Cloud computing for integrated stochastic groundwater uncertainty analysis

Yong Liu a, Alexander Y. Sun b, Keith Nelson c & Wesley E. Hipke c

a National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL, USA
b Bureau of Economic Geology, University of Texas at Austin, Austin, TX, USA
c Arizona Department of Water Resources, Phoenix, AZ, USA

Available online: 01 May 2012

To cite this article: Yong Liu, Alexander Y. Sun, Keith Nelson & Wesley E. Hipke (2012): Cloud computing for integrated stochastic groundwater uncertainty analysis, International Journal of Digital Earth, DOI:10.1080/17538947.2012.687778

To link to this article: http://dx.doi.org/10.1080/17538947.2012.687778

International Journal of Digital Earth
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tjde20
Cloud computing for integrated stochastic groundwater uncertainty analysis

Yong Liu\textsuperscript{a*}, Alexander Y. Sun\textsuperscript{b}, Keith Nelson\textsuperscript{c} and Wesley E. Hipke\textsuperscript{c}

\textsuperscript{a}National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, IL, USA; \textsuperscript{b}Bureau of Economic Geology, University of Texas at Austin, Austin, TX, USA; \textsuperscript{c}Arizona Department of Water Resources, Phoenix, AZ, USA

(Received 2 January 2012; final version received 19 April 2012)

One of the major scientific challenges and societal concerns is to make informed decisions to ensure sustainable groundwater availability when facing deep uncertainties. A major computational requirement associated with this is on-demand computing for risk analysis to support timely decision. This paper presents a scientific modeling service called ‘ModflowOnAzure’ which enables large-scale ensemble runs of groundwater flow models to be easily executed in parallel in the Windows Azure cloud. Several technical issues were addressed, including the conjunctive use of desktop tools in MATLAB to avoid license issues in the cloud, integration of Dropbox with Azure for improved usability and ‘Drop-and-Compute,’ and automated file exchanges between desktop and the cloud. Two scientific use cases are presented in this paper using this service with significant computational speedup. One case is from Arizona, where six plausible alternative conceptual models and a streamflow stochastic model are used to evaluate the impacts of different groundwater pumping scenarios. Another case is from Texas, where a global sensitivity analysis is performed on a regional groundwater availability model. Results of both cases show informed uncertainty analysis results that can be used to assist the groundwater planning and sustainability study.

Keywords: MODFLOW; groundwater sustainability; uncertainty analysis; global sensitivity analysis; cloud computing; Windows Azure; Dropbox; alternative conceptual model; cyberinfrastructure; eScience

1. Introduction

Continuing urbanization and regional climate change will increase the likelihood of extreme hydrological events such as storms and drought in the coming decades, resulting in uncertainties about the continual supply of clean water for human uses and for other living things (Serageldin 2009, Schnoor 2010, Oden \textit{et al.} 2011). In particular, groundwater sustainability is gaining increasing attention in recent years (Brunner and Kinzelbach 2008, Gleeson \textit{et al.} 2011, Pandey \textit{et al.} 2011, Sophocleous 2011). Groundwater is the world’s most extracted raw material with an estimated global withdrawal of 600–700 km\textsuperscript{3} per year (Foster 2006) and serves as the primary source of drinking water for as many as two billion people (Morris \textit{et al.} 2003), among many other usages such as irrigation. Historically, however, the practice of...
groundwater withdrawal is often developed without adequate understanding of the balance between its occurrence in space and time and replenishment through recharge. Groundwater flow simulation models are often used to assist planning and management, but there are several uncertainty factors including incomplete knowledge of the groundwater system and hydrogeological data, parameter uncertainty and scenario uncertainty (e.g. climate change). Several techniques have been developed to address these uncertainty issues including the use of alternative conceptual models (e.g. Rojas et al. 2010) and global sensitivity analysis (e.g. Mishra et al. 2009).

Two general limiting factors in previous studies motivate us to conduct our research in this paper. First, easy-to-use on-demand computing resources are usually not available for the so-called ‘surge computing’ to allow quick execution of an integrated and comprehensive evaluation of combinations of multiple plausible parameters, scenarios, and models. Groundwater modeling codes have consistently become more intricate as developers attempt to better simulate the complex interactions among various components including climate variability (Refsgaard et al. 2010). With the addition of this complexity, modelers are better equipped to quantify and test the uncertainties within a large range of model parameters and scenarios. A significant issue to this approach has been the computing time requirement for running numerous scenarios in a timely manner that is useful to water planners. Second, the complexity of groundwater models has also dramatically increased the number of people that are required to interact with the model. People with specific expertise are needed to analyze and interpret a wide range of data and more stakeholders are involved throughout the entire modeling process (Minciardi et al. 2007). Current practices have not led to easy interaction with collaborators or exchanging large amounts of data.

The emerging cloud computing platform may hold the promise to enable on-demand timely groundwater uncertainty analysis not only for researchers, but also for non-modelers such as stakeholders and decision makers, thus democratizing the science-based decision-making process (Barga et al. 2011). Cloud computing (Buyya et al. 2009, Armbrust et al. 2010) has made the vision of ‘computing as the fifth utility’ (along with water, gas, electricity, and telephone) possible. As part of the eco-system of computing infrastructure, scientific computing in the cloud is still in its infancy (Brandic and Raicu 2011, Gunarathne et al. 2011), especially in the groundwater field (Hunt et al. 2010) and the general Digital Earth community (Yang et al. 2011).

We specifically address two major barriers in this paper from the cloud computing perspective, in addition to the support of enabling parallel execution of large number of ensemble runs in the cloud. First, there exists significant complexity to migrate existing codes to the cloud directly due to license or compiler issues. For example, groundwater modelers often use multiple software written in different programming languages such as MATLAB, FORTRAN, and C. Some of these codes may be portable to the cloud platforms, while other pieces may not be feasible because of the need of a valid license on the cloud or compiler issues. As there is no ‘pay-per-use’ for MATLAB yet in the cloud, it is not possible to use the MATLAB code as it is in the cloud. Second, many of the model runs need to generate a large number of files (could be large or small in size), and there is a significant challenge to seamlessly exchange data between a desktop environment and one or multiple
cloud(s). For example, it is very time-consuming to set up a groundwater model and often this is done by Graphical User Interface (GUI)-based desktop software or some customized code to generate the model input files, usually written in MATLAB or other high-level programming language. Thus it is necessary to use the desktop-based software in conjunction with the cloud-based services and to automate the file exchange between desktop and the cloud seamlessly.

This paper presents our research and experiences using a public cloud, Windows Azure, to develop a groundwater modeling service that can accommodate spatiotemporal modeling and uncertainty analysis of a groundwater system. The contribution of this paper is twofold. First, we have developed a ‘ModflowOnAzure’ service that allows an easy-to-use hybrid desktop + cloud computing including integration of Dropbox for ‘Drop-and-Compute.’ This has enabled two new capabilities that were not available before: automatic data exchange between user’s desktop and the cloud using a Dropbox folder and running desktop-licensed modeling software in a hybrid desktop + cloud environment. Second, two real-world scientific use cases were developed to use this ModflowOnAzure service: an alternative conceptual model-based scenario uncertainty analysis in Arizona and a global sensitivity analysis for a groundwater flow model in Texas. These two scientific use cases both drive the development of the ModflowOnAzure and benefit from the easy-to-use on-demand computing services provided by the ModflowOnAzure to produce new insightful knowledge on the uncertainty of the groundwater systems of interest. In our opinion, the availability of such cloud-computing capability demonstrated through ModflowOnAzure can represent a paradigm shift in groundwater modeling community, where there is an increasing need for uncertainty quantification and risk assessment. In addition, the general ‘Scientific Modelling as a Service’ design principles presented in this paper are also expected to benefit the broad Digital Earth community and the ModflowOnAzure system could be applied to support other geoscience modeling efforts.

The rest of the paper is organized as follows. We first review some relevant scientific cloud computing literature and related work. Then we present the methodologies used for uncertainty analysis in this paper. After that, we present the architectural design and implementation of the ModflowOnAzure system. We then present the details of the two case studies in Arizona and Texas. Results obtained from these two case studies are then presented and discussed. Finally, we discuss some lessons learned, broad applicability of our research framework to the Digital Earth and geoscience community, and our future research directions.

2. Scientific cloud computing and related work

The recently released ‘2011 Tech Trends Report’ by IBM surveyed more than 4000 information technology (IT) professionals from 93 countries and 25 industries who provided their views on future IT trends (IBM 2011). One of the major technical trends identified is cloud computing. The top motivators for adopting cloud include flexibility, scalability, and cost-reduction. In the scientific computing domain, cloud computing has gained major interests in the general high performance and distributed computing area. Foster et al. (2008) started a comprehensive comparison of grid computing and cloud computing and suggested that an eco-system of multiple computing platforms will coexist, with cloud computing being more
service-oriented. The recently finalized official cloud definition from the National Institute of Standards and Technology (NIST) (Mell and Grance 2011) lists five essential characteristics of cloud computing: on-demand self-service, broad network access, resource pooling, rapid elasticity or expansion, and measured service. It also lists three ‘service models’ (software as a service, platform as a service and infrastructure as a service), and four ‘deployment models’ (private, community, public, and hybrid) that together categorize ways to deliver cloud services. Figure 1 shows some examples of different cloud service models. Below we discuss the characteristics, usage, and relevant literature related to these service and deployment models in groundwater, geoscience, and general scientific computing domain.

Hunt et al. (2010) used GoGrid, an infrastructure as a service (IaaS) cloud provider, for a parallel Model-Independent Parameter Estimation (PEST) implementation for groundwater uncertainty analysis. The methodology we use for uncertainty analysis is different from that of Hunt et al. (2010), as outlined in the next section. Moreover, we are primarily interested in using a public cloud that belongs to the category of ‘platform as a service (PaaS)’ (Khalidi 2011), as it provides not only a set of virtual machines (VMs), but also a set of programming models and services to improve the productivity of application development. Windows Azure is one of the PaaS public clouds and is used in this paper.

Launched in February 2010, the Windows Azure platform encapsulates common application patterns into two roles (a Worker role and a Web role), removing the need to manually manage low-level details on the VM level. A Worker/Web role is essentially a Windows 2008 Server running in a VM. In addition, Azure provides data storage services (Calder et al. 2011) which include the Azure Blob Storage (ABS) for storing large objects of raw data; the Azure Table Storage (ATS) for semi-structured data, and the Azure Queue Storage (AQS) for implementing message

---

**Figure 1.** Cloud service models and corresponding examples.
queues. Azure also provides a SQLAzure storage service, which is a type of relational database as a cloud service (Curino et al. 2011). Dewan and Hansdah (2011) have provided a comprehensive survey on different cloud storage from different providers (e.g. Amazon EC2, Microsoft Azure, and Google AppEngine). Note that although Windows Azure also provides a Virtual Machine role, which allows Azure to be used as a provider for IaaS, it is still in beta testing stage as of this writing and thus is not considered in this paper.

Several previous scientific cloud computing projects have used Windows Azure. Agarwal et al. (2011) presented two data-intensive science projects and one of them is called MODISAzure which is built on top of Azure to download, store, reproject, and analyze large amount of MODIS satellite images. About 5 terabytes of MODIS images data are processed in the MODISAzure (Li et al. 2010). They developed a generic worker framework where applications are registered to be executed in the cloud using a desktop client through a workflow (Simmhan et al. 2010). Efficient file transfer is managed to minimize exchange of intermediate files in the workflow between the desktop and the cloud. However, they do not consider ensemble model runs in their design. In contrast, we use Dropbox to automatically move files between desktop and cloud, significantly improving the usability of the cloud. Zinn et al. (2010) described a method to directly use a Transmission Control Protocol (TCP) socket to connect a desktop workflow to Windows Azure VMs for data transfer purpose. However, they only consider data processing but not large-scale ensemble runs of models. Lu et al. (2010b) built a generic parameter sweeping framework called ‘Cirrus,’ that is based on their earlier work called AzureBLAST (Lu et al. 2010a, 2010b), a bioinformatics tool that performs genomic sequence matching in the Azure cloud. They primarily consider parameter sweeping that can be described in a declarative scripting language derived from Nimrod (Abramson et al. 1995) for expressing a parametric experiment. The groundwater uncertainty analysis is far more complex. We not only consider multiple plausible parameters, but also multiple alternative conceptual models as well as different natural recharge realizations and different pumping-demand scenarios that cannot be simply represented as a few plausible parameters but rather very complex codes such as a stochastic streamflow generator used in this paper. No consideration of using desktop tools or supporting file exchanges between the desktop and the cloud is discussed in their papers. Luckow and Jha (2010) presented a BigJob framework for managing groups of Azure Worker roles and for remotely executing tasks on them with a biomolecular application on Azure. Their implementation is experimental and does not consider interaction between desktop and the cloud. Subramanian et al. (2010, 2011) presented their work on processing seismic data processing in traditional clusters, Amazon EC2, and Windows Azure. They concluded that Azure is not suitable for tightly coupled data processing. No their experiments are conducted prior to the release of Windows Azure HPC Scheduler (Azure 2011), which now supports MPI (Message Passing Interface) type tightly coupled parallel computational tasks (normally run on a supercomputer). As pointed out by Knight et al. (2007), loosely coupled and independent ensemble runs are increasingly carried out on the distributed computing platforms, not just on a supercomputer. Knight et al. (2007) even used citizen-donated computing resources to conduct their large-scale ensemble computational experiments. Similarly, our uncertainty analysis consists of a
large number of independent computing tasks, which is ideally suited for a cloud such as Azure.

3. Methodologies and computational requirements for groundwater uncertainty analysis
There are several methods we use in this paper to perform uncertainty analyses. In this section, we describe our methodologies as well as the computational requirements for the groundwater uncertainty analysis. Note that all the groundwater modeling is based on the USGS MODFLOW, one of the most widely used groundwater flow models in the field worldwide (McDonald et al. 2003).

3.1. Global sensitivity analysis applied to groundwater modeling
Sensitivity analysis, which aims at quantifying the effects of variations in model input variables on model outputs, is widely used in various engineering analyses. Two broad categories of methods for sensitivity analysis exist in the literature (Tang et al. 2007, Saltelli et al. 2008, Sudret 2008):

- Local sensitivity analysis, which assesses the local impact of input parameter uncertainty on model outputs by calculating changes in model outputs to perturbations of model inputs around their nominal values;
- Global sensitivity analysis, which not only aims to quantify the model output uncertainty caused by uncertainty in the input parameters, but also identifies input parameters that contribute most to the output uncertainty.

Let us use a groundwater model, the main interest in this study, to clarify the difference between local and global sensitivity analyses. Any groundwater model can be expressed in a generic functional form,

\[ u = f(x, \theta) \]  

where \( x \) is set of spatial and temporal variables, \( u \) is set of model outputs (e.g. hydraulic head), \( \theta \in \mathbb{R}^m \) includes all model parameters (i.e. hydraulic conductivity, storage coefficients, and source/sink terms), and \( m \) is the dimension of parameter space. The local sensitivity analysis calculates the so-called sensitivities by varying one parameter at a time, namely

\[ \frac{\partial f}{\partial \theta_j} = \frac{f(\bar{\theta} + \Delta \theta_j) - f(\bar{\theta}_j)}{\Delta \theta_j} \]  

where \( \bar{\theta}_j \) is the nominal value associated with the \( j \)-th model parameter, and \( \theta_j \) and \( \Delta \theta_j \) is user-defined perturbation level (e.g. 10% of \( \theta_j \)). Local sensitivity analysis has long been used in groundwater modeling (Sun 1994) and is the backbone of some popular parameter estimation packages such as PEST (Doherty 2003) and UCODE (Poeter and Hill 1998). As can be seen from Equation (2), local sensitivity methods are linear and are not appropriate for examining complex nonlinear models that involve parameter interactions. By definition, the parameter interaction between two parameters \( \theta_i \) and \( \theta_j \) refers to the part of model response that cannot be written as a
superposition of effects by \( \theta_i \) and \( \theta_j \) alone (Saltelli et al. 2008). The global sensitivity analysis, on the other hand, evaluates the effect of a parameter while varying all other model parameters simultaneously and it accounts for interactions between parameters. Therefore, global sensitivity analysis can provide a useful tool for reducing model complexity and for design under uncertainty.

Among several global sensitivity analysis methods, the variance-based method is the most popular one. It is based on the decomposition of the variance of model outputs into summands of increasing dimension, which is known as Sobol’ decomposition of model output (Sobol’ 1993)

\[
f(\theta) = f_0 + \sum_{i=1}^{m} f_i(\theta_i) + \sum_{1 \leq i<j \leq m} f_{ij}(\theta_i, \theta_j) + f_{1,2,...,m}(\theta_1, \ldots, \theta_m)
\]

where \( f_0 \) is a constant, and the integral of each summand \( f_{h,1,...,j} \) over any of its independent variables is zero. Assuming the input variables are independent random variables uniformly distributed in \([0,1]\) (which can be scaled if they are not), the total variance of model output is defined as

\[
D = \text{Var}[f(\theta)] = \sum_{i=1}^{m} D_i + \sum_{1 \leq i<j \leq m} D_{ij} + \ldots + D_{1,2,...,m}
\]

in which \( D_{h,1,...,j} \) are the partial variances corresponding to the summands in Equation (3). Using the partial variances, we can calculate Sobol’ sensitivity indices. The two most widely used such indices are given below

\[
S_i = \frac{D_i}{D}, \quad i = 1, \ldots, m
\]

\[
S_{T_i} = 1 - \frac{D_{-i}}{D}
\]

where \( S_i \) is called the first-order sensitivity index of the \( i \)-th parameter, \( \theta_i \), on model output, \( D_{-i} \) represents the average variance resulting from all parameters except for \( \theta_i \), and \( S_{T_i} \) represents the total effect of \( \theta_i \) on model output. Obviously, when there is no interaction between \( \theta_i \) and other parameters, the high-order interaction terms disappear and \( S_{T_i} \) is equal to \( S_i \). The Sobol’ indices are known to be robust descriptors of the sensitivity of a model to its input parameters because they do not assume linearity or monotonicity in the model (Sudret 2008). However, Sobol’ indices are computationally expensive to calculate. Even if we are only interested in obtaining the first-order and total Sobol’ indices, a total of \( N \times (m + 2) \) runs are still required, where \( N \) is the number of Monte Carlo realizations. In this study, we resort to cloud computing, in particular, the ModflowOnAzure platform, to help expedite the global sensitivity analysis without having to port the Windows-based MODFLOW code to UNIX-like platforms. In addition, the MATLAB code that is written to generate the input files does not need to port to the Azure cloud to avoid license issues.
3.2. Alternative conceptual models and risk analysis for groundwater sustainability

In addition to parameter uncertainty, which can be evaluated by using the previous described global sensitivity analysis method, additional uncertainties come from incomplete definitions of the conceptual model structure, or spatial and temporal variations in hydrologic variables (e.g. streamflow conditions in the future are uncertain). Accordingly, the most rigorous way to address hydrologic uncertainty and reduce model bias is to examine many plausible realizations. There is a growing tendency among groundwater modelers to postulate alternative models for a site. Neuman and Wierenga (2003) described various situations in which multiple models are needed. The most often encountered situations are different models of alternative descriptions of groundwater processes and interpretations of hydrogeological data (Ye et al. 2010).

To address future streamflow recharge uncertainty, a stochastic streamflow model is usually created based on historical data and then produces synthesized data for the future. For example, for the case study in Arizona, a stochastic recharge model was developed based on over 70 years of historical streamflow data (Shamir et al. 2005). To account for groundwater flow model uncertainty, the streamflow realizations were translated through six plausible alternative conceptual groundwater flow models (ACMs) – optimized by inverse modeling techniques – for each hypothetical pumping scenario for the Arizona case. Thus, a $100 \times 6$ ACM-ensemble is comprised of 600 realizations. Each realization was simulated for 100 years and includes seven seasonal stress periods per year, each with 10 time steps. Thus, each ensemble consists of $4.2E6$ groundwater head calculation for every active spatially discretized finite difference cell node.

Each of the six ACMs has positive and negative attributes and, for the purposes of this paper, the outputs were evenly weighted, although more sophisticated model averaging techniques could be used (Rojas et al. 2010). It has been assumed that each ACM is an approximation of the truth and that (1) a true model does not exist and cannot be expected in the set of models; and (2) as the number of observations increases the weighting scheme may be modified accordingly; these ideas are consistent with Poeter and Anderson (2005). Depending on numerical solver and convergences criteria, and computer specifications (i.e. CPU and computer memory), it takes about eight hours to simulate a typical 600-realization ensemble – via a DOS batch-file for the Arizona use case (Nelson 2010). In the past water resource managers and researchers have faced numerous challenges in running predictive model scenarios. This has in part been due to the processing time required to run the model scenarios, availability to the technical infrastructure required to run the various types of models, and the inability of non-modelers to adjust key parameters and review the resulting changes in a timely manner. A cloud-based on-demand groundwater uncertainty analysis service developed in this paper can enable a wide range of stakeholders to run large complex models quickly and easily.

4. Architectural design and implementation of ModflowOnAzure

A Windows Azure-based integrated groundwater uncertainty analysis platform has been created that can significantly reduce the computing time and lower the barrier
for the cloud adoption in the groundwater uncertainty analysis community. The design and implementation of ModflowOnAzure are described below.

4.1. The design principles of ‘Scientific Modelling as a Service’

We aim to develop a general framework that supports ‘scientific modelling as a service’ (SMaaS), where modeling analysis can be carried out in the cloud as a service (a type of SaaS). However, unlike business SaaS applications, scientific modeling is much more complex in terms of heterogeneity of tools and license issues, and computational and data intensity. There are several design principles that we consider in this paper, which are not only very essential to the groundwater uncertainty analysis community, but also are applicable in the general geoscience community. Below we describe these sets of principles:

(1) Seamless integration between desktop and cloud environment: Because the majority of the existing groundwater modeling analysis using MODFLOW is performed in the Windows Desktop environment, it is important to allow a seamlessly conjunctive use of both desktop tools and the cloud computing service. This is also true for many other water resources-modeling tools such as the Water Evaluation and Planning tool (WEAP 2011). It is necessary to allow current users to leverage on-demand computing power of the cloud without leaving their familiar desktop environment. Grochow et al. (2010) also proposed a ‘Client + Cloud’ architecture, which is mainly motivated by the availability of Graphical Processor Units (GPUs) in local client hardware for visualization purpose, not by the software constraints we discuss here.

(2) Enabling coexistence of traditional commercial site licenses-based tools and pay-per-use cloud services: As the groundwater-modeling community has been in existence for almost 30 years (the first version of the MODFLOW was released in 1983), there are many commercial GUI-based software programs built around MODFLOW such as VisualMODFLOW (2011). Since setting up a groundwater model is a very complex and time-consuming procedure, these tools provide a valuable means for the researchers. When such GUI-based tools are not used, researchers usually use MATLAB to write scripts to generate input files or post-process the output of the model. As currently there is no per-pay-use MATLAB license in the cloud, it is necessary to allow a coexistence of traditional license-based tools such as VisualMODFLOW or MATLAB and cloud-based parallel model execution services.

(3) Efficient and transparent file and data exchange between the desktop environment and the cloud: Data, including the executable groundwater model and its dependencies such as input files, configuration files, and output files, should be able to transparently move between the desktop environment and the cloud, with minimal manual intervention from the users. Note the cost of data transfers out of Azure to $0.12 per GB in North America and Europe with no cost of moving data into Azure (Whiting 2011). It is necessary to leverage such an uneven two-way data transfer cost policy by minimizing the size of the downloading files while reducing the complexity of sending files to the Azure cloud.
Leveraging the elastic cloud computing platform to provide parallel ensemble execution on-demand with consideration of uncertainty analysis science drivers: The parallel ensemble execution in the cloud can be performed either through multi-threading on a multi-core worker VM node, or by using multiple single-core worker VMs to execute the runs independently, or a combination of both. Although each run of MODFLOW is independent, the driver application can be different depending on different uncertainty analysis methods and scenarios. For example, for a global sensitivity analysis, an external MATLAB code could be driving the execution of thousands of MODFLOW runs. Some advanced coupled optimization and simulation models may require an external genetic algorithm-based application driver (Sun et al. 2012).

4.2. The architecture and implementation of ModflowOnAzure

On the basis of the principles discussed previously, we have designed and implemented a system called ModflowOnAzure. The overall architecture is shown in Figure 2. There are three main parts in the system: Windows Azure application, Dropbox (a non-Azure-based cloud storage and service), and Windows Desktop clients. We describe each part below and summarize how the system works at the end of this section.

Figure 2. Architecture diagram of ModflowOnAzure.
4.2.1. Windows Azure application

Several components are implemented in this part of the system. A Worker role can be instantiated as either a ‘Job Manager’ or a computational worker node at the start-up time when the ModflowOnAzure is deployed in the Azure cloud. By default, the first worker instance is initialized as a Job Manager, which is responsible to listen to the ‘I/O Queue’ where a batch of job requests could be sent from either a desktop client or the Dropbox, create individual MODFLOW job run message in the ‘Run Queue,’ and consolidate, compress, and bundle all output files into one zipped file and send them back to the Blob storage. In addition, the Job Manager also runs a piece of code that monitors the Dropbox folders to detect if any new requests come from the shared Dropbox folders (details are explained in Section 4.2.2). If the Job Manager receives a request, it then generates the number of MODFLOW jobs requested and inserts each individual MODFLOW job run as a single message in the Run Queue so that the rest of the computational worker VM nodes can listen to and run each MODFLOW job. The Job Manager also inserts each piece of information into a job status table in the Azure Table storage so that it can keep track of the status of each run. Each worker node, upon retrieving a job run message from the Run Queue, will download the needed input files, model executables and batch scripts (which are bundled as one zipped file) from the Blob storage to its local VM and start the execution. When an individual run is finished, the worker node sends the output to the Blob storage and updates the job status to ‘finished’ in the Azure Table.

We implemented two levels of parallelization as below:

1. Multicore multithreaded parallelization. This was implemented using .NET 4.0 Task Parallel Library (TPL), specifically, the ‘Parallel.For’ loop provided in the System.Threading.Tasks.Parallel class. For example, for a VM with 8 cores, each instance of the computation worker can, in theory, process 8 ensemble runs in parallel. However, this kind of parallelization was found not to speed up the execution in our study; therefore, we do not use this feature in our case studies. The potential reason for such unsuccessful multi-threaded parallelization could be related to the inefficient fetching, starting, and running multiple independent MODFLOW jobs in a shared virtualized multicore VM. However, we defer a detailed investigation in the future.

2. Master-worker multi-VM parallelization: For this level of parallelization, the messages in the ‘Run Queue’ can dispatch MODFLOW job runs to different worker nodes, resulting in a pleasant parallelization among all the worker nodes.

4.2.2. Integration of Dropbox and Azure for ‘Drop-and-Compute’

A typical scenario is that users often want to use GUI-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 gigabytes (GBs) and users might change them frequently. For example, groundwater modelers often use GUI-based programs such as VisualMODFLOW to generate the streamflow realization boundary conditions. We use the Dropbox (2011) open source APIs (Sharpbox 2011) to synchronize the file transfer among users’ desktop and the Azure Blob storage. Previous work (Cottam 2010) has demonstrated using Dropbox to
submit jobs to Condor, a computational grid, but no literature has been found to use Dropbox in a public cloud computing platform. A user only needs to drag-and-drop a zipped file, which contains input files and model executables with a configuration file indicating number of runs and relative file path name to a batch script, to a shared local desktop folder. The ModflowOnAzure system will then automatically fetch the files into the Azure Blob storage and trigger the ensemble runs in the cloud. Once the execution is completed, the output files will be automatically compressed and put back into the output folder in the user’s Dropbox output folder. The actual computation and data synchronization are completely transparent to the user, which significantly reduce the barrier for adopting cloud computing in the groundwater modeling and analysis field.

The mechanism to implement this Dropbox integration with Azure consists of usage of two Dropbox accounts. One is used as the ModflowOnAzure system account, which can be directly monitored by the Job Manager in the cloud. A user’s Dropbox account then needs to share folders (one input folder and one output folder) with the ModflowOnAzure system’s Dropbox account so that ModflowOnAzure can monitor a personal Dropbox account. At this moment, ModflowOnAzure can only support one user Dropbox account at a time. Future extensions to monitor multiple users’ Dropbox accounts are possible. The implication of integrating Dropbox with Azure goes beyond facilitating an easy-to-use desktop environment, but also opens doors for mobile devices-based on-demand model execution, since Dropbox provides clients in many popular mobile devices such as iPhone, iPad, Android, and Blackberry. This allows groundwater modeling uncertainty analysis to be conducted on-site when needed.

4.2.3. Windows desktop clients

In the global sensitivity analysis case, a user usually writes a MATLAB code to generate the needed input files for the Monte Carlo parameter analysis. Because MATLAB code requires a valid license to use and currently it does not provide a cloud-based pay-per-use license, it is necessary to continue to use the MATLAB code in the desktop environment, where a valid license can be used while still accessing cloud computing power. We implemented a generic approach that wraps a complied MATLAB executable using C# code in the desktop environment and then directly sends these dynamically generated input files with associated MODFLOW executables to the Azure Blob storage as well as messages to the I/O Queue to trigger the parallel execution of MODFLOW runs in Azure. This automation is complementary to the ‘Drop-and-Compute’ Dropbox integration described in the previous section, where input files are already generated as static files, and further improves the usability of ModflowOnAzure for a variety of groundwater uncertainty analysis scenarios. The output files can then be retrieved automatically back to a user’s desktop through this C# wrapper code, eliminating any manual downloading processing. Given that thousands of files and hundreds of GBs could be exchanged between the desktop and the cloud, such automation is critical for conducting computational and data-intensive groundwater uncertainty analysis.
5. Case studies
We apply the ModflowOnAzure system to two groundwater uncertainty analysis case studies to demonstrate not only the computing power of ModflowOnAzure, but also to illustrate the scientific insights that can be obtained using this system.

5.1. Application of global sensitivity analysis to a regional aquifer in Texas
Texas has seen rapid population growth in recent years and is projected to have continuously increasing water demands to meet municipal, agricultural, energy production, and ecosystem needs. To facilitate state-wide water resources management and drought planning, the Texas Water Development Board has developed a number of groundwater availability models (GAM) for all major aquifers in Texas. Our case study involves a regional GAM developed for the hydraulically connected Edwards Trinity Plateau and Pecos Valley aquifers (ETPV) located in West Texas. The ETPV aquifers cover an area of about 115,000 km$^2$ in the semiarid West Central Texas (see Figure 3).

The ETPV GAM is subject to a number of uncertainties because of limited in situ hydrogeologic observations (Anaya and Jones 2009, Young et al. 2009, Sun et al. 2010, 2012). To guide future refinement of the ETPV GAM, the main objective of this case study is to perform a global sensitivity analysis to identify parameters that contribute most to the output variance. We can then design future experiments and/or field campaigns based on insights gained from this global sensitivity analysis study.

The ETPV model has two layers and each layer is discretized using a 400 × 300 grid with a uniform cell resolution of 1.6 × 1.6 km$^2$. The parameter set consists of

![Figure 3. Map of the interconnected ETPV aquifers (the groundwater wells are in circles).](image)
spatially distributed hydraulic conductivity, specific yield/specific storage, and recharge multiplication factors for different zones, resulting in a total of 24 parameters. The model runs in annual periods from 1980 to 1999. We used the algorithm presented in Saltelli et al. (2008), Section 4.5) to calculate the first-order and total Sobol’ indices. A single model run took about 0.3 min to finish on a Dell Precision T3500 computer. So the total serial running time for a Monte Carlo sample size of 1024 is approximately 133 hours, with a total of 26,624 independent runs of MODFLOW.

5.2. Groundwater risk analysis in Arizona

Our second case study is for a site located in Arizona. The state of Arizona actively manages the water resources in areas that have shown severe groundwater overdraft and have limited groundwater resources. In 1994, the Arizona Legislature created the Santa Cruz Active Management Area (AMA) whose goals are preserving ‘safe-yield’ conditions and ‘preventing long-term declines in local water tables.’ In the Santa Cruz AMA there is no direct access to imported surface water supplies, and the storage capacity of productive aquifers is relatively small (Figure 4).

As a result hydrologic impacts from natural recharge, discharge, and groundwater pumping are sensitive to change. Given the complexity and variability of the system, it is impossible to simulate precise hydrologic conditions in the future. To reduce bias of projective deterministic modeling, a stochastic stream recharge model and a suite of plausible groundwater flow models were developed by the ADWR for planning purposes.

Post-processing programs were developed to analyze the model-output data based on user-defined criteria (i.e. simulated head and flow statistics; statistics based on the difference between basecase simulated heads and planning simulated heads; inter-arrival statistics for simulated heads and flows, etc.). Figure 5 shows a histogram illustrating the simulated head distribution for each time-step of that particular ensemble (600 realizations), near Tubac, Arizona.

Thus, simulated head distributions for different ensembles can be compared for different future stresses on the system, man-made or natural, such as different pumping rates, that is, 15,000 acre-feet/year versus 19,000 acre-feet/year. In this case, the base ensemble (say 15K AF/year) would be executed and the raw output saved; then a second ensemble would be executed, where the only difference would be a new pumping demand. Our cloud computing solution significantly reduces the computing time as well as the barrier to use the cloud.

6. Results and discussion

With two case studies, we were able to not only measure the computational performance of ModflowOnAzure, but also able to produce results that are useful for the groundwater uncertainty analysis. Below we discuss our results from these two perspectives.
6.1. Computational performance

For the Texas case study, we deployed the ModflowOnAzure on 19 small VM workers (among which, one was a Job Manager, and the remaining 18 nodes were computing VMs) on Azure with the C# client code driving a MATLAB code to dynamically produce the input files needed for the stochastic Monte Carlo simulation of MODFLOW. Each worker VM had a single non-shared CPU and 1.75 GB memory. There were a total of $1024 \times 26$ (i.e. 26,624) output files, which is

Figure 4. Arizona case study site (Santa Cruz active management area).
about 5.64 GB after being zipped using the DotNetZip library (DotNetZip 2011). The total turnaround time (including data transfers in to and out of Azure) for this case study was about 16.85 hours, with a speedup factor of almost 8 (see Figure 6), compared to the original 133 hours of execution time on a single core.

Note that we did not implement the so-called ‘dynamic scaling’ in Azure, where the number of computational workers can be scaled up and down during run time dynamically. Previous study (Hill et al. 2011) already showed dynamic scaling incurs additional penalties such as up to 10 minutes start-up time for a new VM to be ready for use. We also witnessed the uneven execution time for individual run of MODFLOW on each VM, although the overall speedup was satisfactory. The total turnaround time could have been improved if we rewrote the MATLAB code to something that can be directly used in the cloud part of the ModflowOnAzure (such

Figure 5. Distribution of simulated heads near Tubac, Arizona.

Figure 6. Computational performance of ModflowOnAzure for the global sensitivity analysis case study.
as C#) which could eliminate the time to transfer data into Azure. However, since there is no cost to transfer data into Azure, this is a good trade-off in terms of time and cost. More importantly, we think our approach – leveraging existing MATLAB codes on the desktop to drive the dynamic creation of input files – is a more general one, which allows many existing groundwater modelers to easily leverage ModflowOnAzure without changing their preferred programming languages.

For the alternative conceptual modeling groundwater risk analysis in Arizona, we used a similar deployment in the Azure cloud (18 computational small VMs). We then used the Dropbox as a way to trigger the parallel execution of 600 MODFLOW runs. For the Arizona model, each single MODFLOW run takes about 3 minutes on an Azure small VM, and we have been able to observe near-linear speedup. That’s to say, if there are three runs, it still takes approximately 3 minutes if we use three single-core VMs since we can execute them in parallel. If there are six runs, it takes about 6 minutes to finish all runs using three single-core VMs or 3 minutes using six cores. Thus, we were able to finish 600 runs of MODFLOW in about 100 minutes, compared to the original 8 hours or 480 minutes. Note that the model complexity of Arizona’s case is different from that of Texas’s case in this paper, which explains why near-linear speed-up is possible when each individual model run takes longer, thus amortizing the cost of the communication and data transfers between the Azure cloud and the desktop environment. The on-demand computing power offered through ModflowOnAzure significantly improved the groundwater modelers’ capability to quickly and easily conduct their uncertainty analysis.

6.2. Insights obtained from the Uncertainty Analysis

For the global sensitivity analysis, all 1024 result sets (each set in turn includes results from 26 MODFLOW runs) were post-processed in MATLAB. These results can be used to calculate first-order and total Sobol’ indices for different model outputs or performance metrics, such as water budget, spring outflows, boundary outflows and root-mean-square error (RMSE) of model fit. Here we use RMSE as such a metric, which has been used by Tang et al. (2007) in assessing the uncertainty of a distributed, surface hydrological model. The RMSE is calculated using model predicted and observed heads at observation well locations. Figure 7a,b shows the first-order and total Sobol’ sensitivity indices of RMSE for the five hydraulic conductivity zones (Figure 7c) in Layer 1 of the ETPV model. The figures suggest that, among the five hydraulic conductivity parameters, the RMSE measure is the most influenced by Zone 1, a relatively narrow region located in the higher-elevation Pecos Valley region (see lower-left of Figure 7d). This unconfined alluvial aquifer region has more agricultural irrigation activities than the rest of the modeling area, which can lead to large seasonal variations in groundwater well levels. Zone 1 is also one of the regions where the calibrated model gave poor fit. Interestingly, Zone 4, the largest hydraulic conductivity zone in Layer 1, has little effect on the RMSE. Other significant parameters include the recharge factor for the southeast part of the model (not shown). Figure 7b shows that the effect of parameter interactions is only discernible for Zone 1, but not for other zones, which is not surprising given the coarse model structure. The results suggest that future model refinement should consider increasing both vertical and horizontal model resolution for the Pecos Valley region. Also, more observation points should be added for the area. Note that the results presented here rank the relevant contributions of
input parameters to RMSE and are likely to be different for different output variables. In addition to RMSE, in future analysis we may also study the impact of parameters on different output variables such as boundary fluxes.

For the alternative conceptual models study in Arizona, the ensemble distributions provide a more realistic – and less polarizing – result than any individual deterministic solution. For example, probabilistic distributions can be generated for the percentage of years a given streamflow reach may experience future, net gain conditions for different pumping-demand scenarios (Figures 8 and 9). However, individual solutions for different realizations, alternative conceptual models, and forecasted demands can be presented to better understand impacts of capture, induced recharge, and water level fluctuations, while comparing results with historical ranges.

Other possible variables to consider for future stochastic simulations include alternative evapotranspiration (ET) distributions and/or interactive (gaining vs. losing) stream–aquifer boundaries. The ET rate is a relatively large component of the water budget in this area, currently only one areal ET distribution is simulated. In the future, however, it is plausible that different ET distributions could occur, having a
significant impact on the aquifer. In addition, observation data shows that vertical streambed conductance along some reaches, under certain conditions, may be a function of either gaining or losing streamflow conditions. Although an interactive stream–aquifer boundary does not currently exist, its future development (Nelson 2010) may also serve as another alternative conceptualization for projective purposes. Thus alternative ET distributions as well as nonlinear interactive (gaining vs. losing) stream–aquifer boundaries could be added in combination with the existing streamflow realizations and ACMs. However, these additional combinations would greatly increase ensemble computation times. With the near-linear speedup we observed so far, this may be practical in the future.

7. Lessons learned and applicability in the Digital Earth Community

It is evident from this research that easy-to-use on-demand computing for groundwater risk analysis is made possible with our ModflowOnAzure implementation. Below, we discuss several lessons learned in this research, along with some discussion on broad applicability of our implementation in the Digital Earth community.

First, emerging technologies such as cloud computing can be implemented as easy-to-use enabling tools. The Dropbox integration with Azure implemented in this paper significantly reduces the adoption of the cloud and supports automatic data exchange between users’ desktop and the cloud that was not possible before. Although our implementation uses Dropbox and Azure, the idea is applicable to

![Figure 8. Number of Years with Simulated Net Groundwater Discharge Between NIWTP (Rio Rico) and Tubac as a Function of Pumpage (from 11K to 23K AF/YR): 100 × 6 ACM Ensemble.](image-url)
other data exchange scenarios in the Digital Earth and general scientific computing community where multiple devices and clouds can be integrated to enable automatic data exchange for computing and modeling purposes.

Second, many desktop-licensed scientific software can still leverage the power of the cloud with the conjunctive use of both desktop and the cloud environment, as shown in this paper. This paper only presents MATLAB as an example, but other commercial software such as ArcGIS (2012) or TECPlot (2012) can be used in a similar way.

Third, the approaches and technologies used to implement ModflowOnAzure can be used to support many other Digital Earth and geoscience modeling efforts. ModflowOnAzure can be considered as an instance of a ‘Scientific Modelling as a Service.’ Other models such as the Soil and Water Assessment Tool (SWAT 2012), or the Weather Research and Forecasting (WRF 2012) Model can be supported in the similar way. Furthermore, multiple models could be chained together in the cloud, enabling an integrated earth system science modeling.

8. Summary and conclusions

In this paper, we presented our design and implementation of ModflowOnAzure with two groundwater uncertainty analysis case studies that demonstrated the utility of ModflowOnAzure. When designing ModflowOnAzure, we specifically addressed the following challenges related to the lack of seamless integration of desktop tools and Azure cloud: (1) lack of pay-per-use licenses in the cloud for most commercial

![Simulated Groundwater Hydraulic Heads at Tumacacori](image)

Figure 9. Simulated Groundwater Hydraulic Heads at Tumacacori: Base Model (shown) Realization #2 (‘dry’); Years 1–90 Infrequent Flood Recharge; Years 91–100 Frequent Flood Recharge.
software that is routinely used in the groundwater modeling community; (2) lack of a transparent and easy-to-use data/file exchange mechanism between users’ desktop and the cloud. We then implemented a ModflowOnAzure system that realizes a Master-Worker pleasantly parallel computing style for independent ensemble runs of models, a Dropbox integration for transparent data/file exchange between desktop and Azure environment, and Windows Desktop client tools to wrap MATLAB code to drive dynamically generated ensemble runs across the boundary of the desktop environment and Azure. Note that our implementation of ModflowOnAzure is generic, and thus can support other models as long as they can be executed in a Windows operating system in a command line environment.

Our case studies in Texas and Arizona showed significant speedup in terms of computational performance, shortening ‘time-to-solution’ and providing timely insights on the uncertainty facing the groundwater modeling and risk analysis. Texas water planners have been using the Drought of Record (the ten-year drought from late 1940s to late 1950s) as the worst-case scenario. At the time of this writing (i.e. December 2011), the state of Texas is experiencing the worst one-year drought in its history. As the climate variability is projected to intensify, there is a strong need to evaluate or re-evaluate the GAM across the state for different water allocation scenarios. The access to cloud computing can greatly lessen the time required for conducting the scenario analyses and, therefore, offer an invaluable tool for water resources planning.

Similarly, user-friendly modeling interfaces and faster computational speeds will increasingly allow uncertainty analysis to become a more important component of water management in Arizona. In the Santa Cruz AMA, the uncertainty of future stream recharge and imperfect information about the subsurface will necessarily require examination plausible distributions (Nelson 2010). The management of the water resources in the Santa Cruz AMA is extremely challenging given the complexity of the aquifer system, the sensitivity of the aquifer to climatic change, and the needs of the various stakeholders within the AMA. An added challenge to this area is working with the Mexican Government to develop solutions that work for all the stakeholders. Water managers need a system to analyze possible solutions to these complex issues and the effects of those solutions on the whole system. With this information the water managers and stakeholders can make informed decisions.

As the demand placed on water resources has grown so has the complexity of managing those resources. This complexity has required the development of new methodologies for integrating statistical and qualitative uncertainty in water management by providing a platform that could incorporate a wide range of disciplines such as urban development/planning, groundwater/surface water management, and climate change, just to name a few. In the past, water managers have faced numerous challenges in incorporating these factors into running predictive model scenarios. Our ModflowOnAzure platform provides a solid foundation to build upon that provides a wide range of stakeholders the access to run large complex models quickly and easily.

Lastly, it is expected that this research will benefit the broad Digital Earth and geoscience community to effectively use the power of the cloud to accelerate the model execution and support participatory data-driven decision making. The approaches and technologies developed in this research are expected to be applicable
to many other scientific models. The source code of the ModflowOnAzure is available upon request for research and academic usage.

Acknowledgements

The authors thank Wenming Ye from Microsoft Corporation who provided initial technical guidance on running models in the Azure platform at the early stage of our implementation. Ron Searl at the University of Illinois at Urbana-Champaign helped the implementation of this project in Azure. The first author also thanks Dr Yan Xu, Senior Research Program Manager of the Microsoft Research Connections’ Environmental Informatics Program, who funded the ‘Digital Urban Informatics’ project at NCSA. Krishna Kumar at Microsoft provided no-cost access of an MSDN Premium Account on Azure for the computational experiment conducted in this paper. Benjamin Ruddell at the Arizona State University provided the initial suggestion and made connections on using ADWR’s model as a case study for the cloud computing experiment. Lastly, we thank Barbara Jewett at NCSA who provided professional proofreading for this paper.

References


Cottam, I. 2010. How to make it trivial and uniform for users to access Computational Grid resources (such as: Condor) – a gentle ramp for scientists to access e-science gateways [online]. The University of Manchester, Manchester. Available from: http://wiki.myexperiment.org/index.php/DropAndCompute [Accessed 20 December 2011].


Knight, C.G. *et al.*, 2007. Association of parameter, software, and hardware variation with large-scale behavior across 57,000 climate models. *Proceedings of the National Academy of Sciences*, 104 (30), 12259–12264. doi: 10.1073/pnas.0608144104


Zinn, D. et al., 2010, Streaming satellite data to cloud workflows for on-demand computing of environmental data products. Workflows in Support of Large-Scale Science (WORKS), 2010 5th Workshop on 14 November 2010, 1–8. doi: 10.1109/WORKS.2010.5671841