Project acronym: **CoolEmAll**

Project full title: **Platform for optimising the design and operation of modular configurable IT infrastructures and facilities with resource-efficient cooling**

SEVENTH FRAMEWORK PROGRAMME

D2.1 Report about state-of-the-art simulation and visualisation environments

Author: Andreas Kopecki (USTUTT-HLRS)

Version: 1.0

Date: 31/01/2012

Deliverable Number:	D2.1
Contractual Date of Delivery:	31/01/2012
Actual Date of Delivery:	31/01/2012
Title of Deliverable:	Report about state-of-the-art simulation and visualisation environments
Dissemination Level:	Public
WP contributing to the Deliverable:	WP 2
Author:	Andreas Kopecki
Co-Authors:	Tomasz Kuczynski, Juan Luis Prieto Martinez, Assunta Napolitano, Wojtek Piatek, Ariel Oleksiak, Jean-Marc Pierson, Fabio Luiz Tumiatti

History			
Version	Date	Author	Comments
0.1	28/11/2011	Andreas Kopecki	First template version
0.2	09/12/2011	Andreas Kopecki	Draft version
0.3	12/12/2011	Andreas Kopecki	Draft version
0.4	20/12/2011	Andreas Kopecki, Wojtek Piatek, Tomasz Kuczynski	Draft version
0.5	09/01/2012	Andreas Kopecki, Wojtek Piatek, Juan Luis Prieto Martinez	Draft version
0.6	11/01/2012	Andreas Kopecki	Draft version
0.7	11/01/2012	Andreas Kopecki, Wojtek Piatek	Draft version
0.8	11/01/2012	Andreas Kopecki	Draft version
0.9	12/01/2012	Andreas Kopecki, Assunta Napolitano	Draft version
0.10	16/01/2012	Andreas Kopecki, Tomasz Kuczynski, Assunta Napolitano	Draft version

0.11	16/01/2012	Andreas Kopecki, Wojtek Piatek	Draft version
0.12	16/01/2012	Andreas Kopecki, Juan Luis Prieto Martinez	Draft version
0.13	16/01/2012	Andreas Kopecki	Final draft for QM
0.14	27/01/2012	Andreas Kopecki, Ariel Oleksiak, Jean-Marc Pierson, Fabio Luiz Tumiatti, Wojtek Piatek, Assunta Napolitano	Quality Control Feedback
0.15	31/01/2012	Andreas Kopecki, Ariel Oleksiak	Quality Control Feedback
1.0	31/01/2012	Andreas Kopecki	Final Version

Approval		
Date	Name	Signature
31/01/12	Ariel Oleksiak	

Abstract

One of the major goals of CoolEmAll is to give IT planning staff and decision makers a tool for quantitative assessment of the impact of equipment on the cooling situation in their existing or emerging infrastructure. Arranging racks by careful planning and experience can already yield a balanced air conditioning. But for catching all hot spots, more sophisticated tools are needed. The simulation of the heat dissipation and airflow in the computing room can help to identify all critical hot spots and lead to a better and overall more energy efficient cooling. Advanced visualisation tools that offer a tight loop with the running simulations shorten the time needed for the analysis of the simulation results. This document looks at the current state-of-the-art in simulation, post-processing and visualisation to found the basis for a CoolEmAll Simulation, Visualisation and Decision Toolkit.

Keywords

Simulation, Visualisation, SVD Toolkit, Workload Simulation

Table of Contents

1 The Simulation, Visualisation and Decision Toolkit

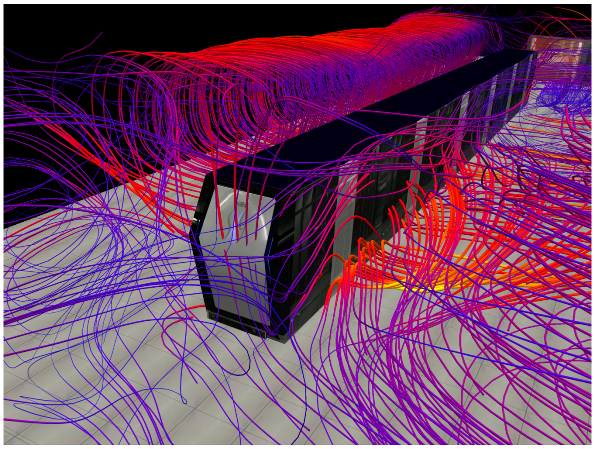


Figure 1: Data centre simulation using CFX

One of the results of the CoolEmAll project will be the development of a simulation, visualisation, and decision support toolkit (SVD Toolkit) for analysis and design of modular IT infrastructures and facilities with resource-efficient cooling. This platform will support IT infrastructure designers, decision makers and administrators in a process of planning new infrastructures or improving the existing ones. The platform will enable the optimisation of both the IT infrastructure design and operation. A modular approach to building IT infrastructures and facilities provides many extension possibilities and a high level of customisation. CoolEmAll will develop a flexible simulation platform integrating models of applications, workload scheduling policies, hardware characteristics, cooling as well as air and thermal flows using computational fluid dynamics (CFD) simulation tools. The flexibility of these models, based on model parameter settings, will ensure the flexibility of the entire CoolEmAll SVD platform. This platform is aimed at capturing the required model settings and simulating these models for a wide range of target applications, workload scheduling policies, IT-Infrastructure, and hardware characteristics. Advanced visualisation tools and user interfaces will allow users to easily analyse various options and optimise energy efficiency of planned IT infrastructures and facilities. The toolkit will also support users to realise a thermal- and energy-aware workload scheduling and resource management. The main CoolEmAll outcomes will be based on existing technologies, which will be extended appropriately to meet the objectives. Thus, a main focus when designing the toolbox will be the features already provided by possible integration candidates as well as the expertise available with the CoolEmAll partners. This document is going to take a look at the current state-of-the-art in simulation, simulation steering and visualisation to have a profound basis for finding the right building blocks when going to design the Simulation, Visualisation and Decision Toolkit. Based on this document it will be possible to put the initial design proposed in the description of work to the test when entering the design phase of the toolkit. This document will set the further course of Work Package 2 in regard of the simulations and visualisation platforms used in the upcoming implementation.

In the following chapters, existing models and tools are evaluated. The survey encompasses thermal-/fluid simulation codes as well as codes specifically targeted at data centres (Section 2), workload simulation models (Section 3) and visualisation packaged (Section 4). It closes with an assessment of the suitability of these tools as a basis for the emerging CoolEmAll SVD Toolkit (Section 5). It is not intended to propose a final decision on the tools of choice, but a guideline and handbook for all CoolEmAll partners designing the final toolkit, helping them in finding the right solutions fitting into the proposed SVD Toolkit design.

2 CFD Simulation Codes

Simulation codes will support the virtual design and analysis of data centre facilities and specific computing resources based on the compute box concept introduced within the CoolEmAll project. For example, using simulation codes, it will be possible to plan the layout and the arrangement of racks in a computing room or nodes within racks taking into consideration selected cooling solution. The SVD Toolkit will also enable assessing how specific applications, workloads, and their management policies affect the energy efficiency of a data centre. Thus, workload simulation tools are surveyed in Chapter 3.

There are a few aspects that are particularly important for the development of the CoolEmAll SVD Toolkit. Thermal dissipation has to be calculated as well as the airflow that is for most parts driven by the cooling of the compute room. Thus we will look at several state-of-the-art simulation codes and assess their suitability for CoolEmAll. The main aspects that drove the preliminary selection of the simulation code covered were the ease and accuracy of coupling both aspects (heat and flow) in the simulation. Others are the costs associated with the acquisition and operation of the code, as well as its extensibility for purposes of CoolEmAll.

In the following sections, two kinds of simulations will be introduced: (1) General purpose simulation packages that can be used to solve the CFD/heat transfer problems and (2) specialised software tools that are dedicated to data centre assessment. As all of them should be sufficiently suitable to solve coupled CFD/heat transfer problems, more important features regarding their suitability for CoolEmAll are as follows:

- Expertise of project partners related to the package,
- user routines for hooking up simulation steering,
- licensing & costs, and
- possible integration into the SVD framework, openness to external tools.

2.1 OpenFOAM

OpenFOAM [1] is not a complete and closed simulation package. Rather it is a C++ toolbox for the development of specific numerical solvers (Figure 2). It is maintained and released as open source under the GNU General Public License. OpenFOAM is generally used for CFD, but can also be applied to other areas like stress analysis, electromagnetics and finance. It amongst others includes solvers for ([2]):

- Incompressible and compressible flows,
- multiphase flows, and
- conjugate heat transfer.

A complete list of solvers can be accessed at [3].

Those readymade solvers can be used out of the box by linking them together in a program code. Additionally, own solvers can be implemented or existing ones can be changed by using the extensive solver library of OpenFOAM. As the OpenFOAM package also contains an extensive library for pre- and post-processing, it is possible to build a complete simulation solution by just using OpenFOAM.

OpenFOAM is first and foremost a C++ library, used primarily to create executables, known as applications. A central theme of the OpenFOAM design is that the solver applications, written using the OpenFOAM classes, have a syntax that closely resembles the partial differential equations being solved. For example the equation $\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \bullet \phi \mathbf{U} - \nabla \bullet \mu \nabla \mathbf{U} = -\nabla p$ is represented by the code

solve(fvm::ddt(rho, U) + fvm::div(phi, U) - fvm::laplacian(mu, U) == - fvc::grad(p));

This makes it quite straightforward to implement own solvers, although extending the library can be daunting due to heavy template use and a lack of documentation.

There are also some pre-made applications included in the OpenFOAM installation. These applications fall into two categories: solvers, that are each designed to solve a specific problem in continuum mechanics; and utilities, that are designed to perform tasks that involve data manipulation. The OpenFOAM distribution contains numerous solvers and utilities covering a wide range of problems.

Also for modelling a flow with heat transfer, several solvers are already implemented. For buoyant, turbulent flow of compressible fluids for ventilation and heat transfer, a steady-state (*buoyantSimpleFoam*) and a transient (*buoyantFoam*) solver exist. Calculation of the heat transfer by radiation can also be covered by using the *buoyantSimpleRadiationFoam* solver. For solving conjugate heat transfer problems, the *chtMultiRegionFoam* is available.

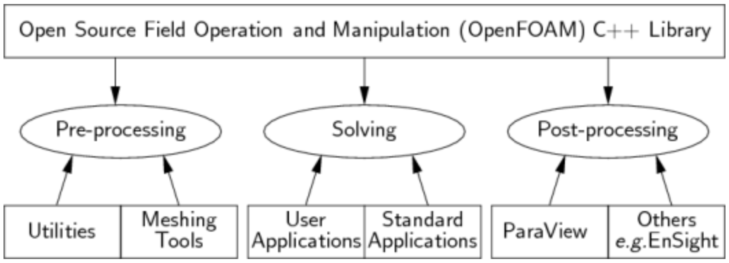


Figure 2: OpenFOAM library architecture

2.1.1 **Possible Suitability for the SVD**

OpenFOAM is a mature library for creating a large variation of solvers for many purposes. Its main advantage is the free availability of the models and the source code as well as its overall extensibility. It is free of charge and publicly available.

Commercial support is supplied by OpenCFD, a company owned by SGI. As the source code is available and modifiable, it is easy to integrate an OpenFOAM based simulation into emerging frameworks like the SVD. Documentation to the different solvers is scarce, though, thus the user has to rely on the community for support or buy a commercial license including support.

2.2 ANSYS CFD

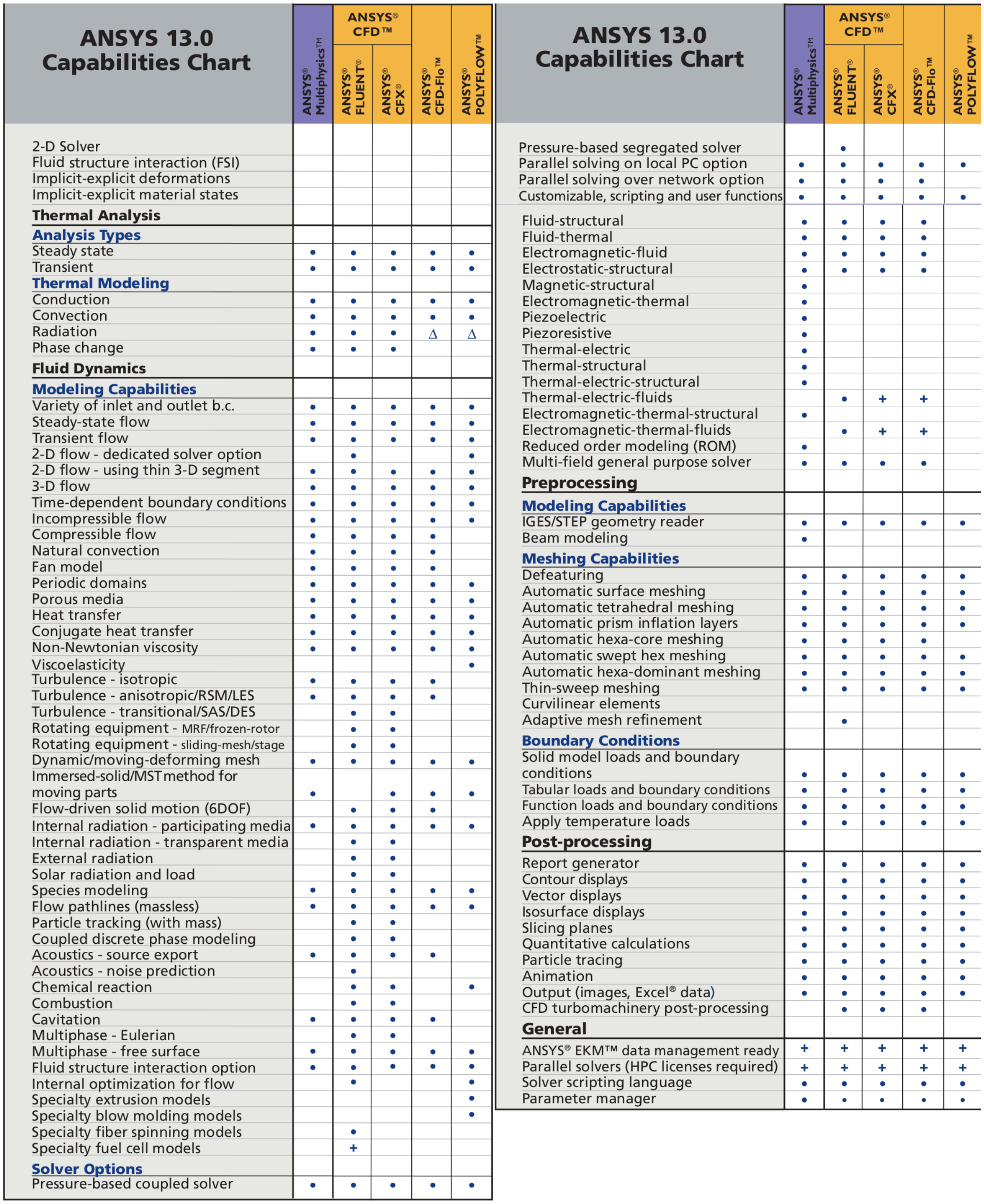
ANSYS CFD [4] is a Computer Aided Engineering (CAE) package containing both flagship solvers from ANSYS, CFX and FLUENT, thus offering a multidisciplinary approach to simulation in that fluid flow models integrates with other types of physics. Thanks to ANSYS Workbench technology, these two well-known products share a common workflow, such as CAD connectivity, geometry modelling, meshing, and post-processing.

The combination of ANSYS FLUENT and ANSYS CFX software provides users with an unprecedented level of fluid dynamics modelling capabilities. ANSYS CFX is a sophisticated CFD solver that enables to solve wide-ranging fluid flow problems of all levels of complexity; whereas ANSYS FLUENT is a general-purpose computational fluid dynamics software package to model flow, turbulence, heat transfer and reactions for industrial applications. For a complete list of the extensive capabilities, refer to Figure 3.

Besides predefined physics, customised applications can be modelled as well. Users can implement specialty physics and can also tailor and script the user environment to implement best practices or further automate the workflow. Thanks to an integrated solution for design of experiments (DOE), analysis of thousands of data points can be carried out in a single user environment, pursuing the optimum in several types of engineering applications design.

The post-processor for ANSYS CFD provides advanced quantitative and high-quality visual post-processing capabilities, including easy creation of charts, high-quality images and videos. ANSYS CFD provides powerful and scalable high-performance computing (HPC) options. Parallel scaling capabilities ensure that simulation efficiently utilises networks of homogenous or heterogeneous processors to maximise computation speed. ANSYS CFX has already been successfully used by COVISE (cf. Section 4.4) to do an in-situ visualisation of a simulation of the HLRS computing room.

ANSYS FLUENT is the simulation model base for CoolSim [5], a data centre cooling requirements assessment solution. It offers an easy to use graphical user interface combined with a library of common cooling and IT equipment for setting up the simulation and displaying the results to the user. CoolSim is deployed as Software as a Service. The model of the data centre can be built using dedicated software that can be run disconnected from the service. After creating the model, it is uploaded to the CoolSim HPC data centre, simulated there, and the results are returned to the user. The result is delivered as an HTML page that contains



ANSYS 13.0 Capabilities Chart	ANSYS® Multiphysics™	ANSYS® CFD™			
		ANSYS® FLUENT®	ANSYS® CFX®	ANSYS® CFD-Flo™	ANSYS® POLYFLOW™
2-D Solver					
Fluid structure interaction (FSI)					
Implicit-explicit deformations					
Implicit-explicit material states					
Thermal Analysis					
Analysis Types					
Steady state	•	•	•	•	•
Transient	•	•	•	•	•
Thermal Modeling					
Conduction	•	•	•	•	•
Convection	•	•	•	•	•
Radiation	•	•	•	Δ	Δ
Phase change	•	•	•		
Fluid Dynamics					
Modeling Capabilities					
Variety of inlet and outlet b.c.	•	•	•	•	•
Steady-state flow	•	•	•	•	•
Transient flow	•	•	•	•	•
2-D flow - dedicated solver option		•			•
2-D flow - using thin 3-D segment	•	•	•	•	•
3-D flow	•	•	•	•	•
Time-dependent boundary conditions	•	•	•	•	•
Incompressible flow	•	•	•	•	•
Compressible flow	•	•	•	•	
Natural convection	•	•	•	•	
Fan model	•	•	•	•	
Periodic domains	•	•	•	•	•
Porous media	•	•	•	•	•
Heat transfer	•	•	•	•	•
Conjugate heat transfer	•	•	•	•	•
Non-Newtonian viscosity	•	•	•	•	•
Viscoelasticity					•
Turbulence - isotropic	•	•	•	•	
Turbulence - anisotropic/RSM/LES	•	•	•	•	
Turbulence - transitional/SAS/DES		•	•		
Rotating equipment - MRF/frozen-rotor		•	•		
Rotating equipment - sliding-mesh/stage		•	•		
Dynamic/moving-deforming mesh	•	•	•	•	•
Immersed-solid/MST method for moving parts	•		•	•	•
Flow-driven solid motion (6DOF)		•	•	•	
Internal radiation - participating media	•	•	•	•	•
Internal radiation - transparent media		•	•		
External radiation		•	•		
Solar radiation and load		•	•		
Species modeling	•	•	•	•	•
Flow pathlines (massless)	•	•	•	•	•
Particle tracking (with mass)		•	•		
Coupled discrete phase modeling		•	•		
Acoustics - source export	•	•	•	•	
Acoustics - noise prediction		•			
Chemical reaction		•	•		•
Combustion		•	•		
Cavitation	•	•	•	•	
Multiphase - Eulerian		•	•		
Multiphase - free surface	•	•	•	•	•
Fluid structure interaction option	•	•	•	•	•
Internal optimization for flow		•			•
Specialty extrusion models					•
Specialty blow molding models					•
Specialty fiber spinning models		•			
Specialty fuel cell models		+			
Solver Options					
Pressure-based coupled solver	•	•	•	•	•

ANSYS 13.0 Capabilities Chart	ANSYS® Multiphysics™	ANSYS® CFD™			
		ANSYS® FLUENT®	ANSYS® CFX®	ANSYS® CFD-Flo™	ANSYS® POLYFLOW™
Pressure-based segregated solver		•			
Parallel solving on local PC option	•	•	•	•	•
Parallel solving over network option	•	•	•	•	
Customizable, scripting and user functions	•	•	•	•	•
Fluid-structural	•	•	•	•	
Fluid-thermal	•	•	•	•	
Electromagnetic-fluid	•	•	•	•	
Electrostatic-structural	•	•	•	•	
Magnetic-structural	•				
Electromagnetic-thermal	•				
Piezoelectric	•				
Piezoresistive	•				
Thermal-electric	•				
Thermal-structural	•				
Thermal-electric-structural	•				
Thermal-electric-fluids		•	+	+	
Electromagnetic-thermal-structural	•				
Electromagnetic-thermal-fluids		•	+	+	
Reduced order modeling (ROM)	•				
Multi-field general purpose solver	•	•	•	•	
Preprocessing					
Modeling Capabilities					
IGES/STEP geometry reader	•	•	•	•	•
Beam modeling	•				
Meshing Capabilities					
Defeaturing	•	•	•	•	•
Automatic surface meshing	•	•	•	•	•
Automatic tetrahedral meshing	•	•	•	•	•
Automatic prism inflation layers	•	•	•	•	•
Automatic hexa-core meshing	•	•	•	•	
Automatic swept hex meshing	•	•	•	•	•
Automatic hexa-dominant meshing	•	•	•	•	•
Thin-sweep meshing	•	•	•	•	•
Curvilinear elements					
Adaptive mesh refinement		•			
Boundary Conditions					
Solid model loads and boundary conditions	•	•	•	•	•
Tabular loads and boundary conditions	•	•	•	•	•
Function loads and boundary conditions	•	•	•	•	•
Apply temperature loads	•	•	•	•	•
Post-processing					
Report generator	•	•	•	•	•
Contour displays	•	•	•	•	•
Vector displays	•	•	•	•	•
Isosurface displays	•	•	•	•	•
Slicing planes	•	•	•	•	•
Quantitative calculations	•	•	•	•	•
Particle tracing	•	•	•	•	•
Animation	•	•	•	•	•
Output (images, Excel® data)	•	•	•	•	•
CFD turbomachinery post-processing		•	•	•	
General					
ANSYS® EKM™ data management ready	+	+	+	+	+
Parallel solvers (HPC licenses required)	+	+	+	+	+
Solver scripting language	•	•	•	•	•
Parameter manager	•	•	•	•	•

Figure 3: ANSYS-CFD capability table (excerpt)

detailed information about rack-by-rack inlet and exit temperatures, 3D thermal mapping of the data centre and racks highlighting trouble areas, 2D and 3D color contour plots of temperature, pressure, airflow and flow distribution through tiles and grills, 3D color animations of temperature distribution and airflow patterns, and others. CoolSim is available as a service subscription and pricing starts somewhere “under $10,000” [6].

2.2.1 **Possible Suitability for the SVD**

ANSYS solutions are well known, powerful and trusted products worldwide. ANSYS CFD turns out to be one of the most used tools for the analysis and optimisation of product development in the design phase. The scalability makes it a fast supporting tool besides the reliability. Nevertheless it is a commercial product, available only on licenses acquisition. CoolSim offers the assessment of the data centre as a service. It offers a HTML interface to the simulation results. During the time of this survey, no closer investigation on the usefulness and extensibility in respect to external components has yet been done, thus at this time it is unknown if CoolSim could be used to provide the simulation part for the SVD Toolkit.

2.3 COMSOL Multiphysics

COMSOL Multiphysics [7] is a finite element analysis solver based software suitable for the simulation of various physics phenomena, especially coupled-field problems (multiphysics).

It features a number of physics application modes that consist of predefined templates and user interfaces offering equations and variables related to several areas of physics: heat transfer, fluid mechanics, electromagnetism, structural mechanics, and so on. In case the predefined equations do not match the problem nature, users can set up custom models in general mathematical terms by entering specific partial differential equations.

COMSOL Multiphysics provides a number of predefined multiphysics couplings, but some coupled-field problems may require the addition of specific modules.

Some examples of modules are:

- The Computational Fluid Dynamics (CFD): Compressible and incompressible flows, turbulence can be modelled under forced and natural convection. Multiphysics-flow simulations enable the coupling of heat transfer issues with several types of flow.
- Heat Transfer Module: Provides a combination of capabilities to model heat transfer via conduction, convection, and radiation, as well as the ability to couple these to other physics, for instance in industrial applications such as electronics cooling and process engineering.

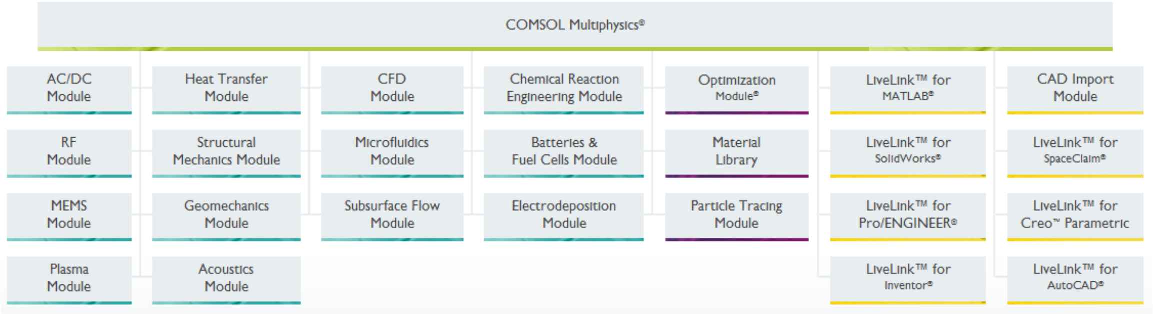


Figure 4: COMSOL modules

The predefined multiphysics couplings are quick entry points for common multiphysics applications, but it is also possible to create the same couplings starting from one application mode and then adding more modes, even later in the modeling. Some examples of predefined coupling are listed in the table below. As mentioned above, some couplings need the availability of additional modules listed in Table 1.

CATEGORY/ FOLDER	NAME	SPACE DIMENSION	REQUIRED MODULES	OPTIONAL MODULES
Electro-Thermal Interaction	Joule Heating	2D, 2D Axi, 3D	None	AC/DC, Heat Transfer, MEMS
Fluid-Thermal Interaction	Fluid-Thermal Incompressible Flow	2D, 2D Axi, 3D	None	
Fluid-Thermal Interaction/Flow with Variable Density	Non-Isothermal Flow (laminar)	2D, 2D Axi, 3D	Chemical Engineering or Heat Transfer	
Fluid-Thermal Interaction/Flow with Variable Density	Turbulent Non-Isothermal Flow	2D, 2D Axi, 3D	Chemical Engineering or Heat Transfer	

Table 1: COMSOL modules for CFD/heat transfer

COMSOL Multiphysics features several visualisation modes: outputs can be presented as static or animated pictures, movies, or as data for further processing. A report generator also allows results to be presented as an html file, along with the model set-up. Model files can also be exported as M-files for further manipulation in MATLAB.

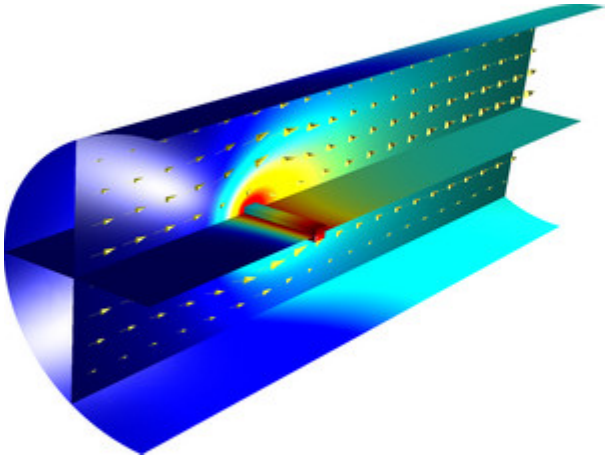


Figure 5: Visual representation of a Fully-coupled Physics with Joule Heating, CFD and Chemical Reactions (from www.comsol.com)

One more package that can be added to COMSOL Multiphysics is the Optimization Lab module. It enables the finding out of an optimum set of parameters – or *optimisation-variable* values – that minimises an *objective function* subject to a number of constraints. The Optimization Lab relies on a gradient-based optimisation algorithm to support for instance the determination of the optimal design of heat exchangers, ventilation channels and so on.

With the addition of a special module, the tool is also able to import geometries from CAD files (building spaces, HVAC components and so on), to facilitate and accelerate the development of the geometry of a physics problem in COMSOL.

COMSOL Multiphysics is a commercial tool. The price is about $7000. The addition of single modules or the optimisation package needs separate licenses, so the cost can increase.

2.3.1 **Possible Suitability for the SVD**

COMSOL Multiphysics features user friendly templates that facilitate the setting up of multiphysics problems like coupled heat transfer and fluid dynamics problems. Users are required to enter the geometry, boundary conditions and parameters. The geometry, like a data centre with its IT equipment and auxiliaries, can be imported from CAD software, speeding up the modelling process. Nevertheless, simulations can be slow and complex as long as the geometry gets more detailed and larger (larger and denser mesh) for instance in case of a large data centre with multiple racks and room integrated cooling components. The visualisation capacity is powerful, differencing between output variables and enabling animated pictures as well. As it is a commercial product, each module requires a paid license that may include maintenance and support, resulting in high costs. Most typically developed models can be found in the library as examples but they are pretty specific.

2.4 STAR-CCM+

Star-CCM+ is a CFD suite developed by CD-adapco [8]. It is a complete rewrite of the former CD-adapco CFD model Star-CD. Star-CCM+ is from ground up designed to simultaneously solve fluid flow and heat transfer problems. As Star-CCM+ is a complete rewrite, not all features supported by the legacy Star-CD are covered by it. Features not included are e.g. changing grids, two-phase flows, or wall condensation. Nevertheless, the features already implemented cover the requirements for CoolEmAll nicely. As it is the main development of CD-adapco and StarCD is maintained for backwards compatibility only, Star-CCM+ should be the preferred codebase.

Products competing in this market are often created from merging two separate solvers, reducing overall effectiveness and accuracy of the code. Star-CCM+ does not suffer from that kind of problem as it is designed to be a coupled fluid/thermal simulation code from the start. Star-CCM+ offers a client-server model that allows preparing a simulation on a local lightweight workstation and committing it directly to a high performance cluster for the actual simulation.

It is possible to couple Star-CCM+ with other running software codes [9]. The easiest way is through file exchange with the other running code. Through the use of user subroutines, it is also possible to couple Star-CCM+ with other external routines that can be used e.g. for optimisation or post-processing and visualisation.

Star-CCM+ is a commercial code that can be quite costly to acquire. Full licenses are about €25,000.

2.4.1 Possible Suitability for the SVD

Star-CCM+ is a high quality, high performance fluid simulation code that would be very suitable for the CoolEmAll SVD. It is very focused on thermal/fluid coupling, offering a very good performance and accuracy in this area. The high price tag makes it difficult to acquire, though, especially for the casual end user of the CoolEmAll SVD toolkit or for development partners who currently do not have a running Star-CCM+ license.

2.5 APC Data Centre Software

APC [10] is a proprietary suite solution developed by Schneider Electric that allows companies to design and optimise all the aspects of their data centres as well as help the administrators to manage the data centre and find possible failures once it is up and running. As for addressing this target, the APC suite focuses in 3 main fields: planning, availability, manageability.

For CoolEmAll the most interesting part of the suite is the planning one that covers airflow simulations and calculating possible carbon and energy footprints. APC [11] offers a simulation tool where a data centre owner can set a room, start placing the racks and the cooling (air or water cooling) and run airflow simulations within the 3D model room generated. A plugin is providing a library with different blade models that can be inserted into the racks. It is also possible to change rack positions within the data centre model and to run the simulation to see how the airflow evolves depending on the position. This can be also used when new equipment is going to be installed in our DC and the owner does not want to change the cooling setup or affect the servers that are already running customer services.

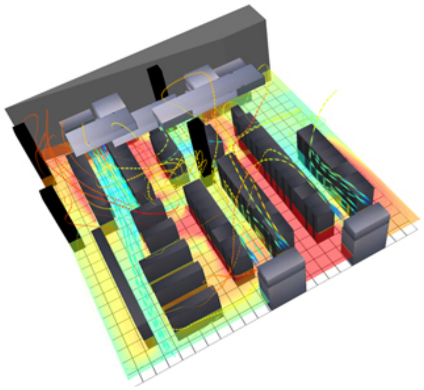


Figure 6: Visualisation of an APC data centre simulation

Another strong feature of the APC suite is the continuous monitoring [12] capability that allows the administrator to keep monitoring the whole server room on the electricity consumption. If any server is over a certain level of electricity consumption or the consumption is zero, or the heating of a certain rack or blade respectively, an alarm is raised. This feature is quite interesting as it might lead to a new simulation and reorganisation of a set of servers in the server room.

2.5.1 **Possible Suitability for the SVD**

The APC data centre software is a big suite for data centres. It is modular in nature, working as a plugin like system. This makes the system flexible, but the data centre owner has to buy licenses for each of the components that he wants to monitor or to implement these himself, such as the cooling system described above, the power manager system or the overall monitoring capability. All this makes the APC suite a very expensive tool for many of the data centre owners. As CoolEmAll is targeted to have a low entry price and will provide similar scalable functionality delivered as an Open Source system, reducing the cost of the design and monitoring, APC is most likely unsuitable for integration into the SVD toolkit.

2.6 6SigmaDC

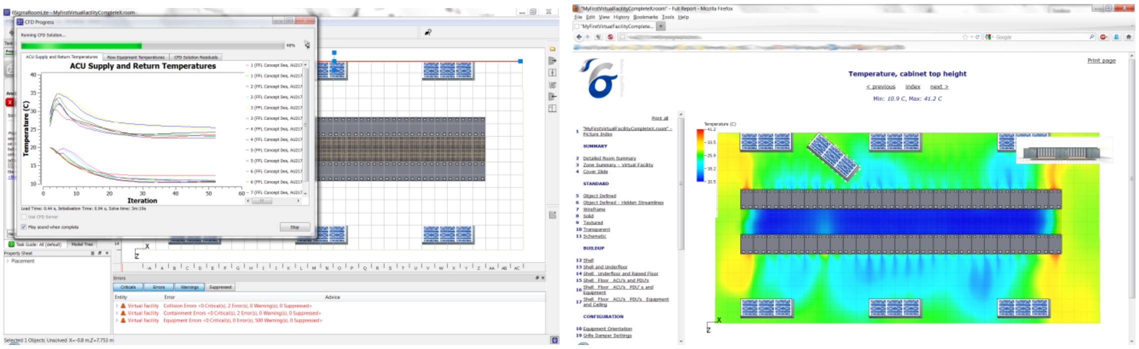


Figure 7: 6SigmaRoomLite simulation and results report

6SigmaDC [13] is a commercial simulation code developed by the company “future facilities” that is dedicated to the simulation of data centres. It combines a graphical user interface with a powerful simulation code that is able to simulate computing equipment on a rack (*6SigmaRack*) or room (*6SigmaRoom*) level. The software is targeted at data centre operators, data centre designers, facilities managers, field designers and equipment designers. 6SigmaDC allows building and testing a virtual test bed for a full data centre facility. It can be used to model the facility by assessing the optimal architectural form and cooling layout. It also helps optimising the deployment of IT equipment to the room in respect of cooling/energy efficiency, power configurations, cabling, and space utilisation. Racks and rooms can be arranged by drag and drop. The components can be retrieved from an accompanying library or by providing the information using a neutral data format. Additional modules are available that allow to design the power configuration for a data centre (*6SigmaPower*) and the stress put to the floor by rack weight (*6SigmaWeight*). 6SigmaDC appears to be the most comprehensive integrated package for data centre assessment available. It is not as fast to start an assessment as e.g. TileFlow (cf. Section 0), but is much more flexible than other tools. Racks can be placed arbitrarily in the data centre, thus enabling a very flexible layout. While the modelling tool itself is sometimes difficult to handle, the possibility to import the data centre layout from a standard CAD application alleviates for that. The solver is reasonably fast, although not as fast as e.g. the TileFlow solver. It is possible though to connect to a remote high performance simulation server. The server is able to run in parallel using distributed memory, possibly speeding up the computation massively through the use of a computing cluster. The CFD server is limited to Windows, though. Most standard high performance computing resources are Linux/UNIX based. Thus, most HPC centres have to provide a dedicated cluster for running the solver.

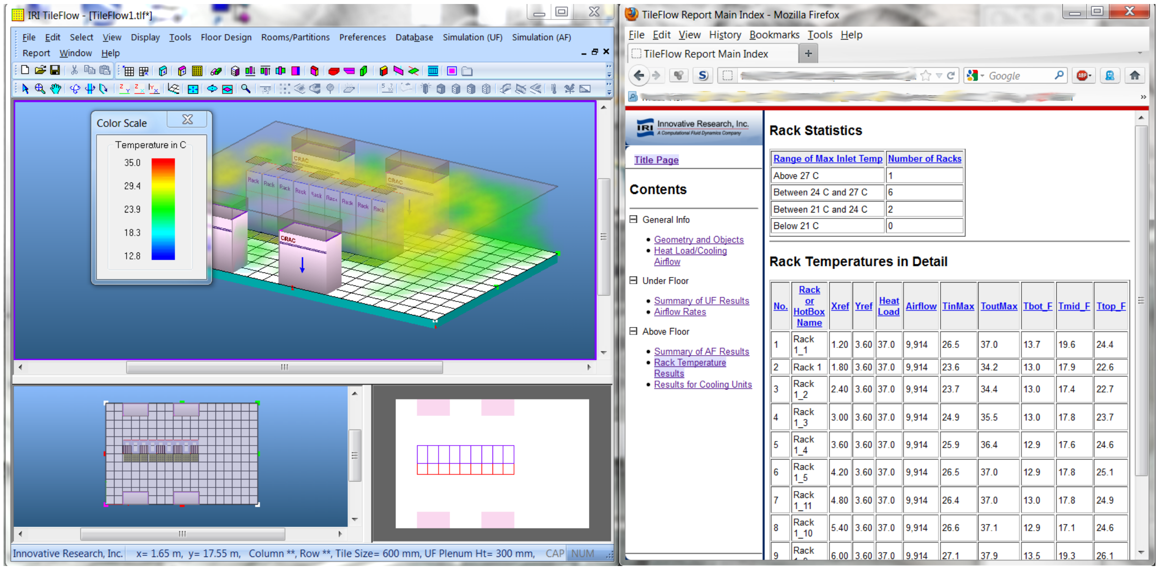
6SigmaDC can also import live power data and other monitoring information such as measured temperatures which can also be updated live using Exchange sets[1]. It is also possible to export. 6SigmaRoom have various import functions such as CSV and some inventory management tools. These are done through a defined API/file format. These are designed for populating a particular room not building a whole model from scratch. Apparently, there are some ongoing activities to export the data results to external, although details are not known.

While the modelling and analysis part of 6SigmaRoom is about €7000, the parallel solver starts at €35000 for a 4 core license. For every other 4 cores, there is an additional €3000 increase for the solver.

2.6.1 Possible Suitability for the SVD

From all the dedicated solutions surveyed, 6SigmaDC looks the most comprehensive and feature complete. It has a MPI-enabled solver that could scale well on many machines, delivering fast and in-situ results for the simulation part. From personal communication it also appears possible to do some live data exchange with 6SigmaRoom, although the details are not yet clarified. It is recommended, if the price tag of the solver is not a criterion for exclusion, to further investigate the possibilities of the 6SigmaDC suite.

2.7 TileFlow


IRI TileFlow - [TileFlow1.tlf*]
File Edit Select View Display Tools Floor Design Rooms/Partitions Preferences Database Simulation (UF) Simulation (AF)
Report Window Help
Color Scale
Temperature in C
35.0
29.4
23.9
18.3
12.8
CRAC
Rack
Innovative Research, Inc. x= 1.65 m, y= 17.55 m, Column **, Row **, Tile Size= 600 mm, UF Plenum Ht= 300 mm, CAP NUM
TileFlow Report Main Index - Mozilla Firefox
File Edit View History Bookmarks Tools Help
TileFlow Report Main Index
Google
Innovative Research, Inc.
A Computational Fluid Dynamics Company
Title Page
Contents
General Info
Geometry and Objects
Heat Load/Cooling Airflow
Under Floor
Summary of UF Results
Airflow Rates
Above Floor
Summary of AF Results
Rack Temperature Results
Results for Cooling Units

Rack Statistics

Range of Max Inlet Temp	Number of Racks
Above 27 C	1
Between 24 C and 27 C	6
Between 21 C and 24 C	2
Below 21 C	0

Rack Temperatures in Detail

No.	Rack or HotBox Name	Xref	Yref	Heat Load	Airflow	TinMax	ToutMax	Tbot_F	Tmid_F	Ttop_F
1	Rack 1_1	1.20	3.60	37.0	9,914	26.5	37.0	13.7	19.6	24.4
2	Rack 1	1.80	3.60	37.0	9,914	23.6	34.2	13.0	17.9	22.6
3	Rack 1_2	2.40	3.60	37.0	9,914	23.7	34.4	13.0	17.4	22.7
4	Rack 1_3	3.00	3.60	37.0	9,914	24.9	35.5	13.0	17.8	23.7
5	Rack 1_4	3.60	3.60	37.0	9,914	25.9	36.4	12.9	17.6	24.6
6	Rack 1_5	4.20	3.60	37.0	9,914	26.5	37.0	12.9	17.8	25.1
7	Rack 1_11	4.80	3.60	37.0	9,914	26.4	37.0	13.0	17.8	24.9
8	Rack 1_10	5.40	3.60	37.0	9,914	26.6	37.1	12.9	17.1	24.6
9	Rack	6.00	3.60	37.0	9,914	27.1	37.9	13.5	19.3	26.1

Figure 8: TileFlow modelling environment and reporting

TileFlow [14] [15] is a software tool for simulating the cooling performance of data centres. As the other integrated software packages, TileFlow constructs a computer model of the data centre and uses CFD to assess airflow patterns and

[1] Personal communication with future facilities representatives

temperature distribution. TileFlow also comes with an extensive data base of air-conditioning, perforated tiles and server racks for easy configuration of the computing room. Supplemental rack data can also be provided by a simple text file created by an asset management program, such as Aperture Vista, and imported into the modelling tool. It is also possible to import rack data from Excel files. TileFlow also integrates under-floor obstructions like pipes or cable into the calculation. TileFlow offers a very intuitive and fast user interface for arranging Racks, CRACs and other objects in the room. As with 6SigmaRoom, it is not possible to place racks arbitrarily in the room as objects have to be places axis-aligned and can only be rotated in steps of 90 degrees. It is also not possible to include a water cooling solution in the simulation workflow; it is only possible to simulate the airflow. The solver of TileFlow is parallelised using OpenMP. This allows TileFlow to scale on shared memory machines. This limits the calculation speed to one machine, making it difficult to generate in-situ simulations needed for quick data centre monitoring. Though intuitive to use, the user interface is the only way to set up a simulation.

2.7.1 **Possible Suitability for the SVD**

Although TileFlow is intuitive to use and has a great user interface it shows some limits that might impact its suitability for CoolEmAll. First the absence of an external API makes it difficult to attach it to other frameworks. It is impossible to set workloads and cooling volumes without resorting to the user interface. The strictly rectangular arrangement possible with TileFlow may be covering all current practical cases and reduces simulation performance requirements, but on the other hand makes it impossible to find novel solutions that are off the road from current practice. The OpenMP parallelisation of the TileFlow solver also requires a high performance shared memory machine to achieve immediate processing times, making it difficult to deliver in-situ simulation results during data centre monitoring.

3 Workload Simulation

The growing importance of energy efficiency in information technologies led to significant interest in energy saving methods for computing systems. Therefore, intelligent resource management policies are gaining popularity when considering the energy efficiency of IT infrastructures. Nevertheless, studies of impact of scheduling strategies on energy consumption require a large effort and are difficult to perform in real environments, even for modular IT infrastructures and facilities. To overcome these issues extensive research has been conducted in the area of computer-based simulation tools. As a result, a wide variety of simulation tools emerged. The following sections contain a classification of simulation features, a comparison of existing workload simulators and detailed description of tools that address the CoolEmAll issues in the most comprehensive way.

3.1 Simulation Features

A comprehensive taxonomy for design of simulation tools to model large and distributed systems has been presented in [16]. However, to compare various simulation tools a more detailed classification of simulation features is needed. In this section a classification including a variety of aspects that should be taken into consideration in a simulation of modern computing systems is presented. Most of these properties are essential to correctly model and simulate data centre applications and workloads as well as energy-efficient scheduling policies introduced in the CoolEmAll project.

A summary of the most significant workload simulator features is illustrated in Figure 9 and explained below. More details can be found in [17].

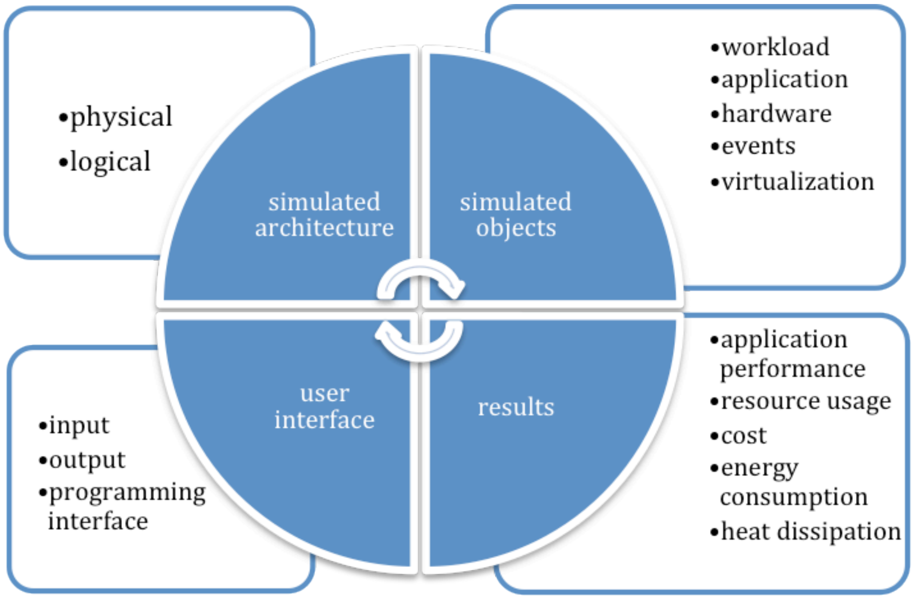


Figure 9: Essential features of workload simulators

Simulated architecture. Simulation tools may differ in terms of the architectures of the modelled systems as well as their level of details. The *logical architecture* of distributed systems may assume centralised control of resources, hierarchical topology of schedulers or fully distributed models. In addition, the *physical topology* may span from computing nodes through racks and clusters up to whole data centres or geographically distributed grids and clouds. Granularity of such topologies may also differ from coarse-grained to very fine-grained modelling single cores, memory hierarchies and other hardware details.

Simulated objects. The simulation environment ought to enable, apart from simulation of resource components, modelling the behaviour of other computing entities. One of the most important elements of simulation is workload which may be taken from real systems or generated synthetically. *Workload* may contain jobs ranging from single sequential jobs, through parallel and distributed jobs, up to the whole workflows containing time dependencies between jobs. More detailed simulations of jobs may require to model *application* performance by taking into account a lot of factors that affect application execution, e.g., processing unit architecture, application characteristics, and input data. Besides complex information about the simulated *hardware* (concerning architecture, characteristics, state, energy profile), a simulation tool may allow performing specific actions on resources, for instance change their states or support advance reservation (AR). It is also important to model the dynamic nature of distributed environments by handling different *resource events*. Modern simulation tools may also support aspects that gained special attention recently, in particular *virtualisation* and energy-aware techniques.

Results of the simulation. The goal of each simulation run is to provide a set of results that allows evaluating a specific computing system or scheduling method. These results may include: *application performance metrics* (e.g., execution time, flow time, waiting time, completion time), *resource usage*, *cost*, *energy consumption* and *thermal effects* caused by computations, etc.

User interface. The user interface determines how the user interacts with the simulation tool. The interface is essential to achieve high experiment execution automation and ease of use. First of all, the *programming interface* may differ between simulators that may be delivered along with programming libraries or more structured frameworks. Usually frameworks allow reducing the complexity of the simulation environment and to enter any modifications more conveniently. In addition to programming interface, simulators may be accompanied with intuitive *graphical user interface* that allows defining simulation parameters much faster and easier than using ordinary configuration files. These graphical interfaces may include user-friendly representation of simulation results to facilitate the analysis of the computing system.

3.2 Comparison of Simulation Tools

Table 1 presents the results of a comparative analysis of simulation tools based on the proposed classification of simulator features. Simulation tools were selected according to their possible suitability to the CoolEmAll project that was evaluated based on their publications, availability, and list of features. The content of the table is based on recent publications concerning the given tools. Moreover, all features were verified whenever possible by experimental studies and a code analysis of the available simulators.

GridSim + CloudSim [18], [19] developed at the University of Melbourne, is a toolkit that provides means for modelling and simulation of base components that constitute parallel and distributed computing environments (grid users, applications, resources, schedulers and resource brokers) and for the study of the involved scheduling algorithms. A flexible and extensible architecture allows to modify components behaviour or to incorporate new ones into the existing infrastructure. Thus, GridSim is commonly adopted by other simulation tools that benefit from its core functionality. CloudSim is similar to GridSim, it is however a separate tool that enables modelling and simulation as well as experimentation of Cloud computing infrastructures.

Alea [20] developed at Masaryk University in Brno, is a Grid and cluster scheduling simulator, designed for examination, testing and verification of various job scheduling strategies. Alea is based on the GridSim toolkit and extends the original basic functionality by introducing some innovative solutions like, e.g. ”on the fly” job reading that leads to a better scalability. The simulator is able to cope with general resource management problems like the heterogeneity of jobs, resources, and dynamic runtime changes. It also supports a specific workload format - MetaCentrum.

SimGrid [21] is a joint project between the University of Hawaii at Manoa, LIG Laboratory in Grenoble and the University of Nancy. It aims to provide core functionalities for the simulation of distributed applications in heterogeneous distributed environments. The main goal of the project is to facilitate research in the area of parallel and distributed large scale systems. In particular, SimGrid provides programming environments to support both researchers who study their algorithms and need to run simulations quickly as well as developers who can develop real distributed applications.

GroudSim [22] toolkit is developed at the University of Innsbruck. It allows simulating both Grid and Cloud computing. The main GroudSim features include: simulation of file transfers, the calculation of costs and background loading. The event module supports resource and network failures as well as recovery events affecting these entities. Simulations can be easily extended by any kind of probability distribution package.

FEATURE		SIMULATION TOOL	GSSIM	GridSim + CloudSim	Alea	SimGrid	GroudSim	GreenCloud
SIMULATED OBJECTS	WORKLOAD	INPUT	real (SWF, GWF) + synthetic (generator)	real (SWF, user-defined - primitive)	real (SWF, GWF, Metacentrum)	XML format (own)	real (GWF)	synthetic (generator)
		MODEL	sequential, parallel, distributed, rigid, mouldable, pre-emptive jobs, workflows	sequential, parallel, distributed, rigid, mouldable, pre-emptive jobs	sequential, parallel, rigid, mouldable jobs	sequential, parallel, distributed jobs, workflows	sequential jobs	sequential, rigid jobs
	APPLICATION		performance modelling	no	no	performance modelling	no	no
	HARDWARE		information (architecture, characteristics, state, energy profile) + control (energy management, state) + AR	information (architecture, characteristics, state) +AR	information (architecture, characteristics, state)	information (architecture, characteristics, state)	information (architecture, characteristics, state)	control (energy management)
	EVENTS		resource dynamics + resource (FTA) and network failures, recovery, + security	resource failures (FTA)	resource failures	resource dynamics + resource failures	resource and network failures, recovery	no
	VIRTUALISATION		yes	yes (CloudSim)	no	no	no	no
SIMULATED ARCHITECTURE	PHYSICAL		user-defined	grid/ cluster/ computing node/ CPU	grid/ cluster/ computing node/ CPU	cluster/ computing node/ CPU	grid/ cluster/ node	cluster/ computing node
	LOGICAL		centralised, decentralised, hierarchical	centralised, hierarchical	centralised, hierarchical	centralised, decentralised	centralised	centralised
USER INTERFACE	PROGRAMMING INTERFACE		framework (introduced layers, plugins)	generic library	framework (introduced layers, plugins)	generic library	framework	framework
	INPUT		comprehensive GUI – experiment editor	configuration files	configuration files	configuration files generator	configuration files	configuration files
	OUTPUT		text statistics, interactive charts	text statistics, charts	text statistics, charts	text statistics, charts	text statistics, charts	text statistics
RESULTS			application performance, cost, resource usage, energy consumption	application performance, cost, resource usage, energy consumption (CloudSim)	application performance, resource usage	application performance, cost, resource usage	application performance, cost, resource usage	resource usage, energy consumption

Table 2: Comparison of simulation tools

GSSIM [23] is a workload execution simulation framework developed by PSNC. It provides an automated framework for experimental studies of various resource management and scheduling policies in distributed computing systems. GSSIM achieves this through a flexible design of architecture and interactions between scheduling components, a possibility of plugging scheduling algorithms into the simulated environment, modelling synthetic workloads and adopting real traces in popular formats, an extensible configuration of the computing infrastructure topology both on logical and physical level, and many other features.

GreenCloud [24] is an extension to the network simulator Ns2 [25] that is developed for the study of cloud computing environments. The GreenCloud offers users a detailed, fine-grained modelling of the energy consumed by the elements of the data centre, such as servers, switches, and links. Moreover, GreenCloud offers a thorough investigation of workload distributions. Furthermore, a specific focus is devoted on the packet-level simulation of communications in the data centre infrastructure that provides a fine-grained control.

As shown in Table 1, most simulators basically focus on addressing specific problem areas. One should note that there is a lack of tools that allow performing specific actions on resources and in this way affecting their behaviour. Moreover, there is a very little support for application performance modelling and incorporation of virtualisation techniques.

For purposes of CoolEmAll, the most important properties of workload simulation tools should be:

- Possibility of including different workload types with various application profiles within the simulation
- Incorporating sophisticated workload management policies including virtualisation techniques and energy management
- Evaluation of thermal aspects in scheduling
- CoolEmAll partners' expertise with the tool
- Possible integration with external tools within the SVD Toolkit

For further analysis, three workload simulation tools with a functionality fitting CoolEmAll needs the most were chosen. Their detailed description is presented in the next section.

3.3 Workload Simulation Tools

3.3.1 **GreenCloud**

GreenCloud is a C++ based simulation environment for energy-aware data centres. It is primarily aimed at experimental studies of communication patterns and detailed evaluation of the energy consumption in currently deployed and future data centre architectures. GreenCloud is a packet level simulator, which means that whenever a data message has to be transmitted a packet structure needs to be allocated in the memory and processed. Although this approach improves the simulation accuracy, it increases the overall simulation time and leads to poor scalability.

GreenCloud focuses on detailed modelling of communication aspects of the data centre network. Being based on the Ns2 platform implementing the TCP/IP protocol reference mode in full, it allows capturing the dynamics of widely used communication protocols such as IP, TCP, UDP, etc. Whenever a message needs to be transmitted between two simulated elements, it is fragmented into a number of packets limited in size by the network MTU. Then, while routed in the data centre network, these packets may become a subject of different network events like link errors or congestion-related losses in network switches.

The execution of each workload object in GreenCloud requires a successful completion of its two main components: computing and communicational. The computing component defines the amount of computing that has to be executed before a given deadline on a time scale. The deadline aims at introducing Quality of Service constraints specified in a Service Level Agreement. The communicational component of the workload defines the amount and the size of data transfers that must be performed prior, during, and after the workload execution. To cover the vast majority of cloud computing applications, GreenCloud defines three types of workloads: computationally intensive workloads that load computing servers considerably, data-intensive workloads that require heavy data transfers, and finally balanced workloads which aim to model the applications having both computing and data transfer requirements.

GreenCloud describes application with a number of computational requirements. Moreover, it specifies communication requirements of the applications in terms of the amount of data to be transferred before and after a task completion. In addition time constraints can be taken into account during the simulation by adding a predefined execution deadline. Nevertheless, GreenCloud does not support application performance modelling. Aforementioned capabilities allow only incorporating simple requirements that need to be satisfied before and during task execution.

To deliver information about the energy usage, GreenCloud distinguishes three energy consumption components: computing energy, communicational energy, and the energy component related to the physical infrastructure of a data centre. This approach enables modelling energy usage associated with computations, network operations and cooling systems. In GreenCloud, the energy models are implemented for every simulated data centre entity (computing servers, core and rack switches). Moreover, due to the advantage in the simulation resolution, energy models can operate at the packet level as well. This allows updating the levels of energy consumption whenever a new packet leaves or arrives from the link, or whenever a new task execution is started or completed at the server.

The server power consumption model implemented in GreenCloud is dependent on the server state and its CPU utilisation and allows capturing the effects of both of the Dynamic Voltage and Frequency Scaling (DVFS) and Dynamic Power Management (DPM) schemes.

At the links and switches level, GreenCloud supports Dynamic Voltage Scaling (DVS) and Dynamic Network Shutdown (DNS) techniques. The DVS method introduced a control element at each port of the switch that – depending on the traffic pattern and current levels of link utilisation – could downgrade the transmission rate. DNS approach allows putting some network equipment into sleep mode.

To perform an analysis using GreenCloud simulator, the researcher needs to prepare text files with a suitable environment configuration. As a result, GreenCloud produces a set of trace files that enable the evaluation of the generated statistics and the simulation process. However, no GUI tool is available to configure a simulation setup or display simulation graphs in a user-friendly way.

GreenCloud simulator is released under the GPL Open Source license.

3.3.1.1 Possible Suitability for the SVD

GreenCloud is a workload simulator that focuses mainly on the evaluation of various energy aspects in modern data centre architectures. It offers the possibility for estimation of the energy consumption and an opportunity for affecting the energy efficiency by enabling energy-aware scheduling. Contrary to what the GreenCloud name may suggest, it does not allow testing the impact of a virtualisation-based approach on resource management. There is also a lack of support for application performance modelling. As the source code is available and modifiable, it can be easily integrated with an external simulation and the visualisation tools within the SVD Toolkit.

3.3.2 GridSim and CloudSim

GridSim and CloudSim are two separate, event-based simulation tools written in Java. Initially CloudSim was based on the GridSim framework, however since the latest release it is an independent tool. Hence, it does not benefit from most of the GridSim functionality. Nevertheless, both simulators have some common features, including the general architectural concept and simulation approach.

Preparing a workload using these tools usually consists in manual creation of dedicated objects that represent tasks to be scheduled. As a convenience, GridSim is able to partially support SWF [26] files and read data in a user-defined file format. Despite that fact, both simulators can handle a wide variety of workload types, including parallel, distributed and pre-emptive jobs.

Similar to GreenCloud, both GridSim and CloudSim define a simple application model that includes computational and data requirements. Although all these constraints are taken into account during scheduling, they do not affect the application execution. Thereby, a researcher is required to put a lot of effort to incorporate an application performance model into his experiments.

GridSim allows creating a simple resource hierarchy containing computing resources that consist of machines and processors. Additionally, it may simulate the behaviour of other components including storage and network resources.

CloudSim focuses on computational resources and provides an extra virtualisation layer that acts as an execution, management, and hosting environment for application services. It is responsible for the VM provisioning process as well as managing the VM life cycle such as: VM creation, VM destruction, and VM migration. It also enables evaluation of different economic policies by modelling the cost metrics related to the SaaS and IaaS models.

The CloudSim framework provides basic models and entities to validate and evaluate energy-conscious provisioning of techniques and algorithms. Each Processing Element entity is extended with an additional Power Model object for managing power consumption. To support modelling and simulation of different power consumption models CloudSim provides some basic implementation of this component. Needless to say, it can be extended for simulating custom user power consumption models of a Processing Element. That allows estimating the current power usage according to the current utilisation metric for the Cloud host. The return parameter is the current power consumption value. This capability enables the creation of energy-conscious provisioning policies that require real-time knowledge of power consumption by Cloud system components. Furthermore, it allows an accounting of the total energy consumed by the system during the simulation period. However, the proposed solution is not appropriate for more sophisticated power management policies. In particular, CloudSim is not sufficient for modeling Dynamic Voltage and Frequency Scaling techniques and managing resources power states.

GridSim and CloudSim are not accompanied by any GUI that would facilitate the configuration of the simulation settings. The user is required to write own methods for establishing the simulation environment and to provide text configuration files. Similarly, although the basic statistics are gathered, the user has to extract them and take care of their presentation. In the case of the CloudSim simulator, he may be assisted by an external CloudAnalyst tool that visualises only high-level simulation setup parameters targeting cloud applications.

GridSim and CloudSim offer several features that may provide a basis for comprehensive simulations. However, due to significant differences in libraries used and API's, they should be considered separately before their application in experimental studies. Besides, possible integration may be a challenging and time-consuming process.

Both simulation tools are available as Open Source under GPL license.

3.3.2.1 Possible Suitability for the SVD

GridSim and CloudSim allow the simulation of Grid and Cloud computing systems, respectively. In contrast to GreenCloud, CloudSim provides only basic means to estimate the energy consumption, while GridSim does not support this functionality at all. In regards of modeling virtualized resource, CloudSim offers complex and extendable capabilities for Cloud computing simulations. On the other hand, none of the tools allows taking into account the application profiles during scheduling. Both tools are commonly adopted and enhanced by other distributed computing simulators and are available without any cost issues. Thus, any modifications for the purposes of the SVD Toolkit would be possible. However, it should be emphasized that adopting GridSim and CloudSim to collaborate with each other may require a lot of effort.

3.3.3 GSSIM

GSSIM is a discrete event, process oriented simulator based on the SimJava framework. Due to the modular and flexible framework architecture, a wide scope of modules that provide capabilities mentioned in Table 2 can be incorporated into the GSSIM environment.

Experiments performed in GSSIM require a description of jobs and tasks which will be scheduled during the simulation. As a basic description, GSSIM uses files in the Standard Workload Format (SWF) [26] or its extension Grid Workload Format (GWF) [27]. In addition to the SWF file, some more detailed description of a job and task can be provided in an additional XML file. This form of description provides the scheduler with more detailed information about task requirements, user preferences and execution time constraints, which are unavailable in SWF/GWF files. Further, the simulator is complemented with an advanced workload generator tool that allows creating synthetic workloads. Moreover, GSSIM supports diverse structures of workloads. In particular, it can simulate scheduling of multiple independent jobs at once, various kinds of parallel jobs, and whole workflows. Moreover, GSSIM is able to handle rigid and mouldable jobs, as well as pre-emptive jobs.

GSSIM allows simulating a wide scope of physical and logical architectural patterns. In particular, GSSIM provides means of simulating complex distributed architectures containing models of the whole data centres, containers, racks, nodes, etc. However, new resources and distributed computing entities can be easily added to the GSSIM environment in order to enhance the functionality of the tool and address the most sophisticated requirements.

GSSIM also provides means to include complex and specific application performance models during simulations. These models can be plugged into the simulation environment through a dedicated API. Implementation of this plugin allows researchers to introduce specific ways of calculating task execution time. The number of parameters including task requirements, task and input data size, detailed description of allocated resources as well as network parameters, can be applied to specify the execution time of a task. Using these parameters developers can for instance take into account the architectures of the underlying systems, such as multi-core processors, or virtualisation overheads, and their impact on final performance of applications.

Apart from complex modelling of the expected application performance, GSSIM allows researchers to take into account energy consumption issues in distributed computing simulations [28]. To simulate energy consumption appropriate models must be used. The main goal of these models is to emulate the behaviour of the real computing resource and the way it consumes energy. Due to a rich functionality and flexible environment description, GSSIM can be used to develop new energy consumption models or to examine energy management strategies. In more detail, GSSIM provides a functionality to define the energy efficiency of resources, dependency of energy consumption on resource load and specific applications, and to manage power modes of resources. The energy consumption models provided by default in GSSIM take into account the definition of resource power usage, resource load and application profiles, and metrics including their energy consumption and heat production.

Moreover, GSSIM provides interfaces, which allow scheduling plugins to collect detailed information about computing resource components and to change their energy states. Presence of detailed resource usage information, current resource energy state description and a functional energy management interface enables an implementation of energy-aware scheduling algorithms. Resource energy consumption becomes in this context an additional criterion in the scheduling process, which uses various techniques to decrease energy consumption, e.g. workload consolidation, turning off unused resources, cutting down CPU frequency and others.

Another essential feature is targeted to model and evaluate emerging cloud computing systems. In the scope of the CoolEmAll project it is important to study consolidation of load in virtualised environments. GSSIM provides interfaces that allow incorporating virtualisation techniques into the resource management simulation. In particular, GSSIM provides means for modelling Cloud computing infrastructures including creation and management of virtual machines. It supports defining policies for provisioning resources to virtual machines and allocation of virtual machines to specific applications. The main goal of the virtualisation extension is to facilitate the researcher in finding trade-offs between performance, energy usage and costs of a given set of applications. Thereby, a user is able to incorporate sophisticated load balancing strategies, including dynamic workload migration, that do not affect the application execution.

PSNC has performed many studies of various resource management policies using the GSSIM simulation environment. GSSIM has been successfully applied in a substantial number of research projects and academic studies. For instance, GSSIM allows Grid computing researchers to study the performance of many scheduling algorithms in complex distributed computing infrastructures. They may vary from simple job scheduling policies, through multi-criteria resource management [29], up to complex scheduling problems with QoS requirements as presented in [30]. As a network simulator, GSSIM has been used to study the problem of resource allocation with network resources for workflow applications that is presented in [31]. It has also been adopted as a simulation framework in European project called Federica [32] in order to test different approaches concerning resource allocation in virtual network architectures.

The GSSIM framework is complemented by the portal [33] that enables online access to the simulator via a user-friendly experiment editor, workload generator and experiments repository. An example of the experiment editor window showing a resource topology is presented in Figure 10. The rich web interface allows executing the simulation experiments remotely, provides access to workloads, resource descriptions and implementations of algorithms, and enables interactive visualisation of the results. Statistics are also delivered in a plain and easy to process CSV text format. In this way, GSSIM provides a comprehensive environment enabling researchers to test resource management algorithms and architectures, and to exchange not only workloads but also results of experiments and implementations of algorithms.

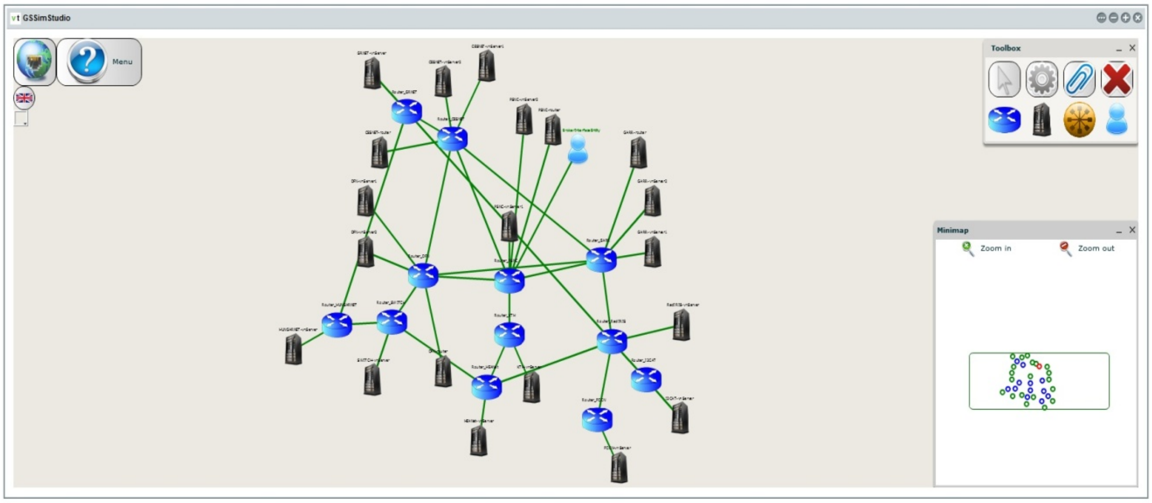


Figure 10: GSSIM GUI

GSSIM is released as Open Source under a GPL license and *is freely available for anyone to download.* Thus there are no licensing and cost issues, besides the fact that GSSIM cannot be directly linked against proprietary, non GPL-code.

3.3.3.1 Possible Suitability for the SVD

GSSIM is a highly customisable and extendable framework that offers a comprehensive simulation environment. With the possibility of adding customised application performance models, GSSIM addresses the application profiling requirements introduced by CoolEmAll and enables a classification in relation to their energy consumption. Complementary, GSSIM's energy module is suitable for achieving most of the SVD Toolkit objectives. GSSIM also provides means to include various types of workload and evaluate different workload management policies both in HPC and Cloud environments. These features, combined with free Open Source code and PSNC's experience, make it a suitable simulator in terms of the SVD objectives. Furthermore, due to a standard format for the input files and the simulation results, GSSIM can be easily integrated with other SVD

Toolkit components.

4 Visualisation

4.1 ParaView

ParaView [34] [35] is an open source data analysis and visualisation platform. It runs on all major operating systems. It is mainly developed and supported by Kitware and Los Alamos National Laboratory. Development started in 2000 with a first release 2002, and meanwhile ParaView offers a complete set of functionality for all major visualisation tasks. It is based on the VTK library [36] that is developed by KitWare. The underlying VTK library allows using a vast amount of visualisation algorithms for post-processing. ParaView is data flow centred, i.e. all operations produce standard data sets that can be processed by other operations that can be linked together to a post-processing workflow. Operations are realised as filter modules that are linked together to form a pipeline.

ParaView operates on structured, unstructured, polygonal and AMR (Adaptive Mesh Refinement) data. ParaView supports all standard post-processing algorithms like cutting- and isosurfaces, vector glyphs and streamlines / particle traces. It is also possible to quickly develop own filters either by extending existing open source modules or by taking the Python Programmable filter [37] allowing to implement own functionalities in Python [38]. Using the Python Programmable filter it is easily possible to create and manipulate data sets with just a few lines of Python code. The scripting itself depends on the vtkProgrammableFilter class that has to be enabled during VTK compile time. While it is possible to use Python scripts as modules, it is also possible to steer ParaView completely from Python. The whole data model is accessible from Python. Changes done with Python are also immediately reflected in the user interface.

Using ParaView, it is possible to visualise data sets on shared and distributed systems. The parallelism of ParaView is implemented on top of MPI. This allows ParaView to run on all common supercomputers, as MPI is the standard communication platform for most parallel high performance codes. Of course this also makes it difficult to run an ad hoc distributed session with ParaView, as MPI has to be installed and configured on all target hosts. When running distributed, it is possible to use ParaView in a client/server mode. The user interface is running on a desktop frontend steering the distributed instance. Data is broken down into chunks suitable for parallel processing including ghost cells needed for algorithms that require some information on neighbouring cells. Most post-processing filters are used unmodified when running in parallel and transparently for the user who does not have to take care of the distribution of his objects. The rendering itself happens either locally at the front end workstation by collecting all geometry data and rendering it locally. For data that is too large to be rendered locally, ParaView is also able to render it at the processing node and sending the image to the front end node. There the incoming images are merged and a complete rendering is created. It is also possible to use a hybrid method that renders coarse levels of detail locally and the full model distributed on the post-processing nodes.

4.1.1 Simulation Integration

ParaView contains readers for several input file formats. Readers include modules for simulation codes like FLUENT, LSDyna, OpenFOAM, and others. ParaView can also read some geometry files like VRML or PLY. All in all, ParaView offers readers for about 35 file formats.

ParaView does not provide simulation monitoring and steering capabilities out of the box. Some solutions were done by external parties. Those did never leave the prototype implementation phase and where included in the vanilla ParaView package for wide acceptance. Biddiscombe et al. [39] implemented a HDF5 DSM virtual file driver and a ParaView plugin to attach a running simulation to ParaView. Both, ParaView and the simulation, read and write to the same shared HDF5 file layer, thus exchanging information.

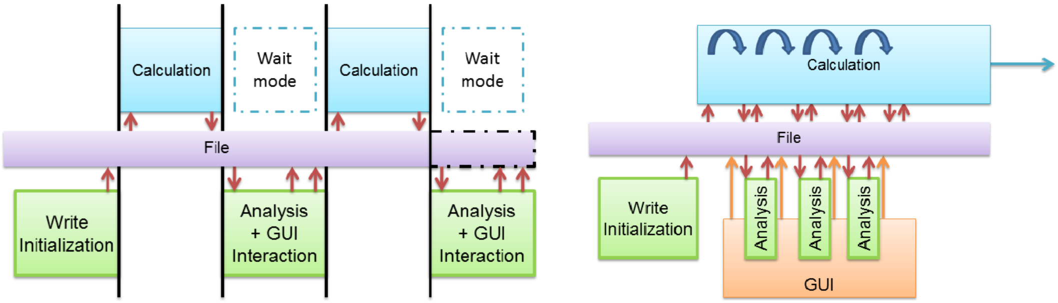


Figure 11: In-situ simulation with ParaView (from [39])

The data exchange can be done in two different modes as illustrated in Figure 11. *Wait* mode synchronously alternates data access between the simulation and the analysis tool. When the calculation starts, the file is locked for all accesses from the simulation steering side. When the calculation is finished, control is handed to the simulation steering and analysis module that can now freely operate on the data. The other mode is called *free* mode. Here, the control flow is not as strict as in wait mode. During the calculation the file is open for modifications from the steering side, although it is still not possible to write and read the file from both sides at the same time. Analysis and post-processing can be done distributed at the compute nodes themselves, saving a lot of bandwidth copying around data from the simulation, or on a dedicated rendering machine by collecting all data in the DSM and transparently reading it at the visualisation side. While the approach using HDF5 is quite standard compliant and straightforward, practical use of the HDF5 format remains uncommon. Almost none of the commercially important simulation codes use it [40]. This significantly limits the usability of this approach.

4.1.2 Possible Suitability for the SVD

ParaView is a sophisticated Open Source tool that is scalable on large clusters. It provides import modules for many simulation codes, free and commercial. Attaching to a running simulation is not possible out of the box. Although possible, as existing coupling approaches show, it is not straightforward to do in-situ analysis with running simulations. As ParaView is Open Source, modifications needed for inclusion into the SVD as a visualisation platform would be possible.

4.2 EnSight CFD

EnSight [41] is a visualisation package created by CEI Software. It is one of the standard visualisation tools commonly used in commercial settings. As such, many simulation codes are able to export to the native EnSight format. Additionally, EnSight reads and writes a large range of simulation code data formats [42]. Besides standard visualisation techniques like cutting planes, isosurfaces or streamlines, EnSight also offers extended visualisation methods. Using Volume Rendering it enables to explore volumetric data. It also contains some methods not found in many visualisation packages like lift/drag calculation, feature extraction, vortex core lines, separation, attachment and much more. It is also possible to create high quality animations and renderings of the visualisation results. For purposes of CoolEmAll, EnSight comes in two different flavours: EnSight CFD and EnSight Standard/Gold/DR. While EnSight Standard/Gold/DR is targeted at general visualisation of all kinds of CAE data, EnSight CFD is focused on visualisation for CFD and offers a limited subset of the complete EnSight platform. An advanced feature missing in EnSight CFD is e.g. the client-server infrastructure that allows running parts of the visualisation post-processing on a remote workstation to distribute the workload. Further features like parallel processing and rendering is limited to the high-end Gold or DR versions. These versions also feature a limited stereoscopic view of the data sets, but no real 3D interaction. The full list of features is available at [43]. EnSight is available on Windows, Linux and Mac, while the DR version sporting the most features is only available for Linux. EnSight offers a API that enables the user to write own tools and applications that use the core EnSight functionality.

EnSight CFD licenses start at $3200/a. The costs for EnSight start at $4000/a for the Standard version and end at $10000/a for the high-end EnSight DR [44].

4.2.1 Simulation Integration

EnSight is one of the best supported visualisation packages in respect to other independent software vendors. There are many simulation packages that allow a direct export of the simulation data to EnSight. For others, standalone converters exist. EnSight thus offers a large amount of built in and contributed readers for all practical data formats.

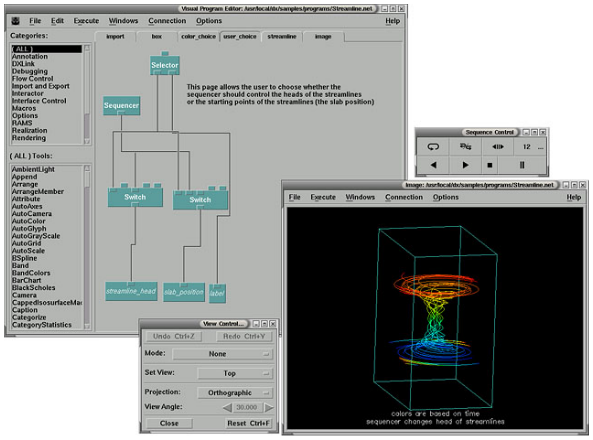
It is possible to do in-situ simulations with EnSight using the co-processing feature [45]. The co-processor is a standalone visualisation service that is able to read simulation output and forward it to the EnSight visualisation platform. This is possible if the simulation is able to write native EnSight files. A direct memory based coupling like it is available in COVISE (cf. Section 4.4) is not possible in EnSight.

4.2.2 **Possible Suitability for the SVD**

EnSight is a well established and well supported visualisation platform. It can do in-situ analysis of simulations, albeit only using a file based interface. That limits fast processing of simulation results to the disk I/O performance of the simulation platform. The high price tag also limits the suitability of EnSight for CoolEmAll, especially during the development phase.

4.3 OpenDX

OpenDX (Open Data Explorer) is a modular visualisation package originally created by IBM and released to the public using the open source IBM Public License (IPL). Thus, OpenDX is freely available in binary and source form and can be modified for the needs of a project if necessary. OpenDX is data flow oriented and is able to visualise data using all standard visualisation techniques like cutting surfaces, iso surfaces or streamlines. Data Explorer is a system of tools and user interfaces for visualising data. In general terms the visualisation of data can be considered a 3-stage process: Describing and importing data, processing the data through a visualisation program and presenting the resulting image [46]. OpenDX consists of several principal components:



This page allows the user to choose whether the sequencer should control the heads of the streamlines or the starting points of the streamlines (the slab position)

Mode: None

Set View: Top

Projection: Orthographic

View Angle: 30.000

colors are based on time
sequencer changes head of streamlines

Figure 12: OpenDX user interface

The **Data Model** is the set of definitions, rules, and conventions used to describe Data Explorer entities (including data fields, geometrical objects, and images). The **Data Prompter** is the user interface for describing data to be imported into Data Explorer. Using the **Data Browser** a data file can be explored, determining the layout and organisation of the data it contains, and transferring this information to the Data Prompter. A high-level **Scripting Language** for creating visualization programs; it can also be used directly in a command mode to perform various tasks; visual programs – i.e., the visualisation programs displayed in the Visual Program Editor window as networks of module icons – are also written in the scripting language; a visual program constructed in this window by the user is translated into the same language when it is saved to disk.

These visual programs are created as data flow networks by using the **Visual Program Editor (VPE)**; as noted above, programs created with this editor are translated into the scripting language by Data Explorer and are stored in that form. The main building blocks that constitute to the visualisation are **Modules**. Modules are distinct post-processing algorithms that can be chained to realise the final visualisation result. They can be directly accessed and manipulated in the Visual Program Editor. Customised Modules can be created by using the **Module Builder**; Modules created here can be used the same way as native Modules. The final output of the visualisation network is presented in the **Image Window**; it offers a rendering of the output as well as some interaction facilities to manipulate the object view. For changing the parameter values used by a visual program, one or more **Control Panels** can be used.

While OpenDX is widely used because of its capabilities and free availability, it can be seen as end-of-life software. The user interface is using the Motif library that is not supported on the Windows platform and no attempts have been made so far to change that. The latest release 4.4.4 dates from 2006 and it can be assumed that no further releases will be done.

4.3.1 Simulation Integration

OpenDX does not provide any importers for external data, but instead relies on the simulation or 3rd party converters to create its native DX format. This also holds for in-situ simulations. From all simulation codes discussed above, only OpenFOAM offers a converter to the OpenDX format.

4.3.2 Possible Suitability for the SVD

As seen above, OpenDX is limited in its availability on all platforms that are of interest for CoolEmAll (mainly Windows). Although it is stable and mature, development on OpenDX is discontinued. It does not scale on parallel platforms like e.g. ParaView. There are parallel implementations like OpenDX-MPI, but they are also discontinued. Importers for most major simulation codes do not exist or are publicly accessible, thus they have to be written from scratch. OpenDX can be seen as legacy software. Use of OpenDX in the CoolEmAll project is not recommended.

4.4 COVISE

COVISE stands for COllaborative VIsualization and Simulation Environment. It is an extensible distributed software environment to integrate simulations, post-processing and visualisation functionalities in a seamless manner. From the beginning COVISE was designed for collaborative working, allowing engineers and scientists to work distributed on a network infrastructure. In COVISE, an application is divided into several processing steps, which are represented by COVISE modules. These modules, being implemented as separate processes, can be arbitrarily spread across different heterogeneous machine platforms. Major emphasis was put on the usage of high performance infrastructures such as parallel and vector computers and fast networks and support for collaborative working environments. The COVISE rendering module COVER supports Virtual Environments ranging from workbenches over Powerwalls, curved screens up to full domes or CAVEs and head mounted displays. The users can thus analyse their datasets intuitively in a fully immersive environment through state of the art visualisation techniques, including Volume rendering and fast sphere rendering. Physical prototypes or experiments can be included into the analysis process through Augmented Reality techniques. COVISE runs on all important platforms, the most important being Microsoft Windows, various Linux distributions and MacOS X Intel and PPC. COVISE is free of charge for academic use, for commercial exploitation a commercial license is available, including support and maintenance. A commercial COVISE license starts at €8000.

COVISE consists of several components that play together to create a visualisation. All components run in independent processes and communicate using messages and distributed shared memory. The main processes are the user interface (UI) called MapEditor, the Controller, the COVISE Request Broker (CRB) and the application modules. The local workstation normally is the one where COVISE is started. On this machine the user interface pops up and the Controller is started. All other processes are created from here, either using exec calls for local processes, or ssh and other remote invocation service calls for processes on remote computers. The user can include additional hosts into a session for remote module execution. On each machine a shared data space exists. The CRB administers the use of this shared data space in a database-like fashion. Simulations can be directly integrated into the COVISE workflow, including all necessary pre- and post-processing tasks. Together with a tight integration of all simulation steering and post-processing tasks in the virtual environment, this allows the creation of virtual testbeds, offering rapid assessment and steering of high performance simulations.

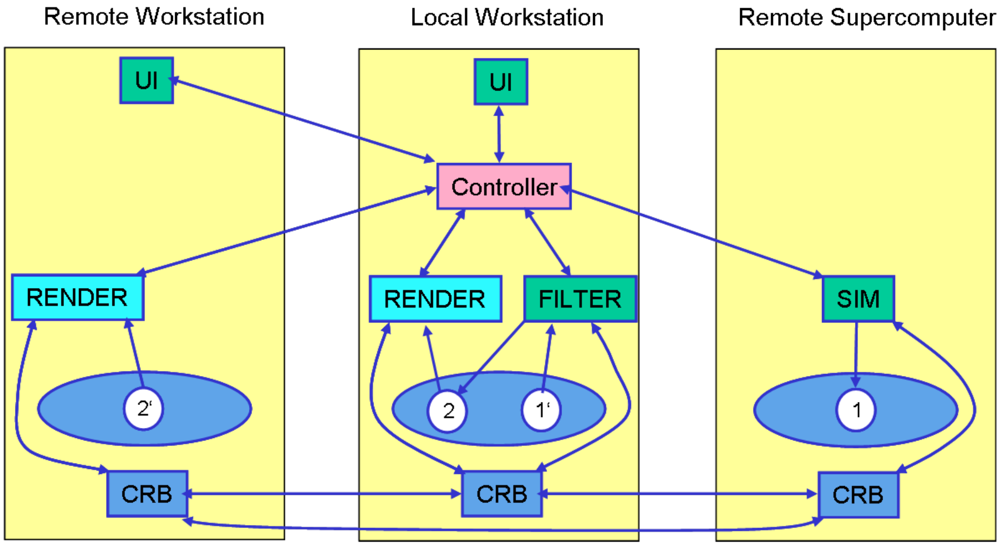


Figure 13: COVISE distributed architecture

COVISE allows several users to work in a collaborative way. One can invite other users on different hosts to participate in the current COVISE session. The user who initiated the collaborative session becomes the master. On all machines a user interface pops up and a CRB is started. Application modules can be started on any host participating in the session. Renderer modules play a special role: in a collaborative session, renderers run locally on each machine. When the master user manipulates the objects in the render window, only small synchronisation information has to be sent to the other renderer modules. When the master rotates the scene, the new transformation matrix is sent to the controller which in turn sends it to all other renderers. Of course, every user can request the master role.

COVISE offers a networked SOAP based API and is accessible by all components that can make use of WebService based components. Thus it is quite convenient to integrate COVISE into existing infrastructures. All aspects of COVISE can be accessed remotely, and even the Rendering Results can be streamed to a local lightweight client.

4.4.1 **Simulation Integration**

COVISE offers a vast amount of reader modules for most of the important simulation codes. It has read modules for a lot of CFD codes including ANSYS CFX / FLUENT and StarCD. Also some multiphysics codes are supported like LS-DYNA, COMSOL or Elmer. It is also possible to import data from other visualisation packages like EnSight or the VTK format used by ParaView.

Through its modular nature, it is easy to integrate running simulation services in COVISE. Those can be manipulated by a multitude of possibilities. Virtual interactors in the virtual environment allow an intuitive interface to boundary conditions and grid generation. Various menus with sliders and checkboxes can be used to also influence those. A 2D interface running on a TabletPC / PocketPC can be used to set specific numerical values. Last but not least, so called “Tangible Interfaces” consisting of a physical mockup of the simulation setup can be used to control mesh generation, boundary conditions and post-processing. Successful integrations of simulation packages include StarCD, FENFLOSS, OpenFOAM and Trans3D.

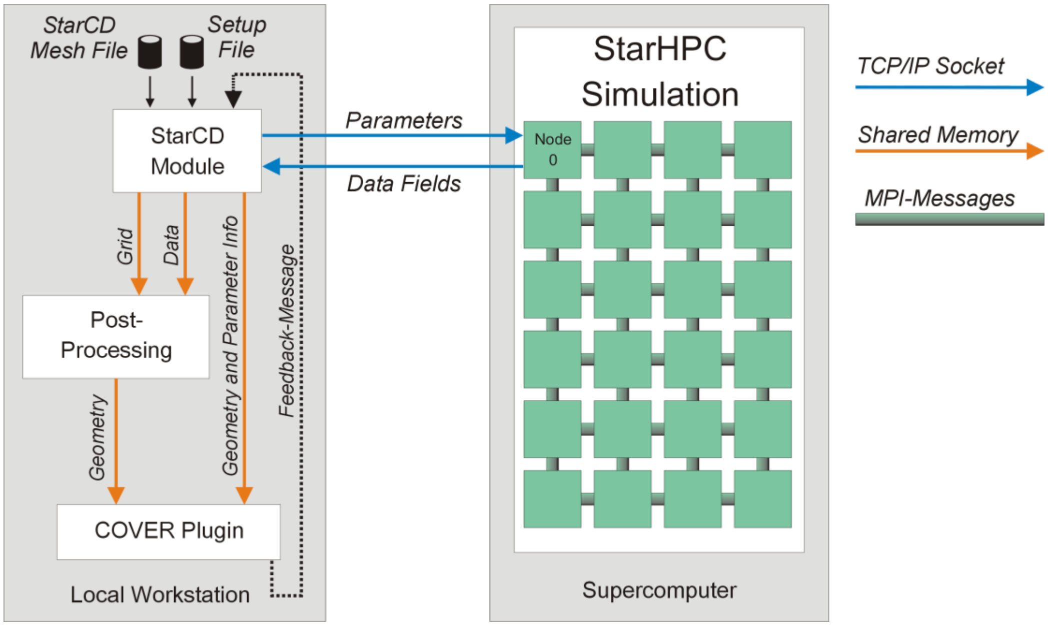


Figure 14: Simulation coupling with COVISE

4.4.2 Possible Suitability for the SVD

COVISE is a mature tool for visualisation that also includes hooks for geometry and grid generation as well as linking directly to various simulation codes. This functionality is proven using several in-house and commercial codes, including ANSYS CFX and StarCD. There is also a limited support for OpenFOAM in place. COVISE has proven to tackle the airflow computation of a data centre and its modular approach makes it easy to include new simulation codes in the workflow. Its extensive SOAP API allows steering COVISE from other components enabling it to fit easily in all emerging WebService oriented infrastructures. The remote rendering capabilities allow using COVISE on a multitude of devices, including mobile platforms. COVISE already includes a grid generator for feeding an airflow computation of a data centre that could be extended for SVD purposes. With its AR capabilities, it is possible to verify the simulation results directly at the rack by overlaying measurements with a rendering of the simulation results. COVISE also sports a multitude of unusual input devices like Tangible Interfaces and 3D interaction devices. Its ability to run on the desktop and in VR environments for single users and collaboratively rounds up the COVISE profile, making it very suitable for inclusion in the final SVD. As it is developed by HLRS, full access to the source code is available and it will be easy to extend COVISE if needed features should be amiss.

4.5 Vitrall

Vitrall [47] is a new, distributed web based visualisation system designed to utilise efficiently multi-GPU server installations. In a nutshell, it allows highly efficient parallel rendering and high-resolution visualisation of well-know data exchange formats among various scientific and engineering applications. Vitrall can be easily installed and maintained on a small GPU server as well as large scale GPU clusters thanks to its distributed and scalable architecture. It can be configured to display virtual scenes in real-time on different visualisation panels, caves or geometric prisms to meet specific users requirements. Moreover, Vitrall can be used in collaborative environments, where many users can simultaneously interact in real-time, share or modify virtual 3D models over the Internet, as the processing part is performed completely on the remote side close to the data sources. The Vitrall application is a very modular solution based on the notion of exchangeable plugins.

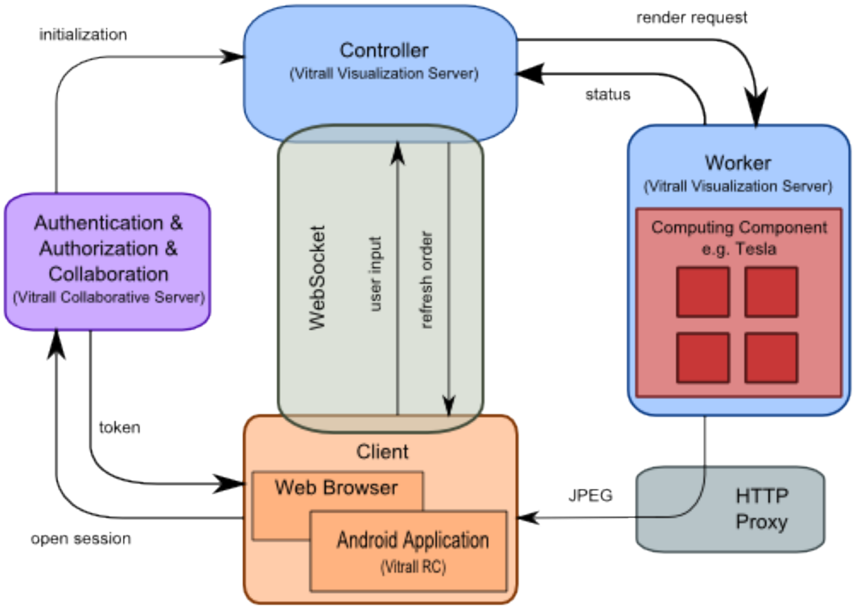


Figure 15: Vitrall architecture

Vitrall is a successor of the Offrenderer project. It is under active development at PSNC. One of the main advantages of Vitrall is that it is an Open Source solution.

The Vitrall rendering abilities are accessible through already existing web protocols: HTTP and WebSockets [48]. HTTP is used for sending compressed images, while user input (information read from sensors and from user forms) is sent with a WebSockets connection. WebSockets technology is a part of the HTML5 specification and is already implemented in most popular web browsers, like Firefox, Chrome, etc.

Currently, two implementations of the Vitrall client are available: one for the web browser and a second for Android [49] based mobile devices.

While the web browser client is very easy to use (in principle it consists of a single web page with some JavaScript code), it has limited input capabilities because of constraints concerning a direct access to sensors from the browser environment. Still, thanks to Firefox's ability to expose the Windows multi-touch interface, Vitrall can provide a multi-touch user experience through a thin web browser client used on a tablet or other Windows based touch-enabled device.

On the other hand, as a part of the Vitrall project, a dedicated Smartphone client was developed for the Android OS platform. This client has full support for sensors like gyroscope, multi-touch, accelerometers and magnetic field sensors. Though it is targeted specifically for work with the Vitrall system, it is a general solution in terms of the different scenarios that can be covered using the Vitrall platform – all information particular to a given scenario like user forms and supported user input methods is sent to the client using an XML based protocol. The Android based client is available in the Android Market and once installed can be used to connect to different Vitrall instances running different visualisation scenarios [50].

It is also worth mentioning that some web browsers available on mobile devices, Safari on iPad for example, already implement the WebSockets specification, allowing the usage of the Vitrall system just out-of-the-box.

One of the outstanding features of the Vitrall system is its flexibility with regard to defining the way how user input should be interpreted. All sensor information is sent in raw form: points of contact for multi-touch enabled devices (Smartphones, tablets), orientation in space for magnetic field sensors and gyroscopes, etc. That information is then interpreted by the Vitrall server, producing signals of a more semantic nature (rotate, translate, click) that are then routed to elements of the 3D scene in a way specific to the visualisation scenario that can be changed dynamically.

Vitrall prototype versions have been successfully demonstrated on various international conferences and workshops, including Supercomputing 2010 [51] and 2011 [52] or TERENA 2011 [53]. Another version of Vitrall has been prepared for the "Tower of Interaction" demo at Future Internet Week in Poznan [54] aiming at a visualisation scenario on 18 HD screens. During the demo, Vitrall provided content for the more than 41 megapixel hexagon prism shaped visualisation tower that rendered 3D objects in real-time. Each side of the hexagon prism presented the rendered object from a different angle with the view shifted by 60 degrees around the tower axis. That demonstration showed the ability to render in parallel (using a multi GPU environment) from many, arbitrarily defined points of view.

4.5.1 **Simulation Integration**

In the scope of the CoolEmAll project it is important to note that Vitrall uses well-known solutions not only regarding the choice of Internet protocols and data formats, but also for the underlying rendering engines. Vitrall uses the OpenSceneGraph [55] library for management of the 3D scene, and an integration of the VTK library is planned for the near future. This should allow a relatively easy definition of a shared interface between the simulation codes and visualisation tools such as COVISE or ParaView and Vitrall.

After the simulation is done, the result can be read from a file or downloaded through HTTP. Data can then be loaded into Vitrall using VTK, for example. Thanks to the Vitrall plugin system it is possible to load data using a dedicated library (if needed), or to implement other communication protocols (for example a virtual file system).

4.5.2 Possible Suitability for the SVD

Vitrall is a visualisation toolkit that enables remote access to visualised models. In particular, it provides a standard web-based interface as well as a thin client for Android. The Vitrall collaborative environment with its innovative natural user interfaces makes it suitable for those CoolEmAll scenarios in which multiple users would like to easily control and analyse simulations results or computational experiments.

Because of the mentioned low requirements (related to both hardware and software) for the Vitrall clients, real-time visualisation can be accessed easily from almost any remote location. In future, if the CoolEmAll SVD toolkit was to be developed even further to visualise real-time data from thermal sensors, that ability could be used by administrators of data centres to instantly watch the visualisation of what is currently happening. It could also be used to directly compare the simulation results to real data for verification purposes. Moreover, the collaborative aspects of Vitrall could be used to enable data centre planners and administrators to share information visually.

The Vitrall core component is free and Open Source, which makes it an interesting complementary option to other tools (such as COVISE) in case of users not able or willing to conform to the licensing schemes of these tools.

It is also worth noting that Vitrall is developed by PSNC. Therefore, changes related to novel functionality or integration that are required to meet CoolEmAll objectives could be relatively easy to perform.

5 Conclusions and Outlook for the SVD Toolkit

When looking at the different simulation codes, it can be seen that all of them can handle coupled heat transfer and flow simulation on a high level. The main differences are in costs (commercial vs. free), availability of source code (Open Source vs. closed source), simplicity (general purpose vs. specialised data centre oriented packages) and extensibility (black box, grey box and white box). For the coming SVD, it will be important to use codes that are cheap and freely available to the developers and possibly also to the future users of the system. Most important is the possibility to integrate the technologies developed in the other CoolEmAll work packages into the simulation workflow. Thus, integrated packages like TileFlow are easy and convenient to use, but unsuitable for CoolEmAll purposes, as they are only operable through their own, limited user interface and write their data in a closed, proprietary file format. When looking at the costs, it is naturally favourable to choose a code that is available to all partners at low costs for the development phase. Thus, solutions like OpenFOAM that are freely available but may be hard to tackle because of the lacking documentation are on one of the top places of the list when deciding which code to use. When finally deploying the SVD, costs may play a lesser role, depending on the chosen exploitation path and target end users. Thus, other codes can be chosen that may be easier to install and set up. A decision what aspect weights more can only be made when a more detailed design of the SVD is underway.

From looking at the products dedicated to data centre simulation it can be seen that these will most likely be too closed in their nature to be successfully be integrated into the vision of CoolEmAll. They are all black box applications that can only be accessed from the native user interface. Automation and data export is not available with those packages. All the dedicated data centre packages included in this survey offered a data import for the rack data in some way or the other. This can only be done using the user interface though and cannot be done automatically. Thus, their use for data centre assessment is limited as the power consumption and heat production boundary conditions for the cooling simulation cannot be updated easily. Also often the reporting capabilities of the dedicated data centre packages are limited to graphical reports, more targeted at a meeting presentation than on the further processing of the resulting simulation data. 6SigmaRoom seems to be an exception here, but as they only provided the Lite version of their package for evaluation purposes that does not include these capabilities, it is only possible to trust in the statements of the sales representatives. Thus, further hands-on investigation is necessary. Also, the main problem of most integrated packages can be the solver. As the user base of the underlying simulation code is much smaller than that of the general purpose codes – they are only used in these dedicated tools, besides CoolSim that uses FLUENT for simulation purposes – it can also be assumed that the general stability of the code of the former is less than that of the latter ones. It is thus not recommended to use current data centre solutions but to rely on general purpose simulation codes for integration.

Referring to the workload simulation tools, it was shown that most simulators basically focus on addressing specific problem areas. Since their general goals are different, they vary in terms of simulated architectures and usually allow modelling and simulating only a subset of distributed computing entities. One should note that there is a lack of tools that allow performing specific actions on resources and in that way affects their behaviour. Moreover, there is very little support for application performance modelling and incorporation of virtualisation techniques. Only a few tools, including GridSim & Cloudsim, GreenCloud and GSSIM offer the functionality suitable for a future SVD toolkit.

Considering the CoolEmAll requirements for the workload simulation tools, it can be seen, that GSSIM offers the most comprehensive simulation environment. Although all of the aforementioned simulators provide means for simulating distributed architectures, only GSSIM is able to model typical structures for data centres with nodes organised within racks or containers. For the SVD Toolkit, it is crucial to evaluate various resource management policies that take into account different aspects of energy- and thermal-aware scheduling. Thus, solutions that address this issue are desirable. CloudSim is targeted to simulate Cloud computing infrastructures. However it provides only a basic model for the evaluation of energy consumption. The main advantage of GreenCloud is the sophisticated energy module. The predominant features of both packages are mostly covered by the GSSIM simulator. Additionally, it supports application performance modelling which is an essential feature to accurately simulate the impact of workloads on energy consumption and temperature. Nevertheless, one should note that none of the simulators provide the means to take into account thermal aspects. This drawback needs to be overcome for the purposes of the CoolEmAll SVD Toolkit. It is worth to mention, that the GSSIM simulation tool is developed by one of the CoolEmAll participants – PSNC. Thus, detailed expertise in workload simulation with GSSIM can be provided to all CoolEmAll partners. Moreover, that also enables possible adjustments and immediate modifications of the workload simulation tool in order to satisfy all the requirements and facilitate the integration process.

For visualisation, similar criteria apply. Most of the considered packages follow a modular approach and are easily extensible. Seen from this point, all packages are suitable besides OpenDX that can be considered end-of-life. ParaView available for free and has a high reputation in the field of visualisation. But it has yet to be successfully used in in-situ simulations. The same is valid for EnSight that – considering the costs – will most likely be less suitable for CoolEmAll than other solutions. COVISE is somewhere in between ParaView and EnSight. As a highly modular system can be extended by all functionality needed by CoolEmAll, easier than it would be using ParaView. As HLRS as the developer is one of the partners internal modifications could be done quite easily if needed and be fed back in the main development tree. As for project partners and educational users COVISE is license free, costs are no issue while developing the SVD Toolkit. All visualisation requirements of a SVD Toolkit are also fulfilled by Vitrall – it is a modular solution that is developed by PSNC. If during the design process any additional required functionality is identified, it can be easily added to the software, since it is Open Source and developed by a project partner. Due to capabilities like thin client support and natural user interfaces for mobile devices, Vitrall is considered as a complementary solution to COVISE.

In this document, an overview is given on the predominant simulation and visualisation tools available on the market. It will serve as a decision guidance for the coming design process of the SVD Toolkit and its resulting design document D2.2 without trying to anticipate the final design decisions. Nevertheless, there are some guidelines already given in the text as well as some estimation on the suitability for the Toolkit. The requirements that served as a basis for these estimations may change in the course of the SVD design, thus resulting in a different view in the final design.

6 Bibliography

[1] OpenCFD Ltd., “OpenFOAM – The Open Source Computational Fluid Dynamics (CFD) Toolbox,” 2011. [Online]. Available: http://www.openfoam.com/. [Accessed 2011].

[2] Wikipedia, “OpenFOAM,” 2011. [Online]. Available: http://en.wikipedia.org/wiki/OpenFOAM. [Accessed 2011].

[3] OpenCFD Ltd., “OpenFOAM – Standard Solvers,” 2011. [Online]. Available: http://www.openfoam.com/features/standard-solvers.php. [Accessed 2011].

[4] ANSYS, [Online]. Available: http://www.ansys.com/Products/Simulation+Technology/Fluid+Dynamics/ANSYS+CFD. [Accessed 09 01 2012].

[5] L. M. Paul Bemis, “Improving Data Center PUE Through Airflow Management,” 2010.

[6] CoolSim, “CoolSim Data Center CFD Modeling Purchase Options,” 2011. [Online]. Available: http://www.coolsimsoftware.com/AboutCoolSim/PurchaseOptions/tabid/44/Default.aspx. [Accessed 2012].

[7] COMSOL, [Online]. Available: http://www.comsol.com/products/multiphysics/. [Accessed 09 01 2012].

[8] CD-adapco, “CD-adapco - Products - STAR-CCM+,” 2012. [Online]. Available: http://www.cd-adapco.com/products/star_ccm_plus/index.html. [Accessed 2012].

[9] CD-adapco, “STAR-CCM+ Version 6.04.014 User Guide,” 2011.

[10] Scheider Electric, “Schneider Electric’s StruxureWare Operations 7.0 Software Platform Brings 3D Airflow Analysis and Smart Phone Support to Enable Improved Data Center Operations - Press Center,” [Online]. Available: http://www.apc.com/site/press_center/index.cfm/schneider-electrice28099s-struxureware-operations-70-software-platform-brings-3d-airflow-analysis-and-smart-phone-support-to-enable-improved-data-center-operations1/. [Accessed 2012].

[11] Schneider Electric, “APC - Statement of Work,” 2012.

[12] Schneider Electric, “Products - Data Center Software,” 2012. [Online]. Available: http://www.apc.com/site/software/index.cfm/Products/?ID=476. [Accessed 2012].

[13] Future Facilities Ltd., “6SigmaDC Data Center Design & Management CFD Simulation Software,” 2011. [Online]. Available: http://www.futurefacilities.com/software/6SigmaDCOverview.php. [Accessed 2012].

[14] Innovative Research, Inc., “TileFlow,” 2011. [Online]. Available: http://inres.com/products/tileflow/. [Accessed 2012].

[15] R. R. Schmidt, K. C. Karki, K. M. Kelkar, A. Radmehr and S. V. Patankar, “Measurements and Predictions of the Flow Distribution Through Perforated Tiles in Raised-floor Data Centers,” in *Proceedings of IPACK’01*, Kauai, 2001.

[16] A. Sulistio, C. Yeo and R. Buyya, “A taxonomy of computer-based simulation and its mapping to parallel and distributed systems simulation tools,” *International Journal of Software: Practice and Experience,* vol. 34, p. 653–673, 2004.

[17] S. Bąk, M. Krystek, K. Kurowski, A. Oleksiak, W. Piątek and J. Węglarz, “GSSIM - a Tool for Distributed Computing Experiments,” *Scientific Programming Journal,* vol. 19, no. 4, pp. 231-251, 2011.

[18] R. Buyya and M. Murshed, “GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing,” *The Journal of Concurrency and Computation: Practice and Experience,* vol. 14, pp. 1175-1220, 2002.

[19] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose and R. Buyya, “CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms,” *Software: Practice and Experience (SPE),* vol. 41, no. 1, pp. 23-50, 2011.

[20] D. Klusacek and H. Rudová, “Alea 2 - Job Scheduling Simulator,” in *3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools 2010)*, Brussels, 2010.

[21] H. Casanova, A. Legrand and M. Quinson, “SimGrid: a Generic

Framework for Large-Scale Distributed Experimentations," in *10th IEEE International Conference on Computer Modelling and Simulation (UKSIM/EUROSIM)*, 2008.

[22] S. Ostermann, K. Plankensteiner and R. Prodan, "Using a New Event-based Simulation Framework for Investigating Different Resource Provisioning Methods in Clouds," *Scientific Programming Journal,* vol. 19, no. 2-3, pp. 161-178, 2011.

[23] K. Kurowski, J. Nabrzyski, A. Oleksiak and J. Węglarz, "GSSIM – grid scheduling simulator," in *Computational Methods in Science and Technology*, vol. 13, Poznań, Institute of Bioorganic Chemistry Scientific Publishers OWN Polish Academy of Sciences, 2007, pp. 121-129.

[24] D. Kliazovich, P. Bouvry and S. Khan, "GreenCloud: A Packet-Level Simulator of Energy-Aware Cloud Computing Data Centers," *The Journal of Supercomputing,* 2010.

[25] Information Sciences Institute, University of Southern California, "The Network Simulator Ns2," 2011. [Online]. Available: http://www.isi.edu/nsnam/ns/. [Accessed 2011].

[26] D. Feitelson, "Parallel workload archive," 2011. [Online]. Available: http://www.cs.huji.ac.il/labs/parallel/workload/. [Accessed 2011].

[27] Delft University of Technology, "Grid workloads archive," 2007. [Online]. Available: http://gwa.ewi.tudelft.nl/. [Accessed 2011].

[28] M. Krystek, K. Kurowski, A. Oleksiak and W. Piątek, "Energy-aware simulations with GSSIM," in *COST Action IC0804 on Energy Efficiency in Large Scale Distributed Systems*, 2010.

[29] K. Kurowski, J. Nabrzyski, A. Oleksiak and J. Węglarz, "A multicriteria approach to two-level hierarchy scheduling in grids," *Journal of Scheduling,* vol. 11, pp. 371-379, 2008.

[30] K. Kurowski, A. Oleksiak and J. Węglarz, "Multicriteria, multi-user scheduling in Grids with advance reservation," *Journal of Scheduling,* vol. 13, no. 5, pp. 493-508, 2010.

[31] M. Mika, G. Waligóra and J. Węglarz, "Modelling and solving grid resource allocation problem with network resources for workflow applications," *Journal of Scheduling,* vol. 14, pp. 291-306, 2011.

[32] Consortium GARR, "Federica," 2011. [Online]. Available: http://www.fp7-federica.eu/. [Accessed 2011].

[33] PSNC, "The Grid Scheduling Simulations Portal," 2011. [Online]. Available: http://www.gssim.org. [Accessed 2011].

[34] Kitware, "ParaView – Open Source Scientific Visualization," 2011. [Online]. Available: http://www.paraview.org/. [Accessed 2011].

[35] J. Ahrens, B. Geveci and C. Law, "ParaView: An End-User Tool for Large Data Visualization," in *The Visualization Handbook*, Elsevier, 2005.

[36] Kitware, "VTK – The Visualization Toolkit," 2011. [Online]. Available: http://www.vtk.org/. [Accessed 2011].

[37] Kitware, "ParaView – Python Programmable Filter," 2011. [Online]. Available: http://www.paraview.org/Wiki/Python_Programmable_Filter. [Accessed 2011].

[38] Python Software Foundation, "Python Programming Language – Official Website," 2011. [Online]. Available: http://www.python.org/. [Accessed 2011].

[39] J. Biddiscombe, J. Soumagne, G. Oger, D. Guibert and J.-G. Piccinali, "Parallel computational steering and analysis for HPC applications using a ParaView interface and the HDF5 DSM virtual file driver," in *EGPGV*, 2011.

[40] The HDF Group, "Summary of Software Using HDF5," 2011. [Online]. Available: http://www.hdfgroup.org/products/hdf5_tools/SWSummarybyName.htm. [Accessed 2011].

[41] CEI Software, "CEI - Creators of EnSight Visualization Software," 2010. [Online]. Available: http://www.ensight.com/. [Accessed 2012].

[42] CEI Software, "CFD Data Formats," 2012. [Online]. Available: http://www.ceisoftware.com/cfd-data-formats/. [Accessed 2012].

[43] CEI Software, "Feature Table," 2012. [Online]. Available: 2012.

[44] CEI Software, "North America Price List," 2012. [Online]. Available: http://www.ceisoftware.com/purchase/price-list-north-america/.

[45] B. Dunn, “Co-Processing,” in *CEIViz'08*, Baltimore, 2008.

[46] Wikipedia, “IBM OpenDX,” 2011. [Online]. Available: http://en.wikipedia.org/wiki/Opendx. [Accessed 2011].

[47] P. Śniegowski, M. Błażewicz, G. Grzelachowski, T. Kuczyński, K. Kurowski and B. Ludwiczak, “Vitrall: web-based distributed visualization system for creation of collaborative working environments.,” in *Parallel Processing and Applied Mathematics*, Toruń, 2011.

[48] W3C, “The WebSocket API,” 2011. [Online]. Available: http://dev.w3.org/html5/websockets/.

[49] Google, “Android,” 2011. [Online]. Available: http://www.android.com/. [Accessed 2011].

[50] G. Grzelachowski, T. Kuczyński and P. Śniegowski, “VitrallRC,” 2011. [Online]. Available: https://market.android.com/details?id=org.vitrall.client. [Accessed 2011].

[51] PSNC, *Live demonstration at PSNC booth. Supercomputing'10 (15-18.11.2010),* New Orleans, 2010.

[52] PSNC, *Vitrall: Mobile-controlled Real-time 3D Visualization - live demonstration at PSNC booth. Supercomputing'11 (14-17.11.2010),* Seattle, 2011.

[53] PSNC, *Live demonstration: Stereoscopic 4K3D rendering from Poznan to Prague. Terena Networking Conference 2011 (16-19.05.2011),* Prague, 2011.

[54] PSNC, *Tower of Interaction - live demonstration at the conference hall. Future Internet Week Poznań (24-28.10.2011),* Poznań, 2011.

[55] OSG Community, “Open Scene Graph Official Website,” 2011. [Online]. Available: http://www.openscenegraph.org/. [Accessed 2011].

[56] J. Rambo and Y. Joshi, “Modeling of data center airflow and heat transfer: State of the art and future trends,” *Distributed and Parallel Databases,* vol. 21, no. 2, pp. 193-225, 2007.