Enhancements on Static Load-balancing Scheme for Parallel Numerical Simulations

Graduate Advisor: Shuichi Ichikawa 931714 Yoshikatsu Fujimura

1. Introduction

Ichikawa[1] showed that the static load-balancing scheme of parallel numerical simulation language NSL is modeled as a combinatorial optimization problem, which can be solved by branch-and-bound method. However, this scheme is only applicable in case $m \ll n$ holds, where m is the number of blocks and n is the number of processors. Though virtual processor approach[2] can alleviate this problem, it does not give the best solution. More general approach is desired.

This paper presents a new static load-balancing scheme that can handle the case of $m \geq n$. The problem is modeled as a packing problem and is solved by branch-and-bound method. This paper also presents some effective approximation algorithms that give good approximations.

2. Model

This paper adopts the calculation model based on the Procedure-1 described by Ichikawa[1]. Computational domain consists of m rectangular blocks, each of which is processed in parallel. Each processor handles one or more blocks, because $m \geq n$ is assumed here. Communication is required between blocks to exchange data on border, but the communication between the blocks allocated in the same processor is accounted as zero. The processing time of a block is given by the sum of calculation time and communication time. The processing time of a processor is the sum of the processing time of the allocated blocks. Total processing time is determined by the biggest of the processing time of all processors. The problem is to find the allocation of m blocks among n processors to make total processing time minimal. This incurs $O(n^m)$ search space.

3. Approximation Algorithm

This optimization problem can be solved by branch-and-bound method. For practical use of branch-and-bound method, good approximation algorithms are essential. This paper propose five approximation algorithms that take both calculation time and communication time into consideration (Approx1,...,Approx5). Further improvement by local search (Local) is attempted. Best Effort method, which applies a couple of approximation algorithms and takes the best result from them, is also evaluated.

4. Result

Figure 1 shows the accuracy of algorithms for various numbers of blocks. The results in Figure 1 are normalized by the optimal solution. Each block is set to square of random generated size in simulations. The number of processor is 4, processing rate of processors is uniform. Approx5 gives a good approximation solution. Local search seems effective. The errors of Approx5, Approx5+Local and Best Effort are all less than 10 % with 4 processors and 32 blocks.

In case of more than 32 blocks, the problem is too hard to find optimization solutions. Therefore, the results shown in Figure 2 are normalized by the Best Effort method. Also in this case, Approx5 and Approx5+Local give good approximation solutions. Local search is still effective.

Figure 3 shows the execution time of approximation algorithms. "Optimization" is the time to find optimal solutions by branch-and-bound method. It is shown that approximations are derived in rather shorter (and practical) time. Approx5 and Approx5+Local are effective yet from this point of view.

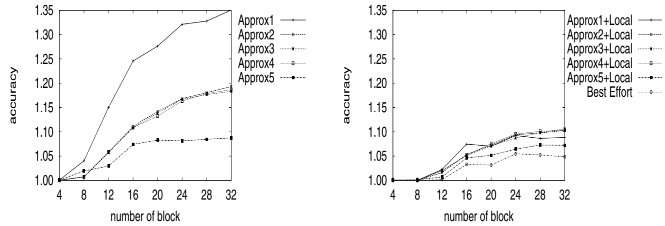

Figure 1: Accruracy of algorithms

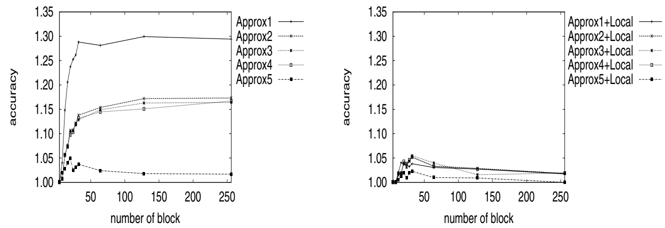

Figure 2: Relative evaluation of algorithms

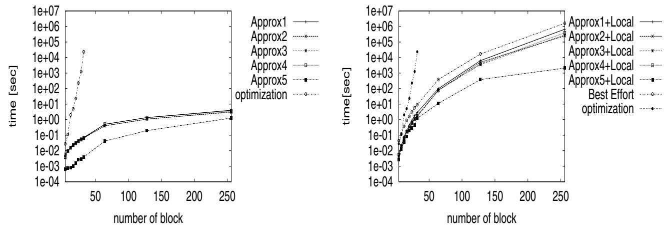

Figure 3: Execution time of algorithms

Even in case of heterogeneous processors, results are similar to the case of uniform processors.

5. Conclusion

It is still difficult to get a good load-balance when dispersion of calculation time is very big among blocks. To overcome this problem, automatic data partitioning is required, as described in the paper [1]. A new method that applies both automatic partitioning and scheduling are desired.

References

[1] S.Ichikawa, T.Kawai, T.Shimada: Static Load Balancing for Parallel Numerical Simulation by Combinatorial Optimization, Trans. IPS. Japan , Vol.39, No.6, pp.1746-1756 (1998).

[2] S.Ichikawa, T.Kawai, T.Shimada: Enhanced Optimization Scheme for Parallel PDE Solver of NSL, Proc. JSPP '98, p.143 (1998).