Running A Seismic Workflow Application on Distributed Resources

S. Callaghan[1], P. Maechling[1], K. Vahi[2], G., Juve[2], E. Deelman[2], Y. Cui[3], E. Poyraz[3], T. H. Jordan[1]
[1]University of Southern California, [2]USC Information Sciences Institute, [3]University of California, San Diego
[1](scottcal, maechlin, tjordan@usc.edu), [2](vahi, gideon, deelman@isi.edu),[3](yfcui, epoyraz@ucsd.edu)

ABSTRACT
In this poster, we present an approach to running workflow applications on distributed resources, including systems without support for remote job submission. We show how this approach extends the benefits of scientific workflows, such as job and data management, to large-scale applications on open-science HPC resources such as Blue Waters, Stampede, and USC HPCC. We demonstrate this approach with SCEC CyberShake, a physics-based seismic hazard application, to run over 470 million tasks via 32,000 jobs submitted to Blue Waters and Stampede.

Keywords
scientific workflows, distributed workflows, seismic hazard analysis, application performance, high throughput

1. INTRODUCTION
Probabilistic seismic hazard analysis (PSHA) provides a technique for quantifying seismic hazard, useful for civic planners, building engineers, and other decision-makers seeking to minimize seismic risk exposure. As part of its program of earthquake system science research, the Southern California Earthquake Center (SCEC) has developed a simulation platform, CyberShake, which performs physics-based PSHA using 3D wave propagation simulations [3]. Using CyberShake, we have created the first physics-based PSHA models of the Los Angeles region. These models are "layered", meaning that high-level hazard maps for a region can be decomposed all the way down to hazard contributions from individual ruptures (Figure 1).

To produce a hazard curve for a location of interest, CyberShake constructs a regular mesh populated with material properties from a community velocity model, which is then used to simulate a tensor-valued wavefield (Strain Green Tensors or SGTs) for both horizontal dimensions. These calculations are performed with parallel MPI codes, including the highly scalable AWP-ODC-SGT [2], along with several pre- and post-processing jobs. Seismic reciprocity is then used to simulate about 415,000 seismograms, which are then distilled into specific intensity measures such as peak acceleration and combined into a hazard curve. These stages require high throughput computing, as they include a large number of loosely coupled, short-running, serial jobs. We use scientific workflows to fulfill the job, data, and metadata requirements of the CyberShake platform [1].

To meet our seismology goals of calculating four seismic hazard maps for the Los Angeles area, we needed to expand the scope of our workflows to support execution at larger scales on two new systems, Blue Waters and Stampede. Section 2 outlines the challenges we confronted in this expansion, and the solutions we engineered. Section 3 discusses the simulations we performed using this approach, which we refer to as CyberShake Study 13.4.

SC'13, November 17–22, 2013, Denver, CO, USA.

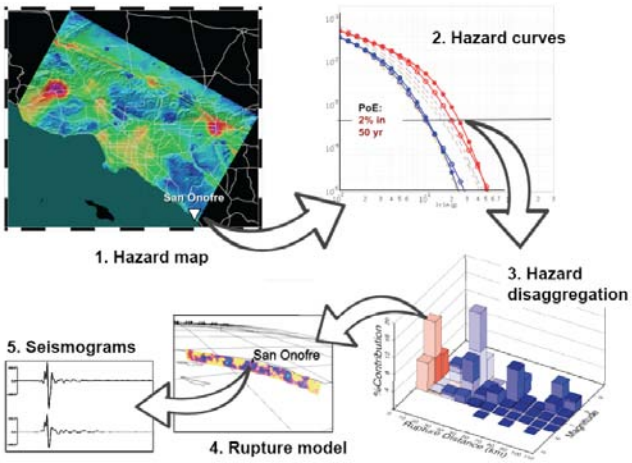

Figure 1: The CyberShake hazard model, showing the layering of information, from hazard maps through hazard curves, disaggregation by rupture, rupture surface, and seismograms. Arrows show how the model can be disaggregated from high levels to access information of progressively lower levels.

2. WORKFLOW CHALLENGES
The workflow system described in this poster extends a large-scale workflow system we constructed for our CyberShake 1.0 Study in 2009 [1]. Expanding the CyberShake workflows to execute on new resources with differing architectures brought challenges. In early 2013, when this work was performed, Blue Waters had many cores, large file systems, and short queues, so it was ideal for rapid SGT calculations. Stampede supported remote job submission via GRAM, so it was more conducive to running the high throughput seismogram reciprocity calculations (the "post-processing" workflow) using the Pegasus Workflow Management System (Pegasus-WMS). Therefore, we decided to split the execution such that the SGT calculation workflow is performed on Blue Waters, and the post-processing workflow is performed on Stampede.

2.1 Blue Waters Job Submission
Our seismology goals required the execution of almost 10,000 SGT workflow jobs. To avoid manually submitting them to the Blue Waters queue, we designed a minimalistic workflow system to automate the execution of the SGT workflows. We begin with a list of all the sites for which we will calculate hazard curves.

We then created a cron job which selects the next site from the list and calls a Python script, which creates a series of PBS scripts defining all the jobs which make up the workflow, along with a description of the dependencies between the jobs. Next this workflow description is executed. We use qsub dependencies to submit all of the SGT workflow jobs to the Blue Waters queue at once, letting the scheduler enforce the job dependencies and avoiding the need for a custom scheduler.

To avoid overloading the Blue Waters scheduler, we added a check to our cron job to first see how many SGT workflow jobs are in the scheduler and only submit new workloads if the number of queued jobs is below a user-defined threshold. Since the scheduler removes dependent (child) jobs if a parent job fails, we implemented logging and restart capabilities to help ensure all jobs are scheduled, run, and completed successfully.

2.2 Distributed Workflow Execution

Our experience with large-scale scientific workflows has shown us that a fully automated approach is necessary to avoid human-induced bottlenecks. Previously, we had been able to orchestrate the execution of both the SGT and post-processing workflows from a SCEC workflow submission server. However, since Stampede and Blue Waters supported different job submission techniques, we designed an automated approach that enables a seamless transition from one system to the other.

To do this, we implemented a new job in the SGT workflow we call Handoff. The Handoff job uses SSH keys to authenticate with the SCEC submission server and fire off a data transfer job, which stages the SGT files from Blue Waters to Stampede in preparation for the post-processing workflow and registers them in the Globus RLS for later discovery. We use the Blue Waters proxy server to generate an X509 user proxy for GridFTP authentication for data transfer. When the data transfer job completes, the Handoff job then adds the site to a queue of post-processing sites on the SCEC submission server. A cron job running on the SCEC server will periodically take sites from the post-processing queue, create a Pegasus DAX workflow representation, plan it into a Condor DAG, and submit it to Condor DAGMan for execution on Stampede.

With this approach, once the initial site list on Blue Waters is populated, both the SGT and post-processing workflows execute without human intervention, a necessity for calculating hazard curves for hundreds to thousands of locations.

3. CYBERSHAKE STUDY 13.4

In Spring 2013, we performed CyberShake Study 13.4 to produce four hazard maps for the Los Angeles area, enabling hazard comparisons between different codebases and velocity models.

Using the approach outlined in Section 2.1, we executed the SGT workflows on Blue Waters. When complete, each set of SGTs was passed off to the SCEC submission server via a Handoff job and queued up for post-processing.

We calculated 1132 sets of SGTs in Southern California in 560 hours, averaging usage on 19,300 cores during that time. In total we submitted 9656 jobs to the Blue Waters queue using our automated workflow system. We limited the number of queued jobs at once to 70, or approximately 7 sites' worth of SGT calculations, to stay within queuing policy limits while being high enough to avoid deadlock. In practice, since only a few of the queued jobs are eligible for execution at any one time – the others are waiting on dependencies – we could increase the queued job cutoff in future studies, increasing throughput.

Our post-processing workflows were executed on Stampede. To facilitate high throughput with the short-running (< 1 minute) seismogram synthesis tasks, we used pegasus-mpi-cluster (PMC), a tool distributed with Pegasus-WMS which wraps serial or thread-parallel tasks in an MPI wrapper and executes them using a master-worker paradigm [4].

We calculated 1132 hazard curves over 739 hours, executing about 470 million tasks in the process. Using PMC enabled us to reduce the number of jobs submitted to the Stampede queue to 21,912, yielding an average throughput of 29.7 jobs and 636,000 tasks per hour. Stampede PMC job times averaged about 5 minutes. For future CyberShake studies conducted on Stampede we plan to modify our workflows to execute fewer, longer-running PMC jobs to better amortize the queuing overhead.

4. CONCLUSION

Scientific workflows continue to be a key tool in managing the complex task and data requirements of large-scale, high throughput applications. Despite the research community's need obvious need for workflow capabilities, as supercomputer architectures become more specialized, support for remote job submission is less frequent. To bridge the gap between our research goals and the capabilities of existing open science supercomputers, we have developed a practical technical approach for integrating compute resources which lack remote job submission support into an automated workflow system, and demonstrated its effectiveness in performing a suite of CyberShake calculations four times larger than previous suites, in 60% of the wallclock time [1]. We plan to continue developing scientific workflow solutions capable of execution on varied resources, enabling us to further push the boundaries of large-scale computational earthquake system science.

5. REFERENCES

[1] Callaghan, S., Maechling, P., Small, P., Milner, K., Juve, G., Jordan, T., Deelman, E., Mehta, G., Vahi, K., Gunter, D., Beattie, K., and Brooks, C.X. Metrics for heterogeneous scientific workflows: a case study of an earthquake science application. *International Journal of High Performing Computing Applications* 25, 3 (Aug 2011), 274-285.

[2] Cui, Y., Poyraz, E., Olsen, K. B., Zhou, J., Withers, K., Callaghan, S., Larkin, J., Guest, C., Chourasia, A., Shi, Z., Day, S. M., Maechling, P., and Jordan, T. H. Physics-based seismic hazard analysis on petascale heterogeneous supercomputers. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (*SC'13), in press.

[3] Graves, R., Jordan, T. H., Callaghan, S. Deelman, E., Field, E., Juve, G., Kesselman, C., Maechling, P. Mehta, G., Milner, K., Okaya, D., Small, P. and Vahi, K. 2011. CyberShake: A physics-based seismic hazard model for Southern California. *Pure and Applied Geophysics*, 168, 3 (Mar. 2011), 367-381.

[4] Rynge, M., Juve, G., Vahi, K., Callaghan, S., Mehta, G., Maechling, P., and Deelman, E. 2012. Enabling large-scale scientific workflows on petascale resources using MPI master/worker. *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment* (Chicago, IL, USA, July 16-20, 2012). XSEDE '12. ACM, New York, NY, Art. 49.