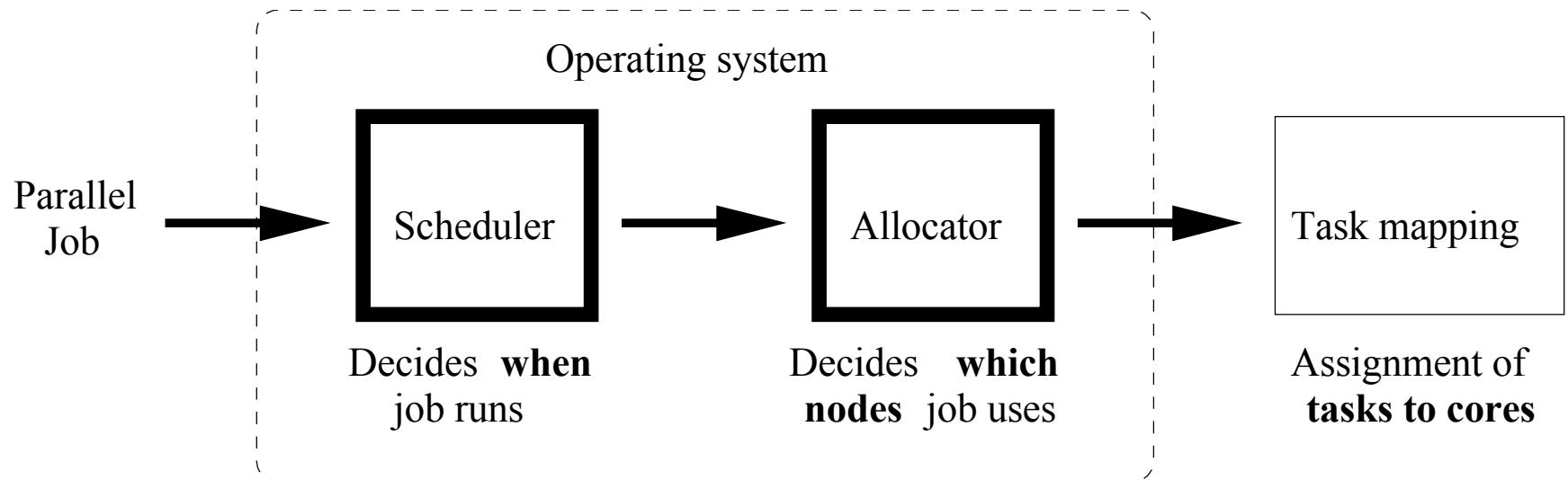


## Local search to improve task mapping

Balzuweit\*, Bunde\*, Vitus Leung, Finley\*, Lee\*  
 SIAM CSE, 18 March 2015

\*Knox College, Galesburg, IL

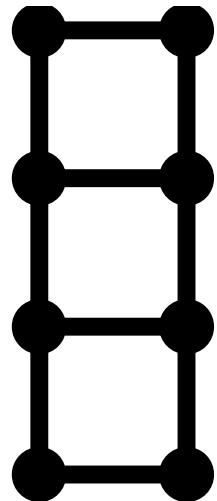
# Parallel (Distributed Memory) Resource Management Pipeline



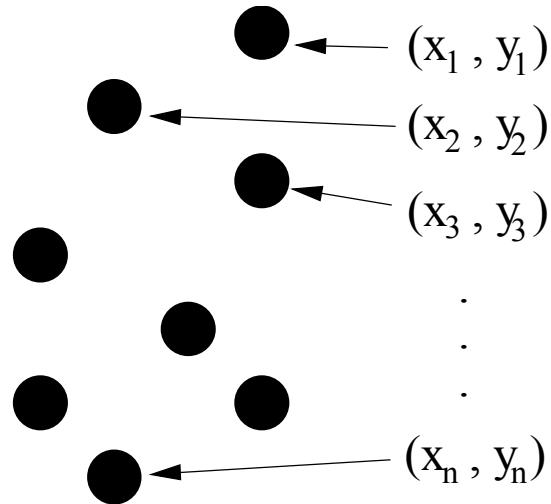
# Task mapping

- Long history [Bokhari, 1981] (general graph model)
- Less important in mid-1980s with wormhole routing
  - Message latency independent of size
- Recent resurgence
  - Almasi et al. 2004
  - Gygi et al. 2006 (application exhibited 1.64 times speedup)
  - Bhatele et al. 2010 (contiguous coordinate model)
  - Hoefer and Snir 2011 (heuristics for NP-Complete general model)
  - Leung et al. 2014 (heuristics for coordinate model, hybrid parallelism)
  - Deveci et al. 2014 (coordinate model vs. general model for miniapps)
- Contention for limited bandwidth
  - Processors continue improving faster than networks
  - Processor counts in state of the art HPC systems continue to grow

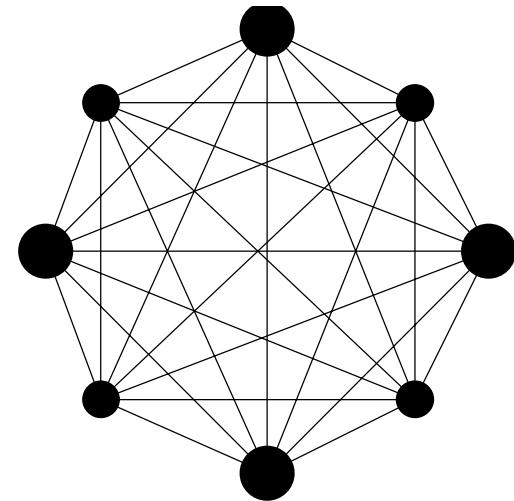
# General view of task mapping



Application  
Task Graph

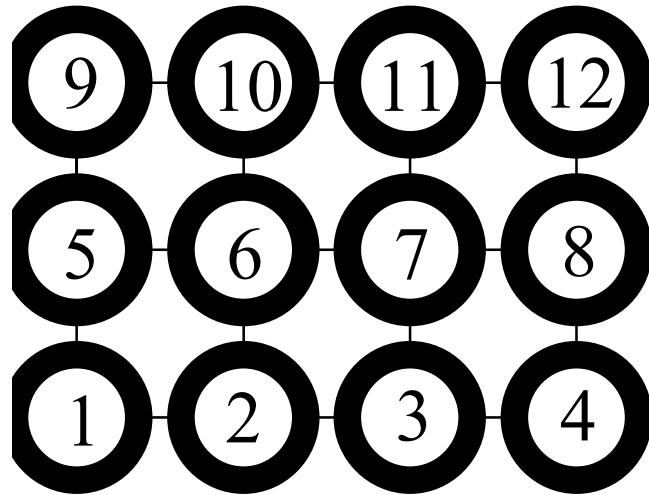


Allocated Processors  
with mesh coordinates

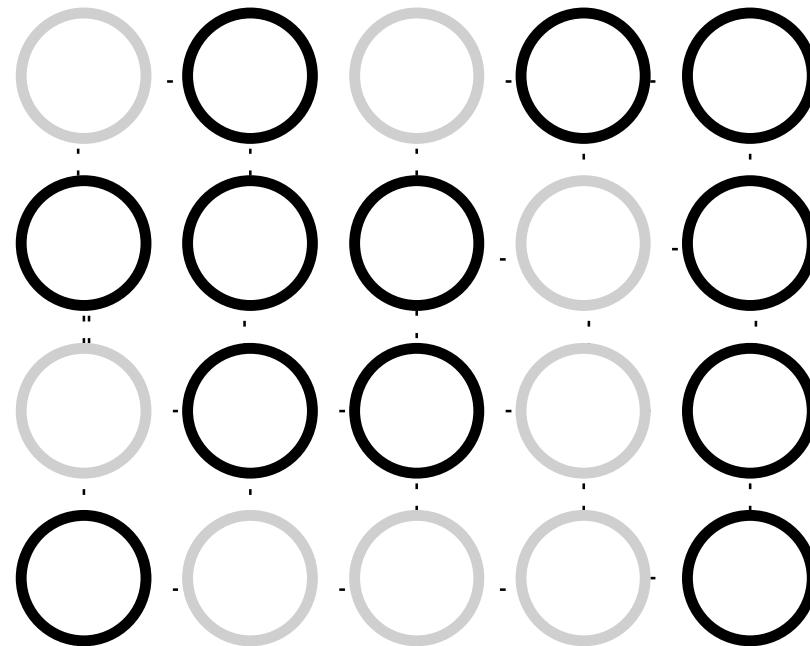


General graph  
view of allocation

# Using recursive coordinate bisection for task mapping (RCB)

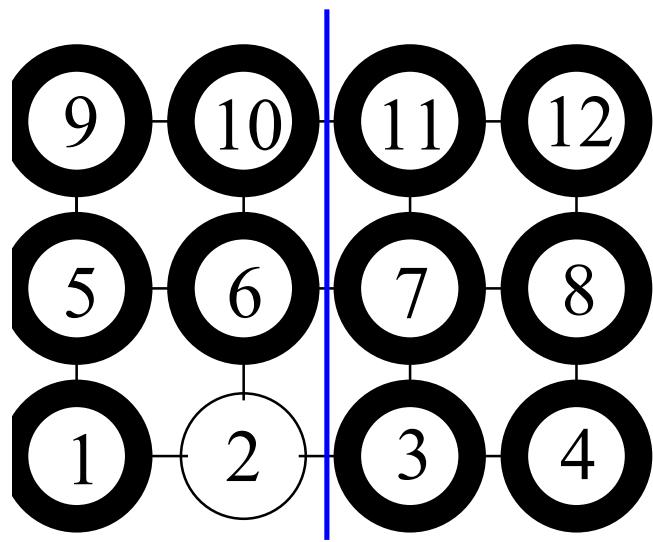


Job

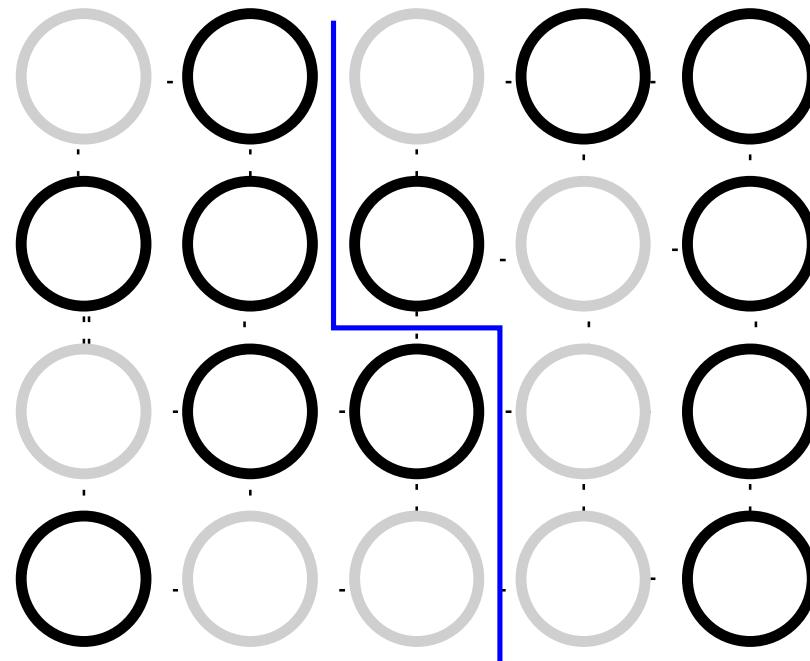


Machine

# Using recursive coordinate bisection for task mapping (RCB)

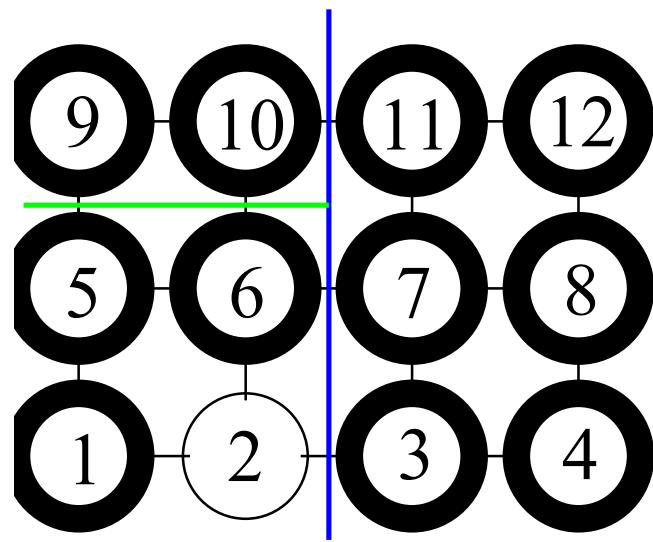


Job

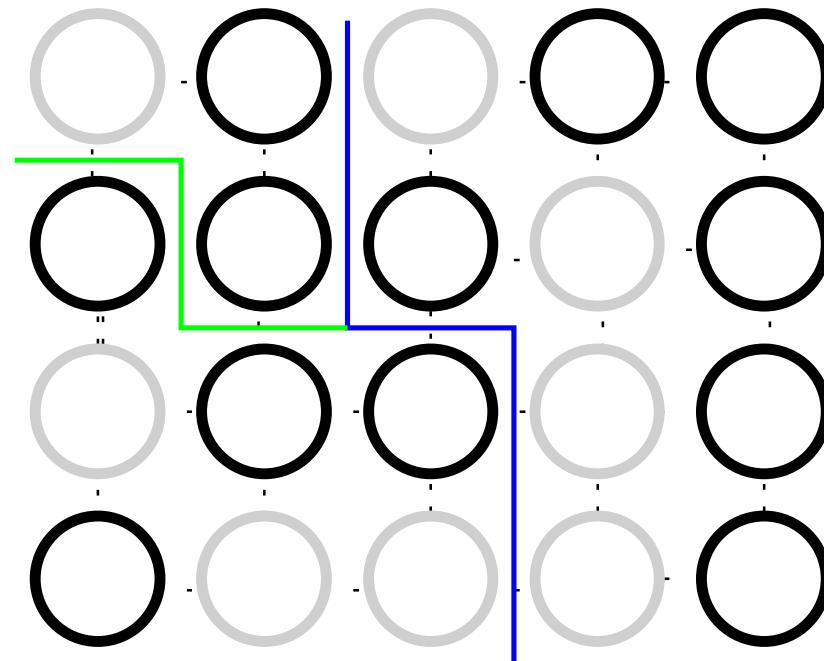


Machine

# Using recursive coordinate bisection for task mapping (RCB)

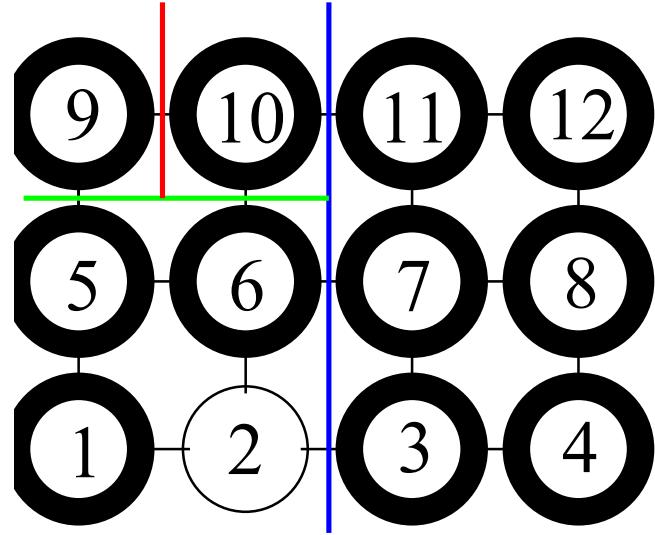


Job

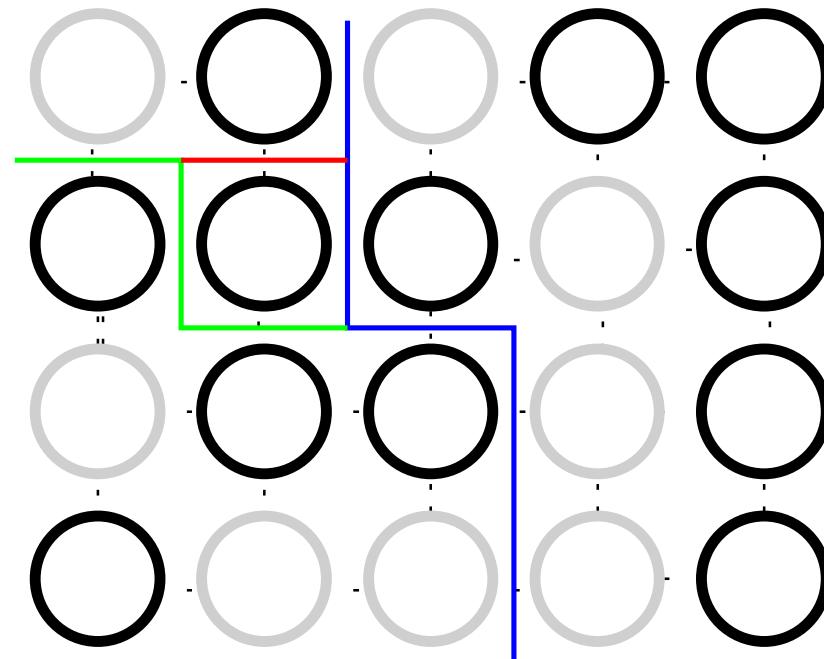


Machine

# Using recursive coordinate bisection for task mapping (RCB)

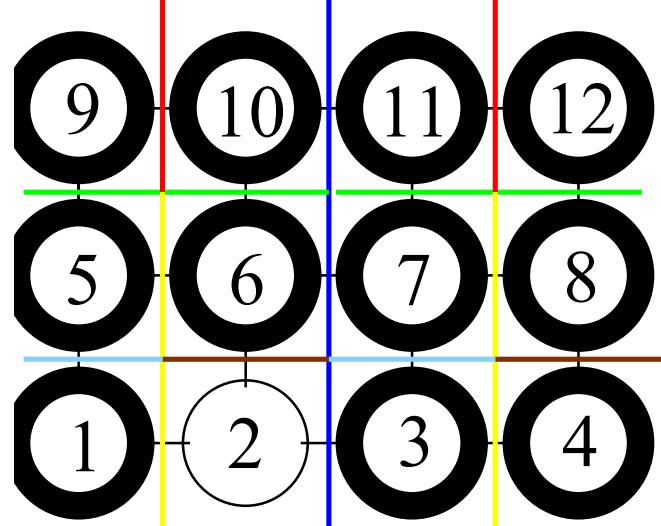


Job

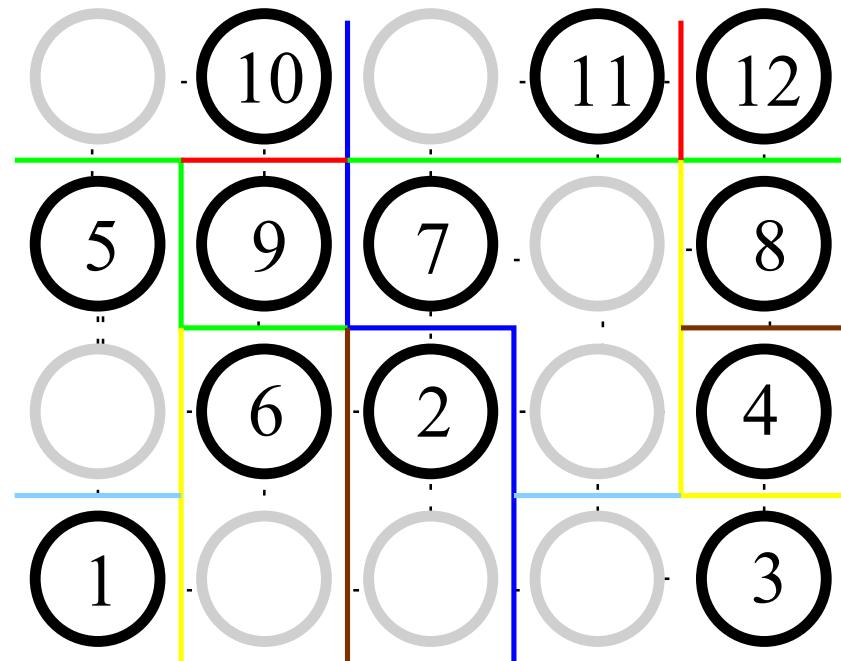


Machine

# Using recursive coordinate bisection for task mapping (RCB)

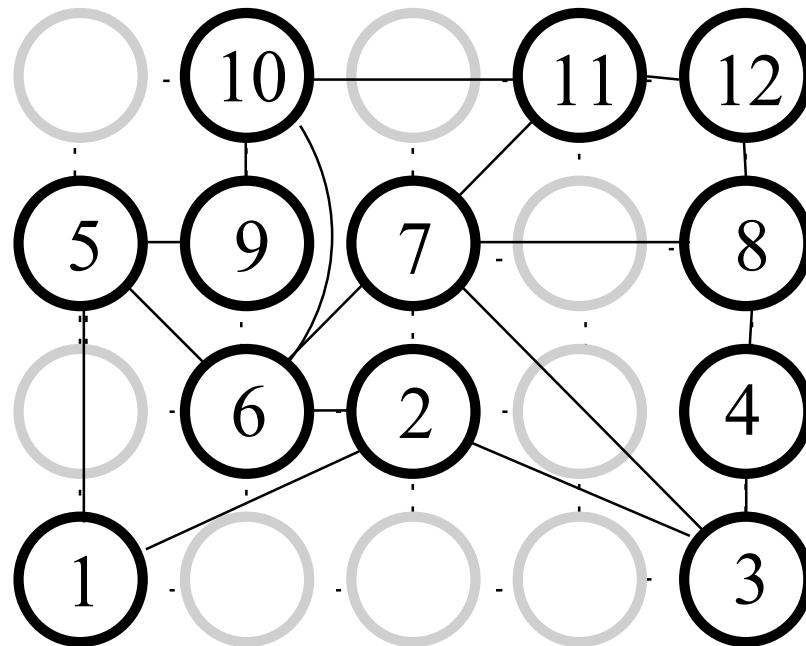
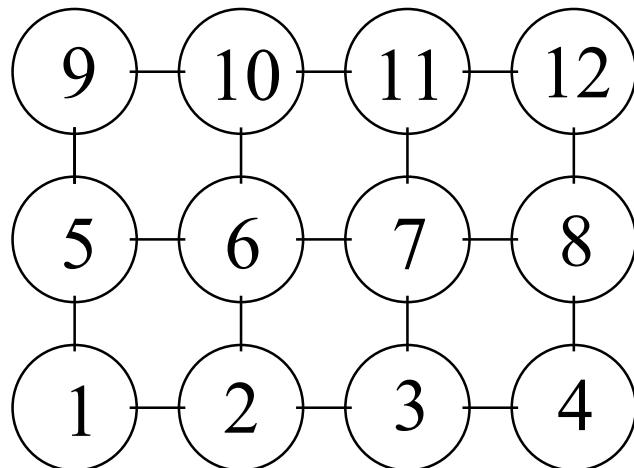


Job

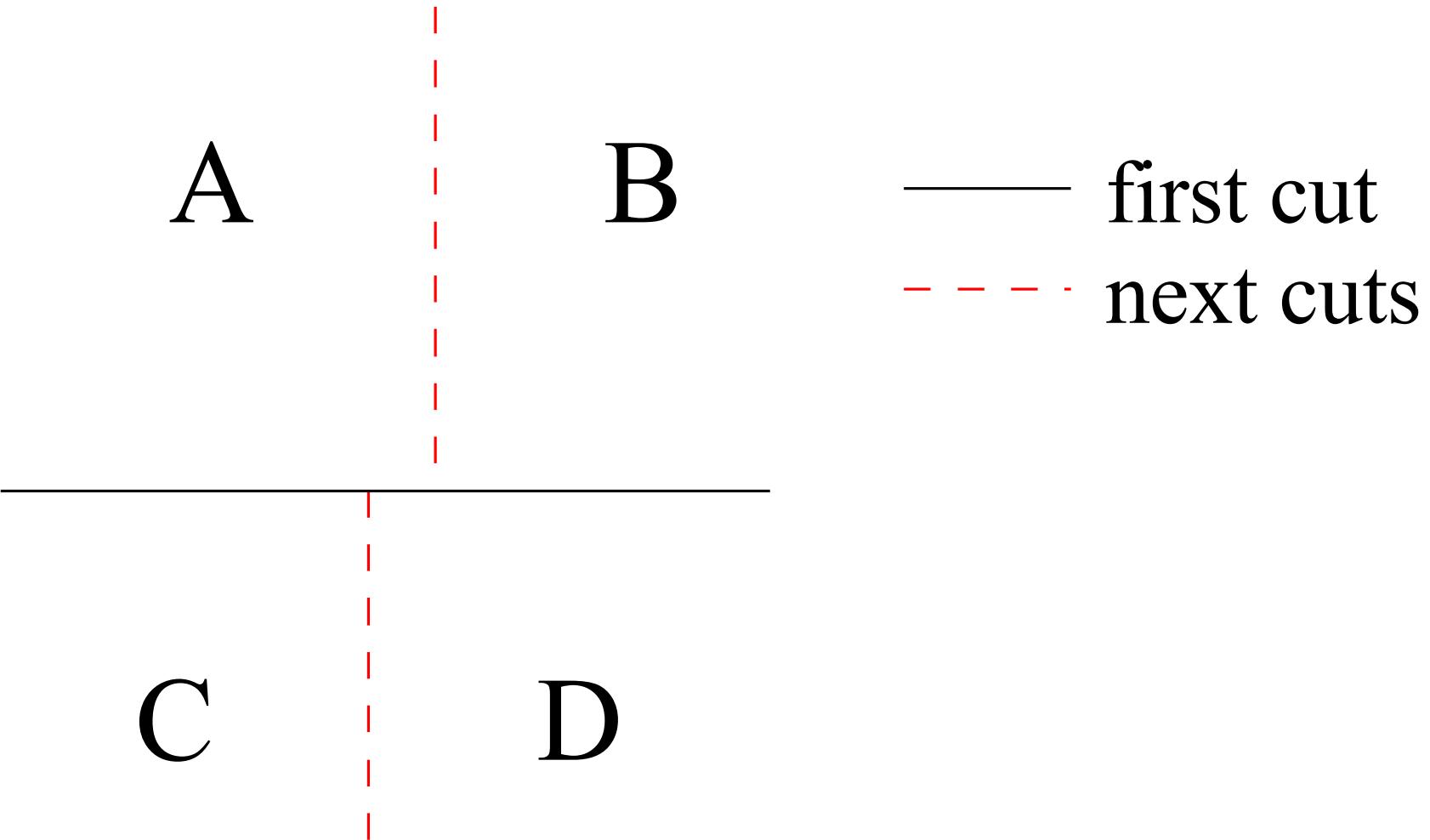


Machine

# Using recursive coordinate bisection for task mapping (RCB)



# Two levels of cuts in decomposition created by RCB



# This presentation

- Local search algorithm, GSearch, improves on RCB by swapping pairs of tasks when doing so improves average distance between communicating tasks
- Demonstrate GSearch in proxy application improves application's total running time
  - While reducing variability in total running time
- Show number of swaps made by GSearch is reasonable in practice
  - Some processor allocations require more
  - Use distribution of swaps made to provide guidance on when to cut off search and avoid pathological cases
- Demonstrate again that RCB is good task mapping algorithm, but local search can improve upon it

# Pseudocode for search component of GSearch (version without a swap limit)

```
do {
    madeSwap = false;
    for  $1 \leq i < \text{num\_tasks}$ 
        for  $i < j \leq \text{num\_tasks}$ 
            if(swapping tasks  $i$  and  $j$  reduces average hops) {
                make the swap
                madeSwap = true;
            }
    } while(madeSwap);
```

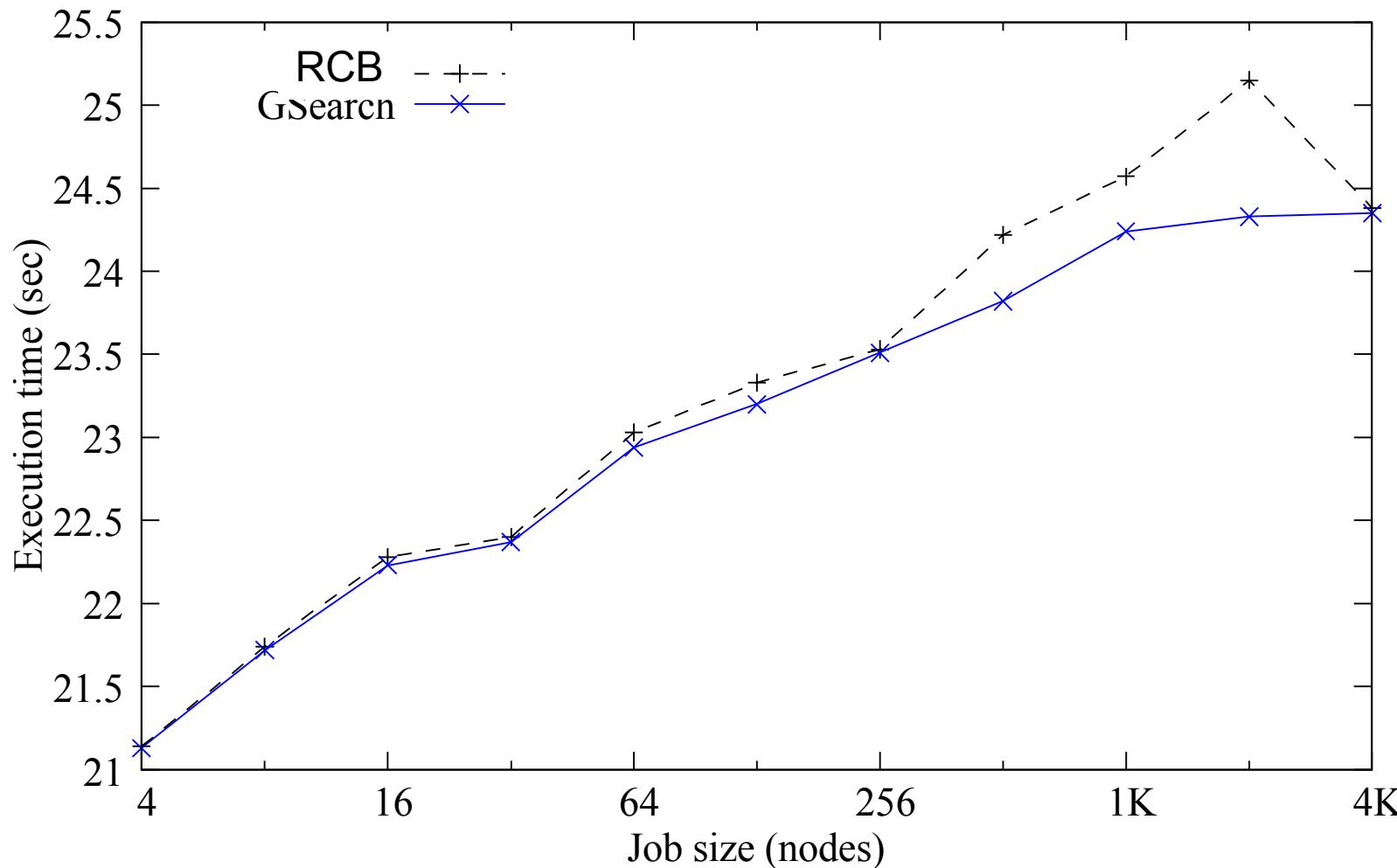
# Cielo miniGhost Experiments

- Los Alamos National Laboratory Cielo machine, Cray XE6
  - 143,104 compute cores in 8,944 compute nodes, dual AMD Opteron 6136 eight-core “Magny-Cours” socket G34 running at 2.4 GHz
  - 272 service nodes, AMD Opteron 2427 six-core “Istanbul” socket F running at 2.2 GHz
  - Gemini 3D torus in 16x12x24 (XYZ) topology, 2 compute nodes (sockets) per Gemini, 6.57x4.38x4.38 (XYZ) TB/s bi-section bandwidth
  - As of November 2013, number 26 on top 500 list
- Application used was miniGhost
  - Boundary exchange using stencil computations in scientific parallel computing, bulk-synchronous message passing code modeled on CTH
- Set of experiments consists of miniGhost runs for various numbers of total cores (16 cores per MPI rank)

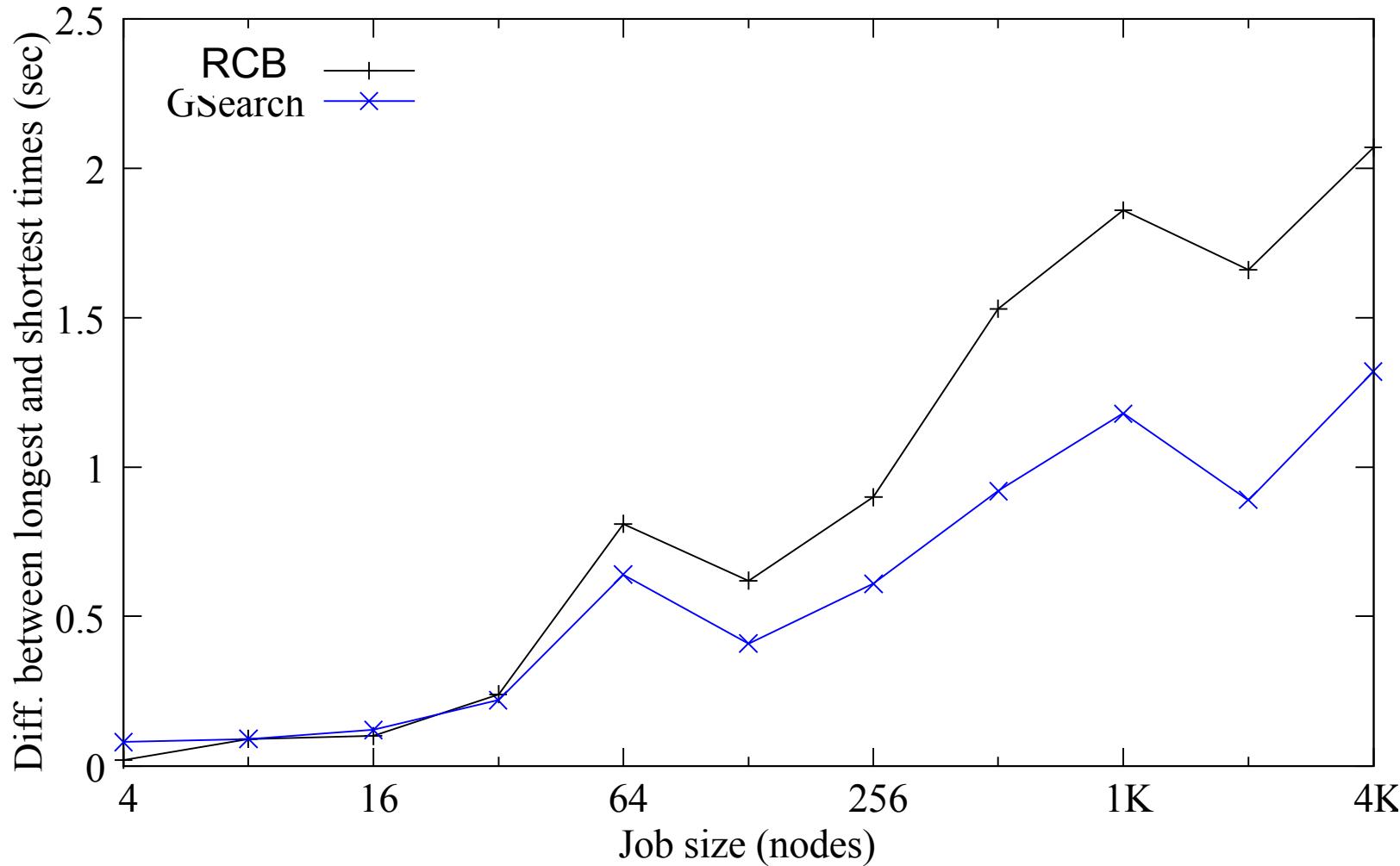
# Job Dimensions used in miniGhost experiments

Nodes	Job Dimensions
4	1 x 4 x 1
8	2 x 4 x 1
16	2 x 4 x 2
32	2 x 8 x 2
64	4 x 8 x 2
128	4 x 8 x 4
256	4 x 16 x 4
512	8 x 16 x 4
1k	8 x 16 x 8
2k	8 x 32 x 8
4k	16 x 32 x 8

# Running time by job size for miniGhost on Cielo (Average over 6 sets of experiments)



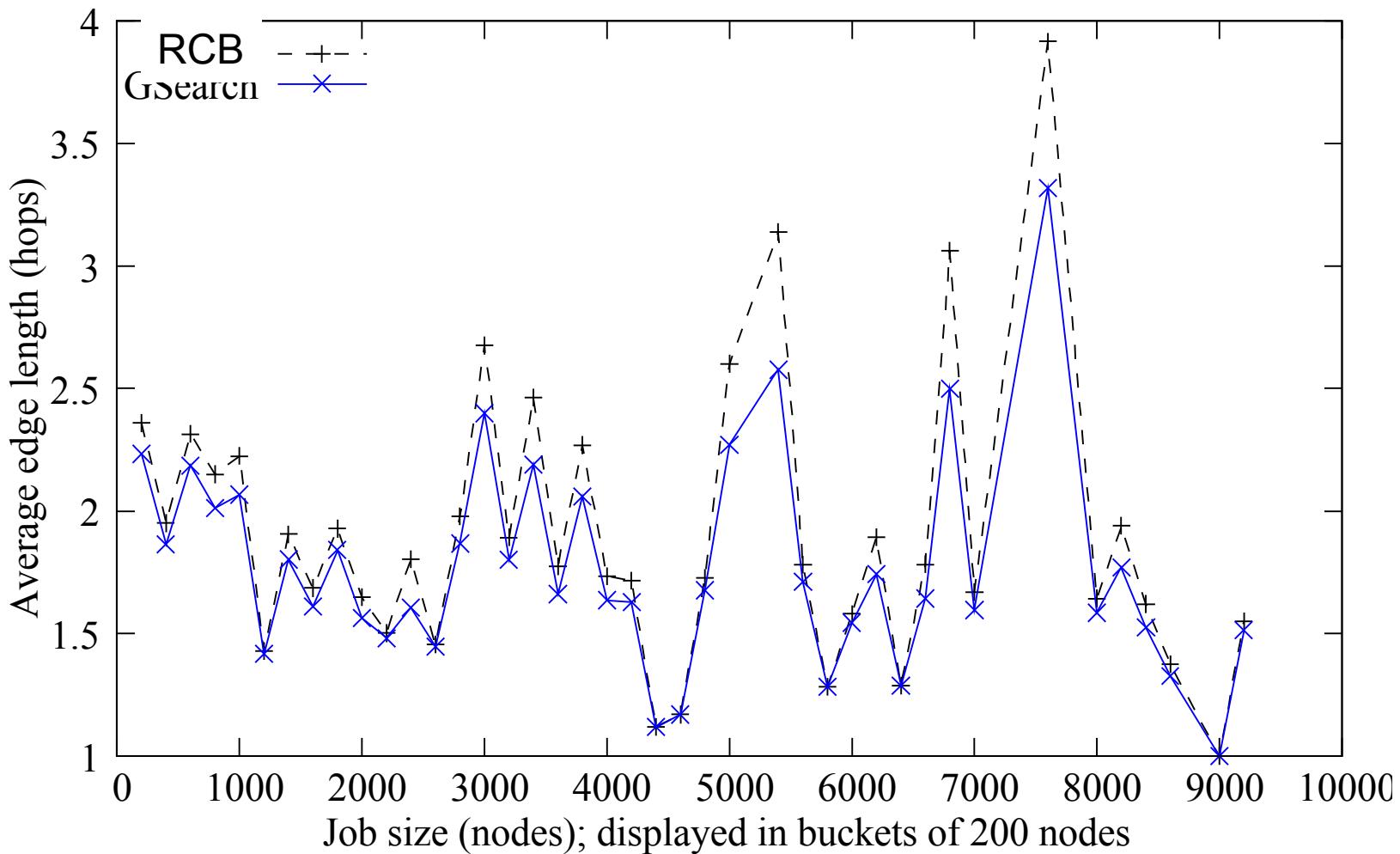
# Difference between max and min running time by job size for miniGhost on Cielo



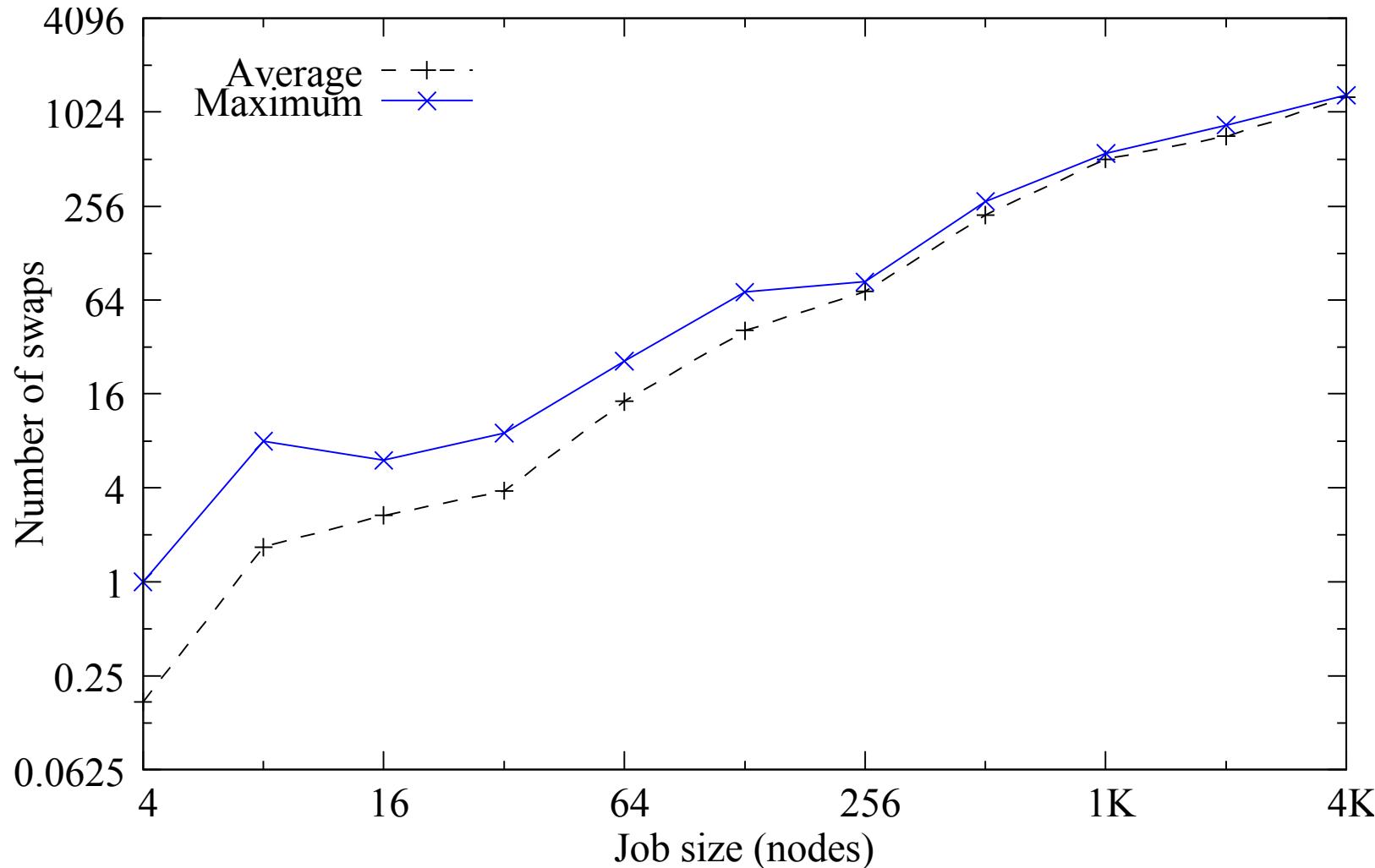
# Simulated miniGhost Experiments

- Since time on large systems is scarce
- Trace-based simulations of more varied scenarios (PWA)
  - Job arrival time, size, running time, and (in many cases) time estimate
  - On machine
    - schedule (EASY),
    - allocate (snake best fit [Lo et al. 1997 and Leung et al. 2002]), and
    - map
  - Summary of trace used in simulations
    - Log name: LLNL-Atlas-2006-2.1-cln, Machine: 24x24x16,
    - # jobs used: 12,474
- Random simulations
- Exhaustive simulations

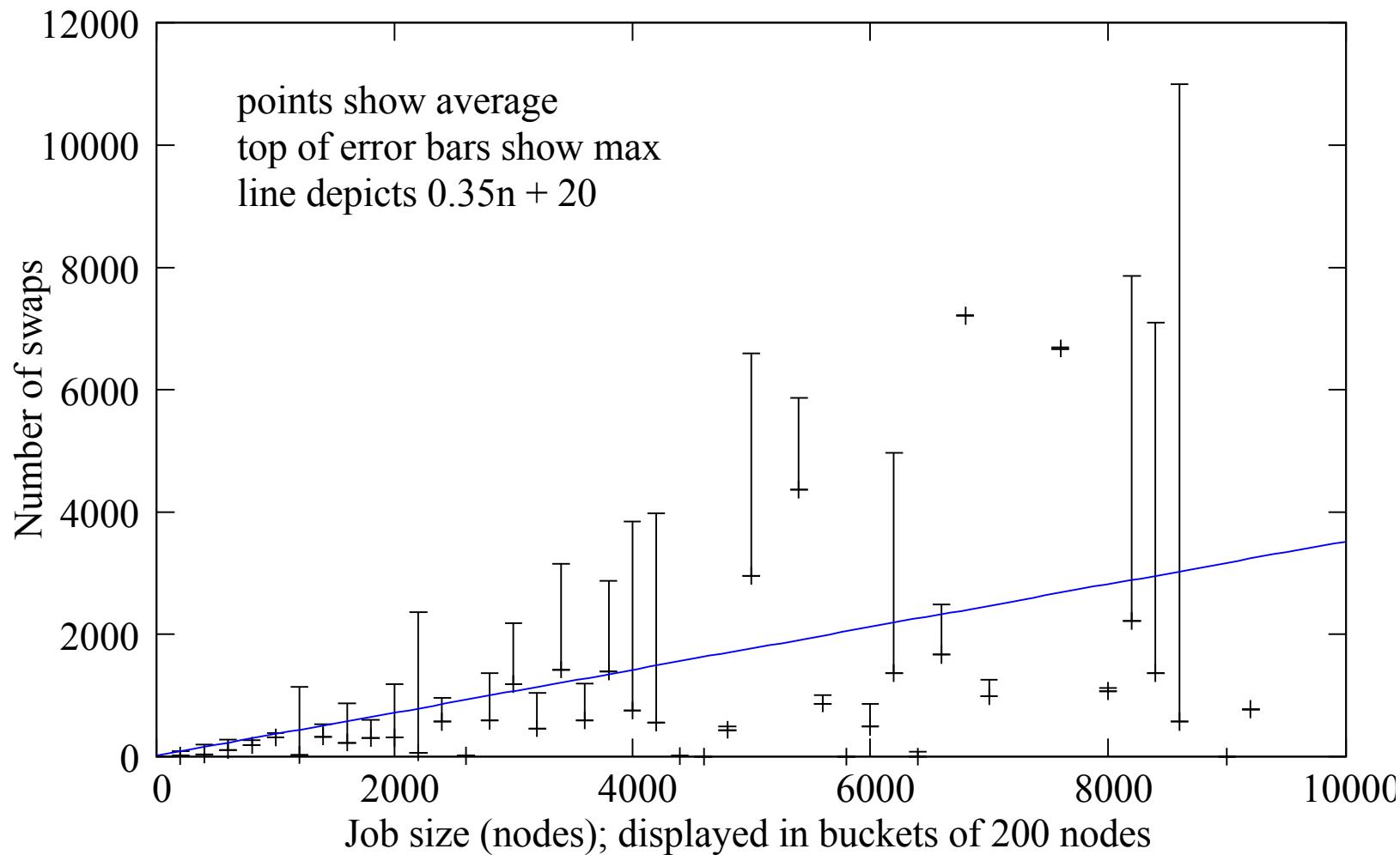
# Average edge length by job size for LLNL-Atlas trace



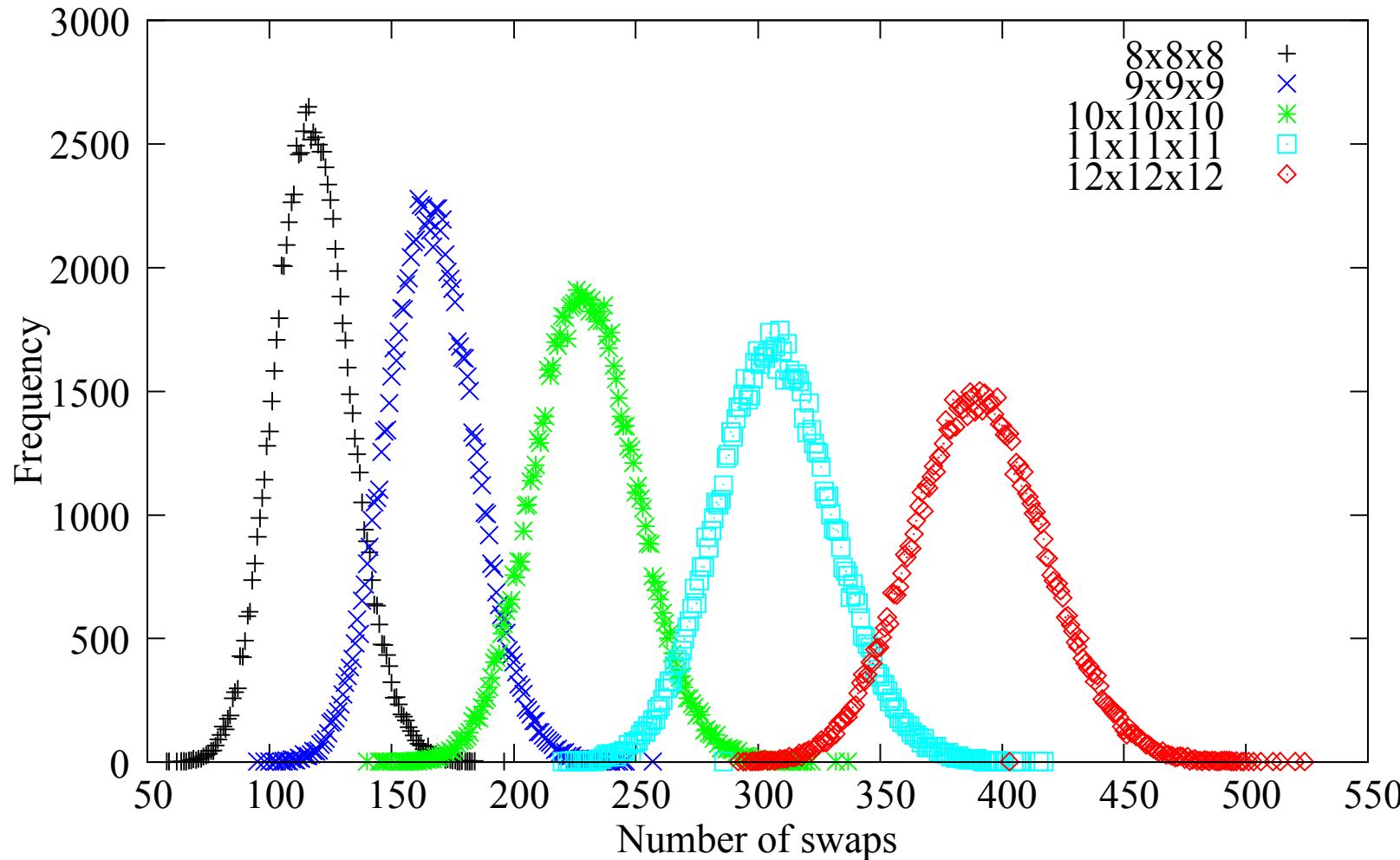
# Number of swaps made by GSearch as a function of job size (average and max)



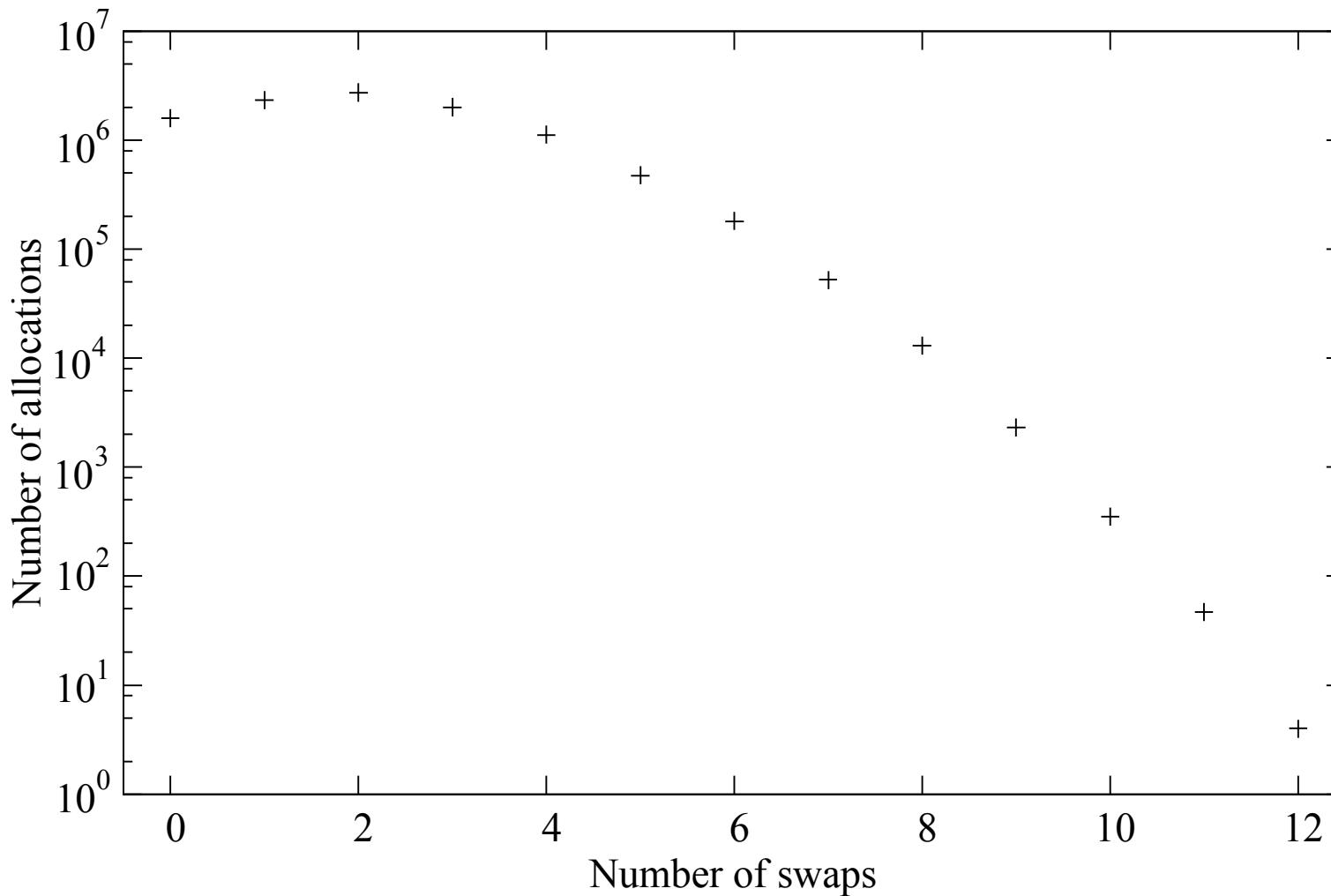
# Number of swaps made by GSearch as a function of job size on LLNL-Atlas trace



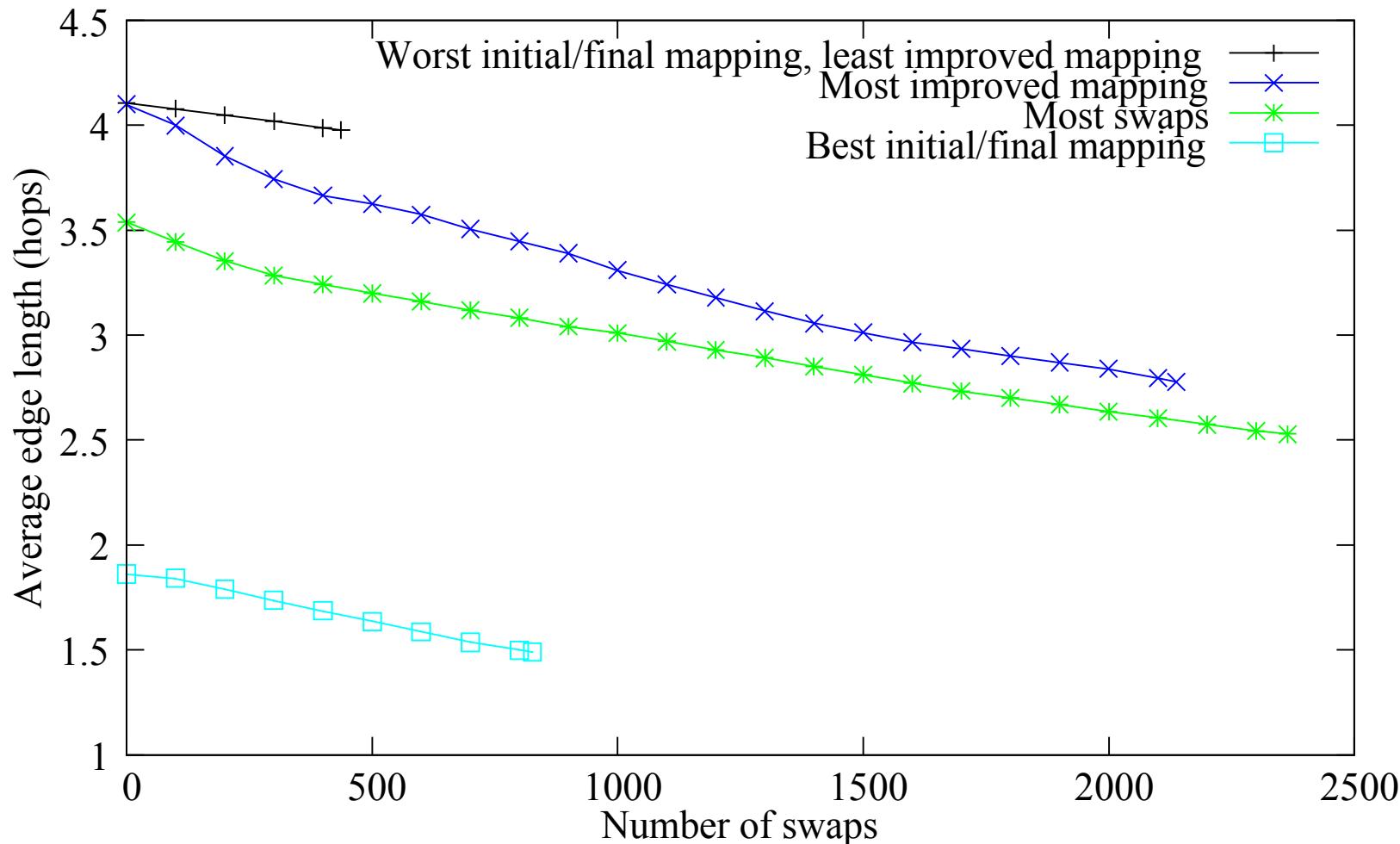
# Swap count frequencies from 100,000 random allocations on $16 \times 24 \times 24$ system



# Swap count frequencies from all possible allocations of $4 \times 2 \times 1$ job on $4 \times 4 \times 2$ system



# Average edge length as function of number of swaps made on trace jobs



# For more information:

- For more information:
  - [vjleung@sandia.gov](mailto:vjleung@sandia.gov)
- Thanks to:
  - NNSA ASC program
  - Sandia National Laboratories LDRD program
- Questions?



# Rest of Minisymposium

- Demonstrating Improved Application Performance Using Dynamic Monitoring and Task Mapping, Gentile
- Process Mapping onto Complex Architectures and Partitions Thereof, Pellegrini
- Topology Aware Process Placement and Data Management, Jeannot

## Lunch

- Locality for Sparse Unstructured Communication Patterns, Tuncer
- Topology Aware Mapping using Graph Models for Exascale Systems, Deveci
- Maximizing Throughput on a Dragonfly Network, Bhatele