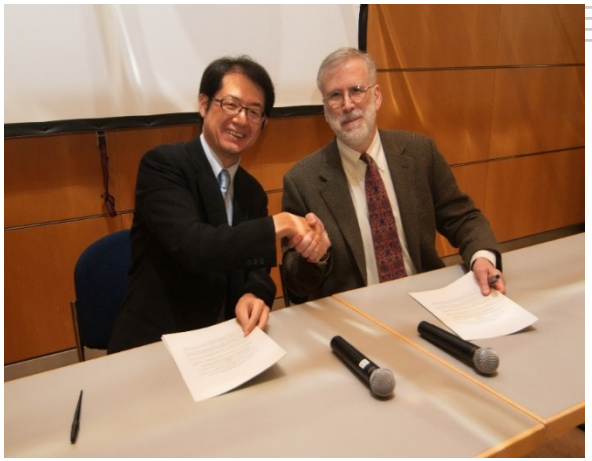
US-Japanese Collaboration

Pete Beckman, Argonne National Laboratory
Yutaka Ishikawa (Mitsuhisa Sato), RIKEN AICS & University of Tokyo
Pavan Balaji, ANL
Jeffrey Vetter, ORNL & Georgia Institute of Technology
Martin Schulz, LLNL

International Collaboration between DOE and MEXT

PROJECT ARRANGEMENT
UNDER THE IMPLEMENTING ARRANGEMENT
BETWEEN
THE MINISTRY OF EDUCATION, CULTURE, SPORTS, SCIENCE AND TECHNOLOGY OF JAPAN
AND
THE DEPARTMENT OF ENERGY OF THE UNITED STATES OF AMERICA
CONCERNING COOPERATION IN RESEARCH AND DEVELOPMENT IN ENERGY AND RELATED FIELDS

CONCERNING COMPUTER SCIENCE AND SOFTWARE RELATED TO CURRENT AND FUTURE HIGH PERFORMANCE COMPUTING FOR OPEN SCIENTIFIC RESEARCH



Yoshio Kawaguchi (MEXT, Japan) and William Harrod (DOE, USA)

Purpose: Work together where it is mutually beneficial to expand the HPC ecosystem and improve system capability

- Each country will develop their own path for next generation platforms
- Countries will collaborate where it is mutually beneficial

- Joint Activities
  - Pre-standardization interface coordination
  - Collection and publication of open data
  - Collaborative development of open source software
  - Evaluation and analysis of benchmarks and architectures
  - Standardization of mature technologies

Technical Areas of Cooperation

- Kernel System Programming Interface
- Low-level Communication Layer
- Task and Thread Management to Support Massive Concurrency
- Power Management and Optimization
- Data Staging and Input/Output (I/O) Bottlenecks
- File System and I/O Management
- Improving System and Application Resilience to Chip Failures and other Faults
- Mini-Applications for Exascale Component-Based Performance Modelling

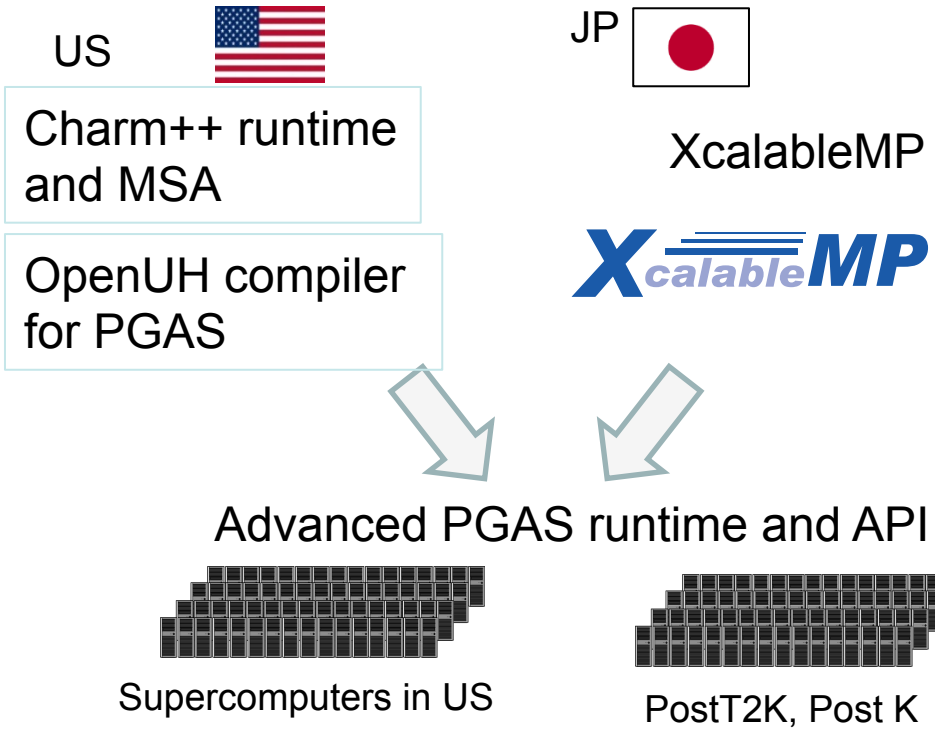
List of Presentations at the first coordination committee

1. Operating System and Runtime
   - Coordinators: Pete Beckman (ANL) and Yutaka Ishikawa (RIKEN)
   - Leaders: Kamil Iskra (ANL) and Balazs Gerofi (RIKEN)
2. Power Monitoring, Analysis and Management
   - Coordinators: Martin Schulz (LLNL) and Hiroshi Nakamura (U. Tokyo)
   - Leaders: Martin Schulz (LLNL), Barry Rountree (LLNL), Masaaki Kondo (U. tokyo), and Satoshi Matsuoka (TITECH)
3. Advanced PGAS runtime and API
   - Coordinators: Peter Beckman (ANL) and Mitsuhisa Sato (RIKEN)
   - Leaders: Laxmikant Kale (UIUC), Barbara Chapman (U. Huston)
4. Storage and I/O
   - Coordinators: Rob Ross (ANL) and Osamu Tatebe (U. Tsukuba)
   - Leaders: Rob Ross (ANL) and Osamu Tatebe (U. Tsukuba)
5. I/O Benchmarks and netCDF implementations for Scientific Big Data
   - Coordinators: Choudary (North Western U.) and Yutaka Ishikawa (RIKEN)
   - Leaders: Choudary (North Western U.) and Yutaka Ishikawa (RIKEN)
6. Enhancements for Data Movement in Massively Multithreaded Environments
   - Coordinators: Peter Beckman (ANL) and Satoshi Matsuoka (TITECH)
   - Leaders: Pavan Balaji (ANL) and Satoshi Matsuoka (TITECH)
7. Performance Profiling Tools, Modeling and Database
   - Coordinators: Jeffery Vetter (ORNL) and Satoshi Matsuoka (TITECH)
   - Leaders: Jefery Vertter (ORNL), Martin Shultz (LLNL), Satoshi Matsuoka (TITECH), and Naoya Maruyama (RIKEN)
8. Mini- /Proxy-Apps for Exascale Codesign
   - Coordinators: Jefery Vetter (ORNL) and Satoshi Matsuoka (TITECH)
   - Leaders: <TBA> and Naoya Maruyama (RIKEN)
9. Extreme-Scale Resilience for Billion-Way Parallelism
   - Coordinators: Martin Schulz (LLNL) and Satoshi Matsuoka (TITECH)
   - Leaders:
10. Scalability and performance enhancements to communication library
    - Coordinators: Pete Beckman (ANL) and Yutaka Ishikawa (RIKEN)
    - Leaders: Pavan Balaji (ANL) and Masamichi Takagi (RIKEN)
11. Communication Enhancements for Irregular/Dynamic Environments
    - Coordinators: Pete Beckman (ANL) and Yutaka Ishikawa, RIKEN
    - Leaders: Pavan Balaji (ANL) and Atsushi Hori (RIKEN)

Meeting in Dec/2014 at RIKEN AICS

Advanced PGAS runtime and API, programming models

- Coordinators
  - US: Peter Beckman, Argonne National Lab.
  - JP: Mitsuhisa Sasto, RIKEN AICS
- Leaders
  - US: Laxmikant (Sanjay) Kale, UIUC
  - Barabara Chapman, Univ. of Huston
  - JP: Mitsuhisa Sato, RIKEN AICS
- Description
  - Each side (RIKEN and Univ. Huston) is developing the programming languages and compilers based on “coarray” PGAS model. US partner, UIUC is working on advanced dynamic runtime based on their experience of Charm++.
  - We plan to explore how exascale system software may be exploited to enable the execution of PGAS programs at extreme scales, and consider extensions to this model for facilitating asynchronous movement of tasks and data.
  - As our deliverables, and APIs are derived.
- How to collaborate
  - Twice meetings per year
  - Student / young researchers exchange, sharing codes
  - Funding:
    - US: X-stack(XPRESS)?, ARGO?
    - JP: FLAGSHIP 2020
- Deliverables
  - Advanced runtime for scale PGAS model for exasclae
  - Pre-standardization of Application Programming Interface for PGAS language runtime



Plan & Status

- Technical Research Collaboration
  - Exploit the possibility to use Charm++ runtime as a part of XMP runtime
    - Use the idea of Multi-phase Share Arrays (MSA) in XMP.
    - Integrations of Charm++ object on to XMP C++ (under development).
  - Share experiences of PGAS model (“coarray”) and runtime technologies with UH group.
    - Extension with Dynamic tasking in node
- Proposed agenda for pre-standarization
  - What is a good API, what level API is appropriate, to implement PGAS models
  - How to mix with MPI and PGAS operations in multithreaded execution.
  - Execution models of PGAS within shared memory node. How to support one-sided comm within a node.
  - Memory consistency model and semantics of one-sided comm of PGAS models
- Status
  - In March, JP team visit UH. We agreed with moving to design of dynamic tasking (in node) with PGAS models (CAF and XMP)
    - JP invite Post-doc from UH for discussion
    - Plan to send student from U. Tsukuba
  - In March, JP team visited UIUC for the discussion on these topics:
    - Discussion with future programming model
    - Charm++ on Argobots
    - How to integrate with Charm++ and XMP
  - JP team visited Arogbots team at ANL. We agreed the followings
    - Design and prototype implementation of OpenMP (Omni OpenMP) using Argobots for the hybrid of OpenMP and XcalableMP
    - Extends this work to design XcalableMP 2.0 for dynamic tasking

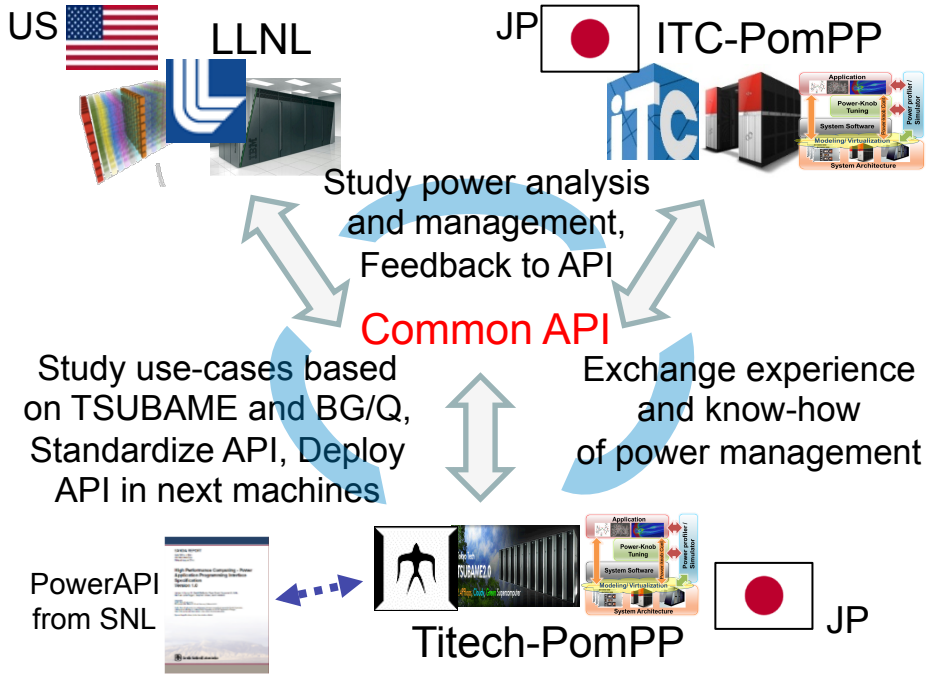
This work will be reported by Pavan

Storage and I/O

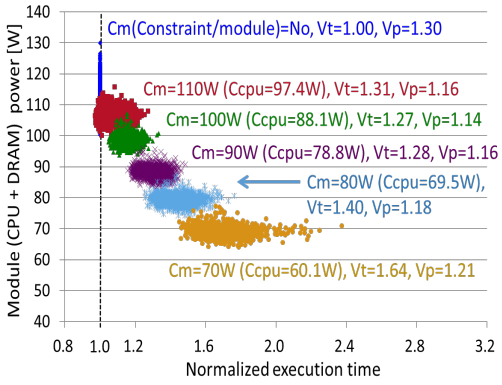
- Collaborators and leaders
  - Rob Ross (ANL) and Osamu Tatebe (U. Tsukuba)
- Motivation
  - Deeper storage hierarchy may be satisfying peak I/O bandwidth requirement at scale, but not for metadata IOPs requirement. Deeper I/O stack may cause high startup cost
- Ongoing projects
  - Triton storage system and CODES simulation projects at ANL, and JST CREST PPFS scale-out storage/ metadata system projects at Univ. of Tsukuba
- Collaboration plan
  - Investigate a common set of requirements and semantics for future data/metadata management design
  - Study the design space including identification of major HW/SW components, representative workloads and relevant methodologies
  - Simulation study of PPFS (Post-Petascsale File System, supported JST CREST) and other scale-out storage/metadata systems

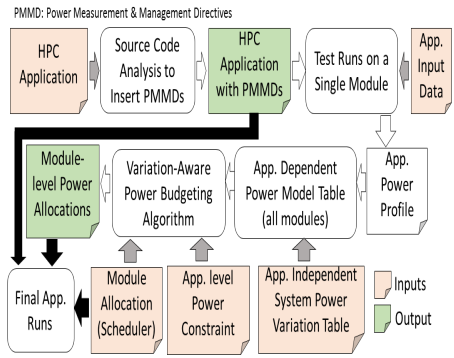
Power Monitoring, Analysis and Management

- Design discussions and feedback on APIs for monitoring, analyzing, and managing power, from the node to the global level
  - Current activity: sharing the designs and requirements for the power and energy monitoring and control interfaces
  - Future plan: trying to unify our designs and converge on common semantics and requirements



- Evaluation of power steering techniques for over-provisioned systems
  - Current activity: investigating a CPU module-wise power-budgeting strategy and publishing a technical paper (*) in cooperation with Kyushu Univ. and Univ. Arizona
  - Future plan: sharing results on the impact of power capping of large scale resources on both side with different applications taken from DOE and MEXT proxy application suites

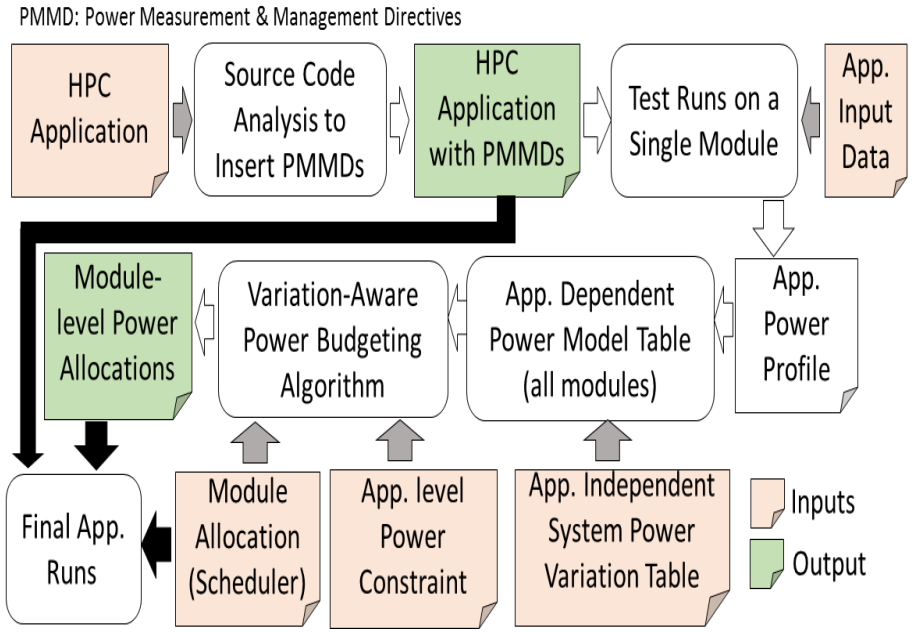


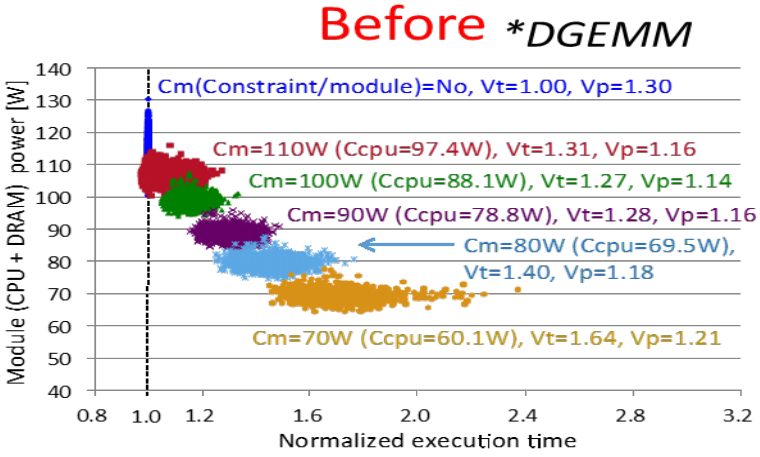


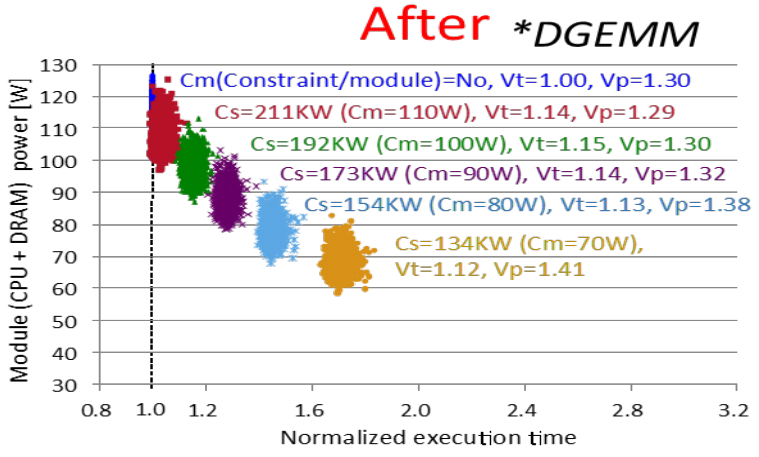
* Y. Inadomi, et al., "Variation-Aware Power Budgeting in Power-Constrained High-Performance Computing", SC'15, 2015

Mitigating Power Variations in Large Scale Clusters

- Power capping turns power into performance variation
  - New system to compensate based on a per application profile
  - Results:
    - Large variation among the computing nodes (left-side figures)
    - Variation aware power budgeting mitigates (right-side figures)



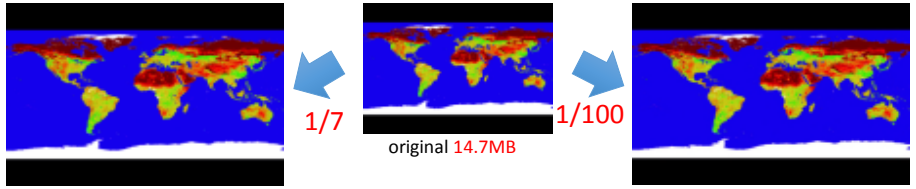




* Y. Inadomi, et al., "Variation-Aware Power Budgeting in Power-Constrained High-Performance Computing", SC'15, 2015

Errors Introduced by Lossy Methods in Scientific Codes

- Errors may be acceptable if we examine processes for developing real scientific applications
  - Scientific modeleing, sensors etc. introduce errors
  - Acceptable Error Tolerances



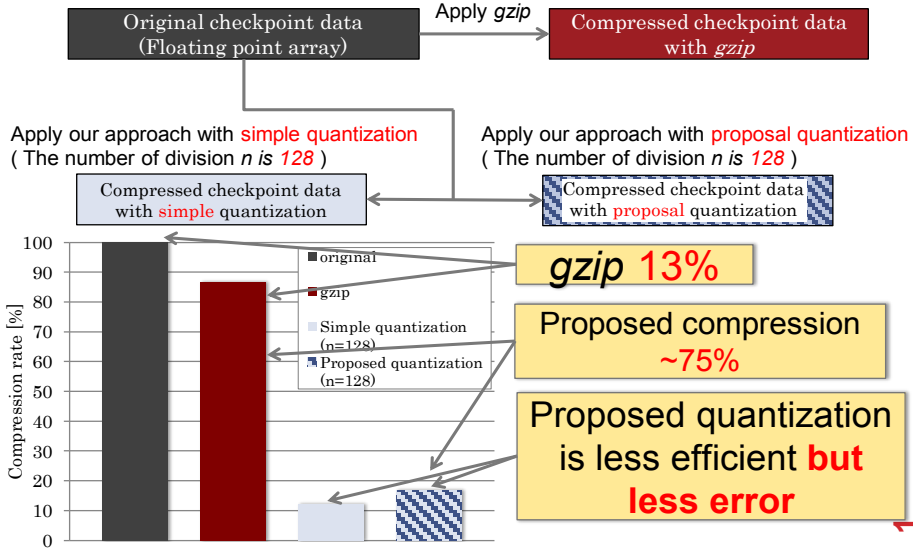
(citation of images : http://svs.gsfc.nasa.gov/vis/a000000/a002400/a002478/)

- **Lossy compression to checkpoint data**
  - **Calculation continues with data including errors => Tolerable?**

LLNL-PRES-670952

LLNL-PRES-670952

Evaluation of Compression Rate



LLNL-TOKYO TECH RESILIENCE

[IPDPS2015 Lossy Checkpoint Compression]

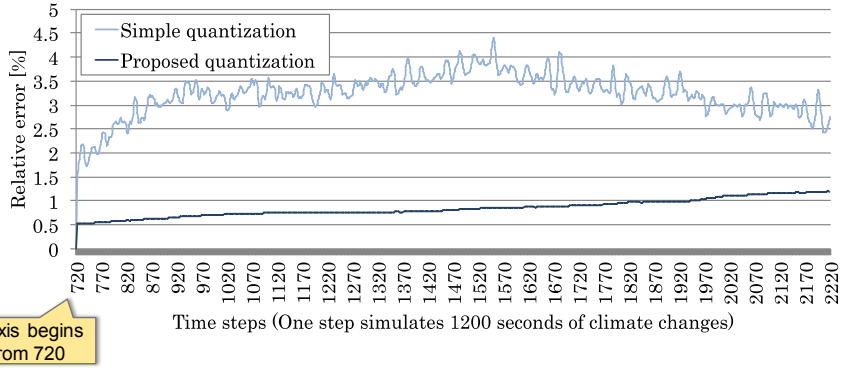
Similar to Soft Error propagation => Further research on error models and their affect (or lack thereof) for Exascale machines

LLNL-PRES-670952

Evaluation of Error Propagation over Time

Lossy checkpoint compression for NICAM (climate code), then recovery scenario

- **Compute with errors => Errors converge, diverge?**
- **Proposed Quantization scheme stabilizes error**



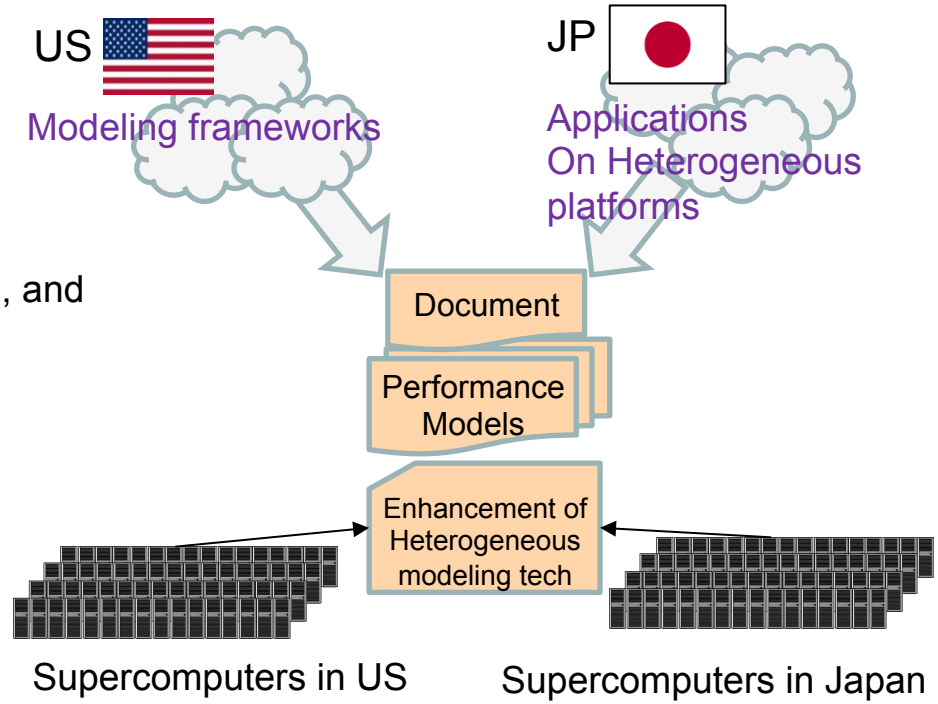
Collaborations with ORNL & RIKEN AICS: Co-design & programming models

- We are now starting discussions on collaborations in the context of US-JP DOE-MEXT collaborations. The two topics are under discussion:
  - Codesign methodology and tools
    - Use tools developed for codesign on both sides to analysis applications proposed in each side.
    - Applications performance studies of both sides.
    - Discussion on the future codesign methodologies
    - (We need to re-arrange some existing topics to avoid conflicts)
  - Advanced Programming models for manycore and accelerators
    - Research on advanced programming model for both of manycore and accelerators
    - To reduce the cost of application development cost by programming models covering both.

- Plan and schedule
  - We will have one-day meeting at RIKEN AICS in Japan on 22th Aug.
  - Research topics of our collaborations will be proposed at the meeting at Cluster 2015, in Sep.

Performance Modeling for Next Generation Applications/Systems

- Coordinators:
  - US: J.S. Vetter, ORNL
  - Japan: Satoshi Matsuoka, TiTech
- Leaders:
  - US: J.S. Vetter, ORNL
  - Japan: Satoshi Matsuoka, TiTech
- Combine expertise on performance modeling, applications, and heterogeneous architectures
- How to collaborate
  - Twice meetings per year
  - Student / young researchers exchange
  - Funding:
    - US: X-Stack, Codesign
    - JP: FLAGSHIP 2020
- Deliverables
  - Deployment of Aspen and Compass performance modeling frameworks
  - Collaborative development of open-source software
    - New application and library models
    - Prototypes of new modeling tools for heterogeneous processing



ANL-Tokyo Tech: Sharing designs and requirements for hybrid communication and threading models

TOKYO TECH Pursuing Excellence

Apps + Runtime

Runtime System

Pros and Cons of MPI+Threads at Large Scale?

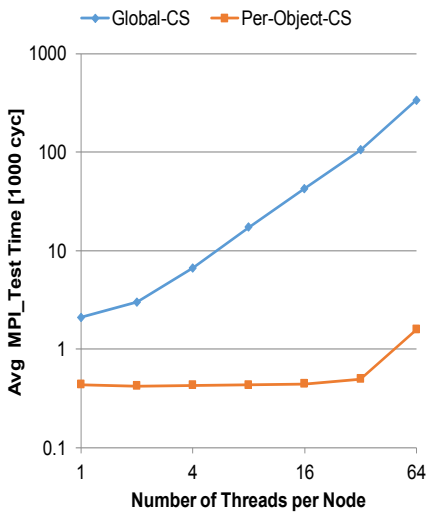
Runtime Contention in Multithreaded MPI due to Thread-Safety

Characterizing Large Scale MPI + Threads[PPMM15, Longer version *Submitted to Journal*]

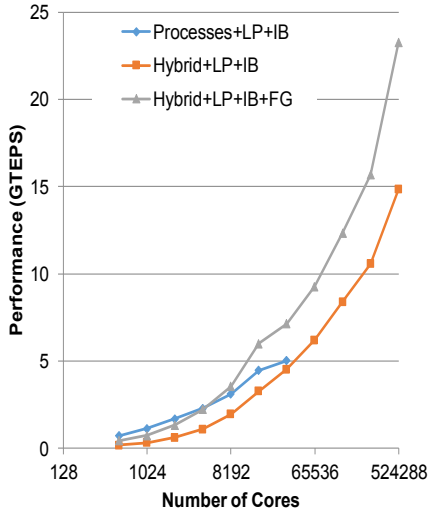
Reducing Contention by Improving Critical Section Arbitration [ACM PPoPP15]

MPI+Threads Alleviates some Drawbacks of MPI-only but Incurs Runtime Contention

Runtime Contention!

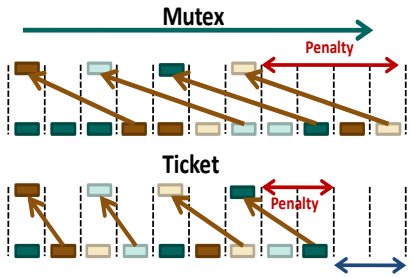


MPI_Test Latency

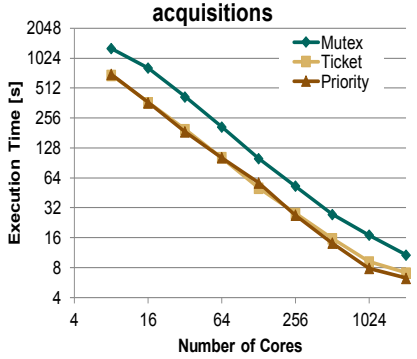


Weak Scaling Performance

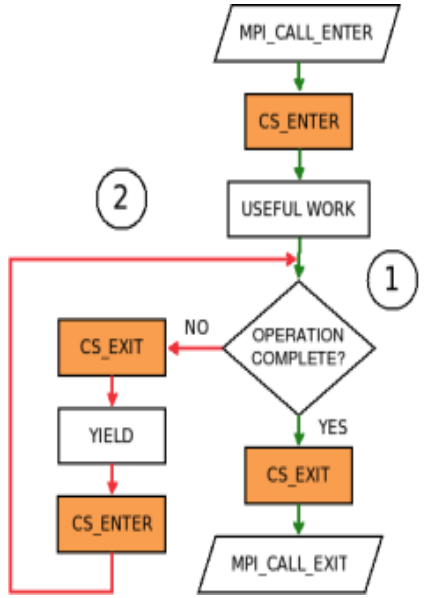
Better Arbitration Policies Implemented



Fairness (FIFO) reduces wasted resource acquisitions



Genome Assembly application results strong scaling



2-Level arbitration policy: (1) has a higher priority

Argonne NATIONAL LABORATORY

Communication and Threading Collaborative Efforts

Pavan Balaji

Computer Scientist and Group Leader

Argonne National Laboratory

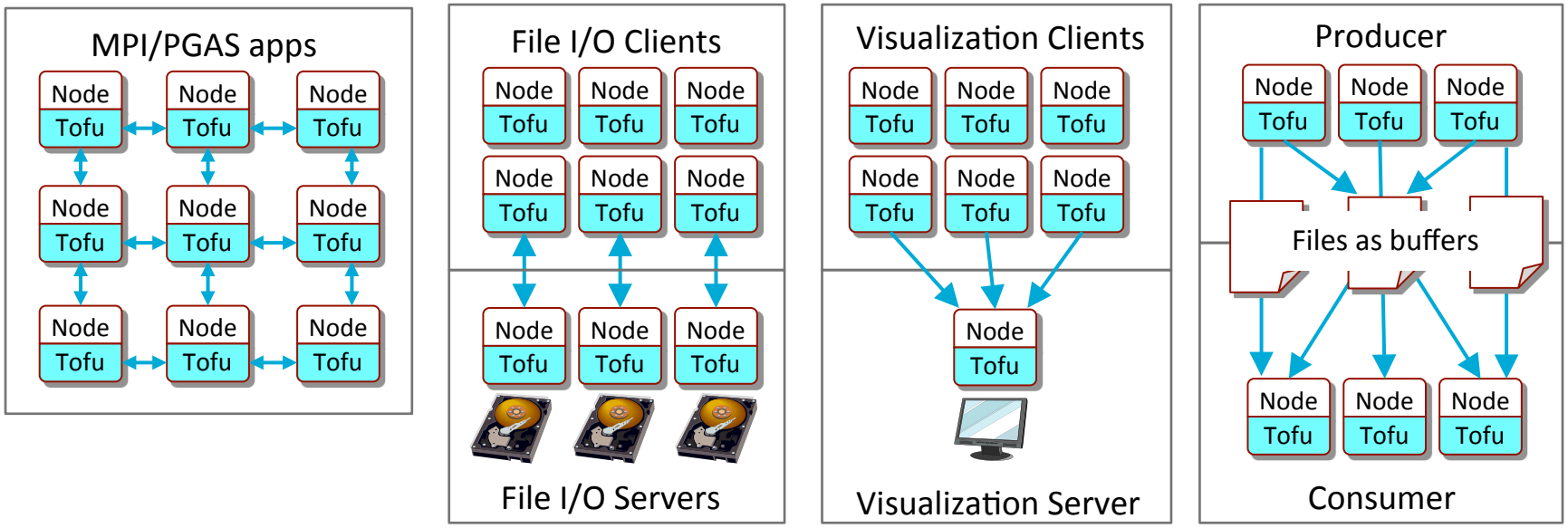
U.S. DEPARTMENT OF ENERGY

Overview of Collaboration Activities

- Communication Libraries
  - Common Low-level network abstractions
    - Pavan Balaji (Argonne) and Masamichi Takagi (RIKEN)
  - Asynchronous communication for irregular MPI applications
    - Pavan Balaji (Argonne) and Yutaka Ishikawa (RIKEN)
  - Dynamic overdecomposition for MPI applications
    - Pavan Balaji (Argonne) and Atsushi Hori (RIKEN)
- Threading Models
  - OpenMP over lightweight threading models
    - Sangmin Seo (Argonne) and Mitsuhisa Sato (RIKEN)
  - Efficient hybrid MPI+threads programming
    - Abdelhalim Amer (Argonne) and Satoshi Matsuoka (Tokyo Tech)

Common Low-level Network Abstractions

Network Abstractions for the Broader Computing



Architecture/programming model is changing

Network architecture

- Higher radix, more shared links

Node architecture

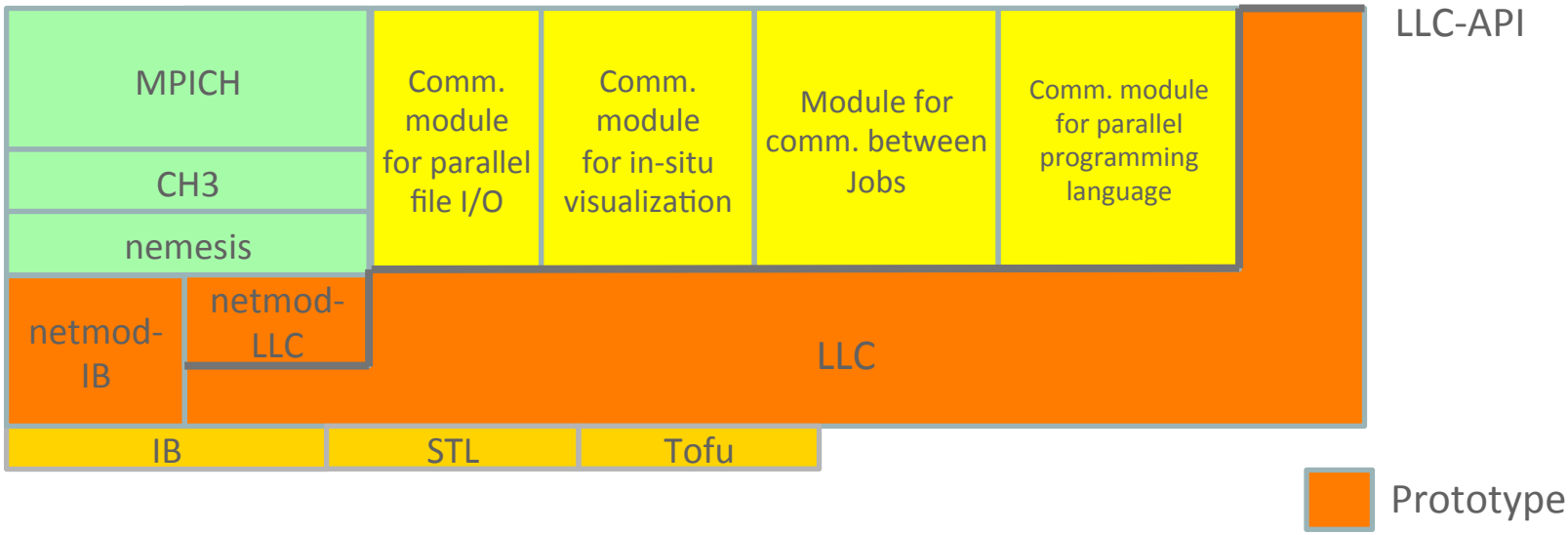
- More cores, less memory per core

Application

- Wide variety of communication models
  - Different programming model (e.g. MPI, PGAS), I/O clients, In-situ visualization, scientific work-flow based applications

Approach

Communication library for next generation computation and communication architecture



Key Capabilities

- Ability to deal with high parallelism on node (multicore/many-core architectures) and on the network (multiple DMA engines and shared communication paths)
- Explicit and introspective resource management (memory is the primary resource today, but network flow-control credits, ability for cache injection, etc., will be considered based on vendor roadmaps)

Argonne NATIONAL LABORATORY

Asynchronous Communication for Irregular MPI Applications

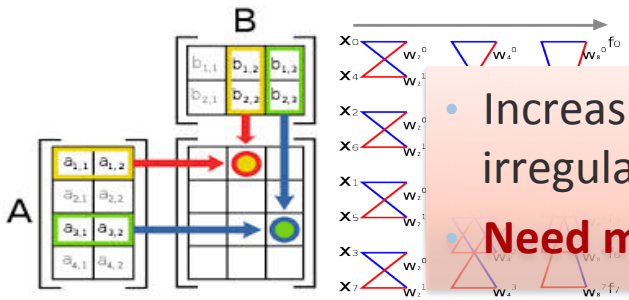
U.S. DEPARTMENT OF ENERGY

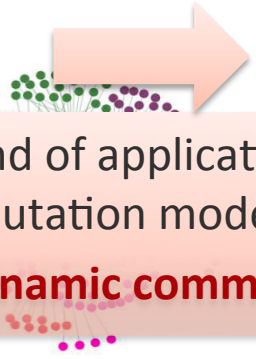
Background

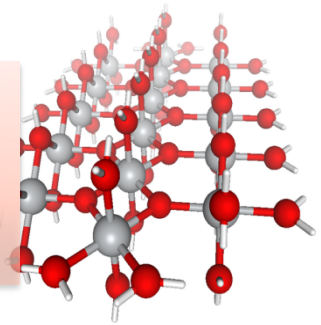
Collaborators:
- US: Pavan Balaji (Argonne)
- JP: Yutaka Ishikawa (RIKEN)

- **Regular computations**
  - Organized around dense vectors or matrices
  - **Regular data movement** pattern, use **MPI SEND/RECV or collectives**
  - More local computation, less data movement
  - Example: stencil computation, matrix multiplication, FFT

- **Irregular computations**
  - Organized around graphs, sparse vectors, more "data driven" in nature
  - Data movement pattern is **irregular and data-dependent**
  - **Growth rate of data movement is much faster than computation**
  - Example: social network analysis, bioinformatics



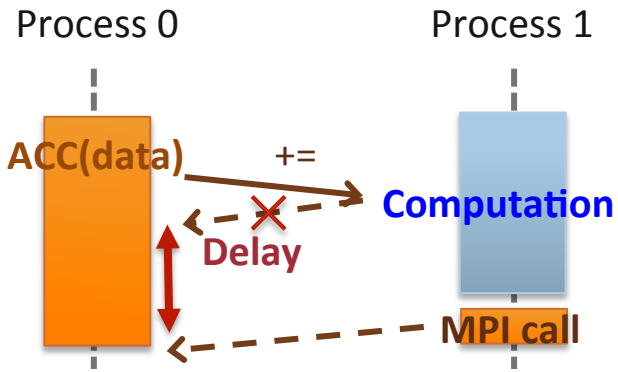




- Increasing trend of applications are moving to irregular computation models
- **Need more dynamic communication model**

Proposal(s)

- **Dynamic/asynchronous communication**
  - Irregular/dynamic communication model
  - "One-sided communication" is **not truly one-sided**

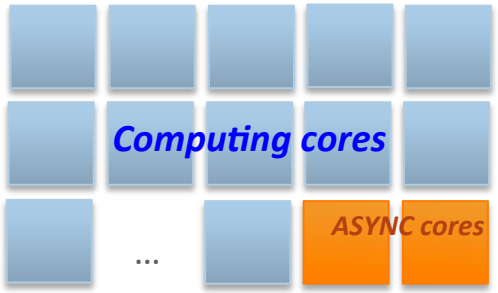


- **Dynamic management of data movement stalls**
  - Shared, but partitioned address space with PVAS
  - Massively parallel and dynamic communication management with "user-level processes (ULPs)"

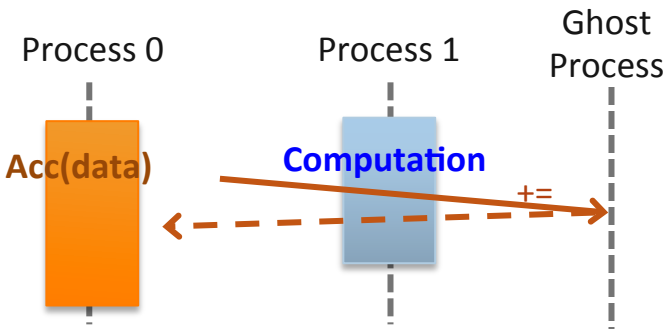
  (new concept being proposed)

Casper: Process-based ASYNC Progress

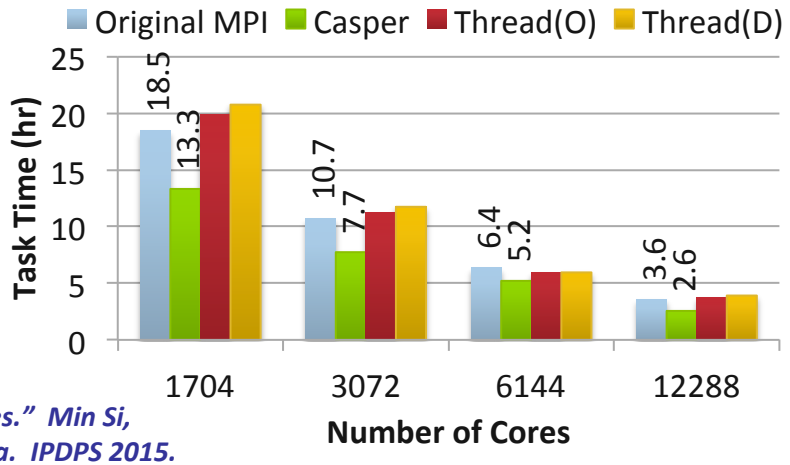
- **Multi- and many-core architectures**
  - "Infinite cores"
  - Not all of the cores are always keeping busy
- **Process-based asynchronous progress**
  - Dedicating **arbitrary number of cores to "ghost processes"**
  - **Ghost process intercepts all RMA operations** to the user processes



Communication with Casper



NWChem CCSD(T) for W21=$(H_2O)_{21}$ with pVDZ



[1] "Casper: An Asynchronous Progress Model for MPI RMA on Many-Core Architectures." Min Si, Antonio Pena, Jeff Hammond, Pavan Balaji, Masamichi Takagi, and Yutaka Ishikawa. IPDPS 2015.

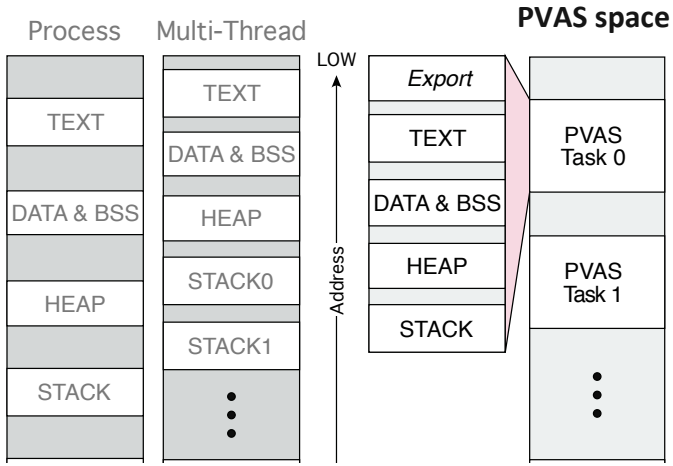
[2] "Scaling NWChem with Efficient and Portable Asynchronous Communication in MPI RMA." Min Si, Antonio J Peña, Jeff Hammond, Pavan Balaji, and Yutaka Ishikawa. CCGrid 2015.

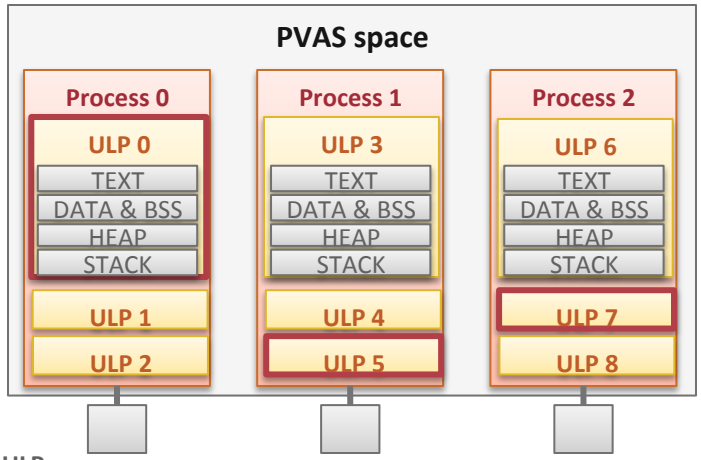
Overdecomposition for MPI Applications

Initial Work: PVAS and ULP

Collaborators:
- US: Pavan Balaji (Argonne)
- JP: Atsushi Hori (RIKEN)

- **Idea: What if processes coexist in the same virtual address space?**
- **Partitioned Virtual Address Space**
  - A process can access data owned by other processes w/o overhead
    - Easy to communicate
    - Easy to share information
- **User-Level Process (ULP) on PVAS**
  - Able to user-level context sw. between PVAS tasks
  - Good affinity with MPI's exec. Model
  - Effectively hide comm. latency
    - S/W impl. of "Hyperthreading"
  - Easy to implement "Ghost Process"
  - Easy to balance loads among "MPI processes" in a node
  - Easy to migrate PVAS task to the other node (in theory)



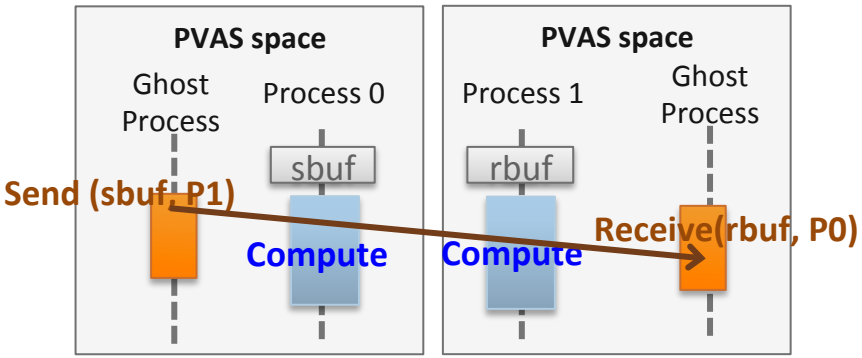


Current status

Personnel Involved

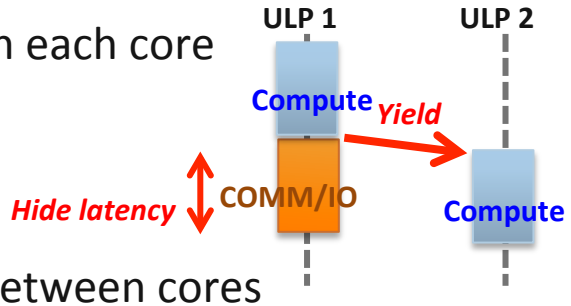
- Min Si
  - Ph.D. student @ U. Tokyo
  - Has been at Argonne for an year and will be there till the end of her Ph.D.
- Atsushi Hori, RIKEN
- Pavan Balaji, Argonne

Two-side Communication with PVAS-Casper



Ongoing MPI optimization

1. Latency hiding on each core



2. Load balancing between cores
   - Migrating MPI process from BUSY core to IDLE core
3. Cooperating with Casper
   - PVAS-based Casper
     - Fully accessible memory space on user processes
     - Support for two-sided/collective modes
   - ULP-based Casper
     - Natively support simultaneous blocking calls

"User-level Process Towards Exascale Systems", Akio Shimada, Atsushi Hori, Yutaka Ishikawa, and Pavan Balaji. Information Processing Society of Japan (IPSJ) workshop, 2014.

Argonne NATIONAL LABORATORY

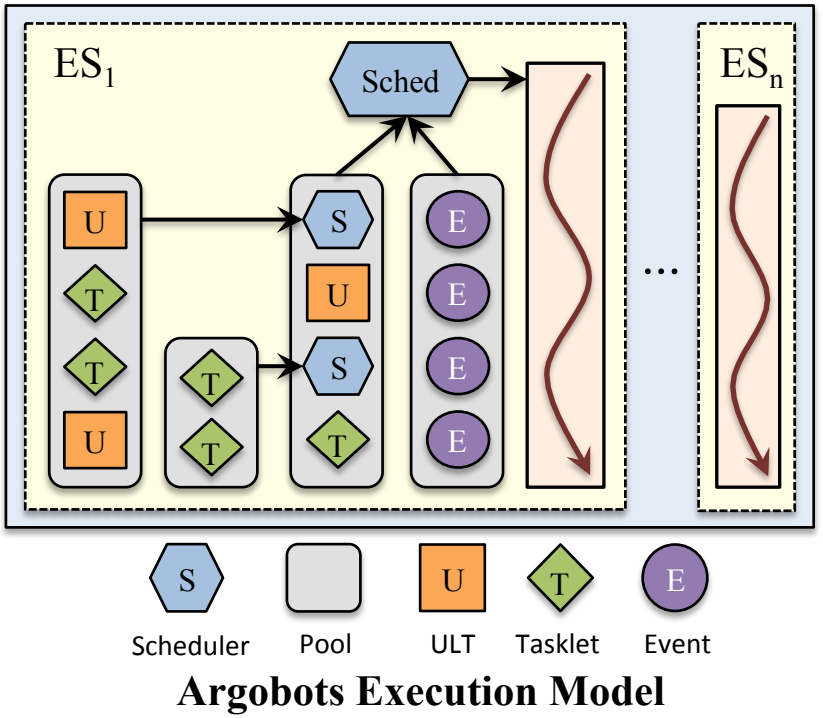
OpenMP over Lightweight Threading Models

U.S. DEPARTMENT OF ENERGY

OpenMP/XMP over Argobots

Collaborators:
- US: Sangmin Seo (Argonne)
- JP: Mitsuhisa Sato (RIKEN)

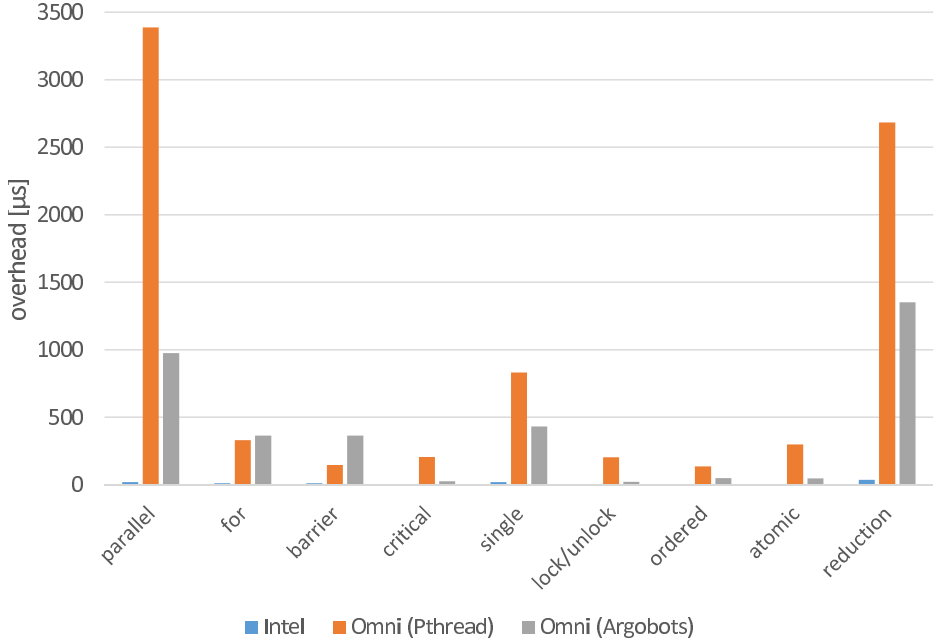
- Objective
  - OpenMP compiler generating work units over Argobots
- Efficient for nested loops and tasks
  - Compiler simply generates ULTs, while the runtime manages them over a fixed set of computational resources
- Argobots
  - Lightweight low-level threading/tasking framework
  - Designed for massive parallelism
    - Execution Streams guarantee progress
    - Work units (ULTs or Tasklets) execute to completion
  - Explicit mapping ULT/Tasklet to ES
    - Scheduling policy decoupled and pluggable

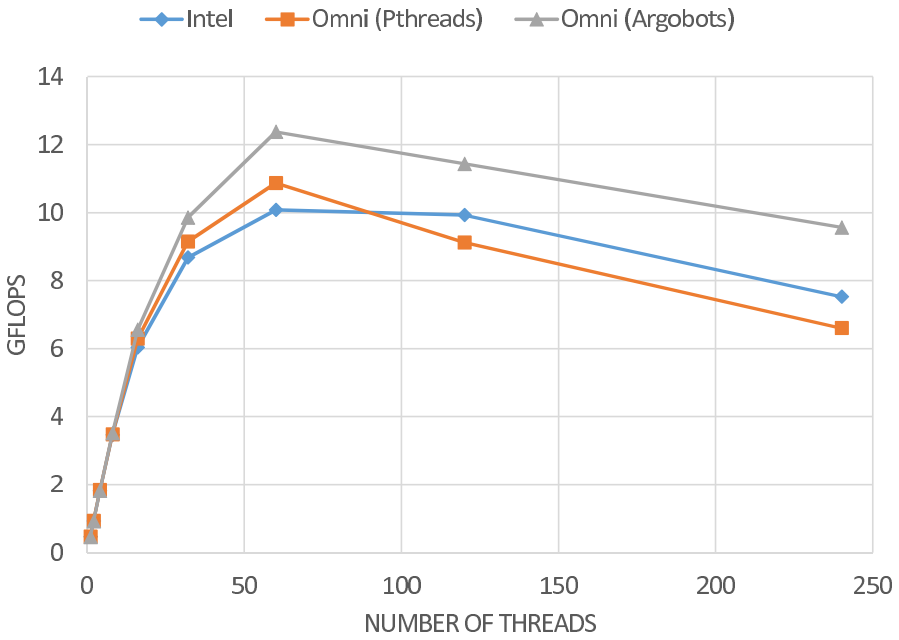


Argobots Execution Model

Research Activities and Status

- Research activities
  - Tasklets vs. ULTs
    - If the compiler can analyze and guarantee that an OpenMP thread or task does not have any blocking calls (e.g., pure computation)
  - Codesigned OpenMP compiler for MPI communication
    - Funneled-mode or restricted parallel communication
    - Separation of blocking and nonblocking communication on separate streams
- Initial work in progress





More than 3X performance improvement in some cases

Argonne NATIONAL LABORATORY

Efficient Hybrid MPI+Threads Programming

U.S. DEPARTMENT OF ENERGY

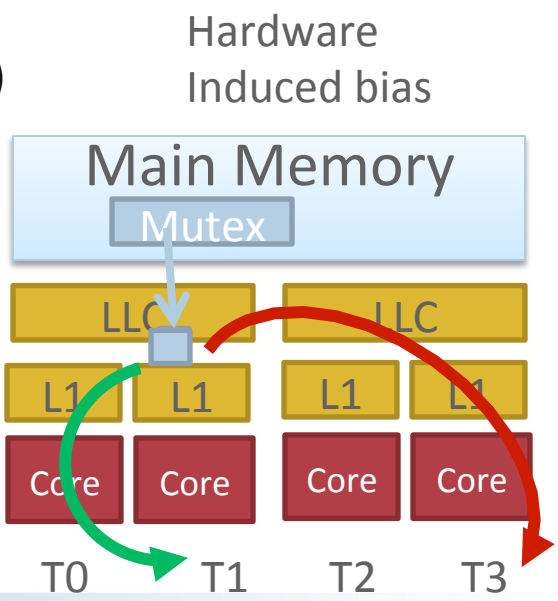
Contention in a Multithreaded MPI Model

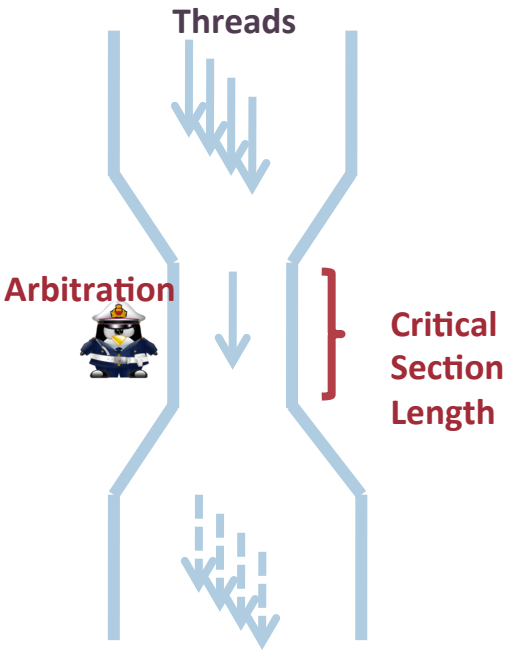
- Large gap between single and multithreaded performance
- Aspects to consider
  - Critical Section Granularity
    - Static code-level granularity
    - Dynamic runtime granularity (for blocking calls)
  - Critical Section Arbitration
    - Hardware induced bias (NUCA)
    - Lack of correlation between messages and CS arbitration
  - Locality awareness
    - Low latency hand-off
    - Cache-friendly arbitration

Collaborators:
- US: Abdelhalim Amer (Argonne)
- JP: Satoshi Matsuoka (Tokyo Tech)

```
MPI_Call(...)
{
  CS_ENTER;
  Local state
  Shared state
  Progress();
  Global Read-only
  ...
  CS_EXIT;
}
```

```
while (!req_complete)
{
    poll();
    CS_EXIT;
    CS_YIELD;
    CS_ENTER;
}
```





ANL-Tokyo Tech: Sharing designs and requirements for hybrid communication and threading models

TOKYO TECH Pursuing Excellence

Apps + Runtime

Runtime System

Pros and Cons of MPI+Threads at Large Scale?

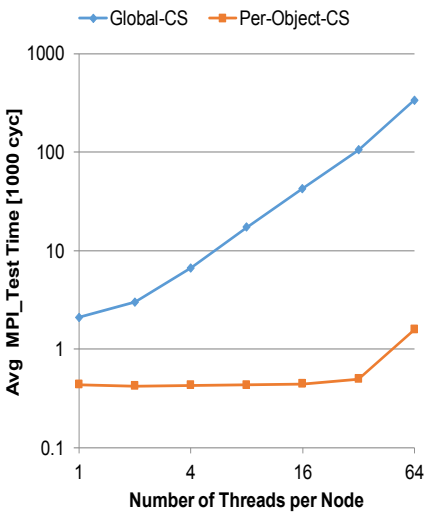
Runtime Contention in Multithreaded MPI due to Thread-Safety

Characterizing Large Scale MPI + Threads[PPMM15, Longer version *Submitted to Journal*]

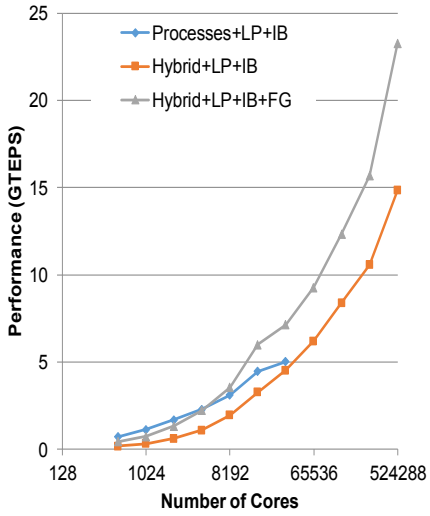
Reducing Contention by Improving Critical Section Arbitration [ACM PPoPP15]

MPI+Threads Alleviates some Drawbacks of MPI-only but Incurs Runtime Contention

Runtime Contention!

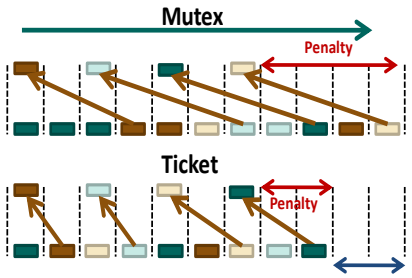


MPI_Test Latency

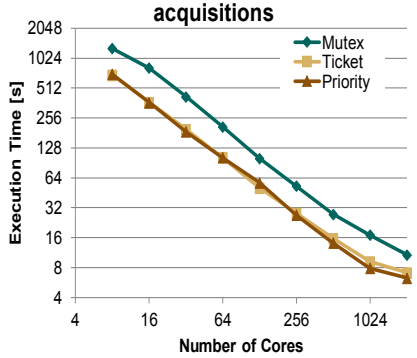


Weak Scaling Performance

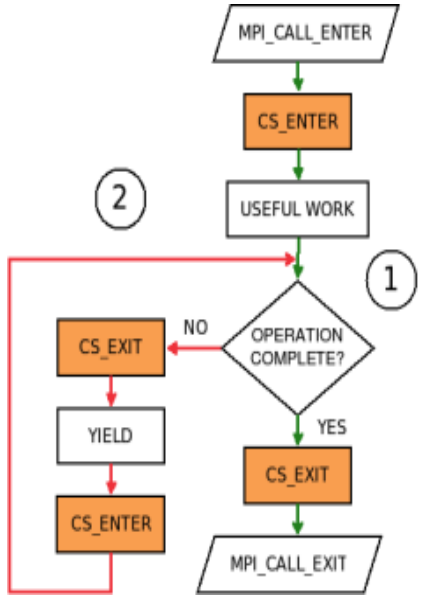
Better Arbitration Policies Implemented



Fairness (FIFO) reduces wasted resource acquisitions



Genome Assembly application results strong scaling



2-Level arbitration policy: (1) has a higher priority