Jun 18th, 2:00 PM - 3:20 PM

Characterizing Spatial Random Fields through a Bayesian Inverse Modeling Framework and the High Throughput Computing Software - HTCondor

Carlos Osorio-Murillo
Brigham Young University, carosocali@gmail.com

Daniel P. Ames
Brigham Young University

Heather Frystacky
University of California - Berkeley

Yoram Rubin
University of California - Berkeley

International Environmental Modelling and Software Society (iEMSs)
7th Intl. Congress on Env. Modelling and Software, San Diego, CA, USA,
Daniel P. Ames, Nigel W.T. Quinn and Andrea E. Rizzoli (Eds.)
http://www.iemss.org/society/index.php/iemss-2014-proceedings

Characterizing Spatial Random Fields through a Bayesian Inverse Modeling Framework and the High Throughput Computing Software - HTCondor

Carlos Osorio-Murillo[1], Daniel P. Ames[1], Heather Frystacky[2], Yoram Rubin[2]
[1] *Brigham Young University - Department of Civil and Environmental Engineering, Provo, Utah, United States*
[2] *University of California Berkeley, Department of Civil and Environmental Engineering, Berkeley, California, United States*
carosocali@gmail.com

Abstract: Spatial Random Fields (SRFs) are used to characterize the behavior of variables which are difficult to measure everywhere. In the case of hydraulic conductivity used in groundwater models, heterogeneity can be modeled through structural parameters of the random fields – parameters that define the field's spatial distribution. Characterization of these structural parameters using a stochastic inverse method requires the evaluation of thousands of potential realizations of the SRF which is a time consuming and challenging task – reducing the adoption of stochastic models. To address this issue, this paper presents the integration of high throughput computing using HTCondor with the MAD# framework – an uncertainty characterization tool that uses the Method of Anchored Distributions (MAD). MAD# is coupled with HTCondor allowing users to search for the optimal location of anchors – statistical devices located in the field – using the parallelization capabilities of HTCondor. As expected, using this approach, the simulation processing time decreases as the number of instances (CPUs) used in the HTCondor Network increases. Also, weekend results show marked improvement over weekday results likely due to fewer interruptions and reassignment of processing tasks between nodes. This demonstration and implementation of an SRF characterization process in a parallel environment shows potential for broader use of the method in environmental modeling.

***Keywords**: Monte Carlo; Inverse Modeling; Random Fields; HTCondor; MAD*

1 Introduction

The high cost of collecting measurements for characterizing Spatial Random Fields (SRFs) is an incentive to evaluate more accurate techniques that integrate additional resources such as models, secondary information, and related variables. Existing information can contribute to the reduction of the uncertainty in the SRFs. Using Inverse Modeling (IM) techniques, it is possible to extract information from simulation or other "forward" models to improve parameter estimation. Although both deterministic and stochastic IM techniques have been introduced, stochastic methods can be more appropriate for SRFs since they do not require following strict Gaussian assumptions usually required in deterministic models. The Method of Anchored Distributions (MAD) is a stochastic IM method focused on characterizing the uncertainty of SRFs (Rubin, Chen, Murakami, & Hahn, 2010). MAD is implemented in the software framework MAD# (Osorio-Murillo, Over, Ames, & Rubin, 2014).

MAD# uses forward models that relate a variable of interest to measurable variables as a means for characterizing the uncertainty at specific locations called "anchors" – statistical devices located in the SRF domain. The inversion process requires evaluating conditional SRFs in a forward model multiple times. The number of evaluations is proportional to the number of observations included in the process. The strategy to localize anchors in the domain depends on characteristics of phenomena and measurements (Yang, Over, & Rubin, 2012). The most time-consuming step in the execution of MAD# is the evaluation of each conditional SRF in the forward model. Although MAD# can run the process in multiple cores, large projects could require multiple computers. MAD# addresses this issue submitting the simulation to a high throughput computing (HTC) environment such as the HTCondor system, which is a software application that shares the unused computing resources of a network. This paper presents how MAD# improves the simulation time depending on the number of nodes in the HTCondor environment and also depending on day of the week.

1.3 Inverse modeling in High Performance Computing

HPC environments have been used extensively in recent years to solve inverse problems in different scientific areas (Goncharsky & Romanov, 2013) (Bertshinger, 2001). The management of large meshes in simulation requires large memory and CPU capacity. The grid dimension in MAD# projects depends on the size and complexity of the forward model and its required inputs. The number of simulations also depends on the number of observations used in the inversion process. MAD# does not improve the performance of the forward simulation model code itself. Rather, MAD# runs multiple instances of the forward model in parallel using a Monte Carlo simulation approach. Monte Carlo simulations are easily parallelized in stochastic models (Barry, 1990). In groundwater modeling, SRFs represent the structure and distribution of geological materials, which are used for representing variability at different resolutions. The high resolution representation of SRFs requires more computational resources (Tompson, Falgout, Smith, Bosl, & Ashby, 1998). Stochastic IM applications use large computational resources; hence HPC is a logical choice for running these applications. However, the adoption of IM techniques would then depend on the accessibility of HPC resources. HTC access is more readily available through technologies such as HTCondor, which allows users access to a large amount of unused computational cycles in a computing infrastructure. Monte Carlo applications have been successfully tested in this platform (Zhou & Mascagni, 2000), which indicates that it is an appropriate technology for testing IM technologies.

1.4 Method of Anchored Distributions and MAD#

The Method of Anchored Distributions (MAD) (Rubin, Chen, Murakami, & Hahn, 2010) uses a Bayesian approach to characterize SRFs. Pre-defined anchors and the structural parameters of the spatial distribution capture global and local variability. Equation 1 shows the Bayesian approximation used to obtain the posterior distribution of the structural parameters and anchors.

$$p(\theta,\vartheta|z_a,z_b) \propto p(\theta,\vartheta|z_a)\, p(z_b|\,\theta,\vartheta,z_a) \qquad \text{Equation 1}$$

Where *p* indicates a probability density function (pdf), θ structural parameters and ϑ anchors. $p(\theta,|z_a)$ is the joint prior distribution of the structural parameters and anchors conditional on the Type-A data vector z_a, and $p(z_b|\,\theta,\vartheta,z_a)$ is the likelihood of observing the Type-B data vector z_b given the structural parameters, anchors and Type-A data. Finally, $p(\theta,\vartheta|z_a,z_b)$ is the joint posterior distribution of the structural parameters and anchors conditional on both Type-A and Type-B data.

MAD# is an extensible IM software framework that implements MAD theory for uncertainty characterization of SRFs. This software application works with a driver approach to support multiple forward models and random field generator applications. MAD# generates configuration files to execute the inversion process in a nested Monte Carlo process (Figure 1a). The prior information (Equation 1) provides the samples that will be evaluated in a first Monte Carlo. Each sample is used to generate multiple conditional realizations using the random field generator. The realization is evaluated in the forward model and the forward model output is stored in a database.

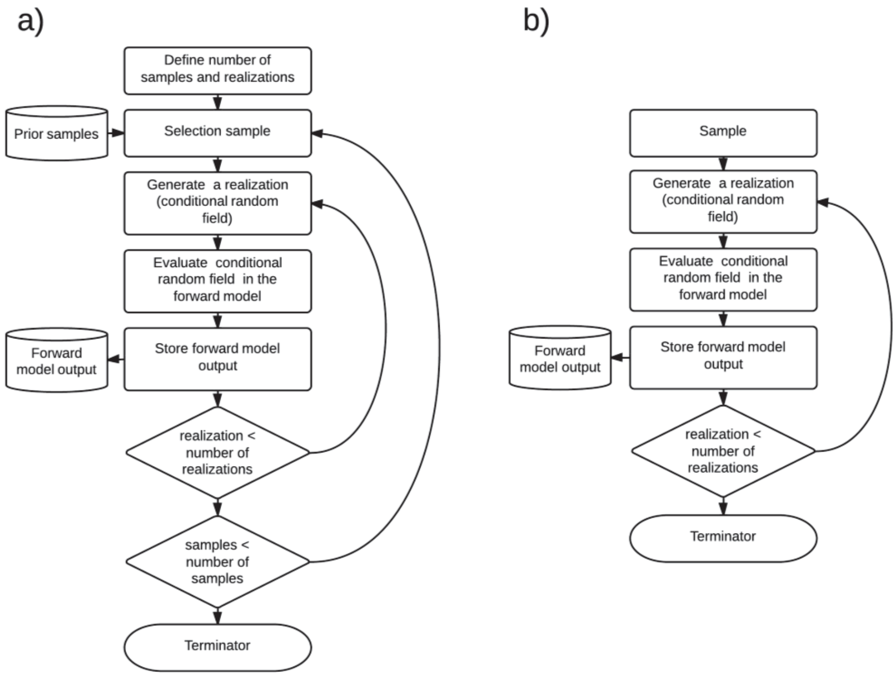


Figure 1 MAD# Monte Carlo process

1.5 HTCondor

The University of Wisconsin-Madison developed a software system called HTCondor that administers and accesses the unused computational resources from a computing infrastructure. HTCondor provides several mechanisms to identify idle workstations candidates to execute jobs. A flocking system allows deployment of jobs in other groups of computers (pools). When a workstation is working on a job, this job can be interrupted by user activity. Then, HTCondor transfers this job to another node in the network. The scheduling process is executed according to the characteristics requested by users like available CPU cycles, memory and operative system. HTCondor works in a Master-Worker schema, where the master node controls the worker nodes and the worker nodes execute the jobs. Each node can also be configured to be able submit jobs in the network..

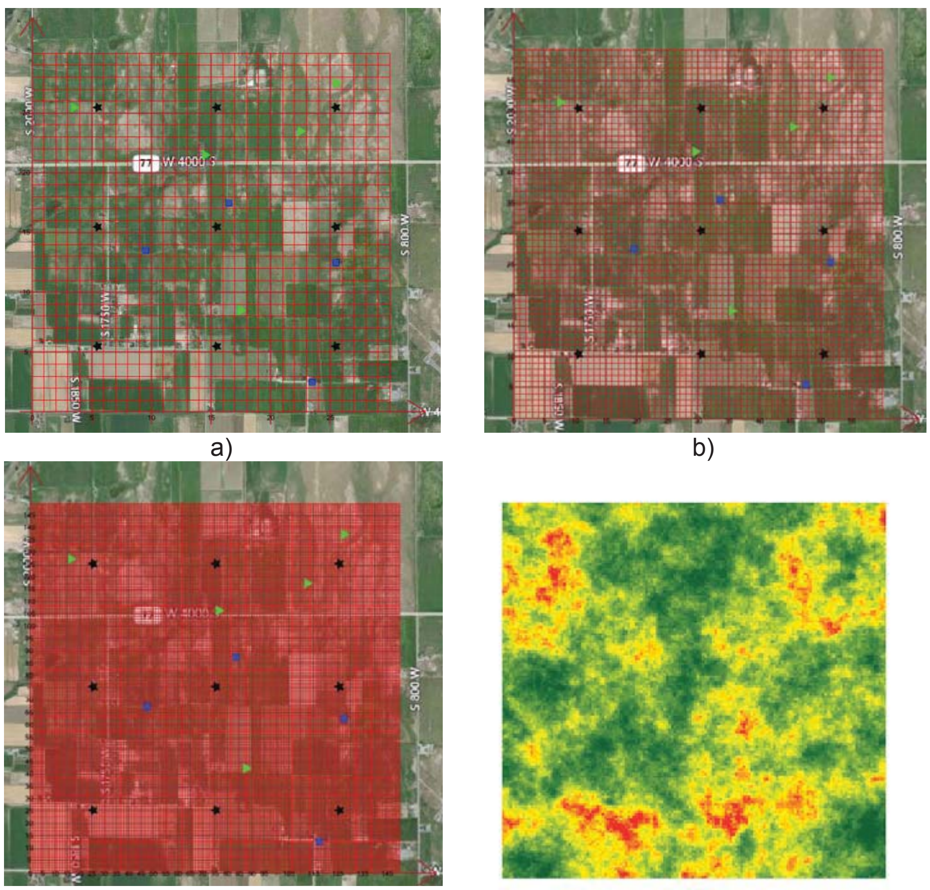
The execution environment of HTCondor allows users to submit jobs in several so called "universes." The universes provide mechanisms to control the communication between the worker nodes. The distributed approach used by MAD# is the universe "vanilla"; although this universe does not provide information on how the jobs are been executed. The output of each job can be transferred using a mechanism for transferring files enabled in this universe. The vanilla environment allows one to submit software packages that can be executed in each node. The MAD# core executable is transferred using this method.

MAD# stores the forward model output per sample; this is a convenient for the parallelization of the Monte Carlo process. Figure 1b shows the process that will be executed in each node of HTCondor. It is clear that is necessary to transfers the random field generator application and the forward model to each node to complete the simulation per sample.

2.0 Method

The efficiency of HTCondor working with MAD# was evaluated using three case studies with different grid densities (Figure 2abc). The common objective of each of these MAD# case studies is to characterize the Log-Transmissivity at the anchor locations. For each case study, the Type-A target variable is Log-Transmissivity and Type-B data are head pressure. The forward model Modflow 1996 (Chiang & Kinzelbach, 2001) is used to relate the Type-A measurements to the Type-B variable. Each Modflow project was configured as steady-state with two boundary conditions at 100m and 90m north and south respectively. Gstat (Pebesma & Wesseling, 1998) was used to generate conditional random fields. The true field in each synthetic dataset is the same as is shown in Figure 2d. The true field was resampled to fit the density grid of each project. The number of anchors and measurements are the same in each case. Additionally, the test cases are evaluated in two scenarios where the status of the HTCondor network is different:

- Weekend Scenario: Computers are fully available.
- Week Scenario : Computer availability can change at any moment.



a) b) c) d)

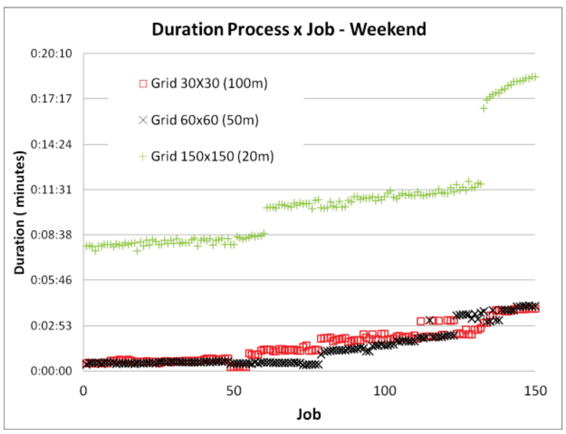
Figure 2 Project grids: a) 100m (30x30), b) 50m (60x60), c) 20 m (150x150). Green triangles: Type-B measurements, Blue squares: Type-A and Black stars: Anchors. d) SRF (True field) Transmissivity.

The global variability of the SRF is defined by an exponential model with the following structural parameters: range, partial sill and mean with values of 300m, 0.15, and -4 respectively. The MAD theory allows one to determine the posterior distribution of an anchor when the structural parameters are known. Equation 2 shows the anchor posterior distribution $p(\vartheta|z_a, z_b)$ conditioned to Type-A and Type-B data which is a simplification of the Equation 1. The prior distribution of anchors is just conditioned to Type-A data $p(\vartheta|z_a)$.

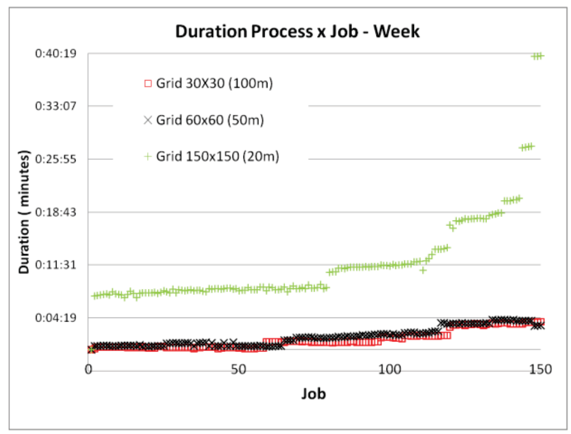
$$p(\vartheta|z_a, z_b) \propto p(\vartheta|z_a)\, p(z_b|\, \vartheta, z_a) \qquad \text{Equation 2}$$

The number of samples evaluated is 300 and the number of realizations per sample is 30, which means that Modflow will run 9000 times. It is not necessary to check the convergence or specific quality of the solution because our objective is determining the time-consumption of the simulation process using the HTCondor infrastructure. Our HTCondor network was divided between two pools of Windows operating system computers with 72 and 243 nodes, respectively.

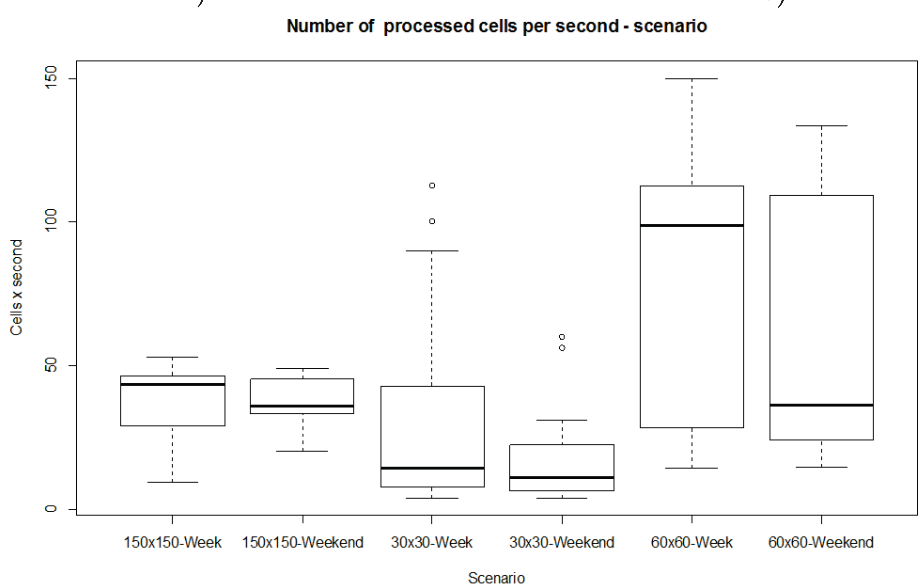
In each test case, 150 jobs are submitted with two samples per node. The HTCondor scheduling process first assigns the jobs in the pool with 72 nodes. When there are not available nodes, the second pool is used using the HTCondor flocking process. Figure 3ab shows the duration time of all jobs in the three test cases per scenario. Using the indicator number of processed cells per second (PCs/S), the performance of the execution of HTCondor among test cases is evaluated (Figure 3c). The variability of the indicator PCs/S in the density grid 60x60 is higher than other cases. The 150x150 grid shows the lowest variability in the indicator PCs/S. This can be explained by the longer duration of this test case in comparison with the other test cases. Although the amount of information transferred for each node is the same in all test cases, the latency can contribute in the variability of the test cases with small sizes (Cai, Judd, Thain, & Wright, 2014).



a)



b)



c)

Figure 3 a) Duration jobs in the Weekend scenario b) Duration jobs in the tipical Week c) Comparison of test cases using indicator processed cells/second in each project

The total duration of the simulations for the 30x30 grid and 60x60 grid in both scenarios were similar (Table 1). The 150x150 grid test case in the weekday scenario had the maximum duration, where some jobs were three times higher than other jobs --noting that the last eight jobs were re-assigned to other nodes (Figure 3b). This was produced by a suspended mechanism of HTCondor, which suspends jobs when a user takes control of a computer.

The duration mean of the jobs in each test case is compared in both scenarios through a significance test, where the null hypothesis considers the duration mean is equal for jobs executed in both scenarios. Table 1 shows that the duration mean is different rejecting the null hypothesis in all cases with a standard significance level of 0.05, which indicates that the execution of the MAD# projects can be affected by the state of the HTCondor network. We observed that the PCs/S indicators were lower in the test cases of the weekend scenario but with lower variability.

We compared the total duration of the MAD IM simulation in the HTCondor network with the estimated total simulation time of a single computer with 8 cores. Table 1 shows the duration statistics and the ratio between total time in HTCondor and approximate duration in one computer. When the grid density is larger the ratio increases. The 150x150 grid computation completed 41 times faster than a single computer in the weekend scenario. Conversely, the total duration in an weekday was 19 times faster. The performance reduction was produced because some computers were claimed by users; the reassignment of processes increases the time twice in the last eight nodes as is shown in Figure 3b.

Table 1 Duration simulation

Evaluation	Total duration one computer (hours)	Total duration (min)	Ratio HTCondor/one computer	Mean(min)	Variance	p-value<0.05
30x30 weekend	0.9	3.97	13.61	1.70	1.18	
30x30 week	0.9	3.83	14.09	1.42	1.52	0.04*
60x60 weekend	1	4.15	14.46	1.46	1.48	
60x60 week	1	4.10	14.63	1.80	1.52	0.02*
150x150 weekend	12.9	18.68	41.43	10.71	9.22	
150x150 week	12.9	40.05	19.33	11.10	14.78	0.02*

The characterization of the anchor located in the center-top of all projects describes the Log-Transmissivity very close to the true value. This anchor captures the information of the measurements because its neighbors are Type-B and Type-A measurements. Figure 4 shows the posterior distributions of all test cases obtained with HTCondor in the weekend scenario.

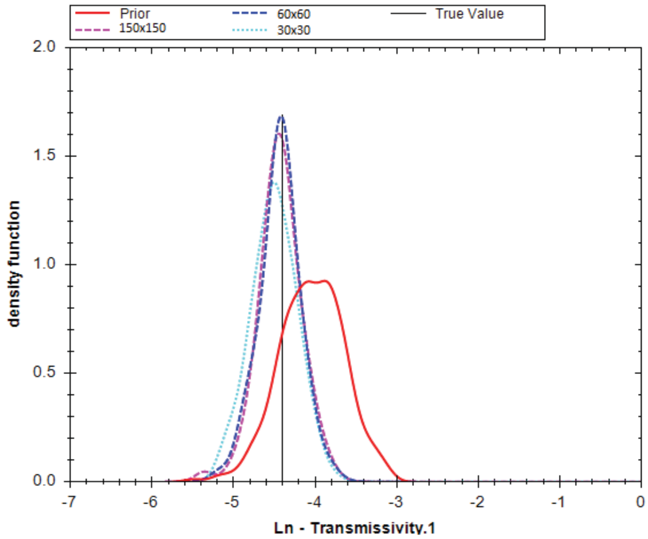


Figure 4 Posterior distributions anchors in the Top-middle in each project

3.0 Conclusions

HTCondor showed a considerable improvement in the total duration of the Monte Carlo process in MAD#. The high variability in the duration time in the small cases studies (30x30 grid and 60x60 grid) is produced by the network latency. The effect of the latency can be reduced by increasing the simulation time per node, which can be achieved by processing more MAD# samples per node. The HTCondor "vanilla" environment was used to submit MAD# jobs, allowing MAD# core to run in each node and to transfer the simulation output files to the master node. Although, the transfer file system works adequately, it is necessary to evaluate the Hadoop File System (HDFS) to reduce the high variability in the response of small MAD# projects. The test case projects with different density grids allowed us to verify the response of different sizes grids – noting that the time improvement increases dramatically with the grid size.

The availability of idle computers in the HTCondor network changes the duration mean and total duration of the execution of MAD#. Although, the main characteristic of HTCondor is the management of idle computers, the reassignment of jobs decreases the performance of the simulation. It is recommended for large projects to use a time-frame with more availability.

REREFENCES

Barry, D. A. (1990). Supercomputers and their use in modeling subsurface solute transport. Reviews of Geophysics , 28 (3), 277-295.

Bertshinger, E. (2001). Multiscale Gaussian Random Fields and Their Application to Cosmological Simulations. The Astrophysical Journal Supplements .

Cai, Y., Judd, K., Thain, G., & Wright, S. (2014). Solving Dynamic Programming Problems on a Computational Grid. Comput Econ .

Chiang, W.-H., & Kinzelbach, W. (2001). 3D-Groundwater Modeling With Pmwin: A Simulation System for Modeling Groundwater Flow and Pollution. Springer Verlag.

Goncharsky, A., & Romanov, S. (2013). Supercomputer technologies in inverse problems of ultrasound tomography. Inverse Problems .

Osorio-Murillo, C., Over, M., Ames, D., & Rubin, Y. (2014). MAD#: A Framework for Inverse Modeling and Uncertainty Characterization. Environmental Modeling and Software(in second revision) .

Pebesma, E., & Wesseling, C. (1998). Gstat: a program for geostatistical modeling, prediction and simulation. Computer & Geosciences , 24 (1), 17-31.

Rubin, Y., Chen, X., Murakami, H., & Hahn, M. (2010). A Bayesian approach for inverse modeling, data assimilation, and conditional simulation of spatial random fields. Water resourses reseach , 46 (W10523).
Tompson, A. F., Falgout, R. D., Smith, S. G., Bosl, W. J., & Ashby, S. F. (1998). Analysis of subsurface contaminant migration and remediation using high performance computing. Advances in Water Resources , 22 (3), 203-221.
Yang, Y., Over, M., & Rubin, Y. (2012). Strategic placement of lacalization devices (such as pilot points and anchors) in inverse modeling schemes. Water Resources Researcg , 48.
Zhou, M., & Mascagni, M. (2000). The cycle server: a Web platform for running parallel Monte Carlo applications on a heterogeneous Condor pool of workstations. 2000 International Workshops on, (pp. 111-118). Toronto, Ont.