A Distributed Infrastructure for Multiscale Biomedical Simulations

Derek Groen[1], Joris Borgdorff[2], Stefan Zasada[1], Carles Bona-Casas[2],
James Hetherington[1], Rupert Nash[1], Alfons Hoekstra[2] & Peter Coveney[1]

1. Centre for Computational Science, University College London, UK.
2. Section Computational Science, University of Amsterdam, the Netherlands

Introduction

Human physiology occurs across a range of spatial and temporal scales. Multiscale modelling allows explicit simulation across different scales, enabling large systems on a relatively coarse-grained level while providing sufficient detail for specific sub-domains of interest [1]. The EU funded MAPPER project [2] aims to facilitate distributed multiscale simulation of complex systems on production-level computing, network and storage infrastructures. The MAPPER project features a range of infrastructure components for multiscale simulation, as well as five exemplar multiscale application domains, including the Virtual Physiological Human. Here, we describe two MAPPER applications which originate from the physiology domain, and present our experience of constructing these distributed multiscale simulations. We also describe the infrastructure components that enable these simulations, discussing their key functionalities as well as the advantages that this infrastructure offers for other applications within the Virtual Physiological Human Initiative.

In-stent Restenosis

The three-dimensional model of in-stent restenosis (ISR3D) [3, 4] models restenosis in coronary arteries after stenting. ISR3D supports the hypothesis that smooth muscle cell proliferation drives the restenosis, which is in turn affected by wall shear stress of the blood flow and by local growth inhibiting drugs, diffused by a drug-eluting stent [3]. ISR3D is based on a two-dimensional model (ISR2D) [5, 6] which runs faster but does not incorporate full stent design, exact blood flow or realistic, curved geometries. The ISR3D model consists of four single scale models: smooth muscle cell proliferation (SMC), thrombus formation (TF), blood flow (BF), and drug diffusing from a drug-eluting stent (DD). Each of these models acts on the same spatial scale, but they exhibit temporal scale separation, acting on the second scale (BF), hour scale (DD), and day scale (SMC). The submodels used in ISR3D are heterogeneous and each of the submodels and scale bridging methods are custom made, with the exception of BF, which uses the Palabos lattice-Boltzman simulator[1].

We have coupled the four submodels using the multiscale coupling library and environment (MUSCLE) [7] and demonstrated the multiscale application as it runs across three European sites (including one EGI and one PRACE site). We provide an overview of how we deployed ISR3D on the MAPPER infrastructure in Figure 1. To make the concurrent bootstrapping and execution of the submodels both possible and convenient, we use the QCG-Broker [8] to reserve our resources in advance and the GridSpace [9] software service to hide the underlying complexity of the simulation execution from the user.

HemeLB

We use the HemeLB [10, 11] lattice-Boltzmann solver to simulate bloodflow in vessels in the human brain, in support of clinical neurosurgery. The behaviour of this bloodflow plays a crucial role in the understanding, diagnosis and treatment of cardiovascular disease; problems are often due to anomalous flow behaviour in the neighbourhood of bifurcations and aneurysms within the brain. Simulation offers the possibility of performing patient-specific virtual experiments to study the effects of courses of treatment with no danger to the patient. HemeLB is specifically designed to efficiently handle sparse topologies, support real-time visualizations and allow for remote steering of the simulation.

Within MAPPER, we are constructing a multiscale simulation by coupling different instances of HemeLB together, each of which runs at a different resolution. We use high resolution instances of HemeLB to model the regions of direct clinical interest, such as an intercranial aneurysm, and apply instances with a slightly more approximate but faster method to model the other parts of the human brain. Still further away, the rest of the

[1] http://www.palabos.org

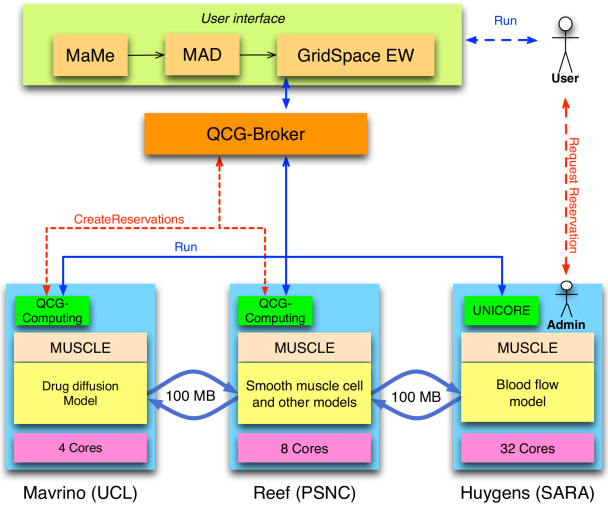

Figure 1: Architectural overview of an example run of the ISR3D distributed multiscale application.

circulatory system can be abstracted to a network model of the vasculature and a pump, i.e. the heart. We have defined a generalized coupling interface for the submodels within this application. This interface will allow us to trivially switch between different data transfer methods to communicate between the submodels. For example, one can then select MPI for optimal local-site performance, MUSCLE for convenient deployment and boot-strapping of the submodels, or MPWide [12] to optimize for performance across long-distance networks.

The distributed MAPPER infrastructure

The MAPPER consortium has, in collaboration with major infrastructure providers such as PRACE and EGI, constructed an infrastructure to support distributed multiscale applications. These applications are particularly demanding, as they use a collection of different submodels mapped to different computational resources. Here we briefly describe the four main software components currently deployed on the MAPPER infrastructure, list their functionalities and explain how other physiology applications could benefit from adopting these tools.

The *QosCosGrid environment* [8] is a multi-layer architecture which is able to deal with computationally intensive, complex and parallel simulations that require execution at multiple computational sites. QosCosGrid middleware is able to combine resources from different administrative domains into a single powerful machine and is tightly integrated with commonly used tools for parallel and multiscale simulations (e.g., OpenMPI [13], ProActive [14] and more recently MUSCLE). QosCosGrid middleware provides the ability to work across heterogeneous computing sites and simplifies many complex deployment, advance reservation and access procedures. *GridSpace 2* [9] is a user interface that allows researchers to conveniently exploit distributed computing platforms. GridSpace 2 uses exploratory programming where each workflow is decomposed into a number of script snippets, each written in a different language if needed. The GridSpace Workbench enables users to execute snippets or entire workflows, while its web portal assists users in developing simulation workflows using popular scripting languages. The Multiscale Application Designer allows researchers to visually compose their workflows and the GridSpace Execution Environment evaluates snippets and executes them on remote sites when needed. The *Application Hosting Environment* (AHE) [15] provides simple desktop and command line interfaces to run applications on a wide range of remote production resources (e.g., UNICORE, Globus or QCG grids), hiding the underlying complexity of the underlying middleware. The AHE is designed to allow scientists to quickly and easily run and monitor unmodified legacy applications on grid resources, manage file transfers. The philosophy of the AHE is based on the fact that very often a group of researchers will want to access the same application, but not all of them will possess the skill or inclination to install the application on remote grid resources. In the AHE, an expert user installs the application and configures the AHE server, so that all participating users can share the same application. The *MUltiScale Coupling Library and Environment* (MUSCLE) [7] provides the means to couple distinct computational models of multiple disciplines. Its aim is to provide a uniform platform to implement submodels using varying programming languages, execute them on heterogeneous machines and couple them across diverse networks. In addition, it considers the temporal and spatial scale at which the submodels are operating using the Complex Automata

(CxA) formalism [16]. MUSCLE features Java and C APIs inside the JADE [17] submodel agent, allowing developers to easily use Java code or C, C++ or Fortran as a language of choice for model implementation. MUSCLE also allows users to map submodels to different machines at runtime.

Discussion

MAPPER aims to establish a persistent infrastructure for distributed multiscale simulations. Within the MAPPER project we are working with seven different application communities to port their multiscale applications to our infrastructure. Outside of the VPH community, we are working the EFDA Task Force on Integrated Tokamak Modelling and canal engineers from ESISAR in France amongst others to ensure that the MAPPER infrastructure becomes a truly useful platform for all communities performing multiscale simulation. In addition to our continued work developing the two application scenarios described in this abstract, we actively encourage external projects to use and contribute to the MAPPER infrastructure.

We are also working with resource providers such as EGI and PRACE to extend the reach of the MAPPER infrastructure, by building on as wide a range of computational resources as possible. MAPPER itself does not operate any computational resources, but builds on the lower level platforms put in place in Europe and beyond. It aims to add extra value to the services that those platforms provide, and to enable computational scientists to operate in a truly multiscale fashion. To that end we have established a task force to work with these resource providers towards integrating their machines with MAPPER. We will report on the progress of this task force, and the ongoing development of the MAPPER infrastructure.

Acknowledgements

We thank our colleagues in the MAPPER consortium and the HemeLB development team. This work has been supported by the MAPPER EU-FP7 project (grant no. RI-261507).

References

[1] P. M. A. Sloot and A. G. Hoekstra. Multi-scale modelling in computational biomedicine. *Briefings in Bioinformatics*, 11(1):142–152, 2010.

[2] MAPPER: Multiscale Applications on European e-Infrastructures- http://www.mapper-project.eu, 2011.

[3] D. J. W. Evans, P. V. Lawford, J. Gunn, D. Walker, D. R. Hose, R. H. Smallwood, B. Chopard, M. Krafczyk, J. Bernsdorf, and A. Hoekstra. The application of multiscale modelling to the process of development and prevention of stenosis in a stented coronary artery. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1879):3343–3360, 2008.

[4] A. Caiazzo, D. Evans, J.-L. Falcone, J. Hegewald, E. Lorenz, B. Stahl, D. Wang, J. Bernsdorf, B. Chopard, J. Gunn, R. Hose, M. Krafczyk, P. Lawford, R. Smallwood, D. Walker, and A. G. Hoekstra. Towards a complex automata multiscale model of in-stent restenosis. In Gabrielle Allen, Jaroslaw Nabrzyski, Edward Seidel, Geert van Albada, Jack Dongarra, and Peter Sloot, editors, *Computational Science ICCS 2009*, volume 5544 of *Lecture Notes in Computer Science*, pages 705–714. Springer Berlin / Heidelberg, 2009.

[5] A. Caiazzo, D. J. W. Evans, J. L. Falcone, J. Hegewald, E. Lorenz, B. Stahl, D. Wang, J. Bernsdorf, B. Chopard, J. Gunn, R. Hose, M. Krafczyk, P. Lawford, R. Smallwood, D. Walker, and A. G. Hoekstra. A Complex Automata approach for In-stent Restenosis: two-dimensional multiscale modeling and simulations. *Journal of Computational Science*, 2(1):9–17, mar 2011.

[6] H. Tahir, A. G. Hoekstra, E. Lorenz, P. V. Lawford, D. R. Hose, Jm Gunn, and Dm J. W. Evans. Multi-scale simulations of the dynamics of in-stent restenosis: impact of stent deployment and design. *Interface Focus*, 1(3):365–373, 2011.

[7] J. Hegewald, M. Krafczyk, J. Tölke, A. Hoekstra, and B. Chopard. An agent-based coupling platform for complex automata. In *Proceedings of the 8th international conference on Computational Science, Part II*, ICCS '08, pages 227–233, Berlin, Heidelberg, 2008. Springer-Verlag.

[8] K. Kurowski, T. Piontek, P. Kopta, M. Mamonski, and B. Bosak. Parallel large scale simulations in the pl-grid environment. *Computational Methods in Science and Technology*, pages 47–56, 2010.

[9] Distibuted Computing Environments - GridSpace Technology - http://dice.cyfronet.pl/gridspace/, 2011.

[10] M. D. Mazzeo, S. Manos, and P. V. Coveney. In situ ray tracing and computational steering for interactive blood flow simulation . *Computer Physics Communications*, 181:355–370, 2010.

[11] M. D Mazzeo and P. V. Coveney. HemeLB: A high performance parallel lattice-Boltzmann code for large scale fluid flow in complex geometries. *Computer Physics Communications*, 178(12):894–914, 2008.

[12] D. Groen, S. Rieder, P. Grosso, C. de Laat, and P. Portegies Zwart. A light-weight communication library for distributed computing. *Computational Science and Discovery*, 3(015002), August 2010.

[13] Open MPI: Open Source High Performance Computing - http://www.open-mpi.org, 2012.

[14] ProActive - http://proactive.inria.fr, 2012.

[15] S.J. Zasada and P.V. Coveney. Virtualizing access to scientific applications with the application hosting environment. *Computer Physics Communications*, 180(12):2513 – 2525, 2009.

[16] A. G. Hoekstra, J.-L. Falcone, A. Caiazzo, and B. Chopard. Multi-scale modeling with cellular automata: The complex automata approach,. In *8th International Conference on Cellular Automata for Reseach and Industry (ACRI 2008) in series Lecture Notes in Computer Science*, volume 5191, pages 192–199. Springer, 2008.

[17] JADE - http://jade.tilab.com, 2012.