A Standard for Batched BLAS Routines

ABSTRACT
We propose an API for Batched Basic Linear Algebra Subprograms (Batched BLAS). We focus on multiple independent BLAS operations on small matrices that are grouped together as a single routine. We aim to provide a more efficient and portable library for multi/manycore HPC systems. We achieve 2x speedups and 3x better energy efficiency compared to vendor implementations. We also demonstrate the need for Batched BLAS and its potential impact in multiple application domains.

DEFINITION
Batched BLAS: multiple independent BLAS operations on small matrices grouped together as a single routine

PROPOSED SPECIFICATION
The function arguments are reminiscent of the BLAS standard.

Batched Level 3
BLAS Dgemm
Calling Sequence

dgemm_batch( enum *transa, enum *transb, integer *m, integer *n, integer *k, double *alpha, double **arrayA, integer *lda, double **arrayB, integer *ldb, double **arrayC, integer *ldc, integer *batch_count, enum *batch_opts, integer *info);

Batched Level 2
BLAS Dgemv
Calling Sequence

dgemv_batch( enum *trans, integer *n, integer *m, double *alpha, double **arrayA, integer *lda, double **arrayB, integer *ldb, double **arrayC, integer *ldc, integer *incx, integer *batch_count, enum *batch_opts, integer *info);

Batched Level 1
BLAS Daxpy
Calling Sequence

daxpy_batch( integer *n, double *alpha, double **x, integer *incx, double **y, integer *incy, integer *batch_count, enum *batch_opts, integer *info);

BATCHED BLAS PERFORMANCE

APPLICATIONS OF BATCHED BLAS

Batched LAPACK Dgetrf Example

Nuclear network simulation (XNet benchmark)
- 150 x 150 matrices
- batch_count = 300+
- Titan Supercomputer at ORNL
- 3x faster than MKL
- 2x faster than MA48 factorization (HSL)

ASTROPHYSICAL THERMONUCLEAR NETWORKS COUPLED TO HYDRODYNAMICAL SIMULATIONS IN EXPLOSIVE BURNING SCENARIOS
- 7x faster using Batched BLAS
- batch_count = 10 to 800+

TECHNOLOGIES
Some of the technologies we may wish to utilize include:

OpenMP
- Multicore
- Accelerators

CUDA
- Fused Kernels
- Multiple Streams

OpenCL

ADVANTAGES

- More efficient and portable implementations
- HPC Numerical library for modern architectures
- Better hardware utilization and energy efficiency
- Encourages, as well as simplifies, community efforts to build higher-level algorithms on top of Batched BLAS

REFERENCES


ACKNOWLEDGMENTS

This material is based upon work supported in part by the European Union’s Horizon 2020 research and innovation programme under the NLA4ET grant agreement No 671633, the U.S. National Science Foundation under Grants No. CSR 1514266 and ACI-1339822, NVIDIA, an INRIA project, and the U.S. Department of Energy. The specification is open for community discussion; we would welcome your comments: info@nlafet.eu