Improved Parallelization of Quantum Espresso

Taylor Barnes
Grace Hopper Postdoctoral Fellow
NERSC

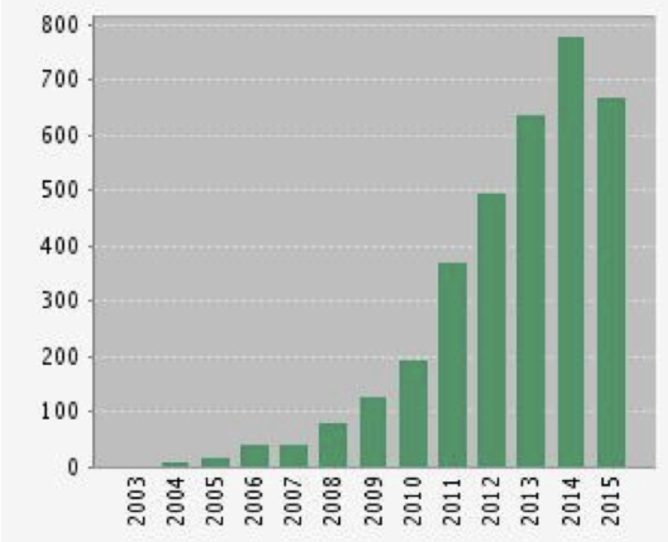
Collaborators: Jack Deslippe, Paul Kent, David Prendergast, and Andrew Canning

Introduction

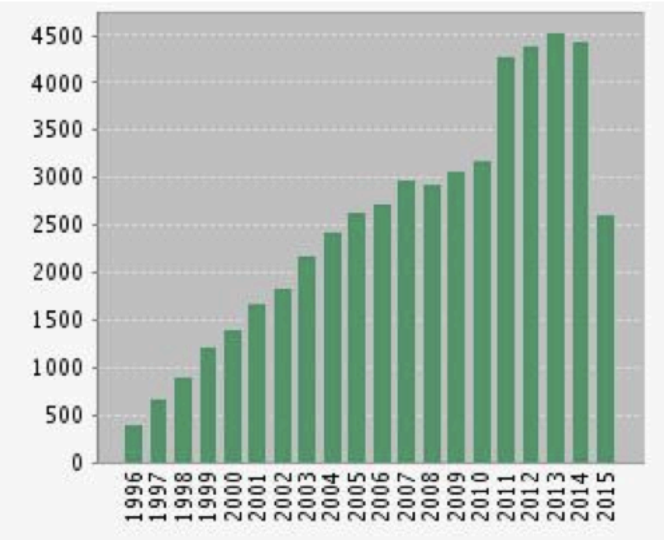
- **Quantum Espresso**
  - ➢Quantum Espresso is a plane-wave DFT code for performing nanoscale simulations
  - ➢We are especially interested in the performance of calculations that incorporate exact exchange

- **Exact Exchange Citations**

HSE



B3LYP



- **Relevance**
  - ➢Important for accurate representation of charge separation/charge transfer
  - ➢Especially desirable for:
    - Electrochemistry at interfaces
    - Batteries
    - Photovoltaic cells
    - Solvated ions

Introduction

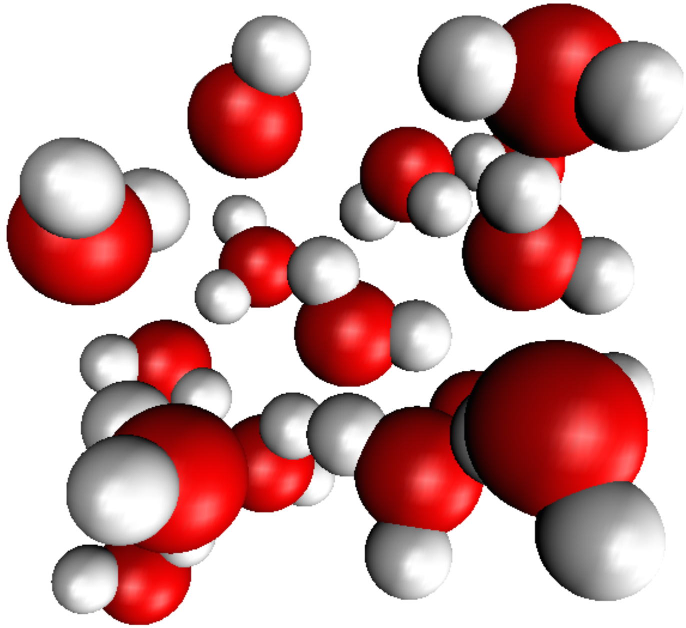
- **Exact Exchange Theory**
  - ➢The primary cost associated with exact exchange is the evaluation of a set of two-electron integrals:

$$K_j \psi_i(x_1) = \left[\int \frac{\psi_i(x_2)\psi_j(x_2)}{|x_2 - x_1|} \, dx_2 \right] \psi_j(x_1)$$

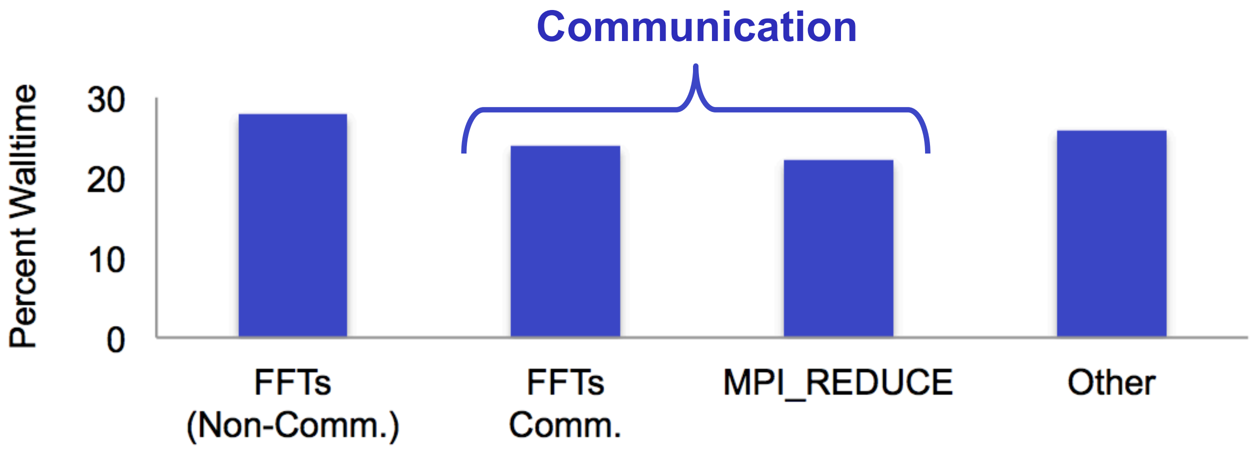
  - ➢QE performs these calculations using an internal copy of the FFTW library.

- **Parallelization Strategy**
  - ➢Within nodes – parallelize over FFTs
  - ➢Between nodes – parallelize over bands

- **The Test Calculations**
  - ➢All profiling was performed on Edison, NERSC's Cray XC30.
  - ➢We used a test system of 16 water molecules:



Performance



- **Communication Issues**
  - ➢Only 35% of the FFT communication time is actually spent in MPI calls
  - ➢MSGSIZ_MAX – legacy restriction on the maximum message size
  - ➢Legacy synchronization calls
  - ➢Result is communicated in real space, even though the G-space grid is ~1/20 the size

Do i=1, number_of_bands

$FFT^{-1}[\ \psi_i(G)\ ] \rightarrow \psi_i(r)$

result(r) = 0

Do j=ibnd_start, ibnd_end

$\rho_{ij}(r) = \psi_i(r)\ \psi_j(r)$

$FFT[\ \rho_{ij}(r)\ ] \rightarrow \rho_{ij}(G)$

$v_{xx}(G) = 4\pi\rho_{ij}(G)/G^2$

$FFT^{-1}[\ v_{xx}(G)\ ] \rightarrow v_{xx}(r)$

$result(r) \mathrel{+}= v_{xx}(r)\ \psi_j(r)$

End Do

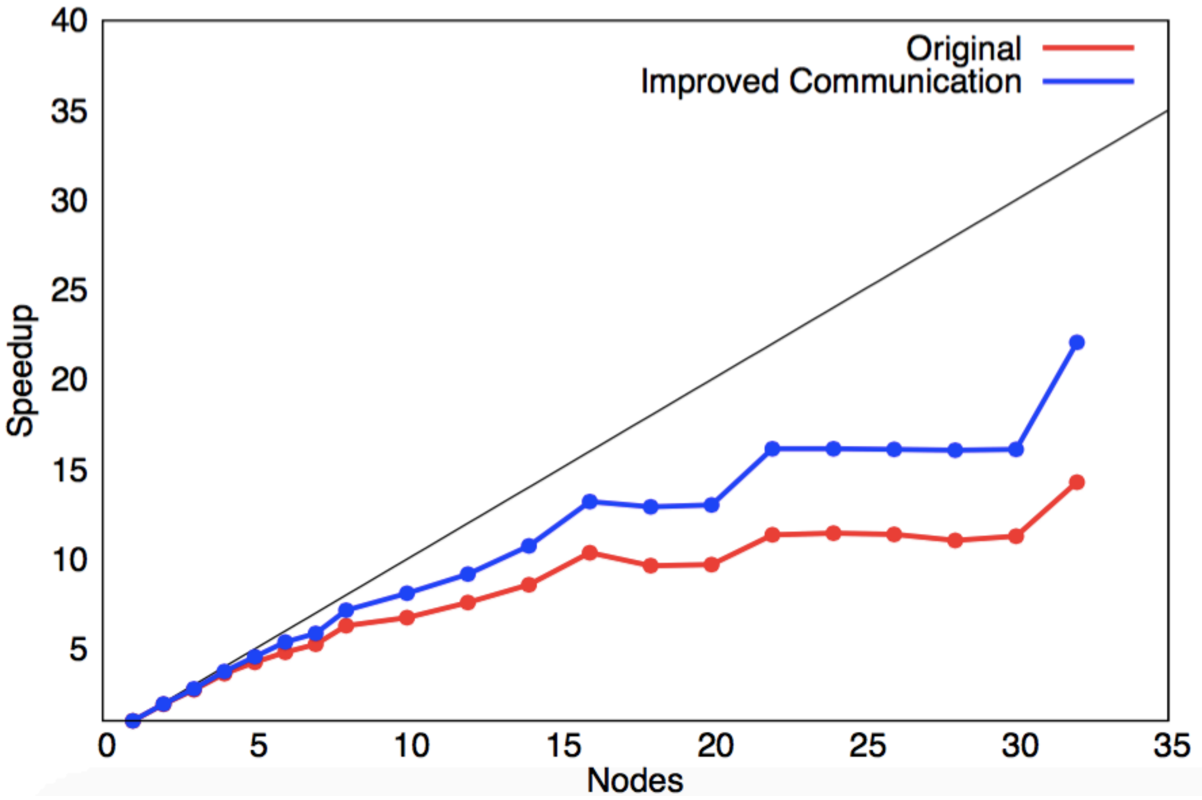
MPI_REDUCE(result)

$FFT[\ result(r)\ ] \rightarrow result(G)$

End Do

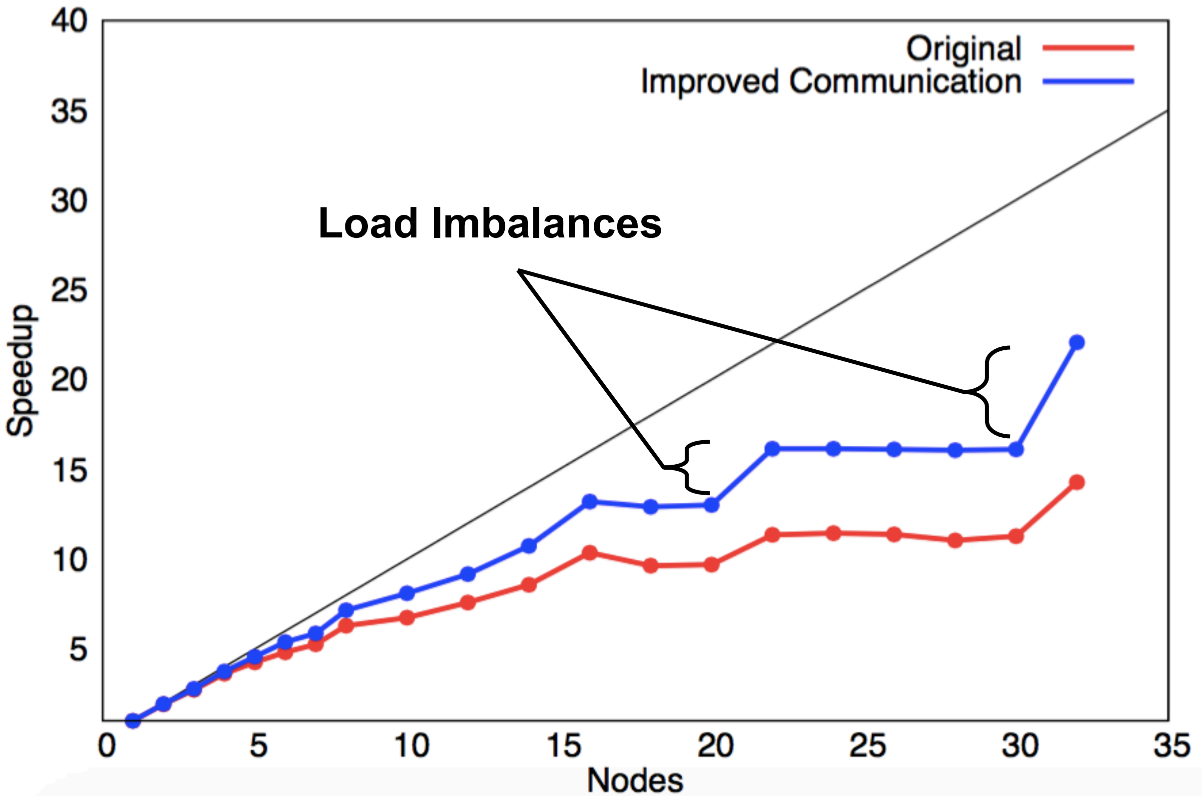
Code Improvements

- **Improved Communication**
  - Addressing the communication issues substantially improves the parallelization efficiency.
  - The efficiency exhibits discontinuities with respect to the number of nodes.



Code Improvements

- **Improved Communication**
  - Addressing the communication issues substantially improves the parallelization efficiency.
  - The efficiency exhibits discontinuities with respect to the number of nodes.
- **Load Imbalances**
  - Each node is assigned an integer number of bands.
  - As the number of nodes approaches the number of bands (64), load imbalances become increasingly problematic.



Can we parallelize over both loops in the exact exchange calculation (i.e., parallelization over band pairs)?

Code Improvements

- **Parallelization Over Band Pairs**
  - Our proof-of-concept parallelization offers several advantages:
    - Improved load balancing
    - Outer loop parallelization
    - Further improvements are possible
- **Additional Work**
  - For iterations in which a subset of bands have converged, our load balancing is currently suboptimal.
  - Every node still stores and communicates information about all bands.
  - Methods that exploit orbital localization may permit more revolutionary improvements.