D5.2.5: Post processing tools for interactive data visualization and exploration

WP5: User tools

Project Acronym	CRESTA
Project Title	Collaborative Research Into Exascale Systemware, Tools and Applications
Project Number	287703
Instrument	Collaborative project
Thematic Priority	ICT-2011.9.13 Exascale computing, software and simulation

Due date:	M30
Submission date:	31/03/2014
Project start date:	01/10/2011
Project duration:	39 months
Deliverable lead organisation	DLR
Version:	1.0
Status	Final
Author(s):	Fang Chen (DLR), Markus Flatken (DLR), Timo Kiviniemi (Aalto University, Finnland)
Reviewer(s)	Luis Cebamanos (UEDIN) Mats Hamrud, ECMWF

Dissemination level	
PU	*PU - Public*

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	20.03.2014	First version of the deliverable	Fang Chen (DLR), Markus Flatken (DLR), Timo Kiviniemi (Aalto University, Finnland)
1.0	25.03.2014	Final version of the deliverable	Fang Chen (DLR), Markus Flatken (DLR), Timo Kiviniemi (Aalto University, Finnland)

Table of Contents

Index of Figures

Index of Tables No table of figures entries found.

1 Executive Summary

In this deliverable, we present the post processing tools and systems designed for enabling data exploration and post processing towards exa-scale. Providing run-time inspection to the ongoing simulation and enabling interactive exploration of the simulation are two major challenges for post processing for large simulations.

While pre-processing of the simulation focuses on mesh creation and partitioning, post-processing of the simulation is targeted at providing visualisations of the simulation outputs, which serves as a tool to explore and analyse the simulation results.

A key concept in data post processing for exa-scale simulation is to provide In-situ data inspection and to minimize moving data around. Inspecting simulation results at run time and providing a first visualization allows the simulation experts to monitor the process of the running simulation and to prevent early failure.

In this work package, we present the in-situ monitoring tool and the underlying system designed and for HemeLB which is further applicable for large simulations in general. Such on-line monitoring client/system provides the user with an in-situ inspection of the on-going simulation. It does not require outputting data to local storage. Instead, rendered image cache is streamed to the monitoring client, thus minimizes the data transfer and keeps maximum data locality for large parallel systems.

In comparison to Deliverable 5.2.4, which presents a system algorithm review, this deliverable continues with the software development of the online-monitoring parts and the front-end interactive applications. Aside from implemented the in-situ monitoring tool, we further integrate the scientific visualization algorithm, which allows further investigation of the HemeLB simulation output. While the in-situ monitoring provides a first step in inspecting run-time simulation results, an integrated visualization system in virtual environments will enable the user with intuitive and explorative perception of each simulation time-step. Moreover, we also present the recently initiated co-design work with an ELMFIRE simulation.

2 Introduction

With the ever increasing computational power of the hardware systems, fluid simulation is heading towards exa-scale computing. The size and complexity of the simulation data present new challenges to data post-processing frameworks and visualization algorithms. Major challenges in post-processing will be that no data is stored on disk and moving data around will be very costly. How to make the best use of existing hardware systems and what exa-scale post-processing algorithms are the questions we try to answer.

The post-processing of simulation data is a process which transforms simulation output into suitable visual representations. The so called post-processing pipeline typically consists of data extraction, filtering, mapping and visualization stages. Visualization—the creation of vivid pictures from simulation outputs in the form of arrays of numbers—has become an indispensable tool for scientists (Ma, et al., 2009). At exascale, it becomes a time-consuming process where efficient and interactive visualization can be a challenging task for post-processing.

The aim of work package 5.2.5 is to present the system, algorithm and tools that we designed and developed for exa-scale post processing applications. In section 2 and 3, we first present the post-processing system architecture and requirements for applications towards exa-scale problem. In section 4, we present the co-designed post processing tools for HemeLB application. Section 5 brief the on-going co-designed for EIMFIRE application.

3 System Architecture

In this section, we briefly describe the system architecture of our post-processing tools. We will elaborate on the system layout in a detailed manner with respect to the HemeLB application in the later sections.

Error! Reference source not found. demonstrates a general post-processing system

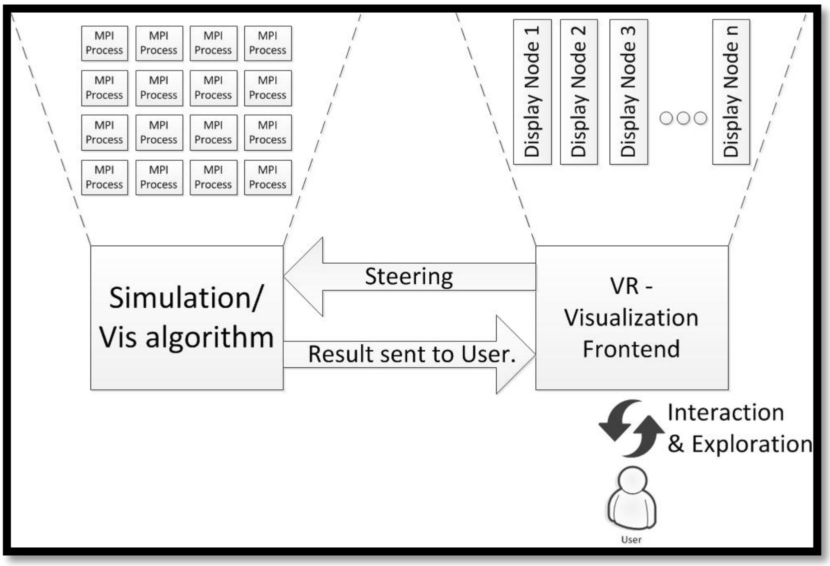


Figure 1 Interactive post processing system architecture for exa-scale systems

for an exa-scale system. To avoid moving data around, the visualisation shares the same process as the simulation. At the same location, simulation output will be cached and visualised. A master node will control and collect not only simulation results, but also visualization output, and send these back to the user front end.

For the purpose of in-situ monitoring, the user front end can be just a single display (see Figure 2) which demonstrates the run-time results of the current simulation step. In this type of system, only an image with given resolution is transmitted to the front screen and thus provide a first insight into the running simulation. The advantage of such a set-up is the minimal amount of data moved over the network, and results in a low latency between the remote systems and the front monitor.

To inspect the data in a more detailed way, a front end can also be a more complex system which utilizes virtual reality techniques that allows interactive user interaction with the data (see Figure 3 . Within this approach, a set of data or image is required to be collected and stored somewhere by the scheduler, which allows for interactive exploration request sent by the VR front end.

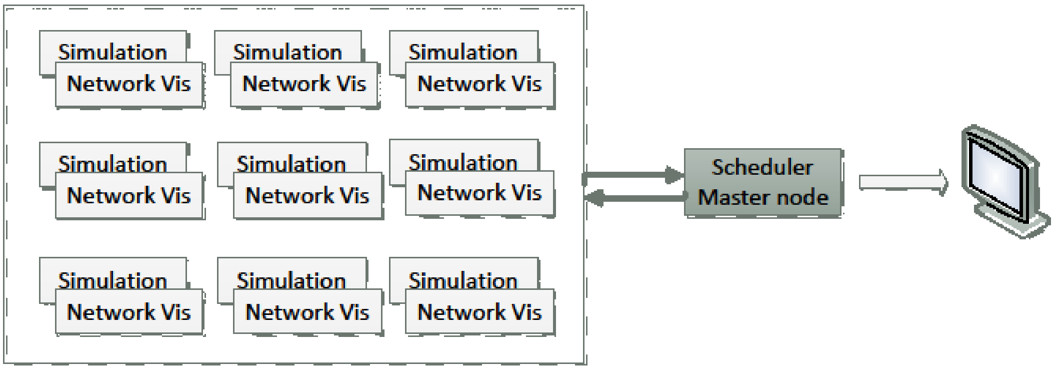


Figure 2 An In-situ monitoring with only a screen showing the network image streamed to the front-end application

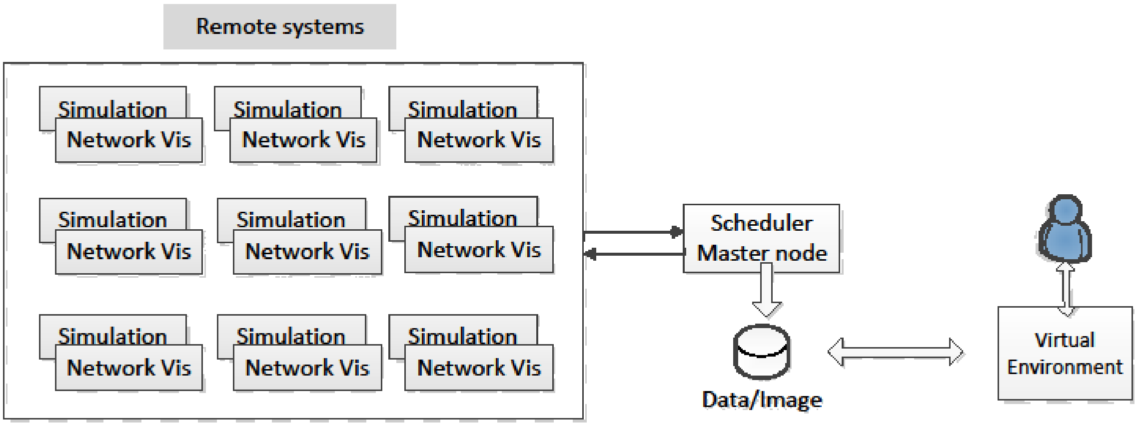
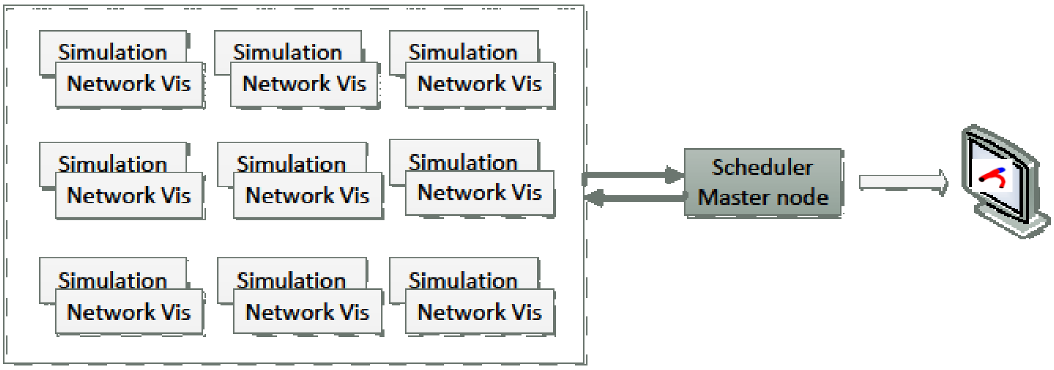


Figure 3 An Interactive post processing architecture with a user interacting on the front end

The different between the two systems is that the former only sends rendered images over networks from remote system to frontend, thus minimize data movements. The latter system requires more communication between the frontend and remote systems, but allows in-depth and intuitive exploration of the current simulation time step. For monitoring a rapid running simulation process the former one is recommended, while for in-depth analysis of the simulation output the second architecture is more suitable.

4 Co-design with HemeLB

4.1 Online-monitoring for a remotely located running simulation



An online-monitoring tool is implemented in python which is able to monitor a large running simulation that is running remotely on a cluster systems without pausing or writing out data to disk. A demonstration of this tool is showing in the video above. A HemeLB simulation is running on a cluster system in real time, and based on the already implemented volume mapping from HemeLB, the online-monitoring client access the network image produced rendered with a resolution of 1024x1024, transfer it over to the frontend and display it as a glTexture on the monitoring window.

Due to the different configurations of the remote cluster systems, communication and connection between front-end application and remote clusters are restricted and also affects the in-situ processing system layout. We specify the two cases in the following section.

4.2 Dynamic allocation of job master node and ssh tunnels

The remote simulation that we want to monitor could be running on another machine, or another cluster, or a cluster within the same perimeter network, or a cluster in a different perimeter networks (also known as Demilitarized Zones (DMZ)). Different firewall settings on different perimeter networks could prohibit direct access and communication between master nodes on a cluster and your local machine. For instance, direct access to working desktops at DLR is not allowed due to security and firewall settings.

Depending on the network characteristic of the remote and the front-end (local machine) system, we consider the following two cases. First, if direct connection between the compute node and the local machine is not allowed (e.g. A cluster in the UK and a local machine at DLR), the results will be first stored at the user's working directory and then copied to la ocal machine for further processing and analysis.

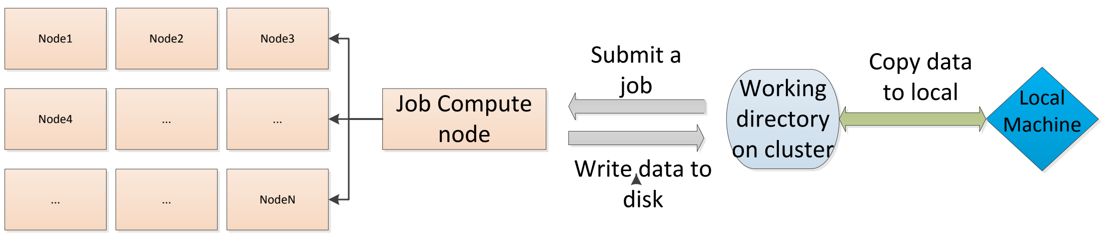


Figure 4 When no directly access to compute node is allowed, data will be first stored on the user's home directly, and then copied to local machine for out-of core access.

In the second case, when direct connection is permitted between a remote computing master node and a local machine, (for example a cluster system and local machine which both reside in the same perimeter network), then we set up an ssh tunnel between the computing master node and the local machine. In this way, the local machine is able to access data on the fly without writing them out. Note that only after submitting the job to the remote system, will the master commutating node be assigned. Therefore, the ssh tunnel can only be set-up after the dynamic allocation of the master node id, see Figure 5.

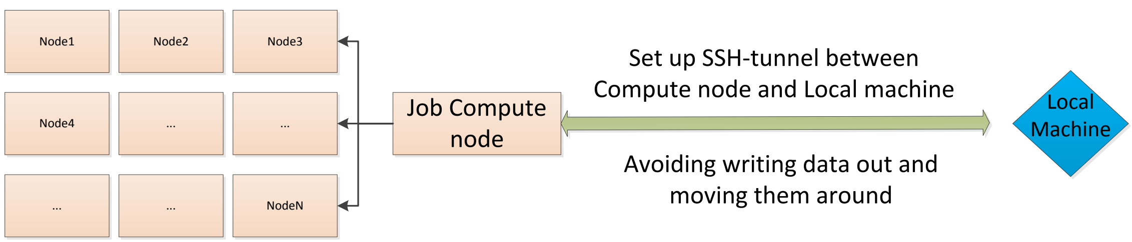


Figure 5 When direct access to compute node is allowed, we dynamically allocated the compute node id, and then set up a tunnel to allow communication between the node which is actually executing the job and our local machine.

4.3 Post processing of the HemeLB simulation

We benchmark the online monitoring tool with the HemeLB simulation in two aspects. First, we benchmark the performance and time needed to perform one step simulation and generate one network image. Then we measure how the image resolution affects the frame-rates on the front-end.

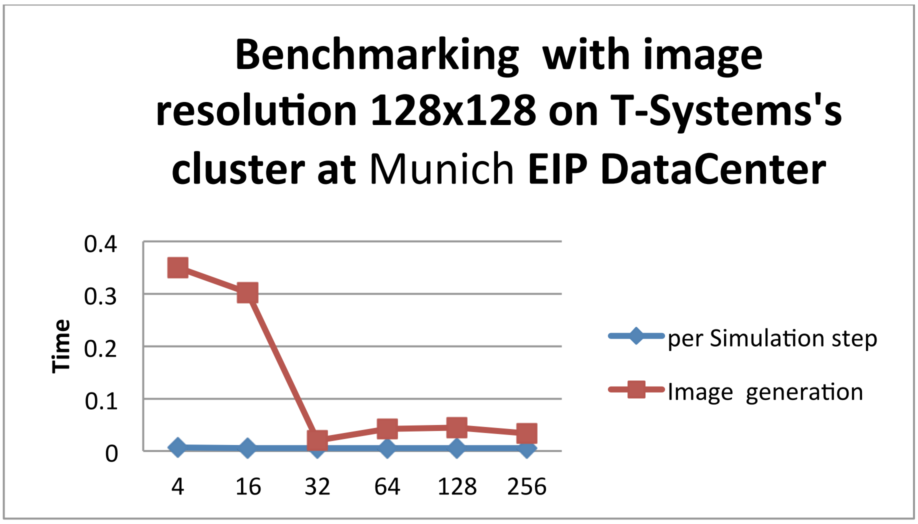


Figure 6 Time measurement for generating image with resolution 128x128

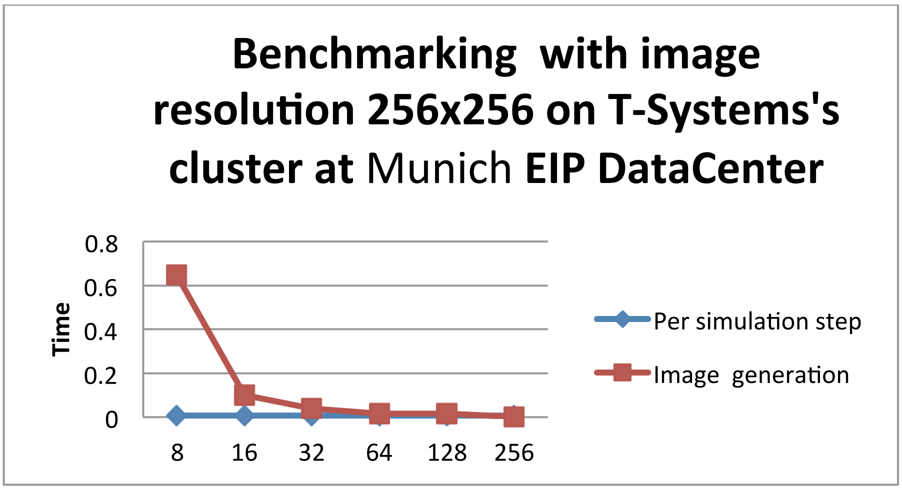


Figure 7 Time measurement for generating image with resolution 256x256

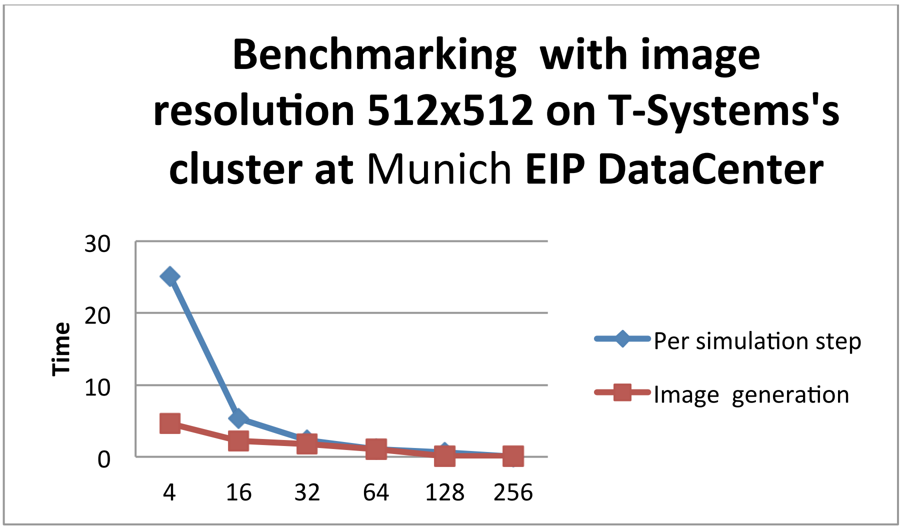


Figure 8 Time measurement for generating image with resolution 512x512

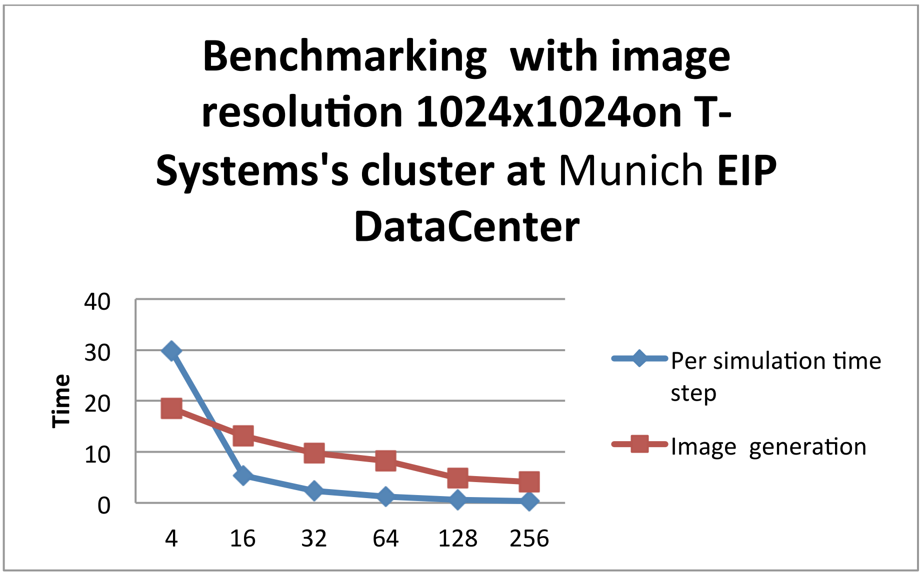


Figure 9 Time measurement for generating image with resolution 1024x1024

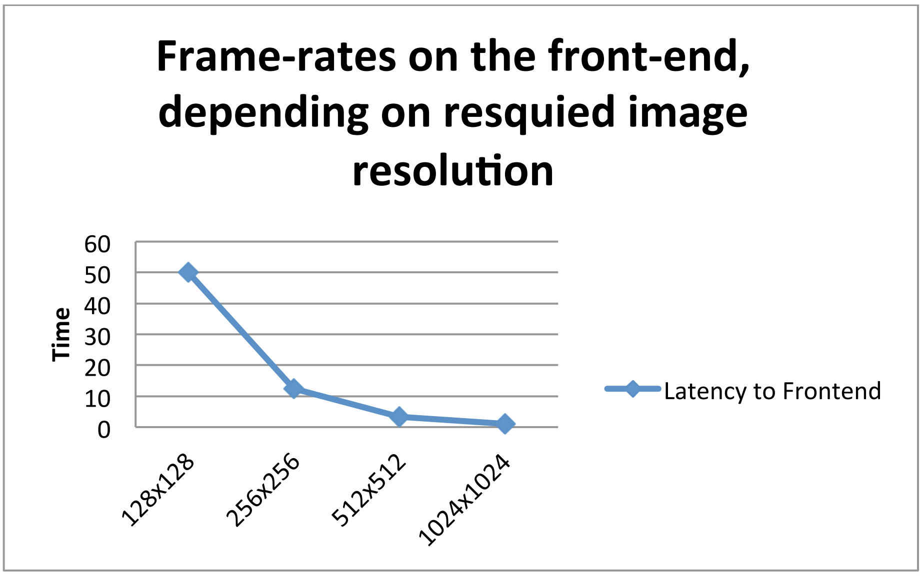


Figure 10 Latency (Front-end framerates) for image display at front-end

We measured the time that is needed to composite an image for the front-end with image resolution 128x128, 256x256, 512x512 and 1024x1024 pixels (respectively see Figure 6, Figure 7Figure 8Figure 9). For each given image resolution, we also measure the latency from the remote to front-end in terms of frame rates, see Figure 10.

Comparing Figure 8,Figure 9Figure 10 to Figure 7, the time needed for generating an image increased as the required image resolution increases. At resolution 128x128 and 256x256, the scaling curve for image generation does not decrease dramatically. This is due to the fact that at smaller resolutions, the image generation is quickly finished and the time needed to collect the data as well as communication among cores remains more or less the same. While going to a higher image resolution (Figure 10), we can observe that there is an obvious decreasing trend in the time needed for image generation with more computational cores.

We observe that with more cores, the computation time for simulation and image generation decreases. However, the non-linear decrease is expected due to the fact that, with increasing number of cores, more time is needed to collect the image from each single core and compose them together. Moreover, with the increased image resolution that is required by the front-end (online monitoring client), the frame rates on the front-end decreases.

In the following figure and video (Figure 11), we demonstrate the interactive exploration tool developed at DLR for analysing an aneurysm data that is based on the described system configuration (Figure 4). In front of a power-wall, the user is able to interact with the Aneurysm dataset, seed particles in the blood flow and trace the dynamics of the blood within the aneurysm. The stereo view in front of the power-wall together with an interacting fly-stick enables the user to naturally navigate through the dataset, allowing intuitive and in-depth exploration of the blood simulation output.

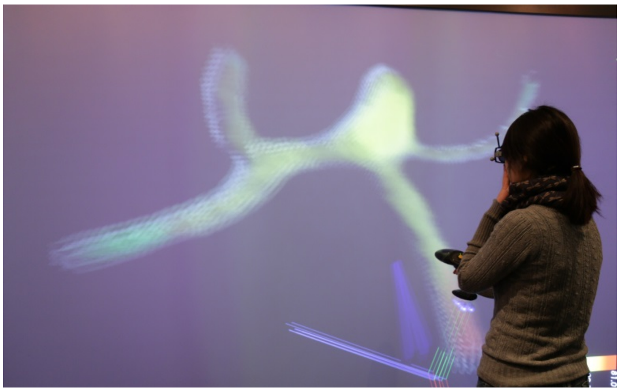


Figure 11 User interacting with an Aneurysm data in front of a power-wall. More details in video <Cresta.mpeg>. Video can be found at CRESTA's svn server https://svn.ecdf.ed.ac.uk/repo/ph/cresta/wp5/postprocessing

5 Co-design with ELMFIRE

During the past six months, we have also established collaborative research with the ELMFIRE simulation group at the Aalto University, Finland. This collaborative work involves developing ideas in data analysis and post processing for EIMFIRE simulations.

5.1 Scientific use case: Gyro-kinetic first principles simulations of plasma turbulence for tokomaks

Understanding turbulent transport is needed for further optimization of fusion reactors but realistic transport time scale simulations of plasma turbulence are computationally very demanding. The aim of the recent ELMFIRE simulations is to increase the understanding of the mechanisms behind the sudden improvement in confinement observed in experiments by investigating the possible role of the radial derivative of a time-varying electric field in triggering transition. The ELMFIRE turbulence simulation code investigates these phenomena with a so-called first principal computer model. This model tracks individual particles providing information on the complex interplay between the magnetic field, the electric field and the particle trajectories. Using the 30 million CPUh granted from 4th PRACE call, a scan over local parameters such as temperature and density starting from experimental Textor parameters was carried out starting from a Textor low confinement mode case for which strong oscillation was observed in the simulations (Kiviniemi, et al., 2012)

5.2 Post processing for Magnetic field simulations

Our collaborative research with the ELMFIRE application aims to develop post processing tools to visualize and analyses the simulation outputs. With increasing computational power, ELMFIRE simulation experts are carrying out simulations of a full distribution of electrons and ions with large numbers of particles. Current simulations are able to produce plausible results for a small FT-2 tokamak (R=0.55 m) and further simulation activities are towards the simulation of one third of a middle-sized Textor tokamak (R=1.75 m).

With the increasing number of particles and geometry complexity, ELMFIRE simulation results in a huge data output that requires new post processing algorithms that can not only handle the large amount of data, but also provide further in-depth analysis of the simulation output. To visualize large simulation data, we need to explore parallel computation of visualization techniques which optimizes the performance of the post processing computations. Scientific visualization, in particular feature extraction of the data provides further analysis of the simulation output.

To bring post processing for ELMFIRE to the next level, the following two key aspects should be considered:

5.2.1 Three dimensional visualization and feature analysis

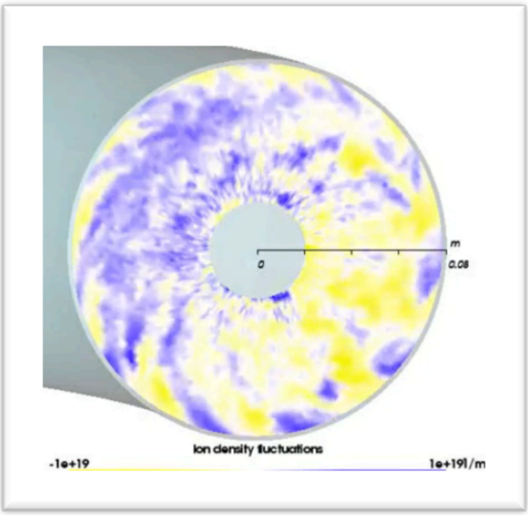


Figure 12 Cross section of the simulation output. From video provide d by Timo Kiviniei, University of Aalto

While simulation experts are still relying on two dimensional color coded results to analyzing their results (Figure 12), scientific visualization, in particular three dimensional feature visualization, can provide engineers with a better understanding of the magnetic field and energy refinement within the Tokomak reactor.

Three dimensional visualization (Figure 13) techniques transfers raw data into intuitive graphical representations which enables the human brain to detect and identify features. Recently research in vector field visualization has pointed out that applying streamlines to reveal magnetic lines within such a Tokomak reactor helps the scientist to understand and combat magnetic islands, which is crucial in understanding the energy particle orbit (Schussmann, et al., 2000). Feature analysis can be brought into the post processing of such simulation data by analyzing the magnetic lines and their behaviour.

5.2.2 Parallelization of the visualization algorithms

A challenge task to visualizing large amount of tokamak magnetic field is the parallelization of the line computations. When computing a large number of streamlines at the same time in parallel, efficient seeding techniques and scheduling becomes

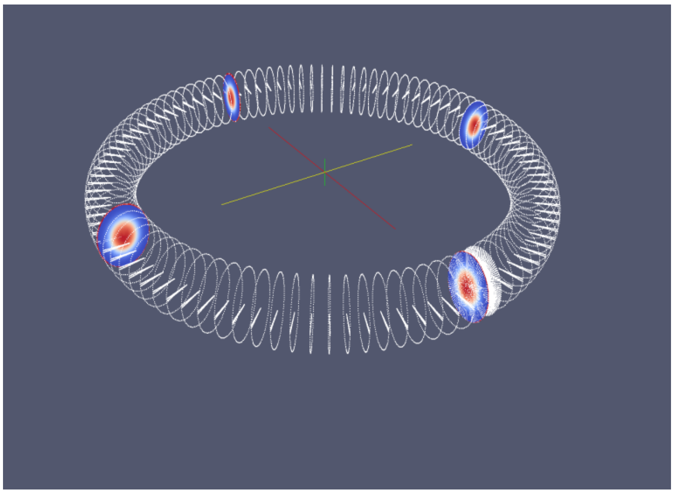


Figure 13 3D visualization of the simulation and geometry

challenging. To apply streamline based visualization methods to a very large vector field, it requires careful balancing of computational demand placed on I/O, memory, communication and processors (Pugmire, et al., 2009).

Going forward, with the co-design activities with ELMFIRE application, we plan to explore and improve parallelization schemes of the magnetic field lines' computation. Our goal is to optimize and leverage parallel resources to achieve scalable and load balanced computations of streamlines of the given simulation output.

6 Conclusion and future work

In this deliverable, we have presented CRESTA's post processing systems, tools developed for exa-scale applications. We have utilized available cluster systems to carry out testing and benchmarking of the proposed system.

Further progress in the co-designed work with the HemeLB application demonstrates system tools that can monitor remote large simulations at run time (the on-online monitoring part) and further allow interactive exploration of the data (front end application). We also presented our recently initiated co-design work with the ELMFIRE application focusing on the optimizing visualization algorithm in order to achieve maximum load balance.

Source code, system requirements as well as demonstration videos can be found at: https://svn.ecdf.ed.ac.uk/repo/ph/cresta/wp5/postprocessing/trunk.

While tacking the current real world problem, especially with the available data sizes and hardware contains, it would be too bold to conclude any scalability guarantee for exascale problem. However, together with the visualization community for large data, we believe that the proposed system architecture and approaches will be a plausible solution and prototype which enables in-situ post processing and minimizes data movements. Online-monitoring, in-situ process and distributed rendering will be key aspects that lead post processing at exa-scale.

Future work within this work package includes developing suitable post processing solutions for large ELMFIRE simulation data in order to achieve scalable and load balanced computations of magnetic field line computation.

7 References

Borkar, S., 2010. *“The exascale challenge.* Taiwan, s.n.

Chen, F. et al., 2012. *Enabling In-situ Pre- and Post-Processing for Exascale Hemodynamic Simulations − A Co-Design Study with the Sparse Geometry Lattice Boltzmann Code HemeLB,.* Salt Lake City, s.n.

Kiviniemi, T., T., S. L., Heikkinen, J. & Janhunen, S., 2012. Gyrokinetic Simulations of the Edge Pedestal in the TEXTOR Tokamak, Contributions to Plasma Physics, 52, 406 (2012).. *Contributions to Plasma Physics,* 52(406).

Ma, K. K., Wang, C., Yu, H. & Tikhonova, A., 2007. In-situ processing and visualization for ultrascale simulations. *Journal of Physics: conference series,* 78(1), pp. 12-43.

Ma, K. L. et al., 2007. *Ultra-scale visualization: Research and education.* s.l.:s.n.

Ma, K.-L.et al., 2009. *Next-Generation Visualization Technologies: Enabling Discoveries at Extreme Scale,* Davis, CA: SciDAC Review.

Pugmire, D. et al., 2009. Scalable Computation of Streamlines on Very Large Datasets. *SC, ACM.*

Schussmann, G., Ma, K.-L., Schissel, D. & Evans, T., 2000. Visualizing DIII-D Tokamak Magnetic Field Lines. *IEEE Visualization,* pp. 501-504.

Tu, T. et al., 2006. *From mesh generation to scientific visualization and end-to-end approach to parrallel supercomputing.* s.l., s.n.