NEW CONCEPTS FOR DATA SERVICES OFFERED BY "VIRTUAL IMPLEMENTATION"

Jean Pierre Antikidis

CNES Programme & Strategy Directorate-ARP, "Space Information Systems"
jean-pierre.antikidis@cnes.fr

ABSTRACT:

The technological evolution of major human activities has been continuously oriented toward higher and higher level of "abstraction". As an example Blue-prints for creation of planes or buildings is fully replaced by virtual computer aided design, electronic circuitry mock-ups is now first tested with the help of SPICE simulators, point to point computer access have been replaced by Internet virtual connectivity. A new evolution is now underway that pave the road to promising new concepts called "Virtual Information Management" in which computers systems will be replaced by their virtual computer equivalents with no need to know anymore where the information stay and what computer produce it. The advent of GRID technology is a first step in this direction but as for Internet and the IP protocol it is only the first stone of a more larger (but ought to be much easier to manage) environment combining to-day present telecommunications progress as well as advanced computer and much higher level operating systems (some call them "UpperWares").Such "dematerialization" of computers replaced by invisible virtual systems embedded into a general GRID like networks could constitute the next quantum leap in human perception of electronic world interactions and the way to create space data ground infrastructures.

1. INTRODUCTION

The technological evolution of major human activities has been continuously oriented toward higher and higher level of "abstraction". As an example Blue-prints for creation of planes or building is fully replaced by virtual conceptual design, electronic circuitry mock-ups is now first designed with the help of SPICE simulators, point to point computer access have been replaced by Internet virtual connectivity.

A new evolution is now underway that pave the road to promising new concepts called "Virtual Information Management" in which computers systems will be replaced by virtual equivalents with no need to know anymore where the information stay and what computer produced it. The advent of GRID technology is a first step in this direction but as for Internet and the IP protocol it is only the first stone of a more larger (but ought to be much easier to manage) environment combining to-day present telecommunications progress as well as advanced computer and much higher level operating systems (some call them "UpperWares").

Internet is a precursor of such a new technical and societal change that could lead soon to a full "dematerialization" of the treatment of the information enabling even small entities (commercial or privates) to create even large scale services by manipulating "virtual definitions of their needed systems and archives"

2. VIRTUAL SYSTEMS AND THEIR RELATION WITH GRIDS SYSTEMS

2.1 The Advent of GRID Technology

In fact a GRID system is basically a set of decentralised systems being "organised " with the help of a suitable Operating system taking care of any materiel induced constraints. The natural evolution of such "MiddleWare" will encompass not only "processing activities" but also the storage of the information. Such improved and much capable access will free us from the practical handling of the stored information. This could allow unprecedented new possibilities in term of flexibility and easy creation of data process profitable to: research centres, added values companies, support to developing countries or emerging European economies. It should also trigger interest in and a much wider use of space-borne data by small industrial entities able to create and rapidly test access to applications (at much lower costs) , thus inducing profitable services.

Methodology trends

From Real =>	to « Virtual »
Blueprint plans (1850)	CATIA (1985) (virtual mockup)
Formal programming (1948)	WINDOWS (1990)
MINITEL Physical links(1970)	INTERNET (1990)
Electronic circuitry (1914)	SPICE simulators(1985)
Processing systems (1960)	GRID+VIRTUAL "Upperware" (2002)

Figure 1. Computing trends

The advent of GRID is opening the door to completely new concepts in term of computerised system implementation. This concept introduced in the recent years is quickly maturing and could be simply described as a way to make an heterogeneous sets of machines connected together able to work in a simplified way trough the use of an appropriate Operating system called MiddleWare. The novelty compared with already existing distributed systems know since long comes from the fact a GRID user is not obliged to deal with each machine and organise their connectivity but may submit its work by just "speaking" with MiddleWare primitives.

This principle is already of big step forward in term of "Virtualisation" since applications are not anymore (in theory)) dependant of the underlying Hardware. One may even "patch" with the help of MiddleWare primitives complex applications.

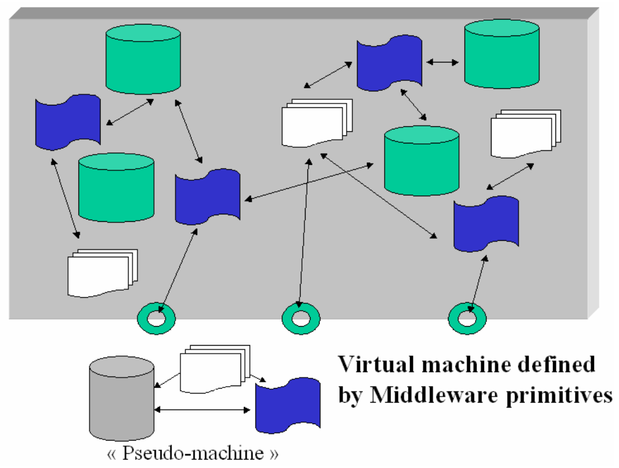


Figure 2. Virtual GRID machine

As usual when a new technological step forward occurs, ideas are first concentrating on the straightforward use of the new tool. This is the to-day situation with GRID mostly considered as way to pool computer resources in a kind of decentralised fashion.

Therefore after a while new advanced ideas are emerging in which GRID is to be considered as a way to create with great flexibility virtual computing structures in an invisible and easy way. This can be considered as the next step of a conceptual evolution in which computerised systems will become fully virtual. It could be achieved theoretically without any GRID but in practice GRID is the most elegant and efficient way to provide the necessary features.

Up to now GRID is seen from the developer side as a set of primitives called "GRID MiddleWare primitives". The standards are continuously evolving, but the solutions proposed for the use of GRIDs can be grouped into several classes of solutions, say:

- **Direct application plug onto a GRID**

This case is usually applicable to specific cases where the needed resources are of primary concern. This justifies the development of specific interface software ("Application tailored"). For obvious reasons it becomes expensive and inflexible for unspecialised uses or quick developments needs.

- **High level compilers or language translators**

This promising approach illustrated for example by the Model Driven Architecture (MDA) , the Data Flow Architecture or some of the new upcoming language translator enable the designer to work on his application by making use of high level primitives that are subsequently transcoded and mapped onto a given set of Computer Resources. These approaches are smart and power-full but requires special skills and a new rethinking of the applications

- **Building blocks & Thematic interfaces trough Web services**

This case is a normal evolution by providing a higher level of services. The MiddleWare primitives are used in the frame of an integrated interface making use of Web based services. This allows several advantages:

  - The user sees the system through a Web interface
  - Submission of jobs to the GRID is also performed with the help of a Web based interface
  - Complex applications can be created by merging several Web services defined by their semantic usage

- **High Level Operating System compounds (UpperWares)**

This is an emerging concept; it has been referred to in several prospective presentations concerning computers and distributed services evolution.

The idea is to make use of higher level descriptors and leave a specific operating system to take care of all necessary actions connected to the handling of MiddleWare primitives. Such an upper layer could be called "UpperWare". Advantages for the developers are obvious but it is also a great challenge to develop.

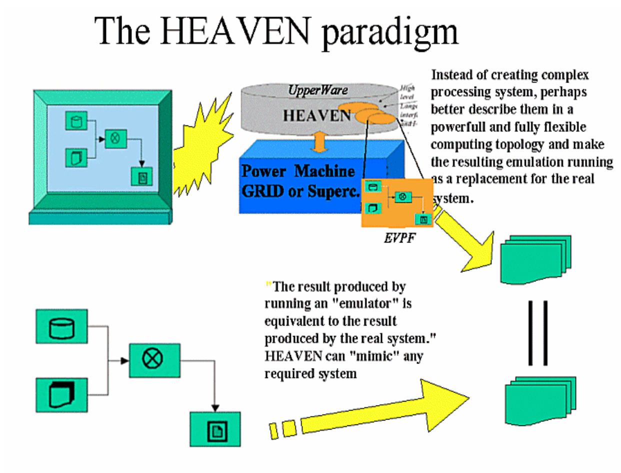


Figure 3. HEAVEN paradigm

- **UpperWare by Emulation of the needed resources (HEAVEN)**

Among the many solutions possible for UpperWare definitions a promising one has been introduced in view of offering to the developer the most easiest possible definition interface (CAD like) in association with the possibility to work with unchanged environments (HW/SW definitions)

An example is illustrated by the HEAVEN project (Hosting of Emulated Applications in a Virtual Environment) that proposes to develop an UpperWare able to handle following functions:

a) Design a complete hardware and software topology (Virtual Private Facility) with the help of CAD like graphical interface (can be in theory as easy as designing an aircraft or an electronic circuit on a computer).
b) Emulate the needed HW/SW environment (Emulated Virtual Private Facility)

c) Make use of this emulation as a working replacement for the real facilities.

2.2 Software Services Implementation with the Help of GRID Based Virtual Private Facilities:

Several authors have commented on the advantages brought by a full system virtualisation and HEAVEN capitalise on these issues by providing an easy to run and generalised implementation based on the "Emulation Packet" (EP) concept. As starting consideration Software Service is a set of Hardware, Software and user interfaces connected in such a way a given end-user service is provided. To make it available we need practically:

- To decide on the needed HW arrangement
- To decide on a suitable SW running on this HW
- To decide on machines interconnections and interfaces with final users

The picture on Figure 4 shows two typical examples of machines connected together and creating an end user service (In this case a meteorological information system and a Satellite data processing service). This arrangement can be called a "Private Facility".

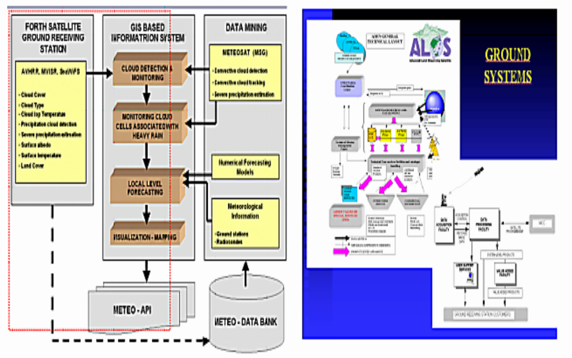


Figure 4. Example of Private facilities

The possibility to "Emulate" Private facilities instead of using "real one" is inducing several advantages by:

- Allowing system designer to create their Virtual Private Facilities without having to touch any HW enabling dynamic system implementation and on the fly reactive and intelligent/self adaptive systems (since just a matter of managing definition macros)
- The need to run legacy applications is served well by virtual emulated machines. A legacy application might simply not run on newer hardware and/or operating systems but very easy to maintain by running in an unmodified virtual machine.
- Emulated machines can be used to provide secure, isolated sandboxes for running un-trusted applications. You could even create such an execution environment. Virtualisation is an important concept in building secure computing networked applications.
- Virtual emulated machines provide the illusion of hardware, or hardware configuration that you do not have. This property is used by HEAVEN to create complex HW software combinations including file or/and external real interfaces.
- Virtual machines can be used to run multiple operating systems simultaneously: different versions, or even entirely different systems, which can be on hot standby. Some such systems may be hard or impossible to run on newer real hardware.
- Virtual machines are great tools for effective and quick service developments. Since they provide isolation, they are safer to work with. They encapsulate the entire state of a running system and allow large design mistakes to be easily corrected at little cost.

Several studies have been recently initiated concerning "Virtual system implementation" illustrated in two complementary concepts called "HEAVEN" and "WAG". No doubt future large scale Earth Observation environmental system will take advantages of these types of approaches.

3. HEAVEN CONCEPT (HOSTING OF EMULATED APPLICATIONS IN VIRTUAL ENVIRONMENTS)

The installations depicted in fig 4 can be installed on a real fashion (Private Facilities) or rely on an exact virtual equivalent (Virtual Private facilities).

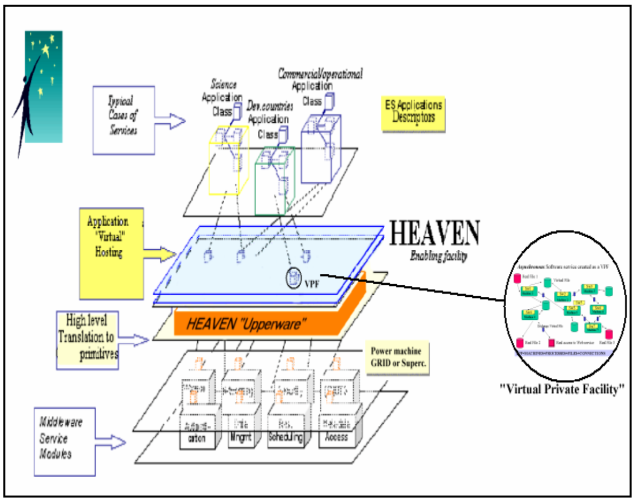


Figure 5. Virtual Private Facilities created by HEAVEN

In its traditional sense, virtualization provides multiple execution environments (virtual machines), each of which is identical to the underlying computer. Each virtual machine looks like a "real" machine to its user. By connecting together virtual machine running their software and connected to their needed file environment, we create a kind "Private facility" that is taking care of a given service. The choice is to rely on the properties induced by running Emulators as an operational substitute to real machine, thus enabling a full compatibility (intellectual and practical) with software developers' habits while bringing an enormous conceptual flexibility.

Figure 6 shows an Asynchronous Software Service created as a VPF. Theses individual machines are connected either by using intermediate storages (like very often happen in real developments) or in case of "direct machine to machine connections by using intermediate "communication virtual stores". (There is no difference inbetween two machines speaking directly or two machines exchanging information with the help of a virtual intermediate step). By accepting this principle any topology of machines can be described as a network of files (real or virtual) communicating with machines. For a given machine, the set of files being used and their description can be stored as a XML description.

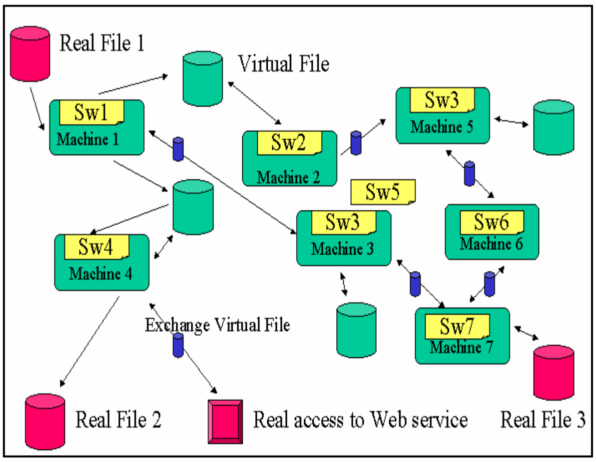


Figure 6.

Implementation in a conventional multi-tasking machine or a super-computer (host machine) is then easily achieved by creating one independent task per emulated machines (ref fig 6 &7). The number of concurrent tasks running in parallel (on a non synchronous fashion) is equivalent to the number of "emulated machines" while the file description associated to a given machine is determining a virtual topology of connectivity in between the different machines. By this way one may come to a very simplified situation in which the host machine is just running emulation tasks connected to predefined sets of files.

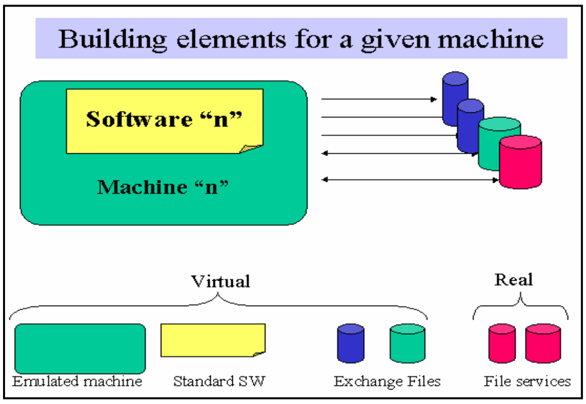


Figure 7. Virtual machine building elements

Another interesting simplification come from the fact any of the emulators is just "seeing" files and does not need to be instructed on the other emulator's situations. This principle allow the entire task to become fully autonomous, thus greatly helping an easy multitasking implementation .By extending this concept it becomes possible to create a new software entity that could be called an "EP" for "Emulation Packet".

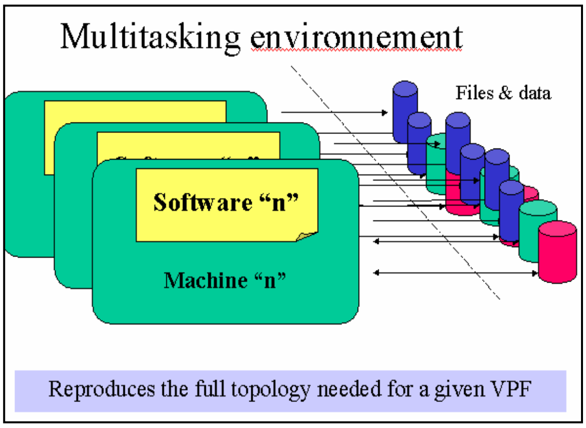


Figure 8. Virtual machines created in a Multitasking environment

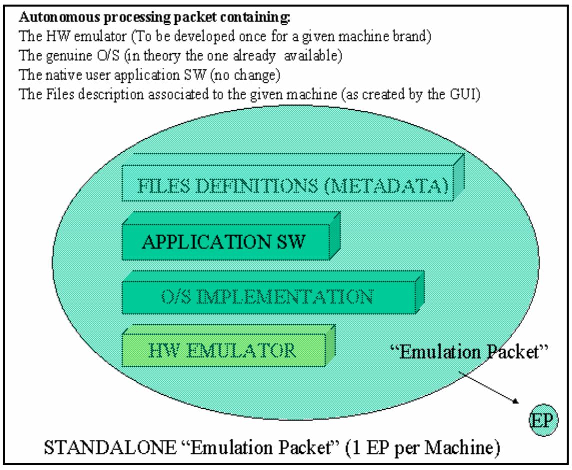


Figure 9. Emulation Packet definition

This new "Packet" somewhat comparable to an "IP" packet (but much larger) is defined as a complete and self contained piece of code containing:

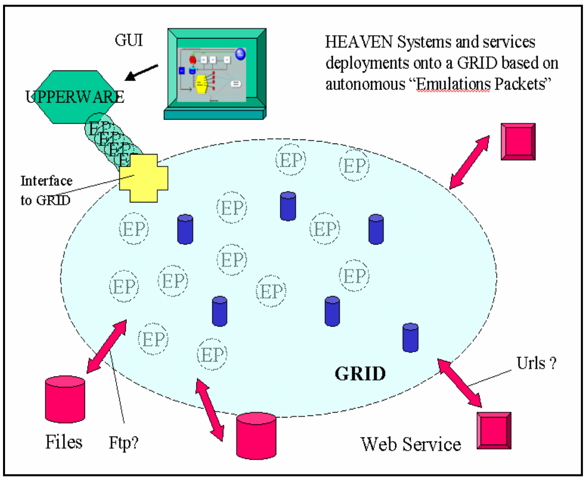
-The HW emulator (To be developed once for a given machine brand)

-The genuine O/S (in theory the one already available)

- The native user application SW (no change)

For example, in a GRID system the HEAVEN UpperWare will be in charge of creating these EPs ("Emulation Packet") and launch them as fully independent tasks within a GRID. Since we accept the whole thing is kept asynchronous, each of the EPs will run as an independent machine and the whole of the launched EPs will reproduce the needed "virtual topology" we need.

By introducing the EP concept one come to a very striking consequence that is a kind of equivalent functioning compared to the Internet Packet (IP). As for the IP, you can launch as many "packets" as you like. Each of these EP is fully independent but knows how to behave without having to refer to anything else than a set of files (might even be called ftp files).



Of course an EP will be much larger (several tens of Mbytes) but the "Packet" principle is kept unchanged. One may then conceive the simplification brought by EP running in an autonomous way inside a GRID !

EPs versus IPs

The efficiency of Internet principle rely on the IP concept by which a data packet once submitted to the system becomes autonomous (no time nor routes relations in between the packets) This allow Internet to handle on an unvisible fashion Billions of packets at a time without need for strong centralised means.

Same stand with HEAVEN "EP" concept (these are much bigger packets of course !) in which Emulation packets once generated becomes fully autonomous with respect to their location or timing. This allows the full compatibility of millions of EPs at a time corresponding to perhaps several thousand of "Private facilities" that look from the user viewpoint as a pseudo HW/SW service implementation under his full control.

4. THE ADVENT OF WIDE AREA GRIDS (WAG) IN ASSOCIATION WITH VIRTUAL UPPERWARES

Up to now the GRID definition has not been much precise, but several classes can be distinguished with:

- Local GRIDs based arrays of computers (1500 at CERN for example)
- Large GRIDS associating several centres (eg: GRID 5000 project)

- Very large GRIDS extending their coverage on a World-wide basis : The Wide Areas GRIDs

By extending the "Virtualisation" technologies here-above described with a large GRID coverage we come to the WAG concept in which data and processing localisation is not anymore relevant. For Earth Observation application this is a very promising tool that may drastically simplify satellites operators interconnections and users services effectiveness. Let's illustrate in with a simple example called "Virtual Storage management"

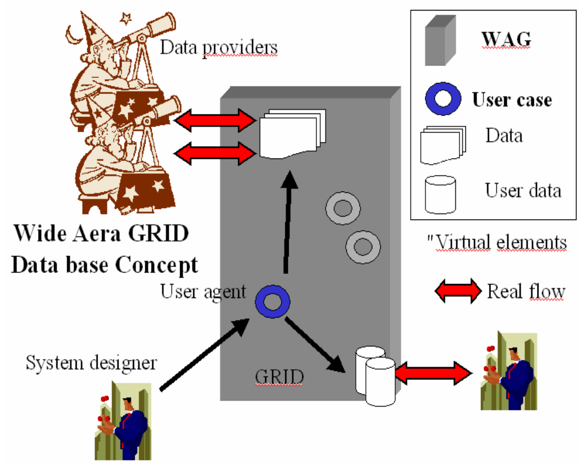


Figure 10. An example of data bases virtual data management

In a conventional data base concept, the activities are logically organised as follow:
- Locate the information (catalogues etc..)
- Get hold of the data (media, access costs, transmission & delays)
- Process the information
- Transfer the result to suitable place

In a WAG data based concept it should lead to :

- Create a Grid connection with the potential databases
- Make their archive accessible from this GRID (without being obliged to know in advanced what will be needed)
- Enable the creation of jobs submitted to the GRID

In fact there is no need anymore to exactly locate where the data stands or whether they are or not dispersed.

Once a database is "seen" from the surrounding GRID it is "seen" from any user application jobs and the handling of communications and data repositioning should be handled by the "UpperWares" itself.

5. PRELIMINARY CONCLUSIONS

Virtual Data management is a logical consequence of GRID concept evolution but by no mean a GRID intrinsic property that brings:

- High computer power provided on a standardised fashion
- Capacity of to create dynamically computer topologies at will
- Automatic handling of communication resources if a distributed system is to be created

In this case emphasis could be to propose not only solutions for final users but new advanced concept that may help final users to take THEMSELVE advantage of the coming GRID (in fact without knowing what is a GRID). This is to some extend the equivalent of the INTERNET concept that never produced final users needs direct answer, but allowed millions of users to create their own systems at a speed of days instead of month with previous technologies. The advent of full virtualisation of computerised system may create a novel gold-mine for computer system designer who may create at low cost even enormous application dispersed all over the word by just "painting" computers and "data bases" as virtual objects embedded on an invisible fashion on a European and why not on the long run World Wide GRID. In short after the Internet and the GRID advent some (r)evolution may come based on a new "vision" based on "virtual systems concepts".

Of course large-scale database and computerised facilities will still be needed and present archiving and processing systems will stay. But their insertion (or immersion) in the frame of GRID large scale topology (WAG) with the help of suitable virtualisation as allowed by advanced operating systems for GRIDS (UpperWares) will induce a drastic change in the way to "think" computerised system deployment. No surprise Europe including ESA and EU will become key players in association with other major European space systems suppliers like CNES. As for GRID and Internet, the coming technological evolution will be based on a natural selection process where success will be mainly driven by users addiction. The advent of "fully virtual" processing and storage systems in which design may become as straightforward as writing a diagram on a piece of paper is to be encouraged since one of the key to lower cost and expand drastically the number of services offered to users. This can be considered as the coming Internet-like evolution for process management (as was Internet for information management). Needless to say most of this evolution will be "power-hungry "in term of needed resources for CPUs, storage and communications, inducing in turn an other level of sizing for most of our archives and computerised facilities (but on a pooled GRID organised fashion).