Scope

Our project's scope is the interface between the scientists and the high-performance computing layer of the Pleiades system. The goal is to provide scientists with a client integrated into VisTrails to submit processing requests to be run at NASA's Pleiades systems and receive updates on the status of their processes. We gather these processes in a remote server, which will also contain a scheduler apart from that from the HPC layer.

Concerns

1. Scalability of the scheduler. bottleneck.
2. MapReduce plugin. Hadoop. can't leverage hadoop directly. own mapreduce solution.

The scalability of the server holding the requests is a concern, because it could be a potential architectural bottleneck. Scientists submit processes and receive updates to and from the remote server through VisTrails. Therefore, increased traffic to this remote server could impede or disrupt scientists from being able to submit their processes and receive updates. A potential solution could be to increase the number of remote servers and allow VisTrails to choose the remote server to use based on the number of current requests of each server.

The availability of a MapReduce plugin is another concern, because there are currently no openly-available ways to split a VisTrails program into a MapReduce program; without such a plugin, the VisTrails program can only be run as a batch job and not in parallel. A possible Hadoop solution could not be found either. A solution to this concern could be to write an independent MapReduce solution, but it would be very resource-intensive.

Principles

The three key principles in our architectural design are the blackboard, client/server, and publisher/subscriber models. The blackboard architecture model will be used to collect different scientists processing jobs to a central server. The goal of this is to decouple the scientists from the HPC servers, and instead, handle the scheduling in a remote scheduling server. This remote server will contain a 'blackboard' of requests to be processed and a scheduler that employs an algorithm to launch scripts from a front-end node in the Pleiades system.

deploy requests to the HPC servers.

The client/server model will be used as well. Specifically, a thin client and fat server will be used to increase responsiveness on the client side. The scheduling will be moved to the designated scheduler server mentioned earlier.

The publisher/subscriber model is used to increase responsiveness of the system. Scientists will subscribe to the scheduler server and receive notifications on the status and progress of their processing requests.

Constraints

A constraint is the accessibility of PBS through the Pleiades front-end nodes. Accounts must be granted to access these nodes. Therefore, to begin development of the plugin, we need to gain developer access to NASA's computing resources. In prototyping the plugin, the different nodes in the Pleiades system and the servers involved in the plugin can be simulated. However, access of NASA's nodes is required to test the integration of the plugin more comprehensively.