

Potential of General Purpose Graphic Processing Unit for Energy Management System

Jean-Charles Tournier, ABB Corporate Research, USA & Switzerland

Vaibhav Donde, ABB Corporate Research, USA & Switzerland

Zhao Li, ABB Corporate Research, USA & Switzerland

Martin Naef, ABB Corporate Research, USA & Switzerland

ABSTRACT

This paper investigates the potential of General Purpose Graphic Processing Unit (GPGPU) for the server and HMI parts of Energy Management System (EMS). The HMI investigation focuses on the applicability and performance improvement of GPGPU for scattered data interpolation algorithms typically used to visually represent the overall state of a power network. The server side investigation focuses on fine grain parallelization of EMS applications by targeting the sparse linear solver. The different performance evaluations show the high potential of GPGPU for the HMI part with a speedup factor up to 100 at the cost of acceptable approximations, while the benefit on the server side varies from a speedup factor of up to 300 to 0 depending on the application.

Keywords: Energy Management System, General Purpose Graphic Processing Unit, HMI, Scatter Data Interpolation Algorithm, Sparse Linear System

INTRODUCTION

Energy Management System (EMS) aims at monitoring, controlling and optimizing the generation and transmission of electrical power through the execution of a variety of applications such as state estimator, contingency analysis, optimal power flow analysis, unit commitment, etc. The EMS architecture is generally made of two distinct parts: a server where the ap-

plications are executed and a human-machine interface (HMI) where the results of the applications are displayed to the network controller.

With the evolution of the power system network into a smarter grid, both parts of EMS will soon reach their computational limits. Indeed, EMS are required (*a*) to handle larger data set from additional measurement devices such as PMU (Phasor Measurement Unit) or smart meter; (*b*) to address larger, possibly continental sized, networks so as to minimize the effects of approximations that should otherwise be done

DOI: 10.4018/jdst.2012040105

at the boundaries of interconnected network areas; (c) to increase situational awareness by executing and reporting applications results at a faster rate, as well as to obtain energy market pricing information as accurately as possible to boost efficiency of such markets.

While many different approaches are envisioned for EMS to deal with larger and more complex power system network, i.e., HPC server (Huang, 2008), cluster (Pourreza, 2010), dedicated hardware (Shi, 2008), this paper investigates the potential of General Purpose Graphic Processing Unit (GPGPU) for the server and HMI parts of EMS. Compared to other approaches, the main advantage of GPGPU is to provide cost efficient commodity hardware with the equivalent computational power of HPC servers from the previous decade.

The HMI investigation focuses on the applicability and performance improvement of GPGPU for scattered data interpolation algorithms typically used to visually represent the overall state of a power network. The approach chosen is to compare several interpolation algorithms implemented on GPGPU with their equivalent implementations on CPU using its full capabilities, i.e., multithreading and SIMD extension (SSE). The evaluation performed on multiple matrices shows that while the CPU implementations are outperformed by the GPGPU ones, their performances can already be acceptable when the full potential of the CPU architecture is used.

Due to the nature of GPGPU, i.e., implementing a Single Instruction Multiple Data (SIMD) programming paradigm, the server side investigation focuses on fine grain parallelization, i.e., sparse linear solver, rather than a coarse grain one, e.g., different applications executed in parallel. Sparse linear solvers are at the heart of most EMS application and their performances greatly determine the overall applications performances. Due to the limited success on implementing sparse direct solvers on GPGPU (Kerr, 2009) and the promising result in implementing iterative solvers on SIMD architectures (Huang, 2008), we are focusing on implementing iterative solvers

for power system applications and comparing their performances against highly performant sparse direct solver libraries, namely SPQR (Davis, 2011) and PARDISO (Schenk, 2008). The performance measurements executed on a typical set of matrices found in various EMS applications, e.g., Jacobian and Gain matrix from state estimation and DC power flow analysis, and representing a wide span of system size ranging from 400 to 40,000 buses, show the limit of GPGPU for this type of applications.

This paper is an extended version of (Tournier, 2011). The remainder of the paper is organized as follows: the following section introduces the main concepts of GPGPU from an architecture and programmability point of view. The next two sections report the performances of using GPGPU for respectively the HMI and server part of EMS, while a last section concludes the paper.

GENERAL PURPOSE GRAPHIC PROCESSING UNIT

From an architecture point of view, both latest CPU and GPGPU package several cores into one die, sharing the same memory channel. For this type of computation architecture, having a high-speed memory bandwidth, through which data can be quickly fed into computational cores, is necessary to keep the computation structure efficient. One of the major differences between CPU and GPGPU is that CPU has a larger cache system and stronger control capabilities; while GPGPU has a more significant number of arithmetic logic unit (ALUs).

As more transistors are devoted to data processing rather than caching and flow control, the computation power offered by GPGPU is significantly larger than the one offered by the latest CPU. Nowadays, the latest version of GPGPU (GTX480) from NVIDIA offers 576 GFLOPS computing power, 20 times more than the latest CPU (i.e., Intel Westmere having six cores). Increasing the computing power by assembling many computational cores into one die, however, escalates memory I/O requests

9 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/article/potential-general-purpose-graphic-processing/66058

Related Content

High-Throughput GRID Computing for Life Sciences

Giulia De Sario, Angelica Tulipano, Giacinto Donvito, Giorgio Maggi and Andreas Gisel (2012). *Grid and Cloud Computing: Concepts, Methodologies, Tools and Applications* (pp. 821-840).

www.irma-international.org/chapter/high-throughput-grid-computing-life/64517/

Using Policy-Based Management for Privacy-Enhancing Data Access and Usage Control in Grid Environments

Wolfgang Hommel (2009). *International Journal of Grid and High Performance Computing* (pp. 15-29).

www.irma-international.org/article/using-policy-based-management-privacy/3963/

Grid-Enabling Applications with JGRIM

Cristian Mateos, Alejandro Zunino and Marcelo Campo (2009). *International Journal of Grid and High Performance Computing* (pp. 52-72).

www.irma-international.org/article/grid-enabling-applications-jgrim/3970/

Implementation of a DES Environment

Gyorgy Lipovszki and Istvan Molnar (2010). *Handbook of Research on Discrete Event Simulation Environments: Technologies and Applications* (pp. 284-316).

www.irma-international.org/chapter/implementation-des-environment/38266/

Performance Evaluation of Reactive Routing in Mobile Grid Environment

L. Shrivastava, G. S. Tomar and S. S. Bhadauria (2011). *International Journal of Grid and High Performance Computing* (pp. 45-53).

www.irma-international.org/article/performance-evaluation-reactive-routing-mobile/56355/