

Data Analysis and Visualization of “Big Data” from HPC

Sean Ahern (Univ. of Tennessee, ORNL)

Dave Pugmire (ORNL)

George Ostrouchov (ORNL, UT)

Scott Klasky (ORNL, UT)

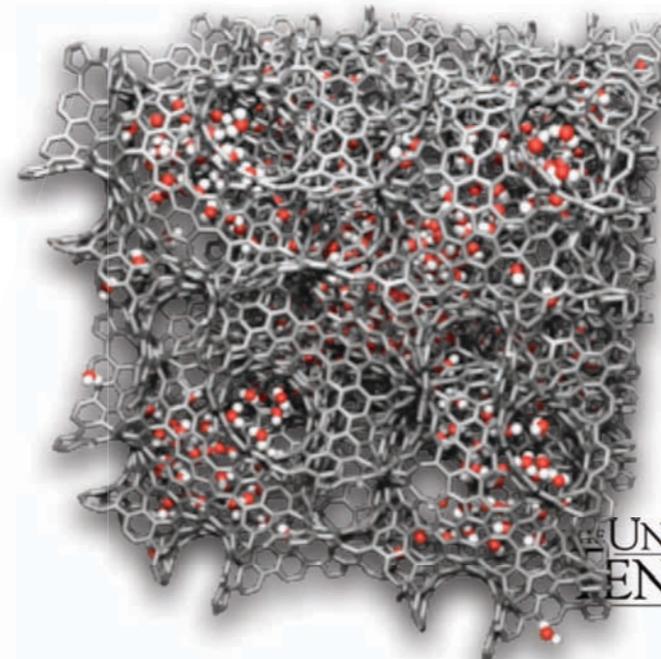
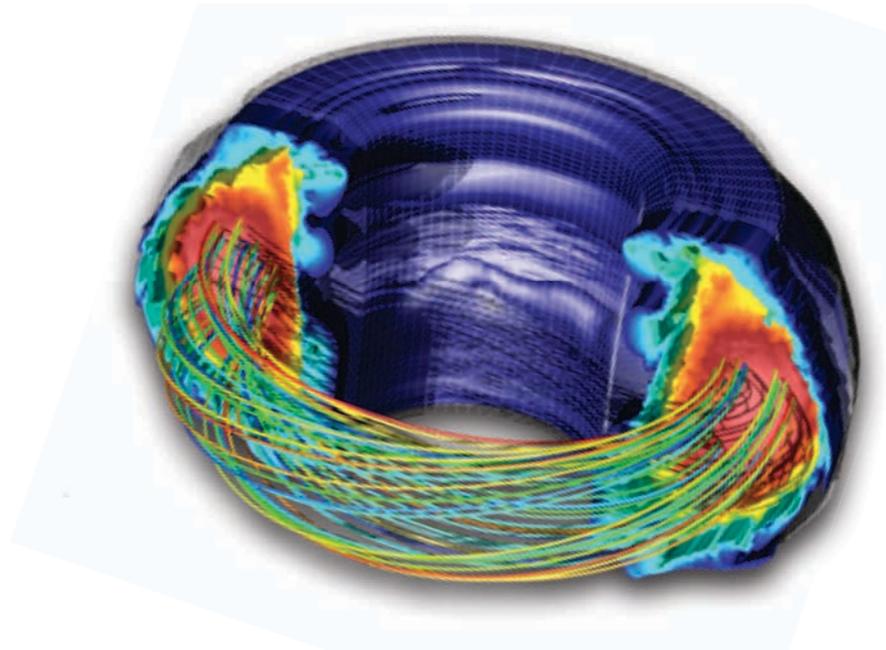
27 March 2013

What is “Big Data”?

- Big data is...

Extracting meaning from large sets of digital information

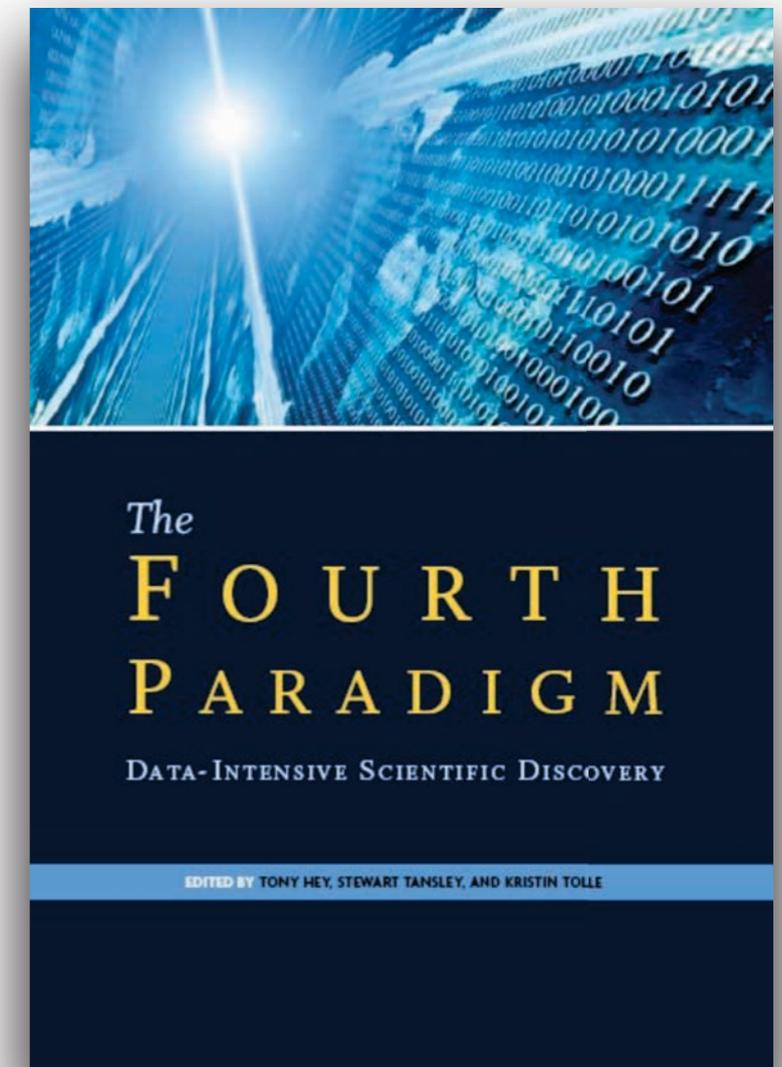
- Large: Beyond the capabilities of commonly used techniques
- HPC has been doing this for decades:
 - Using datasets from large-scale simulation of physical phenomena.



Data may be the “fourth pillar” of science



Data may be the “fourth pillar” of science

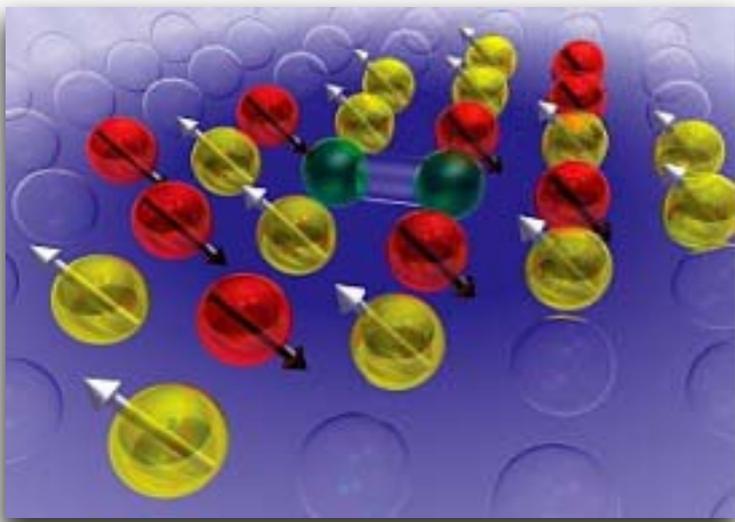


So what's all the hype about?

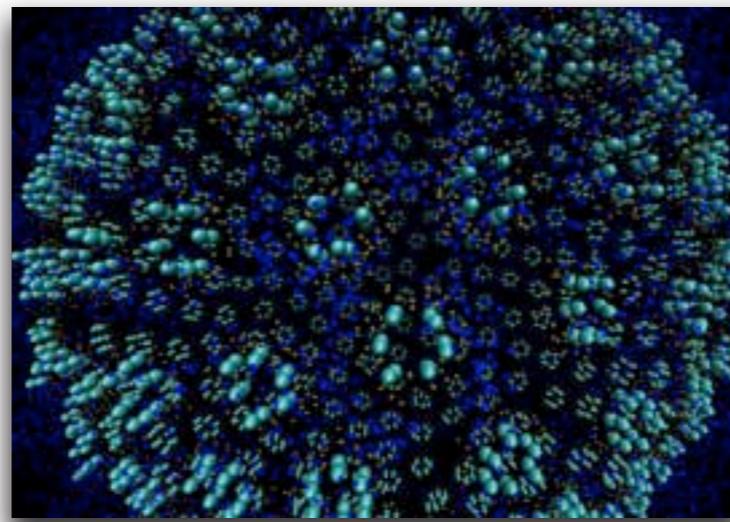
- Much of the hype of the last two years has been about extracting meaning...
 - ...from large unstructured and semi-structured datasets
 - ...from the Business Intelligence community
 - ...using scalable analysis and storage tools like:
Hadoop, Cassandra, HBase, Hive, Pig, MongoDB, ...
- Some successes in scientific computing for unstructured data



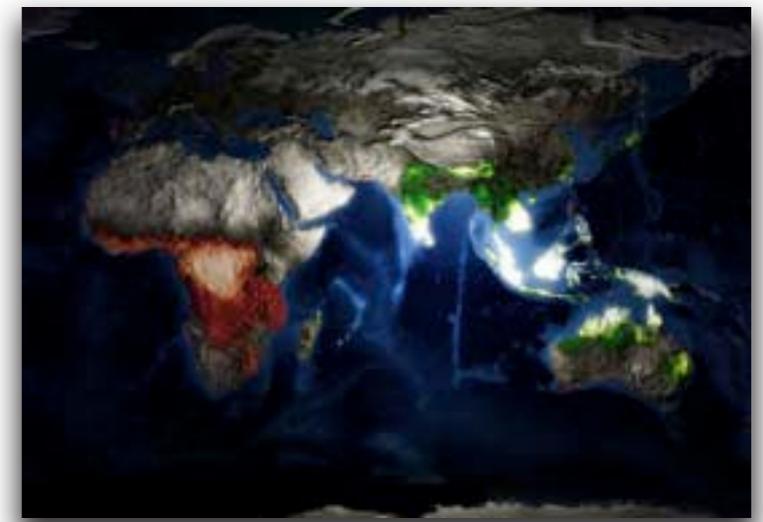
Petascale machines are solving important scientific problems through the generation and analysis of large datasets



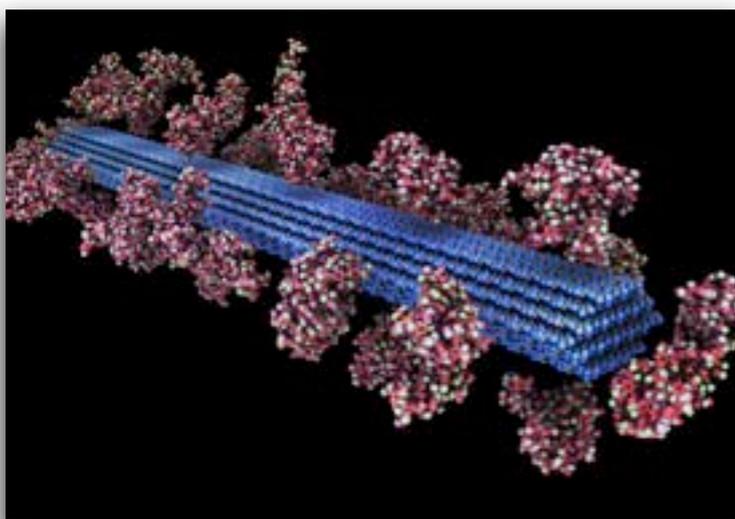
Physics of high-temperature superconducting cuprates



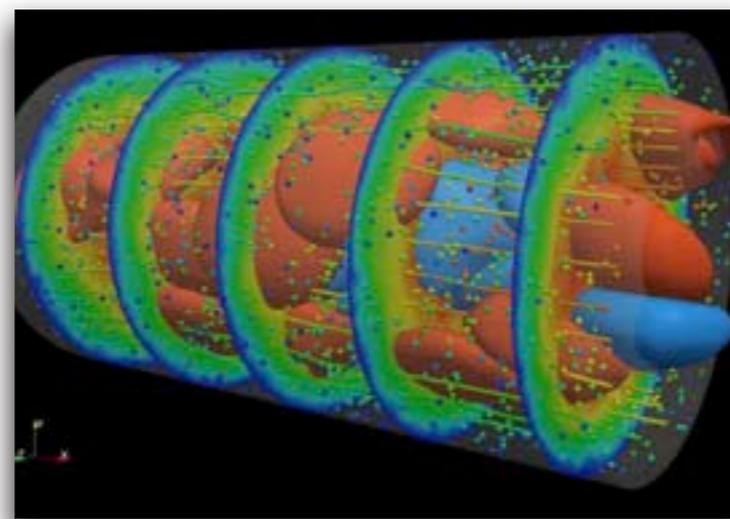
Chemical structure of HIV viral envelope



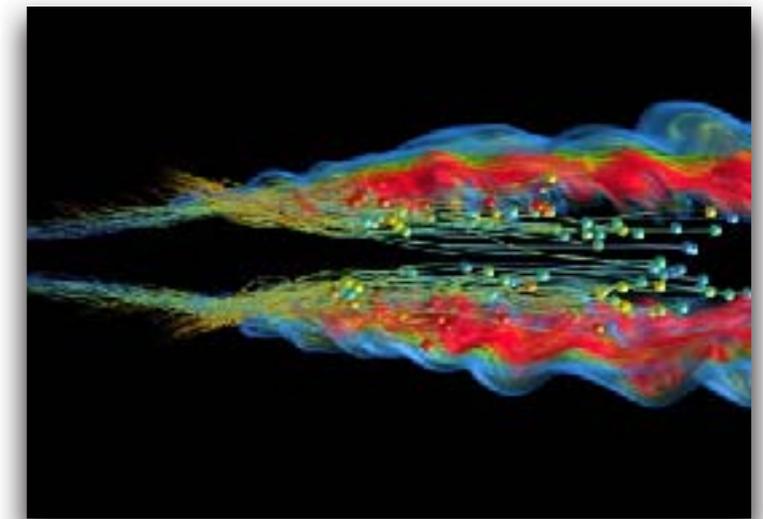
Global climate simulation of CO₂ dynamics and biochemistry



Biological conversion of cellulosic feedstock for biofuels



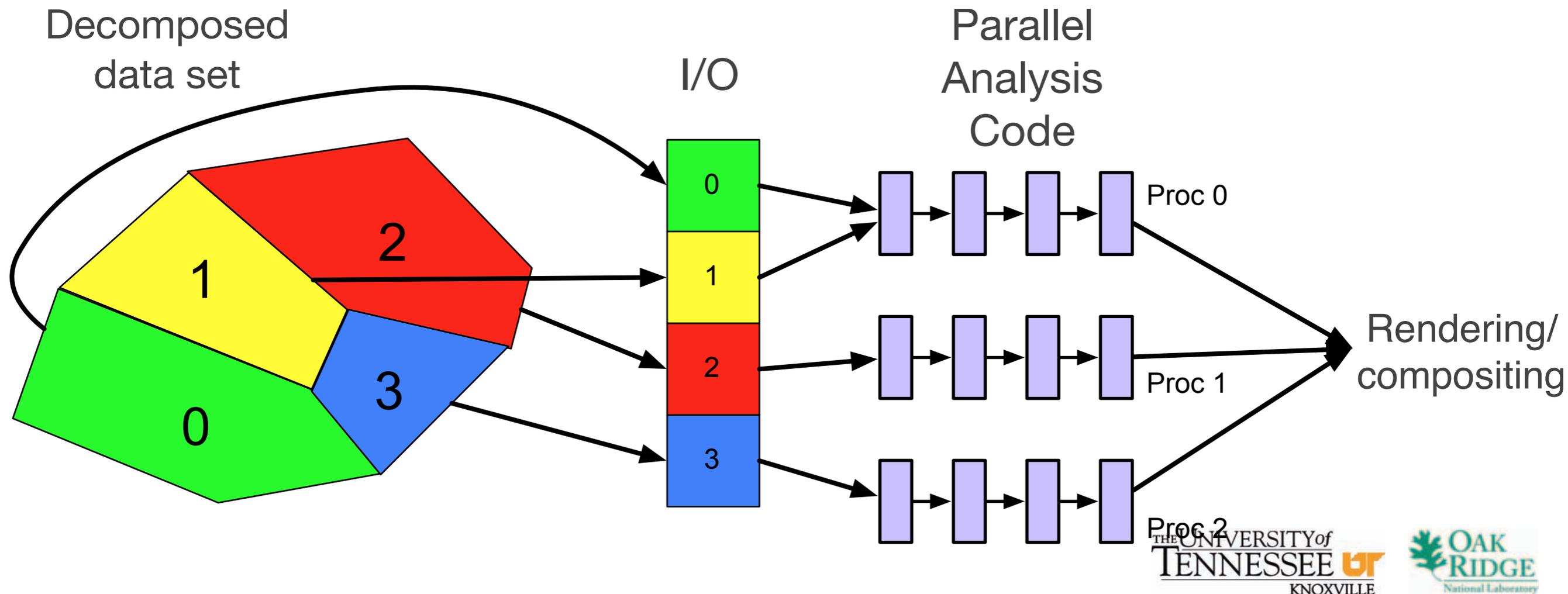
Three-dimensional simulation of blood flow



Combustion of turbulent lean fuel mixtures

Current success comes from having tools that scale to the HPC datasets

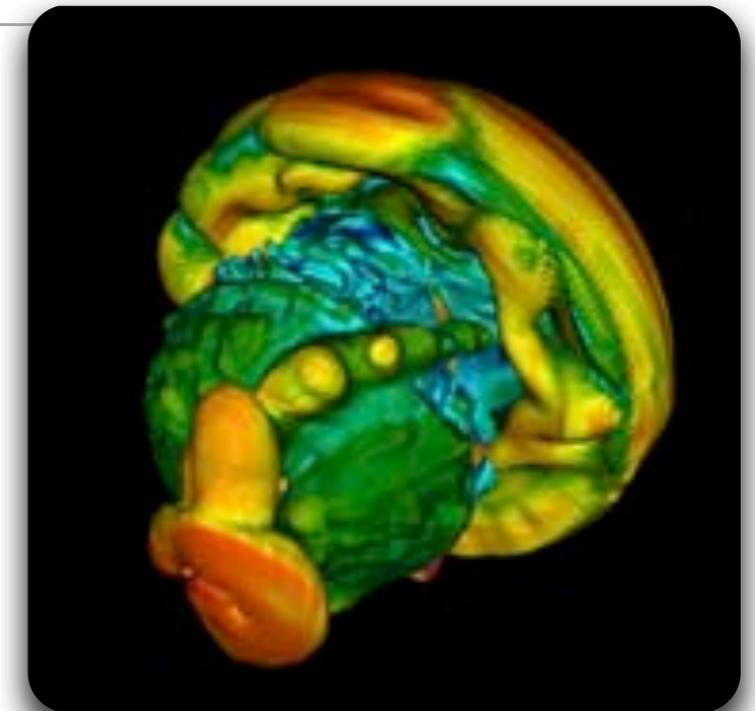
- Analysis methods are generally the same as in serial
- It's the tools and techniques that must scale
- SPMD data parallelism has taken us far -> Petascale+



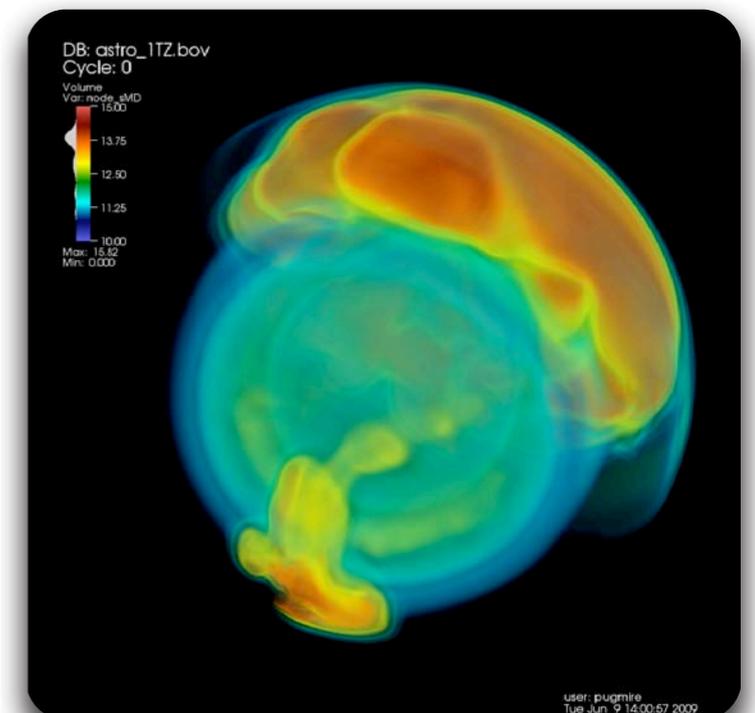
We've scaled to trillions of mesh cells

- Weak scaling study 2009 (isocontouring, volume rendering): ~63 million cells/core

Machine	Type	Problem Size	# cores
Jaguar	Cray XT5	2T	32k
Franklin	Cray XT4	1T, 2T	16k, 32k
Dawn	BG/P	4T	64k
Juno	Linux	1T	16k, 32k
Ranger	Sun Linux	1T	16k
Purple	AIX	0.5T	8k

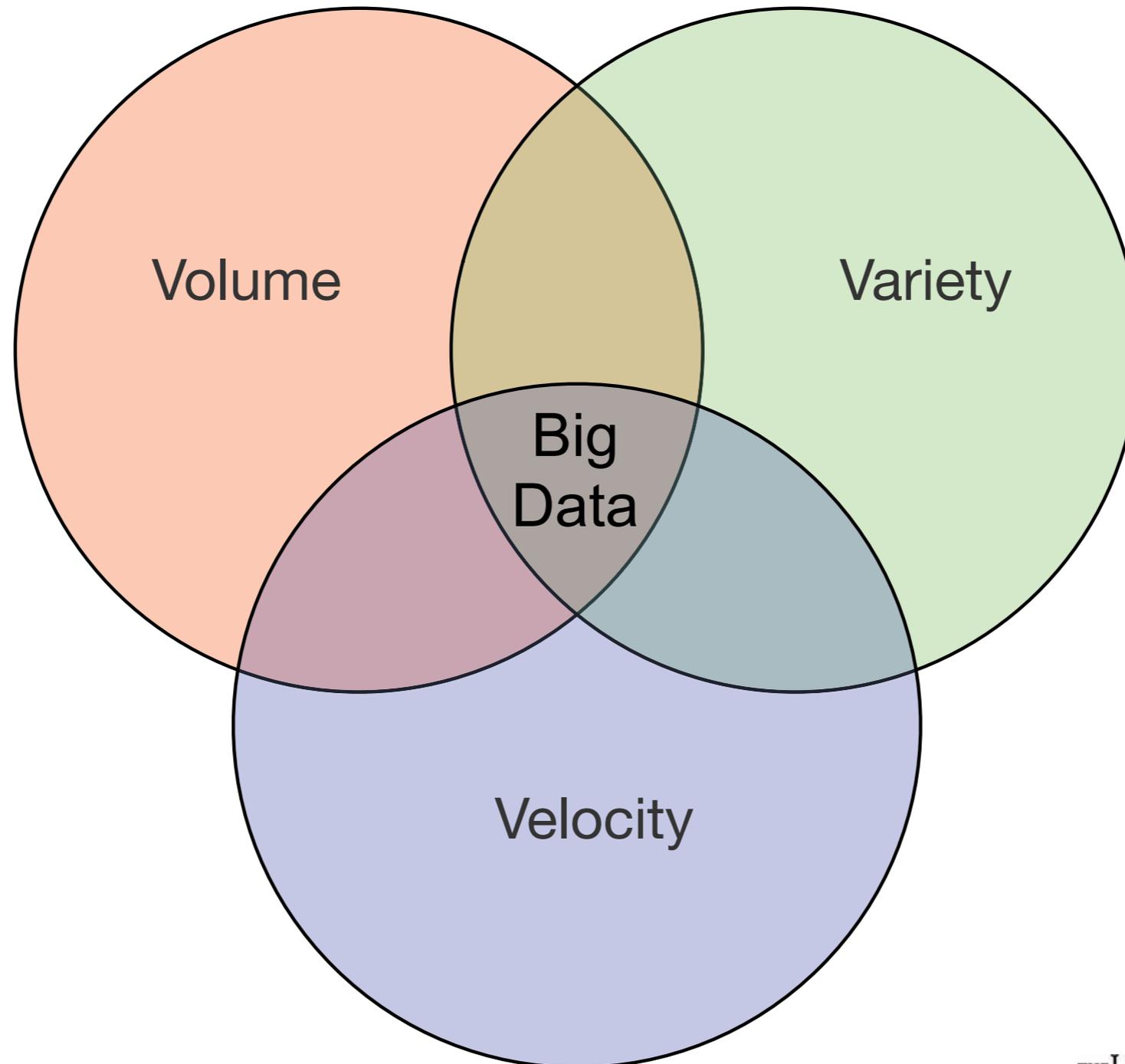


2T cells, 32K procs



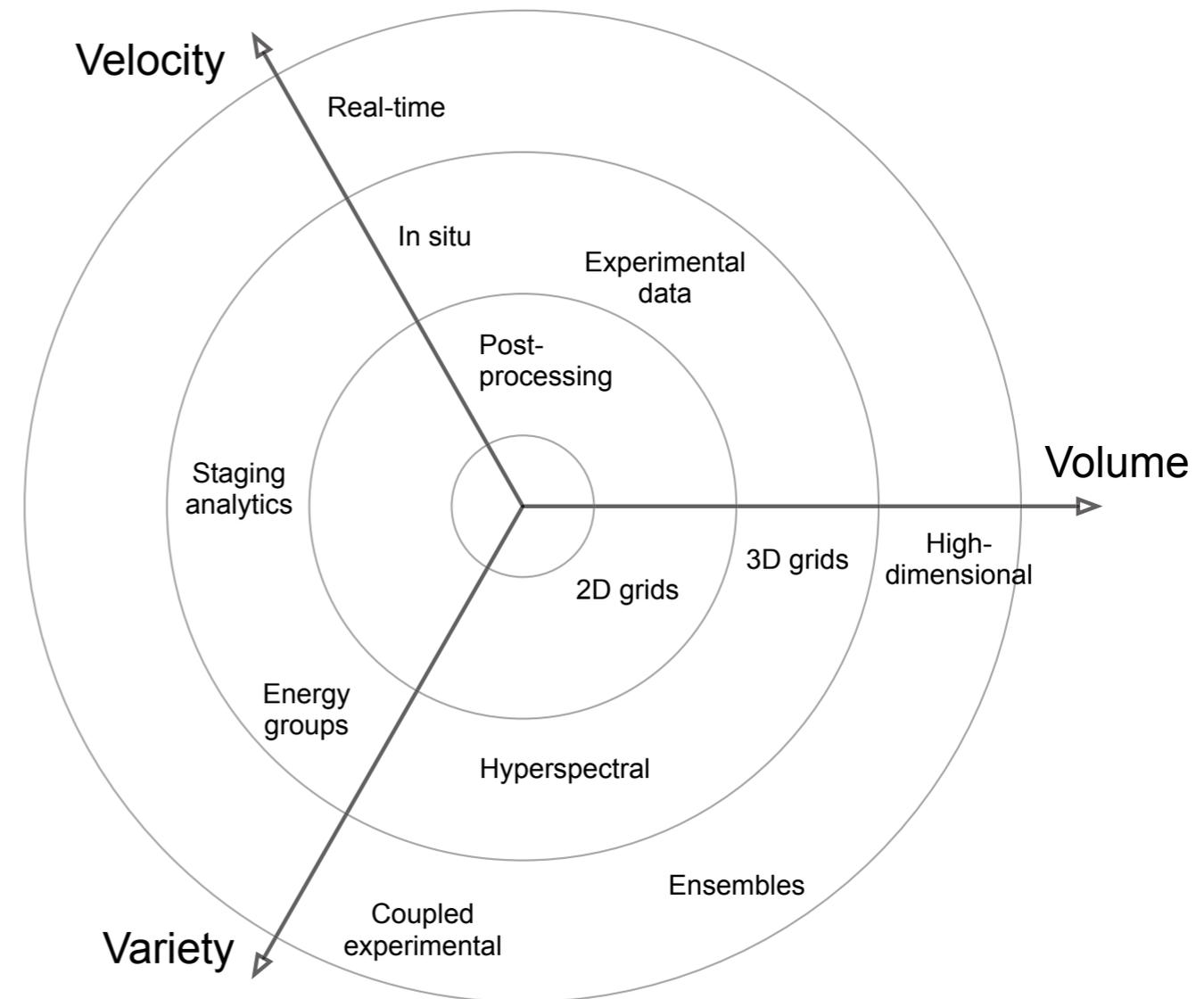
Since this work, people have reached 8 trillion cells = $20,000^3$ cells.

The “Three V’s” of big data as applied to HPC



The “Three V’s” of big data as applied to HPC

- Volume:
 - Increasing mesh resolution
 - Increasing temporal complexity
- Variety:
 - Higher-dimensional data
 - Increasing multivariance
 - Complex data models
 - Ensembles, parameter studies
- Velocity:
 - Future hardware constraints will limit I/O



Data scale limits scientific understanding

- Spatial resolution increasing in many domains
 - Too many zones to see on screen (with or without high-res Powerwall)
- Temporal complexity increasing
 - Climate simulations will have 100,000s of time steps
 - Manually finding temporal patterns is tedious and error-prone
- Multivariate overload
 - Climate simulations have 100-200 variables
 - Growing to thousands
- Issues of data models and domain-specific data
 - E.g., Multigroup radiation fields

Common Themes in Mathematics for Data

Computation Mode	Memory Architecture	Data Partition Availability	Data Passes
Serial or Multithreaded	contiguous	N/A (all)	many
Distributed	partitioned	all	many
Out-of-Core	partitioned	one-at-a-time	one or few
Streaming/ in situ	partitioned	one-at-a-time	one
Mathematics needed	partition coupling math	updating via partition coupling math	one-pass updating via partition coupling math

Minimizing data access (energy) pushes everyone toward one-pass (one-touch) updating.

Common Themes in Mathematics for Data

Computation Mode	Memory Architecture	Data Partition Availability	Data Passes
Serial or Multithreaded	contiguous	N/A (all)	many
Distributed	partitioned	all	many
Out-of-Core	partitioned	one-at-a-time	one or few
Streaming/ in situ	partitioned	one-at-a-time	one
Mathematics needed	partition coupling math	updating via partition coupling math	one-pass updating via partition coupling math

Minimizing data access (energy) pushes everyone toward one-pass (one-touch) updating.

Prediction of Ice and Climate Evolution at Extreme Scales (PISCEES)

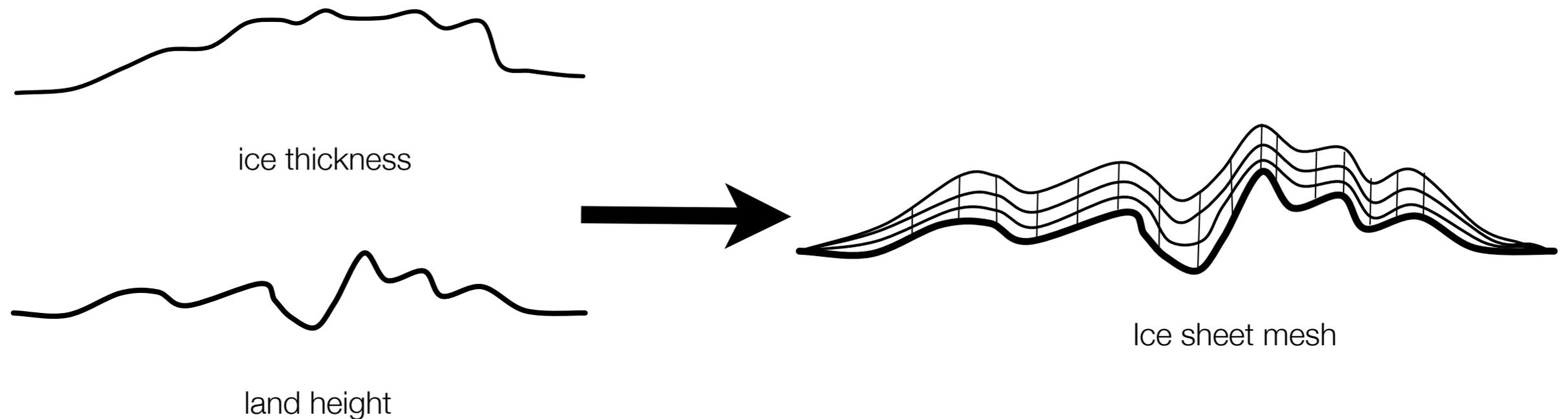
- SciDAC Project that is:
 - Developing robust, accurate, and scalable dynamical cores for ice sheet modeling on a variety of mesh types
 - Evaluate ice sheet models using new tools for V&V and UQ
 - Integrate these models and tools into the Community Ice Sheet Model (CISM) and Community Earth System Model (CESM)
 - Simulate decade-to-century-scale evolution of the Greenland and Antarctic ice sheets



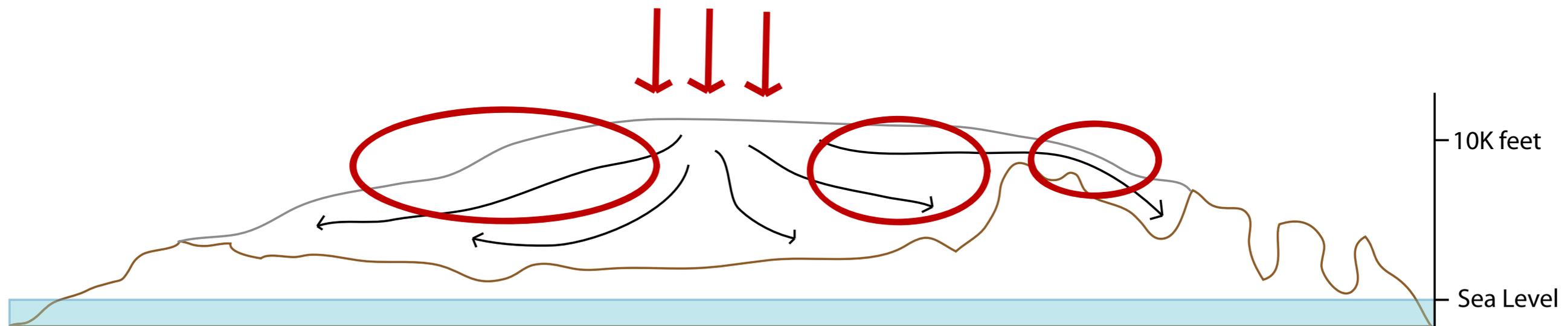
PI: William Lipscomb, Philip Jones (Acting)

Data set structure

- Structured NetCDF files
 - Ice defined by two variables: topg, thk
 - Ice flow sheets defined by N levels



Flow Analysis in PISCEES



We have a scalable integral curve algorithm

Parallelize
over **blocks**

- Use a combination of seed and block parallelization

- Dynamically steer computation
- Allocate resources where needed
- Load blocks from disk on demand

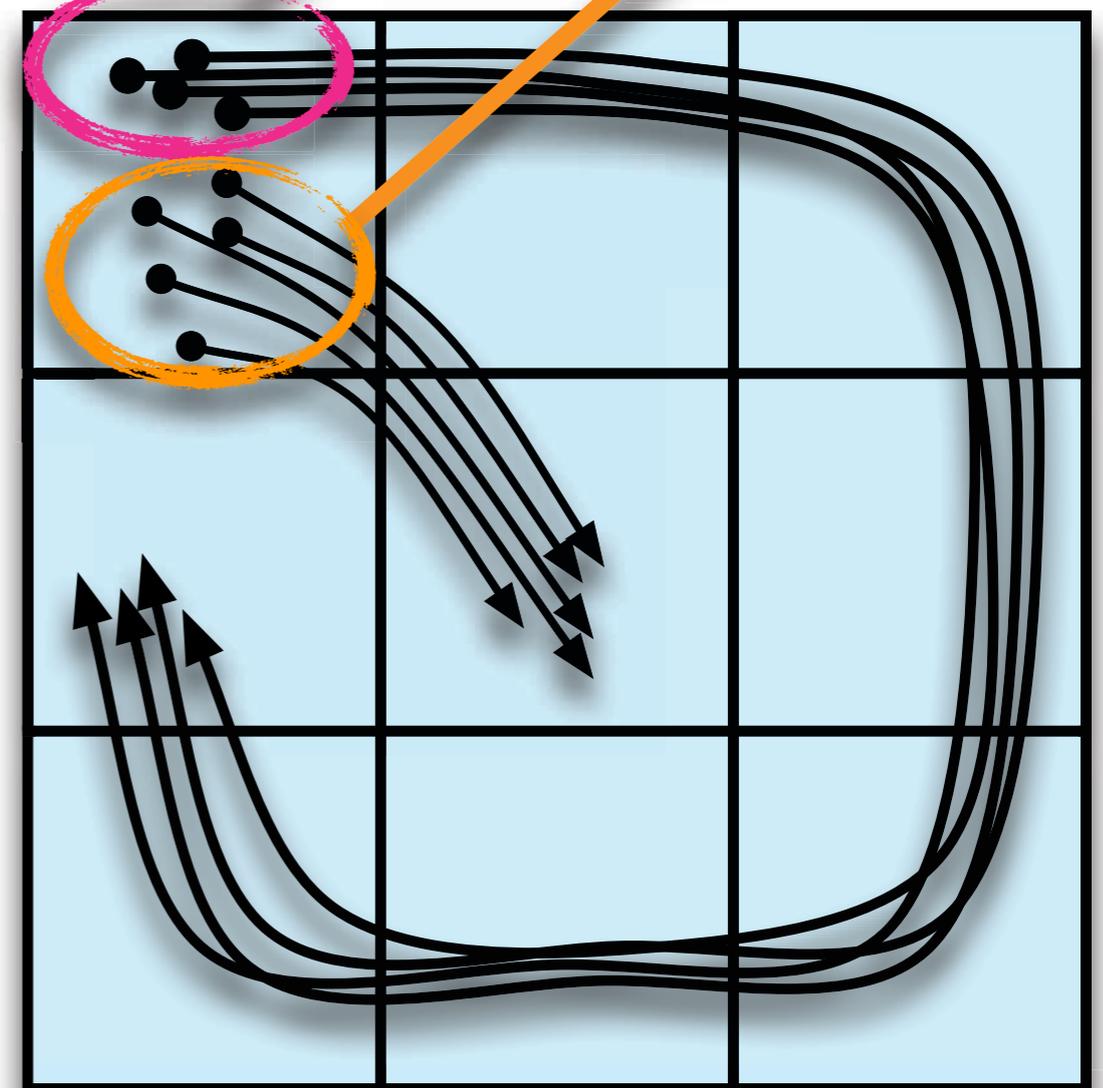
Parallelize
over **seeds**

- Parallelize over seeds **and** blocks

- Goals:

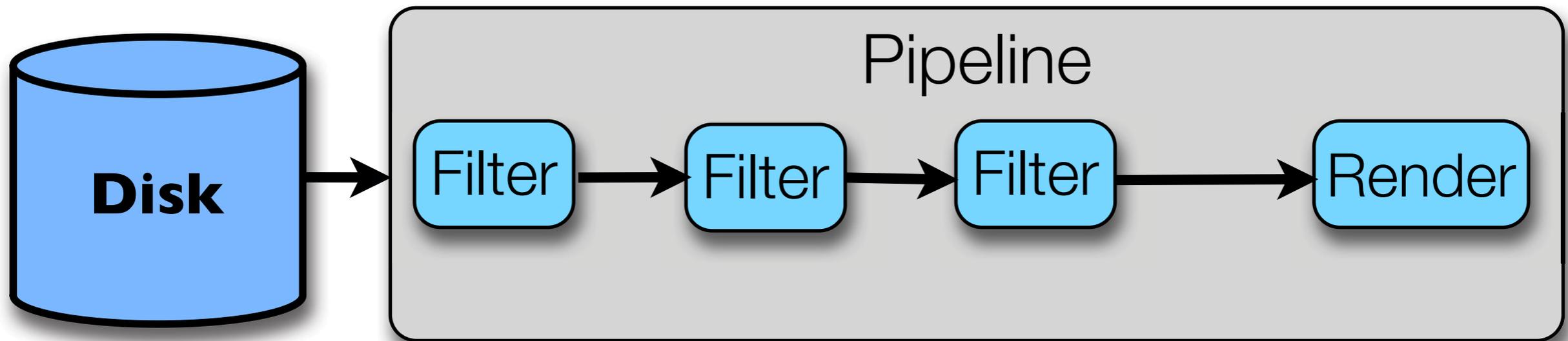
1. Maximize processor utilization
2. Minimizes IO
3. Minimizes communication

- Works adequately in all use cases

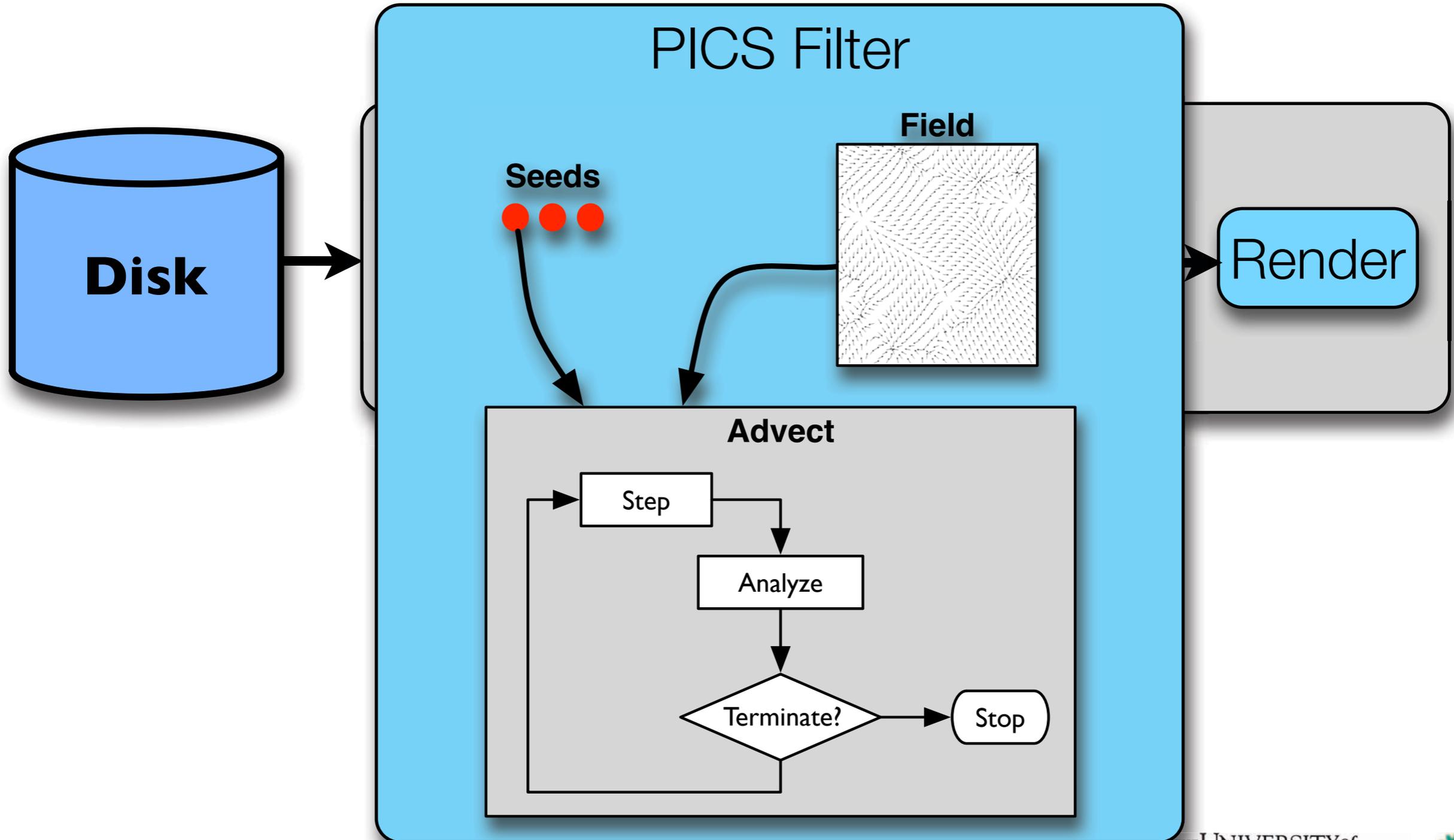


Parallel Integral Curve System (PICS)

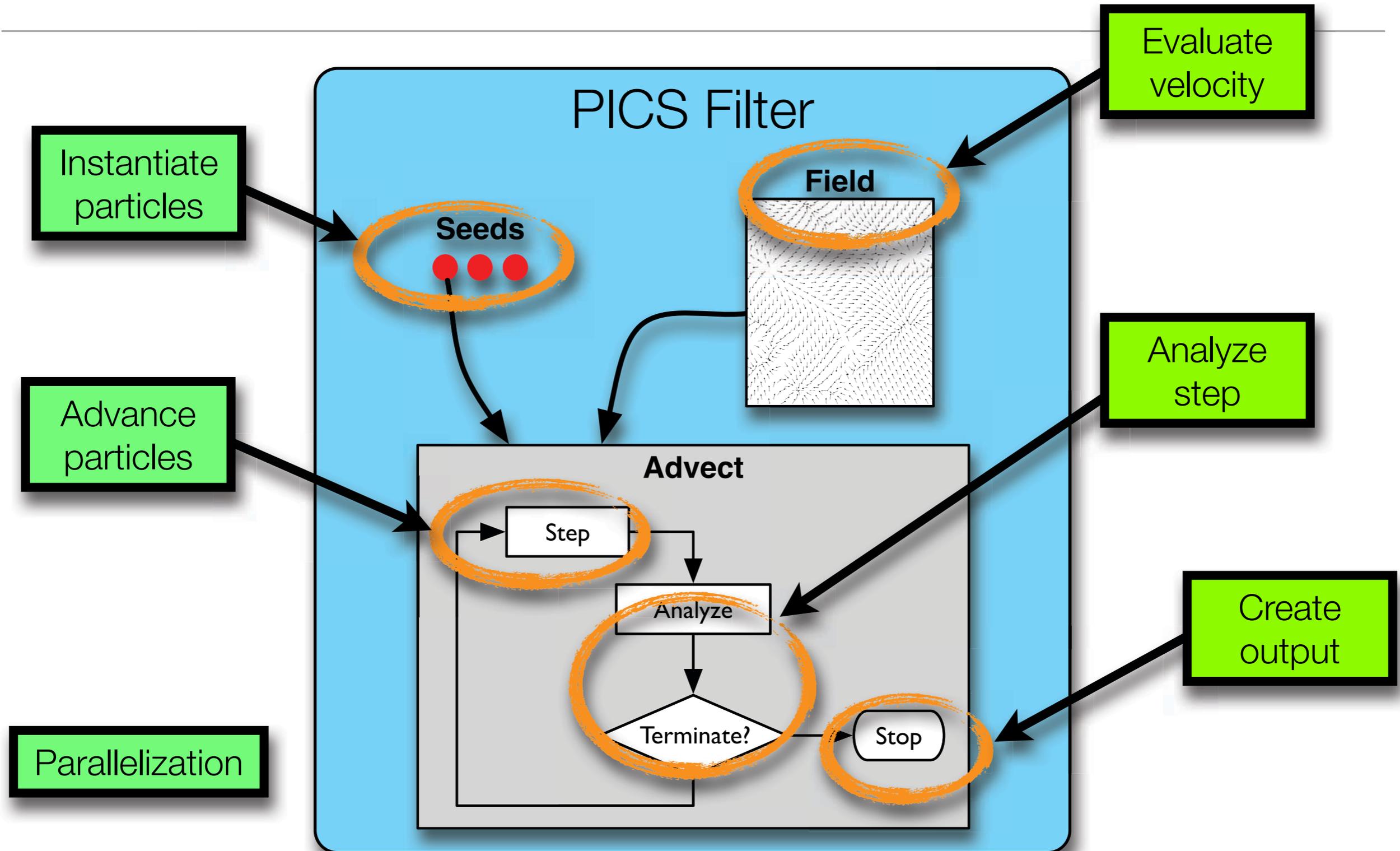
Data flow network



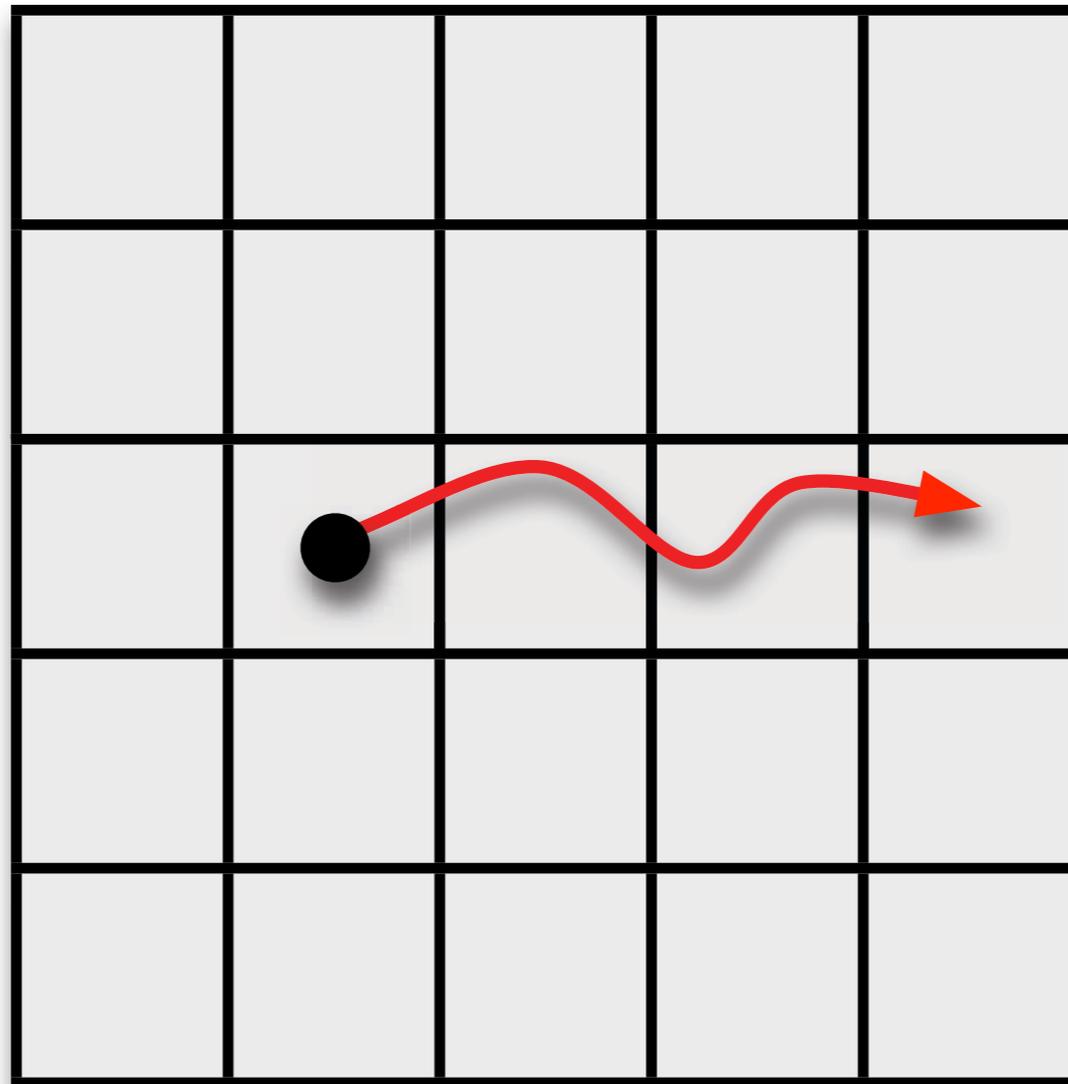
Parallel Integral Curve System (PICS)



PICS Filter Design



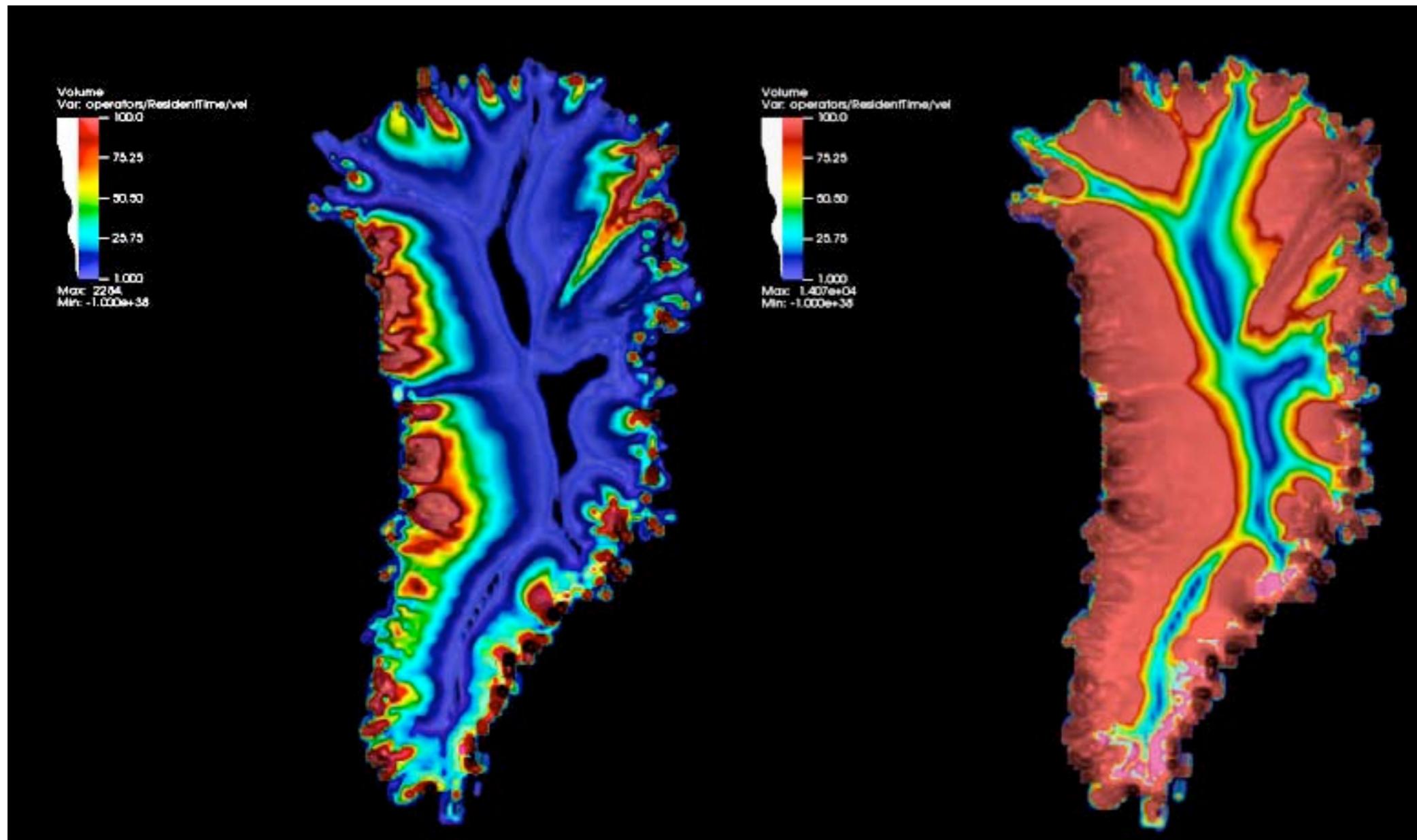
“Distance Traveled” operator



“Distance Traveled” operator

	12.83			

Results



1 year of travel

10 years of travel

Steady state flow distance traveled

Big Data Exploration Requires Flexibility in Analytics

- Analytics for not-so-big data:
 - Science produces great many algorithms
 - Often redundant and related
 - Statistics and mathematics prune, generalize, and connect
 - Still we have hundreds of statistics and mathematics books on data analytics



No reason to expect big data less needy than not-so-big data
Flexibility and a big toolbox are critical for exploration
Need for analytics diversity persists for big data

Programming with Big Data in R: pbdR

- R has unmatched diversity and flexibility for data
- Goals:
 - Shorten time from big data to science insight
 - Productivity, Portability, Performance
- Approach:
 - Implicit management of distributed data details
 - Scalable, big data analytics with high-level syntax
 - Identical syntax to serial R
 - Powered by state of the art scalable libraries
 - Free* R packages

pbdR Core Team

George Ostrouchov^{1,2}, Team Lead
Wei-Chen Chen¹
Pragnesh Patel²
Drew Schmidt²

¹Oak Ridge National Laboratory

²University of Tennessee

*MPL, GPL and BSD licensed

High-Level Syntax

Covariance

Developer

```
Cov.X <- cov(X)
```

```
N <- nrow(X)
mu <- colSums(X) / N
X <- sweep(X, STATS=mu, MARGIN=2)
Cov.X <- crossprod(X.spmd) / (N-1)
```

```
Cov.X <- cov(X)
```

```
N <- allreduce(nrow(X))
mu <- allreduce(colSums(X) / N)
X <- sweep(X, STATS=mu, MARGIN=2)
Cov.X <- allreduce(crossprod(X)) / (N-1)
```

Linear Models

Developer

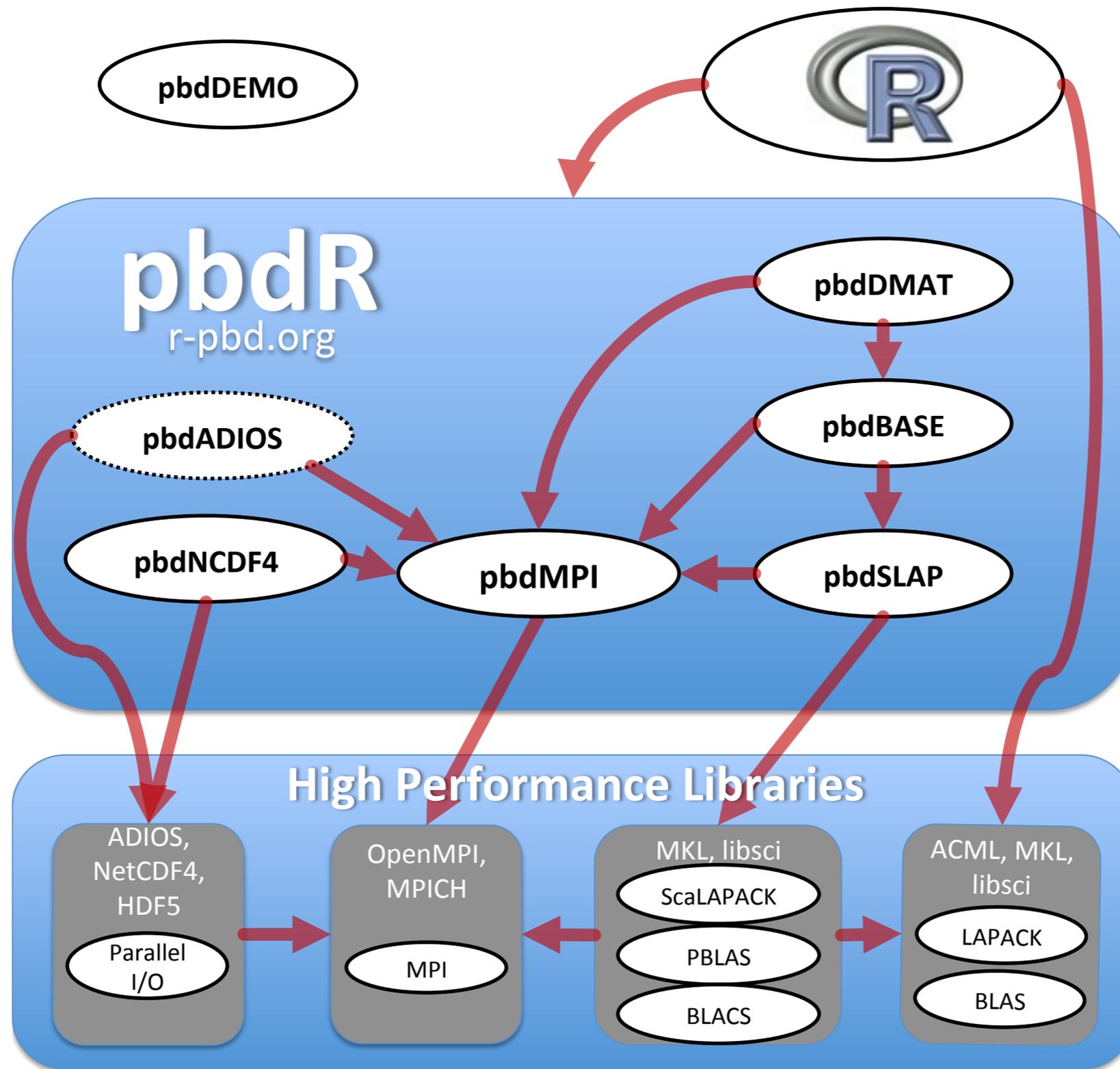
```
Lm.X <- lm.fit(X, Y)
```

```
tX <- t(X)
A <- tX %**% X
B <- tX %**% Y
ols <- solve(A) %**% B
```

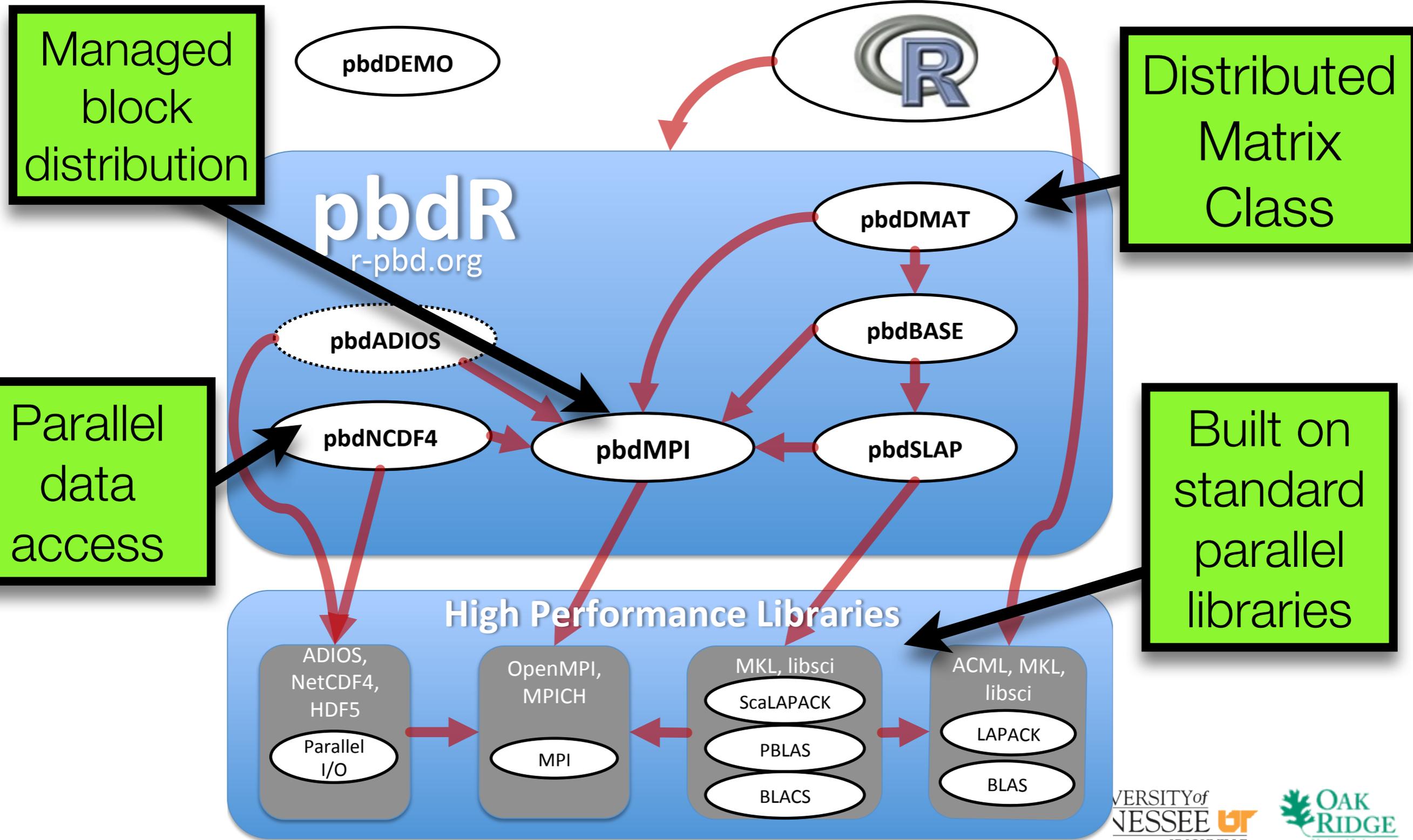
```
Lm.X <- lm.fit(X, Y)
```

```
tX <- t(X)
A <- allreduce(tX %**% X)
B <- allreduce(tX %**% Y)
ols <- solve(A) %**% B
```

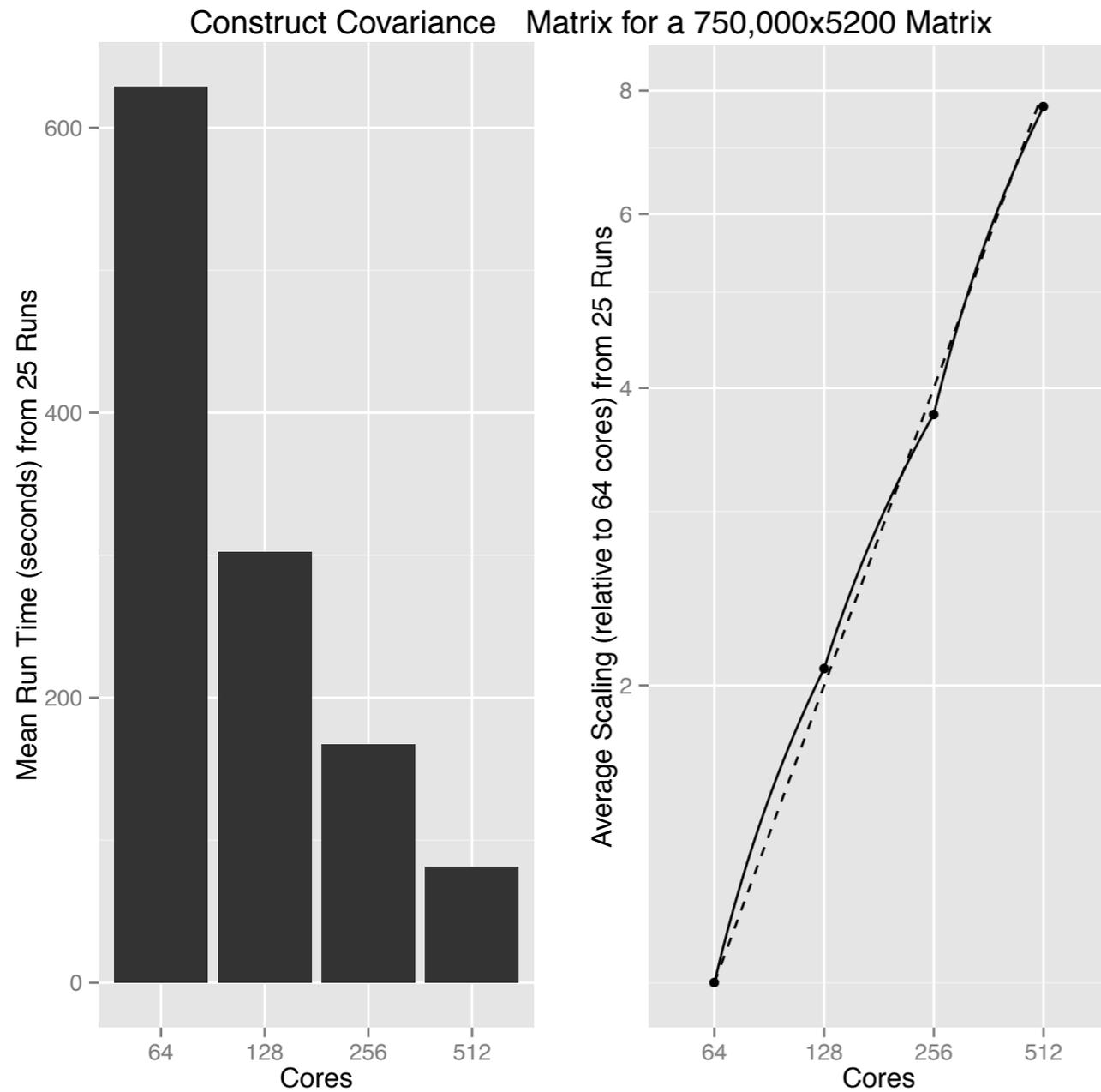
A High-Level Language for Big Data Analytics



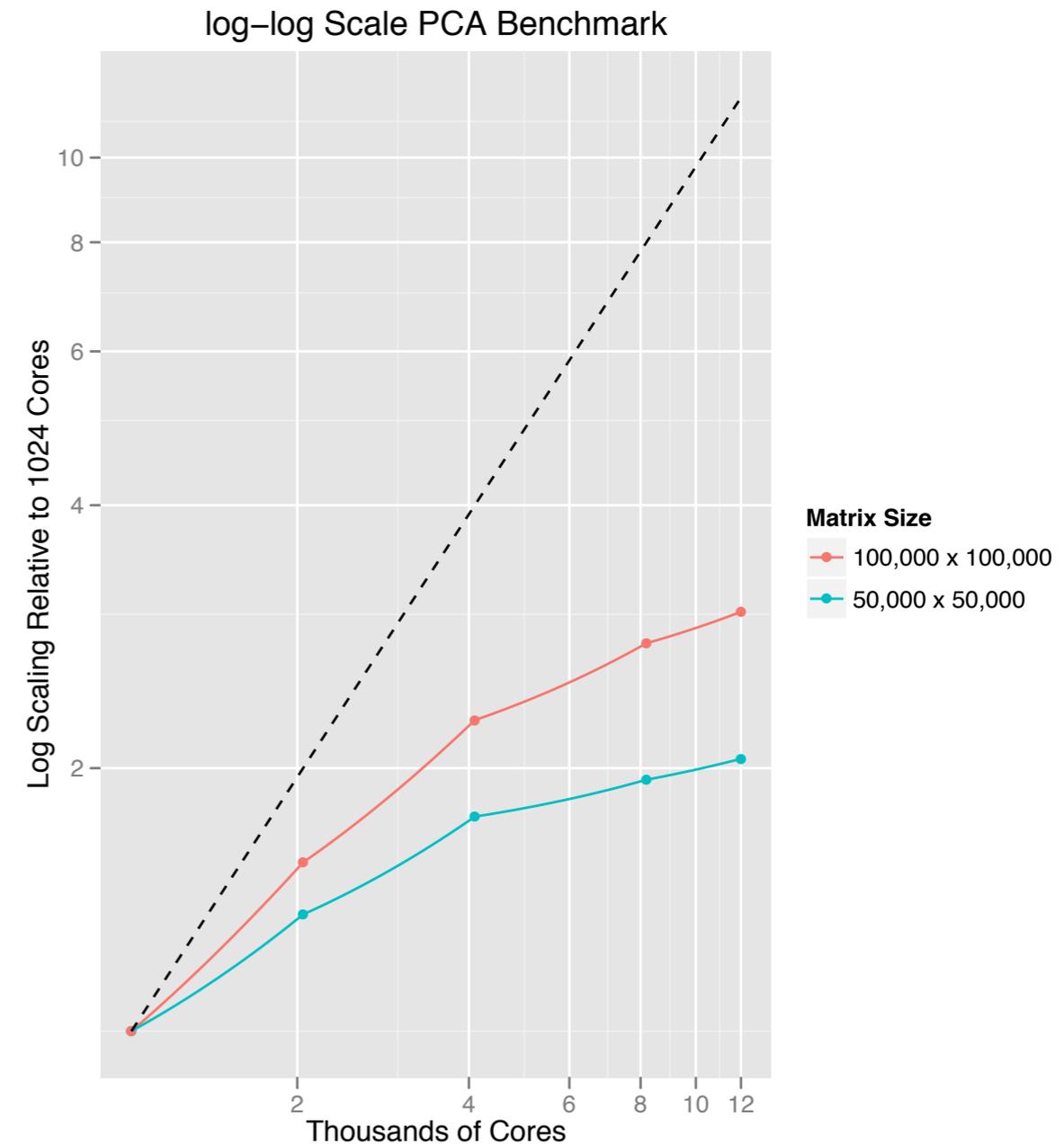
A High-Level Language for Big Data Analytics



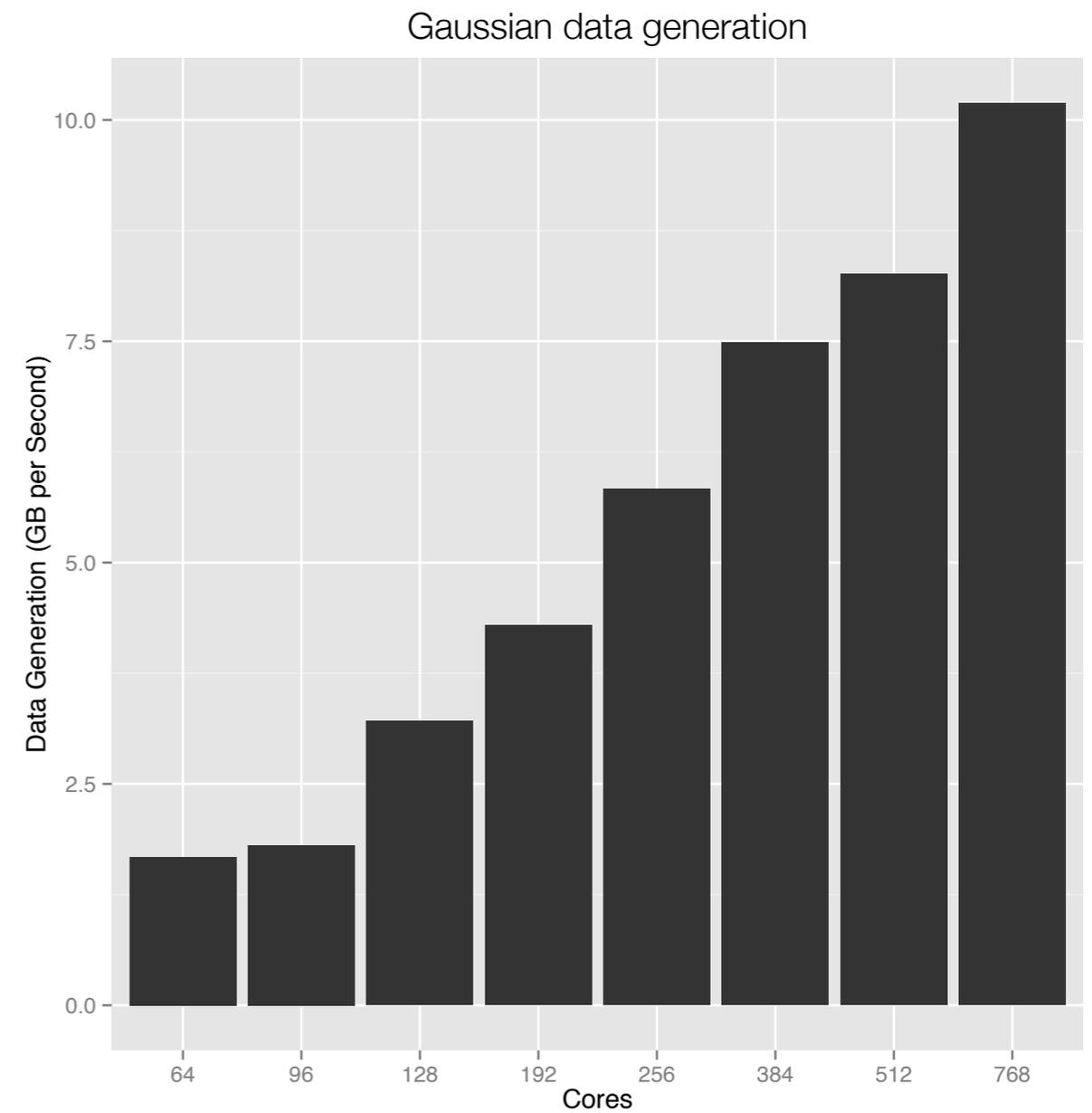
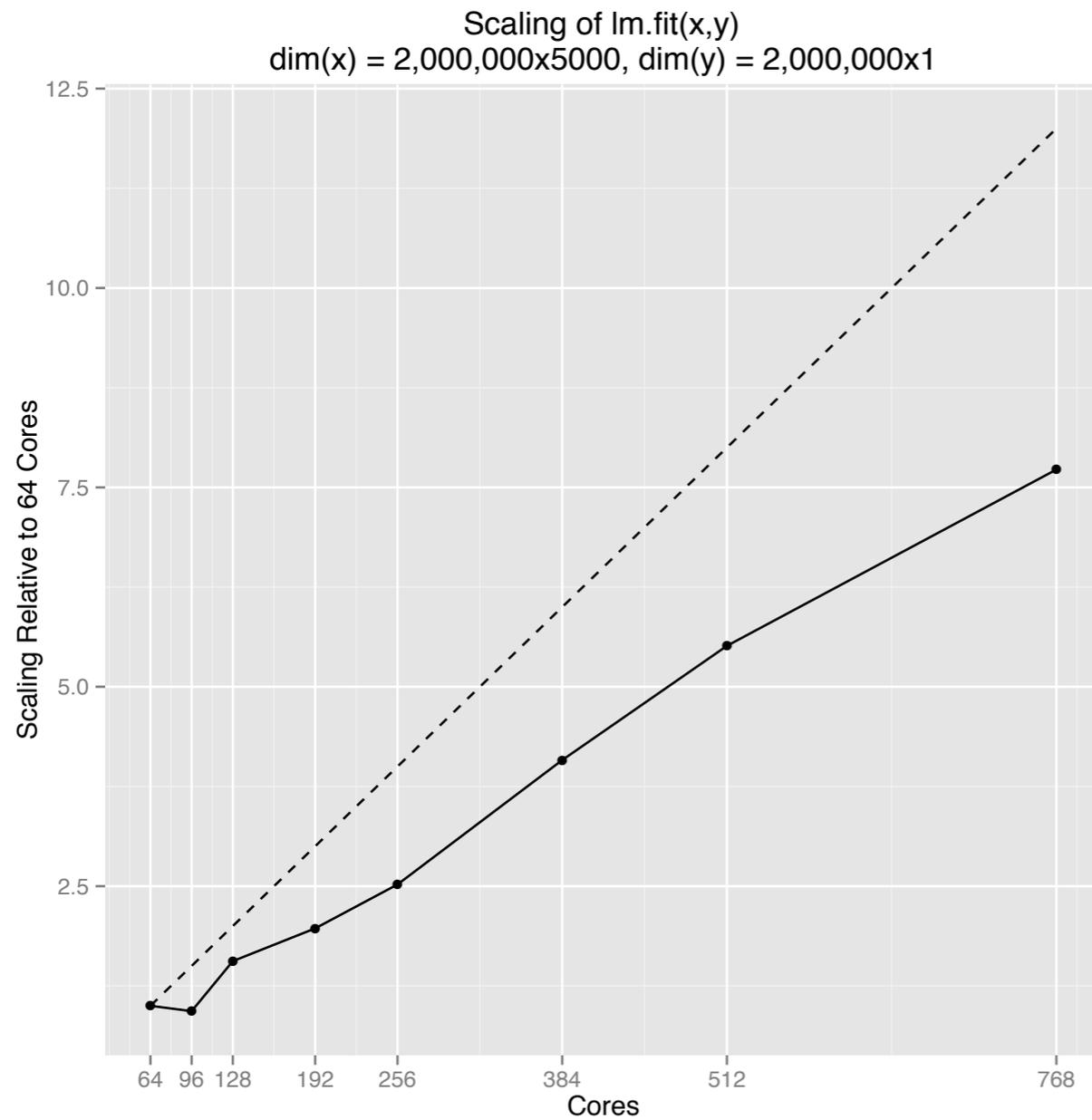
Promising Scalability



Nautilus



Promising Scalability



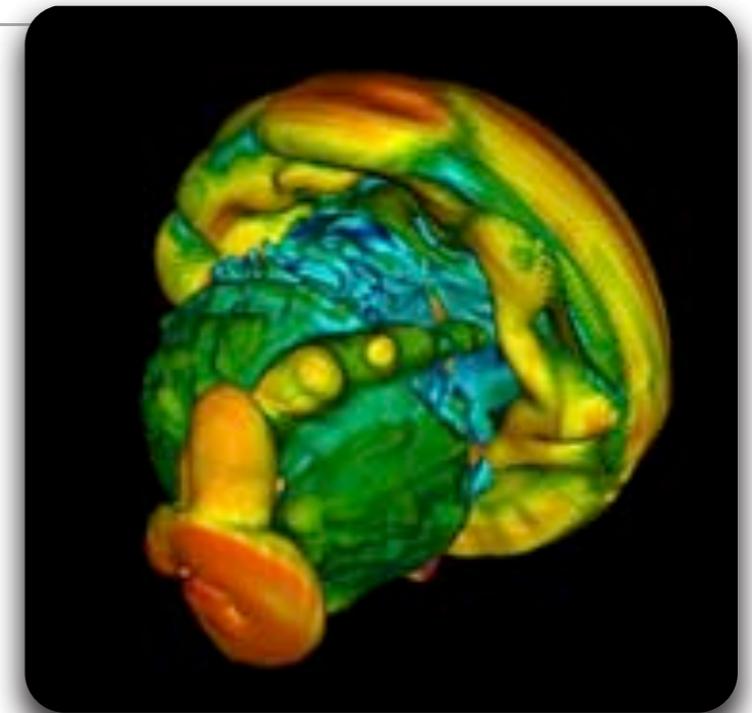
Nautilus

Performance is highly dependent upon I/O rates

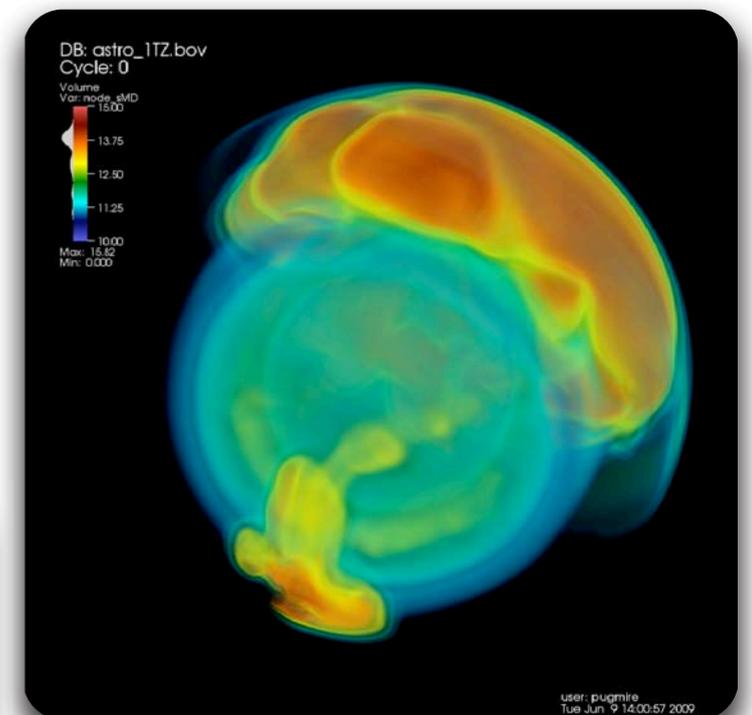
- Weak scaling study 2009 (isocontouring, volume rendering): ~63 million cells/core

Machine	Type	Problem Size	# cores
Jaguar	Cray XT5	2T	32k
Franklin	Cray XT4	1T, 2T	16k, 32k
Dawn	BG/P	4T	64k
Juno	Linux	1T	16k, 32k
Ranger	Sun Linux	1T	16k
Purple	AIX	0.5T	8k

- Appx I/O time: 2-5 minutes
- Appx processing time: 10 seconds



2T cells, 32K procs



Hardware technology curves at the exascale: ~2018

- Billion-way concurrency
- Constrained memory environment
- Very constrained I/O
(in relation to compute capability)

NSF CIF21: “Computing power, data volumes, software, and network capacities are all on exponential growth paths.”

System Parameter	2011	“2018”		Factor Change
		Swim Lane 1	Swim Lane 2	
System Peak	2 Pf/s	1 Ef/s		500
Power	6 MW	≤20 MW		3
System Memory	0.3 PB	32–64 PB		100–200
Total Concurrency	225K	1B×10	1B×100	40,000–400,000
Node Performance	125 GF	1 TF	10 TF	8–80
Node Concurrency	12	1,000	10,000	83–830
Network BW	1.5 GB/s	100 GB/s	1000 GB/s	66–660
System Size (nodes)	18700	1,000,000	100,000	50–500
I/O Capacity	15 PB	300–1000 PB		20–67
I/O BW	0.2 TB/s	20–60 TB/s		10–30

Hardware technology curves at the exascale: ~2018

- Billion-way concurrency
- Constrained memory environment
- Very constrained I/O
(in relation to compute capability)

NSF CIF21: “Computing power, data volumes, software, and network capacities are all on exponential growth paths.”

System Parameter	2011	“2018”		Factor Change
		Swim Lane 1	Swim Lane 2	
System Peak	2 Pf/s	1 Ef/s		500
Power	6 MW	≤20 MW		3
System Memory	0.3 PB	32–64 PB		100–200
Total Concurrency	225K	1B×10	1B×100	40,000–400,000
Node Performance	125 GF	1 TF	10 TF	8–80
Node Concurrency	12	1,000	10,000	83–830
Network BW	1.5 GB/s	100 GB/s	1000 GB/s	66–660
System Size (nodes)	18700	1,000,000	100,000	50–500
I/O Capacity	15 PB	300–1000 PB		20–67
I/O BW	0.2 TB/s	20–60 TB/s		10–30

Implications for analysis & visualization

- Extremely limited ability to move data off HPC resource
 - The de-coupling of simulation codes and visualization codes can no longer continue
- Extremely limited memory space
- Extreme concurrency will break most communication models
 - Pure MPI already breaking at the petascale
- Fine degree of data parallelism will break most data processing models

So what do we do?

- We've had a lot of success moving processing closer and closer to the data.
- Medium-term goal: Take the final step in moving processing closer to the data
 - Right to the source, in memory where it's generated
 - Reduce data every step of the way to the user
 - All data visualization (and analysis) becomes remote
- Long-term goal: Move data processing everywhere the data lives

Injection of analysis into I/O pipelines

- Leverages existing code methods for data movement
 - Lots of compute available
 - Exploits high-capacity HPC interconnect
- Possible to annotate, reduce, and reorder data in flight
- I/O is the primary bottleneck, so it's the logical place to begin adaptation
- Challenges:
 - Memory-constrained
 - Impossible to do serendipitous discovery with in situ alone
 - Temporal analysis can be difficult

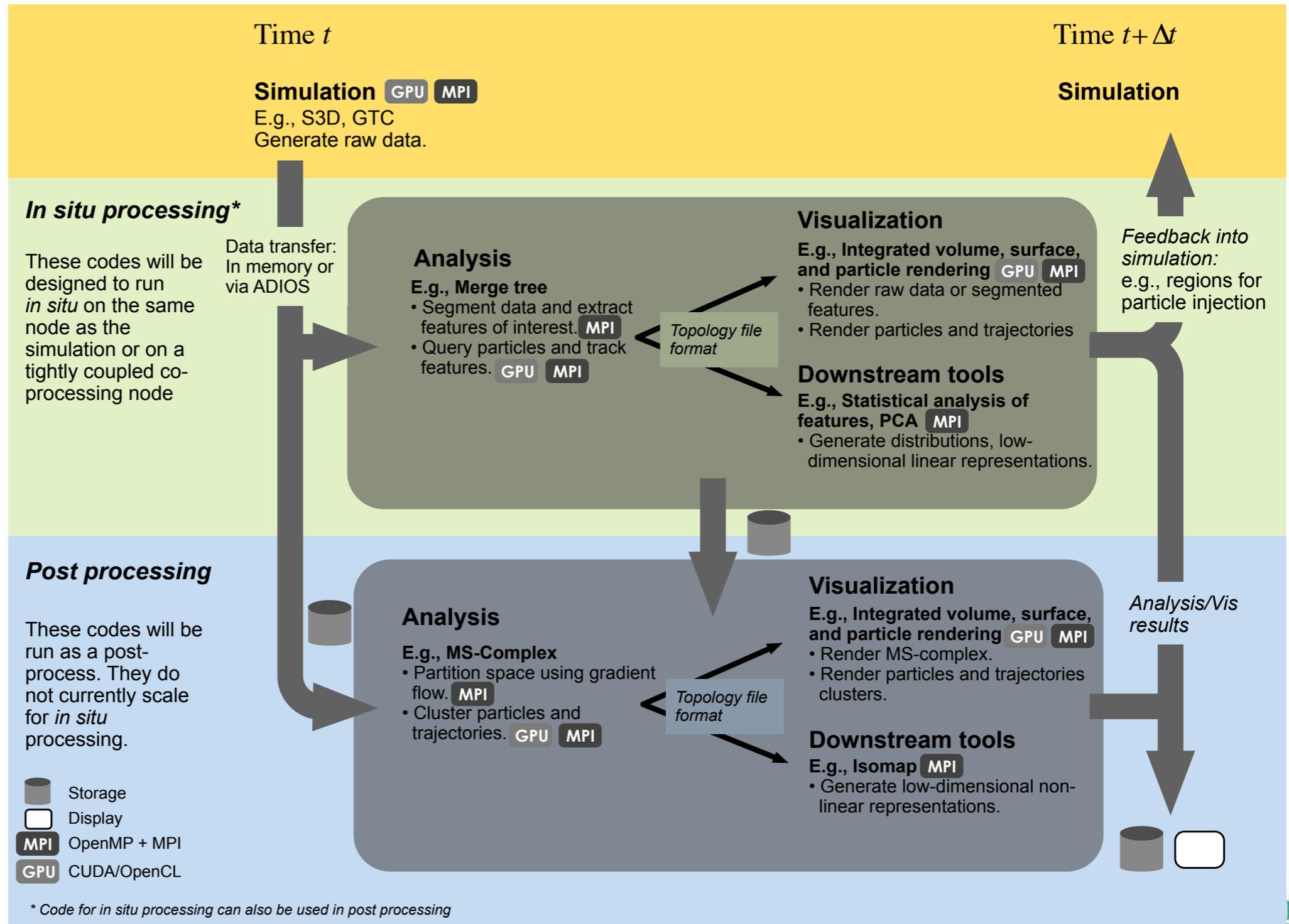
Goal: Tightly-coupled analysis infrastructures

Simulation, analysis, visualization workflow

Simulation

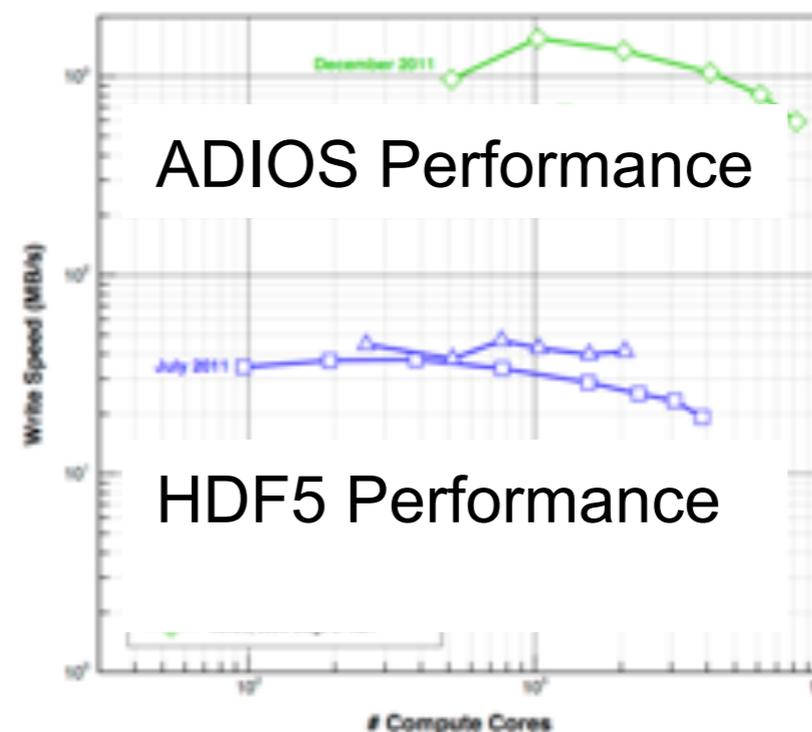
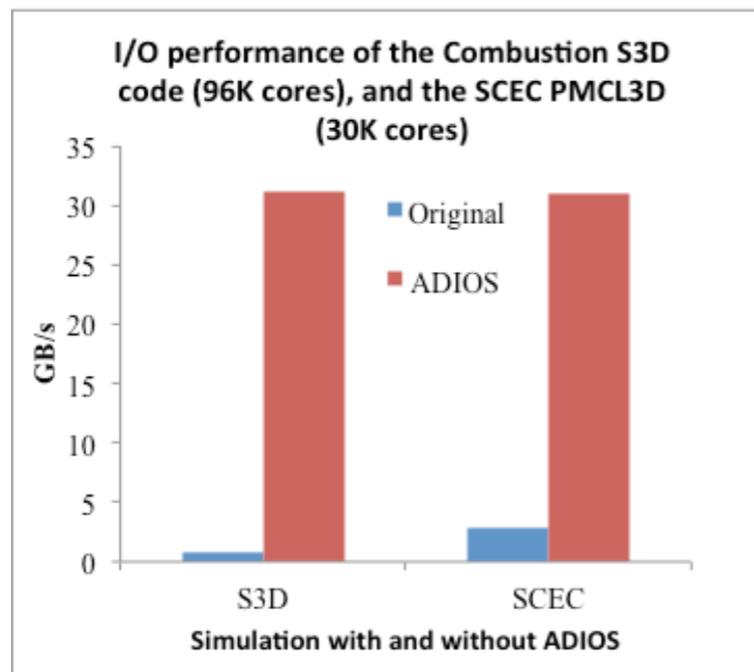
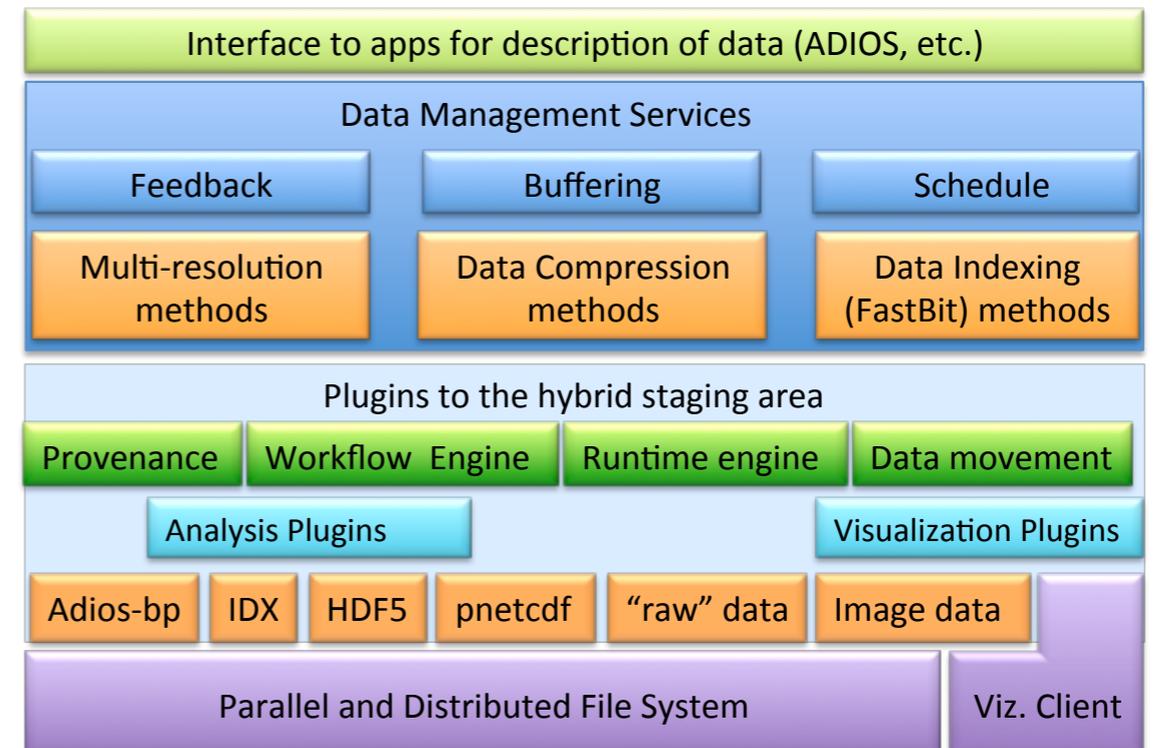
Coupled analysis

Post processing



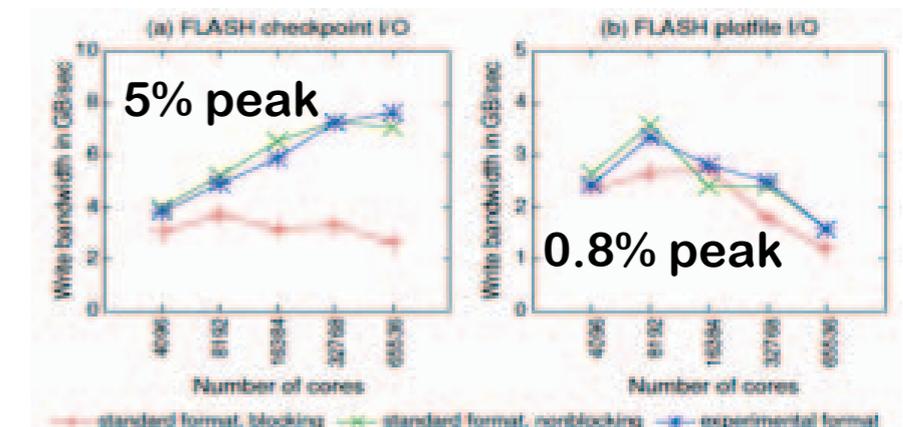
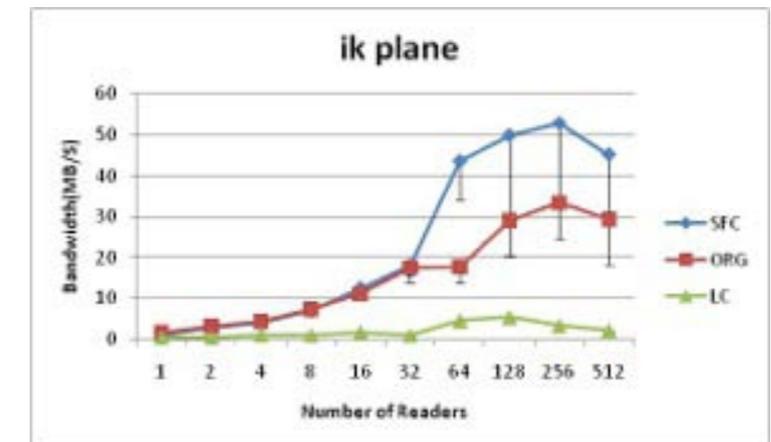
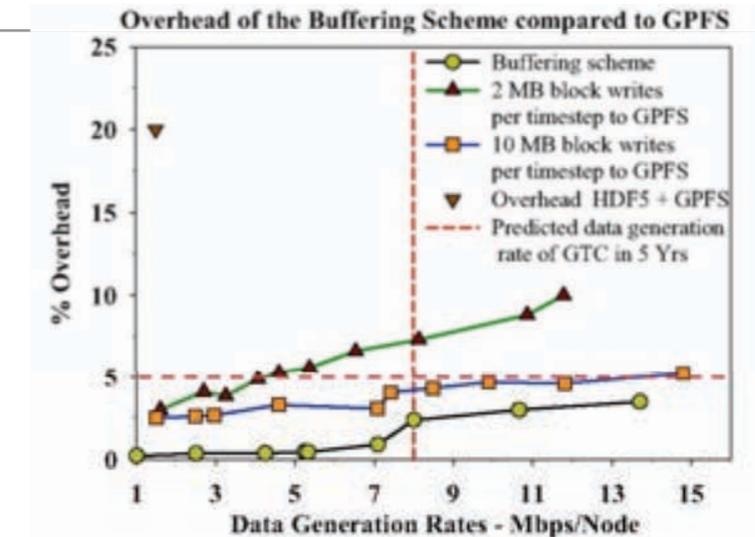
ADIOS: ADaptable I/O System

- An I/O abstraction framework
- Provides portable, fast, scalable, easy-to-use, metadata rich output with a simple API
- Layered software architecture
- Change I/O method on-the-fly
- Abstracts the API from the method used for I/O
- Provides method for injection of analysis and visualization



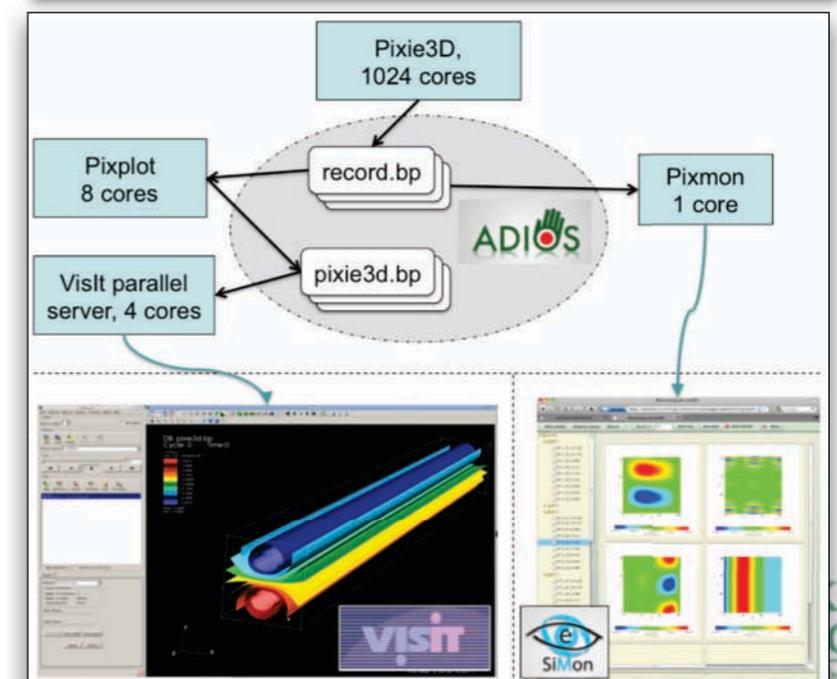
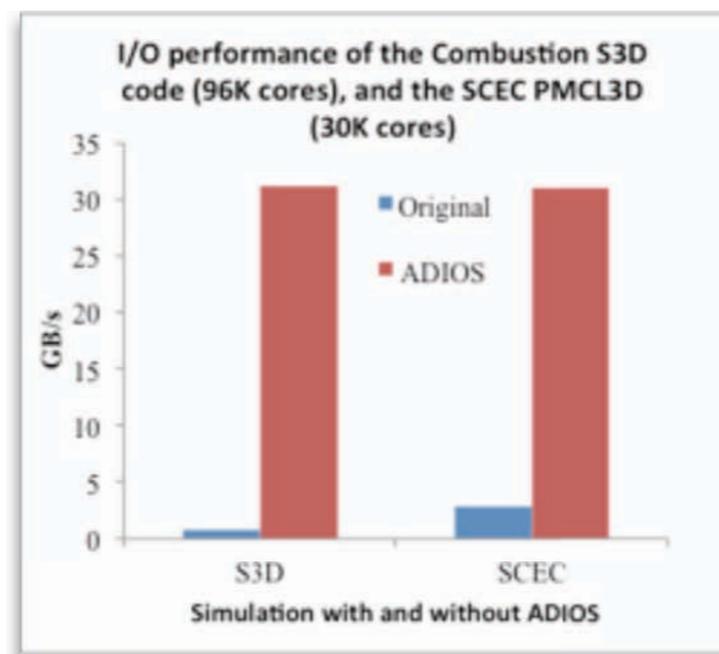
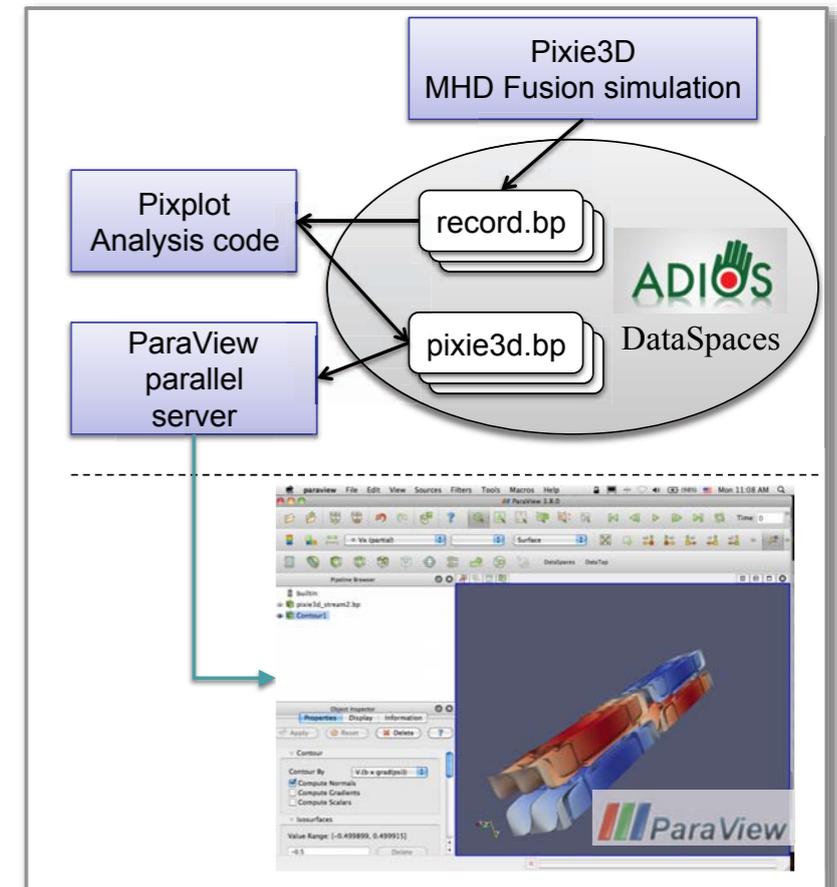
ADIOS: I/O problems and their solutions

- I/O buffering and aggregation:
 - Buffering utilizes max I/O bandwidth at writing
 - Aggregation utilizes max I/O bandwidth at reading
- Data layout optimization:
 - Layout optimization improves read performance
- Reduce synchronization and contention:
 - Log-file format and aggregation decreases lock contention and decrease network movement
 - Scheduled staging decreases impact of asynchronous I/O



Success across many application domains

- Fusion:
 - Using staging towards predictive capabilities
 - Coupled with ParaView, VisIt, and dashboards
- Combustion:
 - Orders of magnitude speed improvement for checkpoint/restart
 - In situ feature detection
- More (astro, AMR, geophysics, ...)



At the same time: Exploit fine-grained parallelism

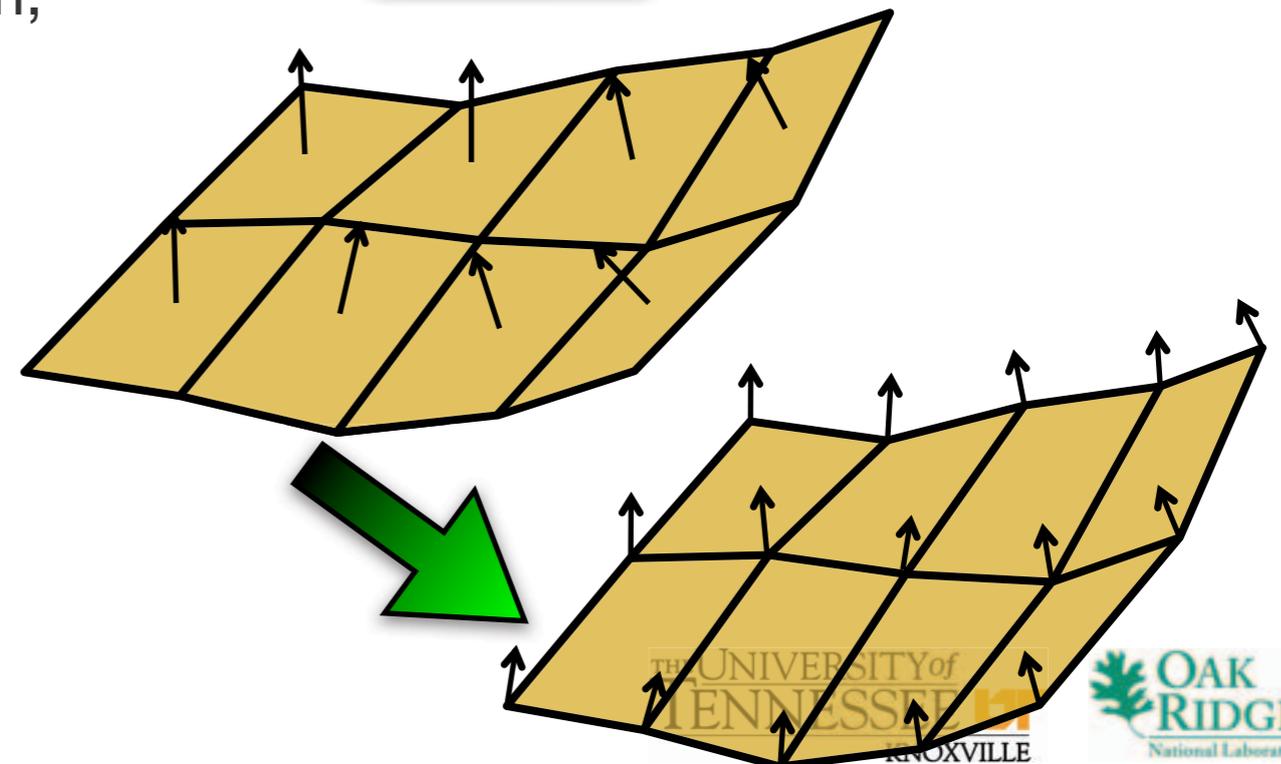
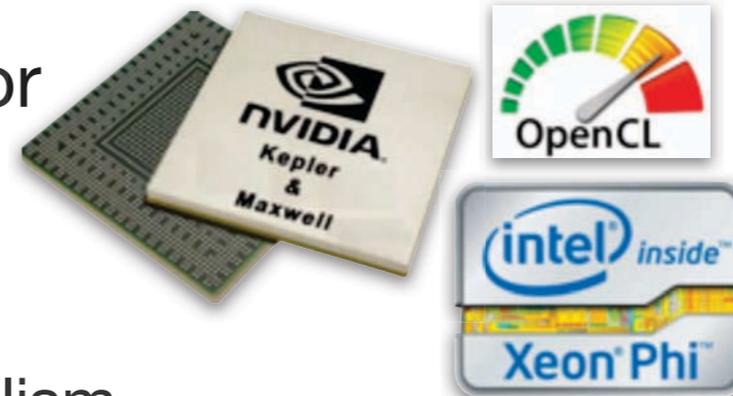
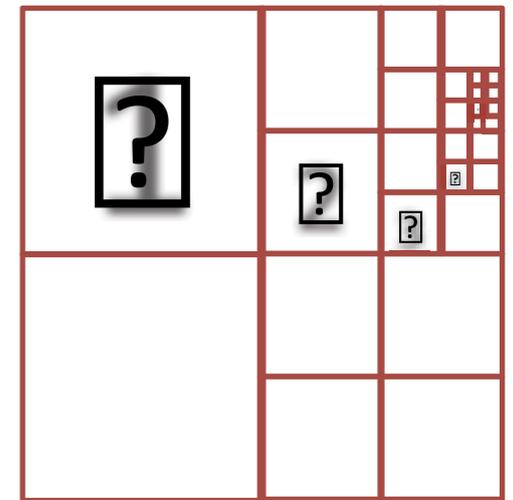
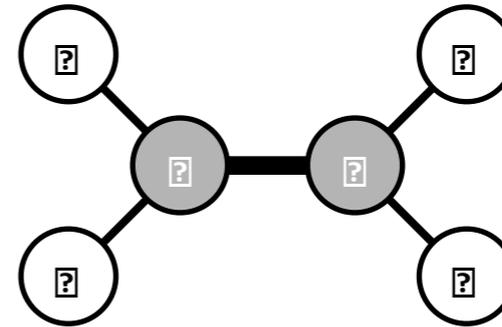
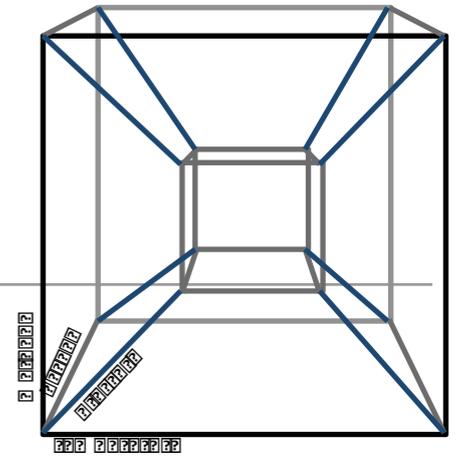
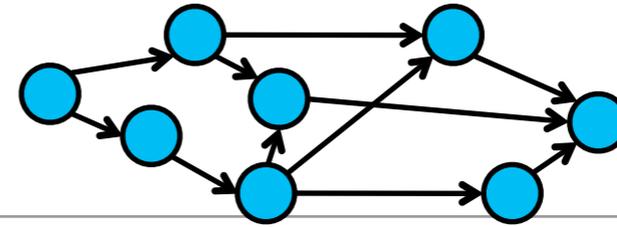
- Memory space per core may be down in the megabytes to 100s of kilobyte range
 - (Though HMC's may alleviate the memory wall and capacity.)
- Data parallelism becomes much more complex at this fine level of detail:
 - How to create and exploit search data structures
 - How to harness communication hierarchies
 - How to handle “ghost” region between data blocks
 - Iterative algorithms (e.g., streamlines) become extremely inefficient
- No analysis or visualization algorithms are written this way.
Everything must be rewritten.

Extreme-scale Analysis and Visualization Library: EAVL

- Update traditional data model to handle modern simulation codes and a wider range of data. Extend current analysis and vis tools to data that cannot be approached today.
- Investigate how an updated data and execution model can achieve the necessary computational, I/O, and memory efficiency.
- Explore methods for visualization algorithm developers to achieve these efficiency gains and better support exascale architectures. Allow easy gradual adoption into existing analysis/vis tools.

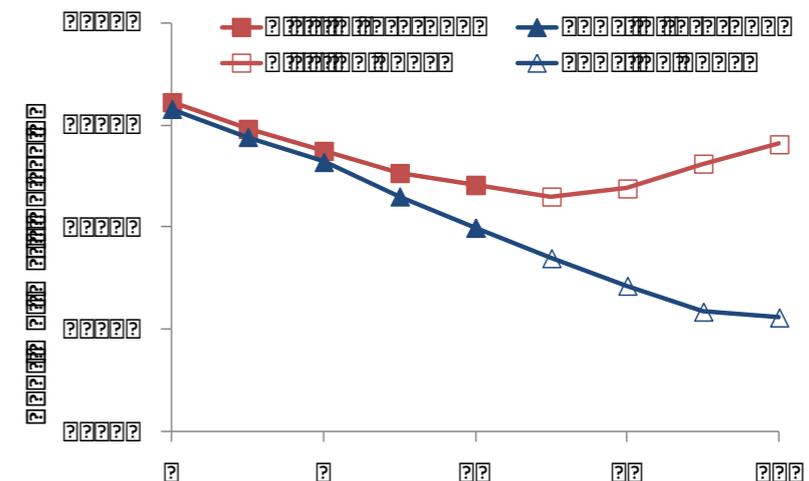
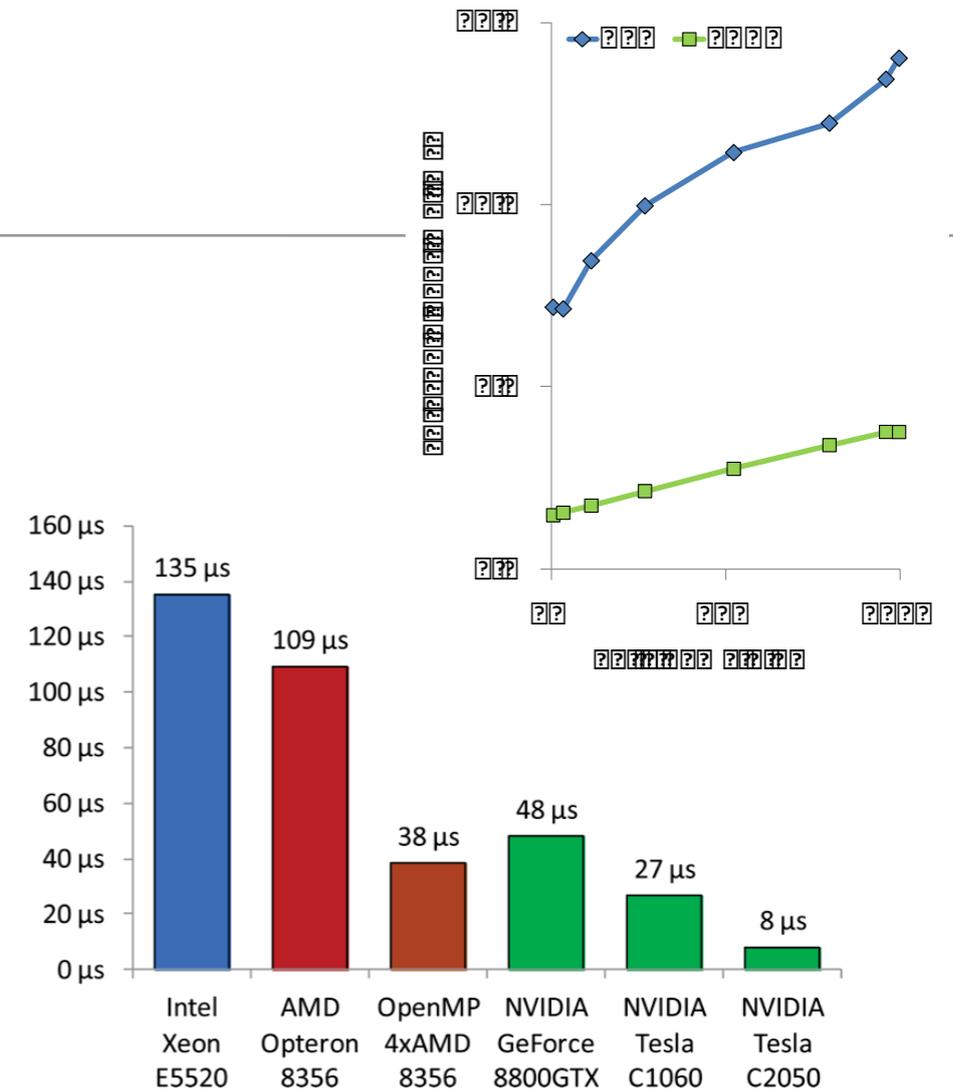
Three primary activities

- Provide a richer data model for scientific data expression
- Provide new execution models for constrained memory, advanced architectures
 - Distributed parallelism, task parallelism, thread parallelism
- Provide programming method to exploit the above: functor + iterator, library of methods



Promising results to date

- Data thresholding is more memory- and computation-efficient.
- Coordinate space transformation moves from $O(n)$ to $O(1)$.
- Structured grids are an order of magnitude more space efficient.
- Efficient execution on heterogenous architectures
- Temporal parallelism provides scalability benefits
- In-place processing provides significant memory benefits



HPC data is becoming more complex

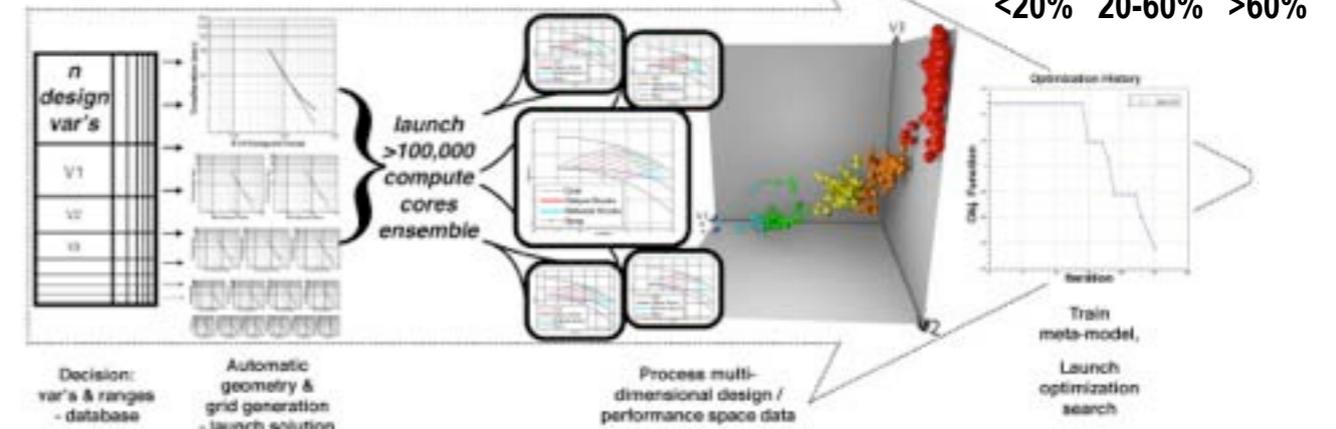
- Workflow tools are becoming more common:
 - Pegasus, Swift, HTCondor, Kepler, Eden, ...
- Ensembles are growing larger
- Parameter studies and Design of Experiments are becoming much more common
- Tracking *provenance* is important
- There are no general tools for processing large runs of simulations

The concept of a “data set” is evolving
...becoming more unstructured.

Turbo Compressor Innovation

Breakthrough aerodynamic design optimization *Science Objectives and Impact*

- Ramgen Power Systems is developing shock wave compression turbo machinery to meet DOE goals for reducing Carbon Capture and Sequestration (CCS) costs
- Complementary goal: design a gas turbine with dramatically lower costs and higher efficiency
- Compressing CO₂ to the required 100 atmospheres represents approximately 33 percent of the total cost of Sequestration



**Leadership-scale ensemble runs on Jaguar support
intelligently-driven design optimization**

OLCF Contribution

- 50x improvement in code scalability with more efficient memory utilization
- Accelerated I/O by 10x with optimizations and ADIOS
- Intelligent use of ensembles to explore parameter space using 240,000 cores

Results

- Transformed Ramgen's aerodynamic design process
- Observed designs with valuable new characteristics, from ensembles not possible without Jaguar
- Created a new workflow paradigm that accelerates design of compressors
- Accelerated computational design cycle for turbo machinery from months to 8 hours!

"The use of Jaguar has cut the projected time from concept to a commercial product by at least two years and the cost by over \$4 million,"
-- Ramgen's CEO and Director Doug Jewett.

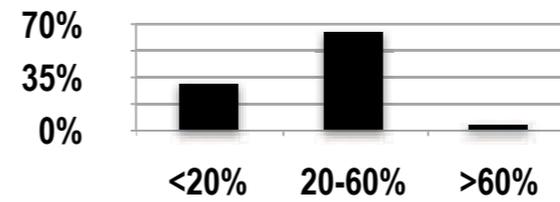
Nuclear Physics

The Limits of the Nuclear Landscape

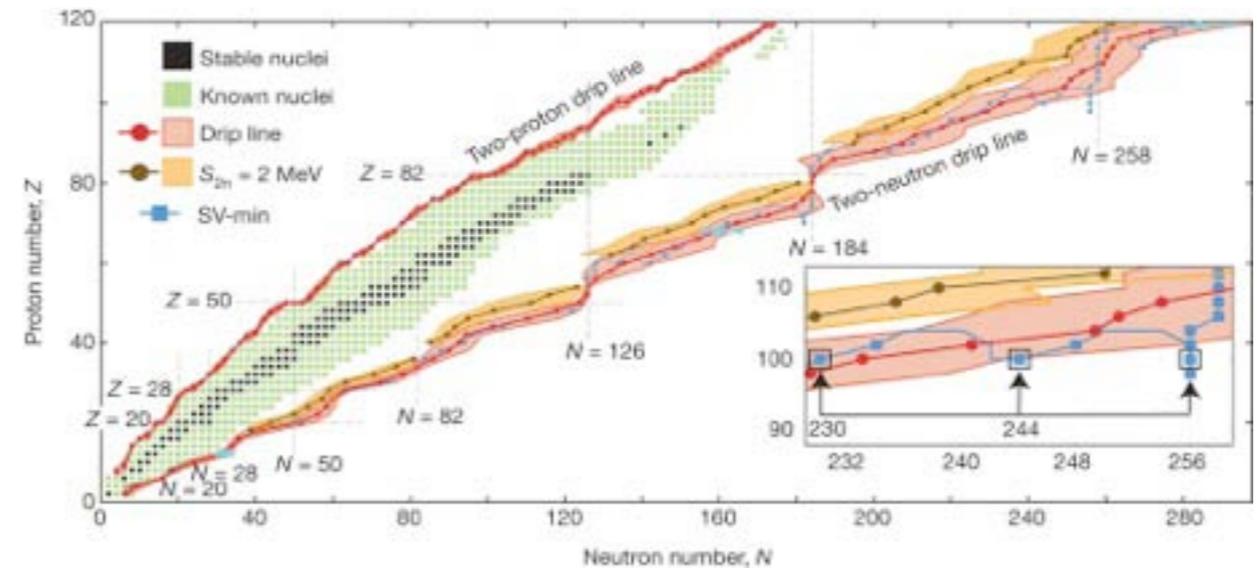
Science Objectives and Impact

- Predict how many protons and neutrons can be bound within a nucleus.
- Identify the nuclear drip lines that denote the limits of nuclear binding.
- Project advances toward the vision of “designer nuclei,” with uses ranging from potential cancer treatments to a better understanding of superconductivity.

Compute Usage



INCITE Project
James Vary, Iowa State University
Allocated hours: 37M
Used hours: 60M



Map of bound even-even nuclei as a function of Z and N . Shown are stable nuclei (black), radioactive nuclei (green), mean drip lines with uncertainties (red), and two-neutron separation line (blue). The inset shows the irregular behavior of the two-neutron drip line around $Z = 100$.

OLCF Contribution

- Custom code (co-authored by OLCF staff member Hai Ah Nam) used density functional theory to solve the nuclear structure, which involved a large algebraic non-linear eigenvalue problem.
- Each ensemble of nuclei took about two hours to calculate on 224,256-processors in Jaguar system.
- Each run evaluated about 250,000 possible nuclear configurations.

Science Results

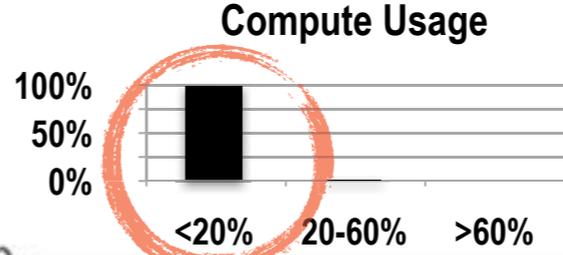
- Accurate calculation of the number of bound nuclei in nature.
- Calculations predicted about 7,000 possible combinations of protons and neutrons allowed in bound nuclei with up to 120 protons.
- Several leading models of nuclear interaction shown to be largely in agreement.

J. Erier, et al., "The Limits of the Nuclear Landscape" Nature, June 2012.

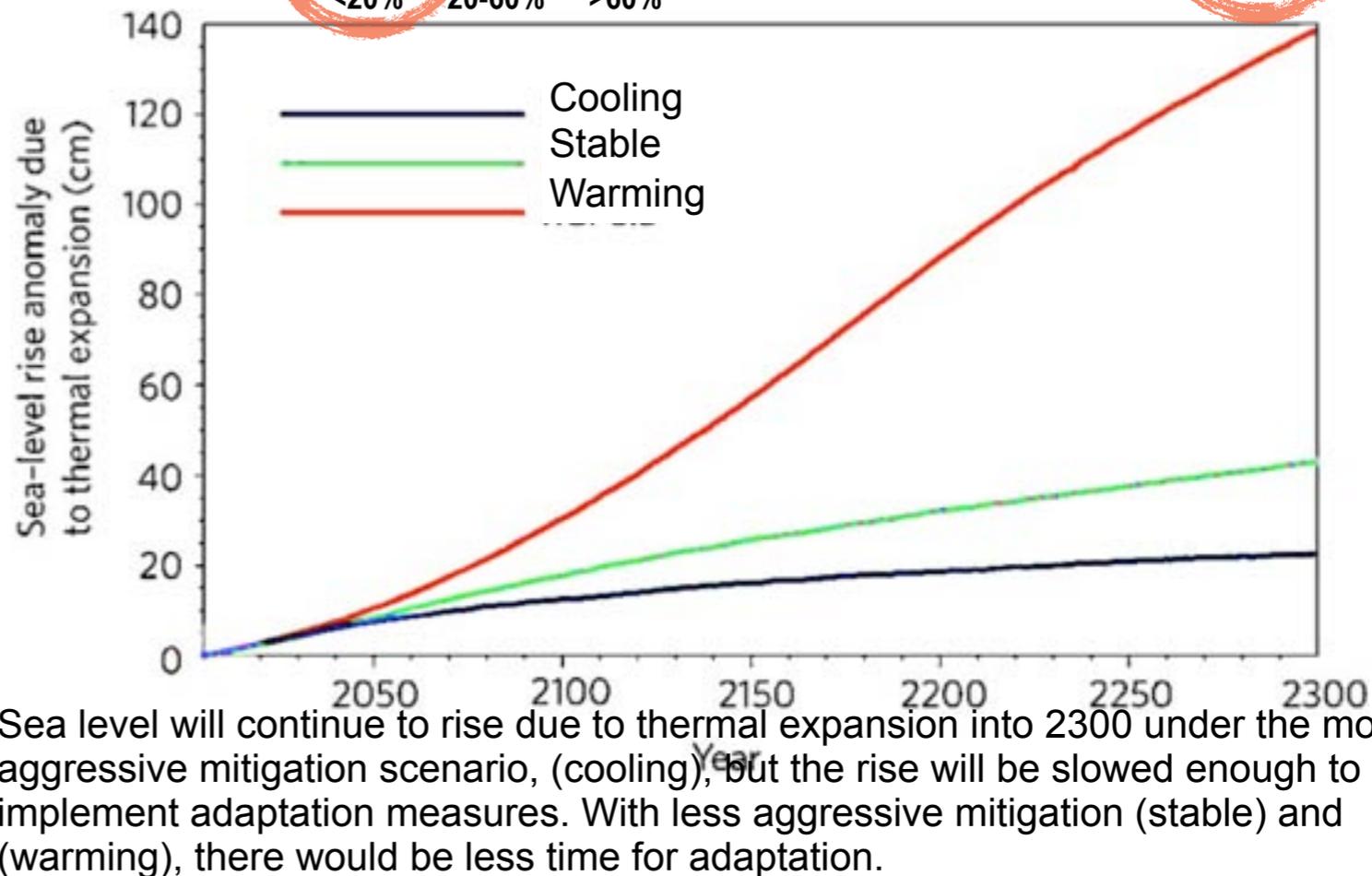
Sea-Level Rise Inevitable

Aggressive Greenhouse Gas Mitigation Can Help Slow Rate *Science Objectives and Impact*

- Determine how much sea level will rise based on three climate-change mitigation scenarios
- Examine mechanisms involved in sea-level rise, including ice melt and thermal expansion of water
- Quantify sea-level rise in response to mitigation strategies



INCITE Project
Warren Washington, NCAR
Allocated hours: 56M
Used hours: 49M



Sea level will continue to rise due to thermal expansion into 2300 under the most aggressive mitigation scenario, (cooling), but the rise will be slowed enough to implement adaptation measures. With less aggressive mitigation (stable) and (warming), there would be less time for adaptation.

OLCF Contribution

- Used CCSM4 to conduct simulation ensembles for each mitigation scenario
- Simulation of 15 separate solutions, 4,500 total simulation years

Science Results

- Sea level will continue to rise in the future, even with aggressive CO₂ mitigation scenarios.
- Aggressive mitigation measures strongly affect the rate of increase
- Mitigation buys important time to implement adaptation measures for inevitable sea-level rise

G.A. Meehl, et al., "Relative outcomes of climate change mitigation...", *Nature Climate Change*, August 2012.



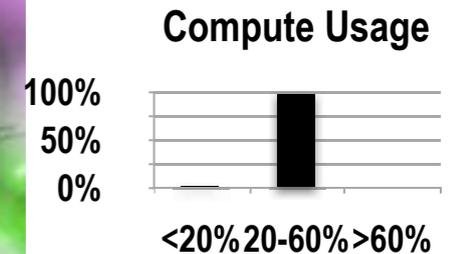
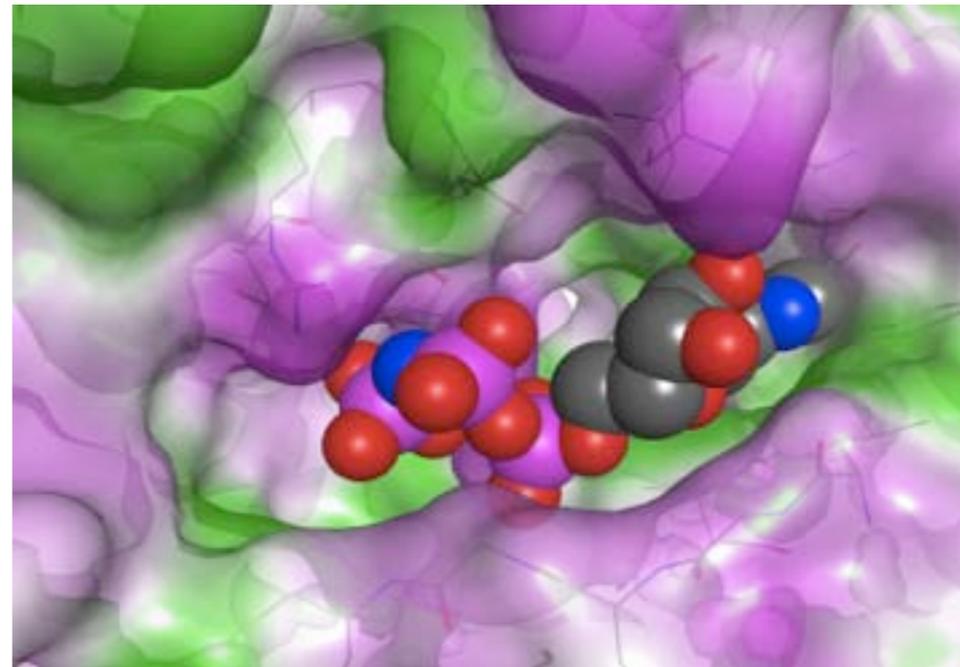
Computational and Molecular Biophysics

High-Performance Computing for Accelerated Drug Discovery

Discretionary Project
Jerome Baudry, UTK/ORNL
Allocated hours: 7M
Used hours: 8M

Science Objectives and Impact

- Test millions of compounds against a specific protein receptor using simulations.
- Find the best match between a compound and its targeted protein receptor, in hopes of creating drugs with a higher degree of specificity and less cross-reactivity, as well as exploring alternative uses for existing drugs.
- Impact: Drastically decrease the time and money spent bringing new, improved drugs to market.



Computational approaches are used to describe how molecular compounds of a drug candidate (displayed in colored spheres) bind to its specific protein receptor target. Image courtesy of Jerome Baudry, UTK & ORNL

OLCF Contribution

Director's Discretion Project

Usage between 1/1/2012–9/30/2012

Used 7.6 M hours of allocation, or 110%

Used variations of AutoDock, a General Public License, open-source software, commonly used for molecular 3D modeling in protein-docking simulations.

Science Results

- Successfully screened 2 million compounds against a targeted receptor in a matter of days, as opposed to months using computing clusters or longer with test-tube methods.
- Allowed scientists to account for specific binding in protein receptors as well as structural variations within the receptor.
- Enabled search of a vast library of molecular compounds to find alternative uses for existing drugs (i.e. “repurposing”).

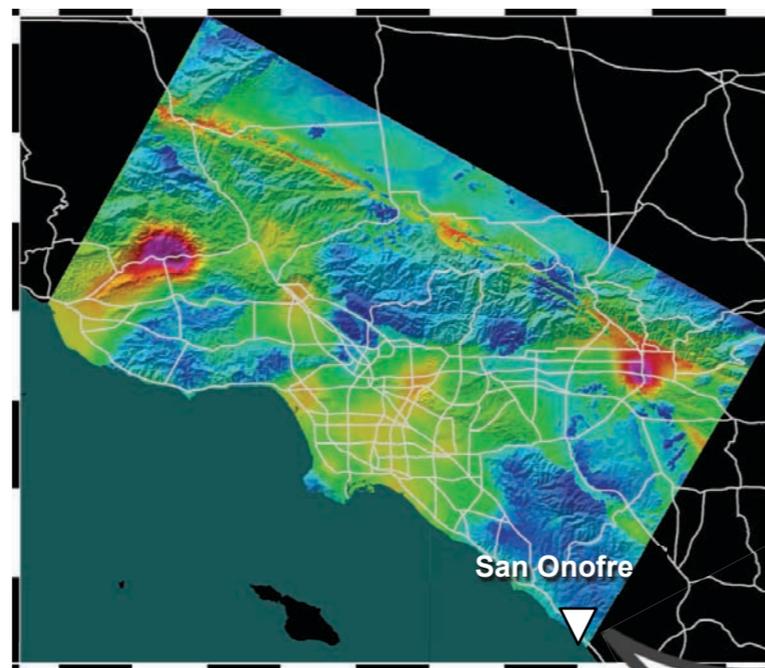
S. Ellingson. “Accelerating virtual high-throughput ligand docking,”
Presented at HPDC12, 2012 .



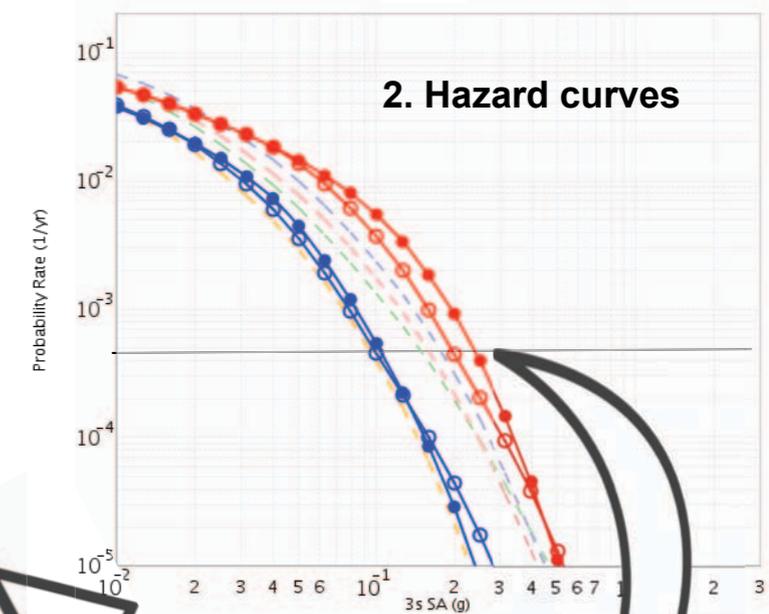
Southern California Earthquake Center (SCEC)

CyberShake workflow

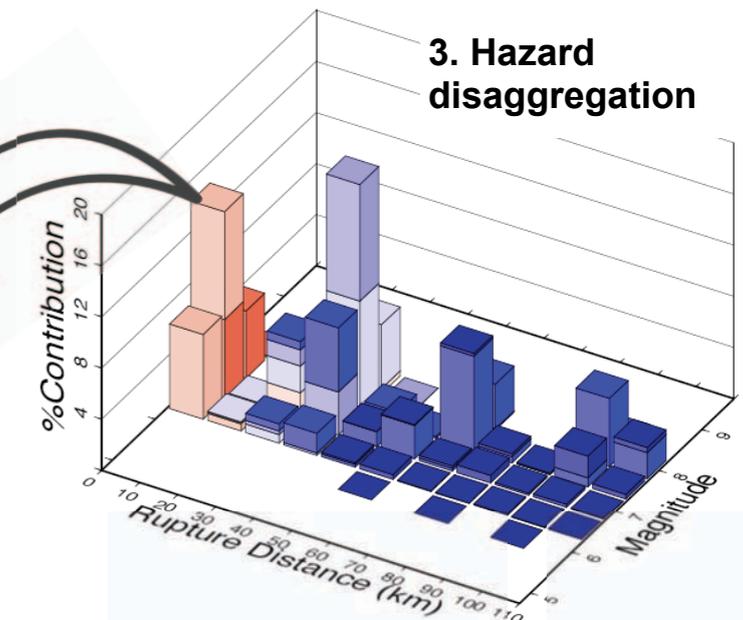
A single large HPC simulation leads to millions of serial analysis jobs



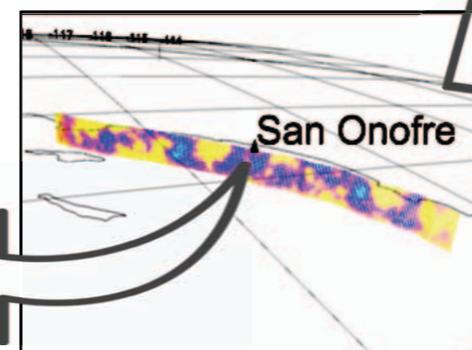
1. Hazard map



2. Hazard curves

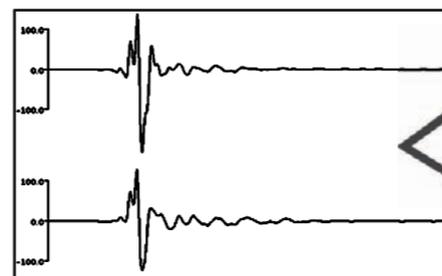


3. Hazard disaggregation



4. Rupture model

5. Seismograms



What does the future hold for big data from HPC?

- Evolving nature of data sets means new research and development directions. We have no general tools to handle these!
- Workflow tools, ensemble processing, and parameter studies ->
 - Need to store “meta” information about simulation results.
 - Promising: Database storage and query methods
- Scalability limitations of traditional files ->
 - Need to store scientific information with new methods.
 - Promising: Object stores and access methods
- Fusion of HPC data and observational data ->
 - Need for algorithms and methods to expand into new data types.
- New data types and descriptions ->
 - Requirement for processing beyond simply distributed-parallel methods
 - Promising: “business analytics” methods for unstructured data

We've got a lot of work to do!

Thanks for your time!