AUSTRALIAN JOURNAL OF BASIC AND APPLIED SCIENCES

ISSN:1991-8178 EISSN: 2309-8414
Journal home page: www.ajbasweb.com

A Hybrid Framework for Dynamic Resource Allocation in Cloud Infrastructure

[1]M. Dhivya and [2]K. Ambika

[1]*PG Scholar, Dept. of Mobile and Pervasive Computing University College of Engineering, Anna University, BIT Campus, Tiruchirappalli, TamilNadu.*
[2]*Assistant professor, Dept. of CSE, University College of Engineering, Anna University, BIT Campus, Tiruchirappalli, TamilNadu.*

Address For Correspondence:
M. Dhivya, PG Scholar, Dept. of Mobile and Pervasive Computing University College of Engineering, Anna University, BIT Campus, Tiruchirappalli, TamilNadu.
E-mail: dhivyamohan05@gmail.com

ARTICLE INFO

Article history:
Received 26 April 2016
Accepted 21 July 2016
Published 30 July 2016

Keywords:
Resource allocation, Dynamic dataflow, scheduling, high velocity data, runtime adaptation, fault tolerance.

ABSTRACT

Cloud computing helps to distribute data and provide many resources to users**.** The difficulty to handle high velocity data stimulates the necessity for distributed continuous dataflow system. In order to increase scalability the resource should be provided dynamically. Hence we develop a concept "Hybrid framework" in which the resource provisioning is done by selecting the alternate tasks using a hybrid approach and the reliability is achieved by using proactive fault-tolerance scheme. Thus the hybrid approach is employed to manage resource.

INTRODUCTION

Due to the trend of increasing usage of high velocity and huge quantity of data, cloud computing resource management plays a vital role. The key elements of resource management are monitoring, allocation and discovery. Hence need for study of computing resources, size and complexity of the data centers are growing rapidly. At the same time cloud computing infrastructures are becoming more popular. Thus the resources in cloud infrastructures are managed by resource provisioning. There are two kinds of resource provisioning, first is the static resource provisioning based on peak demand is not that cost effective because of poor resource utilization during off-peak periods. Secondly, the autonomic resource provisioning which is attained by dynamic dataflow framework. This paper is concerned with the dynamic dataflow.

Resources on the cloud can be organised by the vendor, and used by the customer. One of the major issue in cloud computing is fault-tolerance. The proposed framework also helps to overcome the fault-tolerance issue by using self-healing technique which increases throughput and reliability.

Fault tolerance computing is one that can continue to appropriately perform its job in the presence of hardware shut down and/or software or to operate adequately in the presence of faults. Fault tolerance bear-on with all the inevitably techniques to enable robustness and dependability. The main welfares of applying fault tolerance in cloud computing include failure recovery, lower cost, improved performance metrics. Robustness leads to the assets to providing of a correct service in an opposing situation arising due to an unreliable system environment. Dependability is related to some Quality of Services attributes provided by the system, it includes the parameters like reliability and availability.

To Cite This Article: M. Dhivya and K. Ambika., **A Hybrid Framework for Dynamic Resource Allocation in Cloud Infrastructure**. ***Aust. J. Basic & Appl. Sci.,*** *10(12): 1-6, 2016*

The application that deals with the high velocity variable data for which resource management should be done dynamically are online advertisement, social networking, real time event processing etc. These application experiences fluctuations in the data rate which has to be managed efficiently. Hence the input rate of the data should be monitored and controlled by the process.

The major factor that influences the handling of high velocity variable data are runtime scalability and fault-tolerance. In this paper, we emphasize on both scalability and fault-tolerance by using hybrid framework. We also focus on provisioning of resource and dynamic proactive fault-tolerance concepts in order to enable scalability and reliability which reduces the cost while achieving the QOS needs an addresses the following issues:

1) Autonomous runtime adaptions in response to fluctuations in both input data rates and cloud resource performance.

2) Due to the undetermined latency and lose control over computing node the reliability is not achieved. Reliability is achieved by using fault-tolerance techniques.

The above issues are rectified by using the "Hybrid framework".

2.Related work:

(C. Ouyang *et al*., 2007) Service composition denotes to the creation of new services by amalgamation of functionality provided by existing ones. This model has gained noteworthy attention in the Web services community and is seen as a support for building service-oriented applications. A number of area-specific languages for service composition have been planned with agreement being formed around a process-oriented language known as WS-BPEL (or BPEL). Simple communication primitives that may be pooled using control ow-constructs expressing sequence, branching, parallelism, synchronisation, etc. in the kernel of BPEL. As a outcome, BPEL process description provide themselves to static ow-based analysis techniques. This report aims at endorsing the feasibility of using Petri nets for static analysis of BPEL processes. We present a widespread and thoroughly defined mapping of BPEL constructs into Petri net structures. This leads to the execution of a tool which operates by interpreting BPEL processes into Petri nets and exploiting existing Petri net analysis techniques.

(L. Neumeyer *et al*., 2010) S4 (Simple Scalable Streaming System) is a distributed stream processing engine stimulated by the MapReduce model This engine is used to resolve real-world problems in the perspective of search applications that use data mining and machine learning algorithms. Existing commercial search engines, such as Google, Bing, and Yahoo!, typically offer organic web results in response to user queries and then complement with textual advertisements that provide profits based on a "cost-per click" billing model.To render the most appropriate ads in an peak position on the page, scientists improve algorithms that dynamically estimate the probability of a click on the ad given the context. The framework may include user preferences, geographic location, prior queries, prior clicks, etc. A foremost search engine may process numerous queries per second, which may contain quite a lot of ads per page. To process user feedback, developed S4, a low latency, scalable stream processing engine. This design is mainly driven by enormous scale applications for data mining and machine learning in a production environment. The S4 design is remarkably flexible and affords itself to run in large clusters built with commodity hardware.

(A.Iosup *et al*., 2011) Clouds have the ability to provide their owners the welfares of an economy of scale and, at the same time, become an another possibility for both the industry and the scientific community to self-owned clusters, grids, and parallel production environments. For this potential to become reality, the first generation of viable clouds needs to be proven to be trustworthy. In this work we evaluate the dependability of cloud services. Towards this end, we scrutinize long-term performance traces from Amazon Web Services and Google App Engine, currently two of the largest commercial clouds in production. We find that the performance of about half of the cloud services we examine exhibits yearly and daily models, but also that most services have periods of especially steady performance. Last, through trace based simulation we evaluate the impact of the inconsistency observed for the studied cloud services on three large-scale applications, job execution in scientific computing, virtual goods trading in social networks, and state management in social gaming. We show that the impact of performance inconsistency depends on the application, and give evidence that performance variability can be an important factor in cloud provider selection.

(Benjamin Satzger *et al*., 2011) ESC, a new stream computing engine. It is proposed for computations with real-time demands, such as online data mining. It proposes a simple programming model in which programs are specified by directed acyclic graphs (DAGs).ESC allows programmers to concentrate on the problem at hand and deals with distribution and fault tolerance. ESC does not provide self-optimization and it does not meet the SLA requirement like responsiveness.

(V. Gulisano *et al*., 2012) StreamCloud, a scalable and elastic stream processing engine for processing large data stream volumes. It uses a novel parallelization technique that splits queries into subqueries that are allocated to reduces the distribution overhead. Its elastic protocols exhibit low intrusiveness, the incoming load is managed by effective adjustment of resources. Elasticity is combined with dynamic load balancing to

minimize the computational resources used. The paper presents the system design, implementation and a thorough evaluation of the scalability and elasticity of the fully implemented system. Current Stream Processing Engines do not scale with the input load due to single-node bottlenecks. Additionally, they are based on static configurations that lead to either under or over-provisioning. Hence the StreamCloud provides a scalable stream processing engine.

(Dhanasekar.P *et al*., 2014) As virtualization expands deeper into the enterprise to include mission-critical and resource-intensive applications. Virtualization vendors may still be touting the potential of putting more virtual machines (VM) on a single physical machine (PM). But more VM on single PM ratios are dangerous in production environments and can cause performance problems or worse, outages. In order to protect the overloaded machine from degrading the performance, they proposed algorithms for high load prediction, resource prediction, and low load prediction.

(Y. Simmhan *et al*., 2014) Applications in cyber-physical systems are increasingly coupled with online instruments to perform long-running, continuous data processing. Such "always on" dataflow applications are dynamic, where they need to change the applications logic and performance at runtime, in response to external operational needs. Floe"is a continuous dataflow framework that is designed to be adaptive for dynamic applications on Cloud infrastructure. It offers advanced dataflow patterns like BSP and MapReduce for flexible and holistic composition of streams and files, and supports dynamic recomposition at runtime with minimal impact on the execution. Adaptive resource allocations trategies allow our framework to effectively use elastic Cloud resources to meet varying data rates. To illustrate the design patterns of 'Floe' by running an integration pipeline and a tweet clustering application from the Smart Power Grids domain on a private Eucalyptus Cloud. The responsiveness of our resource adaptation is validated through simulations for periodic and random workloads.

(Alok Gautam Kumbhare *et al*., 2015) Variability in resource performance shown by clouds greatly affects the application QoS which leads to thriving of autonomic methods of provisioning elastic resources and thereby to support such applications on cloud infrastructure.More control over the dataflow cost and QoS used as alternative in the concept. The concept of "dynamic dataflows" which utilize alternate tasks as additional control over the dataflow's cost and QoS. An optimization problem to represent deployment and runtime resource provisioning that allows us to balance the application's QoS, value, and the resource cost.

Two greedy heuristics, centralized and sharded, based on the variable-sized bin packing algorithm. Dynamic Dataflows is an extension to continuous dataflows by incorporating the concept of dynamic PEs. Dynamic PEs consists of one or more user-defined alternative implementations for the given PE, any one of which may be selected as an active alternate at run-time. This is extended to continuous dataflows where alternate selection is an on-going process at runtime. Genetic Algorithm (GA) is a meta-heuristic used in optimization and combinatorial problems. It facilitates the exploration of a large search space by iteratively evolving a number of candidate solutions towards the global optimum. The solution to the optimization problem requires determining the best alternate to activate for each PE, and the type and number of VMs, and the mapping of data-parallel PE instances to these VMs. shared and centralized heuristics variants that differ in the quanta of information needed and the execution pattern of the scheduler. Thus, an optimization problem to represent initial deployment and runtime heuristics for resource provisioning that allows us to balance the application's QoS, value, and the resource cost.

3.Proposed Work:

In this paper, we proposed a hybrid approach to increase the performance of dynamic dataflow. In addition to this, we had incorporated fault-tolerance in order to reduce the cost and improve throughput which had not been addressed in previous works so far. In this approach, initial deployment is carried out by assigning Genetic algorithm approach and runtime adaption is carried out by Centralized heuristic. The fault-tolerance is achieved by using dynamic proactive technique.

Initial deployment:

The initial deployment is further divided into two steps: alternate selection and Resource allocation. This pharse is implemented by using genetic algorithm.

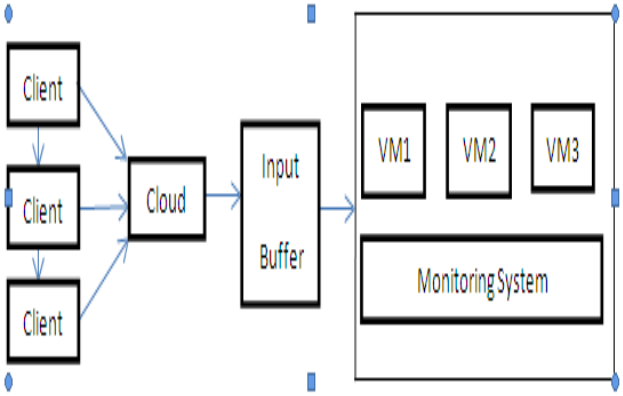

Fig. 1: Proposed System Model

The alternate is selected in terms of chromosomes made up of several genes in GA. The GA initially generates a random population of chomosomes which is the seed to search. Then the genetic operations like crossover and mutation is performed inorder to obtain iterative successive chromosomes. The crossover is done by choosing a pair of parent chromosomes and generating the offspring chromosomes by crossing the genes from parent. Then followed by the mutation operation which alter some parts of the chromosomes and advances the search. Finally the fitness value of each chromosomes is calculated and based on the fitness value the alternate chromosome is selected from the population.

The resource selection is done by using the selection operator in GA

Runtime adaptation:

The runtime adaptation kicks in periodically over the lifetime of the application execution. It considers the current state of the dataflow and cloud resources – available through monitoring – in adapting the alternate and resource selection. The monitoring gives a more accurate estimate of data rates, and hence the resource requirements and its cost.

The runtime adaptation is divided into two steps: alternate selection and resource allocation, Unlike the deployment, both the stages are not run at the same time. Instead, the alternates are selected every m intervals and the resources reallocated every n intervals. The former tries to switch alternates to achieve the throughput constraint given the existing allocated resources, while the latter tries to balance the resources (i.e. provision new VMs or shutdown existing ones) given the alternates that are active at that time. During the alternate selection stage, given the current data rate and resource performance, we first calculate the resources needed for each PE alternate. We then create a list of "feasible" alternates for a given PE, based on whether the current relative throughput is lesser or greater than the expected throughput. Finally, we sort the feasible alternates in decreasing order of the ratio between value to cost, and select the first alternate which can be accommodated using the existing resource allocation.

Fault-tolerance:

Fault tolerance is a major problem to ensure availability and reliability of crucial services as well as application execution. Fault tolerance works as an effectual means to address reliability concerns. Fault tolerance means that system should continue to operate under fault presence. Cloud is exposed to a large number of system failures and the established fault tolerance approaches are less efficient since cloud system's architectural details are not widely available to the users because of the abstraction layers and business model of cloud computing. Fault tolerance that uses the Virtualization Technology (VT) can increase the reliability of applications, but VM migration and consolidations are difficult to achieve.

There are various faults which can occur in cloud computing .Based on fault tolerance policies various fault tolerance techniques can be used that can either be task level or workflow level .

Reactive Fault Tolerance:

Reactive fault tolerance policies reduce the effect of Failures on application execution when the failure effectively occurs. There are various techniques which are based on these policies like Checkpoint/Restart, Replay and Retry and so on.

Proactive Fault Tolerance:

The principle of proactive fault tolerance policies is to avoid recovery from faults, errors and failures by predicting them and proactively replace the suspected components by other working components. Some of the

techniques which are based on these policies are Preemptive Migration, Software Rejuvenation etc.

In this paper, we use dynamic proactive fault-tolerance technique is used to achieve reliability and improve the throughput. In this approach fault tolerance is a proactive mean simply avoid the occurrence of fault when an application or process run or submit inside the VM. It proactively decides that on which VM, which applications are to be run, means the process can be applied on an existing VM or it should be assigned to a fresh newly created VM. These VM will be placed inside a host and host will be placed inside a data center. On which host, which VM will be placed is totally depends on the free resources of host. If the host has much amount of free resources, the VM will be placed on that host as compare to the other host, and after creating host in datacenter and VM in host and after submitting the applications on VMs and after the completion of applications, VMs will be automatically destroyed.

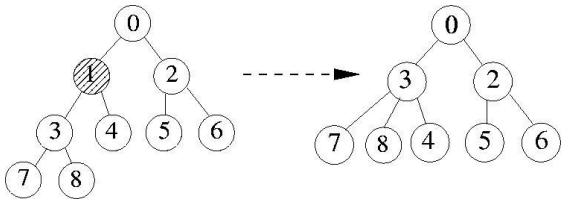

Fig. 2: Rearranging the VM processor when it is destroyed

4.Experimental Setup and Results:

We perform the tests and evaluation in cloudsim simulator. Cloudsim is a new universal and extensible simulation framework that enables continuous modelling, simulation, and experimentation of evolving Cloud computing infrastructures and management services. The simulation framework has the following unique features:

Cloudsim helps for modelling and instantiation of large scale Cloud computing infrastructure, including data centers on a single physical computing node and java virtual machine, a self-contained platform for modelling data centers, service brokers, scheduling, and allocations policies, availability of virtualization engine, which aids in creation and management of multiple, independent, and co-hosted virtualized services on a data center node, flexibility to switch between space-shared and timeshared allocation of processing cores to virtualized services.

These convincing features of CloudSim would speed up the development of new algorithms, methods, and protocols in Cloud computing, hence sponsoring towards quicker growth of the paradigm.The tests were conducted on a i5 machine having configuration: 1.70GHz and 6GB of RAM running a standard windows 7, JDK 1.7 and eclipse juno IDE.

To estimate the overhead in building a simulated Cloud computing environment that consists of a single data center, a broker, three hosts and a user, we executed series of research on this environment and find out the result. In this approach 100 cloudlets are created and they are submitted on the VMs on the basis of the proposed algorithms. The result is in the form of graph and these graph are made on basis of result generated by cloudsim simulator.

The VM load is monitored and when overload occurs the VM resource allocation is done by using GA and Greedy heuristics.

From the result it is stated that this hybrid framework improves the throughput and reliability of the cloud infrastructure when compared to other approaches.

5.Conclusion & Future Work:

In this project, the performance and efficiency of the hybrid framework is improved by adopting hybrid approach which uses GA for initial deployment and the CE greedy heuristic for runtime adaptation is more suitable for optimizing execution of dynamic dataflow framework and effectively controls the high velocity data. The selection of more efficient paths for processing elements had been proposed instead of merely selecting the tasks for processing elements which will control the cost and enhance the QoS with more user sophisticated controls. It also provides reliability by using dynamic proactive fault-tolerance scheme.

REFERENCES

Alok Gautam Kumbhare, Yogesh Simmhan, Marc Frincu, and Viktor K. Prasanna, 2015. “Reactive Resource Provisioning Heuristics for Dynamic Dataflows on Cloud Infrastructure,” in IEEE TRANSACTIONS ON CLOUD COMPUTING, VOLUME.

Simmhan, Y. and A.G. Kumbhare, 2014. “Floe: A continuous dataflow framework for dynamic cloud applications,” CoRR, abs/1406.5977.

Gatti, M., P. Cavalin, S.B. Neto, C. Pinhanez, C. dos Santos, D. Gribel and A.P. Appel, 2014. "Large-scale multi-agent-based modelling and simulation of microblogging-based online social network," in Multi-Agent-Based Simulation XIV. Springer, pp: 17-33.

Fronza, I., A. Sillitti, G. Succi, M. Terho and J. Vlasenko, 2013. "Failure prediction based on log files using random indexing and support vector machines," in Journal of Systems and Software. Elsevier, 86(1): 2-11.

Gulisano, V., R. Jimenez-Peris, M. Patino-Martinez, C. Soriente and P. Valduriez, 2012. "Streamcloud: An elastic and scalable data streaming system," in Transactions on Parallel and Distributed Systems. IEEE, 23(12) 2351-2365.

Iosup, A., N. Yigitbasi and D. Epema, 2011. "On the performance variability of production cloud services," in IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid).

Neumeyer, L., B. Robbins, A. Nair, and A. Kesari, 2010. "S4: Distributed stream computing platform," in IEEE International Conference on Data Mining Workshops (ICDMW).

Ouyang, C., E. Verbeek, W.M. Van Der Aalst, S. Breutel, M. Dumas and A.H. Ter Hofstede, 2007. "Formal semantics and analysis of control flow in ws-bpel," in Science of Computer Programming.Elsevier, 67(2): 162-198.

van Der Aalst, W.M., A.H. Ter Hofstede, B. Kiepuszewski and A.P. Barros, 2003. "Workflow patterns," in Distributed and parallel databases. Springer, 14(1): 5-51.

Braun, T.D., H.J. Siegel, N. Beck, L.L. B¨ ol ¨ oni, M. Maheswaran, A.I. Reuther, J.P. Robertson, M.D. Theys, B. Yao, D. Hensgen and R.F. Freund, 2001. "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems," in Journal of Parallel and Distributed Computing. Elsevier, 61(6): 810-837.

Maheswaran, M., S. Ali, H. J. Siegel, D. Hensgen and R.F. Freund, 1999. "Dynamic mapping of a class of independent tasks onto heterogeneous computing systems," in Journal of Parallel and distributed Computing. Elsevier, 59(2): 107-131.

Goldberg, D.E. and K. Deb, 1990. "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms," in Foundations of Genetic Algorithms, pp: 69-93.

FAGG, G.E., AND J.J. DONGARRA, 2004. Building and using a fault-tolerant MPI implementation. International Journal of High Performance Computing Applications, 18(3): 353-361.

Alim Ul Gias, Rayhanur Rahman, Asif Imran and Kazi Sakib, 2013. International Journal of Web Applications, 5(4).

Sheheryar Malik, Fabrice Huet, 2011. Adaptive Fault Tolerance in Real Time Cloud Computing, IEEE World Congress on services

Ravi Jhawar_, Vincenzo Piuri_, Marco Santambrogio, 2012. A Comprehensive Conceptual System-Level Approach to Fault Tolerance in Cloud Computing, 978-1-4673-07 5/12, *IEEE*.

Dawei Sun, Guiran Chang, Changsheng Mia, Xingwei Wang, 2013. Analyzing, modeling and evaluating dynamic adaptive fault tolerance strategies in cloud computing environment Springer.

Anuj bala, Inderveer Chana. Fault Tolerance-Challenges, Techniques and Implementation in Cloud Computing, IJCSI, 9(1,1).