI DİL

İngilizce, Rusça, Türk Lehçeleri

### YAYINLAR

Özbilgin İ.G. ve Aysan L.: “Tek Kart Bilgisayarlar ile Bulut Oluşturarak MapReduce İşlemleri Denemesi” 10.17671/BTD.88292, 179-189 (2015)