Astronomical Data Analysis Software and Systems VII
ASP Conference Series, Vol. 145, 1998
R. Albrecht, R. N. Hook and H. A. Bushouse, eds.

Parallel Tree N-body Code: Data Distribution and DLB on the CRAY T3D for Large Simulations

U. Becciani, V. Antonuccio-Delogu, M. Gambera and A. Pagliaro

Osservatorio Astrofisico di Catania, Città Universitaria, Viale A. Doria, 6 – I-95125 Catania - Italy

R. Ansaloni

Silicon Graphics S.p.A. St.6 Pal. N.3, Milanofiori I-20089 Rozzano (MI) - Italy

G. Erbacci

Cineca, Via Magnanelli, 6/3 I-40033 Casalecchio di Reno (BO) - Italy

Abstract. We describe a strategy for optimal memory and work distribution. We have performed a series of tests to find an *optimal data distribution* in the Cray T3D memory, and to identify a strategy for the *Dynamic Load Balance* (DLB). The results of tests show that the step duration depends on two main factors: the data locality and the network contention. In a very large simulation, due to network contention, an unbalanced load arises. To remedy this we have devised an automatic work redistribution mechanism which provided a good DLB.

1. Introduction

N-body simulations are one of the most important tools in contemporary theoretical cosmology, however the number of particles required to reach a significant mass resolution is more larger than those allowed even by present-day state-of-the-art massively parallel (Gouhing 1995; Romeel 1997 & Salmon 1997) supercomputers (hereafter MPP). The most popular algorithms are generally based on grid methods like the P^3M. The main problem with this method lies in the fact that the grid has typically a fixed mesh size, while the cosmological problem is inherently highly irregular. On the other hand the Barnes & Hut (1986, hereafter BH) oct-tree recursive method is inherently adaptive, and allows one to achieve a higher mass resolution. Because of these features, the computational problem can easily become unbalanced and cause performance degradation. For this reason, we have undertaken a study of the optimal work- and data-sharing distribution for our parallel treecode.

Our Work- and Data-Sharing Parallel Tree-code (hereafter WDSH-PTc) is based on this algorithm tree scheme, which we have modified to run on a shared-memory MPPs (Becciani, 1996). The BH-Tree algorithm is a $NlogN$ procedure to compute the gravitational force, a more detailed discussion on the BH tree method can be found in Barnes & Hut (1986). For our purposes,

we can distinguish three main phases in each timestep: TREE_FORMATION (TF), FORCE_COMPUTE (FC), UPDATE_POSITION. Besides, in the FC phase we can distinguish two important subphase: TREE_INSPECTION (TI) and ACCEL_COMPONENTS (AC).

2. The WDSH-PTc

During the FC phase each PE computes the acceleration components for each body in asynchronous mode and only at the end of the phase an explicit barrier statement is set. Our results show that the most time-consuming phases (TF, TI and AC) are executed in a parallel regime. Tests were carried out, fixing the constraint that each PE executes the FC phase only for all bodies residing in the local memory. A bodies data distribution ranging from contiguous blocks (coarse grain: CDIR$ SHARED POS(:BLOCK,:)) to a fine grain distribution (tf) (CDIR$ SHARED POS(:BLOCK(1),:)) was adopted. We studied different tree data distributions ranging from assigning to contiguous blocks (tc) a number of cells equal to the expected number of internal cells (N_{Tcell}), [Salmon 1990](coarse grain: CDIR$ SHARED POS_CELL(BLOCK(:N_{Tcell}/N$PES),:)), to a simple fine grain distribution (CDIR$ SHARED POS_CELL:BLOCK(1),:)), (tf).

All the tests were performed for two different set of initial conditions, namely uniform and clustered distribution having 2^{20} particles each and they were carried out using from 16 to 128 PEs. In Tab. 1 we report only the most significant results obtained with 128 PEs. Our results show that the best tree data distri-

	PE#	p/sec	FC phase	T-step	UF
1Mun_tf_bf	128	4129	230.05	249.5	4.22
1Mcl_tf_bf	128	3832	250.32	268.81	4.57
1Mun_tf_bm	128	3547	270.51	290.45	5.90
1Mcl_tf_bm	128	3308	291.63	312.26	6.32
1Mun_tf_bc	128	4875	186.31	205.32	4.14
1Mcl_tf_bc	128	4490	203.37	222.72	4.38
1Mun_tc_bc	128	837	1051.93	1230.0	16.33
1Mcl_tc_bc	128	750	1173.24	1373.4	17.62

Table 1. Results of our tests for 10^6 of particles both in uniform initial conditions (1Mun) and in clustered (1Mcl); being UF the Unbalance Factor. The times in FC phase and in T-step are in second.

bution is obtained using a block factor equal to 1, then we can to conclude that a (tf) should be used for the kind of codes that we deal in this paper.

The fine grain bodies data distribution (bf) is obtained using a block factor $N = 1$; i.e., bodies are shared among the PE but there is no spatial relation in the body set residing in the same PE local memory. The medium grain bodies data distribution (bm) is obtained using a block factor $N = N_{bod}/2 * N\$PEs$; i.e., each PE has two data block of bodies properties residing in the local memory, each block having a close bodies set. At the end the coarse grain bodies

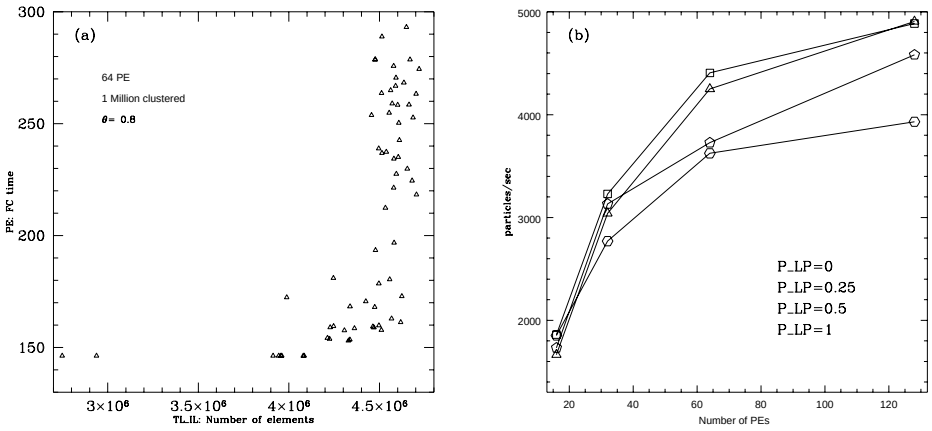

Figure 1. **a)** 64 PE run; **b)** 1 Million of particles: homogeneous configuration

data distribution (bc) is obtained using a block factor $N = N_{bod}/N\$PEs$; i.e., each PE has one close data set block of bodies residing in the local memory. The results reported in Tab. 1 show that the best bodies data distribution, having the highest code performance in terms of particles per second, is obtained using the block factor $N = N_{bod}/N\$PEs$ as expected, due to the data locality effect.

3. Dynamic Load Balance

Here, we present the results of a new DLB strategy, that allows us to avoid any large overhead. The total time spent in a parallel region T_{tot}, can be considered as the sum of the following terms

$$T_{tot} = T_s + KT_p/p + T_o(p) \tag{1}$$

where p is the number of processors executing the job, T_s is the time spent in the serial portion of the code, T_p is the time spent by a single processor ($p = 1$) to execute the parallel region, $T_o(p)$ the overhead time due to the remote data access and to the synchronization points, and K is a constant.

In the FC phase, there are no serial regions, so $T_p \propto$ to the length of the interaction list (IL). Using a coarse grain subdivision, each PE has a block of close bodies in the local memory ($Np = N_{bod}/N\$PEs$); in a uniform distribution initial condition, the PEs having extreme numeration in the pool of available PEs, have a lower load at each timestep. This kind of effect may be enhanced, if a clustered initial condition is used. If the number of PEs involved in the simulation increases we note that the data dispersion on the T3D torus increases. Our results do not show (see Figure 1a) the existence of a relationship between the time spent in the FC phase and the total length of the IL. The adopted technique is to perform a load redistribution among the PEs so that all PEs have the same load in the FC phase. We force each PE to execute this phase

only for a fixed portion of the bodies residing in the local memory NB_{lp}. NB_{lp} is given by

$$NB_{lp} = (N_{bod}/N\$PEs) * P_{lp} \tag{2}$$

being $P_{lp} = \text{const.}$ $(0 \leq P_{lp} \leq 1)$. The FC phase for all the remaining bodies

$$N_f = N\$PEs * (N_{bod}/N\$PEs) * (1 - P_{lp}) \tag{3}$$

is executed from all the PEs that have finished the FC phase for the NB_{lp} bodies. No correlation between the PE memory and the PE, executing the FC phase for it, is found. If P_{lp}=1 all PEs execute the FC phase only for bodies residing in the local memory, on the contrary if $P_{lp} = 0$, $N_f = N_{bod}$ $(NB_{lp} = 0)$, the PEs execute only N_f bodies and the locality is not taken into account. Several tests were performed with different values of P_{lp}. We report in Figure 1b, only, the case with 10^6 particles uniform.

The results obtained lead us that it is possible to fix a P_{lp} value allowing the best code performances. In particular, we note that is convenient to fix the P_{lp} value near to 0, that is maximize the load balance. The data show that, fixing the PEs number and the particles number, the same P_{lp} value gives the best performance both in uniform and clustered conditions. This means that it is not necessary to recompute the P_{lp} value to have good performances.

4. Final Considerations

The results obtained using the WDSH-PTc code, at present, give performances comparable to those obtained with different approaches such as Local Essential Tree (LET) (Dubinski 1996), with the advantage of avoiding the LET and an excessive demand for memory. Besides, a strategy for the *automatic* DLB has been described, which does not introduce a significant overhead.

The results of this work will allow us to obtain, in the next future, a WDSH-PTc version for the CRAY T3E system, using the HPF-CRAFT and the shmem library. The new version will include an enhanced grouping strategy and periodic boundary conditions (Gambera & Becciani 1997).

References

Gouhing, X., 1995, ApJ. Supp., 97, 884

Romeel, D.,& Dubinski, J., Hernquist, L., 1997, New Astronomy, 2, 277

Salmon, J., 1997, Proc. of the 8th SIAM Conf.

Barnes, J., & Hut, P., 1986, Nature, 324, 446

Becciani, U., Antonuccio-Delogu, V., & Pagliaro, A., 1996, Comp. Phys. Com., 99, 9

Salmon, J., 1990, Ph.D. Thesis, Caltech

Dubinski, J., 1996, New Astronomy, 1, 133

Gambera, M., & Becciani, U., 1997, in preparation