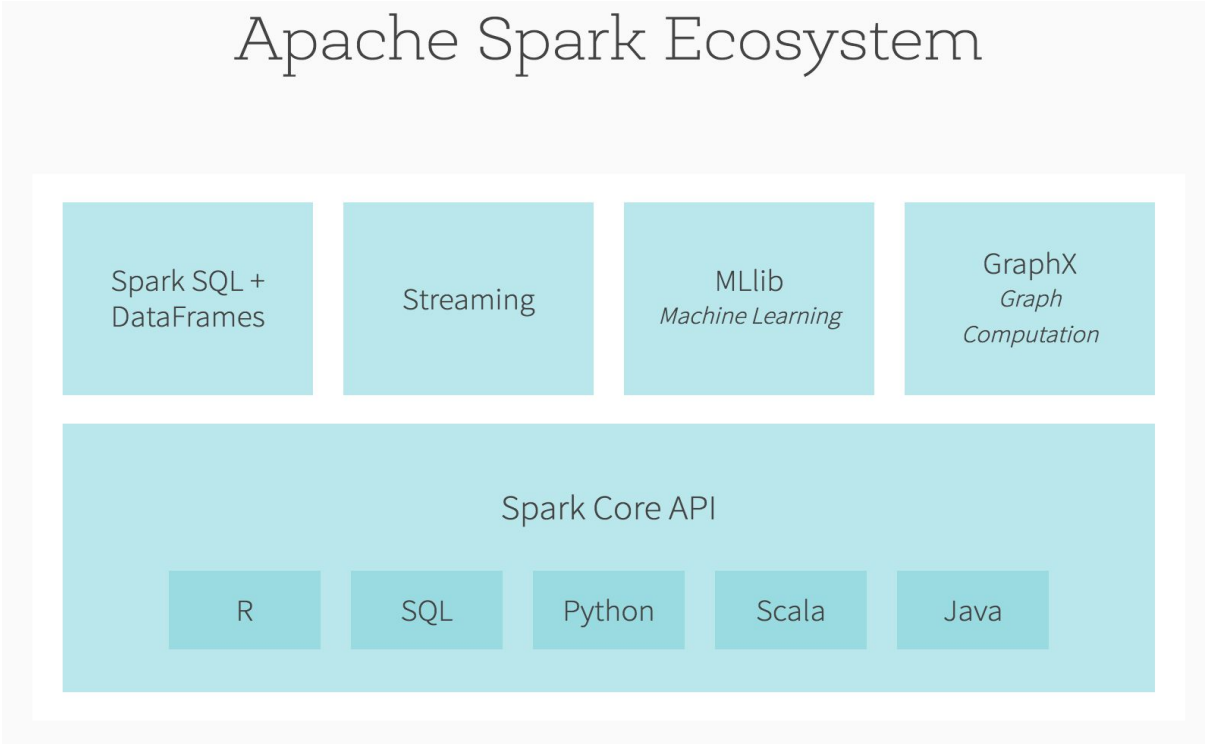
SPARK with DataBricks

What is Spark?

- “Apache Spark is a fast and general-purpose cluster computing system”
- Wikipedia: **Apache Spark** is an open source cluster computing framework. Originally developed at the University of California, Berkeley's AMPLab, the Spark codebase was later donated to the Apache Software Foundation, which has maintained it since. Spark provides an interface for programming entire clusters with implicit data parallelism and fault-tolerance.
- High level API support for Java, Scala, Python and R
- Spark: Cluster Computing with Working Sets -- *Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, Ion Stoica*

And does much more …



Why use Spark?

- 100 terabytes of data stored on solid-state drives in just 23 minutes (Daytona Gray Sort benchmarking challenge)
    - ... that is 100 thousand Gigabytes of data!!
- Spark is promising to speed up application development by 10-100x, make applications more portable and extensible, and make the actual application run 100x faster.
- Interactive queries, Iterative computational tasks such as Machine Learning

Comparisons with MapReduce

- Everything is in memory (no shuffling of data in and out of disk)
- Lesser boilerplate code
- Lazy computation (Transformation and Action … explained later)
- Simply better programming abstractions
  - Don’t have to model every task into a MapReduce paradigm
- Hyper-optimize applications
- outperforms Hadoop by 10x in iterative machine learning jobs
- … but both provide fault-tolerance and load-balancing

How Spark does it? …. Under the hood!

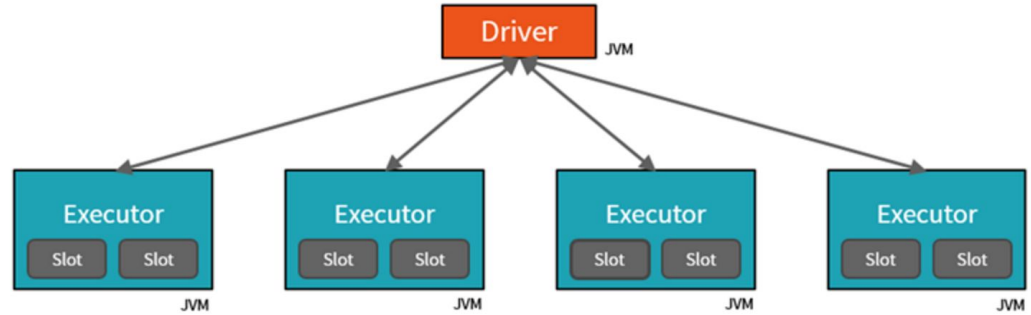
Master Slave architecture

Driver instructs Workers to pull

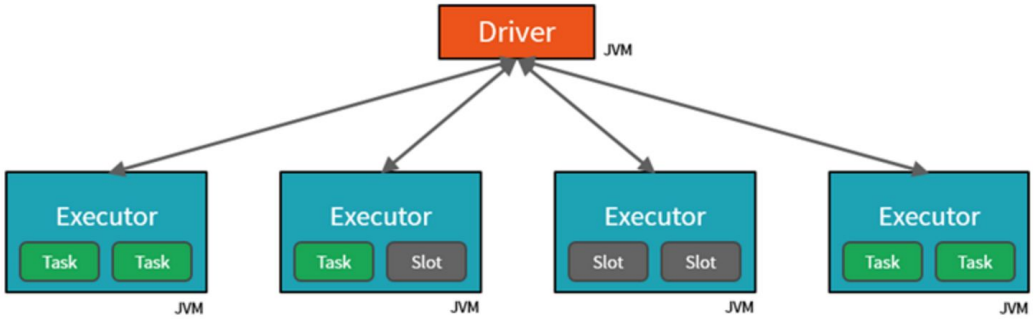
data from HDFS/S3/Redshift etc

Basic Operations:

Broadcast; Take/Collect; Shuffle



The Driver sends Tasks to the empty slots on the Executors when work has to be done:



Storing data the right way ...

The Dataset: The Dataset is Apache Spark's newest distributed collection and can be considering a combination of DataFrames and RDDs. It provides the typed interface that is available in RDDs while providing a lot of conveniences of DataFrames. It will be the core abstraction going forward.

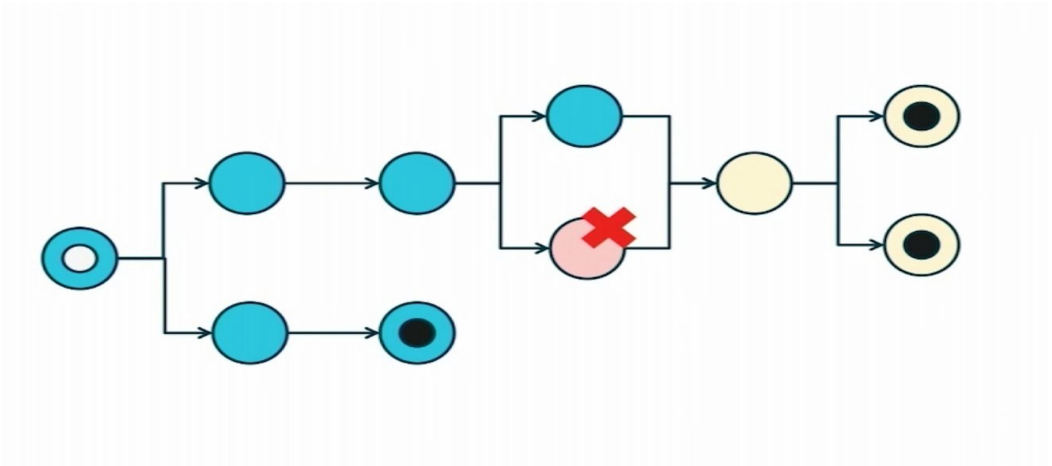
The DataFrame: The DataFrame is collection of distributed `Row` types. These provide a flexible interface and are similar in concept to the DataFrames you may be familiar with in python (pandas) as well as in the R language.

The RDD (Resilient Distributed Dataset): Apache Spark's first abstraction was the RDD or Resilient Distributed Dataset. Essentially it is an interface to a sequence of data objects that consist of one or more types that are located across a variety of machines in a cluster. RDD's can be created in a variety of ways and are the "lowest level" API available to the user. While this is the original data structure made available, new users should focus on Datasets as those will be supersets of the current RDD functionality.

DAGs ...

Breakdown Jobs into Tasks

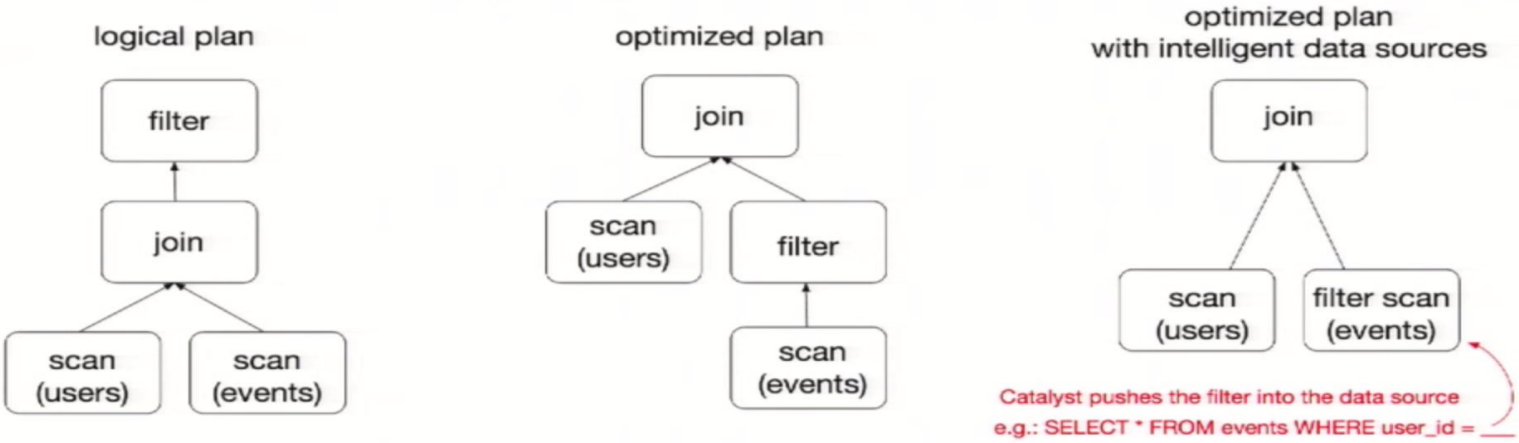
Just track ancestor in case of failure and rebuild since either RDD/DataFrames are immutable and Cacheable (lineage based fault tolerance)



But also DAG Optimizations …

Example Optimization

```
users.join(events, users("id") === events("uid"))
     .filter(events("date") > "2015-01-01")
```



DataBricks!

- Fixed vs Dynamic Allocation of resources
- Running spark:
    - Locally: spark-shell and submit jobs on cluster after local testing
    - Driver program is the Notebook (automatically deploys to clusters) vs Submitting applications to the cluster
- Automatically handles general distributed computation challenges: disk failures, latency on network etc

Live Session!

Spark resources!

- Ted Malaska Oscan talk: https://www.youtube.com/watch?v=x8xXXqvhZq8
- Brian Clapper at Ne Scala 2016: https://www.youtube.com/watch?v=pZQsDloGB4w
- http://cacm.acm.org/magazines/2016/11/209116-apache-spark/fulltext
- UC Berkley Spark courses on Edx: https://courses.edx.org/courses/course-v1:BerkeleyX+CS105x+1T2016/info
- Comparing Hadoop and Spark: https://www.mapr.com/blog/5-minute-guide-understanding-significance-apache-spark
- Catalyst Optimizer: https://databricks.com/blog/2015/04/13/deep-dive-into-spark-sqls-catalyst-optimizer.html
- MapR blog post about Spark: https://www.mapr.com/ebooks/spark/01-what-is-apache-spark.html
- Project Tungsten https://databricks.com/blog/2015/04/28/project-tungsten-bringing-spark-closer-to-bare-metal.html
- More technical blogs: Databricks, Cloudera, IBM
- DataBricks guides:
    - Introduction to Apache Spark on DataBricks
    - DataBricks for Data Scientists
- Live session workbook

Thank You!
Questions?