



EURO

Büyük Veri ile Makine Öğrenmesi Nasıl Yapılır

Ahmet Demirelli

Sabancı Üniversitesi

Ajanda

- Büyük Veri Nedir
- Nerede Saklanır
 - Hadoop
- Nasıl İşlenir / Makine Öğrenmesi Yapılır
 - Apache Spark / Spark ML
- Demo

Büyük Veri Nedir ?

- Lütfen bu soruyu sormaktan ve
- Zorlama tanımlar yapmaktan vazgeçelim
(3V, Volume, Velocity, Variety, 5V ... vs)
- Verimiz büyük veridir eğer ;
 - Saklayamıyorsak (HDD)
 - İşleyemiyorsak (RAM)
 - İşlesek bile mantıklı bir zaman süresinde sonuç alamıyorsak (CPU)
- Çözüm; Dağıtık Saklama ve Dağıtık İşleme
(Distributed Storage, Distributed Processing)

Büyük Veri problemleri

- Geleneksel veri işleme
 - Merkezi bir veri tabanı
 - Veri işlenmek istediğimizde SQL (select)
 - Veri başka bir sunucu veya merkezi veri tabanının üzerinde işlenir
 - Bu yöntemin getirdiği problemler →

Büyük Veri problemleri

- Sınırlı veri işlenebilir
- Merkezi veritabanının yeteneği ile sınırlıyız
- İşleyeceğimiz veri sunucunun hafızasına sığmak zorunda
- Genişletilmesi çok zor
- Çözüm →

Büyük Veri problemleri çözüm ?



- Daha hızlı CPU ve daha fazla RAM
- Dağıtık Mimariler
 - Senkronizasyon
 - Sunucular arası sürekli veri değişimi
 - Bir makine işini yapamazsa veya çökerse ?
 - Kurulması,kullanılması ve kod yazması çok zor

Nerede Saklanır ?

Açık kaynak kodlu dağıtık veri saklama platformları

- CephFS
- MooseFS
- GlusterFS
- Hadoop Distributed FS (HDFS) ****
- DRDB

Detaylı bilgi için ;

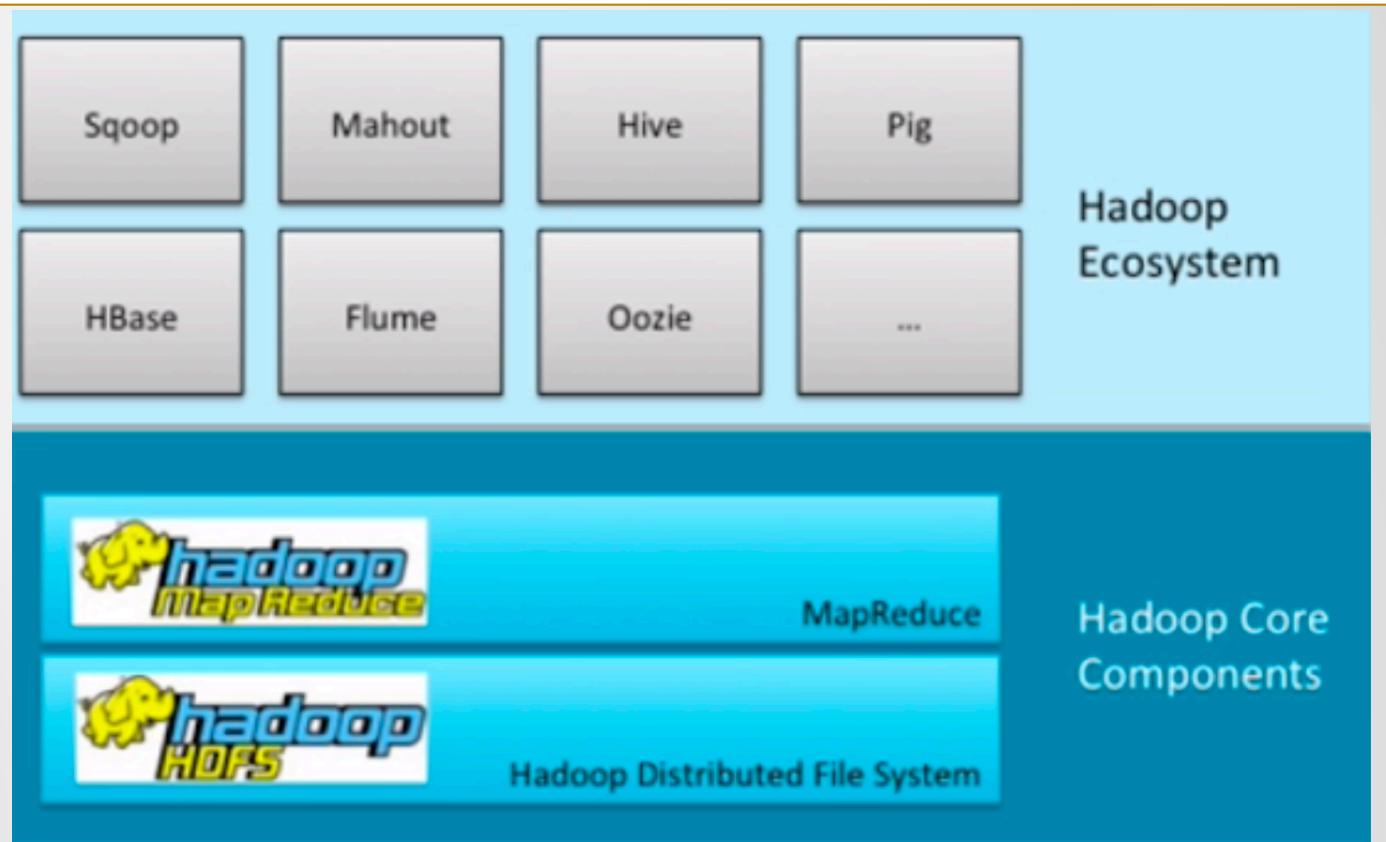
<https://computingforgeeks.com/ceph-vs-glusterfs-vs-moosefs-vs-hdfs-vs-drbd/>

Hadoop

- Google firmasının yayınlamış olduğu iki makaleden ortaya çıkmıştır
 - Google File System(2003)
 - <http://static.googleusercontent.com/media/research.google.com/en//archive/gfs-sosp2003.pdf>
 - MapReduce(2004)
 - <http://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf>
- Apache Software Foundation (<https://hadoop.apache.org>) tarafından açık kaynak kodlu olarak yazılmıştır. (2006)
 - GFS → HDFS
 - MapReduce → MapReduce
- Sektör tarafından Kabul görmüş ve günümüzde yaygın olarak kullanılmaktadır

Hadoop DFS

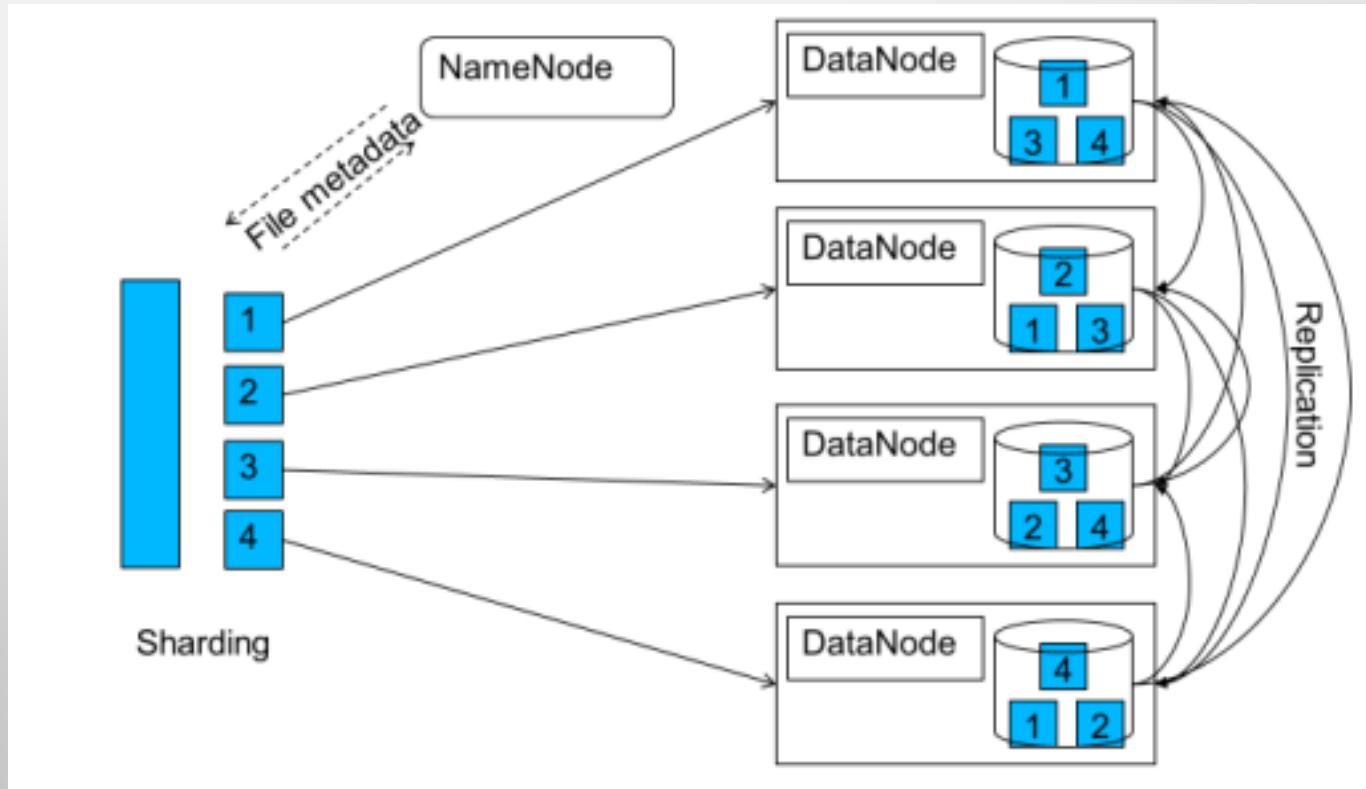
- HDFS - Hadoop sisteminin en önemli parçasıdır
- April 1, 2006 yılında ilk sürümü yanındandı
- GFS ve MapReduce makalelerinin açık kaynak kodlu olarak geliştirildi



HDFS

- HDFS - Hadoop Distributed File System
 - Gerçek bir dosya sistemi (file system) değil
 - GFS örnek alınarak yazılmış - Apache Software Foundation
 - Açık kaynak yazılım
 - Var olan başka bir dosya sistemi ve işletim sistemi (OS) üzerine kurulur
 - Verileri dağıtık tutar (Varsayılan 128 MB bloklar)
 - Verilerin kopyalarını tutma imkanı sunar (Replication)

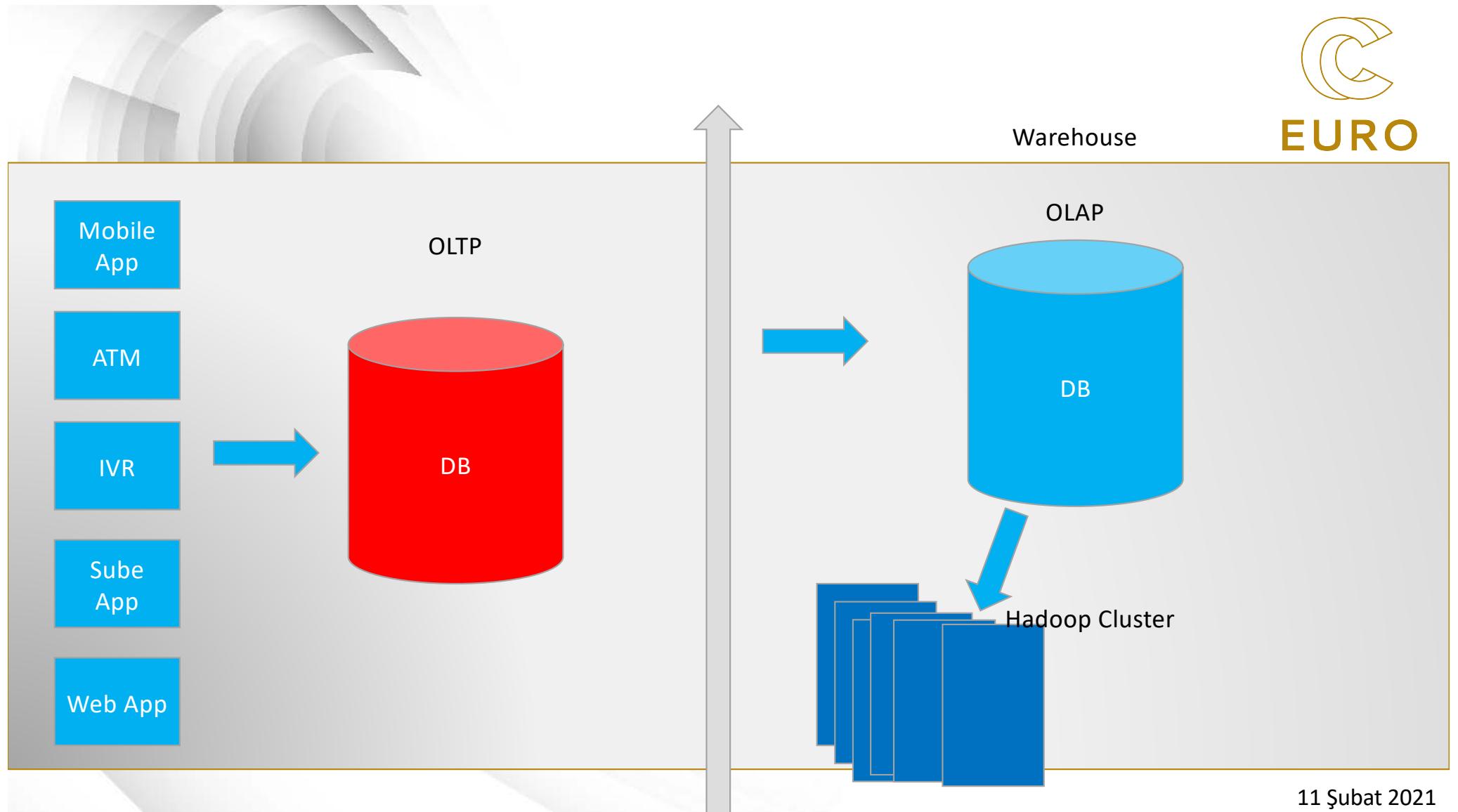
HDFS Dosya dağıtımımı

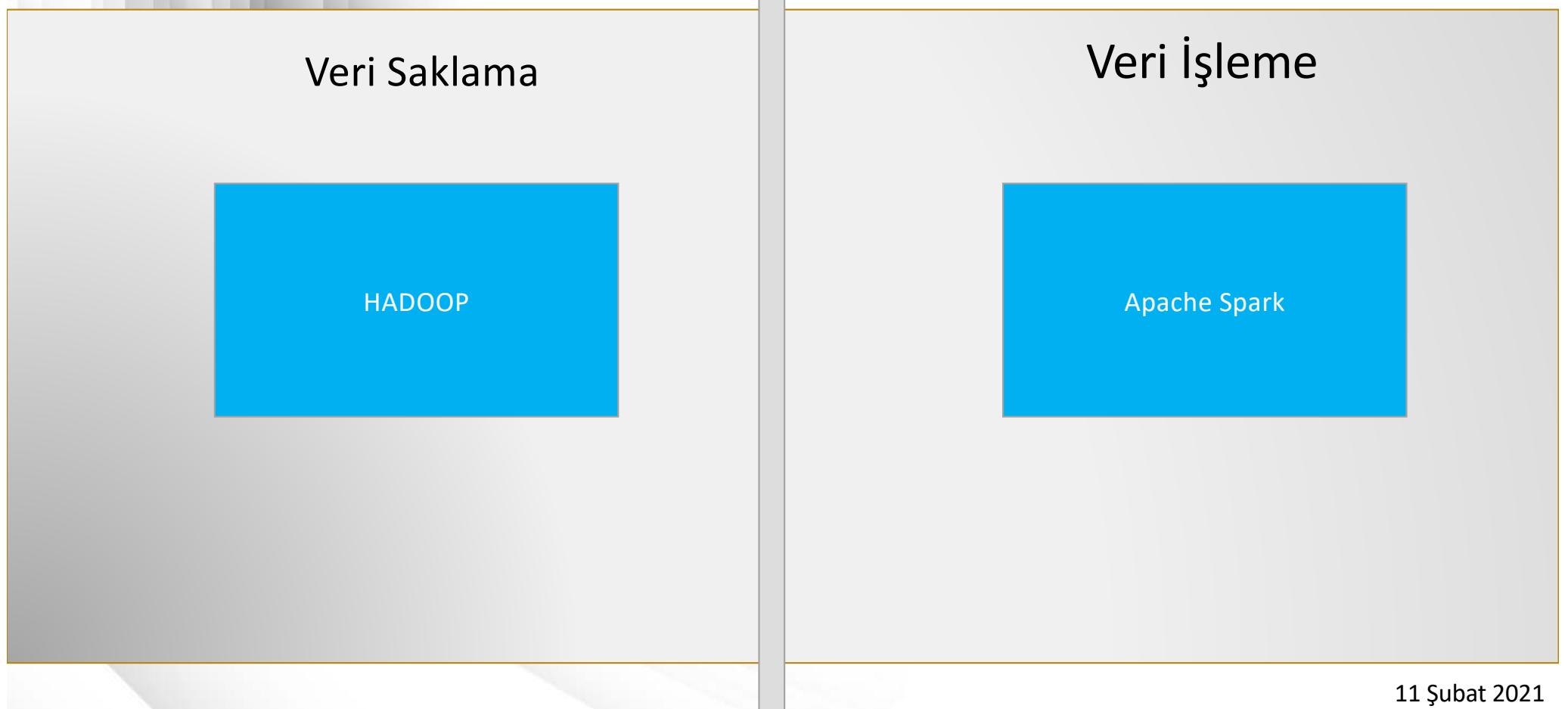


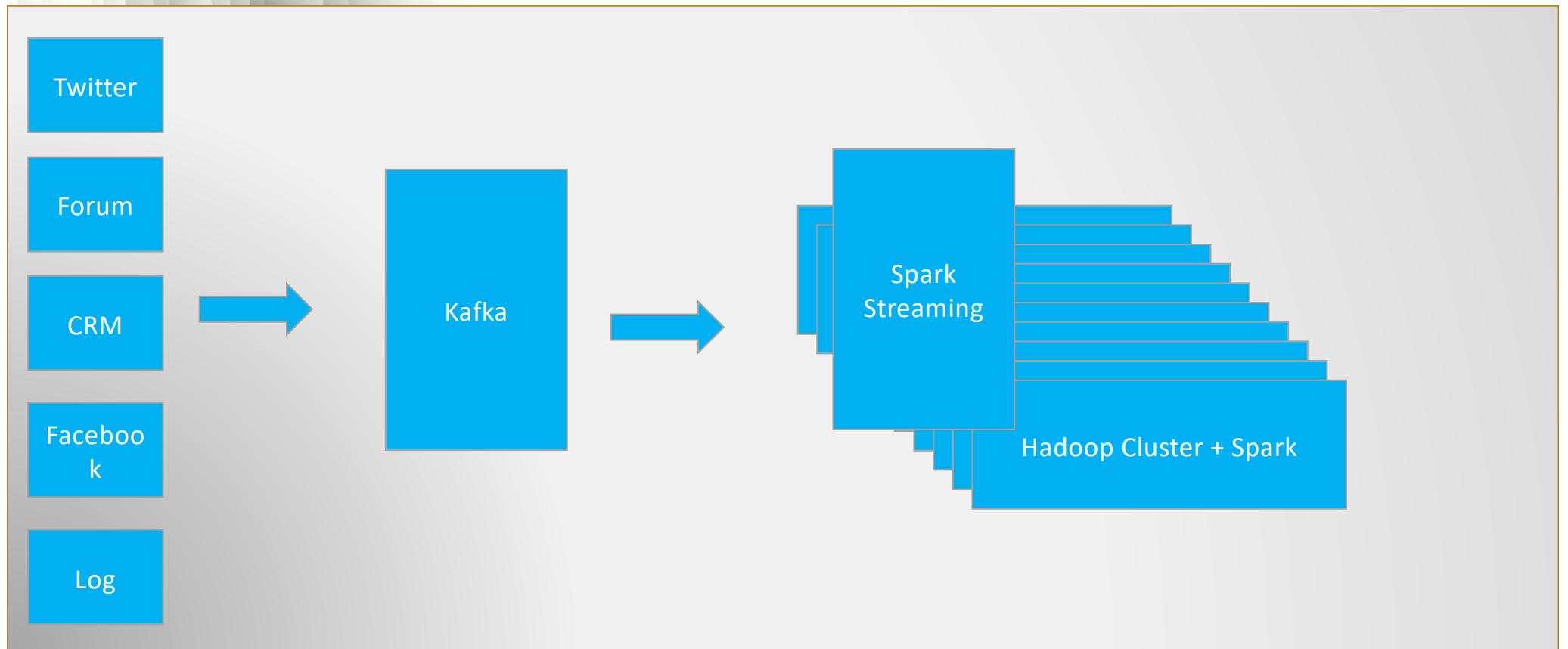
<https://cvw.cac.cornell.edu/mapreduce/images/hdfs.png>

Hadoop nasıl kurulur ?

- Öğrenme amaçlı kurulum →
<https://hub.docker.com/r/cloudera/quickstart/>
- Gerçek kullanım amaçlı;
 - Zor kurulum → <http://hadoop.apache.org> sitesinden manuel kurulum
 - Daha Kolay Kurulum →
https://docs.cloudera.com/documentation/enterprise/6/6.1/topics/install_cm_cd.html
 - Kurulum bile sayılmaz. ☺ → Google Cloud (DataProc)
<https://cloud.google.com/dataproc>







Büyük Veri ve YBH (HPC)



- Spark 3.0 ile GPU desteği eklendi
- Apache Spark ile yazılmış olan Veri bilimi uygulamalarımız GPU üzerinde daha hızlı çalıştırabiliyoruz.
- Bunun için özel kod yazmamız gerekmiyor hatta kod değişikliğine bile gerek yok
- Daha fazla bilgi için:
 - <https://www.nvidia.com/en-us/deep-learning-ai/solutions/data-science/apache-spark-3>
 - <https://developer.nvidia.com/blog/accelerating-apache-spark-3-0-with-gpus-and-rapids/>

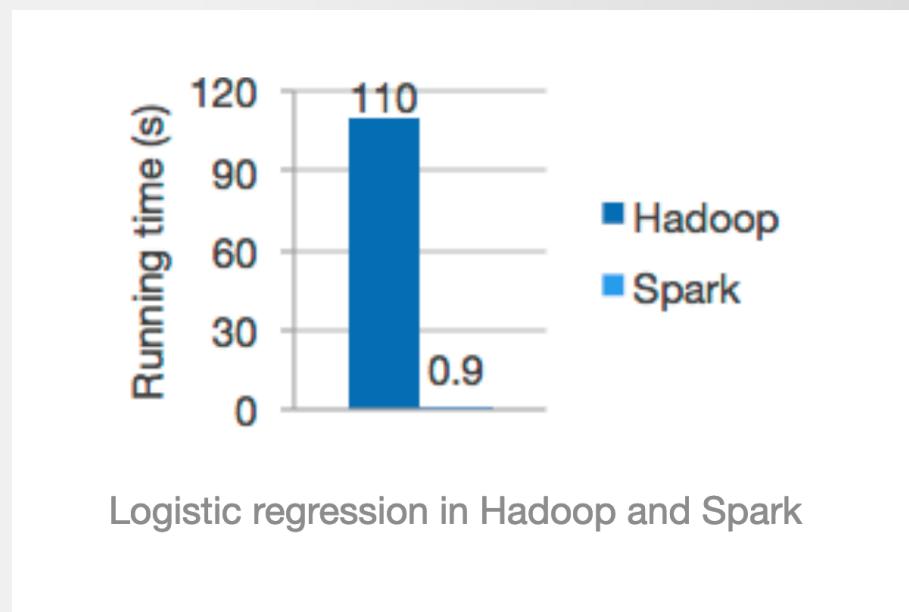
Nasıl İşlenir ?

- Dağıtık veri saklayan bir platformda işlemek en kolayıdır
- Hadoop bunu kolaylaştırır
 - Veriyi dağıtık bir şekilde tutar - HDFS
 - Verinin yerini değiştirmeden!! dağıtık bir şekilde işleyebilir – MAPREDUCE ☺
- MapReduce sektör tarafından sevilmedi ve kabul görmedi
- May 26, 2014 yılında Apache Spark ortaya çıktı
 - MapReduce tan daha hızlıydı
 - Kod daha kolay yazılıyordu
 - Java, Scala ve Python desteği var

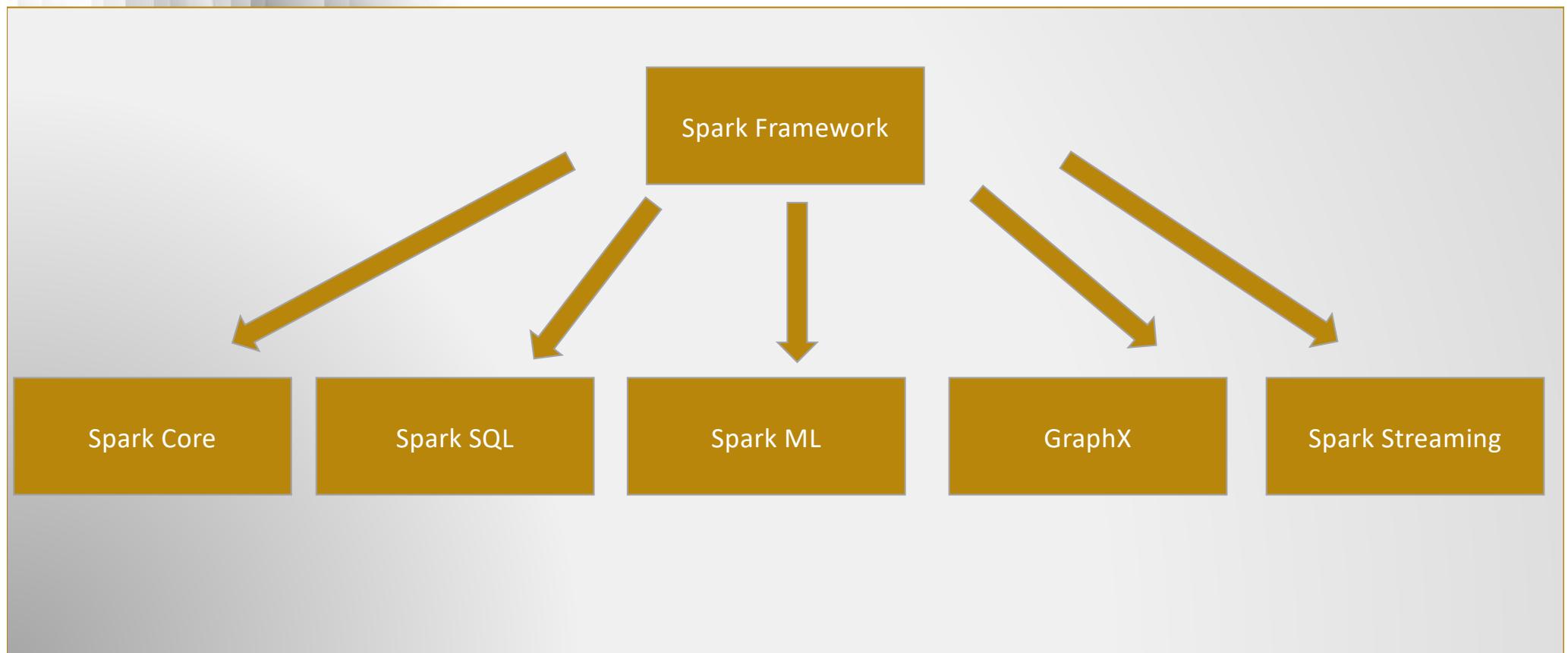
Nasıl İşlenir ?

Apache Spark

1. Tek makinada HDFS olmadan çalışabilir
2. HDFS üzerinde çalışabilir
3. MapReduce tan 100x hızlı çalışır
4. Dağıtık ML algoritmaları vardır
5. Büyük veri işlemeye vazgeçilmez olmuştur



Apache Spark



Büyük Veri İşleme

HADOOP

+

Apache Spark

Eğer sadece ETL, raporlama yapacaksak sadece Hadoop yeterli olabilir çünkü HiveQL kullanarak SQL ile yapabildiğimiz herşeyi yapabiliyoruz

SQL ile makine öğrenmesi yapamayacağımız için mutlaka Apache Spark kullanmamız gerekiyor

Büyük Veri ve YBH (HPC)



- Spark 3.0 ile GPU desteği eklendi
- Apache Spark ile yazılmış olan Veri bilimi uygulamalarımız GPU üzerinde daha hızlı çalıştırabiliyoruz.
- Bunun için özel kod yazmamız gerekmiyor hatta kod değişikliğine bile gerek yok
- Daha fazla bilgi için:
 - <https://www.nvidia.com/en-us/deep-learning-ai/solutions/data-science/apache-spark-3>
 - <https://developer.nvidia.com/blog/accelerating-apache-spark-3-0-with-gpus-and-rapids/>

Uygulama Örnekleri



DEMO

Uyarı : Colab ortamında pyspark çalıştırılabilmek için her sayfanın başında aşağıdaki kodları bir defa çalıştırmak gerekiyor

```
!apt-get update
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
!update-alternatives --set java /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
!java -version
!pip install pyspark
```

Uygulama Örnekleri



Iris Versicolor

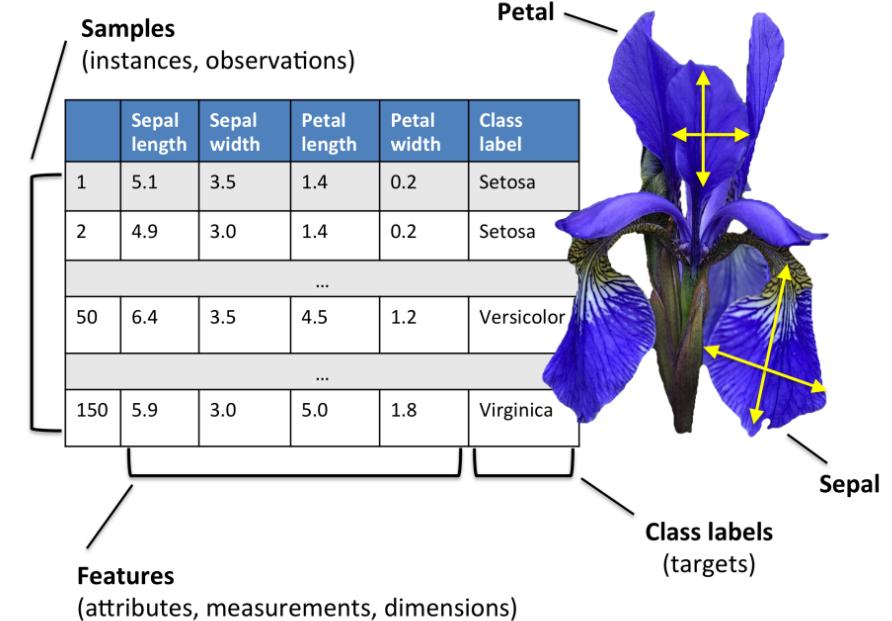


Iris Setosa



Iris Virginica

Çok gizli bir veri seti kullanacağımız!
 Veri setinin ismi : Iris-Dataset
 Lütfen kimse ile paylaşmayın





Teşekkürler



EuroHPC
Joint Undertaking

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 951732. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, United Kingdom, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Switzerland, Turkey, Republic of North Macedonia, Iceland, Montenegro