Comparative Study of Mixed-Precision and Low Rank Compression

A study on Sparse Direct Solvers - PaStiX

Brieuc Nicolas, Mathieu Faverge,
Pierre Ramet, Alicya Lisito,
Mohamed Kherraz

Summary

01

Context

- A sparse direct solver - PaStiX : https://gitlab.inria.fr/solverstack/pastix
- Resolution in double/single precision real/complex
- Arithmetic in full rank or with low rank compression methods
- Users - CalculiX - wanted (and added) Mixed-Precision to their project that uses PaStiX

Subject of the Study

- Mixed Precision was added to PaStiX
- Compare to Low Rank compression

Comparison of mixed-precision solving with regards to full and low rank :

- Time:
  - Should 2x faster than Full Rank
  - Faster than Low Rank ?
- Numerical precision:
  - More refinement needed than Full Rank
  - Better or worse initial solution than Low Rank ?
- Memory usage:
  - Mixed-precision is half the memory footprint of Full Rank
  - More or less than Low Rank ?

Algorithm 1: Refinement Pseudocode Low/Full Rank

Input: Sparse Matrix A_d, vector b_d, tol, ϵ

Output: Approximate solution x_d

$LU_d = factor(A_d, tol)$ /* Get FR/LR Factorized Matrix */

$x_d = solve(LU_d, b_d)$ /* Get Initial Solution */

$res = \| \mathrm{b}_d - A_d x_d \| / * GetInitialResidual * /$

while $res > \epsilon$ **do**

  /* Iterative Method */

  $x_d = solve(LU_d, b_d)$

  $res = \| \mathrm{b}_d - A_d x_d \|$

end

Algorithm 2: Refinement Pseudocode Mixed Precision

Input: Sparse Matrix A_d, vector b_d, ϵ

Output: Approximate solution x_d

$LU_s = factor(A_d)$ /* Get single precision Matrix

$b_s =$ (single) b_d

$x_s = solve(LU_s, b_s)$ /* Get Initial Solution */

$x_d =$ (double) x_s

$res = \parallel \mathrm{b}_d - A_d x_d \parallel / * GetInitialResidual * /$

while $res > \epsilon$ **do**

  /* Iterative Method */

  $b_s =$ (single) b_d

  $x_s = solve(LU_s, b_s)$

  $x_d =$ (double) x_s

  $res = \parallel \mathrm{b}_d - A_d x_d \parallel$

end

02

Factorization

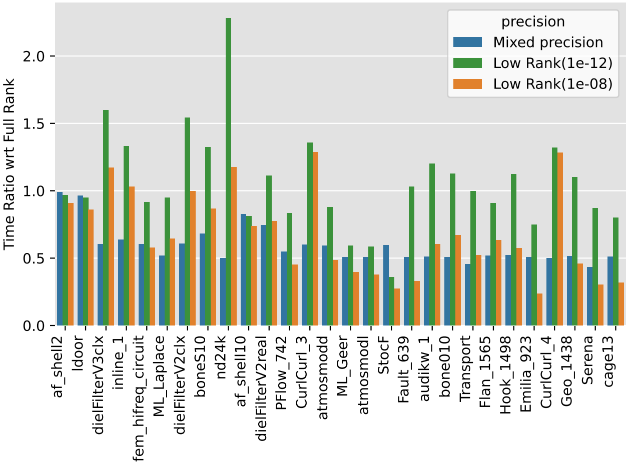
Square Matrices, from SuiteSparse

- Non-zeros: 1.5M - 110M
- Columns/Rows : 50K - 2M
- Memory (Gb) : 0.8 - 78.3
- TFlops : 0.06 - 204.9
- Facto : LU / LDL^t
- Arithmetic : Double Real/Double Complex

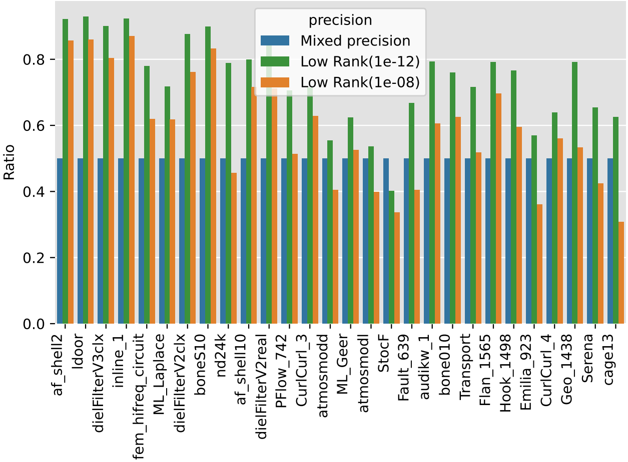
Solver

- PaStiX 6.3.2, Dynamic Multi-Threading
- 2x 16-cores processors Intel® Xeon® Gold SKL-6130 @ 2,1 GHz 92 Gb RAM
- Low Rank tolerance: $1e^{-08}$ - $1e^{-12}$
- Refinement epsilon : $1e^{-12}$

Facto Time Ratio



Facto Memory Usage ratio

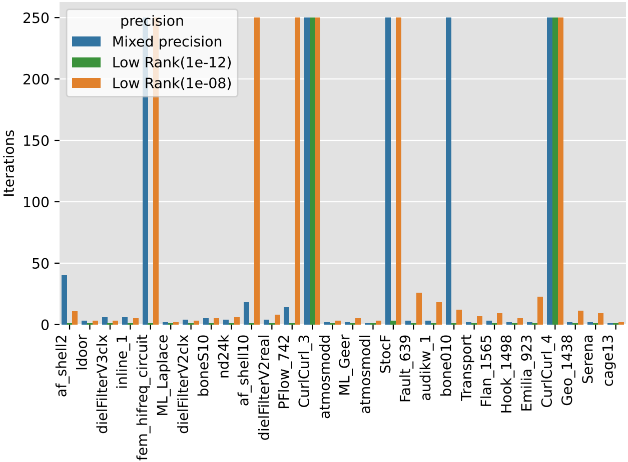


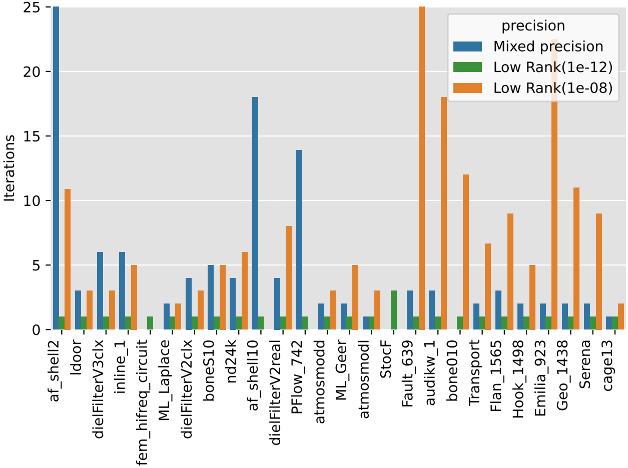
- Mixed-Precision is consistently about 2x faster than Full Rank
- Faster than Low Rank on smaller, less compressible matrices
- Slower with bigger, more compressible matrices
- LR saves more memory on Bigger matrices

03

Time to solution

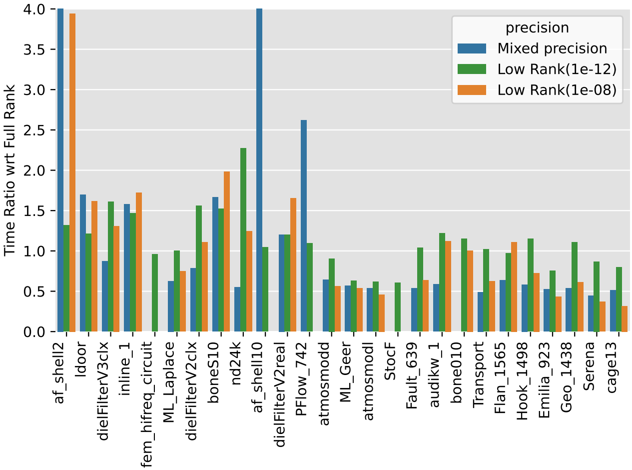
Refinement Iterations in Mixed and Low Rank





- Closeup without non-converging low rank runs
- Mixed-precision reaches a solution in less iterations

Inria



- Mixed-precision reliably halves the Time-To-Solution

Observations

- Viable alternative to Low Rank, with very simple implementation
- Depending on optimized single precision GPU kernels, could be very interesting

follow up studies

- Energy consumption measurements
- Mixed-Precision on a by-block basis