Acceleration of Porous Media Simulations CUG 2011

Kirsten Fagnan, Michael J. Lijewski, George S. H. Pau and Nicholas J. Wright

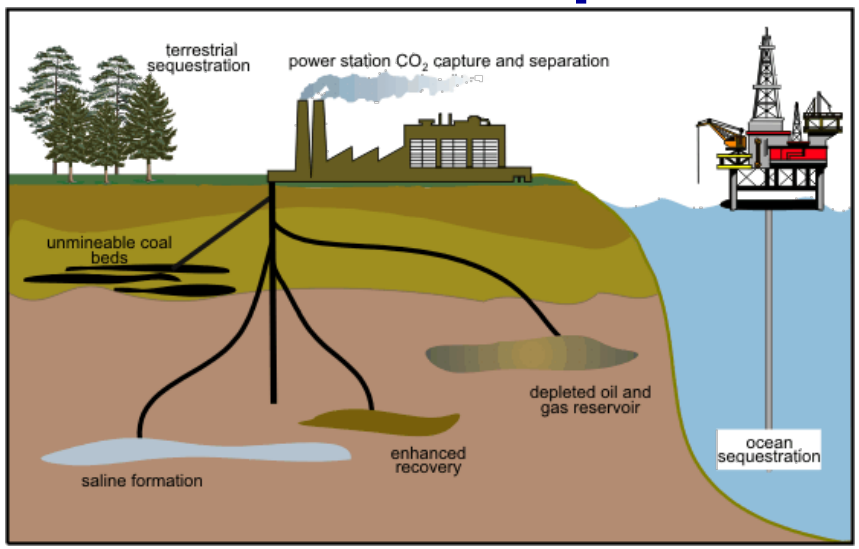
Lawrence Berkeley National Laboratory

Outline

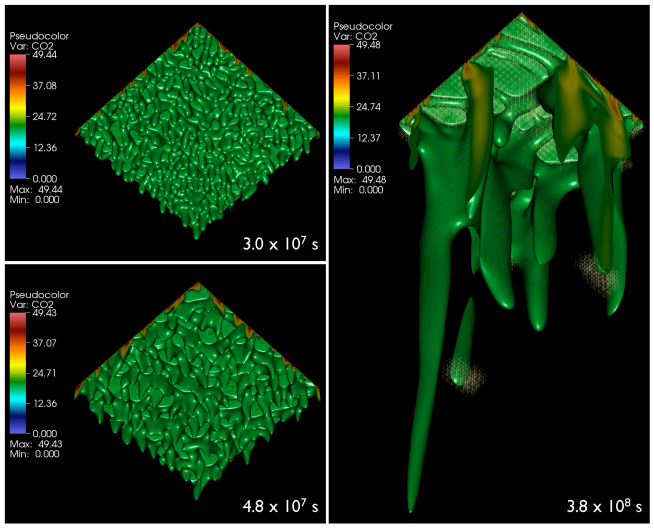
- Background on Porous Media, why are we interested in this problem
- Mathematical Model
- Computational Approach
- How long will it take to simulate out to 25 years?
- Summary and conclusions

Motivation

- **Fluid flow with chemical reactions in a porous material is found in a variety of geophysical processes, e.g.**
  - **Carbon sequestration**



Courtesy of:
http://blog.aapg.org/geodc/wp-content/uploads/2008/12/carbon-sequestration.gif



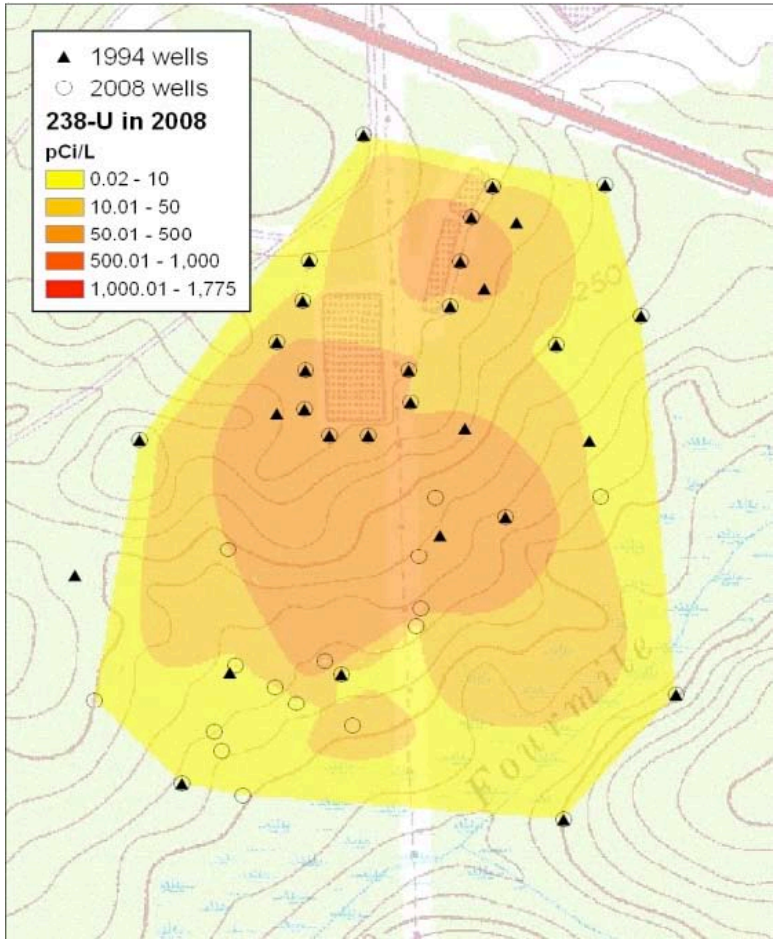
Calculation done by George Pau (LBNL) with PMAMR

Motivation

- **The DOE is also interested in modeling groundwater contamination**

This shows the progression of underground contaminants (Uranium!) at the F-basin site

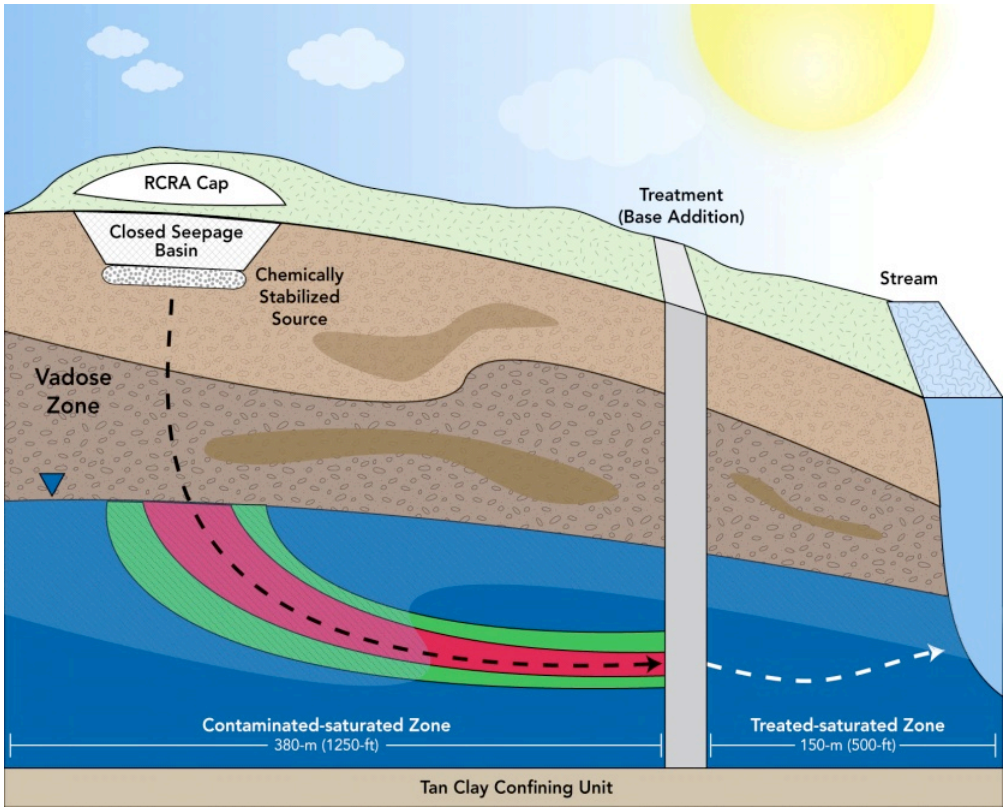
From the ASCEM demo document, 2010



Motivation

- **The DOE is also interested in modeling groundwater contamination**

Cartoon schematic of the computational domain of interest that we approximate in our calculations



From the ASCEM demo document, 2010

Mathematical Model

- **Equations of Interest**

$$\phi \frac{\partial \mathbf{n}}{\partial t} + \nabla \cdot \mathcal{F} = \nabla \cdot \phi D \rho \nabla \frac{\mathbf{n}}{\rho}$$

$$-\nabla \cdot \frac{\kappa}{\mu}(\nabla p - \rho \mathbf{g}) = \sum_{i=1}^{2} \frac{1}{\rho_i} \nabla \cdot \phi \rho D \nabla X_i$$

Mathematical Model (PDE types)

- **Equations of interest**

Hyperbolic!

Parabolic!

$$\phi\frac{\partial \mathbf{n}}{\partial t} + \nabla \cdot \mathcal{F} = \nabla \cdot \phi D \rho \nabla \frac{\mathbf{n}}{\rho}$$

$$-\nabla \cdot \frac{\kappa}{\mu}(\nabla p - \rho \mathbf{g}) = \sum_{i=1}^{2} \frac{1}{\rho_i} \nabla \cdot \phi \rho D \nabla X_i$$

Elliptic!

Computational Model

- **Implicit-pressure Explicit-saturation (IMPES) approach**
  - **Parabolic pressure terms are solved with an implicit multigrid solver => All-to-All communication across MPI tasks**
  - **Hyperbolic terms are solved with an explicit method (2nd order Godunov-type method) => only requires communication in ghost cells**

Adaptive Mesh Refinement

Allows us to use fine grids only around important spatial features (we use Berger-Oliger style AMR).

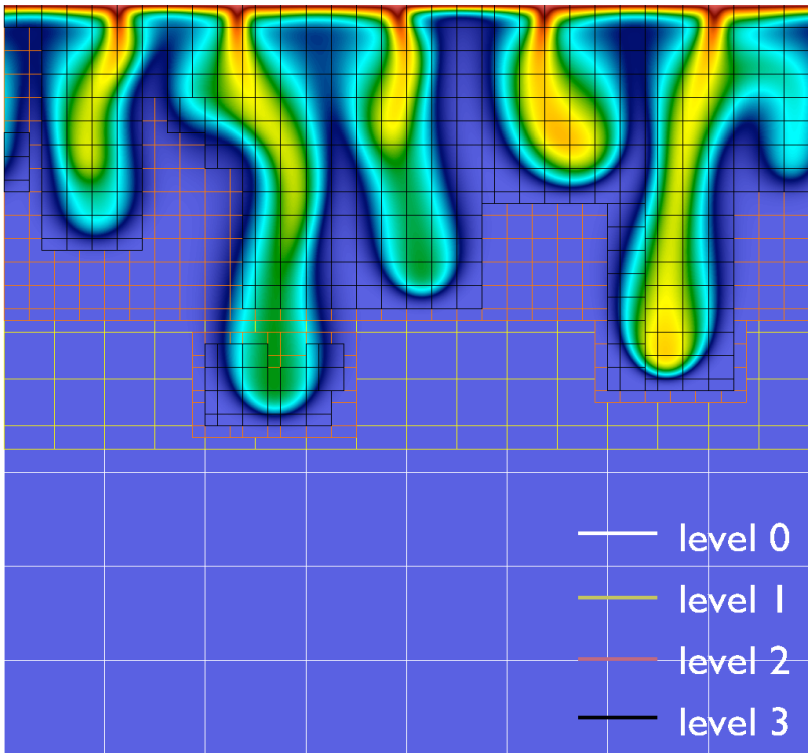


Figure: 2D calculation of fingering present in carbon sequestration – illustrates the use of AMR on Cartesian grids

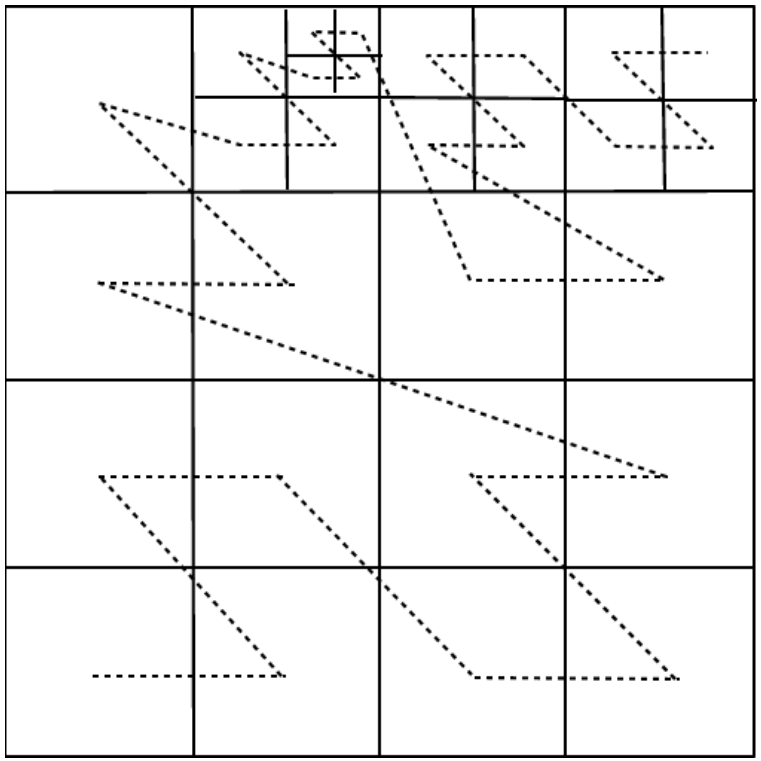


Figure: Load balancing is achieved through the use of a space-filling curve

Chemistry Solver – ASCEM project

- The geochemistry solver that models the interaction of reactants present in the fluid is called point-by-point with data local to each computational grid cell.

How long will it take to simulate out to 25 years?

- Current time step restriction on a grid used to resolve the finest spatial scales of the groundwater contaminant problem: dt ~ 300 seconds
- **25 years/dt ~2,628,000 computational steps!**
- Note: implicit methods do not face the same time-step restriction, but fail to resolve the front of the plume due to numerical dissipation

How can we speed this up?

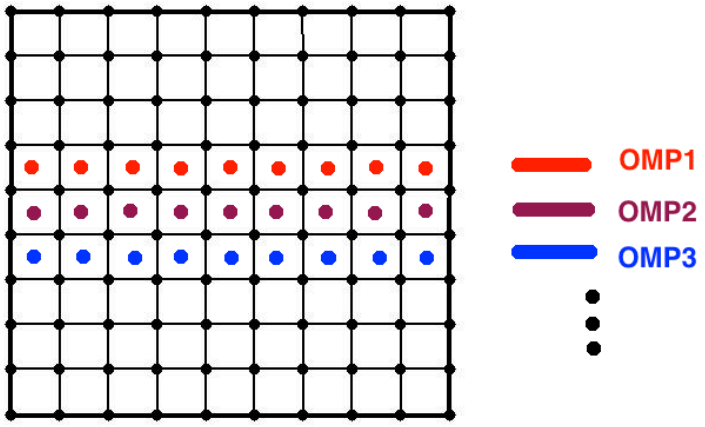
- BoxLib is already parallelized with OpenMP and MPI, a legacy code that is fairly well optimized. (scaling plot without chemistry)
- Profiling of the code indicated that more than 40% percent of the time was being spent in the ASCEM chemistry solver.

OpenMP for Porous Media

- AMR is 'hard' to load balance
  - Minimize the number of MPI tasks
- Chemistry is embarrassingly parallel
  - Takes 40% of runtime*
- Hopper has 24 cores per node and less memory than Franklin
- This implies that we should use OpenMP to speed things up

Chemistry+Hopper => OpenMP

- The chemistry solves were already being spread out across MPI tasks
- The structure of Hopper made threading a logical option
  - embarassingly parallel, but chemistry solver was not threaded or optimized

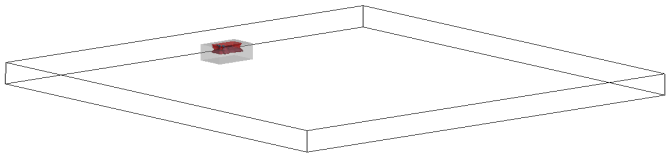


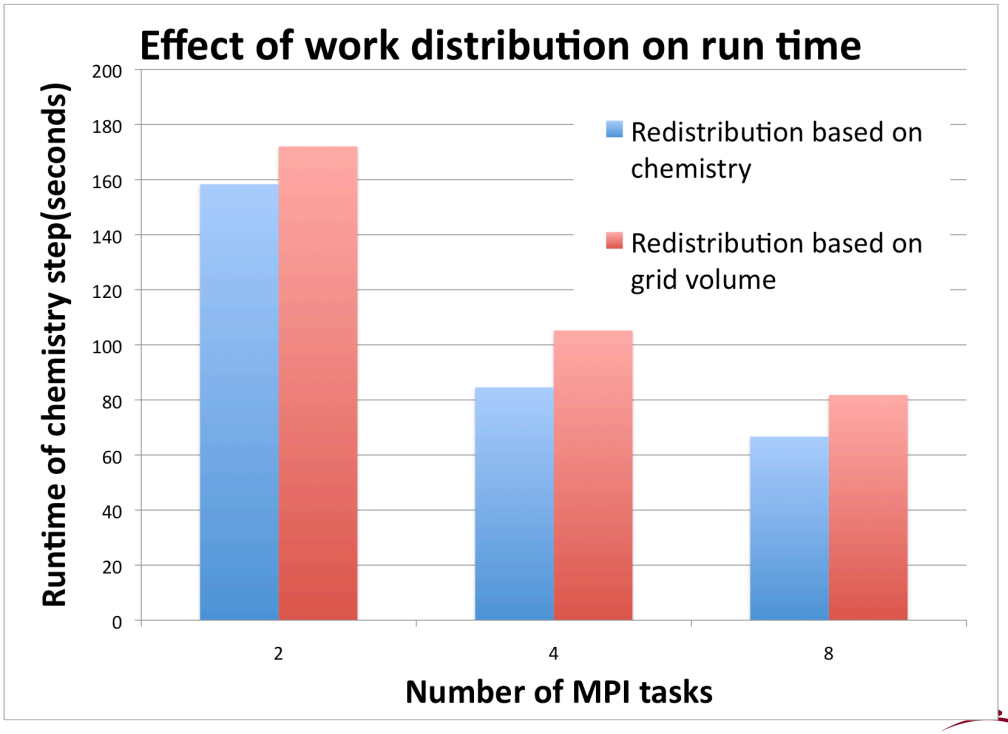
Chemistry code was not optimized!

- When we initially ran the threaded code, it was slower. More threads => longer run time
- We explored the chemistry solver we were using and found that there were several issues – passing large arrays by value, lots of exceptions and no optimization flags for the compiler
- Optimization of this code meant that the chemistry was reduced to %20 of the runt time

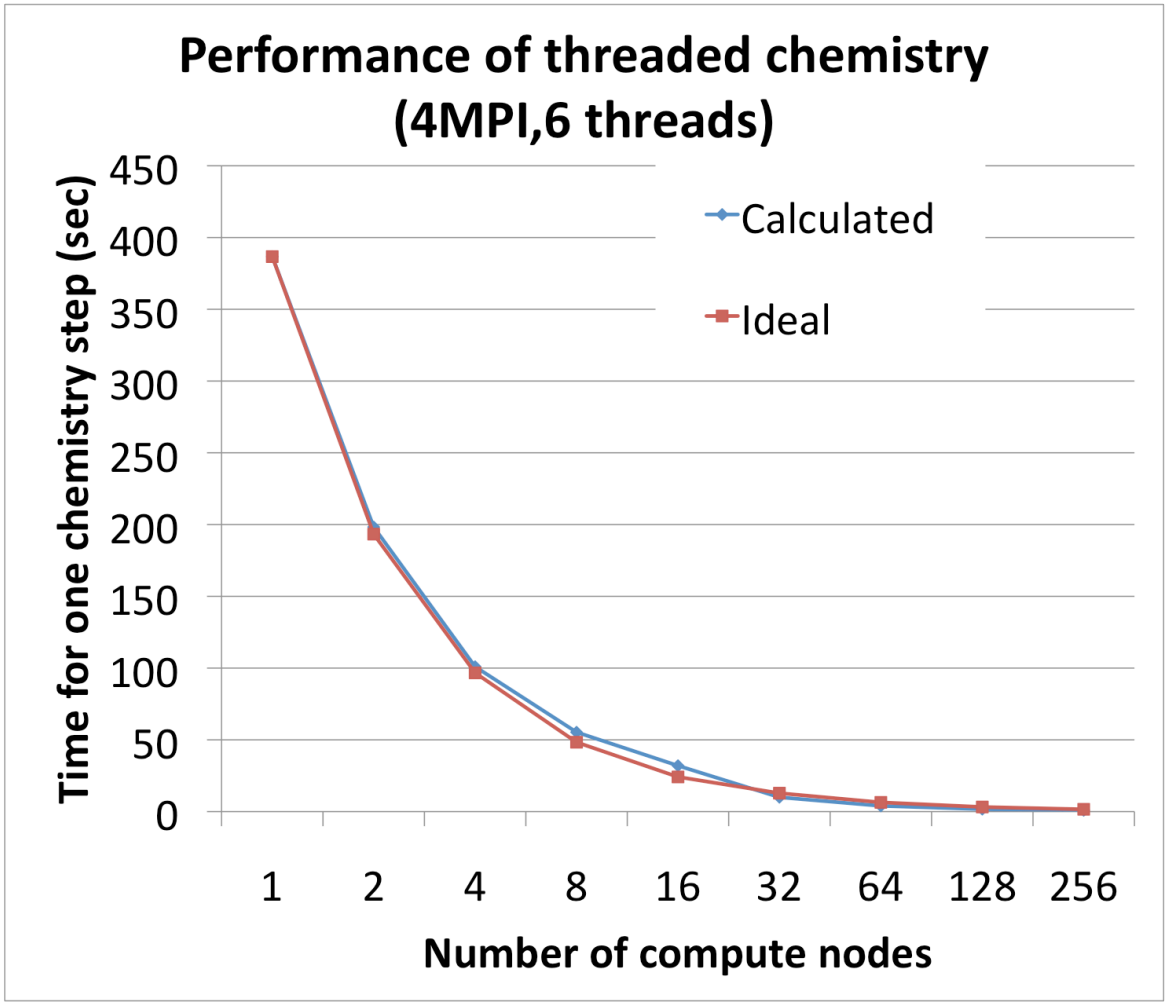
System Simulated

- Problem size: nx=128 ny=128 nz = 128, max grid 64^3
- 2 levels of refinement
- 32 chemical species
  - Grids are distributed based on the difficulty of the chemistry solve

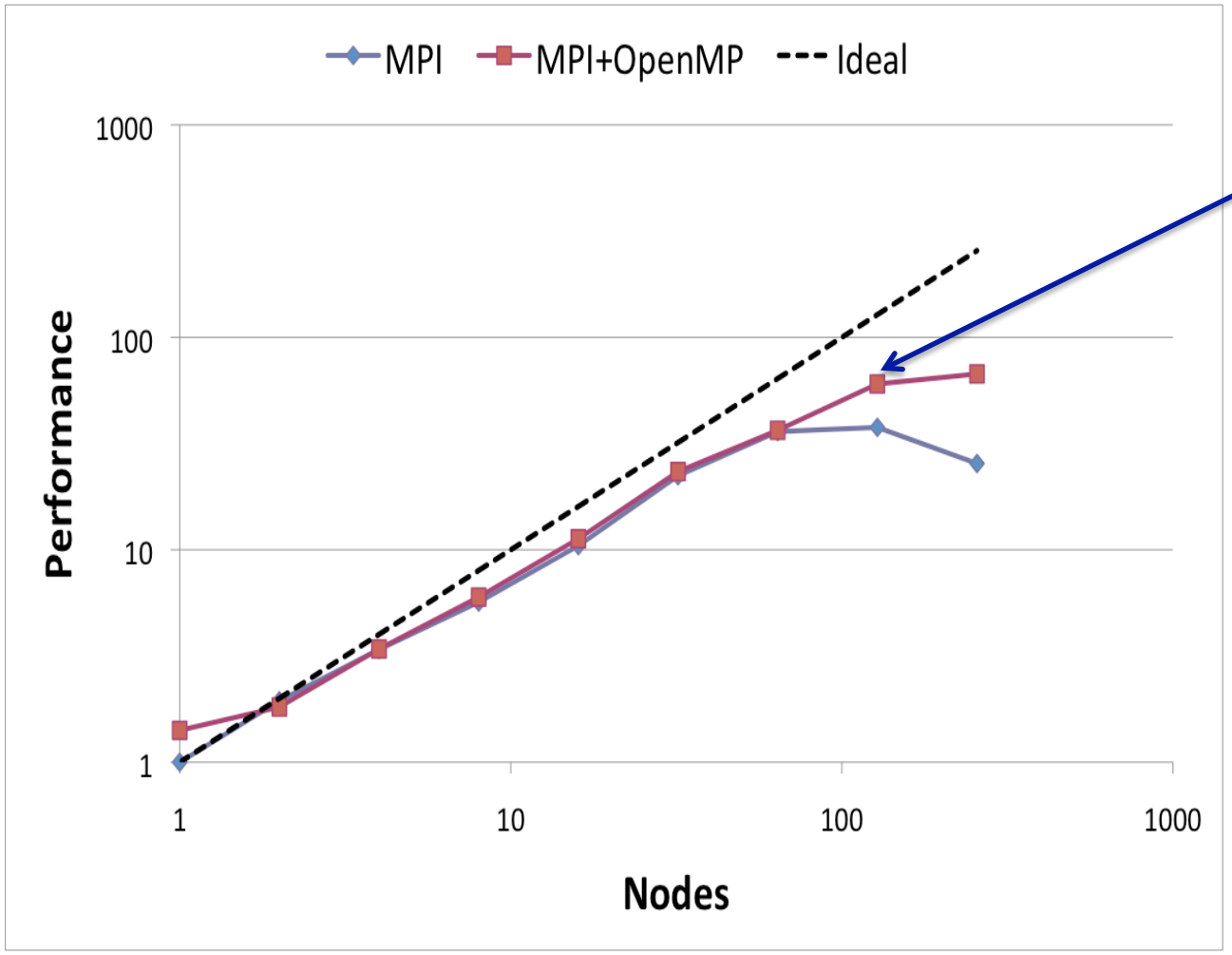




Chemistry Speedup

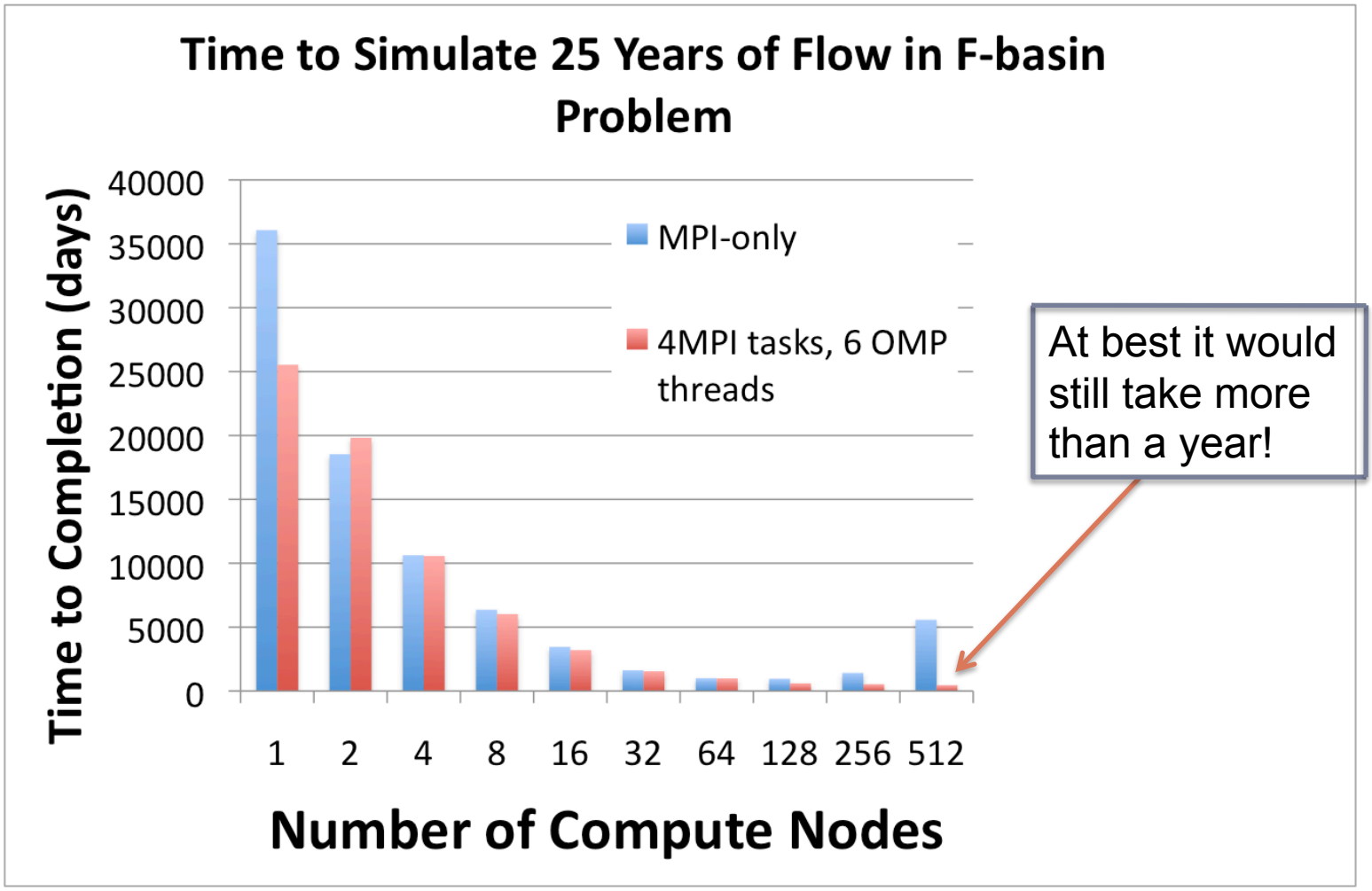


MPI vs. MPI/OpenMP



At 128 nodes MPI+OpenMP starts to outperform MPI-only

How long to simulate 25 years of a realistic problem



Things you may find interesting

- PGI compiler fails to work with threaded C++ code that passes arrays by value instead of by reference (show plot demonstrating that it takes longer with threads)
- This is not good software design, but it only failed to work when using PGI
- Bug submitted to the PGI compiler group

Summary

- We need to simulate out to 25 computational years in order to produce meaningful results
- MPI alone provides insufficient speed-up when modeling large chemical systems
- The introduction of OpenMP allows us to calculate to 25 years in roughly half the time of MPI alone, but it's still not fast enough
- Chemistry solves are now extremely fast, but Multigrid is proving to be the next bottleneck
- We are also working on an algorithmic approach that would allow us to take longer time steps

Acknowledgements

- DOE ARRA funding
- George Pau, Michael Lijewski and Nicholas Wright
- John Shalf, John Bell and Alice Koniges