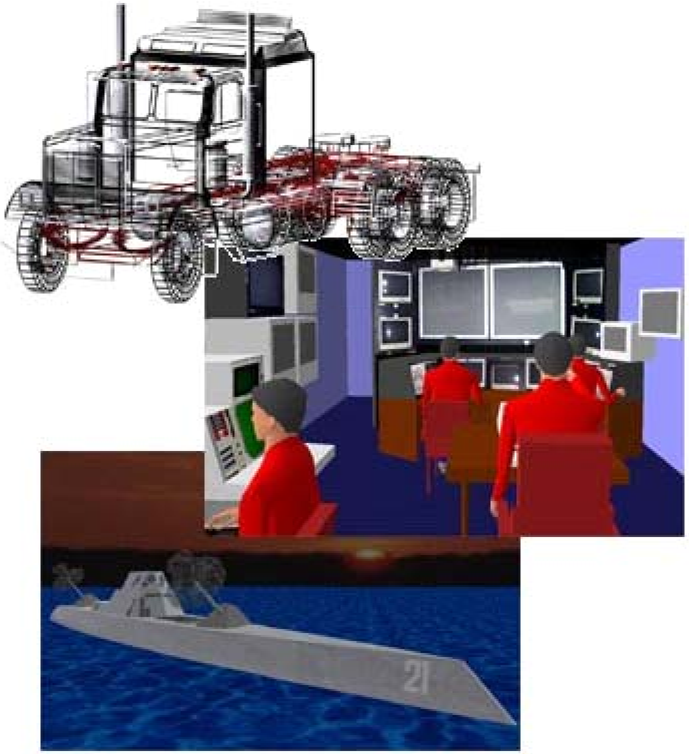
Towards Load Balancing Support for I/O-Intensive Parallel Jobs in a Cluster of Workstations

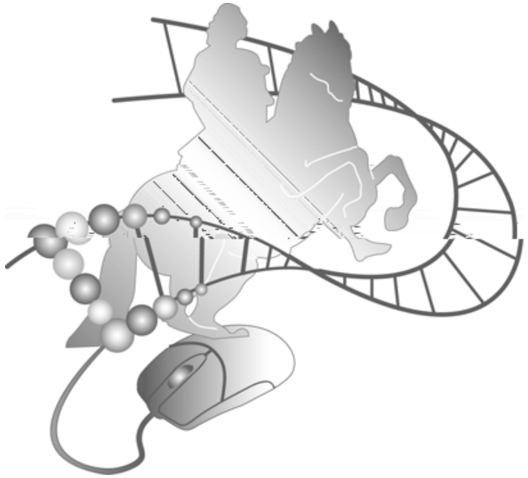
Presented by Dr. Hong Jiang
Department of Computer Science and Engineering
University of Nebraska-Lincoln

I/O-intensive Applications

remote-sensing database systems



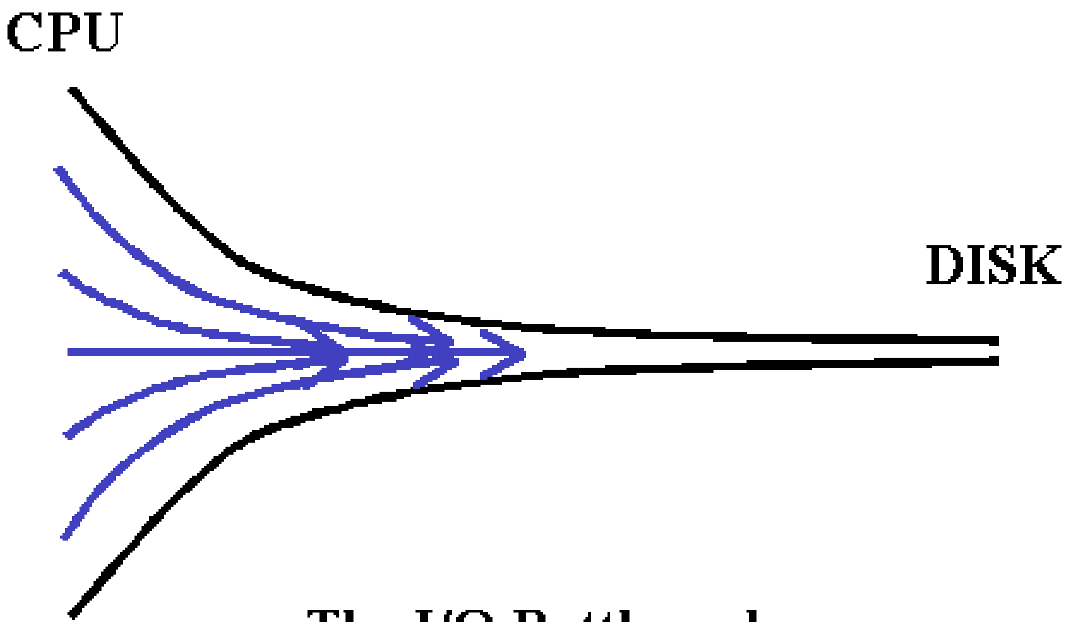
long running simulations



biological sequence analysis

Motivation

- I/O-intensive Applications require input and output of large amounts of data.
- I/O performance can be a potential bottleneck.

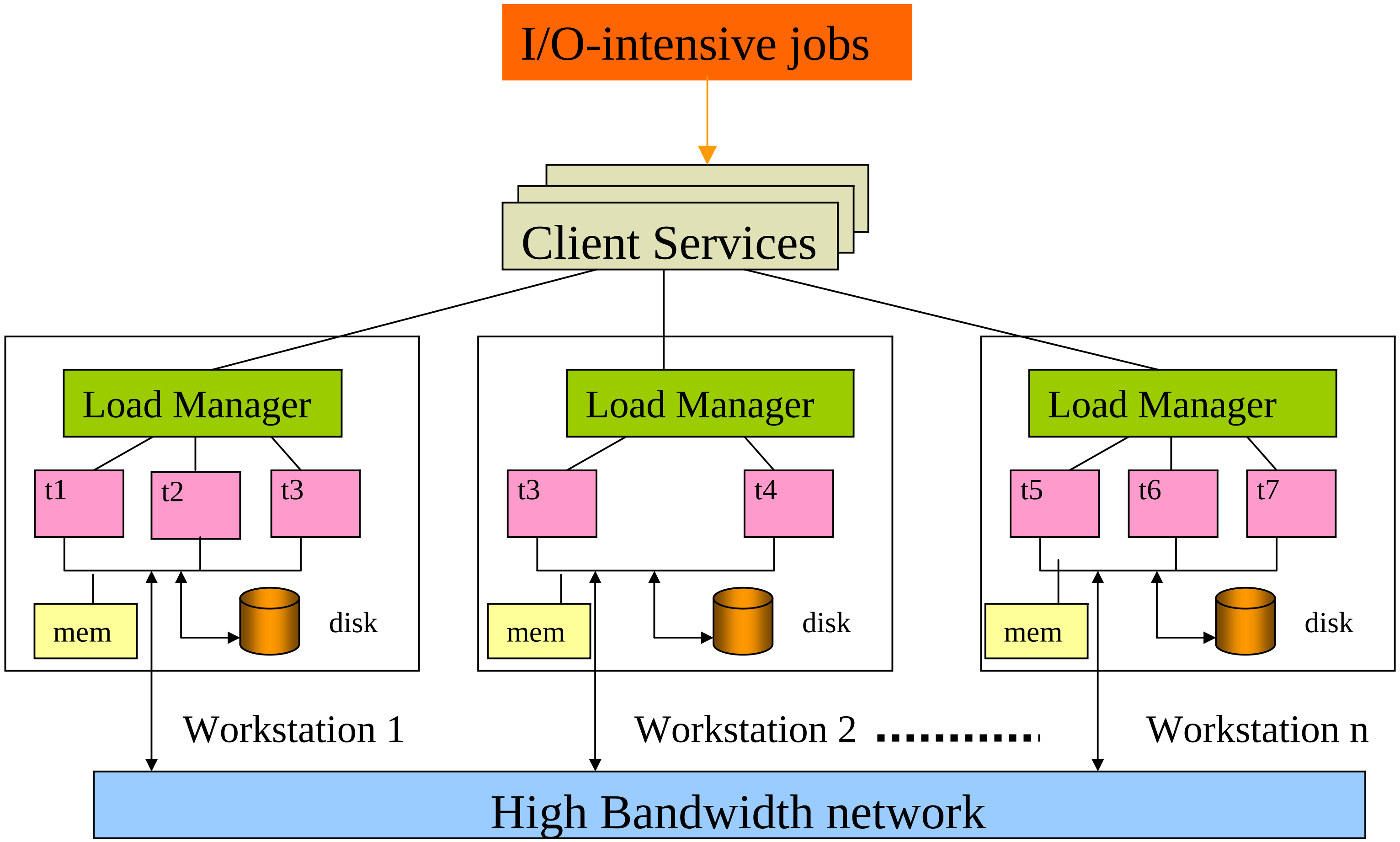


The I/O Bottleneck

Contributions

- Developed two novel I/O-aware load balancing schemes
- Introduced I/O load index
- Evaluate the performance of the proposed load-balancing techniques.

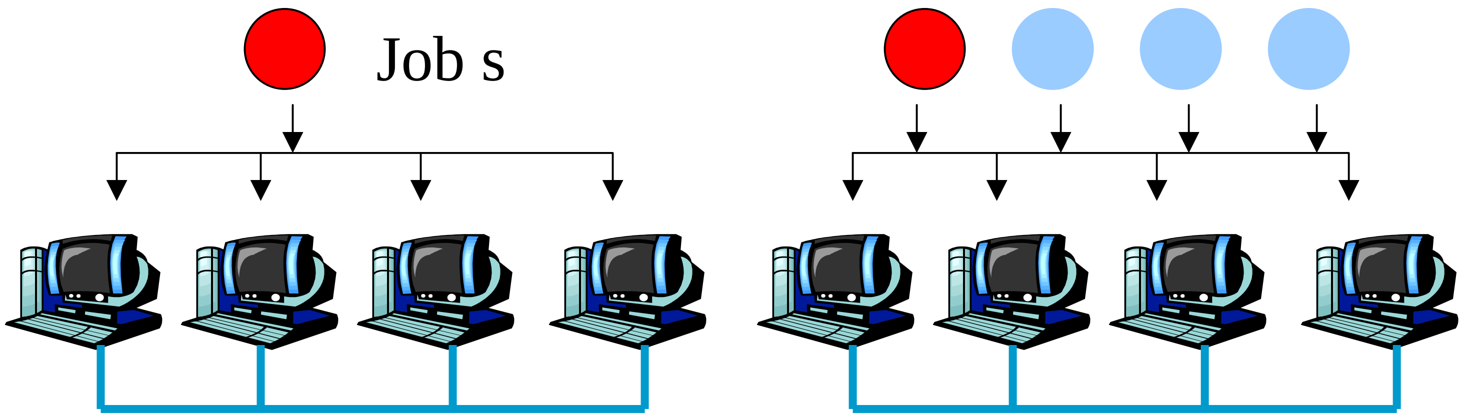
System Architecture



Performance Metric: Slowdown

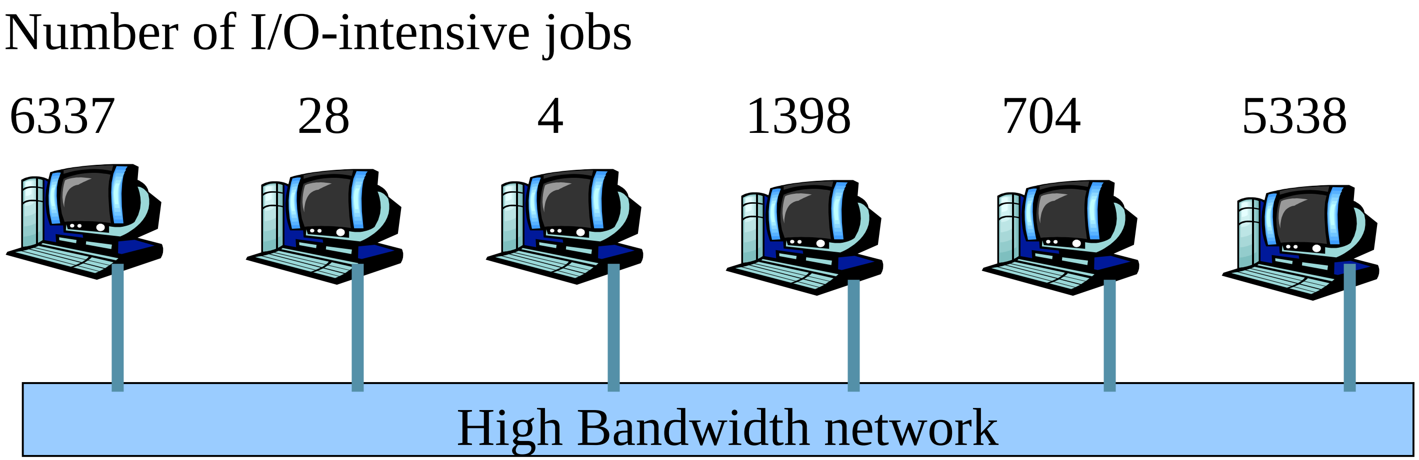
The slowdown of a job is defined as the ratio between the job's execution time in a resource-shared setting and its execution time running in the same system but without any resource sharing.

$$exe\_time_{no-sharing}(s) = 10\,\text{sec}. \qquad time_{sharing}(s) = 45\,\text{sec}.$$



$$slowdown(s) = \frac{time_{sharing}(s)}{time_{no-sharing}(s)} = \frac{45}{10} = 4.5$$

Example



Average Slow Down without load balancing: 25.0

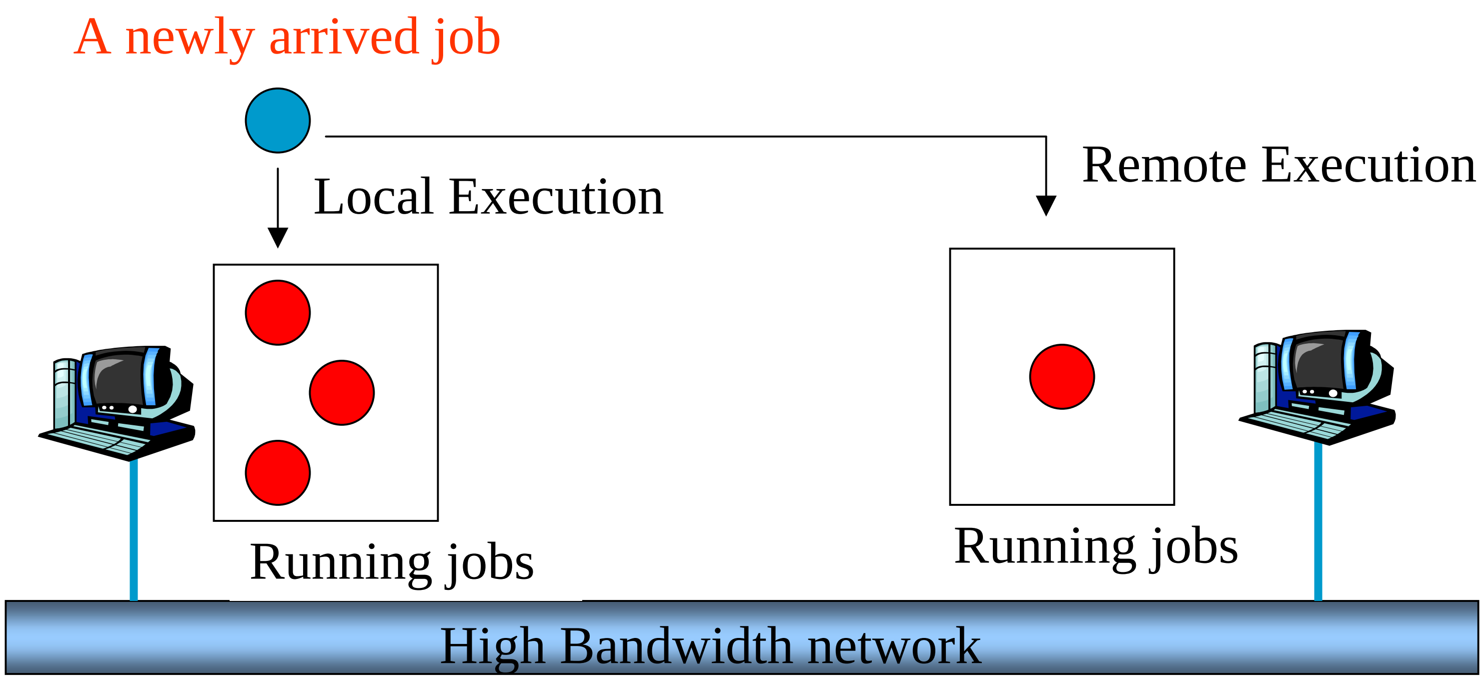
Average Slow Down with I/O-aware load balancing: 2.4

Improved by more than a factor of 10

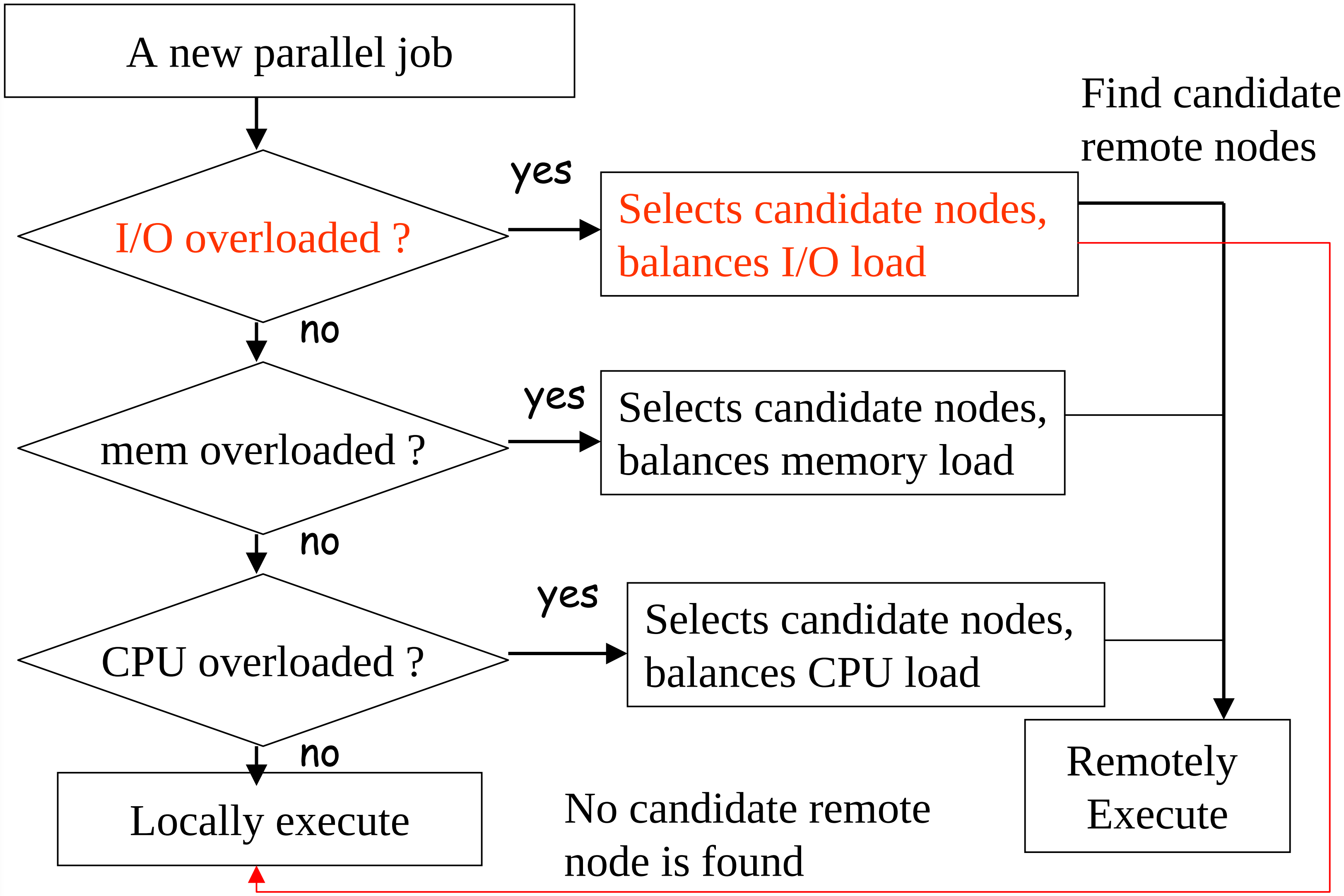
Goals

- Consider multiple resources: Disk I/O, CPU, memory.
- Sustain high performance for I/O-intensive parallel applications.

Load Balancing with Remote Execution

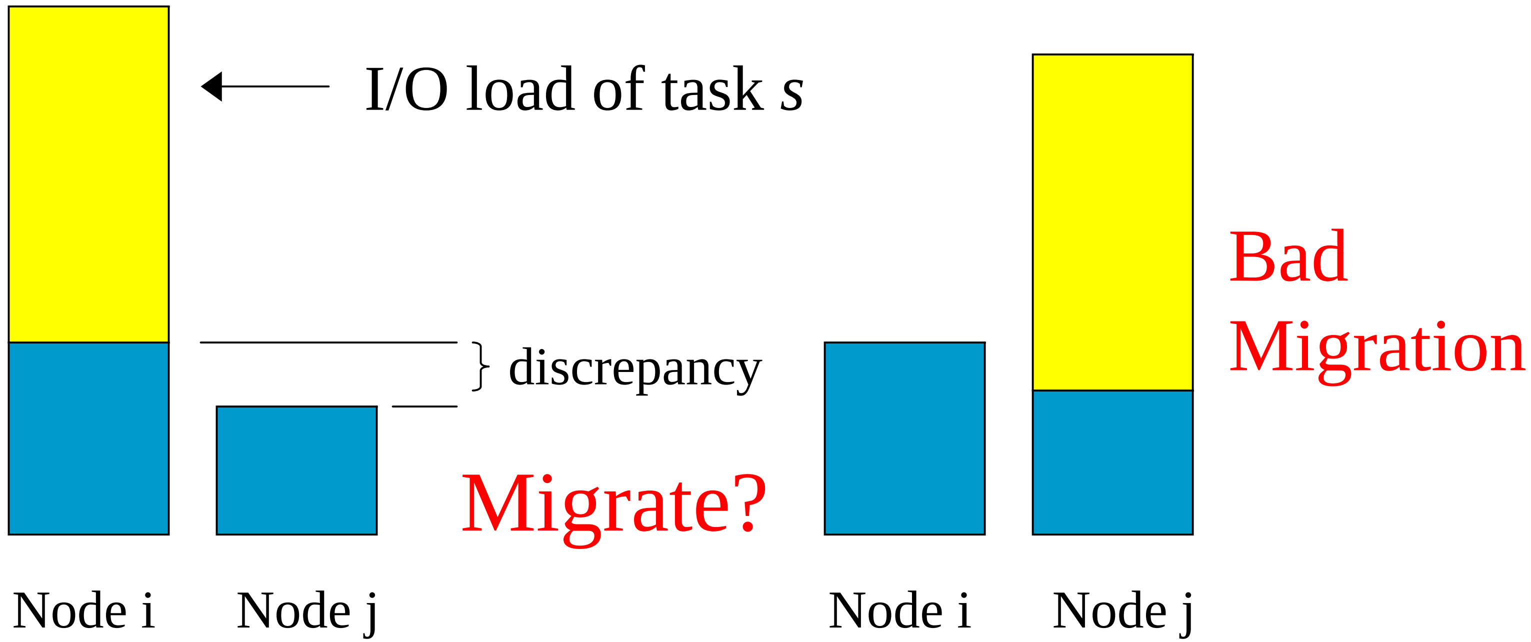


The IOCM-RE Scheme



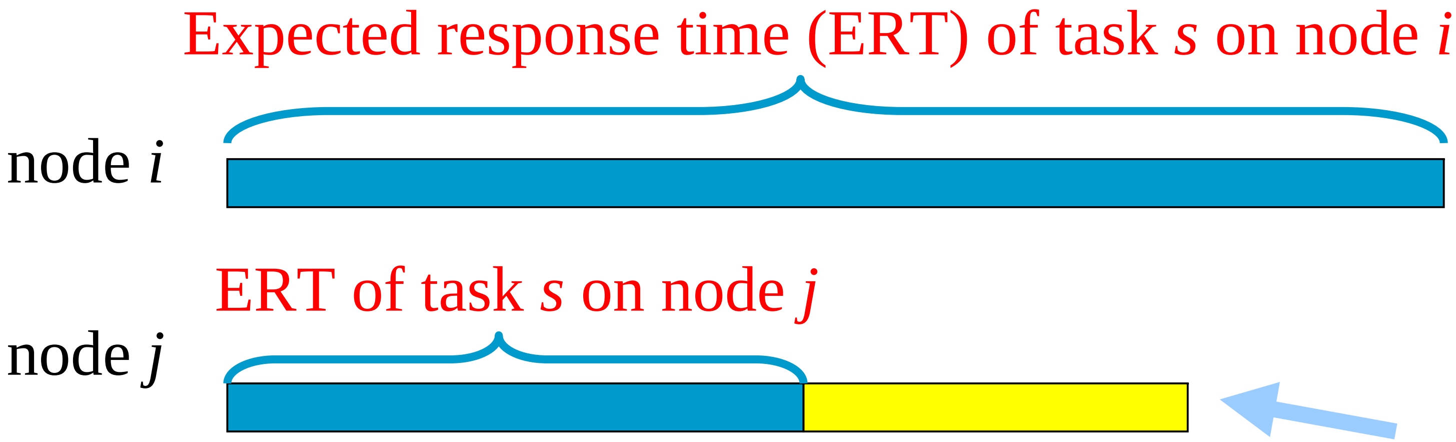
Criterion 1:

Given a task *s* arriving at node *i* and a candidate remote node *j*: The I/O load discrepancy between node *i* and *j* is greater than the I/O load induced by task *s*.



Criterion 2:

Given a task *s* arriving at node *i* and a candidate remote node *j*:



$$r(i, s) > r(j, s) + c_s(i, j)$$

Remote execution cost

Expected response time of task *s* on the local node *i*

Expected response time of task *s* on the remote node *j*

Remote Execution Cost

Given a task *s* arrived in node *i* and a candidate remote node *j*:

Initial data: data initially stored on disk

$$c_s(i,j) = e + d_s^{INIT}\left(\frac{1}{b_{net}^{ij}} + \frac{1}{b_{disk}^{i}} + \frac{1}{b_{disk}^{j}}\right)$$

Fixed cost

Available network bandwidth

Available disk bandwidth

Expected Response Time

Given a task *s* arrived in node *i:*

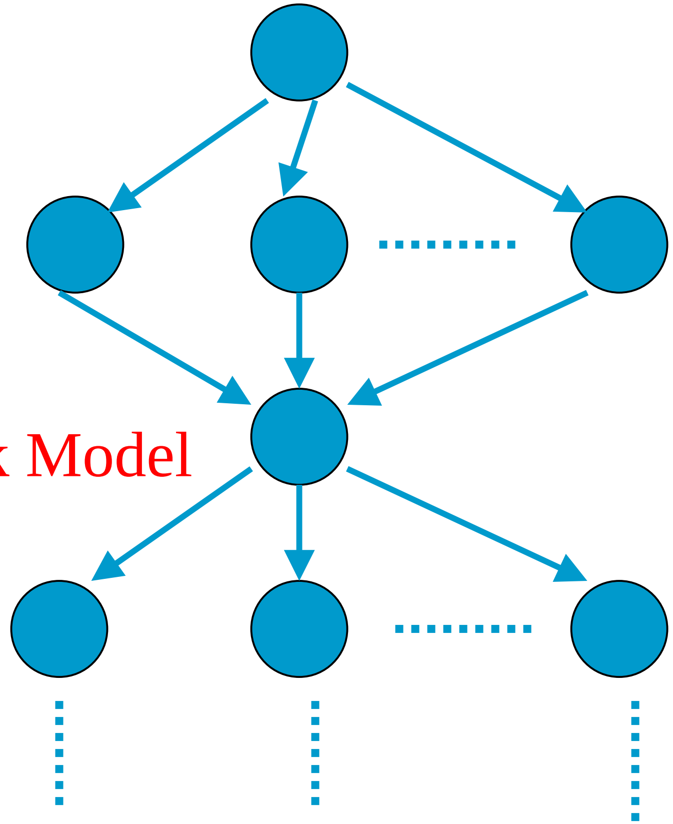
$$r(i,s) = \underbrace{t_s E(L_i)}_{\text{CPU execution time}} + \underbrace{\overbrace{t_s \lambda_s}^{\text{Number of I/O requests}} \overbrace{\left[E(s^i_{disk}) + \frac{\Lambda^i_{disk} E[(s^i_{disk})^2]}{2(1-\rho^i_{disk})} \right]}^{\text{Response time per I/O requests}}}_{\text{I/O processing time}}$$

Trace-driven simulations

- A mixture of sequential and parallel jobs are running.
- The number of tasks in each parallel job is randomly generated according to a uniform distribution between 2 and 32.
- Data sizes of the I/O requests are randomly generated based on a Gamma distribution with the mean size of 256Kbyte
- Compare with CPU-aware (CPU) and Memory-aware (MEM) schemes

Trace-driven simulations (Cont.)

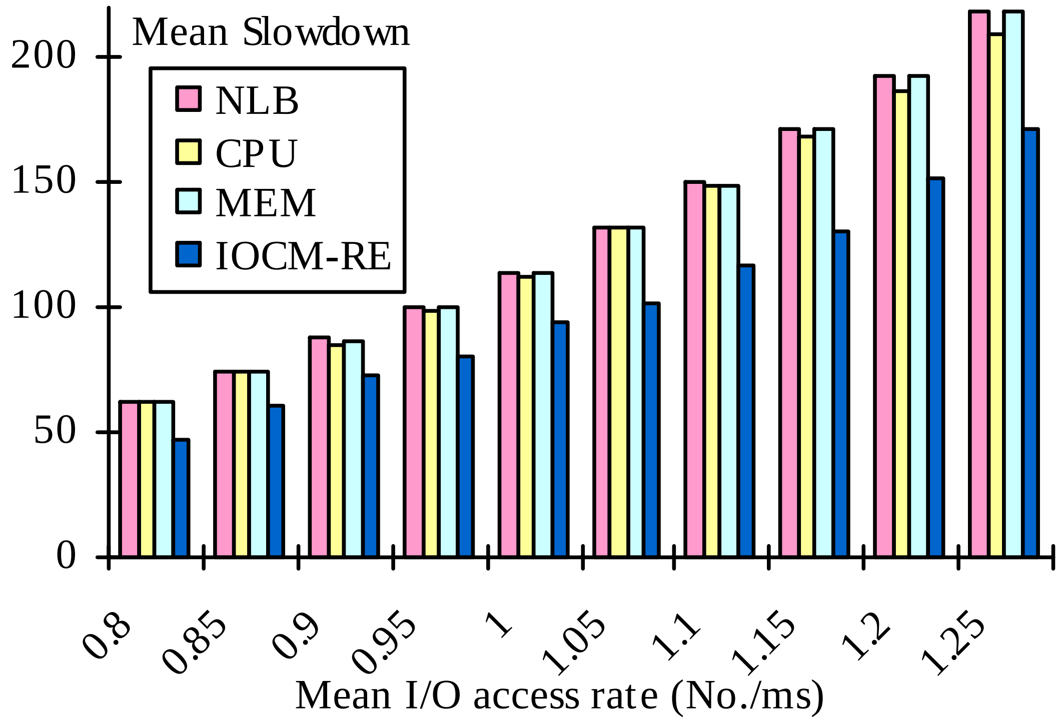
- Simulate a bulk-synchronous style of communication for parallel jobs



Join-and-Fork Model

Performance Evaluation

I/O-intensive Workload Conditions



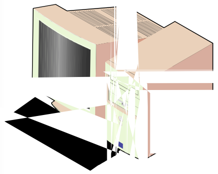
Improvements

Average : 24.5%

Max: 30.7%

- The mean slowdowns increase with the I/O load
- IOCM-RE is the best scheme

IO-CPU-Memory Based Load Balancing with Preemptive Migration (IOCM-PM)



12/14/2003

Select Eligible Tasks for Migration

- Guarantee that the response time of the expected execution on the selected remote node is less than the execution on the current node.

Given a task *s* running on node *i* and
a candidate remote node *k*:

$$r_{PM}(i, s) > r_{PM}(j, s) + c_s(i, j)$$

The migration cost

Expected response time of task *s* running on node *i*

Expected response time of task *s* running on the remote node *j*

Migration Cost

Given a task j running on node i and a candidate remote node k, the expected migration cost, $c_j(i, k)$, is estimated as:

Data to be migrated

Fixed cost

$$f + \frac{m_j}{b_{net}} + \left(d_j^{INIT} + d_j^{W}\right)\left(\frac{1}{b_{net}^{ik}} + \frac{1}{b_{disk}^{i}} + \frac{1}{b_{disk}^{k}}\right)$$

Memory transfer cost

The available bandwidth of the network link between node i and k

The available disk bandwidth in node i and k.

Expected Response Time of a Candidate Migrant

Given a task *s* arrived in node *i:*

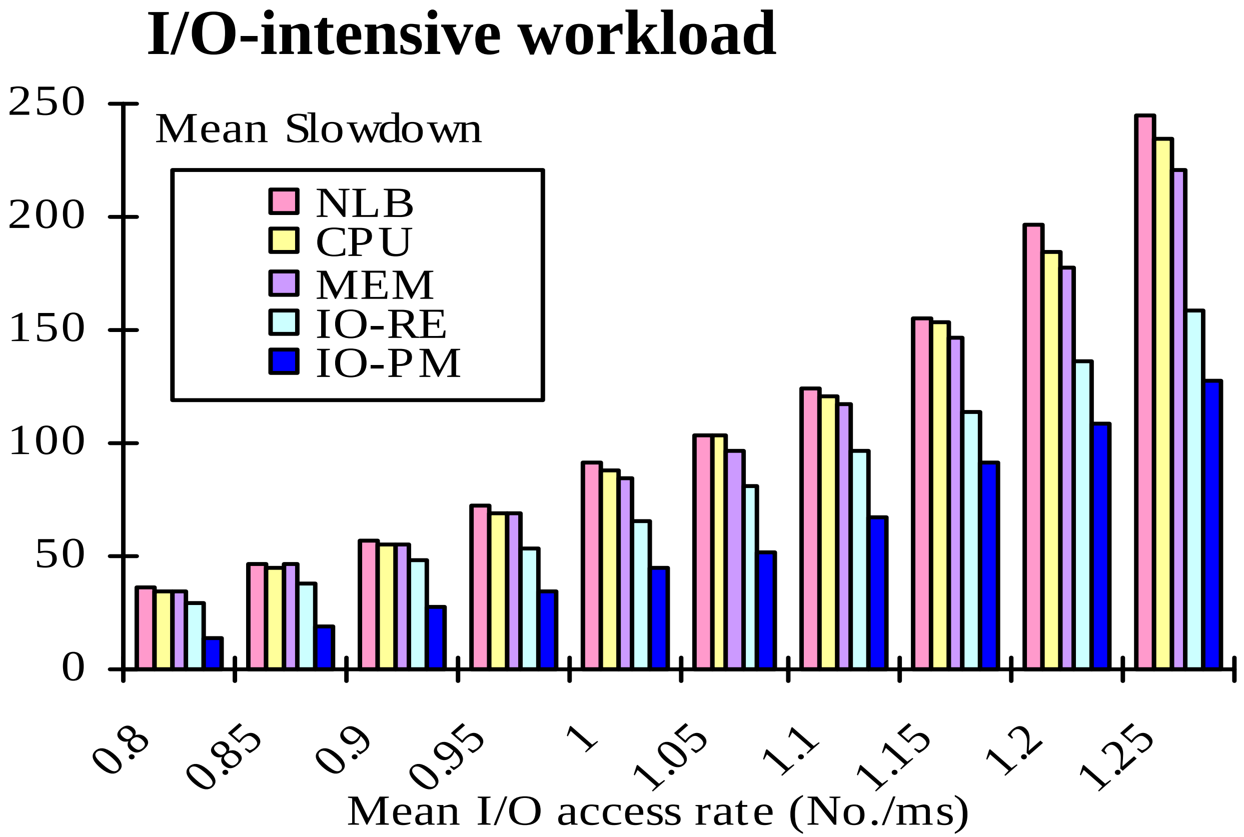
$$r_{PM}(i,s) = \underbrace{(t_s - a_s)E(L_i)}_{\text{CPU execution time}} + \underbrace{\overbrace{(t_s - a_s)\lambda_s}^{\text{Number of I/O requests}} \overbrace{\left[E(s^i_{disk}) + \frac{\Lambda^i_{disk} \times E[(s^i_{disk})^2]}{2(1-\rho^i_{disk})} \right]}^{\text{Response time per I/O requests}}}_{\text{I/O processing time}}$$

Remote Execution vs. Preemptive Migration

Workload	Remote execution	Preemptive Migration
CPU-intensive	+	+++
Memory-intensive	+++	+
I/O-intensive	?	?

- Question: Which one is better for I/O-intensive workload?

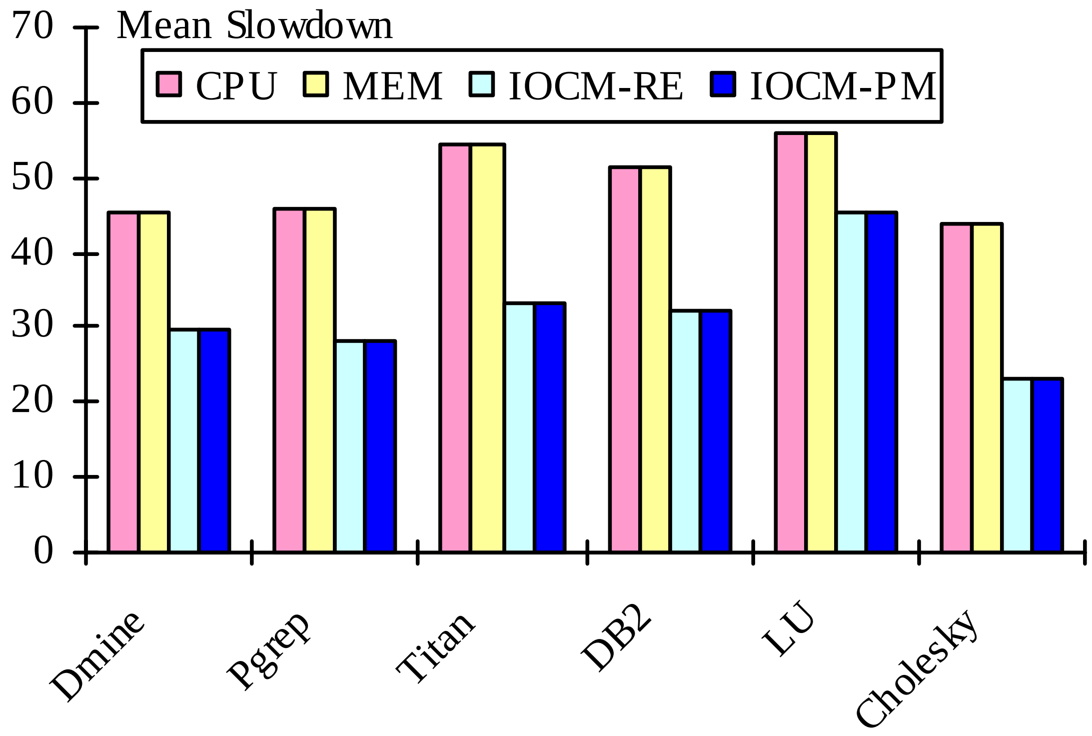
Evaluation of the IOCM-PM Scheme



- IOCM-PM performs the best among all the schemes.

- An average performance improvement of 56.6% over IOPM-RE.

Evaluation of the IOCM-PM Scheme (Cont.)



- IOCM-RE and IOCM-PM benefit all I/O intensive applications
- IOCM-RE and IOCM-PM yield approximately identical performances.

Conclusion

- I/O-aware load balancing schemes with remote execution and preemptive migrations
- Preemptive migration is better than remote execution
- To achieve high performance under a wide spectrum of workload conditions

Q&A