gravitySimulator: Beyond the Million-Body Problem

Stefan Harfst, David Merritt, Peter Berczik and Rainer Spurzem

Rochester Institute of Technology, Department of Physics, 54 Lomb Memorial Dr, NY 14623

Abstract: One of the most cpu-intensive calculations in astrophysics is the gravitational N-body problem. The N -body problem is particularly challenging in the case of galactic nuclei, because of the steeply-rising stellar density profile and the presence of single or multiple supermassive black holes. Not only must the particle advancement be very accurate, but the value of N must be chosen large enough that two-body scattering does not artificially repopulate the loss cone of the central object on time scales shorter than the orbital period. This requirement imposes a minimum value for N, which for typical applications is of order 10^6 or greater.

The state-of-the-art way to deal with such large particle numbers is via the GRAPE special-purpose computers. But the finite on-board memory of the GRAPEs limits the number of particles that can be handled. This limitation can be overcome by linking multiple GRAPE boards into a cluster. Such a cluster ("gravitySimulator") has recently become operational at the Rochester Institute of Technology.

In this poster, we compare different ways of implementing a parallel N-body code on the GRAPE cluster and test their performance.

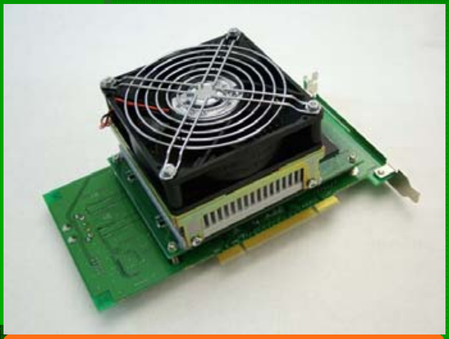


Fig 1: A single GRAPE-6A card fits into the PCI-slot of a common PC and can accelerate the force calculation in an N-body simulation by a factor of 100. The peak-performance is 125Gflops but the card's memory can only hold 128k particles.

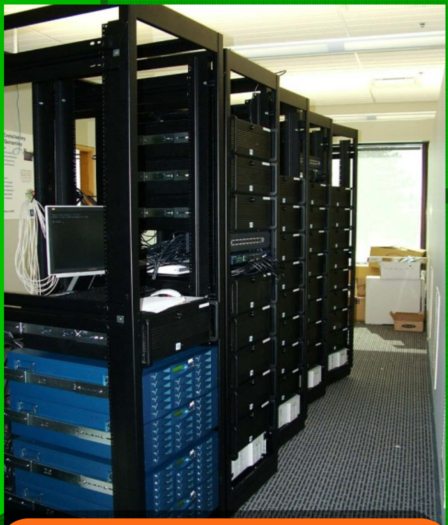
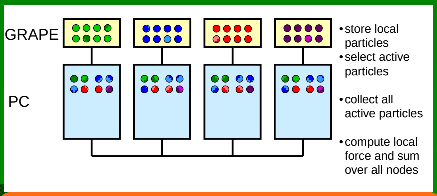


Fig 2: The RIT GRAPE cluster "gravitySimulator" is in operation since February 2005. Here are some of the technical details:

- 1 head node and 32 computing nodes
  - dual 3GHz Xeon processors with 2Gbyte of memory
- 32 GRAPE-6A cards
- 14 Tbyte RAID array
- fast low-latency InfiniBand interconnect
  - 10Gbps
- theoretical peak-performance is 4Gflops
- N up to 4x10^6
- Cost: \$0.5x10^6
- Funding: NSF/NASA/RIT
- Next largest: 24 nodes (University of Tokyo)
  - A similar 32-node cluster will soon be operational in Heidelberg, Germany



- store local particles
- select active particles
- collect all active particles
- compute local force and sum over all nodes

Fig 3: A schematic illustration of our parallel GRAPE-N-body code. Active particles are selected from the local particles and then collected on all nodes. Partial forces are computed and summed up.

Algorithm: The basic algorithm is a direct-summation code (NBODY1) with a fourth-order ("Hermite") integrator. Individual, block time steps are used and a force softening can be applied. Different parallelization schemes have been tested, including systolic (Dorband et al., 2003) and broadcast (Gualandris et al., 2004) algorithms, and a newly implemented scheme (Harfst et al., 2005). In the latter, the N particles are distributed evenly among p processors. After that, the system is advanced by the following steps: 1.) For each particle i the time step Δt_i is calculated and the global minimum is determined. 2.) Active particles, i.e. the particles that need a force update for the current time step, are first selected locally on each node and then collected globally. 3.) After that, partial forces are computed locally and summed up over all processors. 3.) Finally, the time step is completed by advancing the local particles on each node. These steps are repeated until the system has evolved to the desired time.

Results: We have compared the performance of different parallel N-body algorithms on a 32-node GRAPE cluster. The results can be summarized as follows:

- The best performance was achieved with a new parallel scheme in which all nodes simutaneously compute partial forces for all active particles. This ensures that the GRAPE is used most efficiently, since the number of particles requesting a force update is always as large as possible.
- The cluster is used most efficiently when the number of particles per nodes is close to the memory limit of 128k particles of the GRAPE-6A card. For much smaller particle numbers, the computation time becomes short compared to communication time.
- For a few processors the maximum efficiency is between 80% and 90%. For 8 and more processors the efficiency is between 60% and 70%. The efficiency does not strongly depend on the central concentration of the model.
- A run with one million particles on eight nodes reaches a speed of ~675Gflops. A simulation with 4M particles on 32 nodes achieves a speed well above 2Tflops.

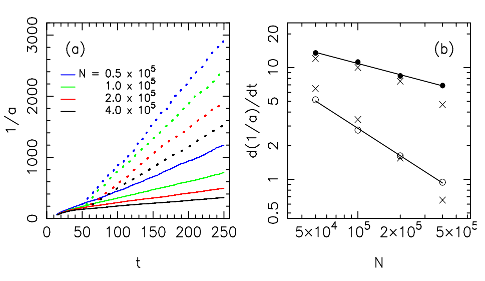


Fig 6: Evolution of the semi-major axis (a) and hardening rate (b) of binary black holes at the center of Plummer-model galaxies. (a) Black holes have masses $M_1=M_2=0.005$ (dashed lines) and $M_1=M_2=0.02$ (solid lines). (b) Filled (open) circles are for $M_1=M_2=0.005(0.02)$. Crosses indicate the hardening rate predicted by a model in which the supply of stars to the binary is limited by the rate at which they can be scattered into the binary's influence sphere by gravitational encounters. The simulations with largest (M_1,M_2) exhibit the nearly 1/N dependence expected in the ``empty loss cone'' regime (see Berczik et al. (2005) for details).

References:

Berczik P., Merrit D., Spurzem R., 2005, astro-ph/0507260
Dorband E.N., Hemsendorf M., Merritt D., 2003, JCP, 185, 484
Gualandris A., Portegies-Zwart S., Tirado-Ramos A., 2004, IEEE
Harfst S., Merritt D., Gualandris A., Berczik P., 2005, in preparation

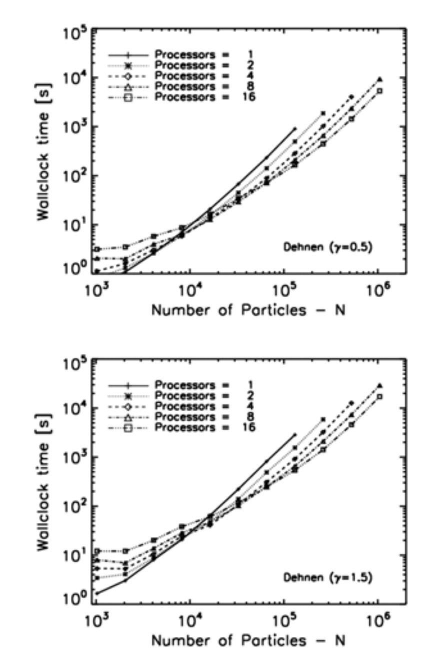


Fig 4: Timing results from simulations using two different Dehnen models. The wallclock time is measured for one full time step Δt=1. Generally, the more concentrated model (γ=1.5) take longer to compute. For larger particle numbers (roughly N>10^4) the wallclock time scales with N^2 as expected and using additional processors decreases computation time. Below 10^4 particles computation time increases for more processors due to the overhead in communication.

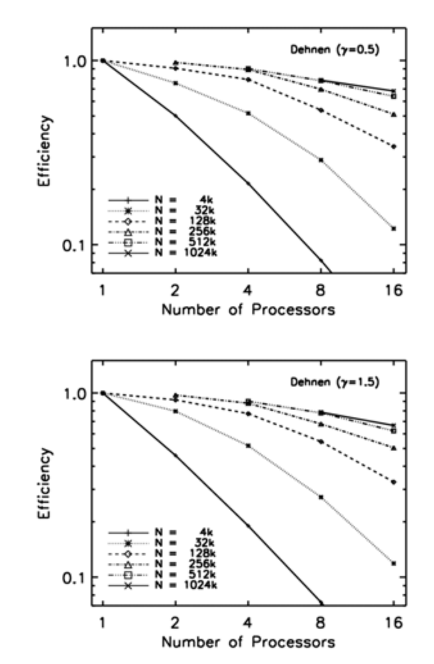


Fig 5: Efficiency – defined as computation time needed on one processor divided by p times the time needed on p processors – for different particle numbers. Unit (perfect) efficiency corresponds to zero communication and latency losses. For low particle numbers, the force calculation is very fast and communication dominates the run time. The code runs most efficiently if the number of particles on each node is close to the maximum (128k) permitted by the GRAPE memory.

For further information and updates about on-going projects including a new visualization tool visit the website of the GRAPE cluster project at

http://www.grapecluster.rit.edu/