Publ. Astron. Obs. Belgrade No. 96 (2017), 257 - 262 *Contributed Paper*

RECOMMENDATION FOR RUNNING PURE N-BODY SIMULATIONS ON COMPUTING FACILITIES IN SERBIA

N. MARTINOVIĆ[1], MIROSLAV MIĆIĆ[1], MILICA MIĆIĆ[1], A. OBULJEN[2], M. SMOLE[2], S. MILOŠEVIĆ[2], A. MITRAŠINOVIĆ[2], M. STOJANOVIĆ[1] and M. SMAILAGIĆ[1]

[1]*Astronomical Observatory, Volgina 7, 11000 Belgrade, Serbia*
E–mail: nmartinovic@aob.rs

[2]*Department of Astronomy, Faculty of Mathematics,*
Studentski trg 16, 11000 Belgrade, Serbia

Abstract. Pure (gravitational) N-body astrophysical simulations are an irreplaceable means of testing dynamics and evolution of astrophysical objects both on large scales (cosmological simulations) and on smaller scales (isolated galaxies, globular clusters, solar system dynamics, etc). One of the obvious major limitations is in computing facilities at disposal for running the simulations. In this talk we will present characteristics of computational facilities available in Serbia. We will discuss their advantages and disadvantages and we will try to give recommendation for optimum and maximum scope of both large scale simulations and small scale simulations which may be performed with those facilities. Quick overview of simulations completed so far will be given as well.

1. INTRODUCTION

For past several decades importance of N-body astrophysical simulations continues to grow. Today N-body simulations are wide-spread tool for analyzing dynamics of astrophysical objects on all scales (eg. Dehnen & Read 2011). Development of computing hardware, optimization of computing codes and their public availability are a reason why it has become standard to run tens of simulations within a single project.

Unfortunately one of the unpleasant issues when doing numerical astrophysics is estimating how much time, called wall time (in contrast to simulation time), it will take to complete a simulation. There is no equivalent to something like exposure calculators from observations, which makes the estimation even more harder. Usually researchers just acquire intuitive sense for duration of a simulation with a given number of particles after certain level of experience. To make things even worse - different codes, different computing hardware and different N-body problems in astrophysics all can yield significantly different simulation duration.

Considering that N-body astrophysical simulations (large scale and galactic simulations) are a relatively new field at Astronomical Observatory in Belgrade, researchers starting in the field have a hard time estimating how much time certain simulations will take, or related to that - on what number of processors simulation should be performed in order to take optimal time of execution. Here we will try to address that issue by giving estimate how much time will be needed for a simulation with N particles with different computational facilities available for use in Serbia. We hope that this will be at least good starting point that will mitigate away portion of time needed for reaching optimal configurations.

2. COMPUTING FACILITIES

There are several computing facilities in Serbia that have been used by the authors of this work, and although there are a few more that can be used as well, we will focus on those with which we had experience, especially because they are a good representation of a computing hardware of various properties and power.

Specifications are given in Table 1. Available machines that have been used cover wide range from single desktop computer ("Phobos"), to a supercomputing facility ("Paradox"). Training and test-model simulations are usually executed on less powerfull machines. Largest share of workload falls upon "Fermi" - a cluster located within Astronomical Observatory acquired through one of its projects. "Paradox" machines are located within Institute of Physics in Belgrade and they are previous iteration ("Paradox" - still available for use) and the new upgrade ("Paradox 4"), which has not yet been tested for astrophysical N-body simulations. Furthermore, Paradox supercomputers are important part of SEE-GRID-SCI[1] project and are part of PRACE[1] network.

Another interesting thing that can be seen in the table is presence of GPUs. Considering major speed-ups with inexpensive equipment (compared to similiar performance reached only with CPUs) when it comes to using GPU codes for N-body simulations, they represent both important testing ground for development and one possible future of the simulations (Aubert Dominique 2011). Focus is on GPUs with CUDA capabilities considering availability and quantity of work done with it so far. Significant speed-ups can be achieved even with commercial desktop CUDA GPUs, which will be shown later.

3. CODES

All the CPU codes that are beeing used for executing N-body astrophysical simulations are paralelized, which is a necessary condition to be able to fully utilize computing power available. Important part of a code is its optimization. Unoptimized codes are not able to fully utilize computing power available to them, rendering usage of higher number of processors or more powerfull machines completely useless (for example due to bad paralelization).

Although there are many publicaly available N-body codes, we focus on GADGET2 and P-GADGET3 (Springel 2005) - both are highly parallelized Tree-PM N-body codes with efficient and tested paralelization.

As for the GPU codes, so far only one has been succesfully run (many codes are still early in development). Code in question is BONSAI (Bedorf et al. 2011), fully GPU N-body Tree-code which does not (for now) support individual softenings and unfortunately it was succesfully run only in a single card mode.

[1] http://www.scl.rs/about-us

Table 1: Computing facilites in Serbia. Specifications are for each node in the system, where second column gives total number of available nodes, third and fourth column are CPU type/number and number of cores (threads) per node, with the fifth column stating amount of RAM memory per node. Last column is GPU unit type (if exists) available on each node.

Name	Nodes	CPU	Cores per CPU	RAM	GPU
Paradox	89	2x E5345	4	8 GB	No
Paradox 4	106	2x E5-2670	8	32 GB	M2090
Fermi	12	2x X5675	6	24 GB	2x M2090
Phobos	1	i7-2600	4/8	8 GB	GeForce 650Ti
Office (Beowulf)	4	i5-3470	4	16GB	No

4. SPEEDUP

When running a simulation the most important issue becomes executing it with the available resources as quickly as possible. But simply increasing used number of processors does not (in vast majority of cases) yield equivalent acceleration in execution time. Not only that, but using number of processors greater than the optimization of a code allows is usually frowned upon due to wasting resources (for example: wasting electricity and/or locking processors thus making them unavailable to other users, but failing to utilize them properly).

Quantity that has been in use for determining optimal processor usage while running parallelized codes is speed-up (Hennessy & Patterson, 2012):

$$S(N) = \frac{T(1)}{T(N)} \tag{1}$$

where T(N) is a time the algorithm takes to finish running on N processors (or threads). So basically S(N) gives how quicker algorithm on N processors would finish than it would on a single processor (in a serial mode), although it can be calculated against execution time on any number of processors. Speed-up is considered linear if it rises as N. It can be super-linear if the speed-up is greater than the N (number of processors), although in vast majority of cases when it comes to N-body simulations it will be sub-linear, that is, it will rise more slowly than N. In those cases there will be a point after which further increase in number of processors wont yield in reasonable speed-up.

Understanding why this happens is given through Amdahl's law (Amdahl 1967) - it states that if P is the paralized portion of the algorithm and (1-P) is the part of the algorithm that can not be paralized (serial part), then algorithm will never execute quicklier than execution time of the serial part, and speed-up would perform as:

$$S(N) = \frac{1}{(1-P) + \frac{P}{N}} \tag{2}$$

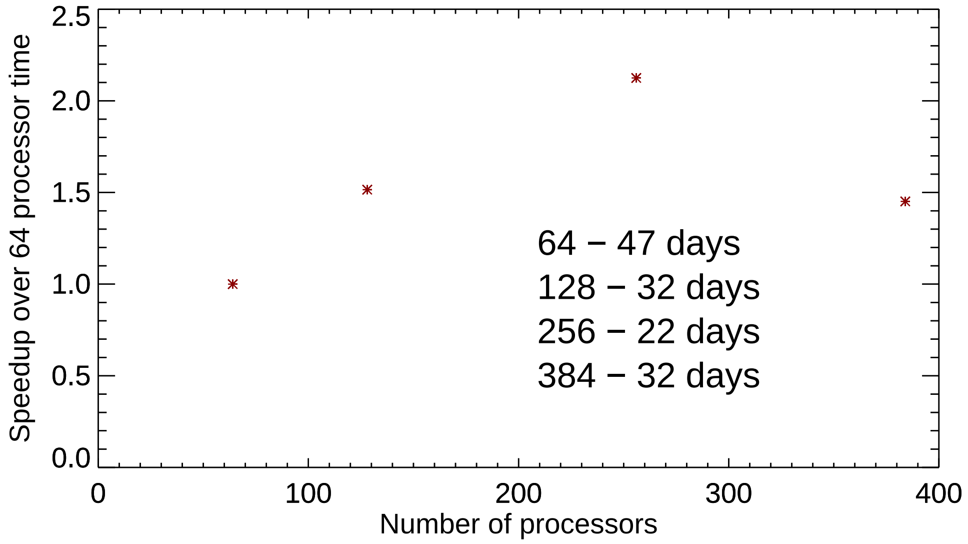

Figure 1: Empirically derived speed-up for an N-body cosmological simulation tested using GADGET2 code for 1 hour on each number of processors. Given are the estimates on total execution time for the simulation based on the retrieved results. It can be seen that there is a peak performance on 256 processors, which was ultimately chosen for the simulation run. Tests and simulation itself were performed at IPB's Paradox supercomputer.

So it is clearly seen that we can by using greater number of processors speed-up only parallelized portion of the code which would behave asymptotically after certain N.

Although it is quite difficult calculating which speed-up is the most optimal, in most cases that turning point can be acquired empirically, by testing at which point a further increase in processor number yields no significant acceleration. One of such tests is represented in Figure 1. where speed-up for a cosmological simulation that was perfomed on "Paradox" supercomputer was tested for a given number of processors prior to full run. Each test run was one hour long. After the tests 256 processors were choosen as platform to execute a simulation.

5. RESULTS AND DISCUSSION

As mentioned before, there is plethora of possible N-body astrophysical simulations with many parameters that can affect the length of execution. But in our case, constraining to a few N-body codes and to several computing facilities can provide enough insight into length of the different types of simulations. Results from some characteristic runs so far are given in Table 2. There we can see number of particles against type of the simulation and its total execution time on a given number of processors. All the simulations were performed using GADGET2, while the last one (isolated one) is performed using BONSAI GPU code.

Table 2: Characteristic results for various simulation types (last column) given against number of particles used in the simulation (first column), total simulation time (second column), number of processors used (third column) and wall-time (fourth column). Last, singled out entry, is the simulation performed with the GPU code.

N	Sim time	Proc	Total days	Type
440k	0.92 Gyr	24	3.4	Minor merger
550k	5 Gyr	24	2.0	Galaxy fly-by
2M	10 Gyr	24	0.65	Fornax
2.2M	10 Gyr	24	0.22	Dwarf Isolated
4.2M	10 Gyr	24	17.6	Dwarf/Fornax
2M	10 Gyr	650Ti	0.17	Fornax

Looking at Table 2 it is possible to gain expectation on running time of the simulations. Not only that, but simple execution time can hint us important things about simulation. For example: first simulation in the list behaves quite distinctly in comparison to the others. For a shorter simulation run it executes quite longer than the others. So here we use simple length of the simulation to assume that there are underlying problems with it - in this particulare case, galaxies involved were slightly unstable, which manifested some time into the simulation slowing it. Simulation was aborted for additional check-ups, revealing mistakes and thus saving wasting resources.

Another interesting thing from Table 2 are both "Fornax cluster" simulations. They were performed with identical initial conditions, but with CPU and GPU codes and here we can see how quicklier GPU code is - for the same simulation, running on only one GPU card (against 24 CPUs) we got speed-up of almost 4 times. If development of GPU codes continues, they will definetly play an important part in the future of the simulations.

In the end we have summarized our experience into the recommendations given in the following list:

- Processors strength is (usually, with modern processors) not an issue;
- For ∼ 100k particles, 8 processors is sufficient;
- For ∼ 1M particles, 30 processors is sufficient;
- For ∼ 10M particles, up to 100 processors;
- For ∼ 100M particles, up to several 100 processors;
- For 1000M+ particles - yet to be seen!

One last thing to consider are possible queues and other time consuming deviations that can be inherently connected with systems. In practice if you are choosing between two cluster (or supercomputers) from our experience the best is to use the one under less load, considering that you might spend significant time (in comparison to simulation length) for example in waiting for resources to be alocated for your simulation. For example, for test simulations with smaller number of particles the most efficient would be simply using “Phobos” or (yet to be named) office cluster (no queues, data immediately available, etc). On the other hand, full simulations with large number of particles should be run on larger supercomputers.

Acknowledgments

During the work on this paper the authors were financially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia through project no. 176021 'Visible and invisible matter in nearby galaxies: theory and observations'.

Numerical results were obtained on the FERMI cluster at the Astronomical Observatory in Belgrade supported by the national research project III44002 and on PARADOX cluster at the Scientific Computing Laboratory of the Institute of Physics Belgrade, supported in part by the national research project ON171017, funded by the Serbian Ministry of Education, Science and Technological Development.

References

Amdahl, Gene M.: 1967, in AFIPS Proceedings of joint computer conference, p. 483-485.

Aubert, Dominique: 2011, in Proceedings of the International Astronomical Union, IAU Symposium, Volume 270, p. 397-400.

Bedorf, J.; Gaburov, E.; Portegies Zwart, S.: 2011, in ASP Conference Proceedings, Vol. 453, p. 325.

Dehnen, W.; Read, J. I.: 2011, *The European Physical Journal Plus*, **126**, 55.

Hennessy, John L.; David A., Patterson: 2012, Computer Architecture: A Quantitive Approach. Waltham, MA: Morgan Kaufmann.

Springel, Volker: 2005, *Mon. Not. R. Astron. Soc.*, **364**, 1105.