Fault-Tolerant, Communication-Avoiding QR Factorization

Camille Coti – *coti@lipn.fr*
LIPN, CNRS UMR 7030, Université Paris 13, Sorbonne Paris Cité

Communication-Avoiding Algorithms

Communications are expensive, flops are not [2]

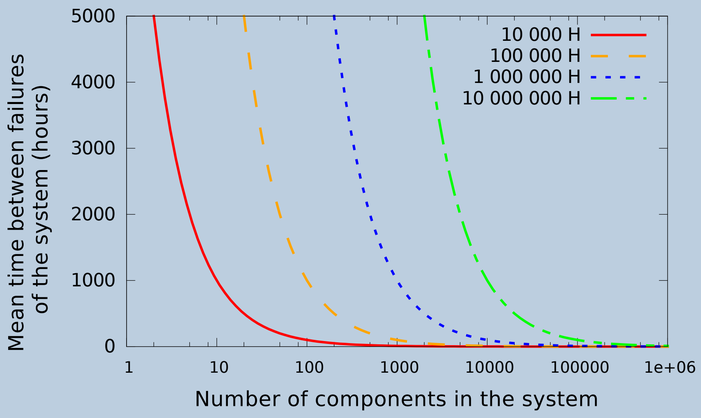
- Lower bounds for communications
- More computations

QR factorization : $A = QR$

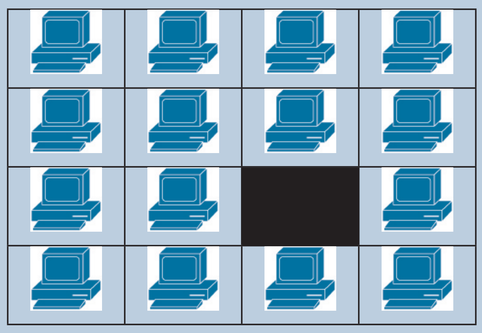
- R upper triangular
- Q orthogonal

Failures in Large-Scale Systems

$$MTBF_{total} = (\sum_{i=0}^{n-1} \frac{1}{MTBF_i})^{-1} \quad (1)$$



At large scale, failures are *unavoidable* [4]



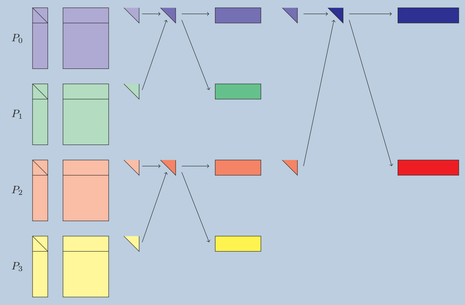
In practice: Blue Waters, MTFB = 4,2H [3]

→ Need to be tolerated!

Update of the trailing matrix

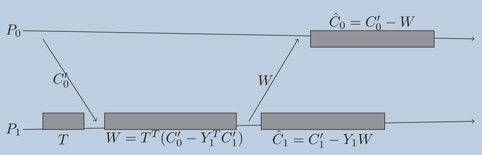
Triggered by the tree of TSQR

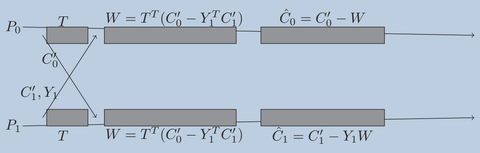
- W and Y produced by the local QR factorizations
- Broadcast on each line of the process grid



Pairwise computation

- Either process can compute T
- P_1 receives C_0 and computes W
- Both processes need W to update their trailing matrix





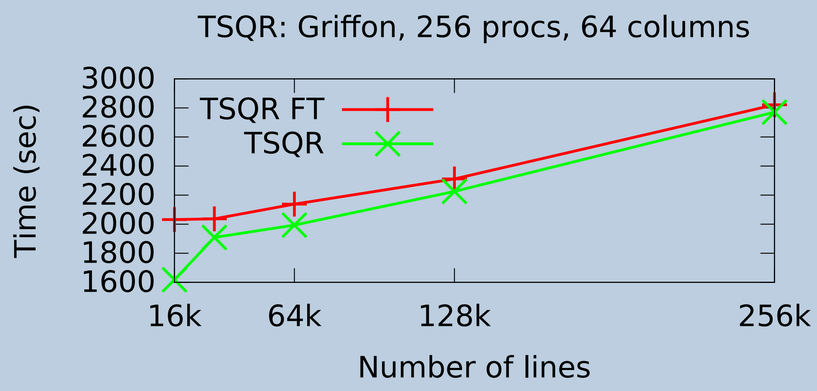
Change the order of the operations

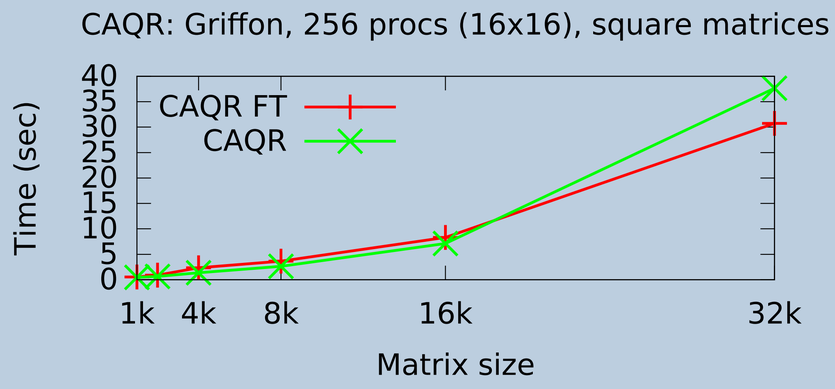
- Both processes compute T
- Exchange C_0, C_1+Y_1
- Both processes compute W and update their trailing matrix

Performance

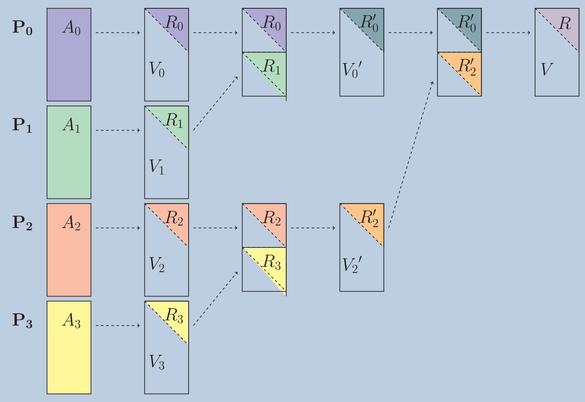
Performance evaluation: using Grid'5000

- Evaluation of the failure-free run-time overhead



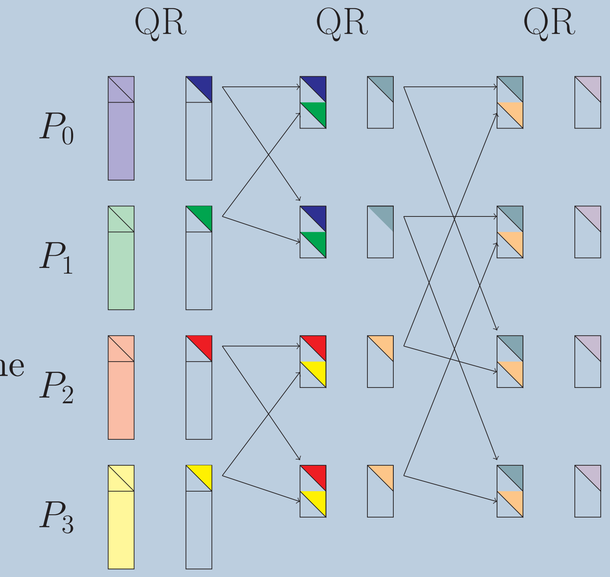


TSQR



Tree structure in the reduction

- At each step, half of the processes stop



Make them work!

- Exchange data
- Both processes hold the same data

Reconstruction of the Q matrix

Redundant computations on different processes

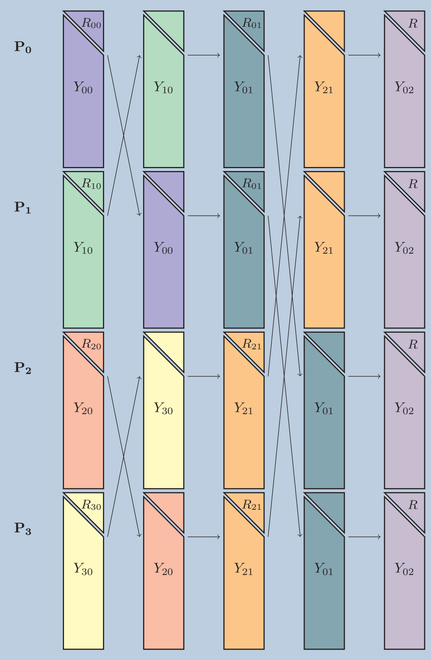
→ redundant pieces of the resulting matrix on different processes

Implicit

- The Q matrix can be represented *implicitly* by the set of Y_{ik} vectors
- Processes that compute the same QR factorizations obtain the same Y vectors: $Y, R = QR(A)$
- Redundant sets of HH vectors are accumulated

Explicit

- Can be computed by applying the orthogonal factors [1]
  - Storing these redundant factors allows reconstruction



Failure recovery

Validation at the end of each phase

- If a process is missing: obtain the data from another process

Recover

- Spawn a new process
- Obtain its data from another process

→ Minimize recomputation

Bibliography

[1] Grey Ballard, James Demmel, Laura Grigori, Mathias Jacquelin, Hong Diep Nguyen, and Edgar Solomonik. Reconstructing Householder vectors from tall-skinny QR. In *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International*, pages 1159–1170. IEEE, 2014.

[2] James Demmel, Laura Grigori, Mark Hoemmen, and Julien Langou. Communication-avoiding parallel and sequential QR factorizations. *CoRR*, abs/0806.2159, 2008.

[3] Catello Di Martino, Zbigniew Kalbarczyk, Ravishankar K Iyer, Fabio Baccanico, Joseph Fullop, and William Kramer. Lessons learned from the analysis of system failures at petascale: The case of Blue Waters. In *44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 610–621. IEEE, 2014.

[4] Daniel A. Reed, Charng da Lu, and Celso L. Mendes. Reliability challenges in large systems. *FGCS*, 22(3):293 – 302, 2006.