Parallel Implementation Details of OpenAtom

An accurate understanding of phenomena occurring at the quantum scale can be achieved by considering a model representing the electronic structure of the atoms involved. The Car-Parrinello *ab initio* Molecular Dynamics (CPAIMD) method [**?**, **?**, **?**, **?**] is one such algorithm which has been widely used to study systems containing $10 - 10^3$ atoms. The implementation of CPAIMD in Charm++ is called OpenAtom [**?**, **?**]. To achieve a fine-grained parallelization of CPAIMD, computation in OpenAtom is divided into a large number of virtual processors which enables scaling to tens of thousands of processors. We will look at the parallel implementation of this technique, understand its computational phases and the communication involved and then analyze the benefit from topology-aware mapping of its objects.

1 Parallel Implementation

In an ab-initio approach, the system is driven by electrostatic interactions between the nuclei and electrons. Calculating the electrostatic energy involves computing several terms: (1) quantum mechanical kinetic energy of non-interacting electrons, (2) Coulomb interaction between electrons or the Hartree energy, (3) correction of the Hartree energy to account for the quantum nature of the electrons or the exchange-correlation energy, and (4) interaction of electrons with atoms in the system or the external energy. Hence, CPAIMD computations involve a large number of phases (Figure 1) with high interprocessor communication. These phases are discretized into a large number of virtual processors which generate a lot of communication, but ensures efficient interleaving of work. The various phases are:

- **Phase I:** In this phase, the real-space representation of the electronic states is obtained from the g-space representation through a transpose based 3-dimensional Fast Fourier Transform (FFT).
- **Phase II:** Electron density in real-space is obtained via reductions from the real-space state representation.
- **Phase III:** Fourier components of the density in g-space are created from the corresponding copy in real-space through a 3D FFT.
- **Phase IV:** Once we have the g-space copy, it is used to compute the "Hartree and external energies" via multiple 3D FFTs which can be performed independently.
- **Phase V:** The energies computed in the previous phase are reduced across all processors and send to the corresponding planes of the different states through multicasts. This is exactly reverse of the procedure used to obtain the density in phase II.
- **Phase VI:** In this phase, the forces are obtained in g-space from real-space via a 3D FFT.

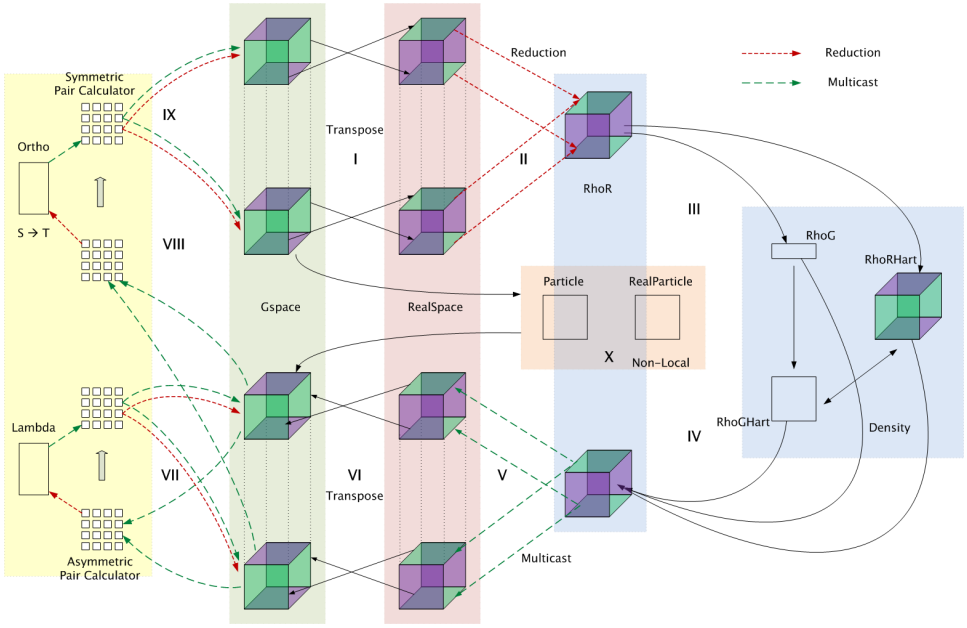

Figure 1: Flow of control in OpenAtom

- **Phase VII:** For functional minimization, force regularization is done in this phase by computing the overlap matrix Lambda (Λ) and applying it. This involves several multicasts and reductions.
- **Phase VIII:** This phase is similar to Phase VII and involves computation of the overlap matrix Psi (Ψ) and its inverse square root (referred to as the S $\rightarrow$ T process) to obtain "reorthogonalized" states. This phase is called orthonormalization.
- **Phase IX:** The inverse square matrix from the previous phase is used in a "backward path" to compute the necessary modification to the input data. This again involves multicasts and reductions to obtain the input for phase I of the next iteration.
- **Phase X:** Since Phase V is a bottleneck, this phase is interleaved with it to perform the non-local energy computation. It involves computation of the kinetic energy of the electrons and computation of the non-local interaction between electrons and the atoms using the EES method [**?**].

For a detailed description of this algorithm please refer to [**?**]. We will now proceed to understand the communication involved in these phases through a description of the various chare arrays involved and dependencies among them.

2 Communication in OpenAtom

The ten phases described in the previous section are parallelized by decomposing the physical system into 15 chare arrays of different dimensions (ranging between one and four). A description of these objects and communication between them follows:

1. **GSpace and RealSpace:** These represent the g-space and real-space representations of the electronic states. They are 2-dimensional arrays with states in one dimension and planes in the other. They are represented by $G(s,p)$ $[n_s \times N_g]$ and $R(s,p)$ $[n_s \times N]$ respectively. GSpace and RealSpace interact through transpose operations in Phase I and hence all planes of one state of GSpace interact with all planes of the same state of RealSpace. RealSpace also interacts with RhoR through reductions in Phase II.

2. **RhoG and RhoR:** They are the g-space and real-space representations of electron density and are 1-dimensional (1D) and 2-dimensional (2D) arrays respectively. They are represented as $G_\rho(p)$ $[N_{g\rho}]$ and $R_\rho(p,p')$ $[(N/N_y) \times N_y]$. RhoG is obtained from RhoR in Phase III through two transposes.

3. **RhoGHart and RhoRHart:** RhoR and RhoG are used to compute their Hartree and exchange energy counterparts through several 3D FFTs (in Phase IV). This involves transposes and point-to-point communication. RhoGHart and RhoRHart are 2D and 3D arrays represented by $G_{HE}(p,a)$ $[N_{gHE} \times n_{atom-type}]$ and $R_{HE}(p,p',a)$ $[(1.4N/N_y) \times N_y \times n_{atom-type}]$.

4. **Particle and RealParticle:** These two 2D arrays are the g-space and real-space representations of the non-local work and denoted as $G_{nl}(s,p)$ $[n_s \times N_g]$ and $R_{nl}(s,p)$ $[n_s \times 0.7N]$. Phase X for the non-local computation can be overlapped with Phases II-VI and involves communication for two 3D FFTs.

5. **Ortho and CLA_Matrix:** The 2D ortho array, $O(s,s')$ does the post-processing of the overlap matrices to obtain the T-matrix from the S-matrix. There are three 2D CLA_Matrix instances used in each of the steps of the inverse square method (for matrix multiplications) used during orthonormalization. In the process, these arrays communicate with the paircalculator chare arrays mentioned next.

6. **PairCalculators:** These 4-dimensional (4D) arrays are used in the force regularization and orthonormalization phases (VII and VIII). They communicate with the GSpace, CLA_Matrix and Ortho arrays through multicasts and reductions. They are represented as $P_c(s,s',p,p')$ of dimensions $n_s \times n_s \times N_g \times N'_g$. A particular state of the GSpace array interacts with all elements of the paircalculator array which have this state in one of its first two dimensions.

7. **Structure Factor:** This is a 3D array used when we do not use the EES method for the non-local computation.

References

[1] R. Car and M. Parrinello. Unified approach for molecular dynamics and density-functional theory. *Phys. Rev. Lett.*, 55:2471–2474, (1985).

[2] G. Galli and M. Parrinello. *Computer Simulation in Materials Science*, 3:283, (1991).

[3] M.C. Payne, M. Teter, D.C. Allan, T.A. Aria, and J.D. Joannopolous. Iterative minimization techniques for ab initio total energy calculations : molecular dynamics and conjugate gradients. *Rev. Mod. Phys.*, 64:1045, (1992).

[4] M. E. Tuckerman. Ab initio molecular dynamics: Basic concepts, current trends and novel applications. *J. Phys. Condensed Matter*, 14:R1297, 2002.

[5] Ramkumar V. Vadali, Yan Shi, Sameer Kumar, L. V. Kale, Mark E. Tuckerman, and Glenn J. Martyna. Scalable fine-grained parallelization of plane-wave-based ab initio molecular dynamics for large supercomputers. *Journal of Comptational Chemistry*, 25(16):2006–2022, Oct. 2004.

[6] Eric Bohm, Glenn J. Martyna, Abhinav Bhatele, Sameer Kumar, Laxmikant V. Kale, John A. Gunnels, and Mark E. Tuckerman. Fine Grained Parallelization of the Car-Parrinello ab initio MD Method on Blue Gene/L. *IBM Journal of Research and Development: Applications of Massively Parallel Systems (to appear)*, 52(1/2), 2007.

[7] H.S. Lee, M.E. Tuckerman, and G.J. Martyna. Efficient evaluation of nonlocal pseudopotentials via euler exponential spline interpolation. *Chem. Phys. Chem.*, 6:1827, (2005).