Spark Internals

CS259 Computer Science Classics
- Aditya Rao
- Vikas Tikoo
Spark

- In-memory cluster processing engine
- Relevant for a large set of big data operations with coarse-grained updates: iterative, streaming, interactive query and batch
- Developed at AMPlab UC, Berkeley
- 20x faster than Hadoop for iterative applications
Spark Runtime
Life a Spark Job:

RDD Objects
- Build operator DAG
- rdd1.join(rdd2)
  .groupBy(...)
  .filter(...)

DAGScheduler
- Split graph into stages of tasks
- Submit each stage as ready
- Agnostic to operators!

TaskScheduler
- Launch tasks via cluster manager
- Retry failed or straggling tasks
- Doesn’t know about stages

Worker
- Execute tasks
- Store and serve blocks
RDD: central abstraction in Spark

- Resilient Distributed Datasets
- read-only, partitioned collection of records
- formed only by deterministic operations
- materialization not necessary (through lineage graphs)
- Users can control: persistence, partitioning
RDD Interface:

1. partitions() - Set of partitions ("splits")
2. dependencies() - List of dependencies on parent RDDs
3. iterator(p, parentIters) - Function to compute a partition given parents
4. preferredLocations(p) - Optional preferred locations
5. partitioner() - Optional partitioning info (Partitioner)
Eg: Hadoop RDD

1. partitions - one per HDFS block
2. dependencies - none
3. compute(partition) - read corresponding block
4. preferredLocation(part) - HDFS block location
5. partitioner - none
Narrow vs Wide Dependencies

Figure 4: Examples of narrow and wide dependencies. Each box is an RDD, with partitions shown as shaded rectangles.
Scheduler Optimizations

- Pipelining narrow operations within a stage
- Picks join algorithms based on parititioning (minimize shuffles)
- LRU used for cache management
Topics not covered here but in final paper:

- Delay scheduling
- Shuffle algorithms
- Spark Streaming
References

- Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing --- Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, Ion Stoica
- Discretized Streams: An Efficient and Fault-Tolerant Model for Stream Processing on Large Clusters Matei Zaharia, Tathagata Das, Haoyuan Li, Scott Shenker, Ion Stoica
- http://feiskyer.github.io/pages/about/spark.html
- http://spark.apache.org