Computers & Geosciences 48 (2012) 334–336

Contents lists available at SciVerse ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Short note

Introduction of a web service for cloud computing with the integrated hydrologic simulation platform ParFlow

Claudius M. Bürger [a,1], Stefan Kollet [b,*], Jens Schumacher [a], Detlef Bösel [c]

[a] *Department of Geosciences, University of Tübingen, Germany*
[b] *Meteorological Institute, University of Bonn, Germany*
[c] *R&H Environmental Ltd., Nürnberg, Germany*

ARTICLE INFO

Article history:
Received 19 July 2011
Received in revised form
22 December 2011
Accepted 10 January 2012
Available online 3 February 2012

Keywords:
Cloud computing
Web service
Hydrologic modeling
Supercomputing
Optimization
Visualization

1. Introduction

In cloud computing, software and data are shared via servers that can be accessed on-demand through basic terminals in conjunction with a Web browser. This results in the efficient utilization of software and hardware infrastructure by multiple users without the need of local software installation and maintenance. Here, this concept is applied and extended to the integrated hydrologic simulation platform ParFlow via a newly developed Web interface.

Until recently there was no unifying definition of cloud computing in the literature to our knowledge. A generic definition in Wikipedia referred to cloud computing as the "provision of computational resources on demand via a computer network" (where resources encompassed software and hardware). In September 2011 the U.S. National Institute of Standards (NIST) finalized its definition of cloud computing (Mell and Grance, 2011) recognizing three different service models: software-as-a-service (SaaS), infrastructure-as-a-service and platform-as-a-service (PaaS). Essential characteristics of cloud computing are On-demand self-service, Broad network access, Resources pooling, Rapid elasticity and Measured service. Well-known examples are the commercially available services by Google (e.g. Googlemail, Googlemaps as SaaS) and Amazon (EC2 as PaaS).

In science, cloud computing has been adopted quite rapidly (see Evangelinos and Hill, 2008; Fienen et al., 2011 for PaaS applications with Amazon's EC2). As a matter of fact, high-performance computing (HPC), i.e. the remote utilization of large-scale interconnected computer resources by a diverse user base, can be seen as a special case of cloud computing. However, applications in "The Cloud" are intuitively associated with on-demand computational resources *and* user interfaces as well as Web services that ensure – appropriate data preparation usually being a prerequisite – a large degree of user-friendliness without limiting software and hardware capabilities.

To date literature on potential and limitations of cloud computing in science is scarce. While web-based database services are relatively abundant in hydrologic sciences (e.g. Kruger et al., 2007; CUAHSI initiative; Kanwar et al., 2010) recent articles by Sterling and Stark (2009) and Hunt et al. (2010) highlight the prerequisites that must be met for cloud computing becoming an integral part of science in connection with HPC, which are in part also relevant for this project. In the opinion of these authors, the main strengths of cloud computing stem from e.g., on-demand utilization of functional, upgraded software and

* Corresponding author. Tel.: +49 228 735195; fax: +49 228 735188.
E-mail addresses: claudius.buerger@uni-tuebingen.de (C.M. Bürger), stefan.kollet@uni-bonn.de (S. Kollet).
[1] Tel.: +49 7071 2973173; fax: +49 7071 295059.

doi:10.1016/j.cageo.2012.01.007

hardware resources without the disadvantage of initial financial investments and costs related to maintenance and upgrades of the system. On the other hand, important requirements of cloud computing in HPC environments are associated with e.g., parallel scalability, and system security and dependability.

While the rationale of this project is based on some of the thoughts presented above, the objective is not to address all the pertinent issues related to cloud computing in scientific applications. The primary goal is a first attempt to develop an intuitive Web interface for the integrated hydrologic simulation platform ParFlow (Jones and Woodward, 2001; Kollet and Maxwell, 2006) in HPC environments that allows the user to access scientific software and hardware resources on-demand from anywhere in world via a terminal in connection with a Web browser, i.e. SaaS according to the NIST definition. In the following, the development strategy and implementation of various capabilities are discussed. So far applications at the beta level deal with simple simulation and optimization problems including post- and pre-processing (visualization) to demonstrate the usefulness of the system.

2. Development strategy, implementation and description of system's components

In the development of the Parflow Web service and interface, the following specifications were defined as guidelines to arrive at a functional beta version for demonstration and application purposes:

- no involvement of the end user with regard to software, hardware configurations and management of HPC resources
- full functionality of ParFlow including parallel simulations
- I/O handling with data up/downloading
- basic optimization capabilities and visualization of output

Fig. 1 shows a schematic diagram of the system components and their functional relationships.

2.1. System's setup

The system consists of a Linux cluster where ParFlow has been installed and tested following standard procedures. A server,

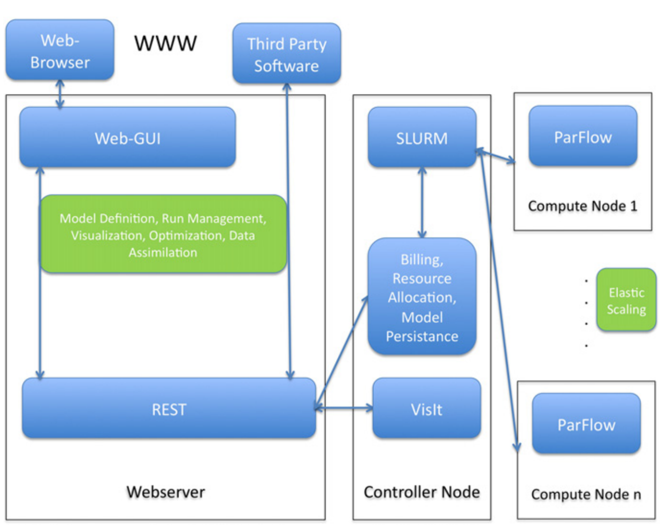

Fig. 1. Set-up of the ParFlow system: relations between the world-wide-web, hardware components (black boxes) and software components (blue boxes) are depicted by arrows and functional descriptions are given inside the green boxes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

hosting the ParFlow Web service and interface, is connected to the cluster and the Internet constituting the gateway for registered users of the system. The Web service communicates with the cluster via a collection of scripts that submit and control simulation jobs using SLURM (https://computing.llnl.gov/linux/slurm), an open-source derivative of the Parallel Batch System, PBS, for cluster resources management. The simulation jobs are defined and submitted by the user via the Web interface which is based on the Model–Viewer–Controller (MVC) architectural pattern (explained in more detail below). At this point the cluster and the upstream server provide all the requirements for setting up hydrologic models, performing simulations, storing output and also visualization. With current hardware resources only a small user base can be handled, however, scalable HPC utilities such as SLURM ensure system adaptation without major changes in the communication infrastructure of the Web service.

2.2. The graphical user interface

The MVC architectural pattern was implemented using Ruby on Rails as the Web framework. Additionally, JavaScript was utilized to enhance the user interface. Communication between Web server and browser was realized with a REST-style architecture (Fielding, 2000). The latter also ensures that all cluster functionality is exposed as an application programming interface (API) to any third party software. This API can be accessed from any device connected to the internet since it is based on HTTP (Hypertext Transfer Protocol). The webserver to cluster communication was handled with ssh system calls where SLURM commands are executed in order to manage numerical simulations on the cluster. Status information is provided in near realtime to the user. The view component of the MVC architectural pattern was implemented as a single web page application where all communication between webserver and web-browser is handled via AJAX (asynchronous JavaScript and XML; http://ajaxpatterns.org) calls.

The webclient was programmed using the Javascript framework Jquery (http://jquery.com/) and the backbone.js library (http://documentcloud.github.com/backbone/). All server objects are imaged onto the client and locally represented as objects or collection of objects. Synchronization is done when new user input is provided and saved. Here, the objects maintain a list of observers that act on any changes of the status of the model or collection of models. Save/update and delete calls are received via the REST interface by the servers and executed. Views are required to show a list of the model objects and to modify individual objects. Modifications are done with HTML templates which are generated or transferred by the server using AJAX calls.

2.3. Model generation, simulation and post-processing

Upon registration and login at www.hipos-model.com, three model examples are provided to the user via the "Models" tab. These models can be used as a starting point for customization or the user can create a new model from scratch by clicking the "New model" button. Upon model activation a short description is shown and the input database (model geometry, input parameters, timing information, pumping well information and solver variables) is made available.

The database can be edited via the "Edit Model" tab where the aforementioned model attributes can be modified. The "Processes" tab allows the specification of the number or processors used in the HPC environment. The tabs follow ParFlow naming convention i.e. each tab name can be found in the ParFlow manual so that user guidance is ensured. The completed model is submitted to the cluster via the "Run" tab, which also includes

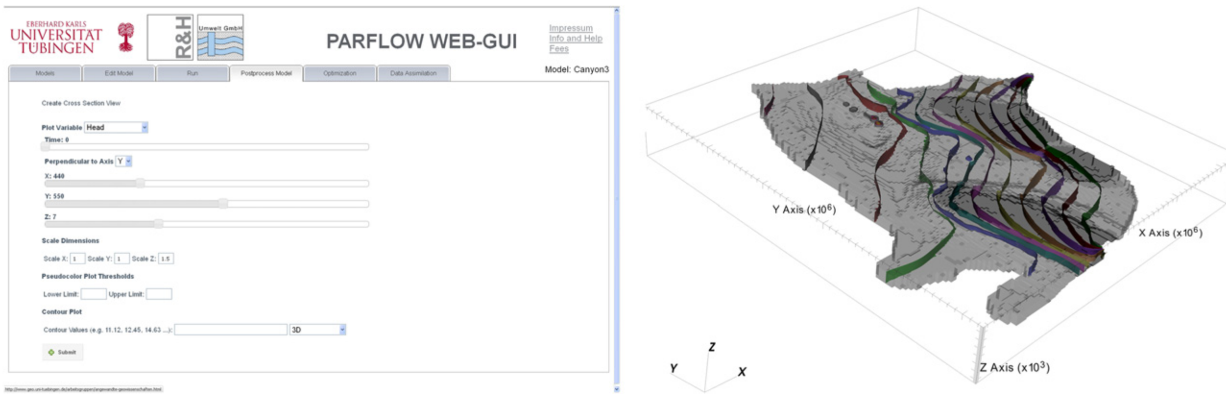

Fig. 2. The "Postprocess Model" tab showing the basic options for visualization (left panel). Visualization example of a contour plot of hydraulic heads resulting from pumping in an unconfined aquifer with zonal heterogeneity (right panel).

near real-time logging of job status variables. Additional options accessible through "Optimization" and "Data Assimilation" tabs are optimization capabilities via PEST (Doherty, 2004) and a basic data assimilation interface for ensemble Kalman filtering adopting the Evenson (2003) algorithm. Complete simulation runs can be downloaded or visualized using basic post-processing tools.

Visualization was implemented using the parallel rendering, open source software VisIt (https://wci.llnl.gov/codes/visit/) developed at the Lawrence Livermore National Laboratory for efficient visualization of very large data sets. VisIt has been installed on the cluster and communicates with the server via password-free ssh.

Under the "Postprocess Model" tab different variables and hydraulic parameters can be defined for plotting. Basic plot types currently available are 3D pseudo color volume and contour plots including cross sections with adjustable thresholds and individual time slice selection. Fig. 2 shows a screen shot of the "Postprocess Model" tab together with a simple contour plot of hydraulic head of an ambient flow field influenced by mild aquifer heterogeneity.

3. Summary and conclusions

The project demonstrates that the framework of cloud computing is useful in providing scientific software in connection with high-performance computing resources to a broad user base. The integrated hydrologic simulation platform ParFlow has been implemented in this framework including an intuitive graphical user interface, and optimization and basic visualization utilities. Registered users can access the system via a terminal and Web browser independent of their geographic location. The interface advances the concept of cloud computing by providing a comprehensive user interface, not only for the application of ParFlow, but also for its use in supercomputer environments without the direct involvement of the end-user. This also opens new possibilities for true grid computing, i.e. the simultaneous utilization of heterogeneous systems located at different geographic locations. The current version of the interface provides full functionality of ParFlow including the use of cluster resources. In the future, additional tools will be made available to simplify the generation of complex model geometries and to perform advanced scientific visualization.

References

CUAHSI. 〈http://www.cuahsi.org/〉.

Doherty, J., 2004. PEST: Model-Independent Parameter Estimation, Users Manual, Version 5. Watermark Numerical Computing, Brisbane, Australia.

Evangelinos, C., Hill, C.N., 2008. Cloud computing for parallel scientific HPC applications: feasibility of running coupled atmosphere–ocean climate models on Amazon's EC2. In: Proceedings of the CCA-08 Conference: Cloud Computing and its Applications, October 22–23, 2008, Chicago.

Evenson, G., 2003. The ensemble Kalman Filter: theoretical formulation and practical implementation. Ocean Dynamics 53, 343–367.

Fielding, R.T., 2000. Architectural Styles and the Design of Network-based Software Architectures. Doctoral Dissertation. University of California, Irvine.

Fienen, N.F., Kunicki, T.C., Kester, D.E., 2011. cloudPEST—A Python Module for Cloud-Computing Deployment of PEST, A Program for Parameter Estimation. Open-File Report 2011-1062. U.S. Geological Survey, Reston, Virginia, March 2011.

Hunt, R.J., Luchette, J., Schreuder, W.A., Rumbaugh, J.O., Doherty, J., Tonkin, M.J., Douglas, Rumbaugh, D.B., 2010. Using a cloud to replenish parched groundwater modeling efforts. Ground Water 48 (3), 360–365.

Jones, J.E., Woodward, C.S., 2001. Newton–Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems. Advances in Water Resources. 24, 763–774.

Kanwar, R., Narayan, U., Lakshmi, V., 2010. Web service based hydrologic data distribution system. Computers & Geosciences 36 (7), 819–826. doi:10.1016/j.cageo.2007.10.017.

Kollet, S., Maxwell, R., 2006. Integrated surface-groundwater flow modeling: a free-surface overland flow boundary condition in a parallel groundwater flow model. Advances in Water Resources 29 (7), 945–998.

Kruger, A., Khandelwal, S.G., Bradley, A., 2007. AHPSVER: a web-based system for hydrologic forecast verification. Computers & Geosciences 33 (6), 739–748.

Mell, P., Grance, T., 2011. The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology. NIST Special Publication 800-145. Computer Security Division Information Technology, Laboratory National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, September 2011.

Sterling, T., Stark, D., 2009. A High-Performance Computing Forecast: Partly Cloudy. Computing in Science & Engineering 11 (4), 42–49.