Cloud Computing: Towards Making Computing a Utility

Mohamed Hefeeda

Qatar Computing Research Institute
Doha, Qatar
mhefeeda@qf.org.qa

Abstract. Cloud computing strives to achieve the long-standing vision of making computing a utility, similar to electricity, telephone, and water services. This article discusses several research challenges that need to be addressed in order to realize the full potential of cloud computing and get computing closer to being a utility.

Key words: Cloud computing, utility computing

Cloud Computing Challenges

The cloud computing paradigm has attracted significant attention from academia, industry, governments, and even individual users. This paradigm promises to achieve the long-standing vision of making computing a utility, similar to electricity, telephone, and water services. This means that computer users receive computing services without worrying about the details of creating, managing, and maintaining the infrastructures providing these services. Just as we receive electricity, for example, without paying too much attention to the complex process of power generation and its associated costs.

Cloud computing offers several advantages, such as reduced cost for setting up and managing IT infrastructures, rapid deployment with elastic scaling up and down of services to meet dynamic user demands, and improved reliability and availability of services. While many algorithms and technologies used in building cloud infrastructures existed before, several new research challenges need to be addressed to realize the full potential of the cloud computing paradigm. These research challenges are summarized in the following subsections.

Cloud Security

Many users perceive more security threats if they were to move their applications and data to a public cloud because of the shared nature of the cloud. While in fact this may not always be the case, since cloud providers typically follow best practices in industry and hire top security experts, way beyond what individual users and organizations can afford. Thus, one of the first tasks in the cloud security area is to clearly identify and document potential security threats resulting

from hosting applications and data on shared cloud infrastructures. To do so, we need to develop a cloud security model that defines standard security metrics, which can be quantified and measured.

A data-centric security model seems to be more appropriate for cloud platforms. In this model, methods for controlling information flow (provenance) within a cloud and across clouds should be developed. Also, end-to-end methods for enforcing security policies should be designed.

In addition, tools to detect and respond to attacks on clouds are needed. These tools should offer multi-level behavior profiling and monitoring, methods for feature selection, data aggregation and correlation, and risk analysis and quantification of various attacks.

Finally, cloud programming models should offer security and privacy-aware APIs, in which users and developers can specify security/privacy requirements of cloud applications.

Cloud Applications

Cloud applications can range from hosting simple desktop applications in a cloud platform to processing web-scale data for creating web indexes for search engines such as Google and Microsoft, and mining social interactions among users for social networking web sites such as Facebook and Twitter. To accelerate wider adoption of cloud platforms for current and future computing applications, we need to identify and characterize "cloudifiable" applications, i.e., applications that can be moved to cloud platforms. This can be done by developing methods and tools to characterize the requirements of cloud applications and to map these requirements to service level agreements (SLAs) offered by cloud platforms.

Also, aggregating and documenting best-practices and case studies for successful (and failed) cloud applications can provide answers to questions such as how and when to cloudify applications. In addition, we should promote "cloud thinking" among users and application developers. Cloud thinking encourages users and developers to think of the cloud as a computing abstraction, not as a number of machines.

Finally, we need to develop management tools and algorithms to:

1. enable automatic scale-out of appli-cations as resources become available,
2. automatically co-schedule applications with complementary resource requirements on the cloud ("compatible multitenancy", e.g., cache-heavy and frequent blocking),
3. support developers to mitigate frequent failures in the cloud ("design for failure"), and
4. provide provable/auditable security requirements.

Cloud Programming

Programming models and tools are essential to design, implement, test, and debug cloud applications. For wide cloud adoption, we need to develop tools

to assist regular users, e.g., scientists and business analysts, to utilize cloud platforms. For example, tools that enable widely-used software packages such as Excel, Matlab, and R to seamlessly utilize cloud infrastructures are needed. These tools should require minimum or zero programming efforts from users. In addition, we should develop multi-level APIs, which can support various granularities for accessing cloud resources.

For example, high-level APIs should be developed to allow cloud users to describe the requirements of their cloud applications without worrying much about the actual programming models used to develop such applications or the hardware resources that will run these applications.

Medium-level APIs should be designed to assist application developers to rapidly develop cloud applications without getting into details such as data replication, caching, fault tolerance, and process scheduling.

Low-level APIs can be used to control cloud resources, e.g., processors, VMs, network, and disk blocks, in fine-grain manner. These APIs should have primitives for specifying and trading off: elasticity, privacy, security, availability, performance, and energy cost for cloud applications.

Finally, we need to identify a small set of programming models for developing cloud applications with diverse requirements, e.g., batch processing, online stream processing, dependency of computation parts on each other, and distribution of input data sets. These programming models should be mathematically formalized in order to provide assurance on cloud applications correctness and performance.

QoS in Clouds

In order to accelerate the adoption of cloud infrastructures by diverse users, cloud providers should consider offering different levels of quality of service (QoS). Clearly specified SLAs for cloud services should be defined. These SLAs should be easy to understand by administrators of IT infrastructures of business with different sizes, which will facilitate moving more applications and data to clouds. SLAs should consider environmental issues, e.g., energy consumption and carbon footprint of applications, as well as application performance metrics such as completion time, availability, and response time. Ideally, SLAs should be transferable from one cloud provider to another. Allocation and management algorithms of cloud resources should be enhanced to enforce SLAs.

Energy-Efficient Clouds

The energy consumption bill makes a sizable portion of the cost of running data centers, and this is portion is increasing relative to other costs including the cost of servers, storage, and networking equipment. We do need to improve the energy efficiency of cloud data centers, not only for reducing costs but also for minimizing the carbon footprint of data centers especially as more of them are being deployed worldwide.

To improve energy efficiency, we first need to define energy consumption metrics for data centers. Current metrics such as PUE (power usage efficiency) are not sufficient as they only give coarse-grain measure for the whole data center. We need more elaborate metrics for the data center as well as for individual applications. Then, we need to design cloud applications that are energy aware, which means that they can adapt their computations based on a given energy budget and they can trade off some performance metrics for energy saving.

In addition, we should consider designing data centers that employ renewable energy sources, such as solar and wind powers. UPS (Uninterruptable Power Supply) units can be utilized to absorb variations and sporadic outages in renewable energy sources. Different organization of UPS units in data centers need to be explored and analyzed. UPS units can be used per server, per rack of servers, per row of racks, or combinations thereof.

Furthermore, research efforts should be targeted to designing servers that approach energy proportionality, as well as data centers that employ low-power processing units such as Graphics Processing Units (GPUs) and asymmetric processors that could have few fast cores and many slower cores.

Finally, we need to develop regulation, taxation, and energy pricing schemes to encourage energy conservation in data centers.

Cloud + X Architectures

We should encourage developers and users to think of cloud as a part of a bigger computing platform in which all components can efficiently be utilized to contribute to the accomplishment of a computational task. For example, parts of a cloud application could run on local desktops or mobile devices while others could run on the cloud. We need to develop resource management tools for clouds composed of heterogeneous elements. We need to design programming models for "Cloud + X" platforms, where X could be a client device, specialized computing resource, or anything else. The programming models should offer services to partition and manage cloud applications.

Cloud Storage Systems

Variability in the performance of cloud storage systems is a major concern for cloud applications. The variability comes from the shared nature of the cloud platform. We first should define the appropriate performance metrics and consistency models for various cloud applications.

Then, we need to improve the cloud middleware layer to reduce performance variability of storage systems. Better schedulers need to be designed to route requests within the cloud storage system in order to meet the performance requirements of cloud applications. In addition, new storage media such as FLASH and tapes should be integrated into cloud storage systems, and tools to efficiently utilize them for different cloud applications need to be designed.

Finally, we need to define guidelines for choosing the appropriate logical storage structure(s) based on the requirements of different cloud applications.

Cloud Legal Frameworks and Standards

Many organizations deal with sensitive data. A clear and legally-binding framework for hosting data and applications is needed. The framework should allow fine access control on data and applications. We also need legal processes to handle hosting data and applications on international clouds. Which laws are enforced on cloud providers? Local or international laws? Currently, many organizations prefer local clouds, which are not always available or efficient.

In addition, organizations would like to have the option to move from one cloud provider to another with minimal effort and disruption of services. We need to design well-defined standards for interoperability across different cloud providers.