

An open source, cloud independent, Java API for high performance cloud computing application design, development, simulation and evaluation

R. Montella, F. Lucarelli, P. Esposito
Department of Applied Science – University of Napoli Parthenope

Abstract. Our emerging research project integrates an IaaS cloud computing technology aware framework for high level application development (JaClouX, JCX) with a cloud simulator (dryRain, dR). With our tools developers can implement, simulate and evaluate their cloud application and then use them in any real cloud environment without any code change. JCX offers a common interface to public and private cloud providers; implements unsupported features in order to make the application provider-aware; makes available High Performance Cloud Computing specific services. dR is integrated in JCX. Developers can perform application dry runs estimating performances and costs. dR is even a standard alone application implementing a cloud simulator based on events.

1. Introduction

The cloud computing technology is changing the approaching way to the need for computing power even in a context of in-house resource management. The High Performance Cloud Computing (HPCC) is a promising application field especially about the possibility of mixing private and public on-demand resources heading to technological, environmental and economic benefits. Traditional E-Science could be straightforwardly turned to Elastic-Science: the future of scientific applications designed for cloud computing deployment is strictly related to the availability of *ad-hoc* middleware as it happened previously with HPC and grid computing.

The rest of the work is so organized: the section 2 is about JaClouX design and implementation; in the section 3 is described dryRain; Related works are discussed in the section 4 and, finally, the section 5 is about the conclusions and the highlights on future developments.

2. JaClouX

Different cloud infrastructure systems, such as Amazon AWS, Open Nebula, Open Stack, Eucalyptus, etc, offer similar services through different web service interfaces, so a cloud-aware component is needed in order to manage virtual machine instances on different clouds. JaClouX (Java bindings for Cloud eXtensions) [4] is a unified Java API and represents a fundamental change in the way clouds are managed, breaking the barriers among proprietary, closed clouds and open scientific clouds. JaClouX is our cloud management component; it is independent of the underlying cloud software and interacts with other components as Images, Instances, Locations and Sizes. Moreover, JaClouX is cloud-aware thanks to a plug in architecture. The support to any cloud software can be added by just

implementing a new Driver component. Cloud software manages virtual machine images and instances in a quite similar way, but the storage and other high level services turn out to be different among the various cloud solutions. JaClouX's driver component provides an abstraction of both Amazon-like S3 and EBS services using the Simple Storage Component (SSC) and the Elastic Block Component (EBC). It provides the missing cloud features by software emulation. The Virtual Machine Listener (VML) implements a complex, but easy to use, monitor and trigger engine. A trigger is defined using a script (anything can programmatically change the trigger status on a virtual machine instance). Listeners register to the trigger and then woke up when the status changes. In a cloud application context, VM instancing is comparable to a job submission in a classic HPC environment. JaClouX provides a component framework in order to manage programmatically the schedule. The VMS (Virtual Machine Scheduler) provides an advanced support of VM instances based scientific workflows. The Virtual Computing Cluster (VCC) permits the dynamical creation of high performance computing cluster on-demand. A VCC is defined specifying the virtual machine image for the head and the working nodes and the number of working nodes instances. VCC manages the user authentication and the execution authorization in a coherent fashion deploying automatically, if needed, the security components. The virtual network file system permits to the head node user to execute on working nodes using secure shell and perform some other operation in the distributed memory parallel programming ecosystem. VCC takes in charge file system related issues abstracting the low level implementation especially regarding the distributed parallel file system. VCC interacts with the cloud scheduler, where possible, in order to locate the virtual computing nodes in the most effective and high performance way: i.e. groups of computing nodes belonging to the same virtual cluster can be allocated on the same real machine, so MPI optimization for virtual networks can be used in the best way; in a similar way, virtual working nodes needing for GPGPU acceleration support are spawn on the real hardware achieving the best possible computing effort (Figure 1).

3. DryRain

The simulator component fills the gap between the development and the deployment cloud tools. Elastic sciences leverage on cloud scalability and on infinite

