The Gain of Resource Delegation in Distributed Computing Environments

Alexander Fölling, Christian Grimme,
Joachim Lepping, and Alexander Papaspyrou

15th Workshop on Job Scheduling for Parallel Processing
April 23, 2010 - Atlanta, GA, USA

Outline

- Motivation
- System Model
- Resource Delegation Policy
- Evaluation
  - Setup
 - Results
- Conclusion and Future Work

Motivation

- Distributed computing infrastructures (DCI) have reached production status
    - More and more users draw its computing resources from Grid and Cloud infrastructures
    - Many DCIs are exhaustively used and produce significant revenue
- Cloud-Infrastructures allow easy on-demand provisioning of resources (enlargement of local resource space)
    - Infrastructure as a Service (IaaS) by virtualization technology
    - Simple access and pricing model
- The temporal extension of the local resource space allows more flexible scheduling decisions
    - Locally, no traditional parallel job scheduling problem with parallel machines (P_m, R_m, Q_m - Model)
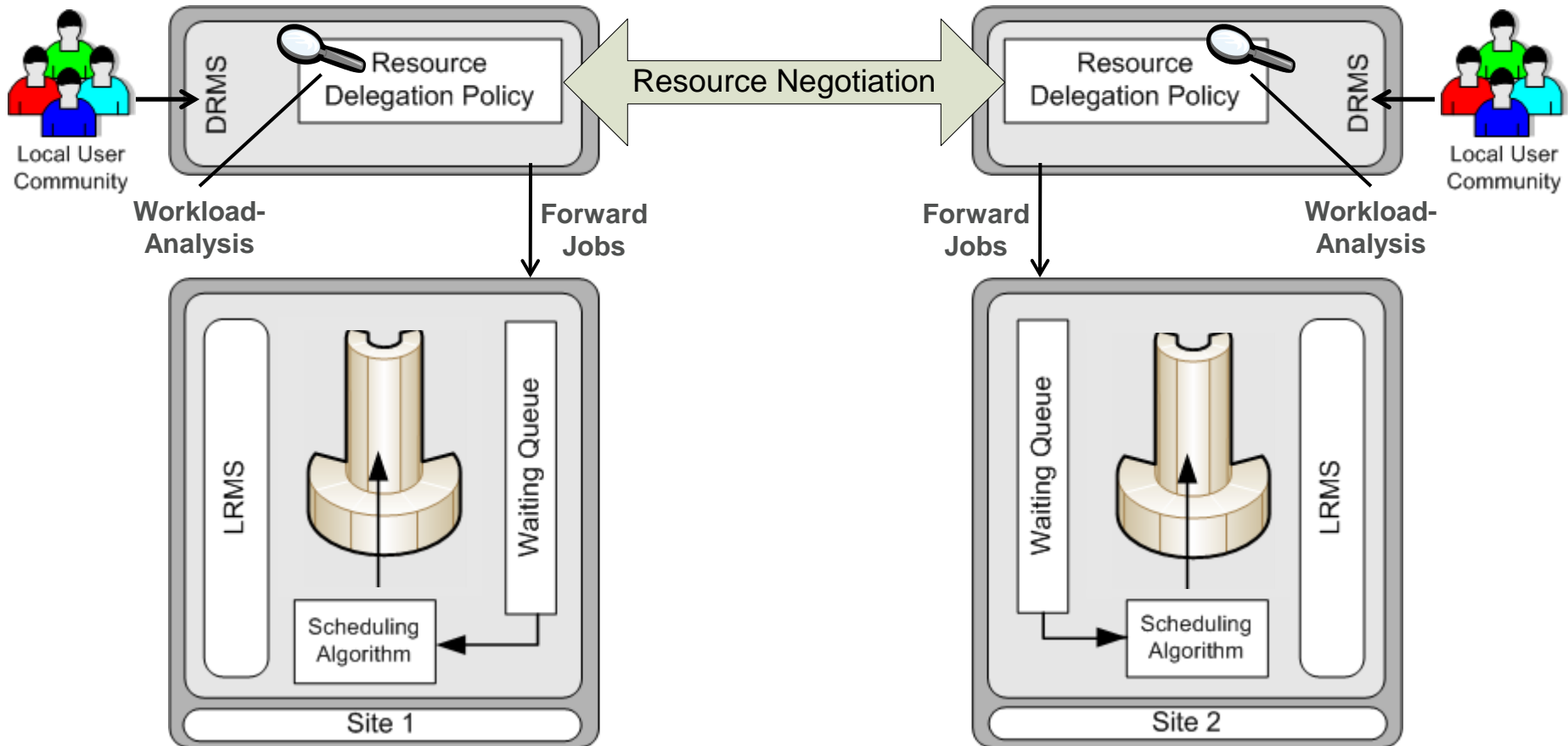   - On-demand resource leasing may improve scheduling performance
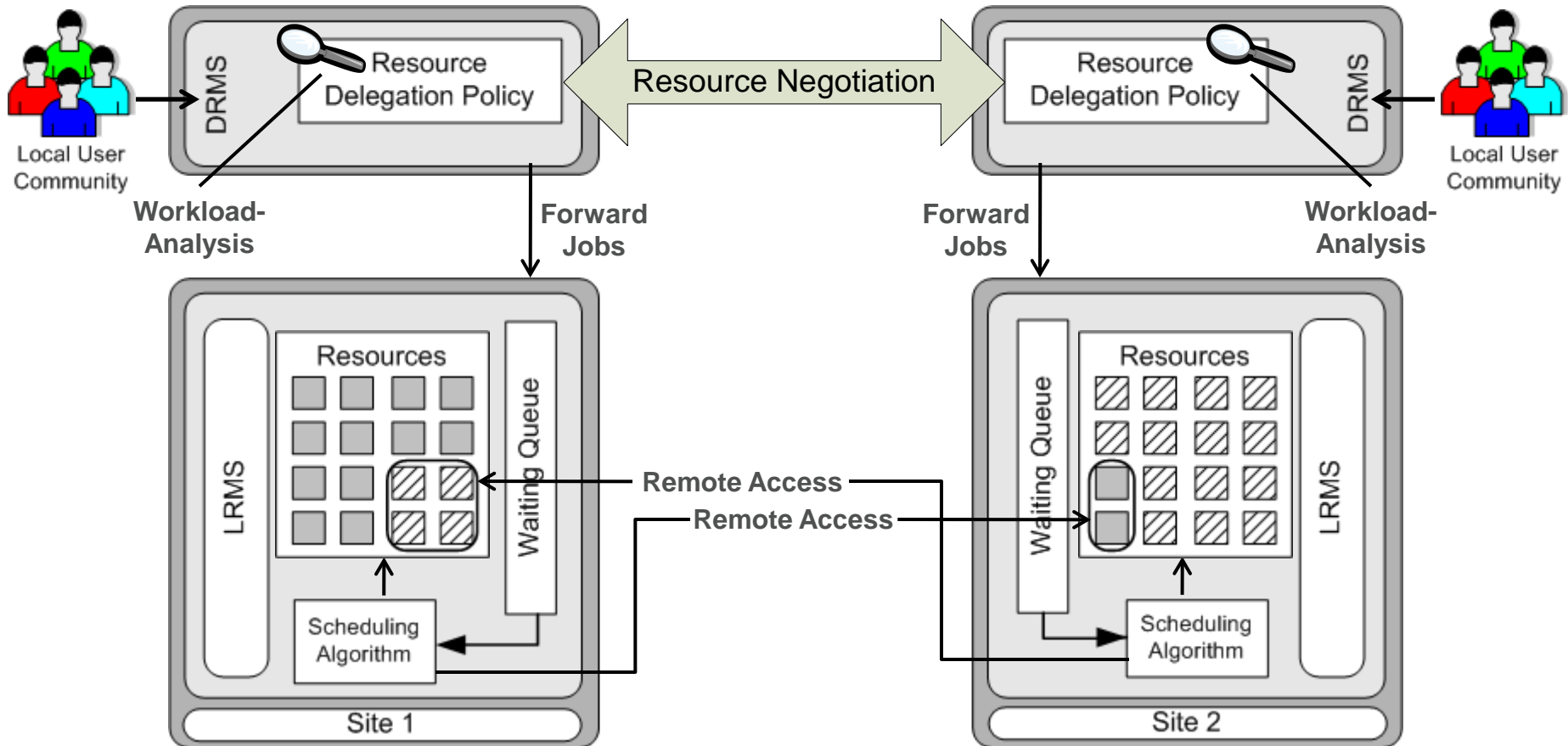
System Model
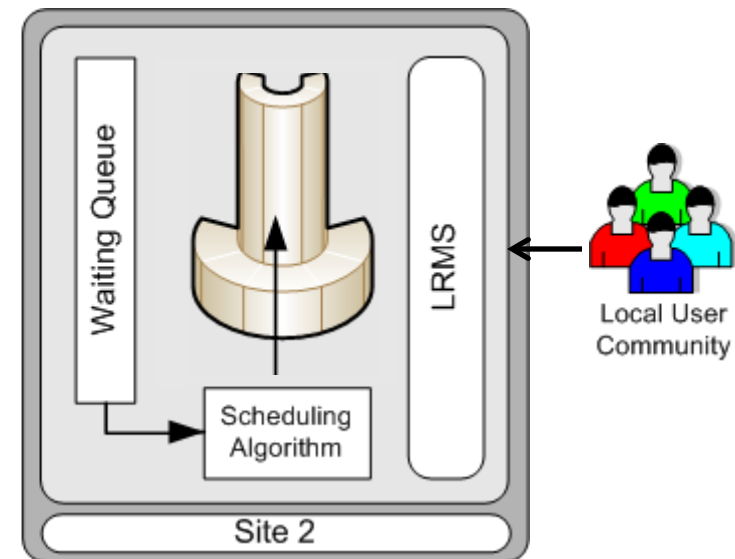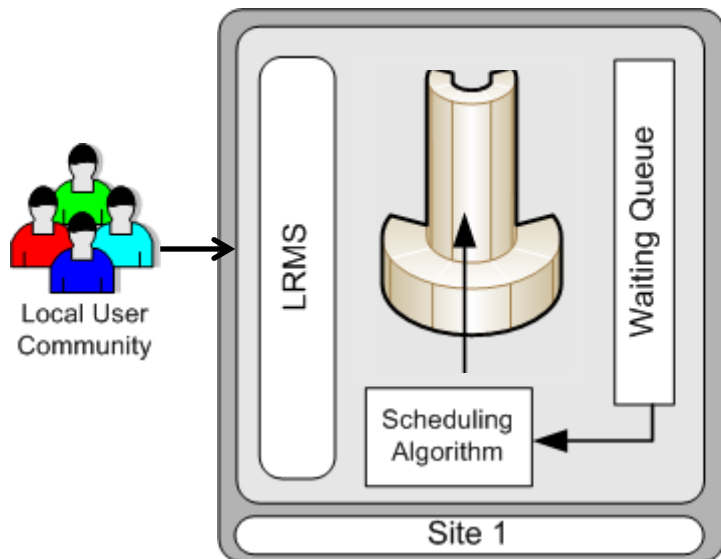
System Model
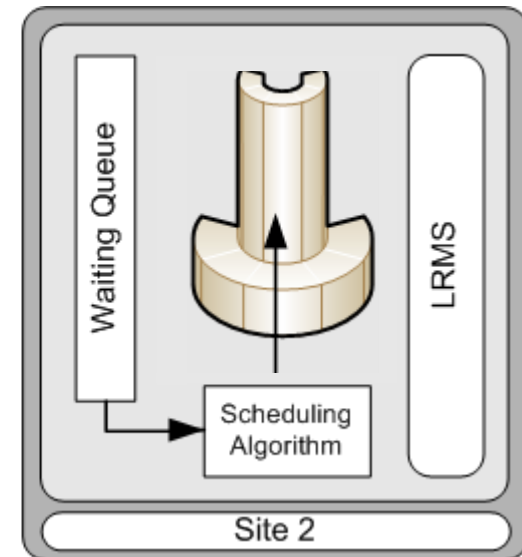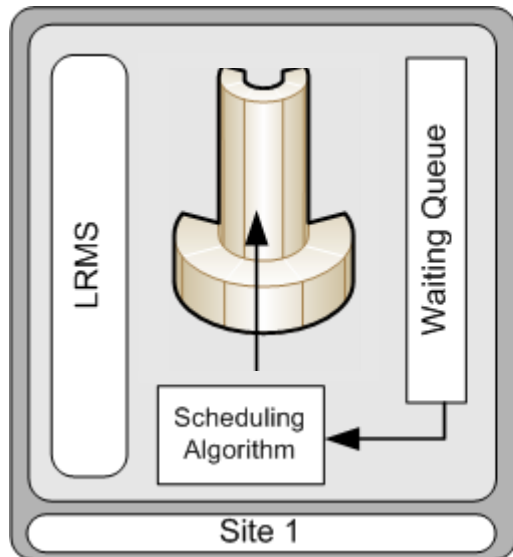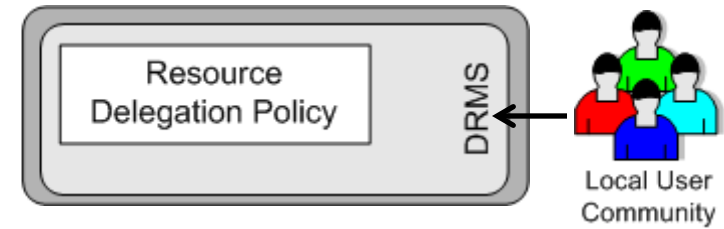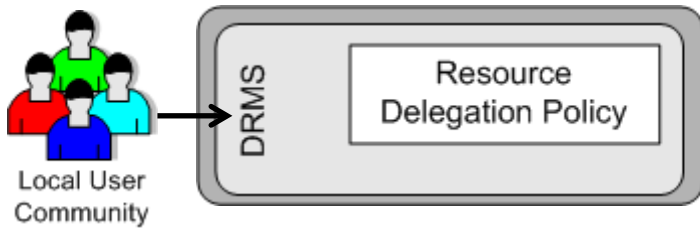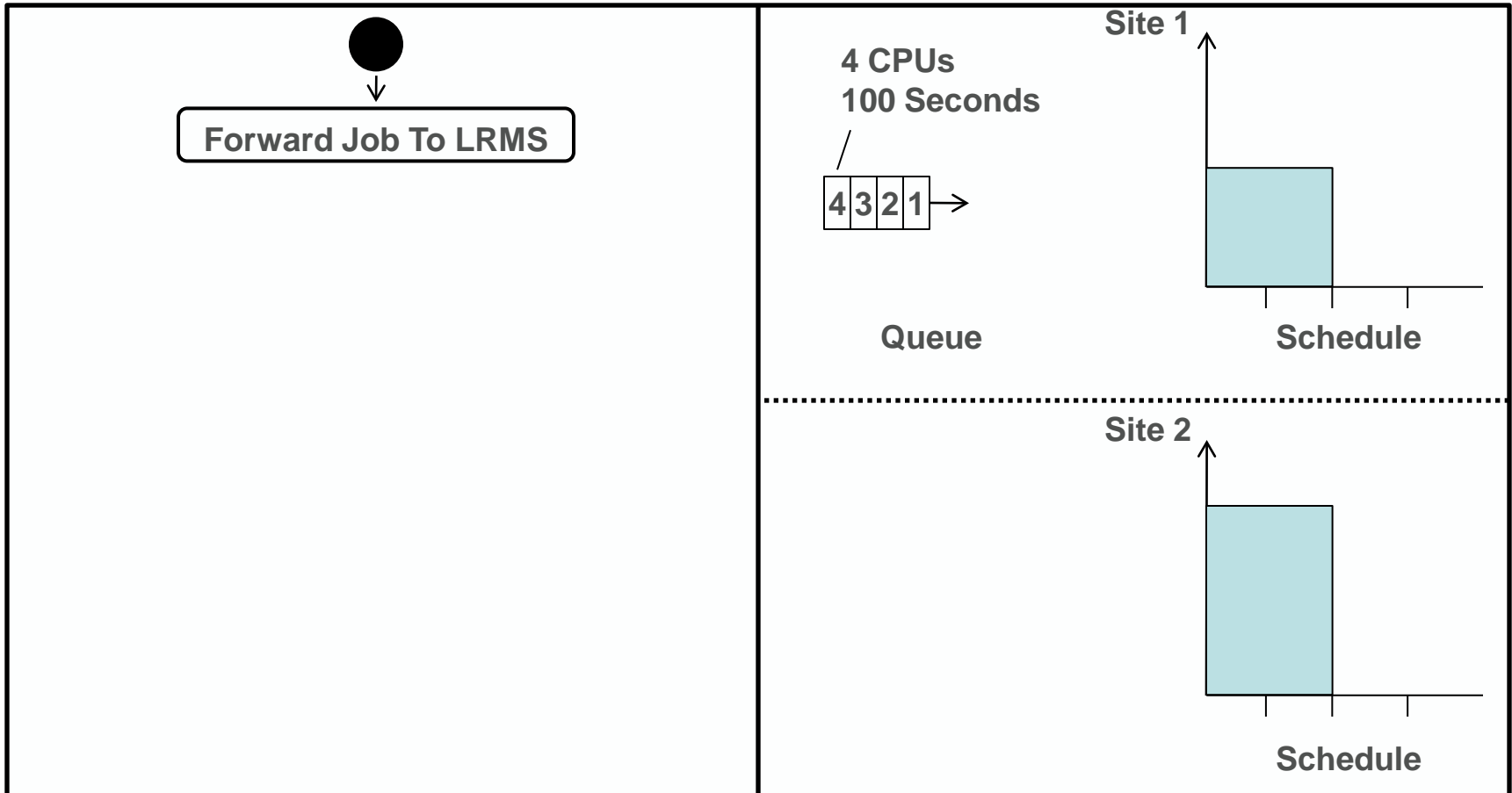
System Model

System Model

Properties of Resource Delegation

- Different from centralized scheduling with multi-site execution
  - No *central scheduling component* but independent sites
  - Scheduler cedes full control to other schedulers when resource access is granted (for a certain period)
- Resource leasing enlarges the local resource space
  - Scheduling decisions are exclusively made by local schedulers
  - Resources might be used immediately or later during leasing period
- Advanced scheduling strategies may support both
  - local allocations under varying machine sizes
  - planning of future resource requirements
- Each participant in a DCI is both resource consumer and resource provider

Submission Triggered Resource Delegation Policy (ST-RDP)
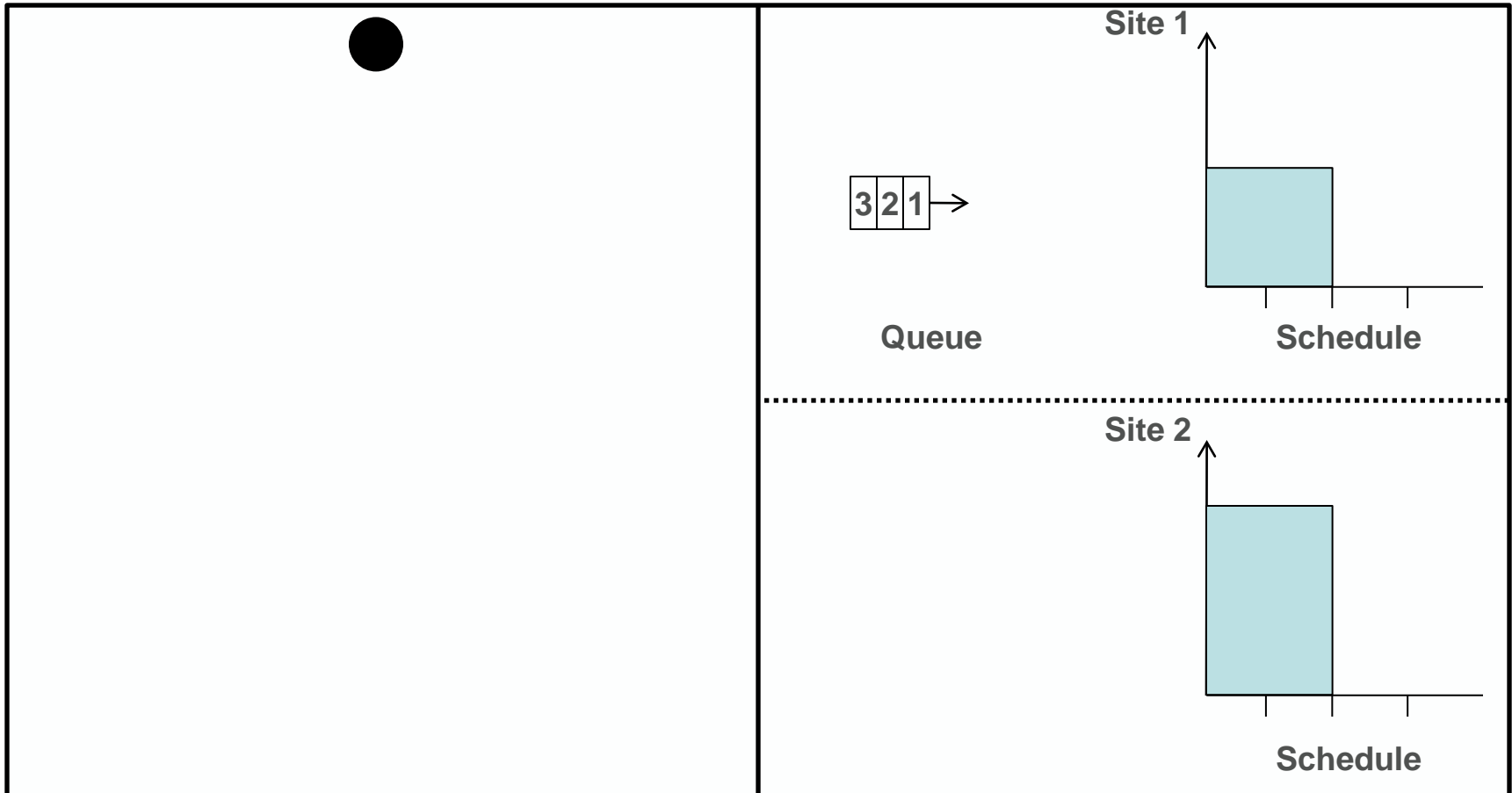
Submission Triggered Resource Delegation Policy (ST-RDP)

Forward Job To LRMS

4 CPUs
100 Seconds

4 3 2 1

Queue

Site 1

Schedule

Site 2

Schedule

Submission Triggered Resource Delegation Policy (ST-RDP)

Forward Job To LRMS

Check Resource Availability

4 CPUs
100 Seconds

4 3 2 1

Queue

Site 1

2 CPUs idle

Schedule

Site 2

Schedule

Submission Triggered Resource Delegation Policy (ST-RDP)

Forward Job To LRMS

Check Resource Availability

[Enough]

**4 CPUs
100 Seconds**

4 3 2 1

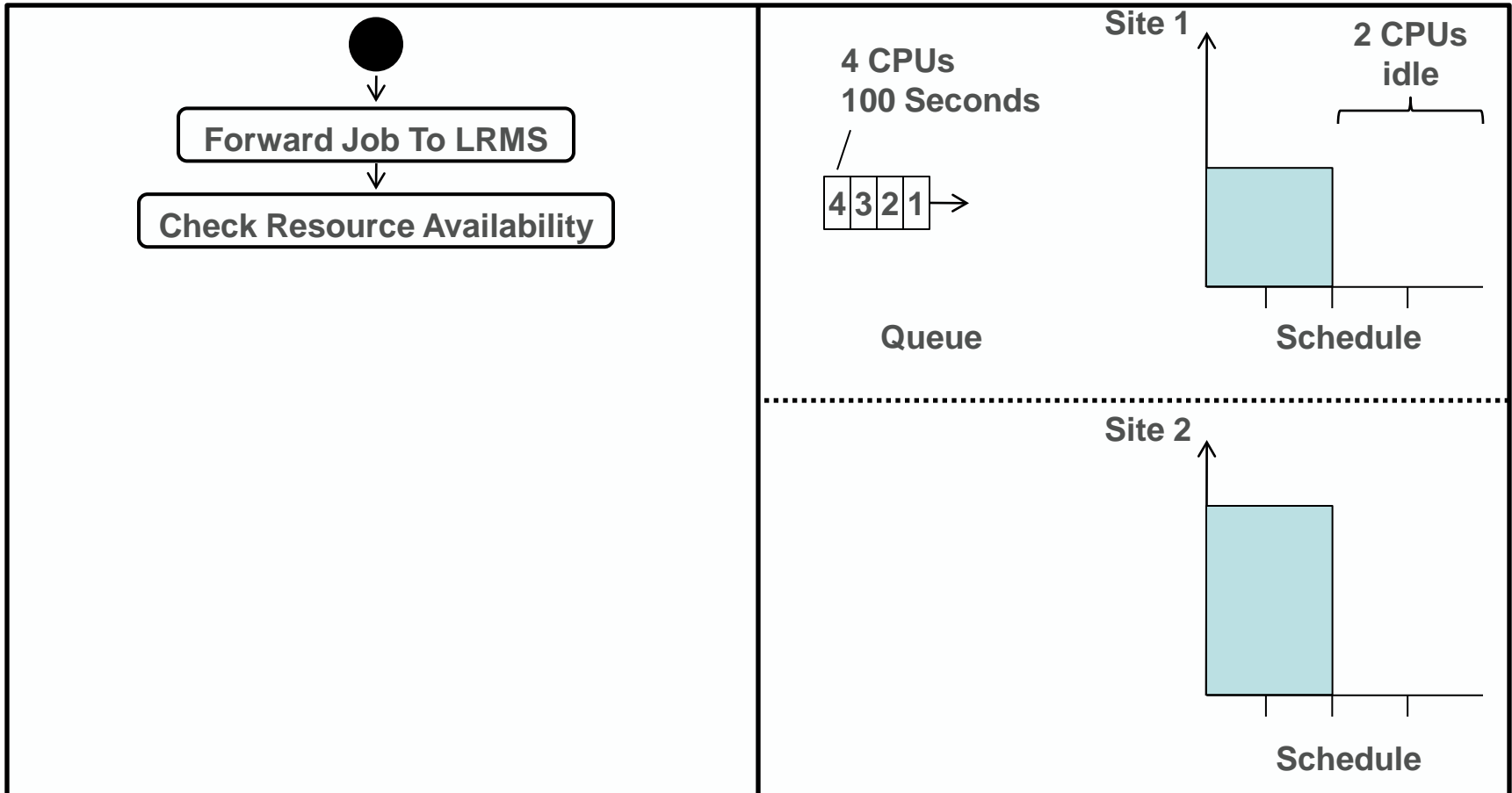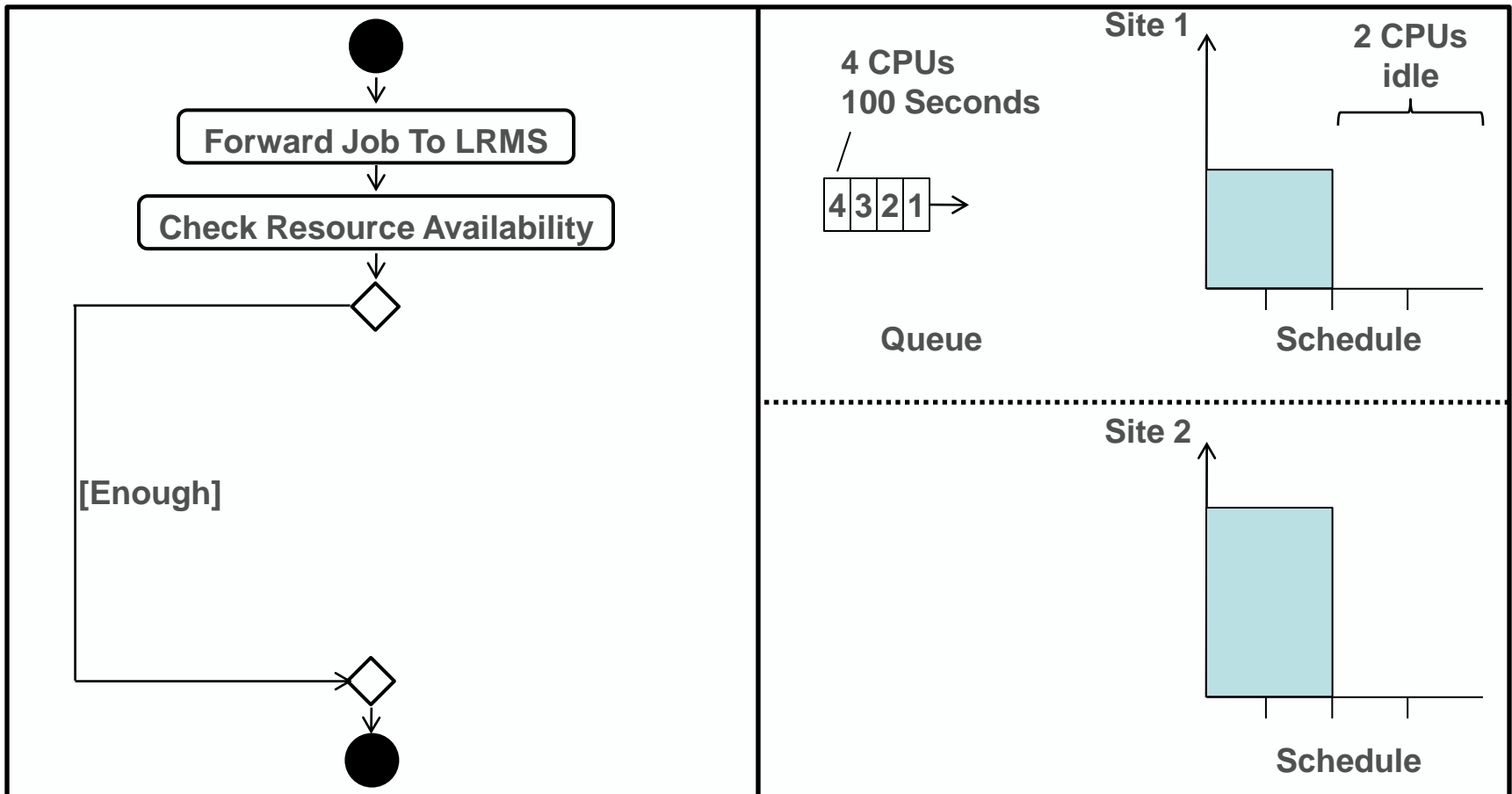**Queue**

Site 1

**2 CPUs
idle**

Schedule

Site 2

Schedule

Submission Triggered Resource Delegation Policy (ST-RDP)

Submission Triggered Resource Delegation Policy (ST-RDP)

Forward Job To LRMS
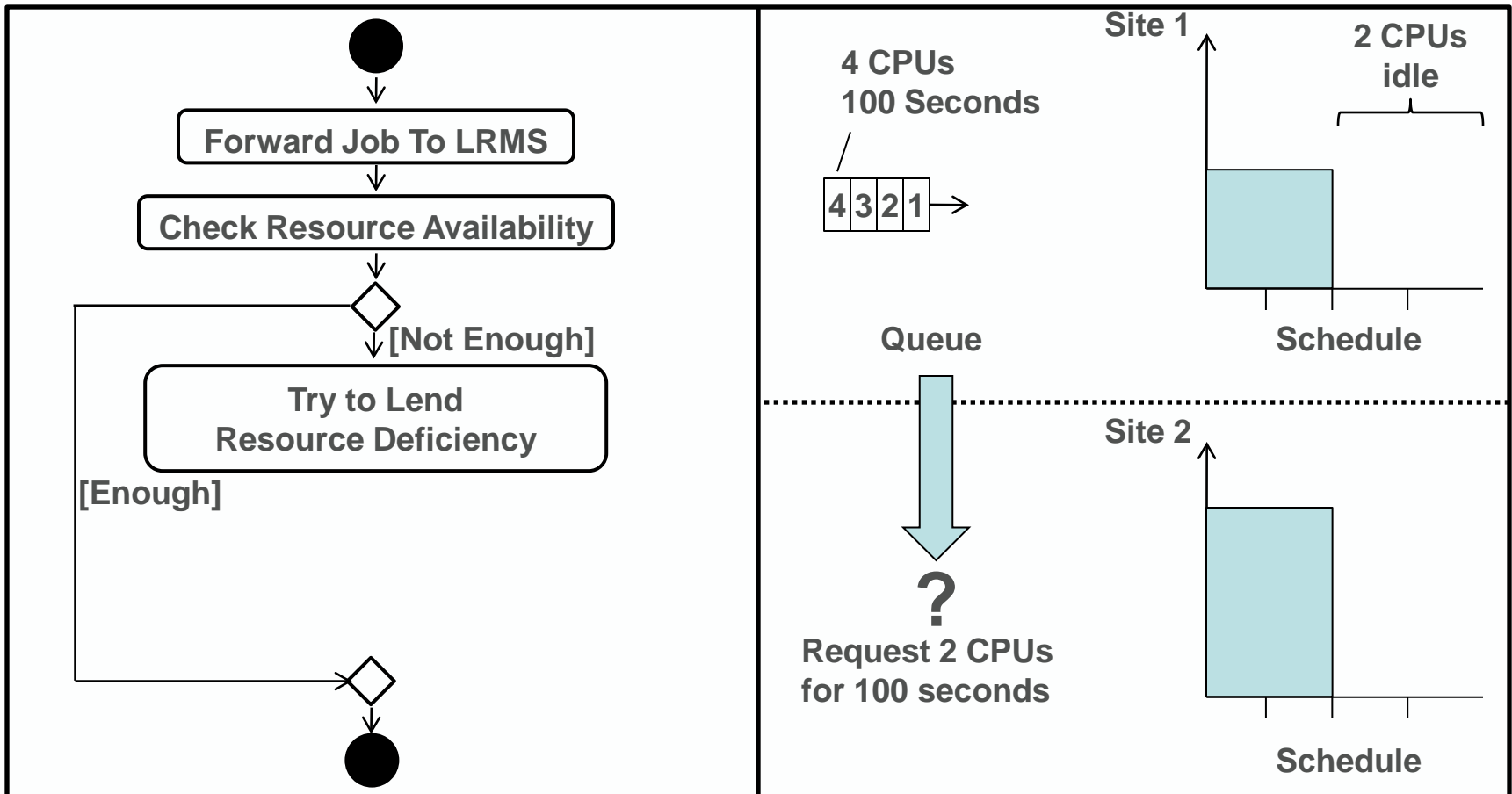
Check Resource Availability

[Not Enough]

Try to Lend Resource Deficiency

[Enough]

[Request Denied]

4 CPUs
100 Seconds

4 3 2 1 →

Site 1

2 CPUs idle

Queue

Schedule

Site 2

?

Request 2 CPUs for 100 seconds

Schedule

Submission Triggered Resource Delegation Policy (ST-RDP)

Submission Triggered Resource Delegation Policy (ST-RDP)

Evaluation Setup
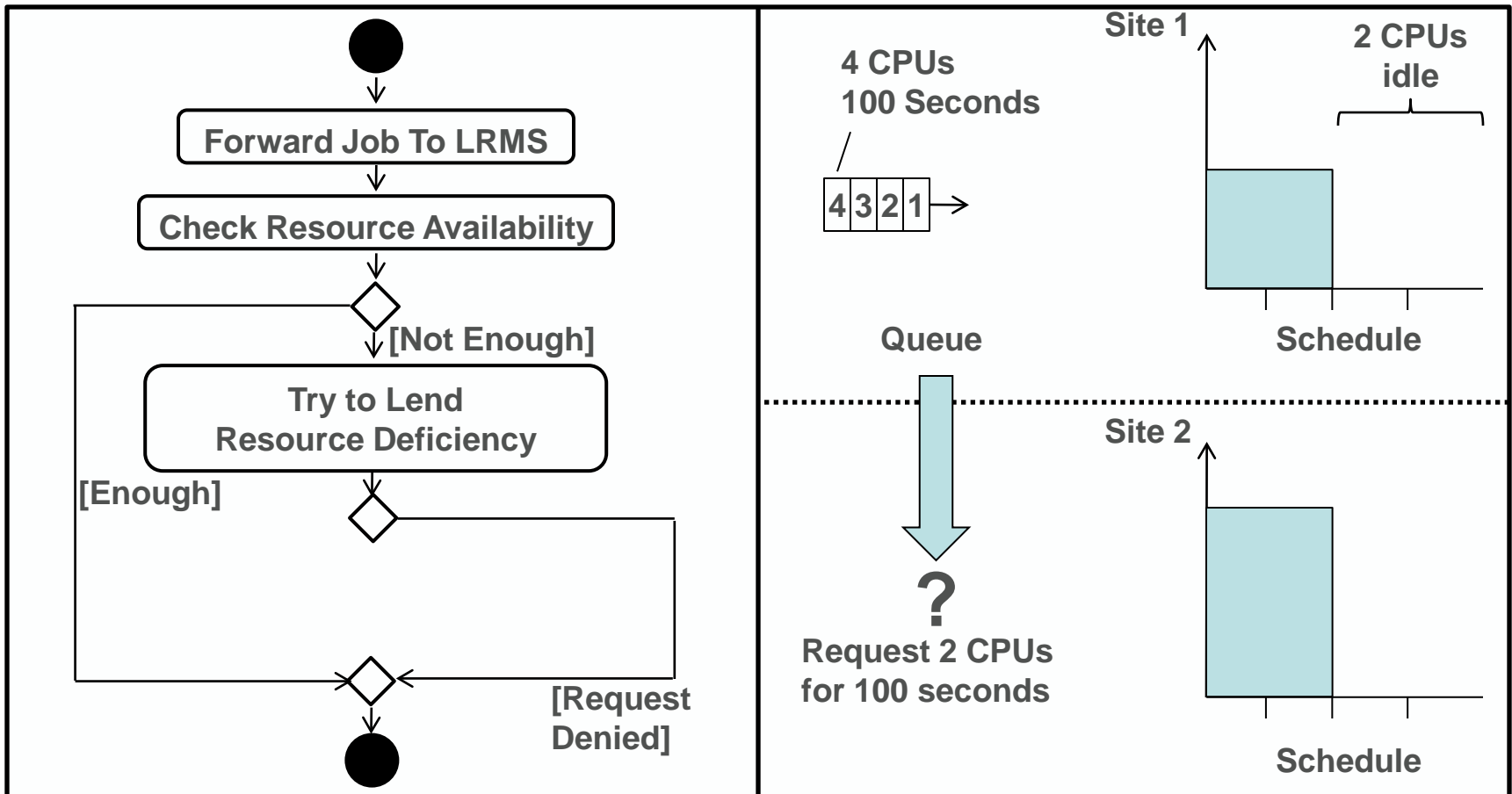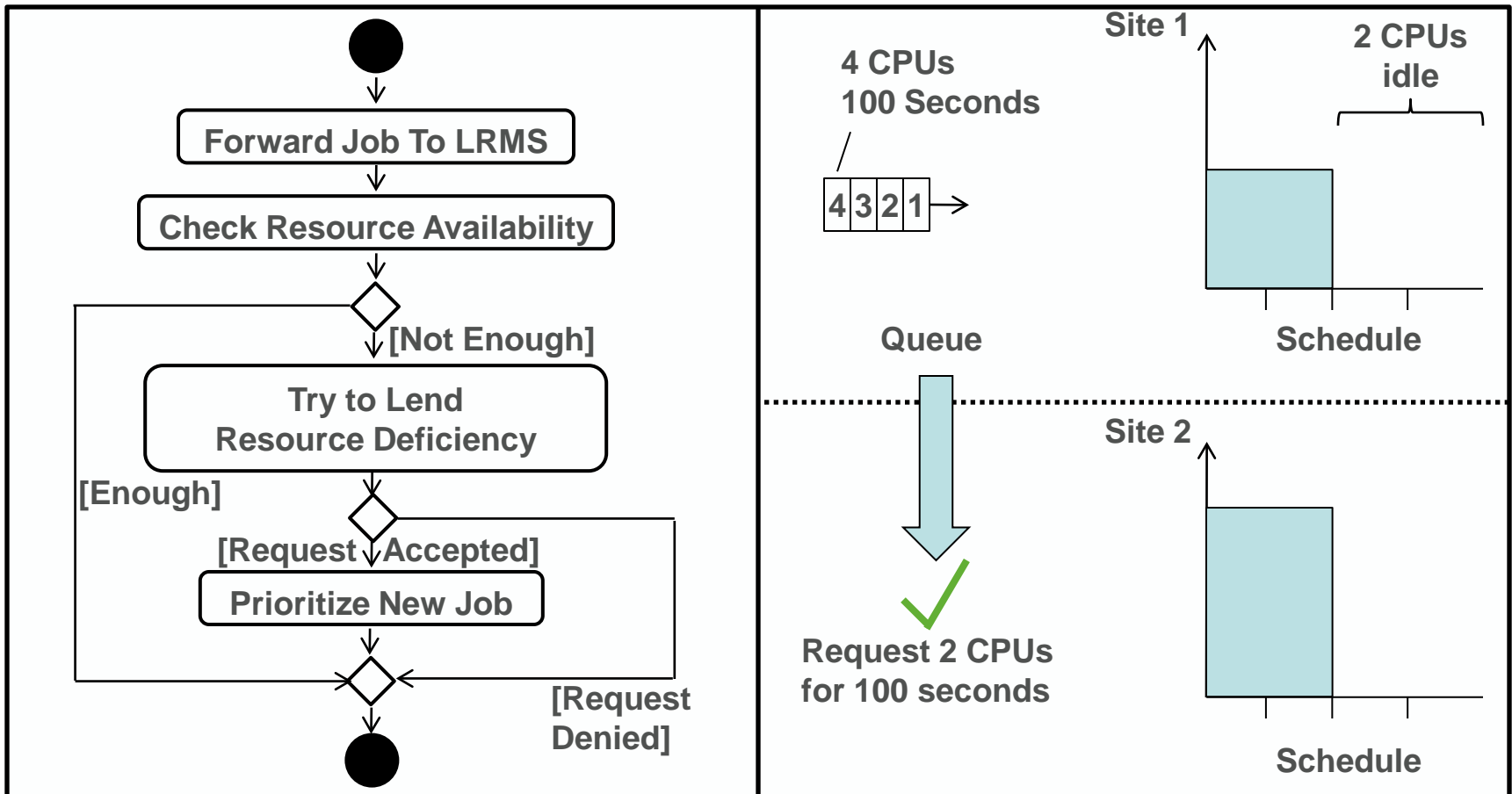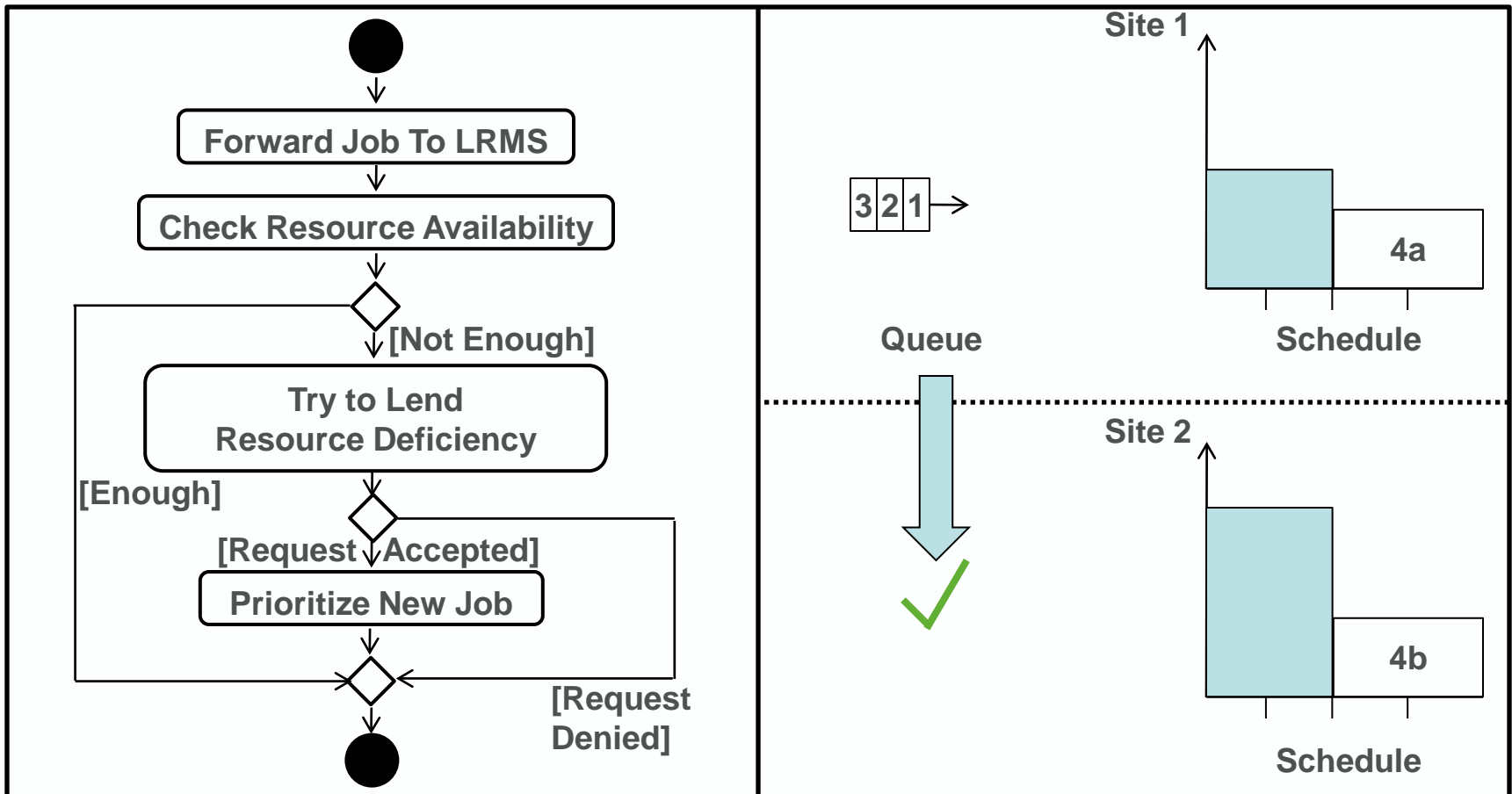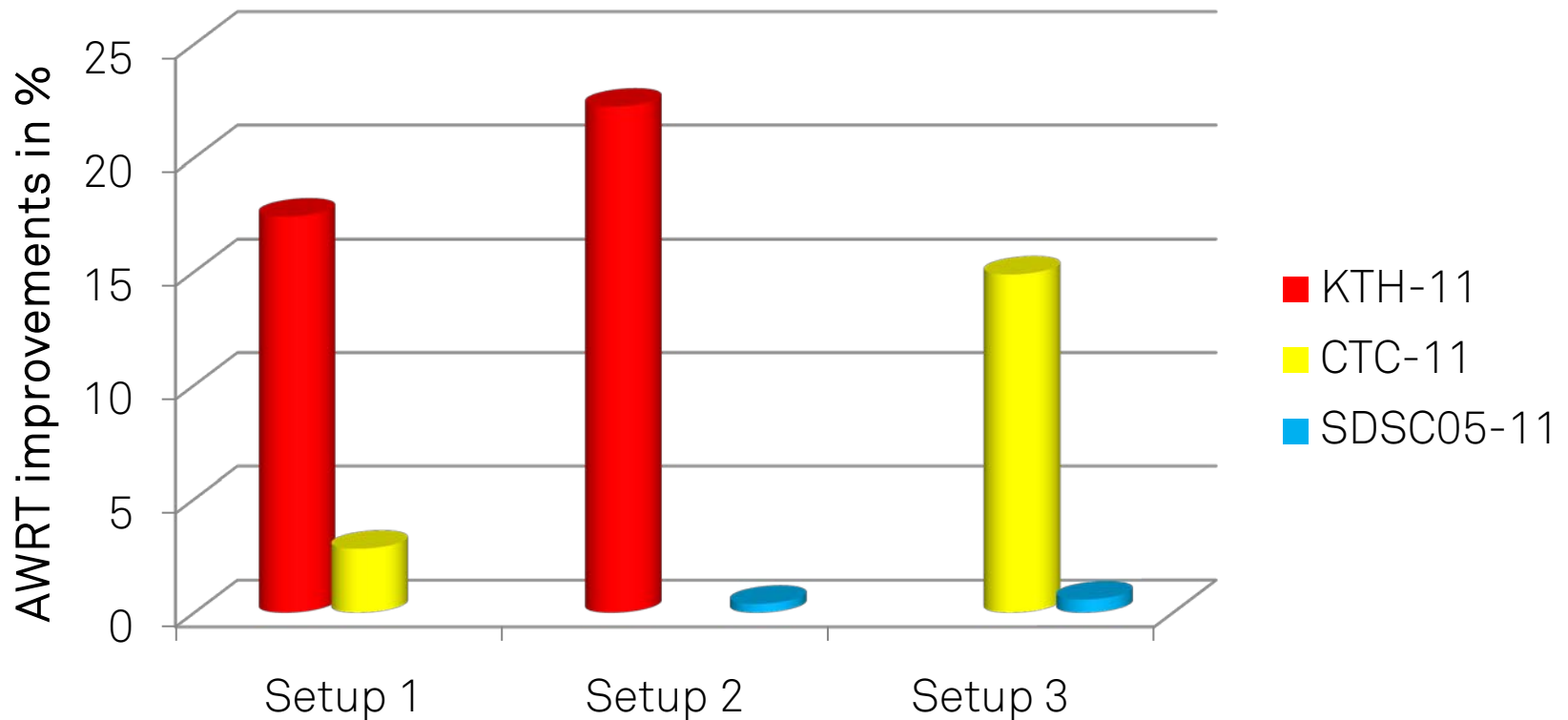
- Input Data
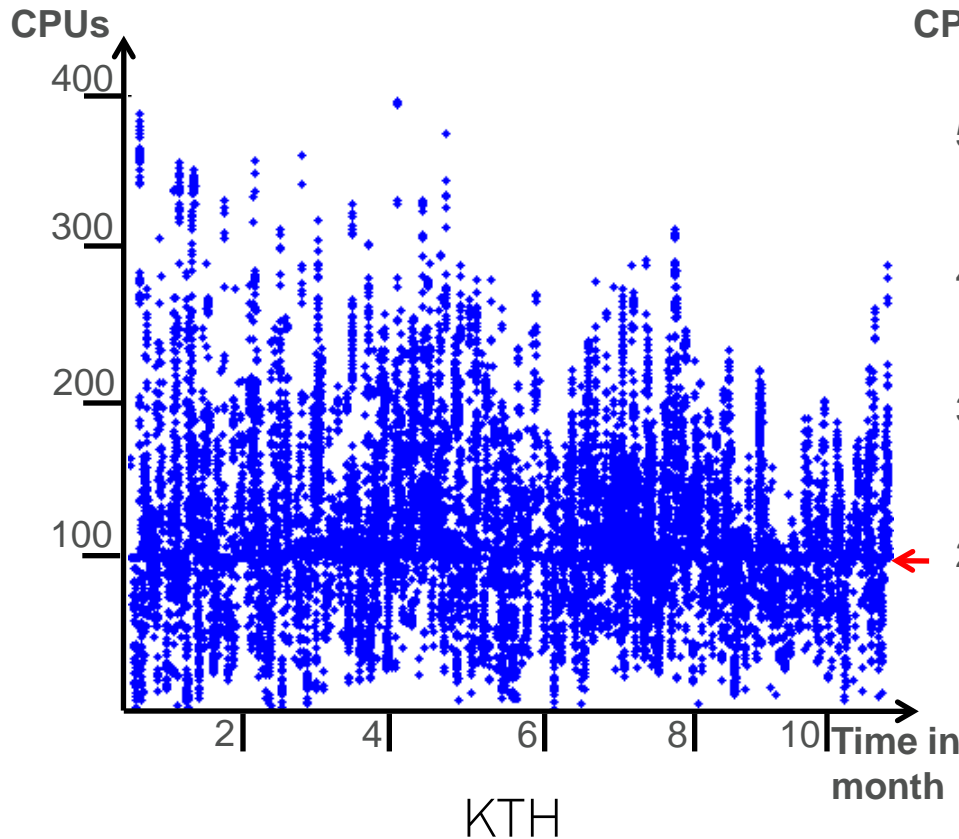 - Real Workload Traces from Parallel Workloads Archive
    - KTH, CTC, SDSC05
    - ~ 100 – 1600 CPUs, ~ 28000 – 74000 Jobs (first 11 months)
- Local Resource Management System
  - EASY Backfilling
- Evaluation objectives for results
  - Improvements in AWRT
  - Reconfiguration behavior

Results: ST-RDP Performance

Results: Reconfiguration behavior

- KTH and CTC 11 month with ST-RDP

KTH

CTC
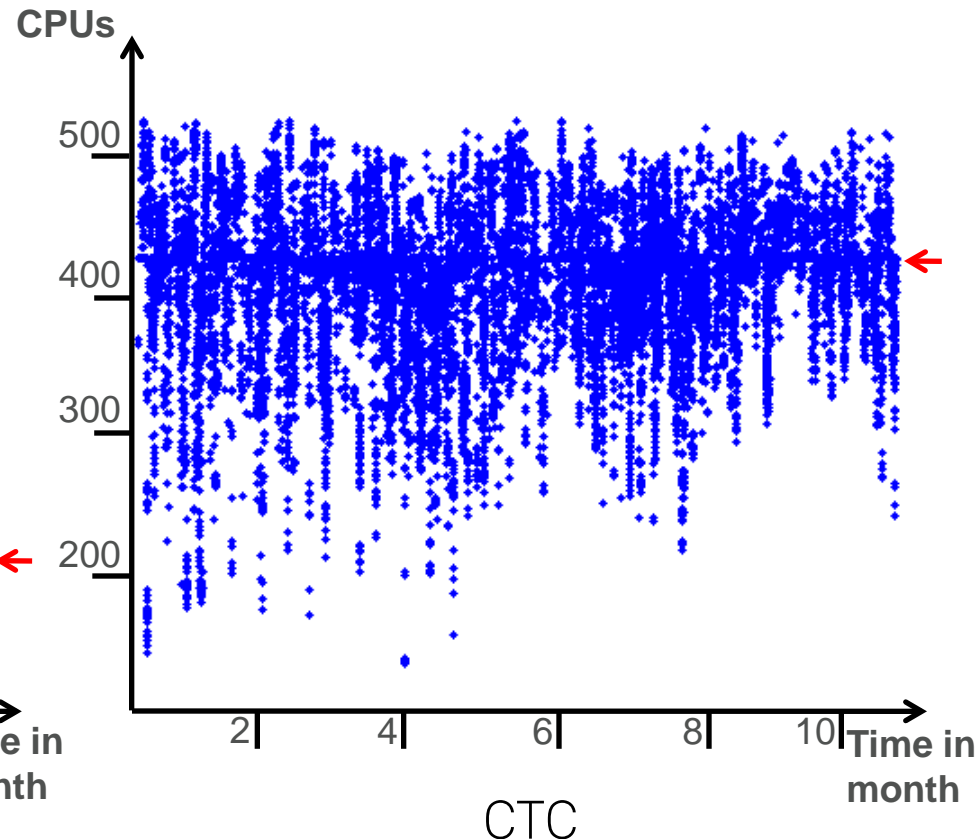
Conclusion

- Proposed new concept for resource delegation in DCIs
  - Parallel job scheduling problems under varying machine sizes
  - The resource requirements can be flexibly negotiated among participants
- Evaluation of a simple resource delegation method
  - Without need for further information exchange
  - Robust in changing environments
- Results show significant benefits for the local scheduling (improvement in AWRT)
  - During operation, many resources are delegated among sites

Future Work

- Application to larger DCI environments
  - Considering additional location policies that decides which site to ask first for delegation

- Long term planning of resource leasing/delegation
  - Not only single job decisions
  - Decisions should be based on workload records (user behavior, submission patterns etc.)
  - Eventually, make decision on predicted user behavior
  - Consider additional parameters like local queue/schedule status

Thank You

Alexander Fölling
alexander.foelling@udo.edu

Joachim Lepping
joachim.lepping@udo.edu

Christian Grimme
christian.grimme@udo.edu

Alexander Papaspyrou
alexander.papaspyrou@udo.edu

Robotics Research Institute
Information Technology Section
TU Dortmund University, Germany

http://www.it.irf.de