Supporting Cloud Bursting for HPC and Data-Intensive Applications

Gagan Agrawal

Ohio State University

agrawal@cse.ohio-state.edu

Topics: Cloud Architectures and Systems, Programming Models for the Cloud

Current Work on Cloud Computing:

We are a research group with interests in high performance and distributed computing. We have been developing and evaluating mechanisms for HPC applications and scientific workflows in cloud environments.

Specific directions include:

Development of an alternate API for data-intensive computing: Our recent work has shown that an variation in the map-reduce API can eliminate sorting and storage of (key,value) pairs, and improve the performance (our paper in CCGRID 2010). We have developed a system MATE (Map-reduce with an AlternaTE API), that provides a high-level, but distinct API. Particularly, our API includes a programmer-managed reduction object, which results in lower memory requirements at runtime for many data-intensive applications. MATE implements this API on top of the Phoenix system, a multi-core map-reduce implementation from Stanford. We have also shown that our API helps provide more efficient fault-tolerance support. We have demonstrated that with our novel approach results in a better performance both in presence and absense of failures (our IPDPS 2010 paper).

Support for application-specific elasticity: We have developed an application adaptation middleware (our papers in SC 2010 and HPDC 2010) that can tune an application to achieve the desired trade-off between QoS, throughput, and computing costs. We have also developed a VM-based consolidation module, which can analyze resource needs of tasks in a scientific workflows, and consolidate tasks to reduce resource and energy costs, while maintaining performance. We are now extending this support for the Amazon EC2 platform. We are also considering MPI applications, and creating a model for their performance.

Relevant Publications:

- David Chiu, Apeksha Shetty, and Gagan Agrawal, “Elastic Cloud Caches for Accelerating Service-Oriented Computations”, in proceedings of SC, November 2010.
- Qian Zhu, Jiedan Zhu, and Gagan Agrawal, “Power-aware Consolidation of Scientific Workflows in Virtualized Environments”, proceedings of SC, November 2010.
- Qian Zhu and Gagan Agrawal, “A Resource Allocation Approach for Supporting Time -Critical Applications in Grid Environments”, in proceedings of SC 2009, Nov. 2009.
- Qian Zhu and Gagan Agrawal, “Resource Provisioning with Budget Constraints for Adaptive Applications in Cloud Environments”, in proceedings of Conference on High Performance Distributed Computing (HPDC), June 2010.
- Tekin Bicer, Wei Jiang, and Gagan Agrawal, “Supporting Fault Tolerance in a Data-Intensive Computing Middleware”, to appear in proceedings of International Parallel and Distributed Processing Symposium (IPDPS), April 2010.

Proposal: Cloud Bursting for HPC and Data-Intensive Computing

For many organizations, one attractive use of cloud resources can be through what is referred to as *cloud bursting* or the *hybrid cloud*. These refer to scenarios where an organization acquires and manages in-house resources to meet its base need, but can use additional resources from a cloud provider to maintain an acceptable response time during workload peaks. Cloud bursting can be an attractive model for organizations with a significant need for high performance and/or data-intensive computing. Despite the interest in high performance computing on the clouds, such organizations can be expected to continue to invest in in-house HPC resources, with considerations like providing the best performance, "security" needs of certain applications, and/or desire for having more control over the resources. At the same time, through cloud bursting, they can avoid over-provisioning of resources, while still providing users better response time.

Significant work is needed to enable use of cloud for various classes of applications, including cloud bursting support. **Our goal is to develop mechanisms and software frameworks as well as address resource allocation problems associated with the execution of MPI programs, GPU-based HPC applications, and data-intensive applications on cloud environments, particularly targeting the cloud bursting use scenario.**

The specific research contributions will be as follows:

MPI Applications:

Software Framework: We will be developing a novel compiler/runtime infrastructure for making existing MPI applications *dynamically moldable* in order exploit *elasticity* of cloud resources. A dynamically moldable application is capable of changing the number of nodes for execution during its execution, by redistributing its data to a different number of nodes. Our runtime framework will also automatically decide when the number of nodes on which the application is executed should be increased or decreased, in view of constraints on cost and/or desired time of completion.

Resource Allocation Problems: Consider a supercomputing center that will be offloading certain jobs to a cloud resource provider. Interesting new scheduling problems arise with the involvement of a cloud provider. For example, one goal could be to address the bi-objective problem of maximizing throughput and minimizing the total cost incurred on the cloud. Another formulation could involve minimizing the total cost incurred on the cloud while meeting all jobs' completion deadlines. We will considering these problems under different assumptions on applications - applications that can only be executed on a fixed (pre-specified) number of nodes, applications that could be *statically moldable*, or could even be *dynamically moldable* and/or *transportable*, i.e. they can be moved from one environment to another during their execution.

GPU Based HPC Applications:

HPC applications are increasingly using GPUs for accelerating computations, and cloud providers are now beginning to offer GPUs. We consider environments where multiple independently developed applications are executing on virtual machines (VMs), and each of them could invoke kernel functions on a GPU. As the number of applications (or VMs) could be larger than the number of GPUs available, we need mechanisms to enable mapping of kernels to actual GPUs, and sharing of GPUs, including possibly consolidating multiple kernels calls on a single GPU.

Software Framework: We will be extending gVirtiS, a recent open source software that allows GPU kernels to be invoked from within a VM, to make mapping and consolidation transparent to the applications. The resulting framework will allow a GPU to be a true shared resource in virtualized environments, in the same fashion as existing VM software allows memory, network bandwidth, and disk bandwidth to be shared. The framework will also allow nuanced resource allocation solutions to be implemented.

Resource Allocation Problems: Sharing of GPU (including multiple kernels sharing the same GPU) creates new resource allocation problems. We will considering a number of different formulations, based on the amount of information available about the applications' use of resources, and different resource allocation goals (maximizing throughput, meeting deadlines, or others).

Data-Intensive Applications:

We consider scenarios where a dataset that is stored partially in a local cluster and partially on a cloud resource needs to be analyzed. Such scenarios could arise because instruments or simulations may generate more data then there is available space on the local resource. In such cases, we need to divide the data processing task across multiple locations, and perform analysis. Moreover, while computing resources for processing the data may be available both locally and on the cloud, they may not be available in the same proportion as the fraction of data stored at each end. This leads to many interesting issues related to software framework design and resource allocation.

Software Framework: Our proposed framework can be viewed as an implementation of map-reduce that can analyze data stored across many geographically distributed resources. Our development will be based on our work with FREERIDE-G, which we will be extending to consider clusters or set of instances that are part of cloud, and data hosted on environments like Amazon S3.

Resource Allocation Problem: Suppose part of the data is resident on a local resource, and another part of the data is resident on a cloud resource. We want to apply reduction like analysis on the aggregate data. We need to decide how computing nodes should be allocated on the local cluster and cloud resource providers.