Experiences Visualizing Multi-cluster Parallel Applications

Lars Ailo Bongo, Otto J. Anshus and John Markus Bjørndalen

larsab@cs.uit.no, otto@cs.uit.no, johnm@cs.uit.no

Department of Computer Science,
University of Tromsø

Abstract

It is hard to make a parallel application scale with regards to speedup when using a cluster of computers. In [12] a speedup of around two or less is reported when using 32 processors. It is also documented how using more processors can result in a slowdown of the application. Our results [3, 4] are in accordance with this when using LAM-MPI [9], PVM [8], and PastSet [1, 15]. We contribute this to several factors, but for this paper we will focus on the configuration of the application onto the cluster(s) used.

The PATHS system [2] allows us to specify a set of configuration parameters about the mapping of threads and processes to computers in the clusters. An application is then configured by creating paths between communicating threads.

Being able to visualize the behavior of an application is an important aid during performance debugging. Steps [5, 6] is a software post-mortem event-based monitor used to analyze and visualize both the communication behavior of applications using PATHS, and the configuration.

We describe our experiences visualizing the configuation and the behavior of a parallel wind-tunnel application comprised of 332 worker threads on a cluster with 30 nodes and 100 processors.

The graph used to visualize the paths has about 3000 nodes and 3000 edges. With a typical 21 inch 1280*1024 display, the user can only see a part of the graph at a time. To examine different parts of the graph, it must be zoomed or the window must be scrolled. The standard display resolution and size makes it difficult to get an overview of the visualized system, and to find specific information.

Visualizing the communications events can be used to get an overview of the behavior of the application. This can be done by showing the time spent computing vs. communicating. However, the high number of threads, and long running time of parallel applications typically makes the visualization too tiny to read on a standard display. The time spent communicating is often much smaller than the computation time, resulting in communication events becoming practically invisible to the naked eye. Similar problems are described in other performance analysis tools [11, 13, 14, 7].

We are interested in examining how can utilize a tiled display wall [10] to do the visualizations. We believe the large size and high resolution can be used to do visualizations with both overviews and details at the same time. A display wall can also support concurrent visualizations at different abstraction levels. The cluster driving the display wall also supports parallelizing the visualizations both with regards to processing and output to the display wall.

References

[1] ANSHUS, O. J., AND LARSEN, T. Macroscope: The abstractions of a distributed operating system. *Norsk Informatikk Konferanse* (October 1992).

[2] BJØRNDALEN, J. M. PhD thesis, University of Tromsø, To be sumbitted 2003.

[3] BJØRNDALEN, J. M., ANSHUS, O., VINTER, B., AND LARSEN, T. Configurable collective communication in lam-mpi. *Proceedings of Communicating Process Architectures 2002, Reading, UK* (September 2002).

[4] BJØRNDALEN, J. M., ANSHUS, O., VINTER, B., AND LARSEN, T. The performance of configurable collective communication for lam-mpi in clusters and multi-clusters. *NIK 2002, Norsk Informatikk Konferanse, Kongsberg, Norway* (November 2002).

[5] BONGO, L. A. Steps: A performance monitoring and visualization tool for multicluster parallel programs, June 2002. Large term project, Department of Computer Science, University of Tromsø.

[6] BONGO, L. A., ANSHUS, O., AND BJØRNDALEN, J. M. Cluster monitoring with Steps: Making the application behaviour visible. *Work in progress* (2003).

[7] FOSTER, I., AND KESSELMAN, C., Eds. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufman, 1999.

[8] GEIST, A., BEGUELIN, A., DONGARRA, J., JIANG, W., MANCHEK, R., AND SUNDERAM, V. *PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Networked Parallel Computing*. MIT Press, 1994.

[9] http://www.lam-mpi.org/.

[10] LI, K., CHEN, H., CHEN, Y., CLARK, D. W., COOK, P., DAMIANAKIS, S., ESSL, G., FINKELSTEIN, A., FUNKHOUSER, T., HOUSEL, T., KLEIN, A., LIU, Z., PRAUN, E., SAMANTA, R., SHEDD, B., SINGH, J. P., TZANETAKIS, G., AND ZHENG, J. Early experiences and challenges in building a scalable display wall system.

[11] MILLER, B. P., CALLAGHAN, M. D., CARGILLE, J. M., HOLLINGSWORTH, J. K., IRVIN, R. B., KARAVANIC, K. L., KUNCHITHAPADAM, K., AND NEWHALL, T. The paradyn parallel performance measurement tools. *IEEE Computer* (1995).

[12] MORTON, D., S.OCONNOR, ZHANG, Z., AND HINZMAN, L. The parallelization of a physically based spatially distributed hydrological code for arctic regions · *ACM Symposium on Applied Computing (SAC'98)* (February 1998).

[13] REED, D. A., AYDT, R. A., NOE, R. J., ROTH, P. C., SHIELDS, K. A., SCHWARTZ, B. W., AND TAVERA, L. F. Scalable performance analysis: The pablo performance analysis environment. *IEEE Proc. Scalable Parallel Libraries Conf.* (1993).

[14] TIERNEY, B., JOHNSTON, W. E., CROWLEY, B., HOO, G., BROOKS, C., AND GUNTER, D. The netlogger methodology for high performance distributed systems performance analysis. In *HPDC* (1998), pp. 260–267.

[15] VINTER, B. *PastSet a Structured Distributed Shared Memory System*. PhD thesis, Tromsø University, 1999.