DEPARTMENT OF CHEMICAL ENGINEERING
COLLEGE OF ENGINEERING | THE UNIVERSITY OF UTAH

Institute for
CLEAN AND SECURE ENERGY
THE UNIVERSITY OF UTAH

A DAG Approach to Tame Complexity in Multiphysics Software on Heterogeneous Architectures

JAMES C. SUTHERLAND

Associate Professor - Chemical Engineering

ABHISHEK BAGUSETTY

M.S. Student

(currently Ph.D. student at U. Pittsburgh)

DOE Awards
DE-NA0002375
DE-NA-000740

National Science Foundation

PetaApps award 0904631
XPS award1337145

Hierarchical Parallelization

- Uintah framework
  - *Domain decomposition data parallelism*
  - *Task parallelism & coarse-grained DAG*
  - *Scales to largest capability machines.*
  - *Minisymposterium on Uintah today!*
- "Expression Library"
  - *"fine-grained" task graph for PDE assembly.*
  - *tasks consist of stencil & field operations*
- Nebo EDSL
  - *GPU & multithreaded execution*
  - *Matlab-style syntax*
- Utilize resources at "high" level & "push down" as parallelism runs out

Distributed task-graph:

- MPI communication
- Coarse-grained, threaded task scheduling

On-node task-graph

- memory management & fine-grained task scheduling.
- thread-pools
- GPU management

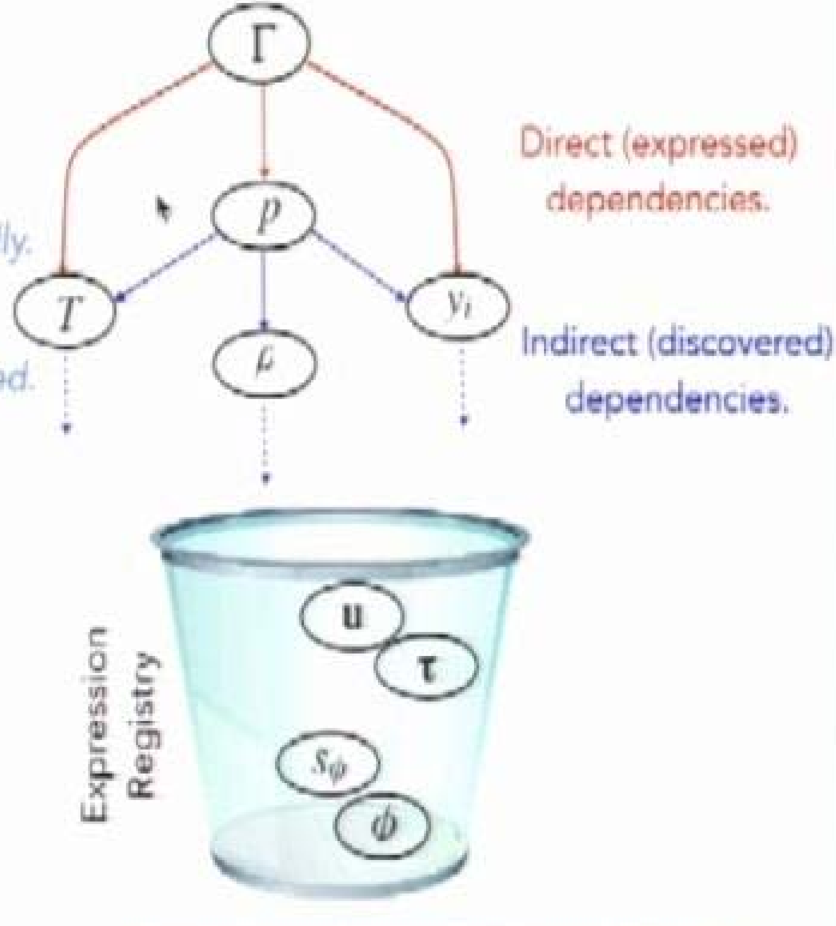
EDSL

- array & stencil operations
- GPU & multithread execution

A Simple Example of DAGs

- Register all expressions
  - *Each "expression" calculates one or more field quantities.*
  - *Each expression advertises its direct dependencies.*
- Set a "root" expression; construct a graph
  - *All dependencies are discovered/resolved automatically.*
  - *Highly localized influence of changes in models.*
  - *Not all expressions in the registry may be relevant/used.*
- From the graph:
  - *Deduce storage requirements & allocate memory (externally to each expression).*
  - *Automatically schedule evaluation, ensuring proper ordering.*
    - Asynchronous execution is critical! (overlap communication & computation)
  - *Robust scheduling algorithms are key.*

$$\Gamma = \Gamma(T, p, y_i)$$

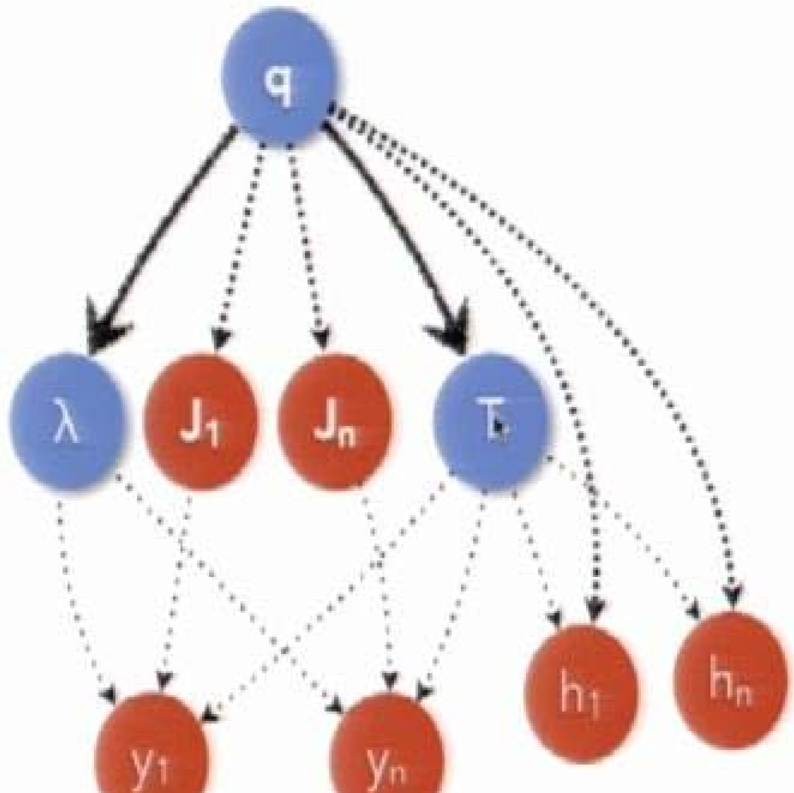


*Notz, Pawlowski, & Sutherland (2012). ACM Transactions on Mathematical Software, 39(1)

Changes in model form are naturally handled

Multi-species mixture heat flux:

$$\mathbf{q} = -\lambda \nabla T + \sum_{i=1}^{n} h_i \mathbf{J}_i$$



DSL: "Matlab for PDEs on Supercomputers"

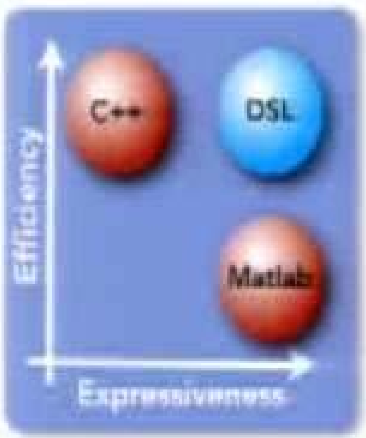
Field & stencil operations:

(and much more)

$$\text{rhs} = -\frac{\partial}{\partial x}(J_x + \mathcal{C}_x) - \frac{\partial}{\partial y}(J_y + \mathcal{C}_y) - \frac{\partial}{\partial z}(J_z + \mathcal{C}_z)$$

```
rhs <<= -divOpX( xConvFlux + xDiffFlux )
        -divOpY( yConvFlux + yDiffFlux )
        -divOpZ( zConvFlux + zDiffFlux );
```

Can "chain" stencil operations.



- 70+ natively supported discrete operators, easily define new ones
- Masks - allow operations on field subsets
- CPU (serial, multicore), GPU, Xeon Phi backends
- conditional operations (vectorized 'if')
- Field can live in multiple locations (GPU, CPU) simultaneously.

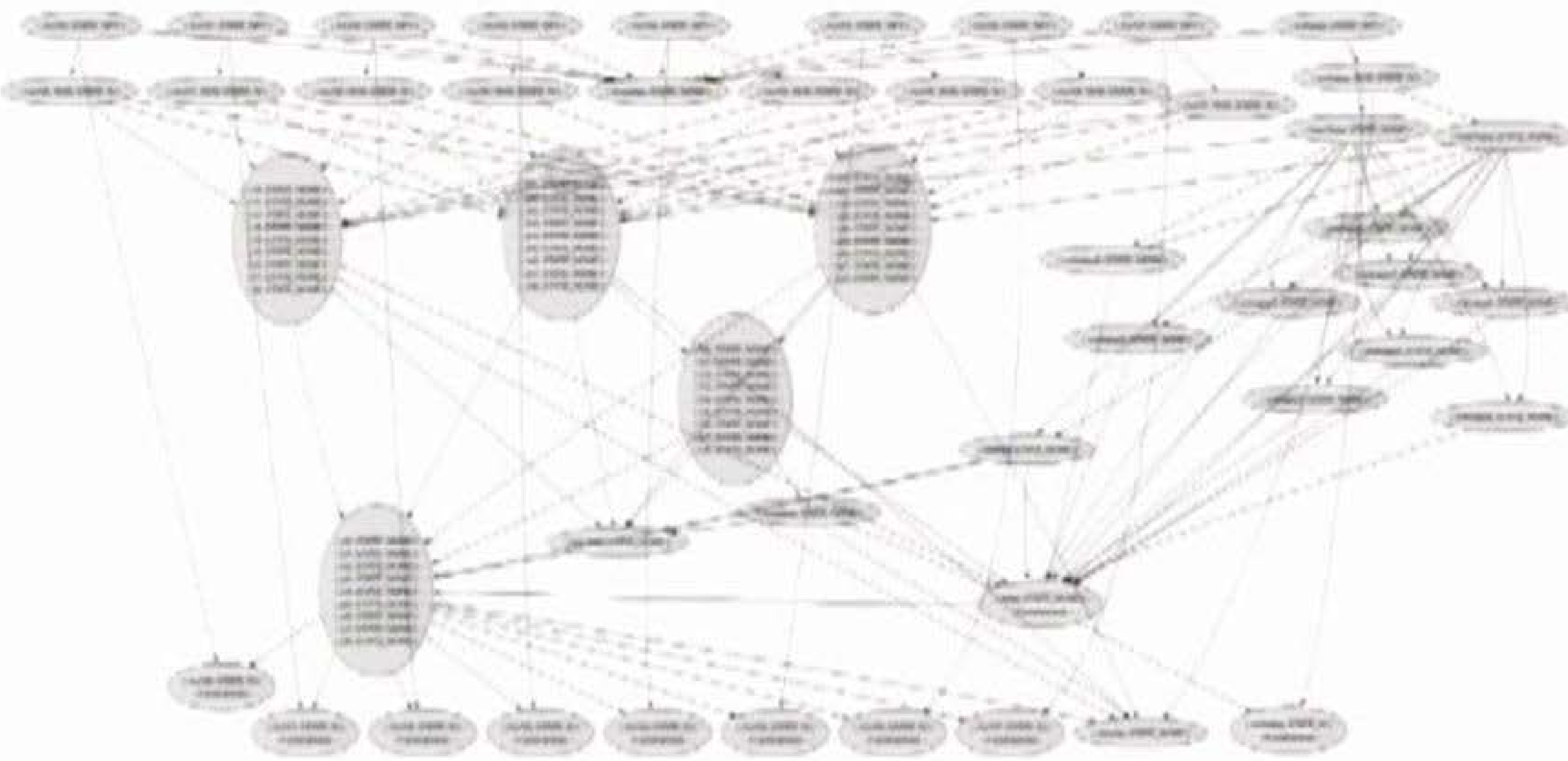
Auto-generate code for efficient execution on CPU, GPU, XeonPhi, etc. during compilation.

Real example: PoKiTT

(Portable Kinetics, Thermodynamics & Transport)

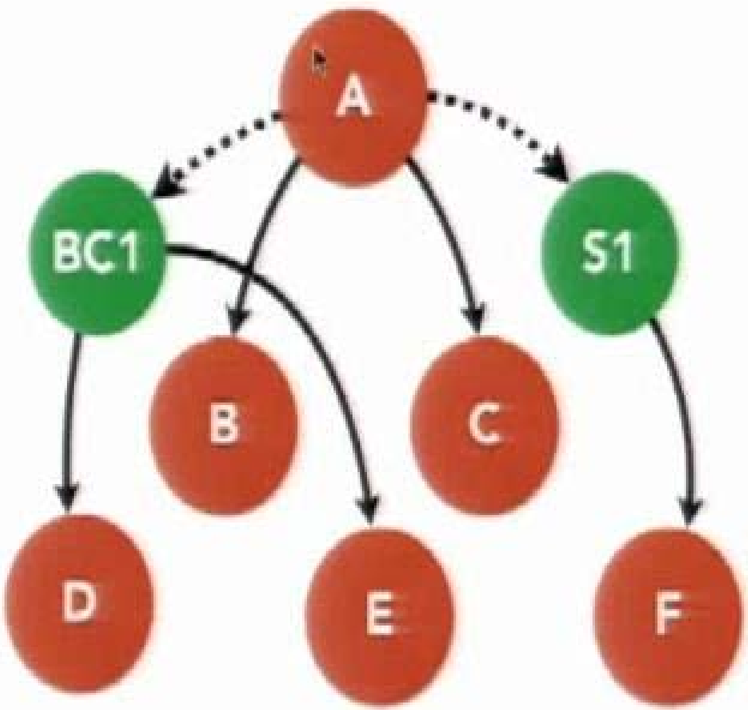
$$\rho \frac{\partial y_i}{\partial t} = -\nabla \cdot \mathbf{J}_i + s_i$$

$$\rho \frac{\partial h}{\partial t} = -\nabla \cdot \mathbf{q}_i$$



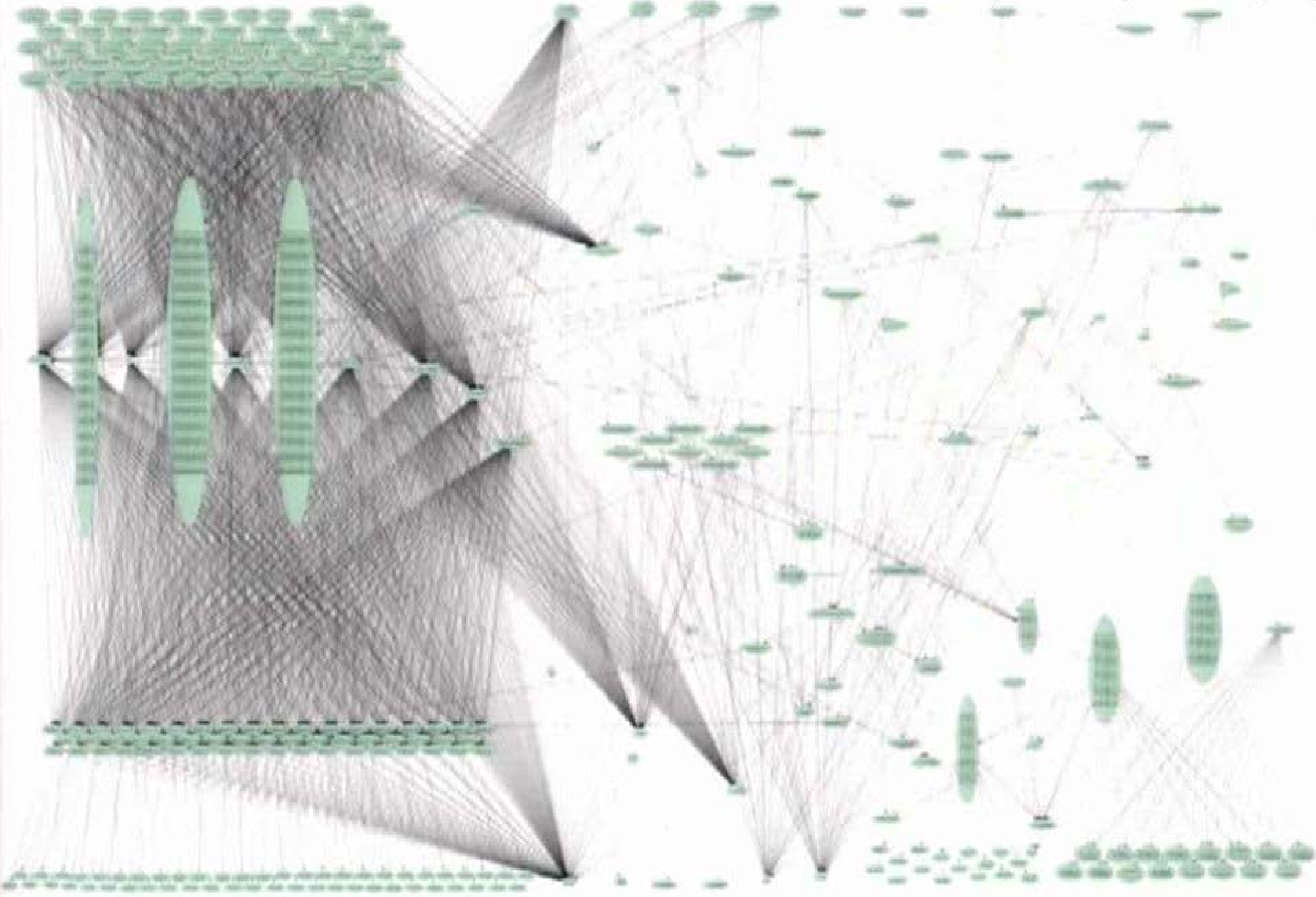
"Modifiers" - injecting new dependencies

- Motivation:
  - **Boundary conditions**: *modify a subset of the computed values.*
  - **Multiphase coupling**: *add source terms to RHS of equations.*
- Modifiers allow "push" rather than "pull" dependency addition.
- Modifiers are deployed after the node they are attached to, and are provided a handle to the field just computed.
- Modifiers can introduce new dependencies to the graph.



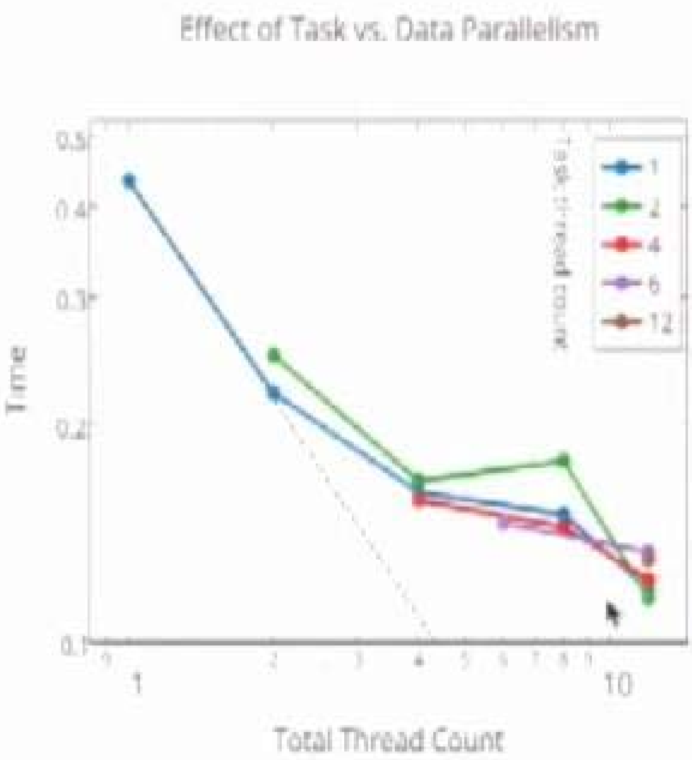
Coal combustion

- 55 PDEs
- ~35 ODEs per particle
- Complex interphase coupling

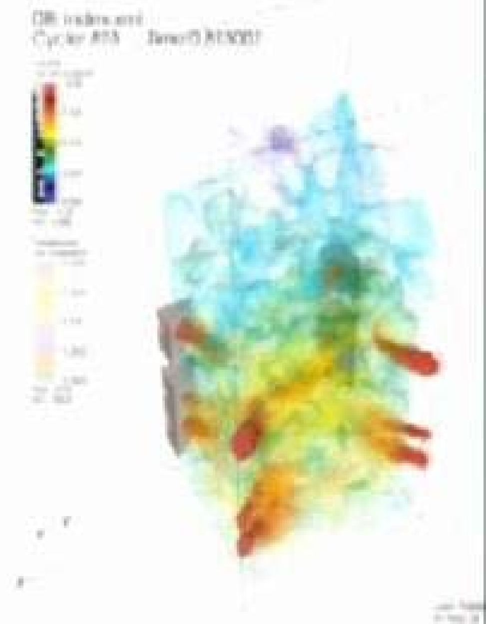


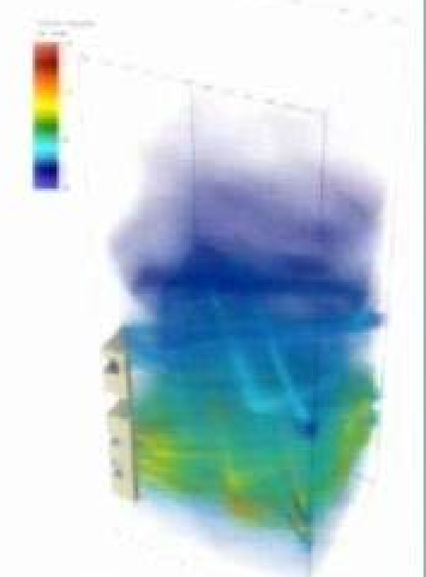
Example: Industrial Coil Boiler Simulation

- Many dozens of "modifiers"
  - *Boundary conditions*
  - *Multiphase couplings*
- "Wide" graph (lots of PDEs)



"Throttle" between task & data parallelism based on width of graph.





Conclusions

- DAG approaches can ***expose*** and ***exploit*** task & data parallelism.
  - *Automatically handle data movement (inter-node and intra-node)*
  - *Overlap communication & computation*
  - *Throttle between task & data parallelism as necessary.*

- More information & examples of applications:
  - *Minisymposterium on Uintah today.*
  - *Large-scale multi physics applications in Uintah/Wasatch*
    - **Wednesday 9:55 AM (Room 250 A)**
  - *Portable Kinetics, Thermodynamics & Transport (PoKiTT)*
    - **Wednesday 2:50 PM (Room 151G)**

DOE Awards
DE-NA0002375
DE-NA-000740

National Science Foundation
PetaApps award 0904631
XPS award1337145