



The CloudSME simulation platform and its applications: A generic multi-cloud platform for developing and executing commercial cloud-based simulations

Simon J.E. Taylor^a, Tamas Kiss^b, Anastasia Anagnostou^{a,*}, Gabor Terstyanszky^b, Peter Kacsuk^c, Joris Costes^d, Nicola Fantini^e

^a Modelling & Simulation Group, Department of Computer Science, Brunel University London, Uxbridge, Middx, United Kingdom

^b Centre for Parallel Computing, University of Westminster, London, United Kingdom

^c MTA-SZTAKI, Budapest, Hungary

^d CMLA Laboratory, ENS Paris-Saclay, France

^e CloudBroker GmbH, Zurich, Switzerland

HIGHLIGHTS

- The CloudSME Simulation Platform that combines an AppCenter with the workflow of the WS-PGRADE/gUSE science gateway framework and the multi-cloud-based capabilities of the CloudBroker Platform.
- How the CloudSME Simulation Platform is used to develop and execute commercial cloud-based simulations on multiple clouds.
- A case study that shows how the CloudSME Simulation Platform is used to create a multiprocessor cloud-based simulation application in Computational Fluid Dynamics on multiple clouds.
- A case study that shows how cloud-based applications that have experiments with multiple runs (parameter sweep) can be implemented on multiple clouds using the Platform.

ARTICLE INFO

Article history:

Received 19 December 2017
Received in revised form 26 April 2018
Accepted 2 June 2018
Available online 20 June 2018

Keywords:

Multi-cloud simulation
Cloud computing
Workflow
Computational fluid dynamics
Agent-based simulation

ABSTRACT

Simulation is used in industry to study a large variety of problems ranging from increasing the productivity of a manufacturing system to optimising the design of a wind turbine. However, some simulation models can be computationally demanding and some simulation projects require time consuming experimentation. High performance computing infrastructures such as clusters can be used to speed up the execution of large models or multiple experiments but at a cost that is often too much for Small and Medium-sized Enterprises (SMEs). Cloud computing presents an attractive, lower cost alternative. However, developing a cloud-based simulation application can again be costly for an SME due to training and development needs, especially if software vendors need to use resources of different heterogeneous clouds to avoid being locked-in to one particular cloud provider. In an attempt to reduce the cost of development of commercial cloud-based simulations, the CloudSME Simulation Platform (CSSP) has been developed as a generic approach that combines an AppCenter with the workflow of the WS-PGRADE/gUSE science gateway framework and the multi-cloud-based capabilities of the CloudBroker Platform. The paper presents the CSSP and two representative case studies from distinctly different areas that illustrate how commercial multi-cloud-based simulations can be created.

© 2018 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Simulation is widely used in science and industry to study a variety of problems across a wide range of domains [1]. Typically to study a problem, a model of a system is built and then simulated

under different conditions [2]. Examples include how a biological system might change over time in response to different drug treatments, how different manpower planning strategies impact healthcare provision, how to optimise the design of a wind turbine, and how production could be improved in a factory. The simulation of some models can be computationally demanding. For example, many numerical simulation techniques such as Computational Fluid Dynamics (CFD) make use of parallelisation and

* Corresponding author.

E-mail address: anastasia.anagnostou@brunel.ac.uk (A. Anagnostou).

high performance computing (HPC) to speed up the execution of a simulation [3]. Simulation studies also often require many experiments to be run. The use of a single computer restricts these to being executed in sequence. This can lead to extremely lengthy or prohibitive experimentation time, especially if the run time of a single simulation is large [4,5]. Distributing runs across multiple computers can reduce this experimentation time [6]. However, access to distributed and/or HPC resources can be prohibitively costly and typically requires large upfront investments.

Cloud computing offers on-demand computing resources that can be rapidly accessed to support a specific task and then discarded [7]. This model of service provision enables end-users to access hardware without capital outlay or the on-going cost of IT management, and access to software potentially on a pay-per-use basis rather than an annual license. The cost of hiring cloud-based computers is relatively low. The possibility of developing cloud-based simulation that can use cloud computing to process large models and multiple simulation runs is therefore attractive. However, different cloud service models, cloud deployment models and associated technologies can make cloud-based application development costly due to the training required to learn different cloud implementation techniques [8]. This can be prohibitive for Small and Medium-sized Enterprises (SMEs).

There have been attempts to create cloud-based simulation applications. Typically, a cloud-based simulation will be created by deploying the simulation on a cloud provided by Amazon or Google. This involves learning and then using the cloud-specific API. To port the simulation to another cloud involves learning another other cloud-specific API. Porting would therefore incur additional training and implementation costs. The challenge taken up by the Cloud-based Simulation platform for Manufacturing and Engineering (CloudSME) project¹ was to investigate if it was possible to create a generic approach to cloud-enabling a simulation that could be used to reduce implementation costs and, where appropriate, to offer it as a commercial product or service. The project developed the CloudSME Simulation Platform (CSSP), an architecture that combines an AppCenter with the workflow of the WS-PGRADE/gUSE [9] science gateway framework and the multi-cloud-based capabilities of the CloudBroker Platform.² The CSSP has been used to implement eleven commercial simulation products across a wide range of industrial applications (see the CloudSME Website³ for examples). This paper therefore describes our Platform and two representative case studies from distinctly different areas that illustrate how commercial multi-cloud-based simulations can be created.

The paper is structured as follows. Related work discusses previous attempts to cloud-enable simulations in Section 2. The CSSP is then described in Section 3 as well as its key components. Section 4 presents two case studies, one showing how parallel multiprocessor simulation applications can be deployed on cloud and how cloud can be used to speed up simulation experimentation and parameter sweeps. Finally, Section 5 summarises the paper and gives future directions for the evolution of the platform.

2. Related work

There are different approaches to developing cloud-based simulations ranging from “one-off” deployments to comprehensive workflow systems. This section gives a short overview that highlights the current limitations of these approaches with respect to multi-cloud deployment and commercial application development.

Cloud-based simulation applications can be split into two types: discrete and continuous. Discrete simulations are typically either discrete-event (used to study complex queuing systems in manufacturing, healthcare, supply chains, etc.) or agent-based, used to study systems where individuals interact with each other (infection spread, social media, traffic, etc.) [1]. Continuous simulation techniques such as CFD are used to analyse flow-related problems ranging from oil and gas storage, vehicle and aircraft design and biomedical engineering. CFD models represent flows in partial differential equations (PDE) and CFD simulations use PDE solver to simulate the system over time [10].

An example of a cloud-based discrete simulation is the distributed agent-based traffic simulator Megaffic that allows adaptive resource provisioning to speed up the execution of a single simulation run [11]. This system uses Bulk Synchronous Processing (BSP) to process and synchronise between elements of a simulation assigned to distributed processors. At each checkpoint a decision is made to add/remove processors and to rebalance the processing load. The simulator was implemented using the Google Compute Engine cloud.

The Scalable Electro-Mobility Simulation Cloud Service (SEM-Sim CS) is the cloud-based version of the Scalable Electro-Mobility Simulation (SEMSim) platform used to study the impact of large-scale electromobility on a city’s infrastructure (i.e. the replacement of the majority of vehicles with electric ones) [12]. SEMSim consists of two types of simulation: an agent-based traffic simulation (SEMSim Traffic) and a discrete-event power system simulation (SEMSim Power). When an experiment is run, the two simulations are linked together as a distributed simulation that synchronises simulation time advance between the discrete simulations. SEM-Sim CS consists of a Dispatch Server (DS) that manages simulation experiment submission and running on a virtual machine on a cloud, a Visualisation Server (VS) to update real-time traffic simulation visualisation and the simulation instances (preconfigured simulation instances deployed onto virtual machines). The cloud service has been deployed on to Google Compute Engine and reports good performance with respect to dedicated HPC.

The D-Mason framework is a parallel version of the Mason library for writing and running distributed agent-based simulations. Carillo, et al. [13] report on implementing D-Mason on Amazon EC2 using StarCluster. Their results show that HPC can perform better than their cloud implementation of this discrete simulation library but at significantly greater cost.

GridSpice [14] is a cloud-based simulation platform for distributed smart power grid simulation that supports the development of models consisting of a transmission network, distribution networks and power generators. Each of these distributed simulation elements are mapped onto a simulation cluster which in turn manages the load by adding or removing processors as necessary. Rather than using BSP, GridSpice uses the agent-based simulation framework Repast Symphony to synchronise time between simulations. The cloud deployment of GridSpice is on Amazon EC2 using StarCluster.

Speedup in continuous simulations is gained from HPC by decomposing models onto multiple processors and then using technologies such as Message Passing Interface (MPI) or Parallel Virtual Machine (PVM) [15] to implement the tightly coupled communication needed by the parallel implementation of a PDE solver. Guzzetti, et al. [16] investigated the impact of different HPC platforms for numerical simulation in computational haemodynamics with the LiFEV (Library for Finite Elements 5). They compared in-house computing clusters, a large-scale university-based HPC cluster and a regional supercomputer with clouds (Penguin’s On-Demand HPC Cloud Service and Amazon’s EC2). They showed that the decomposition and mapping of the CFD model is important in obtaining good performance and that clouds may be utilised for

¹ www.cloudsme-apps.com.

² www.cloudbroker.com.

³ <http://www.cloudsme-apps.com/practical-examples/>.

scientific CFD simulations possibly at lower cost/performance than using a more expensive local computing cluster.

Slawinski, et al. [17] compared four different clouds and showed that cloud can be an effective platform for CFD but deployment effort, cloud costs, application performance, and availability can affect the choice of cloud. Ledyayev and Richter [18] used OpenStack to create their cloud-based platform. They used three case studies in transportation modelling (network optimisation), high energy physics (monte carlo) and materials simulation (CFD) to investigate the commonly occurring scenarios of multiple runs of a single code, single runs of a parallel code and multiple runs of a parallel code. They concluded that cloud is suitable for multiple runs of a single code but needs specialist hardware to support parallel code. O'Leary, et al. [19] created HPCCloud, a cloud/web-based simulation environment platform that utilised web technologies to deliver a Software-as-a-Service advanced modelling and simulation environment. The environment allows a simulation to be submitted to a cluster in a cloud and submits work to these via distributed task queues. The core of the system uses the Girder data management toolkit, Amazon EC2 and StarCluster. Results demonstrated performance comparable to on-site resources.

Several workflow systems have interfaces to cloud. Pegasus workflows can be deployed in a cloud by configuring cloud instances as a HTCondor pool using NSF Chameleon and Amazon [20]. Taverna workflows are executed on clouds by deploying the Taverna Server to manage execution [21]. There are several examples of Taverna running on public clouds and Amazon (<https://taverna.incubator.apache.org/introduction/taverna-in-use/>). Another example of Taverna on cloud is the Virtual Physiological Human (VPH) Share project that developed a web-based collaborative environment for VPH researchers that uses cloud & HPC services.⁴ VPH Share uses the Atmosphere cloud management platform [22] with a Taverna plugin and allows users to switch between clouds including OpenStack, Amazon Web Services EC2, Rackspace, Azure and Google Cloud platform at the workflow level. Galaxy has been implemented on Amazon EC2 by using a scalable approach based on Cloudman to manage work on cloud instances such as Amazon EC2, OpenStack, and OpenNebula. BioCloud [23] consists of the BioCloud Portal and a Workflow Manager/Scheduler based on Galaxy. It uses a multi-cloud approach and acts as a broker across multiple cloud providers by providing a solution to the efficient scheduling of workflow across different clouds (e.g. Amazon and Rackspace) [24]. The Distributed Application Runtime Environment (DARE) science gateway framework [25] offers access to several major cloud resources (Amazon EC2, Eucalyptus and Nimbus) through a manager worker scheme based on its BigJob and the Simple API for Grid Applications (SAGA).

Overall the examples of cloud-based simulations demonstrate good performance when porting a simulation to a cloud. However, all of these are dedicated to a single simulation application and are typically implemented on a single cloud. The reason for this specialised nature for these solutions is that these are all rooted in piecemeal approaches where the aim was to demonstrate the feasibility of cloud-based execution of a particular simulation only. These solutions have not been designed with the broader aim of providing a general platform or executing in a multi-cloud environment. Although, while extending these solutions with such generic capabilities would be technically feasible, it would require developing a complex general system from scratch.

Workflow systems show more widespread utility in that they have been developed for either general scientific needs or the needs for a particular area (e.g. bioinformatics and the provision of commonly needed tools). Most have interfaces to different clouds. This suggests that if the cloud-based simulation examples might

have taken a workflow-based approach then it could be possible to create a multi-cloud approach for cloud-based simulation. From a commercial perspective, however, there appears to be a general lack of core functionality such as billing and charging mechanisms.

In the next section we present our approach to creating a generic multi-cloud platform for developing and executing cloud-based simulations in commercial settings. This platform is generic as it supports a wide range of simulation applications, supports execution in multiple heterogeneous clouds, has full commercial support, and was developed via the integration and further development of existing components in order to significantly reduce its development time and effort.

3. CloudSME platform concept and architecture

The generic layered architecture of the CSSP is shown in Fig. 1. The platform was created by reusing, integrating and further developing existing technologies including the WS-PGRADE/gUSE gateway framework [26] and the CloudBroker Platform [27], and also extending these with new commercial components such as the CloudSME AppCenter. The primary target audience of CSSP are commercial independent software vendors and consultant companies developing and offering various simulation software solutions. However, the platform can also be utilised for academic/research oriented scenarios, as it was demonstrated for example by Temelkovski, et al. [28] where the CSSP was used to execute large scale molecular docking experiments on heterogeneous cloud computing resources from the graphical user interface of the Raccoon2 desktop application [29].

The major objective of the CSSP is to ease developers' efforts when cloud-enabling existing simulation software and to speed up significantly the "cloudification" process. Therefore, the CSSP aims to cater for various scenarios supporting different types of simulation software products (e.g. computational fluid dynamics, process simulation, computer aided design (CAD) software, etc.) and also allowing software vendors to explore different software provisioning models. Another important aspect for software vendors was to enable the resource utilisation of different heterogeneous clouds to avoid being locked-in to one particular cloud provider. The CSSP therefore supports this scenario and allows vendors to port their applications once and then seamlessly switch between multiple cloud providers or even utilise their resources simultaneously.

The CSSP consists of three layers:

1. **Simulation Applications Layer** that allows software vendors deploying and presenting simulation products to end-users as SaaS (Software-as-a-Service) in a wide range of scenarios and deployment models.
2. **Cloud Platform Layer** that provides access to multiple heterogeneous cloud resources and supports the creation of complex application workflows – a PaaS (Platform-as-a-Service) to create and execute cloud-based simulations.
3. **Cloud Resources Layer** that represents the IaaS (Infrastructure-as-a-Service) clouds connected to the platform.

These layers are presented in detail below.

3.1. Simulation applications layer

The top layer of CSSP is the Simulation Applications Layer that consists of the CloudSME AppCenter as a generic entry point to browse and execute various simulation applications. The AppCenter is a new component that was developed during the CloudSME project. The CloudSME AppCenter is a web-based one-stop-shop solution with the goal to provide software products and services by software vendors and service providers to end users that can

⁴ www.vph-share.eu.

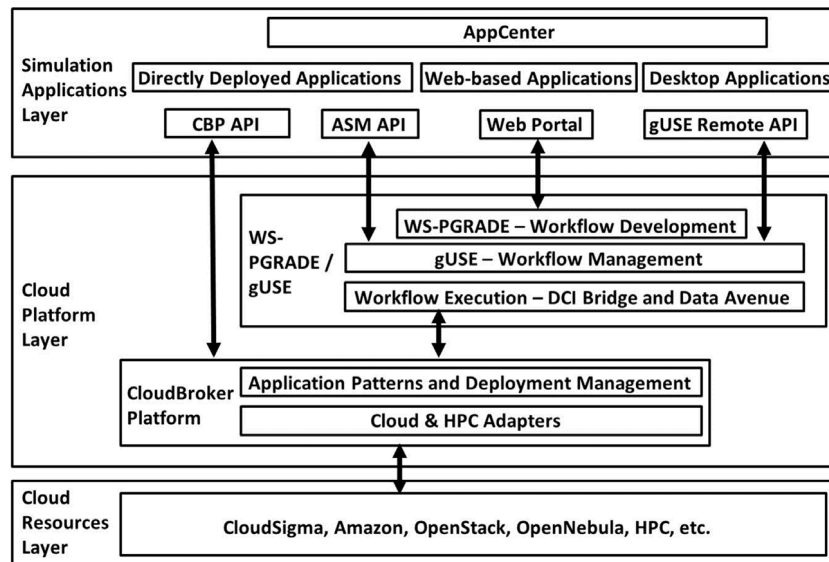


Fig. 1. The CloudSME simulation platform.

utilise them via a single interface. For this, the AppCenter stores information about software products in an accessible way, provides usage scenarios for the software, and offers billing functionality that includes price setting, payment integration and tracking of users' spending.

The AppCenter supports three main deployment models: Directly Deployed Applications, Web-based Applications, and Desktop Applications.

The default usage of the AppCenter is to deploy and offer cloud applications directly from there. After cloud-enabling the application via the CSSP, the software provider registers and publishes the application in the CloudSME AppCenter. Once published, end-users can access and execute these applications directly from the AppCenter and they will be charged for their utilisation based on software provider defined pricing policies (e.g. subscription-based pricing, resource consumption-based pricing, fixed charge for each simulation run, or even free of charge execution). The advantage of this solution is that the AppCenter provides easily customisable options to set up various pricing models and it is also responsible for accounting and billing. Users simply need to add credit to their account and can then execute simulation applications on all cloud resources supported by the selected application. The complexity of multi-cloud execution and billing is handled by the platform. On the other hand, this solution has one notable disadvantage. Software vendors must market and promote the simulation solution via the AppCenter and are also restricted to its user interface. Many software vendors with well-established marketing channels and sophisticated user interfaces do not prefer this option. Our AppCenter therefore supports two additional models.

In the first, vendors extend their traditional desktop software with cloud execution support. In this scenario, the desktop application is extended, using one of the suitable APIs offered by CSSP, to redirect computation intensive simulations to the cloud. In this case the application cannot be directly executed from the AppCenter. However, users can download the package from there, install it on their own machine, and then utilise its cloud extended capabilities. This solution enables vendors to keep the user interface (with minor modification) of their original product and also enables them to further market the solution beyond the AppCenter. On the other hand, using the suitable APIs of the AppCenter, accounting and billing can still be supported on multiple clouds.

In the second additional model, a web-based application is developed by the vendor that is simply linked to the AppCenter.

The web application can provide a complete custom portal-based user interface, and can utilise capabilities of the CSSP and the AppCenter accounting and billing mechanisms via suitable APIs.

In order to facilitate the above development and service provisioning scenarios, the CSSP offers a diverse set of APIs to support developers. To enable the development of applications that are directly deployed in the AppCenter or the extension of desktop applications with cloud support, either the CloudBroker APIs (Java Client Library API or REST API) or the gUSE Remote API can be used. Using the CloudBroker APIs bypasses WS-PGRADE/gUSE and provides direct access from the application to the multi-cloud resources supported by the CloudBroker Platform. Using the Remote API of WS-PGRADE/gUSE enables developers to execute complex application workflows linking multiple application components together. As WS-PGRADE/gUSE is integrated with the CloudBroker Platform, multi-cloud execution capabilities are still fully utilised in this scenario. In case of web-based applications, either the ASM (Application Specific Module) API of WS-PGRADE/gUSE is used that enables the rapid development of a custom portal/gateway in the form of customised Liferay Portlets or a completely custom web interface is developed by embedding either CloudBroker API or gUSE Remote API calls. Alternatively the standard web-based interface to WS-PGRADE/gUSE can also be applied to launch workflows. All APIs are described in Akos, et al. [26].

Fig. 2 shows screenshots of the CloudSME AppCenter illustrating various application scenarios. Calculix and OpenFoam simulation environments are external web-based applications that are linked to the AppCenter, 3DScan insole designer enables users to run jobs directly from the AppCenter, while TransAT is downloadable as a desktop interface extended with cloud support.

3.2. Cloud platform layer

The middle layer of CSSP is the Cloud Platform Layer that consists of the cloud-based services from the CloudBroker Platform and the science gateway framework WS-PGRADE/gUSE. These components were developed prior to CloudSME and their first integration was implemented in the SCI-BUS (Scientific Gateway-based User Support) project [30]. During CloudSME this integration matured significantly and reached commercial production level.

The screenshot displays the CloudSME AppCenter interface. On the left, there is a navigation menu titled "Areas" with categories: All (6), Computational fluid dynamics (2), Computational structural mechanics (1), Discrete event simulation (0), Insole design (2), Manufacturing process optimization (0), and Other (1). The main content area features four application cards:

- CalculiX simulation environment:** Provider: DHCAE Tools GmbH, Area: Computational structural mechanics, Software type: Web Service. Description: DHCAE Cloud extends your CSM (Computational Structural Mechanics) capabilities with a complete CalculiX simulation environment and unlimited HPC resources in the cloud. Includes a "Visit" button.
- 3D Scan Insole Designer:** Provider: INGECON, Area: Insole design, Software type: Cloud application. Description: "3D Scan Insole Designer" helps to automate the design process of a tailored insole validating the 3D scan using a CAD program from a virtual mould of the foot. Includes a "Run job" button.
- TransAT:** Provider: ASCOMP, Area: Computational fluid dynamics, Software type: Desktop. Description: TransAT Cloud is a new way of accessing the power of High Performance Computers (HPC) for CFD simulations. The Cloud version of TransAT compliments the existing more traditional workstation deployment. Includes a "Download" button.
- OpenFOAM simulation environment:** Provider: DHCAE Tools GmbH, Area: Computational fluid dynamics, Software type: Web Service. Description: DHCAE Cloud extends your Computational Fluid Dynamics (CFD) capabilities with a complete OpenFOAM® simulation environment and unlimited HPC resources in the cloud. Includes a "Visit" button.

Fig. 2. The CloudSME AppCenter illustrating various application scenarios.

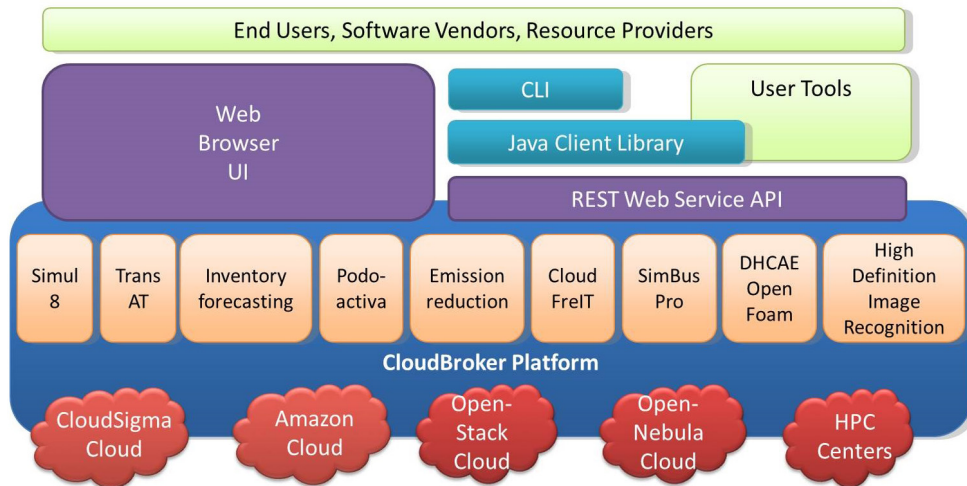


Fig. 3. CloudBroker platform architecture.

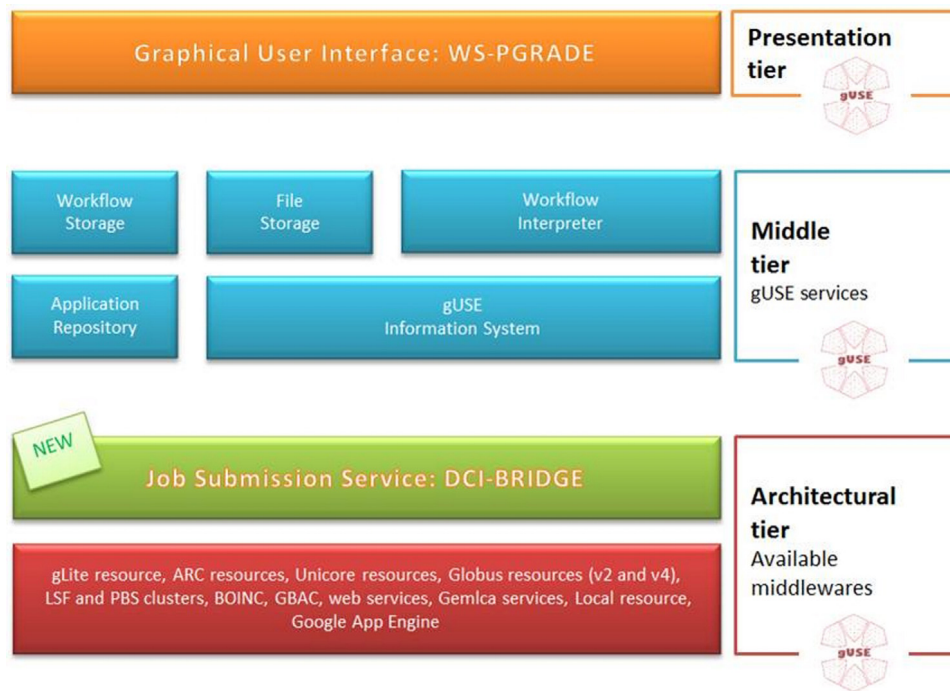


Fig. 4. Generic architecture of WS-PGRADE/gUSE.

3.2.1. The CloudBroker platform

The CloudBroker Platform is a commercial PaaS that supports the management and execution of software on different cloud provider resources. The generic architecture of CloudBroker is shown in Fig. 3.

At the bottom, CloudBroker uses IaaS clouds from resource providers and incorporates adapters both to public and private cloud infrastructures. The platform provides access to a wide range of resources including open source (e.g. OpenStack and OpenNebula) and proprietary (e.g. Amazon and CloudSigma) clouds, and also various HPC resources.

In the middle layer the applications are deployed. CloudBroker supports non-interactive serial and parallel batch processing applications on both Linux and Windows operating systems. The platform itself consists of a set of modules that manage processes, applications, users, finance (accounting, billing and payment), and runtime issues (process monitoring, queuing, resources, storage and images). A scalability and fault handler layer supervises scalability requirements and failure issues. Cloud Provider Access Management oversees the connection to each Cloud technology and can control the number of virtual machines (VMs) started for a given application on a given cloud. Application “patterns” are deployed to CloudBroker in a form that allows the platform when instructed to run the application on a particular cloud and cloud instance type. Two typical patterns are direct installation (an application package and deployment script that allows the installation of the software on a cloud instance) or virtualisation (virtual machine image containing installed software that allows direct deployment to a cloud instance).

At the top, CloudBroker offers various interfaces for access. Its two main operation modes to manage and use software in the cloud are either as direct front-end, or as a back-end middleware service. For the former, the platform can be accessed directly through the Web Browser User Interface. As a back-end for advanced and automatic usage, various APIs are provided for programmatic accessibility. These include REST web service interface, Java client library and Linux shell command line interface (CLI). Via these different APIs, the CloudBroker Platform can be utilised by

front-end software as middleware to allow access to applications in the cloud.

3.2.2. WS-PGRADE/gUSE

gUSE (Grid and Cloud User Support Environment) [26] is an open source scientific gateway framework providing users with easy access to cloud and grid infrastructures. gUSE provides with WS-PGRADE, a Liferay based portal to create and execute scientific workflows in various distributed computing infrastructures (DCIs) including clusters, grids and clouds. The generic architecture of WS-PGRADE/gUSE is presented in Fig. 4.

WS-PGRADE/gUSE consists of three layers: a top presentation layer, a middle management layer, and a bottom architectural execution layer.

The presentation layer (WS-PGRADE) includes a set of Liferay portlets to create, start and control workflows, monitor their execution on various DCIs, and present results to users. WS-PGRADE has a graph editor which can be used to build workflows and specify job configurations. A WS-PGRADE workflow is a directed acyclic graph that defines the execution logic of its components. An example for a WS-PGRADE workflow is presented in Fig. 5. The large boxes are jobs, while the smaller boxes are input and output ports representing input/output files for the jobs. The execution of a job can start when all of its inputs are available. Using this logic the WS-PGRADE workflow engine automates the execution of the workflow. For example, in case of the workflow of Fig. 5 only Gen3 can start executing when the workflow is submitted. MulCross and AddPair are waiting for the result of Gen3 and can start once the output file of Gen3 is available.

The WS-PGRADE workflow concept supports multiple levels of parallelism. Each job of the workflow can in itself be a natively parallel application (e.g. using MPI). The workflow can also have parallel branches (e.g. MulCross/ColMuls and AddPair/ColAdds are in parallel branches in Fig. 5) that can be executed in parallel of different resources. Finally, WS-PGRADE supports parameter sweep applications. Parameter sweep applications are simulations where the same simulation needs to be executed for multiple input

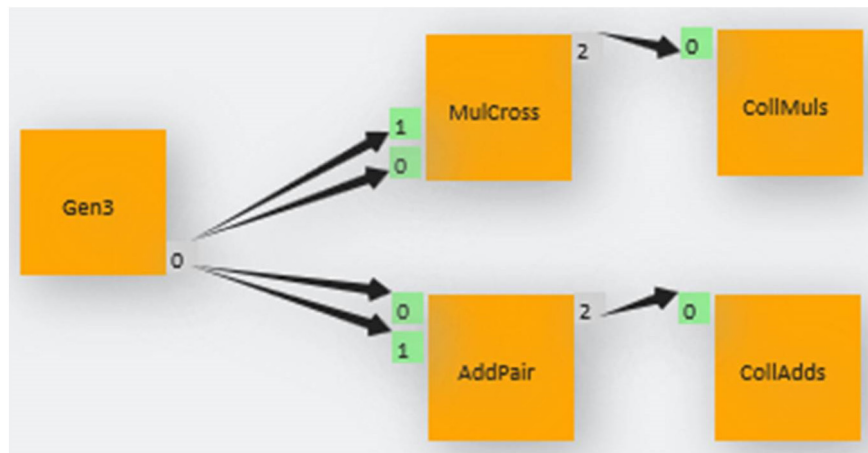


Fig. 5. Example WS-PGRADE workflow.

data sets. This feature enables the same workflow to be submitted with multiple input data sets simultaneously.

The middle/management layer consists of services to support workflow management and execution, e.g. file storage, workflow repository, and workflow interpreter (WFI) to manage the execution of workflows. The bottom architectural execution layer, that consists of the DCI Bridge and its various plug-ins manages job submission as specified in the workflow using the OGSA Basic Execution Service 1.0 (BES) interface [31,32]. As gUSE executes workflows it generates jobs. The DCI Bridge is used to submit and monitor these jobs to various DCIs as specified.

WS-PGRADE/gUSE and the CloudBroker Platform were integrated by creating a new plugin for the DCI Bridge to manage job submission to CloudBroker using its Java API. The plugin also allows authentication credentials to be passed from WS-PGRADE/gUSE and CloudBroker. When a workflow is executed, gUSE automatically submits these credentials to CloudBroker via robot credentials attached to the different nodes of the workflow. The WS-PGRADE/gUSE workflow configuration portlet was also modified to list and allow users to choose from the set of applications and cloud resources available through CloudBroker for a workflow node.

A full description of WS-PGRADE/gUSE gateway framework is available in Kacsuk, et al. [9], and Kacsuk [33] gives a complete overview of WS-PGRADE/gUSE and its applications.

3.3. Cloud resources layer

The bottom layer of CSSP is the Cloud Resources Layer that consists of a range of clouds and HPC resources accessible via the CloudBroker Platform. These currently include CloudSigma and Amazon public clouds, various private clouds based on either OpenStack or OpenNebula, and the HPC resources of, for example, the Cineca Galileo Cluster or the ETH Euler Cluster.

3.4. Usability of CSSP

CSSP is a production level commercial product. Therefore, usability is one of the key issues when further utilising and marketing the platform. The targeted users of the above described technology stack are twofold. On the one hand CSSP targets application developers, typically Independent Software Vendors (ISVs), that can utilise the platform when cloud enabling their simulation products. ISVs use the CSSP as PaaS and develop custom solutions for their customers. On the other hand, end-users, typically manufacturing companies utilise SaaS solutions that are developed using CSSP

technology. The complex technology stack is completely hidden from these end-users and they access either the AppCenter or a custom GUI provided by their ISV.

CSSP technology is commercialised and marketed by start-up company CloudSME UG that was set up by CloudSME project partners. The company's website⁵ supports both above described user categories. For application developers detailed tutorials are available supported by professional consultancy service. For end-users the company operates AppCenters that, after registration, provide access to commercial simulation applications. Currently 23 cloud-enabled applications are available from the two (production and demonstration) AppCenters.

The first users of the technology, both ISVs and end-users, were CloudSME project partners (24 commercial companies participated in the project). Since the end of the EU funded period in 2016, CloudSME UG made steady progress by securing its first commercial contracts and building custom solutions. CloudSME UG is also a partner in the EU funded CloudiFacturing⁶ project where the technology is extended and further developed and where an additional over 50 companies will utilise cloud-based simulation solutions.

4. Parallel multiprocessor and parameter sweep applications on the CloudSME simulation platform

The strength of the CSSP is that it provides support for the fast and efficient "cloudification" of a wide variety of applications. Having multiple entry points and a combination of workflow and multi-cloud features, the CSSP can be utilised in various application scenarios. Essentially, these scenarios fall into three categories:

- Simple virtualised applications where simulation software is launched on a single instance to allow remote, on-demand access,
- Parallel multiprocessor applications that use cloud instances to mimic closely coupled cluster resources, and
- Parameter sweep applications that use multiple cloud instances to speed up simulation experimentation.

Simple virtualised applications use only one instance at a time to run the cloud-based simulation. Although these applications do not require HPC resources, running them on clouds provides scalability in case the number of users increases, and also offers remote on-demand access to the simulation. Terstyanszky, et al. [34] discuss a

⁵ <http://www.cloudsme-apps.com/>.

⁶ <https://www.cloudifactory.eu/>.

commercial application example where a custom web portal was implemented and connected to the CSSP for uploading and validating scanned images for custom foot insole design. In that scenario, the validation portal utilised directly the CloudBroker platform API to execute the application on various cloud resources. Although no parallel execution was required, the portal enabled multiple, dynamically changing numbers of users, in this case podiatrists, to execute insole validations remotely on cloud resources.

This paper focuses on the other two application categories and illustrates how the CSSP can be effectively utilised in those more complex scenarios. Parallel multiprocessor applications may only require CloudBroker as only one experiment is normally run that typically investigates how a system changes over time (e.g. a liquid flowing through a pipe). Parameter sweep applications are supported by the parameter sweep workflow feature of WS-PGRADE/gUSE. These can be developed in the WS-PGRADE workflow system and then mapped to heterogeneous cloud resources via CloudBroker. The workflow can be embedded either in desktop applications using the Remote API or in web-based front-ends via the ASM API of WS-PGRADE. The following sections describe the implementation of both approaches on the CSSP and present short case studies describing its use.

4.1. Multiprocessor applications

Creating cloud-based simulations using a parallel multiprocessor approach on our Platform uses CloudBroker to deploy and manage the application on a cloud. In the following we present a generic approach to use the CloudSME Platform followed by a short case study based on a real industrial application.

4.1.1. Multiprocessor deployment on the CloudSME platform

In our experience most industrial continuous simulation applications consist of some kind of software package running on a PC. The package is typically a sophisticated environment that facilitates the development and visualisation of a 3D model and executes a simulation solver (either locally or on a local cluster using a parallelisation technique). We assume that this mode of use therefore calls for the package to remain on the PC and to have some kind of remote interaction with the cloud (model and solver uploaded to cloud for execution). There is no need for workflow in this example so the application will just interact with the CloudBroker component of our Platform. This is shown in Fig. 6.

The architecture consists of three elements: the Front End software application installed on a user workstation, the deployment on the CSSP and the cloud resources that will run the solver. A cloud-based application is implemented by creating an application package for CloudBroker. This consists of an installation shell script, the zipped solver and associated files (typically including MPI/PVM and scientific libraries) and a licence key (if needed). The deployment of the installation package is done by first requesting the specified cloud instance type, booting Linux virtual machines as necessary and then transferring the application package. The installation shell script is then run and compiles and builds simulation executables based on the designated model. The installation script also needs to build job run scripts needed by CloudBroker when `startjob()` (see Fig. 7) is called to run the application.

Fig. 7 shows the sequence diagram describing the interactions between the user, the front end and our Platform. The Front End can be linked to CloudBroker using several methods as described earlier. In this case we will assume the REST API. As can be seen, the user initially clicks on the “Run on Cloud” button on the Front End (1). The user then specifies the project name, the cloud/instance type and the region (only applicable if the cloud provider has multiple regions) on a configuration form (2–6). This is then translated into a XML job description and sent to CloudBroker using a

POSTrequest command (7). CloudBroker executes `createJob()` that uploads the installation package to the specified cloud/instance type and executes it. This installs and compiles the software on the multiple CPUs of the cloud instance (8). When this is complete, the job is set up and CloudBroker returns a jobID (9). The model and its parameters are then zipped into an INPUT.zip file by the Front End and sent to CloudBroker using a POSTupload command (10). CloudBroker then uploads the file to the instance (11) and confirms back this action to the Front End (12). The Front End then sends POSTsubmitJob with the jobID to CloudBroker and CloudBroker runs the job (13–14). The user then polls the Front End at intervals to determine if the job has finished by clicking on the refresh button (15). When doing this the Front End sends PUTgetJobStatus to CloudBroker and CloudBroker returns the status which is then displayed by the Front End (16–18). A user can then download the results and job costs (19–24). Finally the cost is synchronised against the user’s balance in the AppCenter (not shown). The resources (jobXMLdescription, jobID, fileID and jobCost) and REST API (POSTrequest(), POSTupload(), PUTsubmitJob(), showJobStatus(), GETdownloadlink(), PUTdownloadfile(), GETjobcost(), createJob(), uploadFile() and startJob()) calls are defined in the CloudBroker API documentation (cloudbroker.com/documentation).

4.1.2. Case study

To illustrate our approach we present experiences with the development of a cloud-based version of the TransAT (Transport phenomena Analysis Tool) CFD tool produced by ASCOMP GmbH (<http://ascomp.ch/products/transat-suite/>). TransAT is a commercial CFD tool used to simulate a wide range of single and multi-fluid/component flows with heat transfer. Typical applications of TransAT range from surface-tension dominated flows (e.g. microfluidics systems) to large-scale turbulent flows (e.g. hydrodynamics of ships and submarines) across a range of industries. The tool uses Open-MPI and PETSc (portable, extensible toolkit for scientific computation).

To test the cloud-based simulation a case study was created that investigated capillary movement in a system. This describes the ability of a liquid to flow in narrow spaces without, or against, an external force. It is a phenomenon that occurs at the interfaces between two immiscible fluids or between a fluid and a surface, and has important role in micro structures in several industrial and medical applications, such as micro heat exchangers, lab-on-chips, bio-MEMS and micro cooling electronics [35]. Simulation is typically needed to optimise and develop the design of such devices. In this case study, a 2D axis-symmetric CFD simulation was performed that represents the movement of water up a tube filled with air. The computation domain was equal to 1 cm in the horizontal direction and to 1.5 cm in the vertical direction. Two uniform meshes were run containing 5400 cells (coarse) and 21,600 cells (refined) respectively. Two clouds were compared: Amazon and CloudSigma.⁷ The instance types used in the experimentation were Amazon 16 cores (EC2 Compute Optimised – 4xlarge, 30GB RAM) and CloudSigma 24 cores (60 GHz, 32 GB RAM). Fig. 8 shows the comparative runtime and Table 1 shows the comparative speedup. As can be seen the best relative speedup is 6.28 for 8 CPUs on Amazon EC2 for the refined mesh.

To support commercial application scenarios, the TransAT software was published in the CloudSME AppCenter as a downloadable desktop application, as illustrated in Fig. 9. AppCenter users can simply download the front-end package from the AppCenter free of charge. Once installed, simulations can be built exactly the same way as using the traditional desktop software. However, when the model is ready, users will be offered the option to execute the solver on cloud resources, as shown in Fig. 10. When executing the simulation, the CSSP will charge the user’s account for the consumed resources.

⁷ www.cloudsigma.com.

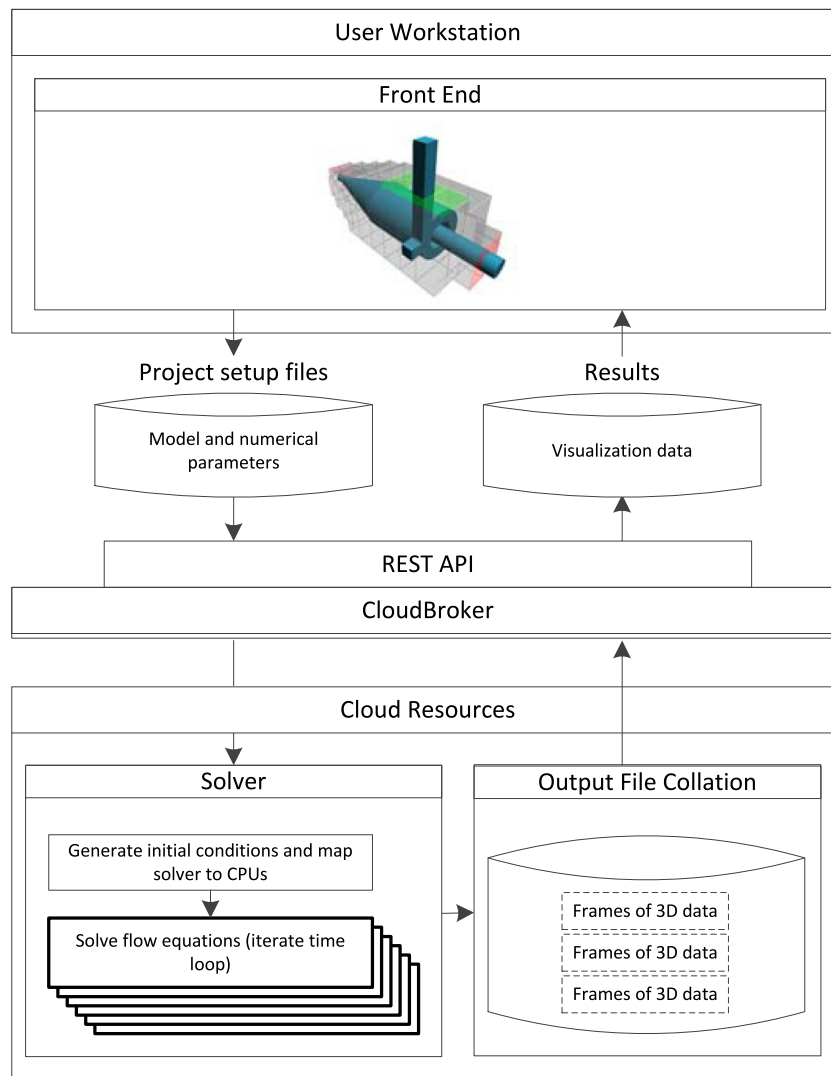


Fig. 6. Multiprocessor cloud-based simulation.

Table 1
Capillary movement speedup.

CPU's	2	4	8	16
Amazon coarse	1.82	3.22	3.80	4.45
CloudSigma coarse	1.95	3.21	4.98	5.00
Amazon refined	1.89	3.46	6.28	4.82
CloudSigma refined	1.79	3.40	5.33	5.98

4.2. Parameter sweep applications

In simulation a parameter sweep is where multiple experiments are performed to examine how a model changes with respect to a range of input parameters. For example, if a model has two parameters X and Y and a user wanted to investigate the behaviour of a model when X changes in the range 100, 200, ... 1000 and Y changes from 80 to 120 in steps of 10, then we would generate a parameter sweep with experiments representing all pairs $(\{100, 80\}, \{100, 90\}, \dots, \{1000, 120\})$. In this example 50 experiments would be generated. Often simulations have many parameters and can take a long time to run a single experiment. This can lead to lengthy or impractical experimentation time [4,5]. The goal of cloud-based simulation for parameter sweeps is therefore to speed up the execution of experimentation by using multiple cloud instances.

To reflect experiences with commercial system development, this case study presents work done with a widely used open source simulation system. The Recursive Porous Agent Simulation Toolkit (Repast) is a cross-platform, agent-based modelling and simulation toolkit. Repast Symphony (RepastS) is a Java-based simulation system that is used for developing a range of agent-based modelling and simulation applications in different fields [36]. Here we discuss the implementation based on RepastS 2.1.0. To enable the parallel execution of parameter sweeps running on multiple cloud resources, this application uses both components of the Cloud Platform Layer. CloudBroker manages deployment on multiple clouds and the parameter sweep functionality of the WS-PGRADE/gUSE workflow engine manages the execution of the simulation experiments and the parameter sweep.

4.2.1. RepastS deployment on the CloudSME simulation platform

The deployment of RepastS consists of two parts: deployment on CloudBroker and creation of the parameter sweep workflow on WS-PGRADE/gUSE.

Deployment on CloudBroker is done by creating an application package consisting of a deployment shell script, an execution shell script and the zipped RepastS environment. For each cloud deployment, CloudBroker is configured to create a virtual machine with a Linux Ubuntu OS image. Using its web interface, CloudBroker

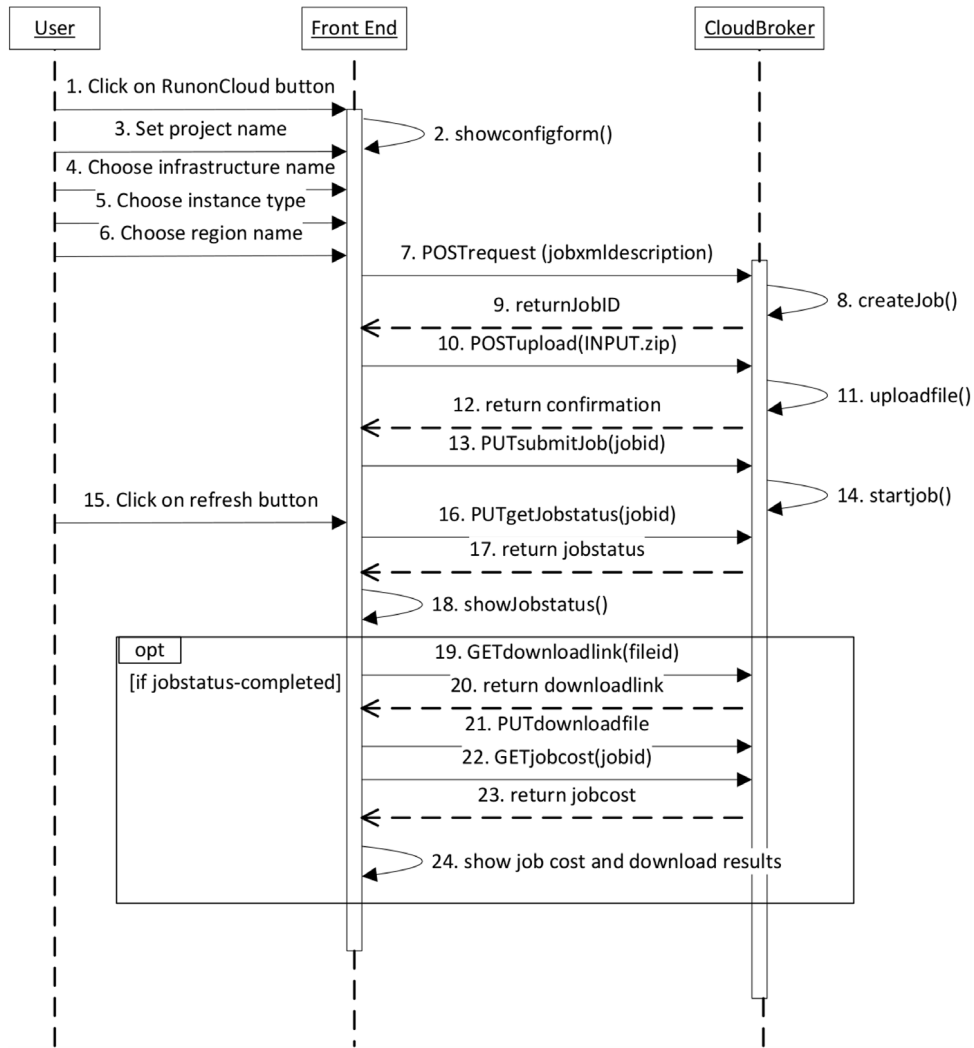


Fig. 7. Sequence diagram for continuous simulation multiprocessor applications.

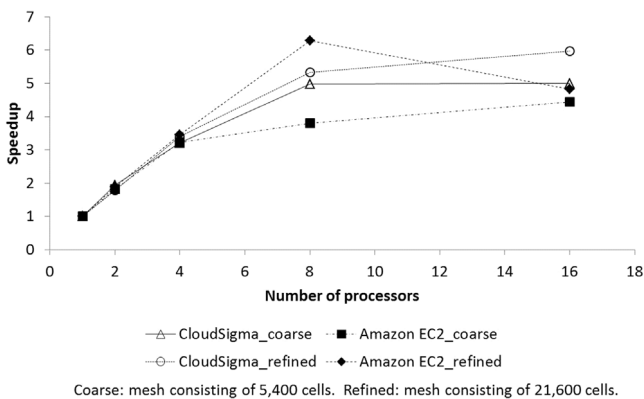


Fig. 8. Capillary movement performance comparison.

creates this virtual machine, transfers the application package to the virtual machine and then runs the deployment shell script. This installs RepastS, Java Runtime Environment and the execution shell script. When a job is started (i.e. a simulation run), the simulation model (a TAR archive consisting of the model source code and the simulation scenario) and the parameter sweep data (an XML

file specifying the input parameters) are transferred to the virtual machine. The execution script then validates these inputs, extracts the model files and runs the simulation. Results are then added to a TAR archive for upload back to the Platform.

The WS-PGRADE/gUSE web interface is used to create the parameter sweep workflow. An abstract workflow graph is first created using the graph editor. From the graph, the concrete workflow is then created and configured to run the selected software on the selected cloud resources. The same abstract workflow can be used to create many concrete workflows by reconfiguring them. Once the graph is completed, it can be saved and used to create a concrete workflow where the jobs can be configured (e.g. the simulation software, the cloud and the region of the resources, and the instance type).

Fig. 11 shows a parameter sweep workflow graph for RepastS execution on two clouds in parallel. In the Generator node (yellow box on the left hand side of the figure) two input ports (small green boxes attached to the node) and two output ports (small grey boxes attached to the node) are added. Input files can be uploaded at the input ports or can be channelled from the preceding node's output ports. In this parameter sweep workflow example the Generator node receives the model archive and an archive that contains as many parameter XML files as we define in the experiment. The Generator then instantiates equal number of cloud instances with the number of parameter files for each cloud. For example, if we

CloudSME Apps for manufacturing and engineering

Information Search Login Register

Home / Computational fluid dynamics / Desktop / TransAT 5.2

TransAT

free

Download

Provided by: ASCOMP

Version: 5.2

Simulation CFD

Free surfaces
Advanced turbulence
Water technology

Multiphase
Heat transfer
Nuclear technology

Atomization
Advanced turbulence
Energy Systems

Multiphase Multiphysics
Oil and Gas

TransAT is a CFD software code created to deal with complex, separated and transient-dominated aerodynamics and hydrodynamics applications requiring smart and fast meshing with advanced turbulence simulation strategies. The software is dedicated to 3D multi-fluid, multi-component flow systems with heat and mass transfer, including phase change. Multiphase flows of all nature are tackled with tailored schemes and models.

TransAT can deal with multi-components, complex-physics fluids present in specific industry verticals. It handles conductive, convective and radiative heat transfer, reactive flows, non-Newtonian flows, sedimentation, hydrates & wax, porous media, etc.

Multiphase flows (Level-Set, N-Phase mixture model), Turbulence (k-epsilon, LES, V-LES), Chemical reactions, Granular flows, Non-newtonian fluids ...

Installation and usage

Follow this step-by-step tutorial and get an overview of the capabilities of TransAT. The software available for download here is the complete GUI of TransAT. It is a desktop application that enables setting up a CFD simulation. The simulation itself runs on the Cloud, through the CloudBroker platform.

- Download Transat by clicking on the button in the left column. The archive is approx. 350MB and contains both a Linux version and a Windows version.

Fig. 9. TransAT in the CloudSME AppCenter.

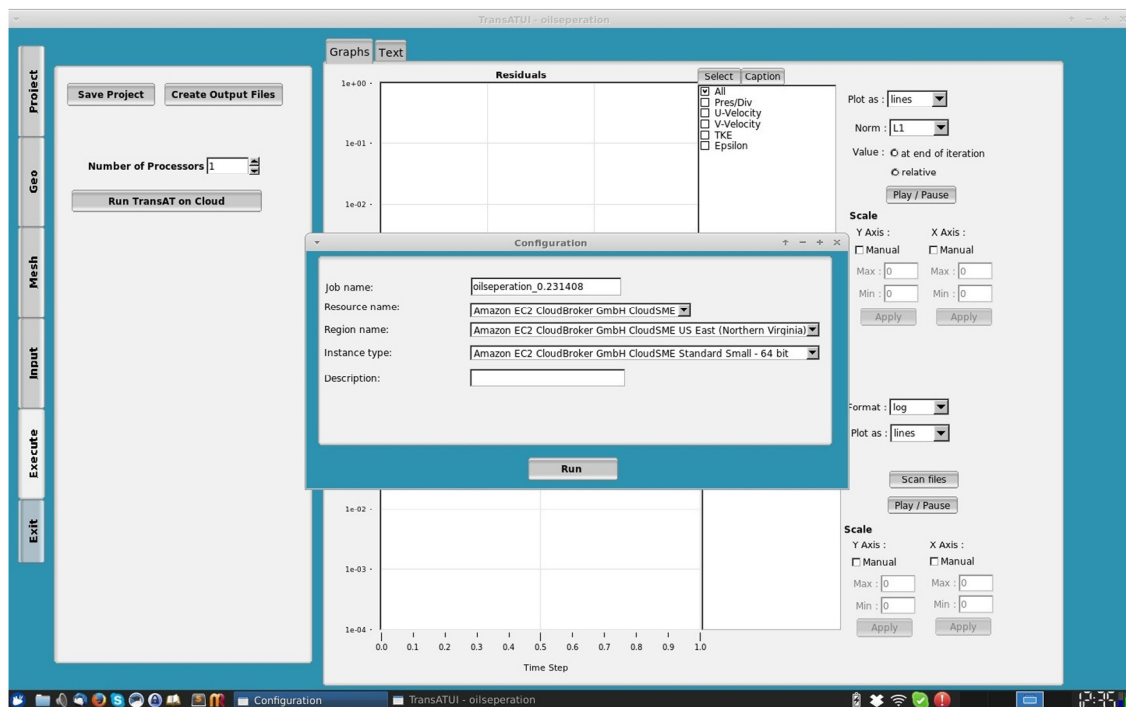


Fig. 10. Mapping a TransAT simulation to cloud resources via CloudBroker.

upload n number of XML files in the workflow graph shown in Fig. 11, the Generator will start in parallel n instances in Cloud-I and n instances in Cloud-II. Input files for the n instances are channelled from the Generator node. When the runs are finished the outputs are collected in the Collector node.

The WS-PGRADE/gUSE web interface provides a list of all concrete workflows in a user's account with the options to configure and submit a workflow. A user can also use the web interface to monitor the status of a submitted workflow and a list of all the jobs instantiated by the submitted workflow and their status. When a job is finished the output files can be downloaded from the web interface.

Fig. 12 shows the interactions between the user, WS-PGRADE/gUSE and CloudBroker. The first three interactions are optional. These steps take place only when creating a new workflow or reconfigure an existing one. Jobs submission starts when the user press the submit button (4) at the WS-PGRADE/gUSE web interface. Then interactions (5–13) and (16–17) are between WS-PGRADE/gUSE and CloudBroker via the DCI Bridge. All this activity is completely transparent to the user with CloudBroker instantiating the virtual machine where RepastS is deployed (5), uploading the model and parameters and executing the simulation (7–10). If the user wishes to see the details of a job (14–15), s/he presses the details button and the status is updated at the

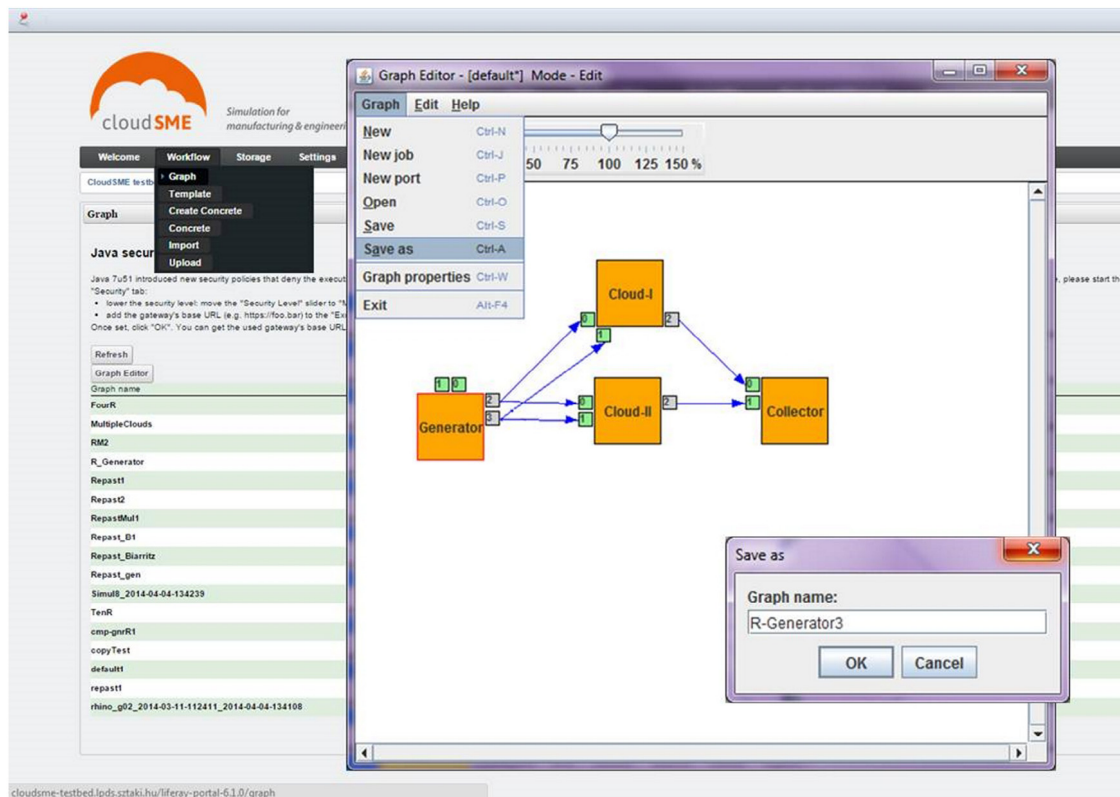


Fig. 11. Workflow graph creation in WS-PGRADE/gUSE web interface. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

WS-PGRADE/gUSE interface. Finally, the user can download the output files (18–20) by pressing the download file output button in the detailed view of a job.

4.2.2. Case study

To test the performance of parameter sweeps for simulation on multiple clouds, we used a neutral case study that represents a generic example of discrete simulation with parameter sweep (i.e. a simulation is run multiple times with different combinations of input parameters). This is an agent-based simulation, technique that is widely used to study complex systems and emergent behaviours from the individual agents' interactions. It represents infectious disease simulation, one example of many systems that agent-based simulation can be used to study [37] and illustrates how simulation can be used to study the changes in individual behaviour and the impact on the spread of a disease. The simulation consists of three types of agents that move in an environment and interact with each other. The agents represent the susceptible, infected and recovered population. The model starts an infection outbreak with an initial population of infected and susceptible agents. Infected agents move close to susceptible agents and infect them while susceptible agents move where the least infected agents are located. Infected and susceptible agents interact with each other in every simulation time unit which is a day in our simulation. Infected agents recover after a period of time and become recovered with a level of immunity. When an infected agent gets in touch with a susceptible agent, the susceptible agent becomes infected. When an infected agent gets in touch with a recovered agent, the recovered agent decreases its immunity. When the immunity level is 0, the recovered agent becomes susceptible and can be infected again. The outbreak occurs annually. When this happens, the population changes to reflect the initial conditions taking into account the population dynamics of the previous year.

We conducted a series of experiments on two cloud resources: the Amazon EC2 commercial cloud and an academic cloud offered by the University of Westminster (UoW), UK. The experiments were performed on cloud instances of various sizes, the specifications of which are shown in Table 2. Each experiment was set up in WS-PGRADE/gUSE by quickly reconfiguring the workflow by selecting a different cloud/instance type.

Our demonstration consisted of a parameter sweep consisting of ten runs (i.e. ten simulations with a different parameter). We conducted ten experiments. These took approximately 200 min to run on a desktop PC (i5-2500 processor at 3.30 GHz speed and 4.00 GB RAM). We ran these experiments on one, two, five and 10 instances of each cloud type. The experiments were distributed equally when run on more than one instances. Fig. 13 shows the comparative runtime by instance and Table 3 shows the speedup when compared to a single PC run. The run-time is the average of five runs. From the results, we observe that Amazon EC2 instances have relatively stable performance and the academic cloud presents a larger variation. For example, five instances of U2 perform worse than two. Also, we have a considerable increase in execution time when running on 10 instances. Types U3 and U4 show similar behaviour. This is suspected to be rooted in variations in resource availability that cause job requests to be queued until resources are available. Similar behaviour with less variation is shown by A3 where the execution time for five instances is increased. In terms of speedup, when running on a single instance for all cloud types in this experiment, apart from A5, the performance is slower than a desktop machine. This is expected since there is an overhead for starting up the virtual machines. Most of the larger instances, at least the commercial ones, present modest speedup. It is expected that for larger simulations there will be better considerable speedup as the longer processing time will compensate the overheads of setting up virtual instances on a cloud. Overall this shows how a user might investigate different

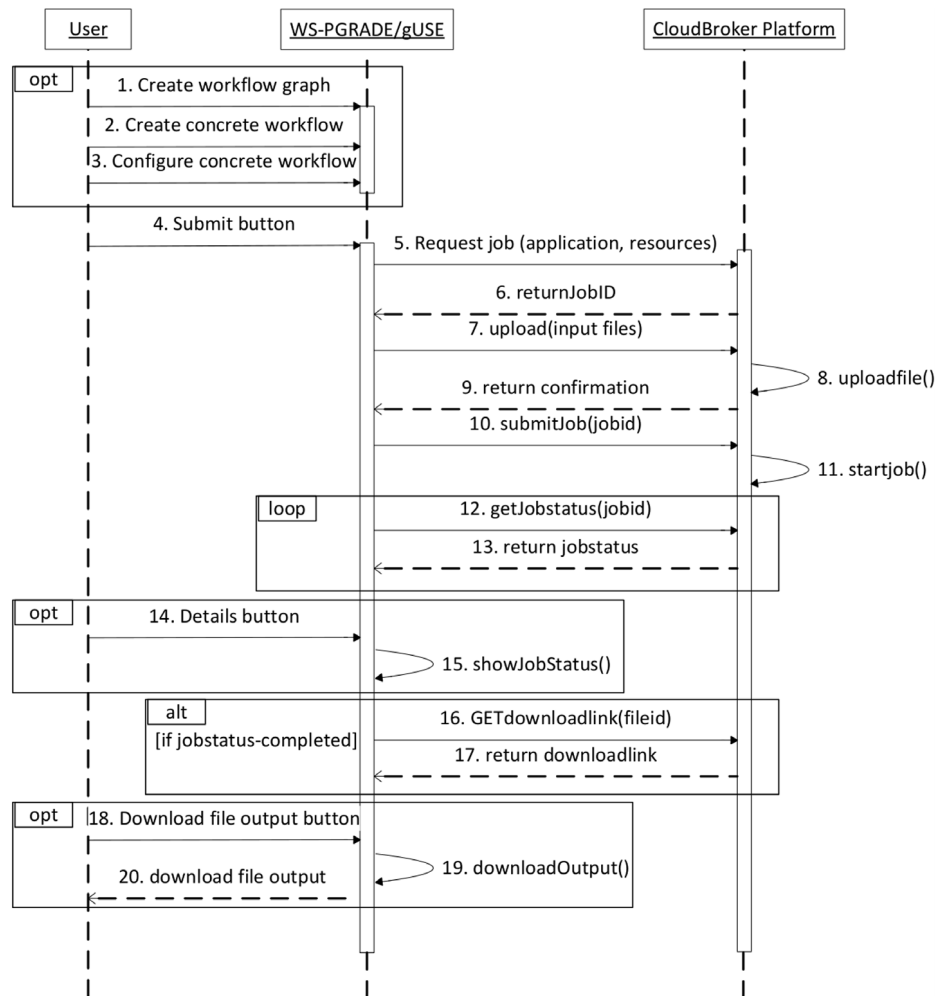


Fig. 12. WS-PGRADE/gUSE and CloudBroker interactions for RepastS jobs.

Table 2
Cloud resources characteristics.

Cloud instance	Number of vCPUs	Processor type	Memory
Amazon baseline micro (A1)	1	High Frequency Intel Xeon Processors with Turbo up to 3.3 GHz	0.5 GiB
Amazon baseline small (A2)	1	High Frequency Intel Xeon Processors with Turbo up to 3.3 GHz	1 GiB
Amazon baseline medium (A3)	2	High Frequency Intel Xeon Processors with Turbo up to 3.3 GHz	4 GiB
Amazon balanced medium (A4)	1	High Frequency Intel Xeon E5-2670 v2 at 2.6 GHz	3.75 GiB
Amazon balanced large (A5)	2	High Frequency Intel Xeon E5-2670 v2 at 2.6 GHz	7.5 GiB
UoW small (U1)	1	AMD Opteron 4122 Processor at 2.2 GHz	20 MB
UoW medium (U2)	2	AMD Opteron 4122 Processor at 2.2 GHz	40 MB
UoW large (U3)	4	AMD Opteron 4122 Processor at 2.2 GHz	80 MB
UoW XL (U4)	8	AMD Opteron 4122 Processor at 2.2 GHz	160 MB

cloud and instance types to choose which is the best for his or her needs.

In terms of commercial use, a developer would follow the same process of application deployment on CloudBroker and workflow development on WS-PGRADE/gUSE. The developer then has the option of creating a dedicated web-based interface using the ASM module of WS-PGRADE/gUSE or by integrating with an existing commercial desktop PC application using the gUSE Remote API. The cloud-based simulation application can then be made available through the AppCenter. Such a commercial application has been implemented based on the Simul8 discrete event simulation package.⁸ The Simul8 desktop package has been extended with

cloud-based execution support using the gUSE Remote API. Additionally, a web portal has also been implemented that utilises Simul8 to support scheduling of production and distribution of products for microbreweries.

5. Conclusions and future work

This paper has described how the CloudSME Simulation Platform can be used to create commercial multi-cloud-based simulation applications using a combination of technologies including the WS-PGRADE/gUSE gateway framework, the CloudBroker Platform and the CloudSME AppCenter. Two case studies have shown how this multiple approach can be used to deploy parallel multiprocessor simulation applications and parameter sweeps. Please note that these two case studies are only representative examples and many

⁸ www.simul8.com.

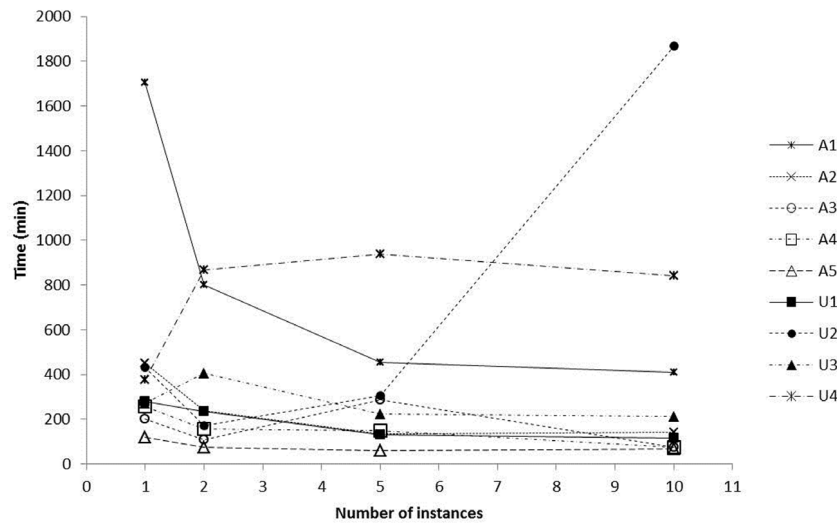


Fig. 13. Cloud-based RepastS infection model performance.

Table 3
Cloud-based RepastS infection model speedup.

Clouds	A1	A2	A3	A4	A5	U1	U2	U3	U4	
Instances	1	0.12	0.44	0.99	0.78	1.68	0.72	0.46	0.74	0.53
	2	0.25	0.84	1.82	1.28	2.67	0.85	1.18	0.49	0.23
	5	0.44	1.49	0.70	1.32	3.39	1.53	0.66	0.89	0.21
	10	0.49	1.40	2.63	2.67	2.99	1.71	0.11	0.93	0.24

other applications of the technology are possible. For example, Chang [38] shows the potential of cloud and medical imaging simulations and Hamid, et al. [39] discusses how systems consisting of multiple cores (i.e. cloud instances) can be used to support hybrid information systems. These and similar applications can potentially be migrated into the CloudSME Simulation Platform and their efficiency can be evaluated in the future.

Future work continues in two directions. The CSSP continues to be improved based on recent research. Distributed simulation is a discrete simulation technique used to either speed up large single simulations or to link together different simulations. The workflow aspect of the CSSP has been investigated to determine if it can be used to automatically compose distributed simulation federations [40]. Work continues in this area to develop parameter sweeps of distributed simulations. Parameter sweeps or large scale simulations can produced large amounts of results. Research has investigated how the CSSP could be extended to support MapReduce applications for large scale data processing [41]. Based on this research work continues in using MapReduce and other related techniques such as clustering and classification and ranking and selection to speed up the processing of simulation results. Approaches to auto-scaling cloud resources using micro services and cloud orchestration are being developed using the MiCADO system [42]. This approach is being extended to investigate efficient scaling techniques for parameter sweeps with time constraints. Work also continues in developing science gateways for the large agent-based simulation community based on [43]. Another potential research direction is the use of the platform to support simulation optimisation. Nelson [44] has identified the impact of cloud computing and associated technologies on optimisation and simulation analytics. For example, Multi-objective Particle Swarm Optimisation involves dividing the simulation search space normally explored by a parameter sweep into subpopulations that are then searched in parallel [45]. The creation of a cloud-based service using this form of multi-objective optimisation would be highly attractive as it could potentially identify the optimal solutions

in a Pareto front faster than a simple parameter sweep. Further cloud resources are also being added to the Platform, in particular dedicated resources from HPC centres.

Based on experiences with BeerOpt case study in the CloudSME project, Saker Solutions, Hobsons Brewery and Brunel University are jointly developing the CraftBrew Whole Brewery Management System that is using cloud-based approaches to improve efficiencies in microbreweries. Further commercial applications are also being created by a start-up company, CloudSME UG, that was founded by nine stakeholders from six European countries, all of them were partners in the CloudSME project. CloudSME UG markets the already available solutions in the AppCenter and successfully pursues further commercial contracts based on the technology.

Acknowledgement

The work presented in this paper was supported by the following projects: EU FP7 CloudSME project (Project no. 608886), EU H2020 COLA project (Project no. 731574), and EU H2020 Cloudi- Facturing project (Project no. 768892).

References

- [1] S. Robinson, Simulation: The Practice of Model Development and Use, second ed., Palgrave Macmillan UK, 2014. <https://he.palgrave.com/page/detail/Simulation/?K=9781137328021> (accessed 18.07.17).
- [2] A.M. Law, Simulation Modeling and Analysis, fifth ed., McGraw-Hill, 2015.
- [3] M. Griebel, T. Dornsiefer, T. Neunhoffer, Numerical Simulation in Fluid Dynamics, Society for Industrial and Applied Mathematics, 1998. <http://dx.doi.org/10.1137/1.9780898719703>.
- [4] M. Frank, C. Laroque, T. Uhlig, Reducing computation time in simulation-based optimization of manufacturing systems, in: Proc. 2013 Winter Simul. Conf., 2013, pp. 2710–2721. <http://dx.doi.org/10.1109/WSC.2013.6721642>.
- [5] D.Z. Zhang, A.I. Anosike, Modelling and simulation of dynamically integrated manufacturing systems, J. Intell. Manuf. 23 (2012) 2367–2382. <http://dx.doi.org/10.1007/s10845-010-0494-0>.
- [6] P. Heidelberger, Statistical Analysis of Parallel Simulation, in: Proc. 1986 Winter Simul. Conf., 1986, pp. 2278–2288.

- [7] P. Mell, T. Grance, *The NIST Definition of Cloud Computing*, National Institute of Standards and Technology, Gaithersburg, MD, 2011. <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>.
- [8] P. Maresova, V. Sobeslav, O. Krejcar, Cost–benefit analysis–evaluation model of cloud computing deployment for use in companies, *Appl. Econ.* 49 (2017) 521–533. <http://dx.doi.org/10.1080/00036846.2016.1200188>.
- [9] P. Kacsuk, Z. Farkas, M. Kozlovsky, G. Hermann, A. Balasko, K. Karóczkai, I. Márton, WS-PGRADE/gUSE generic DCI gateway framework for a large variety of user communities, *J. Grid Comput.* 10 (2012) 601–630. <http://dx.doi.org/10.1007/s10723-012-9240-5>.
- [10] J.D. Anderson, *Computational Fluid Dynamics: The Basics with Applications*, McGraw-Hill, 1995.
- [11] M. Hanai, T. Suzumura, A. Ventresque, K. Shudo, An adaptive VM provisioning method for large-scale agent-based traffic simulations on the cloud, in: *Proc. Int. Conf. Cloud Comput. Technol. Sci.*, 2015, pp. 130–137. <http://dx.doi.org/10.1109/CloudCom.2014.164>.
- [12] D. Zehe, A. Knoll, W. Cai, H. Aydt, SEMSim cloud service: Large-scale urban systems simulation in the cloud, *Simul. Model. Pract. Theory* 58 (2015) 157–171. <http://dx.doi.org/10.1016/j.simpat.2015.05.005>.
- [13] M. Carillo, G. Cordasco, F. Serrapica, C. Spagnuolo, P. Szufel, L. Vicidomini, D-MASON on the cloud: An experience with amazon web services, 2017. http://dx.doi.org/10.1007/978-3-319-58943-5_26.
- [14] K. Anderson, J. Du, A. Narayan, A.E. Gamal, GridSpice: A distributed simulation platform for the smart grid, *IEEE Trans. Ind. Inform.* 10 (2014) 2354–2363. <http://dx.doi.org/10.1109/TII.2014.2332115>.
- [15] R. Jia, B. Sundén, Parallelization of a multi-blocked CFD code via three strategies for fluid flow and heat transfer analysis, *Comput. & Fluids* 33 (2004) 57–80. [http://dx.doi.org/10.1016/S0045-7930\(03\)00029-X](http://dx.doi.org/10.1016/S0045-7930(03)00029-X).
- [16] S. Guzzetti, T. Passerini, J. Slawinski, U. Villa, A. Veneziani, V. Sunderam, Platform and algorithm effects on computational fluid dynamics applications in life sciences, *Future Gener. Comput. Syst.* 67 (2017) 382–396. <http://dx.doi.org/10.1016/j.future.2016.03.024>.
- [17] J. Slawinski, T. Passerini, U. Villa, A. Veneziani, V. Sunderam, Experiences with target-platform heterogeneity in clouds, grids, and on-premises resources, in: *2012 IEEE 26th Int. Parallel Distrib. Process. Symp. Work. PhD Forum*, IEEE, 2012, pp. 41–52. <http://dx.doi.org/10.1109/IPDPSW.2012.20>.
- [18] R. Ledyayev, H. Richter, High performance computing in a cloud using openstack, in: *Cloud Comput. 2014 Fifth Int. Conf. Cloud Comput. GRIDs Virtualization*, 2014, pp. 108–113. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.685.9007> (accessed 17.12.17).
- [19] P. OLeary, M. Christon, S. Jourdain, C. Harris, M. Berndt, A. Bauer, HPCCloud: A cloud/web-based simulation environment, in: *2015 IEEE 7th Int. Conf. Cloud Comput. Technol. Sci.*, IEEE, 2015, pp. 25–33. <http://dx.doi.org/10.1109/CloudCom.2015.33>.
- [20] E. Deelman, K. Vahi, M. Rynge, G. Juve, R. Mayani, R.F. Da Silva, Pegasus in the cloud: Science automation through workflow technologies, *IEEE Internet Comput.* 20 (2016) 70–76. <http://dx.doi.org/10.1109/MIC.2016.15>.
- [21] K. Wolstencroft, R. Haines, D. Fellows, A. Williams, D. Withers, S. Owen, S. Soiland-Reyes, I. Dunlop, A. Nenadic, P. Fisher, J. Bhagat, K. Belhajjame, F. Bacall, A. Hardisty, A. Nieva de la Hidaiga, M.P. Balcazar Vargas, S. Sufi, C. Goble, The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud, *Nucleic Acids Res.* 41 (2013) W557–61. <http://dx.doi.org/10.1093/nar/gkt328>.
- [22] P. Nowakowski, M. Bubak, T. Bartysiński, T. Gubała, D. Haręźlak, M. Kasztelnik, M. Malawski, J. Meizner, Cloud computing infrastructure for the VPH community, *J. Comput. Sci.* (2017). <http://dx.doi.org/10.1016/j.jocs.2017.06.012>.
- [23] E. Afgan, D. Baker, N. Coraor, H. Goto, I.M. Paul, K.D. Makova, A. Nekrutenko, J. Taylor, Harnessing cloud computing with Galaxy cloud, *Nat. Biotechnol.* 29 (2011). <http://dx.doi.org/10.1038/nbt.2028>.
- [24] I.F. Senturk, P. Balakrishnan, A. Abu-Doleh, K. Kaya, Q. Malluhi, Ü.V. Çatalyürek, A resource provisioning framework for bioinformatics applications in multi-cloud environments, *Future Gener. Comput. Syst.* 78 (2018). <http://dx.doi.org/10.1016/j.future.2016.06.008>.
- [25] S. Maddineni, J. Kim, Y. El-Khamra, S. Jha, Distributed application runtime environment (DARE): A standards-based middleware framework for science gateways, *J. Grid Comput.* 10 (2012). <http://dx.doi.org/10.1007/s10723-012-9244-1>.
- [26] B. Akos, F. Zoltan, K. Peter, P. Kacsuk, Building science gateways by utilizing the generic WS-PGRADE/gUSE workflow system, *Comput. Sci.* 14 (2013) 307–325. <http://dx.doi.org/10.7494/csci.2013.14.2.307>.
- [27] Z. Farkas, P. Kacsuk, A. Hajnal, WS-PGRADE/gUSE and clouds, in: P. Kacsuk (Ed.), *Sci. Gateways Distrib. Comput. Infrastructures*, Springer, Cham, 2014, pp. 97–110.
- [28] D. Temelkovski, T. Kiss, G. Terstyanszky, Molecular docking with Raccoon2 on clouds: extending desktop applications with cloud computing, in: *Th Int. Work. Sci. Gateways*, 2017, pp. 19–21. <http://www.westminster.ac.uk/westminsterresearch> (accessed 17.12.17).
- [29] S. Forli, R. Huey, M.E. Pique, M.F. Sanner, D.S. Goodsell, A.J. Olson, Computational protein–ligand docking and virtual drug screening with the AutoDock suite, *Nat. Protoc.* 11 (2016) 905. <http://dx.doi.org/10.1038/nprot.2016.051>.
- [30] T. Kiss, P. Kacsuk, E. Takacs, A. Szabo, P. Tihanyi, S.J.E. Taylor, Commercial use of WS-PGRADE/gUSE, in: P. Kacsuk (Ed.), *Sci. Gateways Distrib. Comput. Infrastructures Dev. Framew. Exploit. By Sci. User Communities*, Springer, Cham, 2014, pp. 271–286. <http://dx.doi.org/10.1007/978-3-319-11268-8-19>.
- [31] M. Kozlovsky, K. Karóczkai, I. Márton, P. Kacsuk, T. Gottdank, DCI bridge: Executing WS-PGRADE workflows in distributed computing infrastructures, in: *Sci. Gateways Distrib. Comput. Infrastructures*, Springer International Publishing, Cham, 2014, pp. 51–67. http://dx.doi.org/10.1007/978-3-319-11268-8_4.
- [32] Á. Hajnal, I. Márton, Z. Farkas, P. Kacsuk, Remote storage management in science gateways via data bridging, *Concurr. Comput. Pract. Exp.* 27 (2015) 4398–4411. <http://dx.doi.org/10.1002/cpe.3520>.
- [33] P. Kacsuk (Ed.), *Science Gateways for Distributed Computing Infrastructures*, Springer International Publishing, Cham, 2014. <http://dx.doi.org/10.1007/978-3-319-11268-8>.
- [34] G. Terstyanszky, T. Kiss, S. Taylor, A. Anagnostou, M. Subira, G. Padula, E.D.M. Alonso, J.M.M. Rapun, Validating scanned foot images and designing customized insoles on the cloud, in: *2016 49th Hawaii Int. Conf. Syst. Sci.*, IEEE, 2016, pp. 3288–3296. <http://dx.doi.org/10.1109/HICSS.2016.411>.
- [35] P. Hanafizadeh, M.H. Saidi, A. Nouri Gheimesi, S. Ghanbarzadeh, Experimental investigation of air–water, two-phase flow regimes in vertical mini pipe, *Sci. Iran.* 18 (2011) 923–929. <http://dx.doi.org/10.1016/j.SCIENT.2011.07.003>.
- [36] M.J. North, N.T. Collier, J. Ozik, E.R. Tataru, C.M. Macal, M. Bragen, P. Sydelko, Complex adaptive systems modeling with Repast Simphony, *Complex Adapt. Syst. Model.* 1 (2013) 3. <http://dx.doi.org/10.1186/2194-3206-1-3>.
- [37] C.M. Macal, Everything you need to know about agent-based modelling and simulation, *J. Simul.* 10 (2016) 144–156. <http://dx.doi.org/10.1057/jos.2016.7>.
- [38] V. Chang, Computational intelligence for medical imaging simulations, *J. Med. Syst.* 42 (2018). <http://dx.doi.org/10.1007/s10916-017-0861-x>.
- [39] N. Hamid, V. Chang, R.J. Walters, G.B. Wills, A Multi-core architecture for a hybrid information system, *Comput. Electr. Eng.* (2017). <http://dx.doi.org/10.1016/j.compeleceng.2017.12.020>.
- [40] N.R. Chaudhry, A. Nouman, A. Anagnostou, S.J.E. Taylor, WS-PGRADE workflows for cloud-based distributed simulation, in: *Proc. Oper. Res. Soc. Simul. Work. Vol.* 2016, 2016, pp. 192–201.
- [41] S. Gugnani, C. Blanco, T. Kiss, G. Terstyanszky, Extending science gateway frameworks to support big data applications in the cloud, *J. Grid Comput.* 14 (2016) 589–601. <http://dx.doi.org/10.1007/s10723-016-9369-8>.
- [42] T. Kiss, P. Kacsuk, J. Kovacs, B. Rakoczi, A. Hajnal, A. Farkas, G. Gesmier, G. Terstyanszky, MiCADO—Microservice-based cloud application-level dynamic orchestrator, *Future Gener. Comput. Syst.* (2017). <http://dx.doi.org/10.1016/j.future.2017.09.050>.
- [43] S.J.E. Taylor, A. Anagnostou, T. Kiss, G. Terstyanszky, P. Kacsuk, N. Fantini, A tutorial on cloud computing for agent-based modeling & simulation with repast, in: *Proc. 2015 Winter Simul. Conf.*, 2015, pp. 192–206. <http://dx.doi.org/10.1109/WSC.2014.7019888>.
- [44] B.L. Nelson, “Some tactical problems in digital simulation” for the next 10 years, *J. Simul.* 10 (2016). <http://dx.doi.org/10.1057/jos.2015.22>.
- [45] A. Atashpendar, B. Dorronsoro, G. Danoy, P. Bouvry, A scalable parallel cooperative coevolutionary PSO algorithm for multi-objective optimization, *J. Parallel Distrib. Comput.* 112 (2018) 111–125. <http://dx.doi.org/10.1016/j.jpdc.2017.05.018>.



Simon J E Taylor is a Professor of Computing and the director of the Modelling & Simulation Group in the Department of Computer Science, Brunel University London (<https://tinyurl.com/ya5zjh8z>). He has a successful track record of research in Modelling & Simulation, particularly with high performance distributed simulation in industry through digital infrastructures. He has developed these solutions with over 20 companies resulting in over 5M GBP in cost savings and increased production. He has successfully applied these experiences in the development of digital infrastructures in Africa and has led innovation in over 10 African countries. He has also led the development of international standards in these areas. He regularly audits and chairs commercial project reviews in distributed computing and Modelling & Simulation. He is a member of the ACM SIGSIM Steering Committee and founder of the *Journal of Simulation*. He has chaired several major conferences and his published over 150 articles.



Tamas Kiss is a Reader in Distributed Computing at the Department of Computer Science and Deputy Director of the University Research Centre for Parallel Computing at the University of Westminster. He holds Masters Degrees in Electrical Engineering, and Computer Science and Mathematics, and Ph.D. in Distributed Computing. His research interests include distributed and parallel computing, cloud, cluster and grid computing. He has attracted over £20 Million research funding and leads national and European research projects related to enterprise applications of cloud computing technologies. He co-authored

more than 100 papers in journals, conference proceedings and as chapters of edited books.



Anastasia Anagnostou is a Research Fellow at the Modelling & Simulation Group at the Department of Computer Science, Brunel University London. She holds a Ph.D. in Hybrid Distributed Simulation, an M.Sc. in Telemedicine and e-Health Systems and a B.Sc. in Electronics Engineering. Her research interests are related to the application of modelling and simulation techniques in the Healthcare and Industry, as well as to contemporary infrastructures for enhancing simulation experimentation. She has been involved in developing simulation applications and HPC infrastructures projects across Europe and Africa.



Gabor Terstyanszky is the Director of the University Research Centre for Parallel Computing (CPC) Centre at the University of Westminster. The Centre specialises in research on large-scale distributed computing infrastructures (DCIs) addressing research challenges of High Performance Computing (HPC) and High-Throughput Computing (HTC), such as clouds, clusters and grids whereby large numbers of computing resources are used to process computationally-demanding problems. His research interests include distributed and parallel computing focusing on targeting research issues of Big Data and Cloud

Computing. He has been involved in more than 15 research projects as either Principal or Co-Investigator. He also had several research grants at various universities in Germany, Spain, and United Kingdom. He published over 140 papers at conferences and journals. He was member of programming committees of several conferences and workshops.



Peter Kacsuk is the Director of the Laboratory of the Parallel and Distributed Systems in the Computer and Automation Research Institute of the Hungarian Academy of Sciences (MTA SZTAKI). He received his M.Sc. and university doctorate degrees from the Technical University of Budapest in 1976 and 1984, respectively. He received the kandidat degree (equivalent to Ph.D.) from the Hungarian Academy in 1989. He habilitated at the University of Vienna in 1997. He received his professor title from the Hungarian President in 1999 and the Doctor of Academy degree (DSc) from the Hungarian Academy of Sciences in

2001. He served as full professor at the University of Miskolc and at the Eötvös Lóránd University of Science Budapest. He has been a part-time full professor at the University of Westminster (London, UK) since 2001. He has published two books, two lecture notes and more than 350 scientific papers on parallel computer architectures, parallel software engineering, Grid and Cloud computing. He is editor-in-chief of the Journal of Grid Computing published by Springer.



Joris Costes is currently an aerospace thermomechanical engineer at ArianeGroup near Bordeaux, France. He earned his engineering degree at ENSTA ParisTech, Paris, France and a Master of science in applied mathematics in 2011 at ENS Paris-Saclay. He had the opportunity to pursue doctoral studies in numerical methods and scientific computing working with Prof. Jean-Michel Ghidaglia at ENS Paris-Saclay. Then he worked as an engineer and a project leader at Eurobios, a small company where he developed solutions in scientific computing for industrial companies.

He moved to ArianeGroup in 2017 and is still associate researcher at ENS Paris-Saclay, CMLA laboratory.



Nicola Fantini is the CEO of the CloudBroker GmbH, CH, and has been software engineer, project manager and entrepreneur in the field of software development for more than 20 years. He holds an M.Sc. degree in Computer Science and Business Administration from the University of Zurich, CH. He has been involved in and leading projects across industries, providing operational, commercial, financial as well as technical solutions.