

**KOCAELİ ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**DAĞITIK SİSTEMLERE GİRİŞ  
PROJE RAPORU**

# **CLUSTER COMPUTING**

**Hakan ÖRCÜN  
Selin Gizem ÖZKAN  
Tuğba ERDEM  
Sinan ÇALIŞIR  
İbrahim ÖZTÜRKLER**

**Doç. Dr. Ahmet SAYAR**

**KOCAELİ 2018**

# İçindekiler

## Giriş

Çeşitli mühendislik uygulamalarında yüksek performanslı bilgisayarlara olan ihtiyaç her geçen gün daha da artmaktadır. Yüksek performanslı bilgisayarlar sayesinde geliştirim sürecinde zamandan tasarruf sağlanır. Böylece yeni ürünler piyasaya çok daha kısa periyotlarda çıkar.

Yüksek performanslı bir sistem işlemcinin gücünün artırılması ile ya da donanımın paralelleştirilmesi ile oluşturulabilir. İşlemci performansı belirli nedenlerden dolayı belirli bir seviyeye kadar artırılabilir. Bu nedenle daha fazla donanımın paralel olarak kullanılmasıyla performans artırılması en çok tercih edilen yoldur [10].

## Şekiller dizini

Şekil 1. Cluster Örneği (Linux).....	1
Şekil 2. Broadcast fonksiyon örneği.....	
Şekil 3. Reduction fonksiyon örneği.....	

## Tablolar dizini

Tablo 1.Reduction fonksiyonunda kullanılan bazı operatörler.....	
--	--

## Simgeler ve kısaltmalar dizini

LAN	: Local Area Network
OpenVMS	: Open Virtual Memory System
SSH	: Secure Shell

## 1.Dağıtık Sistem

Kullanıcılara tek ve tutarlı bir sistem olarak görünen bağımsız bilgisayarlardan oluşan koleksiyona dağıtık sistem denir [11]. Dağıtık hesaplama yöntemlerinde işlem parçaları, ağ (network) üzerinde birbirine bağlı işleme ünitelerine dağıtılır. Bağlantı ağ üzerinde olduğundan son derece ölçeklenebilir bir sistem oluşturulmuş olur.

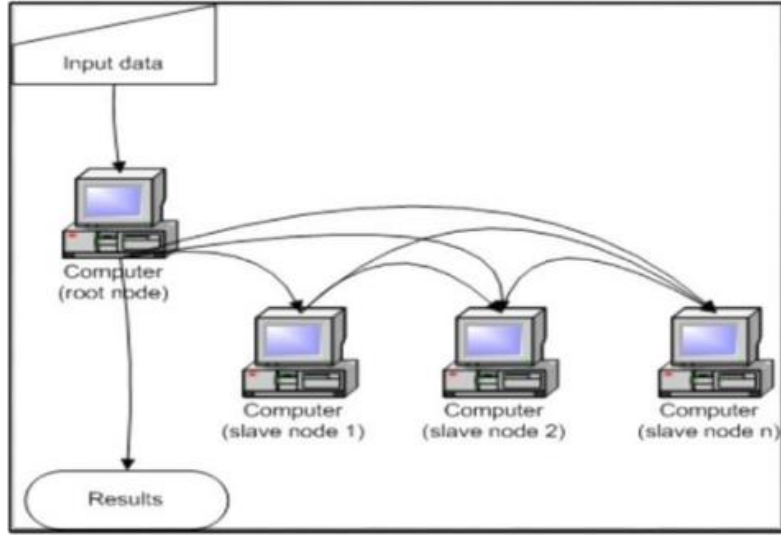
Dağıtık sistemlerin var oluş amacı kaynakları paylaşmaya duyulan gerekliliktir. Bu kaynaklar donanımsal bileşenler olabileceği gibi, dosyalar, veri tabanı, nesneler gibi yazılım varlıkları da olabilir.

Dağıtık hesaplama yapmanın üç çeşidi vardır.

1. Cluster Computing
2. Grid Computing
3. Cloud Computing

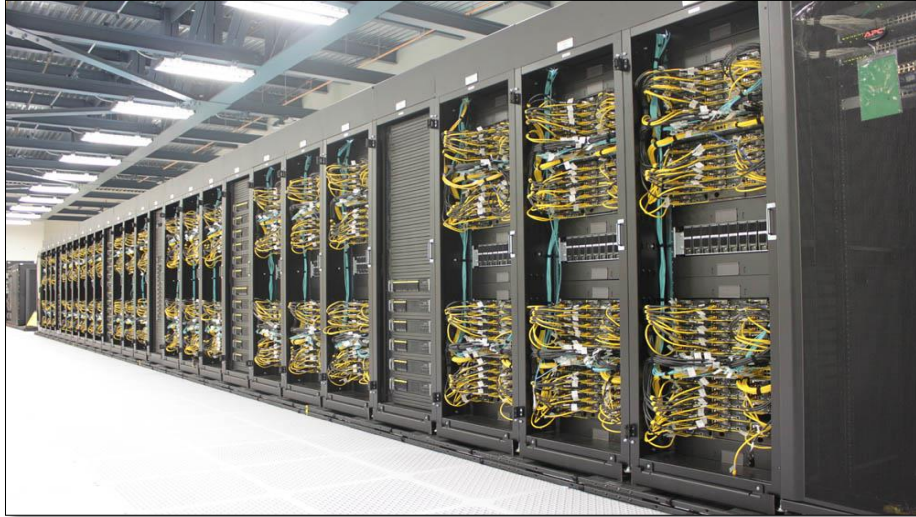
Grid Computing yerini zamanla Cloud Computinge bırakmıştır. Cloud Computingin temellerini Grid Computing oluşturmuştur. Bu proje kapsamında bu konulara değinilmeyecektir.

## 2.Cluster Computing



Şekil1. Cluster Computing

Dağıtık hesaplama yapmanın bir çeşidi olan Cluster Computing daha fazla işlem gücü elde etmek için birden fazla bilgisayarın bir ağ üzerinde birbirine bağlanarak tek bir bilgisayar gibi çalışmasıdır. Bu yöntemle birleştirilen bilgisayarların her biri bu ağdan bağımsız da çalışabilen (stand-alone) bilgisayarlardır. Clusterlar genellikle LAN oluşturulur. Clusterlar e-ticaret, yüksek performanslı veri tabanı uygulamalarını destekleyecek şekilde geliştirilmiştir. Dünyanın en hızlı ilk 500 bilgisayarı içinde en çok kullanılan yöntem budur.



Şekil2. Cluster Örneği (Linux Şirketinin)

Linux, dünya çapında Cluster Computing için bugüne dek en çok kullanılan işletim sistemidir.

## 2.1.Cluster Computing Tarihçesi

Cluster kavramı ilk olarak 1960'larda Greg Pfister'in müşterileri tarafından çalışmalarının tek bir bilgisayara sığmadığını veya desteğe ihtiyaç duyduklarında dile getiriliyor.

Cluster Computing kavramı ise 1967'de IBM'de çalışan Gene Amdahl'ın paralel çalışma yapmanın bir aracı olduğunu dile getirdiği bir Amdahl'ın kanunları adlı makalesinde bahsedilmiştir.

Cluster olarak tasarlanan ilk üretim sistemi, 1960ların ortalarında Burroughs B5700'dür. Bu her biri bir ya da iki işlemciye sahip olan ve iş yükünü dağıtmak için ortak bir disk depolama alt sistemine bağlanan dört bilgisayardan oluşmuştur.

Ticari anlamda ilk Clustering ürünü Datapoint tarafından 1977'de geliştirilen ARCnet'dir.

Daha sonra DEC firması VAX/VMS işletim sistemi (şimdiki OpenVMS) için 1984'te VAXcluster ürünün piyasaya sürdü.

1994 senesinde NASA'da Beowulf projesi ile Cluster Computing başlamıştır.16 tane Intel 486 DX4 işlemci ethernet ile bağlanmıştır.

Microsoft, Sun Microsystems ve diğer donanım ve yazılım şirketleri de Clustering paketleri sunmuştur.

## 2.2.Cluster Computing Avantajları

### 2.2.1. Maliyet Verimliliği

Küme tekniği, üretilen güç ve işlem hızı için uygun maliyetlidir. Ana bilgisayarların kurulumu gibi diğer çözümlere göre daha verimli ve çok daha ucuzdur.

### 2.2.2. İşlem hızı:

Birden fazla yüksek hızlı bilgisayar, birleştirilmiş işleme sağlamak için birlikte çalışır ve böylece genel olarak daha hızlı işlemeye başlar.

### 2.2.3.Gelişmiş ağ altyapısı:

Bir bilgisayar kümesi oluşturmak için farklı LAN topolojileri uygulanır. Bu ağlar, darboğazları önleyen yüksek verimli ve etkili bir altyapı oluşturur.

### 2.2.4.Esneklik:

Ana bilgisayarlardan farklı olarak, bilgisayar kümeleri mevcut özellikleri geliştirmek veya sisteme fazladan bileşenler eklemek için yükseltilebilir.

### 2.2.5. Yüksek kaynak kullanılabilirliği:

Bir bilgisayar kümesinde herhangi bir tek bileşen arızalanırsa, diğer makineler kesintisiz işleme sağlamaya devam eder. Bu artıklık ana sistemlerde eksiktir. [28]

## 2.3.Cluster Computing Dezavantajları

### 2.3.1. Maliyet yüksektir:

Kümenin iyi bir donanıma ve bir tasarıma ihtiyaç duyması nedeniyle, kümelenmemiş bir sunucu yönetim tasarımına kıyasla maliyetli bir maliyet olacaktır. Uygun maliyetli olmamak, bu özel tasarımın temel bir dezavantajıdır.

### 2.3.2.Donanım sıkıntısı:

Kümeleme, bir tane oluşturmak için daha fazla sunucuya ve donanıma ihtiyaç duyduğundan, izleme ve bakım zor olmaktadır.[27]

Sağladığı bu faydaların yanında Cluster yapmaya elverişli olmayan sistemleri Cluster yapmaya çalışmak faydadan çok zarar sağlayacaktır. Bu nedenle paralel programlanamayacak bir sistemi dağıtmaya çalışmak tek bir bilgisayar ile hesaplama yapmaktan daha çok maliyet getirecektir.

## 3.MPI (Message Passing Interface)

MPI (Message Passing Interface) bir bilgisayar iletişim protokolüdür. Dağıtık bellekli bir sistemde paralel program koşan düğümlerin arasındaki iletişim için kullanılan fiilen standart bir protokoldür. MPI uygulamaları Fortran, C, C++ ve Ada programlarından çağrılan kütüphane yordamlarından oluşur. MPI'nın diğer eski mesaj geçirmeli kütüphanelere olan üstünlüğü; taşınabilir (MPI pek çok dağıtık bellekli mimari üzerinde uygulanmıştır) ve hızlı olmasıdır.

MPI, temel olarak süper bilgisayar üzerinde yürütülen bir işlemin tüm bilgisayarlara parçalanarak ayrı ayrı olarak dağıtılması için kullanılan mesaj gönderme ara yüzüdür [6,7]. MPI, C'de bir fonksiyon, Fortran'da bir alt rutindir ve işlemciler arasında haberleşmeyi sağlamak amacıyla kullanılır. Tüm işlem süreci boyunca işlemciler aralarında haberleşirler. Bu nedenle, MPI kütüphanesi bu haberleşmeleri sağlayacak fonksiyon ve alt rutinleri de içermektedir. İlk MPI çalışmalarına Kasım 1992'de resmi olarak ilk sunum yapılması ile başlanmıştır.[6,7] MPI standardı yaklaşık elli kişilik bir grup tarafından birçok farklı organizasyonda tartışılarak geliştirilmiştir. 1993 yılında süper bilgisayarlar için geliştirilen ilk programların taslakları sunulmuş ve bir yıl sonra ilk MPI standardı olan MPI-1.0 bir çok farklı platformda denerek tamamlanmıştır.[8,9] Bu standart, Fortran 77 ve C'de çağrılacak alt rutin ve fonksiyon isimlerini ve sonuçlarını içermektedir. Nisan 1995'de MPI-2 çalışmalarına başlamıştır. Haziran 1995'de MPI-1.1, Temmuz 1997'de MPI-1.2 ve 1997'nin sonlarında ise, MPI-2 sunulmuştur. MPI-2, MPI-1.1'den farklı olarak paralel I/O işlemleri için paketleri, Fortran 90 dosyalarını ve dinamik işlem yöneticisi gibi ek özellikleri içermektedir.

### 3.1.Genel Programlama Yapısı

Bir C programı düşünülürken MPI fonksiyonlarının yer aldığı bir programın yapısı aşağıdaki gibidir [1].

1. Header dosyalarının eklenmesi
2. MPI veri türlerinin belirlenmesi
3. MPI'nin başlatılması
4. MPI haberleşmelerinin yapılması
5. Program içinde yer alan gerekli hesaplamaların yapılması
6. MPI'nin kapatılması
7. Programın sonlandırılması

MPI'da C fonksiyonlarının tümü MPI\_Xxxxx şeklindedir. MPI fonksiyonlarının kullanılabilmesi için C kodlarına başına #include ya da #include "mpi.h" şeklinde MPI header eklenmelidir. Başlangıç MPI fonksiyonu, MPI\_Init(&argc,&argv) fonksiyonudur ve bu fonksiyon MPI programın çalışıp çalışmadığını kontrol eder. Prosesler arası haberleşmeler, MPI\_Comm\_size (comm,&size) ve MPI\_Comm\_rank(comm,&rank) fonksiyonları ile sağlanır. Birinci fonksiyon, sistemde bulunan işlemci (node) sayısını verir. İkinci fonksiyon ise, işlemcilerin 0'dan N'e kadar sayısal olarak sıralamasını sağlar[2,3]. Bu fonksiyon kullanılmazsa, bütün işlemciler 0 ya da -1 değerini alır. MPI'nin sonlandırılması, MPI\_Finalize( ) fonksiyonu ile sağlanır. Bu nedenle bu fonksiyon kodun en sonuna eklenmelidir.

### 3.2.MPI ile Haberleşme

MPI bir haberleşme protokolüdür ve MPI fonksiyonları temel olarak node'ların aralarında haberleşmelerini sağlar. MPI ile iki temel haberleşme şekli vardır[1]. Birincisi, herhangi bir node'un bir mesajı göndermesi ve diğer node'un ya da node'ların bu mesajı alması şeklinde gerçekleşen haberleşme türüdür. İkincisi ise, node'lar arasında verilerin gönderilmesini ve alınmasını içeren haberleşme türüdür.

#### 3.2.1.Point-to-Point Haberleşme

Point-to-point haberleşme bir node'un herhangi bir mesajı diğer node ya da node'lara gönderilmesi ve bu mesajın tüm node ya da node'lar tarafından alınması sürecini kapsar. Point-to-point haberleşmesi, bir node'un bir mesajı gönderip diğer node'ların bu mesajı alması prensibine göre çalıştığından işlem süresince senkronizasyon yoktur[2,3]. Birinci node başlangıç olarak mesaj gönderir. Daha sonra ikinci node mesajı alarak işlemi sürdürür. Yani point-to-point haberleşmesi mesaj gönderme ve alma şeklinde iki yönlü işler. Bu haberleşme boyunca her iki node'un olaya katılması gerekir. Aksi takdirde veri transferi gerçekleşemez[16]. MPI'da mesajlar, mesaj zarfı ve mesaj gövdesi olmak üzere iki kısımdan oluşur. MPI mesaj zarfı, üzerinde gideceği hedefin adresinin, dönüş adresinin, göndermek ve alınmak için gerekli diğer bilgilerin yer aldığı günlük hayatımızda kullandığımız mektup zarfına benzer. MPI mesaj zarfı, gönderilecek işlemciyi gösteren kaynak, alıcı işlemciyi gösteren hedef, kaynak ve hedef işlemcilere ait özel haberleşme özelliklerini içeren haberleştirici ve mesajı etiketlendirmek amacıyla kullanılan işaretçi olmak üzere toplam dört kısımdan oluşur. Mesaj gövdesi ise, mesaj verisi, mesaj veri tipi ve mesaj veri sayısı olmak üzere üç kısımdan oluşur[7]. Point-to-point haberleşmesinde, mesaj gönderilmesi "send" argümanı ile, alınması ise "receive" argümanı ile gerçekleştirilir.

### SEND Parametresi

MPI\_Send fonksiyonu, istenilen bir veriyi herhangi bir işlemciye göndermek amacıyla kullanılır. Bu fonksiyonla tek bir değişken gönderilebileceği gibi bir dizi ya da bir dizinin herhangi bir kısmı da gönderilebilmektedir. Bir mesaj MPI\_Send fonksiyonu kullanılarak gönderildiğinde, mesaj ilk olarak hafızaya alınır ve bloklama fonksiyonları kullanılmamış ise, senkronize olmaksızın hedef işlemciye gönderilir. Gönderilen mesaj mevcut hafızadan daha fazla yer kaplıyorsa, hafıza alanı kullanılamaz ve mesajı göndermekte olan işlemci yeterli hafıza alanı sağlanana kadar mesajı bloklar. MPI\_SEND parametresinin yapısı aşağıdaki gibidir. MPI\_Send(&input\_veri,input\_veri\_sayısı, input\_veri\_tipi,hedef\_node'un\_rankı, işaret\_sayısı,MPI\_COM\_WORLD)

### RECIEVE Parametresi

MPI\_Recv fonksiyonu, node'lar tarafından mesajın gönderildiğini ve bu mesajın kullanıma hazır olduğunu belirtmesi amacıyla kullanılır. Bu amaçla MPI\_Recv fonksiyonu, MPI\_Send fonksiyonundan farklı olarak status şeklinde ekstradan bir argüman içerir. Bir veri hedef node ulaştığında, mesajı işaretine, kaynağına ve alınan veri sayısına status argümanı yardımı ile ulaşılabilir. Eğer alınan mesaj gönderilen mesajdan fazla ise, mesaj gönderme alma sürecinde hata olduğu anlamına gelir. Yani her durumda alınan mesaj ile gönderilen mesaj miktarı aynı olmalıdır. Ayrıca dikkat edilmesi gereken diğer önemli bir nokta ise, gönderilen ve alınan mesaj veri tiplerinin aynı olmasıdır. MPI\_Recv parametresinin yapısı aşağıdaki gibidir.

MPI\_Recv(&output\_veri, output\_veri\_sayısı, output\_veri\_tipi, kaynak\_node'un\_rankı, işaret\_sayısı,MPI\_COM\_WORLD, &status)

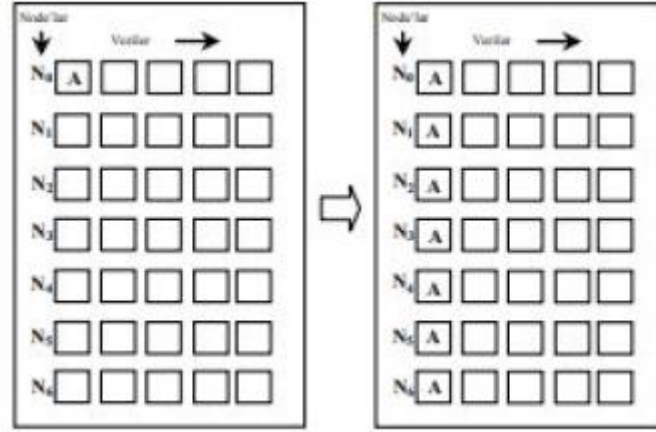
### 3.2.2. Kolektif Haberleşme

Kolektif haberleşme, bir grup node arasında verilerin gönderilmesini ve alınmasını kapsar. Genel olarak düşünüldüğünde, kolektif haberleşme rutinleri kullanılmaksızın yapılacak tüm işlemler MPI\_Send ve MPI\_Recv fonksiyonları kullanılarak yapılabilir. Fakat kolektif haberleşme fonksiyonları, bu işlemlerin çok daha kısa bir şekilde yapılmasını sağlar. Ayrıca point-to-point haberleşmeden farklı olarak kolektif haberleşme fonksiyonları yardımıyla, bir node'un birden fazla node'a (one-to-several) ya da diğer node'ların herhangi bir node'a (several-to-one) veri transferi yapmasına olanak sağlanır[8,9]. Kolektif haberleşmenin kullanılması ile program içinde yazılacak MPI kodlarında önemli ölçüde bir azalma olur. Bu ise, program yazılması sırasında yapılacak hatalarının azalması anlamında gelir. Ayrıca kolektif haberleşme rutinleri kullanılması ile node'lar arası yapılan haberleşmeler azaldığından çalıştırılan programın daha kısa sürede tamamlanması sağlanabilmektedir.

**Barrier Senkronize Edici Fonksiyon:** Bu fonksiyon, node'lar arasındaki senkronizeliği sağlamak amacıyla kullanılır ve her bir node herhangi bir işlem yapmadan önce diğer node'ların belirlenen sürece kadar beklemesini sağlar. MPI\_Barrier fonksiyonunun yapısı aşağıdaki gibidir.

```
.....  
.....  
MPI_Barrier (MPI_COMM_WORLD)  
.....  
.....
```

**Broadcast Fonksiyonu:** Şekil 1'de görüldüğü gibi broadcast fonksiyonu ile ana node'daki istenilen bir veri, haberleşme grubu içinde yer alan diğer tüm node'lara dağıtılır.



Şekil 2. Broadcast fonksiyon örneği

MPI\_Bcast fonksiyonunun yapısı aşağıdaki gibidir.

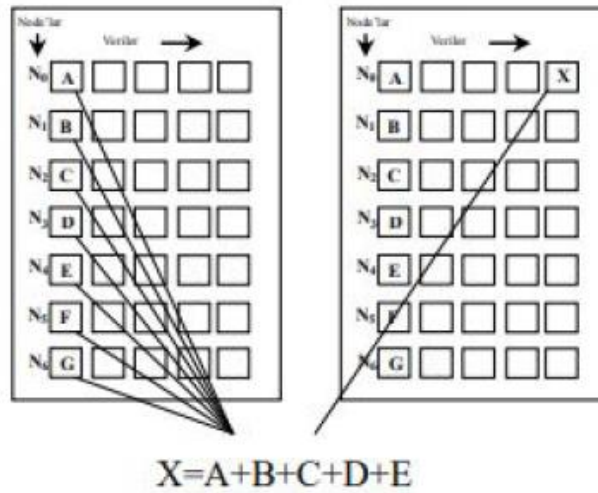
```

.....
MPI_Bcast (&input_veri,input_veri_sayısı,
input_veri_tipi,ana_node'un_rankı,MPI_COMM_WORLD)
.....

```

### Reduction Fonksiyonu

Reduction fonksiyonu, her bir node'dan istenilen verilerin alınarak toplama, çarpma, minimum ya da maksimum değeri bulma gibi işlemleri gerçekleştirerek ana node'da saklanmasını sağlar[6]. Şekil 2'de verilen reduction fonksiyonu tüm node'lardan aldığı verileri toplayarak ana node'da aktarmaktadır.



Şekil 3. Reduction fonksiyon örneği

Tablo 1'de Reduction fonksiyonu içinde kullanılan operatörler verilmiştir. Bu operatörler yardımı ile node'lardan gelen veriler basit bir şekilde aşağıdır.



Operatör	Açıklaması
MPI_MAX	Maksimum Değer
MPI_MIN	Minimum Değer
MPI_SUM	Toplama işlemi
MPI_PROD	Çarpma işlemi
MPI LAND	Mantıksal AND
MPI_LOR	Mantıksal OR

Tablo 1.Reduction fonksiyonunda kullanılan bazı operatörler

## 4. SSH

SSH(Secure Shell/Güvenli Kabuk) ağ üzerinden başka bilgisayarlara erişim sağlamak, uzak bir bilgisayarda komutlar çalıştırmak ve bir bilgisayardan diğerine dosya transferi amaçlı geliştirilmiş bir protokoldür. Güvensiz kanallar(internet vs) üzerinden güvenli haberleşme olanağı sağlar. Bir iletişimde SSH aşağıda belirtilen temel unsurları sağlar.

- Authentication /Kimlik denetimi
- Encryption /Şifreleme
- Integrity /Bütünlük.

### 4.1. Kullanım Alanları

SSH güvenli iletişimin gerektiği her ortamda kullanılabilir. Sadece karşı sisteme bağlanıp komut çalıştırmak ya da dosya aktarımı yapmak için değil, doğasında güvensiz(şifrelenmemiş trafik) olarak çalışan protokoller SSH üzerinden güvenli bir şekilde kullanılabilir. Mesela POP3 servisi ağ üzerinden tüm iletişimini şifrelenmemiş şekilde gerçekleştirir, biz pop3 servisini SSH üzerinden aktararak şifrelenmiş ve güvenli hale getirebiliriz.

### 4.2. Nasıl Çalışır?

SSH, private ve public key olmak üzere 2 adet anahtar üzerinden çalışmaktadır. Bu anahtarlardan public key verileri şifreleyecek ve güvenli erişim talebinizi doğrulayacak şekilde, uzaktan erişim sağlamak istediğiniz sunucuya yerleştirilmektedir. Private Key ise şifrelenmiş verileri çözmek üzere istemci tarafında bulundurulmuş ve gizliliğine oldukça önem verilmesi gereken anahtardır.

Anahtar tabanlı kimlik doğrulamanın en temel kullanımı, otomasyonu güvenli hale getirmektir. Otomasyonu sağlanmış SSH ile dosya aktarımları ve uygulamalar sorunsuz bir şekilde entegre edilebilmekte ve konfigürasyon yönetimi kolaylıkla gerçekleştirilebilmektedir[24].

### 4.3. SSH Kullanımı

SSH genellikle uzak makineye giriş yapmak ve komutları çalıştırmak için kullanılır, ayrıca tünel açma, TCP portlarını ve X11 bağlantılarını iletmeyi de destekler; SSH dosya transferi (SFTP) veya güvenli kopyalama (SCP) protokolleri kullanarak dosyaları aktarabilir. SSH istemci-sunucu modelini kullanır.SSH sunucularına bağlanmak için atanmış standart TCP port numarası 22 dir[25].

### 4.4. SSH Kurulumu

OpenSSH Kurulumu OpenSSH birçok Unix/Linux dağıtımı ile öntanımlı olarak gelmektedir. OpenSSH kurulumu Kullandığınız Linux/UNIX dağıtımına göre değişiklik gösterebilir.

Sistemde kurulu değilse Kullanılan paket yönetim sistemi kullanılarak son sürüm OpenSSH sisteme kurulur.

```
$ sudo apt-get install openssh-server
```

kodu ile OpenSSH kurulumu gerçekleşir[26].

#### 4.5. Anahtar Yönetimi

Unix benzeri sistemlerde, yetkilendirilmiş açık anahtarların listesi genellikle ~/.ssh/authorized\_keys dosyasında, uzaktan oturum açmasına izin verilen kullanıcının giriş dizininde saklanır. Bu dosya, sahibi ve tüm yetkili kullanıcıdan başka bir şey tarafından yazılabilir değil ise sadece SSH tarafından değiştirilebilir. Uzak uçta açık anahtar varsa ve yerel uçta eşleşen gizli anahtar varsa, parolayı yazmanız artık gerekmemektedir (Message-Passing Interface (MPI) yığını gibi bazı yazılımların düzgün çalışması için bu parolasız erişim gerekebilir). Ancak, ek güvenlik için gizli anahtarın kendisi bir şifre ile kilitlenebilir.

Gizli anahtar standart yerlerde de aranabilir ve tam yolu bir komut satırı ayarı olarak belirlenebilir (ssh için -i seçeneği). Ssh-keygen yardımcı programı, her zaman çiftler halinde açık ve gizli anahtarlar üretir.

SSH, ayrıca otomatik olarak oluşturulan anahtarlar tarafından şifrelenen parola tabanlı kimlik doğrulamayı da destekler. Bu durumda, saldırgan meşru sunucu tarafını taklit edebilir, parola isteyebilir ve parolayı elde edebilir (man-in-the-middle saldırısı). Ancak, bu durum sadece, iki tarafın daha önce hiç birbirlerinin kimliklerini doğrulamadığında SSH'nin daha önce kullanılan sunucu tarafının anahtarını hatırladığı için mümkündür. SSH istemcisi, önceden bilinmeyen yeni bir sunucunun anahtarını kabul etmeden önce bir uyarı verir. Parola kimlik doğrulaması kapatılabilir[25].

#### 4.6. SSH Bağlantısı

SSH server kurulu olduğu için ssh <username>@<hostname> ile diğer bilgisayarlara giriş yapılabilir. Bu komuttan

sonra bizden bir şifre istenir. Giriş işleminin daha kolay olması adına, anahtarlar(keys) üretip istemci bilgisayarların authorized\_keys dosyasına kopyalamamız gerekir.

Bu işlem için ssh-keygen -t dsa komutu ile anahtar üretilir ve ssh-copy-id <hostname> ile diğer bilgisayarın authorized\_keys dosyasına üretilen anahtar kopyalanır. Kopyalama işlemi her bir istemci bilgisayar ve sunucu bilgisayar(localhost) için tekrarlanır. Bu adımlar tamamlandıktan sonra openssh-server kurulmuş olur ve istemci bilgisayarlar ile güvenli bir şekilde iletişim sağlanır. Her bir bilgisayarın bilinen host'lara(known\_hosts) eklenebilmesi için bir kerelik ssh ile bağlanması gerekir. Bu adımdan sonra şifresiz ssh'ın aktif edilmesi için eval `ssh-agent` ve ssh-add ~/.ssh/id\_dsa komutları çalıştırılır. Belirtilen adımların hepsini tamamladıktan sonra ssh <hostname> ile şifre kullanmadan güvenli bir şekilde istenilen bilgisayara bağlanılabilir.

#### 5.NFS

Ağ Dosya Sistemi(Network File System), 1984 yılında Sun Microsystems tarafından geliştirilen dağıtık dosya sistemidir. Kullanıcılara, ağ üzerinden, istemci bilgisayardaki dosyalara yerel bilgisayardaymış gibi erişim sağlamaya izin veren bir protokoldür. Benzer bütün protokoller gibi, Açık Ağ Hesaplama Uzak Yordam Çağırımı (Open Network Computing Remote Procedure Call  
ONC RPC) üzerine kuruludur.

## 5.1. Versiyonlar ve Çeşitleri

### 5.1.1. NFSv2

Protokolün ikinci versiyonu(Mart 1989'da RFC 1094[12] tanımlandı), sadece User Datagram Protocol(UDP) üzerinde çalışıyordu. Tasarımcıları, sunucu tarafını durumsuz(stateless) çekirdek protokolün dışında itleme(locking) ile uygulamıştır. Sanal Dosya Sistemi(Virtual File System) arayüzün modüler uygulamaya izin vermesiyle birlikte basit bir protocol olmuştur. Şubat 1986'dan itibaren, uygulamalar Eunice kullanılarak DOS, VAX/VMS gibi işletim sistemlerinde denenmiştir[13].

### 5.1.2. NFSv3

NFS üçüncü versiyonu, ikinci versiyondan yakın bir zaman sonra çıkarılmıştır.[14] Genel motivasyon ikinci versiyondaki yazma performansında yaşanan problemleri azaltmaktır. Haziran 1992'den itibaren ikinci versiyondaki çoğu problem pratikte çözülmüştür ve geriye sadece 2 GB'lık boyut kısıtlaması problemi kalmıştır. Bu problem, Ultrix'in yeni 64-bitlik RISC işlemcilerini sunmasıyla şiddetli bir probleme dönüşmüştür. Üçüncü versiyonun tanıtıldığı dönemde, taşıma katmanında TCP protokolü yaygınlaşmaya başlamıştır. Bazı satıcılar, ikinci versiyona TCP desteği eklemesine rağmen Sun Microsystems bu desteği üçüncü versiyonda eklemiştir. TCP'yi kullanmak, NFS'i Wide Area Network(WAN) üzerinde kullanmayı daha mümkün kılmıştır ve UDP tarafından konulan 8 KB'lık okuma ve yazma limitlerini ortadan kaldırmıştır. Üçüncü versiyonda eklenen bazı özellikler:

- \* 64-bitlik dosya boyutlarına ve 2 GB'dan büyük dosyalar ile çalışma desteklenmiştir.
- \* Sunucuda yazma performansını arttırmak için asenkron yazmalar desteklenmiştir.

### 5.1.3. NFSv4

NFS dördüncü versiyonu[15], Andrew File System(AFS) ve Server Message Block(SMB or CIFS) etkilenilmiş, performans artışları, güçlü güvenlik desteği ve durumlu(stateful) protokol tanıtılmıştır. Dördüncü versiyon, Internet Engineering Task Force(IETF) ve Sun Microsystems tarafından birliktengeliştirilmiş ilk versiyondur. NFS 4.1 versiyonu[16] kümelenmiş sunucuların(clustered server) avantajlarını kullanarak çalışma ve paralel sistemlerde, çoklu sunucular üzerinde dağıtılmış dosyalara erişim imkanı sunmayı hedeflemiştir.

NFS 4.2 versiyonu[17] Kasım 2016'da yayınlanmış olup aşağıdaki özellikler eklenmiştir:

- \* Sunucu-tarafli çoğaltma ve kopyalama
- \* Uygulama girdi/çıkışı
- \* Seyrek dosyalar(Sparse files)
- \* Bellek rezervi
- \* Uygulama veri bloğu(ADB)

### 5.1.4. Diğer Eklentiler

WebNFS, versiyon 2 ve versiyon 3'e uzantı/ekleme olarak yayımlanmış ve NFS'in Web tarayıcıları ile güvenlik duvarları(firewalls) aracılığıyla daha kolay entegre olmasına imkan sunmuştur. Sun Microsystems 2007 yılında istemci-tarafli WebNFS uygulamasını açık kaynak olarak sunmuştur[18].

NFS ile ilişkili çeşitli protokoller:

- \* Bayt-aralıklı danışman Network Lock Manager(NLM) protokolü
- \* NFS kullanıcılarına, NFS sunucularında ne kadar veri saklama kotası kaldığını gösteren, uzaktan kota-rapor protokolü(RQUOTAD) (Örneğin; Google Drive)
- \* Taşıma olarak uzaktan direkt bellek erişiminin(remote direct memory access RDMA) adaptasyonu

\* Güvenilir NFS[19].

## 5.2.Platformlar

NFS, genel olarak Unix Unix benzeri işletim sistemleri ile birlikte kullanılır. Ayrıca MS-DOS[20], Microsoft Windows[21], Novell NetWare[22] ve IBM AS/400[23] gibi işletim sistemleri ile de kullanılabilir.

Server Message Block(SMB) ve NetWare Core Protocol(NCP) Microsoft Windows kullanan sistemlerde NFS'e kıyasla daha çok kullanılır. Apple Filling Protocol(AFP), Apple Macintosh kullanan sistemlerde NFS'e kıyasla daha çok kullanılır.

## 5.3.Genel Uygulama

Unix-tarzı sistem kullanıldığı varsayılarak, bir sunucu makinenin bir başka bilgisayarda (NFS sunucu) bulunan veriye erişmek istediği senaryosu düşünüldüğünde:

1. Sunucu NFS'e yardımcı daemon işlemleri, nfsd'i çalıştırarak, veriyi istemcilere görünür kılmak için uygular/başlatır,
2. Sunucu yöneticisi nelerin görünür olması gerektiğine karar verir ve /etc/exports içerisinde, istenilen klasörlerin parametreleri ve isimleri kaydedilerek exportfs komutu ile hazırlanır.
3. Sunucu güvenlik-yöneticisi, istemcilerin tanındığından ve doğrulandığından emin olur.
4. Sunucu ağ yapılandırması güvenlik duvarı üzerinden uygun istemciler ile görüşülebildiğinden emin olur.
5. İstemci makine, mount komutunu kullanarak paylaşılan dosyaya erişmek için istekte bulunur (İstemci, sunucuya NFS'in hangi portu kullandığını sorar ve sonrasında NFS sunucuya bağlanır).
6. Bu adımlara kadar hata alınmazsa istemci makine, izin verilen dosyalara ağ üzerinden erişim hakkı elde eder ve kullanılan versiyona göre UDP veya TCP üzerinden haberleşme sağlanır.

## 6.Projenin Çalıştırılması

Bu bölümde demoda sunulan programların çalıştırılması ve ilgili kurulumların nasıl yapılacağı Linux işletim sistemi üzerinde anlatılmıştır.

### 6.1.Önkoşullar

#### 6.1.1. MPICH2'nin kurulumu

MPICH2, Argonne National Laboratory tarafından geliştirilen MPI'ın sıklıkla kullanılan uygulamalarından biridir. MPICH2'nin son versiyonu <http://www.mpich.org/> adresinden indirilebilir. Dosya indirildikten sonra:

**tar -xzf mpich2-1.4.tar.gz** komutu ile çıkarılır ve **cd mpich2-1.4** komutu ile dosyanın içine gidilir. Bundan sonra **./configure** komutu ile indirme sırasındaki ayarlamalar yapılabilmektedir. **./configure --disable-fortran** komutu ile fortran kütüphaneleri hariç gerekli kurulumlar yapılır. Yapılandırma işlemi bittiğinde MPICH2 kurulabilir durumdadır ve **make; sudo make install** komutu ile indirme işlemi başlatılır. İndirme işlemi başarılı olarak tamamlanıp tamamlanmadığını kontrol etmek için **mpiexec --version** komutu çalıştırılır. Bu komut ile hangi versiyonun yüklendiği tespit edilir ve indirmenin durumu kontrol edilir.

### Adım 1. hosts dosyasının ayarlanması

Bilgisayarlar arasında iletişim kurulurken sürekli IP adreslerini akılda tutmamak için, ağ içerisinde iletişim kurmak istediğimiz bilgisayarları birbirine tanıtmamız gerekir. hosts dosyası cihaz tarafından host isimlerini IP adresleriyle eşleyen bir dosyadır. Bu dosyanın içerisine iletişim kurmak istediğimiz cihazların host isimlerini ve IP adreslerini giriyoruz. **sudo gedit /etc/hosts** komutu ile hosts dosyası açılır ve içerisine sisteme dahil etmek istediğimiz bilgisayarların host isimleri ve IP adresleri yazılır.

### Adım 2. Yeni kullanıcının oluşturulması

Mevcut kullanıcı hesapları ile de bilgisayar kümesi üzerinde işlem yapılabilir ancak yapılandırma işlemlerini daha basit kılmak için her bir bilgisayarda ayrı ayrı aynı isimde kullanıcı açılır. Her bir bilgisayar üzerinde **sudo adduser <username>** komutu ile yeni kullanıcı açılabilir.

### Adım 3. SSH ayarlanması

Bilgisayarlar ağ üzerinde SSH ile iletişim kurarlar ve NFS ile de veri paylaşımı yaparlar. Bu yüzden ilk adım olarak SSH kurulur. **sudo apt-get install openssh-server** komutu ile SSH indirilir ve 2. adımda oluşturulan kullanıcıya **su - <username>** komutu ile giriş yapılır. Artık SSH server kurulu olduğu için **ssh <username>@<hostname>** ile diğer bilgisayarlara giriş yapılabilir. Bu komuttan sonra bizden bir şifre istenir. Giriş işleminin daha kolay olması adına, anahtarlar(keys) üretip istemci bilgisayarların authorized\_keys dosyasına kopyalamamız gerekir.

Bu işlem için **ssh-keygen -t dsa** komutu ile anahtar üretilir ve **ssh-copy-id <hostname>** ile diğer bilgisayarın authorized\_keys dosyasına üretilen anahtar kopyalanır. Kopyalama işlemi her bir istemci bilgisayar ve sunucu bilgisayar(localhost) için tekrarlanır. Bu adımlar tamamlandıktan sonra openssh-server kurulmuş olur ve istemci bilgisayarlar ile güvenli bir şekilde iletişim sağlanır. Her bir bilgisayarın bilinen host'lara(known\_hosts) eklenebilmesi için bir kerelik ssh ile bağlanması gerekir. Bu adımdan sonra şifresiz ssh'ın aktif edilmesi için **eval `ssh-agent`** ve **ssh-add ~/.ssh/id\_dsa** komutları çalıştırılır. Belirtilen adımların hepsini tamamladıktan sonra **ssh <hostname>** ile şifre kullanmadan güvenli bir şekilde istenilen bilgisayara bağlanılabilir.

### Adım 4. NFS ayarlanması

Master ve client bilgisayarlar arasında veri paylaşımını sağlamak adına NFS üzerinden bir klasör oluşturmamız gerekir.

Sunucu üzerinde NFS kurmak için **sudo apt-get install nfs-kernel-server** komutu çalıştırılır. Hala <username> kullanıcısında olduğu varsayılarak **mkdir <filename>** komutuyla paylaşılması istenen klasör <filename> adıyla oluşturulur. <filename> klasörünü dışa aktarmak için **/etc/exports** dosyasının sonuna **/home/<username>/<filename> \*(rw,sync,no\_root\_squash,no\_subtree\_check)** eklenir. Burada:

rw: Klasör için hem yazma hem de okuma işlemine izin verildiğini,

sync: Yapılan değişikliklerin sadece işlendikten sonra paylaşılan klasöre uygulandığını,

no\_subtree\_check: Eğer paylaşılan klasör çok daha büyük bir dosya sisteminin alt klasörü ise NFS, paylaşılan klasörün üzerindeki bütün klasörleri de tarar. Bu ekleme, tarama işleminin yapılmasını engellemek istediğimizi belirtir.

no\_root\_squash: Root hesapların klasöre bağlanmasına izin verdiğimizizi belirtir. Satır eklendikten sonra **exportfs -a** komutu çalıştırılır ve **sudo service nfs-kernel-server restart** komutu ile NFS sunucusu tekrardan başlatılır.

İstemci üzerinde NFS kurmak için ise `sudo apt-get install nfs-common` komutu çalıştırılır ve `<filename>` ile aynı bir dosya ismi kullanılarak `mkdir <filename>` komutu çalıştırılır. Paylaşılan klasörün bağlanması için **`sudo mount -t nfs <master_name>:/home/<username>/<filename> ~/<filename>`** komutu çalıştırılır. Bu bağlamayı kalıcı yapmak için `/etc/fstab` dosyasına `<master_name>:/home/<username>/<filename> /home/<username>/<filename> nfs yazılır` ve kaydedilir.

### **Adım 5. MPI programlarının çalıştırılması**

Demoda anlatılan programların çalıştırılması için **`mpirun -np <process_sayısı> --hosts <hostname1>,<hostname2>,<hostname3> <calistirilabilir_program>`** şablonunda bir kod çalıştırılır. Bizim örneğimizde **`mpirun -np 10 --hosts hakan,ibrahim,alpbey ./sum_demo`** komutu çalıştırılarak toplama örneği çalıştırılır. Burada işlemi 10 process'e bölmek istediğimizi, hakan, ibrahim, alpbey bilgisayarlarını işlem için kullanmak istediğimizi ve `sum_demo` programını çalıştırmak istediğimizi belirtmiş oluyoruz.

## 6.Kaynakça

- [1] <http://www.beowulf.org>
- [2] <http://www.top500.org/>
- [3] Kaleci, D., Kaya O. A., Karakaplan, M., Şahin A., 'The Linear Acoustic Field Calculation on 15 node Linux Cluster', Balkan Physics Letters, 2008 Special Issue, Boğaziçi University Press, ISSN 1301-8329, 2008, p.215-219.
- [4] Uehara, H., Tamura M., Yokokawa M., 'An MPI Benchmark Program Library and Its Application to the Earth Simulator', Lecture Notes in Computer Science, vol.2327, January 2002,pp 351-356.
- [5] Stefano Cozzini, Axel Kohlmeyer, and Roger Rousseau, 'Benchmark Analysis of 64-bit Servers for Linux Clusters for Application in Molecular Modeling and Atomistic Simulations', Proceedings of the 7th LCI International Conference on Clusters, 2006.
- [6] <http://www-unix.mcs.anl.gov/mpi/>
- [7] <http://lam-mpi.org>
- [8] <http://www.lanl.gov/roadrunner/>
- [9] [http://domino.research.ibm.com/comm/research\\_projects.nsf/pages/bluegene.index.html](http://domino.research.ibm.com/comm/research_projects.nsf/pages/bluegene.index.html)
- [10] <https://www.slideshare.net/saho10/datk-sistemler-programlama>
- [11] [https://drive.google.com/file/d/1MvPU\\_ujUYYYjfGYtMeJ7deuY3drUIJrx/view](https://drive.google.com/file/d/1MvPU_ujUYYYjfGYtMeJ7deuY3drUIJrx/view)
- [12] <https://tools.ietf.org/html/rfc1094>
- [13] Russel Sandberg. "The Sun Network Filesystem: Design, Implementation and Experience" *Technical Report*. Sun Microsystems.
- [14] <https://tools.ietf.org/html/rfc1813>
- [15] <https://tools.ietf.org/html/rfc3010>
- [16] <https://tools.ietf.org/html/rfc5661>
- [17] <https://tools.ietf.org/html/rfc7862>
- [18] <https://yanfs.dev.java.net/>
- [19] Glover, Fred. "A Specification of Trusted NFS (TNFS) Protocol Extensions"
- [20] "Other Software by SUN Microsystems". [www.computinghistory.org.uk](http://www.computinghistory.org.uk). The Centre for Computing History. Erişildi 30 Aralık 2018.
- [21] "Introduction to Microsoft Windows Services for UNIX 3.5". [technet.microsoft.com](http://technet.microsoft.com). Microsoft. Erişildi 30 Aralık 2018.
- [22] "NFS Gateway for NetWare 6.5". [www.novell.com](http://www.novell.com). Novell. Erişildi 30 Aralık 2018.
- [23] "OS/400 Network File System Support" (PDF). [publib.boulder.ibm.com](http://publib.boulder.ibm.com). IBM. Erişildi 30 Aralık 2018.
- [24] <https://medium.com/@vargonen/ssh-nedir-66b1dc0da45>
- [25] [https://tr.wikipedia.org/wiki/Secure\\_Shell\\_\(SSH\)](https://tr.wikipedia.org/wiki/Secure_Shell_(SSH))
- [26] <http://mpitutorial.com/tutorials/running-an-mpi-cluster-within-a-lan/>
- [27] <https://www.quora.com/What-is-cluster-computing-List-disadvantages-of-cluster-computing>
- [28] <https://www.techopedia.com/definition/6581/computer-cluster>