TN

CI

LOCALIZING COMMUNICATION IN SPARSE MATRIX OPERATIONS

Allocation: Illinois/50 Knh
PI: Luke Olson[1]
Collaborator: Amanda Bienz[1]

[1]University of Illinois at Urbana-Champaign

EXECUTIVE SUMMARY

Parallel sparse iterative methods, such as algebraic multigrid (AMG), solve a variety of linear systems in virtually every field of science and engineering. Supercomputers such as Blue Waters provide sufficient memory and bandwidth to solve extremely large systems, enabling the simulation of more complex problems. However, the standard computational kernels for sparse matrix operations in methods such as the AMG algorithm lack the scalability required to take full advantage of current hardware. The Blue Waters allocation has exposed an opportunity to utilize the topology of the underlying communication network to reduce the communication costs associated with the parallel matrix operations that dominate the cost of each iteration of AMG.

RESEARCH CHALLENGE

Sparse matrices arise in many large-scale simulations. Sparse matrix operations such as matrix–vector multiplication and matrix–matrix multiplication are key computational kernels and demand significant resources on the machine, largely in communication among processing elements. When multiple processors on a node of Blue Waters communicate with processors on distant nodes, communication costs are increased further. The key challenge of this project is to limit costly, internode communication through localization of the sparse matrix routines.

METHODS & CODES

Current supercomputing architectures consist of a large number of nodes, each with several multi-core processors. For example, many sparse matrix operations use 16 or 32 processors per node on Blue Waters. The standard approach to sparse matrix communication does not consider the physical location of processors on the network. Yet, the cost of communication varies greatly depending on the locations of each endpoint. For instance, communication between two processes located on the same node incurs a significantly lower cost than communicating between two different nodes.

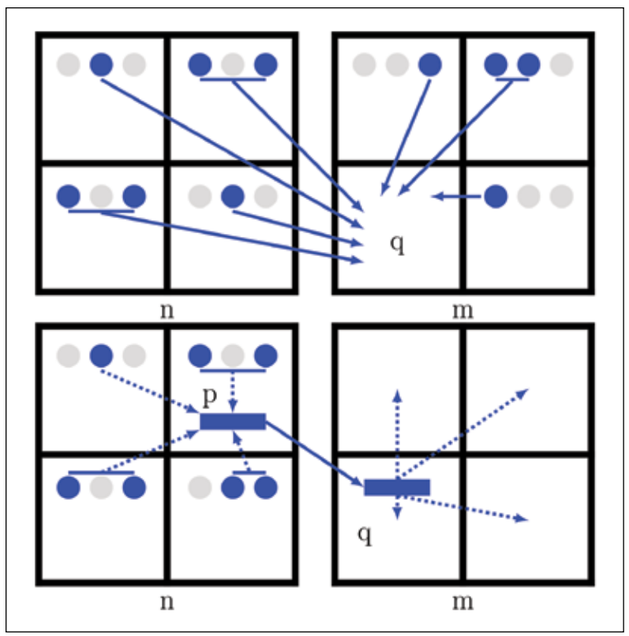

Figure 1: Communication patterns in the *standard* and *topology-aware* algorithms.

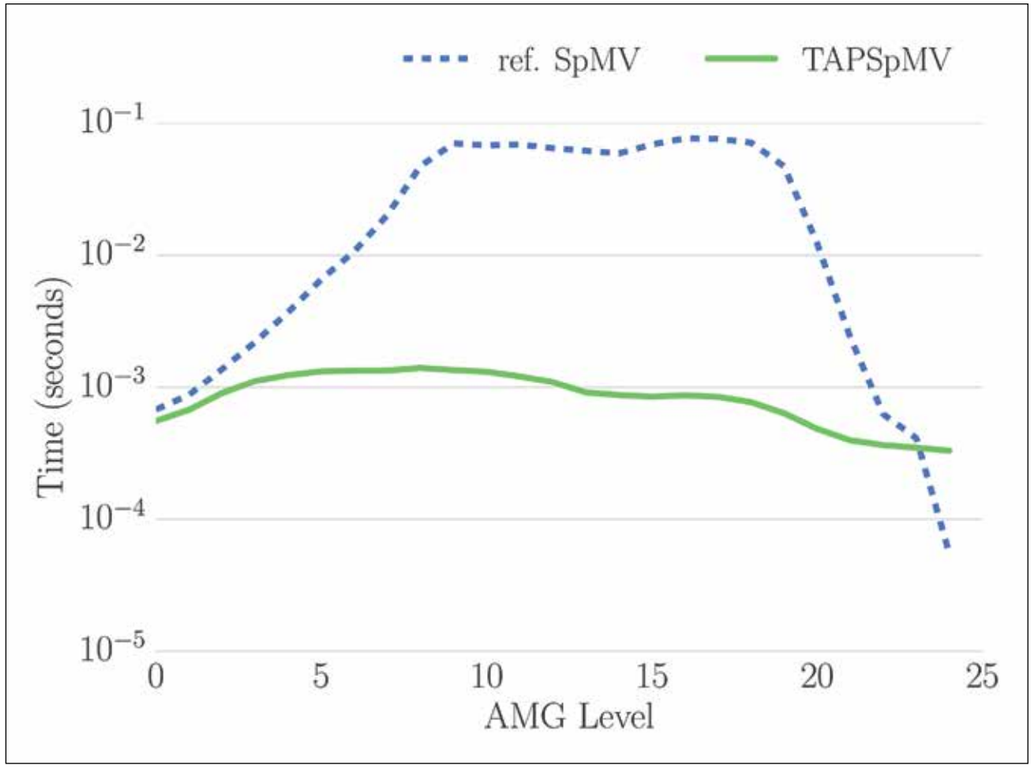

Figure 2: Time for *standard* and *topology-aware* sparse matrix-vector multiplication (SpMV).

The method developed in this work reroutes communication inside of sparse matrix operations so that MPI messages are aggregated on a node before executing internode communication on the network. This is shown in Fig. 1, where processes on nodes *n* and *m* typically send to some process *q*, sending directly to process *q* regardless of the starting location. In contrast, the method developed here considers the topology for communication. All processes on node *n* send everything with destination on node *m* to some local process *p*. After all data are collected, process *p* sends a single message across the network to process *q*, where it is then distributed to processes on node *m*.

RESULTS & IMPACT

Topology-aware methods on Blue Waters have shown the potential to greatly reduce the dominant communication costs of sparse matrix–vector multipliers (SpMVs) when many messages are sent between sets of nodes. Fig. 2 shows a large reduction in cost for SpMVs on coarse levels of an AMG hierarchy; in contrast, standard communication sends a large number of small messages.

The results highlight the value in taking advantage of processor layout and topology in irregular communication demands, such as those introduced through sparse matrix operations. As applications and data demands continue to grow in complexity and dimension, localizing communication will be critical to achieving efficiency and taking advantage of the full capacity of the network.

WHY BLUE WATERS

Blue Waters is an ideal platform for testing scalable algorithms for future machines. The method developed in this project could be extended to additional elements of the machine topology, including the socket level and also the Gemini hubs on the network. The scale and network type made Blue Waters a necessary component of the experimentation to support the algorithm development and modeling.