Unlimited Virtual Computing Capacity using the Cloud for Automated Parameter Estimation

Joseph Luchette[1], Gregory K. Nelson[2], Charles F. McLane III[3], Liliana I. Cecan[4]

[1]*McLane Environmental, LLC., Princeton, NJ, jluchette@mclaneenv.com*

[2]*McLane Environmental, LLC., Princeton, NJ, gnelson@mclaneenv.com*

[3]*McLane Environmental, LLC., Princeton, NJ, cmclane@mclaneenv.com*

[4]*McLane Environmental, LLC., Princeton, NJ, lcecan@mclaneenv.com*

ABSTRACT

Methods are emerging for highly parameterized inversion techniques (i.e. SVD-assist available in PEST). In many inverse problems the modeler is limited by the number of computers available for parallelized inversion problems (i.e. Parallel PEST) and the times at which these computers are available. The cost of purchasing and maintaining a cluster of computers can also be prohibitive.

These limiting factors can be circumvented by the use of cloud infrastructures, such as GoGrid (http://www.gogrid.com/) or Amazon EC2 (http://aws.amazon.com/ec2/) to run the parallelized inversion problems. Cloud infrastructures are on-demand services that allow users to instantly boot-up virtual servers out of a “cloud” or “grid” of hundreds, or sometimes thousands, of servers in the repository. These services are typically inexpensive and cost efficient since they are “pay-by-the-hour”. Therefore, the modeler can have as many virtual servers as desired, at the times desired, and can simply shut them off when the simulation is completed.

To demonstrate the ease and functionality of running Parallel PEST on a Cloud infrastructure, two Parallel PEST parameter estimation problems were set up and solved on GoGrid: (1) estimation of local hydraulic properties in an aquifer using pumping test data, and (2) estimation of regional hydraulic properties of an aquifer over an entire basin based on historical water level data.

Use of cloud infrastructures can essentially eliminate the limiting factors imposed by the number of and availability of network workstations for parallel inversion problems by providing scalable, low cost, on demand computational resources.

INTRODUCTION

Methods are emerging for highly parameterized inversion techniques (i.e. singular value decomposition- or SVD-assist available in PEST). In the inverse problems the modeler is limited by the number of computers available for parallelized inversion problems (i.e. Parallel PEST) and the times at which these computers are available. The cost of purchasing and maintaining a cluster of computers can also be prohibitive.

With these issues in mind, we have examined the feasibility of using cloud computing as a tool for parallelized inversion problems. This paper lists several computational issues we have encountered and details the solution that a cloud infrastructure provides. Finally we end the discussion with several PEST examples run on a cloud infrastructure.

PROBLEM DESCRIPTION

Smaller firms or consulting groups engaged in groundwater modeling often face several computing issues as they move towards parallelized inversion / calibration. These issues include:

- Limited number of processors available for parallelization. Analysts may find themselves physically limited by the number of computers in the office, and by inherent hardware limitations (some machines are too outdated to be used in scientific computing).
- Limited time that processors are available for parallelization. A “cluster” of computers may not be dedicated to the user. This would limit the use of machines to evenings and weekends when others are not using them.
- Costs associated with acquiring and maintaining a group or “cluster” of computers available for parallelization.

These limits are acutely felt when a simulation involves a large number of parameters or long model run times. The model requirements and resource limitations outlined above define the need for scalable computing power. Cloud computing provides a flexible and cost efficient solution by making available to the analyst the equivalent of a large number of machines, when desired and for the desired amount of time.

CLOUD COMPUTING

What is it?

The term “cloud computing” is used in many different contexts and has evolved through several paradigms which can make it a confusing term. In general, it refers to data, software applications and/or computer processing power made accessible as an online service. We’ll look at the term in its’ two parts to help clear up any misconceptions about the term cloud computing. The term “cloud” refers to a collection of interconnected computers in some form of a grid or network. Those computers may be individual servers, clusters of servers or virtual machines (aka, “nodes”) virtualized by hypervisor technologies like VMware or Xen. Clouds are usually given the illusion of being infinite in resources (Armbrust et al, 2009). The second part of the term cloud computing is where the confusion sets in because of the many different ways clouds can be utilized. The types of services offered using clouds can usually fall into one of three categories: 1 - A software application; 2 - A server infrastructure and 3 – A platform (Figure 1).

1. Software-as-a-Service (SaaS): SaaS makes otherwise traditional software available as an online utility. For Instance, one could perform word processing and spreadsheet tasks using an online service like Google Docs instead of purchasing and installing Microsoft Office or installing an Open Office package on a local machine. Those documents would be available everywhere you go with access to the internet. Webmail is probably the most common and the simplest example of SaaS. This same concept can be applied to any traditional software, including modeling packages.

2. Infrastructure-as-a-Service (IaaS): Cloud infrastructure services provide users with the capability to provision new servers instantly via a web interface or Application Programming Interface (API) enabling them to build vast IT infrastructures using traditional IT methods (i.e. networking and load balancing), without the large capital expense that one would otherwise incur by physically purchasing the hardware and software. Most of these services aim their marketing at online service/application providers or e-businesses to maximize their ability to scale quickly to increased website traffic. Amazon EC2, GoGrid, The Rackspace Cloud, (formerly Mosso) are all providers of cloud infrastructure services.

3. Platform-as-a-Service (PaaS): Cloud Platforms are often marketed for developers because they enable one to build application(s) in one or more specific programming languages and deploy those applications on the service provider's cloud which will enable the application to scale automatically as needed. For instance, two current major Cloud Platforms are Windows Azure, and Google App Engine. Windows Azure allows users to write their applications in .NET. Google App Engine allows users to write their application in Java and Python. The model of this service is perfect for those seeking entry into a SaaS market. Prior to cloud services, one would have to invest heavily in the IT infrastructure required to run such a business.

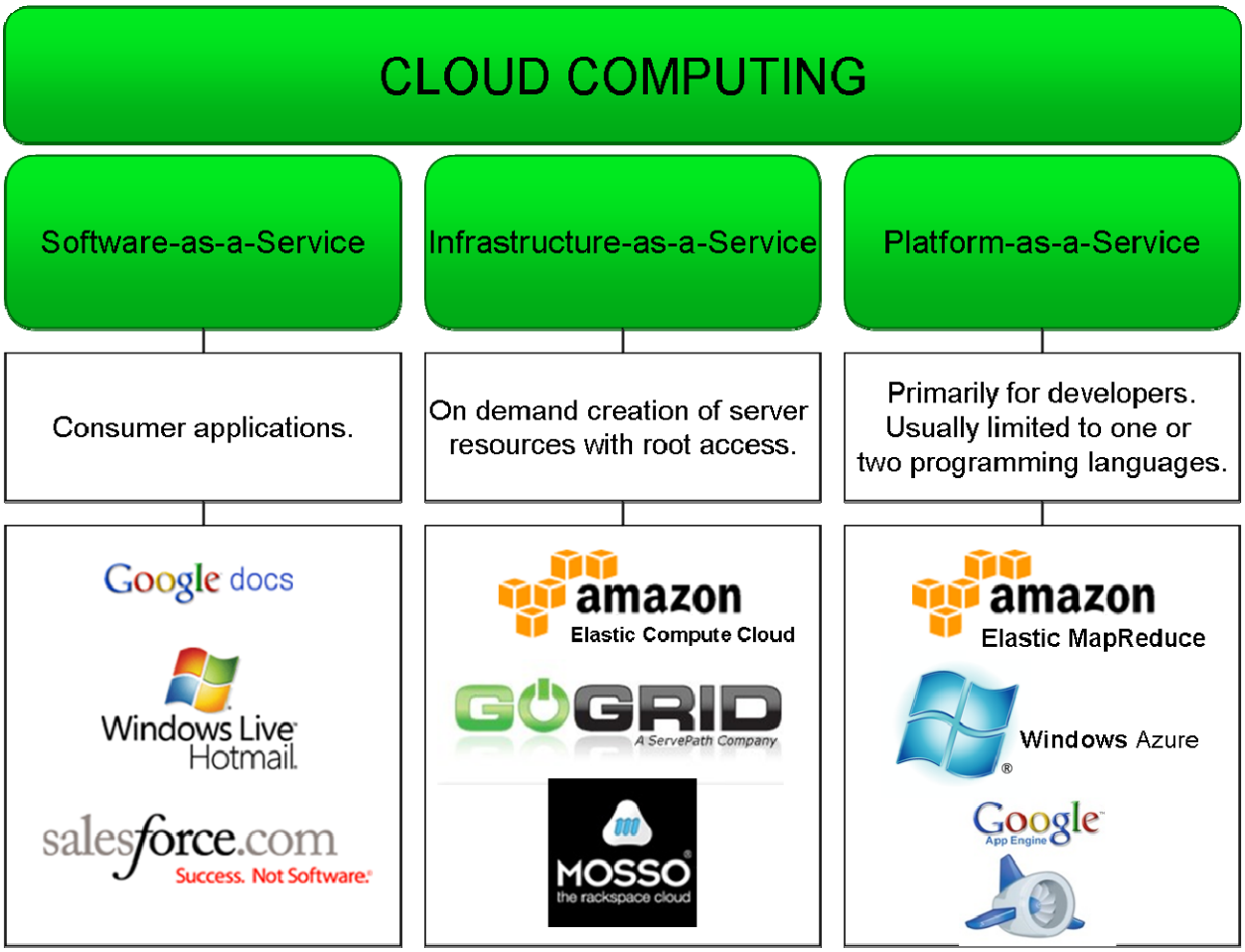

Figure 1: Types of services offered by Cloud Computing with brief explanations and examples of each service.

In the examples in this paper, we use the term cloud computing to refer to an "Infrastructure-as-a-Service" because we are provisioning fully virtualized Windows servers on demand upon which we are capable of running PEST in serial or in parallel. However, all of these service

types have potential for computationally intensive modeling. For example, the vast resources of cloud platforms could be used to host scientific modeling applications as a SaaS. Imagine logging onto a website, entering parameters, uploading your data and clicking a run button. You would get an email notification hours (or days later) when you model run in complete with a link to download the results. The SaaS applications would have to be built upon the architecture made available by the platform, but cloud computing platforms offer unique opportunities for large analytical processing jobs that analyze potentially terabytes of data (Armbrust et al, 2009). One existing example of a service that utilizes a cloud infrastructure is the Amazon Elastic MapReduce Service. Amazon's Elastic MapReduce Service, which utilizes the Hadoop framework, "lets you focus on crunching or analyzing your data without having to worry about time-consuming set-up, management or tuning of Hadoop clusters or the compute capacity upon which they sit" (http://aws.amazon.com/elasticmapreduce/). The Hadoop framework is an open-source framework which allows programmers to easily operate parallel code execution across hundreds of Cloud Computing servers (Armbrust et al, 2009). By hosting the Hadoop framework on EC2, Amazon has effectively created a Cloud Platform which can be used to run modeling applications.

Costs

One of the most appealing features of cloud infrastructure services is the inexpensive utility based (aka "metered") pricing model. This allows users to allocate large amounts of computing resources for variable lengths of time, without ever incurring the up front capital expense of purchasing the equipment. Pricing models are typically based on server RAM and/or CPU capacity and billed by the hour. The cost per hour currently is typically in the range of $0.08 to $1.20 per hour depending on the RAM and CPU capacity per server being utilized. Additional pricing for data transfer is also applied, but is marginal (typically in the range of $0.10 to $0.50 per GB per month) unless you are operating a web server experiencing very high traffic. Optional software may or may not cost extra as well – for instance Amazon EC2 charges more per hour if you are running an instance of Windows Server (as opposed to a Linux or Unix based machine) and even more if you are using SQL Server Standard (this can be anywhere from $0.125 to $3.20 per hour depending on the server RAM and CPU capacity).

Flexibility/Limitations/Reliability

The flexibility, reliability and limitations for cloud infrastructures depend upon both the options provided (such as Operating System, CPU resources, RAM, disk space, data transfer speeds) as well as the services provided (i.e. network speed and reliability, resource availability, support)

as laid out in the Service Level Agreements (SLA). SLAs vary from service to service, but typically guarantee a certain amount of compute and/or storage resources will be available. In some cases you can pay for “reserve” resources to assure they will be available without paying the full usage charges of the resources. SLAs may also have certain money-back guarantees for their infrastructure such as network availability. It is recommended that one look closely at the SLAs when researching different cloud infrastructure services.

Until recently, the Windows operating system was not available on cloud infrastructures. These services were limited to Linux and Unix based machines. This is no longer the case, as you have a plethora of options for operating systems on most cloud infrastructure services. These options often include Microsoft Windows Server 2003, Windows Server 2008, CentOs, Redhat, OpenSolaris, Fedora, Ununtu Linux, and more.

Depending on the service, you may have the option of different CPU speeds (or an equivalent measurement thereof, since there is no actual physical CPU to measure for a single virtual machine). For instance, Amazon EC2 has “standard” and “High CPU” instances which they measure using what they call “EC2 Compute Units”. Also, the option of using both 32 and 64 bit platforms offers a significant boost in computing power.

Each service will typically allow the user to select a variable amount of RAM for each virtual machine as well. One can have as little as 512 MB or up to 15 GB of RAM on a 64-bit platform.

Disk storage space is usually offered as a set amount for a single virtual machine and can vary from 100 GB to over 1,000 GB. Further, most cloud infrastructures have very affordable storage services to supplement their compute services. For instance, Amazon Simple Storage Service (S3) easily allows you to centrally store large amounts of data accessible from your virtual machines, and GoGrid has a service called Cloud Storage which allows the same.

Data transfer speeds vary from service to service and depend largely on the robustness of the service network. The SLA will typically state what the service provider presents as acceptable and unacceptable for network performance standards and what the cost recovery is to the customer if the service provider fails to meet those standards.

Setup and Use

Many cloud infrastructures have online point and click user interfaces for starting and stopping servers which allows the user to easily acquire and then terminate the required computing power. They also have Application Programming Interfaces (APIs) which allow programmers to make calls to the service from code.

Most services will offer “root access” to a virtual machine once it is created. This means that the user can log into the machine as they would a desktop machine. This is done via Remote Desktop Protocol (aka RDP) if using Windows or Secure Shell (SSH) for Linux or Unix based systems. From there, the user can upload files and setup software applications as they would on a desktop machine.

There are various techniques for connecting virtual machines in a network on a cloud infrastructure. For instance, GoGrid provides the user with both a public and private IP address range and each server comes with three network interfaces (NICs), one of which is private. Using the private IP addresses, one is able to build computing networks on the cloud upon which they can run parallel computing applications, such as PEST.

EXAMPLE RUNS

In this section, two examples of running PEST to estimate groundwater model parameters in a cloud environment are illustrated. The first example involves the use of a simple serial *PEST* run used to determine aquifer characteristics from a hypothetical pumping test. In serial processing, each computer has a function and the functions are performed linearly (Figure 2). This simple run was executed to verify that it would be possible to run PEST on a virtual machine in a cloud computing environment.

The second example involves the use of a *Parallel PEST* run to estimate regional aquifer characteristics for a hypothetical regional aquifer. In Parallel processing multiple CPUs or processor cores simultaneously process the program (Figure 2). This second run was used to show that a cloud computing environment could be used for parallel PEST inversion. Each of these hypothetical examples is described in the following sections.

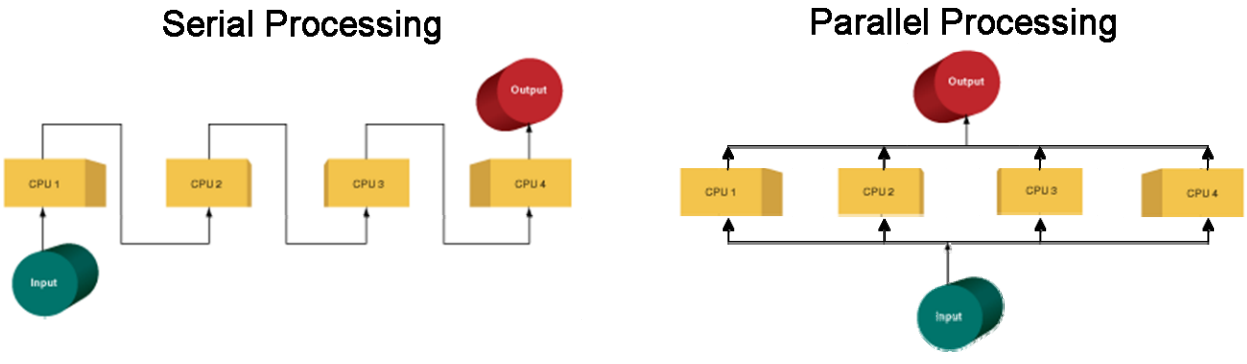

Figure 2: Serial Processing and Parallel Processing diagrams.

Serial Cloud Computing - Pumping Test

Figure 3 shows a cross section of an unconfined, homogeneous aquifer. Figure 3 also shows the locations of a partially penetrating production well (on the far left) and of five partially penetrating observation wells. The USGS program WTAQ (Barlow and Moench, 1999) was utilized to estimate drawdown at the production well and at the observation wells. WTAQ is based on an analytical model of axisymmetric groundwater flow in a homogeneous and anisotropic aquifer (Moench, 1997). The aquifer characteristics presented on Figure 3, and in the Original Model column of Table 1 represent the assumed values of the aquifer and will be the values that the PEST results will be compared to.

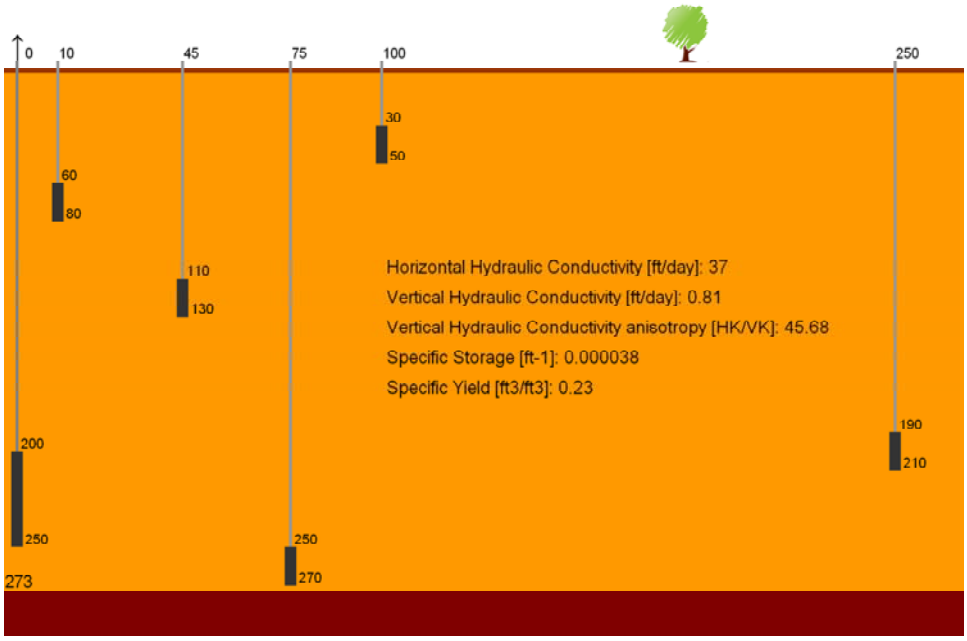

Figure 3: Cross section showing WTAQ unconfined, homogeneous, anisotropic aquifer model. The values at the top of the figure represent radial distance from the pumping well. The values within the cross section along the wells and at the top of the aquitard (brown) are depths below ground surface.

.

Table 1: WTAQ model aquifer parameters			
Parameter	**Original Model**	**Initial**	**Estimated**
Horizontal Hydraulic Conductivity [ft/day]	37	10	37
Vertical Hydraulic Conductivity [ft/day]	0.81	0.1	0.81
Vertical Hydraulic Conductivity anisotropy [Kh/Kv]	45.7	100	45.7
Specific Storage [ft-1]	0.000038	0.00001	0.000038
Specific Yield [ft3/ft3]	0.23	0.25	0.23

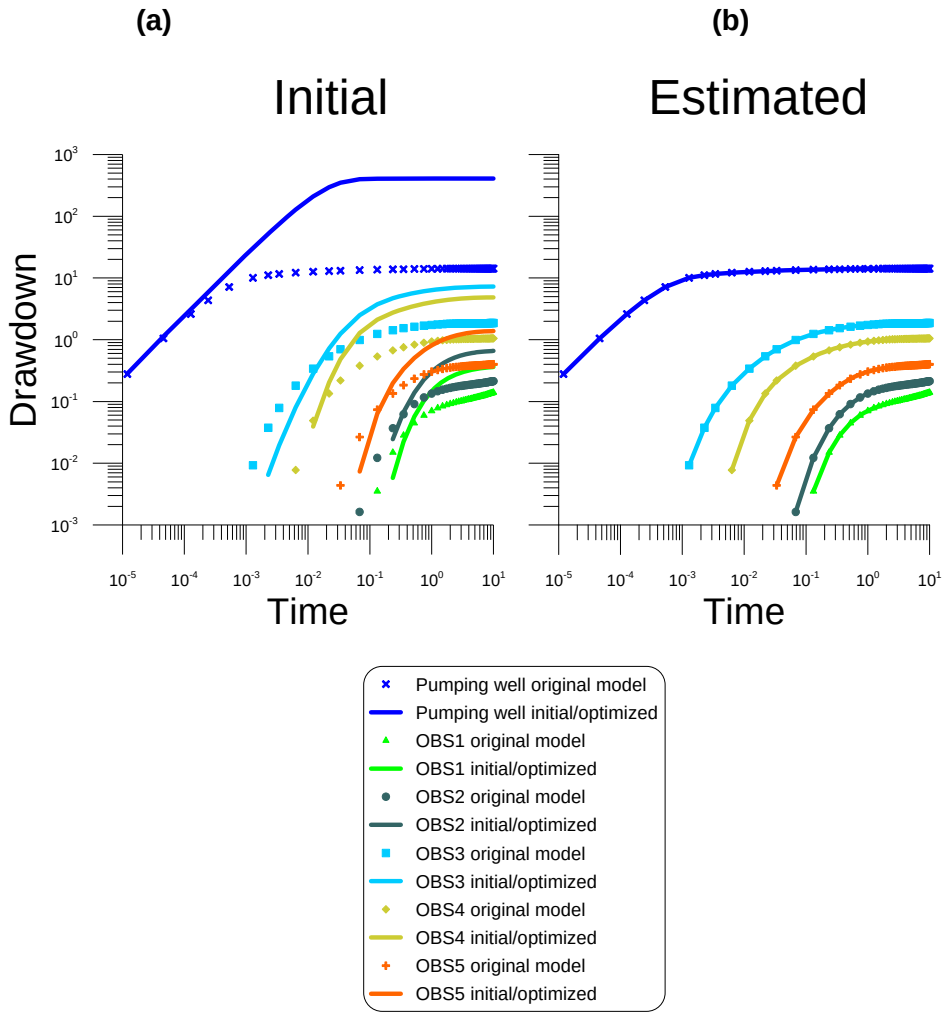

Figure 4: a) Model drawdown data plotted with the calculated drawdown using the initial aquifer characteristics, and (b) Model drawdown data plotted with the estimated aquifer characteristics.

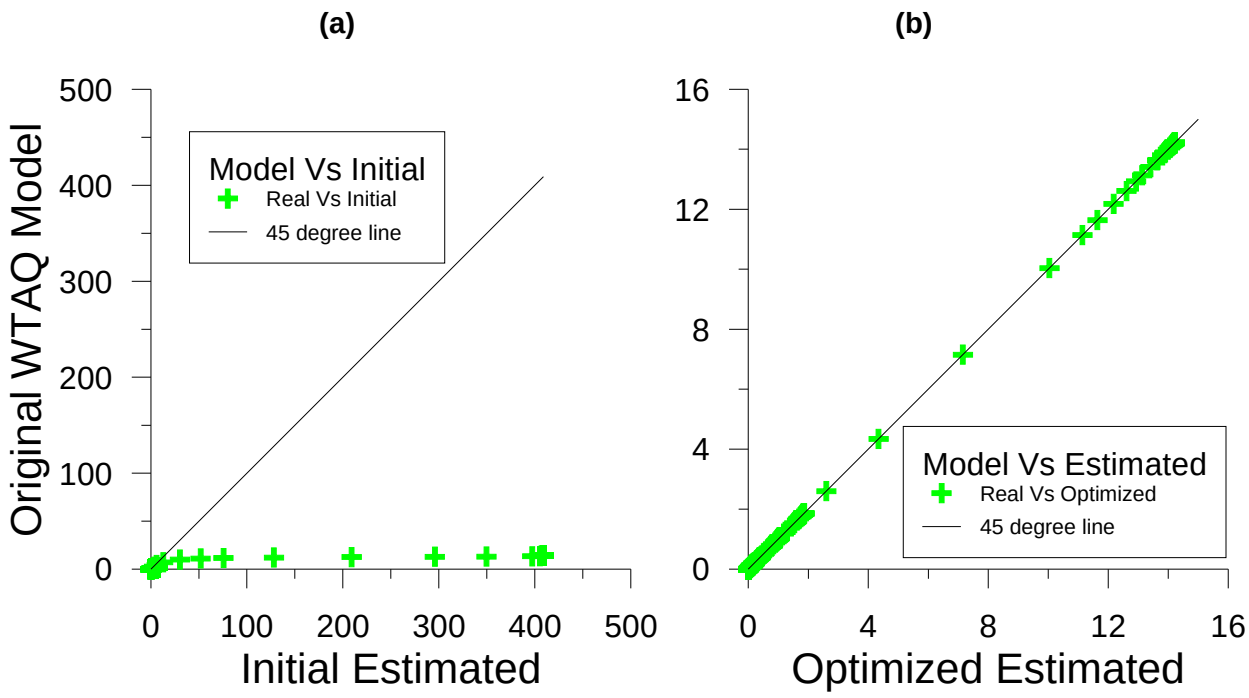

Figure 5: a) WTAQ model's drawdown versus the initial estimate; b) WTAQ model's drawdown versus the PEST optimized model's predicted drawdown.

The PEST run was loaded onto one virtual server, running Windows Server 2003 with 1 GB of RAM, where it ran for approximately 2 minutes. The PEST estimated aquifer characteristics are, for this simple problem, essentially identical to the original model values (Table 1).

Figure 4a presents the original WTAQ model's drawdown versus the drawdown calculated using the initial aquifer characteristics, and Figure 4b shows the original WTAQ model's drawdown versus the PEST optimized model's estimated drawdown.

Figure 5a, presenting the Model vs. Initial drawdown, shows the poor results generated from the initial estimates of aquifer characteristics. Figure 5b, presenting the Model vs. PEST estimated drawdown, shows a good calibration of the model since the data is falling along the 45° line.

For this very simple case, it has been shown that a virtual machine running on the cloud could be used for parameter estimation. This example was straightforward, with the only setup time being that required to copy the needed model files onto the cloud.

Parallel Cloud Computing - Regional Model

For this example, a fictional, relatively simple regional groundwater flow model was created in MODFLOW 2000 (Harbaugh et al., 2000). The model consists of 5 layers with hydrogeologic zonation within each layer (Figure 6). Table 2 the actual or model values of the various aquifer characteristics for the model. The bottom of the model was essentially a no-flow boundary while the top of the model received recharge. The boundaries along the sides of the model were constant flux boundaries. A river was represented in the model as a series of river cells extending from north to south in Layer 1.

Parameter values within the model being used in PEST were changed to the initial values (highlighted values) listed in Table 3. A comparison of water level elevation values in the original WTAQ model versus the initial model (Figure 7a) shows the poor results estimated using the initial parameter values (Table 3). Figure 7b shows the excellent calibration of the PEST estimated water level elevation data and Table 4 presents PEST estimated parameters or aquifer characteristics.

The PEST run was performed using four virtual servers running Windows Server 2003 with a Xeon processor running at 3.00 GHz with 1 GB of RAM each. Optimization runtime was approximately 9 hrs. This runtime of 9 hrs is comparable to a non-cloud computing PEST runtime of 10 hrs on a desktop computer with a Xeon processor running at 2.66 GHz with 3 GB of RAM. The cloud runtime was shorter due to faster virtual servers being used. The model was run 1,568 times over 30 iterations in the cloud optimization. The average runtime of each model run was approximately 1 ½ minutes. The total cost of cloud computing resources used in this model optimization was less than $5.00.

This example further demonstrates that cloud computing can be used for inverse modeling, and model calibration.

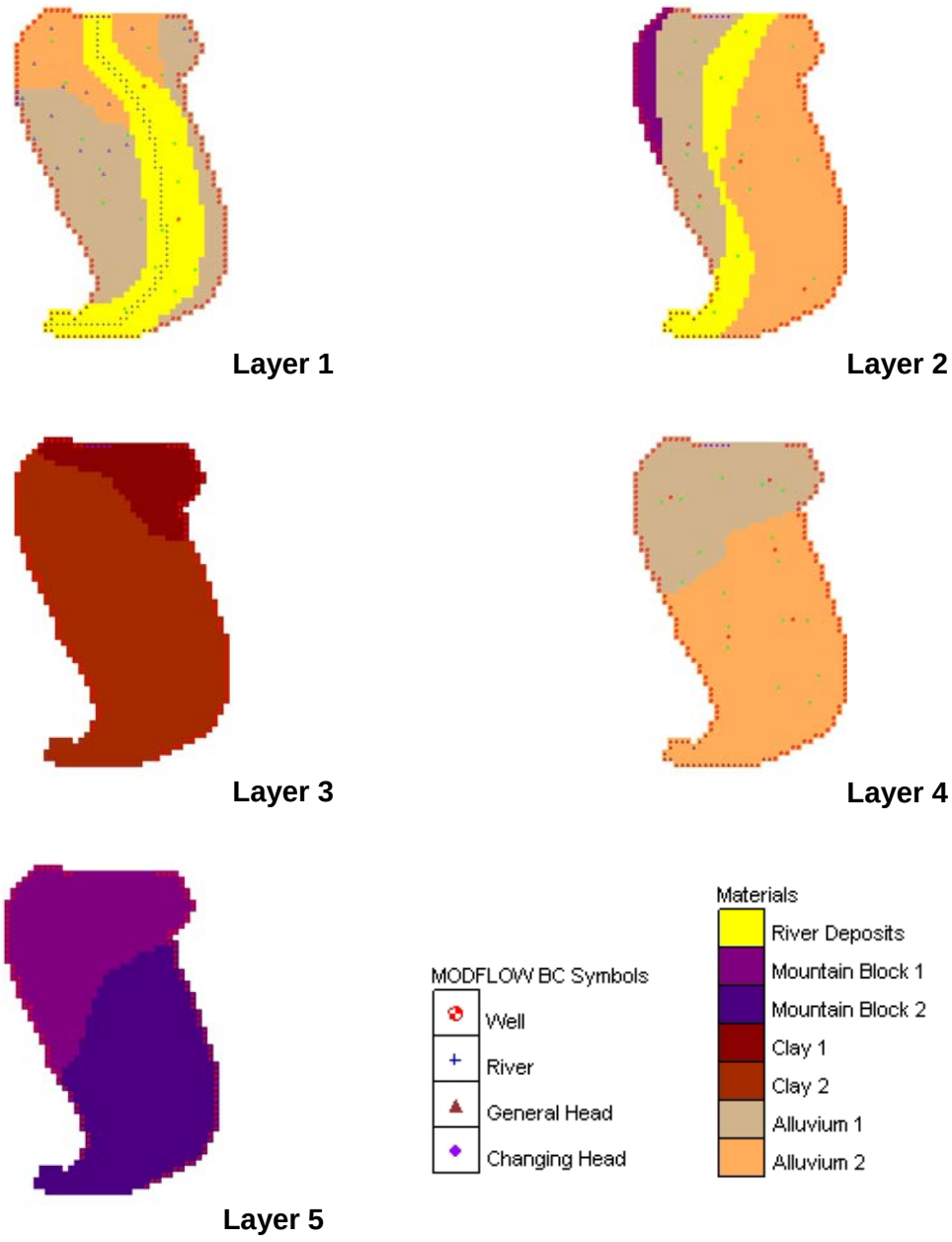

Figure 6: Layering and zonation within the regional MODFLOW model.

Table 2: Model values of the aquifer characteristics for the MODFLOW model							
	River Deposits	**Mountain Block 1**	**Mountain Block 2**	**Clay 1**	**Clay 2**	**Alluvium 1**	**Alluvium 2**
Horizontal K	100	0.0001	0.001	0.003	0.008	12	52
Vertical K	5	0.0001	0.0001	0.00003	0.00005	0.5	13
Horizontal anisotropy	1	1	1	1	1	1	1
Vertical anisotropy	20	1	10	100	160	24	4
Specific Storage	0.000045	0.00001	0.00001	0.00006	0.00002	0.00008	0.0001
Specific yield	0.32	0.05	0.1	0.4	0.05	0.19	0.28

Table 3: Initial values of the aquifer characteristics for the MODFLOW model that were used in PEST							
	River Deposits	**Mountain Block 1**	**Mountain Block 2**	**Clay 1**	**Clay 2**	**Alluvium 1**	**Alluvium 2**
Horizontal K	50	0.0001	0.0001	0.001	0.001	50	50
Vertical K	5	0.00001	0.00001	0.00001	0.00001	5	5
Horizontal anisotropy	1	1	1	1	1	1	1
Vertical anisotropy	10	10	10	100	100	10	10
Specific Storage	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001
Specific yield	0.2	0.1	0.1	0.2	0.2	0.2	0.2

Note: Highlighted values indicate those parameters that were allowed to vary.

Table 4: PEST estimated values of the aquifer characteristics for the MODFLOW model							
	River Deposits	Mountain Block 1	Mountain Block 2	Clay 1	Clay 2	Alluvium 1	Alluvium 2
Horizontal K	100.3	0.00016	0.00018	0.0013	0.0014	12.0	52.5
Vertical K	5.1	0.000026	0.000036	0.000012	0.000029	0.499	11.3
Horizontal anisotropy	1	1	1	1	1	1	1
Vertical anisotropy	19.65	6.14	5.08	102.12	48.38	24.03	4.66
Specific Storage	0.0000001	0.000003	0.000003	0.000003	0.000004	0.000081	0.000100
Specific yield	0.153	0.100	0.100	0.200	0.200	0.194	0.228

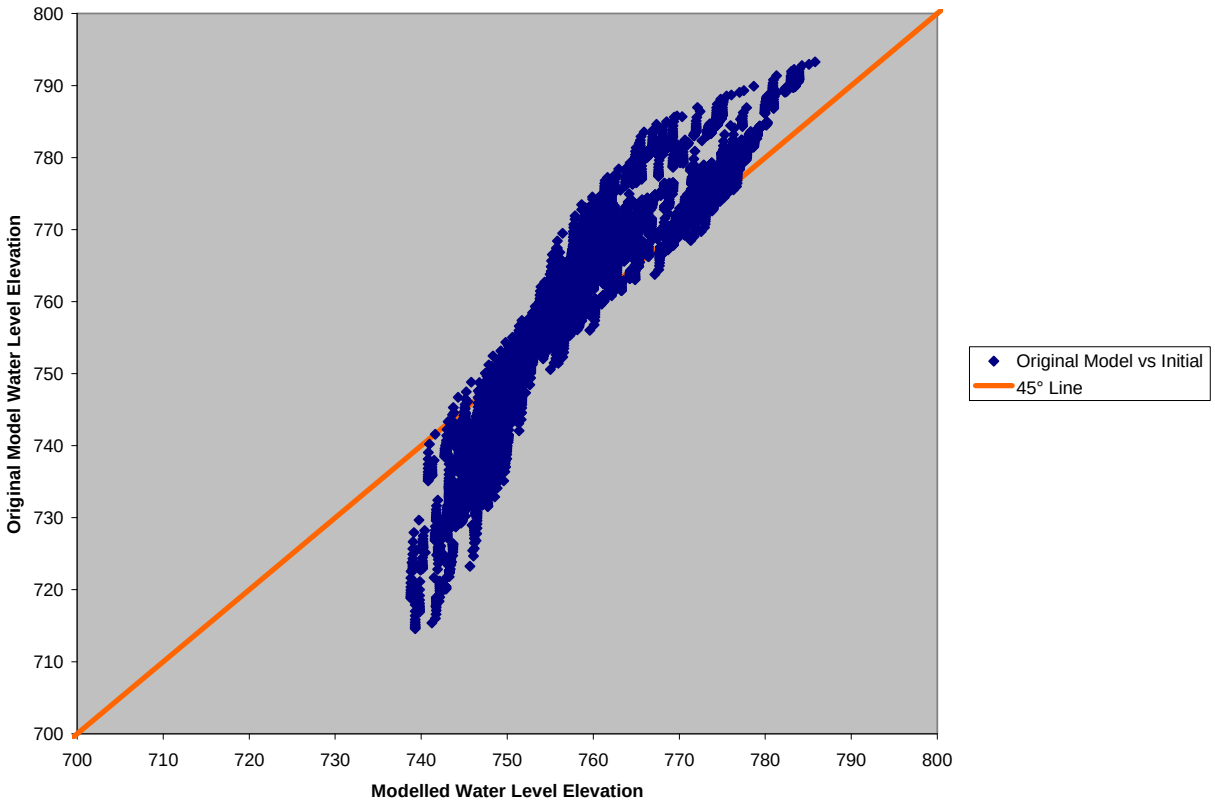

Figure 7a: Plot of the original water level elevation vs. the initial modeled water level elevation.

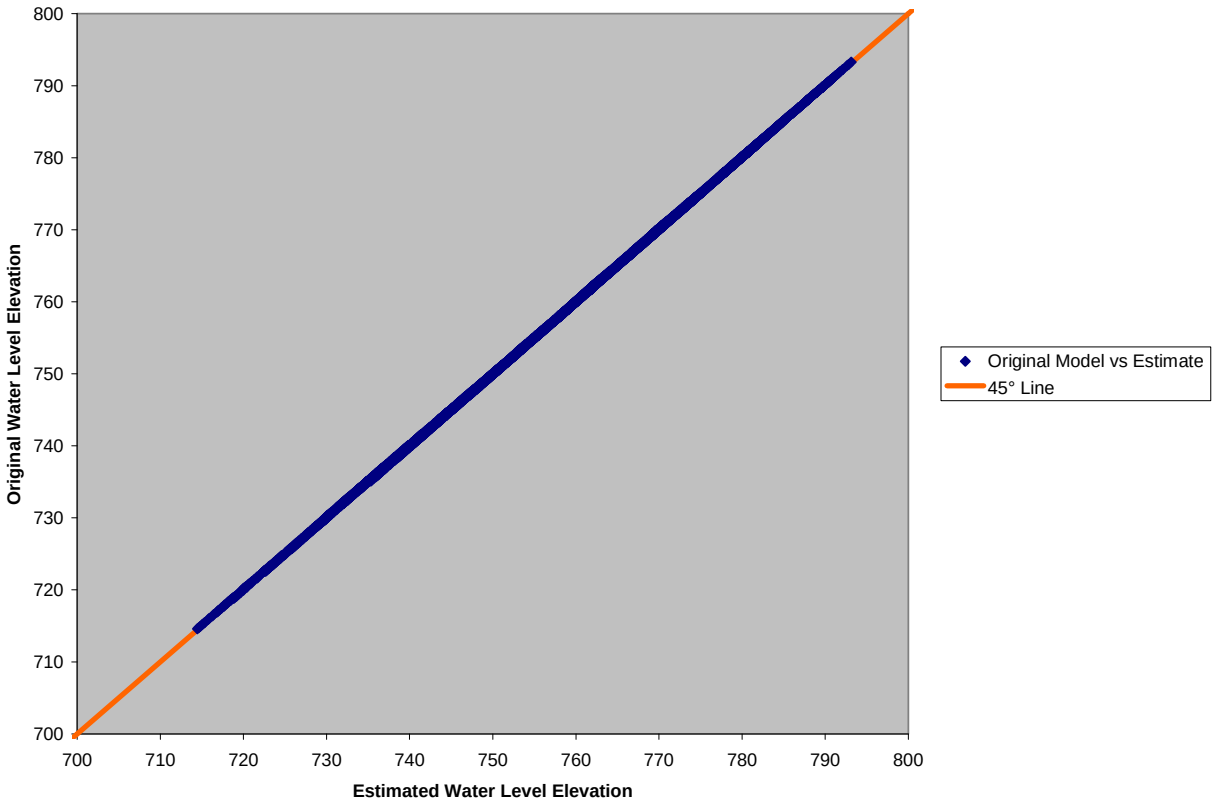

Figure 7b: Plot of the original model water level elevation vs. the PEST estimated modeled water level elevation.

CONCLUSIONS

Two PEST parameter estimation problems were set up and solved on GoGrid: (1) estimation of local hydraulic properties in an aquifer using pumping test data, and (2) estimation of regional hydraulic properties of an aquifer over an entire basin based on hypothetical, historical water level data. These examples demonstrate that PEST parameter optimization can be performed using virtual machines on cloud computing infrastructures.

Use of cloud infrastructures can essentially eliminate the limiting factors of number of, and availability of, network workstations for parallel inversion problems by providing scalable, low cost, on demand computational resources. Analysts are no longer physically limited by the number of computers in the office and by inherent hardware limitations. Therefore they are able to use a seemingly unlimited number of very capable virtual machines to decrease model runtimes. Further, the analysts are not limited by the time that processors are available for

parallelization as is typically experienced in a small to mid-size office network setting. Therefore they can run the models at any time of day.

Running our second example on the GoGrid Cloud cost approximately $5.00 for running 4 virtual machines running for approximately 9 hours. This runtime of 9 hrs is comparable to a non-cloud computing PEST runtime of 10 hrs. The cloud runtime was shorter due to faster virtual servers being used. The computation time could have been further decreased by utilizing more robust CPU resources and/or more virtual machines in parallel.

REFERENCES

Barlow, P.M. and A.F. Moench, 1999. WTAQ — A Computer Program for Calculating Drawdowns and Estimating Hydraulic Properties for Confined and Water-Table Aquifers. U.S. Geological Survey Water-Resources Investigations Report 99-4225.

Armbrust, M. et al, 2009. Above the Clouds: A Berkeley View of Cloud Computing. UC Berkeley Reliable Adaptive Distributed Systems Laboratory - http://radlab.cs.berkeley.edu/.

Harbaugh, A.W., Banta, E.R., Hill, M.C., and McDonald, M.G., 2000, MODFLOW-2000, The U.S. Geological Survey Modular Ground-Water Model - User guide to modularization concepts and the ground-water flow process. U.S. Geological Survey Open-File Report 00-92.

Moench, A.F. 1997, Flow to a well of finite diameter in a homogeneous, anisotropic water table aquifer: Water Resources Research, v. 33, no. 6, p. 1397-1407.