IRis

Northeastern University

Research Thrust R3 Presentations

CenSSIS Research Thrust R3: Image and Data Information Management

January 01, 2007

Design of workflow scheduler

Juemin Zhang
Northeastern University

Waleed Meleis
Northeastern University

David Kaeli
Northeastern University

Recommended Citation

Zhang, Juemin; Meleis, Waleed; and Kaeli, David, "Design of workflow scheduler" (2007). *Research Thrust R3 Presentations.* Paper 4.
http://hdl.handle.net/2047/d10009097

This work is available open access, hosted by Northeastern University.

Scheduling MPI Workflow Applications on Computing Grids

Juemin Zhang, Waleed Meleis, and David Kaeli

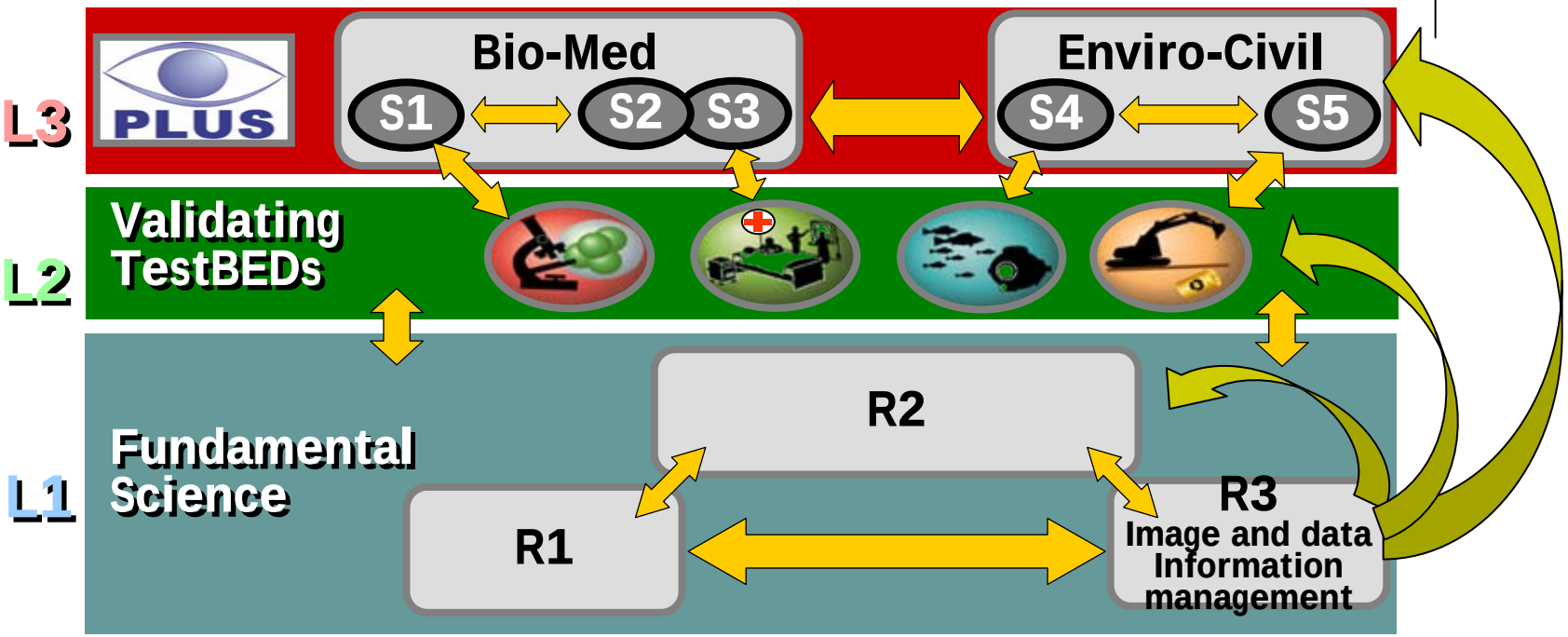
Electrical and Computer Engineering Department, Northeastern University, Boston, MA 02115

jzhang, meleis, kaeli@ece.neu.edu

Acknowledgement

This work is supported in part by CenSSIS, the Center for Subsurface Sensing and Imaging Systems, under the Engineering Research Centers Program of the National Science Foundation (Award # EEC-9986821).

Value added to Censsis



This work falls under Research thrust R3, image and data information management. This work can be applied to image analysis applications in all three levels, including modeling, simulation as well as other areas requiring intensive computation or accessing distributed data set.

Grid-computing

- **Grid problem**
  - flexible, secure, coordinated resource sharing among a dynamic collection of individuals, institutions and resources – referred to as *virtual organizations*. (From "*The Anatomy of The Grid*", by I. Foster, C. Kesselman, and S. Tuecke)
- **Computing Grid:**
  - Multiple independently managed computing sites which are connected to a public network through gateway nodes
  - Computing site:
    - Collection of computing resources (nodes)
    - Single administrative domain (batch job system)
    - Local/private network: connecting all computing resources

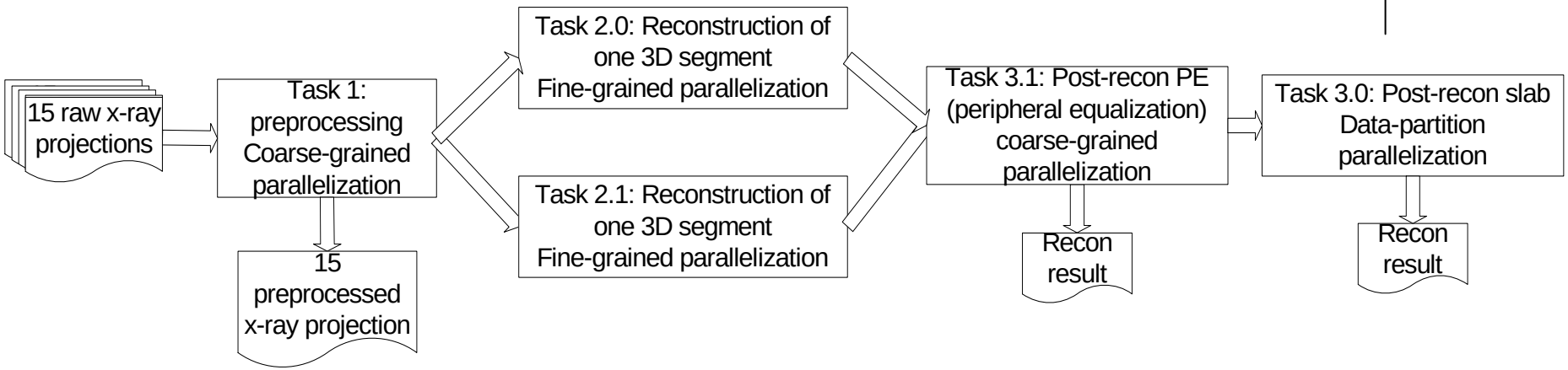
Why Grid-computing

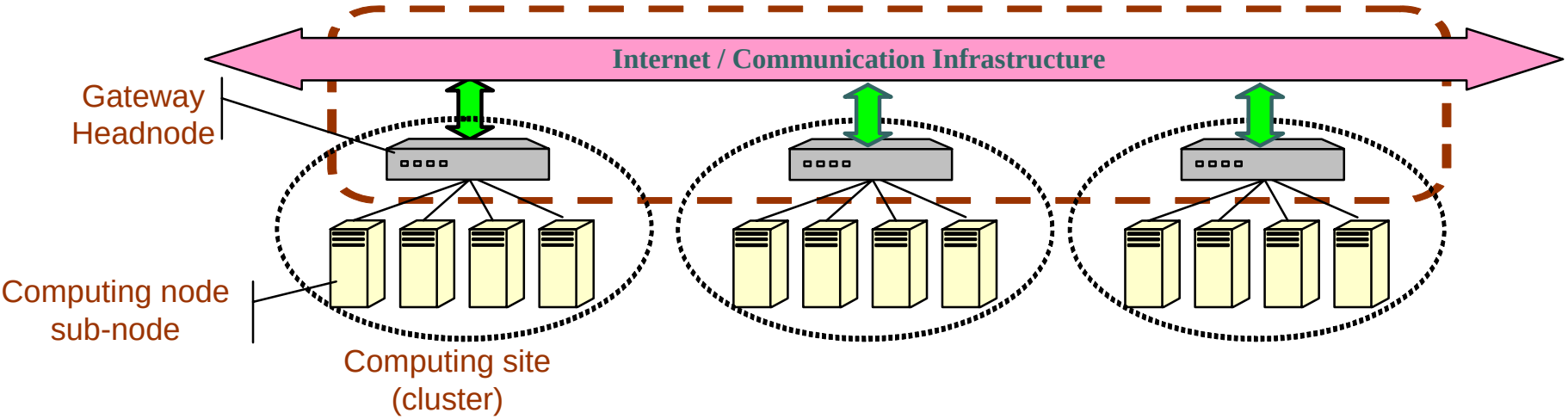
- **Characteristics of computing resources**
  - Increasing number of distributed computing and storage resources are available
  - Low latency and high bandwidth inter-connection
  - Unbalanced loads among resources
- **Characteristics of imaging applications:**
  - Large problems, requiring lots of computation and storage resources
  - Distribute properties, from data acquisition to data access, tend to be distributed among multiple sits.
  - High costs for centralized solution over distributed one

MPI Workflow

- **Workflow**
  - A workflow consists of multiple dependent or concurrent tasks.
  - Dependency: task needs to executed in order
  - Concurrency: tasks are executed in parallel across multiple computing sites
- **MPI Workflow**
  - A task is a parallel MPI execution on multiple computing nodes within a computing site

Tomosynthesis application





The Tomosynthesis image reconstruction process consists of multiple functional tasks, which are executed complying with data their dependency. Tasks are parallelized using MPI library, but each exhibits different parallelism.

Problem Definition

- **Executing MPI Workflow on Grids**
  - Mapping tasks to computing sites
- **Objective:**
  - Performance – tuning the application turn-around time
    - Minimize request queuing time and execution time
  - Throughput – maximize the numbers applications processed during a period of time
  - Resource utilization

MPI Workflow Scheduler

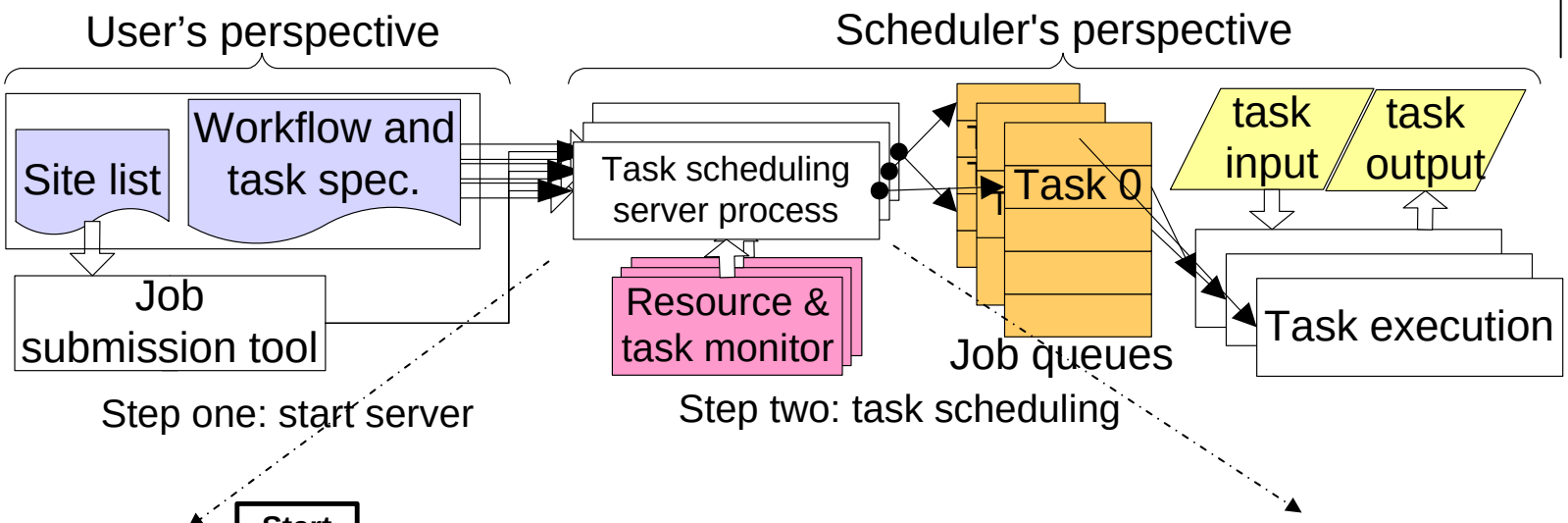
- **Mapping tasks to computing sites**
  - Input:
    - Petri net – workflow execution
    - Task specification: number of nodes
      - Network and physical location transparent
- **Tasks are scheduled, submitted and executed on computing sites of a grid without user interference**
  - Minimize the task request queuing time
  - Minimize the resource co-allocation coordination time

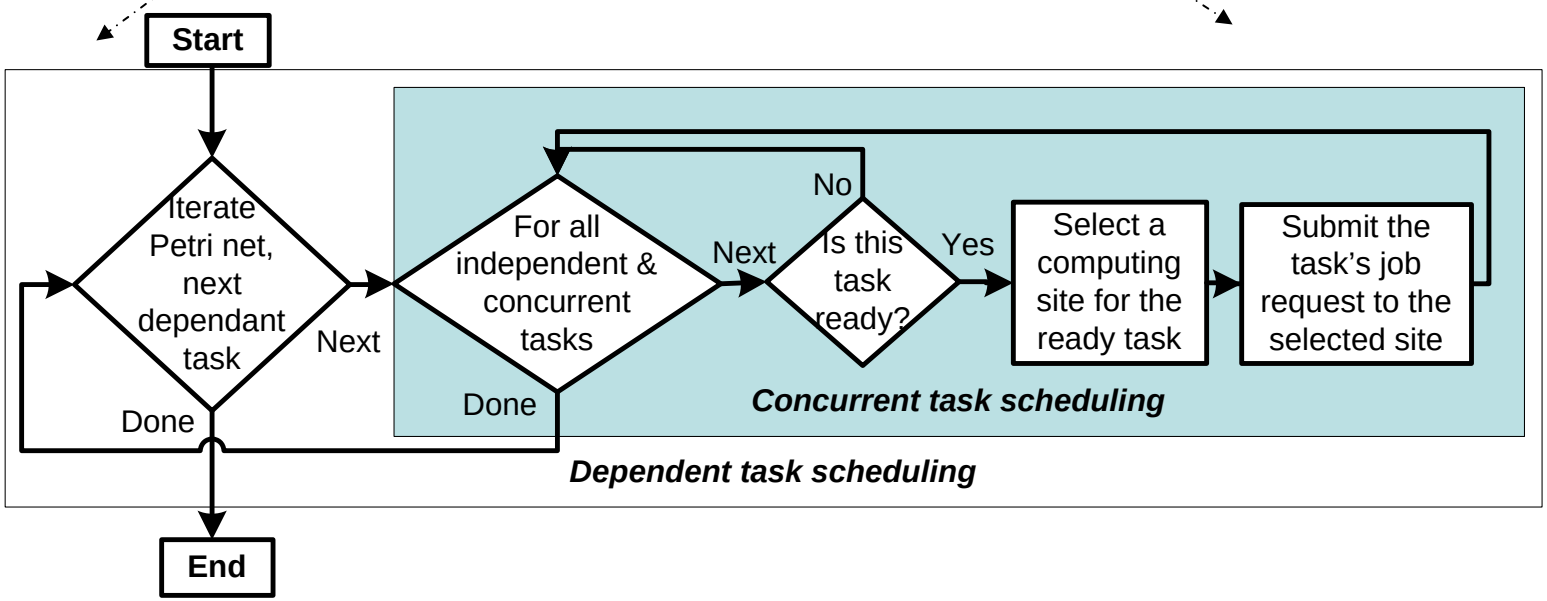
Scheduler Design

- **Part of the complete framework supporting execution of MPI workflow on grids**
  - Message relay, task grouping and task scheduler.

- **Parallel approach**
  - One scheduler process is running on a gateway/headnode of each computing site
  - Message passing is used for inter-process communication
  - Local workload information query
  - Local task submission
  - Collective scheduling decision making process

Task Scheduler Structure





Task Scheduling Algorithm

- **Objective:**
  - For a given task, find a computing site which may yield the shortest queuing time.
- **Task scheduling scheme**
  - Predict site with the shortest queuing time
  - Ranking computing sites by:
    - **The queuing length**
    - **The estimated queuing time**: the queuing length divided by the average system throughput
    - **The number of available resource**

Task Scheduling on Grids

- **Limitations of single-site scheduling decision**
  - Rank is correlated with the task queuing time
    - Assumption: the higher rank may lead to shorter queuing time (not true)
  - Dynamically changing workloads:
    - After tasks are submitted, ranking order may change
- **Our solutions**
  - Duplicate the task request and submit them to different computing sites
  - Using task grouping to resolve redundant task executions at runtime (during MPI initialization)
    - The first running task continues
    - Other redundant task executions starting later will be terminated automatically

Duplicate Task Submission

- **Advantage of task duplication**
  - Dynamically selecting which site to run the task
  - Flooding all computing sites leads the shortest queue time
  - No need to guarantee which computing site has the shortest queuing time

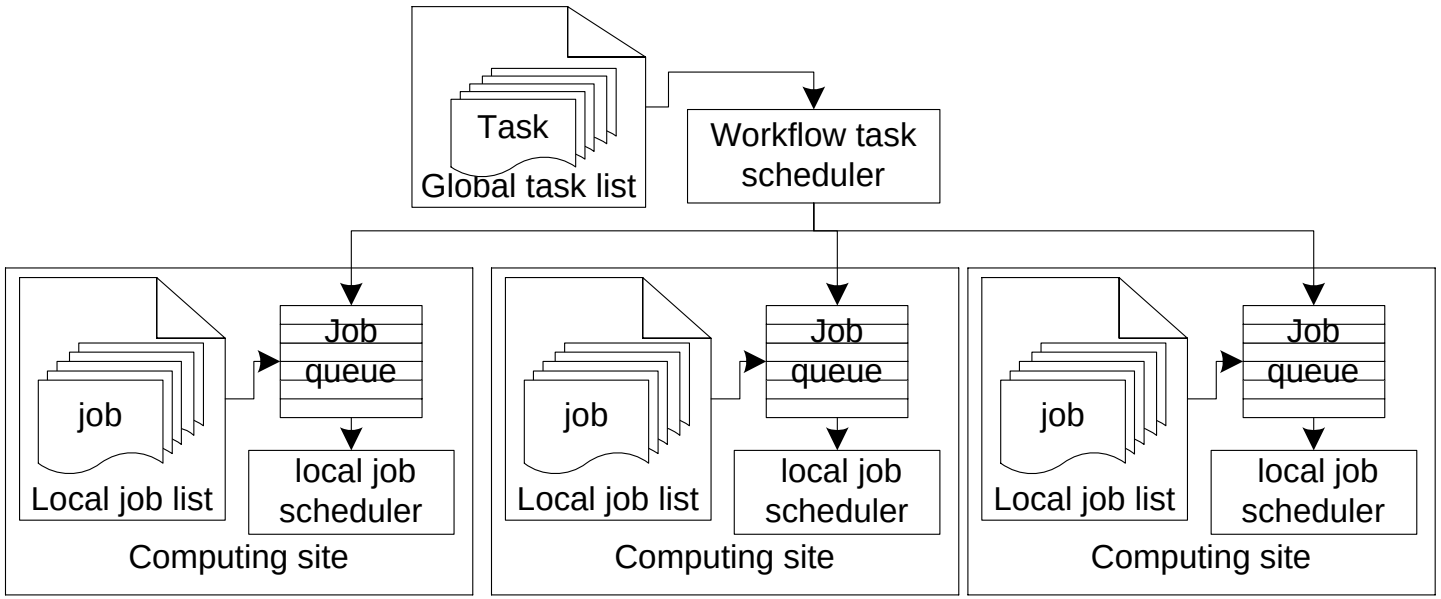
- **Side-effect**
  - There are extra copies of task's requests on different computing sites – higher workload
  - Increase the job queue length and change the job queue scheduling behavior
  - Flooding all computing sites is not favorable for resource management
  - Overheads in resolving duplications.

Modeling Environment

- **Csim based simulation**
  - Computing site: job queue
    - First-come-first-serve, Backfill EASY, and backfill conservative
  - Random workload generation:
    - Inter-arrival time: exponential distribution
    - Job execution time: Zipf distribution
    - Job size: Poisson distribute+higher probability on some special job sizes.
  - Task scheduling schemes:
    - Random selection
    - The queue length,
    - Estimated queue time (queue length / system throughput)
    - available resources

Environment Structure

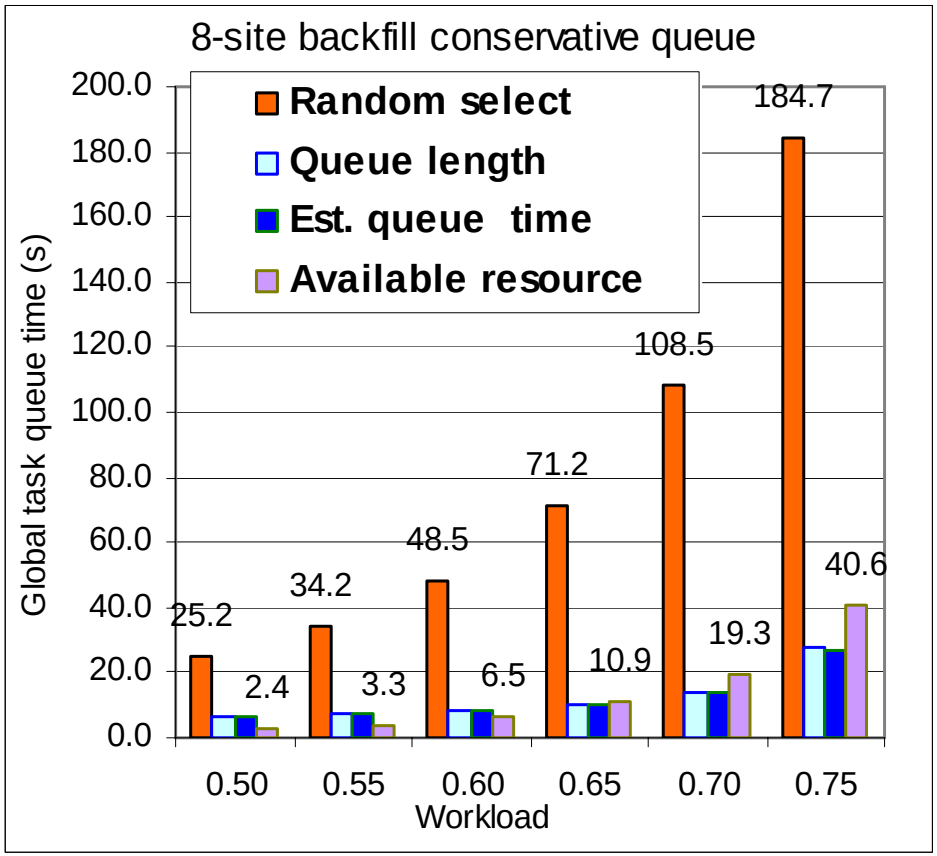
- **Settings**
  - multiple computing sites grid
  - Local workload: 100,000 local jobs for each computing site
  - Global workload:10,000 global tasks for all sites
  - 0.5 ~ 0.75 workload level for all computing sites

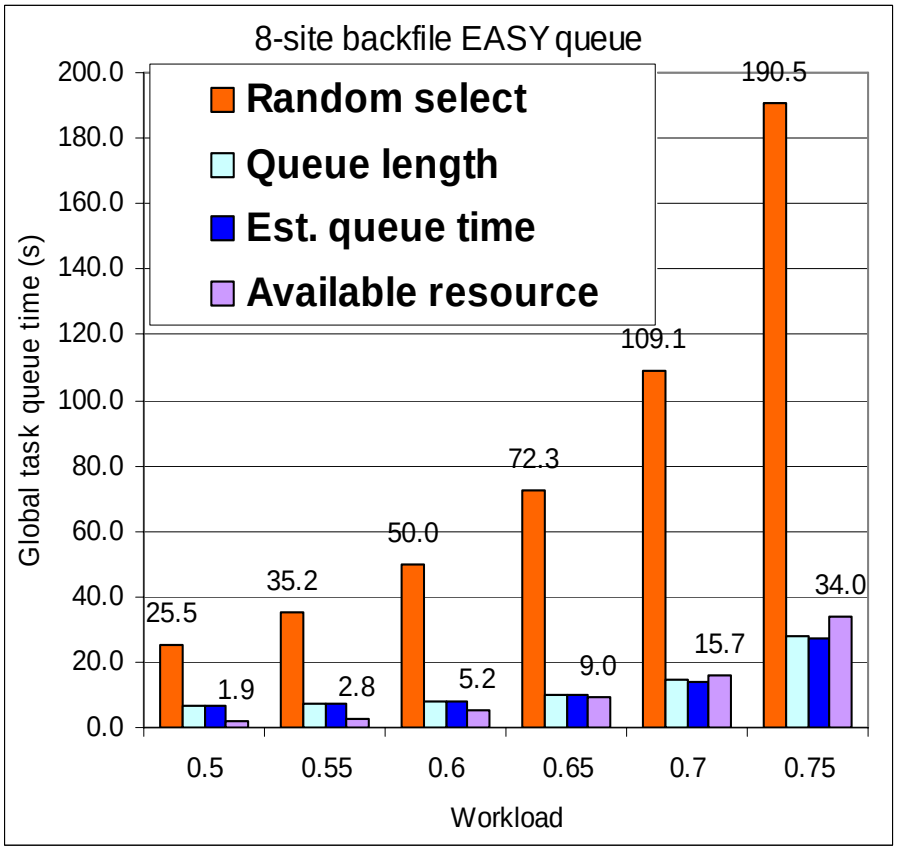


Algorithm Comparison

- **8-site computing grid**

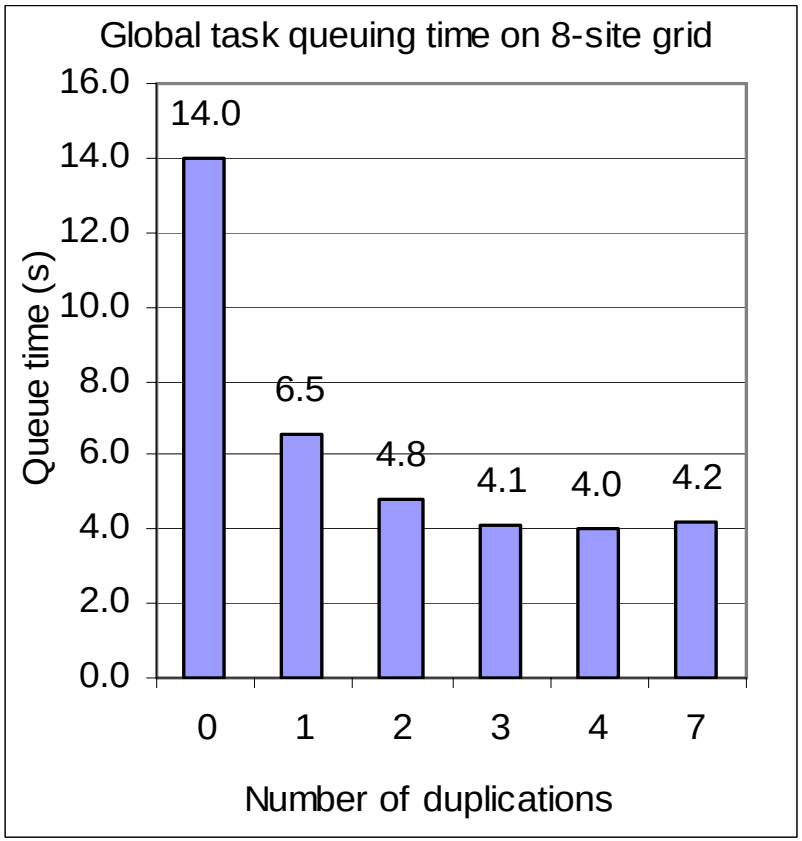
  No duplication is used for each global task

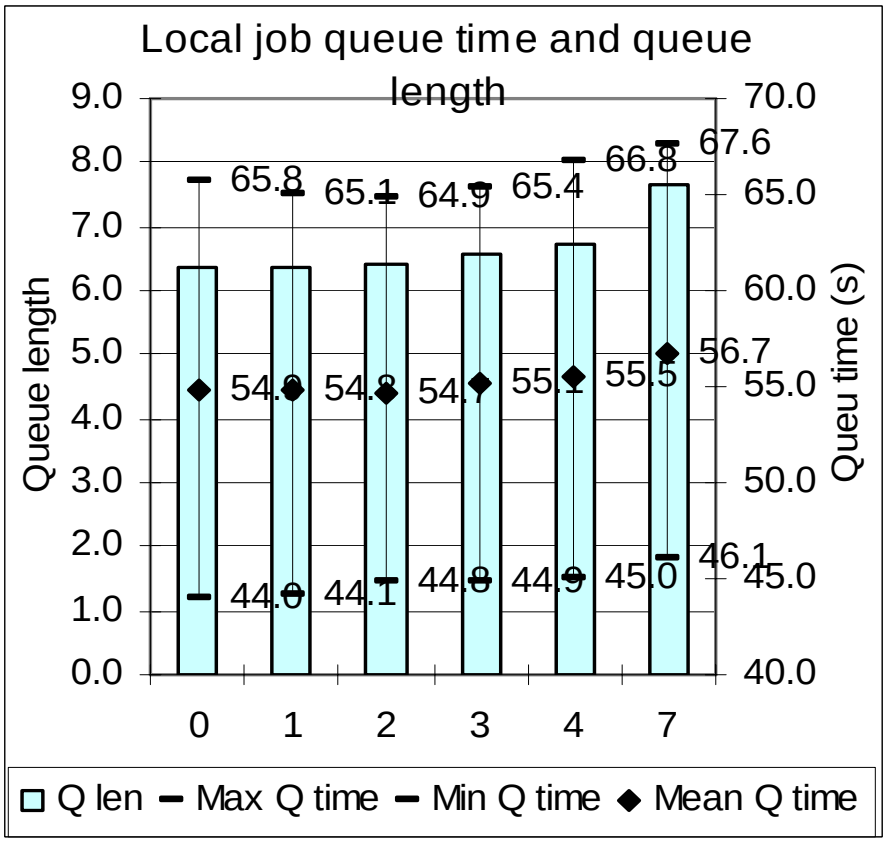




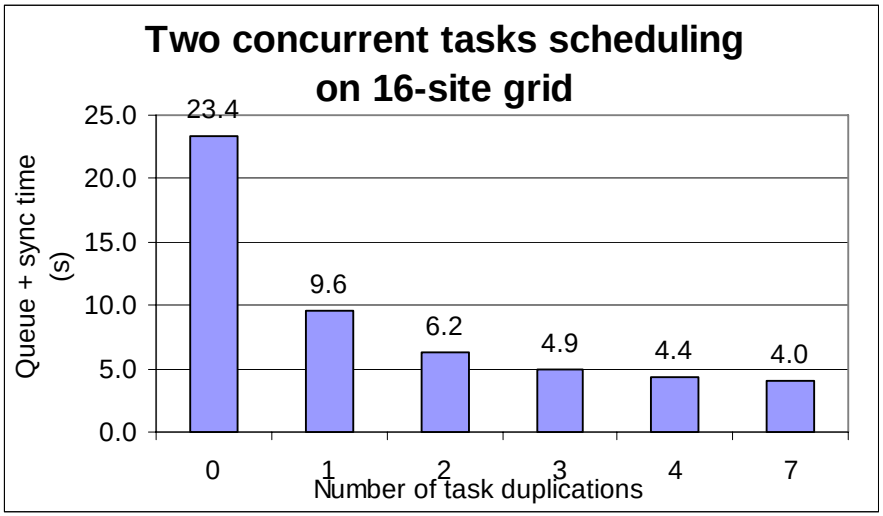
Duplication and Impact

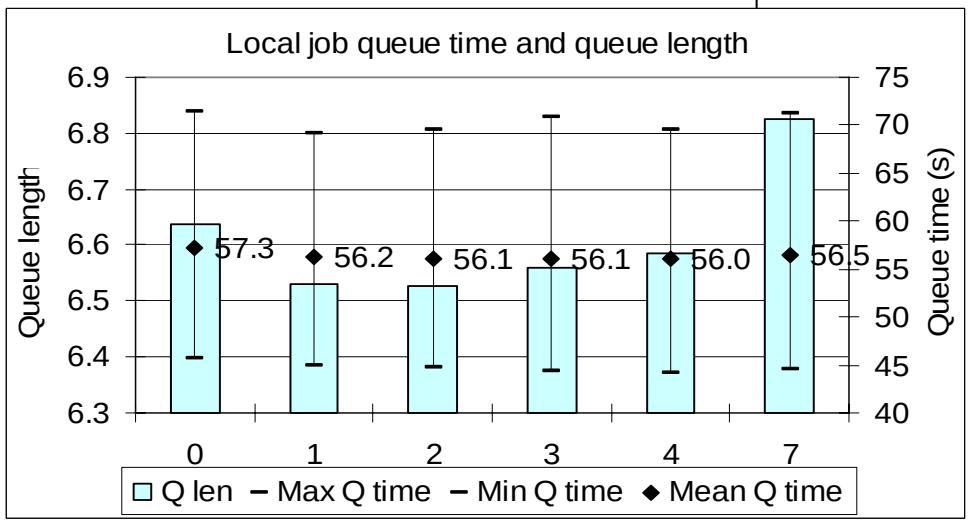
- **8-site grid simulation**
  - **Each site uses backfill conservative queue with 0.7 workload,**
  - **Global task scheduler: the queue length scheduling scheme**

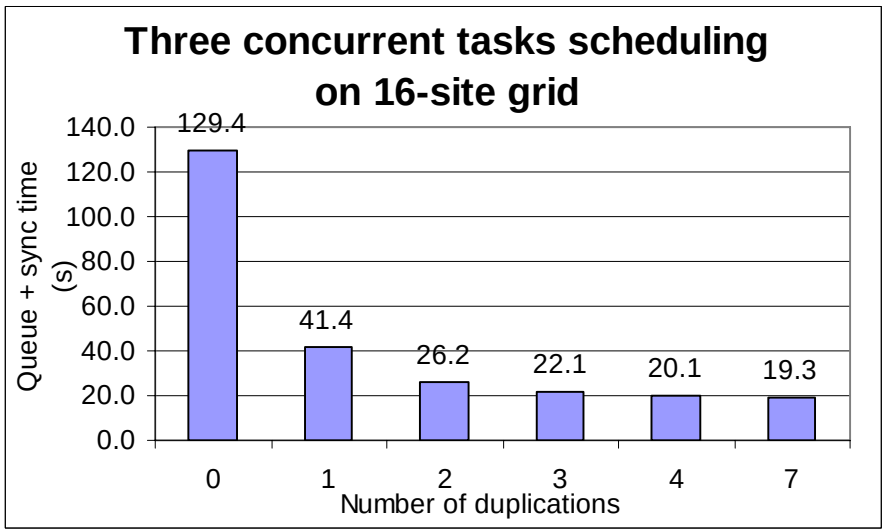


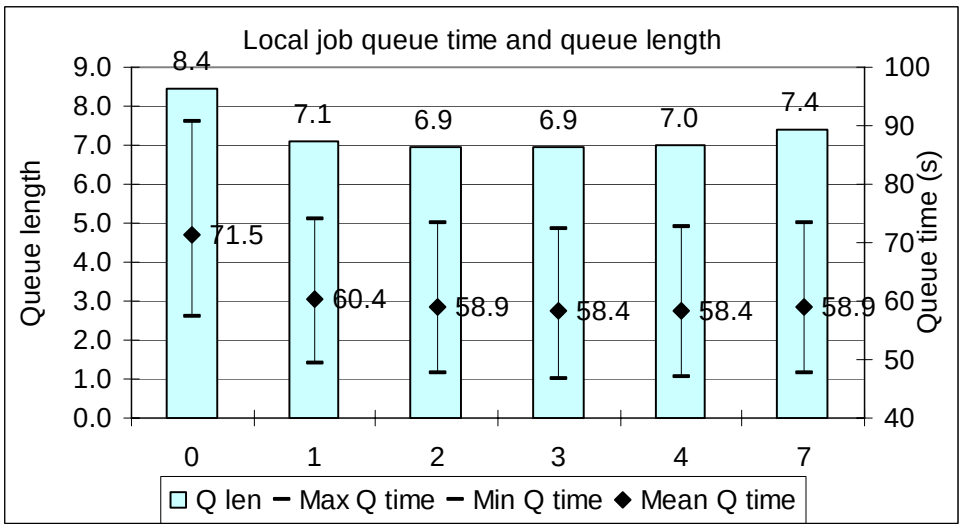


Resource co-allocation









Conclusion

- **When workload is low**
  - The available resource scheduling scheme has the best performance, no task duplication is required.
- **When workload is high (all computing sites are busy)**
  - Random select is worse than others
    - The cost of a bad scheduling decision is very high.
  - The queue length and estimate queue time scheme achieve similar performance.
  - Two or three duplications can reduce the average task queuing time by a factor of 3 to 5

    ✓ No negative impact on local job queuing systems or local jobs