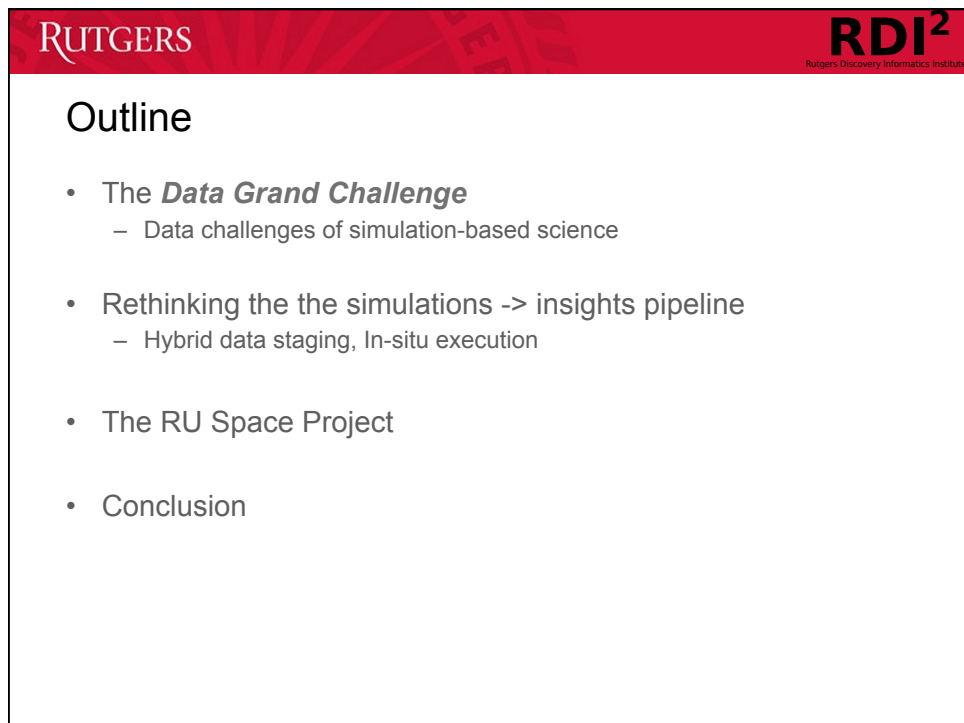





**RUTGERS**  
THE STATE UNIVERSITY  
OF NEW JERSEY

**Rutgers Discovery  
Informatics Institute  
(RDI<sup>2</sup>)**  
New Jersey's Center for Advanced  
Computation

**From Data to Insights - Data  
Management Challenges at Extreme  
Scales**

**Manish Parashar**  
Director, Rutgers Discovery Informatics Institute (RDI<sup>2</sup>)  
Professor, Department of Electrical & Computer Engineering



**RUTGERS**


**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

**Outline**

- The **Data Grand Challenge**
  - Data challenges of simulation-based science
- Rethinking the the simulations -> insights pipeline
  - Hybrid data staging, In-situ execution
- The RU Space Project
- Conclusion

**RUTGERS** **COIC**

## Rutgers Discovery Informatics Institute: RDI<sup>2</sup>



- Fundamentally integrated research, education, ACI and industry partnerships to address core CDS&E / BigData challenges
- Broaden access to state-of-the-art computing technology; integrate multidisciplinary research with ACI and industry partnerships
- Enable large-scale data analytics, computational modeling, and simulations, all of which are playing an increasingly important role in both academic and commercial research and innovation.
- Only university-based advanced computation center in NJ, and one of about ten in US, with an industry partnership program

InformationWeek Hardware nj.com CAMPUS TECHNOLOGY Scientific Computing DatacenterDynamics

June 10, 2012

THE WALL STREET JOURNAL. HUFF POST TECH THE INTERNET NEWSPAPER: NEWS BLOGS VID

Rutgers University, IBM Open Supercomputer Center

Rutgers gets IBM supercomputer to power collaborations with industry

Named "IBM Blue Gene/P," the machine, about the size of two refrigerators, will be one of the most powerful computers in the Northeast, with thousands of central processing units, or CPUs. IBM hopes in the coming year it will make the prestigious TOP 500<sup>®</sup> list of the world's most powerful computers, determined by a group of academic...

HPC Rutgers to Become Big Data Powerhouse

7 supercomputers changing the world

Today's machines are taking on challenges ranging from cancer research to 'Jeopardy' champions.

EXCALIBUR R&D datanami

NJBIZ Building a Smarter Planet Rutgers, IBM to Build HPC Center focused on Big Data Analytics

EXEC ID Ameritrade IBM BlueGene supercomputer to help New Jersey industry

Supercomputer to Power Industry, Rutgers Research

High Performance Computing: The New Imperative is Economic Development and Jobs

Rutgers Gets Big Data Weapon In IBM Supercomputer

Rutgers makes overture to business as it unveils supercomputer

Plugged in to corporate partners

Rutgers Teams With IBM to Build Powerful High-Performance Computing Center in New Jersey

Rutgers' IBM Blue Gene P computer to improve research

Rutgers University in New Jersey recently initiated a High-Performance Computing (HPC) center, powered by an IBM Blue Gene/P supercomputer. Hopes are that the HPC center will become one of the world's most powerful academic supercomputers. The university reports that the goal of this project is to improve research across a multitude of areas, including cancer and genetic research, medical imaging and informatics, advanced manufacturing, environmental and climate research and materials science.



**RUTGERS** **COIC**

## Key Programmatic Areas

### Research

- Provide Rutgers researchers access to computational resources and technical expertise necessary to increase accuracy and scale of their research
- Promote interdisciplinary collaborations to increase grant competitiveness

### Advanced Computing Infrastructure

- Data and compute-centric capabilities
- Experimental platforms
- Expertise

### Education and Training


- Variety of education and training programs for faculty, students and industry
- Masters degrees, certificates, technical modules, industry-specific workshops

### Industry Engagement and Economic Development

- RDI<sup>2</sup>'s Industry Partnership Program will assist private firms in overcoming the cost and knowledge barriers associated with advanced computation
- RDI<sup>2</sup> will promote economic development by attracting new firms to New Jersey and encouraging existing firms to stay in-state

**RUTGERS** **COIC**

## RDI<sup>2</sup> Advanced Computing Infrastructure (Phase I)



**“Excalibur,” an IBM Blue Gene®/P Supercomputer**




**2,000 Nodes, 8,000 Cores, 24 Terabytes of RAM, 300 – 400 Terabytes of memory**

- Future Plans:
  - Phase II - upgrade Blue Gene/P to IBM’s newest Blue Gene model, the Blue Gene/Q
  - Acquisition of a 10 – 12 petabyte storage container with co-located analytics
  - Connectivity to national cyberinfrastructure
- Goal by Phase III is to have one of the top academic supercomputers in the world

**RUTGERS** **CAC**

## RDI<sup>2</sup>' s Industry Partnership Programs


**NSF Cloud and Autonomic Computing (CAC) Center**  
**NSF Center for Dynamic Data Analytics (CDDA)**

-  CAC/CDDA are multidisciplinary NSF centers of excellence in cloud and autonomic computing research
-  Foster long-term collaborative partnerships among industry, academia, and government
-  Have a well-established Industry Partnership Program and many industry partners

**RUTGERS** **RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## Modern Science & Society Transformed by Data

- ❖ **Modern science**
  - ❖ Data- and compute-intensive
  - ❖ Integrative, multiscale, online
  - ❖ 4 centuries of constancy, 4 decades 10<sup>9-12</sup> change!
- ❖ **Multi-disciplinary/scale collaborations**
  - ❖ Individuals, groups, teams, communities, networks
  - ❖ New scientific culture
- ❖ **Sea of Data**
  - ❖ Heroic Age of Digital Observation
- ❖ **Complexity, complexity, complexity!**
  - ❖ Impeding science
  - ❖ Productivity, reproducibility, etc.

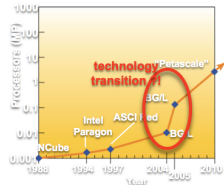
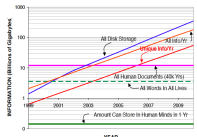
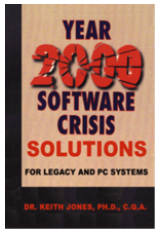


SCIENCE IN THE PETABYTE ERA  
 Ack. E. Seidel

**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## Many Challenges

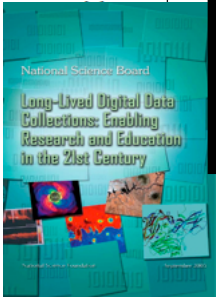
- **Computing**
  - Multicore; large and increasing core counts, deep memory hierarchies
  - New prgm. model, concerns (fault tolerance, energy, etc)
  - New models & technologies: Clouds, grids, hybrid manycore, accelerators, deep storage hierarchies, ...
- **Data**
  - Generating more data than in 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<sup>6</sup>+ lines, many groups contribute, take decades
- **People**
  - Multidisciplinary expertise essential!
  - Appropriate academic program, career tracks...

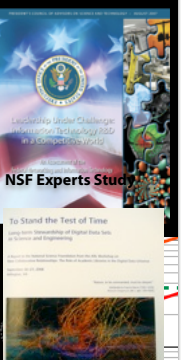
**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## Data Crisis: Information Big Bang

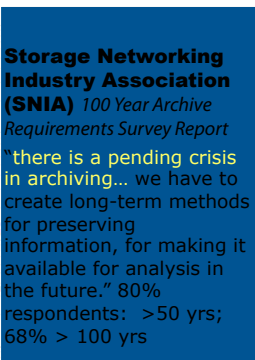
**NSB Report: Long-Lived Digital Data Collections Enabling Research and Education in the 21st Century**



**PCAST Digital Data**





**Industry**



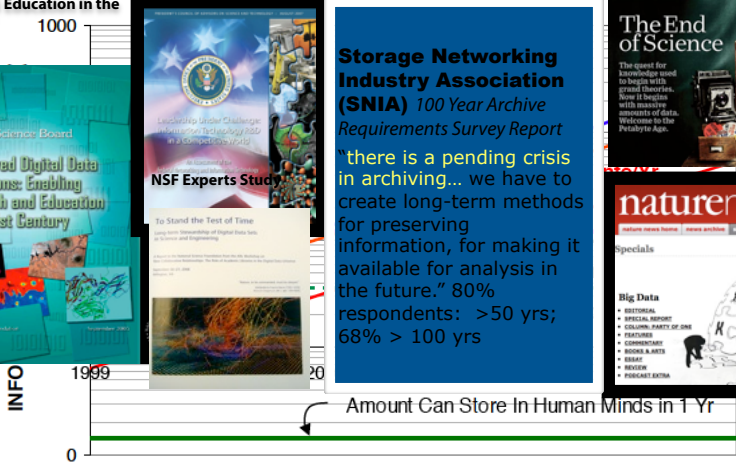
**Storage Networking Industry Association (SNIA) 100 Year Archive Requirements Survey Report**


"there is a pending crisis in archiving... we have to create long-term methods for preserving information, for making it available for analysis in the future." 80% respondents: >50 yrs; 68% > 100 yrs

**Wired, Nature**

Amount Can Store In Human Minds in 1 Yr





Sources: Lesk, Berkeley SIMS, Landauer, EMC

**YEAR**

"Data generation == 4 x Moore's Law"

**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## We know that modern network/ instruments/experiments/... are producing Big Data!!

Large Hadron Collider

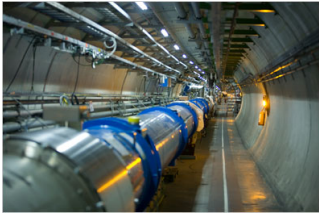


Image credit: Valerio Mezzanotti for The New York Times

Blanco 4m on Cerro Tololo

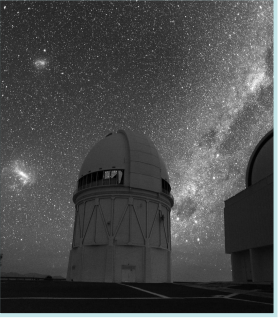
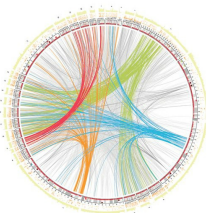
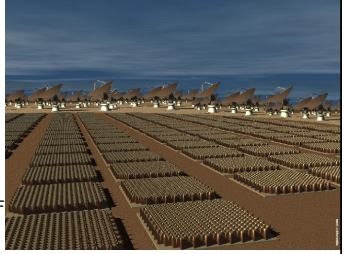


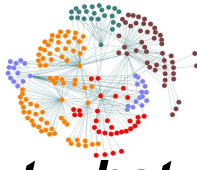
Image credit: Roger Smith/NOAO/AURA/NSF



SKA project



Above is proposed image




## But what about traditional HPC?!

**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute


## Advanced Computing Infrastructure

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




**Titan - Cray XK7**

- 27 PF / 56 K cores
- 16-core CPU + GPU
- Gemini 3D torus
- 710TB memory




**XSEDE**

- Worlds Largest Grid
- 11 Resource Providers



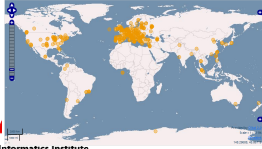
**Sequoia - IBM BG/Q**

- 20 PF / 1.6 M cores
- 18-core processor
- 5D torus
- 1.5PB memory



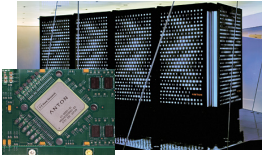
**Modern Datacenters**

- 1M servers
- 50-100 MW



**Worldwide LHC Computing Grid**

- >140 sites;
- ~250k cores;
- ~100 PB disk



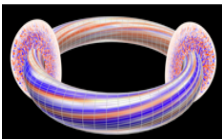
**Special Purpose HW (Anton)**

- > 100 time acceleration of MD simulations


**RUTGERS** **RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## Scientific Discovery through Simulations - II

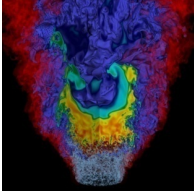
- Scientific simulations running on high-end computing systems generate huge amounts of data!



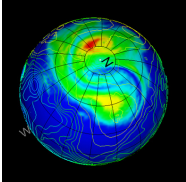
Plasma Fusion



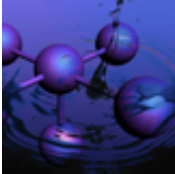
Astrophysics




Combustion



Climate Modeling



Molecular Simulation

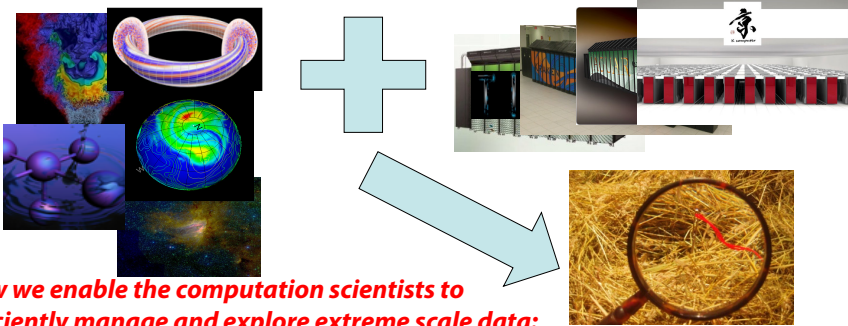


- The volume of simulation data being produced is enormous and continuous, and keeps growing every year!
- Costs involved are huge – systems, operation, scientist efforts, ...

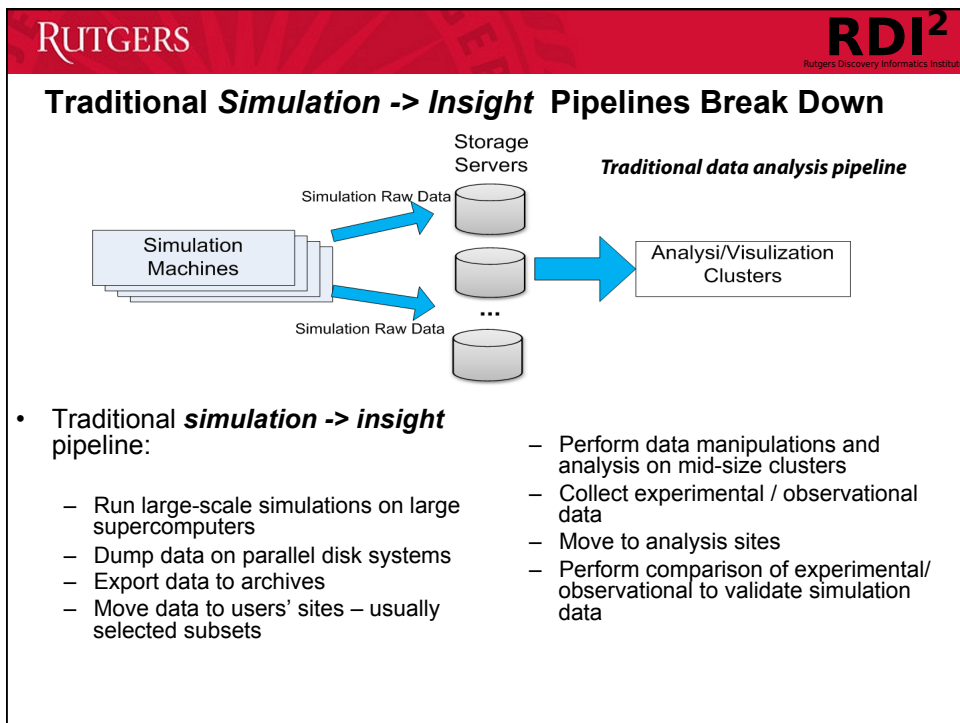
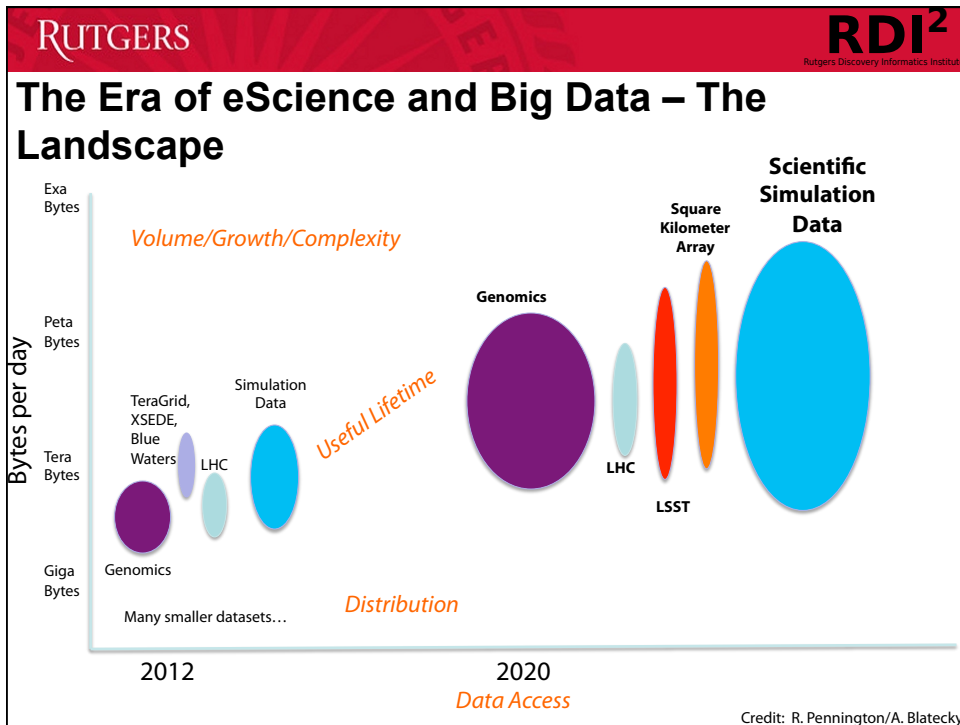
**RUTGERS** **RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## Scientific Discovery through Simulations

- Scientific simulations running on high-end computing systems generate huge amounts of data!
- 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" ??***





**RUTGERS** **RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## 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?
- **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 multi-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!

*Traditional data analysis pipeline*

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

**RUTGERS** **RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## The Cost of Data Movement (I)

- Moving large amount of simulation data between node memory and system wide persistent storage is slow!

CPU-I/O speed disparity

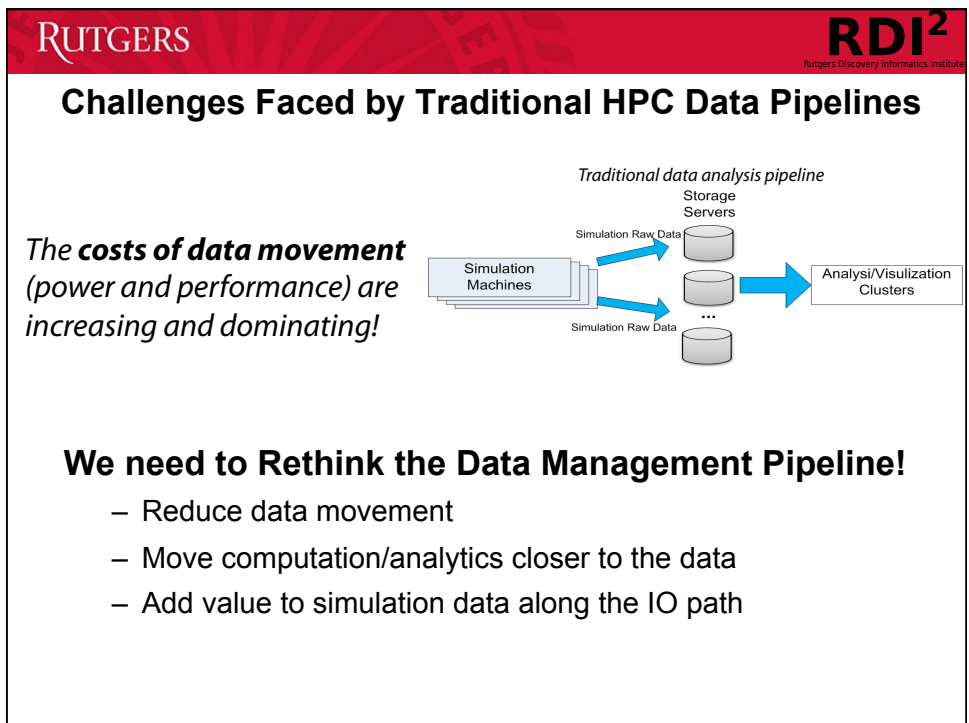
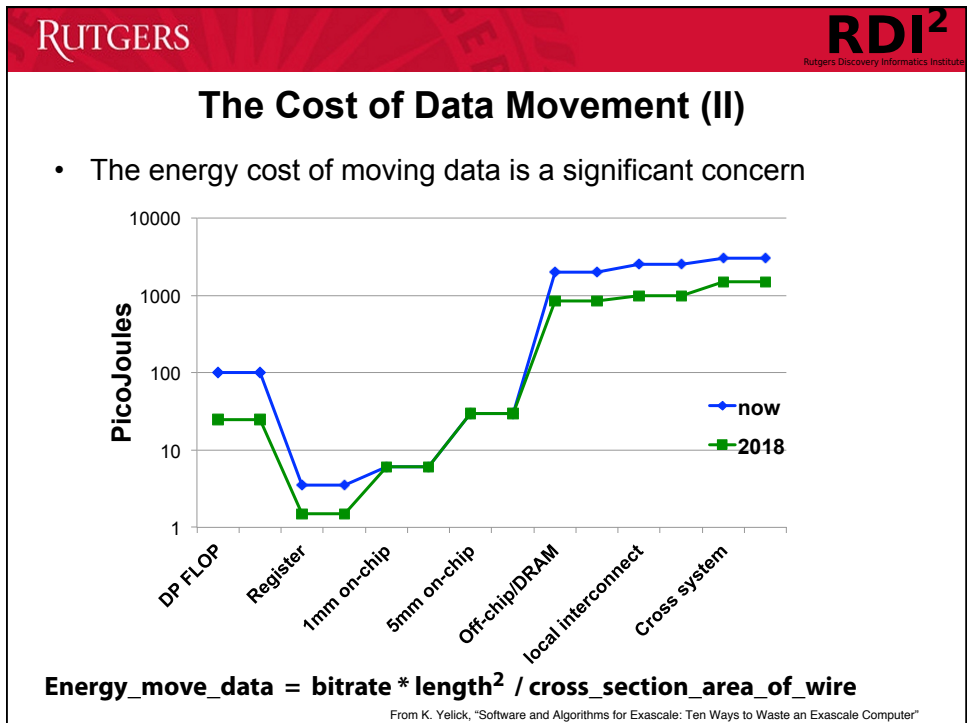
Year

Hard disk seek time —+—  
 CPU cycle time —x—  
 DRAM cycle time —\*—

Access Latency (s)

performance gap

Increasing Latency ↓



**Challenges amplified as we move to Exascale**

**Performing the simulation is not enough – need to analyze results**

- Storage space requirements
  - 35 disks for each dump (No RAID)
  - 1.5 KW/live dump
- Performance requirements
  - 5% overhead, ~1M disks, >40 MW
  - 10% overhead, ~500K disks, >20 MW
  - 50% overhead, ~100K disks, >4 MW
- I/O bandwidth constraint make it infeasible to save all raw simulation data to persistent storage
  - **In situ and in-transit analyses are a necessity**

**Rethinking the Data Management Pipeline - I**

- Objectives
  - Reduce data movement
  - Move computation/analytics closer to the data
  - Add value to simulation data along the IO path
- Use distributed, in-memory **Hybrid Data Staging**, constructed combining application node cores and dedicated staging nodes, to enable customized in-situ/in-transit processing on staged data
- Active Data Management @ **Hybrid Data Staging**
  - **In-situ Computation/Analytics**: move data processing operations to where the simulation data is being generated
  - **In-transit Data Manipulation**: transform/make-right the data as it moves from source to sink
  - **In-situ Coupled Simulation Workflows**: execute interacting scientific applications in-situ on multi-core architecture to increase intra-node data exchanges
  - **Dynamic Binary Code Deployment**: dynamically deploy compiled binary code and execute it within the staging area

**RUTGERS** **RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

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

- Exploit multi-levels in-memory **Hybrid Data Staging** to:
  - Decrease the gap between CPU and IO speeds
  - Dynamically deploy and execute data analytical or pre-processing operations either in-situ or in-transit
  - Improved IO write performance

## 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  $n^{\text{th}}$  simulation time step
  - Execute asynchronously

**Analysis Tasks**

- Analysis Tasks
- Simulation
- Visualization

**EXACT** CENTER FOR EXASCALE SIMULATION OF COMBUSTION IN TURBULENCE

RUTGERS
RDI<sup>2</sup>  
Rutgers Discovery Informatics Institute

## Programming Data Staging Resources

Goal: Effective use of staging resources for flexible in-memory, in-situ processing

- Application-application interactions
  - Code coupling, MxN data redistribution, data transformations
- Querying interfaces
- Application workflows/pipelines
- Analytic plugins/filters
  - Dynamic deployment

- Programming model
  - PGAS, Workflow, Database, etc.
- Abstractions provided
  - Data and control models
- Runtime mechanisms
  - Mapping (locality, heterogeneity), scheduling, etc.

RUTGERS
RDI<sup>2</sup>  
Rutgers Discovery Informatics Institute

## In-Situ/In-Transit Workflows

### Issues/Challenges

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

Computational  
runners

Managers

**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## In-Situ/In-Transit Workflows – Mapping and Scheduling

Mapping: Fully in-situ v/s fully in-transit v/s hybrid

- Task characteristics
  - Parallelization, runtime, memory footprint, communication requirements, input-output characteristics, data sizes etc.
- Heterogeneous capabilities, costs
  - In-situ cores v/s In-transit cores
    - Cores, memory, comm., etc.
- Data locality v/s data movement
- Dataflow
  - Where are the inputs produced, how will the outputs be used?
- Resources state, usage pattern, ...

Scheduling:

- Impact on overall execution, impact end-to-end process?

**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## Third-Party Plugins in the Staging Area

- Data processing plugins in the hybrid staging area
  - In-situ data processing
  - Analytics pipelines
- Many issues
  - Programming (data and control) models for plugins
  - Deployment mechanisms
  - Robustness, correctness, etc.
- Multiple approaches
  - Code, binary, scripts, etc.
  - Several implementations
    - ActiveSpace, SmartTap, etc.
  - E.g., ActiveSpaces (IPDPS 11): Dynamically deploy custom application data transformation/filters on-demand and execute in staging area (DataSpaces)

data\_kernels.o

0x69 0x20 0x61  
0x6d 0x20 0x63  
0x6f 0x6f 0x6c  
0x0

Link

Applications.o

Applications executable

Runtime execution system (Flexec) Staging nodes

gcc -c

data\_kernels.c

```
kernel_min {
  for i = 1, n
    for j = 1, m
      for k = 1, p
        if (min > A(i, j, k))
          min = A(i, j, k)
}
```

Computes nodes

return()

return()

- Provide the programming support to define custom data kernels to operate on data objects of interest
- Runtime system to dynamically deploy kernels to staging resources, and execute them on the relevant data objects

**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## In-Situ Data Management & Analytics @ RU

Shared Space Programming API (data put/get)

<b>DataSpace/ XpressSpace</b>	<b>ActiveSpace</b>	<b>In-Situ coupled workflow execution framework</b>
-----------------------------------	--------------------	---

Distributed In-memory Hybrid Data Staging  
(in-transit or in-situ)

HybridDART

DART Asynchronous Data Transport

Communication Networks  
(Cray Gemini, Cray Portals, Infiniband, IBM DCMF, TCP/IP)

- Virtual Shared-space programming abstraction
- Simple API to *insert* and *retrieve* data
- Online indexing, storage, flexible querying
- In-memory distributed storage
- Efficient asynchronous data transfer

### ADIOS

- DART:** a network independent transport library for high speed asynchronous data extraction and transfer [HPDC08]
- DataSpaces/XpressSpace:** an interaction and coordination framework for memory-to-memory data coupling [HPDC10, CCGrid10,11]
- ActiveSpace:** dynamic deployment and execution of data processing routines on the in-memory staging data [IPDPS11]
- In-situ Execution** of workflows: reduce data movement and increase intra-node data sharing [IPDPS12]

**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## DataSpaces: A Shared Space Abstraction in the Hybrid Staging Area [HPDC'10]

- Semantically-specialized* virtual shared space abstraction in the staging area
  - Shared (distributed) data object
  - Simple put/get/query API
  - Supports application – application, workflows
  - Provide a global-view programming abstraction consistent with the PGAS model (UPC, GA)
- DataSpaces** (HPDC10)
  - Constructed on-the-fly on hybrid staging nodes
    - Indexes data for quick access and retrieval
    - Provides asynchronous coordination and interaction support
  - Complements existing interaction/coordination mechanisms

- Code coupling using DataSpaces
  - Maintain locality for in-situ exchange
  - Complex geometry-based queries
  - In-space (online) data transformation and manipulations

RUTGERS
RDI<sup>2</sup>  
Rutgers Discovery Informatics Institute

## DataSpaces Query Engine: Indexing + DHT

- The global application domain is used to build an overlay and DHT across a dynamic set of DataSpaces servers
  - Use a Hilbert SFC to construct the key space
    - e.g., map multi-dimensional space to a linear space
  - Use the DHT to maintain meta-data information
    - e.g., geometric descriptors for the shared data
- Data objects are indexed with the same SFC and the DHT entries updated

21	-	-	22	25	-	-	26	37	-	-	38	41	-	-	42
20		23	-	-	24		27	36		39	-	-	40		43
19	-	-	18	29	-	-	28	35	-	-	34	45	-	-	44
16	-	-	17	30	-	31	-	32	-	-	33	46	-	-	47
15		12	-	-	11	-	-	10	53	-	-	52	-	-	48
14	-	-	13	8	-	-	9	54	-	-	55	50	-	-	49
1	-	-	2	7	-	-	6	57	-	-	56	61	-	-	62
0		3	-	-	4	-	-	5	58	-	-	59	-	-	63

SFC 1-Dimension Domain

<0 - 17>   <30 - 33>   <52 - 59>

---

<0 - 29>

DHT Keys   Virtual 1-Dimension Domain

<0 - 6>   <7 - 13>   <14 - 20>   <21 - 29>

Node 1   Node 2   Node 3   Node 4

- The SFC maps the global domain to a set of intervals
  - Intervals are non-contiguous and can lead to meta-data load imbalance
  - Second mapping compacts the intervals into a contiguous virtual interval
  - Split the contiguous interval equally to the DataSpaces servers

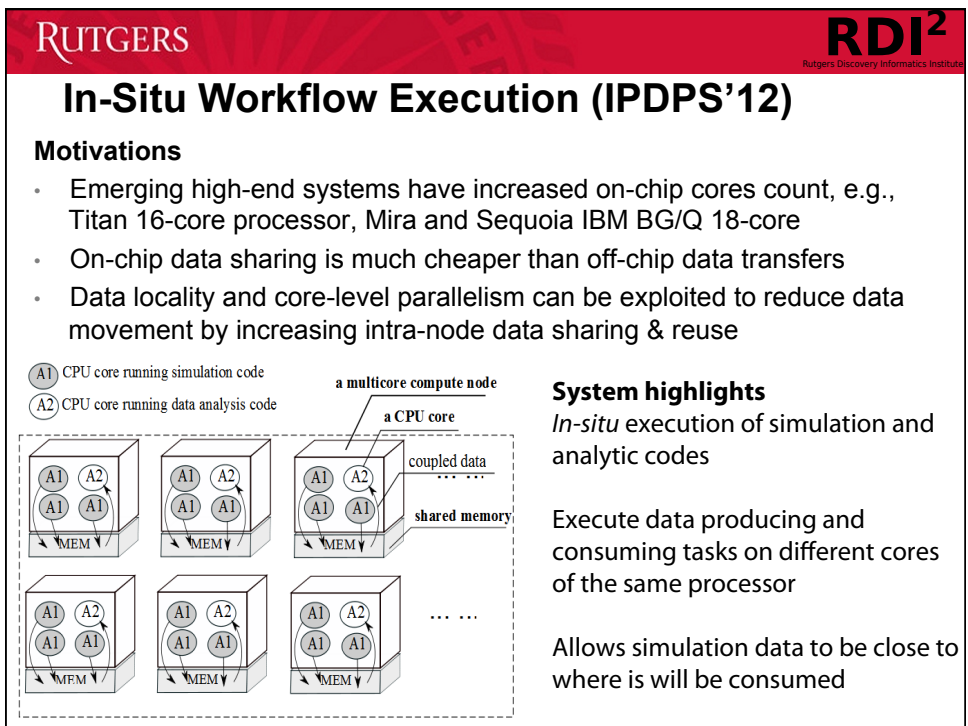
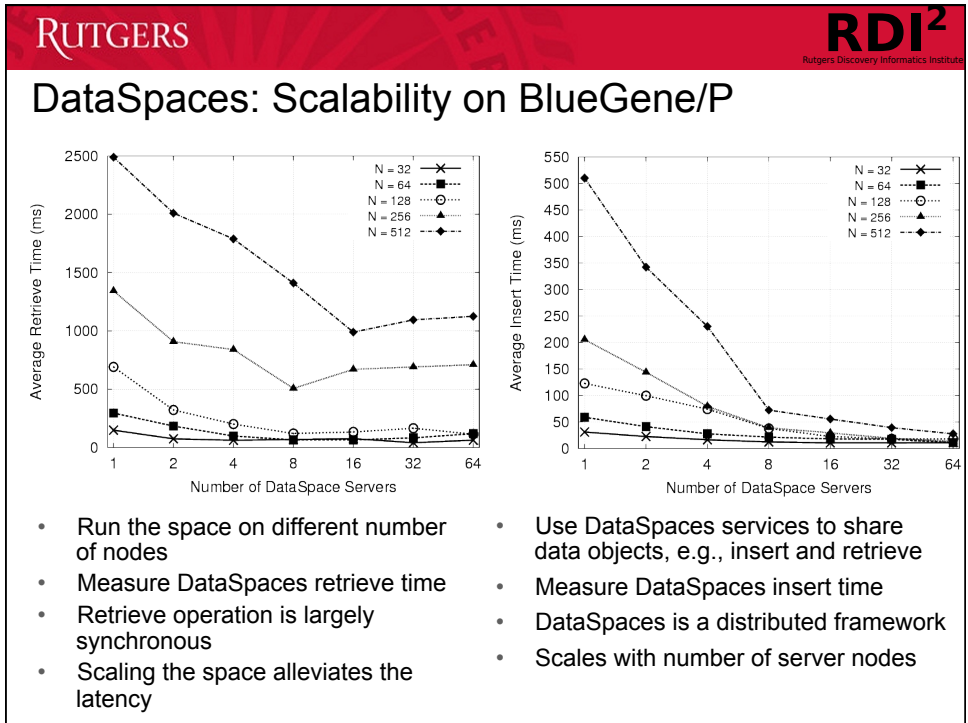
RUTGERS
RDI<sup>2</sup>  
Rutgers Discovery Informatics Institute

## DataSpaces: Scalability on ORNL Jaguar-pf

Processors	Retrieve query (s)	Insert query (s)
8	2.0	1.2
16	2.0	1.2
128	2.8	1.3
256	3.2	1.4
512	2.7	1.5
1024	3.0	1.8

- Evaluate framework scalability with an increasing number of processors
- Use two testing applications that exchange data through DataSpaces
  - Run on M processors and insert data in the space
  - Run on N processors and retrieve data from the space
- Use a weak scaling experiment
  - Amount of data increases with the number of processors
    - Keep the amount of data/processor constant
  - Resembles the behavior of real simulations
- A 128 fold increase in the system size adds 0.5s to the insert time, and 1s to the retrieve time





**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## In-Situ Workflow Execution (I)

**Challenges**

- Locality-aware mapping of computation tasks from separate coupled applications onto processor cores
  - Which CPU core should the task run on? Trade-offs?
- Efficient support for data sharing and exchange between the coupled applications

Scenario 1: Online Data Processing

Scenario 2: Coupled Climate Modeling Simulation

*Two simple in-situ workflow scenarios:*

(1) Online data analysis and

(2) Coupled simulations (climate modeling)

**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## In-Situ Workflow Execution: Evaluation (III)

**Lesson learned:** To meet power budgets, locality-aware mapping/scheduling can be used to reduce data movement significantly, but requires fatter nodes.

Coupled simulation workflow  
(Task1: 512, Task2: 64 cores)

Data analytics pipeline  
(Task1: 512, Task2: 128, Task3: 384 cores)

Pattern	Round-robin	Data-centric
block-block	8	1.5
cyclic-cyclic	8	1.5
block-cyclic	8	7.5
cyclic-block	8	7.5

Pattern	Round-robin	Data-centric
block-block	16	1.5
cyclic-cyclic	16	1.5
block-cyclic	16	14.5
cyclic-block	16	14.5

**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

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

Component	Value
in-situ PVR	0.73
hybrid PVR	5.07
in-situ SSA	1.64
hybrid SSA	1.7
hybrid RTC	2.72 + 2.06 + 119.81
simulation	16

- 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**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## ActiveSpaces: Move Code to the Data [IPDPS'11]

Dynamically deploy binary code on-demand and execute customized data operations (e.g., data transformation/filters) within staging area

Plasma fusion code-coupling scenario: XGC0, M3D-MPP, and auxiliary services for post-processing, diagnostics, visualization

### Advantages

- Reduces network data traffic by transferring only the analytic kernels and retrieving the results
- Reduces application execution time by offloading and executing in parallel data computations
- Kernels defined using native programming language
- Operates only on data of interest

**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## Programming with ActiveSpaces: Data Kernel Execution

```

data_kernels.o
0x69 0x20 0x61
0x6d 0x20 0x63
0x6f 0x6f 0x6c
0x0
            
```

```

Applications.o
            
```

gcc -c → data\_kernels.c

```

kernel_min {
for i = 1, n
for j = 1, m
for k = 1, p
if (min > A(i, j, k))
min = A(i, j, k)
}
            
```

```

Applications executable
            
```

Link

```

Applications.o
            
```

```

abrun
            
```

→

```

Applications executable
            
```

→

```

return()
            
```

Runtime execution system (Rexec) Staging nodes

- Data kernels are user defined and customized for each application
- Implemented using all constructs of the C language
- Compiled and linked with user applications
- Can execute locally, or be deployed and executed remotely in the staging area

**RUTGERS**
**ActiveSpaces: Evaluation-I**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## Data Scaling Experiment

- Use a coupling scenario with two application exchanging data through the space
  - One application inserts data into the space
  - One application retrieves and processes data from the space
- Define custom kernels data filters
  - Transfer data from the space to the application and execute filters locally
  - Transfer the filters to the space, execute on the space and retrieve the result
- The data retrieved and processed was scaled from 1kB to 1GB
- Crossover (sweet) point is interesting:
  - For small data sizes (<= 10kB), it is better to transfer raw data,
  - But for larger sizes (>10kB) is better to offload the kernels

**Code Transfer Performance**

Data size (kB)	code xfer min	code xfer max	code xfer sum	code xfer avg	code xfer CA
1	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001
10	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001
100	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001
1000	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001
10000	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001
100000	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001
1000000	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001

**Saving Performance**

Data size (kB)	saving min	saving max	saving sum	saving avg	saving CA
1	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001
10	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001
100	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001
1000	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001
10000	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001
100000	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001
1000000	~0.0001	~0.0001	~0.0001	~0.0001	~0.0001

**RUTGERS** **ActiveSpace: Evaluation-II**

**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

### Application Performance

- The retrieving application was scaled from 1 to 512 processors
  - ❑ Offloaded interpolation operation to the space (i.e., cylindrical to mesh coordinates)
  - ❑ Sorted particle data in the space
- Time saving of 0.14s per processor -> ~1 hour at application level

Number of processors	code xfer mapping (s)	data xfer pre-mapped (s)	data xfer raw (s)
1	0.01	0.01	0.01
10	0.01	0.01	0.01
100	0.01	0.01	0.01
512	0.01	0.01	20.0

Number of processors	code xfer mapping (s)	saving (s)
1	0.15	0.12
10	0.12	0.05
100	0.15	0.10
512	0.38	0.14

**In-situ viz. and monitoring with staging**

**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

Pixie3D 1024 cores

record.bp

pixie3d.bp

DataSpaces

Pixplot 8 cores

ParaView Server 4 cores

Pixmon 1 core (login node)

ParaView

Pixmon

**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

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

**DOC Overlay**

○ Node network with circular index space; each node manages a part of the index space  
 □ Node index space mapped by SFC to the attribute space  
 ■ Region assigned to node 1 for clustering. The points in this region are mapped to Node 1.  
 ▨ Range in the attribute space, associated with green nodes, as well as node 1

DOC workers executed in-situ on simulation machines

**Simulation Compute Nodes**

● Processor core runs simulation  
 ● Processor core runs DOC worker

One compute node

Benefits of runtime feature extraction and tracking

- (1) Scientists can follow the events of interest (or data of interest)
- (2) Scientists can do real-time monitoring of the running simulations

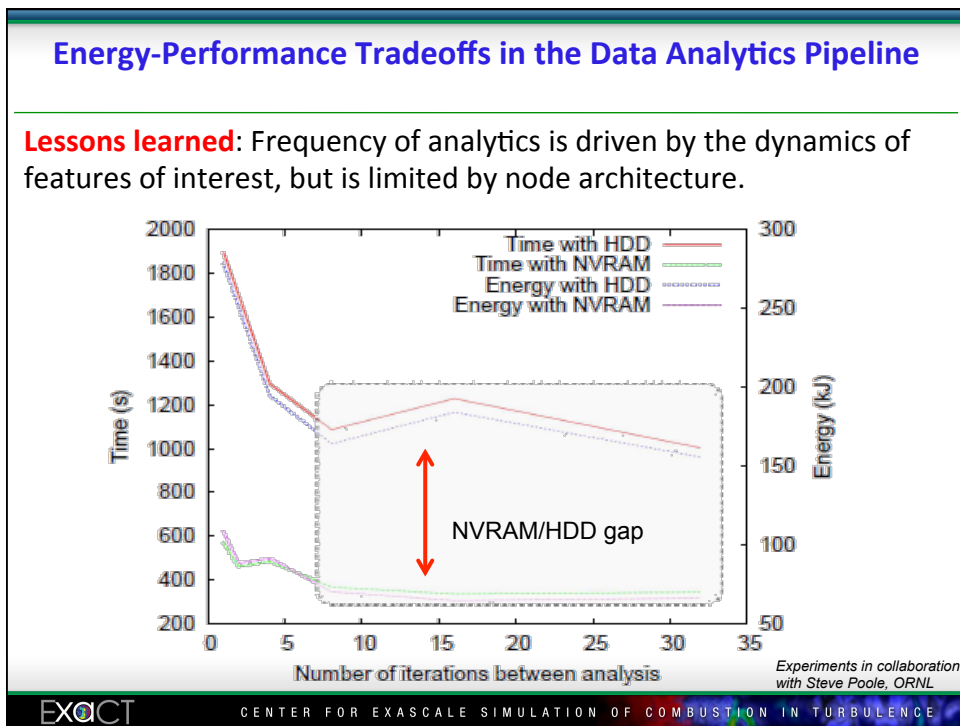
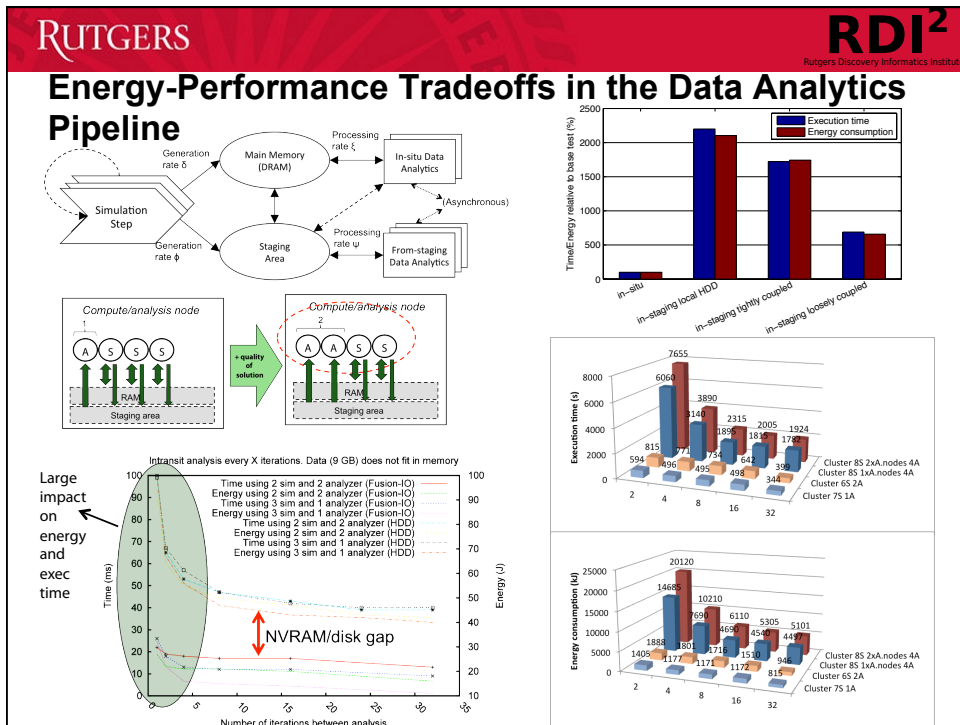
**RUTGERS**
**RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

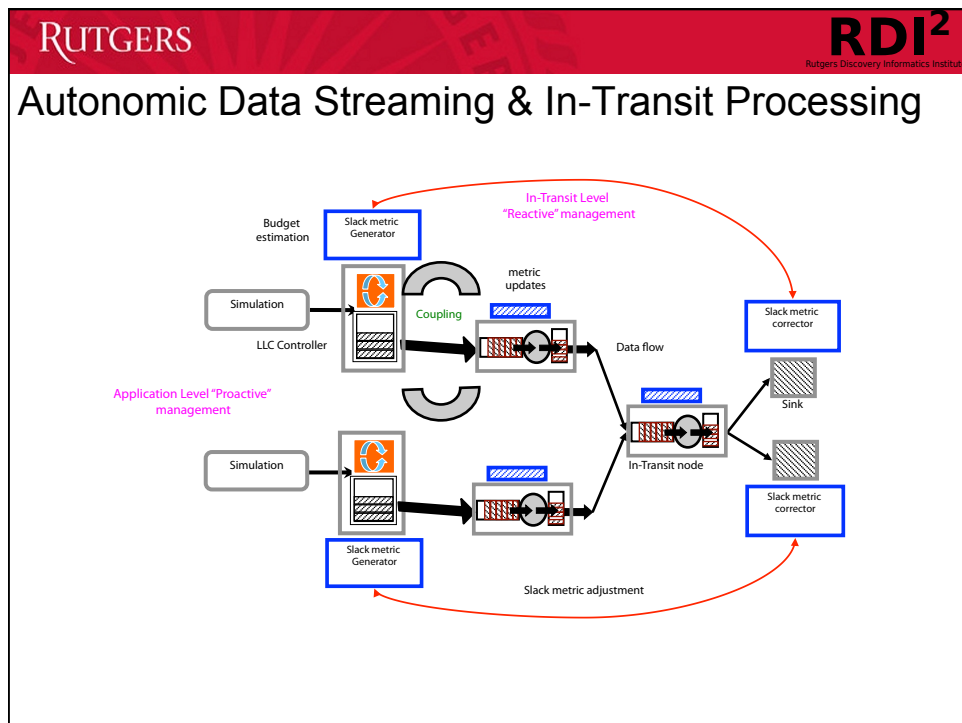
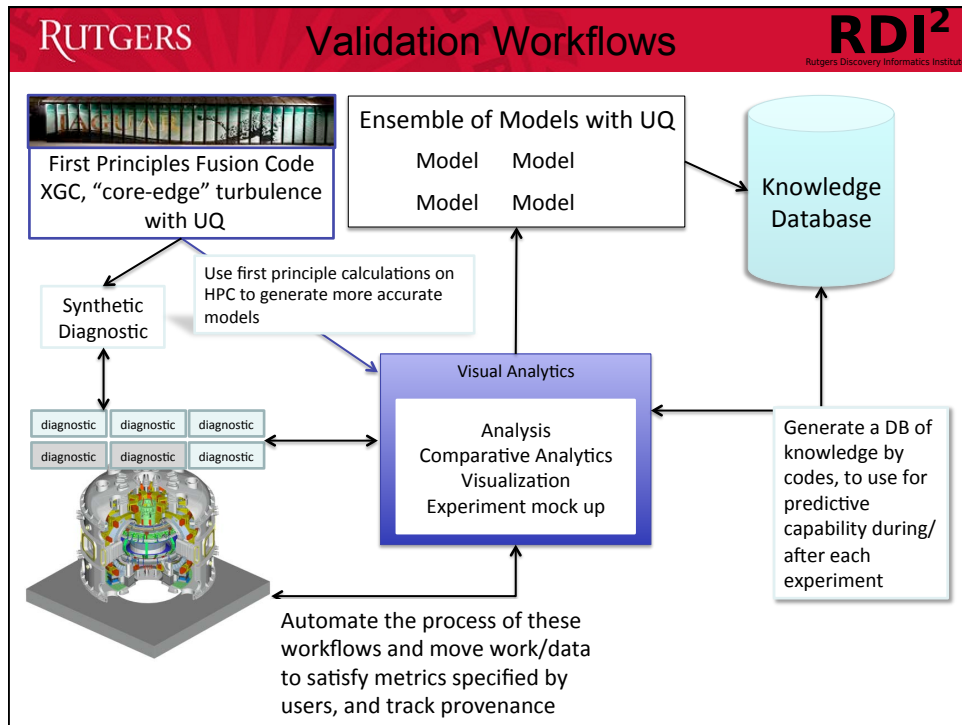
## ActiveSpaces for Remote Online Debugging

```

    graph LR
      subgraph ORNL_Jaguar_Computing_nodes [ORNL Jaguar Computing nodes]
        XGC[XGC Kinetic Code]
        M3D[M3D-OMP equilibrium code]
        AS((ActiveSpace))
        XGC -- "data objects inserted into space" --> AS
        AS -- "data objects retrieved from space" --> M3D
      end
      subgraph ORNL_Jaguar_Login_node [ORNL Jaguar Login node]
        DSPC[DataSpaces Proxy Client]
      end
      subgraph Remote_commodity_machine [Remote commodity machine]
        ASM[ActiveSpace Monitoring]
        Graph[Graph]
      end
      AS <--> DSPC
      DSPC --- WAN((Wide Area Network))
      WAN --- ASM
      ASM --> Graph
  
```

- Use data kernels to debug the science
  - load data kernels to retrieve data for visualization
  - get insights into simulation evolution by analyzing the data
- Use data kernels to steer simulation execution
  - inject data parameters into the space for conditional execution
- Deployed in distributed environments, e.g., Rutgers and ORNL







**RUTGERS** **RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## 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 Rutgers-Spaces project explores solutions at various levels:
  - High-level programming abstractions for in-situ workflows for code coupling and online analytics
  - Efficient runtime mechanisms for hybrid staging, locality-aware mapping and location-aware data movement
  - Support for dynamic code deployment and execution for moving code to data

**RUTGERS** **RDI<sup>2</sup>**  
Rutgers Discovery Informatics Institute

## Thank You!



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

Email: [parashar@rutgers.edu](mailto:parashar@rutgers.edu)  
 WWW: [rdi2.rutgers.edu](http://rdi2.rutgers.edu)

## Next generation data challenges of computational simulations

- At the **architecture** or node level
  - Use increasingly deep memory hierarchies coupled with new memory properties
- At the **system** level
  - Cope with I/O rates and volumes that stress the interconnect and can severely limit application performance
  - Can consume unsustainable levels of power
- At the **extreme scale**
  - Immense aggregate I/O needs with potentially uneven loads placed on underlying resource
  - Can result in data hotspots, interconnect congestion and similar issues