Spark

• “Apache Spark is a fast and general engine for large-scale data processing” - Spark Website
• Provides interactive response times to large amounts of data
• Written in Scala, but can also be used from Java python Python and R
• Fault tolerant
Key concepts

• RDD (Resilient Distributed Dataset)
• Transformations (on RDDs)
• Broadcast variables
• Accumulator variables
• Tasks
• Executors
Resilient Distributed Dataset

RDD
RDD - Overview

- Immutable representation of a dataset
- Deterministic instantiation and transformation
- Distributed (partitions)
- Instantiated by
  - transforming another RDD
  - from an input source, like a file on HDFS
- Computation close to the data
- Fault tolerant (based on lineage)
RDD - Persistence

• Caching is handled by the developer
• An RDD can be cached in memory by calling the cache() method
• The persist() method lets you persist an RDD to
  – Memory
  – Memory and disk
  – Disk only
• Variable methods of serialization and replication
# RDD – Transformations and actions

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>map(f: T =&gt; U)</td>
<td>RDD[T] =&gt; RDD[U]</td>
</tr>
<tr>
<td>filter(f: T =&gt; Bool)</td>
<td>RDD[T] =&gt; RDD[T]</td>
</tr>
<tr>
<td>groupByKey()</td>
<td>RDD[(K, V)] =&gt; RDD[(K, Seq[V])]</td>
</tr>
<tr>
<td>join()</td>
<td>(RDD[K,V],RDD[K,W]) =&gt; RDD[(K, (V, W))]</td>
</tr>
<tr>
<td>partitionByKey(p: Partitioner[K])</td>
<td>RDD[(K, V)] =&gt; RDD[(K, V)]</td>
</tr>
<tr>
<td>count()</td>
<td>RDD[T] =&gt; Long</td>
</tr>
<tr>
<td>collect()</td>
<td>RDD[T] =&gt; Seq[T]</td>
</tr>
<tr>
<td>reduce(f: (T, T) =&gt; T)</td>
<td>RDD[T] =&gt; T</td>
</tr>
<tr>
<td>save(path: String)</td>
<td>Outputs RDD to a storage system, e.g., HDFS, Amazon S3</td>
</tr>
</tbody>
</table>
RDD - Example

```scala
val lines = spark.textFile("hdfs://...")
val errors = lines.filter(_.startsWith("ERROR"))
errors.persist()

// Returns Seq[String]
errors.filter(_.contains("HDFS"))
  .map(_.split('\t')(3))
  .collect()
```

(taken from Spark paper)
SHARED VARIABLES
Broadcast variable

• Immutable variable that is broadcasted to all nodes
• Useful for pre-computed tables, etc.
Accumulator variable

• A variable that supports an “add” operation
• Useful for implementing counters
EXECUTION
Tasks

- A task is the unit of execution in Spark
- Each partition of an RDD is mapped to a task
- Each task is executed by an executor
Spark ecosystem

- Spark Streaming: Live stream processing with Spark
- Shark SQL: SQL interface to RDDs (compatible with Apache Hive)
- MLib: Machine learning library built on top of Spark
- GraphX: Graph analysis on Spark
Resources

- https://spark.apache.org/research.html
QUESTIONS