Journal of Computational Science 22 (2017) 15–25

Contents lists available at ScienceDirect

Journal of Computational Science

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

Multiscale computing in the exascale era

Saad Alowayyed[a,b], Derek Groen[c], Peter V. Coveney[d], Alfons G. Hoekstra[a,e,*]

[a] Computational Science Lab, Institute for Informatics, University of Amsterdam, The Netherlands
[b] King Abdulaziz City for Science and Technology (KACST), Riyadh, Saudi Arabia
[c] Department of Computer Science, Brunel University London, United Kingdom
[d] Centre for Computational Science, University College London, United Kingdom
[e] ITMO University, Saint Petersburg, Russia

ARTICLE INFO

Article history:
Received 7 November 2016
Received in revised form 31 May 2017
Accepted 6 July 2017
Available online 23 July 2017

Keywords:
Multiscale modelling
Multiscale computing
High performance computing
Exascale

ABSTRACT

We expect that multiscale simulations will be one of the main high performance computing workloads in the exascale era. We propose multiscale computing patterns as a generic vehicle to realise load balanced, fault tolerant and energy aware high performance multiscale computing. Multiscale computing patterns should lead to a separation of concerns, whereby application developers can compose multiscale models and execute multiscale simulations, while pattern software realises optimized, fault tolerant and energy aware multiscale computing. We introduce three multiscale computing patterns, present an example of the extreme scaling pattern, and discuss our vision of how this may shape multiscale computing in the exascale era.

1. Introduction

Science is at its most powerful when it can not only convincingly explain the processes at work in natural phenomena but is able to predict what will occur before it does so. Predictions of real world events, such as weather forecasting, when a cement will set, the occurrence of an earthquake or what medical intervention to perform in order to save a person's life, all require the bringing together of substantial quantities of data together with the performance of one or more likely many high-fidelity four (three space and one time) dimensional simulations before the event in question occurs. Such forms of calculation are among the most demanding known in computational science, as they need to be done rapidly, accurately, precisely and reliably, including quantification of the uncertainties associated with them. They are also *multiscale* in nature, as their accuracy and reliability depend on the correct representation of processes taking place on several length and time scales. Only now, as we move toward the exascale era in high performance computing (HPC) can we expect to be able to tackle such problems effectively and, eventually, in a routine manner.

Indeed, multiscale phenomena are everywhere around us [1–7]. If we study the origin and evolution of the universe [8] or properties of materials [9–13], if we try to understand health and disease [3,14–21] or develop fusion as a potential energy source of the future [22], in all these cases and many more we find that processes on quite disparate length and time scales interact in strong and non-linear ways. In short, multiscale modelling is ubiquitous and progress in most of these cases is determined by our ability to design and implement multiscale models of the particular systems under study [1,6,23].

The sheer complexity of such multiscale phenomena still limits our capability to perform high-fidelity simulations that accurately and reliably predict the behaviour of a given system in all situations. Capturing even a few of those coupled processes in a multiscale simulation quickly reaches the limits of contemporary high performance computing at the petascale.

That the importance of multiscale modelling in many domains of science and engineering is still increasing is clearly demonstrated in numerous publications; see, for example, [1,24]. Therefore, we must anticipate that multiscale simulations will become an increasingly important form of scientific application on high end computing resources, necessitating the development of sustainable and reusable solutions for such emerging applications, that is, *generic* algorithms for multiscale computing. As we move into the exascale performance era we need to drastically change the way we use HPC for simulation based sciences [25].

For example, on current resources we can simulate composite nanomaterials using tens of millions of atoms, where the interaction potentials rest in turn on electronic structure and atomistic

[*] Corresponding author at: Computational Science Lab, Institute for Informatics, University of Amsterdam, The Netherlands.
E-mail address: a.g.hoekstra@uva.nl (A.G. Hoekstra).

http://dx.doi.org/10.1016/j.jocs.2017.07.004

simulations of millions of particles but are performed at higher levels of particulate coarse-graining. Such simulations have already led to ground-breaking insights into chemically specific structural self-assembly and large scale materials property prediction [9]. However, they are still limited to at best volumes of matter whose linear dimensions are on the micrometre scale, well below the size one would need to study, e.g., for the formation of fractures on millimetre scales, let alone to predict performance of materials on macro scales such as arise in typical automotive and aerospace applications. Stepping up to that scale requires not only simulating billions of particles, but also dealing with the non-linear increase in the temporal range that needs to be analysed. This in turn amplifies the need for advanced statistical analysis, which itself becomes a further burden on computational resources. Hence, multiscale challenges such as running atomistic simulations on demand in order to inform a series of coarse-grained simulations and, in future also finite element calculations, will clearly lead us far beyond of what is currently possible at the petascale.

Similar “grand challenges” can be found across the entire scientific spectrum. To get from the current state of the art to more realistic macroscopic regimes requires new developments in multiscale computing to extend discrete representations into the continuum level, optimally designed to exploit an increase in computing capabilities by a factor of between 100–1000. Indeed, advanced multiscale algorithms in combination with exascale resources will help us transition to predictive multiscale science. To make this possible, we shall need generic multiscale computing algorithms capable of producing high-fidelity scientific results and scalable to emerging exascale computing systems. We call this *high performance multiscale computing* (HPMC).

In a multiscale simulation, each relevant scale needs its own type of solver. [26,27] Accordingly, we define a multiscale model as a collection of coupled single scale models (loosely defined based on the dominant physical properties that can be computed reliably with a dedicated, so-called “monolithic” solver). We will demonstrate here that one can then identify generic *multiscale computing patterns* (MCPs) arising in multiscale applications that dictate the scope for novel multiscale algorithms at the exascale.

Exascale computing poses a number of key challenges that application developers cannot ignore, such as scheduling and robustness of algorithms and their implementation on millions of processors, data storage and I/O for extreme parallelism, fault tolerance, and reducing energy consumption. [28–31] For these reasons, an incremental approach that attempts to scale up monolithic petascale solutions will not be successful at the exascale. Instead, novel algorithms are needed across the software stack, bridging between the applications and the hardware environments. These algorithms need to be designed specifically to address these exascale challenges in order to guarantee efficiency and resilience. We believe that, drawing on the concept of generic MCPs, we can realise a separation of concerns, where the challenges stated above can be resolved to a large extent on the level of the MCPs, while the multiscale application developers can focus on composing their multiscale simulations. This would then lead to much shorter development cycles for multiscale simulations and much more reliable multiscale computing on exascale machines.

There are also strong computational considerations that dictate a need to shift the paradigm for usage of high performance computers from the conventional promotion of monolithic codes which scale to the full production partition of these computers, to much more flexible computing patterns. This calls for new algorithmic approaches like the ones we introduce here, based on our vision of Multiscale Computing Patterns. To clarify this further, computational scientists have worked out numerous effective ways in which to perform spatial domain decomposition. However, petascale and future exascale machines can only reach these performance levels by aggregating a large number of cores whose individual clock speeds are no longer increasing. As a result these high performance computers are becoming “fatter”, not faster and speed-up is only achievable by efficient parallelism over all the cores. But because the parallelism is usually applied to the spatial domain, we are increasingly simulating larger slabs of matter, applying weak scaling by using more particles, a higher grid resolution or more finite elements. Yet it often it is the temporal behaviour that one is really interested in, and that behaviour is not extended by adopting larger computers of this nature, or by making the problem physically larger. Since the scientific problems of interest usually have timescales which scale as a nonlinear function of the volume of the system under investigation, each temporal update requires more wall clock time for larger physical problems. This is in fact a recipe for disaster: *we are not getting closer to studying large space and long time behaviour with monolithic codes.* To be sure, accelerators (such as GPUs) and special purpose architectures [32–34] can speed up many floating point calculations in particular cases such as molecular dynamics, often by a factor between one and ten, but this is not sufficient to bridge the vast timescales of concern that range from femtoseconds to seconds, hours and years; nor indeed to quantify the uncertainty in today’s still all to prevalent “one-off” simulations.

What *is* needed are more innovative ways of bridging this divide. Multiscale computing as we propose it, is able to do this by deploying its various single scale component parts across such heterogeneous architectures, mapped so as to produce optimal performance and designed to bridge both time and space scales. Thus, we have embarked upon a programme to efficiently deploy componentised multiscale codes on today’s and future high performance computers and, thereby, to establish a new and more effective paradigm for exploiting HPC resources.

The goal of this discussion paper is to share our vision on high performance multiscale computing, introduce some or our initial results, and discuss further research directions and open questions.

2. Multiscale computing

Over the last decade, and with many collaborators, we have developed the so-called Multiscale Modelling and Simulation Framework (MMSF) for designing, programming, implementing and executing multiscale applications [3,26,27,35–42]. This framework has been successfully tested on applications from several fields of science and technology (e.g. fusion [35,43], computational biology [35,44,45], biomedicine [17,35,36,46–52], nanomaterial science [9,13,35], and hydrology [35]). The MMSF offers many benefits: a clear methodology, software and algorithm reuse, the possibility to couple new and legacy codes, heterogeneous distributed computing, and access to unprecedented computing resources [26].

Other approaches, in the same spirit as the MMSF, have been described in the literature. For instance, the U.S. Army Research Laboratory has made important steps forward in establishing multiscale computing approaches for materials modelling, establishing a computational framework for bridging different scales [53,54]. Their primary motivation is to shorten development times and to reduce the cost of evaluating new materials for military use. The framework has been shown to connect between two different scales (macroscopic and microscopic), but can easily be extended to interconnect three or more hierarchical scales. A key feature of their proposed framework is a standalone *Evaluation Module*, which has a light coupling between the macroscopic and microscopic models. This module accepts requests from the macroscopic model to perform runs of the microscopic model, and takes care of the scheduling, execution, and data exchange activities required to fulfil the request. The exchange mechanism is asynchronous,