ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

EFFICIENT ALLOCATION OF DYNAMIC RESOURCES IN A CLOUD

S.THIRUNAVUKKARASU[1], DR.K.P.KALIYAMURTHIE[2]

Assistant Professor, Dept of IT, Bharath University, Chennai-73[1]

Professor& Head, Dept of IT, Bharath University, Chennai-73[2]

ABSTRACT: In recent years ad hoc parallel data processing has emerged to be one of the killer applications for Infrastructure-as-a-Service (IaaS) clouds. Major Cloud computing companies have started to integrate frameworks for parallel data processing in their product portfolio, making it easy for customers to access these services and to deploy their programs. The opportunities and challenges for efficient parallel data processing in clouds are discussed and present the research project Nephele. It is the first data processing framework to explicitly exploit the dynamic resource allocation offered by today's IaaS clouds for both, task scheduling and execution. Particular tasks of a processing job can be assigned to different types of virtual machines which are automatically instantiated and terminated during the job execution. Based on this new framework, the extended evaluations of MapReduce -inspired processing jobs on an IaaS cloud system is performed and compared the results to the popular data processing framework Hadoop.

KEYWORDS: Many-Task Computing, High- Throughput Computing, Cloud Computing, Map Reducing

I. INTRODUCTION

Today a growing number of companies have to process huge amounts of data in a cost-efficient manner. Classic representatives for these companies are operators of Internet search engines, like Google, Yahoo, or Microsoft. The vast amount of data they have to deal with every day has made traditional database solutions prohibitively expensive. In order to simplify the development of distributed applications on top of such architectures, many of these companies have also built customized data processing frameworks. Examples are Google’s MapReduce [1] or Yahoo!’s Map-Reduce-Merge. They can be classified by terms like high throughput computing (HTC) or many-task computing (MTC), de-pending on the amount of data and the number of tasks involved in the computation.

For companies that only have to process large amounts of data occasionally running their own data center is obviously not an option. Instead, Cloud computing has emerged as a promising approach to rent a large IT infrastructure on a short-term pay-per-usage basis. Operators of so-called 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 centers and only charge them for the period of time the machines are allocated. The VMs are typically offered in different types, each type with its own characteristics (number of CPU cores amount of main memory, etc.) and cost. However, instead of embracing its dynamic resource allocation, current data processing frameworks rather expect the cloud to imitate the static nature of the cluster environments they were originally designed for. E.g., at the moment the types and number of VMs allocated at the beginning of a compute job cannot be changed in the course of processing, although the tasks the job consists of might have

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

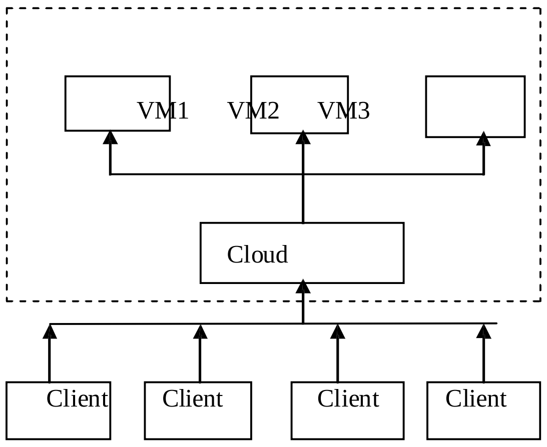
completely different demands on the environment. As a result, rented resources may be inadequate for big parts of the processing job, which may lower the overall processing performance and increase the cost.
In this paper we want to discuss the particular challenges and opportunities for efficient parallel data processing in clouds and present Nephele, a new processing framework explicitly designed for cloud environments. Most notably, Nephele is the first data processing framework to include the possibility of dynamically allocating/deallocating different compute resources from a cloud in its scheduling and during job execution.

II. SCOPE OF THE PAPER

To develop efficient and parallel data processing framework in clouds making it easy for customers to access these services and to give best service data processing framework for cloud environments. Nephele takes up many ideas of previous processing frameworks but refines them to better match the dynamic and opaque nature of a cloud.

3.1 Architecture

Nephele's architecture follows a classic master-worker pattern as illustrated in Fig. 1.



2.1 Problem Statement
Controller
Job Manager

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

The problems like High throughput computing (HTC) or many-task computing (MTC), depending on the amount of data and the number of tasks involved in the computation has to be achieved efficiently. The processing framework then takes care of distributing the program among the available nodes and executes each instance of the program on the appropriate fragment of data.

Infrastructure-as-a-Service(IaaS) clouds, let their customers allocate, access, and control a set of virtual machines (VMs) which run inside their data centers and only charge them for the period of time the machines are allocated. The VMs are typically offered in different types, each type with its own characteristics (number of CPU cores, amount of main memory, etc.) and cost. Allocation of VMs for received instance of job has been a great problem.

III. DESIGN

Based on the challenges and opportunities outlined in the previous section we have designed Nephele, a new Before submitting a Nephele compute job, a user must start a VM in the cloud which runs the so called Job Manager (JM). The Job Manager receives the client's jobs, is responsible for scheduling them, and coordinates their execution. It is capable of communicating with the interface the cloud operator provides to control the instantiation of VMs. We call this interface the Cloud Controller. By means of the Cloud Controller the Job Manager can allocate or deallocate VMs according to the current job execution phase.

The actual execution of tasks which a Nephele job consists of is carried out by a set of instances. Each instance runs a so-called Task Manager (TM). A Task Manager receives one or more tasks from the Job Manager at a time, executes them, and after that informs the Job Manager about their completion or possible errors. Unless a job is submitted to the Job Manager, we expect the set of instances (and hence the set of Task Managers) to be empty. Upon job reception the Job Manager then decides, depending on the job's particular tasks, how many and what type of instances the job should be executed on, and when the respective instances must be allocated/deallocated to ensure a continuous but cost-efficient processing.

3.2 Job Description

Nephele are expressed as a directed acyclic graph (DAG). Each vertex in the graph represents a task of the overall processing job, the graph's edges define the communication flow between these tasks. We also decided to use DAGs to describe processing jobs for two major reasons:

The first reason is that DAGs allow tasks to have multiple input and multiple output edges. This tremendously simplifies the implementation of classic data combining functions. Second and more importantly, though, the DAG's edges explicitly model the communication paths of the processing job. As long as the particular tasks only exchange data through these designated communication edges, Nephele can always keep track of what instance might still require data from what other instances and which instance can potentially be shut down and deallocated.

Defining a Nephele job comprises three mandatory steps: First, the user must write the program code for each task of his processing job or select it from an external library. Second, the task program must be assigned to a vertex. Finally, the vertices must be connected by edges to define the communication paths of the job.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

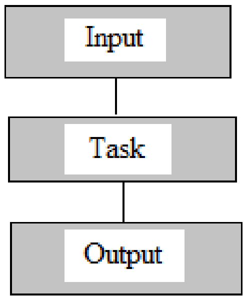


Fig 2. An example of a Job Graph

3.3 Job Processing

After having received a valid Job Graph from the user, Nephele’s Job Manager transforms it into a so-called Execution Graph. An Execution Graph is Nephele’s primary data structure for monitoring the execu- tion of a Nephele job. Unlike the abstract Job Graph, the Execution Graph contains all the concrete information required execute the received job on the cloud.

In order to ensure cost-efficient execution in an IaaS cloud, Nephele allows to allocate/de allocate instances in the course of the processing job, when some subtasks have already been completed or are already running.

IV. PROBLEM DESCRIPTION

4.1 Job Management

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 is submitted from a client to the job manager, each task of the job will be sent to a task manager. The job manager periodically receives heartbeats from the task managers and propagates them to the instance manager.

4.2 Task Execution

The actual execution of tasks in which a job consists of is carried out by a set of instances. Each instance runs a so- called Task Manager. Task manager receives tasks from the job manager and responsible for executing them. After executed them task manager reports the execution result back to the job manager. To supervise the execution of the job, each task manager sends information to the job manager about changes in the execution states of a task.

4.3 Virtual Machine Allocation

Resource allocation is done for each and every received instances of job. Upon job reception the Job Manager then decides, depending on the job’s particular tasks, how many and what type of instances the job should be executed on, and when the respective instances must be allocated/de allocated to ensure a continuous but cost-efficient processing.

Instance manager maintains the set of available compute resources. It is responsible for allocating new compute resources and provisioning available compute resources to the Job Manager. Also it keeps track of the availability of the utilized compute resources in order to report unexpected resource outages. Instance manager receives responses of each task manager. If a task manager has not sent a response, the host is assumed to be dead.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

4.4 Cloud Controlling

The job manager and cloud controller handles heterogeneous instances (compute nodes). Each instance type used for the received job depends on the configuration. When the Task Manager registers with the Job Manager it sends a request arrival rate, which it can handle for the instance.

5. RELATED WORK

In recent years a variety of systems to facilitate MTC has been developed. Although these systems typically share common goals (e.g. to hide issues of parallelism or fault tolerance), they aim at different fields of application.

MapReduce [2] (or the open source version Hadoop) is designed to run data analysis jobs on a large amount of data, which is expected to be stored across a large set of share-nothing commodity servers. MapReduce is highlighted by its simplicity: Once a user has fit his program into the required map and reduce pattern, the execution framework takes care of splitting the job into subtasks, distributing and executing them. A single MapReduce job always consists of a distinct map and reduce program. However, several systems have been introduced to coordinate the execution of a sequence of MapReduce jobs.

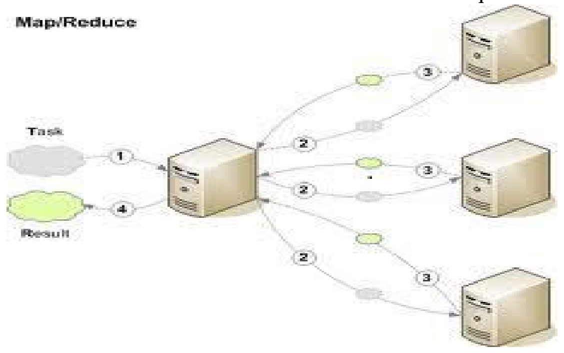


Fig3. MapReduce

MapReduce has been clearly designed for large static clusters. Although it can deal with sporadic node failures, the available compute resources are essentially considered to be a fixed set of homogeneous machines. The Pegasus framework by Deelman et al [3]. has been designed for mapping complex scientific workflows onto grid systems. Similar to Nepehle, Pegasus lets its users describe their jobs as a DAG with vertices representing the tasks to be processed and edges representing the dependencies between them. The created workflows remain abstract until Pegasus creates the mapping between the given tasks and the concrete compute resources available at runtime. The authors incorporate interesting aspects like the scheduling horizon which determines at what point in time a task of the overall processing job should apply for a compute resource. This is related to the stage concept in Nephele.

❑ Google's Map Reduce ❑ map (k1,v1) ❑ (URL, total count)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 5, May 2014

Thao et al.[4] introduced the Swift system to reduce the management issues which occur when a job involving numerous tasks has to be executed on a large, possibly unstructured, set of data. However, it assumes an execution environment which consists of a fixed set of homogeneous worker nodes. The Dryad scheduler is designed to distribute tasks across the available compute nodes in a way that optimizes the throughput of the overall cluster. It does not include the notion of processing cost for particular jobs.

V. CONCLUSION

The challenges and opportunities for efficient parallel data processing in cloud environments and presented Nephele, the first data processing framework to exploit the dynamic resource provisioning offered by today's IaaS clouds. We have described Nephele's basic architecture and presented a performance comparison to the well-established data processing framework Hadoop. The performance evaluation gives a first impression on how the ability to assign specific virtual machine types to specific tasks of a processing job, as well as the possibility to automatically allocate/deallocate virtual machines in the course of a job execution, can help to improve the overall resource utilization and, consequently, reduce the processing cost.

REFERENCES

[1] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In OSDI'04: Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation, pages 10–10, Berkeley, CA, USA, 2004.

[2] Amazon Web Services LLC. Amazon Elastic MapReduce, http://aws.amazon.com/elasticmapreduce/, 2009.

[3]E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G.B. Berriman, J. Good, A. Laity, J.C. Jacob, and D.S. Katz, "Pegasus: A Framework for Mapping Complex Scientific Workflows onto Distributed Systems," Scientific Programming, vol. 13, no. 3, pp. 219-237, 2005.

[4] Y. Zhao, M. Hategan, B. Clifford, I. Foster, G. von Laszewski, V. Nefedova, I. Raicu, T. Stef-Praun, and M. Wilde, "Swift: Fast, Reliable, Loosely Coupled Parallel Computation," Proc. Services, '07 IEEE Congress on, pp. 199-206, July 2007.