The Impact of Information Availability and Workload Characteristics on the Performance of Job Co-allocation in Multi-clusters

William M. Jones
Assistant Professor
Electrical Engineering Department
United States Naval Academy
Annapolis, Maryland

(previously at Clemson University)

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

![Diagram of parallel job scheduling with clusters and job migration](attachment:parallel_job_scheduling_diagram.png)
Cluster 2
running job
waiting job 2
Cluster 1
running job
waiting job 1
waiting job 1
makespan without co-allocation
waiting job 2
waiting job 1
Time
Cluster 2
Cluster 1
running job
waiting job 1
running job
waiting job 2
waiting job 1
waiting job 2
makespan without co-allocation
makespan with co-allocation
improvement
Time
Cluster 1
Cluster 2

Time
running job
waiting job 1
running job
waiting job 1
makespan without co-allocation
waiting job 2
makespan with co-allocation and no NW congestion
waiting job 2
makespan with co-allocation and some NW congestion
execution slowdown

improvement

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
waiting job 1
waiting job 2
execution slowdown
waiting job 1
waiting job 2
worse
Intelligent Scheduling

- Co-allocation
- **Global optimization**
- Manage node and network
- Control NW contention
- Improve turnaround time
- Ensure fairness
Contributions

- Introduced dynamic communication models
- Designed several *online* scheduling strategies
- Ensured fairness among participating clusters
- Implemented simulation framework
- Demonstrated significant improvement; HOWEVER,
- Established response to available information
- Illustrated impact of salient workload characteristics
Response to Communication Intensity

Algorithm Comparison, Synthetic Workload

![Graph showing response to communication intensity](image)
Workload Generation

- Arrival distribution
- Run-time distribution
- Size distribution
- Synthetic workload
  - Poisson arrival process
  - Exponential run-times
  - Uniform sizes
  - Simple

- Realistic workload
  - Derived from actual workload traces
  - More complex
  - Tunability?
  - Dependent parameters
  - More realism
  - ...
Workload Model Job Arrivals - Daily Cycle

![Daily Cycle Graph]

Workload Model Job Arrivals - Weekly Cycle

![Weekly Cycle Graph]

Job Run-time Distribution

![Runtime Distribution Graph]

Jobs Size Distribution

![Size Distribution Graph]
Response to Communication Intensity

Algorithm Comparison, Real Workload

PPBW (Mbps) vs. Job turnaround time (sec)

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

Job turnaround time (sec)

PPBW (Mbps)
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
Information Availability

Synthetic Workload

Realistic Workload

Max improvement possible: 36%
A1 Best: 31%

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)

Synthetic WL w/ real AP (WL B)

Max improvement possible: 28%
A1 Best: 15%

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

Realistic WL w/ small jobs removed (WL C)

A1 performance w.r.t. Information Availability

Max improvement possible: 24%
A1 Best: 14%

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

A1 performance w.r.t. Information Availability

Max improvement possible: 45%
A1 Best: 27%
Conclusions and Future Work

- 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
- Address heterogeneous node resources
- View job checkpointing/restart to recover from bad scheduling decision
- Re-implement simulation framework in OpNet and compare results
Questions ?
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

(Node)

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Per-Processor Bandwidth (PPBW)

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns

Original Execution Time Profile

Local Global

Collective Patterns
Parallel Job Model

Cluster 1

Cluster 2

Cluster 3

Cluster 1

Cluster 2

Cluster 3
Job Co-allocation Model

Cluster 1

Cluster 2

Cluster 3

Cluster 4

Job 1

Job 2

Job 3

Computation

Communication

End event can slide forward or backward in time

Original Execution Time Profile

24
Brief Overview of Previous Research

- 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”
Communication Characterization

Processor Bandwidth Decomposition

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

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

\[ BW^j_i = BW_{Global}^j + BW_{Local}^j \] (2)

\[ BW^j_i = \left( \frac{BW_G}{Deg_G} \right) \cdot EC_G + \left( \frac{BW_L}{Deg_L} \right) \cdot EC_L \] (3)

\[ BW_{Global}^j = \left( n^j_i \cdot PPBW_j \cdot \rho \right) \left( \frac{n^j_T - n^j_i}{n^j_T - 1} \right) \] (4)

\[ BW_{Local}^j = \text{depends on partitioning} \] (5)
Job Co-allocation Model

BW Needed

Cluster 1

Cluster 2

Cluster 3

Cluster 4

Job 1

Job 2

Job 3

BW Alloted

Allotment Algorithm

\[ BW_{sat} = \sum_{j \in J_i} BW^j_i / BW^{max}_i \]

\[ BW_{sd}(k,j) = BW_{allotted}^{(k,j)} / BW^j_k \]
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 \( BW_{alloted}^{(i,j)} = BW_{j}^{i} \). 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

\[ T_R^{TE} = \sqrt{T_C^R + T_P^R} \]

Residual Communication Time

\[ T_C^R = (T_C^{R'} - T_C^\Delta)(BW^{sd'}_{(k,j)})(BW^{sd}_{(k,j)})^{-1} \]

Residual Computation Time

\[ T_P^R = \sqrt{T_P^{R'} - T_P^\Delta} \]

\[ T_P^\Delta = \frac{\Delta T}{T_E} T_P^{R'} \quad \text{and} \quad T_C^\Delta = \frac{\Delta T}{T_E} T_C^{R'} \]
Upper and Lower 'Bounds' for Com-
parison

No Sharing

Migration

Ideal

"There Is No Grid"

"No Shared BW"

"One Big Cluster"
Initial Results – Naive

Turnaround Time vs Bisection Bandwidth (2 Clusters)

- Initial Strategy
- No Share
- Ideal
- Migration Only

Turnaround Time vs Bisection Bandwidth (8 Clusters)

- Initial Strategy
- No Share
- Ideal
- Migration Only

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

Max EC_L = 2 \cdot \left(\min\left(\lceil\sqrt{n_T}\rceil, N A_i\right) + 1\right) \quad (6)

Max BW_L^i = Max EC_L \cdot \left(\frac{BW_L}{Deg_L}\right) \quad (7)

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

\times \left(\frac{(1 - \rho) \cdot PPBW_j}{4}\right)
\[BW_i^{\text{remain}} = BW_i^{\text{avail}} - \text{Max} BW_i^L \quad (8)\]

\[n_{(1,2)} = \frac{1}{2} \left( n_T^j + \sqrt{(n_T^j)^2 - \frac{4BW_i^{\text{remain}}(N_T^j-1)}{PPBW_j}} \right)\quad (9)\]

\[S_{(i,j)}^{(1,2)} = \left( [0, \lfloor n_1 \rfloor] \cup [n_2, n_T^j] \right) \cap [0, n_i^{\text{avail}}] \quad (10)\]

\[X_i^j \in S_{(i,j)}^{(1,2)}, \quad \sum_{i=1}^{N} X_i^j = n_T^j \quad (11)\]
Ensuring Fairness

- Disparate workload intensities
- Different cluster sizes
- Unfair resource sharing
- Overload remote clusters
- Worse than not participating
- Technique to control fairness
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
Fairness, 3 Clusters (100, 100, 100) w/ Increasing Load, 1 Cluster (100) w/ Fixed Load

Job turnaround time (sec)
Arrival Rate on the first 3 clusters (Jobs/Week/Cluster)
A1 on G
A1+B on G
A1+B on FC
A1+B on OC
No Share on FC
Fairness, 3 Clusters (100, 100, 100) w/ Increasing Load, 1 Cluster (50) w/ Fixed Load

- A1 on G
- A1+B on G
- A1+B on FC
- A1+B on OC
- No Share on FC
Fairness, 3 Clusters (200, 100, 100) w/ Increasing Load, 1 Cluster (50) w/ Fixed Load

Job turnaround time (sec)
Arrival Rate on the first 3 clusters (Jobs/Week/Cluster)

A1 on G
A1+B on G
A1+B on FC
A1+B on OC
No Share on FC
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

 pobl

- 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**
  - Intl. Workshop on Scheduling and Resource Management for Parallel and Distributed Systems, *submitted 2006*
  - Intl. Conference on Parallel and Distributed Systems, *submitted 2006*
  - 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
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)

Turnaround Time vs Co-allocation Penalty (4 Clusters)
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

![Graph showing Turnaround Time vs Bisection Bandwidth (4 Clusters)]
Co-allocation Algorithms

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

Bandwidth-Aware Co-allocation

Scheduler
Sat. Level > LSLT ?
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) vs Largest Free Nodes First (B1)
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

Job turnaround time (sec) vs. Bisection bandwidth (Mbps)
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%)

Ideal
Migration Only
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%)

Job turnaround time (sec)

- Ideal
- Migration Only

LSLT (%)
Response to Comm. Intensity

- 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