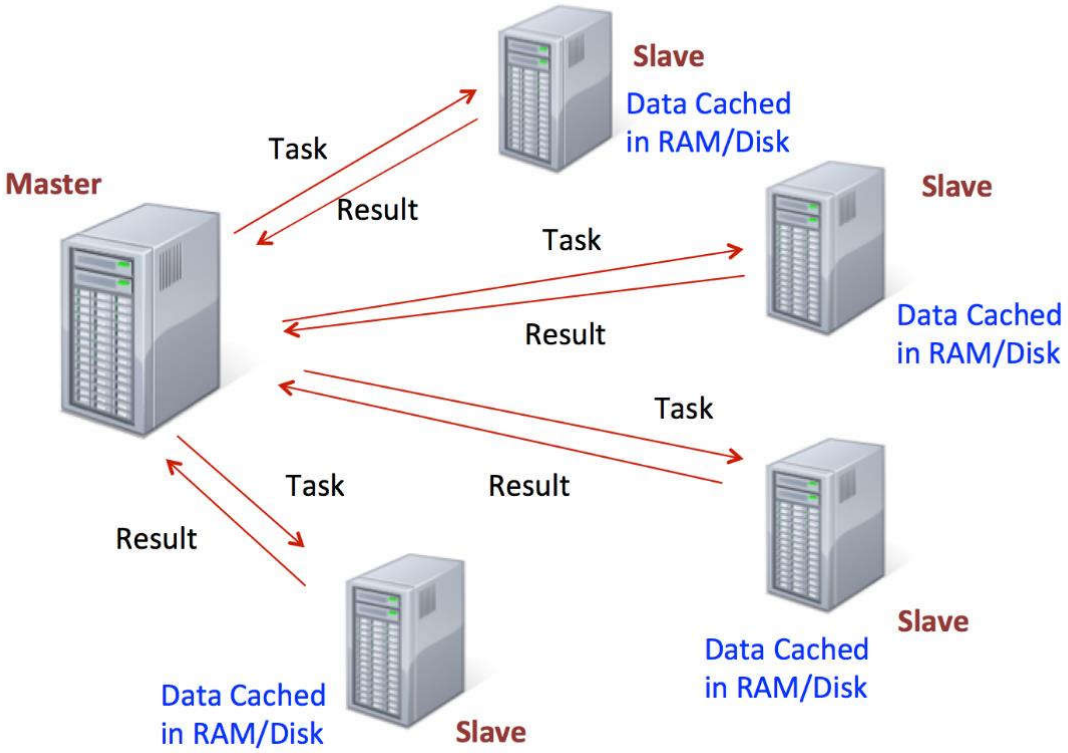
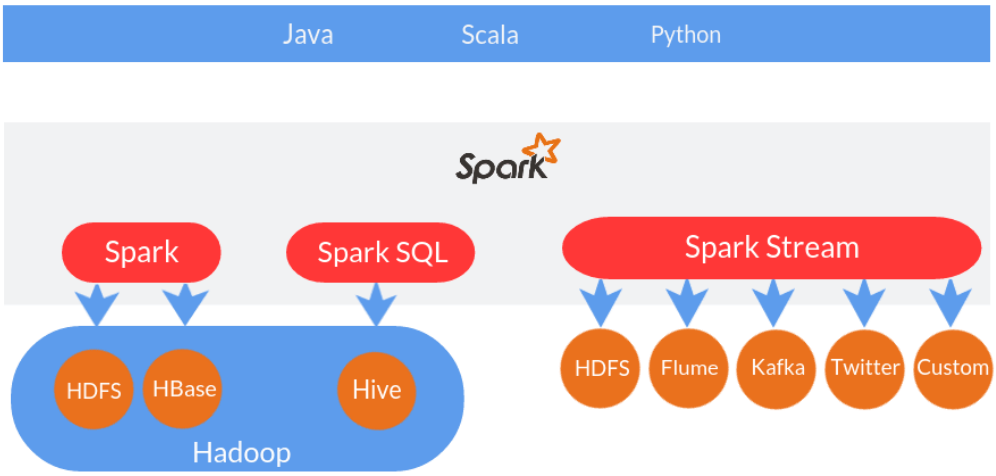


spark-cluster

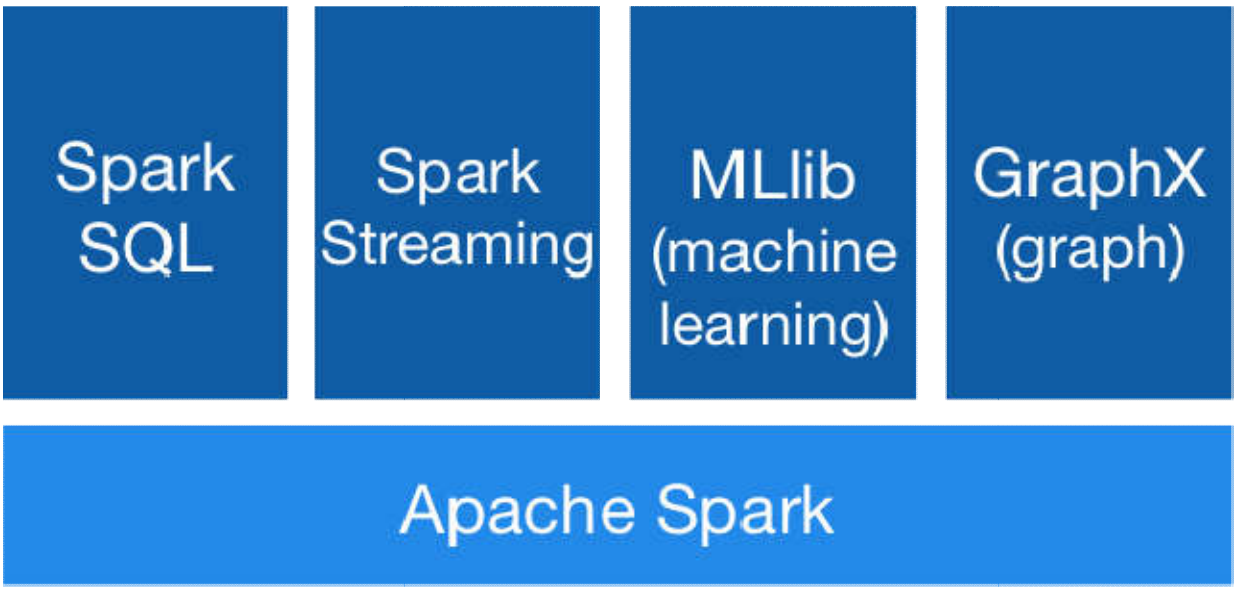
Apache Spark birden fazla makine üzerinde büyük verileri analiz eder.Bu sayede tek bir makine üzerinde işlem yapmak yerine birden fazla makine kullanarak daha hızlı sonuç alabiliriz.Eğer yaptığımız işlemlerden yeterli performans alamıyorsak makine sayısını rahatlıkla artırabiliriz(ölçeklenebilirlik)



spark-analiz

Büyük verileri saklamak için Hadoop HDFS gibi bir yapı yoktur. Apache Spark yalnızca verileri işlemek için geliştirilmiştir.Resimde de görüldüğü gibi Apache Spark ; HDFS,HBase,Kafka,Flume .. gibi büyük verileri saklayan ve büyük verileri transfer eden mesajlaşma sistemlerine ihtiyaç duyar

---



spark-mimarisi

Spark Mimarisi genel olarak Spark SQL, Spark Streaming, MLib, Graphx, Spark Core bileşenlerinden oluşur

Spark Core genel olarak şu işlemleri sağlar

- Memory yönetimi
- Hata durumlarını yönetme
- HDFS,veritabanı .. gibi verilerin okunduğu bileşenlerle bağlantı
- RDD yapısı (Verileri işlememizi sağlayan metodlar bulunur *count,map,foreach..*)
- Slave makineleri üzerindeki task'ların yönetilmesi

Spark SQL ise veriler üzerinde SQL tabanlı işlemler yapmamız sağlar. Parquet, Hive,JSON,İlişkisel veritabanı üzerinde bulunan verilerden SQL tabanlı sorgulamalar yapabiliriz

Bu yapıda Spark SQL,Dataset ve DataFrame kullanabiliriz

Örnekler ;

```
val dataFrame =
    sqlCtx.sql("select avg(latency) from log")
```

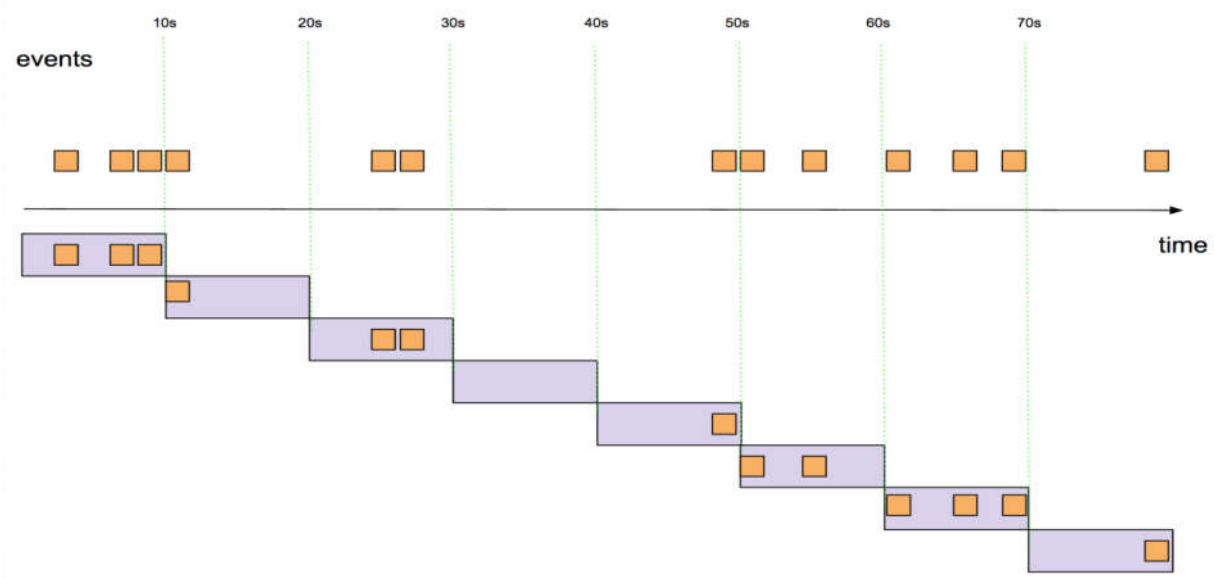
spark-dataframe

```
employeeDF.filter(func.col('department') == 'Business').groupBy('email').count().show()
```

```
+--------------------+-----+
|               email|count|
+--------------------+-----+
|george.schmidt@co...|    1|
|helga.musterfrau@...|    1|
+--------------------+-----+
```

dataframe

Spark Streaming ile verileri anlık olarak analiz edebiliriz.Genel yapı olarak anlık alınan veriler belirli zaman periyotlarına ayrılır(window).Alttaki örnekte 10'ar saniyelik periyotlarda veriler analiz edilebilir.Mesela 10 saniyelik periyotlarda kaç tane **event** geldiği hesaplanabilir.



spark-streaming

MLib kütüphanesi ile makine öğrenimi(machine learning) uygulamaları yazabiliriz. Örnek verirsek ALS kütüphanesi ile filmler hakkında puanlama yapan kullanıcıları analiz ederek, kullanıcıların bir sonraki beğenebileceği filmleri **tahmin edebiliriz**

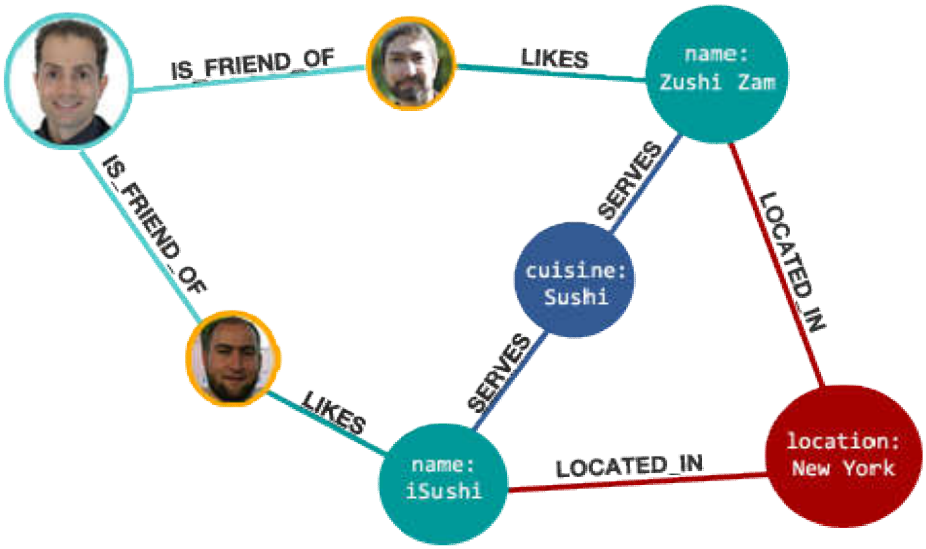
Alttaki örnekte Spark ile ALS kütüphanesinin ne kadar kolay kullanılabileceğini görebiliriz

```
ALS als = new ALS();
MatrixFactorizationModel model =
als.setRank(20).setIterations(10).run(trainingRatingRDD);

Rating[] recommendedsFor4169 = model.recommendProducts(4169, 5);
System.out.println("Recommendations for 4169");

for (Rating ratings : recommendedsFor4169) {
    System.out.println("Product id : " + ratings.product() + "-- Rating : " +
ratings.rating());
}
```

GraphX kütüphanesi ile graph tabanlı hesaplamalar yapabiliriz.Genellikle sosyal ağlar için uygulamalar yazılabilir



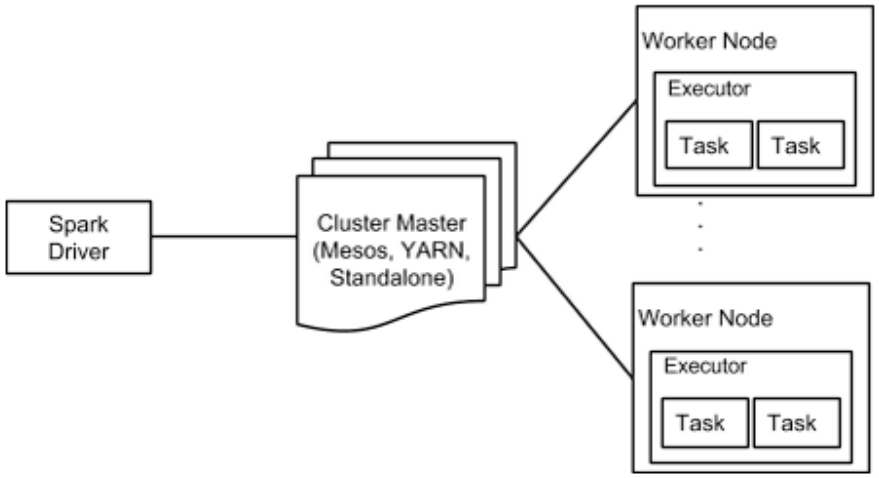
nosql-graph

Cluster Yönetimi

Apache Spark uygulamaları 3 farklı cluster yönetiminde çalışır

- Standalone
- Mesos
- Yarn

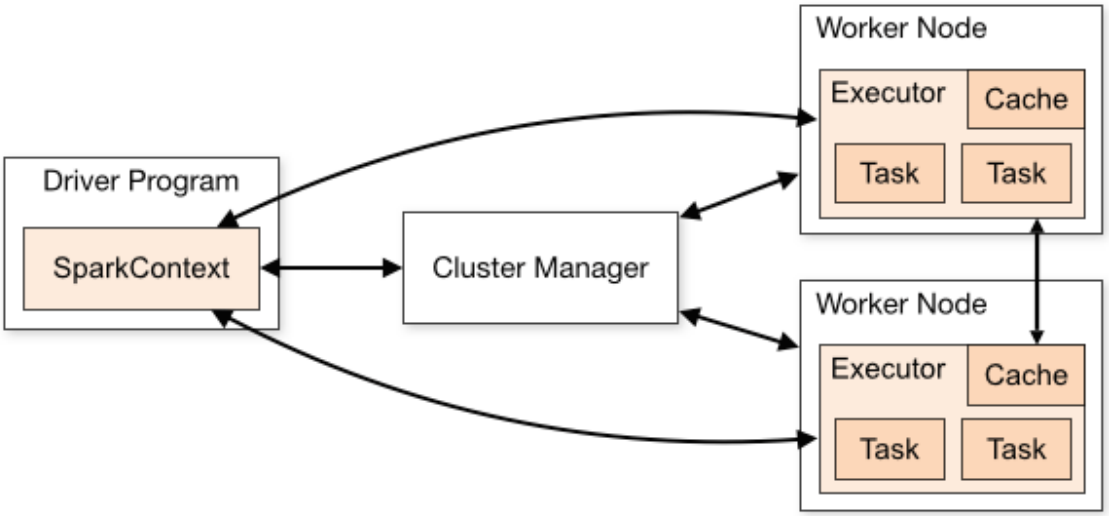
Spark ilk yüklendiğinde Standalone mod ile gelir. Başlangıç için Standalone mod uygundur.**Mesos** ile kaynakların dinamik olarak artıp azaltılması sağlanabilir.Mesela Standalone mod ile bir uygulamaya 2 GB Ram verdiğimizde o uygulamada ihtiyaç olmasa bile 2 GB kaynak kullanır.Fakat **mesos** ihtiyaç durumunda bu kaynağı artırıp azaltır. Bununla beraber Hadoop Cluster üzerinde Spark da kullanmak istiyorsak YARN tercih edilebilir



cluster-manager

Apache Spark ; mimari olarak master-worker çalışma mantığı ile verileri işler.Master genel olarak yapılması gereken işleri Worker makinelerine dağıtır.Worker makineleri ise Master üzerinden aldığı işlemleri gerçekleştirir ve sonuçları Master makinesine iletir

- *Driver Program : Uygulamanın başlatıldığı ana program (main() metodu)*
- *SparkContext : Tüm işlemler SparkContext tarafından koordine edilir.Çalışacak uygulamalar SparkContext tarafından worker makinelerine iletilir*
- *Her uygulama için ayrı bir Executor başlatılır.(Ayrı bir JVM process düşünebiliriz) Buna bağlı olarak her uygulamanın kendine ayrılmış kaynakları vardır(Ram,Core)*
- *Task : Bir uygulamayı çalıştırırken yaptığımız işlemlere göre iş bölümleri yada fonksiyonlar vardır.Bunların herbiri task olarak düşünülebilir*



spark-mimari