Latency hiding of global reductions in pipelined Krylov methods

Bram REPS, University of Antwerp, Belgium

Wim VANROOSE, University of Antwerp, Belgium

Pieter GHYSELS, Lawrence Berkeley National Lab, USA

Many existing mathematical solver libraries were originally not designed with large-scale parallel use in mind and therefore require a reorganization of the underlying algorithms or even a totally new approach. Numerous computational methods spend most of their compute time in linear algebra operations e.g. to compute the solution of large (sparse) linear systems,

$$Ax = b, \tag{1}$$

with matrix $A \in \mathbb{R}^{n\times n}$ and $x, b \in \mathbb{R}^n$. In particular, Krylov subspace methods are a family of solvers that iteratively improve an approximate solution of (1). Starting from an initial guess x_0 and with $r_0 = b - Ax_0$ being the initial residual the i-th approximation x_i is searched in the space $x_0 + \mathcal{K}_i(A, r_0)$, where $\mathcal{K}_i(A, r_0) = \text{span}\left\{r_0, Ar_0, A^2r_0, \ldots, A^{i-1}\right\} \subseteq \mathbb{R}^n$ is called the Krylov subspace that extends in each iteration. In these methods, two key computations bring along the main communication bottlenecks that limit the performance for sparse linear algebra: sparse matrix-vector multiplication (SpMVM), defined by matrix A, and vector-vector multiplications or dot-products.

First, the performance of the SpMVM is mainly hampered by the limited memory bandwidth on modern CPUs. Indeed, it is an operation where the ratio of useful calculations per bytes read from slow memory is low, i.e. with low arithmetic intensity, so that the performance is solely determined by the memory bandwidth.

Second, the latency of the global reductions limits the performance of the dot-products that are necessary for the projections in Krylov methods. A dot-product involves, after the local dot-product on each core, a reduction tree that sums up the contributions from all the cores. The result is then broadcast to all the cores. The communication cost is directly related to the depth of this reduction tree that grows logarithmically as a function of the number of cores. Furthermore, this global reduction operation requires the synchronization of all participating cores and is thus extremely sensitive to the variability in the system.

In this talk we present a way to hide communication in Krylov methods. In so-called *pipelined CG* and *pipelined GMRES* the latency issue of the global reduction operators is circumvented by overlapping time-consuming communication phases with computation steps for the SpMVM and the application of a preconditioner [1, 2]. This approach is attractive since it allows the use of standard preconditioners, as opposed to s-steps methods that require specialized preconditioners [4]. In addition, we cope with the limited memory bandwidth by increasing the arithmetic intensity of the SpMVM in a multigrid method, a method widely used as preconditioner, by vectorizing the stencil computations [3].

Références

[1] P. Ghysels, T.J. Ashby, K. Meerbergen, W. Vanroose, *Hiding Global Communication Latency in the GMRES Algorithm on Massively Parallel Machines*, SIAM J. Sci. Comput., 35, 2013.

[2] P. Ghysels, W. Vanroose, *Hiding global synchronization latency in the preconditioned Conjugate Gradient algorithm*, Parallel Computing, 2013.

[3] P. Ghysels, P. Klosiewicz, W. Vanroose, *Improving the arithmetic intensity of multigrid with the help of polynomial smoothers*, Num. Linear Algebra Appl., 19, 2012.

[4] L. Grigori, S. Moufawad, *Communication Avoiding ILU0 Preconditioner*, Rapport de recherche RR-8266, INRIA, 2013.

Bram REPS, University of Antwerp, 1 Middelheimlaan, B-2020 Antwerpen, Belgium
bram.reps@uantwerpen.be
Wim VANROOSE, University of Antwerp, 1 Middelheimlaan, B-2020 Antwerpen, Belgium
wim.vanroose@uantwerpen.be
Pieter GHYSELS, Lawrence Berkeley National Lab, 1 Cyclotron Road, CA 94720-8150 Berkeley, USA
pghysels@lbl.gov