FoSII - Foundations of Self-Governing ICT Infrastructures

by Vincent C. Emeakaroha, Michael Maurer, Ivona Brandic and Schahram Dustdar

The DSG Group at Vienna University of Technology is investigating self-governing Cloud Computing infrastructures necessary for the attainment of established Service Level Agreements (SLAs). Timely prevention of SLA violations requires advanced resource monitoring and knowledge management. In particular, we develop novel techniques for mapping low-level resource metrics to high-level SLAs, monitoring resources at execution time, and applying Case Based Reasoning for the prevention of SLA violations before they occur while reducing energy consumption, ie, increasing energy efficiency.

Cloud computing is a promising technology for the realization of large, scalable on-demand computing infrastructures. Currently, many enterprises are adopting this technology to achieve high performance and scalability for their applications while maintaining low cost. Service provisioning in the Cloud is based on a set of predefined non-functional properties specified and negotiated by means of Service Level Agreements (SLAs). Cloud workloads are dynamic and change constantly. Thus, in order to reduce steady human interactions, self-manageable Cloud techniques are required to comply with the agreed customers' SLAs.

Flexible and reliable management of SLAs is of paramount importance for both Cloud providers and consumers. On the one hand, the prevention of SLA violations avoids penalties that are costly to providers . On the other hand, based on flexible and timely reactions to possible SLA violation threats, user interaction with the system can be minimized enabling Cloud computing to take roots as a flexible and reliable form of on-demand computing. Furthermore, a trade-off has to be found between proactive actions that prevent SLA violations and those that reduce energy consumption, ie, increase energy efficiency.

The Foundation of Self-governing ICT Infrastructures (FoSII) research project is proposing solutions for autonomic management of SLAs in the Cloud. The project started in April 2009 and is funded by the Vienna Science and Technology Fund (WWTF). In this project, we are developing models and concepts for achieving adaptive service provisioning and SLA management via resource monitoring and knowledge management techniques.

Figure 1 depicts the components of the FoSII infrastructure. Each FoSII service implements three interfaces: (i) negotiation interface necessary for the establishment of SLA agreements, (ii) service management interface necessary for starting service, uploading data, and similar management actions, and (iii) self-management interface necessary to devise actions in order to prevent SLA violations.

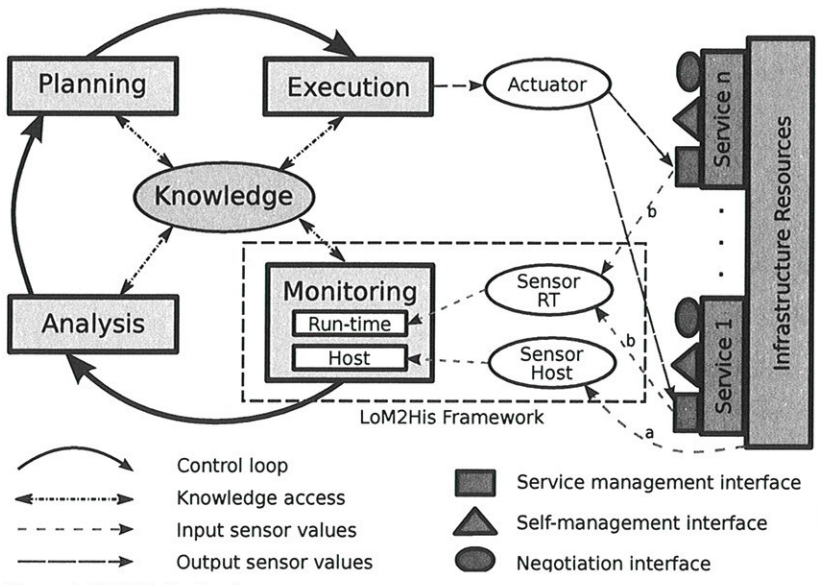

Figure 1: FoSII Infrastructure.

The self-management interface as shown in Figure 1 specifies operations for sensing changes of the desired state and for reacting to those changes. The host monitor sensors continuously monitor the infrastructure resource metrics (input sensor values arrow a in Figure 1) and provide the knowledge component with the current resource status. The run-time monitor sensors sense future SLA violation threats (input sensor values arrow b in Figure 1) based on resource usage experiences and predefined thresholds.

As shown in Figure 1, the Low-level Metric to High-level SLA (LoM2HiS) framework is responsible for monitoring and sensing future SLA violation threats. It comprises the host monitor and the run-time monitor. The host monitor monitors low-level resource metrics such as CPU, memory, disk space, incoming bytes, etc using monitoring agents like Gmond from Ganglia project embedded in each Cloud resource. It extracts the monitored output from the agents, processes them and sends the metric-value pairs through our implemented communication model to the run-time component.

The run-time component receives the metric-value pairs and, based on predefined mapping rules, maps them into equivalent high-level SLA parameters. An example of an SLA parameter is service availability Av, which is calculated using the resource metrics downtime and uptime as follows:
$Av = (1 - downtime/uptime) \times 100$.
The provider defines the mapping rules using appropriate Domain Specific

Languages (DSLs). The concept of detecting future SLA violation threats is designed by defining a more restrictive threshold than the SLA violation thresholds known as threat threshold. Thus, calculated SLA values are compared with the predefined threat threshold in order to react before an SLA violation occurs. In case SLA violation threats are detected, the run-time monitor sends notification messages to the knowledge component for preventive actions.

During the analysis and planning phases the knowledge component then suggests appropriate actions to solve SLA violation threats. As a conflicting goal, it also tries to reduce energy consumption by removing resources from over-provisioned services. Reactive actions thus include increasing or decreasing memory, storage or CPU usage for each service. After the action has been executed the knowledge component learns the utility of the action in this specific situation via Case Based Reasoning (CBR). CBR contains previously solved cases together with their actions and utilities, and tries to find the most similar case with the highest utility for each new case. Furthermore, it examines the timing and the effectiveness of an action, ie, whether the action would have helped but was triggered too late, or was unnecessarily triggered too early, and consequently, it updates the threat thresholds from the monitoring component. In the future, the knowledge component will offer different energy efficiency classes that will reflect the trade-off between preventing violations and saving energy, and it will integrate knowledge about penalties and client's status for prioritizing resource demand requests when resources are scarce.

We have successfully implemented the first versions of the LoM2HiS framework and the knowledge component. First evaluation results of the components have been published in top-ranked international conferences: HPCS 2010, COMPSAC 2010, SERVICES 2010, and CloudComp 2010.

Links:
http://www.infosys.tuwien.ac.at/linksites/FOSII/index.html
http://www.infosys.tuwien.ac.at/
http://www.infosys.tuwien.ac.at/staff/vincent/

Please contact:
Vincent Chimaobi Emeakaroha
Vienna University of Technology / AARIT, Austria
Tel +43 1 58801 18457
E-mail: vincent@infosys.tuwien.ac.at

Large-Scale Cloud Computing Research: Sky Computing on FutureGrid and Grid'5000

by Pierre Riteau, Maurício Tsugawa, Andréa Matsunaga, José Fortes and Kate Keahey

How can researchers study large-scale cloud platforms and develop innovative software that takes advantage of these infrastructures? Using two experimental testbeds, FutureGrid in the United States and Grid'5000 in France, we study Sky Computing, or the federation of multiple clouds.

The remarkable development of cloud computing in the past few years, and its proven ability to handle web hosting workloads, is prompting researchers to investigate whether clouds are suitable to run large-scale scientific computations. However, performing these studies using available clouds poses significant problems. First, the physical resources are shared with other users, which can interfere with performance evaluations and render experiment repeatability difficult. Second, any research involving modification of the virtualization infrastructure (eg, hypervisor, host operating system, or virtual image repository) is impossible. Finally, conducting experiments with a large number of resources provisioned from a commercial cloud provider incurs high financial cost, and is not always possible due to limits to the maximum number of resources one can use. These problems, which would have been limitations for our sky computing experiments, were avoided by our use of experimental testbeds.

We study sky computing, an emerging computing model where resources from multiple cloud providers are leveraged to create large-scale distributed virtual clusters. These clusters provide resources to execute scientific computations requiring large computational power. Establishing a sky computing system is challenging due to differences among providers in terms of hardware, resource management, and connectivity. Furthermore, scalability, balanced distribution of computation, and measures to recover from faults are essential for applications to achieve good performance.

Experimental distributed testbeds offer an excellent infrastructure to carry out our sky computing experiments. We make use of the following two platforms: FutureGrid, a new experimental grid testbed distributed over six sites in the United States, and Grid'5000, an infrastructure for large-scale parallel and distributed computing research composed of nine sites in France. Using the reconfiguration mechanisms provided by these testbeds, we are able to deploy the Nimbus open source cloud toolkit on hundreds of nodes in a matter of minutes. This gives us exclusive access to cloud platforms similar to real-world infrastructures, such as Amazon EC2. Full control of the physical resources and of their software stack guarantees experiment repeatability. Combining two testbeds gives us access to more resources and, more importantly, offers a larger geographical distribution, with network latencies and bandwidth on a par with those found on the Internet. Our project is the first combining these two testbeds, paving the way for further collaboration.

Several open source technologies are integrated to create our sky computing infrastructures. Xen (an open source platform for virtualization) machine virtualization is used to minimize platform (hardware and operating system