Cloud Paradigms and Practices for Computational and Data-Enabled Science and Engineering

Hybrid infrastructures that combine high-performance computing (HPC) resources with cloud infrastructures are emerging as attractive platforms for real-world science and engineering applications, and it's important to understand how these applications can effectively utilize such a hybrid infrastructure. In this article, three key usage modes are explored: HPC in the Cloud, HPC plus Cloud, and HPC as a Service, presenting illustrative scenarios in each case and outlining benefits, limitations, and research challenges.

Cloud computing[1] is revolutionizing the enterprise world, much as the Internet did not so long ago. Clouds are fundamentally changing how enterprises think about IT infrastructures, both internally and externally, by providing on-demand access to always-on computing utilities; an abstraction of unlimited resources; a potential for scaling-up, scaling-down, and scaling-out as needed; and possibilities for IT outsourcing and automation. Clouds also offer a usage-based payment model where users essentially "rent" virtual resources and pay for what they use. Cloud services are typically founded on consolidated and virtualized data centers that exploit economies of scale to provide attractive cost-benefit ratios. Despite being in its early stages, cloud computing is already reshaping the IT world—in fact, according to *The Wall Street Journal*, four out of five businesses are moving or planning to move some of their business functions to cloud services. A recent report by Gartner estimates that cloud services will be a $150 billion industry by 2015 (www.gartner.com/DisplayDocument?id=1378513).

At the same time that cloud computing is redefining IT, extreme data and compute scales are transforming science and engineering research by enabling new paradigms and practices—those that are fundamentally collaborative and information- and data-driven. Large-scale computational and data-enabled science and engineering (CDS&E) applications are providing unprecedented opportunities for understanding and managing natural and engineered systems, and providing unique insights into important problems. Recognizing this data- and compute-driven transformation of science and engineering in the 21st century, the US National Science Foundation is rethinking the national cyber infrastructure framework (CIF21; www.nsf.gov/funding/pgm_summ.jsp?pims_id=504730).

It's only a matter of time before cloud services join more traditional research cyber infrastructure components—such as CIF21-related high-performance computing (HPC) systems, clusters, and grids—in supporting scientific exploration and discovery. Analogous to their role in enterprise IT, clouds can enable the outsourcing of many of the

1521-9615/13/$31.00 © 2013 IEEE
Copublished by the IEEE CS and the AIP

Manish Parashar, Moustafa AbdelBaky, Ivan Rodero, and Aditya Devarakonda
Rutgers University

tedious aspects of research and education, such as deploying, configuring, and managing infrastructure, and can enable scientists to focus on the science. Cloud services and the associated standardization can also improve productivity, facilitate the sharing of research results, and enable the reproducibility of associated computations. Furthermore, and once again analogous to the enterprise space, clouds can democratize access to computational and data resources by providing access to researchers who lack adequate local infrastructure, which has been shown to significantly impact research productivity (www.nersc.gov/assets/Events/MagellanNERSCLunchTalk.pdf). In fact, a recent survey of Department of Energy users conducted by the Magellan team found that the top motivations for users' movement to the cloud was ease of access to computing resources (cited by 79 percent), the ability to control software environments (59 percent), and the ability to share the setup of software and experiments with peers (52 percent; see www-935.ibm.com/services/in/cio/pdf/oic03013usen.pdf). However, it's also critical to look beyond the benefits of outsourcing and understand application formulations and usage modes that are meaningful in a hybrid HPC/grid plus cloud cyber infrastructure—for example, how emerging data- and compute-intensive application workflows can effectively utilize these resources, and also how this hybrid cyber infrastructure can enable new practices in science and engineering.

Here, we explore such usage modes, and how a hybrid HPC/grid plus cloud cyber infrastructure can be effectively used to support real-world science and engineering applications. We also discuss our initial exploration of different usage modes, presenting illustrative scenarios in each case and outlining benefits, limitations, and research challenges. We should note that in this article we use the term *HPC* to refer to more traditional computing resources, such as the production resources that are part of Extreme Science and Engineering Discovery Environment (XSEDE), Open Science Grid (OSG), and other academic computing centers (including clusters, grids, tightly-coupled systems, and supercomputers), as well as their usage modes and applications.

Science and Engineering with Clouds

As noted earlier, clouds are rapidly joining HPC, clusters, and grids as viable platforms for scientific exploration and discovery; understanding application formulations and usage modes for science and engineering applications is important.

CDS&E Applications and Requirements

The wide range of CDS&E applications have been broadly classified, based on their requirements and execution behaviors, into HPC, high-throughput computing (HTC), and many-task computing (MTC). HPC applications are tightly coupled with large amounts of interprocessor communication, and typically require considerable computing power. In addition to large scales, these applications also require fast interconnects among processors. On the other hand, HTC applications are typically loosely coupled, where communication between processors is limited or non-existent. HTC applications also require large amounts of computing, but special high-speed interconnect isn't essential. Finally, MTC applications are a hybrid of the previous two classes. For example, an MTC application might consist of loosely coupled tasks, where each task is a tightly coupled HPC application. Ensemble applications are an example of such MTC applications. They require fast interconnects, can scale to extremely large scales (petascales and even exascales), and involve heterogeneous and dynamic workflows that can effectively use elastic resources.

CDS&E Applications and Clouds

Cloud computing services can support CDS&E applications in multiples ways. They can provide resources for running such applications when local infrastructure isn't available. They can also supplement existing systems to provide additional capacity or complementary capabilities to meet heterogeneous or dynamic needs. For example, clouds can serve as accelerators or provide resilience to scientific workflows by moving the workflow execution to alternative resources when a failure occurs. The cloud abstraction's simplicity can alleviate some of the problems that scientific applications face in current HPC environments. For example, the increasingly important and growing MTC class of applications can benefit from the ease of use, the abstraction of elastic and/or readily accessible resources, and the ability to easily scale up, down, or out. Finally, cloud computing will not only help scientists address today's problems more effectively, but will also let them explore new ways of formulating their applications using, for example, the abstraction of on-demand access to elastic resources provided by the cloud model.

In the remainder of this article, we explore these ideas in more detail and investigate how the cloud abstraction can be effectively used to support real-world scientific and engineering applications.

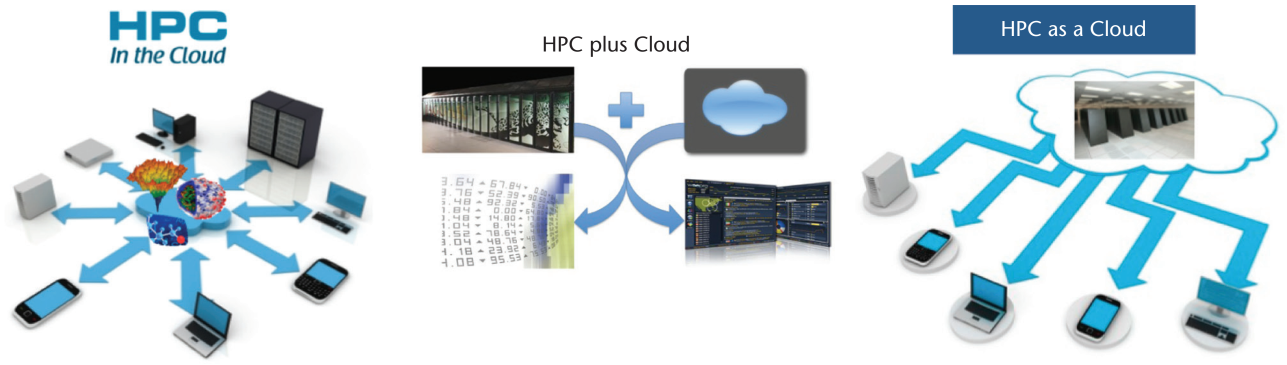

Figure 1. Hybrid high-performance computing (HPC)/grid plus cloud usage modes for supporting real-world science and engineering applications. Each usage mode enables new paradigms that weren't possible before, such as acceleration, ease of use, and on-demand access.

Specifically, we explore three key usage modes (see Figure 1):

- *HPC in the Cloud*, in which researchers outsource entire applications to current public and/or private Cloud platforms;
- *HPC plus Cloud*, focused on exploring scenarios in which clouds can complement HPC/grid resources with cloud services to support science and engineering application workflows—for example, to support heterogeneous requirements or unexpected spikes in demand; and
- *HPC as a Service*, focused on exposing HPC/grid resources using elastic on-demand cloud abstractions, aiming to combine the flexibility of cloud models with the performance of HPC systems.

Each of these approaches has its benefits and limitations, and we believe a clear understanding of the usage modes and associated application scenarios is essential. In particular, we'll explore how cloud models and abstractions can be used to build HPC plus Cloud and HPC as a Service models that are complementary to the HPC in the Cloud approach.

Initial Explorations

We used the CometCloud (www.cometcloud.org) framework developed at Rutgers University to explore CDS&E usage modes on existing cyber infrastructure and public clouds. Specifically, our goal was to explore application formulations and hybrid HPC/grid plus cloud infrastructure usage modes that are meaningful for various classes CDS&E application workflows. Our experiments explored a broad set of federated resources spanning existing public clouds, grids, supercomputing resources, and local clusters. We also explored a broad set of applications classes, including business intelligence, financial analytics, oil reservoir simulations, medical informatics, and document management.

Table 1 summarizes a list of the different applications and platforms explored in this study. The discussion in this section is focused on summarizing our experiments and experiences related to the usage modes outlined in this article, using an ensemble application workflow as the driver. Detailed information and references regarding these experiments can be found elsewhere.[2]

Overview of CometCloud

CometCloud is an autonomic computing engine that enables dynamic and on-demand federation of clouds and grids as well as the deployment and robust execution of applications on these federated environments.[4] It supports highly heterogeneous and dynamic cloud/grid infrastructures, enabling the integration of public/private clouds and autonomic cloudbursts. For example, CometCloud provides dynamic scale-out to clouds to address extreme requirements such as heterogeneous and dynamic workloads and spikes in demands.

The CometCloud programming layer provides a platform for application development and management. It supports a range of paradigms, including MapReduce, Workflow, and Master/Worker/Bag-of-Tasks. The CometCloud autonomic management layer enables the autonomic management of application workflows as well as the federated infrastructure, and ensures that application objectives and constraints are satisfied. Specifically, the autonomic management framework can provision the appropriate mix of HPC/grid and public/private cloud resources based on application requirements and constraints. The framework can also monitor system and application states (such as workload, availability, and delays) and adapt the application and/or the resources to respond

Table 1. Applications and platforms supported by CometCloud.

Applications	Description	Platforms	Description
Value at Risk[2]	Simulates stock prices at time $t + 1$ given prices at time t	Amazon Elastic Compute Cloud (EC2)	Amazon EC2
Medical Image Registration[2]	Determines the spatial alignment between medical images of the same or different subjects	TeraGrid (Extreme Science and Engineering Discovery Environment, or XSEDE)	e-Science grid computing infrastructure
Heat distribution problem using Jacobi Iteration[2]	Calculates heat distribution inside uniform metal plate	FutureGrid Nimbus	Distributed test bed for cloud/grid Open source service package
Asynchronous Replica Exchange[2]	Sampling algorithm to preserve canonical distribution	FutureGrid Eucalyptus	Eucalyptus (open source cloud computing software)
Protein Data Mining[2]	Mines crystal structures and nuclear magnetic resonance using the Protein Data Bank	Rutgers Cluster I	Microsoft HPC cluster
Implicit Parallel Accurate Reservoir Simulator plus Ensemble Kalman Filter (IPARS + EnKF)[3]	Oil reservoir history-matching application with Ensemble Kalman Filter	Blue Gene/P	IBM supercomputer
Decentralized Online Clustering[2]	Proactive self-monitoring of distributed systems	Rutgers Cluster II	Dell Cluster

to changing application requirements or system states (such as by changing the algorithms used or reprovisioning resources). Figure 2 shows a schematic overview of the CometCloud-based autonomic application-management framework.

The Ensemble Kalman Filter Application Workflow

Ensemble applications explore large parameter spaces to simulate multiscale and multiphase models and minimize uncertainty, which is achieved by running thousands of ensembles simultaneously and then assimilating their results. An ensemble filter then analyzes these results before an updated set of realizations is run. Each ensemble can be a traditional parallel HPC application that requires a varying number of processors and fast communication among processors.

In addition, a large and varying number of ensemble members are also typically required to achieve acceptable accuracy, which in turn requires a very large and dynamic pool of HPC resources. The Ensemble Kalman Filter (EnKF) workflow presents an interesting use case due to the heterogeneous computational requirements of the individual ensemble members as well as the overall workflow's dynamic nature.[5]

Figure 3 illustrates the variability between stages of a typical EnKF-based simulation. EnKF is a recursive filter that can handle large, noisy data—in this case, the results and parameters from ensembles of reservoir models that are sent through the filter to obtain the data's "true state." Because the reservoir model varies from one ensemble to another, the ensemble simulation's runtime characteristics are irregular and hard to predict. Furthermore, during execution, if real historical data are available, all the data from the different ensembles at that simulation time must be compared to the actual production data before the simulations are allowed to proceed. This translates into a global synchronization point for all ensemble members in any given stage, which can present challenges. The end-to-end application consists of several stages, and, in general, the number of models generated at each stage varies in size and duration.

HPC in the Cloud

Current cloud platforms can provide effective platforms for certain classes for CDS&E applications—for example, HTC applications. Several early projects have reported successful deployments of applications on existing clouds.[6–8] Running these applications typically involves using virtualized commodity-based hardware, which is provisioned on-demand at commercial cloud providers such as Amazon Elastic Computer Cloud (EC2) or Microsoft Azure.

A recent technical report by Geoffrey Fox and Dennis Gannon provides an extensive study of

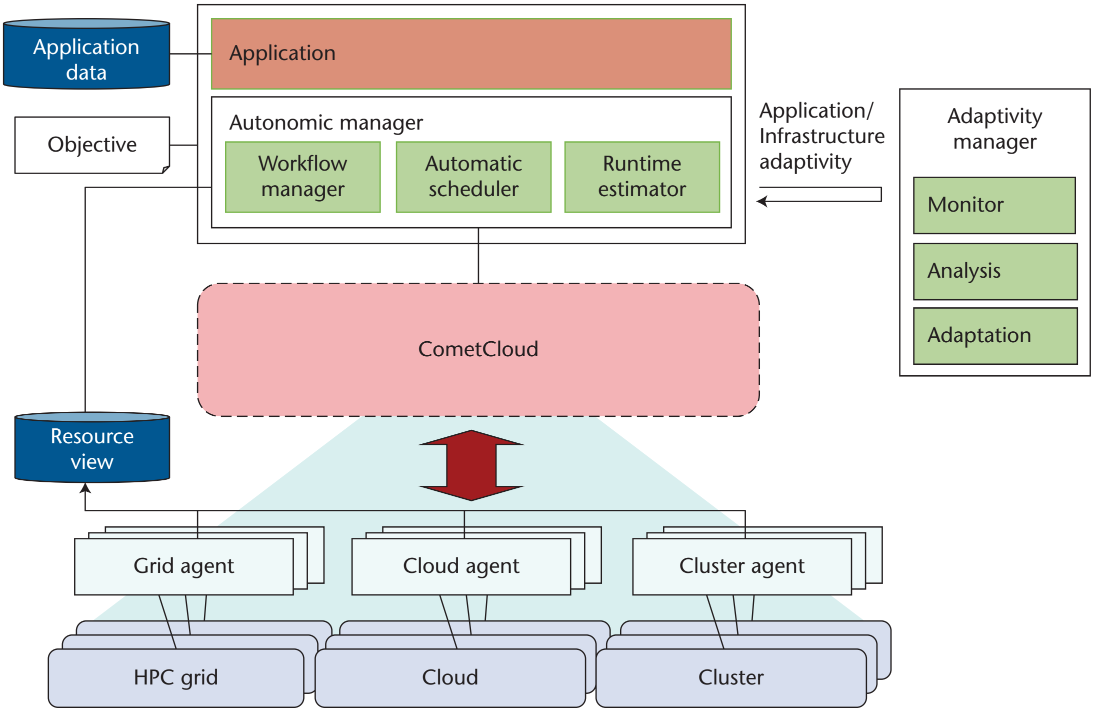

Figure 2. Architectural overview of a CometCloud-based autonomic application-management framework and supported platforms. CometCloud is an autonomic computing engine that enables dynamic and on-demand federation of clouds and grids as well as the deployment and robust execution of applications on these federated environments.[4]

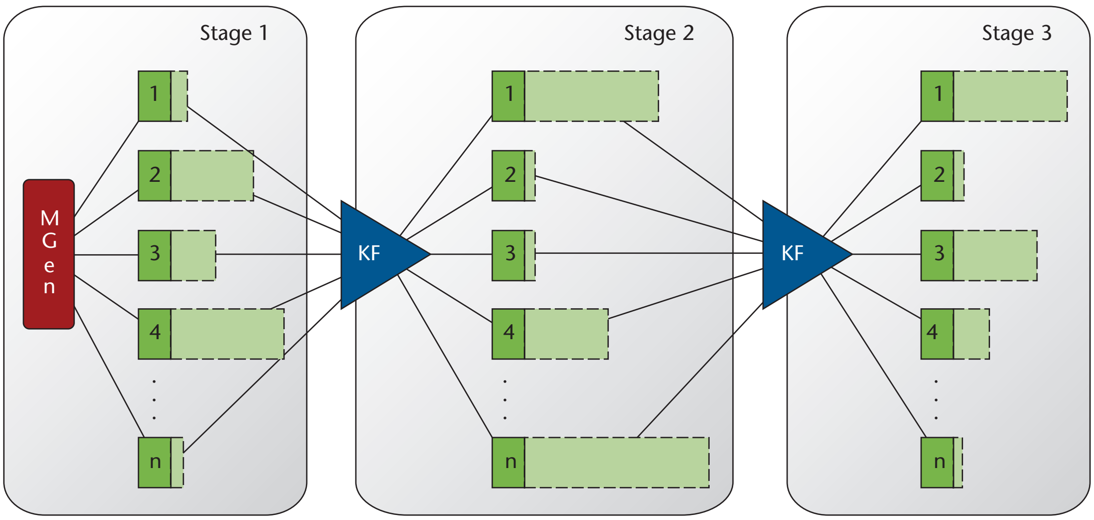

Figure 3. Stages of a typical ensemble Kalman filter-based simulation. The end-to-end application consists of several stages. In general, the number of stages and the number of models generated at each stage can vary in number, size, and duration. Each model in a stage can be a stand-alone application.

HPC applications in the cloud.[9] According to this study, commodity clouds work effectively only for specific classes of HPC applications. Application examples include embarrassingly parallel applications that analyze independent data or spawn independent simulations that integrate distributed sensor data, science gateways and portals, or data analytics that can use MapReduce-like formulations. In general, HPC applications with minimal synchronization and communications requirements, small I/O requirements, and modest scales are well suited to current cloud platforms. In cases

Table 2. HPC as a Service classification.		
Large HPC systems	**Smaller HPC clusters connected together**	
Physical solutions	Physical solutions	Virtualized solutions
Example: Blue Gene/P as a Service (CometCloud)	Examples: Penguin on Demand, Silicon Graphics, and HPCynergy	Examples: EC2 Cluster, Azure, Adaptive Systems, and CloudCycle

where existing application formulations aren't directly suited for available cloud platforms, alternate formulations must be explored before these applications can benefit from cloud services. For example, the asynchronous replica exchange formulation is a novel, decentralized, asynchronous, and resilient formulation of the "replica exchange" algorithm for simulating proteins' structure, folding, and dynamics.[2]

In this investigation, we studied the performance and performance fluctuation of the EnKF workflow on EC2 and FutureGrid.[2] We observed that the underlying hardware impacts performance, as measured by time-to-completion as well as cost. We concluded that reducing the fluctuation on a single instance is a critical step in scaling up to large-core counts and running MPI applications efficiently across several instances. In addition to exploring how CometCloud can support a range of well-suited applications (such as image analysis or value-at-risk computations) on public and private cloud infrastructures, we also explored the Replica Exchange Molecular Dynamics application.[2] This application, which isn't traditionally well-suited for clouds, can be reformulated as an asynchronous and resilient application that we can effectively run on public and private clouds using the messaging and coordination abstractions provided by CometCloud.

HPC Plus Cloud

Running CDS&E applications on more traditional HPC resources and bursting onto a cloud when there's a spike in computing-capacity demand is an attractive approach for heterogeneous and dynamic application workflows. This is becoming even more important as CDS&E applications increasingly explore end-to-end workflows that consist of coupled simulations and integrated I/O and data analytics pipelines. A hybrid cloud plus traditional HPC infrastructure can also support new and potentially more effective usage modes and enable new application formulations that use clouds to achieve acceleration, resilience, or more appropriate cost/power/performance tradeoffs.

Existing efforts that have explored a hybrid infrastructure include Simon Ostermann and his colleagues,[10] who extended a grid workflow application development infrastructure to include cloud resources. They also experimented with the Austrian grid and an academic cloud installation of Eucalyptus using a scientific workflow. Andre Monteiro and his colleagues provide a summary of experiments exploring hybrid infrastructure.[11]

We used CometCloud to explore meaningful usage modes for a hybrid HPC plus Cloud infrastructure. In particular, we used the EnKF workflow and complemented TeraGrid (currently XSEDE) resources with Amazon EC2 public cloud instances.[2] We explored four different usage modes:

- *acceleration*—using clouds as accelerators to reduce the application time to completion;
- *conservation*—using clouds to conserve HPC allocations while satisfying runtime and budget constraints;
- *resilience*—using clouds to handle unexpected situations such as an unanticipated HPC downtime, inadequate allocations, unanticipated queue delays, or the failures of nodes while meeting user objectives; and
- *online analytics/visualization*—using clouds to perform online data analytics or visualization while a complex simulation runs on HPC resources.

HPC as a Service

Because of current commodity clouds' limitations in serving general HPC applications, cloud providers realized the need to provide cloud solutions that are built specifically for HPC applications (for example, hardware with faster processors and interconnects). Some providers have even provided nonvirtualized hardware to deliver the bare-to-the-bone performance that these applications require. This is commonly referred to as *HPC as a Service*—that is, running HPC applications on HPC resources that are exposed as on-demand services using cloud computing abstractions—to take advantage of the cloud model without sacrificing the HPC performance that scientific applications require.

Table 2 summarizes the two main approaches for providing HPC as a Service. The first approach develops abstractions enabling large HPC systems to be provisioned as clouds. We illustrated this approach in previous work using the IBM Blue Gene/P supercomputer.[3] The second approach uses HPC clusters that can be connected to form a large cloud. These HPC clusters can be virtualized or nonvirtualized to provide better performance. The Amazon EC2 Cluster, Azure, Adaptive Systems, and Platform HPC Computing are commercial providers that use the virtualized approach; Silicon Graphics and Penguin Computing on Demand are commercial providers that provide access to nonvirtualized resources. Andrzej Goscinski and his colleagues[12] proposed a similar approach in an academic environment, developing the HPCynergy system to provide HPC as a Service using small clusters.

In an early experiment using CometCloud, we explored how the Cloud abstraction can be effectively used to provide a simple interface for current HPC resources and support real-world applications.[11] In particular, we experimentally validated the Cloud paradigm's benefits, such as ease of use and dynamic allocation, and its application to supercomputers—specifically, an IBM Blue Gene/P system. The CometCloud-based framework essentially transformed the Blue Gene/P into an elastic Cloud, bridged multiple Blue Gene/P systems to create a larger HPC federated Cloud, and supported dynamic provisioning. The exercise demonstrated the elastic Cloud abstraction's ease of use and its effectiveness in improving utilization. The experiment won first place at the Fourth IEEE International Scalable Computing Challenge (SCALE; www.ics.uci.edu/~ccgrid11/SCALE%20Challenge.html). During the experiment, Blue Gene/P resources scaled on-demand from 640 to 22,016 processors, spanning two Blue Gene systems on two continents.

Research Agenda for CDS&E and Clouds

Aggressive cloud computing technology development has resulted in many different classes of cloud services that provide attractive solutions for different types of business applications. It's also clear from previous research and our own experiences that there are real benefits to using clouds and cloud computing abstractions as part of a hybrid cyber infrastructure to support CDS&E. These benefits include, for example, simplifying the deployment of applications and the management of their execution; improving their efficiency, effectiveness, and/or productivity; and providing more attractive cost/performance ratios. Furthermore, clouds and cloud computing abstractions can support new classes of algorithms and enable new application formulations that can potentially revolutionize CDS&E research and education.

However, as we outlined earlier, before CDS&E can fully realize the potential benefits of a hybrid cyber infrastructure that integrates cloud services, several research issues remain. These issues are in addition to core cloud challenges, such as programming models, security, and standardization. Although we envision new usage modes, our research agenda leverages the existing industry standards and best practices that have been defined during the past decade.

Algorithms and Application Formulations for Clouds

A key characteristic of clouds is on-demand access to elastic resources—that is, applications can programmatically access more or less resources to meet changing needs. Having this capability can significantly impact how algorithms are developed and applications are formulated. For example, an application's execution is no longer constrained to a fixed set of resources, and can grow or shrink its resource set based on the science's demands. Thus, the science itself can drive the scale and type of resource involved—based on, for example,

- the levels of refinement required to resolve a solution feature,
- the number of ensembles required to quantify a solution's uncertainty, or
- the type of online analytics services that must be dynamically integrated into the application workflow.

In addition to the usage modes and application scenarios we've discussed, a remaining research challenge is understanding how CDS&E applications can effectively utilize clouds and cloud abstractions as part of a hybrid cyber infrastructure to enable new practices and levels of scientific insights. Research is needed to explore the meaningful science, engineering, and business application scenarios that can take advantage of such hybrid infrastructures; these scenarios include data-intensive, data-driven, sensor-based, and high-throughput applications.

For example, a meaningful HPC plus Cloud use case might consist of simulations with online data analytics/visualization. In such a scenario, exposing the ability to modify goals/configurations to the user based on data analytics feedback will be critical to ensuring impact on the science. In data-intensive computations, for example, the use of online feature tracking lets the scientist adjust application parameters based on an analysis of meaningful features using a public cloud; such an analysis can be performed in a timely manner due to shorter resource provisioning times compared to using a high-end HPC system. We believe that such meaningful scenarios exist in all areas of CDS&E.

Programming Systems and Abstractions

A key remaining research challenge is to develop appropriate programming abstractions and language extensions that can enable CDS&E applications to simply and effectively take advantage of the elastic access to resources and services during application formulation. Furthermore, it might be necessary to define constraints (for example, budgets, data privacy, and performance) to regulate the elasticity, and the programming abstractions should provide support for expressing these constraints so that they can be enforced during execution. Similarly, such annotations can also define possible adaptations, which could then be used to increase performance, manageability, and overall application robustness. Examples here include dynamically increasing the assigned resources to increase the a simulation's resolution under certain convergence constraints, or modifying convergence goals to avoid failure or guarantee completion time.

Cloud service models can also lead to interesting services specialized to CDS&E that provide entire applications or applications patterns or kernels as a service, such as software as a service (SaaS). Furthermore, and arguably more interestingly, they can also export specialized platforms for science as services (PaaS), which encapsulate elasticity and abstract the complexity of the underlying hybrid cyber infrastructure. For example, our own research is exploring the idea of EnKF as a service. Finally, developing effective support tools—such as for debugging, validation, and performance engineering—remains a challenge.

Middleware Stacks and Services

Middleware services will need to support the new CDS&E applications' formulations and services enabled. A key research aspect will be the autonomic management and optimization (in terms of performance, energy, cost, reliability, and so on) of application execution through cross-layer application/infrastructure adaptations. It will be essential for the middleware services to adapt to the application's behavior as well as to the system configuration, which can change at runtime, using the notion of elasticity at the application and workflow levels. Furthermore, appropriate services are needed to provision different types of resources on demand. For example, if we target HPC as a Service and HPC plus Cloud approaches on NSF-funded cyber infrastructure (such as XSEDE, OSG, and FutureGrid) along with commercial clouds (such as Amazon EC2 or Microsoft Azure), we'll need autonomic provisioning and (co)scheduling techniques (including cloud bursting) to support hybrid usage modes. Other important challenges include managing the data that's read in and/or generated by the applications, the clouds' connectivity to the data, and interoperability with and integration of cloud storage models and solutions. Finally, monitoring online data analytics for proactive application/resource management and adaptation techniques will be essential as the scale and complexity of both the applications and hybrid infrastructure grows.

Cloud services are becoming an integral part of the hybrid cyberinfrastructure available to science and engineering applications, and provide complementary capabilities that can potentially have a significant impact on a large range of applications. As a result, understanding meaningful usage modes and rethinking application formulations to effectively utilize such capabilities is essential. This article presented initial research and experiences in this direction. However, as cloud services become more pervasive and better suited to the needs of science and engineering, this will lead to new opportunities for accelerating science and revolutionizing the scientific process—and with it will come a challenging sociotechnical research agenda. CiSE

Acknowledgments

This research was conducted as part of the US National Science Foundation (NSF) Cloud and Autonomic Computing Center at Rutgers University (grant IIP 0758566) and the Rutgers Discovery Informatics Institute. Resources used in this research were made available in part by the NSF through Extreme Science and Engineering Discovery Environment (XSEDE) under award TG-CCR110035 (which is supported by NSF grant number OCI-1053575), and FutureGrid (which is supported in part by the NSF under grant number 0910812), and the National Energy Research Scientific Computing Center (which is supported by the Department of Energy's Office of Science under contract number DE-AC02-05CH11231).

The research presented in this work is supported in part by NSF grants OCI 1339036, OCI 1310283, and DMS 1228203.

References

1. P. Mell and T. Grance, *The NIST Definition of Cloud Computing*, Nat'l Inst. Standards and Technology (NIST) special publication, vol. 800, 2011.
2. M. Parashar et al., "Cloud Paradigm and Practices for CDS&E," *Cloud and Autonomic Computing Center Research Report*, Rutgers Univ., 2012.
3. M. AbdelBaky et al., "Enabling High-Performance Computing as a Service," *Computer*, vol. 45, no. 10, 2012, pp. 72–80.
4. H. Kim and M. Parashar, "CometCloud: An Autonomic Cloud Engine," *Cloud Computing: Principles and Paradigms*, Wiley, ch. 10, 2011, pp. 275–297.
5. S.I. Aanonsen et al., "The Ensemble Kalman Filter in Reservoir Engineering—A Review," *SPIE J.*, vol. 14, no. 3, 2009, pp. 393–412; doi:10.2118/117274-PA.
6. E. Deelman et al., "The Cost of Doing Science on the Cloud: The Montage Example," *Proc. 2008 ACM/IEEE Conf. Supercomputing*, IEEE, 2008, pp. 1–12.
7. 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*, ACM, 2008, pp. 2–34; https://my.cloudme.com/seadog5339/webshare/CloudComputing/Cloud%20Computing/Applications/Cloud%20Computing%20and%20its%20Applications%20-%202008/Paper34-Chris-Hill.pdf.
8. K. Keahey and T. Freeman, "Science Clouds: Early Experiences in Cloud Computing for Scientific Applications," *Cloud Computing and Its Applications*, ACM, 2008, pp. 825–830; http://my.cloudme.com/seadog5339/ws/572011/Cloud%20Computing/Applications/Cloud%20Computing%20and%20its%20Applications%20-%202008/Paper39-Kate-Keahey.pdf.
9. G. Fox and D. Gannon, *Cloud Programming Paradigms for Technical Computing Applications, Proc. Cloud Futures Workshop*, 2012; http://grids.ucs.indiana.edu/ptliupages/publications/Cloud%20Programming%20Paradigms.pdf.
10. S. Ostermann, R. Prodan, and T. Fahringer, "Extending Grids with Cloud Resource Management for Scientific Computing," *Proc. 10th IEEE/ACM Int'l Conf. Grid Computing*, 2009, pp. 42–49.
11. A. Monteiro et al., "Sky Computing," *Proc. 6th Conf. Information Systems and Technologies*, IEEE, 2011, pp. 1–4.
12. A. Goscinski, M. Brock, and P. Church, High-Performance Computing Clouds, Cloud Computing: Methodology, Systems and Applications, CRC Press, 2011.

Manish Parashar *is a professor in the Electrical and Computer Engineering Department at Rutgers University, where he is director of both the Rutgers Discovery Informatics Institute (RDI2) and the US National Science Foundation (NSF) Cloud and Autonomic Computing Center. He's also associate director of the Rutgers Center for Information Assurance. His research interests focus on applied parallel and distributed computing and computational and data-intensive science and engineering. Parashar has a PhD in computer engineering from Syracuse University. He received the IBM Faculty award twice, as well as a US NSF Career award; he is a fellow of IEEE and the American Association for the Advancement of Science (AAAS). Contact him at parashar@rutgers.edu.*

Moustafa AbdelBaky *is a PhD candidate in the Electrical and Computer Engineering Department at Rutgers University; he's also a member of The Rutgers Discovery Informatics Institute (RDI2) and the university's US National Science Foundation Cloud and Autonomic Computing Center, as well as a research intern at IBM T.J. Watson Research Center. AbdelBaky has an MS in electrical engineering and computer engineering from Rutgers University. He's a member of IEEE and the Society for Industrial and Applied Mathematics (SIAM). Contact him at moustafa.a@rutgers.edu*

Ivan Rodero *is an assistant research professor at Rutgers University and a member of the Discovery Informatics Institute (RDI2) and the US National Science Foundation (NSF) Cloud and Autonomic Computing Center. His research interests fall in the broad area of parallel and distributed computing and include high-performance computing, energy efficiency, autonomic computing, grid computing, cloud computing, and data analytics at extreme scales. Rodero has a PhD in computer science and engineering from the Technical University of Catalonia. He is a member of IEEE, ACM, and the American Association for the Advancement of Science (AAAS). Contact him at irodero@cac.rutgers.edu.*

Aditya Devarakonda *is a PhD student at Berkeley College of Engineering. His research interests are in parallel computing, cloud computing, and the design of scalable algorithms. Devarakonda has a BS in electrical and computer engineering from Rutgers University. Contact him at aditya08@cac.rutgers.edu.*