Exascale panel

DOE Computer Graphics Forum
Asheville, NC 27 April 2011

Question

What does visualization on an exascale machine look like?

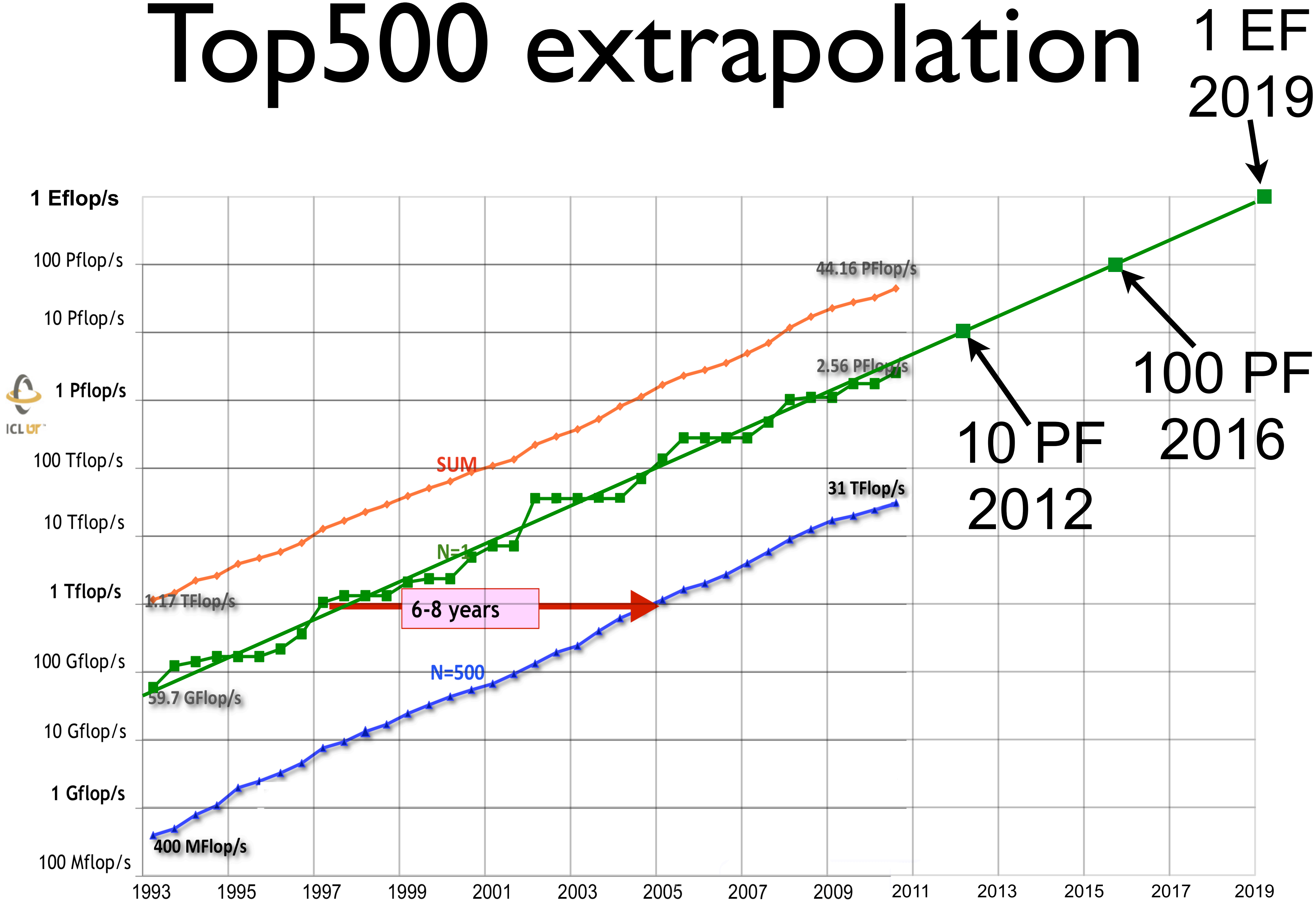
Panelists

David Rogers
(Sandia)

Jeremy Meredith
(ORNL)

John van Rosendale
(William & Mary)

Top500 extrapolation



“Swim lanes”

- Architecture slides borrowed from Andy White (LANL) from Spring Houston Exascale workshop.

Swim lanes affect the number of threads that the system needs to support.

“swim lane” #1

“swim lane” #2

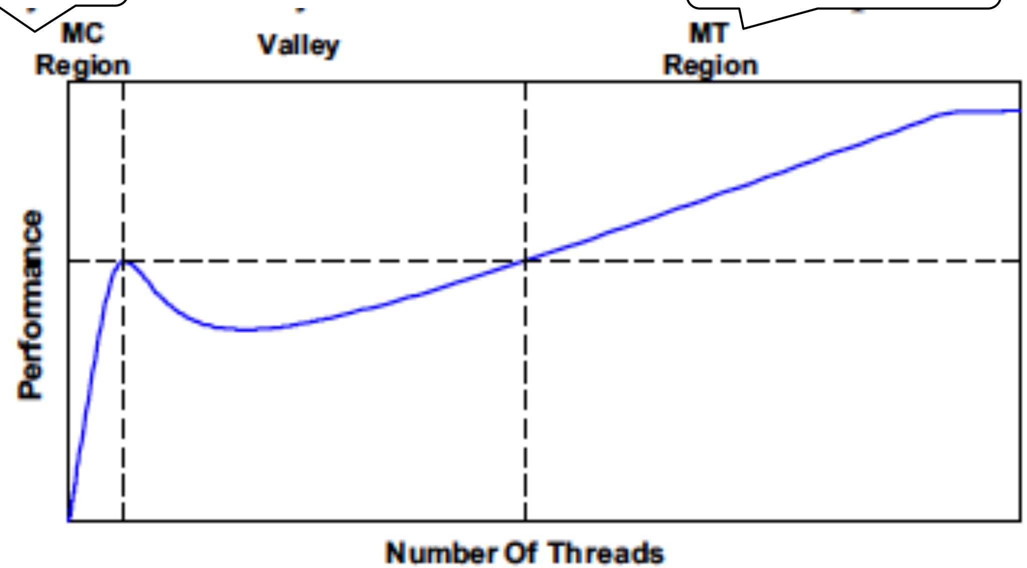
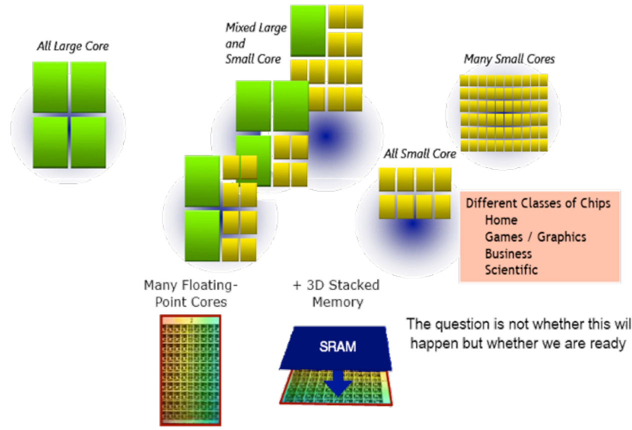


Fig. 1. Performance of a unified many-core (MC) many-thread (MT) machine exhibits three performance regions, depending on the number of threads in the workload.



There are currently two basic design points for achieving high performance in technical applications. In the future it is expected that these design points may (or may not) become more Integrated.

Many-core vs. many-thread machines: stay away from the valley, IEEE 2009

Science Partnership for
Extreme-scale Computing

System architecture targets are aggressive in schedule and scope.

System attributes	2010	"2015"		"2018"	
System peak	2 PF/s	200 Petaflop/sec		≥ 1 Exaflop/sec	
Power	6 MW	15 MW		≤ 20 MW	
System memory	0.3 PB	5 PB		64 PB	
Node performance	125 GF/s	500 GF/s	5 TF/s	1 TF/s	10 TF/s
Node memory BW (consistent with 0.4 B/F)	25 GB/s	200 GB/s	2 TB/s	400 GB/s	4 TB/s
Node concurrency	12	100	1,000	1,000	10,000
System size (nodes)	18,700	400,000	40,000	1,000,000	100,000
Node link BW (consistent with 0.1 B/F)	1.5 GB/s	50 GB/sec	0.5 TB/sec	100 GB/s	1 TB/sec
Mean time before application failure	days	≥ 24 hours		≥ 24 hours	
IO	0.2 TB/s			60 TB/s	

Jeremy Meredith
(ORNL)

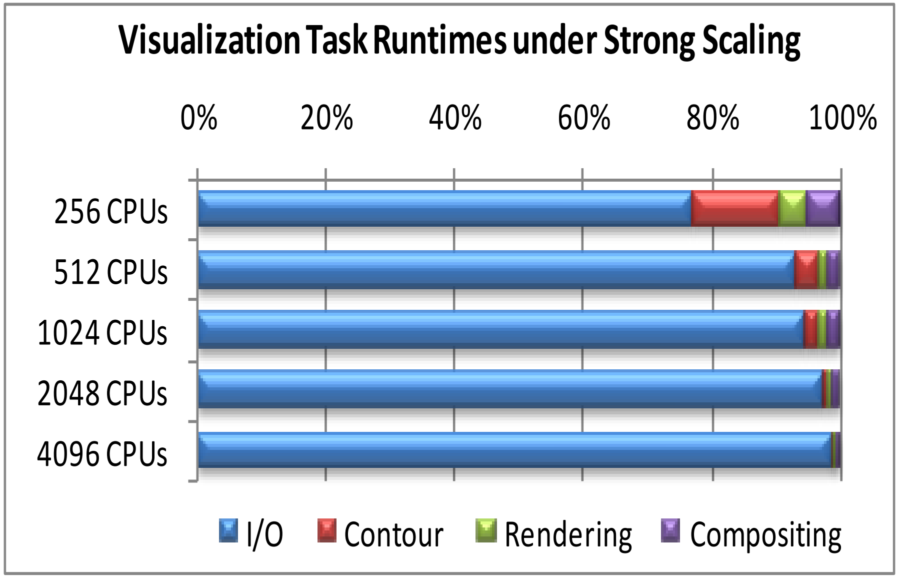
Visualization and Analysis at the Exascale:

Hardware, Software, and The In Situ Silver Bullet?

Jeremy Meredith

DOECGF 2011

Hardware: I/O



Machine	Year	Percent of FLOPS Writable to Disk	Whole-System Checkpoint
ASCI Red	1997	0.075%	300 sec
ASCI Blue Pacific	1998	0.041%	400 sec
ASCI White	2001	0.026%	480 sec
ASCI Red Storm	2004	0.035%	660 sec
ASCI Purple	2005	0.025%	500 sec
NCCS XT4	2007	0.004%	1400 sec
Roadrunner	2008	0.005%	480 sec
NCCS XT5	2008	0.005%	1250 sec
ASC Sequoia	201x	0.001%	3200 sec

- Post-processing vis and analysis is I/O bound
- Relative I/O rates are dropping
  - peak GFLOPS vs GB/sec ratio
  - total GB RAM vs GB/sec ratio

Hardware: RAM

Machine	Year	RAM Bytes / FLOPS
ASCI Red	1997	0.90
ASCI Blue Pacific	1998	1.62
ASCI White	2001	0.49
ASCI Red Storm	2004	0.92
ASCI Purple	2005	0.50
NCCS XT4	2007	0.24
Roadrunner	2008	0.08
NCCS XT5	2008	0.25
ASC Sequoia	201x	0.08

Site (machine)	Sim RAM/Core	Vis RAM/Core	Factor
TACC (Ranger vs Spur)	2.0 GB/core	8.0 GB/core	4×
LLNL (BGL vs Gauss)	0.5 GB/core	6.0 GB/core	12×
NCCS (Jaguar vs Lens)	2.0 GB/core	4.0 GB/core	2×
ALCF (Intrepid vs Eureka)	0.5 GB/core	8.0 GB/core	16×

- Memory is precious
  - Available RAM is growing more slowly than FLOPS
    - RAM per core is also shrinking
  - Dedicated visualization machines may become extinct
    - Lots of our software was designed mostly for them

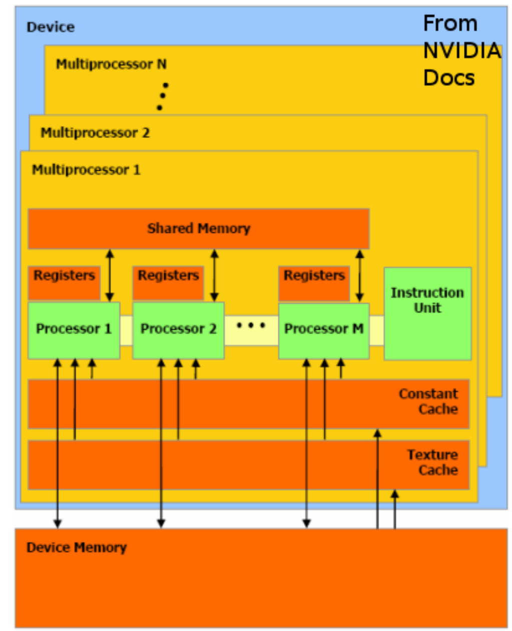
Hardware: Concurrency

Predicted Exascale Machines	
Node Concurrency	1,000 - 10,000
Number of Nodes	1,000,000 - 100,000
Total Concurrency	**1 billion**

- Massive concurrency *across* nodes
  - New types of task-level parallelism paradigms
- Massive concurrency *within* nodes
  - Thread- and data-level parallelism

Hardware: Memory Hierarchy

- With GPUs:
  - registers
  - shared memory
  - cache/texture
  - **global/device**
- CPUs:
  - registers
  - L1/L2/L3 cache
- Node:
  - DRAM
  - **NVRAM (SSD)**



?

In Situ Solves Everything*

- Tightly coupled in situ: share nodes between simulation and visualization/analysis codes
  - Bypasses I/O and storage limitations entirely
  - Incredibly fast; read data from DRAM
  - Prevents many cases of data loss
    - generate images faster than write data sets to disk
- Loosely coupled / concurrent visualization: vis runs simultaneously, but on different nodes
  - Mostly same benefits as above
  - Trades off network speed for increased total RAM

*Not really. A few problems remain….

- Tightly coupled in situ primarily addresses just I/O
  - It exacerbates the RAM limitations
  - Your vis/analysis code must comply with the simulation:
    - degree and type of concurrency
    - memory hierarchy
- The loosely coupled variant shifts the problem
  - Must still be compliant with sim code parallelism
  - Batch/interactive scheduling can be a nightmare
- Must typically know what you want beforehand
- Some analysis needs the entire time sequence
  - Generalized extreme value analysis, PCA
- Legal requirements for raw data archival (climate?)
- So now what?......

"In situ" does not mean making movies while your simulation executes.

- Interactive in situ:
  - VisIt and ParaView can connect *interactive* visualization and analysis to simulations
  - And perform some degree of on-the-fly steering
- In situ as a data-reduction technique:
  - e.g. S3D computing pathlines at a finer temporal resolution than saved full-res data sets
  - e.g. feature analysis to trigger actions like when to start saving more often
- Make use of the hardware features:
  - NVRAM could store key variables for all times on-node
  - Use the discrete memory hierarchy to your advantage:
    - e.g. sim runs on GPU, stages data to host RAM for analysis, I/O

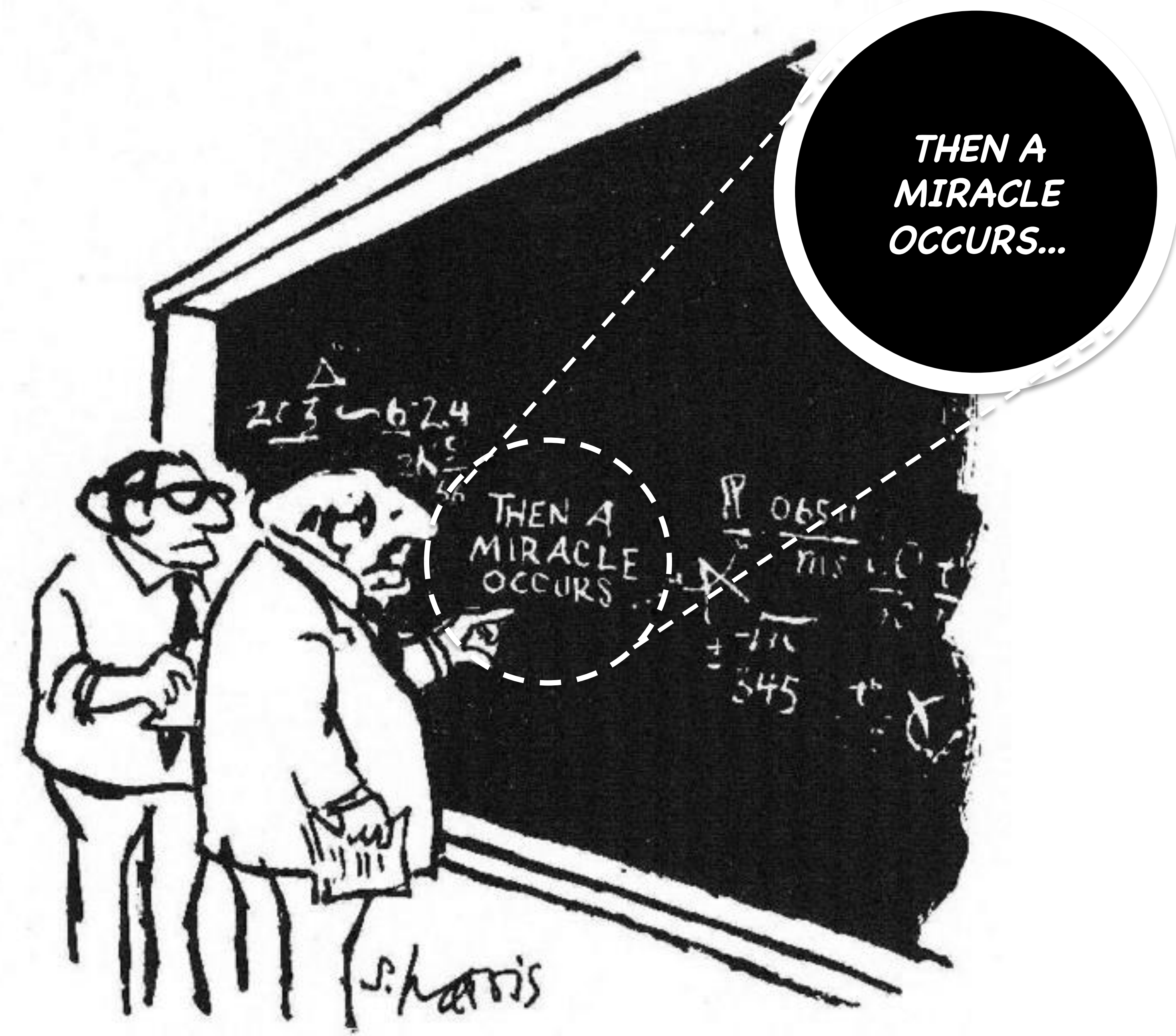
Beyond simply using in situ?

- Be smarter about I/O, RAM, Concurrency
  - Better use of the I/O pipelines we have
  - Hybrid parallelism, e.g. temporal+spatial axes simultaneously
  - Start processing other timesteps in anticipation of user actions
  - Software engineering to reduce library size
  - Write more memory efficient algorithms, e.g. in-place or limited-working-set algorithms
  - Multi-resolution techniques
  - Streaming, out-of-core
  - Data subsetting to avoid I/O and processing
- These are needed for not just in situ, but to keep post-processing analysis viable

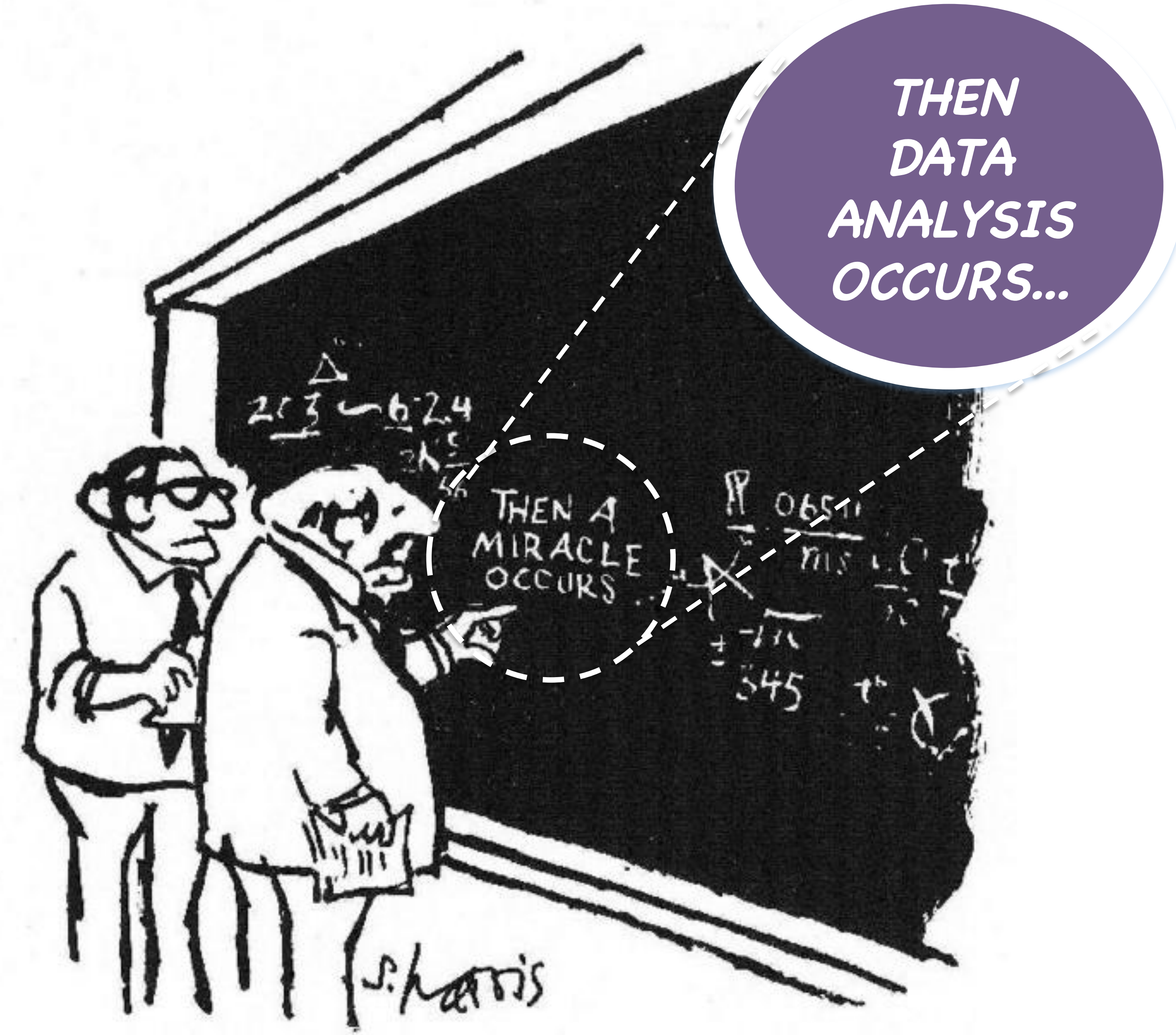
David Rogers
(Sandia)

How to Succeed at Exascale

David H. Rogers
Sandia National Labs



"I think you should be more explicit here in step two."



"I think you should be more explicit here in step two."

Summary of NNSA Workshop
From Petascale to Exascale: R&D Challenges for HPC Sim. Environments

Visualization and Data Analysis at the Exascale

A White Paper for the National Nuclear Security Administration (NNSA) Accelerated Strategic Computing (ASC) Exascale Environment Planning Process

ASC Leads: James Ahrens[1], David Rogers[2], Becky Springmeyer[3],

ASC Participants: Eric Brugger[3], Patricia Crossno[2], Ming Jiang[3], Cyrus Harrison[3], Laura Monroe[1], Bob Tomlinson[1], Dino Pavlakos[2]

ASCR Liaisons: Hank Childs[4], Scott Klasky[5], Kwan-Liu Ma[6]

Los Alamos National Laboratory[1], Sandia National Laboratories[2], Lawrence Livermore National Laboratory[3], Lawrence Berkeley National Laboratory[4], Oak Ridge National Laboratory[5], University of California at Davis[6]

Scope

The scope of our working group is scientific visualization and data analysis. Scientific visualization refers to the process of transforming scientific simulation and experimental data into images to facilitate visual understanding. Data analysis refers to the process of transforming data into an information-rich form via mathematical or computational algorithms to promote better understanding. We share scope on data management with the Storage group. Data management refers to the process of tracking, organizing and enhancing the use of scientific data. The purpose of our work is to enable scientific discovery and understanding. Visualization and data analysis has a broad scope as an integral part of scientific simulations and experiments; it is also a distinct separate service for scientific discovery, presentation and documentation purposes. Our scope includes an exascale software and hardware infrastructure that effectively supports visualization and data analysis.

Assessment of current effort with the ASC program and community

By the late nineties, it was becoming increasingly difficult to efficiently and effectively visualize the largest datasets with existing tools. This was a significant concern to the scientific simulation community, because large-scale results were being generated and needed analysis. The Advanced Simulation and Computing (ASC) Computational Systems and Software Environment (CSSE) program changed this by supporting the development of multi-platform parallel visualization applications and toolkits. This software suite includes open-source visualization applications, ParaView and VisIt, an open source visualization library, the Visualization Toolkit (VTK), and a commercial package, CEI's EnSight. These solutions use parallel and distributed computing methods to offer visualization, imaging, and rendering algorithms. The result of these efforts significantly changed how large-scale, scientific visualization is carried out. Scientists today use parallel supercomputers and commodity visualization clusters to routinely visualize terascale and petascale sized datasets that could have never been effectively analyzed. The ASC visualization and data analysis solutions have become key elements in the computational science efforts supported by many government programs including Office of Science (OSC) Advanced Scientific Computing Research (ASCR) and Biological and Environmental Research (BER), the National Science Foundation (NSF) and the Department of Defense (DOD).

Data analysis seeks to characterize data using higher-level abstractions that can be used to find associations between data elements. Data analysis approaches recognize anomalies, find correlations, categorize data, make predictions, and assist in decision making. There are a broad range of techniques used in data analysis, including statistical methods, dimensionality reduction techniques such as principal component analysis (PCA), vector space modeling, machine learning, and clustering. The choice of analysis technique depends upon not only the type of data being analyzed, but also the intent of the user. Some methods permit exploration of data with a target of discovery, others assist in hypothesis testing, and some produce descriptive summaries.

- **In-situ visualization and data analysis software infrastructure**
- **Advanced data reduction techniques including statistical sampling, compression, multi- resolution and science-based feature extraction approaches**
- **Visualization and data analysis techniques to help understand advanced exascale physics approaches**
- **Implement core visualization and data-analysis capability using a scalable parallel infrastructure**
- **Exascale visualization and data analysis hardware infrastructure**
- **Knowledge infrastructure**

- **See** https://asc.llnl.gov/**exascale**

But Here are the REALLY hard problems ahead

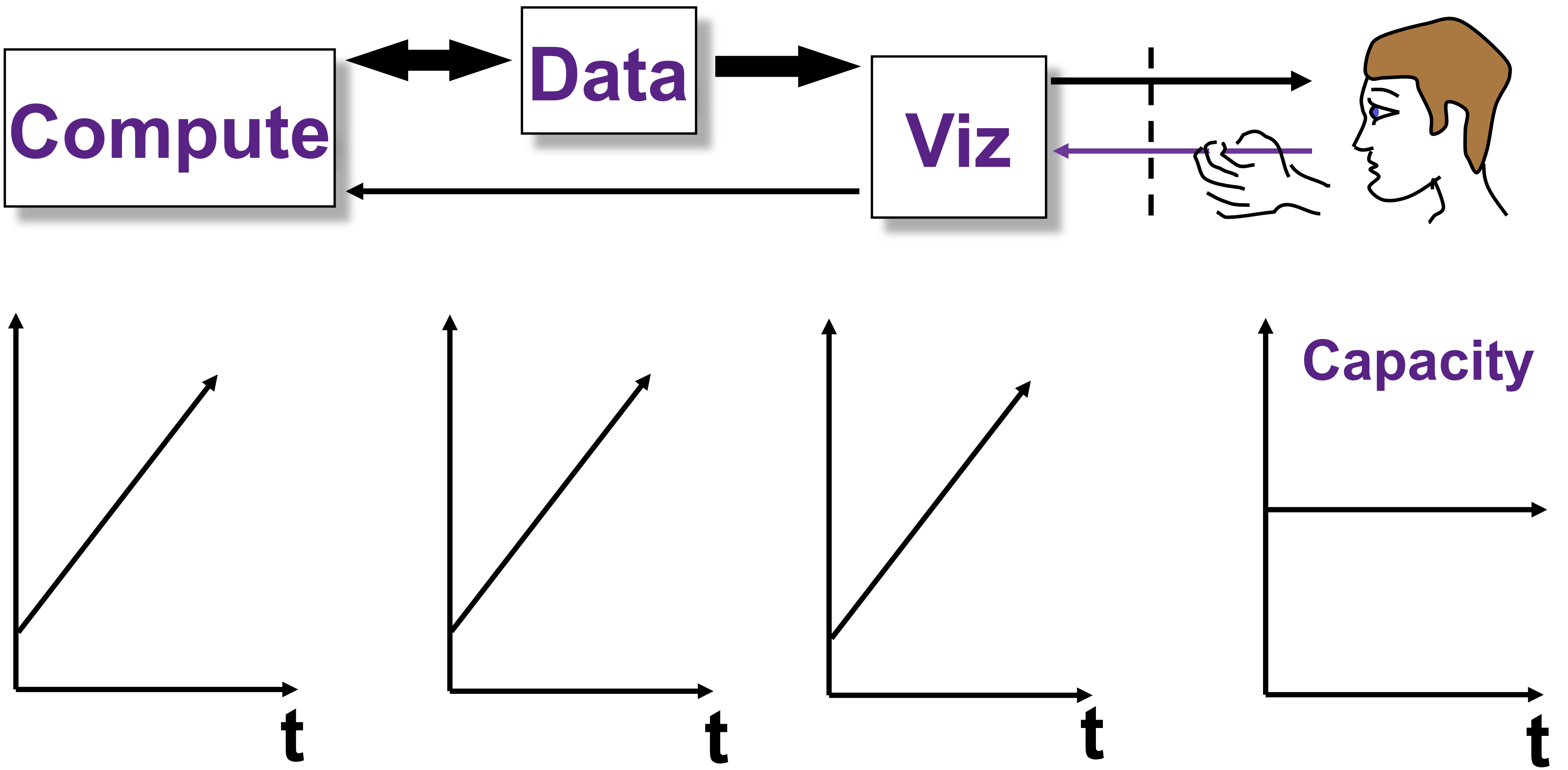
- I/O
  - Don't get me started
  - There will be no files – only queries
- Hardware
  - The HW community doesn't see this as a data-centric problem
  - You'll be parallel, multi-threaded, and power-aware, even if you're not at exascale
- Software
  - There will be no applications – only services
  - Programming model at exascale is unknown
  - Analysis and vis will have to handle resiliency
- Data
  - Don't move your data – move your artifacts
  - Provenance (the new Resiliency)
- Cognition
  - We have to think and design for continuously advancing web-based technologies
  - How will researchers think, a decade from now?
  - How will we search and retrieve insights?
  - How do we understand and debug a billion-way parallel process?
    - Machine behavior + code behavior + results

The Hardest Problem?

- Evolutionary vision for a Revolutionary problem

John van Rosendale
(William & Mary)

Exascale Vis Problem



The Problem ...

Huge datasets (and the exascale computations that created them) have little value unless we can adequately explore them and glean knowledge from them.

In situ vis ...

In situ vis. will always work (since the vis. capability scales with the capability of the HPC platform).

It makes interactive visualization much more awkward.

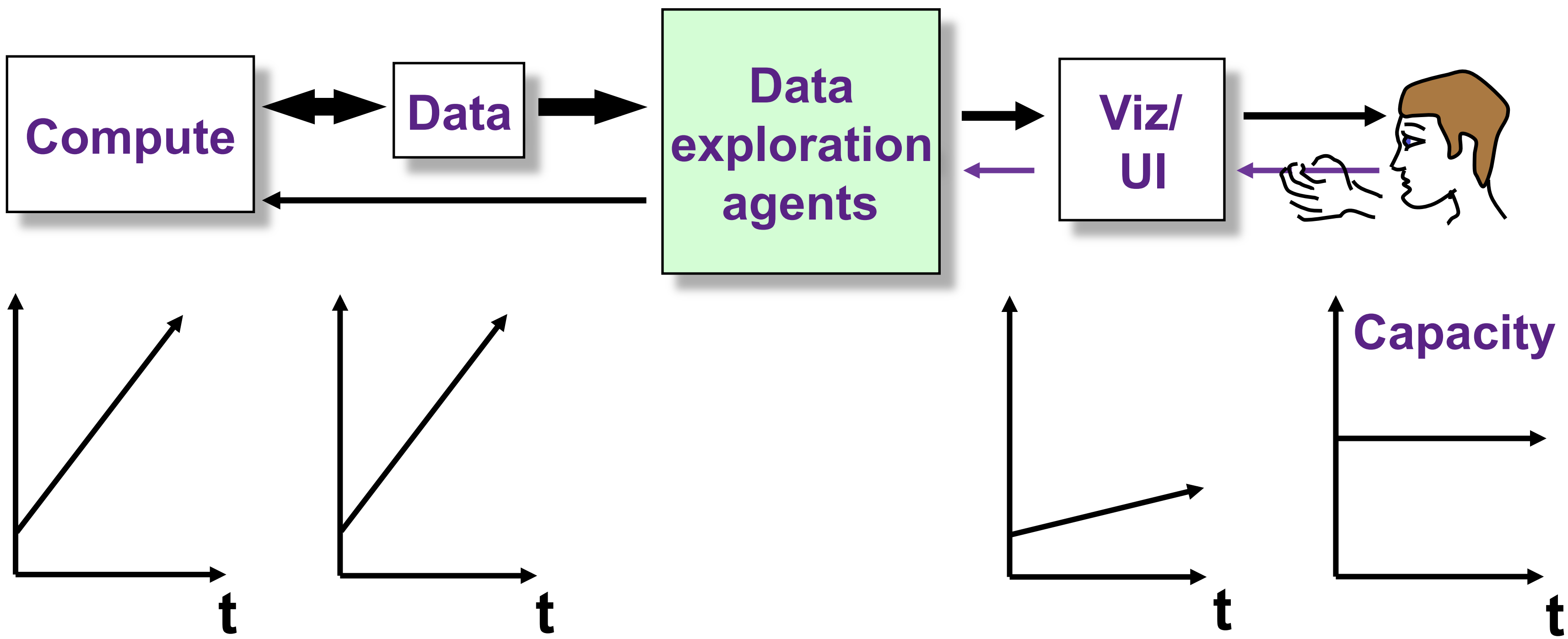
Solves the problem in the sense of speeds and feeds; the challenge of understanding exascale datasets remains.

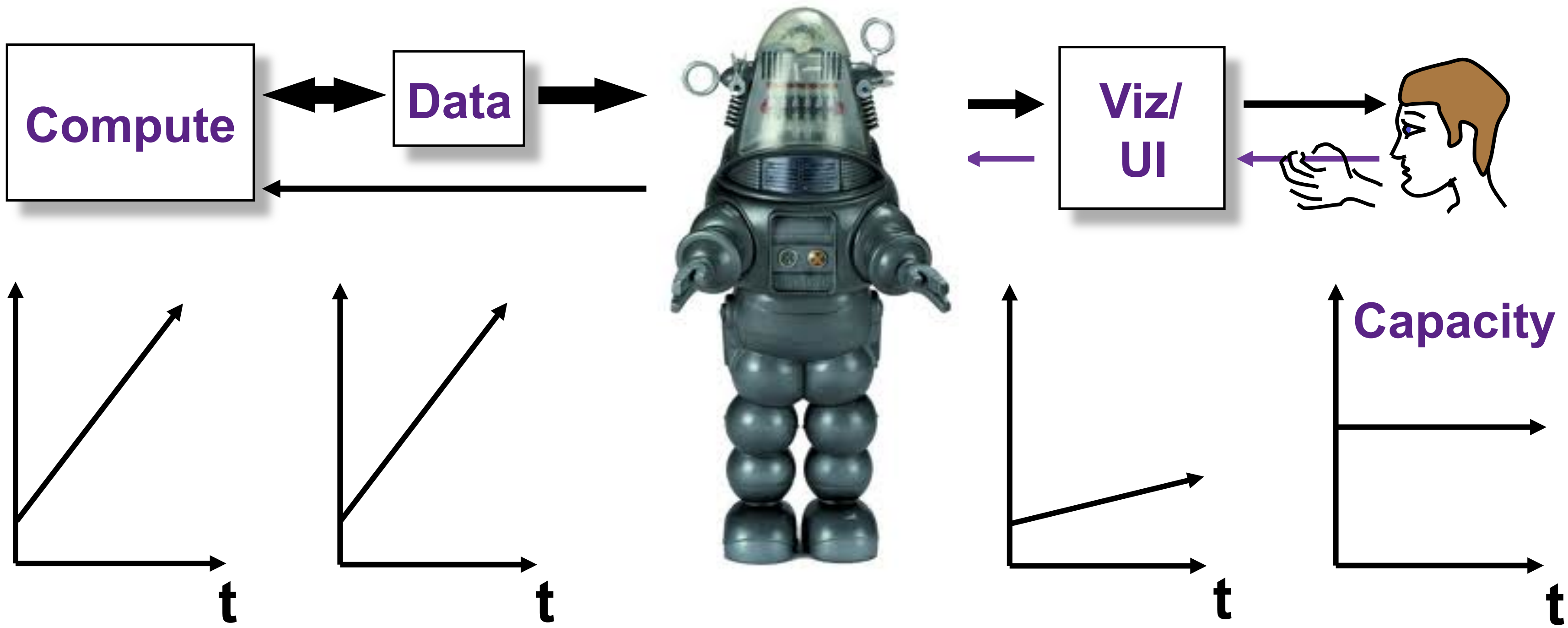
Partial solutions

Machine learning algorithms and agents to explore data.

Better user interfaces (interactive graphics, new displays, graphics algorithms, …).

There are no silver bullets.


Compute
Data
Data exploration agents
Viz/ UI
t
t
t
Capacity
t



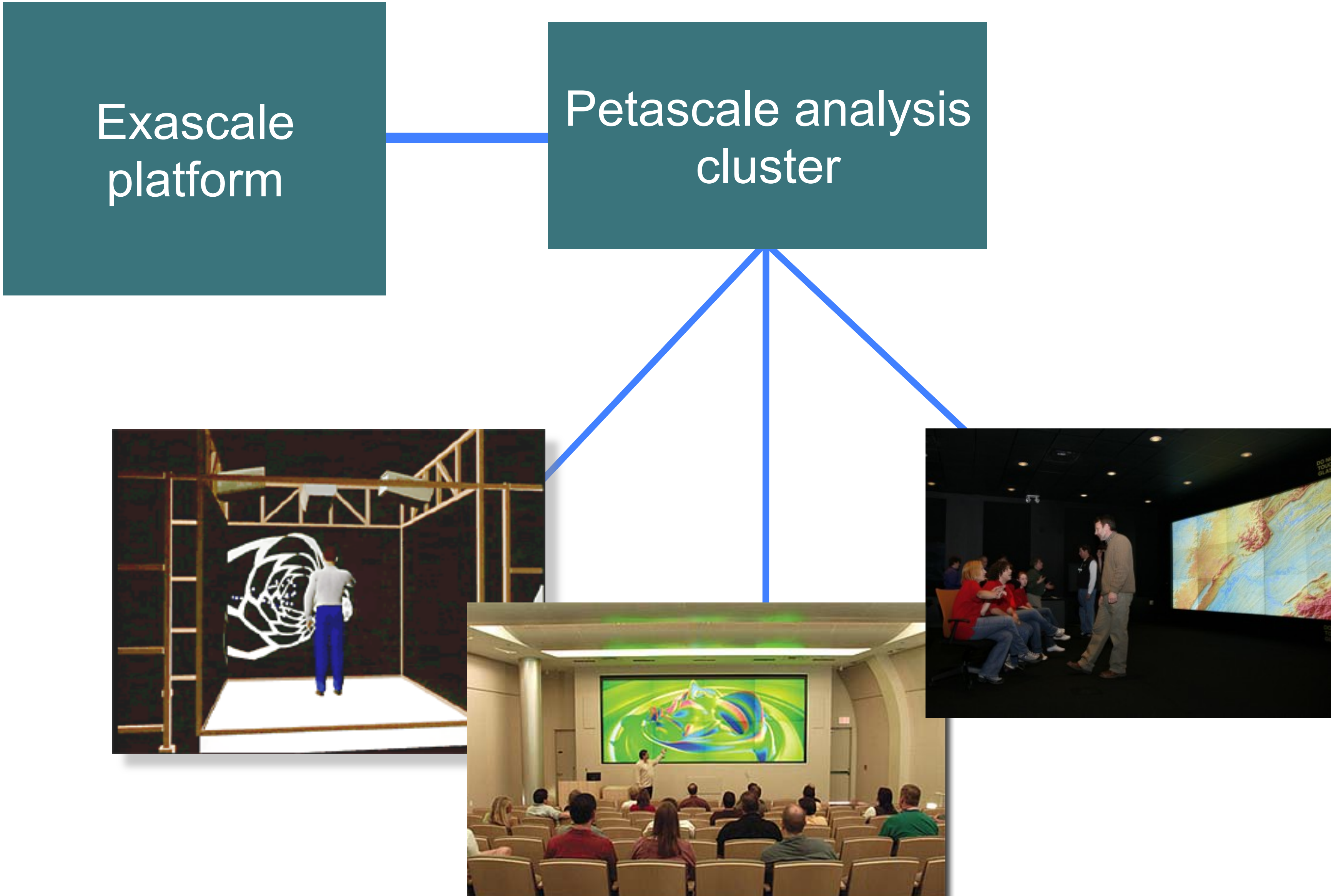
Compute
Data
Viz/
UI
Capacity
t
t
t
t

Panel question: What does visualization on an exascale machine look like?

This is (mostly) not the right question.

Better questions:

- **What's the best way to gain insight from exascale computations?**
- **Where should this computation be run?**
- **What are the relative roles of *in situ* and *out situ* vis?**


Exascale platform
Petascale analysis cluster