

# MPI Collective Algorithm Selection and Quadtree Encoding

Jelena Pješivac-Grbović

Graham E. Fagg, Thara Angskun,  
George Bosilca, Jack J. Dongarra

**Euro PVM/MPI**

**09/20/06**



**INNOVATIVE COMPUTING LABORATORY**

COMPUTER SCIENCE DEPARTMENT  
UNIVERSITY OF TENNESSEE

- » Motivation
- » Decision maps
- » Quadrees
  - » Construction
  - » Evaluation
  - » Code generation
- » Results
- » Discussion and future work

- » MPI collective operations
  - » Frequently used
  - » Can be performance bottleneck
- » MPI collective algorithms
  - » Numerous in literature
  - » Explicit message segmentation
  - » Performance portability issue
- » Tuning collective operations for particular system
  - » **Ideally, automatic tuning**

# Our Approach

## Model:

- » At run time, decision function is invoked to select the "best" algorithm for particular collective call

## Optimization process:

1. MPI collective algorithm performance information (system profiling or performance modeling)
2. Decision map generation
3. Construction of decision trees
4. Decision function automatic code generation

## Model:

- » At run time, decision function is invoked to select the "best" algorithm for particular collective call

## Optimization process:

1. MPI collective algorithm performance information (system profiling or performance modeling)
2. Decision map generation
3. Construction of decision trees
4. Decision function automatic code generation

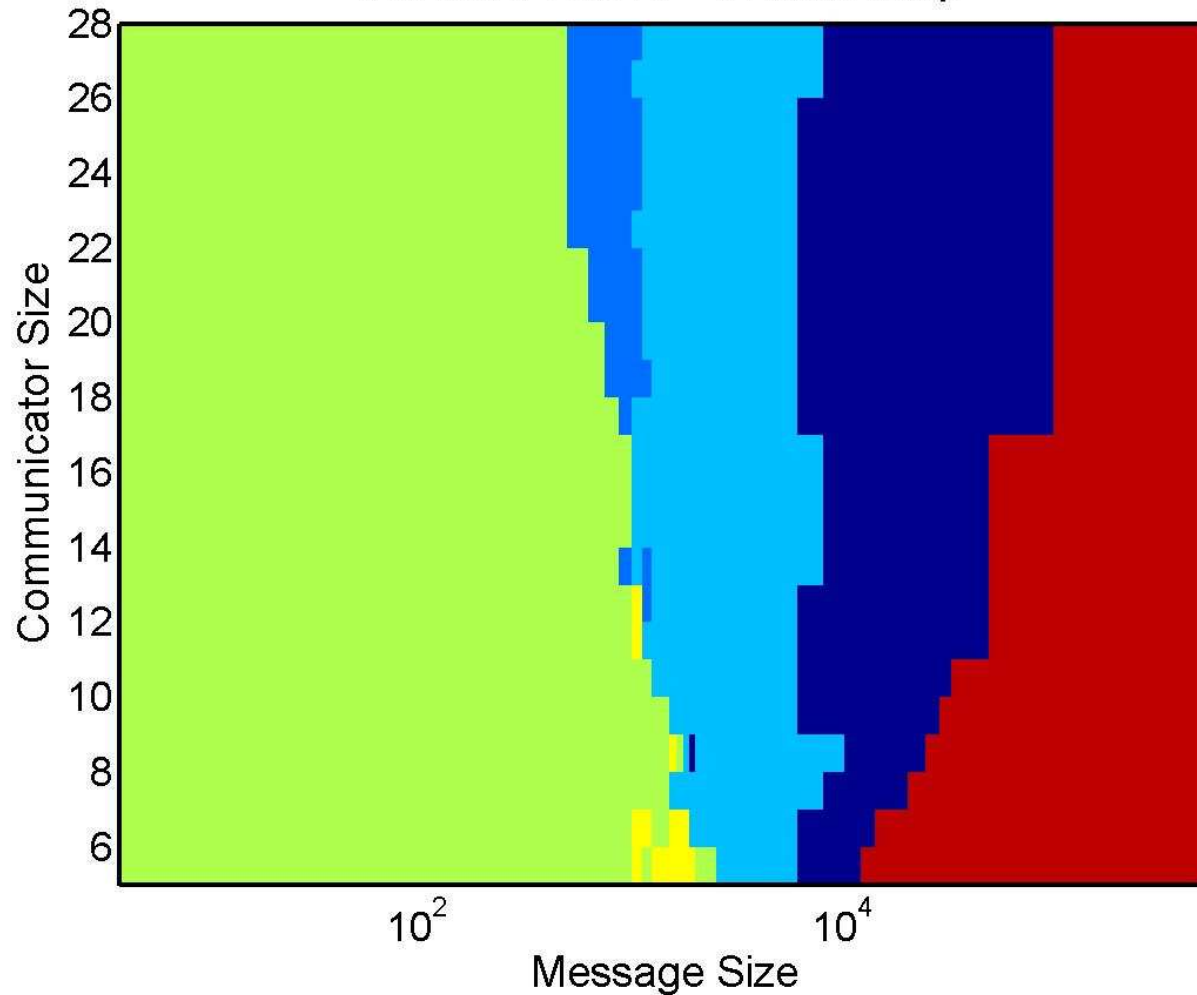
# Decision Map

- » Mapping from input parameter domain to (algorithm, segment size) domain.
- » In current work:
  - » Input parameter domain: **communicator and message sizes**
  - » (algorithm, segment size) pair is **method**.
  - » Mapping function: minimal duration

<b>Communicator size</b>	<b>Message size</b>	<b>Algorithm</b>	<b>Segment Size</b>	<b>Method</b>
3	1	Linear	None	1
3	16	Linear	None	1
...	...	...	...	...
128	256KB	Binary	8KB	13

# Decision Map: Reduce on Grig

Reduce Exact Decision Map



Linear, no seg.

Linear, 1KB.

Binomial,  
no seg.

Binomial, 1KB.

Binary, 1KB.

Pipeline, 1KB.

32 dual CPU nodes  
Intel® Xeon™ 3.2GHz  
Fast Ethernet

# Why Quadtrees?

- » Simple
  - » Construction
  - » Automatic decision function generation
- » Simultaneous search in 2-D space
- » Supports different constraints
  - » Tradeoff between accuracy and tree size

# Constructing Decision Quadtree

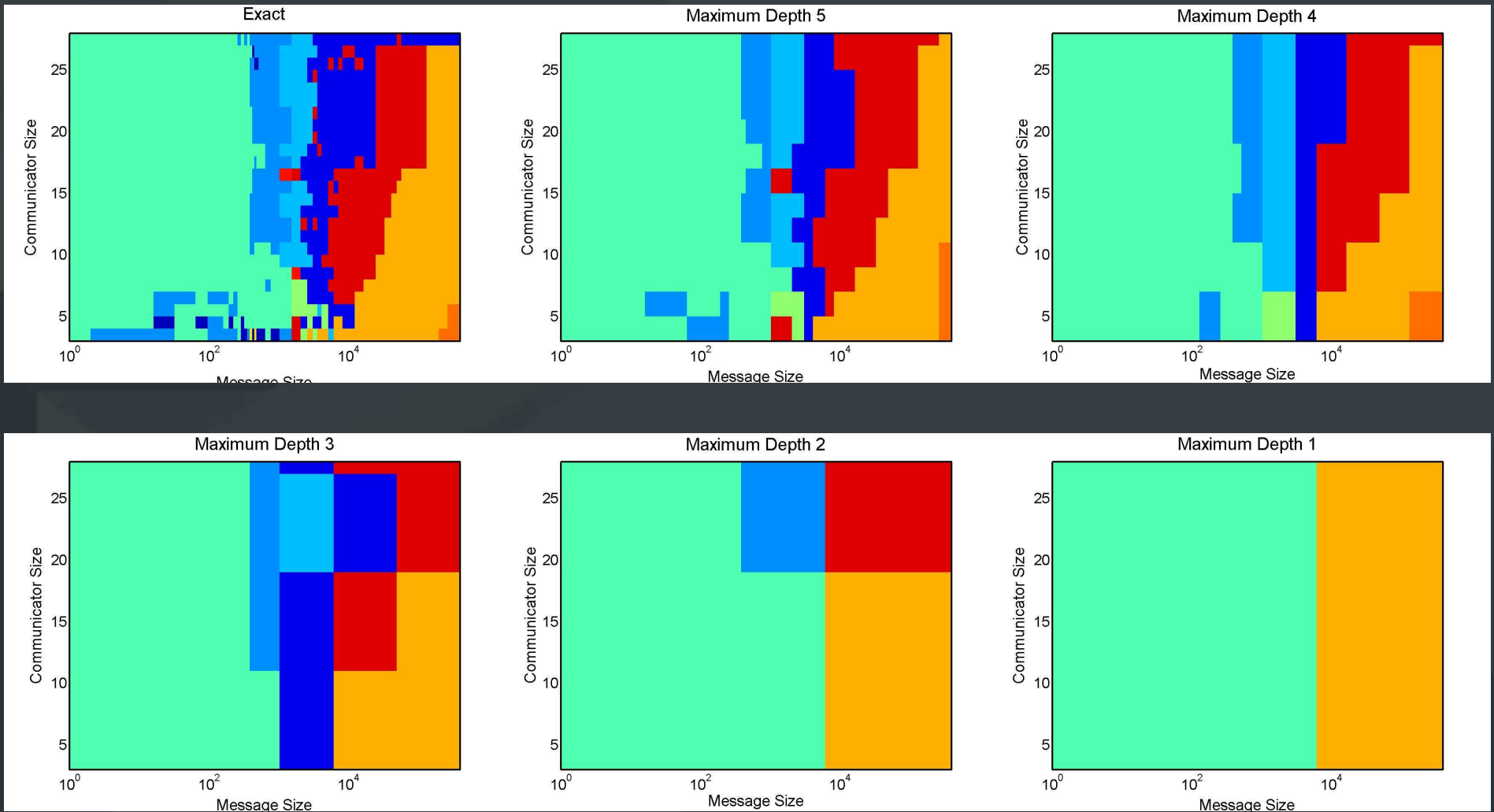
- » Constraints on input data (minor inconveniences)
  - » Complete
  - »  $2^k \times 2^k$  format
- » Constraints to build a tree (flexibility)
  - » Maximum tree depth
  - » Region accuracy threshold



# Maximum Tree Depth

## Broadcast on Grig cluster

32 dual CPU nodes  
Intel® Xeon™ 3.2GHz  
Fast Ethernet

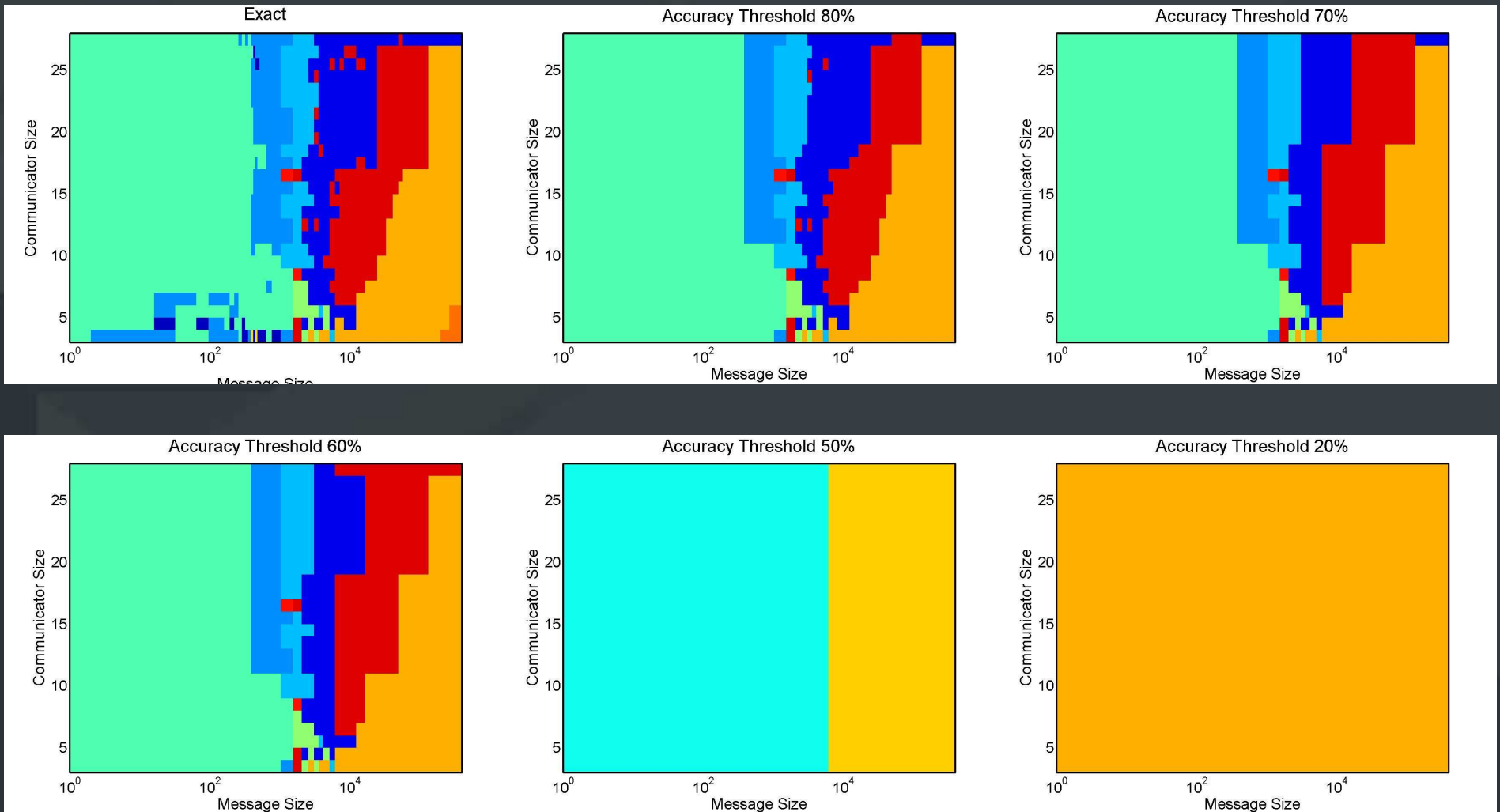




# Region Accuracy Threshold

Broadcast on Grig cluster

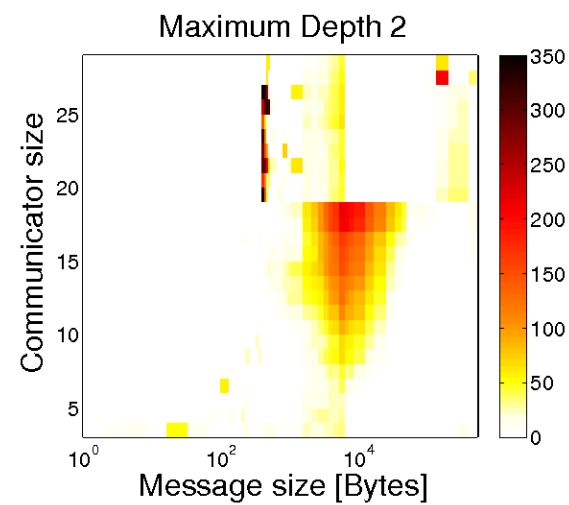
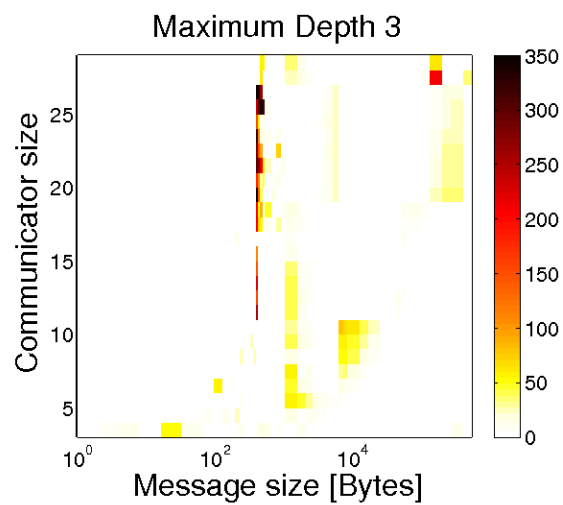
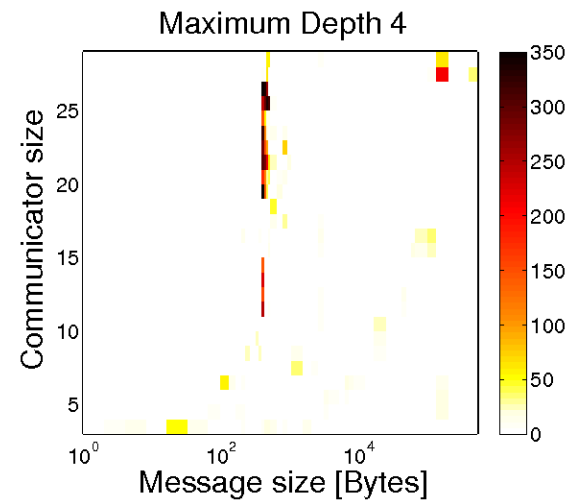
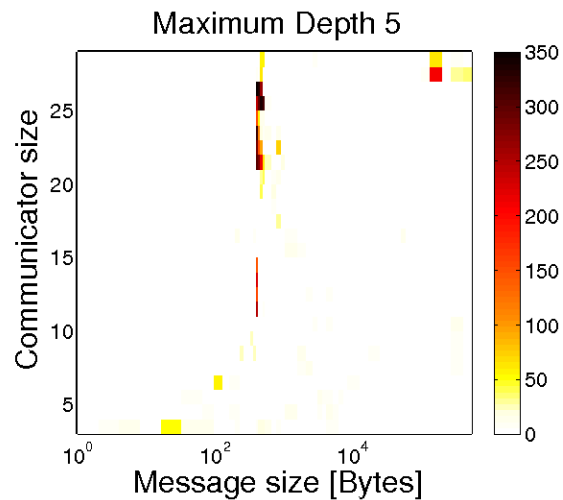
32 dual CPU nodes  
Intel® Xeon™ 3.2GHz  
Fast Ethernet



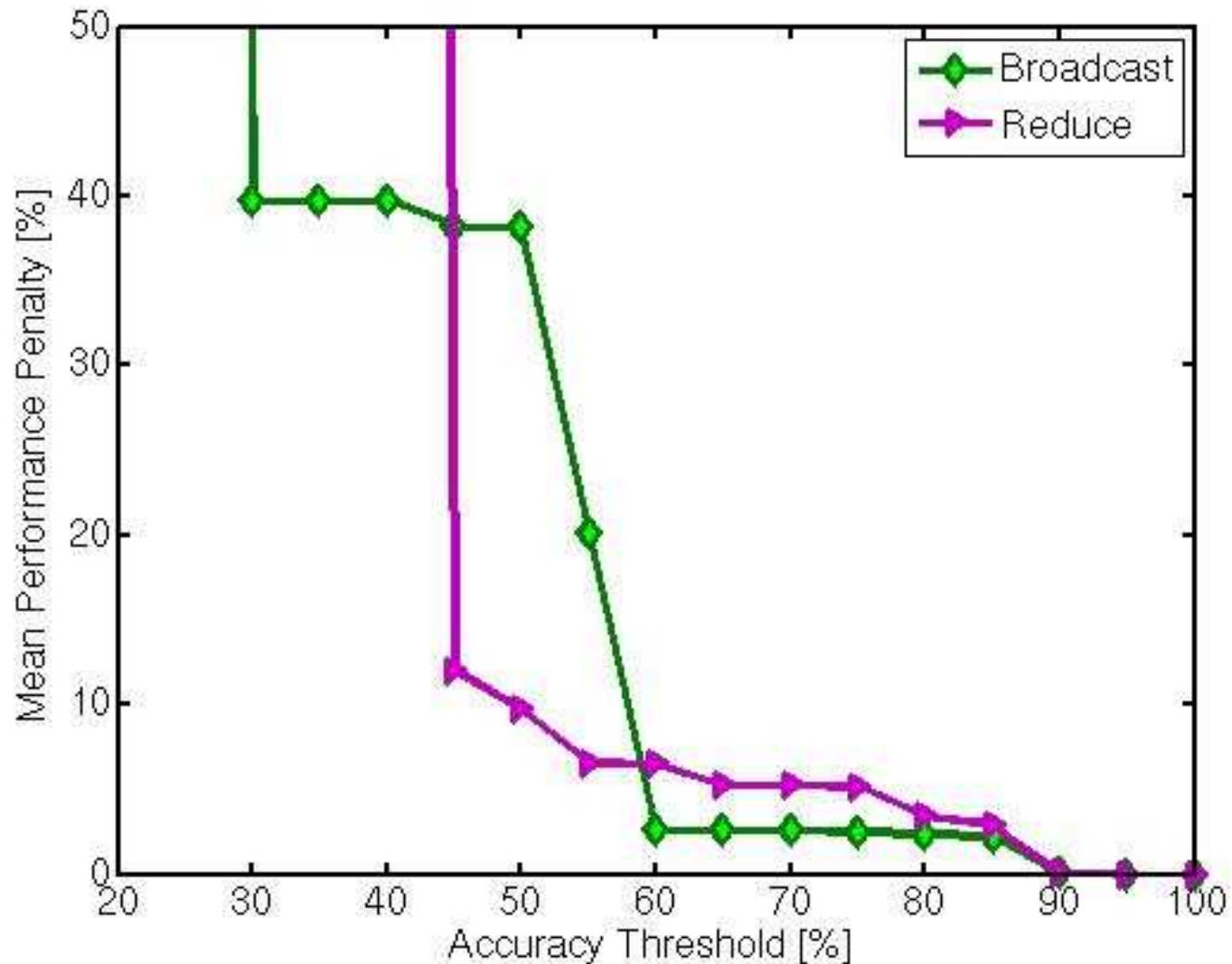
- » Performance penalty
  - » What is performance penalty we will incur by using quadtree decision function instead of exact one?
- » Quadtree size vs. Performance penalty
- » Maximum depth vs. Region accuracy threshold

# Results: Broadcast on Grig Cluster

## Performance Penalty and Maximum Tree Depth

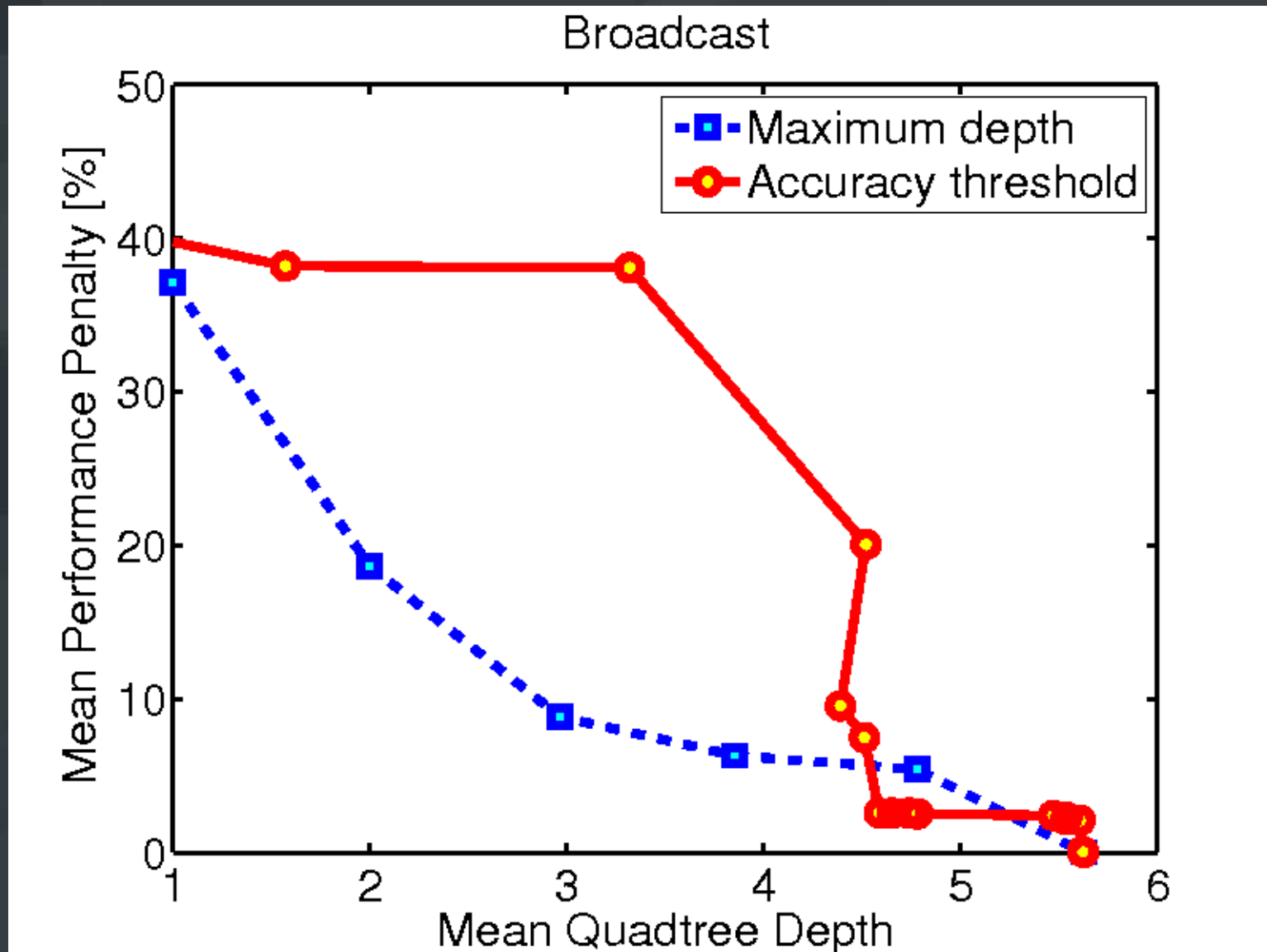


# Results: Broadcast and Reduce on Grid Cluster Performance Penalty and Accuracy Threshold



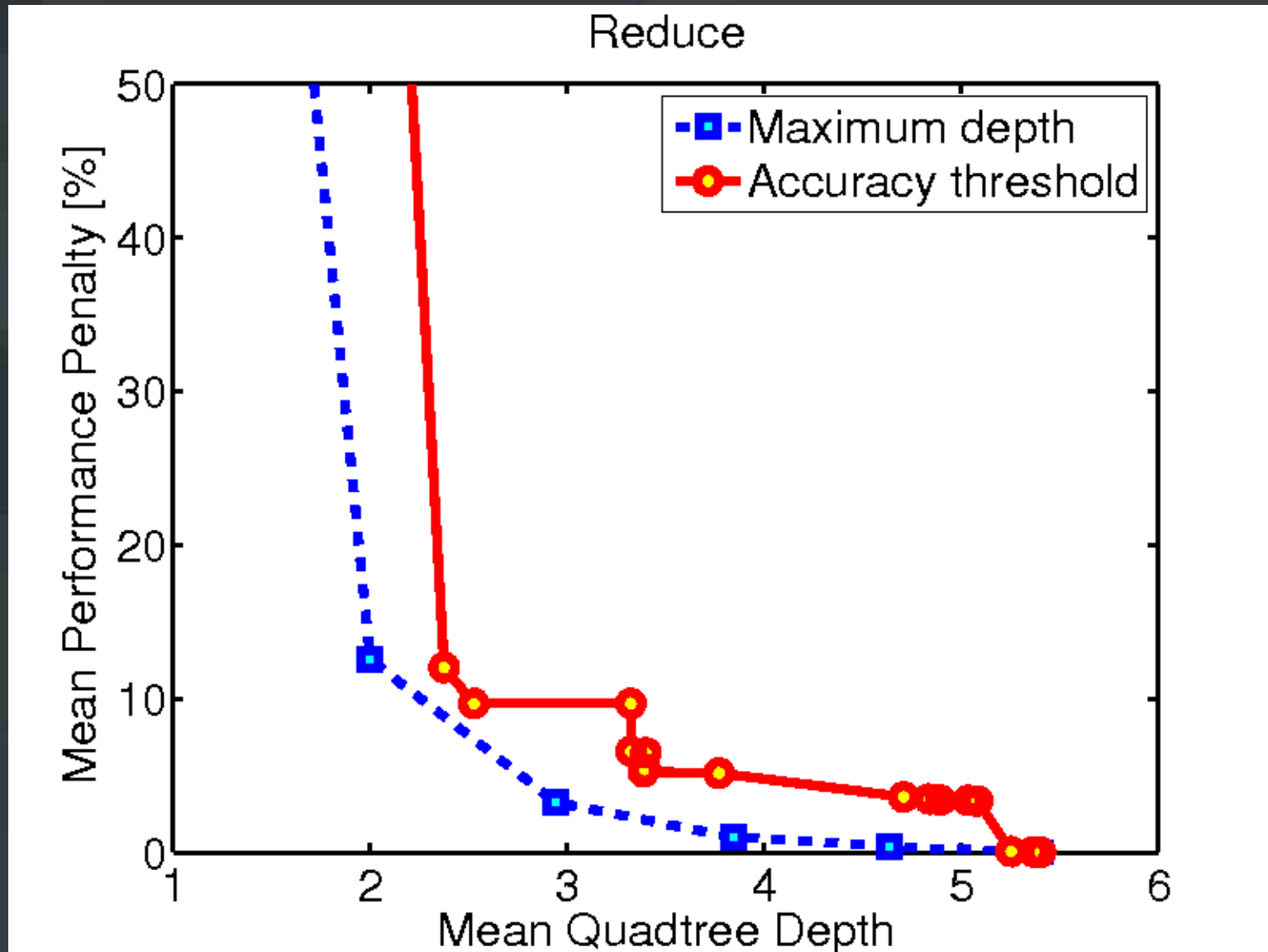
# Results: Broadcast on Grig Cluster

## Performance Penalty vs. Quadtree Size



# Results: Reduce on Grig Cluster

## Performance Penalty vs. Quadtree Size

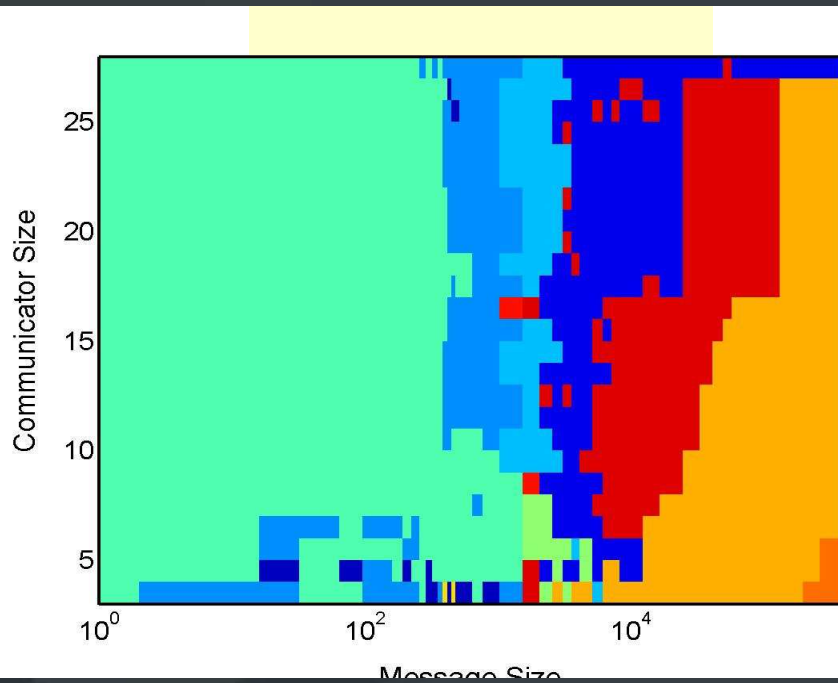




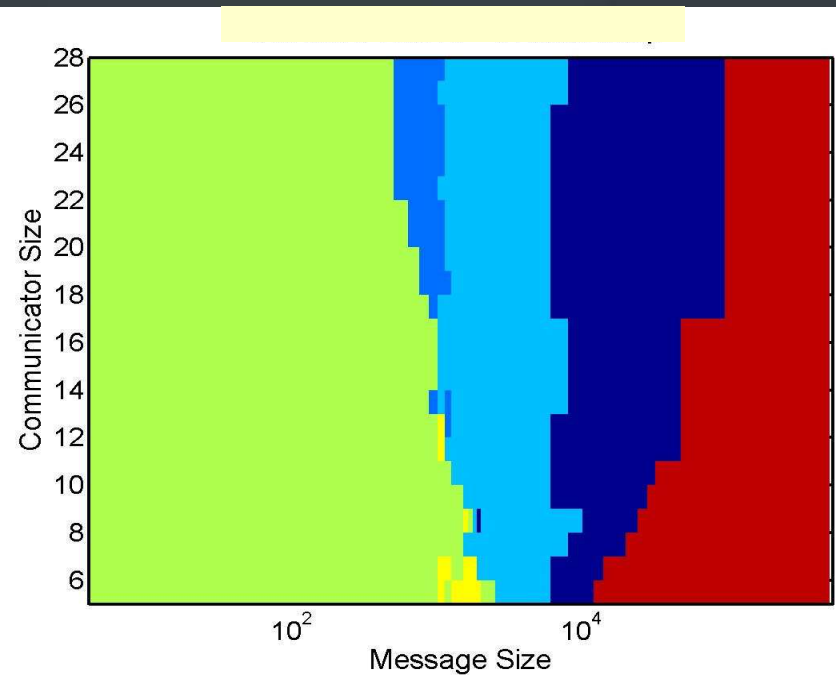
# Results: Accuracy Threshold vs. Maximum Depth

## Where Does the Difference Come From?

**Broadcast, Grig**



**Reduce, Grig**



32 dual CPU nodes  
Intel® Xeon™ 3.2GHz  
Fast Ethernet

# Limitations

- » Considers only communicator and message size
  - » Input parameter set cannot be extended for quadtrees
  - » However, there are octrees...
- » Does not handle “Line” cases elegantly
  - » E.g. methods which are good only for power of two communicator size.
- » Quality of encoding depends on initial data form and fill-in procedure.

# Summary and Conclusions

- » Evaluated applicability of quadtree encoding to MPI collective algorithm selection problem
- » Quadtree of mean depth 3 incurred 5% average performance penalty
- » Quadtree of maximum depth 6 was able to cover complete decision map for data we collected
- » Maximum depth constraint gives more uniform results
- » Accuracy threshold constraint gives better coverage of noisy decision maps

# Summary and Conclusions

- » Evaluated applicability of quadtree encoding to MPI collective algorithm selection problem
- » Quadtree of mean depth 3 incurred 5% average performance penalty
- » Quadtree of maximum depth 6 was able to cover complete decision map for data we collected
- » Maximum depth constraint gives more uniform results
- » Accuracy threshold constraint gives better coverage of noisy decision maps



Quadtrees may be useful tool in automatic decision function generation

- » Study mainstream decision tree algorithms (e.g. C4.5)
- » Study more efficient structures (recursive quadtrees, octrees).
- » Implement decision module for collectives in Open MPI



Thank you!

Any Questions?



[jpjesiva@utk.edu](mailto:jpjesiva@utk.edu)