Transcript name: MapReduce – Part 6 – Scheduling & Task Execution

English

So far we have looked at how Hadoop executes a single job as if it is the only job on the system. But it would be unfortunate if all of your valuable data could only be queried by one user at a time. Hadoop schedules jobs using one of three schedulers.

The simplest is the default FIFO scheduler.

It lets users submit jobs while other jobs are running, but queues these jobs so that only one of them is running at a time.

The fair scheduler is more sophisticated.

It lets multiple users compete over cluster resources and tries to give every user an equal share. It also supports guaranteed minimum capacities.

The capacity scheduler takes a different approach.

From each user's perspective, it appears that the they have the cluster to themselves with FIFO scheduling, but users are actually sharing the resources.

Hadoop offers some configuration options for speeding up the execution of your map and reduce tasks under certain conditions.

One such option is speculative execution.

When a task takes a long time to run, Hadoop detects this and launches a second copy of your task on a different node. Because the tasks are designed to be self-contained and independent, starting a second copy does not affect the final answer. Whichever copy of the task finishes first has its output go to the next phase. The other task's redundant output is discarded.

Another option for improving performance is to reuse the Java Virtual Machine.

The default is to put each task in its own JVM for isolation purposes, but starting up a JVM can be relatively expensive when jobs are short, so you have the option to reuse the same JVM from one task to the next.

This concludes this lesson on Hadoop MapReduce. Thank you for watching.