

Tek Kart Bilgisayarlar ile Bulut Oluşturarak MapReduce İşlemleri Denemesi

Levent AYSAN¹, İzzet Gökhan ÖZBİLGİN¹

¹Bilişim Teknolojileri Anabilim Dalı, Türk Hava Kurumu Üniversitesi, Ankara, Türkiye

leventaysan@hotmail.com, gozbilgin@thk.edu.tr

(Geliş/Received: 22.07.2015; Kabul/Accepted: 09.09.2015)

DOI: 10.17671/BTD.88292

Özet- 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 ARM işlemci kümeleri üzerinde yürütülmesini ve etkinliğinin test edilmesi araştırılmıştır. ARM işlemcili tek kart mini bilgisayarların maliyeti ucuz, enerji tüketimi düşük, karbon salınımı düşüktür. Kümeleme, bulut bilişim, çoklu işlem, paralel işlem ve büyük veri uygulamalarına da uygunluğu da görülmüştür. Tek kart bilgisayar donanımı üzerinde kap temelli sanallaştırma kullanımını denenmemiş bir yaklaşımdır. MapReduce uygulamasında işçi düğüm olarak işlem tecritlenmesi kullanılması da yeni bir uygulamadır.

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

Study on MapReduce Operations Creating Cloud with Single Board Computer

Abstract- 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.

Keywords- single board computer cluster, virtualization, big data, MapReduce, Hadoop

1. GİRİŞ (INTRODUCTION)

Günümüzde öne çıkan bir terim haline gelmiş olan bulut bilişim veya işlevsel anlamıyla çevrim içi bilgi dağıtımı; bilişim aygıtları arasında ortak bilgi paylaşımını sağlayan hizmetlere verilen genel ad olarak tanımlanabilmektedir. Bulut Bilişim, kolay bir şekilde yapılandırılan, hızlı bir şekilde sunuma hazır hale getirilen ve asgari emekle yönetilerek servis sağlayıcı etkileşimine ihtiyacın en aza indirildiği, paylaşımlı bir bilişim kaynakları (Ağ, sunucular, diskler, uygulamalar ve servisler) havuzuna, talep üzerine erişimi olanaklı kılan bir iş modeli olarak karşımıza çıkmaktadır [1]. Bulut bilişim, düşük yönetim çabası veya servis sağlayıcı etkileşimi ile, hızlı alınıp salıverilebilen ayarlanabilir bilişim kaynaklarının paylaşılır havuzuna, istendiğinde ve uygun bir şekilde ağ erişimi sağlayan bir model olmaktadır [2]. Bulut bilişim bu yönüyle bir ürün değil, hizmettir; temel kaynaktaki yazılım ve bilgilerin paylaşımı sağlanarak, mevcut bilişim hizmetinin bilgisayarlar ve diğer aygıtlardan elektrik dağıtıcılarına benzer bir biçimde bilişim ağı üzerinden kullanılmaktadır.

Bulut sözcüğü dosyaların sağlandığı konumu işaret etmektedir. İşleme ve saklama konumlarının aynı aygıtta bulunması durumu klasik bir algı olsa dahi saklama boyutu bulutlara yani saklama ve altyapı hizmeti barındıran hizmetlere doğru kaymaktadır. Bulut bilişim tamamen yeni bir kavram değil nispeten yeni ancak onlarca yıldır kullanılan ızgara hesaplama paradigması, küme hesaplama ve genel olarak dağıtık sistemler gibi diğer teknolojilerle de iç içe bağlantısı vardır[3]. Günümüzde büyük veri sistemlerine ait işlemler yüksek güç tüketimine sahip CISC tabanlı işlemcilerle donanım kümeleri üzerinde yürütülmektedir[4]. Temelde büyük veri işlemleri göz önünde bulundurularak tasarlanmamış bu donanımlar, bu ve birçok farklı nedenden bu işlemlerdeki başarıyı kısıtlamakta ve maliyetleri artırmaktadır.

ARM mimarisi RISC tabanlı bir işlemci mimarisidir, günümüzde 32 bit ve 64 bit sürümleri vardır, genel itibarıyla düşük güç tüketimi, diğer işlemcilere göre yüksek performanslı oluşu ve x86-x64 işlemcilere göre daha uygun fiyatlı olmasından dolayı gömülü sistemlerde, taşınabilir aygıtlarda kullanılan yonga kümelerinde genelde ARM işlemci tercih edilmektedir[4]. ARM firması kendi başına işlemci üretmeyen, tasarımı, lisans, lisans kullanım hakkı satan bir oluşumdur[5]. CISC mimarisinde işlemci yapısı her kuşak işlemci ile beraber daha karmaşıklaşmıştır. CISC mimarisinde özel olarak tasarlanmış olan komutlar yeteri kadar sık kullanılmamaktadır. Tipik bir programda mevcut komutların çok azı kullanılmaktadır. Farklı komutlar farklı sayıda saat çevrimine gerek duyacaklarından performans düşmesi gözlenir. Tüm bu nedenlerden dolayı RISC mimarisi ortaya çıkmıştır. Azaltılmış komut kümesi sayesinde daha hızlı çalışırlar. RISC işlemcilerin komut kümeleri basitleştirildiklerinden çok az yonga kullanırlar.

CISC işlemcilere göre daha çabuk tasarlanabilirler Bu nedenle işlemci jenerasyonları ve karakteristikleri incelenirken ARM referans tasarımı ele alınmaktadır [6].

Açık kaynak kodlu bulut uygulamalarının çalışmasını ve MapReduce işlemlerinin özel olarak tasarlanmış işlemci kümeleri üzerinde yürütülmesini ve etkinliğinin test edilmesi ile bu maliyeti ucuz ve enerji tüketimi, karbon salınımı düşük ARM işlemcili yüksek veri hızlarına sahip bellek ve veri yoluna sahip küçük boyutlu tek kart bilgisayardan oluşmuş kümeler üzerinde bulut bilişimde kullanımı gözlemlenmektedir.

Tek kart bilgisayarlar günümüzde birçok değişik yapılandırma ve özelliklere sahip olabilmektedir. Kısıtlı bir kısmında yardımcı işlemci ve alanda programlanabilir kapı dizileri ile de yapılandırılmış olabilmektedir.

Makalenin bundan sonraki bölümlerinde İlgili Çalışmanın ve diğer çalışmalardan farkı, Mevcut Durum, Tek Kart Bilgisayar İle Bulut Oluşturma, Çalışma Çıktıları Eki ile Değerlendirme ve Sonuç kısımları yer almaktadır.

2. YAPILAN ÇALIŞMALAR (RELATED WORKS)

Değişik tek kart bilgisayarlar ile bir çok küme çalışmaları yapılmıştır. Tek kart üzerinde MapReduce işlemleri Hadoop ve Sector/Sphere ürünleri kullanımı denenmiş çalışmalardır. Çeşitli sunucuların kümelenmesinde sanallaştırma ve daha yeni yaklaşım olarak işletim seviyesi sanallaştırma ile ilgili işlem tecritleme yöntemleri kullanılmaktadır. Docker ürünü ile bu işlem kap temelli sanallaştırma bulut bilişimde yeni bir döneme girilmesini sağlamıştır. Artık bu tarz işlem tecritleyerek sanallaştırma işine uygun değişik ihtiyaçlara yönelik sığa dosyaları bulut hizmeti sağlayıcıların tarafından kullanıma sunulmuştur.

Çalışmamızın diğerlerinden ayrılan asıl iki temel özelliğinden birincisi tek kart bilgisayar donanımı üzerinde kap temelli sanallaştırma kullanımı iken ikincisi ise MapReduce işleminde işçi düğüm olarak işlem tecritlenmesi için LXC kullanılmasıdır.

Çalışmanın temel gözlemlenmesi amaçlanan adımları olarak şunlar sıralanabilir.

1. Tek kart bilgisayar kullanımı
2. Tek kart bilgisayarlara UNIX tabanlı işletim sisteminin kurulması
3. Küme oluşturulması
4. Küme üyesi bilgisayarlar üzerinde sanallaştırma yapılması
5. Sanal makinelerle Hadoop kurulumu yapılması MapReduce işlemlerine dayanan paralel işlemlerin ve uygulamaların çalışmasını sağlanması [7]
6. Büyük Veri işlemlerinin ana omurgasını oluşturan MapReduce içindeki adresleme ve indeksleme işlemlerini denemeye yarayan

- TeraValidate, TeraGen, TeraSort [8] uygulamalarının sınanması
7. Sanal makinelerle bir takım bulut uygulamaları kullanılarak sınanması

MapReduce dağıtık mimari üzerinde çok büyük verilerin kolay bir şekilde analiz edilebilmesini sağlayan bir sistemdir. Veriler işlenirken eşle ve indirge olarak temel iki fonksiyon kullanılmaktadır. Eşleme aşamasında ana düğüm verileri alıp daha ufak parçalara ayırarak işçi düğümlere dağıtırken işçi düğümler işleri tamamladıkça sonucunu ana düğüme geri gönderilmektedir. Reduce aşamasında ise tamamlanan işler birleştirilerek sonuç elde edilmektedir. Eşleme aşamasında analiz edilen veri içerisinden almak istediğimiz veriler çekilmekte, Reduce aşamasında ise bu çektiğimiz veri üzerinde istediğimiz analiz gerçekleştirilmektedir[9]. 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. Hadoop, arka plandaki kümelenmiş bilgisayarların üstüne kurguladığı uygulama seviyesindeki çözümüyle farklı, ucuz bir alternatif olmayı hedeflemektedir.

Donanım olarak 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 menzilini kısıltmaktadır. Bu nedenle aygıtın görev sonunda yer istasyonuna indikten sonra üzerindeki veriler sunuculara aktararak 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.

Kümeleme yapılmaya elverişliliği, işletim sistemi desteği, açık kaynak kod desteği, düşük enerji tüketimleri, paralel dağıtık işlem yapma yeteneği ile Adaptea markasının Parallella [10] ürününün işletim sistemi seviyesi sanallaştırma ile MapReduce işlemlerine de uygunluğu gözlemlenmektedir.

2.1. Mevcut Durum(Current Situation)

Güncel sistemlerle, tek kart bilgisayarlı sistemin gerçek karşılaştırma değerlerinin yeterince isabetli şekilde elde edilmesi yeni bir melez mimarinin, bellek ve veri erişiminde yapılacak iyileştirmelerin varlığı ve tamamen eşlenik olmaması nedeniyle karşılaştırma olanaklı değildir. Ancak uygulama için ARM işlemci ile yaygın olarak kullanılan Intel Xeon işlemcilerin başarımları, güç tüketimi, ısı karakteristikleri, maliyet karşılaştırmaları yapılabilir. Bu konuda gerek akademik, gerekse ticari birimlerin yapmış olduğu birçok ve çeşitli karşılaştırma

bulunmaktadır[11]. Bu karşılaştırmalar incelendiğinde, ARM işlemcilerin hem (işlem/güç) ölçütünde hem de (işlem sığası/Maliyet) ölçütünde Xeon işlemcileri ile (Tablo 1) de görüldüğü üzere 15 kat geride bıraktığı gözlenmektedir [12]. ARM ve Intel örnek işlemci güç tüketimleri Tablo 2 de yer almaktadır.

Tablo 1. İşlemci ve Yonga üzeri sistem karşılaştırması[13].
(Processors and SOC Comparison)

	Calxeda Energy Core ECX-1000	Intel Xeon E3-1240
Çekirdek Frekansı	1, 1 GHz	3, 3 GHz
Mikroişlemci Çekirdekleri	4	4
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
Başarım/W üstünlüğü	15X	

Tablo 2. ARM ve Intel örnek işlemci kıyaslaması.
(ARM and Intel sample processor comparison)

	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 başarımları testleri ve karşılaştırmaları, yeterli olmayacaksa da, sistemin hedeflediği ölçütlere bir referans teşkil etmektedir [14]. Accenture şirketinin büyük veri sistemleri karşılaştırma referansı [15] olarak kullandığı yapılandırma (Tablo 3) görülmektedir.

Tablo 3. Accenture Karşılaştırma Yapılandırması
(Accenture Reference Configuration)

Çalışan/Veri Düğümü Özeti	
Model	Dell R415
Mikroişlemci Tipi	Opteron 4180
Çekirdek Sayısı	12
Çekirdek Hızı (GHz)	2, 6

Çalışan/Veri Düzümü Özeti	
Bellek (GB)	24
Bellek veri yolu hızı	1333 MHz
Her diskin sığası (TB)	2
Toplam Sığa (TB)	8
Toplam Küme	
İstemci Düzümleri	1
Birincil Düzümler	3
Çalışan/Veri Düzümleri	22
Çekirdek Sayısı	264
Bellek (GB)	528

Verilen 24-düzüm 50 TB sığalı Apache Hadoop kümesi yapılandırmasının, enerji, bakım, soğutma gibi kalemlere sahip toplam sahip olma maliyeti aylık 2.914, 58\$ olarak verilmiştir [16]. 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.

2.2. Tek Kart Bilgisayar İle Bulut Oluşturma (Cloud Setup With Single Board Computers)

Çalışmamızda düğüm donanımları için piyasada hâlihazırda bulunan Zynq-7020 temelli yonga üzeri sistemler içeren Parallella kullanılmıştır. Parallella merkezi işlemci ARM çift çekirdekli Cortex işlemci ürünü Xilinx 28nm programlanabilir mantık ile işleme sistemli Zynq SOC kümesini içerir [17]. 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[18],[19].

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 çip iç ağında hızlı bir şekilde birbirine bağlanır. Xilinx firmasının açık kaynak dünyasına teknoloji desteği [20] sayesinde, teknolojik yenilikler alınabilmektedir.

Linaro oluşumu ARM tabanlı işlemci ve kart üreten ve pazarlayan firmaların bir araya gelip ürettikleri donanımların üstünde, işletim sistemi, işletim sistemi çekirdeği ve yan yazılım ihtiyaçları için ortak çözüm oluşturmaya çalışan bir organizasyon olup kar amacı gütmemektedir. İşletim sisteminde Android ve Ubuntu Linux üzerinde değişiklik ve uygunluk çalışmaları yaparak bunları sağlamaktadırlar. Yeni çıkan ARM kart

üzeri işlemcileri destekleyen linux çekirdek oluşturmak, çoklu ortam, güç yönetimi, grafik ve güvenlik desteği ve bu tarz yazılım parçalarını oluşturmak için gerekli ürün zincirlerini de derlemek, hazırlamak, uyum testlerini yapmak gibi çalışmaları mevcuttur. ARM komut setlerinde 32 bit ve 64 bit çalışan Cortex işlemci ailelerine destekleri geliştirilmiştir [21]. Linaro işletim sistemi sayesinde ise çekirdek, sürücü ve bir çok gerekli temel yazılım hazır olarak mevcut olup kolaylıkla derlenebilmektedir.



Şekil 1. Dört adet Parallella tek kart bilgisayar ile oluşturulan küme görülmektedir.
(Showing cluster of four Parallella single board computers)

Düğüm 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, hâlihazırda var olan sistemlere uyarlama kolaylığı gibi kıstaslar göz önünde bulundurularak Debian/Ubuntu türevi bir işletim sistemi olan Linaro 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 Protocol, Aurora Protocol ve Interlaken Protocol 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 inanılması güç hızlara erişen çeşitli güncel uygulamaları mevcuttur [22]. 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 [23].

Parallella tek kart bilgisayarlar açık kaynak donanım olarak Adapteva firması tarafından üretilmektedir. Mimari olarak Xilinx firmasının Zebdboard ürününü temel almaktadır. (Şekil 2) de Xilinx üretimi Zynq-7020 ve Adapteva tarafından üretilen Ephemany yongası görülmektedir. Bu nedenle (Tablo 4) te belirtilen özelliklerin yanı sıra 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 çalışmamızda tercih nedeni olmaktadır.

Parallella tek kart bilgisayar açık kaynak donanım olup devre tasarım dosyaları Creative Commons Attribution-ShareAlike 3.0 lisansı altında yayınlanmaktadır. Ayrıca Alanda programlanabilir mantık devreleri için GPL HDL kaynaklarına haizdir. 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.



Şekil 2. Xilinx firmasının Zynq-7020 ve Adapteva firmasının Ephemany yongası kart üzerinde görülmektedir.

(Xilinx Zynq-7020 and Adapteva Ephemany chip seen on board)

Performans bir MapReduce işi çalıştırıldığında giren-çıkran veri miktarının azaltılması ve kullanılan işlemci miktarının düşürülmesidir. Bu mimarinin performansını görebilmek için hız testi yaparken Hadoop[24] dağıtımı kullanılmıştır. Özellikle giriş/çıkış performansı ve işlemci

zaman kullanımına önem verilmiştir.

Oluşturulan küme bir adet yönetim maksatlı kişisel bilgisayar ve (Şekil 1)de görüldüğü üzere dört adet Parallella içermektedir (7020, RAM 1GB) içermektedir.

Tablo 4. Zynq-7020 Özellikleri
(Zynq-7020 properties)

İşlemci Çekirdek	Dual ARM Cortex™-A9 MPCorewith CoreSight
İşlemci Uzatım	NEON & Single / Double Precision Floating Point for each processor
L1	32 KB Instruction, 32 KB Data per processor
L2	512 KB
Çip üstü Bellek	256 KB
Bellek Arayüz	DDR3, DDR3L, DDR2, LPDDR2, 2x Quad-SPI, NAND, NOR
Çevre Donanım	2x USB 2.0 (OTG), 2x Tri-mode Gigabit Ethernet, 2x SD/SDIO
Logic Hücre	85K
Blok RAM	560 KB
DSP Slices	220

Kullanılan Pc özellikleri şu şekildedir: x64, Intel® Core™ i7-4700MQ, Mikroişlemci @ 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 mikro boyutta güvenli sayısal hafıza kartı kullanılmaktadır. Sürücülerinin IO performansına olumsuz etkisi göz önüne alınarak USB Flash bellek kullanılmamıştır. Boardlar üzerinde ve bilgisayar üzerinde herhangi bir fazladan yapılandırma ve ayarlama yapılmamıştır.

HDFS sayesinde sıradan sunucuların diskleri bir araya gelerek büyük, tek bir sanal disk oluştururlar. Bu sayede çok büyük boyutta bir çok dosya bu dosya sisteminde saklanabilir. Bu dosyalar bloklar halinde (varsayılan 64MB) birden fazla ve farklı sunucu üzerine (varsayılan 3 kopya) dağıtılarak RAID benzeri bir yapıyla yedeklenir. Bu sayede veri kaybı önlenmiş olur. Ayrıca HDFS çok büyük boyutlu dosyalar üzerinde okuma işlemi (streaming) imkanı sağlar, ancak rastlantısal erişim özelliği bulunmaz. HDFS, NameNode ve DataNode işlem süreçlerinden oluşmaktadır

NameNode ana süreç olarak blokların sunucular üzerindeki dağılımından, yaratılmasından, silinmesinden, bir blokta sorun meydana geldiğinde yeniden oluşturulmasından ve her türlü dosya erişiminden sorumludur. Kısacası HDFS üzerindeki tüm dosyalar

hakkındaki bilgiler NameNode tarafından saklanır ve yönetilir. Her kümede yalnızca bir adet NameNode olabilir.

DataNode işlevi blokları saklamak olan işçi süreçtir. Her DataNode kendi yerel diskindeki veriden sorumludur. Ayrıca diğer DataNode'lardaki verilerin yedeklerini de barındırır. DataNode'lar küme içerisinde birden fazla olabilir.

Hadoop Yarn ise iş zamanlayıcı ve kaynak yönetimini yapan bir dizi kütüphane olup Hadoop işlerini zamanlamak ve küme kaynaklarını yönetmek için kullanılır. Hadoop 2.0 ile birlikte gelmiştir. Dağıtık uygulamaları çalıştırmak için kullanılan kaynakların yönetimini sağlamaktadır. 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 [25].

Kümenin üzerinde işletim sistemleri, sanallaştırma ve Hadoop kurulumu ile çeşitli deneme programı çalıştırılmıştır. Kullanılanlar şunlardır. DFSIO, TeraGen, TeraSort, TeraValidate, Ek-1 kurulum ve çıktıları göstermektedir

Tüm bu test kodları çalıştırıldığında çıktıları kayıt edilmiştir. Yaptığımız Test sonuçları Ek-1'de yer almaktadır. Bu işlemler tek kart bilgisayarların bulut davranışını görmemizi sağlamıştır.

TeraGen : MapReduce için veri oluşturur. TeraGen rastgele veriler oluşturmaktadır. Bu görevleri istenilen satır sayısına istenilen sayıda böler ve her map rastgele satır atar [26].

TeraSort : TeraSort işleminin amacı kümenin Bellek/Mikroişlemci 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 çıktı verir.

TeraValidate : Giriş dosyası olarak terasort çıktısını alıp sıralamanın doğru yapıldığının sağlamasını yapmaktadır.

TestDFSIO -read -write : 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 oluşturularak değişik boyut ve sayıdaki dosyalarla test edilmesi şeklinde kullanılmaktadır. Bu test gerçek hayattaki canlı sistemin aynısının iş yükünü modellemese dahi bu tek kart bilgisayar davranışını göstermiştir. Gerçeğe çok daha yakın olabilmesi için çok daha büyük boyutta düz metin dosyalar kullanmak gerekir.

2.3. Değerlendirme (Evaluation)

Yapılan uygulamada 2 tekkart bilgisayar, 3 tekkart bilgisayar, 4 tekkart bilgisayar, ardından 10 tekkart bilgisayara kadar kümeler oluşturulmuştur. Birbirlerini network switch üzerinden görür hale getirilmiştir. Üzerine ubuntu, debian ve linaro işletim sistemleri kurulmuştur. Linaro en hafif kurulum ile en uygun ve en az kaynak sarfiyatlı işletim sistemi olduğu gözlemlenmiştir.

2 paralella üzerine 2 LXC, 3 paralella üzerine 3LXC, 4 paralella üzerine 4LXC, 4 paralella üzerine 6LXC, 4 paralella üzerine 7LXC, 4 paralella üzerine 8LXC kurulu olan değişik kümelerin üzerinde sanallaştırma yapılmıştır.

Bu uygulamaların buluta uygunluğunun denenmesi için çeşitli uygulamalar üzerinde koşturulmuştur. Mikroişlemci, bellek, network ve işlem yoğunluğu açısından incelendiğinde tek kart bilgisayarlar ile yapılan bulut çabasında işlemci gücü açısından yetersizlik gözlemlenmemiştir. Sistem gözleme araçlarından işlemci meşgulliyeti değerlerine bakıldığında 70% üzerine çıktığı gözlemlenmemiştir. İşlemci değerlerinde bekleme, askıda kalma, cevap verememe olmamakta ancak bellek açısından yoğunluk yaşandığı gözlemlenmektedir.

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 artırılması için donanım yani düğüm sayısının artırılması 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 beklemenin 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 işlemci fazla kullanılmıştır.

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 i/o 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.

Dosya barındırma ve paylaşım için, yazılım test otomasyonu araçları için, dinamik içerikli web siteleri sunuculuğu için ve orta katman sunucuları için kullanılması uygun görülmektedir.

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

Uygulama için kullanılan boardlar 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 mimarının 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 gözükmemektedir. Bu düğümler arası bekleme zamanını azaltmakta ancak bu da üreticiden bağımsızlığın önüne engel getirmektedir.

Bu kartların bir kısmında fpga tümleşik olmasına rağmen fpga ile işlemci arası 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 [27].

3. SONUÇ (CONCLUSION)

Gittikçe kullanımı yaygınlaşan tek kart bilgisayarların kümeleme, sanallaştırma ve bulut işlemleri için de kullanılabilmesi gözlemlenmiştir. 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 gözlemlenmiştir.

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. Bu alanlarda yaygın olarak Intel Xeon gibi CISC tabanlı işlemciler kullanılmaktadır. Bu işlemcilerin güç tüketimleri, gerektirdikleri çevresel donanımlar ve soğutma gereksinimleri de dahil edildiğinde ortalama 1 kW düzeyine ulaşmaktadır. Dahası bu işlemcilerin kullanıldığı bilgisayarların bulunduğu basit bir sunucu odasından, Büyük 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, özellikle Türkiye gibi 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 işlemcilere dayanması, düğüm başına 5-10 watt aralığında bir güç tüketimini mümkün hale getirmektedir. 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.

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 ürünün kullanımı kolaylaşmakta, ürünün varlığı daha önce olanaklı olmayan yeni uygulamaların yapılabilmesini mümkün kılmaktadır. 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 (İnsansız hava aracı, gözetleme uyduları vs.) 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 bilgisayar kümelerinin 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.

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 gözlenerek. 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 çalıştırılabilinmektedir.

Ancak yaptığımız uygulamada solid state disk kullanımı ile de yapmak istediğimizde sürücü desteği sıkıntısı yaşanmıştır. Bu konuda uyarlanmış sürücü ihtiyacımızı giderilememiştir. 64 bit tek kart bilgisayarlar henüz yaygınlaşmamıştır, bu da bir çok kısıtı beraberinde getirmektedir.

Yapılan çalışmada MapReduce için Apache Hadoop yerine Sector/Sphere kullanılabilir. Kap temelli sanallaştırma için LXC yerine yeni işlem tecritleyici baştan yazılabilir. Konu ile ilgilenen araştırmacılara FPGA üzerinde MapReduce işlemlerinin gerçekleşmesi 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.

KAYNAKLAR (REFERENCES)

- [1] Y. Kaplan, Bulut Bilişim ve İş Sürekliliği, **Telepati Telekomünikasyon 183 Rapor**, Türkiye, 2010.
- [2] Y. Korkmaz, **Bulut Bilişim: Türkiye İçin Fırsatlar TUBİTAK – UEKAE**, Türkiye, 2008.
- [3] Y. Zhao, I. Raicu, S. Lu, **Cloud Computing and Grid Computing 360-Degree Compared**, Texas, ABD, 2008
- [4] B.Emily, M. Jaikrishnan, S. Karthikeyan, **Power Struggles: Revisiting the RISC vs. CISC Debate on Contemporary ARM and x86 Architectures**, Univ Wisconsin Madison, WI, ABD, 2013
- [5] Arm Company Profile, <http://arm.com/about/company-profile/>, 21.03.2015
- [6] Seal, David. **ARM Architecture Reference Manual**, (2001)
- [7] Y. Liu, H. Zhenjiang, K. Matsuzaki. **Towards Systematic Parallel**

- Programming over MapReduce** Conference Proceedings, 483-485, 25.05.2015
- [8] M. Owen, **TeraByte Sort on Apache Hadoop**, Kalifornia, ABD, 2008
- [9] R. Lammel, **Google's MapReduce programming model**, 2008
- [10] Adapteva Parallella Manual, http://www.parallella.org/docs/parallella_manual.pdf, 20.01.2015
- [11] CoreMark Scores <http://www.eembc.org>, 26.03.2015
- [12] K. Freund, **Redefining Datacenter Efficiency**, Calxeda, 2012
- [13] Armservers <http://armservers.com/2012/06/18/apache-benchmarks-for-calxeda-5-watt-web-server>, 01.01.2015
- [14] specPower http://www.spec.org/power_ssj2008/results/res2011q3/power_ssj2008-20110806-00392.html, 07.01.2015
- [15] Accenture, **Hadoop Comparison Study CloudBased Deployments**, <http://www.accenture.com/sitecollectiondocuments/pdf/accenture-hadoop-deployment-comparison-study.pdf>, 20.20.2014
- [16] Accenture Hadoop on Cloud, <http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Cloud-Based-Hadoop-Deployments-Benefits-and-Considerations.pdf>, 19.04.2015
- [17] Parallella <https://www.parallella.org/2013/08/21/parallella-hardware-update>, 16.02.2015
- [18] Parallella Soft, http://elinux.org/Parallella_Software, 07.01.2015
- [19] Parallella-hw, <http://github.com/parallella/parallella-hw>, 19.01.2015
- [20] Adapteva Referans Tasarım, <http://www.adapteva.com/white-papers/parallella-platform-reference-design>, 20.01.2015
- [21] Linaro Open source for ARM-SOC <http://www.linaro.org>, 29.01.2015
- [22] High Speed Serial, <http://www.xilinx.com/products/technology/high-speed-serial.html>, 02.04.2015
- [23] Multi-Gigabit Transceiver, http://en.wikipedia.org/wiki/Multi-gigabit_transceiver, 20.01.2015
- [24] Apache Hadoop <https://hadoop.apache.org>, 07.01.2015
- [25] Apache Hadoop NextGen MapReduce (YARN), <https://hadoop.apache.org/docs/r2.7.0/hadoop-yarn/hadoop-yarn-site/YARN.html>
- [26] Hadoop terasort <http://hadoop.apache.org/docs/current/api/org/apache/hadoop/examples/terasort/package-summary.html>, 17.04.2015
- [27] General Technical Discuss, <https://forums.xilinx.com/t5/General-Technical-Discussion/bd-p/GenDis>, 07.01.2015

EK-1**Çalışma Çıktıları (Study Outputs)**

Çeşitli hesaplama senaryoları için farklı sanal makine yapılandırılmalı bulut üzerine çalışma sonuçları.
(Study Outputs of Calculations on different virtual machine configurations)

yarn MapReduce	2 sbc	3 sbc	4 sbc	4 sbc	4 sbc	4 sbc	referans
opsiyonları	2 node	3 node	4 node	6 node	7 node	8 node	2node Pc
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 overall	-	-	-	-	-	189.287	112.863
TestDFSIO -read -nrFiles 10 -fileSize 1	-	-	-	-	-	35390	6290
TestDFSIO -write overall	-	-	-	-	-	283.731	111.845
TestDFSIO -write -nrFiles 10 -fileSize 1	-	-	-	-	-	45140	8820

Kurulum İşlem Adımları (Operation Steps)**Linaro Kurulumu (Linaro Installation)**

Parallella Tek kart bilgisayar üzerinde linaro işletim sistemi olarak kullanılmıştır. 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.

LXC Kurulumu(LXCInstallation)

Her bir Parallella için iki ve ya üç adet lxc kurulumu başarılabilmektedir. Bellek yetersizliği nedeniyle ikiden daha fazlasını kurmak bulut için performansı düşürmektedir.

```
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.

Hadoop Kurulumu(Hadoop Installation)

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

1. HDFS okuma yazma hızı dfs.replication parameter ile değişkenlik göstermektedir. dfs.replication=2 olarak kullanılmıştır.
 2. Parallella üzerindeki /etc/hosts dosyası açılarak hostnames ve küme içindeki diğer düğümlerin ip'leri girilmiştir. Her bir node için aynı işlemler tekrar edilmiştir. (Tablo 5) de belirtilen düğüm rolleri ilgili ip taşıyan kartlara atanmıştır.
 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. 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.
 6. Kümenin namenode ayarlarında slave olan Parallella ayarları yazılmıştır. \$hadoop-master-slave.sh içinde slave hostnamesler belirtildi.Tüm bunları yapınca herhangi bir node üzerinden tüm kumeyi başlatıp sonlandırabilir hale gelmiştir.
 7. Namenode başlatmak için: \$ service hadoop-dfs start
Jobtracker başlatmak için: \$ service hadoop-mapred start
veya basit olarak management console da yer alan Start Datanode/Jobtracker butonu da kullanılabilir.
- Test görev otomasyon aracı ile otomatik olarak başlatıldı. job başlatıldığında aşağıdaki Hadoop test sonuçları hem standard output ve hemde HDFS kısmına yazılmıştır.

Tablo 5. Küme ip ve Rol Yapısı
(Cluster IP and Role Configuration)

IP	Hadoop içindeki LXC Düğüm rolleri
172.16.131.63	Namenode (master, slave)
172.16.131.64	SecondaryNameNode (slave)
172.16.131.65	DataNode(slave), DataNode(slave)
172.16.131.66	DataNode(slave), DataNode(slave)

Çıktı Örneği (Output Sample)

```

terasort 8 Node için çalıştırma
hduser@hadoop3:~$ yarn jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar terasort
/out_tera /out_tera_sort
15/04/01 13:57:40 INFO terasort.TeraSort: starting
15/04/01 13:57:42 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using
builtin-java classes where applicable
15/04/01 13:57:49 INFO input.FileInputFormat: Total input paths to process : 2
Spent 1766ms computing base-splits.
Spent 16ms computing TeraScheduler splits.
Computing input splits took 1789ms
Sampling 2 splits of 2
Making 1 from 100 sampled records
Computing parititions took 1380ms
Spent 3207ms computing partitions.
15/04/01 13:57:51 INFO client.RMProxy: Connecting to ResourceManager at hadoop3/10.0.33.102:8050
15/04/01 13:57:57 INFO mapreduce.JobSubmitter: number of splits:2
15/04/01 13:57:58 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1427894434451_0009
15/04/01 13:58:00 INFO impl.YarnClientImpl: Submitted application application_1427894434451_0009
15/04/01 13:58:01 INFO mapreduce.Job: The url to track the job:
http://hadoop3:8088/proxy/application_1427894434451_0009/
15/04/01 13:58:01 INFO mapreduce.Job: Running job: job_1427894434451_0009
15/04/01 13:58:41 INFO mapreduce.Job: Job job_1427894434451_0009 running in uber mode : false
15/04/01 13:58:41 INFO mapreduce.Job: map 0% reduce 0%
15/04/01 13:59:32 INFO mapreduce.Job: map 100% reduce 0%
15/04/01 13:59:57 INFO mapreduce.Job: map 100% reduce 100%
15/04/01 13:59:58 INFO mapreduce.Job: Job job_1427894434451_0009 completed successfully
15/04/01 13:59:59 INFO mapreduce.Job: Counters: 50
  File System Counters
    FILE: Number of bytes read=10406
    FILE: Number of bytes written=340350
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=10202
    HDFS: Number of bytes written=10000
    HDFS: Number of read operations=9
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
  Job Counters
    Launched map tasks=2

```

Launched reduce tasks=1
 Data-local map tasks=1
 Rack-local map tasks=1
 Total time spent by all maps in occupied slots (ms)=92783
 Total time spent by all reduces in occupied slots (ms)=20735
 Total time spent by all map tasks (ms)=92783
 Total time spent by all reduce tasks (ms)=20735
 Total vcore-seconds taken by all map tasks=92783
 Total vcore-seconds taken by all reduce tasks=20735
 Total megabyte-seconds taken by all map tasks=95009792
 Total megabyte-seconds taken by all reduce tasks=21232640

Map-Reduce Framework

Map input records=100
 Map output records=100
 Map output bytes=10200
 Map output materialized bytes=10412
 Input split bytes=202
 Combine input records=0
 Combine output records=0
 Reduce input groups=100
 Reduce shuffle bytes=10412
 Reduce input records=100
 Reduce output records=100
 Spilled Records=200
 Shuffled Maps =2
 Failed Shuffles=0
 Merged Map outputs=2
 GC time elapsed (ms)=3517
 CPU time spent (ms)=**7530**
 Physical memory (bytes) snapshot=375734272
 Virtual memory (bytes) snapshot=1086078976
 Total committed heap usage (bytes)=256647168

Shuffle Errors

BAD_ID=0
 CONNECTION=0
 IO_ERROR=0
 WRONG_LENGTH=0
 WRONG_MAP=0
 WRONG_REDUCE=0

File Input Format Counters

Bytes Read=10000

File Output Format Counters

Bytes Written=10000

15/04/01 13:59:59 INFO terasort.TeraSort: done

terasort Ref

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

15/03/31 14:35:46 INFO terasort.TeraSort: starting

15/03/31 14:36:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

15/03/31 14:36:17 INFO input.FileInputFormat: Total input paths to process : 2

Spent 99ms computing base-splits.

Spent 2ms computing TeraScheduler splits.

Computing input splits took 102ms

Sampling 2 splits of 2

Making 1 from 100 sampled records

Computing partitions took 72ms

Spent 176ms computing partitions.

15/03/31 14:36:17 INFO client.RMProxy: Connecting to ResourceManager at hadoop3/10.0.35.102:8050

15/03/31 14:36:18 INFO mapreduce.JobSubmitter: number of splits:2

15/03/31 14:36:18 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1427789546068_0005

15/03/31 14:36:18 INFO impl.YarnClientImpl: Submitted application application_1427789546068_0005
15/03/31 14:36:18 INFO mapreduce.Job: The url to track the job:
http://hadoop3:8088/proxy/application_1427789546068_0005/
15/03/31 14:36:18 INFO mapreduce.Job: Running job: job_1427789546068_0005
15/03/31 14:36:53 INFO mapreduce.Job: Job job_1427789546068_0005 running in uber mode : false
15/03/31 14:36:53 INFO mapreduce.Job: map 0% reduce 0%
15/03/31 14:37:27 INFO mapreduce.Job: map 50% reduce 0%
15/03/31 14:37:29 INFO mapreduce.Job: map 100% reduce 0%
15/03/31 14:38:03 INFO mapreduce.Job: map 100% reduce 100%
15/03/31 14:38:03 INFO mapreduce.Job: Job job_1427789546068_0005 completed successfully
15/03/31 14:38:03 INFO mapreduce.Job: Counters: 49

File System Counters

FILE: Number of bytes read=10406
FILE: Number of bytes written=340350
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
HDFS: Number of bytes read=10202
HDFS: Number of bytes written=10000
HDFS: Number of read operations=9
HDFS: Number of large read operations=0
HDFS: Number of write operations=2

Job Counters

Launched map tasks=2
Launched reduce tasks=1
Data-local map tasks=2
Total time spent by all maps in occupied slots (ms)=64639
Total time spent by all reduces in occupied slots (ms)=32219
Total time spent by all map tasks (ms)=64639
Total time spent by all reduce tasks (ms)=32219
Total vcore-seconds taken by all map tasks=64639
Total vcore-seconds taken by all reduce tasks=32219
Total megabyte-seconds taken by all map tasks=66190336
Total megabyte-seconds taken by all reduce tasks=32992256

Map-Reduce Framework

Map input records=100
Map output records=100
Map output bytes=10200
Map output materialized bytes=10412
Input split bytes=202
Combine input records=0
Combine output records=0
Reduce input groups=100
Reduce shuffle bytes=10412
Reduce input records=100
Reduce output records=100
Spilled Records=200
Shuffled Maps =2
Failed Shuffles=0
Merged Map outputs=2
GC time elapsed (ms)=50
CPU time spent (ms)=1100
Physical memory (bytes) snapshot=714772480
Virtual memory (bytes) snapshot=2177273856
Total committed heap usage (bytes)=603979776

Shuffle Errors

BAD_ID=0
CONNECTION=0
IO_ERROR=0

```
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=10000
File Output Format Counters
  Bytes Written=10000
15/03/31 14:38:03 INFO terasort.TeraSort: done
```