Editorial

Recent work in utility and cloud computing

This special issue covers some of the recent advances in utility and cloud computing. These articles are extended versions of selected papers from the IEEE/ACM Conference on Utility and Cloud Computing that was held in Melbourne, Australia in 2011. These articles encompass the following broad areas:

(1) Cloud pricing and governance
(2) Cloud runtimes and their support for iterative and scientific applications
(3) Storage systems that scale with data volumes
(4) Schemes that target optimizations in virtualized settings

Given the number of cloud services that are now available across different cloud providers, issues relating to the costs of individual services and resources besides ranking these services come to the fore. Javadi et al. address the problem of characterizing spot price instances in public cloud environments. Amazon's spot instances create a competitive bidding option for the users at lower prices without providing reliability on services. This allows one to make trade-offs between price, performance and reliability. The authors analyze spot instances based on a one-year price history of four of Amazon's EC2 datacenters. Spot price dynamics are characterized using a mixture of Gaussian distributions. Breskovic et al. address the drawbacks of static cloud marketplaces that are incapable of adapting to changing market conditions and dynamics of user service requirements. They address this by channeling demand and supply into a few standardized services that are automatically adapted to user requirements at regular intervals. This work also includes support for automated management of Service Level Agreement mappings needed for trading. Garg et al. propose a framework, SMICloud, for measuring the quality of cloud services and prioritizing them. Given the diversity of available cloud services, customers often find it difficult to decide which services they should use and to justify their selection. SMICloud allows customers to evaluate cloud offerings and rank them based on their ability to meet user's Quality of Service (QoS) requirements. This is achieved by using an Analytical Hierarchical Process (AHP) based ranking mechanism that accounts for different dimensional units of the QoS attributes.

As cloud deployments become pervasive, cloud runtimes must be able to support diverse applications beyond the ones that can be efficiently supported using the traditional MapReduce programming model. Ericson et al. contrast the performance of High Dimensional Data Clustering and Classification Algorithms in two cloud runtimes, Hadoop and Granules. These tasks have application in fields such as pattern recognition, datamining, bioinformatics, and recommendation systems. The algorithms contrasted in this work include k-means, fuzzy k-means, Dirichlet, Latent Dirichlet Allocation for clustering along with the Naïve and Complementary Bayes algorithms for classifications. Performance differences stem from how the runtimes support and manage the lifecycle of individual computations, and how they orchestrate exchange of data between different stages of the computational pipeline during successive iterations of the algorithms. Gunarathne et al. describe Twister4Azure a distributed decentralized iterative MapReduce runtime for Windows Azure Cloud. The system supports multi-level caching of loop-invariant data across iterations as well as caching of any reused data and a mechanism to perform cache-aware scheduling. The benchmarks contrast performance in data-intensive applications such as Multi-Dimensional Scaling, k-means clustering, BLAST+ sequence searching, and SmithWaterman sequence alignment. Performance measurements described in the paper show significant performance improvements over traditional MapReduce runtimes deployed on up to 256 instances and for jobs with tens of thousands of tasks.

Data volumes have been increasing at a steady pace over the past decade resulting in the move towards decentralized storage architectures. The ability to cope with the data at scale is critical for several applications that need to assimilate and operate on the data. Malensek et al. describe the design of a high-throughput storage system, Galileo, for storing data streams generated in observational settings. Galileo accounts for the geospatial and chronological characteristics of such time-series observational data streams while separating metadata from content. To support fine-tuning of queries the system supports fast turnarounds for queries with non-matching data and supports range-queries over the spatial and temporal dimensions. Spillner et al. describe, NubiSave, a cloud storage management system that combines storage resources from multiple providers so that redundancy, security, and other nonfunctional properties can be adjusted adequately to the needs of the storage service consumer. NubiSave provides a generic and extensible architecture for building storage controllers encompassing a superset of the most important existing features. The system covers the entire storage service lifecycle from the consumer's perspective.

Virtualization and server consolidation underpin several Infrastructure-as-a-Service systems. We have two articles that target network performance and memory content similarity for latency sensitive applications. Cheng et al. address the problem of network performance isolation in virtualized settings where predicable network bandwidth and low-jittered network latency are desirable for applications such as media streaming. The paper analyzes the causes of unpredictable network latency in virtualized settings and proposes a solution to ensure network performance isolation

by designing a proportional share CPU scheduling with soft real-time support to reduce scheduling delay for network packets and an in network traffic shaper with feedback control to smooth the packet delay. Gerofi et al. describe an approach for improving the performance of highly available virtual machines. Their approach improves the performance of services deployed over replicated virtual machines by exploiting data similarity within the VM's memory image to reduce network traffic during synchronization. To identify similar memory areas, the authors propose a bit density based hash function on which they build a content addressable hash table. The proposed compression method reduces network traffic allowing for faster propagation of changes.

Geoffrey Fox
School of Informatics and Computing, Indiana University
E-mail address: gcf@indiana.edu.

Shrideep Pallickara
Department of Computer Science, Colorado State University
E-mail address: shrideep@cs.colostate.edu.