KARADENİZ TEKNİK ÜNİVERSİTESİ OF TEKNOLOJİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

2019 – 2020 BAHAR DÖNEMİ
YZM-4008 VERİ MADENCİLİĞİ DERSİ

HAZIRLAYANLAR

352615 – MEHMET KATI

352580 – TUĞBA CAN

352568 – ZEYNEP ÇALAPVERDİ

DERS SORUMLUSU

DOÇ. DR. HAMDİ TOLGA KAHRAMAN

ÖDEV KONUSU: APACHE, APACHE SPARK, APACHE HADOOP VE APACHE KAFKA KONULARI ÜZERİNE YAPILAN ARAŞTIRMALAR

Apache Nedir, Özellikleri Nelerdir?

Resmi ismi Apache Http Server'dır ve Apache Software Foundation tarafından geliştirilmiştir. Web sitesi sahiplerine içeriklerini internet üzerinde yayınlama olanağı sağlar. Apache fiziksel bir sunucu değildir, sadece sunucuda çalışan bir yazılımdır. Sunucu ile web sitesi kullanıcıları (Firefox, Chrome, Safari vs.) arasında bir köprü oluşturarak dosyaları ileri geri taşır. Çapraz platform bir yazılımdır hem Unix hem Windows sunucularda çalışabilir. Apache akıcı ve güvenli iletişimden sorumludur. Modül tabanlı yapısı sayesinde oldukça özelleştirilebilir bir yapıya sahiptir. Apache'nin güvenlik, önbellek, URL yazma, şifre yetkilendirme gibi modülleri bulunur. İstenirse bütün Hostinger planlarında desteklenen Apache yapılandırma dosyası olan .htaccess dosyası ile sunucu yapılandırılması oluşturulabilir.

- Apache, thread tabanlı bir altyapı kullandığından, trafiği fazla olan web sitesi sahipleri performans sorunuyla karşılaşabilir.
- Apache, uygun modülleri yardımıyla farklı yazılım dilleriyle birlikte kullanılabilir. (PHP, Python..)
- Açık kaynak kodlu ücretsiz ticari kullanımlar için kullanılabilir.
- Güvenilir, stabil bir yazılımdır.
- Sıklıkla güncellenir.
- Modül tabanlı yapısı sayesinde esneklik sağlar.
- Kolaylıkla yapılandırılabilir.
- Çapraz platformludur.(Hem Unix hem Windows)
- Wordpress sitelerinde varsayılan olarak çalıştırılabilir.
- Devasa topluluğa sahiptir, destek bulmak kolaydır.
- Çok fazla yapılandırma seçeneği yüzünden güvenlik açıkları oluşabilir.

Ayrıca Apache diğer içerik yönetim sistemleriyle (Joomla, Drupal vs), web frameworkleriyle (Django, Laravel vs) ve yazılım dilleriyle oldukça uyumlu şekilde çalışabilir. Bu sebeple VPS veya paylaşımlı hosting gibi bütün web hosting platformları için oldukça iyi bir seçimdir.

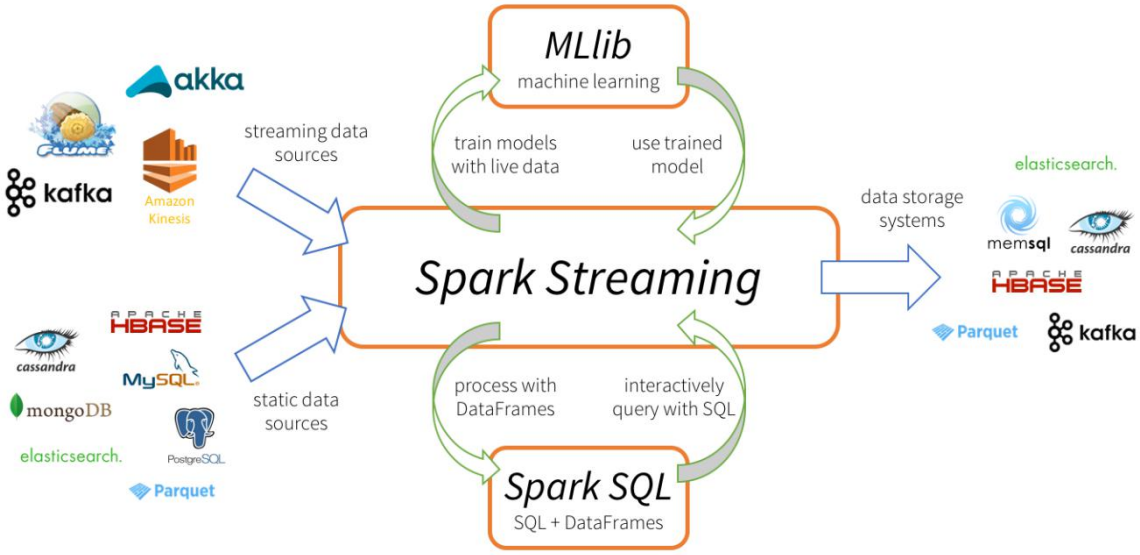
Apache Spark

Veri çalışanlarının veri kümelerine hızlı yinelemeli erişim gerektiren akış, makine öğrenmesi veya SQL iş yüklerini verimli bir şekilde yürütmelerini sağlayan zarif ve etkileyici geliştirme API'lerine sahip hızlı, bellek içi bir veri işleme motorudur. Kısaca verileri paralel olarak işlemeyi sağlar. Apache Spark'ın ana özelliği, bir uygulamanın işlem hızını artıran bellek içi küme hesaplamadır. Spark tüm kümeleri programlamak için örtük veri paralelliği ve hata toleransı ile bir arayüz sağlar. Toplu iş uygulamaları, yinelemeli algoritmalar, etkileşimli sorgular ve akış gibi çok çeşitli iş yüklerini kapsayacak şekilde tasarlanmıştır.

Apache Spark'ın Avantaj ve Dezavantajları

- Büyük ölçekli veri işleme için kullanılan Hadoop MapReduce'dan 100 kat hızlı çalışır.
- Basit programlama katmanı, güçlü önbellekleme ve disk kalıcılığı yetenekleri sağlar.
- Bellek içi hesaplama nedeniyle gerçek zamanlı hesaplama ve düşük gecikme süresi sunar.
- Spark, Java, Scala, Python ve R için üst düzey API'ler sunar. Spark, bu dört dilden herhangi birinde kullanılabilir.

Apache Spark, tüm bileşenlerinin ve katmanlarının gevşek bir şekilde bağlandığı iyi tanımlanmış bir katman mimarisine sahiptir. Bu mimari, çeşitli uzantı ve kütüphanelerle daha da bütünleşmiştir.



Apache Spark mimarisi, iki ana soyutlamaya dayanır:

Esnek Dağıtılmış Veri Kümesi (RDD): Spark Cluster üzerinde, verilerle ilgili hesaplamalar yapmamızı sağlayan bileşendir.

Yönlü Düz Ağaçlar (DAG): Bir işlem yaptırdığımızda, işlem DAG zamanlayıcısına gönderilir. DAG, operatörleri görev aşamalara böler. Kısaca Spark yüksek düzeyde RDD işlemlerini zamanlayan ağaçlı mimari bir bileşene sahiptir.

Spark Ekosistemi

1.Spark Core

- Büyük ölçekli paralel ve dağıtılmış veri işleme için temel motordur.
- Sahip olduğu kütüphaneler ile akış, SQL ve makine öğrenmesi gibi çeşitli iş yüklerine izin verir.
- Bellek yönetimi ve hata kurtarma, bir kümedeki işleri planlamak, dağıtmak ve izlemek ve depolama sistemleriyle etkileşimden sorumludur.

2.Spark Streaming

- Gerçek zamanlı akış verilerini işlemek için kullanılan Spark bileşenidir.
- Gerçek zamanlı veri akışlarının yüksek verimli işlenmesini sağlar.

3.Spark SQL

- Spark'ın işlevsel programlama API'si ile ilişkisel işlemeyi birleştiren yeni bir modüldür.
- SQL veya Hive Query Language aracılığıyla veri sorgulamayı destekler.
- RDBMS veritabanları için Spark SQL performans arttırıcı bir çözüm sunar.

4.GraphX

- GraphX, grafikler ve grafik paralel hesaplamalar için Spark API'dir.

5.MLlib (Machine Learning)

- MLlib, Makine Öğrenimi Kütüphanesi'nin kısaltmasıdır.
- Spark MLlib, Apache Spark'da makine öğrenmesi için kullanılır.

6.*SparkR*

- Dağıtılmış bir veri çerçevesi uygulaması sağlayan bir R paketidir.
- Ayrıca, seçim, filtreleme, toplama gibi işlemleri büyük veri kümelerinde de destekler.

Apache Hadoop

HDFS(Hadoop Distributed File System**):** Sıradan sunucuların disklerini bir araya getirerek büyük ve sanal bir disk oluşturan dosya sistemidir. Bu sayede çok büyük boyuttaki dosyaları sistemde saklayabilir ve işlenmesine olanak sağlar. Hadoop, RDBMS(Relational Database Management System) yani ilişkisel veri tabanı yönetim sistemlerinden farklı olarak verileri tek bir bilgisayarda tutmayıp gelen verileri -her birinin kendine ait işlemcisi ve rami- olan Node'lerde(küme) HDFS dosya sistemi ile denormalize bir şekilde veriyi saklayan ve işlenmesine olanak sağlayan açık kaynak kodlu kütüphanedir.

Hadoop verileri nasıl işler?

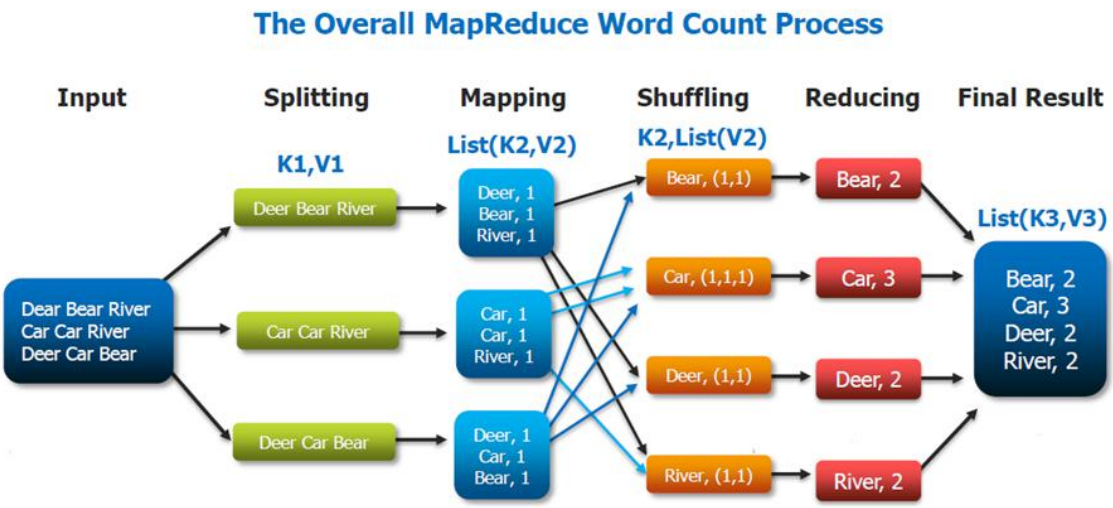
Node'lerde yer alan verileri merkeze toplayıp işlemek yerine SQL diline yakın sorguları Node'lere dağıtarak Node'lerde gerekli işlemi gerçekleştirir. Çünkü her Node'nin kendisine ait işlemcisi ve rami vardır. Bunu yapmasının sebebi ise veriyi merkeze çekip trafik oluşturmamaktır.

Temel Hadoop Bileşenleri Nedir?

En temel Hadoop Bileşenleri HDFS, Map-Reduce ve YARN'dır.

Map-Reduce

Tüm verileri merkeze toplamadığımız için bütün işlemler ayrı ayrı Node'lerde yapılır. Bu işlemler bittikten sonra her Node'den dönen cevap alınır ve toplam sonuç oluşturulur. Bu işlemler bütüne Map-Reduce denir. Map-Reduce işlemleri 6 adımda gerçekleşir:



1-Input: Veri girişlerinin yapıldığı adımdır.

2-Splitting: Gelen veriler bu aşamada işlemesi daha kolay olabilmesi için parçalara bölünür.

3-Mapping: Veriler bu aşamada ilgili düğümlere dağıtılır ve kaç tane yedeği olacağı bu adımda belirtilir. Ve daha sonrasında ver ilgili düğümde işlenir.

4-Shuffling: Her düğümde verilerin sayma işlemi yapılır. Örneğin bir text dökümanını input olarak verdiysek ve sonuç olarak hangi kelimenin kaç defa geçtiğini arıyorsak bu aşamada kelime sayıları Node'lerde belirlenir.

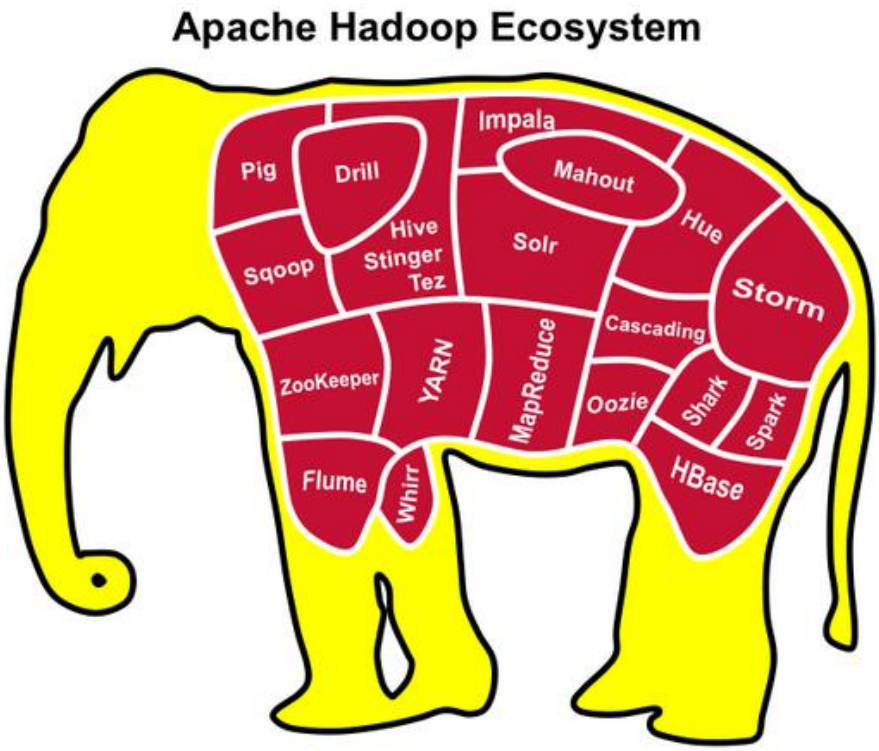
5-Reducing: Her Node'den gelen sonuç bu aşamada toplanır.

6-Final Result: Sonuçlar artık elimizdedir. Bunun raporlamasını yapabiliriz.

Hadoop'un Avantajları ve Dezavantajları

1. Çok fazla olan veri, tek bir bilgisayarda RDBMS yöntemiyle tutulamıyor.
2. Veriyi işlemek bir yana verinin tutulması için bile HDFS dosyalama sistemine ihtiyaç vardır.
3. HDFS ile verinin saklanması ve yedeklenmesi çözülürken büyük veriyi işleme problemi ortaya çıkmıştır. Bunun için Hadoop ekosistemi ve HDFS için araçlar geliştirilmiştir.
4. Hadoop veriyi kopyalayarak yedeklemeye ihtiyaç duyduğundan dolayı ihtiyaç duyulan alanı arttırıyor.
5. Temel Hadoop bileşenlerinin klasik SQL sorgularına sahip olmaması bir eksiklik olarak görülebilir.
6. Veriyi saklarken ağ üzerinde şifrelemediği için güvenlik anlamında yeterli değildir.
7. Hadoop'un temel bileşenleri olan YARN, HDFS ve Map-Reduce yeterli değildir.

Hadoop Ekosistemi



Veri Erişimi: Pig, Hive
Veri Depolama: HBase, Cassandra
Etkileşim, Görselleştirme, Uygulama, Geliştrime: HCatalog, Lucene, Hama, Crunch
Veri Serileştirme: Avro, Thrift
Veri İstihbaratı: Drill, Mohout
Veri Entegrasyonu: Sqoop, Flume, Chuwka
Yönetim: Ambari(Portal)
İzleme: Zookeeper
Orkestrasyon: Oozie

Bazılarının açıklamaları aşağıda verilmiştir.

Hive nedir?

Hive HQL olarak bilinen bir SQL'e çok benzer bir dil ile Hadoop sistemlerinde verilere erişim ve sorgulama gibi işlemleri gerçekleştirir. Gerçek zamanlı sorgulama yapamaz.

Pig: HDFS üzerindeki verinin işlenmesinden sorumludur. Karmaşık veri dönüşüm işlemlerini Java'ya ihtiyaç duymadan Latin gibi betik/script dili ile gerçekleştirmemizi sağlayan Hadoop bileşenidir. Yapısal olan ve yapısal olmayan veriler üzerinde çalışarak veriyi HDFS'de saklayabilir. Hadoop üzerindeki veriyi paralel bir şekilde işler. Peki bunu nasıl yapıyor?
Pig, Latin dilini kullanıyor. Latin dilinde yazılmış görevleri otomatik olarak Java/Map-Reduce görevine çevirir. Kısa tabiriyle Pig, Latin Scriptleri YARN üzerinde çalışan Map-Reduce fonksiyonlarına çevirir. Böylece HDFS Node'leri üzerindeki veri işlenmiş olur.

HBase nedir?

HDFS üzerinde çalışan bir NoSQL veritabanı yönetim sistemidir. SQL desteği sunmaz. Bir HBase sistemi bir grup tablolardan oluşur ve tablolara erişmek için bu tablolarda birincil anahtar kullanılır.

Sqoop nedir?

Yapısal verilerin ETL(Extract/Transform/Load) ile Hadoop'a aktarılması için kullanılır. Bir komut satırı arayüzüne sahiptir.

Ambari nedir?

Hadoop Node'lerini yönetmek için kullanılan web arayüzüdür.

Flume nedir?

Streaming verilerin toplanması ve birleştirilmesi için kullanılır.

HCatalog nedir?

HDFS sisteminde kayıtlı olan her verinin konumunu ve şema bilgisini tutar. Pig Latin dili bu araç üzerinden **HCatLoader**(okuma) **HCatStorer**(Yazma) API'leri ile HCatalog tarafından yönetilen tablolara okuma yazma yapar.

Kafka nedir?

Bir mesajlaşma servisidir. Veriyi Hadoop'a stream olarak aktarırı.

Apache Kafka Nedir?

Pub-sub (Publish-subscribe) tabanlı bir dağıtık mesajlaşma sistemi (Distributed Messaging System) olarak tanımlanabilir. LinkedIn mühendislerinden Jay Kreps liderliğinde geliştirilen ve 2010 yılında Open Source olarak GitHub'a konulan Kafka, 2011 yılında bir Apache Software Foundation Incubator Project olarak önerildi ve 2012 yılında da Apache Kafka adını aldı.

Neden Kafka diye isimlendirilmiş merak edenler için ilk ağızdan açıklamayı da aşağıda görebilirsiniz.

"I thought that since Kafka was a system optimized for writing using a writer's name would make sense. I had taken a lot of lit classes in college and liked Franz Kafka. Plus the name sounded cool for an open source project.

So basically there is not much of a relationship."

Jay Kreps

"Kafka'nın bir yazarın adını kullanarak yazmak için optimize edilmiş bir sistem olduğu için mantıklı olacağını düşündüm. Üniversitede birçok aydınlık ders almıştım ve Franz Kafka'yı sevmiştim. Ayrıca açık kaynak kodlu bir proje için isim kulağa hoş geliyordu.

Yani temelde çok fazla bir ilişki yok."

Jay Kreps

Ne Amaçla Kullanılır?

Gerçek zamanlı veri akışı ve analizi şirketlerin veya kuruluşların anlık olarak gelen güncel bilgiler ile ne stoklamaları neleri satmaları gibi kararları vermesinde yardımcı olur. Şirketler ise karar vermelerini sağlayan bu verileri amaçları doğrultusunda depolayıp analiz etmek için oldukça performanslı olan Apache Kafka'yı kullanırlar.

Örnek olarak popüler ve anlık olarak çok fazla aktif kullanıcıya sahip bir web sitesi, aktif kullanıcıların site içerisindeki davranışlarından haberdar olmak istiyor. Düşünüldüğünde binlerce kullanıcının aynı anda bu sitede aktif olarak bir sürü işlem gerçekleştirdiğini sayfadan sayfaya ilerlediğini, butonlara bastığını ve bu verinin toplanması gerektiğini varsayalım. Oluşan verinin ne kadar hızlı arttığını hayal edin. İşte bu gibi durumlarda şirketler Apache Kafka ve kullandığı mesajlaşma sistemi sayesinde bu kullanıcı loglarını anlık olarak hızlı bir şekilde depolayabilir ve analiz edebilir. Amazon,Netflix, Spotify gibi şirketler bu teknolojiyi aktif olarak kullanmaktadır.

Kafka'nın Avantajları ve Dezavantajları

1. **Hızlı**: Saniyede 2 milyonluk bir yazma performansına sahiptir.
2. **Güvenilir**: Mesajları disk üzerinde saklar ve Cluster'da Replike ederek veri kaybının önüne geçer.
3. **Genişletilebilir**: Sistemi durdurmadan genişletilebilme özelliğine sahiptir.
4. Cursor'ın queue 'da nerede kalındığını consumer bilir.(consumer-centric)
5. Tek bir protokol kullanılıyor. (Native Kafka Protocol)
6. Geliştirme ortamı .NET ise piyasadaki SDK'lardan birini kullanma zorunluluğu vardır. Resmi olarak destek vermiyorlar.

Kullanım senaryoları olarak kullanıcı aktivite takibi, mesajlaşma, Loglama ve Stream Processing örnekleri verebiliriz.

Mesajlaşma Sistemi

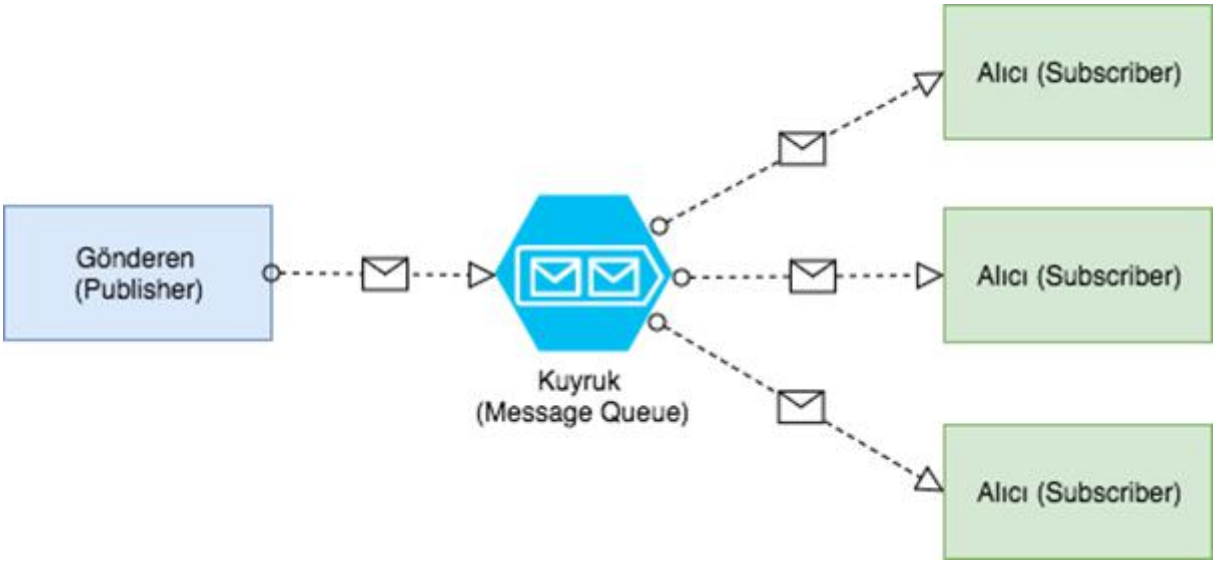
Mesajlaşma sistemi, verinin bir uygulamadan diğerine aktarılmasını sağlar. Distributed Messaging denilen kavram aslında güvenilir bir kuyruk yapısı (Message Queue) olarak da adlandırılır. Mesajlaşma sistemleri 2 yapıda olabilir:

1. Point-to-Point

Bu yapıda birden fazla alıcı olabilmesine rağmen, bir mesaj sadece bir alıcıya iletilir. Mesaj bir alıcıya iletildikten sonra siliniyor şeklinde bir benzetme yapabiliriz.

2. Publisher-Subscriber

Bu yapıda ise mesajlar bir kategori altında toplanırlar. Alıcılar birden fazla kategoriden veri çekebilir. Bu sistemde mesaj gönderen uygulamalara **Publisher**, alıcı uygulamalara da **Subscriber** adı verilir.



Resim-1: Publisher-Subscriber kuyruk yapısı

Kafka Mimarisi

Producer

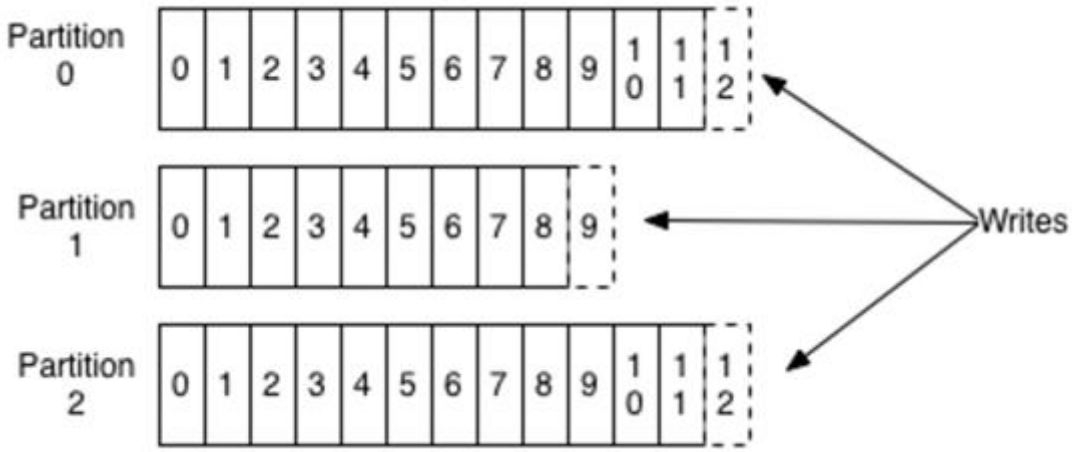
Publisher-Subscriber kelimelerinden Publisher'a karşılık gelir ve bir ya da birden fazla Topic'e mesaj gönderen birimdir.

Topic

Mesajların saklandığı kategorilere Topic adı verilir. Veritabanındaki tablo olarak da düşünülebilir.

Partition

Topic'ler Partition denilen bölümlere ayırılır. **Resim-2**'de göreceğiniz şekilde mesajlar Partition içerisinde Append edilerek yazılır ve baştan sonra doğru okunur.



Resim-2: Topic ve Partition yapısı

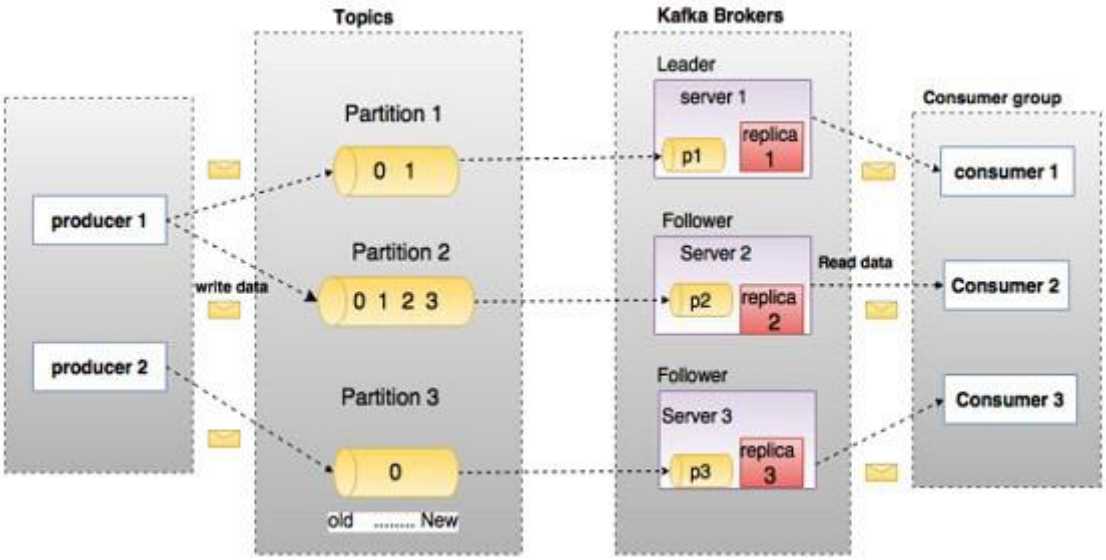
Her Partition farklı bir sunucuda olabilir. Böylece bir Topic birden fazla sunucuya yatay olarak ölçeklendirilebilir.

Broker

Tek bir Kafka sunucusuna Broker adı verilir. Bir diğer deyişle de Kafka Cluster'daki her bir Node'a Broker adı verilir. Cluster içerisinde yer alan Node'lardan biri "lider" olarak tanımlanır ve ilgili Partition için bütün okuma ve yazma işlerinden sorumludur. Diğer Node'lar ise "takipçi" olarak tanımlanır ve Consumer ile benzer şekilde çalışırlar. Eğer lider Node çalışmazsa takipçi Node'lardan birisi lider Node olarak görev alır.

Consumer

Consumer da Publisher-Subscriber kelimelerinden Subscriber'a karşılık gelir Broker'dan veri okur. Bir ya da birden fazla Topic üzerinden veri okuyabilir.



Resim-3: Kafka Mimarisi

ZooKeeper

ZooKeeper, dağıtık uygulamalar geliştirilmesine izin veren, dağıtık bir koordinasyon servisidir. Herhangi bir Broker eklendiğinde ya da çalışmadığı durumlarda, Broker hakkında Producer ve Consumer'ı bilgilendiren bir servis olarak tanımlanır.

Apache Kafka'yı çalıştırabilmek için önce ZooKeeper çalıştırılmalıdır. Çünkü Kafka Partitionlarda tuttuğu Offset bilgileri gibi bilgileri Zookeeperdan alır.