EDITOR **SAN MURUGESAN**
BRITE Professional Services, Sydney, Australia;
san@computer.org

Heterogeneous Cloud Computing: The Way Forward

Stephen P. Crago and John Paul Walters, University of Southern California
Information Sciences Institute

Cloud computing developers face multiple challenges in adapting systems and applications for increasingly heterogeneous datacenter architectures.

A major appeal of cloud computing is that it abstracts hardware architecture from both end users and programmers. This abstraction allows underlying infrastructure to be scaled up or improved—for example, by adding datacenter servers or upgrading to newer hardware—without forcing changes in applications. The long-dominant x86 processor architecture, along with high-level, portable languages such as Java, PHP, Python, and SQL, has helped assure the continued viability of such abstraction. Meanwhile, exponential growth in microprocessor capability, mirroring Moore's law, has helped to improve performance for most applications that execute on general-purpose processors, including those deployed on clouds.

However, as transistors continue to shrink, concurrent limitations on power density and heat removal and the inability to scale down operating voltage any further mean that it's no longer possible to increase microprocessor performance by adding identical, general-purpose cores.[1]

HETEROGENEOUS DATACENTER ARCHITECTURES

These limitations, however, can be addressed by incorporating heterogeneity into processor architectures. Heterogeneous processing elements are able to improve efficiency through specialization: computations that match the specialized processing elements' capabilities can be accelerated, and units not currently active can be turned off to save power. Examples already being developed in the computing industry include graphical processing units (GPUs), vector- or media-functional units similar to the SSE4 instruction set, and encryption units, as well as highly parallel coprocessors such as Intel's Xeon Phi.

Future datacenter architectures will likely resemble that shown in Figure 1, with multiple processors (each of

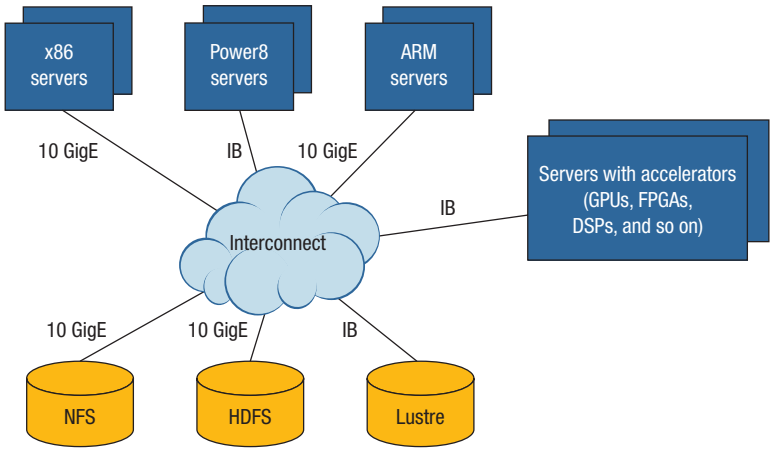

Figure 1. Typical future heterogeneous datacenter architecture. Such centers will contain multiple components: specialized servers and accelerators, including graphical processing units (GPUs), field-programmable gate arrays (FPGAs), and digital signal processors (DSPs); varied storage systems such as a network file system (NSF), Hadoop distributed file system (HDFS), and the like; and flexible interconnects. GigE: Gigabit Ethernet; IB: InfiniBand.

TABLE 1. Heterogeneous datacenter architectures: support requirements for three cloud service models.

Service model	Description	Heterogeneous datacenter architecture support required
Infrastructure as a service (IaaS)	Users provision virtual machines	Heterogeneity exposed to users Bare metal provisioning Virtualized processors, accelerators, networking, and storage
Platform as a service (PaaS)	Programmers target API; framework allocates resources	Heterogeneity may be exposed to programmers Framework manages and schedules heterogeneous resources
Software as a service (SaaS)	Application allocates resources or is developed on top of PaaS	Heterogeneity not visible to users Application or back-end manages heterogeneous resources

which may also have heterogeneous internal components), accelerators, interconnects, and storage systems that, together and individually, provide greater efficiency for specific applications or in particular scenarios. Companies like Microsoft and PayPal that depend on large-scale datacenters are investigating heterogeneous processing elements like these to improve product performance.[2,3]

Developing cloud computing technology compatible with datacenter heterogeneity will require finding ways to optimally exploit varied special-purpose processing elements without losing the advantages of abstraction. To this end, each of the three main cloud services models faces various challenges, as summarized in Table 1.

INFRASTRUCTURE AS A SERVICE

At the lowest level, infrastructure as a service (IaaS) exposes physical and virtual resources to the end user. Virtual machines (VMs) and bare-metal provisioning offer nearly complete OS instance control.

Traditionally, virtualization has imposed a high overhead for performance-sensitive workloads. Today, however, technologies such as single-root I/O virtualization (SR-IOV) and peripheral component interconnect (PCI) passthrough enable direct access to accelerators and networking devices, typically with overhead of 1 percent or less.[4]

Still, as datacenters become more heterogeneous, IaaS deployments will have to expose increasingly varied components, like those shown in Figure 1. Extending homogeneous cloud flexibility to heterogeneous IaaS deployment requires further research in several areas:

- optimal tradeoffs in virtualization performance and functionality (security vis à vis isolation, for example),
- sharing schemes for compute accelerators,
- scheduling techniques to determine job assignments for most efficient resource allocation,
- power and utilization optimization techniques,
- migration mechanisms for jobs having state in accelerators as well as in host processors, and
- cost and prioritization schemes.

Finding ways to exploit new interconnect technologies, such as software-defined networking, and parallel file systems in the context of heterogeneous compute elements also presents interesting research opportunities.

PLATFORM AS A SERVICE

At the level of platform as a service (PaaS), heterogeneity is necessarily

exposed to the framework; it may also be exposed to the programmer, or it may be hidden by libraries or back ends that target heterogeneity. Goals for future research include

- heterogeneity-aware scheduling at the platform level,
- heterogeneous resource allocation among multiple platforms or frameworks sharing the same datacenter,
- software architectures for accelerated libraries, and
- frameworks for application programming that may or may not expose heterogeneity to the programmer.

An example of research targeting programmability improvements for heterogeneous hardware is Microsoft's Catapult framework. This software–firmware interface and implementation for a field-programmable gate array accelerator was designed to improve the performance of the Bing search engine.[3] It provides a valuable use case in exploiting heterogeneous hardware for a commercial datacenter application. Deployed on Bing production servers, Catapult improved Bing's page-ranking throughput by 95 percent per server.

SOFTWARE AS A SERVICE

The software as a service (SaaS) model provides developers the most flexibility because heterogeneity can be hidden within the application software and not exposed to end users. Still, developers building SaaS platforms must keep in mind heterogeneous architectures like those that IaaS and PaaS deliver, and so must address issues involving implementation portability and scalability. Challenges in this area will likely be specific to the software service under development, but will involve making engineering choices about whether to use existing IaaS and PaaS interfaces or to devise custom implementations that target heterogeneity.

CALL FOR COLUMN CONTRIBUTIONS

We welcome short articles (1,200 to 1,500 words) for publication in this column that discuss your ideas for advancing cloud computing or share your experiences in harnessing the cloud. We also solicit articles on topics such as fog computing, cloudlets, cloud forensics, cloud aggregation and integration, service level agreements, and legal issues. Send your proposal/submission to the Editor, San Murugesan, at cloudcover@computer.org.

Processor component heterogeneity is inevitable if we're to continue compute performance improvement. Software development for the cloud will have to target that heterogeneity. Improving virtualization performance in mainstream microprocessors and VM managers can enable software stacks to efficiently support server architectures as they become more heterogeneous. And while provisioning and programming models for heterogeneous hardware are still in early stages, projects like Catapult demonstrate their promise.

Many challenges remain, but ultimately cloud computing will both benefit from and contribute to the improved compute efficiencies and capabilities that have driven IT over the past five decades and should continue into the future.

REFERENCES

1. H. Esmaeilzadeh et al., "Dark Silicon and the End of Multicore Scaling," *Proc. 38th Int'l Symp. Computer Architecture* (ISCA 11), 2011, pp. 365–376.
2. A. Putnam et al., "A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services," *Proc. 41st Int'l Symp. Computer Architecture* (ISCA 14), 2014, pp. 13–24.
3. R. Quick and A. Kolster, "Real-Time Analytics?," *Emergent Use Cases in High-Performance Data Analysis: 54th HPC User Forum*, IDC, 2104; www.idc.com/getdoc.jsp?containerId=251976.
4. J.P. Walters et al., "GPU-Passthrough Performance: A Comparison of KVM, Xen, VMWare ESXi, and LXC for CUDA and OpenCL Applications," *Proc. 7th IEEE Int'l Conf. Cloud Computing* (CLOUD 14), 2014, pp. 636–643.

STEPHEN P. CRAGO is deputy director of Computational Systems and Technology at the University of Southern California (USC) Information Sciences Institute, and holds a joint appointment as a research associate professor in the USC Ming Hsieh Department of Electrical Engineering. His research interests include heterogeneous computing and high-performance and embedded cloud computing. Contact him at crago@isi.edu.

JOHN PAUL WALTERS is a project leader and computer scientist at the USC Information Sciences Institute. His research interests include cloud computing, multicore and accelerator programming, and fault tolerance. Contact him at jwalters@isi.edu.