Proceedings of the Peradeniya University Research Sessions, Sri Lanka, Vol. 15, 16th December 2010

TOWARD THE OPTIMIZATION OF MASTER WORKER APPROACH IN GRID COMPUTING

H. R. O. E. Dhayarathne and S. R. Kodituwakku

Department of Statistics & Computer Science, Faculty of Science, University of Peradeniya

Introduction

Advances in hardware and software technologies lead to the widespread use of loosely coupled, powerful and low-cost networked components known as clusters. Grids are examples of such clusters and are widely used in achieving complex computations. There are several paradigms which increase the effectiveness of Grid computing. The Master-Worker (MW) paradigm (Cesar *et. al.,* 2006) can be nominated as the most common natural programming method for many Grid applications today. In the MW model, one node acts as the controlling master and it has a pool of work that needs to be processed. It simply assigns pieces of work to workers. This is continued until the pool of work is completed. In general, it allocates as many workers as possible to reduce the total execution time and workers are not released until the computation is completed. This strategy does not work well in all situations. Therefore, several approaches have been proposed to solve the Master-Worker allocation problem. For example, the work reported in (Pieczykolan *et. al,* 2007.) and (Morajko *et. al.,* 2005) address the efficiency issue in allocating master and workers. However the efficiency and effectiveness of master-worker allocation is still an open research problem. Our objective is to improve the Master-Worker allocation for achieving high efficiency by keeping the allocated workers usefully busy. Therefore we propose an extension to the generic Master-Worker architecture. An optimization layer in between the Grid service layer and Grid application layer is introduced in order to allocate the Master and Workers in optimal manner.

Materials and Methods

The knowledge about the nodes which are involved in Grid Applications is stored in an organized manner so that it can be used to make decision regarding the Master-Worker allocation. In order to do so an additional layer in between Grid Service Layer and Grid Application Layer is incorporated.

Conceptual Architecture of the Optimization Layer

Architecture of the Conceptual Optimization Layer is depicted in Fig 1. Components of this layer are described here.

Rule Base (Intelligent Agent) stores the information about Master-Worker relationships and updates and keeps all the records of eligible Masters, Master-Worker relationship strengths. **Process Controller** is responsible for creating and maintaining processes corresponding to applications those are executed in the optimization layer.

<table>
<tr><td colspan="3">Optimization Layer</td></tr>
<tr><td rowspan="3">Rule Base
(Intelligent Agent)</td><td>Process Controller</td><td>LOG</td></tr>
<tr><td>Statistical Processor</td><td>Transaction
Transmitter</td></tr>
<tr><td>Transaction Analyzer
& Optimizer</td><td>Job Controller</td></tr>
<tr><td colspan="3">Optimization – Grid Service Interface</td></tr>
</table>

Fig 1. Architecture of the Conceptual Optimization Layer

Transaction Analyzer and Optimizer: involves in deciding how the procedure or procedures in a transaction must be sent to the Grid for better efficiency. Results of the jobs and their respective tasks and transactions submitted to the Grid are recorded in the **LOG**. **Job Controller** submits the jobs which are obtained by analyzer and optimizer to the Grid service layer in order to be executed as parallel jobs. **Transaction Transmitter** would verify at processes controller which transactions are recently arrived from processes. **Statistical Processor holds** the responsibility of extracting relevant statistical information from the applications to utilize the resource allocations.

Statistical approaches in the optimization layer

When considering the statistical constituent of the optimization layer the basic components that deal with it are the Rule Base, the Statistical Processor, and the Transaction Analyzer and Optimizer. Several statistical methods can be adapted and utilized in the optimization layer as statistical approaches. The Set Cover Problem can be seen as a major approach to select the optimal number of subsets those can cover a given set as the objective is to minimize the number of subsets being selected.

Results and Discussion

In the proposed approach, the information about work stations is stored as a knowledge base to create this resource sets concept. So this works as an intelligent agent in allocating Masters and assigning Workers relevantly. If the optimal number of masters is k to cover the given resource set with n points, then it can easily be shown that the time complexity to solve this problem is O(n) and the space complexity is O(1) when the space to store the sets is neglected. Otherwise it is O(n). Generally the maximum complexities of time and space would be O(n) when the set cover problem is concerned. If the total proposed masters are r and the selected masters are k, there is $1/P_k^r$ probability for a selected permutation is an optimal solution.

Let C* be the size of any optimal solution, and G denotes the size of the solution reported by the greedy strategy. Let C== |G|. Then the following bounds can be obtained:

$$C^* \leq C \leq C^*(1 + \log n)$$

Mathematically the greedy approximation algorithm achieves an approximation ratio of H(s), where s is the size of the largest set and H(n) is the nth harmonic number defined as, $H(n) = \sum_{k=1} \frac{1}{k} \leq \ln n + 1$. When the Universe consists of $n = 2^{(i+1)} - 2$ elements the approximation ratio is $\log_2(n)/2$.

Suppose that there are m sets covering all fractions. After t choices, Greedy has at most $(1-1/m)^t$ fraction uncovered.

Proof: At the beginning, there are m sets covering all, so there is a set S_i covering at least 1/m fraction of the elements. After Greedy's first choice, the fraction of uncovered elements is less than 1 − 1/m. In the next step, there are still m sets covering all uncovered elements, so there is a set which covers at least 1/m fraction of them. That is, after Greedy's second step, the fraction of uncovered elements is less than $(1-1/m)^2$. Similarly, after t time-steps, Greedy has at most $(1-1/m)^t$ fraction uncovered.

According to the corollary in (Hochba, 1997), Greedy is a factor $\ln n$ approximation algorithm.

Proof: Optimally sets cover everything. Once less than *1/n* fraction uncovered, it is done.

$$(1-1/Opt)^t < (e^{-1/Opt})^t$$

$$(e^{-1/Opt}) \quad < 1/n$$

$$t \quad > \ln n \times Opt$$

Conclusions

The proposed approach can be used to allocate Masters and Workers in optimal manner. With a knowledge base or a rule base the existing mapping suitability, coupling abilities, distances and communication barriers are being considered as knowledge, and they are being introduced to ease the causes which create Grid delays. Therefore this can be seen as a solution for existing complex Mater-Worker allocation problems in spite of the degree of the Master-Worker relationship. It can be easily adapted for any of the existing Master-Worker paradigms

References

Cesar, A. E. Moreno, J. Sorribes. and Luq E. (2006), Modeling Master/Worker applications for automatic performance tuning, Parallel Computing, Vol. 32(7), pp: 568 - 589 .

Hochba D. S. (1997). Approximation Algorithms for NP-Hard Problems, University of California, New York, USA.

Morajko, E. A., Cesar, P., Margalef C. T., Sorribes J. and Emilio L., (2005). Automatic Tuning of Master/Worker Applications. LNCS, vol. 3648, Springer-Verlag, pp. 95–103.

Pieczykolan, L. J., Korcyl D. K., Slota R., Kryza T. and Kitowski J. (2007). Adapting master-worker paradigm for high throughput applications in grid environment, http://para08.idi.ntnu.no/docs/submission_55.pdf.