

PAPER DETAILS

TITLE: Farkli Is Yükleri Altinda NoSQL Sistemlerinin Performans Analizi

AUTHORS: Yasar DASDEMIR,Burak Cem KARA

PAGES: 1466-1477

ORIGINAL PDF URL: <https://dergipark.org.tr/tr/download/article-file/898532>

Araştırma Makalesi / Research Article

Farklı İş Yükleri Altında NoSQL Sistemlerinin Performans Analizi

Yaşar DAŞDEMİR^{1*}, Burak Cem KARA²

¹Erzurum Teknik Üniversitesi, Bilgisayar Mühendisliği, Erzurum, Türkiye

²İskenderun Teknik Üniversitesi, Elektrik-Elektronik Mühendisliği, Hatay, Türkiye

(ORCID: 0000-0002-9141-0229) (ORCID: 0000-0002-1843-6182)

Öz

NoSQL veritabanı sistemleri gün geçtikçe daha hızlı bir şekilde büyük veri uygulamaları için yaygın olarak kullanılan bir veri platformu haline gelmektedir. Farklı NoSQL çözümleri arasında en iyi seçimi yapabilmek için belirli parametreler ışığında düşük veya yüksek veri yüküne sahip iş yükleri kullanılarak, bu sistemlerin zayıf ve güçlü yönlerinin analiz edilmesi gerekmektedir. Bu analiz için çalışmamızda Yahoo Bulut Hizmet Kıyaslama Aracının sunduğu 6 farklı iş yüklerini kullandık. Çalışmamızda MongoDB ve CassandraDB veritabanı sistemlerinin güncel sürümlerini kullandık. Ayrıca literatür araştırmalarımızda da sıklıkla karşılaştığımız bu iki NoSQL sisteminin son geliştirmelerle beraber kazandıkları yeni yeteneklerin performans analizlerini ortaya koyduk. Bununla beraber popülerliği Çizge (Graph) veri tabanları dünyasında günden güne artan OrientDB'yi, MongoDB ve CassandraDB gibi iki öncü sistemle karşılaştırıp, bu yeni veri tabanı sisteminin NoSQL dünyasına kattıklarını gözler önüne serdik. Elde edilen sonuçlara göre, MongoDB ve Orient DB'nin düşük yükler altında saniye de gerçekleştirdiği iş miktarında CassandraDB'ye göre çok iyi performans gösterdiği, gecikme sürelerinde ise OrientDB'nin, MongoDB'nin de üzerinde bir performans sergilediğini gözlemledik.

Anahtar kelimeler: Büyük Veri, NoSQL, YCSB, MongoDB, CassandraDB, OrientDB.

Performance Analysis of NoSQL Systems Under Different Workloads

Abstract

NoSQL database systems are becoming a widely used data platform for big data applications. In order to make the best choice among different NoSQL solutions, it is necessary to analyze the weaknesses and strengths of these systems by using workloads with low or high data load in the light of certain parameters. In this study, we used 6 different workloads of the Yahoo Cloud Service Benchmark Tool. We used current versions of MongoDB and CassandraDB database systems in our study. In addition, we have introduced the performance analyzes of these two NoSQL systems, which we frequently encounter in our literature research, with the latest developments. However, we have shown that the popularity of OrientDB, which increases day by day in the world of graphical databases, is compared with two leading systems such as MongoDB and CassandraDB, and this system is added to the world of NoSQL. According to the results, we observed that MongoDB and Orient DB performed very well in terms of CassandraDB in the amount of work carried out under low loads, while OrientDB performed well above MongoDB in latency periods.

Keywords: Big Data, NoSQL, YCSB, MongoDB, CassandraDB, OrientDB.

1. Giriş

Endüstri 4.0 ile birlikte günlük hayatımıza hızlı giriş yapan siber ekosistemindeki veri sirkülasyonu, büyük miktarda veri işlenmesinin zorluklarını beraberinde getirmiştir. Bu dönemde organizasyonların dijital ortamdaki bilgi ve verilere ulaşabilme ihtiyacı ve çabası her geçen gün katlanarak artmaktadır. Bu gelişim ve değişimler beraberinde bazı yapısal sorun ve karmaşaları da meydana getirmişlerdir. Siber

*Sorumlu yazar: yasar.dasdemir@erzurum.edu.tr

Geliş Tarihi: 01.04.2019, Kabul Tarihi: 23.09.2019

ortamdaki aygıtlar arasında yaşanan bu değişim ve gelişim, verilerin modellenerek saklanması ve dolayısıyla veritabanı kullanımını zorunlu kılmaktadır. Son yıllarda teknolojinin gelişmesiyle birlikte ortaya çıkan büyük veri kavramı depolama sistemlerinin altyapısının geliştirilmesi ihtiyacını ortaya çıkarmıştır [1]. Bu ihtiyaç sonucunda karar verme ve bilgi keşif faaliyetlerini desteklemek için büyük miktarlarda veriyi etkin bir şekilde kullanmayı amaçlayan teknolojilerin ve yöntemlerin geliştirilmesi ve uygulanması gerekmektedir [2-4]. Son iki yılda, Twitter, Facebook, vb. Gibi sosyal medyanın yaygın kullanımı bugün mevcut toplam verinin yaklaşık %90'ını oluşturuyor ve bu büyük miktarda potansiyel olarak önemli veriler çeşitli dağıtılmış yerlerde heterojen biçimlerde saklanıyor [5]. Sugam Sharma ve arkadaşları büyük veriyi bir petabyte büyüklüğünden öteye taşıyabildiğini iddia eden altı veri modelini ve sorgulama sistemlerinin mimarilerini test ederek, incelenen veri modellerinin yüksek kullanılabilirlik, yüksek ölçeklenebilirlik, sorgu dili gibi vs özelliklerini karşılaştırıp, bir tabloda anlatmıştır [5]. NoSQL veritabanları, geleneksel ilişkisel veritabanları tarafından ele alınamayan web tabanlı uygulamanın performans ve ölçeklenebilirlik gereksinimlerini karşılamak için tasarlanmıştır. Dynamo, Google Dosya Sistemi (Google File System, GFS), Bigtable ve Hadoop gibi büyük ölçekli NoSQL'e odaklanan NoSQL veritabanlarının BASE özelliklerinin analitik incelemesinde yapılmıştır [6]. Büyük verileri Hadoop benzeri yazılımlar ile birden fazla bilgisayarda saklayabiliriz ve yönetebiliriz [7,8]. Hadoop teknolojisinin performansını değerlendirmek için geliştirilen HIBench teknolojisi detaylı analizlerde kullanılmaktadır. Colaso ve arkadaşları detaylı bir kıyaslama çalışması yaparak NoSQL teknolojilerini karşılaştırmışlardır [9].

MongoDB, Cassandra, Hbase, OrientDB ve çok daha fazlası dâhil olmak üzere birçok NoSQL veri tabanı sistemi çözümleri mevcuttur. Bu sistemlerden hangisinin sizin uygulamanız için uygun olduğuna karar vermek zor olabilir [10]. Veri artışına bağlı olarak geliştirilen teknolojilerin ne ölçüde başarılı olduğu belirli analizler yapılarak değerlendirilmektedir, Yahoo Cloud Serving Benchmark (YCSB) projesinin amacı, farklı mimari ve sisteme sahip olan bu veritabanlarının performanslarını kıyaslamak ve karşılaştırmak için bir çerçeve (framework) ve ortak bir iş yükü seti geliştirmektir. Brian F. Cooper ve arkadaşları, YCSB kıyaslama aracını testlerinde kullanma sebebi olarak, iş yüklerinin sistemlere kolay uyarlanması ve genişletilebilir bir yapıya sahip olması olarak anlatmaktadır [11]. NoSQL sistemler sürekli olarak gelişmekte ve dolayısıyla kısa zaman dilimleri içerisinde NoSQL ve Büyük Veri kavramları üzerine yazılan makalelerin, yayınların ve araştırmaların sonuçları ve yazarların değerlendirmeleri geçerliliğini yitirmektedir. 2015 yılında yapılan bir çalışmada, bazı NoSQL sistemlerin performansları test edilmiş ve sürekli olarak gelişen ve yeni sürümleri yayınlanan NoSQL veritabanı sistemleri ile ilgili geçmişte yapılan değerlendirmelerin eskimiş olduğunu vurgulayarak, o dönem için NoSQL sistemlerinin yeni ve güncel bir değerlendirmelerini ortaya koymuşlardır [12-15]. Sonuç olarak incelenen tüm bu çalışmalardan yola çıkarak, bu çalışmada sürekli olarak değişen ve gelişen NoSQL ve Büyük Veri kavramlarına yeni bir bakış açısı katabilmek ve güncel NoSQL veritabanı sistemlerinin performans analizlerini YCSB kıyaslama aracı ile ortaya koyarak, bilişim dünyasının görüşüne sunulmuştur.

1.1. Büyük Veri

Meta Group (şimdiki adı “Gartner Group”) analisti DougLaney, 2001 yılındaki bir araştırma raporunda büyük veriyi 3 boyutlu olarak 3V ile tanımlamıştır. Bu 3V: hacim (volume), hız (velocity) ve çeşitlilik (variety).

Hacim: Büyük verideki veri kümelerinin boyutlarını tanımlamakta kullanılmaktadır. 1.0×10^{100} değerine googol denmektedir.

Hız: Hız, verinin elde edilme hızını göstermektedir. Örneğin Twitter üzerinden her dakika da atılan twit’lerin gerçel-zamanlı işlenmesi ve depolanması için hız önemli bir faktördür.

Çeşitlilik: Çeşitlilik, büyük veri içerisindeki veri yapısı çeşitliliğini göstermektedir. Bu çeşitlilik videolardan, müzik kliplerinden, e-posta mesajlarından oluşmaktadır.

Daha sonraları bu “V” grupları 4V (3V’ye ek olarak Gerçeklik-Veracity), 5V(4V’ye ek olarak Değer-Value) [12], 7V (5V’ye ek olarak Değişkenlik-Variability, Görselleştirme-Visualization), 10V (7V’ye ek olarak Geçerlik-Validity, Mekansal-Venue, Sözcüksel-Vocabulary, Belirsizlik-Vagueness) olarak farklı sayıda V özellikleri ile kullanılmıştır.

Büyük Veri (Big Data) terimi nispeten son yıllarda ortaya çıksa da, detaylı veri analizi için büyük miktarlarda bilgi toplama ve depolama işlemi uzun yıllardır kullanılmaktadır. Büyük verinin kullanılabilmesi belirli bir ölçeklendirmeye ihtiyaç duymaktadır. Bu bilgiler ışığında, ölçeklendirilemeyen veri yönetimi zorlaştırmaktadır[1,16].

1.2. NoSQL veritabanı yönetim sistemleri

Yıllar içerisinde bilgisayar ağlarının daha da büyüyerek karmaşıklaşması ve iletişim teknolojilerinin gelişimi veri depolama sistemleri için bazı ihtiyaçlar ortaya çıkartmaktadır. Bu hızlı gelişim ile Sensör Ağları, Sosyal Medya Ağları, Dijital Kütüphaneler ve Arşivler, Multimedya Koleksiyonları ve Web Veri Hizmetleri gibi kullanılan çeşitli bilgisayar sistemleri ürettikleri bilgiler ile ortaya büyük bir veri oluşturmaktadır [2]. Bu durum çok büyük verilerin üretildiği, bu verilerin saklanması, işlenmesinin ve yönetilmesinin çok önemli olduğu bir tasarıma ihtiyaç duymaktadır. Performans ve esneklik ihtiyacının giderek arttığı bu veri ortamında ilişkisel veritabanı yerine yeni bir veritabanı tasarımı ortaya çıkmaktadır. Bu yeni mimarinin ismi Not-Only SQL (NoSQL) olarak adlandırılmaktadır. NoSQL yani ilişkisel olmayan veritabanları bu açığı kapatmaktadır[1-3]. Ayrıca Büyük Veri yapısını yöneterek ağ yapısındaki işleri kolaylaştırmaktadır.

Veritabanı sistemleri bilgisayar sistemi üzerinde veri tutarlılığını koruyarak sistemi maliyet ve hız açısından dengelerler [9]. Ancak ağ sistemimizde artan çok yönlü veri akışı ile ilişkisel veritabanları daha sınırlayıcı ve esneklikten uzak bir yapı içerisinde olduğu anlaşılmıştır. Bu açıdan, büyük verileri işleyen ve geniş ağa sahip veri bilimcileri büyük veri kavramında ihtiyaç duyulan yapıyı ilişkisel olmayan veri tabanları ile çözüme ulaştırmışlardır. İlişkisel veri tabanlarında bölünmezlik, tutarlılık, izolasyon ve dayanıklılık kavramları temel özelliklerdir [11]. Ancak çoklu yönlü veri akışı bu klasik özelliklerin geliştirilmesi ve yeni veritabanı ortaya çıkartılması ihtiyacı doğmuştur. Ayrıca Büyük Veri patlaması ilişkisel olmayan veri tabanlarının popülerliğinin arkasında asıl katalizördür. Geleneksel veritabanı yönetim sistemleri (Database Management System, DBMS), önceden tanımlanmış şema ile yapılandırılmış veriler için tasarlanmıştır. Dolayısıyla, ilişkisel model (Relational DBMS, RDBMS), yaygın olarak büyük veri olarak bilinen yarı yapılandırılmış, yapılandırılmamış (unstructured) veya diğer veri formlarıyla uğraşmayı çok zor bulmaktadır. Böylece verilerin ölçeklenememesi, esneklik dışı durumlar, düşük performans ve maliyet açısından eksiklikleri bulunan ilişkisel veri tabanlarının yerini NoSQL olarak adlandırılan ilişkisel olmayan veri tabanları almıştır.

NoSQL, modern uygulamaların oluşturulmasında sunulan taleplere cevap olarak geliştirilen çok çeşitli farklı veritabanı teknolojilerini kapsamaktadır. Bir zamanlar sınırlı bir izleyici kitlesine hizmet veren bilgisayar uygulamaları, artık her zaman ulaşılabilir olması gereken, birçok farklı cihazdan erişilebilen ve dünya çapında milyonlarca kullanıcıya ölçeklendirilmiş hizmetler olarak kullanılmaktadır. Kurumlar artık büyük sunucular ve depolama altyapısı yerine açık kaynaklı yazılım ve bulut bilişim kullanarak büyük ölçekli mimariye yönelmektedir.

Veri yapısına göre NoSQL türleri doküman tabanlı, Anahtar-Değer tabanlı, Çizge tabanlı ve Sütun tabanlı veritabanı olarak dört kısımdan oluşmaktadır. Doküman tabanlı veri tabanları esnekler. Doküman olarak adlandırılan nesnelerde saklanan veriler bir anahtara karşılık gelirler. Doküman veritabanlarının, her anahtarı bir dokümana karşılık gelir [17]. Belgeler birçok farklı anahtar değer çiftini, anahtar dizisi çiftlerini veya iç içe geçmiş belgeleri içerebilir. Grafik tabanlı sistemde ise, sosyal medya bağlantıları gibi veri ağları hakkında bilgi depolamak için kullanılmaktadır. Anahtar değer tabanlı sistemler en basit NoSQL veri tabanlarıdır. Veri tabanındaki her bir öge, değeriyle birlikte anahtar bir sembol olarak birlikte depolanır.

Bu çalışmada doküman yönelimli veri modelini kullanan MongoDB, geniş-sütun yönelimli veri modelini kullanan CassandraDB ve grafik yönelimli veri modeline sahip OrientDB kullanılmıştır (Tablo 1).

Tablo 1. NoSQL Veritabanlarının Özellikleri [18]

	Performans (Performance)	Ölçeklenebilirlik (Scalability)	Esneklik (Flexibility)	Karmaşıklık (Complexity)	İşlevsellik (Functionality)
MongoDB	Yüksek	Yüksek	Yüksek	Düşük	Düşük
CassandraDB	Yüksek	Yüksek	Orta	Düşük	Çok Az
OrientDB	Yüksek	Yüksek	Yüksek	Yüksek	Çizge teorisi ile tanımlanır

MongoDB, NoSQL Verilerini doküman (belge) biçiminde saklayabilmek için kullanabileceğimiz ölçeklenebilir (scalable) bir veritabanı biçimidir [19,20]. MongoDB bize kendisini, geliştirme ve ölçekleme kolaylığı için tasarlanmış açık kaynak, belge-yönelimli(document-oriented) veritabanı olarak tanıtmaktadır. MongoDB’de her kayıt, aslında bir dokümandır. Dokümanlar MongoDB’de JavaScript Object Notation (JSON) benzeri Binary JSON(BSON) formatında saklanır. BSON belgeleri, sakladıkları elemanların sıralı bir listesini içeren nesnelerdir. Her bir eleman, bir alan adı ve belirli tipte bir değerden oluşur. NoSQL verilerini saklayıp, son kullanıcının anlık olarak sorgulayabileceği bir sistem için MongoDB tercih nedeni olabilir. Örnek verirsek 500 milyon kaydı olan bir veri kümesi düşünelim. Bu veri kümesinde son kullanıcı tarafından anlık sorgulama ihtiyacı olduğu durumlarda MongoDB gibi bir NoSQL veritabanını kullanabiliriz. Genel olarak anlık sorgulamalarda MongoDB tercih edilen bir sistemdir. Son yıllarda binlerce şirket, yeni uygulamalar geliştirmek, müşteri deneyimini iyileştirmek, hızlı pazarlama sürelerini kısaltmak ve maliyetleri en aza indirmek için MongoDB’yi kullanmaktadır [21].

Apache CassandraDB en yaygın kullanılan geniş-sütun (wide-column) depolama yapısına sahip veritabanıdır. Kendi sorgu dilini (CassandraDB Query Language, CQL) kullanmaktadır. NoSQL sistemleri için ortak sorgu işletim katmanı üzerinde çalışmalarda yapılmıştır[20]. Ayrıca Hadoop (Map-Reduce desteği ile) entegrasyonuna da sahiptir. Günümüzde büyük şirketlerin (Microsoft, IBM, Facebook, Apple gibi) çoğu CassandraDB yazılımını kullanmaktadır.

Popülerliği günden güne artan Çizge veri tabanları dünyasında OrientDB yetenekleri ile öne çıkıyor. Piyasada en çok bilinen Neo4J’den farklı olarak, OrientDB açık kaynaklı. Ayrıca destek almak istediğinizde Orient Technologies firması tarafından desteklenen kurumsal bir sürümü de mevcuttur. OrientDBşirketi, IBM ve Tokyo Institute of Technology tarafından yapılan bağımsız bir kıyaslama çalışmasında, OrientDB’nin tüm iş yükleri arasındaki grafik işlemlerinde Neo4J’den 10 kat daha hızlı olduğunu gösterdiği bilgisini paylaşmaktadır. OrientDB aslında sadece bir Çizge veritabanı değil, aynı zamanda doküman tabanlı bir NoSQLveritabanıdır. OrientDB’nin,TinkerpopGremlin dilini desteklemesi, ACID, Multi-Master yineleme, REST API ve en önemlisi SQL desteği olması bizlere NoSQL veritabanı sistemleri dünyasında ciddi bir yeri olduğunu göstermektedir. OrientDB’yi sevmemizin bir başka nedeni de dağıtık çalışabilmek için arka tarafta Hazelcast’den faydalanmasıdır.

2. Materyal ve Metot

Teknolojinin ilerlemesi ve hızlı gelişmesiyle internet dünyasının gün geçtikçe artan verisini depolayabilmek ve Büyük Veri’yi anlamlı ve yönetilebilir bir yapıya kavuşturmak için ortaya birçokNoSQL uygulaması çıkmıştır. Bu kadar çok NoSQL uygulamasından hangisini senin uygulaman için uygun olduğuna karar vermek zor olabilir, çünkü sistemler arasındaki özellikler farklılıklar gösterebilmektedir. Bu farklılıkları karşılaştırmak ve bir sistemin diğerine göre performansını tespit ederek, hangi NoSQL sistemini kullanacağına karar vermek kolay bir süreç değildir. İşte bu noktada YahooCloudServingBenchmark (YCSB) gibi kıyaslama araçları devreye girmektedir[11,22]. Yahoo tarafından tasarlanan bir framework olan YCSBçeşitli tiplerdeki NoSQL veritabanı sistemlerinin performansını ölçmek ve karşılaştırmak için kullanılan açık kaynaklı bir yazılım paketidir. YCSB projesinin amacı, farklı "anahtar-değer" ve "bulut" hizmet mağazalarının performansını değerlendirmek için bir framework ve ortak bir iş yükü seti geliştirmektir. Bu kıyaslama aracı ile MongoDB, CassandraDB,Hbase ve çok daha fazlası dâhil olmak üzere birçok NoSQL veritabanı sistemlerinin performansını ölçmek ve karşılaştırmak mümkün olabilmektedir. YCSB, okuma, yazma, güncelleme ve arama gibi farklı senaryoları birleştiren altı standart iş yüküyle birlikte gelir[16]. YCSB, günlük hayattaki ihtiyaçların kullanım durumlarını simüle etmek için çeşitli oranlarda CRUD (Oluşturma, Okuma, Güncelleme, Silme) işlemlerini gerçekleştirmek üzere tasarlanmıştır [16,23]. Bu çalışmada, YCSB, düşük veri yükleri altında, gecikme süresi ve saniyede yapılan işlem miktarlarını değerlendirerekbulut hizmet sistemlerinin performansını ölçmek için kullanılmıştır. Tablo 2’de NoSQL sistemlerini karşılaştırırken kullandığımız iş yüklerinin, kullanım parametre oranları ve açıklamaları mevcuttur.

Tablo 2. Kıyaslama (benchmark) parametreleri

No	YCSB	Kıyaslama parametreleri
1	Workload A: Ağır iş yükü güncelleme	%50 okuma ve %50 güncelleme. Bu iş yükü 50/50 oranında okunup yazılmıştır. Örneğin, en son eylemleri kaydeden bir kayıt günlükleri gibi.
2	Workload B: İş yükünü çoğunlukla okuma	%95 oku ve %5 güncelle. Bu iş yükü %95 okuma ve %5 yazma yapmaktadır. Örneğin; fotoğraf etiketleme ve etiket ekleme güncelleme olarak, etiketleri okumak ise okuma olarak kullanılır.
3	Workload C: Sadece okuma	%100 okuma işlemleri. Bu iş yükü %100 okuma yapar. Örneğin herhangi bir yerde üretilen kullanıcı profillerini önyükleme
4	Workload D: Son iş yükünü okuma	%95 okuma ve %5 ekleme. Bu iş yükünde yeni kayıtlar eklenir ve çoğunlukla eklenen kayıtlar kullanılmaktadır. Örneğin, kullanıcı durum güncellemeleri gibi.
5	Workload E: Kısa aralıklar	%95 tarama ve %5 ekleme. Bu iş yükünde bireysel kayıtlar yerine kısa kayıt aralıkları sorgulanır. Örneğin her bir taramanın verilen bir iş parçasıdaki yazıları taraması.
6	Workload F: Oku-değiştir-yaz	%50 oku ve %50 oku-değiştir-yaz. Bu iş yükünde istemci bir kayıt okuyacak, onu değiştirecek ve değişiklikleri tekrar geri yazacak. Örneğin; kullanıcı kayıtlarını okumak, değiştirmek ve kullanıcı aktivitelerini kaydetmek.

3. Bulgular ve Tartışma

Bu çalışmada NoSQL veritabanı sistemleri uygulamalarından MongoDB sürüm 3.6.3, CassandraDBsürüm3.11.2 ve OrientDBsürüm 2.04 programları kıyaslama aracı YCSB-0.12.0 sürümünün Workloads testlerine tabi tutulmuş olup, elde ettiğimiz performans sonuçlarının detaylı karşılaştırılması yapılmıştır. Her bir veritabanına 1.000.000 kayıt yüklenmiştir. Ayrıca her bir veritabanından 1.000.000 operasyon gerçekleştirilmesi istenmiş olup, bu operasyonları gerçekleştirirken 500, 2000, 4000, 6000, 8000 ve 10000'e ayarlanan hedef iş hacimleriyle, her bir veritabanından hedeflenen bu iş hacimlerini gerçekleştirilmesi istenmiştir.

Bu çalışmada, 16 GB RAM, Intel Core i7-6700HQ CPU @ 2.60Hz (8 CPUs) bilgisayar konfigürasyonu kullanılmıştır. Depolama alanı olarak 128 GB 540 MB/s okuma, 130 MB/s yazma hızına sahip ssd disk kullanılmıştır. İşletim sistemi olarak Windows 10, 64 bit-lık sistem kullanılmıştır. İş yüklerini veri tabanlarına uygularken, kullanılan 6 iş yükünde benzer veri kümesine sahip olduğu için, veritabanı boyutunu tutarlı tutmak ve daha verimli ve düzgün karşılaştırmalar elde etmek için, çalışma sırasında aşağıdaki adımlar kullanılmıştır:

1. Workload A parametre dosyasını kullanarak verileri, veritabanına yükle;
./bin/ycsb load basic -P workloads/workloada
2. Veriler, veritabanına yüklendikten sonra aşağıda belirtilen sırayla iş yüklerini çalıştır.
 - 2.1. Workload A iş yükünü çalıştır.
./bin/ycsb run basic -P workloads/workloada
 - 2.2. Workload B iş yükünü çalıştır.
./bin/ycsb run basic -P workloads/workloadb
 - 2.3. Workload C iş yükünü çalıştır.
./bin/ycsb run basic -P workloads/workloadc
 - 2.4. Workload F iş yükünü çalıştır.
./bin/ycsb run basic -P workloads/workloadf
 - 2.5. Workload D iş yükünü çalıştır.
./bin/ycsb run basic -P workloads/workloadd
3. Workload D iş yükünü de çalıştırdıktan sonra Workload E iş yüküne geçmeden veritabanındaki veriler silinir.
4. Veritabanı yeniden başlatıldıktan sonra, Workload E parametre dosyası kullanılarak veriler, veritabanına yüklenir.
./bin/ycsb load basic -P workloads/workloade
5. Veriler, veritabanına yüklendikten sonra Workload E iş yükünü çalıştır.
./bin/ycsb run basic -P workloads/workloade

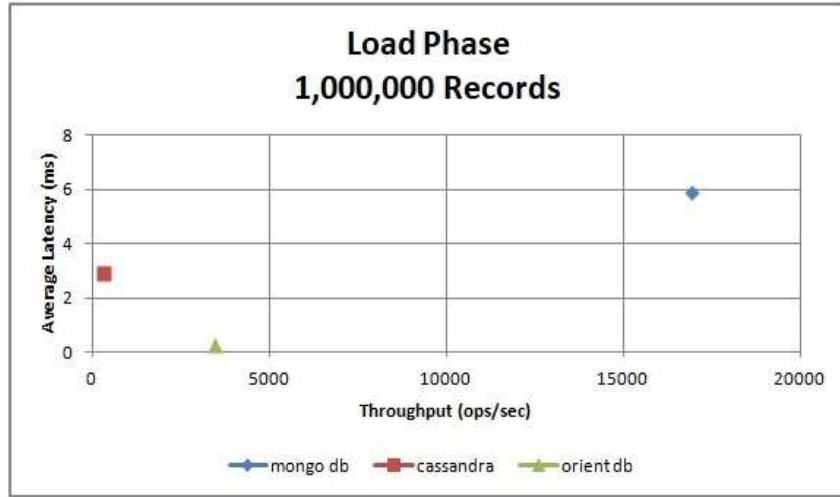
Workload İş yüklerini veritabanlarında çalıştırmak için 5 adım uygulanır.

1. Test edilecek veritabanı sisteme yüklenir.

2. Veri, test edilecek veritabanına yüklenir.
3. İş yükleri çalıştırılırken hangi parametreler kullanılacak ise bunlar belirlenir (threads, target, record count vs.).
4. Parametrelerde belirledikten sonra hangi iş yükünde veritabanı test edilecekse, o iş yükü belirlenir.
5. Seçilen iş yükü belirlenen parametrelerle birlikte çalıştırılır ve test sonuçları alınır.

3.1. Yükleme Evresi (Load Phase)

Yükleme evresinde eklenecek veriler tanımlanır. Altı iş yükünün hepsinin benzer bir veri kümesi olduğundan veritabanı boyutunu tutarlı tutmak istediğimizden dolayı bu aşamada Workload-A iş yükü veri tabanlarına yüklenmiştir. Daha sonra belirli bir sıralama düzeniyle her iş yükü için veritabanı sistemleri teste tabi tutulmuştur. Çalışmamız da her bir veritabanının performansını gözlemleyebilmek için 1 milyon veri yükledik. Şekil 1'de eklediğimiz bu 1 milyon verinin karşısında veritabanı sistemlerinin gösterdiği performans değerlerini ortaya koyduk. En iyi yükleme süresine MongoDB ulaşırken, MongoDB'yi OrientDB ve ardından CassandraDB takip etti, saniyede gerçekleştirilen işlem sayısında da yine MongoDB diğer iki veritabanını sistemine oranla ciddi bir performans farkı ortaya koyduğunu grafik de gözlemliyoruz. Yine grafiği incelediğimiz de bu evrede en düşük ve en tutarlı gecikme sürelerini OrientDB'nin yakaladığını gözlemliyoruz, bunun yanında MongoDB'nin ise bu alanda en kötü performansı ortaya koyduğunu gözlemliyoruz (Şekil 1).



Şekil 1. Yükleme Evresi (1 Milyon kayıt)

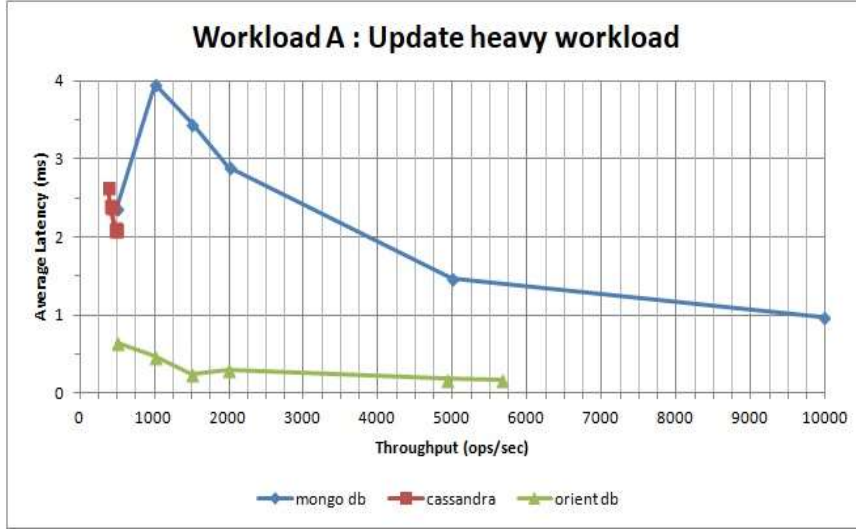
3.2. İşlemsel Evre (Transaction Phase)

Bu evrede bir önceki yükleme evresinde gerçekleşen veri kümelerinin yükleme işlemi sonrasında iş yüklerinin yürütme aşaması gerçekleştirilir. İşlemsel evrede sıralı algoritmamızın kendi içerisinde tutarlı performanslarını yakalayabilmek için YCSB'nin oluşturduğu sıralama algoritmasıyla her bir sistem gerekli testlere tabi tutulmaktadır. Ayrıca yine bu evrede sistemlerin performanslarını daha iyi gözlemleyebilmek adına, sistemlerin saniye de gerçekleştirdiği işlem sayılarına hedefler konularak testler yapılmıştır.

3.2.1. Workload A: Ağır İş Yükü Güncelleme

Workload A, yüzde 50 oranında okuma ve yüzde 50 oranında yazma karışımı içeren yoğun güncelleme işlemlerinin gerçekleştiği bir senaryodur. Bir kullanıcı oturum açıldığında (örneğin bir web uygulamasında), kullanıcı oturumu kapatana kadar veya oturum zaman aşımına uğrayana kadar geçen sürede oluşan oturumla ilgili tüm Oturum verilerini (kullanıcı profili bilgilerini, mesajları, kişiselleştirilmiş verileri ve temaları, önerileri, hedefli promosyonları ve indirimleri) ana bellekte veya oturum deposunda saklar. Workload A iş yükünde, bir kullanıcının son eylemlerini kaydeden bir oturum

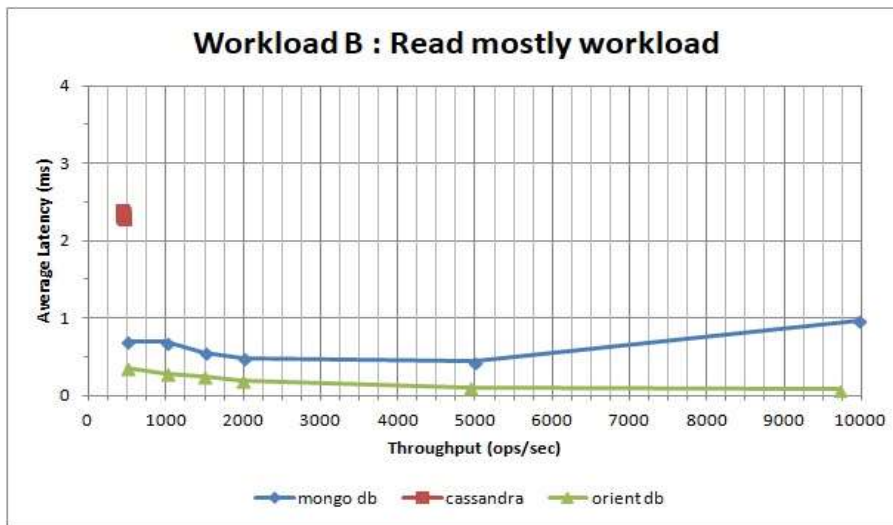
deposudur. Şekil 2'deki WorkloadA test sonuçların incelediğimiz de, ağır güncelleme senaryoları altında OrientDB'nin ortalama gecikme süresi bakımından en iyi performansı gösterdiği açıkça görülmektedir. MongoDB'nin ise gecikme süresi bakımında her ne kadar CassandraDB ve OrientDB'ye kıyasla başarılı bir performans gösteremese de saniye yapılan iş miktarlarında, koyulan 6 hedefi de büyük oranda tutturduğunu gözlemlemekteyiz. CassandraDB'nin ise saniye yapılması istenen iş miktarları hedeflerinin tutturmak da kötü bir performansgösterdiğine tanık olduk (Şekil 2).



Şekil 2. Workload A: Ağır İş Yüğü Güncelleme

3.2.2. Workload B: İş Yükünü Çoğunlukla Okuma

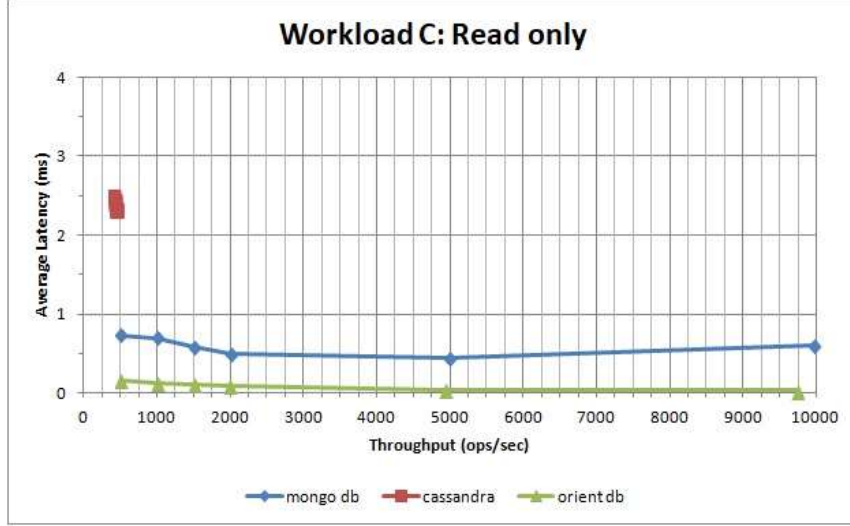
Workload B iş yükü % 95 oranında okuma ve % 5 oranında güncelleme işlemlerinden oluşur. Fotoğraf etiketleme olayını bu iş yüküne örnek verebiliriz. Etiketleme bir güncelleme işlemidir fakat işlemin çoğu etiketlerin okunmasıdır. Workload B iş yükü sonuçlarını incelediğimiz de, OrientDB ve MongoDB'nin, saniyede gerçekleştirilmesi istenen iş hacimlerini başarılı bir şekilde tutturduklarını ve ayrıca, gecikme sürelerinde de yine birbirine yakın ve başarılı performanslar yakaladıklarını gözlemlemekteyiz. CassandraDB ise Workload A'dakine benzer vasat bir performans gösterdiğini gözlemlemekteyiz. (Şekil 3).



Şekil 3. Workload B: İş Yükünü Çoğunlukla Okuma

3.2.3. Workload C: Sadece Okuma

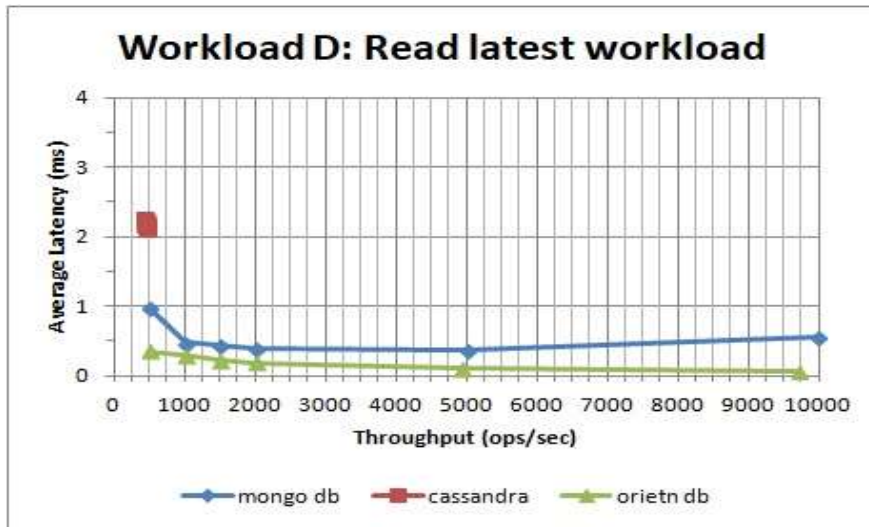
Workload C, %100 oranında okuma işlemine sahiptir. Profillerin başka yerlerde oluşturulduğu kullanıcı profili ön belleği (ör. Hadoop) bu iş yüküne örnek olarak verilebilir. Workload C sonuçları, bize Workload B sonuçlarının birkaç istisna ile tutarlı olduğunu gösterir (Şekil 4). Bu iş yükünde de OrientDB ve MongoDB'nin gerçekleştirilmesi istenen iş hacimlerini başarılı bir şekilde tutturdıklarını ve CassandraDB'nin ise yine ilk 2 iş yükündekine yakın bir performans sergileyerek iş hacminin hedefini artırırken daha yüksek verim veya düşük gecikme sürelerini elde edemedik.



Şekil 4. Workload C: Sadece Okuma

3.2.4. Workload D: Son İş Yükünü Okuma

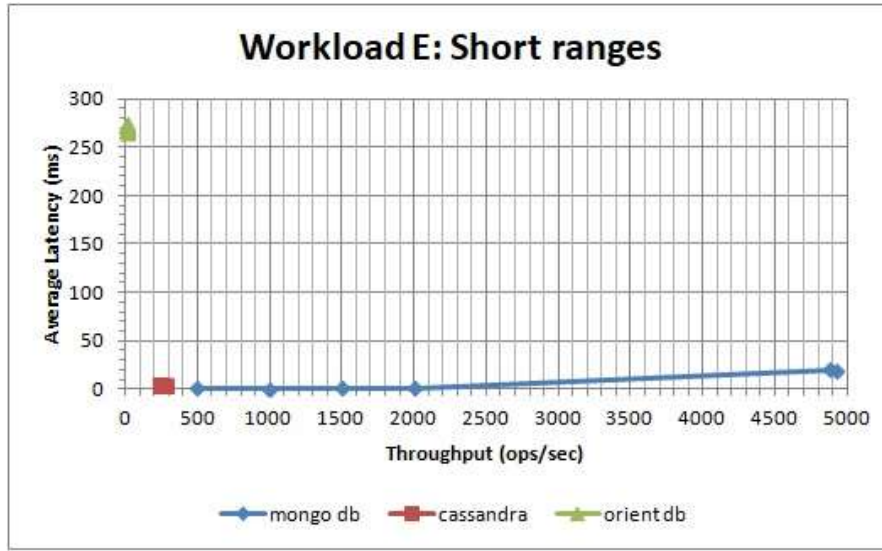
Workload D, %95 oranında okuma ve %5 oranında ekleme işleminin gerçekleştirildiği bir iş yüküdür. Günümüzde popülerlikleri git gide artan sosyal medya platformlarındaki, kullanıcıların profillerinde paylaştıkları metin, fotoğraf, video ve animasyonlu GIF vs. gibi paylaşımlarına sürekli veya belirli periyotlarla yenilerini eklemektedirler. Her zaman en son eklenen kayıtlar en popüler olanıdır. Bu iş yükü senaryosu, kullanıcı durumu güncellemelerini veya en son yayını okumak isteyen kullanıcıları simüle eder. OrientDB ve MongoDB'nin bu iş yükündeki CassandraDB'ye oranla daha yüksek verimle çalıştığını görmekteyiz (Şekil 5). OrientDB gecikme sürelerinden 0,5 ms altında kalarak gayet başarılı bir performansa imza attığını gözlemlemekteyiz.



Şekil 5. Workload D: Son İş Yükünü Okuma

3.2.5. Workload E: Kısa Aralıklar

Workload E iş yükü %95 oranında tarama ve %5 oranında ekleme işlemlerinden oluşur. Bu iş yükünde, bireysel kayıtlar yerine kısa kayıtlar sorgulanır ve işlemlerin % 95'i, kayıt aralıkları üzerinde Arama / Sorgulama gerçekleştiren tarama işlemleridir. Bu iş yüküne, online bir forum da yapılan çevrimiçi bir konuşmadan sonra, iş parçacığı kimliği tarafından kümelenmiş belirli bir iş parçacığındaki iletileri alan tarama işlemlerinin yapıldığı senaryoları örnek olarak verebiliriz. Şekil 6'daki tabloyu incelediğimizde, üç veritabanında bu iş yükünde verimlerinin düşük kaldığını gözlemliyoruz. MongoDB'nin saniyede gerçekleştirilmesi istenen 2000 hedefine kadar verimli çalışabildiği fakat 5000 ve 10000 hedeflerinde çok yüksek gecikme sürelerine çıktığını görmekteyiz. Diğer iş yüklerinde başarılı performanslar sergileyen OrientDB'nin ise bu iş yükünde hem gecikme süreleri olarak hem de saniyede yaptığı işlem sayılarında çok kötü bir performans sergilediğini görüyoruz, CassandraDB'nin ise diğer iş yüklerinde sergilediği performansların altında bir görüntü çizdiğini grafiğimizde gözlemliyoruz.

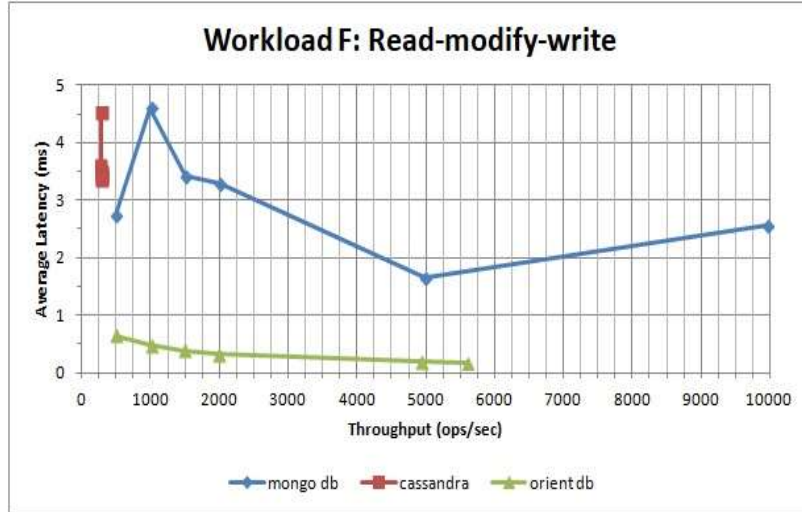


Şekil 6. Workload E: Kısa Aralıklar

3.2.6. Workload F: Oku-Değiştir-Yaz

Workload F iş yükünde, istemcinin bir kaydı okuduğu, değiştirdiği ve değişiklikleri geri yazdığı senaryo ele alınmaktadır. Kullanıcı kayıtlarının kullanıcı tarafından okunup değiştirildiği veya kullanıcının etkinliğini kaydettiği, kullanıcı veri tabanları işlemlerini bu iş yüküne örnek olarak verebiliriz. MongoDB ve OrientDB'nin bu iş yükünde de saniye de yapılması hedeflenen iş hacmini tutturmada en tutarlı performansları gösterdiğini görmekteyiz. Sadece OrientDB'nin 10000 hedefini tutturamadığını fakat gecikme sürelerinde hemen hemen bütün hedeflerde 0.5 ms'nin altında kalarak müthiş bir verimlikle çalıştığını söyleyebiliriz (Şekil 7).

A, B ve C iş yüklerinin genel sonuçları Tablo 3 ile D, E ve F iş yüklerinin genel sonuçları Tablo 4 ile sunulmuştur.



Şekil 7. Workload F: Oku-Değiştir-Yaz

Tablo 3. İş Yükleri Genel Sonuçları (A,B,C)

(T:Throughput, Avg: AverageLatency, -t:hedef iş hacmi, m1:min,m2:maks,sd:standart sapma)

DB	(-t)	WORKLOAD A					WORKLOAD B					WORKLOAD C				
		T	Avg	m1	m2	sd	T	Avg	m1	m2	sd	T	Avg	m1	m2	sd
MongoDB	500	499,85	2,38	0,98	3,96	1,14	499,86	0,7	0,44	0,97	0,19	499,85	0,74	0,44	0,74	0,11
	1000	999,54	3,96				999,51	0,69				999,52	0,69			
	1500	1499,19	3,46				1499,03	0,55				1499,03	0,58			
	2000	1998,26	2,90				1998,18	0,48				1998,23	0,49			
	5000	4989,84	1,48				4991,53	0,44				4989,42	0,44			
	10000	9961,64	0,98				9963,93	0,97				9959,86	0,60			
CassandraDB	500	376,60	2,64	2,10	2,41	0,22	418,89	2,37	2,28	2,4	0,04	429,71	2,31	2,31	2,49	0,07
	1000	417,70	2,38				427,07	2,31				406,96	2,44			
	1500	467,69	2,12				424,40	2,32				428,55	2,32			
	2000	471,14	2,10				433,67	2,28				415,12	2,40			
	5000	473,53	2,10				418,57	2,36				399,54	2,49			
	10000	411,67	2,41				412,56	2,4				431,24	2,31			
OrientDB	500	499,28	0,65	0,17	0,65	0,18	499,27	0,35	0,08	0,35	0,10	499,22	0,16	0,03	0,16	0,05
	1000	997,00	0,48				997,01	0,28				997,20	0,13			
	1500	1492,78	0,25				1493,27	0,24				1493,77	0,11			
	2000	1988,73	0,30				1987,71	0,2				1988,86	0,09			
	5000	4921,79	0,18				4922,61	0,10				4930,65	0,04			
	10000	5648,85	0,17				9704,30	0,08				9726,46	0,03			

Tablo 4. İş Yükleri Genel Sonuçları (D,E,F)

(T:Throughput, Avg: AverageLatency, -t:hedef iş hacmi, m1:min,m2:maks,sd:standart sapma)

DB	(-t)	WORKLOAD D					WORKLOAD E					WORKLOAD F				
		T	Avg	m1	m2	sd	T	Avg	m1	m2	sd	T	Avg	m1	m2	sd
MongoDB	500	499,84	0,98	0,37	0,98	0,22	499,87	1,29	1,17	20,44	9,84	499,87	2,74	1,66	4,60	0,98
	1000	999,54	0,48				999,55	1,17				999,51	4,60			
	1500	1499,14	0,44				1499,04	1,27				1498,95	3,43			
	2000	1998,21	0,40				1998,58	1,42				1998,25	3,29			
	5000	4994,05	0,37				4925,76	20,27				4990,79	1,66			
	10000	9969,39	0,56				4881,54	20,44				9959,56	2,56			
CassandraDB	500	454,72	2,18	2,13	2,27	0,05	237,73	4,19	3,60	4,19	0,20	283,45	3,51	3,37	4,53	0,43
	1000	465,94	2,13				276,23	3,60				287,23	3,46			
	1500	464,31	2,14				248,15	4,02				279,70	4,53			
	2000	452,87	2,19				251,59	3,95				274,61	3,62			
	5000	436,52	2,27				241,12	4,13				292,69	3,37			
	10000	442,81	2,24				249,06	4,01				289,65	3,43			
OrientDB	500	499,21	0,35	0,06	0,35	0,11	3,64	273,63	265,09	273,65	3,50	499,28	0,65	0,17	0,65	0,18
	1000	996,91	0,29				3,66	266,98				996,93	0,48			
	1500	1493,20	0,23				3,70	265,09				1492,46	0,39			
	2000	1987,69	0,19				3,79	270,78				1986,83	0,32			
	5000	4918,54	0,1				3,71	268,88				4929,56	0,2			
	10000	9688,51	0,06				3,73	273,65				5593,81	0,17			

4. Sonuç ve Öneriler

Bu çalışmada, üç farklı NoSQL veritabanı sistemini, yeteneklerini ve farklı operasyonlarda nasıl tepki verdiklerini ortaya koymak için tartıştık ve test ettik. Test ettiğimiz her bir veritabanının mimarisi ve tasarımı nedeniyle, her bir işlem için veritabanıların farklı performans sonuçlarını verdiğini gözlemledik. Elde edilen sonuçlara göre, MongoDB'nin saniye de gerçekleştirilmesi istenen hedefleri yakalamada en başarılı performans sonuçlarına ulaştığı gözlemledik, fakat bunun yanında OrientDB'ye oranla gecikme sürelerin de daha kötü bir performans gerçekleştirdiğini gördük. CassandraDB ise hemen bütün testlerde, saniye de gerçekleştirilmesi istenen hedefleri yakalamada ve her bir işlem arasındaki gecikme süresi performanslarında MongoDB ve OrientDB'nin gerisinde kalmıştır.

Kaynaklar

- [1] Storey V.C., Song I.Y. 2017. Big data technologies and management: What conceptual modeling can do. *Data Knowl. Eng.*, 108: 50–67.
- [2] Ge M., Bangui H., Buhnova B. 2018. Big Data for Internet of Things: A Survey. *Futur. Gener. Comput. Syst.*, 87: 601–614.
- [3] Khan S., Liu X., Shakil K.A., Alam M. 2017. A survey on scholarly data: From big data perspective. *Inf. Process. Manag.*, 53 (4): 923–944.
- [4] Ribeiro A., Silva A., Da Silva A.R. 2015. Data modeling and data analytics: a survey from a big data perspective. *J. Softw. Eng. Appl.*, 8 (12): 617.
- [5] Sharma S., Tim U.S., Wong J., Gadia S., Sharma S. 2014. A Brief Review on Leading Big Data Models. *Data Sci. J.*, 13: 138–157.
- [6] Chandra D.G. 2015. BASE analysis of NoSQL database. *Futur. Gener. Comput. Syst.*, 52: 13–21.
- [7] Shvachko K., Kuang H., Radia S., Chansler R. 2010. The hadoop distributed file system. in 2010 IEEE 26th symposium on mass storage systems and technologies (MSST), pp. 1–10.
- [8] White T. 2009. Hadoop: The Definitive Guide. O'Reilly Media Yahoo! Press.
- [9] Colaso A. 2018. Memory Hierarchy Characterization of NoSQL Applications through Full-System Simulation. *IEEE Trans. Parallel Distrib. Syst.*, 29 (5): 1161–1173.
- [10] Çimrin K.M., Daşdemir Y. 2018. NoSQL Database Systems: Review and Comparison. in International Conference on Artificial Intelligence towards Industry 4.0, pp. 172–177.
- [11] Cooper B.F., Silberstein A., Tam E., Ramakrishnan R., Sears R. 2010. Benchmarking cloud serving systems with YCSB. in Proceedings of the 1st ACM symposium on Cloud computing - SoCC '10.
- [12] Lourenço J.R., Cabral B., Bernardino J., Vieira M. 2015. Comparing NoSQL Databases with a Relational Database: Performance and Space. *Serv. Trans. Big Data*, 2 (1): 1–14.
- [13] Moniruzzaman A.B.M., Hossain S.A. 2013. NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison. *Int. J. Database Theory Appl.*, 6 (4): 1–14.
- [14] Shirinbab S., Lundberg L., Casalicchio E. 2019. Performance Comparison between Scaling of Virtual Machines and Containers using Cassandra NoSQL Database. in The Tenth International Conference on Cloud Computing, GRIDs, and Virtualization, pp. 93–98.
- [15] Yüzük S., Aktaş M.G., Aktaş M.S. 2018. On the Performance Analysis of Map-Reduce Programming Model on In-Memory NoSQL Storage Platforms: A Case Study. in International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism, pp. 45–50.
- [16] Shwaysh M.M. 2018. Security and Performance Comparison of NoSQL Database Systems. Çankaya Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 122s, Ankara.
- [17] Eken S., Kaya F., Sayar A., Kavak A. 2014. Doküman Tabanlı NoSQL Veritabanları: MongoDB ve CouchDB yatay ölçeklenebilirlik karşılaştırması. 7. Mühendislik ve Teknoloji Sempozyumu, pp. 1–7.
- [18] Jowan S.A., Swese R.F., Aldabrzi A.Y., Shertil M. S. 2016. Traditional RDBMS to NOSQL Database: New Era of Databases for Big Data. *J. Humanit. Appl. Sci.*, 29 (29): 83–102.
- [19] Taha I.A. 2017. A Comprehensive Comparison of NoSQL and Relational Database Management Systems. Çankaya Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 75s, Ankara.

- [20] Erden B. 2018. NoSQL Veritabanları için Ortak Sorgu İşletim Katmanının Geliştirimi. Ege Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 94s, İzmir.
- [21] Sagioglu S., Sinanc D. 2013. Big data: A review, in Collaboration Technologies and Systems (CTS). 2013 International Conference, May 2013 pp. 42–47, San Diego, CA, USA.
- [22] Hammood A., Saran M.A. 2016. Comparison Of NoSQL Database Systems : A Study On MongoDB, Apache Hbase, And Apache,” October 2016, pp. 20–23, Tekirdağ, Turkey.
- [23] Aladily A.T. 2015. The Performance-wise Comparision of the most widely used NoSQL databases. Kadir Has Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 62s, İstanbul.