Back-Migration for MPI Jobs in HPC Environments

Chao Wang[1], Frank Mueller[1], Christian Engelmann[2], Stephen L. Scott[2]

[1] Department of Computer Science, North Carolina State University, Raleigh, NC
[2] Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN
mueller@cs.ncsu.edu, phone: +1.919.515.7889, fax: +1.919.515.7896

As the number of nodes in high-performance computing environments keeps increasing, faults are becoming common place. To counter faults, reactive and proactive migration of MPI tasks to a spare nodes have been considered by us in prior work. However, a migrated task could present a bottleneck due to (1) increased hop counts for communication from/to the spare node, (2) reduced resources in heterogeneous clusters (lower CPU/memory/network speed), or (3) placement of multiple MPI tasks on a node if not enough spare nodes are available.

This work contributes *back migration* as a novel methodology in clusters. During a job's execution, MPI tasks record the duration of a timesteps and relay this information to a decentralized scheduler. This scheduler compares the "velocity" of the MPI job *before and after* the migration to decide whether or not to migrate an MPI task back to the original node once this node is brought back online in a healthy state. The decision considers (a) the overhead of back migration and (b) the estimated time for remained part of the job, which is also recorded for the MPI job and communicated between the job and the scheduler. We have implemented the back migration mechanism within LAM/MPI and BLCR based on our work on process-level live migration.

Experiments were conducted on a dedicated Linux cluster comprised of 18 compute nodes. Results were obtained for the NAS parallel benchmarks (NPB), as depicted in Figure 1, where the CPU frequency of the destination node is just half of that on the original node. The condition to benefit from the back migration is:

$$R \times (T_d - T_o) - T_m > 0$$

which means

$$R > T_m/(T_d - T_o)$$

where R is the number of remaining time steps of the benchmark, T_d is the overhead of one time step on the spare/destination node, T_o is the overhead of one time step of the benchmark on the original node, and T_m is the back-migration overhead (assumed to be symmetric to the initial migration overhead to the spare node). For BT, CG, FT, LU and SP class C on 16 nodes, the results in the figure indicate that we can already benefit from back migration if only 0.4-10% of the MPI job execute time remains to be executed. More specifically, a minimum of two time steps for BT and FT, one for CG and LU, and seven for SP are sufficient grounds to justify back migration, or, more generally, when 2.89% on average of execution is still outstanding. Further results assessing the impact of different CPU frequencies and network speeds in heterogeneous clusters are omitted due to space constraints. In general, the larger the amount of outstanding execution, the higher the benefit due to back migration. This illustrates a considerable potential of back migration particularly for large-scale clusters with thousands of nodes, which has not been studied to date.

Acknowledgement: This work was supported in part by NSF grants CCR-0237570 (CAREER), CNS-0410203, CCF-0429653 and DOE DE-FG02-08ER25837. The research at ORNL was supported by Office of Advanced Scientific Computing Research and DOE DE-AC05-00OR22725 with UT-Battelle, LLC.

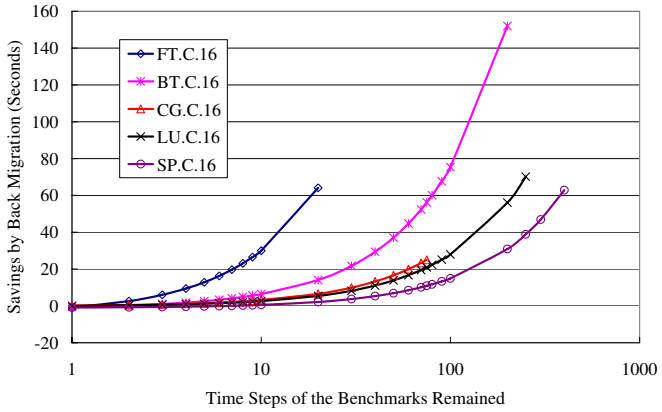

Fig. 1: Savings of Back Migration for NPB Class C on 16 Nodes