



Performance Analysis of Scheduling and Replication Algorithms on Grid Datafarm Architecture for High Energy Physics Applications

Atsuko Takefusa (Ochanomizu Univ.)

Osamu Tatebe (AIST)

Satoshi Matsuoka (TITECH/NII)

Youhei Morita (KEK)

DataGrid

- Grid environment for ubiquitous access and analysis of large-scale data
E.g. CERN Large Hadron Collider (LHC) experiment (starting in 2008)

- High speed file transmission (GridFTP)

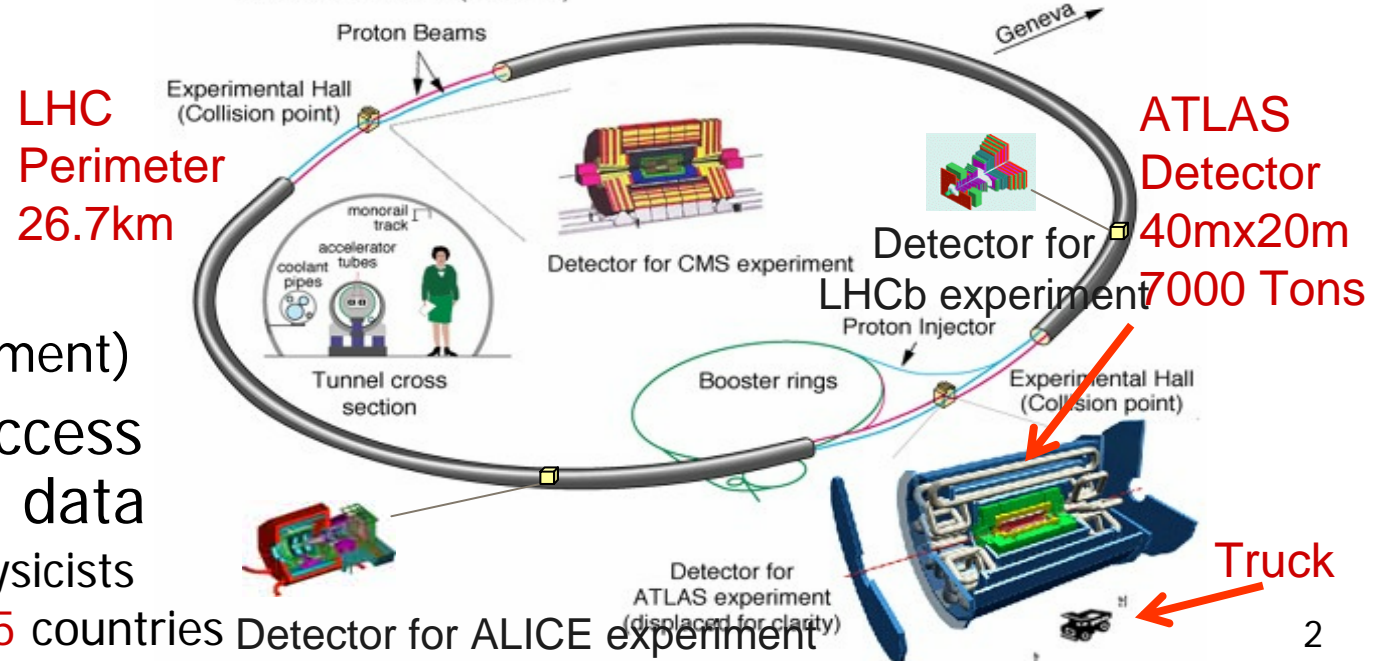
- Data replica management (Giggle, Globus Replica Management)

- High speed access of large-scale data

~2000 physicists

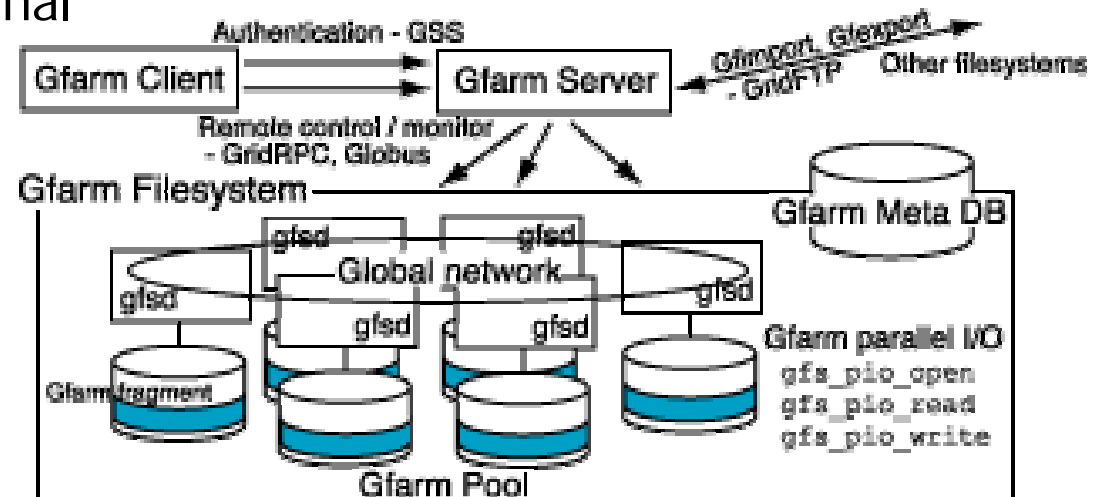
from 35 countries

Large Hadron Collider at CERN
Circumference 26.7 km (16.6 miles)

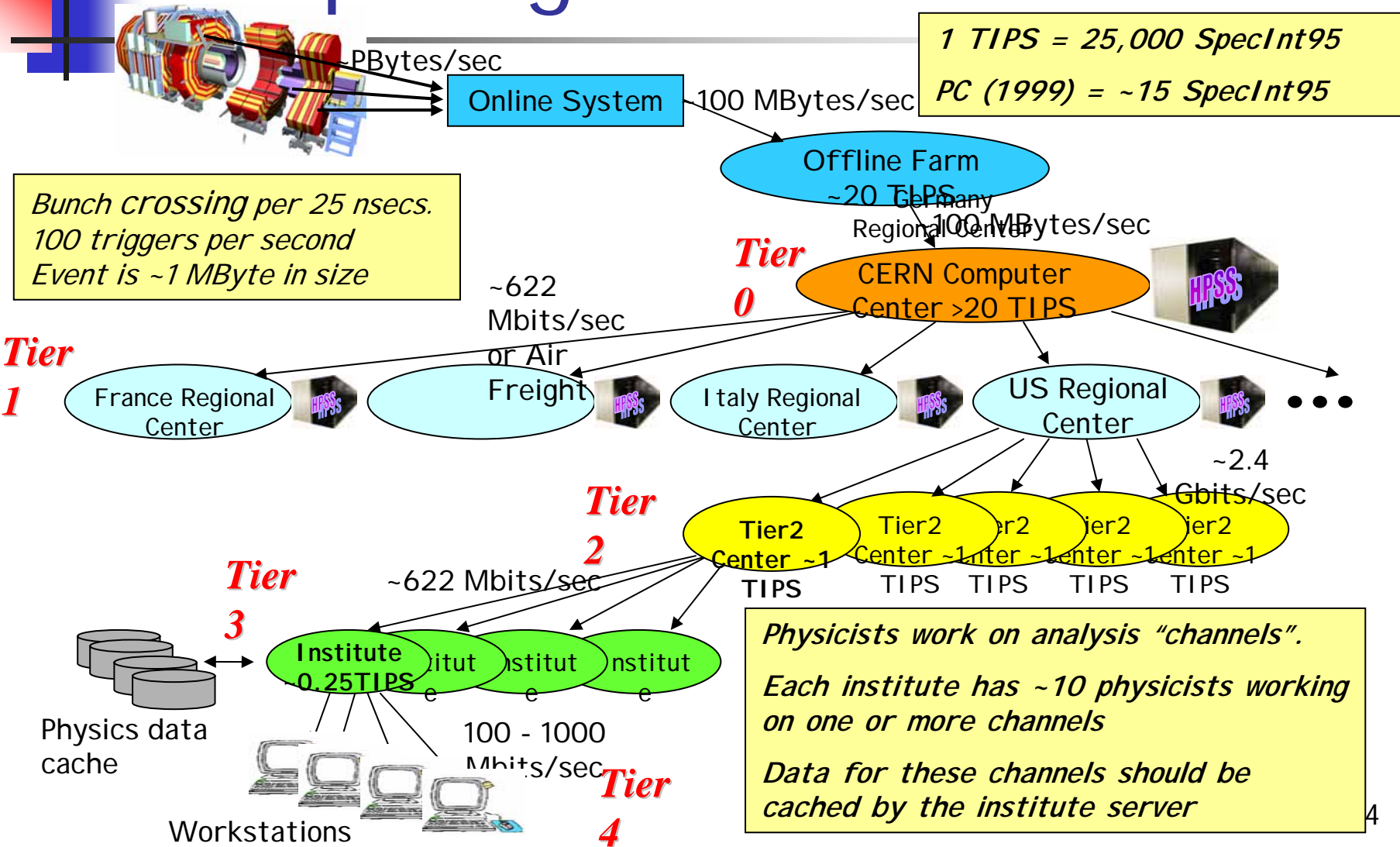


High Speed Data Access for DataGrid

- Existing techniques: HPSS, cluster file systems:
 - Loosely coupled between computational and I/O nodes
 - Difficult to realize scalable local I/O bandwidth
 - Takes about **12 days** to read 1 [PB] data on 1 [GB/sec] bandwidth network!
- Grid Datafarm architecture [AIST, KEK, TITECH, U. of Tokyo]:
 - Fusion of computational and I/O nodes
 - Affinity scheduling enabled over **TB/sec** local I/O bandwidth



MONARC Hierarchical Computing Model for LHC

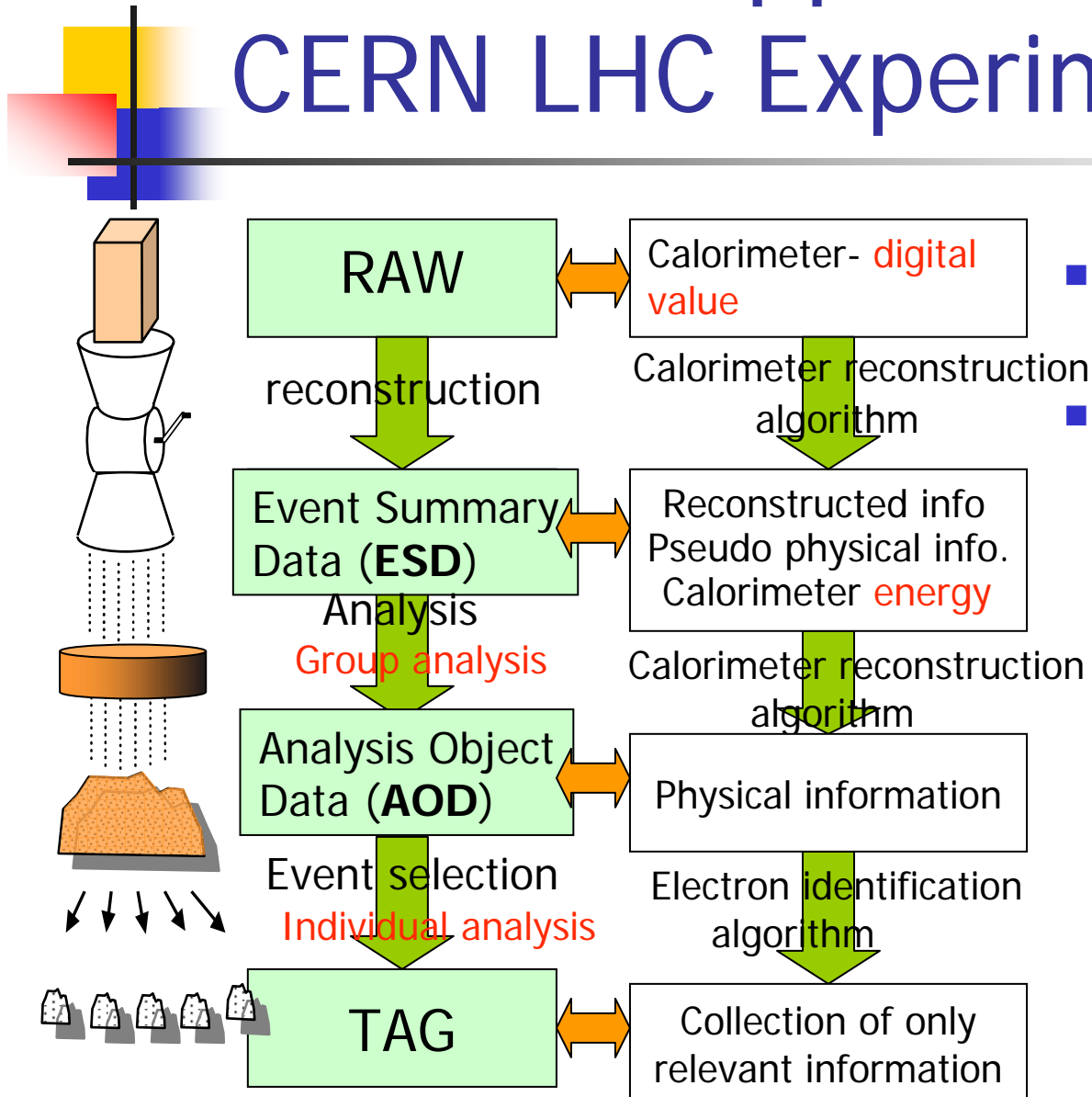




Performance Analysis of Scheduling and Replication Algorithms on Grid Datafarm Architecture

- Comparison of DataGrid models
 - MONARC-style hierarchical model vs. centralized data storage model
- Comparison of scheduling and replication algorithms
 - Owner Computes + background replication vs. MCT + on-demand replication
- Realistic assumptions of job processing for CERN LHC experiments starting in 2008
- Assuming Grid Datafarm architecture
- Experiments on the Bricks Grid simulator

DataGrid Application Scenario: CERN LHC Experiment

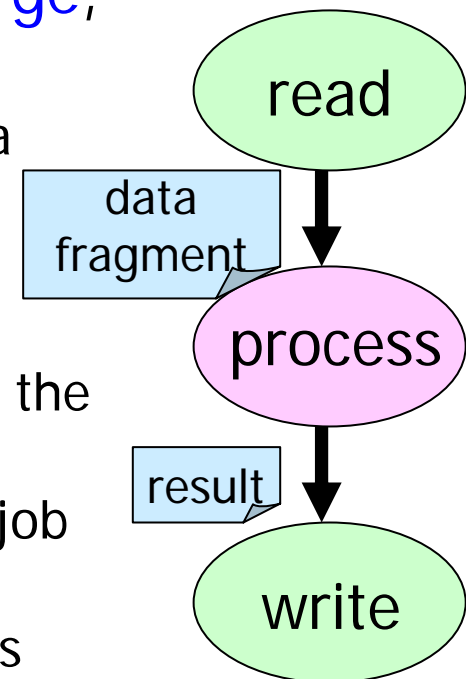


- Collect data from collisions of particles
- Analyze data in different levels of hierarchy
 - RAW→ESD (**Large**)
 - 2-4 times/year
 - ESD→AOD (**Medium**)
 - once/month
 - AOD→TAG (**Small**)
 - once/4 hours

Simulation Model: Job Processing

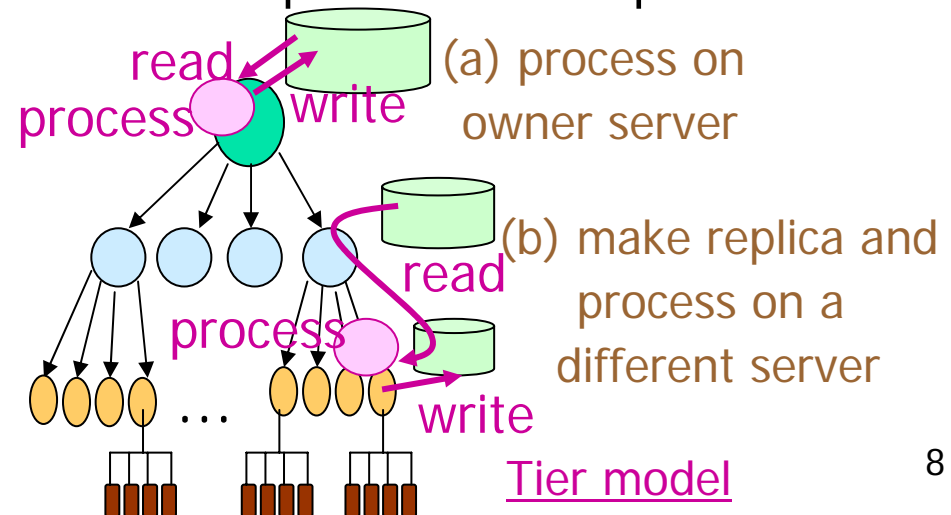
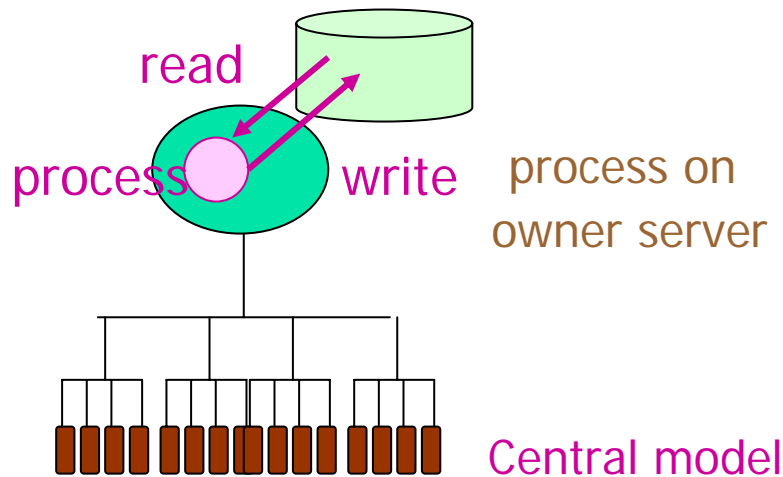
- A typical job for each of the job classes, **Large**, **Medium**, or **Small** is handled as follows:
 - A user (physicist) at the client machine invokes a job
 - The DataGrid scheduler selects a suitable set of servers
 - Each server loads the data fragment required by the job, over the Grid if non-local
 - Each server processes individual portions of the job that the server is assigned to
 - The servers send the output to specified storages
 - (The client receives only statistical data→negligible)
- The time duration to process a job is given as:

$$\text{ResponseTime} = \text{Read} + \text{Process} + \text{Write}$$



Simulation Model: DataGrid Architectures

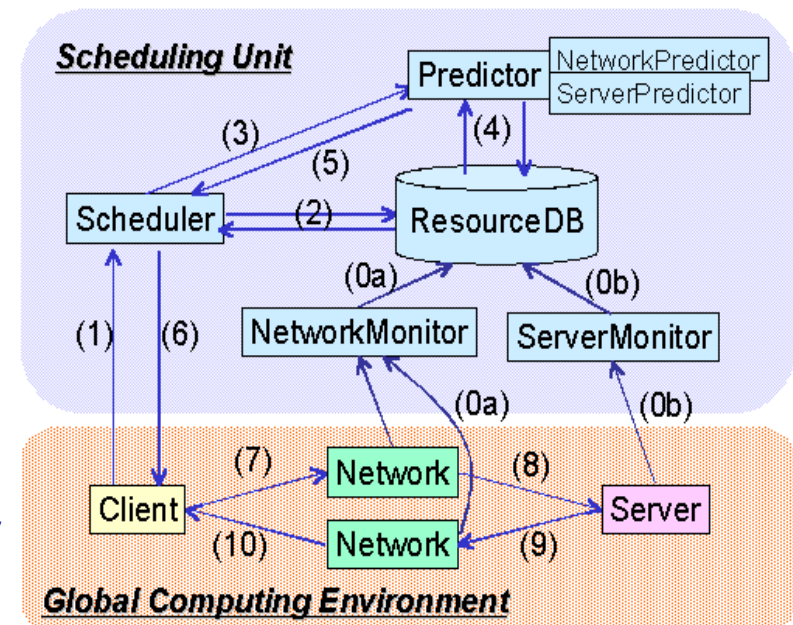
- **Central Model:**
 - All the jobs are processed at a single site
 - Advantages: performance, manageability
- **Tier Model** (MONARC-style multi-tier center model):
 - Jobs are processed in different levels of the hierarchy
 - Must facilitate suitable scheduling and replication policies
 - Advantages: lower power consumption and cost per site



Bricks Grid Simulator

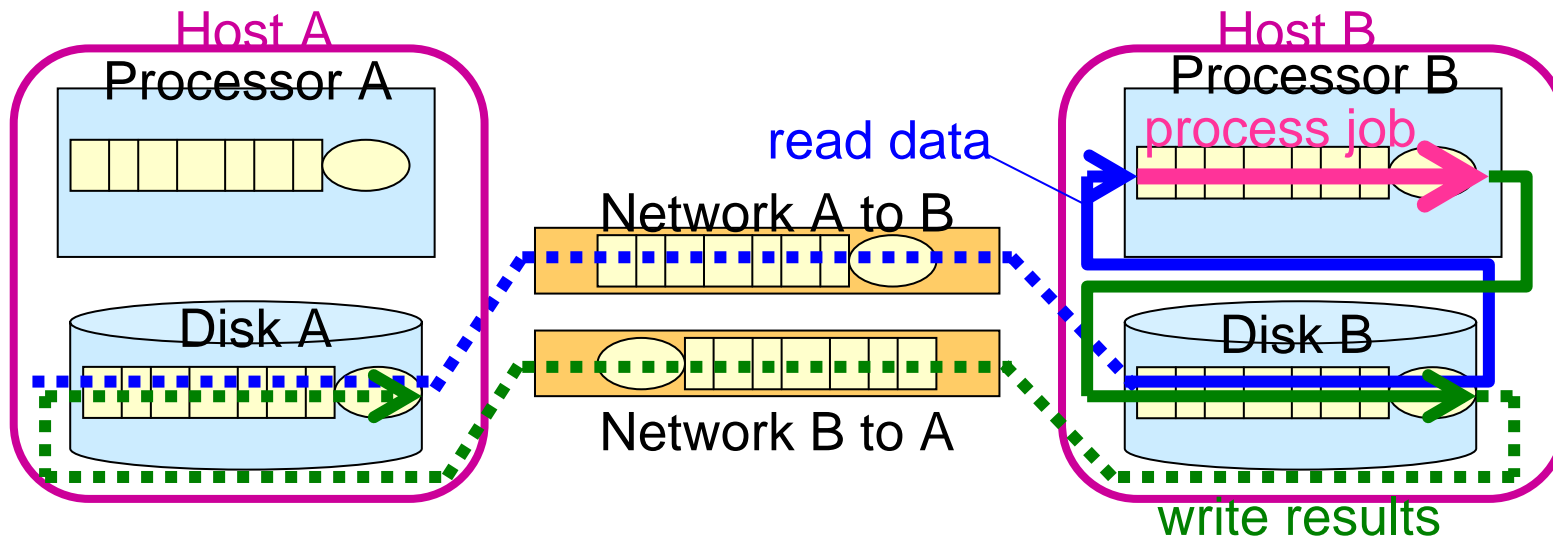
- Java based discrete event simulator
- Represents the behaviors of network and server using queues
- Provides a typical Grid scheduling modules
- Flexible settings of simulation environment
 - Grid topologies (e.g. hierarchical network)
 - Behaviors of Server and network
 - Client model
- Allows to evaluate various scheduling algorithms on dynamic Grid environment

<http://grid-team.is.titech.ac.jp/bricks/>



DataGrid Extension of the Bricks Grid Simulator

- Provides the **Replica Manager** as a Scheduling Unit module
- Provides disk management mechanism
- Represents local disk I/O overheads using queues:
 - Solid lines: a job workflow for the “owner computes” case
 - Solid+dotted lines: a work flow that compute host and data source/destination host are different





Scheduling and Replication Algorithms for Tier Model

- Central Model: file affinity scheduling (disk **owner-computes** rule)
- Tier Model: must facilitate scheduling and replication policies
 - Online scheduling algorithms
→ **on-demand replication**
 - Replication algorithms
→ **background replication**

Online Scheduling Algorithms

(1/2)



Online DataGrid Scheduler determines

- *DataSourceHost* (I/O host)
- *ComputeHost* (compute host)
- *DataDestinationHost* (I/O host)

- **Greedy(MCT: Minimum Completion Time)**
 - Scheduler assigns the job to the host that completes it the earliest

- **OwnerComputes**
 - Scheduler selects a *ComputeHost* that owns the input data and completes the job earliest
 - *DataSourceHost* = *ComputeHost* = *DataDestinationHost*

Online Scheduling Algorithms

(2/2)

■ LoadBound-Read

- Scheduler selects a *ComputeHost* with MCT from the host group which satisfies:

$$Performance_{Specified} > Performance_{Estimated}$$

$$Performance_{Estimated} = ProcessorPerformance / (LoadAverage + 1)$$

- If replicated,
 $DataSourceHost \neq ComputeHost = DataDestinationHost$

■ LoadBound-Write

- Scheduler select a *ComputeHost* which minimizes response time
- If the selected host does not satisfy

$$Performance_{Specified} > Performance_{Estimated}$$

the result is sent to the host that maximizes $Performance_{Estimated}$

- If replicated,
 $DataSourceHost = ComputeHost \neq DataDestinationHost$

Background Replication Algorithm

Replica Manager periodically collect status of hosts and trigger replica creation and migration at its discretion

- LoadBound-Replication:

- Compute $Performance_{Estimated}$ for all the hosts

$$Performance_{Estimated} = ProcessorPerformance / (LoadAverage + 1)$$

- If $Performance_{Specified} > Performance_{Estimated}$, the Replica Manager creates a replica with the largest $AccessRate$ and migrate from the host with the min. $Performance_{Estimated}$ to the host with the max.

$$AccessRate = N_{accesses} / (T_{Current} - T_{Stored})$$

$N_{accesses}$, $T_{Current}$, T_{Stored} : current time, time the data was stored, and the total # of accesses to the data for the duration



Replica Elimination Algorithm

If the disk space for a job turns out to be insufficient, or some x % of hosts do not embody some y % of available disk space within the entire DataGrid, “replica Elimination” is performed

1. Select data that have replicas within the Grid
2. Sort all the selected data by the last recently used (LRU) time
3. Compute Access Rate $AR_{Elimination}$ for the first N data on the list
 $AR_{Elimination} = AccessRate / N_{Copies}$ (N_{Copies} : # of replicas)
4. Select and eliminate replica for data with min. $AR_{Elimination}$ while maintaining the following condition:
 $TotalDiskSize \times Compactness > AvailableDiskSize$
($Compactness$ determines the frequency of elimination)
5. If the above condition is not satisfied, return to step 3 for the next N data (N = 10 in our simulations)

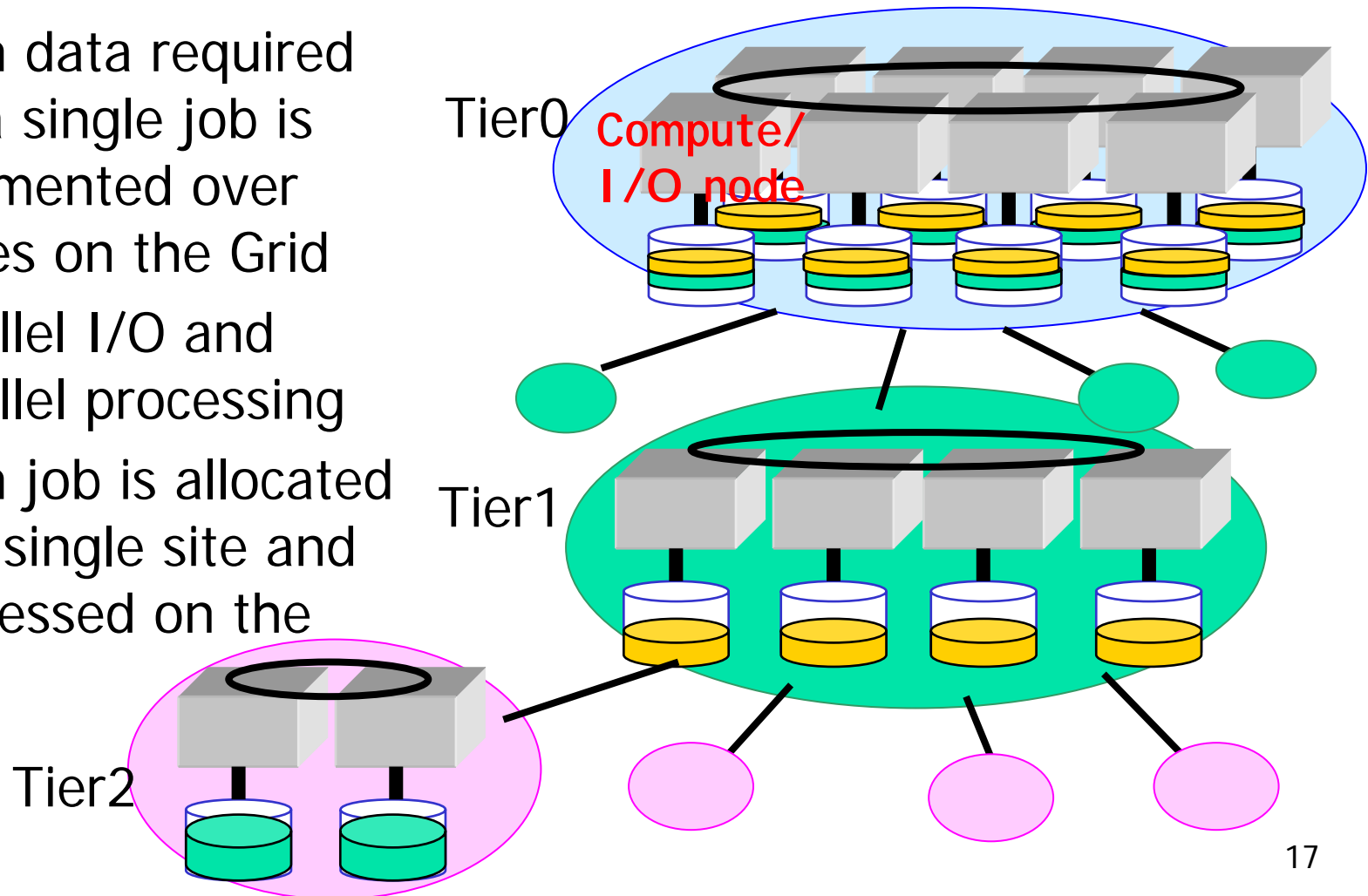


Evaluation on Bricks

- Investigate the performance of our simulation scenarios for the LHC experiment:
 - Grid Datafarm Architecture
 - Central Model vs. Tier Model
 - 5 different policies: 4 scheduling algorithms + 1 replication algorithms
 - Data access localities (random, [time locality](#))
- Compare
[Response time](#), network bandwidth, load average, distribution of data fragments / # of replicas

Experimental Environment for the Grid Datafarm System

- Each data required for a single job is fragmented over nodes on the Grid
- Parallel I/O and parallel processing
- Each job is allocated to a single site and processed on the site



Simulation Settings: Environment

Model	Disk [PB]	Performance [MSpecInt95]	# of Nodes on the Site	Total I/O Bandwidth
Central	2	0.5-1.8	10000	1[TB/sec]
Tier	T0 (x1): 2	0.6/0.5/ 0.4	10000	1[TB/sec]
	T1 (x4): 1	0.3/0.25/ 0.2	5000	500[GB/sec]
	T2 (x16): 0.1	0.03/0.025/ 0.02	500	50[GB/sec]

- Similar in settings to the GriPhyN simulation[Grid2001] for Tier Model
- Queuing theory indicates
 - Est. avg. response time in Central Model is 38.575-1.337 [hours]
 - Central Model with under 0.453318[MSpecInt95] saturates and cannot process LHC jobs
- WAN and Local I/O bandwidth are set to 10[Gbps] and 100[MB/sec]
- On Grid Datafarm architecture aggregation local I/O bandwidth is:

Total I/O Bandwidth = Local I/O Bandwidth × # of nodes on the site



Simulation Settings: Job

Job	Comp. Size [GSI95*sec]	Frequency (Avg.)	Input [TB]	Output [TB]
Large: RAW→ESD	1000	1/4 [months]	1000	100
Medium: ESD→AOD	25	1/1 [months]	100	10
Small: AOD→TAG	5	1/4[hours]	10	0.1

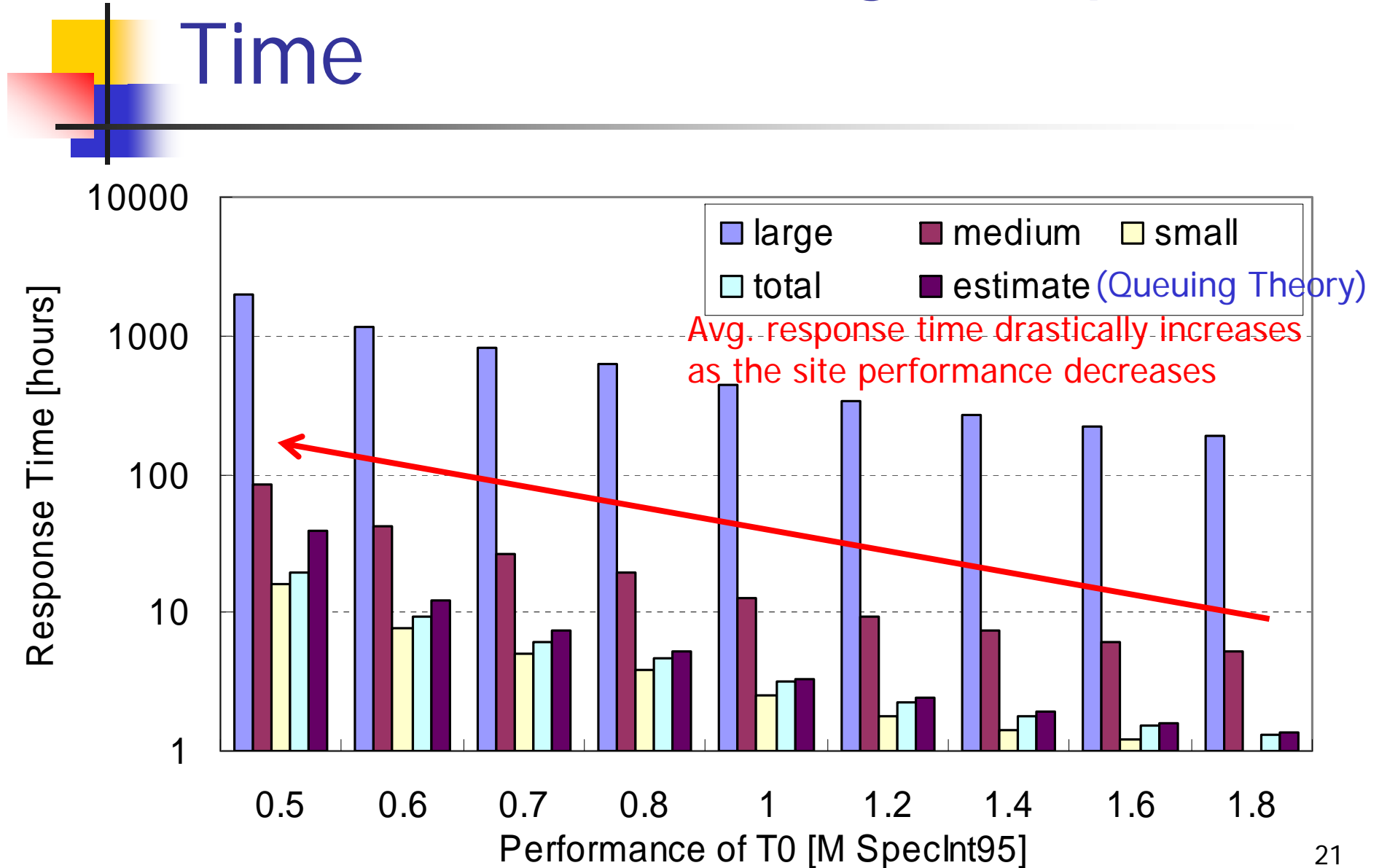
- # of events / job = 1G
- Actual parameters for LHC in the MONARC Report2
- All the initial data(1PBx1, 100TBx2, 10TBx4) are stored in the Tier0 site and the total # of data is increasing during the simulations
- 1 year simulation x 10 for each algorithms
- executed simulation on the Presto III cluster (Dual Athlon MP 1900+, 768MB memory, 256 nodes) at TITECH

5 Combinations of Scheduling and Replication Algorithms

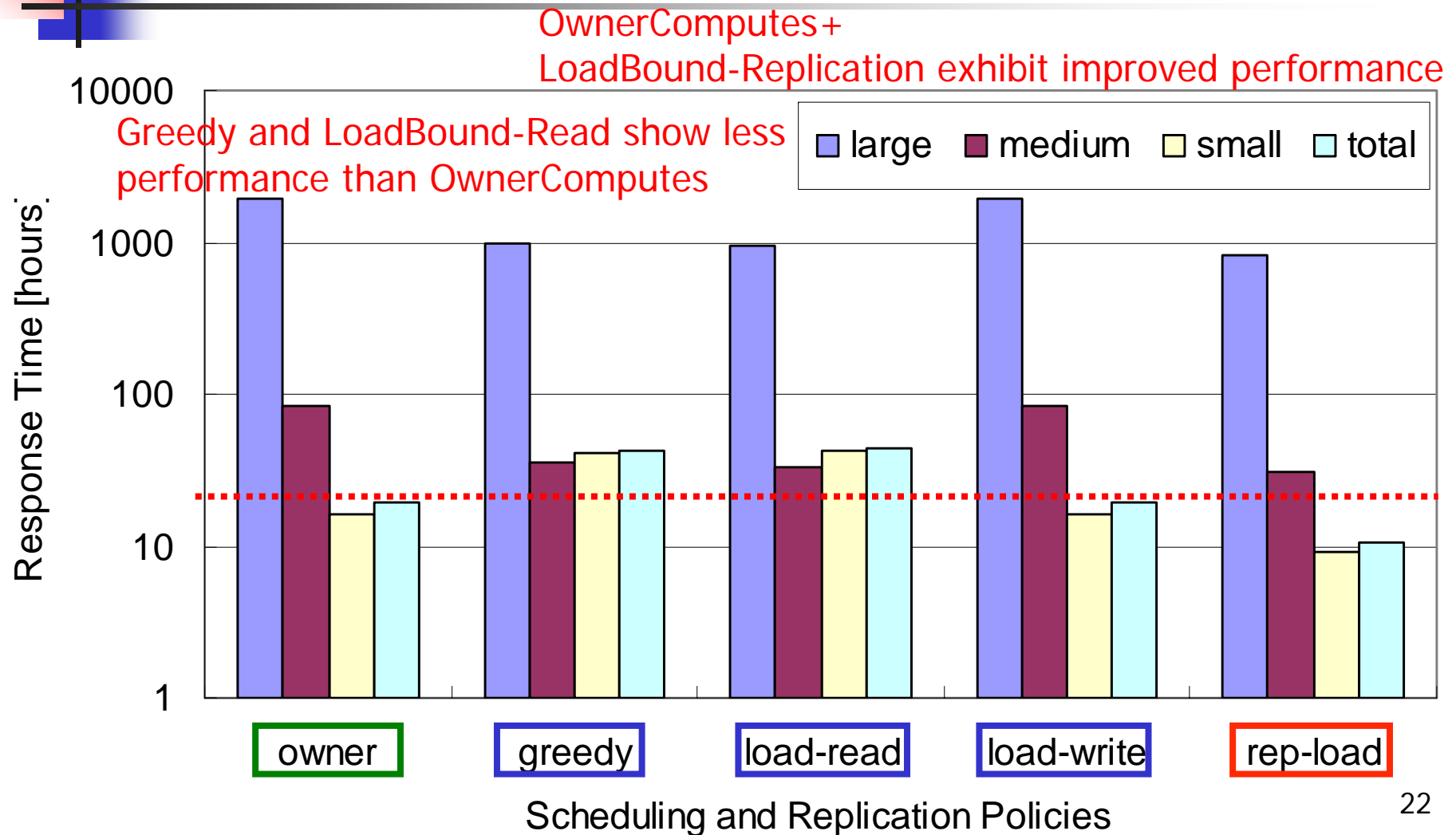
Scheduling Algorithm	Replication Algorithm	ComputeHost Selection	Timing of Replication	Object of Replica Creation	Destination of Replica
Greedy	-	MCT	Read	Input Data	Compute Host
Owner Computes	-	Owner + MCT	-	-	-
LoadBound-Read	-	MCT + Load	Read	Input Data	Compute Host
LoadBound-Write	-	MCT + Load	Write	Output Data	Arbitrary
Owner Computes	LoadBound-Replication	Owner + MCT	Arbitrary	Arbitrary	Arbitrary

Scheduling (on-demand replication) vs. OwnerComputes + (background) replication

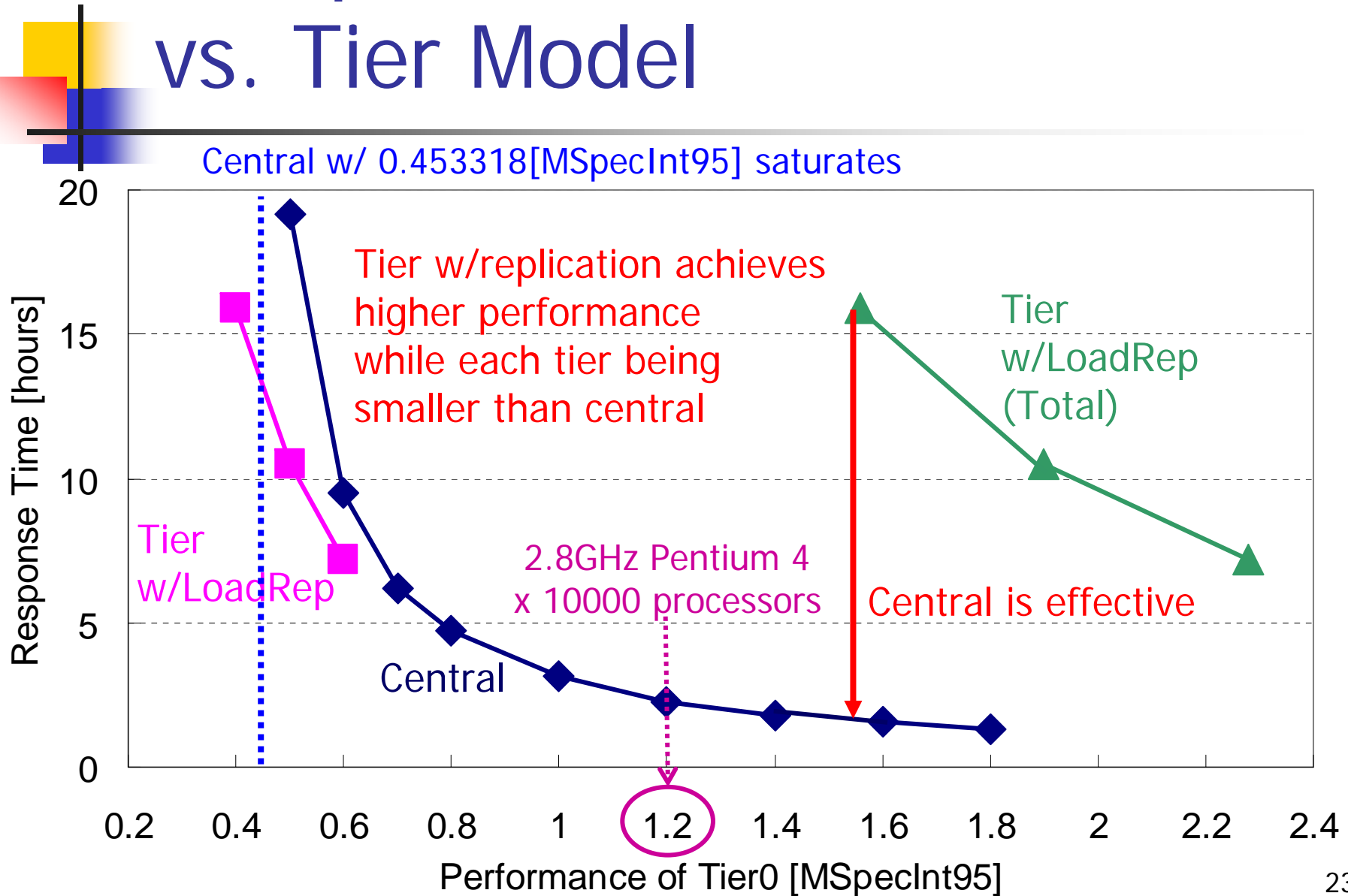
Central Model: Avg. Response Time



Tier Model: Avg. Response Time (T0, T1, T2) = (0.5, 0.25, 0.025)



Comparison of Central Model vs. Tier Model





Related Work: GriPhyN Simulation Study [HPDC-11, '02]

- GriPhyN [Univ. of Chicago et al]
 - Targets the CERN LHC experiment
 - Construct Globus-based peta-scale virtual DataGrid
 - Simulation study for scheduling and replication for DataGrid
 - Propose External Scheduler, Local Scheduler, and Dataset Scheduler for DataGrid scheduling and replication
 - Evaluated External and Dataset Scheduling Algorithms
 - JobDataPresent (OwnerComputes) + replication shows improved performance
 - Not assume the actual parameters for the LHC experiment (small job granularity, shorter duration, # of "original" data is NOT increasing)
- ↔ Experiments on Grid Datafarm architecture
w/ Actual parameters for the CERN LHC



Conclusions

- Evaluate the performance of DataGrid system models by using the Bricks Grid simulator, newly enhanced with DataGrid
 - Affinity scheduling enabled over **TB/sec** local I/O bandwidth on the Grid Datafarm architecture
- Compare Central vs. MONARC-style Tier Model, assuming the Grid Datafarm architecture:
 - Central Model is effective and could be constructed in a feasible fashion at this point
 - In Tier Model, OwnerComputes + LoadBound-Replication proves to be effective
 - Tier Model achieves higher performance while each tier being smaller than central