LITERATURE REVIEW: Optimizing CRVE with batched matrix algebra on GPUs

Scott McNeil
Department of Economics
Carleton University
Ottawa, Canada K1S 5B6
scott.mcneil@carleton.ca

October 11, 2016

1 Introduction

As a common task in science and statistics, considerable work has been put into optimizing GEneral Matrix Multiplication (GEMM), particularly via Graphical Processing Units (GPU). Because of data transfer overhead, most linear algebera algorithms for GPUs only offer an advantages when matrices are large. However, batched proccessing of matrices can provide speed improvements for large groups of small matrix multiplciations[18].

Cluster robust variance estimation (CRVE) is a statisitcal technique for for correcting inference when data appears in distinct groups–such as time period, industry or even high school classroom . The CRVE is an important–and underused–statisitical technique for econometrics [19]. However, it can be computationally expensive for three reasons. First, calculation of the CRVE requires a set of non-trivial matrix multiplications and scales poorly with both the number of observations and the number of regressors. Second, CRVE has shown to be best when combined with bootstrap resampling, especially the wild boostrap, which involves recalculating the CRVE hundreds if not thousands of times [6]. Finally, reasoning about the CRVE often involves Monte Carlo simulations, with tens of thousands of replications.

The challenge to optimizing the CRVE on a GU is the matrix multiplications involved are almost always very small. However, it is possible to recombine the matrix calculations and compute them via a batched calculation. This especially provides benefits for when used with a bootstrap.

2 Literature Review

The CRVE is a generalization of the heteroskedacitiy-robust estimators proposed by White [26], with the original CRVE estimator proposed by [12]. The use of the CRVE with bootstrapping was proposed by Cameron, Gelbach and Miller, often referred to the wild cluster bootstrap (WCB) [6]. A good review of the CRVE is also provided by Cameron, Gelbach and Milelr [7].

There are a number of software implementatioins of CRVE. This includes an implementation in the standard distribution of the Stata statistical language [23] and the Sandwich package for the R programming language [27]. Finally, the wild cluster bootstrap is available through boottest package for Stata [24].

There have been a number of attempts to optimize similar econometric techniques. A general review of parallel programming for econometrics is available in [13]. Of particular interest, [17] provides a GPU-optimized version of the block boostrap, which makes use of the sweep operator to overcome limitations. This solution is similar to the problems inherent optimizing the CRVE and WCB, however the the exact implementation is not directly applicable.

There has been considerable work done to optimize dense linear algebra (DLA) algorithms for parallel processing. For a review of this, including batched processes, see [11]. Implementations of batched DLA algorithms are currently availble both in NVIDIA's cuBLAS library [21] and the MAGMA library from University of Tennessee's Innovative Computing Laboratory (ICL) [5]. The thread blocking used in the MAGMA batched routine due to [20]. This paper will rely on the batched routines in the MAGMA library.

A number of applications of batched GEMM implementations have been presented. For example, matrix exponentiation can be formulated as a batched GEMM [16]. High-order finite element methods can also be computed via batched GEMMs using the cuBLAS library, however, this method shows a drop-off in performance when the size of the matrices are not a multiple of 16 [15]. This method can also be reformuated as a tensor contraction [25]. Conversely, the performance of tensor contractions have been shown be improved through batched GEMMs using the MAGMA library [1].

Further, batched GEMMs are incorporated for a number of other batched DLA algorithms, including QR decomposition, Cholsky decomposition and LU factorizations [14] [9] [2] [8] [10]. Batched processing has also been used for vector reduction and vector scaling [22].

The primary implementation this paper will rely on is the batched GEMM algorithms developed by the ICL group for the MAGMA library. In particular, the group has optimized multiplication of squares matrices of size 32 [4]. They showed that tuning of thread block size is an important aspect for batch GEMMs with smalle matrices, which will also be important for optmizing the CRVE. A second papers explores optimizing GEMMs for square matrices as small as size 16, which this paper will also attempt to incorportate [18].

Further work from the ICL group looks at and batches of variable sized matrices [3]. This is not directly relevant since the size of the matrices at each bootstrap replication will neccesarily be indentical.

References

[1] Ahmad Abdelfattah, Marc Baboulin, Veselin Dobrev, Jack Dongarra, Christopher Earl, Joel Falcou, Azzam Haidar, Ian Karlin, Tzanio Kolev, Ian Masliah, et al. High-performance tensor contractions for gpus. *Procedia Computer Science*, 80:108–118, 2016.

[2] Ahmad Abdelfattah, Azzam Haidar, Stanimire Tomov, and Jack Dongarra. Performance tuning and optimization techniques of fixed and variable size batched cholesky factorization on gpus. In *International Conference on Computational Science (ICCS'16)*, San Diego, CA, 06-2016 2015.9.

[3] Ahmad Abdelfattah, Azzam Haidar, Stanimire Tomov, and Jack Dongarra. On the development of variable size batched computation for heterogeneous parallel architectures. In *The 17th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC 2016), IPDPS 2016*, Chicago, IL, 05-2016 2016. IEEE, IEEE.

[4] Ahmad Abdelfattah, Azzam Haidar, Stanimire Tomov, and Jack Dongarra. Performance, design, and autotuning of batched gemm for gpus. In *High Performance Computing: 31st International Conference, ISC High Performance 2016, Frankfurt, Germany, June 19-23, 2016, Proceedings*, volume 9697, page 21. Springer, 2016.

[5] Emmanuel Agullo, Jim Demmel, Jack Dongarra, Bilel Hadri, Jakub Kurzak, Julien Langou, Hatem Ltaief, Piotr Luszczek, and Stanimire Tomov. Numerical linear algebra on emerging architectures: The plasma and magma projects. In *Journal of Physics: Conference Series*, volume 180, page 012037. IOP Publishing, 2009.

[6] A Colin Cameron, Jonah B Gelbach, and Douglas L Miller. Bootstrap-based improvements for inference with clustered errors. *The Review of Economics and Statistics*, 90(3):414–427, 2008.

[7] A Colin Cameron, Douglas L Miller, et al. Robust inference with clustered data. *Handbook of empirical economics and finance*, pages 1–28, 2010.

[8] Tingxing Dong, Azzam Haidar, Piotr Luszczek, James Austin Harris, Stanimire Tomov, and Jack Dongarra. Lu factorization of small matrices: Accelerating batched dgetrf on the gpu. In *16th IEEE International Conference on High Performance Computing and Communications (HPCC)*, Paris, France, 08-2014 2014. IEEE, IEEE.

[9] Tingxing Dong, Azzam Haidar, Piotr Luszczek, Stanimire Tomov, Ahmad Abdelfattah, and Jack Dongarra. Magma batched: A batched blas approach for small matrix factorizations and applications on gpus. Technical report, 08/2016 2016.

[10] Tingxing Dong, Azzam Haidar, Stanimire Tomov, and Jack Dongarra. A fast batched cholesky factorization on a gpu. In *International Conference on Parallel Processing (ICPP-2014)*, Minneapolis, MN, 09-2014 2014.

[11] Jack Dongarra, M Abalenkovs, A Abdelfattah, M Gates, A Haidar, J Kurzak, P Luszczek, S Tomov, I Yamazaki, and A YarKhan. Parallel programming models for dense linear algebra on heterogeneous systems. *Supercomputing frontiers and innovations*, 2(4):67–86, 2016.

[12] Kenneth A Froot. Consistent covariance matrix estimation with cross-sectional dependence and heteroskedasticity in financial data. *Journal of Financial and Quantitative Analysis*, 24(03):333–355, 1989.

[13] Guangbao Guo. Parallel statistical computing for statistical inference. *Journal of Statistical Theory and Practice*, 6(3):536–565, 2012.

[14] Azzam Haidar, Tingxing Dong, Piotr Luszczek, Stanimire Tomov, and Jack Dongarra. Batched matrix computations on hardware accelerators based on gpus. *International Journal of High Performance Computing Applications*, 29:2, 2015.

[15] Chetan Jhurani and Paul Mullowney. A gemm interface and implementation on nvidia gpus for multiple small matrices. *Journal of Parallel and Distributed Computing*, 75:133–140, 2015.

[16] M Graham Lopez and Mitchel D Horton. Batch matrix exponentiation. In *Numerical Computations with GPUs*, pages 45–67. Springer, 2014.

[17] Javier López-de Lacalle. Gpu parallel implementation of numerical distribution functions for seasonal unit root tests. 2016.

[18] Ian Masliah, Ahmad Abdelfattah, A Haidar, S Tomov, Marc Baboulin, J Falcou, and Jack Dongarra. High-performance matrix-matrix multiplications of very small matrices. In *European Conference on Parallel Processing*, pages 659–671. Springer, 2016.

[19] Brent R Moulton. An illustration of a pitfall in estimating the effects of aggregate variables on micro units. *The Review of Economics and Statistics*, 72(2):334–338, 1990.

[20] Rajib Nath, Stanimire Tomov, and Jack Dongarra. An improved magma gemm for fermi graphics processing units. *International Journal of High Performance Computing Applications*, 24(4):511–515, 2010.

[21] CUDA NVIDIA. Basic linear algebra subroutines (cublas) library, 2013.

[22] Lukas Polok and Pavel Smrz. Fast linear algebra on gpu. In *High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on*, pages 439–444. IEEE, 2012.

[23] William Rogers. Regression standard errors in clustered samples. *Stata technical bulletin*, 3(13), 1994.

[24] David Roodman. BOOTTEST: Stata module to provide fast execution of the wild bootstrap with null imposed. Statistical Software Components, Boston College Department of Economics, December 2015.

[25] Yang Shi, UN Niranjan, Animashree Anandkumar, and Cris Cecka. Tensor contractions with extended blas kernels on cpu and gpu. *arXiv preprint arXiv:1606.05696*, 2016.

[26] Halbert White. A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity. *Econometrica*, 48(4):817–38, May 1980.

[27] Achim Zeileis. Econometric computing with hc and hac covariance matrix estimators. *Journal of Statistical Software*, 11(10):1–17, 2004.