An Early Prototype of an Autonomic Performance Environment for Exascale

Kevin Huck, Sameer Shende,Allen Malony
University of Oregon
Hartmut Kaiser
Louisiana State University
Allan Porterfield, Rob Fowler
RENCI
Ron Brightwell
Sandia National Labs

Outline

- Introduction/Motivation
- XPRESS Overview
- Relevant Components
  - HPX Runtime
  - APEX Concepts
- APEX Prototype
- Experimental Results
- Future Work

Introduction/Motivation

- Extreme-scale computing requires a new perspective on the role of performance observation in the Exascale system software stack
- High concurrency and dynamic operation
- Post-mortem performance measurement and analysis is not good enough
- Requirements:
  - first- and third-person observation
  - *In situ* analysis
  - Introspection across stack layers
  - Dynamic feedback and adaptation

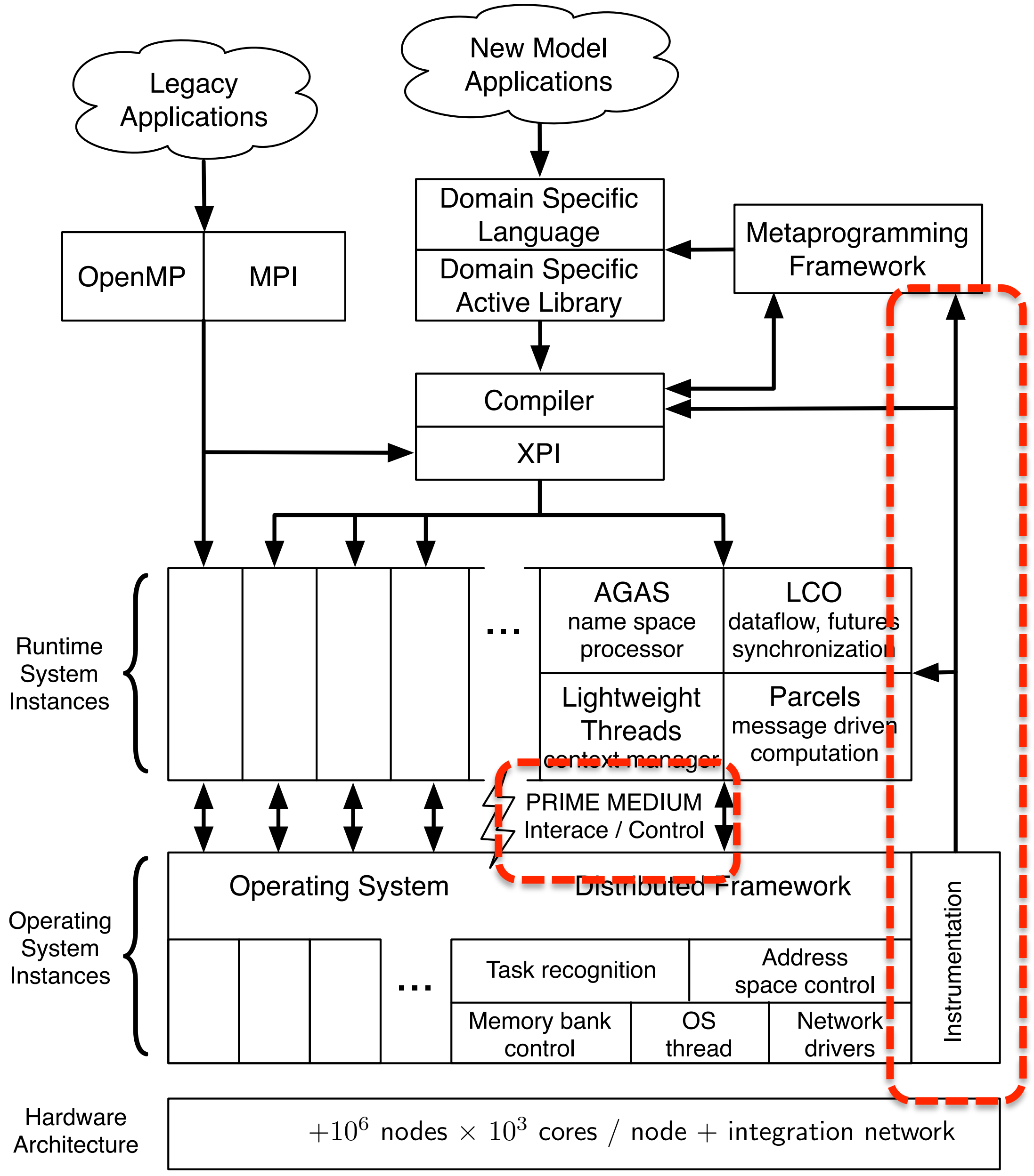
XPRESS

- Runtime system implementing ParalleX, co-designed with an Operating System.
  - Sandia National Laboratories (Ron Brightwell)
  - Indiana University (Thomas Sterling, Andrew Lumsdane)
  - Lawrence Berkeley National Laboratory (Alice Koniges)
  - Louisiana State University (Hartmut Kaiser)
  - Oak Ridge National Laboratory (Chris Baker)
  - University of Houston (Barbara Chapman)
  - University of North Carolina (Allan Porterfield, Rob Fowler)
  - University of Oregon (Allen Malony)
- http://xstack.sandia.gov/xpress/

OpenX

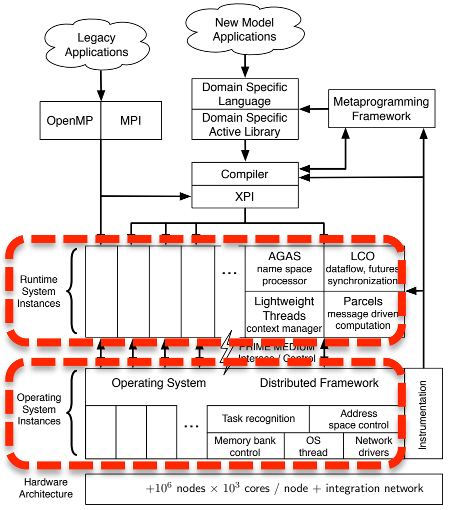
- Integrated Exascale software stack for ParalleX

"APEX"


Legacy Applications
New Model Applications
OpenMP
MPI
Domain Specific Language
Domain Specific Active Library
Metaprogramming Framework
Compiler
XPI
Runtime System Instances
...
AGAS name space processor
LCO dataflow, futures synchronization
Lightweight Threads context manager
Parcels message driven computation
PRIME MEDIUM Interace / Control
Operating System Instances
Operating System
Distributed Framework
...
Task recognition
Address space control
Memory bank control
OS thread
Network drivers
Instrumentation
Hardware Architecture
+10^6 nodes × 10^3 cores / node + integration network

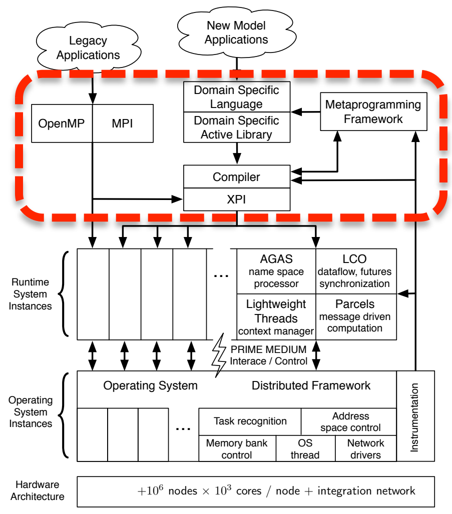
XPRESS: Exascale System Software

- HPX-4: runtime based on OpenX modular software architecture
  - HPX-3: Current ParalleX implementation
- LXK: lightweight kernel OS
  - Based on the Kitten OS


Legacy Applications
New Model Applications
OpenMP
MPI
Domain Specific Language
Domain Specific Active Library
Metaprogramming Framework
Compiler
XPI
Runtime System Instances
AGAS name space processor
LCO dataflow, futures synchronization
Lightweight Threads context manager
Parcels message driven computation
PRIME MEDIUM Interface / Control
Operating System Instances
Operating System
Distributed Framework
Task recognition
Address space control
Memory bank control
OS thread
Network drivers
Instrumentation
Hardware Architecture
$+10^6$ nodes $\times$ 10^3 cores / node + integration network

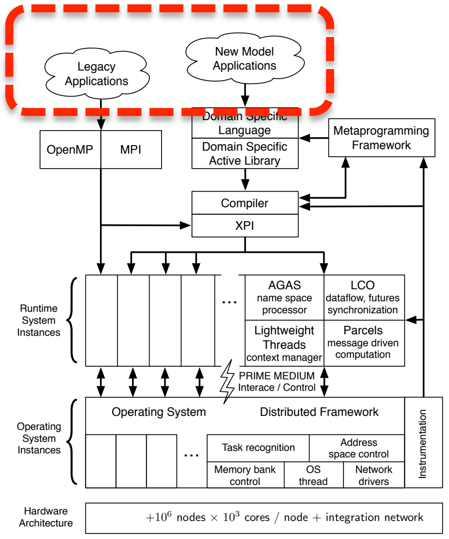
XPRESS: Programming Models and Languages

- XPI – low-level imperative programming interface
- Meta-programming framework to support rapid deployment of future embedded DSL formalisms
  – Example DSL using meta-DSL toolkit
- Legacy mitigation through MPI & OpenMP programming interfaces to the OpenX software stack


Legacy Applications
New Model Applications
OpenMP
MPI
Domain Specific Language
Domain Specific Active Library
Metaprogramming Framework
Compiler
XPI
Runtime System Instances
AGAS name space processor
LCO dataflow, futures synchronization
Lightweight Threads context manager
Parcels message driven computation
PRIME MEDIUM Interace / Control
Operating System Instances
Operating System
Distributed Framework
Task recognition
Address space control
Memory bank control
OS thread
Network drivers
Instrumentation
Hardware Architecture
$+10^6$ nodes $\times$ 10^3 cores / node + integration network

XPRESS: Applications

- Climate Science
  - Community Earth System Model (CESM)
- Nuclear Energy
  - Denovo
- Plasma Physics
  - GTC (HPX implementation)
- Trilinos libraries


Legacy Applications
New Model Applications
Domain Specific Language
Domain Specific Active Library
Metaprogramming Framework
OpenMP
MPI
Compiler
XPI
Runtime System Instances
AGAS name space processor
LCO dataflow, futures synchronization
Lightweight Threads context manager
Parcels message driven computation
PRIME MEDIUM Interace / Control
Operating System Instances
Operating System
Distributed Framework
Task recognition
Address space control
Memory bank control
OS thread
Network drivers
Instrumentation
Hardware Architecture
$+10^6$ nodes $\times$ 10^3 cores / node + integration network

XPRESS: Crosscutting Issues

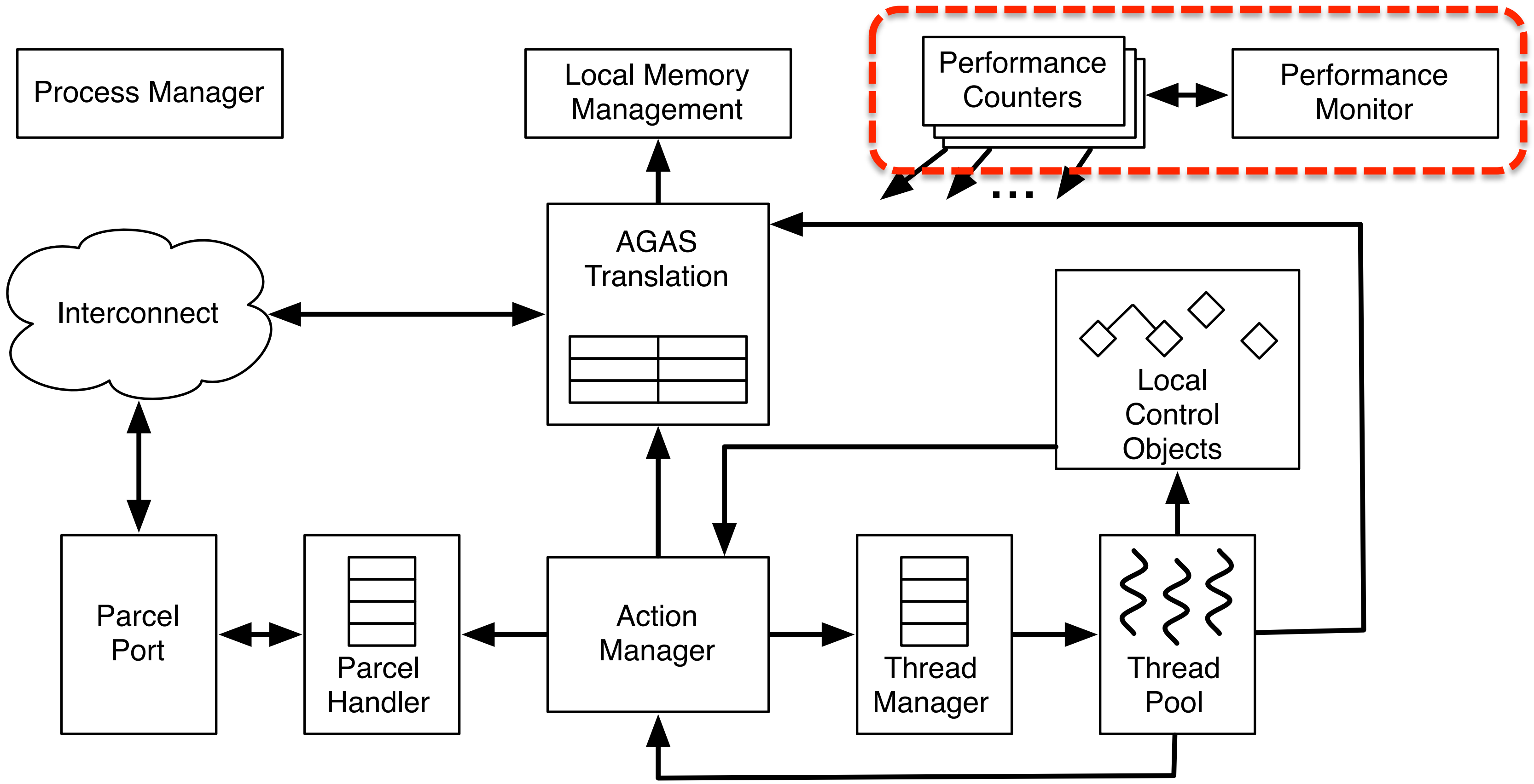
- Automatic control and introspection
- Resilience
  - In memory checkpointing, compute-validate-commit cycle that defers side-effects (isolating error propagation)
- Power management
  - Greater resource utilization per Joule while reducing data movement through *active locality management*
- Heterogeneity

HPX

- First open-source implementation of ParalleX
- Modular runtime system for conventional architectures (NUMA machines, clusters)
- Supports all key ParalleX paradims
  - Parcels & parcel transport layer
  - PX-threads and management
  - Local Control Objects (LCO)
  - Active Global Address Space (AGAS)

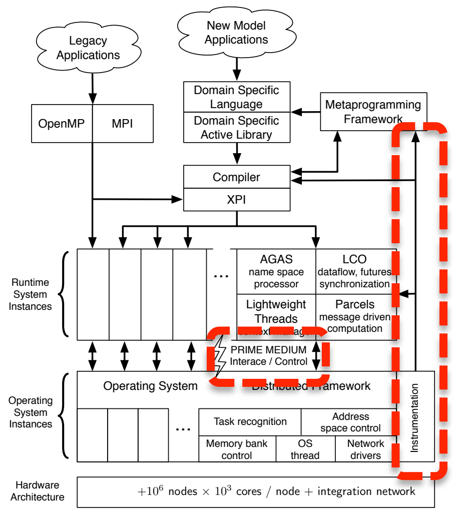
HPX-3 architecture

Thread queue length, PAPI, etc.



APEX: Autonomic Performance Environment for eXascale

- Node, core count, heterogeneity all increasing
- Cores interacting through shared resources
  - Manifested by bottlenecks and queuing at scarce resources both on chip (node) and between nodes
- Measurement and analysis need to be available in real time to all levels of software stack



APEX: Autonomic Performance Environment for eXascale

- Performance observation requirements had been satisfied by local measurements and offline optimization (focus on *first person*)
- Inadequate for exascale
  - Shared resources operate independently of the cores that have hardware monitors built in
- Requirement for *third person* observation
- Make node-wide resource utilization data *and* analysis, energy consumption, health available in real time

Autonomic Performance

- Top down and bottom up performance mapping
- OS (LXK) tracks system resource assignment, utilization, job contention, overhead
- Runtime (HPX) tracks threads, queues, concurrency, remote operations, parcels, memory management
- ParalleX, DSLs and Legacy codes allow language-level performance semantics to be measured

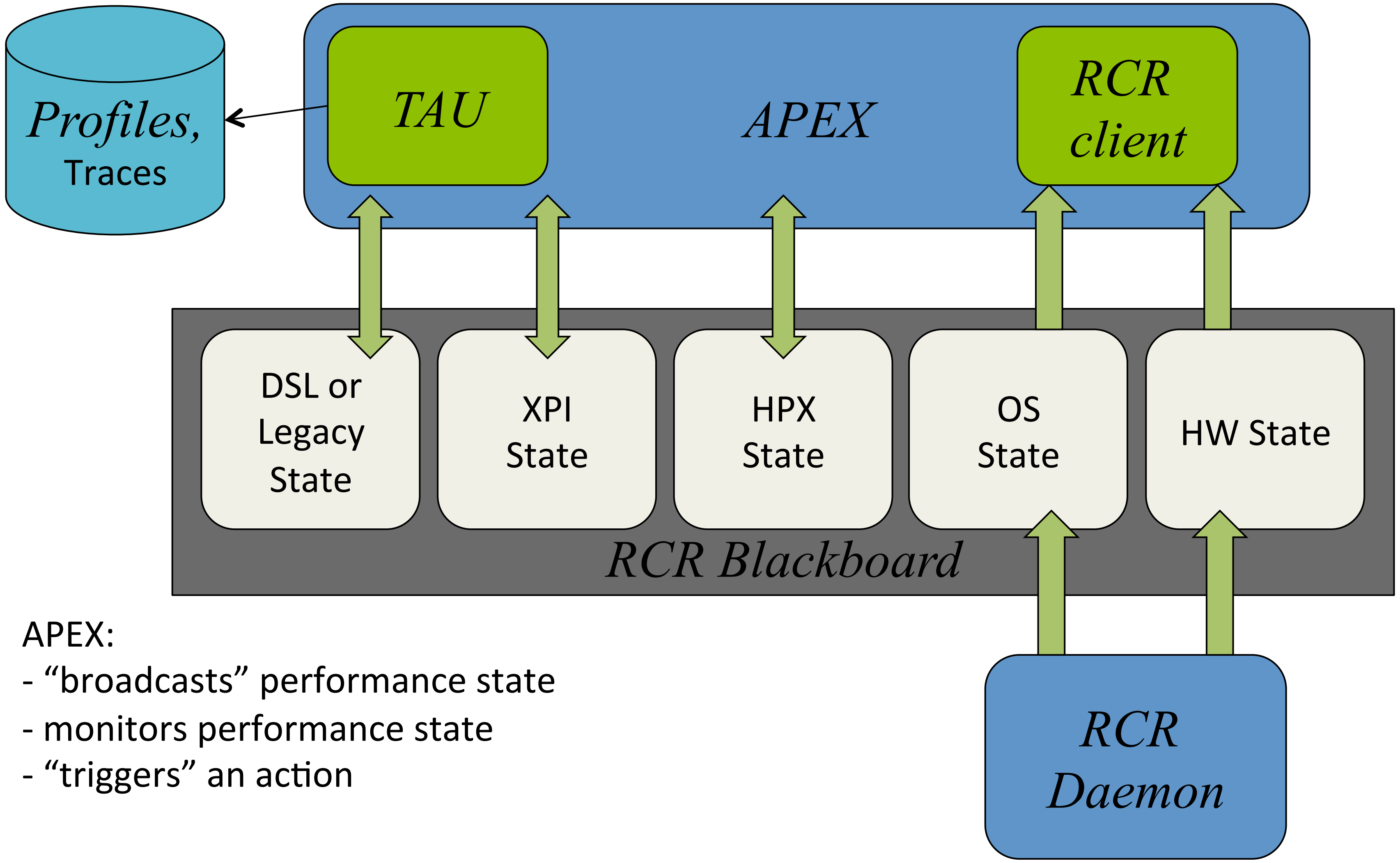
Bottom-Up Development Approach

- Performance introspection across layers to enable dynamic, adaptive operation and decision control
- Extends previous work on building decision support instrumentation (RCRToolkit) for introspective adaptive scheduling
- Resource Centric Reflection
- Daemon monitors shared, non-core resources
- Real-time analysis, raw/processed data published to shared memory region, clients subscribe
- Display/logging tool, adaptive thread scheduler

Top-Down Development Approach

- Couples first person and third person performance views by translating application context down the OpenX stack
- Associates the execution performance state back up
- TAU Performance System used as base for measurement in HPX
- Wrapper interposition libraries for legacy measurement, XPI measurement

APEX: monitoring and writing to RCR



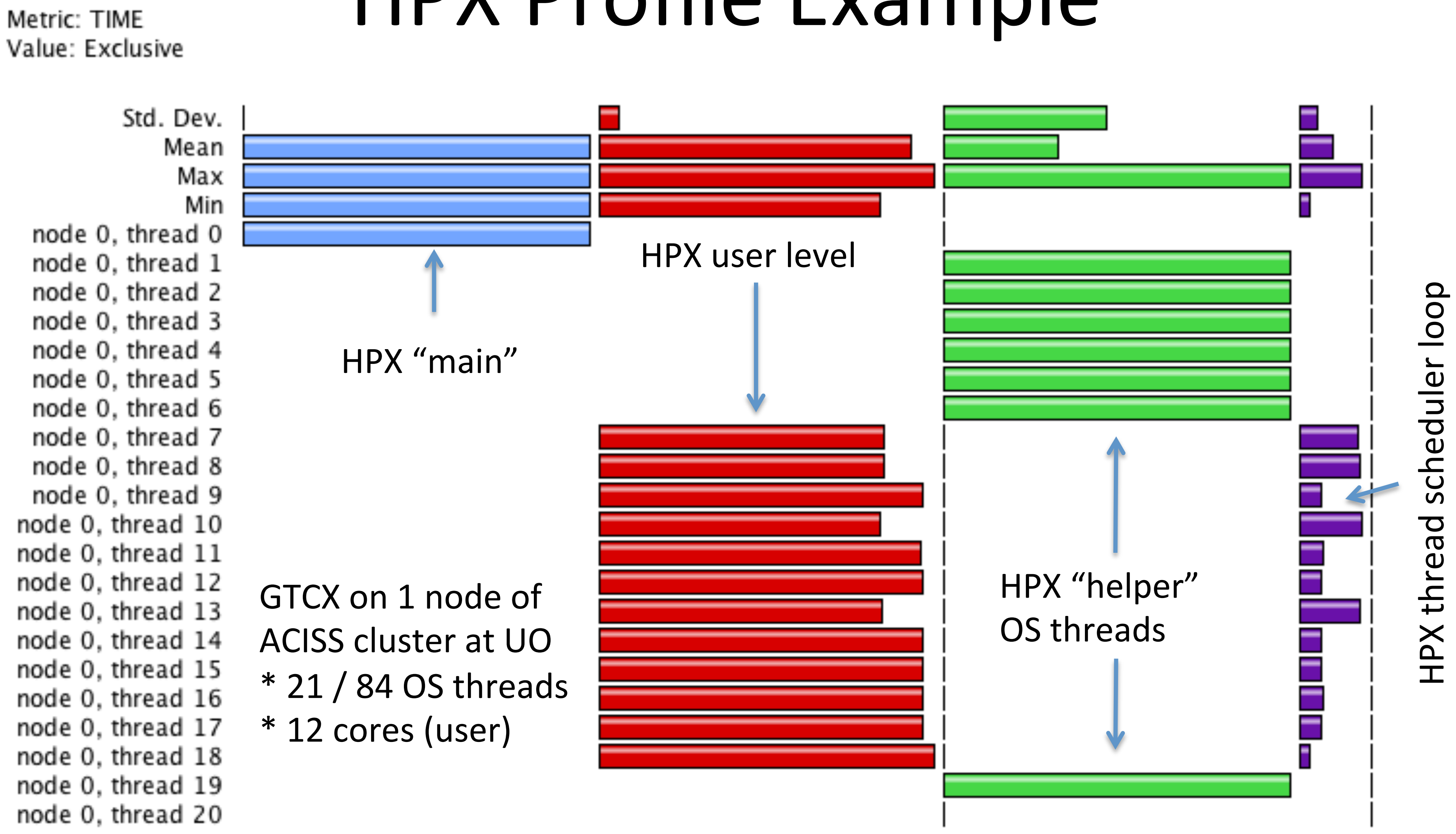
APEX:
- "broadcasts" performance state
- monitors performance state
- "triggers" an action

APEX: Prototype Implementation

- Mapping of HPX performance counters to APEX (TAU) counters
- Intel® ITT Instrumentation of HPX runtime (thread scheduler) mapped to APEX timers
- APEX as RCR client to observe power use
- RCR / HPX thread scheduler integration underway

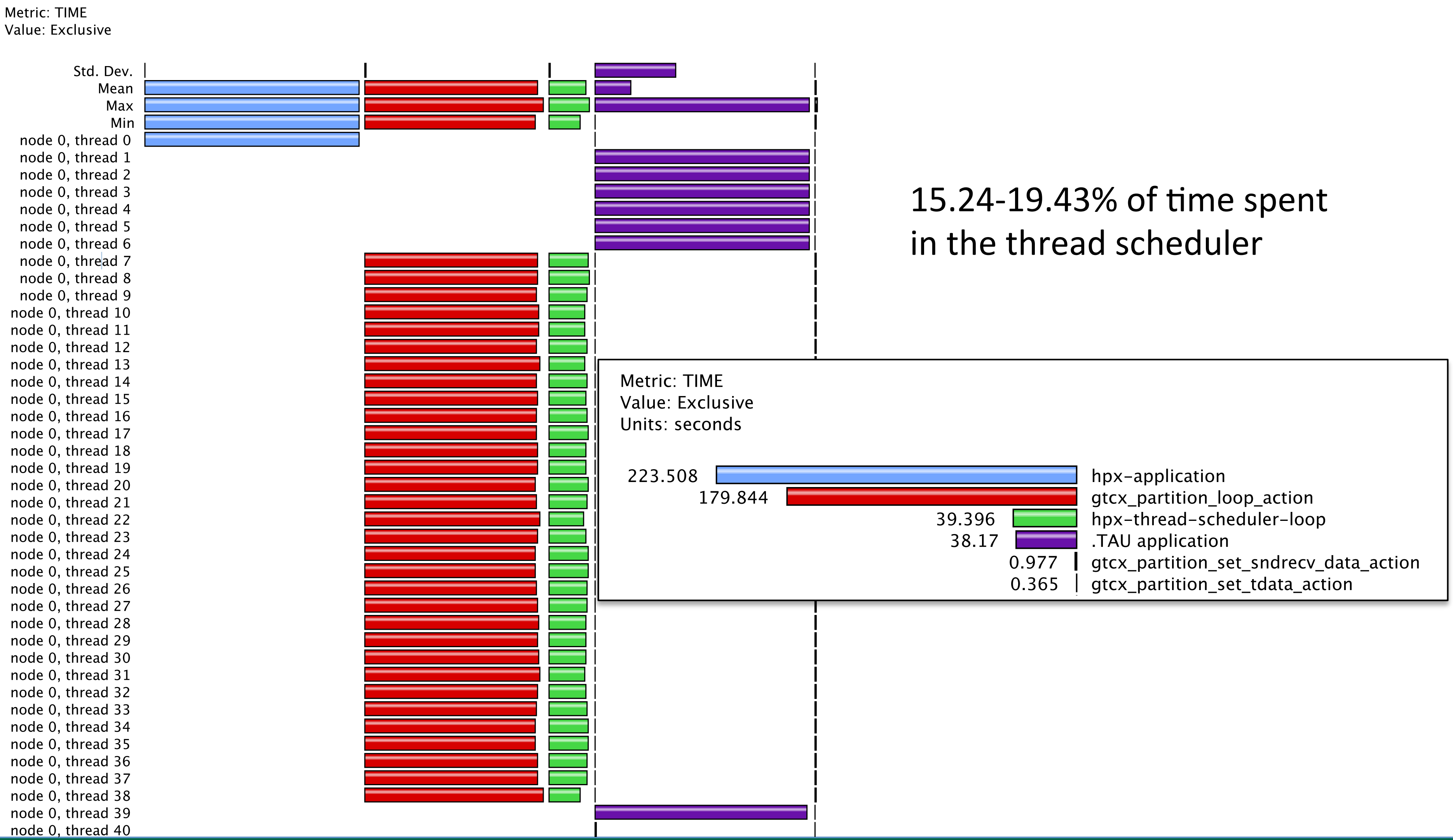
UNIVERSITY OF OREGON

HPX Profile Example



GTCX on 1 node of ACISS cluster at UO
* 21 / 84 OS threads
* 12 cores (user)

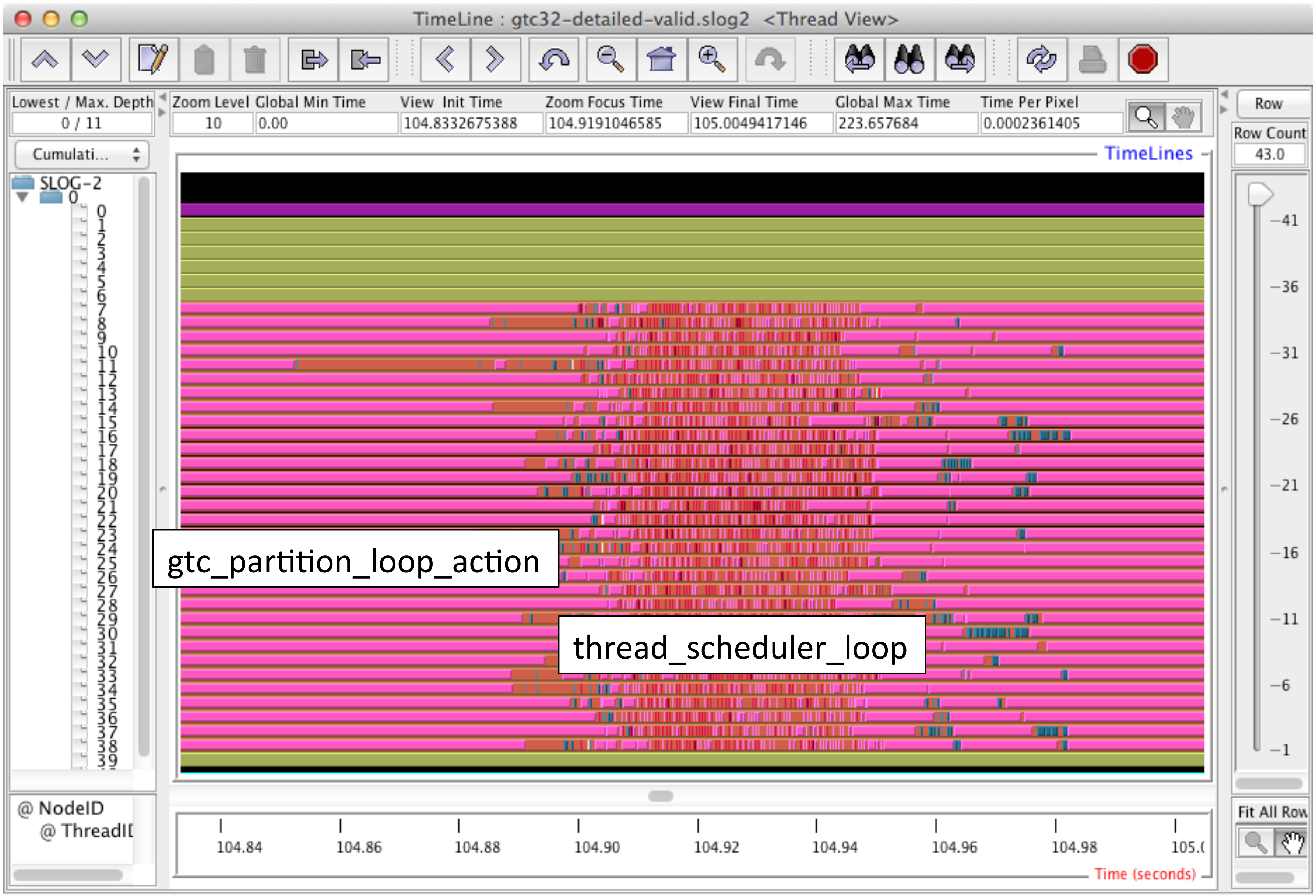
GTC Profile with 32 threads, 1 node



Metric: TIME
Value: Exclusive
Units: seconds

223.508	🟦	hpx-application
179.844	🟥	gtcx_partition_loop_action
39.396	🟩	hpx-thread-scheduler-loop
38.17	🟪	.TAU application
0.977			gtcx_partition_set_sndrecv_data_action
0.365			gtcx_partition_set_tdata_action

15.24-19.43% of time spent in the thread scheduler



gtc_partition_loop_action

thread_scheduler_loop

Future Work

- Distinguish between HPX task execution and preemption/migration
- Performance data not stored in ParalleX model
- Instrumentation of HPX parcel transport, local control objects, global address space
- Expanded information sharing between components
- Runtime analysis and distillation of wide-scale performance data
- Integration with Kitten/LXK OS
- Event based model for triggering runtime corrections

Acknowledgements

- Support for this work was provided through Scientific Discovery through Advanced Computing (SciDAC) program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (and Basic Energy Sciences/Biological and Environmental Research/High Energy Physics/Fusion Energy Sciences/Nuclear Physics) under award numbers DE-SC0008638, DE-SC0008704, DE- FG02-11ER26050 and DE-SC0006925.
- The ACISS cluster is supported by a Major Research Instrumentation grant from the National Science Foundation (NSF), Office of Cyber Infrastructure, grant number OCI-0960354.
- Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. DOE's National Nuclear Security Administration under contract DE-AC04-94AL85000.