# Parallel Session I: Runtime Systems

**Panelists**
Greg Bronevetsky
Zoran Budimlic
Paul Hargrove
Hartmut Kaiser
Rishi Khan
Sriram Krishnamoorthy
Olivier Tardieu

**Moderator:** Vivek Sarkar

March 21, 2013

# Context

- Exascale systems will impose a fresh set of requirements on runtime systems including
  - targeting nodes with hundreds of homogeneous and heterogeneous cores
  - energy, data movement and resiliency constraints within and across nodes.
- This session will focus on the fundamental research challenges that need to be addressed in the area of runtimes for exascale systems. *Both panelists and audience members are expected to play an active role in the discussion.*

# Agenda

- 1-slide presentations by panelists
  - Intra-node MPI: Greg Bronevetsky
  - Open Community Runtime: Zoran Budimlic
  - GASNet: Paul Hargrove
  - HPX: Hartmut Kaiser
  - SWARM: Rishi Khan
  - TASCEL: Sriram Krishnamoorthy
  - X10 Runtime: Olivier Tardieu

- Discussion

# MPI for Shared Memory Systems

MPI

luable Exascale

progra ning model

+

- Lega , ns
- Explicit nables fine-grained res e management

MPI

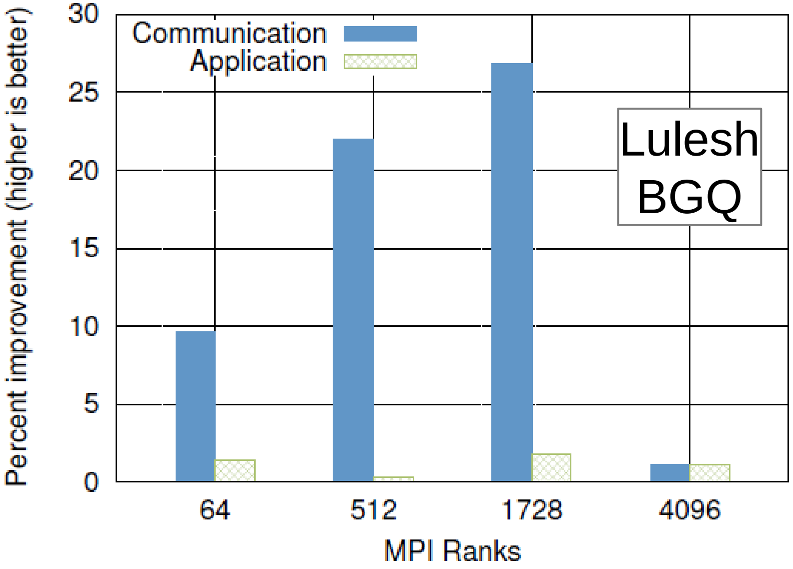
+

MPI

## Approach:

- Share address space of all processes on node
- High performance:
- API extensions
  - Messages are direct copies
  - Synergistic transfer

API extensions



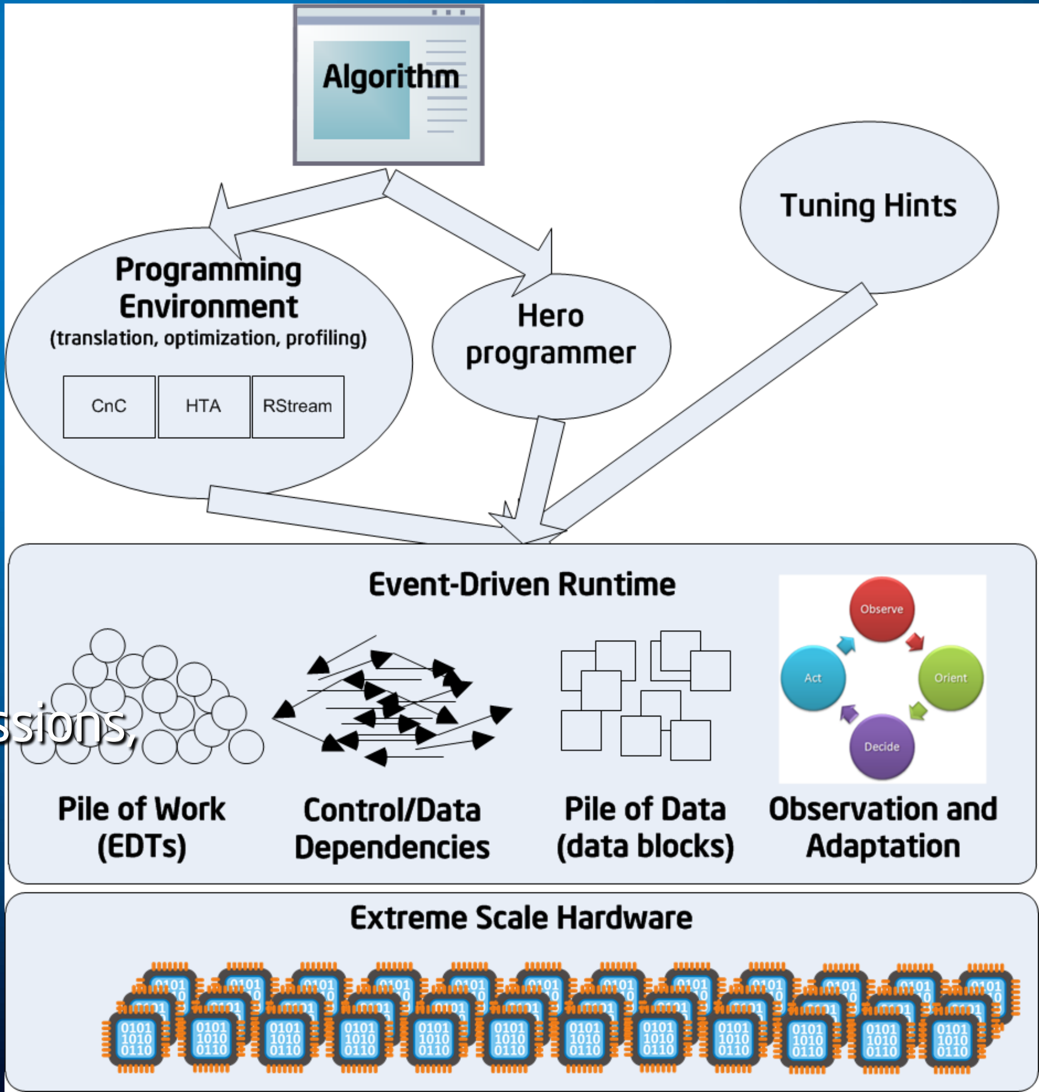
## Goal: high-performance MPI for all device types

- Traditional: inter-node
- Our work: shared memory hardware
- Others: GPU, Phi, …

Andrew Friedley, Greg Bronevetsky, Torsten Hoefler, Andrew Lumsdaine and Dan Quinlan

# Open Community Runtime

- Fine-grained, asynchronous event-driven runtime framework with movable data and computation
- Hosted on 01.org
- Introduced at SC 2012
- Goals
  - Modularity
  - Stable APIs
  - Flexible implementation
  - Transparency
- Development process
  - Continuous integration
  - Quarterly milestones
  - Mailing lists for technical discussions, build status, etc
- Organization
  - Steering Committee
  - Core Team



Algorithm

Programming Environment
(translation, optimization, profiling)

CnC | HTA | RStream

Hero programmer

Tuning Hints

Event-Driven Runtime

Pile of Work (EDTs)

Control/Data Dependencies

Pile of Data (data blocks)

Observation and Adaptation

Extreme Scale Hardware

# GASNet

## GASNet: 1999 to present

- “**G**lobal **A**ddress **S**pace **Net**working”
- API for implementing PGAS languages/libraries (UPC, CAF, Chapel, OpenSHMEM, Ti, and others)
- for compilers and low-level code authors
- widely portable
- MPI-interoperable on most platforms
- performs comparably to (and often better than) MPI send-recv
- has influenced MPI-3 design for one-sided operations (a.k.a. RMA)
- Key API Features include…
  - a rich set of one-sided Put/Get interfaces mapping well to modern RDMA-capable network h/w
  - Active Messages (a.k.a. “Function Shipping” or “Remote Procedure Call”)
    providing powerful mechanism for implementing language-specific features

## GASNet-EX: present and future

- Part of the DEGAS project
- A re-design & re-implementation for an **EX**ascale PGAS environment:
  - Numerous complex nodes
  - Constrained by memory and power
  - Advanced asynchronous clients and multi-client (e.g. UPC+CAF)
  - Resilient implementation with support for resilient clients
- Will support current and future DoE supercomputers
  - Dropping legacy support to improve maintainability
- Apply the lessons learned from GASNet work, including feedback from current and potential clients (Rice, UofH, Cray, IBM …)

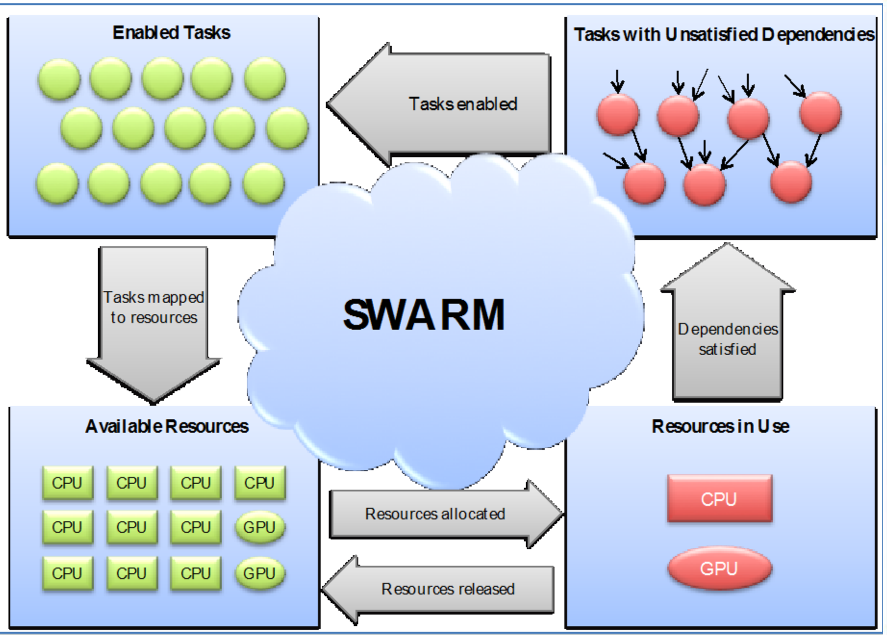
# What’s HPX ?

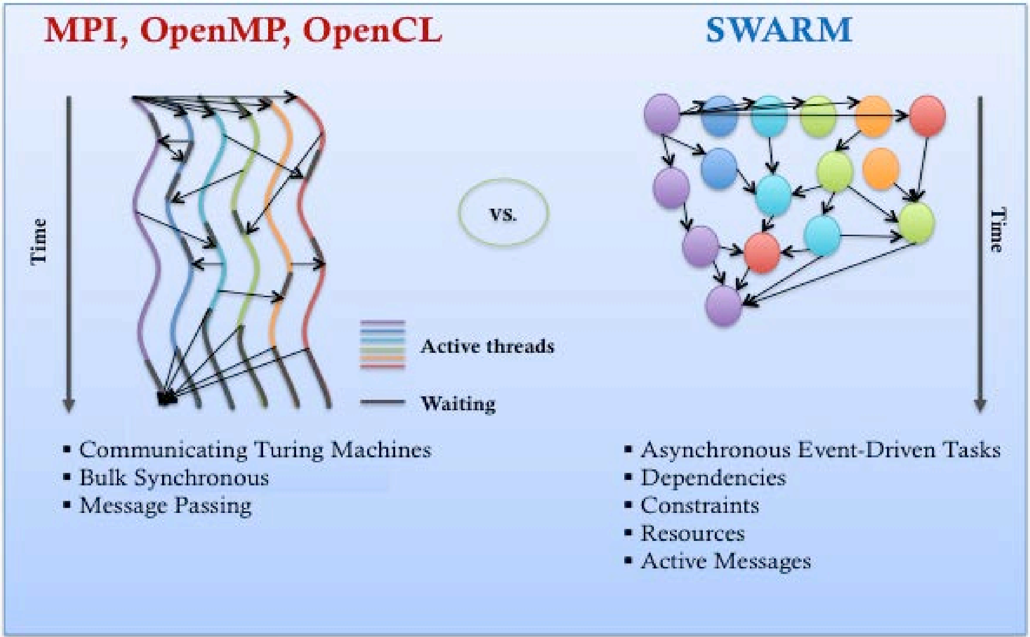
STE||AR

stellar.cct.lsu.edu

- Prefers:
  - Active global address space (AGAS) over PGAS
  - Message driven computation over message passing
  - Lightweight control objects over global barriers
  - Latency hiding over latency avoidance
  - Adaptive locality control over static data distribution
  - Moving work to data over moving data to work
  - Fine grained parallelism of lightweight threads instead of Communicating Sequential Processes
- Open source (github, Boost License)

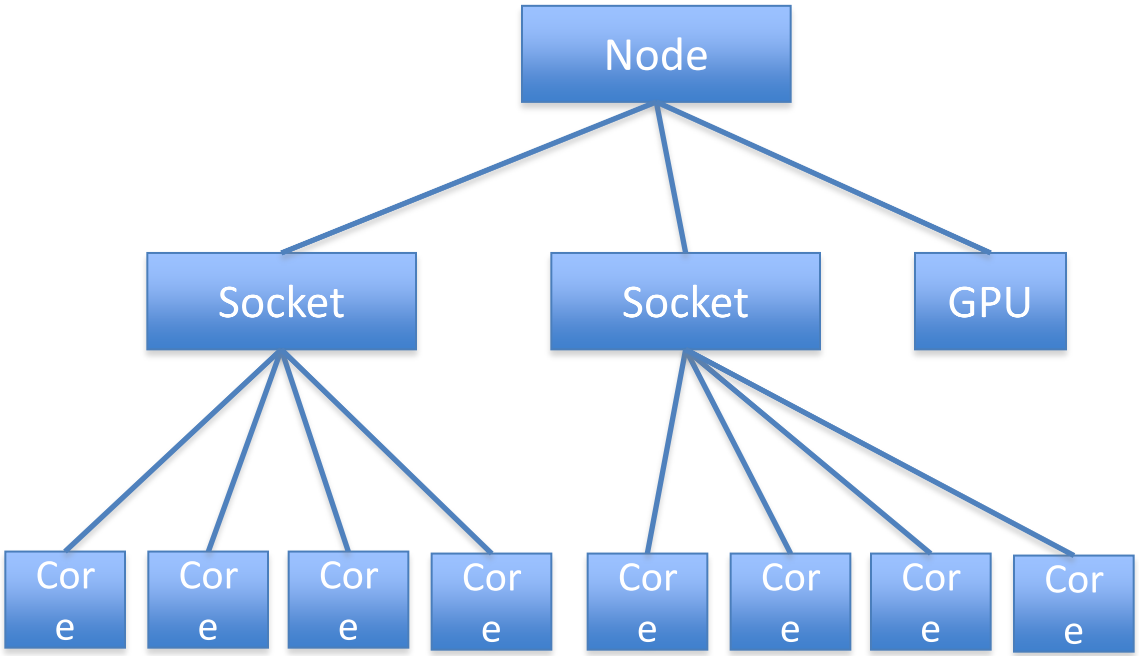
# SWARM: (SWift Adaptive Runtime Machine)

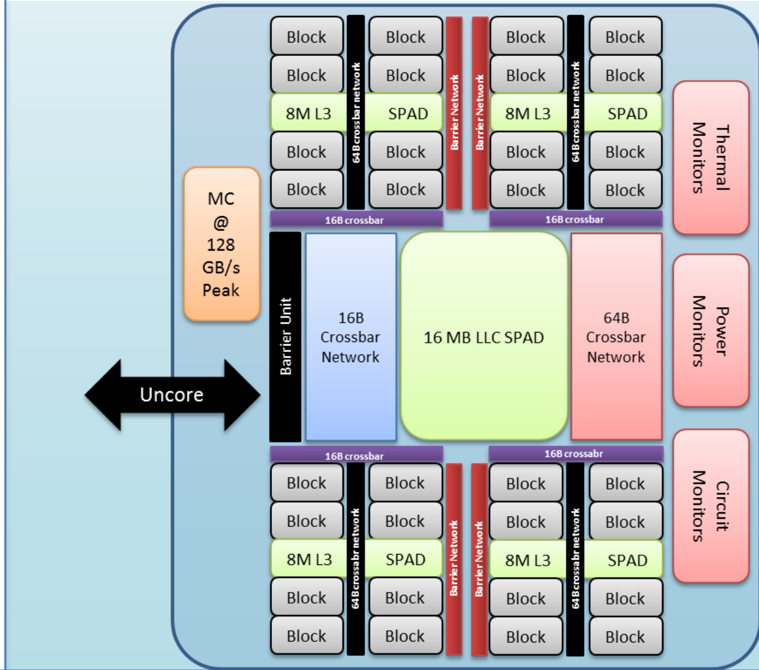




- Communicating Turing Machines
- Bulk Synchronous
- Message Passing

- Asynchronous Event-Driven Tasks
- Dependencies
- Constraints
- Resources
- Active Messages





# TASCEL

- Runtime to study algorithms supporting finer-grained concurrency
  - Scheduling and load balancing
  - Resilience
  - …

- Marketplace session:
  - Using TASCEL
  - Algorithms developed with TASCEL
    - Retentive stealing, persistence-based load balancing
    - Data-driven fault tolerance
    - Tracing work stealing and its applications

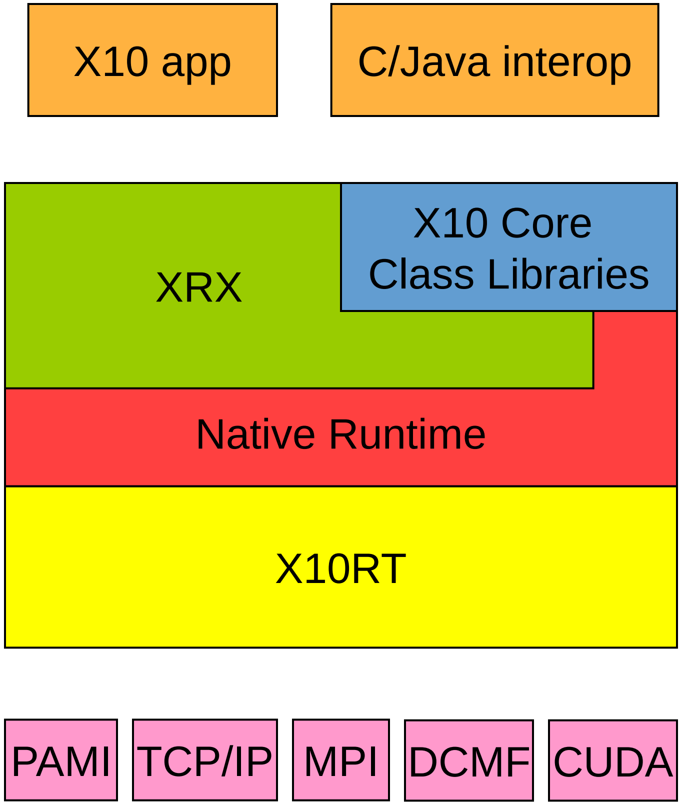
# X10 Runtime

- Open source
- Scales [HPCC’12]
- Implements APGAS
  - PGAS + async + finish + when

- X10 runtime transport
- X10 runtime in X10
  - compiles to C++ and Java

Roadmap to Exascale

- interop. (C, MPI, ROSE) & DSLs
- >> parallelism (many-cores & accel.)
- elasticity & resilience



# Discussion topics (1/3)

**Questions posed to panelists include:**

1. **How should a runtime system be designed to manage billions of threads?**
2. **How should locality optimization on exascale machines be supported at the runtime level?**
3. **How much of the burden of selecting the right granularity of parallelism for a given platform should be placed on the runtime?**
4. **Can exascale binaries be "forward scalable" by default so that hardware changes in parallelism/locality can be exploited entirely in the runtime without requiring re-programming or re-compilation?**

# Discussion topics (2/3)

5. How will runtime memory management support (e.g., malloc/free) be designed for exascale? Will automatic techniques like concurrent garbage collection be more or less relevant at exascale?
6. How can different runtime components cooperatively manage shared resources? For example, cores can be used to support computation, communication and resilience.
7. How can a runtime support both user directives and automated adaptations in an integrated manner?
8. What role can "relaxed synchronization" play in exascale software e.g., allowing tasks to execute even in the presence of data races so long as the probability of wrong answers is shown to be (or made) low?

# Discussion topics (3/3)

9. What role can transactional memory and related ideas for optimistic concurrency play in exascale software? How about actors?
10. What assumptions should exascale runtimes make about exascale operating systems?

Suggestions for additional topics/questions are most welcome!

# Summary of Discussion and Opinions (1/3)

- Lot of deep technical work under way on runtime systems
  - How best to leverage this in the X-Stack program?
- Application programmers will be interested in targeting runtimes so as to transition to new ideas
  - e.g., simpler to target runtime to get rid of bulk synchronization incrementally rather than learning a new (embedded) DSL
- Vertical integration of multiple components in X-Stack is important but can be challenging
  - e.g., high-level scheduler atop low-level scheduler
  - effective integration of compiler and runtime will be very important for X-stack

# Summary of Discussion and Opinions (2/3)

- What is the role of introspection, runtime state, application-dependent policies, and of cost models?
  - Related panel question: How can a runtime support both user directives and automated adaptations in an integrated manner?
  - Need more experience with application developer interacting with adaptive runtimes e.g., use of turbo mode in modern processors
- Why are we putting all runtimes in one “bucket”?
  - We are discussing synergies among runtime efforts in the X-Stack program
- MPI was not designed for use by programmers
  - Despite the original intent, many programmers use MPI
  - Many programmers also use libraries and frameworks, where MPI is hidden from them

# Summary of Discussion and Opinions (3/3)

- What is the ambition for each of the panelist’s projects?
  - Greg B: MPI+MPI is the answer to MPI+X, for many appropriate applications
  - Zoran B: OCR is a low-level runtime for all programming and execution models for exascale/extreme-scale systems
  - Paul H: GASNet covers all areas that are not handled by MPI
  - Hartmut K: HPX execution model and runtime enables efficient support for strong scaling
  - Rishi K: SWARM programming model and runtime helps average developers write applications for distributed heterogeneous systems
  - Sriram K: TASCEL incorporates abstractions from real applications into real programming models
  - Olivier T: X10’s primary goal is to increase programmer productivity at scale

# Technology Marketplace Schedule

- Part 1, 3:30pm – 5:00pm

  Table 1a: Intra-node MPI: Greg Bronevetsky

  Table 1b: Open Community Runtime: Zoran Budimlic

  Table 1c: GASNet: Paul Hargrove

- Part 2, 5:00pm – 6:30pm

  Table 1a: HPX: Hartmut Kaiser

  Table 1a: SWARM: Rishi Khan

  Table 1b: TASCEL: Sriram Krishnamoorthy

  Table 1c: X10 Runtime: Olivier Tardieu