A High-Performance Computing Forecast: Partly Cloudy

Cloud computing is emerging as an important computational resource allocation trend in commercial, academic, and industrial sectors. Yet, because the business model doesn't currently meet all the needs of high-performance computing (HPC)—the demands of capability computing, for example—the relationship between clouds and HPC suggests a partly cloudy forecast.

Cloud computing is the latest and perhaps the most dramatic trend in advanced computing paradigms since the introduction of commodity clusters, which have dominated high-performance computing (HPC) for more than a decade. Clouds offer an amorphous distributed environment of computing resources and services to a dynamic distributed user base. Like clusters, cloud computing exploits economies of scale to deliver advanced capabilities. Unlike clusters, cloud resources are nonspecific and provide basic capabilities but guarantee neither identical properties from run to run nor high availability of specialized system types. Clouds favor generic services and users and, where appropriate, benefit both. The cloud concept amortizes independent and disparate system deployment costs, distributed nationally and potentially internationally across an equally distributed and diverse demand base. Through potentially automated real-time negotiation of user-task allocation to provider services, cloud providers anticipate that they can achieve the win-win condition of high user availability and high supplier utilization, which should benefit all via economy of scale. But HPC exhibits a range of computing needs—some aligning with general processing requirements and others imposing unique demands—and scale. To what extent can the developing cloud front contribute to the challenges of those high-end workflows critical to science, technology, national security, and social applications? Here, we consider the field of HPC, its diverse computing needs, and the degree to which cloud computing can both satisfy and potentially benefit it.

Although supercomputing's focus is certainly pushing the speed frontier—now, beyond a petaflops of sustained performance (Linpack Rmax)—HPC involves many processing requirements that demand enhanced performance along dimensions other than raw speed. In fact, it's through an examination of the total HPC workflow that we can delineate the distinction between those compute modalities that cloud computing can and can't serve well. We suggest the following possible categorization of the HPC computational modalities, from the most demanding to the most general:

- capability computing, which applies the entire machine to a fixed-size problem to reduce response time, a property known as *strong scaling*;

1521-9615/09/$25.00 © 2009 IEEE
Copublished by the IEEE CS and the AIP

Thomas Sterling and Dylan Stark
Louisiana State University

- cooperative computing, which focuses on increasing problem size while maintaining response time on a given resource, a property known as *weak scaling*;
- capacity computing for job-stream throughput;
- data management for massive datasets;
- dataset analysis and visualization;
- support for HPC administrative workloads; and
- network communication across distributed institutions and agents.

Examining this set of distinct modalities reveals the valuable ways in which cloud computing might contribute to HPC. At the same time, we see cloud computing's very real limitations in meeting some of the more stringent needs. This article pursues such exploration by reviewing the nature of each form and the relevance of clouds to it.

Cloud Computing

Clouds originated in the industry's data center. The vast amounts of data collected by Google and Amazon precipitated a new data-processing model that fit both the industry's data analysis and its service-oriented business needs. Cloud computing is a flavor of distributed computing that supports on-demand Internet delivery of services for data-centric applications, combined with a pay-per-usage charge model.[1–3] In this model, the typical application comprises a thin client code local to the end user and a main server code local to the data-compute resource. The user application requests resource allocations from the cloud, and the provider generates a virtual system image that satisfies the resource request. The user pays only for resources consumed, not for operational infrastructure costs.

The growing excitement surrounding cloud computing stems from increasing expectations that it will fulfill the promise of "light-switch computing," whereby end users are as close to the computation that will drive their applications as they are to the electricity that drives their appliances. We feel strongly that cloud computing will come close to this goal by leveraging economy of scale to reduce costs, by amortizing resources to increase availability and utilization, and by fielding industrial systems to increase service reliability. Two reasons why this approach makes financial sense are that the barrier to entry is lowered because users can forgo capital expenditures for infrastructure and operation, and providers receive better return on infrastructure investments because of increased resource use.[4] Finally, the focus on data centers and massive datasets means that cloud computing targets a very real need that's not already directly addressed.[5]

Cloud computing doesn't come without some concerns: hype devalues the cloud brand, network and host system virtualization reduces service-quality guarantees, and resource sharing presents privacy and security dilemmas. The industry has already stretched the cloud computing moniker to encompass almost all applications that use the Internet, from Web-based email such as Gmail to peer-to-peer networks such as BitTorrent. Essentially, any Internet application in which the majority of the work is done at a remote location is cloud computing. But such generalizing begs the question: why is cloud computing acceptable and vaporware isn't?

As a technical term, cloud computing derives from the use of bubble diagrams to depict parts of a system design for which a more detailed definition is irrelevant or unknown. In a similar fashion, the cloud user doesn't worry about the specifics of the cloud infrastructure and, instead, focuses only on the virtualized environment that the cloud vendor provides. This abstraction hides service-quality guarantees regarding host computing capabilities and network performance. Virtualization limits knowledge of the underlying system details and incurs performance costs. With limited knowledge, the user might not be able to optimize an application to the extent possible with a local dedicated installation. Furthermore, providers might not even guarantee they'll use the same underlying hardware per identical resource request,[3] thus making it harder for developers to understand application performance characteristics across similar runs. Likewise, increased sharing of system resources by concurrent users will strain the available network bandwidth. A study by Constantinos Evangelinos and Chris Hill[5] concluded that a virtual compute cluster formed using the commercial Amazon EC2 service could provide similar capability to a low-cost cluster that would traditionally be available in a non-HPC organization, but that commercially available network infrastructures were "one and two orders of magnitude inferior to big computer center facilities."

Finally, resource amortization implies data colocation from different—possibly hostile—organizations, which leads to concerns over privacy, security, and reliability.[2,4,6] The user must trust the provider to ensure that other users of the same cloud can't access a user's data and code. Issues with Internet connectivity will disrupt access to the cloud, cutting off users from their data and applications. An attack focused only on the cloud's Internet connection would be as effective in disabling the cloud as a direct attack on the computer systems themselves. Although these concerns aren't unique to cloud computing, they're greater in this new computing environment and will serve as major limiting factors in its adoption.

Cloud Strengths and Weaknesses

Using a cloud, a client outsources work to remote resource providers and thereby avoids the initial and sustained costs and complications of locally deployed systems. Clouds provide rapid access to new resource capabilities without incurring the delays implicit in upgrading and expanding local resources and personnel to manage them. They also allow rapid downsizing when demand dissipates without the costly process of on-site resource decommissioning or, worse, personnel reductions. But the cost advantage goes far beyond these obvious factors by allowing real-time supply and demand to work in favor of both user institutions and supplier facilities. Through automatic distributed negotiation protocols, cloud environments can provide low cost and high availability to user institutions, regardless of their geographical location, while providing high usage. Clouds also provide a natural environment for creating, managing, and accessing large distributed datasets as well as management of distributed raw data streams from sensors.

To what extent can the developing cloud front contribute to the challenges of those high-end workflows critical to science, technology, national security, and social applications?

Nonetheless, the cloud computing model, in spite of its promise, either imposes constraints in conflict with some HPC requirements or simply fails to adequately support them. Among these constraints is the underlying hardware architecture virtualization, which is valuable for generic usage of diverse cloud resources. Such resources generally provide portability but obstruct targeting algorithm optimizations to specific hardware structures, as is typical of HPC applications. The time-critical overhead that virtualization layers add further degrades the performance efficiency and scalability of some HPC workloads. Another performance issue related to clouds is that users share resources among multiple tasks for both computational and networking functionality. The resulting resource contention inserts sporadic and unpredictable delays, further degrading performance and making optimizations more difficult. Beyond performance are the critical issues of security and reliability. Much data is highly sensitive, such as intellectual property, competitive planning information, or highly classified intelligence from mission-critical agencies with strong national security responsibilities. In these cases, users won't trust remote networking, storage, and processing resources, no matter how well-intentioned they assume the encryption and other implemented measures to be. Therefore, such organizations are unlikely to employ clouds for these purposes, which comprise a significant portion of HPC activity. Similarly, clouds might not provide sufficient reliability to adequately minimize risk—a particularly sensitive issue in time-bounded applications. Again, dedicated systems are more likely the preferred platform in these cases.

Network Communications

Networking is critical to HPC facility operations. The availability of network infrastructure enables—and potentially limits—collaboration among geographically distributed groups. This is also true for computing systems that support execution of distributed tasks (for example, capacity computing, a distinct category that we discuss later). Because the Internet is a key component of the cloud computing model, this new computing regime will exacerbate any pre-existing limitations in the network infrastructure. The ability to manage costs and acceptable application performance response times will determine operational effectiveness. The network will determine the distance between the data and the computation, which means that in the cloud model, if the bandwidth is low, the user must procure additional data storage near the computation. This increased reliance on data communication will likely be the first deciding criterion for whether an organization will adopt cloud computing.

HPC Administrative Workload Support

General administrative tasks are essential to the daily operations of any HPC facility. The work doesn't require the use of specialized or dedicated resources and mostly comprises email, Web management, and general database-supported operations, such as information retrieval and distribution. This class of work is directly applicable to a cloud computing service model, where Internet-based email and database services are already common. But the use of clouds at this level doesn't necessarily follow. The cost of purchasing and operating the small set of workstations or enterprise servers required to support such services isn't prohibitive. If cloud computing is to prove useful in this area, it'll have to show a cost reduction, possibly through lowering the overhead of software maintenance or the per-head cost of software licenses.

Data Analysis and Visualization

Offloading tasks directly associated with computational science, such as data analysis and visualization, is compatible with cloud services in certain cases. This is particularly true for smaller organizations that don't have the full set of software systems needed to support local applications. Occasionally, availability of mid-scale hardware resources, such as enterprise servers, might be useful as well, if queue times don't impede fast turnaround. We can expand this domain to include frequently introducing new or upgraded software packages not readily available at the local site, even if open source. For the cases in which independent systems vendors provide such software, the cost of ownership or licensing might exceed the budget or could be difficult to justify, given the level of intended use. Cloud clients can find preferable incentives for using such software. Cloud availability also removes the need for local expertise in installing, tuning, and maintaining such arcane packages—this is particularly true for small groups or individual researchers. However, a recurring theme is that HPC users tend to be in environments that incorporate high levels of expertise—including motivated students and young researchers—and therefore are more likely to have access to such capabilities. In this case, an individual's peculiarities and his or her situation determine cloud use.

Dataset Management

Although the community measures HPC in flops (floating point operations per second), it's as sensitive, sometimes even more so, to bytes. Much science is data oriented, comprising data acquisition, generation, organization, correlation, archiving, mining, and presentation. Massive datasets, especially those that are intrinsically distributed among many sites, are a particularly rich target for cloud services. Maintenance of large tertiary storage facilities is especially difficult and expensive, even for the most facilities-rich environments. Data management is one area of HPC in which commercial enterprises are significantly advanced—even with respect to scientific computing expertise—and the community applies significant commercial investment to this area compared to the rarified boutique scientific computing community. One very important factor is that confidence in data integrity of large archives could ultimately be higher among cloud resource suppliers because their potentially distributed nature both removes issues of single-point failure (hurricanes, lightening strikes, floods) and enhances their ability to deploy substantial investment due to economy of scale. But one, perhaps insurmountable, challenge might impose fundamental limits in using clouds for data storage for some mission-critical HPC-user agencies and commercial research institutions, such as data security. Because data leakage or corruption could compromise national security or intellectual property protection, it's implausible that organizations will trust such data to remote and sometimes unspecified service entities, no matter what the strength of putative guarantees.

Capacity Computing for Job-Stream Throughput Workflows

A substantial component of the HPC workload falls into the general class of capacity computing, also referred to as *throughput processing* or *job-stream scheduling*. Although not as challenging as the more tightly coupled application classes of message-passing (cooperative or capability computing), throughput computing serves a wide array of application needs in the HPC domain. Capacity computing is characterized by a workflow of essentially independent jobs or processes that execute concurrently. Each is allocated a dedicated processor core and memory with access to global I/O resources. You don't have to perform the jobs that make up the set in any specific order, with the only shared namespace being that of the file I/O in which the job acquires initial arguments and deposits its results. Throughput computing is valuable for parametric studies in which the same job runs many times but with different input datasets on different machines and therefore potentially in parallel—for example, the user can accomplish Monte Carlo simulations—an important HPC application family—by capacity computing techniques. Visualization of time-varying phenomena is yet another example of throughput computing, with each frame rendered independently of the others to make up the image sequence. Condor[7] is a scheduler and distributed-resource manager that effectively handles throughput computing workloads.

Capacity computing is one of the major areas in which cloud environments could have substantial impact on HPC science and technical applications. The potential problems caused by varying system types, virtualization overhead, potentially inadequate system networking, and parallelism that could result from using cloud environments is largely mitigated by the very coarse workloads, lack of need for coordination mechanisms in an independent task set, and almost no intertask communications. The one challenge to throughput computing via clouds is the need for security for some user job sets. As in cases where the information is highly sensitive, such as that related to national security, intellectual property, or business and market planning, institutions are uncomfortable (or forbidden) to exploit non-fire-walled resources such as cloud environments—including capacity computing.

Cloud computing's impact on this HPC domain is significant. It can reduce costs and increase operational efficiency in HPC institutions by outsourcing the deployment, operation, maintenance, and management of resources dedicated to capacity applications. Clouds can provide substantial flexibility for such institutions by enabling a rapid increase in resources for peak demand and burst job scheduling requirements. They can also allow such rare high levels of throughput without having to populate local machine rooms with the scale of systems needed for the maximum processing requirements. Clouds used for throughput computing can also dedicate local machine procurements to the more stringent demands of capability class machines making the major investments where they're required most. Clouds can also provide access to the most up-to-date equipment. Thus, cloud resource providers are more likely to rapidly deploy such state-of-the-art systems with the latest processors for whom demand across the national and international community is assured. Ultimately, the promise of cloud environments for HPC in the realm of capacity computing is substantial and might have a significant enabling effect on the HPC community.

Cooperative Computing with Weak Scaling

The remaining two HPC computing modalities are far more difficult to support via the emerging cloud model. They represent the primary methods of large-scale computer usage, which is HPC's hallmark: capability computing, which imposes the strictest demands on scalability, and what we refer to as cooperative computing, which constitutes the most widely used programming model for HPC with a weaker scaling property. Both modalities rely on increased system size—usually measured in number of nodes—to increase performance (for example, flops) for a single parallel application. Capability computing (described later) demands strong scaling so that users can reduce a fixed-size problem's execution time proportional to the increase in system scale. Cooperative computing also increases a single application's performance with increased system scale but permits the problem size to grow proportionally with system scale.

MPI programming typifies cooperative computing for HPC, which reflects the "message passing" model; a more formal variant is the communicating sequential processes (CSP) model. Although not explicitly limited to this usage pattern, a typical program is a distributed set of similar processes, each statically allocated to a unique processor (core), coordinated with global barriers, and cooperating through the exchange of messages containing intermediate data results. The BSP (Bulk Synchronous Protocol) control-flow pattern further narrows this SPMD (single-program, multiple-data) strategy in a distributed namespace: each process has an independent-user namespace decoupled from the other cooperating processes. For many distributed applications, this form minimizes performance degradation due to latency, overhead, contention, and starvation, largely by avoiding these challenges through coarse-grained concurrency and manual (compile time as opposed to runtime) locality management.

Perhaps the most interesting area of HPC is how cooperative computing relates to cloud environments because in some cases, cloud computing might be able to serve this subdomain, but in other cases, it probably can't do so effectively. Clouds could sufficiently meet the general performance needs of the cooperative applications that use the message-passing models that are uniform across coarse-grained processes and that exhibit unity-stride access patterns to local data of regular structure (for example, partitioned dense matrices). This could be true even with virtualization in place or when users share resources with other unknown user workloads. In this class of computations, the overhead, virtualization, and latency variation among nodes might consume only a small portion of the total execution time, which the local processing activities would dominate. With sufficient specification control when soliciting commodity cluster resources—for example, number of nodes, number of cores per node, amount of memory per core, and minimum bisection bandwidth between nodes—compute-intensive applications might obtain adequate performance in a cloud.

Cooperative computing applications that have more stringent performance requirements probably won't benefit from clouds because they demand higher-caliber resources than generally available, or they rely on user optimizations based on specific a priori hardware knowledge. Resources such as the Cray XT3/4/5 and IBM BG/L/P/Q systems probably won't be widely available on demand, and their allocation is usually tightly controlled—these constraints aren't consistent with the cloud model. Even large commodity clusters might not be appropriate for clouds and therefore not available for cooperative computing applications.

This general class's applications that are more irregular in time and space also might not be able to exploit cloud resources to good effect. Large-scale applications, even those optimized for a specific hardware architecture, could exhibit very low floating-point efficiency—measured as the ratio of sustained performance to potential peak performance at less than 10 percent—and require weeks or months to complete with potentially thousands of dedicated cores. Further losses of performance due to the generality of cloud resources would be unacceptable in many cases. At some future time, autonomic frameworks that exploit autotuning techniques can narrow this disparity between efficiency need and delivery.

One final requirement that might further inhibit cloud environments for HPC applications, even of the cooperative type, relates to I/O bandwidth. External I/O can become a serious bottleneck to application performance if not balanced with application needs. This kind of support is complex because it strongly involves network bandwidths, disk speeds, buffering, and contention for these resources with other concurrent demands. Checkpoint and restart requirements for purposes of long-term reliability can impose further demands on I/O bandwidth, which, if not available, might seriously degrade overall delivered performance. Thus, I/O could further reduce the value of cloud computing to HPC users.

Capability Computing with Strong Scaling

The most challenging HPC class is capability computing. In its strictest sense, it implies strong scaling in which the execution time of a fixed-size application is reduced as system scale increases. This measure represents an application's only true speedup, in contrast to cooperative or capacity computing. Capability computing imposes the most stringent demands on minimizing overhead, exposing application parallelism at many levels, managing memory and network contention, and avoiding or hiding memory access and system-wide communication latencies. Conventionally, system architectures, optimized algorithms based on detailed hardware knowledge and explicit hardware control, and specialized runtime-system software address such issues. Another aspect of such systems is that they generally include the extremes in scale and peak performance and therefore cost as well. Petaflops-scale systems can incorporate millions of cores and hundreds of terabytes of main memory. Costs can be in the many tens or even hundreds of millions of dollars, even though these systems are largely made of commercial mass-market components such as microprocessors and DRAM. Their unique designs and limited market preclude economies of scale, aggravating ultimate deployment price. Among the factors contributing to their uniqueness is the need for interconnect networks with very high bisection bandwidth and very low latency. Additional specialty networks, such as combining networks for global synchronization and reduction operators, can only make this situation worse even as they improve certain kinds of performance. Finally, according to the November 2008 Top500 list's top 10 entries (www.top500.org/lists/2008/11), capability computing systems reflect the most rapid evolutionary changes in form, function, and scale.[7] Given this rapid evolution, these systems require repeated and frequent application code adaptation. Ironically, even as such systems are often oversubscribed in usage, they're limited in the domain and user community. Their extremes in cost and infrastructure requirements greatly constrain their installed base even as they deliver the most dramatic operational properties.

Clouds can provide substantial flexibility for HPC institutions by enabling a rapid increase in resources for peak demand and burst job scheduling requirements.

As a critical (perhaps defining) element of the HPC domain, capability computing is unlikely to be well-served by cloud computing environments.

This mismatch is due to several factors. The basis of cloud computing is the general, almost generic, utility of offered resources that can have the widest cross-cutting application to the broadest user community. This generality isn't the case for extreme-scale capability systems; in fact, quite the contrary. Although organizations can use high-end systems for the more routine computing tasks, they're not cost competitive with commodity clusters, enterprise servers, or single-user workstations. Therefore, a cloud provider is unlikely to invest in a relatively high-cost specialized system because it doesn't coincide with the optimal cloud business model. And users and their institutions won't rely on clouds for capability computing to support and sustain their respective mission obligations when such systems aren't widely available via cloud environments. A second factor is the combination of virtualization and loosely defined system types provided to user workloads. These cloud properties obscure hardware structure and operational properties crucial for optimizing the code to the architecture for best performance and scalability. No two successive runs of the same code could expect to get exactly the same hardware. Therefore, code optimized for one assumed target system will probably be deficient in that respect for the next. System time sharing, which is likely to be a major operational strategy, is counter to the needs for optimized codes, controlling both space and time allocation of resources and again not suitable for capability computing tasks. Together, these factors severely limit the effectiveness of future cloud environments for the HPC's capability computing subdomain.

Cloud computing's potential for the particularly challenging domain of HPC is promising. HPC is multifaceted, with each subcategory exhibiting its own service and system demands. We delineated seven such categories and considered each one in terms of the possible relevance of clouds to satisfy their requirements and deliver cost and operational advantage to HPC end-user institutions. In fact, many application types in the overall HPC workflow are well-suited to the near-term exploitation of cloud services. Furthermore, institutions that take advantage of clouds might benefit substantially in operational and cost-effectiveness as well as in flexibility and responsiveness to internal workload demands.

However, many anticipated properties of distributed cloud environments strongly suggest that clouds can only partly address HPC user needs and that some workload subdomains will remain beyond the capabilities of cloud services. Virtualization, uncertainty of hardware structural details, lack of network control and memory access contention, repeatability, and protection and security all inhibit cloud paradigm adoption for certain critical uses. Also, it's unlikely that a general business model, implicit with clouds, will provide the extreme computing and peak performance that's an HPC hallmark. Finally, protected access to such facilities is a potential source of competitive edge for science, market, and national security, and the agencies that employ them will therefore limit or entirely preclude offering such systems to a cloud-covered processing world.

We concede that the analysis we offer here could prove too conservative. Clouds' suggested constraints might, in fact, serve as challenges for cloud research. Improved security techniques could lessen the usage limitations of highly sensitive processing and data. Private clouds can provide a kind of firewall, even in a distributed setting; organizations might also adopt them to support distributed access by select communities to the specialized and highest-end HPC systems in a manner suggestive of the US National Science Foundation's TeraGrid.[8] In any case, however cloud technology evolves, HPC will participate and benefit to a significant degree. But don't assume that clouds will easily replace the HPC systems that organizations currently deploy to provide the most extremes in capability;

rather, the two world views must coexist, seeking benefits from clouds while achieving HPC's mission-critical requirements.

References

1. M. Vouk, "Cloud Computing—Issues, Research, and Implementations," *Proc. 30th Int'l Conf. Information Technology Interfaces*, Univ. Computing Centre, Zagreb, Croatia, 2008, pp. 235–246.
2. I. Foster et al., "Cloud Computing and Grid Computing 360-Degree Compared," *Proc. Grid Computing Environments* (GCE08), 2008, pp. 1–10.
3. M. Klems, J. Nimis, and S. Tai, "Do Clouds Compute? A Framework for Estimating the Value of Cloud Computing," *WeB 2008—The 7th Workshop on e-Business*, 2008.
4. E. Deelman et al., "The Cost of Doing Science on the Cloud: The Montage Example," *Proc. 2008 ACM/IEEE Conf. Supercomputing*, *High-Performance Networking and Computing*, IEEE Press, 2008, pp. 1–12.
5. C. Evangelinos and C. Hill, "Cloud Computing for Parallel Scientific HPC Applications: Feasibility of Running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2," *Proc. Cloud Computing and Its Applications*, 2008; www.cca08.org/papers.php.
6. D. Gannon, "The Computational Data Center—A Science Cloud," 2008; www.extreme.indiana.edu/~gannon/science-data-center.pdf.
7. D. Thain, T. Tannenbaum, and M. Livny, "Distributed Computing in Practice: the Condor Experience," *Concurrency and Computation: Practice and Experience*, vol. 17, nos. 2–4, 2005, pp. 323–356.
8. C. Catlett et al., "TeraGrid: Analysis of Organization, System Architecture, and Middleware Enabling New Types of Applications," *High-Performance Computing and Grids in Action*, Lucio Grandinetti, ed., IOS Press, 2007.

Thomas Sterling *is a professor of computer science at Louisiana State University (LSU), a faculty associate at the California Institute of Technology, and a distinguished visiting scientist at Oak Ridge National Laboratory. His current research interests include the ParalleX execution model and its practical implementation in computer architectures and programming methods. Sterling has a PhD in electrical engineering as a Hertz fellow from MIT. Contact him at tron@cct.lsu.edu.*

Dylan Stark *is a doctoral candidate in the computer science department at LSU and a graduate research assistant at its Center for Computation & Technology. His current research interests include knowledge management, ontologies, and high-performance symbolic computing, with a focus on parallel processing of dynamic directed graph structures. Stark has an MS in systems science from LSU. Contact him at dstark@cct.lsu.edu.*