

Chapter 20

Fault Tolerant Architecture to Cloud Computing Using Adaptive Checkpoint

Ghalem Belalem

University of Oran (Es Senia), Africa

Said Limam

University of Oran (Es Senia), Africa

ABSTRACT

Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. Failures of any type are common in current datacenters, partly due to the number of nodes. Fault tolerance has become a major task for computer engineers and software developers because the occurrence of faults increases the cost of using resources and to meet the user expectations, the most fundamental user expectation is, of course, that his or her application correctly finishes independent of faults in the node. This paper proposes a fault tolerant architecture to Cloud Computing that uses an adaptive Checkpoint mechanism to assure that a task running can correctly finish in spite of faults in the nodes in which it is running. The proposed fault tolerant architecture is simultaneously transparent and scalable.

1. INTRODUCTION

Cloud computing can be defined as a new style of computing in which dynamically scalable and often virtualized resources are provided as a services over the Internet. Cloud computing has become a significant technology trend, and many

experts expect that cloud computing will reshape information technology (IT) processes and the IT marketplace. With the cloud computing technology, users use a variety of devices, including PCs, laptops, smartphones, and PDAs to access programs, storage, and application-development platforms over the Internet, via services offered by cloud computing providers. Advantages of the

DOI: 10.4018/978-1-4666-1879-4.ch020

cloud computing technology include cost savings, high availability, and easy scalability (Borko & Armando, 2010).

With the development of virtualization technology, more and more companies are using virtual machines to improve the utilization of resources. In a cloud computing environment, companies may purchase computing and storage capability from cloud providers, such as Amazon EC2, and then use the purchased resources to provide services, such as web application services, to users. Technically, Amazon EC2 (Kaushal & Anju, 2011) assigns virtual machines to those companies which deploy the application instances on those VMs. It is expected that many instances of the same application can run on different virtual machines in the same cloud environment (<http://aws.amazon.com/ec2/>).

Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS), so we use that term. The datacenter hardware and software is what we will call a Cloud (Armbrust et al., 2009).

For cloud service, two issues are of great importance, the reliability and the performance of cloud service. Cloud service reliability is concerned with how probable that the cloud can successfully provide the service requested by users. Cloud service performance, on the other hand, is concerned with how fast the cloud can provide the requested service.

As a rule of thumb, all computer engineers know that when a computer system becomes more complex, the system's susceptibility to faults increases. Such rule is fully applicable to Cloud Computing.

Failures of any type are common in current datacenters, partly due to the number of nodes. In spite of the quality improvement in each single component of the nodes, one may argue that, as certain as the sun will rise tomorrow, any com-

ponent will suffer some kind of failure: a disk can break down or a link can fail anytime, and this can lead to computation time wasting when executing medium and long-running tasks and the crash occurs before finishing the task execution. For instance, if a task that takes 24 hours is executing in a node, and this node crashes 5 minutes before the task execution ends, almost one day of execution will be wasted (Goiri et al., 2010).

Dealing with fault tolerance has become a major task for computer engineers and software developers because the occurrence of faults increases the cost of using resources. However, the inclusion of fault tolerance in a system increases the complexity of such system from the user point of view.

In this paper, we proposed a fault tolerant architecture to Cloud Computing that uses a Checkpoint mechanism to assure that a task running can correctly finish in spite of faults in the nodes in which it is running. This mechanism records the system state periodically to establish recovery points. Upon a node crash, the last checkpoint can be restored, and task execution can be resumed from that point. In the development of such architecture, we assumed that, in order to attend the user expectations, the first user expectation is, of course, that his/her application correctly finishes independent of faults in the node. This is obviously the main obligation of any fault-tolerant architecture for Cloud Computing and demands no further discussion. The next requirement is that the fault-tolerant architecture must be easy to use. A fault-tolerant architecture for Cloud Computing must simultaneously be transparent, and scalable.

The rest of this paper is organized as follows. Section 2, the Checkpoint mechanism is presented. In Section 3 we summarize important related works in Cloud Computing. Section 4 presents our fault tolerant architecture, Section 5 shows experiments and Section 6 concludes the paper.

8 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/fault-tolerant-architecture-cloud-computing/67908

Related Content

MapReduce and Hadoop

Luis Rodero-Merino and Gilles Fedak (2012). *Open Source Cloud Computing Systems: Practices and Paradigms* (pp. 197-215).

www.irma-international.org/chapter/mapreduce-hadoop/62371

Respecting the Deal: Economically Sustainable Management of Open Innovation Among Cooperating Companies

Riccardo Bonazzi, Arash Golnam, Yves Pigneur and Alain Wegmann (2012). *International Journal of E-Services and Mobile Applications* (pp. 23-41).

www.irma-international.org/article/respecting-deal-economically-sustainable-management/62253

Fostering FinTech Through Mobile Wallets: An Indian Scenario

R. Arunachalam and R. Amudha (2026). *Emerging Trends and Innovations in Financial Services: A Futurology Perspective* (pp. 21-48).

www.irma-international.org/chapter/fostering-fintech-through-mobile-wallets/384107

The Role of E-Services in the Transition from the Product Focus to the Service Focus in the Printing Business: Case Lexmark

Esko Penttinen, Timo Saarinen and Pekka Sinervo (2009). *Cases on Managing E-Services* (pp. 156-165).

www.irma-international.org/chapter/role-services-transition-product-focus/6420

Plan Optimization by Plan Rewriting

José Luis Ambite, Craig A. Knoblock and Steven Minton (2005). *Intelligent Techniques for Planning* (pp. 121-161).

www.irma-international.org/chapter/plan-optimization-plan-rewriting/24461