NREL
NATIONAL RENEWABLE ENERGY LABORATORY

S3D Direct Numerical Simulation — Preparation for the 10–100 PF era

Ray W. Grout
Scientific Computing

ACSS, 29 March 2012

Ramanan Sankaran	ORNL
John Levesque	Cray
Cliff Woolley, Stan Posey	nVidia
J.H. Chen	SNL

NREL is a national laboratory of the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, operated by the Alliance for Sustainable Energy, LLC.

Key Questions

1. How S3D (DNS) can address the science challenges Jackie identified
2. Performance requirements of the science and how we can meet them
3. Optimizations and refactoring
4. What we can do on *Titan*
5. Future work

The governing physics

Compressible Navier-Stokes for Reacting Flows

- PDEs for conservation of momentum, mas, energy and composition
- Chemical reaction network governing composition changes
- Mixture averaged transport model
- Flexible thermochemical state description (IGL)
- Modular inclusion of case-specific physics
  - Optically thin radiation
  - Compression heating model
  - Lagrangian particle tracking

Solution Algorithm (What does S3D do?)

- Method of lines solution:
  - Replace spatial derivatives with finite-difference approximations to obtain coupled set of ODEs
  - $8\mathrm{th}$ order centered approximations to first derivative
  - Second derivative evaluated by repeated application of first derivative operator

$$\frac{\partial^2 \phi}{\partial x^2} \approx \frac{\partial}{\partial x}\left[\frac{\partial}{\partial x}\phi\right]$$

- Integrate explicitly in time
- Thermochemical state and transport coefficients evaluated pointwise
- Chemical reaction rates evaluated point-wise
- Block spatial parallel decomposition between MPI ranks

Solution Algorithm

- Fully compressible formulation
  - Fully coupled acoustic/thermochemical/chemical interaction
- No subgrid model: fully resolve turbulence-chemistry interaction
- Total integration time limited by large scale (acoustic, bulk velocity, chemical) residence time
- Grid must resolve smallest mechanical, scalar, chemical length-scale
- Time-step limited by smaller of chemical timescale or acoustic CFL

Resolution requirements in detail

1. Kolmogorov lengthscales

$$\eta \approx \frac{\Lambda}{Re_t^{(3/4)}}; \quad \Lambda = k_1 L \quad L = N\Delta x$$

$$\eta > k_2 \Delta x \Rightarrow Re_t^{(3/4)} < \frac{k_1}{k_2} N \Rightarrow Re_t < \left(\frac{k_1}{k_2}\right)^{4/3} N^{4/9}$$

2. Batchelor lengthscales

$$\lambda_\beta = \frac{\eta}{\sqrt{Sc}} \quad Sc = \frac{\nu}{D} \approx \mathcal{O}(1)$$

Hydrogen-air, $Sc \approx 0.2$; n-heptane-air, $Sc \approx 2.4$

3. Chemical lengthscales:

$$\Delta x < \frac{\delta}{Q} \Rightarrow \frac{L}{\delta} < \frac{N}{Q} \quad Q \approx 20$$

Resolution requirements (temporal)

1. Acoustic CFL

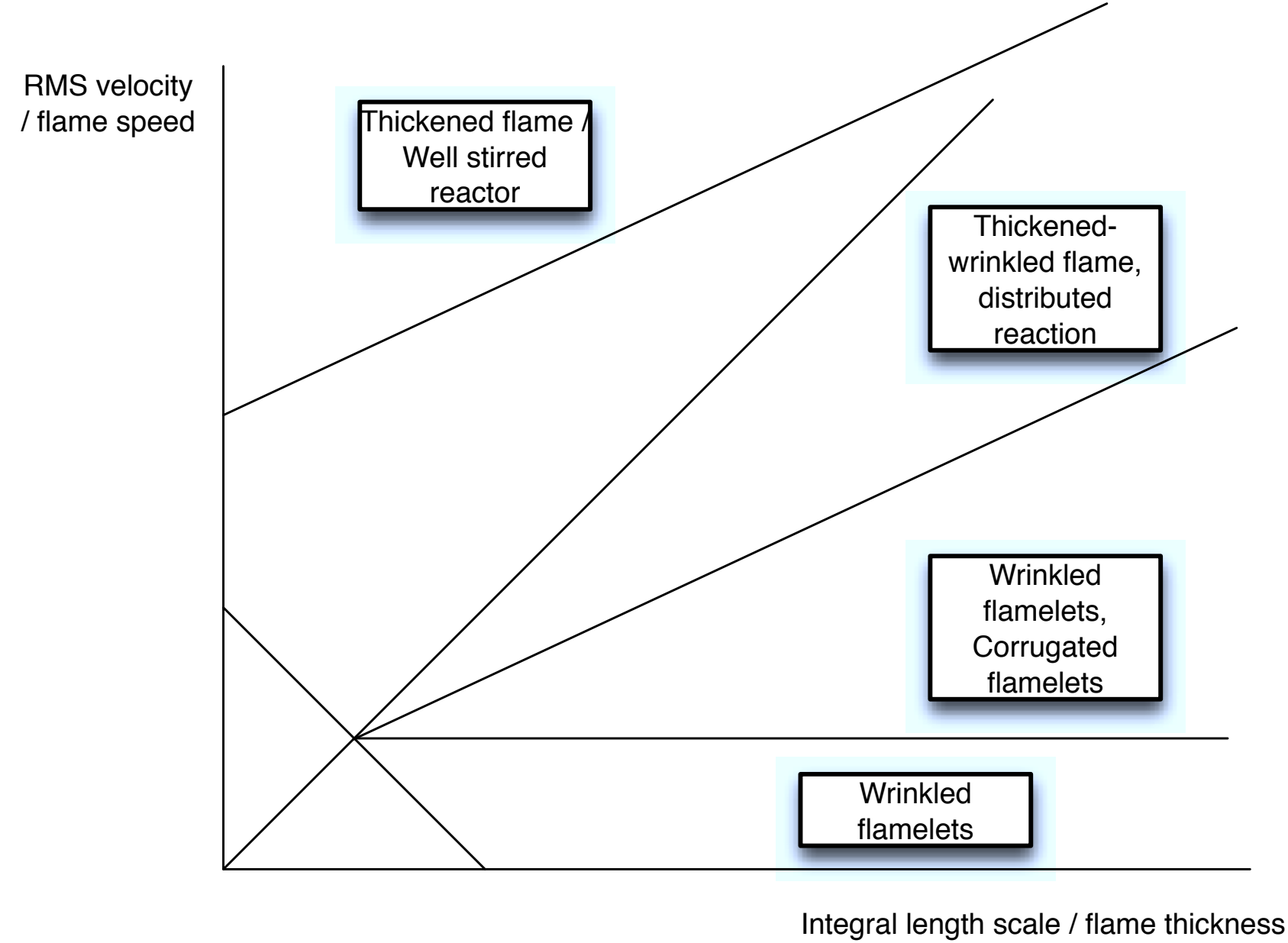
$$\Delta t < \frac{\Delta x}{a} \qquad \frac{\Delta t_a}{\Delta t_{u'}} = Ma$$

2. Advective CFL

$$\frac{\Delta t_a}{\Delta t_{u'}} = Ma$$

3. Chemical timescale
   - Flame timescale $\left(\tau_c \sim \frac{\delta}{s_L}\right)$
   - Species creation rates $\max\left(\dot{S}_i^{-1}\right)$
   - Reaction rates $\max\left(\dot{\omega}_j^{-1}\right)$
   - Eigenvalues of reaction rate jacobian

Combustion regimes



Chemistry reduction and stiffness removal

- Reduce species and reaction count through extensive static analysis and manipulation of reaction mechanism
- Literature from T. Lu, C.K. Law et al.
  - DRG analysis of reaction network
  - Quasi-steady state approximations
  - Partial equilibrium approximations
- *Dynamic* analysis to adjust reactions that are assumed ‘fast’ relative to diffusion at runtime (implications later)

Benchmark problem for development

- HCCI study of stratified configuration
- Periodic
- 52 species n-heptane/air reaction mechanism (with dynamic stiffness removal)
- Mixture average transport model
- Based on target problem sized for 2B gridpoints
- 48^3 points per node (hybridized)
- 20^3 points per core (MPI-everywhere)
- Used to determine strategy, benchmarks, memory footprint
- Alternate chemistry (22 species Ethylene-air mechanism) used as surrogate for ‘small’ chemistry

Evolving chemical mechanism

- 73 species bio-diesel mechanism now available; 99 species iso-octane mechanism upcomming
- Revisions to target late in process as state of science advances
- ‘Bigger’ (next section) and ‘more costly’ (last section)
- Continue with initial benchmark (acceptance) problem
  - Keeping in mind that all along we’ve planned on chemistry flexibility
  - Work should transfer
  - Might need smaller grid to control total simulation time

Target Science Problem

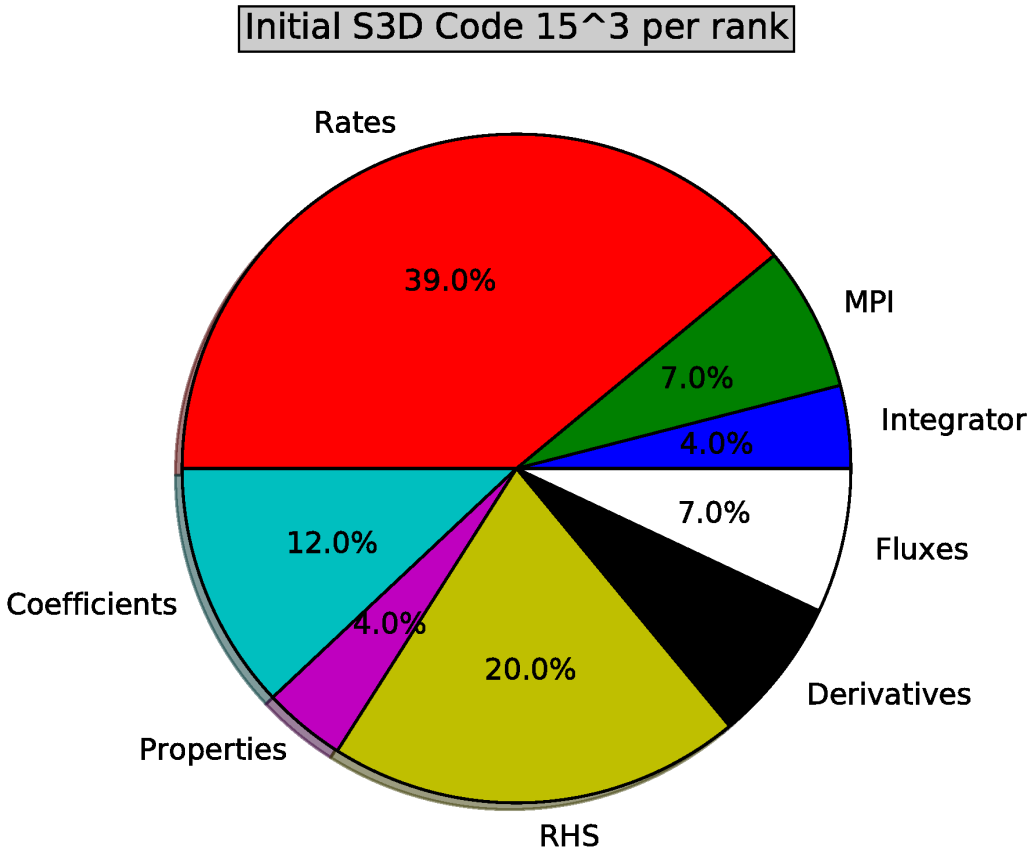
- Target simulation: 3D HCCI study
- Outer timescale: 2.5ms
- Inner timescale: 5ns $\Rightarrow$ 500 000 timesteps
- As ‘large’ as possible for realism:
  - Large in terms of chemistry: 73 species bio-diesel or 99 species iso-octane mechanism preferred, 52 species n-Heptane mechanism alternate
  - Large in terms of grid size: 900^3, 650^3 alternate

Summary (I)

- Provide solutions in regime targeted for model development and fundamental understanding needs
- Turbulent regime weakly sensitive to grid size: need a large change to alter Re_t significantly
- Chemical mechanism is significantly reduced in size from the full mechanism by external, static analysis to $\mathcal{O}(50)$ species

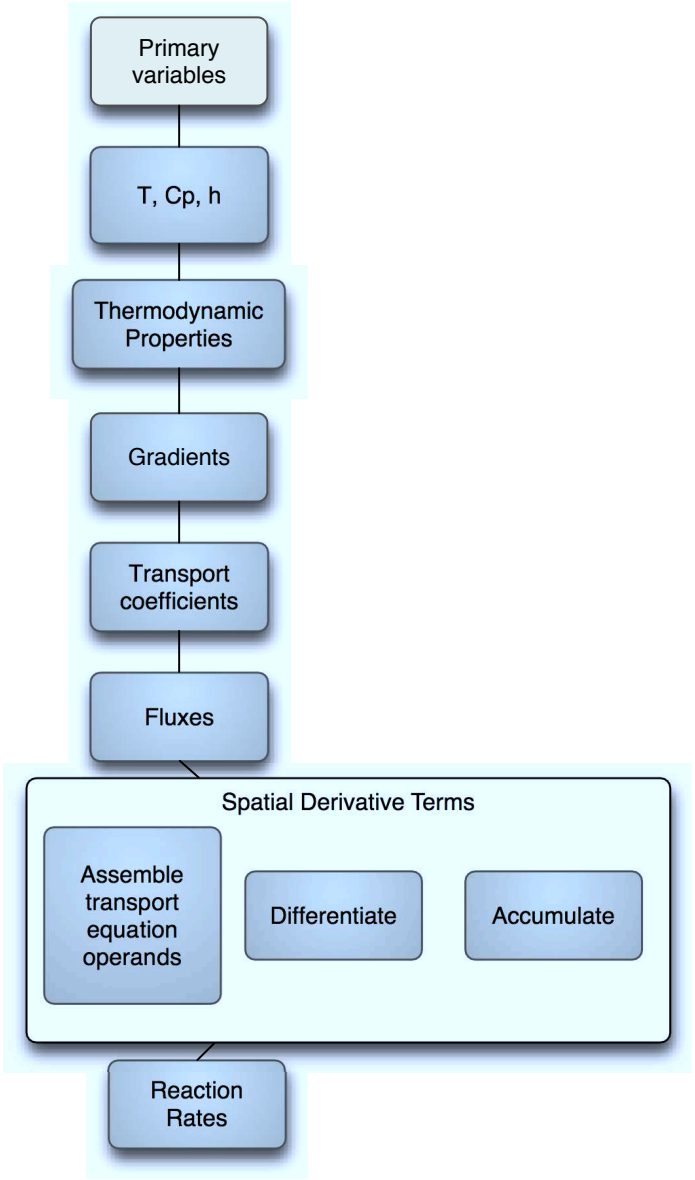
Performance profile for legacy S3D

Where we started (n-heptane)



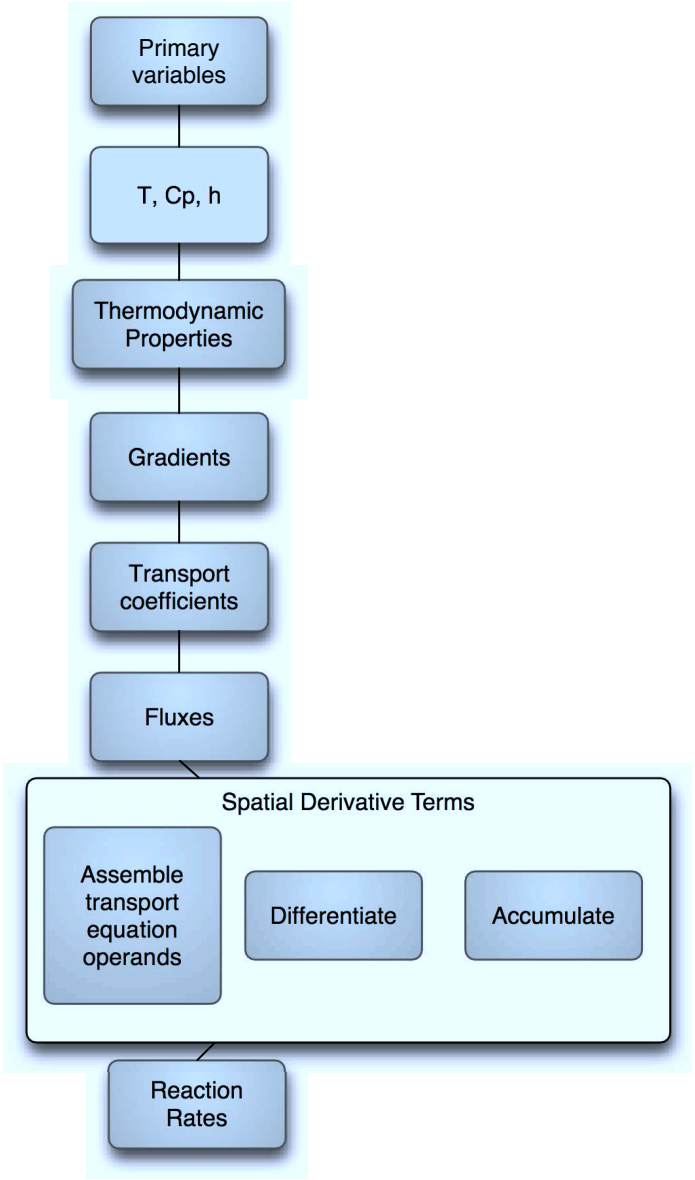
$24^2 \times 16$, 720 nodes	5.6s
$24^2 \times 16$, 7200 nodes	7.9s
48^3, 8 nodes	28.7s
48^3, 18 000 nodes	30.4s

S3D RHS



$$u = \frac{q_u(\vec{x}, t)}{\rho}; \quad Y_n = \frac{q_n(\vec{x}, t)}{\rho}$$

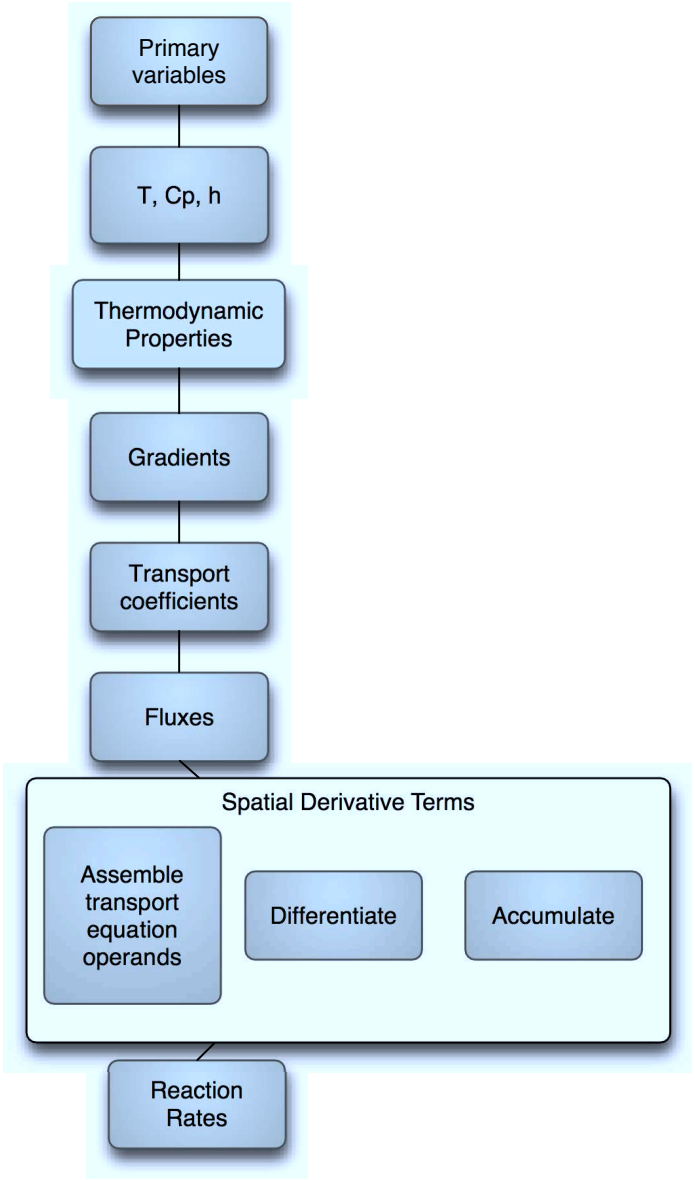
S3D RHS



$$Cp = p^4(T); \quad h = p^4(T)$$

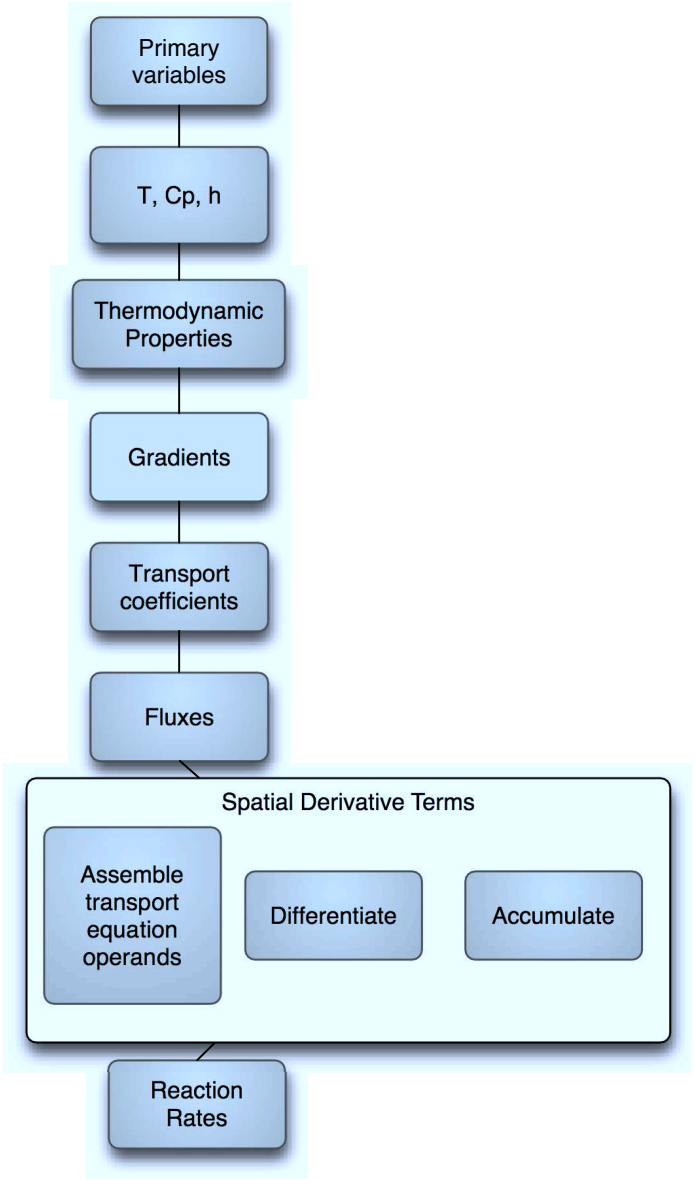
(Polynomials tabulated and linearly interpolated)

S3D RHS



$$p = \rho R T$$

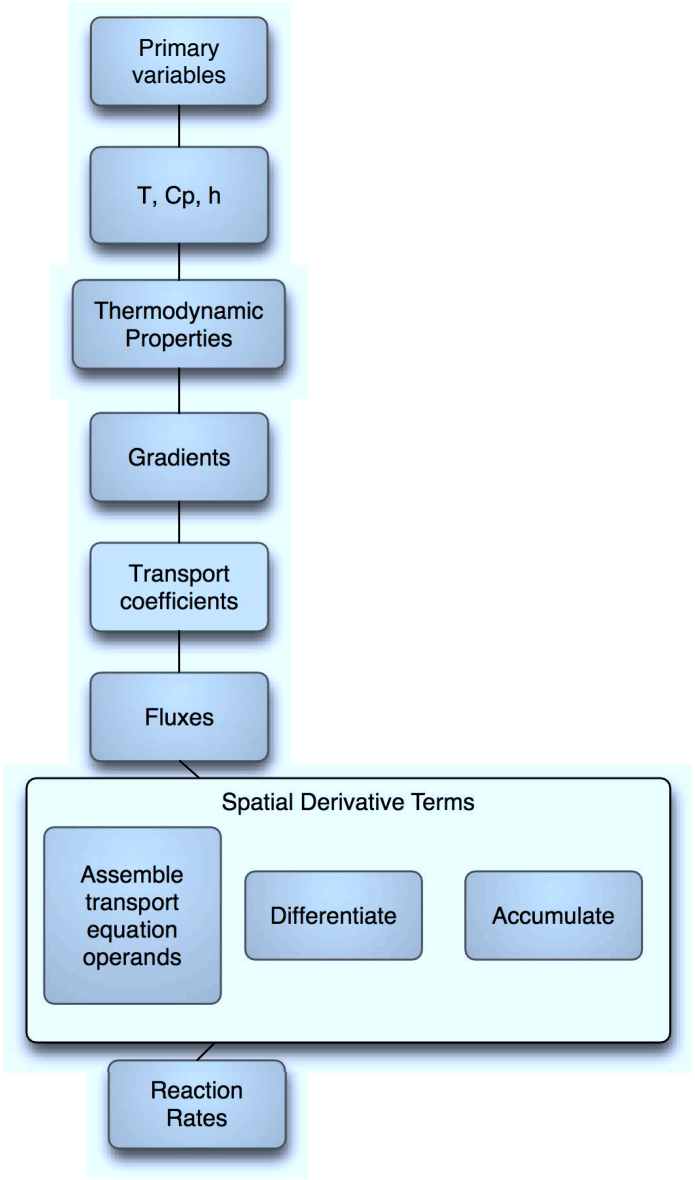
S3D RHS



$$\frac{\partial Y_n}{\partial x_i}; \quad \frac{\partial u_j}{\partial x_j}; \quad \frac{\partial T}{\partial x_i}$$

— Historically computed using sequential 1D derivatives

S3D RHS

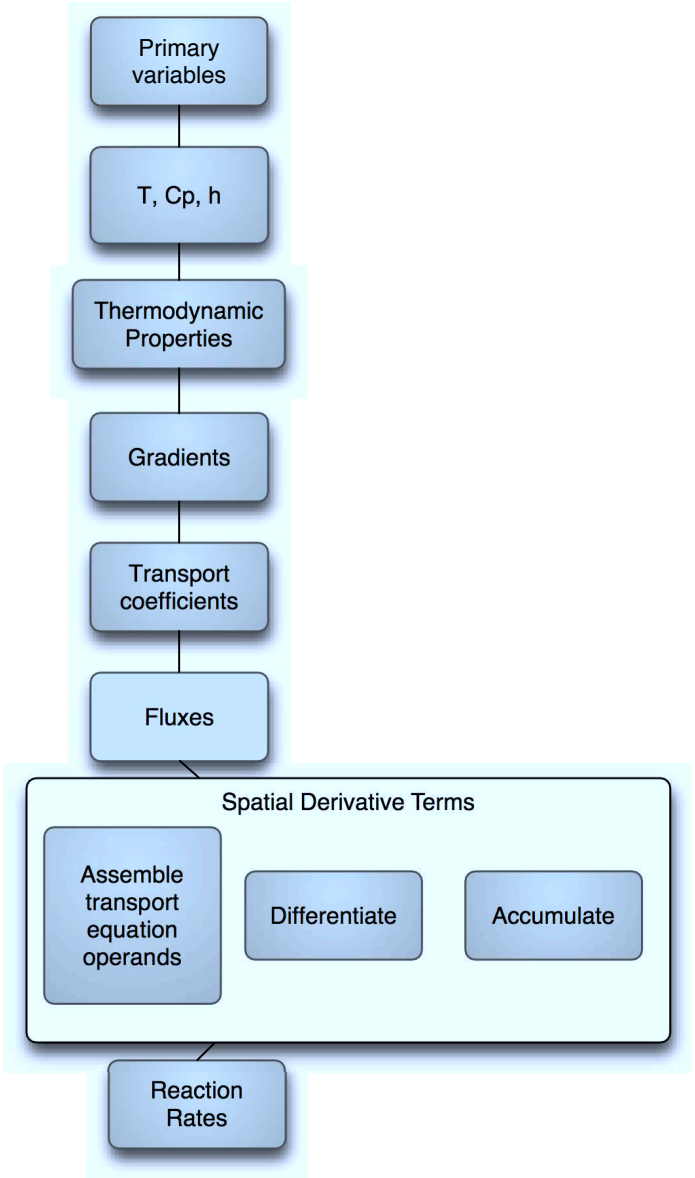


$$\lambda = f\left(T, \vec{X}\right) \quad \mu = f\left(T, \vec{X}\right)$$

$$\overline{D}_n = f\left(T, \vec{X}, p\right)$$

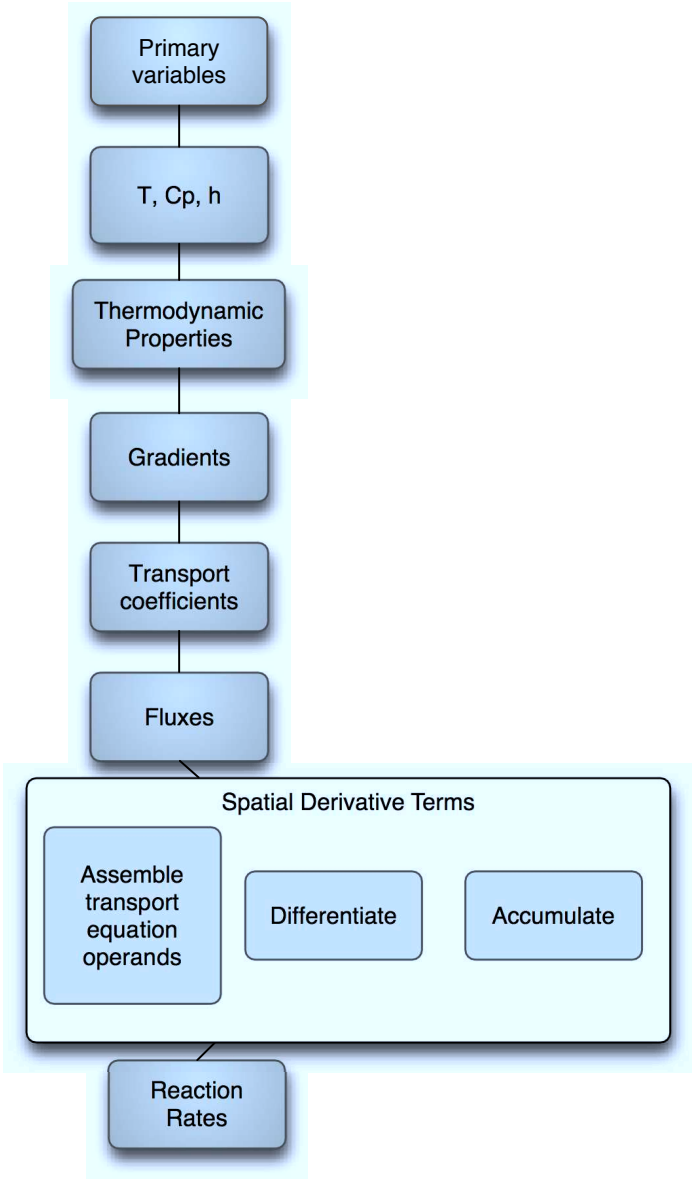
(these polynomials evaluated directly)

S3D RHS



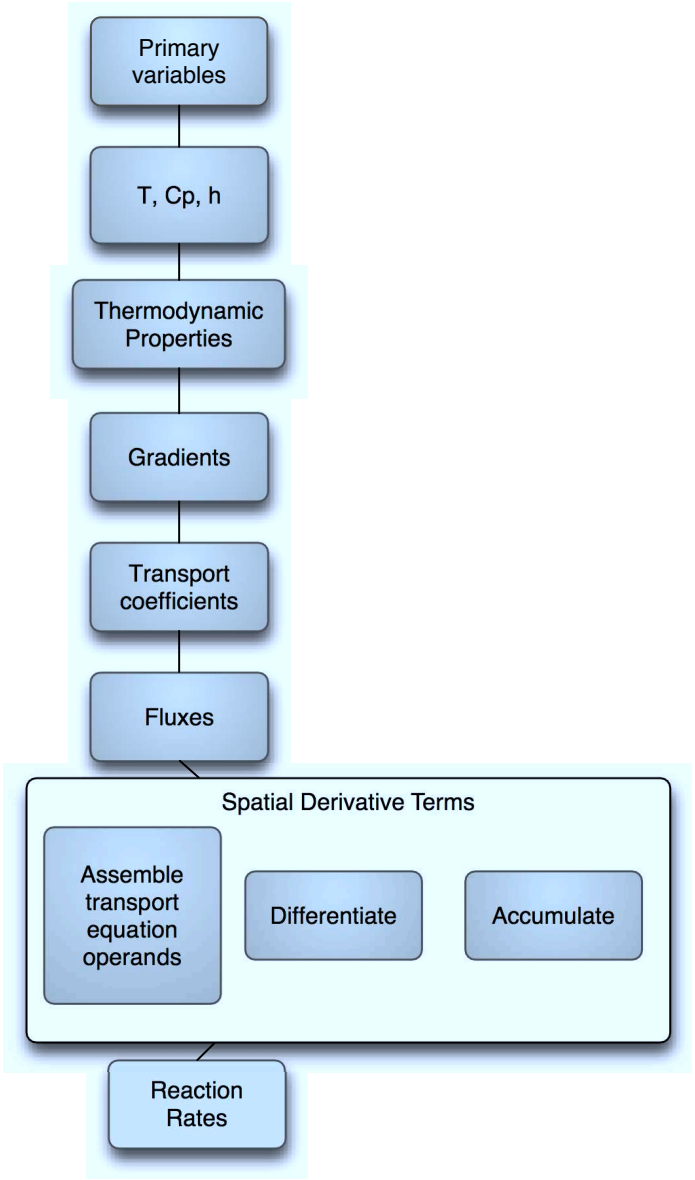
$$\tau_{ij} = 2\mu \left[\frac{\partial u_k}{\partial x_k} - \left(\frac{1}{3} \frac{\partial u_k}{\partial x_k} \right) \right] \delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

S3D RHS



$$\frac{\partial \rho u}{\partial t} = \frac{\partial}{\partial x}\left[-\rho u u - p + \tau_{xx}\right] - \frac{\partial}{\partial y}\left[\rho u v - \tau_{xy}\right] - \frac{\partial}{\partial z}\left[\rho u w - \tau_{xz}\right]$$

S3D RHS



$$A + B \Leftrightarrow C + D$$

$$k_f = A_{fj} T^{\beta_j} \exp\left(\frac{-T_{aj}}{T}\right)$$

$$R_f = [A][B]k_f$$

$$\dot{S}_n = W_n \sum_{j=1}^{M} \nu_{kj} R_j$$

Communication in Chemical Mechanisms

- Need diffusion term separately from advective term to facilitate dynamic stiffness removal
  - See T. Lu et al., Combustion and Flame 2009.
  - Application of quasi-steady state (QSS) assumption *in situ*
  - Applied to species that are transported, so applied by correcting reaction rates (traditional QSS doesn't conserve mass if species transported)
- Diffusive contribution usually lumped with advective term:

$$\frac{\partial}{\partial x}\left(\rho u Y - J_x\right)$$

- We need to break it out separately to correct R_f, R_b

Readying S3D for Titan

— Migration strategy:

1. Requirements for host/accelerator work distribution
2. Profile legacy code (previous slides)
3. Identify key kernels for optimization
   - Chemistry, transport coefficients, thermochemical state (pointwise)
   - Derivatives (reuse)
4. Prototype and explore performance bounds using cuda
5. “Hybridize” legacy code: MPI for inter-node, OpenMP intra-node
6. OpenACC for GPU execution
7. Restructure to balance compute effort between accelerator and host

Chemistry

- Reaction rate — temperature dependence
  - Need to store rates: temporary storage for R_f, R_b
- Reverse rates from equilibrium constants or separate set of constants
- Multiply forward/reverse rates by concentrations
- Number of algebraic relationships involving non-contiguous access to rates scales with number of QSS species
- Species source term is algebraic combination of reaction rates (non-contiguous access to temporary array)
- Extracted as a ‘self-contained’ kernel; analysis by nVidia suggested several optimizations
- Captured as improvements in code generation tools (see Sankaran, AIAA 2012)

Move everything over . . .

— Memory footprint for 48^3 gridpoints per node

	52 species n-Heptane	73 species bio-diesel
Primary variables	57	78
Primitive variables	58	79
Work Variables	280	385
Chemistry Scratch [a]	1059	1375
RK Carryover	114	153
RK Error control	171	234
Total	1739	2307
MB for 48^3 points	1467	1945

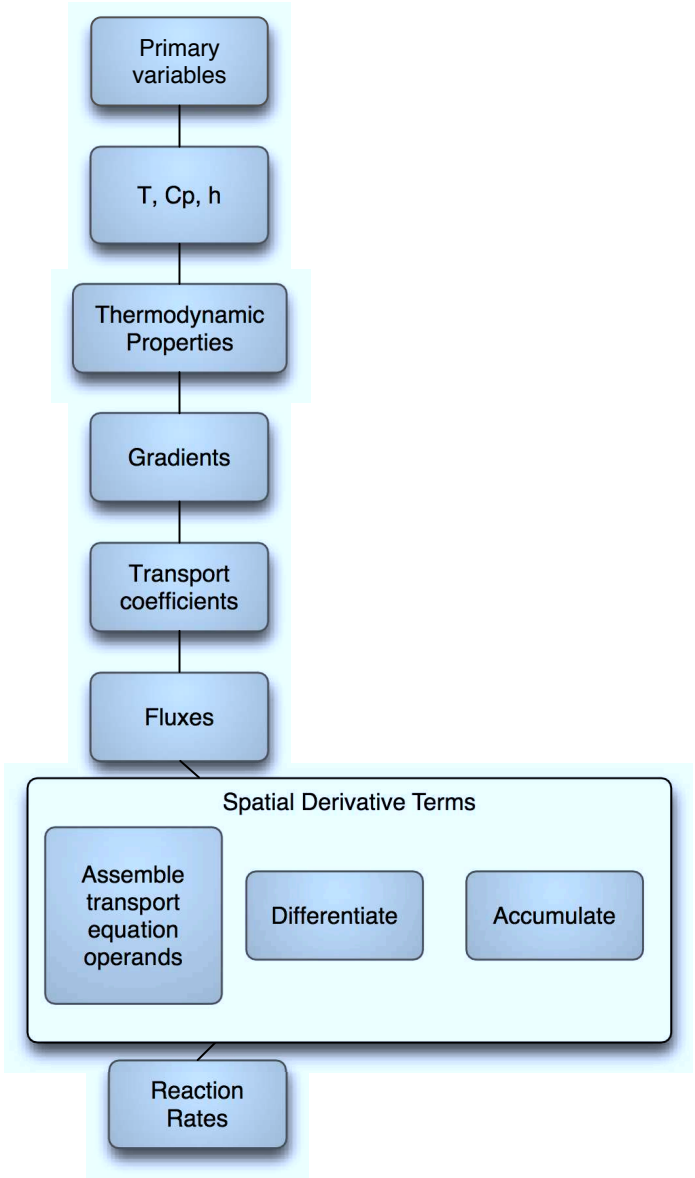
[a] For evaluating all gridpoints together

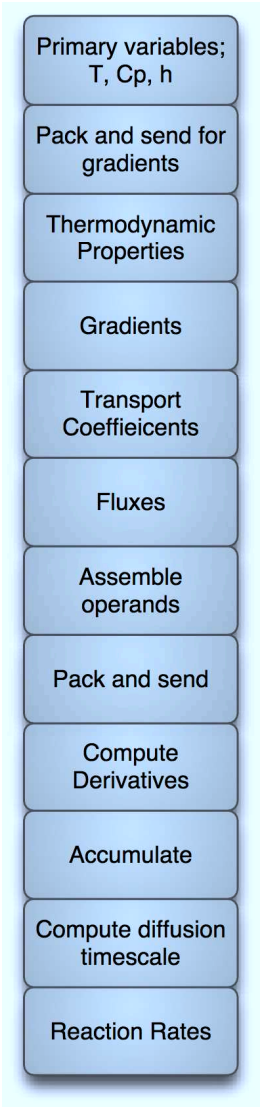
Communication aggregation

```
for all species do
    MPI_IRecv
    snd_left( 1:4,:,:) = f(1:4,:,:,i)
    snd_right( 1:4,:,:) = f(nx-3:nx,:,:,i)
    MPI_ISend
    evaluate interior derivative
    MPI_Wait
    evaluate edge derivative
end for
```

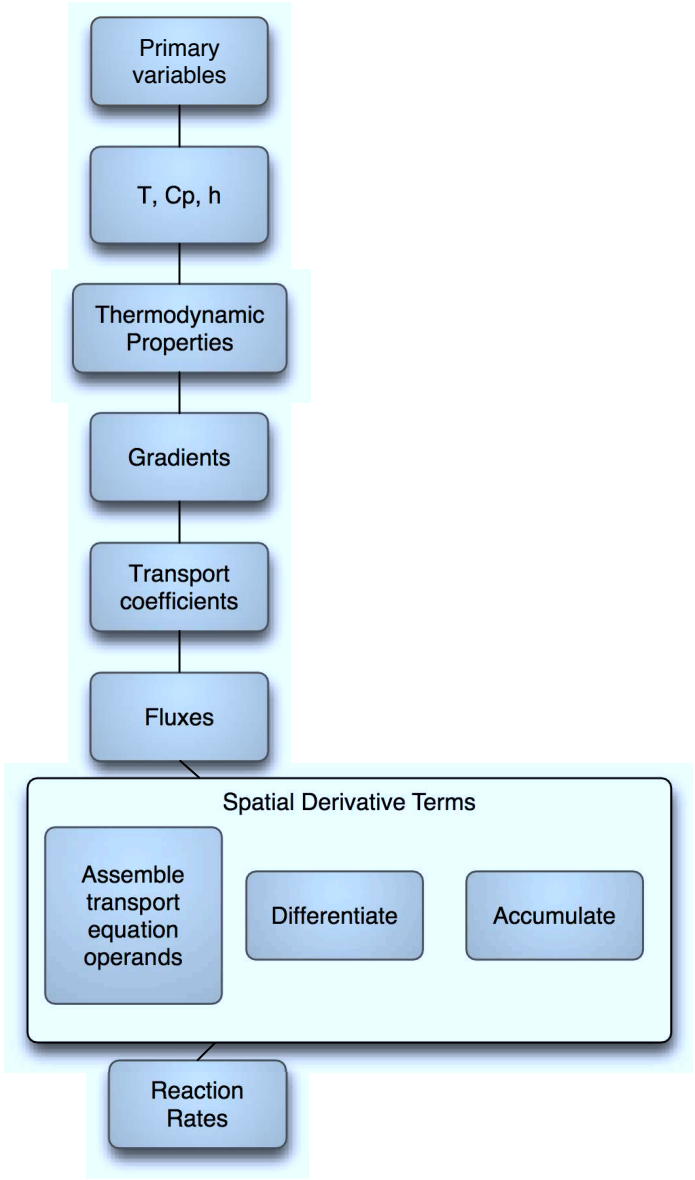
```
for all species do
    MPI_IRecv
end for
for all species do
    snd_left( 1:4,:,:,i) = f(1:nx,:,:,i)
    snd_right( 1:4,:,:,i) = f(nx-3:nx,:,:,i)
end for
for all species do
    MPI_ISend
end for
MPI_Wait
for all species do
    evaluate interior derivative
    evaluate edge derivative
end for
```

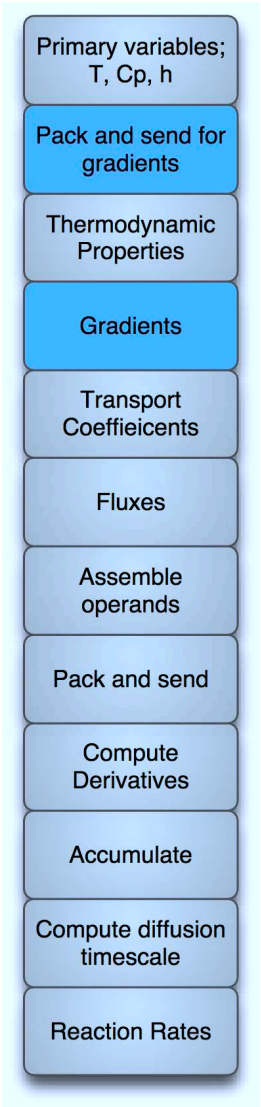
RHS reorganization



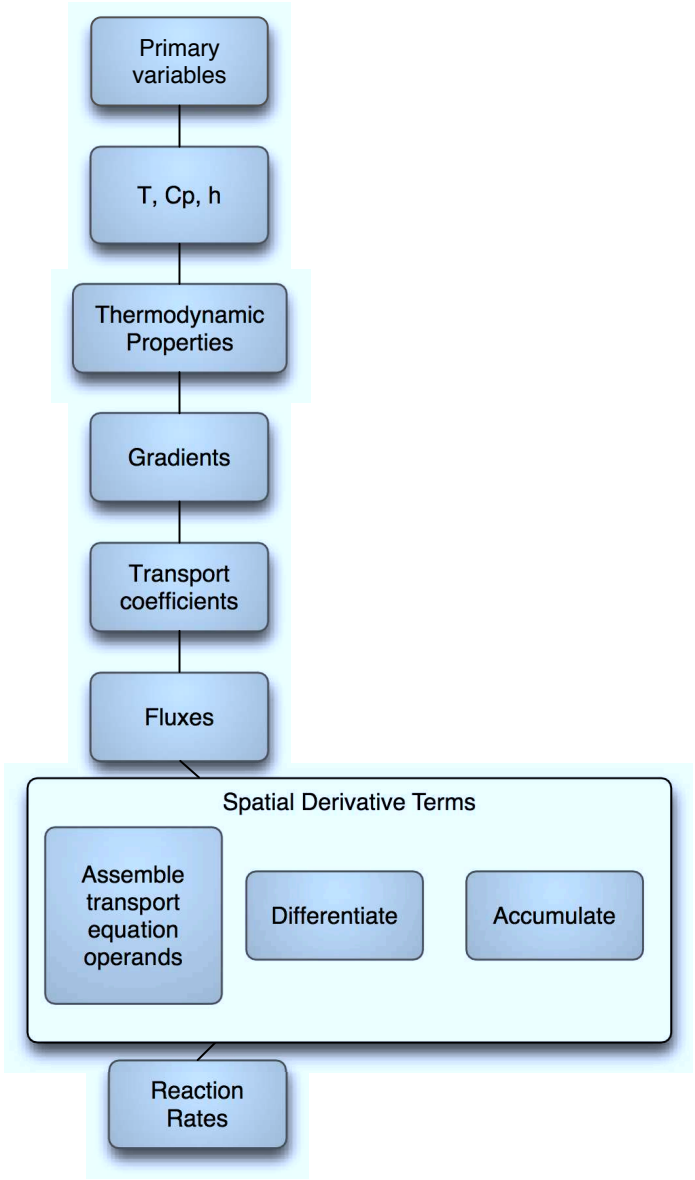


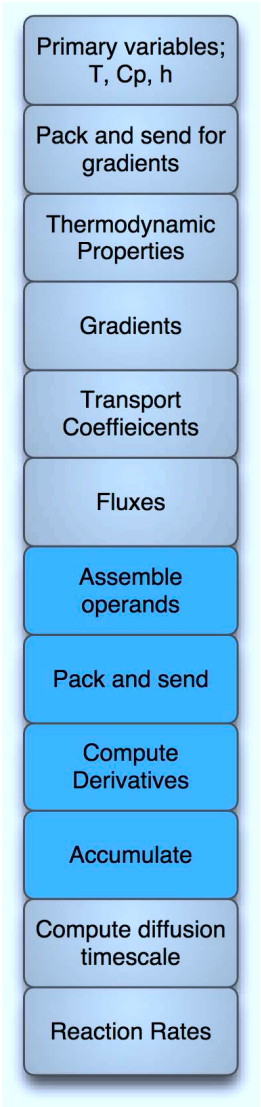
RHS reorganization





RHS reorganization





Optimize ∇Y for reuse

- Legacy approach: compute components sequentially:

  for all interior i, j, k **do**

  $$\frac{\partial Y}{\partial x} = \sum_{l=1}^{4} c_l \left(Y_{i+l,j,k} - Y_{i-l,j,k} \right) sx_i$$

  end for

  for all i, interior j, k **do**

  $$\frac{\partial Y}{\partial y} = \sum_{l=1}^{4} c_l \left(Y_{i,j+l,k} - Y_{l,j-l,k} \right) sy_j$$

  end for

  for all i, j, interior k **do**

  $$\frac{\partial Y}{\partial z} = \sum_{l=1}^{4} c_l \left(Y_{i,j,k+l} - Y_{i,j,k-l} \right) sz_k$$

  end for

- Points requiring halo data handled in separate loops

Optimize ∇Y for reuse

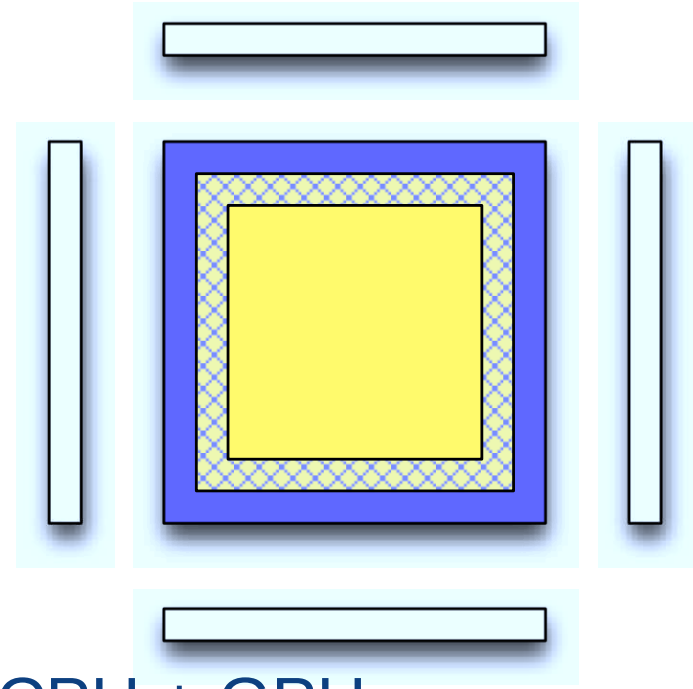
— Combine evaluation for interior of grid

for all ijk **do**
if interior i **then**
$\frac{\partial Y}{\partial x} = \sum_{l=1}^{4} c_l \left(Y_{i+l,j,k} - Y_{i-l,j,k} \right) sx_i$
end if
if interior j **then**
$\frac{\partial Y}{\partial y} = \sum_{l=1}^{4} c_l \left(Y_{i,j+l,k} - Y_{l,j-l,k} \right) sy_j$
end if
if interior k **then**
$\frac{\partial Y}{\partial z} = \sum_{l=1}^{4} c_l \left(Y_{i,j,k+l} - Y_{i,j,k-l} \right) sz_k$
end if
end for

— Writing interior without conditionals requires 55 loops
– $4^3, 4^2(N-8), 4(N-8)^2, (N-8)^3$ points

Restructure to rebalance

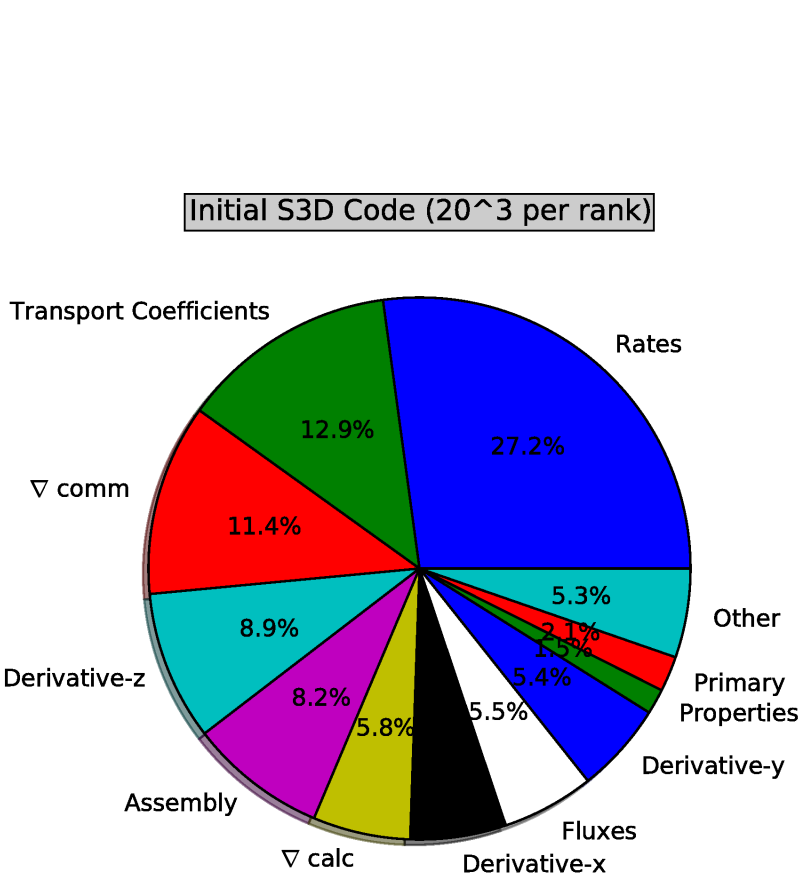
- Compute derivative “inner-halos” on host
- Data traffic ($\sim 30\%$), but move work to host

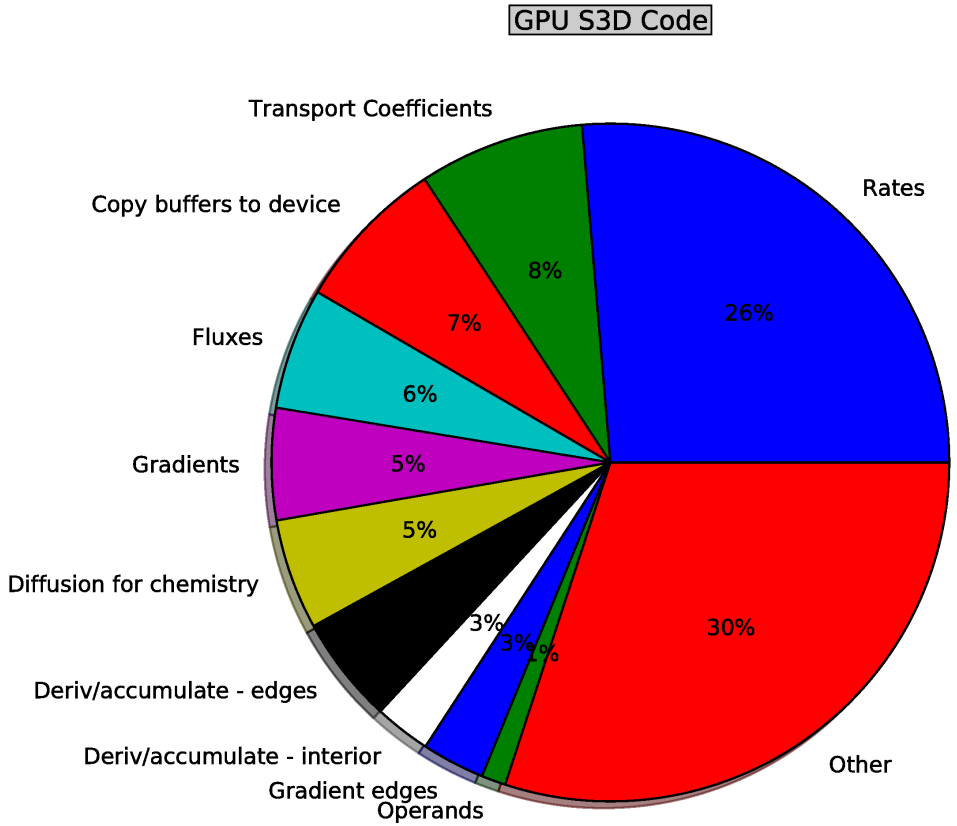


	All GPU	CPU + GPU
Halo→host	$nx^3 - (nx-8)^3$	$nx^3 - (nx-8)^3$
Halo→device	$nx^3 - (nx-8)^3$	0
Interior buffer→host	0	$(nx-8)^3 - (nx-16)^3$
Result→device	0	$nx^3 - (nx-8)^3$

New cost profile

Overall: GPU vs original vs hybrid performance





Summary (2)

1. Significant restructuring to expose node-level parallelism
2. Resulting code is hybrid MPI+OpenMP and MPI+OpenACC (-DGPU only changes directives)
3. Optimizations to overlap communication and computation
4. Changed balance of effort
5. For small per-rank sizes, accept degraded cache utilization in favor of improved scalability

Reminder: Target Science Problem

- Target simulation: 3D HCCI study
- Outer timescale: 2.5ms
- Inner timescale: 5ns $\Rightarrow$ 500 000 timesteps
- As ‘large’ as possible for realism:
  - Large in terms of chemistry: 73 species bio-diesel or 99 species iso-octane mechanism preferred, 52 species n-Heptane mechanism alternate
  - Large in terms of grid size: 900^3, 650^3 alternate

Benchmark problem

— 1200^3, 52 species n-Heptane mechanism

	7200 nodes			18000 nodes		
	XK6 (no GPU)	XK6 (GPU)	XE6 (2 CPU)	XK6 (no GPU)	XK6 (GPU)	XE6 (2 CPU)
Adjustment	3.23	2.2	2.4	1.5	1.0	1.1
Size per node		62^3			48^3	
WC per timestep	8.4	5.6	6	3.9	2.58	2.78
Total WC time (days)	48.6	32.4	34.7	22.6	15	16.1

— Very large by last years' standards — 225M core-hours

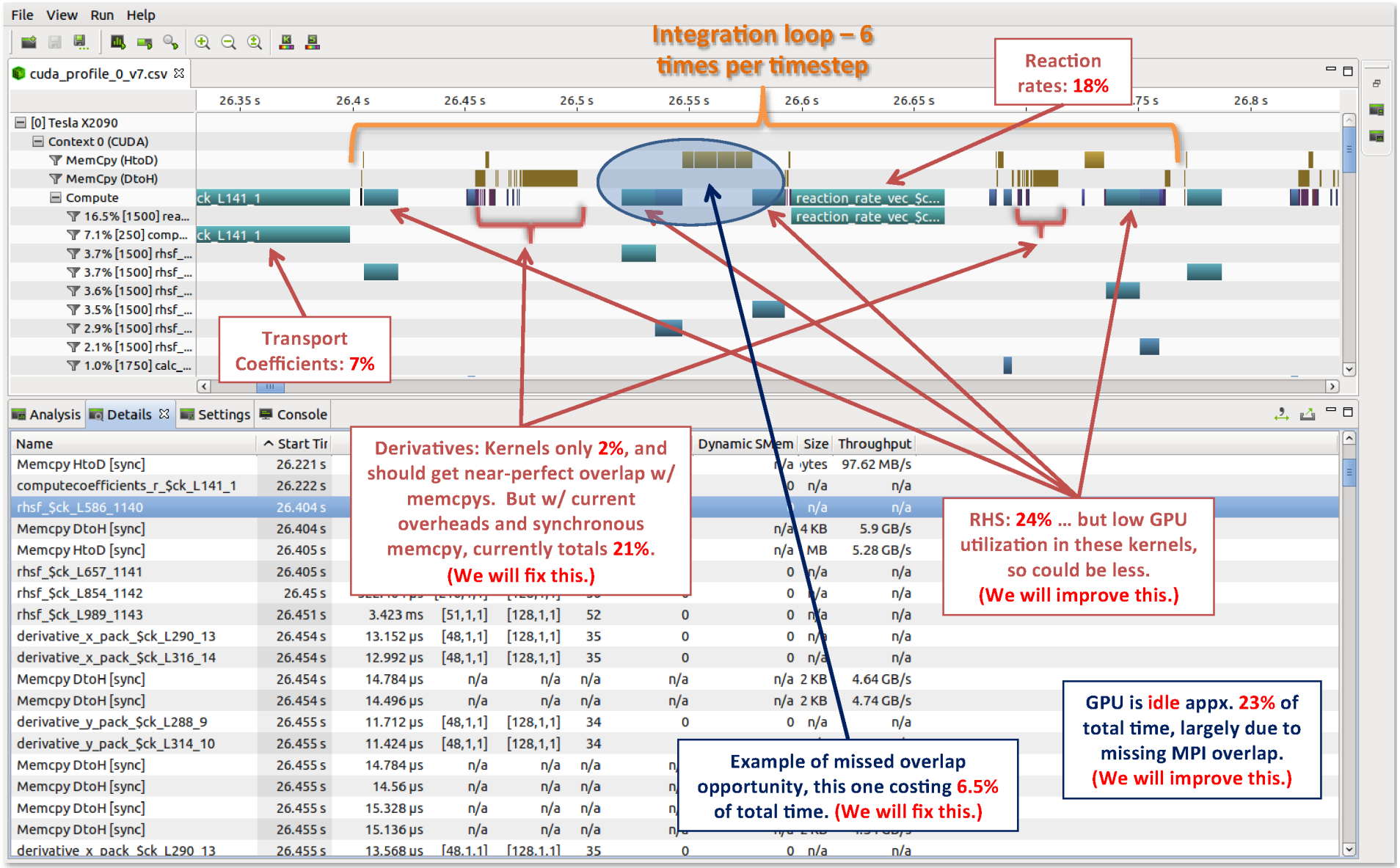
Time to solution

Problem		7200 nodes		18000 nodes	
		CPU	CPU+GPU	CPU	CPU+GPU
650^3, 52 spc	Size per node	35^3		25^3	
		(6859, 665^3)		(17576, 650^3)	
	WC per timestep	1.5	1.0	0.55	0.36
	Total WC time	8.8	5.8	3.2	2.1
900^3, 52 spc	Size per node	46^3		35^3	
		(8000, 920^3)		(17576, 910^3)	
	WC per timestep	3.4	2.3	1.5	1.0
	Total WC time	20	13	8.8	5.8

Time to solution

Problem		7200 nodes		18000 nodes	
		CPU	CPU+GPU	CPU	CPU+GPU
650^3, 73 spc	Size per node	35^3		25^3	
		(6859, 665^3)		(17576, 650^3)	
	WC per timestep	2.1	1.4	0.77	0.51
	Total WC time	12.3	8.1	4.5	3
900^3,73 spc	Size per node	46^3		35^3	
		(8000, 920^3)		(17576, 910^3)	
	WC per timestep	4.8	3.2	2.1	1.4
	Total WC time	28	18	12.3	8.1

Further optimization potential



Temporary storage

- Two of the main time consuming kernels (reaction rates, transport coefficients) generate significant intermediate results
- Reaction rate ‘spill’

S3D Profile Analysis

Limiting Factor Analysis	
MEM	8%
LATENCY	67%
INSN	26%

reaction_rate_vec_$ck_L165_1	% total L1 traffic	% total time [work]	% total time [wait]	% total time
global loads	16%	1%	12%	13%
global stores	9%	1%	0%	1%
local loads	38%	3%	10%	13%
local stores	36%	3%	27%	30%
replays	-	4%	-	4%
dependent insn latency	-	-	10%	10%
control flow	-	1%	8%	9%
integer ops	-	14%	-	14%
fp32 ops	-	3%	-	3%
fp64 ops	-	4%	-	4%
sfu ops	-	0%	-	0%
			total	101%

- We are working to expose another dimension of parallelism to improve this and permit evaluating much large reaction mechanisms.

Future algorithmic improvements

- Second Derivative approximation
- Chemistry network optimization to minimize working set size
- Replace algebraic relations with in place solve
- Time integration schemes - coupling, semi-implicit chemistry
- Several of these are being looked at by *ExaCT* co-design center, where the impacts on future architectures are being evaluated
  - Algorithmic advances can be back-ported to this project

Outcomes

- Reworked code is ‘better’: more flexible, well suited to both manycore and accelerated
  - GPU version required minimal overhead using OpenACC approach
  - Potential for reuse in derivatives favors optimization (chemistry not easiest target despite *exps*
- We already have ‘Opteron + GPU’ performance exceeding 2 Opteron performance
  - Majority of work is done by GPU: extra cycles on CPU for new physics (including those that are not well suited to GPU)
  - We have the ‘hard’ performance
  - Specifically moved work back to the CPU

Outcomes

- Significant scope for further optimization
  - Performance tuning
  - Algorithmic
  - Toolchain
  - Future hardware
- Broadly useful outcomes
- **Software is ready to meet the needs of scientific research now and to be a platform for future research**
  - We can run as soon as the Titan build-out is complete …