Argonne NATIONAL LABORATORY

... for a brighter future

From the Heroic to the Logistical

Programming Model Implications of New Supercomputing Applications

Ian Foster

Computation Institute
Argonne National Laboratory &
The University of Chicago

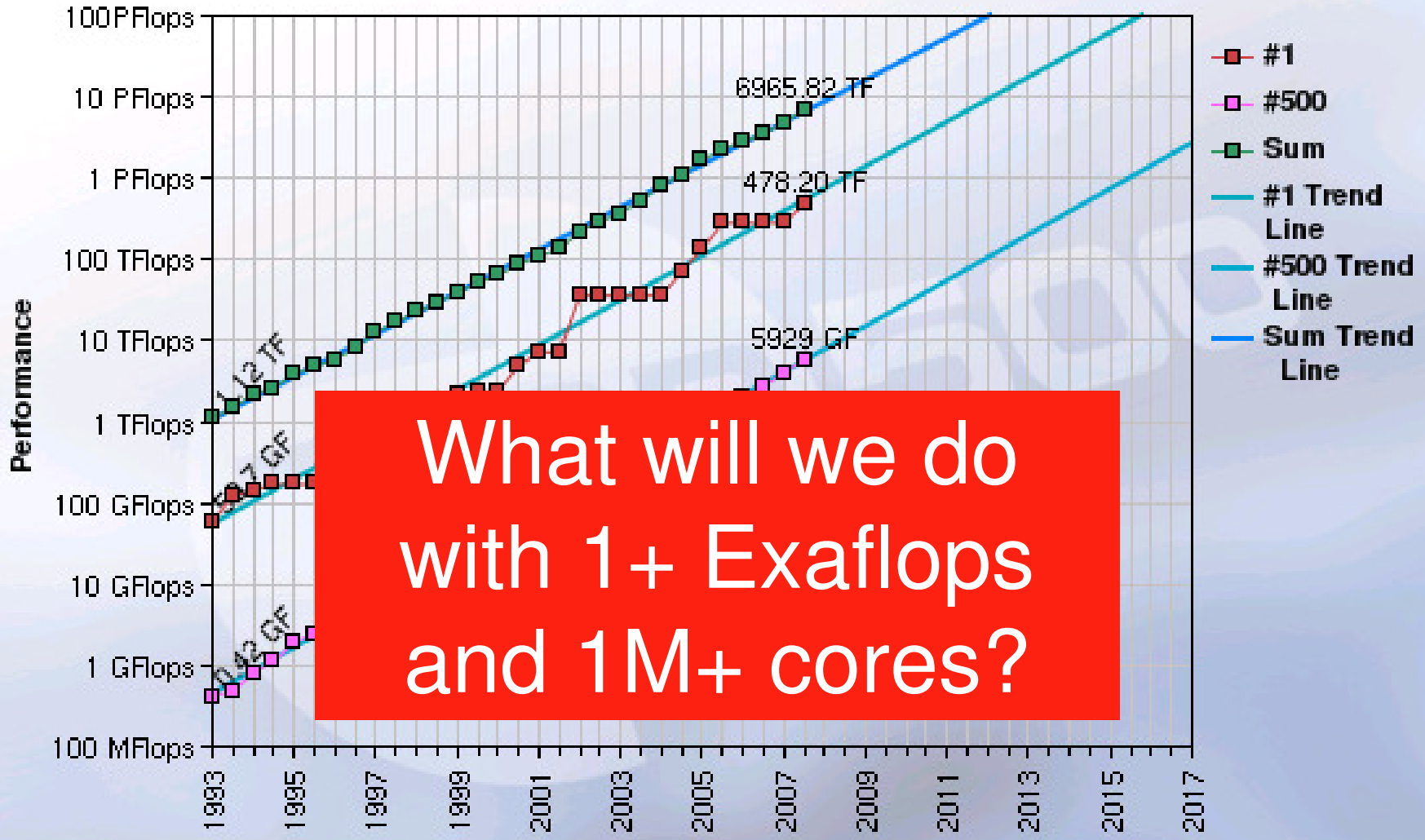
With thanks to: **Miron Livny**, **Ioan Raicu**, **Mike Wilde**, **Yong Zhao**, and many others.

U.S. Department of Energy

UChicago Argonne LLC

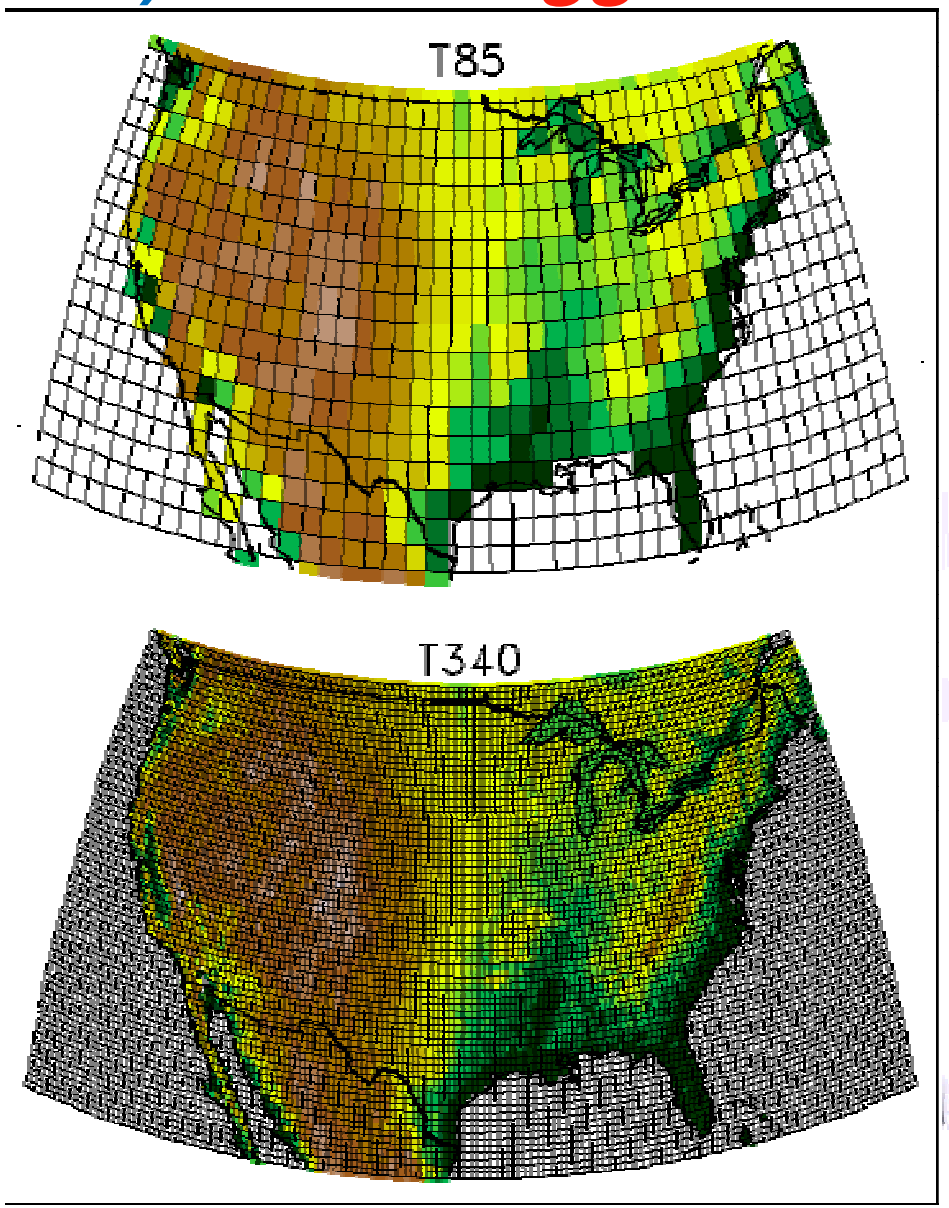
A U.S. Department of Energy laboratory managed by UChicago Argonne, LLC

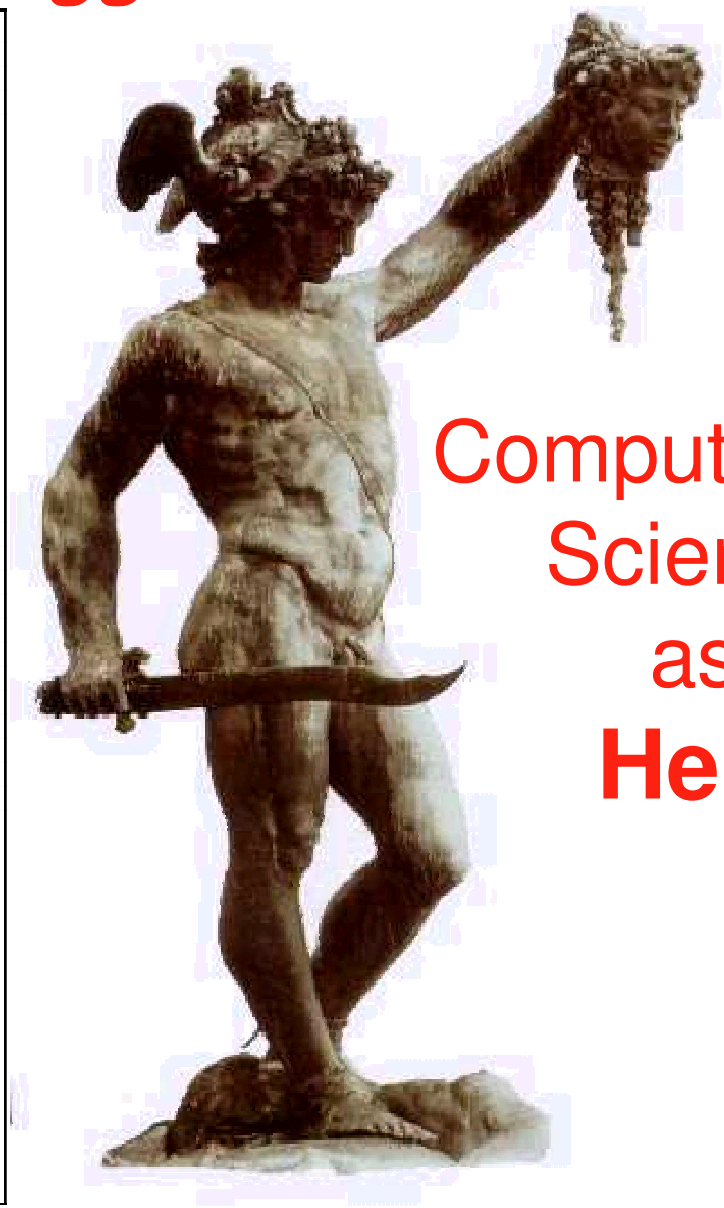
Projected Performance Development



What will we do with 1+ Exaflops and 1M+ cores?

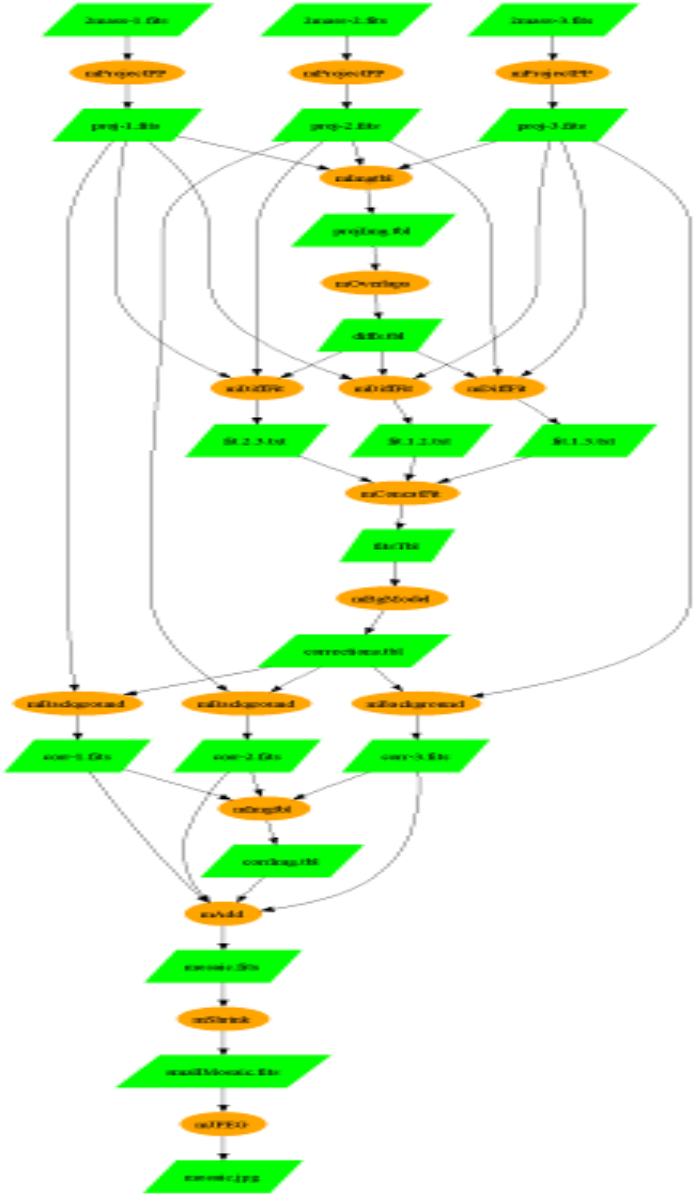
1) Tackle Bigger and Bigger Problems





Computational Scientist as **Hero**

2) Tackle Increasingly Complex Problems



Computational Scientist as **Logistics Officer**

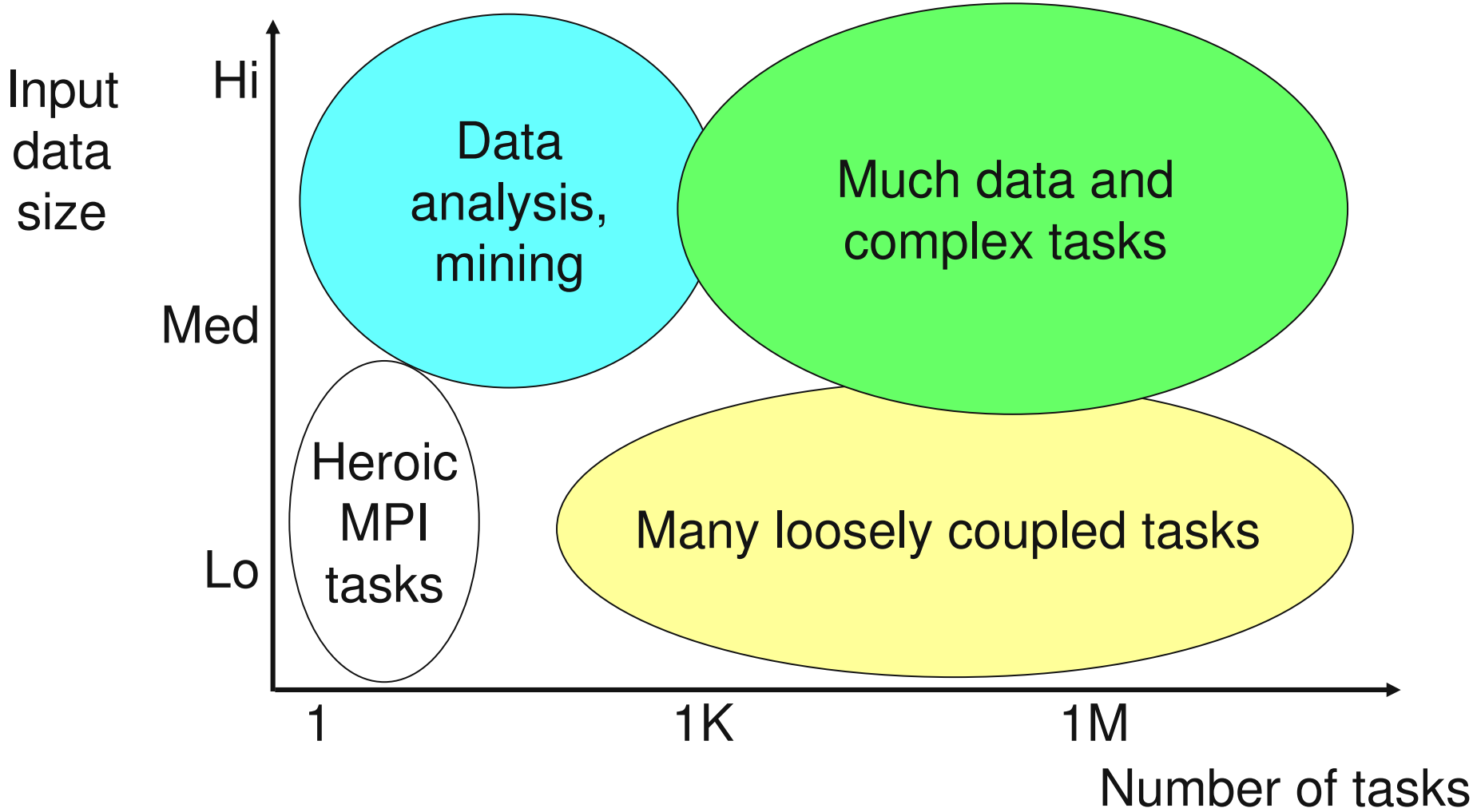
"More Complex Problems"

- Use ensemble runs to quantify **climate model uncertainty**
- Identify **potential drug targets** by screening a database of ligand structures against target proteins
- Study **economic model sensitivity** to key parameters
- Analyze **turbulence dataset** from multiple perspectives
- Perform **numerical optimization** to determine optimal resource assignment in energy problems
- Mine collection of data from **advanced light sources**
- Construct databases of computed properties of **chemical compounds**
- Analyze data from the **Large Hadron Collider**
- Analyze **log data** from 100,000-node parallel computations

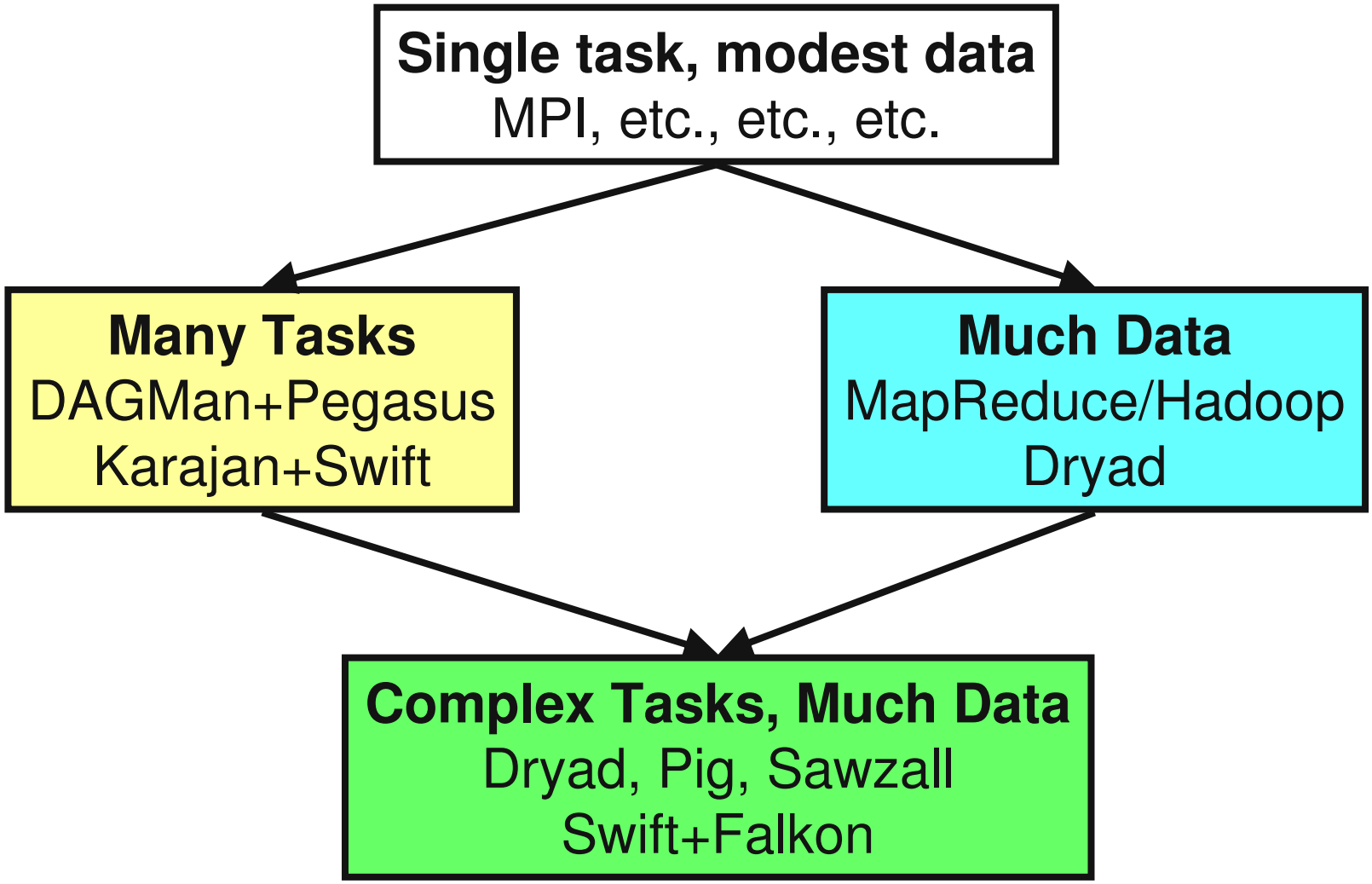
Programming Model Issues

- Massive **task parallelism**
- Massive **data parallelism**
- Integrating **black box applications**
- Complex **task dependencies** (task graphs)
- **Failure**, and other execution management issues
- **Data management**: input, intermediate, output
- **Dynamic task graphs**
- **Dynamic data access** involving large amounts of data
- **Long-running** computations
- Documenting **provenance** of data products

Problem Types

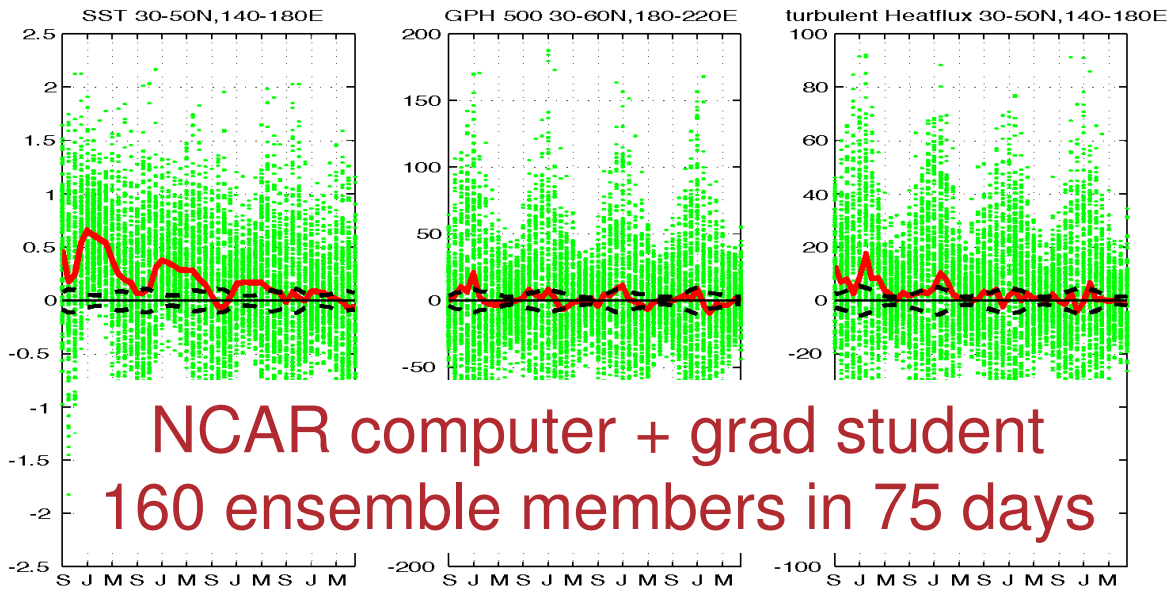


An Incomplete and Simplistic View of Programming Models and Tools



Many Tasks

Climate Ensemble Simulations (Using FOAM, 2005)



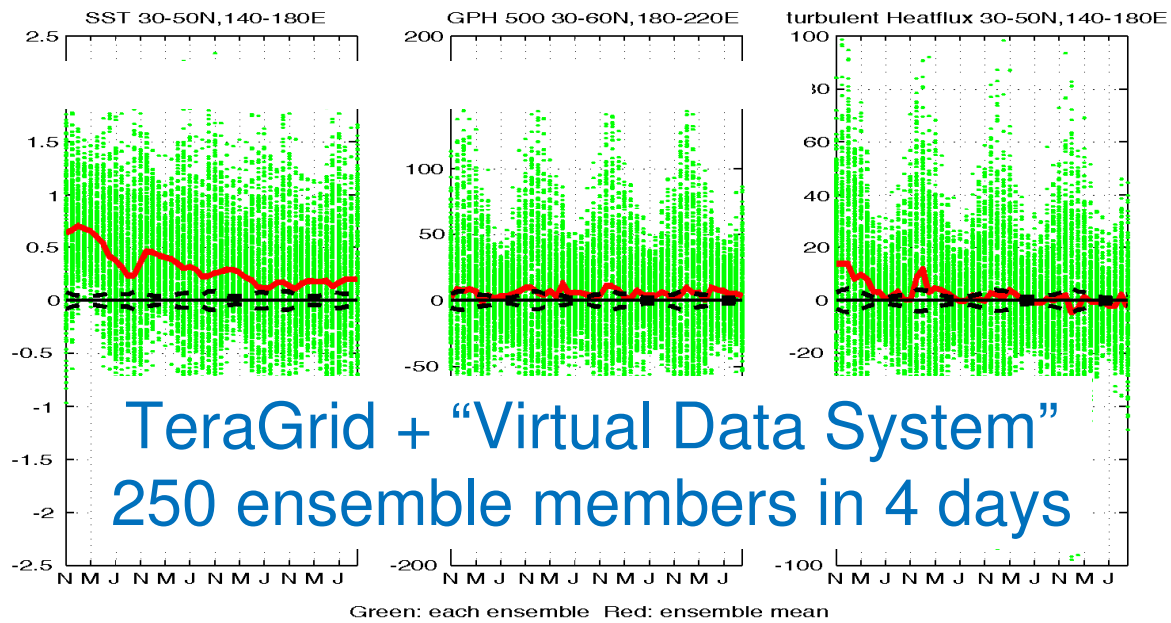
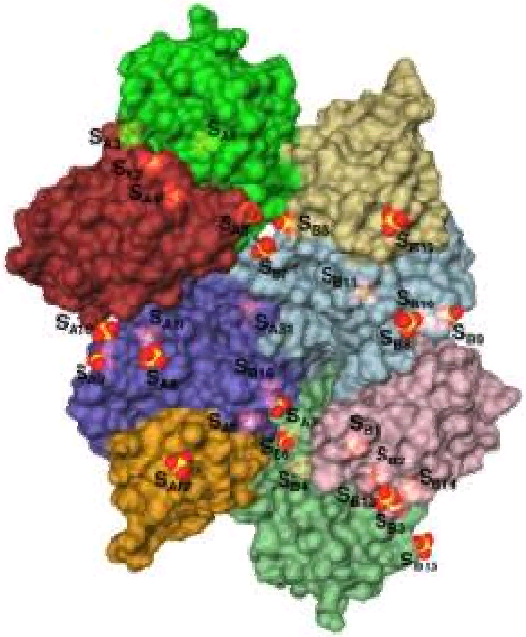
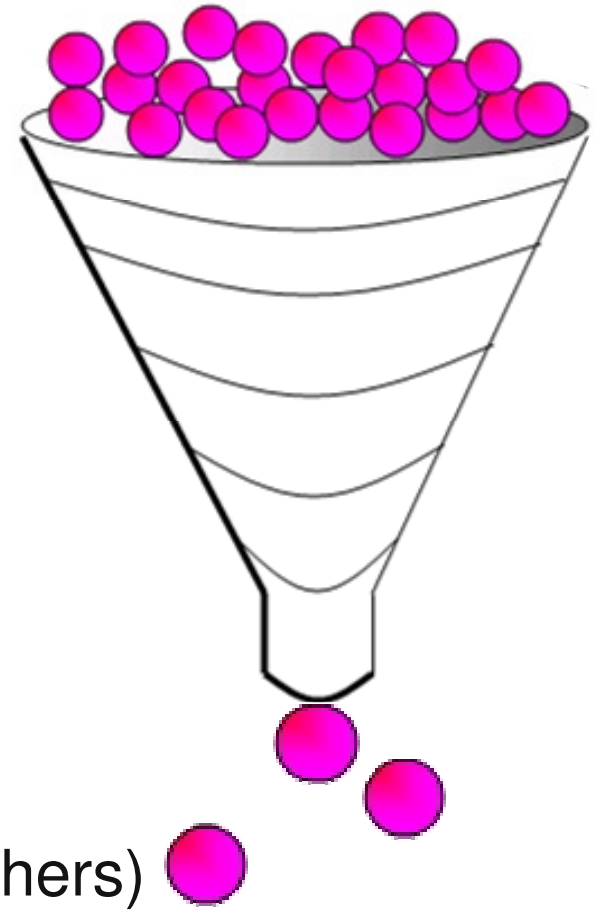


Image courtesy Pat Behling and Yun Liu, UW Madison

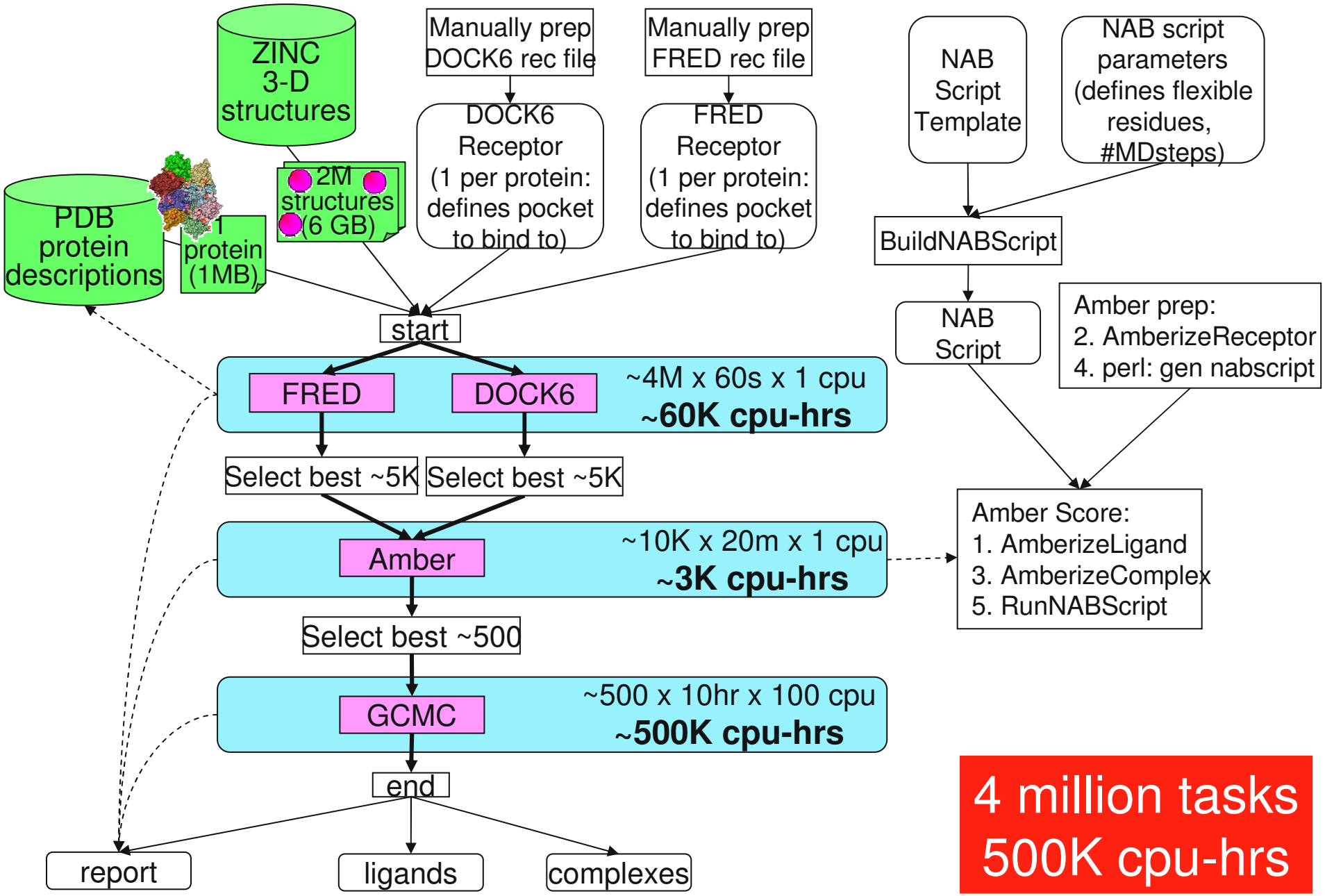
Many Many Tasks: Identifying Potential Drug Targets

Protein target(s) x 2M+ ligands





(Mike Kubal, Benoit Roux, and others)



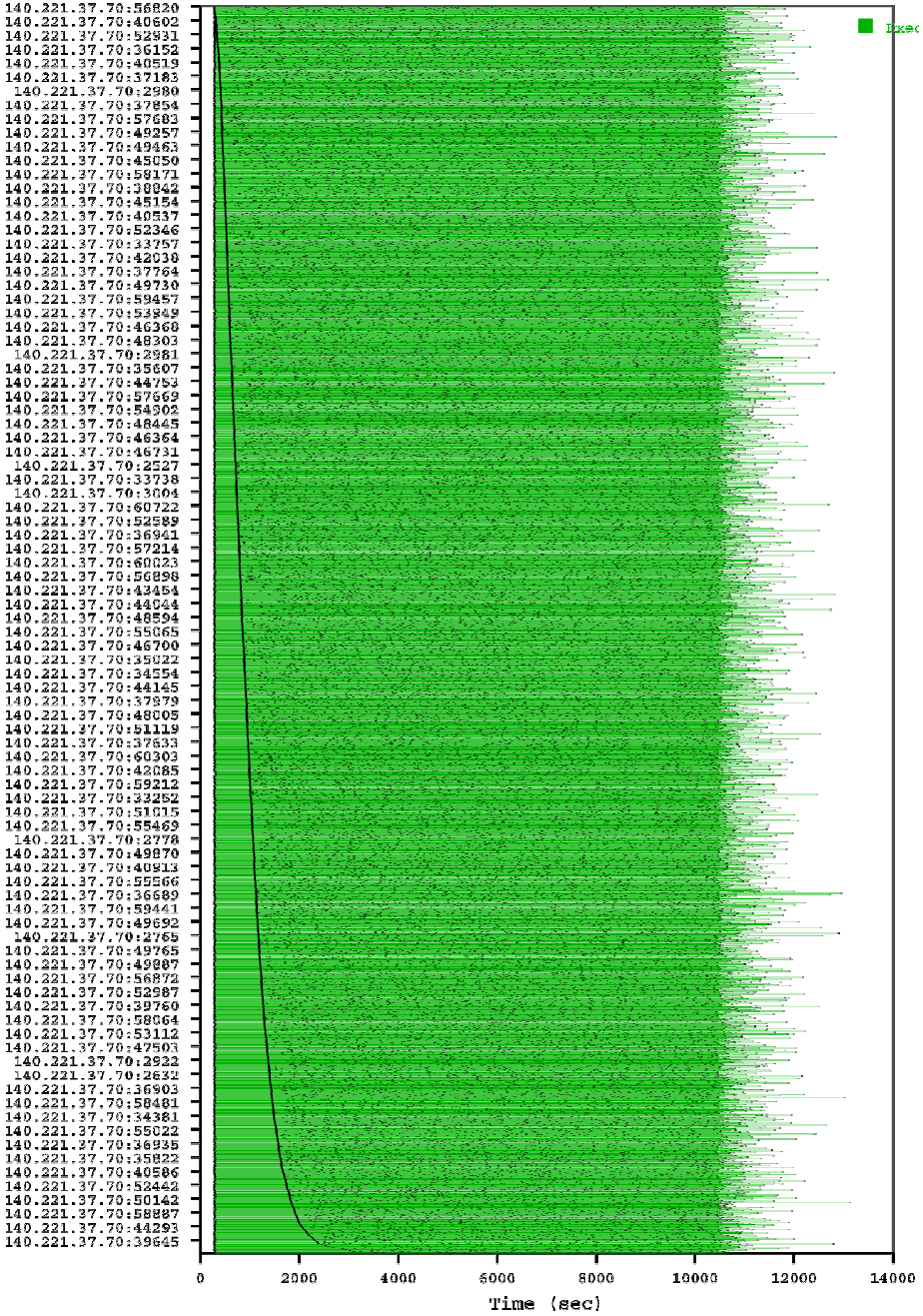
4 million tasks
500K cpu-hrs

DOCK on SiCortex

- CPU cores: 5760
- Tasks: 92160
- Elapsed time: 12821 sec
- Compute time: 1.94 CPU years
- Average task time: 660.3 sec

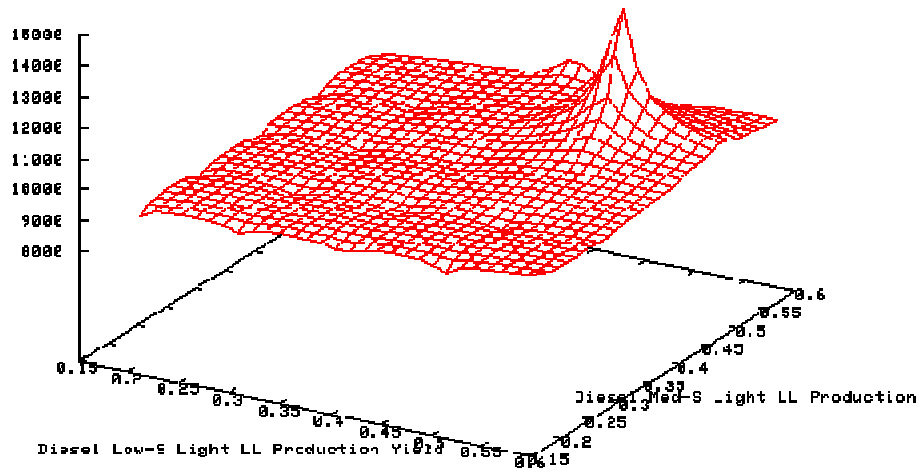
(does not include ~800 sec to stage input data)

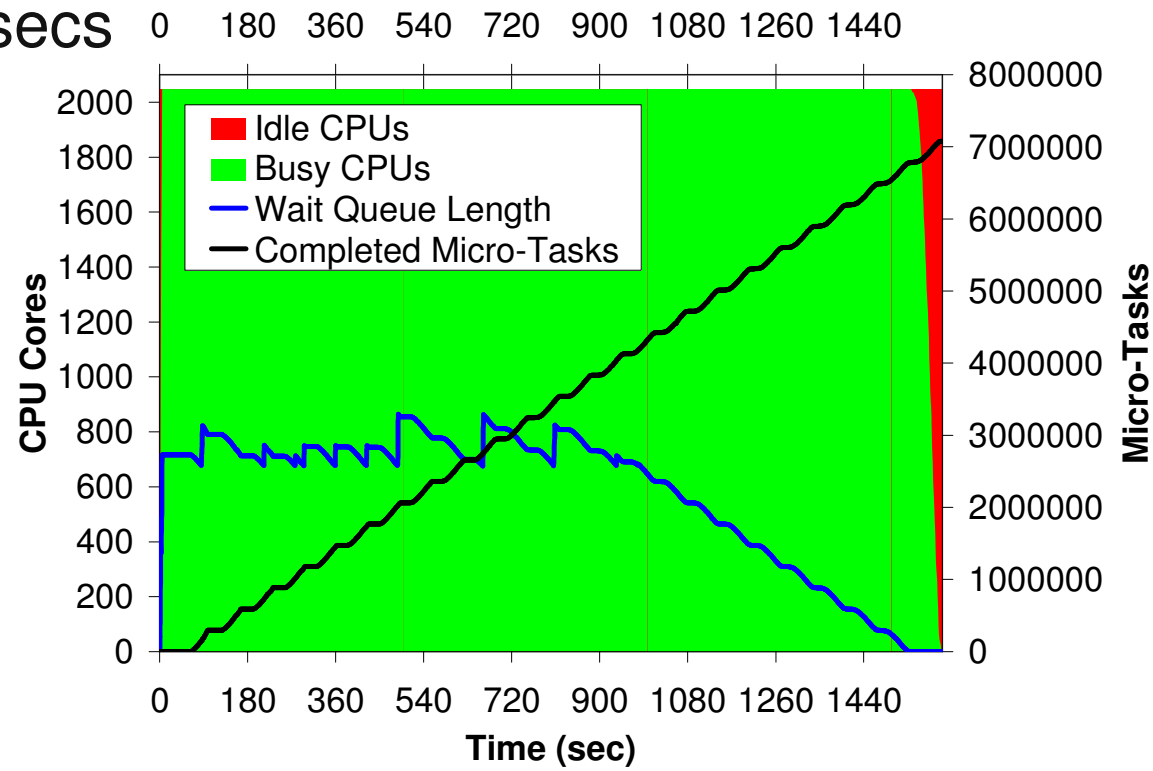
Ioan Raicu, Zhao Zhang

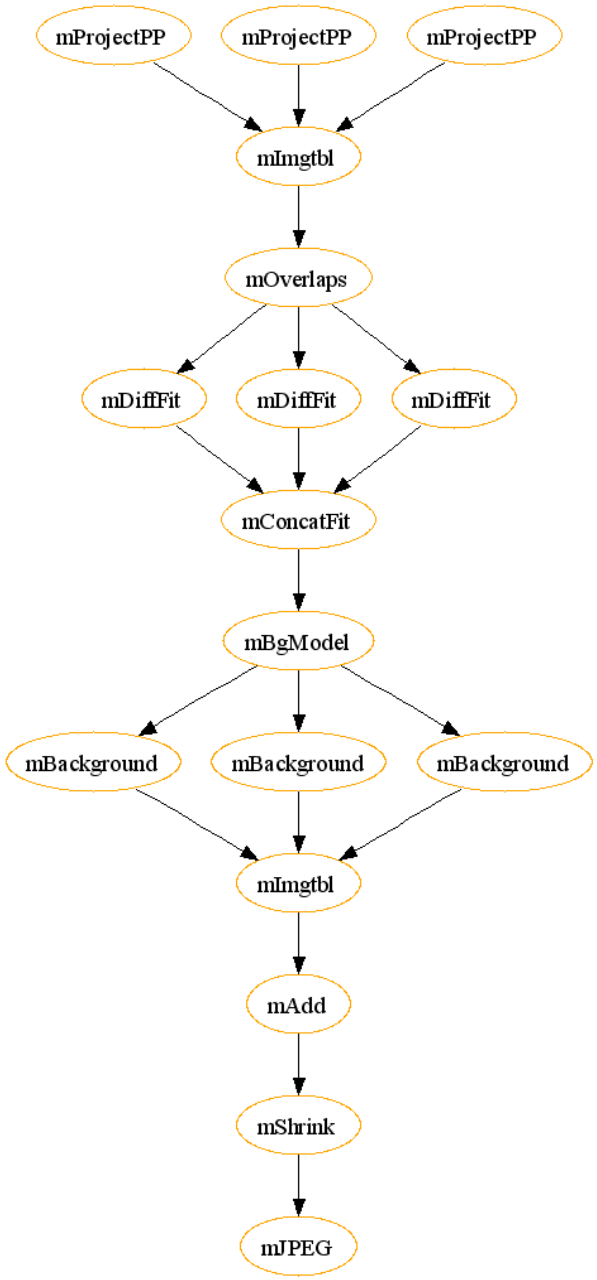


MARS Economic Model Parameter Study

- 2,048 BG/P CPU cores
- Tasks: 49,152
- Micro-tasks: 7,077,888
- Elapsed time: 1,601 secs
- CPU Hours: 894





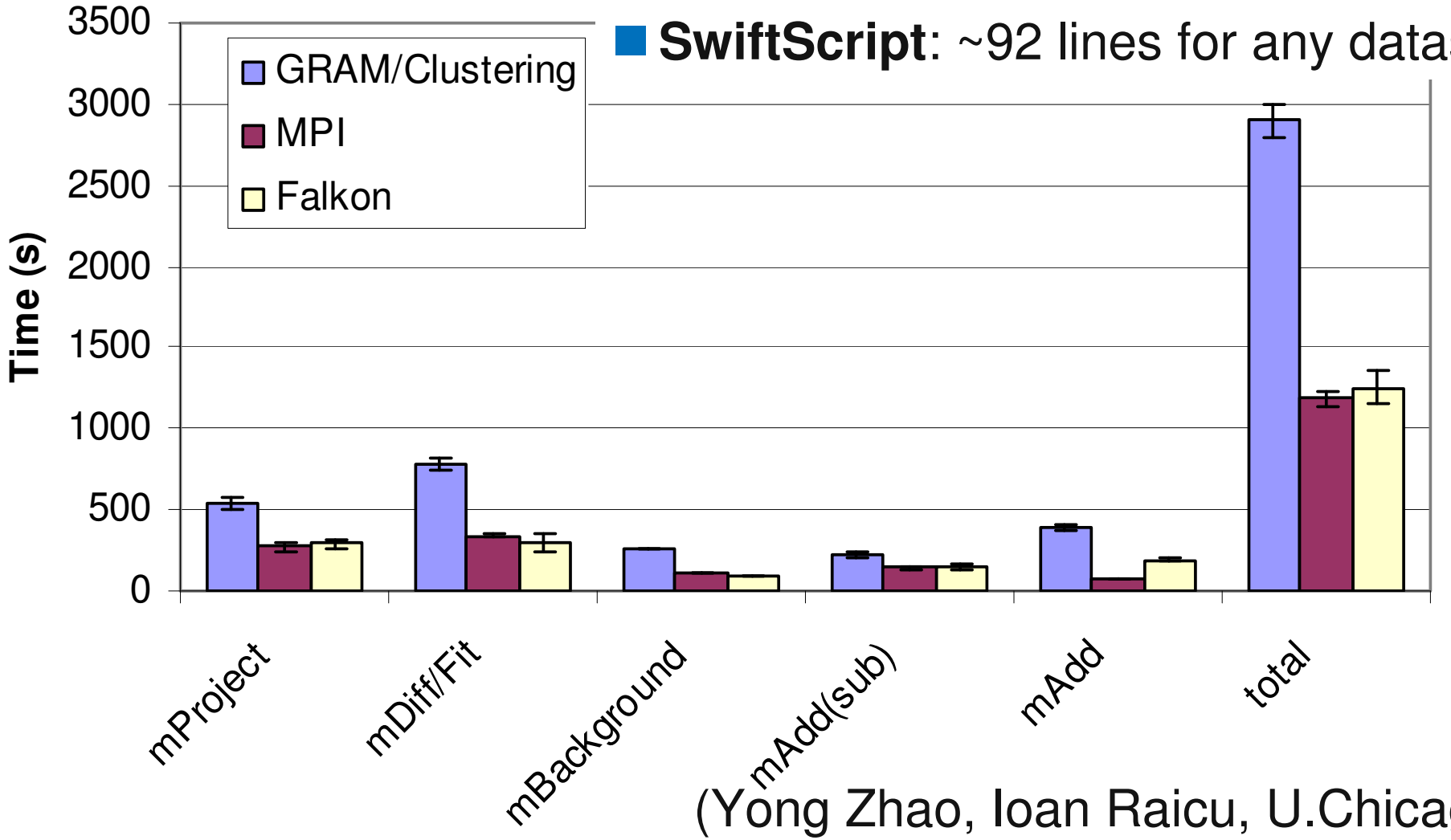


Montage

B. Berriman, J. Good (Caltech)
J. Jacob, D. Katz (JPL)

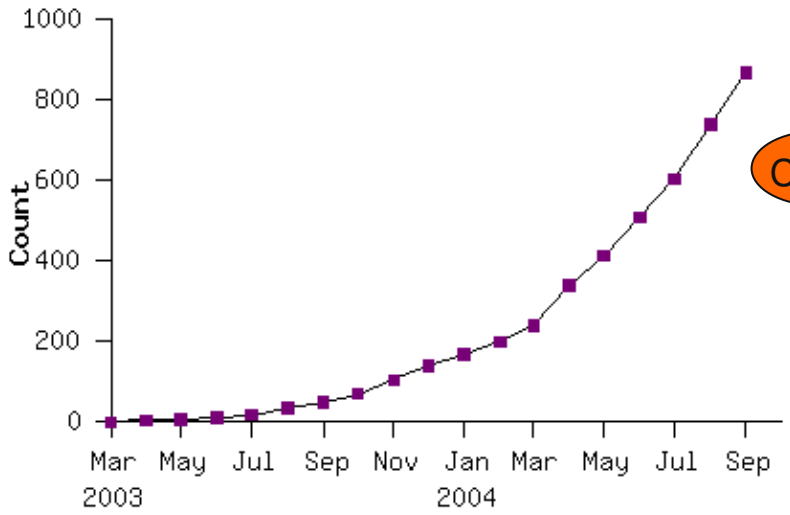
Montage in MPI and Swift

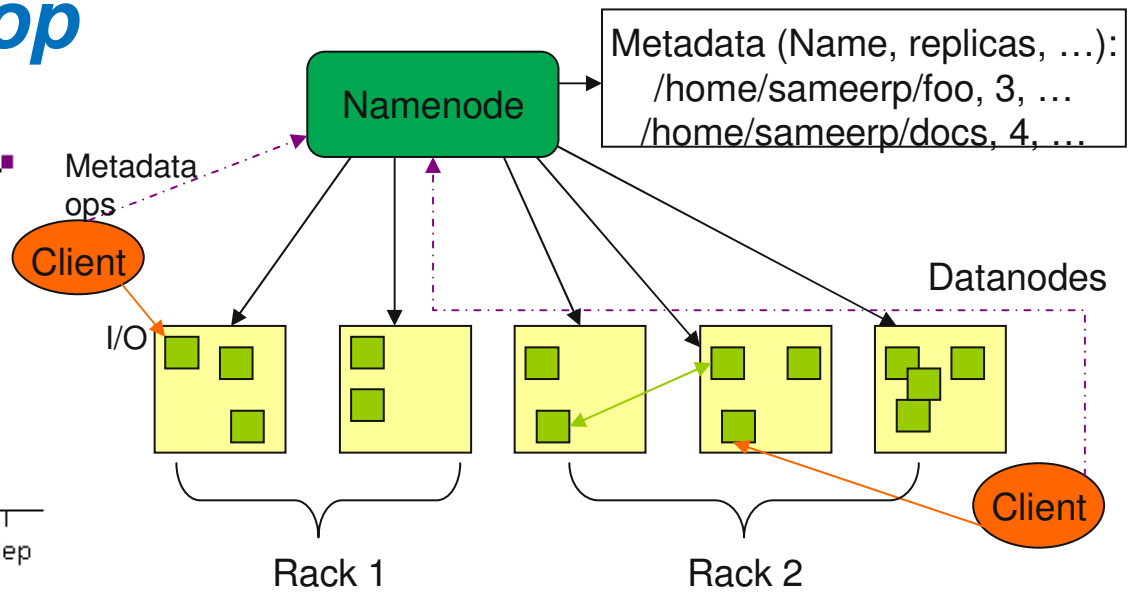
- **MPI**: ~950 lines of C for one stage
- **Pegasus**: ~1200 lines of C + tools to generate DAG for specific dataset
- **SwiftScript**: ~92 lines for any dataset



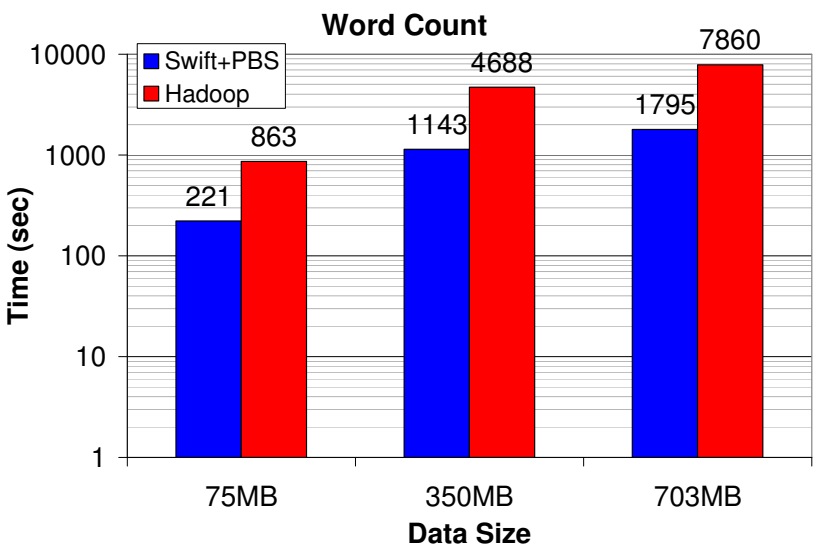
(Yong Zhao, Ioan Raicu, U.Chicago)

MapReduce/Hadoop





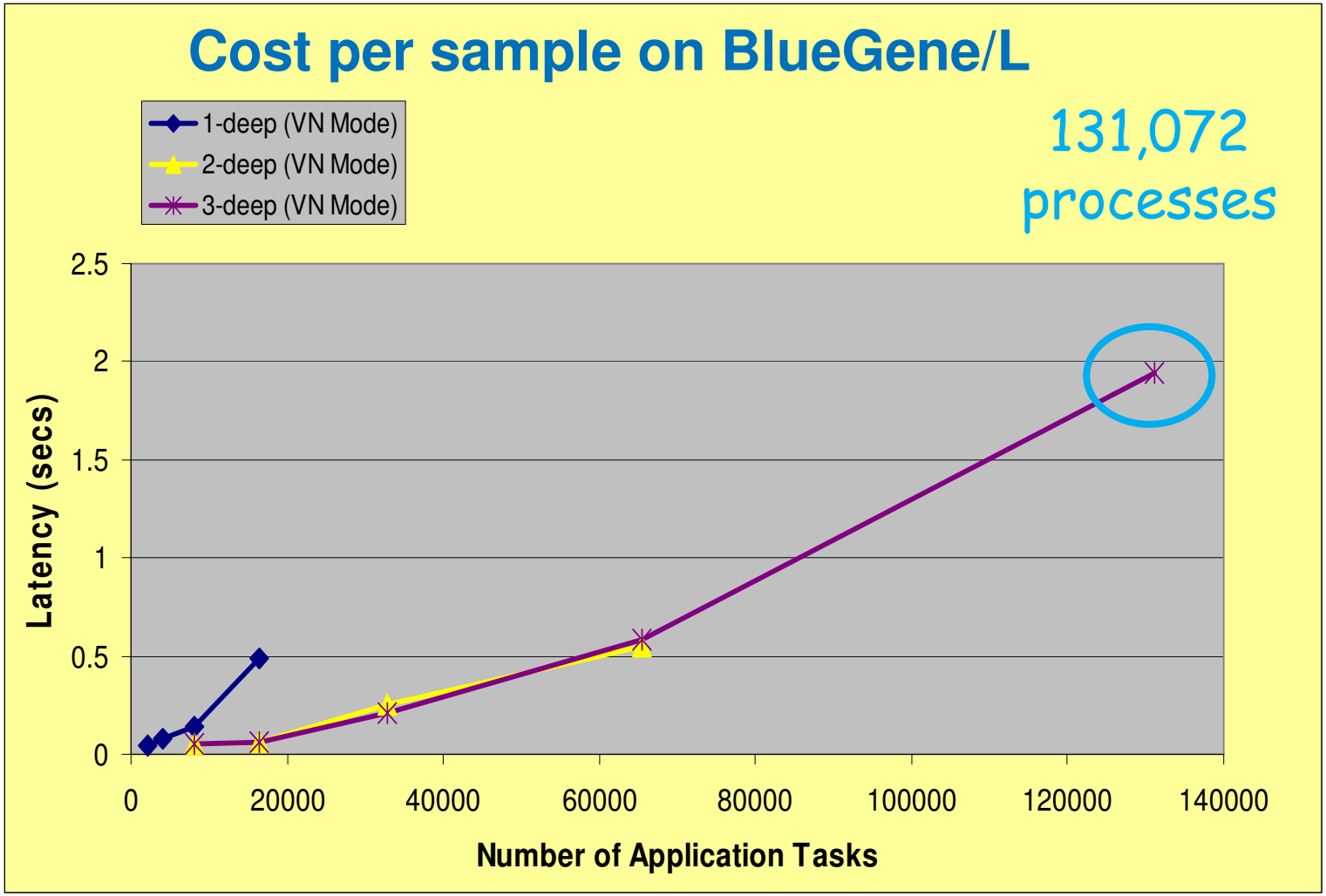
Hadoop DFS Architecture



ALCF: 80 TB memory,
8 PB disk,
78 GB/s I/O bandwidth

Soner Balkir, Jing Tie, Quan Pham

Extreme Scale Debugging: Stack Trace Sampling Tool (STAT)

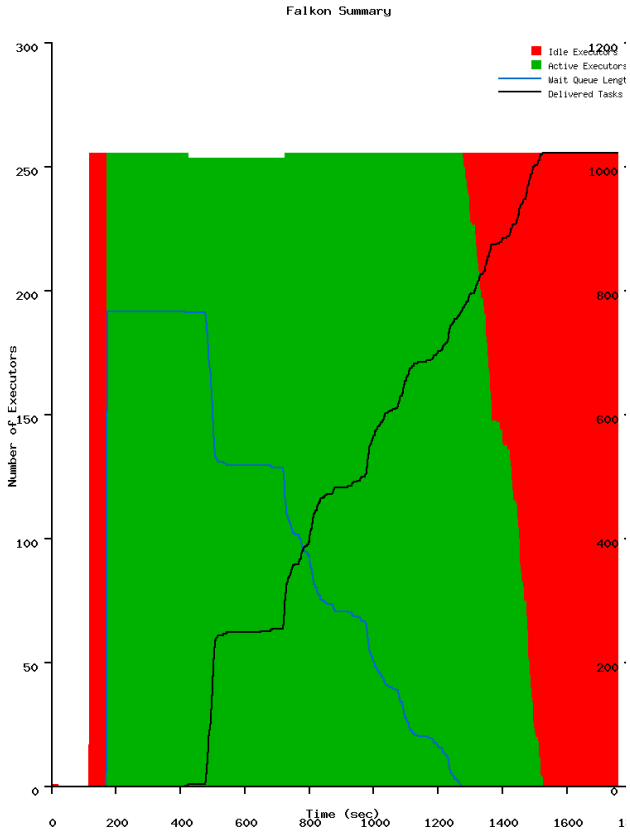
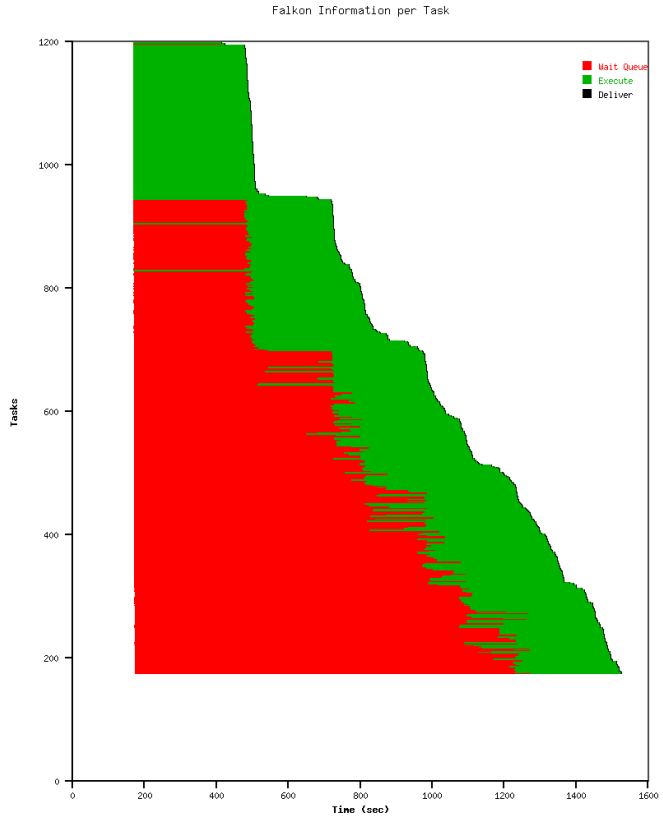
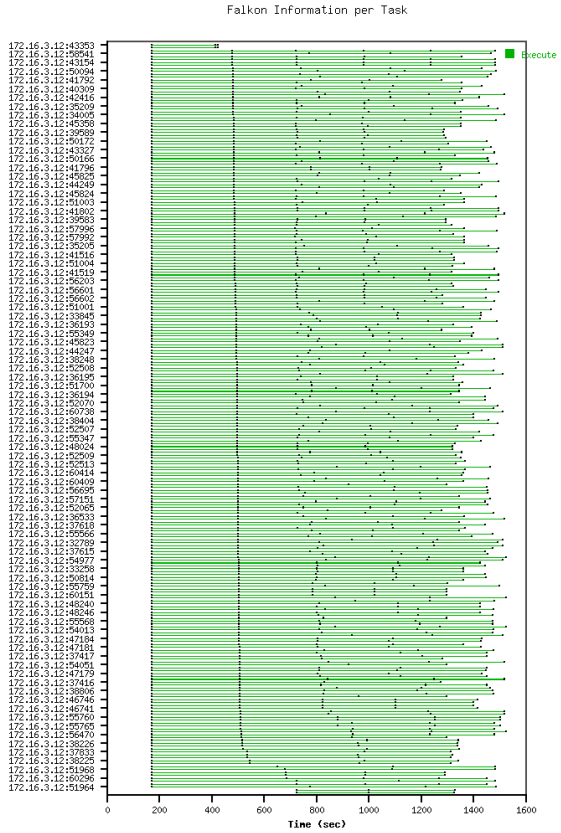


Bart Miller, Wisconsin

Summary

- Peta- and exa-scale computers enable us to tackle new types of problems at far greater scales than before
  - Parameter studies, ensembles, interactive data analysis, “workflows” of various kinds
  - Potentially an important source of new applications
- Such apps frequently stress petascale hardware and software in interesting ways
- New programming models and tools are required
  - Mixed task and data parallelism, management of many tasks, complex data management, failure, …
  - Tools for such problems (DAGman, Swift, Hadoop, …) exist but need refinement
- Interesting connections to distributed systems community

More info: www.ci.uchicago.edu/swift

Amiga Mars – Swift+Falkon

- 1024 Tasks (147456 micro-tasks)
- 256 CPU cores