VIRTUAL ENVIRONMENTS FOR SPATIAL DATA INFRASTRUCTURES ON COMPUTING GRIDS

T. Nguyen [a] *, L. Wang [a], J.P. Antikidis [b]

[a] INRIA, 655, Av. de l'Europe, Montbonnot, F-38334 Saint-Ismier, France - (tnguyen, lwang)@inrialpes.fr
[b] CNES, 18, Av. Edouard Belin, F-31401 Toulouse Cedex 9, France – antikidis@cst.cnes.fr

KEYWORDS: Spatial Data Processing, Virtual services infrastructures, Grid and cluster computing

ABSTRACT:

Spatial Data Processing and Infrastructures are the core of future remote sensing and photogrammetry for the Earth Observation industry. For this to become an operational and production reality, quantum leap breakthroughs are to be achieved, concerning in particular image processing, error correlations, alert definitions, best usage practices, data encryption and code validation. Because problems that are expected to be orders of magnitude larger than current single discipline applications, like weather forcasting, are likely to be addressed, e.g., environmental disaster prevention and emergency management, new computing technologies are required. Among these technologies are wide area grids and distributed computing, as well as cluster and grid-based environments. It is clear that large PC-clusters and wide area grids are currently used for demanding numerical applications, e.g., nuclear and environmental simulation. It is not so clear however which approaches are currently the best for developing Spatial Data Processing and Infrastructures. A first approach takes existing grid-based computing environments and deploys, tests and analyzes Spatial Data Processing applications. A second approach executes legacy Spatial Data Processing and Infrastructures codes to characterize grid-based environments for adequate architectural hardware and software adequacy.

We advocate in this paper the use of a grid-based infrastructure that is designed for a seamless approach by the users, i.e., the Spatial Data Processing and Infrastructures designers, although it relies on a sophisticated computing environments based on computing grids, i.e., wide-area computing grids, connecting heterogeneous computing resources: mainframes, PC-clusters and workstations running multidisciplinary codes and utility software, e.g., visualization tools.

1. INTRODUCTION

HEAVEN is a European scientific consortium including industrial partners from the aerospace, telecommunication and software industries, as well as academic research institutes. The goal is to define, develop and provide test-beds for emerging applications and business for the forthcoming Information Society and to explore new usage of computing technologies in the economy and industry for the next decades. The approach is based on concepts defined by the HEAVEN consortium.

Currently, the HEAVEN consortium works on a project that aims to create advanced services integration platforms. They must supporting complex applications in research, science, business and community services. Along the line of the strategic objective "GRID-based systems for complex problem solving", it is an R&D project in the field of "Enabling Application Technologies" based on GRID infrastructures. It is intended to enable "*virtual private grids*" supporting various hardware and software configurations for users using a suitable high-level description language. This will become the basis for future generic services allowing the integration of services without the need to deploy specific grid infrastructures. This approach will permit the development of new business models, e.g., spatial data processing charged on the process added-value and not on the data origin or volume.

The users can define their own "virtual" computing environments by selecting the appropriate computing resources required or reuse and compose existing virtual environments (Ruth, 2005). The approach is generic by allowing various application domains to benefit from potential hardware and software resources located on remote computing facilities in a simple and intuitive way. Basically, the user interface provides an icon-based request facility that allows defining dynamically the virtual ad-hoc computing environments best-suited to particular applications involving Spatial Data Processing and Infrastructures.

These applications may require particular image data (time and location constrained, process and acquisition constrained, etc) and have particular post-processing requirements (e.g., multiple data source and correlations for multidisciplinary analysis in environmental applications). Because emerging applications will presumably handle data sets orders of magnitude larger than current ones, as well as pull new unforeseen applications, it is not possible to design from scratch a new type of environments nor the service and data integration that will be required in the next decades. A prerequisite of upcoming SDI environment is therefore to support open design and scaling. Further, security and authentication support will be mandatory. For historical and technical reasons, these subjects are currently being addressed by the grid computing community. They implement however technical approaches and the state-of-the-art makes it difficult for the casual end-users to get the fluent expertise needed for using grids on a daily basis.

The goal of the HEAVEN project is precisely to overcome this technology barrier by filling the gap between the existing grids middleware and the application designers, and ultimately the end-users.

The computing resources are defined by services available presenting sets of standardized interfaces and performing

---

[a] INRIA, Institut National de Recherche en Informatique et Automatique is the French National Institute for Research in Computer Science and Automatic Control.

[b] CNES, Centre National d'Etudes Spatiales is the French National Space Agency.

specific tasks: application workflow, tasks and data synchronization, input data streams, output visualization tools, monitoring facilities, etc (S3, 2005). Services can be composed and hierarchically defined. Transparent access to heterogeneous hardware and software operating systems is also guaranteed.

2. VIRTUAL ENVIRONMENTS

2.1 Emulation vs. simulation

There are currently several projects aiming at providing users and developers with virtual computing environments. There exist two complementary and dual approaches, depending on the way existing computing resources are used.

One approach is to emulate complex computing infrastructures on ad-hoc software. The second approach is to simulate simple environments running on complex infrastructures.

The first approach tends to virtualize complex environments running on simpler infrastructures: XEN [Barham, 2003] and User-Mode Linux UML [UML] are examples of such projects. Also, VMware, a “virtual infrastructure software”, is a commercial product in this class [VMware].

Another example is the XEN virtual machine monitor which “uses virtualization to present the illusion of (running) many smaller virtual machines, each running a separate operating system instance” (Fugure 1). This is referenced as the “emulated virtualization” in (XEN White Paper) and dubbed “para-virtualization” in (Barham;, 2003). To some extent the Linux-VServer [Linvserv] private virtual servers that focus on isolation and security for private user spaces is a similar approach. This is what we simply call here the *emulation* approach.

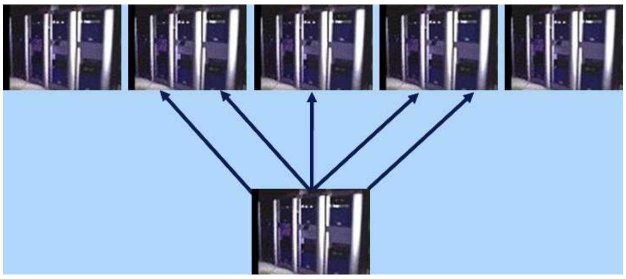


Figure 1. The virtualization approach.

The second approach tends to simplify the users view of complex environments: OpenSSI [Walker, 1999], Vgrid (vgrid, 2003) and Kerrighed [Kerri] are such examples. They provide single systems images (SSI) to simulate single computing environments running on a set of underlying systems which are connected together (Figure 2). This is what we call here a *simulation* approach.

Emulation lends itself nicely to secure and multiple isolated instances of (possibly heterogeneous) systems running concurrently on the same underlying infrastructure. It provides complex environments suited to the application needs, at the price of possibly lower performance. But theoretically, any complex system can be designed using this emulation approach.

Simulation in contrast does not provide superior functionalities with respect to the underlying infrastructure. Its main goal is to mask the complexity of the underlying environments. It is basically made of multiple instances of (Linux) operating systems and computing resources (files, servers, etc) and provides a single interface to the users. It thus simplifies for the end-user access, logging, and automating execution, load balancing, failure recovery (by component substitution) and so forth. Simulation here provides superior functionalities and simpler interfaces to the users.

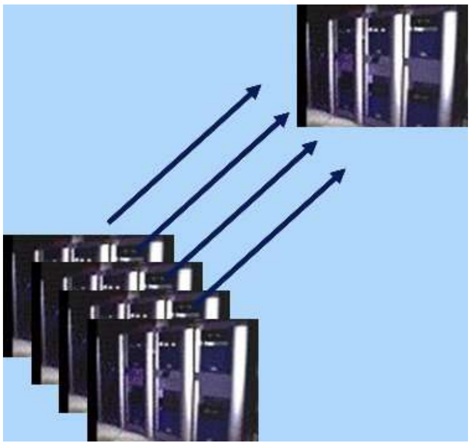


Figure 2. The Single System Image.

Both approaches can be considered as virtualization approaches, although they are very different in their goals and deployment, because in both cases, the users are ultimately made unaware of the underlying computing infrastructure.

A direct benefit from both approaches is that various tasks can be automated (load-balancing, task relocation), made transparent (remote access to files). Further, the virtualization approach improves interoperability by using dedicated application environments, usability, security (task isolation, file protection) and performance (dynamic allocation of processors to threads).

Another side-effect is that the underlying hardware and software environments are masked to the users. Consequently, various (heterogeneous) computing resources can be used, and their location is ultimately unknown. This clearly improves extensibility and scalability by masking the underlying infrastructures as well as adaptability (infrastructure changes are made transparent to the applications).

Access to resources connected to a local high-speed network is a de facto goal for simulation environments, which clearly aim the cluster-computing arena (using for example cluster-wide file access, TCI/IP, single cluster-wide naming, etc).

Access to local or wide-area grids can be seamlessly hooked to the simulated environments because dedicated computers can be connected which are in charge of the communication with the networks.

This is where the computing grids step in. Note that they are not strictly required in our approach. However, the fact that they have long been advertised as providers of huge raw computing power cannot be ignored. They provide here the power to *emulate* the necessary infrastructures required by the complex environments supporting the applications being designed: spatial data infrastructures.

In these environments, the hardware, software and sensor devices are defined by virtual constructs. The application services can likely be designed using virtual constructs which are in fine implemented by generic functionalities run by the underling infrastructures.

Because of the duality of the various goals associated with emulation and simulation, we have adopted the first approach to design and implement virtual computing environments in the HEAVEN consortium project. The main benefits are:

- to provide complex application services deployed on a generic infrastructure
- to be hardware and software independent
- to be platform independent

- to be grid infrastructures independent
- to isolate various topologies of virtual machines from one another

In the latter case, it is possible to deploy and test various software and hardware configurations before their production use. It is also possible to design specific services for the automatic tuning and scaling of infrastructure environments, depending on the applications and services being deployed.

These options make the HEAVEN virtual environments open, scalable and generic. Their design allows for legacy software to run unchanged, therefore fostering user acceptance. They allow new complex applications to be designed involving powerful and heterogeneous, distributed resources. They can be tailored to each particular applications needs without any consideration for resource ownership. They foster new business models where resources can be charged on the added application-value and not on the resource consumption. They also foster new business paradigms where resources can be outsourced to computer resources brokers. The applications can therefore be deployed and run without the users ever owning the computing resources needed by the applications nor the data being processed.

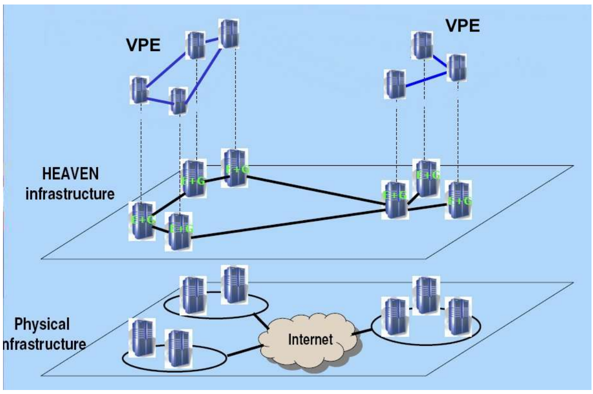


Figure 3. The HEAVEN infrastructure.

2.2 Virtual environments

By large, complex and production environments dedicated to the deployment, monitoring and execution of distributed and multidiscipline applications remain to be seen. High-performance scientific computing has taken the lead in world-wide grid computing for two decades. But the price to pay however is the expertise level required when deploying and running grid middleware, e.g., Globus, UNICORE.

When compared to the Internet, grids are still in their stone age for their ease of use.

A major concern is therefore the seamlessness of such environments. Because new challenging applications are also foreseen in such wide-area grid environments, like disaster prevention, risk and crisis management, a huge simplification of complex computing environments is mandatory. This is what we call “breaking the wall”, i.e., breaking the complexity and technology barrier hampering the widespread use of state-of-the-art technologies for societal and environmental benefits (Kotzinos, 2004).

It is therefore of utmost interest to consider the virtualization approaches described above (Section 2.1).

Clearly, upcoming applications require sophisticated computing infrastructures, which are distributed, parallel, multidiscipline, heterogeneous and they are used by management teams that are not aware of the intricacies of computer technology.

The deployment environments for these new applications require therefore the complex infrastructures handled by the simulation approach above, together with the complex application environments handled by the emulation approach above.

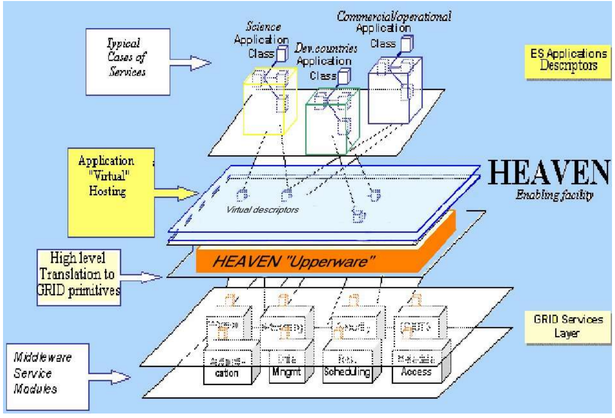


Figure 4. The HEAVEN approach.

The future of virtualization, and consequently of virtual environments, lies therefore in the conjunction and cross-fertilization of both the emulation and simulation approaches.

On the one hand, emulation as defined above (Section 2.1) allows building complex environments out of simpler infrastructures. They allow for example the design of various concurrent dynamic and non-overlapping environments on simpler infrastructures. This is fundamental for the design of secure and dynamic services tailored to specific and complex applications.

On the other hand, simulation as defined above (Section 2.1) emphasizes the use of simplified interfaces to complex systems, which is fundamental for the usability and acceptance of complex technology infrastructures. In particular, the single user view of heterogeneous, distributed and multidiscipline computing systems and resources is similar to the required interfaces to modern spatial data infrastructures.

In the HEAVEN approach, the applications are deployed on Virtual Machine Topologies which are instances of the virtual applications environments (Figure 3). Virtual Machines Topologies are instantiations of concurrent and possibly overlapping networks of Virtual Machines. Virtual Machines are instances of abstract hardware and software configurations which are defined by the application designers to comply with the applications requirements. They include processors, hard drives, memory and bandwidth characteristics, sensors, and comply with specific QoS and SLA requirements. Although different, non distributed and non grid, but oriented to highly configurable operating systems, the Virtual Virtual Machine concept used to execute specific virtual machines which were specialized instances of a generic one. It had some similarities with this approach (Folliot, 2000).

2.3 Underware, middleware, upperware

Virtual environments are tools and facilities dedicated to the design, deployment execution, monitoring and maintenance of large applications on distributed resources. These resources may be computers, file archives, sensors, visualization environments, etc. The users do not need to own any one of them. He or she may have access to and use any combination of them among a set of available resources whenever he or she is granted the appropriate rights to do so, using a simple laptop or sophisticated apparatus, e.g., an immersive visualization environment

He does not need any technical knowledge of the underlying software and hardware tools, except that one he or she is currently using. The technical infrastructure, may it be a state-of-the-art middleware for grid computing or a large cluster of commodity PC connected through a high-speed fiber-optics network is made totally transparent to him/her.

In order to implement this approach, we need a software layer masking the underlying infrastructure. Because hardware, operating systems and i/o devices are sometimes referred to as *underware* (Walker, 1999), and because *middleware* is the de facto naming for grid management and interface software, we name this new layer the *upperware*.

The upperware is the generic service layer used to virtualize the resources used by the applications. It masks the actual hardware and software resources, making possible the design, management and concurrent use of dynamic, possibly overlapping and cooperating sets of private computing infrastructures (Figures 3). In this respect, the upperware enables secure virtual private computing environments to co-exist, in a way similar to virtual private networks designed to co-exist on real communication networks: they are called here Virtual Machines Topologies (VMT).

The upperware is built on top of existing grid middleware. It is therefore a requisite that is made compatible with current and upcoming grid technology standards (OGSA, WSRF, GT4).

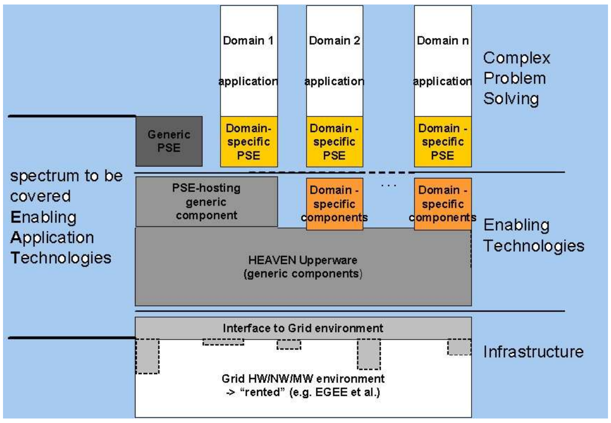


Figure 5. The HEAVEN architecture

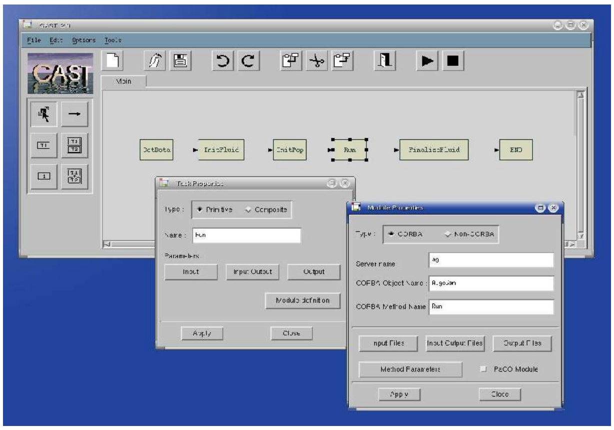
3. UPPERWARE DESIGN

From the user point-of-view, the interface with the virtual application environment is a high-level graphic interface that masks the resource distribution and technical definitions. It is a set of dependent tasks connected by a workflow graph (Figure 6). This approach leaves all the technical aspects to a further step, while focusing on the application logic only. The tasks can be connected by a control flow graph formed by sequence, parallel, interleaved and imbedded loops.

The tasks correspond to executable codes that are located transparently for the users on remote sites. It is the responsibility of the application designers to define which resources the application needs, where they should be located if required, and which complementary properties they should exhibit (availability, QoS, etc). None of these resources are required to be local and to belong to the users and designers. Brokering protocols and usage grants are therefore supported by the upperware. Submission of such grants can be negotiated on a permanent or one shot policy. The upperware appears therefore as a general resource broker, negotiating with the remote systems the availability and use of resources, based on the local policies and granted access rights.

The HEAVEN upperware is a software layer that is based on existing grid infrastructures, e.g., EGEE, RENATER, etc. As such, it interfaces both the user communities through the high-level graphic interfaces described above, and the underlying computing environments. It fills the gap between them and the application problem-solving environments (Figure 4, 5). It includes generic components for interface with grids (invocation and negotiation with remote resource brokers, authentication and authorization, grants negotiations, etc). It also supports specific components dedicated to particular application requirements (interfaces with sensor management systems, with visualization tools, etc). Finally, it is the basis on which the particular application domains solve problems.

There are several ways to implement the upperware, for example relying on a generic Web services implementation [12] and the corresponding Web Services Reference Framework defined by GGF [13]. Another option is the CORBA component-based architecture [14]. There are even full Java implementations of grid-aware middleware [20]. The first option is preferable since it guarantees the compatibility with the OGSA architecture [7] and the hopefully soon available Globus GT4 [8, 9, 10]. Further, compatibility with forthcoming versions of other middleware such as UNICORE [20], which are now interoperable with Globus [11], will be supported. There is however no guarantee that a backward compatibility with previous versions of Globus Toolkits (GT2 and GT3) will even be supported by GGF [6]. Therefore, this is not a priority concern for HEAVEN.



Type : ● Primitive ○ Composite

Name : Run

Type : ● CORBA ○ Non-CORBA

Server name ag

CORBA Object Name : AlgoGen

CORBA Method Name Run

☐ PaCO Module

Figure 6. The user interface.

4. UPPERWARE DEPLOYMENT

The obvious advantages of the virtual application enviroments are their ability to mask the technical aspects of grid technology to the application designers and users. The example depicted by Figure 8 is an aerodynamics optimization application running on three remote PC-clusters located in different locations at INRIA centres and connected by a high-speed gigabits/sec network (Figure 7). The end-users never interact directly with the underlying middleware and network. The application designers have to define the abstract tasks involved, the corresponding executable codes (by their name and access paths) and the resulting data files (by their names and access paths also). An example application design using the CAST software is given by Figure 6.

Our testbed is built on a computing grid involving two remote research centres: INRIA Rhône-Alpes in Grenoble and INRIA Sophia-Antipolis, near Nice, in France. It includes several Linux workstations and three high-performance PC-clusters (Figure 7). The computing resources are described in Table 1.

Concerning hardware, INRIA Rhône-Alpes and INRIA Sophia-Antipolis are connected to a high speed network provided by the VTHD ("Vraiment Très Haut Débit") project. The VTHD network is the support platform for the French initiative for the New Generation Internet backed by the RNRT (Réseau National pour la Recherche en Télécommunications). The VTHD network is based on *2.5* Gbps links based on the VDM network of France Telecom production network.

Concerning software, the UNICORE middleware is used for the management of the grid infrastructure [UNICORE]. The application clients, running the CAST software, are located on the Linux workstations, which are connected by a 100Mbps intranet network. A dedicated workstation is used as a gateway for the clusters and the VTHD network.

The UNICORE servers are running on both the clusters and the gateway. The UNICORE client runs on the Linux workstations, as well as the CAST application clients. A detailed analysis of the testbed and test applications, as well as the performance evaluation of the testcases is given in (Nguyen, 2005b).

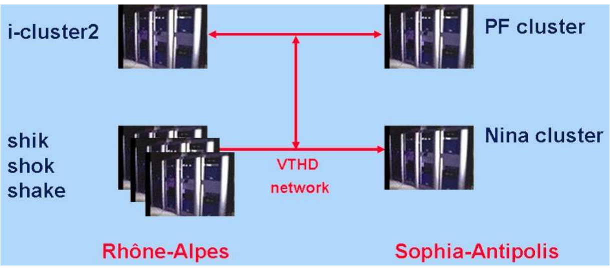


Figure 7. The testbed architecture.

5. CONCLUSION

The exponential growth of distributed and cluster computing on wide-area grids is a challenge for computer scientists and all the communities of users today. If grid-computing is to break the casual-users barrier, like the Internet did ten years ago, many challenges remain to be addressed. One of the fuzziest and creeping challenges is the ease of use and best-practice standards for grids. There are still no clear tools and methodologies answering these questions today. Ease of use will clearly convince reluctant user communities from the scientific, industry and business arena to adopt this promising technology. One approach is to devise new interfaces to grids that will help the users to abstract their applications from the technicalities of the underlying and ever-growing technologies supporting the computerized world.

This paper presents a new paradigm based on the full virtualization of resources involved in the applications. It abstracts all the resources involved in a technology independent upperware. This is a software layer that builds on existing grid middleware, taking benefit from the Web Services technology to build transparently standard abstractions masking the underlying grid infrastructures. We call them "Virtual application environments" because there is no need to own any of the resources involved. Consequently, it paves the way for new business models. It is in no way another grid middleware. It is instead a software layer masking the intricacies and technical details that no user community can today fully understand, deploy and maintain without the help of dedicated teams of computer science experts.

The generic, open and scalable HEAVEN upperware is currently being tested on a variety of demanding applications, including multi-physics code validation, multi-objectives optimization in aeronautics (Figure 8) and multidiscipline environmental monitoring, etc.

Resource	Hardware & OS	Software
PF @ Sophia Antipolis	Cluster: 19 nodes, 100Mbps Fast-Ethernet, 1 Node: 2 Pentium III @ 933Mhz, Linux Kernel 2.4.2 & LSF	UNICORE server & CAST
NINA @ Sophia Antipolis	Cluster: 16 nodes, 3 Gigabit-Ethernet, 1 Node: 2 Xeon @ 2Ghz Linux Kernel 2.4.2 & LSF	
i-cluster2 @ Rhône Alpes	Cluster: 100 nodes, 1 Gigabit-Ethernet, 1 Node: 2 Itanium @ 900 MHz (64 bits), Linux Red Hat Advanced Server 3.0	UNICORE server
Shok	Workstation: 1 Pentium III @ 1 Ghz, 100Mbps Fast-Ethernet, Linux Fedora Core 2	CAST, UNICORE client & server
Shik		
Shake		

Table 1: Computing resources for the testbed.

The testbed uses a grid that includes several heterogeneous PC-clusters and workstations connected to a high-speed network. Future work includes its deployment on the nation-wide Grid5000 (Grid5000) network and the combination of both the emulation and simulation approaches (Figure 9).

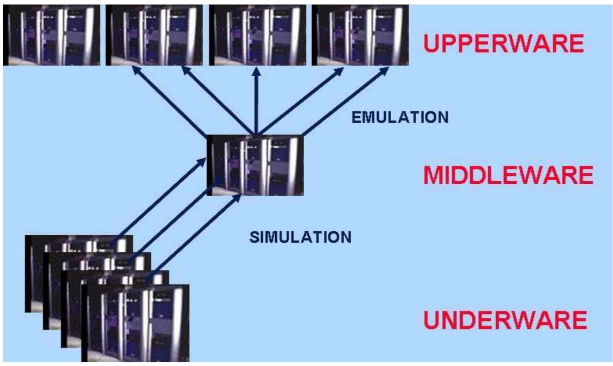


Figure 9. The future of upperware: combining simulation and emulation.

ACKNOWLEDGEMENTS

HEAVEN ("Hosting European Applications in Virtual Environments") is a consortium involving:

- CNES, French Space Agency, Program and Strategy Directorate, « Space Information Systems » (F),
- LogicaCMG (NL),
- DATAMAT (I),
- Econet (Hu),
- INRIA, project OPALE (F),
- Beijing University of Aeronautics and Astronautics (China),
- Fraunhofer IAO (G),
- European Aeronautic Defence and Space Company (EADS), Corporate Research Centre (F),
- SciSys (GB),
- IACM, FORTH (Gr),

- University of Cyprus, Department of Computer Science, Nicosia (Cy),
- Institut Pierre-Simon Laplace, CNRS-CEA-IRD-CNES (F),
- Institut de Physique du Globe de Paris, CNRS (F)

and other candidate organizations and industries.

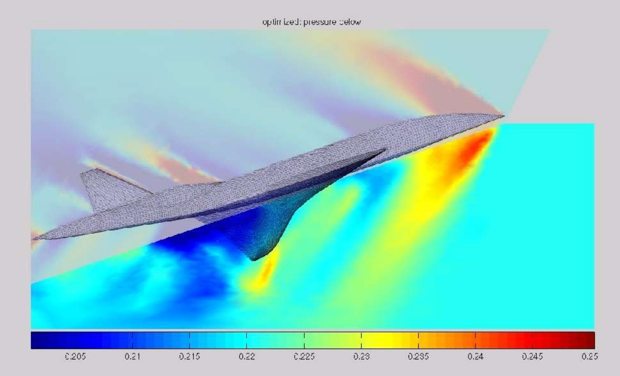


Figure 8. Project OPALE: supersonic wing optimization (Janka, 2004).

REFERENCES

Barham P., et al. Xen and the Art of Virtualization. ACM 19th Symposium on Operating Systems Principles. Lake George (NY). USA. October 2003.

Cactus Project, http://www.cs.arizona.edu/cactus/

E. Christensen et al. “The Web Services Definition Language 1.1”,W3CNote,March2001. http://www.w3.org/TR/2001/NOTE-wsdl-20010315

Condor Project, http://www.cs.wisc.edu/condor/

K. Czajkowski et al., “The WS Resource Framework”, The Folliot P. The Virtual Virtual Machine project. Simposio Brasileiro de Arquitetura de Computadores e Processamento de Alto Desempenho. SBAC 2000.

Globus Alliance, March 2004. http://www.globus.org/wsrf/specs/ws-wsrf.pdf

Globus Alliance, “Migrating from GT2 to GT4”, http://wwwunix.globus.org/toolkit/docs/development/4.0-drafts/index.html

Grid Interoperability Project, http://www.grid-interoperability .org/

Globus Alliance, www.globus.org

Grid5000, www.grid5000.org

Janka A, Andreoli M. Desideri J.A. Free form deformation for multilevel 3D parallel optimization in aerodynamics. International Parallel CFD Conference. Universita de Las Palmas. Gran Canaria (Spain). May 2004.

Kerrighed, kerrighed.org

Kotzinos D., Chrysoulakis N. GIS Web Services and satellite imaging as a framework for environmental monitoring: the design and implementation of a vegetation indices calculation testcase. Proc. 20th International Society for Photogrammetry and Remote Sensing Congress “Geo-imagery bridging continents”. Istanbul (TK). July 2004. http://www.isprs.org/istanbul2004/index.html

Legion Worldwide Virtual Computer, http://www.cs.virginia.edu/ ~legion

Linvserv, http://linux-vserver.org/

Nguyen G.T., L. Wang, J.P. Antikidis. Virtual environments for multiphysics code validation on computing grids. International East West High Speed Flow Field Conference (EWHSFF 2005). Beijing (China). October 2005.

Nguyen G.T., L. Wang, J.P. Antikidis. Virtual application environments on computing grids. Third International Conference on Computing, Communications and Control Technologies (CCCT’05), Austin, Texas (USA). July 2005.

Nguyen G.T., L. Wang. Implementation and performance evaluation of an airfoil shape optimization application on computational grids. Paper submitted for publication. May 2005.

Nguyen G.T., J.P. Antikidis. Virtual computing environments for problem-solving on grids. International Parallel CFD Conference. Universita de Las Palmas. Gran Canaria (Spain). May 2004.

Nguyen G.T. Grid-computing in Multidisciplinary CFD optimization problems. International Parallel CFD Conference. Invited lecture. Russian Academy of Sciences. Institute of Mathematical Modeling. Moscow (Russia). May 2003.

Nguyen G.T. Integration of multidiscipline applications in grid-computing environments. PARA02 Conference on Applied Parallel Computing. Center for Scientific Computing. Helsinki (Finland). June 2002.

Nguyen G.T. An Integration platform for metacomputing applications. International Conference on Computational Science ICCS 2002. Amsterdam (NL). April 2002.

Object Management Group, The Corba Component Model. http://www.omg.org/technology/documents/formal/components.htm

Opale project: http://www.inria.fr/recherche/equipes/opale.en.html

Ruth P., Jiang X., Goasguen S. Virtual distributed environments in a shared infrastructure. IEEE Computer. May 2005.

S3 Initiative: a European vision for Software, Services and Systems. Version 31-01-05. European Commission. IST FP6. January 2005.

UNICORE project, http://www.unicore.org/

UML, www.uml.org

Vgrid: The Vienna grid project. Http://vgrid.sourceforge.net. 2003.

Vmware: http://www.vmware.com

VTHD project. http://www.vthd.org/

Walker B., Steel D., Implementing a full Single System Image UnixWare Cluster: Middleware vs. Underware. Proc. Intl Conf. on Parallel and Distributed Processing Techniques and Applications. Las Vegas (Nv). USA. June 1999.

Xensource. Xen – the Art of Virtualization. A Xensource White Paper. http://www.xensource.com/files/xensource_wp.pdf