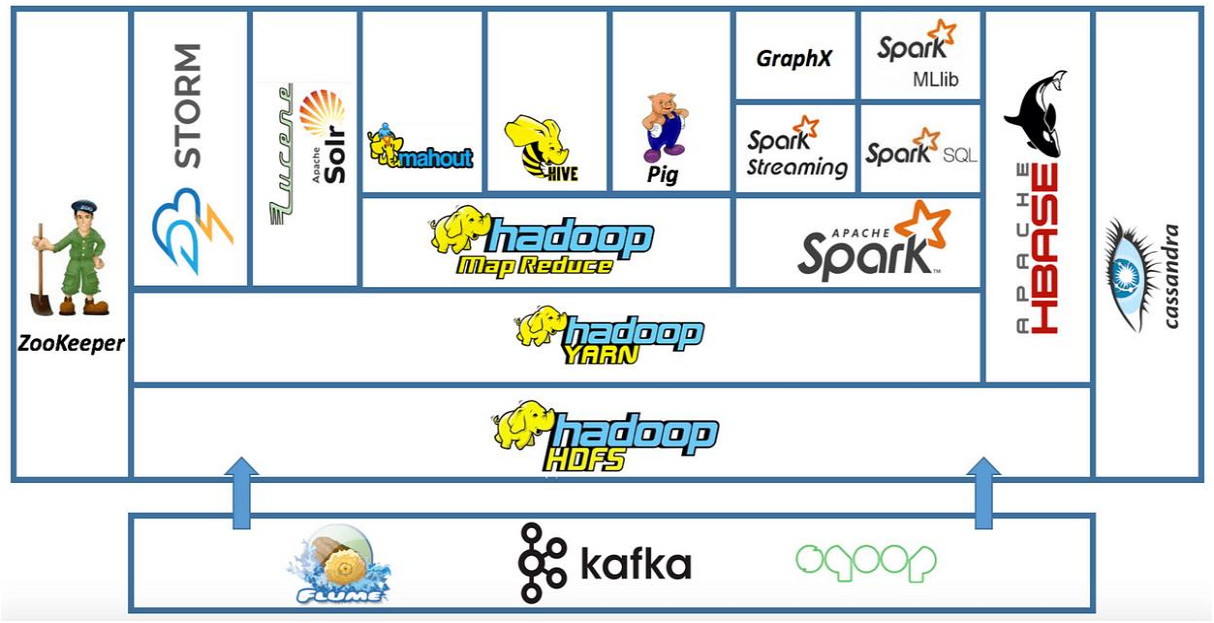
Apache Hadoop

Apache® Hadoop® projesi, **güvenilir, ölçeklenebilir ve dağıtılmış bilgi işlem için açık kaynaklı yazılım geliştiren bir girişimdir.** Apache Hadoop yazılım kütüphanesi, **büyük veri kümelerinin bilgisayar kümeleri arasında dağıtılmış şekilde işlenmesine olanak tanıyan bir çerçeve sunar.** Bu çerçeve, basit programlama modelleri kullanır ve tek bir sunucudan binlerce makineye ölçeklenebilir. Her bir makine, yerel hesaplama ve depolama yeteneklerine sahiptir. (https://hadoop.apache.org/, 2024)

Hadoop, yüksek kullanılabilirlik sağlamak için donanım güvenilirliğine bel bağlamaz. **Bunun yerine, uygulama katmanında hataları tespit etmek ve işlemek üzere tasarlanmıştır.** Böylece, arızalara yatkın olabilecek bir bilgisayar kümesinde bile yüksek kullanılabilirliğe sahip bir hizmet sunar. Bu yaklaşımıyla, **dağıtık veri işleme ve depolama çözümleri için esnek ve güvenilir bir platform sağlar.** (https://hadoop.apache.org/, 2024)

Hadoop, MapReduce programlama modelini kullanan **JAVA programlama dili** ile geliştirilmiş popüler, açık kaynaklı bir Apache projesidir.



Şekil- 1 Hadoop Temsili

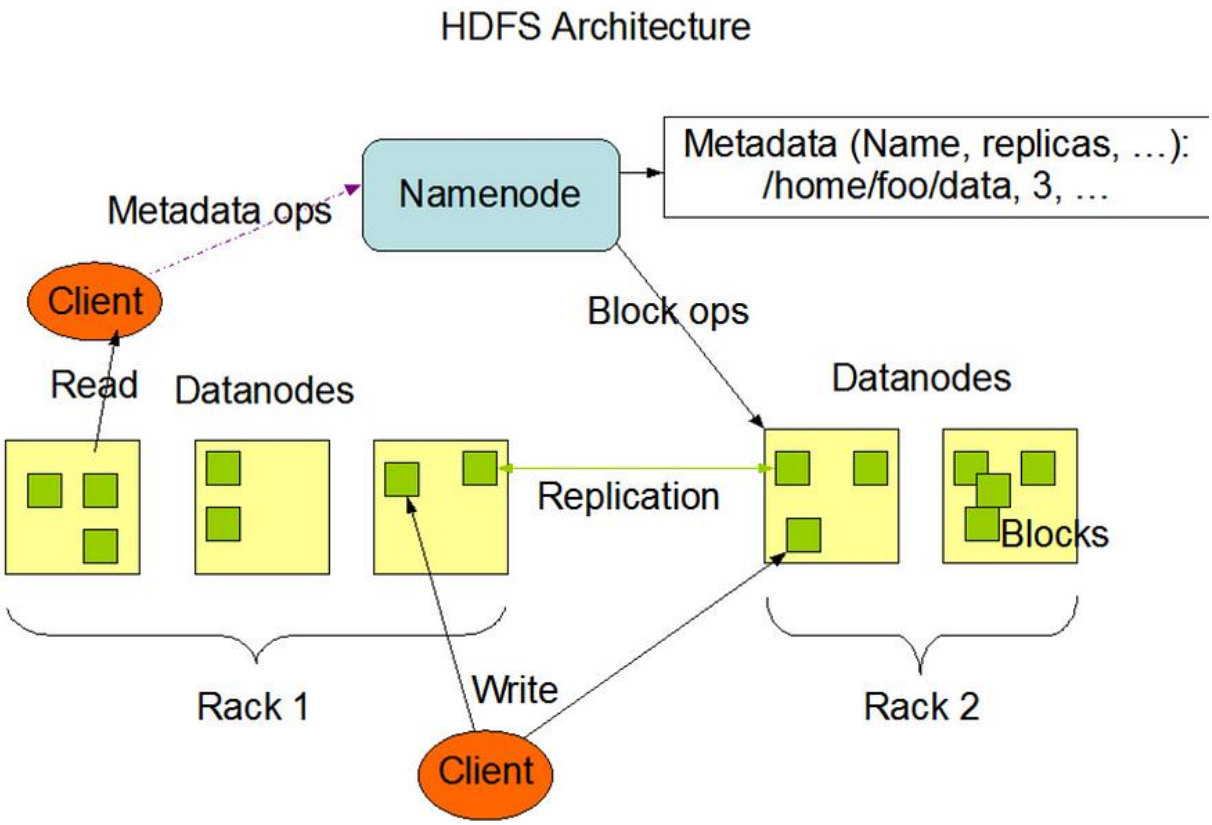
Hadoop Bileşenleri:

1. HDFS (Hadoop Distributed File System- Hadoop Dağıtılmış Dosya Sistemi)

HDFS (Hadoop Distributed File System), hataya dayanıklı, güvenilir, ölçeklenebilir ve kendi kendini onarabilen bir dağıtılmış dosya sistemidir. MapReduce ile entegre çalışarak depolama ve hesaplama işlemlerini destekler. Basit sunucuların disklerini birleştirerek büyük bir disk alanı oluşturabilir ve maliyet etkin bir çözüm sunar.

HDFS, her türde veriyi kabul eder, yüksek bant genişliği sağlar ve verileri otomatik olarak optimize eder. En önemli özelliklerinden biri, hata toleransıdır: bir düğüm arızalansa bile veriyi aktararak hizmetin devamını sağlar. Veri kaybını önlemek için dosyaları birden fazla düğümde çoğaltır, bu yönüyle RAID'e benzese de doğrudan RAID kullanmaz.

Meta veriler NameNode adlı özel bir sunucuda, uygulama verileri ise DataNode'larda depolanır. Sunucular arası iletişim TCP tabanlı protokollerle sağlanır. Google Dosya Sistemi'nden (GFS) esinlenerek, veri güvenliği için dosyaları birden fazla DataNode üzerinde çoğaltır.



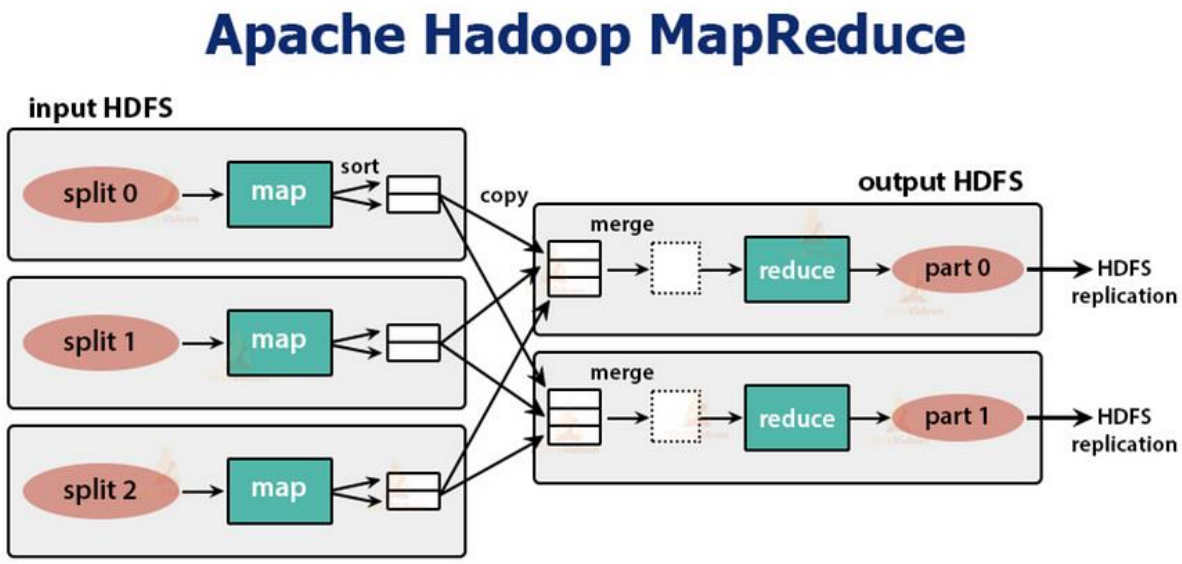
Şekil- 2 Hadoop Dosya Sistemi

2. MapReduce

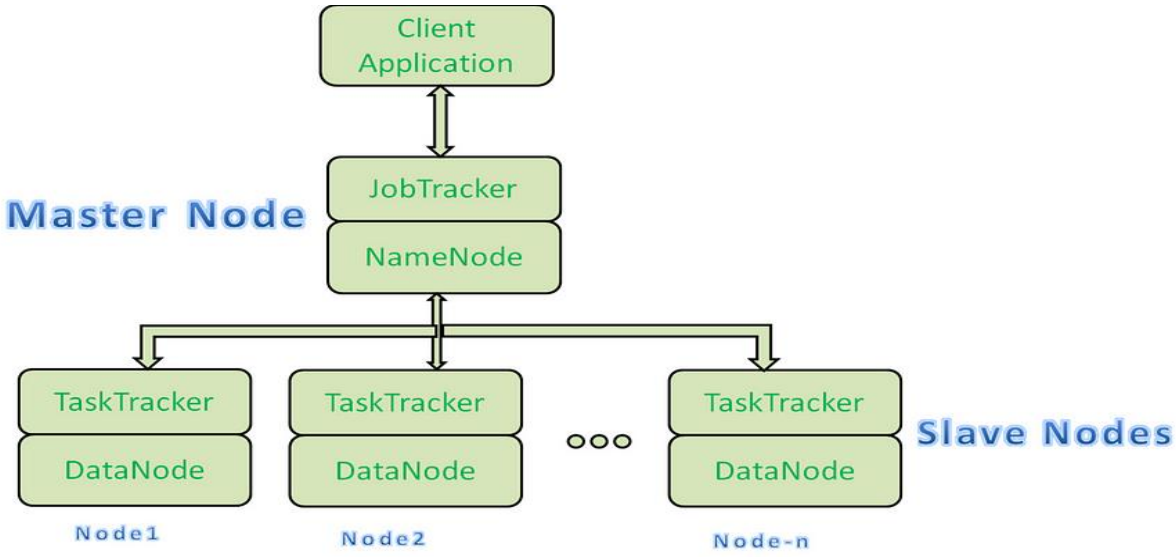
Google tarafından geliştirilmiş dağıtık programlama modeli olan MapReduce, büyük veri kümelerini performanslı bir şekilde paralel olarak işleyen ve analiz eden uygulamalara için kullanılır.

İlk olarak Map(haritalama) aşaması gerçekleşir veriler filtrelenir ve sonrasında sonuç elde etme için Recude(indirgeme) işlemine geçiş yapılır.

Bir Map işlemi tamamlandığında Recude işlemine geçiş yapabilabilmektedir. Tüm Map işlemlerinin beklenmesine gerek duyulmamaktadır.



Şekil- 3 Dağıtık Modelleme



Şekil- 4 Dağıtık Model

HDFS'e benzer şekilde MapReduce, JobTracker ana düğümde ve TaskTracker her uç düğümde çalışmaktadır.

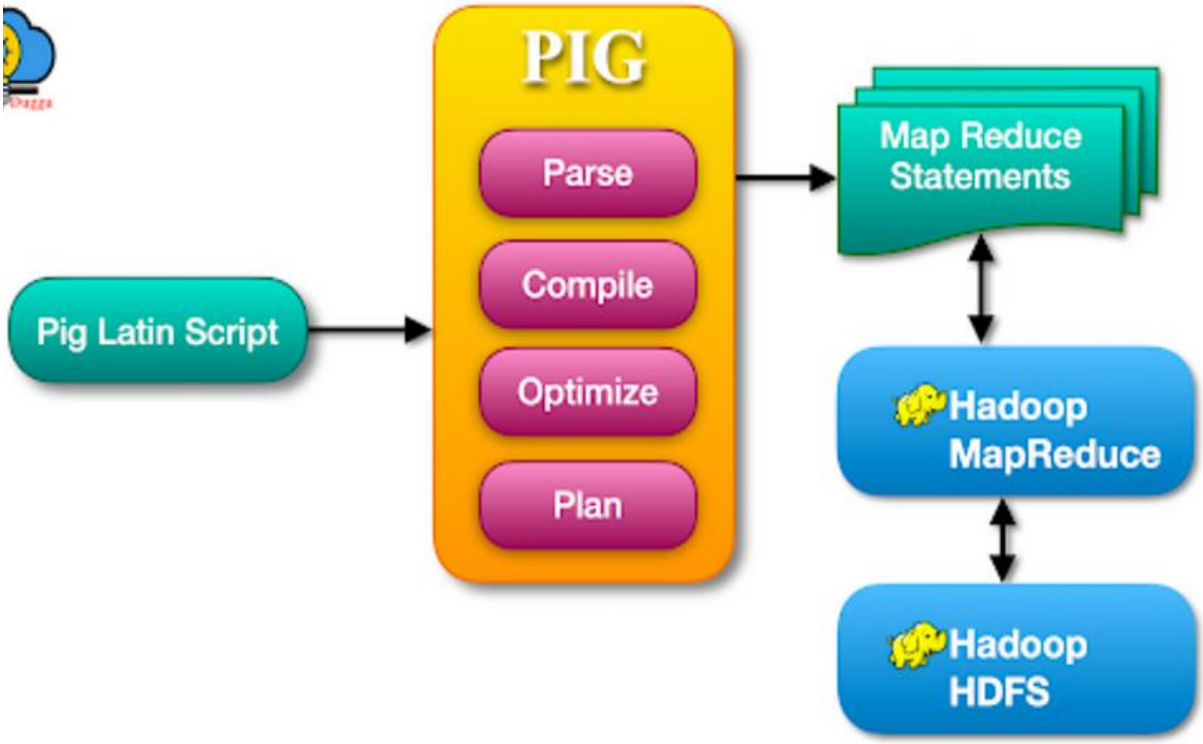
3. HBase

HDFS dosya sistemi üzerine inşa edilmiş, gerçek zamanlı okuma yazma erişimi olan, dağıtılmış, hataya dayanıklı, ölçeklenebilir, ilişkisel olmayan yani noSQL bir veri tabanıdır. HBase, rasgele gerçek zamanlı okuma/yazma işlemleri için büyük veritabanlarında kullanılabilmektedir.

HBase'de SQL sorgu dili desteği bulunmamaktadır fakat Hive ile yazılmış SQL sorgularının erişimi için geliştirilen bir Hive/HBase projesi bulunmaktadır. HBase küme yönetimi için ZooKeeper kullanmaktadır.

4. Pig

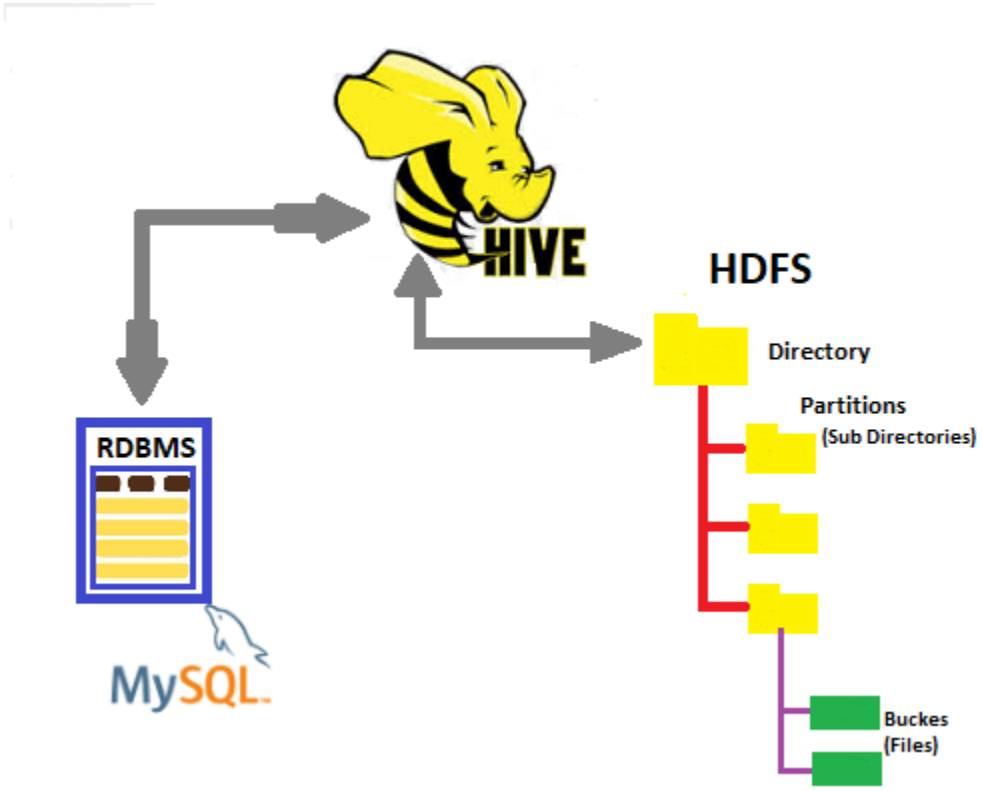
Pig, Hadoop üzerinde çalışan MapReduce programları oluşturma için kullanılan yüksek seviyeli bir Apache projesidir. Verilerin toplu olarak işlenmesi için tasarlanan bu programlama dili, daha az satır kod yazarak geliştirme ve test süresini azaltmaktadır. Bu programlama dili ile daha az kod yazarak işlem gerçekleştirilir fakat yürütme hızında daha yavaş olduğu görülmektedir.



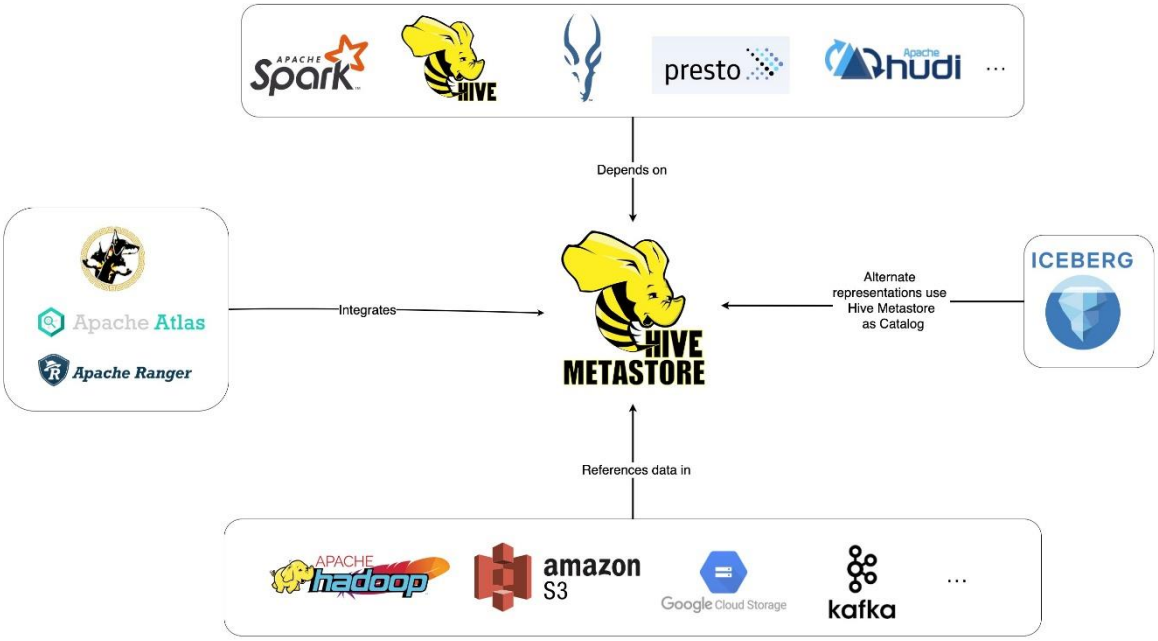
Şekil- 5 PIG Modeli

5. Hive

Facebook tarafından geliştirilen ve SQL türü sorgu dili ile Hadoop'u sorgulamak için tasarlanmış bir veri ambarı teknolojisidir. Hadoop üzerine inşa edilmiş bu programlama dili HiveQL ile kullanıcılar tabloları ve sütunları tanımlar. Veriler tanımlanan bu tablolara yüklenir ve buradan okunurlar. SQL benzeri programlama dili kullanan HiveQL ile analizler, raporlar, özetler oluşturulabilmektedir. HiveQL ile oluşturulan sorgular MapReduce işlemi başlatırlar ve toplu işleme için uygundur fakat Hive gerçek zamanlı sorgulamalar yapamazlar.



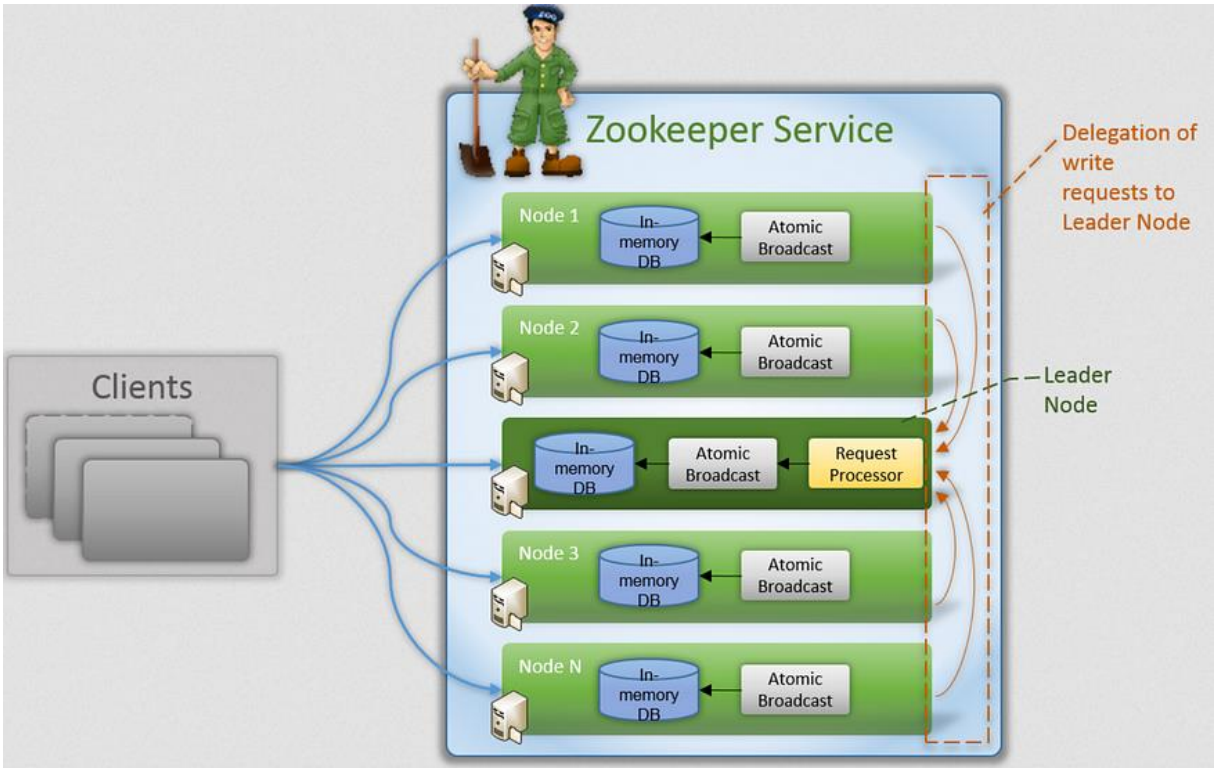
Şekil- 6 HIVE Gösterimi



Şekil- 7 HIVE Şematik Sorgusu

6. ZooKeeper

Dağıtık sistemleri senkronize etmek için kullanılan Apache lisanslı açık kaynaklı bir projedir. Yüksek performanslı, ölçeklenebilir bu yazılım projesi ile eşitleme, adlandırma ve konfigürasyon yönetimi gibi üst düzey hizmetler dağıtık sistemlerde uygulanabilmektedir.



En güzel özelliklerinden birisi de her sunucuda aynı anda birden fazla istemci bağlantısını yönetebilmesidir. Her istemci canlı olduğunu bilmek ve sunucuya bağlı olduğundan emin olmak için ZooKeeper sunucusuna sinyal gönderirler. Eğer bu sinyal işleminde bir süre sunucudan cevap alınmazsa, istemci kümedeki başka bir sunucuya bağlanır.

Hadoop Neden Önemli?

- **Büyük Veri ile Çalışır**: Petabaytlarca veriyi işleyebilir.
- **Yatay Ölçeklenebilirlik**: Daha fazla sunucu ekleyerek kolayca genişletilebilir.
- **Ekonomik**: Açık kaynaklı olduğu için ticari alternatiflere kıyasla daha düşük maliyetlidir.
- **Hata Toleransı**: Veriler, kopyalanarak birden fazla düğümde saklanır. Bu, donanım arızalarına karşı dayanıklılık sağlar.

Kullanım Alanları:

- **Finans**: Dolandırıcılığı tespit etmek ve risk analizi yapmak.

- **Sağlık**: Hasta verilerini analiz etmek ve yeni ilaçlar geliştirmek.
- **Pazarlama**: Kullanıcı davranışlarını anlamak ve kişiselleştirilmiş kampanyalar geliştirmek.
- **E-ticaret**: Öneri sistemleri oluşturmak.

ÖRNEK KULLANILAN GERÇEK SEKTÖR

9 Şubat 2008'de Yahoo! Inc., dünyanın en büyük Hadoop üretim uygulamasını başlattığını duyurdu. Yahoo! Arama Web Haritası, 10.000'den fazla çekirdeğe sahip Linux kümelerinde çalışan bir Hadoop uygulamasıdır ve her Yahoo! web arama sorgusunun bir parçası olarak kullanılır. Yahoo!, birden fazla Hadoop kümesine sahiptir, ancak hiçbir HDFS dosya sistemi veya MapReduce işi birden fazla veri merkezine bölünmez.

Her Hadoop küme düğümü, Hadoop dağıtımını içeren bir Linux görüntüsüyle önyüklenir. Bu kümelerin yaptığı işlerin, Yahoo!'nun arama motoru için dizin hesaplamalarını içerdiği bilinmektedir. Haziran 2009'da Yahoo!, Hadoop sürümünün kaynak kodunu açık kaynak topluluğunun kullanımına sunmuştur.

2010 yılında Facebook, 21 PB depolama kapasitesine sahip olan dünyanın en büyük Hadoop kümesine sahip olduğunu iddia etti. Haziran 2012'de bu kapasitenin 100 PB'ye ulaştığını açıkladılar. Aynı yılın ilerleyen dönemlerinde, Facebook'un Hadoop kümesindeki verilerin günde yaklaşık yarım PB büyüdüğü duyuruldu.

2013 itibarıyla Hadoop'un benimsenmesi geniş ölçüde yaygınlaştı ve Fortune 50 şirketlerinin yarısından fazlası Hadoop'u kullanıyordu.

ÖRNEK KULLANIMI

HDFS, verilerin dağıtık bir şekilde saklandığı Hadoop'un temel bileşenidir.

Örnek: Veriyi Yükleme ve Depolama

Elimizde büyük bir müşteri verisi dosyası olduğunu düşünelim:

```
customer_id,customer_name,city
1,John Doe,New York
2,Jane Doe,San Francisco
3,Bob Smith,Los Angeles
...
```

Bu dosyayı **Hadoop** ekosistemine aktarmak için şu işlemleri yaparız:

Komut

```
hadoop fs -put customers.csv /data/customers/
```

Sonuç: Veri, HDFS'ye yüklendi ve dağıtık olarak saklandı.

MapReduce, büyük veri setlerini analiz etmek için kullanılan bir programlama modelidir.

Örnek: Şehir Bazında Müşteri Sayısını Hesaplama

Bu aşamada yukarıdaki müşteri verilerini işleyerek şehir bazında müşteri sayılarını hesaplarız.

Map Görevi:

- Her satırı işler ve şehir adını alır. **Örneğin**:

```
New York 1
San Francisco 1
Los Angeles 1
```

Reduce Görevi:

- Aynı şehir adlarını toplar ve birleştirir:

```
New York 100
San Francisco 50
Los Angeles 75
```

Komut

```
hadoop jar customer_analysis.jar CityCount /data/customers /output/customers_city_count
```

Sonuç: Şehir bazında müşteri sayısı hesaplandı.

HBase, Hadoop üzerinde çalışan bir NoSQL veritabanıdır.

Örnek: Müşteri Verilerini HBase'de Saklama

Müşteri bilgilerini hızlı sorgulama için **HBase'e** aktarabiliriz.

Komutlar:

1. HBase tablosu oluşturma

```
create 'customers', 'info'
```

2. HDFS'deki veriyi tabloya ekleme

```
hadoop jar hbase-loader.jar LoadToHBase /data/customers customers
```

Sonuç: Müşteri verileri HBase tablosunda saklandı ve hızlı sorgulamalar için hazır.

Pig, veri işleme için kullanılan yüksek seviyeli bir betik dilidir.

Örnek: Yaş Ortalamasını Hesaplama

Müşteri verilerinde bir de yaş kolonunun olduğunu düşünelim. Bu veriyi işleyip yaş ortalamasını bulalım.

Pig Script:

```
customers = LOAD '/data/customers/' USING PigStorage(',') AS (id:int, name:chararray, city:chararray, age:int);
grouped = GROUP customers ALL;
average_age = FOREACH grouped GENERATE AVG(customers.age);
DUMP average_age;
```

Sonuç: Pig, yaş ortalamasını hızlıca hesaplar.

Hive, SQL benzeri bir dil (HiveQL) ile büyük veri analizine olanak sağlar.

Örnek: SQL ile Şehir Bazında Sorgulama

Müşteri tablosunu **Hive** üzerinden sorgulayarak, şehir bazında kaç müşteri olduğunu görebiliriz.

Hive Komutları:

```
CREATE TABLE customers (
    id INT,
    name STRING,
    city STRING,
    age INT
) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
```

Veriyi içeri aktarma:

```
LOAD DATA INPATH '/data/customers/' INTO TABLE customers;
```

Şehir bazında müşteri sayısı sorgulama:

```
SELECT city, COUNT(*) FROM customers GROUP BY city;
```

Sonuç: Hive, SQL benzeri bir dil kullanarak veri analizi yapmayı kolaylaştırır**.**

ZooKeeper, Hadoop ekosistemindeki bileşenlerin koordinasyonu için kullanılır.

Örnek: Hadoop Cluster Sağlığını İzleme

ZooKeeper, Hadoop bileşenlerinin durumunu takip ederek sistemin kararlı çalışmasını sağlar.

ZooKeeper, doğrudan kullanıcı etkileşiminden çok altyapı süreçlerini yönetir. Örneğin, bir MapReduce görevi çalışırken**:**

- **Görev Yönetimi**: ZooKeeper, bir görevin tamamlandığını veya başarısız olduğunu raporlar.
- **Sistem Sağlığı**: Hadoop düğümlerinin bağlantı durumlarını izler.

Komut: ZooKeeper durumunu kontrol etmek için:

```
zkServer.sh status
```

Sonuç: Cluster bileşenleri sorunsuz çalışıyor mu, ZooKeeper üzerinden kontrol edilir.

Genel Akış

1. **HDFS**: Veriler yüklendi.
2. **MapReduce**: Veriler analiz edildi.
3. **HBase**: Hızlı erişim için veriler saklandı.
4. **Pig** ve **Hive**: Veri üzerinde analiz ve sorgular yapıldı.
5. **ZooKeeper**: Sistem koordinasyonu ve sağlık kontrolü sağlandı.