MÜHENDİSLİK FAKÜLTESİ

Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi

International Journal of Engineering Research and Development

Cilt/Volume:10 Sayı/Issue:3 Aralık/December 2018 **https://doi.org/10.29137/umagd.484995**

Seçilmiş Konferans Makalesi / Selected Conference Article

Dağıtık Dosya Sistemi Kullanarak Sağlık Alanına Özel Dosya Sistemi Geliştirilmesi

Developing a Health-Specific File System Using a Distributed File System

Mahmut ÜNVER [1], Erdal ERDAL [2], Atilla ERGÜZEN [*2]

[1.] Kırıkkale Üniversitesi, Bilgisayar Teknolojileri Bölümü, 71450, Kırıkkale, Türkiye

[2.] Kırıkkale Üniversitesi, Bilgisayar Mühendisliği Bölümü, 71450, Kırıkkale, Türkiye

Başvuru/Received: 19/11/2018 **Kabul/Accepted**: 17/12/2018 **Son Versiyon/Final Version**: 31/12/2018

Öz

İnternet kullanımının artmasıyla birlikte özellikle bilgisayar, telefon ve tablet gibi dijital cihazların kullanımı artmıştır. Ayrıca, Nesnelerin İnterneti kavramı gelişen teknolojiyle hayatımıza girmeye başladı. Sağlık sektörü, bu teknolojik gelişmelerden en çok etkilenen sektörlerden biridir. Sağlık sektöründe kullanılan dijital ekipmanlar, dünyada üretilen verilerinin %10'una sahiptir. Nesnelerin İnterneti kavramının sağlık alanına girmesi de bu alandaki veri büyüklüğünü hızla artttıran önemli faktör olmuştur. Bu çalışmanın amacı, sağlık sektöründe kullanılabilecek özel bir dağıtılmış dosya sistemi geliştirmektir. Önerilen yöntem, sağlık alanında veri depolamak için ölçeklenebilir, sağlam ve sunucusuz dağıtılmış bir mimari önermektedir. Sistemde, diğer dağıtık sistemlerden farklı olarak name node veya sunucu gibi çalışan bir makine bulunmaktadır. Güvenlik önceliklidir ve statik IP ve şifreleme yöntemleri kullanılmıştır. Önerilen sistem, yüksek performans sağlar. Ayrıca, literatürde kullanılan diğer name node mimarisini kullanan sistemlere göre sistemin dayanıklılığı daha iyidir. Deney sonuçlarına göre geliştirilen sistem NoSQL sistemlerine göre% 93, ilişkisel veritabanı yönetim sistemlerine göre% 78 ve işletim sistemlerine göre % 71 daha iyi sonuç vermektedir.

Anahtar Kelimeler

Dağıtık Dosya Sistemi, Medikal Büyük Veri, Dosya Sistemi.

Abstract

With the increase of Internet usage, the use of digital devices, especially computers, telephones and tablets has increased. Besides, the concept of Internet of Things started to enter our lives with the developing technology. The health sector is one of the sectors most affected by these technological developments. It has 10% of all world data of digital equipment used in health sector. With the introduction of the Internet of Things concept into the field of health, the data size in this area has started to increase very rapidly. The aim of this study is to develop a health-specific distributed file system. The proposed method proposes a scalable, robust and serverless distributed architecture for storing data in the healthcare field. In the system, unlike other distributed systems, there isn't a machine running as name node or server. Security is prioritized, and static IP and encryption methods have been used. The proposed system provides high performance. In addition, the system has better durability compared to other name node architecture systems that is in the literature. According to experimental results, the developed system has a better result than 93% according to NoSQL systems, 78% according to relational database management systems and 71% compared to operating systems.

Key Words

Distributed File System, Medical Big Data, File System

munver@kku.edu.tr

1. GIRIŞ

Sonyıllarda çeşitli faktörler dünyada üretilen elektronik verinin büyüklüğünü son derece arttırmıştır. Bu faktörlerin en önemlisi kuşkusuz internet kullanımının yaygınlaşmasıdır. İnternet kullanan kişi sayısı, 2000 yılında dünya nüfusunun % 6'sı iken bu oran 2005 yılında %15,8'e, 2010 yılında %29,2'ye ve 2016 yılında %46,2'e yükselmiştir (Anonim, Internet Statics, 2016). Aynı şekilde, 2015 yılındaki anlık yaklaşık 3 milyar günlük ortalama internet kullanıcı sayısı, 1960 yılının dünya nüfusuna erişmiştir (Anonim, Virtualization&Cloud, 2017). 2016 yılında da bu sayı 3,4 milyara ulaşmıştır. Hergün dünya üzerinde 2,5 Exabyte veri üretilmektedir. Ayrıca şu anda dünyadaki verinin 90%'ı son iki yılda üretilmiştir. 2011 yılında dünya da üretilen toplam veri 295 Exabyte iken bu sayının 2023'te 163 Zettabyte'a ulaşacağı tahmin edilmektedir (Anonim, Mearian, 2018). Bu miktardaki büyük veri, farklı sektörler yoluyla ortaya çıkmaktadır. Bunların en önemlileri; sağlık, ticaret, eğitim, sanayi gibi faktörlerdir. Aynı şekilde, sosyal medya kullanımı da veri hacmini arttıran önemli bir faktördür.

Veri mktarı, sağlık sektöründe dikkat çekici bir şekilde artmıştır. Teşhis için kullanılan elektronik tıbbi aygıtlar bu artışta önemli etkendir. Amerika Birleşik Devletleri'nde, 2011 yılında sağlık sektöründe üretilen veri miktarı 150 Exabyte iken bu sayının 2020 yılında 2,3 Zettabyte'a ulaşacağı tahmin edilmektedir (Dean 2009), (Erguzen ve Erdal, 2017).

Büyük veri hacminin artmasıyla, verinin depolanması ve yönetilmesi sorunu da ortaya çıkmaktadır. Bu problem klasik veri tabanı yönetim sistemleri ile çözülmeye çalışılmış ancak veri hacmi arttıkça veri tabanı yönetim sistemlerinin bu problemi çözemeyeci ortaya çıkmıştır. Bunun devamında da ilişkisel veri tabanlarının yerine dağıtık dosya sistemleri büyük veriyi yönetmek için kullanılmıştır.

Yapılan çalışma, hybrid bir mimarı ortaya çıkartmıştır. Sistem yapısında, işletim sistemi dosya yapısı ve dağıtık dosya mimarisi birlikte kullanılmıştır. Sunucusuz bir mimarı tasarlanmıştır. Dağıtık Dosya sistemlerinde var olan name node kullanılmamıştır. Name node'un görevi replica node tarafından yapılmaktadır. Name node'un çalışmaması nedeniyle ortaya çıkacak olan sistemin tamamının çalışmaması problemi ortadan kaldırılmıştır. Tüm sistemde sadece bir node kalsa bile sistem çalışmasına devam edebilmektedir. Bu da sistemimizin sahip olduğu özelliklerden biridir.

2. YAPILAN ÇALIŞMALAR

Büyük veriyi yönetme ve depolama problemini çözmek için yapılan ilk dağıtık dosya sistemi çalışmaları 1975 yılında başlamıştır (Thomson ve Abadi, 2015). Bu çalışmalardan biri olan ROE mimarisinin amacı, replica tutarlılığı, kolay kurulum, dosyalara güvenli izin yetkilerinin verilmesi ve ağ şeffaflığıdır (Ellis ve Floyd., 1983). Network File System (NFS), 1984 yılında Sun Microsystems tarafından geliştirilmeye başlanmıştır. UNIX sistemlerde en çok kullanılan DFS'dir. Communication için Remote Procedure Call (RPC) modelini kullanır (Sandberg, Goldberg, Kleiman, Walsh ve Lyon, 1985). Temel tasarım yapısı, klasik Unix dosya sisteminin dağıtık çalıştırılmasıdır. Sanal dosya sistemi, bir ara katman gibi kullanılmıştır. Bu sayede, istemciler farklı dosya sistemleriyle kolayca çalışabilmektedirler. NFS'te hata toleransı yüksektir. Dosyaların durumları ile ilgili bilgi tutulur, istemciden kaynaklanan bir hata durumunda sunucu haberdar edilir. NFS'te dosya replicationu yapılmaz. Tüm sistem replication yapılır. Dosyalar önbelleklendeikten sonraönbellekteki kopya ile serverdaki kopya karşılaştırılır. Zamanları farklı ise dosya değişmiş demektir ve önbellekteki atılır (Coulouris, Dollimore, Kindberg ve Blair., 2011). NFS ile sadece dosya sistemi paylaşılabilir, yazıcı ve modem paylaşılamaz. Paylaştırılacak olan nesneler, bir dizinin(directory) parçası olabileceği gibi bir dosya da olabilir. NFS'de her uygulamanın yerel diske kurulması gerekli değildir. Server üzerinden uygulamalara paylaşım sağlanabilir. Aynı makine hem server hem client olabilir. Bütün bunların sonucunda NFS, veri depolama maliyeti düşük bir system olarak ortaya çıkmıştır.

2006 yılında duyurulan, CEPH dosya sistemi, nesne depolama işlemi yapan benzer sistemlerin bir üst katmanında yer alır. Bu katman, data ve metadata yönetimini ayırır. Bunu güvenilmez nesne depolama aygıtları (OSDs) için tasarlanmış olan rasgele veri dağıtma fonksiyonu (CRUSH) ile gerçekleştirir. Bu fonksiyonu allocation tablosu yerine koyar. CEPH ile dağıtık veri replikasyonu, hata algılama ve kurtarma işlemleri, yerel dosya sisteminde çalışan nesne depolama aygıtlarına dağıtılır. Bu sayede, sistem performansı güçlendirilmiş olur (Weil, Brandt, Miller, Long ve Maltzahn, 2006).

2007'de Hadoop Dağıtık Dosya Sistemi ile MapReduce paralel hesaplama yönteminden meydana gelen Hadoop tanıtılmıştır. Çok büyük veri (big data) kümelerinin analizi ve dönüştürülmesini sağlayan bir frameworktur. HDFS, büyük veriyi standart sunucular uzerinde bloklara ayırarak dağıtır. Veri güvenliğini sağlamak için, blokları kopyalayarak sunucularda yedekler (Shvachko, Kuang ve Radia, 2010).

2016 yılında, Mohammed S. Al-kahtani ve Lutful Karim ölçeklenebilir bir dağıtık sistem frameworku (Al-Kahtani ve Karim, 2017) sunmuştur. Ölçeklendirmeyi merkezi sunucu üzerinde yapmaktadır. Önerilen framework, veri toplama sonucu verilerin miktarı arttıkça, veri işleme işini sunucu tarafından diğer bilgisayarlar tarafına aktarmaktadır. Diğer bir tabirle, sistem veri miktarı arttığı zaman dağıtık olarak çalışmaktadır.

Ergüzen ve Ünver'in yaptığı çalışmada (Ergüzen ve Ünver, 2018), dağıtık dosya sistemi ile işletim sistemi dosya yapısını kullanan bir mimari tasarlanmıştır. Sistemde, data node ve replica nodelar kullanılmıştır. Tasarlanan sistemde, 4 adet data node istemci sunucularının isteklerini TCP/IP bağlantısı ile cevaplamaktadır.

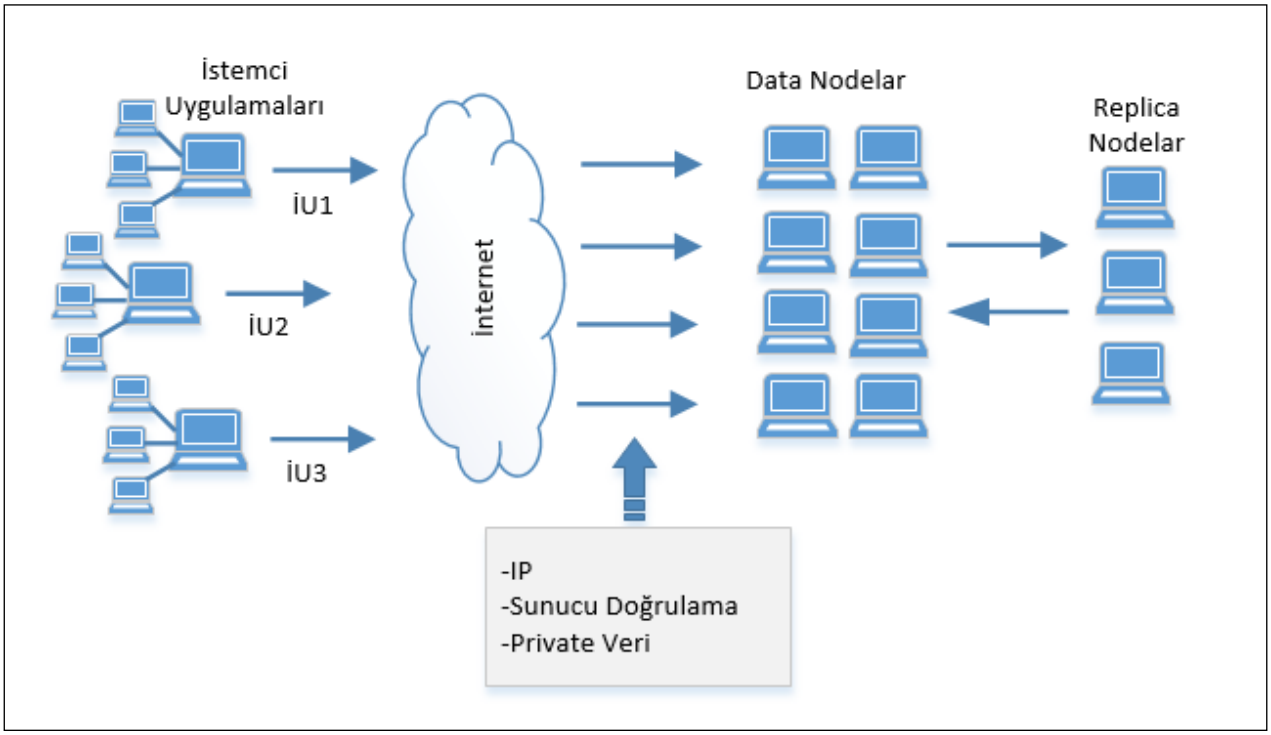
3. SISTEM MIMARISI

Günümüzde dağıtık dosyalama sistemleri 2 ana kategoride ele alınabilir:

a) Dosya sistemi ve cluster yönetimi,
b) İhtiyaç duyulan verinin node'lar tarafından en kısa sürede ve tutarlı bir şekilde analizinin yapılması (veri madenciliği). Bizim çalışmamızda veri saklama ve nodeların bir bütün olarak çalışmasını sağlayacak bir platform geliştirilmiştir.

Bu çalışmada büyük veri için hızlı, güvenli, sunucusuz, kolay yönetilebilir, ölçeklendirilebilir, sürdürülebilir, güçlü bir dağıtık sistem ortaya çıkmıştır. Sistem orta ölçekli bir sağlık kuruluşunda ortaya çıkan büyük veriyi depolayabilir ve yönetebilir. Bu çalışmanın farklılıkları aşağıda detaylı bir şekilde anlatılmıştır.

Sistem bileşenleri şunlardır: İstemci uygulamaları, data nodelar, sanal dosya sistemi ve replica nodelardır.



Şekil 1. Sistem Genel Yapısı

3.1. İstemci Uygulamaları

İstemci uygulamaları, sistemden dosyalama hizmeti alan uzaktaki uygulamalardır. Uygulamalar tüm isteklerini TCP/IP üzerinden yapmaktadırlar. Bu yüzden system, platformdan bağımsızdır. Ancak bunların statik bir IP'ye sahip olması gerekmektedir. İstemci uygulamalar, dosyalama ve depolama amaçlı olarak 'write once read many' şeklinde bu sistemi kolaylıkla kullanabilmektedir. Hizmet almak isteyen uygulama, bağlantı için Dynamic Link Library'yi kendi sistemine kurmalıdır. Bu sistemde GetAvailableDatanodeIp, ChangePassword, SaveFile, ReadFile, DeleteFile, UpdateFile, GetFileFromReplica methodları mevcuttur. İstemci uygulaması, kendilerine verilen symetric cryptography ile gönderilen kullanıcı adı, şifre ve sahip olduğu statik IP ile kendilerine ayrılan data node'a bağlanır. İstemci uygulamaların sayısında herhangi bir sınır yoktur.

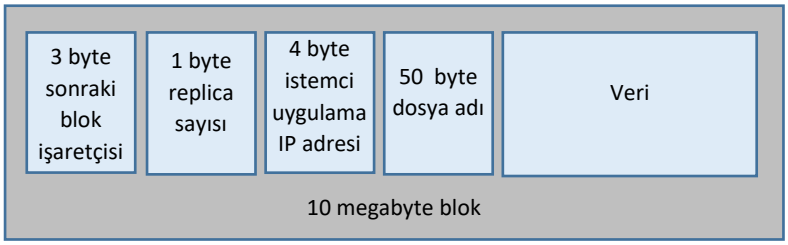
3.2. Düğümler

İki tür node kullanılmaktadır. Data node'lar, bilindiği gibi, istemciden gelen dosyaları saklamak ve yönetmek için kullanılmaktadır. Hizmet alan istemci uygulamaları, data node'lar ile TCP/IP ağ servisi aracılığı ile iletişime geçer. Bu çalışmada özellikle data node'ların her biri farklı uygulamalara hizmet veren aynı seviyedeki sunuculardır. Yazılım, system geliştiriciler tarafından geliştirilen dynamic link library ile, data node'lara güvenli bir bağlantı sağlamaktadır. Bu bağlantı Symmetric-key algoritması ile yapılmaktadır. Bununla birlikte uygulama sunucuları, data node'lardan hizmet alan, sabit bir IP adresine sahiptir. Sistemde istemci uygulamaları ile iletişime geçebilecek 8 tane data node kullanılmıştır. Daha büyük kuruluşlarda istemci uygulamaları sayısı daha fazla olacaktır. Bu sayede iletişime geçilebilecek uygulama sayısı arttırılmış olmaktadır.

Replica node'lar, data node'ların sakladığı dosyaların kopyasını saklamak için kullanılır. Data node'lar kendisine yüklenen dosyaları asenkron olarak replica node'lara gönderir. Replica node'lar bu dosyaları temel işletim sistemi dosyası içerisinde saklar. Burada sadece işletim sisteminin dosyalama hizmeti kullanılır, özel bir dosya yapısı veya herhangi bir dosya işleme stratejisi kullanılmaz. Bu, replica node'ların bir dosya sunucusu gibi çalıştığı anlamına gelir. Replica node'lar, yalnızca işletim sistemi dizin yapısındaki dosyaları depolamaktan sorumlu olan dosya sunucusu gibi çalışmaktadır.

3.3. Sanal Dosya Sistemi

İletilmesi gereken veriyi kendi dosya sisteminde saklayacak şekilde bloklara ayırır. Kendine ait disk dosya listeleme yapısına sahiptir. Sanal Dosya sistemi, diskte işletim sisteminin kullanmadığı kalan alanının tamamını kullanabilmektedir. Bu da ortalama bir diskte %75-85 oranında olmaktadır. Tutulan herbir blok, veriyi oluşturan diğer bloku işaret eder. Verinin kaç bloktan oluştuğu, verinin bloklarını sırası gibi veriler yine sanal dosya sisteminde tutulur. Sanal dosya sistemi 10 megabytelık bloklara ayrılmıştır. Bir blok, şekil 2 de gösterildiği gibi, 3 bytelık bir sonraki blok işaretçisinden, bir bytelık tutulduğu replica sunucusu sayısından, 4 bytelık istemci uygulamasının IP adresinden, 50 byte dosyaadından ve veri bloğundan oluşmaktadır.



Şekil 2. 10 megabytelık blok yapısı.

3.4. Sistemin Özellikleri

1. Sistem sunucusuz çalışmaktadır. Data nodelar sunucu gibi çalışmaktadır.
2. Küçük ve orta ölçekli kuruluşlarda optimum faydayı sağlayacak şekilde blok büyüklüğüne sahiptir.
3. Sanal dosya sistemi, temel işletim sisteminden kolayca üretilmiştir.
4. İstemci uygulamaları başlangıç adresini tutar.
5. 8 adet data node kullanılmıştır.
6. Sistemdeki herhangi bir makine dosya sunucusu oalrak çalışabilmektedir.
7. Name node veya server node kullanılmadığından, bunların çalışmamaması durumunda sistem çökmesi problemi, name node kullanılmadığı için ortadan kalkmıştır.
8. 128-bit simetrik kriptoloji kullanılarak bağlantı güvenli hale getirilmiştir.
9. Hybrid mimariden dolayı okuma yazma performansı iyidir.

3.5. Sistem Performansı

Kullanılan Hadoop konfigürasyonu, bir name node ve üç data node'a sahiptir. Her bir node'ta Redhat Enterprise Linux Server 6.0 çalışmaktadır. Ayrıca, herbir node'ta Java-1.6.0, MongoDB-3.4.2 ve Hadoop-2.7.2 kurulmuştur. CouschBaseDB ise, bucketlar, 20 megabyte'lık clusterlardan meydana gelmektedir. Sistemin 6 gigabyte RAM ve 200 gigabyte disk alanı mevcuttur. Veri tabanı sunucusu olarak Microsoft SQL Server 2014 kullanılmıştır.

Tasarlanan sistem diğer dosya sistemleri ile karşılaştırılmıştır. Karşılaştırma sürecinde aynı donanım ve yazılım kullanılmaya çalışılmıştır. Test sonuçlarına göre, tasarlanan sistemin performansının NoSQL sisteminden %93, RDBMS'den (İlişkisel Veritabanı

Yönetim Sistemi) %78 ve İşletim Sisteminden %71 daha iyi olduğu ortaya çıkmıştır. Bu sonuçlar, 4 data node kullanılan sistemler ile performansında farklılık olmadığını göstermiştir. Özellikle Hadoop'un 64 megabyte boyutundaki blok büyüklüğünün yerine tasarlanan sistemdeki 10 megabyte blok büyüklüğü, 64 megabytetan daha küçük dosyalar üreten sistemler için optimum çalışma ortamı sağlamaktadır. Bu sayede küçük boyutlu veri bloklarına sahip kuruluşlar için depolama alanı kazanımı sağlanmış olacaktır. Sanal Dosya sistemini yönetebilmek için işletim sistemi altındaçalışan kolay bir system geliştirilmiştir. Hem bitmap hemde bağlı liste yapısı kullanılarak ortaya çıkarılan hybrid sistem, 10 megabytelık sabit büyüklükteki blokları yönetmek için kullanılmıştır.

4. SONUÇ

Bu çalışmada, özellikle küçük ve orta büyüklükteki big data setleri için hızlı, güvenli, sunucusuz, güçlü, sağlam, yönetilebilir, ölçeklendirilebilir dağıtık dosya sistemi geliştirilmiştir. Sistem, çoğu dağıtık sistemlerde olmayan platformdan bağımsız bir sistem ortaya koymaktadır. TCP/IP protokolünü kullanır. Server node, head node veya name node kullanılmamaktadır. Sistem sunucusuzdur. Bu sayede sistemin sürekliliği sağlanmış olur. Bir data node sistemde düzgün çalışmadığında, diğer node'lar istekleri cevaplayabilir ve sistemin çalışması devam eder. Gerekirse, sadece windows servis yordamı node'a yüklenerek sisteme sınırsız sayıda data node kolaylıkla eklenebilir. Bu sistemin diğer sistemlere göre üstünlüğüdür.

Sistem güvenliği, diğer dağıtık dosya sistemlerinde olduğu gibi kabul edilebilir bir düzeye sahiptir. Güvenlik iki yolla sağlanmaktadır. Birincisi, data node'tan hizmet veren istemci makinelerin IP'ler kontrol edilerek, ikincisi ise, uygulama yazılımının data node'lara gönderdiği verileri şifrelemektir.

Replica işlemi, işletim sisteminin dosya işlemleri sistem çağrılarını kullanır (Windows'da API'ler). Data node'lar, kendilerine eş zamansız olarak yüklenen dosyaları replica node'lara gönderir. Sistemin kolay yönetilebilir olmasını sağlamışve hata toleransını arttırmıştır.

Bu çalışma ile, veri güvenliği sağlanmış, performansı diğer dağıtık sistemlere göre daha iyi, sunucusuz bir sistem ortaya çıkmıştır. Sistem küçükve orta ölçekli kuruluşların büyük veri depolama problemlerini için optmum maliyet ile çözebilecek bir sistem olmuştur. Data node sayısının arttırılması sistem performansını etkilememiştir.

REFERANSLAR

Al-Kahtani M. S., Karim L. (2017). An Efficient Distributed Algorithm for Big Data Processing, Arabian Journal for Science and Engineering, 42 (8): 3149-3157.

Anonim, Internet Statistics (2016). http://www.internetlivestats.com/internet-users/#trend (Erişim Tarihi:16.07.2016)

Anonim, Virtualization&Cloud (2017). http://www.vcloudnews.com/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily/ (Erişim Tarihi:17.02.2017).

Anonim, L. Mearian (2018). Scientists calculate total data stored to date: 295+ exabytes, https://www.computerworld.com/article/2513110/data-center/scientists-calculate-total-data-stored-to-date--295--exabytes.html (Erişim Tarihi: 30.03.2018)

Coulouris G., Dollimore J., Kindberg T., Blair G. (2011). Distributed Systems: Concepts and Design 5th, USA: Addison-Wesley Publishing Company.

Dean B. B. (2009). Use of Electronic Medical Records for Health Outcomes Research, Medical Care Research and Review, 66 (6), 611-638.

Ellis A., Floyd R. A. (1983). The ROE File System, 3rd Symposium on Reliability in Distributed Software and Database Systems, Clearwater Beach, FL, USA.

Erguzen A., Erdal E. (2017). "Medical Image Archiving System Implementation with Lossless Region of Interest and Optical Character Recognition," Journal of Medical Imaging and Health Informatics, 7, 1-7.

Ergüzen A., Ünver M. (2018). Developing a File System Structure to Solve Healthy Big Data Storage and Archiving Problems Using a Distributed File System, Appl. Sci., 8(6), 913.

Sandberg R., A., Goldberg D., Kleiman S., Walsh D., Lyon B. L. (1985). Design and Implemantation of The Sun Network File System, USENIX Conference and Exhibition, Portland, Oregon, USA.

Shvachko K., Kuang H., Radia S. (2010). The Hadoop Distributed File System, Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on, Incline Village, NV, USA.

Thomson A., Abadi D. J (2015). CalvinFS: Consistent WAN Replication and Scalable Metadata Management for Distributed File Systems, 13th USENIX Conference on File and Storage Technologies (FAST ’15), Santa Clara, CA, USA.

Weil S. A., Brandt S. A., Miller E. L., Long D. D. E., Maltzahn C. (2006). Ceph: A Scalable, High-Performance Distributed File System, OSDI '06 7th Symposium on Operating Systems Design and Implementation, Seattle, Washington, USA.