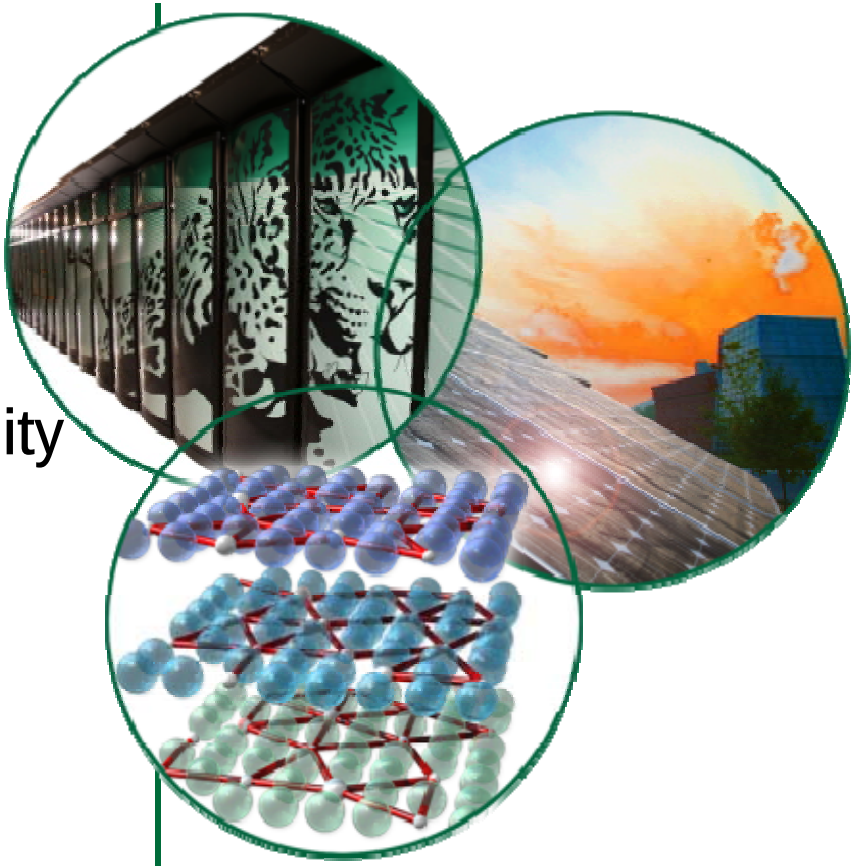
Major Software Challenges as we move towards Exascale



Al Geist

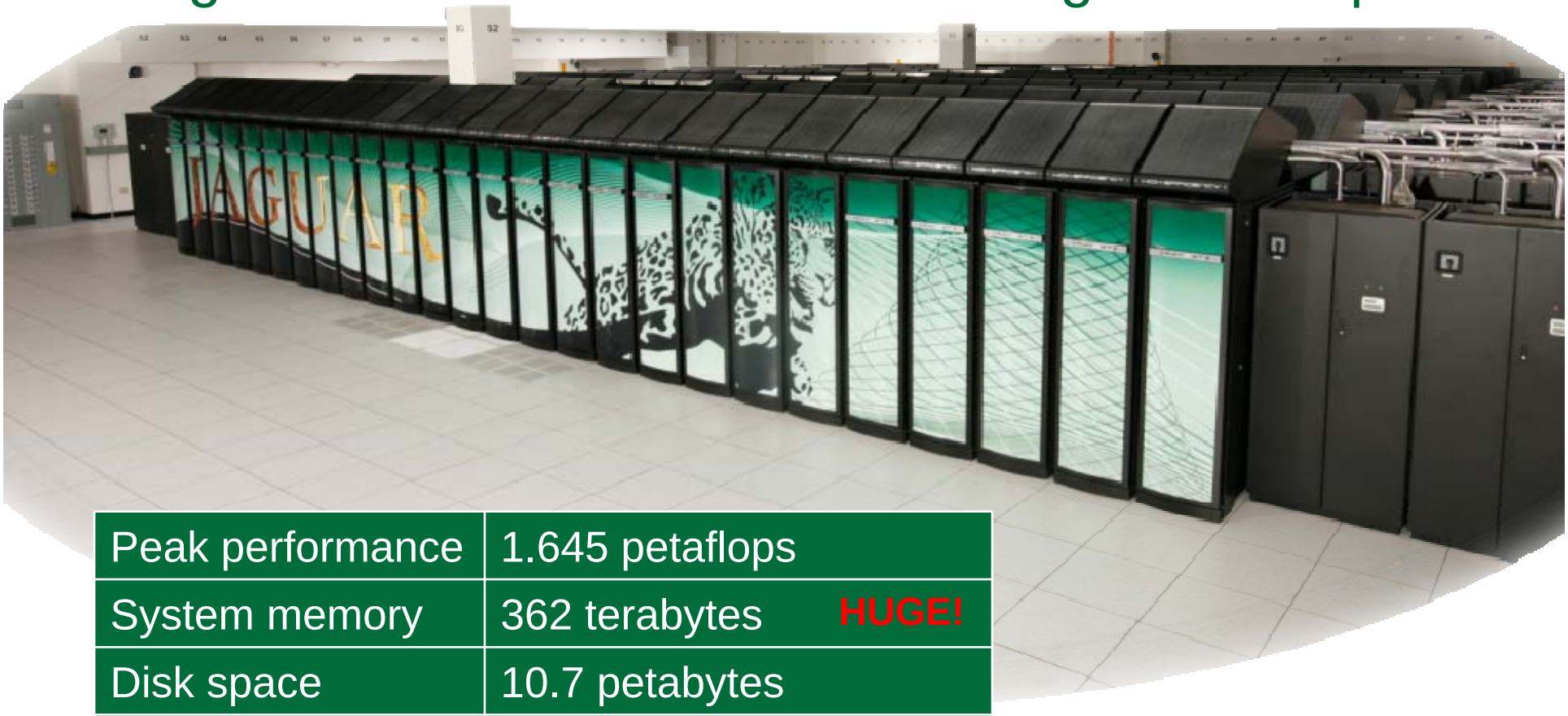
Oak Ridge Leadership Computing Facility

SOS 13
Hilton Head South Carolina
March 10, 2009

U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY

Jaguar: Department of Energy Leadership computer
Designed for science from the ground up



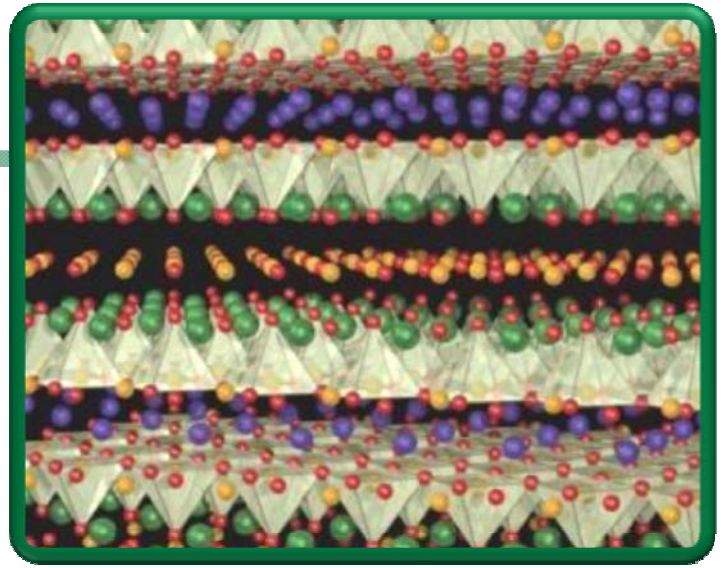
Peak performance	1.645 petaflops
System memory	362 terabytes **HUGE!**
Disk space	10.7 petabytes
Disk bandwidth	240 gigabytes/second
Interconnect BW	532 terabytes/second
Number cores	181504

Jaguar enables Sustained Petaflop Applications

Superconductivity

This project solves the Hubbard Model with disorder to investigate why Cooper pairs develop at different temperatures in different materials.

DCA++ is running at 1.35 PF on 150,000 cores
Winner of Gordon Bell Prize, 2008



PI: Thomas Schulthess, ETH/ORNL
Code: DCA++
Allocation: 60,000,000 hrs

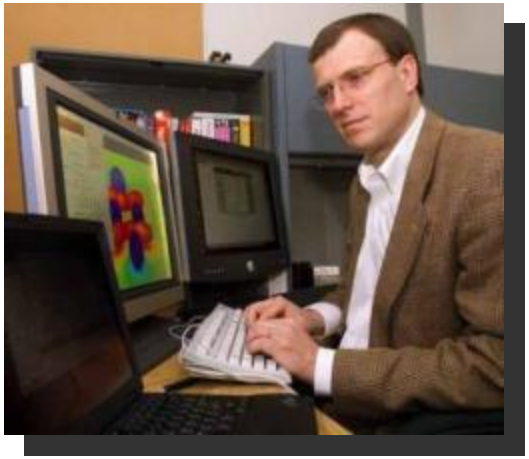
Magnetic Nanoparticles

This project studies the thermodynamics of magnetic nanoparticles.

LSMS sustained 1.05 PF on Jaguar

More to come

Leadership Computing is Transforming World Energy, Environment, Security



"We finally have a true leadership computer that enables us to run calculations impossible anywhere else in the world.

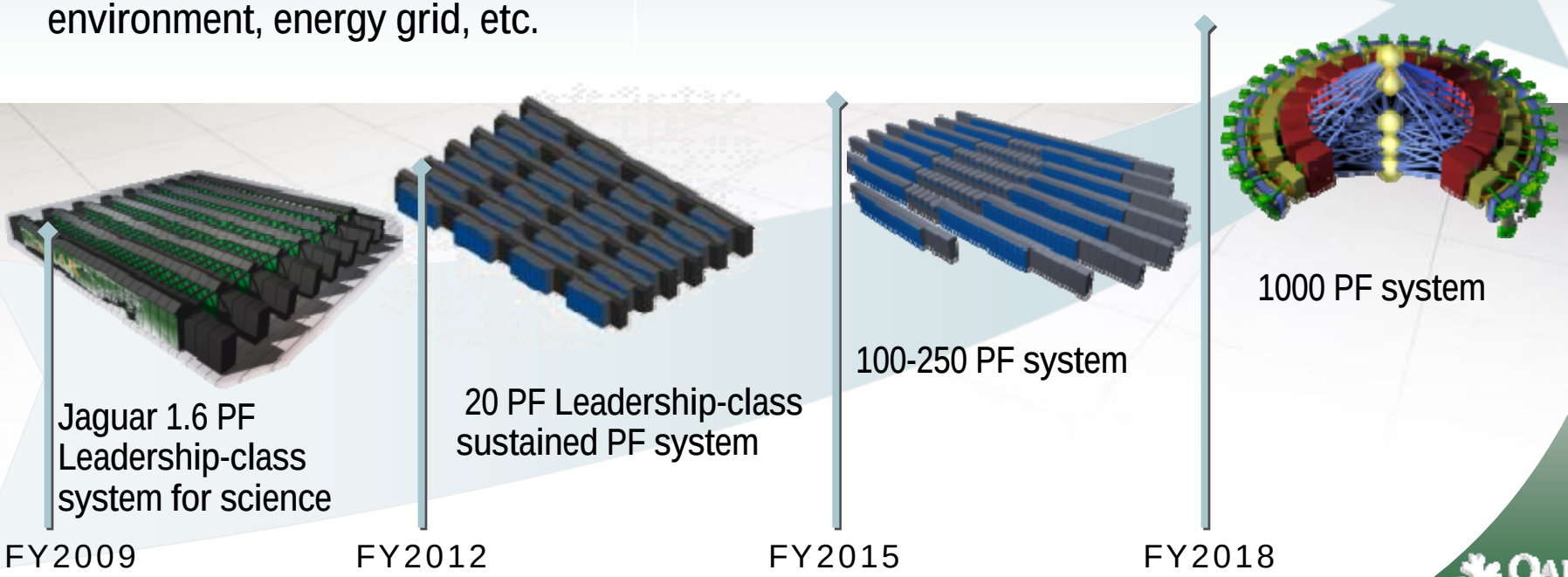
"The huge memory and raw compute power of Jaguar combine to transform the scale of computational chemistry.

"Now that we have NWChem and MADNESS running robustly at the petascale, we are unleashing a flood of chemistry calculations that will produce insights into energy storage, catalysis, and functionalized nano-scale systems."

Robert Harrison
ORNL and University of Tennessee

ORNL's Roadmap to Exascale Applications
What are the Major Software Challenges?

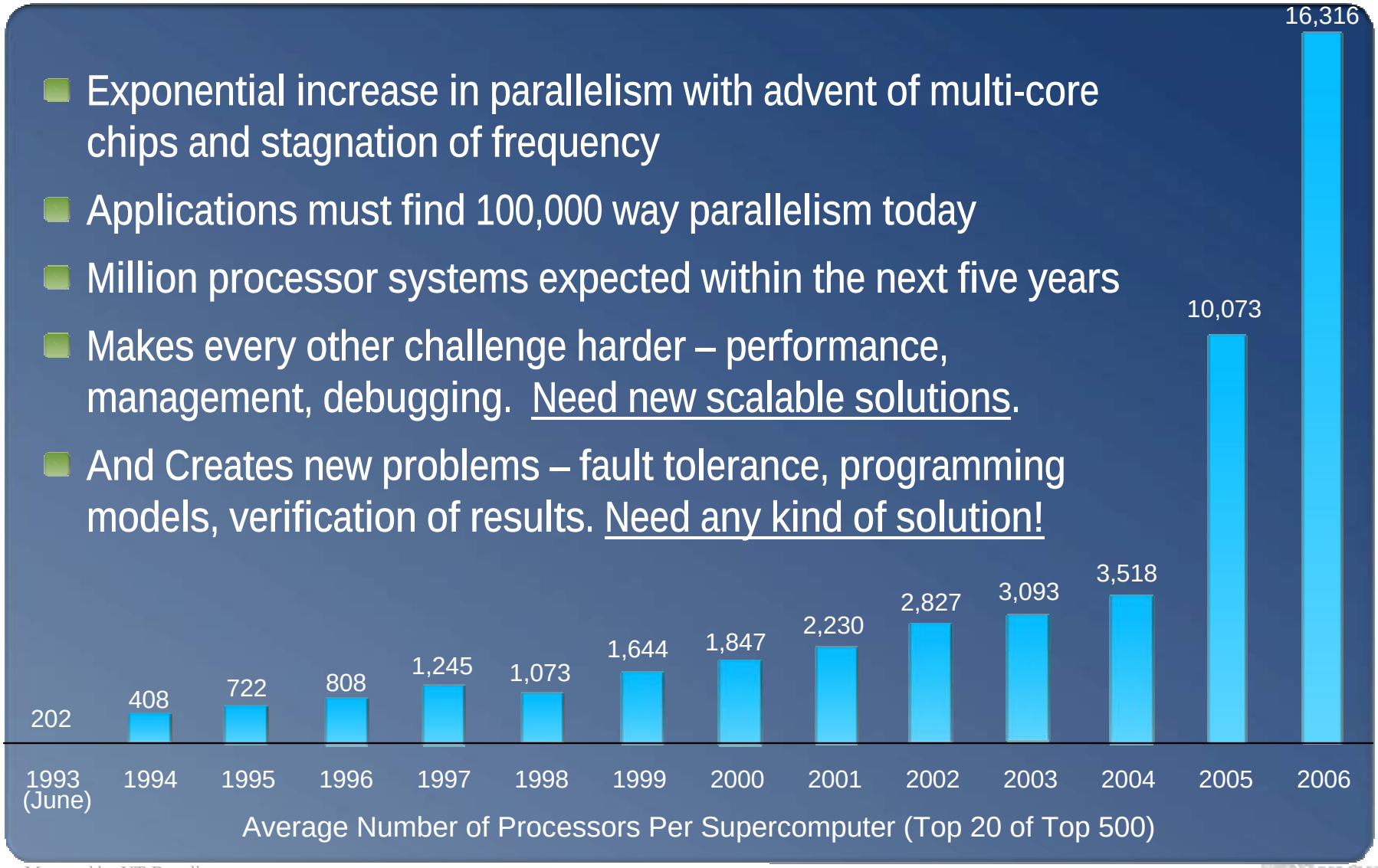
Mission: Deploy and operate the computational resources required to tackle global challenges	Vision: Maximize scientific productivity and progress on the largest scale computational problems
• Deliver transforming discoveries in materials, biology, energy technologies, etc. • Ability to investigate otherwise inaccessible systems: climate, environment, energy grid, etc.	• Providing world-class computational resources and specialized services for the most computationally intensive problems • Providing stable hardware/software path of increasing scale to maximize productive applications development



#1 Challenge: Extreme Growth of Scale

$$\frac{\delta s}{\delta t}$$

- Exponential increase in parallelism with advent of multi-core chips and stagnation of frequency
- Applications must find 100,000 way parallelism today
- Million processor systems expected within the next five years
- Makes every other challenge harder – performance, management, debugging. Need new scalable solutions.
- And Creates new problems – fault tolerance, programming models, verification of results. Need any kind of solution!



#2 Challenge:
Memory Wall - Bandwidth, Latency, Volume

- Applications will have to deal with increasing hierarchies of memory Already five levels of direct access memory and growing.
- Relative Bandwidth in and out of multi-core chip decreasing as core count goes up but number of pins stays the same
- Latency increasing as memory gets further away (in cycles)
- Memory per core decreasing in absolute terms and relative terms (arrays and buffers keeping track of a million nodes)

Applications need to be rewritten to include:
Software strategies to mitigate high memory latencies.
Hierarchical algorithms to deal with bandwidth across the memory hierarchy.

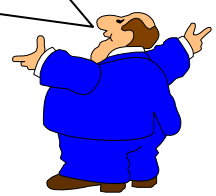
#3 Challenge: Faults become Continuous

Failure will go from being a rare event to being continuous at all levels.

–Hardware
–OS
–Libraries
–Compilers
–Apps

Need Coordinated Response Across levels

With petascale systems We have already entered the world of continuous failure.



The Exascale "monster in the closet" is Silent Errors

- Gates flip spontaneously in memory or processor (seen at LANL, UVa) **(**cosmic rays, temperature, voltage fluctuations, harmonics)
- Bits changed in flight between nodes (seen at SNL in Red)
- Can't be detected or predicted by definition (except by redundancy)

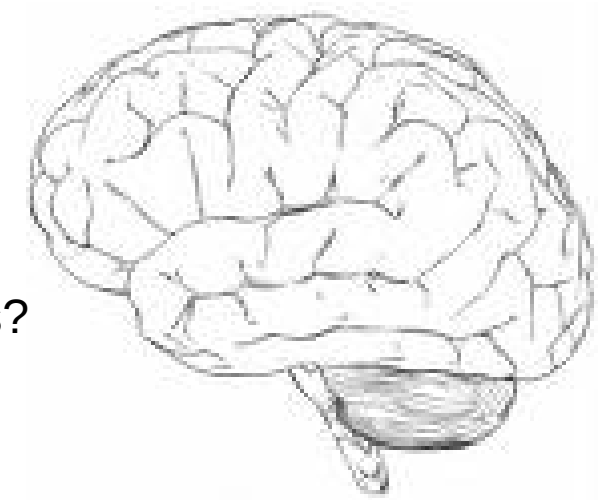
<u>Requires paradigm shift in SW development:</u>
Put less effort in eliminating failure and more effort into fault oblivious software at all levels

Nature has already had to address Continuous Failure on billions of processors

Human Brain

Laid out flat – 0.25 m^2 (not 3D) 2mm thick with 6 layers
wire diameter 10 nm, neuron diameter 4 microns
100 billion neurons (1×10^{11}),
0.15 quadrillion connections (1.5×10^{14})
150,000 km of wiring
Wt 1.3 Kg
Frequency 30 hz
Power 25 W

ZetaOps?



Continuous failure – 85,000 die/day, millions misfire/sec

Errors in calculations – make mistakes (examples visual illusions)

Self correction – sensor fusion, redundant calculation

Self healing – reprogram around physical damage

#4 Challenge: Heterogeneous Systems

Heterogeneity exists at many different levels in modern supercomputers.
-- Node types, Operating systems; Interconnection networks:

-- **Heterogeneity within compute node seems to be the new definition**

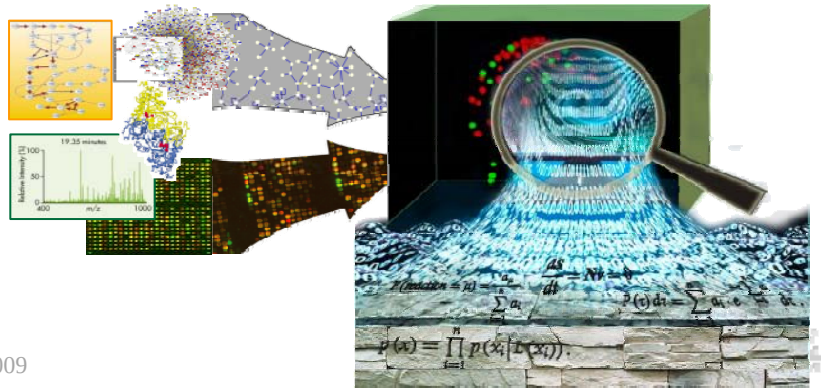
- System management, job scheduling, efficient resource utilization, and load balancing all become much more complex
- Application development – need for heterogeneous parallelization, lack of compilers, debuggers, and performance tools for heterogeneity
- Lack of programming models and standards for programming heterogeneous multi-core processors.

Roadrunner Node board w/ IBM Cell processors

#5 Challenge: Data Volume and Complexity

- **Volume:** Exascale applications running on as many as a million processors are likely to generate data at a rate of several TB/s
- **Dynamic Reduction:** Will require on-the-fly analysis and reduction
- **Performance and Reliability at Scale:** Parallel file system that will scale to millions of clients while maintaining required performance and reliability
- **Data management** - provenance, data movement, archiving, metadata
- **Different data formats**
- **Turning data into scientific discoveries**

data to results to knowledge



#6 Challenge - It is all about the Science: Applications get Bigger and more Complex

- Scaling limitations of present algorithms
- Innovative algorithms for multi-core, heterogeneous nodes
- Software strategies to mitigate high memory latencies
- Hierarchical algorithms to deal with BW across the memory hierarchy
- Need for automated fault tolerance, performance analysis, and verification
- More complex multi-physics requires large memory per node
- Tools can't handle huge source files
- Model coupling for more realistic physical processes
- Need for Scientific Workflows
- Scalable IO for mining of experimental and simulation data

Questions?

Summary
Top Six Software Challenges for the next decade:

- Extreme increase in parallelism
- Memory wall
- Continuous faults
- Heterogeneity
- Managing and exploring huge data
- Increased complexity of Science Applications

13 Managed by UT-Battelle
for the Department of Energy