

# The New New Library

*Georgia Tech*  
→ Richard Vuduc

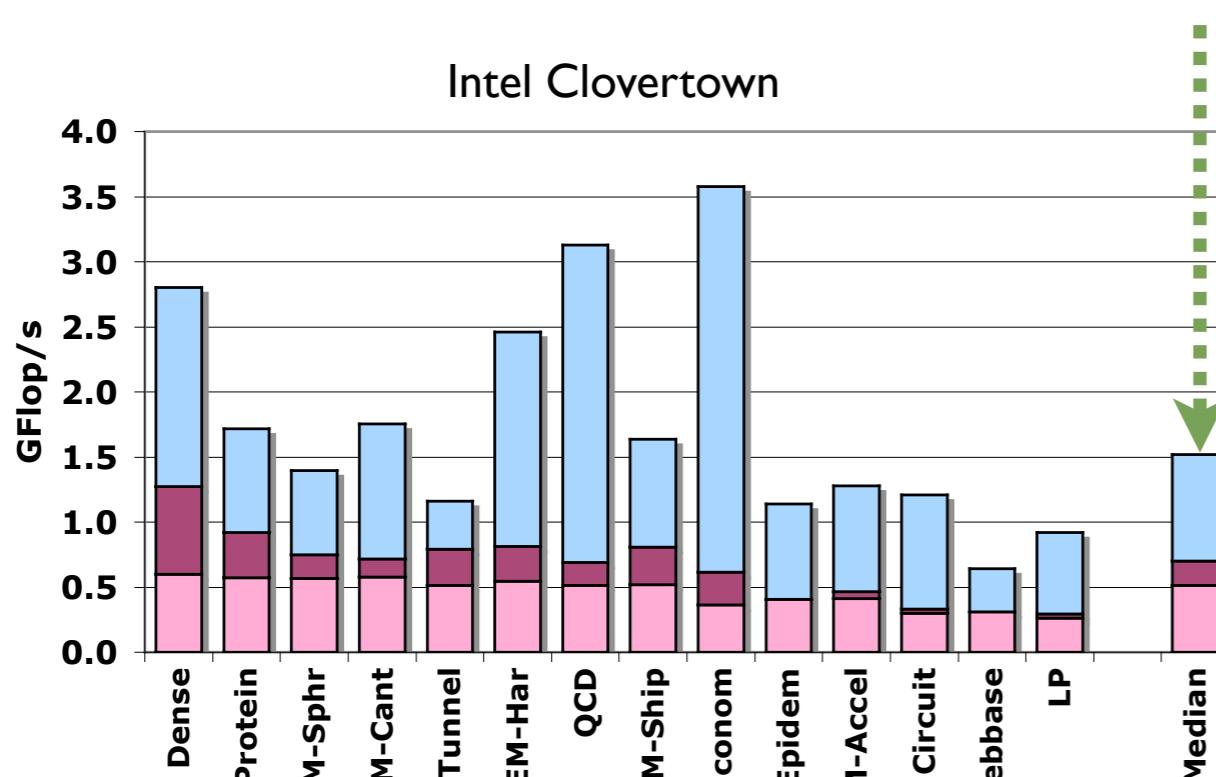
Sandia CSRI Panel on Libraries – June 4, 2008

# Proverb

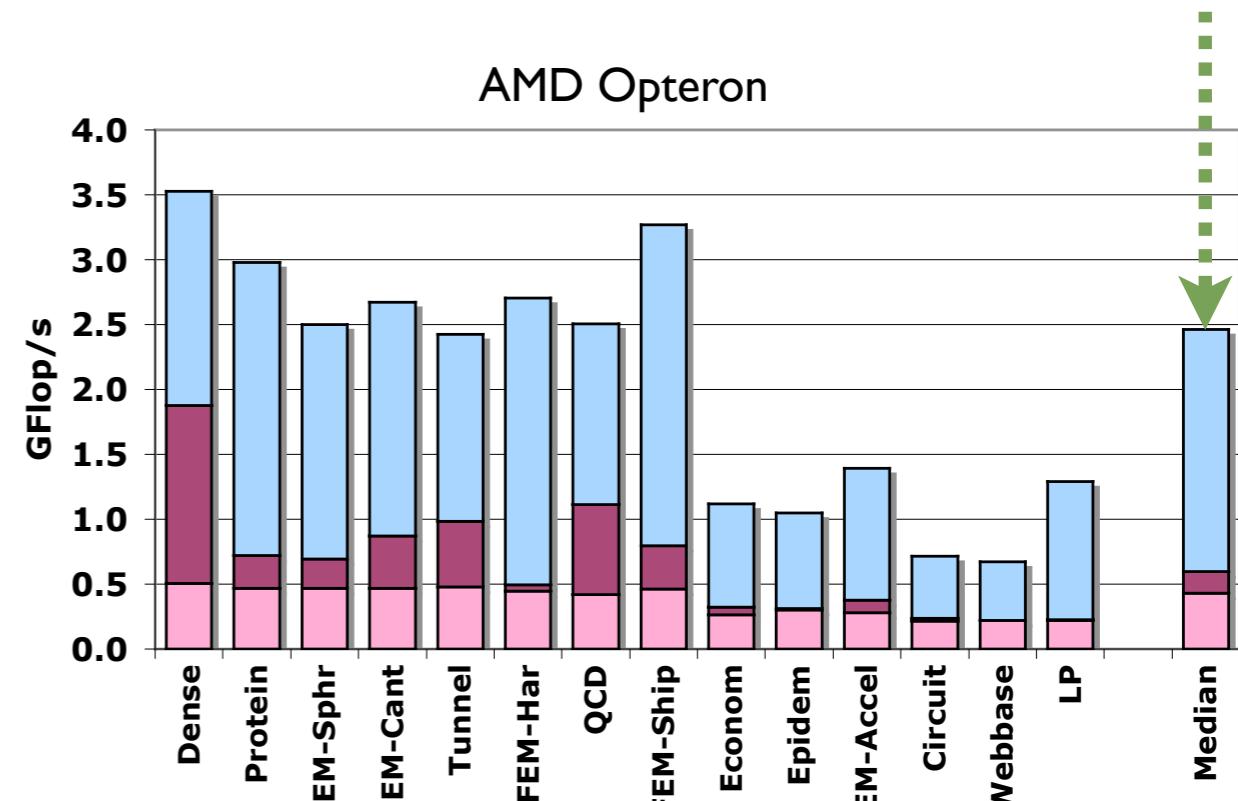
**A movement  
begins as a vision,  
runs as a business,  
and  
ends as a racket.**

Question: In what stage is MPI?

**~2x on 2x4 cores**



AMD Opteron



- Naïve Single Thread
- MPI(autotuned)
- Pthreads(autotuned)

Multicore-specific  
vs. “Off-the-shelf”

Williams, Oliker, Vuduc, Shalf, Demmel, Yelick

Sparse matrix-vector multiply (SpMV)

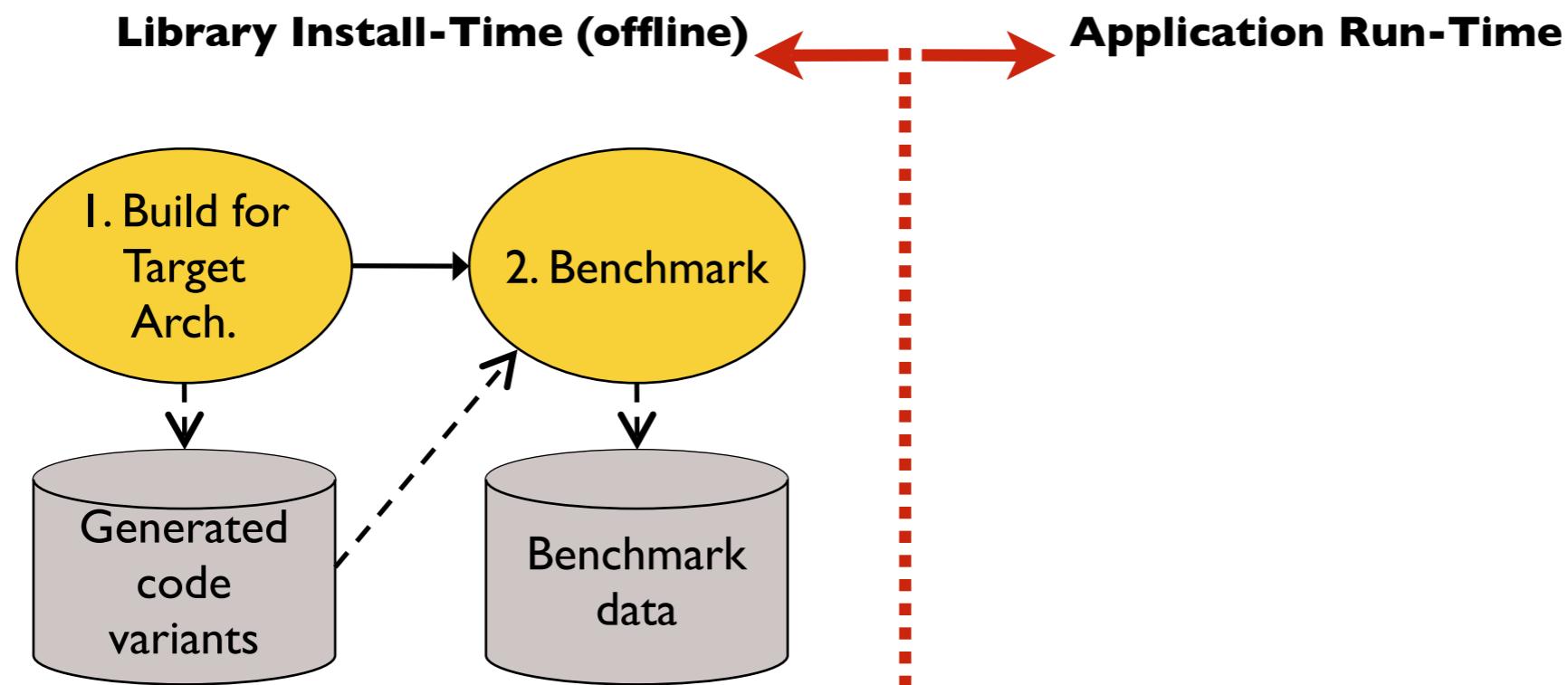
PETSc 2.3.0+MPICH 1.2.7+OSKI 1.0.1h

# How OSKI tunes

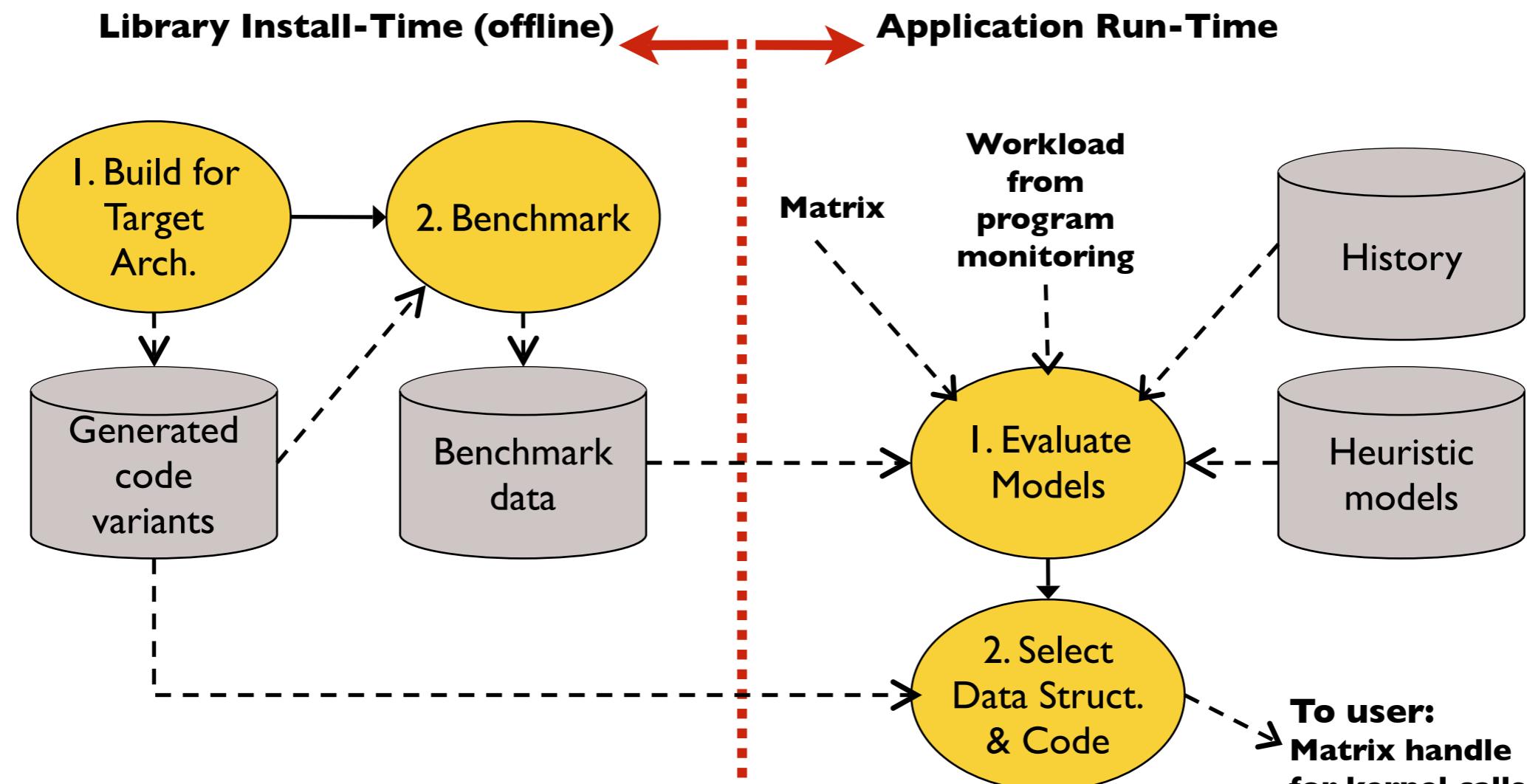
**Library Install-Time (offline)** ← → **Application Run-Time**



# How OSKI tunes



# How OSKI tunes



# The New New Library

- ▶ Write library generators, not libraries
  - ▶ Retarget / autotune, possibly at run-time
  - ▶ Specialize to application
- ▶ Blur boundaries between...
  - ▶ Algorithms
  - ▶ Languages & programming models
  - ▶ Compilers
  - ▶ Operating systems

# A library is an instance of an algorithm

- ▶ What do these have in common?

$$\forall q \in Q : \quad F(q) = \sum_{r \in (Q - \{q\})} C \frac{r - q}{\|r - q\|^3}$$

*Force computation*

$$\forall q \in Q : \quad \text{AllNN}(q) = \operatorname{argmin}_{r \in R} d(q, r)$$

*All nearest neighbors*

$$\forall q \in Q : \quad \text{KDE}(q) = \frac{1}{|R|} \sum_{r \in R} K(q, r)$$

*Kernel density estimation*

- ▶ Generalized n-body algorithms (A. Gray, et al., @ Georgia Tech)

$$\Psi(Q, R) = \bigotimes_Q \bigoplus_R f(Q, R)$$

*MapReduce-like!*

# A library is a language

- ▶ Could build library-aware compilers (e.g., LLNL/ROSE)

```
DataType A;  
// ... operations on A ...  
A.exchange();  
  
// ...  
  
DataType B;  
// ... operations on B ...  
B.exchange();
```

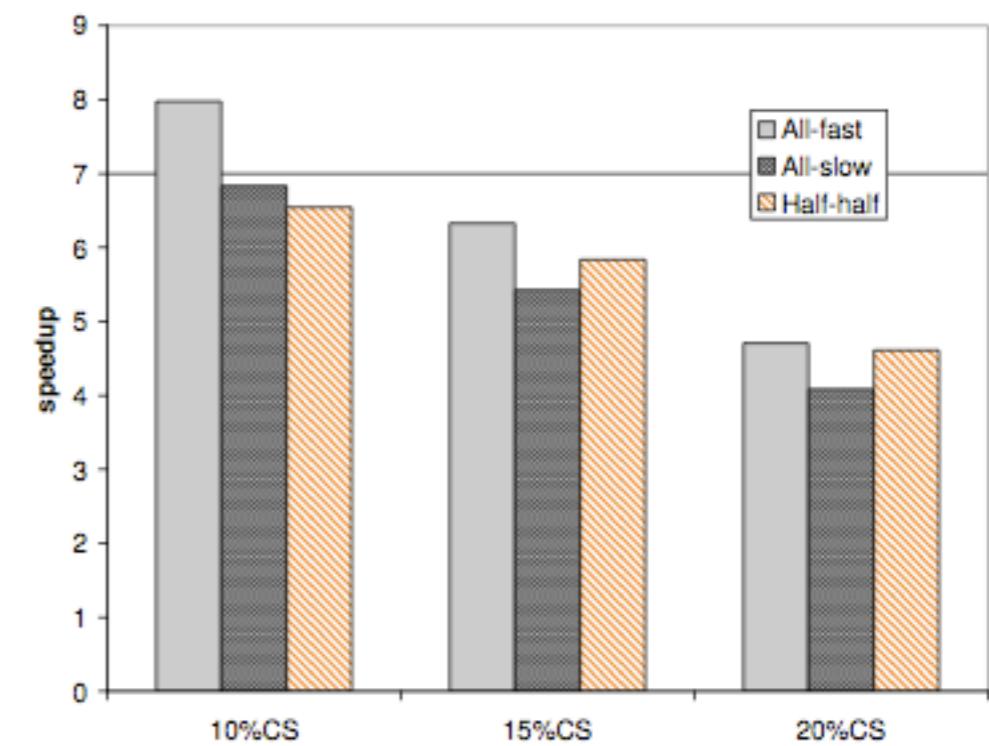
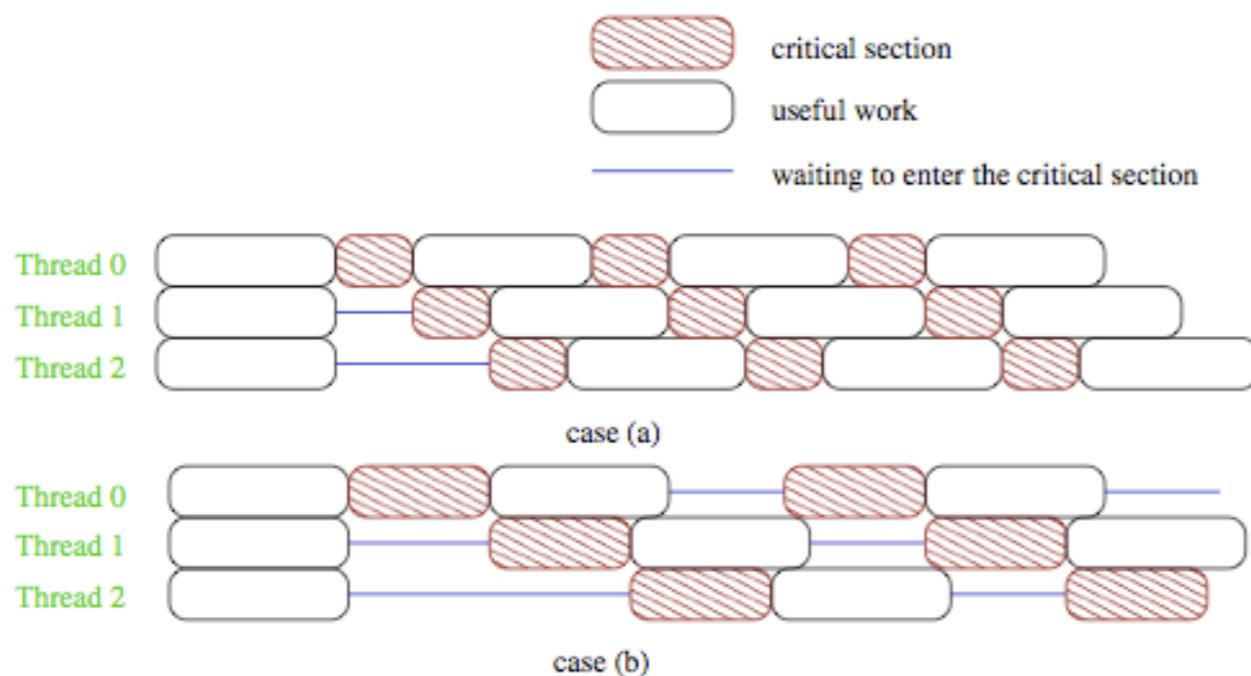


```
DataType A;  
// ... operations on A ...  
// ...  
DataType B;  
// ... operations on B ...  
bulkExchange(A, B);
```

- ▶ 10x improvement by hand

# A library informs scheduling

- ▶ Example: Critical section-aware scheduling  
(Hyesoon Kim, et al., Georgia Tech)



# Wildly asynchronous algorithms

- ▶ Chaotic relaxation – Chazan & Miranker ('69)
- ▶ Discrete-event approach to numerical algorithms
  - ▶ Nutaro, et al. (ORNL)
  - ▶ AVI – Lew, Marsden, et al.
  - ▶ Karimabadi, Omelchenko (SciberQuest)

- ▶ Trends?
  - ▶ Data structure-centric
  - ▶ DAG-based
  - ▶ Wildly asynchronous
- ▶ Library issues
  - ▶ Sufficiently express key abstractions?
  - ▶ Generate “libraries?”
  - ▶ Enrich scheduling interfaces?
  - ▶ Expose tuning knobs?