Institute for
CLEAN AND SECURE ENERGY
THE UNIVERSITY OF UTAH

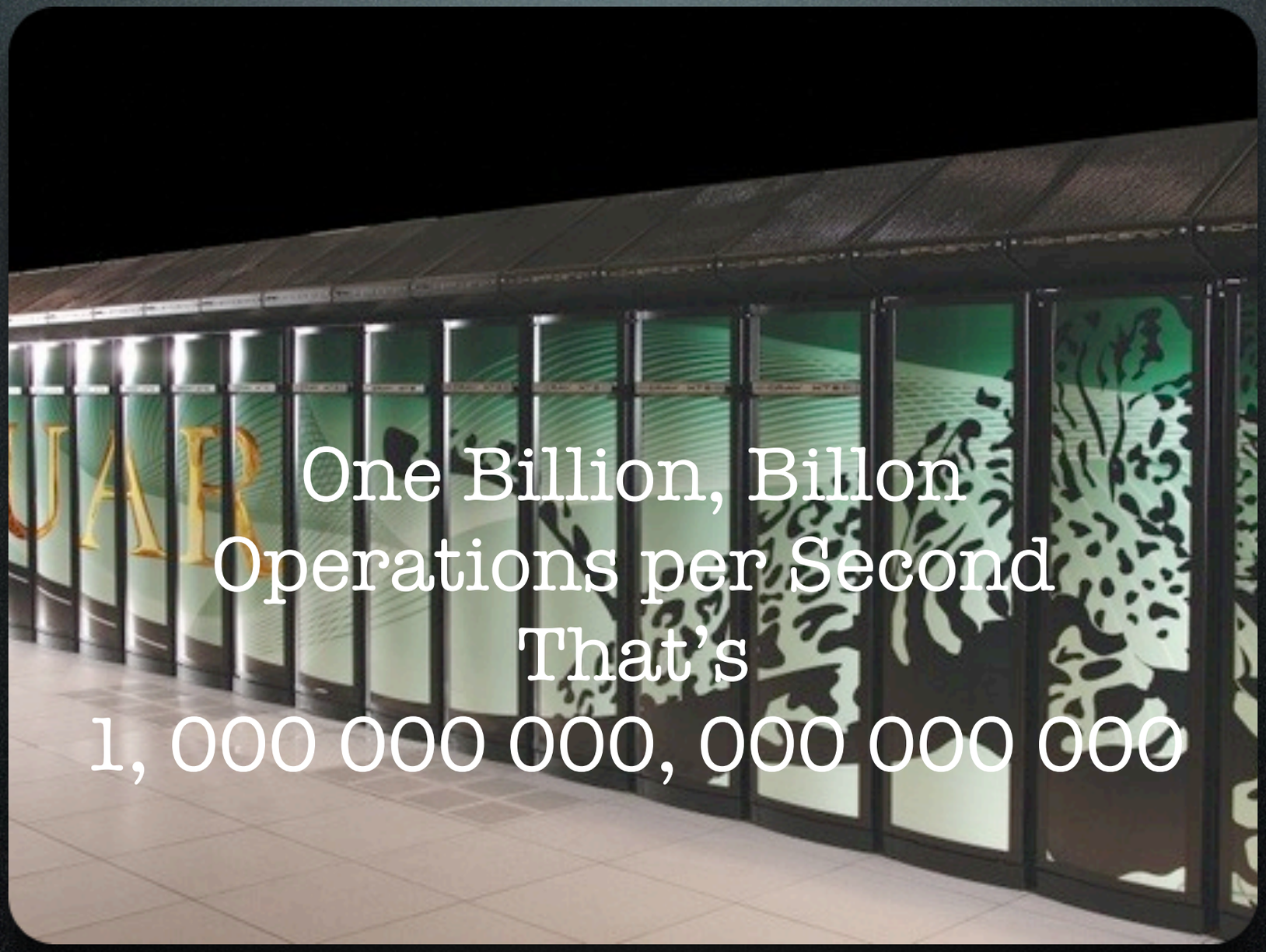
A Novel Computational Framework for Reactive Flow and Multiphysics Simulations

James C. Sutherland
and
Tony Saad

AICHE Annual Meeting
October 16 - 21, 2011
Minneapolis, MN, USA

This material is based upon work supported by the National Science Foundation under Grant No. PetaApps-0904631.

This research was sponsored by the National Nuclear Security Administration under the Accelerating Development of Retrofitable CO2 Capture Technologies through Predictivity program through DOE Cooperative Agreement DE-NA0000740


One Billion, Billon
Operations per Second
That's
1, 000 000 000, 000 000 000

Exascale

Challenges

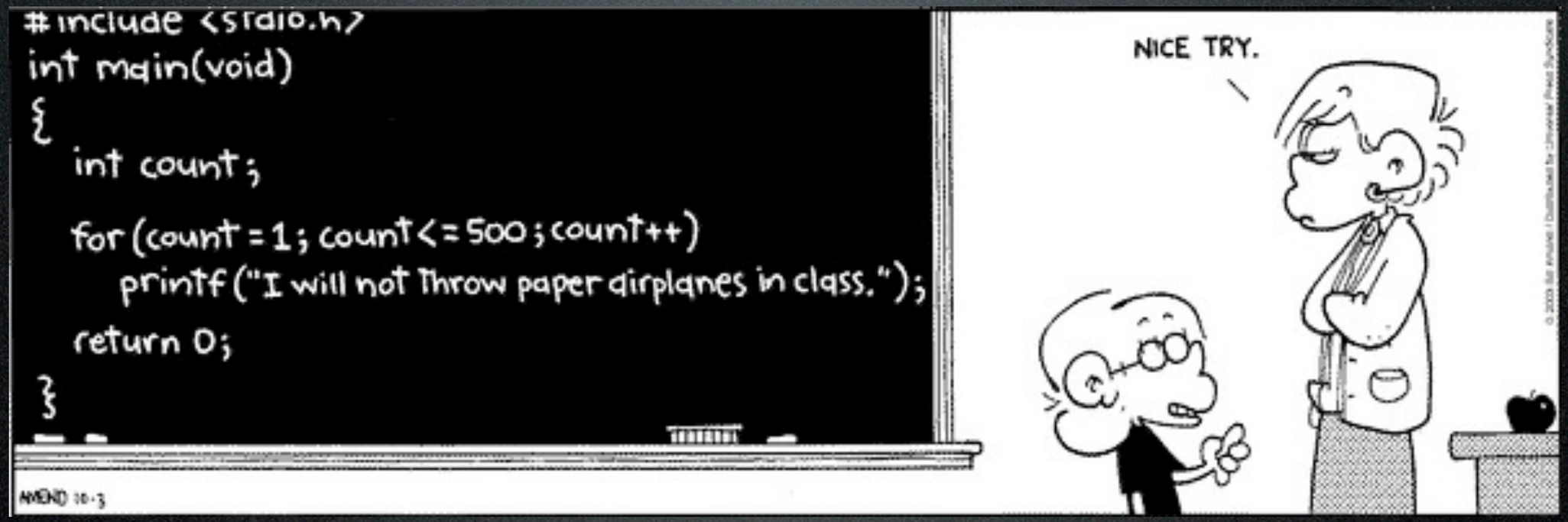
“It says it’s sick of doing things like inventories and paryrolls, and it wants to make some breakthroughs in astrophysics”

• Hardware

2 Giga-Watts of power!!!

Power is a dominant const

- Hardware
- Software



- Hardware
- Software
- Money

Software

formulate problem

difference equations

algorithm

IMPLEMENT

other discretization

What If ?

parallel framework

other models

courtesy of Philip J. Smith,
Institute for Clean & Secure Energy

Software Complexity

self assembly

mesh independent

cluster, thread, GPU

Imagine...

Data Dependencies!

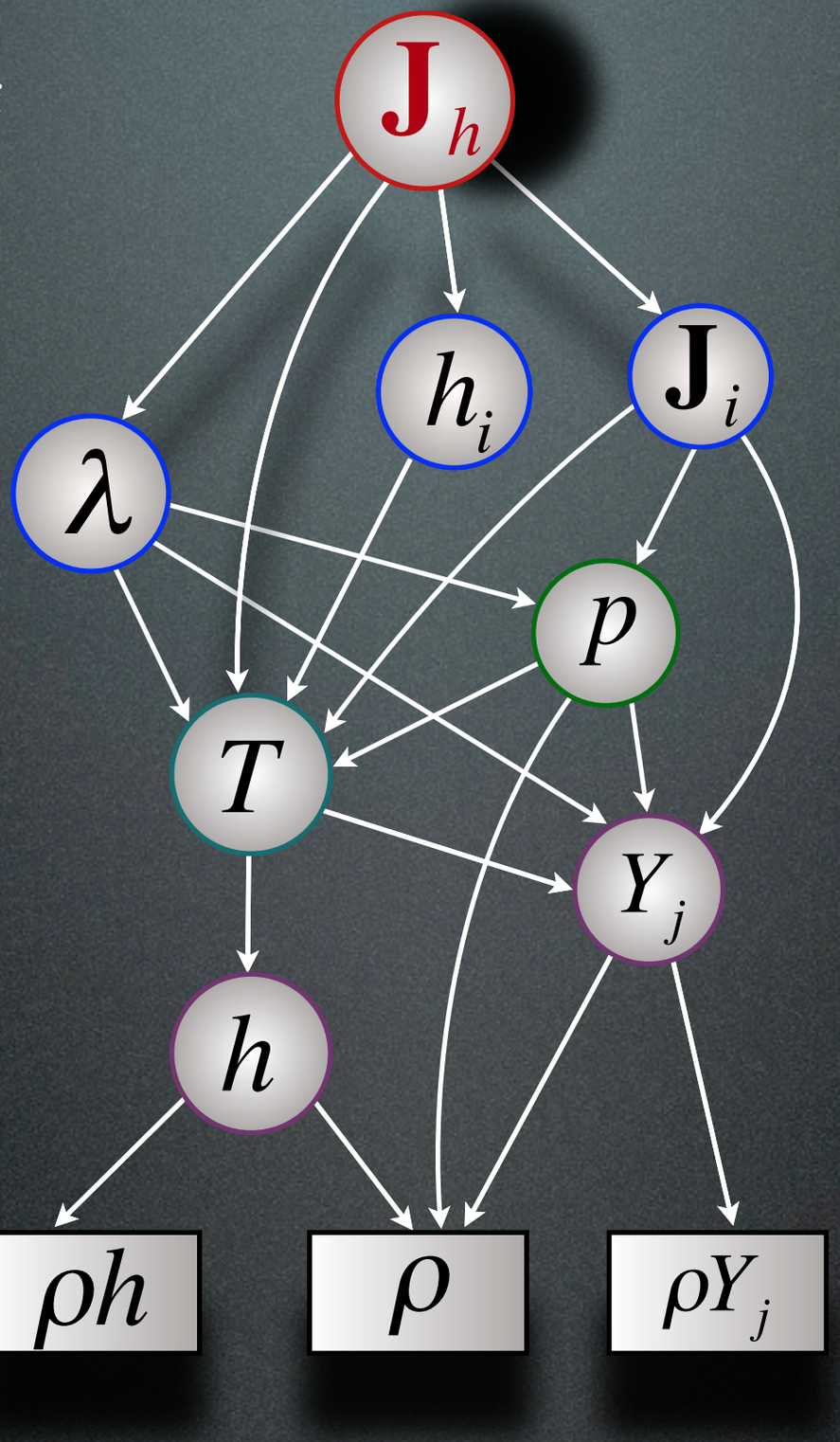
$$\mathbf{J}_h = -\lambda \nabla T + \sum_{i=1}^{n_s} h_i \mathbf{J}_i$$

MODEL A

$$\lambda = \lambda_0 = \text{const}$$

$$\mathbf{J}_i = -\sum_{j=1}^{n_s} D \nabla Y_j$$

$$h_i = h_i(T)$$



MODEL B

$$\lambda = \lambda\left(T, p, Y_i\right)$$

$$\mathbf{J}_i = -\sum_{j=1}^{n_s} D_{ij}\left(T, p, Y_k\right) \nabla Y_j - D_i^T\left(T, p, Y_k\right) \nabla T$$

$$h_i = h_i(T)$$

Notz, P. K., Pawlowski, R. P., and Sutherland, J. C., "Graph-based software design for managing complexity and enabling concurrency in multiphysics PDE software," Vol. IV, 2010.

Expression Concepts

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \mathbf{u}\phi = \nabla \cdot \Gamma \nabla \phi + S_\phi$$

- An Expression is a software representation of a mathematical expression

- An Expression computes fields it represents

- Each Expression indicates which expressions it depends on

In Practice...

- Register all expressions
- Determine root expression
- Construct tree
- Deduce storage requirements, and other metrics from graph

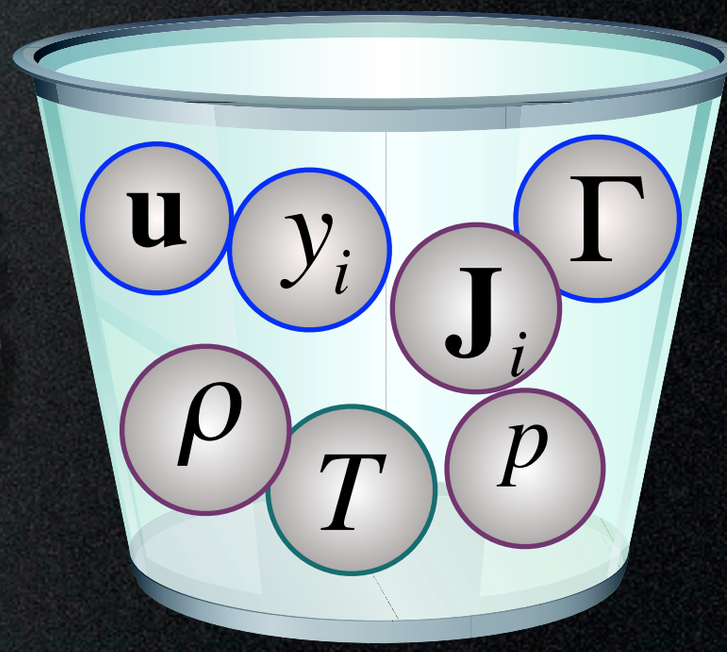
$$\frac{\partial \phi}{\partial t} + \nabla \cdot \mathbf{u}\phi = \nabla \cdot \Gamma \nabla \phi + S_\phi$$

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

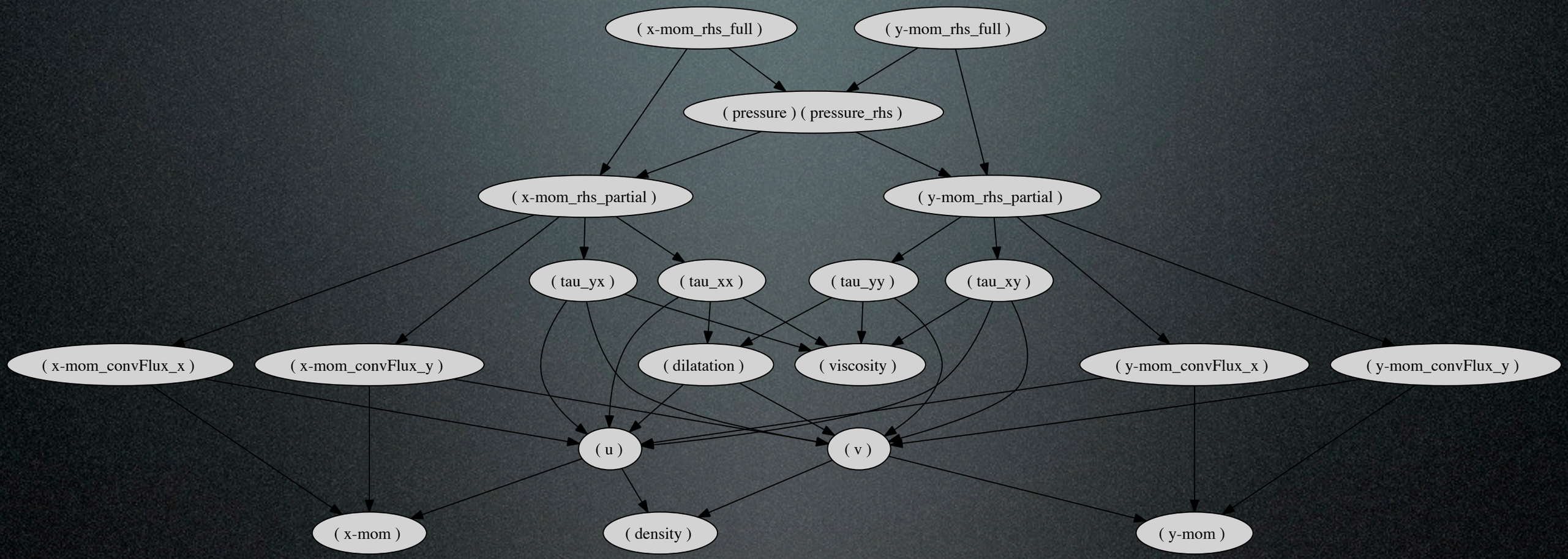
Direct dependencies.

Indirect (discovered) dependencies.

- Execute graph in reverse order: That's the algorithm!

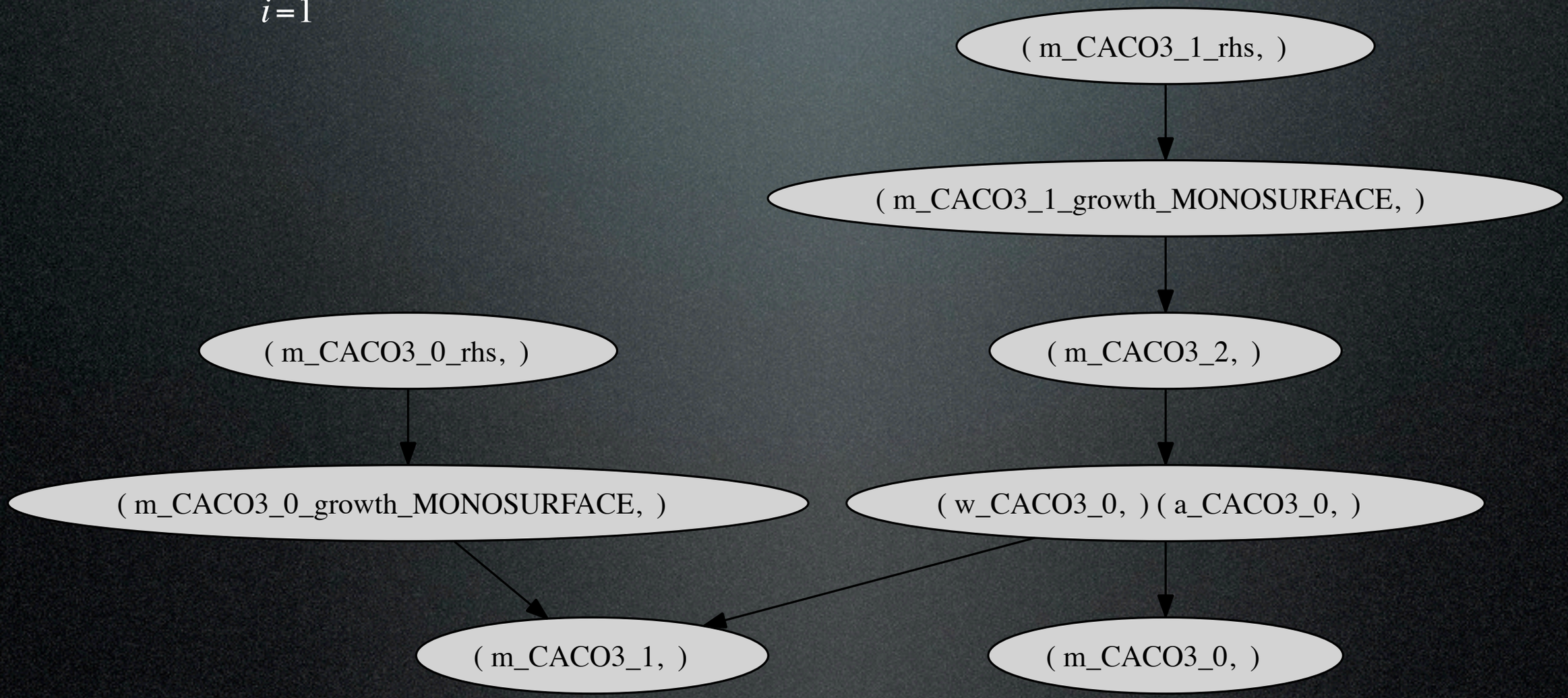


Example


(x-mom_rhs_full)
(y-mom_rhs_full)
(pressure) (pressure_rhs)
(x-mom_rhs_partial)
(y-mom_rhs_partial)
(tau_yx)
(tau_xx)
(tau_yy)
(tau_xy)
(x-mom_convFlux_x)
(x-mom_convFlux_y)
(dilatation)
(viscosity)
(y-mom_convFlux_x)
(y-mom_convFlux_y)
(u)
(v)
(x-mom)
(density)
(y-mom)

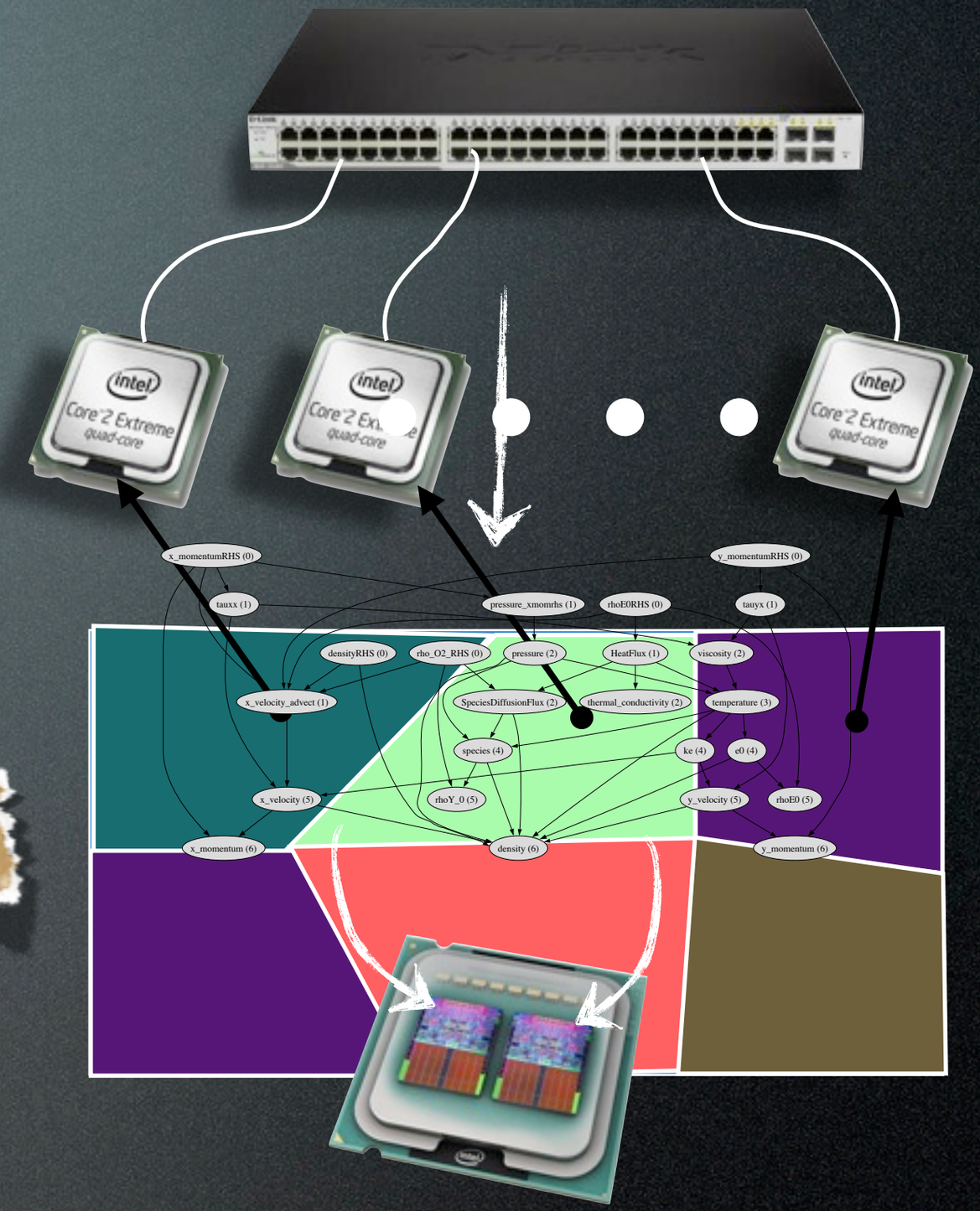
$$\frac{\partial m_k}{\partial t} + m_{k+1} = 0; \quad k = 0,1,\ldots,2n-1$$

$$m_{n+1} = \sum_{i=1}^{n} w_i r_i^i$$



Parallelism

- Domain Decomposition



- Algorithm Decomposition

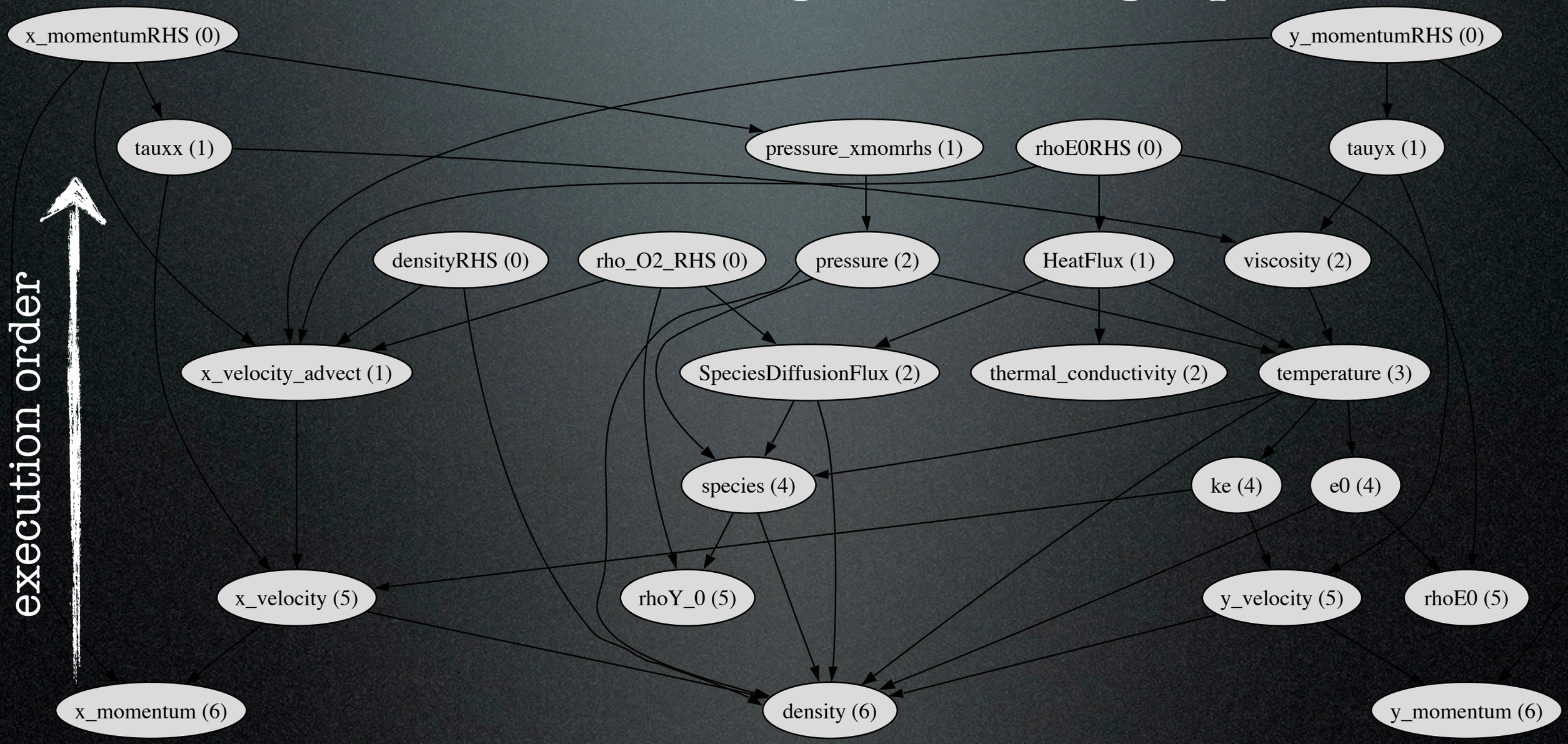
- Fine Grained Parallelism

$$\nabla^2\phi + s$$

One expression
(calculated on a patch/workset)

Priority Queue Threading

Allows “backfilling” based on graph



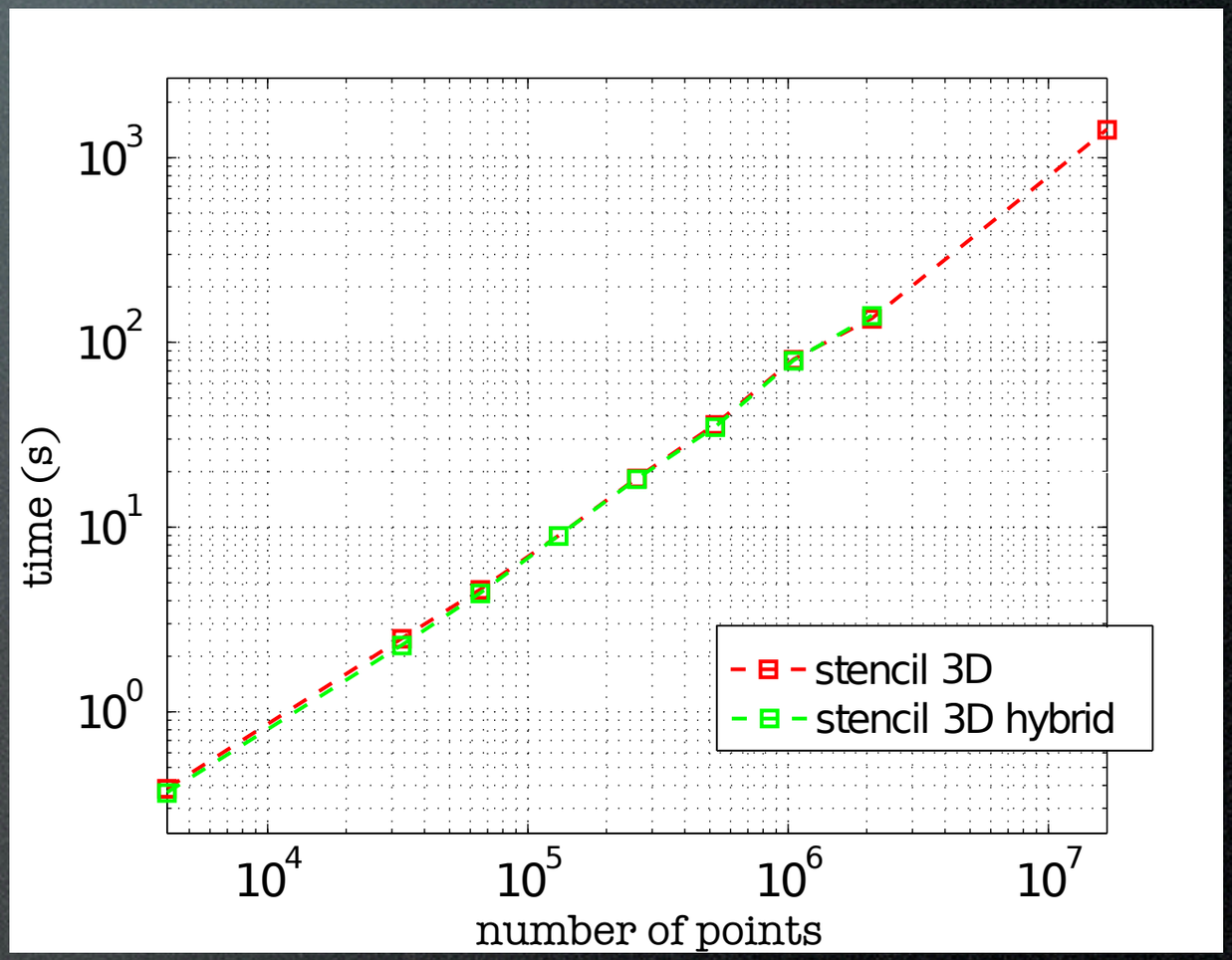
Each expression receives signals from its dependents when they complete execution. When all are done, the expression enters the priority queue.

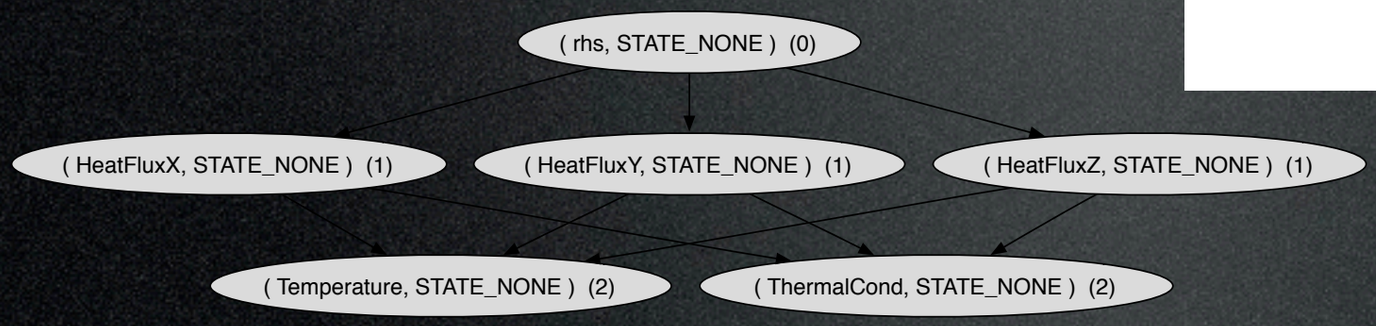
Overhead?

$$\frac{\partial T}{\partial t} = -\frac{1}{\rho c_p} \nabla \cdot (-\lambda \nabla T)$$

- Staggered, structured FV mesh
- Gradient, interpolant & divergence operators.

- The overhead of the expression graph approach does not contribute in any meaningful way to the execution time.





- Independent of parallel framework

Acknowledgments

- James C. Sutherland (PI)
  Assistant Prof. Chemical Engineering
- Matthew Might
  Assistant Prof., School of Computing
- Devin Robinson
  M. S. Student
- Christopher Earl
  Ph.D. Candidate

This material is based upon work supported by the National Science Foundation under Grant No. PetaApps-0904631.

This research was sponsored by the National Nuclear Security Administration under the Accelerating Development of Retrofitable CO2 Capture Technologies through Predictivity program through DOE Cooperative Agreement DE-NA0000740

Questions?

This material is based upon work supported by the National Science Foundation under Grant No. PetaApps-0904631.

This research was sponsored by the National Nuclear Security Administration under the Accelerating Development of Retrofitable CO2 Capture Technologies through Predictivity program through DOE Cooperative Agreement DE-NA0000740