

Ölçeklenebilir, Yüksek Erişilebilir ve Performanslı Bir Takip ve İzleme Sistemi Mimarisi: Karşılaştırmalı Bir Çalışma

Burak İbrahim Sevindi, Ethem Cem Özkan, Turan Bahattin Özen

TÜBİTAK BİLGEM Yazılım Teknolojileri Araştırma Enstitüsü
{burak.sevindi, cem.ozkan, turan.ozen}@tubitak.gov.tr

Özet. Bu çalışmada, takip ve izleme sistemlerinde kullanılmak üzere geliştirilen iki farklı mimari alternatif karşılaştırılmıştır. Alternatiflerden birincisi, sütun bazlı dağıtık bir NoSQL sistemine dayanmaktadır. İkinci mimari alternatif ise, dağıtık kuyruk ve önbellek sistemleri ile ilişkisel bir veritabanı çözümünün birlikte kullanılmasına dayanmaktadır. Alternatif mimariler, bulut ortamında gerçekleştirilen yük testleri aracılığıyla ölçeklenebilirlik, erişilebilirlik ve performans açısından karşılaştırılmıştır. Yapılan değerlendirmede, sütun bazlı dağıtık NoSQL sistemine dayanan alternatifin, geliştirilmekte olan takip ve izleme sistemi için daha avantajlı olduğu sonucuna ulaşılmıştır.

Anahtar Kelimeler: yazılım mimarisi, performans, ölçeklenebilirlik, erişilebilirlik

1 Giriş

Takip ve izleme sistemleri, ürünlerin geçmişteki ve mevcuttaki yerlerini kaydetmek ve sorgulamak için kullanılan sistemlerdir. Ürünler tek tek veya gruplar halinde takip edilebilirler. Ürünlerin tek tek izlendiği duruma tekil takip denir. Tekil takiple izlenen ürünlerin her biri benzersiz bir ürün numarasına sahiptir. Bu ürün numarası kullanılarak, ürünün dolaşımı sırasında uğradığı her nokta takip ve izleme sistemine tekil bildirim olarak kaydedilir. Gruplar halinde takip edilen ürünler genellikle aynı tipteki ürünlerdir ve bu ürünler için lot bazlı takip yapılır. “Lot” terimi, genellikle bir ürünün topluca üretilen bir grubuna verilen addır. Bir ürünün belirli bir zamanda yapılan 10000 adetlik üretimi bir lot olarak düşünülebilir. Bir lotta kaç tane ürün olacağı, ürünü üreten firmanın belirlediği bir mevzudur. Lot bazlı üretilen ürünlerin her birinin benzersiz bir numarasının olması zorunlu değildir, ancak bir lottaki ürünlerin topluca benzersiz bir lot numarası olur ve lot bildirimleri bu lot numarası üzerinden yapılır.

Bu çalışmada, ürünlerin tekil seviyede takibini sağlayacak platformlarda kullanılacak iki farklı mimari alternatif önerilmiştir. Bu mimari alternatifler, sisteme saniyede 30000 bildirim yapılacağı öngörüsü doğrultusunda, bu yükü kaldıracak bir performansla sahip olması amacıyla tasarlanmıştır. Sistemin, kullanıcılarının üretim, lojistik, bilgi sistemleri gibi farklı altyapıları ile entegre olacağı ve bu kullanıcıların faaliyetlerinin 7/24 devam ettiği var sayıldığında, yüksek erişilebilirlikte olması gerekmektedir.

Ayrıca, takip ve izleme sistemlerinde takip edilen ürün tipleri seneler içinde kademeli olarak artabilmektedir. Bu nedenle, sistemin başlangıç aşamasındaki kaynak ihtiyacı ile yıllar içinde tam kapasite kullanıma eriştiği noktadaki kaynak ihtiyacı arasında büyük farklar olabilmektedir. Dolayısıyla, mimari alternatiflerin ölçeklenebilir bir yapıya sahip olması gözetilmiştir.

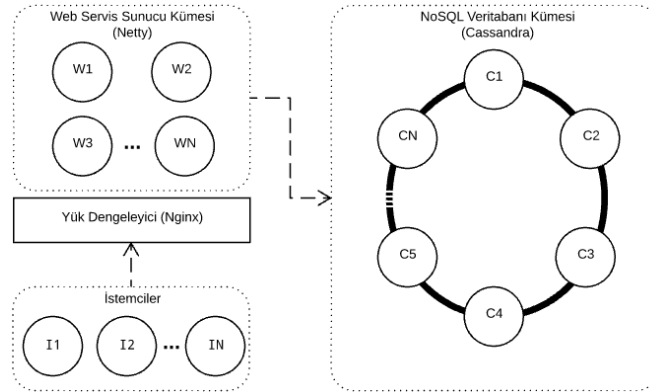
2 Alternatif Mimarilerin Yapıları

Bu bölümde önerilen mimari alternatiflerin yapıları anlatılmıştır. Alternatif mimariler, Alternatif Mimari 1 ve Alternatif Mimari 2 olarak isimlendirilmiştir. Bu bölümdeki diyagramlarda bileşenler arası bağımlılıklar kesikli oklarla gösterilmiştir. Beraberce bir küme oluşturan bileşenlerin etrafı kesikli çizgilerle çevrelenmiştir. Bu kümeler, mimarideki dağıtık (distributed) alt sistemleri temsil etmektedir (İstemciler hariç).

2.1 Alternatif Mimari 1

Alternatif Mimari 1'in yapısı Şekil 1'de gösterilmiştir.

Şekil 1. Alternatif Mimari 1



Alternatif Mimari 1, üç ana bileşenden oluşmaktadır:

- Web Servis Sunucu Kümesi: İstemcilerin bildirim gönderirken kullandıkları Web servisleri sunan Web sunucu kümesidir. Web sunucusu yazılımı olarak, tıkanmasız (non-blocking) özelliğe sahip Netty Web sunucusu¹ kullanılmıştır.
- Yük Dengeleyici: Web Servis Sunucu Kümesi'ndeki sunuculara istemcilerden gelen yükü dengelemekle görevli olan yük dengeleyici yazılımıdır. Yük Dengeleyici olarak yazılım tabanlı ve tıkanmasız bir yük dengeleyici olan Nginx² programı kullanılmıştır.

¹ <http://netty.io/>

² <https://www.nginx.com/>

- NoSQL Veritabanı Kümesi: Sütun tabanlı bir NoSQL veritabanı olan Apache Cassandra³ teknolojisi kullanılarak oluşturulan dağıtık veri deposudur.

Alternatif Mimari 1’de kilit rol oynayan mimari bileşen sütun tabanlı NoSQL veritabanı Apache Cassandra’dır. Cassandra çok yüksek hacimli veriyi yönetmek amacıyla geliştirilen, yüzlerce düğümü destekleyen, dağıtık bir veri depolama sistemidir [1]; erişilebilirlik modelini Amazon Dynamo’dan [2], veri modelini Google BigTable’dan [3] almıştır.

Cassandra’nın seçilmesinde pay sahibi olan önemli özellikler, doğrusal ölçeklenebilir ve performanslı yapısı, yüksek erişilebilirlik sağlaması ve işlem bazında ayarlanabilir tutarlılık (consistency) seviyesi şeklinde sıralanabilir.

Cassandra’nın doğrusal ölçeklenebilir yapısı geliştirilen sistemde kademeli ürün takibi yapılmasına olanak sağlamaktadır. Cassandra’nın doğrusal ölçeklenebilme yeteneği, veriyi tutarlı bir anahtarlama (consistent hashing) fonksiyonuyla eşdüzey (peer-to-peer) düğümlere dağıtma özelliğinden gelmektedir. Cassandra $[-2^{63}, +2^{63}]$ aralığında değer üreten bir anahtarlama fonksiyonu kullanmaktadır. Bu değer aralığı, küme oluşturulurken kümedeki toplam düğüm sayısına göre düğümlere paylaşılır. Bu sayede, bir kaydın kümede hangi düğümde saklanacağı, kaydın birincil anahtar değerinin anahtarlama sonucu kolaylıkla bulunur.

Cassandra’nın eşdüzey yapısı sayesinde tek noktadan aksaklık (single point of failure) durumlarıyla karşılaşmaz. Buna ek olarak, kayıtların “Replication Factor” parametresiyle ayarlanan sayıda düğüme kopyalanması erişilebilirliğe olumlu katkı sağlayan başka bir özelliktir.

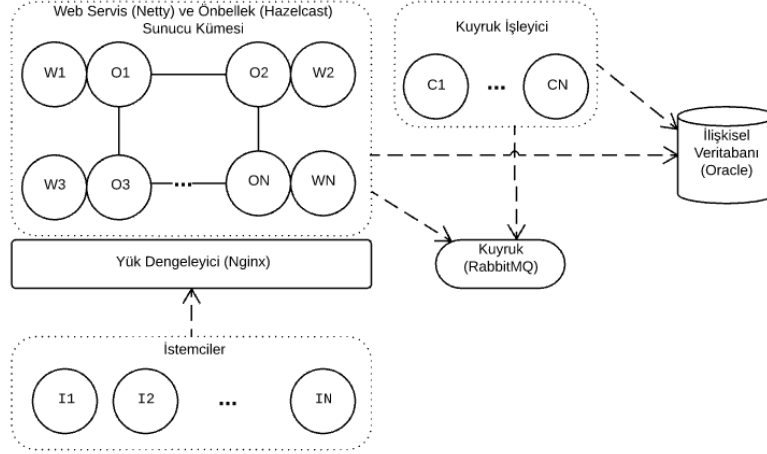
Alternatif Mimari 1’de Cassandra kullanılmasının başka bir önemli nedeni de, Cassandra’nın yazma tutarlılığı (write consistency) parametresini sorgu bazında ayarlamaya izin vermesidir. Bu sayede, tekil bazlı ve lot bazlı izlemede farklı yazma tutarlılıkları kullanmak mümkün olabilmektedir. Lot bazlı izlemede, istemciler arası çekişme durumu (race condition) ortaya çıkabilir. Dolayısıyla, lot bazlı takipte yazma tutarlılığı parametresi daha kısıtlayıcı bir değere ayarlanabilir. Yazma tutarlılığını duruma göre daha az kısıtlayıcı ayarlayabilmek özellikle performans açısından önemli avantajlar sağlayabilmektedir. Lot bazlı – tekil bazlı takip örneğindeki ihtiyaçta görülen ve tutarlılığı daha az gözeterek erişilebilirlik, performans, network bölünmesine karşı dayanıklılık gibi başka bir takım parametrelerde iyileştirme yapabileme yeteneği sunan dağıtık veritabanı özellikleri genellikle CAP teoremi [4] ve nihai tutarlılık (eventual consistency) [5] başlıkları altında daha ayrıntılı incelenmektedir.

2.2 Alternatif Mimari 2

Alternatif Mimari 2’nin yapısı Şekil 2’de gösterilmiştir.

³ <http://cassandra.apache.org/>

Şekil 2. Alternatif Mimari 2



Alternatif Mimari 2, beş ana bileşenden oluşmaktadır:

- Web Servis ve Önbellek Sunucu Kümesi: Alternatif Mimari 1'deki gibi, ürün hareketleri işlevlerini istemcilere servis olarak sunan ve W1, W2, şekilde temsil edilen Web sunucularının yanına, İlişkisel Veritabanı'ndan okuma performansını artırmak amacıyla dağıtık Hazelcast⁴ önbellek düğümleri kurularak elde edilen sunucu kümesidir.
- Yük Dengeleyici: Alternatif Mimari 1'deki ile aynı işleve sahiptir.
- Kuyruk: Bildirimlerin İlişkisel Veritabanı'nda saklanmadan önce biriktirildiği RabbitMQ⁵ dağıtık kuyruk sistemidir. Mimaride bir tane fiziksel sunucu üzerinde birden çok kanal (channel) barındıran bir Kuyruk bileşeni kullanılmıştır.
- Kuyruk İşleyici: Kuyruk'ta biriktirilen değişiklikleri topluca İlişkisel Veritabanı bileşenine yansıtan bileşendir.
- İlişkisel Veritabanı: Bildirimlerin nihai olarak saklandığı ve sorgulandığı Oracle⁶ ilişkisel veritabanı yönetim sistemidir.

Alternatif Mimari 2'de Kuyruk ve İlişkisel Veritabanı bileşenleri mimaride kilit rol oynar. İlişkisel veritabanlarının genel çalışma prensibi, bir hareketle (transaction) ilgili değişikliği veritabanına yansıtmadan önce, bir günlük prosesi aracılığı ile (örn. Oracle LGWR) diskte genellikle günlük kütüğü (log file) olarak adlandırılan bir alana yansıtmak şeklindedir. Günlük prosesi, günlük kütüğünde kayıt oluşturunca hareket onayı (transaction commit) gerçekleşmiş olur. Günlük prosesleri ve günlük kütükleri farklı veritabanı yönetim sistemlerinde farklı isimlerle anılırlar, ancak günlük proseslerinin ortak noktası, veri tutarlılığını sağlamak amacıyla yalnızca bir tane olmalarıdır. Bu durum, ilişkisel veritabanlarında yatayda ölçekleme yapmanın önünde bir engel teşkil

⁴ <https://hazelcast.com/>

⁵ <https://www.rabbitmq.com/>

⁶ <http://www.oracle.com/us/corporate/features/database-12c/index.html>

eder. Alternatif Mimari 2’de bu duruma çözüm olarak RabbitMQ dağıtık kuyruk sistemi kullanılmıştır. Bu çözümde, verilerle ilgili değişiklikler RabbitMQ Dağıtık Kuyruk Sunucuları’nda saniyeler cinsinden ayarlanabilir bir süre boyunca biriktirilir ve bu değişiklikler topluca Kuyruk İşleyici bileşeni tarafından İlişkisel Veritabanı’na yansıtılır. RabbitMQ birçok fiziksel sunucuda aynı anda çalışabilen kalıcı bir kuyruk sistemidir dolayısıyla ölçeklenebilir ve tek noktadan aksaklığa karşı dayanıklıdır.

3 Alternatif Mimarilerin Değerlendirilmesi

Bu bölümde mimari alternatifleri sınamak için yapılan yük testlerinin sonuçları verilmiştir. Testlerdeki bildirimlerin %20’sini tekil bildirimler, %80’ini lot bildirimleri oluşturmaktadır. Testler Amazon Web Services (AWS) bulut ortamında gerçekleştirilmiştir⁷. Alternatif Mimari 1’deki Cassandra kümelerinin Replication Factor parametresi 3, Read ve Write Consistency parametreleri QUORUM olarak ayarlanmıştır. Alternatif Mimari 1 için sonuçlar Tablo 1’de, Alternatif Mimari 2 için sonuçlar Tablo 2’de gösterilmiştir.

Tablo 1. Alternatif Mimari 1 Yük Testi Koşum Ortamı ve Sonuçları

	Koşum 1	Koşum 2	Koşum 3
Saniyedeki bildirim sayısı	5000	10000	30000
Koşum Süresi	10 dakika	10 dakika	1 saat
NoSQL Veritabanı Sunucu Adedi (c3.4xlarge)	3	4	14
Web Servis Sunucu Adedi (c3.4xlarge)	1	1	3
Yük Dengeleyici Sunucu Adedi (c3.4xlarge)	1	1	1
İstemci Sunucu Adedi (c3.4xlarge)	1	1	1
Web Servis Kümesi (Netty) CPU Kullanım Oranı	%5-%15 arası	%20-%40 arası	%40-%45 arası
NoSQL Veritabanı Kümesi (Cassandra) CPU Kullanım Oranı	%10-%20 arası	%10-%35 arası	%10-%80 arası

Tablo 2. Alternatif Mimari 2 Yük Testi Koşum Ortamı ve Sonuçları

	Koşum 1	Koşum 2	Koşum 3
Saniyedeki bildirim sayısı	5000	10000	30000
Koşum Süresi	10 dakika	10 dakika	1 saat
İlişkisel Veritabanı Sunucu Adedi (c3.4xlarge)	1	1	1

⁷ AWS sunucu tipleri için bkz: <https://aws.amazon.com/ec2/instance-types/>

Web Servis (Netty) ve Önbellek (Hazelcast) Sunucu Kümesi için Sunucu Adedi (c3.4xlarge)	2	4	10
Kuyruk Sunucu Adedi (Koşum 1 ve 2: c3.4xlarge, Koşum 3: c3.8xlarge)	1	1	1
Kuyruk İşleyici Sunucu Adedi (c3.4xlarge)	1	1	1
Yük Dengeleyici Sunucu Adedi (c3.4xlarge)	1	1	1
İstemci Sunucu Adedi (c3.4xlarge)	1	1	1
Web Servis Kümesi (Netty) ve Önbellek (Hazelcast) Sunucu Kümesi CPU Kullanım Oranı	% 15-%22 arası	%25-%30 arası	%40-%45 arası

4 Sonuç

Her iki mimari tasarım da saniyede 30000 bildirim kabul edilebilir seviyelerde işleyebilmiştir. Ancak yapısal açıdan ve veri tutarlılığı açısından karşılaştırıldığında, Alternatif Mimari 1, Alternatif Mimari 2'den daha üstündür. Alternatif Mimari 1'de daha az mimari bileşen kullanıldığı için, sistem yönetimi açısından Alternatif Mimari 1 daha avantajlıdır. Bileşen sayısının artmasının başka bir dezavantajı da olası aksaklık noktalarının artmasıdır. Alternatif Mimari 2'de, bildirimler Kuyruk bileşeninden İlişkisel Veritabanı bileşenine yazılana kadar sorgulanamaz, bu durum, sorgulama ve güncelleme işlemleri sırasında veri tutarlılığı açısından sorun oluşturmaktadır. Alternatif Mimari 1'de veri tutarlılığı seviyesi duruma göre ayarlanabilir bir yapıdadır ve bu da özellikle tekil bildirimlerde önemli avantaj sağlamaktadır.

Referanslar

1. Lakshman, A., Prashant M.: Cassandra – A Decentralized Structured Storage System. ACM SIGOPS Operating Systems Review. 44.2, 35-40 (2010)
2. DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., Vogels, W.: Dynamo: Amazon's Highly Available Key-value store. ACM SIGOPS Operating Systems Review 41.6, 205-220 (2007)
3. Chang, F., Dean, J., Ghemawat, S., Hsieh, W.C., Wallach, D.A., Burrows, M., Chandra, T., Fikes, A., Gruber, R.E.: Bigtable: A Distributed Storage System for Structured Data. ACM Transactions on Computer Systems (TOCS), 26.2, 4 (2008)
4. Brewer, E.: CAP Twelve Years Later: How the "Rules" Have Changed. Computer 45.2, 23-29 (2012)
5. Vogels, W.: Eventually Consistent. Communications of the ACM, 52.1, 40-44 (2009)