Annual Report for Blue Waters Professor Allocation

- **Project Information**
  - Title: Algorithms for extreme scale systems
  - PI: William Gropp, University of Illinois Urbana-Champaign
  - Collaborators: Luke Olson, University of Illinois Urbana-Champaign
  - Contact: wgropp@illinois.edu

- **Executive summary (150 words)**

Continued increases in the performance of large-scale systems will come from greater parallelism at all levels. At the node level, we see this both in the increasing number of cores per processor and the use of large numbers of simpler computing elements in GPGPUs. The largest systems must network tens of thousands of nodes together to achieve the performance required for the most challenging computations. Successfully using these systems requires new algorithms and new programming systems. My research looks at the effective use of extreme scale systems. The most notable result over the last year has been the development of a new communication model that is both simple and a far better fit to multicore systems. This model has inspired new algorithms for sparse matrix-vector products. Other work has show improved scaling for new formulations of the Conjugate Gradient method.

- **Description of research activities and results**
  - *Key Challenges:* At extreme scale, even small inefficiencies can cascade to limit the overall efficiency of an application. New algorithms and programming approaches are needed to address barriers to performance.
  - *Why it Matters*: This work directly targets current barriers to effective use of extreme scale systems by applications. For example, Krylov methods such as Conjugate Gradient are used in many applications currently being run on Blue Waters (MILC is one well-known example). Developing and demonstrating a more scalable version of this algorithm would immediately benefit those applications. In the longer term, the techniques that are developed will provide guidance for the development of highly scalable applications.
  - *Why Blue Waters:* Scalability research relies on the ability to run experiments at large scale, requiring tens of thousands of nodes and hundreds of thousands of processes and cores. Blue Waters provides one of the few available environments where such large-scale experiments can be run. In addition, only Blue Waters provides a highly capable I/O system, which we plan to use in developing improved approaches to extreme-scale I/O.
  - *Accomplishments:* Over the last year, we have shown that the classic "postal" model of communication performance is not suitable to today's multicore processors. This model has inspired new algorithms that can significantly improve communication performance; in some sparse matrix-vector product tests for Multigrid, we have seen as much as an order of magnitude improvement in performance. We continue to explore alternative

formulations of conjugate gradient that eliminate some of the strict barrier synchronization as well as better use the memory hierarchy, as well as algorithmic approaches to resilience that exploit the multilevel representation in multigrid. Current work on alternative formulations of Krylov models has been limited by the fact that the nonblocking MPI_Allreduce on Blue Waters is functional but does not provide the expected (or perhaps hoped for) performance, particularly in terms of the ability to overlap the Allreduce operation with other communication and computation.

- **List of publications and presentations associated with this work**

Publications

1. Gropp, W., L.N. Olson, and P. Samfass, *Modeling MPI Communication Performance on SMP Nodes: Is it Time to Retire the Ping Pong Test*, in *Proceedings of the 23rd European MPI Users' Group Meeting*. 2016, ACM: Edinburgh, United Kingdom. p. 41-50.
2. Eller, P.R. and W. Gropp, *Scalable non-blocking preconditioned conjugate gradient methods*, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2016, IEEE Press: Salt Lake City, Utah. p. 1-12.
3. Bienz, A., et al., *Reducing Parallel Communication in Algebraic Multigrid through Sparsification.* SIAM Journal on Scientific Computing, 2016. **38**(5): p. S332-S357.

Presentations

These are some of the presentations that included reference to Blue Waters:

1. Meeting the Communication Needs of Applications, Keynote for First International Workshop in Communication Optimizations in HPC, November 2016, Salt Lake City, UT.
2. Thinking About Parallelism and Programming, Presentation in the Ken Kennedy award session at SC16, November 2016, Salt Lake City, UT. A video of this presentation is available.
3. Do You Know What Your I/O Is Doing? (and how to fix it?), at Workshop on Clusters, Clouds, and Data for Scientific Computing (CCDSC 2016), near Lyon, France, October, 2016.
4. Modeling MPI Communication Performance on SMP Nodes: Is it Time to Retire the Ping Pong Test, with Luke Olson and Philipp Samfass, at EuroMPI16, September, 2016, Edinburgh, Scotland.
5. MPI: The Once and Future King, Keynote at EuroMPI16, September, 2016, Edinburgh, Scotland.
6. Reproducibility of Computations and Distributed Data Structures, at Workshop on Batched, Reproducible, and Reduced Precision BLAS, May 18-19, 2016, Knoxville, TN.
7. MPI+MPI: Using MPI-3 Shared Memory as a Multicore Programming System, at SIAM Conference on Parallel Processing for Scientific Computing, April 2016, Paris, France.

Plan for next year

These projects have made good progress over the last year and are expected to expand their need for scalability studies. In addition, based on the experience with poor I/O performance, we expect to begin looking at parallel I/O approaches, beyond what can be supported in our PAID IO project. The research efforts for the next year include

1. Optimizing parallel sparse matrix-vector product by optimizing intra- and inter-node communication
2. Parallel I/O autotuning and adaptivity
3. Communication optimized Krylov methods.
4. Resilient algorithms for Multigrid and multigrid preconditioned Krylov methods.

Most of these experiments study behavior at scale and typically need only a short run time but with 10,000-20,000 nodes. In order to produce timings at scale that are consistent, reproducible, and accurate, a typical run may require anywhere from a few minutes to 30 minutes per test. Thus, tests at scale may require 1,000-10,000 node-hours each. Because these tests are being used to evaluate different algorithms, most of which are developed as a result of evaluating the results of experiments on Blue Waters and at scale, it is difficult to determine a priori the amount of time that will be needed. Over the past year, we were careful to limit the scale for tests in order to limit the amount of resources consumed; as a result, we used a little under 50,000 node hours. In the upcoming year, I expect several projects to run at full scale by the end of the year. If each of the 4 projects requires 2 tests at full scale (each taking 20 minutes at 20,000 nodes), along with a sequence of scaling tests (another 50%) and some development time, 80,000 node hours would be needed. Depending on the progress of the algorithm development efforts, more time (as much as the 245,000 node-hour allocation) or less may be required. An exact estimate simply is not possible for this type of basic computer science research. For a specific request, 80,000 node hours should be sufficient; however, the option for more time, up to the original allocation, is highly desirable.

Few other resources will be needed. While some IO scalability studies may require files in the multi-Terabyte range, these will be temporary files. Similarly, little networking is expected.

Estimated distribution of time: Q1: 20%, Q2: 20%; Q3: 30%; Q4: 30%

Rationale for the distribution: In the first quarter, the research projects will continue. Over the course of the year, greater scalability will be investigated, requiring larger runs. In addition I expect to add work on graph algorithms, again running some benchmarks at scale.