

Resource Allocation in Cloud using Neighbor and Cluster Power Concept

Sridhar P¹, Kavitha B²

Senior Lecturer, Department of Computer Science and Engineering, IRT Polytechnic College
Chennai, Tamilnadu, India

Lecturer, Department of Electronics and Communication Engineering, IRT Polytechnic College
Chennai, Tamilnadu, India

ABSTRACT: In recent times, the resource management in parallel and distributed systems plays a vital role to increase the performance. Resource allocation is a core and challenging problem in a distributed computing systems. Conventional or traditional resource allocation aims to achieve efficient utilization of the resources in the distributed systems even in the presence of communication delay in the network. However these methods still present several challenges in minimizing or reducing the execution time to meet the expanding needs of the applications. Many researchers have tested and exploited many techniques to improve the performance of resource management system. In this paper resource allocation is proposed to be in a dynamic manner. In fixed allocation of the resources the jobs are allotted to the resources directly without considering the resource parameters. In the proposed case, the jobs are allotted and executed dynamically based on the resource metrics. The proposed algorithm select the suitable neighbor resources with minimum transfer delay and maximum computing power to yield reduced completion time of the jobs.

KEYWORDS: Neighbor Resource, Resource Allocation, Computing Power, Completion Time, Transfer Delay

I. INTRODUCTION

Cloud computing is another, developing and arising pattern in the field of parallel and distributed systems. Cloud is a geographically heterogeneous framework. It explores all the resources dynamically to the customers through third party vendors. Clients are more profited by distributed computing through Internet. Highlights of Cloud computing like scalability, reliability and on demand services pulls in the clients in various fields. Different types of clouds such as private cloud, public cloud and hybrid cloud fulfill the prerequisites of the various clients. For the best utilization of cloud resources, proper and negotiable Service Level Agreements are made with the third parties.

Task driven and Data driven computation of applications is a challenging area of recent research. The divisible load execution literature is an active area of current research, influenced by suitable allocation of resources in the network. Now-a-days handling multiple jobs for submission and execution in an effective way is very important for gaining maximum goals. In case of large scale applications, the loads may be partitioned to smaller ones by divisible load paradigm addressed in Cheng and Robertazzi [3]. A formal mathematical framework proposed by Bharadwaj et al. [4] referred as Divisible Load Theory (DLT), addresses effective scheduling and processing of loads in networks. This paradigm utilizes the processor computation capacity and communication delay to find the optimal solution in scheduling and proper processor allocation. Scheduling divisible loads in heterogeneous environment based on computation and communication delay can be realized by multiround algorithm in [5]. Normally the divisible load scheduling algorithm distributes the load to many processors with prior knowledge of network parameters. But for the unknown resources, work in [6] incorporates the adaptive scheduling of loads to the processors using MPI implementation.

In real life applications, many loads will be arriving at many different locations of network. In that case, loads must meet suitable computing facility which may not be available at the required locations of the network. This is a challenging problem in many application domains. In such domains, to meet the computational demands of the

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

applications efficient scheduling and suitable resource allocation must be carried out. The work [1] finds the static scheduling strategy and dynamic scheduling strategy for multi-source divisible loads. Both the strategies show the significant results.

Now-a-days users need a dynamic provisioning of resources according to their requirements. Resources provisioning with high computation power and minimum execution time can be done effectively adopting many approaches. One of the challenges in this area is, to provide suitable and efficient resource allocation for the users to achieve minimum response time.

In this paper we propose an approach for suitable dynamic resource allocation using neighbor concept for successful completion of the jobs in heterogeneous environment. In our approach initially computation power for all resources are calculated and resources having maximum computation power are grouped. Subsequently suitable resources are selected from the group based on minimum transfer delay to improve the performance.

II. RELATED WORK

Parallel and Distributed Computing is an efficient tool for growing needs of computation capacity in emerging scientific and engineering applications [7],[8]. To achieve optimal computing facility, it is essential to search for algorithm which suits efficient scheduling and resource allocation.

Parallel application can be benefited by task parallelism and data parallelism. Many researchers have shown that mixed parallelism (mix of both task and data parallelism) contributes better performance [9],[10],[11]. The algorithm Critical Path Reduction (CPR) [12] and Critical Path and Allocation (CPA) [13] yields better results than existing approach like Two Step Allocation and Scheduling (TSAS) [9]. The decoupled nature of processor allocation followed by scheduling in TSAS and CPA is improved in [2].

With the estimated run time of the task and other parameters, [2] performs the best resource allocation and scheduling in an integrated approach using locality-conscious scheduling algorithm. Resource allocation shall be done based on computing power and minimum transfer delay.

III. PROPOSED STRATEGY –NEIGHBOR SELECTION AND CLUSTER POWER COMPUTATION

Resource allocation is a challenging task in heterogeneous environment for parallel computing. The necessity of searching appropriate computing resources for many applications requires efficient resource management system. Our approach selects the suitable resources by calculating computing power and neighborhood resource. Jobs arriving in the system may be distributed to resources in different ways. The proposed approach deploys the job to the suitable resources having maximum computation power. A suitable resource is fixed as the first resource from the group of resources having maximum computation power. The neighbor resources for the first resource are selected from the group based on minimum transfer delay. The jobs are allotted to the first resource for execution. The jobs will be distributed to the neighbor resource if the total time on the job in the resource exceeds the estimated time of the job.

A. Cluster Power Calculation

The algorithm-1(a) estimates the computation power for each cluster by considering the maximum MIPS and total number of machines. The cluster power is product of the speed (MIPS) and total number of machines in each cluster. The resources with maximum speed and number of machines in each cluster are grouped for further processing. From the selected group the initial resource is chosen arbitrarily. The number of resources selected for job allocation is minimal and also computes the maximum jobs at a reduced completion time.

Algorithm 1 (a). Cluster power Calculation

```
ClusterPower()
{
  For each cluster
  x = maximum MIPS of resource in each Cluster
  y = total no. of machines in each cluster
  // Computing Power
```

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

```
cp = x* y
pg[i] = RES_ID with number of machines in a cluster with their computation rate
done
}
```

B. Neighbor Resource Selection

Algorithm-1(b) calculates the transfer time delay between the initial resource and all available resources. The minimum transfer delay is a random value. The estimated delay is assumed to be 1.5 times of the minimum transfer delay. Transfer time delay is generated for each resource and is compared with the estimated delay. The resource having transfer time delay less than or equal to the estimated delay and also available in the cluster power group are treated as neighbor resources. A minimal number of resources are chosen for processing jobs instead of using all the available resources. The computing power of neighbor resource should be greater than or equal to the computing power of the initial resource.

Algorithm 1 (b). Searching for Neighbor Resource

```
FindNeighbor()
{
    // Calculate the cluster power
    Clusterpower();
    For each resource
        flag = true;
    // random number generator
    ttd = Random();
        if (flag) // minimum transfer delay
            mintd = ttd;
            flag = false;
        else
            if (mintd > ttd )
                mintd = ttd;
            end
        end
    y[i] = mintd;
    done
    edelay = mintd * 1.5;    //eligible delay

    // Neighbor Resource Selection
    For each resource
        if y[i] <= edelay and RES_ID = pg[i]
            select as neighbor resource;
        end
    done
}
```

C. Job Allocation

Selected initial resource and all its neighbor resources are assumed to be under no load condition. Each job is submitted with its estimated time for completion. The jobs are allotted to the initial resource for processing. The completion time of job is calculated and compared with estimated time of each job for submission. If the estimated time is greater than the calculated time (completion time), the job is assigned to that resource. If not, then the job is assigned to the neighbor resource.

In case of arrival of more jobs, each job is verified with the initial resource for execution before assigning it to the neighbor resource. The jobs are then directed to the neighbor resource if the total time of the job to be executed exceeds

International Journal of Innovative Research in Science, Engineering and Technology

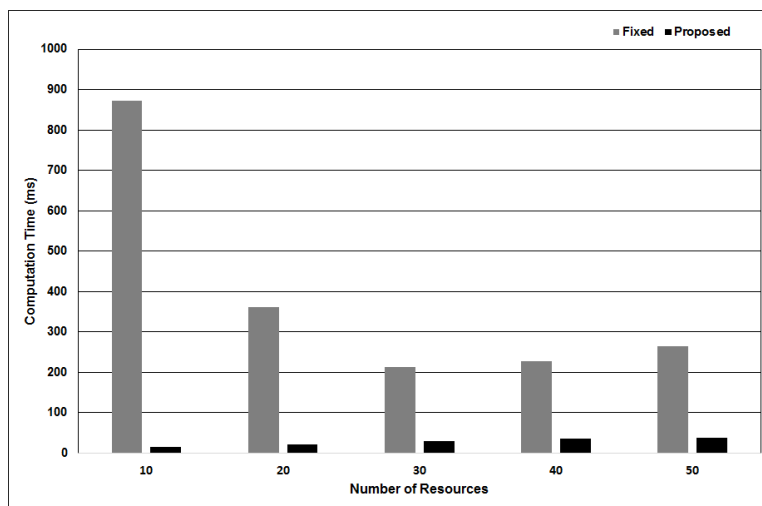
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

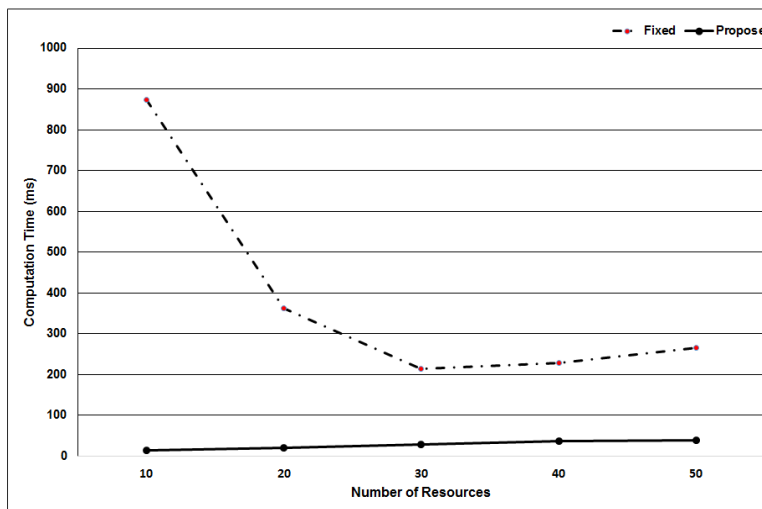
the estimated time of the job. Jobs are further allotted to the next neighbor resource if the same condition occurs in the first neighbor.

IV. PERFORMANCE ANALYSIS

Simulation is carried using CloudSim 2.1 tool kit. The jobs are allotted to the resources in a fixed manner without considering any resource parameters. Simulation is carried by varying the number of resources for a set of 50 jobs of varying size. Number of resources is varied from 10 to 40 with same set of jobs. Simulation is also performed for the proposed approach with the same set of jobs and resources. The comparison is made by considering the completion time of jobs. The completion time of jobs using the proposed algorithm is very less when compared to the fixed scheduling. From the Fig. 1 (a) and Fig. 1 (b) it is acknowledged that the proposed algorithm is efficient.



(a)



(b)

Fig.1. (a) and Fig 1 (b) Comparison of Proposed with Fixed Resource Allocation

V. CONCLUSION

Dynamic provisioning of resources is a vital role for on-demand applications in Cloud. Resources are allotted considering computation power along with communication delay. In our approach the resources are grouped initially

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

based on maximum computation power and suitable neighbor resources are selected from the group considering minimum transfer delay, thus reducing the number of resources required for processing. This approach allows the cloud to use minimum resources so that the remaining resources are better utilized for other applications or processes.

This paper addresses the strategy of selecting minimum resources by computing the power of clusters and also selection of neighbor resources for processing. The simulation results convey the performance improvement by the proposed algorithm. The proposed approach can be extended with cost as a constraint.

REFERENCES

1. Jingxi Jia, Bharadwaj Veeravalli and Jon Weissman, "Scheduling Multisource Divisible Loads on Arbitrary Networks", IEEE Trans. Parallel and Distributed Systems, Vol. 21, No. 4, pp 520-531, 2010
2. Naga Vydyanathan, Sriram Krishnamoorthy, Gerald M. Sabin, Umit V. Catalyurek, Tasin Kurc, Ponnusamy Sadayappan and Joel H. Saltz, "Integrated Approach to Locality-Conscious Processor Allocation and Scheduling of Mixed-Parallel Applications", IEEE Trans. Parallel and Distributed Systems, Vol. 20, No. 8, pp 1158-1171, 2009
3. Y.C Cheng and T.G. Robertazzi, "Distributed Computation with Communication Delays", IEEE Trans. Aerospace and Electronic Systems Vol. 24. No. 6, pp. 700-712 1988.
4. Bharadwaj, D. Ghose, V.Mani and T.G. Robertazzi, "Scheduling Divisible Loads in Parallel and Distributed Systems", CS Press, 1996
5. Y. Yang, K. van der Raadt and H. Casanova, "Multi-round Algorithms for Scheduling Divisible Loads", IEEE Trans. Parallel and Distributed Systems, Vol. 16, No. 11, pp 1092-1102, 2005
6. D. Ghose, H.J. Kim and T.H. Kim, "Adaptive Divisible Load Scheduling Strategies for Workstation Clusters with Unknown Network Resources", IEEE Trans. Parallel and Distributed Systems, Vol. 16, No. 10, pp 897-997, 2005
7. V. Kumar, A. Grama, A. Gupta and G. Karypis, "Introduction to Parallel Computing: Design and Analysis of Algorithms", Benjamin-Cummings, 1994
8. Michael J. Quinn, "Parallel Computing – theory and practice", McGRAW-HILL, 1994
9. S. Ramaswamy, S. Sapatnekar and P. Banerjee, "A framework for exploiting task and data parallelism on distributed memory multicomputers", IEEE Trans. Parallel and Distributed Systems, Vol. 8, No. 11, 1098 - 1116, 1997
10. S. Chakrabarti, J. Demmel and K. Yelick, "Modeling the Benefits of Mixed Data and Task Parallelism", Proceedings of the seventh annual ACM symposium on Parallel algorithms and architectures, (SPAA'95) pp. 74-83, 1995
11. Saniya Ben Hassen, Henri E. Bal, and Cerial J.H. Jacobs, "A Task and Data Parallel Programming Language based on Shared Objects", ACM Transactions on Programming Languages and Systems (TOPLAS), VOL. 20 No. 6, pp1131-1170, 1998
12. C. Nicolescu and P.P.Jonker, "Cpr: Mixed Task And Data Parallel Scheduling For Distributed Systems", Proceedings 15th International, Parallel And Distributed Processing Symposium., Pp. 31, 2001
13. A. Radulescu and A.J.C Van Gemund, "A Low-Cost Approach Towards Mixed Task And Data Parallel Scheduling", International Conference On Parallel Processing, 2001., Pp 69-76, 2001
14. Rodrigo N. Calheiros, Rajiv Ranjan, César A. F. De Rose and Rajkumar Buyya, "Cloudsim: A Novel Framework For Modeling And Simulation Of Cloud Computing Infrastructures And Services", 2009