TRANSACTION PROCESSING APPLICATIONS IN CLOUD COMPUTING

Filip Mitrevski
St. Kliment Ohridski University
Faculty of Information and Communication Technologies
Bitola, Macedonia

Darko Pajkovski
St. Kliment Ohridski University
Faculty of Information and Communication Technologies
Bitola, Macedonia

Tome Dimovski
St. Kliment Ohridski University
Faculty of Information and Communication Technologies
Bitola, Macedonia

ABSTRACT

Cloud computing is a recent attractive term in the IT world. The term "Cloud Computing" comes out of the idea for centralizing the storage and computation in distributed data. Its long term goals are to provide a flexible, on – demand package to the cloud user, giving him much more freedom, flexibility and reliability at the same time, achieving all of the above by using a simple "utility computing model". It promises to bring on-demand pricing, less IT overhead and an ability to scale IT up and down quickly.

The focus of this work falls down on transaction processing applications which work in multi – processing and cloud environments. All major vendors have adopted a different architecture for their cloud services. As a result, in this paper we will be reviewing some of them and their fundamental approaches on improving Cloud Transactions.

I. INTRODUCTION

Transaction processing has been an important software technology in the last five decades. The government, telecommunication sectors, finance, transportation and military are all dependent on the transaction processing applications for their services, namely order processing, banking, electronic reservations, telephone switching, etc. Transaction processing systems are used by many large hardware and software vendors such as IBM, Microsoft, Google, Amazon, Oracle, Dell and their revenue for transaction processing products and services is in the tens of billions of dollars per year.

Cloud computing is a computing service offered over the Internet, in which the software is seen as a service and the applications and data are stored on multiple servers (locations).

In the current cloud computing architecture (Fig.1) there are data centers which are able to provide services to all of the clients participating in the same cloud.

Cloud computing allows us to move the processing effort from the local devices such as laptops, personal computers from various locations, to the data center facilities. For example, in such a way, any device could be able to solve some complex differential equations by simply passing the specific arguments to a data center service which will be capable to give back the result in a very short time. However, the security of data and applications becomes a very major issue.

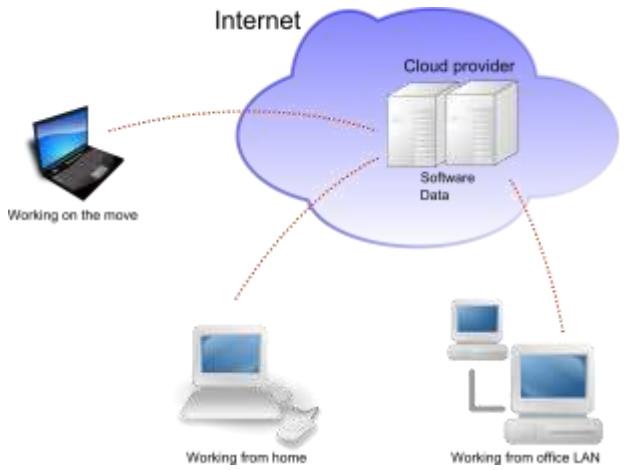

Figure 1: Cloud Computing Architecture

The main benefits of the cloud computing are the following:

- Flexibility – If our needs increase over time, it is very easy to scale up the cloud capacity, likewise if our needs scale down again, the flexibility is baked into the service.
- Disaster recovery – There are several solutions implemented as a "cloud-based" backup and recovery that save cut down the time penalty.
- Automatic software updates – Eliminate the need to spend time maintaining the system manually.
- Increased collaboration – Cloud based workflow and file sharing applications help to provide the updates in real time, gives them full visibility of their collaborations.
- Work from anywhere – Any device connected to the internet is able to do the job.
- Security – If the client computer crushes, there is almost nothing lost because everything is stored into the cloud in real time.

One of the main advantages of cloud computing is the promise of (virtually) infinite scalability so that IT administrators needn't worry about peak workloads. Finally, cloud computing provides flexibility and reliability at the same time in the utilization and management of both hardware and

software, which translates into savings in both "production time" and cost.

The major players in this field of cloud computing are Google, Amazon, Yahoo, Microsoft and some hardware manufacturers like IBM, Dell, HP, Intel. In this paper we will be reviewing some of them and their services, applications and fundamental approaches on improving cloud transactions.

II. RELATED WORK

With the emergence of cloud computing, some studies have evaluated the performance, scalability and reliability of cloud computing infrastructures. Some of them compared the performance of Hadoop, like an open source Java-based programming framework versus the more traditional (SQL-based) database systems [1]. The results of related studies on cost-consistency trade-off for OLTP workloads in the cloud have been reported [2].

OLTP (online transaction processing) is a special "class" consisted of software programs, capable of supporting transaction-oriented applications on the Internet, which is a much better approach compared to the traditional "distributed transaction" model – companies have many challenges regarding the performance when dealing with multiple nodes and databases required for mission-critical transactions. The second challenge comes from a case when the companies are highly distributed and they are communicating with their business partners who might be located all around the world. These are the main reasons why the traditional "distributed transaction" model is slow and unreliable.

Managing data in highly distributed OLTP environments means containing customer and product data which must be read from and written to constantly and in real-time in order to support the quality of each transaction. For example, this type of transaction takes place when we take out money at an ATM machine. Once our card is validated, a debit transaction takes place against our current balance to reflect the amount of cash that is being withdrawn. This type of transaction also takes place when we deposit money into our accounts and the balance gets an update.

III. DISTRIBUTED DATABASE ARCHITECTURES

Recently, distributed database architectures have found a place in cloud computing. First, the classic multi-tier database application architecture is described and then, two other variations of this architecture are also described such as replication, partitioning [12].

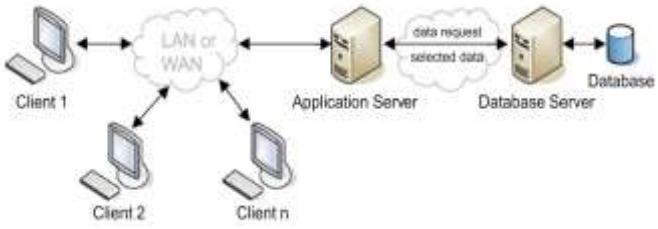

Figure 2: Distributed Database Architecture

A. Classic

Requests from the clients are dispatched to an available machine which runs a web and an application server. Afterwards, a web server handles the HTTP request from the clients and the application server executes the code specified in some program language with embedded SQL, which is shipped to the database server and interprets this request, returns a result and updates the database. The interface between the database server and the database itself includes shipping physical blocks of data (64K blocks), by using GET and PUT requests.

B. Partitioning

The difference between classic database architecture and partitioning is simple: The usage of a separate database server for controlling each partition in a database which is logically partitioned. In any database literature there are many examples for partitioning schemes: vertical partitioning vs. horizontal partitioning, round-robin vs. hashing vs. range partitioning [3].

For cloud computing the database architecture for partitioning was first founded by Force.com, by the platform that runs the "Salesforce" application. In Force.com, the partitioning includes a whole server-side application stack, with web and application servers. Here, all the requests to the same tenant are handled by the same app, web and database server.

C. Replication

In replication as with partitioning, there are several database servers and each of them saves copy of the whole database. The most important characteristic of replication is the mechanism to keep the replicas consistent. The replication is recommended to be transparent, therefore the requests are routed automatically to the Master. If the replication is not transparent, certain applications direct all update requests to the database server, which controls the Master copy and the Master server propagates all committed updates when these have been successfully committed.

IV. CLOUD SERVICES

In this section we will describe the alternative services offered by several big players of cloud computing world: namely Amazon (AWS), Google, Microsoft and Oracle. Their services are different in many aspects like the business model which is in use, software components used at all tiers and by the programming model. Only the Google App engine provides a service which can do automatic scalability and persistent data storage whilst fully automated hardware resources for all tiers. Amazon provides a service called "AutoScaling" which is used to automatically scale-out and scale-down EC2 machines for web tier.

A. Google App-Engine

Google App-Engine [5] is a platform for development and deployment of web applications which provides a whole platform as a service ("PaaS") to the cloud users. Several languages are supported in Google App-Engine like Python, Java and any extension of JVM languages, both with

embedded SQL for access to the database. The main advantage of the Google App-Engine is the automatic scaling of the resources consumed by an application, depending on the workload, so the cloud user doesn't have to worry about the spikes in the traffic or data. These applications [4] are intended for social networking start-ups, event-based websites and public institutions (school, universities, governments) etc.

Google App-Engine provides live migration on engine's instances to nearby hosts while active-even while under extreme load (up to 1.5 TB with their working SSD storage). Also, Google App-Engine offers optimal pricing for per-minute-billing, sustained-use, and special pricing for particular use.

B. Amazon EC2

One of the leading web based services that Amazon provides to the public, is Amazon's Elastic Compute Cloud, or simply called EC2[7]. EC2 is a web service, which provides resizable and secure compute capacity in cloud environments. At the same time, it has been specifically designed for the developers in order to make web-scale cloud computing easier. One of the main advantages that EC2 provides is the simple web service interface, which allows the user to control and obtain the desired capacity with minimal effort, providing total control over the computing resources. EC2 also reduces the time required to boot server instances to just a few minutes. What makes EC2 "unique" is the fact that it allows the user to pay only for the capacity that he actually uses – cuts down unnecessary cost and space usage to the server side as well.

Amazon EC2 Features:

- Elastic Web-Scale Computing: Increasing and decreasing capacity in a few minutes, commissioning many server instances simultaneously.
- Completely Controlled: Complete control of the instances together with root access, remote control using web service APIs
- Flexible Cloud Hosting Services: Multiple instance types, software packages and operating systems.
- Integrated Design: EC2 is integrated in most AWS services
- Reliability: Proven infrastructure, highly reliable environment.
- Security: EC2 works in pair with Amazon VPC, providing robust networking.
- Pre-defined instances: Allows the in-experienced user pre-built packages according to his needs, ranging from General Purpose Instances, which provide a certain "baseline" level of performance, up to GPU Graphics Instances, featuring up to several NVIDIA GRID GPU's.
- Elastic Load Balancing: Achieving greater fault tolerance, dynamically providing the correct amount of load balancing needed to response the incoming traffic, detection of unhealthy instances.

C. Gnubila – GPaaS (G Platform as a Service)

One of the products that *gnubila* provides is G Platform as a Service, or GPaaS [8]. GPaas is based on the G Platform, which evolved from a "simple" application server into a native cloud platform which includes DBMS, BPM and middleware technologies. All of this is coupled together with a kit of tools for maintenance for applications and services. A development framework which supports a user interface which provides infrastructure capabilities, session management (authentication, authorization, protection) and most importantly, transaction integrity, scalability and reliability are being supplied from the integrated development and deployment tools.

GPaas includes an environment dedicated for development, named GDeveloper, which can customize the User Interface based on HTML, CSS and JavaScript, making it easy for people to work together with social networks.

GPaaS advantages for developers:

- Faster time to market: Using Metadata as an approach instead of code makes changes to the application easy
- Simplified Deployment: All focus is put towards development and innovation
- Scalability: Use of hardware on demand, eliminates the need to write code for scalability
- Simplified application architecture: Fast data access, eliminates bottlenecks found in web applications, completely configurable
- Flexible data model
- Hybrid architecture: Providing transitions from local machines and server to a cloud model.
- Data management from multiple databases

D. Microsoft Azure

Microsoft Azure [6] represents a set of cloud services using .NET and SQL Server. The focus of this service falls in the category of Platform as a Service (PaaS). It is actually between a complete application framework like Google App-Engine and hardware virtual machines solutuon like Amazon EC2.

By using Windows Azure, customers can run applications and stored data on internet accessible machines owned by Microsoft. Applications in Microsoft Azure run only in user mode - no administrative access is allowed here. The primary goal of Windows Azure platform is to support a large number of simultaneous users compared to the Google App-Engine, which is more interested in small applications with light workloads. Also, Windows Azure differs from Amazon EC2 and Google App-Engine with regard to the pricing. Windows Azure charges a monthly or hourly flat fee depending on the database size with unlimited connectivity and virtual machine size.

E. CloudTran & Oracle Coherence

CloudTran [9] is specifically made to enable developers easy use of the IMDG (In-memory data grids) architecture. It provides ACID-property transactions across nodes. CloudTran is also "Built for Scale" – enables the usage of industry

standard components in order to quickly and easily build scalable, transactional applications. This gives us transactions with ACID-property without using phase commits, therefore the applications scale and run at grid speed all the time. CloudTran is suitable for use in environments where there is a need for scalable and fast transactions, coupled with easy recoverability of data.

CloudTran represents a middleware solution that works alongside, and also uses the features of Oracle's Coherence. Data which is bigger than the memory cache size is partitioned into multiple sets. All of the data sets are stored across nodes in the cluster – stores all the data which is required for an application in IMDG.

Oracle Coherence represents an IMDG tool which can take care of data objects stored in the RAM across servers organized in a cluster. The number of servers [10] can be modulated easily, together with the amount of RAM that is available. IMDGs use distributed caching in order to increase the performance and reduce the latency to existing databases.

Oracle Coherence provides several core functions and benefits:

- Caching – Consistent view of cached data, making data analysis easy for applications – maximizing the parallel capabilities of the data grid.
- Analytics – Aggregating and sorting of data, parallelizing operations across an entire data grid, ensuring that server failures do not affect calculation results.
- Transactions – Guaranteed data consistency in extreme transaction procession workloads.
- Events – Event handling mechanisms, capable of dealing with intense event rates, such as stream processing and continuous query for desktop applications.

When compared to other leading products, such as EC2 for example, Coherence provides scalability and flexibility in a similar way, enabling applications to scale linearly and dynamically in order to reduce and make the cost more predictable and to maximize the resource utilization. By offloading the processing from back end systems [11], Coherence supports continually growing application loads with minimal risk of data loss.

V. Conclusion

Cloud computing is a recent attractive term in the IT world that provides a service which is offered over the Internet, where the applications and the data are stored on multiple servers (locations). Corporations like Google, Amazon, Microsoft, Oracle are already providing cloud services. Their products in the likes Google App-Engine, Amazon EC2, Microsoft Azure, CloudTran and Gnubila are one of the best in the market with their ease of use, availability aspects, reliability and flexibility. At the same time they are achieving all of the above by using a simple utility computing model.

In this paper we have evaluated the benefits from each product as a package, enabling users to choose what is best for them, depending on their specific needs.

VI. References

[1] A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden, and M. Stonebraker. A Comparison of Approaches to Large-Scale Data Analysis. In Proc. of SIGMOD, pages 165–178, 2009 *(references)*

[2] T. Kraska, M. Hentschel, G. Alonso, and D. Kossmann. Consistency Rationing in the Cloud: Pay Only when it Matters. In Proc. of VLDB, volume 2, pages 253–264, 2009

[3] S. Ceri and G. Pelagatti. Distributed databases principles and systems. McGraw-Hill, Inc., 1984.

[4] Persistent Systems. Google app engine. http://www.persistentsys.com

[5] Intoducing Google app engine https://cloud.google.com/

[6] D. Chappell. Introducing windows azure. http://go.microsoft.com/ December 2009

[7] Amazon Elastic Compute Cloid: https://aws.amazon.com/ec2/

[8] G Platform as a Service: http://www.gnubila.com/geas/gpaas

[9] CloudTran Overview: http://www.cloudtran.com/productOverview.php

[10] Indu Arora, Dr. Anu Gupta. Modeling and Designing Integrated Framework for Data Management of Transactional Applications in Cloud. http://www.ijser.org/researchpaper%5CModeling-and-Designing-Integrated-Framework-for-Data-Management.pdf

[11] Features and benefits of Oracle Coherence: http://www.mythics.com/about/blog/the-features-and-benefits-of-oracle-coherence

[12] Donald Kossmann, Tim Kraska and Simon Loesing. An Evaluation of Alternative Architectures for Transaction Processing in the Cloud, pages 2-3, 2010