Adaptable System Software for A Brave New World of Revolutionary Computer Architectures

Presented by

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

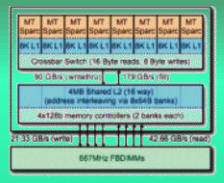
We're experiencing an architectural renaissance

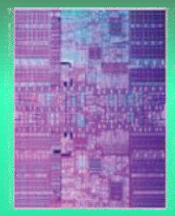
Factors to Change

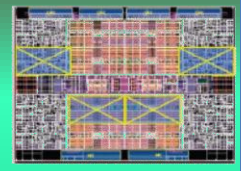
- **Moore's Law – Number of transistors per IC doubles every 24 months**
- **No Power Headroom – Clock speed will not increase (and may decrease) because of power**

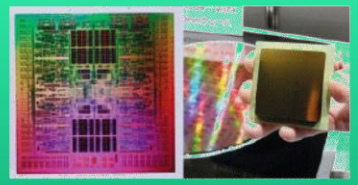
*Power α Voltage² * Frequency*
Power α Frequency
Power α Voltage³

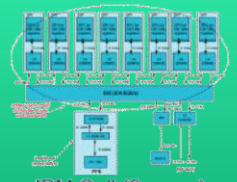
Multicore

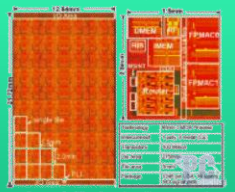

Sun Niagra2 (8 cores)


IBM Power 7 (8 cores)


Intel Xeon(8 cores)

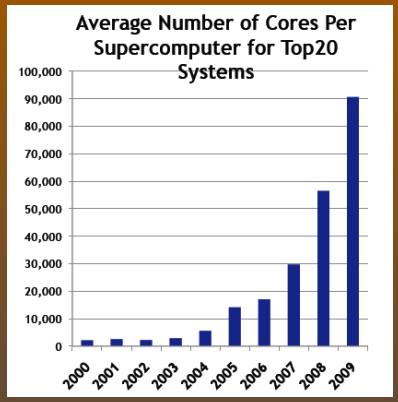

Fujitsu Venus (8 c


IBM Cell (9 cores)

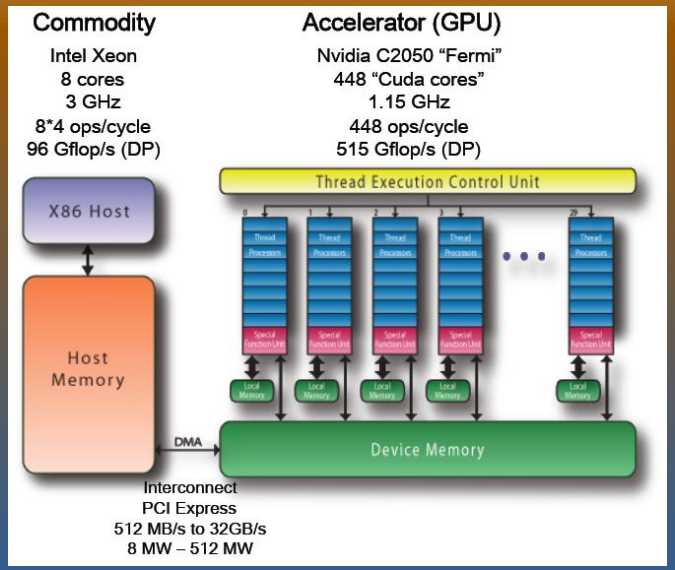

Intel Polaris [experimental] (80 cores)


AMD Istambul (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

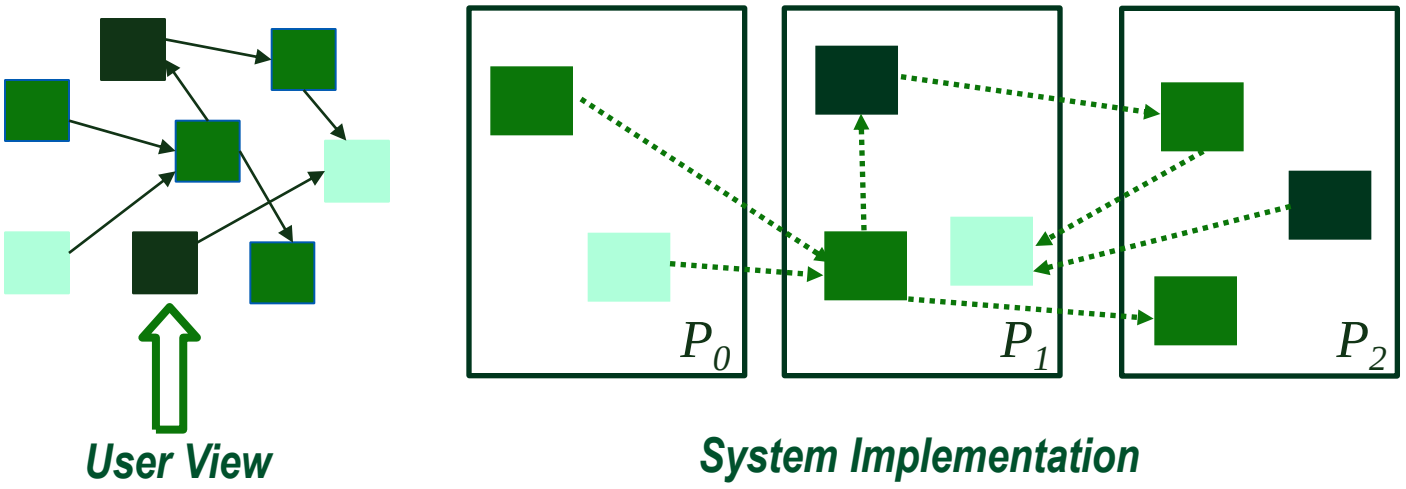
- Computational work often includes large amounts of state which places additional demands on successful work migration schemes
- For widespread acceptance from the Linux community, the effort to validate and incorporate HPC-originated advancements into the Linux kernel must be minimized

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)



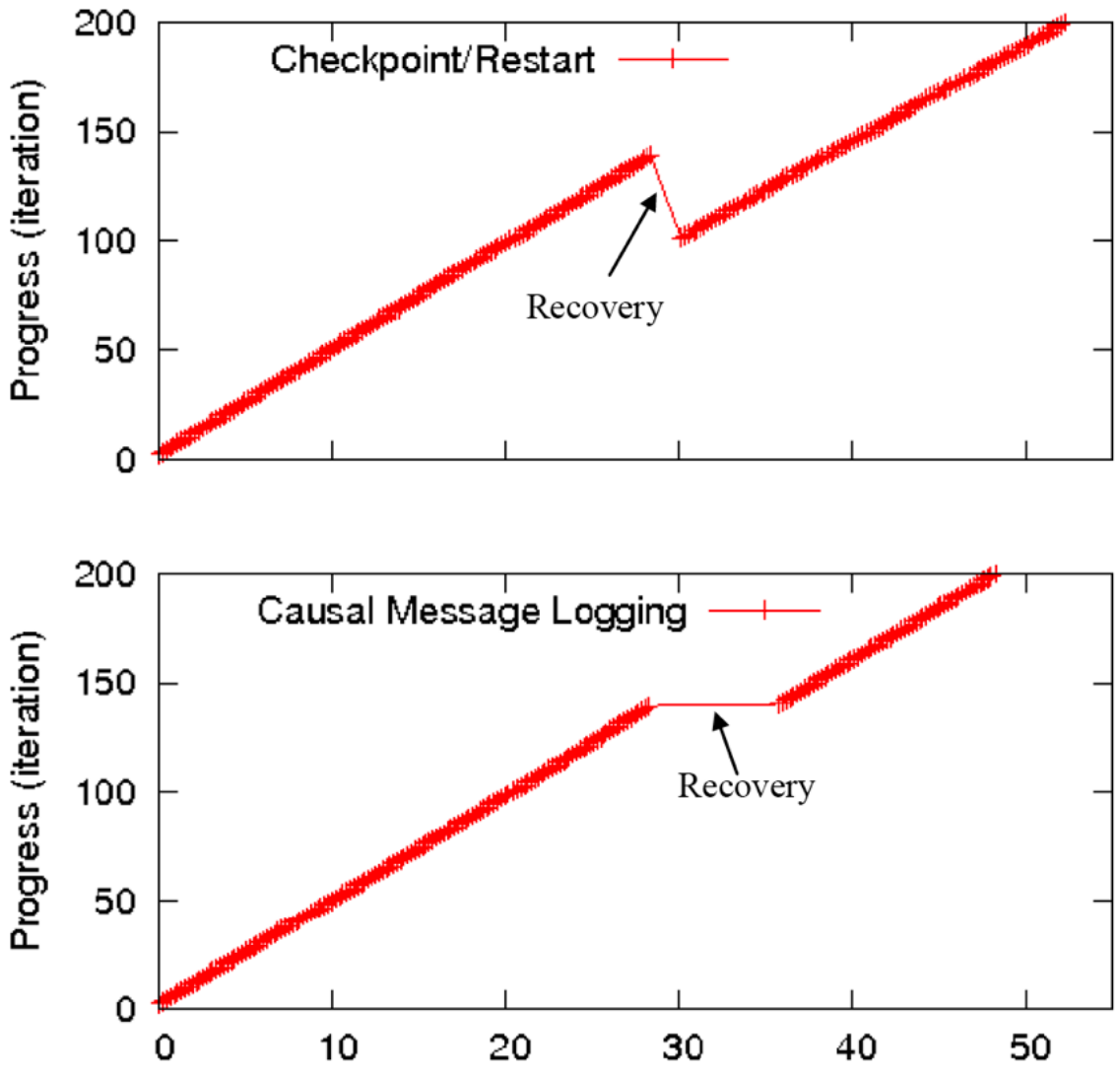
HPC Colony technology – Fault tolerance enabled by Charm++

- 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

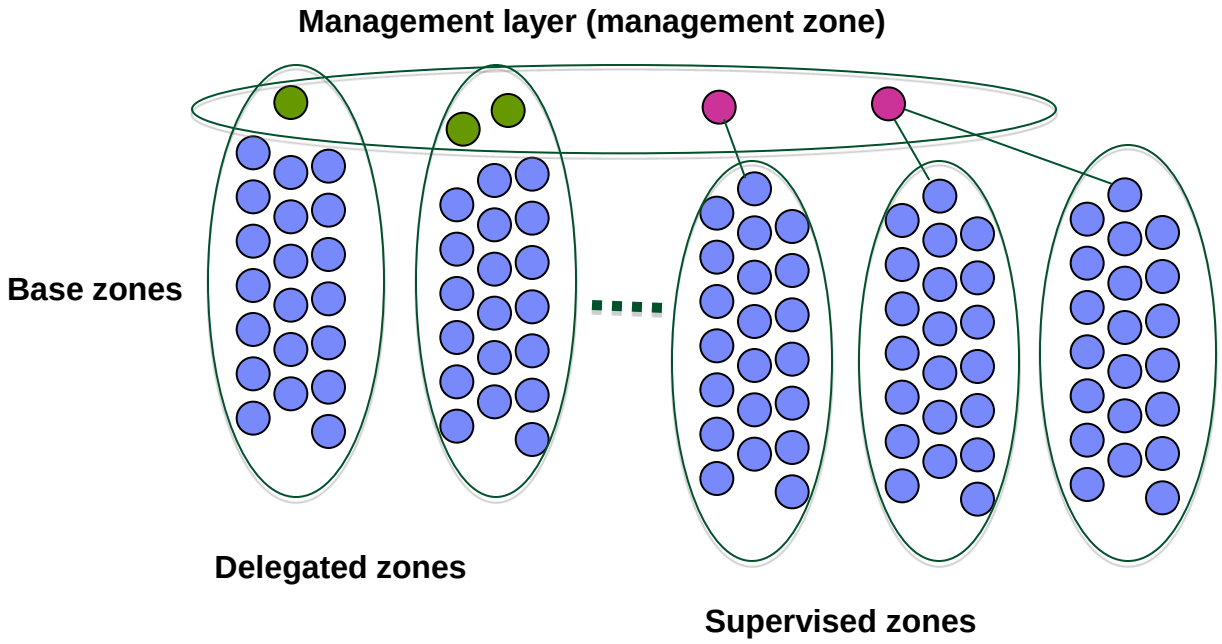
- A scalable, fully distributed, messaging, membership and monitoring infrastructure
- Develop a stand-alone distributed infrastructure that will utilize peer-to-peer and overlay networking technologies, while utilizing HPC platform unique features and architecture
- Focus on:
  - Membership – report 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

Recovery of Failure (7-point stencil)

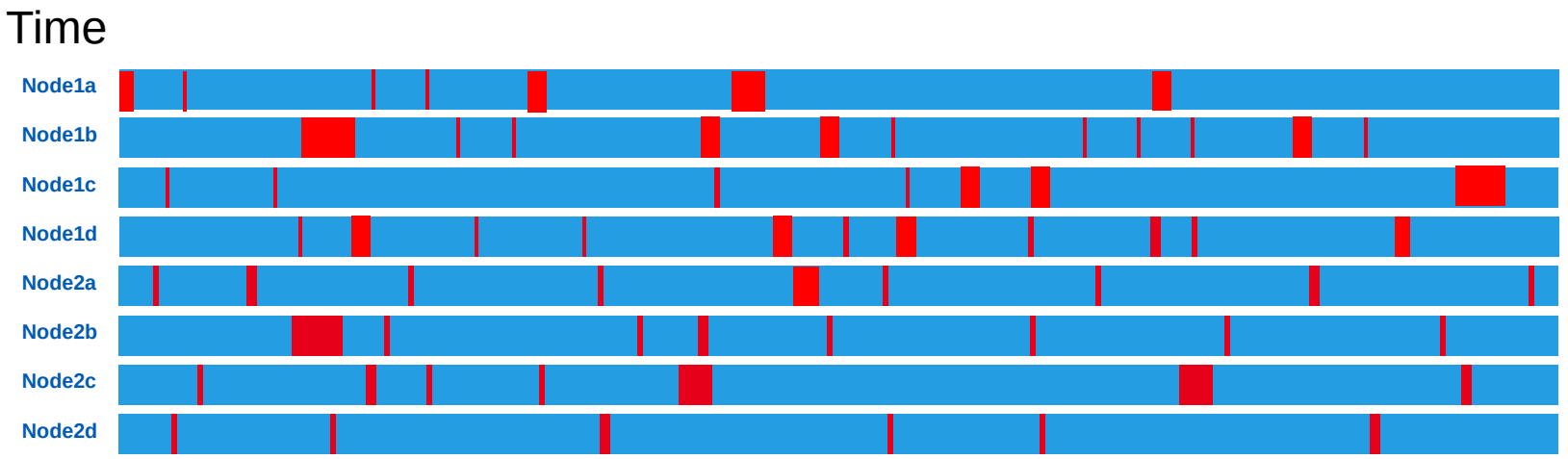


HPC Colony technology – New hierarchical communications schemes

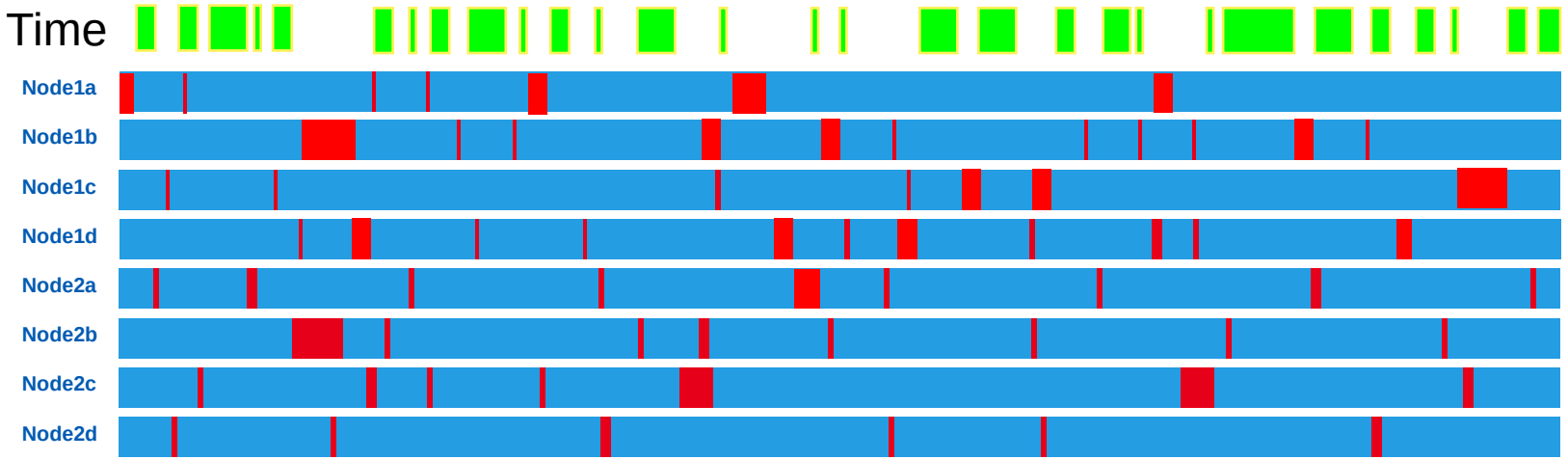
- Multiple base zones federated by a management layer
- Nodes in management layer form a zone, too
- Management nodes either delegates or supervisors
- Support for ~1000 zones x ~1000 members = 1M nodes



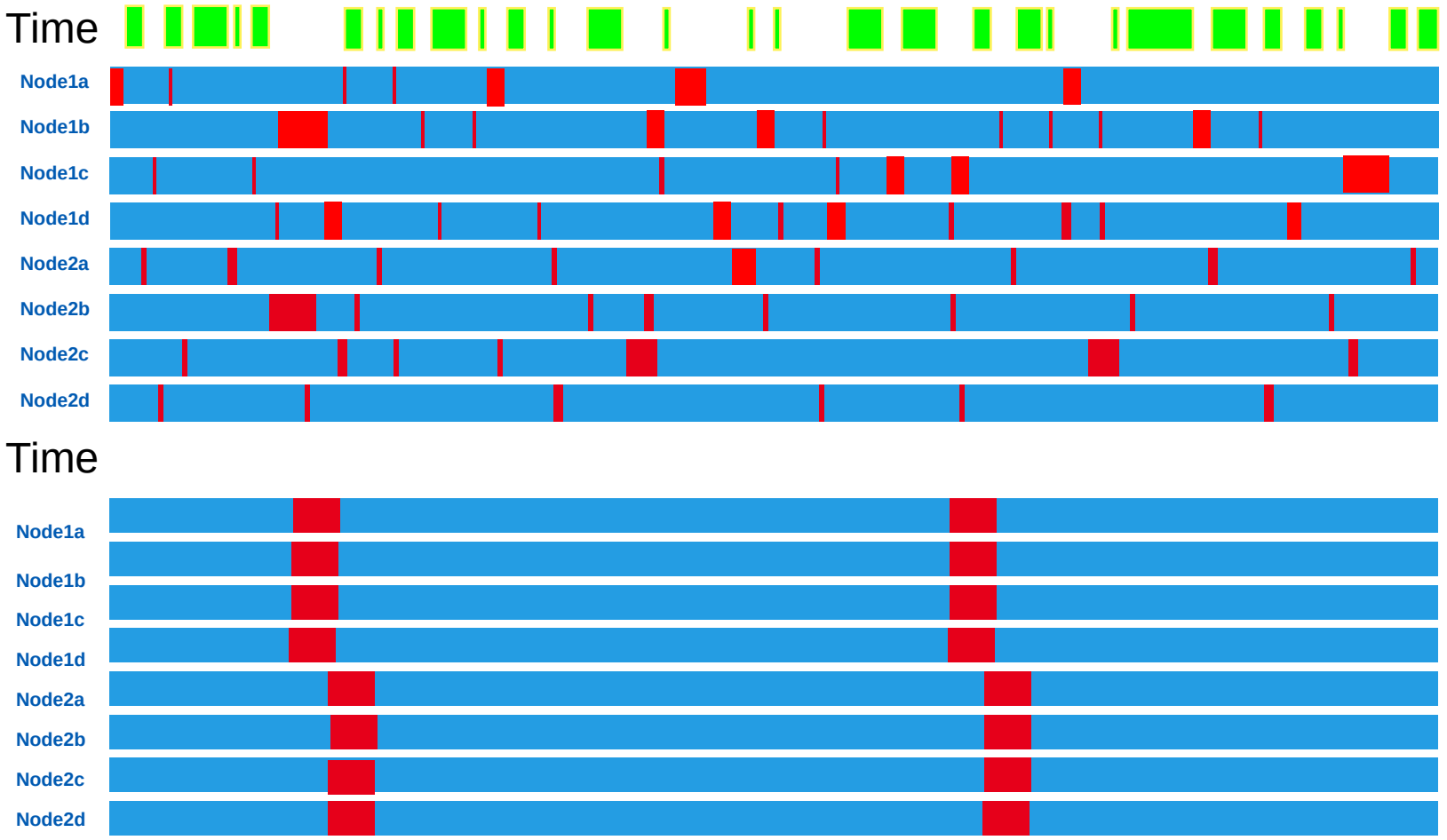
HPC Colony technology – Coordinated scheduling



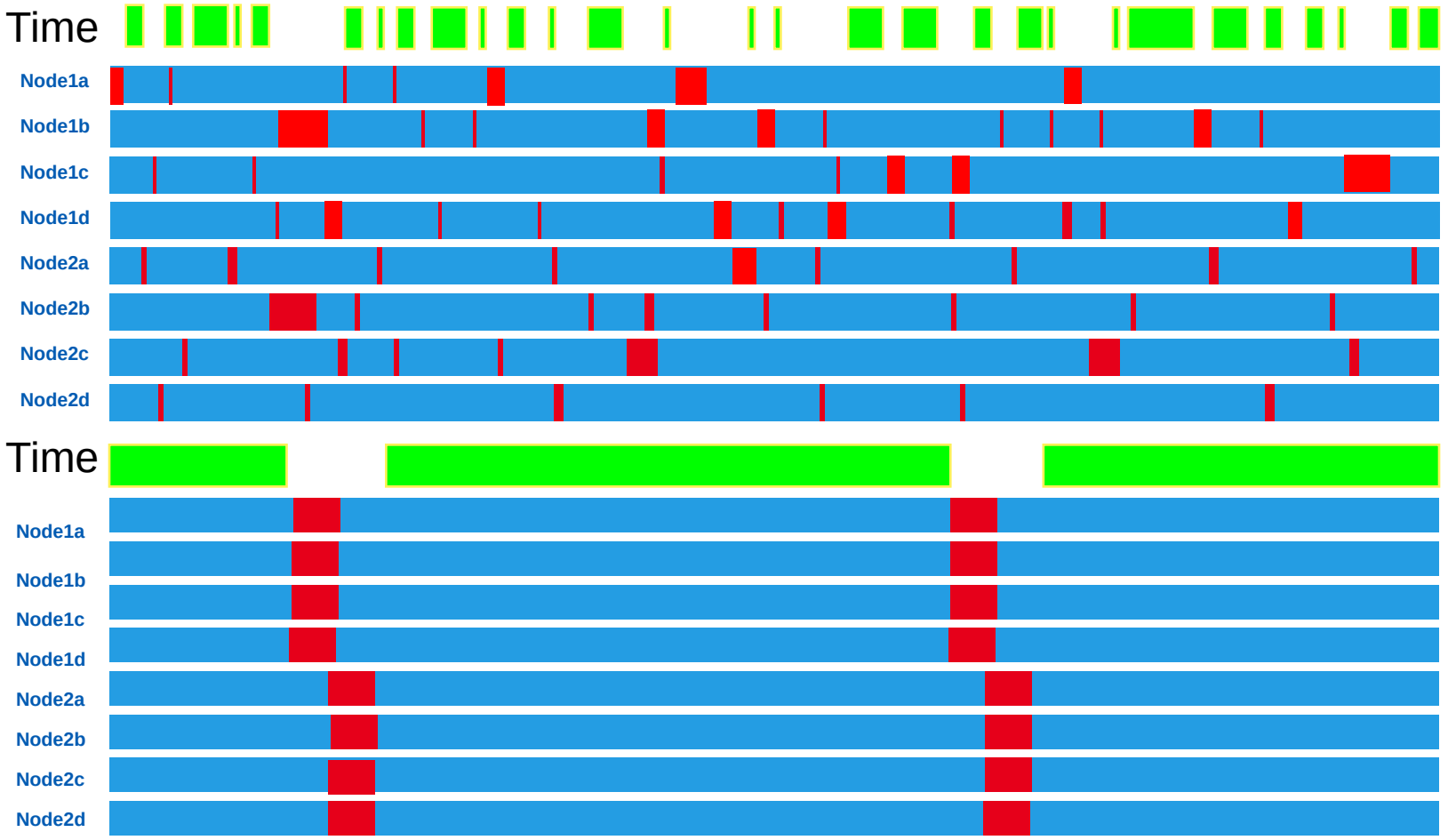
HPC Colony technology – Coordinated scheduling



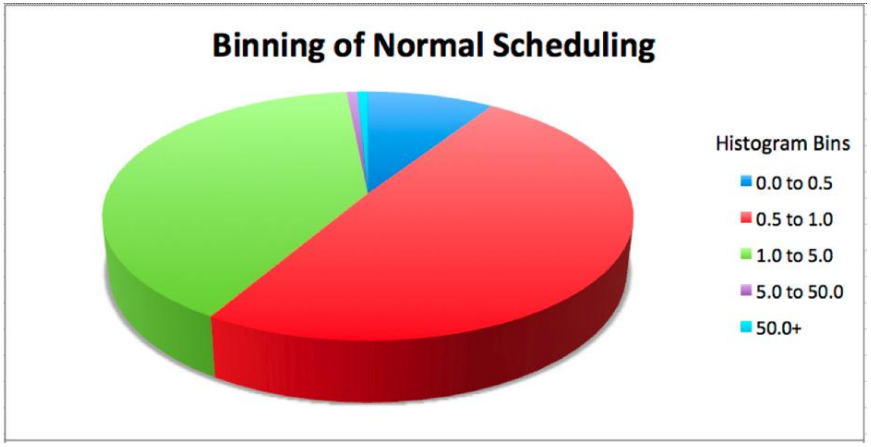
HPC Colony technology – Coordinated scheduling

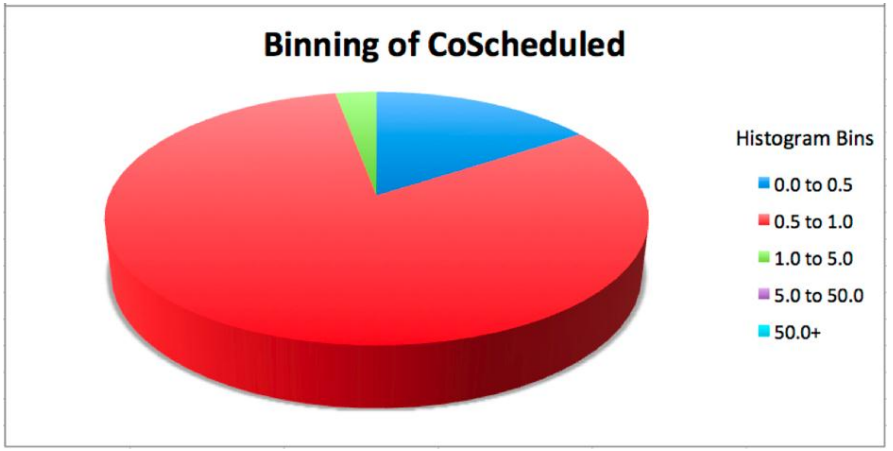


HPC Colony technology – Coordinated scheduling



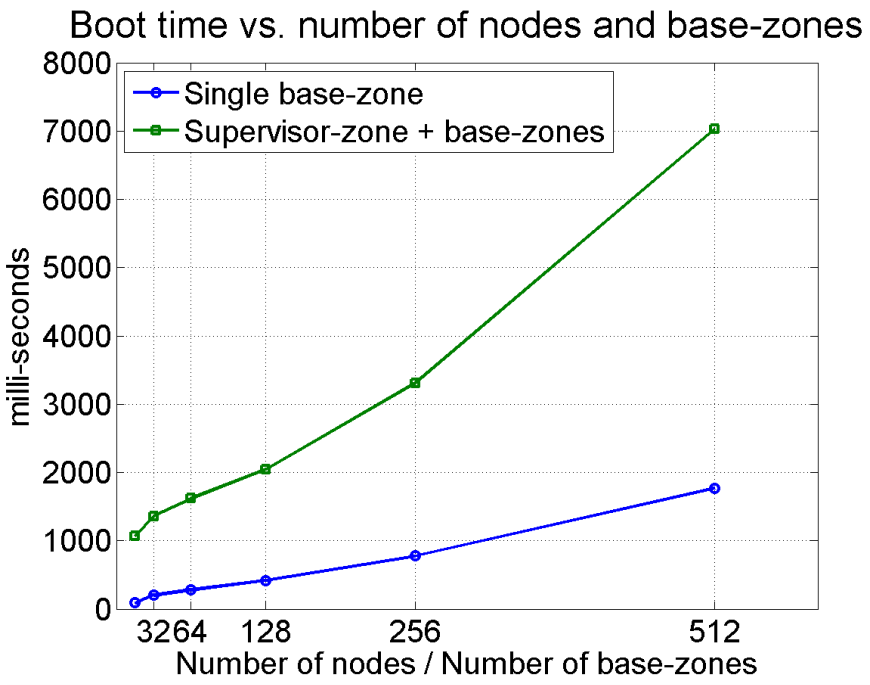
HPC Colony technology –
Improvements from Coordinated Scheduling

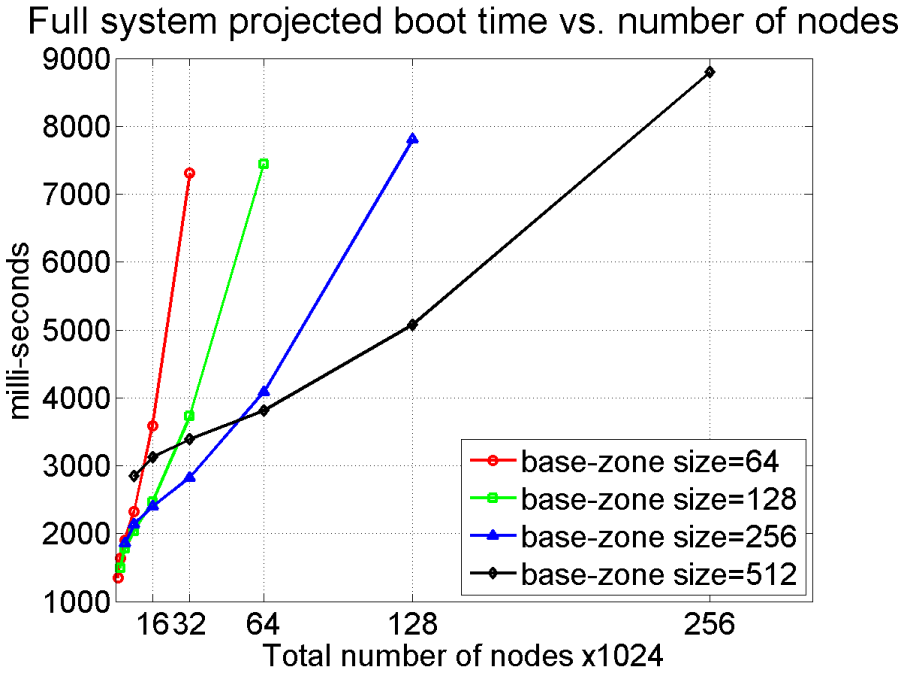




Coordinated and uncoordinated schedulings. The above figure portrays histogram bins in a pie-chart to provide an indication of the relative timing of runs. The top chart gives results without scheduling, and bottom chart gives results for coordinated scheduling.

HPC Colony technology – Membership Services

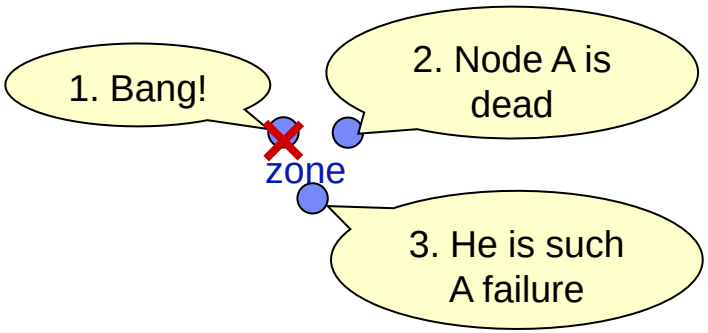


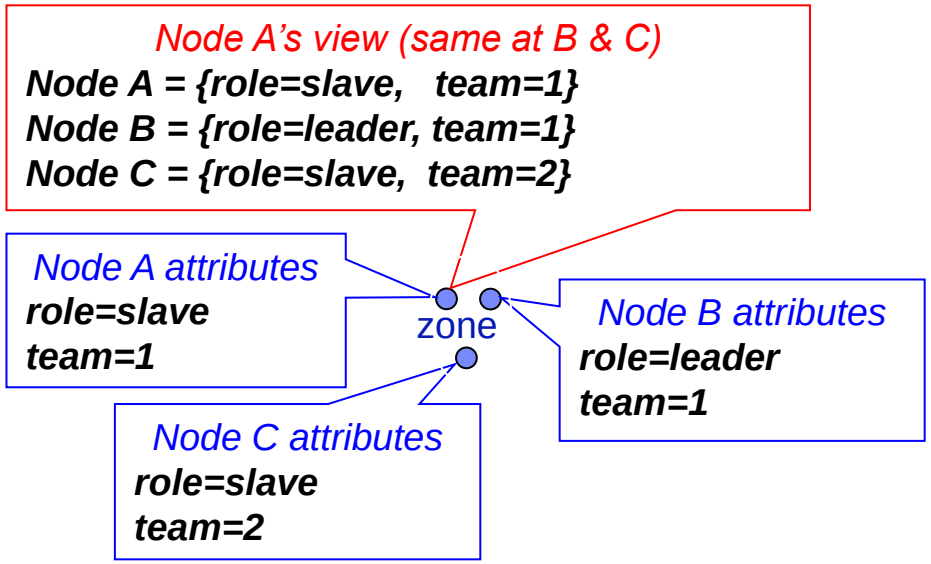


a 512-node zone yields a stable view in $T_{Base}(512) \sim 1.8s$

512-node base-zones x 512-node supervisor zone would boot in $T_{Full}(256K) \sim T_{Sup}(512) + T_{Base}(512) \sim 8.8s$

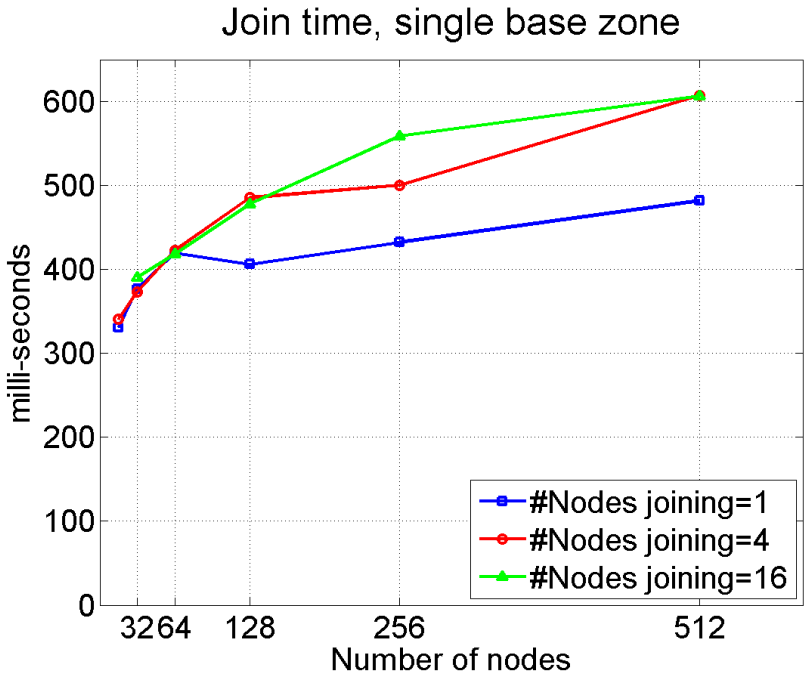
Attribute Service

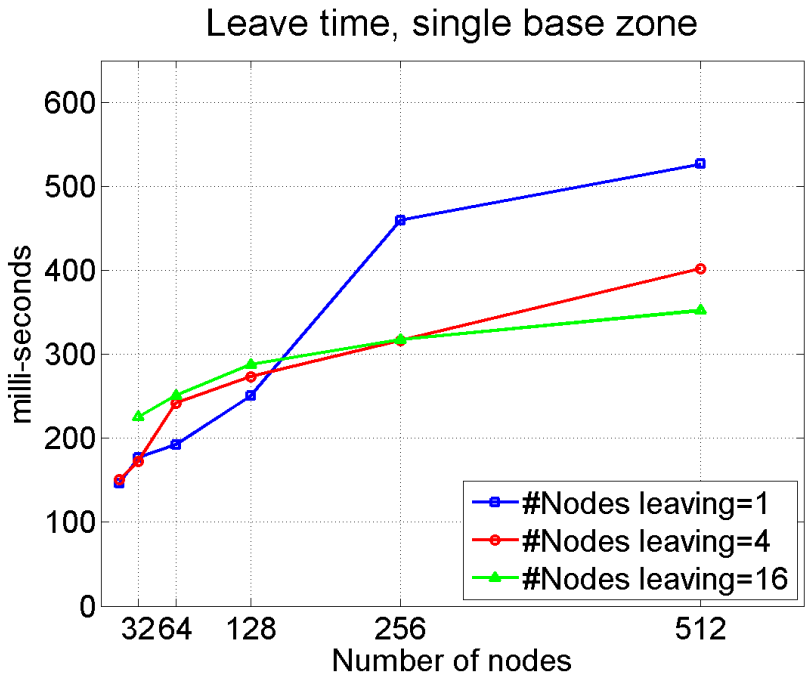




Efficiently replicate slowly changing state information to peer nodes
Supports information sharing between peers to enable discovery of deployed services, supported protocols, server roles, etc.

HPC Colony technology – Membership Services

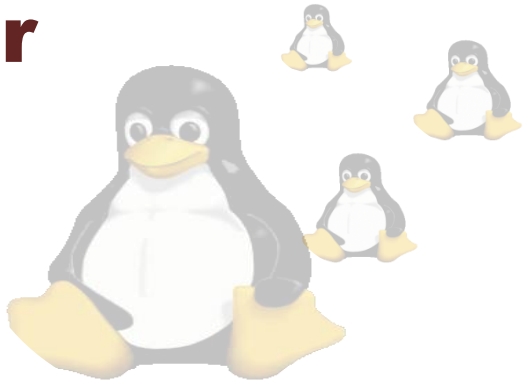




16 concurrent joins to every member at $T_{Join}(512,16)$~0.6s

16 concurrent leaves to every member at $T_{Leave}(512,1)$~0.35s

HPC Colony summary for year



Accomplishments

- **Recent publications or presentations**
  - Celso L. Mendes and Laxmikant V. Kale, "Adaptive MPI", Blue Waters PRAC Fall Workshop, Urbana, October 2010.
  - Abhinav Bhatele, "Mapping your Application on Interconnect Topologies: Effort versus Benefits", George Michael HPC Fellow Presentation at Supercomputing'10, New Orleans, November 2010.
  - Esteban Meneses, "Clustering Parallel Applications to Enhance Message-Logging Protocols", 4th Workshop INRIA-Illinois Joint Laboratory on Petascale Computing, Urbana, November 2010.
  - Eric Bohm, "Scaling NAMD into the Petascale and Beyond", 4th Workshop INRIA-Illinois Joint Laboratory on Petascale Computing, Urbana, November 2010.
  - Eric Bohm, Chao Mei, Yanhua Sun and Gengbin Zheng, "Charm++ Tutorial", Chinese Academy of Sciences, Beijing, China, December 2010.
  - Abhinav Bhatele, "Topology Aware Mapping", University of Illinois (presented by telecom to the Chinese Academy of Sciences", December 2010.
  - Laxmikant V. Kale, "State of Charm++", Charm++ Workshop, Urbana, April 2011.
  - Osman Sarood, "Temperature-Aware Load Balancing for Parallel Applications", Charm++ Workshop, Urbana, April 2011.
  - Abhinav Bhatele, "New Developments in the Charm++ Load Balancing Framework and its Applications", Charm++ Workshop, Urbana, April 2011.
  - Esteban Meneses and Xiang Ni, "Fault Tolerance Support for Supercomputers with Multicore Nodes", Charm++ Workshop, Urbana, April 2011.
  - Eric Bohm, "Charm++ Tutorial", Charm++ Workshop, Urbana, April 2011.
- **Formal software releases**
  - Charm++ v.6.2.1 released, new version planned
  - Colony version of Linux 2.6.16.54 (with coordinated scheduling) demonstrated
- **Students and postdocs deployed**
  - Esteban Meneses, University of Illinois at Urbana-Champaign
  - Yanhua Sun, University of Illinois at Urbana-Champaign

Progress

- **Success Stories**
  - During this period, we developed the first co-scheduling Linux kernel designed for High Performance Computing. A bulk-synchronous-parallel benchmark improved 285% in execution time performance under the new kernel.
  - Developed a new power aware load balancing strategy which has shown improvements for both execution time and power consumption. The new scheme takes advantage of dynamic voltage and frequency scaling (DVFS) hardware capabilities.
  - We completed the initial implementation of a multi-zone scalable membership service as well as the low level design of the new Distributed Hash Table to be used for key-value pairs within SpiderCast.
  - Our new adaptive task mapping strategies show improvements for the Weather Research and Forecasting (WRF) model. For 1,024 nodes, the average hops per byte reduced by 63% and the communication time reduced by 11%..
  - Developed new causal-based message logging scheme with improved performance and scalability.
  - We also completed the design and implementation of a new dynamic load-balancing technique. Results for the BRAMS weather forecasting model show much higher machine utilization and reduction of mothan 30% in execution time.

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)