Computational Challenges and Opportunities for Nuclear Astrophysics

Bronson Messer

Acting Group Leader
Scientific Computing Group
National Center for Computational Sciences

Theoretical Astrophysics Group
Oak Ridge National Laboratory

Department of Physics & Astronomy
University of Tennessee

OLCF
Oak Ridge Leadership Computing Facility

U.S. DEPARTMENT OF ENERGY

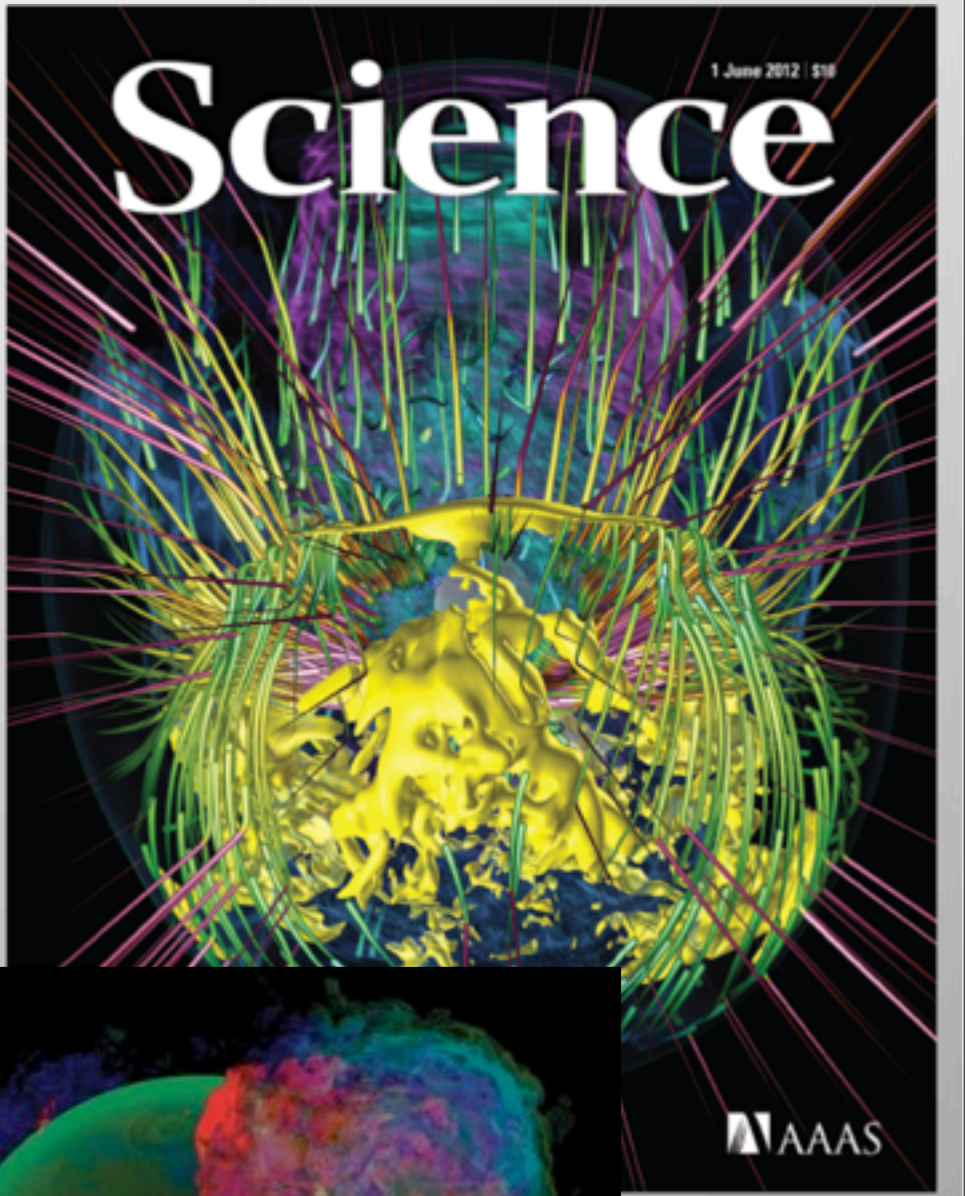
Oak Ridge National Laboratory

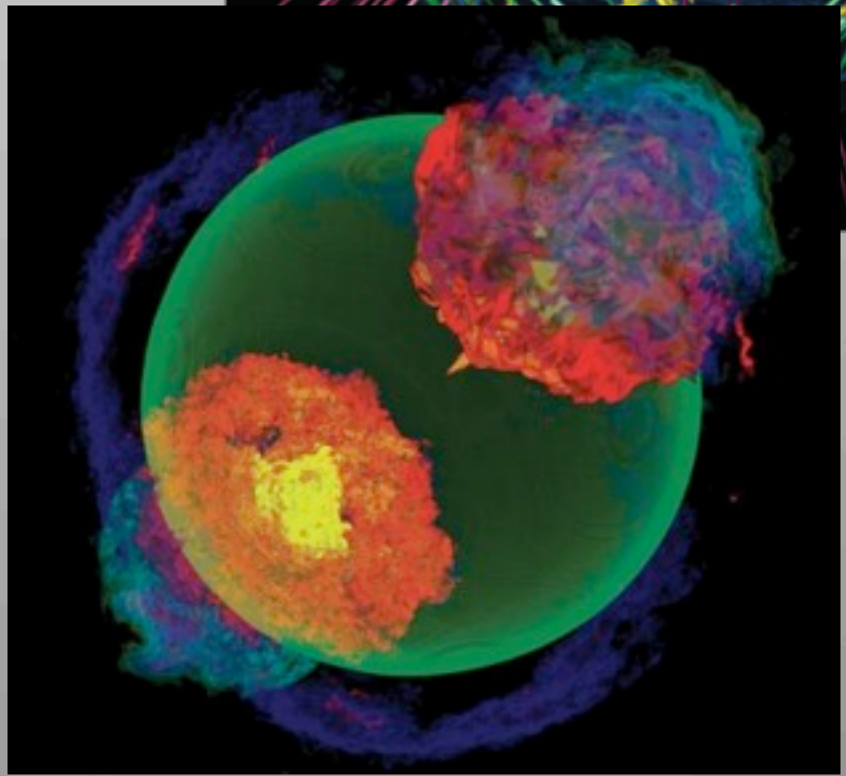
Summary

- The future is now! Computers are not getting faster from the perspective of a nuclear (astro-)physicist. They are only getting “wider.”
- The Xeon Phi/GPU/BG\Q choice is no choice at all. They are all versions of a single narrative.
- Stellar astrophysics is rife with unrealized parallelism, but architectural details and memory (i.e. cost, power) constraints will present considerable challenges. Additional support (for both “application scientists” and our CS/Math collaborators) will be required to surmount these challenges.
- Bulk-synchronous execution is a terrible way to try to exploit near-future architectures. A new programming model will require considerably more effort than a simple multi/many-core port.
- Managing large simulations is something we can barely do know, but how about managing 1000’s, 10’s of thousands, or 100’s of thousands of simulations? We should not expect to rely on solutions to be thrown over the fence from developers in other communities.

Nuclear astrophysics INCITE allocations from 2010 - present

- Average number of cpu-hours/project - 152 M
- In aggregate, just less than 10% of the total available each year from 2010 - 2012
  - Allocations at NERSC are also above-average in size
- This excellent record is due to
  - the formulation of large, important problems
  - demonstrated ability to efficiently exploit the largest computational platforms
- Will this trend continue to the exascale? Can we continue to solve big problems efficiently?





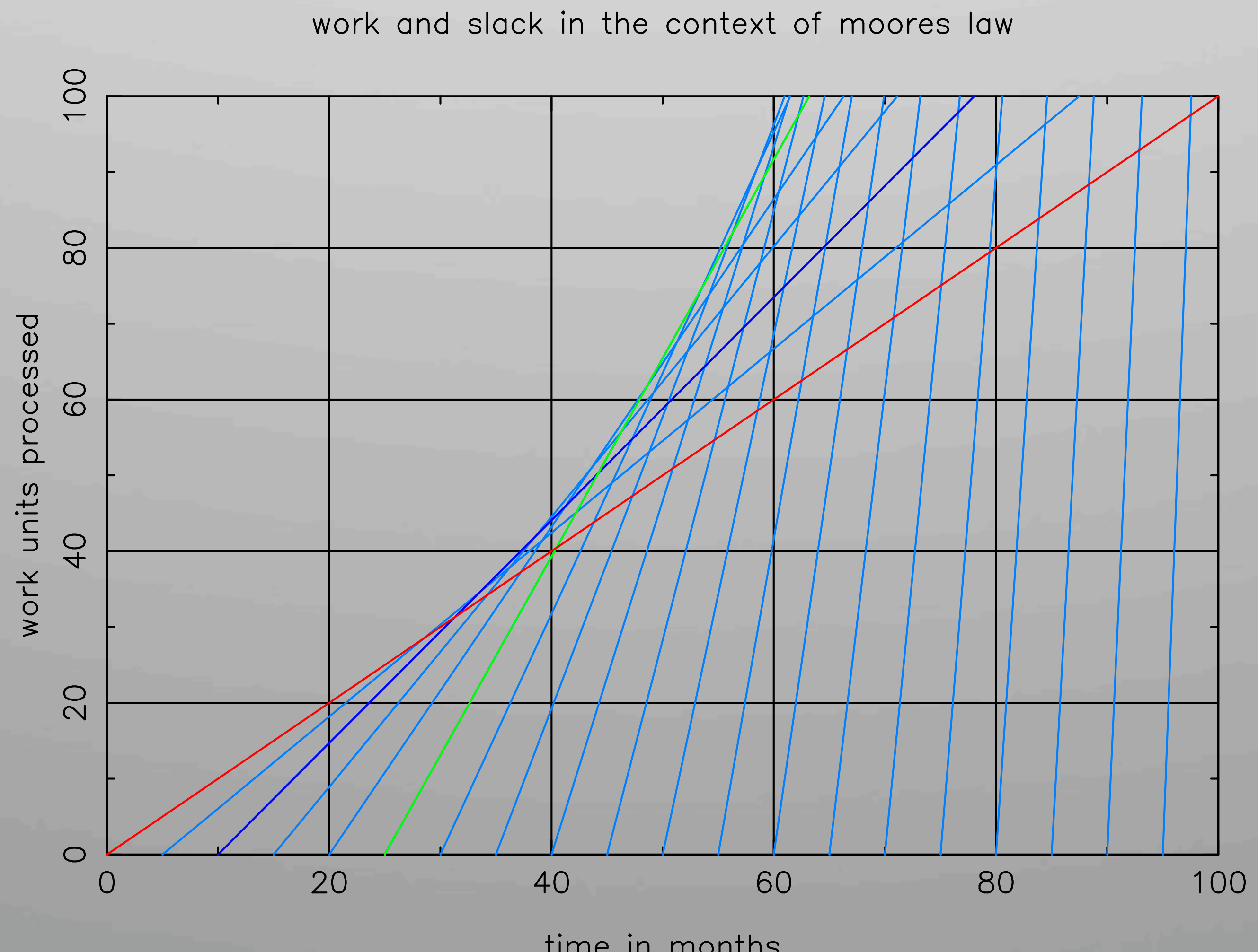
The Effects of Moore's Law and Slacking [1] on Large Computations

Chris Gottbrath, Jeremy Bailin, Casey Meakin, Todd Thompson,
J.J. Charfman
Steward Observatory, University of Arizona

[1] This paper took 2 days to write

Abstract

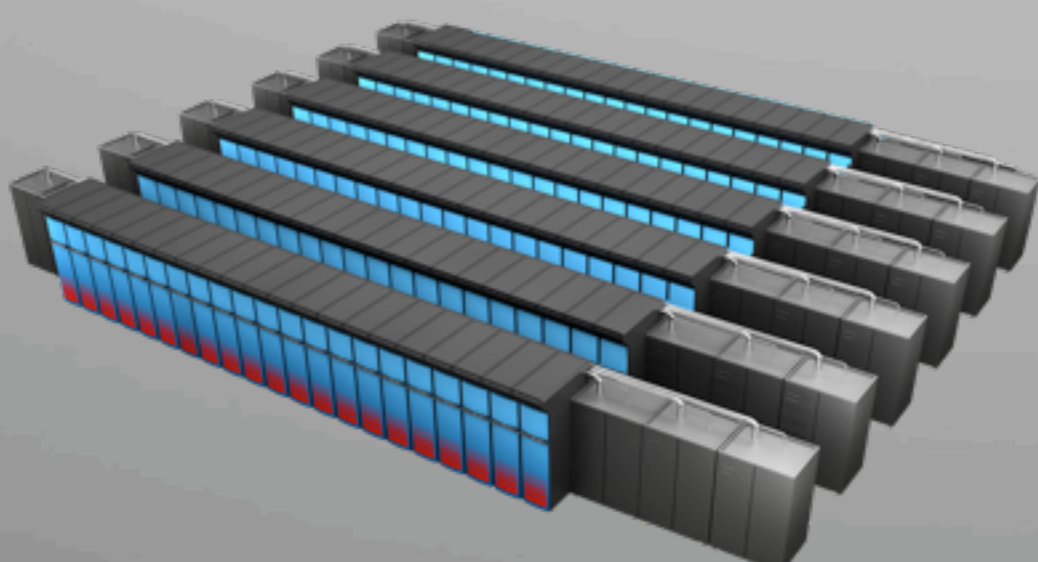
We show that, in the context of Moore's Law, overall productivity can be increased for large enough computations by 'slacking' or waiting for some period of time before purchasing a computer and beginning the calculation.



astro-ph/9912202

ORNL's "Titan" System

- Upgrade of existing Jaguar Cray XT5
- Cray Linux Environment operating system
- Gemini interconnect
- 3-D Torus
- Globally addressable memory
- Advanced synchronization features
- <u>AMD Opteron 6200 processor (Interlagos)</u>
- New accelerated node design using NVIDIA multi-core accelerators
  - **2011: 960 NVIDIA M2090 "Fermi" GPUs ("titandev")**
  - **2012: 20 PF - NVIDIA "Kepler" GPUs**
- 20 PFlops peak performance
  - Performance based on available funds
- 600 TB DDR3 memory (2x that of Jaguar)

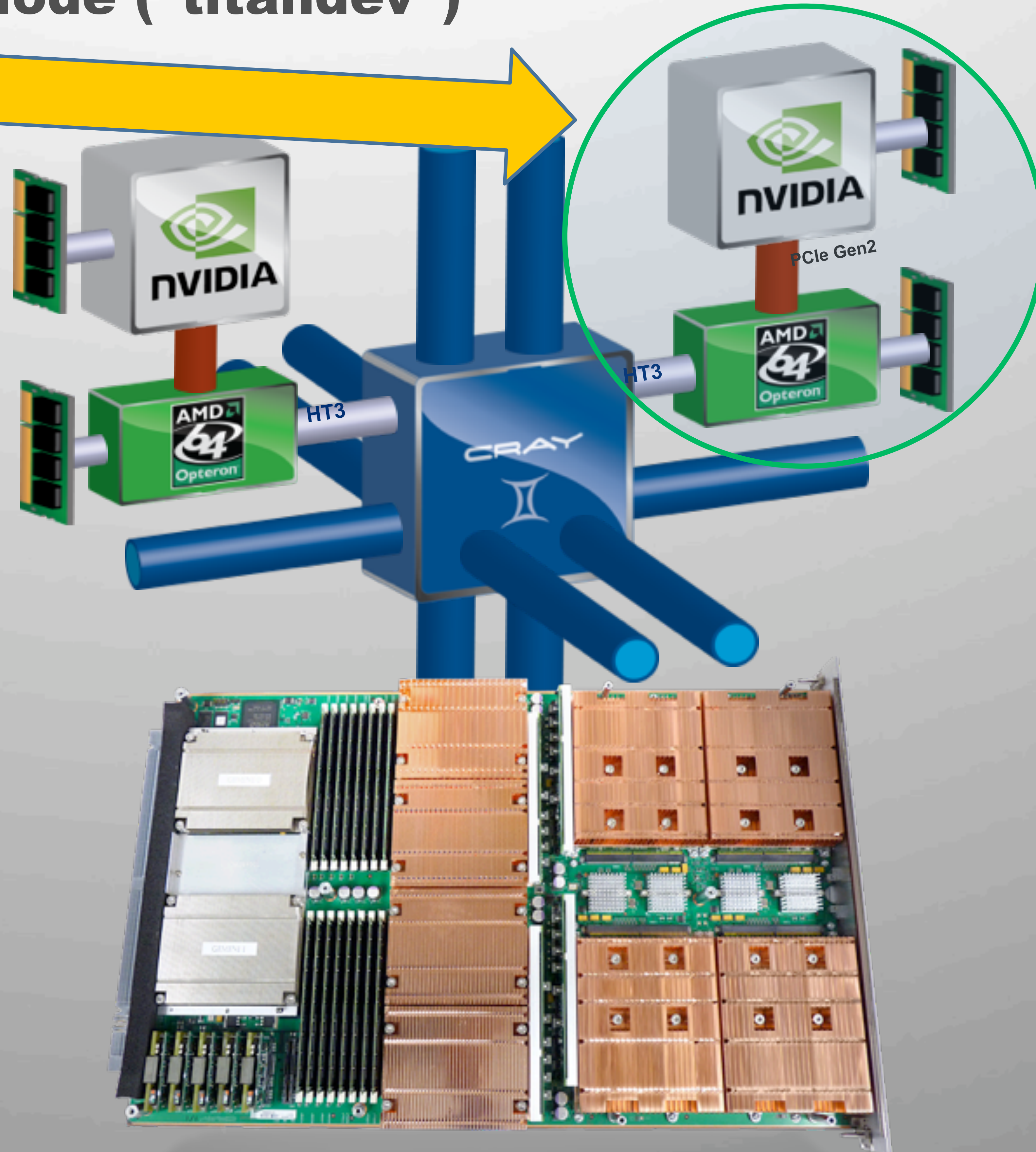


Titan Specs	
Compute Nodes	18,688
Login & I/O Nodes	512
Memory per node	32 GB + 6 GB
NVIDIA "Fermi" (2011)	665 GFlops
# of Fermi chips	960
NVIDIA "Kepler" (2012)	>1 TFlops
Opteron	2.2 GHz
Opteron performance	141 GFlops
Total Opteron Flops	2.6 PFlops
Disk Bandwidth	~ 1 TB/s

OLCF

Cray XK6 Compute Node ("titandev")

XK6 Compute Node Characteristics
AMD Opteron 6200 "Interlagos" 16 core processor @ 2.2GHz
Tesla M2090 "Fermi" @ 665 GF with 6GB GDDR5 memory
Host Memory 32GB 1600 MHz DDR3
Gemini High Speed Interconnect
Upgradeable to NVIDIA's next generation "Kepler" processor in 2012
Four compute nodes per XK6 blade. 24 blades per rack



Multicore CPU
+
Many-Core GPU

NERSC

This year

NERSC-6 Grace “Hopper”

Cray XE6

Performance
- 1.2 PF Peak
- 1.05 PF HPL (#5)

Processor
- AMD MagnyCours
- 2.1 GHz 12-core
- 8.4 GFLOPs/core
- 24 cores/node
- 32-64 GB DDR3-1333 per node

System
- Gemini Interconnect (3D torus)
- 6392 nodes
- 153,408 total cores

I/O
- 2PB disk space
- 70GB/s peak I/O Bandwidth

Franklin - Cray XT4

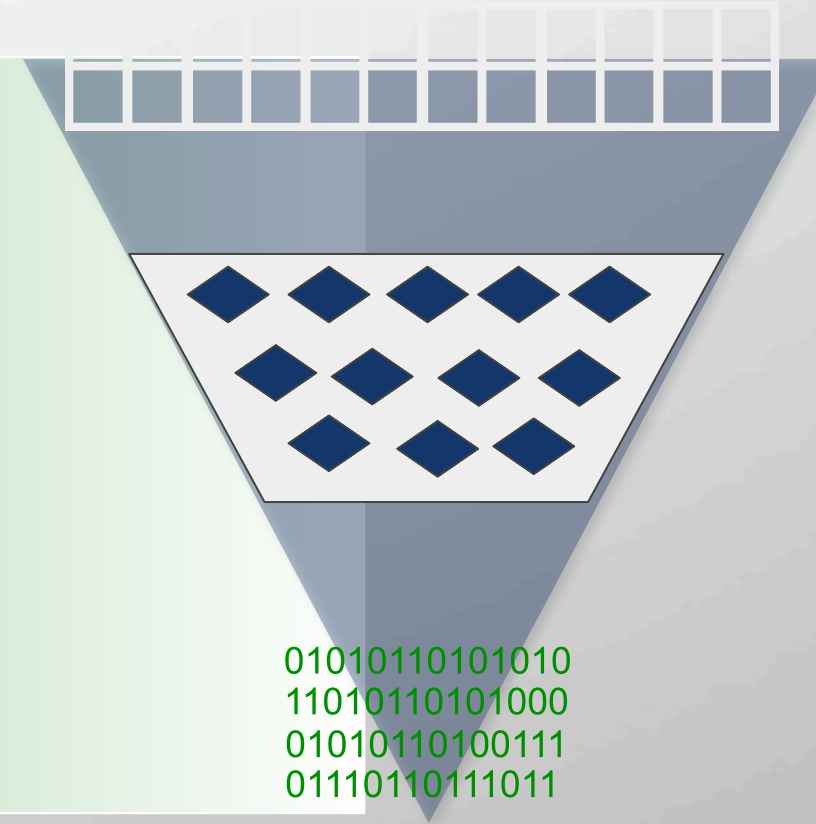
- 38,288 compute cores
- 9,572 compute nodes
- One quad-core AMD 2.3 GHz Opteron processors (Budapest) per node
- 4 processor cores per node
- 8 GB of memory per node
- 78 TB of aggregate memory
- 1.8 GB memory / core for applications
- /scratch disk default quota of 750 GB

- Light-weight Cray Linux operating system
- No runtime dynamic, shared-object libs
- PGI, Cray, Pathscale, GNU compilers

Last year

Hierarchical Parallelism

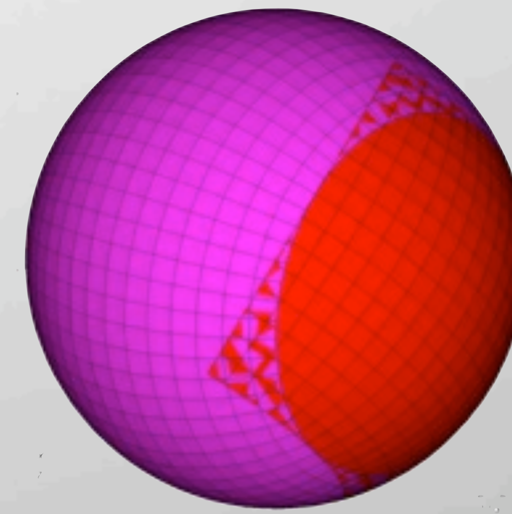
- **MPI parallelism between nodes (or PGAS)**
- **On-node, SMP-like parallelism via threads (or subcommunicators, or…)**
- **Vector parallelism**
  - SSE/AVX/etc on CPUs
  - GPU threaded parallelism



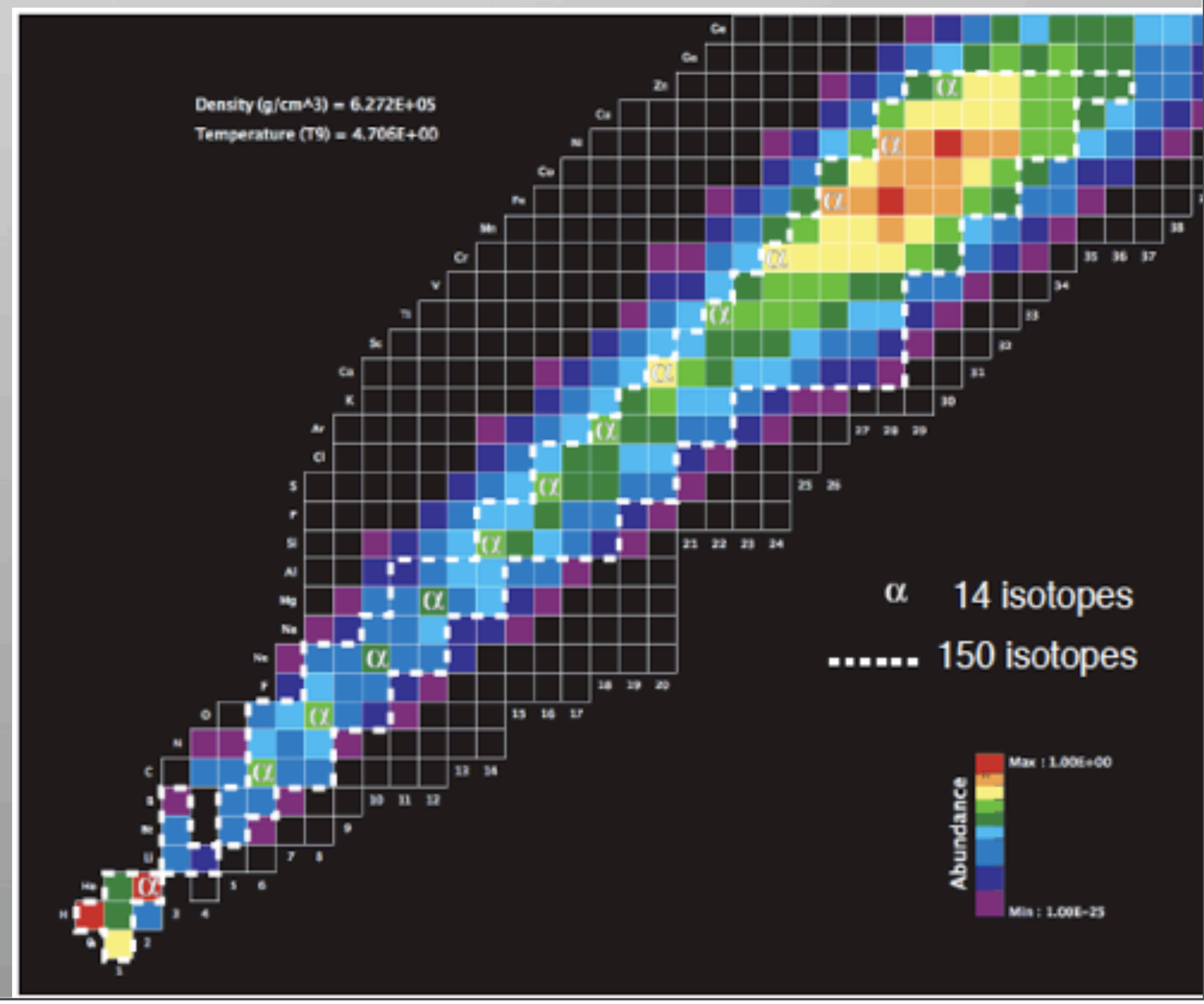
- **Exposure of unrealized parallelism is essential to exploit all near-future architectures.**
- **Uncovering unrealized parallelism and improving data locality improves the performance of even CPU-only code.**
- **Experience with vanguard codes at OLCF suggests 1-2 person-years is required to "port" extant codes to GPU platforms.**
  - **Likely less if begun today, due to better tools/compilers**

Good news! Stellar astrophysics tends to have a lot of unrealized parallelism at present

- **Simulation codes for stellar evolution and explosions**
  - Exemplars of “multiphysics application codes”
  - Typically many degrees-of-freedom per spatial grid point
    - radiation transport
    - nuclear burning
  - Spatial domains typically parallelized via domain decomposition



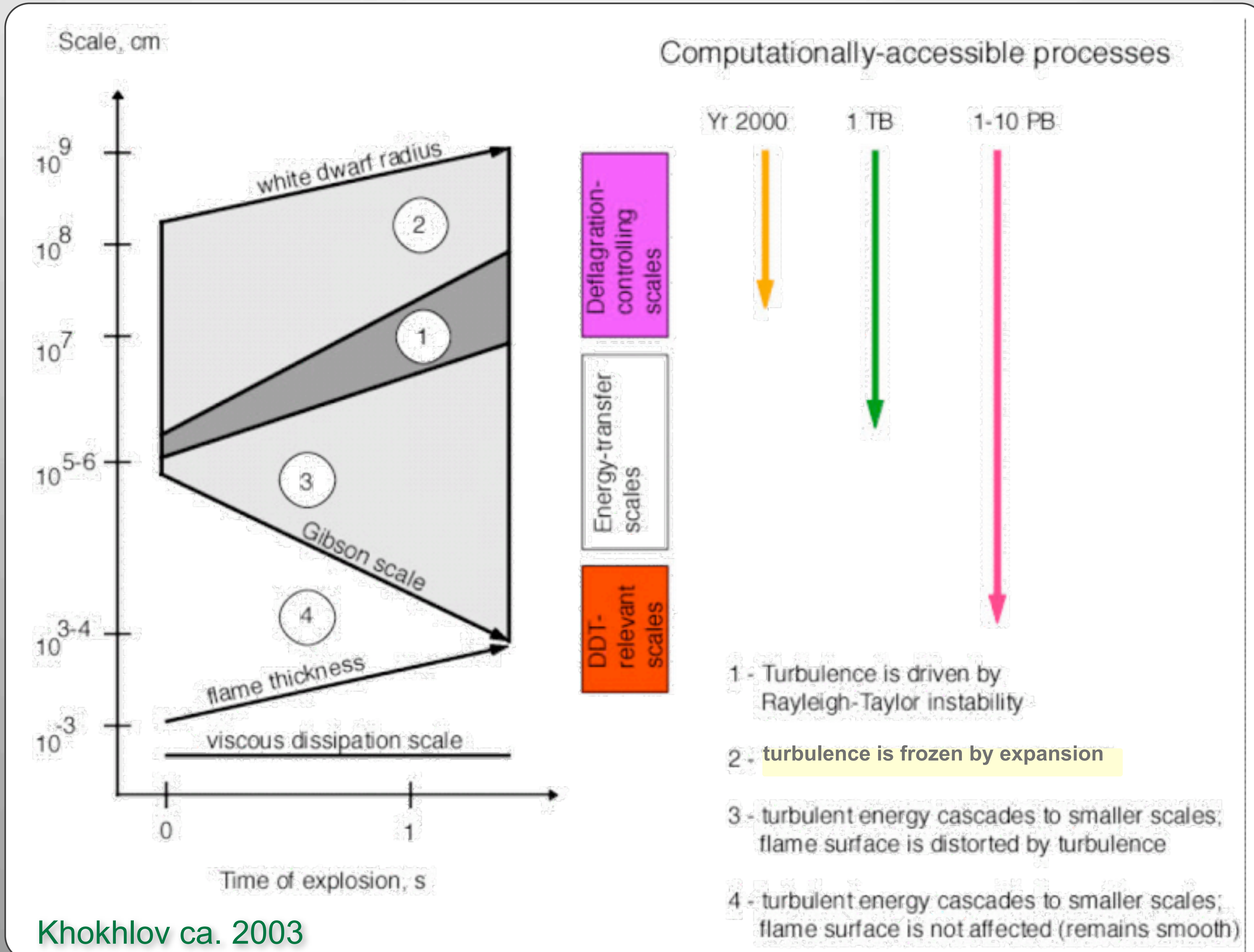
- **Related, computationally-intensive topics will, perhaps, have to work harder to identify additional parallelism outside of large stellar simulations, but plenty of opportunity exists.**
  - high-density physics
  - nuclear structure and reactions



Posited Exascale Specs

System attributes	2010	"2015"		"2018"	
System peak	2 PF	200 PF/s		1 Exaflop/s	
Power	6 MW	15 MW		20 MW	
System memory	0.3 PB	5 PB		32–64 PB	
Node performance	125 GF	0.5 TF	7 TF	1 TF	10 TF
Node memory BW	25 GB/s	0.1 TB/s	1 TB/s	0.4 TB/s	4 TB/s
Node concurrency	12	O(100)	O(1,000)	O(1,000)	O(10,000)
System size (nodes)	18,700	50,000	5,000	1,000,000	100,000
Total node interconnect BW	1.5 GB/s	150 GB/s	1 TB/s	250 GB/s	2 TB/s
MTTI	day	O(1 day)		O(1 day)	

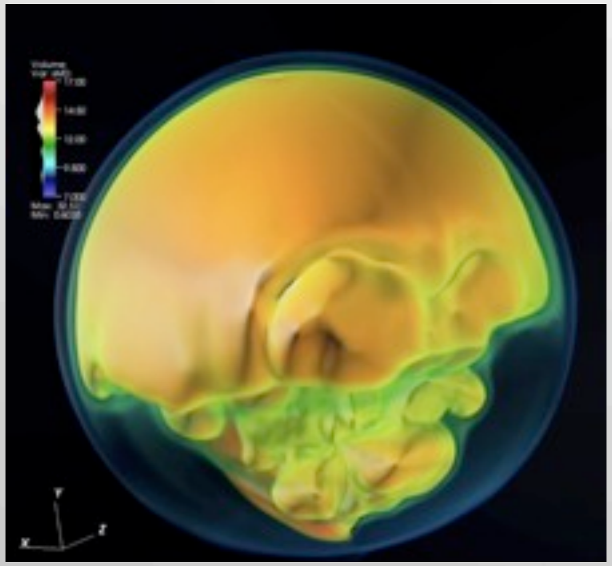
Achieving high spatial (or phase-space, etc.) resolution will be very difficult.



Total memory on the entire exascale system will be O(10 PB)

Example goals/highlights from NP Exascale Workshop report (2009)

http://science.energy.gov/~/media/ascr/pdf/program-documents/docs/Np_report.pdf



Full quantum kinetics

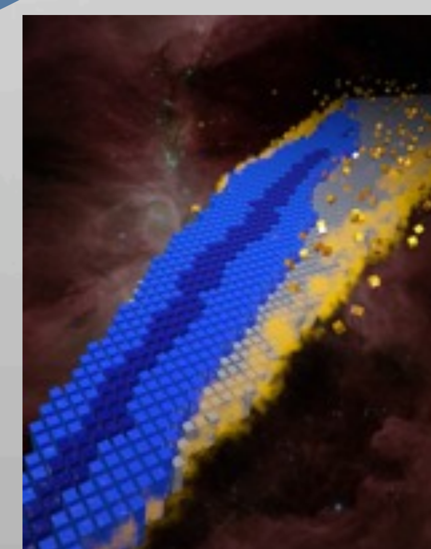
Large (precision) nuclear network

Multi-energy, multi-angle neutrino transport

150-species nuclear network

Multi-energy neutrino transport and coherent neutrino flavor mixing

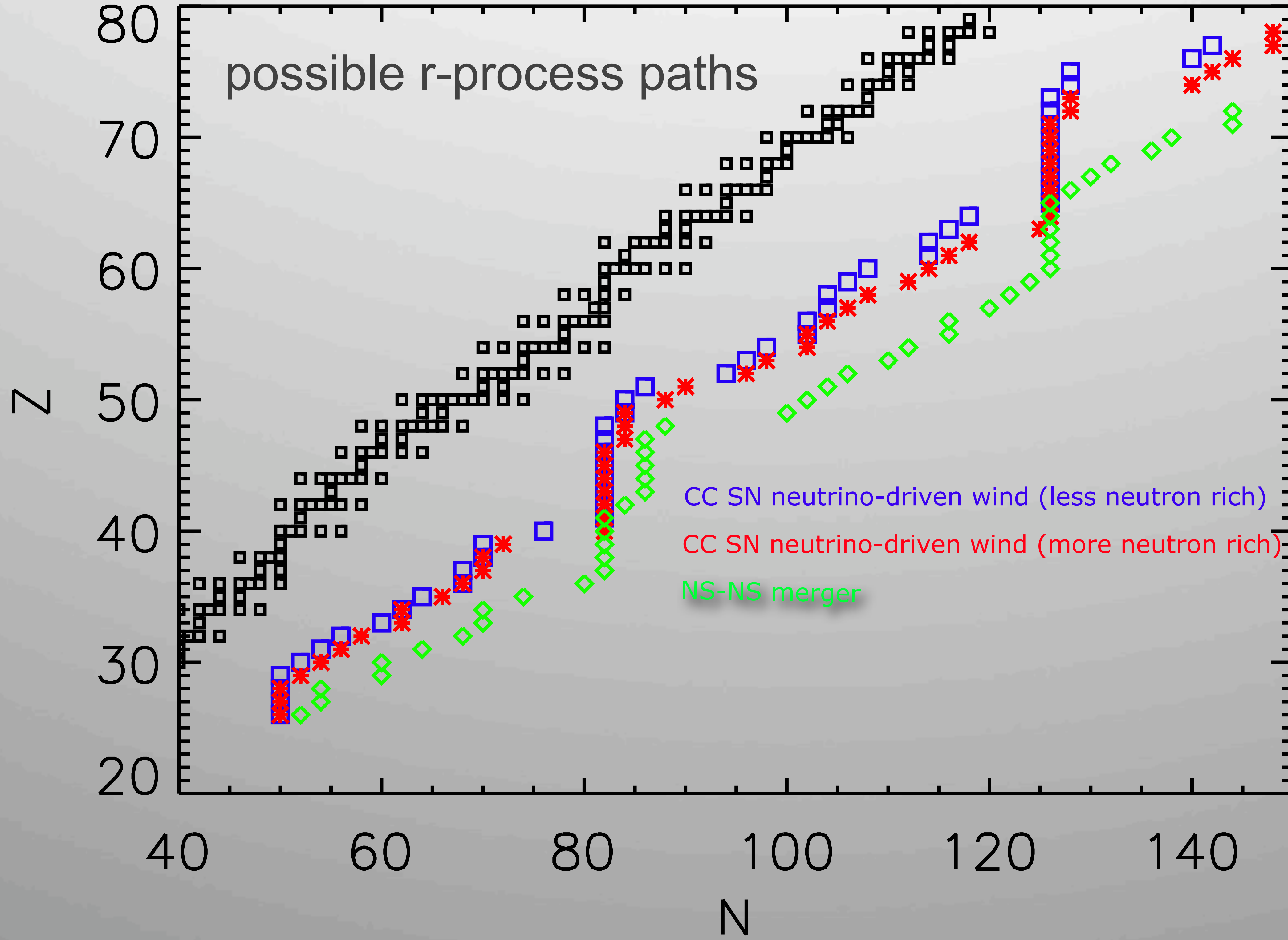
CC SNe



10x Tera | 100x Tera | Peta | 10x Peta | 100x Peta | Exa-flop year sustained

All of these goals are attainable, but will require new algorithms and implementations to bridge the gap to the posited architectures.

Simulation is required to guide experiment



FACILITY FOR
RARE ISOTOPE BEAMS

OLCF

OAK RIDGE
National Laboratory

Locally bulk-synchronous programming model is not a viable path for maximum performance on these new platforms

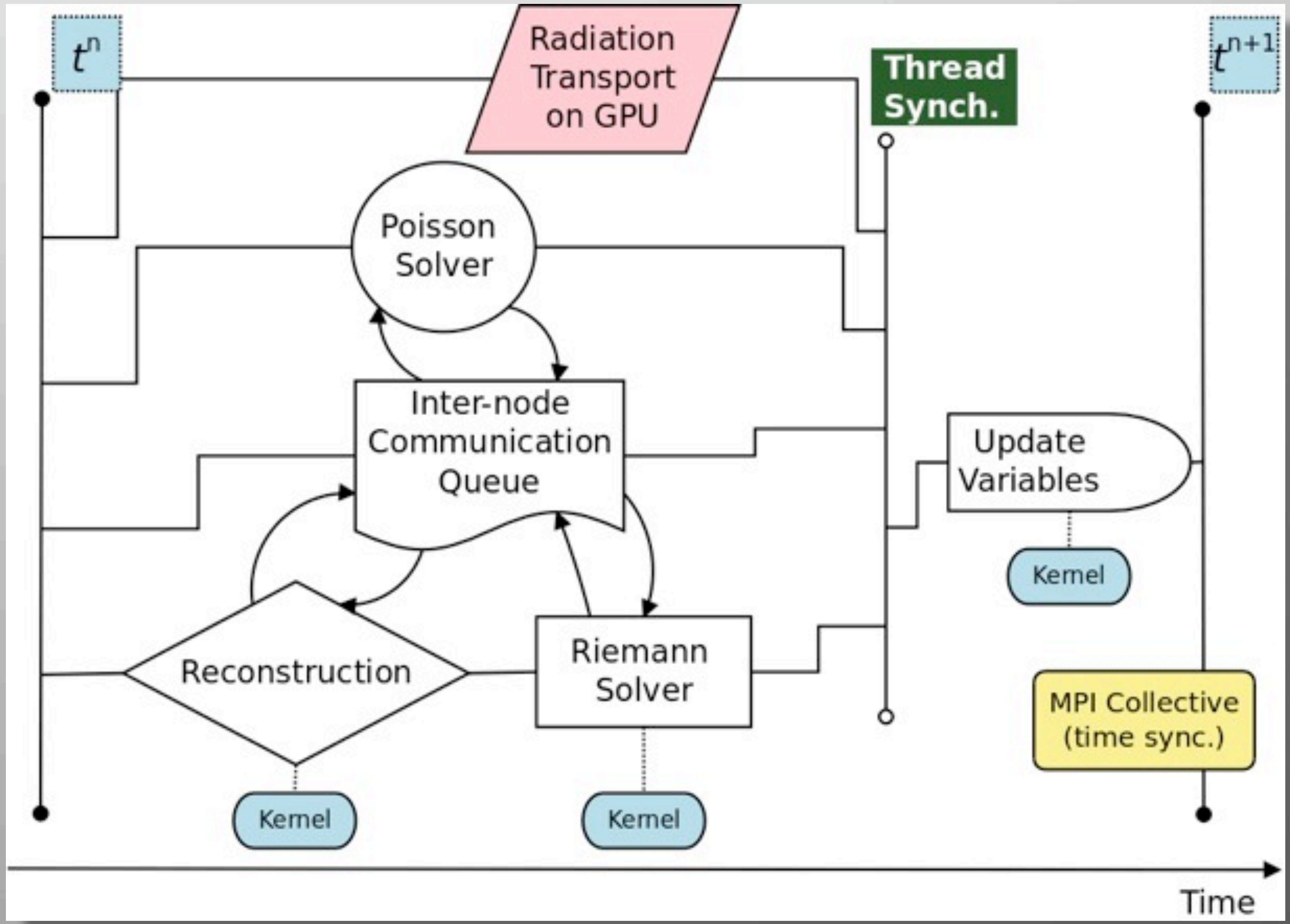
- **FLOP/s are cheap and moving data is expensive**
- **Even perfect knowledge of resource capabilities at every moment and perfect load balancers will not rescue billion-thread SPMD implementations of PDE simulations**
- **Cost of rebalancing frequently is too large, but the Amdahl penalty of failing to rebalance is fatal**
- **To take full advantage of asynchronous algorithms, we need to develop greater expressiveness in scientific programming**
  - **Create separate threads for logically separate tasks, whose priority is a function of algorithmic state, not unlike the way a time-sharing OS works**
  - **Join priority threads in a directed acyclic graph (DAG), a task graph showing the flow of input dependencies; fill idleness with noncritical work or steal work**

Comments taken directly from keynote address by David Keyes at EU-US HPC Summer School, June 2012

Asynchronous execution models via task scheduling

- **Examples exist already in other domains**
  - MAGMA (linear algebra)
  - MADNESS (DFT)
  - Uintah (terrestrial combustion)

- **Operator-split physics modules become "tasks" associated with execution threads**



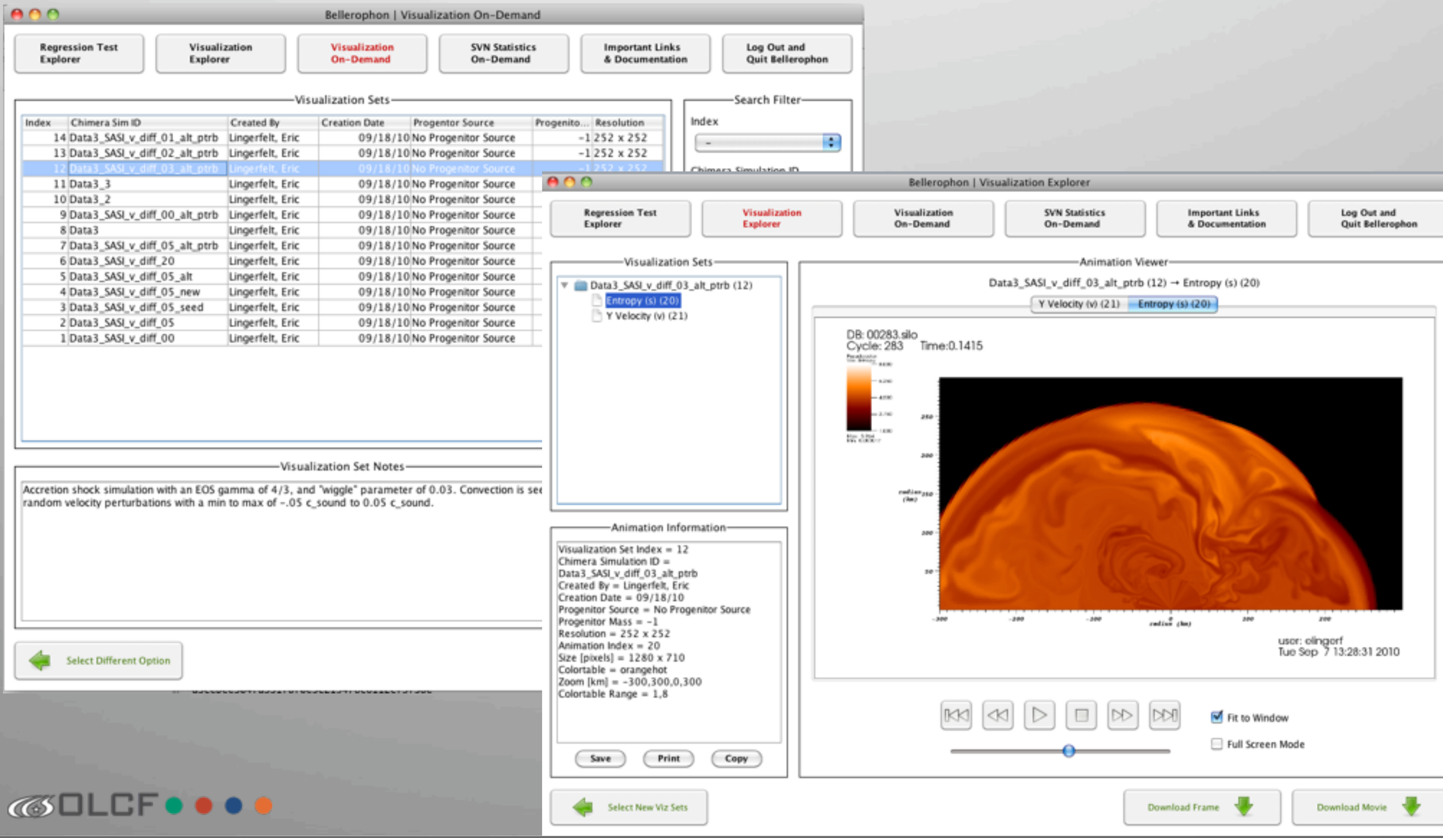
Will the exascale (or before) machine be primarily a "strong-scaling" platform?

- Memory constraints provide a hard ceiling for spatial resolution and number of unknowns.
  - bytes/FLOP goes down by an order of magnitude
- Simulations will be certainly be larger, but likely not as large as one would expect if scaling with FLOPs is assumed.
  - no more than ~10x the number of MPI ranks?
  - this connotes no more than factors of ~2 in resolution in each dimension for 3D
- OK: considerable understanding can be realized by fully exploring parameter space.
  - progenitor mass, rotation, metallicity
  - transport approximations, additional physics

Simulation, code, and data management become even harder

bellerophon

- **Revision control, regression testing, viz, workflow...**



Summary

- The future is now! Computers are not getting faster from the perspective of a nuclear (astro-)physicist. They are only getting "wider."
- The Xeon Phi/GPU/BG\Q choice is no choice at all. They are all versions of a single narrative.
- Stellar astrophysics is rife with unrealized parallelism, but architectural details and memory (i.e. cost, power) constraints will present considerable challenges. Additional support (for both "application scientists" and our CS/Math collaborators) will be required to surmount these challenges.
- Bulk-synchronous execution is a terrible way to try to exploit near-future architectures. A new programming model will require considerably more effort than a simple multi/many-core port.
- Managing large simulations is something we can barely do know, but how about managing 1000's, 10's of thousands, or 100's of thousands of simulations? We should not expect to rely on solutions to be thrown over the fence from developers in other communities.