



# Bulut Bilişimde Yük Dengeleme Mekanizmasının Analitik Modellemesi ve Performans Değerlendirmesi

*Araştırma Makalesi/Research Article*

 Yönel KIRSAL<sup>1</sup>\*,  Ersin ÇAĞLAR<sup>2</sup>

<sup>1</sup>Lefke Avrupa Üniversitesi, Elektrik ve Elektronik Mühendisliği, Lefke, Kuzey Kıbrıs TR-10 Mersin, Türkiye

<sup>2</sup>Lefke Avrupa Üniversitesi, Yönetim Bilişim Sistemleri, Lefke, Kuzey Kıbrıs TR-10 Mersin, Türkiye

[ykirsal@eul.edu.tr](mailto:ykirsal@eul.edu.tr), [ecaglar@eul.edu.tr](mailto:ecaglar@eul.edu.tr)

(Geliş/Received:19.04.2020; Kabul/Accepted:01.07.2021)

DOI: 10.17671/gazibtd.723411

**Özet**—Günümüzde, bilişim teknolojileri alanındaki hızlı gelişim, yeni bilgi işlem dizilerinin/paradigmalarının geliştirilmesine ve yayılmasına öncülük etmektedir. Bulut bilişim teknolojisi bu yeniliklerden sadece bir tanesidir. Bulut bilişim, internet tabanlı bir teknoloji olup diğer bulut kullanıcılarına donanım ve yazılım paylaşımı sağlayan bir teknolojidir. Bulut bilişim ve internet teknolojisinin hızlı gelişmesi ile tüm dünyada bulut bilişimin kullanıcılarında önemli bir artış olmuştur. Bu hızlı kullanıcı artışı, birçok sorunu da beraberinde getirmiştir. Bu sorunların en önemlilerinden bir tanesi yük dağılımı ve dengeleme mekanizmasıdır. Bu nedenle, bu çalışmada daha iyi sistem performans sonuçlarının elde edilebilmesi amacıyla analitik modeller farklı yük dengeleme mekanizmalarına uygulanmıştır. Performans değerlendirme için ortalama yük sayısı, ortalama tepki süresi ve engelleme olasılığı sonuçları sunulmuştur. Önerilen modellerin verimliliğini ve doğruluğunu göstermek için ise simülasyon oluşturulmuştur. Elde edilen sonuçların simülasyon sonuçları ile eşleştiği görülmektedir. Ek olarak, elde edilen sonuçlardan dinamik yük dengelemenin özellikle sistemin yoğun olduğu zamanlarda bulut bilişimdeki statik yük dengeleme mekanizmasından daha iyi performans gösterdiği açıkça görülmektedir.

**Anahtar Kelimeler**—bulut bilişim, yük dengeleme, performans analizi, kuyruk teorisi.

## Analytical Modelling and Evaluation of Load Balance Mechanism in Cloud Computing

**Abstract**—Recently, rapid improvement in the field of information technology has led to the development and deployment of new computing paradigms. Cloud computing is just one of these innovations. Cloud computing is an internet-based computing technology that shares hardware and software resources over the internet with other cloud users. With the rapid development of cloud computing and internet technology, there has been a significant increase in the number of users and become most popular technology all around the world. As rapidly increasing number of cloud users, many challenges have arisen. Load balancing mechanisms is one of the key challenges in cloud computing environment. Thus, in this paper, analytical models and evaluation of dynamic and static load balancing mechanisms are presented in order to obtain better performance results. Mean queue length, mean response time and blocking probability results are presented for performance evaluation. Simulation is also created in order to show the efficiency and accuracy of the proposed models. The obtained results show good agreement with the simulation results. In addition, it is clearly seen from the obtained results that dynamic load balancing outperforms to static load balancing mechanism in cloud computing especially in a loaded system.

**Keywords**—cloud computing, load balancing, performance evaluation, queueing theory.

## 1. GİRİŞ (INTRODUCTION)

Son yıllarda internet teknolojisindeki hızlı ilerleme, birçok bilişim sisteminin kullanılmasında büyük bir artışa yol açmıştır. İnternette meydana gelen ilerlemeler ile birlikte bulut bilişim teknolojisinin, şirketler ve internet kullanıcıları tarafından en fazla tercih edilen sistemlerin başında geldiği görülmektedir [1]. Bulut bilişim sistemlerinin bu kadar tercih edilip popüler olmasındaki en büyük etken kullanıcılarına sağladığı imkânlardır. Bu imkânların başında bilişim kaynaklarının diğer kullanıcı(lar) ile paylaşılması gelmektedir. Bilişim kaynaklarının başında ise veri tabanı, depolama alanı, sunucu ve yazılım gelmektedir [2,3]. Bu imkânlar sayesinde işletmeler bulut bilişim teknolojisini kullanarak yazılımlar, depolama alanları gibi ihtiyaçlarda maliyeti minimuma indirmişlerdir [4,5]. Bulut bilişim teknolojisinin sağladığı imkânlardan dolayı artan popülaritesinin yanında, bulut bilişim teknolojisi sağlayan şirketlerin karşılaştığı ciddi sorunlar da bulunmaktadır. Bu tehlike yaratan sorunların başında yük dengesi ve dağılımı gelmektedir [6]. Bulut bilişim sistemlerinin kullanıcı sayısı günden güne hızla artmaya devam etmektedir. Giderek artan kullanıcı sayısı ile birçok bulut bilişim sistemindeki sunucularında aşırı yüklenme oluşmaya başlamıştır. Oluşan bu aşırı yüklenmeden dolayı bulut bilişim sistemleri kullanıcılarına talep etmekte oldukları imkânlar ya gecikerek sunulmakta veya hiç sunulmamaktadır. Yük dengesi mekanizmaları, kullanıcılar ve bulut bilişim teknolojisi sağlayan şirketler açısından büyük bir önem arz etmektedir ve servisin sağlanmasında meydana gelebilecek gecikmeler ve/veya hiç servis sağlanamaması bir takım sorunlara yol açabilmektedir [7]. Yük dengeleme mekanizmaları sadece bulut bilişim sistemlerinde değil, diğer tüm bilişim sistemlerinde de servis kalitesini artıran önemli bir etkidir. Yük dengeleme, sunucu mimarilerinde artan yükü ve trafiği azaltabilmek için bir sunucu kümesinde bulunan sunucular arasındaki yükü paylaşma işlemidir [8]. Yük dengeleme mekanizmaları, bulut bilişim sistemlerine minimum yanıt süresi ile maksimum verim sağlama imkânı vermektedirler [9]. Verimden en iyi şekilde yararlanmayı, yanıt süresini azaltmayı ve herhangi bir kaynağın aşırı yüklenmesini engellemeyi hedefleyen yük dengeleme algoritmalarının bulut bilişim sistemleri performansları üzerinde önemli bir etkiye sahiptir. Yük dağılımının bulut sunucularını uzun bir süre meşgul durumda tutmasıyla kuyrukta servis bekleyen ve biriken birçok istek oluşmaktadır. Yük dengelenmesinde yaşanan bu sorun, bekleyen isteklerin artmasının yanında aşırı yüklenen sunucuların da arızalanıp servis dışı kalmasına sebep olmaktadır [10]. Yük dengeleme mekanizmaları statik ve dinamik olarak ikiye ayrılır. Statik yük dengeleme (SYD) stratejisi, homojen ve istikrarlı ortamlar için iyi sonuç verir. Ancak esnek değildir ve sistemin yapısındaki dinamik değişikliklere uyum sağlayamaz [11]. SYD stratejisinin aksine, dinamik yük dengeleme (DYD) mekanizmaları yük dağılımı yaparken yük ve sunucu koşullarını dikkate alır. Kısacası sisteme gelen yükler sabit şekilde sunuculara atanmamıştır ve sistemin yoğunluğuna göre değişiklik göstermektedirler [12].

Bulut bilişim servislerinde yükün dengelenememesinden dolayı ortaya çıkan sorunları önleyebilmek için araştırmacılar tarafından literatürde pek çok araştırma bulunmaktadır. Kuyruk teorisi, bulut bilişim sistemlerinin modellenmesinde ve analizinde etkin bir şekilde kullanılmaktadır. Kuyruk teorisi, matematiksel modellerde performans ölçütlerini değerlendirmek ve önerilen sistemin verimliliğini artırmak için sıklıkla kullanılmaktadır. Buna ek olarak, kuyruk teorisinin literatürde birçok uygulaması bulunmaktadır. Örneğin, [13]'da, önerilen sistemde tek sanal bilgisayar ele alınıp dinamik yük geçişlerinin analitik performans modeli değerlendirilmiştir. Yapılan çalışmada dinamik geçişin hizmet reddi senaryolarını ve toplam gecikmeyi azaltabileceği gösterilmiştir. Fetal ve diğerleri [14] bulut bilişimde isteğe bağlı hizmetlerin performansını optimize etmek için M/M/m kuyruk modelinin performansını incelemişlerdir. Ghosh ve diğerleri [15], büyük ölçekli sanallaştırılmış bulut sistemlerinin performans değerlerinin matematiksel çözümü için etkileşimli bir stokastik model yaklaşımı önermiştir. Sunet ve diğerleri [16], kuyruk modellerine dayalı olarak birden çok sanal bilgisayarın verimli geçişi için yeni bir teknik geliştirmişlerdir. Engelleme oranını değerlendirmek için, varış talebini M/M/C/C kuyruk teorisi kullanarak modellemiştirler. Benzer bir şekilde, Salahet ve diğerleri [17], bulutta barındırılan uygulamalar ve hizmetler için Markov zincirlerine dayalı analitik model önermişlerdir. Önerilen modelde, yanıt süresi, iş hacmi veya istek kaybı olasılığı gibi belirli bir hizmet seviyesi elde etmek için gereken sanal bilgisayar sayısını hesaplamışlardır. Fakat yukarıda bahsi geçen çalışmalarda DYD mekanizması ve/veya kesin sonuç veren matematiksel çözüm tekniği kullanılmamıştır. Buna ek olarak, bu çalışmada önerilen sistem diğer sistemlerden farklı olarak iki katman içeren açık kuyruk modeli kullanılarak analiz edilmiştir. Önerilen sistemin yapısında, sunucular içerisinden yük dağılımını yapan bir tane kontrol sunucusu ve işlemleri yapan birçok sanal sunucu bulunmaktadır. Kullanıcı istekleri kontrol sunucusu tarafından alınmakta ve ilk giren ilk çıkar esasına göre sunuculara aktarılmaktadır. Fakat diğer yük dengeleme mekanizmalarından farklı olarak kontrol sunucusu yeni gelen istekleri servis etmek için en az meşgul olan sunucuyu buluncaya kadar dinamik bir şekilde değişmektedir. Bu nedenle açık kuyruk modeli önerilen sistemin yapısını tam olarak aktardığı için kullanılmıştır. Literatürde, açık kuyruk modelleri, sadece bulut bilişim sistemleri ile sınırlı kalmayıp birçok araştırma da kullanılmıştır [15-17].

Ghafari ve diğerleri [18] yaptıkları çalışmada güç tüketimini dengelemek için yük dengeleme mekanizması geliştirmişlerdir. Geliştirilen modelde arı kolonisi algoritması ile birlikte yapay zekâ kullanılarak güç kullanımı dengelenmiştir. Pathak ve diğerleri [19] bulut sistemlerinde iki tane doğrusal olmayan yük dengeleme algoritmasını CloudSim simülasyon ortamında karşılaştırarak analiz etmişlerdir. Bunun yanında Yadav [20] da yaptığı çalışmada bulut sistemlerinde farklı yük dengeleme algoritmalarının tepki süresi ve servis süresi parametrelerine göre karşılaştırmıştır. Bir diğer araştırmada ise Nair ve diğerleri [21] CloudAnalyst

simülasyon yazılımı kullanarak farklı parametrelere göre yük dengeleme algoritmalarını analiz etmiş ve yük dağılımını en verimli yapan algoritmayı bulmayı amaçlanmıştır. Bu araştırmalar yanında literatürde birçok yük dengeleme algoritmasının farklı simülasyon yazılımları kullanılarak da analiz edildiği görülmektedir [22-25]. Bahsi geçen araştırmalarda temel olarak iki farklı analiz yapılmıştır. Birincisi, araştırmacıların önerdikleri yük dengeleme algoritmaları literatürde var olan algoritmalarla karşılaştırılmıştır. İkincisi ise araştırmacılar literatürde var olan yük dengeleme algoritmalarını bir birleriyle karşılaştırmışlardır. Yapılan her iki tür analizde de farklı servis kalite parametreleri kullanılmış ve yük dengelemede en verimli algoritmayı tespit etme amaçlanmıştır. Fakat bahsedilen analizler simülasyon yazılımları kullanılarak yapılmış analizlerdir. Bilindiği üzere matematiksel modellemeler ve analizler simülasyona göre daha hızlı performans değerlendirmesi için gerekli olan sonuçları elde etmektedir. Bu nedenle bulut bilişim sistemleri gibi çok servis sunucusuna ait sistemlerde matematiksel modeller sıklıkla literatürde kullanılmaktadır [26-32].

Bu çalışmada, daha iyi sistem performans sonuçları elde etmek için analitik modeller ve kesin çözüm taktiği sunulmaktadır. Statik ve dinamik yük dengeleme mekanizmalarının performans değerlendirilmesi kesin sonuç veren Spektral genişleme metodu ile hesaplanmıştır [27]. Bu çalışmada, SYD mekanizmasının literatürde kullanılan gibi ele alınmasına karşın DYD mekanizması literatürün dışında yeni bir önerme yapılarak ele alınmıştır. Yapılan önermede, kontrol sunucusu yeni gelen istekleri servis etmek için en uygun (boşta olan) sunucuyu buluncaya kadar dinamik bir şekilde değişmektedir. Diğer bir deyişle, önerilen modelde DYD mekanizması sisteme gelen yükleri kuyruk boyutu dolu olan sunucu, kuyruğu veya diğer sunucuların istekleri herhangi bir iletici alışverişi gerçekleştirmeden en uygun sunucuyu bularak servis etmektedir.

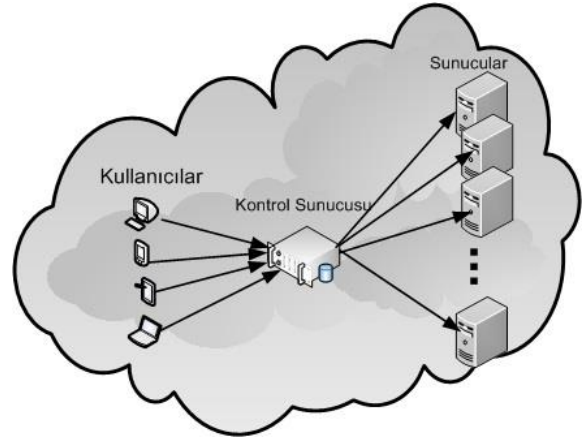
Çalışmanın geri kalanı aşağıdaki gibi organize edilmiştir: Sistem modeli Bölüm II'de verilmektedir. Bölüm III'te, önerilen sistemin matematiksel modellemesi ve kesin çözüm tekniği gösterilmiştir. Sayısal sonuçlar ve performans değerlendirmesi Bölüm IV'te sunulmaktadır. Son olarak, analiz sonuçları ve gelecek çalışmalar Bölüm V'te verilmektedir.

## 2. MODEL TANITIMI (MODEL DESCRIPTION)

Bu bölümde, bulut bilişim sistemlerinin yük dengeleme mekanizması ile sistemdeki sunucuların yük dağılımının performansını değerlendirmek için matematiksel modeller sunulmuştur. Önerilen sistem Şekil 1'de verilmiştir.

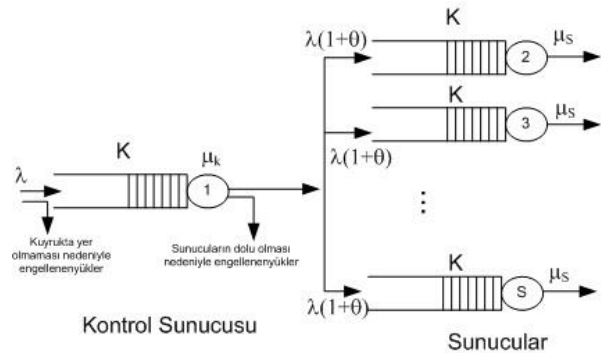
Şekil 1'de gösterildiği gibi, birçok sunucudan oluşan bulut sistemindeki büyük bir veri merkezi ele alınmıştır. Gerçek hayatta Google, Microsoft, Yahoo ve Amazon gibi büyük veri merkezleri çok sayıda sunucu içermektedir [9]. Bu sunucular içinden yük dağılımını yapan bir tane kontrol sunucusu bulunmaktadır. Kullanıcı istekleri kontrol

sunucusu tarafından alınmakta ve ilk giren ilk çıkar (FIFO) esasına göre sabit bir şekilde işlem görebilmesi adına sunuculara aktarılmaktadır. Bu çalışmada yük dağılımının performansını artırmak adına matematiksel model önerilmiştir.



Şekil 1. Önerilen sistem (Proposed model)

Bağımsız, rastgele gelen istekler ile birlikte kuyruk modeli, literatürde sunulan benzer çalışmalara örnek teşkil edecek şekilde sunulmuştur. Önerilen sistem Şekil 1'den de görüldüğü üzere iki katman içeren açık kuyruk modeli olarak modellenebilir ve bu yapı da Şekil 2'de detaylıca gösterilmiştir. Ardışık gelen istekler arasındaki varış süreleri bağımsızdır ve katlanarak  $1/\lambda$  ile dağıtılır. Diğer bir deyişle, kullanıcı istekleri ortalama varış oranına  $\lambda$  göre Poisson dağılımı ile sisteme katılmaktadır [1,6,13-17,26-32]. Kullanıcı istekleri sunucuların yoğunluğuna bakılmaksızın sabit bir şekilde sırasıyla sunuculara dağıtılır. Bu, SYD mekanizmasının çalışma prensibidir. Kontrol sunucusu ortalama  $1/\mu_k$  servis süresi ile servis edilir.

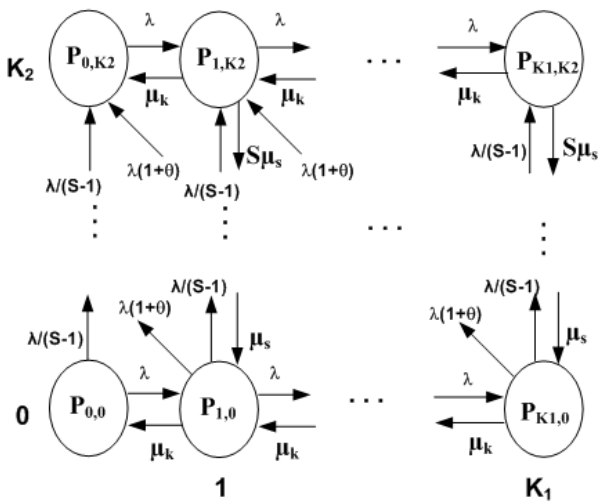


Şekil 2. Önerilen sistemin iki katman içeren açık kuyruk modeli (The proposed two dimensional open queuing system)

Kontrol sunucusu ve sunucular  $K$  kadar kullanıcı isteğini sisteme kabul etmektedir. Diğer bir deyişle  $K$ , kontrol sunucusunun ve sunucuların kuyruk kapasitesidir.  $S$  ise sunucu sayısıdır. Kontrol sunucusu yeni bir kullanıcı isteği geldiğinde, sıradaki kullanıcı isteklerinin sayısı,  $K$  eşliğinden fazla ise, yeni kullanıcı isteklerini engeller; aksi takdirde sisteme kabul eder. Diğer bir yandan sistemde  $S-1$  sunucu olduğunu varsayalım (1. sunucu kontrol sunucusu), kullanıcı istekleri kontrol sunucusu tarafından her sunucuya aynı olasılıkla,  $1/S-1$  ile eşit olarak dağıtılır.

Buna göre, her sunucuya kullanıcı isteklerinin gelmesi,  $\lambda/S-1$  varış oranıyla Poisson sürecini takip eder. Sistemdeki sunucuların homojen olduğunu ve her sunucu zamanının ortalama  $1/\mu_s$  servis süresi ile üstel bir dağılıma sahip olduğu varsayılmaktadır. Dolayısı ile sistemdeki sunucular M/M/K/L kuyruk modeline göre modellenebilir. Fakat kontrol sunucusu ve işlem sunucularının kuyruk modellerini ayrı ayrı çözmeyip iki katmanlı açık kuyruk matematiksel modeli uygulanmıştır. Dolayısıyla iki katman içeren açık kuyruk modeli kesin Spektral genişleme metodu kullanılarak analiz edilip performans sonuçları elde edilmiştir. Tüm sunucuların işleme kapasitesi ve boyutlarında eşit olduğu homojen bir bulut ortamında faaliyet gösterdiği farz edilmektedir. Her sunucu kuyruğunda, en fazla K kadar isteği alabilen bir kuyruk vardır. Gelen istek, eğer varsa boş kuyruk alanı olan bir sunucu kuyruğuna alınır. Kuyruk bekleyen tüm sunucuların dolması durumunda, varış talebi reddedilir. Sonuç olarak, bir kullanıcı talebi tüm kuyruklar dolu olduğundan veya yetersiz kontrol sunucusu kuyruğu nedeniyle engellenebilir.

Özet olarak, SYD mekanizmasında, kontrol sunucusu kullanıcılardan gelen istekleri alır ve daha sonra bunları sistemdeki sunuculara sırasıyla eşit olarak dağıtır. Bu çalışmada önerilen DYD modelinde ise, kontrol sunucusu sisteme gelen istekleri en uygun durumda olan sunucuya tüm sistemi ele alarak yönlendirmektedir. Şekil 2'de gösterildiği üzere sistemde S kadar sunucu bulunmaktadır. 1. sunucu kontrol sunucusu olduğundan, sistemde istekleri servis etmek için S-1 kadar sunucu bulunmaktadır (Şekil 2'de ikinci kısım, 2, 3,...S). S sunucusuna gelen yükler, kuyruğun dolu oluşundan dolayı, S-1 sunucusunun kuyruk sırasına başarılı şekilde yerleşebilir. Buna ek olarak S-1 kuyruğu da dolmuş ise sisteme gelen yeni istek S-2 sunucusunu veya S-3 vb. sunucularını servis alana kadar denemektedir. Eğer sisteme gelen istekler yer bulamaz ise sistemden engellenir. Önerilen DYD mekanizmasının Markov modeli Şekil 3'te gösterilmiştir.



Şekil 3. Dinamik yük dengeleme mekanizmasının geçiş diyagramı (The transition diagram of dynamic load balancing mechanism)

Statik yük dengeleme mekanizması her bağımsız sunucu için, basitçe bir M/M/1/K kuyruk modeli olarak ele

alınabilir. M/M/1/K kuyruk modeli literatürde bilinen ve sıklıkla kullanılan bir modeldir. DYD mekanizmasını modellerken ise sisteme gelişler SYD mekanizması temel alınarak yapılmıştır. Dolayısı ile DYD mekanizması, yük dağıtımları için  $\lambda(1+\theta)$  olarak hesaplanmıştır.  $\theta$  engelleme olasılığı olup  $P_K$  ya eşittir.  $\theta$  sistemde aşağıdaki gibi elde edilebilir.

$$\theta = P_K = \frac{\lambda(1-\theta)^K}{\mu} \frac{1 - \frac{\lambda(1-\theta)}{\mu}}{1 - \frac{\lambda(1-\theta)^{K+1}}{\mu}} \quad (1)$$

$\theta$ 'yı elde etmek için, (1) denklemini matematiksel modelde tekrar ederek (özyinelemeli metodu) çözebiliriz. Kullanıcı isteklerinin varış oranının Poisson varışının ve hizmet sürelerinin katlanarak dağıtıldığı varsayılmıştır. Bu varsayımlar her zaman geçerli olmayabilir [26,27]. Fakat Poisson gelişlerinin ve üstel hizmet süresinin literatürde kullanıldığını ve gerçek sistemlere yeterli şekilde yakınlaştırılmasını sağlayabildiğini göstermişlerdir.

Burada, yük dengeleme mekanizmalarının performansını değerlendirmek için üç önemli çıktı ele alınmıştır. Birincisi, isteklerin sistem tarafından engellenme olasılığı olan istek engelleme olasılığıdır. İkincisi, bir isteğin sunucuya gelmesinden sunucunun istekleri servis ettiği zamana kadar geçen ortalama istek tepki süresidir. Üçüncüsü de ortalama sistemin istek sayısıdır. Aşağıda, önerilen yük dengeleme mekanizmasının performans modelleri matematiksel olarak sunulmuştur.

### 3. ÖNERİLEN SİSTEMLERİN MATEMATİKSEL MODELLEMESİ (ANALYTICAL MODELLING OF PROPOSED SYSTEMS)

Bu bölümde, önerilen sistemin matematiksel modellemesi ve çözümü detaylı bir şekilde verilmiştir. Önerilen sistemin çözümü için kesin Spektral genişleme metodu kullanılmıştır [26,27]. Spektral genişleme metodu, her tür ağ performans, başarımların değerlendirilmesi ve süreklilik sistemlerinin çözümünde sıklıkla kullanılan bir çözüm tekniğidir. Pratik sistemlerden elde edilen belirli türdeki Markov modellerini çözer. Referanslar [26] ve [27] de kesin Spektral genişleme metodu hakkında daha fazla bilgi bulunabilir. Spektral genişleme metodu A, B ve C şeklinde üç ana matris ile ifade edilmektedir. Matris A anlık geçiş oranlarının, (i,j) durumundan (k,j) durumuna (soldan sağa) ve ana köşegen üzeri sıfır olarak tanımlanır. Bunlar modelin yalın yanal geçişleridir. Matrisler B ve C ise sırasıyla, yukarı doğru ve aşağı doğru bir adım geçiş matrisleridir. Şekil 3'te önerilen sistemin anlık geçiş değişkenleri gösterilmiştir. Şekil 3'te gösterildiği üzere matris A sadece  $\lambda$  ve  $\mu_k$  parametrelerine bağlıdır.  $K_1=1+K$  ve  $K_2=(S-1)+K$  kapasiteli bir sistemde geçiş matrisleri büyüklüğü  $(K_1+1) \times (K_2+1)$  dir. A ve  $A_j$  durum geçiş matrisi Şekil 4'te gösterilmiştir. Bununla birlikte matris B bir adım yukarı ve C bir adım aşağı geçiş oranı olarak tanımlandığından sırasıyla B ve  $B_j$ , Şekil 5'te C ve  $C_j$  durum geçiş matrisi Şekil 6, 7, 8'de gösterilmiştir.

$$A_j = \begin{bmatrix} 0 & \lambda & 0 & 0 & 0 & 0 & 0 \\ \mu_k & 0 & \lambda & 0 & 0 & 0 & 0 \\ 0 & \mu_k & 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & \mu_k & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & \mu_k & 0 & \lambda \\ 0 & 0 & 0 & 0 & 0 & \mu_k & 0 \end{bmatrix}$$

Şekil 4. A ve  $A_j$  durum geçiş matrisi (The state transition matrix of A and  $A_j$ )

$$B = B_j = \begin{bmatrix} \lambda/(S-1) & 0 & 0 & 0 & 0 \\ \lambda(1+\theta) & \lambda/(S-1) & 0 & 0 & 0 \\ 0 & \lambda(1+\theta) & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \lambda/(S-1) & 0 \\ 0 & 0 & 0 & \lambda(1+\theta) & \lambda/(S-1) \end{bmatrix}$$

Şekil 5. B ve  $B_j$  durum geçiş matrisi (The state transition matrix of B and  $B_j$ )

$$C = \begin{bmatrix} \mu_s & 0 & 0 & 0 & 0 & 0 \\ 0 & 2\mu_s & 0 & 0 & 0 & 0 \\ 0 & 0 & 3\mu_s & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (S-1)\mu_s & 0 \\ 0 & 0 & 0 & 0 & 0 & S\mu_s \end{bmatrix}$$

Şekil 6. C durum geçiş matrisi (The state transition matrix of C)

$$C_j = \begin{bmatrix} S\mu_s & 0 & 0 & 0 & 0 & 0 \\ 0 & S\mu_s & 0 & 0 & 0 & 0 \\ 0 & 0 & S\mu_s & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & S\mu_s & 0 \\ 0 & 0 & 0 & 0 & 0 & S\mu_s \end{bmatrix}$$

Şekil 7.  $j > S$  için  $C_j$  durum geçiş matrisi (The state transition matrix of  $C_j$  when  $j > S$ )

$$C_j = \begin{bmatrix} \min(1, j)\mu_s & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & \min(S-1, j)\mu_s & 0 \\ 0 & 0 & 0 & \min(S, j)\mu_s \end{bmatrix}$$

Şekil 8.  $j \leq S$  için  $C_j$  durum geçiş matrisi (The state transition matrix of  $C_j$  when  $j \leq S$ )

Çözüm metodu matrisler elde edildikten sonra aşağıdaki anlık geçişler ile gelişir. Hem sınırlı hem de sınırsız kuyruk sistemleri için, tüm durum olasılıkları şöyle tanımlanabilir.

$$v_j = (P_{0,j}, P_{1,j}, \dots, P_{K,j}); \quad j = 0, 1, 2, \dots, K \quad (2)$$

Bu çalışmada, sınırlı kuyruk sistemleri kullanıldığı için sistem kuyruk kapasiteleri  $K_1$  ve  $K_2$  olarak verilmiştir. Bu

durumda, kuyruk dolu olduğunda gelen görevler engellenir. Bu yüzden sınırlı kuyruk sistemleri için kararlı durum denge denklemleri aşağıdaki gibi ifade edilebilir:

$$v_0[D_0^A + D_0^B] = v_0A_0 + v_1C_1 \quad (3)$$

$$v_j[D_j^A + D_j^B + D_j^C] = v_{j-1}B_{j-1} + v_jA_j + v_{j+1}C_{j+1}; \quad 1 \leq j \leq M-1 \quad (4)$$

$$v_j[D^A + D^B + D^C] = v_{j-1}B + v_jA + v_{j+1}C \quad (5)$$

$$M \leq j \leq L$$

$$v_L[D^A + D^C] = v_{L-1}B + v_LA \quad (6)$$

Denklem 6yı kullanarak da aşağıdaki denklem tanımlanabilir.

$$v_jQ_0 + v_{j+1}Q_1 + v_{j+2}Q_2 = 0 \quad (M-1) \leq j \leq (L-2) \quad (7)$$

Buna ek olarak da, karakteristik matris polinomu  $Q(x)$ ;  $Q(\lambda) = Q_0 + Q_1\lambda + Q_2\lambda^2$  ve  $(\beta) = Q_2 + Q_1\beta + Q_0$  ile tanımlanır. Ayrıca,  $\psi Q(\lambda) = 0$ ;  $|Q(\lambda)| = 0$   $\phi \bar{Q}(\beta) = 0$ ;  $|\bar{Q}(\beta)| = 0$ . Bununla birlikte  $\beta$  ve  $\phi$  sırasıyla,  $\bar{Q}(\beta)$  değerinin öz değer ve öz vektörlerdir. Ayrıca,  $\phi$  vektördür ve  $\phi = \phi_0, \phi_1, \dots, \phi_K$  ve  $\beta = \beta_0, \beta_1, \dots, \beta_K$ . Yukarıda verilen matrisleri ve vektörleri değerlendirdiğimizde sistemin durum olasılıkları formu aşağıdaki gibi elde edilir.

$$P_{i,j} = \sum_{k=0}^K (a_k \psi_k(i) \lambda_k^{j-M+1} + b_k \phi_k(i) \beta_k^{L-j}), \quad (8)$$

$$M-1 \leq j \leq K$$

Dolayısı ile önerilen yük dengeleme modellerinin performans değerlendirmesi için normalizasyon denklemi aşağıdaki gibi verilmiştir.

$$\sum_{j=0}^K v_j e = \sum_{j=0}^{K_1} \sum_{i=0}^{K_2} P_{i,j} = 1.0 \quad (9)$$

Önerilen sisteme göre tüm matrisler elde edildikten, vektörleri değerlendirdikten ve normalizasyon yapıldıktan sonra bütün durum olasılıkları,  $P_{i,j}$  elde edilir.  $P_{i,j}$  elde edildikten sonra performans değerlendirmesi kolayca analiz edilebilir.  $P_{i,j}$ 'den çeşitli başarımların değerleri oldukça kolay hesaplanabilir. Bu çalışmada performans değerlendirmesi için ortalama yük sayısı, ortalama tepki süresi ve engelleme olasılığı sonuçları sırasıyla denklem (10), (11) ve (12) kullanılarak hesaplanmıştır.

$$\sum_{i=0}^{K_1} i \sum_{j=0}^{K_2} P_{i,j} \quad (10)$$

$$\sum_{i=0}^{K_1} P_{i,K_1} \quad (11)$$

$$\frac{\sum_{i=0}^{K_1} i \sum_{j=0}^{K_2} P_{i,j}}{\sum_{i=0}^{K_1} \mu_i \sum_{j=0}^{K_2} P_{i,j}} \quad (12)$$

Benzer modeller matematiksel performans değerlendirmesi için literatürde analiz edilmiştir [1,6,13-17,26-32]. Fakat bulut bilişimde yük dengeleme mekanizmasının analitik modellemesi ve performans değerlendirmesi için tam çözüm tekniği ile ele alınmamıştır. Buna ek olarak bu çalışmada önerilen yük dengeleme mekanizması diğer sistemlerden farklı olarak iki katman içeren açık kuyruk modeli kullanılarak analiz edilmiştir.

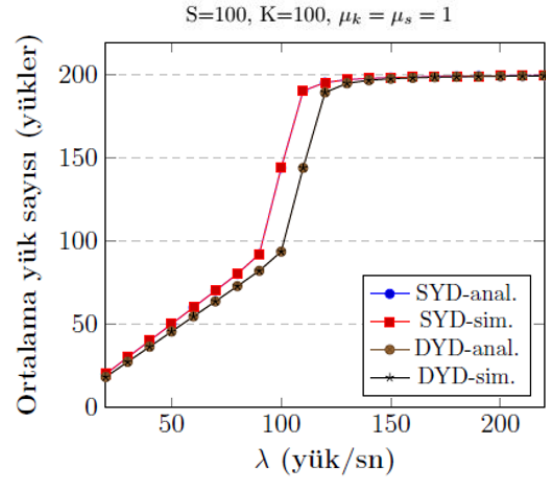
#### 4. BAŞARIM DEĞERLENDİRMESİ (PERFORMANCE EVALUATION)

Bu bölümde, çeşitli sistem parametreleri altında incelenen yük dengeleme mekanizmalarının performanslarını değerlendirmek ve karşılaştırmak için geliştirilen analitik model sonuçları gösterilmiştir. Aynı zamanda, analitik modeller simülasyonla da doğrulanmıştır. Simülasyon sonuçlarını elde etmek için C++'da sistemlerin davranışını simüle eden simülasyon oluşturulmuştur. Simülasyon, esas olarak doğrulama amacıyla kullanılmıştır. Fakat bu tür sistemlerin performans değerlendirmesi için de kullanılabilir. Çünkü bu çalışmada dikkate alınan simülasyon modeli sunulan Markov modellerinden ziyade gerçek senaryoyu simüle etmektedir. Geliştirilen simülasyon programı, M/M/1 ve M/M/c gibi iyi bilinen kuyruk teorisi modellerinin yanı sıra literatürden elde edilen sonuçlar [24,28] kullanılarak da doğrulanmıştır. Şekillerden de görüleceği gibi simülasyon ve analitik model sonuçları birbirlerini eşlemektedir.

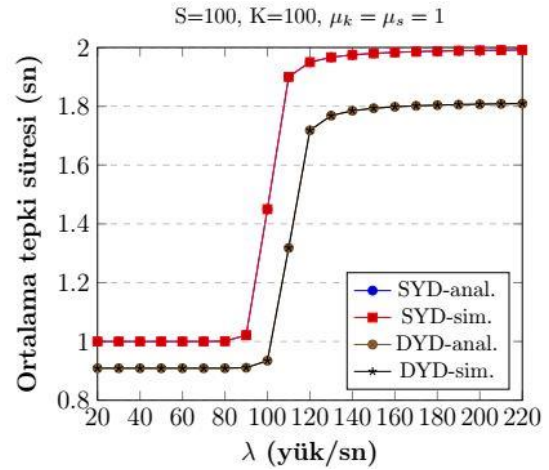
Performans değerleri, değişen yüklerin ortalama varış oranına ( $\lambda$ ) göre tüm analizlerde hesaplanmaktadır. Buna ek olarak verilen tüm sonuçlar sunucular kısmına bakarak analiz edilmiştir. SYD ve DYD mekanizmalarının ortalama yük sonuçları, ortalama tepki süresi ve engelleme olasılığı sonuçları simülasyon sonuçları ile sırasıyla Şekil 9, 10 ve 11'de verilmiştir. Tüm analizler için parametreler literatürden de yararlanılarak şu şekilde alınmıştır:  $S=100$ ,  $K=100$ ,  $\mu_k=\mu_s=1$ . Tüm şekillerden de görüleceği üzere DYD mekanizması SYD mekanizmasına göre daha iyi performans sonuçları elde etmektedir. Tüm sonuçlardan da görüleceği üzere, DYD mekanizması performans ölçütleri üzerindeki etkisini açıkça göstermektedir.

Şekil 9'da, sistemde bulunan ortalama yük sayıları sonuçları sisteme gelen yükler cinsinden her iki mekanizma için gösterilmiştir. Sistemde 100 tane sunucu (kontrol sunucusu dâhil) ve kuyruk kapasiteleri 100 olarak alınmıştır. Bu nedenle Şekil 9'da görüldüğü üzere  $K_1$  ve  $K_2$  en fazla 200 olabilmektedir. Diğer bir deyişle sistem dolu olduğunda ortalama yük sayısı 200 olmaktadır. Şekilde de net bir şekilde DYD mekanizmasının daha iyi servis verdiği açıkça görülmektedir. Özellikle sistemin yoğun olduğu durumlarda ( $\lambda$  değerinin 80 ve 120 arasında olduğunda) DYD mekanizması meşgul olmayan sunucuları dinamik olarak bularak sistemden servis almalarını sağlamaktadır. Yüklerin varış oranını

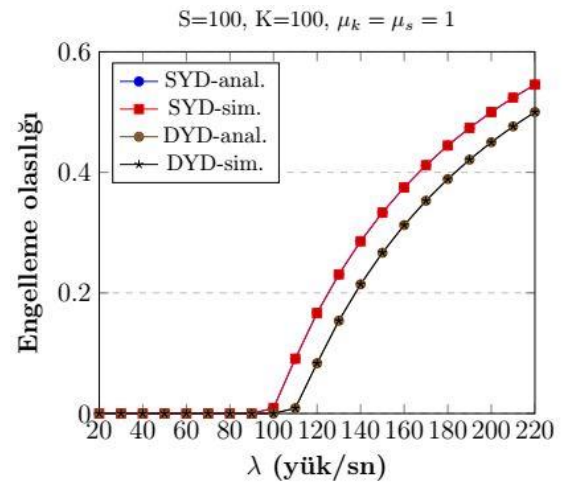
artırdığımızda, sistemdeki yüklerin gerekli hizmeti aldığından istek sayısında azalma görülmektedir. Diğer bir deyişle, boşta bulunan sunucu ve/veya kuyruklara yükler yerleşip servis edilebilmektedirler.



Şekil 9. Ortalama yük sayısı (Mean queue length)



Şekil 10. Ortalama tepki süresi (Mean response time)



Şekil 11. Engelleme olasılığı (Blocking probability)

Diğer yandan sistemin çok yoğun olduğu durumlarda ( $\lambda$  değeri 140 ve üzeri olduğunda) ise her iki mekanizma da en fazla 200 yük kabul edebildiğinden eşit sonuçlar vermektedir. Sistemdeki trafiğin az olduğu durumlarda ise DYD çok az olmasına rağmen yine de SYD mekanizmasından daha iyi sonuçlar vermektedir. İki mekanizmanın ortalama tepki süresi ve engelleme olasılıkları sonuçlarının karşılaştırması Şekil 10 ve 11'de gösterilmiştir. Şekil 10 ve 11 sonuçları da DYD mekanizmasının yük sayısını artırdığımızda sistemin verimliliğini artırdığını açıkça göstermektedir. Sistemde DYD mekanizması kullanıldığında daha az gecikme olacağı ve engellenen yüklerin de azalacağı görülmektedir. Şekil 10'da görüldüğü üzere DYD mekanizması sisteme gelen yükleri daha hızlı bir şekilde servis etmektedir. Bu da SYD mekanizmasında boşta olan sunucuları boş bırakmayarak onlara yükleri dinamik bir şekilde göndererek ve servis almalarını sağlayarak gerçekleşmektedir. Bu da sistemin daha hızlı tepki vermesini sağlamaktadır. Örneğin  $\lambda=100$  olduğunda SYD için tepki süresi 1.45sn iken DYD için 0.9snlere düşmektedir. Şekil 10 ve 11'de ise belirgin bir şekilde sistemin yoğun olduğu durumlarda DYD mekanizması çok daha iyi performans göstermektedir. Sonuçlardan da anlaşılacağı üzere önerilen DYD mekanizması önerilen sistemin performansını önemli ölçüde artırmıştır.

## 5. SONUÇ (CONCLUSION)

Bu çalışmada, bulut bilişim sistemleri için iki yük dengeleme mekanizması matematiksel modellerle incelenip daha sonra da performans analizi için karşılaştırılmıştır. Statik ve dinamik yük dengeleme mekanizmalarının performans değerlendirilmesi kesin sonuç veren Spektral genişleme metodu kullanarak hesaplanmıştır.

Bu çalışmada, statik yük dengeleme mekanizması literatürde kullanılan gibi alınmasına karşın dinamik yük dengeleme mekanizması literatürün dışında yeni bir önerme yapılarak ele alınmıştır. Dinamik yük dengeleme mekanizmasında kontrol sunucusu yeni gelen istekleri servis etmek için en uygun sunucuyu buluncaya kadar dinamik bir şekilde değişmektedir. Bu nedenle çalışmada önerilen yük dengeleme mekanizması diğer sistemlerden farklı olarak iki katman içeren açık kuyruk modeli kullanılarak analiz edilmiştir. Bu mekanizmaların performans değerleri olan, ortalama yük sayısı, ortalama tepki süresi ve engelleme olasılığı sonuçlarını elde etmek için analitik modeller geliştirilmiştir. Statik ve dinamik yük dengeleme mekanizmaları için analitik modeller duruma bağlı Markov zincirini temel alınmıştır. Markov zincirinin durum olasılıklarını çözdükten sonra performans değerleri elde edilmiştir. Modelleri doğrulamak ve kabul edilebilir olduğunu göstermek için simülasyon da kullanılmıştır.

İki yük dengeleme mekanizması arasında, dinamik yük dengeleme mekanizmasının daha iyi performans sonuçları verdiği açıkça görülmektedir. Sonuçlardan da görüleceği üzere dinamik yük dengeleme sunucu kümesinin kaynaklarını daha verimli bir şekilde kullanabilir, böylece

bekleme süresini ve kullanıcıların yaşadığı engelleme olasılığı azalır. Bunlara ek olarak önerilen modelin çözümü için kullanılan kesin Spektral genişleme modeli, her tür ağ performansı, sistem analizi ve yapılandırması için kullanılabilir.

## KAYNAKLAR (REFERENCES)

- [1] E. Çağlar ve Y. Kirsal, "Analytical Modelling and Performance Evaluation of Security Issues For Cloud Computing", **27th Signal Processing and Communications Applications Conference (SIU)**, IEEE, 2019.
- [2] C. Paşaoğlu, ve E. Cevheroğlu, "Bulut Bilişim Sistemleri Kapsamında Kişisel Verilerin Şifreleme Yöntemleri ile Korunması", *Bilişim Teknolojileri Dergisi*, 13(2), 183-195, 2020.
- [3] H. Özcan, ve B. G. Emiroğlu, "Bulut Tabanlı Öğrenme Yönetim Sistemi Seçiminde Bulanık Çok Kriterli Karar Analizi Yaklaşımı", *Bilişim Teknolojileri Dergisi*, 13(1), 97-111, 2020.
- [4] Ç. Çetin, N. Yaman, L. Sabah, E. Ayday, "Bulut Bilişim (Cloud Computing) Teknolojisinin Uzaktan Algılama ve Coğrafi Bilgi Sistemlerinde Uygulama Olanakları", **Türkiye Ulusal Fotogrametri ve Uzaktan Algılama Birliği VII. Teknik Sempozyumu**, 23-25, 2013.
- [5] D. Goutam, A. Verma, N. Agrawal, "The Performance Evaluation Of Proactive Fault Tolerant Scheme Over Cloud Using Cloudsim Simulator", **Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)**, 171-176, IEEE, 2014.
- [6] M. R. Mesbahi, M. Hashemi, A. M. Rahmani, "Performance evaluation and analysis of load balancing algorithms in cloud computing environments", **2016 Second International Conference on Web Research (ICWR)**, 145-151, IEEE, 2016.
- [7] E. N. Desyatirikova, O. V. Kuripta, Y. S. Stroganova, ve I. P. Abrosimov, "Quality Management in IT Service Management Based On Statistical Aggregation And Decomposition Approach", **In 2017 International Conference Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS)**, 500-505, IEEE, 2017.
- [8] S. K. Mishra, B. Sahoo, ve P. P. Parida, "Load balancing in cloud computing: a big picture", *Journal of King Saud University-Computer and Information Sciences*, 2018.
- [9] V. N. Volkova, L. V. Chemenkaya, E. N. Desyatirikova, M. Hajali, A. Khodar, ve A. Osama, "Load balancing in cloud computing", **2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)**, 387-390, IEEE, 2018.
- [10] K. Hashizume, D. G. Rosado, E. FernandezMedina, ve E. B. Fernandez, "An Analysis of Security Issues For Cloud Computing", *Journal of Internet Services and Applications*, 4-5, 2013.
- [11] A. S. Hanamakkanavar, ve V. S. Handur, "Load Balancing In Distributed Systems: A survey", *Proceedings of International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)*, 14(2), 2015.

- [12] H. Siar, K. Kiani, A. T. Chronopoulos, "A Combination Of Game Theory And Genetic Algorithm For Load Balancing In Distributed Computer Systems", *International Journal of Advanced Intelligence Paradigms*, 9(1), 82-95, 2017.
- [13] H. Khazaee, J. Mistic, V. B. Mistic, "Performance of an iaas cloud with live migration of virtual machines", **In 2013 IEEE Global Communications Conference (GLOBECOM)**, 2289–2293, 2013.
- [14] L. Guo, T. Yan, S. Zhao, C. Jiang, "Dynamic performance optimization for cloud computing using m/m/m queueing system", *Journal of Applied Mathematics*, 2014.
- [15] R. Ghosh, F. Longo, V. K. Naik, K. S. Trivedi, "Modeling and performance analysis of large scale iaas clouds," *Future Generation Computer Systems*, 29(5), 1216–1234, 2013.
- [16] K. Salah, K. Elbadawi, R. Boutaba, "An analytical model for estimating cloud resources of elastic services," *Journal of Network and Systems Management*, 24(2), 285–308, 2016.
- [17] J. Vilaplana, F. Solsona, I. Teixido, J. Mateo, F. Abella, and J. Rius, "A queuing theory model for cloud computing", *The Journal of Supercomputing*, 69(1), 492–507, 2014.
- [18] N. Joshi, K. Kotecha, D. B. Choksi, S. Pandya, "Implementation of Novel Load Balancing Technique in Cloud Computing Environment", **2018 International Conference on Computer Communication and Informatics (ICCCI)**, 1-5, IEEE, 2018.
- [19] K. Pathak, G. Vahinde, "Comparison of Particle Swarm Optimization And Genetic Algorithm For Load Balancing In Cloud Computing Environment", *International Journal of Research in Computer & Information Technology (IJRCIT)*, 1(1), 2015.
- [20] A. Yadav, "Comparative Analysis Of Load Balancing Algorithms In Cloud Computing", *International Journal of Enhanced Research in Management & Computer Applications*, 4(9), 2015.
- [21] A. Nair, S. Anand, S. A. Sinha, "Performance Booster For Load Balancing In Cloud Computing With My Load Balancer Technique", *International Journal of Recent Technology and Engineering*, 8(1), 2019.
- [22] G. Megharaj, K. G. Mohan "A Survey On Load Balancing Techniques In Cloud Computing", *IOSR Journal of Computer Engineering (IOSR-JCE)*, 18(2), 55-61, 2016.
- [23] R. Gao, J. Wu, "Dynamic Load Balancing Strategy for Cloud Computing with Ant Colony Optimization", *Future Internet*, 7, 465-483, 2015.
- [24] Y. Zhu, D. Zhao, W. Wang, H. He, "A Novel Load Balancing Algorithm Based on Improved Particle Swarm Optimization in Cloud Computing Environment", **International Conference on Human Centered Computing**, Springer, 634–645, 2016.
- [25] V. S. Handur, S. Belkar, S. Deshpande, P. R. Marakumbi, "Study of load balancing algorithms for Cloud Computing", **2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT)**, Bangalore, India, 173-176, 2018.
- [26] Y. Kırşal, "Performability evaluation and optimization analysis of repairmen for large-scale networks", **25th Signal Processing and Communications Applications Conference (SIU)**, 1-4, 2017.
- [27] Y. Kırşal, "Analytical modelling and optimization analysis of large-scale communication systems and networks with repairmen policy", *Computing*, 100(5), 503-527, 2018.
- [28] S. Grishechkın, **Queueing Theory, Probability Theory and Mathematical Statistics**, 1, 455, 2020.
- [29] J. Cao, Z. Ma, S. Guo, X. Yu, "Performance analysis of non-exhaustive wireless sensor networks based on queueing theory", *International Journal of Communication Networks and Distributed Systems*, 24(2), 186-213, 2020.
- [30] S. K. Majhi, S. S. Pal, S. Bhuyan, S. K. Dhal, "Queuing Analysis of Cloud Load Balancing Algorithms", *International Journal of Knowledge-Based Organizations (IKBO)*, 8(1), 50-67, 2018.
- [31] A. Pourghaffari, M. Barari, S. Sedighian Kashi, "An efficient method for allocating resources in a cloud computing environment with a load balancing approach", *Concurrency and Computation: Practice and Experience*, 31(17), e5285, 2019.
- [32] F. Ebadifard, S. M. Babamir, "Autonomic task scheduling algorithm for dynamic workloads through a load balancing technique for the cloud-computing environment", *Cluster Computing*, 1-27, 2020.