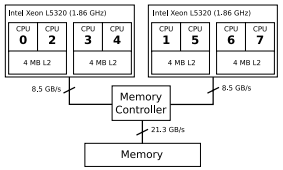
Characterization of Intra-node Topology and Locality

Kevin T. Pedretti
ktpedre@sandia.gov

Introduction

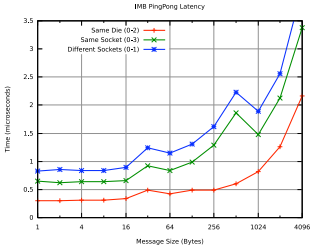


This poster examines the effects of intra-node topology and locality on a recent Intel dual-socket, quad-core compute node (above). As chip-level multiprocessors evolve to incorporate more cores and increasingly complex memory hierarchies, the performance impacts presented will become more pronounced.

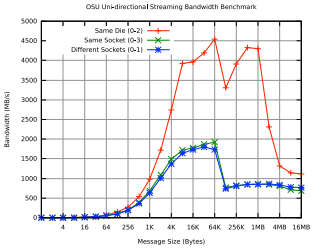
Conclusions:

- The process to processor mapping is found to significantly impact performance on this platform.
- There is no one-size-fits-all solution. The best mapping will vary depending on application characteristics.
- This provides motivation for exploring OS-level heuristic techniques for automatic exploitation of intra-node locality as well as application-level interfaces for expressing the desired locality at run-time.

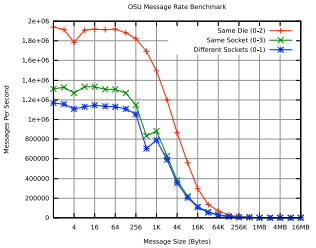
MPI Micro-benchmarks



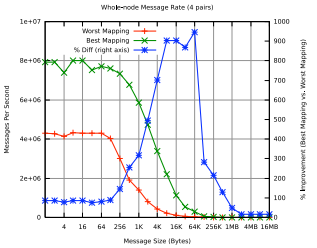
Intra-die communication has the lowest latency due to the shared L2 cache. Off-die communication requires the involvement of the memory controller, adding overhead.



Intra-die message transfers that fit in the shared L2 cache are very efficient. Off-die communication requires cache line movement, adding overhead.



Messaging rate is important for emerging programming models that tend to send many small messages. Cores that share a cache obtain significantly higher rates.

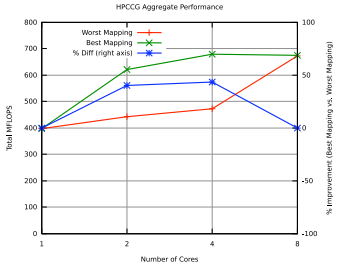


The best mapping places communicating processes on the same die. The worst mapping places them on different sockets, leading to considerable contention.

HPCCG Mini-application

The HPCCG mini-application is a simple conjugate gradient solver that represents an important workload for Sandia. The majority of its runtime is spent performing a sparse matrix vector multiply, which is a very memory bandwidth intensive operation. A fixed-size problem per core was used to obtain these results (approximately 130 MB per core).

CPUs	MFLOPS	%Difference
0 & 1	621	–
0 & 2	442	-41%
0 & 3	472	-24%
0 & 4	472	-24%
0 & 5	621	–
0 & 6	621	–
0 & 7	621	–



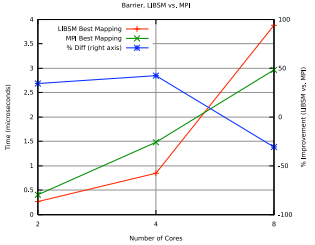
When running on two of the eight cores (table on left), lower performance is obtained when using cores in the same socket due to contention for the memory controller link and reduced cache availability. Before running this experiment, a more logical numbering of CPUs was expected. The actual numbering is shown in the system block diagram above.

For this platform, HPCCG can only efficiently utilize four cores. Using all eight cores does not increase performance and wastes power. However, it is important to use the right four cores. The graph on the left shows that there is almost a 50% performance difference between the best and worst mapping of MPI processes to cores. The best mapping distributes MPI processes one per die. The worst mapping places all MPI processes on the same socket.

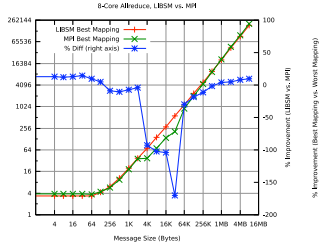
LIBSM Library

LIBSM allows MPI applications to take advantage of intra-node shared memory pools and collective operations. It was developed by Sandia in 2001 for distributed memory machines with SMP nodes. It has been ported to x86/Linux for this work in order to evaluate its performance on a modern platform.

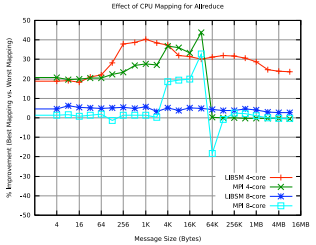
LIBSM performs intra-node barriers via atomic loads and stores to shared memory. Contention becomes a problem at eight cores.



Small intra-node Allreduces have slightly lower latency with LIBSM. MPI likely uses a better algorithm for mid-range messages.



Topology mapping effects are noticeable for both LIBSM and MPI, especially when using four cores.



National Nuclear Security Administration

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories