ISSN: 0975-766X
CODEN: IJPTFI
Available Online through *Research Article*
www.ijptonline.com

CHARACTERIZATION AND PROFILING OFSCIENTIFIC WORKFLOWS

Sangeeth S, Srikireddy Sai Kiran Reddy, Viswanathan M*, P.Balakrishnan
School of Computing, SASTRA University, Thanjavur-613401.
Email: thambivv@gmail.com

Received on 06-08-2016 *Accepted on 10-09-2016*

Abstract

Scheduling distributed applications can be challenging in a multi-cloud environment due to the lack of knowledge about the application characteristics. In order to realize a versatile multi-cloud scheduling algorithm, knowledge about the application's runtime behavior over various resources is needed. Besides, not all applications exhibit the same kind of resource consumption pattern at all stages. Thus, looking into the resource consumption pattern, extracting the knowledge and classifying the applications can help in better decision making in a multi-cloud environment. The aim of this project is to create a profiler component that monitors the resource consumption of applications and stores it in a profile database. This profiler monitors the whole workflow, so that any user constructed workflow's execution can be estimated by the use of certain Machine Learning algorithms using which an Analytical Model can be created and the estimation is carried out using the profiled data. Using the profile database, the execution time of an workflow can be estimated based on the resource consumption data of the workflow. In this project, we profile certain scientific applications and monitor their resource consumption pattern on different hardware configurations. After profiling, the applications are classified as being memory-intensive, cpu-intensive etc and using the resource consumption pattern of the application, execution characteristics of a particular workflow can be predicted. This is done with the help of certain machine learning algorithms like k-NN classifier, k-NN regression and SIGAR API is used for extracting resource consumption data.

Keywords: Characterization of applications, Profiling, Resource consumption, scheduling.

Introduction

The significant advancements in several disciplines of science and engineering lead to the evolution of complex applications that require huge amounts of computational power and storage capabilities.These applications pose various scalability challenges in several aspects, like computing power, storage capacity, memory, and network

bandwidth.These challenges are well handled by cloud computing. Multi-Cloud, a class of InterCloud, enables the concurrent usage of resources from multiple cloud providers using third party libraries which is independent from cloud providers. This gives the flexibility to the organizations to mix and match their available resources from virtualized environments, private, and/or public clouds to obtain a suitable resource pairs for their applications.

Cloud systems like Kaavo, Scalr and RightScale are being widely used but, in these systems, the required additional resources are not selected in a best-effort manner during application scaling.This is what we intend to address in our project. Nowadays, researchers need access to different scientific workflows in order to get an idea about the performance of their work. We intend to characterise scientific workflows across different scientific fields based on their resource consumption patterns. This characterization is entirely based on resource consumption pattern of applications that provides information in various dimensions about the various operations that are present in each workflow.

This profiled information includes CPU utilisation, memory and other parameters, based on which we classify the applications as CPU-intensive, memory-intensive and so on.

This classification is done based on profiled data upon which machine learning algorithms are applied. Such a classification helps in prediction of future workflow executions. Besides, the cloud user need not specify the exact hardware configuration details on which his/her application needs to run. This is a necessity as of now, but, our project aims to predict and thus, suggest, the apt configuration for successful execution of such scientific applications.

Related Works

G Juve et al's paper[1] provides a characterization of workflows from six diversescientific applications, including astronomy, bioinformatics, earthquake science, andgravitational-wave physics. Information about the I/O, Memory and other computational characteristics are used to perform this characterization.

This paper[2] specifies how Stampede monitoring infrastructure is integrated with Pegasus Workflow Management System and Triana Workflow Systems to perform real time monitoring and troubleshooting capabilities.

Kepler[3] is a project to develop framework for design, execution and deployment of Scientific Workflows. It is an open source system that allow scientists to design and execute scientific workflows efficiently Grid based technologies to distributed computers.

Pegasus[4] is used to map complex scientific workflows onto distributed resources. This improves application performance by restructuring the workflow which clusters multiple tasks in a work flow into single entities.

Problem Statement

The existing system in multi-cloud environment does not allocate resources based on the resource consumption patterns of workflows. Cloud users have to explicitly state their requirements and specify exact hardware configuration details. Also, the existing system allocates computational resources to these applications only when a request arises. So, we intend to add an intelligence mechanism to this allocation process by characterising workflows which in turn allows a better suited cloud instance to be allocated to these applications.

Our objectives are as follows:

- Build a profiler component for profiling of various scientific applications.
- Used profiled data to study resource consumption patterns.
- Classify applications based on obtained resource consumption patterns.
- Predict executions of future instances of these profiledworkflows using machine learning algorithms.

Proposed Work

Main aim of our project is to collect data about resource consumption of some scientific applications so as to create a profile database which contains information about these applications. Applying machine learning algorithms on this data set produces a general inference about this data set. Generic inference obtained from machine learning algorithms are used to add intelligence to allocation of infrastructural resources to these applications when a request arises in cloud services to execute these applications in cloud. Knowledge about these applications beforehand adds an intelligence to this overall process. Another advantage of this model is that users working in research fields like astronomy,earthquake doesn't necessarily need to have knowledge about system configurations. As our system predicts execution time of an already profiled application, it gives users a fair idea about hardware requirements needed to execute an application within a specific deadline.

The initial setting of our project involves gathering data about resource consumption of an application when it executes. That is, we need a background program that executes simultaneously when an application executes and gather resource consumption data such as memory consumption, processor consumption etc. So to gather data, we have used an open-sourced java API named SIGAR. SIGAR stands for System Information Gatherer and Reporter. A java program was developed to gather resource consumption data of an application. SIGAR API was imported to this java program, thus enabling this java program to use the functionalities of SIGAR API. The information extracted needs to be stored in a database so that it can be used in the subsequent steps of the process which involves application

of machine learning algorithms on this data set. Thus, a database System was needed, so MS-Access was used. Connection between the java program and database was established using JDBC-ODBC driver, which serves as bridge between the java program and database. All the extracted data from the application is stored in this database.

Since our project involves profiling of scientific overflows, it was necessary to install scientific overflows. Since there was a need for these applications to be executed using different configurations, VirtualBox was used and Ubuntu was installed as guest OS. Scientific applications such as Montage, HMMER were installed and these applications were executed using different configurations and resource consumption details of these applications were transferred to the profile database.

The machine learning algorithms used in our project are:

- k-means Clustering
- k-NN Classification
- k-NN Regression

Our project aims to group the data points available in our data set base on their characteristics. Since the data points available in our data set doesn't have a class label (lack of grouping), we need to cluster these data instances. So k-means clustering algorithm is used to cluster these data instances as memory- intensive, processor –intensive etc. The datainstances are clustered using k-means clustering algorithm thus each data instance obtaining a class label. Now the database has profiled data with each data instance containing innate knowledge in the form of class label.

k-NN Classifier is used when a new data instance of a particular application which is already profiled needs to be classified based on the already existing training data. A new data instance is created when a user requests for cloud services. So this data instance is given a class label using k-NN Classifier. Most frequent label of the k nearest neighbours is assigned to new data instance. Closeness of data instances is obtained using Euclidean distance. In this way, new data instances are classified to a particular cluster.

k-NN Regression is used to predict execution time of a new data instance of a particular application which is already profiled. Average of execution time of the k nearest neighbours is computed and assigned to the execution time of the new data instance.

Closeness of data instances is obtained using Euclidean distance. In this way, execution time of a particular application for a particular configuration can be predicted. This information gives user a clear understanding about the infrastructural resources needed to complete execution of an application within a given deadline.

Experimental Setup

NetBeans is a free, open-sourced software which eases the process of software development by providing Integrated Development Environment.The NetBeans IDE is primarily intended for development in Java, but also supports other languages like C/C++, Python etc. It can be executed on different platforms like Windows, Linux,MAC OS etc.

Java is one of the widely used programming languages which can be used in different platforms. Many applications ranging from web applications small gaming applications are built using java. One of the main features of java is the fact that once compiled it can be used executed anywhere across any platform any number of times.Writing in the Java programming language is the primary way to produce an intermediate code that will be used as byte code in a JVM(Java Virtual Machine). Java is an Object-oriented language that is highly robust, secure, architecture-independent, portable and dynamic.

MS Access is a database system provided by Microsoft that allows users to create relational database and perform various types of actions with the database. Database created using MS-Access can be used to integrate with different programming languages thus aiding in the process of application development.

In our project, the resource consumption data obtained from SIGAR API is stored in a database created using MS-Access. This database stores the data which is used by machine learning algorithms in the subsequent steps of the project.

Virtual Box is a hypervisor provided by Oracle which supports the creation and management of guest virtual machines. The guest virtual machines can have Windows, Linux, MAC OS etc or any variants of these operating systems as their OS. These guest machines can be allocated RAM and CPU capability as per the wish of the user. VirtualBox is a software which can be deployed across different platforms like Windows, Linux etc.

VirtualBox users can access multiple guest OSs under a single host OS. Many operations can be performed on guest like starting, pausing and stopping independently within its own virtual machine (VM) without interfering with othervirtual machines. VirtualBox supports both hardware and software virtualization.

In our project, there is a need for the applications to be executed among different RAM and CPU cores configuration as there is a need to collect data consumption across different configurations so as to understand behaviour of these applications. This collected data set will help us to understand the characteristics of these applications.

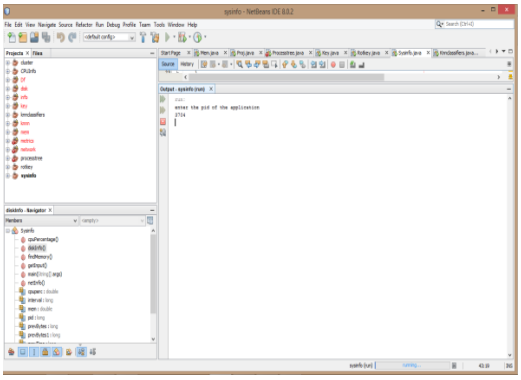
SIGAR API is an open-sourced API used for gathering information related to resource consumption. SIGAR is an acronym for System Information Gatherer and Reporter.

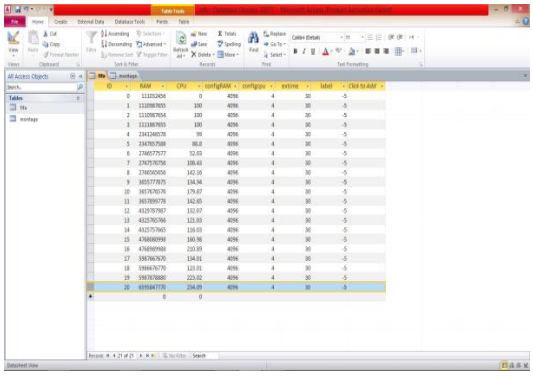
The SIGAR API provides a package for gathering system information such as system memory, CPU, load average etc. Resource consumption pattern is available in all operating systems, but each operating system has its own set of protocols.Using SIGAR, users can access this information regardless of the operating System used.Our project uses this APIto gather data about resource consumption of an application.

We will also require a Laptop/Personal Computer satisfying above mentioned software requirements with a minimum of 4GB RAM.

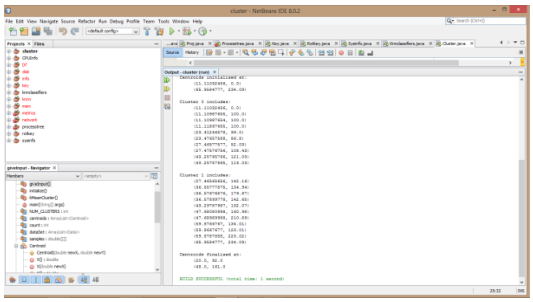
Results and Discussion

The following screenshots consists of output screens of our project's code and snapshots of the database. Database snapshots are used so as to showcase the changes that will be reflected in database when an algorithm uses the data set.

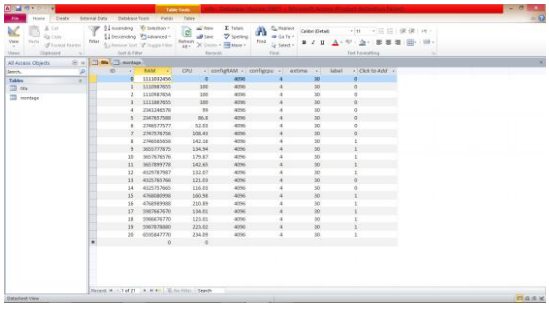




The first screen shot is the output screen of Sysinfo.java. The java code asks PID of a process as input and the resource consumption data that is extracted using SIGAR API which is used in Sysinfo.java is stored into database using JDBC-ODBC connection. Initially all the data instances have a default class label.

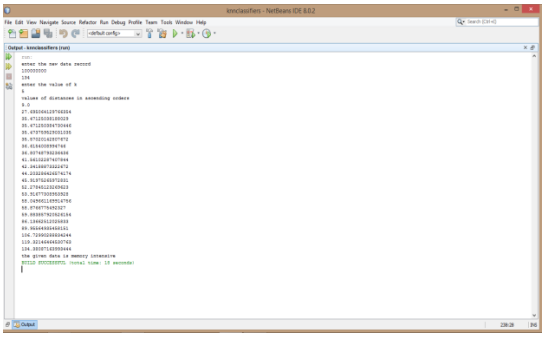


Clustering of data



Extracted data

The resource consumption data that is stored in the database is now given as input to k-mean Clustering algorithm. The k-means clustering algorithm clusters data and now the data instances can be confined to a particular cluster. The output screen shows the clusters that were formed using the data set. Now the data instances in the database are assigned labels attained from k-means clustering algorithm.



k-NN Classification.

Now when a new data instance needs to be classified, the data instance along the already profiled data is passed on to the k-NN Classifier algorithm which classifies the new data instance and updates the new data instance in the profile database. The output screen of Knnclassifiers.java shows distances of new instance from the data available in database in ascending order.

Conclusion and Future Work

To conclude, this project addsan intelligence mechanism to the overall process of allocating infrastructural resources to a scientific application when the request arises from a cloud user for the execution of a particular scientific application. The scientific workflows which are already profiled are stored in a database and when a cloud instance needs to be allocated for execution of an already profiled application, the already profiled data gives an idea about the nature of the application as applications are classified as memory-intensive, processor-intensive and so on. The database containing the profiled data gives users an idea about infrastructural resources that will be required to execute an application within a prescribed user-specified deadline. Thus, this sort of a system will make the whole process of allocating infrastructural resources or cloud instances more optimised thereby resulting in a better system.

In addition to the proposed system, we can also suggest guidelines to developers of scientific applications after studying the variations in application behaviour. Using this information, the developers can implement techniques to overcome this problem. This contributes towards improving the developer's algorithmic efficiency.

References

1. Juve, G., Chervenak, A., Deelman, E., et al."Characterizing and Profiling Scientific Workflows." *University of Southern California Information Sciences Institute*(2013).
2. Vahi, K., Harvey, I., Samak, T., et al."Case Study into Using Common Real-Time Workflow Monitoring Infrastructure for Scientific Workflows" *Journal of Grid Computing*, Vol 11, Issue 3, pp 381-406 (2013).
3. Altintas, I., Berkley, C., Jaeger, E., Jones, M., et al. "Kepler: an extensible systemfor design and execution of scientific workflows." *16th International Conference on Scientific andStatistical Database Management (SSDBM),*pp. 423–424. IEEE Computer Society (2004).
4. Deelman, E., Singh, G., Su, M.-H., Blythe, J., Gil, Y., et al."Pegasus: aframework for mapping complex scientific workflowsonto distributed systems."*Scientific Programming.*13(3)219–237 (2005).

Corresponding Author:

Sangeeth S,

Email: *thambivv@gmail.com*