Lessons From Analyzing Fan-In Communications

Terry Jones, Bradley Settlemyer
Computer Science and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN USA
{trj, settlemyerbw} @ornl.gov

Abstract— **We present a study of an important class of communication operations—the fan-in communication pattern. By its nature, fan-in communications form 'hot spots' that present significant challenges for any interconnect fabric and communication software stack. Yet despite the inherent challenges, these communication patterns are common in both applications (which often perform reductions and other collective operations that include fan-in communication such as barriers) and system software (where they assume an important role within parallel file systems and other components requiring high-bandwidth or low-latency I/O). Our study determines the effectiveness of differing client-server fan-in strategies. We describe fan-in performance in terms of aggregate bandwidth in the presence of varying degrees of congestion, as well as several other key attributes. Results are presented for two representative supercomputers (a large Cray Aries-interconnect based super-computer and a large Gemini-interconnect based super-computer). Finally, we provide recommended communication strategies based on our findings.**

Keywords—Networks and Interconnects; performance modeling and evaluation; scalability

I. Introduction

Fan-in communication patterns are collections of data transfer between two groups of distributed nodes in which one group is larger in node count than the other (*M to N* communications where $M > N$). Many times, the smaller group of nodes consists of a single node forming an *N-ary tree* graph (see POSTER's *Motivation Section Figure* 1). Fan-in communications may be characterized in terms of scalability (incorporating both bandwidth and latency aspects), fairness, and performance variability. Because of their pervasiveness and their capacity to be an impediment to overall performance, fan-in communications are important to both applications and system software.

Our interest stems from research of user-level reservation schemes within a High Performance Computing (HPC) context. Providing a flexible and acceptable quality-of- service (QoS) capability within an HPC parallel file system has proved to be a very challenging problem. Fan-in patterns intrinsically create hot-spots; congestion caused by hot-spot traffic can significantly degrade the performance of a computer network therefore requiring specialized techniques for optimizing one or more fan-in attributes. A reservation scheme coordinates message traffic flow by establishing a multiplexing capability. In our previous work, we presented several techniques to improve the fine-grained communication of a sophisticated application (NAMD) using the uGNI library for Gemini [1], fan-in based time synchronization protocols [2], as well as performance and scalability measurements of key parallel file system components [3]. Our current work is focused at user-level communication strategies and is being explored with a tool we have developed on top of MPI which we call *reservation* [4]; this paper extends the work presented in [5]. Using the *reservation* tool, we are able to arbitrarily establish the scale of the fan-in tree as well as the other pertinent communication factors including the number of large message-size clients, the number of small message-size clients, the associated buffer sizes, and details about request queues and fairness properties. We have chosen to implement our strategies over MPI which carries certain benefits and liabilities. Unlike our earlier work, MPI provides less control than programming the uGNI layer directly. On the positive side, all techniques that we have employed are available to uGNI-based implementations (the reverse is not necessarily true), our methods are available to a wide audience including MPI-based applications, and we were able to fully explore the strategies we wished to investigate over an MPI layer. Finally, our work seeks to address general solutions and we do not require pre-determined coordination as in some collective reductions or gather operation strategies.

Our contributions therefore are: *reservation*, a performance analysis tool designed specifically to explore fan-in communication patterns; an assessment of the fan-in capabilities of Gemini 3D-torus interconnect; a comparison to a different topology (Cray Aries, a Dragonfly topology); and a collection of recommended communication strategies based on our findings.

II. Testing Environment & Methodologies

Frequently in parallel machines, it is useful to have a client-server arrangement. While multiple client scenarios are common, it is often desirable to limit the number of servers to avoid inconsistencies in data state. Such an arrangement naturally results in fan-in communications. Fan-in communication patterns are collections of data transfer between two groups of distributed nodes in which one group is larger in node count than the other (*M to N* communications where $M > N$), see Figure 1. Many times, the smaller group of nodes consists of a single node forming an *N-ary tree* graph.

A. A Tool For Evaluating Fan-Ins: Reservation

The Reservation tool is an MPI program capable of measuring the performance for various fan-in configurations. It measures performance for a single-queue design as well as a dual-queue design. The dual-queue measurements result from large-buffers being handled separately than small buffers and control messages. In addition, it permits adjustable sizes for both "small" messages and "large" messages. Finally, messages may be either matched with wildcard match-any tags, or with specific matching..

III. Experimentation and Data Measurements

We began our investigation by conducting multiple measurements of a basic fan-in scenario in which a variable number of large buffer clients exchange messages with one root node; the average results are presented in Figure 5. The graph (and the subsequent graphs) show aggregate bandwidth as measured at the fan-in root on the Y-axis versus the number of large-buffer clients on the X-axis (higher numbers on Y-axis is better). In each test, *large-buffer* is defined as 4MB and *small buffer* is defined as 4K. The four series of the POSTER's *Results Section Figure* 1 present results for: (a) Aries interconnect with only large-buffer clients; (b) Aries interconnect with both large-buffer clients and small-buffer clients where the number of small-buffer and large-buffer clients are arranged in a 4:1 ratio; (c) Germini interconnect with only large-buffer clients, (d) Gemini interconnect with both large-buffer clients and small-buffer clients – also with small-buffer and large-buffer clients in a 4:1 ratio.

Since POSTER's *Results Figure* 1 presents averages, it's interesting to look a little deeper into the data and check for variability between runs. The POSTER's *Results Section Figure* 2 present individual runs on both Gemini and Aries. The remarkable consistency of the Aries interconnect stands out in Figure 7. In fact, the multiple runs appear to be a single series as each indivdual series lies almost exactly on its predecessor:

The submitted manuscript has been authored by a contractor of the U.S. Government under Contract No. DE-AC05-00OR22725. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

References

[1] Yanhua Sun, Gengbin Zheng, Chao Mei, Eric Bohm, James Phillips, Laxmikant Kale, and Terry Jones. Optimizing Fine-grained Communication in a Biomolecular Simulation Application on Cray XK6. *The 25th International Conference for High Performance Computing, Networking, Storage and Analysis (SC'12)*. Salt Lake City, UT. November 2012.

[2] Terry Jones, and Gregory Koenig. "Clock Synchronization in High-end Computing Environments: A Strategy for Minimizing Clock Variance at Runtime." *Journal of Concurrency and Computation: Practice & Experience*, Volume 25, Issue 6, doi: 10.1002/cpe.2868, pages 881-897, April 25, 2013.

[3] Feiyi Wang, Mark Nelson, Sarp Oral, Scott Atchley, Sage Weil, Bradley W. Settlemyer, Blake Caldwell, and Jason Hill. 2013. Performance and scalability evaluation of the Ceph parallel file system. In *Proceedings of the 8th Parallel Data Storage Workshop* (PDSW '13). ACM, New York, NY, USA, 14-19. DOI=10.1145/2538542.2538562 .

[4] ORNL LDRD Project: Towards a Resilient and Scalable Infrastructure for Big Data. Oak Ridge National Laboratory.

[5] Terry Jones, Bradley Settlemyer. "Fan-in Communication On A Cray Germini Interconnect." *CUG 2014*, Lugano, Switzerland. May 2014.