

High Performance Pseudo-analytical Simulation of Multi-Object Adaptive Optics over Multi-GPU Systems

Ahmad Abdelfattah¹, Eric Gendron², Damien Gratadour², David Keyes¹, Hatem Ltaief¹, Arnaud Sevin², and Fabrice Vidal²

¹ Extreme Computing Research Center, Division of Computer, Electrical, and Mathematical Sciences and Engineering, KAUST, Thuwal, KSA

{Ahmad.Ahmad,David.Keyes,Hatem.Ltaief}@kaust.edu.sa

² LESIA, Observatoire de Paris, CNRS, UPMC, Universite Paris Diderot

{Eric.Gendron,Damien.Gratadour,Arnaud.Sevin,Fabrice.Vidal}@obspm.fr

Abstract. Multi-object adaptive optics (MOAO) is a novel adaptive optics (AO) technique dedicated to the special case of wide-field multi-object spectrographs (MOS). It applies dedicated wavefront corrections to numerous independent tiny patches spread over a large field of view (FOV). The control of each deformable mirror (DM) is done individually using a tomographic reconstruction of the phase based on measurements from a number of wavefront sensors (WFS) pointing at natural and artificial guide stars in the field. The output of this study helps the design of a new instrument called MOSAIC, a multi-object spectrograph proposed for the European Extremely Large Telescope (E-ELT)¹. We have developed a novel hybrid pseudo-analytical simulation scheme that allows us to accurately simulate in detail the tomographic problem. The main challenge resides in the computation of the tomographic reconstructor, which involves pseudo-inversion of a large dense symmetric matrix. The pseudo-inverse is computed using an eigenvalue decomposition, based on the divide and conquer algorithm, on multicore systems with multi-GPUs. Thanks to a new symmetric matrix-vector product (SYMV) multi-GPU kernel, our overall implementation scores significant speedups over standard numerical libraries on multicore, like Intel MKL, and up to 60% speedups over the standard MAGMA implementation on 8 Kepler K20c GPUs. At 40,000 unknowns, this appears to be the largest-scale tomographic AO matrix solver submitted to computation, to date, to our knowledge and opens new research directions for extreme scale AO simulations.

1 Introduction

Astronomical programs characterizing high redshift galaxies to study their formation and evolution, require to observe a large number of objects in parallel in

¹ <http://www.eso.org/public/teles-instr/e-elt>

the largest field possible to achieve a sufficient statistic for a reasonable observing time. This is the principle of multi-object spectroscopy. Moreover, it is also necessary to resolve these galaxies so as to perform integral field spectroscopy on their structures. The required resolution should be much better than atmospheric seeing, typically 50 to 100 milli-arcseconds, and therefore requires the implementation of adaptive optics (AO), an instrumental technique for the compensation of dynamically evolving aberrations in an optical system (i.e., due to atmospheric turbulence in the case of a ground based telescope). One of the instruments proposed for the future European Extremely Large Telescope (E-ELT) is MOSAIC [8], a multi-object integral field (multi-IFU) spectrograph for the analysis of distant galaxies, a merger of the EAGLE and OPTIMOS-EVE phase A projects [5,15]. It must be equipped with a specific AO concept, called multi-object AO (MOAO).

In this paper, we present an efficient approach for simulating the behavior of a MOAO system on extremely large telescopes, based on a novel hybrid, pseudo-analytical simulation scheme, somewhere in between the end-to-end and purely analytical approaches, that allows us to simulate in detail the tomographic problem as well as noise and aliasing with a high fidelity. The advantage of this pseudo-analytical approach is its accuracy, as compared to a pure Fourier approach (as developed for instance in [16]), since it is using the same reconstructor as the one that would be used on sky, while being extremely fast as compared to a standard end-to-end approach. The main challenge resides in the computation of the tomographic reconstructor which is split in three phases: 1) the eigen decomposition of a large dense symmetric matrix (typically $40\,000 \times 40\,000$ elements, or greater) corresponding to the covariance matrix of the turbulence using a divide-and-conquer algorithm, 2) the explicit pseudo-inversion computation of the covariance matrix and 3) the computation of the tomographic reconstructor using matrix-matrix multiplication kernel. Thanks to their high memory bandwidth and their compute-intensive capabilities (high ratio floating-point operations per memory byte loaded i.e., the so-called *surface to volume* effect), hardware accelerators, such as GPUs, are natural candidates for such workloads. Our contributions are twofold. We have further optimized the existing multi-GPU symmetric eigensolver [21] from the Matrix Algebra on GPU and Multicore Architectures library [2] (MAGMA) by integrating a new symmetric matrix-vector product (SYMV), which represents one of the main performance bottlenecks for symmetric eigensolvers due to its memory-bound nature (phase 1). We have also developed a linearly scaling matrix-matrix multiplication kernel on multi-GPUs (phases 2 and 3).

The remainder of the paper is organized as follows. Section 2 introduces the novel MOAO approach. Section 3 presents the mathematical model for simulating the MOAO technique. Section 4 recalls the major computational steps of the dense symmetric eigensolver. Section 5 describes the parallel implementation of the overall tomographic reconstructor which includes an efficient matrix-matrix multiplication kernel on multi-GPUs. Section 6 highlights the performance results on multi-GPUs. Also, performance comparisons against the state of the art,

high performance dense linear algebra software libraries are shown on x86 as well as GPUs i.e., Intel MKL [13], CULA [1] and the standard MAGMA implementation. Finally, Section 8 summarizes the results of this paper and presents the ongoing work.

2 The Multi-Object Adaptive Optics Technique

Measuring the wavefront disturbances is achieved, by conventional AO systems, using a wavefront sensor (WFS), which is of a Shack-Hartmann type for most of systems currently in operations [6]. The WFS splits optically the telescope pupil into a number of sub-apertures and makes as many images of a sufficiently bright stellar guide source in each sub-apertures. The exact position of each image, influenced by the turbulence, allows to determine the local slope (i.e., derivative) of the wavefront in front of each sub-aperture. The WFS measurement ends up with a vector field, sampled as the sub-aperture pattern, and describing the wavefront gradient over the pupil area.

The high redshift galaxies are much too faint to provide guide sources for the wavefront measurement for AO and one should find field stars bright enough to ensure this measurement. To obtain 100% sky coverage, a critical aspect for cosmological programs, it is necessary to create artificial guide stars by the backscattering of a laser beam on the Sodium layer of the atmosphere [10] so as to deal with the low stars density found in cosmological fields. These artificial stars are called Laser Guide Stars (LGS) as opposed to Natural GS (NGS). Whatever type of GS, natural or laser, they can be used to measure the atmospheric turbulence in directions that are not those directions of interest (those galaxies to be observed). Tomography algorithms must thus be developed to allow optimal reconstruction of the turbulent volume and the calculation, by projection on the different directions of interest, of the correction to be applied.

Moreover, the fields of interest are very large (5 to 10 arc minutes) compared to the capabilities of conventional AO and a deformable mirror (DM), or even several, compensating the whole field of view is not an adequate solution. Additionally, serious problems arise when trying to implement such optical designs. In fact, only the galaxies must be corrected in this large field, i.e., small patches of few arcseconds, but not the entire field. In the MOAO concept, a specific optical train is placed in the direction of each object of interest including a dedicated DM to ensure correction. Aligned with the linear approach of wavefront reconstruction used in classical AO systems, the tomographic reconstructors proposed up to now in the literature are linear operators [7]. The input data is a vector that concatenates all the measurements taken at a given moment of all the WFSs staring at NGS and LGS. On output, the multiplication by the tomographic matrix will produce a vector that will represent either the phase in the volume (expressed in a suitable basis), or the voltages of a DM.

We have chosen to follow an approach that we have used on the CANARY experiment [9]: the “Learn & Apply”. The tomographic reconstructor is aimed at retrieving the wavefront measurements that a virtual sensor would see when

looking at a source located on the scientific target and called *truth* sensor (TS). As finding this reconstructor is an inverse problem, it is searched using a minimal mean square error (MMSE) approach, relying on priors on turbulence parameters (Kolmogorov assumption, global Fried parameter, $C_n^2(h)$ profile, wind speed profile, etc.) in order to constraint it and provide regularization. This reconstructor can then be used either to control a real system, or in our case to compute the reconstruction error using an analytical model for the various terms of the system error budget. From this reconstructor, we derive numerically the covariance matrix of the tomographic error, including aliasing and propagated noise. We are then able to simulate the point-spread function (PSF) associated to this covariance matrix of the residuals. The obtained long exposure PSF is then multiplied, in the Fourier space, by the product of the optical transfer functions (OTF) corresponding to bandwidth and fitting errors.

3 Mathematical Model

Because we aim to simulate the image quality attained on the E-ELT using MOAO, the end product we are looking for is the long-exposure point spread function (PSF). The latter is the Fourier transform of the optical transfer function (OTF). Under the hypothesis of stationarity of the phase, it has been shown, for instance in [19] that the OTF can be written as $OTF(\rho/\lambda) = OTF_{\text{tel}}(\rho/\lambda) \exp(-\frac{1}{2}D_\phi(\rho))$, with OTF_{tel} the optical transfer function of the telescope, and $D_\phi(\rho)$ the structure function of the residual phase.

We will assume that the residual errors induced by the AO correction will be made of three independent terms: 1) a term due to the DM fitting error induced by the limited number of actuators on the DM, 2) a term due to temporal error, induced by the finite system bandwidth and 3) a term made of the tomographic error, the associated aliasing, and the noise propagated from the measurements through the tomographic reconstructor. A structure function will be associated to each of these terms, that will be computed from the power spectral density of the residual phase for the first two items, exactly as proposed in [17,14]. The computation of the third term is explained below. We will assume that these 3 terms behave as independent processes. Thus, the structure function of the residual phase can be written as the following sum: $D_\phi(\rho) = D_{\text{fit}}(\rho) + D_{\text{bw}}(\rho) + D_{\text{tomo}}(\rho)$.

While the computation of the first two structure functions: $D_{\text{fit}}(\rho)$ and $D_{\text{bw}}(\rho)$ is not compute intensive, the computation of the last term: $D_{\text{tomo}}(\rho)$ requires a lot of computing power, especially in the case of the E-ELT, as explained below. As mentioned in the previous section, in our current design for MOAO, an on-axis “truth sensor” is used to calibrate the interaction matrix of the system, that will allow us to control the DM from this virtual WFS measurements minimizing the calibration errors. If we call \mathbf{t} the measurements of the truth sensor and \mathbf{v} the voltages applied on the DM, we can calibrate the interaction matrix D by soliciting each actuator of the DM one by one : $\mathbf{t} = D \mathbf{v}$ and we can control the DM from the TS measurements using $\mathbf{v} = D^\dagger \mathbf{t}$, where D^\dagger is the generalized

inverse of D , possibly with some filtered modes. D^\dagger is usually computed by doing a singular value decomposition (SVD) of $D = U \cdot [\text{diag}(\lambda_i)] \cdot V^t$. The negligible singular values are then filtered out and D^\dagger can then be calculated as follows:

$$D^\dagger = V \cdot [\text{diag}(1/\lambda_1, 1/\lambda_2, \dots, 1/\lambda_k)] \cdot U^t, \quad (1)$$

where k is the numerical rank of D . The tomographic error vector \mathbf{e} , as it would be measured by a noiseless truth sensor, would be $\mathbf{e} = \mathbf{t} - R\mathbf{m}$, where R is the tomographic reconstructor used on the system to drive the DM. Given a particular reconstructor, we can thus compute the covariance matrix C_{ee} of \mathbf{e} as follows:

$$C_{ee} = C_{tt} - C_{tm}R^t - RC_{tm}^t + RC_{mm}R^t. \quad (2)$$

The structure function of the phase tomographic error $D_{\text{tomographic}}(\rho)$ can then be deduced from the statistical covariance matrix of the DM actuators, C_{vv} (the subscript v stands for *volts*). The matrix C_{vv} is computed using $C_{vv} = D^\dagger C_{ee} D^{\dagger t}$, with C_{ee} given in Eq. 2 and D^\dagger explained in Eq. 1. To compute C_{ee} it is thus necessary to introduce a given tomographic reconstructor. It has been shown, for instance in [20], that the Minimum Mean Square Error (MMSE) tomographic reconstructor can be written as:

$$R = C_{tm} \cdot C_{mm}^{-1}, \quad (3)$$

where C_{mm} stands for the covariance matrix between all the measurements of all the WFS of the instrument, and C_{tm} is the covariance matrix between the measurements of the factious truth sensor, and all the other system measurements. In the case of the E-ELT, C_{mm} is an extremely large matrix (40k x 40k or greater) and its inversion is thus the most compute intensive part of our pseudo-analytical model. It must be noted that the inversion of matrix C_{mm} in the previous equation is not a strict inversion, as the null space of C_{mm} may not be empty. Inverting C_{mm} may be done using eigen decomposition, and filtering out the negligible eigenvalues.

4 Dense Symmetric Eigensolver Algorithm

The LAPACK dense symmetric eigensolver (DSYEVD) is composed of three computational stages. The matrix is first reduced to tridiagonal (DSYTRD) form using orthogonal transformations based on Householder reflectors, which guarantees numerical stability. The reflectors are saved in the reduced lower or upper part of the matrix, depending on which part is considered, since they will be required at the last stage. The second stage extracts all eigenvalues from the tridiagonal matrix and optionally computes all eigenvectors using a divide-and-conquer algorithm (DSTEDC). The third stage corresponds to the back transformation where all orthogonal transformations from the first stage are applied by block to the eigenvector matrix (DORMTR). If only eigenvalues are needed, the routine DYSTRD is called followed by DSTERF, which calculates only the eigenvalues out of the tridiagonal matrix and has an algorithmic complexity of $O(n^2)$

compared to $O(n^3)$ for DSYTRD. One of the main performance bottlenecks of DSYEVD is DSYTRD due to its expensive panel factorization, which requires loading into memory the whole unreduced part of the matrix (i.e., the trailing submatrix) at each single reduction step to perform Level 2 BLAS operations (memory-bound) i.e., the symmetric matrix-vector product (DSYMV). The update of the trailing submatrix is however compute-intensive and relies on high performance Level 3 BLAS operations (compute-bound). When all eigenvectors are additionally needed, DSTEDC and DORMTR are also based on successive calls to Level 3 BLAS kernels and easily achieve high performance close to the matrix-matrix multiplication kernel performance (DGEMM) on modern parallel architectures.

5 Implementation Details

The DSYMV is a memory-bound kernel that represents the main bottleneck in the DSYEVD algorithm. We present an optimized DSYMV kernel, which is a variant of a previously version proposed by some of the authors [3]. The new version has some improvements, such as the elimination of the need to a workspace for global reduction, and the use of atomic operations to allow multiple threads working on the same output location. These new optimizations are suitable for the Kepler architecture². In contrast with the old design [3], the DSYMV is BLAS compliant and achieves higher occupancy on the GPU for relatively small matrices. For the multi-GPU DSYMV, the matrix layout over the GPUs is decided by the upper level algorithm (DSYTRD). A multi-GPU version of this algorithm is proposed by MAGMA [21]. The matrix layout is block-column 1D cyclic distribution. Since we intend to integrate our DSYMV into MAGMA, we use the same layout.

Once the eigenvalue decomposition is complete, the pseudo inverse of the covariance matrix can be computed, $C_{mm}^{-1} = U \cdot E^{-1} \cdot U^t$, where U is the matrix of eigenvectors and E is a diagonal matrix containing the eigenvalues. Afterwards, the tomographic reconstructor can be computed as in Equation 3, where C_{tm} is a rectangular matrix of a typical size $3.5k \times 40k$. It is not trivial, though, to perform such operations on huge matrices. One optimization for the pseudo inverse is to compute the square root of E^{-1} and multiply it by U . This multiplication is simplified to scaling the columns of U by the square roots of E^{-1} . The resulting matrix (say \bar{U}) can then be used to compute C_{mm}^{-1} , since $C_{mm}^{-1} = \bar{U} \cdot \bar{U}^T$. We propose a statically scheduled DGEMM on multi-GPU systems. The proposed kernel performs the standard BLAS operation, $C = \alpha A \cdot B + \beta C$, where A , B , and C have the dimensions $m \times k$, $k \times n$, and $m \times n$, respectively³. The design is based on processing matrices with tiles. cuBLAS DGEMM [18] is used to perform the

² <http://www.nvidia.com/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf>

³ NVIDIA's cuBLAS-XT library provides a similar kernel, but it is not freely available on multi-GPUs.

product at the tile level. It is a highly optimized kernel that achieves approximately 1.1 Tflop/s in double precision on a Kepler GPU. Since the matrices might not fit into GPUs’ main memory, our implementation is an out-of-core DGEMM. The tiles are exchanged between CPU and GPU(s) as needed during computation. The communication overhead should be hidden by useful computation. Figure 1 shows how the work is assigned among four GPUs, where A and B are both processed in the non-transposed mode. For simplicity, assume that m , n , and k can be fully divided into a given tile size. The block rows of A are assigned to GPUs in a 1D cyclic manner. Each GPU reads a block row of A , tile by tile, and does all the computation associated with it. A GPU reads B in block columns, tile by tile, and writes the corresponding result in C (1D cyclic block row, tile by tile). An important point is the memory consumption per GPU.

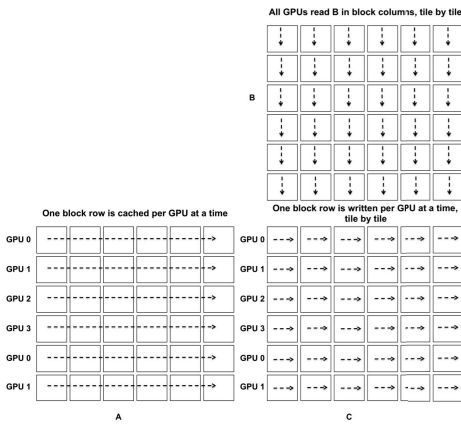


Fig. 1. Strategy and memory requirement for the proposed DGEMM-MGPU

square matrices beyond 100k in double precision beyond the targeted size here.

From Figure 1, we can determine the device memory requirement for the proposed DGEMM-MGPU, for a sample matrix whose dimension k fits into six tiles. Each GPU requires as many tiles as necessary to store a block row of A . The implementation uses double buffers for B and C tiles in order to overlap communication with computation. The total memory requirement (M), in bytes, per GPU is given by, $M = (kT + 4T^2)P$, where P is the precision, expressed in bytes, and T is the tile size. Only tile sizes higher than 1000 are considered, to ensure approximately 1.1 Tflop/s performance per tile. Even with a large tile size of 5k, the kernel can process

6 Experimental Results

The experiments have been conducted on a system equipped with Intel Sandy Bridge CPU (2 sockets \times 8 cores per socket), and accelerated with eight Kepler K20c GPUs (ECC off), each with 5 GB of memory. We use CUDA Toolkit 5.5 and Intel MKL (Intel Composer XE 2013) to build MAGMA-1.4.0. CULA R17 does not distribute DSYEVD and provides only the DSYEV algorithm (symmetric eigensolver using the QR iteration) on single GPU. All computations are done in double precision. Results are properly averaged across multiple runs.

Starting with the DSYMV, The proposed kernel outperforms the state-of-the-art implementations, including MAGMABLAS-1.4.0, cuBLAS-5.5, and CULA-R17. Figure 2(a) shows the performance of the proposed kernel against the

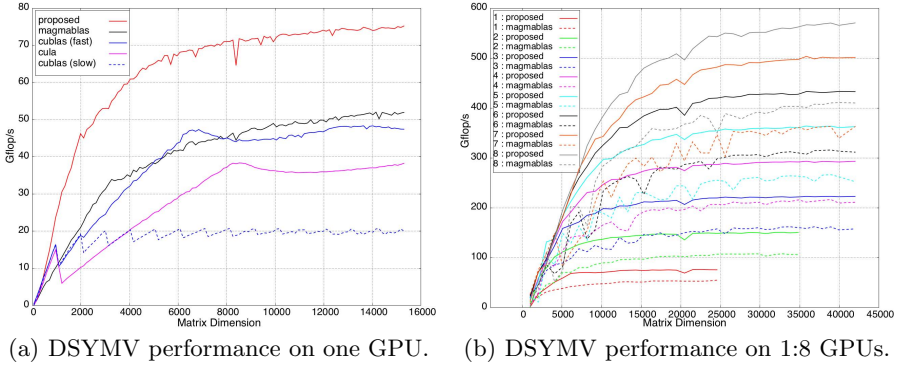


Fig. 2. DSYMV performance on single and multi-GPUs (Kepler K20c - ECC off)

mentioned high performance libraries, on a single Kepler K20c GPU. The Figure shows an asymptotic improvement of 97% against CULA-R17, 56% against cuBLAS-5.5, and 44% against MAGMABLAS-1.4.0. For small matrix dimensions (less than 2000), the improvement is up to 2x against the best implementation, which is a crucial result for the DSYTRD algorithm. The performance is about 88% of the sustained peak bandwidth performance. cuBLAS appears twice in the Figure, since it provides two implementations. In addition to our multi-GPU implementation, only MAGMABLAS provides the DSYMV kernel on multi-GPUs. Figure 2(b) shows the performance of both implementations, on a single node with 8 GPUs. The asymptotic performance speedup over 8 GPUs is up to 40%.

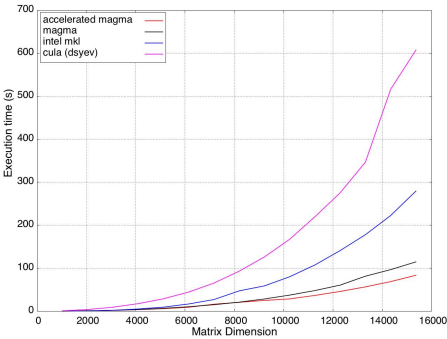
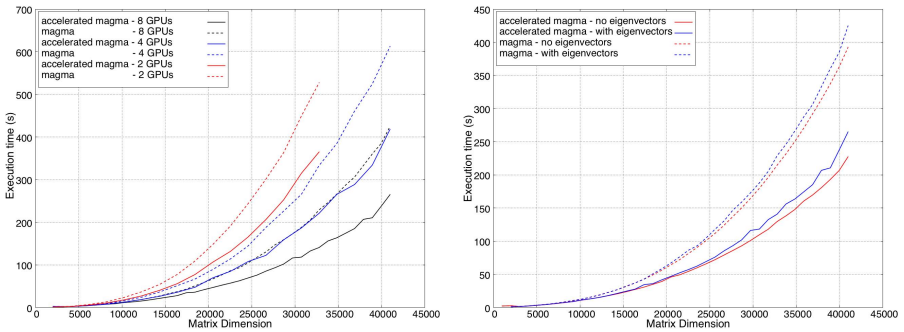


Fig. 3. DSYEVD Execution time using one GPU against multi-core system.

here, although our application, in principle, will filter out up to 20% of the eigenpairs with very low magnitude. Figure 3 shows the execution time of the DSYEVD algorithm for the aforementioned libraries. GPU accelerated libraries

The dominant part of DSYEVD is the DSYTRD stage, which is, in turn, dominated by symmetric matrix-vector product. Thanks to MAGMA being open source, we were able to replace MAGMABLAS DSYMV kernel with the one we propose. For the single GPU case, we compare the three libraries offering symmetric eigenvalue decomposition, Intel MKL, MAGMA-1.4.0, and CULA-R17. For MAGMA, we run an additional accelerated version (from now on referred to as accelerated MAGMA) where our proposed DSYMV is used. All eigenvalues and eigenvectors are computed

uses one Kepler K20c GPU (ECC off). The accelerated MAGMA DSYEVD using our DSYMV kernel is faster than the original MAGMA DSYEVD by 35%. It achieves 3.4x speedup against MKL, and up to 7.2x against CULA DSYEV. Figure 4(a) shows the execution time for the DSYEVD-MGPU, where the accelerated MAGMA achieves speedups up to 45%, 60%, and 70%, on 2, 4, and 8 GPUs, respectively. We notice that the original MAGMA routine has a preprocessing step for a workspace, performed every time before the DSYMV-MGPU routine is called. Since our DSYMV-MGPU routine does not need a workspace, we save the initialization time in addition to the saving due to the more optimized routine. Figure 4(b) shows the overhead of computing the eigenvectors over a run that computes only eigenvalues. Our results still show that the dominant part in the operation is the DSYTRD part, since the backward transformation phase is compute-bound and can be done very efficiently on the GPU.



(a) DSYEVD scalability against the number of GPUs. (b) Overhead of eigenvector computations on 8 GPUs.

Fig. 4. DSYEVD execution time analysis

Table 1. Performance (Tflop/s) of the pseudo inversion

Tile size	4 GPUs	8 GPUs
1000	3.50	5.39
1600	3.60	6.04
2000	3.99	6.36
2500	4.03	7.74
4000	3.52	5.23
5000	4.16	8.05

balance, which will result in a performance drop. However, for the matrix sizes of the application, we can achieve performance that is very close to the peak, if we choose the right tile size. The computation of the pseudo inverse involves multiplication of two $40k \times 40k$ matrices. Using different tile sizes, the performance is summarized in Table 1, for 4 and 8 GPUs. It is clear that the larger

We also present the performance of the DGEMM kernel, which is designed specifically for the application, although it can be easily modified to serve as general purpose DGEMM. Communicating tiles between the CPU and the GPU is done asynchronously, so that the GPU can process existing tiles while receiving new ones, and the CPU can do useful work concurrently with the GPU. In this case, the useful work is scheduling more tiles to be processed. Figure 1 shows that our static scheduling strategy may suffer from load imbalance,

the tile, the better the performance. Tile size 5000 achieves performance that is very close to the sustained peak.

Finally, we present the overall simulation performance. Thanks to the proposed DSYMV and DGEMM kernels, the accelerated MAGMA finishes the simulation in 263.49s. This is 60% better than an original MAGMA implementation (421.10s). It is also 17.5x faster than Intel MKL on 16 core Intel Sandy Bridge processor (4656.25s). To prove how dominant the DSYEVD is, our results show that it takes 241.07s on accelerated MAGMA, 399.15s with an original MAGMA, and 4370.50s using Intel MKL. The other phases are less than 10% of the total run time, for the accelerated MAGMA case.

7 Related Work

It is noteworthy to mention that there exist other numerical algorithms for dense symmetric eigensolver, which tries to workaroud DSYTRD's bottlenecks by introducing a two-stage tridiagonal reduction. The original dense matrix is first reduced to band form using efficient compute-intensive kernels from which a bulge chasing algorithm is applied to chase down the off-diagonal elements until the final tridiagonal matrix is formed. This allows to cast most of Level 2 BLAS operations into Level 3 BLAS operations and increases significantly the overall symmetric eigensolver. This two-stage approach has been first introduced by Bischof et. al [4]. Haidar et. al [11,12] have further improved it on shared-memory multicore architecture. All aforementioned implementations run only on multicore and are very challenging to port on GPUs due to the non-conventional kernels involved in the bulge chasing procedure. The authors presented only performance results based on a MATLAB implementation.

8 Conclusion and Future Work

This paper has presented 1) an efficient implementation of the DSYMV kernel on multi-GPUs, which is a critical Level 2 BLAS operation for the dense symmetric eigensolver DSYEVD and 2) an optimized asynchronous DGEMM kernel on multi-GPUs. Thanks to both implementations and the multi-object adaptive optics approach, the overall application accurately solves unprecedented problem scale in the adaptive optics field (up to our knowledge) and reaches high performance on multi-GPUs compared to the standard MAGMA implementation on 8 Kepler K20c GPUs (up to 60% speedup). New research directions for extreme scale AO simulations can be envisaged by replacing the actual dense symmetric eigensolver to compute the tomographic reconstructor with the Cholesky-based symmetric matrix inversion. This would permit to calculate the explicit inverse directly without intermediary computational steps as well as to port this overall application to distributed-memory systems with GPUs more easily. This method would also allow to capture and to better handle the noise propagated from the measurements through the tomographic reconstructor. In addition, the covariance matrix generation has not been investigated here since the time taken to

compute it is not dominant for the problem sizes studied in the paper. However, it would have to be considered for large problem sizes, especially when targeting distributed-memory environment. Future possibilities also include exploitation of the low rank character of the co-variance matrix and reuse of information between instances that are currently treated as independent.

Acknowledgment. This work is supported by the ANR grant ANR-12-MONU-0022 of the French Ministry of Research. We thank NVIDIA for their continuous support and hardware donations. We also thank the CSCS Swiss National Supercomputing Centre for granting us access on their GPU computing platforms.

References

1. CULA Dense Free Edition, <http://www.culatools.com/>
2. Matrix Algebra on GPU and Multicore Architectures. Innovative Computing Laboratory, University of Tennessee, <http://icl.cs.utk.edu/magma/>
3. Abdelfattah, A., Dongarra, J., Keyes, D., Ltaief, H.: Optimizing Memory-Bound SYMV Kernel on GPU Hardware Accelerators. In: Daydé, M., Marques, O., Nakajima, K. (eds.) VECPAR. LNCS, vol. 7851, pp. 72–79. Springer, Heidelberg (2013)
4. Bischof, C.H., Lang, B., Sun, X.: Algorithm 807: The SBR Toolbox—software for successive band reduction. *ACM Transactions on Mathematical Software* 26(4), 602–616 (2000)
5. Cuby, J.-G., Morris, S., Fusco, T., Lehnert, M., Parr-Burman, P., Rousset, G., Amans, J.-P., Beard, S., Bryson, I., Cohen, M., Dipper, N., Evans, C., Ferrari, M., Gendron, E., Gimenez, J.-L., Gratadour, D., Hastings, P., Hubert, Z., Hugot, E., Jagourel, P., Laporte, P., Lebrun, V., Le Mignant, D., Madec, F., Myers, R., Neichel, B., Morris, T., Robert, C., Schnetler, H., Swinbank, M., Talbot, G., Taylor, W., Vidal, F., Vivès, S., Vola, P., Welikala, N., Wells, M.: EAGLE: a MOAO fed multi-IFU NIR workhorse for E-ELT. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 7735 (July 2010)
6. Davies, R., Kasper, M.: Adaptive Optics for Astronomy. *Annual Review of Astronomy and Astrophysics* 50, 305–351 (2012)
7. Ellerbroek, B.L., Vogel, C.R.: TOPICAL REVIEW: Inverse problems in astronomical adaptive optics. *Inverse Problems* 25(6), 063001 (2009)
8. Evans, C., Puech, M., Barbuy, B., Bastian, N., Bonifacio, P., Caffau, E., Cuby, J.-G., Dalton, G., Davies, B., Dunlop, J., Flores, H., Hammer, F., Kaper, L., Lemasle, B., Morris, S., Pentericci, L., Petitjean, P., Schaerer, D., Telles, E., Welikala, N., Ziegler, B.: ELT-MOS White Paper: Science Overview and Requirements. ArXiv e-prints (February 2013)
9. Gendron, E., Vidal, F., Brangier, M., Morris, T., Hubert, Z., Basden, A., Rousset, G., Myers, R., Chemla, F., Longmore, A., Butterley, T., Dipper, N., Dunlop, C., Geng, D., Gratadour, D., Henry, D., Laporte, P., Looker, N., Perret, D., Sevin, A., Talbot, G., Younger, E.: MOAO first on-sky demonstration with CANARY. *Astronomy and Astrophysics* 529, L2 (2011)
10. Gratadour, D., Gendron, E., Rousset, G.: Intrinsic limitations of Shack-Hartmann wavefront sensing on an extended laser guide source. *Journal of the Optical Society of America A* 27(26), A260000 (2010)

11. Haidar, A., Ltaief, H., Dongarra, J.: Parallel Reduction to Condensed Forms for Symmetric Eigenvalue Problems using Aggregated Fine-Grained and Memory-Aware Kernels. In: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2011, pp. 8:1–8:11. ACM, New York (2011)
12. Haidar, A., Ltaief, H., Dongarra, J.: Toward a High Performance Tile Divide and Conquer Algorithm for the Dense Symmetric Eigenvalue Problem. *SIAM J. Scientific Computing* 34(6) (2012)
13. Intel. Math Kernel Library, <http://software.intel.com/en-us/articles/intel-mkl/>
14. Jolissaint, L., Christou, J., Wizinowich, P., Tolstoy, E.: Adaptive optics point spread function reconstruction: lessons learned from on-sky experiment on Altair/Gemini and pathway for future systems. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 7736 (July 2010)
15. Navarro, R., Chemla, F., Bonifacio, P., Flores, H., Guinouard, I., Huet, J.-M., Puech, M., Royer, F., Pragt, J.H., Wulterkens, G., Sawyer, E.C., Caldwell, M.E., Tosh, I.A.J., Whalley, M.S., Woodhouse, G.F.W., Spanò, P., di Marcantonio, P., Andersen, M.I., Dalton, G.B., Kaper, L., Hammer, F.: Project overview of OPTIMOS-EVE: the fibre-fed multi-object spectrograph for the E-ELT. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 7735 (July 2010)
16. Neichel, B., Fusco, T., Conan, J.-M., Petit, C., Rousset, G.: PSD-based simulation algorithm for Wide FoV AO design: application to ELT studies. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 7015 (July 2008)
17. Rigaut, F.J., Veran, J.-P., Lai, O.: Analytical model for Shack-Hartmann-based adaptive optics systems. In: Bonaccini, D., Tyson, R.K. (eds.) Adaptive Optical System Technologies. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 3353, pp. 1038–1048 (September 1998)
18. Tan, G., Li, L., Triechle, S., Phillips, E., Bao, Y., Sun, N.: Fast Implementation of DGEMM on Fermi GPU. In: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2011, pp. 35:1–35:11. ACM, New York (2011)
19. Veran, J.-P., Rigaut, F., Maitre, H., Rouan, D.: Estimation of the adaptive optics long-exposure point-spread function using control loop data. *Journal of the Optical Society of America A* 14, 3057–3069 (1997)
20. Vidal, F., Gendron, E., Rousset, G.: Tomography approach for multi-object adaptive optics. *Journal of the Optical Society of America A* 27(26), A260000 (2010)
21. Yamazaki, I., Dong, T., Solc, R., Tomov, S., Dongarra, J., Schulthess, T.: Tridiagonalization of a dense symmetric matrix on multiple GPUs and its application to symmetric eigenvalue problems. In: Concurrency and Computation: Practice and Experience (2013)