Dynamic Load Balance Techniques for Distributed Computations on Parallel Heterogeneous Clusters

Yu.P. Galyuk^2, V.P. Memnonov^{*1}, and V.I. Zolotarev^2

^{1}St.Petersburg State University, Inst.Math.Mech.,
28, Universitetski pr., St.Petersburg, 198504, Russia

^{2} Petrodv. Telecommun. Center,
1, Ulyanovskaya st., St.Petersburg, 198504, Russia

Key words: Dynamic load balance tachnique, metacomputer.

Introduction

Contemporary distributed memory parallel clusters have very high computational potential and at the same time their price growth is relatively reasonable. So they are very valuable for solution of time-consuming CFD problems including of course those which are numerically simulated by statistical MonteCarlo methods. In the latter case for reducing statistical scattering in calculation results one needs very large samples which could be obtained simply by increasing computer number in a cluster or the number of the clusters itself. But with this cluster enlarging several new issues come into play. First of all the cluster could be enlarged most easier by older computers and thus it becomes heterogeneous. So it must be supplied with some at least static load balancing. For expanded times of simulation one inevitably will meet also with computer performance time instabilities which will demand some dynamic load balancing. And finally very large number of computers used for one MPI-problem increases its crash possibility as the result of a breakdown of any single computer. So it is very important at least partly to diminish all these drawbacks with the help of some load balance technique.

Dynamic load balancing for Monte Carlo simulation

In the present paper we consider two such procedures for Monte Carlo simulation of a filter flow in transitional regime. Large statistical scattering is a standard difficulty in all applications of the Monte Carlo simulations but it is especially severe for flows with small values of mean gas velocities relative to the thermal molecular speeds. The statistical scattering in our filter problem, which is of this kind, has been strongly diminished by enlarging our computer number and the samples through Internet-connection several clusters into a metacomputer. Special dynamic load balance technique under our distributed memory conditions was developed for maintaining an optimal performance of each processor being at our disposal. A quantitative estimation of its performance is given. It is shown in particularly that even with communications through just 1 Mb/s Interntet channel it is possible to obtain the efficiency of processor's employment as high as 93%-94%. This is only 2%-3% lower than for pure separate computation on that particular cluster. The technique appeared to be very effective also for cases when performances of some processors can be substantially changed during the calculation itself, for example, because of inserting by the other users their computational tasks on the same computer. Thus problem of so to say dynamical heterogeneity has been overcomed.

Problem description

We consider the flow through two-dimensional infinite in Z-direction channel which connects two reservoirs with the same gas, the particle densities of them being n1 on the left and n2=0.8n1 in the right. Molecular interaction assumed to be the hard sphere one and their

[*] E-mail: pokusa@star.math.spbu.ru

interaction with the channel walls is taken to be diffuse reflection. At the both of the reservoirs temperatures and Maxwell velocity distribution functions were considered to be the same and not perturbed by outgoing streams. The channel width was changing from λ/7 to 7λ, where λ is mean free path under atmospheric density n1. With the help of DSMC method we were computing profiles of the mean velocity and density inside the channel and some other parameters.

Distributed multi cluster computations

Separate parallel MPI implementations of the DSMC method for this problem were employed on three clusters, which were operated under different nets: 100Mb/s Ethernet, 1Gb/s SCI and 1Gb/s Myrinet. All of them had distributed memory, two of them being linked into a local net by 100Mb/s channel, the Internet connection with the third cluster "Paritet" of Institute for High Performance Computing was realized through 1Mb/s Internet channel, which nevertheless sometimes showed much lower real performance, especially for transference of relatively large files. Each processor of the whole system produced independent realizations; the results were sent to the corresponding leading one and then gathered at one of them for averaging and outputting. Our dynamic load balance (DLB) technique employs the transference of small files between different clusters for circulation of control information, which contains the number of realizations remaining to be done for production the required statistical sample. This information is accessible to every computer. Each processor in its turn did not have initially task to perform any fixed number of independent realizations, being guided only by this control information. When, for instance, as a result of an insertion by some user an additional job for a computer and its performance diminishes correspondingly, the others automatically take on its load portion of our problem. This is correct also for a problem that has parallelization by space decomposition, when the latter could be placed within a single cluster.

Quantitative performance estimate for the present DLB Technique

For this purpose we have used comparison of the values for mean average execution time t_{av} of a fixed simulation problem realization. First we have calculated t_{av} and corresponding efficiency E for its execution at the cluster “Paritet” alone by using different numbers of processors N. In the Fig. 1 this efficiency E is represented by the bold solid curve with the right E-axis, the simple solid curve there shows corresponding times t_{av} with the left axis t_{av} in seconds. Then we have connected several clusters by our DLB technique and measured these times t_{av} for the same cluster again, including of course time for final averaging and outputting. Result is represented by the dashed curve which only slightly, by several percents, overshoots the solid one. This means that the efficiency of the processor performance utilization in “Paritet” by our DLB technique is likewise only the same percents lower.

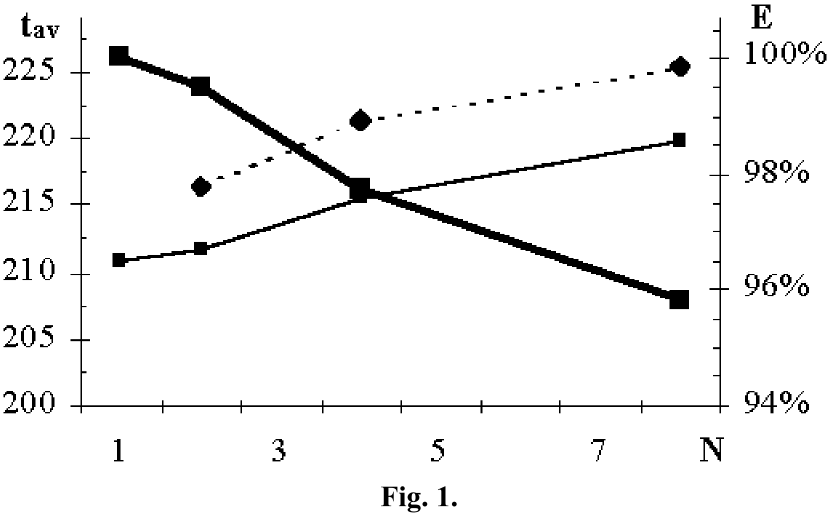

Fig. 1.

Some Numerical Simulation Results

On the Fig. 2 as an example two curves for different channel half widths *a* represent the ratio Q of the molecular volume rate W_r,W_r=U/(Δn/Δx), related to W_0=V_T/(n1/a), while U is the average molecular velocity at the outlet section and V_T is most probable velocity. This ratio Q depends upon the channel flow resistance. The typical Knudsen minima are seen in both cases in its dependence on the reverse Knudsen number Kn^{-1}.

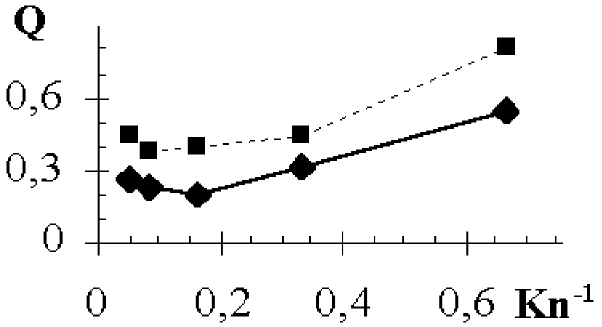

Fig. 2.

Computer breakdown resistant load balancing

It should be mentioned that presented above DLB technique was tested on clusters which had processor numbers *N* not large, *N<12.* However for clusters with large number of processors it displays certain limitations. First of all a bottleneck effect seems to appear. When many processors nearly simultaneously try to open the same control file they inevitably fall into idle queue. We have found that already with *N=32* the bottleneck exhibits itself by 10% time lag while executing some fixed program in comparison with simple static load balancing. So for the case N>>1 a new load balance technique was developed which avoids the bottleneck by combining an initial static load balancing with the dynamic one. The latter is accomplished by monitoring the appearance or not appearance of certain files and by utilizing *system* commands for their quick recording. The same technical tricks allow us to diminish crucial consequences of a computer breakdown in one of the clusters by automatically monitoring this event and disconnecting the latter from the others. Thus the whole problem is not interrupted and stopped, only the total statistical sample being diminished by the portion initially intended for the misfortune cluster.

Conclusion

The load balance techniques developed in this paper could be applied depending on the value of processor number *N.* They improve efficiency of processors utilization in clusters and reliability of computations, as it has been shown for an example with Monte Carlo flow simulation.