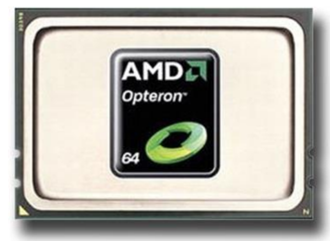
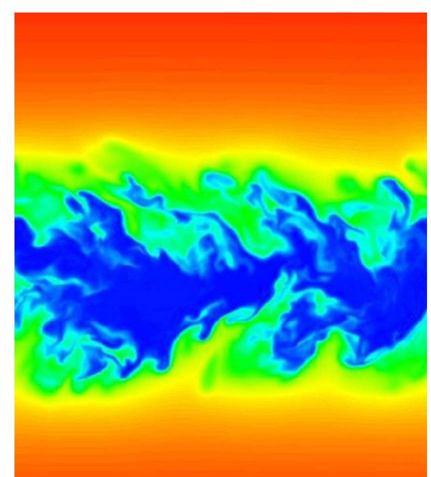
Hybrid Simulations Using CPU-GPU Paradigm for Reacting Flows in *Accelerating Industrial Competitiveness through Extreme-Scale Computing*



+



Jerry Lee, Vaidya Sankaran,
United Technologies Research Center
UTC, East Hartford

Acknowledgement: Vivek Venugopal, Hui Gao for helping to implement the GPU code

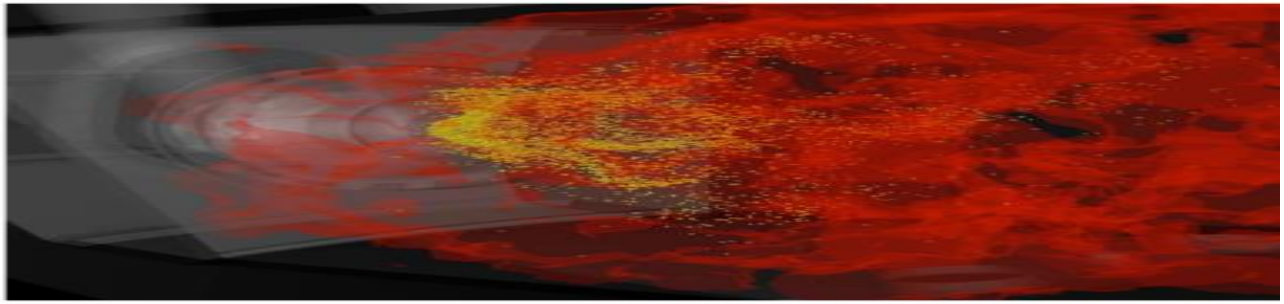
Special Thanks to

Dr. Ramanan Sankaran,

Dr. Suzy Tichenor & Dr. Jack Wells

Oak Ridge National Laboratories, USA.

Reactive flow adds a lot more PDEs to cold flow CFD



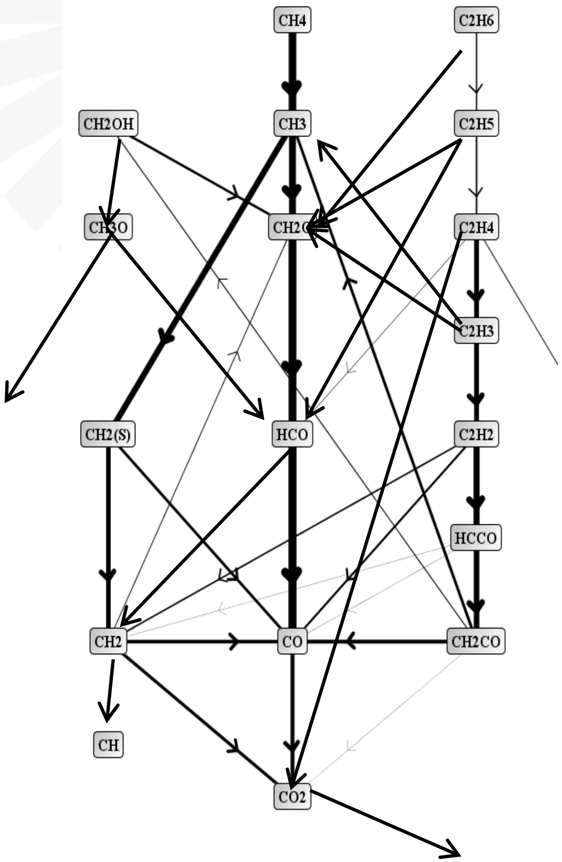
$$\frac{\partial \rho A}{\partial t} = \frac{\partial \rho u_i A}{\partial x_i} + \frac{\partial \rho V_i A}{\partial x_i} + \dot{\omega}_A$$

$$A = 1, u_i, e, Y_\mu;\ \mu = 1, \ldots, N_s; i = 1,2,3$$

$$N_s \sim 40$$

Combustion adds a lot more PDEs, 9x cold flow

"Fuel+O_2→ product+heat" has lot of paths, steps, and transcendental functions



ONE step:

$$H + O_2 + (M) \rightarrow HO_2 + (M)$$

$$\frac{dY_H}{dt} = k[H]^{\alpha}[O_2]^{\beta}[M]$$

$$k = k_{\infty}\left(\frac{\mathrm{Pr}}{1+\mathrm{Pr}}\right)F$$

$$k_0 = A_0 T^{\beta_0} \exp(-E_0 / RT)$$

$$k_{\infty} = A_{\infty} T^{\beta_{\infty}} \exp(-E_{\infty} / RT)$$

$$\mathrm{Pr} = \frac{k_0[M]}{k_{\infty}}$$

$$\log(F) = \left(1 + \left(\frac{\log(\mathrm{Pr}) + c}{n - d(\log(\mathrm{Pr}) + c)}\right)^2\right)^{-1} \log(F_c)$$

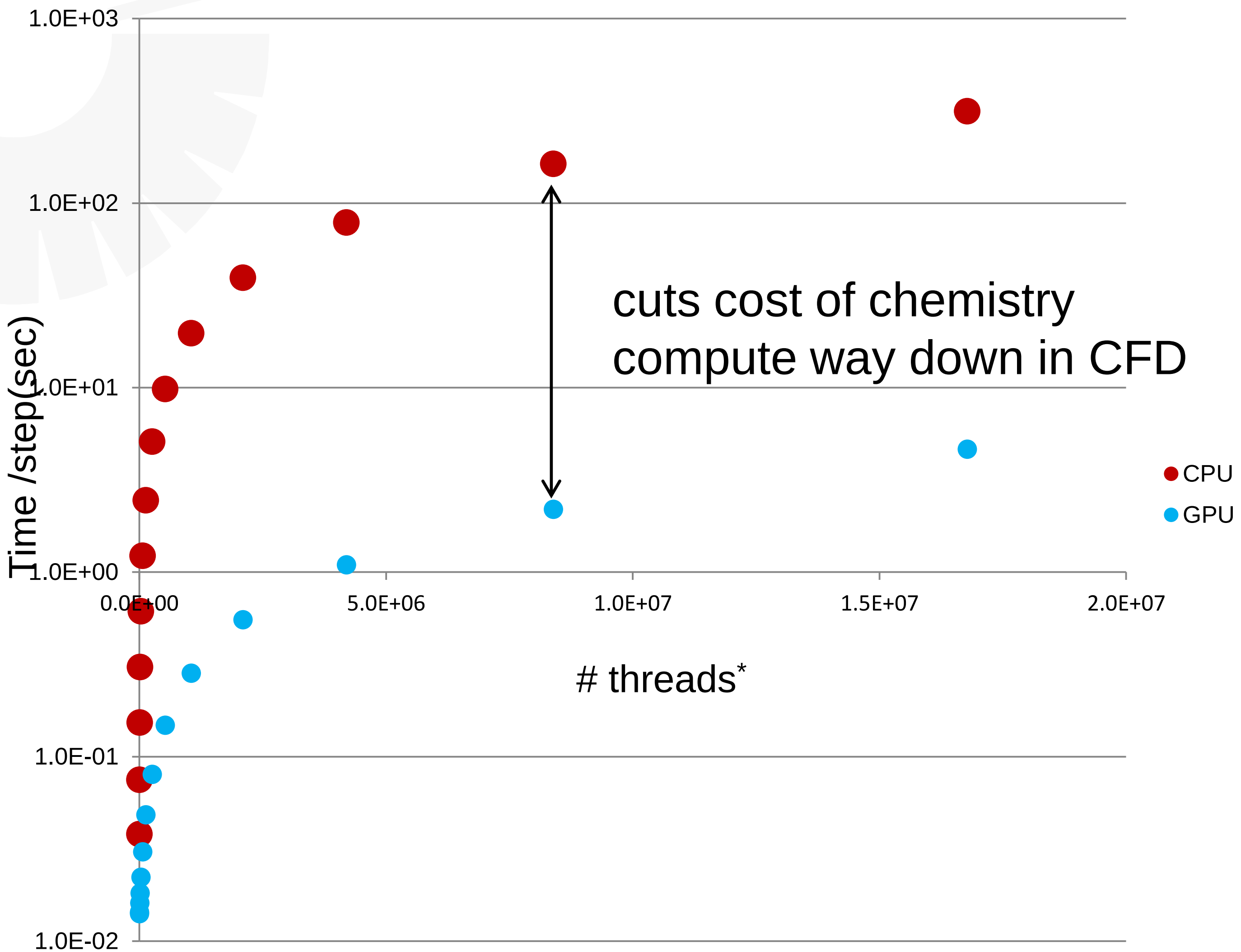
$$F_c = (1-\alpha)\exp(-T/T^{***}) + \alpha \exp(-T/T^{**}) + \exp(-T^{**}/T)$$

$$c, n, d = \text{linear functions of } \mathrm{F_c}$$

14 transcendental

~200-300 steps for jet fuel!

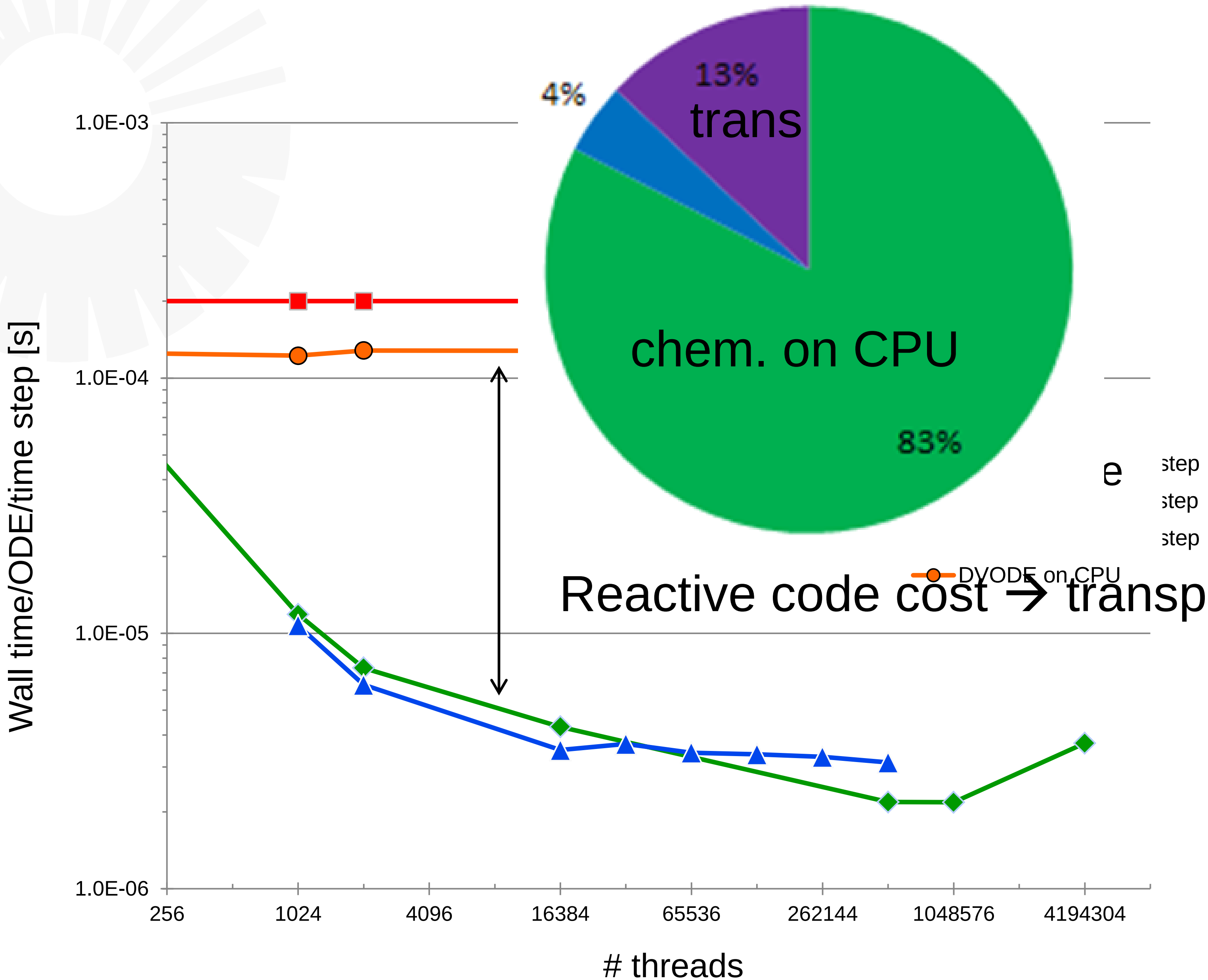
Performance of standalone GPU chem solver (explicit)



* 1 thread = 1 DOE system

United Technologies Research Center

Performance of standalone GPU chem solver (implicit)



Reactive code cost → transport code

Operator splitting in CFD → thread the chemistry compute

Integrate terms in tandem

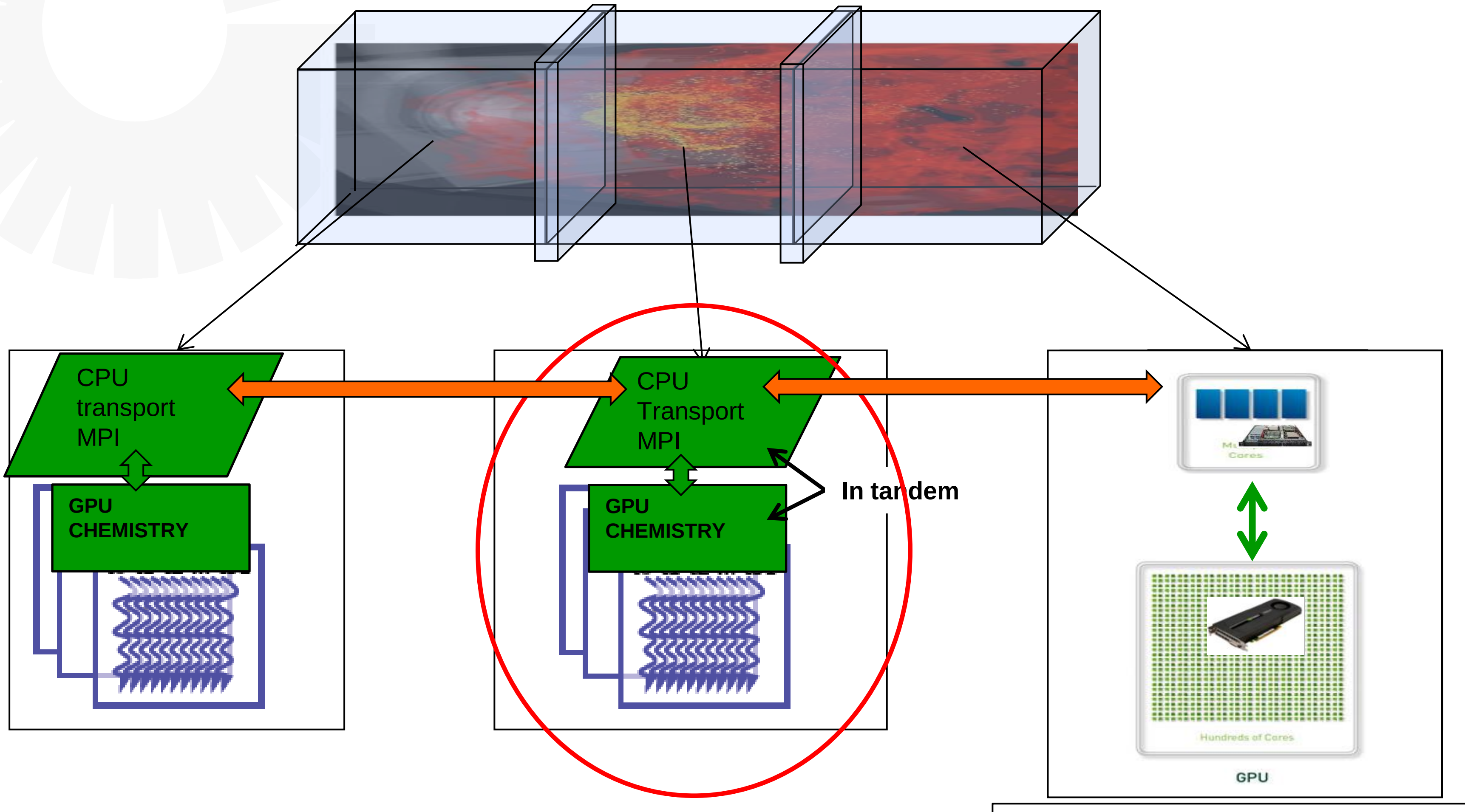
$$\frac{\partial \rho A}{\partial t} = \frac{\partial \rho u_i A}{\partial x_i} + \frac{\partial \rho V_i A}{\partial x_i} + \dot{\omega}_A$$

Independent of neighbors

Collect from many cells and do threading

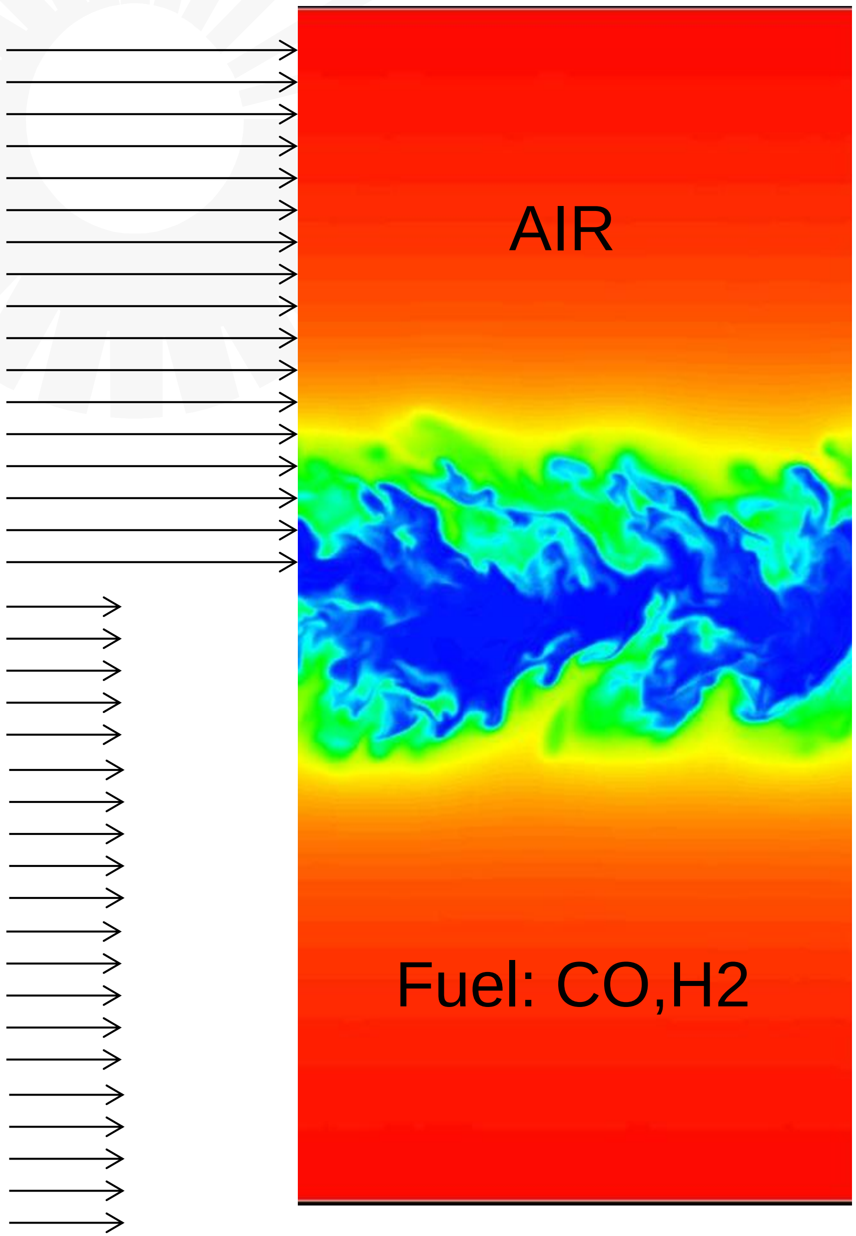
Overall acceleration depends on chem. compute load

domain decomposition → CPU transport, GPU chemistry



One GPU keeps up with all 16 CPU cores?

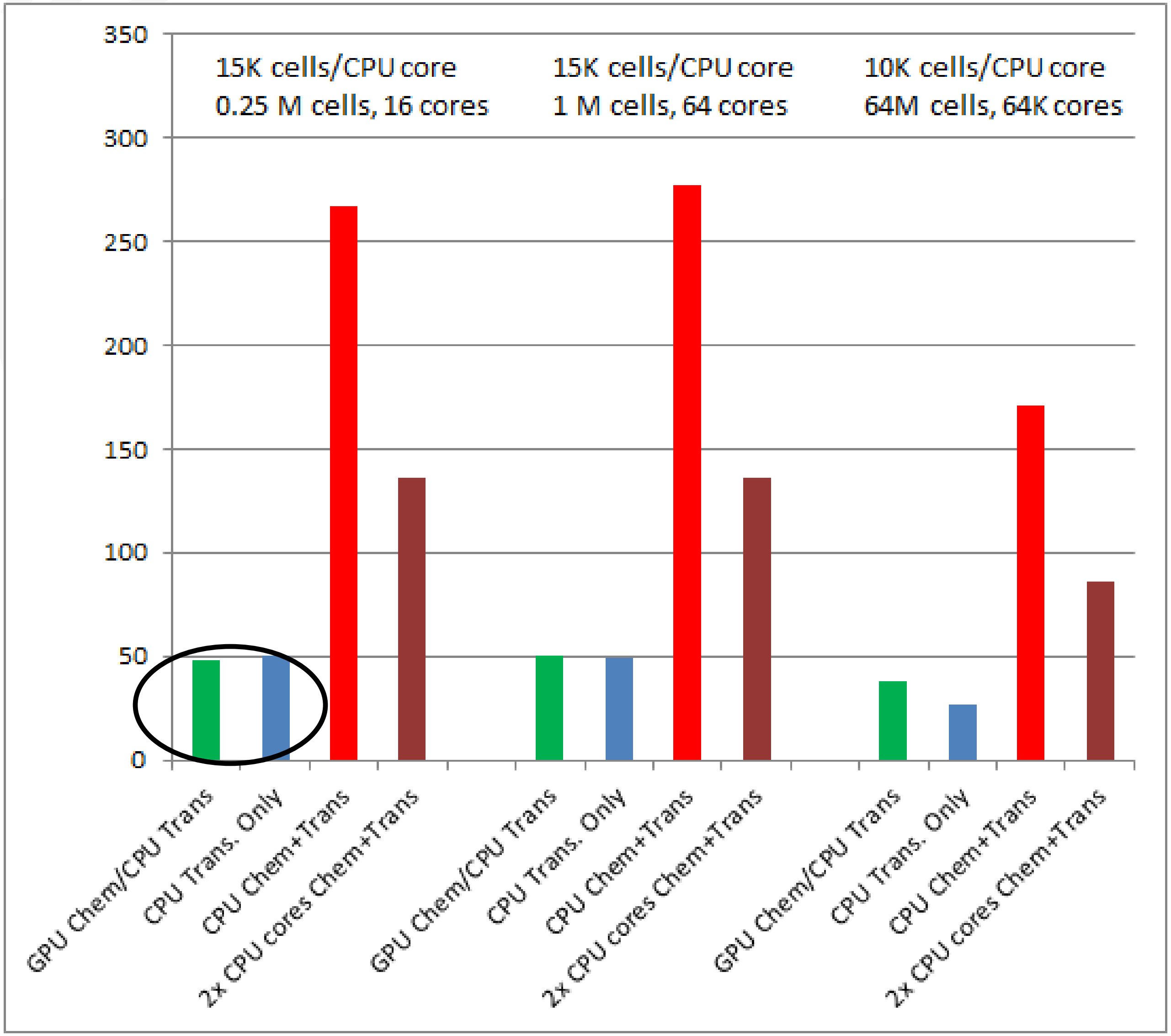
GPU-CPU hybrid code tested on shear layer turbulent flame



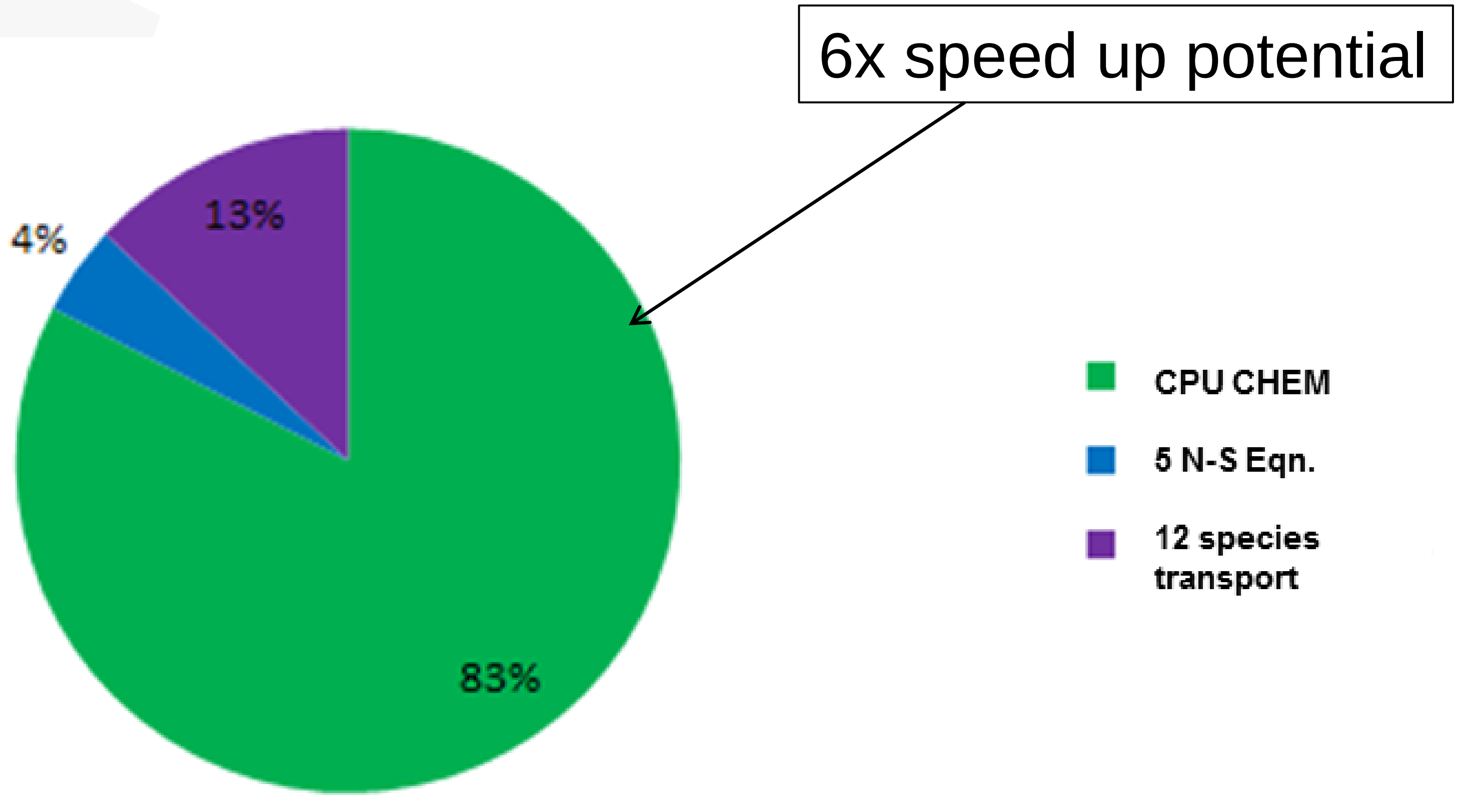
3D Direct Numerical Simulation

1. fully compressible reactive code
2. detailed chemical kinetics
3. detailed multicomponent transport
4. three dimension
5. all scales fully resolved

GPU chemistry hidden completely

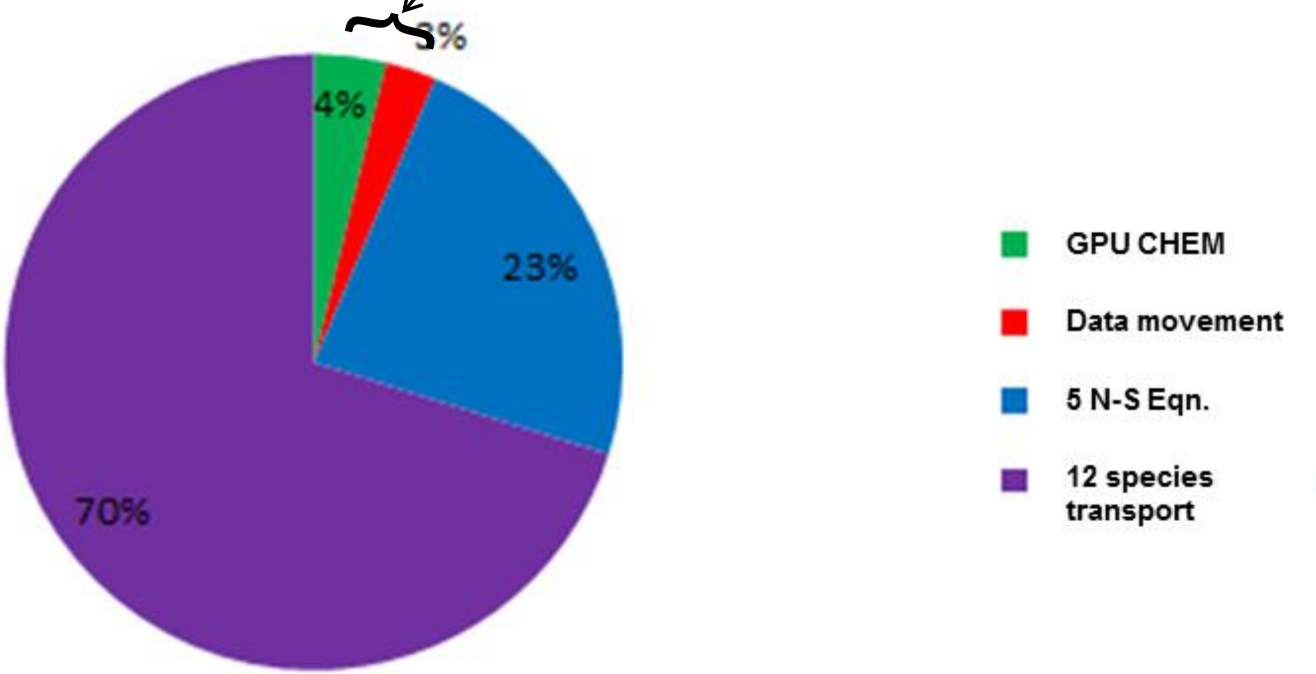


CPU only runs → chemistry takes a lion's share (12 scalars)

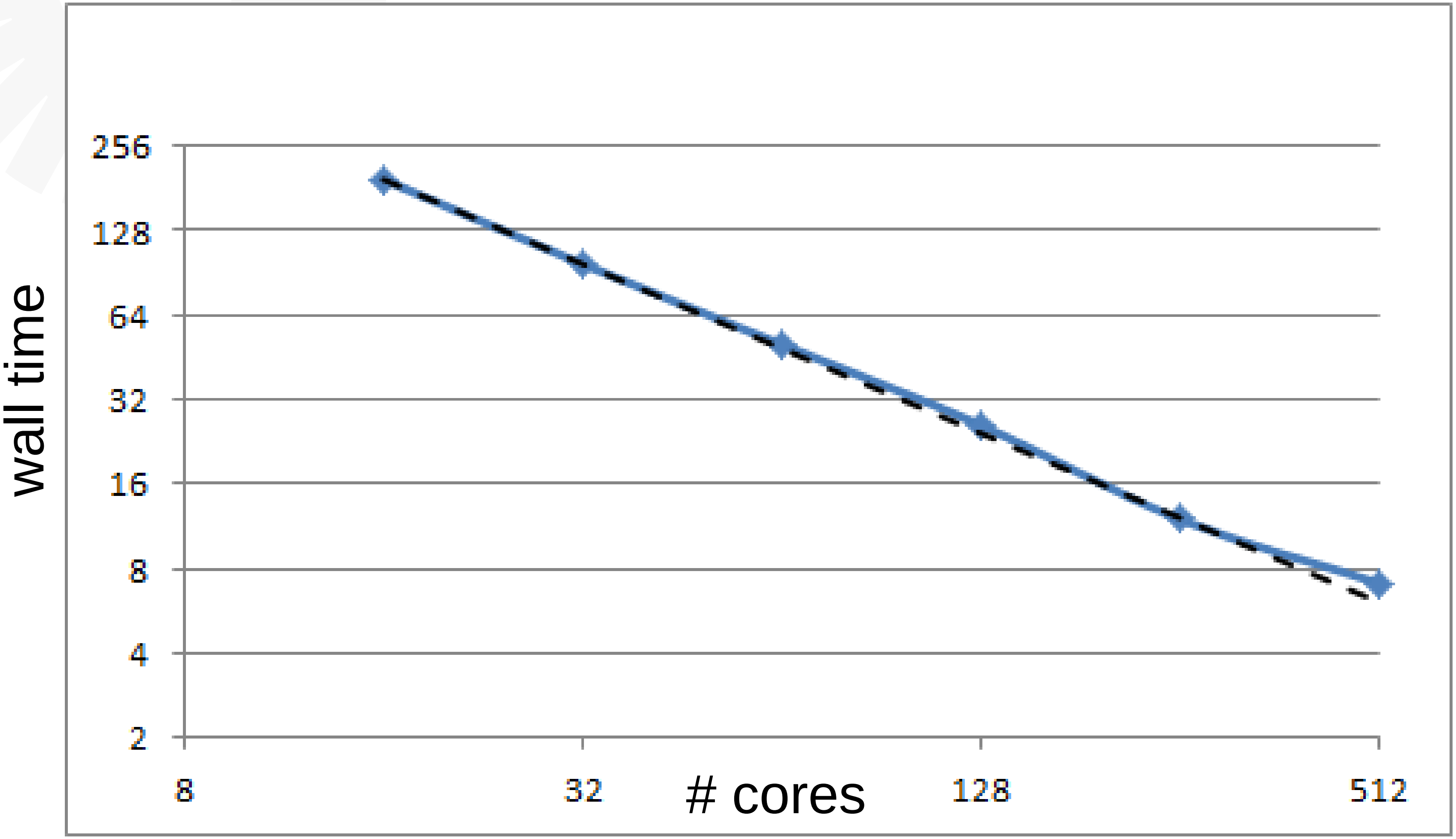


GPU: 83% to 7% reduction in chemistry compute load

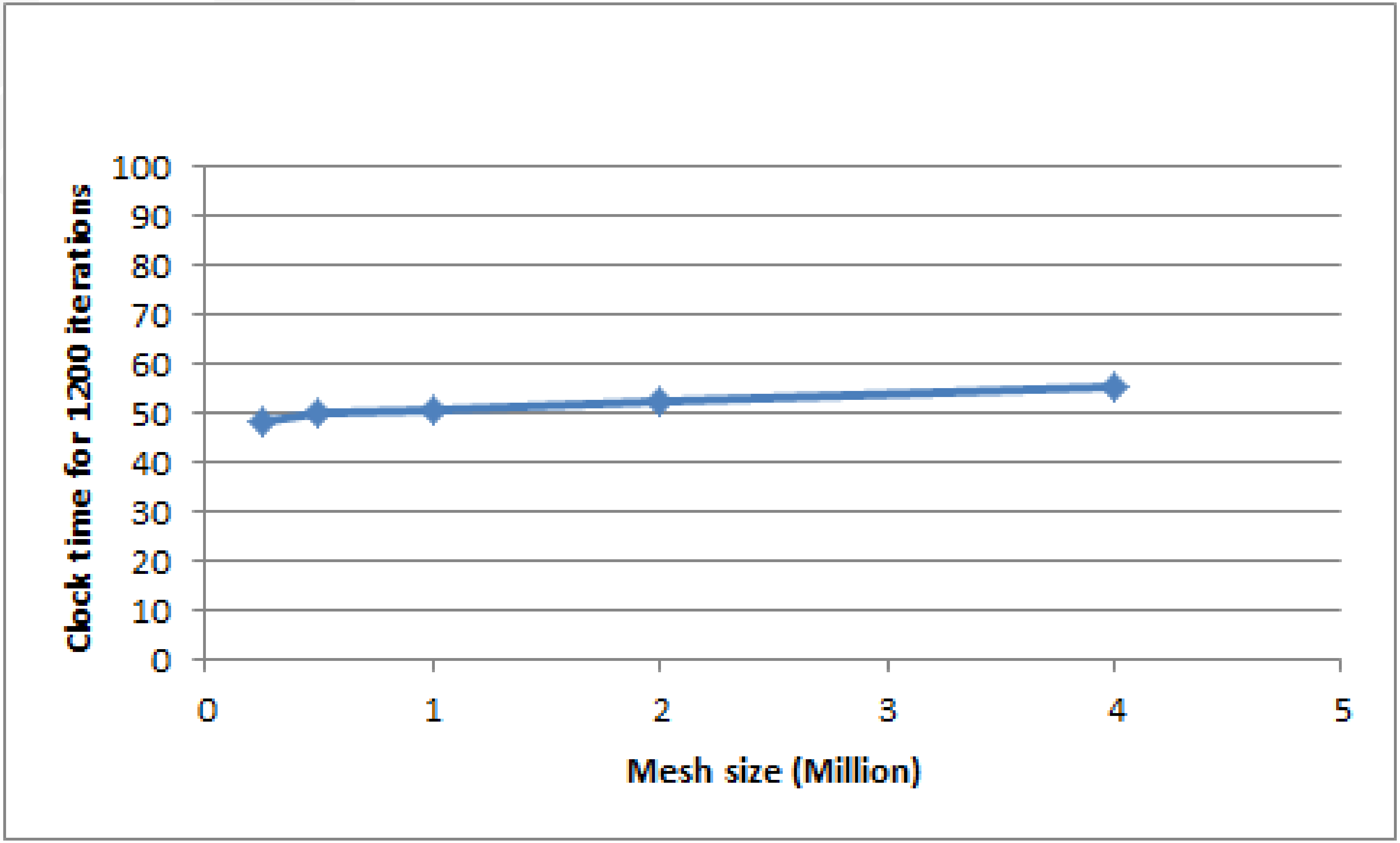
Capacity for bigger chemistry
35-40 species → good for jet fuel!



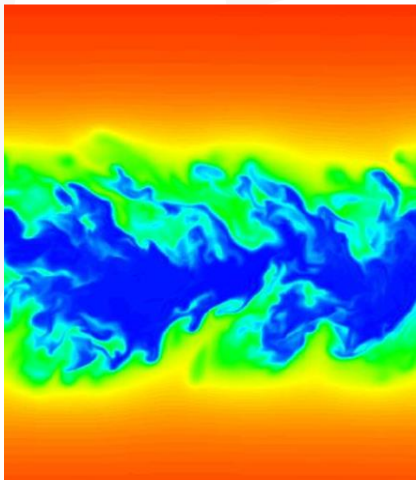
Strong scalability of CPU-GPU hybrid



Weak scalability of CPU-GPU hybrid good up to 64M cells

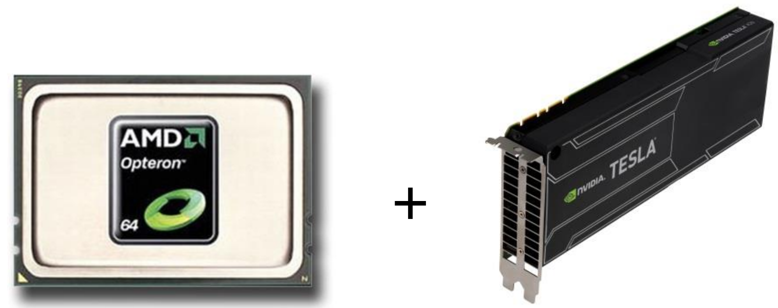


Looking forward: GPU-CPU hybrid is a significant tech. enabler



doable with

~100 nodes



Tune existing or create new turbulent-chemistry model

http://www.happynews.com/news/11142008/visualizing-unseen-forces-turbulence.htm

Courtesy of Stanford University Center for Turbulence Research