




Olay tabanlı sistemlerde karmaşık olayların tespiti amaçlı gerçek zamanlı izleme alt yapısı

Detecting complex events with real time monitoring infrastructure on event-based systems

Mehmet Sıddık AKTAŞ^{1*} 

¹Bilgisayar Mühendisliği Bölümü, Elektrik-Elektronik Fakültesi, Yıldız Teknik Üniversitesi, İstanbul, Türkiye.
aktas@yildiz.edu.tr

Geliş Tarihi/Received: 23.04.2018, Kabul Tarihi/Accepted: 16.11.2018
* Yazışılan yazar/Corresponding author

doi: 10.5505/pajes.2018.28044
Araştırma Makalesi/Research Article

Öz

Günümüzdeki birçok yeni uygulamada; veri, sınırlı saklama ortamlarına yönetilemeyecek büyüklüklere varabilmektedir. Bu tür uygulamalarda verinin sürekli veri akışı formatında olduğu görülmektedir. Bu veri üzerindeki sorgulamalar, klasik veri tabanlarından farklı olarak, bildirim sorguları kullanılarak gerçekleştirilmektedir. Bildirim sorguları, sürekli veri akışı üzerinde filtreleme yapma olanağı sağlamaktadır. Bunun yanı sıra sorgulardaki koşullara uygun verilerin, izlenebilmesi yeteneğini kazandırmaktadır. Sürekli veri akışı formatında veri yapısı içeren sistemler, birim verinin gerçekleşen bir olay olarak değerlendirilebileceği, Olay-tabanlı Sistemler olarak değerlendirilebilir. Bu tür sistemlerde; iş süreçlerinin tetiklenmesini sağlayacak karmaşık olayların gerçek zamanlı tespit edilmesi, bir ihtiyaç haline gelmektedir. Bu araştırma kapsamında, bu eksikliğin giderilmesine yönelik Karmaşık Olay İzleme Altyapısı önerilmektedir. Önerilen altyapı, Olay-tabanlı Sistemlerdeki iş süreçlerinin başlatılma ön koşullarını oluşturan, karmaşık olayların tespit edilmesini sağlamaktadır. Önerilen altyapının kullanılabilirliğini göstermek adına, tıklama verilerinin sürekli veri akışı formatında üretildiği e-ticaret web siteleri için bir prototip uygulama geliştirilmiştir. Prototip uygulama, ticaretin gerçekleştiği sahadan (e-ticaret ortamı) toplanan müşteri hareket verilerini sürekli olarak izlemektedir. Zaman damgası ile etiketlenmiş her bir tıklama verisi gerçekleşen bir olay verisidir. Belirli bir zaman aralığında gerçekleşen temel olayların bir araya gelmesiyle oluşan karmaşık olaylar tespit edilerek, gerçekleşen ticari aktiviteler için uygun iş süreçlerinin çalıştırılması sağlanmaktadır. Geliştirilen uygulama, performans ve ölçeklenebilirlik açısından değerlendirilmiş, önerilen altyapının kullanılabilirliğini gösteren sonuçlar elde edilmiştir.

Anahtar kelimeler: Karmaşık olay işleme, Olay tabanlı sistemler, Sürekli veri akışı, Gerçek zamanlı izleme sistemleri

Abstract

Nowadays, in many recent applications; the data size become too large to be managed within the limited storage environments. In such applications, the data is a streaming data and has a continuously flowing data format. Queries on streaming data are performed using declarative queries, as opposed to queries made on classical databases. Declarative queries provide the ability to filter on a continuous data stream. In addition, they provide the ability to follow the data that match the conditions in the queries. Systems with a data structure in a continuous data flow format can be considered Event-based Systems, in which unit data can be considered as a real event. In such systems; real-time detection of complex events that will trigger business processes is becoming an important need. Within the scope of this research, a Complex Event Monitoring Infrastructure is proposed to address this deficiency. The proposed infrastructure provides the detection of complex events, which form the preconditions for initiating business processes in Event-based Systems. In order to demonstrate the usability of the proposed infrastructure, a prototype application has been developed for e-commerce web sites, where the click-stream data is generated in a streaming data format. Prototype application continuously monitors customer click-stream data collected from the Web site, where the trade takes place (e-commerce environment). Each click-stream data, labeled with a timestamp, is considered as an event data. By combining the basic events, occurring within a certain time period, complex events are identified and the appropriate business processes for corresponding commercial activities are triggered. The developed application was evaluated from the perspective of performance and scalability, and positive results were obtained indicating the usability of the proposed infrastructure.

Keywords: Complex event processing, Event-Based systems, Continuous data streaming, Real-Time monitoring systems

1 Giriş

Sürekli veri akışı formatındaki veriyle uğraşan organizasyonlarda, veriden elde edilecek gerçek zamanlı bilgiye dayalı olarak, anlık olayların tespiti ve bu olayların tetikleyebileceği iş süreçlerinin/kurallarının çalıştırılması önemli bir gereksinim haline gelmiştir. Organizasyonlar bu sayede, sisteme yönelik potansiyel tehlikelerden oluşan riskleri azaltabilecek ve kullanıcılarına sundukları hizmet kalitelerini, anlık geri dönüşler yaparak, arttırabileceklerdir. Günümüzdeki organizasyonlarda, iş süreçlerinin oldukça karmaşık yapıda olması, iş süreçlerinin başlatılması için gerekli ön koşulların karmaşıklığını da arttırmaktadır. Sürekli veri akışı yapısındaki verilerden yola çıkılarak, gerçekleşen karmaşık olayların tespiti

ve bu olaylarla ilgili olabilecek iş süreçlerinin bulunması ve çalıştırılması, organizasyonların daha hızlı aksiyonlar üretebilmesinin önünü açacaktır.

Bu araştırmanın amacı; satış ve pazarlama sektörünün doğru müşteriye, doğru zamanda, doğru ürünle ilgili anlık kampanyalar, indirimler önerilebilmesi için bir yöntem geliştirmektir. Geliştirilen yöntem ile dinamik kampanya kurallarının tanımlanmasına izin verilerek satış öncesi, anı ve sonrası müşterilere anında pazarlama faaliyetlerinde (kampanya, indirim, bonus vb.) bulunma imkânı tanınabilecektir. Anlık pazarlama faaliyetleri farklı iletişim kanallarından (SMS, e-posta, web site pop-up mesajlar, telefon) hızlı bir şekilde gerçekleştirilmelidir. Satış ve pazarlama

sektöründe en önemli ihtiyaç, “doğru müşteriye doğru zamanda, doğru ürünü, doğru kampanya” ile sunabilmektir. Bu durumda satış ve pazarlama alanında; kolay uygulanabilir, müşterilere gerçek zamanlı pazarlama faaliyetlerinde bulunabilen, tarihsel veri üzerinde yüksek performanslı veri madenciliği tabanlı analizler yapabilen bir uygulamaya duyulan ihtiyaç, araştırmamızın temel başlatılma nedenini oluşturmaktadır.

Günümüzde veri büyüklüğünün hızla artması yeni veri keşfetme yöntemlerinin ortaya çıkmasına sebep olmaktadır. Büyük veri kavramı, hızla artan veri üretimi sonrası ortaya çıkmış bir terimdir. Büyük veri, özgün veri madenciliği araçlarıyla yönetilememektedir. Büyük veriyi yönetmek için iki ana yöntem bulunmaktadır. Bunlar örnekleme ve dağıtık sistemlerin kullanılmasıdır. Örnekleme yöntemi, yönetilemeyecek kadar büyük verinin örnek bir kısmı üzerinde analiz işlemlerinin gerçekleştirilmesidir. Tüm veri kullanılmadığı için yaklaşık bir sonuç vermektedir. Günümüzde en popüler yöntem, Eşle İndirge yapısı tabanlı, dağıtık sistemler yöntemidir. Eşle İndirge; büyük veri üzerinde, paralel veri işleme olanağı sağlayan bir programlama modelidir. Günümüzde yaygın kullanılan, Hadoop ve Spark projeleri; büyük veri üzerinde, Eşle İndirge programlama modelini gerçekleyen açık kaynaklı projeler olarak karşımıza çıkmaktadır [1]. Her iki platform da, HDFS adı verilen dağıtık veri tabanı üzerinde paralel veri işleme olanağı sunmaktadır. Bu proje kapsamında, sürekli gelen veriyi işlemede kullanılan Eşle İndirge fonksiyonları için ise Spark platformu kullanılmıştır.

Verinin gelme hızı ve verilere uygulanan işlemlerin çevikliğinin iş hayatında önemli rol üstlenmekte olduğu David Meerman Scott’un “Real-Time Marketing & PR” kitabında açıklanmaktadır [2]. Yeni iş fırsatlarının tespit edilebilmesi için gelen verilerin gerçek zamanlı olarak işlenmesi büyük önem taşımaktadır. Gerçek zamanlı gelen verinin işlenip önemli sonuçların elde edilmesi için Karmaşık Olay İşleme (Complex Event Processing) yönteminden faydalanılmaktadır. Karmaşık Olay İşleme, oluşan olayların akışını analiz eden ve bunlardan sonuç çıkaran bir yöntemdir. Projemizde, bir elektronik ticaret sitesinde oluşan olayların analiz edilip, müşterilere kampanya oluşturacak uyarıların (karmaşık olayların) yaratılması için, Karmaşık Olay İşleme tekniği kullanılmaktadır. Oluşturulan karmaşık olaylara karşılık gelen kurallarının uygulanması için ise Drools Kural Motoru kullanılmaktadır.

Yukarıda bahsedilen motivasyonu gerçekleştirmek amacıyla araştırma sorularımızı şu şekilde belirliyoruz: 1) Satış öncesi, anı ve sonrasında, müşterilerin oluşturdukları anlık gelişen olayların, gerçek zamanlı olarak işlenerek, müşterilere anında pazarlama faaliyetleriyle (kampanya, indirim, bonus vb.) dönüş yapabilme yeteneği sağlayacak yazılımın mimarisi nasıl olmalıdır ve nasıl geliştirilebilir? 2) Müşterilerden farklı kaynaklar aracılığıyla toplanacak büyük ölçekli olay verilerinin saklanabilmesi, sorgulanabilir olması, eşle-indirge programlama modeli kullanılarak kümelenebilmesi, yeteneklerini sağlayacak bir yazılım mimarisi nasıl olmalıdır ve nasıl geliştirilebilir?

Bu makalenin katkılarının ise şu şekilde özetliyoruz. E-ticaret alanına yönelik olarak, satış anı ticari aktivitelerin toplanması, anlık olarak işlenmesi ve işlenen veriden varılacak sonuçlara göre anlık faydalar (kampanya, indirim, bonus vb.) üretilmesini sağlayacak olan bir sistem geliştirilmiştir. Bu kapsamda,

e-ticaret siteleri gibi olay tabanlı sistemlerde karmaşık olayların tespiti amaçlı gerçek zamanlı izleme alt yapısı için bir soyut yazılım mimarisi önerilmiştir. Önerilen mimarinin kullanılabilirliğini göstermek amacıyla, bir prototip uygulama geliştirilmiş ve performans testleri gerçekleştirilmiştir. Geliştirilen sistem, e-ticaret sitelerindeki, tüketicilerin anlık hareketlerini izleyerek, tüketiciye yönelik anlık eylemler üretebilmesini sağlayabilecek niteliktedir. Elde edilen performans sonuçları, önerdiğimiz mimarinin başarılı olduğunu göstermektedir.

Bu makalenin organizasyon yapısı şu şekildedir: Giriş bölümünde, projeyi geliştirmekteki motivasyonumuzdan bahsedilmiştir. Literatür incelenmesi kısmında projeyi gerçekleştiren araştırma yaptığımız konularda geçmişte yapılmış çalışmalar anlatılmıştır. Önerilen yazılım mimarisi kısmında, sürekli veri akışı olan sistemlerde, karmaşık olay işleme amaçlı gerçek zamanlı izleme alt yapı yazılımı detaylarıyla anlatılmıştır. Geliştirme kısmında, geliştirilen prototip uygulamada kullanılan teknolojik ürünler ve uygulanan yöntemlerden bahsedilmiştir. Değerlendirme kısmında projenin farklı parametreler kullanılarak çalıştırılması elde edilen sonuçlar karşılaştırılmış, prototip uygulamanın verimli çalışabileceği iş yükü belirlenmiştir. Sonuç kısmında ise projeyi gerçekleştirdikten sonra elde ettiğimiz faydalar anlatılmıştır.

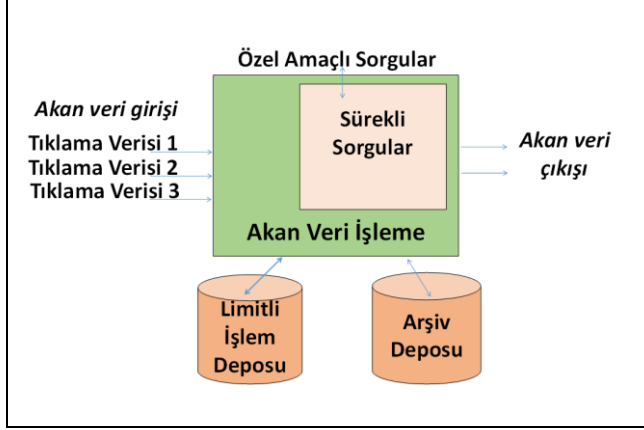
2 Literatür İncelenmesi

Günümüzde, geliştirilen en yeni dağıtık sistem uygulamalarında, sürekli akan veri modeli daha yaygın olarak kullanılmaktadır [3]. Verinin sürekli üretilmesi ve boyutunun artması durumunda gerçek zamanlı akan veri modeline gereksinim duyulmaktadır. Gerçek zamanlı akan veri modeli; sensor ağları ve ağ trafiği denetlemeleri, tıklama verilerinin gerçek zamanlı analiz edilmesi gibi farklı uygulamalarda kullanılmaktadır. Akan veriyi işlemek için çeşitli yaklaşımlar bulunmaktadır. Bu yaklaşımlardan biri akış halinde gelen veriyi belirli büyüklükte olan bir pencere üzerinde analiz etmektir. Burada, son gelen n adet veri pencerede tutularak gerekli işlemler gerçekleştirilmektedir [4]. Bu yöntemde, sürekli akan veriler için pencere üzerinde yer alan veriler güncellenmekte ve gerekli fonksiyonlar uygulanmaktadır.

Akan veri yönetim sisteminin üst düzey organizasyonu Şekil 1’de gösterilmektedir. Herhangi bir sayıdaki akan veriler sisteme gelebilir ve gelen akan verilerin tipleri ve gelme süreleri aynı olmayabilir. Akan veri yönetim sistemini, geleneksel veri yönetim sisteminden ayıran en önemli özellik, akan verilerin gelme hızının kontrol altında tutulabilmesidir. Şekil 1’de resmedildiği gibi, akan veriler büyük ölçekli bir arşiv deposunda tutulabilmektedir. Ancak, böylesi bir arşiv deposu üzerinde istenilen sorguların, sınırlı bir zaman dilimi içinde yapılamayacağı varsayılmaktadır. Limitli işlem deposunda ise akan verilerin hepsi veya bir kısmının sorgu sonuçları tutulabilmektedir. Limitli işlem deposu, akan verilere yapılan sorgu hızına göre disk veya ana bellek olabilmektedir [4]. Apache Spark ve Apache Storm platformları, akış halinde gelen veriler üzerinde uygulama geliştirilebilmesini sağlayan, dağıtık veri işleme platformlarıdır. Her iki platform da ücretsiz, açık kaynaklı, gerçek zamanlı hesaplama platformlarıdır [5],[6].

Geçmişe dönük temas verileri zaman içinde büyük veri yığınları oluşturmaktadır. Büyük veri niteliği gösteren böylesi bir verinin, belirli bir zaman dilimi içinde, analiz edilmesi için, toplu-güdümlü veri işleme tekniklerine ihtiyaç duyulmaktadır [7]. Apache Hadoop, toplu-güdümlü veri işleme modelini

başarıyla gerçekleyen bir kütüphane sunmaktadır. Apache Hadoop, kümelerle dağılmış büyük verinin dağıtık olarak işlenmesini sağlayan yazılım kütüphanesidir. Kendi yerel işlemcisi ve belleğini içeren tek bir sunucudan binlerce makineye kadar ölçeklenebilmektedir [8].



Şekil 1: Akan veri yönetimi.

Eşle-İndirge modeli, sürekli büyüyen veri karşısında çözüm üretmek için tasarlanmıştır ve farklı hesaplama problemlerine başarılı bir şekilde uygulanmaktadır [9]-[11]. Eşle-İndirge programlama modeli Hadoop Kütüphanesi ile gerçekleştirilmektedir. Dağıtık ortamda, olay verileri üzerinde analizler yaparak, kaynakların takibini gerçekleştiren çalışmalar mevcuttur [12]-[17],[25]. Gerçekleşen olay verileri üzerinde analizler yaparak, gerçekleşen olaylar hakkında bilgi veren sistemler üzerinde farklı çalışmalar yapılmıştır [18]-[20],[21]. Buradaki çalışmada ise tıklama verileri üzerinden bilgi çıkarımı yapılarak, anlık aksiyonlar alınması konusunda kararlar verilmektedir.

Bir e-ticaret sitesi üzerinden otomatik olarak toplanacak tıklama verileri, ziyaretçilerin gezinme davranışlarını temsil eden başlıca veri kaynakları olmaktadır [22]. Bu veriler, kullanıcının davranışını anlamak için toplanacak ve farklı soyutlama seviyelerinde dönüştürülecektir. Burada kullanılmayan planladığımız bazı metrikleri maddeler halinde sıralıyoruz.

Web sitesi trafiği metrikleri: Web kullanımındaki en yaygın gösterge web sitesi trafiğidir. Web sitesi trafiği; web sitesine gelen ziyaretçiler tarafından gönderilen ve alınan verilerin miktarını yansıtmaktadır. Bu gösterge, bir web sitesinin genel popülaritesini değerlendirmek için sıklıkla kullanılır, ancak web sitesinin belirli bölümlerine veya sayfalarına da uygulanabilir. Burada, web sitesi trafiğini belirlerken şu temel metrikler kullanılmaktadır. (a): Ziyaret sayısı, (b): Tekil ziyaretçi sayısı, (c): Ziyaret edilen toplam sayfa sayısı.

Ziyaret sayısı ve tekil ziyaretçi sayısı: Sitenin belirli bir süre zarfında ziyaret edildiği, ziyaret sayısı olarak bilinen toplamdaki oturum sayısı, genellikle web trafiğini gösteren bir metrik olarak kullanılacaktır. Bu metriğin bir türevi ise, benzersiz ziyaretçiler olarak bilinmektedir. Bu metrik, farklı kullanıcılar veya makinelerden kaynaklanan oturumların sayısını ifade etmektedir. Bu metriğin hesaplaması, her ziyaretçi için benzersiz olan çerezlerin kullanımı ile yapılmaktadır. Bu istatistik genellikle farklı zaman dilimlerinde (örneğin haftalık olarak) listelenmektedir. Burada, metriğin değerlerinde, doğal bir değişkenliğin dışında, önemli bir değişiklik olup olmadığını gözlemlenmektedir.

Ziyaret edilen toplam sayfa sayısı: Ziyaret edilen toplam sayfa sayısı, genellikle toplam kullanıcı isteği sayısına eşdeğerdir. Bu metrik, web sitesi trafiğini belirleyebilmek için başka bir yöntemdir. Bu metriğin değeri, tek bir sayfanın "byte" olarak ortalama boyutu ile birleştirildiğinde, bir web servisinin yükünü tahmin etmek için kullanılabilir. Pazarlamacılar, yeni bir reklam yerleşiminin başarısını ölçmek için, belirli bir sayfaya olan toplam istekleri araştırmak için bu metriği kullanabilmektedir.

Web sitesi etkileşim seviyesi metrikleri: Bir web sitesine olan trafiği izlemek, kullanıcıları web sitesinde biraz daha zaman geçirmeye teşvik etmek ve kullanıcıların web sitesini daha yoğun bir şekilde kullanmaları için ilgilerini çekmek amaçlı yapılmaktadır. Bu kavram genellikle web analizi bağlamında yapışkanlık olarak veya etkileşim seviyesi olarak bilinmektedir. Etkileşim seviyesinin yüksek değerleri her zaman bir web sitesinin avantajına olan bir durum değildir. Bir e-ticaret sayfası üzerinde, kullanıcıların, ürün veya hizmet bilgilerini ve açıklamaları içeren içerik sayfalarında gezinirken, daha fazla zaman harcaması tercih edilen bir durumdur. Ancak, etkileşim seviyesi metrikleri; kayıt sayfası veya gönderim bilgileri sayfası gibi sayfalar için, web sitesi üzerindeki sürecin verimli bir şekilde yönlendirilip yönlendirilmediğini göstermektedir. Etkileşim süresini değerlendirmenin birçok yolu vardır. Burada, etkileşim seviyesini belirlerken şu metrikler kullanılmaktadır: Hemen Çıkma Oranı, Ziyaret Edilen Sayfa Sayısı, Ortalama Oturum Sayısı.

Hemen Çıkma Oranı: Bir web sitesi için etkileşim seviyesi değerlendirmesi kategorisine giren önemli bir metrik, hemen çıkma oranı olarak adlandırılmaktadır. Hemen çıkma oranı, web sitesine gelen, ancak birkaç saniye sonra siteyi incelemeyen çıkan veya yalnızca tek bir sayfayı ziyaret eden ziyaretçilerin yüzdesini ifade etmektedir. Diğer metriklere benzer şekilde; hemen çıkma oranı, web sitesi içindeki belirli bölümler veya tek web sayfaları için hesaplanabilmektedir.

Ziyaret Edilen Sayfa Sayısı: Bu metrik; bir web sitesi oturumu için, ziyaret derinliği ve etkileşim seviyesini göstermek için kullanılan yaygın bir metriktir. Bu metrik, tek bir kullanıcı için incelenebileceği gibi, tüm veri kümesi içinde incelenebilir. Örneğin, bir web sitesi için, oturum başına ziyaret edilen ortalama sayfa sayısı, tüm veri kümesi için hesaplanarak, genelleştirilebilmektedir. Oturum başına ziyaret edilen sayfa sayısının çokluğu, ziyaretçilerin sayfadaki etkileşim seviyesinin yüksek olduğunu göstermektedir.

Ortalama Oturum Süresi: Etkileşim süresi, ortalama oturum süresi metriği kullanılarak (belirli bir zaman penceresinde) belirlenebilir. Bu metrik; tüm web sitesi, bazı bölümleri ve hatta belirli web sayfaları için hesaplanabilmektedir. Bir web sitesinde geçirilen toplam sürenin, toplam oturum sayısına oranı alınarak hesaplanmaktadır. Oturum Başına Ziyaret Edilen Sayfa Sayısı ve Ortalama Oturum Süresi metrikleri beraber kullanıldığında, kullanıcıların ne kadar hızlı bilgiyi özümseyebildiğini belirleyebilmek mümkün olmaktadır.

Dönüşüm metrikleri: Bir e-ticaret platform için, dönüşüm analizi, doğrudan karla ve gelirle ilgili olduğu için en alakalı iş metriği olma eğilimindedir [23]. E-ticarette başarılı bir dönüşüm, bir ziyaretin web sitesi tarafından ürün yada ürünler satın almak için, yönlendirildiğinde gerçekleşir. Örneğin, bir kullanıcı sanal sepetine bir öğe eklediğinde bir dönüşüm gerçekleşir. Dönüşüm oranlarını ölçmek için, en çok kullanılan dönüşüm metriği, dönüşüm oranı metriğidir.

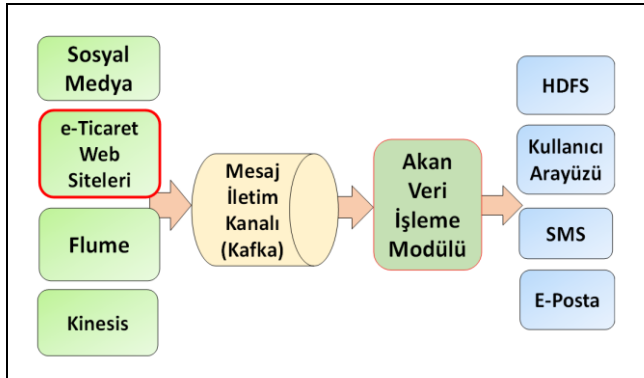
Dönüşüm oranı metriği: Bu metrik, e-ticaret platformları için, çevrimiçi satın alma işlemi gerçekleştiren web sitesi ziyaretlerinin yüzdesi olarak tanımlanmaktadır. Dönüşüm tanımı, farklı web siteleri veya işletmeler için farklı olabilmektedir. Dönüşüm oranlarını iyileştirmek için, e-ticaret alanında, alıcıların altta yatan nedenlerini incelemek için, çeşitli çalışmaların yapıldığını görmekteyiz [24]. Burada, işletme yöneticileri, satın alma işleminin ardındaki nedenleri anlayabilirse, daha fazla ziyaretçiyi alıcıya dönüştürmek için etkili stratejiler kullanabilmektedirler.

Bu çalışma kapsamında, tıklama verilerinin dağıtık bir veri işleme platformu üzerinde yönetilmesi üzerinde çalışılmıştır. Tıklama verilerinin anlaşılması ve metrikler kullanılarak özniteliklere dönüştürülmesi kapsam dışı bırakılmıştır.

3 Önerilen yazılım mimarisi

Önerdiğimiz sistem, akan verinin, ölçeklenebilir, yüksek verimli, hata toleranslı bir şekilde gerçek zamanlı olarak işlenmesini sağlamaktadır.

Genel Mimari: Şekil 2’de gösterildiği üzere Akan Veri İşleme Modülü, yayımla/üye-ol mesaj iletim kanalı üzerinden gelen verilerle gerçek zamanlı olarak beslenmektedir. Bu veriler çok farklı ortamlarda oluşabilir ve gerçek zamanlı işleme amaçlı olarak, akan veri işleme modülüne gönderilmektedir. Belirtilen ortamlara örnek olarak; büyük ölçekteki veriyi toplama, oluşturma, ve aktarma sistemleri (örnek Apache Flume), e-ticaret web siteleri, sosyal medya siteleri (örnek Twitter, Facebook) verilebilir. Şekil 2’de gösterildiği üzere, bu araştırma kapsamında, e-ticaret web siteleri üzerinde oluşan olaylar, çalışma-zamanlı olarak toplanmakta ve konu bazlı bir iletişim kanalı üzerinden, Akan Veri İşleme Modülü’ne aktarılmaktadır. Akan Veri İşleme Modülüne entegre olarak çalışan Karmaşık Olay İşleme Modülünde, gelen veriler önceden belirli olan kurallara çerçevesinde işlenmektedir. Önceden belirlenen kurallara uyan koşullar oluştuğunda, farklı aksiyonları tetikleyecek karmaşık olaylar üretilmektedir.

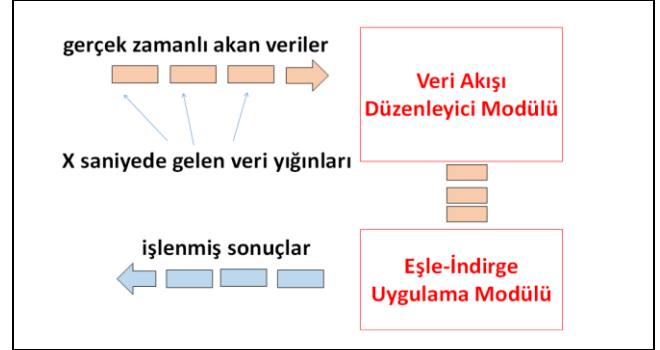


Şekil 2: Önerilen sistemin soyut yazılım mimarisi yapısı.

Akan Veri İşleme Modülü: Bu modülün ürettiği işlenmiş veriler, bir Kullanıcı Arayüzü üzerinden son kullanıcılar tarafından izlenebileceği gibi, farklı iletişim kanalları (SMS vb.) üzerinden de kullanıcılara uyarı şeklinde iletilebilir. Veri Akışı İşleme Modülü, veri akışı üzerinde karmaşık olay işleme tekniği gibi farklı yöntemlerin dağıtık platform üzerinde uygulanmasına olanak sağlamaktadır.

Veri Akışı Düzenleyici Modülü: Şekil 3’te resmedildiği üzere, bir Veri Akışı Düzenleyici Modülüne (örnek Spark Streaming) gelen gerçek zamanlı akan veriler, eşle-indirge programlama

modeli kullanılarak işlenebilecek şekilde bir dönüşüme uğramaktadır. Bu işlem bir Düzenleyici Modül üzerinde gerçekleştirilmektedir. Bu modül farklı frekansta gelen verilerin, Eşle-İndirge işlemine tabi tutulacak formata dönüştürülmesini gerçekleştirmektedir.



Şekil 3: Akan veri işleme modülünde verinin işlenmesi.

Eşle-İndirge Uygulama Modülü: Sisteme gelen verilerin üzerinde Eşle-İndirge işlerinin uygulanarak, sonuç verilerin oluşturulmasını sağlayan bir modüldür. Bu modül, veri akışı üzerinde dağıtık ortamda, gerçek zamanlı olarak Eşle-İndirge işlerinin gerçekleştirilmesi ile sorumludur. Şekil 3’te belirtilen “işlenmiş sonuçlar” verileri, eşle-indirge programlama modeli ile gerçekleştirilen fonksiyonların ürettiği sonuçları ifade etmektedir. Bu şekilde, gerçek zamanlı akan veriler; sisteme gelen ham verileri ifade etmektedir. Örneğin; sisteme sürekli olarak gelen, 0 ve 1’lerden oluşan bir sayı şeridi düşünelim. Bu sayı şeridi üzerinde, N uzunluğunda, sayı şeridi üzerinde hareket eden bir pencere düşünelim. Bu pencere içine düşen 1’lerin sayısını sayan, sayma fonksiyonunu, eşle-indirge programlama modeliyle gerçekleştirdiğimizi varsayalım. Böylesi bir senaryoda, 0 ve 1 değerleri, gerçek zamanlı veriyi ifade etmektedir. Bu örnekte; Eşle-İndirge programlama modeli ile gerçekleştirilmiş sayma fonksiyonunun ürettiği sayısal değerler, işlenmiş sonuçları ifade etmektedir.

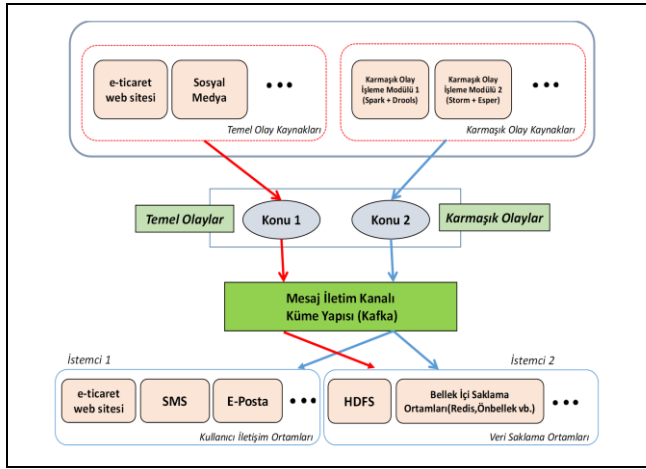
Karmaşık Olay İşleme Modülü: Bu modül kapsamında, toplanan olay verileri üzerinde, karmaşık olay işleme analizleri yapılarak, önceden belirlenmiş beyan edilen (declarative) sorgular için, sorgulara yanıt verilebilecek durumların oluşup oluşmadığı takip edilecektir. Kullanıcının önceden belirlenmiş bir aktivite yada aktivite serisini, belirli bir zaman aralığında tamamladığının tespiti durumunda, anlık aksiyonlar üretilmektedir.

HDFS Veri Saklama Ortamı: Büyük veri mertebesindeki, temas hareketleri sonucu oluşan tıklama verileri üzerinde; raporlama amaçlı, farklı denetimsiz öğrenme algoritmaları, paralel veri işleme teknikleri (eşle-indirge programlama modeli) kullanılarak gerçekleştirilmektedir. Tüm temas verileri, Şekil 4’te belirtilen HDFS platformu üzerinde tutulmaktadır. Segmentasyon analizleri ile, çevrimiçi e-ticaret platformu üzerinde gerçekleştirilmiş tüm kullanıcı aktivitelerinin zaman eksenini üzerinde analiz edilmesi, yarım kalmış aktivitelerin tespiti gibi anlamlı çıkarımlar yapılabilmektedir.

Bellek içi Veri Saklama Ortamı: Eşle-İndirge programlama modeline kullanılarak oluşturulan raporlar, birer JSON objesi olarak bellek-içi saklama ortamında tutulmaktadır. Burada oluşturulan her rapor, son kullanıcının talep ettiği anda, kısa sürede ulaşılabilmektedir.

Konu bazlı yayımla/üye-ol mesaj iletim kanalı (publish-subscribe based message bus): Önerdiğimiz

sistemin sahip olduğu ana yeteneklerden biri, kendisine iletilen anlık verilerinin sorunsuz ve sonradan hızlı bir şekilde okunacak ve sorgulanabilecek bir şekilde kayıt edilmesini sağlamaktır. Verilerin, veri kaynağından, akan veri işleme modülüne aktarımı, bir konu tabanlı yayımla/üye-ol iletişim metodu üzerinden yapılmaktadır. Bu yöntemle, verilerin, herhangi bir veri kaybı olmadan, gerçek zamanlı olarak toplanması, kaydedilmesi ve işlenmesi mümkün olabilmektedir. Bunu sağlamak amacıyla Şekil 4'te resmedildiği üzere iki temel konu üzerinden verinin yayınlanması ve ilgili konulara üye olunarak verinin toplanması sağlanmaktadır. Önerilen sistem mimarisinde böyle bir iletişim kanalının olmasının en büyük faydası, yüksek hızda gelebilecek olan verilerin herhangi bir veri kaybına uğramadan sisteme alınabilmesinin sağlanmasıdır.



Şekil 4: Önerilen veri yayınlama ve toplama modeli.

Veri Modellemesi: Burada elde edilen verileri bir olay (event) olarak nitelendiriyoruz. Bir e-ticaret sitesinden elde edilen olay verisi içinde e-ticaret sitesinde gerçekleşen farklı aktiviteler (ziyaret aktiviteleri, satın alma aktiviteleri gibi) ile ilgili veriler bulunabilmektedir. Bu çalışma kapsamında bir olay' ı şu şekilde tanımlamaktayız.

Olay (Event): Bir kullanıcının, e-ticaret sitesi web sayfalarında yer alan nesnelere tıklaması sonucu oluşan etkileşim bir olay olarak tanımlanmaktadır. Olay verisi içinde, olayın gerçekleştiği zaman, olayı tetikleyen kullanıcı bilgisi, olayda tetiklenen web sayfası üzerindeki nesne bilgileri gibi veriler yer almaktadır.

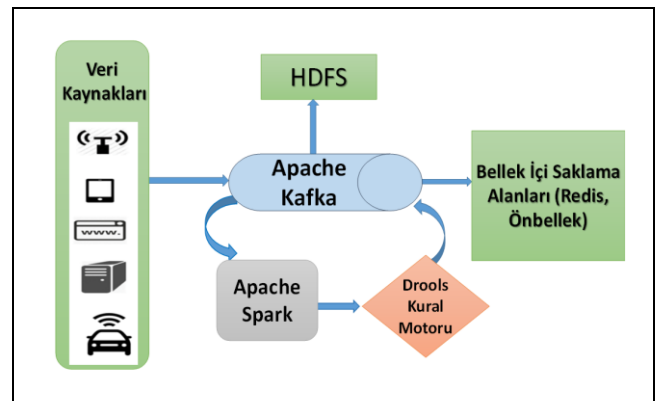
Toplanacak olay verileri üzerinde, karmaşık olay inceleme analizleri yapılarak ve önceden belirlenmiş beyan edilen sorgular için, sorgulara yanıt verebilecek, karmaşık durumların oluşup oluşmadığı takip edilmektedir. Bu kapsamda toplanacak olay verilerinin, hızlı okuma (run-time analiz edilmesi), yazma (dağıtık veri tabanına kaydedilmesi) ve sorgulama (runtime beyan-edilen sorgularda filtrelenmesi) açılarından, en optimum şekilde yönetilebilmesi için, olay verilerinin modellenmesi sağlanmıştır. Her bir olay verisi içinde bulunması gereken minimum veri-öğeleri tespit edilmiştir.

Veri modellemesi üzerinde çalışılırken, takip edilen aktiviteler aşağıdaki gibi belirlenmiştir: **Ziyaret aktiviteleri:** Butip ziyaret aktivitelerine örnek olarak; Ana sayfaya temas durumunda oluşturulan aktiviteler (VISIT_HOMEPAGE), herhangi bir dijital kaynağa temasla oluşturulan aktiviteler (VISIT_RESOURCE), herhangi bir fırsat, kampanya önerisine temas durumunda oluşturulan aktiviteler

(VISIT_SATELSEVENT) şeklindeki aktiviteler verilebilir. **Listeleme ve arama aktiviteleri:** E-ticaret uygulamasında birden fazla ürünün listelenmesi ya da bir sayfada sıralama şeklini değiştirilmesi yada bir ürünün aranması (LISTING_SORT, LISTING_FILTER, SEARCH) aktiviteleridir. **Kullanıcı ilişkili aktiviteler:** Kim olduğu bilinen/bilinmeyen kullanıcıların ilk temas anında oluşturulan aktivitelerdir. **Sipariş aktiviteleri:** Anonim ya da kimliği bilinen müşterinin sepetine ürün ekleme aktiviteleridir. **Sipariş süreci aktiviteleri:** Tamamlanmış siparişin satış sonrası süreçte statü ilerleme aktiviteleridir (ORDER_STATUS, ORDER_CREATED, ORDER_INVOICED etc.). Bir temel aktivite olayının içinde bir ürün ile ilgili olarak, kullanıcı ID' si, ürün ismi, ürün kodu, ürünün adedi, ürünün fiyatı, ürünün toplamdaki fiyatı gibi veri-öğeleri bulunmaktadır. Yine bir temel aktivite olayının içinde kullanıcı ile ilgili olarak, kullanıcının ismi, soy ismi, cinsiyeti, doğum günü, cep telefon numarası, e-mail adresi gibi veri öğeleri bulunabilmektedir. Veri modellemesi çalışmamız; bir karmaşık olayın, temel olaylardan oluşan bir pattern olduğunu, belirli bir zaman aralığında gerçekleştiğini, önceden belirlenmiş zamansal ve birleştirme operatörleri içerebileceğini (örneğin: X=>Y=>Z 30 sn.): 30 sn'lik zaman dilimi içinde, önce X, sonra Y ve ardından Z dijital öğeleri ile temas gerçekleşmiştir.) ortaya koymaktadır. Hem temel olay verilerinin modellenmesi hem de karmaşık olay verilerinin modellenmesi çalışmaları yapılırken, proje çıktısının yüksek performansta anlık yanıtla verebilme özelliğine sahip olması gerektiği düşünülerek, çalışmalar yürütülmüştür.

4 Gerçekleştirme

Önerdiğimiz yazılım mimarisinin gerçekleştirilmesi Java teknolojileri kullanılarak yapılmıştır. Java versiyonu olarak 1.8.66 versiyonu seçilmiştir. Yayımla/Üye-ol mesaj iletim kanalı için Apache Kafka (<http://kafka.apache.org/>) projesi; Kural Motoru olarak Apache Drools Rule Engine (<http://www.drools.org/>) projesi ve Akan Veri İşleme sistemi olarak Apache Spark (<http://spark.apache.org/>) projeleri kullanılmıştır. Yararlanılan bu alt yapı projeleri, önemli bir geliştirici topluluk desteği olan, açık kaynaklı projeler olmaları nedeniyle seçilmiştir. Geliştirdiğimiz prototip yazılımın genel mimarisi Şekil 5'te verilmektedir.



Şekil 5: Gerçekleştirilen sistemin genel yapısı ve kullanılan teknolojiler.

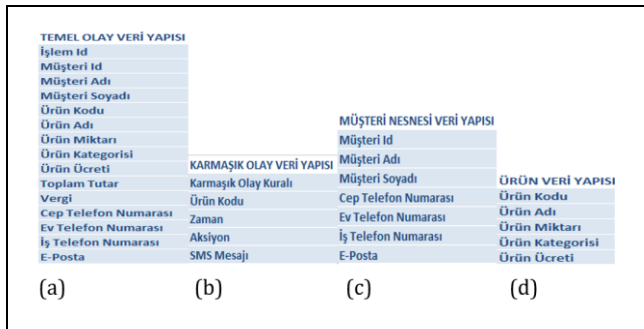
E-ticaret Anlık Pazarlama Uygulaması Veri Modeli: Araştırmada kullanılan veri modeli bir e-ticaret sitesinde anlık pazarlama, kampanya, bonus, indirim gibi faaliyetleri gerçekleştirebilmek için tasarlanmıştır. Veri modelinde, Temel Olay Veri Yapısı (Şekil 6a), Karmaşık Olay Veri Yapısı (Şekil 6b),

Müşteri Nesnesi Veri Yapısı (Şekil 6c), ve Ürün Veri Yapısı (Şekil 6d)'te resmedilmektedir.

Önerdiğimiz sistemde her bir müşteri hareketi bir olay (event) olarak değerlendirilmektedir. Temel olay veri yapısında, müşterinin e-ticaret sitesine temas ettiği süreçte, aldığı ürünler ile ilgili bilgiler ve yaptığı harcamalar ile ilgili bilgiler bulunmaktadır. Olay yapısında bulunan veri-öğeleri Şekil 6a'da resmedilmektedir. Şekil'den de görüleceği üzere, olay veri yapısında İşlem Id, Müşteri Id, Müşteri Adı, Müşteri Soyadı, Ürün Kodu, Ürün Adı, Ürün Miktarı, Ürün Kategorisi, Ürün Ücreti, Toplam Tutar, Vergi, Cep Telefon Numarası, Ev Telefon Numarası, İş Telefon Numarası, E-Posta gibi veri öğeleri bulunabilmektedir.

Önceden belirlenen sürekli sorgulara (kurallara) uygun olay zincirinin oluşması durumunda, karmaşık olay (complex event) yaratılmaktadır. Karmaşık olay verisi, sistemin alması gereken aksiyonları içermektedir. Karmaşık olay veri yapısı tasarlanırken, bir e-ticaret sitesinin kullanıcılarına anlık indirim, bonus, kampanya gibi faaliyetleri aksiyon olarak gerçekleştirebileceği düşünülmüştür. Şekil 6b karmaşık olay veri yapısını resmetmektedir. Şekil'den de görüleceği üzere, karmaşık olay veri yapısında Karmaşık Olay Kuralı, Ürün Kodu, Zaman, Aksiyon ve SMS mesajı gibi veri öğeleri bulunabilmektedir. Olay verisi ve karmaşık olay verisinin yanısıra, müşteri bilgilerini ve ürün bilgilerini içeren veri yapıları da bulunmaktadır. Müşteri bilgileri, e-ticaret sitelerine ait CRM gibi legacy sistemlerde bulunabilmektedir.

Prototip uygulamamız için, bu tür bir CRM sisteminden elde edilebilecek bazı müşteri bilgileri, Müşteri Nesnesi Veri Yapısı'nda tanımlanmıştır. Müşteri Nesnesi Veri Yapısı Şekil 6 c'de resmedilmektedir. Şekil'den de görüleceği üzere, Müşteri Nesnesi Veri Yapısında Müşteri Id, Müşteri Adı, Müşteri Soyadı, Müşteri Telefon Numaraları, E-posta gibi veri öğeleri bulunabilmektedir. Müşterilerin satın alma işleminde bulunduğu ürünlerin bilgileri Ürün Veri Yapısı şeklindedir. Ürün Veri Yapısı Şekil 6d'de resmedilmektedir. Şekil'den de görüleceği üzere, Ürün Veri Yapısında Ürün Kodu, Ürün Adı, Ürün Miktarı, Ürün Kategorisi, Ürün Ücreti gibi veri öğeleri bulunabilmektedir.



Şekil 6: Müşteri hareketlerinin oluşturduğu olay yapısı.

Karmaşık olay oluşumu tetikleyecek kurallar, bir kural motoru üzerinde tasarlanarak gerçekleştirilmiştir. Tablo 1'de tek tip olay durumunu içeren örnek kuralları, Tablo 2'de aynı anda iki farklı tip olayı içeren olayları veriyoruz. Buradaki örnekler de, bir müşterinin alabileceği ürün sayısı ve yaptığı harcama miktarı belirli bir eşik değerini geçmesi durumunda, karmaşık olaylar oluşturulması tasarlanmıştır.

Tablo 1'de verilen iki örnekten, ilkinde; e-ticaret sitesi üzerinde, belirli bir zaman penceresi içinde, alınan belirli bir ürün sayısının 5 sayısını geçtiği anda tetiklenecek bir kural

verilmektedir. Tablo 1'de verilen ikinci örnekte; belirli bir zaman penceresi içinde, alınan ürün miktarı tutarı 5000 TL'yi geçtiği zaman tetiklenecek bir kural verilmektedir. Tablo 2'de ise her iki ön-koşulun, aynı zaman penceresi içinde gerçekleştiği anda tetiklenecek bir kural verilmektedir.

Tablo 1: Önerilen sistemde kullandığımız tek tip olaydan oluşan bazı kurallar.

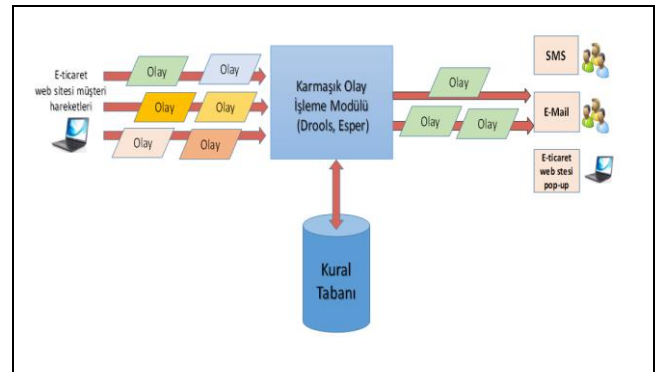
rule "product quantity"
when
\$stream : QuantityStream(quantityCount > 5)
then
\$stream.setAssing(true);
End
rule "total price"
when
\$stream : TotalPriceStream(totalPrice > 5000.0)
then
\$stream.setAssing(true);
End

Tablo 2: Önerilen sistemde kullandığımız birden fazla tip olaydan oluşan bazı kurallar.

rule "quantity & price"
when
\$stream1 : QuantityStream(quantityCount > ""5)
\$stream2 : TotalPriceStream(totalPrice > "5000.0", this
after [1 m 30 s, 3 m] \$stream1)
then
\$stream.setAssing(true);
End

Prototip Sistemin Çalışma Senaryosu: Geliştirdiğimiz prototip uygulamayla temel amacımız müşterinin bir e-ticaret sitesi üzerinde satış öncesi, satış anı ve satış sonrası hareketlerini, takip ederek, müşterilere anlık indirim, bonus, kampanya önerilerinin yapılması gibi faaliyetlerin yapılabilmesini sağlamaktadır. Bu öneriler farklı iletişim kanalları üzerinden (SMS, e-mail vb.) yapılabilir.

Geliştirdiğimiz sistemin yüksek performansta çalışan bir kampanya yönetimi sistemi olması, aranan isterler arasındadır. Şekil 7, sistemin çalışma senaryosunu resmetmektedir. Bu senaryoda, e-ticaret sitesi kullanıcı bir ürün alırken, site üzerinde farklı amaçlı temaslara örnek olarak, bir ürünün fiyatına göre listelenmesi, ürünün görüntülenmesi, ürünün alışveriş sepetine konulması, satın alınması gibi faaliyetler gösterilebilir.



Şekil 7: Karmaşık olay işleme yapısı.

Bu faaliyetler sonucu üretilen olaylar, Karmaşık Olay İşleme Modülü'ne gönderilmektedir. Karmaşık Olay İşleme Modülü,

kural tabanında bulunan kurallara göre karmaşık olayları üretmektedir. Üretilen karmaşık olaylara sebep olan müşterilere anlık indirim, bonus, kampanya önerileri yapılmaktadır. Kampanya önerileri SMS, e-posta ve e-ticaret sitelerine gönderilen pop-up iletişim kanalları sayesinde müşteriye ulaşmaktadır.

Sistemimiz gerçekleştirilirken Akan Veri İşleme Modülü için Apache Spark kullanılmıştır. Spark bir yayınlı/üye-ol mesaj iletim kanalı üzerinden gelen verilerle gerçek zamanlı olarak beslenmektedir. Bu amaçla farklı yayınlı/üye-ol iletişim sistemleri incelenmiş ve aralarından performans açısından en uygun olanı (Apache Kafka) seçilerek önerilen sistem mimarisinin gerçekleştirilmesinde kullanılmıştır. Apache Kafka temelde log kaydına benzer bir yapıda kayıtları tutan ve bu kayıtları diğer sistemlere mesajlaşma kuyruğu şeklinde sunan bir yapıdır. Hata toleranslı, yüksek verimli, yatay olarak ölçeklenebilir yapıdadır. Kafka bir veya daha fazla sunucu ile küme yapısında çalışabilmektedir. Kafka'ya gelen akan verilerin Spark'ın işleyeceği Eşle-İndirge veri yapısına dönüştürülmesi için, Veri Akışı Düzenleyici Modülü olarak Spark Streaming kullanılmıştır. Apache Spark gerçek zamanlı akan veriler üzerinde Eşle-İndirge tabanlı uygulama geliştirmemizi sağlamaktadır.

Müşterilerden, farklı kaynaklar aracılığıyla toplanacak basit ve karmaşık olay verilerinin, büyük veri alt yapısında saklanması, sorgulanabilir olması gerekmektedir. E-ticaret web sitesi üzerinde gerçekleşen müşteri hareketlerinin oluşturduğu olaylar Apache Kafka ya gönderilmektedir. Spark streaming, Kafkaya gelen akan verileri gerçek zamanlı olarak kullanabilmektedir. Spark Streaming ile olay bazlı, asenkron, ölçeklenebilir, hata toleranslı uygulamalar geliştirilebilmektedir. Farklı akan veri kaynaklarından (Kafka, Twitter, TCP socket) gelen gerçek zamanlı olarak, Spark Streaming'e entegre edilebilmektedir.

Gerçek zamanlı veri yüksek seviye fonksiyonlarla (map, reduce, join, window) işlenebilmekte ve işlenen veri, dağıtık dosya sistemlerine yazabilmektedir.

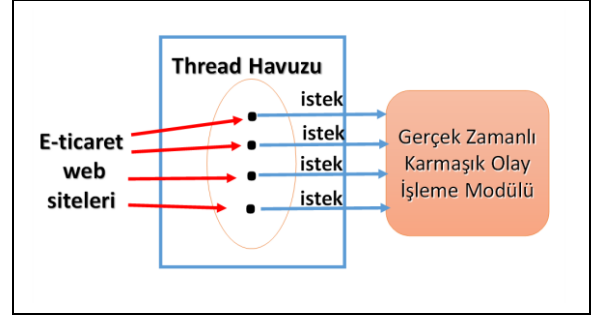
Sistemimizde verinin korunması amacıyla Kafka ya gelen olaylar aynı zamanda Hadoop Dağıtık Dosya Sistemi'ne kaydedilmektedir. Kafka'dan Spark Streaming'e gelen olaylar belirlediğimiz kurala göre kural motoru tarafından seçilmektedir. Kural motorları, uygulama kodundan harici olarak oluşturulan iş kurallarını çalıştıran tak-çıkır yazılım parçalarıdır. Böylece ana uygulama kodunu değiştirmeden değişen iş kurallarına göre yazılım kolaylıkla konfigür edilebilmektedir. Sistemimizde kural motoru olarak Drools kullanılmıştır.

5 Değerlendirme

Bu kısımda sistem oluşturulurken kullanılan teknolojik araçlar ve sistem çalışma performansından bahsedilmektedir. Sistem gerçekleştirilirken Java 1.8.66 sürümü kullanılmıştır. Zaman fonksiyonu olarak, Java'nın 1.8.66 sürümünde bulunan System.currentTimeMillis fonksiyonu kullanılmıştır.

Sistemimiz, orta katman mesaj iletim kanalı Kafka 0.9.0 sürümü kullanılarak gerçekleştirilmiştir. Büyük veri işlemlerini gerçekleştirmek için Spark 1.5.2 sürümü kullanılırken, olaylar üzerinde önceden belirlenmiş kuralları çalıştırmak için Drools 6.3.0 sürümlü kural motoru kullanılmıştır. Sistemimiz şu özelliklerdeki sunucu üzerinde çalıştırılmıştır: Intel i7-3630QM (2.40 GHz) işlemci, 8GB RAM, Ubuntu 14.04.3 LTS işletim sistemi.

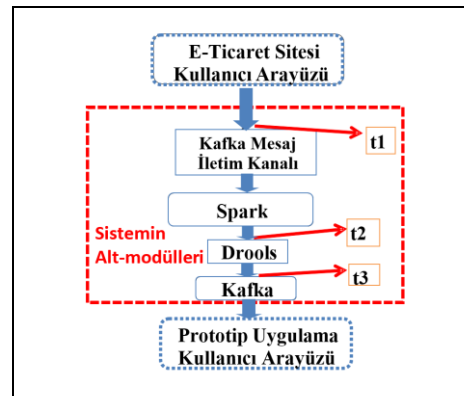
Prototip uygulamanın performansını ölçmek için; sistem performansı, standart erişim operasyonları üzerine artan iş yükü uygulanarak araştırılmıştır. Bu test işleminde, Kafka mesaj iletim kanalı ve Spark veri işleme katmanının üzerinde geliştirdiğimiz yazılımın performansı incelenmiştir. Şekil 8'de görüldüğü üzere, sistem üzerindeki yük artarken sistemin performansı değerlendirilmiştir.



Şekil 8: Birden çok e-ticaret sitesinin eş zamanlı olarak prototip uygulamaya istek göndermesi.

Şekilde thread havuzunda bulunan her bir thread, sistemimizi kullanan farklı e-ticaret sitelerini temsil etmektedir. E-ticaret sitelerinin her biri, olaylar üreterek sistemimize eş zamanlı istek yollamaktadırlar. Bu koşullar altında, sisteme saniyede gelen olay sayısı hesaplanarak sistemin performansı değerlendirilmiştir. Saniyede gelen olay sayısı artırılarak sistemin cevap verme süresi incelenmiştir.

Saniyede sistemimize gelen olay sayısına karşılık, sistemin alt bölümlerindeki çalışma süreleri elde edilmiştir. Şekil 9'da kırmızı noktalarla ayrılmış kısım prototip uygulamayı, mavi düz çizgilerle çizilmiş kutularla ayrılan modüller, prototip uygulamanın alt modüllerini göstermektedir. Şekilde gösterildiği üzere sistemin performansı daha iyi anlayabilmek için, sistemin alt modüllerinin performansı ayrı ayrı incelenmektedir. t1 süresi, kullanıcıların e-ticaret web sitesindeki hareketleri sonucu oluşan olayların, Kafka mesaj iletim kanalına aktarıma başladığı süreyi göstermektedir. t2, Kafka'da bulunan olayların Spark üzerinde gerçek zamanlı olarak eşle-indirge veri yapısına dönüştürülme işleminin bittiği süreyi göstermektedir. Dolayısıyla t2-t1 süresi Kafka mesaj iletim kanalının ve Spark'ın olayları işleme süresidir. t3 süresi, Spark'tan gelen olayların, Drools kural motoru üzerinde bulunan kurallar kullanılarak çalıştırılıp, oluşan karmaşık olayların tekrar Kafka üzerinde gönderiminin başladığı süredir. Dolayısıyla t3-t2 süresi Drools kural motorunun, olaylar üzerinde kuralları çalıştırma süresidir.

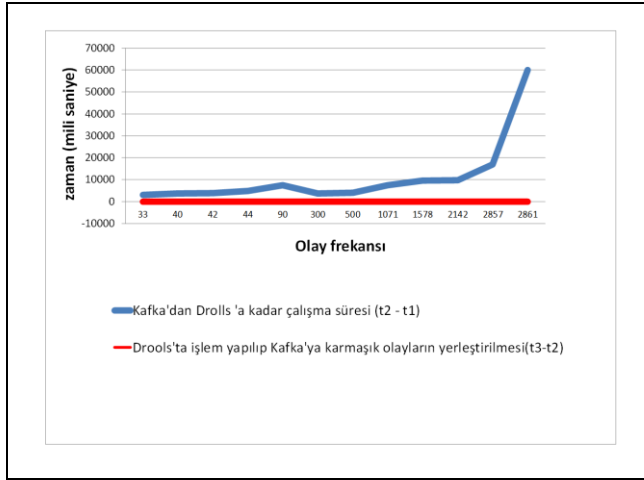


Şekil 9: Test ortamı, alt bölümleri ve zaman değerleri.

Şekil 10'da bulunan grafikte x eksenini olay frekansını göstermektedir. Y eksenini ise sistemde bulunan alt bölümlerin, olay frekans değerlerine karşılık ortalama çalışma süresini (t2-t1 ve t3-t2) göstermektedir. Burada, olay frekansını aşağıdaki şekilde tanımlıyoruz.

Olay frekansı: 1 sn.'de içinde e-ticaret sitelerinden sisteme gelen olay miktarını göstermektedir. Saniye bazında olayların gelme sıklığının nasıl değiştiğini ortaya koymak için kullandığımız bir metriktir.

Şekil 10'un analiz edilmesiyle sistemin artan iş yükü karşısında nasıl bir tepki verdiği görülmektedir. Değerlendirmenin sonucunda, prototip uygulamamızın artan iş yükü karşısında ihtiyaç duyduğu çalışma süresi ihmal edilebilecek kadar az olduğu görülmektedir. Şekil 10'da görüldüğü üzere, sistemin performansı 2857 olay/sn. değerine kadar neredeyse aynı kalmaktadır. 2857 olay/sn. değeri; sistemin performansının, saniyede gelen olay sayısının fazla olmasından dolayı düşmeye başladığı eşik değerdir. Bu eşik değeri, sistemimizde bulunan olayları kuyruk halinde tutan Kafka mesaj iletim kanalının getirdiği ek yükten kaynaklanmaktadır. Ek yük, Kafka'ya verimli olarak çalışabileceğinden fazla miktarda olayın eş zamanlı gelmesinden kaynaklanmaktadır. Kafka'nın verimli olarak çalışabildiği olay frekansı, prototip uygulamanın da verimli olarak çalışabildiği olay frekansıdır. Kafka verimli olarak çalışabileceği olay frekansı eşik değerini geçtikten sonra, mesaj iletim kanalına sırayla gelen her bir olay, bir önceki gelen olaydan daha fazla kuyrukta beklemektedir. Bu da sistemin performansını düşürmektedir.



Şekil 10: Değerlendirme sonuçları grafik gösterimi.

Şekil 10'da görüldüğü üzere, olayların Drolls kural motorunda ortalama çalıştırılma süresi (t3-t2) değişim göstermemektedir ve ihmal edilebilecek kadar azdır. Drolls kural motoru yüksek olay oranlarında sistemimizle uyumlu bir şekilde çalışmaktadır.

6 Sonuçlar

Bu makalede, olay tabanlı sistemler üzerinde, karmaşık olay işleme tekniğine dayalı gerçek zamanlı izleme sistemleri tanıtılmıştır. Araştırma kapsamında katkılarımızı aşağıda özetliyoruz.

E-ticaret alanında; satış anı müşteri hakaretlerinin gerçekleştiği ortamlardan (e-ticaret siteleri, mağaza zincirleri vb.) toplanan olayları izleyebilen ve anında aksiyonlar üretebilen bir aktivite izleme ve yanıtalama alt yapısı için yazılım

mimarisi öneriyoruz. Burada, anlık işlenen veriden varılacak sonuçlara göre anlık faydalar (kampanya, indirim, bonus vb.) üretilmesini sağlayacak olan bir sistem geliştirdik. Önerilen mimarinin kullanılabilirliğini göstermek amacıyla geliştirilen prototip uygulamanın performans testleri gerçekleştirilmiştir. Geliştirilen sistem, e-ticaret sitelerindeki, tüketicilerin anlık hareketlerini izleyerek, tüketiciye yönelik anlık eylemler üretebilmesini sağlayabilecek niteliktedir.

Önerilen sisteminin prototip'inin performansı, sistem üzerine uygulanan artan iş yükü ile değerlendirilmiştir. Değerlendirmeden çıkan sonuca göre önerilen sistemi yüksek olay frekans değerleri için bile verimli bir şekilde çalıştığı gözlemlenmiştir. Elde edilen performans sonuçları, önerdiğimiz mimarinin başarılı olduğunu göstermektedir.

7 Kaynaklar

- [1] Bifet A. "Mining big data in real time". *Journal of Informatica*, 37(1), 15-20, 2013.
- [2] Scott M. *Real-Time Marketing and PR, Revised: How to Instantly Engage Your Market, Connect with Customers, and Create Products that Grow Your Business Now*, Wiley Desktop Editions Series, John Wiley & Sons, 2011.
- [3] Mishra N, Meyerson A, Guha S, Motwani R. "Streaming-data algorithms for high-quality clustering". *Proceedings of the 18th International Conference on Data Engineering, ICDE-2002*, San Jose, CA, USA, 26 February-1 March 2002.
- [4] Leskovec J, Rajaraman A, Ullman J. *Mining of Massive Datasets*, Cambridge University Press, www.mmds.org, 2010.
- [5] Shanahan G, Laing D. "Large scale distributed data science using apache spark". *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, NSW, Australia, 10-13 August, 2015.
- [6] Surshanov, S. "Using apache storm for big data". *Computer Modelling & New Technologies*, 19(3B), 14-17, 2015.
- [7] Tas Y, Baeth MJ, Aktas M. "An Approach to Standalone Provenance Systems for Big Social Provenance Data", *12th International Conference on Semantics, Knowledge and Grids*, Beijing, China, 15-17 August 2016.
- [8] White T. *Hadoop: The Definitive Guide*. 3rd ed. Sebastopol, CA, O'Reilly, 2015.
- [9] Baeth M, Aktas M. "Detecting misinformation in social networks using provenance data". *Concurrency and Computation: Practice and Experience*, doi: 10.1002/cpe.4263, 31(3), 1-9, 2019.
- [10] Dean J, Ghemavat S. "MapReduce: Simplified data processing on large clusters". *6th Symposium on Operating System Design and Implementation (OSDI-2004)*, San Francisco, CA, 3 October 2004.
- [11] Husain M, Jalab H, Rohani V. *Optimized-Memory Map-Reduce Algorithm for Mobile Learning*. Editors: Zaman HB, Robinson P, Smeaton AF, Shih TK, Velastin S, Jaafar A, Ali NM. *Advances in Visual Informatics*, 249-256, Boston, MA, Springer, 2015.
- [12] Ranjan, R. "Streaming big data processing in data center clouds". *IEEE Cloud Computing*, 1(1), 78-83, 2014.
- [13] Aktas M. "Hybrid cloud computing monitoring software architecture". *Concurrency and Computation: Practice and Experience*, 4694, 30(21), 1-9, 2018.
- [14] Aktas M, Astekin M. "Provenance aware run-time verification of things for self-healing Internet of Things applications". *Concurrency and Computation: Practice and Experience*, 4263, 31(3), 1-9, 2019.

- [15] Fox G, Aktas M, Aydin G, Bulut H, Pallickara S, Pierce M, Sayar A, Wu W, Zhai G. *Real Time Streaming Data Grid Applications*. Editors: Davoli F, Palazzo S, Zappatore S. Distributed Cooperative Laboratories: Networking, Instrumentation and Measurements, 253-267, Boston, MA, Springer, 2006.
- [16] Fox G, Aktas M, Aydin G, Gadgil H, Pallickara S, Pierce M, Sayar A. "Algorithms and the Grid". *Computing and visualization in science*, 12(3), 115-124, 2009.
- [17] Baeth M, Aktas M. "An approach to custom privacy policy violation detection problems using big social provenance data". *Concurrency and Computation: Practice and Experience*, 30(21), 1-9, 2018.
- [18] Chen P, Plale B, Aktas M. "Temporal representation for mining scientific data provenance". *Future generation computer systems*, 36, 363-378, 2014.
- [19] Chen P, Plale B, Aktas M. "Temporal representation for scientific data provenance". *IEEE 8th International Conference on E-Science (e-Science)*, Chicago, IL, USA, 8-12 October 2012.
- [20] Baloglu A, Aktas M. "BlogMiner: Web blog mining application for classification of movie reviews". *5th International Conference on Internet and Web Applications and Services (ICIW)*, Barcelona, Spain, 9-15 May, 2010.
- [21] Aktas M, Plale B, Leake D, Mukhi N. *Unmanaged workflows: Their provenance and use*. Editors: Liu Q, Bai Q, Giugni S, Williamson D, Taylor J. Data Provenance and Data Management in e-Science, 59-81, Berlin, Heidelberg, Springer, 2012.
- [22] Jensen S, Plale B, Aktas M, Luo Y, Chen P, Conover H. "Provenance capture and use in a satellite data processing pipeline". *IEEE Transactions on Geoscience and Remote Sensing*, 51(11), 5090-5097, 2013.
- [23] Palmer J. "Web site usability, design, and performance metrics". *Information systems research*, 13(2), 151-167, 2002.
- [24] Kaushik A. *Web Analytics: An Hour A Day*, 1st Edition, Indianapolis, Indiana, USA, Wiley Publishing, 2007.
- [25] Aktas M, Pierce M, "High-performance hybrid information service architecture". *Concurrency and Computation: Practice and Experience*, 22(15), 2095-2123, 2010.