KAUST BASIC LINEAR ALGEBRA ROUTINES ON GPUs

KBLAS

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Extreme Computing Research Center

KAUST BLAS (KBLAS) is a high performance CUDA library implementing a subset of BLAS as well as LAPACK routines on NVIDIA GPUs. Using recursive and batch algorithms, KBLAS maximizes the GPU bandwidth, reuses locally cached data and increases device occupancy. KBLAS supports operations for regular dense and hierarchical low-rank matrix workloads. KBLAS provides, therefore, the critical building blocks not only for the traditional high performance dense linear algebra libraries, but also for the emerging numerical libraries supporting hierarchical low-rank matrix computations. This is a major leap toward leveraging GPU hardware accelerators for high performance low-rank matrix approximations and computations, which currently drives the research agenda of the linear algebra scientific community.

RECURSIVE ALGORITHMS: GPU-Resident POTRF

```
Procedure POTRF_rec(uplo, n, A, lda, info)
    if n ≤ 32 then                          // at the recursion stop size
        blockDim ← dim3(32, 32)
        POTRF_Kernel <<< 1, blockDim >>> (uplo, n, A, lda,
          info)
    else                                    // invoke recursion
        n1 ← largest power of 2 less than n
        n2 ← n − n1
        POTRF_rec(uplo, n1, A, lda, info)   // recursive POTRF
        TRSM(Right, Lower, Trans, NonUnit,
             n2, n1,
             1, A, lda,
               A + n1, lda)
        SYRK(Lower, NoTrans,
             n2, n1,
             −1, A + n1, lda,
             1, A + n1 * lda + n1, lda)
        POTRF_rec(uplo, n2, A + n1 * lda + n1, lda)
          // recursive POTRF
    return;
```

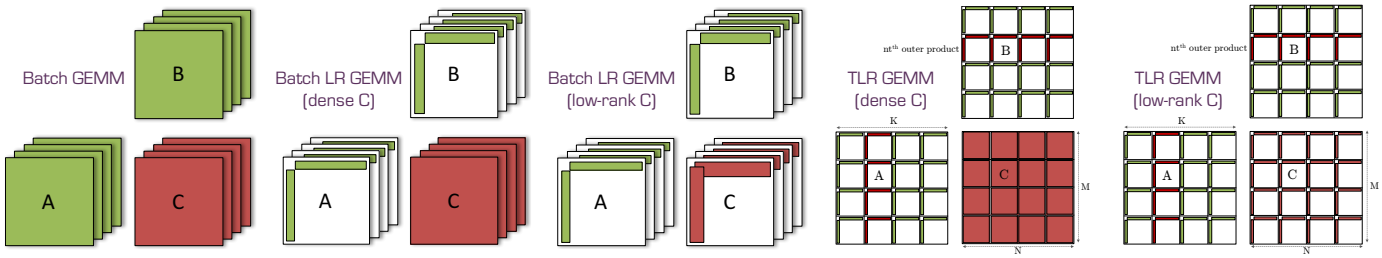
KBLAS 3.0

- Legacy Level-2 BLAS: (‡◇ ⚭ ○) SYMV, GEMV, HEMV.
- Legacy Level-3 BLAS: (‡◇ ⚭ ○) TRSM, TRMM, GEMM (⚭ only).
- Batch Level-3 BLAS: (‡◇ ⚭ ○= ⁎) TRSM, TRMM, SYRK.
- Batch Triangular: (◇‡ ⚭ ○= ⁎) TRTRI, LAUUM.
- Batch Symmetric: (◇‡ ⚭ ○= ⁎) POTRF, POTRS, POSV, POTRI, POTI.
- Batch General: (◇‡ ⚭ ○= ⁎) GESVJ, GERSVD, GEQRF.
- **Batch Tile low-rank GEMM (◇ ◇ ‡ ○ =).**
- **GPU-Resident POTRF kernel (◇ ‡ ○).**

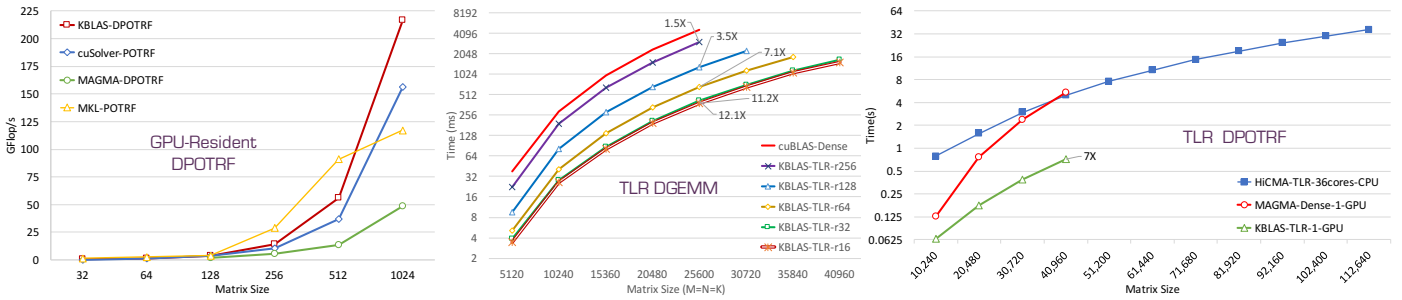
‡ Standard precisions: s/d/c/z.
‡ Real precisions: s/d.
◇ Very small matrix sizes.
◇ Arbitrary sizes.

○ Single-GPU support.
⚭ Multi-GPU support.
= Uniform batch sizes.
⁎ Non-strided and strided variants.

BATCH ALGORITHMS: Tile Low-Rank GEMM – Bottom-Up Approach



PERFORMANCE RESULTS – NVIDIA V100 GPU



DOWNLOAD KBLAS AT: https://github.com/ecrc/kblas

A collaboration of

With support from

Sponsored by

Office Of Sponsored Research

KAUST BASIC LINEAR ALGEBRA ROUTINES ON GPUs

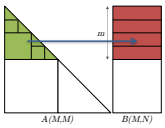
KBLAS

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

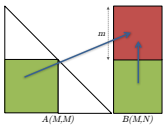
Extreme Computing Research Center

KAUST BLAS (KBLAS) is a high performance CUDA library implementing a subset of BLAS as well as Linear Algebra PACKage (LAPACK) routines on NVIDIA GPUs. Using recursive and batch algorithms, KBLAS maximizes the GPU bandwidth, reuses locally cached data and increases device occupancy. KBLAS supports operations for regular dense and hierarchical low-rank matrix workloads. KBLAS provides

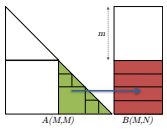
RECURSIVE ALGORITHMS: TRMM and TRSM



Rec. TRMM: $B_1 = \alpha A_1 B_1$
Rec. TRSM: $A_1 X_1 = \alpha B_1$

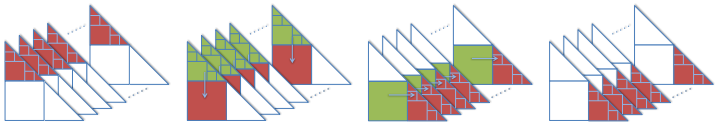


TRMM: GEMM $B_1 = \alpha A_2^T B_2 + B_1$
TRSM: GEMM $B_2 = \alpha B_2 - A_2 B_1$



Rec TRMM: $B_2 = \alpha A_3 B_2$
Rec TRSM: $A_3 X_2 = B_2$.

BATCH ALGORITHMS: Recursive Cholesky POTRF



1. Rec-POTRF
2. Rec-TRSM
3. Rec-SYRK
4. Rec-POTRF

KBLAS HIGHLIGHTS

- KBLAS Level-2 (o): SYMV & HEMV
- KBLAS Level-3 (o): TRMM & TRSM

NVIDIA cuBLAS 6.0

NVIDIA cuBLAS 8.0

KBLAS 2.0

- Legacy Level-2 BLAS: (‡ ◊ ∞ o) SYMV, GEMV, HEMV.
- Legacy Level-3 BLAS: (‡ ◊ ∞ o) TRSM, TRMM, GEMM (∞ only).
- Batch Level-3 BLAS: (‡ ◊ ∞ o= ⁎) TRSM, TRMM, SYRK.
- Batch Triangular: (◊ ‡ ∞ o= ⁎) TRTRI, LAUUM.
- Batch Symmetric: (◊ ‡ ∞ o= ⁎) POTRF, POTRS, POSV, POTRI, POTI.
- Batch General: (◊ ‡ ∞ o= ⁎) GESVJ, GERSVD, GEQRF.

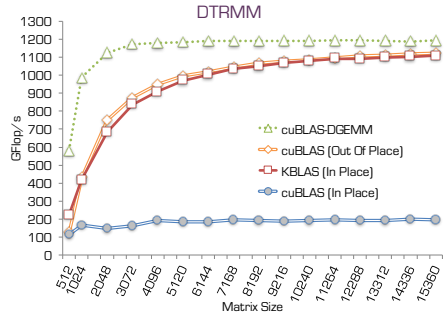
‡ Standard precisions: s/d/c/z.
‡ Real precisions: s/d.
◊ Very small matrix sizes.
◊ Arbitrary sizes.

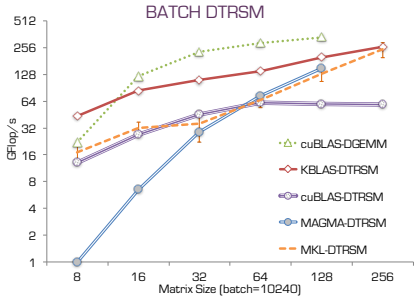
o Single-GPU support.
∞ Multi-GPU support.
= Uniform batch sizes.
⁎ Non-Strided and Strided variants.

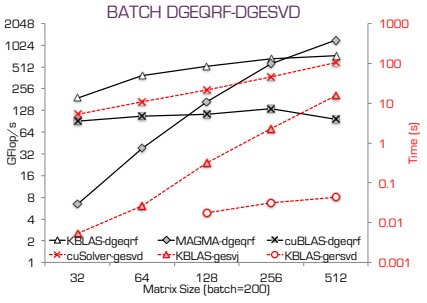
CURRENT RESEARCH

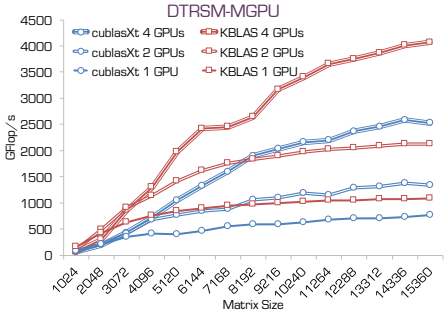
- Half Precision Legacy and Batch BLAS.
- Tile Low-Rank (TLR) BLAS on GPUs.
- Adaptive Cross Approximation (ACA) on GPUs.
- Vectorized Batch BLAS on x86.

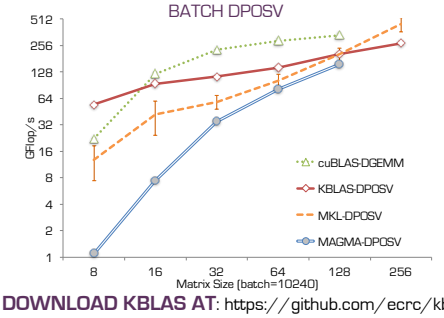
PERFORMANCE RESULTS

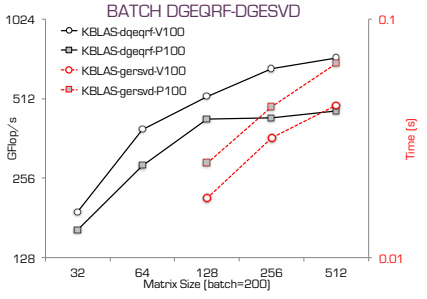












DOWNLOAD KBLAS AT: https://github.com/ecrc/kblas

A collaboration of

With support from

Sponsored by