# Cloud Computing for Geosciences

Geoffrey C. Fox and Marlon E. Pierce

Pervasive Technology Institute, Indiana University

Cyberinfrastructure has closely tracked commercial best practices for over a decade, but we believe there is still much to learn about correct strategies for building distributed systems for collaborating scientists and related communities. In this position paper, we discuss the opportunities to the geo-sciences if Cloud Computing strategies currently used by commercial data centers are adopted.

We base our conclusions on several geospatially themed projects that we have participated in. These include the NASA-funded QuakeSim project (1), the USGS-funded FloodGrid project, and the NSF-funded PolarGrid project (www.polargrid.org). Our lab has developed cyberinfrastructure software to support these distributed spatial applications, and we have also investigated Cyberinfrastructure architecture generally (2). Our applications include Geospatial Information System (GIS) Grid services based on Open Geospatial Consortium standards and real-time streaming Global Positioning System processing infrastructure (3). As can be seen from this list of applications, we take a very broad view of the problems that Cyberinfrastructure (CI) must support. Computing and data storage are just two aspects; we also need to manage real-time data streams, integrate third party capabilities (such as map and data providers), and build interactive user interfaces that act as Science Gateways (4). We take here a heterogeneous view of Cyberinfrastructure: it could include GIS services provided by state and local governments as well as Globus services on the TeraGrid.

The current flagship deployments of cyberinfrastructure, such as the NSF TeraGrid, do not take a comprehensive approach towards cyberinfrastructure and are dominated by the requirements of traditional high performance computing users. Arguably the NSF DataNet program will address the data-centric needs of cyberinfrastructure, such as long-term storage and preservation of observational and experimental data, but this program is in its infancy. We advocate generally for the adoption of Cloud Computing approaches to CI, which we believe will offer a broader approach to infrastructure. Cloud Computing-like infrastructure is of particular interest to Spatial CI applications, which provides important use cases that help clarify the capabilities an end-to-end CI deployment should provide. The recently funded Future Grid project (futuregrid.org) can serve as a testbed for evaluating these Cloud approaches to geospatial CI.

**Cyberinfrastructure and Cloud Computing:** Cloud Computing is a marketing term that is usually left poorly defined. However, because of its potential value to academic computing, academic surveys and initial investigations exist (see for example 5, 6). We will focus on two specific aspects: Cloud Computing to provide infrastructure and Cloud Computing to provide runtime management.

*Infrastructure as a Service:* Clouds may be defined as Web services that control the life cycles of virtual machines and virtual storage. The very well known Amazon and Microsoft Azure cloud systems fall in this category. Xen (www.xen.org) is a popular technology for virtualizing server farms and data centers based on Linux; Microsoft similarly has Hyper-V for Windows Server 2008-based farms. Through Web services and virtualization, users can create and control their own resources. The virtual machines can come preconfigured with useful software. For example, one may imagine checking out a virtual machine or cluster that comes pre-configured with geospatial software. Less well known than the virtual machine but at least as important is the virtual block storage device. The best example of this is the Amazon Elastic Block Store, which can be attached to a virtual machine to provide additional file space. These attached file systems don’t need to be empty. As Amazon’s public data sets illustrate (aws.amazon.com/publicdatasets/), we can create libraries of public and community data sets (files or databases) that can be checked out by individual users. Finally, we note the major Cloud vendors all have very scalable but flat database technologies in their infrastructure. These include Google’s BigTable, Microsoft Azure’s Table Service, and Amazon’s SimpleDB. These lack the full functionality of relational databases but work very well as Cloud spreadsheets.

Although we have focused on commercial cloud infrastructure, it is possible to set up a cloud using Open Source software on existing server farms and clusters. Example software includes Eucalyptus (9), Nimbus (7), and OpenNebula (www.opennebula.org). Production academic cloud installations based on these and related technologies are becoming available. The NanoHUB project at Purdue University is one of the most prominent (8).

*Runtime management as a Service:* Although one may want to use a Cloud to outsource infrastructure, it is also desirable to have higher-level tools that can harness the elastic computing power of Cloud installations. The idea is that the Cloud provides some specific suite of capabilities on top of its infrastructure, and the Cloud user does not drill down to the underlying operating system. Apache Hadoop is a relevant example of this. Hadoop is an implementation of two ideas promulgated by Google: the Google File System and Map-Reduce (11). Strictly speaking, Hadoop and its competitors don’t need to run on Cloud infrastructure, but the two are a good match (see for example Amazon's Elastic Map Reduce, aws.amazon.com/elasticmapreduce/). Map-reduce is essentially a highly parallelizable approach for managing multiple serial computing tasks in distributed environments. Although it can be applied to a wide range of problems (10), it generally is designed to support problems with data-file parallelism; that is, we need to apply an operation or a sequence of operations to huge input files or data sets that can be split into smaller fragments. Map-Reduce and its competitors (prominently, Microsoft’s Dryad (12)) are designed to solve the world’s largest data-file parallel problem, search. However, they are also well adapted to geospatial problems such as image processing. The notion of file parallelism can also be extended to network streams and other standard input/output mechanisms. Processing and mining sensor streams in a large sensor Web is an obvious application for stream data parallelism in Spatial CI. Although not supported by Hadoop, this is an intended feature of Dryad and has been explored by research groups (13, 14).

**References**

1. Donnellan, A., et al (2006) QuakeSim and the Solid Earth Research Virtual Observatory, *Pure and Applied Geophysics,* Volume 163, Numbers 11-12, 2263-2279.
2. Fox, G., Lim, S., Pallickara, S., Pierce, M. (2005) Message-based Cellular Peer-to-Peer Grids: Foundations for Secure Federation and Autonomic Services, *Journal of Future Generation Computer Systems*, 21(3), 401–415. (2005).
3. Granat, R., Aydin, A., Pierce, M.E., Qi, Z., and Bock, Y. (2007) Analysis of streaming GPS measurements of surface displacement through a web services environment, *CIDM:* 750-757 (2007).
4. Wilkins-Diehr, N., Gannon, D., Klimeck, G., Oster, S., Pamidighantam, S. (2008): TeraGrid Science Gateways and Their Impact on Science. *IEEE Computer* 41(11): 32-41.
5. Youseff, L.; Butrico, M.; Da Silva, D (2008) Toward a Unified Ontology of Cloud Computing. Page(s): 1-10 Digital Object Identifier 10.1109/GCE.2008.4738443.
6. Jha, S., Merzky, A., Fox, G (2009) Using clouds to provide grids with higher levels of abstraction and explicit support for usage modes. *Concurrency and Computation: Practice and Experience* 21(8): 1087-1108.
7. Foster, I., et al. (2006) Virtual Clusters for Grid Communities, *CCGRID*: 513-520.
8. Klimeck, G., et al (2008), nanoHUB.org: Advancing Education and Research in Nanotechnology, *IEEE Computers in Engineering and Science (CISE),* Vol. 10, 17-23 (2008).
9. Nurmi, D., et al (2008) The Eucalyptus Open-source Cloud-computing System, in *Proceedings of Cloud Computing and Its Applications*, Chicago, IL (October 2008).
10. Chu C-T, et al (2006). Olukotun, Map-Reduce for Machine Learning on Multicore, *NIPS*: 281-288.
11. Dean, J., Ghemawat, S. (2008) MapReduce, Simplified Data Processing on Large Clusters. Commun, *ACM* 51(1): 107-113.
12. Isard, M., Budiu, M., Yu Y., Birrell, A., Fetterly, D. (2007) Dryad, Distributed Data-Parallel Programs from Sequential Building Blocks, *EuroSys*: 59-72.
13. Ekanayake, J.; Pallickara, S.; Fox, G. (2008) MapReduce for Data Intensive Scientific Analyses*. IEEE Fourth International Conference on eScience '08*  
    7-12 Dec. 2008 Page(s):277 - 284 Digital Object Identifier 10.1109/eScience.2008.59
14. Pallickara, S.; Ekanayake, J.; Fox, G. (2008) An Overview of the Granules Runtime for Cloud Computing. *IEEE Fourth International Conference on eScience '08*, 7-12 Dec. 2008 Page(s): 412 - 413 Digital Object Identifier 10.1109/eScience.2008.101.