ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

A Best Vigorous Resource Allocation Autonomously In Cloud

S.Rameez Raja[1], S.Rinesh MBA.,M.E.,(Ph.D)[2]

PG Scholar, Department of CSE, Faculty of Engineering, Karpagam University, Coimbatore, Tamil Nadu, India[1]

Assistant Professor, Department of CSE, Faculty of Engineering, Karpagam University, Coimbatore, Tamil Nadu, India[2]

ABSTRACT: This paper describes cloud computing, a computing platform for the next generation of the Internet. The paper defines clouds, explains the business benefits of cloud computing, and outlines cloud architecture and its major components. Readers will discover how a business can use cloud computing to foster innovation and reduce IT costs. Introduction Enterprises strive to reduce computing costs. Many start by consolidating their IT operations and later introducing virtualization technologies. Cloud computing takes these steps to a new level and allows an organization to further reduce costs through improved utilization, reduced administration and infrastructure costs, and faster deployment cycles. The cloud is a next generation platform that provides dynamic resource pools, virtualization, and high availability. Cloud computing describes both a platform and a type of application. A cloud computing platform dynamically provisions, configures, reconfigures, and de provisions servers as needed. Cloud applications are applications that are extended to be accessible through the Internet. These cloud applications use large data centers and powerful servers that host Web applications.

KEYWORDS: Data sets storage, computation-storage trade off, computation- and data-intensive application, cloud computing, DOPS, PSM, Virtual machine

I. INTRODUCTION

The latest emergence of Cloud computing is a significant step towards realizing this utility computing model since it is heavily driven by industry vendors. Cloud computing promises to deliver reliable services through next-generation data centers built on virtualized compute and storage technologies. Users will be able to access applications and data from a "Cloud" anywhere in the world on demand and pay based on what they use.
Most cloud services built on top of a centralized architecture may suffer denial-of-service (DoS) attacks, unexpected outages, and limited pooling of computational resources. On the contrary, volunteer computing systems (or Desktop Grids) can easily aggregate huge potential computing power to tackle grand challenge science problems. In view of this, we propose a novel cloud architecture, namely self-organizing cloud (SOC), which can connect a large number of desktop computers on the Internet by a P2P network. In SOC, each participating computer acts as both a resource provider and a resource consumer. They operate autonomously for locating nodes with more abundant resource or unique services in the network to offload some of their tasks, meanwhile they could construct multiple VM instances for executing tasks submitted from others whenever they have idle resources. We focus on two key issues in the design of SOC:

1) The multiattribute range query problem in a fully decentralized environment for locating a qualified node to satisfy a user task's resource demand with bounded delay

2) To optimize a task's execution time by determining the optimal shares of the multi-attribute resources to allocate to the tasks with various QoS constraints, such as the expected execution time and limited budget.

As a fundamental difference to existing approaches, we formulate such a resource allocation problem to be a convex optimization problem. Given a task with its resource requirements and a budget, we first prove that the optimal resource allocation on a qualified node that can minimize a task's execution time does exist. We further show that it is nontrivial to solve such a convex optimization problem directly via a brute-force strategy and the interior point method. By relaxing the problem definition, we propose an algorithm to optimize the task execution time on a qualified resource node, given its preset budget and tolerable quality of service (QoS). The proposed algorithm involves only $O(R^2)$ adjustment steps, where R denotes the number of resource attributes (or dimensions). We further propose a dynamic optimal proportional-share (DOPS) resource allocation algorithm with $O(R^3)$ complexity, by incorporating the proportional-share model (PSM). The key idea is to dynamically scale the amount of resources at each dimension among running tasks proportional to their demand, such that these tasks could use up the maximum capacity of each resource type at a node.

To locate qualified nodes in the SOC environment, we design a fully decentralized range query protocol, namely pointer-gossiping CAN(PG-CAN), tailored for DOPS. Existing P2P desktop Grids favor CAN-based or Chord-based resource discovery protocols. Every joining node registers its static resource attributes (e.g., CPU architecture, OS version) or maximum capacity on the CAN/Chord overlay, so that other users could find the most matched node within a logarithmic (or sub-linear) number of routing steps. Such a design is feasible for a P2P desktop Grid because the resources of a selected node can only be used exclusively by a single task. However, due to dynamic resource provisioning technologies used in cloud, the frequent resource repartitioning and reallocation (e.g., upon task arrival or completion) make it a challenging problem to locate a node containing a combination of available resources along all the R resource attributes that would satisfy the requirements of a submitted task. The proposed PG-CAN range query protocol in this work aims to find the qualified resources with minimized contention among requesters based on task's demand. It is unique in that for each task, there is only one query message propagated in the network during the entire course of discovery. This is different from most existing multi attribute range query solutions that require to propagate multiple sub-queries along multiple dimensions in parallel.

To mitigate the contention problem due to analogous queries in CAN, our range query protocol proactively diffuses resource indexes overthe network and randomly route query messages among nodes to locate qualified ones that satisfy tasks' minimal demands. To avoid possibly uneven load distribution and abrupt resource overutilization caused by uncoordinated node selection process from autonomous participants, we investigate three node selection policies, namely double-check policy, queuing policy, and extra-virtual-dimension (VD) policy.

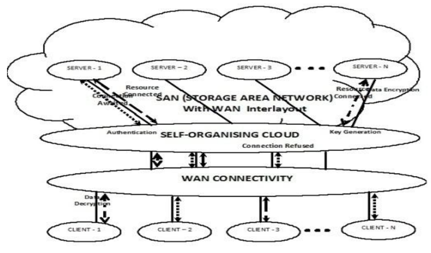


Fig 1: Resource Allocation and Data Transmission Enhancement in Self-Organizing Cloud

II. BACKGROUND REVIEW

Data centres (DCs) are a crucial component of the cloud computing paradigm. They house the computational resources and associated equipment required to provide the services available on the cloud. Some clouds are shared environments

where multiple cloud users utilize the same equipment. Hence, there is potential for both unintentional and malicious service interference between users. The effects of service interference can be seen on current clouds. The throughput of medium instances on Amazon's EC2 can vary by 66% and it has been conjectured, based on anecdotal evidence that the reason for this is a lack of algorithms which manage bandwidth allocation between users. There are numerous mechanisms which can be used for controlling and managing computational, memory and disk re- sources. We
compare and contrast the performance and monetary cost-benefits of clouds for desktop grid applications, ranging in computational size and storage. We address the following questions: (i) What are the performance tradeoffs in using one platform over the other? (ii) What are the specific resource requirements and monetary costs of creating and deploying applications on each platform? (iii) In light of those monetary and performance cost-benefits, how do these platforms compare? (iv) Can cloud computing platforms be used in combination with desktop grids to improve cost-effectiveness even further? The scientific analyses are usually computation intensive, hence taking a long time for execution. Workflow technologies can be facilitated to automate these scientific applications. Accordingly, scientific workflows are typically very complex. They usually have a large number of tasks and need a long time for execution. During the execution, a large volume of new intermediate data will be generated. They could be even larger than the original data and contain some important intermediate results. After the execution of a scientific workflow, some intermediate data may need to be stored for future use.

III. FORECASTING DATA SETS

We need a highly practical cost-effective runtime storage strategy in the cloud, which can solve the following two problems: 1) store the generated application data sets with a cost close to or even the same as the minimum cost benchmark, and 2) take users' (optional) preferences on storage into consideration. We utilize a algorithm, which was used for static on-demand minimum cost benchmarking of data sets storage in the cloud. We enhance the algorithm by incorporating users' (optional) preferences on storage that can offer users some flexibility. Based on the enhanced algorithm, we propose a runtime local optimization- based strategy for storing the generated application data sets in the cloud.

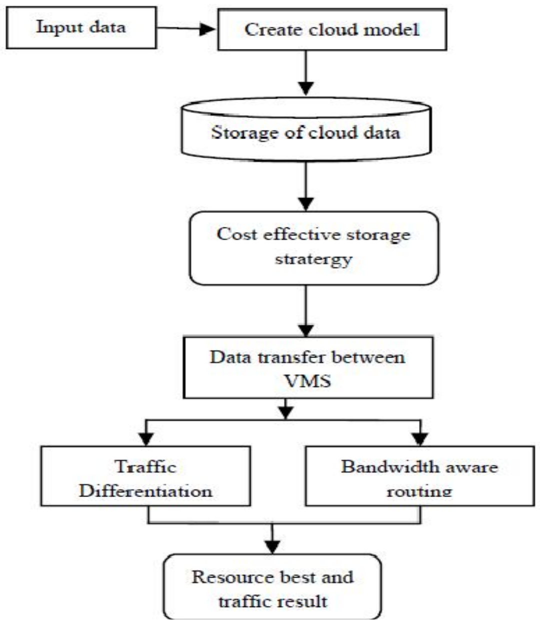


Fig 2: Data set storage strategy

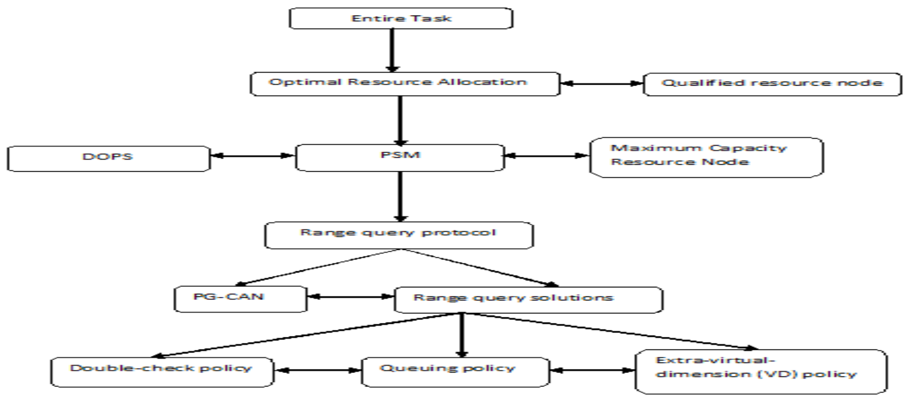


Fig 3: Architecture Design

A. Local optimization

We introduce our local-optimization-based data sets storage strategy, which is designed based on the algorithm. The philosophy is to derive localized minimum costs instead of a global one, aiming at approaching the minimum cost benchmark with highly practical time complexity.

B. Cost transitive tournament

We utilize a Cost algorithm, which was used for static on-demand minimum cost benchmarking of data sets storage in the cloud. We enhance an algorithm by incorporating users' (optional) preferences on storage that can offer users some flexibility. Based on the algorithm, we propose a runtime optimization-based strategy for storing the generated application data sets in the cloud.

C. Data dependency graph

Our DDG is based on data provenance, which depicts the dependency relationships of all the generated data sets in the cloud. With DDG, we can manage where the data sets are derived from and how to regenerate them. It is a acyclic graph based on data provenance in scientific application. Dataset once generated, whether it should be stored or deleted it should be stored in the data dependency graph.

IV. DATA SET COST REDUCTION ALGORITHM

We use the computation cost and storage cost to implement the algorithms . The communication cost also should be included in this, for including this cost the factors such as Jitter, Delay , Path availability and link availability they should find the minimum cost by using cost transitive algorithm , then in a cost effective way they should find the shortest path .

- This algorithm should have less time complexity.
- Should have a good practical runtime computation complexity
- Generation cost-based strategy, in which we store the data sets that incur the highest generation costs.
- Cost rate-based strategy reported in which we store the data sets by comparing their own generation

cost rate and storage cost

```
Algorithm : DATA SET COST REDUCTION Algorithm
Input : Start dataset s , end dataset si
Output : Set of dataset
01 begin
02 genCost = 0;
03 for ( every dataset sj, where si → sj ;
04 create an edge
05 weight =0;
06 genCost = genCost+ dn;
07 weight = weight + (dn +gencost )
08 set si , sj >= weight
09 x = Set of dataset
10 return x
```

V. RELATED WORK

The work mentioned mainly focuses on the comparison of cloud computing systems and the traditional distributed computing paradigms, which shows that applications running in the cloud have cost benefits. They did not touch the issue of computation and storage tradeoff in the cloud.

A. Evaluation setting

A simulation toolkit enables modeling and simulation of Cloud computing systems and application provisioning environments. The CloudSim toolkit supports both system and behaviour modeling of Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. It implements generic application provisioning techniques that can be extended with ease and limited effort. Currently, it supports modeling and simulation of Cloud computing environments consisting of both single and inter-networked clouds (federation of clouds). Moreover, it exposes custom interfaces for implementing policies and provisioning techniques for allocation of VMs under inter-networked Cloud computing scenarios. In this module we are creating cloud users and datacenters and cloud virtual machines as per our requirement

B. Problem analysis

Users can deploy their applications in unified resources without any infrastructure investment, where excessive processing power and storage can be obtained from commercial cloud service providers. With the pay-as-you-go model, the total application cost in the cloud highly depends on the strategy of storing the application data sets, e.g., storing all the generated application data sets in the
cloud may result in a high storage cost, because some data sets may be rarely used but large in size;in contrast, deleting all the generated data sets and regenerating them every time when needed may result in a high computation cost.

C. Overall performance

It is introducing two new parameters that can represent users' preferences and provide users some flexibility in using the storage strategy. The two parameters are denoted as T and λ. T is the parameter used to represent users' tolerance on data accessing delay. Users need to inform the cloud service provider about the data sets that they have requirements on their availabilities. For a data set di, this needs regeneration, Ti is the delay time that users can tolerant when they want to access it.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

λ is the parameter used to adjust the storage strategy when users have extra budget on top of the minimum cost benchmark to store more data sets for reducing the average data sets accessing time. Based on users' extra budget, we can calculate a proper value of λ7, which is between 0 and 1. We multiply every data set di's storage cost rate (i.e., yi) by λ, and use it to compare with di's regeneration cost rate (i.e., genCost(di)*vi) for deciding its storage status. Hence, more data sets tend to be stored, and literally speaking, data sets will be deleted only when their storage cost rates are (1/ λ) times higher than their regeneration cost rates.

$$(\forall d_i, d_j \in DDG \wedge d_i \to d_j) \Rightarrow \exists e{<}d_i, d_j{>}$$

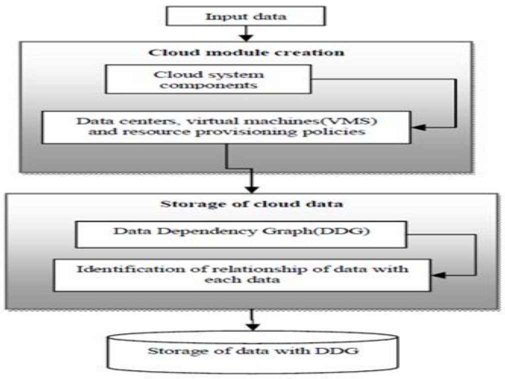
To incorporate the parameter of data accessing delay tolerance

$$e{<}d_i, d_j{>} \to \forall d_k \in DDG \wedge (d_i \to d_k \to d_j) \wedge \left(\frac{genCost(d_k)}{Price_{cpu}} < T_k\right).$$

Design uses two main techniques: traffic differentiation and bandwidth-aware routing. The first allows bulk transfers to exploit spare bandwidth without interfering with existing traffic, while the second achieves efficient use of the spare resources.

Traffic differentiation:

Traffic differentiation is necessary to allow bulk transfers to use left-over bandwidth without affecting best-effort traffic. It separates traffic into best-effort traffic that is delay sensitive and bulk traffic that is delay tolerant. Best-effort traffic is forwarded without any change, while bulk traffic is forwarded with strictly lower priority, i.e., bulk traffic is sent only when there is no best-effort traffic waiting to be sent.


Input data
Cloud module creation
Cloud system components
Data centers, virtual machines(VMS) and resource provisioning policies
Storage of cloud data
Data Dependency Graph(DDG)
Identification of relationship of data with each data
Storage of data with DDG
Fig 4: Dataset storage diagram

Bandwidth-aware routing:

To achieve efficient use of spare resources, transit ISPs would have to modify the default routing within their networks. Intra-domain routing today is not optimized for bandwidth: ISPs do not use all possible paths between a pair of nodes, they do not necessarily
pick the paths with the most available bandwidth, and they do not adapt their paths dynamically as the available bandwidths on the links vary. It addresses these limitations by employing bandwidth-aware routing that is optimized to

pick potentially multiple paths between a pair of nodes to deliver the most data, independent of the paths' latencies. It also periodically recomputes its routes to account for changes in network conditions.

Performance Evaluation:

Compare the existing and proposed system with the following parameters Cost-effectiveness, Efficiency, data transferred computation and data intensive applications in the cloud.

VI. CONCLUSION

With a realistic monetary model, we propose a solution which can optimize the task execution performance based on its assigned resources under the user budget. We prove its optimality using the KKT conditions in the convex-optimization theory.

In order to further make use of the idle resources, we design a dynamic algorithm by combining the above algorithm with PSM and the arrival/completion of new tasks. This can give incentives to users by gaining an extra share of unused resource without more payment. Experiments confirm achieving a super optimal execution efficiency of their tasks is possible. DOPS could get an improvement on system throughput by 15percent 60 percent than the traditional methods used in P2P Grid model, according to the simulation.

In existing we built a prototype supporting live migration of VMs between any two nodes on the Internet (even though they are behind different NATs). In the future, we develop

- A fault-tolerance support for a (DOPS-based, PG-CAN-enabled) SOC system;
- We will also conduct sensitivity analysis of how violation of our model assumptions would impact the optimal resource allocation.

REFERENCES

1. M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “Above the Clouds: A Berkeley View of Cloud Computing,” Technical Report UCB/EECS-2009-28, Feb. 2009.
2. Y. Drougas and V. Kalogeraki, “A Fair Resource Allocation Algorithm for Peer-to-Peer Overlays,” Proc. IEEE INFOCOM ’05, pp. 2853-2858, 2005.
3. M. Feldman, K. Lai, and L. Zhang, “The Proportional-Share Allocation Market for Computational Resources,” IEEE Trans. Parallel and Distributed Systems, vol. 20, no. 8, pp. 1075-1088, Aug. 2009.
4. D. Gupta, L. Cherkasova, R. Gardner, and A. Vahdat, “Enforcing Performance Isolation across Virtual Machines in Xen,” Proc. Seventh ACM/IFIP/USENIX Int’l Conf. Middleware (Middleware ’06), pp. 342-362, 2006.
5. O.H. Ibarra and C.E. Kim, “Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors,” J. ACM, vol. 24, pp. 280-289, Apr. 1977.
6. J.S. Kim et al., “Using Content-Addressable Networks for Load Balancing in Desktop Grids,” Proc. 16th ACM Int’l Symp. High Performance Distributed Computing (HPDC ’07), pp. 189- 198, 2007.
7. W.K. Mark Jelasity and M. van Steen, “Newscast Computing,” Technical report, Vrije Universiteit Amsterdam, 2006.
8. J.E. Smith and R. Nair, Virtual Machines: Versatile Platforms for Systems And Processes. Morgan Kaufmann, 2005.
9. O. Sinnen, Task Scheduling for Parallel Systems, Wiley Series on Parallel and Distributed Computing. Wiley-Interscience, 2007.
10. “The Role of Memory in Vmware Esx Server 3: On Line At: http:// www.vmware.com/pdf/esx3_memory.pdf,”technical report, 2012.
11. C.A. Waldspurger, “Memory Resource Management in Vmware Esx Server,” http://www.usenix.org/events/osdi02/tech/waldspurger.html, 2012.
12. J.P. Walters, V. Chaudhary, M. Cha, S. Guercio Jr., and S. Gallo, “A Comparison of Virtualization Technologies for hpc,” Proc. IEEE 22nd Int’l Conf. Advanced Information Networking and Applications (AINA ’08), pp. 861-868, 2008.