Implementing and Testing I/O in a Scientific Application on T3D

Elda Rossi, CINECA, Casalecchio di Reno (BO), Italy, *Roberto Ansaloni*, Cray Research S.r.l., Milano, Italy, and *Stefano Evangelisti*, Università di Bologna, Italy.

ABSTRACT: *We will describe the implementation of an out-of-core algorithm on the CRAY T3D. The introduction of heavy I/O activity was necessary to handle larger problems, and required particular attention in order to maintain good performances. We were able to nearly reach disk peak performance; we obtained tranfer rate of 38 Mbyte/s using two DD-60 channels on I/O Phase I and 75 Mbyte/s with four DD-60 channels on I/O Phase III.*

As an application, we used the algorithm to compute the Full-CI energy of the Be_2 molecule with a [9s2p1d] basis set (all electrons), with a Full-CI space of more than one billion (10^9) of symmetry-adapted determinants. We needed 27 iterations to get convergence; a single iteration on the CRAY T3D at CINECA (64 processors) required about four hours of elapsed time, 30 minutes of which spent in I/O activity. Due to the scalability of our code, substantially larger calculations could be performed provided that more processors and a larger amount of disk space were available.

Introduction

The present work describes the implementation of the *out-of-core* version of our Full-CI algorithm on the Cray T3D massively parallel computer.

Full Configuration Interaction (Full-CI) is a potentially exact quantum chemistry method, characterized by an extremely high computational complexity [1-9]. Therefore it is mainly used to investigate the accuracy of approximate methods. A few years ago, we developed a new Full-CI algorithm [9-12], implemented on CRAY Y-MP and C90 computers, with full exploitation of the vector and parallel (shared memory) characteristics of these architectures [13]. A preliminary *in-core* version (i.e., keeping all large vectors in memory) of the algorithm was then implemented on the CRAY T3D [14].

In order to handle larger problems we decided to write an *out-of-core* version of the algorithm, with all the Full-CI vectors kept on disk and copied to memory by blocks when needed, even if we were aware of introducing an heavy I/O activity in our problem. It is usually believed that I/O operations are not efficient on massively parallel processing (MPP) architectures and in fact on the T3D I/O performance does not scale with the number of processors. This work was done in order to investigate to what extent the introduction of I/O degrades the performance of the algorithm.

We will present some considerations on the *out-of-core* version of the code, whose performance is found to be comparable to that of the original *in-core* version. The results obtained so far show that overhead due to I/O operations is rather limited. Obviously, the use of the disk storage enables us to study systems with much larger Full-CI spaces. As an extreme application, we performed a benchmark calculation on the Be_2 molecule (all electrons) with a [*9s2p1d*] basis set. In this case the dimension of the Full-CI space is more than one billion (10^9) determinants in D_{2h} symmetry.

I/O Considerations

First of all, we rewrote the algorithm in order to reduce the number of I/O operations: In Figure 1. the final algorithm is sketched. The **X** and **Y** matrices are the important data structures of the problem, since their dimension scales directly with the dimension of the problem. They are block diagonal, and there are M_s of such blocks for each matrix, where M_s is the number of symmetry blocks, typically ranging from 2 to 8. In the original *in-core* version, the two matrices were dimensioned **(max_s,max_s,M_s)** and were distributed among processors by column blocks. In the present *out-of-core* version we store the matrices on disk, keeping in memory only one symmetry block of each matrix at a time. The two matrices are then dimensioned **(max_s,max_s)**. Since the whole **X** is needed to build a single block of **Y**, M_s read operations of the whole vector **X** are needed to build **Y**. Globally, for each iteration, the whole X is read

(M_S+1) times and written once, while **Y** is read and written once.

The T3D system handles the I/O requests through the C90 host, independently from the connection of the I/O subsystem [15]. The maximum I/O performance that an application can attain is thus independent from the number of T3D processors used: it is rather a function of the number of disk channels available on the host system, since I/O requests on different channels can proceed in parallel.

In the *out-of-core* version of the Full-CI algorithm, the basic I/O operations consist in the transfer to/from disk of an **X** or **Y** symmetry block. Each symmetry block is represented as a symmetric matrix distributed by columns across the N_P processors. A fairly sophisticated algorithm has been designed to minimize the amount of data transferred while using large I/O buffers in order to achieve high transfer rates. The symmetric nature of the matrix allows us to reduce the data transferred only to about one half (lower diagonal part) of the distributed matrix: this implies that after the read operation the upper diagonal part must be rebuilt by a symmetrization operation from the lower part.

In order to better balance the load and maximize the I/O record size, the lower diagonal part of the right half of the distributed matrix is stored in the scratchable upper diagonal part of the left half of the matrix (see Fig. 2). In this way the data to be transferred are contained entirely in the memory of the first (N_P/2+1) processors that handle the whole I/O.

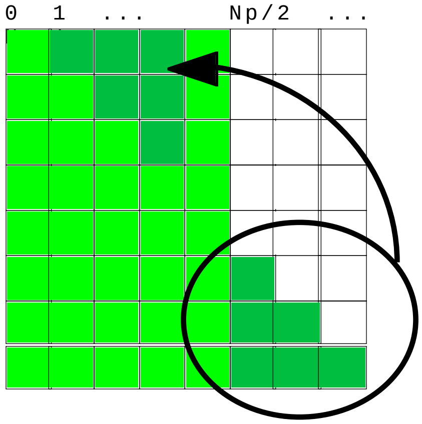

Figure 2. Data redistribution in a symmetric matrix

Several techniques have been employed in order to achieve maximum performance [16].

The record size has been rounded up to a multiple of the physical disk block (16 Kbytes in case of a DD-60). This *well-formed* request causes the data to be transferred directly to/from the physical device avoiding intermediate buffering activities. Files have been preallocated on contiguous disk chunks to get efficient disk activity.

```
do iter = 1, max_iter
      n0, e0 =0
      do sym_Y = 1, Ms
         Y = 0
         do sym_X = sym_Y + 1, ..., sym_Y  !  cyclic loop
            read (X)                        !  sym_X block
            Y = Y + alfa_beta (X)
         enddo
         Y = Y + beta_beta (X)
         Y = Y + transpose (Y)
         n0 = n0 + <X|X>
         e0 = e0 + <X|Y>
         write (Y)                          !  sym_Y block
      enddo
      do sym = 1, Ms
         read (X)                           !  sym block
         read (Y)                           !  sym block
         Y = Correction (X, Y, n0, e0)     !  see eq. (9)
         X = X + lambda Y
         write(X)                           !  sym block
      enddo
      check convergence
   enddo
```

Figure 1. The final algorithm

Direct-access I/O has been chosen to allow different processors to simultaneously access different records on the same file. In particular, the Asynchronous Queued I/O (AQIO) library [16] has been used to achieve high performance: in fact we were able to speed transfers near the theoretical peak of the disk (20 Mbyte/s in case of DD-60 disks). The processors' I/O activity is synchronized to minimize disk contention: I/O requests coming from different processors are issued in an ordered way so that disks are accessed sequentially. Therefore the I/O is seen as direct-access by the application but as sequential access by the disks.

The inter-processor communication is performed using the Cray *SHMEM library routines* [17]. In particular the data redistribution that follows the read operation is implemented with the high-performance (126 Mbyte/s) SHMEM_PUT routine [18].

The I/O transfer rate is further increased by splitting the data across files located on disks connected to different channels. Due to CINECA configuration (until last year), we could use only two channels in parallel and we were able to double the pure I/O transfer rate achieving 36 Mbyte/s out of a peak performance of 40 (20 for each channel) Mbyte/s. By this technique the I/O performance can scale with the number of disk channels, being limited by the peak bandwidth of the channel connecting the T3D to the C90 (200 Mbyte/s).

At the beginning of 1996, CINECA T3D was upgraded to 128 processors and eigth DD-60 disks were connected to the T3D IOC via *Phase III* path over 4 different channels.

We ran some preliminary tests using these *Phase III* disks in order to verify the effectiveness of the new configuration. The new I/O cluster is connected directly to T3D through a new (slave) I/O gateway: in case of *well-formed* I/O, data do not pass through the C90 host and this should result in a higher transfer rate or, at least, I/O performance should not be anymore dependent on C90 activity. Moreover, we intended to verify our scalability hypothesys about the number of indipendent I/O channels available.

The results of our preliminary tests show that the I/O performance does scale with the number of channels: this is true provided that the I/O request is *well-formed* and the environment variable MPP_AGENT_PH2IO is set to ON. It should be pointed out that, in a *Phase III* context, the definition of *well-formed* request is slightly different. In addition to the length of the request, that must be a multiple of the disk block, the user area must be aligned to an eigth words boundary in the memory of the T3D processor [19]. At a user level, this condition can be ensured with proper mppldr directives.

```
mppldr -D'dalign=min-size; dalignsz=64'
```

where min-size is the minimum size in bytes of the data structures to be aligned.

With both these conditions honored, we were able to obtain transfer rates in excess of 75 Mbyte/s, out of a peak of 80, writing/reading a generic matrix of (30x2048x128) words evenly distributed among 128 processors. The I/O operations were performed by all the processors, in an ordered way, using AQIO library routines on four different files preallocated contiguously on file system partitions associated to four different disk channels.

Applications

As an extreme application of this algorithm we performed a FULL-CI calculation on the ground state of Be_2 molecule, with all the electrons correlated and a [*9s2p1d*] basis set. This basis set is far too small to give a correct description of the extremely weak bond of this molecule, nevertheless, we think that the present Full-CI result will be useful to calibrate Truncated CI calculations that we plan to perform on this system with much larger basis sets in the future.

This basis set leads to a Full-CI problem of 8 electrons in 40 orbitals, with a Full-CI space of 1,061,893,156 determinants in the D_{2h} symmetry. The eight symmetry blocks have very uneven dimensions, going from a minimum of about 88 to a maximum of 187 million determinants each. We performed a total of 27 iterations, and we get an energy stable at least at one mhartree. Detailed information on the computation are contained in a paper recently submitted to Chem. Phys. Letters.

In Table 1 we report the timings for the Be_2 problem. It is important to stress that these values are only approximate since the I/O performance of the T3D strongly depends on the amount of activity present on the C90 host. The timings are those extracted from the fastest of a series of jobs run during normal production time. A single iteration requires about four hours ten minutes of elapsed time on the CINECA T3D (64 processors). The alfa_beta routine is the most time-consuming part of the algorithm, and also contains the greatest part of the I/O. However the I/O activities only account for about 10% of the time spent in alfa_beta.

Table 1. Timings of the most important parts of the program (one iteration).

routine	time (sec)
iteration	
15050.7	
X_HX	14408.4
alfa_beta	12945.2
read X	1273.7
beta_beta	1035.6
transpose	7.6
write Y	188.6
correction	543.4

Table 2 reports the I/O performance figures. The **X** matrix was written on two different files, preallocated on contiguous areas of two different DD-60 disks. Due to space constrains it was not possible to do the same for the two files containing the Y matrix. These files had to be allocated partially onto slower (DD-42) disks, and this is why the I/O for the Y matrix is slower. The peak transfer rate of a single DD-60 disk is 20 Mbyte/s, therefore the maximum speed expected is 40 Mbyte/s. The measured transfer rates are in excess of 32 Mbyte/s. Considering the fact that time spent for data re-arrangement is included, they are rather satisfactory. Tests performed with the same algorithm but with a matrix that does not require data re-arrangement, were in fact able to achieve a rate of 38 Mbyte/s.

Table 2. Timings and transfer rate of the READ and WRITE operations (one iteration)

operation	sec/call	calls	Mbyte/s	Gbyte
read X	160.8	9	32.2	4.09
write X	121.8	1	36.0	4.09
read Y	214.2	1	23.1	4.09
write Y	188.6	1	23.3	4.09
total I/O	1971.6	12		49.09

The availability of the T3E, next generation Cray MPP system, will greatly improve I/O performance scalability. In fact the T3E is a stand-alone system that does not require to be connected to a host system to perform I/O operations. Furthermore every group of four processors can share an I/O channel directly connected to I/O devices.

The present calculation gives also information on the possibility of larger benchmarks in the near future. The theoretical limit of the algorithm, using the largest T3D available today (1024 processors), is about 25 billion determinants in D_{2h} symmetry. The code has shown good scalability properties with respect to cpu, so we could foresee only a moderate increment in the elapsed computing time. More critical is the disk storage and the time required for the I/O activity, that, however, could be improved if a big amount of high speed disks and channels were available. To store the Full-CI vectors for this hypothetical problem, approximately 200 Gbytes of disk space would be required and, considering a maximum transfer speed of 200 Mbyte/s, we could expect a time of about 6000 seconds, for each iteration, devoted solely to I/O activities. Thus, the complete calculation would require about 7 hours of elapsed time for one iteration. Moreover, MPP systems architecture is moving towards increased I/O performance and scalability. For all these reasons, we believe that Full-CI benchmarks of the order of 10^{10} determinants are quickly becoming a realistic possibility.

This work was partially supported by EEC within the “Human Capital Mobility Program”. All the calculations were performed on CINECA Cray T3D.

References

[1] P.E.M. Siegbahn, Chem. Phys. Lett. **109**, 417 (1984).

[2] P.J. Knowles and N.C. Handy, Chem. Phys. Lett. **111**, 315 (1984).

[3] J. Olsen, B.O. Roos, P. Jœrgensen and H.J.Aa. Jensen, J. Chem. Phys. **89**, 2185 (1988).

[4] S. Zarrabian, C. R. Sarma and J. Paldus, Chem. Phys. Lett. **155**, 183 (1989).

[5] R.J. Harrison and S. Zarrabian, Chem. Phys. Lett. **158**, 393 (1989).

[6] J. Olsen, P. Jœrgensen and J. Simons, Chem. Phys. Lett. **169**, 463 (1990).

[7] P.J. Knowles, Chem. Phys. Lett. **155**, 513 (1989);
P.J. Knowles and N.C. Handy, J. Chem. Phys. **91**, 2396 (1989).

[8] A.O. Mitrushenkov, Chem. Phys. Lett. **217**, 559 (1994).

[9] G.L. Bendazzoli and S. Evangelisti,J. Chem. Phys. **98**, 3141 (1993);
G. L. Bendazzoli and S. Evangelisti, Int. J. Quantum Chem. Symp. **27**, 287 (1993).

[10] G.L. Bendazzoli and S. Evangelisti, Chem.Phys.Lett. **185**, 125 (1991);
S. Evangelisti and G.L. Bendazzoli, Chem. Phys. Lett. **196**, 511 (1992);
G. L. Bendazzoli, S. Evangelisti, and L. Gagliardi,Int. J. Quantum Chem. **51**, 13 (1994).
G.L. Bendazzoli and S. Evangelisti, Int. J. Quantum Chem. **55**, 347 (1995).

[11] S. Evangelisti, G.L. Bendazzoli and L. Gagliardi, Chem. Phys., **185**, 47 (1994).

[12] S. Evangelisti and G.L. Bendazzoli, Nuovo Cimento D **17**, 289 (1995);
S. Evangelisti G.L. Bendazzoli and L. Gagliardi, Int. J. Quantum Chem. **55**, 277 (1995).

[13] R. Ansaloni, S. Evangelisti, G. Paruolo and E. Rossi, Int. J. Supercomputer Applications, **6**, 351 (1992).

[14] S. Evangelisti, G.L. Bendazzoli, R. Ansaloni and E. Rossi, Chem. Phys. Lett., **233**, 353 (1995);
R. Ansaloni, E. Rossi, and S. Evangelisti, in “High Performance Computing and Networking”, B. Hertzberger and G. Serrazzi (Eds.), Lecture Notes in Computer Science, **919** (Springer, 1995).
E.Rossi, R. Ansaloni, and S. Evangelisti, in “1994 Fall CUG Proceedings, Tours, France”, Karen Winget (Ed.), 64 (1994).

[15] Cray T3D System Architecture Overview, Cray Research Inc. (1993).

[16] Application Programmer's I/O Guide, Cray Research Inc. (1995).

[17] SHMEM Technical Note for Fortran, Cray Research Inc. (1994).

[18] R.W.Numrich, P.L.Springer, J.C.Peterson, in ``High Performance Computing and Networking", W. Gentzsch and U. Harms (Eds.), Lecture Notes in Computer Science, **797** (Springer, 1994).

[19] R. K. Koeninger, in “1995 Spring CUG Proceedings, Denver, Colorado”, Bob and Karen Winget (Ed.), 93 (1995).