The SOR as a Benchmarking Tool for Single- Multi- and Many-core Systems

Anne C. Elster [1]
Department of Computer and Information Science
Norwegian University of Science and Technology (NTNU), Trondheim, Norway
elster@idi.ntnu.no

Dedicated to the memory and work of Prof. David M. Young

Analyzing and bechmarking modern computing systems with several levels of cache, multiple cores, many cores such as GPUs, and distributed memory is becoming increasingly challenging. In this paper, we show how a flexible implementation of the SOR method as can be used as a benchmarking tools for such systems. The SOR method for solving linear equations, which Dr. David M. Young introduced in his doctoral work at Harvard in 1950, has had a major impact on many scientific fields. With red-black ordering the method parallelizes well on both vector and disptributed memory computing systems leding to renewed interest in it since the 1980s.

As processors become faster, the impact of memory latency and bandwidth on performance is increasing. To combat this problem, processors are continuously getting larger and more aggressive caches. If the problem does not fit in cache, blocking is needed (e.g. row bands). For the multi-CPU case, it pays off to parallelize across the rows. By having each CPU start in the middle of its assigned domain and spread out toward the edges in both directions, you effectively end up with two bands per CPU. The computation on each CPU would be the same as for the multicore single-CPU, parallelizing across the columns for each core

We have previouly shown (Elster & Holtet (2002)) that increasing the boarder sizes and doing redundant computations is an effective techique to overcome the latency factor on SMP clusters with dedicated communications links. By computing not only the border cells, but a border n cells wide, we can allow the neighbor node to compute n iterations before a new exchange is necessary. While this does increase bandwidth cost by a factor n, this will give a good speedup if latency dominates the bandwidth. I.e. if one benefits from wide boarders, one is hiding a lot of latency.

With Spampinato (Parco2009) we tested the NVIDIA s1070 sytem with 4 GPus each with 240 cores, Here, boarder bands of 1 were sufficient for good SOR performanance, but varying the dimensions up to 11GB, the application became I/O bound.

The SOR's stencil based structure can be found in many parallel applications, thus testing a system with a flexible test-implementation of a paralellized SOR, will uncover many of the system parameters one can expect will affect performance. Note also that the optimization techniques developed in our test suite can be applied to several applications with similar structure.

[1] This presentation represents joint work with Thorvald Natvig as well as Robin Holtet and Daniele Spampinato