Calton Pu
School of Computer Science, Georgia Institute of Technology
<calton.pu@cc.gatech.edu>

Topics of interest:
8. Cloud Self-Monitoring and Autonomic Control
11. Cloud Test-Beds

Summary of current research activities:

In the Elba project, we have developed a set of software tools to create and manage large scale experiments to measure the performance of complex benchmarks, with focus on N-tier applications that contain dependencies among their components. Example benchmarks include RUBiS (based on eBay) and RUBBoS (based on Slashdot), which are typically implemented (in a 3-tier configuration) with web servers, application servers, database servers, and a set of client machines that generate the workload. We have run thousands of experiments using a variety of software and hardware configurations on several cloud environments, including the Emulab, Amazon EC2, and Open Cirrus. The analyses of these experimental results have found interesting phenomena previously considered rare. For example, a concrete demonstration of *multi-bottleneck* is due to the dependencies among component servers (see Figure 1 on the next page). Multi-bottlenecks involve simultaneous resource allocation, a situation that only happens with dependencies, which would not arise in workloads with independent jobs running on independent nodes.

Research Plan (if funded by the NSF)

Embarrassingly parallel applications such as web search and data analytics can be decomposed and processed by many nodes in a cluster using software tools such as MapReduce. The scalability of embarrassingly parallel applications is well known. For example, the left graph of Figure 1shows the growth of number of servers allocated to Animoto in a 3-day period, from tens of servers in 4/15 to thousands of servers in 4/18 of 2008. The importance of embarrassingly parallel applications has been recognized by NSF: a simple search for "MapReduce" in award title and abstract yielded 31 awards from all 3 divisions of CISE, OCI and some other programs.

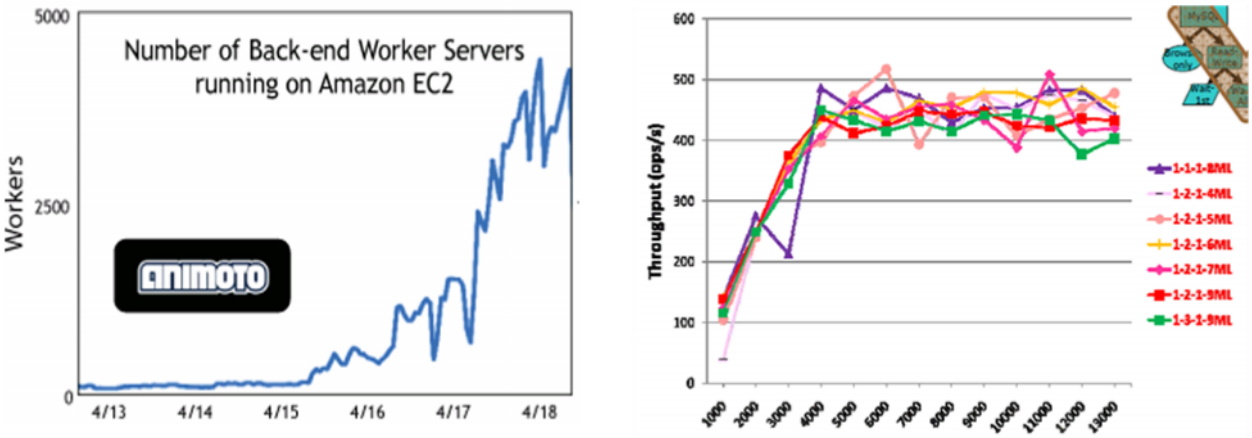

Figure 1 (Left) Unboundedly scalable *Vs.* (Right) Limited throughput despite increasing nodes

In contrast to embarrassingly parallel applications, N-tier applications such as e-commerce and social networks have mutual dependencies among system components that lead to less predictable bottlenecks. Instead of unbounded scalability, we show an example of limited throughput on the right graph of Figure 1, which shows the measured performance (throughput on the Y-axis) does not increase for higher workloads (above 4000 users on the X-axis) in a RUBBoS benchmark (a N-tier application based on Slashdot). The graph shows that the throughput remains the same regardless of the addition of several application servers and database servers, with 7 different configurations yielding approximately the same throughput. Our experiments show that no single hardware resource is a bottleneck, with all hardware average utilization levels significantly below 60%.

In the RUBBoS example (Figure 1), the throughput limit was due to a dependency among the replicated database servers (varying from 4 to 9 replicas). To find this kind of problems, the current industry practice is to run N-tier applications in clouds as a staging area to verify their achievable performance. Unfortunately, changes in configuration often require new verification experiments due to the difficulties in predicting bottlenecks when complex dependencies exist. We plan to conduct systematic large scale experiments to improve our understanding of applications with dependencies. This improved understanding includes finding these performance limitations due to dependencies among system components and a performance model that can predict application performance over a wide range of configurations and workloads.

Unfortunately for researchers interested in applications with dependencies, in contrast to MapReduce, a search for "n-tier" in NSF award title and abstract yielded *zero* awards. One might conclude from this unequivocal statistics that it is hopeless to request NSF funding for research on N-tier applications. To the contrary, we argue that the research focused on application with dependencies (such as N-tier applications that cannot be easily decomposed into independent components) is equally important and complementary to embarrassingly parallel applications. Consequently, it is high time to remedy this lopsided inequality and consider including research on applications with dependencies into NSF portfolio.