Improving the Performance of Parallel SpMV Operations on NUMA Systems with Adaptive Load Balancing

Christian NEUGEBAUER [a] Rudolf BERRENDORF [a,1], Florian MANNUSS [b]

[a] *Bonn-Rhein-Sieg University of Applied Sciences, Sankt Augustin, Germany*

[b] *Saudi Arabian Oil Company, Dharan, Saudi Arabia*

Abstract. For a parallel Sparse Matrix Vector Multiply (SpMV) on a multiprocessor, rather simple and efficient work distributions often produce good results. In cases where this is not true, adaptive load balancing can improve the balance and performance. This paper introduces a low overhead framework for adaptive load balancing of parallel SpMV operations. It uses statistical filters to gather relevant runtime performance data and detects an imbalance situation. Three different algorithms were compared that adaptively balance the load with high quality and low overhead. Results show that for sparse matrices, where the adaptive load balancing was enabled, an average speedup of 1.15 (regarding the total execution time) could be achieved with our best algorithm over 4 different matrix formats and two different NUMA systems.

Keywords. SpMV, adaptive load balancing, NUMA, Chain-on-Chain Partitioning (CCP)

1. Introduction

The multiplication of a sparse matrix with a dense vector ($y \leftarrow Ax$) is an important operation that is used repeatedly in iterative solvers and contributes to a large amount of the runtime in many applications in natural science and engineering. Much effort has been spent in the past optimizing this operation, e.g., [7,1,13,9]. To do so, several parameters have to be considered simultaneously, among them the utilization of the matrix' nonzero structure, the target processor architecture, and the storage format for the sparse matrix. One additional important point to consider for a parallel implementation is the load balance, which is again influenced by the parameters mentioned above and others, too. To find a static load distribution, usually after the matrix structure is known, a certain cost model is used and then a partitioning algorithm is applied once. Often the found partition delivers a good load balance, but sometimes (e.g. the cost model does not fit reality) a load imbalance appears. As the SpMV might be iterated

[1] Corresponding Author: Rudolf Berrendorf, Bonn-Rhein-Sieg University of Applied Sciences, Grantham-Allee 20, 53757 Sankt Augustin, Germany; E-mail: rudolf.berrendorf@h-brs.de.