

Chapter 35

Cloud-TM: An Elastic, Self-Tuning Transactional Store for the Cloud

João Barreto

Technical University Lisbon, Portugal

Roberto Palmieri

Sapienza Università di Roma, Italy

Pierangelo Di Sanzo

Sapienza Università di Roma, Italy

Paolo Romano

Technical University Lisbon, Portugal

ABSTRACT

By shifting data and computation away from local servers towards very large scale, world-wide spread data centers, Cloud Computing promises very compelling benefits for both cloud consumers and cloud service providers: freeing corporations from large IT capital investments via usage-based pricing schemes, drastically lowering barriers to entry and capital costs; leveraging the economies of scale for both services providers and users of the cloud; facilitating deployment of services; attaining unprecedented scalability levels. However, the promise of infinite scalability catalyzing much of the recent hype about Cloud Computing is still menaced by one major pitfall: the lack of programming paradigms and abstractions capable of bringing the power of parallel programming into the hands of ordinary programmers. This chapter describes Cloud-TM, a self-optimizing middleware platform aimed at simplifying the development and administration of applications deployed on large scale Cloud Computing infrastructures.

INTRODUCTION

The rapidly expanding market of commercial Cloud Computing infrastructures currently offers solutions of different flavors. Depending on the nature of the resources made available on demand by the Cloud platform, such flavors include Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-

Service (SaaS). While some of these solutions are reminiscent of the Application Service Provider (ASP) paradigm, in practice, cloud computing platforms work differently than ASPs. Examples include those offered by Amazon Web Services, AT&T's Synaptic Hosting, AppNexus, GoGrid, Rackspace Cloud Hosting, and to an extent, the HP/Yahoo/Intel Cloud Computing Testbed, and the IBM/Google cloud initiative.

DOI: 10.4018/978-1-4666-6539-2.ch035

Instead of owning, installing, and maintaining the software for their costumers (often in a multi-tenancy architecture), cloud computing vendors typically maintain little more than the hardware, and give customers a set of virtual machines in which to install their own software. However, getting additional computational resources is not as simple as a magic upgrade to a bigger, more powerful machine on the fly (with commensurate increases in CPUs, memory, and local storage); rather, the additional resources are typically obtained by allocating additional server instances to a task. For example, Amazon's Elastic Compute Cloud (EC2) apportions computing resources in small, large, and extra large virtual private server instances, the largest of which contains no more than eight cores. If an application is unable to take advantage of the additional server instances by offloading some of its required work to the new instances which run in parallel with the old instances, then having the additional server instances available will not be much help.

Thus, one of the main challenges that needs to be faced to bring about the potential of cloud computing, and ultimately consolidate its business model, is the development of programming models and tools that simplify the design and implementation of applications for the cloud, so as to bring the power of parallel computing into the hands of ordinary programmers.

Unfortunately, designing and implementing software services that are actually able to match the scalability potentialities of large scale, shared-nothing Cloud infrastructures is far from being a trivial task.

One of the most crucial issues to tackle when developing large scale distributed application is certainly related to how to manage concurrent manipulations to the shared state of the application. The challenge here is to identify mechanisms that able to ensure adequate consistency levels while being:

1. Simple and familiar for the programmers, highly efficient and scalable.
2. Fault-tolerant and highly available.

Decades of literature and field experience in areas such as replicated databases, Web infrastructures, and high performance computing have led to the development of a plethora of different approaches to ensure state consistency in distributed platforms, and taught a fundamental, general lesson. The design space of distributed state consistency mechanisms is so vast that no universal, one-size-fits-all solution exists, as the efficiency of individual state management approaches is strongly affected by both:

1. The characteristics of the incoming workload, such as the ratio of read/write operations, as well as the spatial/temporal locality in the data access patterns.
2. The scale of the system, e.g. number of nodes and local vs. geographical distribution.

The complexity of this problem is hence further exacerbated in Cloud Computing platforms precisely because of the feature that is regarded as one of the key advantages of the cloud: its ability to elastically acquire or release resources, de facto dynamically varying the scale of the platform in real-time to meet the demands of varying workloads.

This chapter describes the architecture of a novel middleware platform for service implementation in Cloud Computing platforms that is being developed in the context of the EU project Cloud-TM.

At the core of the Cloud-TM platform lies the abstraction of a Distributed Software Transactional Memory (DSTM). DSTM is a recently proposed extension of the Transactional Memory (TM) programming paradigm, which was originally introduced to simplify the development of concurrent, though not distributed, programs.

31 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/cloud-tm/119882

Related Content

Social Implications of Big Data and Fog Computing

Jeremy Horne (2018). *International Journal of Fog Computing* (pp. 1-50).

www.irma-international.org/article/social-implications-of-big-data-and-fog-computing/210565

Sharing Medical Information by Means of Using Intelligent Agents and Cloud Computing

Mauricio Paletta (2014). *Cloud Computing Applications for Quality Health Care Delivery* (pp. 140-170).

www.irma-international.org/chapter/sharing-medical-information-by-means-of-using-intelligent-agents-and-cloud-computing/110433

An Authentication Technique for Accessing De-Duplicated Data From Private Cloud Using One Time Password

Prakash Mohanand Saravanakumar Chelliah (2019). *Cloud Security: Concepts, Methodologies, Tools, and Applications* (pp. 435-445).

www.irma-international.org/chapter/an-authentication-technique-for-accessing-de-duplicated-data-from-private-cloud-using-one-time-password/224586

Evaluating the Performance of Monolithic and Microservices Architectures in an Edge Computing Environment

Nitin Rathoreand Anand Rajavat (2022). *International Journal of Fog Computing* (pp. 1-18).

www.irma-international.org/article/evaluating-the-performance-of-monolithic-and-microservices-architectures-in-an-edge-computing-environment/309139

Fog Computing Qos Review and Open Challenges

R. Babu, K. Jayashreeand R. Abirami (2018). *International Journal of Fog Computing* (pp. 109-118).

www.irma-international.org/article/fog-computing-qos-review-and-open-challenges/210568