1 Extensions on the parallel version of the MAGIC program

Chris-Kriton Skylaris
Department of Chemistry, Lensfield Road, Cambridge CB2 1EW, UK

December 1999

1.1 Introduction

MAGIC [1] is a Density Functional Theory (DFT) program that was developed as a collaborative project between BNFL and the University of Cambridge. The range of applications of the program is in the calculation of properties of molecules containing heavy atoms. Such molecules are encountered in many chemical processes of industrial importance, for example in the fabrication of new nuclear fuels or the reprocessing of spent ones.

MAGIC as a serial program was developed on Silicon Graphics workstations and this is where its first applications were run. However the need to run larger applications led to the consideration of porting MAGIC to larger, high-performance computing platforms. This task was undertaken initially by Dr Andrew Ioannou who in the autumn of 1998 ported part of MAGIC on the 256 processor Cray T3E platform of CINECA and subsequently parallelised the computationally-intensive steps. This version of MAGIC was able to perform calculations on molecules at a fixed geometry.

1.2 Parallel Implementation

The parallelisation of MAGIC has been carried out with portabilty and modularity in mind, a philosophy followed throughout its development. For this reason the functions provided by the Message Passing Interface (MPI) library [2, 3] were used for control and communication between the Processing Elements (PEs). The use of the MPI library ensures the portability of the version to other parallel computers of the Multiple Instruction Multiple Data (MIMD) type. The Single Program Multiple Data (SPMD) paradigm has been followed for the parallelisation of MAGIC. This means that each PE runs a copy of the same program *asynchronously* and the required different functionality per PE is achieved with the inclusion of branching statements. The branching statements control the tasks performed by each PE by calling MPI functions. The work is distributed to the PEs by using a *master-slave*

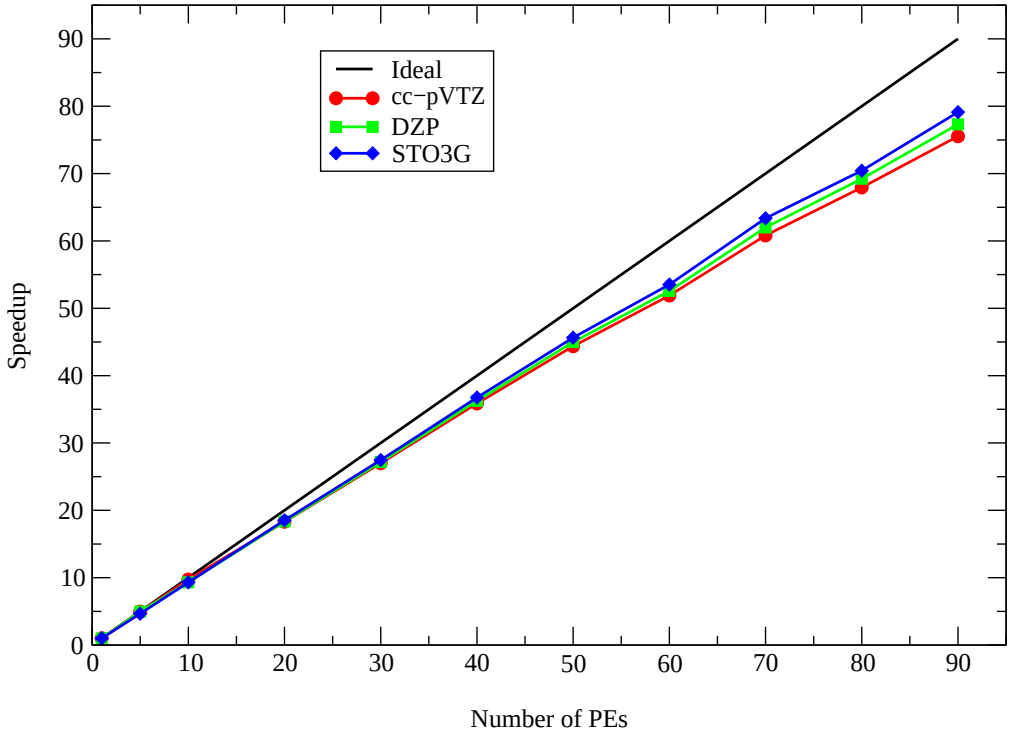
model. One PE (the master) maintains a constant and uniform feed of input data to the rest of the PEs (slaves) and collects the results as they are generated. The tasks performed and the code structure is such that the communication between PEs takes an insignificant fraction of the total time of a calculation.

The current project extends the capabilities of the original parallel version of MAGIC by the addition of the gradient of the energy and a geometry optimiser based on the BFGS method [4]. This will enable the prediction of molecular geometries, one of the most successful applications of Quantum Chemistry. The calculation of the gradient involves the calculation of derivatives of molecular integrals and represents by far the bulk of the work during a geometry optimisation. Therefore, the parallelisation effort was directed to this part and in particular to the code for the derivatives of the two-electron and DFT integrals.

1.3 Performance tests

The correctness of the parallel gradient code was tested by comparing with example results produced by the serial version of MAGIC on the Origin 2000 platform of CINECA.

The performance, or scaling of the gradient with increasing number of PEs was also examined. Here some representative results are presented for the $Si_{12}O_{11}H_{26}$ silicate structure using the uniform electron gas exchange-only functional [5] with the STO3G (189 basis functions), DZP (594 basis functions) and Dunning cc-pVTZ (1243 basis functions) basis sets. The gradient was calculated several times with various numbers of PEs. The figure shows a plot of the speedup compared to the time taken to calculate the gradient with one slave PE as a function of the number of slave PEs. It can be seen that the scaling with increasing number of PEs is close to the ideal linear case, producing a speedup between 75 and 79 when 90 PEs are used. Compared to the DZP basis set, the cc-pVTZ basis set contains more polarisation functions with up to d functions on hydrogen atoms and f functions on oxygen and silicon. It can be seen that the percentage of "linear" work in the code is high and does not change significantly by varying the size of calculation or the type of functions. This is necessary for an efficient parallel implementation and calculations with the parallel version of MAGIC should now enable the study of systems that are not possible to examine in a reasonable amount of time with the linear version.



1.4 Conclusions

A major extension to the parallel version of MAGIC that allows for optimisation of molecular geometries has been implemented. Testing of this parallel implementation with calculations of varying size showed near linear scaling. Calculations on molecules much larger than those possible with a serial version in a reasonable amount of time are now feasible, increasing the applicability of the code to more chemical processes. Studies of large inorganic systems are underway, in close collaboration with Dr Andrew Willetts from the University of Cambridge and Professor Paolo Palmieri and Dr Laura Gagliardi from the University of Bologna.

1.5 Acknowledgements

I am grateful to CINECA for the award of an ICARUS grant and to Professor Paolo Palmieri for his great interest in this project. Stimulating discussions with Sigismondo Boschi, Paolo Malfetti, Elda Rossi and other members of CINECA are gratefully acknowledged.

References

[1] A. Willetts, L. Gagliardi, A. G. Ioannou, A. M. Simper, C.-K. Skylaris, S. Spencer and N. C. Handy, Int. Rev. Phys. Chem. in press (1999).

[2] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard*, vesrion 1.1 June 12, 1995. Available by anoynymous ftp from ftp.mcs.anl.gov.

[3] P. S. Pacheco, *Parallel programming with MPI*, Morgan Kaufmann, (1997).

[4] J. D. Head and M. C. Zerner, Chem. Phys. Lett. 122 (1985) 264.

[5] R. G. Parr and W. Yang, *Density-Functional Theory of Atoms and Molecules*, Oxford University Press, (1989).