Improved GROMACS Scaling on Ethernet Switched Clusters

Carsten Kutzner[1], David van der Spoel[2], Martin Fechner[1], Erik Lindahl[3], Udo W. Schmitt[1], Bert L. de Groot[1], and Helmut Grubmüller[1]

[1] Department of Theoretical and Computational Biophysics, Max-Planck-Institute of Biophysical Chemistry, Am Fassberg 11, 37077 Göttingen, Germany
[2] Department of Cell and Molecular Biology, Uppsala University, Husargatan 3, S-75124 Uppsala, Sweden
[3] Stockholm Bioinformatics Center, SCFAB, Stockholm University, SE-10691, Stockholm, Sweden

Abstract. We investigated the prerequisites for decent scaling of the GROMACS 3.3 molecular dynamics (MD) code [1] on Ethernet Beowulf clusters. The code uses the MPI standard for communication between the processors and scales well on shared memory supercomputers like the IBM p690 (Regatta) and on Linux clusters with a high-bandwidth/low latency network. On Ethernet switched clusters, however, the scaling typically breaks down as soon as more than two computational nodes are involved. For an 80k atom MD test system, exemplary speedups Sp_N on N CPUs are $Sp_8 = 6.2$, $Sp_{16} = 10$ on a Myrinet dual-CPU 3 GHz Xeon cluster, $Sp_{16} = 11$ on an Infiniband dual-CPU 2.2 GHz Opteron cluster, and $Sp_{32} = 21$ on one Regatta node. However, the maximum speedup we could initially reach on our Gbit Ethernet 2 GHz Opteron cluster was $Sp_4 = 3$ using two dual-CPU nodes. Employing more CPUs only led to slower execution (Table 1).

When using the LAM MPI implementation [2], we identified the all-to-all communication required every time step as the main bottleneck. In this case, a huge amount of simultaneous and therefore colliding messages "floods" the network, resulting in frequent TCP packet loss and time consuming re-trials. Activating Ethernet flow control prevents such network congestion and therefore leads to substantial scaling improvements for up to 16 computer nodes. With flow control we reach $Sp_8 = 5.3$, $Sp_{16} = 7.8$ on dual-CPU nodes, and $Sp_{16} = 8.6$ on single-CPU nodes.

For more nodes this mechanism still fails. In this case, as well as for switches that do not support flow control, further measures have to be taken. Following Ref. [3] we group the communication between M nodes into $M - 1$ phases. During phase $i = 1 \ldots M - 1$ each node sends clockwise to (and receives counterclockwise from) its i^{th} neighbouring node. For large messages, a barrier between the phases ensures that the communication between the individual CPUs on sender and receiver node is completed before the next phase is entered. Thus each full-duplex link is used for one communication stream in each direction at a time.

We then systematically measured the throughput of the ordered all-to-all and of the standard MPI_Alltoall on 4 – 32 single and dual-CPU nodes, both for LAM 7.1.1 and for MPICH-2 1.0.3 [4], with flow control and without. The throughput of the ordered all-to-all is the same with and without

B. Mohr et al. (Eds.): PVM/MPI 2006, LNCS 4192, pp. 405–406, 2006.

flow control. The lengths of the individual messages that have to be transferred during an all-to-all fell within the range of 3 000 . . . 175 000 bytes for our 80k atom test system when run on 4 – 32 processors. In this range the ordered all-to-all often outperforms the standard MPI_Alltoall. The performance difference is most pronounced in the LAM case since MPICH already makes use of optimized all-to-all algorithms [5].

By incorporating the ordered all-to-all into GROMACS, packet loss can be avoided for any number of (identical) multi-CPU nodes. Thus the GROMACS scaling on Ethernet improves significantly, even for switches that lack flow control.

In addition, for the common HP ProCurve 2848 switch we find that for optimum all-to-all performance it is essential how the nodes are connected to the ports of the switch. The HP 2848 is constructed from four 12-port BroadCom BCM5690 subswitches that are connected to a BCM5670 switch fabric. The links between the fabric and subswitches have a capacity of 10 Gbit/s. That implies that each subgroup of 12 ports that is connected to the fabric can at most transfer 10 Gbit/s to the remaining ports. With the ordered all-to-all we found that a maximum of 9 ports per subswitch can be used without losing packets in the switch. This is also demonstrated in the example of the Car-Parinello [6] MD code. The newer HP 3500yl switch does not suffer from this limitation.

Table 1. GROMACS 3.3 on top of LAM 7.1.1. Speedups of the 80k atom test system for standard Ethernet settings (Sp), with activated flow control (Sp_{fc}), and with the ordered all-to-all (Sp_{ord}).

	single-CPU nodes						dual-CPU nodes				
CPUs	1	2	4	8	16	32	2	4	8	16	32
Sp	1.00	1.82	2.24	1.88	1.78	1.73	1.94	3.01	1.93	2.59	3.65
Sp_{fc}	1.00	1.82	3.17	5.47	8.56	1.82	1.94	3.01	5.29	7.84	7.97
Sp_{ord}	1.00	1.78	3.13	5.50	8.22	8.64	1.93	2.90	5.23	7.56	6.85

References

1. van der Spoel, D., Lindahl, E., Hess, B., Groenhof, G., Mark, A.E., Berendsen, H.J.C.: GROMACS: Fast, Flexible, and Free. J. Comput. Chem. 26 (2005) 1701–1718
2. The LAM-MPI Team. http://www.lam-mpi.org/
3. Karwande, A., Yuan, X., Lowenthal, D.K.: An MPI prototype for compiled communication on Ethernet switched clusters. J. Parallel Distrib. Comput. 65 (2005) 1123–1133
4. MPICH-2. http://www-unix.mcs.anl.gov/mpi/mpich/
5. Thakur, R., Rabenseifner, R., Gropp, W.: Optimization of collective communication operations in MPICH. Int. J. High Perform. Comput. Appl. 19 (2005) 49–66
6. Hutter, J., Curioni, A.: Car-Parrinello molecular dynamics on massively parallel computers. ChemPhysChem 6 (2005) 1788–1793