A Study of Sparse Matrix Methods on New Hardware: Advances and Challenges

Athanasios Fevgas, Department of Electrical & Computer Engineering, University of Thessaly, Volos, Greece

Konstantis Daloukas, Department of Electrical & Computer Engineering, University of Thessaly, Volos, Greece

Panagiota Tsompanopoulou, Department of Electrical & Computer Engineering, University of Thessaly, Volos, Greece

Panayiotis Bozanis, Department of Electrical & Computer Engineering, University of Thessaly, Volos, Greece

ABSTRACT

Modeling of numerous scientific and engineering problems, such as multi-physic problems and analysis of electrical power systems, amounts to the solution of large scale linear systems. The main characteristics of such systems are the large sparsity ratio and the large number of unknowns that can reach thousands or even millions of equations. As a result, efficient solution of sparse large-scale linear systems is of great importance in order to enable analysis of such problems. Direct and iterative algorithms are the prevalent methods for solution of linear systems. Advances in computer hardware provide new challenges and capabilities for sparse solvers. The authors present a comprehensive evaluation of some, state of the art, sparse methods (direct and iterative) using modern computing platforms, aiming to determine the performance boundaries of each solver on different hardware infrastructures. By identifying the potential performance bottlenecks of out-of-core direct methods, the authors present a series of optimizations that increase their efficiency on flash-based systems.

Keywords: Direct Solvers, Flash, GPU Computing, Iterative Solvers, Multi-Physics, Scientific Computing, Smart Grid, Solid State NVM, Sparse Matrix Algorithms, SSD

1. INTRODUCTION

Modeling of real world problems, such as multiphysics problems or modern power systems require the efficient solution of systems of linear equations, whose main characteristic is the large and sparse system matrix. Especially for the power systems, operation functions such as power flow, state estimation and contingency analysis, involve exhaustive sparse matrix calculations with near real time demands. The vast majority of methods that are utilized in the solution of large-scale sparse systems are categorized either as direct, or iterative. Direct methods are based

DOI: 10.4018/IJMSTR.2015070101
on the factorization of the coefficient matrix and have been widely utilized in the past, mainly because of their robustness. They also have the property of reusability of factorization results for time-evolving problems with fixed time-step. Their main disadvantage is the lack of scalability with respect of the dimension of the linear system, which makes such methods prohibitively expensive for very large-scale problems, in both execution time and memory requirements.

On the other hand, iterative methods are more computationally - and memory - efficient as they involve only inner products and matrix vector products. As a result, they constitute a better alternative for large sparse linear systems in many respects. In addition, they possess a kind of reusability property for time-dependent simulation, where the solution of the last time-step can be utilized as a good initial guess for the next time-step and greatly accelerate the convergence rate of the iterative method. These features make iterative methods much more suitable for the solution of large-scale multiphysics problems.

Another feature that a solution algorithm must employ is the exploitation of the computational capabilities of the underlying hardware. Parallel architectures such as multi-core processors or GPUs offer a large amount of computational resources. However, harnessing their potentials is not a trivial task. Direct algorithms dispose limited parallelism, thus their mapping onto parallel architectures does not provide the expected results. On the contrary, iterative algorithms are highly parallelizable, thus exploiting parallel architectures can greatly accelerate the solution of the underlying system. Recently, several algorithms that utilize external storage (out-of-core) have been proposed to alleviate the increased memory requirements of direct methods. These algorithms are designed to efficiently access factor data stored in secondary storage, and therefore enabling the solution of large-scale linear systems.

The performance of each solving method is tightly coupled with the underlying hardware, resulting to a variety of algorithms aiming to exploit specific architectures’ computational resources. Under this perspective, employing out-of-core algorithms along with the latest storage hardware, such as flash based solid state drives, can provide a highly advantage. Therefore, utilizing flash based storages enables the solution of very large-scale linear systems with decreased overhead, as they provide orders of magnitude better I/O performance than traditional magnetic disks.

Researching on challenges and opportunities for sparse solvers on new hardware technologies, we conducted a quantitative performance study focusing on the solution of large-scale, sparse systems. These linear systems are mainly derived from power systems problems or multiphysics simulations. We organize our work as follows: Section 2 presents an introduction to sparse linear systems mathematical background along with a short description of the problems and the challenges concern power systems and multiphysics simulations respectively. Several existing and upcoming hardware technologies are described in Section 3, while the sparse solvers utilized in this paper are presented in Section 4. The experiments and the results are in Section 5. Our efforts to identify a design and implementation roadmap for out of-core solvers that exploit flash storage is presented in Section 6, and finally conclusions and future work are appeared in 7.

2. BACKGROUND

2.1. Sparse Linear Systems

Typically, simulation of multidomain/multiphysics problems amounts to the solution of linear systems with the form $Ax = b$, where the coefficient matrix $A$ is very large and sparse. For such type of systems, direct or iterative methods can be employed to calculate the vector $x$. Direct methods are robust and conclude in a predictable number of steps. Although they do not
Continuous User Authentication Based on Keystroke Dynamics through Neural Network Committee Machines
www.igi-global.com/chapter/continuous-user-authentication-based-keystroke/59674?camid=4v1a