ViCRA: Vigorous Cloud Resource Allocation for Coextensive Data Processing

Ulya Sabeel
Department of Computer Science and Engineering,
Amity School of Engineering & Technology,
Amity University Gurgaon, Haryana, India

Abstract— Vigorous resource allocation is one of the biggest challenging problems in the area of cloud resource management and in the last few years, has attracted a lot of attention from researchers. The improvised coextensive data processing has emerged as one of the best applications of Infrastructure-as-a-Service (IaaS) Cloud. Current data processing frameworks can be used for static and homogenous cluster setups only. So, the resources that are allocated may be insufficient for larger parts of submitted tasks and increase the processing cost and time unnecessarily. Due to the arcane nature of the cloud, only static allocation of resources is possible rather than dynamic. In this paper, we have proposed a generic coextensive data processing framework (ViCRA) whose working is based upon Nephele architecture that allows vigorous resource allocation for both task scheduling and realization. Different tasks of a processing job can be assigned to different virtual machines which are automatically initiated and halted during the task realization. This framework has been developed in C#. The experimental results profess that the framework is effective for exploiting the vigorous cloud resource allocation for coextensive task scheduling and realization.

Keywords— Cloud Computing, Coextensive Data Processing, Resource Management, Task Scheduling, Vigorous Resource Allocation

Introduction

Cloud Computing is the essence of modern computing systems nowadays. It is the technology that is used to ingress remotely stored data with the help of the internet. It is promptly reinforcing itself as the next step in the augmentation and implementation of large number of distributed applications. Nowadays, the companies like the search engines (Yahoo, Bing, Ask.com, AOL, Microsoft or Google) have to process large amounts of data in a manner that is cost effective. Such a huge chunk of data is very difficult and costly to handle using traditional database systems. In order to deal with these problems, these companies have promoted an architecture based upon commodity servers. These problems are subdivided into independent subtasks, distributed to the available machines and their computations done in parallel [1].

Many companies have come up with new frameworks to handle such data. Some examples are Google’s MapReduce [2], Microsoft’s Dryad [3], or Yahoo!’s Map-Reduce-Merge [4]. All of these frameworks differ in their designs but share common objectives of fault tolerance, parallel programming and optimization. For large companies with large amounts of data, a single centralized database won’t be feasible. Therefore, they take cloud computing as an option and rent Infrastructure on pay-as-you-go basis. The owners of these companies like Amazon EC2 [5], allow the users to allot, ingress, and regulate a set of virtual machines (VMs) which run inside their data centers. Google’s MapReduce framework (Hadoop) [6] uses the above design in its cloud. Amazon has also indulged Hadoop into its infrastructure [7].

All these frameworks take into account only the static nature of cloud instead of the dynamic. For the on-demand resource catering, several authors have presented their views. In [7], the authors have proposed a method to handle peak loads in BPEL using Amazon EC2. In [8], the authors propose a technique for resource abstraction over grid computing and cloud resources for scientific workflows. Both the above techniques use batch-driven workflows instead of data intensive and pipelined workflows which are used by Nephele. Thus, our system adopts the same. In [9], the authors have proposed a dynamic resource allocation method that uses both preemptive as well as non preemptive approach for cloud. In [10], the authors present an organizational model for dynamic resource allocation in workflow systems that uses a policy based normative system.

Nephele is the first data processing framework for vigorous resource allocation that provides both coextensive task scheduling and realization offered by today’s IaaS clouds [11]. The main goal of this paper is to propose a framework which is based upon Nephele architecture and allows vigorous resource allocation for both task scheduling and realization. The paper aims to decrease the overload from main cloud and optimize the cloud performance by dividing all the tasks of the cloud using the cloud storage, job manager and task manager and assigning different jobs to different virtual machines which can be automatically...
instantiated and stopped during task realization thus making
the entire process dynamic.

The rest of the paper has been divided into the following
sections: section I consists of proposed methodology,
section II experimental results, section III conclusion and
future work and finally followed by references.

I. PROPOSED METHODOLOGY

Vigorous resource allocation in the cloud is used to
assign the available resources in a balanced and economic
way dynamically. The Vigorous Resource Scheduler (VRS)
continuously observes the resource utilization and
intelligently allots the available jobs among different virtual
machines based upon their needs. In the existing systems,
the cloud processing frameworks have been designed for
static and homogenous clusters only and do not consider the
dynamic nature of cloud. The allotted resources are
insufficient for larger parts of jobs and can increase
processing time and cost. The largest threat in such systems
is the cloud’s ambiguity with anticipation to capitalize on
the data locality.

The proposed system uses the High Throughput Computing
(HTC) and Many Task Computing (MTC) to hide problems
of coextensive data processing and fault tolerance. The
framework which is based upon ‘Nephele Architecture’
takes care of separating the tasks into subtasks, their
distribution and realization. This allows vigorous resource
allocation for both task scheduling and realization. Different
tasks of a processing job can be assigned to different virtual
machines which are automatically initiated and
halted during the task realization. The comparison of the existing
and proposed systems is given in table I.

<table>
<thead>
<tr>
<th>TABLE I. COMPARISON BETWEEN EXISTING &amp; PROPOSED SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>S.No</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

A. System Architecture

The proposed system architecture is given in figure 1. It
follows the Master-Slave pattern. Initially, the client
approaches with the task to the main cloud. From the main
cloud, the task is handled over to the Job Manager (JM),
which acts as the Master here. The JM processes the jobs
and distributes them to the Task Managers (TM), which acts
as its slaves. The JM also communicates with the cloud
operator interface called Cloud Controller, whose role
together with the JM, is to control the initialization,
processing and halting of Virtual Machines (VMs)
according to the current job realization phase. Task
Manager’s (TMs) role is to receive one or more tasks from
the JM, process them until completion and then inform the
JM about the current status/errors. It is the responsibility of
the JM to receive the results, decide which job and how
many instances of the job should be realized. Accordingly,
the respected tasks are allocated/ deallocated to ensure a
cost effective, dynamic and coextensive data processing.
The diagram for ViCRA framework is shown in figure 1.

![Diagram for ViCRA framework](image)

Figure 1 shows the proposed ViCRA framework that can
be used for vigorous cloud resource allocation for coextensive
data processing in the cloud. The tasks after being
forwarded from the Cloud Controller to the JM, are
scheduled an allocated to the best TMs and finally the
response sent back to Cloud Controller and the Client.

B. ViRCA Algorithm

The algorithm proposed is a cyclic job scheduling algorithm
in which the jobs access the system resources in a cyclic
manner for load balancing and coextensive data processing
in order to accomplish a good quality of service in
multitasking cloud environments. The complete schematic
diagram for ViRCA algorithm is depicted in figure 2.
II. EXPERIMENTAL RESULTS

This section describes the environment, hardware and software requirements that have been used to design the framework ViCRA and to perform the experiments successfully. The system requirements are given in table II.

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System (PC)</td>
<td>Intel Core i3 processor</td>
</tr>
<tr>
<td>2</td>
<td>Operating System</td>
<td>Windows 8</td>
</tr>
<tr>
<td>3</td>
<td>Database</td>
<td>SQL Server 2008</td>
</tr>
<tr>
<td>4</td>
<td>Framework</td>
<td>Microsoft Visual Studio 2010</td>
</tr>
<tr>
<td>5</td>
<td>Coding Language</td>
<td>C#, asp.net</td>
</tr>
</tbody>
</table>

The depiction of GUI for ViCRA Framework is given in figure 3. Figure 4 depicts the registration page for the Job Manager (Admin) or client. Both Job Manager/Client have same interface for registration in the cloud service. Figure 5 depicts the Account created for Job Manager (Admin). Figure 6 shows the Job Manager (Admin) updating the details. Figure 7 depicts Job Manager (Admin) scheduling the tasks to Task Managers. Figure 8 depicts the Job Manager (Admin) handling the requests from the client. Figure 9 depicts the Job Manager (Admin) views the response from the Task Managers. The framework is entirely built in C# and asp.net and tested on a windows 8 platform. The experimental results reveal that the framework is successful and allows vigorous resource allocation for both task scheduling and realization.
In this paper, a Generic Framework named ViCRA has been proposed. ViCRA allows vigorous resource allocation for both task scheduling and realization using a cyclic job scheduling algorithm. The main aim of this paper is to improve the ability to automatically adjust to resource over and under utilization during the job realization. The proposed system removes the drawback of the existing system that is acyclic job execution and improperly used resources. During the functioning of this system, different tasks of a processing job can be assigned to different virtual machines which are automatically initiated and halted during the task realization thus reducing the overall costs. Besides, it can be used to cater to the dynamic nature of the cloud along with the static. Experimental results reveal that the proposed technique is effective for vigorous resource allocation for both task scheduling and realization. In future, the focus will be on proposing more sophisticated techniques that will cater to this problem of automatic adjustment of resources, their initialization, processing and execution.
REFERENCES


