Experiments for IaaS, PaaS, and SaaS on Heterogeneous Clouds

Stephen P. Crago, University of Southern California / Information Sciences Institute
and Ming Hsieh Department of Electrical Engineering, crago@isi.edu, 703-812-3729
John Paul Walters, USC/ISI, jwalters@isi.edu, 703-812-3706

An important characteristic of cloud computing is that it abstracts hardware architectures from end users and programmers. This abstraction allows the underlying infrastructure to be changed without forcing change upon applications. Maintaining this abstraction has been aided by the dominance of the x86 processor architecture and high-level, general-purpose portable languages. Meanwhile, the relentless drive of Moore's Law and improving microprocessor performance have improved the performance of most applications that execute on general-purpose processors. Cloud computing is predominantly built on these general-purpose processors and has been able to exploit these performance improvements. Unfortunately, this historic trend for homogeneous, general-purpose microprocessors is coming to an end because of limitations in power density and heat removal. One way to overcome these limitations, which the computing industry is already adopting, is by incorporating heterogeneity into processor architectures. The Chameleon and CloudLab projects have both recognized the importance of heterogeneity and have incorporated it into their hardware platforms.

In order to continue to obtain improvements in processing performance, developers of cloud computing technology will need to figure out the best ways to exploit heterogeneous processing elements , such as those shown in the architecture in Figure 1, without losing the advantages of the cloud abstraction. The means of achieving this goal are likely to be different for different forms of cloud computing, from Software-as-a-Service to Platform-as-a-Service to Infrastructure-as-a-Service. It is important that the NSFCloud architectures support experiments at each of these levels that allow researchers to explore ways to take advantage of heterogeneity at all levels of the software stack, from hypervisors to resource allocators and schedulers to application frameworks.

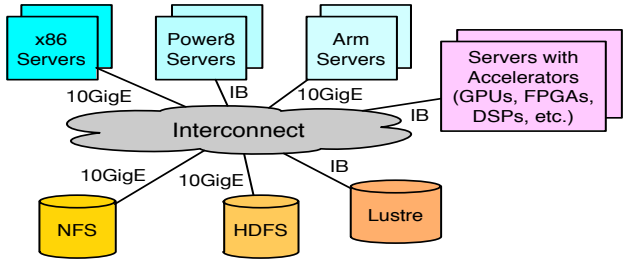

Figure 1. Example Heterogeneous Data Center Architecture

At the lowest level, Infrastructure-as-a-Service (IaaS) exposes physical and virtual resources to the end-user. Traditionally, virtualization has imposed a high overhead for performance-sensitive workloads, so it is important to support experimentation at the bare-metal level. We would like to run experiments to measure virtualization

overhead relative to bare metal for different processors. These experiments will include technology such as SR-IOV and PCI passthrough direct access to accelerators, networking and storage devices benchmarked across a variety of workloads and scales, starting with one node and moving up to hundreds or thousands of processors. It is important that the conductors of experiments can control the host operating systems and drivers since these can have significant affects on virtualization performance. Even lower level access, such as IPMI (Intelligent Platform Management Interface), may be useful. Cluster-level allocation of resources (including compute, network, and storage) to experiments is important to allow experimentation with IaaS stacks. Instrumentation to measure both performance and power in both the host and the virtual machines will also be required (performance more important than power since power can be estimated when performance is known). In order to support mapping of applications to heterogeneous hardware, it is also important to ensure that the appropriate hardware-specific run-time libraries are supported, and preferably development tools, although development tools could be run remotely at experimenter sites, as long as they are not invoked as part of a run-time system.

Experiments for Platform-as-a-Service (PaaS), including technologies such as Spark, Hadoop, and Hive, will allow heterogeneity to be exposed to the framework and possibly to the programmer We plan to conduct research to support heterogeneity-aware resource management and scheduling at the platform level for such frameworks. We plan to conduct experiments by developing such frameworks for heterogeneous cloud platforms. Since scalability is important for these frameworks, thousands of cores should be supported, with dedicated networks or subnets so that interconnect traffic effects can be controlled. The PaaS services may be built on virtual or bare-metal resources so both options should be supported, and scalability often is emphasized, so experiments should be able to support both small-scale early experiments and large experiments with thousands of nodes. Experimenters should be able to install cluster-level resource managers and schedulers. Software-as-a-Service experiments are not our focus, but we would expect to have similar requirements as PaaS.

Service Model	**Support for Heterogeneity**	**Experiment Characteristics**
IaaS	Must be exposed to users. Bare metal provisioning; virtualization of processors, accelerators, network, and storage	Support for bare-metal, different virtualization models, and different hardware combinations needed, medium scalability
PaaS and SaaS	Heterogeneity may be exposed to programmers. Framework manages and schedules heterogeneous resources.	Similar support for heterogeneity needed, dedicated network, medium to high scalability

Table 1**. Experimental Requirements for Cloud Service Models and Heterogeneity.**