# Implementation of First-Principles Molecular Dynamics on Large Scale Computers

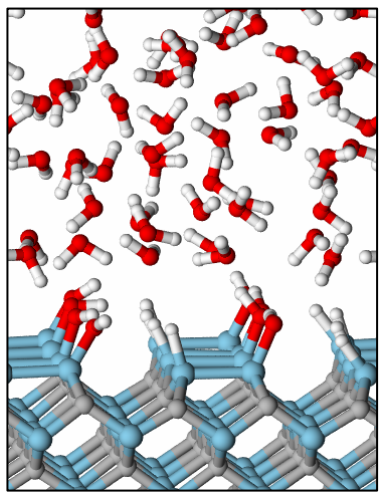
François Gygi
University of California, Davis
fgygi@ucdavis.edu
http://eslab.ucdavis.edu

FADFT Workshop, July 23, 2007

UCDAVIS
UNIVERSITY OF CALIFORNIA

# First-Principles Simulations

- Our goal: Simulate high-Z metals (Mo, Ta, Pu, …) from first principles, without input from experiments
- The approach: Molecular dynamics: an atomic-scale simulation method
  - Compute the trajectories of all atoms
  - extract statistical information from the trajectories

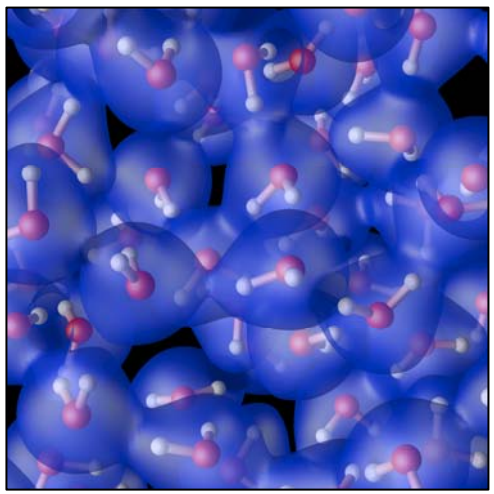


Atoms move according to Newton's law:

$$m_i \ddot{\mathbf{R}}_i = \mathbf{F}_i$$

# First-Principles Simulations

- Why "First-Principles"?
  - Avoid empirical models and adjustable parameters
    - Goal: applications to extreme conditions (high pressure, etc.) where no experimental data is available
  - Use fundamental principles: Quantum Mechanics
  - Must describe ions and electrons consistently and simultaneously



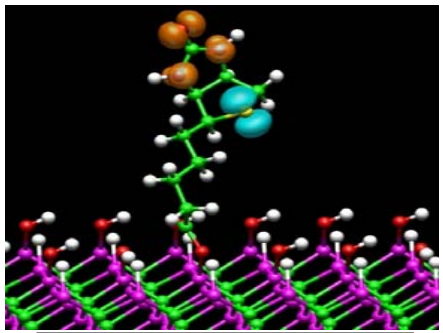
At each time step:

1) Compute the electronic structure
2) Derive interatomic forces
3) Move atoms

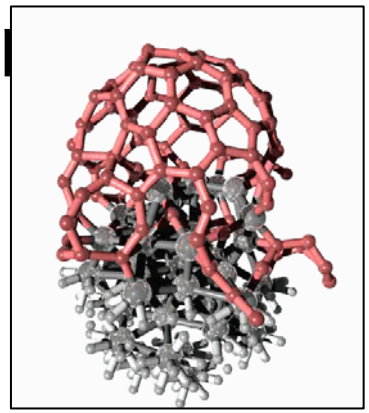
UC DAVIS
UNIVERSITY OF CALIFORNIA

# First-Principles Simulations

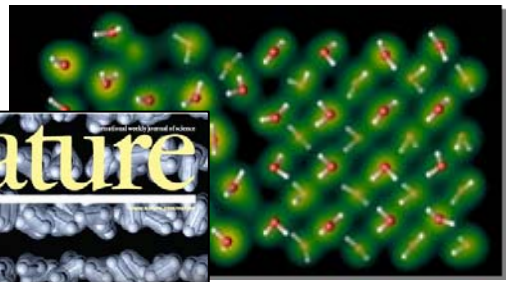
- The approach is applicable to very diverse
  - Chemistry
  - Nanotechnology
  - Semiconductors
  - Biochemistry
  - High-pressure physics



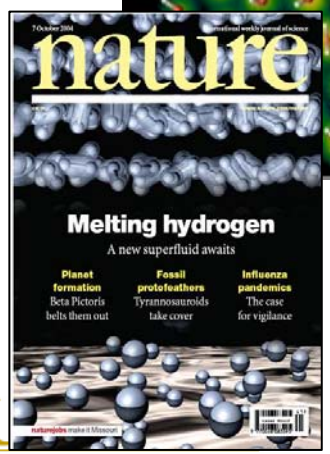
Biotin on silicon carbide

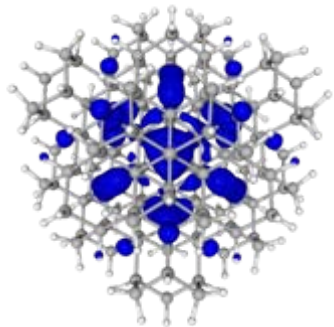


Growth of a carbon nanotube on an iron catalyst



Ice-water interface

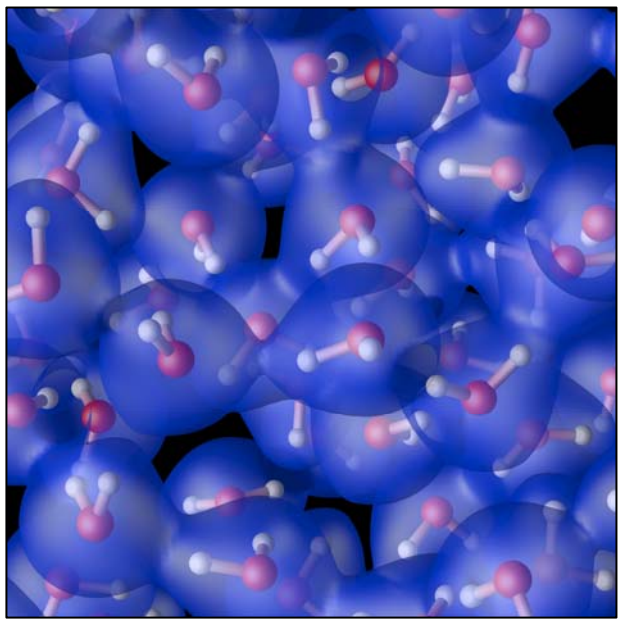


Silicon quantum dot

# First-Principles Simulations

- The computation of the electronic structure is the most expensive part of the simulation



>99% of CPU time

At each time step:

1) Compute the electronic structure
2) Derive interatomic forces
3) Move atoms

# Computing the electronic structure

- Density Functional Theory: the Kohn-Sham equations
  - solutions represent molecular orbitals (one per electron)
  - molecular orbitals are complex scalar functions in $R^3$
  - coupled, non-linear PDEs

$$\begin{cases} -\Delta\varphi_i + V(\rho,\mathbf{r})\varphi_i = \varepsilon_i\varphi_i \quad i = 1\ldots N_{\text{el}} \\ V(\rho,\mathbf{r}) = V_{\text{ion}}(\mathbf{r}) + \int \dfrac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,d\mathbf{r}' + V_{\text{XC}}(\rho(\mathbf{r}),\nabla\rho(\mathbf{r})) \\ \rho(\mathbf{r}) = \displaystyle\sum_{i=1}^{N_{\text{el}}} |\varphi_i(\mathbf{r})|^2 \\ \int \varphi_i^*(\mathbf{r})\,\varphi_j(\mathbf{r})\,d\mathbf{r} = \delta_{ij} \end{cases}$$

# Computing the electronic structure

- Periodic boundary conditions: all solutions of the form

$$\psi_{\mathbf{k},n}(\mathbf{r}) = e^{i\mathbf{k}\cdot\mathbf{r}}\varphi_{\mathbf{k},n}(\mathbf{r})$$

  must be included (Bloch theorem)
- Solutions are represented as Fourier series

$$\varphi_{\mathbf{k},n}(\mathbf{r}) = \sum_{|\mathbf{k}+\mathbf{q}|^2 < E_{\text{cut}}} c_{\mathbf{k}+\mathbf{q},n}\, e^{i\mathbf{q}\cdot\mathbf{r}}$$

- Electronic charge density

$$\rho(\mathbf{r}) = \sum_{n} \int_{BZ} \left|\psi_{\mathbf{k},n}(\mathbf{r})\right|^2 d^3\mathbf{k}$$

# Computing the electronic structure

- A periodic solution is represented by the matrix of complex Fourier coefficients $c_{qn}$

$$\varphi_n(\mathbf{r}) = \sum_{|\mathbf{q}|^2 < E_{\text{cut}}} c_{\mathbf{q},n} e^{i\mathbf{q}\cdot\mathbf{r}}$$

- The matrix of coefficients $c_{qn}$ must have orthogonal columns

- Dimensions of C: $10^6 \times 10^4$

# Controlling Numerical Errors

- The goal is high accuracy
- All numerical errors must be controlled
  - Convergence of Fourier series $\varphi_n(\mathbf{r}) = \sum_{|\mathbf{q}|^2 < E_{\text{cut}}} c_{\mathbf{q},n} e^{i\mathbf{q}\cdot\mathbf{r}}$
  - Convergence of system size (number of atoms)
  - Convergence of k-space integration $\rho(\mathbf{r}) = \int_{BZ} \left|\psi_{\mathbf{k},n}(\mathbf{r})\right|^2 d^3\mathbf{k}$
- We need to systematically increase
  - Plane-wave energy cutoff
  - Number of atoms

**BlueGene/L allows us to ensure convergence of all three approximations**

# Algorithms used in FPMD

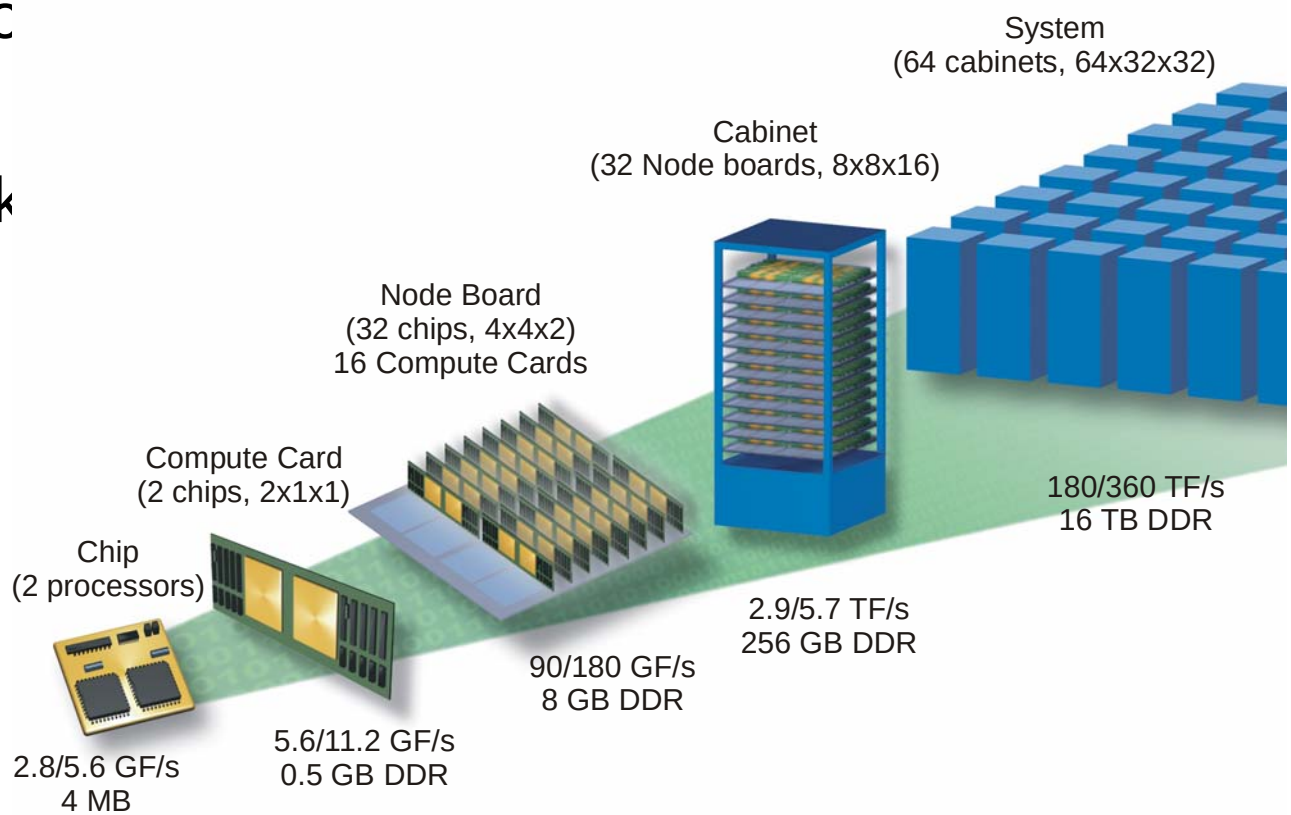
$$\begin{cases} -\Delta\varphi_i + V(\rho,\mathbf{r})\varphi_i = \varepsilon_i\varphi_i \quad i = 1\ldots N_{\text{el}} \\ V(\rho,\mathbf{r}) = V_{\text{ion}}(\mathbf{r}) + \int \dfrac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,d\mathbf{r}' + V_{\text{XC}}(\rho(\mathbf{r}),\nabla\rho(\mathbf{r})) \\ \rho(\mathbf{r}) = \sum_{i=1}^{N_{\text{el}}} |\varphi_i(\mathbf{r})|^2 \\ \int \varphi_i^*(\mathbf{r})\,\varphi_j(\mathbf{r})\,d\mathbf{r} = \delta_{ij} \end{cases}$$

- Solving the KS equations: a constrained optimization problem in the space of coefficients $c_{qn}$
- Poisson equation: 3-D FFTs
- Computation of the electronic charge: 3-D FFTs
- Orthogonality constraints require dense, complex linear algebra (e.g. $A = C^H C$)

Overall cost is $O(N^3)$ for $N$ electrons

# The Platform: BlueGene/L

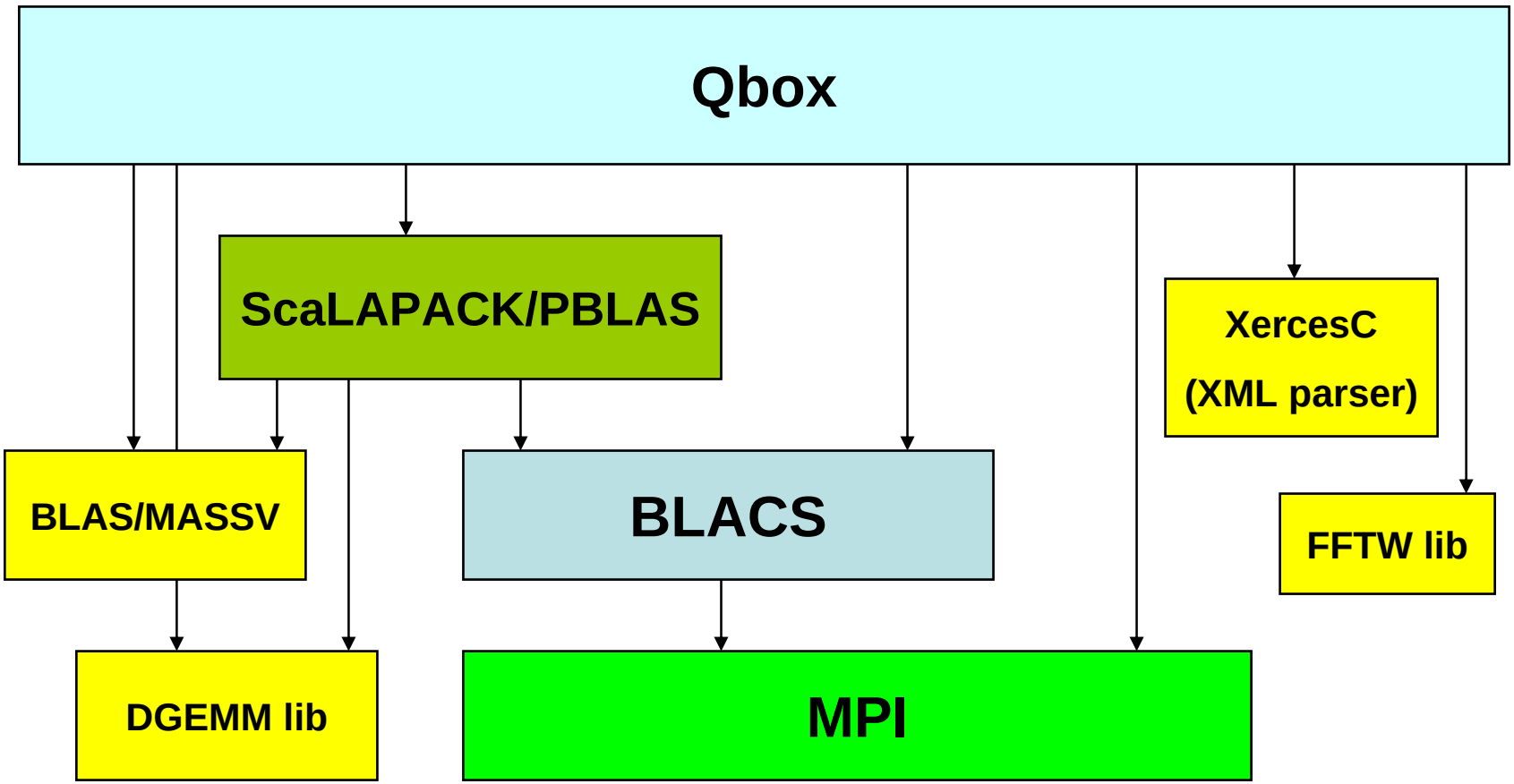
- 65,536 nodes, 128k CPUs
- 3D torus netwc
- 512 MB/node
- 367 TFlop peak



# Qbox code: main features

- C++/MPI implementation of First-Principles Molecular Dynamics
- DFT/GGA exchange-correlation
- Plane-wave, norm-conserving pseudopotentials
- Designed for large-scale parallel platforms
- Main design constraint: small memory footprint (< 512MB per task)
- Built on optimized parallel libs: PBLAS, ScaLAPACK
- XML interface
- Used on various parallel platforms (BG/Ls, Linux clusters)

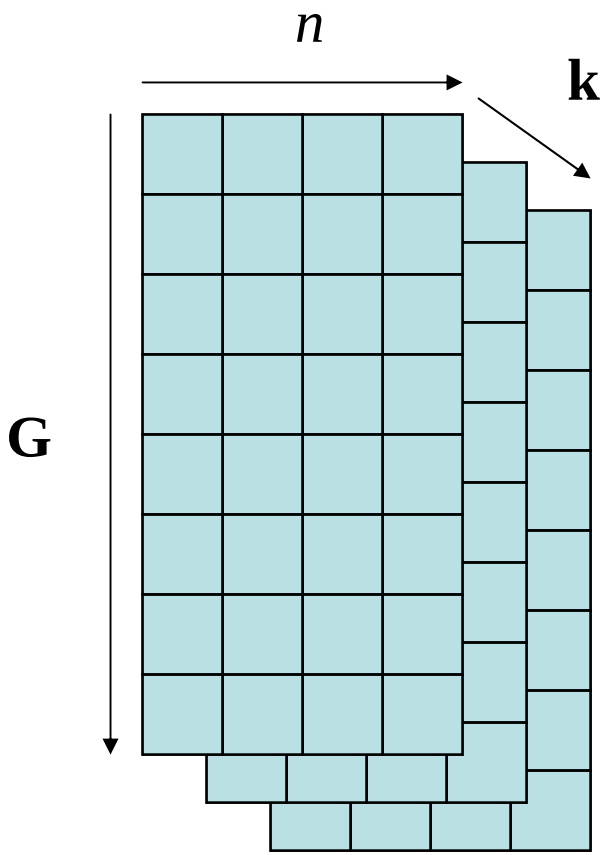
# Qbox code structure



# Data layout

- Distributed plane-wave coefficients

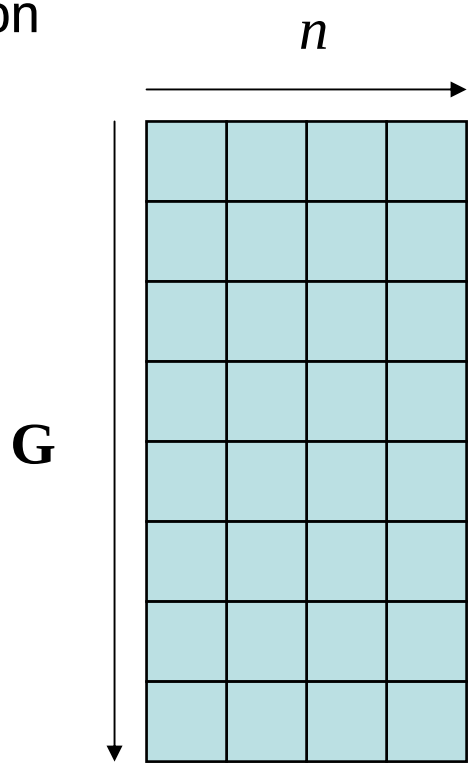
$$\varphi_{n\mathbf{k}}(\mathbf{r}) = \sum_{|\mathbf{k}+\mathbf{G}|^2 < E_{\text{cut}}} c_{n,\mathbf{k}+\mathbf{G}} e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}}$$



# Data layout

- Single k-point wavefunction:
  - ScaLAPACK matrix block distribution

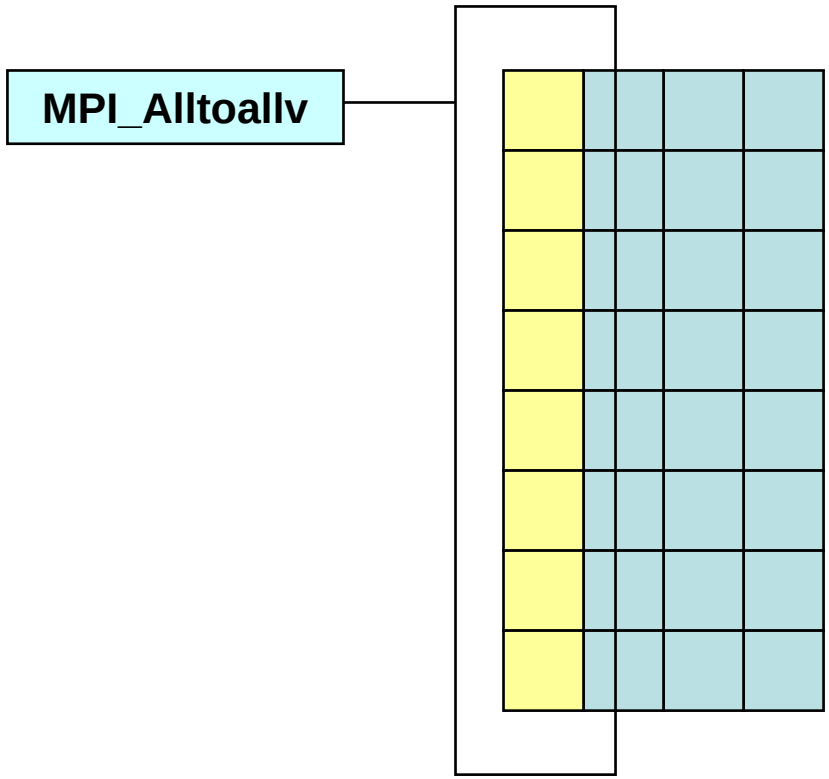
$$\varphi_{n\mathbf{k}}(\mathbf{r}) = \sum_{|\mathbf{k}+\mathbf{G}|^2 < E_{\text{cut}}} c_{n,\mathbf{k}+\mathbf{G}}\, e^{i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}}$$




- Dimensions of C: $10^6 \times 10^4$
- Typical process grid: 512x16

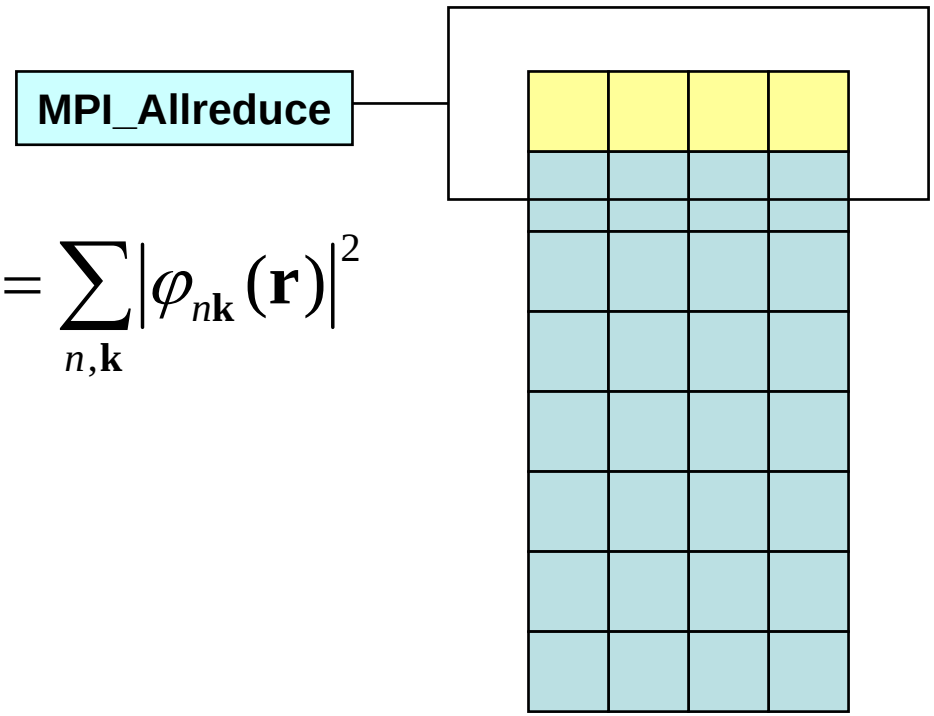
# Communication patterns

- 3-D Fourier transforms $\varphi_{n\mathbf{k}}(\mathbf{r}) \leftrightarrow c_{n,\mathbf{k}+\mathbf{G}}$



# Communication patterns

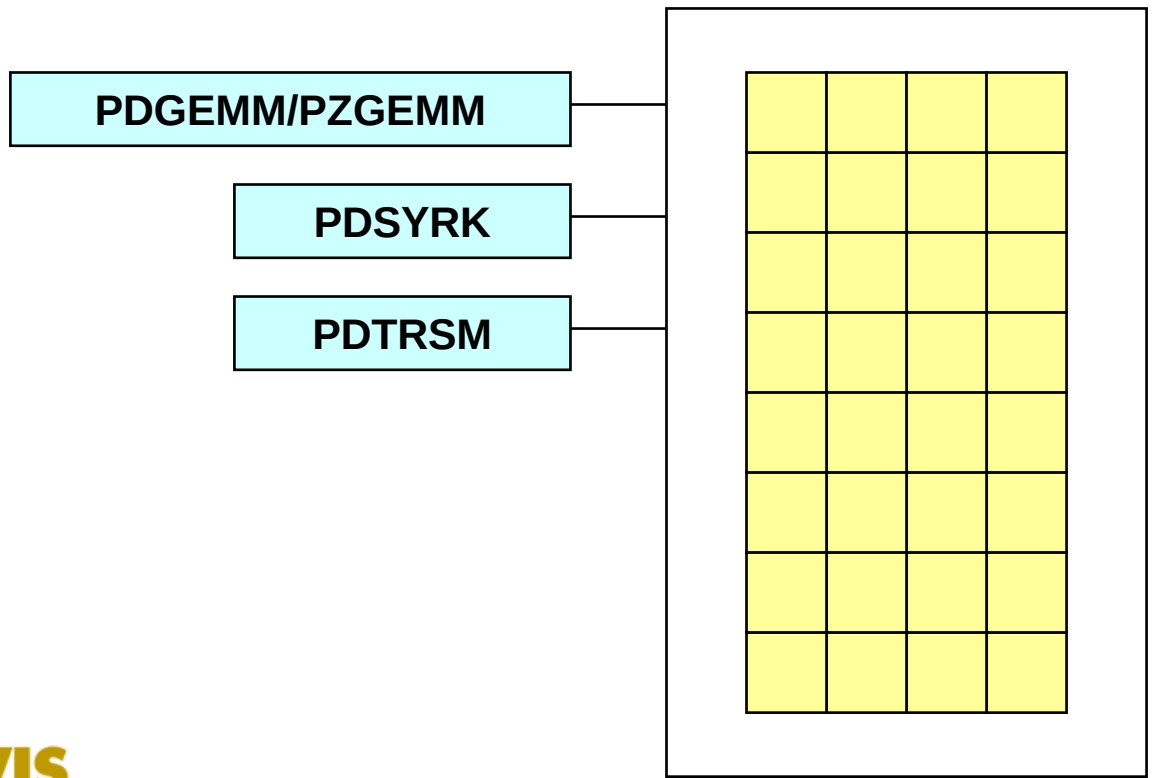
- Accumulation of electronic charge density




$$\rho(\mathbf{r}) = \sum_{n,\mathbf{k}} \left|\varphi_{n\mathbf{k}}(\mathbf{r})\right|^2$$

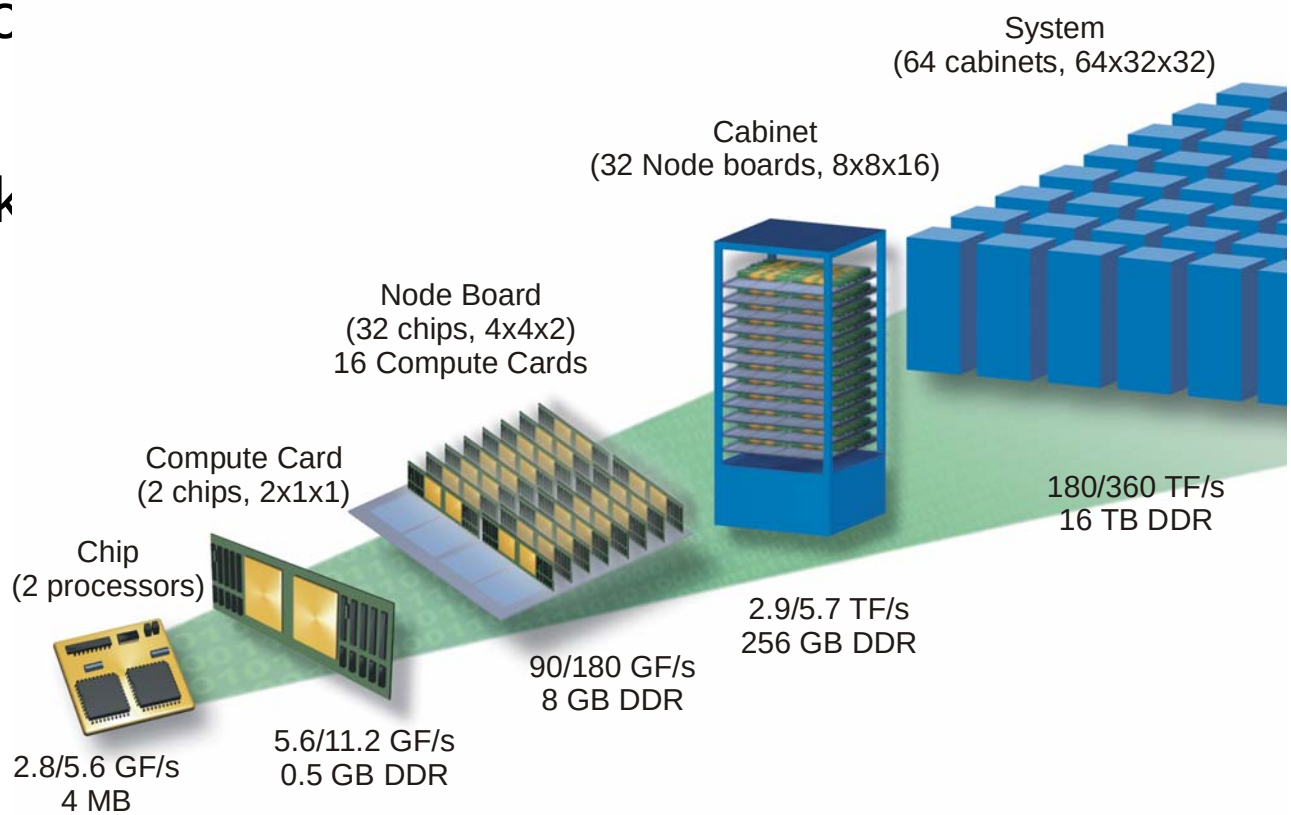
# Communication patterns

- Other operations: (orthogonalization, non-local potential energy, Ritz diagonalization)
  - use the ScaLAPACK linear algebra library



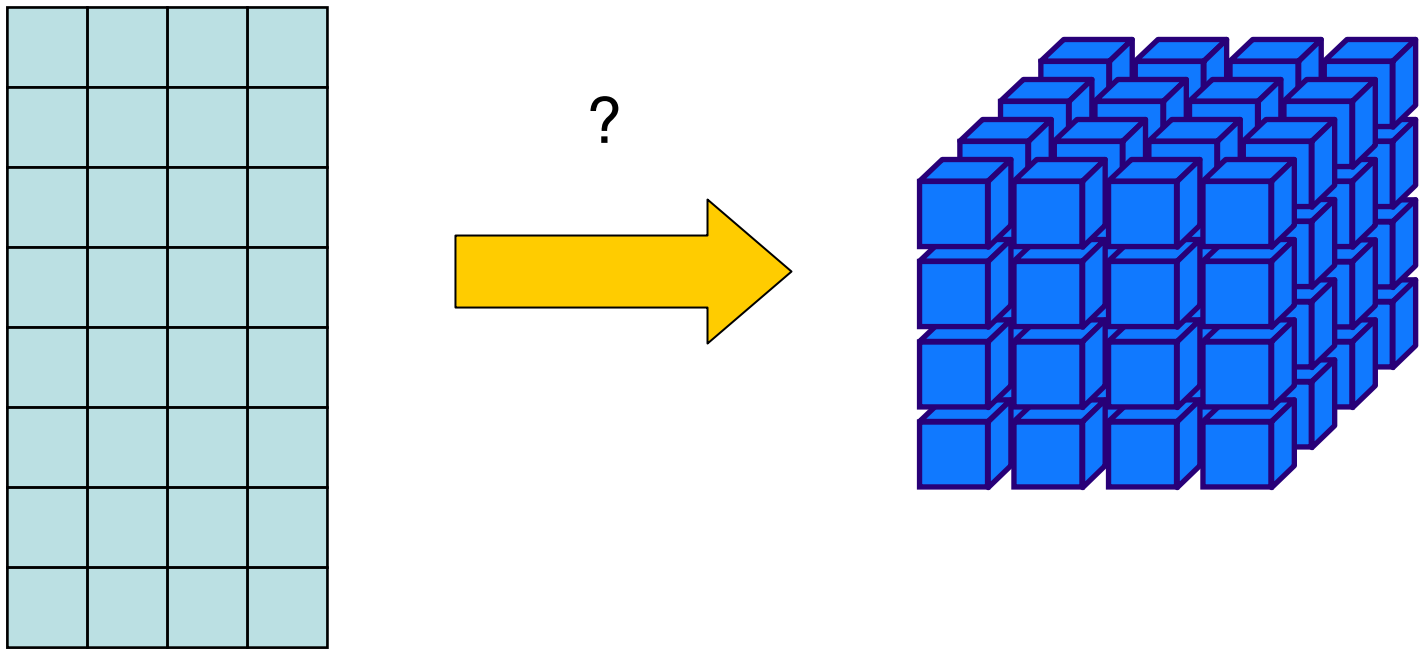
# The Platform: BlueGene/L

- 65,536 nodes, 128k CPUs
- 3D torus netwc
- 512 MB/node
- 367 TFlop peak



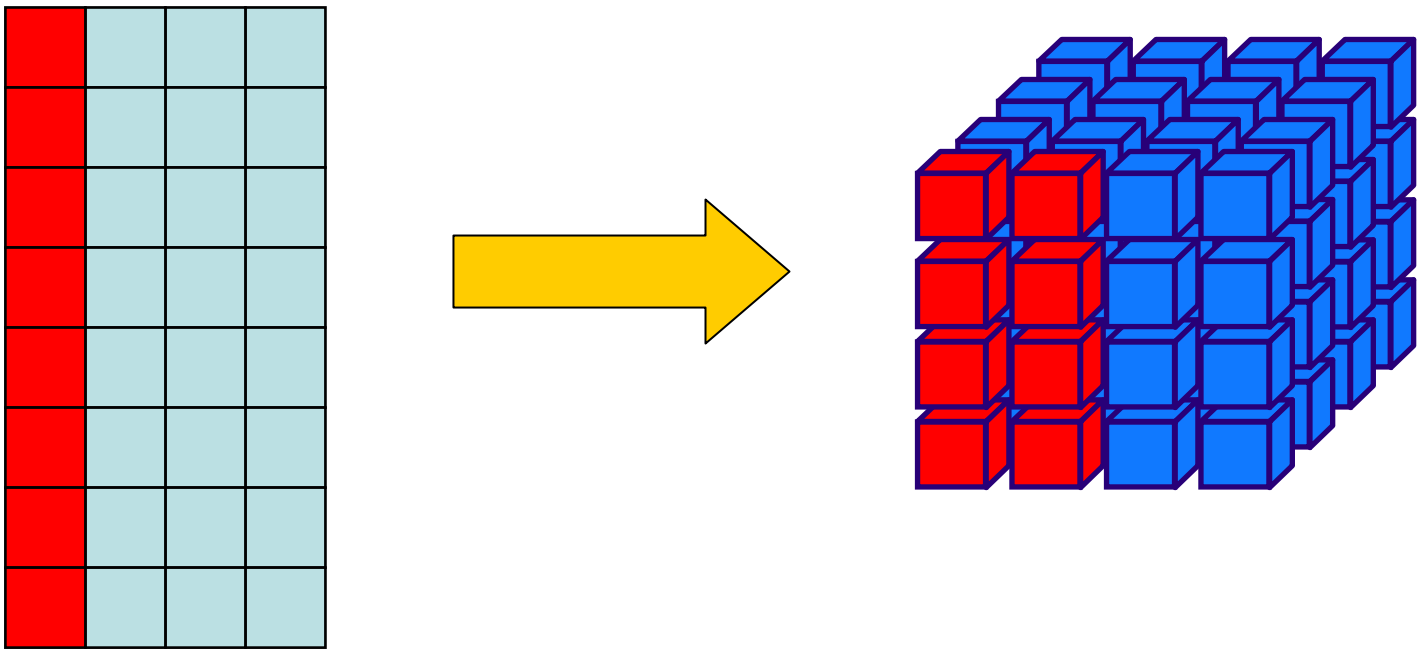
# Mapping tasks to physical nodes

- Mapping a 2-D process grid to a 3-D torus



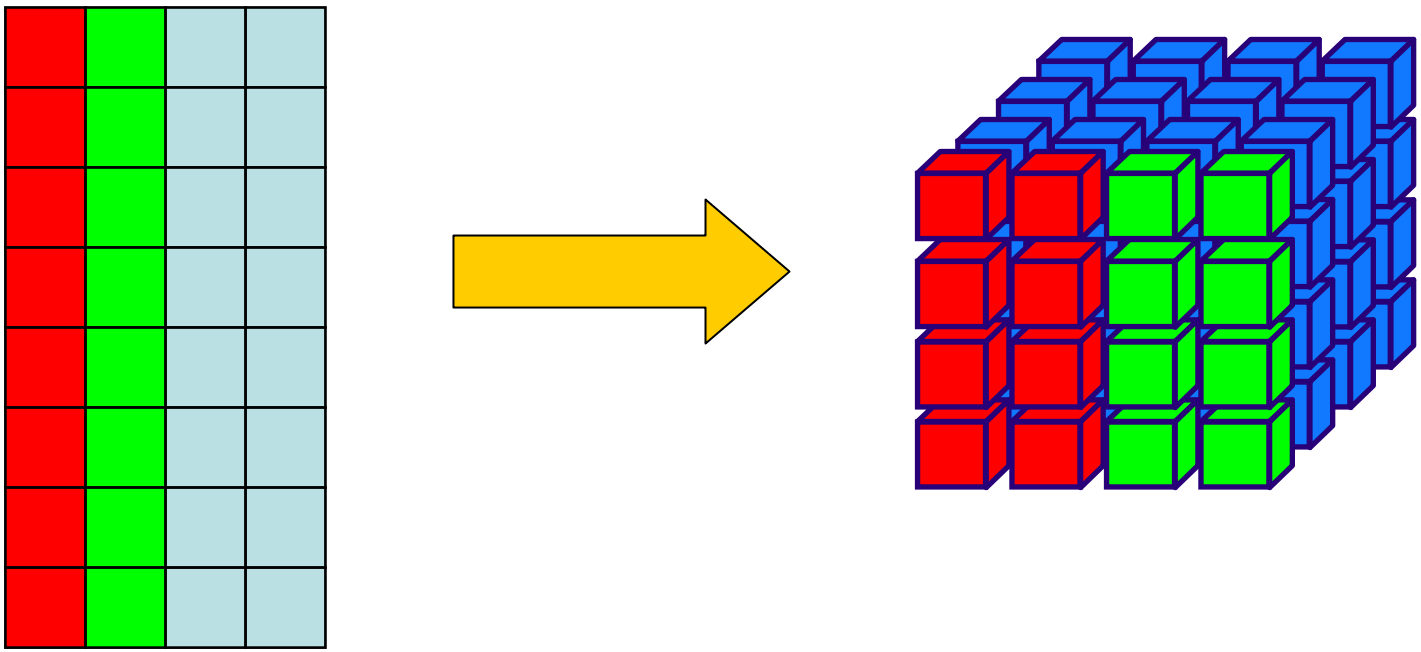
# Mapping tasks to physical nodes

- Mapping a 2-D process grid to a 3-D torus



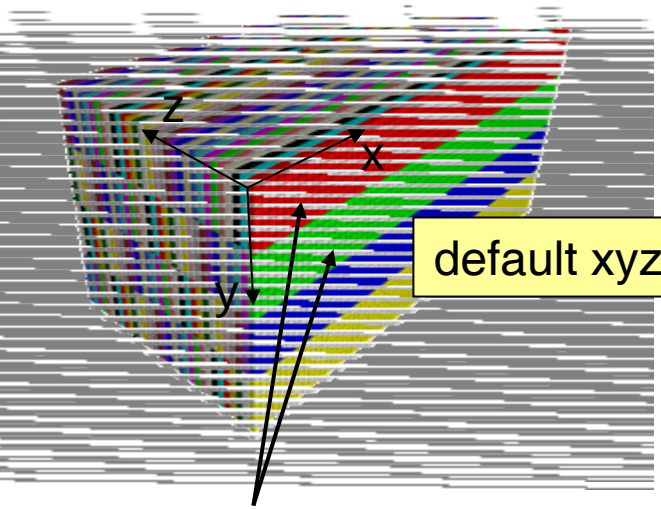
# Mapping tasks to physical nodes

- Mapping a 2-D process grid to a 3-D torus



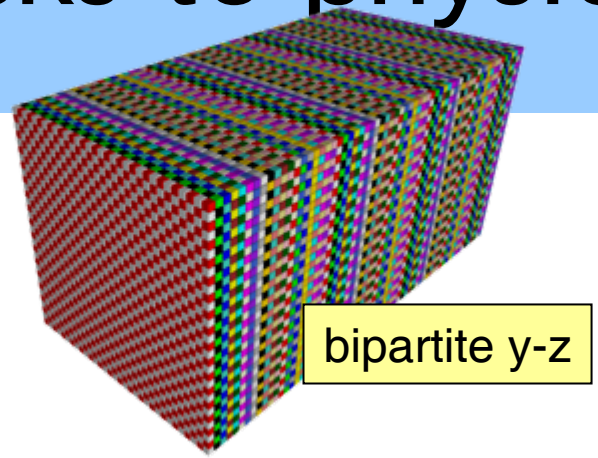
# Mapping tasks to physical nodes

65536 nodes, in a
64x32x32 torus

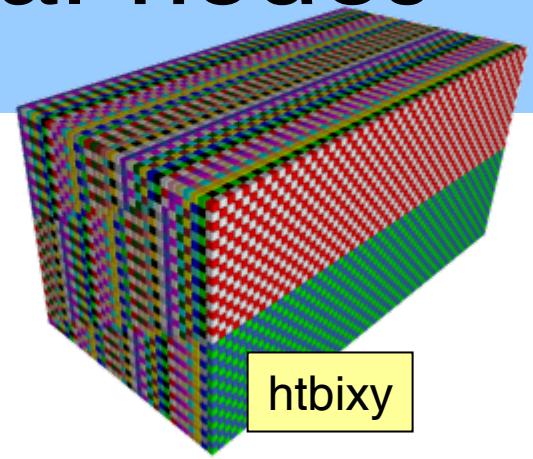



default xyz

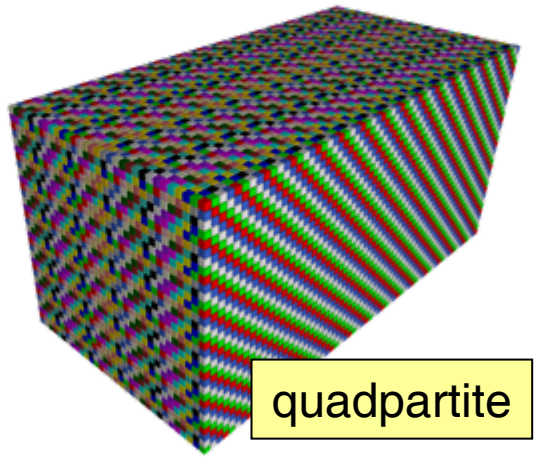
512 tasks per MPI
subcommunicator



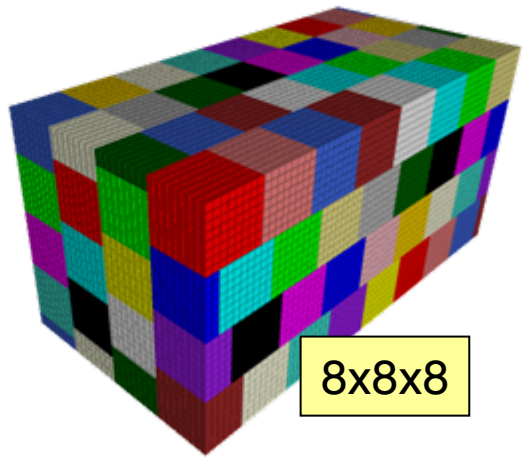
bipartite y-z



htbixy

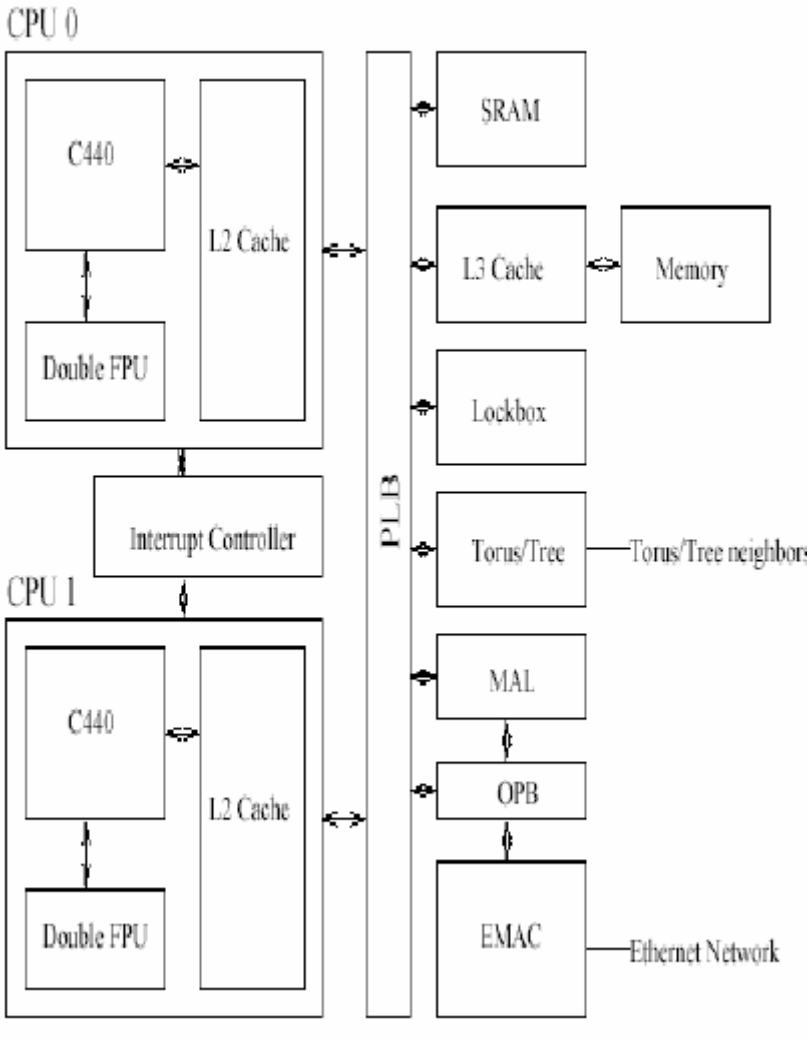


quadpartite



8x8x8

# Single-node kernels




- Dual core
- Dual FPU
- Three-level cache memory hierarchy
- L1 caches not coherent
- L2, L3 coherent

# Single-node kernels

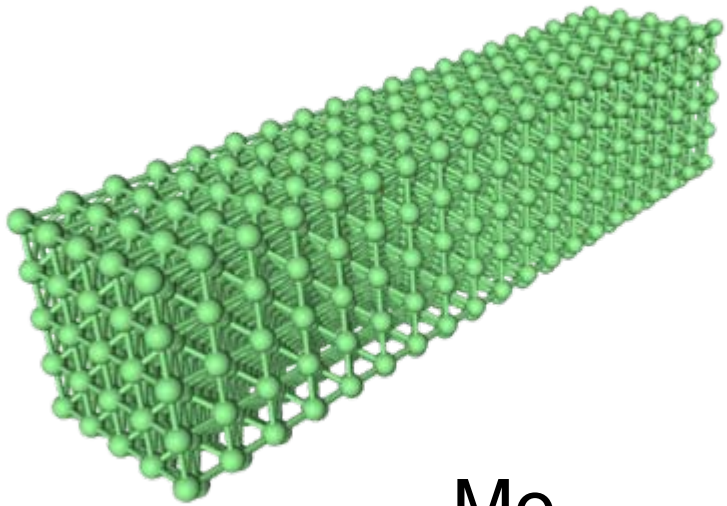
- Exploiting the BG/L hardware
  - Use double FPU instructions ("double hummer")
  - Use both CPUs on the node
    - use virtual node mode, *or*
    - program for two cores
  - We use BG/L in co-processor mode
    - 1 MPI task per node
    - Use second core using dual-core kernels
- DGEMM/ZGEMM kernel (John Gunnels, IBM)
  - Hand optimized, uses double FPU very efficiently
  - Algorithm tailored to make best use of L1/L2/L3
  - Dual-core version available: uses all 4 FPUs on the node
- FFTW kernel (Technical University of Vienna)
  - Uses hand-coded intrinsics for DFPU instructions

# ZGEMM Performance

- ZGEMM on one processor per node
  - In excess of 98% of peak (2.76/2.80 GF/node)
- ZGEMM in co-processor mode
  - 97% of peak (5.43/5.60 GF/node)
  - Uses fork/join construct
  - Flushes caches to maintain coherence
- Kernel performance
  - Over 99.5% of peak

# The Test Problem

- Electronic structure of a 1000-atom Molybdenum sample
- p semi-core electrons included
- 12,000 electrons
- 32 non-local projectors for pseudopotentials
- 112 Ry plane-wave energy cutoff
- High-accuracy parameters

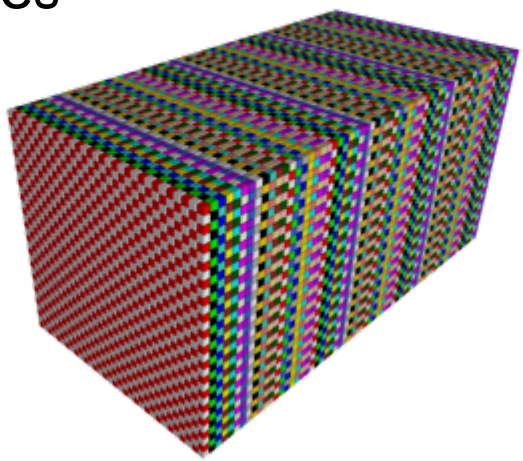


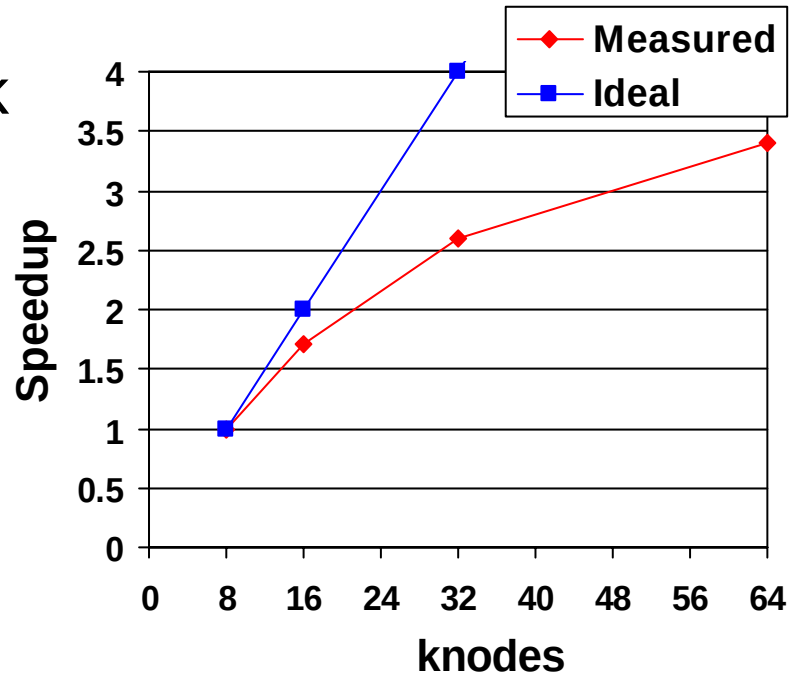
$Mo_{1000}$

# Qbox performance results

- Single k-point calculation (k=0)
- single-core dgemm library
- co-processor mode
- bipartite y-z mapping at 64k nodes

46.70 TFlops on 64k nodes

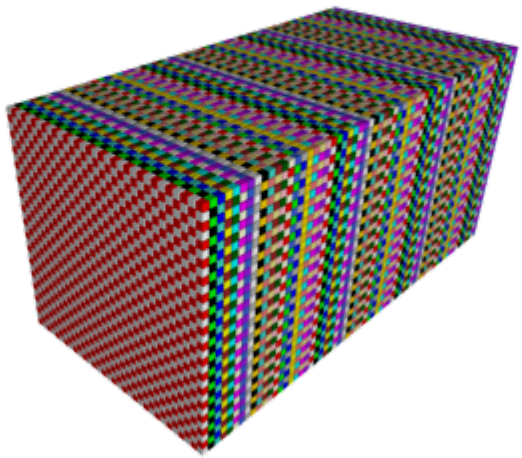


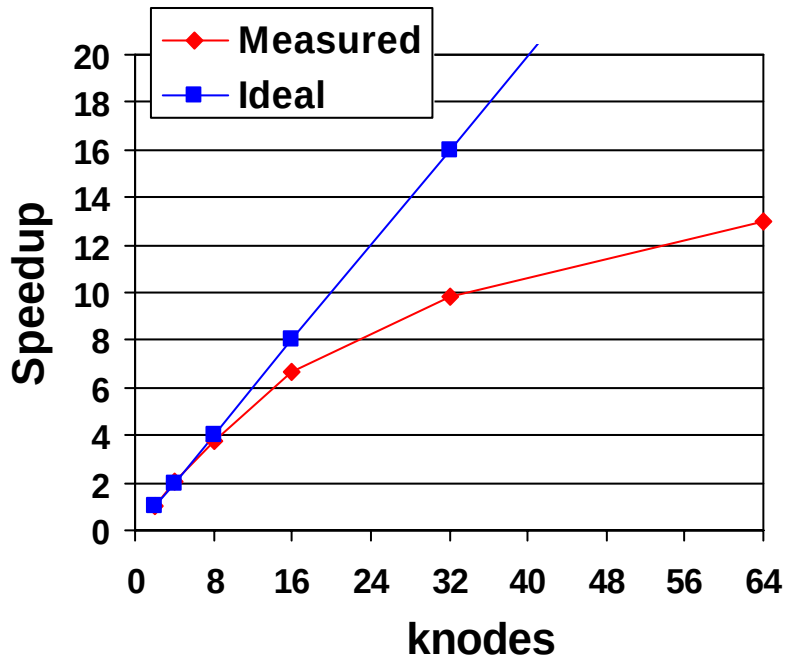


# Qbox performance results

- Single k-point calculation (k=0)
- dual-core dgemm library
- co-processor mode
- bipartite y-z mapping

64.0 TFlops on 64k nodes






UCDAVIS
UNIVERSITY OF CALIFORNIA

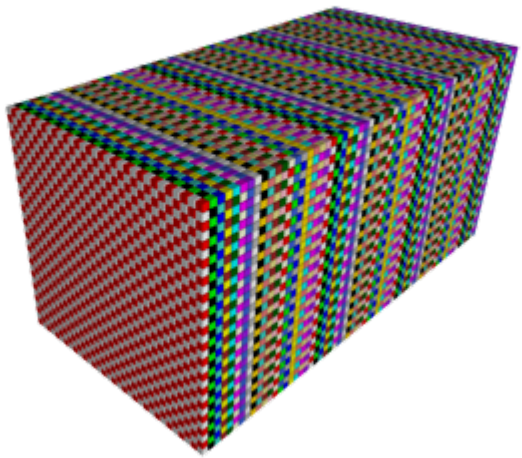
# Qbox performance results

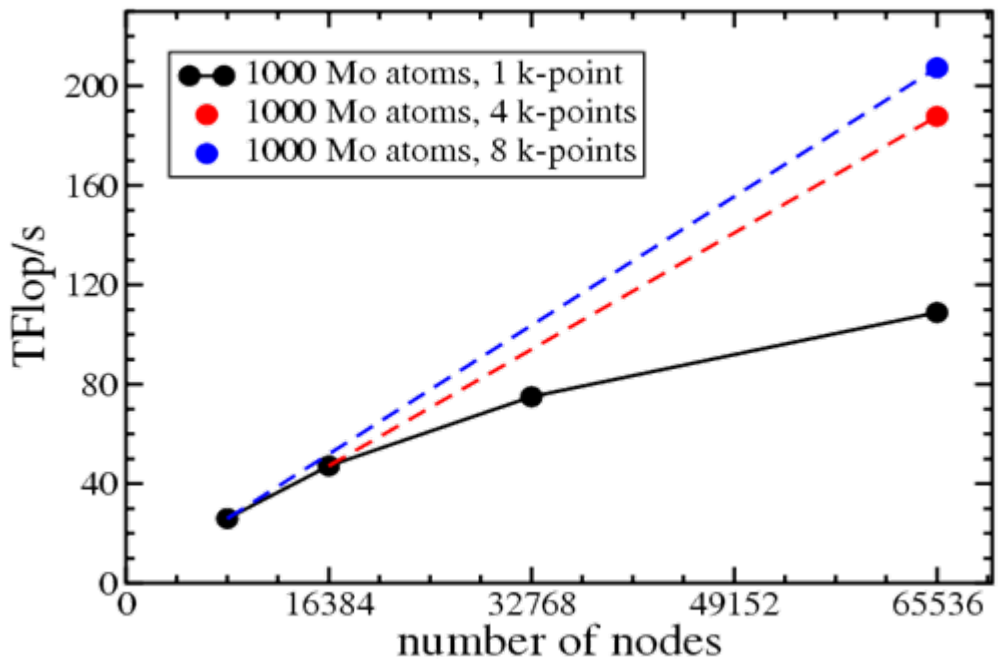
- Multiple k-point calculations
- complex arithmetic
- dual-core zgemm library
- co-processor mode

**1 k-point: 108.8 TFlop/s (30% of peak)**

**4 k-points: 187.7 TFlop/s (51% of peak)**

**8 k-points: 207.3 TFlop/s (56% of peak)**

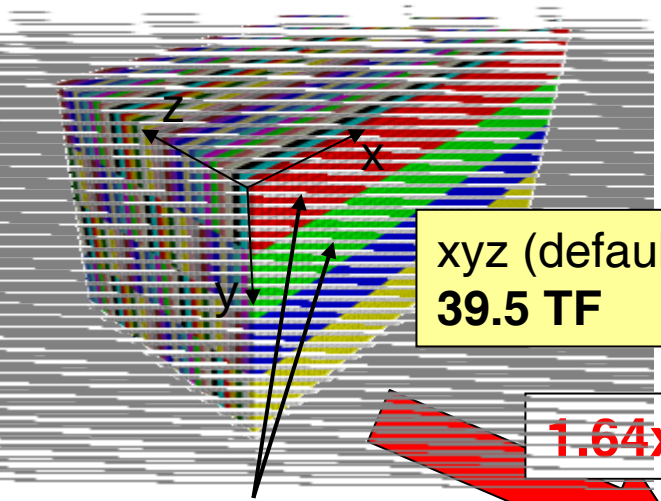





UCDAVIS
UNIVERSITY OF CALIFORNIA

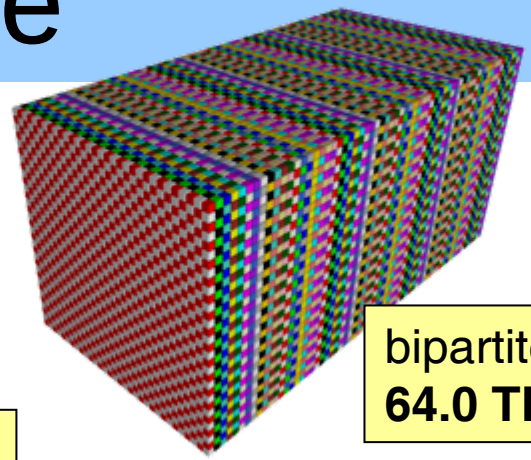
# Node mapping impacts performance

65536 nodes, in a 64x32x32 torus

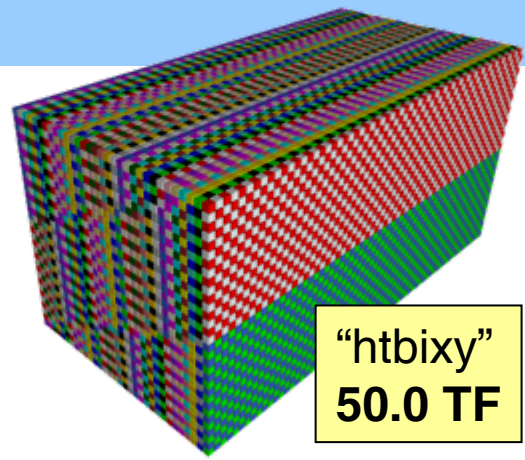



xyz (default)
**39.5 TF**

512 tasks per MPI subcommunicator

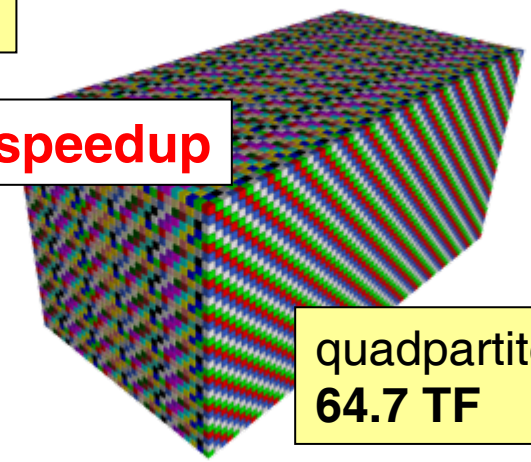


bipartite
**64.0 TF**

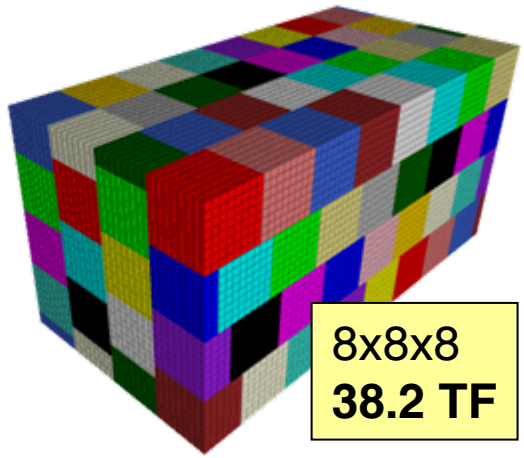


"htbixy"
**50.0 TF**

**1.64x speedup**



quadpartite
**64.7 TF**

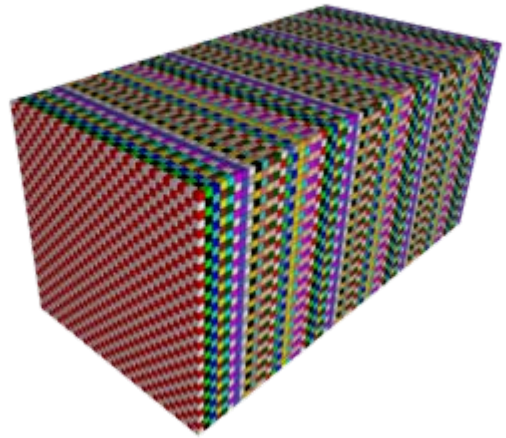


8x8x8
**38.2 TF**

**Node mapping significantly affects performance**

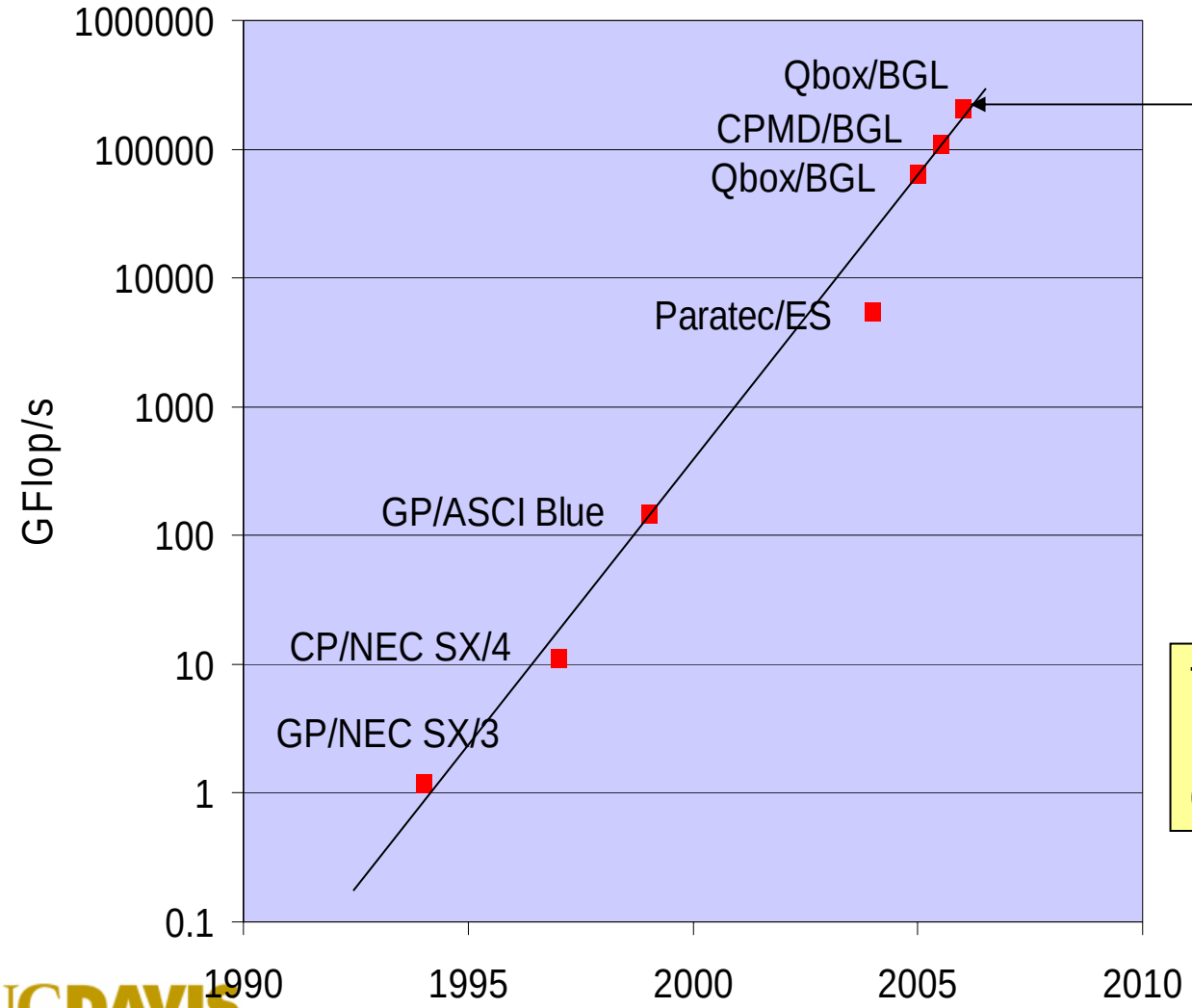
# Node Mapping Optimization

- Analysis of the MPI tree broadcast algorithm in sub-communicators
  - The performance of MPI_Bcast is unsatisfactory for checkerboard mappings: the tree algorithm results in communication bottlenecks
  - Using a space-filling curve to assign tasks to nodes improves performance
  - performance gain: ~7 TFlops



- Modifications of the BLACS communication library
  - unnecessary Type_commit operations were removed in the BLACS
  - performance gain: ~3 TFlops

# History of First-Principles MD performance

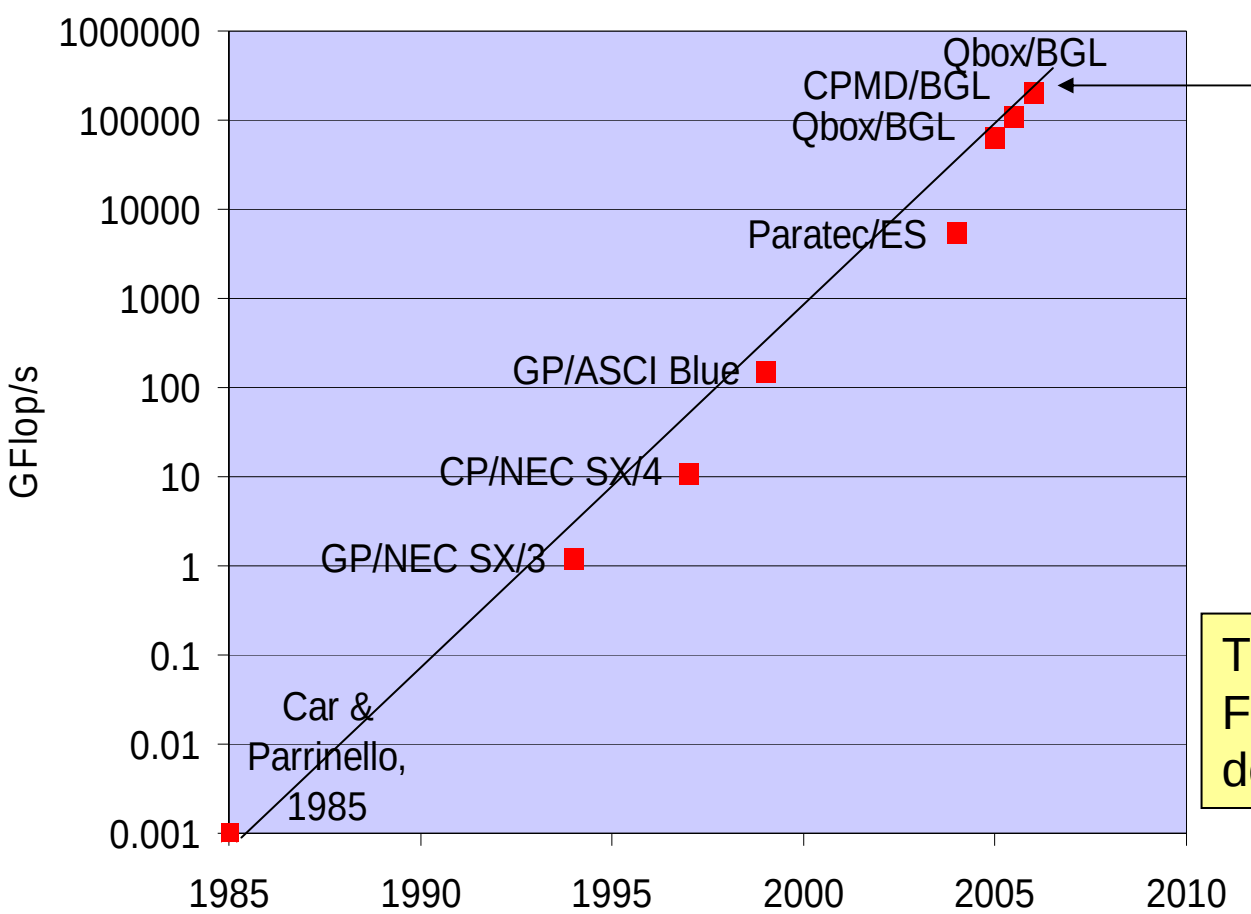



207.3 TFlops sustained

The performance of FPMD codes has doubled every 8 months

UCDAVIS UNIVERSITY OF CALIFORNIA

# History of FPMD performance




207.3 TFlops sustained

The performance of FPMD codes has doubled every 8 months

# Computational cost of first-principles simulations

- The computational cost of solving the Kohn-Sham equations is O($N^3$)

$$\begin{cases} -\Delta\varphi_i + V(\rho,\mathbf{r})\varphi_i = \varepsilon_i\varphi_i \quad i = 1\ldots N_{\text{el}} \\ V(\rho,\mathbf{r}) = V_{\text{ion}}(\mathbf{r}) + \int \dfrac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\,d\mathbf{r}' + V_{\text{XC}}(\rho(\mathbf{r}),\nabla\rho(\mathbf{r})) \\ \rho(\mathbf{r}) = \sum_{i=1}^{N_{\text{el}}} |\varphi_i(\mathbf{r})|^2 \\ \int \varphi_i^*(\mathbf{r})\,\varphi_j(\mathbf{r})\,d\mathbf{r} = \delta_{ij} \end{cases}$$

# $O(N^3)$ algorithms onlarge parallel platforms

- Storage of electronic wavefunctions: $O(N^2)$ (N wavefunctions described in a volume O(N))
- Number of operations: $O(N^3)$ (orthogonalization)

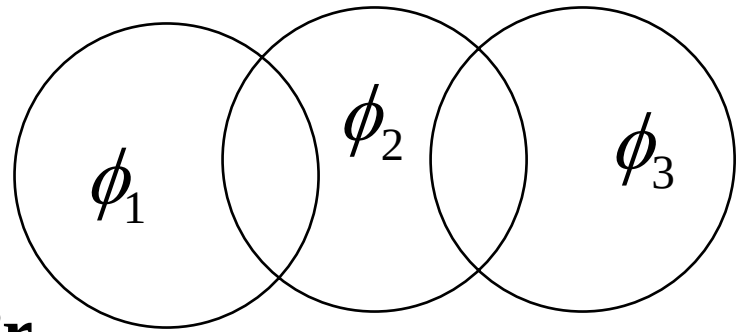
| Problem size | # ops | # CPUs | ops/CPU | storage | storage/CPU |
|---|---|---|---|---|---|
| N | $N^3$ | N | $N^2$ | $N^2$ | N |
| N | $N^3$ | $N^2$ | N | $N^2$ | 1 |
| N | $N^3$ | $N^3$ | 1 | $N^2$ | 1/N |

# Linear-scaling methods

- **Achieve *linear scaling: O(N)* operations**
- **Introduce approximations to reduce the computational cost from $O(N^3)$ to $O(N)$.**
- **Several approaches proposed in the past 10 years**
- **Most successful approach: represent the solutions of the Kohn-Sham equations in terms of non-orthogonal, localized functions.**
- **An important goal is *controlled accuracy*, i.e.**
  - **Simple parameters (e.g. grid spacing) to control numerical accuracy**
  - **As robust as $O(N^3)$ methods**

# Linear-scaling methods

- **Domain decomposition approach: wavefunctions are localized in spherical, overlapping domains.**



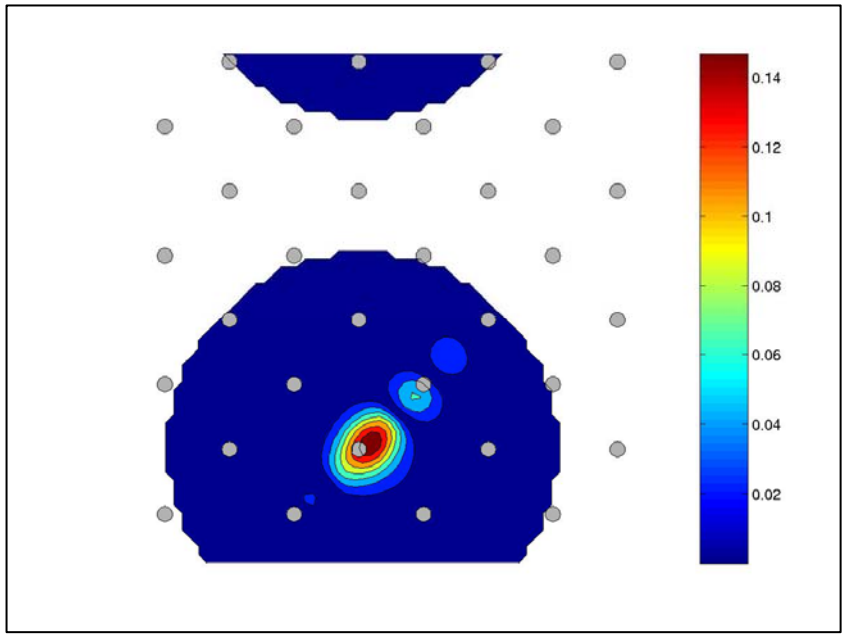

$$S_{ij} = \left\langle \phi_i \middle| \phi_j \right\rangle = \int_{\Omega} \phi_i^*(\mathbf{r})\, \phi_j(\mathbf{r})\, d^3\mathbf{r}$$

$$H_{ij} = \left\langle \phi_i \middle| H\phi_j \right\rangle = \int_{\Omega} \phi_i^*(\mathbf{r})\, H\phi_j(\mathbf{r})\, d^3\mathbf{r}$$

$$E(Y) = \mathrm{tr}(S^{-1}Y^T HY) \quad Y \in R^{M\times N} \quad S = Y^T Y$$

# Localization of orbitals

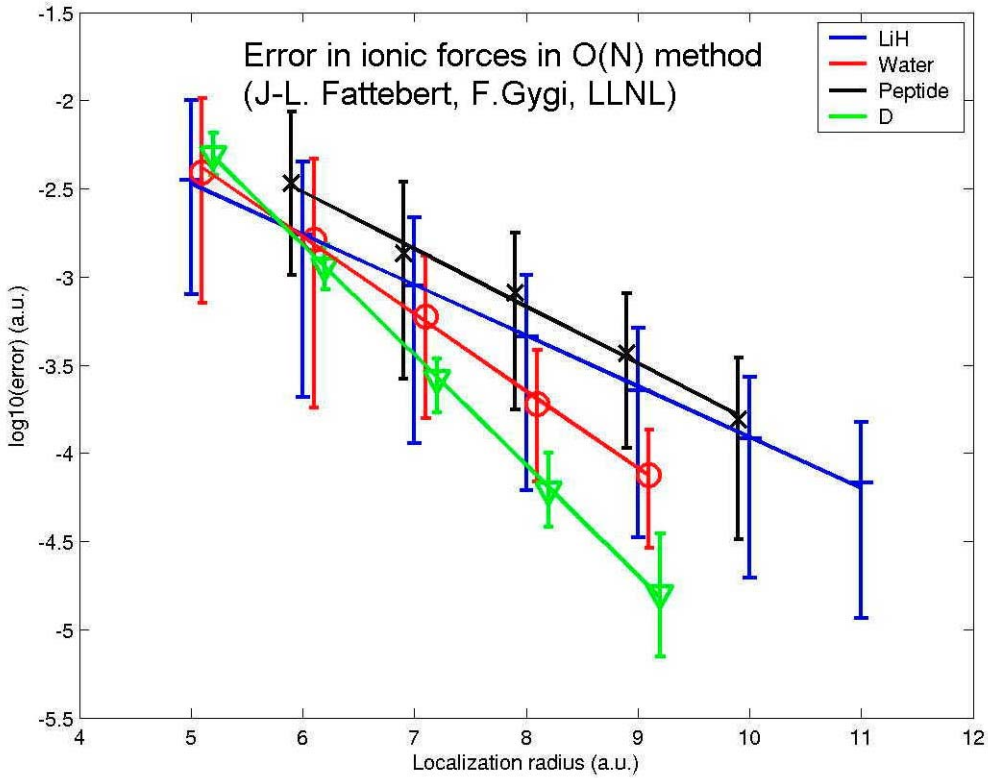
- Spherical domains are attached to atoms or bonds




- "Linear-Scaling First-Principles Molecular Dynamics with Controlled Accuracy", J.L.Fattebert and F.Gygi, Comp. Phys. Comm. *162*, 24 (2004).

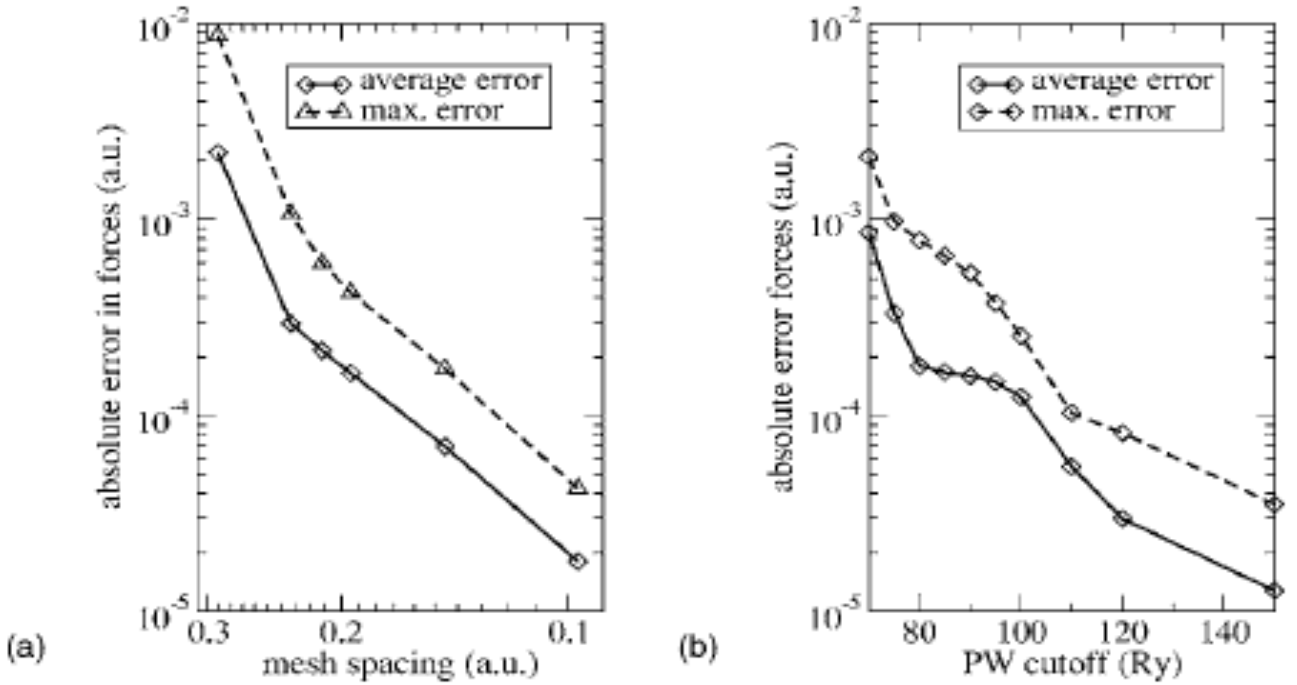
# *O(N)* with controlled accuracy

- Errors in computed ionic forces are decaying exponentially for large localization radii
- Errors are computed by comparison with *O(N³)* method with same numerical approximations



# O(N) with controlled accuracy

- The accuracy of ionic forces is critical for molecular dynamics simulations



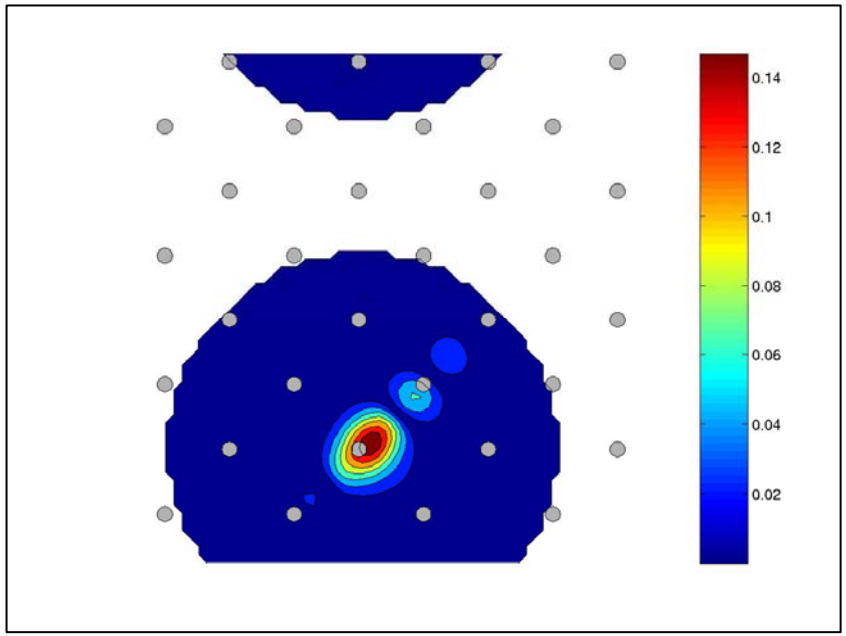

Reference: 400Ry Plane-wave calculation

$H_2O$, 32 molecules

J.L. Fattebert and F. Gygi, Phys. Rev. B 73, 115124 (2006)

# MD: using adaptive localization centers (ALC)

- Spherical domains *move* during the MD simulation
- The positions of the centers are recalculated at each time step using the center of charge of each orbital




J.L. Fattebert and F. Gygi, Phys. Rev. B 73, 115124 (2006)

# O(N) Molecular Dynamics

- $H_2O$ 32 molecules MD simulation with adaptive localization centers

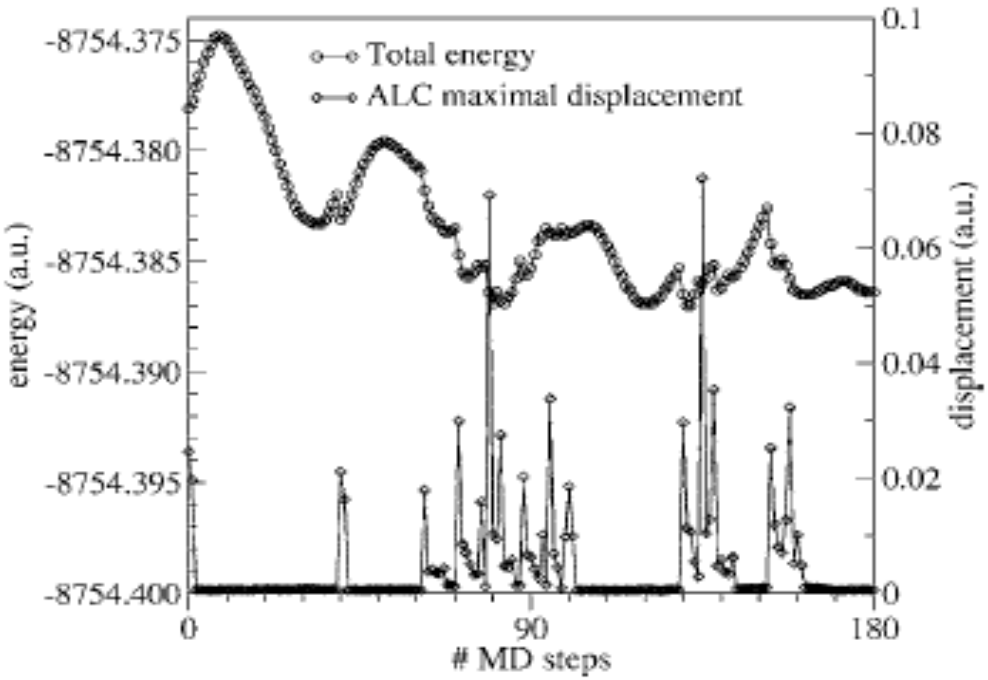



FIG. 2. Total energy and ALC displacements during molecular dynamics simulation of water (512 molecules) at 300 K with localization radius of 9 Bohr.

# Controlling energy drift in MD

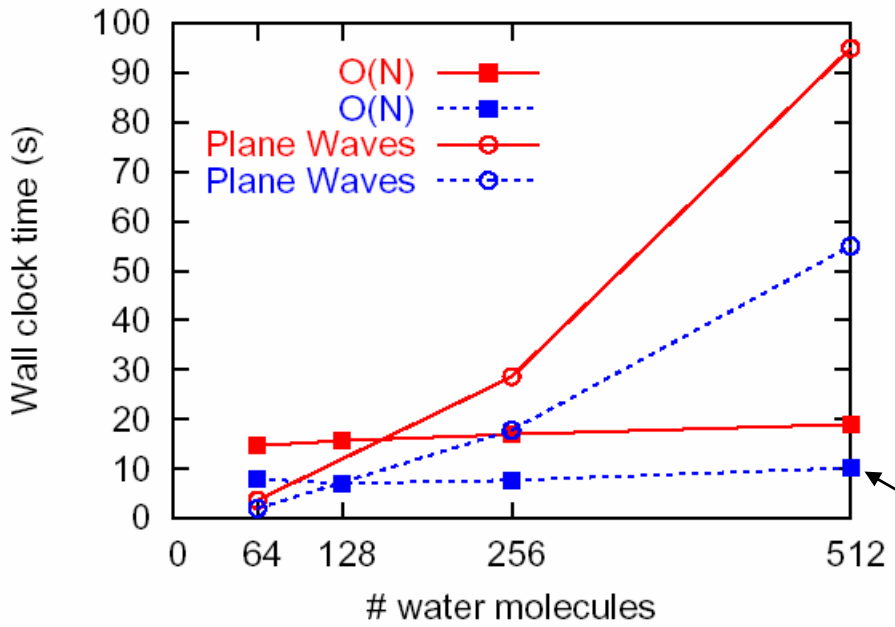
- $H_2O$ 32 molecules MD simulation

TABLE I. Measure of energy drift and number of SC steps required for convergence.

| Localization radius (Bohr) | Energy drift (mHa/at/ps) | No. SC iteration/ MD step |
|---|---|---|
| 8 | –0.222 (–47 K/ps) | 29 |
| 9 | –0.103 (–22 K/ps) | 23 |
| 10 | +0.001 (0 K/ps) | 14 |

# Comparison with O(N³): simulations of liquid water

Timings for 1 electronic step (processors Itanium 2, 1.4 GHz, Quadrics switch)




Red: 64 molecules/ 48 CPUs
(48 to 384 CPUs)

Blue: 64 molecules/ 96 CPUs
(96 to 768 CPUs)

$(H_2O)_{512}$ in 10 s./iteration

# Linear-scaling methods

- Review articles:
  - G.Galli, “Linear Scaling Methods For Electronic Structure Calculations and Quantum Molecular Dynamics Calculations”, *Current Opinion in Solid State and Materials Science*, 1, 864 (1996).
  - S.Goedecker, “Linear Scaling Electronic Structure Methods”, *Rev. Mod. Phys*. 71, 1085 (1999).
- Our recent work on *O(N)* with controlled accuracy
  - J.L.Fattebert and F.Gygi, Comp. Phys. Comm. *162*, 24 (2004).
  - J.L. Fattebert and F. Gygi, Phys. Rev. B 73, 115124 (2006)

# Linear-scaling methods

- Outstanding remaining issues
  - The choice of localization radii is difficult *a priori*
    - Some orbitals may require larger radii than others
    - The description of unoccupied orbitals is more difficult than occupied orbitals
  - The description of metallic systems is not satisfactory
  - The initial choice of localization centers is not obvious
    - Use atoms in some systems, bonds in others
  - Moving localization centers in a smooth, continuous way

More research is needed to develop linear-scaling electronic structure methods that are suitable for MD simulations

# Summary

- High-performance implementations of First-Principles Molecular Dynamics can be developed on large scale computers (up to 128k CPUs)
- Highly tuned single-node kernels are used
- Node mapping is critical for this application
- Linear scaling (O(N)) FPMD is the subject of more investigations to achieve *controlled accuracy*

Thanks to:
- Lawrence Livermore National Laboratory
- DOE ASC
- University of California Lawrence Livermore National Laboratory

Contract W-7405-Eng-48