INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING

ICPP/2021/CHICAGO/USA

acm In-Cooperation

sighpc

AUGUST 9-12, 2021

Recursion Brings Speedup to Out-of-Core TensorCore-based Linear Algebra Algorithms

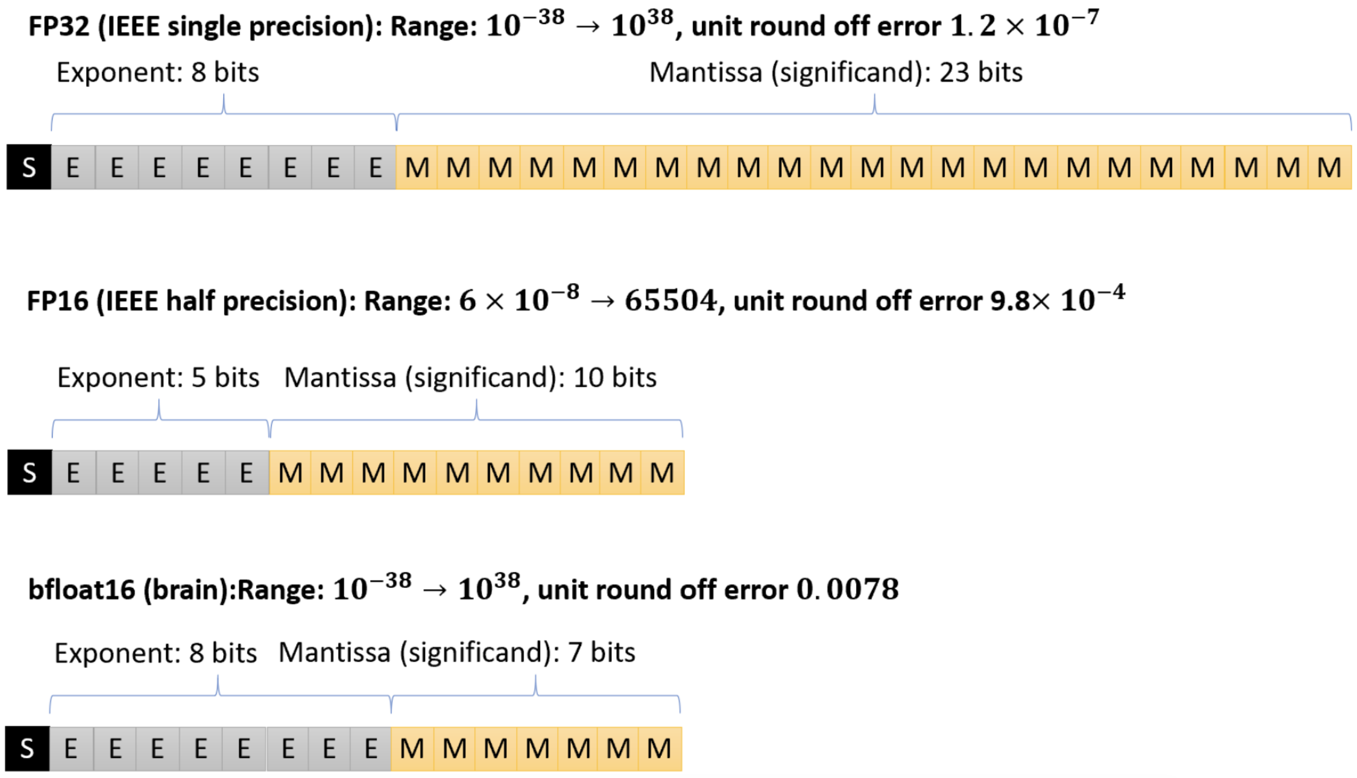
A Case Study of Classic

Gram-Schmidt QR Factorization

Shaoshuai Zhang, Panruo Wu
University of Houston

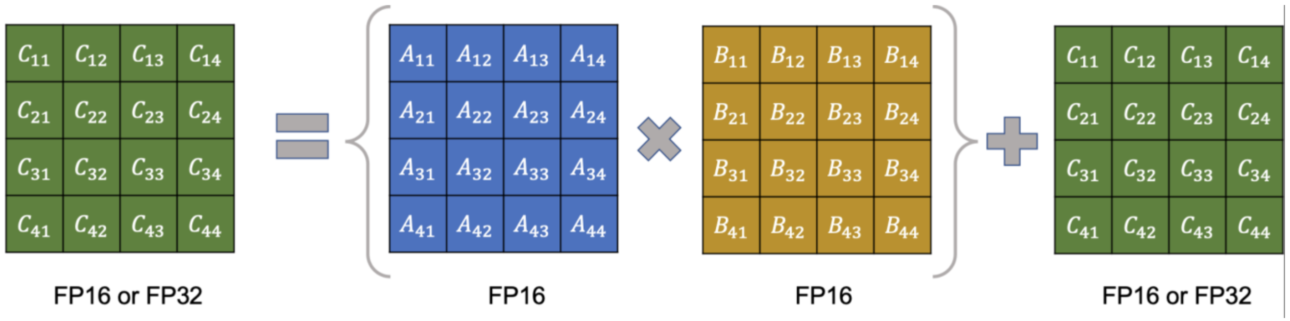
Half Precision Arithmetic and Tensor Core

- Two standard: FP16 and bfloat16
- Bfloat16 has a wider range(the same as FP32) but larger unit round off error
- Nvidia's GPUs only support FP16, except its latest Ampere Architecture (introduced in May, 2020)



Half Precision Arithmetic and Tensor Core

- Tensor Core only supports fused general matrix multiplications (GEMMs)



- The TC-GEMM is much faster than SGEMM and DGEMM

DGEMM	SGEMM	HGEMM	TC-GEMM
5.8 TFLOPS	11.9 TFLOPS	23.1 TFLOPS	97.3 TFLOPS

- We can use Tensor Core by cublas library or WMMA intrinsic functions

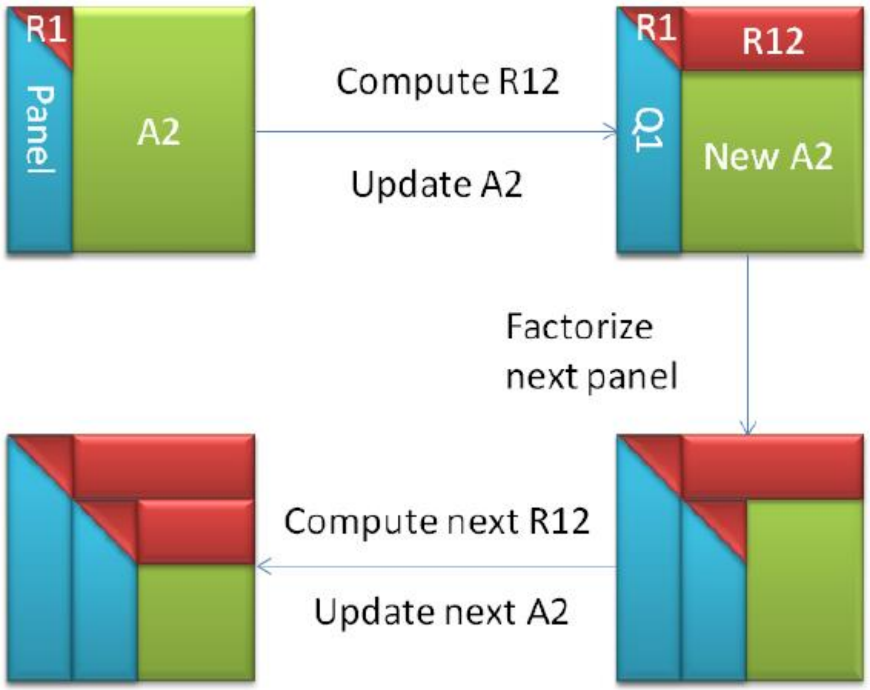
QR factorization

- Decompose a matrix A into a product of an orthogonal matrix Q and an upper triangular matrix R
- Classic Gram-Schmidt QR Factorization

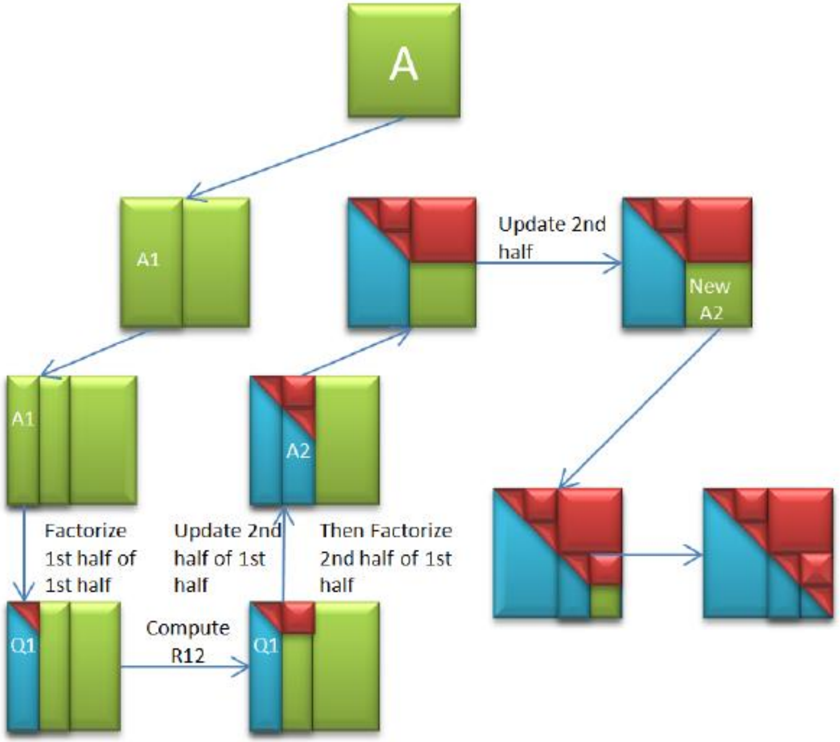
$$
\begin{aligned}
u_1 &= a_1, & q_1 &= u_1/||u_1|| \\
u_2 &= a_2 - \mathrm{Proj}_{u_1}(a_2), & q_2 &= u_2/||u_2|| \\
u_3 &= a_3 - \mathrm{Proj}_{u_1}(a_3) - \mathrm{Proj}_{u_2}(a_3), & q_3 &= u_3/||u_3|| \\
&\vdots & &\vdots \\
u_n &= a_n - \sum_{j=1}^{n-1} \mathrm{Proj}_{u_j}(a_n), & q_n &= u_n/||u_n||
\end{aligned}
$$

QR factorization

Blocking strategy



Recursive strategy



Out-of-Core Processing

- Typically used when the memory is limited
- Disk-CPU Out-of-Core
  - SOLAR
  - ScaLAPACK
- CPU-GPU Out-of-Core
  - cuBLASXt
  - BLASX

Performance Analysis

- Why Recursion?
- Data movement
- Overlap ratio

Data Movement

- Blocking
  - Host to Device $$(k+2)mn + \frac{n^2}{2} - \frac{nb}{2}$$
  - Device to Host $$\frac{1}{2}[(k+1)mn + n^2 + nb]$$
- Recursive
  - Host to Device $$2(\log_2 k + 1)mn + \frac{mn}{2} - \frac{nb}{2}$$
  - Device to Host $$(\frac{1}{2}log_2^k + 1)mn + \frac{n^2}{2}$$

Data movement time	Recursive	Blocking
Host to device	37.9s	47.2s
Device to Host	19.3s	22.3s

Overlap Ratio in GEMMs

- Inner product

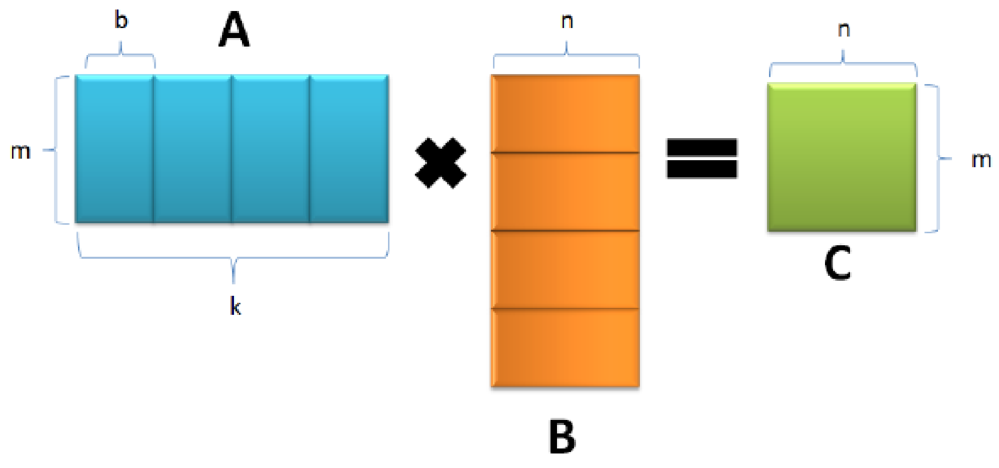


Figure 3: Out-of-core inner product tiling strategy in recursive QR factorization

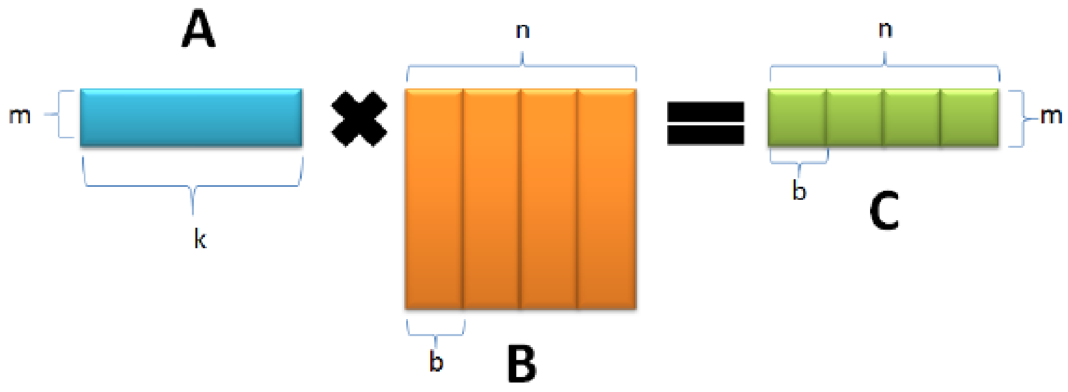


Figure 4: Out-of-core inner product tiling strategy in blocking QR factorization

Overlap Ratio Analysis: Inner product

- The time cost of in-core GEMM tiles needs to be larger than the time cost of data movement from device to host and host to device

	Recursive	**Blocking**
Device to Host	$\frac{4mk+4nk}{R_m}$	$\frac{4kb}{R_m}$
GEMM	$\frac{2mnk}{R_g}$	$\frac{2mkb}{R_g}$
The smallest m	30,000	15,000

- Assume the Rg is 90TFLOPs and Rm is 12GB/s

Overlap Ratio in GEMMs

- Outer product

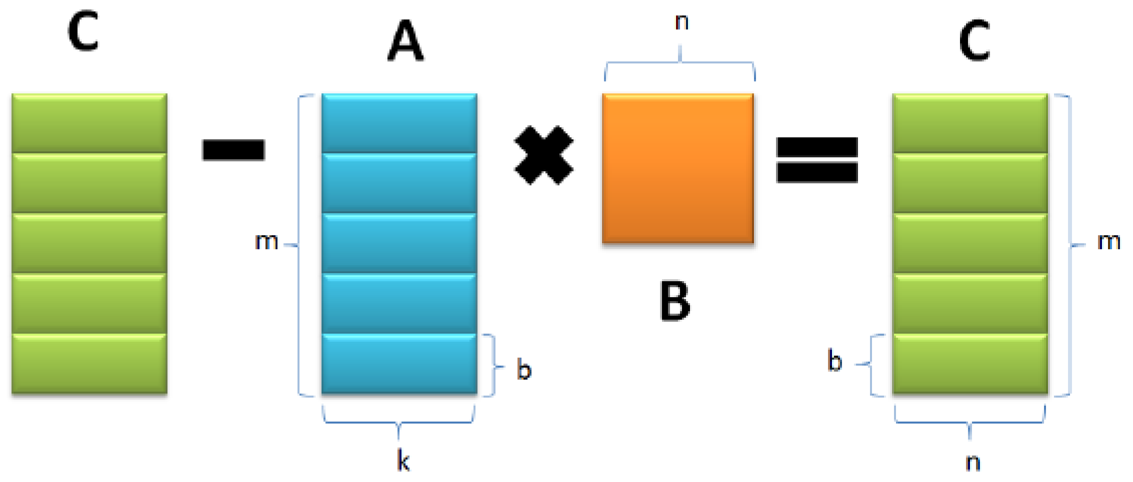


Figure 5: Out-of-core outer product tiling strategy in recursive QR factorization

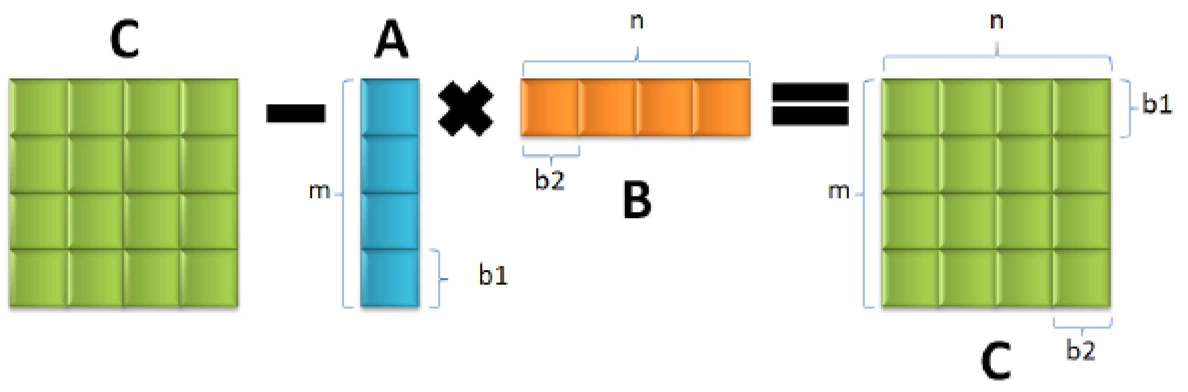


Figure 6: Out-of-core outer product tiling strategy in blocking QR factorization

Overlap Ratio Analysis: Outer product

	Recursive	**Blocking**
Device to Host	$\frac{4bk+4bn}{R_m}$	$\frac{4b_1b_2}{R_m}$
GEMM	$\frac{2bkn}{R_g}$	$\frac{2b_1kb_2}{R_g}$
The smallest m	30,000	15,000

Results of OOC GEMMs: Inner Product

Single Block Time Cost	Recursive	Blocking
Host to device	693ms	728ms
GEMM	1408ms	1337ms
Device to Host	1306ms	81ms
In-core flops	99.9TFLOPs	52.6TFLOPs

Overall Time cost	Recursive	Blocking
Synchronous	18183ms	14920ms
Synchronous flops	62.0TFLOPs	33.0TFLOPs
Asynchronous	12932ms	11286ms
Asynchronous flops	87.1TFLOPs	43.6TFLOPs

Table 1: Inner product behaviors, recursive matrix size is 65536*131072*65536 with blocksize 16384, blocking matrix size is 16384*131072*114688 with blocksize 16384

Results of OOC GEMMs: Inner Product

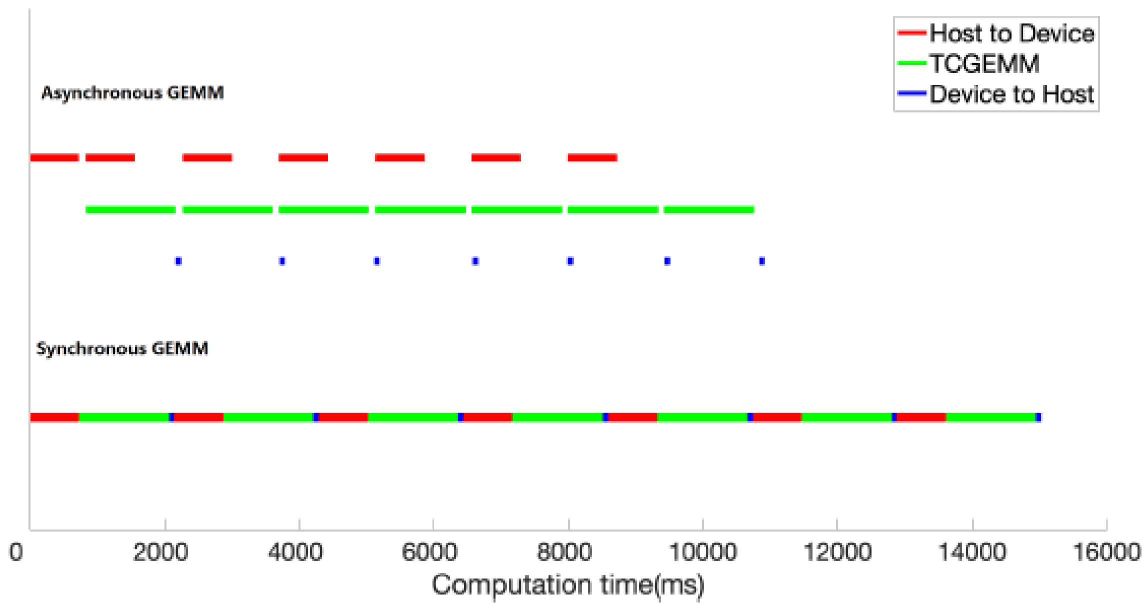


Figure 7: The timeline of computing max inner product GEMM in 0.13M*0.13M in blocking QR factorization, the matrix size is 16384*131072*114688, the blocksize is 16384.

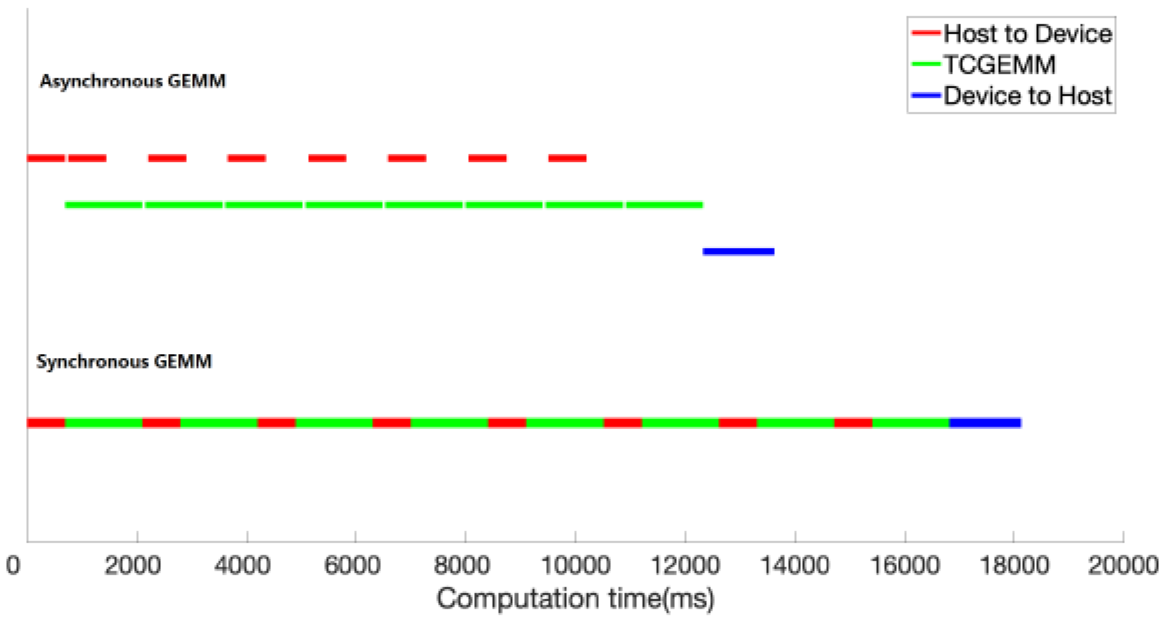


Figure 8: The timeline of computing max inner product GEMM in 0.13M*0.13M in recursive QR factorization, the matrix size is 65536*131072*65536, the blocksize is 16384.

Results of OOC GEMMs: Outer Product

Single Block Time Cost	Recursive	Blocking
Host to device	347ms	86ms
GEMM	654ms	89ms
Device to Host	163ms	81ms
In-core flops	107.6TFLOPs	98.8TFLOPs
Overall Time cost	**Recursive**	**Blocking**
Synchronous	14129ms	5119ms
Synchronous flops	60.3TFLOPs	34.7TFLOPs
Asynchronous	11517ms	11286ms
Asynchronous flops	97.7TFLOPs	96.2TFLOPs

Table 2: Outer product behaviours, recursive matrix size is 131072*65536*65536 with blocksize 8192, blocking matrix size is 131072*16384*114688 with blocksize 16384 and 16384

Results of OOC GEMMs: Outer Product

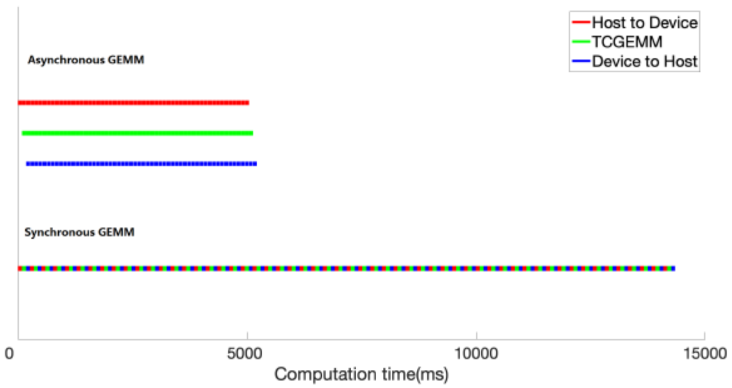


Figure 9: The timeline of computing max outer product GEMM in 0.13M*0.13M in blocking QR factorization, the matrix size is 131072*16384*114688, the blocksize b_1, b_2 is 16384 and 16384.

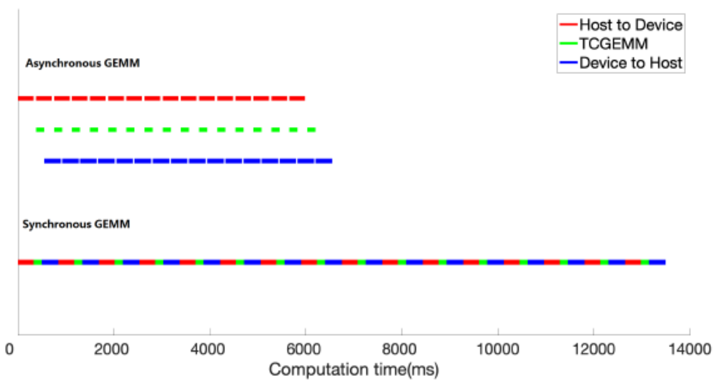


Figure 11: The timeline of computing outer product GEMM with QR blocksize 8192, the matrix size is 131072*16384*131072, the inside GEMM blocksize b_1, b_2 is 32768 and 32768.

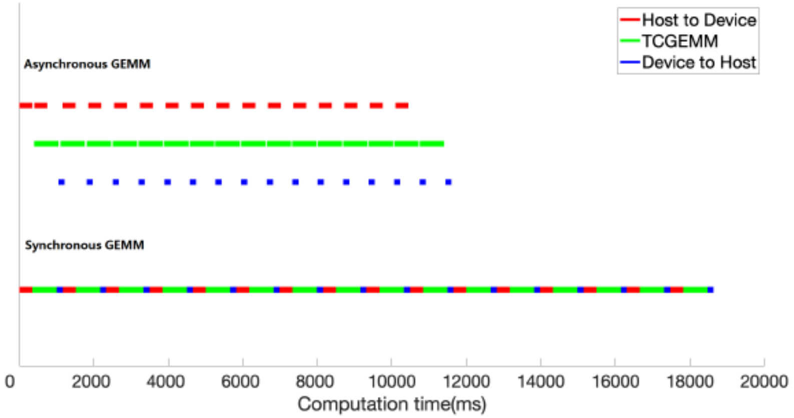


Figure 10: The timeline of computing max outer product GEMM in 0.13M*0.13M in recursive QR factorization, the matrix size is 131072*65536*65536, the blocksize is 8192

Implementation and Optimization

- GEMM-Level Implementation and Optimization
  - Using cuda streams
  - Change blocksize gradually
- QR-Level Implementation and Optimization
  - Cut off unnecessary data movement
    - Cutting off some move-in operations of the panel
  - Enable cross BLAS operation overlapping
    - Hide the move-out operations between panel factorization and GEMMs

Final Results: block size 16384

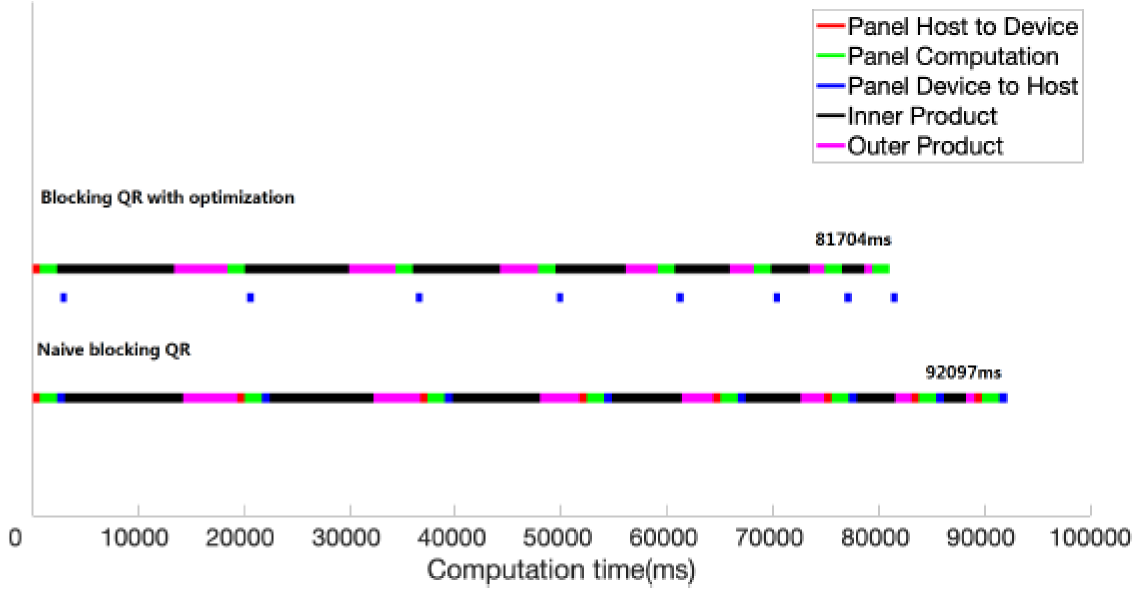


Figure 12: The timeline of computing blocking out-of-core QR, the blocksize is 16384.

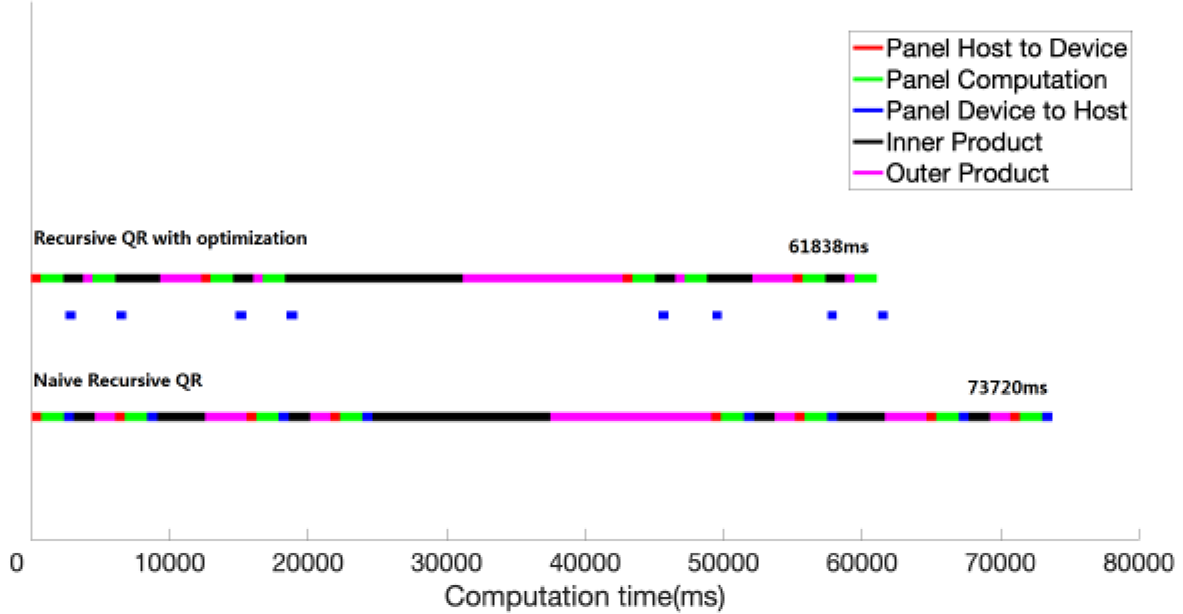


Figure 13: The timeline of computing recursive out-of-core QR, the blocksize is 16384.

Final Results: block size 8192

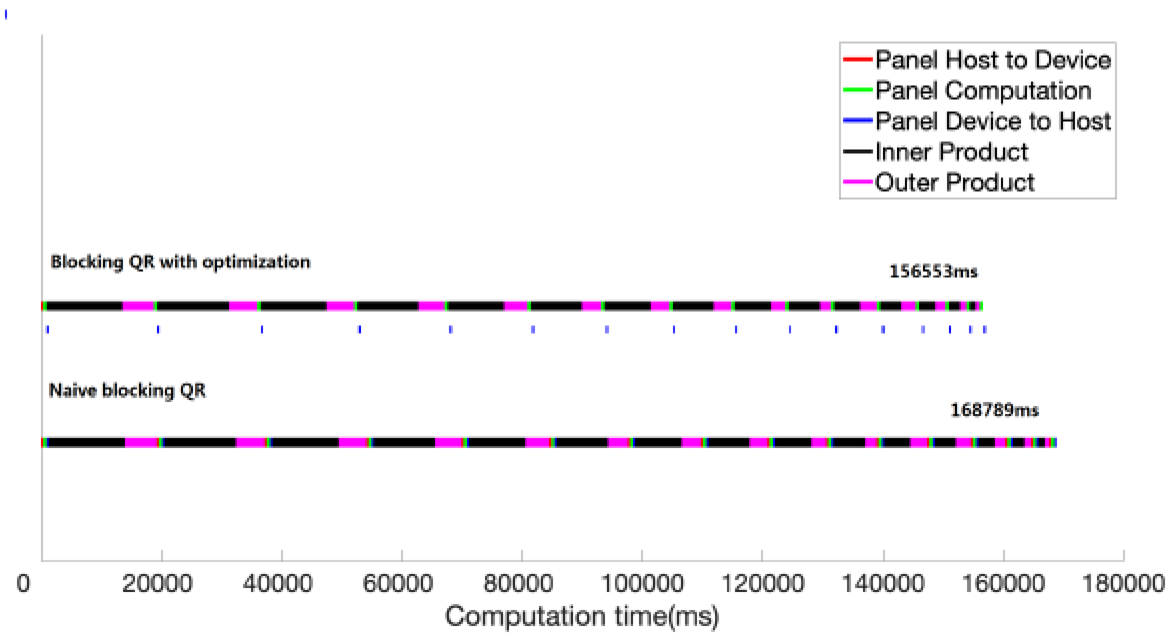


Figure 14: The timeline of computing blocking out-of-core QR, the blocksize is 8192.

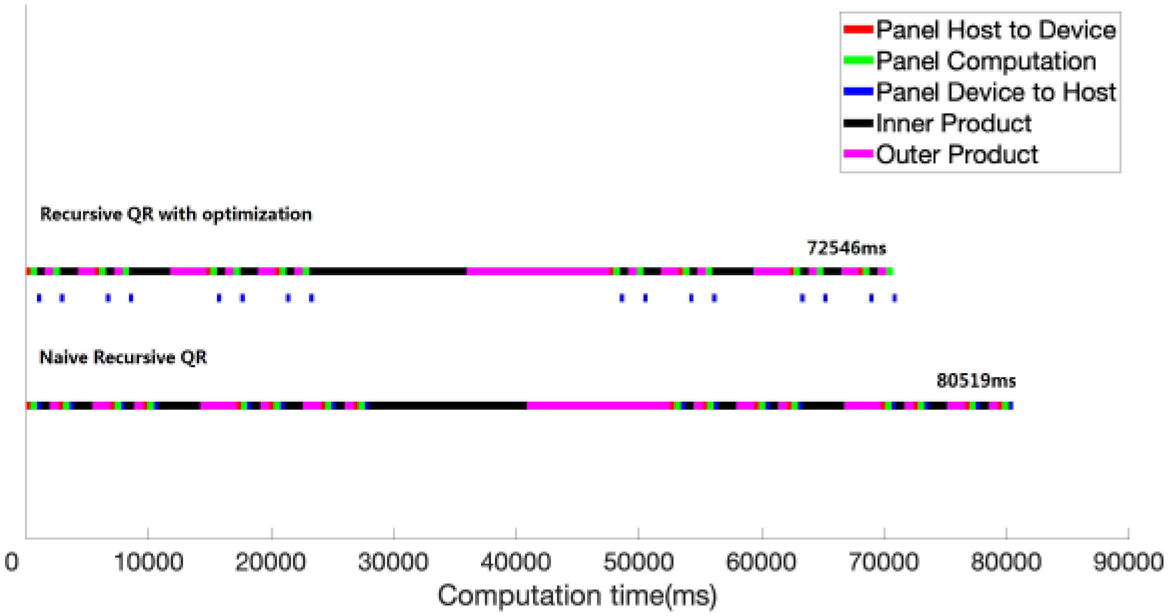


Figure 15: The timeline of computing recursive out-of-core QR, the blocksize is 8192.

Final Results: Different sizes and shapes

Partition	Recursive	Blocking
Matrix Size	65536*65536	
GEMMs	10.5s	18.9s
Panel	2.7s	2.7s
Matrix Size	262144*65536	
GEMMs	38.5s	77.0s
Panel	9.0s	9.0s

Table 4: The total time cost of GEMMs and panel of two different sizes of QR factorization (65536*65536 and 262144*65536) with blocksize 8192.

Summary

- In terms of out-of-core processing, the recursive strategy has better performance
  - Less data movement
  - Higher in-core GEMMs rate
  - Higher overlap ratio
- The higher (computation speed)/(memory capacity) ratio is, the more speedup brings by recursion
  - RTX 20,30 series have similar TensorCore computation speed, but much smaller memory

Future work

- Deploy such algorithms on A100
- Try to use the same strategy for LU and Cholesky factorization
  - The factorization steps are very similar: panel factorization and trialing matrix update using GEMMs

Thanks!