Pre-Proposal

Xi He
xxh2229@rit.edu

Committee members
Chair:Alan Kaminsky

January 26, 2010

1 Problem Description

Parallel Java (PJ) is an API and middleware for parallel programming in 100% Java on shared memory multiprocessor (SMP) parallel computers, cluster parallel computers, and hybrid SMP cluster parallel computers. Current implementation of PJ spawns a separate SSH session for each back-end processor to start each back-end process and authenticate each back-end process into the user's account. However, SSH is too "heavyweight": it creates too many processes, and it takes too long to get the job started.

Another potential improvement on PJ is its web interface. Right now PJ's web interface only displays the PJ job queue status. To facilitate easy access and manipulation of PJ, more functionalities should be added to its web interface.

2 Project Goal

The goal of the project is two-fold. One is to design and implement a different back-end process launching and authentication scheme that is faster and doesn't require SSH. Another goal of the project is to enable end users to submit jobs, change the order of jobs in the queue, and cancel jobs using PJ's web interface. Proper credentials are also supplied;

3 Proposed Work

The proposed job launching scheme works as follows:

- PJ accepts a user's request and prepares for running the user's job.
- PJ detects if there is a daemon on behalf of the user running on each back-end node.
- If no, PJ start a daemon on each back-end node and start running the job. Let the daemons and PJ share an authentication key.
- If yes, PJ can communicate with daemons and submit the job to daemons with all messages authenticated using a message authentication code (MAC).

In the proposed PJ web interface, there are different views for different users.

- For the anonymous user, he can only see the PJ job queue status.
- For the login user, he can submit new jobs to PJ. He can also cancel his jobs which are queuing or running.
- For the administrator, he can change the priority of a job. He can also cancel any jobs queuing and running.