

Bringing *ab initio* Electronic Structure Calculations to the Nano Scale through High Performance Computing

James Currie

Rachel Cramm Horn

Paul Rulis

University of Missouri – Kansas City University of Missouri – Kansas City University of Missouri – Kansas City
 Department of Physics and Astronomy Department of Physics and Astronomy Department of Physics and Astronomy
 5110 Rockhill Road 5110 Rockhill Road 5110 Rockhill Road
 Kansas City, MO, 64110 Kansas City, MO, 64110 Kansas City, MO, 64110
 1-816-235-2501 1-816-235-2501 1-816-235-5945
 jecyrd@mail.umkc.edu rec44f@mail.umkc.edu rulis@umkc.edu

ABSTRACT

An *ab initio* density functional theory based method that has a long history of dealing with large complex systems is the Orthogonalized Linear Combination of Atomic Orbitals (OLCAO) method, but it does not operate in parallel and, while the program is empirically observed to be fast, many components of its source code have not been analyzed for efficiency. This paper describes the beginnings of a concerted effort to modernize, parallelize, and functionally extend the OLCAO program so that it can be better applied to the complex and challenging problems of materials design. Specifically, profiling data were collected and analyzed using the popular performance monitoring tools TAU and PAPI as well as standard UNIX time commands. Each of the major components of the program was studied so that parallel algorithms that either modified or replaced the serial algorithm could be suggested. The program was run for a collection of different input parameters to observe trends in compute time. Additionally, the algorithm for computing interatomic interaction integrals was restructured and its performance was measured. The results indicate that a fair degree of speed-up of even the serial version of the program could be achieved rather easily, but that implementation of a parallel version of the program will require more substantial consideration.

Categories and Subject Descriptors

D.2.8 [Software Engineering]: Metrics – *performance measures*.

General Terms

Algorithms, Measurement, Performance.

Keywords

Density functional theory, atomic orbitals

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright ©JOCSE, a supported publication of the Shodor Education Foundation Inc.

1. INTRODUCTION

Advanced materials have played a pivotal role in recent technological progress, often causing the demand for designer characteristics or novel properties to outpace our ability to understand these complex materials at a fundamental level. This pressure to master materials at the nanoscale has pushed forward the development of many theoretical approaches and the implementation of many computational methods. A particular area of interest includes structures with defects that are on the order of 10 nanometers in size because many bulk structural and electronic properties of materials are dominated by the properties of the defect. Density functional theory (DFT) based approaches represent the current state of the art for the application of theory to materials problems that require both high accuracy and high efficiency. This is a position somewhat between the larger scale molecular dynamics methods and the smaller scale but often more accurate quantum chemical methods. Some of the issues that DFT is well suited to deal with include catalysis processes [15], configuration of ultra-dilute dopants in crystal structures [27], and the determination of the tensile strength of bioceramics [5]. DFT is a computational, quantum mechanical framework for the modeling of materials and it is being actively applied with much success across a wide breadth of fields within a growing number of scientific domains. Density functional theory was created in a sequence of two papers by Hohenberg and Kohn [14] and Kohn and Sham [17]. They presented the method as one that reduces the problem of determining the many-body ground state wave function to one of determining only the charge density. For a system of N interacting electrons this reduces the problem from a space of $3N$ dimensions down to a space of just three. A type of mean-field approach is used that solves a one-electron problem where the potential is derived from the charge density distribution of all the electrons in the system. A self-consistent field (SCF) cycle iterates through determination of the wave-function, the charge density distribution, and the potential until there is no change in these terms. Since its original inception, much work has been done to enhance this method, and a review of the theory can be found in a paper written by Peter Blöchl [1]. Presently, there exist a variety of different implementations of DFT that can be divided into a few prominent camps based on the choice of basis functions used to expand the system wave function, the representation of the potential function, and the representation of the charge density [20]. Each of the main approaches to DFT has its own set of advantages and disadvantages that make it particularly applicable to one range of materials and problems or another.

Although DFT provides an incredible simplification of the quantum mechanical many-body problem while still retaining excellent accuracy, the types of problems that are of the most interest still require prohibitive amounts of time for even the fastest computer processor to solve. Hence, parallel processing has become an invaluable tool and many DFT program codes have been adapted to take advantage of this high performance computing (HPC) capability. Interestingly though, the programming style required to develop parallel algorithms is significantly different from the approach of serial algorithms which sometimes makes it difficult to parallelize an existing code and gain as much efficiency as is desired. Therefore, when parallelizing an existing serial application, performance analysis of the existing algorithm lends some helpful insight about which sections of a program are the most computationally expensive and why. This could be used to determine whether the algorithm should be simply modified for a parallel execution environment or if it needs to be totally rewritten.

This paper describes the beginnings of a concerted effort to modernize, parallelize, and functionally extend a particular DFT based program so that it can be applied to the complex and challenging problems of materials design. Specifically, profiling data was collected and analyzed using widely available and portable external libraries and standard UNIX/Fortran time commands. Then, each of the major components of the program was studied so that parallel algorithms could be suggested that either modified or replaced the serial algorithm.

2. METHODS

The primary focus of this development work is the Orthogonalized Linear Combination of Atomic Orbitals (OLCAO) method [29]. This is a density functional theory based method that uses Gaussian based atomic orbitals in the solid state wave function expansion and atom centered Gaussian functions for an analytical description of the potential and charge density distribution functions. OLCAO has found particular application in the study of the electronic structure, bonding, and spectroscopic properties of large and complex materials systems ranging from amorphous solids [9–11, 16] and complex crystals [3, 18, 19, 30] to those containing large scale structures such as grain boundaries (GBs) [6, 21, 23], intergranular glassy films (IGFs) [7, 8, 25], and passive defects [4, 24]. Of particular interest to the development work that was started as a part of the Blue Waters – Undergraduate Petascale Education Program (BW-UPEP) Internship is the capability of the OLCAO method with regard to performing core level spectroscopic calculations such as x-ray absorption near edge structure or electron energy loss near edge structure (XANES/ELNES) [12, 13, 24, 27]. An extension of the normal spectral calculation is being developed within OLCAO whereby spectra are computed for every atom in a model and then brought together to form an image that correlates spectral features with atomic structure [24]. This spectral imaging technique will be quite computationally intensive and while the OLCAO program is efficient and capable of being used to compute the XANES/ELNES spectra of rather large systems it is still a serial application and thus the calculation times can be quite lengthy, sometimes lasting more than a few days. There are two key mathematical operations that are performed in OLCAO. The first is the analytic calculation of a set of integrals between atomic orbitals in various forms given for s-type orbital in Equations 1-4.

$$\langle s_A | s_B \rangle = \int e^{-\alpha_1 \vec{r}_A^2} e^{-\alpha_2 \vec{r}_B^2} d\vec{r} \quad (1)$$

$$\langle s_A | -\nabla^2 | s_B \rangle = \int e^{-\alpha_1 \vec{r}_A^2} (-\nabla^2) e^{-\alpha_2 \vec{r}_B^2} d\vec{r} \quad (2)$$

$$\left\langle s_A \left| \frac{1}{\vec{r}_C} e^{-\alpha \vec{r}_C^2} \right| s_B \right\rangle = \int e^{-\alpha_1 \vec{r}_A^2} \frac{1}{\vec{r}_C} e^{-\alpha_3 \vec{r}_C^2} e^{-\alpha_2 \vec{r}_B^2} d\vec{r} \quad (3)$$

$$\left\langle s_A \left| e^{-\alpha \vec{r}_C^2} \right| s_B \right\rangle = \int e^{-\alpha_1 \vec{r}_A^2} e^{-\alpha_3 \vec{r}_C^2} e^{-\alpha_2 \vec{r}_B^2} d\vec{r} \quad (4)$$

Equation 1 represents the overlap of two s-type Gaussian orbitals. Equation 2 also shows integration of s-type Gaussians but with the Laplacian operator for the computation of the kinetic energy. Equation 3 accounts for the contribution of the nuclear interaction to the total potential. Equation 4 is a three center integral between s-type Gaussians that is used for determining the Coulombic electron – electron contribution to the total potential. Higher angular momentum integrals can be derived from these equations via repeated differentiation making progressively more complicated formulas [29]. The second major operation is the processes of solving the eigenvalue problem that obtains the wave function expansion coefficients and associated energy eigenvalues. This requires the complete diagonalization of a large matrix. Some key parameters that affect the cost of these operations are the number of atoms in the system, the number of basis functions used for each atom (typically identified as either a minimal, full, or extended basis), the number of k-points (Brillouin zone integration sampling points), and the number of terms used to describe the potential function. Parallelization of these operations could significantly decrease the calculation time or, if the calculation time is maintained, it would allow for the study of much larger systems.

In addition to the time stamps given by OLCAO, we used tools such as the Tuning and Analysis Utilities (TAU) [26] and the Performance Application Protocol Interface (PAPI) [2]. TAU and PAPI exist as a set of library function calls that are instrumented into the source code of a program to track a wide variety of information ranging from simple subroutine runtimes to the number of times that specific CPU events, such as a cache miss or an execution branch, occur. The TAU package is described as a “portable profiling and tracing toolkit for performance analysis of parallel programs.” [28] However, it does have particular uses in the analysis of serial programs as well. PAPI, similar to TAU, is a portable kit that accesses hardware performance counters physically present on modern microprocessors. The aim of PAPI is “to see, in near real time, the relation between software performance and processor events.” [22] Memory access overhead and branching performance were of particular interest in this study to understand where bottlenecks in OLCAO may exist. Memory access overhead is a fundamental problem, especially for HPC, because the processors often operate at data rates that are much faster than the data transfer rates to memory so that if the CPU is not well supplied with data it will sit idle. Understanding the cache performance of a program can help the programmer reorganize data structures to provide more efficient memory access. Conditional branch prediction is another important area because modern CPUs maintain a pipeline of operations and when a branch is mispredicted the pipeline must be flushed and refilled at substantial cost to efficiency. These were the primary tools that were used to explore the characteristics and efficiency of the algorithms in the OLCAO program suite.

The first step to parallelizing OLCAO is to develop a base line understanding of its serial execution performance in terms of the

compute time for different sections of the code under different conditions. To do this, a series of different materials systems were selected and parameters that control the computational cost of the calculation were varied systematically. Two machines were used to collect the performance data, a local workstation using the commercially available AMD Phenom II X6 1090T processor and the Pittsburgh Supercomputing Center's Blacklight machine which uses the Intel Xeon X7560 processor. Smaller system calculations were performed on the local workstation while the larger system calculations made use of Blacklight. The performance trends were observed using a variety of tools. Specifically, timing data for different sections of the OLCAO program were collected first by using Fortran write statements that recorded the current time. This gave an initial understanding of the effects of parameter changes and did not require any modifications to the program code because such write statements already existed. Then, particular sections of the code were identified and studied with TAU and PAPI because they are more fine grained tools for understanding the reason why a particular algorithm behaves as it does. The data that was collected using TAU and PAPI included the rate of branching misprediction for the combination of different compiler flags and changes in the algorithm for a particular section of code. The code in the select section was restructured with the intent of reducing the number of conditional statements encountered overall.

3. RESULTS AND DISCUSSION

The profiling techniques described in section 2 were applied to the OLCAO program for three different material systems with a systematic variation of parameters. The systems used in the study were a 907 atom model of an IGF within crystalline β - Si_3N_4 , a ten base pair periodic model of DNA with 650 atoms (computed on Blacklight), and a series of supercells of pure Al (computed with a local workstation). Details of the key structural parameters of each system are provided in **Table 1** and an illustration of each system is provided in **Figure 1**.

Table 1: Crystal Structure Details

	IGF	DNA	Al Full Cell
a (Å)	14.533	30.000	4.050
b (Å)	15.225	30.000	4.050
c (Å)	47.420	39.208	4.050
α (°)	90	90	90
β (°)	90	90	90
γ (°)	90	90	90
# of Atoms	907	650	4
# of Electrons	4288	2220	12
Matrix	9111	4740	52
Dimension	(Full Basis)	(Full Basis)	(Full Basis)
Elements	Si, N, O	C, H, N, Na, O, P	Al

The IGF and DNA material systems were specifically chosen as representatives of particular classes of materials that are of current research interest and therefore represent the types of systems likely to be encountered by the OLCAO program in practice. Also, these systems possess specific features that can make their comparison helpful. The IGF and DNA models both have a relatively large number of atoms but the dimension of the

interaction matrices for the IGF and the number of electrons is larger in it compared to the DNA model by a factor of about two. The dimension is larger because the Si in the IGF includes unfilled 3d atomic orbitals in its basis while the DNA model has no atoms with 3d orbitals in its basis. The number of electrons is larger because the IGF contains no H and thus every atom has more electrons. For the IGF and DNA models it is also possible to easily and realistically alter the number of independent terms in the potential function representation. This is done by changing the threshold criteria for how similar two atoms need to be with respect to their local environment before they can share the same potential function values. This "sharing of values" between potential sites means that fewer terms are used in the potential function and thus that fewer independent interaction matrices need to be created. This issue is a particular characteristic of the representation of the potential function in OLCAO and is not generally applicable to all DFT based methods.

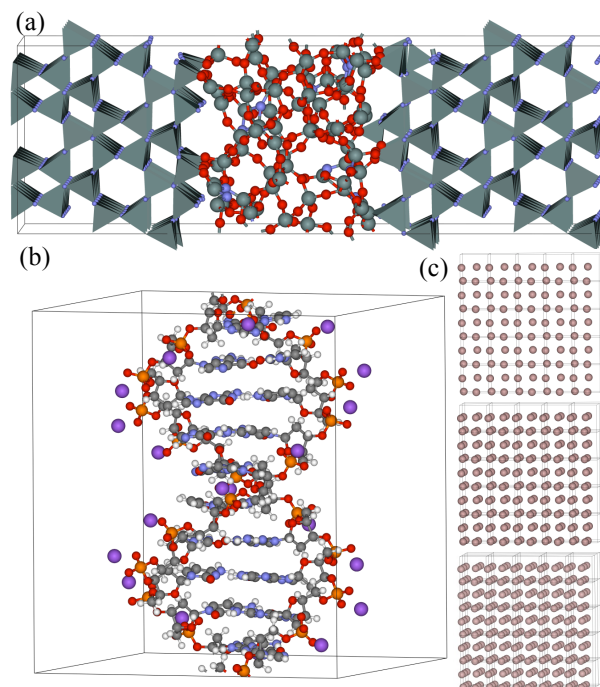


Figure 1: Ball and stick models of the material systems studied. (a) An intergranular glassy film model in β - Si_3N_4 ; (b) A ten base pair periodic model of DNA with Na counter ions; (c) A sequence of three crystalline Al supercells $5 \times 5 \times 1$, $5 \times 5 \times 2$, and $5 \times 5 \times 3$.

The first level of analysis was done with a simple measurement of the amount of time that different segments of the OLCAO program took to run calculations on the IGF model and the DNA model when the number of terms in the potential function was modified. In particular, two key sections, identified as "Setup" and "Main," were analyzed to find the most time expensive parts of their code. The "Setup" (integrals, electrostatics, exchange-correlation) and "Main" (Secular Equation [preparation, solution], and everything else) programs comprise the self-consistent field (SCF) implementation of OLCAO. These programs and their components are illustrated schematically in **Figure 2**.

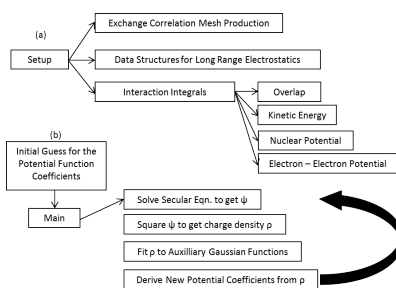


Figure 2: Schematic of the components of the SCF portion of the OLCAO program. (a) Setup; (b) Main.

The runtime analysis of these components is shown in **Figure 3**. In the IGF system, the ratio of the calculation time for the integrals compared to the rest of the setup calculation is quite large (**Figure 3a**). We also see that the long-range coulomb and exchange correlation calculations are almost unaffected by the number of potential terms, with the majority of the time being spent with the interaction integrals. In the sections of main we see that as the number of terms increases the preparation time of the secular equation quickly grows to be much larger than the time it takes to solve the secular equation. In this case the preparation is simply the task of reading the packed matrix data stored on disk by setup, unpacking it, and applying a coefficient to each matrix before accumulating it.

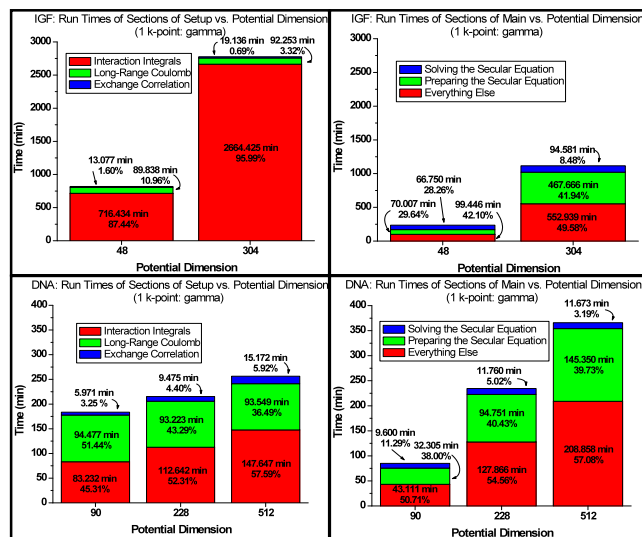


Figure 3: Illustration of the execution time for the IGF and DNA models for various sections of the OLCAO Setup and Main programs and various number of potential terms as measured by Fortran time records. (a) Setup and IGF; (b) Main and IGF; (c) Setup and DNA; (d) Main and DNA.

When the same type of analysis is done for the DNA system, as shown in **Figures 3c and 3d**, we see a similar trend to that of the IGF system, where the interaction integrals grow more costly as the number of potential terms increases. Also, the long-range Coulomb calculations have more of a presence here due to the larger cell size and thus a larger number of reciprocal space cells needed for the convergence of the Ewald summation series. Considering the run length of the program “Main”, the trend is similar to the case of the IGF. The similarity of the trend is

expected, but a comparison of the two systems presents some confusion. The total time of even the longest run of Setup for the DNA model (about 260 minutes with 512 terms in the potential function) is less than the shortest IGF model run with only 48 terms in the potential function (about 815 minutes). This is likely due in large part to the larger size of the interaction matrix for the IGF model (i.e. it has many more basis functions), but there may also be something more subtle at work. That is, the number of nearest neighbors for a given atom in the DNA model is, on average, less than that of the IGF model because DNA is a molecule with exposed surface boundaries and lots of H. This will substantially reduce the number of interaction integrals that will need to be calculated resulting in an interaction matrix that is not only smaller, but also sparser.

The pure AI model was chosen because of its simplicity and for the fact that we could vary specific parameters with less worry about how the innate characteristics of the model might mask the effect of a change in other parameters such as the number of k-points or the choice of a full, minimal, or extended basis. The results of the timing runs for a constant number of terms in the potential function are shown in **Figure 4**.

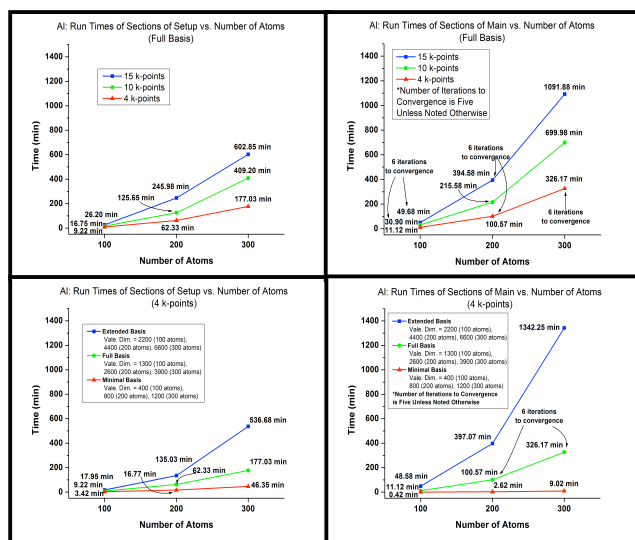


Figure 4: Illustration of the execution time for calculations of various sections of the OLCAO Setup and Main programs for three AI supercell models with variation in the number of k-points and the size of the basis set. (a) Setup and k-points; (b) Main and k-points; (c) Setup and basis size; (d) Main and basis size.

In the study of the AI runtime performance we see that OLCAO is more sensitive to changes in the basis than to increases in the number of k-points. When k-points are added the effect is multiplicative so that doubling the number of k-points effectively doubles the cost of the calculation while when the basis size is increased a non-linear trend is observed. For the 300 atom case, an approximate doubling of the basis size from full to extended more than doubles the cost of the calculation for both Setup and Main. This is expected because the basis size affects the interaction integrals matrix size which scales as the square of the dimension. This is clear in the “Main” section because it shows an increase in calculation time for the extended basis that is approximately four times greater than for the full basis. In setup this change was much less exaggerated with only about a three

times increase. One important note is that as the system size increases the number of k-points needed for a high resolution calculation decreases. Because OLCAO is typically used for large and complex systems, the number of k-points is often just equal to one.

Beyond the crude timing data, a series of calculations were performed to study the performance of some of the most costly components of the program using TAU and PAPI. The multicenter interaction integrals take a significant portion of the overall time which is typical for all atomic orbital based methods. Another component that is more specific to the OLCAO program is the orthogonalization procedure. This modifies the resultant interaction matrix to force the valence orbitals to be orthogonal to the core orbitals. For large systems this could be a costly step because a sequence of matrix-matrix multiplications is required. The program code for computing those integrals and for doing the orthogonalization has performed well for decades, but it has also not been evaluated for efficiency in just as long. Hence a concerted effort is underway to evaluate and possibly improve this aspect of the program.

For the analysis of the OLCAO program using TAU and PAPI a different set of Al supercells was used. They were $1 \times 1 \times 1$, $2 \times 2 \times 2$, and $3 \times 3 \times 3$ supercells of the full cell so that the models had 4, 32, and 108 atoms respectively. The goal of the orthogonalization subroutine analysis was simply to count the number of branches encountered by the program for a given compiler optimization level. The collected data is shown in **Table 2**. For level two compiler optimization (obtained using the -O2 compiler flag) the branch misprediction rate was slightly better than that obtained with level three optimization. However the total number of branches encountered by the program under level two optimization was significantly greater than that encountered under level three optimization. This result may have been expected, but an unusual second result was that the misprediction rate for the $2 \times 2 \times 2$ supercell for both levels of optimization was significantly higher than for both of the other two supercells. This exercise indicates that interpreting the results of higher level code analysis can have important subtleties that can be easily overlooked.

Table 2: Al Supercell Orthogonalization Subroutine Branching Data

Cell and Optimization	Miss-Predicted (%)	Correctly Predicted (%)	Total
1x1x1 -O2	3.92	96.06	7748064
1x1x1 -O3	3.97	96.03	7749479
2x2x2 -O2	6.52	93.48	62052410
2x2x2 -O3	6.81	93.19	59604903
3x3x3 -O2	3.01	96.99	1222164262
3x3x3 -O3	3.86	96.14	957394561

The analysis of the integration subroutine was also performed as a comparison between level two and level three compiler optimization, and it was also performed for the case of a modification in the algorithm versus the unmodified algorithm. The essential modification is that depending on the particular atoms it may be necessary to perform only s-type with s-type integration, or perhaps s-type with p-type or only up to p-type with p-type. In other cases the integral may need to include all the way up to the complicated d-type with d-type integral. The

subroutine that performs the integrals checks along the way to determine which integrals to do, but it was observed that this sequence of “if-else” blocks in the code was repetitive such that if the section was rewritten one block could replace a sequence of three or four. This replacement option was present multiple times in the algorithm. The data from the sequence of TAU and PAPI runs are shown in **Tables 3 through 6**. The different types of integrals correspond to those given in Equations 1 through 4 plus all of the similar integrals of the higher angular momentum orbitals. Again, the trend is clear. The higher level of optimization had a higher percentage of branch miss-predictions, but the total number of branches was significantly less. When the comparison is between the old and the new algorithm the total number of branches drops in all cases, but again the branch miss-prediction percentage increases.

Table 3: Original Integration Subroutine With -O2 Optimization.

Integral Type	Miss-Predicted (%)	Correctly Predicted (%)	Total
1	1.00	99.00	53795827
2	1.09	98.91	67004191
3	1.44	98.58	208187973
4 (Avg.)	1.35	98.65	310677930

Table 4: Original Integration Subroutine With -O3 Optimization.

Integral Type	Miss-Predicted (%)	Correctly Predicted (%)	Total
1	1.95	98.08	9767131
2	2.58	97.42	9766631
3	2.37	97.63	63389710
4 (Avg.)	2.04	97.96	66161685

Table 5: Modified Integration Subroutine With -O2 Optimization.

Integral Type	Miss-Predicted (%)	Correctly Predicted (%)	Total
1	0.92	99.08	53415195
2	0.99	99.01	66378834
3	1.16	98.84	207903892
4 (Avg.)	1.24	98.76	310464531

Table 6: Modified Integration Subroutine With -O3 Optimization.

Integral Type	Miss-Predicted (%)	Correctly Predicted (%)	Total
1	1.96	98.04	8988970
2	2.75	97.25	9140602
3	2.92	97.08	60653657
4 (Avg.)	2.40	97.60	57420935

4. CONCLUSION

The sections of the OLCAO program associated with the SCF calculation were analyzed for their performance characteristics using both simple timers and more complicated instrumented performance monitoring tools. The results confirm some previously held beliefs about the relative computational cost of various components of the setup and main programs and also introduced some new and as yet unexplained results. Clear results include the fact that the computation of the interaction integrals is generally the most expensive but that preparing the Ewald summation data structures can become time consuming when the cell size becomes large. The overall cost of the interaction integrals calculation scales primarily with the number of terms in the potential function. The calculation cost scaled linearly with an increase in the number of k-points while it increased super-linearly (quadratically) with an increase in the size of the basis. Generally, the timing data was observed to be crude but it was also good at giving a big picture of the behavior of the program and so a key conclusion is that sometimes it is not necessary to apply highly sophisticated methods to get a good initial understanding. However, it is often difficult to totally isolate one particular variable, especially when attempting to model realistic calculation performance. This may be because multiple effects are correlated or because the model under study has characteristics that are not apparent simply in terms of the number of atoms, potential terms, matrix dimension, etc. This was exemplified by the smaller average atom density in the DNA model compared to the IGF model. The conclusion is that extreme care must be taken when interpreting any profiling results as it may be possible for subtle effects to skew the data.

Additional profiling data was obtained for the specific subroutines that involve the interaction integrals computed in OLCAO. The results indicated that even a simple attempt at code restructuring was able to produce noticeable results. We can conclude from this that while the algorithm may be fast because of its analytic nature, it is quite likely that significant improvements can be obtained by implementing a better approach to the calculation. The reasoning is that if even a straightforward modification can produce such dramatic results, then it is likely that a deeper degree of consideration will produce more substantial results. This also demonstrates that even though the profiling mission was to set the stage for later parallelization, it is entirely possible that the analysis will spark a deeper understanding of a subroutine or algorithm that will remain entirely serial but which may have a significant effect on the overall efficiency.

Another important conclusion that can be drawn from this study is that good profiling must come from the intelligent application of multiple tools, both crude and advanced. The fine grained advanced tools can often help provide the insight needed to understand why a particular algorithm behaves the way it does, but crude measurements can provide the appropriate context in which to interpret that data. It is not possible to naively apply a powerful tool kit and expect it to do all the work and provide clear results. Rather, deep consideration is required to sort out the significance of the results for the variety of probing techniques.

5. ACKNOWLEDGMENTS

Special thanks to SHODOR, the National Computational Science Institute (NCSI), the National Center for Supercomputing Applications (NCSA), and everyone else involved in the Blue Waters Undergraduate Petascale Education Program (BW-UPEP).

The BW-UPEP program supports a number of undergraduate students working on a diverse range of computational science problems. Support includes a two week intensive workshop at the NCSA facility on the University of Illinois – Urbana Champaign campus, pairing with a faculty advisor, a stipend, and a forum for the student to attain help if needed throughout the internship.

The BW-UPEP program has given me (J.C.) the chance to experience a number of things I might not otherwise have been exposed to. The two week workshop was particularly effective in teaching me the tools I would need to effectively carry out my research. Specifically, I learned how to develop parallel algorithms using MPI, OpenMP, and CUDA. Particular to my individual project, I learned how to use many technical tools like TAU and PAPI, and the important role that these tools can play while trying to improve a computationally intensive application. I was also given the opportunity to attend two conferences, Teragrid '11 (now XSEDE) and SC '11. I gave poster presentations at both conferences and this provided me with important experiences such as how to create and give scientific presentations. Furthermore, in attending the conferences I was able to meet and interact with members of the broader HPC community which was very inspirational. The experience as a whole has persuaded me to pursue a career in computational science.

Finally thanks to XSEDE and the Pittsburgh Supercomputing Center for providing access to the resources that were used to do this project.

6. REFERENCES

- [1] Blöchl, P.E. 2011. Theory and Practice of Density-Functional Theory. *arXiv:1108.1104*. (Aug. 2011).
- [2] Browne, S. et al. 2000. A Portable Programming Interface for Performance Evaluation on Modern Processors. *International Journal of High Performance Computing Applications*. 14, 3 (Aug. 2000), 189–204.
- [3] Caruso, A.N. et al. 2009. Direct evidence of electron spin polarization from an organic-based magnet: [Fe^{sup II}](TCNE)(NCMe)[sub 2]][Fe^{sup III}]Cl[sub 4]]. *Physical Review B (Condensed Matter and Materials Physics)*. 79, 19 (2009), 195202–5.
- [4] Chen, Y. et al. 2002. Ab-initio calculation of Si-K and Si-L ELNES edges in an extended inactive defect model of crystalline silicon. *Materials Transactions*. 43, 7 (2002), 1430–1434.
- [5] Ching, W.Y. et al. 2009. Ab initio elastic properties and tensile strength of crystalline hydroxyapatite. *Acta Biomaterialia*. 5, 8 (2009), 3067–3075.
- [6] Ching, W.Y. et al. 2006. Ab initio modeling of clean and Y-doped grain boundaries in alumina and intergranular glassy films (IGF) in β -Si₃N₄. *Journal of Materials Science*. 41, 16 (2006), 5061–5067.
- [7] Ching, W.Y. et al. 2009. Ab initio tensile experiment on a model of an intergranular glassy film in β -Si₃N₄ with prismatic surfaces. *Applied Physics Letters*. 94, 5 (2009), 051907–3.
- [8] Ching, W.Y. et al. 2010. Theoretical study of the elasticity, mechanical behavior, electronic structure, interatomic bonding, and dielectric function of an intergranular glassy film model in prismatic β -Si₃N₄. *Physical Review B*. 81, 21 (Jun. 2010), 214120.

- [9] Ching, W.Y. 1982. Theory of amorphous SiO_2 and SiO_x . I. Atomic structural models. *Physical Review B*. 26, 12 (Dec. 1982), 6610–6621.
- [10] Ching, W.Y. 1982. Theory of amorphous SiO_2 and SiO_x . II. Electron states in an intrinsic glass. *Physical Review B*. 26, 12 (Dec. 1982), 6622–6632.
- [11] Ching, W.Y. 1982. Theory of amorphous SiO_2 and SiO_x . III. Electronic structures of SiO_x . *Physical Review B*. 26, 12 (Dec. 1982), 6633–6642.
- [12] Ching, W.Y. and Rulis, P. 2008. Ab initio calculation of the O-K, N-K, Si-K, Si-L3, Y-K, Y-L3 edges in the Y-Si-O-N system: A strategy for ELNES/XANES spectral modeling in complex materials. *Physical Review B Condensed Matter and Materials Physics*. 77, 3 (2008), 035125/1–035125/17.
- [13] Ching, W.Y. and Rulis, P. 2008. Large differences in the electronic structure and spectroscopic properties of three phases of AlPO_4 from ab initio calculations. *Physical Review B Condensed Matter and Materials Physics*. 77, 12 (2008), 125116/1–125116/7.
- [14] Hohenberg, P. and Kohn, W. 1964. Inhomogeneous Electron Gas. *Physical Review*. 136, 3B (1964), B864.
- [15] Honkala, K. et al. 2005. Ammonia Synthesis from First-Principles Calculations. *Science*. 307, 5709 (Jan. 2005), 555–558.
- [16] Huang, M.-Z. and Ching, W.Y. 1994. Electronic and transport properties of perfect sp^2 -bonded amorphous graphitic carbon. *Physical Review B*. 49, 7 (1994), 4987.
- [17] Kohn, W. and Sham, L.J. 1965. Self-Consistent Equations Including Exchange and Correlation Effects. *Physical Review*. 140, 4A (1965), A1133.
- [18] Liang, L. et al. 2010. Mechanical properties, electronic structure and bonding of a- and b-tricalcium phosphates with surface characterization. *Acta Biomaterialia*. 6, 9 (2010), 3763–3771.
- [19] Liang, L. et al. 2009. Theoretical study of the large linear dichroism of herapathite. *Physical Review B (Condensed Matter and Materials Physics)*. 80, 23 (2009), 235132–5.
- [20] Martin, R.M. 2004. *Electronic Structure: Basic Theory and Practical Methods*. Cambridge University Press.
- [21] Mo, S.-D. et al. 1999. Electronic structure of a grain-boundary model in SrTiO_3 . *Physical Review B*. 60, 4 (1999), 2416.
- [22] PAPI: <http://icl.cs.utk.edu/papi/>. Accessed: 2012-05-30.
- [23] Rulis, P. et al. 2004. Ab initio ELNES/XANES spectral calculation of polar and non-polar grain boundaries in β - SiC . *Acta Materialia*. 52, 10 (2004), 3009–3018.
- [24] Rulis, P. et al. 2009. Spectroscopic imaging of electron energy loss spectra using *ab initio* data and function field visualization. *Ultramicroscopy*. 109, 12 (2009), 1472–1478.
- [25] Rulis, P. and Ching, W. 2011. Theoretical ELNES spectra of Si-K, Si-L, N-K, and O-K edges of an intergranular glassy film model in β - Si_3N_4 . *Journal of Materials Science*. 46, 12 (2011), 4191–4198.
- [26] Shende, S.S. and Malony, A.D. 2006. The Tau Parallel Performance System. *International Journal of High Performance Computing Applications*. 20, 2 (May. 2006), 287–311.
- [27] Tanaka, I. et al. 2003. Identification of ultradilute dopants in ceramics. *Nat Mater*. 2, 8 (2003), 541–545.
- [28] TAU - Tuning and Analysis Utilities: <http://www.cs.uoregon.edu/Research/tau/home.php>. Accessed: 2012-05-30.
- [29] Wai-Yim Ching and Paul Rulis 2012. *Electronic Structure Methods for Complex Materials: The Orthogonalized Linear Combination of Atomic Orbitals*. Oxford University Press.
- [30] Zhong, X. and Ching, W.Y. 1990. Calculation of local orbital moments of conduction electrons in $\text{Nd}_2\text{Fe}_{14}\text{B}$. *Journal of Applied Physics*. 67, 9 (May. 1990), 4768–4770.