

**KÜMELEME METODUYLA SANAL SINIF
UYGULAMALARINDAKİ PERFORMANS
ARTIŞININ DEĞERLENDİRİLMESİ**

**2011
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ**

Oğuzhan MENEMENCİOĞLU

**KÜMELEME METODUYLA SANAL SINIF UYGULAMALARINDAKİ
PERFORMANS ARTIŞININ DEĞERLENDİRİLMESİ**

Oğuzhan MENEMENCİOĞLU


**Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır**

**KARABÜK
Şubat 2011**

Oğuzhan MENEMENCİOĞLU tarafından hazırlanan “KÜMELEME METODUYLA SANAL SINIF UYGULAMALARINDAKİ PERFORMANS ARTIŞININ DEĞERLENDİRİLMESİ” başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Baha ŞEN

Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı



Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 03 / 02 / 2011

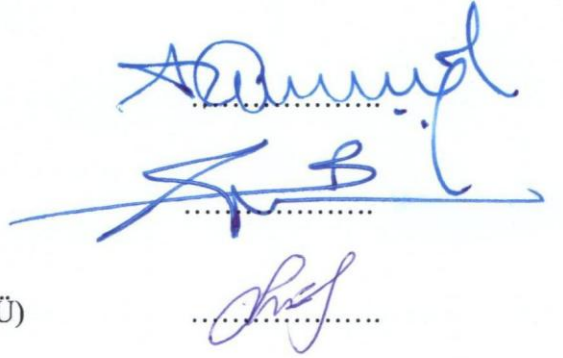
Ünvanı, Adı SOYADI (Kurumu)

İmzası

Başkan : Prof. Dr. Abdullah ÇAVUŞOĞLU (KBÜ)

Üye : Yrd. Doç. Dr. Baha ŞEN (KBÜ)

Üye : Yrd. Doç. Dr. Mehmet Akif NACAR (HRÜ)



...../...../2011

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Doç. Dr. Nizamettin KAHRAMAN

Fen Bilimleri Enstitüsü Müdürü



“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”

Oğuzhan MENEMENCİOĞLU

ÖZET

Yüksek Lisans Tezi

KÜMELEME METODUYLA SANAL SINIF UYGULAMALARINDAKİ PERFORMANS ARTIŞININ DEĞERLENDİRİLMESİ

Oğuzhan MENEMENCİOĞLU

Karabük Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:

Yrd. Doç. Dr. Baha ŞEN

Şubat, 2011, 105 sayfa

Uzaktan eğitimde video kullanımına paralel olarak gelişen video konferans uygulamasında video kalitesini ve hizmet kalitesini arttırmak ve öğrenci sayısının artışından kaynaklanan teknik problemleri en aza indirmek asıl amaç olarak görülmüştür.

Genelde uzaktan eğitim sistemi altyapısı (öğrenim yönetim sistemi ve sanal sınıf uygulaması) tek ve bağımsız sunucular ile tasarlanmaktadır. Her biri açık kaynak kodlu işletim sistemi üzerinde çalışan, öğrenim yönetim sistemi ve sanal sınıf uygulaması ile hizmet verilen belirli sayıda öğrencinin yer aldığı bir uzaktan eğitim sistemi altyapısında, hizmet kalitesini hizmet verilen öğrenci sayısı ile birlikte arttırmanın yolları irdelenmiştir. Çalışmada özellikle sanal sınıf uygulamalarında performans artırımını ile ilgilenilmiştir.

Tek ve bağımsız sunucu ile sağlanan Openmeetings sanal sınıf uygulamasının en uygun ağ altyapısında, hizmet verilebileceği öğrenci sayısı yaklaşık olarak yüz kişidir. Bin öğrencinin üzerindeki sayıya hizmet vermeyi planlayan bir uzaktan eğitim sistemi altyapısında tezin ve bu çalışmanın önemi açıkça ortaya çıkmaktadır.

Sistemde performans artışı sağlamak için ilk akla gelebilecek yöntem yeni, daha güçlü sunucular almaktır. Yeni, daha güçlü sunucular alarak istenen performansı sağlamak hem çok masraflı hem de yeterli ve stabil değildir. Tezde detaylandırıldığı üzere buna alternatif sürdürülebilir, yönetilebilir ve stabil bir yöntem önerilmiş; modellenmiş ve gerçekleştirilmiştir. Bu tezde bin öğrenci sayısının üzerindeki uzaktan eğitim sistemi alt yapı planlamasında, sanal sınıf uygulaması açısından dikkate alınacak yapılandırma çözümleri paylaşılmıştır. Kümelenmiş bir sanal sınıf uygulaması üzerinde dinamik bir yük dengeleme modeli tasarlanarak, uygulaması gerçekleştirilmiştir.

Tasarlanan modelin gerçekleştirilmesi için bir sunucu üzerinde, iki sanal sunucu oluşturulmuştur. Diğer bir sunucu ise öğrenim yönetim sistemi için kullanılmıştır. Böylece hem sanallaştırılmış hem de kümelenmiş bir sistem oluşturulmuştur. Tek bir sunucunun ürettiği sonuçlar ile kümeden elde edilen sonuçlar kıyaslanmış ve tartışılmıştır. Çalışmanın geliştirilebileceği yönler de irdelenerek, çalışma tamamlanmıştır.

Anahtar Sözcükler : Küme, kümeleme, yük dengeleme, video konferans, web konferans, sanal sınıf uygulaması ve öğrenim yönetim sistemi

Bilim Kodu : 902.1.014

ABSTRACT

M.Sc. Thesis

EVALUATION OF THE PERFORMANCE INCREASE ON VIRTUAL CLASSROOM APPLICATION VIA CLUSTERING METHOD

Oğuzhan MENEMENCİOĞLU

Karabuk University

Graduate School of Natural and Applied Sciences

Department of Computer Engineering

Thesis Advisor:

Assist. Prof. Dr. Baha ŞEN

February 2011, 105 pages

Improving the qualities of video and service in video conferencing application developed in parallel with the use of video in distance education and minimizing technical problems arising from the increase in the number of students were considered as the main aim.

In general, infrastructures of distance education system (learning management system and virtual classroom application) are designed with standalone servers. Ways of improving the quality of service in collaboration with the number of students whom the services provided were examined in a distance education infrastructure where a certain number of students involved and the services were provided by the learning management system and virtual classroom application that each of them is running on open source operating system. Increasing the performance server in an appropriate network infrastructure. Therefore, the significance of this

study becomes more apparent when it is aimed to provide a service more than a thousand of students in a distance education system infrastructure.

The first method which is conceivable is buying new and more powerful servers in order to provide performance improvements in the system. Providing the required performance by buying new and more powerful servers is both rather expensive and also is not sufficient and stable. As the details were given in this thesis, an alternative, sustainable, manageable, and stable method is proposed, modelled and implemented within the settings of the study. In this thesis, the configuration solutions that should be taken into consideration were discussed in terms of virtual classroom application on planning the infrastructure for distance education system where the number of the students is over one thousand. A dynamic load-balancing model was designed and its application was developed on a clustered virtual classroom application.

For the realization of the designed model: two virtual servers were formed on a real server. Another server was used for the learning management system. In this way, both virtualized and clustered system was formed. Both results obtained from standalone server and cluster were compared and discussed in the thesis. The study was completed also by examining the developable aspects of system.

Key Words : Cluster, clustering, load balancing, video conference, web conference, virtual class application and learning management system

Science Code : 902.1.014

TEŐEKKÜR

Tez alıőmamda danıőmanlıęını yapan; bilgi birikimi ile bana yol gsteren, tecrübelerinden faydalandıęım Sayın Yrd. Do. Dr. Baha ŐEN'e sonsuz teőekkür ederim.

Sanal iőletim sistemi, Linux, sanal sınıf uygulaması kurulumu gibi konularda yardımını esirgemeyen Sayın Arő. Gör. Ferhat ATASOY'a alıőmalarına katkısından dolayı teőekkür ederim.

Yüksek Lisans eęitimim boyunca da desteklerini esirgemeyen ve onlara vakit ayıramamamı anlayıőla karőılayan aileme yürekten teőekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
KABUL.....	ii
ÖZET	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ.....	xii
ÇİZELGELER DİZİNİ	xiii
SİMGELER VE KISALTMALAR DİZİNİ.....	xiv
BÖLÜM 1	1
GİRİŞ	1
BÖLÜM 2	3
KÜME.....	3
2.1. TARİHÇESİ.....	3
2.2. KÜME TÜRLERİ	11
2.2.1. Yük Dengeleme.....	11
2.2.2. Yüksek Kullanılabilirlik.....	11
2.2.2.1. Aktif-Pasif	11
2.2.2.2. Aktif- Aktif	12
2.2.3. Yüksek Performanslı Hesaplama	12
2.3. BEOWULF	12
2.4. KÜME BİLEŞENLERİ.....	13
2.4.1. Düğümler.....	13
2.4.2. Ara Bağlantılar	14
2.4.3. Hafıza	14
2.4.4. Donanım Yönetimi	14

2.4.5. Yönetim Yazılımı	14
2.4.6. Uygulama Yazılımı	15
2.5. YÜK DENGELEME.....	15
2.5.1. Yük Dengeleme Avantajları	18
2.5.1.1. Ölçeklenebilirlik	18
2.5.1.2. Kullanılabilirlik	18
2.5.1.3. Yönetilebilirlik.....	19
2.5.1.4. Güvenlik	20
2.5.1.5. Hizmet Kalitesi	21
2.6. YÜK DENGELEME TÜRLERİ.....	21
2.6.1. Sunucu Tabanlı Yük Dengeleme.....	21
2.6.2. Switch (Donanım) Tabanlı Yük Dengeleme	22
2.6.3. DNS Tabanlı Yük Dengeleme.....	23
2.7. YÜK DENGELEME MODELLERİ.....	24
2.7.1. Merkezleştirilmiş Model	24
2.7.2. Dağıtık Kontrol Modeli	27
2.8. YÜK DENGELEME POLİTİKALARI.....	27
2.8.1. Lider Politikası	28
2.8.2. Rastgele Politikası	28
2.8.3. NFS Politikası.....	29
2.8.4. Zaman Çizgisi Politikası	30
2.8.5. Eşik Politikası.....	30
2.8.6. Hepsini Birer Kez Dene Politikası	30
2.8.7. LRU Politikası.....	32
2.8.8. En Kısa Kuyruk Politikası.....	33
BÖLÜM 3	36
SANAL SINIF UYGULAMALARI.....	36
3.1. TEKNİK GEREKSİNİMLER.....	42
3.1.1. Ağ	42
3.1.2. Donanım	43
3.1.3. Ortam.....	43
3.2. İLETİŞİM STANDARTLARI	44

3.2.1. Ses Standartları.....	45
3.2.2. Video Standartları.....	46
3.2.3. Kontrol/İletişim Standartları.....	46
3.3. VIDEO KONFERANS STANDARTLARI.....	47
3.4. AĞ ALT YAPISI	47
3.4.1. Band Geniřliđi Gereksinimleri.....	47
3.5. SANAL SINIF UYGULAMARININ KARŐILAŐTIRILMASI	48
3.2. TERCİH EDİLEN SANAL SINIF UYGULAMASI: OPENMEETINGS.....	51
3.3. OPENMEETINGS YAPISI	52
3.3. OPENMEETINGS TERCİH EDİLMESİNİN NEDENLERİ	53
BÖLÜM 4	54
UYGULAMA	54
4.1. GELİŐTİRİLEN YAZILIMIN TEKNİK ALTYAPISI	55
4.1.1. Gerçek Sunucu(ların)nun İőletim Sistemi	56
4.1.2. Sanal Sunucuların İőletim Sistemi.....	56
4.1.3. Öğrenim Yönetim Sistemi Ve Sanal Sınıf Uygulaması	57
4.2. GELİŐTİRİLEN YAZILIMIN PSEUDECODE GÖSTERİMİ	57
4.3. KULLANILAN YÖNTEM.....	58
4.3. GELİŐTİRİLEN YAZILIMIN TANITIMI	61
BÖLÜM 5	66
SONUÇ VE ÖNERİLER	66
KAYNAKLAR	68
EK AÇIKLAMALAR A. UYGULAMANIN KAYNAK KODLARI	70

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1.	İlk 500 süper bilgisayarın arasında kümenin trendi.	6
Şekil 2.2.	Yüksek performanslı küme hesaplamasının yapısı.	7
Şekil 2.3.	İşletim sistemlerinin dağılımı.	9
Şekil 2.4.	Basit yük dengeleme.	17
Şekil 2.5.	DNS tabanlı yük dengeleme ile trafik dağıtımını.	24
Şekil 2.6.	Merkezileştirilmiş kontrol modeli.	25
Şekil 2.7.	Merkezi yük dengeleme için sıra modeli.	25
Şekil 2.8.	Dağıtık kontrol protokolü.	26
Şekil 2.9.	Dağıtık yük dengeleme modeli.	27
Şekil 2.10.	Round robin politikası durum geçiş diagramı.	31
Şekil 2.11.	En kısa kuyruk politikasının durum geçiş şeması.	33
Şekil 3.1.	Bire-çok web konferans.	41
Şekil 3.2.	Çoğa-çok web konferans.	41
Şekil 3.3.	Openmeetings'in yapısı.	52
Şekil 4.1.	Yük dengeleme algoritmasının sonlu durum makinesi ile gösterimi.	54
Şekil 4.2.	Aktif sunucu ayarları penceresi.	62
Şekil 4.3.	Mevcut sunucular için konfigürasyon penceresi.	63
Şekil 4.6.	Yük dağılımı.	65

ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 2.1. İşletim sistemi ailesinin istatistikleri.....	8
Çizelge 2.2. İşletim sistemi istatistikleri.	8
Çizelge 3.1. Popüler iletişim standartları.	44
Çizelge 3.2. Sanal sınıf uygulamalarının karşılaştırılması.	48

SİMGELELER VE KISALTMALAR DİZİNİ

KISALTMALAR

PC	: Personal Computer
LMS	: Learning Management System
DNS	: Domain Name System
FLOPS	: Floating Point Operations Per Second
PANTS	: Application Node Transparency System
RTT	: Round Trip Time
MPP	: Massively Parallel Processing
ASIC	: Application Specific Integrated Circuit
OEM	: Original Equipment Manufacturer
SMP	: Symmetric Multiprocessing
NAT	: Network Address Translation
VIP	: Virtual IP
IM	: Instant Messaging
FDDI	: Fiber Distributed Data Interface
ATM	: Asynchronous Transfer Mode
SCI	: Scalable Cohorent Interface
OGSI	: Open Grid Services Infrastructure
MPI	: Message Passing Interface
HPL	: Human placental lactogen
HPC	: High performance computing
SCSI	: Small Computer System Interface
IDE	: Integrated Drive Electronics
URL	: Uniform Resource Locator
HTTP	: Hypertext Transfer Protocol
FTP	: File Transfer Protocol
IP	: Internet Protocol

RFC 1918 : Address Allocation for Private Internets
OEM : Original Equipment Manufacturer
ASIC : Application Specific Integrated Circuit
Gbps : Gigabit per second
FCFS : First come, first served
LRU : Least Recently Used
NFS : Network File System
NRR : Net Run Rate
BSD : Berkeley Software Distribution
GNU : General Public License
LDCELP : (Düşük Gecikme Kodlu Uyarılmış Doğrusal Tahmin)
PCM : Pulse Code Modulation
LDCELP : Low Delay Code Excited Lineer Prediction
PSTN : Public Switched Telephone Network
POTS : Plain Old Telephone Service
ISDN : Integrated Services Digital Network
Annex A/B : Asymmetric Digital Subscriber Line (ITU G.992.1 Annex A/B)
CIF : Common International Format
FCIF : Full Common Intermediate Format
QCIF : Quarter CIF
DCT : Discrete Cosine Transform
SubQCIF : Sub Quarter CIF

BÖLÜM 1

GİRİŞ

Uzaktan eğitimin popülaritesi gittikçe artmaktadır. Mekândan ve kısmen zamandan bağımsız eğitim alma imkânı sunan uzaktan eğitimin asenkron içeriklerinin yanında eşzamanlı içeriklerle de desteklenme ihtiyacından dolayı sanal sınıf uygulamaları doğmuştur. Böylece öğretim elemanları ile öğrenciler arasındaki mesafe sanal sınıflar sayesinde kaldırılmaktadır. Sanal sınıf uygulamaları eşzamanlı olarak farklı yerlerden derse katılan öğretim elemanı ve öğrenciler için kamera, ses, bilgisayar ekranı, bilgisayar uygulaması, doküman vb. veya bunlardan birkaçını paylaşarak gerçek sınıf ortamıyla farklılığı en aza indirmeye çalışır.

Bu imkânları sunmak için çok ciddi bir altyapıya gereksinim olduğu aşikârdır. Eğitim veren kurum planlamasında öğrenci sayısına göre sanal sınıf uygulamasını sunacağı altyapıyı teşekkül ettirmelidir.

Uzaktan eğitim sistemleri genelde öğrenim yönetim sistemi, sanal sınıf uygulaması ve veri tabanı için birer sunucu kullanılarak kurulmaktadır. Her bir uygulama için tek ve bağımsız bir sunucu çalıştırmak tercih edilmektedir. Tek ve bağımsız bir sunucu ile hizmet veren bir sanal sınıf uygulaması ile birkaç sene içerisinde öğrencilere yeterli hizmetin verilemeyeceği öngörüldüğü için performans artırımı yapılmasına gerek görülmüştür. Çalışmada, piyasadaki mevcut ticari ve açık kaynak kodlu sanal sınıf uygulamaları incelenmiş, aralarındaki fiyat/performans karşılaştırması ve mevcut kullanılan donanım üzerinde yapılabilecek değişiklikler değerlendirilmiştir.

İkinci bölümde küme kavramı, küme kavramının avantajları ve dezavantajları tartışılmıştır. Çeşitleri ve kullanım amaçlarından bahsedilmiştir. Uygulama bölümünde geliştirilen çözüme esas teşkil eden temel bilgiler burada ifade edilmiştir. Kümeleme mantığının bir sonucu olarak ortaya çıkmış olan ve uygulamanın temelini

oluřturan yk dengeleme kavramı tanıtılmıřtır. Yk dengelemenin kmeye ve hizmet kalitesine sađladıđı faydalar sunulmuřtur.

çnc blmde sanal sınıf uygulamaları hakkında bilgiler verilmiřtir. Genel olarak tanım, ticari ve aık kaynak kodlu yazılımların incelenmesi ve karřılařtırılması sunulmuřtur. İhtiya duyulan donanımlardan ve performans artıřı iin muhtemel zmlerden kısaca bahsedilmiřtir. Bu nerilerden mevcut yapıya gre en etkin bulunan yntem gerekleřtirilip drdnc blmde sunulmuřtur

Uygulama blmnde, aık kaynak kodlu sanal sınıf uygulamasında performans artırımını amacıyla tasarlanan yapı, geliřtirilen kod ve gerekleřtirilmesi aktarılmıřtır.

Sonuç ve neriler blmnde ise geliřtirilen zmn sisteme sađladıđı maliyet ve performans kazanç oranı, faydaları, etkinliđi, fiyat/performans aısından diđer yntemlerle kıyaslanması deđerlendirilmiřtir. Ayrıca neri olarak da geliřtirilmeye aık ynleri sıralanmıřtır.

BÖLÜM 2

KÜME

Küme, bağımsız bilgisayarların yazılım ve ağ yoluyla tümleşik sistemde birleştirilmesi anlamına gelen ve çok kullanılan bir terimdir. En temel düzeyde, iki ya da daha fazla bilgisayar bir problemi çözmek için kullanıldığı zaman, bu küme olarak nitelendirilmektedir. Kümeler tipik olarak yüksek kullanılabilirlik, büyük güvenilebilirlik veya bir bilgisayarın sağlayabileceğinden daha büyük hesaplama gücü sağlayan yüksek performanslı hesaplama için kullanılmaktadır [1].

Bir başka deyişle küme, ticari bileşenlerden yapılandırılmış ve ticari yazılımlar çalıştıran paralel bilgisayardan oluşmaktadır. Bir küme, her biri bir ya da daha fazla işlemci, kümedeki (ve sadece kümedeki) bütün işlemciler tarafından paylaşılan bellek ve düğümler arasında veri taşınmasına izin veren ağa bağlanmış ilave çevresel aygıtlar (diskler gibi) içeren düğümlerden oluşmaktadır [2].

Küme; güvenilirliği, kullanılabilirliği ve/veya performansı arttırmak için birbirine bağlanmış; düğümlerin (bilgisayarların) gruplanması şeklinde de tanımlanabilmektedir. Kümeye ilk bakışta salt hesaplama düğümüne odaklanılmasına rağmen, tam olarak küme; bağlı ağlar, donanım yönetimi, yönetim yazılımı, depolama ve uygulama yazılımı gibi çeşitli düğüm türlerinden oluşmuştur. Her düğüm kısmen farklı olmak zorunda olmasına rağmen, bu bileşenlerin her biri sistemin faydalı bir çözüm gibi işlev görmesini sağlamak için kapsanması gerekmektedir [3].

2.1. TARİHÇESİ

Bilgisayar kullanımındaki anahtar sebep işlerin hızını arttırmaktır. Araştırmacılar daha geniş ve daha kompleks problemleri çözmek için her zaman daha güçlü bilgisa-

yarları arzu etmektedir. Daha çok hesaplama gücüne ulaşmak için, iş yükünü işlemciler arasında dağıtan paralel hesaplama 1960'lardan beri revaçtır. 1960'lardan 1990'lara kadar yaygın olan, genellikle süper bilgisayar denilen, çoğu paralel makineler, aslında vektörel (paralel) bilgisayardı ve çok büyük ölçekte paralel sistemdi (Massively Parallel Processing, MPP). Bu sistemler aşırı derecede pahalı olduklarından sadece belirli organizasyonlarda kullanılabilmişlerdir.

Bilişim teknolojileri çok hızlı değişim göstermektedir. Artık "süper hesaplama" terimi daha az sıklıkta kullanılmaktadır ve 1990'lardan beri daha geniş olarak, daha genel bir terim olan "Yüksek Performanslı Hesaplama" kullanılmaktadır. Yüksek performanslı hesaplama terimi, hesaplama yeteneği mevcut en yüksek kalitede kişisel bilgisayarlardan ve iş istasyonlarından daha güçlü (hesaplama gücü büyüklük sıralamasında en az ikinci) olan makine tiplerinin çeşitliğine işaret etmektedir. Bugün "yüksek performanslı hesaplama" Teraflops hesaplama aralığında, hesaplama yeteneği anlamına gelmektedir [4].

1980'lerde bilgisayarın performansının en çok, daha hızlı ve daha etkili işlemciler üreterek arttırılacağına inanılmıştı. Bu fikre esasında bazı problemleri beraber çözmek için iki veya daha çok bilgisayarı birbirine bağlamak demek olan paralel işleme tarafından meydan okunmuştur. 1990'ların başından itibaren pahalı ve özelleşmiş uygun paralel süper bilgisayarlardan iş istasyonlarının ağına doğru artan düşünce değişikliği eğilimi olmuştur. Bu geçişi mümkün kılan itici güçler arasında iş istasyonları ve ağlar için çok hızlı bir gelişme ve ticari yüksek performanslı bileşenlerin kullanılabilirliği bulunmaktadır. Bahsedilen teknolojiler bilgisayar (Kişisel bilgisayarlar veya iş istasyonları) ağlarını paralel işleme için cazip bir araç yapmaktadır ve bu durum dolayısıyla düşük maliyetli ticari süper hesaplama yol açmaktadır [5].

Bilgisayarın performansını arttırmak için, aşikâr yol işlemcilerin, sistem veri yollarının, bellek ve diğer bileşenlerin işlem hızlarını arttırmaktır. Tarihi sürece bakıldığında, maksimum işlemci hızı her on sekiz ayda yaklaşık olarak iki katına çıkmaktadır. Bu üssel artış yüksek performanslı hesaplamanın ana baskı unsurlarından biri olarak görülmektedir. Fenomen göreceli olarak uzun süre devam

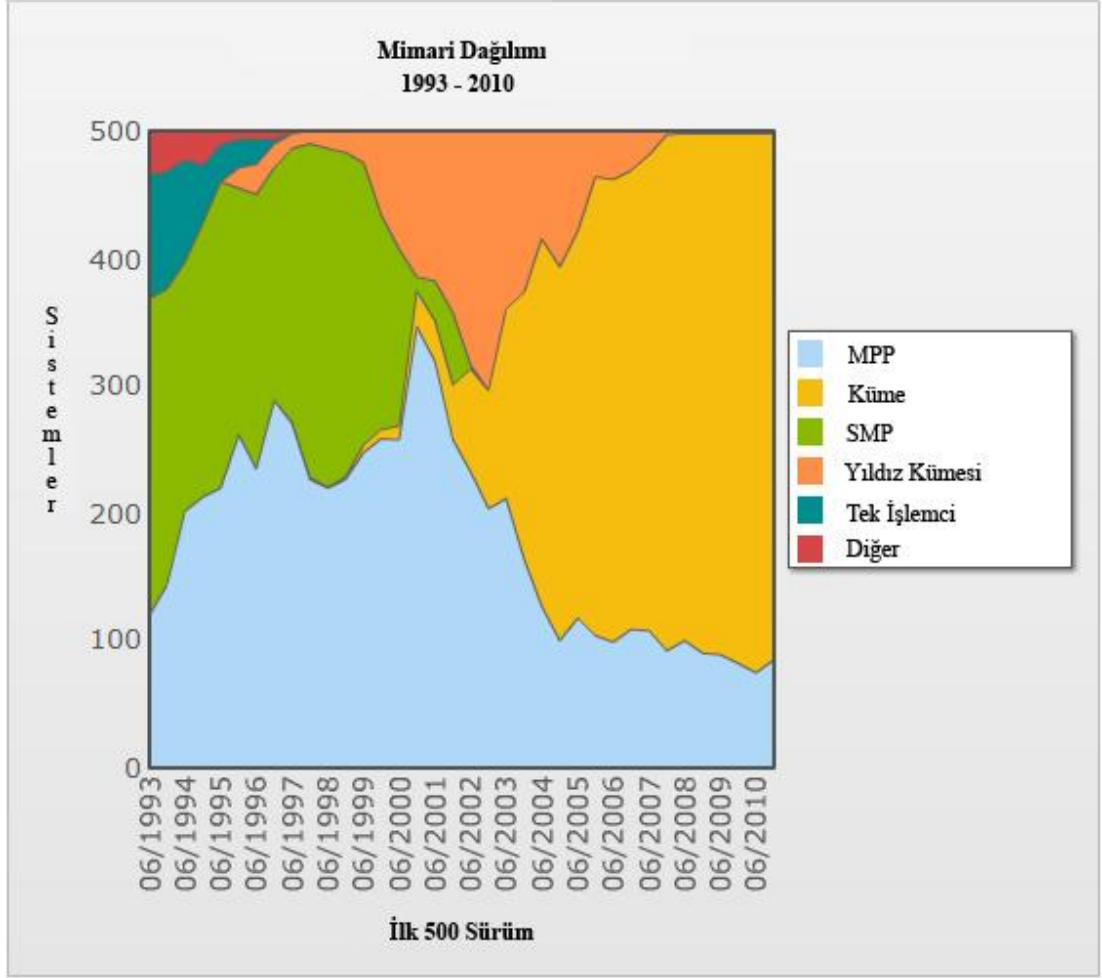
edebilir olduđu halde, gelecekteki geliřmelerin ışık hızı ile sınırlı olduđu görölmektedir [4].

Performansı arttırmanın bir başka yolu çoklu işlemcileri birlikte çalıştırmaktır ve onların hesaplama gücünü birleřtirmektir. Bu yaklaşım paralel hesaplama veya paralel işleme olarak nitelendirilmiştir. Geçtiğimiz kırk yılda, birbiriyle bağdařmayan birçok paralel mimari önerilmiştir. Bu paralel makineler dikkatli bir yapı gerektirdiğinden ve uzmanlaşmış işletim sistemi ve yazılım araçları çalıştırdığından, pahalıya mal olan yapılanmaya ve idameye eğilim göstermektedir.

1990'ların ortalarından itibaren ağ teknolojilerindeki geliřmeler ile küme hesaplaması gittikçe yaygın hale gelmiştir. Bir küme hesaplama sistemi yüksek performanslı ağ ile bağlanmış birkaç bağımsız bilgisayar ile oluşturulmuştur. Kümedeki bilgisayarlar genel amaçlı, hazır kişisel bilgisayarlar, iş istasyonları veya küçük simetrik çok çekirdekli sistemler olabilmektedir. Ağ sistemi bu bilgisayarları birbirine bağlamak için yüksek hızlı, düşük gecikmeli iletişim sağlamaya ihtiyaç duymaktadır. Bu iletişim Fast Ethernet, FDDI, ATM, SCI, Gigabit Ethernet veya Quadrics Ağı ve Myrinet gibi bazı firmaya özel teknolojiler olabilmektedir.

Küme hesaplaması ve ilişkili teknolojiler olgunlařtıkça, yüksek performanslı hesaplama sistemleri bu paradigmayı kullanarak yapılandırılmıştır. İlk 500 süper bilgisayarların trendine bakıldığında Şekil 2.1'de gösterilen bu geliřim görölebilmektedir.

Şekil 2.1 top500.org tarafından incelenmiş dünyadaki ilk 500 süper bilgisayarın mimari istatistiklerini listelemektedir. 1997'de Berkeley NOW ilk beş yüz listesinde ilk görölen küme olmuştur. Bugün ise tamamen kümelerin hâkim olduđu görölmektedir. Güncel listedeki IBM Blue Gene gibi MPP'lerin çoğu aslında mesaj aktarma sistemleridir.



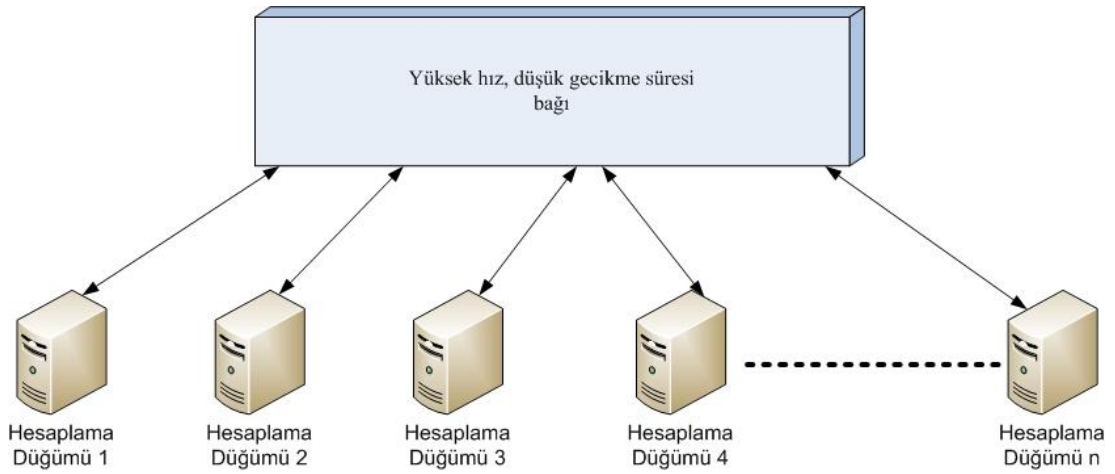
Şekil 2.1. İlk 500 süper bilgisayarın arasında kümenin trendi [6].

Yüksek performanslı küme hesaplamasının avantajları şöyle özetlenebilir:

- 1- Maliyet etkinlik: Kümeler ticari cihazlardan yapılandırılabilirler ve ticari yazılımlar çalıştırılır.
- 2- Esneklik: Hemen hemen her kişisel bilgisayar, iş istasyonu veya simetrik çok çekirdekli sistemde kümelenebilir.
- 3- Genişletilebilirlik: Daha çok hesaplama gücüne ihtiyaç duyulduğunda, çok zorlanmadan sisteme daha fazla düğüm eklenebilir
- 4- Kolay yönetilebilirlik: Sistemi genel, hazır görüntüleme araçları kullanarak kontrol etmek; özelleşmiş, patentli görüntüleme araçlarını kullanarak kontrol etmekten daha kolaydır. Dahası, bir tek düğümün çökmesi bütün sistemi etkilemeyecektir ve çöken düğümü sistemin kalanını kesintiye uğratmadan tamir etmek mümkündür.

- 5- Kolay entegre edilebilirlik: Bir kümeyi, standartlaştırılmış OGSİ ara yüzünü kullanarak bir global dağıtık Grid hesaplama ortamıyla birleştirmek kolaydır.

Şekil 2.2 bir yüksek performanslı kümenin temel yapısını göstermektedir. Küme sistemlerinde, temel olarak bütün düğümler yerel dosya sistemine birer işletim sistemi kurulmuş bağımsız bilgisayardır. Kümede paralel hesaplamayı desteklemek için, örneğin MPI'ın uygulaması gibi ilave kütüphaneler gerekebilmektedir.



Şekil 2.2. Yüksek performanslı küme hesaplamasının yapısı [4].

Yüksek performanslı hesaplama dünyasında hâlihazırdaki işletim sistemlerine Unix tabanlı sistemler tarafından hükmedilmektedir. İlk beş yüz sistemin işletim sistemi ailelerinin istatistikleri Çizelge 2.1'de, işletim sistemlerinin istatistikleri Çizelge 2.2'de ve işletim sistemleri dağılımı Şekil 2.3'de sunulmuştur. Yüksek performanslı hesaplama dünyasında hâlihazırdaki işletim sistemlerine 2004 Uluslararası Süper Bilgisayar Konferansı'na göre, ilk 500 sistemden işletim sistemlerinin çoğunluğunun yaklaşık yüzde 55'i Linux, yüzde 40'ı diğer Unix sistemler ve yüzde 5'den azı Windows platformudur [4].

İlk 500 tablosunda, Rmax HPL donanımsal karşılaştırmasında erişilmiş maksimum bilgisayar performansını (Gflops ile ölçülmüş) ifade etmektedir.

Rpeak teorik uç performansı ifade etmektedir. Belirli bir periyotta, genellikle makinenin devir zamanında tamamlanabilen kayan noktalı toplamaların ve çarpımların sayısını sayarak belirlenmiştir.

Çizelge 2.1. İşletim sistemi ailesinin istatistikleri [7].

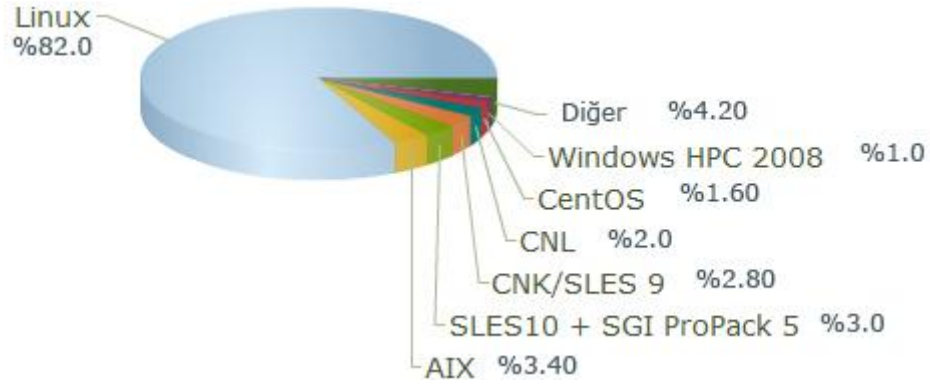
İşletim Sistemi Ailesi	Sayı	% Oranı	Rmax Toplamı (GF)	Rpeak Toplamı (GF)	İşlemci Toplamı
Linux	459	% 91.80	38587202	58399775	5146933
Windows	5	% 1.00	412590	509350	59072
Unix	19	% 3.80	1399566	1796003	109970
BSD Based	1	% 0.20	122400	131072	1280
Mixed	16	% 3.20	3151334	3819110	1155072
Totals	500	% 100	43673092.54	64655310.70	6472327

Çizelge 2.2. İşletim sistemi istatistikleri [8].

İşletim Sistemi	Sayı	% Oranı	Rmax Toplamı (GF)	Rpeak Toplamı (GF)	İşlemci Toplamı
Linux	410	82.00 %	33646164	52386289	4542764
AIX	17	3.40 %	1253486	1635885	94288
SLES10 + SGI ProPack 5	15	3.00 %	1331023	1503818	135200
CNK/SLES 9	14	2.80 %	3021054	3663462	1134592
CNL	10	2.00 %	1271718	1630601	178577
CentOS	8	1.60 %	1099080	1326517	114792
Windows HPC 2008	5	1.00 %	412590	509350	59072
SuSE Linux Enterprise Server 9	4	0.80 %	258017	393332	59504
Redhat Linux	4	0.80 %	361590	446020	48800
RedHat Enterprise 4	3	0.60 %	109580	151341	14736
SUSE Linux Enterprise Server 10	2	0.40 %	106110	113779	14328
RedHat Enterprise 5	2	0.40 %	129120	139795	11928
Super-UX	1	0.20 %	122400	131072	1280
Cell OS	1	0.20 %	35480	38836	3650
SUSE Linux	1	0.20 %	274800	308283	26304
UNICOS/SUSE Linux	1	0.20 %	35200	42598	8192
UNICOS/lc	1	0.20 %	95080	113050	12288
Open Solaris	1	0.20 %	110600	121282	12032
Totals	500	100%	43673092.54	64655310.70	6472327

Bu arada ilk 500' deki sistemlerin yüzde sekseni Intel işlemciler kullanmaktadır. Bu fenomen, küme hesaplamaya hakim trendle örtüşmektedir. Kişisel bilgisayar marketinde Wintel'in (Windows+Intel) başarısı gibi, Lintel (Linux+Intel)

bilgisayarlar yüksek performanslı hesaplama dünyasında önemli rol oynamaktadır. İnsanlar aynı performansa sahipken, daha düşük maliyetli ve daha uyumlu bir çözümü tercih etmektedir.



Şekil 2.3. İşletim sistemlerinin dağılımı [9].

Ticari yazılım ve donanımlı, Linux tabanlı kümeler çoğu kez Beowulf sistemler olarak tanımlanmaktadır. Linux açık kaynak kodlu olduğundan ve gittikçe stabil hale geldiği ve güçlendiği için, geçmişte yüksek performanslı hesaplama kümelerinde hakim işletim sistemi olması beklenmekteydi. Endüstri devleri bile, kendi işletim sistemlerini bir kenara bırakıp, yüksek performanslı hesaplama sistemleri için iştihakla Linux çözümleri sağlamışlardır. Örneğin IBM ilk 500 listesinde 149'u Linux kümeleri olan 226 süper bilgisayar sistemine sahiptir [4]. Şuanda %91,80 hâkimiyet oranıyla işletim sistemi öngörüsünün ne kadar haklı olduğu ortadadır.

Kümelerin yapılandırılmasında düğümlerde kullanılan işlemci ve işletim sistemleri hakkında verilen bilgilerle birlikte, yapılandırma çeşidine göre kümeler şöyle sınıflandırılabilir:

- 1- Tahsis edilmiş kümeler,
- 2- Tahsis edilmemiş kümeler.

Bu iki olgu arasındaki ayırım, kümedeki iş istasyonlarının sahipliğine dayanmaktadır. Tahsis edilmiş kümeler olgusunda, bir özel birey kendi iş istasyonuna sahip değildir ve kaynaklar paylaşılmaktadır; böylece paralel hesaplama bütün küme üzerinde gerçekleştirilir. Tahsis edilmemiş küme alternatifi ise, şahısların kendi iş

istasyonlarına sahip olduđu ve bu olguda uygulamaların bořta iřlemci çeviriminden çalarak çalıřtırıldıđı yerdir. Bunun gerekçesi, en yođun anda bile, çođu iř istasyonu iřlemci çeviriminin kullanılmıyor olması gerçeđine dayanmasıdır. Tahsis edilmemiř iř istasyonlarının dinamik deđiřen kümesi üzerindeki paralel hesaplama, uyarlanabilir paralel hesaplama da denir.

Tahsis edilmemiř iř istasyonlarının kullanıldıđı yerde, iř istasyonun sahipleri ve iř istasyonuna uygulamalarını çalıřtırmak için ihtiyaç duyan uzak kullanıcılar arasında bir çekiřme mevcuttur. Öncekiler kendi iř istasyonlarından hızlı etkileřimli cevap beklerlerken, sonrakiler sadece herhangi boř CPU çevirimi kullanan hızlı uygulama dönüř süresiyle ilgilenmektedir. İřlem kaynaklarını paylařmadaki bu vurgu düđüm sahipliđi kavramlarını ortadan kaldırırken; iřlem taşıma ve yük dengeleme stratejileri gibi karmařıklıklar için ihtiyaç ortaya çıkarmaktadır. Bu gibi stratejiler kümelere yeterli etkileřimli performans dađıtmaya izin verdiđi kadar talep edilen seri ve paralel uygulamalara paylařılmıř kaynakları sunmaya da izin vermektedir.

Açıkça, iř istasyon ortamı, iletiřim yođun olmayan, yüksek mesaj bařlatma gecikmeleri ve düřük band geniřliđi görülebilen uygulamalara daha uygundur. Eđer bir uygulama daha yüksek iletiřim performansı gerektiriyorsa, Ethernet gibi, mevcut olan LAN mimarileri, bunu sađlamaya yeterli olmamaktadır.

Bilimde ve endüstride geleneksel olarak, bir iř istasyonu, UNIX platformunun bazı sınıflarına; yönetimsel iřler ve kelime iřleme için olan kiřisel bilgisayar tabanlı makinelerin baskın fonksiyonuna iřaret etmektedir. 1995’li yıllardan beri iřlemci performansında ve UNIX iř istasyonlarının ve kiřisel bilgisayar tabanlı makinelerin çekirdek-düzey fonksiyonelliklerinde çok hızlı bir yakınsama olmuřtur, bu yüksek performanslı Pentium tabanlı makinelerin ve Windows NT iřletim sisteminin giriřiyle iliřkilendirilebilir. Bu yakınsama hesaplama kaynaklarının bazı formları gibi, paralel hesaplama için, kiřisel bilgisayar tabanlı sistemler kullanmada artan bir ilgi bařlatmıřtır. Bu faktör kiřisel bilgisayarların kısmen düřük maliyeti ve ilk amacı bazı iřbirliđi yolları ile bu kaynakları kontrol altına almak olan birkaç yazılım projesini bařlatan akademik çevre ve endüstrinin her ikisinde de yaygın kullanılabilirliđi ile bađlantılıdır [5].

2.2. KÜME TÜRLERİ

Farklı kullanımlarına rağmen, kümeler üç genel türe ayrılır:

- 1- Yük dengeleme (Load Balancing)
- 2- Yüksek kullanılabilirlik (High Availability)
- 3- Yüksek performanslı hesaplama (High Performance Computing)

Bir kapsamlı küme çözümünde, küme türlerinin her üçü de bulunabilir. Örneğin bir özel çevre çözüme erişimi sağlamak için Web sunucuların yük dengeleme kümesini kullanmalıdır. Sistem, yüksek kullanılabilirlik kümesinde kullanıcı kimlik denetimini sağlamak için iki sunucu kullanmalıdır. Son olarak, çözümün yüksek performans hesaplama kümesi kullanıcının hesaplamalarını yerine getirmek için kullanılmalıdır.

2.2.1. Yük Dengeleme

Yük dengeleme kümeleri, iş yükünün tek bir kaynağı bunalttığı yerde, iş yükünü özdeş kaynaklara dağıtmak için kullanılır. Yük bu kaynaklar arasında ya hepsini birer kez dene algoritması veya daha detaylı ve uyarlanabilir algoritmalar ile dağıtılabılır. Yük dengeleme kümesinin genel örneği Web sunucuların bir kümesidir.

2.2.2. Yüksek Kullanılabilirlik

Daha önce bahsedildiği gibi, kümeler artıklık kullanımı yüzünden tek sunucudan daha fazla arttırılmış güvenilirlik sağlamak için kullanılabilir. Böyle küme tiplerine yüksek kullanılabilirlikli veya yük devretme kümeler denir. Yüksek kullanılabilirlikli kümelerin iki standart tipi vardır.

2.2.2.1. Aktif-Pasif

Bir aktif-pasif yüksek kullanılabilirlikli kümede, arta kalan sunucular boşa dururken bir sunucu aktif olarak hizmeti sunar. Eğer aktif sunucu çökerse, yüksek

kullanılabilirlik yazılımı pasif sunuculardan birisini aktif duruma getirir ve gelecek hizmet taleplerini yeni aktif sunucuya transfer eder.

2.2.2.2. Aktif- Aktif

Bir aktif-aktif yüksek kullanılabilirlikli kümede, hizmeti aktif olarak kümedeki bütün bilgisayarlar sunar. Eğer bir aktif sunucu çökerse, gelecek hizmet talepleri bir başka aktif sunucuya yönlendirilir. Bir aktif-aktif yüksek kullanılabilirlikli kümede, dikkate değer olan hizmet taleplerinin yüksek kullanılabilirlikli küme düğümleri arasında nasıl dağıtılacağıdır.

2.2.3. Yüksek Performanslı Hesaplama

Birçok hesaplama problemi bir tek hesaplama kaynağı kullanılarak zamanında çözülemez. Bunun sonucu olarak, hesaplama düğümleri, daha büyük hesaplama problemlerini adresleyebilen bir birleştirilmiş sistem sağlamak için bir araya gruplandılar. HPC kümeleri, örneğin, insan genomunu açmak için, bir bölgede hava tahminleri üretmek için, petrol rezervinin büyüklüğünü ve durumunu modellemek için veya bir çizgi filmde pikselleri işlemek için gereklidir. Yüksek performanslı hesaplama sistemindeki her bir düğüm ya komşu düğüme sıkı sıkıya bağlıdır veya bağımsız hareket eder. HPC kümeleri için yeni faydalar durmaksızın tanımlanır ve sistemleri büyüklüğü, kompleksliği arttırmak için zorlar. 1998'de köşe teknolojisini (origin- edge) kestiği kabul edilen dört teraflop sistemlerin, hesaplama endüstrisi bir milyon teraflops hesaplamaya ilerlediği için artık değersiz olduğu görülmektedir. Sonuç olarak, enerji kısıtlarına ve bu HPC sistemlerinin yönetimine özenle önem verilmesi gerekmektedir [3].

2.3. BEOWULF

Beowulf kümeleri bir özel sistem ağında açık kaynak kodlu yazılım (Linux) altyapısı ile ticari donanım temelli ölçeklenebilir performans kümesidir. Tasarımcı makine ekleyerek performansı nispeten arttırabilmektedir. Beowulf kümesi oluşturmak için ticari donanım, piyasadaki ürünlerden herhangi biri; bağımsız hesaplama düğümleri

mümkün olduğunca basit olarak her biri Linux çalıştıran ve bir dosya sistemi paylaşan, ağa bağlanmış iki bilgisayar veya olabildiğince kompleks yüksek hızlı, düşük gecikmeli ağ ile bağlı 1024 düğüm kullanılabilir.

Beowulf basitçe kümeleri genel amaçlı standart donanımlar ile oluşturulan Sınıf I kümeler; özel amaçlı donanımlar ile oluşturulan Sınıf II kümeler olmak üzere iki sınıfa ayırmıştır. Sınıf I kümeler tamamen ticari donanım ve SCSI, Ethernet ve IDE gibi standart teknoloji kullanan yazılım kullanarak yapılandırılmıştır. Tipik olarak daha yüksek performansa ulaşmak için özelleşmiş donanım kullanan Sınıf II kümelerden daha hesaplı olmaktadır.

Genel kullanım alanları simülasyonlar, biyoteknoloji ve petro-kümeleme; finansal market modelleme, veri madenciliği ve akış işleme ve ses ve oyunlar için internet sunucular gibi geleneksel teknik uygulamalardır.

Beowulf programları genelde C ve Fortran gibi diller kullanılarak yazılır. Paralel hesaplamayı gerçekleştirmek için mesaj göndermeyi kullanırlar [1].

2.4. KÜME BİLEŞENLERİ

Kümeler incelenirken bütün dikkatler hesaplama sunucularında toplanmışken, kümeleri gerçekten faydalı unsurlar yapmak için gerekli birkaç bileşen bulunmaktadır.

2.4.1. Düğümler

Bir HPC kümesinin düğümleri kümenin merkezindedir. Onlar sisteme erişmeyi, sistem tarafından yönetilen platformu ve zor hesaplama problemleri için beygir gücünü sağlamaktadır. Bu düğümler tek işlemcili; Linux, Windows, AIX veya diğer işletim sistemlerinin herhangi bir kombinasyonunu çalıştıran simetrik çok işlemcili (SMP) sunucular olabilir. Aslında, hesaplama düğüm mimarilerinin bir heterojen karışımı ile HPC kümeleri bulmak oldukça yaygındır.

2.4.2. Ara Baęlantılar

Bir kümede, aęlar sistemi bir arada tutmayı saęlar. Aęlar dıřarıdan sisteme eriřim, hesaplama sunucularına kendi aralarında iletiřim kurmaları için bir metot ve sistemler kapalı olsa bile yöneticinin erişebileceęi bir yol sağlamaktadır.

2.4.3. Hafıza

Genellikle gözden kaçmasına rağmen, kümenin hafıza sisteminin kümeye işlem yapaması için en temel unsurdur. Merkezileřmiř hafıza bileřsensiz hesaplama düęümü işlem yapacak veya öncelikle onların işlemleri bitmeden verilerini kaydedecek hiçbir yere sahip olmayacaktır. Eęer hafıza altyapısının altında yatan dizayna gereken önem verilmezse, en güçlü HPC sistemi bile onların giriş çıkıř taleplerinin tamamlanmasını bekleyen atıl hesaplama kaynaklarına dönüşecektir.

Küme hafıza sisteminin tartıřması mevcut hafıza donanımından disklerin en tepesinde oturan dosya sistemine genişletilmelidir. Küme mimarisi performansa, uygulanabilirliğe ve HPC kümelerinin taleplerini genişletilebilme talepleriyle buluřturmak için uygun özel yazılım (middleware) tanımlamaya odaklanması gerekmektedir. Bu hususları dikkate alan bir küme dosya sisteminin en mükemmel örneęi IBM General Parallel File System (GPFS™) dir.

2.4.4. Donanım Yönetimi

Bir kümenin donanım yönetim kabiliyetleri donanım kontrol araçlarını içerir. Bu araçlar kullanıcıya çökme hataları için donanımı izleme; sunucuların gücünü açma ve kapatma; sunuculara uzaktan konsolla erişimi saęlama izni vermektedir.

2.4.5. Yönetim Yazılımı

HPC sistemlerin büyüklüęü ve komplekslięi arttıka, güçlü yönetim yazılımlarına olan ihtiyaç da artmıřtır. Yöneticilerin işletim sistemini her küme düęümüne bir CD ile kurabildięi veya bütün düęümlerin aygıt yazılımını bir disket aracılıęıyla

güncelleyebildiği günler geçmişte kalmıştır. Şimdi yöneticiler birkaç dakikada binlerce düğüme kurulum yapan, bütün düğümlerinde aynı anda paralel komutlar çalıştıran ve bütün düğümlerde paralel olarak aygıt yazılımı güncelleyen yönetim yazılımına sahiptir.

Ayrıca küme yönetim yazılımı başlığında IBM Tivoli® Workload Scheduler LoadLeveler® gibi toplu iş planlama yazılımı da kapsamıştır. Bu yazılım kullanıcılara kümeye bir iş kabul etmeye ve uygun kaynaklarda acilen yürütülsün diye sistem kaynaklarında dağıtmaya veya sistem kaynak sırasındaki diğer işler tamamlanır tamamlanmaz yürütmeye izin vermektedir. Açık kaynak kodlu Ganglia izleme takımı gibi araçlar yöneticilere HPC kümelerinin kaynaklarının nasıl kullanıldığını izlemeye imkânı vermektedir.

2.4.6. Uygulama Yazılımı

Bir kümenin uygulama yazılımı hesaplama düğümleri üzerinde çalışan teknik veya bilimsel koddur. HPC uygulama yazılımına örnekler hava tahmin modelleri, gen montaj sıralaması, finansal model tahmini veya animasyon piksel sayısal işlemedir [3].

2.5. YÜK DENGELEME

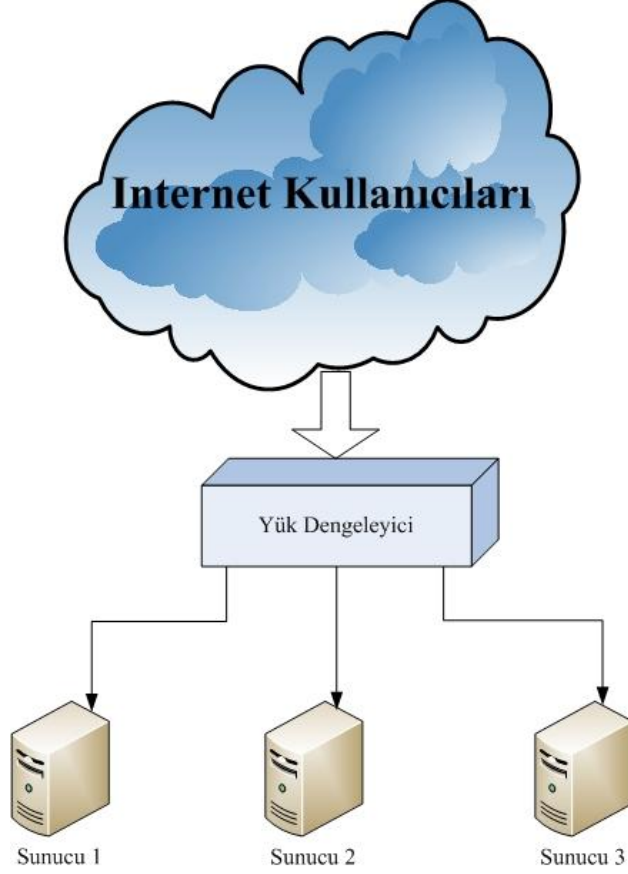
Cevap verme zamanını kabul edilebilir limitlerde tutmak için, Web sunucu yöneticileri genellikle eş zamanlı olarak açık olan bağlantı sayısını sınırlandırır. Talep sayısı bu limiti aştığında, bundan sonraki bütün kullanıcı talepleri reddedilir. Bu yüzden genelde ya bekleme süresi yüksektir ya da kullanıcı talebi tamamlanmamıştır. Her iki durum hoşnutsuz ziyaretçiler olmasına yol açabilir ve çoğunun siteden uzaklaşması muhtemeldir. İşleri daha karmaşık hale getirmek için, trafik günün saati, haftanın günü veya yılın ayına göre bile değişebilir. Bu sezonsal veya periyodik dalgalanmalar bile bir site için uygun kapasite planlamayı daha zorlaştırır.

Bu problem için, daha güçlü sunucu eklemek, ön bellekleme veya dış kaynak kullanımı gibi çeşitli mümkün çözümler bulunmaktadır. Fakat yük yönetimi metotlarının hepsinin kendine göre problemleri mevcuttur. (Örneğin, daha güçlü sunucu bir süre için istenilen şekilde çalışabilir, fakat bu ölçeklendirilebilir olmayacaktır ve sunucu yükseltmek ve bakımı için kesintiye neden olabilecektir.) Ayrıca, eğer sunucu kapasitesi en tepe noktada yük tabanlı planlanmış ise, yoğun olmayan saatlerde ekstra kapasite israf edilmiş olacaktır. Diğer taraftan dış kaynak tüketiminin yüksek bedelli bir etiketi vardır ve buna rağmen hizmet kalitesinde limitli kontrole sahip olabilmektedir. Ön bellekler, aşırı yüklenmiş sunuculardan kaynaklanan yavaş cevaplama zamanı problemini ele almak için kullanılmaktadır. Web sunucu ve tarayıcılar arasındaki yol boyunca bir yerde duran ön bellek, web içeriği için talepleri keser ve müsait olduğunda taleplere cevap vermeye çalışır. Bu taleplere ön bellekten hizmet sağlanmadığında, Web sunucuya yönlendirilirler. Çoğu web sitede sunulan dinamik içerik karşısında ön bellekleme engelleme artmaktadır.

Gerçekçi ve etkili bir yol, kümelenmiş çoklu web sunucularını ve bu sunucular arasında yük dengelemeyi kullanmaktır. Daha büyük mesafeden kaynaklanan ağ gecikmesini adreslemek için birçok organizasyon çoklu web sunucularını bambaşka şehirlere, eyaletlere hatta ülkelere konuşturmuştur. Bir sunucuya kullanıcı hizmet talepleri bazı yönlendirme algoritmaları baz alınarak yönlendirme yapılmaktadır. Hayati olarak sistem performansı bu yönlendirme algoritmalarına bağlıdır. Yük yönetiminin bu metodu pratikteki hizmet kalitesini arttırmak için gösterilmiştir ve yaygın olarak kullanılmaktadır. Sunucu kapasitesi için çok hassas planlara ihtiyaç duymaması, anlık durumda ya yeni bir sunucu satın alarak ya da başka yerden kullanılmayan kapasiteyi çalıştırarak mevcut olan kapasiteye ekleme yapabilmesi çoklu sunucu kullanmanın büyük avantajıdır [10].

Yük dengelemeyi tanımlamak gerekirse, ağ tabanlı araçları kullanarak site trafiğini birkaç sunucu arasında dağıtan işlem ve teknoloji olarak tanımlamak mümkündür. Bu araçlar bir site için sevk edilmiş trafiği engellemekte ve bu trafiği çeşitli sunuculara yönlendirmektedir. Son kullanıcı için yük dengeleme işlemi tamamen saydamdır. Bir tek URL'nin arkasında genelde düzinelerce veya yüzlerce sunucu

iřletimi olabilmektedir. Őekil 3.1’de yk dengelemenin en basit gsterimini grmek mmkndr.



Őekil 2.4. Basit yk dengeleme [11].

Bir yk dengeleyici Őu fonksiyonları yerine getirmektedir:

- 1- Bir siteye ynelmiř ađ tabanlı trafiđi (web trafiđi gibi) nler.
- 2- Trafiđi birbirinden ayrı taleplere bler ve hangi sunucunun ayrıık taleplere cevap vereceđine karar verir.
- 3- Kullanılabilir sunucular iin bir gzetim sađlar, trafiđe cevap verdiklerinden emin olur. Eđer kullanılabilir deđillerse, rotasyondan hari tutar.
- 4- Birden fazla nitenin devre dıřı kaldıđı senaryo iin fazlalık (yedek) sađlar.
- 5- URL’leri okuma, cookie’leri nleme ve XML dnřm gibi Őeyler yaparak ierik farkında dađılım sađlar [11].

2.5.1. Yük Dengeleme Avantajları

Yük dengeleyici ile dağıtımın birçok faydası vardır. Bunlardan birkaçını şöyle sıralamak mümkündür:

- 1- Ölçeklenebilirlik (Scalability)
- 2- Kullanılabilirlik (Availability)
- 3- Yönetilebilirlik (Manageability)
- 4- Güvenlik (Security)
- 5- Hizmet Kalitesi (Quality Of Service)

2.5.1.1. Ölçeklenebilirlik

Yük dengeleyiciler kullanıcı taleplerini uygun olan gerçek uygun sunuculara dağıttığı için, sanal sunucunun kolektif işleme kapasitesi, bir sunucunun kapasitesinden açık ara daha büyüktür. Yük dengeleyiciler, kullanıcı taleplerini bütün gerçek sunucular arasında dağıtmak için bir yük dağıtma algoritması kullanmaktadır. Eğer algoritma mükemmelse, sanal sunucunun kapasitesi bütün gerçek sunucuların toplam işlem kapasitesine eşit olacaktır. Yük dağıtma algoritmalarının etkinliğini içeren birkaç faktörden dolayı bu nadir rastlanır bir durumdur. Buna rağmen, sanal sunucu kapasitesi bütün gerçek sunucuların toplam işlem kapasitesinin yaklaşık yüzde 80-90'ı olsa bile, bu mükemmel ölçeklenebilirlik sağlamıştır.

2.5.1.2. Kullanılabilirlik

Yük dengeleyiciler gerçek sunucuların ve onların çalıştırdığı uygulamaların durumunu sürekli gözlemlemektedir. Eğer bir gerçek sunucu veya uygulama durum kontrolünde çökerse, yük dengeleyici bu sunucuya herhangi bir kullanıcı talebi göndermeyi iptal eder. Çökmüş sunucu tarafından işletilen her mevcut bağlantı ve talep kaybedilmiş olmasına rağmen, yük dengeleyici bütün gelecek talepleri çalışan gerçek sunuculardan herhangi birine yöneltecektir. Eğer yük dengeleyici yoksa, sunucunun veya uygulamanın durumunu kontrol eden ağ izleme aracına güvenilerek kullanıcı taleplerinin manüel olarak farklı bir gerçek sunucuya yöneltilmesi

gerekmektedir. Yük dengeleyici bunu anında net bir şekilde yaptığından, erişilememe süresi dramatik olarak minimize olur. Çöken sunucu tamir olduğu zaman, yük dengeleyici durumu tespit etmektedir ve bu sunucuya talepleri yönlendirmeye başlamaktadır.

2.5.1.3. Yönetilebilirlik

Eğer sunucunun donanımı yükseltilmeye ihtiyaç duyarsa veya işletim sisteminin ya da uygulama yazılımının yeni sürüme yükseltilmesi gerekirse; sunucu kapatılmalıdır. Yükseltme talebin az olduğu saatlerde planlanabileceği halde, bir erişilemeyen zaman dilimi muhakkak olacaktır. Bazı kurumların bu zaman dilimini zarar görmeden atlatması mümkün olmayacaktır. Eğer özellikle çeşitli zaman aralıklarında dünyanın her yerinden kullanıcılar tarafından sunuculara erişiliyorsa, bazı kurumların gerçekten yoğunluğun az olduğu saat aralığı bulması mümkün olmayacaktır. Bir yük dengeleyici kullanarak, net bir şekilde, erişim engellemesi olmadan sunucu bakıma alınabilmektedir. Yük dengeleyiciler yeni talep ulaştırılmasını engelleyerek ve var olan her bağlantının sonlanmasını bekleyerek sunucunun otomatik zararsız kapanmasını sağlayabilmektedir. Bütün mevcut bağlantılar kapanır kapanmaz; sunucu bakım için, güvenli olarak çevrimdışıya alınabilmektedir. Yük dengeleyici taleplere hizmet vermeye onları kalan gerçek sunucular arasında dağıtarak devam ettiği için, bu kullanıcılar için tamamen saydam olacaktır.

Yük dengeleyiciler uygulamaları sunuculardan ayırarak da yönetilebilirliğe yardım etmektedir. Örneğin kullanılabilir on sunucu ile web (HTTP) ve dosya transfer (FTP) olmak üzere iki uygulamanın çalışacağı bir sisteme ihtiyaç duyulduğunda: ftp uygulamasını iki sunucu ile ve daha çok talep olacağı için http uygulamasını sekiz sunucu ile çalıştırmak uygun olacaktır. Yük dengeleyici olmaksızın, ftp uygulaması için iki sunucunun IP adresleri arasında ve http uygulaması için sekiz sunucunun IP adresleri arasında, round-robin sağlamak için DNS kullanılabilirdi. Eğer ftp uygulamasına talep birden artarsa, bu uygulamayı başka bir sunucuda çalıştırma ihtiyacı ortaya çıktığında, üçüncü sunucu IP adresini eklemek için DNS'in modifiye edilmesi gerekmektedir. Bunun yürürlüğe girmesi uzun zaman alacaktır ve performans problemine hemen çözüm olmayacaktır. Eğer bir yük dengeleyici

kullanılmış olsaydı, sadece bir VIP ilan edilmesi yeterli olacaktı. FTP için sunucu 1-2 ve web uygulamaları için 3-8 sunucuları ile VIP'yi birleştirmek için yük dengeleyici ayarlanabilir. Buna “bağlama” denmektedir. Bütün FTP talepleri iyi bilindiği üzere port 21'den alınmaktadır. Yük dengeleyici hedef TCP port tabanlı talep tipini tanımaktadır ve uygun sunucuya yönlendirmektedir. Eğer FTP için talep artarsa sunucu 3 FTP uygulamasını çalıştırması için ayarlanabilir ve VIP'e bağlanabilir. Böylece yük dengeleyici FTP uygulamasını çalıştıran üç sunucu olduğunu tanımaktadır ve talepleri üçü arasında dağıtmaktadır, böylece FTP talepleri için hemen artan toplam işleme kapasitesi sağlamaktadır. Uygulamayı bir sunucudan diğerine taşıma imkânı veya kullanıcılar için kesinti olmadan verilen uygulama için fazladan sunucular eklemek, sunucu yöneticileri için güçlü bir araçtır.

Yük dengeleyiciler, içerik yönetimi olarak bilinen, büyük miktarda içeriği yönetmeye de yardımcı olmaktadır. Bazı sunucular, bir sunucunun sağlayabileceğinden çok fazla içeriğe sahip olabilmektedir. Sunucular her grubun içeriğinin belirli parçasını sunmakla sorumlu olduğu farklı gruplar halinde organize edilebilmektedir. Yük dengeleyici, http taleplerinde, talepleri uygun URL tabanlı gruba yönlendirmektedir.

Yük dengeleyiciler işletim sisteminden etkilenmemektedir çünkü standart ağ protokolü tabanlı işlem yapmaktadırlar. Yük dengeleyici yükü her sunucuya işletim sistemine bakmaksızın dağıtabilmektedir. Bu yöneticilere farklı sunuculardan küme oluşturabilme hatta toplam işleme kapasitesini ölçeklemek için her bir sunucunun avantajlarını alma imkânı sunmaktadır.

2.5.1.4. Güvenlik

Yük dengeleyiciler, sunucular için ön uç olduğundan, yük dengeleyiciler sunucuları kötü niyetli kullanıcılardan koruyabilmektedir. Birçok yük dengeleme ürünü, sunuculara ulaşmaktan dolayı belirli atak tiplerini durduran birkaç güvenlik özelliği ile birlikte gelmektedir. Gerçek sunucular, dış kullanıcıların doğrudan erişimini engellemek için RFC 1918 standardında tanımlandığı gibi özel IP adreslerinde verilmiş olabilir. Özel IP adresleri internette açık değildir. İnternetteki herkes, özel IP adresi olan bir alan adıyla iletişime geçmek için ağ adres dönüşümü (NAT) sağlayan

bir cihaza katlanmaları gerekmektedir. Yük dengeleyiciler kullanıcı taleplerini farklı gerçek sunuculara yönlendirmenin ve dağıtmanın bir parçası olarak ağ adres dönüşümü sağlayan doğal orta dereceli araçlardır. Yük dengeleyicide sanal IP bir genel IP adresi olabilir böylece internet kullanıcıları sanal IP'ye erişmeleri mümkün olacaktır. Fakat yük dengeleyicinin arkasındaki gerçek sunucuları bütün iletişimi yük dengeleyici ile yapmaya zorlamak için özel IP'ye sahip olmaları gerekebilecektir.

2.5.1.5. Hizmet Kalitesi

Hizmet kalitesini birçok farklı yoldan tanımlamak mümkündür. Sunucu veya uygulama cevaplama süresi, verilen uygulama hizmetinin kullanılabilirliği veya kullanıcı tipine dayanan farklılaştırılmış hizmetler sağlama yeteneği olarak tanımlanabilir. Örneğin, sık uçuş yapanlar için program bilgisi sağlayan bir web site; platin üyelerine altın veya gümüş üyelerinkinden daha iyi cevaplama süresi sağlamak istemesi mümkündür. Yük dengeleyiciler talep paketlerindeki bazı bilgilere dayanan kullanıcıları ayırmak için; onları bir veya bir grup sunucuya yönlendirmek için; veya hizmetin arzu edilen sınıfını sağlamak amacıyla IP paketinde öncelik bitini ayarlamak için kullanılabilir [12].

2.6. YÜK DENGELEME TÜRLERİ

Yük dengeleme cihazları genellikle iki türde olur: switch tabanlı yük dengeleyici veya sunucu tabanlı yük dengeleyici. Her biri genel avantajlara ve dezavantajlara sahiptir, fakat bu fazlasıyla satıcının teknolojiyi nasıl uyguladığına bağlıdır.

2.6.1. Sunucu Tabanlı Yük Dengeleme

Sunucu tabanlı yük dengeleyiciler genellikle bir standart işletim sistemi çalıştıran kişisel bilgisayar tabanlı ünitelerden oluşmaktadır. Cisco'nun LocalDirector'ünü ve F5'in BIG-IP'sini sunucu tabanlı yük dengeleyiciye örnek olarak göstermek mümkündür. Sunucu tabanlı yük dengeleme fonksiyonları sunucuların işletim sisteminin ağ yığınının tepesinde çalışan yazılım kodları tarafından sağlanmaktadır. Genellikle, kullanılan işletim sistemi, BSDI gibi ticari işletim sisteminin orijinal

ekipman imalatçı versiyonu (Original Equipment Manufacturer, OEM) veya Linux, FreeBSD gibi modifiye edilmiş ücretsiz işletim sistemini versiyonu olabilmektedir. Cisco'nun LocalDirector'u gibi bir yük dengeleyicide, bütün işletim sistemi üretici tarafından yazılmıştır.

Elde edilmesi kolay olduğundan, yaygın kullanılan işletim sistemlerine kaynakları kodlamak için sunucu tabanlı yük dengeleyiciler geliştirmek kolay olmaktadır. Bu kısa kodlara ve yeni özellikli geri dönüşlere yardım edebilmektedir, ama engel de olabilmektedir. Daha kısa kod zamanları ile, yazılım hataları daha yaygın olabilmektedir. Bu kolay geliştirme evreleri, sunucu tabanlı yük dengeleyicilerin tipik olarak esnek oldukları anlamına gelmektedir. Yeni özellikler hızlıca bellekten yollanabilmektedir ve makineler performans izlemenin yeni ve yaratıcı yollarını diğer işler kadar üzerlerine alabilmektedir.

2.6.2. Switch (Donanım) Tabanlı Yük Dengeleme

Switch tabanlı yük dengeleyicileri, ayrıca donanım tabanlı yük dengeleyiciler olarak da bilinmektedir, paket yeniden yazma fonksiyonlarını sağlamak için Uygulamaya Özel Entegre Devre (ASIC) yongalarına dayanan cihazlar olarak tanımlamak mümkündür. ASIC yongaları Pentium veya akrabalarından daha çok özelleşmiştir. Pentium ve PowerPC yongaları Quake III veya Microsoft Word gibi çok geniş türde yazılım çalıştırmaya imkân veren genel komut kümesine sahip bulunmaktadır. ASIC yongası bir işten soyutlamanın birkaç katmanını kaldıran bir işlemcidir. Bu uzmanlaşma nedeniyle, ASIC yongaları genelde yazılım işlerini bir genel işlemciden daha hızlı ve daha etkili sağlamaktadır. Dezavantajı yongaların hiç esnek olmaması olarak gösterilebilir. Eğer yeni bir iş gerekirse, o zaman bir ASIC dizaynının yeniden inşa edilmesi gerekebilmektedir. Ancak IP protokolü değişmeden kalacağı için, bu fonksiyonları bir ASIC'e aktarmak mümkün olacaktır.

Alteon ve Cisco CSS yük dengeleme switch hatları olduğu kadar Foundry'nin SeverIron serisi, karakterize edilen switch tabanlı yük dengeleyiciye örnek olarak gösterilebilmektedir.

Switch tabanlı yük dengeleyiciler için kod geliřtirmek daha zor kabul edilmektedir. Genelde řahsi mimariler veya en azından minimum geliřtirme kaynađı üzerinde alıřmaktadırlar. Bu yzden kod daha yavař sonuları fakat daha stabildir.

Switch tabanlı rnler ayrıca genelde daha hızlı alıřmaktadır. ASIC yongaları tek bařına yazılımdan daha etkili olmaktadır. Ayrıca Gbps'ler dzeyinde trafiđi ele alacak dahili band geniřliđi omurgalara sahiptirler. Kiřisel bilgisayarlar ise daha ok genel I/O trafiđine ynelmiřtir ve IP veya paket trafiđi iin optimize edilmemiřtir.

2.6.3. DNS Tabanlı Yk Dengeleme

Yk dengeleme bir teknoloji veya geerli bir rn olmadan nce site yneticileri DNS round robin olarak bilinen yk dengeleme iřlemi zerinde alıřmaktaydı. DNS round robin (hepsini birer kez dene), bir alan adını (hostname) birden fazla IP adresi ile iliřkilendirmeye izin veren DNS'in bir fonksiyonunu kullanmaktadır. Her DNS kaydı bir alan adını (www.karabuk.edu.tr gibi) bir IP adresine (208.185.43.202 gibi) eřleřtiren A kaydı olarak bilinen yapıda oluřturulmuřtur. Genelde bir alan adı iin sadece bir IP verilmiřtir. ISO'nun DNS sunucusunun altında, BIND 8, www.karabuk.edu.tr iin:

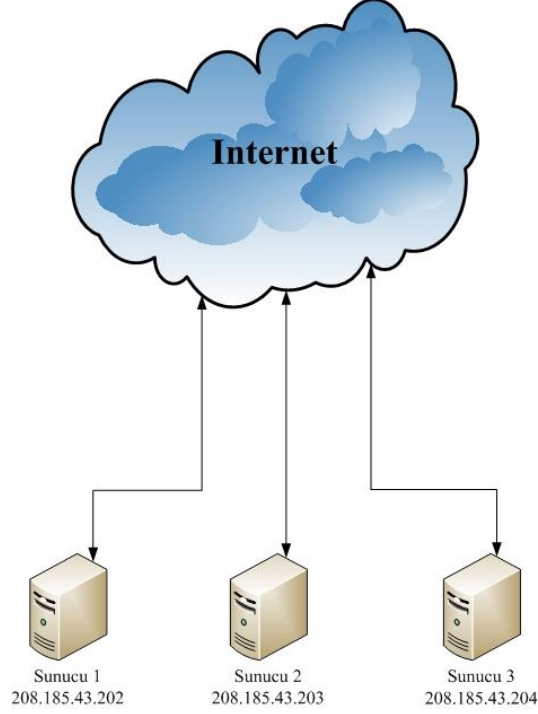
```
www.karabuk.edu.tr  IN                208.185.43.202
```

řeklinde grnen DNS kaydının yapısını gstermektedir. DNS round robin ile bir alan adına oklu IP adresi vermek, listelenen IP adreslerine trafiđi ařađı yukarı eřit olarak dađıtmak mmkn hale gelmektedir. rneđin, www.karabuk.edu.tr sitesi iin yk paylařtırmak istediđimiz 208.185.43.202, 208.185.43.203 ve 208.185.43.204 IP adresleri ile  sunucu olsun.

 IP adresi iin DNS sunucudaki konfigrasyon řyle grnecektir:

```
www.karabuk.edu.tr  IN                A                208.185.43.202
                    IN                A                208.185.43.203
                    IN                A                208.185.43.204
```

Sonuçta www.karabuk.edu.tr için yöneltilen trafiğin listelenen üç IP adresi arasında dağıtılması Şekil 3.2’de gösterilmiştir [11].



Şekil 2.5. DNS tabanlı yük dengeleme ile trafik dağıtımı [11].

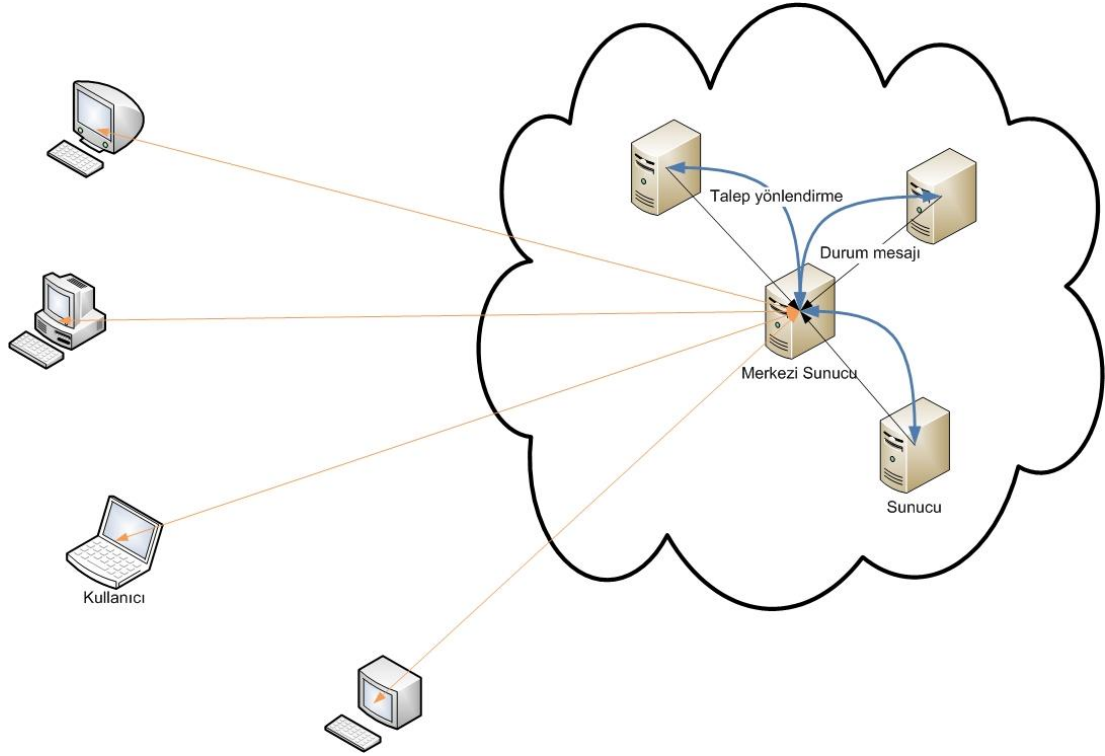
2.7. YÜK DENGELEME MODELLERİ

Merkezi ve dağıtık olmak üzere iki yük dengeleme modeli bulunmaktadır.

2.7.1. Merkezileştirilmiş Model

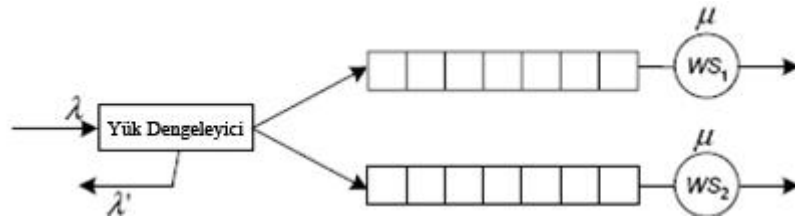
Bu modelde, bir merkezi sunucu global durumu idame ettirir ve şahsi sunucular ile periyodik olarak haberleşerek her bir sunucunun yükünden haberi olmaktadır. Merkezi sunucudaki yük bilgisinin hassasiyeti merkezi sunucuya diğer sunucuların yük durumları ile ilgili bilgiyi ne sıklıkta mesaj gönderdiklerine bağlı olmaktadır. Genellikle kullanılan UDP yayın protokolünü hesaba katmadan, diğer sunuculardan merkezi sunucuya ve tersine mesajların kaybolmasını engellenmesi amacıyla sunucu iletişimi için TCP kullanılmaktadır. Yayın için gelen talepleri, merkezi sunucu kümedeki en az yüklü sunucuya yönlendirmektedir. Bu model bu yüzden, aşağıda tanımlanan dağıtık modelin aksine Şekil 2.6’da gösterilen şekliyle, tam olarak bir

sunucu her kullanıcı talebini sunduğu için, tek-yayın tek-sunucu modeli olarak karakterize edilebilmektedir. Eğer bir sunucu merkezi sunucuya kendi durumu hakkında herhangi bir mesaj görmekte başarısız olursa, pasif olarak kabul edilmektedir. Hiçbir kullanıcı talebi, yeniden aktif olana ve durum mesajları göndermeye devam edene kadar bu sunucuya yönlendirilmemektedir [13].



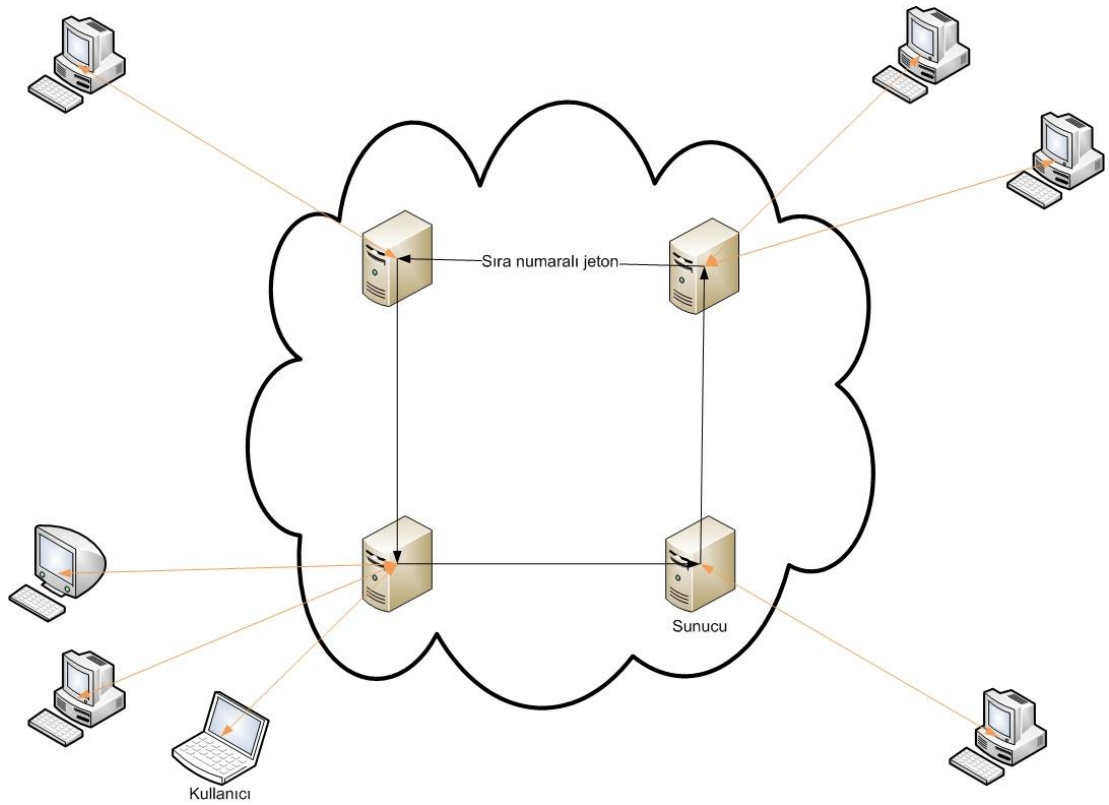
Şekil 2.6. Merkezileştirilmiş kontrol modeli [13].

Şekil 2.7'de gösterildiği gibi iki özdeş web sunucu WS_1 ve WS_2 tanımlanmaktadır. Her bir sunucu sonlu K tampon alanına sahip olduğu varsayılmaktadır. Kullanıcı talepleri ilk olarak, onları bir yönlendirme politikası kullanarak Şekil 2.7'deki iki sıradan birine yönlendiren bir merkezi yük dengeleyiciye gelmektedir.



Şekil 2.7. Merkezi yük dengeleme için sıra modeli [10].

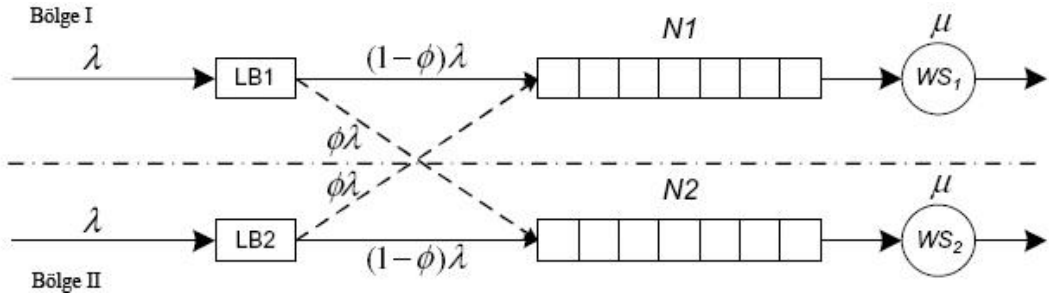
Kullanıcı taleplerinin ulaşmasının, λ varış oranı ve μ bulunma oranı ile üssel olarak dağıtılmış her Web sunucu için hizmet zamanı ile bir Poisson işlemi takip ettiği kabul edilir [10]. Poisson olasılık dağılım fonksiyonu; n olay sayısı, p olayın olma olasılığı, $X = 0,1,2, \dots$ için $\lambda = np$ gösterildiği yerde $P(X) = \frac{e^{-\lambda}\lambda^x}{x!}$ dir [11]. Yük dengeleyicinin sonsuz bir kapasitesi olduğu ve tam olarak kuyruk uzunluğunu bildiği varsayılmaktadır ki gecikmeye meydan vermeden acil olarak bir talebi kuyruklardan birine yönlendirebilmektedir. Her bir sunucuda talepler ilk gelen ilk yerine getirilir prensibine (FCFS) göre işlenmektedir. Gelen talepler eğer sadece bütün kuyruklar doluyorsa reddedilmektedir, reddedilme oranı λ' 'dir. Her bir kuyrukta sonlu tampon olduğundan ve eğer kuyruk doluyorsa talepler reddedileceğinden, $\rho = \frac{\lambda}{2\mu} < 1$ olması istenmemiştir [10].



Şekil 2.8. Dağıtık kontrol protokolü [14].

2.7.2. Dağıtık Kontrol Modeli

Çoğu merkezileştirilmiş çözüm bir noktada toplanan aşırı iş yükünün getirdiği başarısızlığa mahkûm olmaktadır. Merkezi sunucuda, limitli kaynak kullanılabilirliği yüksek yük altında tıkanıklığa neden olabilmektedir. Diğer taraftan ise, kurgulanması basit olduğundan kolay uygulanabilirlik avantajı bulunmaktadır. Dağıtık kontrol modeli dağıtık mimari kullanır. Model her bir kullanıcı talebini kümedeki sunuculara bölmek ve dağıtmak için, jeton gönderme şeması çalıştırmaktadır. Kümedeki her bir sunucu böylece kullanıcılara kapasitesi ölçüsünde hizmet sağlamaktadır [14].



Şekil 2.9. Dağıtık yük dengeleme modeli [10].

Merkezileştirilmiş yük dengeleme modelinin esneklik ve yüksek kullanılabilirlik gibi bazı avantajlara sahiptir. Fakat ağ gecikmesi problemini çözememektedir. Örneğin uzak bir bölgeden gelen kullanıcı hizmet talepleri web sunucusuya ulaşmak için uzun bir yol kat etmek zorundadır. Bu ağ tıkanması oluşturacaktır ve aynı zamanda ağdaki sıra etkisinden dolayı yüksek gecikmeye neden olacaktır. Son yıllarda içerik sağlayıcılar web veya uygulama sunucularını ağ köşelerine yerleştirmektedir ve coğrafik olarak dağıtılmış sunucular arasında yükü dinamik olarak dağıtmaktadır. Bu sunucu seçimini ve yük optimizasyonunu arttırmaktadır. Dağıtık yük dengeleme için genel bir model Şekil 2.9'da gösterilmiştir [10].

2.8. YÜK DENGELEME POLİTİKALARI

PANTS tarafından sağlanmış yedi adet yük dengeleme politikası bulunmaktadır:

- 1- Lider (Leader Policies),

- 2- Zaman çizelgesi (Time Line Policies),
- 3- Eşik (Threshold Policies),
- 4- Rastgele (Random Policies)
- 5- LRU (Least Recently Used Policies),
- 6- Hepsini Birer Kez Dene (Round Robin Policies)
- 7- NFS (Network File System).

İşlemleri dağıtmak için her politika farklı yaklaşımlar sergilemektedir. Bununla birlikte bazı politikalar çok benzerdir ve bir politika genelde bir diğerinin anahtar durumlarının değiştirilerek modifiye edilmiş halidir. Bu politikalardan birkaçında, sistemdeki düğümler mevcut iş yüklerine göre çok yüklü veya az yüklü olarak sınıflandırılmaktadır. Önceki araştırmalar yükü ölçmek ve düğümleri çok yüklü veya az yüklü diye sınıflandırmak için gerekli metotları keşfetmiştir. Politikalarda, bir işlem çok yüklü düğümde başlatılmak üzere olduğunda, politika işin transfer edilebileceği daha az yüklü bir düğüm tanımlamaya çalışmaktadır.

2.8.1. Lider Politikası

Uygulanan ilk politika Lider politikası olmuştur. Bu politikada, lider denen bir düğüm, diğer düğümlerin durumlarının izlerini tutmaktadır ve işleri dağıtmak için talepleri almaktadır. Sistem performansının ve stabilitenin azaldığı kümede, küçük bir küme bile olsa, Lider düğümün kullanılması bir tıkanıklığa neden olabilmektedir.

2.8.2. Rastgele Politikası

Rastgele politikası, her ikisinin de az yüklenmiş düğümlerin listesinden seçmek için rastgele seçimi kullanması açısından Lider politikası ile çok benzerlik göstermektedir. Rastgele politikasında en önemli farklılık her bir düğümün, düğümlerin kendi durumlarını yayınlaması tarafından güncellenen, kendi listesini idame ettirmesidir. Bu yolla, rastgele politikası yetki dağıtmıştır ve lider düğümünün rolü elenmiştir. Kısmi düğüm durum bilgisine dayalı rastgele iş dağıtan openMosix gibi sistemlerde benzer politikalar bulunabilmektedir [15].

Rastgele politikasının analizi basittir. Eđer bütün ulařımlar Poisson iřlemine λ oranı ile takip ederse ve eđer ulařan yayın rastgele ikiye bۆlünürse, o zaman her bir sunucuya ulařan iřlem de $\lambda/2$ oranında Poisson süreci olmaktadır. Bařka bir deęiřle, iki özdeş $M/M/1/K$ kuyruk ile karřı karřıya kalınmıřtır. İlk kuyruk dikkate alınmıřtır. Sistemde (ya kuyrukta ya da hizmette) n talep bulunma olasılıęını

$$p_n = \frac{(1-p)p^n}{1-p^{K+1}}, 0 \leq n \leq K \quad (2.1)$$

denklemini vermektedir. Bütün sistemin normalize edilmiř reddedilme oranı (NRR) böylece kolayca řu řekilde

$$NRR = \frac{\lambda'}{\lambda} = \frac{1}{2}p_K + \frac{1}{2}p_K = \frac{(1-p)p^K}{1-p^{K+1}} \quad (2.2)$$

hesaplanabilmektedir. Her bir sunucudaki ortalama talep sayısını

$$N = \sum_{i=0}^K i \cdot p_i = \begin{cases} \frac{p}{1-p} - \frac{(K+1)p^{K+1}}{1-p^{K+1}}, & p \neq 1, \\ \frac{K}{2}, & p = 1, \end{cases} \quad (2.3)$$

denklemini vermektedir. Sistemdeki ortalama cevap süresi $2N$ 'i geđerli varıř oranına bölerek hesaplanabilmektedir [10]:

$$W = \frac{2N}{\lambda' - \lambda} = \frac{2N}{\lambda(1-p_K)} = \frac{2N(1-p^{K+1})}{\lambda(1-p^K)}. \quad (2.4)$$

2.8.3. NFS Politikası

NFS Politikası, rastgele politikası ile benzerdir ve hâlihazırda sistemdeki bütün düęümlerin ulařabildięi NFS, daęıtık dosya sistemindeki dosyaları kullanmaktadır. Kullanılabilir (az yüklü) ve kullanılamaz (çok yüklü) diye iki dizin bulunmaktadır. Bu dizinlerdeki her dosyaya bir düęümün IP adresi olan, bir dosya adı verilmiřtir. Bir iř talebi alındıęında, uygun dizin tarafından bir dosya rastgele seęilmektedir ve iř bu düęüme transfer edilmektedir. Eđer kullanılabilir dizin bořsa, o zaman kullanılamaz

dizinden bir düğüm seçilmektedir. Geniş bir kümede bu paylaşılmış dosyalara erişim bir tıkanmaya neden olabileceğinden NFS politikası yetersiz kalacaktır. NFS politikasının aksine, geniş bir kümede rastgele düğüm seçilmiş olsa performans artması bile mümkün olabilecektir.

2.8.4. Zaman Çizgisi Politikası

Zaman çizgisi ve eşik politikaları birbirine benzerdir. Bu politikalarda, bir düğüm iş talebi için ihtiyaç duyulduğunda, az yüklenmiş düğümler tarafından dinlenen çoklu yayın adresine bir mesaj yayınlanmaktadır. İlk cevaplayan işi ifa eder. Zaman çizgisi politikası oldukça basittir ve hiçbir optimizasyon sağlamamaktadır. Bu yaklaşımın dezavantajı geniş kümelerde birden çok az yüklü düğüm olduğunda yüksek ağ yükü oluşturabilmesidir.

2.8.5. Eşik Politikası

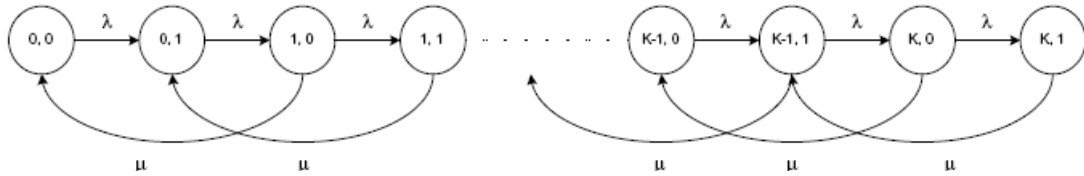
Eşik politikası gereksiz cevaplar gönderilme sayısını azaltmak için bir yaklaşım sağlamaktadır. Şuanda desteklenmesine rağmen bu politikalar onaylanmamıştır ve bu nedenle ileride bahsedilmeyecektir.

2.8.6. Hepsini Birer Kez Dene Politikası

Hepsini birer kez dene politikasında, bütün düğümlerin IP adresine göre sıralanmış, kullanılabilir ve kullanılamaz düğümleri içeren listesi idame ettirilmektedir ve politika listedeki bir sonraki düğümü işaret eden bir pointer saklamaktadır. Bir talep alındığında, listedeki hâlihazırdaki işaret edilen düğüm bildirilecektir ve işaretçi listedeki bir sonraki düğüme arttırılacaktır. Listedeki son düğüm bildirildikten sonra, işaretçi listedeki ilk düğüme ayarlanmaktadır. Böylece politika, düğümlerin durumlarını dikkate almayarak, düğümlerin listesinin başından sonuna sürekli dönmektedir. Teorikte, bunun arkasındaki motivasyon, yükün baştan sona düğümleri gezen bu döngü ile kümeye dengeleneceğidir. Bir satırda bir düğümün iki veya daha çok kez seçilebildiği rastgele seçim politikalarının hepsi için ortak olan bir problemi adreslemesi anlamına gelmektedir. Bu politikada çözüm, bir düğüm bir iş talebini

aldığında, bütün diğer düğümler de aynı duruma gelene kadar bir tane daha alamayacağını temin edilmesidir.

Bu politikanın beklenen davranışı işler eşit büyüklükte olduğu zaman iyi, fakat işlerin büyüklükleri değişkenlik gösterdiğinde yetersiz işlemedir. Bu politikanın listede geçiş yapmasındaki basitlikten kaynaklanmaktadır: Alışılmamış büyüklükteki bir tek iş talebi için büyüklüğünü yansıtmaları için düğüm seçimindeki sırayı alt üst etmelidir fakat yapmayacaktır. Örneğin, beş düğümlük bir küme ve işlerin birim zamanda beş defa dağıtıldığı işleme testi gibi bir test varsayalım. Eğer bütün işler eşit büyüklükteyse, onlar doğru olarak düğüm 1, 2, 3, 4, 5 düğümlerine gönderilecektir ve düğüm 1'deki iş bittiği zaman tekrar 1'e gönderilecektir. Böylece, her düğüm birim zamanda sadece bir iş yürütecektir. Üçüncü işin anormal şekilde büyük olduğunu varsayılırsa böylece ikinci seferde listede düğüm 3'e ulaşıldığında o hala üçüncü işi yapıyor olacaktır. En ideal olarak politika düğüm 3'ü atlamalıdır ve işi bir sonraki müsait düğüme göndermelidir. Politika uygulandığında durum bu değildir ve düğüm 3'e kümede dengesiz yük yapacak iş gönderilecektir [15].



Şekil 2.10. Round robin politikası durum geçiş diyagramı [10].

Kullanıcı taleplerinin ulaşma işlemi bir Poisson işlemi olduğundan, talepler arasında geçen zaman $\frac{1}{\lambda}$ ortalamalı üssel rastgele değişkenlerdir, böylece her web sunucu arasında geçen zaman $\frac{2}{\lambda}$ ortalama ile dağılmış tip-2 Erlang'dır. Dolayısıyla iki benzersiz $E_2/M/1/K$ kuyrukları ile karşılaşmıştır. İlk kuyruk dikkate alınmıştır. $E_k/M/1$ Kuyruğunun analizini takip ederken, sistemdeki n kullanıcının olasılığını q_{nj} temsil etmektedir ve $j(j = 0,1)$ durumunda olan varışı temsil etmektedir, o zaman sistemde n kullanıcının olasılığını $p_n = q_{n0} + q_{n1}$ formülü ile hesaplanan p_n takip etmektedir. Kuyruk modelinin geçiş durum diyagramı Şekil 3.4'de tanımlanmıştır, dengeleme eşitlikleri ise şöyledir:

$$q_{n-1,1} = \left(1 + \frac{1}{2p}\right) q_{n0} - \frac{1}{2p} q_{n+1,0}, 1 \leq n \leq K - 1, \quad (2.5)$$

$$q_{n0} = \left(1 + \frac{1}{2p}\right) q_{n1} - \frac{1}{2p} q_{n+1,1}, 1 \leq n \leq K - 1, \quad (2.6)$$

$$q_{K-1,1} = \left(1 + \frac{1}{2p}\right) q_{K0}, \quad (2.7)$$

$$q_{K,0} = \frac{1}{2p} q_{K1}, \quad (2.8)$$

$$q_{01} = q_{00} + \frac{1}{2p} q_{11}, \quad (2.9)$$

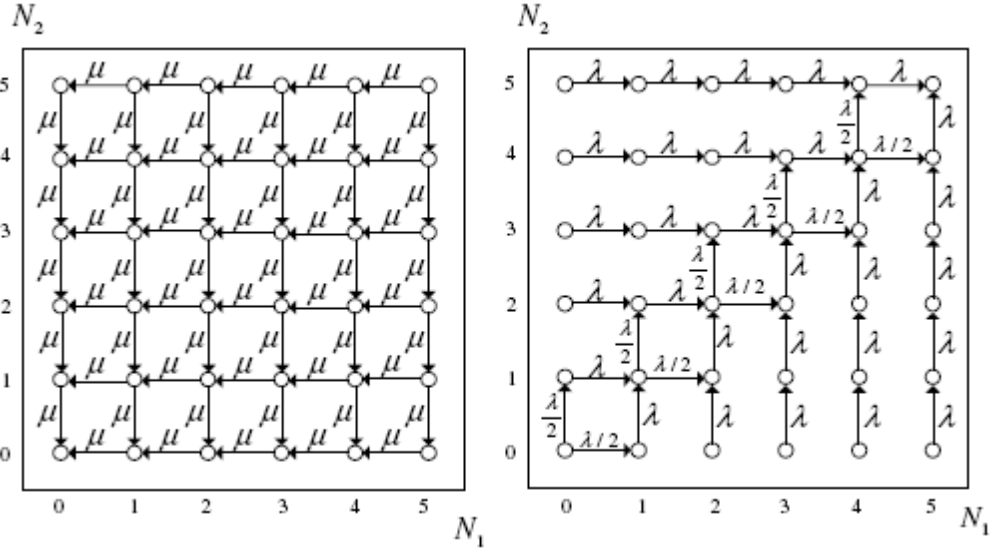
$$q_{00} = \frac{1}{2p} q_{10}. \quad (2.10)$$

Dengeleme eşitliklerinden q_{nj} için kapalı form çözümler elde etmek zor olduğundan, nümerik olarak kolaylıkla hesaplanabilir ve dolayısıyla $N = \sum_{n=0}^K np_n$ olduğu yerde beklenen hesap ölçümleri $NRR = p_K$ ve $\frac{2N}{\lambda(1-p_K)}$ gibi olacaktır [10].

2.8.7. LRU Politikası

LRU politikası, düğümlerin listesinin her bir düğümden hali hazırda yürütülen işlerin sayısına ve sonra her bir düğüme en son iş gönderilme zamanına göre sıralanmış olduğu, her birini bir kez dene politikasının bir modifikasyonu olarak tanımlanmaktadır. Bu, önceki politikada verilen problemi çözmektedir ve örneğin bir düğüm hali hazırda iki iş yürütüyorsa, diğer düğümler de ikişer iş yürütene kadar başka iş gönderilmeyeceğinden emin olunur. İlave olarak, bir düğüme gönderilen en son işin izlenmesi, her birini birer kez dene politikasındaki gibi, eğer tüm düğümler aynı sayıda iş yürütüyorsa o zaman bir düğüm bir iş alabilir, diğer bütün düğümler de iş alana kadar bir tane daha alamamasından emin olmayı sağlamaktadır. Bu iki modifikasyon iş büyüklüğüne rağmen artırılmış performanstan emin olması gerekmektedir. Her iki politikada ortak beklenen muhtemel dezavantaj, her ikisinin de çoklu kaynak düğümlerini ele alacak şekilde dizayn edilmemeleridir. Bu birden

fazla düğümden iş taleplerinin başlatılmasıdır. Listeler ve düğüm durumu, düğümler arasında paylaşılmadığından, işin optimal olmayan şekilde dağıtılması sonucunu verebilmesi mümkün olmaktadır. [15].



Şekil 2.11. En kısa kuyruk politikasının durum geçiş şeması [10].

2.8.8. En Kısa Kuyruk Politikası

Ayrıca bir diğer politika ise En Kısa Kuyruk Politikasıdır. Yük dengeleyici, alınan talep doğrultusunda, kuyruktaki en az talep sayısı ile onu iki web sunucudan birine göndermektedir. Eğer sayı eşitliği varsa, talep rastgele yönlendirilmektedir. $WS_i, i = 1, 2$ sistemde taleplerin sayısını N_i ifade etmektedir. Sistemin mevcut durumu (N_1, N_2) ile ifade edilebilmektedir. $p_{ij} = Pr[N_1 = i, N_2 = j]$ tanımlanır. Açık bir biçimde, eğer $i, j < 0$ veya $i, j > K$ 'dan herhangi biri ise $p_{ij} = 0$ 'dır. Geçiş durum diyagramı Şekil 2.7'de gösterilmiştir. Bu model için, verilen

$$Q_{ij} = \begin{cases} P_{i,j-1}, i \geq j + 2, \\ P_{i-1,j}, j \geq i + 2, \\ P_{i,j-1} + \frac{1}{2}P_{i-1,j} & , i = j + 1, \\ \frac{1}{2}P_{i,j-1} + P_{i-1,j}, j = i + 1, \\ P_{i-1,j} + P_{i,j-1}, i = j \end{cases} \quad (2.11)$$

Q_{ij} 'nin i ve j 'nin farklılığına dayandığı yerde, dengeleme eşitlikleri şöyledir:

$$(1 + p)p_{ij} = \frac{1}{2}(p_{i+1,j} + p_{i,j+1}) + pQ_{ij}, 1 \leq i \leq K, 1 \leq j \leq K, \quad (2.12)$$

$$\left(\frac{1}{2} + p\right)p_{0j} = \frac{1}{2}(p_{0,j+1} + p_{1j}), 1 < j \leq K, \quad (2.13)$$

$$\left(\frac{1}{2} + p\right)p_{i0} = \frac{1}{2}(p_{i+1,0} + p_{i,1}), 1 < i \leq K, \quad (2.14)$$

$$\left(\frac{1}{2} + p\right)p_{01} = \frac{1}{2}(p_{02} + p_{11}) + \frac{1}{2}\rho p_{00}, \quad (2.15)$$

$$\left(\frac{1}{2} + p\right)p_{10} = \frac{1}{2}(p_{11} + p_{20}) + \frac{1}{2}\rho p_{00}, \quad (2.16)$$

$$\rho p_{00} = \frac{1}{2}(p_{01} + p_{10}), \quad (2.17)$$

$$p_{KK} = \rho(p_{K-1,K} + p_{K,K-1}). \quad (2.18)$$

p_{ij} elde etmek için yukarıdaki eşitlik 3.16'daki eş zamanlı denklemleri çözmek, sonra reddedilme oranı ve ortalama cevaplama süresini hesaplamak gerekmektedir. p_{ij} 'yi bulmaktaki zorluk, geniş bir K 'da çok zaman alabilen, $(K + 1) \times (K + 1)$ kümesindeki eş zamanlı denklemlerin çözülmesi gerekmesidir. Buna ek olarak sistemdeki taleplerin toplam sayısının dağılımı hakkında ön bilginin olmamasıdır [10].

[10]'da detaylı bir şekilde incelenen ve ispatlanan yaklaşım ile bu model için performans ölçümleri

$$N = \sum_{i=0}^K \sum_{j=0}^K (i + j)p_{ij} \quad (2.19)$$

olduğu yerde şöyle elde edilebilir:

$$NRR = p_{KK} \quad (2.20)$$

$$W = \frac{N}{\lambda(1-p_{KK})} \tag{2.21}$$

BÖLÜM 3

SANAL SINIF UYGULAMALARI

Fiziksel olarak aynı mekânda yer almayan öğrencilerin ders içeriklerini/komutlarını almasına odaklanmış yeni eğitim alanı uzaktan eğitim olarak tanımlanmaktadır. Eğiticiler ve öğrenciler ya asenkron (tercihe göre email veya metin bazlı iletişim) yada onlara gerçek zamanlı iletişim kurma imkanı sağlayan (senkron) teknolojiyi kullanarak iletişim kurmaktadır.

Web 2.0'ın gelmesi uzaktan öğrenme sınıflarının artan beklentileri ile; dinamik içerik, içeriğin kolay birleştirilmesi ve zengin interaktif deneyimleri getirmiştir. Eğiticiler ve öğrenciler sanal sınıfta daha çok etkileşim talep etmektedir. Sınıf aktif bir alan ve canlı; metin bazlı, interaktif olmayan sınıfların mevcut standardını arttırmak veya değiştirmek için gerçek zamanlı iletişime ihtiyaç duyulan bir yer olarak ifade edilmektedir [16].

Uzaktan eğitim derslerinde video kullanımı uzun süreli bir gelenektir. Birçok üniversite ve eğitim kurumu uzaktan eğitim derslerini seksenlerden bu yana sabit telefon üzerinden talimatlı televizyon hizmeti ve kablolu televizyon kullanarak video ile vermektedir [17]. İnternetin ortaya çıkması ile ve internete evlerden erişimde band genişliğinin artışı ile, web yayını aracılığıyla uzaktan eğitim videolarının dağıtımını daha yaygın hale; sanal sınıf, video konferans üzerinden daha çok etkileşim sağlamaya müsait hale gelmiştir [16-17]. Video konferans için birçok teknoloji kullanmak mümkündür; eğiticiler uzaktaki öğrencilerine sadece standart elektronik medya yerine gerçek zamanlı çift yönlü video, ses ve veri iletişimi dağıtarak daha etkileşimli uzaktan öğrenme deneyimi sağlayabilmektedir [16].

Öğrenmeyi ve öğretmeyi desteklemek ve etkileşimi ve işbirliğini kolaylaştırmak için web konferans kullanımı dünya ölçeğinde birçok yüksek öğretim ortamında ana akım olmaktadır.

Yüz yüze oturumlar ile sanal işbirliği araçlarını yer değiştirmek, binlerce kilometre uzaktaki iş arkadaşları ile günlük bazda çalışmak, tamamen çevrimiçi olan bir konferansa katılmak bahsedilen teknolojinin getirilerindedir.

Web konferans yazılımı: gerçek zamanlı, internet tabanlı işbirliği sağlamaktadır ve anlık mesajlaşma (metin chat), VoIP, ses konferansı, video konferansı, paylaşılmış beyaz tahta ve paylaşılmış uygulama veya masaüstü gibi araçları genellikle içermektedir. Bir çok web konferans yazılım paketleri bu araçların birkaçını birleştirmektedir.

Hedef öğrencilerin çoğu kütüphane veya yüz yüze çalışma gruplarına erişimi olmayan uzak bölgelerde yaşamaktadır. Dahası, öğrencilerin büyük çoğunluğu olgun yaşta, tam zamanlı çalışan ve yoğun çalışma programı içinde çalışmaya vakit ayırabilen kimseler olmaktadır.

İlk örnekler eğitici ve öğrencilerin tahsis edilmiş odalardaki oturumlara senkron olarak katılması esasına dayanmaktadır. 1990'larda görsel-yazısal teknoloji ile üniversitedeki ilk denemelerinden sonra, öğrenci ve öğreticilerin tahsis edilmiş odalardaki oturumlara katılmalarını gerektirdiğinden ve bu nedenle de esneklikten uzak ve bütün öğrencilere ulaşamadığından teknoloji terk edilmiştir. O zamandan beri çevrim içi düşünme, çevrim içi materyal dağıtma ve asenkron&senkron metin bazlı tartışma ile öğrenim yönetim sistemlerine odaklanmaya yönelinmiştir. Mevcut literatür bu yaklaşımın risklerinden haber vermektedir ve yeni teknolojilerin nasıl mevcut geniş öğrenim yönetim sistemlerinin çok ötesinde çevrim içi pedagoji aldığını özetlemektedir. Literatür limitli araçlar ile tek, sıkıca odaklanmış ortam yerine, gevşek bağlı öğrenme ortamı inşa etmek için çeşitli araçların koleksiyonunun kullanılmasını desteklemektedir.

Asenkron tartışma grupları sosyo-yapılandırmacı öğrenme paradigmalarında çok başarılı görünmesine rağmen, örneğin iş kursunun içeriğinde; bilim, matematik veya istatistik gibi alanlarda sembol tabanlı iletişim için faydalı olamama eğilimi göstermektedir. Bazı disiplinler, örneğin metin veya ses tabanlı kanalla bir konu hakkında konuşurken (eş zamanlı) paylaşılan beyaz tahtaya yazı yazmak veya çizim yapmak gibi standart öğrenim yönetim sistemleriyle tam olarak kullanılabilir olmayan çevrimiçi iletişim için özelleşmiş araçlar gerektirir. Bu araçların eksikliği ve eğiticiler ve öğrenciler için düş kırıklığı web işbirliği araçlarının çeşitliliğini üniversitedeki kişiler tarafından deneye yol açmıştır. Elle yazılmış gönderilere izin veren bir özgür senkron chat istemcisi matematik ve istatistik disiplinlerinde test edilmiştir. Diğer çalışmalar, üniversitedeki keşfedilmemiş araçları araştıran Web 2.0 yapısını 3B çevrim içi sanal ortamlar ile pedagojinin uygulaması konusunda ilgili olarak araştıran Byl ve Taylor tarafından yapılan çalışmaları ve Hafeez-Baig and Danaher tarafından mobil öğrenme teknolojilerinde yapılan araştırmaları geliştirmeye yöneltmiştir [17].

Temel olarak, gerçek zamanlı konferanslar, web toplantıları veya canlı toplantılar da denen, web konferans yazılımı eşzamanlı iletişime ve mesafeler arası işbirliğine odaklanmıştır. Medya zenginliği teorisine ve medya eşzamanlılık teorisine göre, web konferans uygulamaları iş birliği için önemli rol oynamaktadır.

Tipik olarak, web konferans ya sunum için veya işbirliği için kullanılır. Bu nedenle, e-öğrenme perspektifinden olduğu kadar bilgi yönetimi tarafından da görülebilir, her iki alana da uygulanabilmektedir. Ancak günümüzde odaklanma sunumdan çok işbirliğine doğru kaymaktadır.

Web-konferansın tanımı hakkında, web konferans gittikçe gerçek zamanlı, senkron (eşzamanlı), işbirliğinin web tabanlı formu olarak anlaşılmasına rağmen, ya senkron veya asenkron (eşzamanlı olmayan) işbirliğine odaklanan iki temel açıklama bulunabilmektedir. Bu bakış açısı özellikle IDC, Gartner ve Frost&Sullivan gibi bilişim analistleri tarafından yayılmıştır.

Piyasaya yönelik olarak, web konferansı, sanal bir konferans veya toplantı yapılması için bir araç olarak internet kullanımı olarak tanımlamak mümkündür. Bu nedenle konferans uygulamaları: değişim, oluşturulma ve belirlenmiş veya spontane çevrimiçi toplantılar veya etkinlikler sırasında iki veya daha fazla kullanıcı tarafından bilgi incelenmesi için gerçek zamanlı bağlantı sağlamaktadır. Chat, olay planlama, video köprüleme, sunum sonrası düzenleme ve olay yönetimi gibi fonksiyonellikler geçtiğimiz yıllarda bazı satıcılar tarafından uygulamalara eklenmiştir.

Fakat bu gibi özelliklerde konferans hizmeti sunan birkaç ürün bulunmaktadır. Takım işbirliği sağlayan uygulamalardan bazıları eşzamanlı chat, uygulama paylaşımı ve çevrimiçi (online) sunum desteği de sunmaktadır. Fakat bu uygulamalar sadece müşteri/ürün odaklı pazarda veya özel iş süreçlerinde küçük grupların işbirliğine destek verebilmektedir. Video konferans uygulamalarının genel amaçlı toplantısının temel amacına hizmet etmemektedirler. Benzer olarak, birçok anlık mesajlaşma uygulaması eşzamanlı fonksiyonellikler sunmaktadır, fakat daha kompleks özelliklerin paylaşımı maalesef eksiktir.

Web konferans ticari iletişimi arttıran uygulamaların bir dizisini temin eden bir sanal konferans oda ortamı sağlamaktadır. Bütün bu uygulamalar iki temel kategoride gruplanabilmektedir [18]:

- 1- Sunum,
- 2- İş birliği.

Synnes'in 1999 da tanımladığı gibi, sanal sınıf ortamı geniş ölçekli dağıtık dersler (ders vermenin geleneksel metodu, ayrıca proje sunumları ve seminerler), sanal topluluklar (aktif olduğunda bütün katılımcıların katıldığı sürekli oturumlar), sanal öğreten odaları (çoğu öğretenin mesai saatlerinde ulaşabildiği sanal geçitler) ve sanal grup odaları (öğrencilerin grupta çalışırken katıldıkları, her grup için bir oturum) için kullanılabilir. Sanal sınıf ortamı, bütün katılımcılar arasında etkili olarak veri dağıtmak için IP Multicast denen bir ağ tekniği kullanmaktadır. Eğer IP Multicast kullanılamıyorsa, o zaman katılımcılar ya birbirlerine doğrudan (aynı oturumda

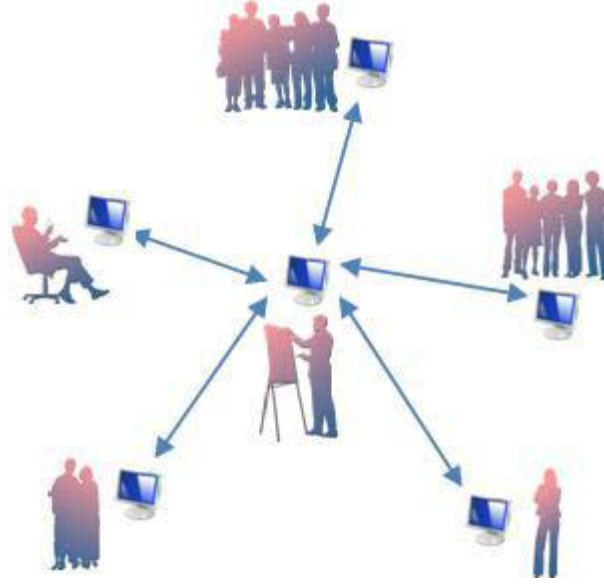
yalnızca iki kullanıcı sınırlaması ile) veya bir proxy sunucu (kullanıcı sayısı sınırlaması olmaksızın ama artan band genişliği kullanımı ile) ile bağlanmaktadır [19].

Web konferans platformları kullanıcılara iletişim için internet sayesinde bir çevrimiçi forumda birbirleriyle buluşma imkanı sağlayan yazılımdır. Web konferans yazılımı iki kategoriye ayrılabilir. Bir kategori, bazı yapılar ile sadece ses, video konferans ve toplantı moderatörü için daha zahmetsiz basitleştirme sağlayan platformlar içermektedir. Web konferans yazılımlarının çoğu ikinci kategoride yer almaktadır. İkinci kategori, doküman veya dosya paylaşımı için özellikler, paylaşılmış masaüstü erişimi, eşzamanlı düzenleme ve verinin paylaşılma, düzenleme ve web toplantısı esnasında kopyalanmasına imkân veren iletişimin diğer elektronik formlarını sağlamaktadır.

Yüz yüze gibi bir katılım sağlamaya izin veren video konferans uygulamalarının aksine, web konferans platformları bundan fazlasına izin verir. Katılımcılar ses ve videoyu, paylaşılan bilgisayar uygulamaları (paylaşılan beyaz tahta, masaüstü uygulama paylaşımı) ile birleştiren çevrimiçi toplantılar düzenleyebilmektedir. Her yıl uzaktan kompleks veri iletimi için geniş ve daha güçlü internet altyapısı kullanılmaktadır. Gerçek zamanlı video görüntülerin iletimi; beyaz tahta, masaüstü veya özel pencerelerin (uygulamalar) paylaşımı için yol sunmaktadır.

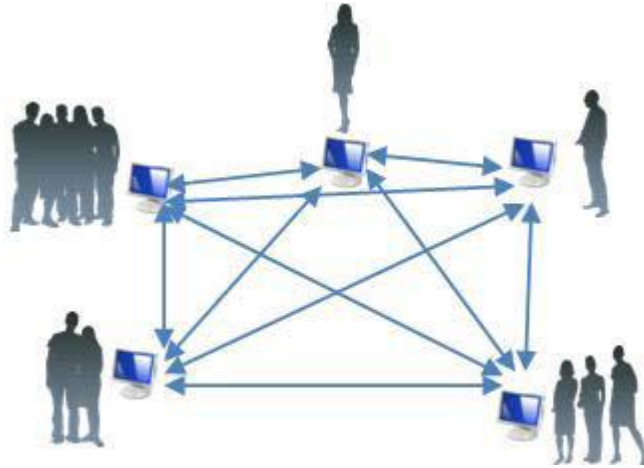
Web konferansın iki tipi bire çok ve çoğa çok web konferans olarak tanımlanabilmektedir.

Genel olarak web konferans araçları grubun bir üyesinin çok noktaya (hoparlörden dinleyici) bilgi gönderdiği tek yönlü olarak tasarlanmıştır. Çoğu diğer web konferans yazılımı temel çift yönlü veya çok boyutlu iletişim araçları da (oylama, anlık mesajlaşma, beyaz tahta, video besleme(ler) ve bir sunum paylaşabilen veya kendi masaüstünde bir uygulamayı gösterebilen sunucuya dönüt) sunmaktadır. Konferansın iki yolu da grup üyelerine içeriği gerçek zamanlı değiştirmeye izin vermektedir.



Şekil 3.1. Bire-çok web konferans [20].

Çoğa çok konferans Collins tarafından web işbirliği sistemi olarak adlandırılmaktadır. Web konferans yazılımı eğitim programları, ürün tanıtımları, durum raporlama, yazılım testi, veri paylaşma ve hızlı oylama gibi aktivitelere sahip olmayı planlayan toplantı grubu katılımcıları için uygundur. [20]



Şekil 3.2. Çoğa-çok web konferans [20].

Grup karar destek sistemleri çoğa çok iletişim paradigması ile bire çok kolaylaştırıcı rolünü birleştirmektedir. Web konferans sunuma odaklanmıştır ve grup karar destek sistemleri toplantı etrafında yapılandırılmıştır. Grup karar destek sistemleri beyin fırtınası, liste analizi, değerlendirme, derecelendirme, öncelik verme, ayırma,

sınıflandırma, gruplama, organize etme, problem çözme, risk değerlendirme, strateji planlama, konsensus sağlama, bilgi alma, ürün geliştirme gibi aktiviteleri desteklemektedir [20].

3.1. TEKNİK GEREKSİNİMLER

Sanal sınıf uygulamasının çalıştırılabilmesi için bazı altyapısal gereksinimleri bulunmaktadır. Bu gereksinimlerden bazıları kritik derecede önemlidir. Bazılarının ise performans arttırmaya etkisi mevcuttur.

3.1.1. Ağ

Video konferans araçlarının gereksinimlerinden bir tanesi ağ kalitesidir. Yeterli ağ kaynağı (band genişliği) veya güvenilirliği (veri kayıpsız ağ) olmaksızın sanal sınıf uygulaması gibi bir çalışma yapmak mümkün olmamaktadır. Bu nedenle altyapı planlamasının en başında, yeterli ağ kaynağının mevcut olduğundan emin olunması gerekmektedir.

Düşük güvenilirlikli (trafiğin kaybının büyük miktarı ile) bir ağ ortamı hızlıca daha yararsız hale getirecektir. %1-2'lik bir kayıp oranı sesi, dinlemek için yorucu bir hale getirecektir. Bu tekrarlama veya interpolasyon gibi şemayı onararak kısmen çözülebilir, fakat ses kalitesini de oldukça etkileyecektir. İhtiyaç fazlası veriyi göndermek veya veriyi önemli ve daha az önemli diye parçalara bölmek gibi veri kaybını önleyen şemalar bulunmaktadır, fakat bu metotların çoğu ağa yük getirir ve durumu daha da kötüleştirir. Bu yüzden, her oturum için gönderilecek verinin miktarını sınırlandırarak ağ tıkanmalarını önlemek önemli olmaktadır.

IP Multicast kullanma iyi bir seçenektir (ağ kaynak kullanımını ciddi derecede azaltmaktadır) fakat kurma ve idame ettirme teknolojisi komplekstir. Sonuç genelde ağın ya hata trafiğine doymuş, verinin bir yönde (bir yöndeki bütün trafiğin kesilmesi) transfer ediliyor, bütün verilerin transfer edilemiyor (bütün trafiğin kesilmesi); ya da verinin belli aralıklarla kesilmesi olmaktadır.

Ağ problemleri, özellikle IP Multicast kullanılırken, bir çok öğretici için yönetilebilirlik açısından çok gelişmiştir. Böylece ağı idame ettirecek ağ destek grubuna ihtiyaç olmaktadır. Ağ idamesi genellikle çoklu organize ve diğer güvenlik önlemleri kadar firewall için politikalar gerektirdiğinden maalesef başarılması kolay bir iş değildir.

3.1.2. Donanım

Video konferans araçlarına ilişkin bir diğer gereksinim, geri bildirim riski atmayarak veya sessiz olarak, ses donanımının uygulamada kurulumunun zor olması gerçeğidir. Problemlerden kaçınmak için ses seviyesinin başarılı şekilde kurulmuş olma tecrübesini temel olarak gerektirmesi, insanların bu ortamları kullanmayı neden kompleks bulduklarının en çok karşılaşılan sebebidir. Sihirbazlar veya kurulum araçları iyi olan bazı ortamlarda kullanılabilir. Bunlar tecrübesiz kullanıcılar tarafından maalesef çok az kullanılsa da, en çok ihtiyaç duyulandır.

Kurulum başlanınca, bir başka problemle karşı karşıya kalınmaktadır. Ortam nadiren kesin donanım için yazılmıştır, böylece ortama ilave olarak kullanıcı işletim sistemini ve sürücülerini (DirectX sürücüler gibi ikincil desteğe ihtiyaç duyabilen Windows altındaki video sürücülerini özellikle zordur) yönetmek zorundadır. Bu çoğu için üstesinden gelinmesi çok büyük zorluk olacaktır ve masaüstlerinde çalışan her şey için yardıma ihtiyaçları olacaktır.

Kullanıcıları donanım problemleri ile karşı karşıya bırakmaktan kaçınmak için çözüm; makineleri çalışma merkezlerinde kurmak veya destek merkezinin kurulum ve bakımla ilgilenmesini sağlamaktır. Bir başka çözüm öğrencileri öncelikle ortam üzerinde kendi sistemlerini nasıl kurulacaklarını ve kendi sistemlerine nasıl bakım yapılacağını içeren bir kurs almaya zorunlu tutmaktır.

3.1.3. Ortam

Video konferans araçlarının (maksimum 100 Kbps video) düşük band genişliği konfigürasyonunda kullanılan video kalitesine gösterilen genel reaksiyon kötü

görünmesi olmaktadır. İnsanlar video kalitesini televizyonda ulaşılan yayınımla karşılaştırmaktadır, ve tabii ki bu daha iyi fark edilmektedir. Sonuç genelde, birkaç öğrenenin dersin kaydının bütünü izlediği olgusu ile desteklenen, ilgi kaybıdır. Ses kalitesi aynı karşılaştırmadan geçmektedir, fakat aynı derecede değildir. Bu problemi ele almanın iki yolu bulunmaktadır. Bir yolu algılanan kaliteyi daha fazla veri göndererek veya daha etkili bir kodek kullanarak arttırmaktır. Diğer yol, geleneksel pasif ders modelini kullanmak yerine; öğreneni basitçe aktif ederek, kapsamaktır.

Diğer genel reaksiyon, sanal topluluğa (sanal geçit veya sürekli paylaşılan oturum) nadiren katıldıklarında öğrenenin diğer öğrenenlerin ne zaman çevrim içi olduklarını bilmemesidir. Genelde öğrenenler sanal topluluğa katılmadan önce kimin mevcut olduğunu bilmek ister. Katılmadan önce kimin dâhil olduğunun bilgisini sunmak veya hangi oturumlara öğrenenin arkadaşlarının katıldığına dair bir listeye sahip olmak bu problemin basit çözümüdür. Çoğu diğer ortamda basit fonksiyonellikler bulunmaktadır.

Son olarak genel teknik gereksinim kolay kullanımdır. Eğer ortamın kullanımı kolay değilse, o zaman birkaç öğrenci ders içeriğinde yer alacaktır. Kaçırdıkları derslerin kayıtlarını izleyen öğrenci grupları tespit edilmiştir. İlk bakışta dersi tartışmak için bir araya geldikleri düşünülebilir, fakat ortamı kompleks olarak algılamış ve bu nedenle onu nasıl yöneteceğini bilen bir öğrenen olarak sonradan katılmıştır. Daha az esnek fakat ortamın daha basit versiyonu eğitim içeriklerinde dikkate alınmalıdır [19].

3.2. İLETİŞİM STANDARTLARI

Aşağıdaki çizelgede popüler iletişim standartları sunulmuştur.

Çizelge 3.1. Popüler iletişim standartları [16].

	H.320	H.321	H.322	H.323 V1/V2	H.324
Onay Tarihi	1990	1995	1995	1996/1998	1996

Çizelge 3.1. (Devam ediyor).

Ağ	Dar bantlı anahtarlanmış sayısal ISDN	Geniş band ISDN ATM LAN	Ağ anahtarlama garantili band genişliği paketi	(Ethernet) Ağ anahtarlama garanti edilmemiş band genişliği paketi	PSTN veya POTS, analog telefon sistemi
Video	H.261 H.263 H.264	H.261 H.263	H.261 H.263	H.261 H.263 H.264	H.261 H.263
Ses	G.711 G.722 G.722.1 G.728	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.722.1 G.723 G.728 G.729 A/B	G.723
Çoklama	H.221	H.221	H.221	H.225.0	H.223
Kontrol	H.230 H.242	H.242	H.242 H.230	H.245	H.245
Çok Noktalı	H.231 H.243	H.231 H.243	H.231 H.243	H.323	
Data	T.120	T.120	T.120	T.120	T.120
İletişim Ara yüzü	I.400	AAL I.363 AJM I.361 PHY I.400	I.400 & TCP/IP	TCP/IP	V.34 Modem
Metinsel Chat	T.140	T.140	T.140	T.140	T.140
Kriptolama	H.233 H.234	H.233 H.234		H.235	H.233 H.234

3.2.1. Ses Standartları

Ses standartları şöyledir [16]:

- 1- G.711: 3.1 kHz analog sesin 48, 56 veya 64 kbps akışta kodlandığı yerde ses frekansının darbe kod modülasyonudur. Diğer hiçbir standart aynı ölçüde desteklenmediğinde kullanılmaktadır.
- 2- G.722: 7 kHz ses 48, 56 veya 64 kbps akışta kodlanmaktadır. Yüksek kalite sağlar, fakat band genişliği almaktadır.
- 3- G.722.1: 7 kHz ses düşük çerçeve kayıplı sistemlerde ahizesiz işlemler için 24 ve 32 kbps'da kodlanmaktadır.
- 4- G.723: 5.3 kbps & 6.4 kbps'da telekomünikasyon için 3.4 kHz çift oranlı konuşma kodeğidir.

- 5- G.728: 3.4 kHz düşük gecikme kodlu uyarılmış doğrusal tahmin (LDCELP), 16 kbps akışta kodlanmış 3.4 kHz analog sesdir. Bu standart düşük bit hızında güzel kalitede sonuçlar sağlamaktadır.
- 6- G.729 A/B: ASCELP yöntemi kullanılarak 8 kbps akışta kodlanmış sabite yakın ses kalitesi sağlayan 3.4 kHz konuşma kodeğidir. Annex A indirgenmiş karmaşıklık kodeğidir ve Annex B sessizlik önleme ve sessizlik giderici ton neslini desteklemektedir.

3.2.2. Video Standartları

Video standartları şöyledir [16]:

- 1- H.261: 352x288 (CIF or FCIF) ve 176x144 (QCIF) desteklemektedir. DCT tabanlı algoritma 2B'den 6B'ye ISDN iletişim için ayarlanmıştır.
- 2- H.263: H.261'in gelişmiş türevidir, POTS veri oranları için ayarlanmıştır. QCIF ve CIF üzerinde H.261'den daha iyi video sunarken, en çok QCIF ve SubQCIF (128x96 – SQCIF) amaçlanmıştır.
- 3- H.264: ITU ve ISO arasında işbirliği noktası. Yarım band genişliğinde aynı kaliteyi sağlayan H.263'ün geliştirilmiştir.

3.2.3. Kontrol/İletişim Standartları

Kontrol/iletişim standartları şöyledir [16]:

- 1- H.221: H.320'de kullanılan 64'den 1920 Kbps'a kanallarda görsel işitsel uzaktan hizmetler için iletim çerçeve yapısını tanımlamaktadır.
- 2- H.223: düşük bit oranlı multimedya iletişim için paket yönelimli mültepleks protokol bilirler; Annex A & B, 3G324M'de kullanıldığı gibi mobil uzatmanın hafif ve orta hata eğilimli kanallarını ele almaktadır.
- 3- H.225: medya akışı için mültepleks iletim formatlarını & garanti edilmemiş LAN hizmet kalitesinde senkronizasyonu belirlemektedir.
- 4- H.230: çerçeve senkronluk kontrolünü ve işitsel-görsel sistemler için işaret sinyallerini tanımlamaktadır.

- 5- H.242: H.320 tarafından kullanılan görsel işitsel terminaller arasında 2 Mbps üzerinde sayısal kanallarda iletişimi kuran protokolü ve kontrol prosedürlerini tanımlamaktadır.
- 6- H.243: kontrol prosedürlerini ve üç veya daha fazla görsel işitsel terminallerin arasında H.320 çok nokta konferans iletişimini kuran protokolü tanımlamaktadır.
- 7- H.245: H.323 & H.324 mültimedya iletişimleri için kontrol prosedürlerini ve protokolü tanımlamaktadır.

3.3. VIDEO KONFERANS STANDARTLARI

H.323 temel olarak terminaller arasında nasıl iletişim kurulacağını ve sırasıyla IP tabanlı ağ gerçekleştirmelerini tanımlayan standartlar kümesi olarak belirtilmektedir. H.323 iletişimde kapsanan dört temel bileşen bulunmaktadır: gerekli terminal, tercihe binaen geçit denetçisi, geçit ve çok noktalı kontrol ünitesi (MCU). Bu bileşenler uzak sınıfta etkileşimin çoğunluğunu sağlamaktadır.

3.4. AĞ ALT YAPISI

3.4.1. Band Genişliği Gereksinimleri

Uzaktan eğitim sınıfının herhangi bir tipinin yayınlanmasını planlayan kurumlar muhtemelen sınıfları minimum sorun ile sunmak için gerekli band genişliğine sahip olacaktır. Üniversitedeki genel olarak bir bağlantı üniversitenin büyüklüğüne bağlı olarak her hangi bir yerde 10 ila 150 Mbit arasındadır. Spektrumun sonu iyi olsa da kullanılabilir band genişliği miktarı öğrenci sayısına dayanan sınıf ölçeklendirilebilirliğini etkileyecektir. Ağ band genişliği ne kadar çok olursa o kadar iyi olur, her zaman doğru olmaktadır.

Uzaktan eğitim ile ilgilenmeye odaklanıldığında, eğer özellikle sınıf gerçek zamanlı etkileşim türüne sahipse band genişliği bir numaralı harici etken olmaktadır. Etkileşim için video konferans kullanıldığında, her uzak son nokta en az 256 Kbps download ve upload hızına sahip olmaya ihtiyaç duymaktadır. Bu son uçlar, yerleşim

yeri geniş bandı (DSL, kablo, ISDN), kurumsal düzey geniş bandı (DSL, kablo, ISDN) veya kurumsal düzey kiralık hatlardan (T1, T3, vb.) herhangi birinin üzerinde olması gerekir anlamına gelmektedir [16].

3.5. SANAL SINIF UYGULAMALARININ KARŞILAŞTIRILMASI

Çizelge 3.2. Sanal sınıf uygulamalarının karşılaştırılması [21].

Program	Lisans Modeli	Kapasite	Linux	Mac OS X	Microsoft Windows	Audio Desteği	Video Kalitesi	Video Desteği	Mesajlaşma Desteği	Masatsüti paylaşırma Desteği	Upload PPT	Upload PDF	Co-browsing	Taşınabilir Cihaz Desteği	Break-Out Sessions	Güvenli Erişim	Şifreli İletişim
Adobe Acrobat Connect	Proprietary	1-500	✓	✓	✓	✓	VGA	✓	✓	✓	✓	✓	✓	✓	✓	✓	X
BigBlueButton	LGPL + GPL	1-25	✓	✓	✓	✓	VGA	✓	✓	✓	✓	✓	X	X	X	X	X
BuddyMeeting	GPL + FREE	Upto 10	✓	✓	✓	✓	VGA	✓	✓	✓	✓	✓	X	X	X	X	X
Cisco Unified Meeting Place	Proprietary	1-500	✓	✓	✓	✓	VGA	✓	✓	✓	✓	✓	?	?	✓	✓	X

Çizelge 3.2. (Devam ediyor).

IBM Lotus Sametime	Genesys Meeting Center	Fuze Meeting	Eliminate	Dindim	Citrix GoToMeeting	Program
Proprietary	Proprietary	Proprietary	Proprietary	GPL +Proprietary	Proprietary	Lisans Modeli
?	125+	55+	2 - ?	20 - 250	15	Kapasite
✓	X	✓	✓	✓	X	Linux
✓	X	✓	✓	✓	✓	Mac OS X
✓	✓	✓	✓	✓	✓	Microsoft Windows
✓	✓	✓	✓	✓	✓	Audio Desteği
VGA,HQ	VGA	QVGA	QVGA	VGA	VGA	Video Kalitesi
✓	✓	✓	✓	✓	✓	Video Desteği
✓	✓	✓	✓	✓	✓	Mesajlaşma Desteği
✓	✓	✓	✓	✓	✓	Masatıstı paylaştırma Desteği
✓	✓	✓	✓	✓	X	Upload PPT
✓	X	✓	✓	✓	X	Upload PDF
?	✓	X	✓	✓	X	Co-browsing
?	✓	✓	?	?	X	Taşnabilir Cihaz Desteği
?	✓	X	✓	?	X	Break-Out Sessions
✓	X	X	X	✓	✓	Güvenli Erişim
X	X	X	X	X	X	Şifreli İletişim

Çizelge 3.2. (Devam ediyor).

PharMethod PharmaCAST	Openmeetings	omNovia Web Conference	Netviewer	IBM Lotus Sametime Unyte	Program
Proprietary	EPL	Proprietary	Proprietary	Proprietary	Lisans Modeli
1-Unlimited	1-25	2-5,000	1-100	?	Kapasite
X	✓	✓	X	✓	Linux
✓	✓	✓	X	✓	Mac OS X
✓	✓	✓	✓	✓	Microsoft Windows
✓	✓	✓	✓	✓	Audio Desteği
VGA, HD	VGA	VGA	VGA	VGA,HQ	Video Kalitesi
✓	✓	✓	✓	✓	Video Desteği
✓	✓	✓	✓	✓	Mesajlaşma Desteği
X	✓	✓	✓	✓	Masaüstü paylaşırma Desteği
✓	✓	✓	?	✓	Upload PPT
X	✓	✓	?	X	Upload PDF
X	✓	✓	?	?	Co-browsing
announced	?	?	X	?	Taşınabilir Cihaz Desteği
✓	?	?	?	?	Break-Out Sessions
✓	X	X	✓	✓	Güvenli Erişim
X	X	X	X	X	Şifreli İletişim

Çizelge 3.2. (Devam ediyor).

Program	Tokbox	VenueGen	WebEx	WebTrain
Lisans Modeli	GPL +Proprietary	Proprietary	Proprietary	Proprietary
Kapasite	20-25	1-500	1-100	200
Linux	X	X	✓	X
Mac OS X	✓	X	✓	X
Microsoft Windows	✓	✓	✓	✓
Audio Desteği	✓	✓	✓	✓
Video Kalitesi	VGA	VGA	VGA	VGA
Video Desteği	✓	✓	✓	✓
Mesajlaşma Desteği	✓	✓	✓	✓
Masaüstü paylaşma Desteği	X	✓	✓	✓
Upload PPT	X	✓	✓	✓
Upload PDF	X	✓	✓	✓
Co-browsing	?	✓	✓	✓
Taşınabilir Cihaz Desteği	?	X	✓	?
Break-Out Sessions	?	X	✓	✓
Güvenli Erişim	X	X	✓	✓
Şifreli İletişim	X	X	✓	✓

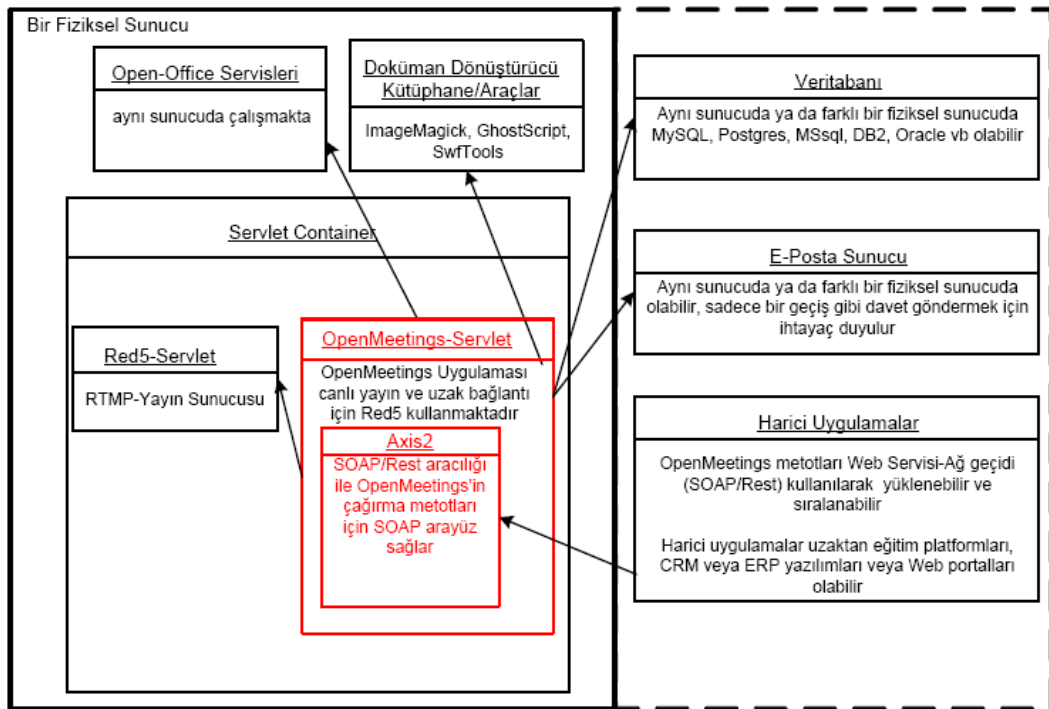
3.2. TERCİH EDİLEN SANAL SINIF UYGULAMASI: OPENMEETINGS

OpenMeetings internette anında konferans ayarlamaya izin veren ücretsiz tarayıcı tabanlı bir yazılımdır. Özelliklerinden bazıları şöyledir: [22]

- 1- Video/ses
- 2- Masaüstü paylaşma
- 3- Kayıt (Bütün oturum ve görüşme stili)
- 4- Çizim, yazma&düzeltilme, sürükle bırak, boyutlandırma, resimler, semboller ile beyaz tahta

- 5- Döküman dahil etme (tga, .xcf, .wpg, .txt, .ico, .ttf, .pcd, .pcds, .ps, .psd, .tiff, .bmp, .svg, .dpx, .exr, .jpg, .jpeg, .gif, .png, .ppt, .odp, .odt, .sxw, .wpd, .doc, .rtf, .txt, .ods, .sxc, .xls, .sxi, .pdf)
- 6- Çoklu dil desteği (Arapça, Basit Çince, Geleneksel Çince, Çekçe, Almanca, İngilizce, Fransızca, Galiçyaca, Macarca, Endonezya dili, İtalyanca, Japonca, Korece, Farsça, Portekizce, Rusça, İspanyolca, İsveççe, Tayland Türkçesi, Ukraynaca, Fince) ve uyarlanabilirlik
- 7- API entegrasyonu ve hazır modüller
- 8- Konferans odası için davetiye ve doğrudan link gönderme
- 9- LDAP konektör
- 10- Uzak SOAP – tek oturum açma, entegrasyon ve uzaktan yönetim için ağ geçidi
- 11- Sistem moderasyonu, kullanıcı/organizasyon/moderasyon sistemi
- 12- Yedekleme ve dil modülü (Dil Editörü, Yedekleme Paneli)
- 13- Özel ve genel (sadece organizasyon için) konferans odaları
- 14- Kullanılan teknolojiler, teknoloji portföyünü görme.

3.3. OPENMEETINGS YAPISI



Şekil 3.3. Openmeetings'in yapısı [22].

3.3. OPENMEETINGS TERCİH EDİLMESİNİN NEDENLERİ

Çalışmada OpenMeetings yazılımının tercih edilme sebepleri şunlardır [22]:

- 1- Ücretsiz olması,
- 2- Moodle ile entegre çalışması,
- 3- Görüntüleri flash olarak yayınladığı için band genişliğini en uygun şekilde kullanması,
- 4- Dosya ve ekran paylaşma desteği,
- 5- Çoklu dil desteği,
- 6- Linux(CentOS) ve Microsoft tabanlı sunucularda sorunsuz çalışması ve taşınabilirliği,
- 7- E-posta desteğidir.

Öğrenci memnuniyeti, hangi uzaktan eğitim video dağıtım türü ile yapılırsa yapılsın yaklaşık olarak aynı olmaktadır. Algılanan video kalitesini, teknik problemleri ve akışı kontrol tercihlerini öğrenci davranışlarını belirleyen unsurlar arasında saymak mümkün olmaktadır [23].

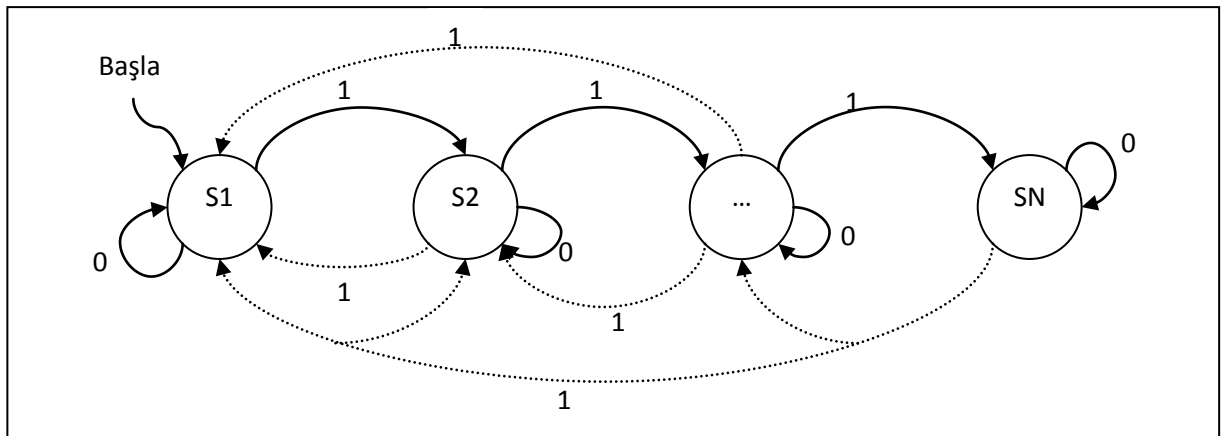
Tez ile doğrudan amaçlanan kullanıcı sayısını arttırmak ve artan kullanıcı sayısına karşın yaşanabilecek teknik problemleri önlemektir. Sistemden hizmet alabilecek kullanıcı sayısını arttırmanın dolaylı olarak mevcut kullanıcılar için algılanan hizmet kalitesini de arttırması doğal sonuçtur.

BÖLÜM 4

UYGULAMA

Geliştirilen yöntem sınıflandırma açısından tamamen yazılımsal yük dengeleme metodudur. N adet sunucuya sahip bir küme yapısı kurulmasının varsayıldığı ve iki adet sunucu ile de gerçekleştirilen yöntemin algoritmasının detaylarını sunmadan önce sağlanan çözüm sonlu durum makinesi ile ifade edilmiştir.

N adet sunucu ile kurulan kümede her hangi bir sunucu S_1 diye ifade edilsin ve bu sunucunun başlangıç durumu olduğunu varsayalım. Yazılım yeni eklenecek etkinlik öncesinde: kümede mevcut sunucunun yükünü tespit etmektedir. Verdiği sonuç durumlara giriş değeri olarak verilmektedir. Durum makinesinde S_1 durumuna giriş olarak 0 değeri: yük dengeleme algoritmasının S_1 durumu ile ifade edilmiş sunucunun etkinlik eşik değerini ve kullanıcı talep eşliğini aştığı zaman uygulanır. 1 Değeri ise sonlu durum makinesinin mevcut durumu ile ifade edilen sunucunun etkinlik eşliğini veya kullanıcı talep eşliğini aştığı durumda uygulanır.



Şekil 4.1. Yük dengeleme algoritmasının sonlu durum makinesi ile gösterimi [21].

Burada dikkat edilecek nokta S_1 durumuna giriş olarak 1 değeri uygulandığı zaman durum makinesi S_k diye ifade edilen, kümedeki mevcut sunuculardan yazılımın yük dağılımlarını tespiti sonucunda bulunduğu en uygun halde olan sunucuyu ifade eden, duruma geçiş yapmasıdır. Kümede boşta veya uygun durumda bulunan N adet sunucuya kadar bu işlem geçerlidir.

Burada herhangi bir duruma 1 değeri uygulanması sunucunun hedeflenen maksimum yüke ulaştığını gösterirken, bir sonraki durumda ifade edilen sunucu aslında sabit değildir. Çalışma zamanında en az oturma sahip olan veya üzerinde en az kullanıcı talebi bulunan sunucunun ifade edildiği durum makinenin geçeceği bir sonraki durumdur.

Eğer yazılım boşta ve uygun durumda bir sunucu bulamazsa, sunuculara aşırı yükü de en az yoğun durumdaki olandan en yoğununa doğru sıralayarak dağıtır. Bu geçişler durum makinesinde kesikli çizgilerle ifade edilmiştir.

4.1. GELİŞTİRİLEN YAZILIMIN TEKNİK ALTYAPISI

Öğrenim yönetim sistemi ile sanal sınıf uygulaması farklı sistemlerdir. Genelde farklı sunucularda çalışmaları tercih edilmektedir. Bu doğal bir yük dengeleme sağlamaktadır. Farklı gereksinimlere sahip oldukları için tercih edilen sunucu, işletim sistemi ve üzerinde çalışan hizmetlerde farklı olmaktadır.

Yazılımın çalışacağı alt yapı detaylandırılacak olursa, bağımsız sunucular yerine kümelenmiş sunucular kullanılmıştır. Gerçek sunucular performans ve verimliliği artırmak için sanallaştırılmıştır. Böylece pratik olarak dört çekirdekli ve 4 gigabyte belleğe sahip bir sunucudan iki sanal sunucu elde edilmiştir. Çalışmada bu şekilde bir yolun tercih edilmesinin nedeni n adet sunucu satın alındığında $2n$ adet sunucu ile sistemi yapılandırmanın mümkün olduğunu göstermektir. Böylece hem sanallaştırılmış hem de kümelenmiş bir ortam elde edilmiştir. Teknik detaylar şöyledir.

4.1.1. Gerçek Sunucu(ların)nun İşletim Sistemi

Gerçek sunucuda işletim sistemi olarak Xen sanal makine monitörü (Xen Hypervisor) kullanılmaktadır. Xen sanal makine monitörü x86, x86_64, IA64, ARM ve diğer işlemci mimarileri için güçlü, etkili ve güvenli özellikler kümesi sunan açık kaynak endüstri standart'ıdır. Windows®, Linux®, Solaris®, BSD işletim sistemleri çeşitlerini içeren misafir işletim sistemlerinin geniş bir aralığına destek vermektedir.

Xen sanal makine monitörü, işletim sistemi ile yer değiştirerek doğrudan bilgisayar donanımı üzerinde çalışan yazılım katmanıdır. Böylece bilgisayar donanımına aynı anda çoklu misafir işletim sistemi çalıştırma imkanı sunmaktadır. Xen.org topluluğu Xen sanal makine monitörünü GNU ücretsiz yazılım lisansı altında geliştirip sürdürmektedir [24].

4.1.2. Sanal Sunucuların İşletim Sistemi

CentOS Red Hat firmasının dağıtımı olan Red Hat Enterprise Linux (RHEL) kaynak kodları üzerine kurulu, ve bu dağıtım ile uyumlu bir linux dağıtımıdır. Bağımsız bir grup tarafından geliştirilen işletim sisteminin isminin açılımı The Community ENTerprise Operating System biçimindedir. İsminden de anlaşılacağı üzere açık kaynak enterprise sunucu yazılımıdır. Masaüstü işletim sistemlerinden farklı olarak sanallaştırma, apache, mysql, postgresql, ftp, dns, ldap gibi sunucu yazılımları paket içeriğine dahildir. Grafik arayüz ve dokümantasyon ile kolay kurulum ve kullanım kolaylığı sağlamaktadır.

CentOS işletim sisteminin tercih edilmesindeki gerekçeler şunlardır [25]:

- 1- Ücretsiz olması,
- 2- Stabil ve güvenli olması,
- 3- Geniş paket kütüphanesi,
- 4- Kullanıcı dostu grafik ara yüz desteği,
- 5- 2014 yılına kadar destek verilmesidir.

4.1.3. Öğretim Yönetim Sistemi Ve Sanal Sınıf Uygulaması

Web tabanlı uzaktan eğitim uygulamaları senkron, asenkron ve karma olarak yapılmaktadır. Öğrenci ve öğretmenin eş zamanlı olarak yaptığı ders senkron; öğrencinin öğretmenden, zamandan ve mekandan bağımsız olarak ders materyali ile öğrenme faaliyeti gerçekleştirmesi asenkron; ikisinin birlikte kullanıldığı sistemler karma olarak tanımlamak mümkündür. Uygulamalar özel yazılımlar ile yapılmaktadır [25]. Çalışmada özelliklerinin detaylarına girilmese de yazılım olarak Moodle kullanılmıştır.

Sanal sınıf Uygulaması olarak ise önceki bölümde detaylarından bahsedilmiş olan OpenMeetings yazılımı kullanılmıştır.

4.2. GELİŞTİRİLEN YAZILIMIN PSEUDECODE GÖSTERİMİ

Ek'te uygulamanın kaynak kodları sunulmuştur. Öğretim yönetim sistemine bir sanal sınıf etkinliği eklemek istendiğinde çalışan kodun pseudocode gösterimi ise aşağıda sunulduğu şekilde olmaktadır.

```
belirle etkinlik // aktif sunucu toplam aktif etkinlik sayısını
belirle sayi // aktif sunucu toplam öğrenci sayısını
if etkinlik >= eşik veya sayi >= kapasite then
    for her kayitli sunucu icin
        belirle etkinlik // aktif etkinlik sayısını
        belirle sayi // öğrenci sayısını
    for son
        sirala // sayı ve etkinliğe göre sunucuları sıralar
        ayarla aktif sunucu // en uygun sunucuyu aktif sunucu yap
if son
ekle etkinlik
```

Aktif olan bir etkinliğe tıklandığında çalışan kodlar için pseudocode gösterimi aşağıda sunulmuştur.

```
belirle sunucu kimligi // etkinlik tablosundan sunucu id belirle
belirle sunucu // konfigürasyon tablosundan sunucu ayarlarını seç
cagir yönlen(sunucu)
```

Sisteme eklenmiş bir sunucu sistemden kaldırılmak istendiğinde çalışan kodlar için pseudecode gösterimi de aşağıda sunulmuştur.

```
fonksiyon yuk()
fonksiyon yuk_dengele (x, y)
fonksiyon sirala(a)
fonksiyon dizi_sirala (a, b)
al sunucu id
cagir yuk()
cagir sirala(1)
if tanimli(sunucu id) then
    sil sunucu // istenen sunucu konfigürasyonunu sil
    belirle kullanılabilir sunucular
    for her kullanılabilir sunucu için
        kullanılabilir sunucular = cagir dizi_sirala(1, kullanılabilir
sunucular)
        kullanılabilir sunucular = cagir yuk_dengele(kullanılabilir
sunucular, sunucu id)
    for son
if son
don
```

4.3. KULLANILAN YÖNTEM

Yeni bir sanal sınıf etkinliği eklenmek istendiğinde, kümedeki düğümlerin yük dağılımlarına göre yeni etkinlik en uygun olan düğüme yönlendirir. Eğer az yüklenmiş veya yüksüz bir düğüm yoksa veya bütün düğümler aşırı yüklenmişse de mevcut durumda en az iş yükü olana talep yönlendirilmektedir.

Küme çalışmayı denemek için iki adet sunucu ile gerçekleştirilmiştir. Bu optimum ağ araçları ve optimum ağ trafiğine sahip sistemde beş yüz kullanıcıya sanal sınıf uygulamasında hizmet verilebilmesi anlamına gelir. Eğer küme beş ve daha çok sayıda sunucu ile oluşturulursa bu sistemin istenilen sayıya ve kaliteye ulaşmasını sağlar. Bağımsız veya kümelenmiş sunucular için ortak olan, sistemde ne kadar güçlü makineler kullanılırsa performansın o kadar artacağıdır. Kümeyi oluşturan sunucuların bağımsız sunucular kadar tampon bellekleri yüksek, işlemcileri güçlü olması gerekmez de daha güçlü sunucular kullanılabilirse alınan sonuç da iyileşecektir.

Burada sunucuların iş yüklerinin tespit edilmesi ve iş yüklerine göre taleplerin dağıtılması dikkate değer yöndür. Çalışmanın odağında da bu problem ve çözümü bulunmaktadır.

İş yükünü tespit etmek için öğrenim yönetim sisteminde aktif olan eşzamanlı ders etkinliklerinin sayısının tespiti gerekmektedir. Bağımsız sunucu ile sağlanan sanal sınıf uygulamasında aktif olsun, pasif olsun bütün etkinlikler bu sunucuya doğal olarak yönlendirilmiştir. Yük dengeleme ile kümedeki bilgisayarlara (sunuculara) yükün dağıtılması yani bu etkinliklerin yönlendirilmesi gerekmektedir. Sonra da hangi sunucuya ne kadar yük dağıldığının tespiti ve takibi gerekmektedir.

Kümemizin lider politikasına uyan yönleri bulunmaktadır. Lider politikasındaki lider düğümünü öğrenim yönetim sisteminin üzerinde çalıştığı uygulama sunucusunun temsil etmekte olduğu düşünülebilir.

Bağımsız sunucu ile sağlanan sanal sınıf uygulamasından farklı olarak lider düğüm diyebileceğimiz sunucu, öğrenim yönetim sistemindeki aktif sanal sınıf uygulaması etkinliklerini sayar ve Openmeetings sunucuları üzerindeki dağılımlarını tespit eder. Bu dağılıma göre sunucuları sıralar ve yeni gelen etkinlik talebini en uygun sunucuya yönlendirir.

i Adet sunucu ile oluşturulan kümede, k kullanıcı j Openmeetings etkinliğinin oluşturduğu talep şu şartlarda oluşturulur [10]:

$$\sum_{i=1}^m N_{current_{i,j}} + 1 \leq \sum_{i=1}^m N_{allocated_{i,j}} \quad (4.1)$$

m Sunucu sayısı, n etkinlik sayısı olarak tanımlandığında, $i = 1, \dots, m$ ve $j = 1, \dots, n$, yani her sunucu için:

$$\sum_{j=1}^n N_{current_{j,k}} + N_{current_{n+1,k}} \leq N_{capacity_i}. \quad (4.2)$$

Burada j etkinliğin her birinden k adet kullanıcı ile sunucu i den mevcut talep sayısı $N_{current_{i,j}}$, sunucu i den k adet kullanıcı ile j etkinliğin her birisi için tahsis edilmiş talep sayısı $N_{allocated_{i,j}}$, i sunucusunun hizmet verdiği kullanıcı sayısı $N_{current_{i,k}}$ ve maksimum kapasitesi de $N_{capacity_i}$ 'dir.

Sistemde her sunucu için ön görülmüş etkinlik sayısı: $N_{allocated_{i,j}} = 3$, ayarlanmışsa, yeni bir etkinlik talebi geldiğinde ilk sunucu için formül 4.1'de sunucu üzerindeki etkinlik sayısına bakılır ve ancak sayı üçten küçük veya eşitse etkinlik bu sunucuya eklenebilmektedir.

Yine sistemde herbir sunucu için maksimum kullanıcı sayısı: $N_{capacity_i} = 200$ olarak belirlenmişse, formül 4.2'de yeni gelen etkinlik talebi kullanıcı sayısı sunucunun hizmet verdiği toplam kullanıcı sayısına eklendiğinde $N_{capacity_i}$ 'yi aşmıyorsa o sunucuya eklenebilmektedir.

Bu kontroller sistemde tanımlı bütün sunucular için kontrol edilir ve en optimum sunucuya talep yönlendirilir. Böylece yük optimal olarak dağıtılmaktadır.

4.3. GELİŞTİRİLEN YAZILIMIN TANITIMI

Kümelenmiş sanal sınıf uygulaması için birden çok sunucu konfigürasyonun alınması, belirlenen kriterlere göre gelen taleplerin bu konfigürasyonlar arasında dağıtılması gerekmektedir.

Bir sanal sınıf sunucusuyla öğrenim yönetim sisteminin entegre olabilmesi için gereken parametreler Şekil 4.2’de de görülmektedir. Yazılım ile sağlanan yenilik; sistemde bulunan n adet sunucu için bu parametreleri alabilmesinin yanında o anki iş yüküne göre aktif sunucunun kayıtlarını da tutmasıdır. Aktif sunucu öğrenim yönetim sistemine yeni bir sanal sınıf etkinliği eklenmek istendiğinde iş yükünün en az olduğu önceden seçilmiş ve etkinliğin, yani yeni taleplerin yönlendirileceği sunucuya karşılık gelmektedir.

Aktif sunucu da sistemde kayıtlı konfigürasyonlardan sadece birisidir. Ayrıca sisteme kayıtlı bütün sunucu konfigürasyonları da Şekil 4.2’de görüldüğü üzere listelenmektedir. Burada listelenen konfigürasyonlarda değişiklik yapmak mümkündür.

You are logged in as Öğuzhan MENEMENÇİOĞLU (Logout)

Kümelemiş Sanal Sınıf Uygulama Platformu

KSSUP > Administration > Modules > Activities > OpenMeetings

Blocks editing on

OpenMeetings

Be careful modifying these settings - strange values could cause problems.

Aktif OpenMeetings/Red5 sunucu adresi (rtmphostfocalörnek: yourhost.com)	193.140.8.12
Aktif OpenMeetings/Red5 sunucu http Port (red5httpportörnek: 5080)	5080
Aktif OpenMeetings Yönetici Kullanıcı adı (OpenMeetings Usernameörnek: admin)	root
Aktif OpenMeetings Yönetici şifresi (OpenMeetings Userpassörnek: admin)	1q2w3e
Aktif Konferans Dili (OpenMeetings 1 ve 14 arasında değer)	1
Aktif Konferans Genişlik (1000)	1080
Aktif Konferans Yükseklik (640)	680
OpenMeetings/Red5 Her sunucu için kapasite (kapasiteörnek: 500)	250

OpenMeetings/Red5 sunucu adedi (adetörnek: 3)

OpenMeetings/Red5 her sunucu için oda (esikörnek: 3)

- Ana Sayfa
- Sunucu1
- Sunucu2
- Sistem Yüku

Site Administration

- Notifications
- Users
- Courses
- Grades
- Location
- Language
- Modules
- Activities
 - Manage activities
 - Assignment
 - Chat
 - Database
 - Forum
 - Glossary
 - OpenMeetings
 - Quiz
 - Resource
 - SCORM/AICC
- Blocks
- Filters
- Security
- Appearance
- Front Page
- Server
- Networking
- Reports
- Miscellaneous

Search

Admin bookmarks
bookmark this page

Şekil 4.2. Aktif sunucu ayarları penceresi.

Appearance

- Front Page
- Server
- Networking
- Reports
- Miscellaneous

Search

Admin bookmarks
bookmark this page

OpenMeetings/Red5 Her sunucu için kapasite (kapasiteörnek: 500)

OpenMeetings/Red5 sunucu adedi (adetörnek: 3)

OpenMeetings/Red5 her sunucu için oda (eşikörnek: 3)

- Ana Sayfa
- Sunucu1
- Sunucu2
- Sistem Yükü

Sunucu1

OpenMeetings/Red5 sunucu adresi 1 (rtmhostlocalörnek: yourhost.com)
OpenMeetings/Red5 sunucu http Port 1 (red5httpportörnek: 5080)
OpenMeetings Yönetici Kullanıcı adı 1 (OpenMeetings Kullanıcı adıörnek: admin)
OpenMeetings Yönetici şifresi 1 (OpenMeetings şifresiörnek: admin)
Konferans Dili 1 (OpenMeetings 1 ve 14 arası değer)
Konferans Genişliği 1 (1000)
Konferans Yüksekliği 1 (640)
OpenMeetings/Red5 her sunucu için talep kapasitesi 1 (kapasiteörnek: 500)

193.140.9.23
5080
root
1q2w3e
1
1280
680
200

Sunucu 1 Ayarlarını Sil

[[DeğişiklikleriKaydet]]

Moodle Docs for this page

You are logged in as Oğuzhan MENEMENÇİOĞLU (Logout)

Home

Şekil 4.3. Mevcut sunucular için konfigürasyon penceresi.

Kümeye yeni eklenecek sunucular için de konfigürasyon kaydı yapabilmek mümkündür. Kümeye yeni sunucu ilave etmek için sunucu adedi ComboBox'ından (açılır kutu) sunucu numarasını arttırmak yeterlidir. Açılır pencereden kümede sahip olunan sunucu sayısı bir arttırılır. Şekil 4.3' de gösterilen sekmelere bir sunucu için daha ayarların alınabileceği bir sekme eklenmiş olunur. İlgili ayarlar TextBox'lara (metin alanlarına) girildikten sonra Değişiklikleri Kaydet butonu tıklanarak sunucu ekleme işlemi tamamlanmış olmaktadır.

Aynı ComboBox'ın değeri azaltılmaya çalışılırsa, bu durum kontrol edilmektedir, bunun karşılığı konfigürasyon silme talebidir. Sistem buna karşılık "Sunucu konfigürasyonu silme işlemi yapılmak isteniyorsa Sunucu X Ayarlarını Sil butonu kullanılması gerekmektedir!" uyarısı vermektedir.

Konfigürasyon silmek için yukarıda bahsedildiği gibi Sunucu X Ayarlarını Sil butonu kullanılmaktadır.

Kümede bulunan yükü ve dağılımı görmek için ise sistem yükü linki tıklanmalıdır. Sistemdeki etkinlikleri, aktif olma durumlarını ve hangi sunucuya yönlendirildiklerini görmek mümkündür.

Sistemdeki sunucu sayısı değişmeden diğer ayarlarda değişiklik yapmak oldukça basittir. Ama eğer sistem yöneticisi belirli nedenlerle sistemdeki sunucu sayısını azaltmaya karar verirse sistemin bakması gereken bir çok faktör vardır. Öncelikle silinmek istenen sunucu üzerinde dağılmış yük olup olmadığı tespit edilmeli ve eğer yük varsa kalan sunuculara bu yük eşit oranda yeniden dağıtılmalıdır.

- Site Administration
 - Notifications
 - Users
 - Courses
 - Grades
 - Location
 - Language
 - Modules
 - Activities
 - Manage activities
 - Assignment
 - Chat
 - Database
 - Forum
 - Glossary
 - OpenMeetings
 - Quiz
 - Resource
 - SCORM/AICC
 - Blocks
 - Filters
 - Security
 - Appearance
 - Front Page
 - Server
 - Networking
 - Reports
 - Miscellaneous

Search

Admin bookmarks
bookmark this page

OpenMeetings

Be careful modifying these settings - strange values could cause problems.

Go Back

Load Balancer Results

Number	Name	Max.	User	Activity	Host IP	Host Port	Candidate	Request Number
1	105.1	50	1	1	193.140.9.23	5080		60
2	104.1	50	1	1	193.140.8.12	5080		90
3	103.1	50	1	1	193.140.8.12	5080		60
4	102.1	100	1	1	193.140.9.23	5080		150
5	101.1	50	1	1	193.140.9.23	5080		50

Şekil 4.6. Yük dağılımı.

BÖLÜM 5

SONUÇ VE ÖNERİLER

Bu çalışma ile bağımsız sunucu ile sınırlı sayıda kullanıcıya hizmet verebilen sanal sınıf uygulamasının hizmet verebildiği sayının arttırabileceği gösterilmiştir. Bu model ile kurulan yapıda sanal sınıf uygulamasının optimum ağ koşullarında beş yüz kullanıcıya kesintisiz hizmet vermesi sağlanmıştır. Sunuculardaki iş yükü: etkinlik ve kullanıcı sayıları izlenebilmektedir. İş yükünün dağılımı yük dengeleme politikasının doğru çalıştığını göstermektedir.

Bağımsız sunucu üzerinde çalışan sanal sınıf uygulamasının yerine bu çalışmada önerilen modeli test etmek için bir gerçek sunucu üzerinde iki sanal sunucu ile oluşturulmuş bir küme kullanılmıştır. Bu küme sınırlı olmasına rağmen geliştirilen uygulama ile yük dengeleme yapılabildiğini göstermesi açısından yeterlidir.

Bu kümede iki sunucu arasında yük dengeleme yaparken kullanılan dengeleme politikası Lider, Hepsini Birer Kez Dene ve LRU Politikalarının modifikasyonudur. Önerilen yöntemde etkinlik sayısı eşik değeri ve kullanıcı sayısı eşik değeri esas alınmıştır.

Modelde, yukarıda detaylandırılan politikalardan, Hepsini Birer Kez Dene'nin güçlü yanı kullanılarak her bir sunucuya eşit iş yükü dağılması sağlanmıştır. Bunu sağlamak için kontrol edilen etkinlik sayısı eşiğinin aşıp aşılmadığıdır. Bu noktada modelin eksik kalabileceği yön, talep edilen etkinliklerin kullanıcı sayılarının farklı olmasıdır. Yani eşit yük dağıtılmış gibi görünürken esasen iş büyüklüğü nedeniyle bunun sağlanamamasıdır. Model bu eksikliğini aşmak için, LRU Politikasının güçlü yanı olan, iş büyüklüğünü dikkate almaktadır. İş büyüklüklerine ise kullanıcı sayısı eşiği kontrol edilerek bakılmaktadır. Modelde karşılaşılabilecek bir handikap ise lider politikasından kaynaklanabilir. Çünkü bütün taleplerin tek bir düğümden

dağıtılması aşırı yoğunluk anında bir tıkanıklığa neden olabilir.

Geliştirilen uygulamanın uzaktan eğitim kapsamında bin ve üzerinde kullanıcıya hizmet verecek yapılar tasarlanırken, alt yapı çalışması esnasında büyük maliyetler ile satın alınacak ticari yazılımların yerine kullanılabilmesi ve hedeflenen sayıya göre oluşturulacak küme ile başarımların sağlanabileceği öngörülmektedir.

Ayrıca bu çalışmaya referansla yapılacak sonraki çalışmalarda, burada uygulanan modelin zayıf yönlerinin güçlendirilebileceği ve daha ölçeklendirilebilir çalıştırılacağı öngörülmektedir. Bu çerçevede iki yol izlemek mümkündür. Birinci yol olarak çalışmada kullanılan lider politikası değiştirilmeden daha iyi bir model önerilmelidir. Lider düğüm olarak öğrenim yönetim sistemi, Moodle sunucusu yerine, Openmeetings sunucusunu tercih etmek izlenebilecek kuvvetli alternatif olarak düşünmek mümkündür.

Diğer yol için de eğer lider politikasının terk edilmesi düşünülecekse öğrenim yönetim sistemi ve web konferans yazılımlarının ayrı iki yazılım olduğu unutulmalıdır. İki ayrı sistemin entegre çalıştırıldığı ve geliştirilecek yük dengeleme politikasının bu bağlamda çalışması gerektiği hesaplanarak yöntem geliştirilmelidir. Bu şartlar altında modelin kullandığı politika yerine daha optimum politikaların geliştirilebileceği de öngörülmektedir.

Bin kişiye hizmet verebilecek bir ticari sanal sınıf uygulamasını lisansı ile birlikte satın almak için ortalama fiyat 10,000\$ ile 100,000\$ arasında değişmektedir. Sunucu ve diğer ağ donanımlarının masrafı buna dâhil değildir. Açık kaynak kodlu bir yazılım için de sunucu ve ağ donanımları gerektiğinden, değerlendirmede bunları kapsam dışında tutabilmek mümkündür. Uygulama hizmet olarak satın alındığında (Hosted Server) ise ortalama yıllık 3,000\$ civarında bir bedel ödenmektedir.

Geliştirilen uygulama açık kaynak kodlu bir uygulamadır. Edilen tek masraf sunucu ve ağ donanımları için yapılanlardır. Geliştirilen uygulama ile yukarıda belirtilen miktarların kurumun bünyesine kalması ve başka ihtiyaçlara kanalize edilmesini sağlamak mümkün hale gelmektedir.

KAYNAKLAR

1. İnternet: Beowulf.org “Beowulf Project Overview”
<http://www.beowulf.org/overview/index.html> (2010)
2. Gropp, W., Lusk, E. and Sterling, T., “Beowulf cluster computing with linux, 2nd ed.”, *The MIT Press*, London, 1-21 (2003).
3. Lascu, O., Donovan, T., Hiller, S., McAleer, S., Niesz, L., Saunders, S., Wu, K. and Zutenis, P., “Configuring and managing AIX clusters using xCAT 2, 1st ed.”, *IBM*, 3-18 (2009).
4. Huang, B., “Network performance studies in high performance computing environments”, Master of Science Thesis, *Faculty of Graduate Studies, The University of Western Ontario*, London, Ontario, Canada, 4-7 (2004).
5. Baker, M. and Buyya, R. “Cluster computing: the commodity supercomputer”, *Softw. Pract. Exper.*, 29(6): 551–576 (1999).
6. İnternet: “Top500 Supercomputer Sites”,
<http://www.top500.org/overtime/list/36/archtype> (2010).
7. İnternet: “Top500 Supercomputer Sites”,
<http://www.top500.org/stats/list/36/osfam> (2010).
8. İnternet: “Top500 Supercomputer Sites”, <http://www.top500.org/stats/list/36/os> (2010).
9. İnternet: “Top500 Supercomputer Sites”,
<http://www.top500.org/charts/list/36/os> (2010).
10. Zhang, Z. and Fan, W., “Web server load balancing: a queueing analysis”, *European Journal of Operational Research*, 186: 681–693 (2008).
11. Tekin, V., N., “İstatistiğe Giriş, 2. Baskı”, *Seçkin Yayıncılık*, 183-187 (2008).
12. Bourke, T., “Server load balancing”, *O'Reilly & Associates*, Sebastopol, 1-8, 24-30 (2001).
13. Kopperapu, C., “Load balancing servers, firewalls, and caches”, *John Wiley & Sons, Inc.*, New York, USA, 7-24 (2001).

14. Mundur, P. and Arankalle, P., “Optimal server allocations for streaming multimedia applications on the Internet”, *Computer Networks*, 50: 3608–3621 (2006).
15. Adams, D., A., “Optimal load balancing in a beowulf cluster”, Master of Science Thesis, *Worcester Polytechnic Institute*, Worcester, Mass., USA, 5-7 (2005).
16. Internet: “Valdosta State University”, http://pride.valdosta.edu/Whitepaper_Distance_Learning.pdf (2011).
17. Reushle, S. and Loch, B., “Conducting a trial of web conferencing software: why, how, and perceptions from the coalface”, *Turkish Online Journal of Distance Education*, 9 (3) : 19-28 (2008).
18. Mayrhofer, D., Back, A. and Hubschmid, R., “Web-conferencing software tools”, *Institute of Information Management of the University of St. Gallen*, Switzerland, 3-7 (2004).
19. Synnes, K., Söderström, T. and Parnes, P., “Learning in desktop video-conferencing environments”, *Proceedings of WebNet 2001*, Orlando, Florida, USA, 1-6 (2001).
20. Suduc, A., Bîzo1, M. and Filip, F., G., “Exploring multimedia web conferencing”, *Informatica Economică*, 13 (3): 5-17 (2009).
21. Şen, B., Menemencioğlu, O., Atasoy, F. ve Özcan, C., “Kümelenmiş sanal sınıf uygulaması”, *Akademik Bilişim 2011*, Malatya, 1-8 (2011).
22. İnternet: “OpenMeetings, Open-Source Web-Conferencing”, <http://code.google.com/p/openmeetings/> (2011).
23. Reisslein, J., Seeling, P. and Reisslein, M., “Video in distance education: ITFS vs. web-streaming: evaluation of student attitudes”, *Internet and Higher Education*, 8: 25-44 (2005).
24. İnternet: “Xen Hypervisor”, <http://www.xen.org/> (2011).
25. Şen, B., Atasoy, F. ve Aydın N., “Düşük maliyetli web tabanlı uzaktan eğitim sistemi uygulaması”, *Akademik Bilişim 2010*, Muğla, 1-12 (2010).

EK AÇIKLAMALAR A

UYGULAMANIN KAYNAK KODLARI

UYGULAMANIN KAYNAK KODLARI

```
<?php //mod_form.php Modified 12.11.2010 21:43:31 Oguzhan
MENEMENCIOGLU
$CFG;
$ip;
//if (($count >= $threshold) || ($user_number['result'] >= ($CFG-
>openmeetings_capacity)))
{
$prefix = $CFG->prefix; $prefix = $prefix.config;
$sql = @mysql_query("select distinct id, name, value from $prefix where name like
'%openmeetings_red5host%'");
$array = array();
for ($i = 0; $i < @mysql_num_rows($sql); $i++)
{
$temp = @mysql_fetch_array($sql);
if ($temp['name'] != 'openmeetings_red5host')
{
unset($temporay); $temporay = array();
$temp_id = $temp['id'];
echo $i.'. Id: '.$temp_id.'<br \>';
$temp_value = $temp['value'];
// 0 IP
array_push($temporay, $temp['value']);
//echo $i.'. Id: '.$id.'<br \>';
$prefix = $CFG->prefix; $prefix = $prefix.config;
$plug = substr($temp['name'], 21, 1); echo '. plug : '.$plug.'<br \>';
$name = "openmeetings_red5port".$plug; echo $i.'. port name: '.$name.' ';
$q = "select value from $prefix where name = '$name'"; echo '. sql : '.$sql.'<br \>';
$array_peripheral = @mysql_fetch_array(@mysql_query($q));
// 1 PORT
array_push($temporay, $array_peripheral['value']);
```

```

$name = "openmeetings_openmeetingsAdminUser".$plug; echo $i.'. name: '.$name.'
';
$array_peripheral = @mysql_fetch_array(@mysql_query("select value from $prefix
where name = '$name'"));
// 2 USER
array_push($temporay, $array_peripheral['value']);
$name = "openmeetings_openmeetingsAdminUserPass".$plug; echo $i.'. name:
'.$name.' ';
$array_peripheral = @mysql_fetch_array(@mysql_query("select value from $prefix
where name = '$name'"));
// 3 PASS
array_push($temporay, $array_peripheral['value']);
$name = "openmeetings_openmeetingsLanguage".$plug; echo $i.'. name: '.$name.' ';
$array_peripheral = @mysql_fetch_array(@mysql_query("select value from $prefix
where name = '$name'"));
// 4 LANGUAGE
array_push($temporay, $array_peripheral['value']);
$name = "openmeetings_openmeetingsiFrameWidth".$plug; echo $i.'. name:
'.$name.' ';
$array_peripheral = @mysql_fetch_array(@mysql_query("select value from $prefix
where name = '$name'"));
// 5 WIDTH
array_push($temporay, $array_peripheral['value']);
$name = "openmeetings_openmeetingsiFrameHeight".$plug; echo $i.'. name:
'.$name.' ';
$array_peripheral = @mysql_fetch_array(@mysql_query("select value from $prefix
where name = '$name'"));
// 6 HEIGHT
array_push($temporay, $array_peripheral['value']);
$name = "openmeetings_capacity".$plug; echo $i.'. name: '.$name.' ';
$array_peripheral = @mysql_fetch_array(@mysql_query("select value from $prefix
where name = '$name'"));
// 7 CAPACITY

```



```

array_push($temporay, $array_peripheral['value']);
$prefix = $CFG->prefix; $prefix = $prefix.openmeetings; $prefix_ = $CFG->prefix;
$prefix_ = $prefix_.course_modules;
$q = "
select count(*) as c from $prefix
where
(server_id = '$temp_id') and
(
(id, course) in      (select instance, course from $prefix_ where visible = '1')
)
"; echo $q.<br \>';
$array_count = @mysql_fetch_array(@mysql_query($q));
echo $i.'. number of meeting: '.$array_count['c'].'<br \>';
// 8 MEETING NUMBER
array_push($temporay, $array_count['c']);
// 9 SERVER ID
array_push($temporay, $temp['id']);
$prefix = $CFG->prefix;
$q = "
SELECT count(distinct (u.id)) as result
FROM ".$prefix.".user u
JOIN ".$prefix.".role_assignments ra ON u.id = ra.userid
JOIN ".$prefix.".role r ON r.id = ra.roleid
JOIN ".$prefix.".context c ON ra.contextid = c.id
JOIN ".$prefix.".course co ON c.instanceid = co.id
JOIN ".$prefix.".openmeetings om ON co.id = om.course
JOIN ".$prefix.".course_modules cm ON om.id = cm.instance and om.course =
cm.course
WHERE
r.id in (3, 4, 5) and
om.server_id = '$temp_id' and
cm.visible = '1'
";

```

```

// and (assesstimestart > ".time()." or assesstimeend < ".time().")
echo $q.<br />';
$array_count = @mysql_fetch_array(@mysql_query($q));
// 10 USER NUMBER
array_push($temporay, $array_count['result']);
array_push($array, $temporay);
unset($temporay);
}
}
echo '<pre>';
print_r($array);
echo '</pre>';
for ($i = count($array) - 1; $i >= 0; $i--)
{
$swapped = false;
for ($j = 0; $j < $i; $j++)
{
if ($array[$j][10] > $array[$j + 1][10])
{
for ($k = 0; $k < count($array[$j]); $k++)
{
$tmp = $array[$j][$k];
$array[$j][$k] = $array[$j + 1][$k];
$array[$j + 1][$k] = $tmp;
}
$swapped = true;
}
}
if (!$swapped) break;
}
echo '<pre>';
print_r($array);
echo '</pre>';

```

```

$prefix = $CFG->prefix; $prefix = $prefix.config;
$sql = @mysql_fetch_array(@mysql_query("select value from $prefix where id =
".$array[0][9].""));
$sql = $sql[0];
$sql = "update $prefix set value = ".$sql." where name = 'openmeetings_red5host'";
if (!(mysql_query($sql)))
echo ' Load Balancer, host update has become unsuccessful!./n';
$names = array( "openmeetings_red5port",
"openmeetings_openmeetingsAdminUser",
"openmeetings_openmeetingsAdminUserPass",
"openmeetings_openmeetingsLanguage",
"openmeetings_openmeetingsiFrameWidth",
"openmeetings_openmeetingsiFrameHeight",
"openmeetings_capacity");
for ($i = 0; $i < 7; $i++)
{
$sql = "update $prefix set value = ".$array[0][$i + 1]."" where name =
".$names[$i].""";
if (!(mysql_query($sql)))
echo ' Load Balancer, '.$names[$i].' update has become unsuccessful!./n';
}
}
$ip = $array[0][9];
require_once ('moodleform_mod.php');
class mod_openmeetings_mod_form extends moodleform_mod {
function definition() {
$mform =& $this->_form;
//-----
/// Adding the "general" fieldset, where all the common settings are showed
$mform->addElement('header', 'general', get_string('general', 'form'));
/// Adding the standard "name" field
$mform->addElement('text', 'name', get_string('Room Name', 'openmeetings'),
array('size'=>'64'));

```

```

$mform->setType('name', PARAM_TEXT);
$mform->addRule('name', null, 'required', null, 'client');
// Adding the "Room Type" field
$mform->addElement('select', 'type', get_string('Room Type', 'openmeetings'),
array('1'=>'Conference Room', '2'=>'Audience Room', '3'=>'Restricted Room',
'0'=>'Show Recording'));
// Some description
$mform->addElement('static', 'description_room_recording_id', get_string('The
Recording Field is only used if Room Type is Recording', 'openmeetings'), null);
// Adding the "Available Recordings to Shows" field
// $recordings = array('1'=>'Recording 1', '2'=>'Recording 2');
$recordings = array();
$openmeetings_gateway = new openmeetings_gateway();
if ($openmeetings_gateway->openmeetings_loginuser()) {
$recordingsArray = $openmeetings_gateway-
>openmeetings_getRecordingsByExternalRooms();
foreach ($recordingsArray as $key => $value) {
//there is a bug, if a List has the length of 1 the type is wrong
if (is_array($value)) {
//echo "Das Element " . $value["flvRecordingId"] . " enthŠlt den Wert: " .
$value["fileName"] . "<br>";
$recordings[$value["flvRecordingId"]] = $value["fileName"];
} else {
//echo "Das Element " . $recordingsArray["flvRecordingId"] . " enthŠlt den Wert: " .
$recordingsArray["fileName"] . "<br>";
$recordings[$recordingsArray["flvRecordingId"]] = $recordingsArray["fileName"];
break;
}
}
}
$mform->addElement('select', 'room_recording_id', get_string('Available Recordings
to Shows', 'openmeetings'), $recordings);
// Adding the "Number of Participants" field

```

```

$mform->addElement('select', 'max_user', get_string('Max User', 'openmeetings'),
array('2'=>'2', '4'=>'4', '8'=>'8', '16'=>'16', '24'=>'24', '36'=>'36', '50'=>'50',
'100'=>'100', '200'=>'200', '500'=>'500', '1000'=>'1000'));
// Adding the "Is Moderated Room" field
$mform->addElement('select', 'is_moderated_room', get_string('Wait for teacher',
'openmeetings'), array('1'=>'Participants need to wait till the teacher enters the
room', '2' => 'Participants can already start (first User in Room becomes
Moderator)'));
// Adding the "Room Language" field
$language_array = array ('1' => 'english',
    '2' => 'deutsch',
    '3' => 'french',
    '4' => 'italian',
    '5' => 'portugues',
    '6' => 'portugues brazil',
    '7' => 'spanish',
    '8' => 'russian',
    '9' => 'swedish',
    '10' => 'chinese simplified',
    '11' => 'chinese traditional',
    '12' => 'korean',
    '13' => 'arabic',
    '14' => 'japanese',
    '15' => 'indonesian',
    '16' => 'hungarian',
    '17' => 'turkish',
    '18' => 'ukrainian',
    '19' => 'thai',
    '20' => 'persian',
    '21' => 'czech',
    '22' => 'galician',
    '23' => 'finnish',
    '24' => 'polish',

```

```

        '25' => 'greek',
        '26' => 'dutch',
        '27' => 'hebrew');
$mform->addElement('select', 'language', get_string('Room Language',
'openmeetings'), $language_array);
// Adding the optional "intro" and "introformat" pair of fields
$mform->addElement('htmleditor', 'intro', get_string('Comment', 'openmeetings'));
$mform->setType('intro', PARAM_RAW);
//$mform->addRule('intro', get_string('required'), 'required', null, 'client');
$mform->setHelpButton('intro', array('writing', 'richtext'), false, 'editorhelpbutton');
$mform->addElement('format', 'introformat', get_string('format'));
$mform->addElement('date_time_selector', 'assesstimestart', get_string('Beginning'));
$mform->addRule('assesstimestart', null, 'required', null, 'client');
$mform->addElement('date_time_selector', 'assesstimeend', get_string('Ending'));
$mform->addRule('assesstimeend', null, 'required', null, 'client');
$mform->addElement('hidden', 'server_id', $ip);
//-----
// add standard elements, common to all modules
$this->standard_coursemodule_elements(array('groups'=>true, 'groupings'=>true,
'groupmembersonly'=>true));
//-----
// add standard buttons, common to all modules
$this->add_action_buttons();
}
}
?>

<?php //openmeetings.php Modified 12.11.2010 21:43:31 Oguzhan
MENEMENCIOGLU Exp
$openmeetings_red5host;
$openmeetings_red5port;
$openmeetings_openmeetingsAdminUser;
$openmeetings_openmeetingsAdminUserPass;

```

```

if (isset($_SESSION['instance']))
{
/*echo '<script type="text/javascript">alert("Isset true!");</script>';*/
$instance = $_SESSION['instance'];
//echo $instance;
$prefix = $CFG->prefix; $prefix = $prefix.openmeetings;
$stemp = @mysql_fetch_array(@mysql_query("select server_id from $prefix where
id = '$instance'"));
$stemp = $stemp['server_id'];
//echo '<br \>ip: '.$stemp;
$prefix = $CFG->prefix; $prefix = $prefix.config;
$stemp = @mysql_fetch_array(@mysql_query("select name, value from $prefix
where id = '$stemp'"));
$openmeetings_red5host = $stemp['value'];
$i = substr($stemp['name'], 21, 1);
//echo '<br \>x: '.$name;
$name = "openmeetings_red5port".$i;
$stemp = @mysql_fetch_array(@mysql_query("select value from $prefix where
name = '$name'"));
$openmeetings_red5port = $stemp['value'];
$name = "openmeetings_openmeetingsAdminUser".$i;
$stemp = @mysql_fetch_array(@mysql_query("select value from $prefix where
name = '$name'"));
$openmeetings_openmeetingsAdminUser = $stemp['value'];
$name = "openmeetings_openmeetingsAdminUserPass".$i;
$stemp = @mysql_fetch_array(@mysql_query("select value from $prefix where
name = '$name'"));
$openmeetings_openmeetingsAdminUserPass = $stemp['value'];
unset($_SESSION['instance']);
}
else
{
/*echo '<script type="text/javascript">alert("Isset false!");</script>';*/

```

```

$openmeetings_red5host = $CFG->openmeetings_red5host;
$openmeetings_red5port = $CFG->openmeetings_red5port;
$openmeetings_openmeetingsAdminUser = $CFG-
>openmeetings_openmeetingsAdminUser;
$openmeetings_openmeetingsAdminUserPass = $CFG-
>openmeetings_openmeetingsAdminUserPass;
}
require_once($CFG->dirroot.'/mod/openmeetings/lib/nusoap.php');
//require_once($CFG->dirroot.'/lib/soaplib.php');

class openmeetings_gateway {

var $session_id = "";

/**
 * Change server
 *
function change_active_server($id)
{
$CFG;
$prefix = $CFG->prefix; $prefix = $prefix.config;
$temp = @mysql_fetch_array(@mysql_query("select value from $prefix where id =
$id"));
$openmeetings_red5host = $temp['value'];
return $openmeetings_red5host;
}
*/

/**
 * TODO: Get Error Service and show detailed Error Message
 */
function openmeetings_loginuser() {
$USER, $CFG;

```



```

$client_userService = new
nusoap_client_om("http://".$openmeetings_red5host.":".$openmeetings_red5port."/o
penmeetings/services/UserService?wsdl", "wsdl");
$client_userService->setUseCurl(true);
//echo "Client initd". "<br/>";
$error = $client_userService->getError();
if ($error) {
echo '<h2>Constructor error</h2><pre>' . $error . '</pre>';
echo '<h2>Debug</h2><pre>' . htmlspecialchars($client->getDebug(),
ENT_QUOTES) . '</pre>';
exit();
}
$result = $client_userService->call('getSession');
if ($client_userService->fault) {
echo '<h2>Fault (Expect - The request contains an invalid SOAP body)</h2><pre>';
print_r($result); echo '</pre>';
} else {
$error = $client_userService->getError();
if ($error) {
echo '<h2>Error</h2><pre>' . $error . '</pre>';
} else {
//echo '<h2>Result</h2><pre>'; print_r($result); echo '</pre>';
$this->session_id = $result["return"]["session_id"];
//echo '<h2>Result</h2><pre>'; printf(); echo '</pre>';
$params = array(
'SID' => $this->session_id,
'username' => $openmeetings_openmeetingsAdminUser,
'userpass' => $openmeetings_openmeetingsAdminUserPass
);
//$params = array();
$result = $client_userService->call('loginUser',$params);
//echo '<h2>Params</h2><pre>'; print_r($params); echo '</pre>';
if ($client_userService->fault) {

```

```

echo '<h2>Fault (Expect - The request contains an invalid SOAP body)</h2><pre>';
print_r($result); echo '</pre>';
} else {
    $err = $client_userService->getError();
    if ($err) {
        echo '<h2>Error</h2><pre>' . $err . '</pre>';
    } else {
        //echo '<h2>Result</h2><pre>'; print_r($result); echo '</pre>';
        $returnValue = $result["return"];
        //echo '<h2>returnValue</h2><pre>'; printf($returnValue); echo '</pre>';
    }
}
}
}
}
//echo 'returnValue: '.$returnValue;
if ($returnValue>0){
return true;
} else {
return false;
}
}
}

/**
 * TODO: Check Error handling
 *
 * @deprecated this method is deprecated
 *
 */
function openmeetings_createroom($openmeetings,$roomtypes_id)
{
    $course_name = 'MOODLE_COURSE_ID_'.$openmeetings->course.'_NAME_'.$openmeetings->name;

```

```

$client_roomService = new
nusoap_client_om("http://".$openmeetings_red5host."/".$openmeetings_red5port."/o
penmeetings/services/RoomService?wsdl", true);
Serr = $client_roomService->getError();
if (Serr) {
echo '<h2>Constructor error</h2><pre>' . Serr . '</pre>';
echo '<h2>Debug</h2><pre>' . htmlspecialchars($client->getDebug(),
ENT_QUOTES) . '</pre>';
exit();
}
$params = array(
'SID' => $this->session_id,
'name' => $course_name,
'roomtypes_id' => $roomtypes_id,
'comment' => 'Created by SOAP-Gateway for Moodle Platform',
'numberOfPartizipants' => 4,
'ispublic' => true,
'videoPodWidth' => 270,
'videoPodHeight' => 280,
'videoPodXPosition' => 2,
'videoPodYPosition' => 2,
'moderationPanelXPosition' => 400,
'showWhiteBoard' => true,
'whiteBoardPanelXPosition' => 276,
'whiteBoardPanelYPosition' => 2,
'whiteBoardPanelHeight' => 592,
'whiteBoardPanelWidth' => 660,
'showFilesPanel' => true,
'filesPanelXPosition' => 2,
'filesPanelYPosition' => 284,
'filesPanelHeight' => 310,
'filesPanelWidth' => 270
);

```

```

$result = $client_roomService->call('addRoom',$params);
if ($client_roomService->fault) {
echo '<h2>Fault (Expect - The request contains an invalid SOAP body)</h2><pre>';
print_r($result); echo '</pre>';
} else {
$result = $client_roomService->getError();
if ($result) {
echo '<h2>Error</h2><pre>' . $result . '</pre>';
} else {
//echo '<h2>Result</h2><pre>'; print_r($result["return"]); echo '</pre>';
return $result["return"];
}
}
return -1;
}

function openmeetings_createroomwithmod($openmeetings) {
$USER, $CFG;
$course_name = 'MOODLE_COURSE_ID_'. $openmeetings->course.'_NAME_'. $openmeetings->name;
//echo "CourseName: ".$course_name."<br/>";
//echo $client_userService."<br/>";
$client_roomService = new
nusoap_client_om("http://". $openmeetings_red5host.":". $openmeetings_red5port. "/o
penmeetings/services/RoomService?wsdl", true);
$result = $client_roomService->getError();
if ($result) {
echo '<h2>Constructor error</h2><pre>' . $result . '</pre>';
echo '<h2>Debug</h2><pre>' . htmlspecialchars($client->getDebug(),
ENT_QUOTES) . '</pre>';
exit();
}
$isModeratedRoom = false;

```

```

if ($openmeetings->is_moderated_room == 1) {
    $isModeratedRoom = true;
}
$params = array(
    'SID' => $this->session_id,
    'name' => $course_name,
    'roomtypes_id' => $openmeetings->type,
    'comment' => 'Created by SOAP-Gateway for Moodle Platform',
    'numberOfPartizipants' => $openmeetings->max_user,
    'ispublic' => true,
    'appointment' => false,
    'isDemoRoom' => false,
    'demoTime' => 0,
    'isModeratedRoom' => $isModeratedRoom
);
$result = $client_roomService->call('addRoomWithModeration',$params);
if ($client_roomService->fault) {
    echo '<h2>Fault (Expect - The request contains an invalid SOAP body)</h2><pre>';
    print_r($result); echo '</pre>';
} else {
    $err = $client_roomService->getError();
    if ($err) {
        echo '<h2>Error</h2><pre>' . $err . '</pre>';
    } else {
        //echo '<h2>Result</h2><pre>'; print_r($result["return"]); echo '</pre>';
        return $result["return"];
    }
}
return -1;
}

function openmeetings_createRoomWithModAndType($openmeetings) {
    $USER, $CFG;

```

```

$course_name = 'MOODLE_COURSE_ID_'. $openmeetings-
>course.'_NAME_'. $openmeetings->name;
//echo "CourseName: ".$course_name."<br/>";
//echo $client_userService."<br/>";
$client_roomService = new
nusoap_client_om("http://". $openmeetings_red5host.":". $openmeetings_red5port."/o
penmeetings/services/RoomService?wsdl", true);
$err = $client_roomService->getError();
if ($err) {
echo '<h2>Constructor error</h2><pre>' . $err . '</pre>';
echo '<h2>Debug</h2><pre>' . htmlspecialchars($client->getDebug(),
ENT_QUOTES) . '</pre>';
exit();
}
$isModeratedRoom = false;
if ($openmeetings->is_moderated_room == 1) {
$isModeratedRoom = true;
}
$params = array(
'SID' => $this->session_id,
'name' => $course_name,
'roomtypes_id' => $openmeetings->type,
'comment' => 'Created by SOAP-Gateway for Moodle Platform',
'numberOfPartizipants' => $openmeetings->max_user,
'ispublic' => true,
'appointment' => false,
'isDemoRoom' => false,
'demoTime' => 0,
'isModeratedRoom' => $isModeratedRoom,
'externalRoomType' => 'moodle'
);
$result = $client_roomService-
>call('addRoomWithModerationAndExternalType',$params);

```

```

if ($client_roomService->fault) {
echo '<h2>Fault (Expect - The request contains an invalid SOAP body)</h2><pre>';
print_r($result); echo '</pre>';
} else {
$err = $client_roomService->getError();
if ($err) {
echo '<h2>Error</h2><pre>' . $err . '</pre>';
} else {
//echo '<h2>Result</h2><pre>'; print_r($result["return"]); echo '</pre>';
return $result["return"];
}
}
return -1;
}

function openmeetings_getRecordingsByExternalRooms(){
$USER, $CFG;
$client_roomService = new
nusoap_client_om("http://".$openmeetings_red5host.":".$openmeetings_red5port."/o
penmeetings/services/RoomService?wsdl", true);
$err = $client_roomService->getError();
if ($err) {
echo '<h2>Constructor error</h2><pre>' . $err . '</pre>';
echo '<h2>Debug</h2><pre>' . htmlspecialchars($client->getDebug(),
ENT_QUOTES) . '</pre>';
exit();
}
$params = array(
'SID' => $this->session_id,
'externalRoomType' => 'moodle'
);
//We prefer the List ?!
$result = $client_roomService-
>call('getFlvRecordingByExternalRoomTypeByList',$params);

```

```

if ($client_roomService->fault) {
echo '<h2>Fault (Expect - The request contains an invalid SOAP body)</h2><pre>';
print_r($result); echo '</pre>';
} else {
$err = $client_roomService->getError();
if ($err) {
echo '<h2>Error</h2><pre>' . $err . '</pre>';
} else {
//echo '<h2>Result</h2><pre>'; print_r($result); echo '</pre>';
return $result["return"];
}
}
return -1;
}

/*
* Usage if this Method will work if you have no need to simulate always the same
user in
* OpenMeetings, if you want to do this check the next method that also remembers
the
* ID of the external User
*/

function openmeetings_setUserObject($username, $firstname, $lastname,
$profilePictureUrl, $email) {
$USER, $CFG;
$client_userService = new
nusoap_client_om("http://".$sopenmeetings_red5host.":".$sopenmeetings_red5port."/o
penmeetings/services/UserService?wsdl", true);

$err = $client_userService->getError();
if ($err) {
echo '<h2>Constructor error</h2><pre>' . $err . '</pre>';
}
}

```



```

echo '<h2>Debug</h2><pre>' . htmlspecialchars($client->getDebug(),
ENT_QUOTES) . '</pre>';
exit();
}
$params = array(
'SID' => $this->session_id,
'username' => $username,
'firstname' => $firstname,
'lastname' => $lastname,
'profilePictureUrl' => $profilePictureUrl,
'email' => $email
);
$result = $client_userService->call('setUserObject',$params);
if ($client_roomService->fault) {
echo '<h2>Fault (Expect - The request contains an invalid SOAP body)</h2><pre>';
print_r($result); echo '</pre>';
} else {
$error = $client_userService->getError();
if ($error) {
echo '<h2>Error</h2><pre>' . $error . '</pre>';
} else {
//echo '<h2>Result</h2><pre>'; print_r($result["return"]); echo '</pre>';
return $result["return"];
}
}
return -1;
}
/**
* Sets the User Id and remembers the User,
* the value for $systemType is any Flag but usually should always be the same,
* it only has a reason if you have more then one external Systems, so the $userId will
not
* be unique, then you can use the $systemType to give each system its own scope

```

```

* so a unique external user is always the pair of: $userId + $systemType
* in this case the $systemType is 'moodle'
*/
function openmeetings_setUserObjectWithExternalUser($username, $firstname,
$lastname,
$profilePictureUrl, $email, $userId, $systemType) {
$USER, $CFG;
$client_userService = new
nusoap_client_om("http://".$openmeetings_red5host.":".$openmeetings_red5port."/o
penmeetings/services/UserService?wsdl", true);
$error = $client_userService->getError();
if ($error) {
echo '<h2>Constructor error</h2><pre>' . $error . '</pre>';
echo '<h2>Debug</h2><pre>' . htmlspecialchars($client->getDebug(),
ENT_QUOTES) . '</pre>';
exit();
}
$params = array(
'SID' => $this->session_id,
'username' => $username,
'firstname' => $firstname,
'lastname' => $lastname,
'profilePictureUrl' => $profilePictureUrl,
'email' => $email,
'externalUserId' => $userId,
'externalUserType' => $systemType
);
$result = $client_userService->call('setUserObjectWithExternalUser',$params);
if ($client_roomService->fault) {
echo '<h2>Fault (Expect - The request contains an invalid SOAP body)</h2><pre>';
print_r($result); echo '</pre>';
} else {
$error = $client_userService->getError();

```

```

if ($err) {
echo '<h2>Error</h2><pre>' . $err . '</pre>';
} else {
//echo '<h2>Result</h2><pre>'; print_r($result["return"]); echo '</pre>';
return $result["return"];
}
}
return -1;
}

function openmeetings_setUserObjectAndGenerateRoomHashByURL($username,
$firstname, $lastname,
$profilePictureUrl, $email, $userId, $systemType, $room_id, $becomeModerator) {
$USER, $CFG;
$client_userService = new
nusoap_client_om("http://".$openmeetings_red5host].":".$openmeetings_red5port."/
openmeetings/services/UserService?wsdl", true);
$err = $client_userService->getError();
if ($err) {
echo '<h2>Constructor error</h2><pre>' . $err . '</pre>';
echo '<h2>Debug</h2><pre>' . htmlspecialchars($client->getDebug(),
ENT_QUOTES) . '</pre>';
exit();
}
$params = array(
'SID' => $this->session_id,
'username' => $username,
'firstname' => $firstname,
'lastname' => $lastname,
'profilePictureUrl' => $profilePictureUrl,
'email' => $email,
'externalUserId' => $userId,
'externalUserType' => $systemType,
'room_id' => $room_id,

```

```

'becomeModeratorAsInt' => $becomeModerator,
'showAudioVideoTestAsInt' => 1
);
$result = $client_userService-
>call('setUserObjectAndGenerateRoomHashByURL',$params);
if ($client_roomService->fault) {
echo '<h2>Fault (Expect - The request contains an invalid SOAP body)</h2><pre>';
print_r($result); echo '</pre>';
} else {
$result = $client_userService->getError();
if ($result) {
echo '<h2>Error</h2><pre>' . $result . '</pre>';
} else {
//echo '<h2>Result</h2><pre>'; print_r($result["return"]); echo '</pre>';
//echo 'return<br>';
return $result["return"];
}
}
return -1;
}

function
openmeetings_setUserObjectAndGenerateRecordingHashByURL($username,
$firstname, $lastname,
                $userId, $systemType, $recording_id) {
$result = $client_userService->call('setUserObjectAndGenerateRecordingHashByURL',
                $username, $firstname, $lastname, $userId, $systemType, $recording_id);
print_r($result);
$result = $client_userService->getError();
if ($result) {
echo '<h2>Constructor error</h2><pre>' . $result . '</pre>';
}
}

```

```

echo '<h2>Debug</h2><pre>' . htmlspecialchars($client->getDebug(),
ENT_QUOTES) . '</pre>';
exit();
}
$params = array(
'SID' => $this->session_id,
'username' => $username,
'firstname' => $firstname,
'lastname' => $lastname,
'externalUserId' => $userId,
'externalUserType' => $systemType,
'recording_id' => $recording_id
);
$result = $client_userService-
>call('setUserObjectAndGenerateRecordingHashByURL',$params);
if ($client_roomService->fault) {
echo '<h2>Fault (Expect - The request contains an invalid SOAP body)</h2><pre>';
print_r($result); echo '</pre>';
} else {
$error = $client_userService->getError();
if ($error) {
echo '<h2>Error</h2><pre>' . $error . '</pre>';
} else {
//echo '<h2>Result</h2><pre>'; print_r($result["return"]); echo '</pre>';
return $result["return"];
}
}
return -1;
}
}
?>

```

<?php // \$Id:delete.php, v 1.0 25.11.2010 11:35:00 Oguzhan MENEMENCIOGLU

Exp \$

```
require_once(".././config.php");
```

```
$CFG;
```

```
$value = array();
```

```
$list = array();
```

```
$deleted_value = array();
```

```
$pure_value = array();
```

```
$available_server_list = array();
```

```
function load()
```

```
{
```

```
$prefix = $CFG->prefix; $prefix = $prefix.openmeetings;
```

```
$array = @mysql_query("select id, course, name, max_user, server_id from  
$prefix");
```

```
for ($i = 0; $i < @mysql_num_rows($array); $i++)
```

```
{
```

```
$temp = @mysql_fetch_array($array);
```

```
$activity = $temp['id'];
```

```
$course = $temp['course'];
```

```
$server_id = $temp['server_id'];
```

```
$temp = $temp['id'];
```

```
$prefix = $CFG->prefix;
```

```
$sql = "
```

```
SELECT count(distinct (u.id)) as user_count
```

```
FROM ".$prefix."user u
```

```
JOIN ".$prefix."role_assignments ra ON u.id = ra.userid
```

```
JOIN ".$prefix."role r ON r.id = ra.roleid
```

```
JOIN ".$prefix."context c ON ra.contextid = c.id
```

```
JOIN ".$prefix."course co ON c.instanceid = co.id
```

```
JOIN ".$prefix."openmeetings om ON co.id = om.course
```

```
JOIN ".$prefix."course_modules cm ON om.id = cm.instance and om.course =  
cm.course
```

```

WHERE
r.id in (3, 4, 5) and
co.id = ".$course." and
om.server_id = ".$server_id." and
cm.visible = '1'
";
$user_count = @mysql_fetch_array(@mysql_query($sql));
$prefix = $CFG->prefix; $prefix = $prefix.config;
$temp = @mysql_fetch_array(@mysql_query("select name, value from $prefix
where id = '$server_id'"));
$name_id = substr($temp['name'], 21, 1);
$temp = array($server_id, 1, $user_count['user_count'], $name_id, $activity);
array_push($value, $temp);
}
for ($i = 0; $i < count($value); $i++)
{
$temp = array($value[$i][0], $value[$i][1], $value[$i][2], $value[$i][3]);
if ($i == 0)
array_push($list, $temp);
else
{
$decision = true;
for ($k = 0; $k < count($list); $k++)
{
if ($list[$k][0] == $value[$i][0])
{
$list[$k][1] += $value[$i][1];
$list[$k][2] += $value[$i][2];
//echo $list[$k][2].'\<br \>';
$decision = false;
break;
}
}
}
}
}

```

```

if ($decision == true)
{
array_push($list, $temp);
}
}
//echo $value[$i][0].' '.$value[$i][1].' '.$value[$i][2].' '.$value[$i][3].'
'.$value[$i][4].'<br \>';
}
for ($k = 0; $k < count($list); $k++)
{
$temp = $list[$k][0];
$prefix = $CFG->prefix;
$sql = "
select count(distinct (id)) as activity_count from ".$prefix."openmeetings
where
(server_id = '$temp') and
(
(id, course) in      (select instance, course from ".$prefix."course_modules where
visible = '1')
)
";
$activity_count = @mysql_fetch_array(@mysql_query($sql));
$list[$k][1] = $activity_count['activity_count'];
}
}
function refresh_load_status($get, $server)
{
for ($i = 0; $i < count($value); $i++)
{
//echo $value[$i][0].' '.$server.'<br \>';
if ($value[$i][0] == $server)
for ($j = 0; $j < count($get); $j++)
if ($get[$j][0] == $server)

```



```

for ($k = 0; $k < count($list); $k++)
{
if ($list[$k][0] == $server)
{
$list[$k][1] -= $value[$i][1];
$list[$k][2] -= $value[$i][2];
break;
}
}
}
for ($i = 0; $i < count($list); $i++)
for ($j = 0; $j < count($get); $j++)
if ($list[$i][0] == $get[$j][0])
{
$get[$j][1] = $list[$i][1];
$get[$j][2] = $list[$i][2];
}
$temp1 = array_sort_common(1, $get);
$temp2 = array_sort_common(2, $get);
//$list = array_sort_common(1, $list);
if (($temp1[0][1] <= $temp2[0][1]) && ($temp1[0][2] <= $temp2[0][2]))
return $temp1;
else if (($temp1[0][1] <= $temp2[0][1]) && ($temp2[0][2] <= $temp1[0][2]))
return $temp2;
else if (($temp2[0][1] <= $temp1[0][1]) && ($temp2[0][2] <= $temp1[0][2]))
return $temp2;
else
return $temp1;
}
function array_sort($sort_field)
{
for ($i = count($list) - 1; $i >= 0; $i--)
{

```

```

$swapped = false;
for ($j = 0; $j < $i; $j++)
{
if ($list[$j][$sort_field] > $list[$j + 1][$sort_field])
{
for ($k = 0; $k < count($list[$j]); $k++)
{
$tmp = $list[$j][$k];
$list[$j][$k] = $list[$j + 1][$k];
$list[$j + 1][$k] = $tmp;
}
$swapped = true;
}
}
if (!$swapped) break;
}
}
function array_sort_common($sort_field, $array_sorted)
{
for ($i = count($array_sorted) - 1; $i >= 0; $i--)
{
$swapped = false;
for ($j = 0; $j < $i; $j++)
{
if ($array_sorted[$j][$sort_field] > $array_sorted[$j + 1][$sort_field])
{
for ($k = 0; $k < count($array_sorted[$j]); $k++)
{
$tmp = $array_sorted[$j][$k];
$array_sorted[$j][$k] = $array_sorted[$j + 1][$k];
$array_sorted[$j + 1][$k] = $tmp;
}
$swapped = true;
}
}
}
}

```

```

}
}
if (!$swapped) break;
}
return $array_sorted;
}
$count = $CFG->openmeetings_red5number;
if ($count == 1)
echo '<script type="text/javascript">alert("You have only one server configuration. So
you cannot delete server configuration via this way!.."); window.location=
"../../admin/module.php?module=openmeetings";</script>';
load();
array_sort(1);
if (isset($_GET['name_id']))
{
$name_id = $_GET['name_id'] + 1;
$prefix = $CFG->prefix;
$name = 'openmeetings_red5host'.$name_id;
$ddeleted = @mysql_fetch_array(@mysql_query("select * from ".$prefix."config
where name = '$name'"));
$sql = "delete from ".$prefix."config where name = '$name'";
{
$q = array(
"delete from ".$prefix."config where name = 'openmeetings_red5port'.$name_id.",
"delete from ".$prefix."config where name =
'openmeetings_openmeetingsAdminUser'.$name_id.",
"delete from ".$prefix."config where name =
'openmeetings_openmeetingsAdminUserPass'.$name_id.",
"delete from ".$prefix."config where name =
'openmeetings_openmeetingsLanguage'.$name_id.",
"delete from ".$prefix."config where name =
'openmeetings_openmeetingsiFrameWidth'.$name_id.",

```

```

"delete from ".$prefix."config where name =
'openmeetings_openmeetingsiFrameHeight'".$name_id.""",
"delete from ".$prefix."config where name = 'openmeetings_capacity'".$name_id.""");
$count--;
echo $count;
$sql = "update ".$prefix."config set value = $count where name =
'openmeetings_red5number'";
echo $sql.'<br \>';
$count = $CFG->openmeetings_red5number;
$count;
$sql = @mysql_query("select * from ".$prefix."config where (name like
'%openmeetings_red5host%') and (name != 'openmeetings_red5host')");
for ($i = 0; $i < mysql_num_rows($sql); $i++)
{
$result = @mysql_fetch_array($sql);
array_push($available_server_list, array($result['id'], 0, 0, 0));
}
$available_server_list = array_sort_common(2, $available_server_list);
$active = @mysql_fetch_array(@mysql_query("select * from ".$prefix."config
where (name = 'openmeetings_red5host')"));
//echo 'deleted: '.$deleted['value'].' active: '.$active['value'].'<br \>';
if ($deleted['value'] == $active['value'])
{
echo "Active server configuration will be deleted!..".<br \>;
$t = @mysql_fetch_array(@mysql_query("select name from ".$prefix."config where
(id = ".$available_server_list[0][0].")"));
$i = substr($t['name'], 21, 1);
$t = @mysql_fetch_array(@mysql_query("select value from ".$prefix."config where
name = 'openmeetings_red5host'".$i."""));
$q = "update ".$prefix."config set value = ".$t['value']."' where name =
'openmeetings_red5host'";
mysql_query($q);

```

```

$t = @mysql_fetch_array(@mysql_query("select value from ".$prefix."config where
name = 'openmeetings_red5port'".$i.""));
$q = "update ".$prefix."config set value = ".$t['value']."' where name =
'openmeetings_red5port'";
mysql_query($q);
$t = @mysql_fetch_array(@mysql_query("select value from ".$prefix."config where
name = 'openmeetings_openmeetingsAdminUser'".$i.""));
$q = "update ".$prefix."config set value = ".$t['value']."' where name =
'openmeetings_openmeetingsAdminUser'";
mysql_query($q);
$t = @mysql_fetch_array(@mysql_query("select value from ".$prefix."config where
name = 'openmeetings_openmeetingsAdminUserPass'".$i.""));
$q = "update ".$prefix."config set value = ".$t['value']."' where name =
'openmeetings_openmeetingsAdminUserPass'";
mysql_query($q);
$t = @mysql_fetch_array(@mysql_query("select value from ".$prefix."config where
name = 'openmeetings_openmeetingsLanguage'".$i.""));
$q = "update ".$prefix."config set value = ".$t['value']."' where name =
'openmeetings_openmeetingsLanguage'";
mysql_query($q);
$t = @mysql_fetch_array(@mysql_query("select value from ".$prefix."config where
name = 'openmeetings_openmeetingsiFrameWidth'".$i.""));
$q = "update ".$prefix."config set value = ".$t['value']."' where name =
'openmeetings_openmeetingsiFrameWidth'";
mysql_query($q);
$t = @mysql_fetch_array(@mysql_query("select value from ".$prefix."config where
name = 'openmeetings_openmeetingsiFrameHeight'".$i.""));
$q = "update ".$prefix."config set value = ".$t['value']."' where name =
'openmeetings_openmeetingsiFrameHeight'";
mysql_query($q);
$t = @mysql_fetch_array(@mysql_query("select value from ".$prefix."config where
name = 'openmeetings_capacity'".$i.""));

```

```

$q = "update ".$prefix."config set value = ".$st['value']." where name =
'openmeetings_capacity';
mysql_query($q);
}
for ($i = 0; $i < count($list); $i++)
if ($deleted['id'] == $list[$i][0])
$available_server_list = refresh_load_status($available_server_list, $deleted['id']);
echo '<pre>';
print_r($available_server_list);
echo '</pre>';
for ($i = 0; $i < count($value); $i++)
{
echo 'deleted: '.$deleted['id'].'value: '.$value[$i][0].'available_server_list:
'.$available_server_list[0][0].'\<br \>';
if ($deleted['id'] == $value[$i][0])
{
if ($available_server_list[0][0] != $deleted['id'])
{
$q = "update ".$prefix."openmeetings set server_id =
".$available_server_list[0][0]." where id = ".$value[$i][4]."";
mysql_query($q);
//$sql = "update ".$prefix."openmeetings set server_id =
".$available_server_list[0][0]." where id = ".$value[$i][4]."";
$available_server_list[0][1]++;
}
else
{
if (count($available_server_list) == 1)
{
$q = "update ".$prefix."openmeetings set server_id =
".$available_server_list[0][0]." where id = ".$value[$i][4]."";
mysql_query($q);

```

```

//$sql = "update ".$prefix."openmeetings set server_id =
".$available_server_list[0][0]." where id = ".$value[$i][4].""";
$available_server_list[0][1]++;
}
else
{
$q = "update ".$prefix."openmeetings set server_id =
".$available_server_list[1][0]." where id = ".$value[$i][4].""";
mysql_query($q);
//$sql = "update ".$prefix."openmeetings set server_id =
".$available_server_list[1][0]." where id = ".$value[$i][4].""";
$available_server_list[1][1]++;
}
}
$available_server_list = array_sort_common(1, $available_server_list);
}
}
$sql = @mysql_query("select * from ".$prefix."config where (name like
'%openmeetings_red5host%') and (name != 'openmeetings_red5host') order by
name");
for ($i = 0; $i < mysql_num_rows($sql); $i++)
{
$result = @mysql_fetch_array($sql);
if ($deleted['id'] == $result['id'])
{
for ($j = $i + 1; $j < mysql_num_rows($sql); $j++)
{
$name_id = substr($result['name'], 21, 1);
$result = @mysql_fetch_array($sql);
$temp = substr($result['name'], 21, 1);
$q = array(
"update ".$prefix."config set name = 'openmeetings_red5host".$name_id.'" where id
= ".$result['id'],

```

```

"update ".$prefix."config set name = 'openmeetings_red5port".$name_id.'" where
name = 'openmeetings_red5port".$temp.'";",
"update ".$prefix."config set name =
'openmeetings_openmeetingsAdminUser".$name_id.'" where id = $temp",
"update ".$prefix."config set name =
'openmeetings_openmeetingsAdminUserPass".$name_id.'" where id = $temp",
"update ".$prefix."config set name =
'openmeetings_openmeetingsLanguage".$name_id.'" where id = $temp",
"update ".$prefix."config set name =
'openmeetings_openmeetingsiFrameWidth".$name_id.'" where id = $temp",
"update ".$prefix."config set name =
'openmeetings_openmeetingsiFrameHeight".$name_id.'" where id = $temp",
"update ".$prefix."config set name = 'openmeetings_capacity".$name_id.'" where id
= $temp");
for ($k = 0; $k < count($q);$k++)
mysql_query($q[$k]);
}
}
}
}
}
}
else
//echo '<script type="text/javascript">alert("Alert! Deleting Server Configuration
Process Has Not Completed..!"); window.location=
"../../admin/module.php?module=openmeetings";</script>';
?>

```


ÖZGEÇMİŞ

Oğuzhan MEMEMENCİOĞLU 1984 yılında Konya Ereğli’de doğdu. İlköğrenimini Karaman’da tamamladı. Karaman Anadolu Lisesinde başladığı orta öğretimini, Sivas Fen Lisesi’nde devam ettikten sonra bu okuldan ayrılarak yine Karaman’da Fatih Lisesi’nde tamamladı. 2002 yılında orta öğretimden mezun olduğu yıl Gazi Üniversitesi Endüstriyel Sanatlar Eğitim Fakültesi Bilgisayar Öğretmenliği Bölümünü kazanarak bu bölümde öğrenim hayatına devam etti. 2006 yılında lisans diploması almaya hak kazandı. 2007 yılında Karaman Sabiha Gökçen İlköğretim okulunda ve birkaç okulda iki yıl kadar Bilişim Teknolojileri Öğretmenliği yaptı. 2008 yılında Gazi Üniversitesi Bilişim Enstitüsü Bilgisayar Eğitimi Anabilim Dalında başladığı yüksek lisans eğitimini 2011 yılında Karabük Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında tamamladı. Karabük Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Yazılım Anabilim Dalında Araştırma Görevlisi olarak 2009 yılında başladığı görevini halen sürdürmektedir.

ADRES BİLGİLERİ

Adres: Karabük Üniversitesi
Mühendislik Fakültesi
Balıklarkayası Mevkii / KARABÜK

Tel: (533) 560 1396

E-posta: omenemencioglu@karabuk.edu.tr