Shivnath Babu
Department of Computer Science
Duke University
shivnath@cs.duke.edu

My research interests include:

- Topic 8: Cloud Self-Monitoring and Autonomic Control
- Topic 3. Data Portability, Consistency, and Management

Current Research

Timely and cost-effective processing of large datasets ("big data") has become a critical ingredient for the success of many academic, government, and industrial organizations. The combination of MapReduce frameworks and cloud computing is an attractive proposition for these organizations. However, even to run a single program in a MapReduce framework, a number of tuning parameters have to be set by users or system administrators. Users often run into performance problems because they do not know how to set these parameters, or because they do not even know that these parameters exist. With MapReduce being a relatively new technology, it is not easy to find qualified administrators.

I am working on techniques to automate the setting of tuning parameters for MapReduce programs. The objective is to provide good out-of-the-box performance for ad hoc MapReduce programs run on large datasets. This feature can go a long way towards improving the productivity of users who lack the skills to optimize programs themselves due to lack of familiarity with MapReduce or with the data being processed. The problem is challenging because, unlike declarative SQL queries, MapReduce programs are often expressed in programming languages like Java, C++, Python, and R. Furthermore, MapReduce programs allow data to be read from and written to files in a distributed filesystem, rather than requiring rigid schema definitions like conventional database systems.

Hadoop has emerged as one of the most popular implementations of a MapReduce framework. A number of academic, government, and industrial organizations use Hadoop in production deployments for applications such as Web indexing, data mining, report generation, log file analysis, machine learning, financial analysis, scientific simulation, and bioinformatics research. My longer-term goal in this work is to develop a MapReduce framework for the cloud that builds on Hadoop while adapting to user needs and system workloads to provide good performance automatically, without any need for users to understand and manipulate the many tuning knobs in Hadoop. While the architecture of this system will be guided by work on self-tuning database systems, I expect new analysis practices over big data to pose new challenges that will lead to different design choices.

Proposed Research

Using the Cloud to take the Bite out of System Administration (Or, Tuning and Testing as a Cloud Service)

Despite the emergence of cloud platforms, the task of administering a large system continues to remain notoriously hard. There have been a number of efforts in recent years to simplify system administration. These efforts include system-level mechanisms like virtualization, computational frameworks like MapReduce, and tools that leverage statistical machine-learning techniques to analyze instrumentation data collected from systems. In spite of these efforts, current solutions for administrative tasks like benchmarking, tuning, troubleshooting, and capacity-planning remain far from satisfactory. More often than not, these tasks are done manually by system administrators using their experience, intuition, expert knowledge of system internals or data, tips from tuning manuals, or even guesses.

The availability of pay-as-you-go compute and storage resources on the cloud enables us to rethink how to address several system administration tasks with minimal manual involvement. Specifically, resources allocated on demand on the cloud can be used in order to run *experiments* that collect missing instrumentation data essential to generate the knowledge required to address certain administrative tasks satisfactorily and efficiently. I call this approach *automated experiment-driven management*; an approach made possible by the allocation of a *workbench for experiments* on the cloud.

Consider a database system running on the cloud. Suppose a complex SQL query run as part of a periodic report-generation application in this system is performing poorly. Today, a human user or administrator has to perform a manual SQL-tuning process in order to improve the performance of the repeatedly-run query. Instead, automated experiment-driven management will work as follows. An experiment workbench will be allocated on the cloud using resources similar to those used to run the database system. A current snapshot of the production data needed to run the query will be taken and loaded on to the workbench. Different plans for the query will be tried on the workbench in order to measure their actual performance. Once a query plan that is much better than the earlier (poor) plan is found, the new plan can either be incorporated directly into the production database system or shown as a recommendation to the database administrator. The SQL tuning accomplished, the resources in the workbench can be released; with the whole process incurring only the pay-as-you-go charges for the resources allocated for tuning purposes. In this manner, automated experiment-driven management can remove most of the manual labor involved in system administration.

The availability of a powerful workbench for automated, online experiments enables us to rethink the implementation of several system administration tasks. Cloud computing and associated mechanisms like snapshots and virtualization will provide the foundations for such a workbench.

Cluster Provisioning for Hadoop Workloads in the Cloud

Users can now leverage pay-as-you-go resources on the cloud to meet their data analytics needs. For example, Amazon Elastic MapReduce allows users to instantiate a Hadoop cluster on EC2 nodes, and run workflows. The typical workflow on Elastic MapReduce accesses data initially from S3, does in-cluster analytics, and writes final output back to S3. The cluster can be released when the workflow completes, and the user pays (only) for the resources used. While Elastic MapReduce frees users from setting up and maintaining Hadoop clusters, the burden of cluster provisioning is still on the user. In particular, users have to specify the number and type of EC2 nodes (from among 10+ types) as well as whether to copy data from S3 into the in-cluster HDFS. The space of provisioning choices is further complicated by Amazon Spot Instances which provide a market-based option for leasing EC2 nodes. In addition, the user has to specify the Hadoop-level as well as the MapReduce-job-level configuration parameters for the provisioned cluster.

Can we develop techniques that can automatically determine the best cluster and Hadoop configurations to process a given workload on Amazon Elastic MapReduce (or any other cloud platform) subject to user-specified goals? The user could have multiple preferences and constraints for the workload, which poses a multi-objective optimization problem. For example, the goal may be to minimize the monetary cost incurred to run the workload, subject to a maximum tolerable workload completion time.