Introducing the MADNESS numerical framework for petascale computing

W. Scott Thornton, Nicholas Vence, and Robert Harrison *University of Tennessee*

ABSTRACT: We discuss how MADNESS (multiresolution adaptive numerical environment for scientific simulation) attempts to address issues of complexity, efficiency, and scalability with special emphasis on applications in chemistry and physics.

KEYWORDS: density functional theory, time evolution, multiresolution, wavelets, task parallelism

1 Introduction

The number of applications and disciplines benefiting from the nation's investment in massively-parallel computers has shrunk significantly as we have progressed from giga- to tera- to peta-scale computation. Certainly, some sub-fields of chemistry and physics are satisfied by the power of modern desktops or small clusters, but many others have pressing needs for high-end computation and yet are struggling to field their applications at the tera- and peta-scales. Advances in these fields are being inhibited by the lack of functionality on the largest computers and the time lag between theoretical innovation in small research groups and its realization in widely available, state-of-the-art codes.

One particular motivation for this work is referred to by Colella as the *semantic gap*. Why is it that we can usually state our problem in a few lines or pages of equations and yet its computational realization in software is so much larger? Much scientific discovery now takes place at the interfaces between disciplines, which leads to the most demanding present applications now being multi-physics. The intrinsic complexity of such simulation is in addition to the complexity inherent to the parallel computing platform, and the changing nature of scientific programming (no longer just FORmula TRANslation) is also reflected in shifts from regular data structures with conventional dense or sparse solvers to irregular data structures with fast hierarchical solvers. Finally, Each additional scale of parallelism multiplies the development time and cost by a large factor.

Below, we discuss how MADNESS attempts to address many of the above issues with special emphasis on applications in chemistry and physics.

2 Fast computation in many dimensions

MADNESS (multiresolution adaptive numerical environment for scientific simulation) [Harrison04, Yanai04, Beste06, Sekino08] started with support from the DOE as an environment for fast and accurate numerical simulation in chemistry but rapidly expanded to include applications in nuclear physics (HF and DFT for nuclei) and atomic and molecular physics (time-evolution of few-electron systems in intense laser fields). It is now running at full scale on the NSF and DOE Cray supercomputers at UT/ORNL, and elsewhere, and is open-source under the GPL2, hosted on Google.

For many applications the enabling numerical capability is the application as a single sparse-matrix-vector-product the inverse of many physically important differential operators (i.e., Green's functions). This eliminates the need for preconditioned iterative solvers and builds in the correct asymptotic form of the solution.

From the outset MADNESS emphasized a high-level environment for the composition of massively parallel science applications. For instance, expectation value of a Hamiltonian

$$E=\langle\psi|-\frac{1}{2}\nabla^2+V|\psi\rangle+\int \psi^2(x)\frac{1}{|x-y|}\psi^2(y)\,dx\,dy$$

is compactly expressed as the following code fragment that with appropriate headers and typedefs will compile and execute in parallel (*k* is the order of the underlying numerical approximation, *rlo* (i.e., r_{lo}) is the shortest lengthscale to be resolved, and *thresh* is the desired precision).

```
operatorT op = CoulombOperator(k,
rlo, thresh);
functionT rho = psi*psi;
double twoe =
inner(apply(op,rho),rho);
double pe = 2.0*inner(Vnuc*psi,psi);
double ke = 0.0;
for (int axis=0; axis<3; axis++) {
    functionT dpsi = diff(psi,axis);
    ke += inner(dpsi,dpsi);
}
double energy = ke + pe + twoe;
```

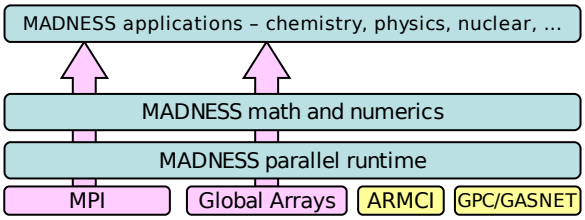
Vector and matrix APIs enable expression of coarser levels of parallelism, and each MADNESS operation can be modified to run without waiting for completion to permit overlap of multiple unrelated operations.

The guaranteed speed and precision arising from dynamic adaptive refinement and fast algorithms based upon multiresolution analysis makes this much more than simple object-oriented encapsulation. Indeed, it establishes a

numerical calculus equivalent to that of mathematics and enables facile translation of many of the equations of physics and chemistry into practical code rather than the currently standard tedious and error prone manipulation of large sparse lists of expensive integrals. Robust and efficient code requires an end-to-end error analysis so that all intermediates have no more and no less than the requisite precision. To guarantee precision every function and the results of applying operators to functions can change the mesh refinement, and in electron structure we may have thousands of functions (electronic wave functions) each with its own distinct mesh. This very dynamic environment forced the development of the new runtime that presently sits upon MPI but a tuned version is being developed specifically for the Cray XT using portals.

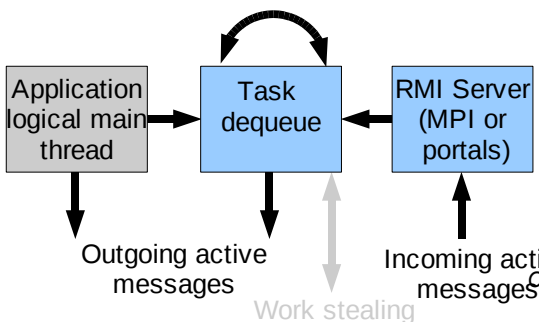
3 Scalable parallel runtime

In addition to the primary funding from DOE SciDAC, support from the NSF for a collaboration with Sadayappan (computer science) enabled us to be much more ambitious with the scope of our parallel runtime and programming environment, and support from the DARPA as part of their evaluation of the three high-productivity languages (X10, Chapel, Fortress) enabled incorporation of some of their concepts and evaluation of them in a real application [Barrett08]. This work has resulted in the MADNESS parallel runtime being a very high-level environment for the composition of a wide range of parallel algorithms.



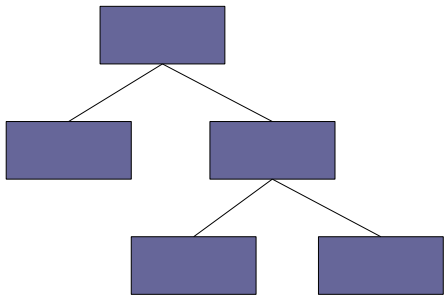
Layered architecture of MADNESS.

The central elements of the parallel runtime are a) use of futures [Fridman76, Baker77] for hiding latency and managing dependencies, b) global namespaces so that applications are composed using names or concepts central to the application rather than having to map onto a partitioned linear memory model and/or do explicit pointer management, c) non-process centric computing through remote method invocation in objects in global namespaces, and d) dynamic load balancing and data redistribution. The placement of data and scheduling of computation should be the responsibility of an intelligent runtime. Dependencies are expressed by using futures to pass results of still incomplete or pending tasks (even remote operations) as arguments of new tasks that are scheduled for execution once all dependencies are satisfied (using callbacks). An SMP node presently runs one MPI process that includes the main application thread, a remote method-invocation (RMI) server thread, and a pool of computational threads to which tasks may be submitted either locally or remotely. By relieving the programmer of these many burdens, he/she is freed to focus upon science and algorithm, and is responsible primarily only for expressing concurrency. Many of these concepts appear in [Cilk] and [Charm++], though MADNESS works portably in a distributed memory environment and unlike Charm++ MADNESS is fully compatible with legacy software, which is a nearly fundamental constraint in scientific computing. Other motivating projects include [ACE] and [STAPL] but the former emphasizes distributed systems rather than HPC and STAPL is not available for use by others. Due to limited computer science resources the MADNESS internal parallel runtime is actually far from intelligent and only provides limited dynamic data redistribution. Both of these are central to programmability and performance for many applications at the petascale and part of the focus of our proposed activities.



MADNESS thread SMP thread structure.

3.1 Data structures used in MADNESS



Binary tree.

The multiresolution adaptive nature of the numerical runtime forces the architecture to meet stringent requirements. For example, functions need to be adaptively refined each time that they are modified. This implies that the data structure representing a function needs to be dynamic and able to seamlessly adapt to the complex and irregular flow of data. Moreover, these manipulations need to be carried out in a scalable fashion employing a high degree of parallelism.

A data structure that meets these requirements is that of a multidimensional binary tree where each node of the binary tree would possibly contain either scaling or wavelet coefficients. In the scaling function basis, the only nodes to contain coefficients would be the leaf nodes. Conversely, in the wavelet basis, only the top node would contain scaling function coefficients.

Another requirement of the implementation of this data structure is that algorithms employed inside of the numerical

runtime need to be able to traverse the binary tree recursively and in parallel. For example, when a function is first projected into the scaling function basis, the algorithm walks down the tree, computing the wavelet coefficients, and checking for error bound requirements, in a recursive fashion. In this and many other instances, the algorithm has no a priori knowledge about which function nodes will contain coefficients and which will not.

Other algorithms require random access to the coefficients or need to access them in a manner that is inefficient if restricted to a linked tree. For these reasons and for generality, the coefficients are actually stored in a globally addressable, distributed hash table. The container employs a user-definable processor map between items in the container and SMP nodes. One benefit of using a processor map for the decomposition of function nodes is that it provides a natural way of load-balancing by reshuffling the map.

3.2 Task-oriented computing

To employ a maximum degree of parallelism, MADNESS relies on an asynchronous model of computing. Large operations such as an inner product between to functions or the convolution of two functions are divided in to smaller operations called *tasks*. Each task is then executed somewhere on the machine in an asynchronous manner. This modality of parallel computing is what we term as *task-oriented computing*.

Task-oriented computing complements the more common data parallelism. One can exploit task parallelism within a given SMP node by using threading. Coarse-grain parallelism is taken care of by domain decomposition, while fine-grain parallelism is expressed by having different threads that are local to each node execute tasks in parallel.

Fine-grain parallelism is implemented by having each SMP node host a task queue that functions like a virtual processor by holding tasks to be executed by a pool of worker threads. This works as follows:

1. In the main execution thread (not a worker thread), the runtime decomposes a high level mathematical instruction into a bevy of lower level tasks.
2. Each one of these tasks is then submitted to the task queue on a particular SMP node.
3. Each task, having a number of dependencies such as input data or parameters, will remain in a wait state until all of these dependencies are satisfied.
4. Once all of the dependencies of a given task are satisfied, the task will then enter a "ready to run" state where it will eventual be executed by a worker thread from the thread pool.

One feature of this approach is that tasks can execute locally or remotely; this provides another avenue of load-balancing. If certain tasks queues were to become idle, they could steal tasks from other nodes. In addition to work stealing algorithms, this approach provides a setting for fault tolerance. If one node were to become disabled, its data and tasks could be redistributed among all of the other nodes in a seamless fashion.

3.3 Futures

To maintain causality, that is to satisfy the dependencies between tasks, MADNESS makes use of a construct known as a *Future*. A future is the result of an asynchronous computation. When a task is submitted, all of the task's dependencies (input arguments) are wrapped in a Future. Tasks count the number of unsatisfied dependencies, and remain in a waiting status until all have been satisfied. As dependencies become satisfied, the Future object notifies its parent task. When all dependencies have been satisfied, the task will submit itself to be executed.

4 Density functional theory and band structure

4.1 Density functional theory

A standard problem in computational physics and chemistry applications is to compute the ground state of the following Hamiltonian:

$$H = \sum_{k}^{N} \left(-\frac{1}{2}\nabla_k^2 + V_k(r_k) \right) + \frac{1}{2}\sum_{k,l}^{N} \frac{1}{|r_k - r_l|}$$

The first and second terms describe the single-particle kinetic and external potential energy pieces, respectively. The last term describes the Coulomb interaction between different electrons. "N" is the number of electrons. The solution to this Hamiltonian is a multidimensional wavefunction,

$$\psi(r_1, r_2, r_3, \ldots, r_N).$$

Three main difficulties arise in solving this problem.

1. The shear size of the problem is intractable. "N" is on the order of 10^{23}. If one were to compute the contribution to the many-body wavefunction of one electron per microsecond, the calculation would take longer than the lifetime of the universe.
2. Electrons are fermions, therefore the many-body wave function must be antisymmetric under exchange of any two electrons.
3. The Coulomb interaction entangles different

electrons. If it were zero or it could be ignored, then the first two terms would split into N single-particle Hamiltonians.

A breakthrough in modern theoretical physics and chemistry was the advent of density functional theory. Density functional theory, or DFT, reformulates that problem in terms of the electronic density rather than the many-body wavefunction. The electronic density is defined as

$$n(r)=N\int d^3r_2\ldots d^3r_N\left|\psi(r,r_2,\ldots,r_N)\right|^2 .$$

In 1964, Hohenberg and Kohn reformulated the many-body Hamiltonian by showing that there is a unique mapping between the electronic density, $n(r)$, and the external potential, $v(r)$ up to a constant. The implies that the total energy of an electronic system is a functional of the electronic density.

$$E=\int v(r)n(r)+F[n(r)]$$

Furthermore, the energy is variational with respect to $n(r)$, and hence the total energy will match the ground state energy when $n(r)$ coincides with the true ground state electronic density.

The implementation of density functional theory is centered around taking the functional derivative of this energy functional with respect to the electronic density. This functional derivative, along with the fact that the density can be constructed from fictitious, non-interacting single-particle states, allows the problem to be reduced to solving the following single-particle Hamiltonian.

$$\left[-\frac{1}{2}\nabla_j^2+V_{eff}[n](r)\right]\phi_j(r)=\epsilon_j\phi_j(r)$$

$\phi(r)$ are fictitious, single-particle electronic states. The density can be constructed as

$$n(r)=\sum_j\left|\phi_j(r)\right|^2 .$$

The effective potential is a functional of the density, $n(r)$.

$$V_{eff}[n](r)=V_{ext}(r)+\int d^3s\frac{n(r)}{|r-s|}+V_{xc}[n](r)$$

The first term is the nuclear potential. The second and third terms are the Hartree and exchange-correlation terms, respectively. Due to the effective potential being a functional of the electronic density, the system of equations must be solved self consistently.

4.2 Periodic systems

In a perfect crystalline material the effective potential is periodic.

$$V_{eff}(r+R)=V_{eff}(r)$$

$$T\equiv lattice\ translation$$

The single-particle wavefunctions have the form of Bloch states.

$$\phi_{n,k}(r)=e^{ik\cdot r}u_{n,k}(r)$$

The function $u_{n,k}(r)$ is also is periodic within a translation of a lattice vector. $e^{ik\cdot r}$ is a global phase factor that has a periodicity of the entire crystal. Expanding the effective potential and the Bloch states in a Fourier basis, the single-particle Hamiltonian is transformed into the following matrix equation.

$$\left[\frac{1}{2}(k+G)^2-\epsilon_G^{(k)}\right]\alpha_G^{(k)}+\sum_{G'}V(G-G')\alpha_G^{(k)\prime}=0$$

This method is know as the *plane-wave expansion*. The benefits of the plane-wave technique is that it is the most natural representation for a periodic system. Moreover, the kinetic energy and Coulomb operators are diagonal in this representation. Unfortunately, the high-frequencies associated with core electrons (those closest to the nuclei) require many plane-waves for accuracy. One way to circumvent this problem is to represent the effect of the core electrons and the nuclear potential into a smooth effective potential. This is known as the *pseudopotential method*. The global extent of the basis functions also makes it hard to derive algorithms that scale less than cubically with the system size.

Another electronic structure technique used is sold-state physics in the *linearized augmented plane-wave method (LAPW)*. The LAPW method is built around partitioning the simulation space into two different regions. In the region around the core of the nuclei, one constructs an atomic sphere, inside of which mathematical objects are representation by atomic-like orbitals. In between these atomic spheres (the interstitial region) one expands in plane-waves. A disadvantage of the LAPW method is that composing new applications requires an intimate knowledge of this complex basis set.

In contrast, the adaptive basis of MADNESS can efficiently treat both the core and interstitial regions on an equal footing and the multiresolution algorithms (combined with local orbital methods) naturally yield fast algorithms. Finally, we have already noted above the very high level at which MADNESS applications are composed.

5 Time evolution of electronic wave functions

Advances in short pulse laser technology have opened the door for experimentalists to probe new regions of fundamental

science. Theory and computation have been searching for new algorithms to outpace Moore's Law as they race to keep up. These few cycle laser pulses delve into the attosecond regime which gives us, for the first time, a tool to probe electron motion around the atom.

We solve the time-dependent Schrödinger equation (TDSE),

$$i\frac{d}{dt}\psi(r,t)=H(t)\psi(r,t)$$

using a time stepping method (see below).Presently this is being performed in three (one electron) and four dimensions (one electron and one internuclear distance). Upon completion of the femto-second laser pulse, analysis of the wave packet is complicated by the finite separation of the ionized electrons that continue to repel each other. Experimentally observable differential cross sections can be obtained by projecting the time evolved wave function onto the scattering eigenstates of the system.

The adaptive MADNESS basis enables use of box so large that the wave packet does not reach the edge of the simulation volume. This avoids the need to introduce a complex absorbing potential or other technique to avoid spurious reflections.

The propagation scheme employed is the fourth-order accurate gradient symplectic integrator due to Chin et al. [Chin01].

$$U(t)=e^{-itV(t)/6}e^{-it\hat{T}/2}e^{-i2t\tilde{V}(t/2)/3}e^{-it\hat{T}/2}e^{-itV(0)/6}$$

$$\tilde{V}=V-\frac{t^2}{48}[V,[T,V]]$$

where $\hat{T}$ and V are the kinetic and potential energy operators, respectively. Matrix elements of the free-particle propagator $e^{-it\hat{T}}$ (combined with a projector onto a band limit corresponding to a maximum grid resolution) are computed so that the operator may be applied as a single sparse matrix-vector product in the wavelet basis.

6 Acknowledgments

This research was supported by the Division of Chemical Sciences, Geosciences, and Biosciences, Office of Basis Energy Sciences, U.S. Department of Energy, under contract DE-AC05-00OR22725 with Oak Ridge National Laboratory, managed and operated by UT-Battelle, LLC, and in part by ORNL Laboratory Directed Research and Development Funds. This research was performed in part using the resources of the Center for Computational Sciences at Oak Ridge National Laboratory under contract DE-AC05-00OR22725. This document describes activities performed under contract number DE-AC0500OR22750 between the U.S. Department of Energy and Oak Ridge Associated Universities.

7 Contact Information

Scott Thornton and Nicholas Vence are graduate students in the physics at the University of Tennessee. Robert Harrison is a professor of chemistry at the University of Tennessee.

8 References

[ACE] http://www.cs.wustl.edu/~schmidt/ACE.html

[Baker77] H. Baker (August 1977). "The Incremental Garbage Collection of Processes". Proceedings of the Symposium on Artificial Intelligence Programming Languages, SIGPLAN Notices 12.

[Barrett08] R. F. Barrett, S. R. Alam, V. F. de Almeida, D. E. Bernholdt, W. R. Elwasif, J. A. Kuehn, S. W. Poole, and A. G. Shet, Exploring HPCS Languages in Scientific Computing, in Rick Stevens, editor, SciDAC 2008, 14-17 July 2008, Seattle

[Beste06] A. Beste, R. J. Harrison and T. Yanai, “Direct computation of general chemical energy differences: Application to ionization potentials, excitation, and bond energies”, J. Chem. Phys. 125, 074101 (2006) [Charm++] http://charm.cs.uiuc.edu

[Cilk] http://supertech.csail.mit.edu/cilk/

[Charm++] http://charm.cs.uiuc.edu

[Chin01] S.A. Chin and C.R. Chen “Fourth order gradient symplectic integrator methods for solving the time-dependent Schrödinger equation,” J. Chem. Phys. 114 (2001) 7338.[Friedman76] D. Friedman (1976). "CONS should not evaluate its arguments". S. Michaelson and R. Milner, editors, Automata, Languages and Programming, pages 257-284. Edinburgh University Press, Edinburgh.

[Harrison04] R. J. Harrison, G. I. Fann, T. Yanai, Z. Gan and G. Beylkin, “Multiresolution Quantum Chemistry: Basic Theory and Initial Applications,” J. Chem. Phys., 121 (2004) 11587

[Sekino08] H. Sekino, Y. Maeda, T. Yanai, R. J Harrison,”Basis set limit Hartree-Fock and density functional theory response property evaluation by multiresolution multiwavelet basis,” J Chem Phys. 2008 Jul 21;129 (3):034111[Yanai04] T. Yanai, G. Fann, Z. Gan, R. Harrison and G. Beylkin, Multiresolution quantum chemistry: Hartree-Fock exchange J. Chem. Phys. 121 (14) (2004) 6680-6688.

[STAPL]http://parasol.tamu.edu/groups/rwergergroup/research/stapl