Advances in Engineering Software 104 (2017) 38–50

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Research paper

Parallelisation of an interactive lattice-Boltzmann method on an Android-powered mobile device

Adrian R.G. Harwood[a,*], Alistair J. Revell[b]

[a] Research Associate, School of Mechanical, Aerospace and Civil Engineering, The University of Manchester, Sackville Street, M1 3BB, United Kingdom
[b] Senior Lecturer, School of Mechanical, Aerospace and Civil Engineering, The University of Manchester, Sackville Street, M1 3BB, United Kingdom

ARTICLE INFO

Article history:
Received 16 July 2016
Revised 18 November 2016
Accepted 22 November 2016
Available online 6 December 2016

Keywords:
Android
Mobile computing
Interactive simulation
Lattice Boltzmann method
Java concurrency

ABSTRACT

Engineering simulation is essential to modern engineering design, although it is often a computationally demanding activity which can require powerful computer systems to conduct a study. Traditionally the remit of large desktop workstations or off-site computational facilities, potential is now emerging for *mobile computation*, whereby the unique characteristics of portable devices are harnessed to provide a novel means of engineering simulation. Possible use cases include emergency service assistance, teaching environments, augmented reality or indeed any such case where large computational resources are unavailable and a system prediction is needed. This is particularly relevant if the required accuracy of a calculation is relatively low, such as cases where only an intuitive result is required. In such cases the computational resources offered by modern mobile devices may already be adequate. This paper proceeds to discuss further the possibilities that modern mobile devices might offer to engineering simulation and describes some initial developments in this direction. We focus on the development of an interactive fluid flow solver employing the lattice Boltzmann method, and investigate both task-based and thread-based parallel implementations. The latter is more traditional for high performance computing across many cores while the former, native to Android, is more simple to implement and returns a slightly higher performance. The performance of both saturates when the number of threads/tasks equal three on a quad-core device. Execution time is improved by a further 20% by implementing the kernel in C++ and cross-compiling using the Android NDK.

1. Introduction

Modelling and simulation is an integral part of modern engineering as it allows the user to improve their understanding of physical scenarios and complex systems. Depending on the context, this knowledge may be used in a variety of ways; e.g., to inform a design decision, to aid in education of key concepts, or to identify level of risk for a given scenario. Due, in part, to both improved understanding and perpetually increasing computational power, we have become accustomed to a regular increase in the accuracy of these simulations. The calculations themselves are generally conducted on high-end computer facilities either housed locally or via a high-bandwidth interconnect to a high performance computing (HPC) facility. Due to the nature of the software and the skills required to manage and process the data, there are well defined processes in place to assure the quality of the simulation results. For these reasons, and others, the running of computer simulations tends to fall under the remit of an experienced engineer and are typically orchestrated from a desk-based computer.

However, in the era of big data and pervasive computing, it is no longer impractical to envisage the coordination and indeed the running of simulations via or on-board a mobile device. There is no question that mobile devices, be that tablet computers or mobile phones, are lighter, more portable and often cheaper than laptops, desktops and servers currently being used for engineering simulations. Having simulation results presented directly to a individual using this platform can allow qualitative analysis to be performed in situations where such information has previously been unavailable. For example, in emergency scenario analysis, the mobile device may be used to capture surroundings using the built-in camera and contaminant sources using the touch screen. A local simulation is then used to provide the user with an immediate safe route of navigation. Alternatively, an interactive wind tunnel can be effectively given to a class of students to enhance education and learning. Mobile devices may also be given to physicians and used

* Corresponding author.
E-mail address: adrian.harwood@manchester.ac.uk (A.R.G. Harwood).

http://dx.doi.org/10.1016/j.advengsoft.2016.11.005

in combination with patient derived imagery to provide improved diagnostic information at the point of care [1].

Over recent years, the prevalence of desktop computing has reduced, and the use of laptop and tablet devices has grown to fill this gap. Leveraging the many core graphics processing units (GPUs) typically available on mobile devices can deliver a significant boost in processing power. However, it is unlikely that a single device will reach the power available in current HPC clusters in the near future. Instead of matching the accuracy of "conventional" modelling and simulation methods, which is likely to always require significant computing power, there is arguably a role for a faster simulation tool, that trades accuracy for speed, in order to attain a level of human-interactivity. Particularly where used to complement and enhance human decision making, or even to provide fast approximations to inform an automated decision making system as one of many environmental data available to the system.

In order to assess the suitability of mobile platforms for performing local, interactive engineering simulation, this article reports the development of two different parallel design patterns for performing interactive, grid-based fluid dynamics simulations on an Android-powered mobile device. The Android operating system is selected as the development platform, due to the availability and affordability of suitable hardware and the fact that it currently has the largest market share for mobile devices [2].

Although mobile devices often feature both a GPU and a CPU, the present study explores only use of the CPU. The development of custom software for mobile GPU is not yet widely supported and is left for a future publication. Our simulations use the lattice-Boltzmann method (LBM) to simulate flow physics [3] and is introduced in the next section.

The primary aim of the present study is to propose different approaches to interactive flow simulation using LBM on an Android device. This includes the implementation and cross-comparison of several candidate frameworks in order to assess the potential for mobile devices, either used alone or for multi-device parallelism. The completion of these aims provides baseline data and a design template on which other types of interactive engineering simulation on a range of devices may be built.

2. Use of mobile devices for engineering simulation

A survey of device ownership in the US in 2015 [2] revealed that 68% of the US population owned a smartphone and 45% owned a tablet. Globally, smartphone ownership hit 1000 million in 2012 and is set to exceed 2500 million by 2020. Ownership is spread across all continents with South Korea leading the way, where 88% of the population own a smartphone. Ownership in so-called *advanced economies*, including the US and much of Europe, is approximately 68% on average [4].

Modern mobile devices are designed to include a multi-core CPU and a GPU, providing similar versatility to a desktop computer. The computational power of these chips has increased approximately ten-fold since 2009 [5] and available RAM has also increased with more than 3 GB typical of current high-end Samsung Android devices (Fig. 1). Current engineering workstations may have 16 cores and 64 GB RAM with which to perform a local simulation – a 4x increase in CPU cores and a 16x increase in memory. Furthermore, due to active cooling mechanisms on desktop computers, clock speeds are often much higher increasing computing power further.

Halpern et al. [5] show that power consumption for mobile chips has increased although further increases in power consumption are yielding less of a gain in performance causing a power saturation at about 1.5W. Instead, the most recent smartphone design has increased the number of cores rather than increasing the power per core. One may conclude that mobile hardware development is hence limited by its power-conserving motivation.

However, in defence of mobile devices, the wide-spread ownership of mobile devices, combined with the clear, albeit restricted, increases in power and capability, makes the platform a potential candidate for running *smaller scale* engineering simulations on site without reliance on external resources or connectivity. Although an individual device may not be able to offer the power of an HPC facility, simulations could be performed on a network of devices connected by a local network implemented via Bluetooth or Wi-Fi Direct.

High-end HPC facilities, at present, typically offer $\mathcal{O}(10^5)$ cores and $\mathcal{O}(10^3)$GB of RAM according to the TOP500 list. However, in practice, these resources are shared amongst many users with individual jobs using much smaller allocations. Access to such facilities is also generally restricted. Theoretically, if all 2×10^9 smartphones globally are assumed to be quad-core with 2GB RAM (c.2013), a global P2P *smartphone computer* could offer 8×10^9 cores with 4×10^9 GB of memory. This is purely hypothetical but illustrates the compute potential for even the mobile devices in a single office block or city.

In light of hardware limitations for individual mobile devices, it is expected that in order to run a simulation locally, there will be a trade-off between the level of model complexity (and hence simulation accuracy) and the speed with which a result can be obtained. However, mobile platforms have the potential to provide sufficient computing power for rapid simulation to an *acceptable and situation-appropriate* degree of accuracy. This can only be realised with the development of a suitable framework for engineering simulation in this context.

2.1. Integration with existing infrastructure

In our increasingly connected world, mobile devices also have the option to off-load tasks with a high resource demand to more suitable systems [6]. In the case of engineering simulation, mobile devices may provide input data such as local wind speed and direction, measured structural loads or geometry and materials all recorded locally on the device, to a remote HPC facility which performs a potentially demanding calculation using these system data. A data-reduced result may then be returned for the user to inspect. It may also be possible to perform some part of the simulation locally as coarse approximation to the problem physics while simultaneously performing a more detailed analysis remotely which may be viewed or incorporated into the local platform at a later time Fig. 2.

However, presently HPC facilities are expensive to build and maintain and access is typically restricted. Furthermore, network connectivity is not available in every location and may also suffer from reduced bandwidth or unreliability. Common interconnect between HPC facilities and external terminals is of the order of 1Gb/s which gives a maximum theoretical throughput of 125MB/s. The fastest mobile data connections in the UK at present use the LTE-A (4G+) standard and will theoretically support such a transfer rate [7]. However, this service is, at present, only available in select areas and at a premium subscription cost to a user. Typically, connection speeds may be as low as 12 MB/s depending on the infrastructure available. A sensible alternative may therefore be to develop approaches to performing the calculation locally on one or more available devices [8].