Temperature-aware Asynchoronized Jacobi Solver on Heterogeneous Mobile Devices

[a] University of the Moon, , Earth, , ,

Abstract

Example abstract for the Nuclear Physics B journal. Here you provide a brief summary of the research and the results.

Keywords: keyword 1, keyword 2, keyword 3, keyword 4

1. Introduction

Mobile devices are dominating our daily life. More and more applications are developed for mobile devices. And many of them, especially those related to graphics and vision, involve with tasks that can be finally formulated as energy optimization problems. High performance is critical for interactive feedback in many of these applications. Thus, formulating these problems linearly are especially appreciated, as they can be solved rather quickly. Even though, the limited computing power of the CPU on mobile devices still cannot meet the need for instant solving of linear system in many applications.

On the other side, multi-core architecture has become an irreversible trend, due to issues in thermal/power constraints, reliability issues and design complexity etc. Most current mobile devices are equipped with multiprocessor system-on-chips (MPSoC). Especially, heterogeneous multi-core architecture, which consists of different core types, can offer significant advantages in performance, power, area, and delay, compared with homogeneous counterpart (consisting a set of identical cores). Besides, heterogeneous multi-core are perfect fit for the dark silicon regime (the increasing power density on chip prevents all the cores to be switched on at the same time) as only the cores suited for an application need to be switched on. The heterogeneity of MPSoC can be either in performance or in function. Taking the Samsung Exynos 7420 MPSoC as an example, which is designed with the ARM big.LITTLE architecture (a classical heterogeneous architecture in performance), the MPSoC integrates four high-performance out-of-order ARM Cortex-A57 cores (big cores) and four low-power in-oerder ARM Cortex-A53 cores (little cores). It also has a GPU, Mali-T760 MP8, which is functionally different from the CPUs (see Figure **??** for details on the architecture of MPSoC).

These mobile processors have the potential to deliver high-performance if all the available resources can be harnessed by the software, in a way to use the cores that are most power efficient for the current computing need without negatively impacting the performance. However, designing algorithms to run efficiently on such heterogeneous MPSoC is not easy. We need to select appropriate core type for each compute-intensive kernel, translate and optimize the corresponding kernel in a high-level language to an implementation suitable for the selected core type; carefully handling data sharing among processors to alleviate overheads such as synchronization, data transfer and mapping; workload should be well balanced among these heterogeneous cores with accurate performance estimation of computing tasks on different types of cores.

In this paper, we present a ....Although as a stand-alone solver it is not so efficient , Jacobi and other stationary iterations are often used as a building block for complex algorithms such as multigrid methods. Jacobi iteration has a low computation complexity and is very suitable for parallel implementation.

2. Related work

2.1. Numerical Solver for CG

The pervasive usage of numerical solver for linear or nonlinear problems in computer graphics domain has promote the research on efficient and accurate solver. Earlier works devoted to highly efficient CPU solvers Toledo (2003) or high-level CPU solvers Shearer and Wolfe (1985). Later on, researchers began to shift their sights to the acceleration of general or basic blocks on GPUs, such as sparse matrix conjugate gradient solver Bolz et al. (2003) and multigrid solver Goodnight et al. (2005). Spontaneously, GPU based non-linear solvers are developed to overcome the performance limitations of their CPU counterparts, tailored for specific applications such as deformable mesh tracking Zollhöfer et al. (2014), shape-from-shading Zollhöfer et al. (2015), computational imaging Heide et al. (2014), model-to-frame camera tracking Nießner et al. (2013), face reconstruction Thies et al. (2016) and cloth simulation Wu et al. (2022) etc. To avoid the tedious and suffering process of hand-writting GPU solvers, domain-specific languages DeVito et al. (2017); Mara et al. (2021) are designed to allow users specify the high-level energy description only, with various solver generated automatically. And thus users can further explore tradeoffs in numerical precision, matrix-free methods and solver approaches.

This paper targets on parallelization of jacobi iteration on heterogeneous processors of mobile devices to gain performance improvements by overcoming the challenges in load imbalance, inefficient memory access, serialization and excessive synchronization. We investigate how an asynchronous version