ModflowOnAzure: An On-demand “Groundwater Modeling as a Service” Solution

Yong Liu
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
1205 W. Clark St.
Urbana, IL 61801, USA
yongliu@ncsa.illinois.edu

Yan Xu, Wenming Ye
Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
{yanxu, wenming.ye}@microsoft.com

Abstract

Cloud computing has emerged as one of the preferred platforms for on-demand, event-driven computing. However, literature on event-driven scientific modeling on the cloud is sparse. This paper presents our vision and architecture to provide a “Scientific Modeling as a Service”in an on-demand fashion. We choose one of the most widely used groundwater models, MODFLOW2000, from USGS to develop a ModflowOnAzure service on the Windows Azure platform as our first implementation. Some specific issues on handling scientific models in Azure are presented including file transfer and synchronization through the dropbox API and provenance tracking in the Cloud using the Open Provenance Model.

1. Introduction

Scientific modeling can be both computationally-demanding and data-intensive, which is a good candidate for cloud computing platforms. There is a strong need for supporting on-demand, event-driven ensemble modeling in a stochastic simulation mode. In this paper, we develop an architecture for “Scientific Modeling as a Service” (SMaaS) with an initial implementation of ModflowOnAzure, a user-driven on-demand execution of stochastic simulation of MODFLOW[4] using an example use case from the Arizona Department of Water Resources (ADWR).

2. An eScience Use Case

There is a growing need for sustainable groundwater management and decision support in the nation. For example, stakeholders of Arizona water resources management need to periodically run the groundwater model developed at ADWR for evaluating different scenarios while the group meets. A common requirement is that ensemble runs of different scenarios are needed at the time of the meeting. Although each execution of the MODFLOW may be on the order of minutes, hundreds of stochastic runs of such models can quickly increase the computing time to several hours, significantly hinder the decision makers’ capability to perform science-based decision. Such surge computing need can be ideally met by using the Cloud computing.

3. A “Scientific Modeling as a Service” Framework

Building a “Scientific Modeling as a Service (SMaaS)” framework in the Cloud is our solution to solve such surge computing needs for science-based decision making. In this SMaaS framework, we specifically address the following common scientific modeling challenges:

(1) File synchronization and transfer between desktop and Cloud: A typical scenario is that users often want to use a GUI (Graphic User Interface)-based desktop software to generate a set of ensemble input files. These files are then used as part of the inputs for the ensemble runs. The size of these files can be on the order of GBs and users might change them frequently. For example, modelers often use GUI-based Visual MODFLOW to generate the streamflow realization boundary conditions. We use the Dropbox [1] to synchronize the file transfer among users’ desktop, web storage and the Cloud. Specifically, we use Dropbox’s web API in the Azure Worker role to retrieve/deposit the input/output files. This also avoids uploading large amount of data through the Web interface.

(2)Message-content-based worker role instantiation and random model execution based on ensemble sets of inputs. It is ideal to randomly sample from a huge set of model data files with different combinations of inputs when running stochastic models. Typically, users already have some batch files to run the model. In our cloud implementation, we leverage the durable storage of Azure Blob to store the input data as well as the batch scripts that are needed and we generate a randomized/rewritten batch script based on the user preference when user starts the execution, either from the web interface or a workflow. The worker role will execute the job retrieved from a shared job queue based on different type of message contents (run the model, synchronize the files etc.).

(3) Provenance tracking across the desktop/servers and Cloud. It is often desirable to have an end-to-end provenance tracking of the modeling process. In our design, we use the Azure Table (a table which records RDF triples about causal relationships) to store provenance information that we record in both the Web Role and Worker Role using the Open Provenance Model (OPM)[7]. We then can mashup the Cloud-based OPM records with other OPM records generated locally (non-cloud platforms).

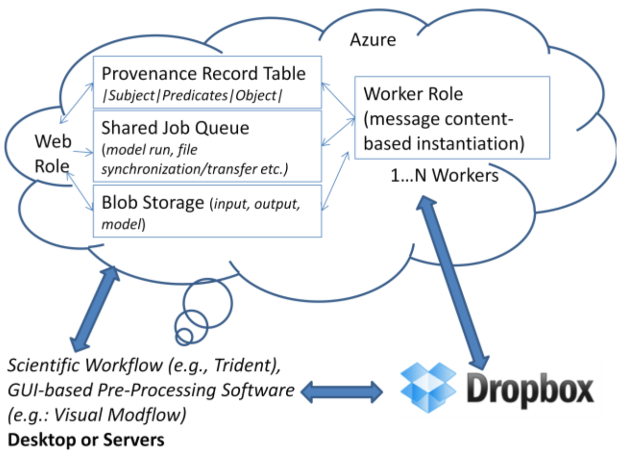

Figure 1. Architecture of Scientific Modeling as a Service

As can be seen in Figure 1, a typical usage of this SMaaS framework is as follows. First, because of the usage of dropbox, users can continually use his/her desktop folder structures without worrying about how to use the cloud resources. The dropbox will automatically synchronize both input/out of the model regardless which sides (desktop application or the Cloud worker) change the files. The Cloud worker will also automatically harvest the files if there is any change made in the dropbox. Second, once the input files and models are in the Azure Blob storage, user can start the model execution through the Web Role interface, where user can browser the available model and inputs and decides how many runs s/he needs by submitting a job request with a number of realizations. This job request can also be done through a workflow such as Trident. Thirdly, a message content-based worker will continuously retrieve the messages from the shared job queue. If a job request wants to execute 100 runs of MODFLOW, then a worker will populate the shared job queue with another hundred jobs with randomly generated batch scripts for the execution. The number of workers can be instantiated by posting a different job in the queue so that the configuration change can be made dynamically. Finally, the output can be put into the Blob as well as sent back to the dropbox, which in turn will show up in a user's desktop's file structure. This saves users' effort on locating the output from the cloud. OPM records are stored in the table and we can use other external tool such as Tupelo[6] to mashup the cloud generated OPM with other OPM records in the workflow.

4. An Implementation: ModflowOnAzure

Our initial implementation of the above described framework results a prototype called ModflowOnAzure, where we can run ensemble runs of MODFLOW2000, a widely used USGS groundwater model using a real-world stochastic case study from ADWR. The initial capability of the prototype allows users to execute the MODFLOW on-demand in the cloud.

5. Related Work

There are some limited published literatures in running scientific application in Azure [3][5]. However, none of the previous work considered provenance capture in the cloud and using the dropbox for file synchronization in their architecture design. For running MODFLOW2000 in the cloud, [2] uses GoGrid for comparing the performance and execution time without consideration of the issues addressed in this paper in section 3.

6. Conclusion

In conclusion, we present a general "Scientific Modeling as a Service" (SMaaS) framework with an initial implementation of ModflowOnAzure, a user-driven on-demand execution of stochastic simulation of MODFLOW, a groundwater model. Future work will continue our implementation with dropbox integration and provenance tracking in the cloud.

7. Acknowledgment

The authors would like to acknowledge Keith Nelson, Dale Mason, and Wesley E. Hipke from Arizona Department of Water Resources who provide the model and data that were used in this paper.

8. References

[1] Dropbox: https://www.dropbox.com/developers
[2] Hunt, Randall J et al., 2010, Using a Cloud to Replenish Parched Groundwater Modeling Efforts, Ground Water, 48 (3), doi: 10.1111/j.1745-6584.2010.00699.x
[3] Lu, W. et al., AzureBlast: A Case Study of Developing Science Applications on the Cloud, in Proceedings of the 1st Workshop on Scientific Cloud Computing (Science Cloud 2010), Chicago,IL
[4] McDonald, M. G., et al., 2003, The History of MODFLOW. Ground Water, 41: 280–283. doi: 10.1111/j.1745-6584.2003.tb02591.x
[5] Simmhan, Y. et al., "Bridging the Gap between Desktop and the Cloud for eScience Applications," Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on ,July 2010, doi: 10.1109/CLOUD.2010.72
[6] Tupelo: http://www.tupeloproject.org
[7] Open Provenance Model: http://openprovenance.org/