SAND2017-13777R

December 2017 ECP ST Project Review

ECP Project WBS 2.3.5.04 (SNL ATDM Software Ecosystem)

PM: Ron Brightwell (Sandia Labs)

12/20/2017

Sandia National Laboratories

EXASCALE COMPUTING PROJECT

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Project Overview

Qthreads

- Ron Brightwell – project manager
- OS/On-Node Runtime (OS/ONR) Team
  - Stephen Olivier – technical lead, runtime systems
  - Kevin Pedretti – technical lead, operating systems
  - Andrew Younge – containers and virtualization
  - Kurt Ferreira, Scott Levy – MPI and noise characterization
  - Ryan Grant – interconnects, network stack, MPI Forum
  - Noah Evans – runtime and operating systems
- ASD Technology Demonstrator Team
  - Michael Tupek
  - Jesse Thomas
  - Patrick Xavier
- Funding: $1500k per year
  - $1275k OS/ONR
  - $225k Tech Demo

Impact Goal	Impact Metric
Maximize the impact of Kokkos and AMT programming models developed under SNL ATDM project	The number of SNL legacy applications using Kokkos and AMT models
Develop OS/R resource management policies and mechanisms appropriate for SNL ATDM applications	The number of SNL ATDM and ECP applications and vendor OS/R environments using technologies developed in this project

Project Plan

- On-node runtime ATDM effort started in FY16, combined OS/R effort in FY17
- Four Main Thrust areas
  - Containers and virtualization technology
  - Characterizing applications’ MPI usage and sensitivity to system noise
  - Lightweight operating systems
  - Runtime systems for on-node multithreading
- Technology demonstrator
  - Pave path to integrate ATDM-developed technologies into the wider ASC integrated code suite, principally the Sierra engineering analysis applications
  - Produce demonstration applications to drive development of Asynchronous Many-Task Scheduling toolset
  - Enable leveraging of Sierra-developed technologies to support ATDM application milestones (for example stk::simd and stk::search)

Significance for ECP and ASC

- Provide system software support for the ATDM applications and the libraries (*e.g.,* Kokkos and Darma) on which they are built
- Demonstrate use of technologies for exploiting emerging architectures and programming models in contexts relevant to ASC Integrated Codes (*i.e.,* Sandia's non-ATDM mission applications)
- Prepare for efficient use of current and future ATS platforms and the coming exascale systems
- Coordinate with broader ECP efforts to prepare the software ecosystem for exascale

Long-term Deliverables (FY19 and Beyond)

Project	Milestone
TD	Demonstrate integration of DARMA AMT enabled module with MPI based solvers in the tech demonstrator
TD	Enhance and optimize stk::simd toolset for use on ATS-2
OSR	Deploy containers and related technologies on advanced architecture testbed systems
OSR	Evaluate performance of ATDM workloads running on vendor lightweight kernel OS/R stacks
OSR	Test and evaluation of node resource management and runtime on ATS-1/2
OSR	Characterize OS/R resource usage for ATDM workloads and assess impact on performance
OSR	Contribute to OpenMP specification and vendor engagement in support of OpenMP 4.x-5.0 to meet the needs of Kokkos and ATDM apps
OSR	Contribute to MPI specification and vendor engagement in support of standards compliant and performant MPI implementation for Kokkos and ATDM applications
OSR	Contribute to utility thread interface (UTI) specification with Intel, RIKEN, and CEA. Engage with ATS vendors to prototype and deploy on ATS systems

Work for future deliverables beyond FY19:

- System software for application enablement on successive ATS systems as they are deployed
- Early evaluation advanced hardware developed in Path Forward and exascale testbeds
  - Interconnect technology
  - Novel developments in node architectures
- Deeper interactions with Integrated Codes for use of ATDM technologies in mission apps

Delivery of Capabilities

- Strong track record
  - Production lightweight OS deployment on ASC platforms, most recently Red Storm
  - Qthreads runtime used as tasking layer in Cray's Chapel runtime
  - Portals network semantics adopted by multiple hardware vendors
- Influence in development of standards like MPI and OpenMP
- Incorporating container technologies into ATDM DevOps workflows
- Qthreads runtime, Kitten OS available open source on github
  - Qthreads uses BSD license, Kitten uses GPL
  - Possible inclusion in OpenHPC

Milestone Progress: Summary

Project	Completed FY17 Milestone
TD	Evaluate the performance of technology demonstrator that exercise ATDM NGP components and exhibit a range of load balancing and data movement scenarios that are representative of Sandia engineering codes
OSR	Requirements Gathering for OS Services
OSR	Characterize OS jitter signatures critical to performance at exascale
OSR	Refactored and optimized Qthreads/Kokkos tasking implementation for manycore
OSR	OS support for on-node resource management and containerization
OSR	Prototype of on-node system software resource management
OSR	Develop SNL-OS support for Trinity ATS-1 platform

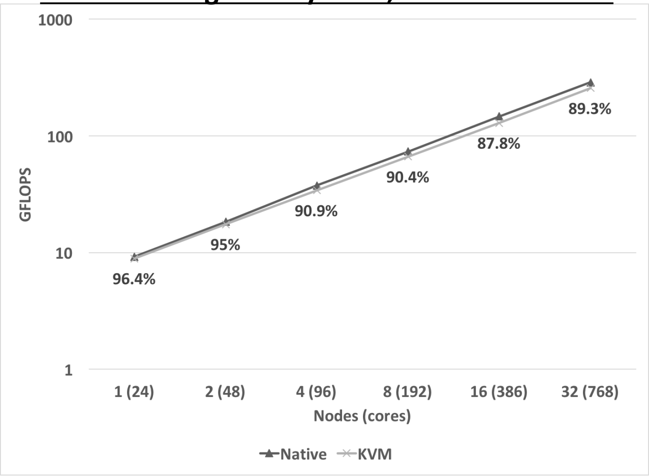
Goal	Metric
Maximize the impact of Kokkos and AMT programming models developed under SNL ATDM project	The number of SNL legacy applications using Kokkos and AMT models
Develop OS/R resource management policies and mechanisms appropriate for SNL ATDM applications	The number of SNL ATDM and ECP applications and vendor OS/R environments using technologies developed in this project

Highlights: Enabling Diverse Software Stacks on Supercomputers using High Performance Virtual Clusters

- Problem
  - HPC, Large-Scale Data Analytics, and Cloud have significantly different OS/R requirements
  - Containers cover many use cases, but not ones where different OS kernels are required
- Approach
  - Add hypervisor capability to supercomputer compute node OS
  - Build virtual clusters using a collection of virtual machines

Impact: First demonstration of virtual clusters on Cray systems. Cray has reproduced results in house and are working with us on tech transfer.

HPCG Running on Cray XC30, Native vs. Virtual



Spark-PERF Running on Cray XC30 in Virtual Cluster

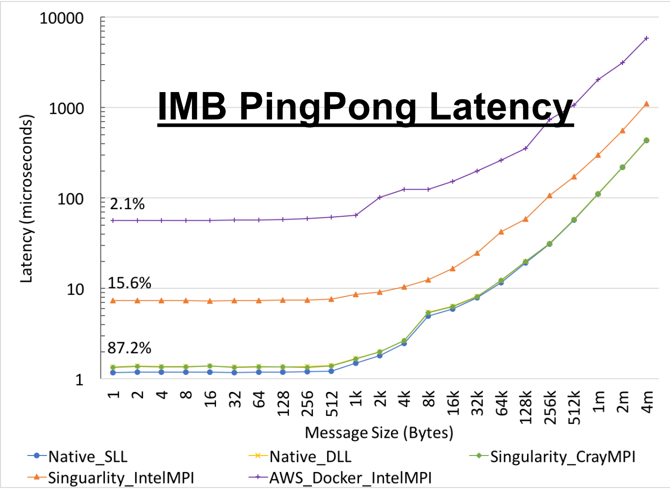
Scale	Throughput	Aggr-by-key	Aggr-by-key-int	Aggr-by-key-naive	Sort-by-key	Sort-by-key-int	Count	Count-filter
0.001	2.6585	0.106	0.1085	0.199	0.114	0.1125	0.034	0.0575
0.01	2.6285	0.219	0.1905	0.4135	0.3065	0.3765	0.0395	0.0935
0.1	2.683	0.474	0.437	0.9605	0.839	0.7075	0.056	0.1495
1	2.6975	2.24	1.886	5.19	2.976	1.797	0.162	0.2665
10	2.642	15.429	47.629	32.9335	5.378	3.9455	1.1095	1.1935

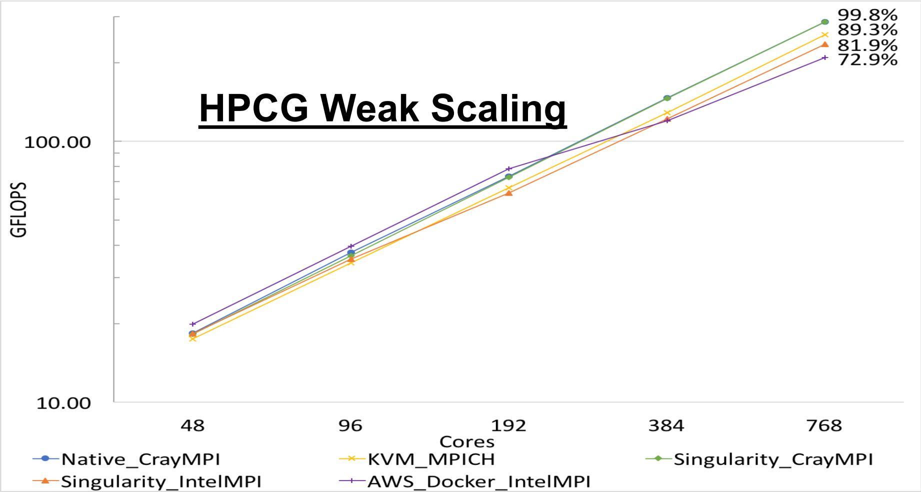
Cluster’17 Paper: Enabling Diverse Software Stacks on Supercomputers using High Performance Virtual Clusters

A Tale of Two Systems: Using Containerization to Deploy HPC Applications on Supercomputers and Clouds

- Problem
  - Supercomputers are scarce resources, busy and expensive
- Approach
  - Leverage Singularity containers to enable initial application dev/test in cloud, seamlessly move to supercomputer when needed for higher performance
  - Compared performance of same containers running on Cray and Amazon EC2 (for similar hardware)

Impact:
Container portability from laptops, to clouds, to supercomputers with native performance





CloudCom'17 Paper: "A Tale of Two Systems: Using Containerization to Deploy HPC Applications on Supercomputers and Clouds"

Highlights: Orchestrating Specialized OS/R’s in Multi-Enclave Environments

- Problem
  - Multi-kernel OS/R’s for exascale: Intel mOS, RIKEN McKernel, DOE Hobbes
  - No common infrastructure for deploying, managing, and composing these OS/R’s
- Approach
  - Develop generalized OS/R agnostic interfaces for managing and configuring multiple OS/R enclaves running on the same compute node

Impact:
Effective coordination of on-node resources between multiple OS/R environments with excellent performance isolation capabilities

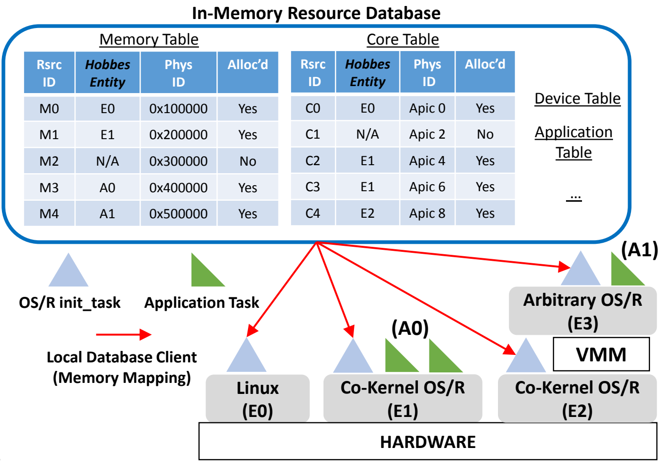
Leviathan Node Manager

Entity: Any piece of software that can manage a raw piece of hardware

Resource: Any piece of hardware that is functionally isolatable

Example:

4 enclaves running different OS/R’s coordinate through Leviathan to allocate and map physical resources (e.g., cores, memory, gpu devices)



In-Memory Resource Database

Memory Table

Rsrc ID	*Hobbes Entity*	Phys ID	Alloc’d
M0	E0	0x100000	Yes
M1	E1	0x200000	Yes
M2	N/A	0x300000	No
M3	A0	0x400000	Yes
M4	A1	0x500000	Yes

Core Table

Rsrc ID	*Hobbes Entity*	Phys ID	Alloc’d
C0	E0	Apic 0	Yes
C1	N/A	Apic 2	No
C2	E1	Apic 4	Yes
C3	E1	Apic 6	Yes
C4	E2	Apic 8	Yes

Device Table

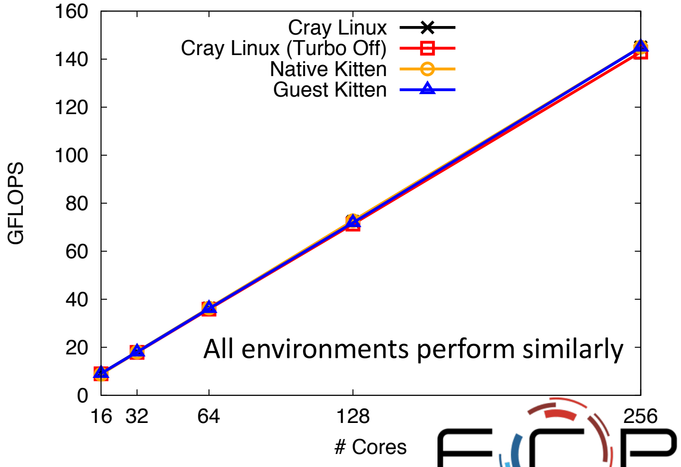
Application Table

...

HPCG Running on Leviathan

32 Cray XC30 Nodes

3 Enclaves: Native, LWK (Kitten), VM (Guest)



All environments perform similarly

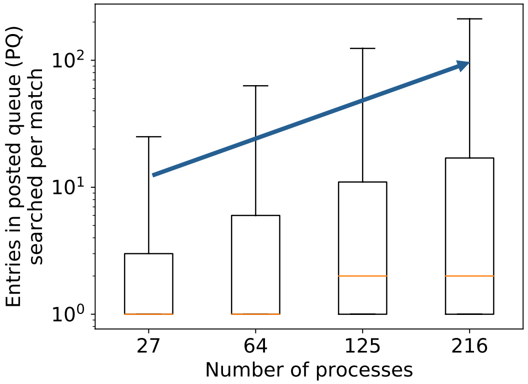
SC’17 Poster; conference paper in submission

Highlights: MPI Usage Characterization via Simulation

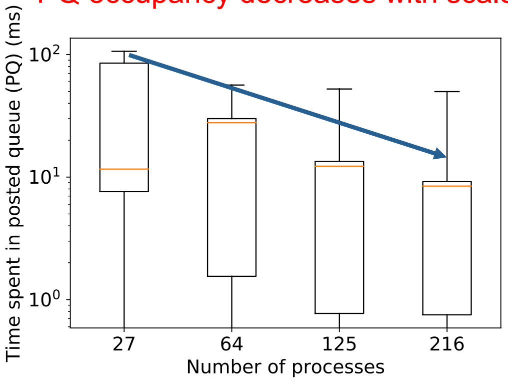
- Problem
  - Extreme-scale communication performance limited by speed of MPI match time, but system behavior not well understood
- Approach
  - Extended **LogGOPSim** simulator to track MPI resource usage without perturbing application
  - Track sizes and occupancy times of posted receive and unexpected message queues (results shown for MILC)

Impact:
Understanding of MPI matching behavior to guide hardware implementation choices.

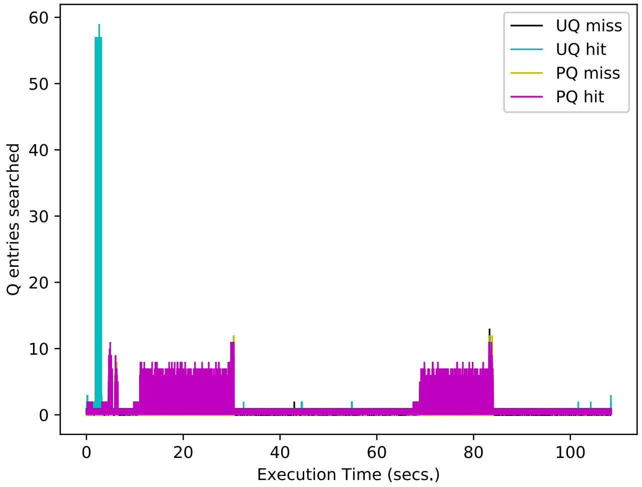
PQ search depth increases with scale



PQ occupancy decreases with scale



Search depths vary throughout execution

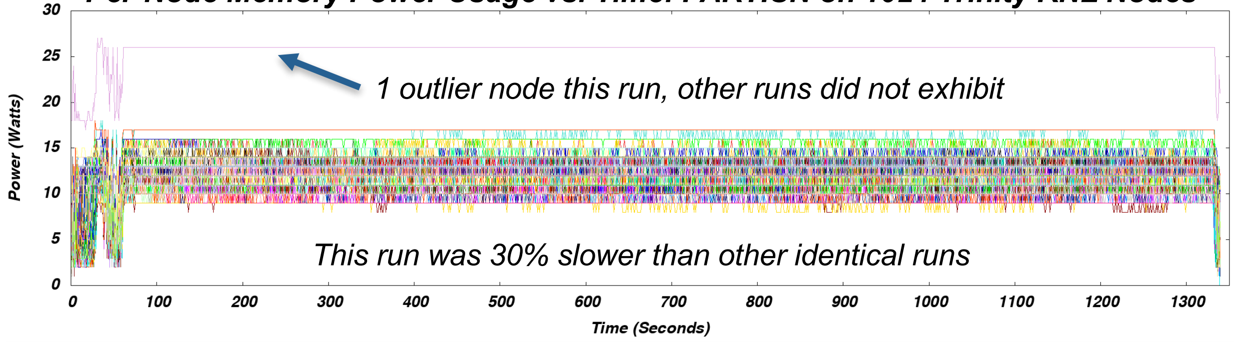


EuroMPI/USA ’17 Paper: “Characterizing MPI Matching via Trace-based Simulation

Highlights: Scalable Monitoring to Diagnose Runtime Variability

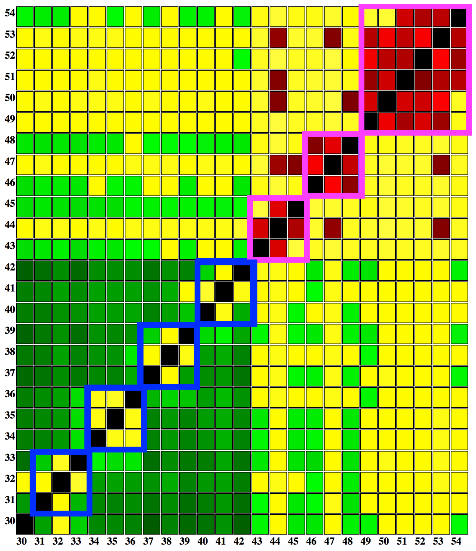
- Problem
  - Performance variability is significant on modern systems and getting worse
  - Common question from users: “Why does my application performance vary so much?”
- Approach
  - Leverage scalable system monitoring infrastructure (LDMS)
  - Analyze and identify actionable metrics associated with application performance degradation
- Key results
  - Shared network contention and I/O are key sources of variability and can be measured
  - Power usage differences across nodes can be useful for identifying anomalous system issues

Per-Node Memory Power Usage vs. Time: PARTISN on 1024 Trinity KNL Nodes



1 outlier node this run, other runs did not exhibit

This run was 30% slower than other identical runs



Analysis of Cray Aries Network Counters Identify Network Links with High Congestion (Red)

Impact: Infrastructure for collecting + analyzing large volumes of actionable system monitoring data

CUG’17 Paper: “Runtime Collection and Analysis of System Metrics for Production Monitoring of Trinity Phase II”

Integration and Readiness

- Support other ATDM projects, e.g., Kokkos and Darma in SNL software stack
  - Qthreads runtime also available through Spack
- Focus on performance and scalability
  - Most performance testing done using Sandia’s ASC CSSE testbeds
  - Larger scale testing using CTS and ATS systems

Related Projects

- Sandia ATDM Programming Models (Kokkos and Darma)
  - We provide enabling system software support and a bridge for their technologies to Sandia’s ASC Integrated Codes
- SNL ASC Integrated Codes (Non-ECP ASC Software)
  - These are the NNSA mission applications, current and future consumers of ATDM technology
- ARGO project
  - The other ECP OS project led by Argonne
- Non-DOE Software
  - mOS, Intel’s lightweight OS that we are evaluating
  - McKernel, RIKEN
  - Cray Chapel language for high productivity HPC and Sandia Multithreaded Graph Library (MTGL) use the Qthreads runtime system

Next Steps (FY18)

Project	Milestone Title
TD	Demonstrate efficient combination of Kokkos on-node parallelism with AMT in the contact/multiscale tech demonstrator
TD	Develop stk::simd into toolset usable by ATDM applications, support and optimize the stk::simd toolset for ATS-1
OSR	Integrate Kokkos-enabled contact into multiscale AMT technology demonstrator
OSR	Coordinate with ATDM DevOps to make plan for utilizing containers for build and testing of Trilinos
OSR	Prototype usage of containers and related technologies to support ATDM developer workflows
OSR	Evaluate lightweight kernel operating systems on advanced architecture testbed, with vendor and ACES engagement to investigate performance and tech transfer of Sandia lightweight kernel capabilities
OSR	Resource manager applied to DARMA+Kokkos use case scenarios
OSR	Runtime system pathfinding and development targeting ATS-1/2
OSR	Characterization of MPI resource usage for ATDM workloads and its impacts on performance
OSR	Prototype of message-based open source simulation framework capable of quantifying MPI resource usage for MPI-based ATDM workloads
OSR	Contribute to OpenMP, MPI, and UTI (Utility Thread Interface) specifications and vendor engagement in support of standards compliant and scalable implementations to meet the needs of Kokkos and ATDM apps

Risks and Issues

Risk/Issue	Mitigation
Shifting architectural landscape for node designs and interconnect technology	Track Path Forward efforts and leverage the advanced architecture testbeds
Resistance to new programming models among IC developers	Engage early and often with IC code teams
Overhead of reporting though multiple channels diverts effort from the technical work	Management commitment to streamline reporting (Can ECP leaders help reduce burden?)

December 2017 ECP ST Project Review

ECP Project WBS 2.3.5.04 (SNL ATDM Software Ecosystem)

PM: Ron Brightwell (Sandia Labs)

12/20/2017

Sandia National Laboratories

EXASCALE COMPUTING PROJECT

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.