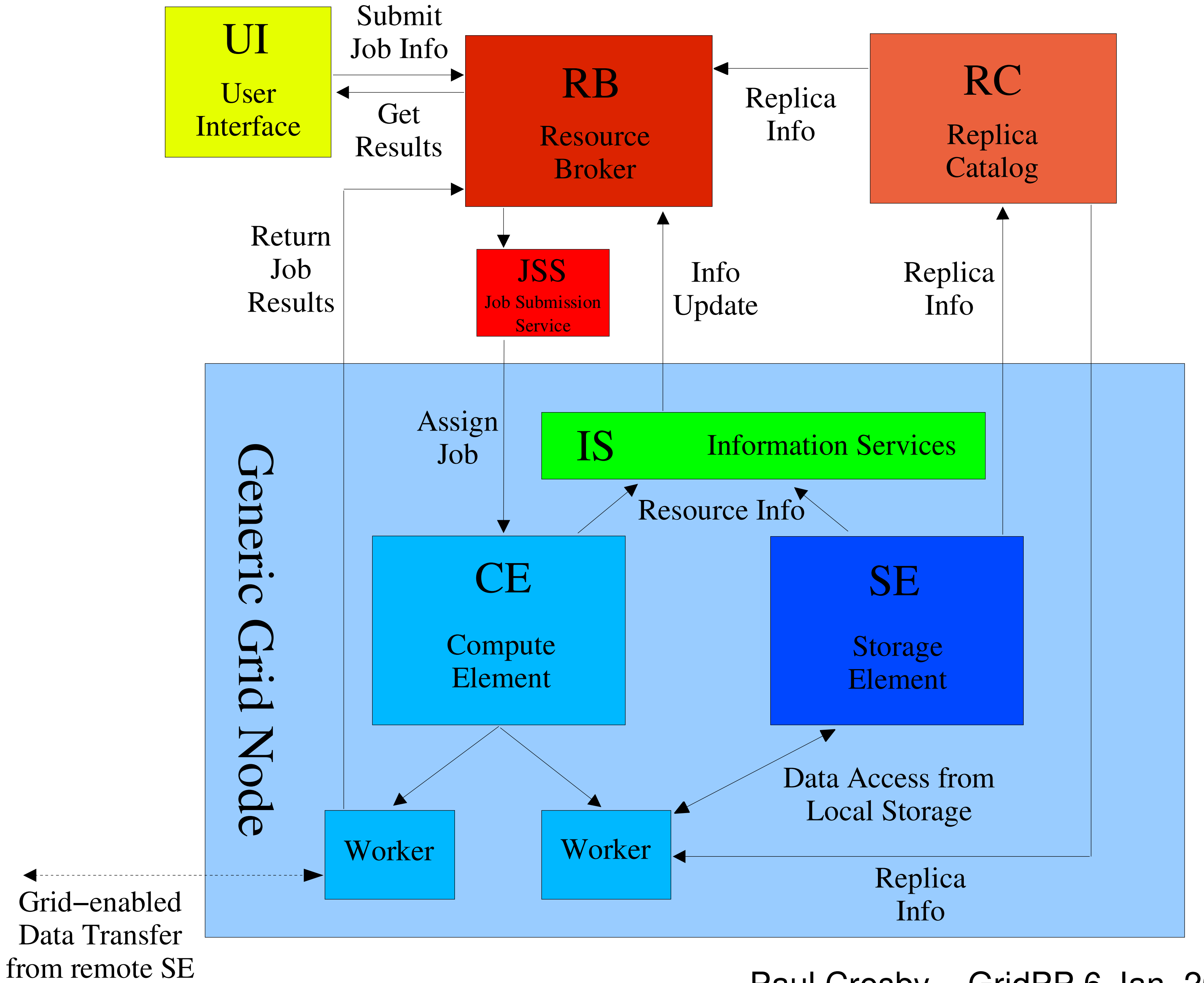
EDGSim
Simulating a HEP GRID

Paul Crosby
pac@hep.ucl.ac.uk
University College London

- Job Scheduling in a HEP Grid
- Simulating the European Data Grid
- Simulated Job Scheduling

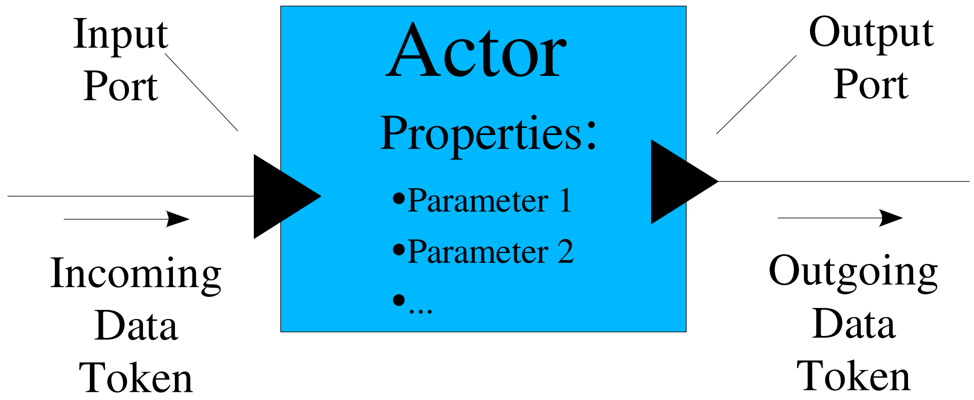


Paul Crosby – GridPP 6 Jan. 2003

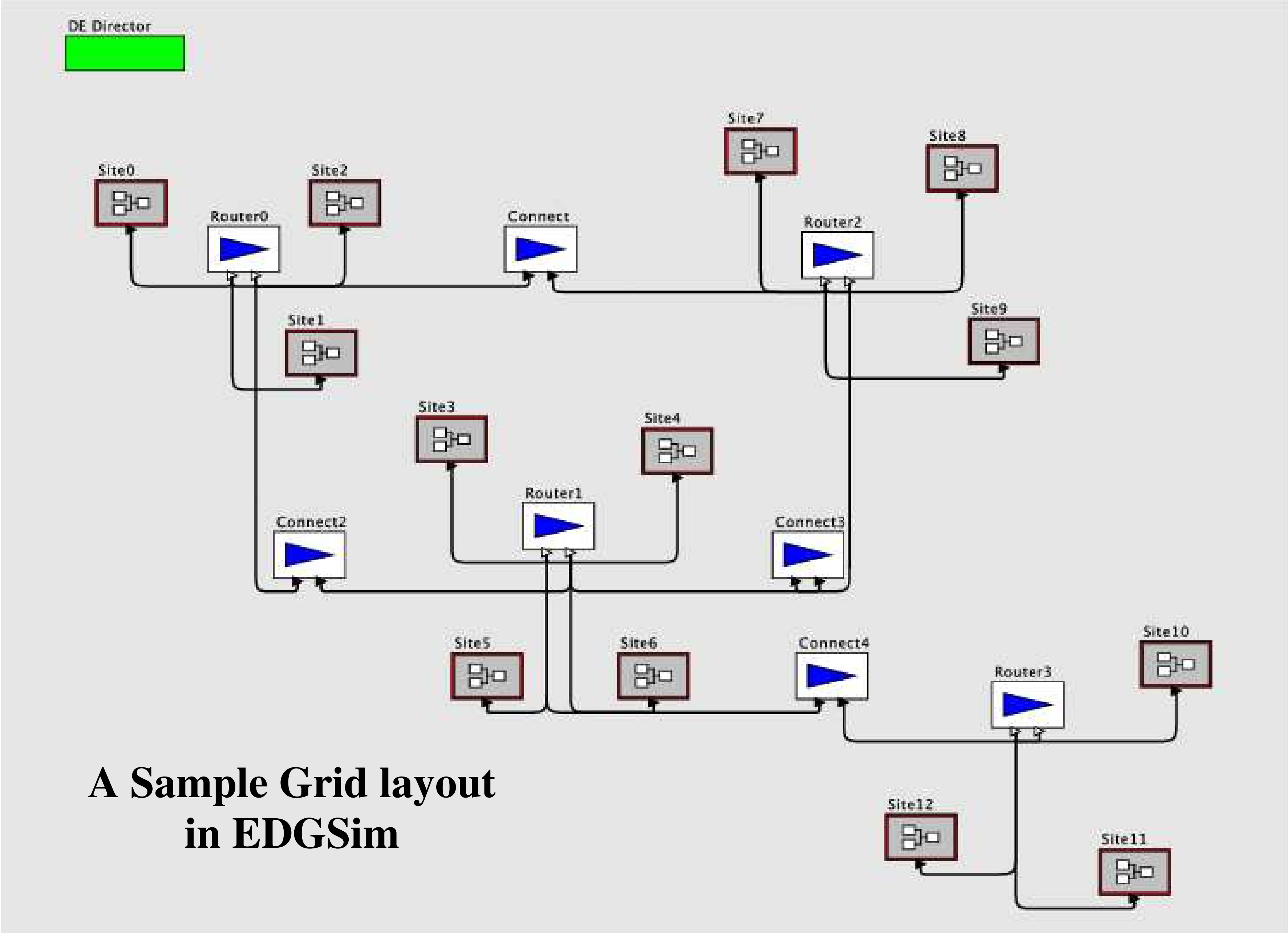
Job Scheduling and the Resource Broker

- Resource Broker gets information about:
  - Computational resources
  - Storage resources
  - Replicas of data
- Uses this info to choose "best" resource for a job
- Potential conflict: Process jobs as quickly as possible, or optimise use of resources?
- Who makes the choice?
  - User describes resource preferences (ClassAds)
  - RB has scheduling policy (Grid accounting?)

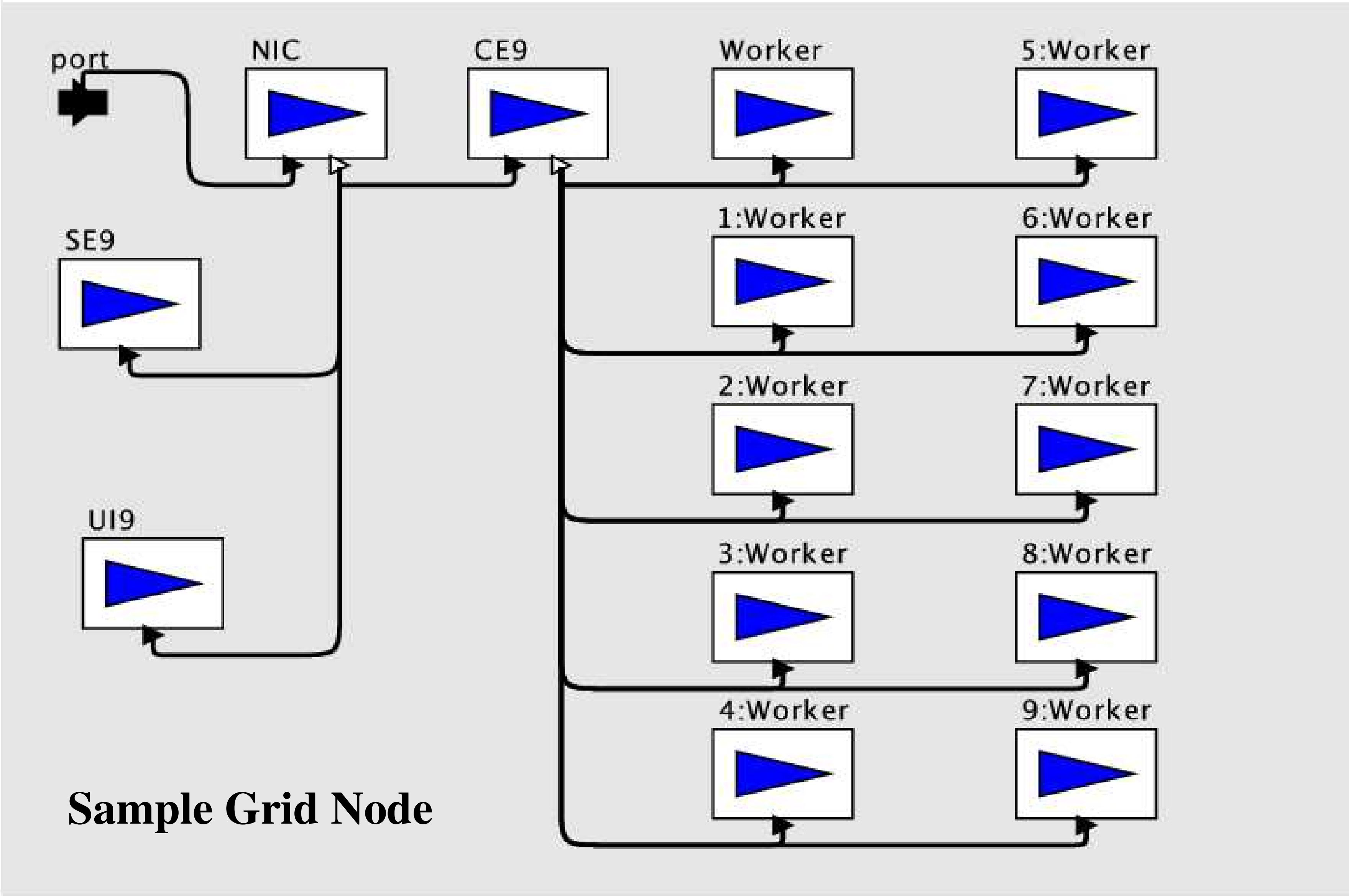
Building EDGSim with Ptolemy II



- Entities represented by "Actor" objects
- Actors communicate by passing data Tokens
- Simulation events time–stamped and processed in chronological order (the Discrete Event model)
- Actors assembled & sim run from GUI



A Sample Grid layout in EDGSim


port
NIC
CE9
Worker
5:Worker
1:Worker
6:Worker
SE9
2:Worker
7:Worker
UI9
3:Worker
8:Worker
4:Worker
9:Worker
Sample Grid Node

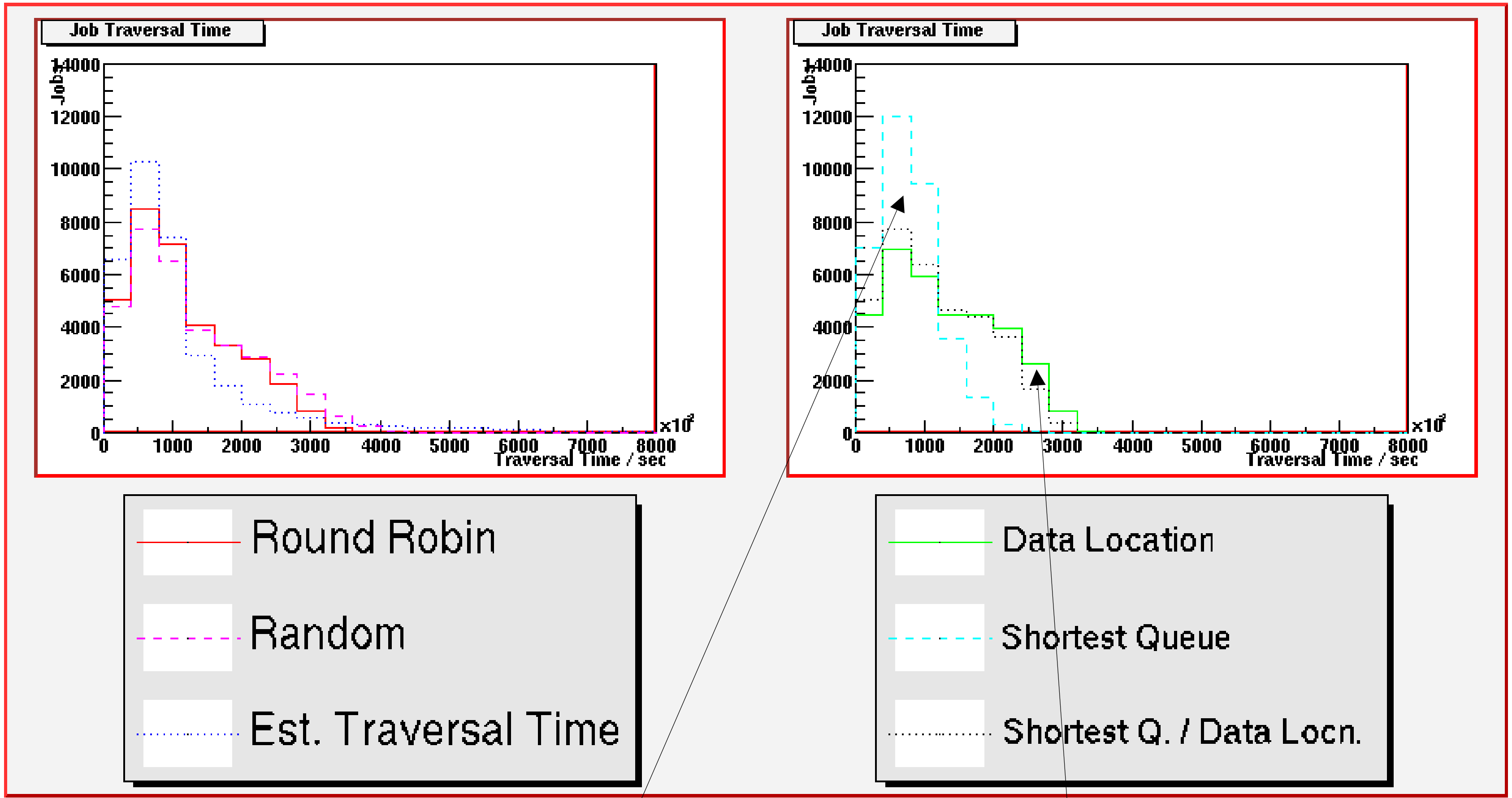
Running EDGSim

- Job parameters & distribution chosen
- Grid topology decided (Number of nodes, CE / SE resources at each node, initial data file distribution)
- Scheduling algorithm for Resource Broker chosen (rather than real–life ClassAd matching)
- Jobs generated for a given period of time, then simulation runs until all jobs have been processed

Scheduling Policies

- Base scheduling on dynamic information about:
  - Estimated Traversal Time
  - Queue length
  - Location of required data
  - Combinations of the above
- Measure performance with:
  - Individual job processing times
  - Overall job throughput
  - Resource utilisation (network use, idle CPUs etc.)

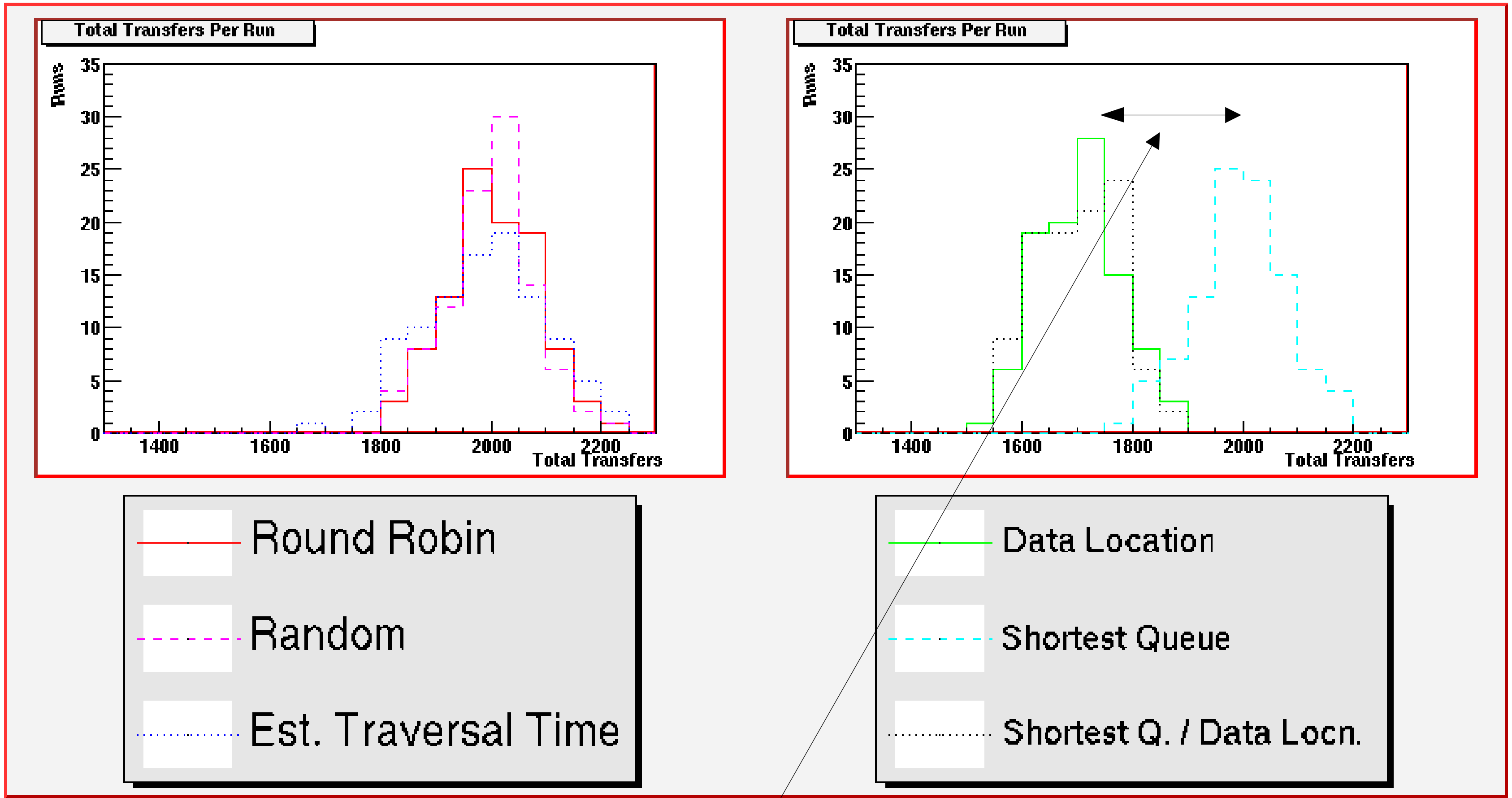
Sample EDGSim Results: Job Traversal Time



Queue length algorithm better & more consistent

Data locn. not doing so well...

Sample EDGSim Results: Network Load



~15% reduction in network load with data locn. algorithm

Summary of Results

- Estimated Traversal Time algorithm unreliable if job lifetime is dominated by job running time
- Min. queue length algorithm provides most stable results
- Combining algorithms can produce good results and use resources more efficiently
- A poor combination (or even complementary algorithms combined badly) can make things worse!

What Next for EDGSim?

- More realistic job load:
  - Job distribution (busy / quiet periods?)
  - Jobs with output (Monte Carlo)
- Better representation of data handling:
  - Network topology
  - data management & optimisation
- Multiple users – complementary / competing scheduling

Verification of Results

- A real Grid testbed (GridPP? SAM Grid?) for comparison
- Set up virtual Grid with similar topology and resources
- Generate a batch of dummy jobs to submit (or even better – real physics jobs?)
- Comparison of results with other simulations (such as OptorSim)

Conclusions

- The structure and flow of information around an EDG style Grid has been simulated using Ptolemy II software
- Several job scheduling policies have been tested using this simulated Grid
- Even a simplistic model of scheduling in a Data Grid reveals useful information about the effectiveness of these policies
- More results at:

  http://www.hep.ucl.ac.uk/~pac/EDGSim/