CUG 2008 HELSINKI • MAY 5–8, 2008
CROSSING THE BOUNDARIES

A Micro-Benchmark Evaluation of Catamount and Cray Linux Environment (CLE) Performance

Jeff Larkin
Cray Inc.
<larkin@cray.com>

Jeff Kuehn
ORNL
<kuehn@ornl.gov>

UT-BATTELLE
OAK RIDGE NATIONAL LABORATORY

CRAY
THE SUPERCOMPUTER COMPANY

Does CLE waddle like a penguin, or run like a catamount?

THE BIG QUESTION!

Overview

- Background
  - Motivation
  - Catamount and CLE
  - Benchmarks
  - Benchmark System
- Benchmark Results
  - IMB
  - HPCC
- Conclusions

BACKGROUND

Motivation

- Last year at CUG “CNL” was in its infancy
- Since CUG07
  - Significant effort spent scaling on large machines
  - CNL reached GA status in Fall 2007
  - Compute Node Linux (CNL) renamed Cray Linux Environment (CLE)
  - A significant number of sites have already made the change
  - Many codes have already ported from Catamount to CLE
- Catamount scalability has always been touted, so how does CLE compare?
  - Fundamentals of communication performance
    - HPCC
    - IMB
- What should sites/users know before they switch?

Background: Catamount

- Developed by Sandia for Red Storm
- Adopted by Cray for the XT3
- Extremely light weight
  - Simple Memory Model
    - No Virtual Memory
    - No mmap
  - Reduced System Calls
    - Single Threaded
    - No Unix Sockets
    - No dynamic libraries
  - Few Interrupts to user codes
- Virtual Node (VN) mode added for Dual-Core

Background: CLE

- First, we tried a full SUSE Linux Kernel.
- Then, we "put Linux on a diet."
- With the help of ORNL and NERSC, we began running at large scale
- By Fall 2007, we released Linux for the compute nodes
- What did we gain?
  - Threading
  - Unix Sockets
  - I/O Buffering

Background: Benchmarks

- HPCC
  - Suite of several benchmarks, released as part of DARPA HPCS program
    - **MPI performance**
    - Performance for varied temporal and spatial localities
  - Benchmarks are run in 3 modes
    - SP – 1 node runs the benchmark
    - EP – Every node runs a copy of the same benchmark
    - **Global – All nodes run benchmark together**

- Intel MPI Benchmarks (IMB) 3.0
  - Formerly Pallas benchmarks
  - Benchmarks standard MPI routines at varying scales and message sizes

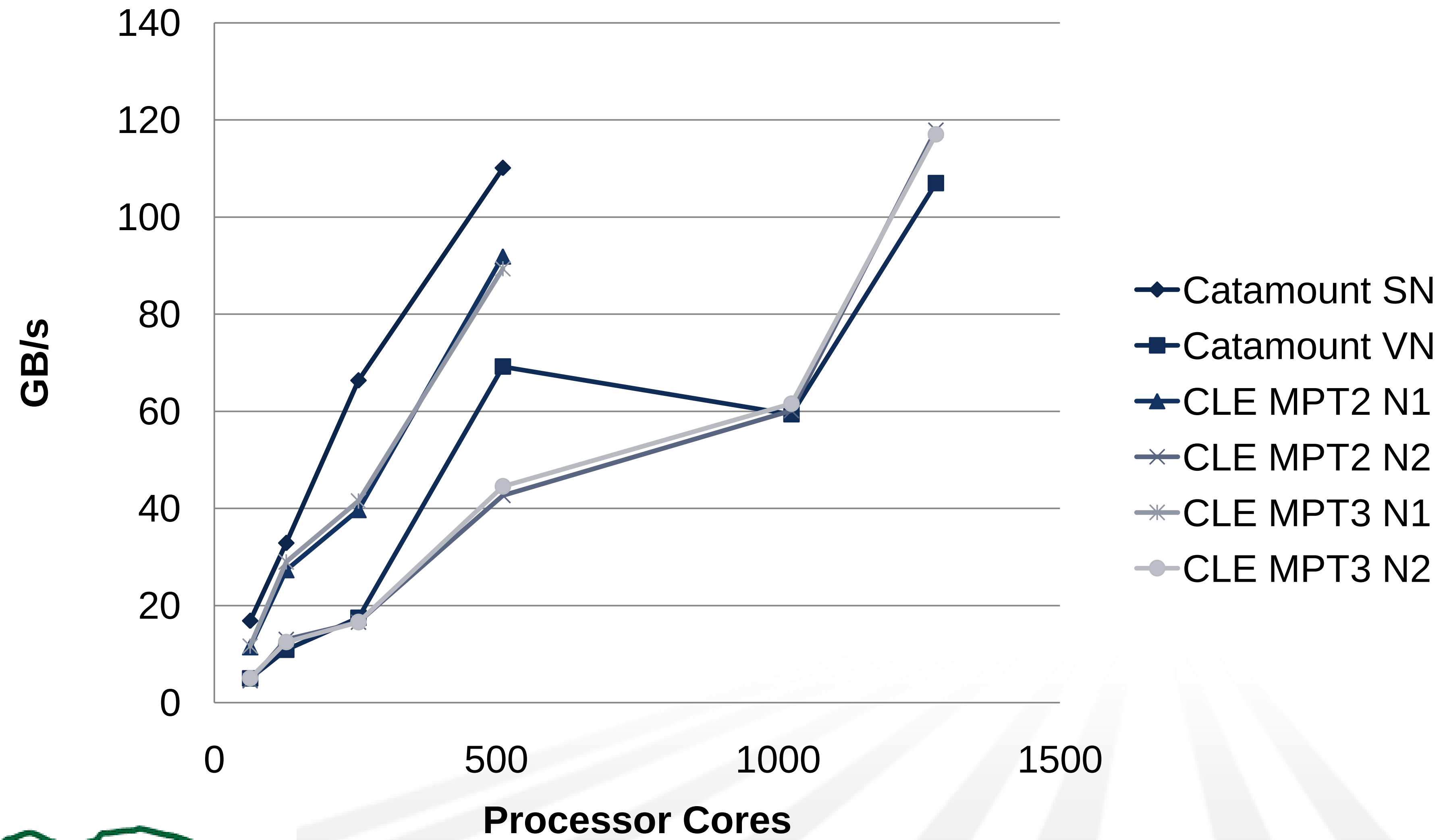
Background: Benchmark System

- All benchmarks were run on the same system, “Shark,” and with the latest OS versions as of Spring 2008
- System Basics
  - Cray XT4
  - 2.6 GHz Dual-Core Opterons (Able to run to 1280 Cores)
  - DDR2-667 Memory, 2GB/core
- Catamount (1.5.61)
- CLE, MPT2 (2.0.50)
- CLE, MPT3 (2.0.50, xt-mpt 3.0.0.10)

BENCHMARK RESULTS

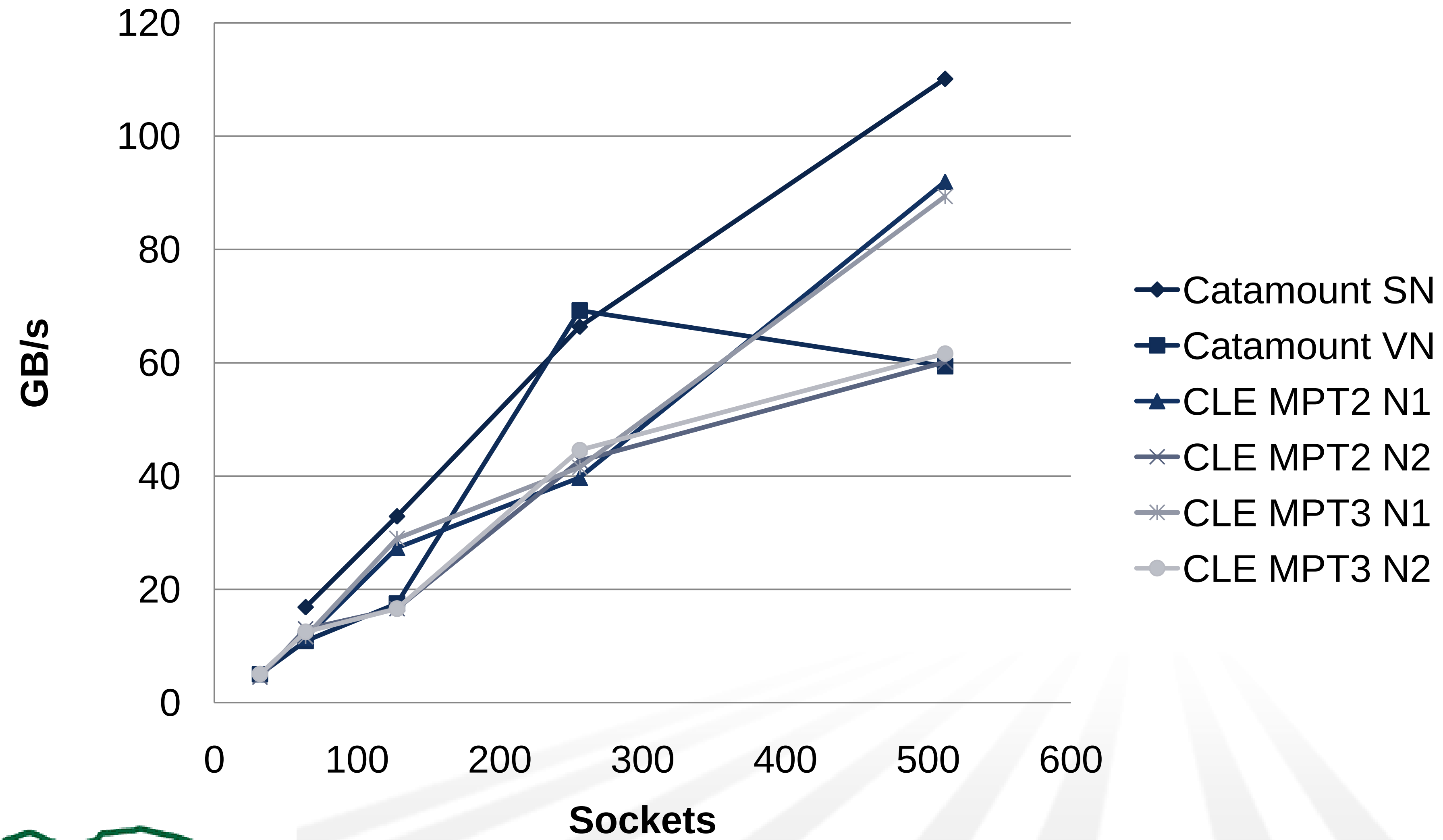
HPCC

Parallel Transpose (Cores)



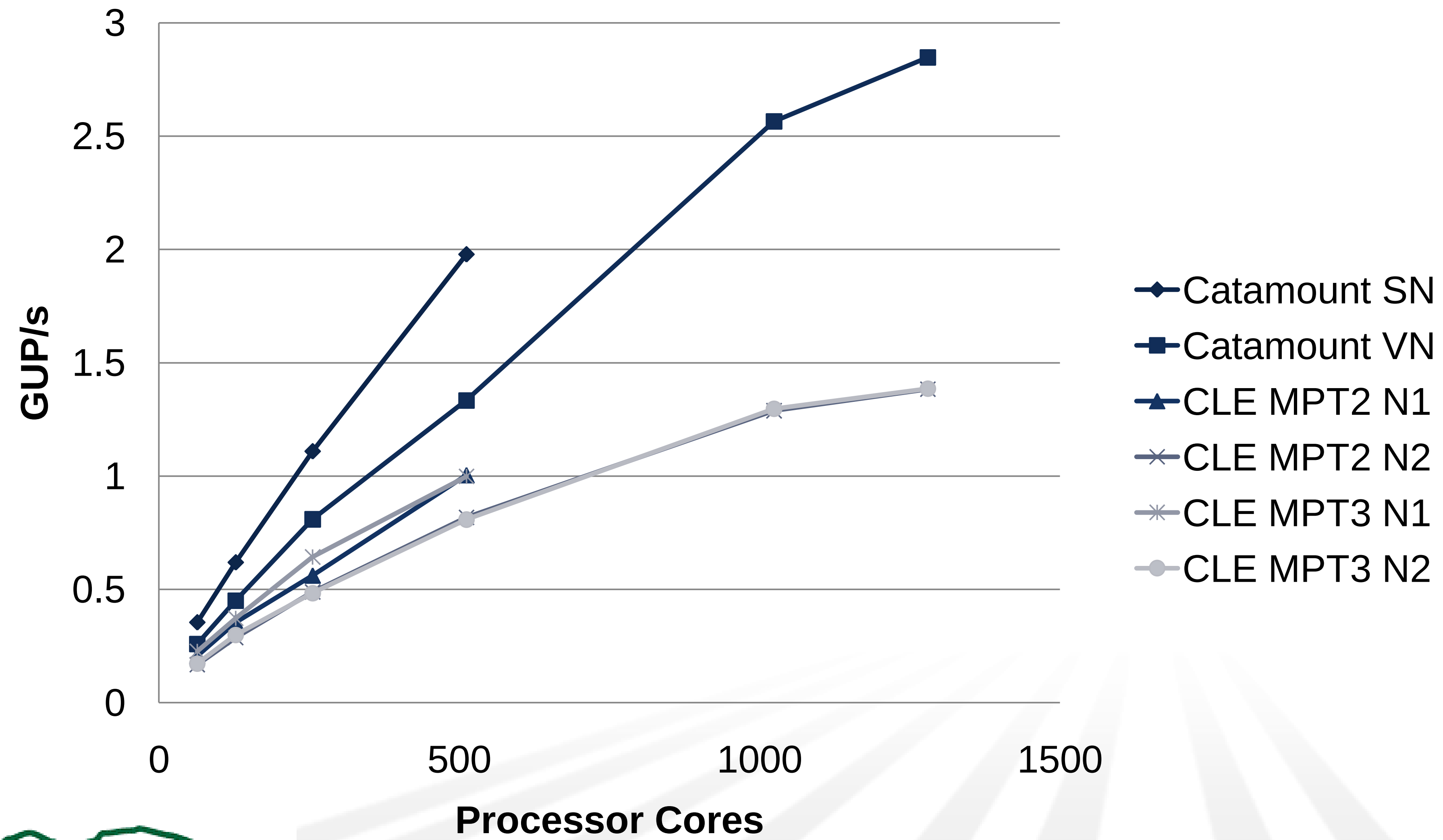
UT-BATTELLE
OAK RIDGE NATIONAL LABORATORY

Parallel Transpose (Sockets)

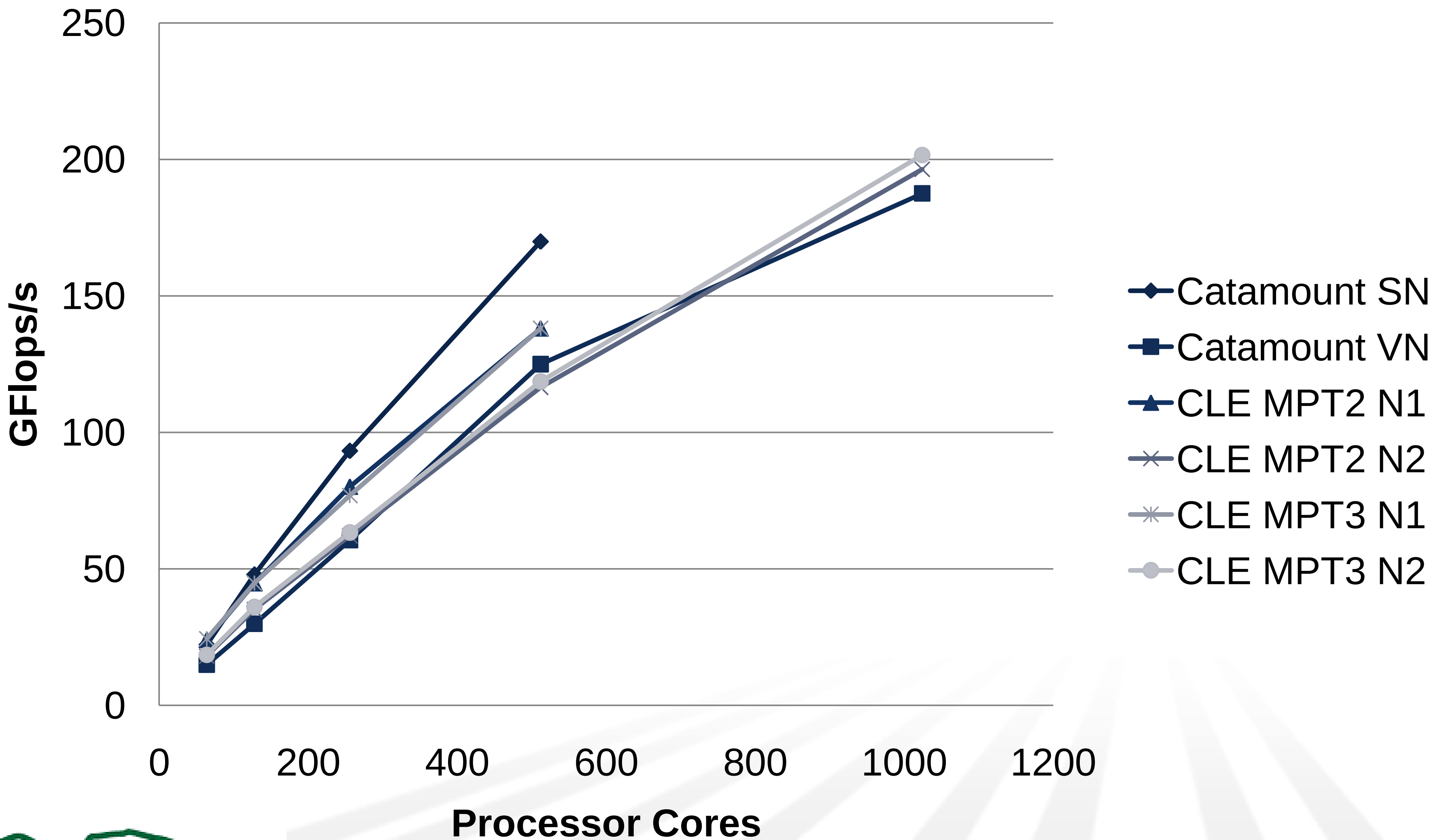


UT-BATTELLE
Oak Ridge National Laboratory

MPI Random Access

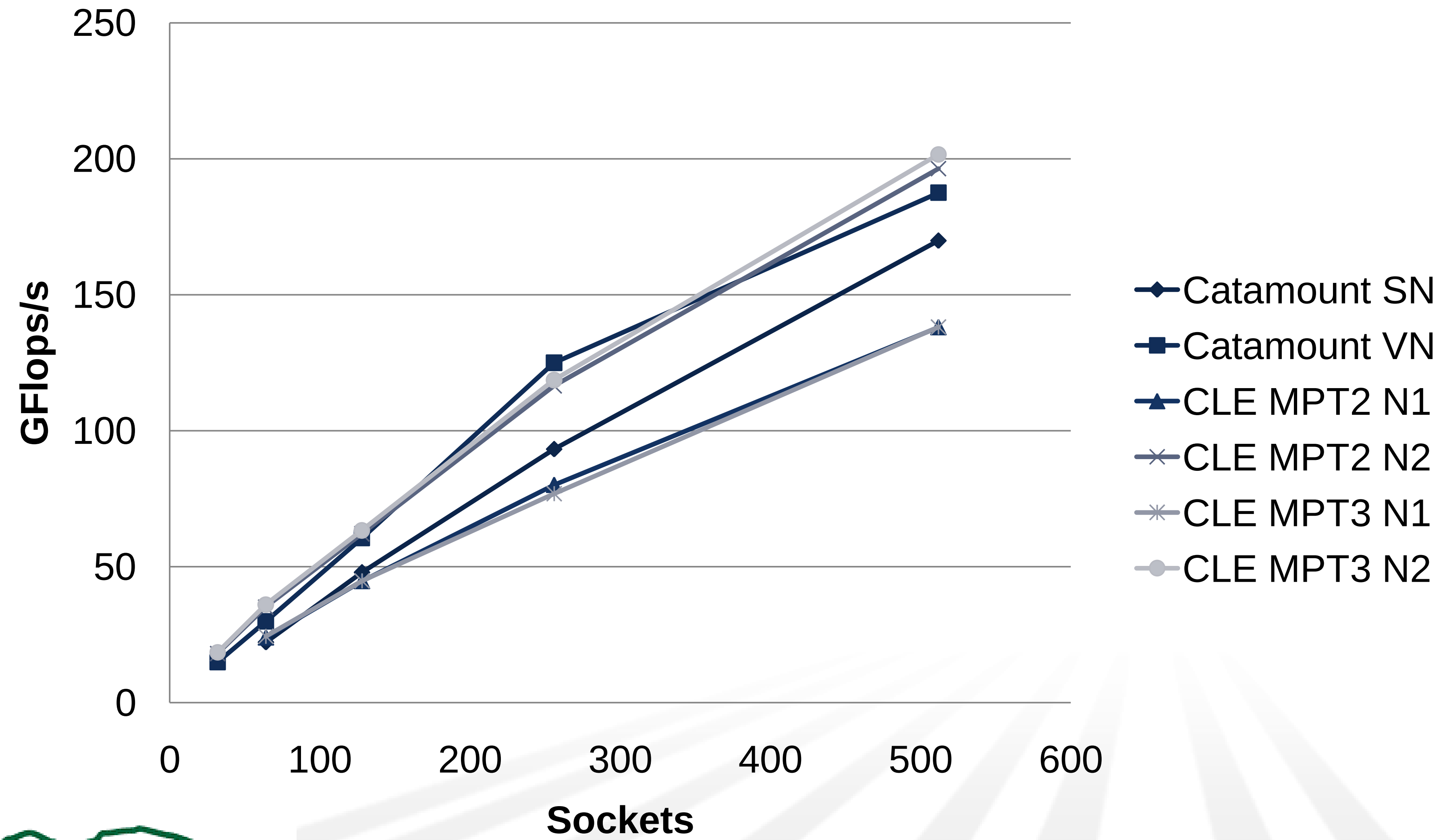


MPI-FFT (cores)



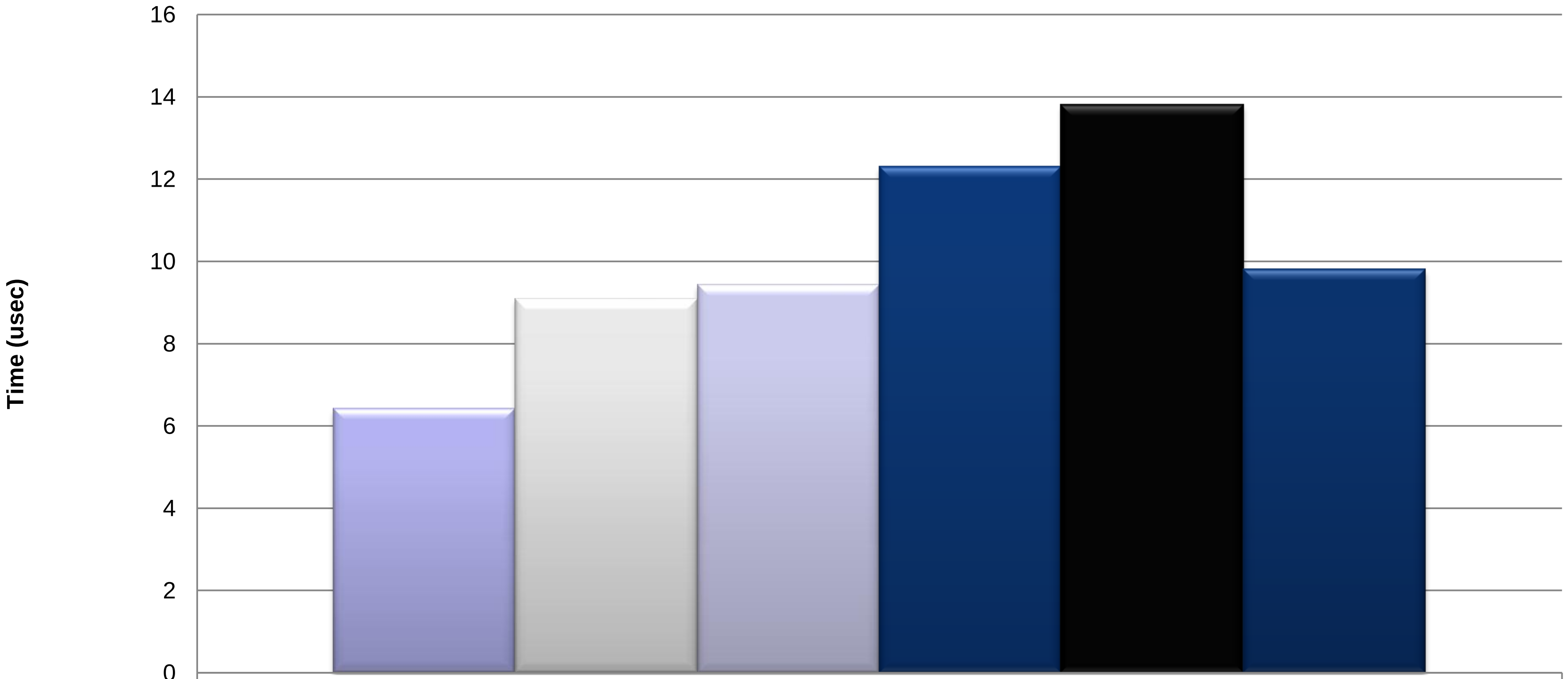
UT-BATTELLE
OAK RIDGE NATIONAL LABORATORY

MPI-FFT (Sockets)



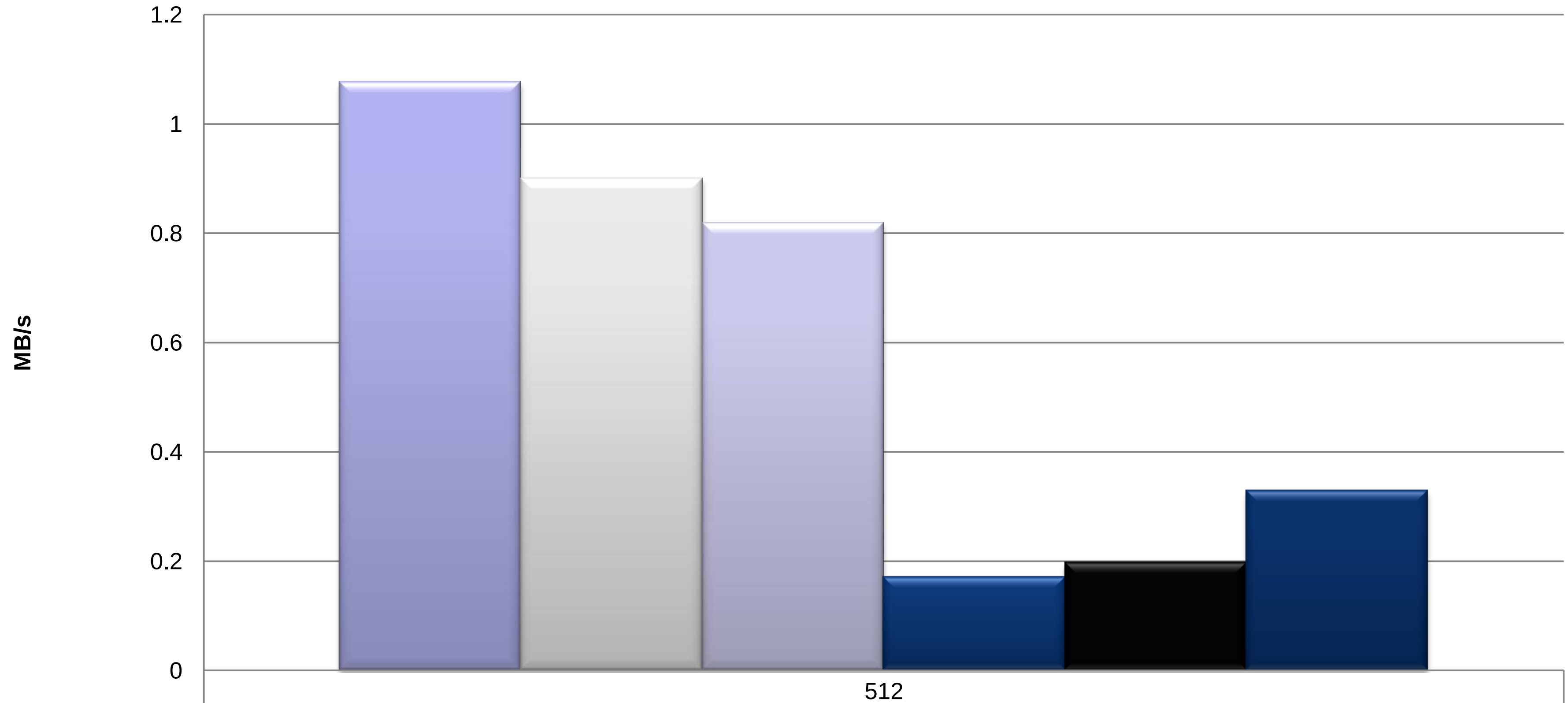
UT-BATTELLE
Oak Ridge National Laboratory

Naturally Ordered Latency



	512
Catamount SN	6.41346
CLE MPT2 N1	9.08375
CLE MPT3 N1	9.41753
Catamount VN	12.3024
CLE MPT2 N2	13.8044
CLE MPT3 N2	9.799

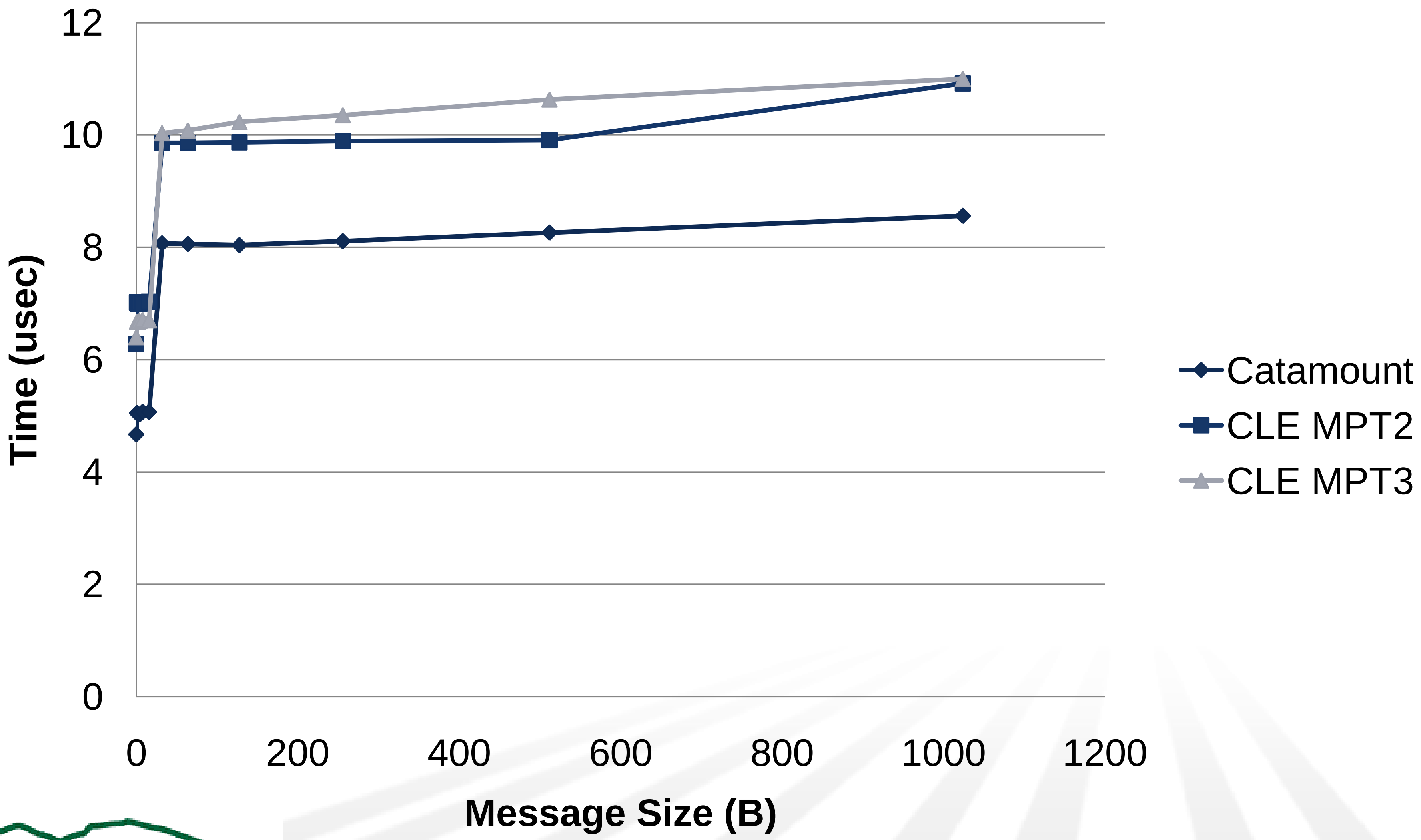
Naturally Ordered Bandwidth



	512
Catamount SN	1.07688
CLE MPT2 N1	0.900693
CLE MPT3 N1	0.81866
Catamount VN	0.171141
CLE MPT2 N2	0.197301
CLE MPT3 N2	0.329071

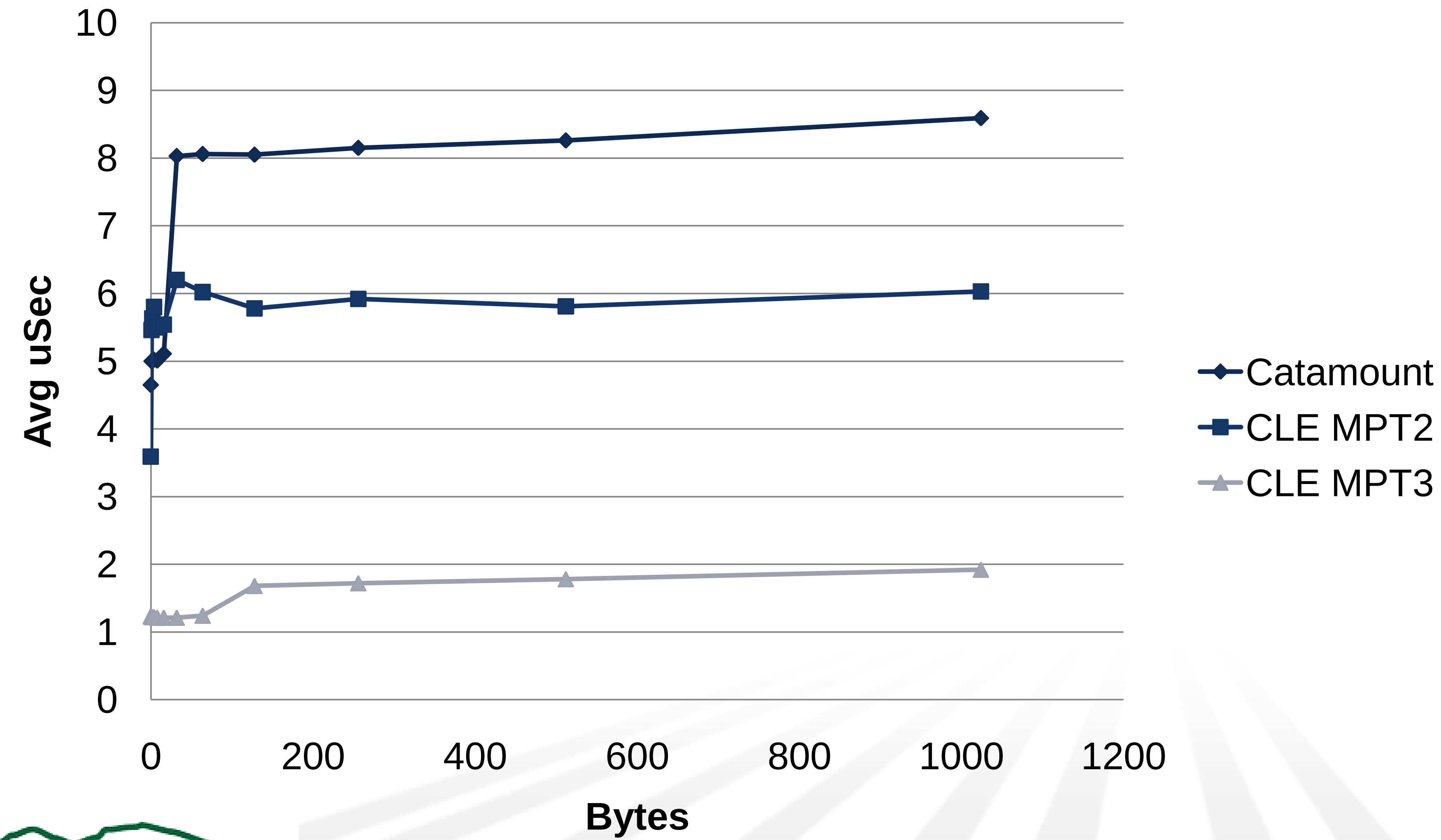
IMB

IMB Ping Pong Latency (N1)

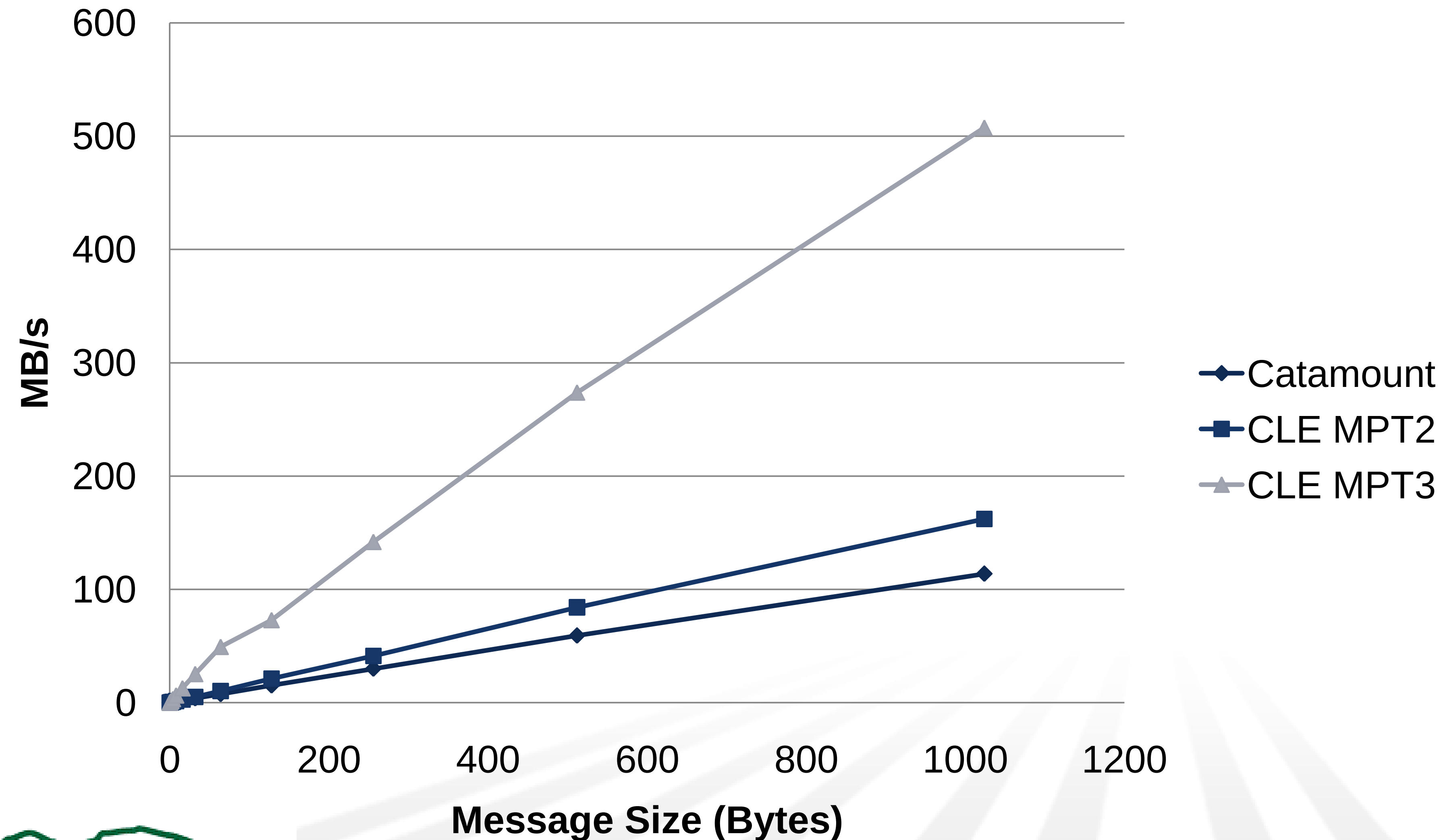


UT-BATTELLE
OAK RIDGE NATIONAL LABORATORY

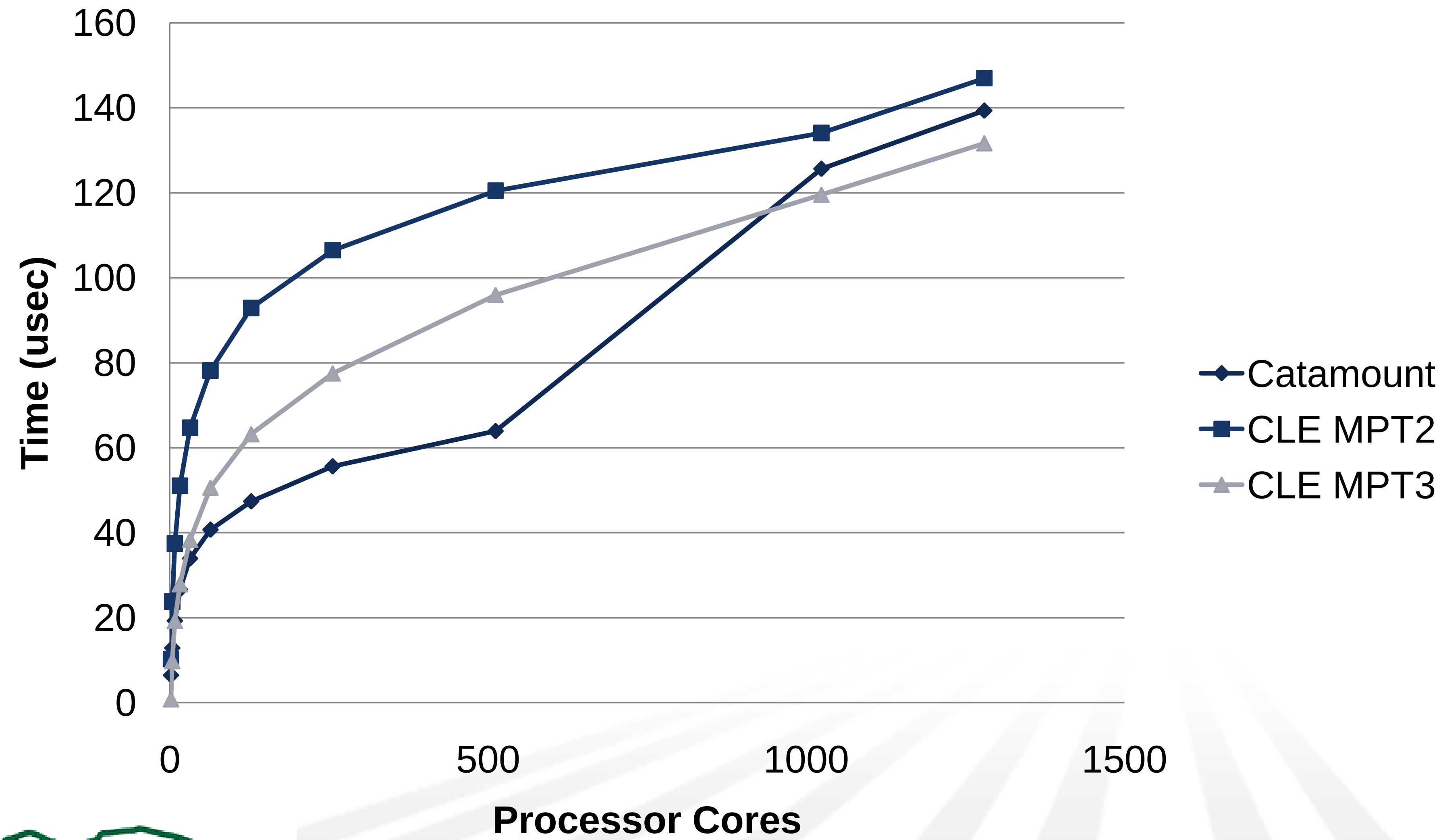
IMB Ping Pong Latency (N2)



IMB Ping Pong Bandwidth

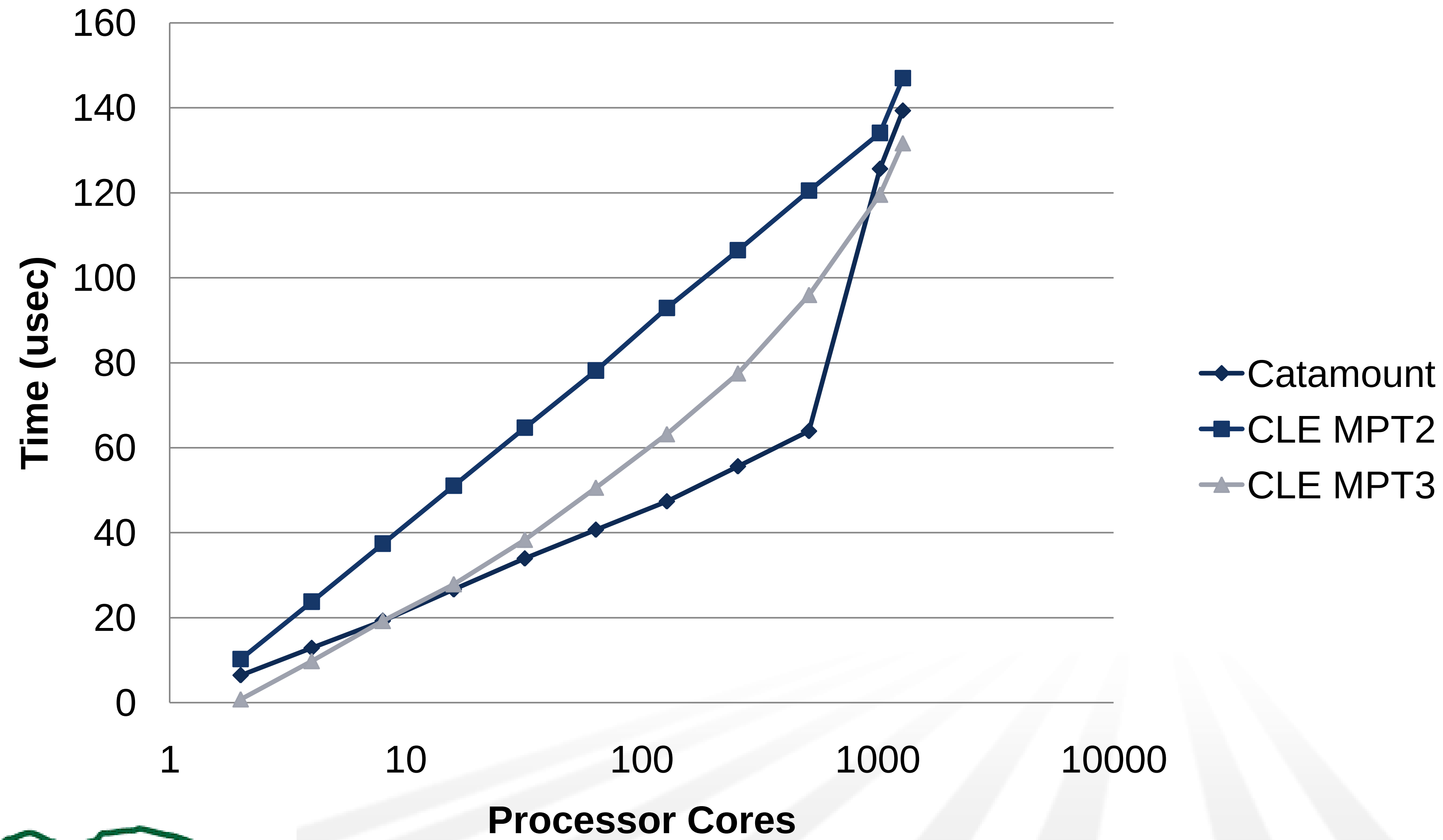


MPI Barrier (Lin/Lin)

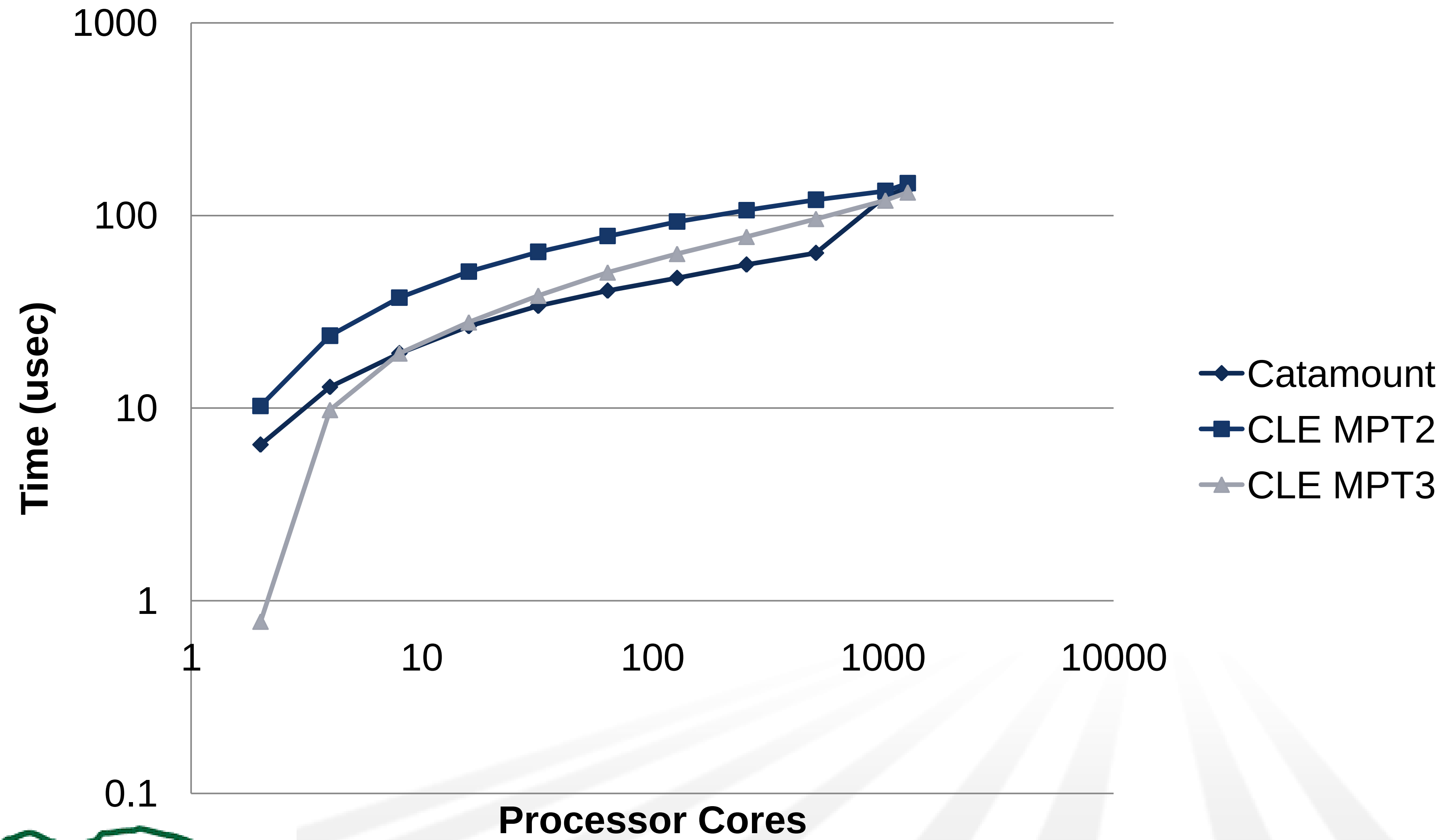


UT-BATTELLE
OAK RIDGE NATIONAL LABORATORY

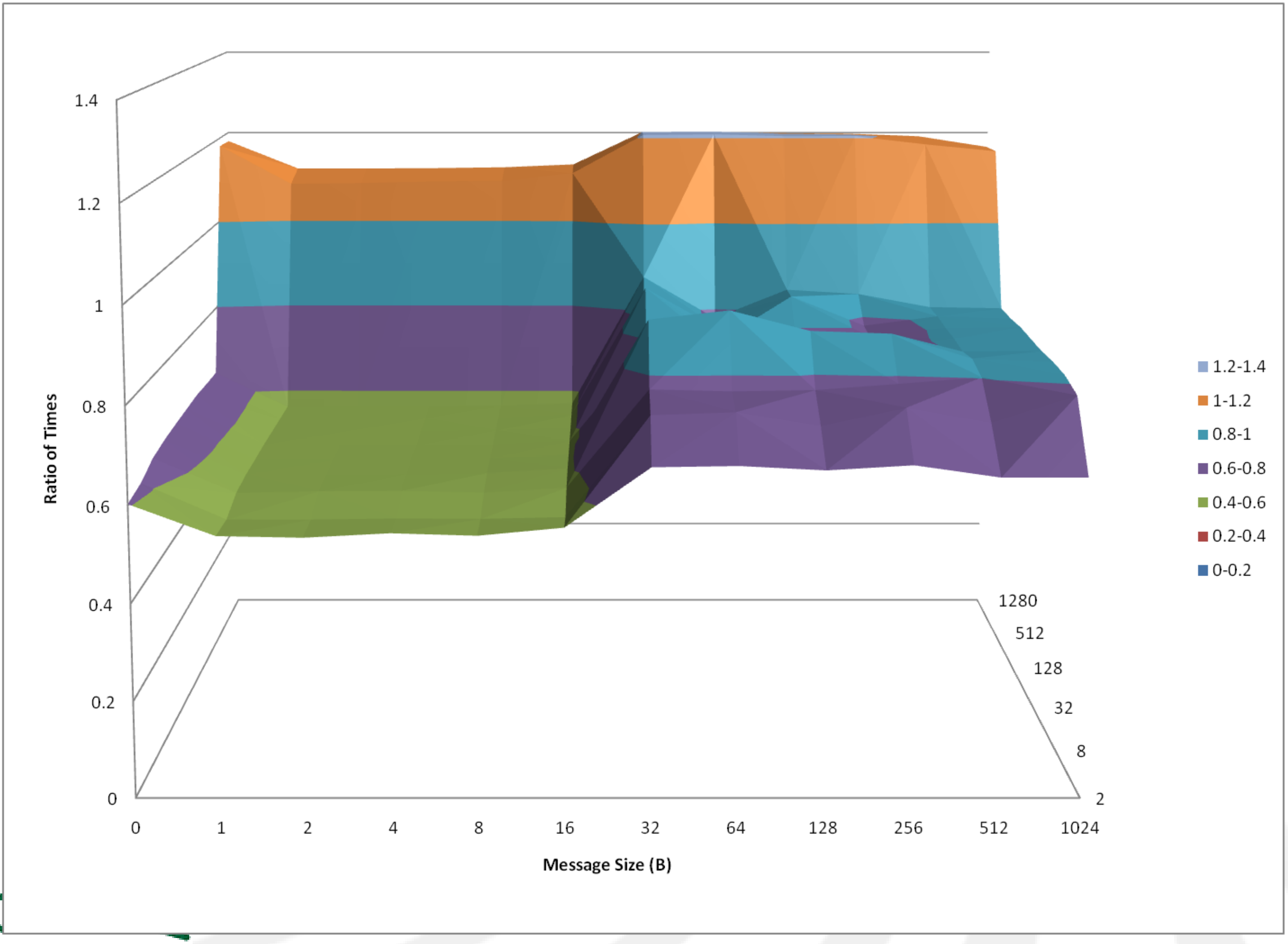
MPI Barrier (Lin/Log)



MPI Barrier (Log/Log)

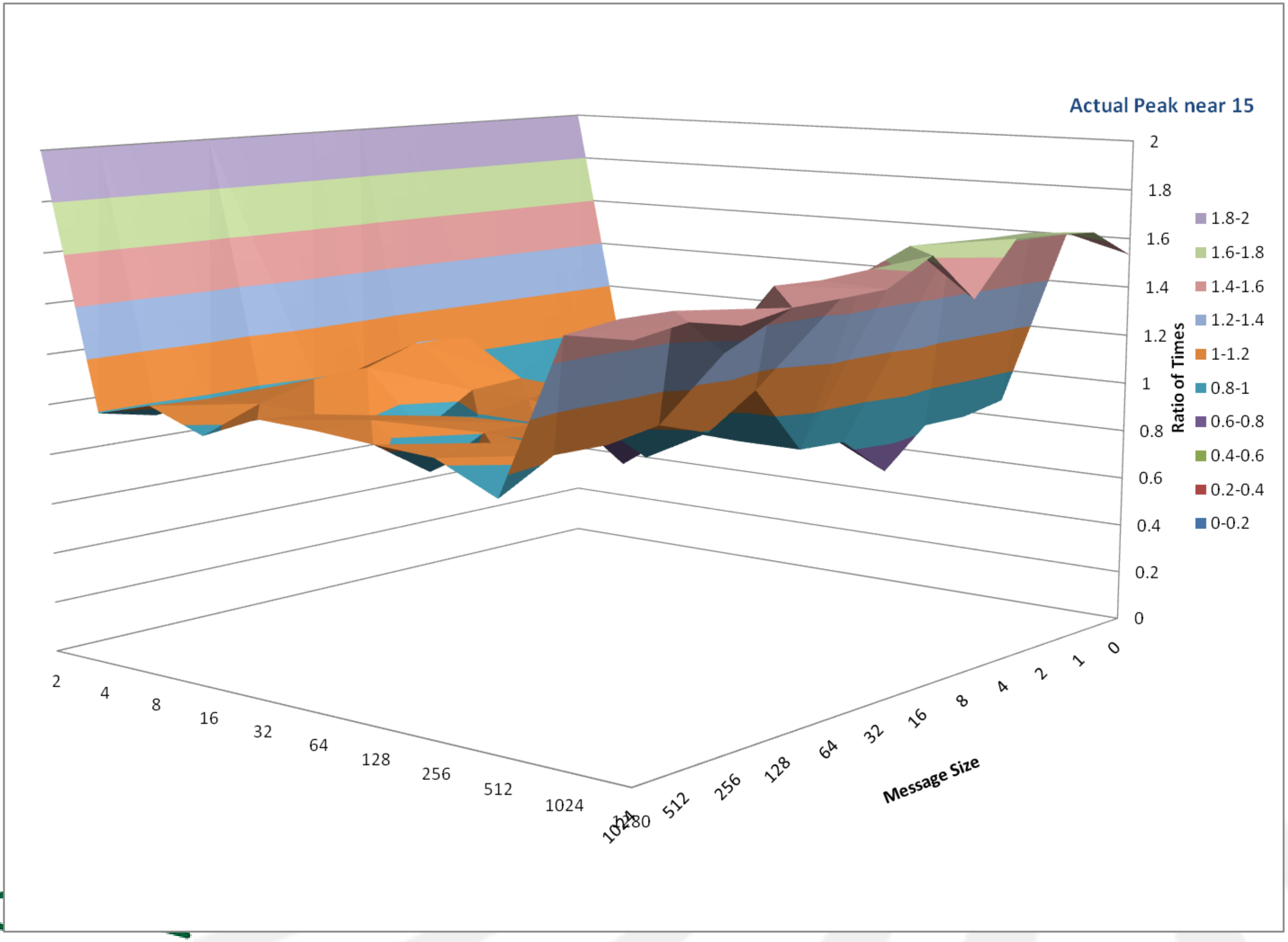


SendRecv (Catamount/CLE MPT2)

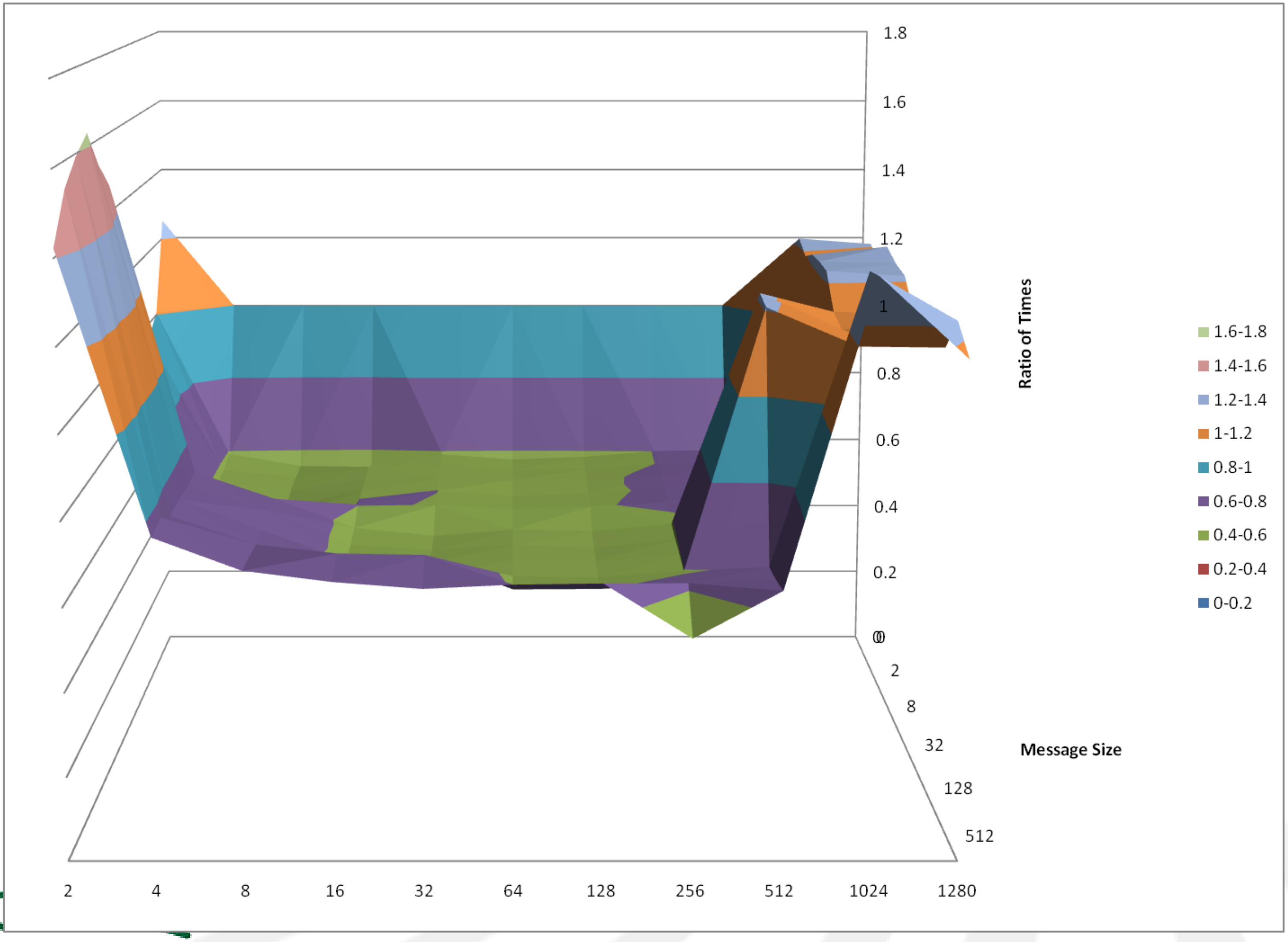


UT-BATTELLE
OAK RIDGE NATIONAL LABORATORY

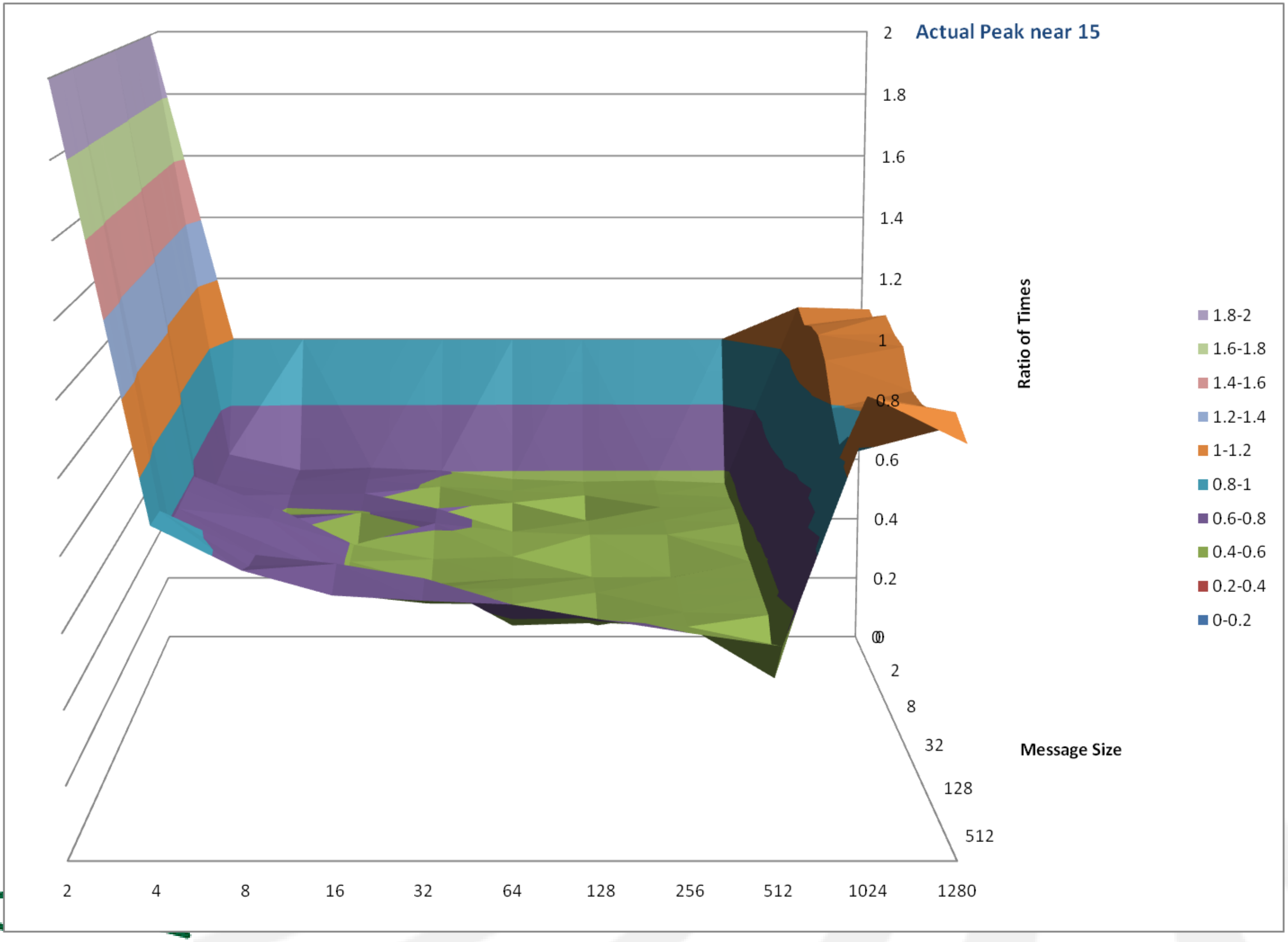
SendRecv (Catamount/CLE MPT3)



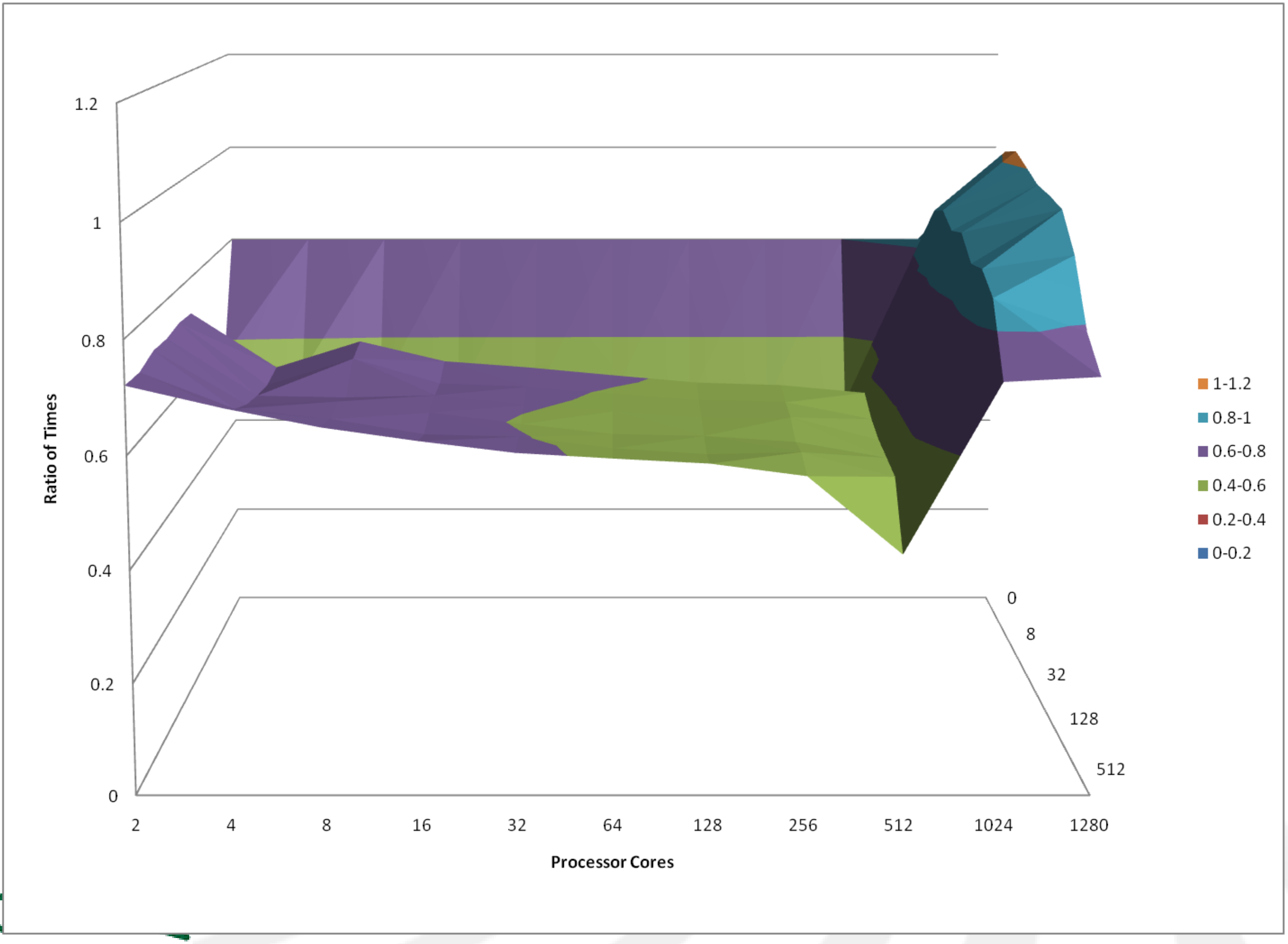
Broadcast (Catamount/CLE MPT2)



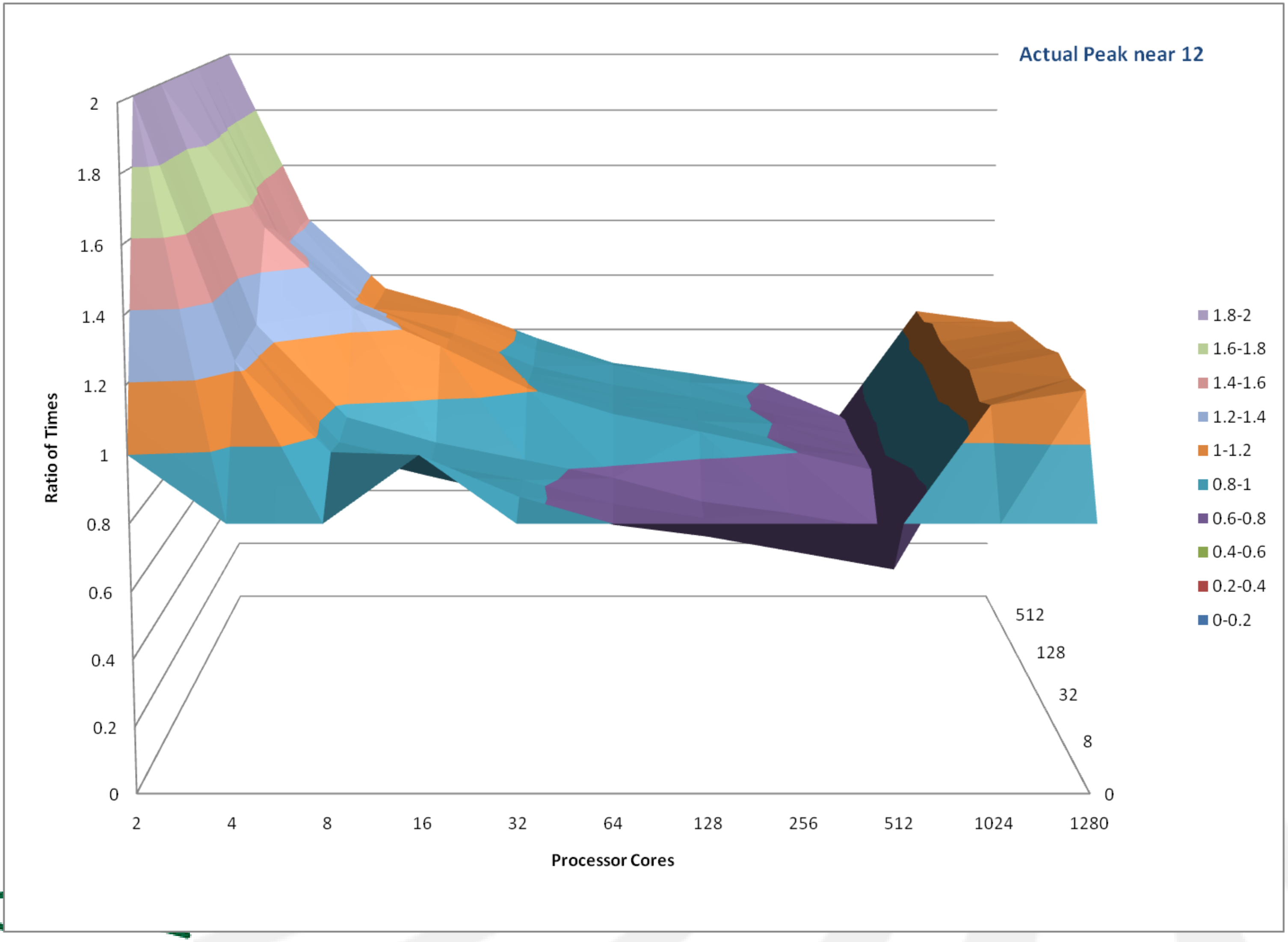
Broadcast (Catamount/CLE MPT3)



Allreduce (Catamount/CLE MPT2)

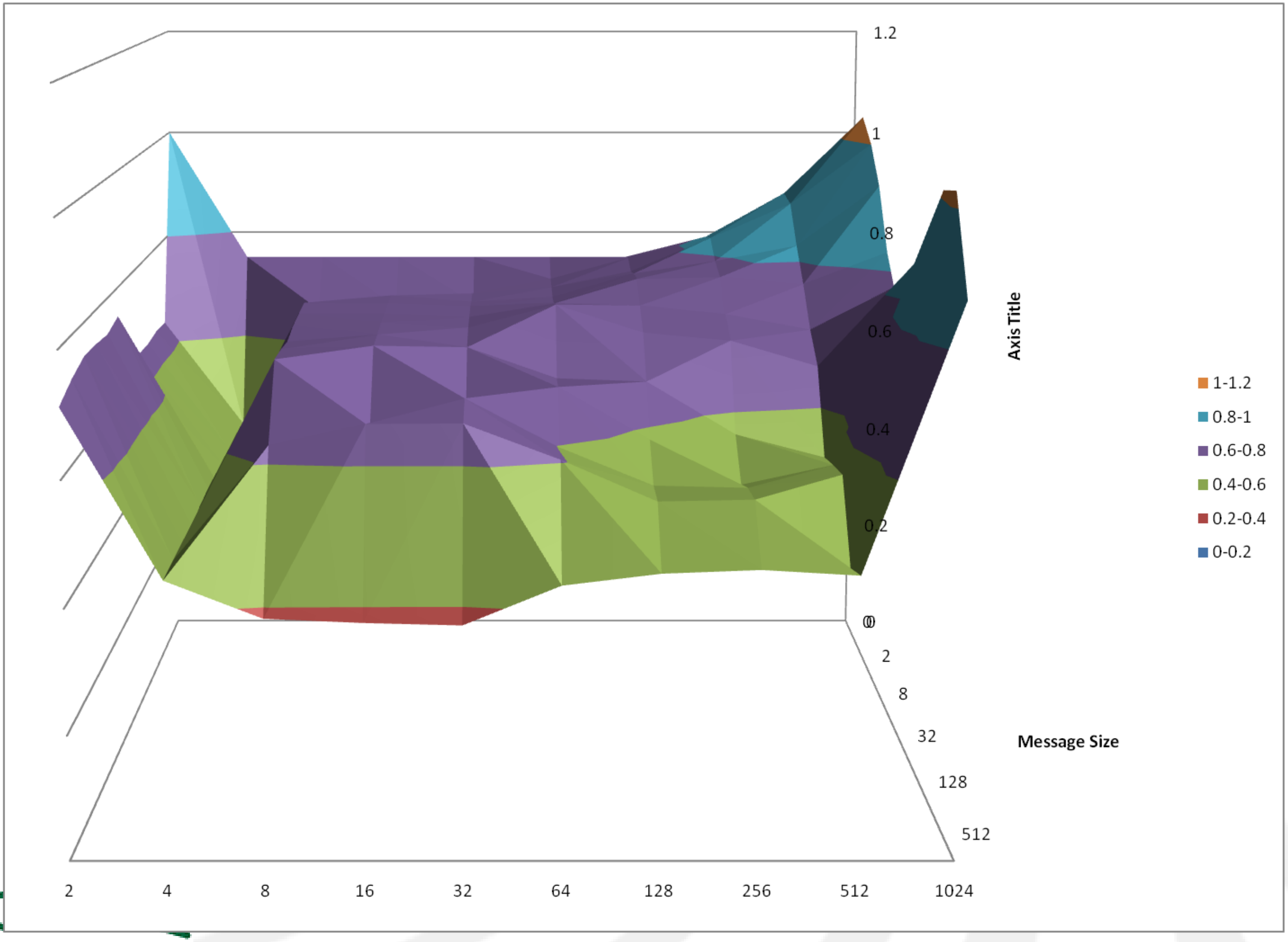


Allreduce (Catamount/CLE MPT3)

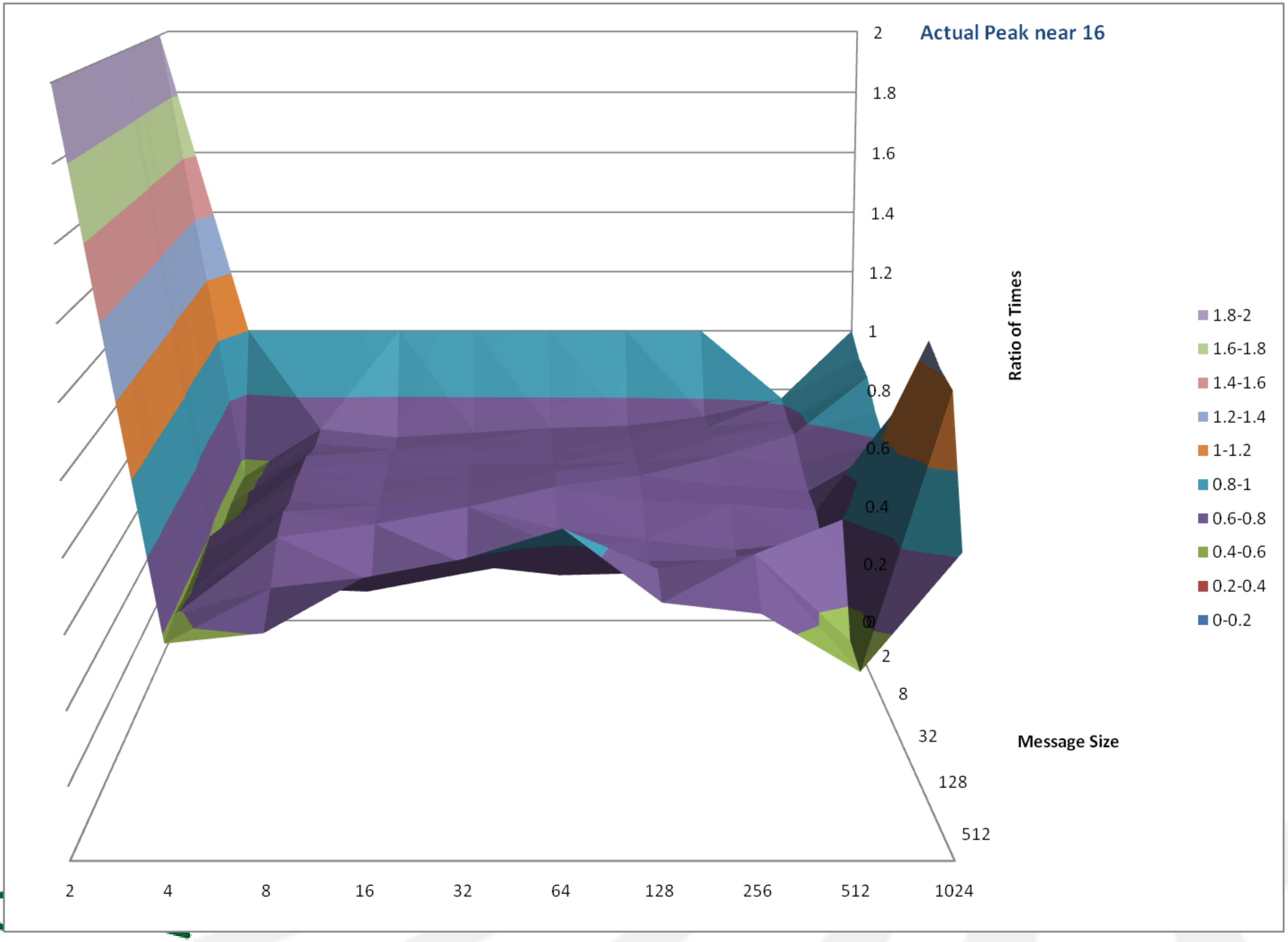


UT-BATTELLE
OAK RIDGE NATIONAL LABORATORY

AlltoAll (Catamount/CLE MPT2)



AlltoAll (Catamount/CLE MPT3)



CONCLUSIONS

What we saw

Catamount

- Handles Single Core (SN/N1) Runs slightly better
- Seems to handle small messages and small core counts slightly better

CLE

- Does very well on dual-core
- Likes large messages and large core counts
- MPT3 helps performance and closes the gap between QK and CLE

What's left to do?

- We'd really like to try this again on a larger machine
  - Does CLE continue to beat Catamount above 1024, or will the lines converge or cross?
- What about I/O?
  - Linux adds I/O buffering, how does this affect I/O performance at scale?
- How does this translate into application performance?
  - See "Cray XT4 Quadcore: A First Look", Richard Barrett, et.al., Oak Ridge National Laboratory (ORNL)

Does CLE waddle like a penguin, or run like a catamount?

CLE RUNS LIKE A BIG CAT!

Acknowledgements

- This research used resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.
- Thanks to Steve, Norm, Howard, and others for help investigating and understanding these results

UT-BATTELLE
OAK RIDGE NATIONAL LABORATORY