Distributed and Parallel Dense Block Linear Algebra using YML and XMP

SPPEXA workshop at Aachen

Jérôme Gurhem

CNRS/Maison de la Simulation Saclay, Université Lille 1

15th March 2018

Serge Petiton, CNRS/Maison de la Simulation Saclay, Université Lille 1

Miwako Tsuji, RIKEN Advanced Institute for Computational Science

Outline

Introduction

- Post-Petascale ($>10^{15}$ FLOPS)
- Limited compute power for a processor with only one core
- Core/processors and processors/machines increase
- Need for programming paradigms adapted to post-petascale architecture
- Example : YML with XMP

YML/XMP

YML

- Development and execution environment for parallel and distributed applications
- Already deployed on K
- Deployment and experiments possible on Poincare and ROMEO thanks to Thomas Dufaud's VM

XMP

- PGAS language based on directives
- Allows users to develop parallel programs for distributed memory systems
- Parallelization of sequential code by adding directives
- The compiler translates directives to MPI calls

YML/XMP

- XMP components for YML pass the number of processes per task and the mapping of the data on the processes

YML/XMP

Applications

- Block LU factorization (presented during last SPPEXA worshop in Tokyo)
- 3 applications to solve dense linear systems with different block methods
  - Block Gauss Elimination
  - Block Gauss-Jordan Elimination
  - Block LU factorization

YML/XMP components implemented

- Matrix - vector product
- Matrices product
- Generation and save for matrices and vectors
- Resolution of linear systems

Experiments

Tests were executed on Poincare at Maison de la Simulation, Romeo at Reims and K.

Block resolution of the linear system $Ax = b$ with Gauss, Gauss-Jordan and LU factorization

matrix size : $16\,384 \times 16\,384$
$\#\text{processes} = 1024$

Figure 1: Test parameters for each application

Blocks	Procs/task					
	16	**32**	**64**	**128**	**256**	**512**
2 × 2			x	x	x	
4 × 4	x	x	x	x	x	x
8 × 8	x	x	x	x	x	x
16 × 16	x	x	x	x	x	x

Results on Poincare@MDS and K@RIKEN for Block Gauss

For the case XMP only, the data is already deployed on the nodes although they are loaded from disk in the 1x1 block case.

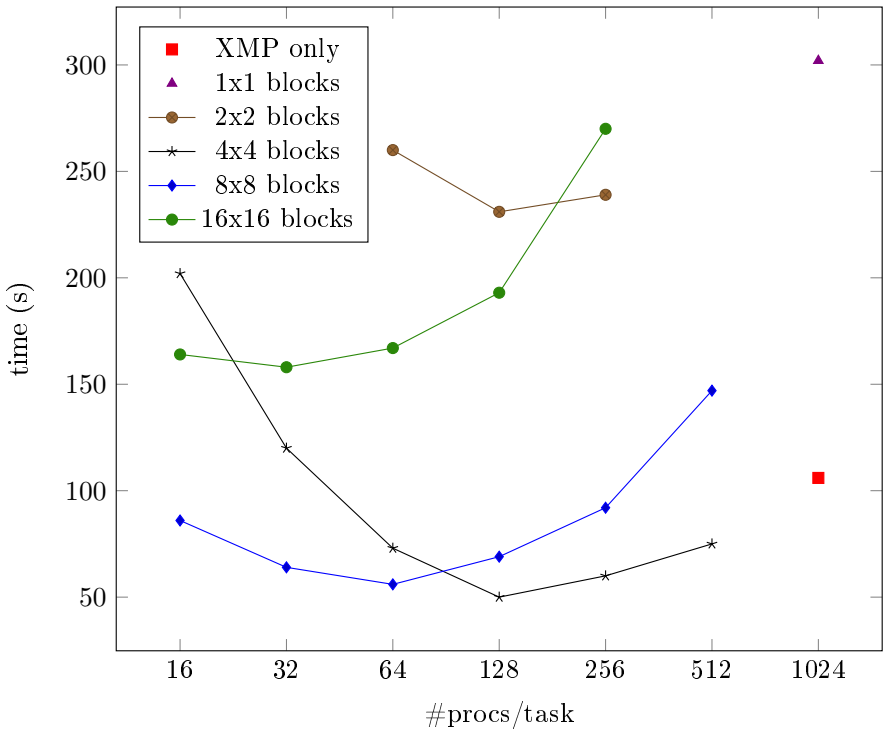

Figure 2: Block Gauss to solve linear system on Poincare

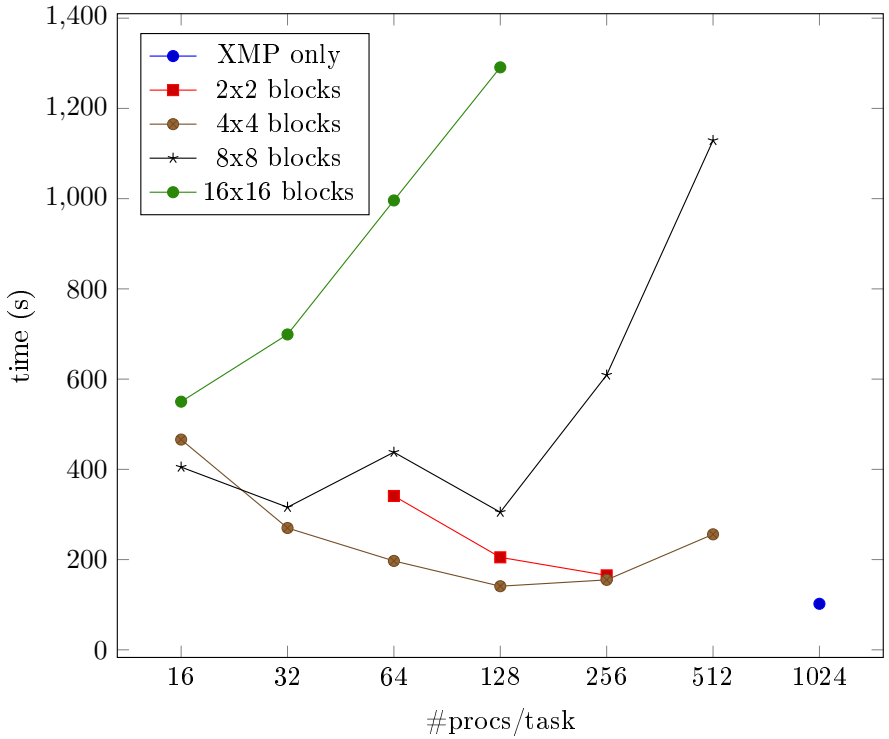

Figure 3: Block Gauss to solve linear system on K

Conclusion and future work

- Replace the YML server IO.
- YML seems to be a programming paradigm interesting to obtain great performances on current machines and especially on the new exascale machines.
- Publication of articles to show the results (one in proceedings and another one in writing)

Thank you for your attention.

Part of the results is obtained by using the K computer at the RIKEN Advanced Institute for Computational Science.