

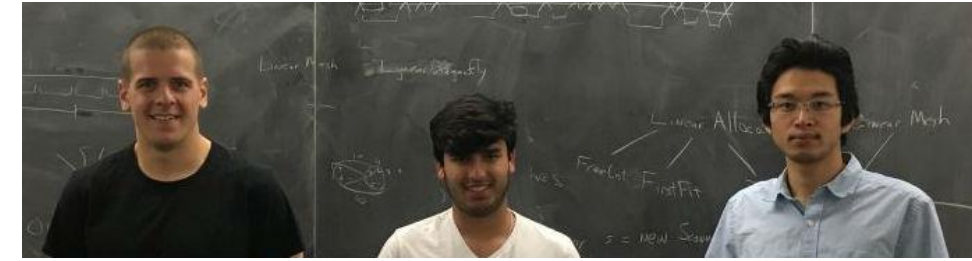


Allocation Algorithms for the Dragonfly Topology

Harry Carpenter, Rishav Sharma, Shogo Akiyama

hhcarpenter@knox.edu, rsharma@knox.edu, sakiyama@knox.edu

Mentor: David Bunde, dbunde@knox.edu



Introduction

We studied allocation algorithm performance in the context of parallel computing systems with over 10,000 processors running parallel jobs. We studied the Dragonfly machine, which is a hierarchical network with three levels: switch, group and system. Computing nodes are attached to switches, switches are organized into groups, and a number of groups compose the overall system. Our goal was to reduce the use of “extra groups” and “extra switches”, those used beyond the number required to get enough nodes for the job. We implemented several allocation algorithms based on a variety of approaches and evaluated them with simulations using traces from real system logs in the Parallel Workloads Archive. We discovered that one of our algorithms was greatly superior to the others, always requiring the fewest extra groups and nearly always requiring the fewest extra switches. This algorithm uses a multi-step greedy approach, making a series of decisions to handle easy cases.

Linear Allocations

Best

Looks for the best interval of nodes and allocates the job.

First

Utilizes the first interval of free processors that is big enough.

Free

Assigns the first available processors based on order specified when allocator is created.

Does not care about intervals.

Looks for free nodes and allocates the job in it.

Buddy System Allocations

Description

- All groups as one block that can split into smaller groups based on job size.
- Checks whether a job can be allocated into a smaller block. If it can, it splits the block.
- Allocates the job with the smaller block. After the job is done, it merges the blocks again.

DragonflyMBS

If there is a job that doesn't fit the group's size, it splits the group into switches and nodes, and allocates the job there.

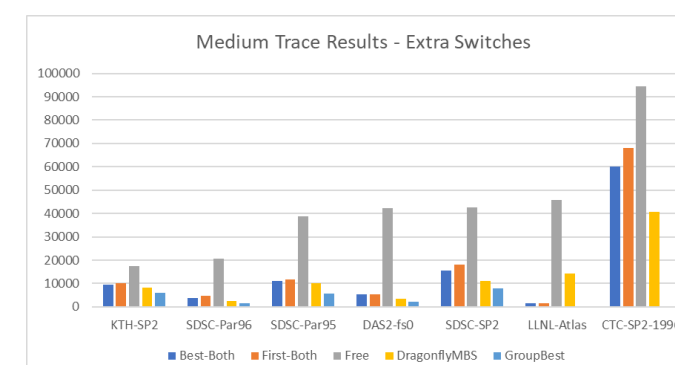
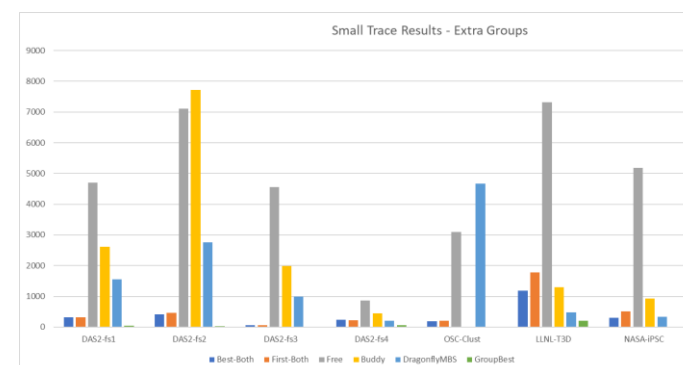
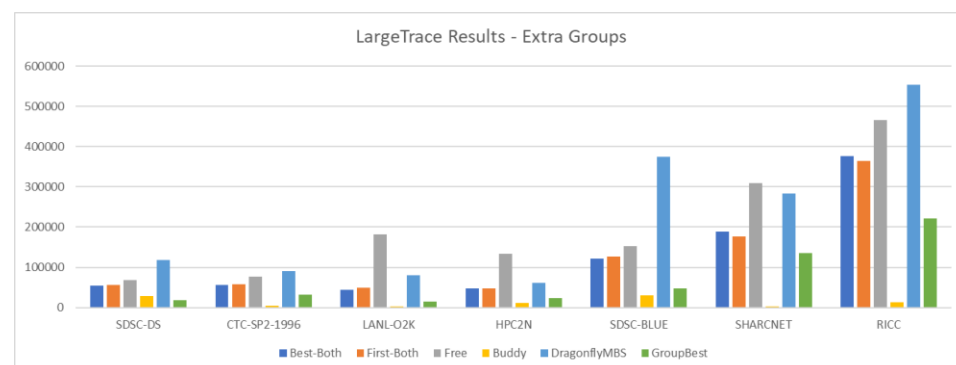
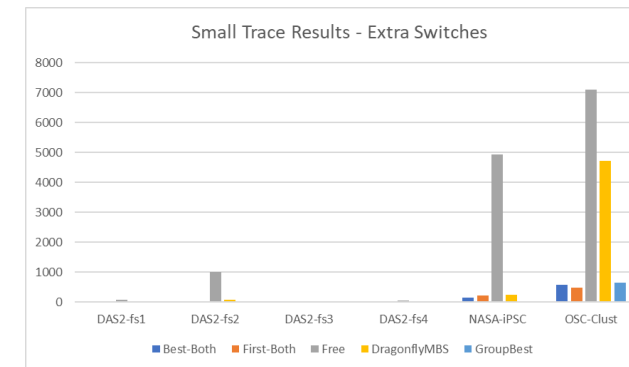
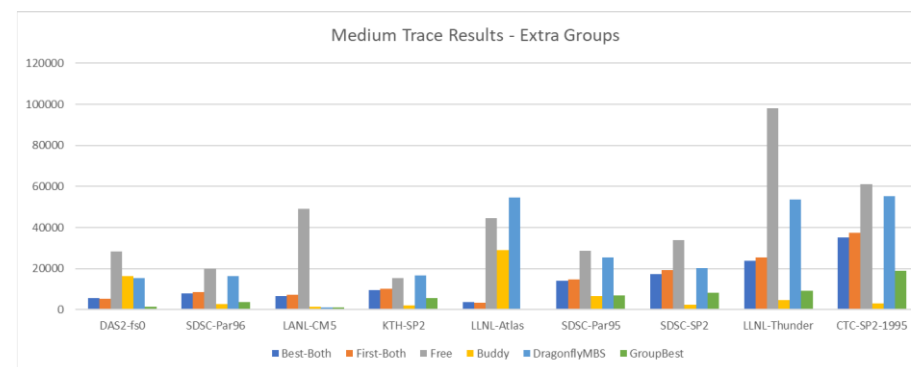
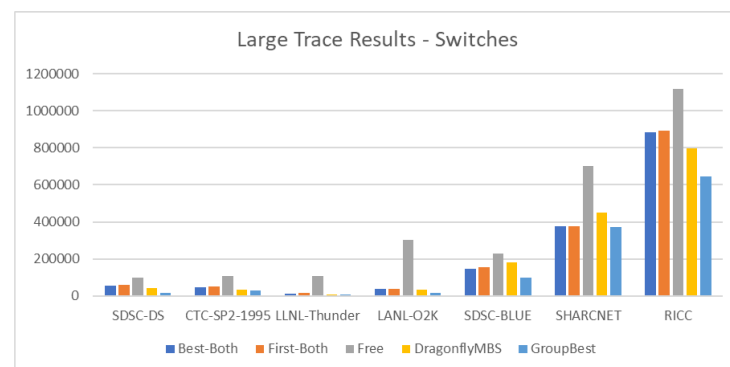
Rounding

If there is a job that doesn't fit in a block, rounds the size, using extra nodes

DragonflyGroupBest

- While the job size is larger than a group and there are complete groups, assign it one
- Next assign the job part of the smallest incomplete group that is larger than it.

Results



References

Parallel Workloads Archive,
<http://www.cs.huji.ac.il/labs/parallel/workload/logs.html>

Bunde, David, and Vitus Leung. *PREMAS: Simulator for Resource Management*, 2014.

Hastings, Emily, et al. *A Comparison of Dragonfly Global Link Arrangements*. 2 Feb. 2016

Acknowledgements

- Paul K. and Evalyn Elizabeth Cook Richter Memorial Fund
- National Science Foundation under grant CNS-1423413