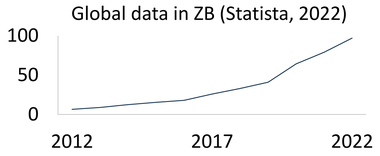
Parallel numerical linear algebra for large datasets

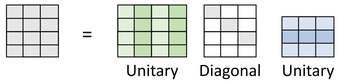
Evelyne Ringoot, Julian Samaroo, Valentin Churavy, Alan Edelman

Objective: As big data becomes widespread, we need adapted parallel algorithms

Large datasets: Total amount of **data in the world is exploding**



Singular Value Decomposition: SVD is the core algorithm used in data analysis as it reveals low-rank approximations



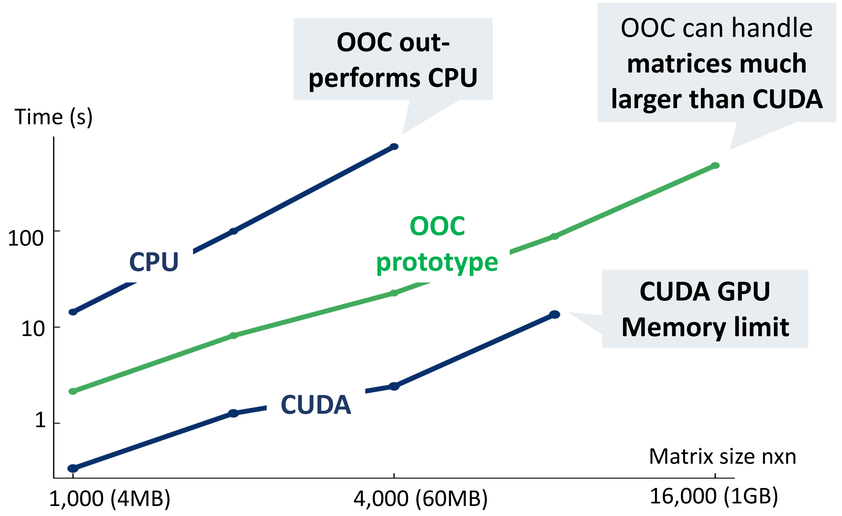
E.g. machine learning, image processing

Out-of-Core (OOC): GPU algorithms speed up calculations for large datasets but are limited in memory. **Out-of-Core algorithms** take advantage **of the speed of GPUs and the memory of CPUs.**

Julia-native: The **Julia-native implementation** allows to take advantage of all the **features of the Julia** HPC language (e.g. support for half-precision) and to make the algorithm **available to non-experts**

An Out-of-Core GPU singular value decomposition illustrates Julia capabilities for large datasets

Singular values calculation time of QR-based algorithms

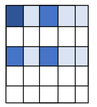


1. **Successful prototype of out-of-core GPU SVD**
   Algorithm is faster than CPU algorithms, and handles bigger matrices than CUDA
2. **Wide target user base in Julia Language**
   Syntax of Julia makes algorithm available to non-experts alike
3. **Potential for more algorithms for large datasets**
   SVD implementation shows merits applicable to other linear algebra algorithms

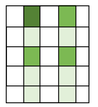
Methods: QR algorithm for Parallel Out-of-Core Block-Bidiagonalization

Algorithm: QR block-bidiagonalization (Haidar et al 2013)

Communication: Out-of-core (Kabir et al 2017)

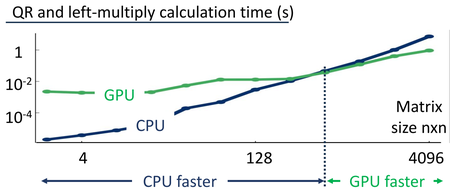


RQ sweep: top panel in GPU while iterating over rows

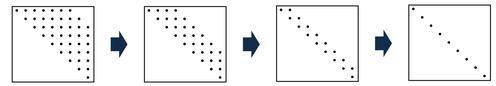


LQ sweep: left panel in GPU while iterating over columns

Block sizes: Significant QR speed-up on GPU only for **large block sizes** (>2048x2048)



Block-bulge chasing: consecutive band-width reductions of factor 2 on CPU to optimize number of flops



What's next?

Optimize prototype for latency

Maximize the **GPU computing capacity** utilization and **overlap communication and calculation**

Adapt for multi-GPU and HPC setting

Split the **parallel execution** of QR on the blocks over **different GPUs**, as required communication is limited

Expand Julia-native algorithm stack

Develop the stack of Julia algorithms for **large data numerical linear algebra capabilities**

References

[1] Statista, "Volume of data/information created, captured, copied, and consumed worldwide (...)." Statista chart, Sept. 2022.
[2] A. Haidar, J. Kurzak, and P. Luszczek, "An improved parallel singular value algorithm and its implementation for multicore hardware," in *SC '13: Proc. ICHPCNSA*, 2013
[3] K. Kabir, A. Haidar, S. Tomov, A. Bouteiller, and J. Dongarra, "A framework for out of memory svd algorithms," in Proc. *ISC HPC 2017, 2017*, Springer-Verlag, 2017
[4] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," SIAM Review, 2017.

Calculations obtained using Nvidia GeForce RTX 3050 GPU (4GB), AMD Ryzen 7 5800H Radeon Graphics (16GB)