Design and Evaluation of Multiobjective Game Theoretic Scheduling Algorithm for BoT and increase in Parallel Processing of Tasks.

Pramod Vijay Duraphe, Manoj Ravsaheb Bharpure, Sidhharth Sunil Tamboli, Shivram Santosh Gawande

Abstract—**Nowadays, it is interesting issue to schedule large number of BoTs over heterogeneous systems like Hybrid Cloud. If problem is NP-Complete then it is more complex to process in parallel it over that heterogeneous system and get optimal solution. Here, it proposes a Scheduling algorithm which gives a better solution compare to other algorithms. Algorithm is beneficial to compute better results while considering parameters such as, execution time, bandwidth and storage requirement. It gives aggregate result for given problems.In case of NP-complete problem, it is curious issue to schedule large scale parallel computing applications on heterogeneous systems like Hybrid cloud. End user want to meet Quality of Service requirement (QoS).To process huge number of Bag of-Task (BOT) concurrently in such environment with QOS is a Big problem. For that it needs a exact solution. It addresses the scheduling problem of large-scale applications inspired from real-world, divided by a large number of homogeneous and concurrent multiple-of-tasks that are the main sources of bottlenecks but open great potential for accumulation.Here method proposes Multi objective scheduling algorithm to schedule BOTs. Scientific application such as NP Complete problem which takes more time to find the results. Here MOGT Scheduling algorithm proposed in such a way that it optimize the schedule i.e computation time while dividing the Bag of task(BoTs) into Sub Bag of Task(SubBoTs)means it results in increase in parallel computation and find the result in limited number of steps.**

Keywords: Multi-objective scheduling, Bags-of-tasks, Hybrid clouds, NP Complete.

INTRODUCTION

Dissertation Idea

Use of Multiobjective game theoretic Scheduling algorithm for execution of SubBoT takes less time as compared to BoT.Here it consider the objective as computation time while considering constraint as Storage and Bandwidth.

Increase in parallelism after dividation there is increasing parallelism of BoTs in SubBoTs gives faster updates which could be possible by Scheduling algorithm and depend on those update scheduler can take a decision in less time. In this scheduling ,it executes the tasks faster where the processor utilization will be more.

More will be the parallel computation more resource power it will need. For that it requires the hybrid cloud resources to have a better performance for NP Complete application. Computing systems such as clouds have evolved over decade to support various types of scientific applications with dependable, consistent,pervasive and inexpensive access to geographically-seperated high-end computational capabilities. To program such a large and scalable infrastructure like weakly combine-based coordination models of legacy software components such as bags-of-tasks (BoT), hybrid clouds and work flows have emerged as one of the most successful programming paradigms in the scientific community.

From the end user's perspective, minimizing execution time are preferred func-tionalities, whereas from the systems perspective system-level efficiency and [1]fair-ness can be considered as a good motivation such that the applications with large scale computation should be allocated with more resources. Currently, only a few schemes can deal with both perspectives, such as optimizing user objectives makespan while fulfilling other constraints and providing a good efficiency and fairness to all users. While, many applications can generate huge data sets in a relatively short time, such as the Large Hadron Collider produced 5-6 pb of data per year, which must be accommodated and efficiently handled through suitable scheduling bandwidth and storage constraints.

Increase Parallel Computation in Scheduling There are many existing Multi-Objective Evolutionary Algorithms such as Multiobjective Genetic Algorithm (MOGA) and Multi objective Evolutionary Pro-gramming (MOEP)which work on makespan minimization having high complexity as compared to the Multiobjective Game Theoretic Scheduling -MOGTS Algo-rithm.

Here the motivation is that to schedule Bag of Task effectively which can compute the BoT in minimum makespan[6] while considering objectives such as Makespan,Storage and Bandwidth.As a part of more parallelism ,increase in mul-tithreadingwhich can result in faster task execution.

*Assumption: It is to assume that as the parallel computation increase ,it need more hardware resources.

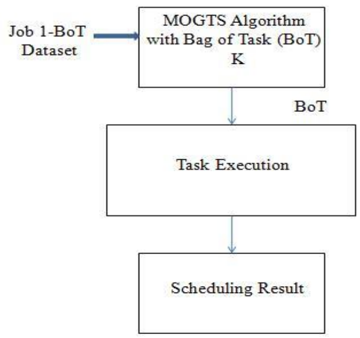

Figure 1.1: BoT System

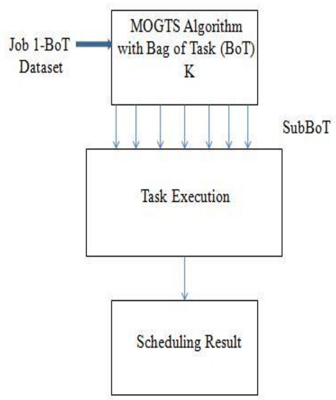

Figure 1.2: Proposed SubBoTSystem

USE OF CLOUD PLATFORM

Previously many scientific applications use Grid Computing services where resources are distributed over network due to that it took much time to aggregate result. To avoid this drawback here it is feasible to use cloud based services where the resources are arranged in one server rack and user can have a remote access.

Here in case , if private resources could not perform better for heavy computation that time there is a need to access paid services.

LITERATURE SURVEY

1.Jia Yu and RajkumarBuyya says,[3]

Over the last few years, Grid technologies have progressed towards a service-oriented paradigm that enables a new way of service provisioning based on utility computing models. Users consume these services based on their QoS (Quality of Service) requirements. In such pay-per-use Grids, work flow execution cost must be considered during scheduling based on user's QoS constraints. In this paper, it propose a budget constraint based scheduling, which minimizes execution time while meeting a specified budget for delivering results. A new type of genetic algorithm is developed to solve the scheduling optimization problem and they test the scheduling algorithm in a simulated Grid testbed.

2. I. Stephie Rachel, Joshua Samuel Raj, V. Vasudevan says,[4]

The application system while executing in a Grid environment may encounter a failure. This phenomenon can be overcome by a reliable scheduler who plays the major role of allocating the applications to the reliable resources based on the reliability requirement of the applications given by the users. The reliability requirement considered in this paper is deadline and budget which is also the quality of service requirement needed for the applications. In this paper, based on deadline and budget as a main factor the tasks are scheduled to the reliable processors.

3.DanielGrosu, Anthony T. Chronopoulos, Ming-Ying Leung says,[5]

In this paper it formulate the static load balancing problem in single class job distributed systems as a cooperative game among computers. It is shown that the Nash Bargaining Solution (NBS) provides a Pareto optimal allocation which is also fair to all jobs. It proposes a cooperative load balancing game and present the structure of the NBS. For this game an algorithm for computing NBS is de-rived. which shows that the fairness index is always 1 using NBS which means that the allocation is fair to all jobs. Finally, the performance of our cooperative load balancing scheme is compared with that of other existing schemes.

4. PreetamGhos says,[6]

This paper proposes cost-optimal job allocation schemes based on a fair pricing strategy for distributed systems where the nodes can have bandwidth constraints and, subsequently, might encounter high communication delays in job transfer. A job allocator receives discrete, serial batch jobs from the users and assigns them to heterogeneous nodes for completion. Todays distributed computing systems incorporate different types of nodes with varied bandwidth constraints which should be considered while designing cost-optimal job allocation schemes for better system performance. This paper proposes a fair pricing strategy for job allocation in bandwidth-constrained distributed systems. The strategy formulates an incomplete information, alternating-offers bargaining game on two variables, such as price per unit resource and percentage of bandwidth allocated, for both single and multiclass jobs at each node. It present a cost-optimal job allocation scheme for single-class jobs that involve communication delay and, hence, the link bandwidth. For fast and adaptive allocation of multiclass jobs, it describes three efficient heuristics and compare them under different network scenarios. The results show that the proposed algorithms are comparable to existing job allocation schemes in terms of the expected system response time over all jobs.

5.L.F. Bittencourt, E.R.M. Madeira, and N.L.S.D. Fonseca says,[7]

Schedulers for cloud computing determine on which processing resource jobs of a work ow should be allocated. In hybrid clouds, jobs can be allocated either on a private cloud or on a public cloud on a pay per use basis. The capacity of the communication channels connecting these two types of resources impact the makespan and the cost of work flows execution. This paper introduces the schedul-ing problem in hybrid clouds presenting the main characteristics to be considered when scheduling work flows, as well as a brief survey of some of the scheduling al-gorithms used in these systems. To assess the influence of communication channels on job allocation, we compare and evaluate the impact of the available bandwidth on the performance of some of the scheduling algorithms.

6.Benoit, A. Marchal, L. Pineau, J.F. Robert, Y. Vivien, F. says,[8]

Scheduling problems are already di cult on traditional parallel machines. They become extremely challenging on heterogeneous clusters, even when embar-rassingly parallel applications are considered.For instance, consider a bag-of-tasks application i.e., an application made of a collection of independent and identical tasks, to be scheduled on a master worker platform.On the contrary,if the plat-form gathers heterogeneous processors, connected to the master via different-speed links, then the previous strategies are likely to fail dramatically. This is because it is crucial to select which resources to enroll before initiating the computation.

In this paper, they still target fully parallel applications, but they introduce a much more complex (and more realistic) framework than scheduling a single application.Theyenvision a situation where users, or clients, submit several bag-of-tasks applications to a heterogeneous master-worker platform, using a classical clientserver model. Applications are submitted online, which means that there is no a priori (static) knowledge of the workload distribution at the beginning of the execution. When several applications are executed simultaneously, they compete for hardware as network and CPU resources.

7.RaduProdan, Simon Ostermann says,[9]

Achieving high-performance in Grid environments is often approached in the community as a pure scheduling problem that maps application components also called services or activities onto distributed resources. For certain objective functions such as execution time, this is an NP-complete optimization problem that fascinates researchers for discovering advanced heuristics that find good solutions. The validation performed is often limited to simulation based on synthetic applications that uses workload information collected from some real Grid traces.

Application Model:

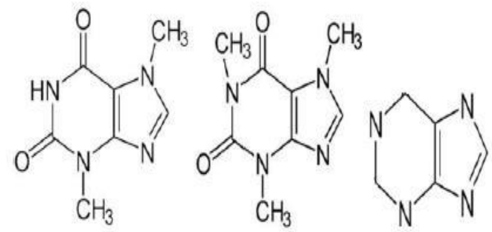

Figure 1.3: RP-GD common pattern search

RP-GD algorithm: RP-GD mines a representative set directly from graph databases. Jianzhong Li, Yong Liu, and Hong Gao adopt the idea of online algorithm to devise RP-GD.Whenever some frequent sub graph mining algorithm generates a frequent sub graph,RP-GD attempt to discover representative R from the current representative set RS such that R can cover P where p in any close frequentsubgraphand R is one of the representative pattern from set of representative patterns RS. When there exists no representative in RS that can cover P, build a new representative Rnew that can cover P using some greedy strategies, and put Rnew that is newly discovered representative into RS. Thus, RP-GD can derive a representative set by scanning closed frequent sub graphs once.

System Architecture

System architecture of our system is as shown in figure 5.3.

How it works?

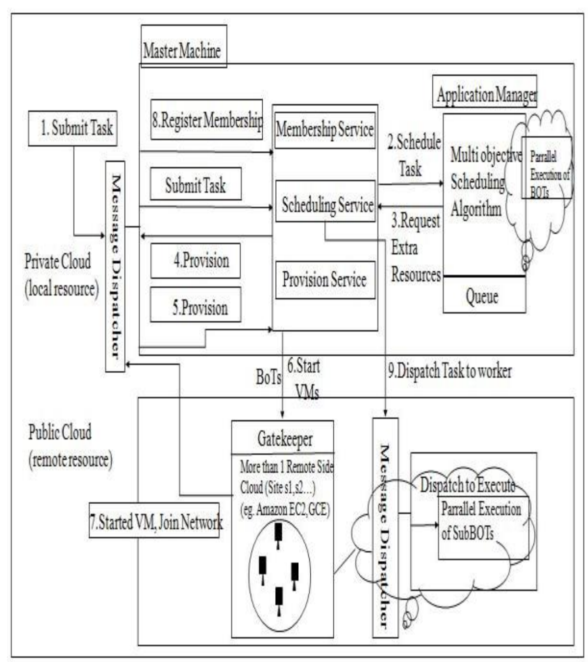

Figure 5.3: Proposed System

1.SubmitBoT through message dispatcher to private cloud.

There services as,

*Membership service
*Scheduling service
*Provision Services

2.Need to schedule BoT so apply MOGTS algorithm.

3.If in case extra resources required for better performance and faster computation then it request for resources to public cloud.

4.Provision.

5.Provision of extra resources is done.

6.Gateway for public cloud "Start VM's".

7.It starts the VM's and join the network.

8.Register the membership.

9.After registration worker node at public cloud are ready to work now.

Here, the concept is that after dividing BoT into SubBoT we need to have powerful resources for better results in parallel computation.This is the only purpose to use Hybrid cloud service.

MATHMATICAL MODULE

In Multi Objective Game Theory Scheduling,
Here it Propose a reliable System which gives a best solution.
It focuses mainly on four Constraints such as,
1. Execution Time
2. Network Bandwidth
3. Storage Requirement

Set Theory:
S=I,O Where I is the input contains
I = (AS, K, M, m_i, ρk_i, σK, ∈)
Require: b_i, br_k
Compute: $\delta^{s(l)}$, $\theta^{s(l)}$
Where O is the output:
O=(Result of NP complete problem)

Here,

- n=Application
- AS=Set of Application
- K=No. OfBoTs
- M=No. of Sited
- mi=No. of Processors on site s_i, $i \in [1, k]$
- ρk_i(m * n) : ETC matrix
- δ_k=Number of Tasks of BoTs $k(k \in [1, m])$
- bl_i=bandwidth limit of site s_i, $(i \in [1,m])$
- $Tk = (k \in [1, m])$
- $\delta^{s(l)}$=Task distribution matrix
- $\theta^{s(l)}$=Resource Allocation Matrix.

Ai=Makespan of application where Ai, $i \in [1,n]$ is maximum completion of its BoTs.

Here,
1] It states that, to minimize Execution time of all application F(x),

Minimize F(x) = (f(x), c(x)),

s.t $h_i(x) \leq \lambda_{x,i}$, i∈[1, M] $g_i(x) \leq sl_i$; i∈[1,M]   x∈S

2] Matrix which delivers the expected execution time ρ_{ki} of task in each BOT

$k \in [1, k]$
$P_{ki} = \{pc_{ki}, pc_{ki}\} \geq poki;$
$P_{ki} = \{pc_{ki} + (po_{ki} - pc_{ki})\} = po_{ki} pc_{ki} < po_{ki}$

3] where,
$Po_{ki} = d_{ki}/b_{ki}$

4] Data bandwidth can be calculated as, $\lambda_x, i \leq \sum_{k=1}^{k} \theta ki * bki = \sum_{k=1}^{k} . \frac{\theta ki * dki}{\rho oki}$

5] The objective of each manager is to minimize execution time while fulfilling storage and bandwidth constraint, expressed as

$$f_k(\Delta) = \frac{\delta k}{\beta k} = \frac{\delta k}{\sum_{i=1}^{M} * \frac{\delta ki}{pki}}$$

$$h_i(\Delta, \beta) = \sum_{k=1}^{k} \frac{\theta ki dki}{pki} \geq \lambda x, i$$

$$g_i(\Delta) = \sum_{k=1}^{k} srk * \theta ki \leq sli$$

6] Resource allocation of BOTs on,

$$S_i, \theta_i = mi * \frac{\delta ki * pki * \omega ki}{\sum_{k=1}^{k} \delta ki * pki * \omega ki}$$

7] Weight factor ω_{ki} for BOTs T_k,

$$\rho \omega ki = \frac{\frac{\min i \in [1,M] Pki}{Pki}}{\sum_{k=1}^{M} \frac{\min i \in [1,M] Pki}{Pki}} = \frac{\frac{1}{pki}}{\sum_{k=1}^{M} pki}$$

$$b\omega ki = \frac{\frac{pki}{dki}}{\sum_{k=1}^{k} \frac{pki}{dki}}$$

$$s\omega ki = \frac{\frac{1}{srki}}{\sum_{k=1}^{k} \frac{1}{srki}}$$

8] Makespan can be expressed as, aggregated execution time divided by number of processors;

argmin($\sum_{k=1}^{k} fk(\Delta)$)

9] For initial step of matrix, BoT considers bandwidth and processors available,

$$\delta ki = \delta k \frac{\frac{mi}{pki}}{\sum_{k=1}^{M} \frac{mi}{pki}}$$

$$b_{ki} = \lambda x, i \frac{dki}{\sum_{k=1}^{M} dki}$$

10] After initial step this matrix will continue,

$\theta^{s(l)} = \theta(\Delta^{s(l-1)}), \Delta^{s(l)} = (\Delta\theta^{s(l)})$

Set Theory:
Morphism

A morphism is a map between two objects in an abstract category.

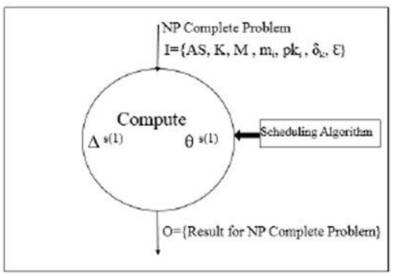

Fig. 5. Set Theory

1) A general morphism is called a homomorphism,
2) A morphism f:Y− >X in a category is a monomorphism if, for any two morphismsu,
v:Z− >Y, fu=fv implies that u=v,
3) A morphism f:Y− >X in a category is an epimorphism if, for any two morphismsu,
v:X− >Z, uf=vf implies u=v,
4) A bijectivemorphism is called an isomorphism (if there is an isomorphism between two objects, then we say they are isomorphic),
5) A surjectivemorphism from an object to itself is called an endomorphism, and
6) An isomorphism between an object and itself is calledan automorphism.

CONCLUSION

In proposed system, it uses Game Theoretic Scheduling algorithm for Sub-BoTs which efficiently schedules the BoTs over a cloud and works for NP Complete problem also achieve good solution for it. It work on all the parameter as makespan ,storage and bandwidth which gives aggregate better results.It gives exact solution for each BoTsto execute on specific site which is most suitable for them.Proposed system which divides BoTsinto a sub-BoTs as per assigned deadlines.As a future scope it can be implemented as a model for Astrological problems like scientific study of Galaxy or Scientific problems in Physics which takes lot of time for computation.

REFERENCES

[1]G. Lovasz, F. Niedermeier, and H. De-Meer, Performance tradeoffs of energy-aware virtual machine consolidation, Journal of Networks Software Tools and Applications, vol. 16, pp. 37-38, 2013.

[2] J. X. Chen, Energy efficient design of cloud data center, Cloud IDC, vol. 57, pp. 481-496, 2011.

[3] Jia Yu and RajkumarBuyya, Deep network and service management for cloud computing and data centers: A report on CNSM 2012, Journal of Network and Systems Management, vol. 21, pp. 707-712, 2013.

[4] I. Stephie Rachel, Joshua Samuel Raj, V. Vasudevan, State-of-the-art research study for green cloud computing, Journal of Supercomputing, vol. 65, pp. 445-468, 2013.

[5] DanielGrosu, Anthony T. Chronopoulos, Ming-Ying Leung, Simple and effective dynamic provisioning for power-proportional data centers, IEEE Transactions on Parallel and Distributed Systems, vol. 24, pp. 1161-1171, 2013.

[6] PreetamGhos, Energy conscious scheduling for distributed computing systems under different operating conditions, IEEE Transactions on Parallel and Distributed Systems, vol. 22, pp. 1374-1381, 2011.

[7] L.F. Bittencourt, E.R.M. Madeira, and N.L.S.D,Fonseca,Power consumption constrained task scheduling using enhanced genetic algorithms, in Evolutionary Based Solutions for Green Computing. Springer Berlin Heidelberg, 2013, pp. 139–159.

[8] Z. Zhou, F. Liu, H. Jin, B. Li, B. Li, and H. Jiang, Onarbitrating the power Performance tradeoff in SaaS clouds, in Proceedings IEEE Infocom, 2013, pp. 872- 880.

[9] Benoit, A. Marchal, L. Pineau, J.F. Robert, Y. Vivien, F. Hybrid meta-heuristic algorithm for job shop scheduling with due date time window and release time, The International Journal of Advanced Manufacturing Technology , vol. 67, pp. 59–71.2013.

[10] RaduProdan, Simon Ostermann, erformance and energy aware cluster-level scheduling of compute-intensive jobs with unknown service times, Simulation Modeling Practice and Theory , vol. 1, pp. 239–250, 2011.

Pramod Vijay Duraphe Computer Engineering, SPP University, Pune, India

Manoj Ravsaheb Bharpure, Computer Engineering, SPP University, Pune, India

Sidhharth Sunil Tamboli, Computer Engineering, SPP University, Pune, India

Shivram Santosh Gawande, Computer Engineering, SPP University, Pune, India