H2020-FETHPC-2014: GA 671633

NLAFET Working Note 4

Workshop on Batched, Reproducible, and Reduced Precision BLAS

Sven Hammarling

July, 2016

Document information

This preprint report is also published as MIMS EPrint 2016.41, Manchester Institute for Mathematical Sciences School of Mathematics, The University of Manchester.

Acknowledgements

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under the grant agreement number 671633.

Workshop on Batched, Reproducible, and Reduced Precision BLAS.

Innovative Computing Laboratory
University of Tennessee
May 18th - 19th, 2016

Workshop website: http://bit.ly/Batch-BLAS-2016

Sven Hammarling

School of Mathematics
The University of Manchester
Alan Turing Building
Manchester, M13 9PL, UK

sven@ma.man.ac.uk

July 15, 2016

1 Introduction

The principal purpose of the workshop was to present and discuss proposals for a set of batched BLAS, a set of reproducible BLAS and a set of reduced precision BLAS. The aim is to define a standard interface for each of these sets of BLAS.

The event also included a reception on the evening of the 17th May and a dinner on the 18th May, both kindly sponsored by Intel.

The workshop was opened by Jack Dongarra with a welcome and an introduction to the workshop, followed by each of the participants introducing themselves.

Links to the talks can be found in the above workshop webpage, which also has links to further information on the Batched BLAS and the Reproducible BLAS.

This report mainly highlights, briefly, those parts of the presentations that are immediately relevant to the purpose and aims of the workshop. A number of other interesting and important topics were raised, particularly in relation to reproducibility and replicability; please see the presentations for the detail.

In the main and for the sake of brevity, only the discussions related to the proposals have been included.

2 The Batched BLAS

There were nine presentations related to the Batched BLAS. In particular, there were presentations discussing proposed specifications for a set of Batched BLAS routines.

The Batched BLAS are intended for HPC applications where a large number of the same small matrix operations, such as matrix multiplication, can be solved simultaneously.

2.1 Overview of the Draft for Batched BLAS

Stan Tomov, ICL, UTK

This presentation was concerned with the need and motivation for the Batched BLAS, gave the specification of the proposed calling sequence and examples of their use.

There was considerable discussion about the proposed API. Some of the issues raised included:

- the aliasing rules.
- Concerns about the BATCH_VARIABLE option for *batch_opts*, such as when to use that rather than several calls with BATCH_FIXED, the amount of data needed when *batchcount* is large.
- The effect of a large value of *batchcount*.
- The number of cases that need to be implemented.

Other discussion centred on the notion of locality and if some utilities, such as the handle in CUBLAS, could be provided to help with the issues.

2.2 Reference Implementation and Testing of Batched BLAS Routines

Mawussi Zounon, University of Manchester

This presentation described the reference implementation, accuracy testing and experimental results.

The speaker felt that the accuracy bound for batched DGEMM was too pessimistic for large problems, but Jim Demmel felt that one should just declare victory and not worry about it!

In the discussion concern was expressed at the use of the XERBLAS, being the same as the BLAS and LAPACK, but that was allayed when it was pointed out that it is actually XERBLA_BATCH. The question was raised as to whether, or not, we want batching for reproducible BLAS and reduced precision BLAS as well?

2.3 Example of Cholesky's Efficient Implementations

Jacob Kurzak. ICL, UTK

This presentation gave a detailed description of implementing Cholesky and batched Cholesky, also referring to the use of the Bench-testing Environment for Automated Software Tuning, BEAST[1].

2.4 High-Performance Batched Computations

Azzam Haidar. ICL, UTK

This presentation concentrated on the batched computations within MAGMA[2] on CPU, MIC (Many Integrated Core architecture) and, especially, GPU, showing, for example, how to speed up classical *LU* when using the Batched BLAS.

2.5 Towards Variable Size Batch BLAS (and LAPACK): A Sneak Peak into MAGMA's vbatched Routines

Ahmad Abdelfattah, ICL, UTK

This presentation concentrated on the need for and status of variable size Batched BLAS routines, particularly in relation to MAGMA. The variable size routines allow for the same operation to be performed on many small matrices of varying sizes. Performance gains were shown for matrix-matrix and matrix-vector multiply, triangular solve and Cholesky.

[1] icl.utk.edu/beast/

[2] The ICL dense linear algebra library for GPU and multicore architectures. icl.cs.utk.edu/magma/

2.6 Batched Routines in Preconditioning: The Future of Incomplete Factorization Preconditioners

Hartwig Anzt, ICL, UTK

This presentation showed the need for batch triangular solve, in the use of incomplete factorizations, for the solution of sparse matrix problems. The need for batch gather and scatter routines was also highlighted.

2.7 Do we need to support alternative data formats for Batched BLAS?

Jonathan Hogg, STFC, RAL

This talk presented the need for an interleaving option in the Batched BLAS, motivated by their use in the sparse Cholesky factorization. In this option the elements of the matrices to be operated upon would be interleaved in memory. For efficiency the columns would need to be aligned, so an alignment flag was proposed.

The possibility of a set of microBLAS was also considered and there was some support in the discussion here, and following other presentations.

2.8 Intel MKL® GEMM_BATCH

Sarah Knepper, Intel

This talk presented the current state of the Intel MKL Batched BLAS, looking particularly at the batched GEMM routine. The Intel version has arguments for grouping the same sized matrices together. A very helpful table indicating the current interfaces for different implementations of batched GEMM was given.

It seems fair to say to there was a mixed reception to the grouping, with concerns over efficiency, but also support for including grouping as an option. It was pointed out that the fixed size case can be handled by having the group count equal to one and the group size equal to the batch count. There was further discussion on optional arguments (handles). There was a request for both row and column strides (tda as well as lda).

2.9 Batched Linear Algebra Operations: BLAS for Many Very Small Problems

Hatem Ltaief, KAUST

This talk concerned a KAUST project, HiCMA: Hierarchical Computations on Manycore Architectures, particularly the use of recursive BLAS, the GEMM based BLAS, batched BLAS, and hierarchical BLAS for $\mathcal{H}$–matrix computations motivated by the European Extremely Large Telescope project.

3 Reproducibility

Software reproducibility concerns getting bitwise identical results from multiple runs of a program with the same input. The first two talks in this section discussed software reproducibility, the third presentation also talked about replicability of research results.

3.1 Reproducible BLAS

Jim Demmel, UCB

This talk presented motivation for a set of reproducible BLAS (ReproBLAS), gave seven design goals for a reproducible sum, and discussed each of the design goals. The current state of the ReproBLAS was presented, including suggested naming scheme and interface[3]. The talk finished with a set of issues for discussion.

It was suggested that atomics may be appropriate, although others felt that this may not be possible, or at least needs care.

3.2 Reproducibility of Computations and Distributed Data Structures

William Gropp, University of Illinois Urbana-Champaign

This talk was concerned with reproducibility, not accuracy, of computational results from complete programs, not just from lower level functions. The talk opened with some computational reproducibility issues, went on to discuss approaches for obtaining reproducibility and showed that data distributions are critical.

3.3 Reproducibility & Replication: Tools, Policies and Processes

Michael Heroux, Sandia National Laboratories

This talk covered containers, in particular the Trilinos docker project[4], ACM replicated computational results (RCR) and the DOE software productivity and sustainability plan; the last two of these being not just concerned with software, but with replicability of research results. The presenter is editor in chief for ACM TOMS and outlined the TOMS RCR initiative.

The feeling in the meeting seemed to be very positive.

4 BLAS for Different Precisions

The two talks in this section discussed BLAS for both reduced and extended precision.

[3] See: bebop.cs.berkeley.edu/reproblas
[4] hub.docker.com/r/sjdeal/webtrilinos/

4.1 BLAS Interface for Different Precisions

Jack Dongarra, ICL, UTK

This talk first reminded the audience of the discussions at the BLAS Technical Forum[5], mentioned the Nvidia 16 bit BLAS, added to support half-precision floating point, and suggested a naming scheme for various precision BLAS.

4.2 XBLAS and More

Greg Henry, Intel

This presentation first looked at the extended precision BLAS (XBLAS), reminding the audience of the functionality of the current XBLAS and noting that that the XBLAS has 980 routines, followed by a discussion of what is needed to make the XBLAS relevant to today's architectures, and a proposal for a slimmed down set of XBLAS. The presenter also gave further details of the Intel MKL Batched BLAS, including an application example.

There was some discussion of Batched BLAS versus OpenMP.

5 Related Presentations

There were four talks in this section presenting work relevant to the goals of the workshop.

5.1 Bench-testing Environment for Automated Software Tuning (BEAST): Programming Autotuners

Piotr Luszczek, ICL, UTK

This talk introduced the BEAST[6] tool for autotuning, giving the motivation for BEAST, examples of use and performance and future work.

5.2 Parallel Numerical Linear Algebra for Future Extreme-Scale Systems

Bo Kågström,Umeå University

This presentation described the project NLAFET[7] funded under the European Commission's program Horizon 2020. The goals of the project have much in common with those of this workshop.

5.3 Communication avoiding algorithms for iterative methods

Laura Grigori, Inria

[5] www.netlib.org/blas/blast-forum/

[6] See the reference in Section 2.3

[7] Numerical Linear Algebra for Future and Emerging Technologies. www.nlafet.eu/

This was a follow on presentation to the previous presentation, describing a particular topic being investigated under the NLAFET project. The computational approach, an example application and performance results were presented.

5.4 Towards ATLAS 4.0

Clint Whaley, Louisiana State University

ATLAS[8], which stands for Automatically Tuned Linear Algebra Software, is a long standing project which includes the BLAS. This presentation described the work being done towards the next release of the software.

6 Vendor Presentations

In this session six vendors gave presentations about their software.

6.1 Faster Code.... Faster: Intel® Math Kernel Library aka *Intel® MKL*

Shane Story, Intel

In this talk the presenter reminisced about the BLAST Technical Forum and discussed the Intel Math Kernel Library, including a description of the option of Conditional Numerical Reproducibility (CNR). It was also mentioned that they are getting requests for an integer version of GEMM (IGEMM).

6.2 ARM Performance Libraries

Chris Goodyer, ARM

This talk introduced ARM, its performance libraries and issues related to the workshop.

6.3 NAG and the BLAS

Mick Pont, NAG Ltd

This talk introduced NAG, the NAG Library and its relationship to the BLAS, as well as a discussion of reproducibility. In discussing new functionality in the Library, a request was made for some additional BLAS functionality, particularly the multiplication of triangular matrices, which arise in computing matrix functions.

6.4 Batched, Reproducible and Reduced Precision BLAS – Some Thoughts

Bobby Cheng, MathWorks

[8] math-atlas.sourceforge.net/

This presentation described the relationship between the BLAS and MATLAB, the importance of reproducibility and how MATLAB handles it, and expressed the view that the topics of this workshop are very important to The MathWorks.

6.5 CUBLAS

Sharan Chetlur, Nvidia

This talk described the Nvidia CUBLAS, including the Batched routines, reduced and mixed precision routines, and reproducibilty of the CUBLAS. The new interface using the "stride" parameter (in the current CUBLAS) to move between the different matrices in the batch easily was also mentioned.

6.6 Cray Scientific and Math Libraries

Aaron Collier, Cray

This talk outlined the contents of the CSML Libraries.

7 Wrap Up

Jack Dongarra, ICL, UTK

Jack Dongarra closed the workshop by briefly summarising the outcomes, suggesting what comes next and showing the group photo and a Gary Larson style cartoon! Jack said that it is intended to have a report on the workshop (this report) and a follow on meeting.

The meeting expressed their grateful thanks to Jack and his colleagues Leighanne, Teresa and Tracy for an excellent workshop.

8 Brief Summary

The stated purpose of the workshop was to look into defining a standard interface for the Batched BLAS, Reproducible BLAS, and Reduced Precision BLAS. In addition a proposal was made for revising the mixed (extended) precision BLAS (XBLAS), With that in mind, proposals were presented for:

- A set of Batched BLAS. The proposed functionality is along the lines of the standard BLAS. The proposals mainly differed in the options for batch types (fixed, variable, group).
- A set of Reproducible BLAS (ReproBLAS). The current state of the ReproBLAS can be found at bebop.cs.berkeley.edu/reproblas.

- A set of reduced precision BLAS. Whilst the motivation for the proposal was 16 bit (half precision) arithmetic, the proposed naming scheme allows for the possibility of other precisions, such as quad precision.
- A revised set of XBLAS.

As a reminder, the presentations can be found at the workshop webpage bit.ly/Batch-BLAS-2016, as well as links to further information on the Batched BLAS and ReproBLAS.

Comments on the proposals would be welcome. Please send these, and questions about the proposals presented in this report to Jack Dongarra at dongarra@icl.utk.edu.