RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY

Addressing Big Data Challenges in Simulation-based Science

Manish Parashar*

Rutgers Discovery Informatics Institute (RDI2)
NSF Center for Cloud and Autonomic Computing (CAC)
Department of Electrical and Computer Engineering

http://dataspaces.org

***Hoang Bui, Tong Jin, Qian Sun, Fan Zhang and other ex-students and collaborators….**

© Manish Parashar, Rutgers University

Outline

- ***Data Grand Challenges***
  - Data challenges of simulation-based science
- Rethinking the simulations -> insights pipeline
- The ADIOS/DataSpaces Project
- Conclusion

RUTGERS

Modern Science & Society Transformed by Compute & Data

New paradigms and practices in science and engineering

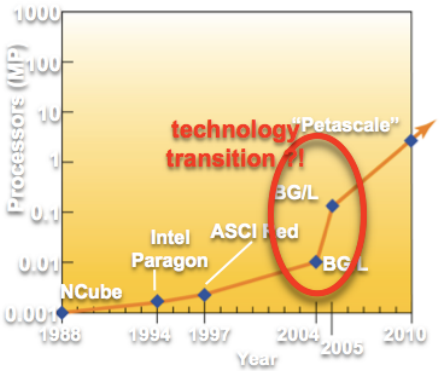
- Inherently multi-disciplinary
- Data-driven, data and compute-intensive
- Collaborative (university, national, global)

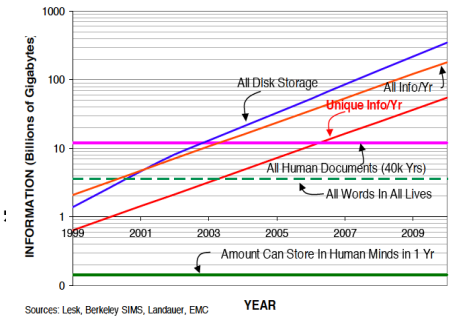
Many Challenges

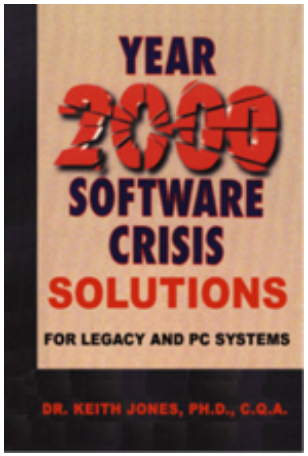
- Computing, Data, Software, People

Many Challenges….

- Computing
  - Multicore; large and increasing core counts, deep memory hierarchies (TH-2: 54.9 PF, 3.12 M Cores, 1.4 PB, 25 MW)
  - New prgm. model, concerns (fault tolerance, energy, etc)
  - New models & technologies: Clouds, grids, hybrid manycore, accelerators, deep storage hierarchies, …
- Data
  - Generating more data than in all of human history: preserve, mine, share?
  - How do we create "data scientists/engineers"?
- Software
  - Complex applications on coupled compute-data-networked environments, tools needed
  - Modern apps: 10^6+ lines, many groups contribute, take decades to develop, very long lifetimes
- People
  - Multidisciplinary expertise essential!
    - Appropriate academic program, career tracks…



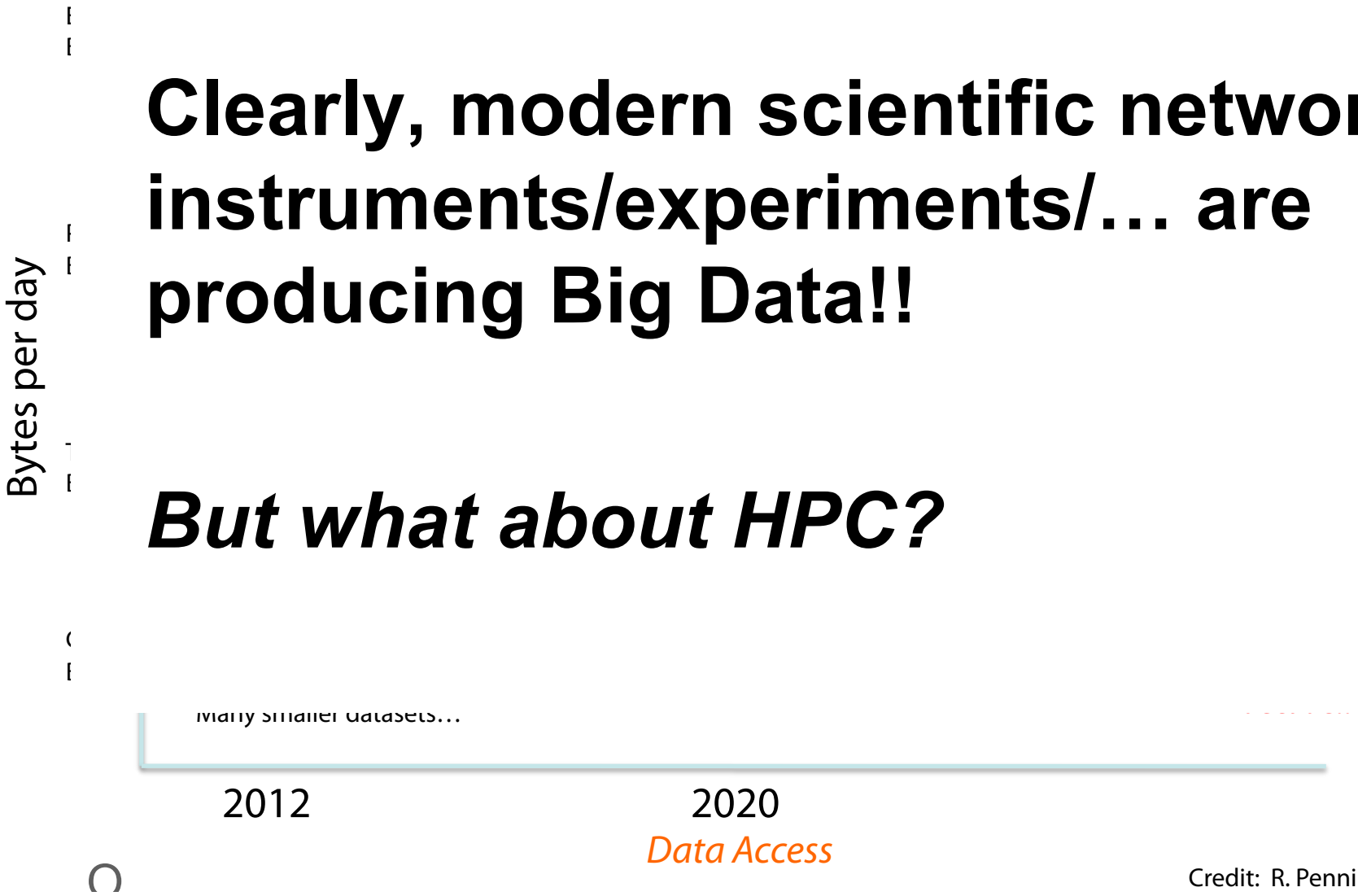




The Era of eScience and Big Data

Clearly, modern scientific network/ instruments/experiments/… are producing Big Data!!

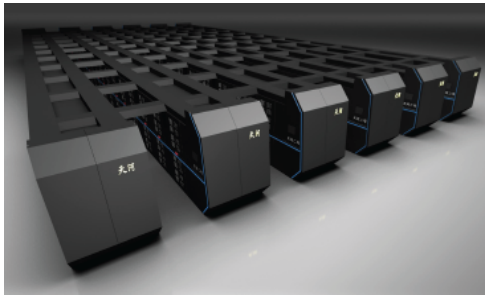
But what about HPC?



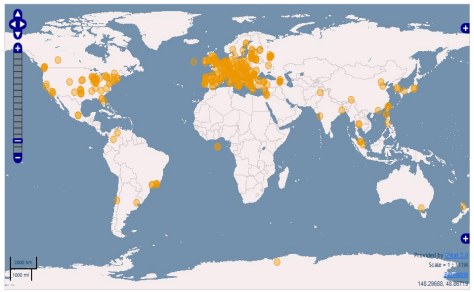
Credit: R. Pennington/A. Blatecky

Advanced Computing Infrastructure

- Large scale, distributed, heterogeneous, multicore/manycore, accelerators, deep storage hierarchies, experimental systems



TH-2
- 54.9 PF / 3.12 M cores
- 32K Xeon + 48K Phi
- TH Express-2
- 1PB memory

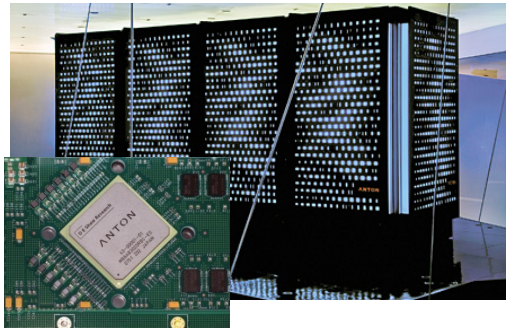


Worldwide LHC Computing Grid
- >140 sites;
- ~250k cores;
- ~100 PB disk

Titan - Cray XK7
- 20+ PF / 300 K CPU cores
- 18,688 GPUs
- Gemini 3D torus
- 710TB memory

Modern Datacenters
- 1M servers
- 50-100 MW

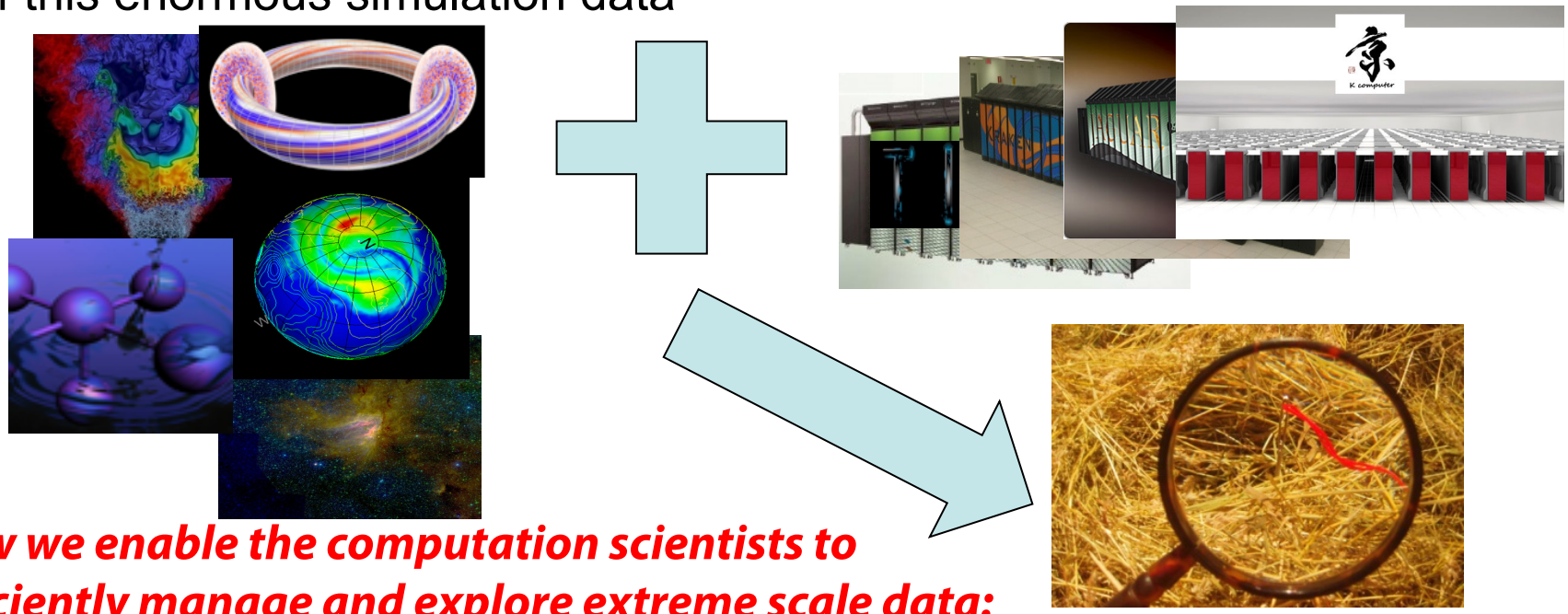
Sequoia – IBM BG/Q
- 20 PF / 1.5 M cores
- 18-core processor
- 5D torus
- 1.5PB memory



Special Purpose HW (Anton)
- > 100 time acceleration of MD simulations

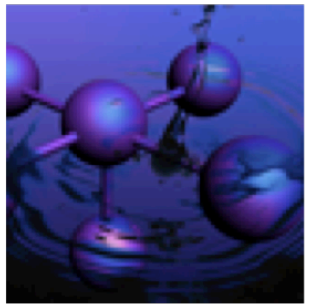
Scientific Discovery through Simulations

- Scientific simulations running on high-end computing systems generate huge amounts of data!
  - If a single core produces 2MB/minute on average, one of these machines could generate simulation data between ~**170TB** per hour -> ~**700PB** per day -> ~**1.4EB** per year
- Successful scientific discovery depends on a comprehensive understanding of this enormous simulation data

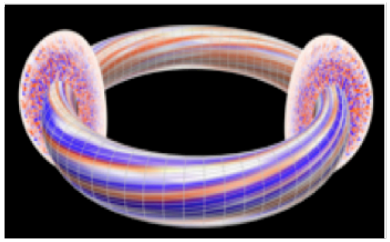


How we enable the computation scientists to efficiently manage and explore extreme scale data: "find the needles in haystack" ??

Scientific Discovery through Simulations

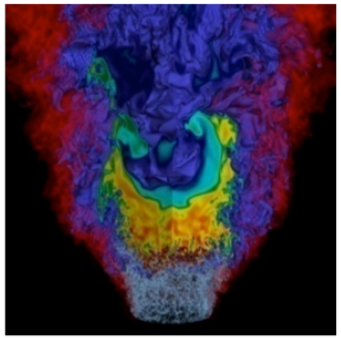


Molecular Simulation

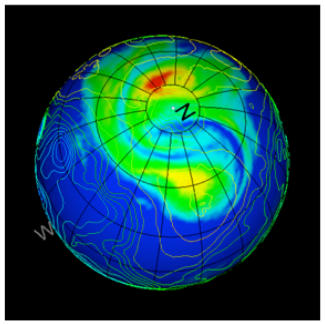


Plasma Fusion

Astrophysics



Combustion



Climate Modeling

- Complex workflows integrating coupled models, data management/ processing, analytics
  - Tight / loose coupling, data driven, ensembles
- Advanced numerical methods (E.g., Adaptive Mesh Refinement)
- Integrated (online) uncertainty quantification, analytics
- Complex, heterogeneous components
- Large data volumes and data rates
- Data re-distribution (MxNxP), data transformations
- Dynamic data exchange patterns
- Strict performance/overhead constraints

Traditional *Simulation -> Insight* Pipelines Break Down

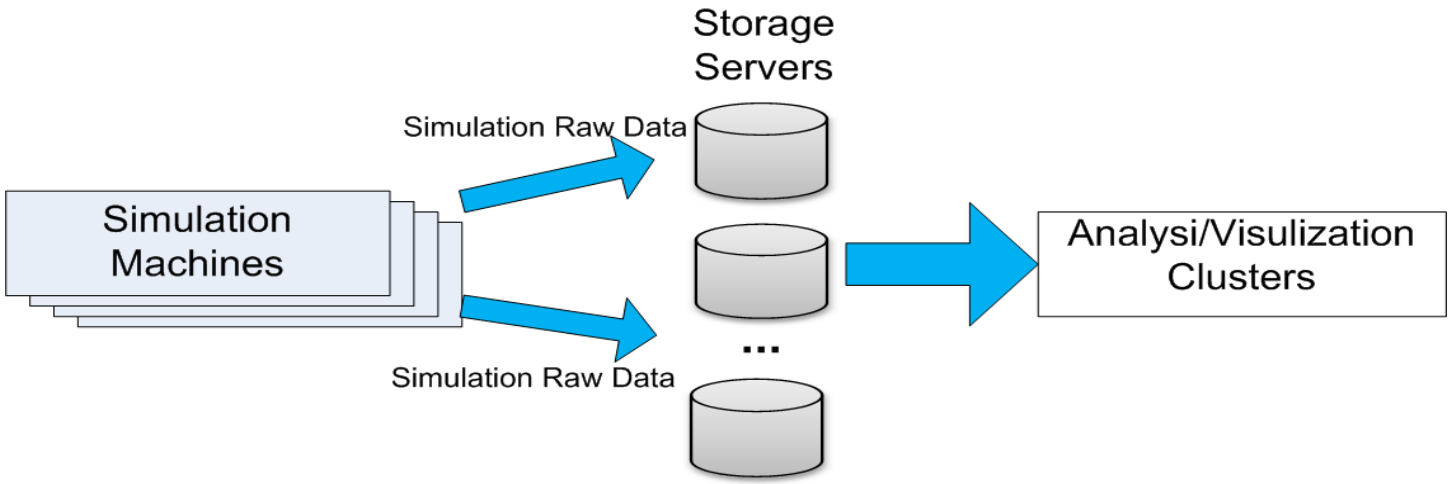


Figure. Traditional data analysis pipeline

- Traditional ***simulation -> insight*** pipeline:
  - Run large-scale simulation workflows on large supercomputers
  - Dump data on parallel disk systems
  - Export data to archives
  - Move data to users' sites – usually selected subsets
  - Perform data manipulations and analysis on mid-size clusters
  - Collect experimental / observational data
  - Move to analysis sites
  - Perform comparison of experimental/observational to validate simulation data

Challenges Faced by Traditional HPC Data Pipelines

- **Data analysis challenge**
  - Can current data mining, manipulation and visualization algorithms still work effectively on extreme scale machine?

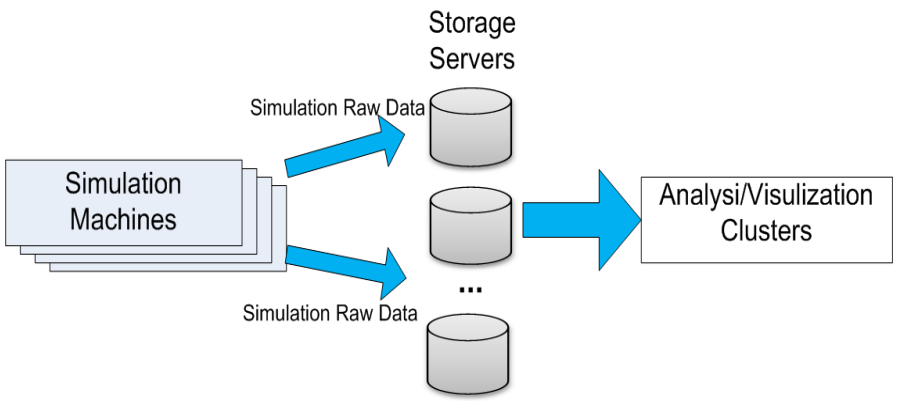


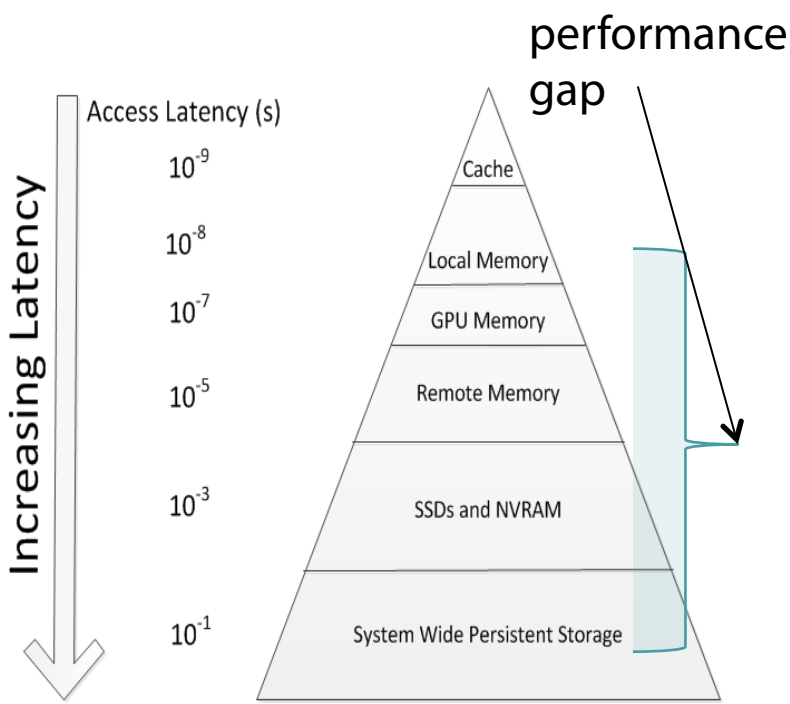
Figure. Traditional data analysis pipeline

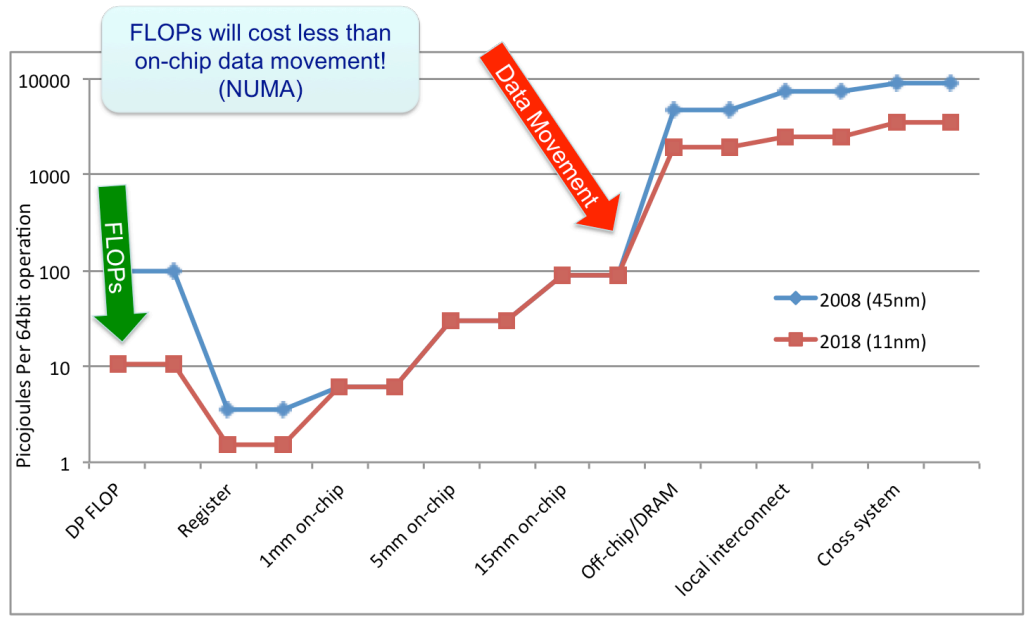
- **I/O challenge**
  - Increasing performance gap: disks are outpaced by computing speed
- **Data movement challenge**
  - Lots of data movement between simulation and analysis machines, between coupled mutli-physics simulation components -> longer latencies
  - Improving data locality is critical: do work where the data resides!
- **Energy challenge**
  - Future extreme systems are designed to have low-power chips – however, much greater power consumption will be due to memory and data movement!

*The **costs of data movement** are increasing and dominating!*

The Cost of Data Movement

- Moving data between node memory and persistent storage is slow!
- The energy cost of moving data is a significant concern





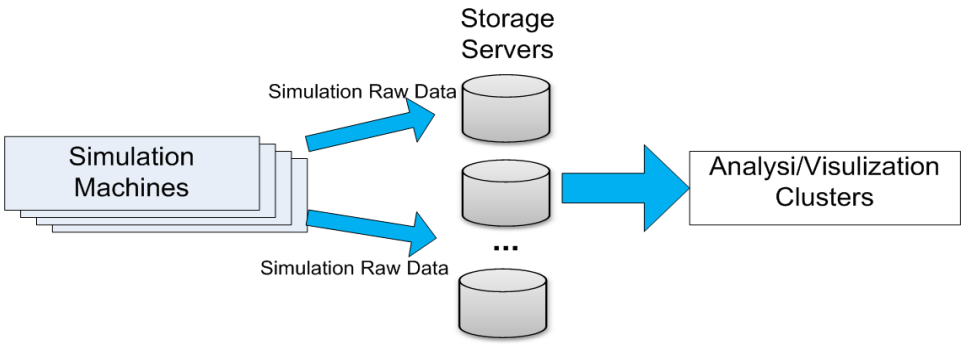
$$\text{Energy_move_data} = \frac{\text{bitrate} * \text{length}^2}{\text{cross_section_area_of_wire}}$$

From K. Yelick, “Software and Algorithms for Exascale: Ten Ways to Waste an Exascale Computer”

Challenges Faced by Traditional HPC Data Pipelines

*The **costs of data movement** (power and performance) are increasing and dominating!*

Traditional data analysis pipeline



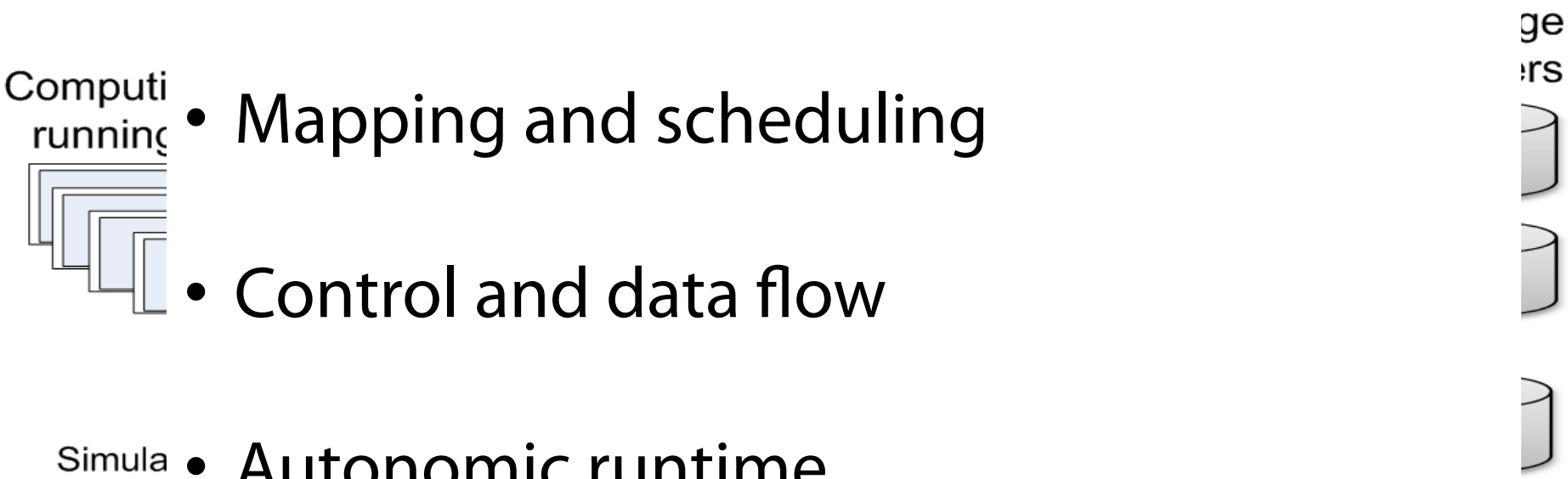
We need to Rethink the Data Management Pipeline!

- Reduce data movement
- Move computation/analytics closer to the data
- Add value to simulation data along the IO path

Rethinking the Data Management Pipeline – Hybrid Staging + In-Situ & In-Transit Execution

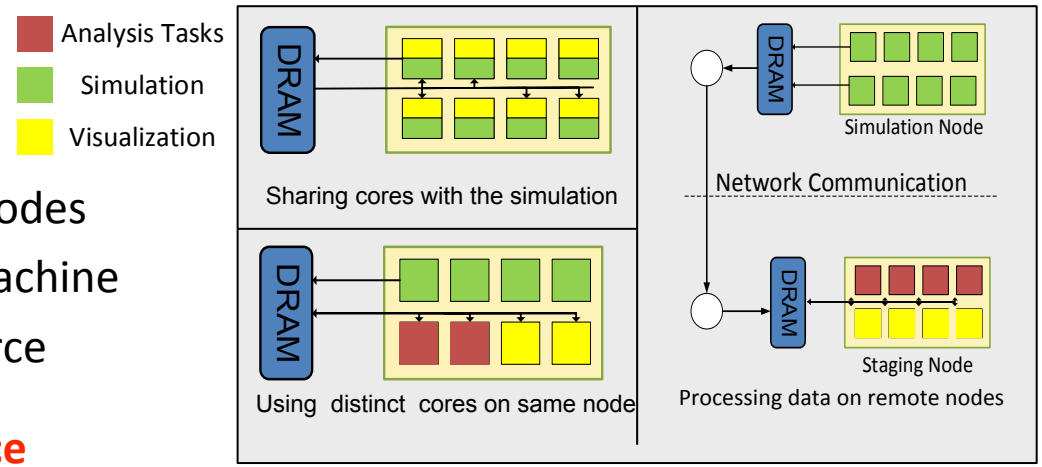
Issues/Challenges

- Programming abstractions/systems
- Mapping and scheduling
- Control and data flow
- Autonomic runtime

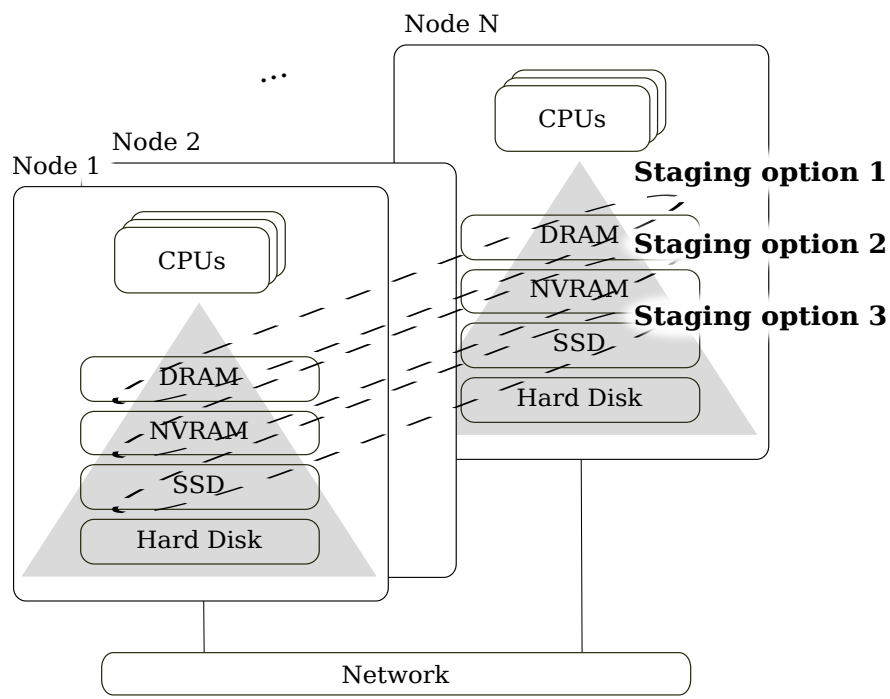


Design space of possible workflow architectures

- **Location of the compute resources**
  - Same cores as the simulation (in situ)
  - Some (dedicated) cores on the same nodes
  - Some dedicated nodes on the same machine
  - Dedicated nodes on an external resource



- **Data access, placement, and persistence**
  - Direct access to simulation data structures
  - Shared memory access via hand-off / copy
  - Shared memory access via non-volatile near node storage (NVRAM)
  - Data transfer to dedicated nodes or external resources

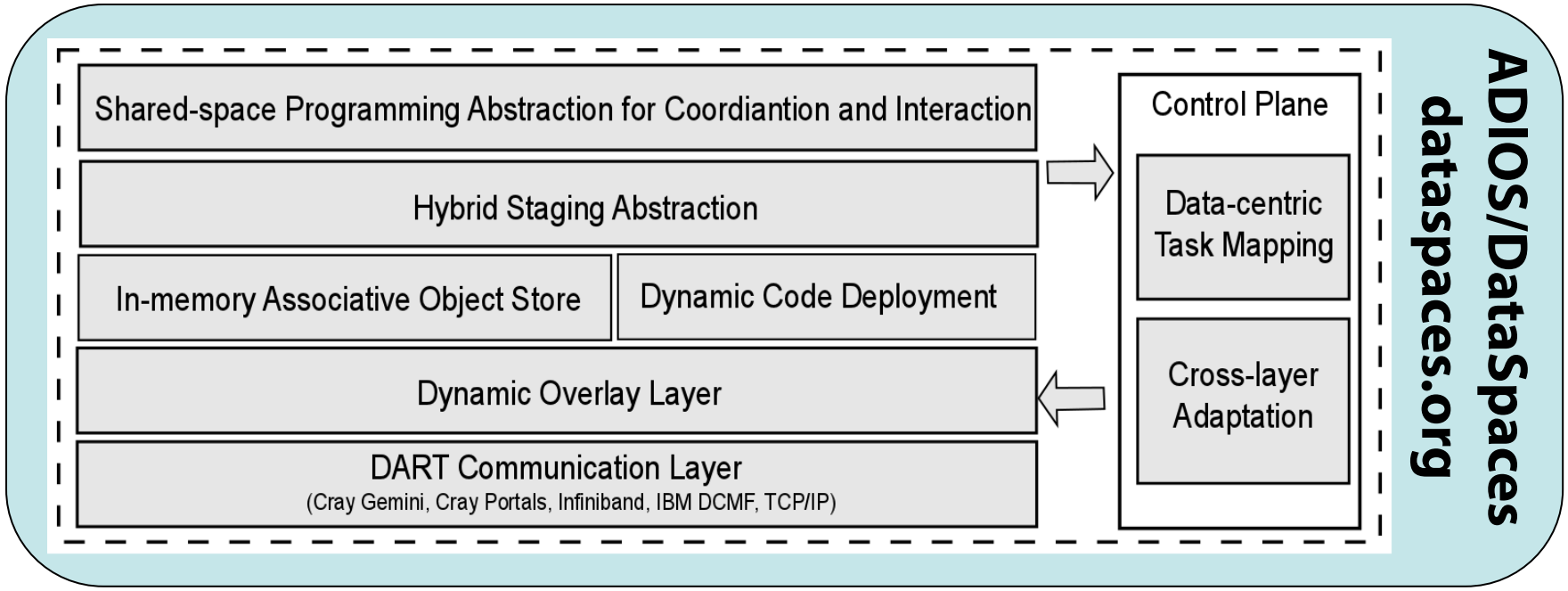


- **Synchronization and scheduling**
  - Execute synchronously with simulation every n^{th} simulation time step
  - Execute asynchronously

RUTGERS

DataSpaces

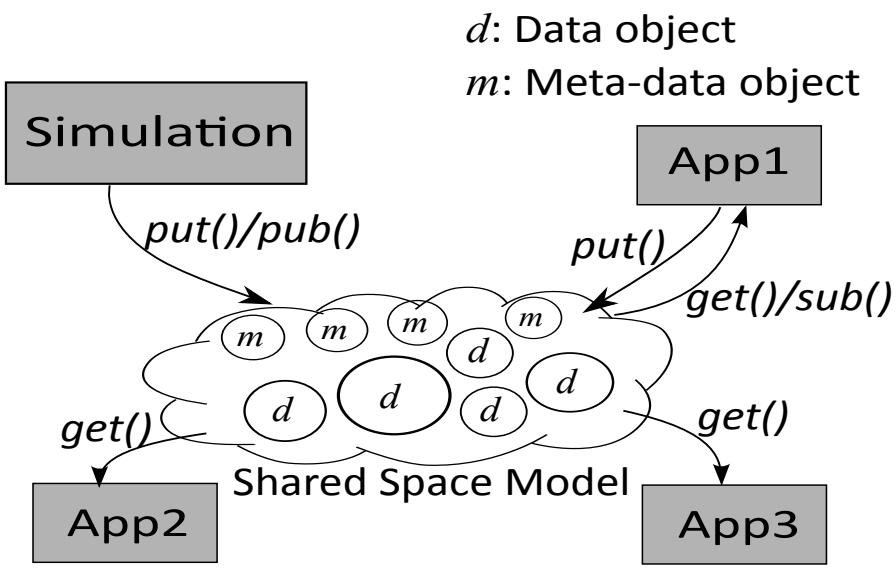
In-situ/In-transit Data Management & Analytics



- Virtual shared-space programming abstraction
  - Simple API for coordination, interaction and messaging
- Distributed, associative, in-memory object store
  - Online data indexing, flexible querying
- Adaptive cross-layer runtime management
  - Hybrid in-situ/in-transit execution
- Efficient, high-throughput/low-latency asynchronous data transport

DataSpaces: A Scalable Shared Space Abstraction for Hybrid Data Staging [HPDC10, JCC12]

- Virtual shared-space abstraction
  - Simple API for coordination, interaction and messaging
  - Provides a global-view programming abstraction consistent with PGAS
  - Distributed, associative, in-deep-memory object store
  - Online data indexing, flexible querying

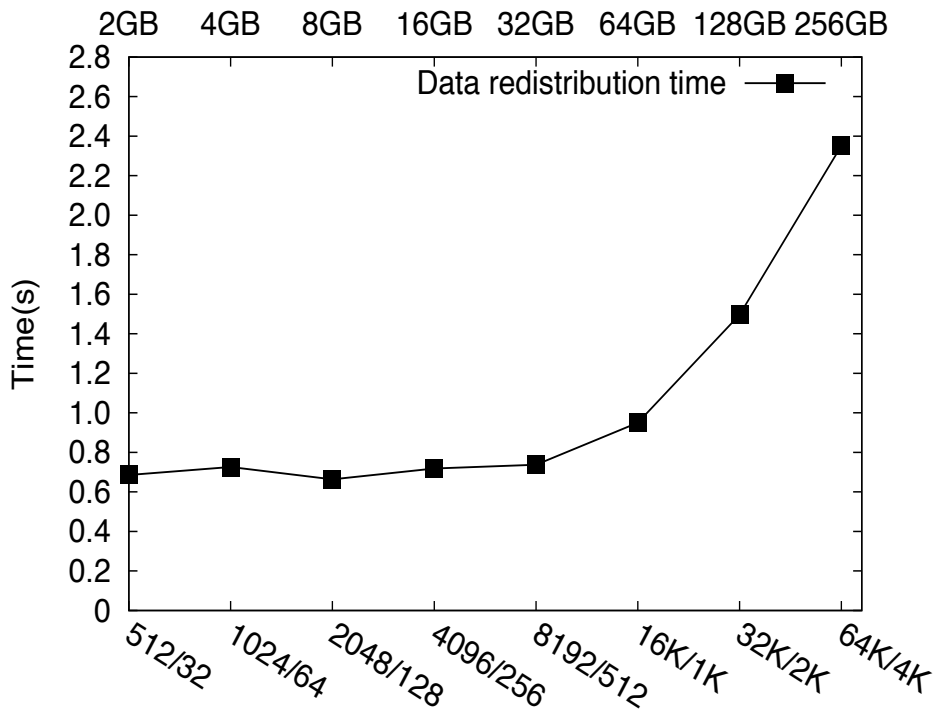


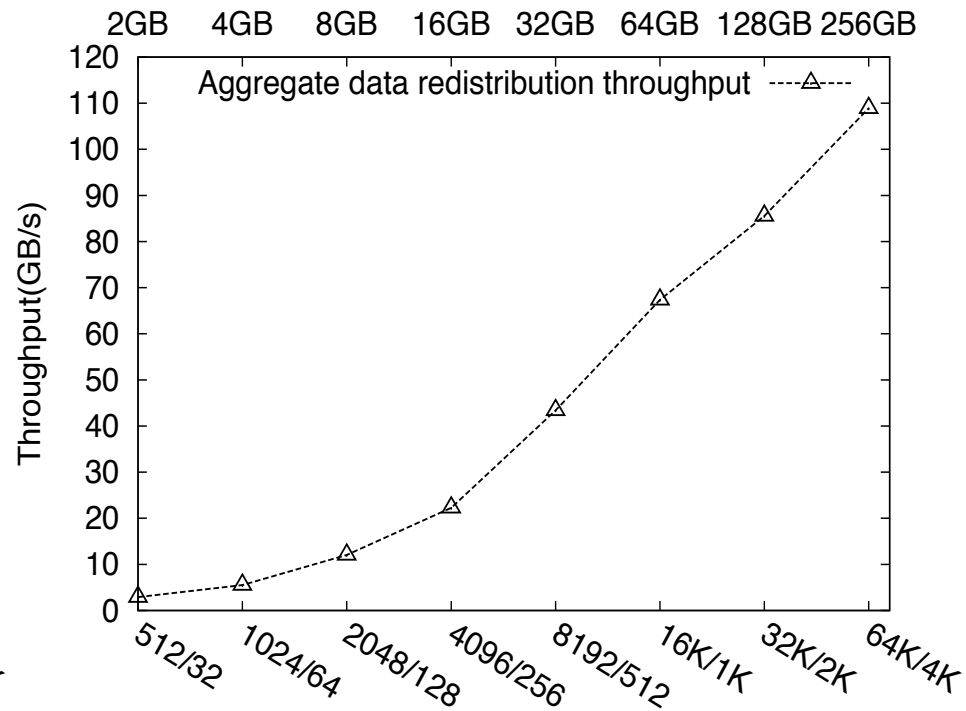
- Adaptive cross-layer runtime management
  - Hybrid in-situ/in-transit execution
  - Data-centric mappings

- High-throughput/low-latency memory-to-memory asynchronous data transport

- Dynamic coordination and interaction patterns between the coupled applications
  - Transparent data redistribution
  - Complex geometry-based queries
  - In-space (online) data transformation and manipulations

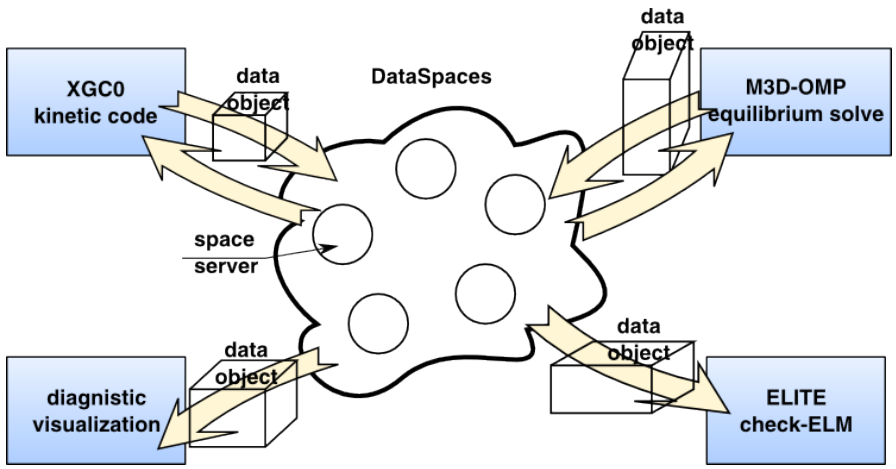
DataSpaces: Scalability on ORNL Titan



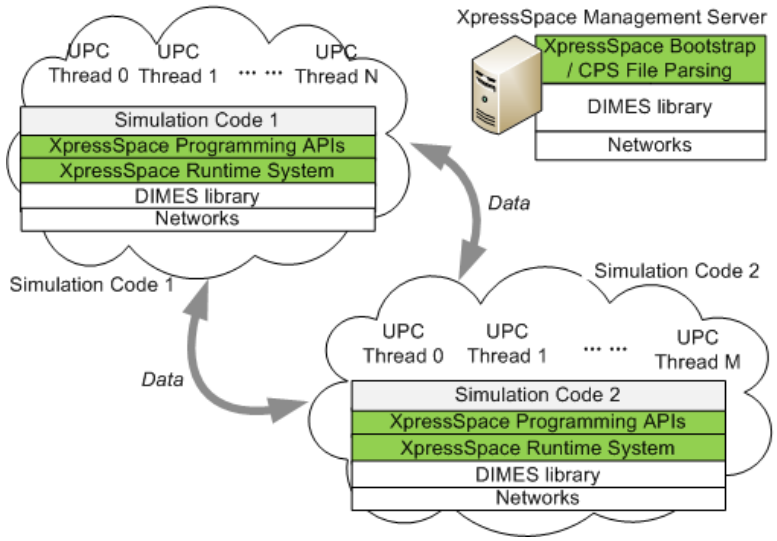


- Evaluate weak scaling with an increasing number of processors
- Applications redistribute data through DataSpaces
  - Application 1 runs on M processors and insert data in the space
  - Application 2 runs on N processors and retrieve data from the space
- Result: A 128 fold increase in the application sizes from 512 to 64K writers, total data size exchanged per step is increased from 2GB to 256GB

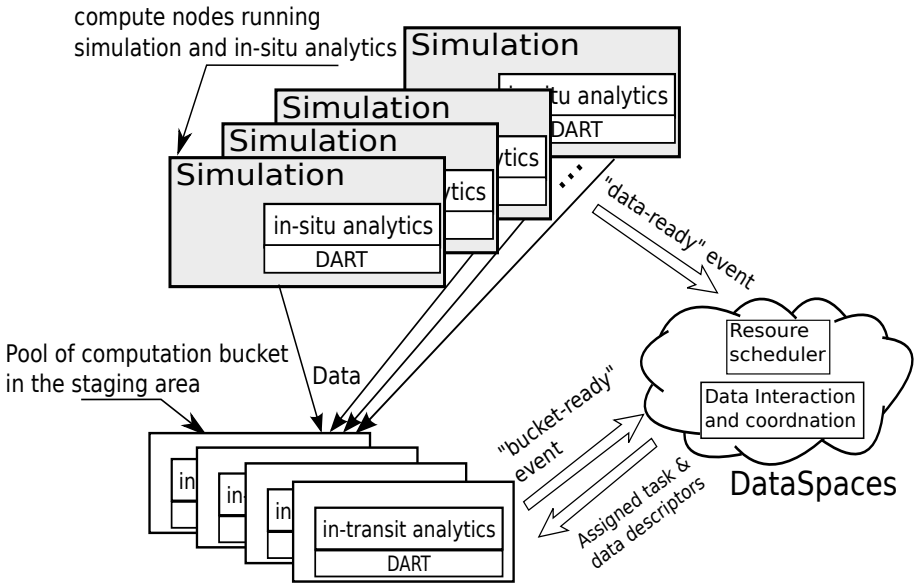
DataSpaces: Enabling Coupled Scientific Workflows at Extreme Scales



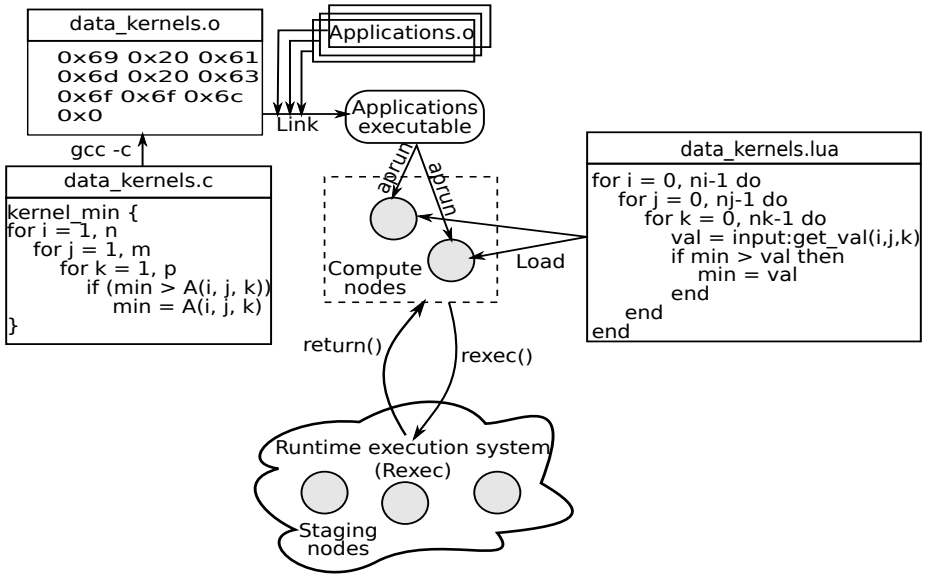
Multphysics Code Coupling at Extreme Scales [CCGrid10]



PGAS Extensions for Code Coupling [CCPE13]



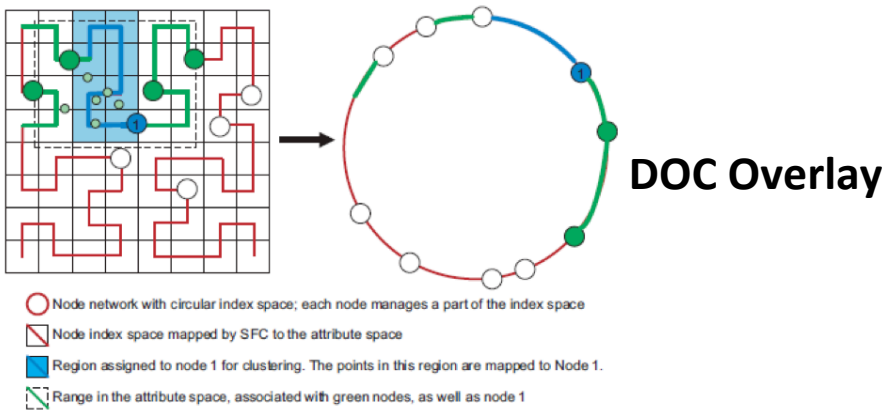
Data-centric Mappings for In-Situ Workflows [IPDPS12]

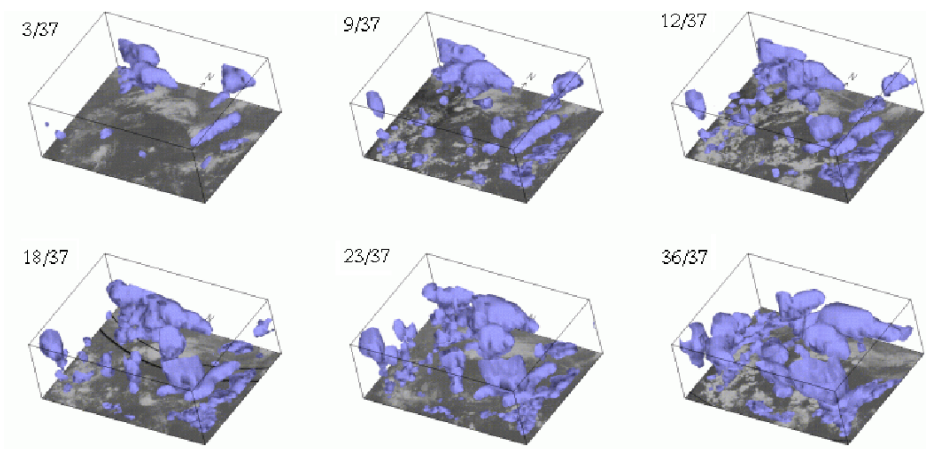


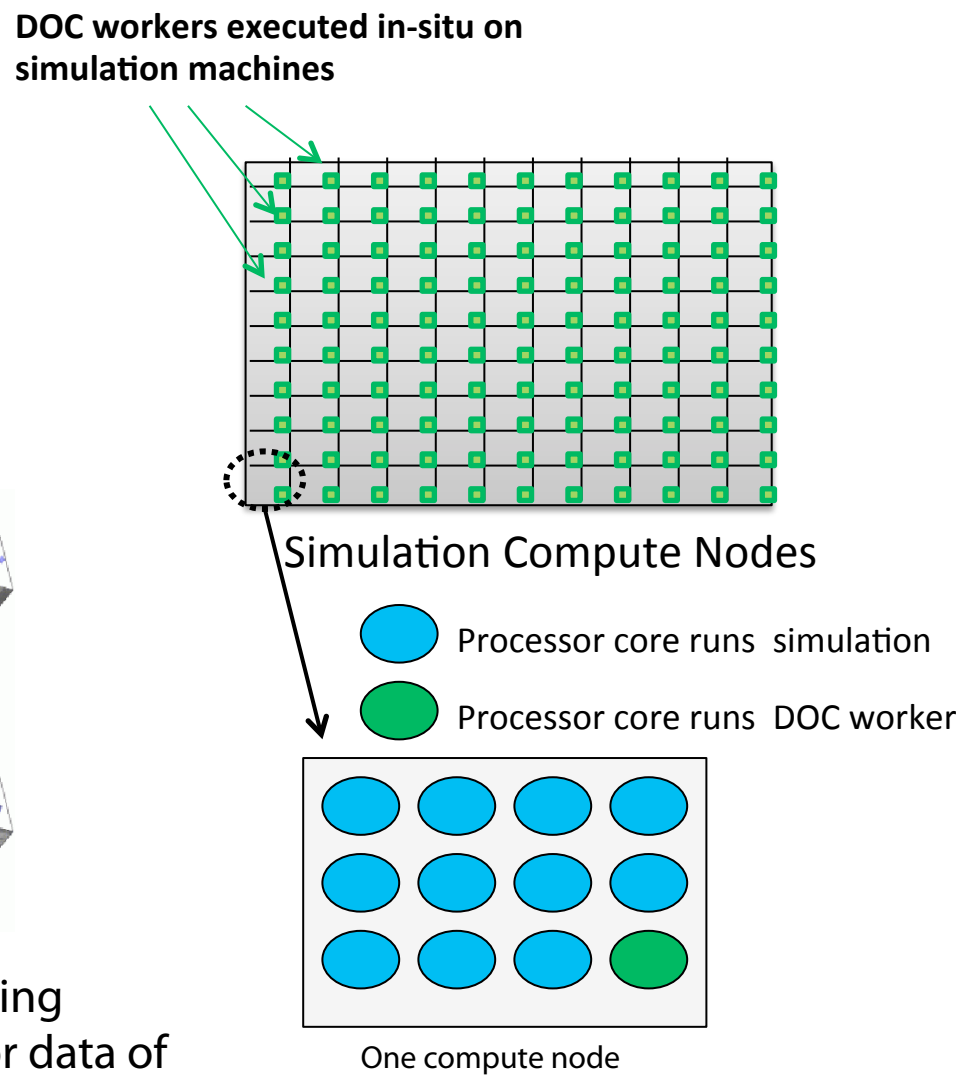
Dynamic Code Deployment In-Staging [IPDPS11]

RUTGERS

In-Situ Feature Extraction and Tracking using Decentralized Online Clustering (DISC'12, ICAC'10)



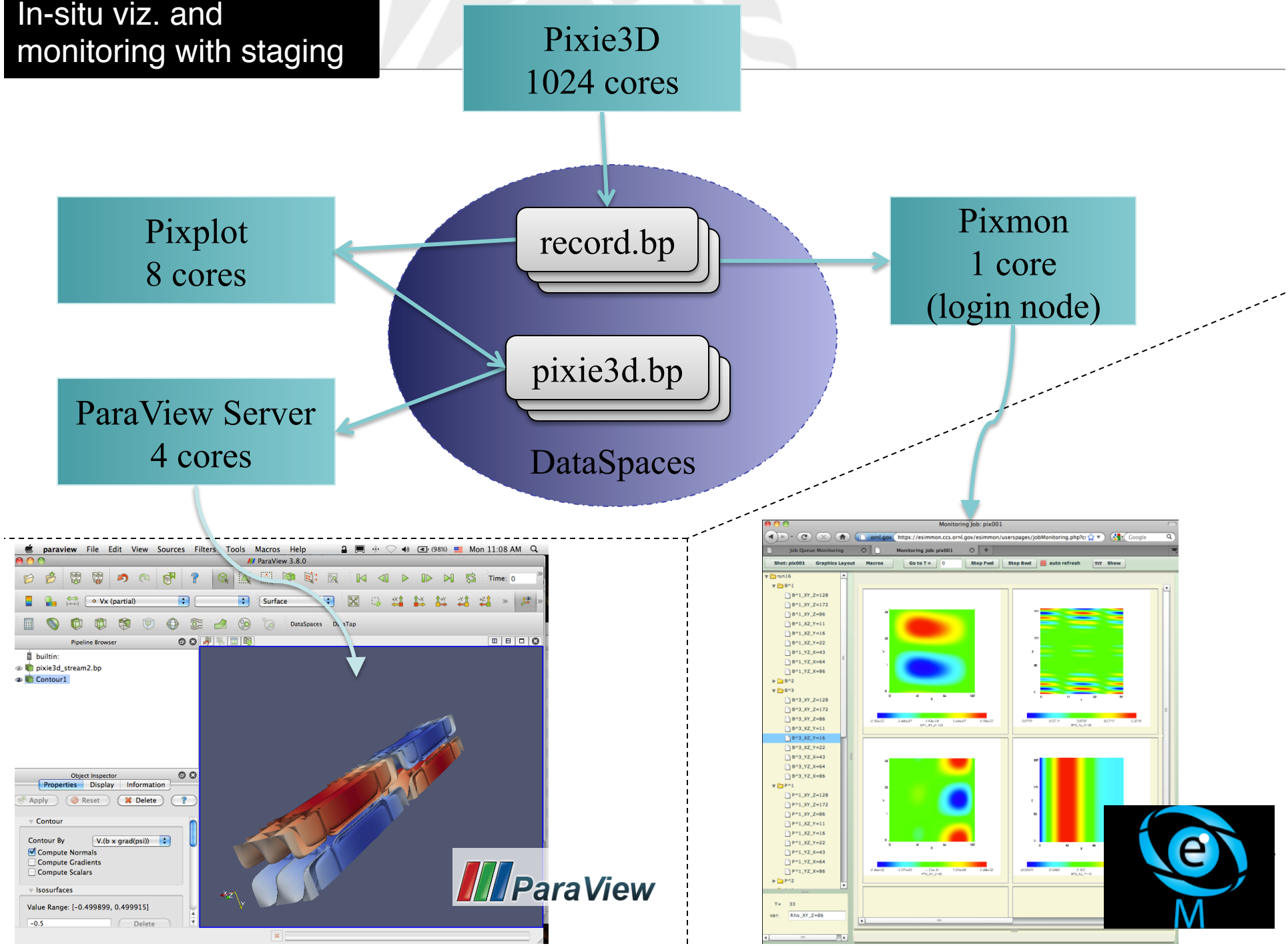




Benefits of runtime feature extraction and tracking

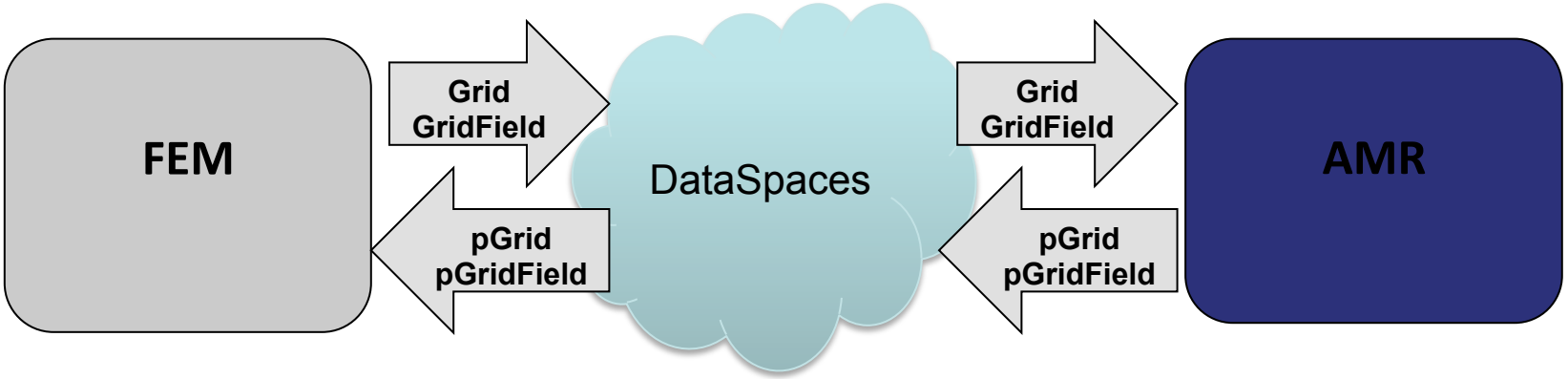
(1) Scientists can follow the events of interest (or data of interest)

In-situ viz. and monitoring with staging



AMR-as-a-Service using DataSpaces

- FEM-AMR workflow



- **Components:**
  - FEM: model uniform 3D mesh of near-realistic engineering problems such as heat-transfer, fluid flow and phase transformation(Grid and GridField)
  - AMR: localize areas of interest where the physics is important (pGrid and
  - pGridField) to allow truly realistic simulations

- **Goals:**
  - Enable in-memory data coupling between FEM and AMR code
  - Allow multiple AMR codes to be plugged in and read Grid/GridField data as FEM progresses.

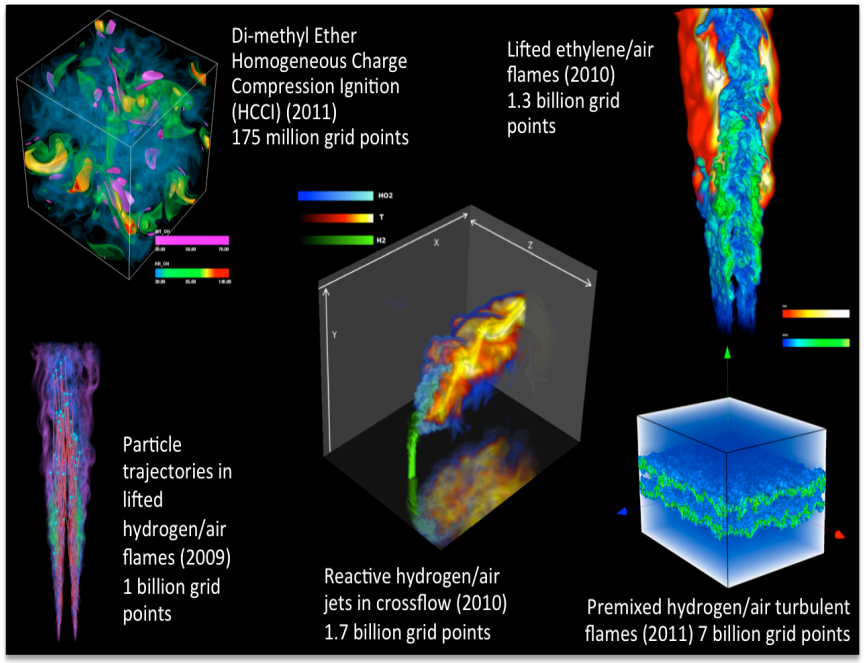
Overview of Research Work @ RU

- Programming abstractions / system
  - **DataSpaces:** Interaction, coordination and messaging abstractions for coupled scientific workflow [HPDC10, CCGrid10, HiPC12, JCC12]
  - **XpressSpaces:** PGAS extensions for coupling using DataSpaces [CCGrid11, CCPE13]
  - **ActiveSpaces:** Dynamic code deployment for in-staging data processing [IPDPS11]
- Runtime mechanisms
  - **Data-centric Task Mapping:** Reduce data movement and increase intra-node data sharing [IPDPS12, DISC12]
  - **In-situ & In-transit Data Analytics**: Simulation-time analysis of large volume data by combining in-situ and in-transit execution [SC12, DISC12]
  - **Cross-layer Adaptation:** Adaptive cross-layer approach for dynamic data management in large scale simulation-analysis workflow [SC13]
  - **Value-based Data Indexing and Querying** : Use FastBit to build in-situ, in-memory value-based indexing and query support in the staging area
  - **Power/performance Tradeoffs**: Characterizing power/performance tradeoffs for data-intensive simulations workflows [SC13]
  - **Data Staging over Deep Memory Hierarchy**: Build distributed associative object store over hierarchical memory storage, e.g. DRAM/NVRAM/SSD [HiPC 13]
- High-throughput/low-latency asynchronous data transport
  - **DART:** Network independent transport library for high speed asynchronous data extraction and transfer [HPDC08]

Integrating In-Situ and In-Transit Analytics (SC'12)

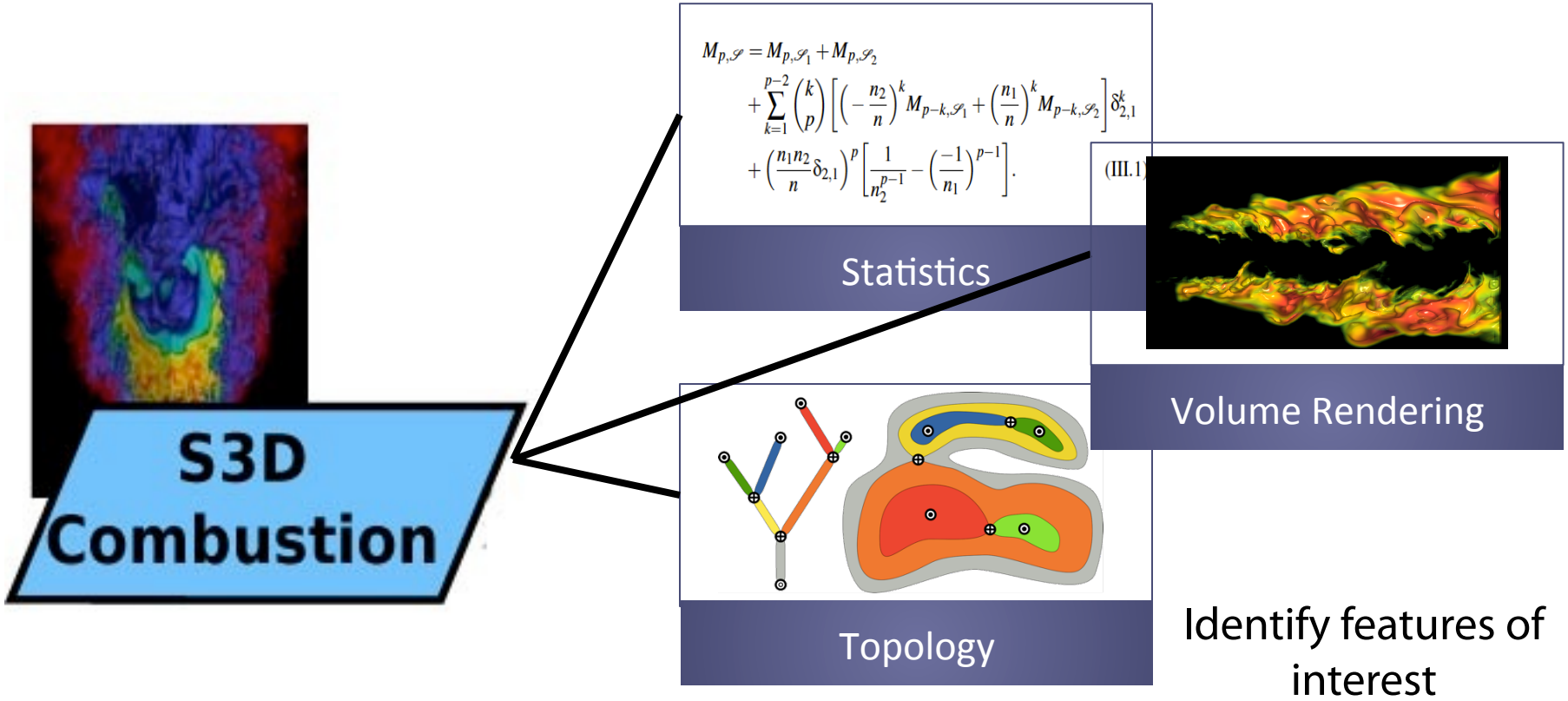
ExaCT CENTER FOR EXASCALE SIMULATION OF COMBUSTION IN TURBULENCE

- S3D: First-principles direct numerical simulation
- Simulation resolves features on the order of 10 simulation time steps
- Currently on the order of every 400th time step can be written to disk
- Temporal fidelity is compromised when analysis is done as a post-process



Recent data sets generated by S3D, developed at the Combustion Research Facility, Sandia National Laboratories

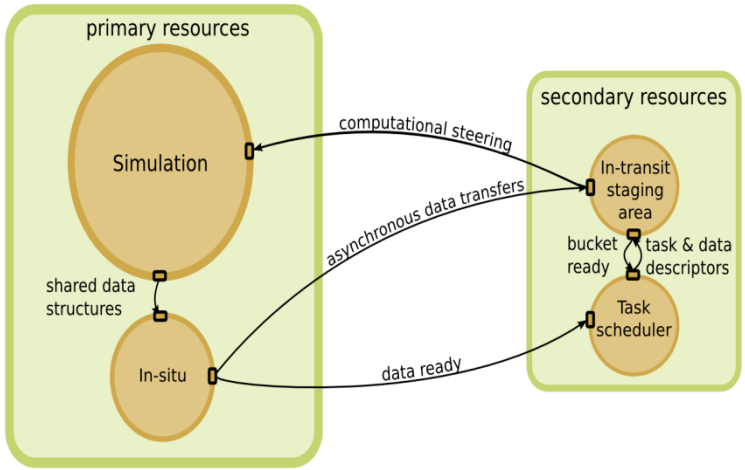
In-situ Topological Analysis as Part of S3D*

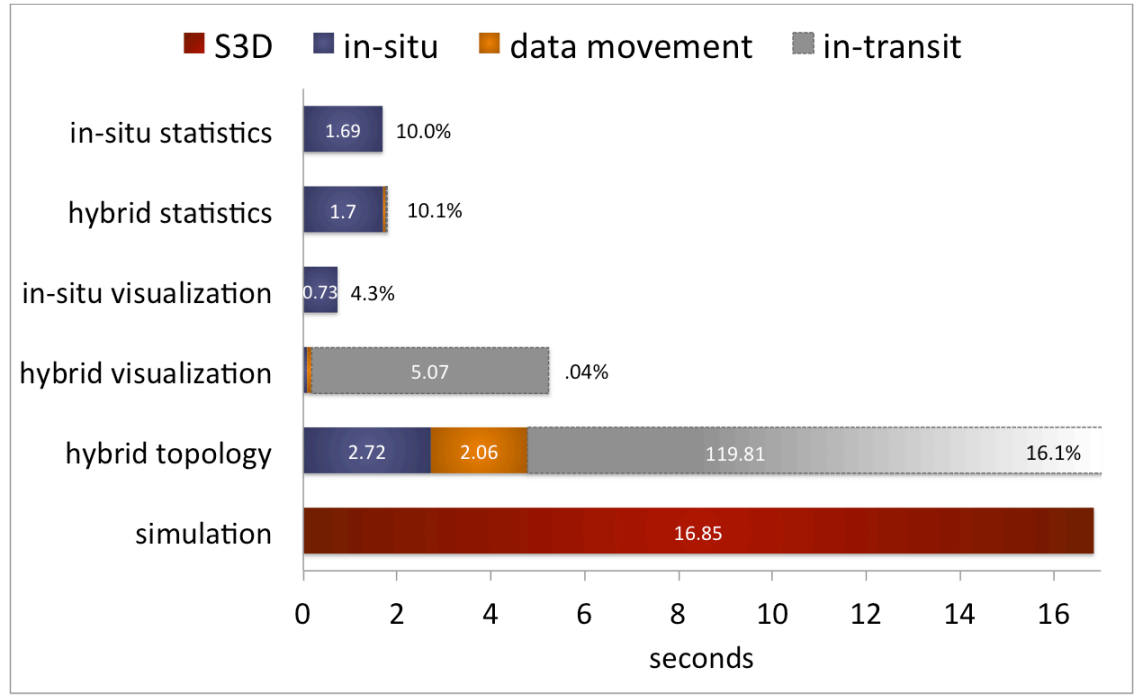


Identify features of interest

*J. C. Bennett et al., *"Combining In-Situ and In-Transit Processing to Enable Extreme-Scale Scientific Analysis"*, SC'12, Salt Lake City, Utah, November, 2012.

Integrating In-Situ and In-Transit Analytics (SC'12)



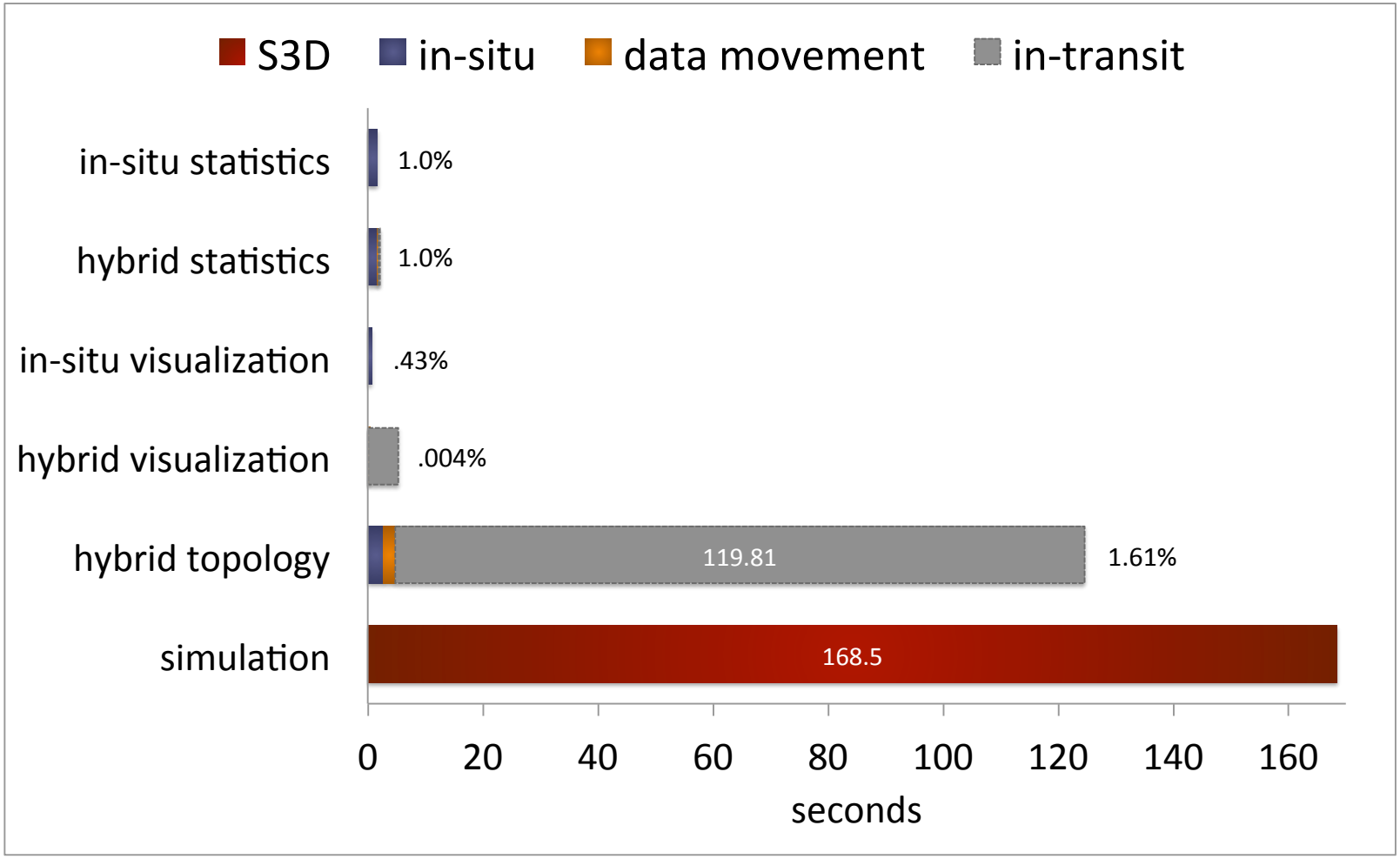


- Primary resources execute the main simulation and in situ computations
- Secondary resources provide a staging area whose cores act as buckets for in transit computations

- 4896 cores total (4480 simulation/in situ; 256 in transit; 160 task scheduling/data movement)
- Simulation size: 1600x1372x430
- All measurements are per simulation time step

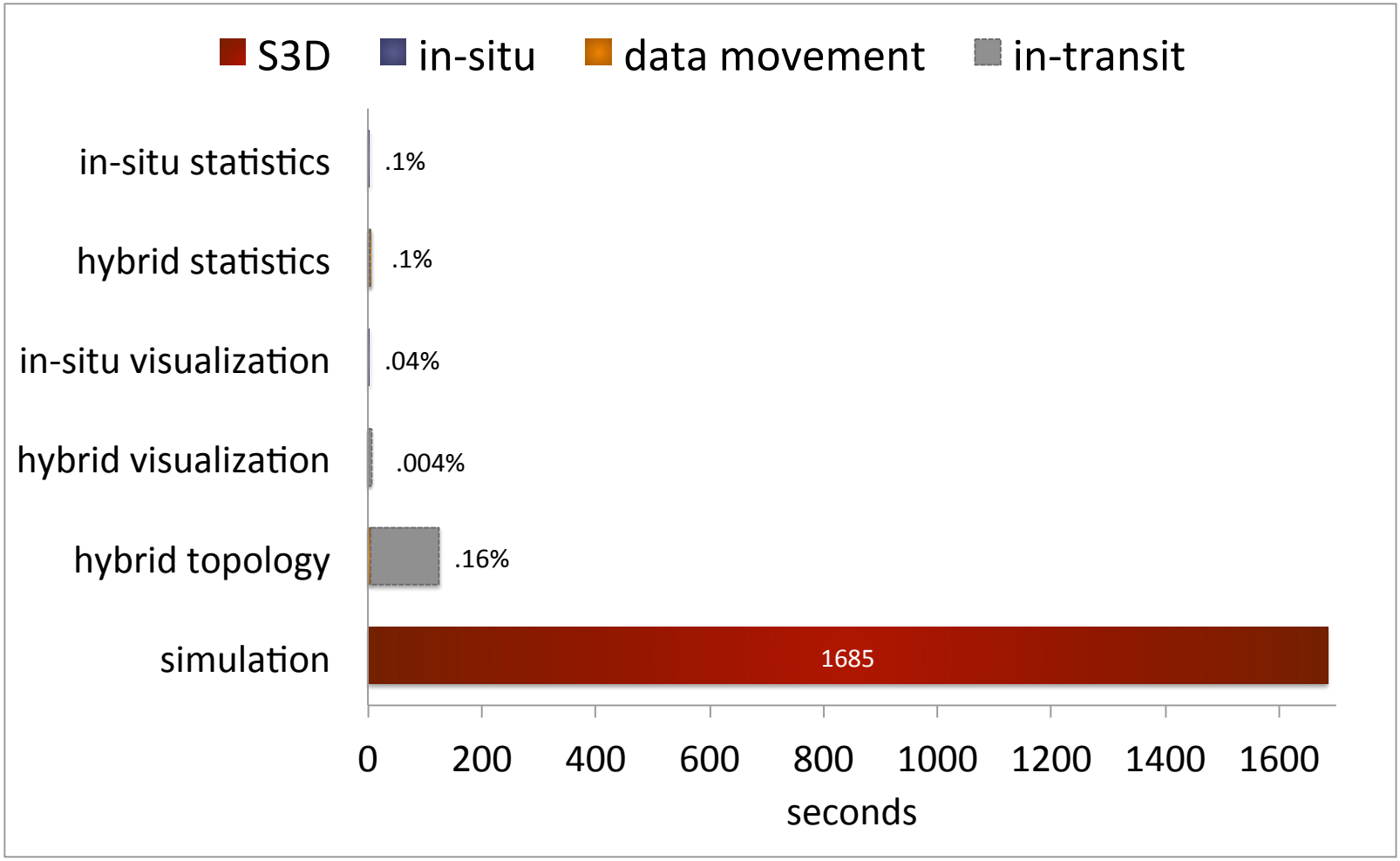
RUTGERS

Simulation case study with S3D: Timing results for 4896 cores and analysis every 10th simulation time step



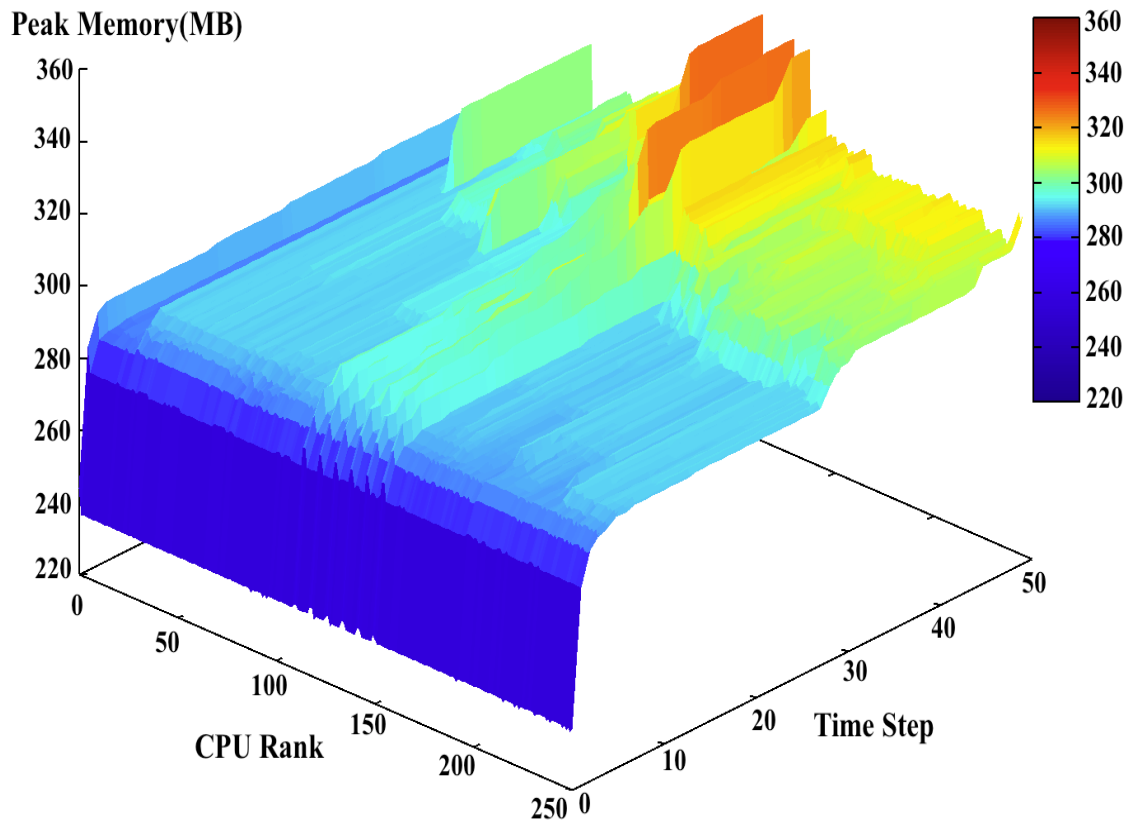
RUTGERS

Simulation case study with S3D: Timing results for 4896 cores and analysis every 100th simulation time step



Cross-Layer Adaptation for Dynamic Data Management (SC13)

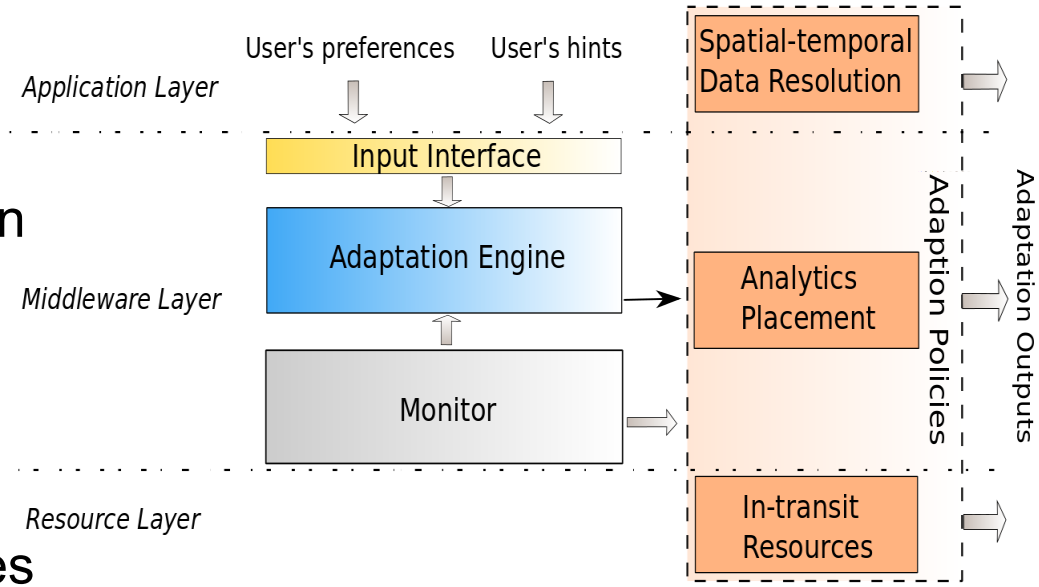
- Coupled simulation-analytics workflow based on dynamic formulations such as Adaptive Mesh Refinement (AMR) running at extreme scales present new challenges for in-situ/in-transit data management
  - Large and dynamically changing volume of data
  - Dynamic imbalanced data distribution
  - Heterogeneous resource (memory, CPU, etc.) requirements



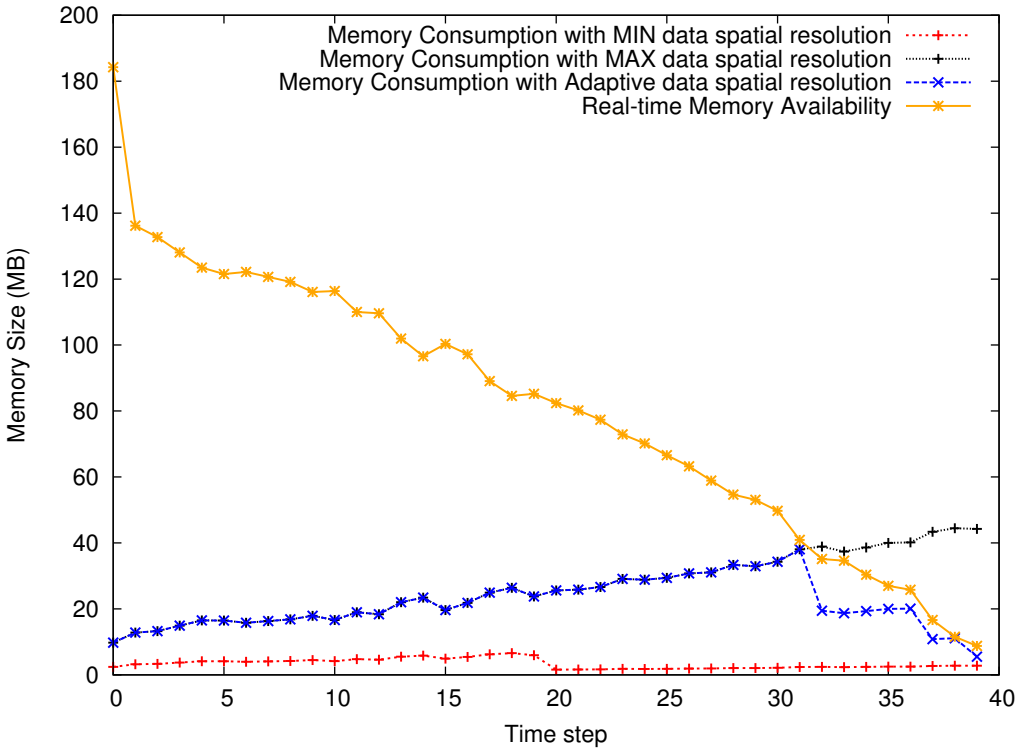
Cross-Layer Adaptation for Dynamic Data Management

Dynamic cross-layer adaptations that can respond at runtime to the dynamic data management and data processing requirements

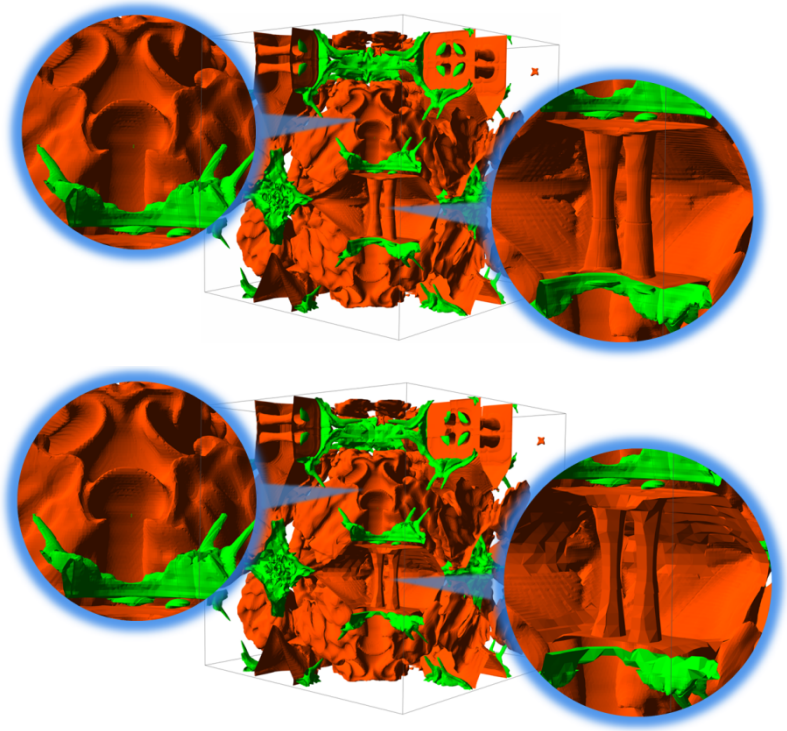
- Application layer: Adaptive spatial-temporal data resolution
- Middleware layer: Dynamic in-situ/in-transit placement and scheduling
- Resource layer: Dynamic allocation of in-transit resources
- Coordinated approaches: Combine mechanisms towards a specific objective (e.g. minimized time-to-solution)



Cross-Layer Adaptation for Dynamic Data Management



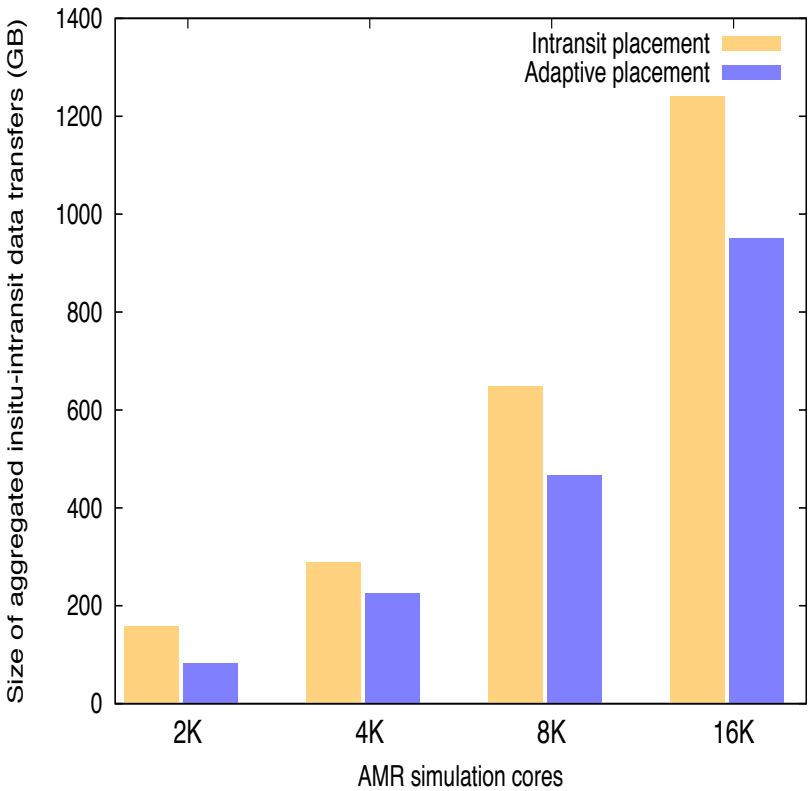
Application layer adaptation of the spatial resolution of data using user-defined down-sampling based on runtime memory availability.

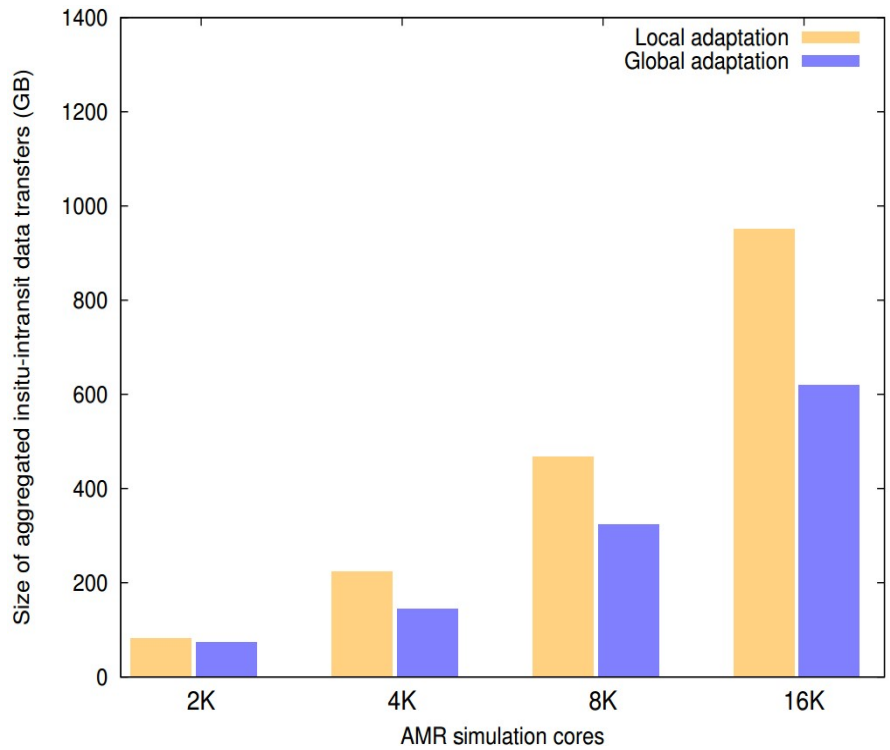


Application layer adaptation of the spatial resolution of the data using entropy based data down-sampling.

(Top: full-resolution; Bottom: adaptive resolution)

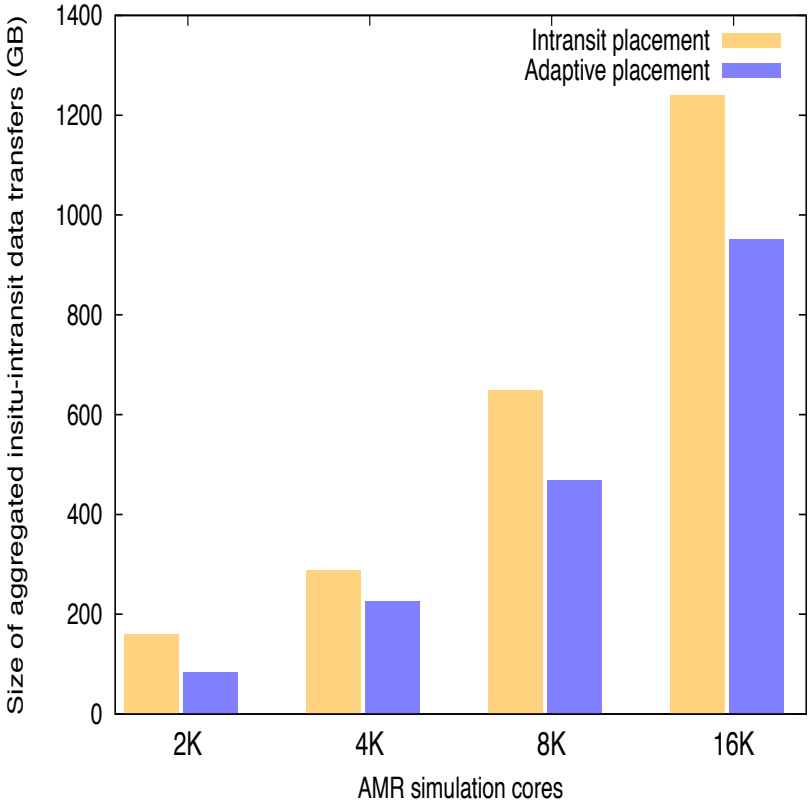
Cross-Layer Adaptation for Dynamic Data Management





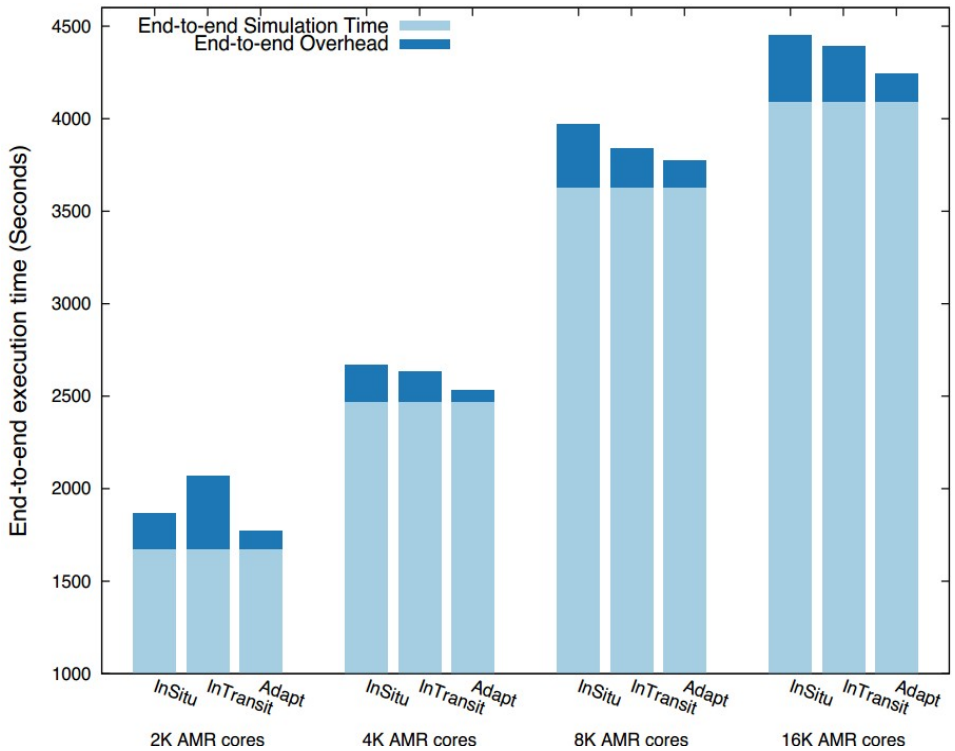
Amount of data transfer using static placement, adaptive placement and combined adaption (adaptive data resolution + adaptive placement)

Cross-Layer Adaptation for Dynamic Data Management



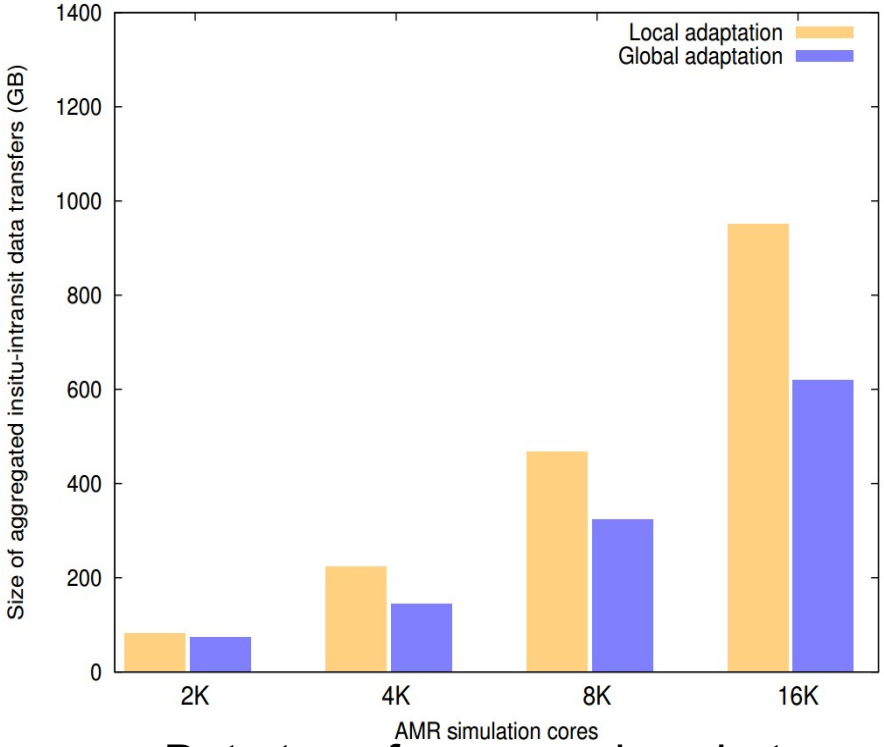
a. Data transfer with/without middleware adaptation at different scales

Adapt the placement on-the-fly, utilizing the flexibility of in-situ (less data movement).

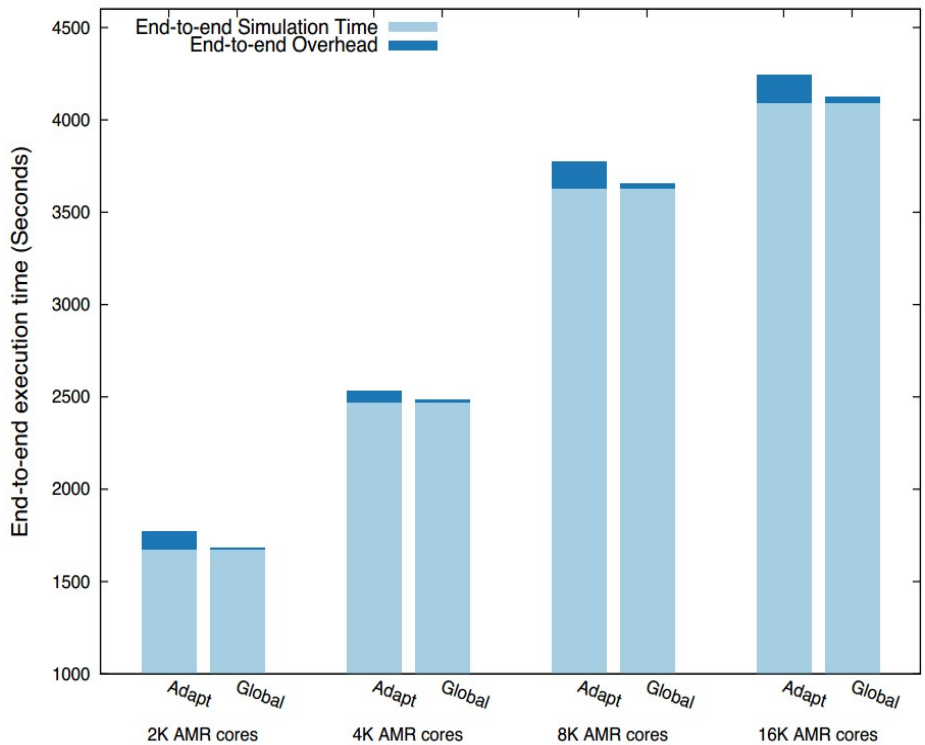


b. Comparison of cumulative end-to-end execution time between static placement (in-situ/in-transit) and adaptive placement. End-to-end overhead includes data processing time, data transfer time, and other system overhead.

Cross-Layer Adaptation for Dynamic Data Management



a. Data transfer comparison between performing adaptive placement and performing combined cross-layer adaption (adaptive data resolution + adaptive placement)



b. Comparison of cumulative end-to-end execution time between adaptive placement and combined cross-layer adaption (adaptive data resolution + adaptive placement).

Scalable In-Memory Data Indexing and Querying for Scientific Simulation Workflows

Motivation

- Query-driven data analysis is an important technique for analyzing the enormous amounts of data produced by large-scale scientific simulations
  - Flame front tracking in combustion simulations: scientists need to formulate a set of queries to discover the data points whose values lie within a certain range

Problems of traditional file-based approach

- Parallel I/O operations become the dominating cost factor, and introduces significant overhead to both index building and query processing
- Only data from selected steps is written for post-processing, thus some highly intermittent and transient phenomena could be lost

Goal

- Enable parallel in-memory indexing and querying to support online query-driven data analysis for large scale scientific simulations

RUTGERS

Scalable In-Memory Data Indexing and Querying for Scientific Simulation Workflows

Highlight

- Parallel in-memory indexing and querying on dedicated staging nodes
- Scalable and efficient online indexing/query performance
- Support SQL-like query syntax and simple querying APIs
- Flexible framework that can be integrated with different index techniques (current implementation uses FastBit – compressed bitmap index)

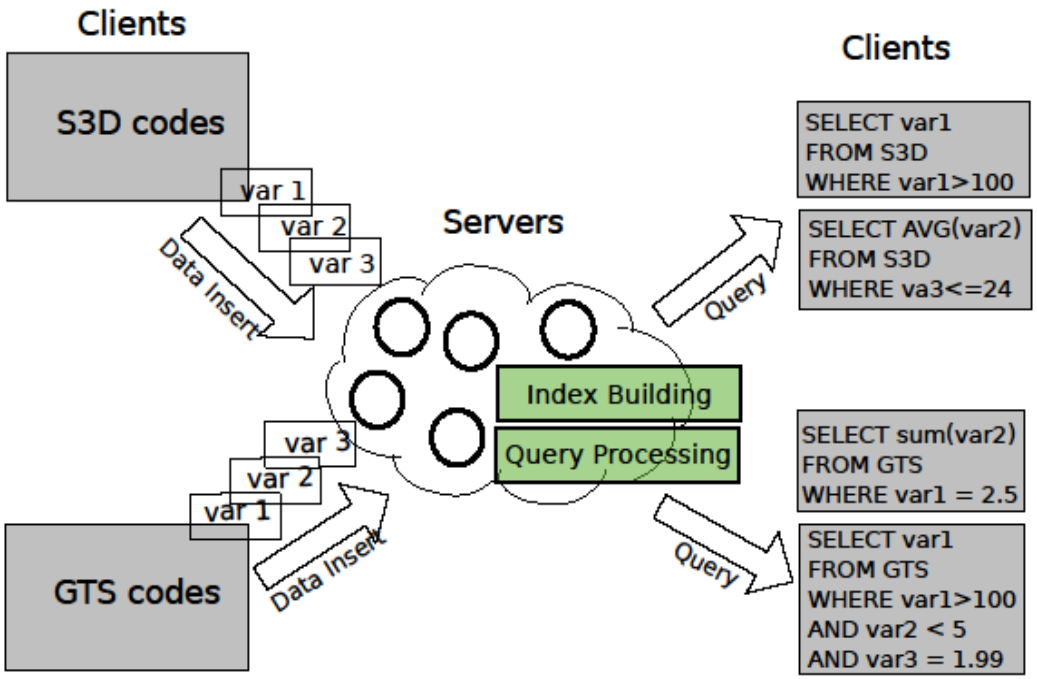


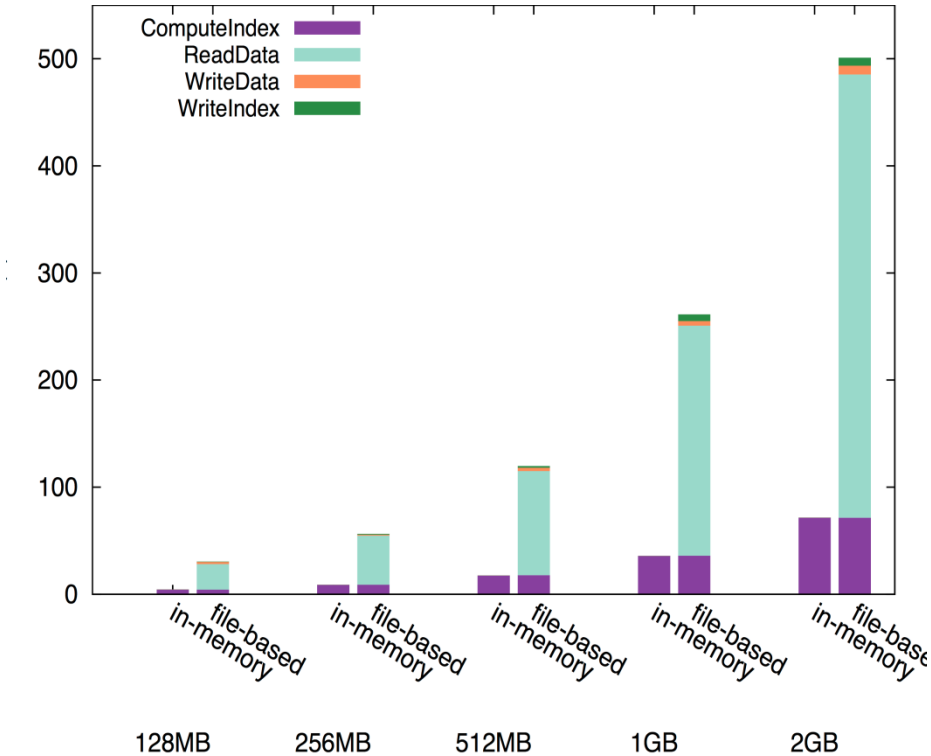
Figure. Conceptual overview of the presented framework, and interaction between scientific simulations and querying applications

RUTGERS

Scalable In-Memory Data Indexing and Querying for Scientific Simulation Workflows

Performance comparison with file-based approach

- I/O overhead is the dominating cost for file-based approach
- DataSpaces approach is significantly faster for both index building and querying



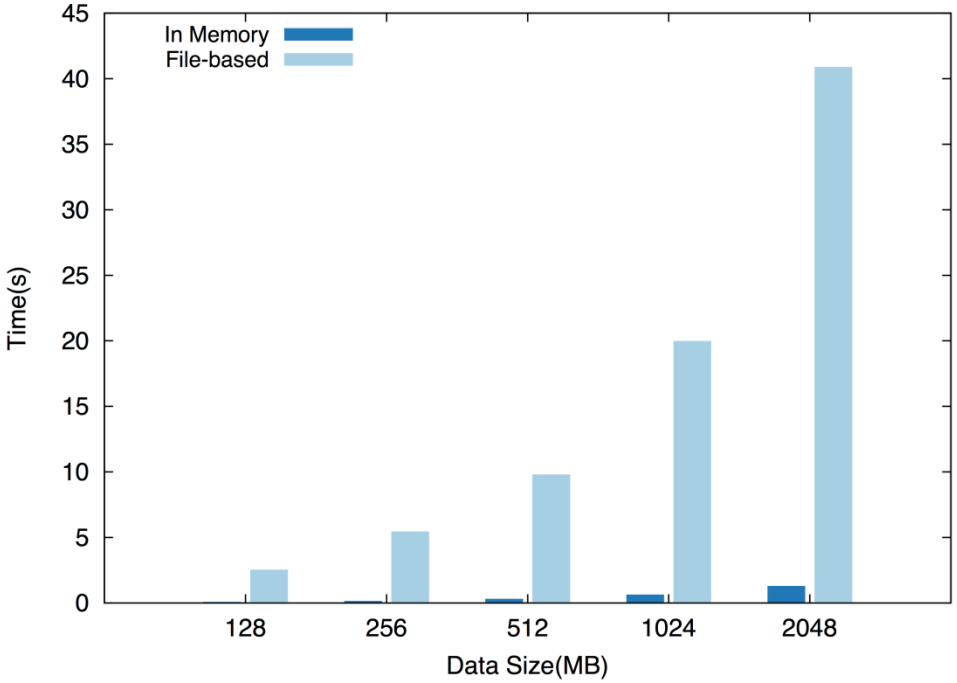


Figure. (Left) Breakdown of index building time; (Right) Query processing time

RUTGERS

Scalable In-Memory Data Indexing and Querying for Scientific Simulation Workflows

Performance with increasing number of cores

- Increase the number of index/query servers from 32 to 1k
- The index building time reduces from 30.7s to 2.1s for 128GB data set.
- The query processing time decreases significantly for different selectivities.

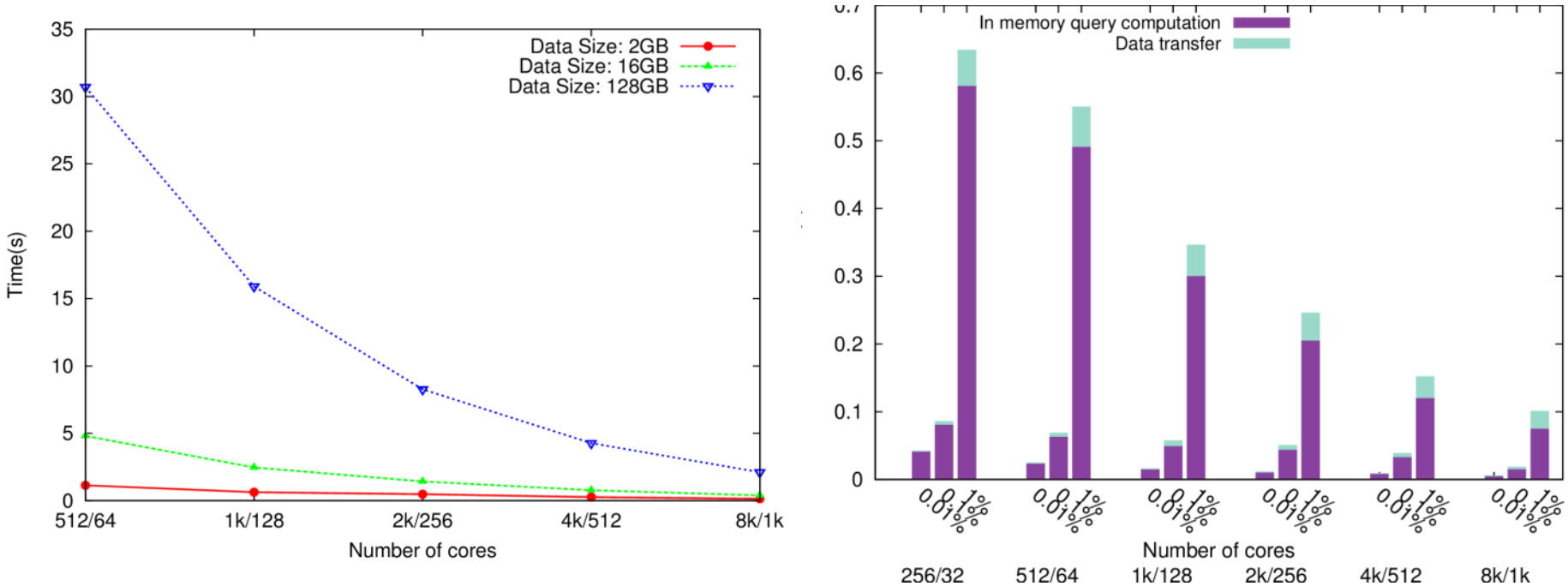


Figure. (Left) Index building time for different data size; (Right) Breakdown of query processing time for different query selectivities

RUTGERS

Scalable In-Memory Data Indexing and Querying for Scientific Simulation Workflows

Scaling Performance

- Increase both the total data size and number of index/query servers
- Our approach shows overall good scalability for both index building and query processing

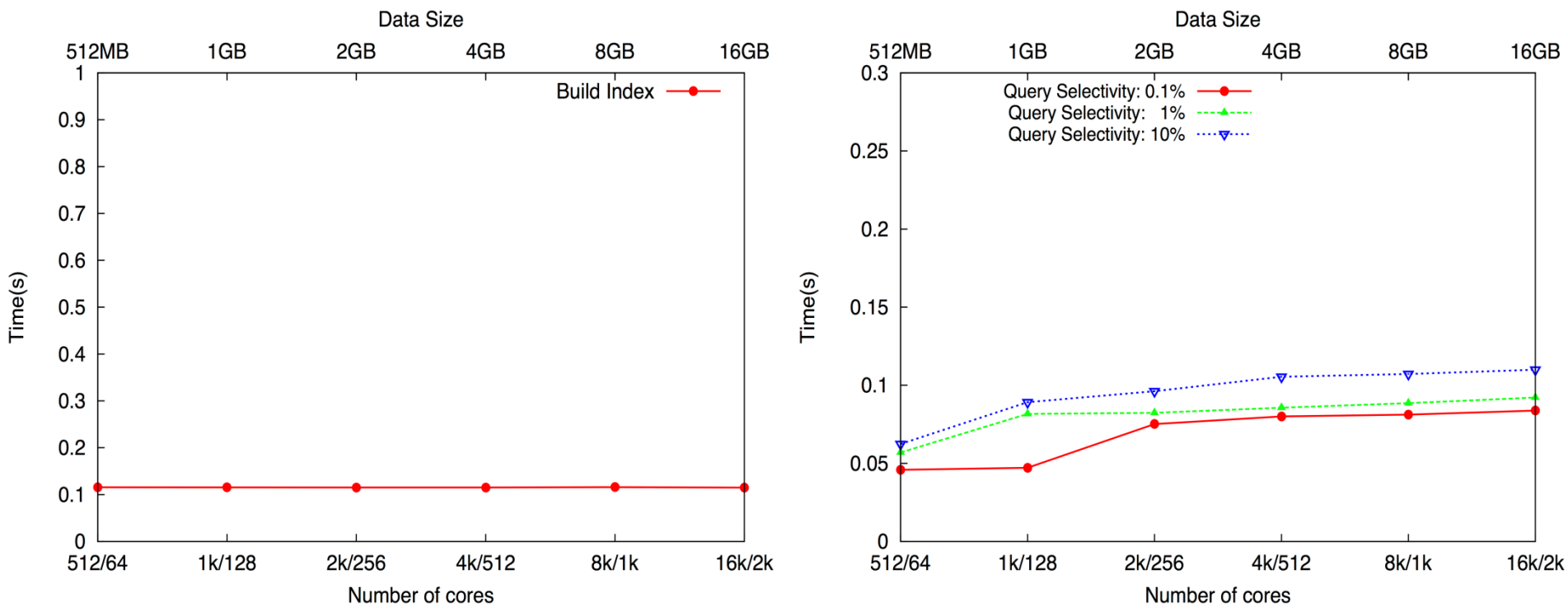


Figure. (Left) Index building time; (Right) Query processing time

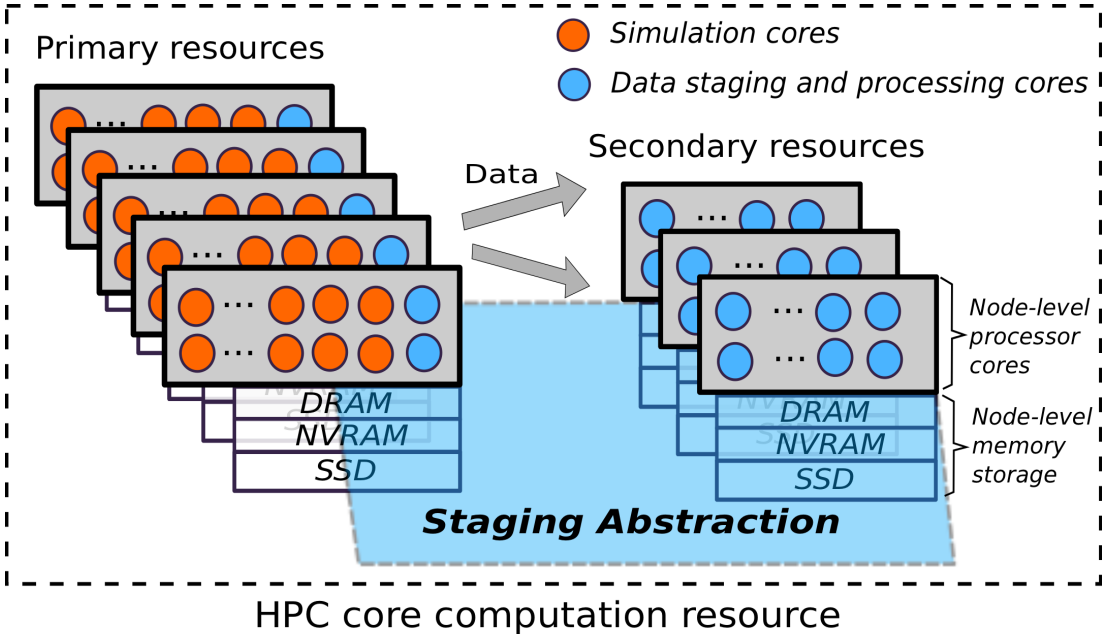
Data Staging over Deep Memory Hierarchy

Motivation

- Given small DRAM capacity per core, even aggregated memory on dedicated nodes will not be sufficient for staging data

Hybrid Staging

- Spans horizontally across the memory of compute nodes of both primary and secondary resources
- Spans vertically across the multi-level memory hierarchy, e.g., DRAM/ NVRAM/SSD, to extend the capacity of in-memory data staging



EPSI @ Rutgers -- Objective

- Enable tightly coupled XGC1 and XGCa workflow using hybrid staging -- XGC1 and XGCa processes on the same compute node, and exchange data through on-node memory
  - Explore two different execution model
    - Concurrent coupling: execute XGC1 and XGCa concurrently on different set of processor cores of each compute node
    - Sequential coupling: execute XGC1 and XGCa sequentially on the same set of processor cores of each compute node

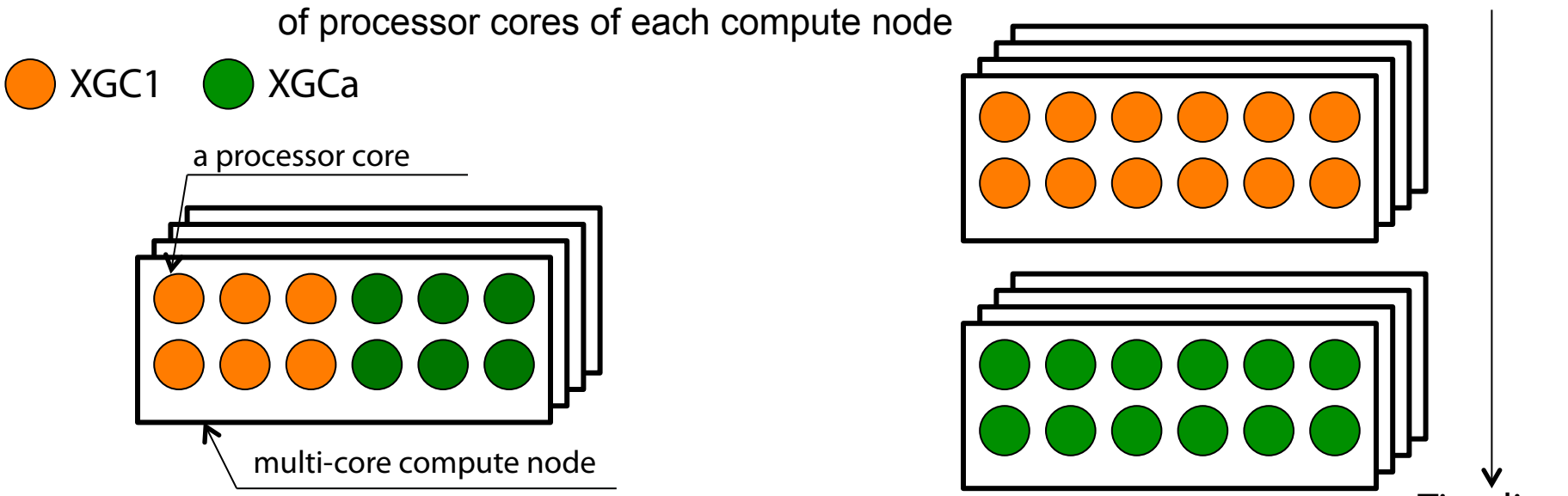
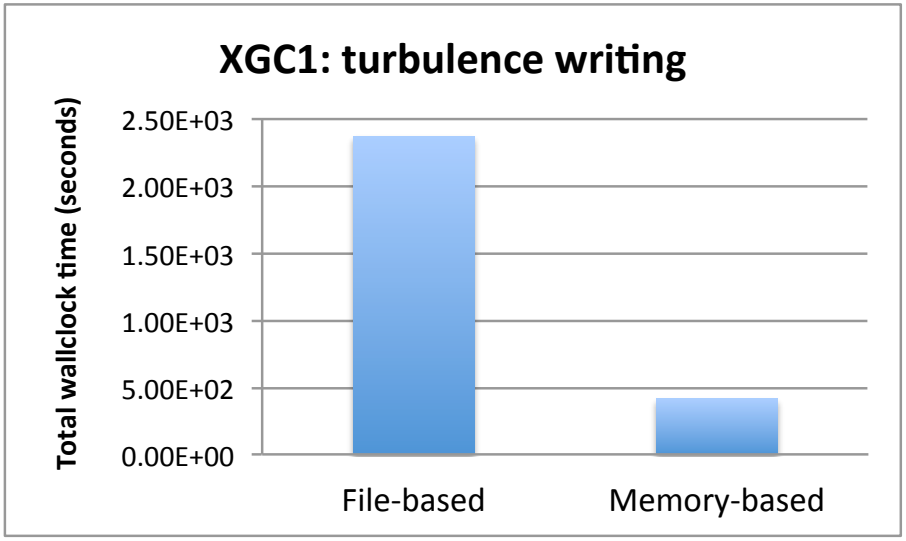
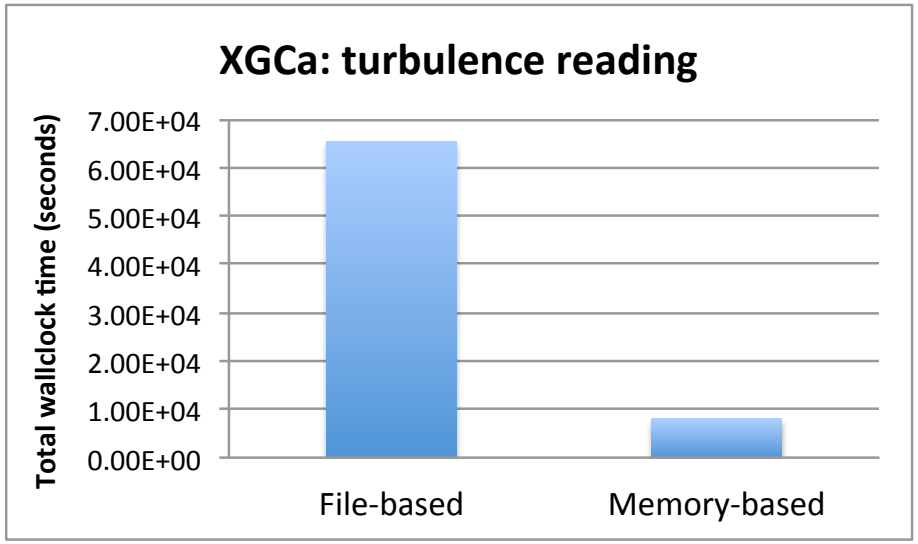


Illustration of tightly coupled execution of XGC1 and XGCa: (Left) concurrent coupling; (Right) sequential coupling

EPSI XGC1-XGCa Coupled Workflow

- Preliminary evaluation results on ORNL Titan
  - File-based: ADIOS/BP method, Memory-based: ADIOS/DataSpaces





Comparison between file-based and memory-based coupling. (Left) XGC1 turbulence writing; (Right) XGCa turbulence reading. Note: Y-axis is total wallclock time (seconds), which is the accumulated sum over all processes and all time steps.

Summary & Conclusions

- Complex applications running on high-end systems generate extreme amounts of data that must be managed and analyzed to get insights
  - Data costs (performance, latency, energy) are quickly dominating
  - Traditional data management/analytics pipelines are breaking down
- ***Hybrid data staging, In-situ workflow execution***, & ***Dynamic code deployment*** can address this challenges
  - Users to efficiently intertwine applications, libraries, middleware for complex analytics
- Many challenges; Programming, mapping and scheduling, control and data flow, autonomic runtime management….
- The ADIOS/DataSpaces project explores solutions at various levels:
  - High-level programming abstractions for in-situ / in-transit workflows for code coupling, online analytics, UQ, etc.
  - Efficient adaptive runtime mechanisms for hybrid staging, locality-aware mapping and location-aware data movement, performance/energy tradeoffs
  - Support for dynamic code deployment and execution for moving code to data

Thank You!

EPSi
Edge Physics Simulation

Green HPC

SDAV
Scalable Data Management, Analysis, and Visualization

ExaCT

ADIOS

SciDAC
Scientific Discovery through Advanced Computing

RDAV

CPES
Center for Plasma Edge Simulation

Manish Parashar, Ph.D.
Prof., Dept. of Electrical & Computer Engr.
Rutgers Discovery Informatics Institute (RDI^2)
Cloud & Autonomic Computing Center (CAC)
Rutgers, The State University of New Jersey

Email: parashar@rutgers.edu
WWW: rdi2.rutgers.edu
WWW: dataspaces.org

TASSL The Applied Software Systems Laboratory

2013 R&D 100

CAC Cloud and Autonomic Computing Center

OLCF Oak Ridge Leadership Computing Facility

Oak Ridge National Laboratory