DS Design

The doc explains the framework and applications built around it for the search engine project. The framework written is general purpose but is too similar to Hadoop framework (Too primitive compared to hadoop framework. It doesn’t have HA options and all).

Architecture

The design principle of the framework is master slave. There are basically three kind of applications which run in the framework

1. Master
2. Fire-up
3. Slave Applications

Here are some of the common terminologies used in the document.

1. **Node:** Node is the logical representation of a machine. It has a node manager (Fireup) running on it which allows the master application to create the processes on this node. You can create multiple nodes in the single physical machine.
2. **Master:** Master is an application running on a node in the domain/cluster and provides the following functionalities. You can extend the functionalities of the master by extending the *MasterProcess* class.
   1. Creation of the processes
   2. Managing of the processes
   3. Killing the processes
3. **Slaves:** These are the actual processes which are of the use. The framework allows you to write these slave applications by extending the *SlaveProcess* class.

Node management

The nodes are managed by the node-manager process which allows the following

1. Creation of a process
2. Log management
3. Killing the process

Creating a node.

Below are the steps to follow to create a node

1. Start a fireup instance (node manager)
2. Register it to master. (Using master’s IP and IPC port)

Below is the sequence diagram of the node creation.

[node creation]

Registering a node.

1. Master creates a ticket for the node and assigns it to the node.
2. Node uses this ticket each further communication. The processes created in the node must inherit the same the ticket.
3. Master also sends the maximum JAR version of the applications it has.
4. Master uses the given scheduler to schedule the jobs in the node.

Process creation.

1. The master sends the process creation arguments such as jar name and command line arguments to node manager.
2. The node manager checks whether the JAR specified is latest and if not, it downloads it from the master using HTTP protocol.
3. The node manager creates the process with below command line arguments.
   1. Ticket (assigned by master)
   2. PID (a random)
   3. Stdout and Stderr file names. (Used for log management)
   4. Host and port of the master.
   5. Other arguments sent by the master.
4. The process after start reports to master about its start.
5. The master introduces this new application/guest to all the other applications.

Introducing guest

1. All new applications in the domain are considered as guest and are introduced to other apps after the guest reports to master.
2. A *IntroPDU* is sent to all other applications
3. After receiving the *IntroPDU* the applications say *“Hi”* to this new application by sending a *HiPDU.* This helps the guest in knowing the ports on which other applications run.
4. After receiving a *HiPDU* the applications send back a “Hi” (similar *HiPDU*).

Heartbeat

1. The apps running in the framework send heartbeat signals to the master about its activeness after every HEARTBEAT\_INTERVAL milliseconds.
2. The master waits 3 \* HEARTBEAT\_INTERVAL time for the heartbeat of an app and decides that the app is hung or dead after this time.