# SciCloud: Scientific Computing on the Cloud

Satish Srirama, Oleg Batrashev and Eero Vainikko
Distributed Systems Group
Institute of Computer Science, University of Tartu
J. Liivi 2, Tartu, Estonia
{srirama, olegus, eero}@ut.ee

***Abstract*— SciCloud is a project studying the scope of establishing private clouds at universities. With these clouds, researchers can efficiently use the already existing resources in solving computationally intensive scientific, mathematical, and academic problems. The project established a Eucalyptus based private cloud and developed several customized images that can be used in solving problems from mobile web services, distributed computing and bio-informatics domains. The poster demonstrates the SciCloud and reveals two applications that are benefiting from the setup along with our research scope and results in scientific computing.**

***Keywords: Cloud computing, scientific computing, GRID, mobile web services, Mobile Host, Eucalyptus***

## I. INTRODUCTION & PROJECT SCOPE

The main goal of the scientific computing cloud (SciCloud) project [8] is to study the scope of establishing private clouds at universities. With these clouds, students and researchers can efficiently use the already existing resources of university computer networks, in solving computationally intensive scientific, mathematical, and academic problems. Traditionally such computationally intensive problems were targeted by batch-oriented models of the GRID computing domain. The current project tries to achieve this with more interactive and service oriented models of cloud computing that fits a larger class of applications. The established interoperable private clouds also provide better platforms for collaboration among interested groups of universities and in testing internal pilots, innovations and social networks.

The project mainly targets the development of a framework, including models and methods for establishment, proper selection, state management (managing running state and data), auto scaling and interoperability of the private clouds. Once such clouds are feasible, the networks can also be leased to governmental institutions for such diverse applications as drug discovery, seismic analysis, and back-office data processing in support of e-commerce and web services. SciCloud also focuses at finding new distributed computing algorithms and tries to reduce some of the scientific computing problems to MapReduce algorithm [3]. The project thus shall ultimately benefit the cluster, cloud and grid community.

In achieving these goals, we tried to establish a Eucalyptus [5] based private cloud on our already existing cluster resources. We have developed several customized machine images that can be used in solving problems from mobile web services, distributed computing and bio-informatics domains. This poster addresses the deployment details of SciCloud and reveals some of the domain specific applications that are using the private cloud.

## II. SCICLOUD ARCHITECTURE

Cloud computing [2] is a style of computing in which, typically, real-time scalable resources are provided “as a service (aaS)” over the Internet to users who need not have knowledge of, expertise in, or control over the cloud infrastructure that supports them. The provisioning of cloud services can be at the Infrastructural level (IaaS) or Platform level (PaaS) or at the Software level (SaaS). A cloud computing platform dynamically provisions, configures, reconfigures, and de-provisions servers as requested. This ensures elasticity of the systems deployed in the cloud. Cloud computing mainly forwards utility computing model, where consumers pay based on their usage. Servers in the cloud can be physical machines or virtual machines.

While there are several public clouds on the market, Google Apps (example of SaaS, includes Google Mail, Docs, Sites, Calendar, etc), Google App Engine [6] (example of PaaS, provides elastic platform for Java and Python applications) and Amazon EC2 (example of IaaS) are probably most known and widely used. Amazon EC2 [1] provides an instance of a virtual machine image that allows full control over the operating system. It is possible to select a suitable operating system, and platform (32 and 64 bit) from many available Amazon Machine Images (AMI) and several possible virtual machines, which differ in CPU power, memory and disk space. This functionality allows freely select suitable technologies for any particular task. In case of Amazon EC2 price for the service depends on machine size, its uptime, and used bandwidth in and out of the cloud.

There are also free implementations of cloud infrastructure e.g. Eucalyptus [5]. Eucalyptus allows creating private clouds compatible with Amazon EC2. Thus the cloud computing applications can initially be developed at the private clouds and can later be scaled to the public clouds. The setup is of great help for the research and academic communities, as the initial expenses of experiments can be reduced by great extent. With this primary goal we have set up a SciCloud on a cluster consisting of 8 nodes of SUN FireServer Blade system with 2-core AMD Opteron Processors, using Eucalyptus technology. With the SciCloud, students and researchers can efficiently test their applications, in solving computationally intensive scientific, mathematical, and enterprise problems.

## III. SciCloud Application frameworks

While several applications are obvious from such a private cloud setup, we have used it in solving some of our research problems. Here we address two such problem domains, for which we have developed customized images. We also have images supporting scientific computing and bio-informatics.

### *A. DOUG*

DOUG (Domain decomposition On Unstructured Grids) [4] is an open source software package for parallel iterative solution of very large sparse systems of linear equations with up to several millions of unknowns. It handles 2D and 3D problems that arise typically with finite element discretization of elliptic partial differential equations. DOUG has been developed at the University of Tartu (Estonia) and the University of Bath (UK) since 1995. To achieve fast processing times, DOUG uses graph partitioning techniques to achieve good load balancing. DOUG has been experimentally shown to be of superior speed to comparable methods, especially with its aggregation based domain decomposition methods. Parallelization is implemented through MPI (Message Passing Interface) trying to overlap computations and communication where at all possible.

Our experience with GRID and ordinary private clusters revealed that installing DOUG is not a trivial task, as it needs a number of specific libraries and proper configuration of MPI development environment. The administrator of a cluster has usually no knowledge about these details, while it is straightforward for the application developers and the advanced users of the application. The software configuration and administrative tasks have been one of the major problems with GRID establishment, which we observed to lessen by using customized images and flexible models of the cloud computing. Our solution is to let administrator prepare general cluster image with compilers, MPI environment and common scientific libraries installed and allow others to extend this image to their needs. The resulting image can again be used by everyone.

DOUG also needs high speed connections between the nodes. Since the SciCloud instances are running on a single cluster, network latency is acceptable. To prove this, we experimentally measured performance latencies for several scientific computing tasks run on clusters and cloud nodes. We considered the performance of processors using BLAS (Basic Linear Algebra Subprograms) libraries, sparse matrix vector multiplication in CG (Conjugate Gradient) method, and network latencies using MPI routines and NAS parallel benchmarks. These experiments further supported our point of having programmer administrable nodes and thus justified the requirement for private clouds in some domains.

### *B. Mobile Web Service Provisioning*

SciCloud is also relevant and used in solving other enterprise problems; Web services [7] are going mobile. A Mobile Enterprise can be established in a cellular network by participating Mobile Hosts [9], which act as web service providers, and their clients. Mobile Hosts enable seamless integration of user-specific services to the enterprise, by following web service standards, also on the radio link and via resource constrained smart phones. Several applications were developed and demonstrated with the Mobile Host in healthcare systems, collaborative m-learning, social networks and multimedia services domains. Challenges associated with establishing Mobile Enterprise were studied extensively and the research realized a mobile web service mediation framework (MWSMF) [9] that helps in offering proper quality of service (QoS) and discovery mechanisms for the provided services.

While the MWSMF was successful in achieving the integrational requirements of the Mobile Host and the Mobile Enterprise, cellular networks are very prone to sudden surges. For example, number of Mobile Hosts providing the services and the number of services provided by the Mobile Hosts can explode while some events are underway; like Olympics or national elections etc. Thus to scale Mobile Enterprise to the loads possible in cellular networks, it was necessary to shift some of its components to the cloud computing. So we shifted some of the components and load balancers at different levels to the SciCloud. From this analysis, we observed that a single MWSMF node can handle around 100-130 concurrent mobile web service requests with 100% success rate and adding an additional node adds around 100 new concurrent requests to the total capability. Consequently we showed that MWSMF and its components are horizontally scalable, thus allowing to utilize elasticity of cloud platform to meet load requirements of Mobile Enterprise in an easy and quick manner.


## Acknowledgment

The research is supported by the Estonian Science Foundation under Mobilitas program and the European Regional Development Fund through the Estonian Centre of Excellence in Computer Science.


## References


[1] Amazon Inc, "Amazon Elastic Compute Cloud (Amazon EC2)", http://aws.amazon.com/ec2/ [URLs last visited on 12th Jan 2010]

[2] M. Armbrust et al., "Above the Clouds, A Berkeley View of Cloud Computing", Technical Report, University of California, Feb, 2009.

[3] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", OSDI'04: Sixth Symposium on Operating System Design and Implementation, Dec, 2004.

[4] DOUG, "DOUG development site", http://www.dougdevel.org/

[5] Eucalyptus Systems Inc, "Eucalyptus", http://www.eucalyptus.com/

[6] Google Inc, "App Engine Java Overview", http://code.google.com/appengine/docs/java/overview.html

[7] K. Gottschalk, S. Graham, H. Kreger, and J. Snell, "Introduction to web services architecture", IBM Systems Journal: New Developments in Web Services and E-commerce, 41(2):178–198, 2002.

[8] S. N. Srirama, "Scientific Computing on the Cloud (SciCloud)", http://ds.cs.ut.ee/research/scicloud

[9] S. N. Srirama, M. Jarke, "Mobile Hosts in enterprise service integration", International Journal of Web Engineering and Technology (IJWET), 5(2):187-213, 2009. Inderscience Publishers.