Towards exascale with the ANR-JST japanese-french project FP3C (Framework and Programming for Post-Petascale Computing)[1]

G. Antoniu[4], T. Boku[11], C. Calvin[1], P. Codognet[6], M. Daydé[5], N. Emad[7], Y. Ishikawa[10], S. Matsuoka[8], K. Nakajima[10], H. Nakashima[9], R. Namyst[2], S. Petiton[3], T. Sakurai[11], M. Sato[11]

[1]CEA/DEN/DANS/DM2S, CEA Saclay
[2]INRIA Bordeaux
[3]INRIA Saclay
[4]INRIA Rennes
[5]IRIT / CNRS – University of Toulouse
[6]Japanese –French Laboratory on Informatics
[7]PRISM / CNRS –University of Versailles
[8]Tokyo Institute of Technology
[9]University of Kyoto
[10]University of Tokyo
[11]University of Tsukuba

[1] Funded by the ANR-JST joint program under **Project ANR/JST- 2010-JTIC-003**

1 . INTRODUCTION

The Japanese-french project FP3C targets the post-petascale and the next generation of exascale systems. Its goal is to study software technologies, languages and programming models to achieve the performance promised by post-petascale computing.

The project consortium regroups some of the HPC key players in Japan and in France:

- Universities of Kyoto, Tokyo and Tsukuba, Tokyo Institute of Technology, Japanese-French Laboratory on Informatics
- CEA Saclay, CNRS (IRIT and PRISM), INRIA (Bordeaux, Rennes and Saclay),

with a strong connection with RIKEN AICS at Kobe in Japan.

The ability to efficiently program and exploit these future high performance systems is considered as a strategical issue by all research national agencies worldwide.

The exascale computers are expected to a large-scale highly hierarchical architecture with many-core processsors possibly mixed with accelerators with up to millions of cores and threads. As a consequence, existing operating and runtime systems, languages, programming paradigms and parallel algorithms would have, at best, to be adapted, and may become often obsolete.

Exploiting these ultra large-scale parallel systems will require runtime systems allowing the management of huge amount of distributed data, minimizing the energy consumption, and exhibiting fault resilient properties. In addition to these features, accelerating technology, based on GPGPU and many-core processors, are also crucial issues for post-petascale computing systems.

Efficient exploitation (programing, execution) of large scale systems is an important challenge. The hierarchical nature of these architecture and the scalability issues (parallelization at different levels: intra / inter computational nodes composed of many core processors and/or accelerators) complexifies the algorithm design. Existing programming framework and runtime systems together with new approaches should be examined, experimented and evaluated. Benchmarks and libraries adapted to the post-petascale systems will have to be proposed.

The FP3C project aims at defining a framework for high performance computing on the road to exascale. We cover most of the issues related to the emergence of post-petascale computing: from high level languages to efficient exploitation of the underlying architecture at the runtime level or using specific extensions for accelerators. Our approach is illustrated with experiments on a set of numerical codes (including numerical libraries) that can be seen as a benchmark. One of the goal of the project is to proceed to experiments on some of the most recent computers available in Europe (e.g. CURIE Computer at TGCC/France) and in Japan (e.g. K Computer at Kobe).

In this presentation, we proceed to an overview of the various aspects of the FP3C project.

2. PROGRAMMING FRAMEWORK

Our programming framework mainly combines the use of three basic approaches: YML, XscalableMP and StarPU.

YML is a high level graph description language developed at PRISM Versailles (http://yml.prism.uvsq.fr/). Initially introduced for grid and peer-to-peer systems, it is based on a component approach. It provides portability by hiding the details of the underlying middleware. It is used for expressing in an easy way the concurrency of the tasks of a dependency graph.

XscalableMP (see http://www.xcalablemp.org/) is a directive based language extension for scalable and performance-aware parallel programming. The project is lead by university of Tsukuba and the XscalableMP working group involves several other partners (universities of Tokyo, Kyoto Kyushu; RIKEN, NIFS, AICS, JAXA, JAMSTEC/ES, Fujitsu, NEC, Hitachi). It generates C plus MPI code from a C code with directives expressing data and work mapping. It is used within the YMP components to exploit a lower grain parallelism.

An example of XscalableMP (XMP) is given below:

```
#pragma xmp nodes p(4)
#pragma xmp template t(0:7)
#pragma xmp distribute t(block) onto p
int a[8];
#pragma xmp align a[i] with t(i)
int main(){
#pragma xmp loop on t(i)
        for(i=0;i<8;i++)
                a[i] = i;
```

StarPU developed in the Runtime INRIA team at Bordeaux (see http://runtime.bordeaux.inria.fr/StarPU/) is an unified runtime system for heterogeneous architectures. It allows programmers to take advantage of the available CPUs and GPUs without modifying their codes.

The FP3C approach is then to propose a prototype for GPU/CPU work sharing, using XMP - dev (an extension of XMP for accelerating devices) as a parallel programming framework for distributed memory system, and StarPU as a runtime system to support GPU/CPU task management. This provides a high level programming model based on XMP-dev for GPU utilization coupled with StarPU for easy programming on GPU/CPU co-working (work sharing).

The Kyoto Group is investigating of a framework to describe application codes in the form of local-view kernels, tightly collaborating with TiTech (Tokyo Institute of Technology) Group in order to extend their DSL for GPU-oriented stencil computation to make it applicable to wider range of applications and platforms including multi-core/ multi-socket clusters.

We also proceed to evaluation of a BlobSeer-WAN-HGMDS prototype. Blobseer (see http://blobseer.gforge.inria.fr) provides a large-scale distributed service capable of storing binary objects in order of TBytes which is of particular interest for post-petascale systems.

3. BENCHMARK AND NUMERICAL ALGORITHMS

Several numerical algorithms and applications have been selected (some of these are used as a benchmark) to illustrate the main issues of the FP3C programming framework. Experiments on clusters of multicore and GPU both in France and in Japan have been performed including executions on Tsubame 2, T2K and K computers in Japan and CURIE and GRID'5000 in France.

FP3C considers several applications:

- eigenvalue computation for molecular orbital computation and reactor physics applications;
- nano-material simulation;
- Finite Element Method for ill-conditioned solid mechanics problems;
- 3D finite-volume based simulation code for groundwater flow problems through heterogeneous porous media ;
- real space density functional theory (Hermitian standard eigenvalue computation);
- nuclear physic computation;
- superconductivity.

These applications make use of typical numerical kernels such as the finite element method, the solution of sparse linear systems and sparse eigenvalue problems. Several issues related to these kernels are carefully studied within the project including their interfacing with YML and XMP:

- Sparse eigensolvers: ERAM, MERAM solvers (see [1]), parallel eigensolver for Semi-sparse matrices (see [10]), hybrid SSM / MERAMshifted block Krylov linear solver, experiments on GPU for some of the eigensolvers (see for example [14]);
- Linear solvers :
    - Parallel preconditioners: ILU-type ([2], [3]),) and scalable multigrid preconditioning ([4], [5]);
    - Experiments with the MUMPS sparse direct solver (http://mumps.enseeiht.fr) in some aforementioned eigensolvers and in preconditioners;
    - Block Cimmino hybrid solver ([9]);
- Auto-tuning for linear and eigen solvers: GMRES by considering restart and orthogonalization issues, tuning and auto-tuning of some of the main components of Krylov methods[15], introduction of the notion of smart-tuning.
- Experiments with FEM on GPU ([6], [7],[8]).
- Parallel algorithms for combinatorial optimization: use of meta-heuristics (local search, Tabu search, …) with application to artificial intelligence, SAT, … (see [11], [12], [13]).

The GRID-TLSE web site (http://gridtlse.org/) is used as a test-bed for some of the experiments using MUMPS.

4. CONCLUSION

The FP3C is an attempt to define a programming framework for post-petascale computers. The various approaches (programming models and APIs) will be incorporated into the benchmark we have defined in order to perform experiments. The last year of the project (2013-2014), the integration of the different software systems and tools will be performed simultaneously with tutorial and demos.

We will describe some of the aim issues and results of the FP3C project at the conference.

REFERENCES

[1] N. Emad, S. Petiton, and G. Edjlali. Multiple explicitly restarted arnoldi method for solving large eigenproblems. *SIAM Journal on scientific computing SJSC, Volume 27,Number*, 27:253{277, 2005}.

[2] Hayashi, M., Ohshima, S. and Nakajima, K.: Parallel ILU Preconditioner based on Extended Hierarchical Interface Decomposition for Heterogeneous Environments, IPSJ SIG Technical Reports (in Japanese), IPSJ-HPC11130005 (2011).

[3] Hayashi, M. and Nakajima, K.: OpenMP/MPI Hybrid Parallel ILU(k) Preconditioner for FEM Based on Extended Hierarchical Interface Decomposition for Multicore Clusters, submitted to 10th International Meeting on High-Performance Computing for Computational Science (VECPAR 2012), Kobe, Japan (2012).

[4] Nakajima, K.: Parallel Preconditioning Methods for Iterative Solvers in Multi-Core Era, Mathematical foundation and development of algorithms for scientific computing, RIMS Kokyuroku 1773, Research Institute for Mathematical Science, Kyoto University, 1-10 (in Japanese) (2011).

[5] Nakajima, K.: New Strategy for Coarse Grid Solvers in Parallel Multigrid Methods using OpenMP/MPI Hybrid Programming Models, ACM Proceedings of PPoPP/PMAM 2012, (2012) (in press)

[6] Ohshima, S., Hayashi, M., Katagiri, T. and Nakajima, K.: Implementation and Evaluation of 3D Finite Element Method for CUDA, IPSJ SIG Technical Reports (in Japanese), IPSJ-HPC11129020 (2011)

[7] Ohshima, S., Hayashi, M., Katagiri, T. and Nakajima, K.: Implementation of Matrix Assembly in 3D Finite Element Method for CUDA, IPSJ SIG Technical Reports (in Japanese), IPSJ-HPC11130011 (2011)

[8] Ohshima, S., Hayashi, M., Katagiri, T., Nakajima, K.: Implementation and Evaluation of 3D Finite Element Method Application for CUDA, submitted to 10th International Meeting on High-Performance Computing for Computational Science (VECPAR 2012), Kobe, Japan (2012)

[9] Arioli, M., Duff, I.S., Noailles, J., Ruiz, D.: A block projection method for sparse matrices. In: SIAM Journal on Scientific and Statistical Computing (1992) 47–70

[10] T. Sakurai and H. Sugiura, A projection method for generalized eigenvalue problems, J. Comput. Appl. Math., 159 (2003), pp. 119–128.
[11] Y. Caniou and P. Codognet, "Communication in Parallel Algorithms for Constraint-Based Local Search," in IEEE Workshop on new trends in Parallel Computing and Optimization (PCO'11), held in conjunction with IPDPS 2011, Anchorage, USA, IEEE Press, 2011.
[12] Y. Caniou, P. Codognet, D. Diaz, and S. Abreu, "Experiments in Parallel Constraint based Local Search", in EvoCOP'11, 11th European Conference on Evolutionary Computation in Combinatorial Optimisation, Torino, Italy, LNCS series, Springer Verlag, 2011.
[12] D. Diaz, F. Richoux, P. Codognet, Y. Caniou, and S. Abreu "Constraint-Based Local Search for the Costas Array Problem", proceedings of LION6, 6th Learning and Intelligent OptimizatioN Conference, Paris, France, January 2012, Lecture Notes in Computer Science 7219, Springer Verlag 2012.
[13]
Aquilanti, P., Petiton, S., and Calandra, H.: Parallel GMRES Incomplete Orthogonalization Auto-Tuning, Procedia CS 4, 2246-2256 (2011)
[14] J. Dubois, C. Calvin and S. Petiton, Accelerating the explicitly restarted arnoldi method with GPUs and autotuned matrix vector product. SIAM Journal on Scientific Computing, Volume 33, Issue 5, Pages 3010-3019 (2011).
[15] S. Petiton, C. Calvin, J. Dubois and F. Boillod-Cerneux, "Auto-tuned Hybrid Asynchronous Krylov iterative eigensolver on petascale computer.", Juin 2012, PMAA 2012, Londres, Royaume-Uni.
[16] J. Dubois, S. Petiton and C. Calvin "Improving Scalability with Asynchronous Hybrid Methods for non-Hermitian Eigenproblems", ICCS 2011, Tsukuba, Japan. Juin 2011.
[17] C. Calvin, J. Dubois, "Performances of Krylov Solvers for Reactor Physics Simulation on Petascale Architectures", SIAM Conference on Computational Science and Engineering 2013, Boston, USA.
[18] C. Calvin, L. Drummond, F. Boillod-Cerneux, J. Dubois, G. Ndongo Eboum, "Auto-tuning and Smart-tuning Approaches for Efficient Krylov Solvers on Petascale Architectures", SIAM Conference on Computational Science and Engineering 2013, Boston, USA.
[19] S. Petiton, C. Calvin, J. Dubois, F. Boillod-Cerneux, "Auto-Tuned Hybrid Asynchronous Krylov Iterative Eigensolvers on Petascale Supercomputers", SIAM Annual Meeting 2012, Minneapolis, USA.
[20] C. Calvin, N. Emad, S. Petiton, J. Dubois and M. Dandouna, "MERAM for neutron physics applications using YML environment on post petascale heterogeneous architecture", SIAM Conference on Applied Linear Algebra (SIAM ALA 2012), Juin 2012, Valence, Espagne.
[21] S. Petiton, C. Calvin, J. Dubois, L. Decobert, R. Abouchane, P-Y. Aqulianti, F. Boillod-Cerneux "Krylov Subspace and Incomplete Orthogonalization Auto-tuning Algorithms for GMRES on GPU Accelerated Platforms", 15th SIAM Conference on Parallel Processing for Scientific Computing (Savannah 2012).
[22] C. Calvin, S. Petiton, J. Dubois and F. Boillod-Cerneux "An Eigenvalue Solver using a Linear Algebra Framework for Multi-core and Accelerated Petascale Supercomputers", 15th SIAM Conference on Parallel Processing for Scientific Computing (Savannah 2012).