Argo

An Exascale Operating System and Runtime Research Project

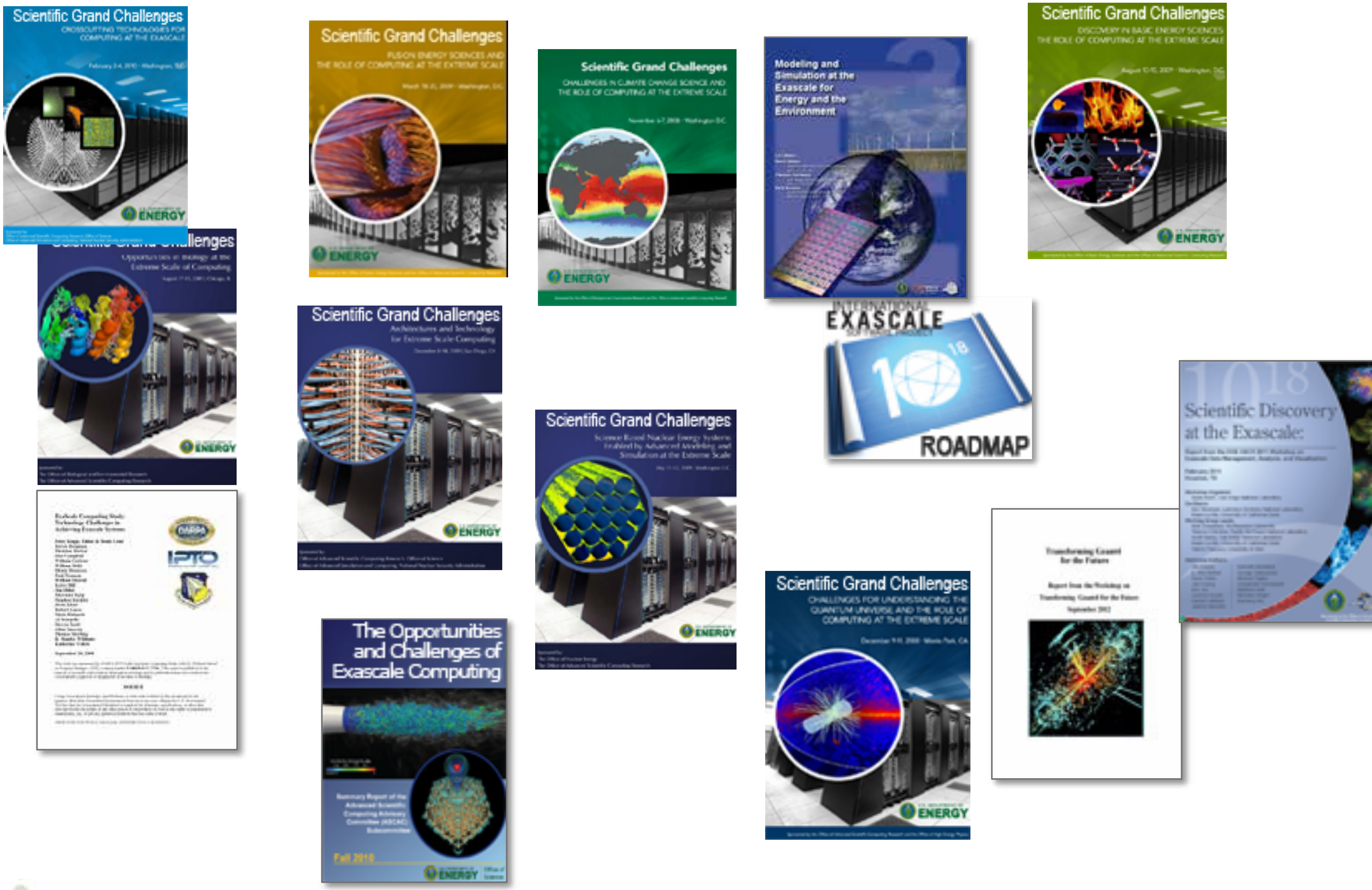
Pete Beckman

Argonne National Laboratory

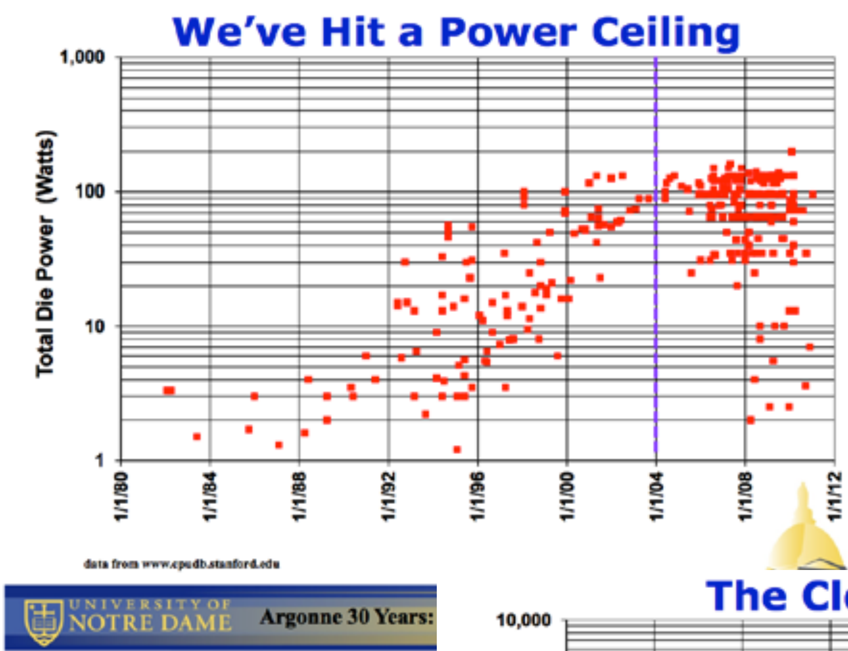
Director, Exascale Technology and Computing Institute

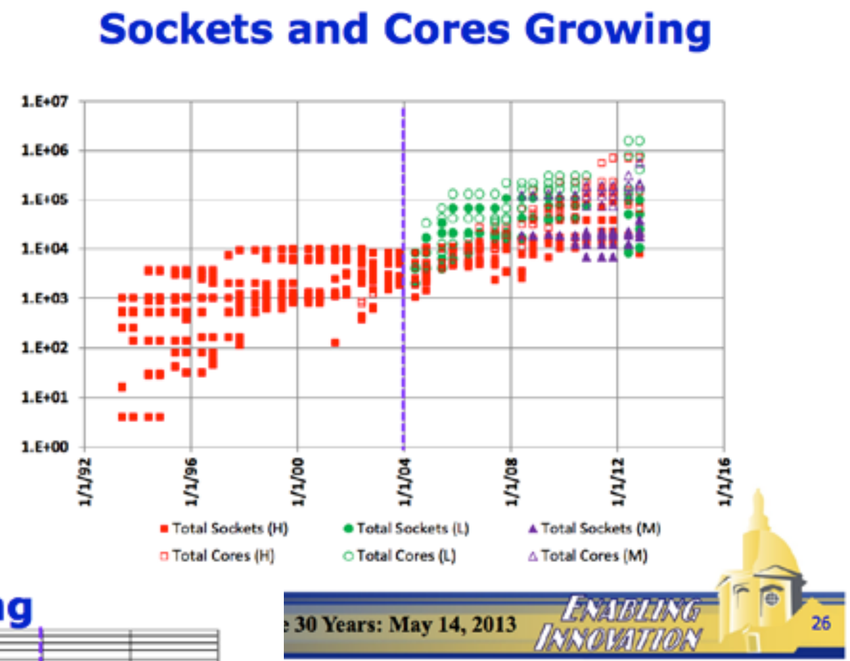
Co-Director, Northwestern University – Argonne Institute of Science and Engineering

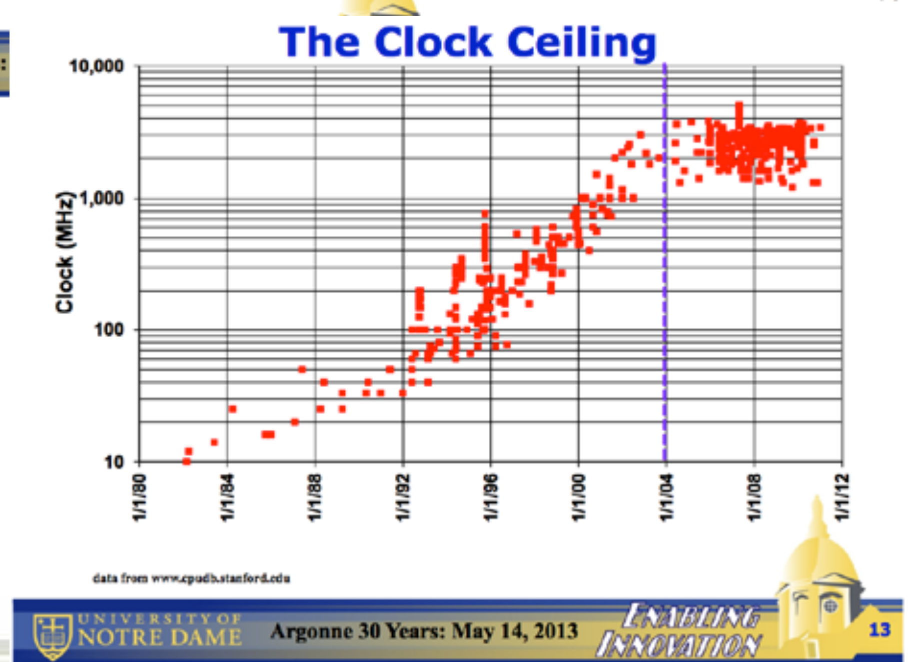
Waiting for 6+ years…



Data from Peter Kogge, Notre Dame







The Argo Team:

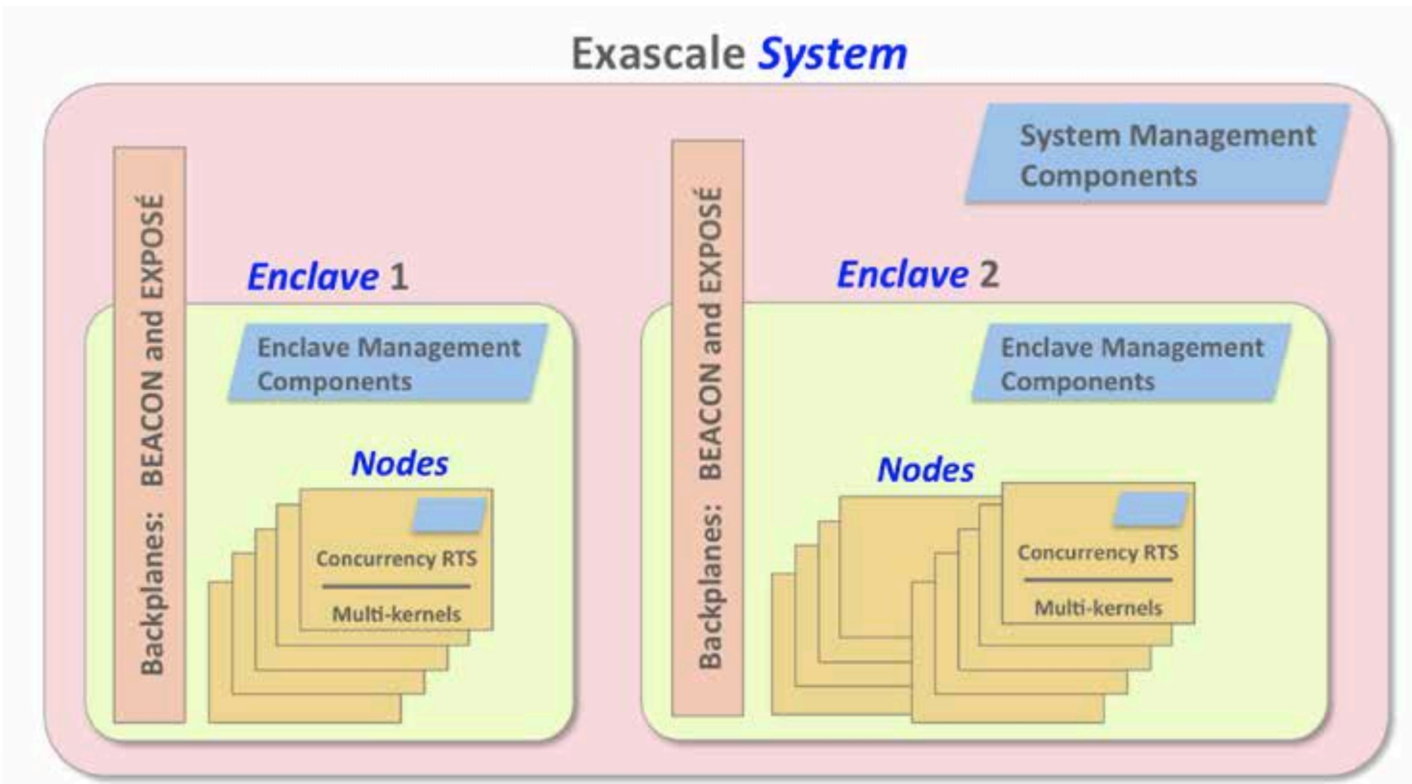
- **ANL**: Pete Beckman, Marc Snir, Pavan Balaji, Rinku Gupta, Kamil Iskra, Rajeev Thakur, Kazutomo Yoshii
- **BU**: Jonathan Appavoo, Orran Krieger
- **LLNL**: Maya Gokhale, Edgar Leon, Barry Rountree, Martin Schulz, Brian Van Essen
- **PNNL**: Sriram Krishnamoorthy, Roberto Gioiosa, David Callahan
- **UC**: Henry Hoffmann
- **UIUC**: Laxmikant Kale, Eric Bohm, Ramprasad Venkataraman
- **UO**: Allen Malony, Sameer Shende, Kevin Huck
- **UTK**: Jack Dongarra, George Bosilca

Argo Key Innovation Areas:
(Focusing on Global OS/R)

- Node OS
- Lightweight Runtime for Concurrency
- Event, Control, and Performance Backplanes
- Global Optimization

Key New Argo Abstractions

- Enclave
  - (recursive)
  - tree-based hierarchy and recursive decomposition
  - At each level in the hierarchy, four key aspects change: granularity of control, communication frequency, goals, and data resolution.

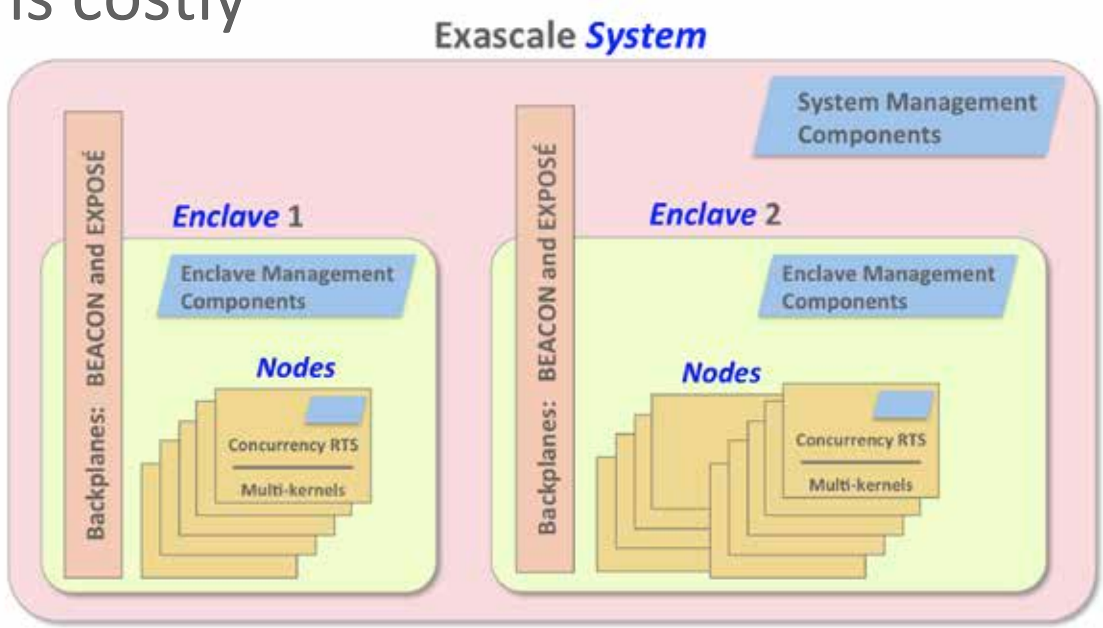


Benefits

- Embedded feedback and response mechanisms
  - Self-aware, Goal-based
  - `#include <sanjay_presentation.pptx>`
- Meta-handle for enclaves
  - Can write meta-programs for enclave
    - (manage parallelism, task-manager, etc)
  - Allows application-specific fault managers, streaming I/O handlers, many-task UQ engines, and event-based coordination of coupled components
  - `#include <sanjay_presentation.pptx>`
- Hierarchical, coordinated, global system can set and manage power budgets, respond to faults, support enclave components that leverage machine learning, and manage intranode parallelism.

Argo: Resource Management Design Principles

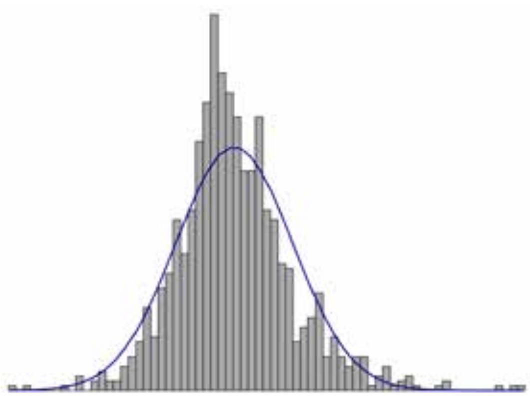
- Resource management is hierarchical, and managers are stackable
- Resource managers are integrated
- Resource managers are customizable and adaptable
- Sharing is avoided whenever possible
- Strict enforcement is costly

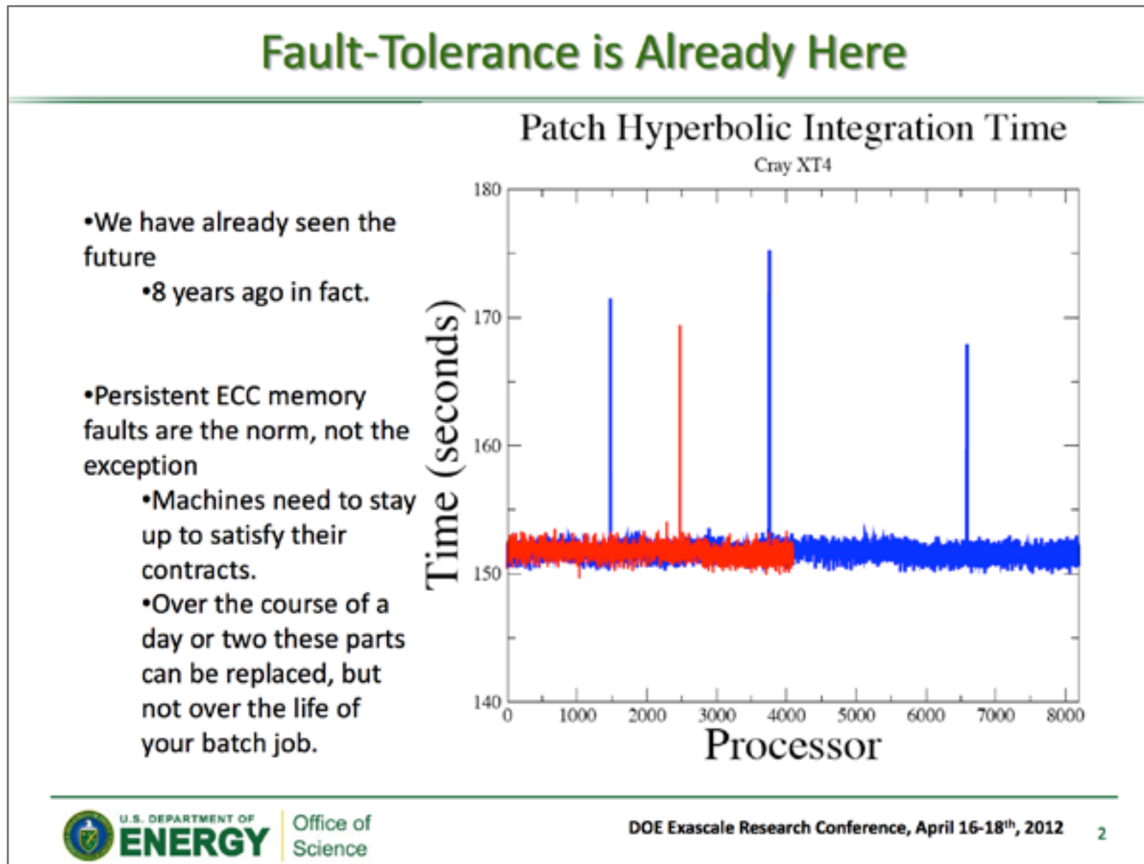


A Peek Into Research Areas

Threads/Tasks: Managing Exploding Parallelism

- Dynamic parallelism and decomposition
  - Programmer cannot hand-pick granularity / resource mapping
    - (equal work != equal time)





Fault-Tolerance is Already Here

- We have already seen the future
  - 8 years ago in fact.
- Persistent ECC memory faults are the norm, not the exception
  - Machines need to stay up to satisfy their contracts.
  - Over the course of a day or two these parts can be replaced, but not over the life of your batch job.

U.S. DEPARTMENT OF ENERGY | Office of Science

DOE Exascale Research Conference, April 16-18th, 2012 2

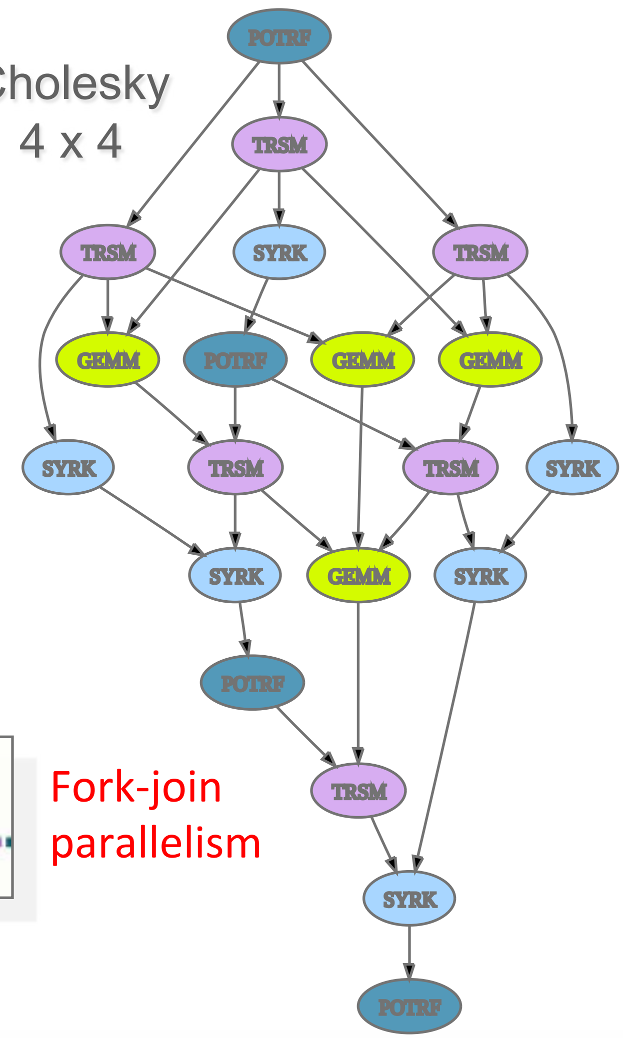
Variability is the new norm:
- Power
- Resilience
- Intranode Contention

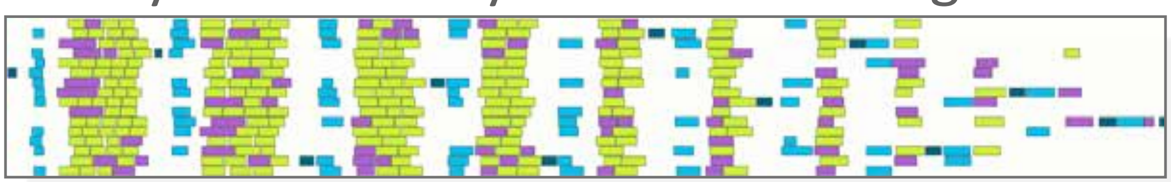
From Brian Van Straalen

PLASMA: Parallel Linear Algebra s/w for Multicore Architectures

- Objectives
  - High utilization of each core
  - Scaling to large number of cores
  - Shared or distributed memory
- Methodology
  - Dynamic DAG scheduling
  - Explicit parallelism
  - Implicit communication
  - Fine granularity / block data layout
- Arbitrary DAG with dynamic scheduling

Cholesky
4 x 4





Fork-join parallelism

DAG scheduled parallelism

Time

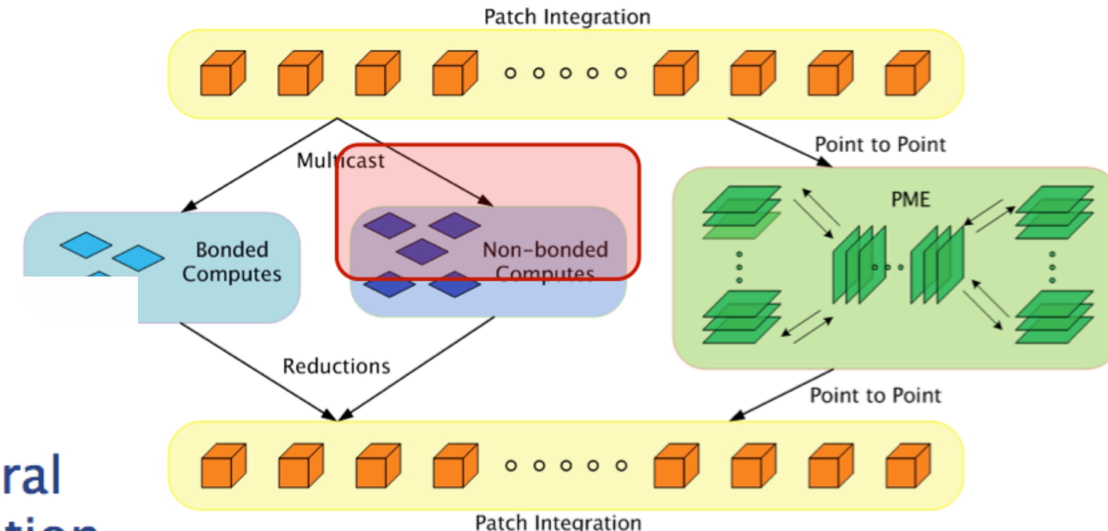
Courtesy Jack Dongarra:

Courtesy: Laxmikant Kale

Charm++
(the run-time and execution model)

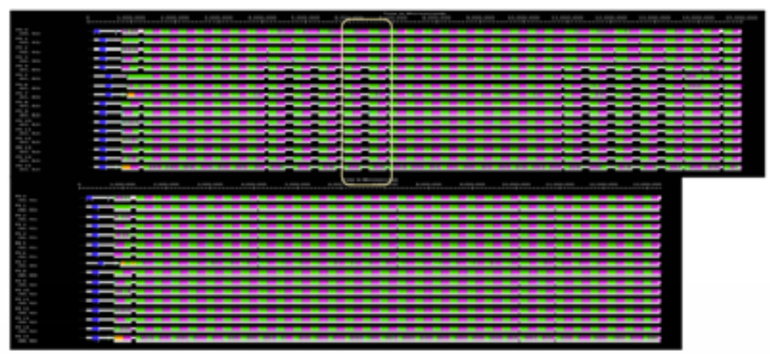
Parallelization Using Charm++

The computation is decomposed into "natural" objects of the application, which are assigned to processors by Charm++ RTS



- Charm++/AMPI style "virtual processors"
  - Decompose into natural objects of the application
  - Let the runtime map them to processors
  - Decouple decomposition from load balancing

Benefits of Temperature Aware LB



Google (re-discovers) OS Noise

Component-Level Variability Amplified By Scale
A common technique for reducing latency in large-scale online services is to parallelize sub-operations across many different machines, where each sub-operation is co-located with its portion of a large dataset. Parallelization happens by fanning out a request from a root to a large number of leaf servers and merging responses via a request-distribution tree. These sub-operations must all complete within a strict deadline for the

Reducing Component Variability
Interactive response-time variability can be reduced by ensuring interactive requests are serviced in a timely manner

Living with Latency Variability
The careful engineering techniques in the preceding section are essential for building high-performance interactive services, but the scale and complexity of modern Web services make it infeasible to eliminate all latency variability. Even if such perfect behavior could

Software techniques that tolerate latency variability are vital to building responsive large-scale Web services.

BY JEFFREY DEAN AND LUIZ ANDRÉ BARROSO

The Tail at Scale

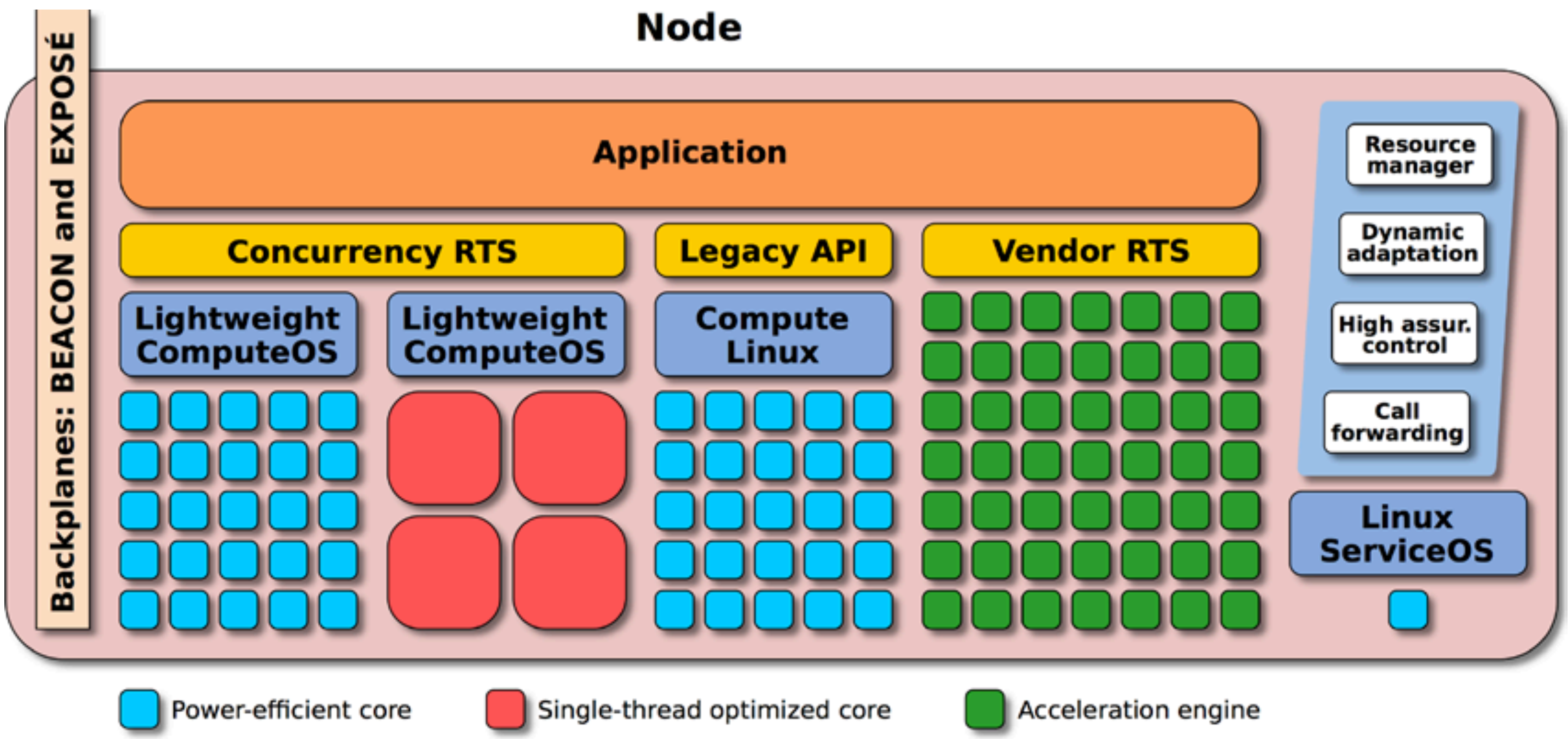
Probability of one-second service-level response time as the system scales and frequency of server-level high-latency outliers varies.

Argo Parallelism (Threads/Tasks)

- Move away from SPMD block synchronous
- Link lightweight thread/task runtime into OS
- Support data dependency driven computation
- Explore memory placement
- Explore pluggable schedulers
- Hardware support for lightweight activation
  - (e.g. BG/Q wake-on, etc)

Project Lead: Sanjay Kale

Core-Specialization for Node OS/R



Project Lead: Kamil Iskra

Memory: Technology Summary from Rob Schreiber

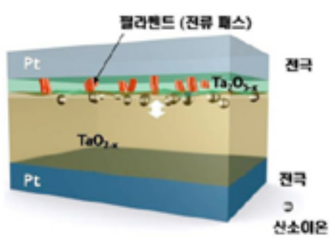
New memory on the horizon

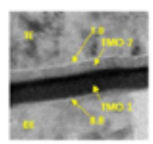
- Spin-Torque-Transfer RAM (STTRAM)
  - Grandis (54nm, acquired by Samsung)
- Phase-Change RAM (PCRAM)
  - Samsung (20nm, diode, up to 8Gb)
  - Micron and Nokia – In phones now
- Resistive RAM (ReRAM)
  - Panasonic (180nm process, 4-layer xpoint)
  - Unity Semi (64MB, acquired by Rambus)

EVERSPIN GRANDIS RENESAS TOSHIBA

SAMSUNG numonyx hynix hp Panasonic ideas for life Rambus

ReRAM





Samsung, HP-Hynix, Sandisk, Toshiba

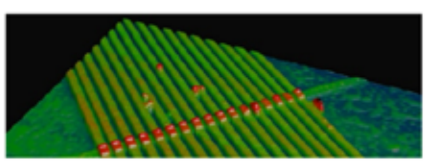
32Gb test chip (Sandisk/Toshiba. 24 nm. ISSCC 2013)

Fast (tens of nsecs) for both read and write

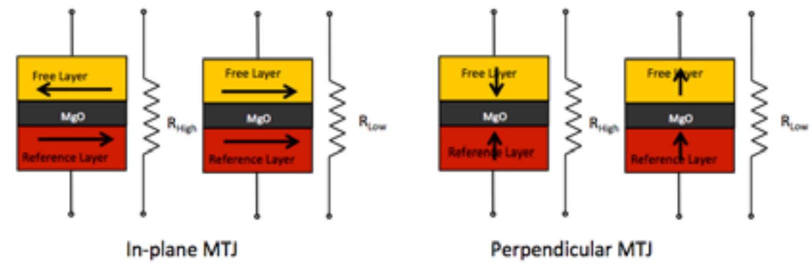
Good data retention and reliability

3D -- 2 to 4 layers

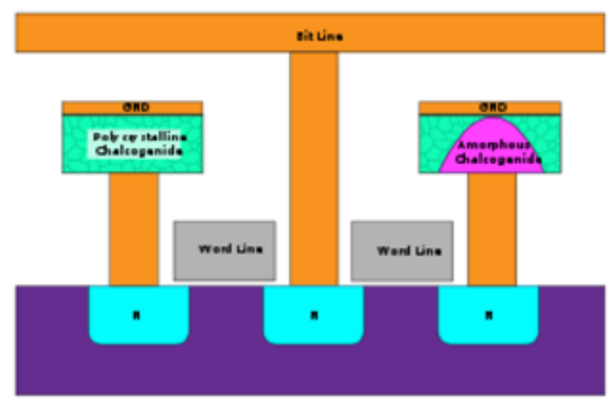
MLC possible



Spin transfer torque (STTRAM)

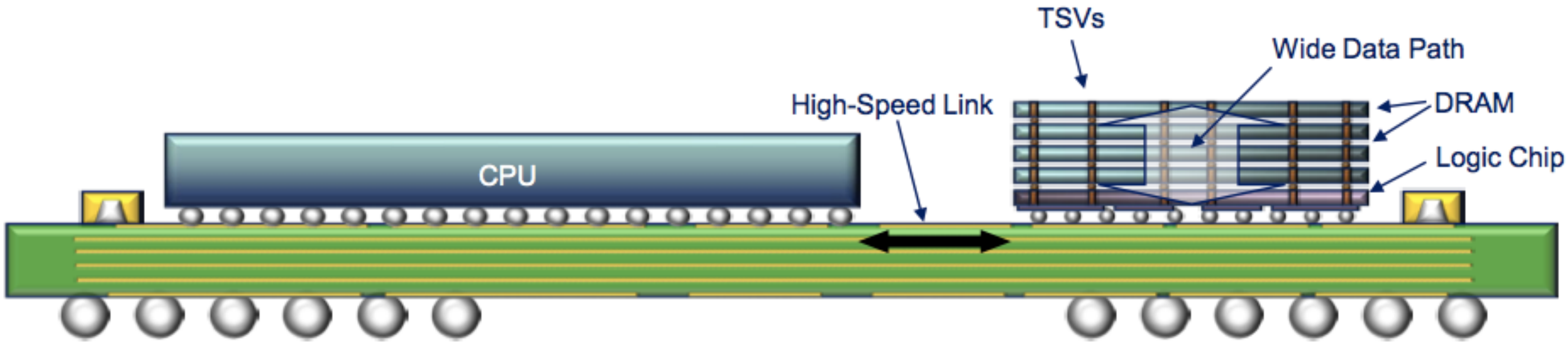


PCRAM



- Shipping today
- MLC (limited by resistance drift)
- Slow, expensive writes
- Wearout issue

Significant Portion of Memory will be non-volatile



- Helps reduce power
- Helps with resilience
- Helps with cost
- How do we represent this in the OS/R?

Power/energy trace tools

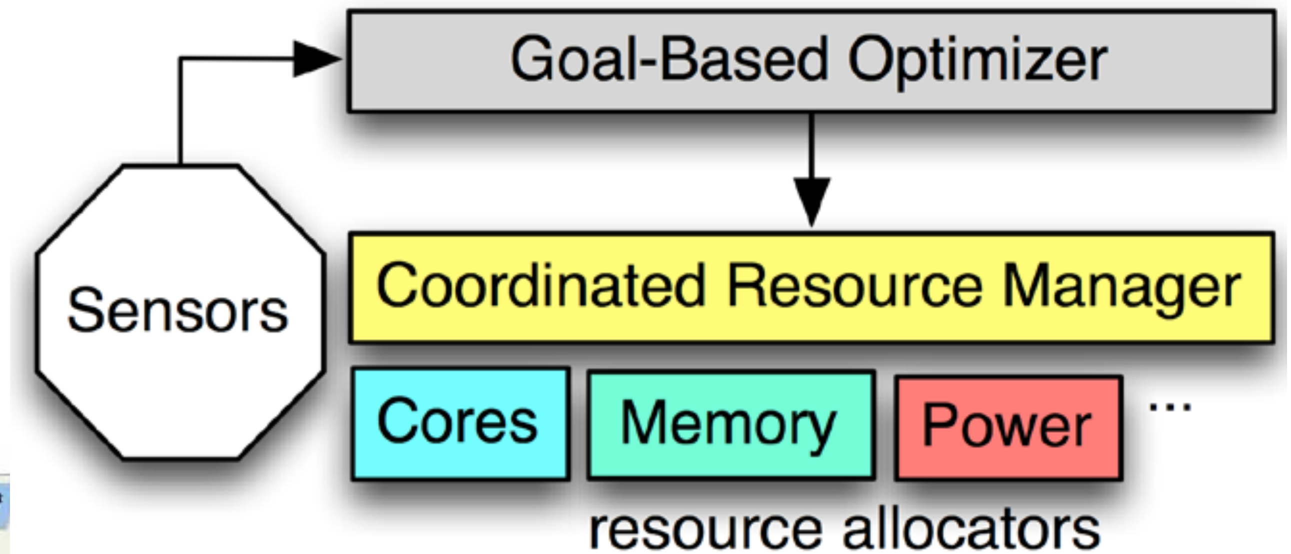
- A command line tool
- No source code modification is required
- Sampling the power consumption with specified interval
- Summarize the total energy consumption

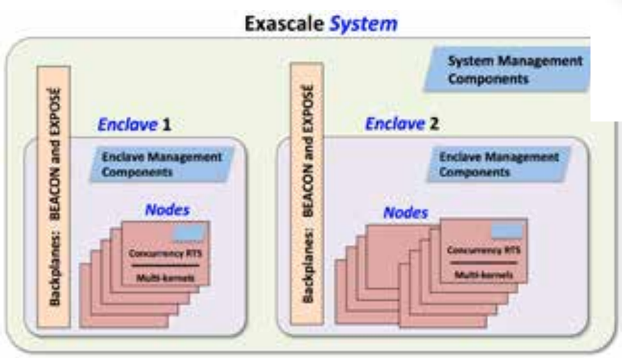
e.g.

```
$ etrace ./app
SOCKET0_ELAPSED=2.000681
SOCKET0_PKG_ENERGY=71.604248
SOCKET0_PP0_ENERGY=44.639069
$ etrace -o file -i 0.1 ./app   # output to file
```

Global View

- Leverage goal-based optimization concepts
- “Self-Aware” Computing





Optimization Lead: Hank Hoffmann

Global View Project Leads: Marc Snir, Rajeev Thakur

Backplane Project Leads: : Allen Malony, Sameer Shende

Wrapup:

- Node OS
- Lightweight Runtime for Concurrency
- Event, Control, and Performance Backplanes
- Global Optimization

Questions?

3 year project: WE NEED POSTDOCS AND GRAD STUDENTS TO COME TO ARGONNE AND HELP!