ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

IARJSET

International Advanced Research Journal in Science, Engineering and Technology

ICRAESIT - 2015

B V Raju Institute of Technology, Narsapur

Vol. 2, Special Issue 2, December 2015

Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud

Manjula.J[1]

Assistant Professor, CSE, BVRIT, Narsapur, Medak (District), Telangana, India[1]

Abstract: 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. However the processing frameworks which are currently used have been designed for static, homogeneous cluster setups and disregard the particular nature of a cloud. Consequently, the allocated compute resources may be inadequate for big parts of the submitted job and unnecessarily increase processing time and cost. We discuss here the opportunities and challenges for efficient parallel data processing in clouds and present our research project Nephele. 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, we perform extended evaluations of Map Reduce-inspired processing jobs on an IaaS cloud system and compare the results to the popular data processing framework Hadoop.

Keywords: Cloud Computing, Hadoop, Iaas, Amazon, Virtual Machine.

I. INTRODUCTION

Today 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 [5].companies have popularized an architectural paradigm based on a large number of commodity servers. Problems like processing crawled documents or regenerating a web index are split into several independent subtasks, distributed among the available.companies only have to process large amounts of data occasionally running their own data center isobviously not an option. Instead, cloud computing has emerged as a promising approach to rent a large itinfrastructure on a short-term pay-per-usage basis. Operators of so-called iaas clouds, like amazon ec2 [2], 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.in recent years ad hoc parallel data processing has emerged to be one of the killer applications for infrastructure-as-a-service (iaas) clouds. Cloud computing providing unlimited infrastructure to store and execute customer data and program. As customer you do not need to own the infrastructure, they are merely accessing or renting; they can forego capital expenditure and consume resources as a service, paying instead for what they use. 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. However, the processing frameworks which are currently used have been designed for static, homogeneous cluster setups and disregard the particular nature of a cloud. Consequently, the allocated compute resources may be inadequate for big parts of the submitted job and unnecessarily increase processing time and cost. In this paper, we discuss the opportunities and challenges for efficient parallel data processing in clouds and present our research project nephele. Nephele 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, we perform extended evaluations of mapreduce-inspired processing jobs on an iaas cloud system and compare the results to the popular data processing framework hadoop. 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 micros ft. The vast amount of data they have to deal with every day has made traditional database solutions prohibitively expensive. Instead, these companies have popularized an architectural paradigm based on a large number of commodity servers. Problems like processing crawled documents or regenerating a web index are split into several independent subtasks, distributed among the available nodes, and computed in parallel. 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 , microsoft's dryad , or yahoo!'s map-reducemerge[3] . They can be classified by terms like high-throughput computing (htc) or many-task computing (mtc), depending on the amount of data and the number of tasks involved in the

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ICRAESIT - 2015

B V Raju Institute of Technology, Narsapur

Vol. 2, Special Issue 2, December 2015

computation . Although these systems differ in design, their programming models share similar objectives, namely hiding the hassle of parallel programming, fault tolerance ,and execution optimizations from the developer. Developers can typically continue to write sequential programs. 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. 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 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.

II. RESEARCH BACKGROUND

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy & company strength. Once these things are satisfied, then next step is to determine which operating system and language can be used for developing the tool. Once the programmer start building the tool the programmers need lot of external support. This support can be obtained from senior programmer, from book or from website. Before building the system the above consideration are taken into account for developing the proposed system. Authors proposed[1] a parallel data processor centered around a programming model of so called Parallelization Contracts (PACTs) and the scalable parallel execution engine Nephele. The PACT programming model is a generalization of the well-known map/reduce programming model, extending it with further second-order functions, as well as with OutputContracts that give guarantees about the behavior of a function. We describe methods to transform a PACT program into a data flow for Nephele, which executes its sequential building blocks in parallel and deals with communication, synchronization and fault tolerance. Our definition of PACTsallows to apply several types of optimizations on the data flow during the transformation. The system as a whole is designed to be as generic as (and compatible to) map/reduce systems, while overcoming several of their major weaknesses: 1) The functions map and reduce alone are not sufficient to express many data processing tasks both naturally and efficiently. 2) Map/reduce ties a program to a single fixed execution strategy, which is robust but highly suboptimal for many tasks. 3) Map/reduce makes no assumptions about the behavior of the functions. Hence, it offers only very limited optimization opportunities. With a set of examples and experiments, we illustrate how our system is able to naturally represent and efficiently execute several tasks that do not fit the map/reduce model well.

III..NEPHELE FRAMEWORK

NEPHELE is developing a dynamic optical network infrastructure for future scale-out, disaggregated at a center. NEPHELE builds on the enormous capacity of optical links and leverages hybrid optical witching to attain the ideal combination of high bandwidth at reduced cost and power compared to current datacenter networks. The project is performing multidisciplinary research, extending from the datacenter architecture to the overlaying control plane and to the interfaces with the application, in order to deliver a fully functional networking solution. Driven by user needs, NEPHELE's end-to-end development path aims to bridge innovative research with near-market exploitation, achieving transformational impact in datacentre networks that will pave the way to exascale infrastructures. NEPHELE is a research project on optical network technologies, supported by the Horizon2020 Framework Programme for Research and Innovation of the European Commission. The three-year project started officially on February 1, 2015 and brings together seven leading European universities, research centers and companies. The ability to process large numbers of continuous data streams in a near-real-time fashion has become a crucial prerequisite for many scientific and industrial use cases in recent years. While the individual data streams are usually trivial to process, their aggregated data volumes easily exceed the scalability of traditional stream processing systems. At the same time, massively-parallel data processing systems like MapReduce or Dryad currently enjoy a tremendous popularity for data-intensive applications and have proven to scale to large numbers of nodes. Many of these systems also provide streaming capabilities. However, unlike traditional stream processors, these systems have disregarded QoS requirements of prospective stream processing applications so far. In this paper we address this gap. First, we analyze common design principles of today's parallel data processing frameworks and identify those principles that provide degrees of freedom in trading off the QoS goals latency and throughput. Second, we propose a highly distributed scheme which allows these frameworks to detect violations of user-defined QoS constraints and optimize the job execution without manual interaction. As a proof of concept, we implemented our approach for our massively-parallel data processing framework Nephele and evaluated its effectiveness through a comparison with Hadoop Online. For an example streaming application from the multimedia domain running on a cluster of 200 nodes, our approach improves the processing latency by a factor of at least 13 while preserving high data throughput when needed.

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ICRAESIT - 2015

B V Raju Institute of Technology, Narsapur

Vol. 2, Special Issue 2, December 2015

The actual execution of tasks which a Nephele [4, 5] 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. Our current strategies for these decisions are highlighted at the end of this section.

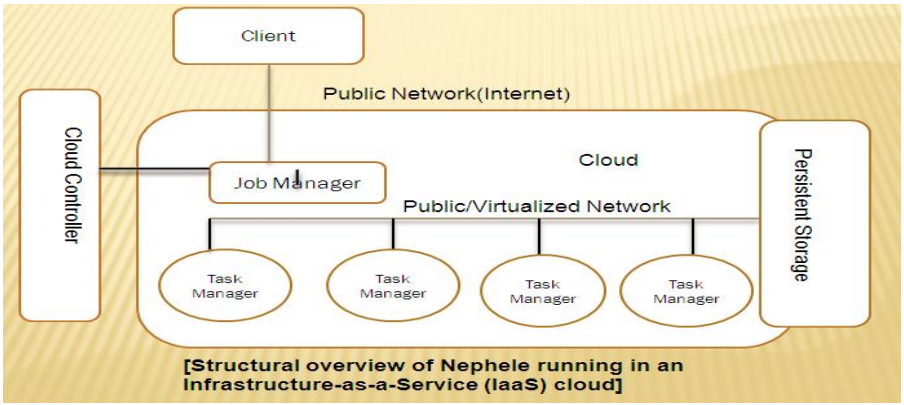

Figure1: Architecture of Nephele

IV. CONCLUSION

In this paper, we have discussed 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. With a framework like Nephele at hand, there are a variety of open research issues, which we plan to address for future work. In general, we think our work represents an important contribution to the growing field of Cloud computing services and points out exciting new opportunities in the field of parallel data processing. For the future we plan to integrate compression for file and network channels as a means to trade CPU against I/O load. Thereby, we hope to mitigate this drawback.

REFERENCES

[1] Nephele/PACTs: A Programming Model and Execution Framework for Web-Scale Analytical Processing, 2010, "Dominic OdejKao, Battré Stephan Volker Markl, Ewen Fabian Hueske Daniel Warneke".
[2] Amazon Web Services LLC, "Amazon Elastic Compute Cloud(Amazon EC2)," http://aws.amazon.com/ec2/, 2009.
[3] Amazon Web Services LLC,"Amazon Elastic MapReduce,"http://aws.amazon.com/elasticmapreduce/, 2009.
[4] D. Battre´, S. Ewen, F. Hueske, O. Kao, V. Markl, and D. Warneke, "Nephele/PACTs: A Programming Model and Execution Framework for Web-Scale Analytical Processing," Proc. ACM Symp. Cloud Computing (SoCC '10), pp. 119-130, 2010.
[5] R. Chaiken, B. Jenkins, P.-A. Larson, B. Ramsey, D. Shakib, S.Weaver, and J. Zhou, "SCOPE: Easy and Efficient Parallel Processing of Massive Data Sets," Proc. Very Large Database Endowment, vol. 1, no. 2, pp. 1265-1276, 2008.

BIOGRAPHY

I, J. Manjula working as Assistant Professor in Department of Computer Science and Engineering at B.V.Raju Institute of Technology, Narsapur, Medak District, Telangana. I did my M.Tech (CSE) in 2012 and B.Tech (CSE) in 2006. My research interests are Computer Networks, Artificial Intelligence, Algorithms, and Cloud Computing.