Improving Parallel Job Scheduling Performance In Multi-clusters Through Selective Job Co-allocation

William M. Jones  
Dissertation Defense  
Friday, November 11th, 2005  
3PM, Riggs 100-A  

Parallel Architecture Research Laboratory  
Clemson University  
Clemson, South Carolina  

http://www.parl.clemson.edu/beosim
Cluster Computing

- COTS computers
- Computational resources
- Network interconnect
- “Single” supercomputer
- Parallel libraries
- File I/O services
- Solve larger problems
- Multiple users
- Pervasive alternative
- Multiple clusters ...
What is a multi-cluster?

- Computational resource
- Geographic co-location
- Dedicated network
- Single domains
- Multiple participants
- Distributed resources
- Hierarchical NW topology
- Job scheduling ...
Parallel Job Scheduling

- Local execution
- Job migration
- Job co-allocation (*multi-site*)
  - Map across boundaries
  - Sharing resources
  - Network BW contention
  - Can help or hurt
- Some example scenarios ...
makespan without co-allocation

Cluster 2
running job
waiting job 2

Cluster 1
running job
waiting job 1

waiting job 1
waiting job 2
makespan without co-allocation

makespan with co-allocation

improvement

Cluster 2

running job

waiting job 1

Cluster 1

running job

waiting job 2

waiting job 1

waiting job 2

Time
Cluster 2

Cluster 1

Time

running job

waiting job 1

running job

waiting job 2

makespan without co-allocation

makespan with co-allocation and some NW congestion

makespan with co-allocation and no NW congestion

improvement

execution slowdown

waiting job 1

waiting job 2
makespan with co-allocation and extreme NW congestion

makespan without co-allocation

makespan with co-allocation and no NW congestion

worse

execution slowdown
Intelligent Scheduling

- Co-allocation
- *Global optimization*
- Manage node and network
- Control NW contention
- Improve response time
- Ensure fairness
Multi-cluster job scheduling strategies can make use of **network architecture parameters** and current **link utilizations** as well as parallel **program communication characteristics** to significantly improve average **job turnaround time** while ensuring **fairness**.
Related Work

- Very few co-allocation studies
- Fixed co-allocation penalty
- Fixed time to send a non-local message
- Static view of network effects
- Do not treat network as a consumable resource
- Network contention / slowdown effects
Contributions

- Introduce dynamic models
- Identify necessary parameters
- Demonstrate significant improvement
- Establish response to available information
- Illustrate impact of salient workload characteristics
- Ensure fairness among participating clusters
- Optimizations + fairness
- Simulation framework
Simulation Framework

Parallel Job Model
- Computation
- Communication

Multi-cluster Model
- ICN
- Nodes

Workload Generator

Queueing System

Intelligent Schedulers
- Job characterization
- Network topology
- Fairness policies

Performance Evaluation
- Job turnaround time
- Fairness

Simulator
Multi-cluster Topology

- Star layout
- Ideal “flat” switches
- Single NIC
- Single CPU
- Homogeneous
Parallel Job Model

- Communication-centric
- Computation & Comm.
- Collective communication
  - Local access pattern
  - Global access pattern

Per-Processor Bandwidth (PPBW)

Collective Patterns

Original Execution Time Profile

Local
Global
Parallel Job Model

Cluster 1

Cluster 2

Cluster 3

Cluster 1

Cluster 2

Cluster 3
Job Co-allocation Model
Workload Generation

- Arrival distribution
- Run-time distribution
- Size distribution
- Synthetic workload
  - Poisson arrival process
  - Exponential run-times
  - Uniform sizes
  - Simple vs. ease of use

- Realistic workload
  - Derived from actual workload traces
  - More complex, but less easy to use
Performing Co-allocation

- Strategies with access to link utilizations only
- Ultimate performance depends on job communication characteristics
- Co-allocation can improve or degrade system performance
- *Scheduler needs communication structure and bandwidth requirements*
- *Determine partition sizes that do not saturate links*
- Guarantees that performance is never worse than migration only
- Controls network contention during increased levels of communication
- Focus on our best scheduler “A1”
Response to Communication Intensity

Algorithm Comparison, Real Workload

Job turnaround time (sec)

PPBW (Mbps)

First-Fit
B1
B3
A1
Zero Comm. Cost
Migration Only
Remaining Outline

- Impact of information availability
  - Cost-benefit relationship of performing co-allocation

- Response to changes in salient workload characteristics
  - Which characteristics affect co-allocation the most

- Ensuring fairness among participating clusters
  - Non-negative experience during disparate workload intensities
Information Availability

Synthetic Workload

Max improvement possible: 36%
A1 Best: 31%

Realistic Workload

Max improvement possible: 9%
A1 Best: 8%
Factors that Limit Co-allocation

- Communication intensity
- Limited node fragmentation
- Number of candidate jobs
- Serial or very small jobs
- Periodicity of arrival process
- Comparison to migration only
- Modified workloads
Effect of Arrival Process

Realistic WL w/ Poisson AP (WL A)

Max improvement possible: 28%
A1 Best: 15%

Synthetic WL w/ real AP (WL B)

Max improvement possible: 9%
A1 Best: 8%
Effect of Small Jobs

Realistic WL w/ small jobs removed (WL C)

Max improvement possible: 24%
A1 Best: 14%

Realistic WL w/ small jobs removed and Poisson AP (WL D)

Max improvement possible: 45%
A1 Best: 27%
Ensuring Fairness

- Disparate workload intensities
- Different cluster sizes
- Unfair resource sharing
- Overload remote clusters
- Worse than not participating
- Technique to control fairness
Fairness Via Conservative Backfilling

- Out of order execution
- No delay to start time
- Two-tiered approach
- Backfill local jobs first
- Local backfill schedules
- Consider remote job backfill
- Constrain non-local node use
- Prevents local job starvation

[Diagram showing local and remotely queued jobs with time and node timelines]

Nodes

Time

Job 1 (running)
Job 2 (running)
Job 3 (queued)
(available nodes)

Remotely queued jobs

Job 1 (queued) Job 2 (queued)
Fairness, Syn. WL, 3 Clusters (100, 100, 100) w/ Inc. Load, 1 Cluster (100) w/ Fixed Load

Arrival Rate on the first 3 clusters (Jobs/Day/Cluster)

A1 (Orig) on Grid
A1+Backfill on Grid
A1+Backfill on Fixed Cluster
A1+Backfill on Other Clusters
No Share on Fixed Cluster

Job turnaround time (sec)
Fairness, Syn. WL, 3 Clusters (100, 100, 100) w/ Inc. Load, 1 Cluster (50) w/ Fixed Load

Arrival Rate on the first 3 clusters (Jobs/Day/Cluster)

- A1 (Orig) on Grid
- A1+Backfill on Grid
- A1+Backfill on Fixed Cluster
- A1+Backfill on Other Clusters
- No Share on Fixed Cluster

Job turnaround time (sec)
Fairness, Syn. WL, 3 Clusters (200, 100, 100) w/ Inc. Load, 1 Cluster (50) w/ Fixed Load

Arrival Rate on the first 3 clusters (Jobs/Day/Cluster)

A1 (Orig) on Grid
A1+Backfill on Grid
A1+Backfill on Fixed Cluster
A1+Backfill on Other Clusters
No Share on Fixed Cluster

Job turnaround time (sec)
Conclusions

- Job co-allocation can improve system performance
- Access to job characteristics and link utilizations are necessary
- Control network contention and manage bandwidth as a resource
- Performance highly sensitive to salient workload characteristics
- Traditional backfilling effectively prevents unfairness
Future Work

- Add model for data staging during migration and co-allocation
- Add model for job checkpointing to recover from bad decisions
- Modify job model to reflect phases of comm. & comp.
- Refine network model to include more general topologies
- Refine slowdown model to address effects due to NW protocols
Scholarly Contributions

Publications

- Special Issue of Journal of Supercomputing, 2005
- Intl. Conference on Cluster Computing, 2004
- Intl. Parallel and Distributed Processing Symposium, 2004

Poster Sessions

- NASA booth at Supercomputing Conference, 2004
- NSF site visits for CAEFF, 2003, 2004

Other

- MS thesis – Workload Generation, Nishant Shrivastava, to appear May 2005
- MS thesis – Simulator Internals, Louis Pang
- SURE project – BeoSim Visualizer, Michael Bassily, 2004
Questions ?
Communication Characterization

Processor Bandwidth Decomposition

\[ PPBW_j = BW_G + BW_L \]
\[ = \rho \cdot PPBW_j + (1 - \rho) \cdot PPBW_j \]  

Demand on link \( i \) due to job \( j \)

\[ BW^j_i = BW_{Global}^j + BW_{Local}^j \]  

\[ BW^j_i = EC_G \cdot \left( \frac{BW_G}{Deg_G} \right) + EC_L \cdot \left( \frac{BW_L}{Deg_L} \right) \]  

\[ BW_{Global}^j = (n^j_i \cdot PPBW_j \cdot \rho) \left( \frac{n^j_i}{n^j_T} \right) \]  

\[ BW_{Local}^j = \text{depends on partitioning} \]
Bandwidth saturation
\[ BW_{i}^{sat} = \sum_{j \in J_i} \frac{BW_{j}^i}{BW_{i}^{max}} \]

Communication slowdown
\[ BW_{sd}^{(k,j)} = \frac{BW_{allo}^{(k,j)}}{BW_{k}^j} \]
Bandwidth Allotment Algorithm

\[
BW_{i}^{uc\_sat} = \frac{\sum_{\forall j \in J_{uc}} BW_{alloted}^{(i,j)}}{BW_{i}^{avail}}
\]

**Step 1:** *Initialization* - For every job \( j \), let \( BW_{alloted}^{(i,j)} = BW_{i}^{j} \). For every link \( i \), let \( BW_{i}^{avail} = BW_{i}^{max} \). Let the unconstrained set of nodes, \( J_{uc} = J \) (set of all jobs). Let the set, \( J_{i} \) be the set of all jobs that span link \( i \).

**Step 2:** *Saturation detection* - For every link, calculate \( BW_{i}^{uc\_sat} \). While there exists at least one \( BW_{i}^{uc\_sat} > 1.0 \), continue, else goto **Step 5**.

**Step 3:** *Saturation correction* - Identify the link with the largest \( BW_{i}^{uc\_sat} \) (most saturated link) from **Step 2**, and globally reduce the allotted bandwidth of every job in \( J_{i} \cap J_{uc} \) by a factor of \( BW_{i}^{sat} \).

**Step 4:** *Update state* - Remove each of the modified jobs from the set \( J_{uc} \). For each of the modified jobs, remove their allotted bandwidth from the available bandwidth, \( BW_{i}^{avail} \) on each link over which they span. Goto **Step 2**.

**Step 5:** *Termination* - DONE.
Co-allocation Model Continued

- State changing events
  - New co-allc’ed job
  - Co-allc’ed job finishes
- Residual execution times

Residual Execution Time

\[ T_{RE}^R = \sqrt{T_C^R + T_P^R} \]

Residual Communication Time

\[ T_{RC}^R = (T_{C}^{R'} - T_{C}^{A}) (BW_{sd}^{sd'}) (BW_{sd}^{sd})^{-1} \]

Residual Computation Time

\[ T_{RP}^R = \sqrt{(T_{P}^{R'} - T_{P}^{A})} \]

\[ T_{P}^{A} = \frac{\Delta T}{T_{E}} T_{P}^{R'} \quad \text{and} \quad T_{C}^{A} = \frac{\Delta T}{T_{E}} T_{C}^{R'} \]
Scheduler Model

Scheduling Iteration

Begin → FCFS Select → Local Allocation → Job Migration → Co-allocation → Reloop → Done

Bandwidth-Aware Co-allocation

Sat. Level > LSLT? → Scheduler
Initial Results – Naive

Effects become exacerbated as the number of clusters is increased
Max BW Due to 2D Mesh

\[ \text{Max } EC_L = 2 \cdot (\min(\lceil \sqrt{n_T} \rceil, NA_i) + 1) \] (6)

\[ \text{Max } BW_i^L = \text{Max } EC_L \cdot \left( \frac{BW_L}{\text{Deg}_L} \right) \] (7)

\[ = 2 \cdot (\min(\lceil \sqrt{n_T} \rceil, n_i^{\text{avail}}) + 1) \]

\[ \times \left( \frac{(1 - \rho) \cdot PPBW_j}{4} \right) \]
Bandwidth-Aware Needed

\[ BW_{i}^{\text{remain}} = BW_{i}^{\text{avail}} - \max BW_{L}^{i} \quad (8) \]

\[ n_{(1,2)}^{i,j} = \frac{1}{2} \left( n_{T}^{j} \mp \sqrt{(n_{T}^{j})^2 - \frac{4BW_{i}^{\text{remain}}(N_{j}^{j} - 1)}{PPBW_{j}^{j}}} \right) \quad (9) \]

\[ S_{1}^{(i,j)} = \left( [0, n_{1}] \cup \left[ n_{2}, n_{T}^{j} \right] \right) \cap \left[ 0, n_{i}^{\text{avail}} \right] \quad (10) \]

\[ X_{i}^{j} \in S_{1}^{(i,j)}, \sum_{i=1}^{N} X_{i}^{j} = n_{T}^{j} \quad (11) \]
Dynamic Model Comparison

- 100 nodes per cluster
- 1Gbps intercluster network links
- Size = $UNIF[10, 90]$
- All GLOBAL no LOCAL communication
- 70% comp. / 30% comm.
- Poisson AP 150 interarrival time average
- All jobs in same simulation have same BSBW
- Varying PPBW
- Initial execution time, EXP w/ average 225
- 4,000,000 jobs run per simulation
Dynamic Model 2 Clusters

Turnaround Time vs Bisection Bandwidth (2 Clusters)

Turnaround Time vs Co-allocation Penalty (2 Clusters)
Dynamic Model 4 Clusters

**Turnaround Time vs Bisection Bandwidth (4 Clusters)**

- **First-fit**
- **Fixed**
- **No Share**
- **Ideal**
- **Migration Only**

**Turnaround Time vs Co-allocation Penalty (4 Clusters)**

- **First-fit**
- **Fixed**
- **No Share**
- **Ideal**
- **Migration Only**
Dynamic Model 8 Clusters

Turnaround Time vs Bisection Bandwidth (8 Clusters)

Turnaround Time vs Co-allocation Penalty (8 Clusters)
Baseline Performance

- Good performance
- Potentially unrealistic
- No a priori knowledge?
- More meaningful
Co-allocation Algorithms

- A1 – Constraint Satisfaction
- B1 – Largest Free Nodes First
- B2 – Least Saturated Link First
- B3 – Big-small Chunking
- B4 – Load-balancing
Experimental Setup

- Synthetic workload
- Poisson AP 150 IAT
- Exp. ST *(initially)* 225 avg.
- 70% comp. / 30% comm.
- 1 Gbps IC network links
- 400,000 jobs, *UNIF*[10, 50]
- 100 nodes per cluster
CS Baseline Case (A1)  Largest Free Nodes First (B1)
Least Sat. Link First (B2)

Load Balancing (B4)
Big-Small (B3, 70%)

Big-Small (B3, 80%)
Big-Small (B3, 85%)

Big-Small (B3, 90%)
Algorithm Comparison with LSLT fixed at: 100%

- A1 Algorithm (Satisfy)
- B1 Algorithm (Largest free)
- B2 Algorithm (Least sat. link)
- B3 Algorithm (Chunking) (85%)
- B4 Algorithm (Load-balance)
- Ideal
- Migration Only
Algorithm Comparison at Low Job BSBW

- A1 Algorithm (Satisfy)
- B1 Algorithm (Largest free)
- B2 Algorithm (Least sat. link)
- B3 Algorithm (Chunking) (85%)
- B4 Algorithm (Load-balance)
- Ideal
- Migration Only
Algorithm Comparison at High Job BSBW

- A1 Algorithm (Satisfy)
- B1 Algorithm (Largest free)
- B2 Algorithm (Least sat. link)
- B3 Algorithm (Chunking) (85%)
- B4 Algorithm (Load-balance)
- Ideal
- Migration Only
Algorithm Comparison with LSLT fixed at: 100%

- A1 Algorithm (Satisfy)
- B1 Algorithm (Largest free)
- B2 Algorithm (Least sat. link)
- B3 Algorithm (Chunking) (85%)
- B4 Algorithm (Load-balance)
- Ideal
- Migration Only
Chunking Algorithm Comparison with LSLT fixed at: 100%

A1 Algorithm (Satisfy)

B3 Algorithm (Chunking) (70%)
B3 Algorithm (Chunking) (80%)
B3 Algorithm (Chunking) (85%)
B3 Algorithm (Chunking) (90%)

Ideal

Migration Only
Chunking Algorithm Comparison at Low Job BSBW

- A1 Algorithm (Satisfy)
- B3 Algorithm (Chunking) (70%)
- B3 Algorithm (Chunking) (80%)
- B3 Algorithm (Chunking) (85%)
- B3 Algorithm (Chunking) (90%)

Job turnaround time (sec)

- Ideal
- Migration Only

LSLT (%)
Chunking Algorithm Comparison at High Job BSBW

- A1 Algorithm (Satisfy)
- B3 Algorithm (Chunking) (70%)
- B3 Algorithm (Chunking) (80%)
- B3 Algorithm (Chunking) (85%)
- B3 Algorithm (Chunking) (90%)

Ideal
Migration Only
100 nodes per cluster
1Gbps intercluster network links
Size = \( UNIF[10, 50] \)
2D and GLOBAL \( \rho = UNIF[0, 1] \)
70% comp. / 30% comm.
Poisson AP 150 interarrival time average
All jobs in same simulation have same PPBW
Varying BSBW
Initial execution time, EXP w/ average 225
400,000 jobs run per simulation
Information Availability

- Jobs expose info. randomly at rate x%
- 100 nodes per cluster
- 1Gbps intercluster network links
- Size = $UNIF[10, 50]$
- 2D and GLOBAL $\rho = UNIF[0, 1]$
- 70% comp. / 30% comm.
- Poisson AP 150 interarrival time average
- $PPBW = UNIF[0, 300]$
- Initial execution time, EXP w/ average 225
- 400,000 jobs run per simulation
Empirical Verification

Slowdown Effect on Job Execution Time and Link Utilizations

- Observed Time
- Observed Link Util.
- Predicted Time (model 1)
- Predicted Time (model 2)
- Predicted Link Util.
- BW Needed (Mbps)
Expand Job & NW Models