



Cloud GPU-based simulations for SQUAREMR



George Kantasis^a, Christos G. Xanthis^{b,a}, Kostas Haris^a, Einar Heiberg^{b,c}, Anthony H. Aletras^{a,b,*}

^aLaboratory of Computing and Medical Informatics, School of Medicine, Aristotle University of Thessaloniki, Thessaloniki, Greece

^bLund Cardiac MR Group, Department of Clinical Physiology and Nuclear Medicine, Lund University, Skåne University Hospital in Lund, Lund, Sweden

^cDepartment of Biomedical Engineering, Faculty of Engineering, Lund University, Lund, Sweden

ARTICLE INFO

Article history:

Received 2 August 2016

Revised 10 November 2016

Accepted 16 November 2016

Available online 17 November 2016

Keywords:

Magnetic Resonance Imaging

Relaxometry

Simulations

Cloud

ABSTRACT

Quantitative Magnetic Resonance Imaging (MRI) is a research tool, used more and more in clinical practice, as it provides objective information with respect to the tissues being imaged. Pixel-wise T_1 quantification (T_1 mapping) of the myocardium is one such application with diagnostic significance. A number of mapping sequences have been developed for myocardial T_1 mapping with a wide range in terms of measurement accuracy and precision. Furthermore, measurement results obtained with these pulse sequences are affected by errors introduced by the particular acquisition parameters used. SQUAREMR is a new method which has the potential of improving the accuracy of these mapping sequences through the use of massively parallel simulations on Graphical Processing Units (GPUs) by taking into account different acquisition parameter sets. This method has been shown to be effective in myocardial T_1 mapping; however, execution times may exceed 30 min which is prohibitively long for clinical applications. The purpose of this study was to accelerate the construction of SQUAREMR's multi-parametric database to more clinically acceptable levels.

The aim of this study was to develop a cloud-based cluster in order to distribute the computational load to several GPU-enabled nodes and accelerate SQUAREMR. This would accommodate high demands for computational resources without the need for major upfront equipment investment. Moreover, the parameter space explored by the simulations was optimized in order to reduce the computational load without compromising the T_1 estimates compared to a non-optimized parameter space approach.

A cloud-based cluster with 16 nodes resulted in a speedup of up to 13.5 times compared to a single-node execution. Finally, the optimized parameter set approach allowed for an execution time of 28 s using the 16-node cluster, without compromising the T_1 estimates by more than 10 ms.

The developed cloud-based cluster and optimization of the parameter set reduced the execution time of the simulations involved in constructing the SQUAREMR multi-parametric database thus bringing SQUAREMR's applicability within time frames that would be likely acceptable in the clinic.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Quantitative Magnetic Resonance Imaging (MRI) is an important research and clinical tool that provides objective data of the imaged tissue by measuring MR parameters such as spin-lattice relaxation time (T_1) and spin-spin relaxation time (T_2). Pixel-wise T_1 quantification (T_1 mapping) may be used to measure the extracellular volume (ECV) of the myocardium [1] while T_2 mapping may be used to detect edematous myocardium [2]. To date, multiple methods have been proposed for myocardial T_1 mapping, including the Modified Look-Locker Inversion recovery (MOLLI)

[3], the shortened breath-hold MOLLI (sh-MOLLI) [4], the Saturation recovery Single-shot Acquisition (SASHA) [5] and the inversion-saturation hybrid method (SAPPHIRE) [6]. Among them, MOLLI has become a relatively mature method for T_1 mapping and is used in the clinic. Although MOLLI yields highly reproducible results with precision, it systematically underestimates the T_1 values. Moreover, several studies have already demonstrated how various protocol parameters affect its performance [7,8].

Recently, SQUAREMR, a novel method for improving T_1 measurements with the aid of massively parallel MRI simulations, was proposed [9]. SQUAREMR employs comprehensive MR simulations of clinical pulse sequences by means of a large set of simulation elements (isochromats) that possess specific tissue characteristics, such as relaxation constants. A database of all possible simulated signals is then constructed to cover a selected

* Corresponding author at: Laboratory of Computing and Medical Informatics, School of Medicine, Aristotle University of Thessaloniki, Thessaloniki, Greece.

E-mail address: aletras@auth.gr (A.H. Aletras).

range of tissue characteristics. Unlike curve-fitting methods, SQUAREMR finds the database entry where the acquired signal best matches the simulated signal (via the least sum of squares difference) and then assigns the tissue characteristics of that database entry to the acquired signal. This can be done on a pixel-by-pixel basis (image reconstruction). SQUAREMR has been shown to improve the T_1 estimates in MOLLI pulse sequences and has demonstrated consistent variance across different MOLLI acquisition schemes for a wide range of T_1 and T_2 values.

In brief, from an implementation standpoint, the previously presented SQUAREMR platform is shown in Fig. 1 in block diagram format. The dashed lines represent the conventional approach of T_1 mapping while the solid lines represent the SQUAREMR approach. Each time an examination was performed on the MRI scanner, the identical parameter set was sent to the simulation platform which performed the simulations and returned the database of expected signal intensities. Because of the large simulation parameter space and the fact that some of the parameters were selected during the examination by the MR operator (e.g. Field of View, Flip Angle, Slice Thickness) while some other parameters were acquired directly from the patient (e.g. Heart Rate), it was unrealistic to pre-compute all possible databases. This raised a computational challenge as the simulations had to be performed in a post hoc manner.

SQUAREMR is based on MRISIMUL [10], a comprehensive Bloch equation simulation platform that utilizes the high performance of graphics processing units (GPUs) to accelerate the computationally intensive task of MR simulations. However, despite using high performance computing (multi-GPU server equipped with four TESLA C2075 cards), the increased computational load of SQUAREMR results in execution times that may exceed 30 min [9]. This is mainly due to the incorporation of some realistic aspects of the MR experiment (such as excitation slice profile) and the large population of spins covering a wide range of T_1 and T_2 values. Last, additional simulation parameters, such as B_0 inhomogeneity, could further increase the computational load.

MRISIMUL works by simulating a pulse sequence in discrete tissue simulation elements called isochromats, where each represents a collection of spins with the same characteristics (T_1 , T_2 etc.). The continuous waveforms of a pulse sequence (G_x , G_y , G_z , and RF) are represented digitally in MRISIMUL by a series of discrete samples i.e. “pulse sequence samples”. Each pulse sequence sample represents the waveforms G_x , G_y , G_z , and RF at discrete time intervals dt . Each pulse sequence sample is sequentially applied to every isochromat using the Bloch equations in order to simulate the evolution of the magnetization vectors. (Note: in previous reports of MRISIMUL the term “time steps” was used instead of “pulse sequence samples”).

Over the past few years, cloud computing has gained widespread acceptance as a storage, computation and networking solution to resource demanding problems in business and science. Cloud computing is a recent architecture where a centralized commodity service provides computer resources to its users. These resources include data storage, computing power, application software and network services. Cloud computing allows for rapid scaling of computer clusters in response to fluctuating, unpredictable and burst workload demands while retaining high coherence and low management cost. Unlike other conventional computer architectures, such as local cluster setups of multiple nodes, there is little to no upfront investment whereas its rapid and seamless scalability allows for timely response to bursts of computational needs. In the field of healthcare, the prospects of the integration of cloud computing [11] and its application in medical imaging systems [12], as well as the security and privacy concerns arising from such usage [13], have been explored. In the field of MRI, cloud-based approaches are used to deal with computationally intensive problems such as anatomical landmark detection [14], structure contouring [15], reconstruction [16], volume rendering [17]. Preliminary results for MRI simulations have also been presented [18].

In this study, a cloud-based GPU approach for creating the SQUAREMR database is presented. The specific aim was to improve the execution time of SQUAREMR by optimizing the simulation parameters and by using a cloud-based cluster of GPU-enabled nodes. It was hypothesized that the optimization of the simulation parameter space would improve SQUAREMR performance without compromising the T_1 estimation accuracy by more than 10 ms from the original results. Moreover, it was hypothesized that a cloud implementation utilizing up to 16 GPU-equipped nodes would speed up the database creation to about half a minute, which we considered to be appropriate for future implementation in the clinic.

2. Methods

2.1. Parameter space optimization

The SQUAREMR parameter space was defined as a discrete three dimensional space consisting of T_1 values, T_2 values and the number of isochromats along the slice direction. The number of database entries was directly proportional to the number of T_1 and T_2 values used, whereas the number of isochromats along the slice direction affected the simulation of the excitation slice profile. The range of T_1 values was selected between realistic values of 200 ms and 1600 ms, whereas the range of T_2 values was selected between 20 ms and 250 ms. Different step values resulted either in

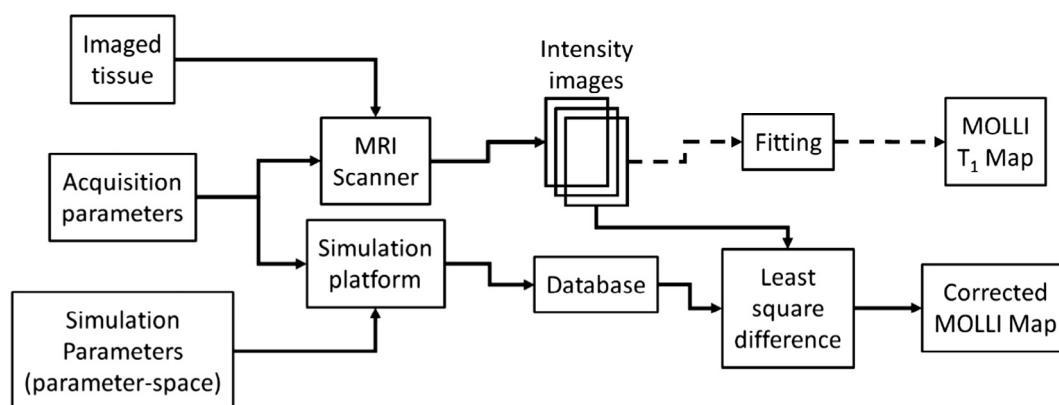


Fig. 1. Block diagram of the SQUAREMR platform. The dashed line indicates the conventional MOLLI workflow.

Download English Version:

<https://daneshyari.com/en/article/5404687>

Download Persian Version:

<https://daneshyari.com/article/5404687>

[Daneshyari.com](https://daneshyari.com)