NLAFET
Parallel Numerical Linear Algebra for Future Extreme Scale Systems

Umeå University · INRIA · Rutherford Appleton Laboratory · University of Manchester · University of Tennessee

A Standard for Batched BLAS Routines

ABSTRACT

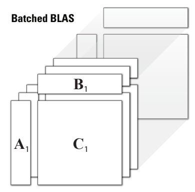
We propose an API for Batched Basic Linear Algebra Subprograms (Batched BLAS). We focus on multiple independent BLAS operations on small matrices that are grouped together as a single routine. We aim to provide a more efficient and portable library for multi/manycore HPC systems. We achieve 2x speedups 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.

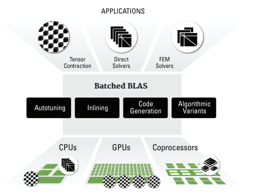
DEFINITION

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

Numerous applications require Batched BLAS:

- Structural mechanics
- Astrophysics
- Direct sparse solvers
- High-order FEM simulations





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   *beta,
              double   **arrayC,
              integer  *ldc,
              integer  batch_count,
              enum     batch_opts,
              integer  *info);
```

Batched Level 2 BLAS DGEMV Calling Sequence

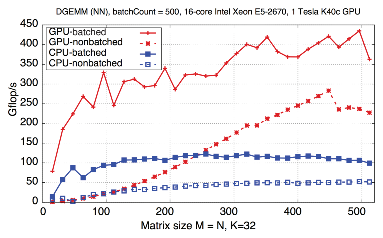
```
dgemv_batch(  enum     *trans,
              integer  *m,
              integer  *n,
              double   *alpha,
              double   **arrayA,
              integer  *lda,
              double   **x,
              integer  *incx,
              double   *beta,
              double   **y,
              integer  *incy,
              integer  batch_count,
              enum     batch_opts,
              integer  *info);
```

Batched Level 1 BLAS DAXPY Calling Sequence

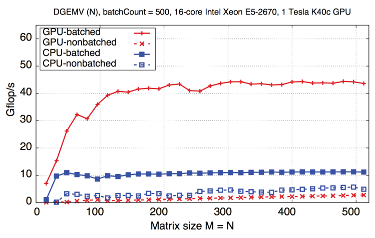
```
daxpy_batch(  integer  *n,
              double   *alpha,
              double   **x,
              integer  *incx,
              double   **y,
              integer  *incy,
              integer  batch_count,
              enum     batch_opts,
              integer  *info);
```

BATCHED BLAS PERFORMANCE

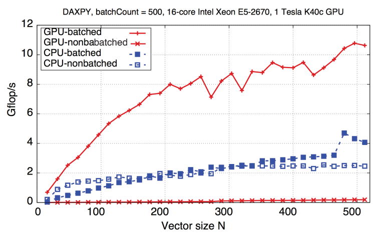
Batched Level 3 BLAS DGEMM Example



Batched Level 2 BLAS DGEMV Example

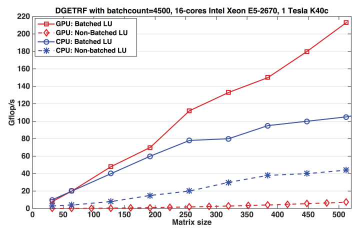


Batched DAXPY Example



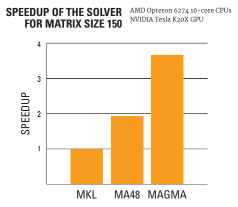
APPLICATIONS OF BATCHED BLAS

Batched LAPACK DGETRF Example



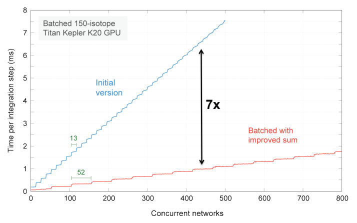
Nuclear network simulation (XNet benchmark)

- 150 x 150 matrices
- batch_count = 100+
- 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

The specification is open for community discussion; we would welcome your comments: **info@nlafet.eu**

REFERENCES

[1] A. Haidar, T. Dong, S. Tomov, P. Luszczek, and J. Dongarra, Framework for Batched and GPU- resident Factorization Algorithms Applied to Block Householder Transformations, ISC HPC, Springer LNCS, Frankfurt, Germany, July 12-16, 2015.

[2] A. Haidar, S. Tomov, P. Luszczek, and J. Dongarra, MAGMA Embedded: Towards a Dense Linear Algebra Library for Energy Efficient Extreme Computing, 19th IEEE High Performance Extreme Computing Conference (HPEC 2015), Best Paper Award, IEEE, Waltham, MA, September, 2015.

[3] A. Haidar, T. Dong, P. Luszczek, S. Tomov, and J. Dongarra. Batched matrix computations on hardware accelerators based on GPUs. International Journal of High Performance Computing Applications, first published on February 9, 2015 as doi:10.1177/1094342014567546.

[4] J. Dongarra, I. Duff, M. Gates, A. Haidar, S. Hammarling, N. Higham, J. Hogg, P. Lara, M. Zounon, S. Relton, and S. Tomov, A Proposed API for Batched Basic Linear Algebra Subprograms, UTK Computer Science Technical Report, (available at https://bit.ly/batched-blas), March 2016.

ACKNOWLEDGMENTS

This material is based upon work supported in part by the European Union's Horizon 2020 research and innovation programme under the NLAFET grant agreement No 671633, the U.S. National Science Foundation under Grants No. CSR 1514286 and ACI-1339822, NVIDIA, and the U.S. Department of Energy.