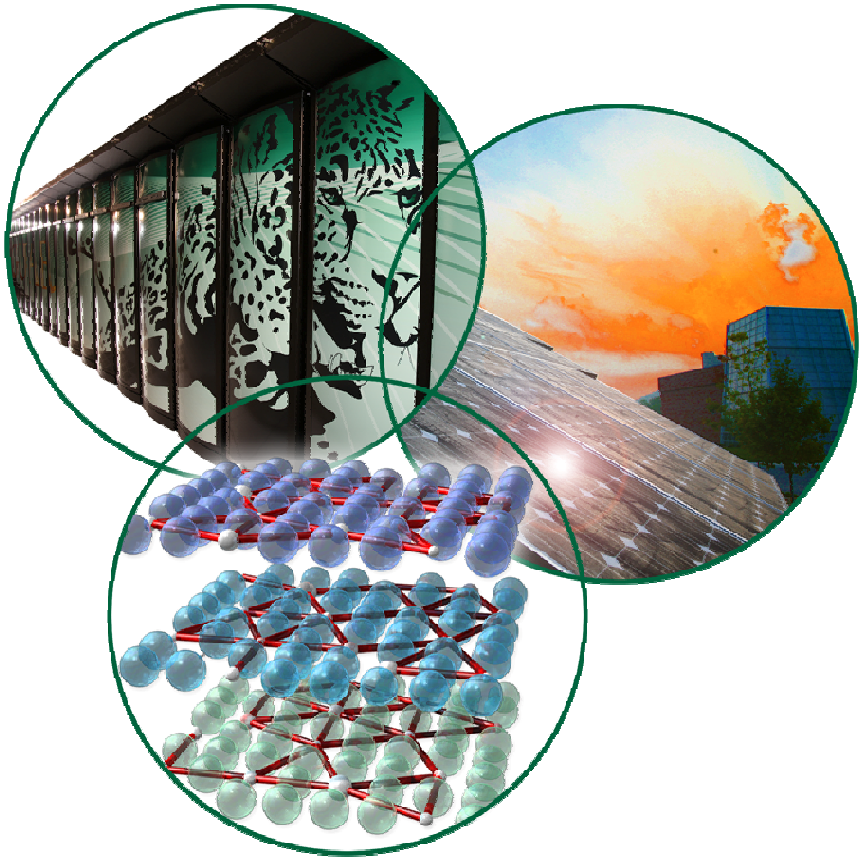
# Modeling Subsurface Reactive Flows with PFLOTRAN + PETSc Using Leadership-Class Computing




**Richard Tran Mills**

**Computational Earth Sciences Group**

**Computer Science and Mathematics Division**

**Oak Ridge National Laboratory**



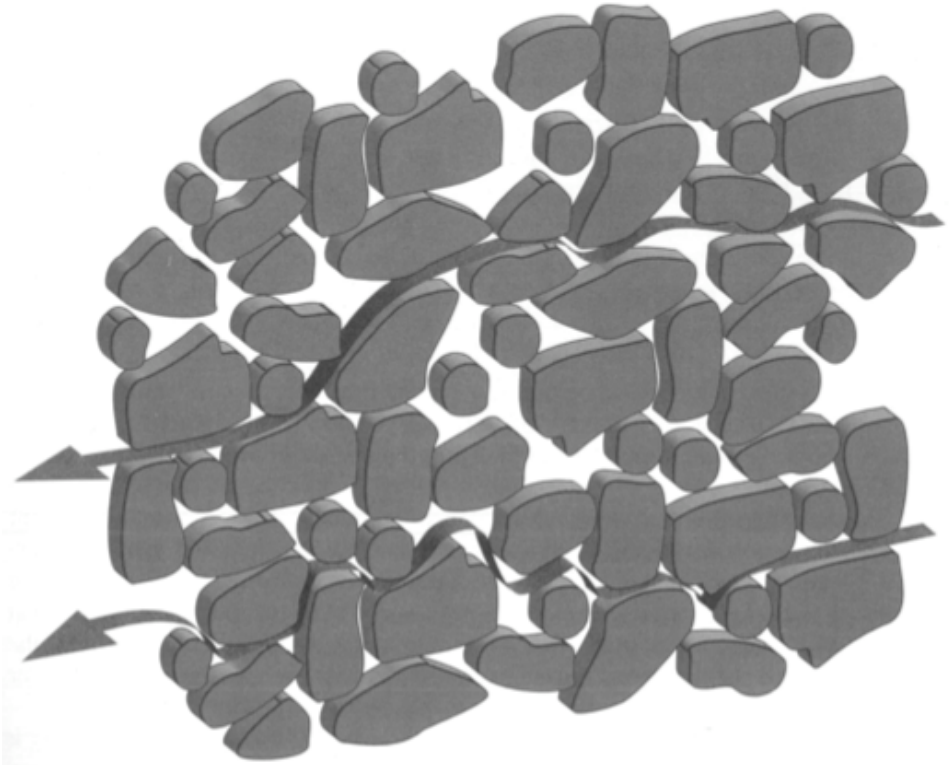
Managed by UT-Battelle
for the Department of Energy

# Outline

- Subsurface flow and reactive transport
- Governing eqns, discretization, solution
- Code architecture and PETSc framework
- Hanford 300 Area application and performance
- Ongoing research and associated challenges
- Summary and future directions

# Porous media flow

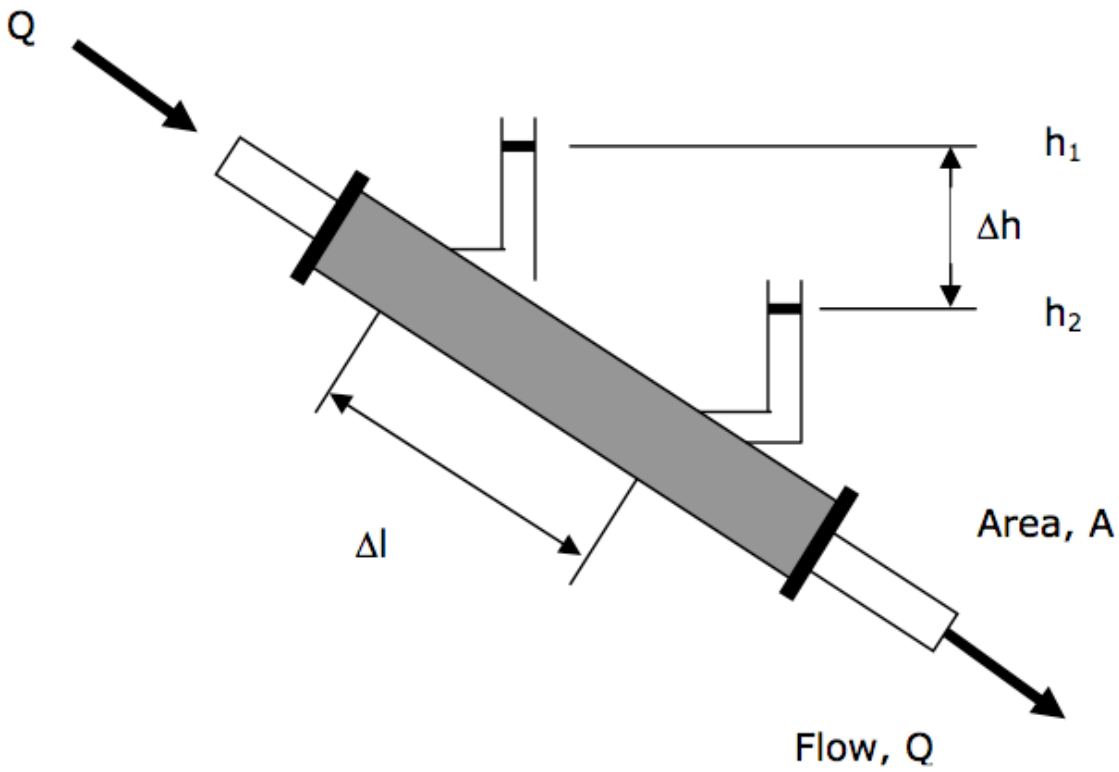
- **Continuity equation (mass conservation)**
- **Darcy's law in place of momentum eqn.**



(Figure from Hornberger et al., 1998, *Elements of Physical Hydrology*)

# Porous media flow

- **Continuity equation (mass conservation)**
- **Darcy's law in place of momentum eqn.**




$$q = \frac{Q}{A} = -K\nabla h$$

(Figure from *Groundwater Hydrology* lecture notes, Prof. Charles Harvey, MIT)

# PFLOTRAN governing equations

Mass Conservation: Flow Equations

$$\frac{\partial}{\partial t}(\phi s_\alpha \rho_\alpha X_i^\alpha) + \nabla \cdot \left[ q_\alpha \rho_\alpha X_i^\alpha - \phi s_\alpha D_i^\alpha \rho_\alpha \nabla X_i^\alpha \right] = Q_i^\alpha$$

$$q_\alpha = -\frac{k k_\alpha}{\mu_\pi} \nabla (p_\alpha - W_\alpha \rho_\alpha g z) \qquad p_\alpha = p_\beta - p_{c,\alpha\beta}$$

Energy Conservation Equation

$$\frac{\partial}{\partial t}\left[ \phi \sum_\alpha s_\alpha \rho_\alpha U_\alpha + (1-\phi)\rho_r c_r T \right] + \nabla \cdot \left[ \sum_\alpha q_\alpha \rho_\alpha H_\alpha - \kappa \nabla T \right] = Q_e$$

Multicomponent Reactive Transport Equations

$$\frac{\partial}{\partial t}\left[ \phi \sum_\alpha s_\alpha \Psi_j^\alpha \right] + \nabla \cdot \left[ \sum_\alpha \Omega_\alpha \right] = -\sum_m v_{jm} I_m + Q_j$$

*Total Concentration*

$$\Psi_j^\alpha = \delta_{\alpha l} C_j^\alpha + \sum_i v_{ji} C_i^\alpha$$

Total Solute Flux

$$\Omega_j^\alpha = (-\tau \phi s_\alpha D_\alpha \nabla + q_\alpha)\Psi_j^\alpha$$

Mineral Mass Transfer Equation

$$\frac{\partial \phi_m}{\partial t} = V_m I_m \qquad \phi + \sum_m \phi_m = 1$$

# PFLOTRAN governing equations

Mass Conservation: Flow Equations

$$\frac{\partial}{\partial t}(\phi s_\alpha \rho_\alpha X_i^\alpha) + \nabla \cdot \left[ q_\alpha \rho_\alpha X_i^\alpha - \phi s_\alpha D_i^\alpha \rho_\alpha \nabla X_i^\alpha \right] = Q_i^\alpha$$

$$q_\alpha = -\frac{k k_\alpha}{\mu_\pi} \nabla (p_\alpha - W_\alpha \rho_\alpha g z)$$

$$p_\alpha = p_\beta - p_{c,\alpha\beta}$$

Darcy's law
(homogenized momentum eq.)

Energy Conservation Equation

$$\frac{\partial}{\partial t}\left[ \phi \sum_\alpha s_\alpha \rho_\alpha U_\alpha + (1-\phi)\rho_r c_r T \right] + \nabla \cdot \left[ \sum_\alpha q_\alpha \rho_\alpha H_\alpha - \kappa \nabla T \right] = Q_e$$

Multicomponent Reactive Transport Equations

$$\frac{\partial}{\partial t}\left[ \phi \sum_\alpha s_\alpha \Psi_j^\alpha \right] + \nabla \cdot \left[ \sum_\alpha \Omega_\alpha \right] = -\sum_m v_{jm} I_m + Q_j$$

*Total Concentration*

$$\Psi_j^\alpha = \delta_{\alpha l} C_j^\alpha + \sum_i v_{ji} C_i^\alpha$$

Total Solute Flux

$$\Omega_j^\alpha = (-\tau \phi s_\alpha D_\alpha \nabla + q_\alpha)\Psi_j^\alpha$$

Mineral Mass Transfer Equation

$$\frac{\partial \phi_m}{\partial t} = V_m I_m$$

$$\phi + \sum_m \phi_m = 1$$

# PFLOTRAN governing equations

Mass Conservation: Flow Equations

$$\frac{\partial}{\partial t}(\phi s_\alpha \rho_\alpha X_i^\alpha) + \nabla \cdot \left[ q_\alpha \rho_\alpha X_i^\alpha - \phi s_\alpha D_i^\alpha \rho_\alpha \nabla X_i^\alpha \right] = Q_i^\alpha$$

$$q_\alpha = -\frac{k k_\alpha}{\mu_\pi} \nabla (p_\alpha - W_\alpha \rho_\alpha g z) \qquad p_\alpha = p_\beta - p_{c,\alpha\beta}$$

Relative permeability depends on saturation -- introduces nonlinearity

Energy Conservation Equation

$$\frac{\partial}{\partial t}\left[ \phi \sum_\alpha s_\alpha \rho_\alpha U_\alpha + (1-\phi)\rho_r c_r T \right] + \nabla \cdot \left[ \sum_\alpha q_\alpha \rho_\alpha H_\alpha - \kappa \nabla T \right] = Q_e$$

Multicomponent Reactive Transport Equations

$$\frac{\partial}{\partial t}\left[ \phi \sum_\alpha s_\alpha \Psi_j^\alpha \right] + \nabla \cdot \left[ \sum_\alpha \Omega_\alpha \right] = -\sum_m v_{jm} I_m + Q_j$$

*Total Concentration*

$$\Psi_j^\alpha = \delta_{\alpha l} C_j^\alpha + \sum_i v_{ji} C_i^\alpha$$

Total Solute Flux

$$\Omega_j^\alpha = (-\tau \phi s_\alpha D_\alpha \nabla + q_\alpha)\Psi_j^\alpha$$

Nonlinear function of the concentrations of primary chemical components

Mineral Mass Transfer Equation

$$\frac{\partial \phi_m}{\partial t} = V_m I_m$$

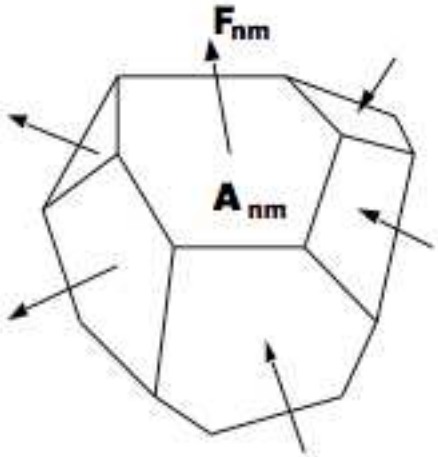
$$\phi + \sum_m \phi_m = 1$$

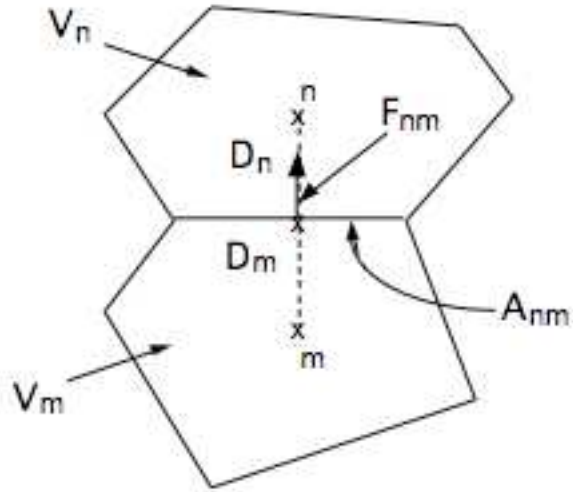
# Integrated finite-volume discretization

**Form of governing equation:**

$$\frac{\partial A}{\partial t} + \nabla \cdot F = S \qquad F = qpX - \phi D p \nabla X$$

Integrated finite-volume discretization




Discretized residual equation:

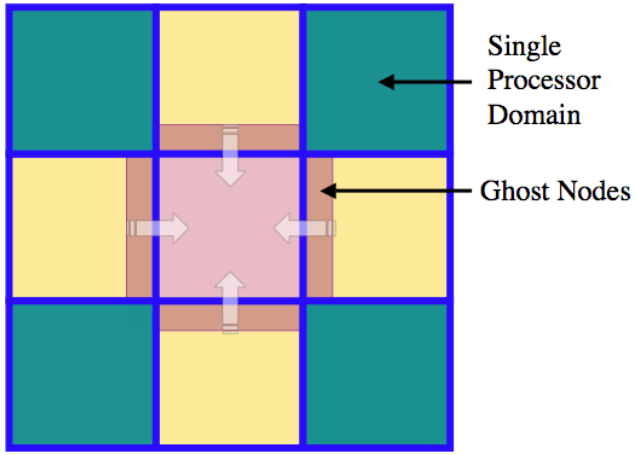
$$R_n = \left(A_n^{k+1} - A_n^k\right)\frac{V_n}{\Delta t} + \sum_{n'} F_{nn'} A_{nn'} - S_n V_n$$

$$F_{nn'} = (q\rho)_{nn'} X_{nn'} - (\phi D \rho)_{nn'} \frac{X_n - X_{n'}}{d_n + d_{n'}}$$

(Inexact) Newton iteration: $\sum_{n'} J^i_{nn'} \delta x^{i+1}_{n'} = -R^i_n \qquad J^i_{nn'} = \frac{\partial R^i_n}{\partial x^i_{n'}}$

# Domain-decomposition parallelism

- **PFLOTRAN parallelism comes from domain decomposition.**
- **Each processor responsible for nodes in one subdomain. (Plus associated vector, matrix entries)**



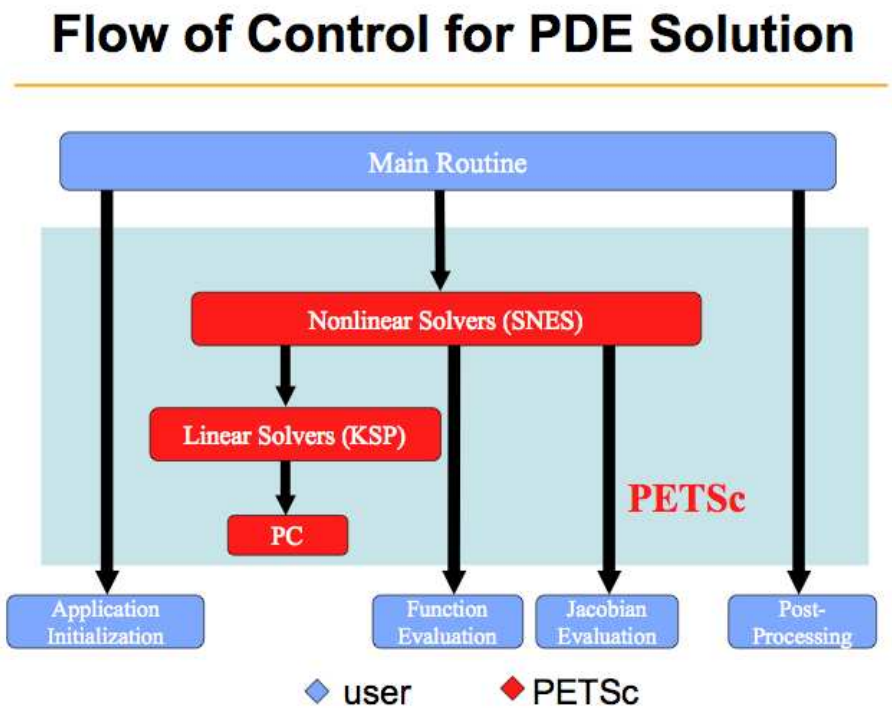

- **Accumulation terms are simple: Each processor calculates terms for the nodes it owns.**
- **Flux terms more complicated:**
  - **Must calculate fluxes across subdomain boundaries.**
  - **Scatter/gather of *ghost nodes* required (*halo exchanges*)**

# Outline

- Subsurface flow and reactive transport
- Governing eqns, discretization, solution
- Code architecture and PETSc framework
- Hanford 300 Area application and performance
- Ongoing research and associated challenges
- Summary and future directions

# PFLOTRAN architecture

- Built on top of PETSc, which provides
  - Object-oriented management of parallel data structures,
    - Create parallel objects (e.g., matrices and vectors) over a set of processors.
    - Methods (e.g., `MatMult()` )called on those objects handle parallel coordination.
  - Parallel solvers and preconditioners,
  - Efficient parallel construction of Jacobians and residuals
- We provide
  - Initialization, time-stepping, equations of state, post-processing
  - Functions to form residuals (and, optionally, Jacobians) on a local patch (PETSc routines help us with patch formation by setting up scatter/gather contexts)



# Building PFLOTRAN with PETSc

- PETSc has allowed us to develop a complex, scalable code in very little time:
  - Initial TRAN by Glenn Hammond for DOE CSGF practicum
  - Initial FLOW by Richard Mills for DOE CSGF practicum
  - Subsequent development of multiphase modules by Peter Lichtner and Chuan Lu during Lu's postdoc
  - Rapid improvements during SciDAC project
- PETSc is more than just a "solvers package"
- Provides a comprehensive framework:
  - Parallel mesh and associated linear algebra object management
  - Nonlinear solvers
  - Linear (iterative) solvers
  - Performance logging and debugging
  - Interfaces to many other packages



OAK RIDGE National Laboratory

# PETSc mesh management

- PETSc has greatly facilitated mesh management
- DM for general grids (experimental)
- DA for structured grids
- Provides support for
  - Scatter/gather of ghost points according to a stencil
  - Mapping between natural (application), global (PETSc), and local (ghosted) orderings
  - Hierarchies of grids via DMMG objects
    - Multilevel solve via DMMGSolve().

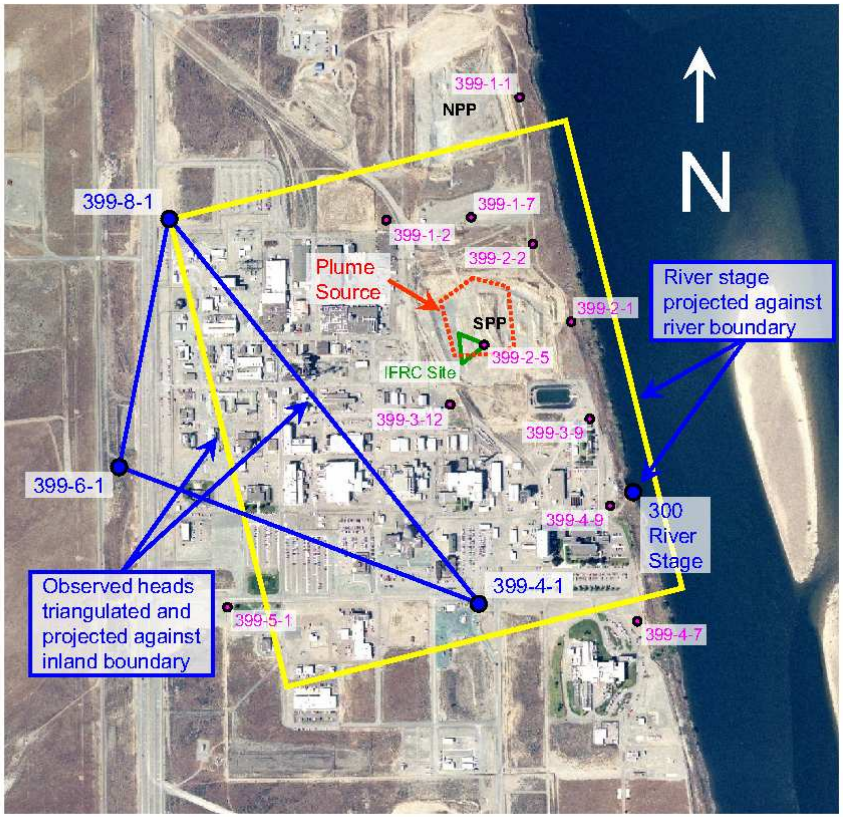
# PETSc nonliner solver framework

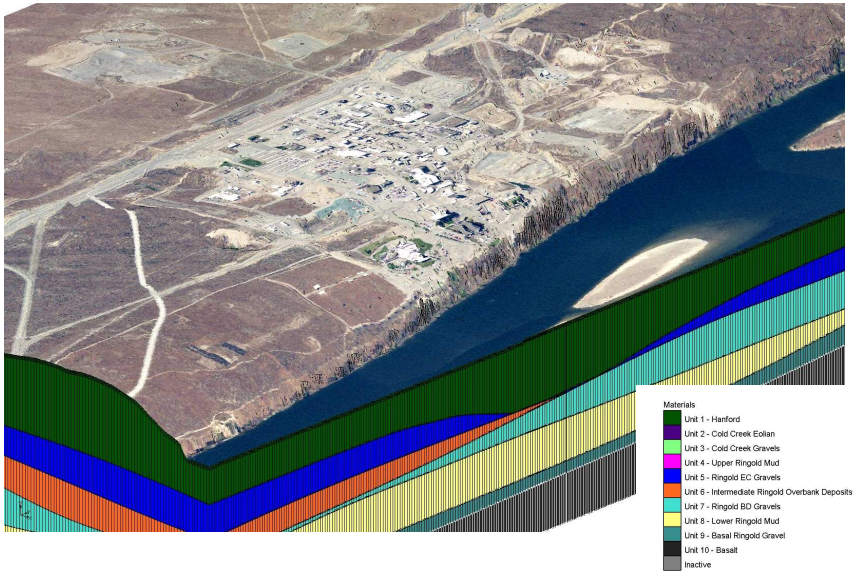
- Inexact Newton with various globalization strategies (line search, trust region)
- User provides SNES with
  - Residual: `PetscErrorCode (*func) (SNES snes, Vec x, Vec r, void *ctx)`
  - Jacobian: `PetscErrorCode (*func) (SNES snes, Vec x, Mat *J, Mat *M, MatStructure *flag, void *ctx)`
- Our functions expect a patch (local, contiguous region at single refinement level).
- Assembly of patch facilitated by PETSc (or SAMRAI in case of AMR)

# Outline

- Subsurface flow and reactive transport
- Governing eqns, discretization, solution
- Code architecture and PETSc framework
- Hanford 300 Area application and performance
- Ongoing research and associated challenges
- Summary and future directions

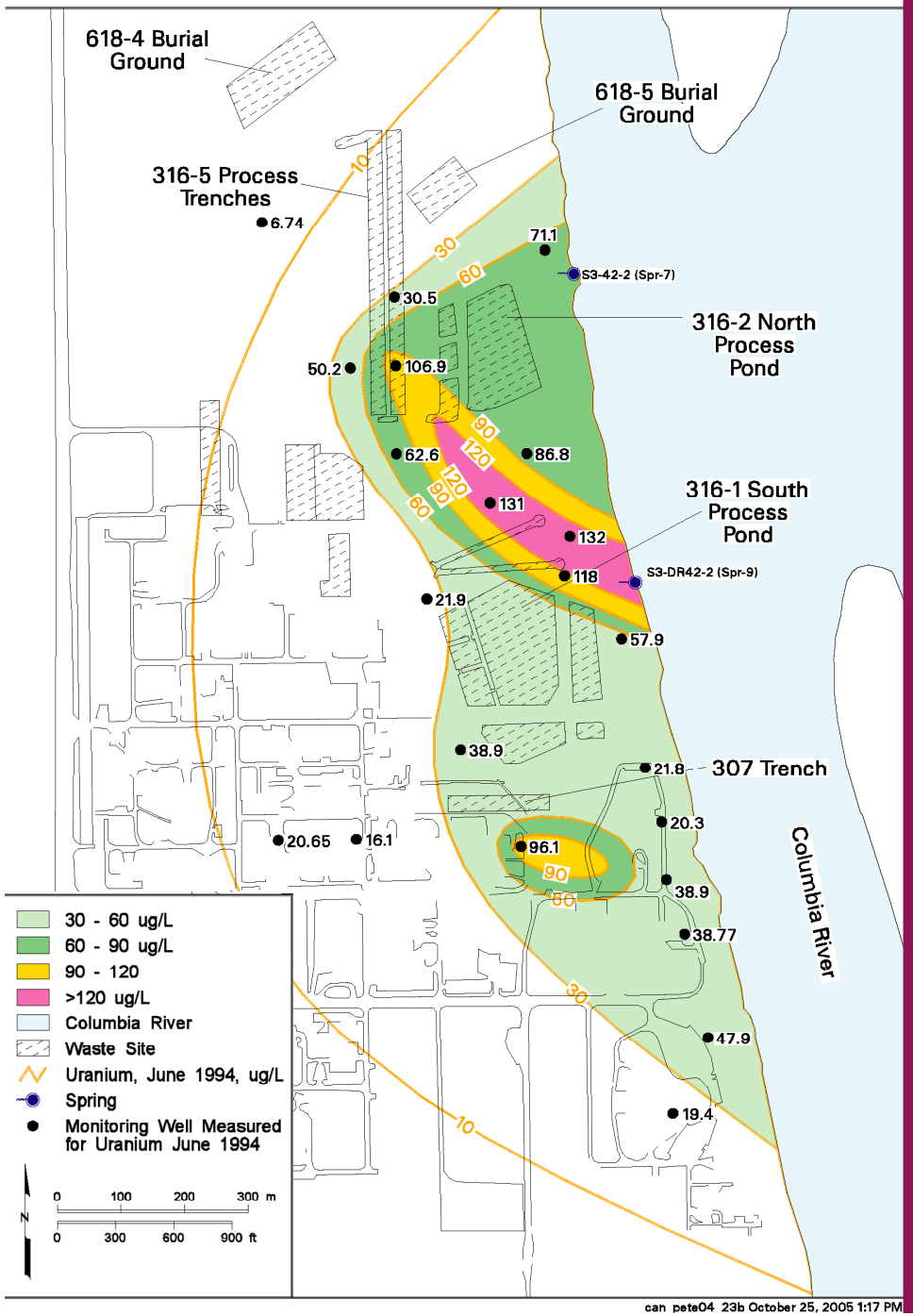
# Hanford 300 Area Layout



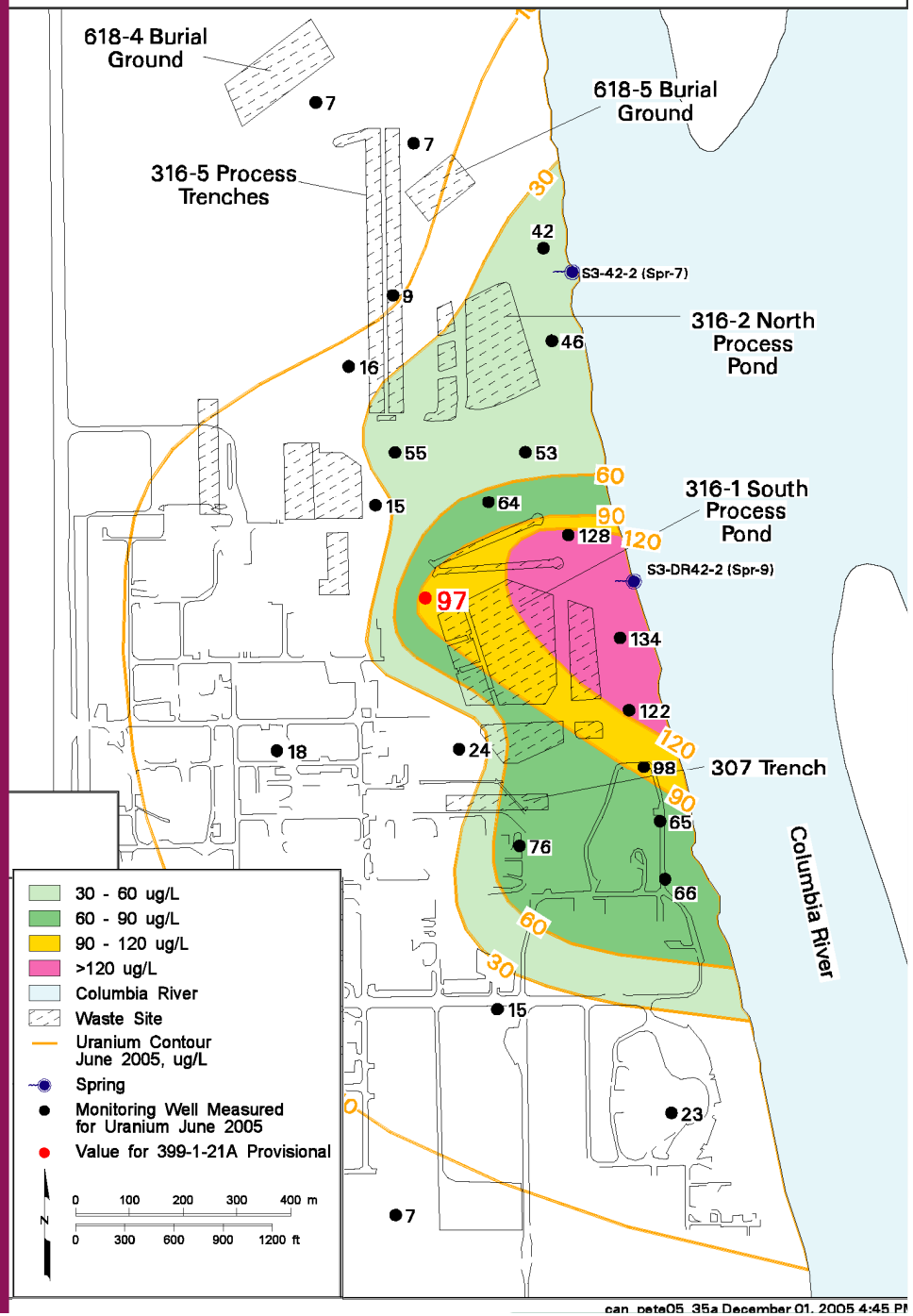

Domain size: 850 m x 1 km x 20 m

## Shaded 300 Area Uranium, June 1994




can pete04 23b October 25, 2005 1:17 PM

## Shaded 300 Area Uranium, June 2005




can pete05 35a December 01, 2005 4:45 PM

# Hanford Modeling Challenges

- 3D Domain: length and time scales
  - field scale domain (~km)
  - hourly river fluctuations, ~ 1000 year predictions
  - fast flow rates (5 km/y)

- Complex chemistry: Na-K-Ca-Fe-Mg-Br-N-$CO_2$-P-S-Cl-Si-U-Cu-$H_2O$ (~15 primary species)
- Multiscale processes (μm-m)
- Highly heterogeneous sediments
  - fine sand, silt; coarse gravels; cobbles
- Variably saturated environment
- Frequent and large river-stage fluctuations

# Jaguar: ORNL Cray XT5

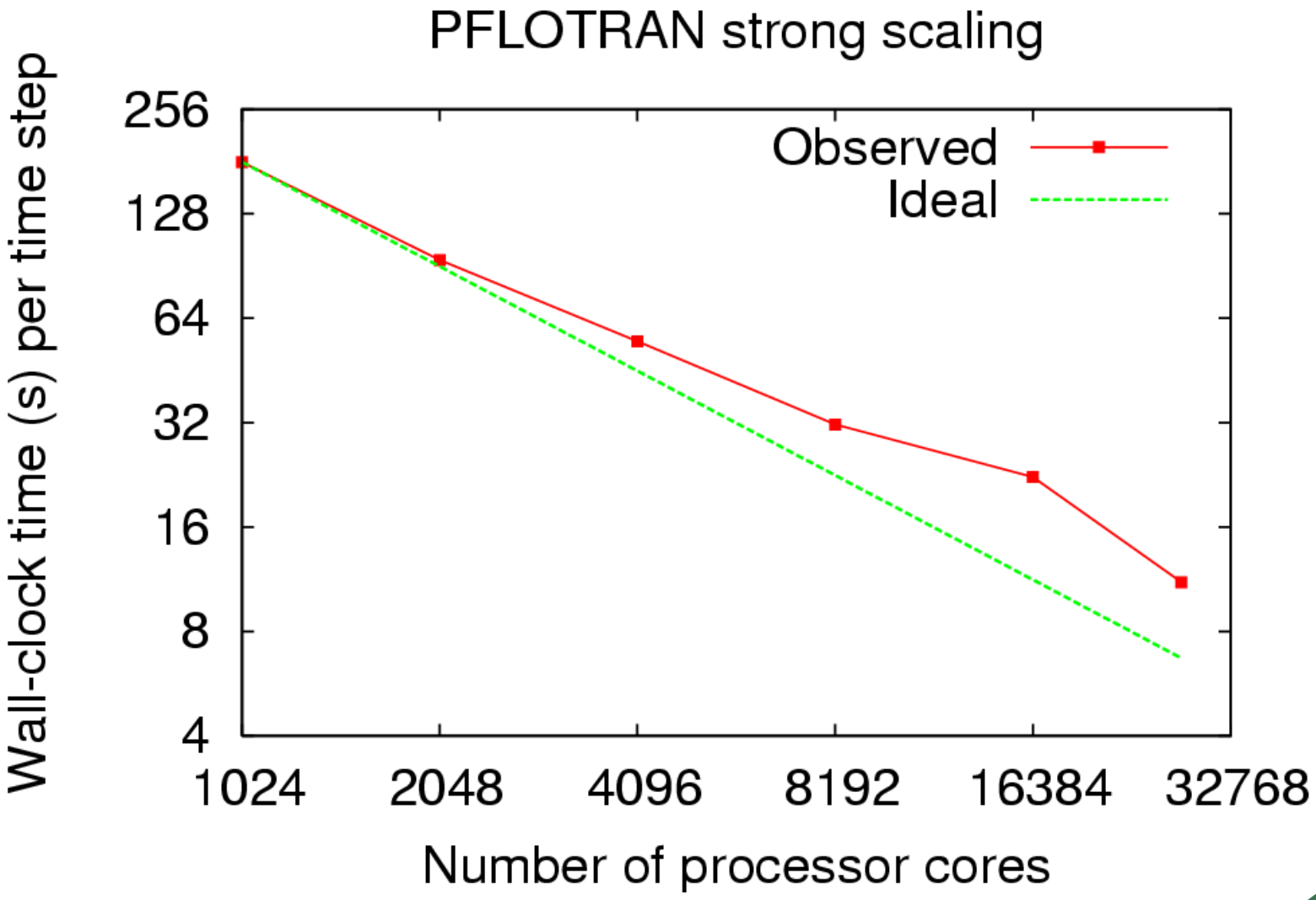
- 37538 quad-core 2.3 GHz Opteron compute nodes (150152 CPU compute cores)
  - Also additional nodes to handle OS services (I/O, etc.)
- 1.4 petaflops theoretical peak performance
- 300 Terabytes aggregate RAM; 10000 Terabytes parallel disk storage

# Solvers for Hanford 300 benchmark

- Inexact Newton method
  - Often With cubic line search for transient flow problems
- BiCGstab linear solver
- Preconditioned with block-Jacobi
- ILU(0) applied on each block
- Nothing fancy, but we have been surprised by how well this has worked!
- (Have also tried simple geometric multigrid; algebraic multigrid with Hypre, but have not yet gotten better performance out of them at high core counts.)

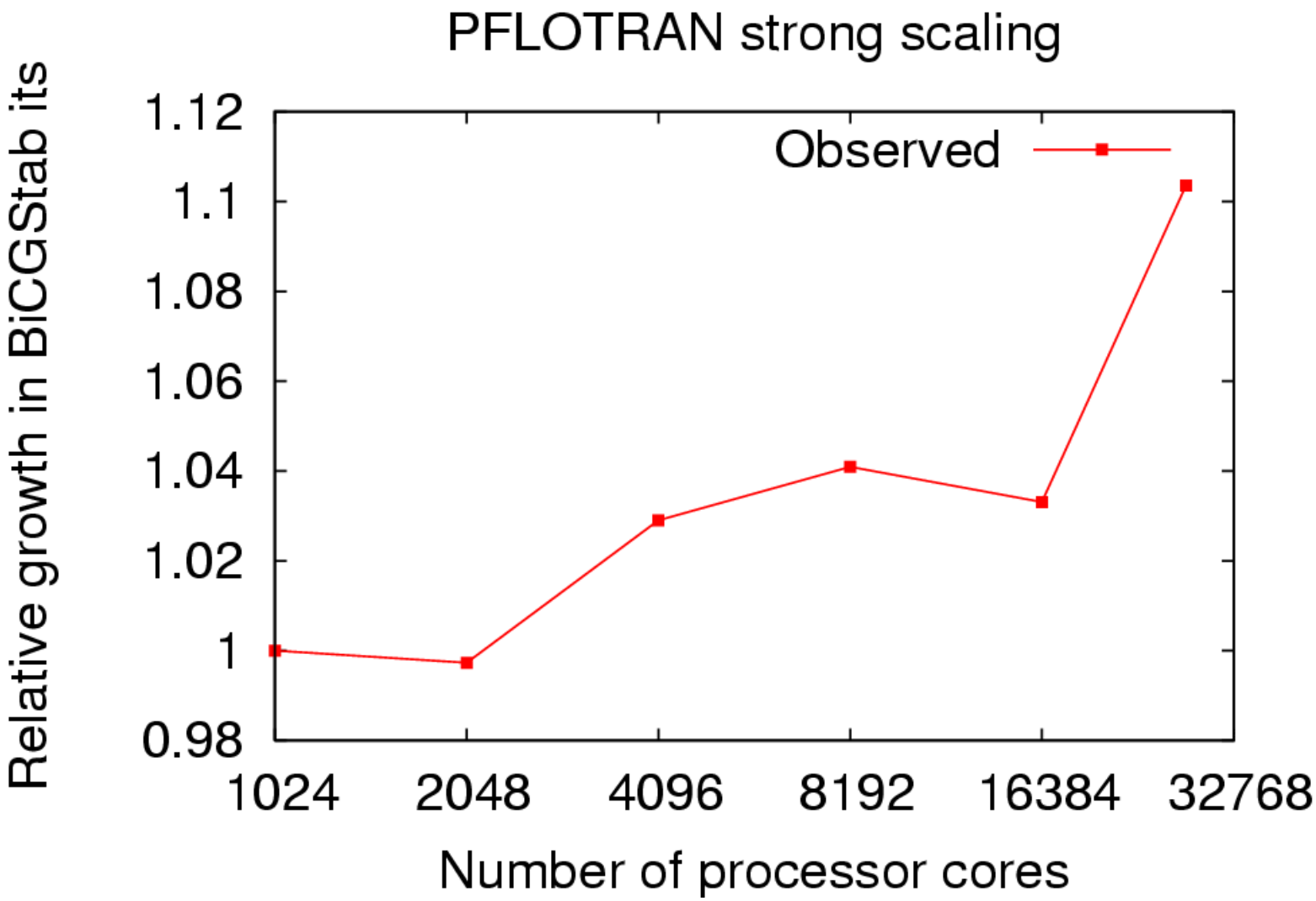
# Hanford 300 area: Strong scaling

- **270 million DoF (1350 x 2500 x 80 grid)**



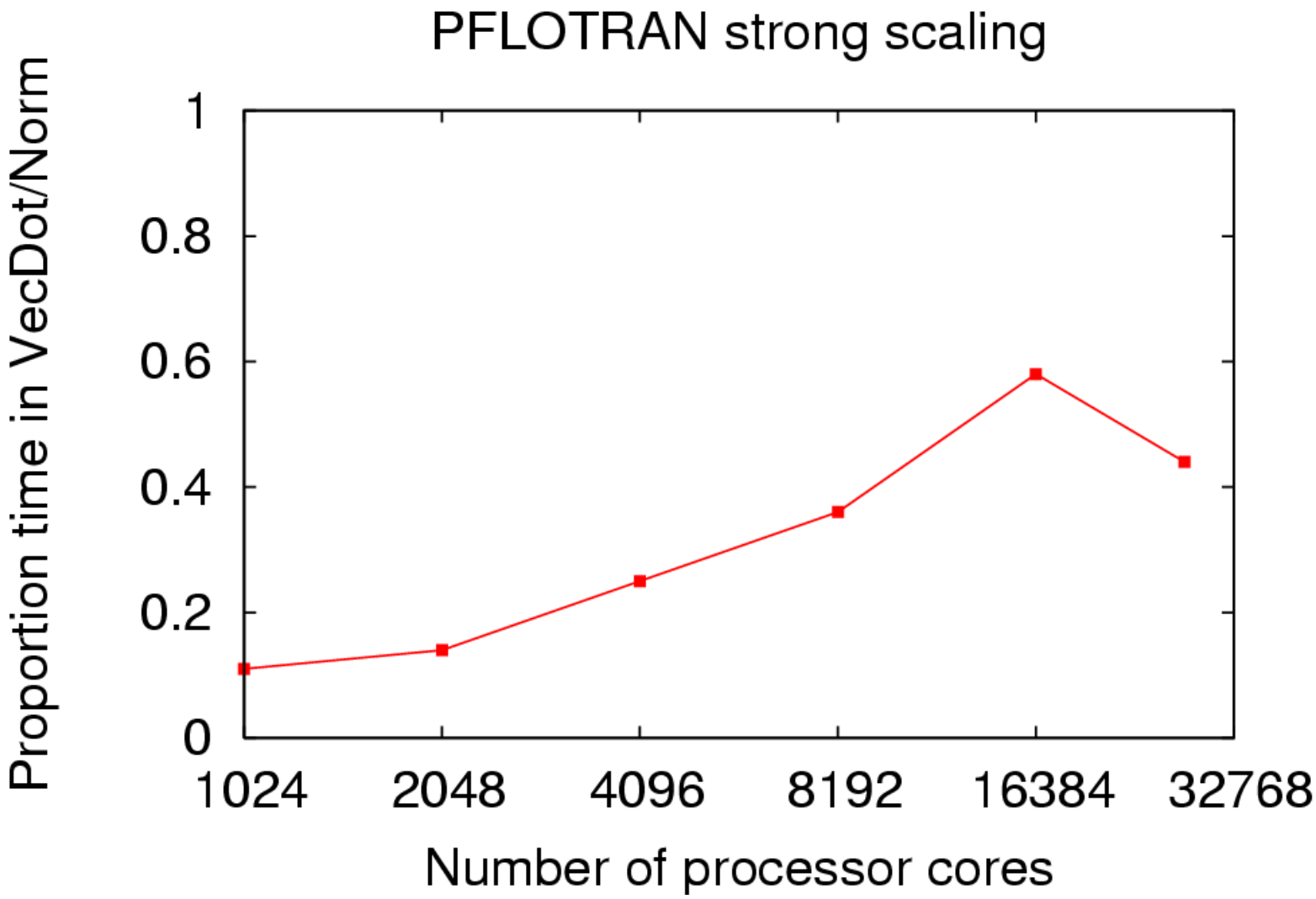
# 270 M DoF: Flow: BiCGstab its

- **270 million DoF (1350 x 2500 x 80 grid)**



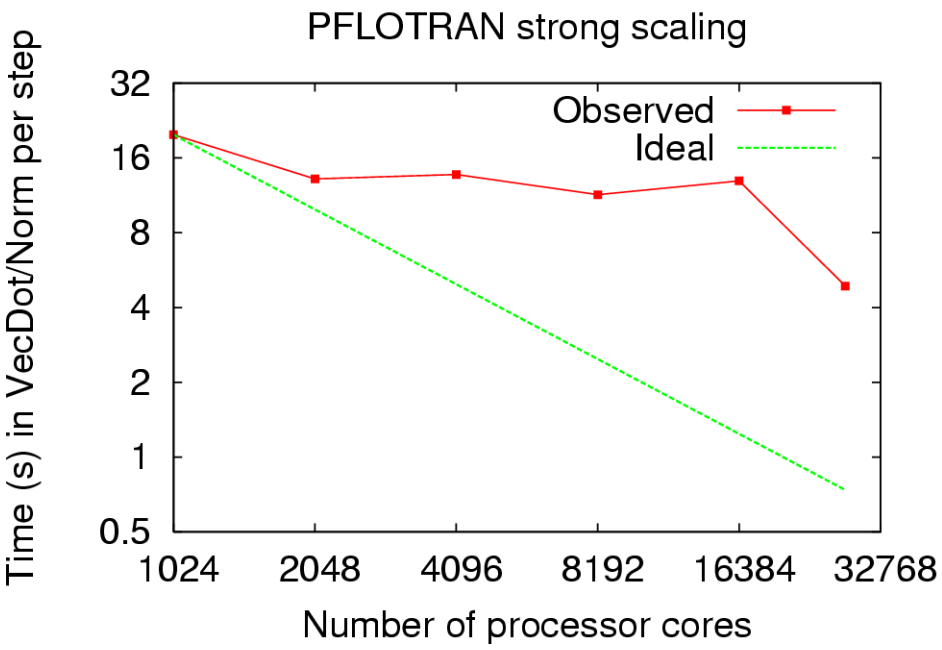
# Hanford 300 area: Strong scaling

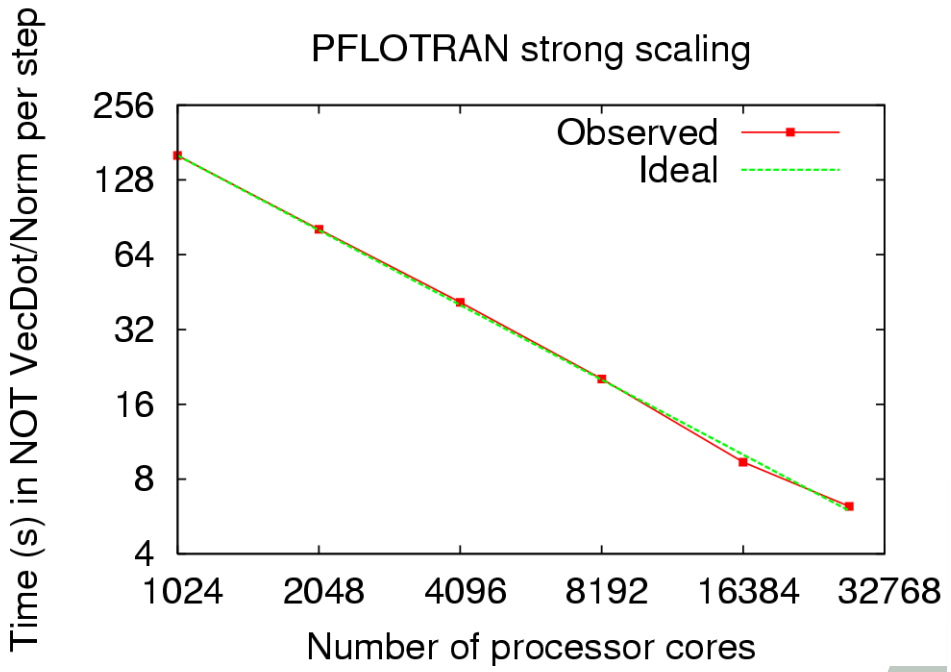
- **270 million DoF (1350 x 2500 x 80 grid)**



# Hanford 300 area: Strong scaling

- **270 million DoF (1350 x 2500 x 80 grid)**



# BiCGStab improvements

- **Cost of MPI_Allreduce() calls inside Krylov solver are big scalability barrier**
- **Original PETSc BiCGstab had 4 allreduces/iteration (including convergence check)**
- **Reworked PETSc BiCGstab has 3**
- **Also added "Improved" BiCGStab (IBCGS)**
  - **Considerably more complicated: requires transpose matrix vector product, extra vector operations**
  - **Only 2 MPI_Allreduce()'s per iteration required**
  - **By lagging residual norm calc., can wrap everything into one MPI_Allreduce(), at cost of doing one additional IBCGS iteration**

16384 core,
30 time step run:

| Group | BCGS | IBCGS |
|---|---|---|
| MPI_SYNC | 196 | 120 |
| MPI | 150 | 79 |
| User | 177 | 200 |

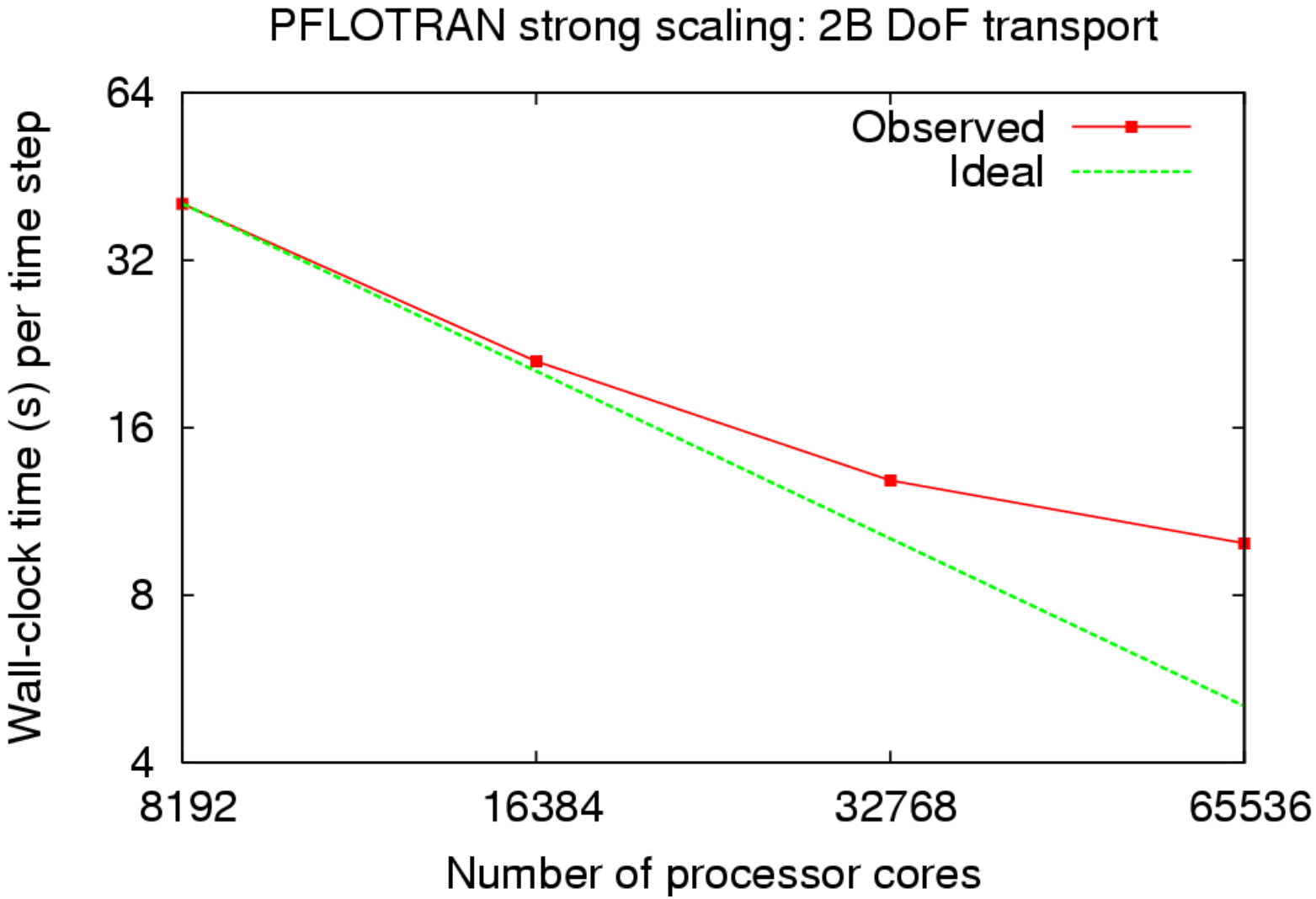
# Possible near-term improvements

| | Savings | Required | Time (focused) |
|---|---|---|---|
| Reorganize MatSolve | 3-4% | Coding | 20-30 hours |
| BiCGStab w/ 1 reduction DONE | 20% | Coding | 10 hours |
| Eisenstat-Walker | 20-30% | Thinking and experimentation | 24 hours |
| Geometric multigrid | 0-50% | Coding, experimentation, thinking | 30 (?) hours |
| GOOD preconditioner | 0-70% | Brilliance | ??? |

Maybe improve exact same run by 35% in next six months.

# 2B DoF: Flow + Transport: Transport

- **2 billion DoF (850 x 1000 x 160 grid, 15 species)**
  (Limit of 32-bit indices!)

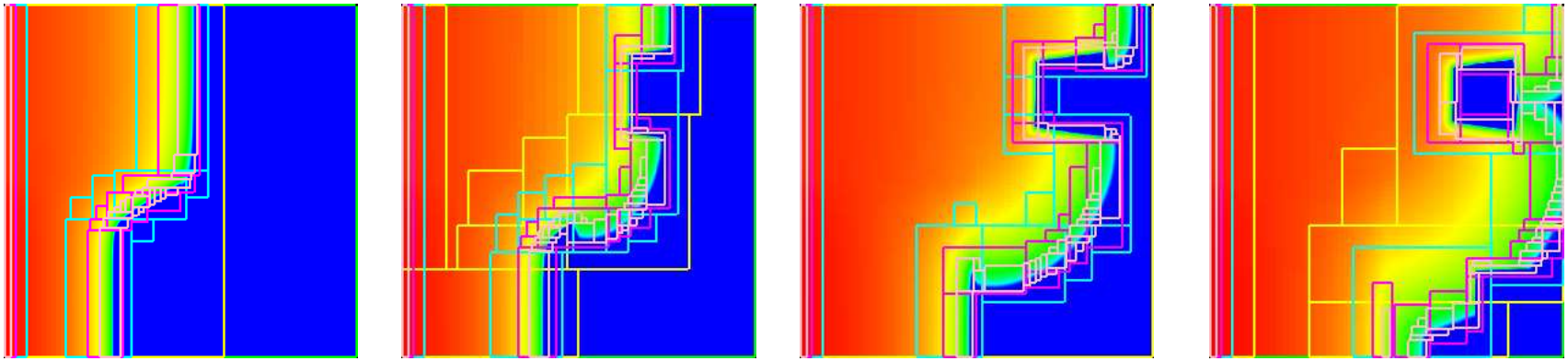


# What about I/O?

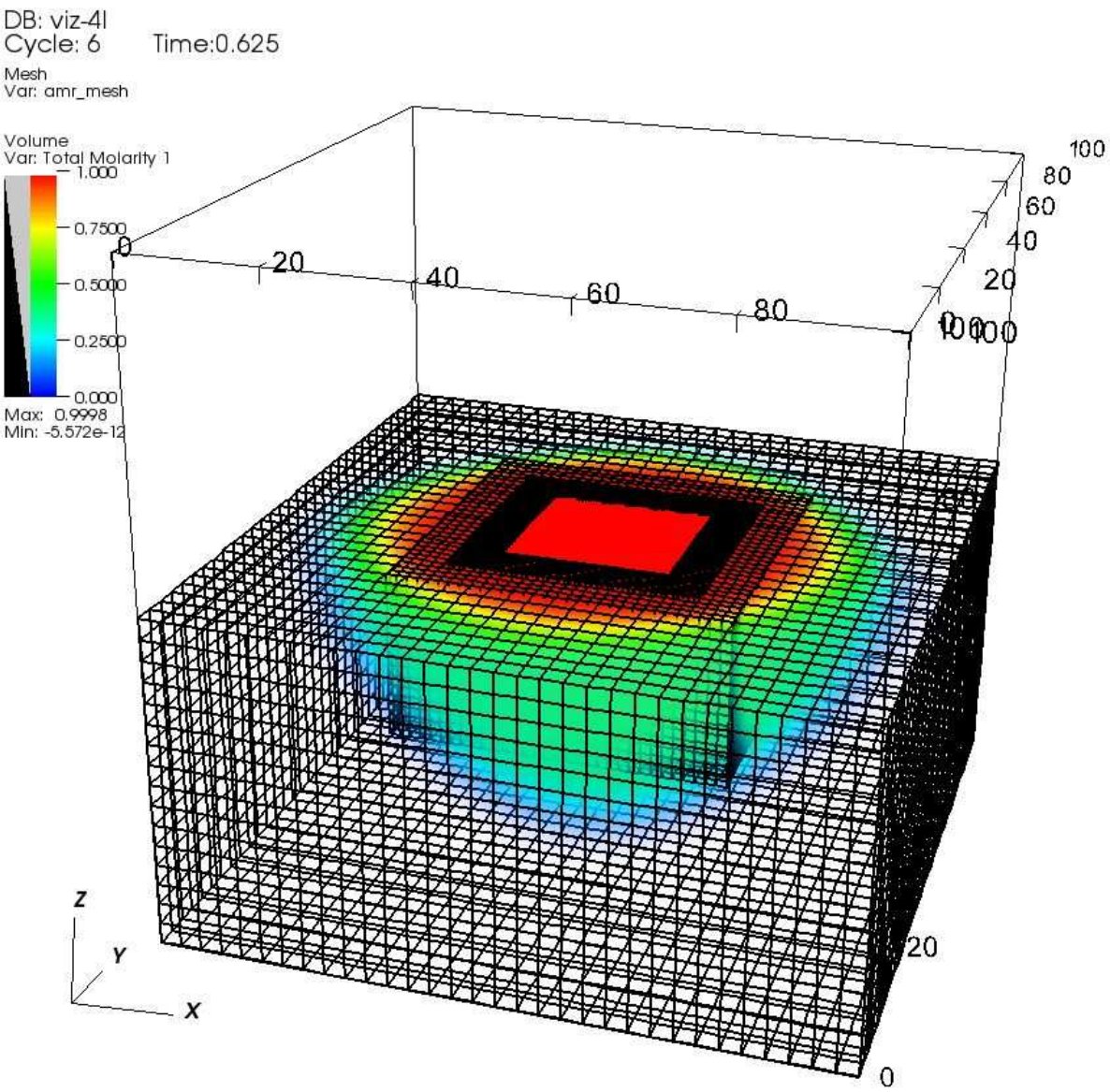
- Above 8000K cores, serial I/O becomes impractical
- Added parallel IO:
  - MPI-IO backend to PETSc Viewers (for checkpoints)
  - Parallel HDF5 input/output (for input files, simulation output files)
- This fixed lots of problems, but initialization was still a big problem above 16384 cores
  - Lots of reads from single, large HDF5 file
    - Every process parses entire file
    - Each process read chunk of data set starting from its offset
  - Problem: Only 1 meta-data server! All the file/open closes hammer it!
  - Solution: only subset of procs read, then broadcast to others

# Adaptive mesh refinement (AMR)

- **Incorporating AMR via the SAMRAI package from LLNL.**
- **AMR introduces local fine resolution only in regions where needed: Significant reduction in costs for simulations w/ localized fine features.**
- **AMR provides front tracking capability in the primary grid that can range from centimeter to tens of meters.**
- **Sub-grid scale models can be introduced in regions of significant activity and not at every node within the 3D domain.**

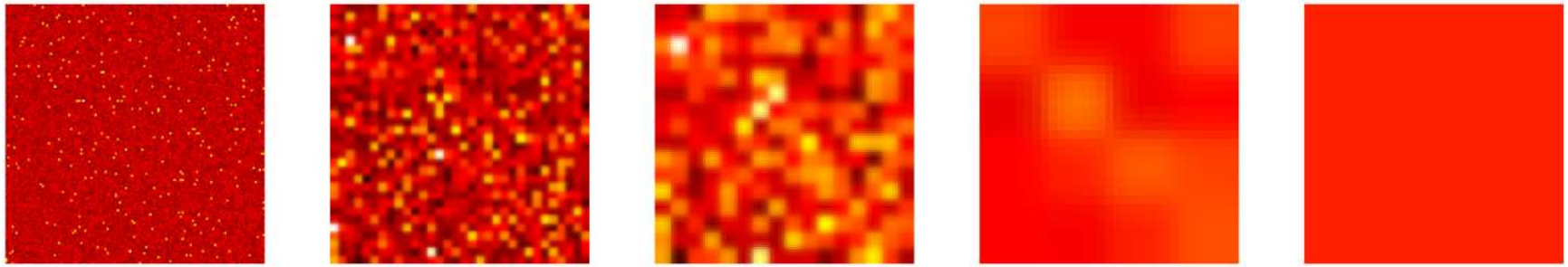


# Adaptive mesh refinement (AMR)



# Upscaling

- **Governing equations depend on averages of highly variable properties (e.g., permeability) averaged over a sampling window (REV).**
- **Upscaling and AMR go hand-in-hand: as the grid is refined/coarsened, material properties such as permeability must be calculated at the new scale in a self-consistent manner.**



Above: A fine-scale realization (128 x 128) of a random permeability field,

$$\kappa(x,y) = \zeta^{-\ln(\alpha)}, \ \zeta \text{ uniformly distributed in } (0,1), \ \alpha = 5$$

followed by successively upscaled fields ($N$ x $N$, $N$ = 32, 16, 4, 1) obtained with Multigrid Homogenization (Moulton et al., 1998)

# Outline

- Subsurface flow and reactive transport
- Governing eqns, discretization, solution
- Code architecture and PETSc framework
- Hanford 300 Area application and performance
- Ongoing research and associated challenges
- Summary and future directions

# Where now?

Using PETSc, we have developed a code, PFLOTRAN, that is now

- Highly modular, object-oriented, extensible
- Can scale all the way from laptop computers to the largest scale computers in the world
  - Not nearly as efficient as we might like…
  - …but capable of solving leadership-class simulation problems NOW

Current focus is now on

- Unstructured grid implementation.
  Critical for several problems we want to run on Jaguar.
- Multiple-continuum support
- Making structured adaptive mesh refinement really work
- Developing better solvers/preconditioners
  - Multilevel (geometric and algebraic approaches)
  - Physics-based
  - For flow solver, able to deal with phase transitions

Also need to think about

- Exotic and emerging architectures (e.g., GPGPUs, Cell, many cores)
- How to visualize simulations with billions of DoFs?

# Further simulation studies

- Hanford 300 Area
  - Geochemical transport of uranium in variably saturated soil
  - ~ 500M-1B grid cells (.5 x .5 x .5 m resolution)
  - 10-20 chemical degrees of freedom
  - ~ 10 billion total degrees of freedom
- CO2 sequestration at SACROC unit
- CO2 sequestration in Illinois basin
- Radionuclide migration at Nevada Test Site
- Contaminant migration at Oak Ridge sites
- Other applications?