T.C.
NAMIK KEMAL ÜNİVERSİTESİ
BİLİMSEL ARAŞTIRMA PROJELERİ
KOORDİNASYON BİRİMİ (NKÜBAP)

BİLİMSEL ARAŞTIRMA PROJELERİ
SONUÇ RAPORU

NKUBAP.00.17.AR.15.08 nolu Proje

Yapay Zeka ve Veri Madenciliği Uygulamalarında Yüksek Başarımlı Hesaplama Yazılımlarının Kullanımı ve Örnek Tasarım ile Performans Analizleri

Yürütücüsü:
Yrd. Doç. Dr. Erkan ÖZHAN
2016

NKUBAP.00.17.AR.15.08 no'lu "Yapay Zeka ve Veri Madenciliği Uygulamalarında Yüksek Başarımlı Hesaplama Yazılımlarının Kullanımı ve Örnek Tasarım ile Performans Analizleri" adlı proje Namık Kemal Üniversitesi Bilimsel Araştırma Proje Birimi tarafından desteklenmiştir.

T.C.
Namık Kemal Üniversitesi
Bilimsel Araştırma Projeleri Birimi

Yapay Zeka ve Veri Madenciliği Uygulamalarında Yüksek Başarımlı Hesaplama Yazılımlarının Kullanımı ve Örnek Tasarım ile Performans Analizleri

(Proje No: NKUBAP.00.17.AR.15.08)

Proje Ekibi:

Yürütücüsü:
Yrd. Doç. Dr. Erkan ÖZHAN

TEKİRDAĞ-2016

ÖNSÖZ

Verilerin ve veri işleme tekniklerinin arttığı günümüzde önemli sorunlardan bir olan büyük verilerin işlenmesi sürecini gerçekleştirmek analistler için zor bir süreçtir. Bu süreçte yol gösterici olması adına yapılan bu bilimsel araştırma projesinde yüksek başarımlı hesaplama yöntemleri incelenmiş ve kurulan düzenekle avantaj sağlaması muhtemel yöntemler denenmiştir.

Bu projenin (NKUBAP.00.17.AR.15.08) gerçekleşmesinde desteklerinden dolayı Namık Kemal Üniversitesi Bilimsel Araştırmalar Birimine, bu araştırma projesini gerçekleştirmek için sık sık ayrı kaldığım çocuklarım İdil ve Eymen'e, anlayışını ve desteğini hep yanımda hissettiğim eşim Şeniz ÖZHAN'a sonsuz teşekkürlerimi sunarım.

Tablo Listesi

Denklemler Listesi

Şekiller Listesi

Özet

Verileri depolama ve işleme araçlarının sayısı oldukça artmıştır. Bu artış beraberinde ilk yıllarda depolama ortamlarının yetersizliğini gündeme getirse de yaklaşık olarak her yıl defalarca katlanan depolama kapasitesi sayesinde bu sorun büyük ölçüde giderilmiştir. Ancak bu defa da depolanan büyük verilerin analiz edilmesi ve faydalı bilginin ortaya çıkarılması aşamasında, verinin büyüklüğü ve dolayısı ile analizi gerçekleştiren mikroişlemci ve RAM bellek sorunları ortaya çıkmıştır. Özellikle internetin her gün devasa veriler ürettiği günümüzde bu sorun daha da belirgin bir hal almıştır. Yapay zeka ve veri madenciliği teknikleri büyük verilerden anlamlı ve faydalı bilgiler elde etmeyi amaçlayan birbiri ile derin ilişkili iki disiplindir. Büyük veriler üzerinde bu disiplinlerin ortaya koyduğu metodlar, araştırmacılar tarafından sıklıkla kullanılmaktadır. Yüksek başarımlı hesaplama ise araştırmacıların karşılaştığı donanım yetersizlikleri ve uzun analiz sürelerini ortadan kaldırmayı amaçlamış bir diğer disiplindir.

Bu projede yüksek başarımlı hesaplama yöntemlerinin yapay zeka ve veri madenciliği uygulamalarında kullanılabilirliği araştırılmıştır. Araştırmanın ilk bölümünde yapay zeka, veri madenciliği ve yüksek başarımlı hesaplama konularında bilgi verilmiş. İkinci bölümde araştırmada kullanılan yöntem ve gereçler anlatılmış, farklı işletim sistemleri üzerinde yapay zeka, veri madenciliği ve yüksek başarımlı hesaplama yazılımlarının kullanımı ve performansları örnek veri seti ile incelenmiş, son bölümde ise araştırmada elde edilen bulgular ve sonuçlar aktarılarak tartışmaya açılmıştır.

Anahtar Kelimeler: *Yapay zeka, Veri Madenciliği, Yüksek Başarımlı Hesaplama*

Abstract

The number of data storage and data processing devices has greatly increased. Although this increase has caused the insufficiency of storage media to be put on the agenda in the previous years, this problem has been tackled on a big scale thanks to the storage capacity multiplied many times. However, this gave rise to the problems of huge amount of data and thus the microprocessor and RAM memory which perform the analysis at the stage of analysing the huge amount of data and extracting useful knowledge. This problem has become more obvious especially at present day because of the massive amount of daily data produced by internet. Artificial intelligence and data mining are two deeply interrelated disciplines aiming to obtain meaningful and useful knowledge from huge amounts of data. The methods developed by these two disciplines on these huge amounts of data are frequently used by researchers. High performance computing is also another discipline which aims to provide solutions to insufficient hardware and long periods of analysis.

In this project, whether the methods of high performance computing can be used in artificial intelligence and data mining applications has been searched. In the first part of the study some information related to artificial intelligence, data mining and high performance computing has been given and in the second part the methods and equipment used in the study has been mentioned, and the use and the performance of artificial intelligence, data mining and high computing software on different operating systems have been examined using a sample data set. In the final part the data and results obtained have been presented and opened to debate.

Keywords: *Artificial Intelligence, Data Mining, High Performance Computing.*

1. GIRIŞ

Günümüzde veri depolama, işleme ve giriş-çıkış kaynaklarında büyük artış yaşanmaktadır. Bu artışın sonucu olarak ta karşımıza büyüklü küçüklü veri grupları çıkmaktadır. Bu çalışmada veri gruplarının analizinde temel sorun haline gelen performans kaybı ve donanım yetersizliğinin ortadan kaldırılması için geliştirilmiş yüksek başarımlı hesaplama yazılımları araştırılmıştır.

Büyük veri kümelerinin analiz edilerek işe yarar bilginin ortaya çıkarılması ve bu verileri kullanan otomatik veya yarı otomatik sistemlerin tasarlanması veri madenciliği ve yapay zeka bilim dallarının başlıca uğraşı içerisinde yer almaktadır. Ancak veri gruplarındaki devasa artış beraberinde bazı önemli sorunlarda getirmiştir. Bu sorunların başında ise veri analiz sürelerindeki uzunluk ve büyük donanım gereksinimleri gelmektedir. Büyük veri işlerken bellek sınırları zorlanmakta hızla tükenmektedir. Gerçek zamanlı uygulamalarda algoritmalar anlık gelen verileri daha hızlı işlemek zorunda kalırlar (Witten & Frank, Data Mining Practical Machine Learning Tools and Techniques, 2005). Gündelik işler için yeterince gelişkin olan günümüz teknolojisi bilgisayarları yoğun ve büyük verilerin analizi söz konusu olduğunda yetersiz kalabilmektedir (Akı & Uçar, 2010). Pek çok modern şirket bilgisayar temelli bilgi süreçleri ve alt yapılarına artık daha çok güveniyor. Temel bilgi teknolojileri gelişmeye ve giderek daha karmaşık hale gelmeye başladıkça, verileri analiz etmek, yorumlamak ve keşfetmek daha da zor bir hale geliyor (Werner, 2008). Yüksek başarımlı hesaplama, karmaşıklığı yüksek problemlerin çözümü için disiplinler arası hesaplamalardan, genetik verilerin işlenmesine kadar, hatta yer bilimleri, uzay araştırmaları gibi konuları da kapsayarak çok çeşitli uygulama alanına hitap etmektedir (Aygün & Akçay, 2015). 1970 lerin başlarında üretilen tek-çip genel amaçlı mikroişlemciler, standart iş istasyonları ve PC kümelerinde 1980 lerin sonuna kadar yeterince olgunluğa erişememişti. Teorik olarak maksimum performans açısından rekabet 1990 lı yıllarından sonunda ortaya çıktı (Hager & Wellein, 2011). 1975 ve 1995 yılları arasında bilimsel yüksek performanslı bilgisayarlar çeşitli şirketler tarafından pazara sunulmuştu. Özellikle 2000'li yılların başlarından itibaren çok çekirdekli işlemciler günlük kullanımda daha sıklıkla karşılaşılmaya başlanmış ve işlemlerin eş zamanlı olarak yapılması mümkün olmuştur. Önemli bilimsel problemler olağanüstü miktarda hesaplama zamanı gerektirir. Bu sorunu ele almak amacıyla büyük işlem hacimli bilgisayarlar geliştirilmiştir (Flynn, 1966). Bunu sağlamak üzere paralel hesaplama teknikleri geliştirilmiştir. Görüntü işlemede paralel hesaplamayı üç ana başlıkta toplayabiliriz. i) Veri paralel, ii) Görev paralel, iii) Pipeline paralel (Prajapati & Vij , 2011). Pipeline yaklaşımı içeren sistemlerin temel olduğu bu yeni nesil teknolojiler, çok çekirdekli bir merkezi işlem birimi veya eş zamanlı çalışan uzaktan buluta erişimli hesaplama makinaları olabilmektedir (Aygün & Akçay, 2015).

Günümüzde oldukça artan veri kümeleri karşısında ortaya çıkan en büyük sorunlardan biride yüksek performans sağlayan analiz ortamlarının dizaynıdır. Bu analiz ortamları genellikle işletim sistemleri üzerine inşa edilmekte ve dolaylı olarak işletim sisteminin yapısı analiz sürelerini de etkilemektedir. Bu çalışmada ayrıca Microsoft Windows ve Linux işletim sistemleri üzerinde makine öğrenmesi algoritmalarının bazıları Weka yazılımı kullanılarak, hem tekli hem de küme hesaplama tekniği ile test edilmiştir.

Veri yoğunluğunun oldukça arttığı günümüzde bu verileri analiz etmekte kullandığımız birçok yöntem vardır. Analiz yöntemleri genellikle bir veya daha fazla algoritma kullanarak bilgisayar yazılımları ile yapılmaktadır. Algoritma, bazı verileri

veya veri setini iyi tanımlanmış hesaplama adımlarından geçirerek değer veya değerler kümesini halinde çıktı olarak veren hesaplamalı adımlar dizisidir (Cormen, Leiserson, & Stein, 2009). Klasik yöntemler analiz edilmesi çok zordur. Çünkü veriler içerisinde gözle görülmeyen ilk bakışta fark edilemeyecek karmaşıklıkta ilişkiler olabilir. Bu ilişkileri ortaya çıkarmak için istatistik, veri madenciliği, yapay zekanın dallarından olan makine öğrenmesi, yapay sinir ağları, uzman sistemler gibi disiplinlerin yol göstericiliğinden faydalanılır. Bu çalışmada makine öğrenmesi algoritmalarını içerisinde barındıran ve platform bağımsız çalışabilen Weka (Waikato Environment for Knowledge Analysis) yazılımı kullanılmıştır. İşletim sistemi olarak Windows 2012 Server HPC Pack ve Linux Fedora 24 dağıtımı kullanılmıştır. Kümeleme teknikleri ise büyük verilerin analizinde kullanılan birçok tekniği bünyesinde barındırır.

Farklı işletim sistemleri üzerinde performans analizi yapılan değerli çalışmalar literür de mevcuttur. Tanaka ve arkadaşları (Tanaka, Uehara, & Mori, 2008), 2008 yılında yaptıkları çalışmada Windows işletim sistemine, Linux'un Fedora 7 dağıtımını sanal makine olarak kurmuş ve grid hesaplama performans testleri yapmışlardır. Testlerde Windows üzerine sanal makine ile kurulmuş olan Linux grid hesaplama performansının Windows grid performansından daha iyi olduğunu görmüşlerdir. Bunun nedeni olarak da Linux çekirdek modu direktiflerinin Windows'tan daha az olmasını göstermişlerdir.

Martin ve arkadaşı (Borriss & Dannowski, 1998) ve arkadaşları, yaptıkları araştırmada ATM cihazlarının TCP üzerinden veri gönderme hızlarını Linux ve Windows işletim sistemleri kullanarak 3 farklı bilgisayar donanımı üzerinde analiz etmişler ve çalışmalarında elde ettikleri sonuçları yayınlamışlardır.

Ristov ve arkadaşı, paralel hesaplama performansını Linux platformları ile Windows Azure Cloud arasında araştırmışlardır. DMMM algoritması kullarak gerçekleştirdikleri testlerde n boyutlu matris hesaplaması yapmışlar ve matris boyutuna göre performansı ölçümlemişlerdir. Küçük boyutlu matrislerde Windows'un büyük boyutlu matrislerde ise Linux'un maliyet performans açısından avantajlı olduğunu çalışmalarında belirtmişlerdir. Cache bellek yoğunlaşmalarının performansı etkilediğini görmüşlerdir (Ristov & Gusev, 2013).

Lancester ve arkadaşı (Lancaster & Takeda, 1999), yaptıkları araştırmada Alpha işlemciler kullanan grid hesaplama sistemindeki MPI (Message Passing Interface- Mesaj geçiş arayüzü) arayüzünü ve compiler ortamını Windows ve Linux işletim sistemleri kullanarak test etmişlerdir. Complier ortamında Windows'un, MPI arayüzünde ise Linux'un avantajlı olduğunu görmüşlerdir.

Yazarlar (Panda & Nag, 2015), yaptıkları araştırmada java tabanlı bir API yardımıyla 3 farklı kriptolama algoritmasını Linux ve Windows işletim sistemleri üzerinde test etmişler bu algoritmalardan ikisinin Windows ve Linux'te diğerine göre şifreleme ve çözme işleminde işlem süresi olarak daha iyi olduğunu, ancak bellek kullanımlarının daha fazla olduğunu tespit etmişleridir. Ayrıca şifreleme ve çözme sürecinin ise Linux üzerinde daha avantajlı olduğunu belirtmişlerdir.

2. YÜKSEK BAŞARIMLI HESAPLAMA

Yüksek başarımlı hesaplama, hesaplamalı işlemlerde zamanı kısaltan verimi artıran yazılım ve donanım içerikli faaliyetler bütünü olarak tanımlanabilir.

Günümüzde işlem hızını ve kapasitesini artırmaya yönelik olarak birçok yöntem ortaya çıkıştır. Bunlar arasında;

- Bellek Paylaşımlı Paralel Hesaplama
- Dağıtık Paralel Hesaplama
- Grid
- GPU (Grafik işlem birimi)kullanarak hesaplama

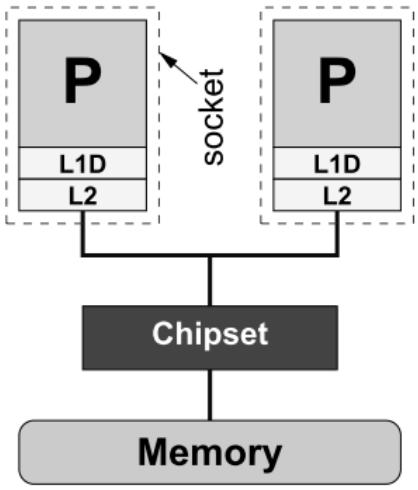
Paralel Hesaplamalı sistemleri paylaşımlı ve dağıtık olmak üzere iki sınıfa ayırabiliriz. Hesaplama yöntemleri önemli bir yere sahiptir. Bu yöntemlerin hepsine bulut bilişim adı verilmektedir. Gelişim açısından bulutlar (platform, altyapı, yazılım) bilgi teknolojilerinin gelişimindeki üçüncü dalga olarak görülmektedir (Udoh, 2011). Paralel programlar, yoğun hesaplama sorunlarını çözmek için özel olarak tasarlanmıştır (Lea, 1999). Görüntü işleme, örüntü tanıma, gibi birçok veri madenciliği ve yapay zeka uygulamalarında zamanı kısaltarak daha çok algoritma testi yapmaya imkan vermekte ve dolayısıyla performansı artırıcı bir rol oynamaktadır.

Yöntemi ne olursa olsun sonuçta hız ve dolayısı ile zamandan tasarruf ana önceliktir. Ancak unutulmaması gereken bir diğer noktada veri madenciliği araştırmalarında bir biri ile ilişkili verileri bulmak üzere yapılan analizlerde büyük veriler karşımıza çıkmaktadır. Büyük veriler ise donanım yetersizlikleri ile zaman kaybını doğurabilmektedir.

2.1 Bellek Paylaşımlı Paralel Hesaplama

Bir problemin çözümü için çok sayıda hesaplama elemanı (core-çekirdek) bir araya getirildiğinde paralel hesaplamadan söz ediyoruz demektir. Tüm modern süper bilgisayar mimarileri ağırlıkla paralel bağlıdır ve büyük ölçekli süper bilgisayar işlemci sayısı giderek artmaktadır (Hager & Wellein, 2011).

Bellek paylaşımlı paralel bilgisayarlar, paylaşılan fiziksel adres alanında ortak çalışan mikroişlemciler sistemidir (Hager & Wellein, 2011).



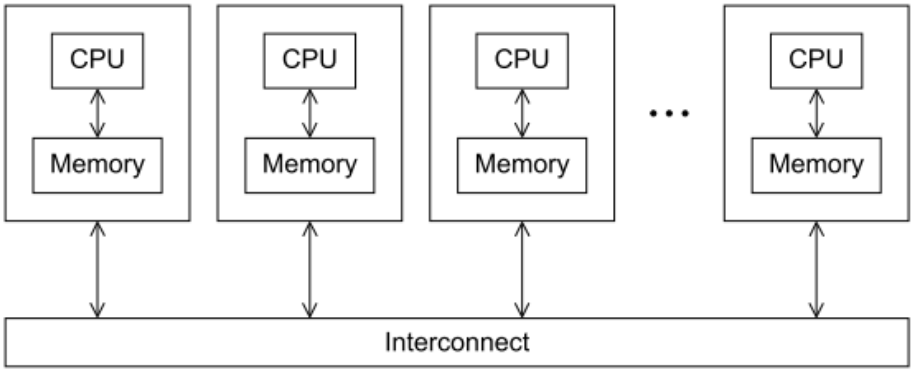
Şekil 1. Frontside bus(FSB) Tek düze bellek paylaşımlı iki tek çekirdekli işlemci (Hager & Wellein, 2011).

Paylaşılan bellekli sistemlerde, şekil 1.'de görüldüğü gibi her işlemci veya her çekirdek doğrudan belleğe erişebilir. Dağıtılmış bellek sistemlerinde ise her işlemcinin kendine özgü belleği vardır. Bazı sistemlerde ise nispeten küçük özel bellek paylaşılan bellek olmak üzere hibrid bir yapı kullanılır (Pacheco, 2011). Bellek paylaşımı tabanlı sistemlerde aynı olan çok sayıda işlemci uyumlu saat frekansı ile işleme başlar ve aynı belleği paylaşırlar (Çelik & Özmen, 2009).

İşlemci üzerinde paralellik, bir görevin bölümlerinin işlemci grubu arasında eşit dağıtılması yoluyla gerçekleştirilir. Öncelikle belirlenen kurala ve dağıtılmış veriye göre her işlemci kendi hesaplamasını yapar ve sonucu önceden belirlenmiş olan işlemciye gönderir. Bu süreç yazılım yoluyla gerçekleştirildiğinden esnek ve ekonomik bir yöntemdir (Çelik & Özmen, 2009).

2.2 Dağıtık Paralel Hesaplama

Bir programcı bakış açısından dağıtılmış bellekli bir sistem, şekil 2.'de görüldüğü gibi çekirdek koleksiyonunun bellek çiftleri halinde bir ağa dağıtılmasıdır.



Şekil 2. Dağıtılmış bellek sistemi (Pacheco, 2011)

1990 ların başlarında dağıtık hesaplama da ise temel sorun parçalar arasındaki iletişimin kurulmasıydı. Bu sorunun çözümü için geliştirilen mesaj geçiş arayüzü MPI (Message Passing Interface) günümüzde ticari veya ücretsiz uygulamalarda kullanılan standart haline geldi (Hager & Wellein, 2011). Günümüzde MPI dağıtık sistemler arasında standart iletişim yöntemi haline gelmiştir. Her işlemci, yerel belleğine doğrudan, diğer belleklere ise oradaki işlemcilere ileti yollayarak erişir (Eraslan, 2007). Doğrudan veya dolaylı fiziksel ara bağlantı ağları ile işlemciler arasında mesaj gönderilip alınabilir. Eğer iki işlemci doğrudan birbirine bağlı değilse ara düğümler yardımıyla (MPI) bu bağlantı gerçekleştirilir. MPI, işlemciler arası iletişim, veri koordinasyonu ve senkronizasyonu gibi görevleri ara bağlantıyı kullanarak gerçekleştirir (Rauber & Rünger, 2010). Bu yöntemin avantajı işlemci sayısı ile doğru orantılı olarak belleğinde artmasıdır. Bellek erişimi ağ ortamına girilmediği için daha hızlı olur. İşlemciler arasında iletişim sorunları yaşanması, yük dağılımının programcı tarafından dengelenmesi bu hesaplama yönteminde dezavantaj oluşturan bir durumdur.

2.3 Grid Hesaplama

Distributed computing olarak adlandırılan dağıtık bilgi işleme yönteminin sanallaştırılmasını sağlayan çözüm mimarisine kısaca Grid Hesaplama denilmektedir.

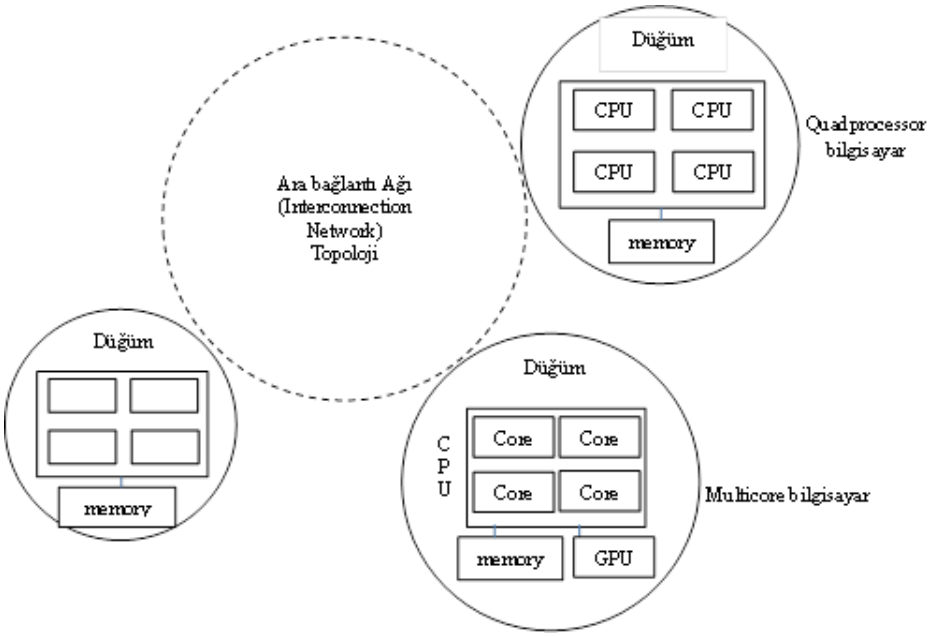
Grid hesaplamadaki temel amaç dağıtık bilgi işleme ve veri kaynaklarının kullanmakta olduğu işlemci güçleri, ağ kapasiteleri ve depolama kapasiteleri ile tek büyük bir sistem yaratmaktır ve oluşturulan bu sistem tamamen birbirinden bağımsız çalışmakta olan ve birbirine benzemeyen sistemlerin bir araya gelerek oluşturduğu sanal bir işleme gücüdür (IBM).

Küçük tesislerin elektrik ihtiyaçlarını tüm hat üzerinden karşılamaları grid yapısı için bir örnek olabilir. Grid yapısı yüksek güç sağlayan büyük ölçekli tedarikçiler olarak görülebilir (Resch & Gabriel, 2011).

2.4 Dağıtık Hesaplama

Küme hesaplamanın temel felsefesi, şekil 3'de gösterildiği gibi ayrık durumdaki CPU ve RAM gibi hesaplamayı gerçekleştiren temel bilgisayar kaynaklarının düğümler halinde bir görev için organize edilmesidir. Dağıtık hesaplama ise birden çok bilgisayarda toplu olarak ortak bir amacı gerçekleştirmek için CPU birlikteliği kuran yöntemdir (Vasoya & Koli, 2016). Dağıtık ve küme hesaplama, büyük veri kümelerini basit programlama modelleri kullanarak bilgisayar kümleri üzerinde işlemektir (Yang, Zhang, Hu, & Lin, 2015). Dağıtık hesaplamanın değişik ve zorlu görevleri vardır (Sinha, Saini, & Srikanth, 2014). Bunlar arasında, iş parçalarının tanımlanması, bu parçaların eş zamanlı ve paralel olarak çoklu işlemcilere eşlenmesi, giriş çıkışları dağıtma ve programla ilişkilendirme, çoklu işlemciler ile veri paylaşımının yönetimi ve senkronize edilmesi sayılabilir (Vasoya & Koli, 2016).

Küçük ve orta ölçekli bir ağ mimarisi için, bir veya daha fazla switch kullanılarak cluster düğümleri birbirine bağlanabilir. Bu bağlantıya ek te yapılabilir ve alt yapı olarak 1 veya 10 GBps'lik kabin bağlantıları kurulabilir (Guo, 2013). Bu çalışmada kümeleme yöntemi kullanılmıştır.



Şekil 3. Kümeleme Mimarisi (Nielsen, 2016)

Ağa bağlanan her bilgisayar gerekli konfigürasyonlar yapıldıktan sonrakümenin bir parçası olur. Küme içerisinde genellikle donanım ve iletişim açısından en avantajlı

olan bilgisayar ana düğüm olarak belirlenir. Diğer bilgisayarlar ise hesaplama düğümü olarak mimaride yerini alır. Ana düğüm noktasında literatürde iş işleyicisi (JobTracker) ve iş yöneticisi (JobManager) olmak üzere genellikle iki bileşen vardır. Bu bileşenler düğümler arasındaki iş dağılımını ve yönetimini sağlar. Kümeye giren her bilgisayar aynı özelliklere sahip olmayabilir. Ortak ağ protokolünü desteklemeleri yeterlidir ki günümüzde çoğu sistem bunu destekler.

2.5 Başarım Ölçütleri

Yüksek başarımlı hesaplama yapan sistemlerin başarımını ölçerken İşlem hızlanması (speedup) ve Verimlilik(efficiency) olmak üzere iki ölçüt kullanılır (Wilkinson & Allen, 2005). İşlem hızlanması hesaplanırken (1)'de gösterildiği gibi;

$$S(p) = \frac{Tek\ işlemcili\ sistemin\ yürütme\ zamanı}{p\ adet\ çoklu\ işlemcinin\ yürütme\ zamanı} \quad (1.1)$$

Hesaplama yapılır. Burada S(p) hızlanma faktörü olarak isimlendirilir. Tek mikroişlemcili sistemin yürütme zamanı t_s, aynı işlemin sonuçlandığı çok mikroişlemcili sistemin yürütme zamanı t_p olmak üzere, (2)'deki denklemi elde edebiliriz (Wilkinson & Allen, 2005).

$$S(p) = \frac{t_s}{t_p} \quad (1.2)$$

Verimliliği ölçerken ise 3'teki denklemde gösterildiği gibi bir hesaplama yapılabilir.

$$E = \frac{Tek\ işlemci\ kullanımında\ yürütme\ zamanı}{Çoklu\ işlemci\ kull.\ yürütme\ zamanı\ X\ işlemci\ sayısı} \quad (1.3)$$

Bu denkemi;

$$E = \frac{t_s}{t_p\, x\, p} = \frac{S(p)}{p} x100 \quad (1.4)$$

şeklinde ifade edebiliriz. Bu sayede verimi yüzdelik olarak ifade etmemiz de mümkün olabilir.

3. YAZILIM ÖRNEKLERİ

Günümüzde yüksek başarımlı hesaplamayı destekleyen yazılımların sayısı çok fazla olmamasına karşın giderek artan önemi ve ihtiyacın artması ile giderek sayısı artmaktadır. Bu yazılımları işletim sistemleri ve paket programlar olarak iki gruba ayırabiliriz.

3.1 İşletim Sistemleri

İşletim sisteminde yüksek başarımlı hesaplamanın verimini artırması için gerekli olan bileşenlere baktığımızda, özellikle MPI (Message Passing Interface) kullanımı ve cluster desteği bulunması son derece önemlidir. Yüksek başarımlı hesaplama için

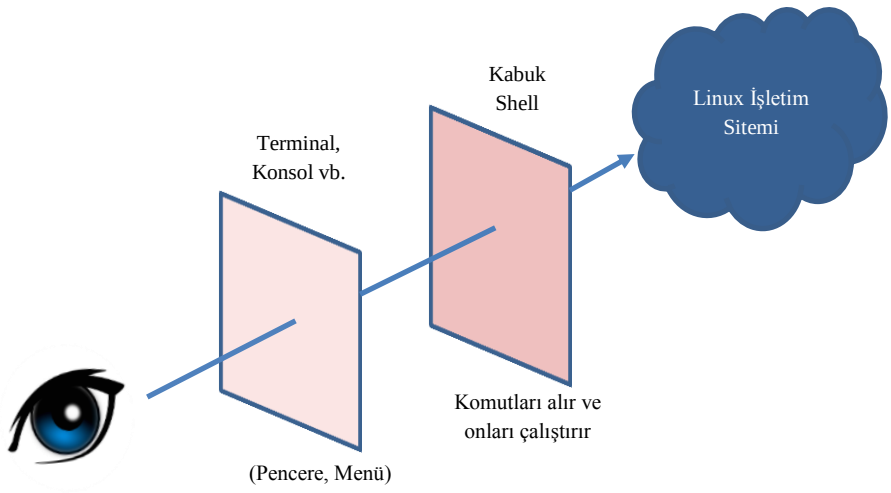
en yaygın kullanılan işletim sistemleri genellikle Linux çekirdeğine sahip olan Red Hat Fedora, SUSE, CentOS dağıtımlarıdır (Eadline, 2009).

Microsoft'un server ailesi yazılımlarına ise sonradan eklenen HPC Pack isimli eklenti ile cluster oluşturma, yönetme ve izleme gibi yüksek başarımlı hesaplama sistemlerinde olması gereken birçok özellik yüklenebilmektedir.

Microsoft HPC Pack teknolojisi Windows Server üzerine inşa edilmiş bir yüksek performanslı hesaplama (HPC) çözümdür. Windows HPC çözümü, Windows HPC küme ortamı için dağıtım, yönetim, iş planlaması ve izleme araçları kapsamlı bir dizi ve HPC uygulamaları geliştirmek ve çalıştırmak için esnek bir platformu bir araya getirir (Microsoft;, 2014).

Windows ve Linux sık kullanılan işletim sistemlerindendir. Kullanıcılarına benzer işlemler sunsalar da çalışma mekanizması olarak Linux ile Windows arasında çok fark vardır. İşletim sistemi üzerinde çalışan uygulamalar sistem kaynaklarına erişmek ister. Buna sistem çağrısı denir. Windows ve Linux te sistem çağrıları davranış olarak aynı şekilde görünseler de bunun için farklı bileşen ve fonksiyon tanımları kullanmışlardır. Windows sistem çağrılarını dll mekanizması üzerinden API(Application Programming Interface-Uygulama Programlama Arayüzü) ler ile alır (Li, Yang, & Ma, 2012). Linux ve Windows arasındaki en belirgin fark çekirdek yapıları ve dosya sistemlerinde kendini gösterir (Skendzic, Kovacic, & Jugo, 2011).

Windows işletim sisteminin merkezi uygulama tabanlı program bölümlerinden oluşur. Çekirdek alanında, çalışma alanı, çekirdek, aygıt sürücüleri ve donanım katmanı prosesleri yürütülür. Çekirdek bölümündeki programlar sistem verilerine ve donanımlara erişebilirler. Geliştirilen yazılımlar kullanıcı modunda çalışırlar ve sistem verileri ile donanımlara erişimleri sınırlıdır. Windows yüksek modüler mimariye sahiptir. Her sistem fonksiyonu işletim sisteminin bir bileşenini kontrol eder. Tüm yazılımlar, sistem fonksiyonlara erişmek için standart arayüz olan API'yi kullanırlar (Stallings, 2012).



Şekil 4. Linux'te Terminal ve Kabuk (Barrett, 2016)

Linux işletim sistemi açık kaynak kodlu UNIX tabanlı bir işletim sistemidir. Diğer popüler işletim sistemleri gibi simgeler, pencereler ve fare kontrolü ile grafik tabanlı kullanıcı ara yüzüne sahiptir. Linux'ün gerçek gücü, komutların yazılıp çalıştırıldığı komut satırı arayüzü(terminal) ve Shell(kabuk) çağrılarıdır. Şekil 1'de gösterildiği gibi terminal ve kabuk birbirinden farklıdır. Terminal görsel ara yüzden kabuk ise komutları alma ve çalıştırma ile görevlidir. Birçok Linux komutu girdiyi alıp ve çıktıyı üretir. Linux dosya sistemi parçalıdır. Root dizini en üst seviyedir ve diğer dizin ve dosyalar kökten aşağı doğru yayılırlar (Barrett, 2016).

İşletim sistemi, dosyalama, depolama, ağ bağlantısı ve diğer ihtiyaçları karşılar. Çoğu kullanıcı nadiren çekirdeğe dikkat eder (Barrett, 2016).Çekirdek, işletim sisteminin merkezinde bulunan bir bileşendir. Temel olarak donanım ve uygulama yazılımlarını birbirine bağlar. Bir çekirdek oluşturulurken birbirine bağlı iki ana mimari vardır: mikroçekirdek(çok küçük parçalar halinde) ve monolitik çekirdek (büyük benzersiz tek parça). Linux'ün birçok dağıtımı vardır ancak temelde çekirdekleri benzer yapıdadır (Castro, 2016).

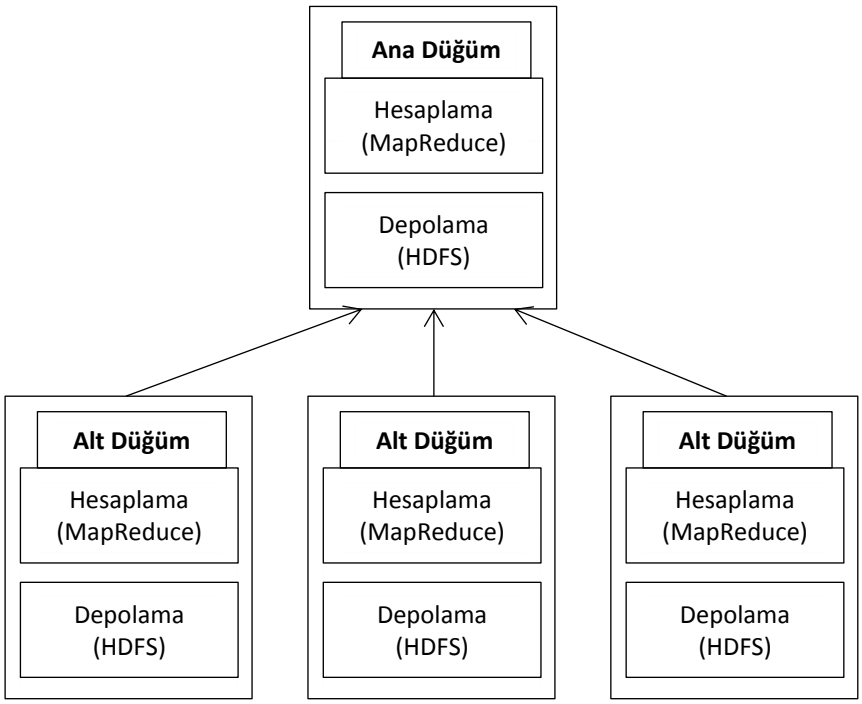
3.2 Geliştirme Ortamları ve Paket Programlar

Yapay zeka ve veri madenciliği uygulamalarına baktığımızda önemli ölçüde bilgisayarların çeşitli algoritmalar için hesaplama amacıyla kullanıldığını görürüz. Bu nedenle hesaplama yazılımlarının birçoğu önemli bir sorun haline gelmiş olan hesaplama süresi ve sistem gereksinimlerini azaltmak için bünyelerinde yüksek başarımlın hesaplama yöntemlerini gerçekleştirmeyi sağlayan eklentiler ve ara yüzler geliştirmişlerdir. Bu ise paralel çalışmayı destekleyen programlar yazmaya imkan veren yazılım geliştirme ortamları sayesinde mümkün olmuştur.

OpenMp Archtecture Review Board(OpenMP ARB), donanım ve yazılım üreticileri birliği tarafından kurulmuş bir birliktir. OpenMP, bellek paylaşımlı paralel programlamaya imkan veren, birçok işletim sistemi mimarisinde çalışabilen (Solaris, IBM AIX , HP-UX, GNU/LINUX, MAC OS X), basit ve esnek bir arayüz sağlayan, taşınabilir, ölçeklenebilir, çok çekirdekli işlemcilerle uyumlu, API desteği olan bir uygulama geliştirme arayüzüdür (OpenMP ARB, 2015).

Java ve. NET programlama dilleri paralel programlamaya imkan vermektedir. Bu amaçla paralel programlamayı desteklemek için Paralel FX kütüphanesi (PFX) ve Task Parallel Library (TPL)'i geliştirmişlerdir. Bu sayede paralel iş parçacıkları geliştirmek ve programlamak mümkün olabilmektedir (Sen, Ranjan;Microsoft Corporation, 2008).

Veri madenciliğinde büyük veri başlı başlına bir uğraşı alanıdır. Büyük verilerin depolanması, hızlı ve verimli bir şekilde analiz edilmesi günümüzde önemli bir bariyerdir. Bu bağlamda Linux dağıtımları üzerinde kullanılabilen Apache Hadoop geliştirilmiştir. Hadoop, birbirine paralel çalışan yüzlerce, binlerce hesaplama düğümü ile çok büyük veri setlerinin işlenmesi için geliştirilmiş yüksek ölçekli, açık kaynak kodlu depolama platformudur (IBM International Business Machines Corp.). Paralel veri işleme ve büyük veri türlerini işlemek için uygun bir platform sağlar. Hadoop'un HDFS (Hadoop Distributed File System) ve MapReduce/YARN olmak üzere iki ana bileşeni vardır. Şekil 5.'te görüldüğü gibi ana düğüm noktası üzerinde ve alt(ikincil) düğüm noktaları üzerinde de benzer yapı bulunur ve ana düğüm noktasına bağlıdırlar. İkincil düğüm noktalarına daha fazla düğüm eklemek mümkündür (Holmes, 2012).



Şekil 5. Yüksek seviye Hadoop Mimarisi (Holmes, 2012)

HDFS, Hadoop'un dağıtık dosya sistemi bileşenidir. Büyük dosyalar üzerine yazma ve okuma görevlerinde yüksek başarıma sahiptir. Veri kaybına karşı donanım hatalarını yazma hatalarını tolere edebilecek bir sisteme sahiptir (Holmes, 2012).

MapReduce, büyük veri setleri oluşturmak ve işlemek için programlama modelinin uygulama bileşenidir. Özel bir eşleşme fonksiyonu, ortak anahtar ve değer çiftlerini belirleyip oluşturarak bunları ortak anahtar/değer çifti şeklinde işler. Bir indirgeme fonksiyonu tüm ortak değerleri benzer ortak anahtarlar ile birleştirir (Dean & Ghemawat, 2004).

MapReduce programlama modeli birçok farklı amaç için Google tarafından başarıyla kullanılmıştır. Bu model paralel ve dağıtık sistemler konusunda uzman olmayan programcılar tarafından kullanımı kolaydır (Dean & Ghemawat, 2004).

Hadoop, içerisinde bulunan HDFS ve MapReduce bileşenlerinin üzerinde birçok aracıda ihtiva eder. Bunlar arasında;

- Yüksek Seviye Diller: Crunch, Cascading, Pig, Hive.
- Tahminleme Araçları: RHadoop, RHIPE, R, Mahout

sayılabilir. Bu araçlar yardımıyla istemciler Hadoop ile bağlantı kurarak tahminleme analizlerini gerçekleştirebilirler. Google, Facebook, Twitter, Yahoo!, eBay, Samsung, AOL ve daha birçok büyük şirket Hadoop kullanmaktadır (Holmes, 2012). Hadoop platformunun özellikle veri analizinde birçok analist tarafından tercih edilen R yazılımına uyumlu olması önemlidir.

MathWorks şirketi tarafından geliştirilmiş olan Matlab yazılımı paralel hesaplama teknikleri ve Hadoop kullanımına uygun araçlar sunmuştur. Masaüstü bilgisayarlar için MapReduce kullanımını da sunmaktadır (The MathWorks, Inc.). Yine Stata yazılımı da paralel hesaplama desteği mevcut bir diğer veri analiz programıdır.

Weka(Waikato Environment for Knowledge Analysis), Waikato Üniversitesinde geliştirilmiş, makine öğrenmesi alanında birçok algoritmayı ve aracı içerisinde

barındıran bir yazılımdır. Bu yazılıma Ekim 2011 tarihinde, Pentaho şirketi tarafından paralel hesaplamayı desteklemek üzere WekaServer adında bir eklenti yayınlamıştır (Pentaho A Hitachi Group Company). Bu eklenti kurulduktan sonra istenildiği kadar hesaplama bilgisayarı ana düğüm noktasına bağlanabilir. Ana düğüm noktası makine öğrenmesi algoritmalarından gelen iş görevlerini hesaplama düğümlerine göndererek daha hızlı ve yük dağılımlı olarak hesaplanmasını sağlayabilir. Hesaplama düğümlerinin durumu ise web tabanlı olarak anlık görülebilmektedir.

3.3 Weka Yazılımı

Bu çalışmada analiz ve deney yazılımı olarak Weka programı kullanılmıştır. Weka, Yeni Zellanda'daki Waikato Üniversitesi tarafından *Waikato Environment for Knowledge Analysis* adı ile geliştirilmiş yazılımdır. Sistem java programlama dili ile yazılmış ve GNU lisansı ile dağıtılmaktadır. Weka yazılımı birçok işletim sistemi altında çalışabilir. Platform bağımsız diyebiliriz. Tek bir ara yüzde çok sayıda öğrenme algoritmasını sağlar. Bunların yanında veri setleri üzerinde indirgenim gibi işlemleri yapmaya sağlayan düzenleme araçları da sağlar (Witten, Frank, & Hall, Data Mining Practical Machine Learning Tools and Techniques, 2011).

JDBC sürücüsü sayesinde veri tabanı gibi kaynaklara erişim mümkündür. Kümeleme, sınıflandırma, regresyon birliktelik kuralları yöntemlerini kullanarak analizler yapabilir (Bell, Machine Learning Hands-On for Developers and Technical Professionals, 2015). Weka yazılımı için son zamanlarda geliştirilen Paket Yöneticisi (Package Manager) ile birçok yeni geliştirilmiş veya ek özellikler katılmış algoritma yanında WekaServer gibi kümeleme gerçekleştirmeye imkan sağlayan eklentiler ilave etmek mümkündür. Bu çalışmada kümeleme için WekaServer eklentisi kullanılmıştır.

3.4 Yapay Zeka ve Veri Madenciliği

Makine öğrenmesi, bilgisayarların örnek veri veya geçmiş uzman deneyimlerini kullanarak, belirli ölçütlere göre başarımlarını artıracak biçimde programlanmasıdır (Alpaydın, 2013). Algoritması bilinmeyen durumlarda geçmiş deneyimler ve veriler kullanarak makine öğrenmesi yöntemleri girdinin çıktısını tahminleyebilir. Bunu geçekleştirmek için de öğrenme algoritmaları adı altında birçok algoritma geliştirilmiştir. Makine öğrenmesinin geçmişi 1950 yılına dayanır. 1950 yılında Alan Turing "Makineler düşünebilir mi?" sorusunu yöneltmiştir (Bell, Machine Learning Hands-On for Developers and Technical Professionals, 2015). Günümüzde bu soruya hala cevap aranmakla birlikte önemli mesafeler alınmıştır. Geleneksel istatistik, veri kümeleri arasındaki ilişkileri tespit etmek için tümdengelim yöntemini kullanır, makine öğrenmesi, yapay sinir ağları gibi yapay zeka öğrenme teknikleri ise veri kümeleri arasındaki zayıf desenleri bulmak için endüktif yöntemi izler (Nisbet, Elder, & Miner, 2009). Makine öğrenmesi işlem süreci şekil 6.'da gösterilen adımlardan oluşur.

Veri kolleksiyonu oluşturmak için günümüzde oldukça fazla sayıda ortam bulunmaktadır. Makine öğrenmesi sürecinin ilk iki aşaması veri madenciliği disiplininin ilgi alanına girer. Bu çalışma ise bu süreçte üçüncü basamakta gerçekleşen işlemlerle ilgilidir.

Ortaya çıkan veri kümelerinin analizinde her araştırmacı yazılım ve donanım temelli bir düzenek kurmaktadır. Analizleri bu düzenek üzerinde gerçekleştirirken

algoritmalardan faydalanır. Algoritma türlerinden biride makine öğrenmesi algoritmalarıdır. Günümüzde, yerel ağ veya kampüs türündeki ağlarda Windows ve Linux de dahil olmak üzere farklı işletim sistemlerinden oluşur (Uemura, Nakajima, & Sato, 2007).

Şekil 6. Makine öğrenmesi işlem süreci (Bell, Machine Learning Hands-On for Developers and Technical Professionals, 2015).

3.5 Weka Yazılımı

Weka, Yeni Zellanda'daki Waikato Üniversitesi tarafından *Waikato Environment for Knowledge Analysis* adı ile geliştirilmiş yazılımdır. Sistem java programlama dili ile yazılmış ve GNU lisansı ile dağıtılmaktadır. Weka yazılımı birçok işletim sistemi altında çalışabilir. Platform bağımsız diyebiliriz. Tek bir ara yüzde çok sayıda öğrenme algoritmasını sağlar. Bunların yanında veri setleri üzerinde indirgenim gibi işlemleri yapmaya sağlayan düzenleme araçları da sağlar (Witten, Frank, & Hall, Data Mining Practical Machine Learning Tools and Techniques, 2011).

JDBC sürücüsü sayesinde veri tabanı gibi kaynaklara erişim mümkündür. Kümeleme, sınıflandırma, regresyon birliktelik kuralları yöntemlerini kullanarak analizler yapabilir (Bell, Machine Learning Hands-On for Developers and Technical Professionals, 2015). Weka yazılımı için son zamanlarda geliştirilen Paket Yöneticisi (Package Manager) ile birçok yeni geliştirilmiş veya ek özellikler katılmış algoritma yanında WekaServer gibi kümeleme gerçekleştirmeye imkan sağlayan eklentiler ilave etmek mümkündür. Bu çalışmada kümeleme için WekaServer eklentisi kullanılmıştır.

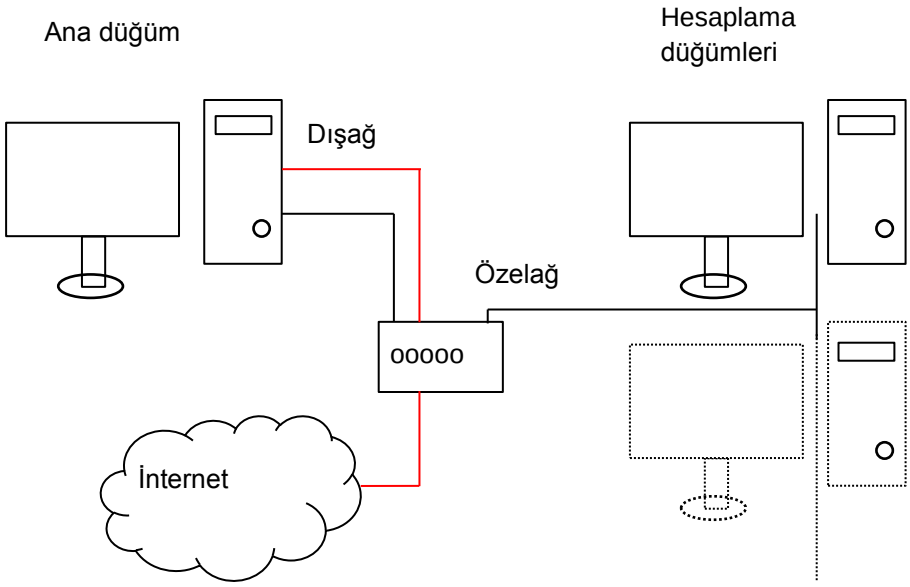
4. GEREÇ ve YÖNTEM

Çalışmada kullanılan yazılımlar ve donanım mimarisi tablo1.'de gösterilmiştir. İki farklı işletim sistemi ana düğüm ve hesaplama düğümlerine yüklenmiş ve istenilen işletim sistemine göre başlangıç ekranında seçim yapılarak çalıştırılmıştır.

Küme Elemanı	Özellikleri
Ana Düğüm (MasterNode)	– Intel i7 4790K 4.6 GHz 8 Core CPU – 10.00 GB DDR3 1333 MHz RAM – 240 GB SSD HardDisk – 10/100 Mbps Ethernet Kartı – 1 Gbps Ethernet Kartı – Windows Server 2012 HPC Pack işletim sistemi – Linux Fedora 24 Cloud Server 64 bit dağıtımı – Weka 3.9.0 ve WekaServer Eklentisi – MySQL veri tabanı – Java 64bit – 10/100 Mbps Ethernet Kartı
Hesaplama Düğümü (Slave Node)	– Intel i5 4590 3.30 GHz 4 core CPU – 4.0 GB 1600 MHz RAM – 7200 rpm 1 TB HardDisk – Windows Server 2012 HPC Pack işletim sistemi – Linux Fedora 24 Work Station 64 bit dağıtımı – Weka 3.9.0 ve WekaServer Eklentisi – Java 64 bit –
Ara Bağlantı Elemanları ve Switch	– 8 port 10/100 Mbps switch – Cat6 kablo

Tablo 1. Test elemanları ve özellikleri

Tablo 1'de listelenen yazılım ve donanım elemanları şekil 7'de gösterildiği gibi kurulmuştur. Ana düğüm üzerine çift ethernet kartı bulunmaktadır. Bu sayede ana düğüm noktasında hesaplama düğümleri ve dış ağ arasındaki bağlantı aynı anda sağlanmıştır. Şekil 7'te görüldüğü gibi özel ağda bulunan hesaplama düğümlerine özel bir ip aralığı ile ip verilmiştir. Bu sayede dış ağ bloğu ile farklı ip bloğuna sahip olmuşlar ve dış ağ ile aralarındaki tek bağlantı ana düğüm olmuştur. Ana düğüm hesaplama düğümleri için ağ geçidi görevini de yerine getirmektedir. Hesaplama düğümlerinin sayısı istenildiği kadar artırılabilir. Bu çalışmada asgari gereklilik olan 1 ana düğüm ve 1 hesaplama düğümü bilgisayarı kullanılmıştır. Ana düğüm ayrıca etki alanı yöneticisi olarak kullanılmıştır.



Şekil 7. Örnek model düzeneği-Küme yapısı ve mimarisi

Ana düğüm noktası önce Windows işletim sistemi ile başlatılmış ve Weka yazılımı bu işletim sistemi üzerinde çalıştırılmıştır.

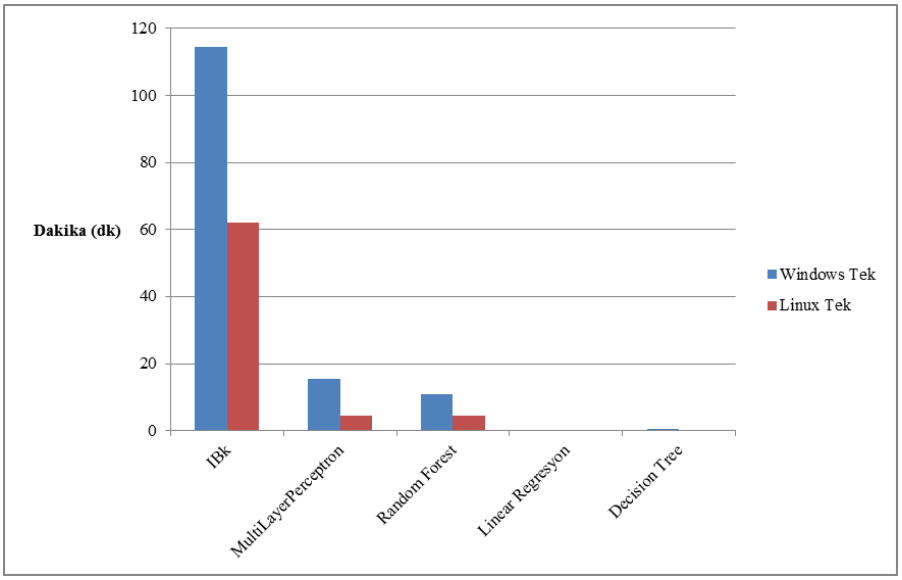
Test için kullanılan veri seti Kaliforniya Üniversitesi Irvine kampüsü, Makine Öğrenmesi ve Uzman Sistemler Merkezi veri havuzundan alınan Skin Segmentation (Dhall & Bhatt, 2016) adlı veri setidir. Bu veri seti, 3.2 MB boyutunda, 245057 satır ve 4 sütundan oluşmaktadır. Text halinde indirilen bu veri seti ana düğüm de bulunan MySQL veritabanına aktarılmıştır.

Ana düğüm noktası üzerinde Weka yazılımı MySQL veritabanına bağlanarak test verileri transfer edilmiştir. İlk olarak sadece ana düğümde Windows işletim sistemi çalıştırılarak, test verisi IBk(k-NN), MultiLayerPerceptron, Random Forest, Lineer Regresyon ve Decision Table makine öğrenmesi algoritmalarıyla ayrı ayrı işlenmiştir. Bu algoritmalar sık kullanılan makine öğrenmesi sınıflandırma algoritmalarıdır. Weka yazılımının bilgi ekranından her bir algoritmanın işleme başlama ve bitirme zamanları kaydedilmiştir. Aynı işlem ana düğümde Linux işletim sistemi çalıştırılarak tekrarlanmıştır.

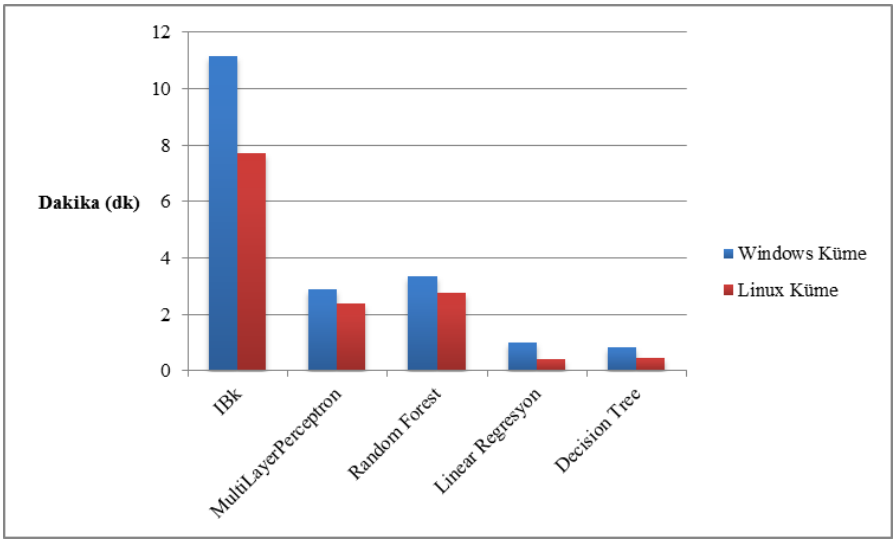
İkinci olarak algoritmalar ve test verisi, ana düğüm ve hesaplama düğümünde Windows işletim sistemi üzerinde WekaServer eklentisi yardımıyla küme hesaplama yöntemi ile analiz edilmiştir. Her bir algoritma için başlama ve bitiş zamanları yine kayıt altına alınmıştır. Son olarak ana düğüm ve hesaplama düğümü Linux işletim sistemi ile açılmış, WekaServer eklentisi kullanarak aynı işlemler tekrarlanmıştır.

5. BULGULAR ve TARTIŞMA/SONUÇ

Tek bilgisayarda iki farklı işletim sistemi ile gerçekleştirilen analizlerde elde edilen işlem süreleri dakikaya çevrilerek şekil 8'de gösterilmiştir. Lineer regresyon ve Decision table algoritmaları çok kısa sürmüştür. Özellikle işlem yükü bakımından yoğun hesaplama gerektiren IBk ve MultiLayer Perceptron algoritmalarının çalışma sürelerine bakıldığında süreç daha iyi anlaşılmaktadır.



Şekil 8. İşletim sistemleri tek bilgisayar üzerinde çalıştırıldıklarında işlem süreleri (dk).



Şekil 9. İşletim sistemleri küme bilgisayar halinde çalıştırıldıklarında işlem süreleri (dk).

Küme halinde yine iki farklı işletim sistemi ile homojen oluşturulan düzenekte gerçekleştirilen deneyler sonunda elde edilen işlem süreleri ise şekil 9'de gösterilmiştir. Burada tek olarak çalışan bilgisayarların işlem süreleri ile küme halinde çalışan bilgisayarların işlem sürelerini karşılaştırırsak küme hesaplamanın süreyi ne kadar kısalttığı net görülebilir. Daha ayrıntılı veriler tablo 2 ve tablo 3'te gösterilmiştir. Tablolar incelendiğinde işlem süresi tek bilgisayarda çok kısa süren algoritma çalışma zamanlarının küme bilgisayar yönteminde arttığı görülmektedir.

	Tek İşlem Zamanı (dk)	
Algoritmalar	**Windows**	**Linux**
IBk	114,62	62,00
MultiLayerPerceptron	15,47	4,40
Random Forest	10,82	4,42
Linear Regresyon	0,12	0,05
Decision Tree	0,68	0,18

Tablo 2. Tek bilgisayar işlem zamanları

	Küme İşlem Zamanı (dk)	
Algoritmalar	**Windows**	**Linux**
IBk	11,15	7,72
MultiLayerPerceptron	2,90	2,38
Random Forest	3,37	2,78
Linear Regresyon	1,00	0,42
Decision Tree	0,85	0,45

Tablo 3. Küme bilgisayar işlem zamanları

Veri ekosisteminin artık oldukça geniş bir alana sahip olduğunu söyleyebiliriz. Bu ekosistemde doğal olarak ortaya çıkan devasa veri kümelerinin analiz edilerek veriden bilginin ortaya çıkarılması oldukça önemli ve bir o kadar da zor bir süreç haline dönüşmüştür. Büyük miktardaki verilerin analiz edilmesinde yüksek başarımlı hesaplama tekniklerini ve yazılımlarını kullanmak günümüz şartlarında oldukça avantajlı bir durum olarak karşımıza çıkmaktadır. Bu çalışmada bahsedilen yöntem ve yazılımlar araştırmacıların büyük veri analizi problemini çözmelerine yardımcı olabilir. En çok kullanılan yazılımlar, araçlar ve yetenekleri bu çalışmada araştırmacıların bilgisine sunulmaya çalışılmıştır. Özellikle Hadoop platformu büyük ölçekli şirketler tarafından sık kullanılmaktadır. Hadoop platformunu kullanmak isteyen araştırmacılar, özellikle Linux işletim sistemi kullanımı konusunda daha önceden deneyimleri yoksa sorun yaşayabilirler. Bunun yanında gelecekteki çalışmalarda yüksek başarımlı hesaplamaya imkan veren yazılımların test verileriyle birebir karşılaştırılması ve performanslarının görülmesi faydalı olabilir.

Bu çalışmada elde edilen veriler analiz edildiğinde diğer bir ortaya çıkan sonuç küme hesaplama yönteminin tek bilgisayar yöntemine göre analiz sürelerini kısalttığıdır. Ancak tek bilgisayar üzerinde çok kısa süren analizlerin küme yöntemini gerektirmediği de görülmüştür. Çalışma da ortaya çıkan bir diğer sonuç ise önceki çalışmalarda sanal makine ortamında elde edilen sonuçların gerçek ortamda da görülmesi yani Linux işletim sisteminin küme hesaplama da işlem zamanı açısından daha avantajlı olduğudur. Windows ve Linux işletim sistemi küme hesaplama yöntemleri arasında çok büyük farklar olmasa da Linux'un çekirdek yapısından kaynaklana bir hız avantajının ve farkının olduğu da açıktır. Çalışmaya düzeneğin

kurulması açısından bakıldığında Linux işletim sistemi küme ayarlarının Windows işletim sistemine göre daha uzun sürdüğü görülmüştür.

İki işletim sistemi arasındaki küme hesaplama açısından yapılan performans araştırması farklı kümeleme yöntemini destekleyen yazılımlarla tekrarlanarak benzer sonuçları verip vermeyeceği araştırmacılar tarafından incelenebilir. Bunun yanında bu iki sistemin RAM bellek kullanımları da araştırılabilir.

6. TEŞEKKÜR

Bu çalışma, NKUBAP.00.17.AR.15.08 proje no ile kayıtlı bulunan, Namık Kemal Üniversitesi, Bilimsel Araştırma Projeleri Birimince desteklenmiştir. Bu nedenle Namık Kemal Üniversitesi, Bilimsel Araştırma Projeleri Birimi'ne teşekkür ederim.

Çalışmada kullanılan test verilerini aldığım Kaliforniya Üniversitesi Irvine kampüsü, Makine Öğrenmesi ve Uzman Sistemler Merkezi'ne teşekkür ederim.

7. PROJEDEN YAPILAN YAYINLAR

Sözlü Bildiri:

Özhan, E. (2016a). Farklı İşletim Sistemleri Üzerinde Kümeleme Yöntemi ile Makine Öğrenmesi Algoritmalarının Performans Testleri (Machine Learning Algorithms Performance Tests with Clustering Methods on Different Operating Systems.) In *International Conference on Computer Science and Engineering* (pp. 362–367). Tekirdağ, Turkey: (UBMK 2016).

Poster Bildiri:

Özhan, E. (2016b). Yapay Zeka ve Veri Madenciliği Uygulamalarında Yüksek Başarımlı Hesaplama Yazılımları (High Performance Computing Software in Artificial Intelligence and Data Mining Applications). In *International Conference on Computer Science and Engineering* (pp. 719–724). Tekirdağ, Turkey: (UBMK 2016).

8. KAYNAKLAR

Uemura, Y., Nakajima, Y., & Sato, M. (2007). Direct Execution of Linux Binary on Windows for Grid RPC Workers. *2007 IEEE International Parallel and Distributed Processing Symposium* (s. 1-8). Long Beach: IEEE.

Akı, M., & Uçar, E. (2010). Trakya Üniversitesi Yüksek Başarımlı Hesaplama Sistemi (TRAKYAHPC) Kurulumu ve Web Arayüzü Tasarımı Üzerine Bir Örnek Olay . *II. ULUSAL YÜKSEK BAŞARIMLI ve GRID HESAPLAMA KONFERANSI .* İstanbul.

Alpaydın, E. (2013). *Yapay Öğrenme.* İstanbul: MIT, BÜTEK.

Aygün, S., & Akçay, M. (2015). Yüz Tanıma Teknolojilerinde Yüksek Başarım için Paralel Hesaplama. *4. Ulusal Yüksek Başarımlı Hesaplama Konferansı.* Ankara.

Barrett, D. J. (2016). *Linux Pocket Guide.* Sebastopol: O'Reilly Media, Inc.

Bell, J. (2015). *Machine Learning Hands-On for Developers and Technical Professionals.* (C. Long, Dü.) Indianapolis, Canada: John Wiley & Sons, Inc.

Bell, J. (2015). *Machine Learning Hands-On for Developers and Technical Professionals.* Indianapolis, Indiana: John Wiley & Sons, Inc.

Borriss, M., & Dannowski, U. (1998). TCP Performance over ATM on Linux and Windows NT. *ICATM-98., 1998 1st IEEE International Conference on.* Colmar.

Castro, J. D. (2016). *Introducing Linux Distros.* New York : Springer Science+Business Media New York.

Cormen, T. H., Leiserson, C. E., & Stein, C. (2009). *Introduction to Algorithms.* Cambridge, Massachusetts - London, England: The MIT Press.

Çelik, A., & Özmen, A. (2009). Dağıtık Paralel Sistemler Hakkında Kıyaslamalı Bir Çalışma: PVM ve MPI. *5. Uluslararası İleri Teknolojiler Sempozyumu (IATS'09).* Karabük.

Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. *OSDI'04: Sixth Symposium on Operating System Design and Implementation.* San Francisco.

Dhall, A., & Bhatt, R. (2016, Temmuz). UC Irvine Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Skin+Segmentation adresinden alınmıştır

Eadline, D. (2009). *High Performance Computing For Dummies.* Hoboken, NJ: Wiley Publishing, Inc.

Eraslan, G. (2007, Mart 4). *Paralel Programlama ve MPI.* Temmuz 20, 2016 tarihinde https://seminer.linux.org.tr/wp-content/uploads/ParalelProgramlamaveMPI.pdf adresinden alındı

Flynn, M. (1966). Very high-speed computing systems. *Proceedings of the IEEE, 54*(12), 1901-1909.

Guo, S. (2013). *Hadoop Operations and Cluster Management Cookbook.* Birmingham, İngiltere: Packt Publishing Ltd.

Hager, G., & Wellein, G. (2011). *Introduction to High Performance Computing for Scientists and Engineers.* Boca Raton: CRC Press Taylor & Francis Group.

Holmes, A. (2012). *Hadoop in Practice.* New York: Manning Publications Co.

IBM. (tarih yok). *http://www-05.ibm.com/tr/solutions/edu/grid.html*. (IBM) Temmuz 20, 2016 tarihinde http://www-05.ibm.com/tr/solutions/edu/grid.html adresinden alındı

IBM International Business Machines Corp. . (tarih yok). *Hadoop-IBM*. (International Business Machines Corp.) Temmuz 22, 2016 tarihinde http://www.ibm.com/analytics/us/en/technology/hadoop/#hadoop-resources adresinden alındı

Lancaster, D., & Takeda, K. (1999). Comparative Performance of a Commodity Alpha Cluster running Linux and Windows NT. *Cluster Computing, 1999. Proceedings. 1st IEEE Computer Society International Workshop on.* Melbourne.

Lea, D. (1999). *Concurrent Programming in Java: Design Principles and Patterns,.* Boston: Addison Wesley.

Li, R., Yang, N., & Ma, S. (2012). An Approach of Windows Memory Management Simulation on Linux. *2012 Third World Congress on Software Engineering.* Wuhan.

Microsoft;. (2014, Kasım 18). *Microsoft Corp.* (Microsoft) Temmuz 21, 2016 tarihinde https://technet.microsoft.com/tr-tr/library/cc514029(v=ws.11).aspx adresinden alındı

Nielsen, F. (2016). *Introduction to HPC with MPI for Data Science.* (I. Mackie, Dü.) Switzerland, Switzerland: Springer International Publishing.

Nisbet, R., Elder, J., & Miner, G. (2009). *Handbook of Statistical Analysis and Data Mining Applications.* Burlington, MA: Elsevier Inc.

OpenMP ARB. (2015, Temmuz 14). *The OpenMP® API specification for parallel programming*. (The OpenMP® API specification for parallel programming) Temmuz 21, 2016 tarihinde http://openmp.org/wp/about-openmp/ adresinden alındı

Pacheco, P. (2011). *An Introduction to Parallel Programming.* Burlington: Morgan Kaufmann Publishers is an imprint of Elsevier.

Panda, M., & Nag, A. (2015). Plain Text Encryption Using AES, DES and SALSA20 by Java Based Bouncy Castle API on Windows and Linux. *Second International Conference on Advances in Computing and Communication Engineering.* Dehradun.

Pentaho A Hitachi Group Company. (tarih yok). *Weka Server*. (Pentaho A Hitachi Group Company) Temmuz 24, 2016 tarihinde http://wiki.pentaho.com/display/DATAMINING/Weka+Server adresinden alındı

Prajapati , H., & Vij , S. (2011). Analytical Study of Parallel and Distributed Image Processing. *International Conference on Image Information Processing (ICIIP 2011).* Himachal Pradesh.

Rauber, T., & Rünger, G. (2010). *Parallel Programming for Multicore and Cluster Systems.* New York: Springer-Verlag Berlin Heidelberg.

Resch, M., & Gabriel, E. (2011). Supercomputers in Grids. *Cloud, Grid and High Performance Computing: Emerging Applications* (s. 1-9). içinde Almanya: IGI Global.

Ristov, S., & Gusev, M. (2013). Performance vs Cost for Windows and Linux Performance vs Cost for Windows and Linux. *Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on.* San Francisco.

Sen, Ranjan;Microsoft Corporation. (2008, Eylül). *Developing Parallel Programs*. (Microsoft Corporation) Temmuz 21, 2016 tarihinde https://msdn.microsoft.com/en-us/library/cc983823.aspx adresinden alındı

Sinha, A., Saini, T., & Srikanth, S. V. (2014). Distributed Computing Approach to Optimize Road Traffic Simulation. *IEEE Parallel, Distributed and Grid Computing (PDGC)*, (s. 360-364). Solan.

Skendzic, A., Kovacic, B., & Jugo, I. (2011). Decreasing Information Technology expenses by using emulators on Windows and Linux Platforms. *MIPRO, 2011 Proceedings of the 34th International Convention.* Opatija.

Stallings, W. (2012). *Operating Systems: Internals and Design Principles.* New Jersey: Prentice Hall.

Tanaka, K., Uehara, M., & Mori, H. (2008). A Case Study of a Linux Grid on Windows using Virtual Machines. *22nd International Conference on Advanced Information Networking and Applications.* Okinawa.

The MathWorks, Inc. (tarih yok). *MATLAB MapReduce and Hadoop*. (The MathWorks, Inc.) Temmuz 23, 2016 tarihinde http://www.mathworks.com/discovery/matlab-mapreduce-hadoop.html?requestedDomain=www.mathworks.com adresinden alındı

Udoh, E. (2011). *Cloud, Grid and High Performance Computing: Emerging Applications.* Fort Wayne Indiana: IGI Global.

Vasoya, A., & Koli, N. (2016, Mart). Mining of association rules on large database using distributed and parallel computing. *Procedia Computer Science, 79*, 221-230.

Werner, D. (2008). *Data Mining Techniques in Grid Computing Environments.* UK: John Wiley & Sons, Ltd.

Wilkinson, B., & Allen, M. (2005). *Parallel Programming Techniques And Applications Using Networked Workstations And Parallel Computers.* Upper Saddle River, NJ: Pearson Prentice Hall .

Witten, I. H., & Frank, E. (2005). *Data Mining Practical Machine Learning Tools and Techniques* (2 b.). San Francisco, CA, USA: Elsevier Inc.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining Practical Machine Learning Tools and Techniques.* Burlington,, Amerika Birleşik Devletleri: Morgan Kaufmann Publishers.

Yang, Z., Zhang, C., Hu, M., & Lin, F. (2015). OPC:A Distributed Computing and Memory Computing-based Effective Solution of Big Data. *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, (s. 50-53). Chengdu.