

NextGenMapReduce

This wiki tracks development of [Next Generation Apache Hadoop MapReduce](#) (jira: MAPREDUCE-279).

Overview

The fundamental idea of the re-architecture is to divide the two major functions of the **JobTracker**, resource management and job scheduling/monitoring, into separate components. The new **ResourceManager** manages the global assignment of compute resources to applications and the per-application **ApplicationMaster** manages the application's scheduling and coordination. An application is either a single job in the classic [MapReduce](#) jobs or a DAG of such jobs. The [ResourceManager](#) and per-machine [NodeManager](#) server, which manages the user processes on that machine, form the computation fabric. The per-application [ApplicationMaster](#) is, in effect, a framework specific library and is tasked with negotiating resources from the [ResourceManager](#) and working with the [NodeManager](#)(s) to execute and monitor the tasks.

The [ResourceManager](#) supports hierarchical application queues and those queues can be guaranteed a percentage of the cluster resources. It is pure scheduler in the sense that it performs no monitoring or tracking of status for the application. Also, it offers no guarantees on restarting failed tasks either due to application failure or hardware failures.

The [ResourceManager](#) performs its scheduling function based the resource requirements of the applications; each application has multiple resource request types that represent the resources required for containers. The resource requests include memory, CPU, disk, network etc. Note that this is a significant change from the current model of fixed-type slots in Hadoop [MapReduce](#), which leads to significant negative impact on cluster utilization. The [ResourceManager](#) has a scheduler policy plug-in, which is responsible for partitioning the cluster resources among various queues, applications etc. Scheduler plug-ins can be based, for e.g., on the current [CapacityScheduler](#) and [FairScheduler](#).

The [ResourceManager](#) has two main components:

- Scheduler - The core scheduler which allocates resources to applications based on the chosen policy (capacity guarantees, fairness etc.)
- [ApplicationsManager](#) - The component of the [ResourceManager](#) which is responsible for accepting job-submissions, negotiating the first container for running the appropriate [ApplicationMaster](#) and provides service for restarting the [ApplicationMaster](#) container on failure.

The [NodeManager](#) is the per-machine framework agent who is responsible for launching the applications' containers, monitoring their resource usage (cpu, memory, disk, network) and reporting the same to the Scheduler.

The per-application [ApplicationMaster](#) has the responsibility of negotiating appropriate resource containers from the Scheduler, launching tasks, tracking their status & monitoring for progress, and handling task-failures.

Source & Documentation

The source for the first-cut is available in the [MR-279](#) branch in Apache Hadoop [MapReduce](#):

```
$ svn co http://svn.apache.org/hadoop/mapreduce/branches/MR-279/
```

We are currently in the process of adding design/implementation documentation, but some links for current reference:

- <https://issues.apache.org/jira/browse/MAPREDUCE-279>
- <http://developer.yahoo.com/blogs/hadoop/posts/2011/02/mapreduce-nextgen/>
- <http://developer.yahoo.com/blogs/hadoop/posts/2011/02/mapreduce-nextgen-scheduler/>

Development Process

Everyone is welcome to contribute, we'd love that! Just be aware we'll be moving fast. Thus, you'll need to watch the branch. We plan to use more of [mapreduce-dev@](#) and [hadoop](#) wiki and less of [jira](#) to coordinate. We'll send out email statuses to allow everyone to track, maybe even use the wiki to maintain todo lists. We'll learn as it goes, make changes to the dev process as appropriate. Please shout out if you are interested in specific areas to let others know; they could be anything - development, code-reviews, build, docs, tests etc. The horses for specific courses as you contribute:

- [MapReduce ApplicationMaster](#) - Sharad & Vinod
 - Availability - Sharad
- RM - Arun & Mahadev
 - Scheduler - Arun & Mahadev
 - [ApplicationsManager](#) & Availability - Mahadev
- [NodeManager](#) - Chris & Vinod
- Security - Vinod

Please use [NextGenMapReduceTrack](#) to follow and track various pieces of on-going development on Next Generation [MapReduce](#).

We are tracking testing at [NextGenMapReduceDevTesting](#).

Writing Applications for NextGen MapReduce i.e. YARN

Take a look at [PoweredByYarn](#) to see applications being developed for YARN.

See [WritingYarnApps](#) on more information on how to write one.