

Towards Enabling Mid-Scale Geoscience Experiments Through Microsoft Trident and Windows Azure

Eran Chinthaka Withana¹, Beth Plale^{1,*}, and Craig Mattocks²

¹School of Informatics and Computing, Indiana University, Bloomington, Indiana, USA

²Miami University, Miama, Florida, USA

June 18, 2011

1 Introduction

Geoscience experiments we have encountered have a wide range of resource demands depending on the underlying prediction models being used. We have identified two categories of geo-science applications based on resource consumption. One class of applications demand HPC hardware and have strict software requirements. The second class of applications take very small amount of time to execute a single job on a desktop computer but often used in ensemble runs which schedules large number of small jobs. Weather Research and Forecast model(WRF) [1], the numerical weather prediction model for mesoscale weather predictions, falls into the compute intensive category of applications. It has long been useful and continues to be useful in a variety of scientific domains. But the compute intensive nature of WRF has not only become a challenge but also has enforced serious limitations on its usability for mid-scale computer scientists who often scramble to find sufficient computational resources to test and run their codes. The probabilistic ensemble execution of the Sea, Lake and Overland Surges from Hurricanes (SLOSH) [2] storm surge prediction model falls into the second class of applications and enables the study of change in the strength and impact of storms that start over the oceans. SLOSH has given access to and manipulation of climate model scenarios for emergency management and personnel and local government officials.

The unique computational model of cloud computing can benefit the scientific endeavor in that it conveys a sense of infinite compute resources available the instant they are needed, eliminating the need for advance reservations. Additionally, compute and storage resources are rented to users on an on-demand basis and users are billed according to their usage [3]. The pay as you go model eliminates an up-front commitment by users. Experiments can be started small and scaled as need grows. The pay-as-you-go model also encourages scientists to budget for and use resources they can and are willing to pay for. Cloud computing resources can be utilized to serve as extra resources for the mid-scale scientist who do not have sufficient compute resources to schedule and execute geo-science applications.

This paper describes progress to date in a collaborative project between computer scientists, and atmospheric and climate scientists to develop a framework for utilizing cloud resources to run large scale geoscience experiments. Leveraging ongoing efforts in building and executing meteorological models on Windows HPC Server, orchestration of atmospheric workflows through the Trident Scientific Workflow Workbench, and extending components developed in the Linked Environments for Atmospheric Discovery (LEAD) project [4], we use the Azure cloud [5] as a platform for executing both numerical weather prediction models and numerical storm surge ensemble runs from which storm surge predictions can be made. Each of these is described below:

Weather forecast modeling The Weather Research Forecast Model (WRF) is designed to serve both operational forecasting and atmospheric research needs and is being used by scientists across multiple domains. The architecture of WRF supports a high degree of computational parallelism For example, during support for the NSF funded Vortex2 [6]

*To whom correspondence should be addressed. Email: plale@cs.indiana.edu

effort, we executed short term forecasts using WRF on 512 cores of the Indiana University supercomputer, Big Red. Each run was executed at 10km resolution on a 800km x 800km domain. This experiment configuration took 50 minutes of all 512 cores of Big Red supercomputer. Each experiment took 1 GB of input and generates 5 GB of output.

Storm surge modeling Probabilistic ensemble executions can be used for storm surge prediction within a regional-sized basin. These simple simulations, which take only a few minutes to run on a medium-sized workstation, gain their power when run in vast numbers. We are targeting the Sea, Lake, and Overland Surges from Hurricanes (SLOSH) model. In a typical execution scenario, 15,000 instances of SLOSH are run so as to avoid statistical approximations. The individual instances are independent, so amenable to high degrees of parallelism for which cloud resources are well suited. A single instance of a SLOSH ensemble run takes 3 GB of input and generates 8 GB of output, making it well suited to data center execution.

In this whitepaper we briefly discuss the framework that enables a researcher to use cloud computing resources to run geo-science experiments. We build on our experiences in successful deployment of the Weather Research Forecast (WRF) model, WPS, and GRADS visualization toolkit on Windows HPC Server and Azure. The WRF demonstration was done using the VM support built into Azure. We will discuss the infrastructure we have built to utilize Trident for large scale workflows, including Sigiri [7], our framework for managing interactions with Grids and Clouds. A focus of the future work is on fault tolerant ensemble execution using Azure worker roles, orchestrated through Trident, with metadata capture and management of data results.

2 Motivation

From our experience in working with scientists in different domains and evaluating existing technical solutions we identify the following requirements for a framework that can be used in an eScience environment. The framework must be able to

- provide a uniform and interoperable interface for external entities to interact with it.
- support heterogeneous compute resource manager interfaces and operating platforms from grids, IaaS, PaaS clouds, departmental clusters.
- be extensible to support new and future resource managers with minimal changes.
- provide monitoring and fault recovery, especially when working with utility computing resources.
- provide light-weight, robust and scalable infrastructure.
- be integrated to variety of workflow environments.
- provide ease of installation and maintenance.

Assessing existing research points to several open challenges. The Carmen [8] project provide a cloud environment that has enabled collaboration between neuroscientists. This framework allows scientists to upload, share and analyse data. Carmen system requires all the programs to be packaged as WS-I [9] compliant Web services so that the services can be dynamically deployed, using Dynasoar [10] dynamic service deployment infrastructure. Recent improvements have enabled scientists to use Windows Azure resources to schedule and execute scientific jobs. But the strict requirements it enforces on applications to be deployed is challenging for certain category of scientific applications. For example, WRF requires strict software and hardware requirements, so integrating it to the current Carmen infrastructure could be a daunting task. Condor [11] pools could be utilized to unify certain compute resource interactions. But Condor uses Globus toolkit [12] (and GRAM underneath) and does not have adequate support for ensemble run configurations and failure recovery and reliability [13] that our scientific applications demand. Further, Condor overlooks the failure modes of a cloud platform. Our system attempts to address the gaps in existing functionality throughout a uniform interface, and with built-in support for cloud failure modes when carrying out massively parallel applications.

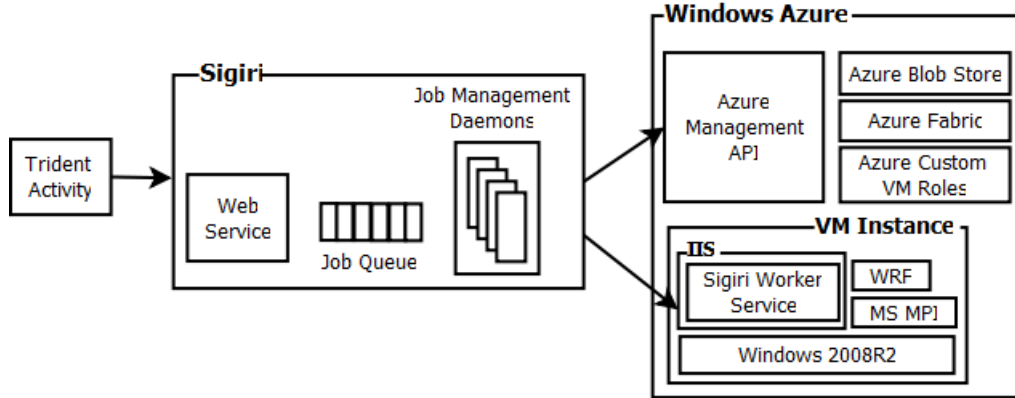


Figure 1: Integration Framework for Microsoft Trident, Windows Azure and Sigiri for scientific job executions

3 Proposed Framework

The framework shown in Figure 1 will enable a researcher to use cloud computing resources to run parallel experiments using the Trident workbench and Azure cloud resources. This framework is an extension to our resource abstraction service, Sigiri [7], which manages interactions with Grids and Cloud computing resources. The framework implementation has three main components 1.) Trident activity to interact with Sigiri, the resource abstraction service, 2.) Sigiri Web service and 3.) Sigiri daemons for job execution management within Windows Azure. Each are described below.

Trident Scientific Workflow Workbench activity The Trident [14] workbench is a Windows desktop tool that supports construction of workflows. Suppose a workflow is a directed graph of edges and nodes where the nodes are task execution and the edges are either control or data flow between nodes. A workflow node is implemented in Trident as a C activity. Once deployed, an activity is then registered for later integration into a workflow. To enable the interaction between Trident and Sigiri, we implemented a special Trident activity that collects information related to the execution of a scientific job, including paths for input data files, and job submission requests to Sigiri Web service. Once submitted, the Trident activity uses the return job handle to continuously monitor the state of the job through queries to the Sigiri Web services API. Once the job completes the activity will output the location of output files for use by the next activity.

Sigiri web Service Sigiri, a light weight resource abstraction service [7], enables the interaction with variety of compute resources by exposing a uniform API for the user. Upon the receipt of a job into the system it queues this job in the system. Also the management capabilities within Sigiri enable one to monitor the job status using its Web services based query API.

Sigiri Azure Daemon Sigiri maintains separate daemon workers to interact with compute resources it manages. Each daemon can interact with their respective job managers, using their exposed job management interfaces, to get the jobs scheduled and monitor afterwards. Once a job is queued within Sigiri, the respective daemon will pick up the job, performs necessary transformation and starts the interactions with resource managers. Likewise, the Sigiri Azure daemon will pickup jobs that are required to be launched in Windows Azure and request for workers to schedule the jobs. Current implementation supports using virtual machine roles within Windows Azure, but the proposed extension will also support worker roles to schedule and execute jobs.

4 Applications

Our group did the first successful deployment of the Weather Research Forecast (WRF) [1] model onto Azure using the VM worker roles, that is, support built into Azure to host and execute applications as virtual machines. In this experiment, initial conditions from North American Mesoscale Model (NAM) were pre-processed using WRF Pre-processing System (WPS) before feeding into WRF model. Once WRF is executed, the GrADS [15] visualization tool renders the required visualizations of the weather model. ARWPost is an intermediate step used to generate input files

that can be read by GrADS. The complete workflow of these applications is scheduled and executed within Windows Azure using our framework. Since the software stack needed to execute WRF experiment has specific software requirements (need for MPI [16] and Cygwin) we incorporate the use of customized VM roles to schedule these jobs in Windows Azure resources. The customized VM we create to execute WRF on Microsoft MPI layer is stored at the Azure Blob Store. Once a job is submitted to our framework, Sigiri employs the customized VM images to create VM roles within Azure. Jobs are then submitted and monitored for completion.

Our framework will be extended to manage the job submissions and the life cycle of the large number of worker instances created on demand to run SLOSH ensembles. This extension will also include the support for an ensemble planner that can create a job execution plan connecting one SLOSH job with one and only configuration file. These jobs will then be queued into our job execution framework to be executed in Azure cloud using the custom SLOSH worker role instances. One of the major difficulties in handling a large number of VM instances in the cloud is fault tolerance. With many components involved, both from our infrastructure and the cloud provider, there are numerous components that can go down. Also the scheduled jobs can fail for numerous reasons including communication failures, network partitions, data outage, etc. There has to be a supporting infrastructure that will monitor our system and also the jobs scheduled to guarantee fault tolerance in our system. The proactive replication infrastructure within Sigiri will provide the necessary fault tolerance for job executions to guarantee the eventual execution of large number of jobs scheduled in the cloud.

References

- [1] J. Michalakes, J. Dudhia, D. Gill, T. Henderson, J. Klemp, W. Skamarock, and W. Wang, “The weather research and forecast model: Software architecture and performance,” in *Proceedings of the 11th ECMWF Workshop on the Use of High Performance Computing In Meteorology*. Citeseer, 2004, pp. 156–168.
- [2] C. Jelesnianski, J. Chen, W. Shaffer, U. S. N. Oceanic, A. Administration, and U. S. N. W. Service, *SLOSH: Sea, lake, and overland surges from hurricanes*. US Dept. of Commerce, National Oceanic and Atmospheric Administration, National Weather Service, 1992.
- [3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “Above the clouds: A Berkeley view of cloud computing,” EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, 2009.
- [4] K. Droegemeier, D. Gannon, D. Reed, B. Plale, J. Alameda, T. Baltzer, K. Brewster, R. Clark, B. Domenico, S. Graves *et al.*, “Service-oriented environments for dynamically interacting with mesoscale weather,” *Computing in Science & Engineering*, vol. 7, no. 6, pp. 12–29, 2005.
- [5] “Windows azure platform,” <http://www.microsoft.com/windowsazure/>.
- [6] B. Plale, C. Herath, and E. C. Withana, “Towards proxy workflow execution in environmental research: Application to vortex2,” *Environmental Research Workshop*, July 2010.
- [7] E. C. Withana and B. Plale, “Sigiri: Uniform abstraction for large-scale compute resource interactions,” School of Informatics and Computing, Indiana University, Bloomington, Indiana, Tech. Rep. TR693, 2011, <https://www.cs.indiana.edu/cgi-bin/techreports/TRNNN.cgi?trnum=TR693>.
- [8] P. Watson, P. Lord, F. Gibson, P. Periorellis, and G. Pitsilis, “Cloud computing for e-science with carmen,” in *2nd Iberian Grid Infrastructure Conference Proceedings*. Citeseer, 2008, pp. 3–14.
- [9] K. Ballinger, D. Ehnebuske, C. Ferris, M. Gudgin, C. Liu, M. Nottingham, and P. Yendluri, “Basic profile version 1.1,” *WS-I Specification*, vol. 8, pp. 1–1, 2004.
- [10] P. Watson, C. Fowler, C. Kubicek, A. Mukherjee, J. Colquhoun, M. Hewitt, and S. Parastatidis, “Dynamically deploying web services on a grid using dynasoar,” 2006.

- [11] J. Frey, T. Tannenbaum, M. Livny, I. Foster, and S. Tuecke, "Condor-G: A Computation Management Agent for Multi-Institutional Grids," *Cluster Computing*, vol. 5, no. 3, pp. 237–246, 2002.
- [12] I. Foster, "Globus Toolkit Version 4: Software for Service-Oriented Systems," *Network And Parallel Computing: IFIP International Conference, NPC 2005, Beijing, China, November 30-December 3, 2005: Proceedings*, 2005.
- [13] S. Marru, S. Perera, M. Feller, and S. Martin, "Reliable and Scalable Job Submission: LEAD Science Gateways Testing and Experiences with WS GRAM on TeraGrid Resources ," *TeraGrid Conference*, June 2008.
- [14] R. Barga, J. Jackson, N. Araujo, D. Guo, N. Gautam, K. Grochow, and E. Lazowska, "Trident: Scientific workflow workbench for oceanography," in *IEEE Congress on Services-Part I, 2008. SERVICES'08*, 2008, pp. 465–466.
- [15] B. Doty, "Using the grid analysis and display system (grads)," *Center for Ocean-Land-Atmosphere Interactions (COLA), College Park, MD, University of Maryland*, 1985.
- [16] M. Snir, *MPI—the Complete Reference: The MPI core*. The MIT Press, 1998, vol. 1.