Developing a Cloud Computing Charging Model for High-Performance Computing Resources

Matthew Woitaszek
National Center for Atmospheric Research
Boulder, CO 80305
Email: mattheww@ucar.edu

Henry M. Tufo
University of Colorado, Boulder
Boulder, CO 80309
Email: tufo@cs.colorado.edu

***Abstract*—This paper examines the economics of cloud computing charging from the perspective of a supercomputing resource provider offering its own resources. To evaluate the competitiveness of our computing center with cloud computing resources, we develop a comprehensive system utilization charging model similar to that used by Amazon EC2 and apply the model to our current resources and planned procurements. For our current resource, we find that charging for computational time may be appropriate, but that charging for data traffic between the supercomputer and the storage/front-end systems would result in negligible additional revenue. Similarly, charging for data storage capacity at currently typical commercial rates yields insufficient revenue to offset the acquisition and operation of the storage. However, when we extend the analysis to a capacity cluster scheduled for deployment in the first half of 2010 that will be made available to users through batch, Grid, and cloud interfaces, we find that the resource will be competitive with current and anticipated cloud rates.**

I. Introduction

Infrastructure-as-a-service (IaaS) cloud computing is emerging as a viable platform for scientific computing workloads and has the potential to augment traditional high-performance computing resource providers such as supercomputer centers because of its "infinite capacity" and compelling cost structure provided by the extreme economies of scale involved. The decision to migrate from traditional, often Grid-enabled, resources at supercomputing centers to direct-billed cloud computing resources such as Amazon Elastic Computing (EC2) is now an option available to project investigators and supercomputing center directors alike. Most analyses of cloud computing economics thus approach this migration from the perspective of a user or project, such as a business with an established data center operation, a startup with no infrastructure, or a researcher with a small local cluster and a larger problem. For many of these cases, variability in utilization leads to possible cost savings, especially if the target application can scale both up and down commensurate with demand [1], does not require a high-performance (low latency) network interconnect, and the problem can be engineered to leverage cloud resources [2], [3]. For many scientific computing campaigns, and in particular those requiring loosely coupled resources for limited yet very specific periods of time, the on-demand nature of cloud computing as a utility may be irresistable when compared to large capital investments or the batch-queued cyberinfrasture at national supercomputing centers.

In this paper, we approach the economics of cloud computing from the perspective of a supercomputer center resource provider. This approach yields two lines of inquiry: comparing the cost-effectiveness of our specialized resource to the general-purpose resources available from cloud providers, and determining whether it is advantageous to become a cloud computing resource provider ourselves. Of course, many scientific applications require large numbers of processors tightly coupled with high-performance networks; these high-performance computing (HPC) workloads are still best served by supercomputing centers because commodity cloud resource providers do not invest in HPC interconnects as there is at present no compelling business case to do so. However, many scientific workflows also involve separate processing steps complimentary to HPC activities, such as postprocessing and visualization, that could be effectively run on cloud computing providers as well.

To better understand the economic implications of our procurement and operation of computing systems, and in particular to enable comparisons with commercial cloud resource providers such as Amazon EC2, in early 2008 we developed a charging model for our existing IBM Blue Gene/L supercomputer "Frost". We extended our system's accounting mechanism to record the quantity of data moved to and from the Blue Gene/L rack on a per-job basis and collected this detailed operational data for 324 days. From this data, we examine possible revenue generation first using a naive model assuming amortized costs and full utilization and subsequently using measured computational utilization, data transfer, and data storage data. We identified two major components of our computational workload that could benefit from offloading to cloud computing resources, and our results suggest that outsourcing these workloads would in fact be more cost efficient than running them on the Blue Gene/L if we did not already own the machine. Finally, we calculate a utilization-naive amortized charging model for a replacement supercomputer and conclude that it is sufficiently sized to enjoy the economies of scale required for its internal operation to be less expensive than Amazon EC2 given sufficient utilization.

The remainder of this paper is organized as follows: Section II begins with a review of relevant background and related work as well as the analysis of the workload on our Blue Gene/L that led to this investigation. Section III analyzes the

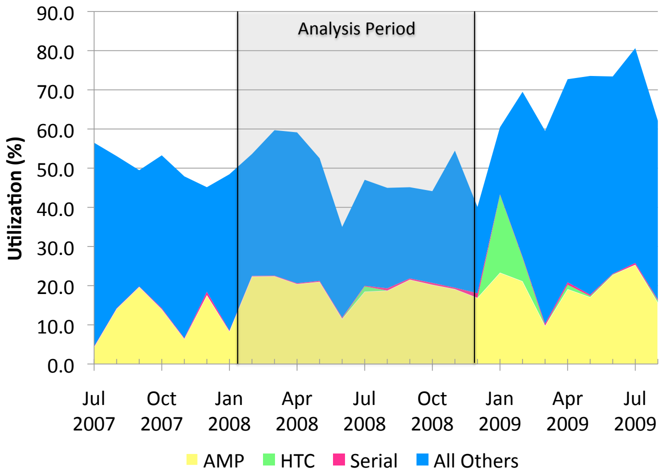


Fig. 1. Two-year utilization for the NCAR Blue Gene/L emphasizing loosely-coupled (AMP), many-task (HTC), and serial jobs. Detailed data transfer information was collected for a 324 day analysis period.

charging model used by Amazon EC2 and develps a comparable charging model for our system; section IV describes the implementation of the model via scheduler and system instrumentation. Section V evaluates the cost efficiency of the Blue Gene/L first ignoring and then including utilization metrics. Section VI then briefly examines our system's competitiveness for the small portion of its workload that could be outsourced as well as the estimated competitiveness of a replacement system scheduled for deployment in the first half of 2010.

II. Background

The designation "cloud computing" has been applied to a wide array of technologies at various levels of abstraction ranging from remotely-hosted on-demand virtual machines to high-level services from which developers can construct highly-scalable software [1] [4]. While many researchers are expressing interest in the application of cloud-enabled programming methodologies such as MapReduce to a variety of problems, using volunteer-computing paradigms for independent tasks [5], or using clouds for specific workflows, our interest is on the on-demand provisioning of computational resources capable of running customized virtual machines. The primary selling point for outsourced computational resources is the economy of scale not available to smaller organizations. As succinctly put by Werner Vogels, the CTO of Amazon, "If you're not in the business of running large data centers, leave it to someone who is [6]."

From the perspective of an investigator choosing between capital expenditures and outsourcing, this is highly relevant advice. However, supercomputing centers *are in the business of running large data centers*. The scientific computing resource marketplace may thus become competitive. Where investigators would previously seek funding for their own resources or request computational time on shared cyberinfrastructure, it is now possible to budget for computational campaigns in real dollars. Traditional supercomputer centers retain their monopoly on capability computing for tightly-coupled massively parallel jobs requiring specialized interconnects, but for jobs requiring loosely-coupled resources, cloud resource providers are a compelling alternative.

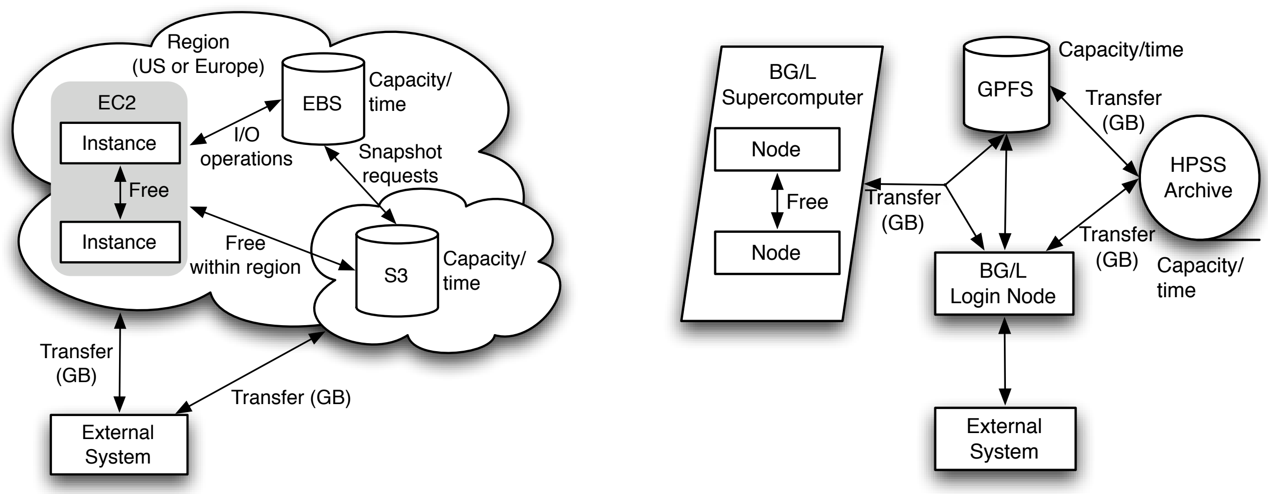
Recent work in the area of dynamic clusters also simplifies the outsourcing of computational capacity. For example, Globus Virtual Workspaces supports the dynamic creation of a virtual cluster with a user-defined software stack [7]. These virtual clusters can then be used to run MPI jobs that do not require tightly-coupled communications. More recent work encourages "elastic clusters" that first consist of privately owned resources but expand to cloud resources during periods of high demand [8].

III. Design

The IBM Blue Gene/L was designed to support highly-scalable codes with tightly-coupled parallelism. We have run our single-rack Blue Gene/L (Frost) for about four years serving a community of NCAR, university, and, more recently, TeraGrid users. Since 2007, the system's utilization has increased from about 50% to about 80% (see Figure 1), and in October of 2009, the system was expanded to four racks. However, an analysis of the workload on the system suggests that about 20% of the system's workload would be easily offloadable to, and perhaps more appropriately run on, outsourced cloud-computing resources (as briefly described later).

Supercomputing centers have a long history of charging for computational time using units of arbitrary value. The units of choice vary across organizations and systems and may be directly derived from a current system or normalized to a historically significant system, but are fundamentally merely a scaled expression of the amount of time utilized on a processing resource (e.g., CPU hours). For example, scientists writing proposals to obtain computational time on NSF TeraGrid systems request allocations on specific machines using machine-specific service units (SUs) that are generally CPU hours on the system being requested. When allocation requests expand to multiple machines, the allocation may be expressed with multiple line items for each system's SU request, or converted using charging factors based on benchmarks into normalized "TeraGrid Cluster" SUs. Occasionally, user behavior is subtly steered using the charging policy; for example, on some systems, a user can "purchase" queue priority at a higher per-hour charging rate.

Cloud computing resource providers are different from government and academic supercomputing centers in that they actually charge *money* in exchange for resource consumption. In addition to the consumption of time on processors, additional activities such as data transfer, I/O operations, and persistent storage are metered and billed, and customers can often pay up front or in bulk to obtain discounted rates of service. To address the economics of cloud computing from our perspective as a supercomputing resource provider, we develop a charging model comparable with commercial



(a) Amazon EC2

(b) NCAR Blue Gene/L

Fig. 2. Metered and monitored activities for computational jobs running on Amazon EC2 and the NCAR Blue Gene/L. Both platforms monitor computational time and storage consumption over time, but vary with metering treatment of network data transfers and I/O.

systems by including multiple metered components in addition to computational time, normalize the metric (e.g., to dollars) based on the actual cost involved with purchasing and operating systems, and develop appropriate metering mechanisms.

A. Analyzing the Amazon EC2 Charging Model

Amazon's cloud computing services consist of several distinct components that are charged independently. The EC2 service provides instances of virtual machines with user-specified images, and its associated Elastic Block Storage (EBS) service provides block-based storage usable by virtual machine images running within EC2 [9]. A separate but related Simple Storage Service (S3) provides additional storage capacity using proprietary transactional interfaces and can be used directly by developers or serve as a store for backups of EC2 images [10].

Computational resources are charged by time, storage by capacity over time, and network transfers by data volume (see Figure 2a). Computational time is charged in integer hours always rounded up. Data transfers into and out of the EC2 cloud computing environment are more expensive than data transfers within the environment. Moreover, Amazon associates its EC2 and S3 resources with geographic regions, such as the United States or Europe, and activities crossing geographic boundaries cost more than those that remain within a geographic region. This serves as motivation to develop workflows that keep data and computation proximate. Amazon also hosts many large publicly available data sets within their environment to encourage the use of EC2 for academic research. Some data transfers, such as from EC2 to S3, are free within regions. From the perspective of implementing HPC workflows on EC2, charges for computational time and storage capacity over time are straightforward and generally easily estimated by users with experience on similar computational platforms. However, EC2's charge for I/O operations to EBS is not readily estimatable for many users of scientific codes, as traditional supercomputer centers monitor capacity, throughput data is readily calculated, and I/O operation counts are almost never expressly benchmarked.

In terms of computational resources, Amazon describes its EC2 systems in terms of a normalized "EC2CU" unit that is advertised as roughly the capacity of a 2007-era 1.0-1.2 GHz Opteron or Xeon processor. EC2 clients may choose from a buffet of resource choices named as if they were pizzas (medium, extra large, quadruple extra-large, etc.), with variants designed to satisfy jobs by emphasising high amounts of memory per core or large processor counts, but the instances present similar memory per core at an identical cost of \$0.085/CPUh.

B. Designing a Charging Model for the NCAR BG/L

The NCAR Blue Gene/L, like many supercomputers run by research organizations, has always been allocated and accounted using computational time as the sole metric. However, because the unit of allocation on a Blue Gene/L system is a 64-core partition including at least one I/O node dedicated to the job, it is possible to identify the amount of data traffic entering and leaving the Blue Gene/L computational system on behalf of every job and consequently every user. The same constraint would also be satisfied for any cluster-like system that allocates entire nodes to a single job (and user) instead of allowing multiple small jobs from multiple users to be packed on a single node. The virtualization systems providing cloud resources are similarly able to produce per-user activity metering.

The resulting charging model includes computational time, storage capacity as capacity over time, and data transfer by data volume (see Figure 2b). Computational time is charged for actual time used including the system's set-up and tear-down overhead (about 1-4 minutes per job) for any portion of each 64-node partition used. Data transfer to and from the supercomputer itself is metered regardless of whether the data transfer is to high-performance storage or a support system or a login or control node. Transfers to and from external (e.g., Internet) hosts and the front-end support hosts cannot easily be metered because the front-end hosts are typical multi-user Linux servers.

The system's high-performance storage restricts total utilization using quotas and is monitored by recording the amount of storage in use by each user on a daily basis. Finally, data access to and from external archival storage can be recorded (but is not at the present time), and per-user data storage is also reported on a periodic basis. In this paper, we examine computational utilization, data transfers, and data storage, using naive and utilization-aware methods to produce possible charging examples and revenue estimates.

IV. Method

As a TeraGrid resource, the NCAR Frost Blue Gene/L supercomputer has always been configured to collect computational resource utilization data in the form of CPU hours reported as site-specific service units. In early 2008, we began to more thoroughly investigate the behavior and resource requirements of jobs running on the system. This investigation was motivated by a desire to establish the I/O requirements of jobs running on the system prior to the installation of a new storage subsystem coupled with the desire to size any new storage to support increasingly available 10Gbps wide-area network connectivity.

Instrumenting Frost's scheduler to collect the additional data was straightforward. Frost uses the Cobalt scheduler from Argonne National Laboratory [11] that features a robust job state transition intercept and callback feature. This feature allows custom scripts to be run at several job state transitions with access to a selection of job metadata and control variables. Most notably, custom scripts can be run before a job is dispatched to the system and after a job terminates. We orginally used this feature to record job accounting data directly to the central accounting database as soon as a job terminates to provide immediate accounting feedback. For this investigation, we modified the system accounting process to also include computational, network, and storage utilization.

As usual, computational utilization is calculated as the amount of CPU hours used by a job. The network utilization was calculated by polling the Force10 E1200 network switch using SNMP before and after every job. With this mechanism, true job-specific network utilization and network utilization data is available for all jobs. We also added code to use the GPFS file system's quota reporting command to determine storage utilization before and after each job. However, this data does not accurately describe the storage used by a job (on a job-specific basis), as multiple jobs running concurrently interfere with each other in the statistics by writing to shared storage. Thus, storage charging is based on daily usage snapshots instead. The extended data collection mechanism is in addition to the operational accounting system used on Frost. Care was taken to ensure that the operational accounting recording procedure was executed before attempting to collect the additional metrics. Any failures in the experimental data collection software do not affect NCAR's TeraGrid accounting process or the system's published utilization metrics.

TABLE I
AMORTIZED 5-YEAR COSTS FOR OPERATING A SINGLE-RACK IBM BLUE GENE/L SUPERCOMPUTER.

IBM Blue Gene/L rack	$1,200,000
IBM support contract	450,000
Front-end nodes	500,000
Force10 network switch	250,000
Hardware cost over 5 years	480,000
Annual staffing	140,000
Annual power and cooling	21,000
Total cost per year	$641,000

The extended system and network utilization data collection mechanism operated for 324 days (from 7 Jan 2008 to 26 Nov 2008) and recorded complete detailed accounting information for 144977 out of 146189 jobs (99.17%) constituting 7886819 out of 7946881 consumed CPU hours (99.24%). Thus, 0.83% of the jobs run on the system did not have detailed accounting information recorded. We believe that the most common cause of omission was the failure of the module that polled file system quota data, as we later discovered that occasionally the process would block without completing. These missing jobs represent 0.76% of the CPU hours consumed during the analysis period. With loss rates of less than 1.0%, we consider the data sufficient for charging trend analysis. This underscores the importance of the metering mechanism – if the usage wasn't recorded, the user can't be charged. In the 324-day analysis period, 7,886,819 CPUh were consumed and 365,275 GB of data were transferred to and from the Blue Gene/L system.

V. Results

A. Amortized Computation-Only Charging

The most straightforward method of describing the ownership and operating cost of a supercomputing system is to calculate the per-year cost including the initial hardware and support purchase plus continuing operating expenses (see Table I). In the case of Frost, the initial equipment averages to $480,000/year for 5 years (although the equipment was covered under a support contract only for only the initial four years), with an additional $161,000/year for a partial FTE system administrator, part-time student, and annual power and cooling (calculated at 25 KW, $0.066/KWh, and PUE 1.4).

The most simple model examines computational time only. With 17,940,480 CPUh available on Frost in a single year, and a total amortized cost of $641,000 per year for ownership and operation, each *available* CPU hour would have to be billed at $0.0357 to break even. Unused hours and downtime are lost forever.

B. *Utilization-Aware Computation and Data Transfers*

1) Computation: While the amortized charging model provides a useful pricing baseline, it assumes that the underlying system is fully utilized. In the case of Frost, the system generally increased in utilization from 50% to 80% over the two year period of July 2007 to July 2009 (refer back to Figure 1), shortly after which three additional Blue Gene/L racks were procured to satisfy the demand. During the analysis period, the system's utilization was 49.5%, so with perfect hindsight the naive computation-only price would have to be multiplied by just over two to break even at $0.0721/CPUh.

2) Data transfers: The charging model for the Blue Gene/L also includes data transfers to and from the supercomputing system, so both computational time and data transfers for every job run on the system can contribute to total revenue (see Figure 3). Linear combinations of charging for CPU hours and GB transfered can produce total revenue capable of satisfying arbitrary utilization metrics. However, it is clear from the analysis period that charging for data transfers to and from the Blue Gene/L system is quite useless. Although over 365 TB of data were moved to and from the system, charging "reasonable" values produces negligible revenue. For example, matching EC2's lowest price point at $0.10/GB only produces $36,527 – not even within the correct order of magnitude to contribute to the system's amortized hardware and annual operational costs. Even three times EC2's price point produces little revenue compared to the charge for computational time, so we conclude that for supercomputing operations, data *transfers* are not a useful metering metric.

The lack of revenue from data transfer charging is likely due to the different nature of data flows in supercomputing applications and cloud computing applications. For example, data moves to and from a supercomputer only while a job is running, so data transfers are limited by the number of jobs and their duration. Amazon's cloud computing resources, on the other hand, are often used to provide web-based services to a large public audience, which may result in data transfers from a hosted instance to a very large community of independent users.

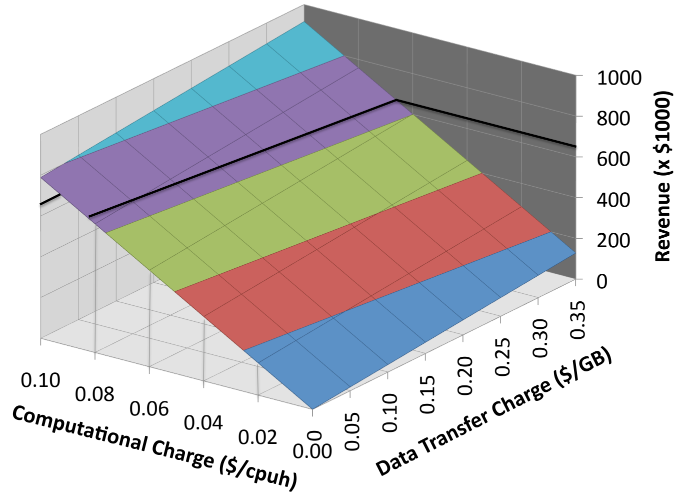


Fig. 3. Estimated possible revenue for Frost during the analysis period by charge for computational time (CPU hours) and data transfers (GB). The level mark indicates break-even cost including system utilization.

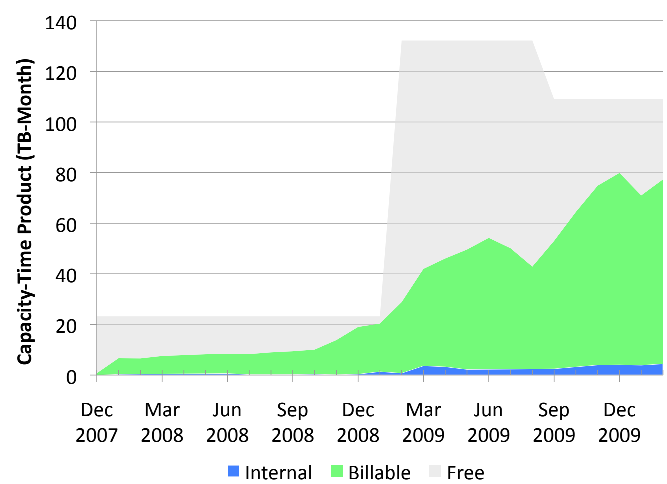


Fig. 4. Two-year data storage utilization for the NCAR Blue Gene/L system in TB-months (using powers of two, e.g., TiB-months) measured daily and calculated as average monthly usage.

C. *Data Storage*

The amount of data stored on Frost's disk systems has increased from roughly 6TB to over 80TB during the two-year analysis period of 2007 and 2009 (see Figure 4). In January 2009, Frost's 19TB storage system was expanded with oa new 109TB storage cluster just as the existing system was nearing capacity; after a vetting period, the original storage was decomissioned in favor of the newer hardware. During a two-year period, 699139 TB-months of data were stored.

For a nominal analysis of potential revenue, if storage during these two years was billed at the same rate as EC2's current pricing ($0.10/GB-month), only about $69,914 of revenue would have been generated compared to the ~$250K required to purchase the expanded system (omitting operational costs, but moderated by the system's use for special projects for over a year before being released to users). However, it is important to note that the price of the storage procured in 2005 and used through 2008 was substantially more expensive the price of the 109TB storage system procured in 2007, and current storage prices are substantially less than that system.

This highlights the intuitive importance of adjusting the charge for data storage based on the purchase price of the storage itself. For Frost, it would thus be more useful to analyze charging in epochs based on storage upgrades, starting with the most expensive storage in 2007 and establishing new lower prices as time progresses based on anticipated utilization over a disk pool's lifetime. This is difficult for Frost, as some storage was obtained as part of the original system purchase, a total of three storage systems were rotated through operation during the computational system's lifetime, and user data has been rotated through these systems as part of their lifecycle as well.

However, it is important to note that for the case of charging for supercomputing resources using cloud-based model, storage charging (unlike computational time charging) can be just as flexible for supercomputing resource providers as for commodity cloud resource providers. For the supercomputing resources, the purchase of a supercomputer is often a single large investment constrained by the scalability of the interconnect; it must be purchased at once and is difficult to expand in a scalable fashion over time. Storage, on the other hand, can be and often is purchased in increments, even by supercomputing resource providers. Thus for storage, supercomputing providers have the same ability to incrementally purchase resources and lower prices over time.

VI. Comparison with Cloud Computing Providers

Our analysis of the workload on Frost during the observation period indicates that about 20% of the system was consumed by a workflow servicing a single science gateway that is a viable candidate for execution on cloud computing resources. Having developed naive and retrospective charging models for the Blue Gene/L, we can apply the models to estimate the cost effectiveness of Frost and EC2 for applications that can be efficiently executed on either platform. While this is not intended to serve as comprehensive analysis of the viability of cloud computing for general scientific workloads (a topic that has been examined extensively in the literature), this provides an opportunity to predict how a typical principal investigator (PI) might react when presented with a bill for Frost and whether or not that PI would choose to remain on the system or to seek a cloud computing platform as an alternative.

The application consuming roughly 20% of Frost's computational time runs a set of coordinated independent tasks performing evaluation functions for a genetic algorithm. Thus, it maintains a very small quantity of data on each machine (on the order of kilobytes), produces about 45 MB of output for the final results, and does not require a high-performance interconnect. The workflow is already capable of running on virtual machines and its data operations would generate minimal ongoing cost. On the modern processors found on the TeraGrid, such as the AMD Barcelona chips in Kraken (before its recent upgrade to 6-core Istanbul chips) and Ranger, which are roughly similar to those available in typical EC2 instances [2], the code experiences a speedup between 4.74 and 5.22. Thus, a typical run that takes 24 hours on Frost and consumes 3,072 CPUh would complete in 4-6 hours on a modern cloud infrastructure platform. On EC2, the outsourced computation would cost between \$43 and \$65, compared to \$221 based on our *a posteriori* Blue Gene/L model. Even though this paper-only estimate neglects EC2 data charges, it is unlikely that the task would produce more than \$150 in data charges for the $\sim$50 MB of data produced and stored.

Even ignoring data transfers and storage in this estimate, the phenomenal speedup of the transition from the power-efficient processors in Frost to a current architecture provides a substantial decrease in execution time. If Frost usage was billed using money, the PI of this particular application would almost certainly make the transition. However, as Frost is allocated using a typical computing award process, its low cost (free to the PI) will likely remain a compelling price point, so the comparison between Frost's *replacement* system and current cloud-computing resource providers, presented in the next section, is more relevant.

TABLE II
Anticipated amortized annual costs for operating a 2,736-socket 184 TF Westmere-based commodity cluster.

System and Container (\$8M / 5 years)	\$1,600,000
Annual staffing	600,000
Annual power and cooling	365,000
Total cost per year	\$2,565,000

VII. Future Cluster Cost Estimation

We are currently in the process of architecting a new supercomputer system to eventually replace Frost that features 2736 sockets (16416 to 32832 cores depending on final processor selection and 6, 8, or 12 cores/socket) connected using QDR Infiniband, and including 960 TB of storage. Combined with operational staffing and power expenses, the system is estimated to cost \$2,565,000 per year (see Table II). Depending on the number of cores in sekected, the per-CPUh break-even charge (assuming 100% utilization, and including the cost of the attached storage) will be 0.89 cents/CPUh to 1.7 cents/CPUh.

While the cost analysis for Frost suggested that moving non-parallel jobs to commercial cloud computing resources would be feasible, the rough pricing for Frost's replacement suggests that we may soon yet again be able to offer cycles to our community more cheaply than they may be obtained commercially. However, it is important to note that these estimates are made at a single point in time, and that Amazon has the ability to alter their pricing model much more frequently than our four-year acquisition cycle.

The prior analysis includes the purchase of 960 TB of storage with the system's acquision. About \$750K/\$8M of the purchase price is attributable to storage, which normalizes to \$150,000/year. If the computational and storage costs are calculated separately, the break-even full-utilization charge drops from 0.89 cents/CPUh to 0.63 cents/CPUh, further increasing the computational price advantage. The \$750K thus corresponds to a storage acquisition price of \$0.76/GB. Amortized over five years, and assuming an even ramp-up to capacity over that time, we would have to bill 2.543 cents/GB-month to recover costs. This is substantially less than current cloud storage hosting rates even with the additional expense of providing an aggregate throughput to disk exceeding 16 GB/s from 1368 nodes.

VIII. Discussion

The implementation of a computational, network transfer, and data storage charging system for Frost demonstrated that for Frost, when subjected to the constraint that the unit charge be roughly equivalent to current EC2 rates, none of these methods produced sufficient revenue to cover the costs of the system and its operation. While computational time charging is sufficiently close to cover expenses given *a posteriori* utilization, it is striking that data transfer charges produce negligible revenue especially given the intricacies of collecting the data required to charge for data transfers. For data storage charging, the pricing model for storage at present (e.g., using 2009-2010 storage prices and cloud storage rates) cannot be reconciled with the three-year-old purchase cost.

The charging and cost analysis for Frost's outsourcable jobs demonstrated that EC2 could be more cost-efficient than Frost for a portion of its workload. This is not surprising as the comparison is between a resource purchased in 2005 to support a specific type of low-power HPC capability computing and current processor technologies. Of course the current technology has better performance, and so it is possible for an amortized owned resource to no longer appear cost competitive with on-demand cloud resources that are billed hourly. A similar result is obtained when comparing Frost to a new commodity cluster.

The continuing management decision with Frost is thus when to simply unplug it. At that point, its users move jobs to other systems based on their application characteristics and capabilities. However, until we decide to unplug Frost, its continuing operation is essentially *free* to us and the users, as the organization pays for the power, the staffing is already in place, and the system was officially depreciated by the finance department years ago. It is unlikely that a short-term solution will be to remove the resource and contribute the now-unemployed system administrator's salary and overhead to a bulk Amazon EC2 purchase. Firing the administrator now would provide only 1,647,058 CPUh on EC2; not having purchased Frost (and waiting five years for the cloud computing technology to develop) would have provided 6,482,352.

As clusters age over time and new computers can provide more cycles at less cost, it becomes increasingly cost-effective to target offloadable workloads to outsourced resources. One possibility is to use discretionary control of capital equipment funds to supplement the waning years of supercomputers with outsourced resources until the next purchase makes in-house hosting again advantageous. This approach is quite limited, however, in that it relies on the portability of a particular scientific application from a supercomputer to a cloud, which has been shown to be limited in the literature [12], [2]. In order for this to be possible, researchers with appropriate workloads would need to be encouraged to deploy their software using mechanisms making the use of outsourced resources possible.

In addition to outsourcing workloads, it is also possible for HPC resource providers to support cloud-based execution. Indeed, to encourage developers to embrace the cloud-computing technologies that can provide the flexibility to make offloading possible, it is essential that HPC resource providers support these mechanisms. From the user's perspective, the IaaS cloud model emphasises the availability of capacity on demand. From the resource provider perspective, however, obtaining a high system utilization is economically feasible, but too much leaves some users unsatisfied or leads to queuing.

In order to effectively support both traditional batch queued HPC and cloud-based HPC as a resource provider, a bit of cultural adjustment may be necessary. Traditional HPC environments thrive by maximizing utilization while trying to keep job wait time minimal but often nonzero. To support cloud-based execution, at least some portion of resources must always be available to immediately fulfill requests of "reasonable" size. This problem can be satisfied using a combination of an existing allocation process and system utilization management. For example, if cloud-based projects are subject to an allocation process, ranges of virtual machine availability can be described by the project and perhaps adjusted over time. Thus, while infinite resources would not be immediately available, finite resources could always be immediately available through a service-level agreement.

IX. Future Work and Conclusions

In this paper, we examined the application of a billing model similar to that used for cloud computing resources including computational time, data transfers, and storage charging for our existing supercomputer. In order to generate break-even revenue on the Blue Gene/L, we would need to charge between 3 and 5 cents per CPU hour, a rate not competitive with the performance of current commodity processors available on modern cloud computing resources for applications not requiring a high-performance interconnect. Surprisingly, the analysis demonstrated that for the applications run on our Blue Gene/L, charging for data transfers would produce negligible revenue. The system planned for acquisition in mid-2010 will be substantially more cost-competitive than outsourced EC2 computational and storage resources.

Although our supercomputer systems will almost certainly remain allocated by committee instead of billed directly, we intend to support the cloud computing IaaS approach on our future systems. Even if these systems system cannot provide *unlimited* resources with no delay, it will most certainly be able to provide limited resources to a limited user community with no delay. By adding cloud support, HPC projects can continue to run using familiar interactive and Grid interfaces, and projects that could be run on commercial clouds can be migrated to virtual machine platforms by their developers over time and at their convenience.

Acknowledgments

Thanks to Theron Voran and Narayan Desai for their assistance with the Cobalt scheduler used on Frost. This analysis described the use of computer time on a system supported by NSF MRI Grants #CNS-0421498, #CNS-0420873, and #CNS-0420985; NSF sponsorship of the National Center for Atmospheric Research; the University of Colorado; and a grant from the IBM Shared University Research (SUR) program.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, Feb 2009. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

[2] C. Evangelinos, P. Lermusiaux, J. Xu, P. Haley, and C. Hill, "Many task computing for multidisciplinary ocean sciences: Real-time uncertainty prediction and data assimilation," in *Proceedings of the Second ACM Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS)*, November 2009.

[3] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The cost of doing science on the cloud: the montage example," in *SC '08: Proceedings of the 2008 ACM/IEEE conference on Supercomputing*. Piscataway, NJ, USA: IEEE Press, November 2008, pp. 1–12.

[4] L. Youseff, M. Butrico, and D. Da Silva, "Toward a unified ontology of cloud computing," in *Proceedings of the Grid Computing Environments Workshop, 2008. (GCE08)*, November 2008.

[5] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. P. Anderson, "Cost-benefit analysis of cloud computing versus desktop grids," in *IPDPS '09: Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1–12.

[6] W. Vogels, "Amazon Web Services," *Keynote talk at USENIX Large Installation System Administration conference (LISA09)*, November 2009.

[7] I. Foster, T. Freeman, K. Keahey, D. Scheftner, B. Sotomayer, and X. Zhang, "Virtual clusters for grid communities," in *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid*, May 2006.

[8] Globus Virtual Workspaces. http://workspace.globus.org/.

[9] Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/.

[10] Amazon Simple Storage Service (Amazon S3). http://aws.amazon.com/s3/.

[11] N. Desai, *Cobalt: Component-Based Lightweight Toolkit (HPC job management suite)*, http://trac.mcs.anl.gov/projects/cobalt.

[12] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "A performance analysis of EC2 cloud computing services for scientific computing," in *Cloudcomp 2009: Proceedings of Cloudcomp 2009*, October 2009.