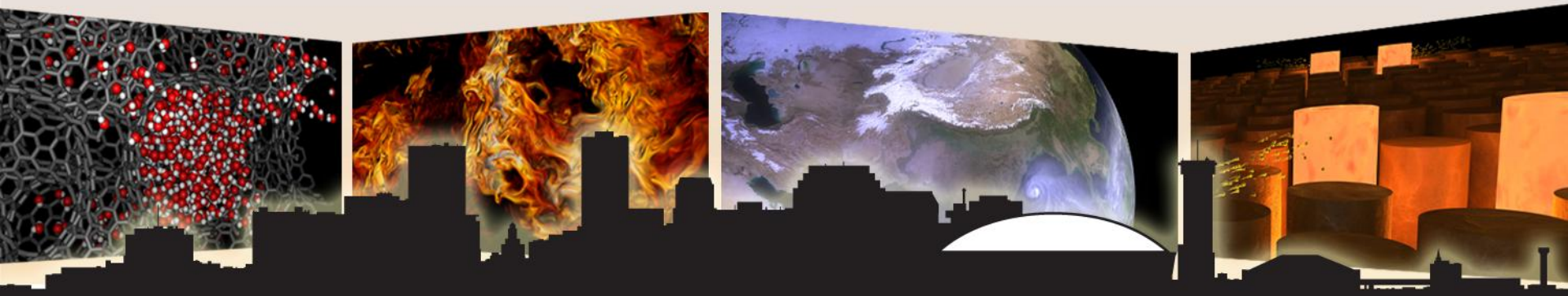
Embracing the Unexpected—System Software for Unrestricted Computing

Presented by

Terry Jones (ORNL), Laxmikant Kalé (UIUC)
Celso Mendes (UIUC), Esteban Meneses (UIUC)
Lukasz Wesolowski (UIUC), José Moreira (IBM)
Eliezer Dekel (IBM), Gennady Laventman (IBM)
Yoav Tock (IBM), Benjamin Mandler (IBM)



U.S. DEPARTMENT OF ENERGY

OAK RIDGE National Laboratory

We're experiencing an architectural renaissance

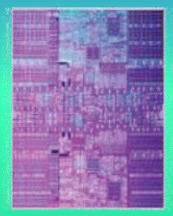
Factors to Change

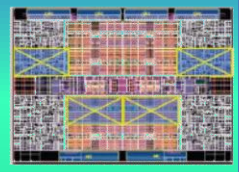
- **Moore's Law – Number of transistors per IC doubles every 24 months**
- **No Power Headroom – Clock speed will not increase (and may decrease) because of power**

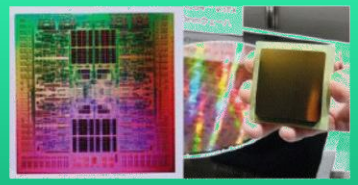
*Power α Voltage^2 * Frequency*
Power α Frequency
Power α Voltage^3

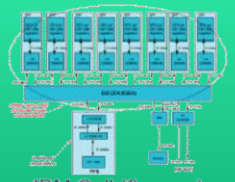
Multicore

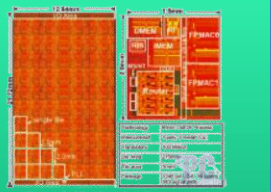
Sun Niagra2 (8 cores)


IBM Power 7 (8 cores)


Intel Xeon(8 cores)

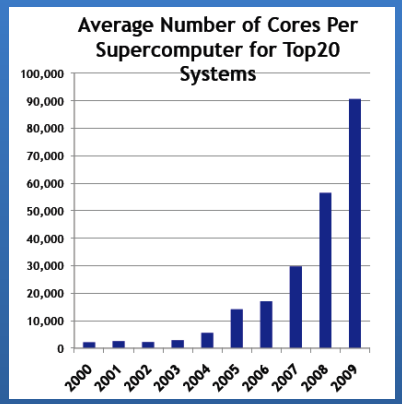

Fujitsu Venus (8 c


IBM Cell (9 cores)

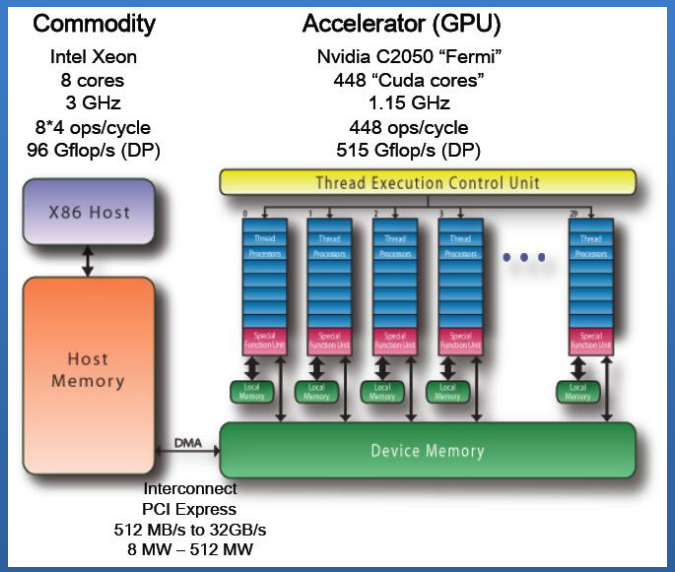

Intel Polaris [experimenta
(80 cores)

AMD Istanbul (6 cores)

Increased Core Counts



Commodity + Accelerator



The HPC Colony Project is providing adaptive system software for improved resiliency and performance

Collaborators

Terry Jones, Project PI | Laxmikant Kalé, UIUC PI | José Moreira, IBM PI

Objectives

- Provide technology to make portable scalability a reality
- Remove the prohibitive cost of full POSIX APIs and full-featured operating systems
- Enable easier leadership-class level scaling for domain scientists through removing key system software barriers

Approach

- Automatic and adaptive load-balancing plus fault tolerance
- High performance peer-to-peer and overlay infrastructure
- Address issues with Linux to provide the familiarity and performance needed by domain scientists

Challenges

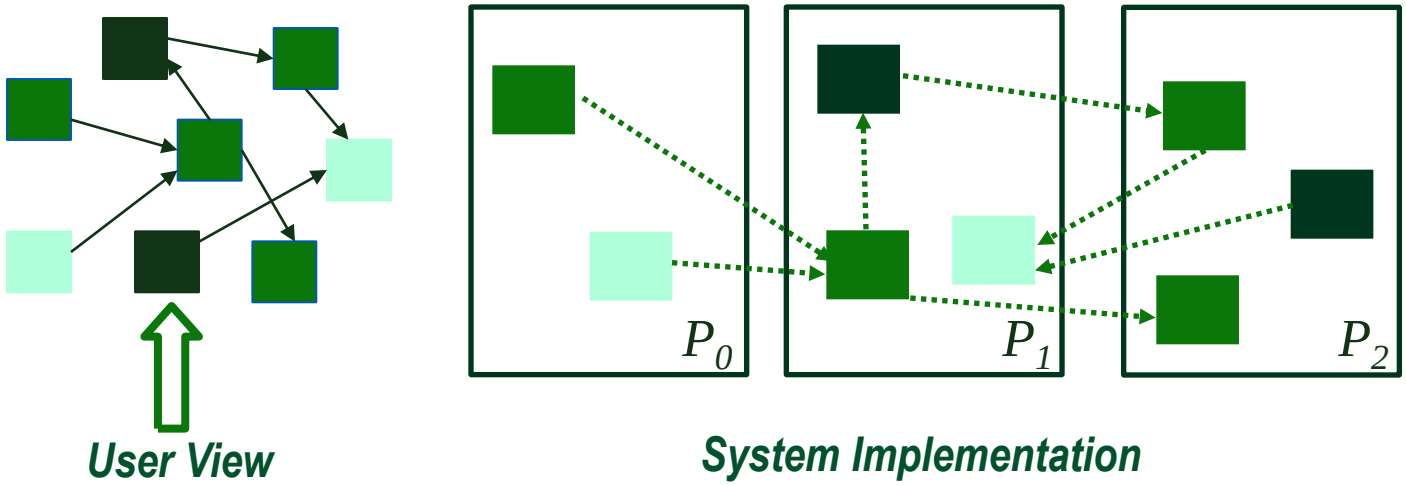
- Computational work often includes large amounts of state which places additional demands on successful work migration schemes
- For widespread acceptance from the Linux community, the effort to validate and incorporate HPC-originated advancements into the Linux kernel must be minimized

Impact

- Full-featured environments allow for a full range of programming development tools including debuggers, memory tools, and system monitoring tools that depend on separate threads or other POSIX API
- Automatic load balancing helps correct problems associated with long running dynamic simulations
- Coordinated scheduling removes the negative impact of OS jitter from full-featured system software

HPC Colony technology –
Processor virtualization with migratable objects

- Divide the computation into a large number of pieces
  - Independent of the number of processors
- Let the runtime system map objects to processors
- Implementations: Charm++, Adaptive-MPI (AMPI)



HPC Colony technology – Fault tolerance enabled by Charm++

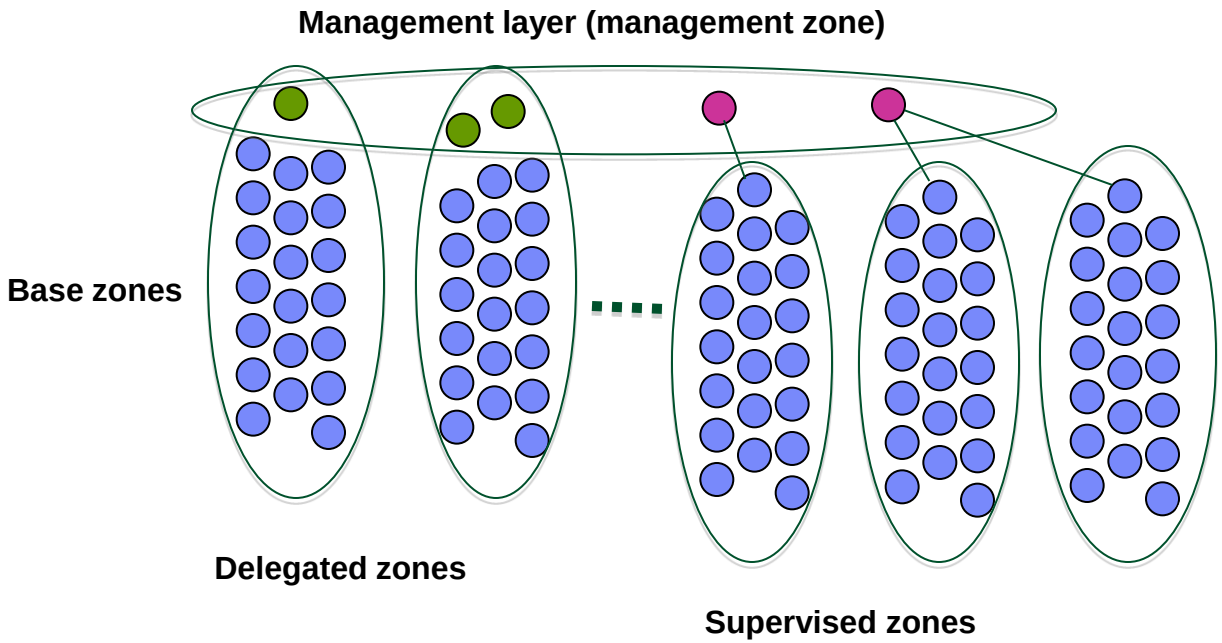
- Automatic checkpointing / fault detection / restart
  - Scheme 1: checkpoint to file-system
  - Scheme 2: In-memory checkpointing
- Proactive reaction to impending faults
  - Migrate objects when a fault is imminent
  - Keep “good” processors running at full pace
  - Refine load balance after migrations
- Scalable fault tolerance
  - Using message-logging to tolerate frequent faults in a scalable fashion

HPC Colony technology –
SpiderCast for high performance communications

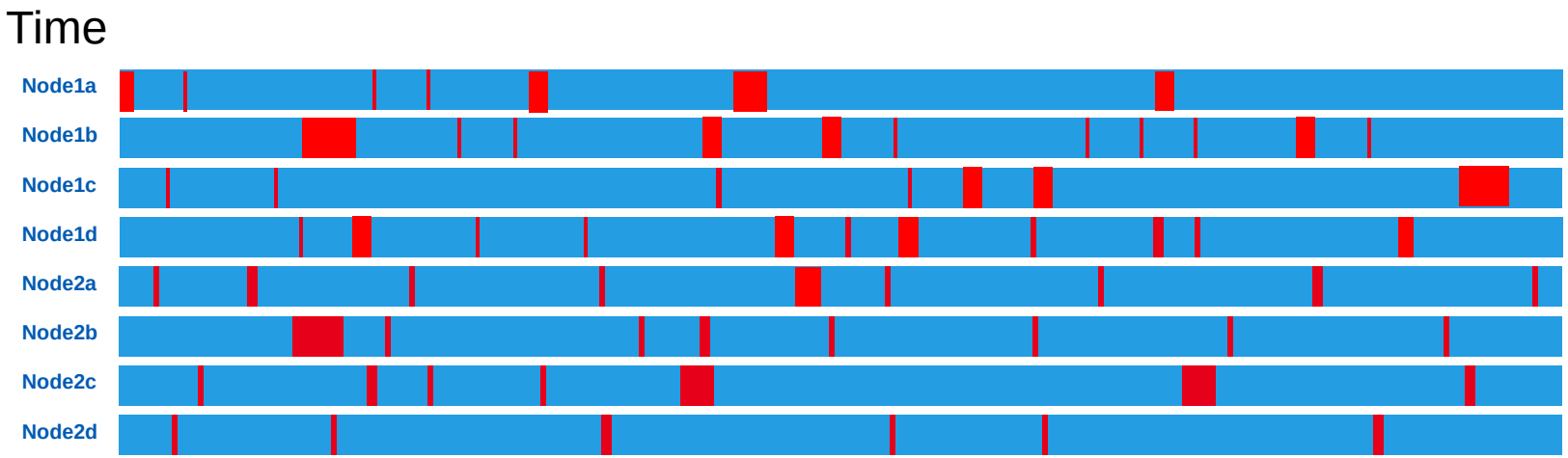
- A scalable, fully distributed, messaging, membership and monitoring infrastructure
- Develop a stand-alone distributed infrastructure that will utilize peer-to-peer and overlay networking technologies, while utilizing HPC platform unique features and architecture
- Focus on:
  - Membership – report which processes are alive; discover and report failing processes
  - Monitoring – collect load and performance statistics
  - Scalable group services – multicast and lightweight pub/sub
- A set of services targeted for:
  - Increasing performance and scalability of scientific computing by providing said services to load balancing, scheduling, fault tolerance, and parallel resource management system software
  - Enabling general-purpose workloads by providing missing distributed software services and components in the OS/Middleware level

HPC Colony technology – New hierarchical communications schemes

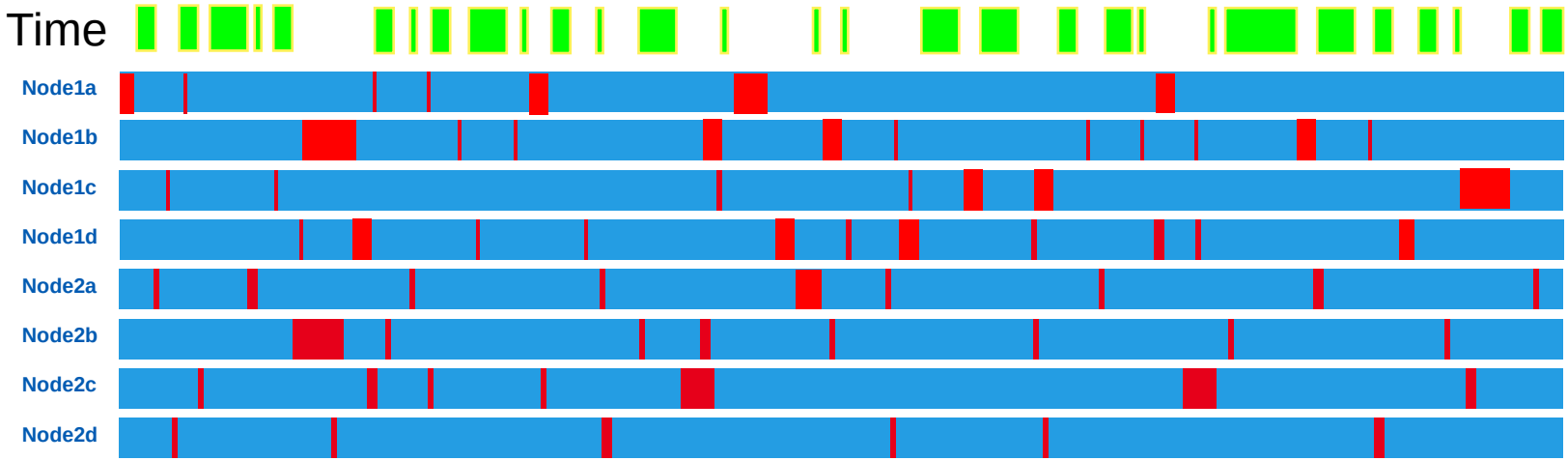
- Multiple base zones federated by a management layer
- Nodes in management layer form a zone, too
- Management nodes either delegates or supervisors
- Support for ~1000 zones x ~1000 members = 1M nodes



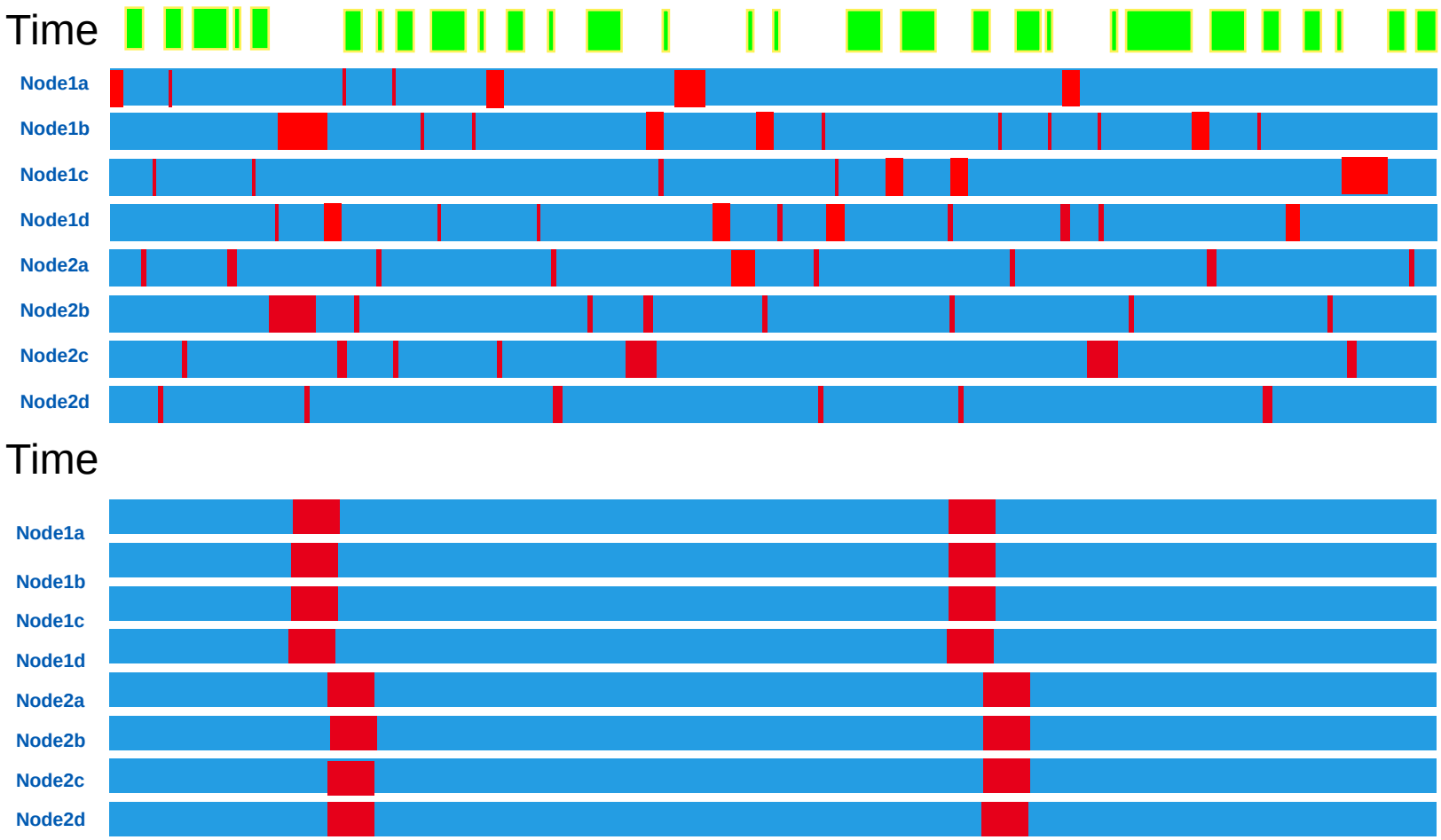
HPC Colony technology – Coordinated scheduling



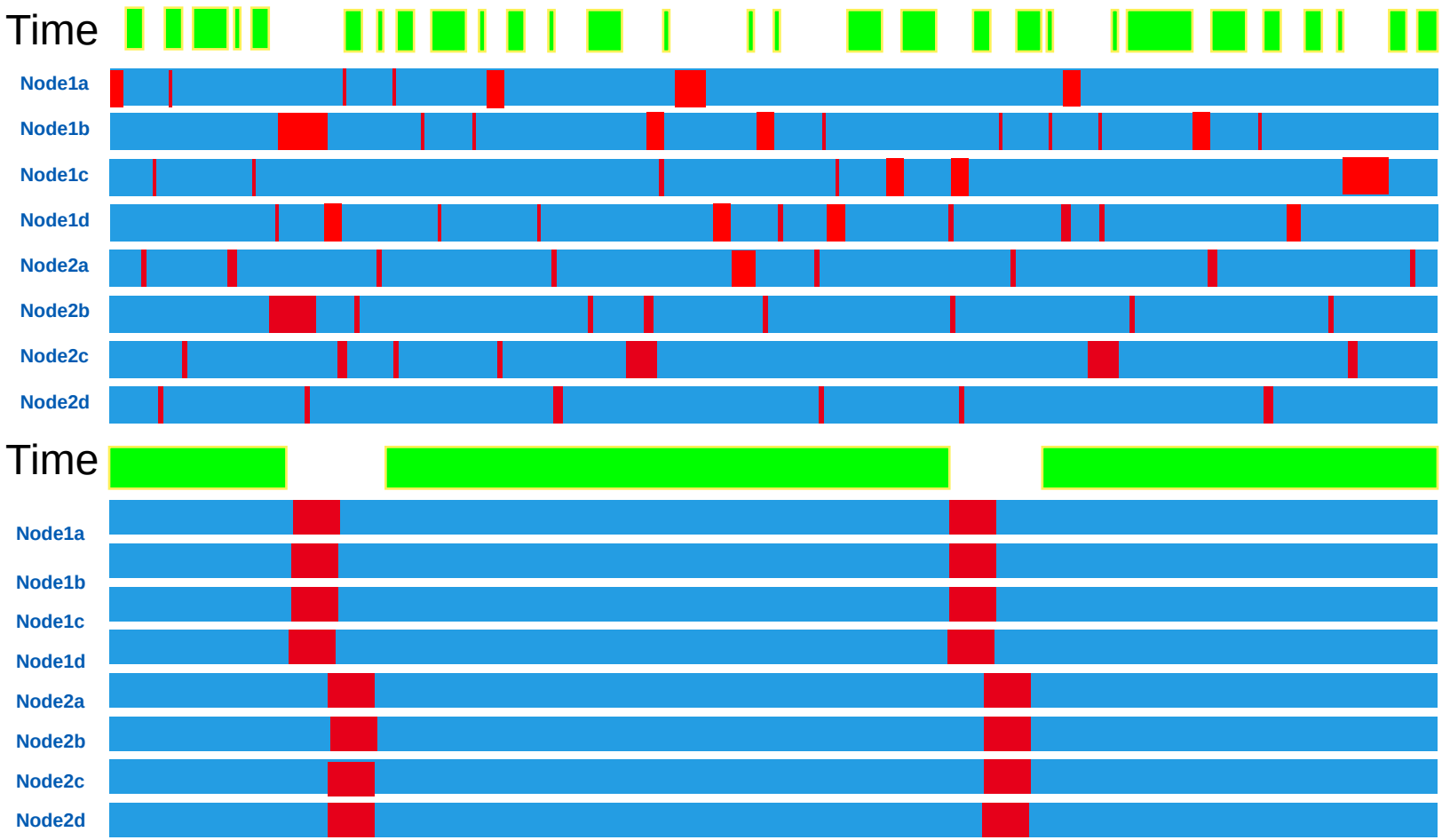
HPC Colony technology – Coordinated scheduling



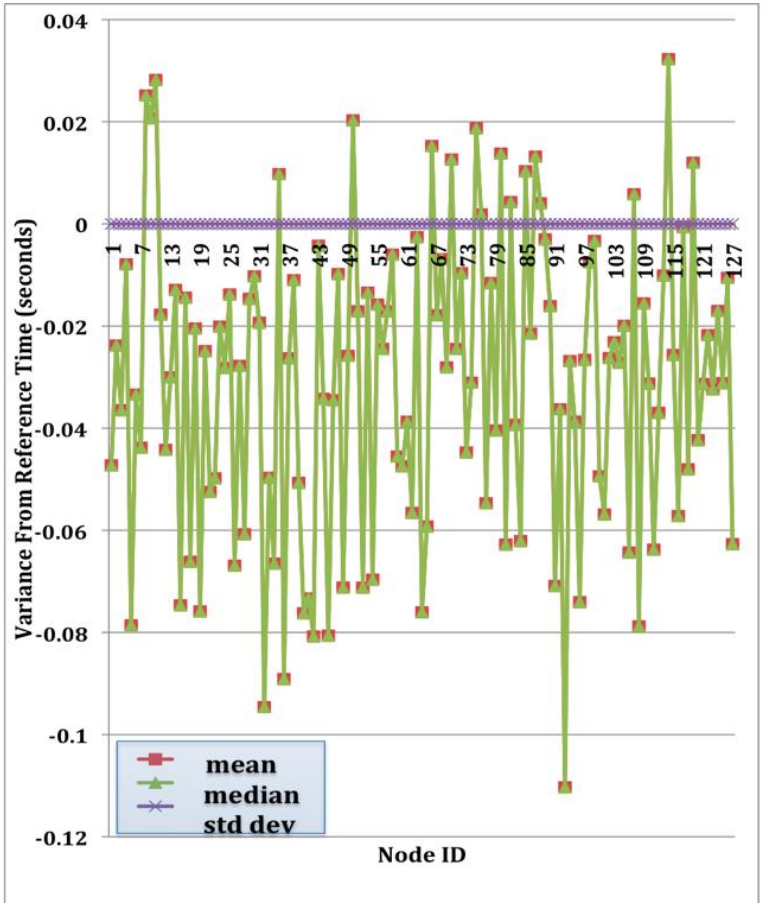
HPC Colony technology – Coordinated scheduling



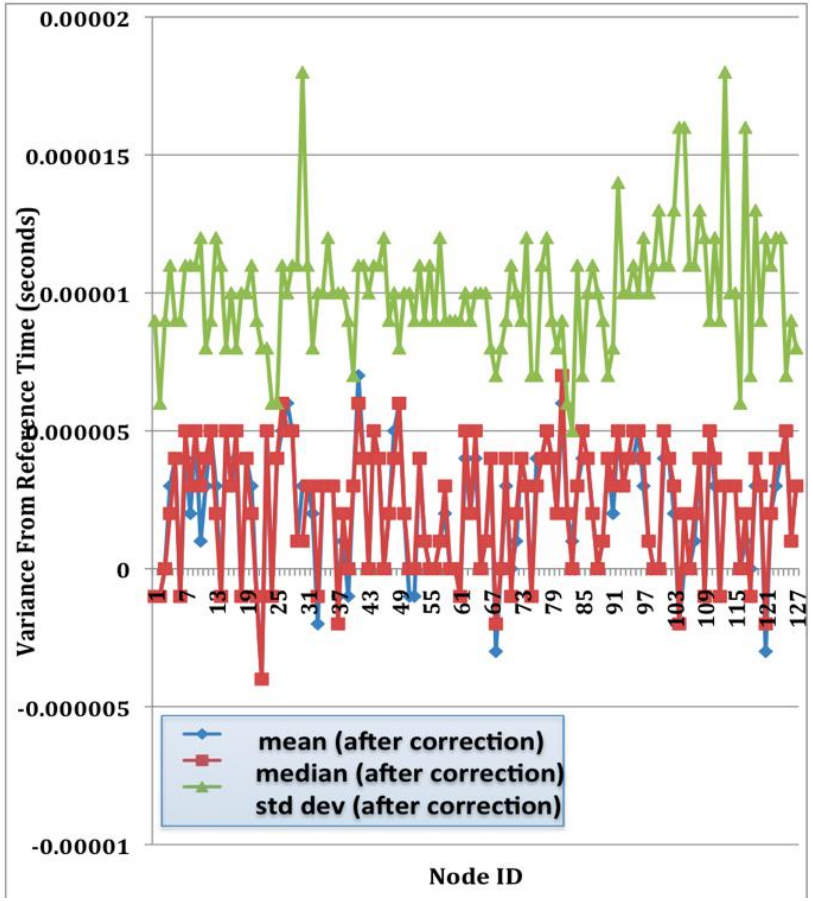
HPC Colony technology – Coordinated scheduling



HPC Colony technology – High precision global time coordination

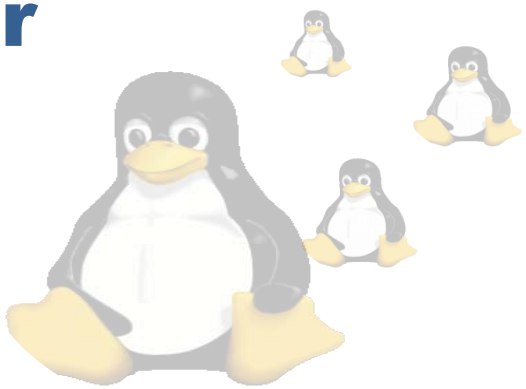


Global time variance under NTP



Global time variance under Colony

HPC Colony summary for year



Accomplishments

- **Recent publications or presentations**
  - Terry Jones, Gregory Koenig, *A Clock Synchronization Strategy for Minimizing Clock Variance at Runtime in High-End Computing Environments*, Proceedings of 22nd International Symposium on Computer Architecture and High Performance Computing, Rio de Janeiro, Brazil, October 2010.
  - Terry Jones, Andrew Tauferner, and Todd Inglett. *Linux OS Jitter Measurements at Large Node Counts using a BlueGene/L*, Technical Report ORNL/TM-2009/303, Oak Ridge National Laboratory, November 2009.
  - Josh Thompson, David W. Dreisigmeyer, Terry Jones, Michael Kirby, and Josh Ladd. *Accurate Fault Prediction of BlueGene/P RAS Logs Via Geometric Methods*, First International Workshop on Fault-Tolerance for HPC at Extreme Scale, Chicago, June 2010.
  - Esteban Meneses, Celso L. Mendes, and Laxmikant V. Kalé. *Team-Based Message Logging: Preliminary Results,* 3rd Workshop on Resiliency in High Performance Computing (Resilience) in Clusters, Clouds, and Grids (CCGRID 2010), Melbourne, Australia, May 2010.
  - Abhinav Bhatele, Eric Bohm and Laxmikant V. Kalé, "Optimizing Communication for Charm++ Applications by Reducing Network Contention," accepted for publication in Concurrency and Computation: Practice and Experience (EuroPar special issue), 2010.
  - Gengbin Zheng, Esteban Meneses, Abhinav Bhatele and Laxmikant V. Kalé, "Hierarchical Load Balancing for Large Scale Supercomputers," Accepted at the Third International Workshop on Parallel Programming Models and Systems Software for High-End Computing (P2S2), San Diego , CA, September 2010.
  - Yoav Tock, Benjamin Mandler, *SpiderCast: Distributed Membership and Messaging for HPC Platforms: An Architectural Overview and High Level Design,* Colony-II technical report, January 2010.
  - Yoav Tock, Benjamin Mandler, Gennady Laventman, *SpiderCast: Distributed Membership and Messaging for HPC Platforms: Publish-Subscribe and DHT Services High Level Design,* Colony-II technical report, May 2010.
- **Formal software releases**
  - Charm++ v.6.2.0 released March 2010, new version planned
  - Colony version of Linux 2.6.16.54 (with coordinated scheduling) planned
- **Students and postdocs deployed**
  - Esteban Meneses, University of Illinois at Urbana-Champaign
  - Lukasz Wesolowski, University of Illinois at Urbana-Champaign

Progress

- **Success Stories**
  - 2010 INCITE Award
  - Better message logging (512 proc job had 73% reduction in message log volume)
  - New synchronized clock scheme
  - Initial design of Spidercast
  - New DHT service (see Tock10b)
  - Topology aware load balancing proven with OpenAtom
  - Coordinated scheduling proven on BlueGene/L
  - Hierarchical and hybrid load balancing proven and in runtime distribution
- **Next Steps**
  - Coordinated scheduling on Cray-like architectures
  - "Team" based protocols
  - Initial testing of Spidercast overlay publish/subscribe techniques

Contact

Terry Jones
trj@ornl.gov

For further info
http://www.hpc-colony.org
http://charm.cs.uiuc.edu
http://www.research.ibm.com/bluegene

Partnerships and acknowledgments
DOE Office of Science – major funding provided by FastOS 2
Colony Team

Core
Terry Jones (ORNL), José Moreira (IBM), Eliezer Dekel (IBM), Roie Melamed (IBM), Yoav Tock (IBM), Benjamin Mandler (IBM), Laxmikant "Sanjay" Kalé (UIUC), Celso Mendes (UIUC), Esteban Meneses (UIUC), Lukasz Wesolowski (UIUC)

Extended
Bob Wisniewski (IBM), Todd Inglett (IBM), Andrew Tauferner (IBM), Edi Shmueli (IBM), Gera Goft (IBM), Avi Teperman (IBM), Gregory Chockler (IBM), Sayantan Chakravorty (UIUC)