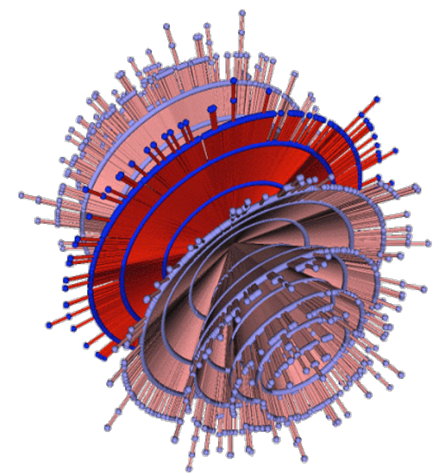
Scalable & Efficient Parallelization of Graph Methods for Boolean Satisfiability and Power Grid Analysis on the Cray XMT

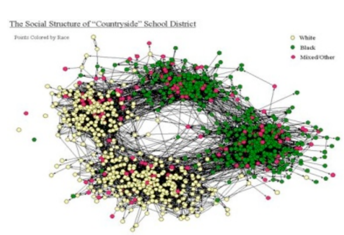
Daniel Chavarría-Miranda, Henry Huang, Heidi Sofia
Pacific Northwest National Laboratory (PNNL)

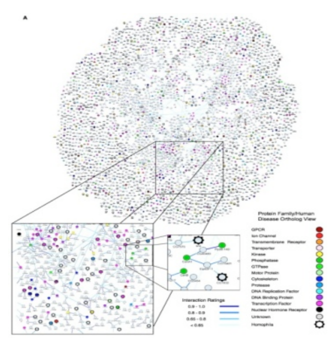
Parallel Processing for Graphs

- Requirements
  - sophisticated data representations
  - dynamic structural updates
  - fine-grain communication and synchronization events
  - algorithms that exploit topological characteristics
- Challenges
  - large data sizes
  - heterogeneous data
  - dynamic, irregular work loads
  - dynamic/temporal nature of data

“Six degrees of separation”





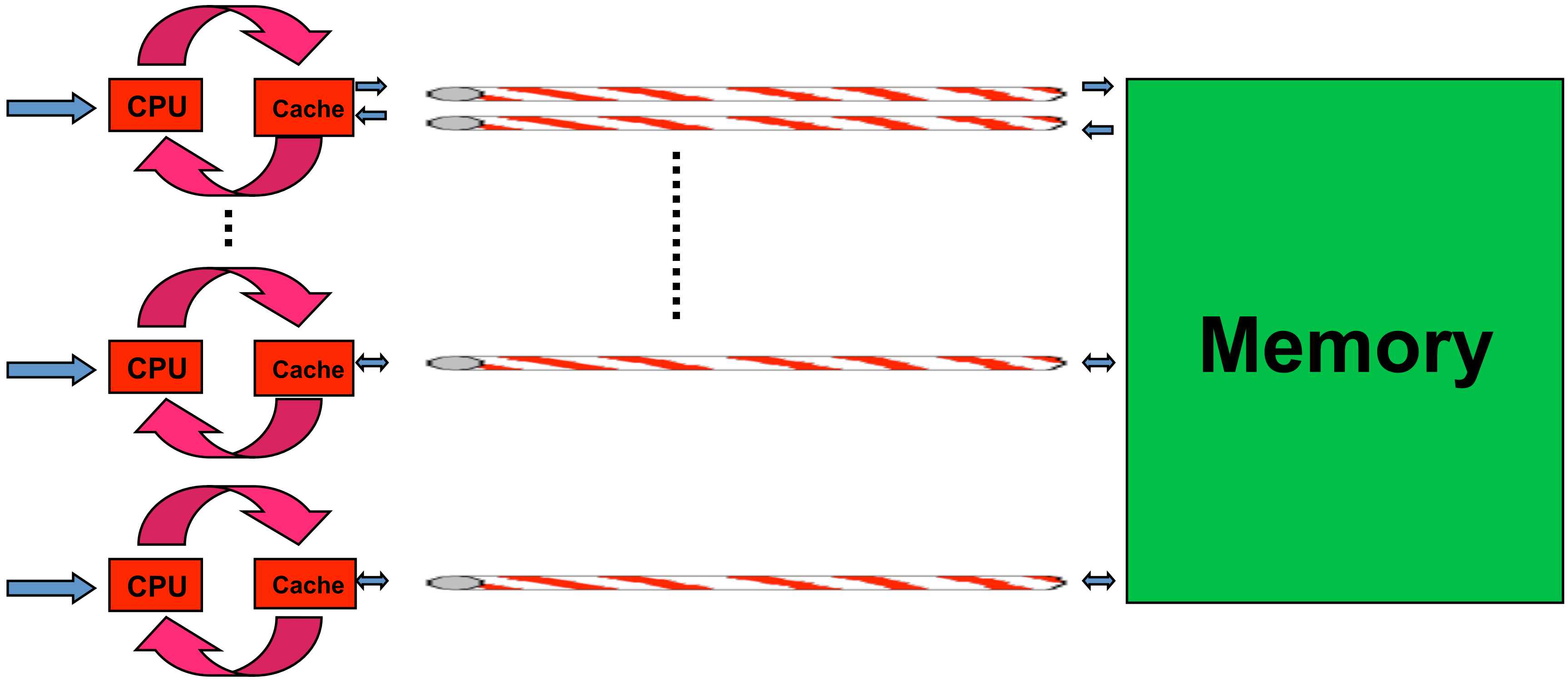


dynamic, high-dimensional data,
low graph diameters, power law distribution

Outline

- Parallel Processing for Graphs
- ☞ Cray XMT
- Boolean Satisfiability
  - Survey Propagation
    - Graph Formulation
- Power Grid Contingency Analysis
  - Graph-based problem formulation
- Conclusions

Today's parallel computer



AVOID THE STRAWS OR STARVE

Commodity cluster programming

- Place data near computation
- Access data in order and reuse
- Partition program into independent, balanced computations
- Minimize synchronization and communication operations
- Avoid modifying shared data
- Avoid adaptive and dynamic computations

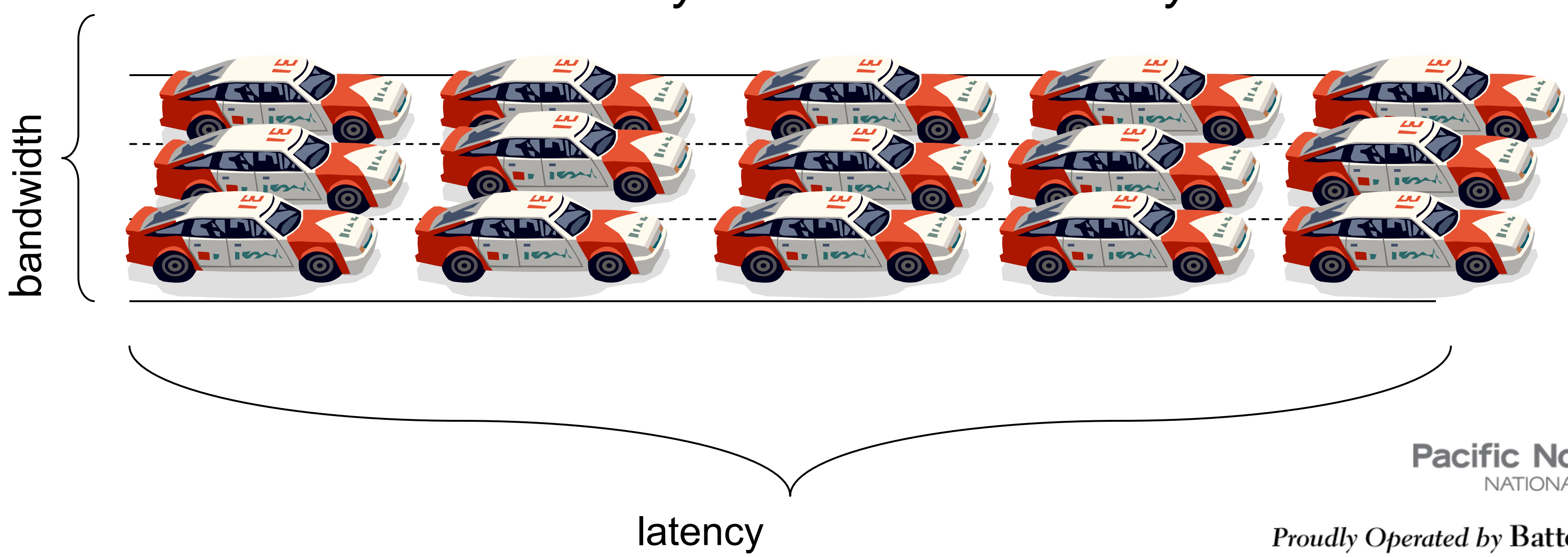
FORGET EVERY THING YOU LEARNED IN THEORY AND ALGORITHMS CLASSES

Significant constraints for parallel graph processing

Hiding memory latencies

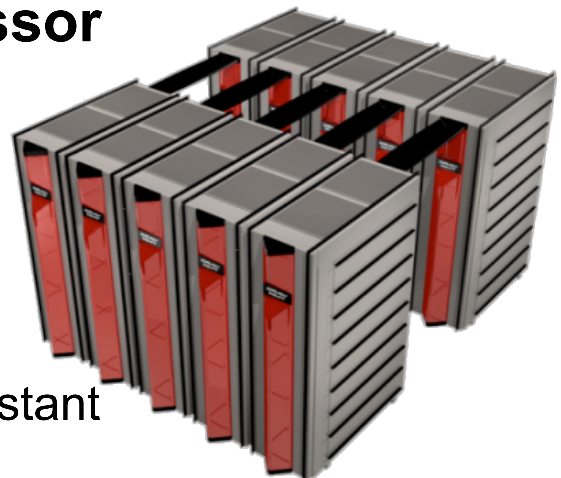
- Caches
  - Reduce latency by storing some data in fast, nearby memory
- Vectors
  - Amortize latency by fetching *N* words at a time
- Parallelism
  - Hide latency by switching tasks
  - Multithreading uses “Little’s Law:”

*concurrency = bandwidth * latency*



What does the XMT Do?

- **Tolerate latency via extreme multi-threading**
  - Each processor has hardware support for 128 threads
  - Context switch in a single tick
  - No cache or local memory
  - Context switch on memory request
  - Multiple outstanding loads
- **Remote memory requests do not stall processor**
  - Other streams work while the request gets fulfilled
- **Light-weight, word-level synchronization**
  - Minimizes access conflicts
- **Hashed Global Shared Memory**
  - Logically contiguous memory is placed on physically distant memory banks (at a 64-byte granularity)
  - Minimizes hotspots



Ideally suited for parallel graph processing

Outline

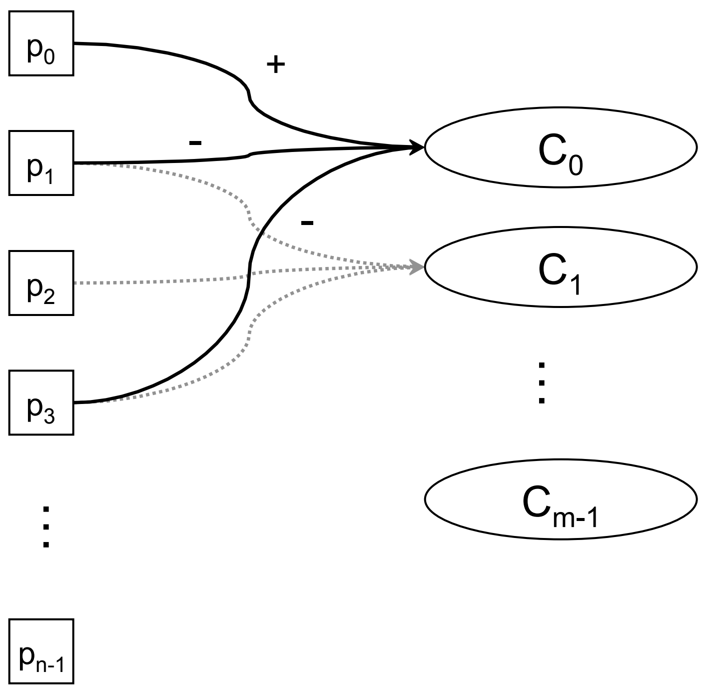
- Parallel Processing for Graphs
- Cray XMT
- ☞ Boolean Satisfiability
  - Survey Propagation
    - Graph Formulation
- Power Grid Contingency Analysis
  - Graph-based problem formulation
- Conclusions

Boolean Satisfiability

- Solve Boolean formula consisting of *and*, *or*, *not* with *N* variables and *M* clauses
  - The *k-satisfiability* or k-SAT problem is usually studied
  - In k-SAT, clauses have at most *k* literals
- Usually, formulas are represented in Conjunctive Normal Form (CNF) or Disjunctive Normal Form (DNF)
- CNF represents formulas as a conjunction of disjunctions of literals: $F = \left(\bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} l_{i,j} \right) \right)$
- In general, the k ≥ 3 SAT problem is NP-complete
- However, Boolean satisfiability has many applications and its “efficient” solution is of practical importance

Survey Propagation

- Survey Propagation (SP) is a *message-passing* algorithm
  - Boolean formulas are represented as bipartite graphs
  - One type of nodes corresponds to variables, another to clauses
  - Edges indicate that a variable appears in a specific clause, the edge includes information on whether the variable appears in the positive or negative form



Survey Propagation (cont.)

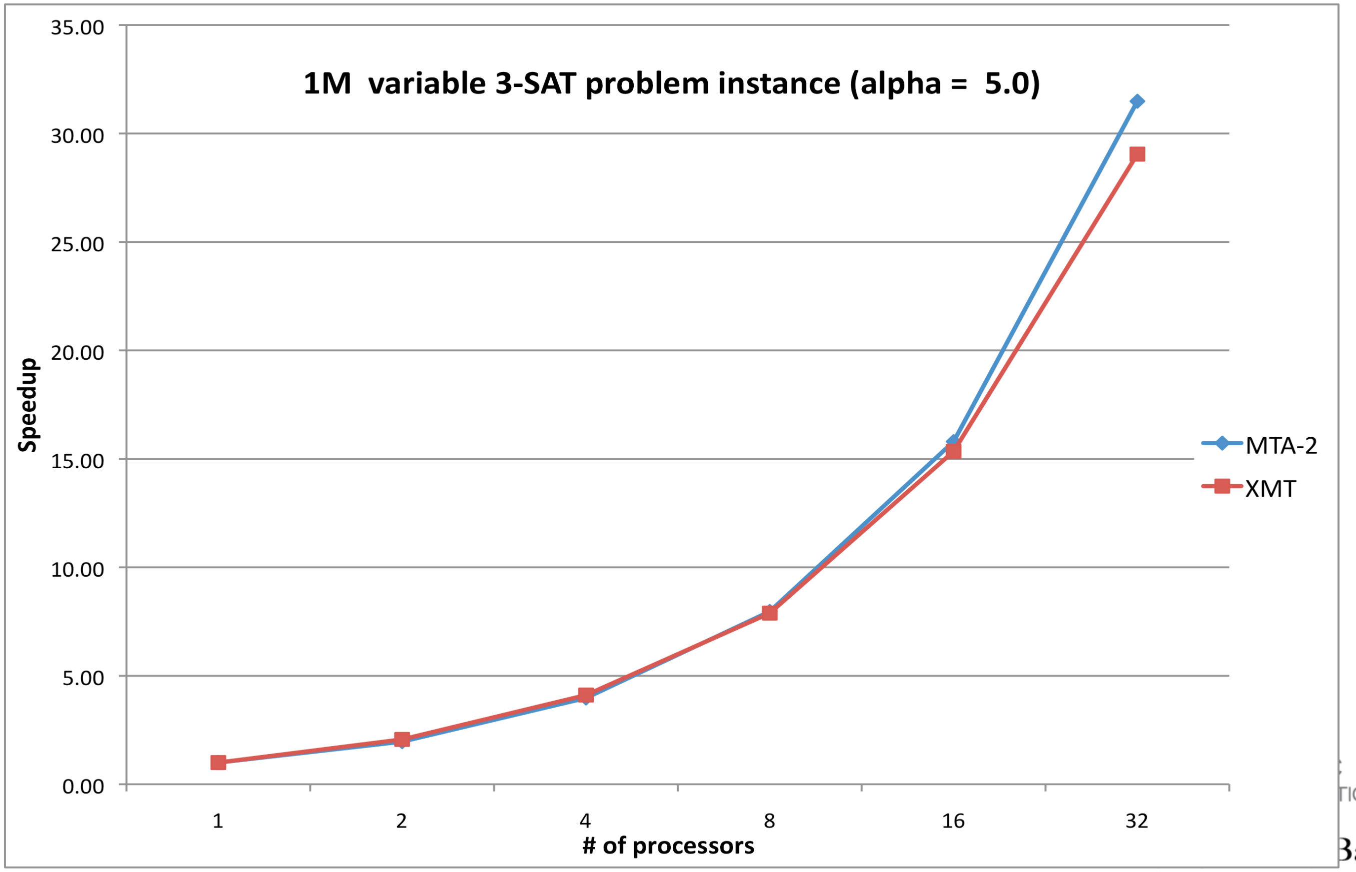
- Messages passed between the two types of nodes indicate probabilities of variables taking true or false Boolean variables
  - Probabilities are used to compute weights for individual variable & clause nodes
  - The variable with the highest weight difference (bias) is selected for *fixing* its value accordingly
  - Probability distributions are then updated and the information is propagated along the edges for nodes to recompute their weights
  - This process continues until a fixed point is reached
  - At this point, SP has a partial solution to the Boolean formula which is can then be easily completed using traditional SAT solver
- See A. Braunstein, M. Mezard, and R. Zecchina. Survey propagation: an algorithm for satisfiability. *Random Structures and Algorithms*, 27(2):201-226, September 2005.

Survey Propagation (cont.)

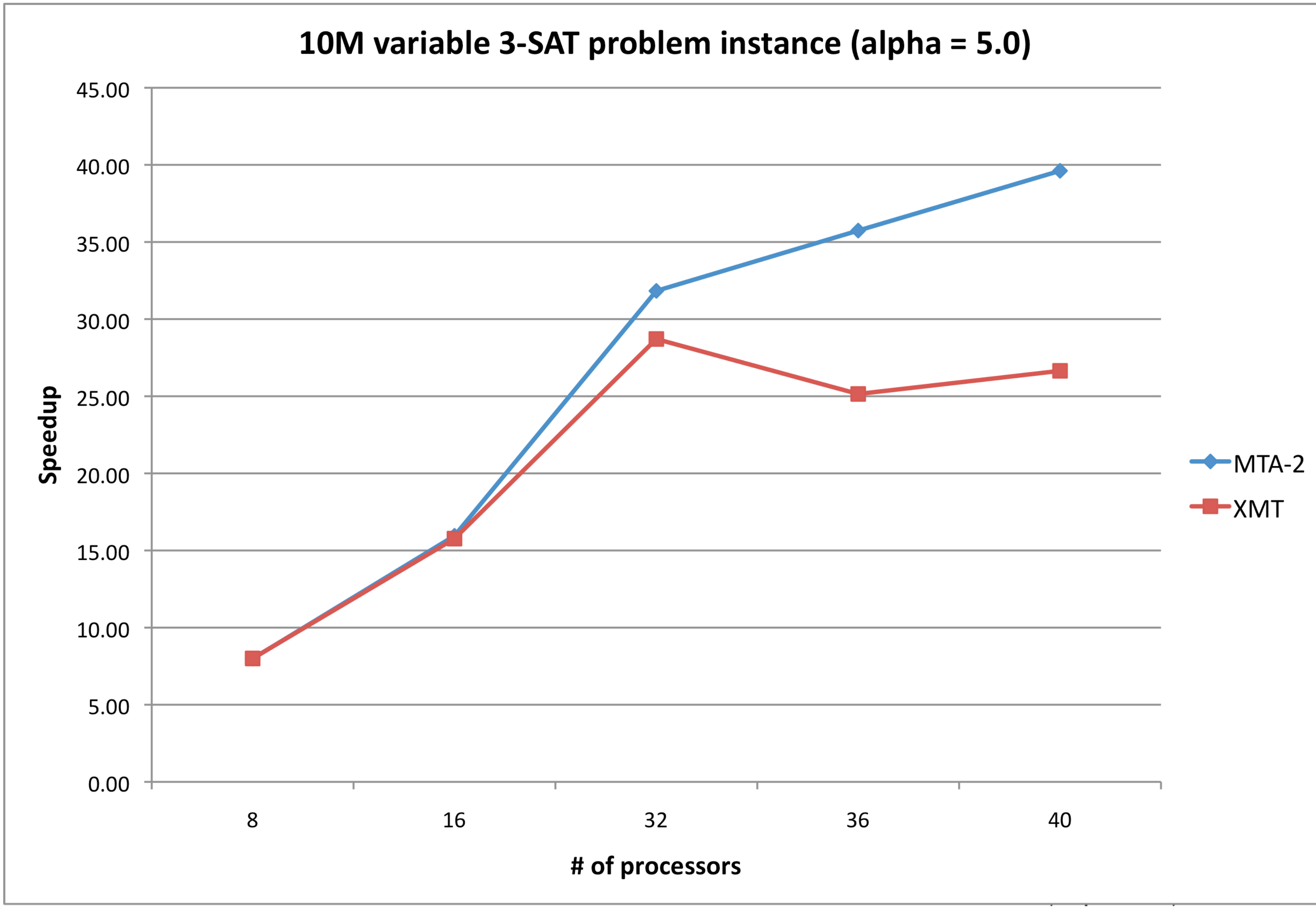
- Distributed Survey Propagation enhances the original SP formulation
  - Enables variable nodes to make decisions of when to fix their value with purely local information
  - The original SP formulation required fixing the variable with the globally largest bias
    - It also required processing the messages between nodes in the graph in a serial order
  - Distributed SP enables a fully parallel implementation
- See J. Chavas, C. Furtlehner, M. Mezard, and R. Zecchina. Survey-propagation decimation through distributed local computations. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11016, 2005.

Survey Propagation on the Cray XMT

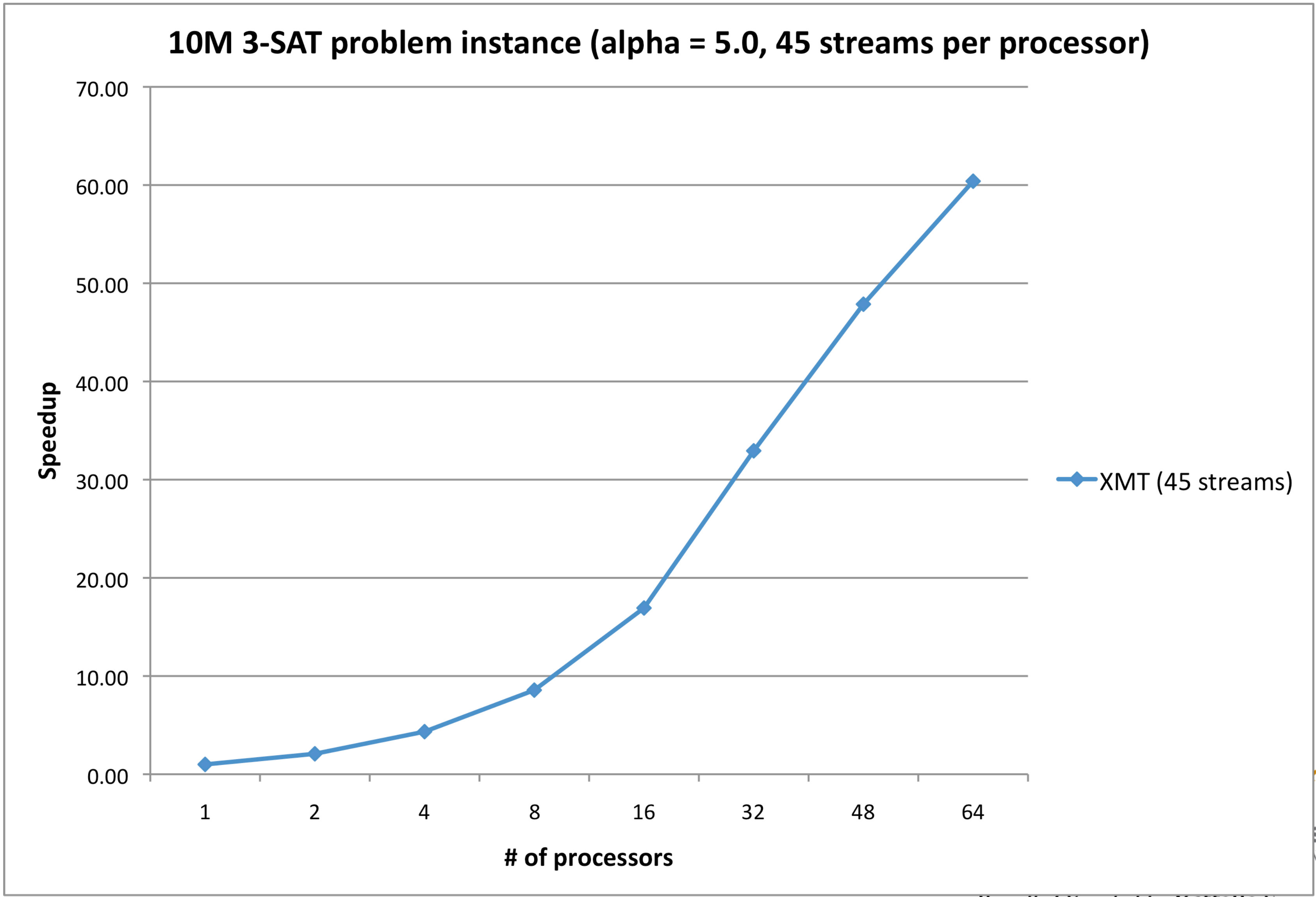
- We have implemented Distributed SP on the Cray MTA-2 & XMT
  - Conducted scalability experiments for large random k-SAT instances



Survey Propagation on the Cray XMT (cont.)



Survey Propagation on the Cray XMT (cont.)

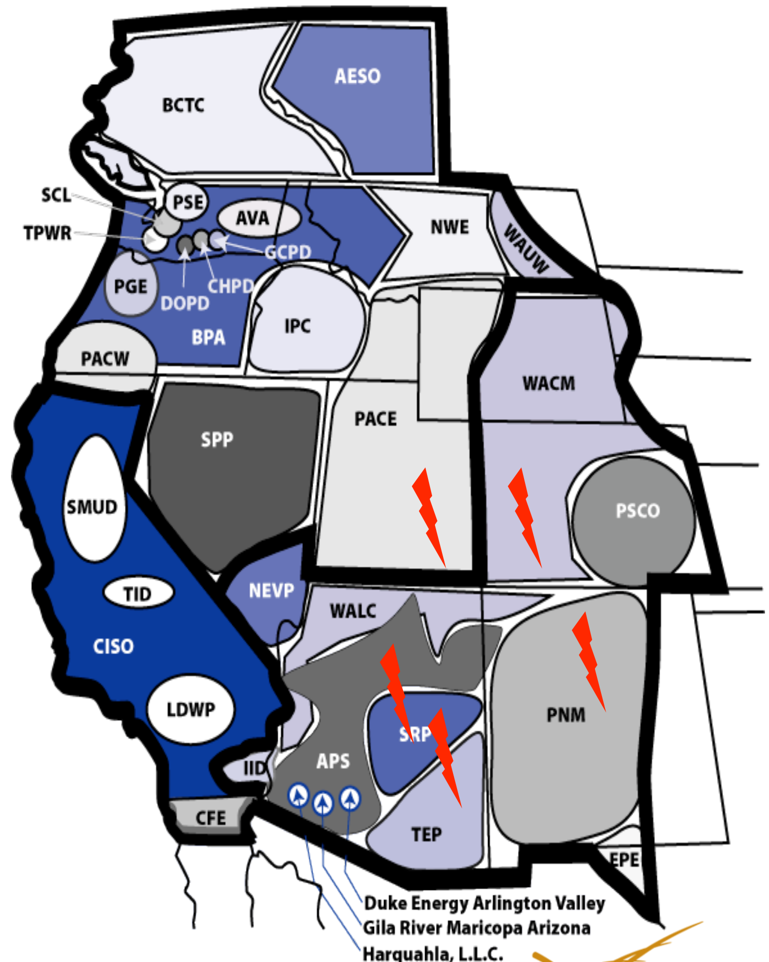


Outline

- Parallel Processing for Graphs
- Cray XMT
- Boolean Satisfiability
  - Survey Propagation
    - Graph Formulation
- ☞ Power Grid Contingency Analysis
  - Graph-based problem formulation
- Conclusions

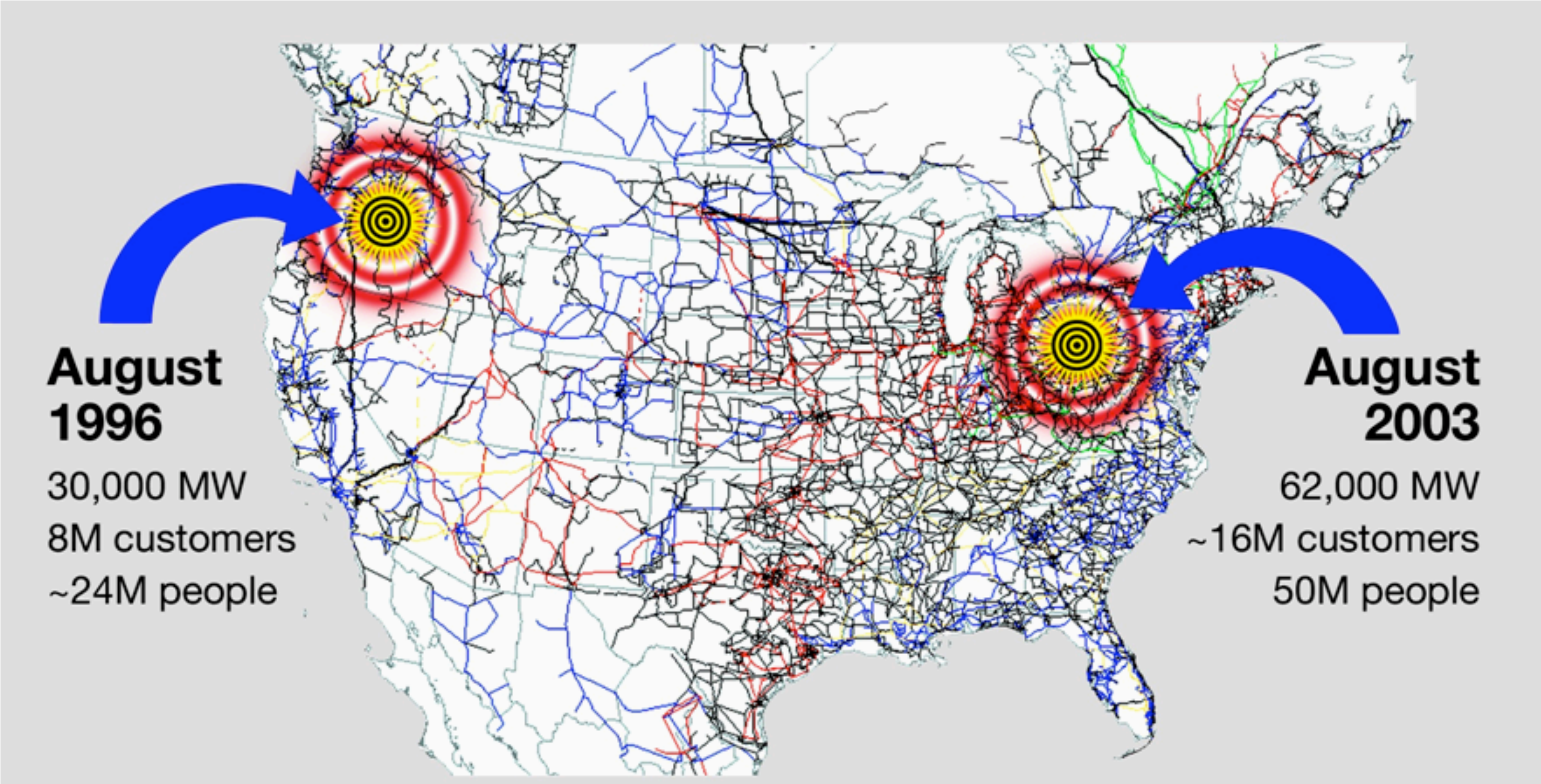
Role of Contingency Analysis

- From "N-1" to "N-x"
  - To improve situational awareness
- From Balancing Authorities to a Wide Area
  - Example: 35 BAs in west
  - Further require **"N-x"** CA
  - To better understand cascading failures
- N-x Contingency Analysis
  - Result in a large number of cases. "N-5" → 10^{20} cases for the west =~ 10^{20} seconds + lots of data
  - Needs: better contingency selection and post-processing



Project led by Henry Huang @ PNNL

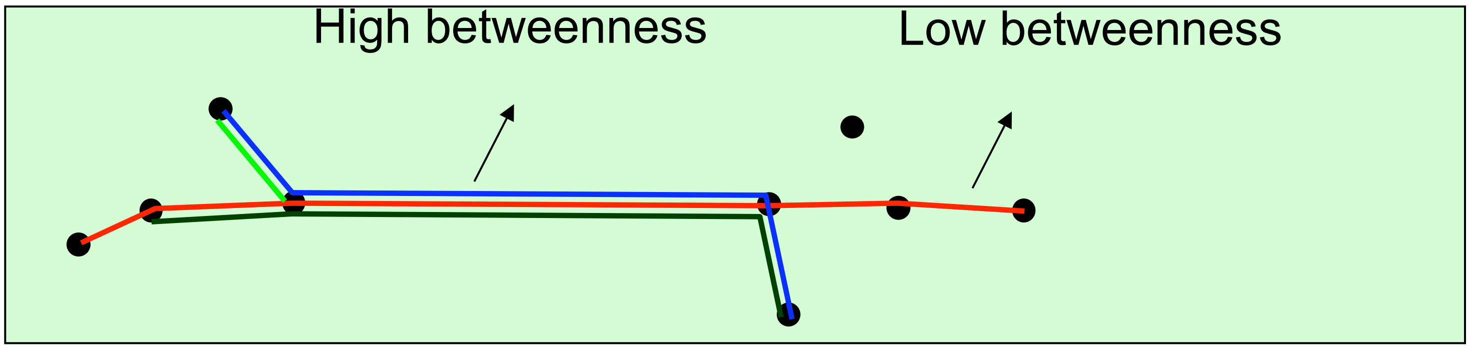
Consequences of Poor Situational Awareness



The need to improve situational awareness became clear.

Edge Betweeness Centrality for Weighted Graphs

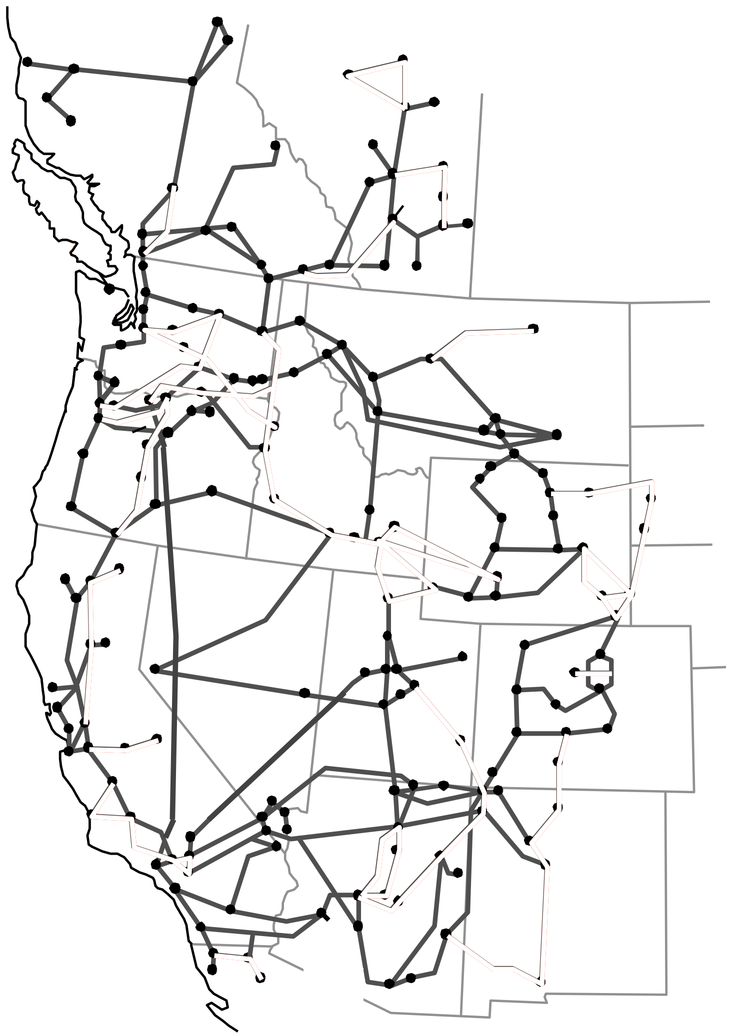
- **Contingency selection with graph centrality**
  - High betweenness identifies heavily traveled edges in graph



Power Grid Centrality

Betweennness Centrality

— non-critical elements



"N-x"

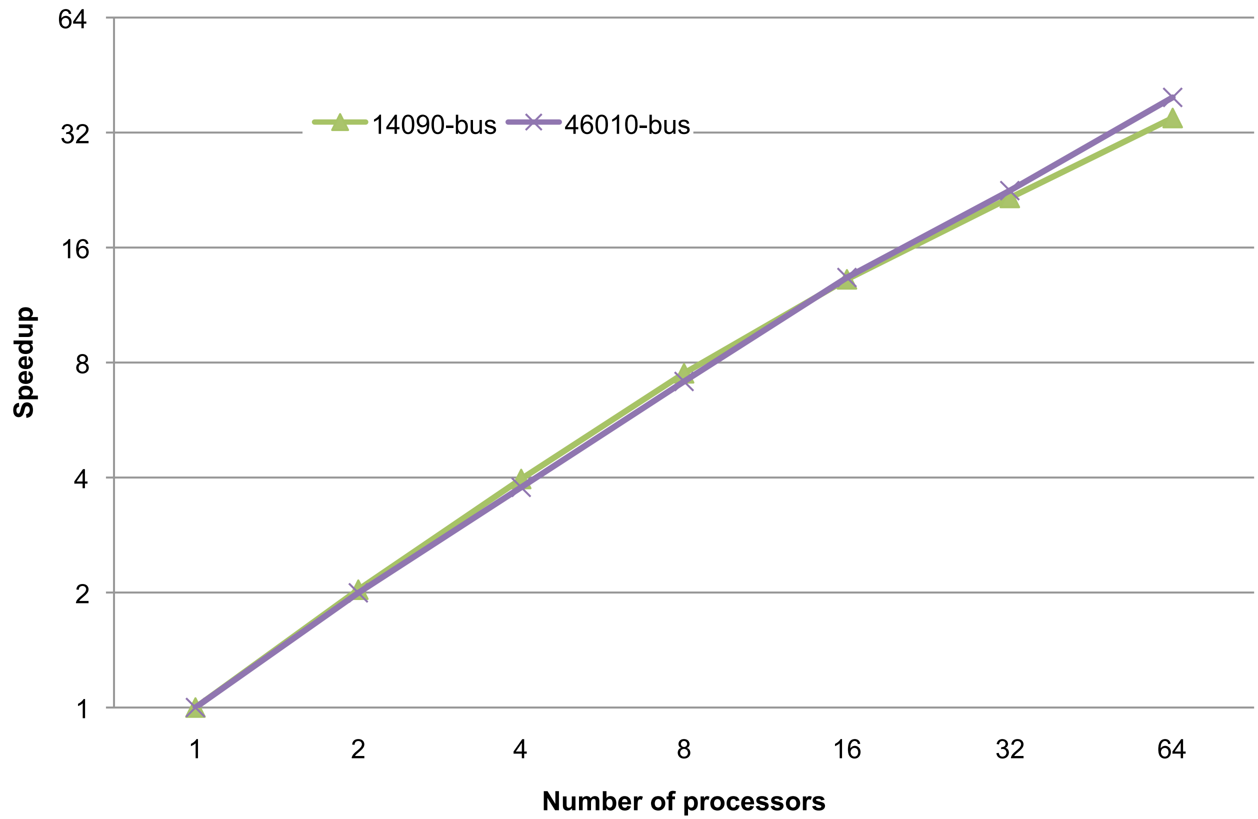
C_x^{N-b}

Western Power Grid

"N-5" ➔ $C_5^{17000} \sim= 10^{20}$

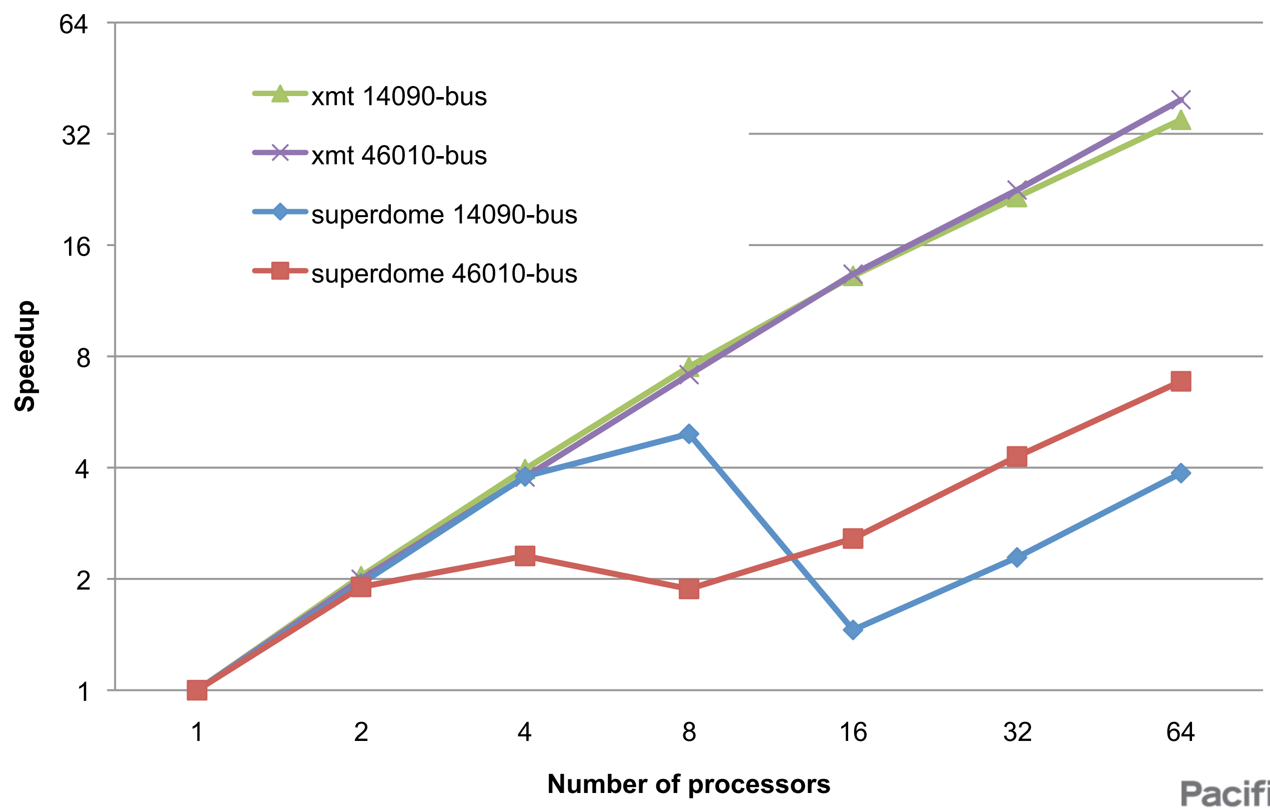
Performance on XMT

- Good scalability on XMT



Performance on shared-memory cache-based platform

- Compared XMT performance with HP Superdome
  - Speedup on Superdome encounters a “knee” phenomenon



Conclusions

- Presented two case studies on the use of the Cray XMT for parallel computation on graphs
- Hardware & software features of the XMT are a better fit than other platforms for graph computation
  - Global shared memory
  - Very fine-grained threading
  - Efficient, fine-grained synchronization
  - **Latency hiding through multithreading**
- Ongoing work on the use of multithreaded systems for graph computations and other irregular applications
  - Part of the Center for Adaptive Supercomputing Software – MultiThreaded architectures (**CASS-MT**)
- More information at http://cass-mt.pnl.gov
  - Please talk to me if you want an account on our XMT