PANFISH: A Multi-Cluster Submission System

Christopher Churas
University of California
San Diego
9500 Gilman Drive, MC 0446
La Jolla CA, 92093-0446
churas@ncmir.ucsd.edu

Abel W. Lin
University of California
San Diego
9500 Gilman Drive, MC 0446
La Jolla CA, 92093-0446
awlin@ncmir.ucsd.edu

Jeffrey S. Grethe
University of California
San Diego
9500 Gilman Drive, MC 0446
La Jolla CA, 92093-0446
jgrethe@ncmir.ucsd.edu

Mark H. Ellisman
University of California
San Diego
9500 Gilman Drive, MC 0608
La Jolla CA, 92093-0608
mark@ncmir.ucsd.edu

ABSTRACT

Scientific data is growing at an explosive rate. The volume of information, in both public domain databases and private research projects, is stressing the current capacity of high-performance computational systems. In the field of metagenomics, this growth of data is best represented by data generated from next generation sequencing (NGS) platforms. Already, CRBS's current remote cluster execution (RCE) system enables access to several XSEDE clusters. Following current cluster computing usage norms, a workflow (and its component jobs) is limited to a single cluster resource for computation. With continually growing data sizes and resultant computational requirements, however, it has become necessary to coordinate multiple clusters for a single workflow. PANFISH is designed to address this need.

Using XSEDE compute resources (or any external compute resource) typically requires the following three operations to be performed: upload data, run workflow, and copy back results. PANFISH (semi-)automatically coordinates these operations across multiple cluster resources. Furthermore, PANFISH has been expressly designed to enable job submission in a process similar to invocation of jobs on a single local cluster running Oracle/Sun/Open Grid Engine (SGE) - with the added advantage that those jobs can optionally be sent to multiple XSEDE resources.

Invoking PANFISH *chum* uploads data to designated XSEDE resources. Next, a *cast* command submits the jobs to the respective clusters (*cast* is a dropin replacement for SGE qsub). Jobs are monitored via the id returned from *cast* (as with any normally submitted SGE job). Finally, upon job completion, a *land* command is used to retrieve the data from the XSEDE resource. Specifically, *cast* submits what is known as a shadow job to a set of shadow queues that correspond to the local cluster and other XSEDE resources. SGE itself handles the scheduling of these shadow jobs to the appropriate shadow queue. Once scheduled and running, these shadow jobs examine what queue they fall under and notify the PANFISH daemon. The PANFISH daemon on the local compute cluster then submits those jobs to the corresponding compute resource notifying the shadow job upon completion. The shadow job then exits letting the user know the work has completed. The *chum* and *land* commands are wrappers that simplify the transfer of data to and from remote resources.

Part of the challenge of using multiple XSEDE resources in an coordinated matter is to address the heterogeneity of the resources. For example, with regard to schedulers, XSEDE resources located at SDSC and RCAC utilize PBS while resources at TACC utilize SGE. To address this heterogeneity, PANFISH was designed to require no external libraries and no configuration adjustment on the remote clusters other than enabling ssh access from the local site.

Using PANFISH, we are able to submit jobs from a single workflow across several XSEDE clusters (from multiple locations). Already in beta use, we anticipate PANFISH will shortly replace the current RCE system (which itself has been successfully used for the past year and a half to send over 1,800 workflows to XSEDE resources utilizing 150+ CPU years). While PANFISH has been developed using a parallel implementation of the BLAST application, it is designed to accept most "pleasantly parallel" applications. Moving forward, we have already begun to utilize PANFISH for 3D image analysis and segmentation algorithms.

XSEDE13 2013 San Diego, California USA