

## Computational Chemistry on Parallel Computers

R. J. Harrison, R. Shepard, and A. F. Wagner

Theoretical Chemistry Group, Chemistry Division, Argonne National Laboratory

The recent successful adaptation of mainline computational chemistry codes to parallel computers introduces a new era of cost-effective, computer-intensive chemistry applications and paves the way for future applications on massively parallel centralized computers being developed under the High Performance Computer and Communications Initiative. Parallel computer architecture offers the promise of inexpensive supercomputing for the price of effort in algorithm adaptations to parallelism. In Chemical Sciences-supported work by the Theoretical Chemistry Group at Argonne National Laboratory, beginning efforts at algorithm changes in computational chemistry codes has resulted in program performances on the Group's 12-processor Alliant computer superior to that on one-processor Cray X-MP or Y-MP computers. The effort so far has focused on sophisticated and highly accurate electronic structure production codes for determining the forces between atoms and molecules responsible for chemical structure, spectra, and reactivity. Some effort has also been invested in trajectory simulations of molecular dynamics. The American-made Alliant computer (model FX/2812) is one of the latest generation of shared-memory group- or division-size computers that generally cost about an order of magnitude less than the laboratory- or university-size computers such as Crays.

### DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

jr

## Computational Chemistry on Parallel Computers

Robert Harrison, Ron Shepard, and Albert Wagner  
Theoretical Chemistry Group  
Chemistry Division  
Argonne National Laboratory  
Argonne, IL 60439

Parallelism is the future of supercomputer design because of the cost savings in manufacturing many small processors which have in aggregate the power of one large processor. However, this savings can only be realized if software applications can be redesigned to evenly and continually distribute the work over all processors. The extent and success of redesign of a particular software application depends on the general nature of the parallel architecture. Two of the most convenient forms of parallelism for computational chemistry applications are shared-memory or distributed-memory multiple-instruction architectures where each processor is independently programable and has direct access to either a common memory (shared-memory model) or its own memory (distributed-memory model). The Theoretical Chemistry Group at Argonne National Laboratory has recently acquired a shared-memory multiple-instruction computer in order to carry out its dual mission of theoretical studies in combustion chemistry and studies of parallelization strategies for chemistry codes. The computer selected is an American-made, 12 processor Alliant FX/2812 computer. After initial modest efforts at software redesign, our key computational chemistry application (an electronic structure code) and several detailed models of parallel applications have been successfully ported to the FX/2812. Performance of these codes is generally comparable or superior to that obtained on single processor Cray X-MP or Y-MP computers.

The most heavily used and computationally intensive code of the Theoretical Chemistry Group is the multi-reference, singles- and doubles-excitation, configuration interaction (MRSDCI) electronic structure code called COLUMBUS. This code directly calculates the potential energy of the nuclei and electrons for a given geometrical arrangement of atoms and molecules. Variation of this energy with geometry defines the

forces that are responsible for chemical structure, spectra, and reactivity. The MRSDCI approach is particularly appropriate for describing reactive processes where bonds are broken and formed. The COLUMBUS embodiment of the MRSDCI approach has been fine-tuned for non parallel computers by an international team of collaborators headed by the ANL Theoretical Chemistry Group and is available to all chemists. The initial parallelization of this code has been completed, and the performance of four important routines in the COLUMBUS program package have been examined in detail.

The first of these routines is the evaluation of integrals over an atomic orbital (AO) basis set. (The MRSDCI method solves the partial differential Schroedinger Equation for the electronic wavefunction and energy by a basis set expansion.) For typical applications, many millions of integrals must be computed and the independence of each integral makes the application highly parallel and scalar. The initial parallelization effort produces timings with four processors that are comparable to a Cray X-MP, and with eight processors that are about twice as fast.

The second routine investigated in detail calculates the Self-Consistent Field (SCF) wavefunction, the zeroth-order wavefunction produced by COLUMBUS. This wavefunction is expressed in terms of molecular orbitals (MOs) which are the starting points for subsequent calculations of greater accuracy. Conventional SCF methods iteratively process the integrals discussed above, refining the MOs in each iteration. The SCF program whose performance was examined in detail is a direct SCF code, i.e., one that recomputes the integrals each iteration. Thus the performance characteristics are not contaminated by problems related to I/O storage and retrieval. The code was parallelized over construction of the Fock matrix ( $O(N_{\text{orb}}^4)$  work) while its diagonalization was left serial ( $O(N_{\text{orb}}^3)$  work),  $N_{\text{orb}}$  being the number of basis functions. This code is essentially scalar with  $O(N_{\text{orb}}^2)$  communication required between processes. By three processors, the timing is comparable to a one-processor Cray X-MP. By ten processors, it is three times faster.

The third routine is a Four-Index Transformation which converts the integrals from the AO basis to the MO basis ( $O(N_{\text{orb}}^5)$  work). This is a vector, as opposed to scalar, operation dominated by matrix multiplications of  $N_{\text{orb}}$  by  $N_{\text{orb}}$  matrices. There is  $O(N_{\text{orb}}^4)$  interprocessor communication, which accounts for approximately <5% of the elapsed time. The resulting code shows an approximately linear speed-up in the timings with the number of processors until about six processors. Beyond six processors, only quite modest improvements in the timings are obtained resulting in values that are about two to three times slower than a one-processor Cray. In principle, the matrix multiply operation should run at full speed on all vector computers (of which the FX/2812 is one). However, most vector code inevitably involves many memory accesses and performance is thus limited by available external memory bandwidth (to either the shared or global memory). In the FX/2800 architecture, processors are paired to a single memory port. When both processors are busy, the memory bandwidth of either processor is halved. Thus, with 12 processors, a FX/2812 can supply six processors with full access to memory. Each additional processor after six will not substantially improve timings for an application limited by access to memory, as the four-index transformation is.

The last routine examined in detail is the Configuration Interaction (CI) program. The CI step dominates the total application time for studies on larger molecules. This routine uses the integrals calculated by the four-index transformation to compute a highly accurate final wavefunction and energy. In simple terms, the algorithm corresponds to the iterative diagonalization of a large sparse matrix (too large to store), being dominated by matrix-vector and matrix-matrix products of dimension  $N_{\text{orb}}$ . The computation expense is at least  $O(N_{\text{orb}}^6)$ . The code is parallelized over segments of the expansion and result vectors. However, the act of segmenting the expansion and result vectors introduces an overhead even on one processor. More effort at algorithmic changes should eliminate much of this overhead. In this version the interprocessor communication is  $O(N_{\text{orb}}^4)$ , and is negligible compared to the computation for few processors. The timings show a flattening out of

performance after about six processors, as expected for vector-dominated applications, but the flattening out is less severe than for the four-index transformation step. By eight processors, the timing is more than twice as fast as that of a one-processor Cray X-MP.

Although most of the work has focused on parallelization of electronic structure calculations, one trajectory dynamics application has been briefly examined. By their nature, the work in the calculation of trajectories can be readily distributed to different processors and memory bandwidth is not often a problem. In the particular application examined, a simple molecular dynamics (MD) code was parallelized (but not vectorized) for the FX/2800. In an application involving the propagation of 864 argon atoms with Leonard-Jones potentials for 400 steps with periodic boundary conditions, timings indicate near linear speedup with processors. By twelve processors, the timings were superior to a one processor Cray Y-MP. This code is fully vectorized on the Cray, suggesting that vectorization on the FX/2812 could further improve the comparison.

All the codes discussed above have been parallelized using the portable message passing toolkit, TCGMSG, developed by the Theoretical Chemistry Group. This toolkit is installed on a variety of both shared-memory and distributed-memory parallel computers as well as serial computers. Thus all of the code described above can be directly run on computers of completely different character, ensuring high portability. In particular, the High Performance Computer and Communications Initiative is spurring the development of massively parallel centralized computers with a two order of magnitude increase in power over that of a single processor Cray. The Touchstone Delta project is the first example of this effort. This computer uses 512 of the same processors used in the FX/2812, only in a distributed memory format. The codes discussed should directly run on the Touchstone, as will be tested by the Group when the machine becomes available this summer.

**END**

---

**DATE**

**FILMED**

**3/30/94**

