Application-Based Fault Tolerance for Numerical Linear Algebra at Large Scale

General Description

Fault tolerance

- Critical challenge required for large scale systems
- Difficult to predict all possible failures
- Ensure the correct termination of subroutines

Approaches to handle failures

- System-level: specific middleware
- Application-level: application handles failures

Thesis Approach

- Based on fail-stop failures and application-based approach
- Add fault-tolerant mechanisms to computation kernels

Fault Tolerance

Communication-Avoiding

- Properties to design new scalable and robust fault-tolerant algorithms

Fault tolerance cost

- Consider executions without fault-tolerance
- Measure overhead injected by fault-tolerance mechanisms
- Measure recovery procedure
- Recover executions with as little overhead as possible

Formal verification

- Fault tolerance verification with formal methods
- Reliability, robustness, correct functioning

Matrix Factorizations

Los Tres Amigos

- Part of the basic linear algebra kernel
- Reduces computational complexity of matrix operations

$$A = \begin{matrix} U \\ L \end{matrix}$$

$$A = \begin{matrix} R \\ Q \end{matrix}$$

$$A = \begin{matrix} L^T \\ L \end{matrix}$$

TS/CA Algorithms

TS algorithm

- Tall-and-skinny matrices
- Data distributed along a 1D
- TSLU, TSQR, TS-Cholesky

CA algorithm

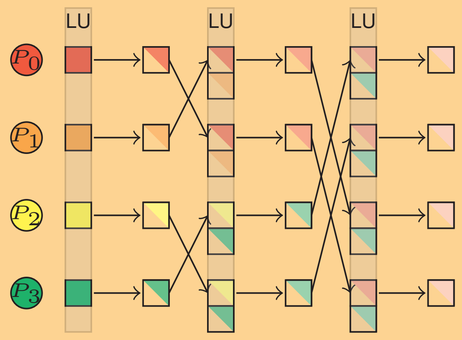
- Potencially square matrices
- Minimizes inter-process communication using a 2D grid of processes
- CALU, CAQR, CA-Cholesky

Research Proposal

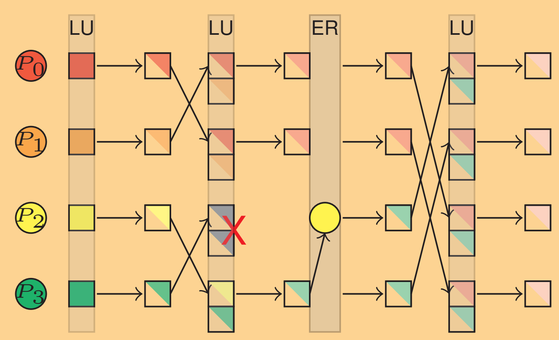
- Fault-tolerant TS/CA versions
- FT-TSLU, FT-TSQR, FT-TS-Cholesky
- FT-CALU, FT-CAQR, FT-CA-Cholesky
- Recover from crash-type errors
- Re-spawn failed processes, communication restoration
- Keep track on the intermediate results

TSLU/FT-TSLU Example

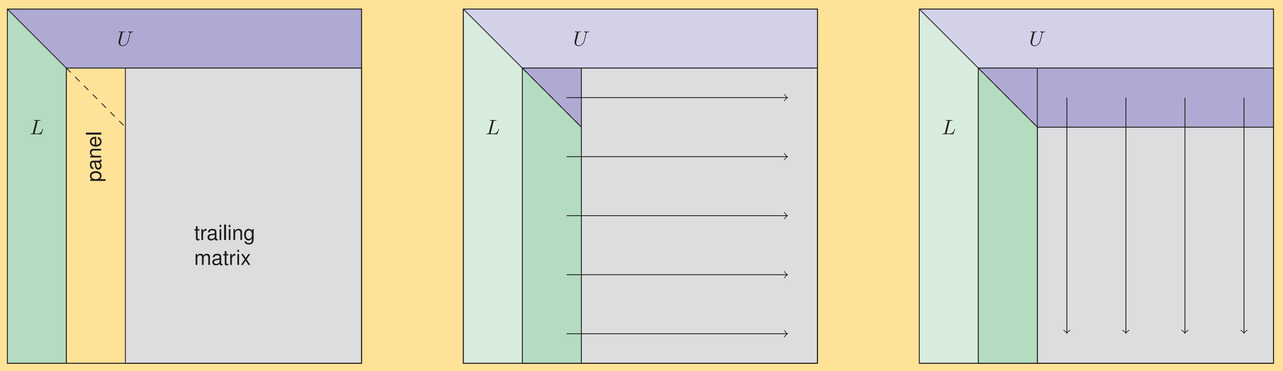
Failure-free



1st step failure and recovery



CALU/FT-CALU Example



LU Algorithms Results

Tall and Skinny Formal Model

- Already designed and implemented algorithms validating the approach
- Currently designing new fault-tolerant mechanisms for QR/Cholesky
- Formal model to prove how failures can be represented and modeled under development
- It helps in proofs design for future fault-tolerant algorithms

Camille Coti, Laure Petrucci, Daniel Alberto Torres González

Alliance Sorbonne Paris Cité