

PROJECT PROPOSAL PHPC-2016*A High Performance Implementation of a 2D Elliptic Equation Solver:
the Poisson's Equation Case at Scale*

Principal investigator (PI)	Vincent Keller
Institution	EPFL-SCITAS
Address	Station 1, CH-1015 LAUSANNE
Involved researchers	Only PI
Date of submission	May 9, 2016
Expected end of project	July 1, 2016
Target machine	Mount Everest
Proposed acronym	POISSON

Abstract

Solving the Poisson equation has been largely investigated during the past decades. In this case, we present a fully functional parallel implementation of a 2D stencil-based scheme that solves the Poisson equation. Our application is coded in C and uses the industrial standards MPI and OpenMP in its hybrid implementation. Several f functions will be calculated.

1 Scientific Background

A parallel iterative finite difference method for solving the 2D elliptic PDE Poisson's equation on a distributed system using Message Passing Interface (MPI) and OpenMP is presented. This method is based on a domain decomposition where the global 2D domain is divided into multiple sub-domains using horizontal axis. The number of subdomains is specified by the number of processes. The Poisson's equation is solved by explicit iterative schemes. The global error is shared by all processes.

The Poisson's equation is:

$$\Delta\varphi = f \quad (1)$$

that is in 2D

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \varphi(x, y) = f(x, y) \quad (2)$$

with $\partial\Omega = 0$

The equation is discretized on a 2D rectangular domain of size NxN . We use centered finite differences to approximate (2). This will lead to a linear system $Ax = b$. Using a Jacobi iterative method to solve it and expressing φ^{k+1} as a function of φ^k , we obtain

$$\varphi_{i,j}^{k+1} = \frac{1}{4} \left(\varphi_{i+1,j}^k + \varphi_{i-1,j}^k + \varphi_{i,j+1}^k + \varphi_{i,j-1}^k - f_{i,j} h^2 \right) \quad (3)$$



Figure 1: Solution of the Poisson Equation with $N = 512$, $l2 = 0.005$, $iteration = 1018$

where $h = \frac{1}{N+1}$

For our preliminary results, we set $f_{i,j} = -2\pi^2 \sin(\pi i h) * \sin(\pi j h)$

2 Implementations

The application is implemented in C [1]. We have investigated three implementations of the same solver :

- A shared memory version using the OpenMP paradigm [2]
- A distributed memory version using the MPI library [3]
- An hybrid version using both MPI and OpenMP where the checkpoint files are written in binary using MPI-IO.

The code has been fully debugged using the `gdb` debugger. The `Valgrind` tool has been used to remove all the memory leaks.

2.1 Optimizations

Beside the standard sequential optimizations (loop reordering, vectorization, etc.), we put a special effort to tackle the main bottleneck of the beta-versions of the MPI and hybrid implementations : the I/O part. The current production version implements a BMP-file writer in parallel using MPI-IO.

3 Prior results

We ran our production version on the Bellatrix cluster at EPFL. This machine presents the following characteristics [4] :

- 424 compute nodes each with 2 Intel 8-cores Sandy-Bridge CPU running at 2.2 GHz, with 32 GB of memory

- Infiniband QDR 2:1 connectivity
- GPFS filesystem
- Total peak performance : 119 TF

3.1 Strong scaling

!! FIXME !!

3.2 Weak scaling

!! FIXME !!

4 Resources budget

In order to fulfill the requirements of our project, we present hereafter the resource budget.

4.1 Computing Power

!! FIXME !!

4.2 Raw storage

!! FIXME !!

4.3 Grand Total

Total number of requested cores	!! FIXME !!
Minimum total memory	!! FIXME !!
Maximum total memory	!! FIXME !!
Temporary disk space for a single run	!! FIXME !!
Permanent disk space for the entire project	!! FIXME !!
Communications	!! FIXME !!
License	own code (BSD)
Code publicly available ?	!! FIXME !!
Library requirements	!! FIXME !!
Architectures where code ran	!! FIXME !!

5 Scientific outcome

This work will (probably) lead to a good grade for its author. Additionnaly, the present project proposal example can be reused in the future for a real submission at EPFL or in a larger high performance computing center such as the Swiss National Supercomputing Center CSCS in Lugano.

References

- [1] Kernighan, Brian W. and Ritchie, Dennis M., *The C Programming Language Second Edition*, Prentice-Hall, Inc., 1988
- [2] Dagum L. and Menon R., *OpenMP: An Industry-Standard API for Shared-Memory Programming*, IEEE Computational Science & Engineering, Volume 5 Issue 1, pp 46-55, January 1998
- [3] The MPI Forum, *MPI: A Message-Passing Interface Standard*, Technical Report, 1994
- [4] Scientific IT and Application Support, <http://scitas.epfl.ch>, 2015

Appendices

A Vincent Keller, PhD – CV

Vincent Keller is a computer scientist born in Lausanne, Switzerland. He received his Master degree in Computer Science from the University of Geneva (Switzerland) in 2004. From 2004 to 2005, he holds a full-time researcher position at the University Hospital of Geneva (HUG). He was involved at HUG on simulating blood flows in cerebral aneurysms using real geometries constructed from 3D X-rays tomography slices. The numerical method used was Lattice-Boltzman Method (LBM). He received his PhD degree in 2008 from the cole Polytechnique Fdrale de Lausanne (EPFL) in the HPCN and HPC Grids fields. His supervisor was Dr. Ralf Gruber. He developed a low-level application-oriented monitoring system (VAMOS) and the Resource Broker of the anos (Intelligent ApplicationN-Oriented System) framework prototype. In 2009, he holds a full-time researcher position at Fraunhofer SCAI (University of Bonn) in Germany. He was the Scientific Coordinator of the IANOS project. In 2010, he was application analyst in the IT service at University of Zrich. Between September 2010 and September 2014, he was application analyst in the Center for Advanced Modeling Science (CADMOS) for the EPFL&University of Lausanne projects. Since September 2014, he is full member of the SCITAS (Scientific IT and Application Support) team led by Prof. Jan Hesthaven and Dr. Vittoria Rezzonico. Dr. Vincent Keller is the author or coauthor of more than 30 peer-reviewed papers and one book (HPC@GreenIT, Springer Verlag, Heidelberg, Germany, October 2010). His research interests are in HPC applications analysis, Grid and cluster computing and energy efficiency of large computing ecosystems.

Areas of expertise :

- High Performance Computing, Computational Science and Engineering
- Parallel computing
- Optimization, Parallelization, Parallel I/O, Scientific Libraries
- Scientific Applications and Performance Analysis, Algorithms
- Energy Efficiency in Large Computing Ecosystems