Research Statement

Sang-Min Park

Across industry and academia, people are witnessing the accelerating transition from small-scale, closed, in-house computing and data storage to mega-scale, open, and service-oriented infrastructure, called **Cloud computing**. In the last few years, as a graduate student at the University of Virginia, I am conducting research that can make a significant impact to this emerging field. Specifically, I have been addressing resource scheduling problems to deliver more predictable computation to HPC applications while resource utilization on the shared infrastructure is sustained at a high level, thereby satisfying both end users and infrastructure operators. The outcome of my research has been promising; my thesis shows that the application of rigorous scheduling theory (i.e., control theory) to virtualized resource containing mechanism (i.e., VM scheduler) can deliver predictable computation to important applications, including eScience and MapReduce programs.

Since 2002, my personal pursuit has been in the area of high performance, distributed systems. My approach to such research problems is to design a potential solution and then to prototype a working system. While pursuing a PhD, my passion in research and software development has incorporated sound principles and rigorous tools. Finding an important research problem, understanding the problem to the deepest extent, and personalizing the problem by emotionally committing to it have become my guiding principles. The application of sound tools (e.g., control theory), which is not usually used in my area, has been highly effective in addressing the issues that are increasingly important regarding Cloud computing. Below, I address why scheduling problems are important in large-scale, shared infrastructures; present my control-theoretic, virtualized approach; and sketch out my future research directions.

1. Background

Today, we continue to hear the two interrelated arguments:

*"**Utilization** is the factor that many in the industry hate talking about because the industry-wide story is so poor. The McKinsey report says that enterprise server utilization is actually down around 10%…."* (James Hamilton, from his blog article)

*"Obstacle number 5-**Performance unpredictability**: the obstacle concerns the scheduling for virtual machines for some classes of batch processing programs, specifically for high performance computing…." (Above the Clouds: A Berkeley View of Cloud Computing)*

Ironically, if the infrastructure has idling capacity, they could be provisioned to an application that is currently suffering from low performance. ***Unpredictably low performance on underutilized infrastructure***, in fact, is not a new problem with Cloud computing. National supercomputing centers and campus-wide clusters have experienced the similar problem in which some jobs are sitting in a queue unpredictably long because of the idling jobs that are exclusively reserved or allocated. At many institutions, it is not the lack of capacity that causes poor performance, but the lack of intelligent scheduling that would prioritize computing requests according to user-guided performance goals.

A representative type of applications that can greatly benefit from predictable infrastructure is the E-Science application. Today, many scientific applications require adaptive processing of environmental sensor data in real-time, so that discoveries/predictions such as tornado warnings (LEAD project) can be formulated in time. There are stimulating use cases in domains as diverse as weather forecasting, coastal hazard prediction, and patient-specific medical modeling. However in today's supercomputers and clusters, the batch queue system may introduce unpredictably long wait times regardless of job's urgency. The Cloud still does not offer a reasonable performance guarantee. Lack of predictable and adaptive computing infrastructure has been the major barrier to the full realization of adaptive E-Science.

2. Dissertation Research

The goal of my dissertation research is to create a resource scheduling framework that delivers predictable computation to compute/data-intensive applications while sustaining the infrastructure's utilization at the highest level possible. There are two key components in my own direction:

1) ***Virtualized performance container***: my work leverages O/S-level virtualization technologies to isolate performance among concurrent applications, and dynamically resize the VM's capacity to meet the user's performance goals. Therefore, unlike the previous approaches that exclusively schedule an application to an entire node, I essentially create a time-sharing abstraction on which multiple applications share underlying system resources in more predictable way.
2) ***Application and extension of control theory***: my framework allows users to express an application's performance requirements in terms of its runtime progress (sometimes the progress requirements can be automatically drawn from a higher level description about performance goals). The critical task is to multiplex underlying system resources to co-existing performance containers such that all applications running inside each container meet their progress requirements. I leverage and extend the classic control theory widely used in engineering and mathematics (e.g., cruise control system). The control theory is especially powerful as it offers a provable guarantee on the accuracy and stability of scheduling decision.

In the early phase, I argued that the combination of a control-theoretic scheduler with the virtualized performance container can create a **predictable time-sharing abstraction** that can unleash emerging adaptive E-Science applications from prevailing batch-queue mode. I believe that the argument has been well received by the community, as my paper presenting the idea was **nominated for the best student paper at *Supercomputing 2008,* and subsequently an NSF award has been granted** to my advisor to pursue it further. Followed by these promising results, other interesting questions were formulated to make the early idea into a practical system. **As a result, 5 papers were published in competitive venues in the last two years**.

3. Future Direction

I would like to continue researching how to deliver predictable computation to scientific users and more efficiently operated Cloud infrastructures. I am eager to apply the control theoretic approaches to scheduling mega-scale datacenter resources. Being in academia with limited-scale resources, I have not explored the intricate issues that are unique in truly large scale. Specifically, I am interested in studying the interoperability of reliability-targeted management systems with performance-oriented schedulers like the one I presented. Furthermore, as loosely-coupled data parallelism such as MapReduce and Hadoop gain significant momentum, I would like to extend my control-theoretic scheduling to address unpredictable performance problems in the emerging areas (early results are to be published in ICAC'10).

My broad future research interests can be summarized as follows:

1) Job and data scheduling in distributed systems, clusters, and Cloud computing.
2) MapReduce/Hadoop programming system
3) Infrastructure for data-intensive applications
4) Control theory applications in software systems

In summary, as computational infrastructure grows at an unforeseen rate, there is a growing demand for predictable performance as well as efficient operation. I offered control theoretic scheduling over virtualized containers as a good candidate for solving the problem. I believe there must be a good science of infrastructure performance that would further guide continuous engineering efforts. I am looking forward to applying my research principles and experiences in the management of large-scale infrastructure.