Non-blocking Preconditioned Conjugate Gradient Methods for Extreme-scale Computing

Paul R. Eller¹, William Gropp (Advisor)¹
¹University of Illinois at Urbana-Champaign

Abstract
To achieve the best performance on extreme-scale systems we need to develop more scalable method variations. For PCG, dot products limit scalability because they are a synchronization point. Non-blocking methods provide potential to hide most of this cost of the allreduce and avoid synchronization cost due to performance variation across cores.

Preconditioned Conjugate Gradient Method (PCG)
- Iterative algorithm for solving large sparse systems of linear equations.
- Preconditioners accelerate convergence.
- Can rearrange PCG to:
  • Reduce communication latency using a single allreduce (L56PCG, PIPECG).
  • Overlap communication and computation using non-blocking allreduces (NBPCG, PIPECG).
- Optimizations introduce vector operations and initialization costs.

Scalable Conjugate Gradient Methods
- L56PCG
- NBPCG
- PIPECG

Non-blocking Allreduce
- Each method equivalent to PCG in exact arithmetic.
- Implemented custom solvers in PETSc.

Merged Vector Operations
- Merging vector operations avoids cost of extra vector reads.
- Rearranged methods still require additional writes to memory.
- Requires additional computations, but these are cheap compared to memory accesses.
- PCG and L56PCG read some vectors from cache when multiple vectors fit in cache.

Performance Modeling
- Determine memory access and compute costs with modified STREAM benchmark.
- Model communication with LogGOPS.
- Compute parameters with Netgauge.
- Analyze performance using strong scaling tests.

General Observations
- Non-blocking methods should perform better than blocking methods as the vector operations cost decreases and allreduce cost increases.
- Non-blocking methods should perform well while the MatVec and/or PC have enough computation to hide allreduce cost.
- NBPCG initially should outperform PIPECG due to lower vector operations cost.
- PIPECG should scale better due to overlapping cost of allreduce with computation of both the MatVec and PC.

Conclusions
- Non-blocking solvers provide potential to improve performance at scale due to hiding cost of allreduce and avoiding synchronization.
- Current implementations cannot yet outperform standard PCG.
- Performance models show potential for NBPCG and PIPECG to be more scalable than PCG.
- Ability to minimize impact of noise may be key benefit.

Impact of Noise
Noise throughout PCG limits performance by causing all processes to wait for slowest process at sync points.
- Computational noise sources: Operating system processes, error correction, etc.
- Communication noise sources: Contention in network, varying distances between nodes, varying size/number of messages, etc.
- Performance models predict ability to minimize impact of noise may be a key advantage to non-blocking solvers.

Solver Performance Results
- Ran tests on Blue Waters on up to 32k cores using a 5-point 1-billion row Poisson matrix and block Jacobi ILU preconditioner.
- Implementations reference at Sandia as SAND2015 6374C.