

Cloud Computing-System Implementation for Business Applications

S. Silas Sargunam

Department of Management Studies, Anna University Regional Campus, Tirunelveli, India

Email: silassargunam@gmail.com

Received 23 March 2016; accepted 13 May 2016; published 18 May 2016

Copyright © 2016 by author and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Nowadays, companies are faced with the task of processing huge quantum of data. As the traditional database systems cannot handle this task in a cost-efficient manner, companies have built customized data processing frameworks. Cloud computing has emerged as a promising approach to rent a large IT infrastructure on a short-term pay-per-usage basis. This paper attempts to schedule tasks on compute nodes so that data sent from one node to the other has to traverse as few network switches as possible. The challenges and opportunities for efficient parallel data processing in cloud environments have been demonstrated and Nephele, the first data processing framework, has been presented to exploit the dynamic resource provisioning offered by the IaaS clouds. The overall utilisation of resources has been improved by assigning specific virtual machine types to specific tasks of a processing job and by automatically allocating or deallocating virtual machines in the course of a job execution. This has led to substantial reduction in the cost of parallel data processing.

Keywords

Data Processing, Schedule Tasks, Resource, Cloud Computing

1. Introduction

Today many companies are processing huge amounts of data in a cost-efficient manner. Classic examples are operators of Internet search engines, like Google, Yahoo, or Microsoft. The vast amount of data they have to continuously deal with has made traditional database solutions prohibitively expensive. Instead, these companies have popularized an architectural paradigm based on a large number of commodity servers. Problems are split into several independent subtasks, distributed among the available nodes, and computed in parallel. Many of these companies have also built customized data processing frameworks. Examples are Google's Map Reduce, Microsoft's Dryad, or Yahoo!'s Map-Reduce-Merge.

Cloud computing has emerged as a promising approach to rent a large IT infrastructure on a short-term pay-per-usage basis. Operators of Infrastructure-as-a-Service (IaaS) clouds, like Amazon EC2, let their customers allocate, access, and control a set of Virtual Machines (VMs) which run inside their data centres and only charge them for the period of time the machines are allocated.

The cloud's virtualized nature helps to enable promising new use cases for efficient parallel data processing. However, it also imposes new challenges. This paper attempts to schedule tasks on compute nodes so that data sent from one node to the other has to traverse as few network switches as possible. And the overall throughput of the network can be improved by minimising bottlenecks.

2. Literature Survey

M. D. Dikaiakos [1] has analyzed the architecture and management of cloud computing infrastructures highlighting several outstanding issues such as security, privacy and power efficiency. He has concluded that the hosted applications' business models must have a well defined pathway to monetizing cloud computing.

Hakan Erdogmus [2] concludes that the core concept of cloud computing revolves around software as a service, or SaaS. He also concludes that from this core concept, a complex concoction of paradigms, concepts, and technologies of cloud computing emerges.

Sean Marston *et al.* [3] have identified various issues and has observed that while the researchers in the computer science community are making rapid progress, equal importance should be given to the business perspective.

Christ Weinhardt *et al.* [4] have analysed the business opportunities of the Cloud Computing paradigm and has presented a framework of business model for Clouds. He has also discussed the challenges involved in the implementation of cloud computing.

Lijun Mei [5] has highlighted research challenges in task decomposition, task distribution, and task coordination. He has also analyzed various issues of implementing cloud computing.

Paul. T. Jeager [6] has defined cloud computing as a computing platform that resides in a large data center and is able to dynamically satisfy a wide range of needs, from scientific research to e-commerce. Cloud computing issues are viewed as a part of larger issues of public policy attempting to respond to rapid technological evolution.

Mladen. A. Vouk [7] has analysed the security related issues of cloud computing. He has also studied the commercial aspects of cloud computing in terms of Return on Investment and Total Cost of ownership.

The literature survey has shown that there exists a need for task decomposition, task distribution, and task coordination in order to further improve the efficiency of cloud computing. The task is to exploit the dynamic resource provisioning offered by the IaaS clouds in order to achieve efficient parallel data processing in cloud environments. The overall resource utilization is attempted to be improved by assigning specific virtual machine types to specific tasks of a processing job and by automatically allocating/deallocating virtual machines in the course of a job execution.

3. System Requirements

The capability to exploit the dynamic resource provisioning to assign specific virtual machine types to specific tasks of a processing job, as well as the possibility to automatically allocate or deallocate virtual machines in the course of a job execution, will improve the overall resource utilization and, consequently, reduce the processing cost.

The requirements of the system in terms of hardware and software specifications are given below.

3.1. Hardware Configuration

- Pentium IV 2.4 GHz.
- 40 GB + Hard Disk.
- 512 MB RAM.

3.2. Software Configuration

- Operating system: Windows XP.
- Platform: Java.
- Tool: CloudSim.

- Back End: MySQL.

Nephele is a processing framework which is explicitly designed for cloud environments. It has the capability to dynamically allocate or deallocate different computing resources from a cloud.

4. Module Description

4.1. Network Module

Server-Client computing or networking is a distributed application architecture that partitions tasks or workloads between service providers (servers) and service requesters, called clients. A server machine is a high-performance host that is running one or more server programs which shares its resources with clients. Clients therefore initiate communication sessions with servers which await incoming requests.

4.2. Location Based Services (LBS)

In particular, users are reluctant to use LBSs, since revealing their position may link to their identity. When a user wishes to pose a query, she sends her location to a trusted server, the anonymizer through a secure connection. The latter obfuscates her location, replacing it with an Anonymizing Spatial Region (ASR). The ASR is then forwarded to the LS. The LS retrieves a candidate set (CS) that is guaranteed to contain the query results for any possible user location inside the ASR. The AZ receives the CS and reports the subset of candidates that corresponds to her original query.

4.3. System Model

The ASR construction at the anonymization process abides by the user's privacy requirements. Particularly, specified an anonymity degree K by u , the ASR satisfies two properties: (i) it contains u and at least another $K - 1$ users, and (ii) even if the LS knew the exact locations of all users in the system.

- This system proposes an edge ordering anonymization approach for users in road networks, which guarantees K -anonymity under the strict reciprocity requirement.
- This System identifies the crucial concept of border nodes, an important indicator of the CS size and of the query processing cost at the LS.
- This system considers various edge orderings and qualitatively assesses their query performance based on border nodes.
- This System design efficient query processing mechanisms that exploit existing network database infrastructure, and guarantee CS inclusiveness and minimality. Furthermore, they apply to various network storage schemes.
- This System devise batch execution techniques for anonymous queries that significantly reduce the overhead of the LS by computation sharing.

4.4. Scheduled Task

Recently, considerable research interest has focused on preventing identity inference in location-based services. This offers privacy protection in the sense that the actual user position u cannot be distinguished from others in the ASR, even when malicious LS is equipped/advanced enough to possess all user locations. This spatial K -anonymity model is most widely used in location privacy research/applications, even though alternative models are emerging.

4.5. Query Processing

In this case, the queries are evaluated in a batch. The network-based anonymization and processing (NAP) framework, the first system for K -anonymous query processing in road networks is used. NAP relies on a global user ordering and bucketization that satisfies reciprocity and guarantees K -anonymity. The ordering characteristics that affect subsequent processing are identified and the alternatives are qualitatively compared. Subsequently, the query evaluation techniques that exploit these characteristics are proposed. In addition to user privacy, NAP achieves low computational and communication costs, and quick responses overall. It is readily deployable, requiring only basic network operations.

5. System Architecture

The architecture of the system adopted in the study has been presented schematically in **Figure 1**.

Client-Server networking is a distributed application architecture that partitions tasks between service providers and service requesters. A server machine is a high-performance host that is running one or more server programs which shares its resources with clients. A Virtual Machine (VM) must be initially started by the user in the cloud.

The job manager is the central component for communicating with clients, creating schedules for incoming jobs, and supervising the execution of the jobs. If a job graph is submitted from a client to the job manager, each task of the job will be sent to a task manager. The Job Manager receives the jobs, schedules them and coordinates the execution of these jobs. The Job Manager corresponds with the interface to control starting of the VMs. This interface is called the Cloud Controller. With the help of the Cloud Controller, the Job Manager can allocate/deallocate VMs. This is done based on the current job being executed.

A task manager receives tasks from the job manager and executes them. After having executed them, it reports the execution result back to the job manager. A Task Manager informs the Job Manager about their completion or possible errors. The task manager also periodically reports to the job manager that it is still running.

The execution of job tasks is carried out by a set of instances. Every instance runs a Task Manager (TM). Upon receipt of job, the Job Manager then decides, how many and what type of instances the job should be executed and when the corresponding instances must be allocated/deallocated to ensure a cost-efficient processing.

6. System Implementation

Implementation is the carrying out, execution, or practice of a plan, a method, or any design for doing something. In an information technology context, implementation encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, running, testing, and making necessary changes.

6.1. User Involvement

Incorporating user knowledge and expertise leads to better solutions. User participation in the design and operation of information systems has several positive results. First, if users are duly involved in systems design, their priorities and business requirements can be incorporated. Second, they are more likely to respond positively to the change process.

6.2. Debugging

Debugging is a methodical process of finding and reducing the number of defects. Debugging tends to be harder when various subsystems are tightly coupled, as changes in one may cause bugs to emerge in another.

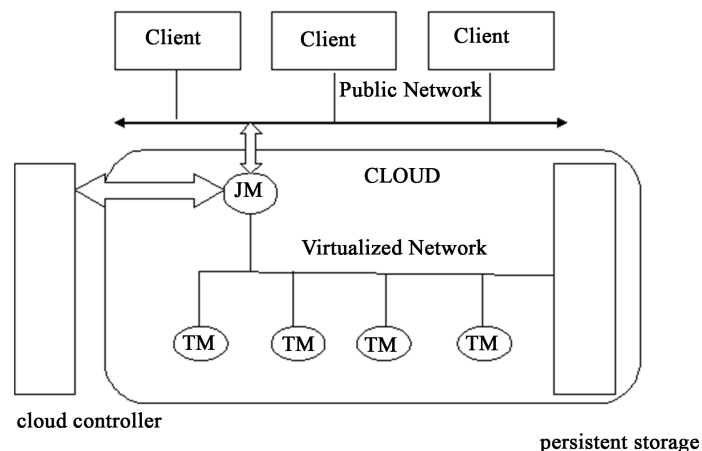


Figure 1. System architecture.

6.3. Maintenance

Project management is the discipline of planning, organizing, securing, and managing resources to achieve specific goals. A project is a time-bound endeavour with specific objectives and often constrained by funding or deliverables. The primary challenge of project management is to achieve the project goals and objectives while honouring the preconceived constraints. The other challenge is to optimize the allocation of inputs necessary to meet pre-defined objectives.

7. Evaluation

Testing encompasses a set of activities that are conducted systematically to confirm whether the project has achieved the ultimate goal or not. This begins at the module level and works towards the integration of the entire system.

7.1. Unit Testing

Unit testing focuses on testing the modules of the system independently of one another and identify error. The needed test cases were developed for various modules in the system. All the modules are tested to display the requested details and to ensure navigation between pages visited and settings. The setup module is tested for proper parameters necessary for connections. The schema comparator module is checked for all the constraints necessary to perform the comparison. The data pipe line module is checked for integrity of values after pipelining. The query analyzer module is tested for generation of syntax based on the selection. Thus all the modules are unit tested.

7.2. Integration Testing

After each module has been tested individually, they were integrated and the system underwent integration testing for its correctness and consistency. The top-down integration scheme is used to test the modules. The main module is tested first followed by the sub-modules to detect errors that could occur due to added complexities. The system elements have been properly integrated and it is found that they perform their tasks as expected.

7.3. Functional Testing

Functional tests were performed for each module to test whether the functional requirements of each of the modules is satisfied. The functional test ensures that the system produces the required results for the given inputs under typical operating conditions. To justify the functional test, consider that the user wants to pipeline a table form the user. It then gets the pipeline option from the user and the pipelines the data from source to destination and then returns the number of rows pipelined. Thus the functionality is ensured.

7.4. Validation Testing

The final step involves validation testing, which determines whether the software functions as the user expected. The end user rather than the system developer conducts this test. Most software developers conduct this test as a process called “Alpha and Beta Testing”.

The compilation of the entire project is based on the full satisfaction of the end users. In the project validation testing is made in various forms. In questions entry form, only the correct answer will be accepted in the answer box.

8. Result

The performance of Nephele is compared with that of Hadoop which is an open source software. Hadoop has been designed in such a way as to run on multitudes of nodes in IaaS clouds.

Map Reduce programme was chosen to test the comparative performance of Hadoop and Nephele. The incoming data words between the map and the reduce phase of each input data set were counted using the word count code. The size of the files and the time taken for executing the word count by Hadoop and Nephele are shown in [Figure 2](#).

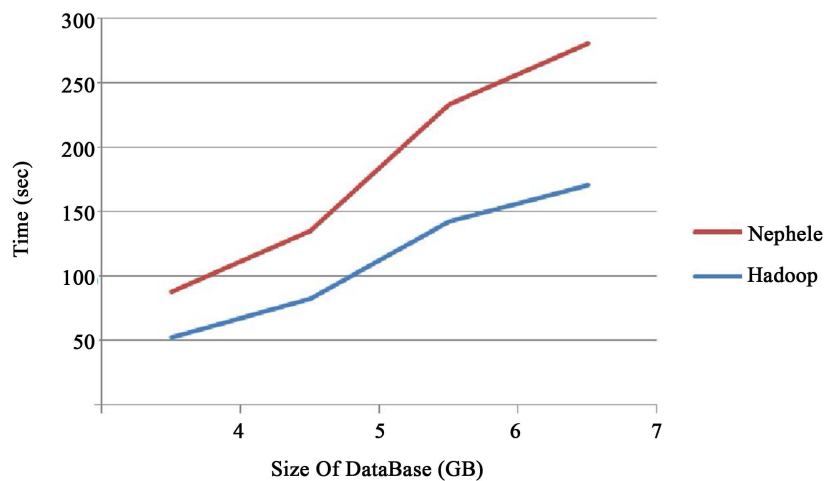


Figure 2. Comparative performance of Nephele and Hadoop.

It could be inferred from the graph that the average time taken by Nephele to execute the MapReduce job is lesser than the average time taken by Hadoop to execute the same.

9. Conclusions

The challenges and opportunities for efficient parallel data processing in cloud environments have been demonstrated and Nephele, the data processing framework, has been found to exploit the dynamic resource provisioning offered by the IaaS clouds.

Nephele's basic architecture has been described and a performance comparison to the well-established data processing framework Hadoop has been performed. The ability to assign specific virtual machine types to specific tasks of a processing job, as well as the capability to automatically allocate/deallocate virtual machines in the course of a job execution, can help to improve the overall resource utilization. This would be particularly useful in bringing down the execution cost of processes in general, and business related processes in particular. This study makes significant contribution to the growing field of Cloud computing services and explores new opportunities in the field of parallel data processing.

References

- [1] Dikaiakos, M.D. (2009) Cloud Computing: Distributed Internet Computing for IT and Scientific Research. *IEEE Journal of Internet Computing*, **13**, 10-13. <http://dx.doi.org/10.1109/MIC.2009.103>
- [2] Erdogmus, H. (2009) Cloud Computing: Does Nirvana Hide behind Nebula. *IEEE Journal of Software*, **6**, 4-6. <http://dx.doi.org/10.1109/ms.2009.31>
- [3] Martson, S., Li, Z., Bandyopadhyay, S., Ghalsasi, A. and Zhang, J.H. (2009) Cloud Computing: The Business Perspective. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1413545
- [4] Weinhardt, C., et al. (2010) Cloud Computing—A classification, Business Models and Research Directions. *Journal of Business and Information Systems*, **1**, 391-399. <http://dx.doi.org/10.1007/s12599-009-0071-2>
- [5] Mei, L.J. (2008) A Tale of Clouds: Paradigm Comparisons and Research Issues. *IEEE Asia-Pacific Services Computing Conference*, Yilan, 9-12 December 2008, 464-469.
- [6] Jaeger, P.T. (2011) Cloud computing and Information Policy: Computing in a Policy Cloud. *Journal of Information Technology and Politics*, **5**, 269-283. <http://dx.doi.org/10.1080/19331680802425479>
- [7] Vouk, M.A. (2012) Cloud Computing—Issues, Research and Implementations. *Journal of Computing and Information Technology*, **16**, 235-246. <http://dx.doi.org/10.2498/cit.1001391>