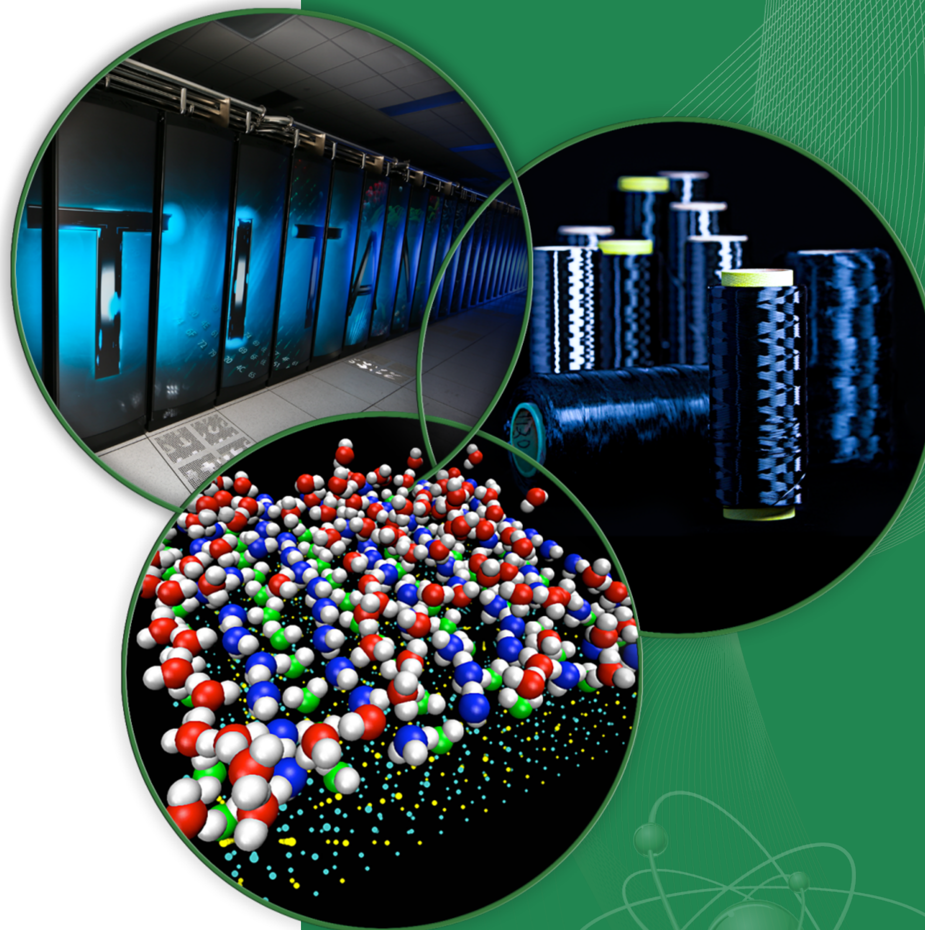
Automated Characterization of Parallel Application Communication Patterns

Philip C. Roth
Jeremy S. Meredith
Jeffrey S. Vetter

Oak Ridge National Laboratory

17 June 2015



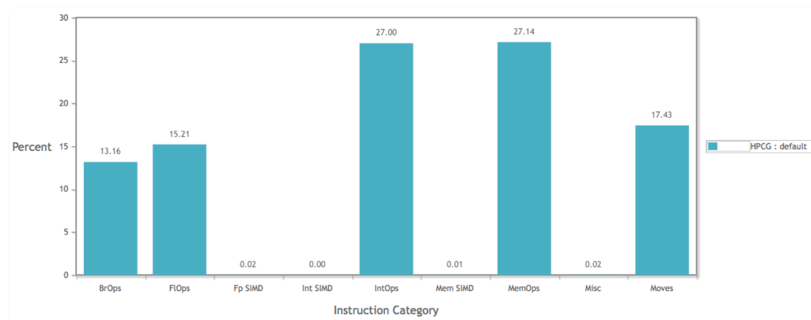
ORNL is managed by UT-Battelle
for the US Department of Energy

OAK RIDGE
National Laboratory

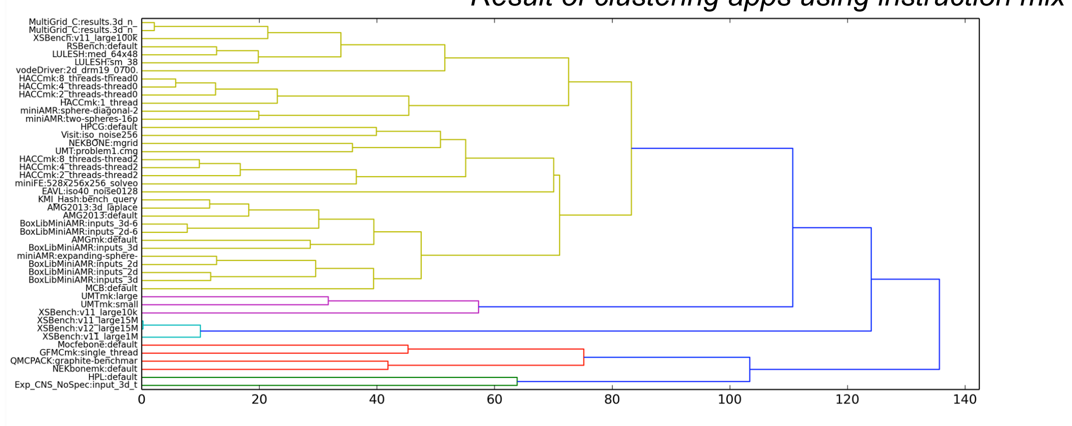
Background: Oxbow

- Characterize application demands independent of performance
  - System design
  - Representativeness of proxy apps
- Characterization on several axes:
  - Communication (topology, volume)
  - Computation (instruction mix)
  - Memory access (reuse distance)
- Online database for results with web portal including analytics support
- http://oxbow.ornl.gov

Instruction Mix, HPCG, 64 processes



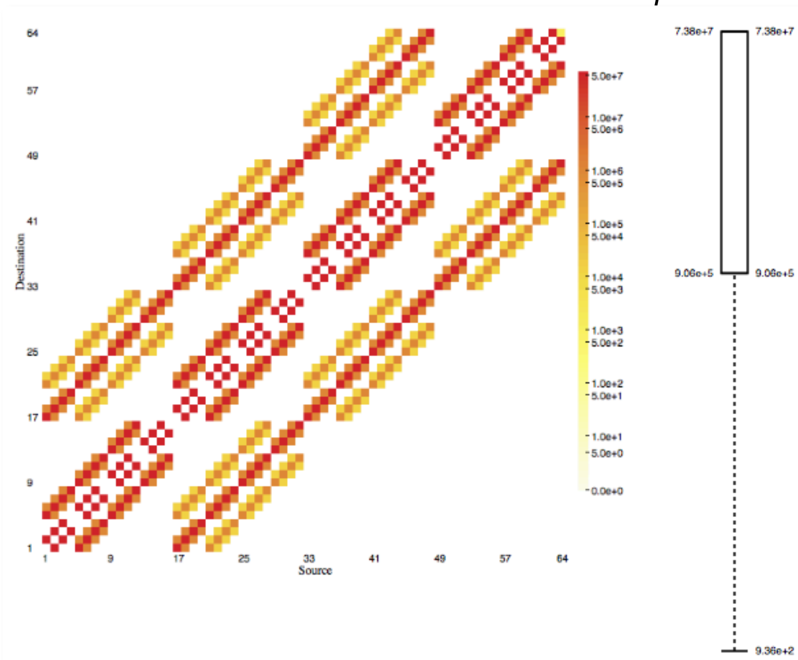
Result of clustering apps using instruction mix



Background: mpiP

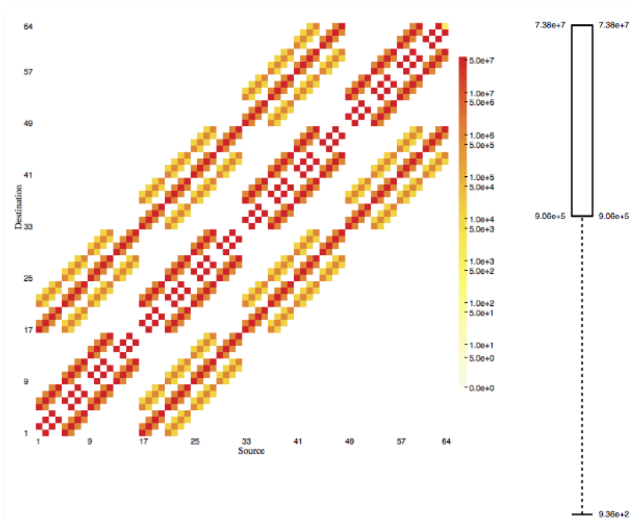
- Lightweight communication and I/O profiler for MPI programs
- Interposes instrumentation using PMPI interface
- For Oxbow, we modified mpiP to track:
  – Sender, receiver, volume for point to point operations
  – Root, destination(s) and volume for rooted collectives
  – Processes involved, volume for rootless collectives

Communication matrix for HPCG, 64 processes



The Problem

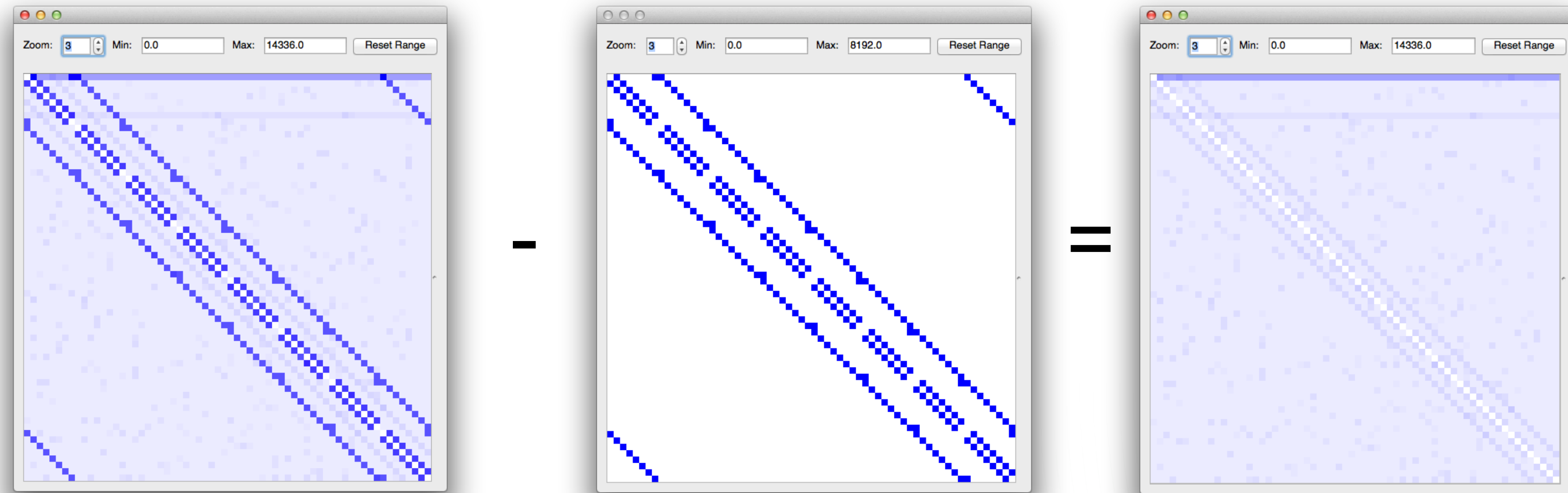
- We want concise way to express application communication demands
- E.g., “3D Nearest Neighbor plus broadcast and reduce” instead of:



- But…expertise needed to identify patterns from communication matrices

Inspiration: Sky Subtraction

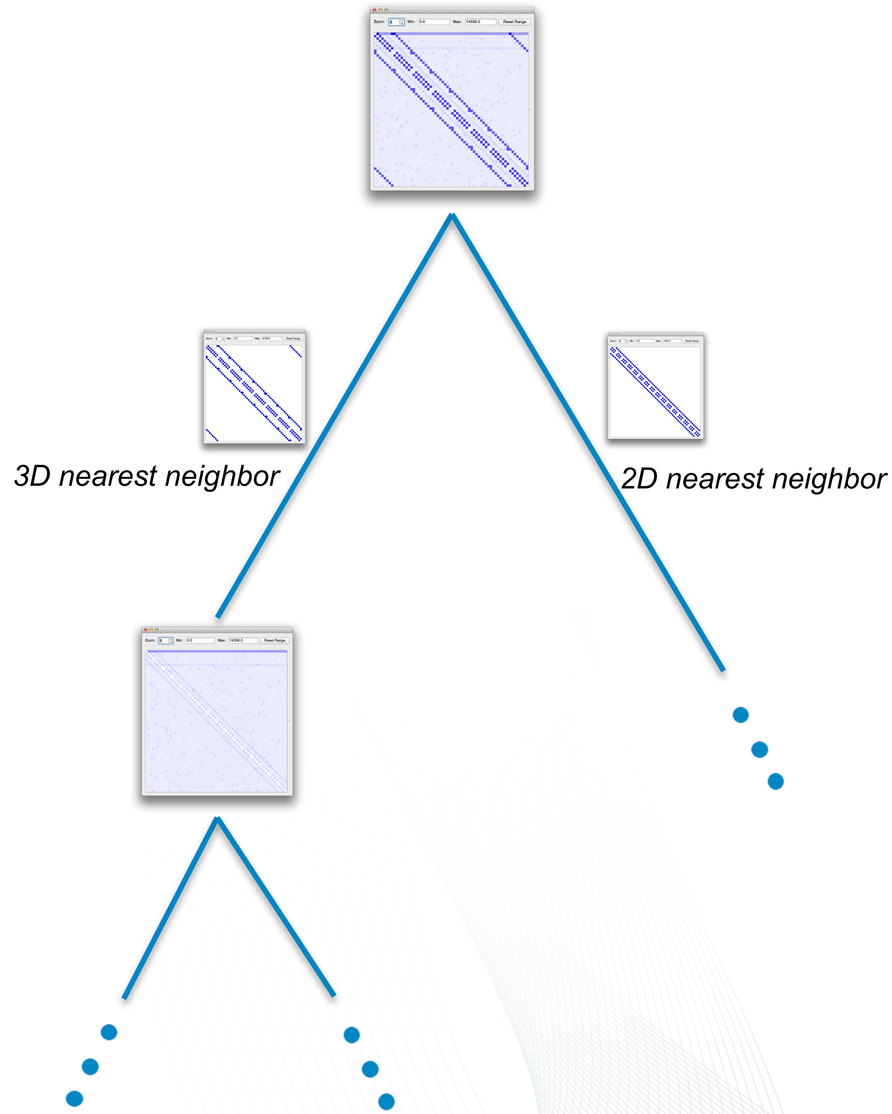
- Inspired by sky subtraction: *given an image, remove the known to make it easier to identify the unknown*



Recognizing and removing the contribution of a 2D nearest neighbor pattern in a synthetic communication matrix. This represents ***one step*** *in a search-based approach.*

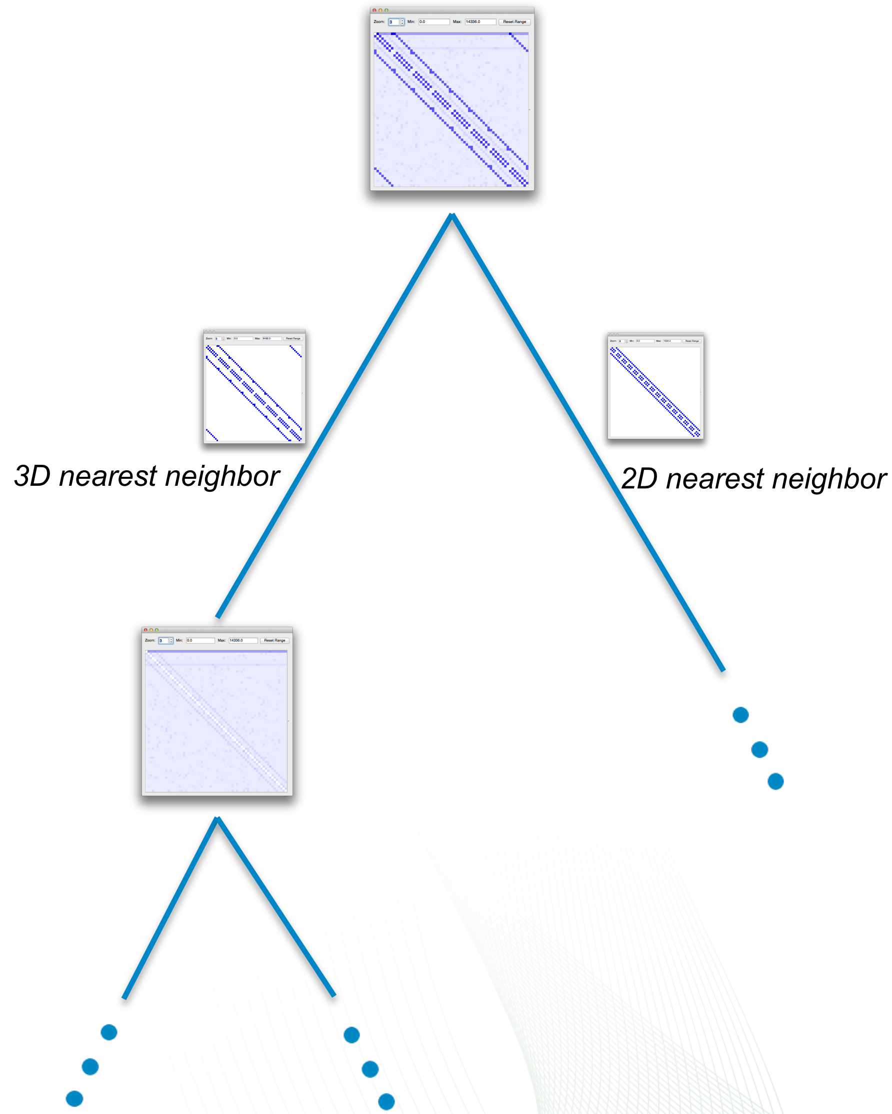
Our Approach: Search Results Tree

- Automated search using a library of patterns
- Search results tree
  - Each node represents a communication matrix with an associated *residual* that captures how much volume is represented by the matrix
  - Each edge represents a recognized pattern in parent node's matrix; subtracting that pattern results in child node's matrix



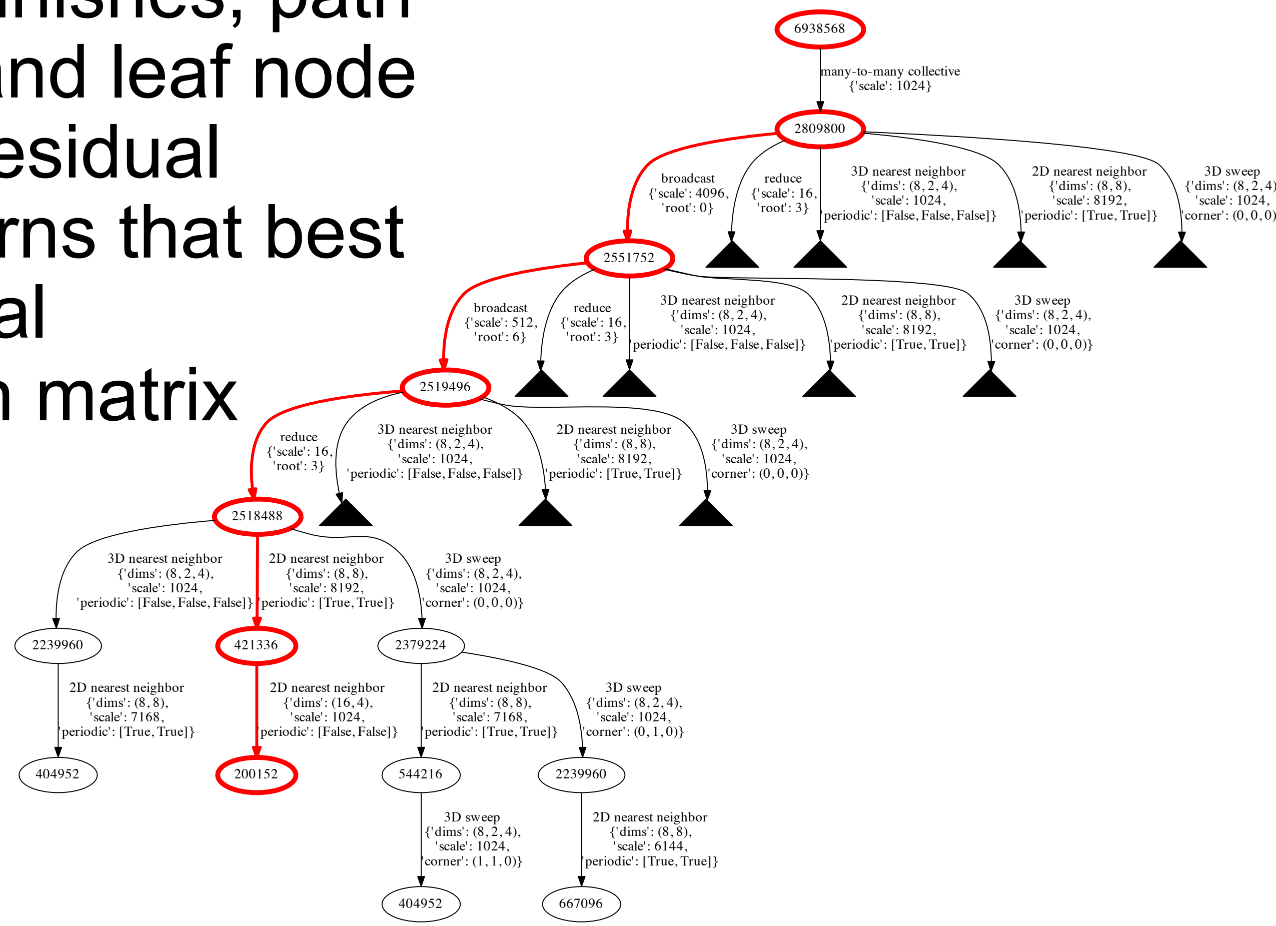
Our Approach: Algorithm Overview

- Associate original communication matrix with root node
- For each pattern in pattern library
  - Attempt to recognize the pattern in node's matrix
  - If recognized, add child node and edge to search result tree
- For each child node added, recursively apply search starting at child node



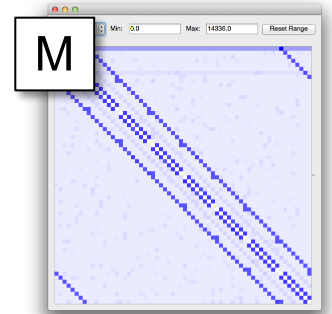
Our Approach: Final Result

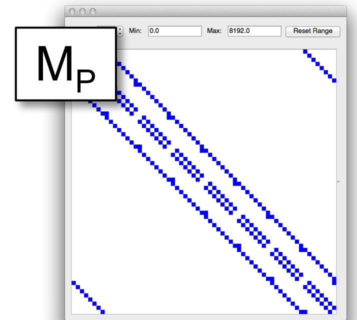
- When search finishes, path between root and leaf node with smallest residual indicates patterns that best explains original communication matrix



Pattern Recognition

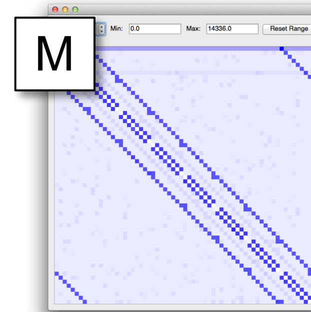
- Library of scale-independent pattern generators and recognizers
- Given matrix M and recognizer for pattern P
  - Determines number of processes represented in M
  - Checks entries in M that should be non-zero, keeping track of smallest non-zero value seen (for computing scale)
    - Detects 2D and 3D pattern dimensions based on non-zero diagonals
    - Detects root process for broadcasts and reduce
    - Detects origin corner for 3D wavefront
  - If all entries that should be non-zero are non-zero, reports that pattern is recognized and provides the scale

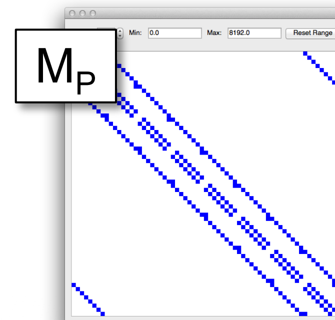


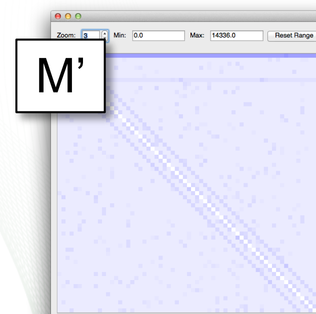


Pattern Removal

- When pattern P is recognized in M at scale S
  - Generator for P generates matrix M_P with scale 1
  - Search forms $M - SM_P = M'$, computes residual of M', and adds child node for M' with edge labeled with P and S



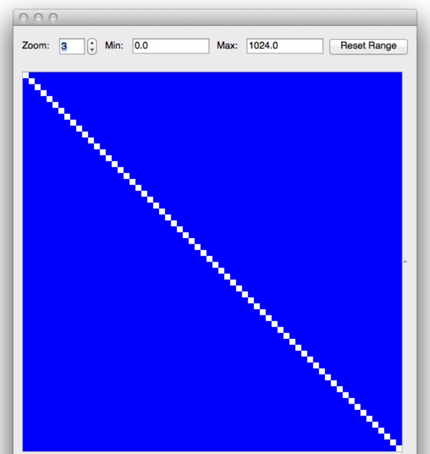


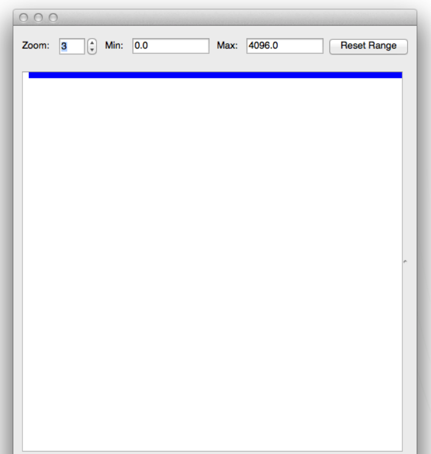


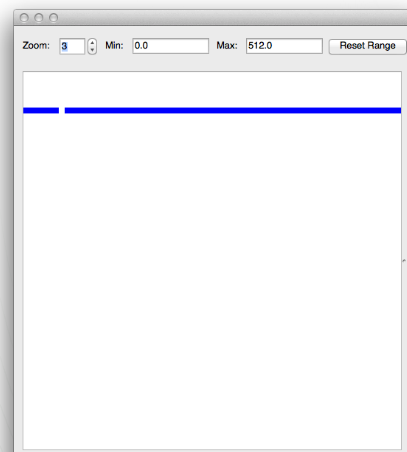
Recognition Order

- Recognition order matters
  - E.g., search recognizes and removes pattern for a broadcast, but doing so precludes subsequent recognition of all-to-all
- At each node in search results tree search attempts to recognize all patterns so search considers all permutations of pattern orderings
- As search speed optimization, always considers rootless collectives first
  - Otherwise, will recognize a long sequence of rooted collectives
  - Rooted collective sequence is equivalent from a topology/message volume perspective

Rootless collective pattern







Broadcast patterns from root 0 (left) and root 6 (right)

Residual

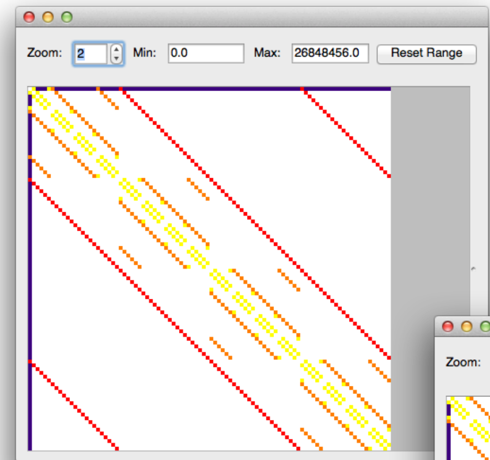
- Current residual definition is sum of values in matrix, representing amount of traffic to be explained
  - Lower is better
  - Simple to understand
  - Simple to compute
  - But, absolute values can be very large, even for modest-sized programs
- Alternatives possible
  - Keep definition and express as percentage of original message volume
  - Statistics (e.g., average volume across all processes)

Implementation

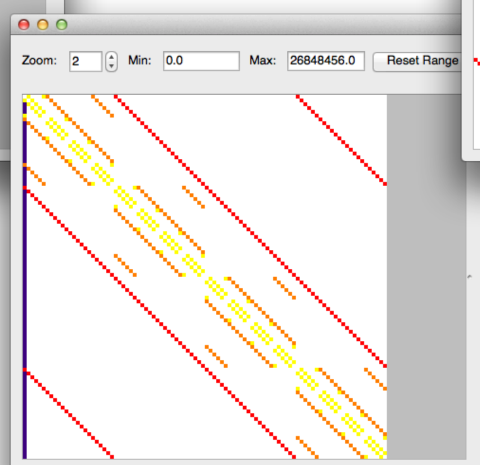
- AChax, implemented in Python using NumPy and SciPy matrix support
- Each pattern is a Python class with Recognize and Generate methods
  - Many-to-many
  - Broadcast
  - Reduce
  - 2D Nearest Neighbor
  - 3D Nearest Neighbor
  - 3D Wavefront (sweep) from a corner
  - Random (generate only)
- AChax search engine reports collection of patterns and their scale that best explains original communication matrix, and optionally:
  - Matrix identified as having smallest residual
  - Log of search actions
  - Search results tree in format that can be visualized by GraphViz
  - Sequence of files containing intermediate matrices on path between tree root and leaf with lowest residual

Example: LAMMPS

- Communication matrix collected using mpiP from 96 process run on Keeneland Initial Delivery System, EAM benchmark problem
- Basically a 3D Nearest Neighbor pattern, but imperfect pattern (red circle in last figure)

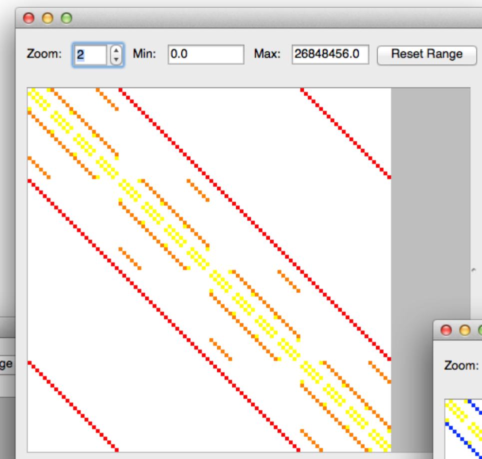


Original matrix

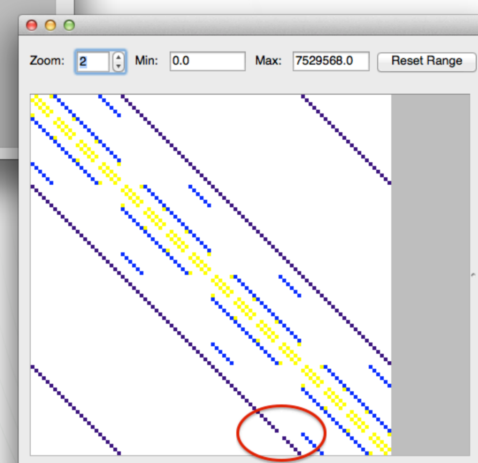


After removing broadcast

After removing reduce



After removing 3D nearest neighbor, dimensions (4,4,6), periodic



SUPER INSTITUTE FOR SUSTAINED PERFORMANCE, ENERGY, AND RESILIENCE

OAK RIDGE National Laboratory

LAMMPS: Expressing Search Results

- Search results trees are useful but not particularly concise
- We use an expression using parameterized pattern names with scale coefficients, e.g.,

$$
\begin{aligned}
C_{LAMMPS} = {} & 13354 \cdot Broadcast(root : 0) + \\
& 700 \cdot Reduce(root : 0) + \\
& 19318888 \cdot 3DNearestNeighbor(\\
& \quad dims : (4, 4, 6), \\
& \quad periodic : True)
\end{aligned}
$$

Future Directions

- Expand pattern library
  - Always more patterns to support
  - Irregular patterns
- Handle imperfect patterns (e.g., LAMMPS example) with nearness score
- Add better support for operation identification
  - We can recognize an all-to-all pattern, but cannot discern which rootless MPI operation was used using just the matrix, nor say whether it was truly an all-to-all or a naïve sequence of broadcasts
  - Incorporating tracing and/or profiling data may help
- Truly scale-independent expressions
  - Modeling integration (e.g., ASPEN)

Future Directions (II)

- Search optimizations
  - Parallelize the search
  - Prune the search by recognizing search path prefixes that are permutations
    - Recognition order matters, but having recognized A, B, C on one path results in same matrix as other path that recognized C, A, B as long as scales match – don't need to continue search from both
- Phase-aware characterization
  - mpiP can generate per-phase communication matrices
- Using image recognition algorithms for pattern matching

Acknowledgments

- This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). This research is sponsored by the Office of Advanced Scientific Computing Research in the U.S. Department of Energy.

Summary

- We are researching an approach for automatically characterizing communication patterns of message passing applications
- We look for combination of simple patterns that best explains observed communication behavior using:
  - Automated search through large search space
  - Pattern generator library
  - "Subtraction" of recognized patterns from observed communication
- rothpc@ornl.gov
- http://ft.ornl.gov/~rothpc