IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

k-ACO: Information Retrieval from Cloud using Optimum Data Migration based on K-Means & ACO

Vishwas Raval

Assistant Professor, CSE Department, Faculty of Technology & Engineering, The M S University of Baroda

Abstract: Information retrieval is a tricky process, especially when the application contains large amount of data to be processed. This may require computing the indices of the records, processing of the records, placing the records at its proper place based on some hashing techniques and so on. Based on the type of processing required, the retrieval process can also be categorized into the data and compute intensive storage and/or retrieval. Data-intensive and compute intensive systems encompass terabytes to petabytes of data. They require massive storage and intensive computational power in order to execute complex queries and generate timely results. In addition to this, the pace at which the data is growing all over the world, adds fuel into the fire. Retrieving data from such a large volume is like finding a needle from a haystack. This becomes more difficult when the data is stored on clouds and when data is to be migrated from one place to another in order to fulfill a request or provide some service. Such migration requires optimum resource selection from the appropriate neighbor to provide the service to the end user. This paper proposes resource selection technique using K-means Clustering and Ant Colony Optimization for data intensive applications where data is to be migrated from one source to the other.

Keywords: Cloud Computing, VM Allocation, Virtual Machine, CloudSim, Clustering.

1. INTRODUCTION TO CLOUD COMPUTING AND DATA INTENSIVE RETRIEVAL

Cloud computing is, in general, defined as anything that is hosted on Internet and delivered at user's disposal. The rapid growth of the Internet and WWW led to vast amounts of information available online. In addition, business and government organizations create large amounts of both structured and unstructured information which needs to be processed, analyzed, and linked. Data-intensive computing is a class of parallel computing applications which use a data parallel approach to processing large volumes of data typically terabytes or petabytes in size and typically referred to as big data. Computing applications which devote most of their execution time to computational requirements are deemed compute-intensive, whereas computing applications which require large volumes of data and devote most of their processing time to I/O and manipulation of data are deemed data-intensive [1].

Scientific computing involves the construction of mathematical models and numerical solution techniques to solve scientific, social scientific and engineering problems. These models often require a huge number of computing resources to perform large scale experiments or to cut down the computational complexity into a reasonable time frame. These needs have been initially addressed with dedicated high-performance computing (HPC) infrastructures such as clusters or with a pool of networked machines in the same department, managed by some "CPU cycle scavenger" software such as Condor. With the advent of Grid computing new opportunities became available to scientists: in a complete analogy with the power Grid, the computing Grid could offer on demand the horse power required to perform large experiments, by relying on a network of machines, potentially extended all over the world. Computing Grids introduced new capabilities such as dynamic discovery of services, the ability of relying on a larger number of resources belonging to different administrative domains and of finding the best set of machines meeting the requirements of applications. The use of Grids for scientific computing has become so successful that many international projects led to the establishment of world-wide infrastructures available for computational science.

The Open Science Grid originally conceived for facilitating data analysis for the Large Hadron Collider, actually hosts 25000 machines and provides support for data intensive research for different disciplines such as biology, chemistry, particle physics, and geographic information systems. Enabling Grid for E-SciencE (EGEE) is an initiative funded by the European Commission that connects more than 91 institutions in Europe, Asia, and United States of America, to construct the largest multi-science computing Grid infrastructure of the world. TeraGRID is an NSF funded project that provides scientists with a large computing infrastructure built on top of resources at nine resource provider partner sites. It is used by 4000 users at over 200 universities that advance research in molecular bioscience, ocean science, earth science, mathematics, neuroscience, design and manufacturing, and other disciplines. These are only the most representative examples of scientific Grid computing.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

Cloud computing, the current emerging trend in delivering IT services, can address many of the aforementioned problems. By means of virtualization technologies, Cloud computing offers to end users a variety of services covering the entire computing stack, from the hardware to the application level, by charging them on a pay per use basis. Another important feature, from which scientists can benefit, is the ability to scale up and down the computing infrastructure according to the application requirements and the budget of users. By using Cloud based technologies scientists can have easy access to large distributed infrastructures and completely customize their execution environment, thus providing the perfect setup for their experiments. Moreover, by renting the infrastructure on a pay per use basis, they can have immediate access to required resources without any capacity planning and they are free to release them when resources are no longer needed. Cloud computing provides a flexible mechanism for delivering IT services at each level of the computing stack: from the hardware level to the application level. Hardware appliances and applications are provisioned by means of hardware virtualization and software-as-a-service solutions, respectively. This makes the spectrum of options available to scientists wide enough to cover any specific need for their research.

Cloud computing delivers infrastructure, platform, and software that are made available as subscription-based services in a pay-as-you-go model to consumers. These services are referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) in industries. Cloud computing is Internet-based computing. Although many formal definitions have been proposed, NIST provides a somewhat more objective and specific definition here [2].

2. PROBLEM DETAILS AND THE PROPOSED SOLUTION

With the advancement of the Cloud, there are new possibilities opening up on how applications can be built on the Internet. There are large scaled software systems such as social networking sites and e-commerce applications have already been built and have gained popularity enormously. These applications can be benefitted greatly by using cloud services to minimize costs and improve service quality to the end users.

IT giants Google, Yahoo!, Microsoft, IBM own their clouds and the data are being dumped on their clouds. Since cloud is a distributed environment and the data would be scattered across geographic locations, in order to provide a service to the end users, a huge amount of data might have to be migrated from one place to other. This requires the selection of the appropriate host for accessing resources, data etc. and creating a virtual machine (VM) to execute the applications in order to fulfill the request so that the execution becomes more efficient and access cost can be lowered.

This paper proposes the method for the selection of resources i.e. to select one data centre for creating VM and several other data centres for accessing replicas required by the task. The task is submitted to VM. The proposed approach is divided into two phases:

In first phase, it finds a set of data centres for the given task to access all the replicas required, and then it finds an appropriate data centre from the set to create a VM in order to execute the task. Here, the data centre is selected based on the minimum propagation delay from other data centres in the set of all. The idea behind this is to reduce the data transfer time and access cost of data as well by selecting an appropriate set of data centres.

In the second phase, a cost aware resource selection is performed to find a set of data centres such that every data centre, which has been selected, contains the replicas of the data required for reducing transfer times based on propagation delay. This makes sure that the total access cost of these replicas remains as low as possible.

For, both of this case, the K-means clustering approach is used to create a cluster of the VMs based on the type of request and allocate the VM accordingly.

For the execution of data intensive retrieval where data migration is necessary, it requires to find the appropriate data centres where the data is available and then migrate the data at the soruce where the request is generated or create a VM where the data is available so that migration will not take place.

A cloud environment can be considered as a set of M data centres D = {d1, d2, dm}, which are connected by links of different bandwidths. For an application composed of a set of N independent tasks (or jobs) J = {j1, j2, jN} (N >>M), each job j in J, requires a set of K datasets, denoted by Fj, that are accessed on a subset of D.

Consider a task j that has been submitted to a VM, which is created on data centre d, for execution. For each dataset, the time needed to transfer it from df to d is denoted by Tt (f, df, d). The estimated data transfer time for the Vm, Tt (j), is the maximum value of all the times for transferring all the datasets required by the Vm.

Where Rt (df) is the time span from requesting for fd to getting the first byte of f. And In addition, the data access cost C(j) in our research is a function of c(f) , the access cost of each replica f. Here, we consider that each replica, whether on local data centre or on remote data centre.

K-Means follows the partitioned clustering approach. It involves partitioning the given data set into specific number groups called Clusters. Each cluster is associated with a centre point called centroid. Each point is assigned to a cluster with the closest centroid. Proposed dynamic VM allocation algorithm using clustering is as:

Input: List V of Virtual Machine`s with their location around the globe, List D of datacenters

Algorithm:

1. Select K points according to the number of datacenters in D
2. Choose datacenter from D
3. Form K clusters of VM`s from V by assigning closest centroid based on the distance in

DOI 10.17148/IJARCCE.2015.41247

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

terms of propagation delay
4. if delay is beyond the threshold then
- send the request to the cluster where data is available
Else
- Recompute the centroid of each cluster
- Arrange all the requested VM`s in cluster form
- Allocate the VM`s to the available Host
- If all the VM`s are allocated
- Assign the VM`s cluster to the selected datacenter
End if
5. Repeat [2] until D is empty
6. If all the VM`s are created in the datacenters
7. Send the cloudlets to the created VM`s
8. Compute the Cost of execution

The Initial centroid will be chosen by generating graph considering VMs as nodes and finding shortest distance among all using John-Dijksta Algorithm. At later stage, the cluster are formed based on K-means combined with Ant Colony Optimization technique [18][19]. K-Means generates different clusters in different runs. In this, the clusters will be created according to the application.

For the data intensive and large number of data to be transferred then, clusters will be formed according to the distance (propagation and transmission delay) and closest nodes will form a cluster so as to provide faster data transfer. But, as stated in the algorithm, if the delay is beyond a threshold limit then, instead of data migration, the request is sent to the place where data is available so as to execute it at that place only.

3. THE IMPLEMENTATION AND RESULTS

3.1 Implementation in CloudSim

In CloudSim, different classes are there that support the simulation environment for the cloud computing [13][14]. So in order to implement our own policy, it is essential to have knowledge about existing allocation policies and the classes that support these allocation strategies. Since clustering is the new concept in the CloudSim, so some new classes are also created in CloudSim to compute our working policies. We have created two new classes:

Clustering –This class implements the clustering algorithm using K-means and ACO.

DelayMatrix –This class implements John-Dijkstra algorithm

ClusteringTestDemo – This class simulates the algorithm.

CloudSimExample6 – This class implements non-clustered approach.

DataCenter – This class created data centers for simulation and calculates the execution cost.

DatacenterBroker class is the place where the VM allocation policies are carried out. Different functions are there in this class that help to process the virtual machines and their assignment to the datacenters.

3.2 Simulation Results

The test performed this simulation contains five data Centres and five brokers. Data centre contain number of hosts that are connected by high capacity network links with random propagation delays and bandwidth. The numbers of migration requests were 20,40,60,80,100,120. Results of the experiment are shown in the graph. Here, cloudlet refers to the jobs or request for the execution.
Figure 1, 2, 3 and 4 gives performance comparison for three techniques - worst case submission of cloudlets, random submission of cloudlet and proposed algorithm. Figure 1 compares the three techniques for values of makespan (maximum of the completion times of data centres). Clearly the proposed technique performs better than other techniques. This is attributed to the selection of minimum delay datacentres for cloudlet execution. Figure 2 compares the three techniques for flow time parameter (sum of completion times of data centres). Again the proposed technique performs better and the performance improves with increasing number of cloudlets. Figure 3 compares the three techniques for fractional delay parameter - ratio beaten time spent in communication (file transfer plus cloudlet transfer). Again the proposed technique outperforms other techniques. Figure 4 compares the three techniques for average turnaround time (average of the completion times of cloudlets). The proposed technique gives better performance for this metric also.

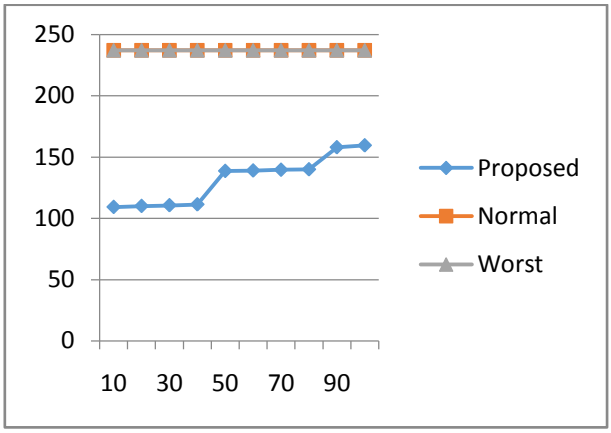

Figure 1 Cloudlet vs. Makespan

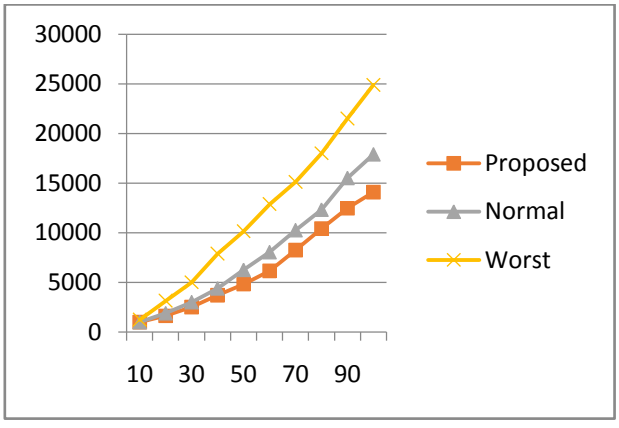

Figure 2 Cloudlet Vs. Flowtime

DOI 10.17148/IJARCCE.2015.41247

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

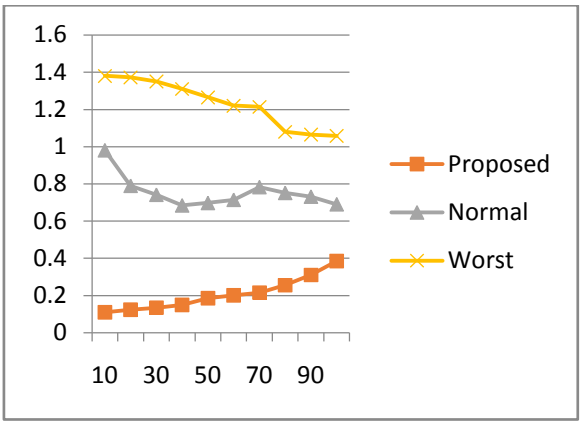

Figure 3 Cloudlet vs. Fractional Delay

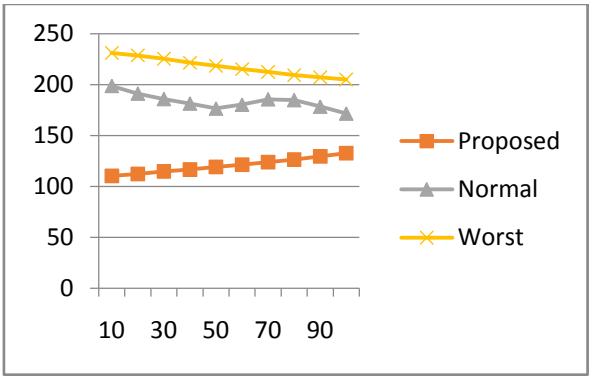

Figure 4 Cloudlet Vs. Avg. Turnaround Time

4. CONCLUSION

This paper proposes a novel cluster based information retrieval technique where the data migration is necessary to serve the request. The method is proved better against the non-clustered approach.

REFERENCES

[1] Handbook of Cloud Computing, "Data-Intensive Technologies for Cloud Computing," by A.M. Middleton. Handbook of Cloud Computing. Springer, 2010

[2] A nist notional definition of cloud computing, http://www.csrc.nist.gov/ groups/ SNS/cloudcomputing.

[3] J. Yu and R. Buyya, "A Novel Architecture for Realizing Grid Workflow using Tuple Spaces," in GRID '04: Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing. Washington, DC, USA: IEEE, 2004, pp. 119–128.

[4] S. Venugopal, R. Buyya, and L. Winton, "A Grid service broker for scheduling e-Science applications on global data Grids," Concurrency and Computation: Practice & Experience, vol. 18, no. 6, pp. 685–699, May 2006.

[5] S. Pandey, W. Voorsluys, M. Rahman, R. Buyya, J. E. Dobson, and K. Chiu, "A grid workflow environment for brain imaging analysis on distributed systems," Concurrency and Computation: Practice & Experience, vol. 21, no. 16, pp. 2118–2139, 2009.

[6] S. Y. Ko, R. Morales, and I. Gupta, "New Worker-Centric Scheduling Strategies for Data- Intensive Grid Applications," in Middleware, ser. Lecture Notes in Computer Science, R. Cerqueira and R. H. Campbell, Eds., vol. 4834. Springer, 2007, pp. 121–142.

[7] T. Glatard, J.Montagnat, D. Lingrand, and X. Pennec, "Flexible and Efficient Workflow Deployment of Data-Intensive Applications on Grids with MOTEUR," International Journal of High Performance Computing Applications, vol. 22, no. 3, pp.347–360, 2008.

[8] V. Bhat, M. Parashar, H. Liu, N. Kandasamy, M. Khandekar, S. Klasky, and S. delwahed, "A self-managing wide-area data streaming service," Cluster Computing, vol. 10, no. 4, pp. 365–383, 2007.

[9] S.M. Park and M. Humphrey, "Data throttling for data-intensive workflows," in IPDPS '08: Proceedings of the 22nd International Parallel and Distributed Processing Symposium. IEEE, 2008, pp. 1–11.

[10] R. Duan, T. Fahringer, R. Prodan, J. Qin, A. V. on, and M. Wieczorek, "Real world workflow applications in the askalon grid environment," in European Grid Conference (EGC 2005), ser. Lecture Notes in Computer Science. Springer Verlag, February 2005.

[11] Chervenak, E. Deelman, M. Livny, M.-H. Su, R. Schuler, S. Bharathi, G. Mehta, and K. Vahi, "Data placement for scientific applications in distributed environments," in GRID '07: Proceedings of the 8th IEEE/ACM International Conference on Grid Computing. Washington, DC, USA: IEEE, 2007, pp. 267–274.

[12] B. Ludascher, I. Altintas, C. Berkley, D.Higgins, E. Jaeger,M. Jones, E. A. Lee, J. Tao, and Y. Zhao, "Scientific workflow management and the Kepler system: Research Articles," Concurrency and Computation: Practice & Experience, vol. 18, no. 10, pp. 1039–1065, 2006.

[13] R Calheiros, R Ranjan, D Rose, R Buyya, "CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services" in Technical Report, GRIDS-TR-2009-1, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, 2009.

[14] W Bhathiya., R Buyya, R Ranjan, "CloudAnalyst: A CloudSim based Visual Modeller for Analysing Cloud Computing Environments and Applications" in 2010 24th IEEE International Conference on Advanced Computer Theory and Engineering (ICACTE), pp. V4-285–V4-288.

[15] S Venugopal, R Buyya, "An SCP-based heuristic approach for scheduling distributed data intensive applications on global grids", Journal of Parallel and Distributed Computing, Vol. 68, No. 4, 2008, pp:471-

[16] R Buyya, R Ranjan, N Rodrigo. Calheiros, "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities", in The 2009 International Conference on High Performance Computing and Simulation, HPCS2009, pp.1-11.

[17] S Pandey, R Buyya; "Scheduling and Management of Data Intensive Application Workflow in Grid and Cloud Computing Environments", PhD Thesis, 2010.

[18] http://en.wikipedia.org/wiki/K-means_clustering

[19] Yucheng Kao, Kevin Cheng; "An ACO-Based Clustering Algorithm", Volume 4150 of the series Lecture Notes in Computer Science pp 340-347