Space Operations Communicator
Vol. 5, No. 4, October – December 2008

Use of Virtualization Techniques for Ground Data Systems

Juan Prieto[1]
IsFreelance, Madrid,28029,Spain

José Feiteirinha[2] and Pedro Bizarro[3]
CISUC/DEI, University of Coimbra, 3030-290 , Portugal

Eduardo Gómez[4] and Mauro Pecchioli[5]
European Space Agency, Darmstadt, 64404, Germany

As the number of satellites in operation increases, space agencies are faced with larger and more complicated computer infrastructures to support the operations. This paper reviews a number of solutions currently applied by many enterprises to similar problems and shows how they can be tailored to the space operations domain.

Nomenclature

VM	=	Virtual machine
LEOP	=	Launch and early orbit phase
COTS	=	Commercial off-the-shelf (for software products)
OS	=	Operating system
Host OS	=	OS running in a physical machine
Guest OS	=	OS running in a virtual machine
SAN	=	Storage area network

I. Introduction

The infrastructure used by the European Space Agency to support space operations has evolved across the years from centrally managed mainframes to networks of workstations. In such networks, a small number of server workstations (typically more powerful than client workstations) do the centralised processing while a large number of client workstations perform the local processing. This approach is fully consistent with the IT trends over the last few years: processing power is relatively cheap and can be increased by replacing workstations while bandwidth is relatively expensive and can only be increased by replacing complex cabling infrastructures. By using this approach, the information exchanged between client and servers can be tailored to the final user demands achieving a reduction in bandwidth.

While the approach presented above has been successfully applied in many missions, it is beginning to raise concerns because it results in a large number of underutilized machines deployed (this problem is often known as the “server sprawl” problem). The main reasons for having such a large number of underutilized machines are:

- High availability requirements. To achieve higher availability, the failure rate likelihood is decreased by having each workstation isolated from other systems and fully dedicated to one particular purpose.
- Reducing mean time to restore. Additional hardware spare systems are left in standby to reduce mean time to

[1] Freelance consultant, IsFreelance, c/Arzobispo Morcillo 62-28029 Madrid, Spain.

[2] Researcher, Universidade de Coimbra, Dep de Eng InformáticaPolo II, 3030-290 Coimbra, Portugal.

[3] Assistant Professor, Universidade de Coimbra, Dep de Eng InformáticaPolo II, 3030-290 Coimbra, Portugal.

[4] Software Engineer, Ground Systems Engineering, Robert-Bosch-Str. 5-64293 Darmstadt, Germany.

[5] Head of the Operation Centre System Infrastructure Section, Ground Systems Engineering, Robert Bosch Str. 5-64293 Darmstadt, Germany.

restore and further increase availability, leading to a further increase in the number of workstations.

- High performance requirements. Hardware is pessimistically over-allocated to guarantee good performance even in worst case scenario usage.
- Long lifetime of ground data systems. Several missions last for many years (more than a decade in some cases). The system must stay operational during this period. New missions, however, rely on new hardware. New and old systems must therefore coexist in common areas and computer rooms.

In addition to this, the fast pace of evolution of hardware demands frequent upgrades to the latest version of an operating system (as older versions will often not run on new hardware). This means that either existing applications are migrated to the new baseline or the old hardware infrastructure is maintained alongside the new one. This leads to yet another increase in the number of machines.

This large number of machines running different baselines imposes severe requirements on e.g. space, cooling systems and energy consumption and causes a significant increase in maintenance costs.

These problems are by no means specific to the space domain. Most enterprises have experienced similar scenarios in the last few years and a number of solutions are appearing in the market. They are normally referred to as consolidation techniques as they attempt to consolidate the resources.

Consolidation is the generic term for describing the approaches to achieve an efficient usage of computer resources in order to reduce the total number of servers or server locations that an organization requires. Although consolidation could be designed at application level, this is quite complicated in mission critical scenarios. This would imply:

- Coordination in application design so that they can run together on the same platform (i.e. hardware, operating system, 3rd party products).
- Detailed testing of applications running together to ensure that they do not interfere under any circumstance.

Consolidation by virtualization represents a better alternative. Virtualization is one of the most promising current technologies to achieve consolidation. Virtualization is applied at the stage of the physical consolidation (i.e. replace a number of smaller servers with a larger server). It consists in enabling the possibility to run different virtual machines in parallel on the same physical hosts, each one appearing to higher level layers as completely independent computers, running different operating systems without any interaction with other virtual machines.

Virtualization can help to reduce the number of machines in several ways

1) Virtual machines act as containers for applications and ensure that they are executed without interfering with each other. Many applications and/or operating systems can therefore share a machine transparently, thus enabling a better hardware utilisation.

2) Virtual machines acts as an isolation layer between old operating systems and new hardware, ensuring that legacy applications, which only run on specific versions of the operating system, can be executed in the latest hardware along with more recent applications.

3) Virtual machines can be deployed faster than real machines as the software can be preconfigured. Applications can be pre-packed and distributed to the relevant node.

II. Requirements for the virtualization of operations

The main goal of the activity described in this paper is to determine how virtualization technologies can be used to address the problems of ground data systems described in the previous section. In particular, the work has focused on how virtualization can help the realization of the three following main objectives:

- Reduce the amount of different hardware platforms to be maintained in both the development and operational environments.
- Reduce the level of dependency of the software systems from the underlying physical hardware, so upgrades in the hardware resources and native operating systems do not lead to the adaptation of the application software.
- Simplify product packaging, installation and deployment

The main requirements of a virtualized environment are summarized in Table 1 below

Table 1. Main requirements of the virtualization activity

Requirement
The deployment of virtualization shall allow the sharing of hardware resources between different applications, potentially running under different operating systems.
The deployment of virtualization shall allow the sharing of hardware resources between different missions and teams (i.e. heterogeneous groups of people with little connection between them).
The deployment of virtualization shall allow the scalability of the physical and virtual hardware resources, so more resources can be allocated for a certain application or system when required.
The deployment of virtualization shall make the software applications independent from upgrades in the physical hardware.
Virtual machines sharing the same hardware shall not be affected by any problem caused by a different virtual machine.
The performance impact of running the systems within virtual machines shall be identified and minimised.
The new concept shall not imply significant changes in the architecture or implementation of existing applications.

In addition to the above requirements the deployment of a virtualized environment has to be consistent with existing policies and procedures, in particular in connection with critical satellite operations such as launches or critical manoeuvres.

The key issue to consider is that most applications are used for performing mission critical operations, which has a number of implications in the way the system is used and managed. These issues directly affect the consolidation approach, to ensure that the new concept maintains the same level of reliability.

Although operations are important in all mission phases, there are some periods in which the activity is especially critical, which corresponds to the LEOP phase, and in general the critical actions and manoeuvres throughout the mission lifetime. In these periods, the requirements for the system reliability are particularly strong.

A key idea about operations is that it has to be ensured that the system will respond as expected at the time it is required. For general applications it is acceptable to ensure an average availability of the resources, but not for operations. For example, processing data off-line may need to ensure that the time required to complete a task is adequate but it does not matter if sometimes the system works at peak performance and at some other times, due to a lack of available resources, it does no work at peak performance. For operations in real-time, it is mandatory to ensure that the resources will be available at the time they are needed. This makes the sharing of resources between applications more complex, because it is not enough to calculate average resources required, but also to establish mechanisms to ensure that peak of resources are also covered.

In addition to this, a distinction needs to be made between client and server machines. Servers run critical applications that are used system wide, while client run local processing and can be replaced in a relatively easy way. Therefore the reliability requirements to be placed on servers are far more demanding than those on clients. This needs to be taken into account when designing the architecture.

The system has to be predictable, so no random behaviour may happen. Users need a full confidence on the behaviour of the system.

The performance of the system has to be ensured in any case, so it is necessary to reserve and allocate the resources in a way the systems can use them when necessary, considering the different contingencies that may occur. The consolidation approach is intended to maintain this, but optimising the resources compared to the current situation of dedicated resources.

The specific reliability requirements considered for the deployment of virtualization are summarized in Table 2

Table 2. Specific reliability requirements for the virtualized environment

Requirement
The new concept shall ensure that the resources required for operations are available for a system whenever required, considering peaks of activity and possible contingencies.
The new concept shall be predictable, so the behaviour of the systems is known in any situation and no non-deterministic behaviour is possible.
The new concept shall isolate the different systems sharing resources so no failure in a system can affect the behaviour of the rest.
The new concept shall provide a fault tolerant environment, where redundant resources are available in case of failure of both servers and clients.
There shall be no single point of failure.
The new concept shall be compatible with existing redundancy mechanism implemented, including the warm and hot redundancy.
The new concept shall minimise the probability of a failure of several operational clients at the same time.

III. Virtualization technology

The virtualization concept has existed since the 1960’s, but virtualization of mainstream platforms (such as x86 or x86-64) has only recently become common with the appearance of Virtual PC in 1997 and VMware in 1998.

Currently there are a few well established vendors as well as a few known open source projects offering virtualization services. However, the market is far from being stabilized. Recently both Intel and AMD launched hardware support for virtualization making it easier for new software vendors to enter the market. The biggest advantage of having hardware support for virtualization is increased performance and reliability of the system.

Most virtualization systems fully emulate the hardware peripherals. This means, that a virtual machine has access to a pool of virtual devices. Virtual device drivers can be ported to new hardware allowing the VMs to be insensitive to hardware evolution.

There are different approaches to virtualization; the most common ones are described below.

A. Emulation

An emulator is an application that allows an application, created for a specific computer architecture, to run, unmodified, in another computer architecture. That is, the emulator allows one system to reproduce the execution results of running one program as if that program was being run on another system. However, emulation is slow because emulators have to perform many operations many operations that would otherwise be executed by hardware.

On the other hand, emulation is the only technique available when the emulated architecture is different from the host's architecture. While other approaches are significantly faster, they only work when both the guest and host architectures are the same.

Not all emulators use the same techniques. Some use software interpretation, others dynamic translation or just-in-time compilation (JIT).

B. Full and Native Virtualization

Full virtualization is conceptually similar to emulation, in that it provides a complete abstraction of the underlying hardware, allowing the execution of an unmodified guest. Since full virtualization is designed to run applications compiled for the same architecture as the host system, much of the code is executed directly on the host's hardware, without any kind of intermediate translation.

However, there are some special operations, such as I/O instructions, that need to be trapped and simulated in order to avoid affecting the state of other virtual machines. Those instructions, which incur in a noticeable performance overhead, are handled by a special piece of software called the hypervisor.

Native virtualization reduces the performance overhead of executing those special instructions by using hardware support for virtualization. That is, some new processors have special instructions designed just to speed up the execution of those special instructions.

Native virtualization has gained strong support recently in the x86 processor families. Both Intel and AMD recently introduced sets of new instructions that extend the x86 architecture. Those new instruction sets are called Intel Virtualization Technology (or simply Intel VT, sometimes also referred by its code name “Vanderpool”) and AMD Virtualization (or simply AMD-V, sometimes also referred by its code name “Pacifica”). Although Intel VT and AMD-V new virtualization instructions are not fully compatible, they both allow the efficient implementation of native virtualization. Most native virtualization products support both sets of instructions.

If extra hardware is added to the physical machine, that new hardware can be made available to the guest operating systems by reconfiguring the VMs. Note that this reconfiguration is not automatic.

C. Paravirtualization

Paravirtualization takes the approach of making special modifications to the guest OS so that it is aware that is executing inside a virtual machine. Then, the modified guest OS, instead of trying to execute those problematic instructions mentioned above, delegates them to a virtual machine monitor, or hypervisor, through a “hypercall”. The hypervisor runs in privileged mode and executes the instructions appropriately.

Note that while in paravirtualization the guest calls the hypervisor directly to execute a special instruction, in full virtualization these instructions need to be intercepted and trapped, which is significantly slower.

Paravirtualized machines typically run very close to their native performance. The main disadvantage of paravirtualization is that the guest OS must be modified, which means that it is not possible to execute a proprietary OS (e.g., Microsoft Windows and Solaris) as a guest OS.

D. Operating System Level Virtualization

This approach is very different from the previous ones in that it only provides separated environments, or isolated “sandboxes” inside the same operating system. This means there is only a single kernel running concurrently, which results in low overhead. However, it is not possible to run different operating systems using this approach only. In addition, since all guests run on top of the same operating system kernel, operating system level virtualization is vulnerable to problems in the kernel. If that single OS kernel dies, all applications are likely to be affected. Note that if a hypervisor dies, all VMs running on top of the hypervisor die as well. However, with OS-level virtualization, the single-point of failure is much bigger (code-wise and complexity wise). That is, it is much more likely for an OS to fail than for a hypervisor to fail. Another problem with this approach is that certain applications cannot be run on those private environments. For instance, a kernel-based NFS server cannot run on such a system. (Note that this is a limitation on the way some servers are implemented. Nevertheless, some services use operating system features that are non-virtualizable.) Only strict user-level services can coexist in parallel on those partitions.

Typical operating system virtualization solutions allow for the isolation of file-system and network devices between each virtual environment and the definition of policies regarding memory limits, disk and CPU quotas, among other useful features.

E. Application Virtualization

Application Virtualization approaches provide sandboxes where applications, their files and settings run isolated from operating systems. The Java Virtual Machine uses the application virtualization approach. Note that application virtualization approaches isolate applications from the host OS.

F. Conclusions

While all of the above approaches have their uses, we are mainly interested in server consolidation of critical services. This restriction excludes emulation (because it is too slow) and operating system virtualization (because it does not completely isolate one system from faults that may occur in another system running in the same hardware). In addition, OS-level virtualization does not provide isolation from hardware and offers bad performance isolation. Both aspects would be among the main requirements of any virtualization solution to be used.

Paravirtualization has very good performance. However, paravirtualization is not appropriate because it requires modified guest operating systems. These modifications could possibly make the paravirtualization solutions more sensitive to hardware evolution. In addition, these modifications to guest operating systems require significant effort from the companies providing paravirtualised solutions. As such, some experts claim that paravirtualization will not survive the test of time.

IV. Proposed architecture

The following diagram (figure 1) provides an overview of the deployment of the new ground data systems architecture based on virtualization from the point of view of the hardware and the systems.

The new deployment is divided into four main areas:

Central management resources, running the systems for the control of the virtual environment. They are the central services, the virtualization management server, and the central management system. They would normally run on physical machines, so they are external to the virtualised environment for increasing the reliability. All central services are deployed in a redundant configuration, with independent repositories synchronised (the software tools and databases will provide the synchronisation capabilities).

Application level resources (in special computer rooms and not accessible to the end user). They correspond to the main hardware platforms running the virtualization products and the applications within the virtual machines. They can be either high-end servers or standard servers, using enterprise level virtualization or workstation virtualization products (heterogeneous environment depending on the resources needed and the overall strategy for scalability). All of them are configured and controlled from the central management systems.

Application level resources (accessible to the user). They can be either simple consoles (just for connecting remotely to the virtual machines running the applications), or actual workstations running a single client virtual machine and the console together.

Other resources, which correspond to the network and other facilities which interact with the consolidated environment.

The central services have a repository containing the master “virtual disks” and the configuration for all virtual machines. Virtual disks are large files inside the physical machine that contain the files system of virtual machines (i.e. a “virtual” file system inside the physical file system). In the proposed architecture there are three virtual disks per virtual machine: one with the virtual operating system and COTS, a second disk with a swap partition and a third one with the applications. This ensures that different areas of responsibility (generic software support, application support, etc.) are correctly handled as each disk can be treated separately. Administrative users can create new virtual machines in selected hardware platforms using the central management system via web, so the applicable virtual disks and virtual machine configuration are setup automatically in the target hardware platform, and the central configuration is updated. Then, the virtual machine can be started from the central management system, and it connects to the central services to get its configuration, to act as a specific instance in the environment.

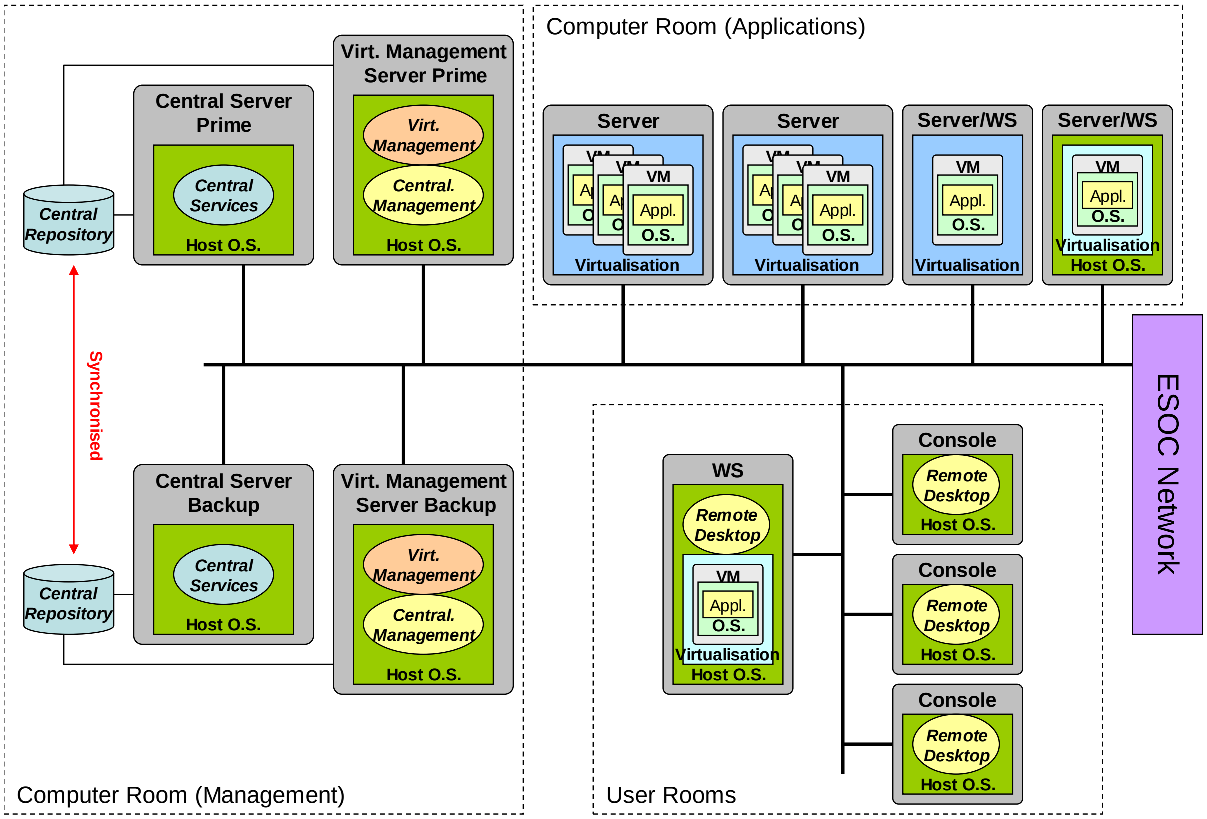

Figure 1. General systems deployment

The computer support team controls the behaviour of the physical and virtual machines using the different monitoring and control tools (at the level of hardware or virtual machines).

The users start the user connection tool from the console machines, in order to select the virtual machine to connect to, and then login in a session for using the applications.

A. Applications Deployment

The proposed new concept does not change significantly the way in which systems are deployed. However, there are some specific points which need to be addressed when considering an environment based on virtual machines sharing the same hardware resources.

The main change introduced in the applications deployment is the concept of generic application “virtual disk”, which contains a “generic” version of the application potentially applicable for any configuration. This application needs to access a configuration service in order to download the specific configuration applicable. The proposed approach is to deploy a separate virtual machine to act as configuration server, to avoid this function becoming dependent on specific servers.

This concept is critical to the proposed architecture as it allows the automated configuration of an application in any virtual machine and makes a potential migration to cope with availability requirements easier.

B. Levels of consolidation

There are different alternatives to be considered for the deployment of virtualization. They are commented in the following paragraphs.

The **scenario 1 (lowest consolidation)** is the simplest approach (figure 2). It would imply a low level of sharing.

In this concept, each hardware platform would be dedicated to a given application, but running on virtual machines to achieve operating system independence. The clients would run on standard workstations in user areas. Clients can be shared by different missions. Shared clients have virtual machines for each relevant mission, but only one is used at a time. The only platform where a higher level of sharing is recommended are external servers, which only serve external requests and do not perform critical operations.

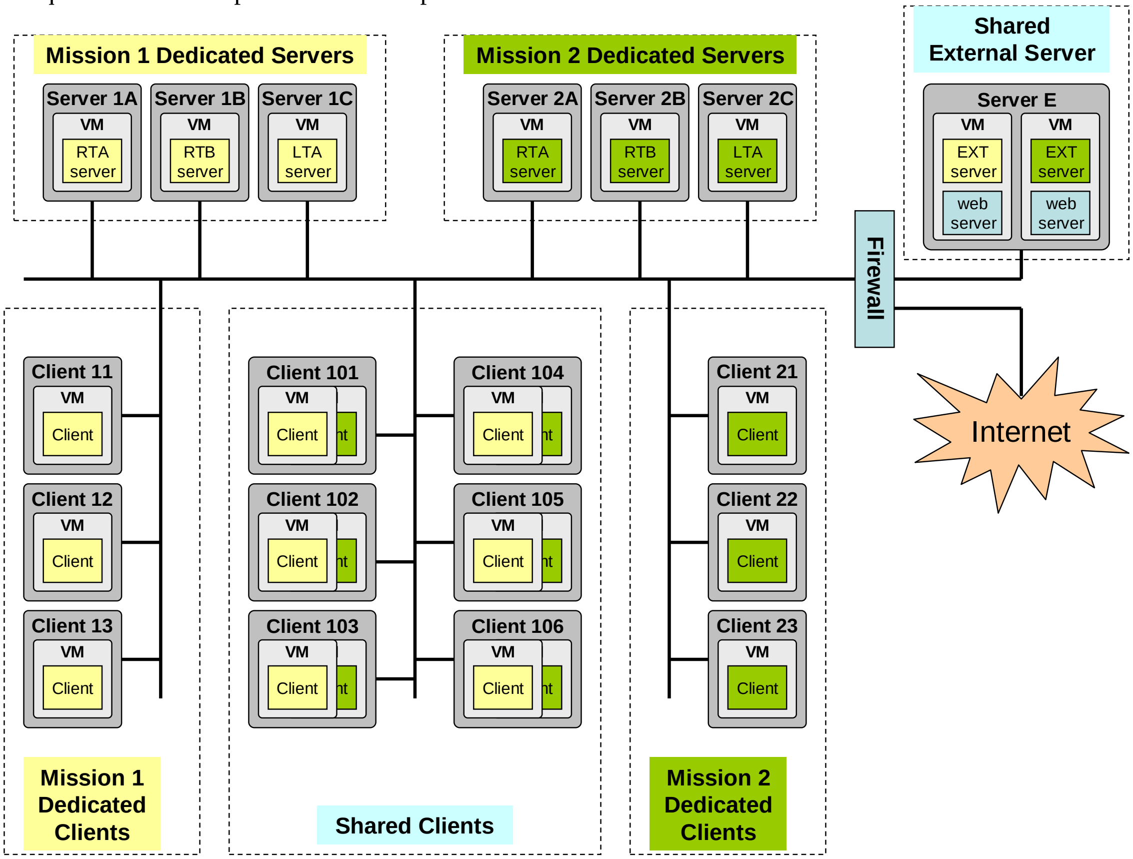

Figure 2. Lowest consolidation scenario (scenario 1)

The **scenario 2 (clients consolidation)** considers the possibility to consolidate several clients into the same hardware platforms (figure 3). It introduces the concept of console machines. In this approach, the servers are kept separated as in the previous case, in order to avoid any kind of resource-sharing problem. In the server side no side effect between missions can occur, so risks at server side are not present. However, consolidating clients introduces the problem of losing several clients with a single hardware failure. To mitigate this, it will be necessary to distribute the client virtual machines in a way that a single failure does not affect many critical clients of the same mission.

Figure 3 shows a possible deployment implementing the proposed concept of client consolidation.

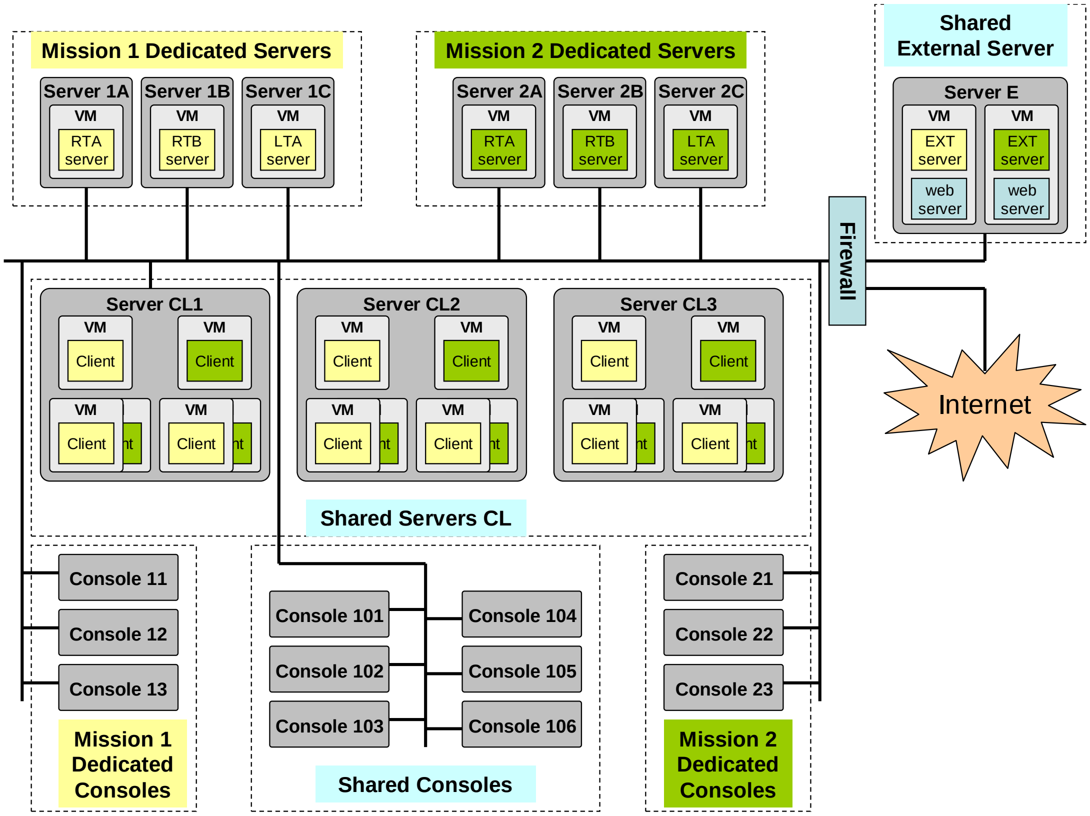

Figure 3. Client consolidation scenario (scenario 2)

As shown in the picture, the 12 client platforms of the scenario 1 are replaced with 3 powerful platforms and 12 simple consoles. In case of failure of a hardware platform, a maximum of 3 clients of the same mission are lost. Considering prime/backup families, the loss is even less important (combining prime and backup clients on the same platform).

The **scenario 3 (full consolidation, figure 4)** considers the possibility to consolidate both clients and servers sharing the resources (combining the approaches of the previous two scenarios). It provides the benefits of the consolidation at all levels, although it is necessary to consider the possible risks of hardware failures, which have to be minimised by a clever distribution of the systems.

As shown in the picture, the 6 server platforms and 12 client platforms of the scenario 1 are replaced with 5 powerful platforms and 12 simple consoles (initially reusing the available workstations).

The previous scenarios provide examples of consolidation approaches with a few virtual machines of two missions represented, to explain the concept. However, **using more powerful hardware, higher levels of consolidation can be achieved** (for example a 32 CPU cores machine would consolidate a lot more systems). However, it is necessary to consider the availability and redundancy concepts, in order not to cause a general problem affecting many systems due to a single hardware failure. A possible way to mitigate this is the use of virtualization high availability solutions as additional possibilities, but implies the use of Storage Area Network (SAN) solutions as described later.

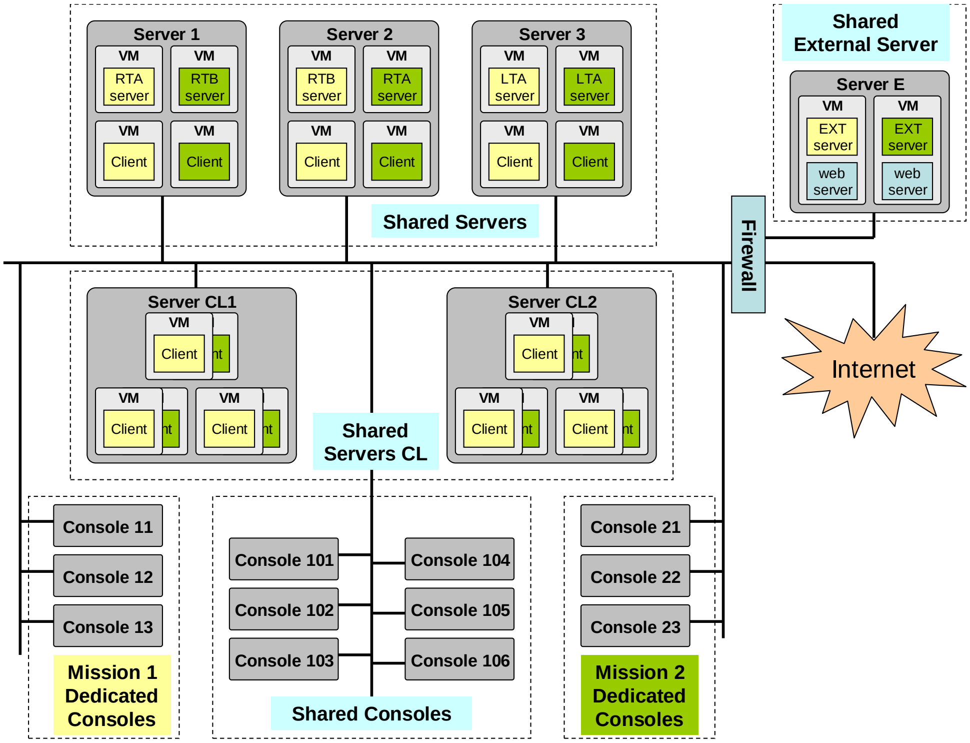

Figure 4. Full consolidation scenario (scenario 3)

C. Use of SAN and advanced virtualization functions

SANs (Storage Area Network) are a way of centralising and consolidating storage resources. They are able to hide from the operating system that the file accesses are not local. The use of virtualization technologies in combination with a SAN storage solution is very common in the IT market, because it provides additional advantages.

In the case a SAN or a similar storage solution is used, the following approach is proposed for the deployment:

The SAN would be split into two replicated areas, each one located physically in a different computer room. A maximum distance of around 10km can be achieved using fibre channels (which SANs use for connecting to the systems).

The two areas would be automatically synchronised using a remote mirroring solution, so the data would be the same in the two storage areas.

The hardware platforms running virtualization would be connected to the SAN of the areas to which they belong.

All virtual disks and virtual machines configuration files would be stored in the SAN and available from the different hardware platforms (although a single instance of each virtual machine can be running at a time).

Figure 5 shows the proposed deployment.

The main advantage of this type of deployment is that the same information is available at all the hardware platforms, which makes the environment more dynamic. The following advanced functions would become possible:

Advanced hardware level redundancy: in case of a failure in any hardware platform, the virtual machines running on it can be immediately restarted in a different hardware platform. As all the information is stored in the SAN disks, the virtual machine can be restarted without losing its data and status stored persistently. The downtime of the systems in case of hardware failure would be reduced significantly. In the case of software failure, the redundancy would be achieved using the applications standard mechanisms, starting an isolated backup system in which the failure was not propagated.

Dynamic allocation of hardware platforms: It would be possible to make the environment far more dynamic, because any virtual machine could be started in any of the platforms connected to the SAN. This would allow a better usage of the resources depending on the current load.

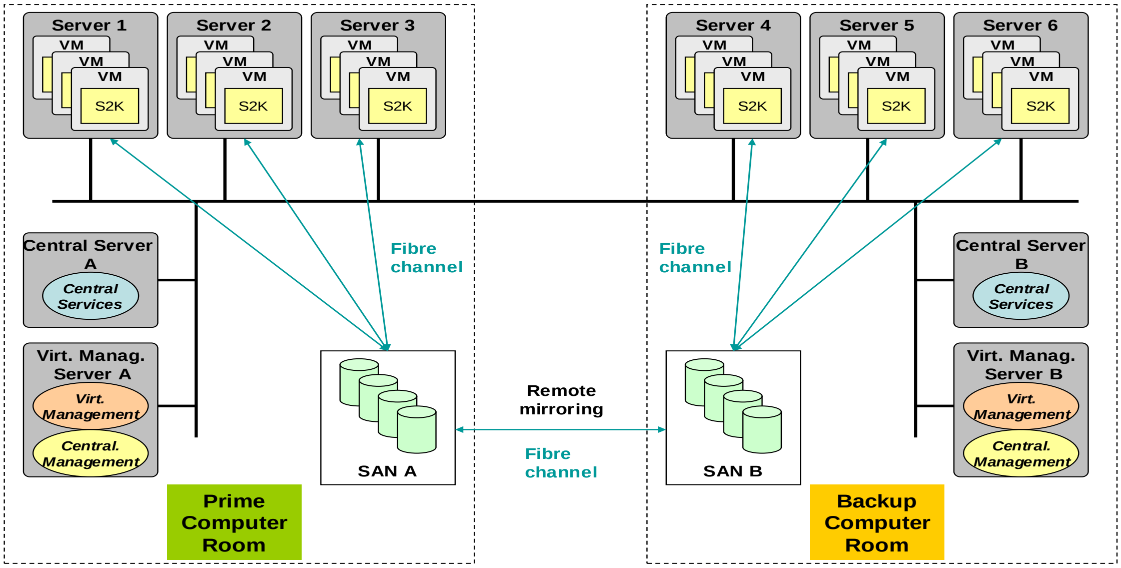

Figure 5. Deployment using a SAN

The use of SANs can extend the benefits of virtualization. Unfortunately, they require rather expensive infrastructure and therefore have not been considered in the architecture and only presented as an option

D. Open Points

There are some open points and issues related to the proposed architecture, which are dependent on the specific environment and tools. They need further analysis before the proposed architecture could be deployed. These points are:

- The certification schemes of the hardware, operating systems (inside virtual machines) and virtualization products are not fully clear. A list of supported platforms and systems is provided by most virtualization vendors, but it is still not clear how the hardware and operating system vendors support their systems when using virtualization technology. This point needs to be negotiated directly with the companies providing support and virtualization technology.
- Data backups: existing backup solution may not work in a virtualized environment. At least backups on physical machines would have to be reviewed to ensure that files representing virtual machines disks (often of large sizes) are not backed up.
- Out-of-band management: The integration of the virtual environment with out of band management tools is not straightforward. Many of these tools used the machine serial ports to ensure availability even in the case of major failures (e.g. network interface down). Most Virtualization packages, however, do not provide access to the serial port.
- Integration with existing network services: virtualization offers different possibilities for networking. They rely on “virtual” network devices and/or sharing the network resources of the physical machine. This needs to be integrated with the company network policies. Network load needs to be reviewed as an increase in the traffic may happen, depending on the specific approach selected and the specific virtual machines running. One final consideration needs to be made regarding networking: depending on the specific virtual networking approach some virtual machines may route their network traffic via the relevant external network router even if they are running on the same physical machine.

E. Deployment for operational systems

The highest level of consolidation can be achieved in the development environment. Potential interferences between applications are not as critical as they would be in operations and availability is not a major concern. While it is possible to virtualize only the development environment, the benefits would be limited. The applications would need to be revalidated without virtualization before transference to the operational systems. The virtualization of the operational environment requires special considerations to be taken into account. Most of these considerations are not issues specific to virtualization, as they also affect the system level deployment in physical environments.

The **overall organisation of the missions** within the operational environment becomes a key point to consider when sharing hardware resources. Two main approaches are possible:

- General sharing, in which the hardware resources are a pool which can be used by any mission (following a predefined agreement for each mission).
- Group sharing, in which missions are organised in groups which procure the necessary hardware to support them. This approach fits with the ESOC concept of the mission families, so there might be resources for Earth observation, interplanetary, etc.

As a general rule, the maximum consolidation is achieved with an overall sharing, so more resources are available for all missions interested. However, the coordination can be simplified in less consolidated environment based on mission families.

The **distribution of the clients** has to be considered. The following ideas have to be considered when selecting how to deploy the clients:

- It is necessary to minimise the probability of having several clients failing at the same time. For this reason, the same hardware platform shall never run several virtual machines for critical clients of the same family within a mission.
- Some clients can share the physical platform where the servers run. This is a good approach, because it reduces the physical network traffic between clients and servers (depending on the networking approach), and the reliability is not too degraded because in the event of a failure of the server platform, the clients are not usable anyway unless switching to another server.

The **redundancy** is another key factor. In its simplest form, redundancy is implemented by using separate machines to perform the same functionality. In such a scenario backup systems must run on a separate physical machine. This simple redundancy concept can be expanded one level using virtualization. The idea is that the virtual machine files can be stored in a disk shared by two different physical platforms, to be initially started in one of the platforms while the other is stopped. In case of failure, it is possible to restart the virtual machine immediately from the other platform, automatically taking the same identity in the network reducing the downtime to a minimum. This would be a very simple procedure when using virtualization, because all the information necessary to reconstruct a virtual machine is stored in the disk.

The last specific point for operational systems is **how to implement the "freeze"** (i.e. the isolation of a specific system), to support a mission under critical operations. Since the architecture proposed relies on common elements which would need to be frozen too, a completely separate environment would be needed. The idea would be that this specific environment is frozen, while the standard central services are still working normally in order to support the rest of missions. To implement this, the virtual machines under the freeze would be configured in a special way to locate the central services in a different place. Changes made in the critical operations environment would be propagated to the general environment immediately, while changes in the general environment would not be applied to the area under the freeze. Using approach the amount of systems under freeze could be minimised, and the operations of the rest of missions are carried out in a more efficient way.

V. Results

A prototype with a set of basic features was developed to evaluate the architecture proposed in this paper. The prototype was demonstrated using virtual machines based on different versions of ESA's mission control system software (SCOS-2000). The demonstration was used to validate the proposed approach and to show the automatic configuration procedure. The system was tested using VMware's virtualization products and the overall results were good. Some features were evaluated using WMware Server which runs on top of the operating system. Specific tests related to fault isolation, performance or reliability were carried out using WMware ESX (which runs directly on top of the physical machine). For comparison purposes, some experiments use other virtualization products (mainly Virtual Iron). The results of the testing are summarized in the following sections.

A. Performance degradation of the virtual machines

These tests aim at analysing the overhead introduced by the virtualization layer. They execute the same operation with 1 virtual machine (VM), two VMs, four VMs and eight VMs. The time spent on the execution of the operation is measured. The machine used had a dual core CPU. The ideal results would be:

- The time spent in executing the operation should increase linearly with the number of virtual machines.
- The overhead introduced by virtualization should be as small as possible.

(Note: the time used by the physical machine without the virtualization layer is shown for reference purposes with the label “host)

The following areas were benchmarked

- Use of CPU (figure 6): the results were positive with only a 10% overhead introduced by virtualization. The trend was linear. (note: as the CPU had two cores the time spent with two virtual machines is almost the same as the time spent with one). VMware showed better results than the alternative technology (Virtual Iron);
- Use of the network (figure 7): good results too and almost non-existent overhead for VMware;
- Access to memory (figure 8): Again good results for VMware;
- Access to disk (figure 9): This was the only performance test considered unsuccessful and has lead to new recommendations for the architecture. In principle, disk intensive applications such as databases or archives should not be virtualised. Further analysis may change this recommendation. The adoption of SANs in particular may offer a major performance improvement. VMware behaved better than the alternative product in the sense that the time spent grew linearly with the number of virtual machines. It is, however, significantly larger than the time spent by the machine without virtualization.

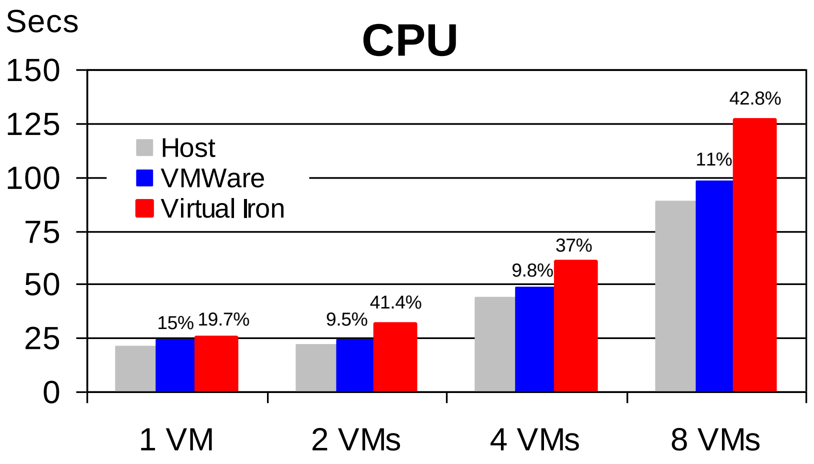

Figure 6. Performance degradation in CPU operations

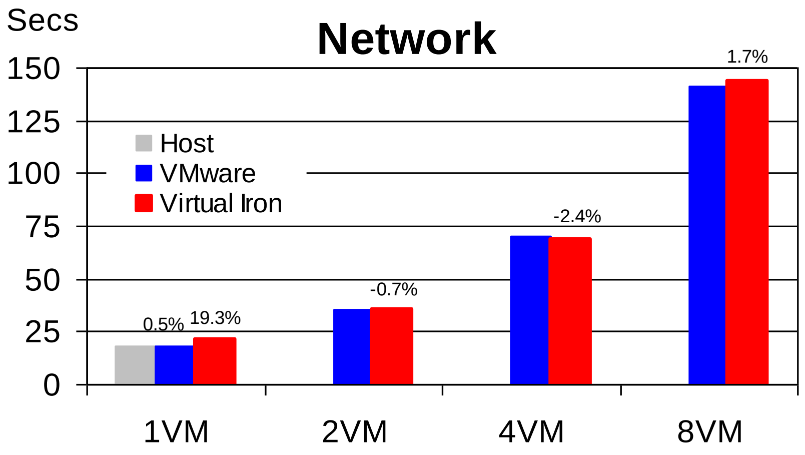

Figure 7. Performance degradation in networking operations

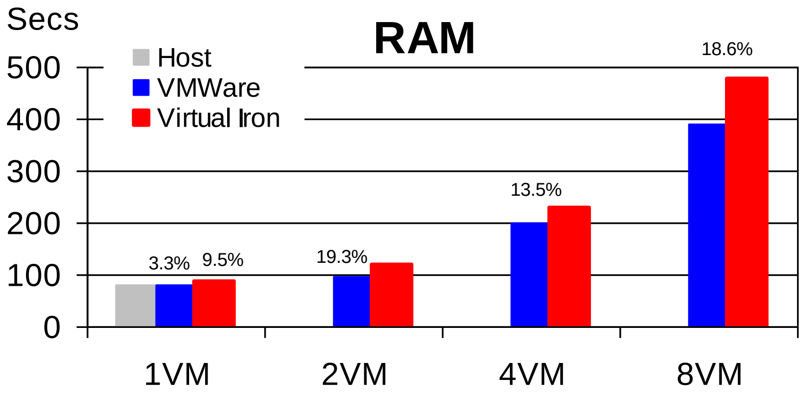

Figure 8. Performance degradation in memory access operations

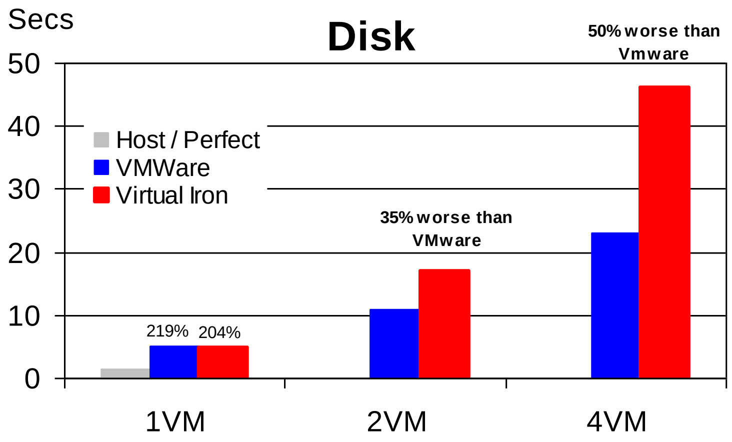

Figure 9. Performance degradation in Disk intensive operations

B. Performance isolation of the virtual machines

The results of this test are shown in figures 10 and 11. The goal of the test was to verify whether resources can be allocated in a consistent manner to a specific virtual machine. The test creates two virtual machines and gives them different priority (CPU allocation). The difference in priority is expressed as the quotient between the number of shares given to machine 1 (in blue) and the number of shares given to machine 2 (in red). A higher number of shares implies higher priority. Both machines are then given a specific task which takes a known amount of time and both start executing this task at the same time.

The physical machine contains only one processor, therefore the time has to be shared between both tasks. The total CPU time of execution should be constant, while the scheduled CPU bandwidth and the time of the higher priority machine should both change according to the priority settings (it should be remembered that CPU Time equals CPU Bandwidth *x* Execution Time).

The expected results (which match the experiment precisely) are:

- The machine with least priority will always finish after the same amount of time (that is the time that it takes to execute the task twice)
- The machine with highest priority will always finish first: the higher the priority, the sooner this machine will finish. When the difference in priority is significant, the first machine will complete its work almost before the second one starts processing (it will take roughly one half of the time).

Performance isolation (by total time)

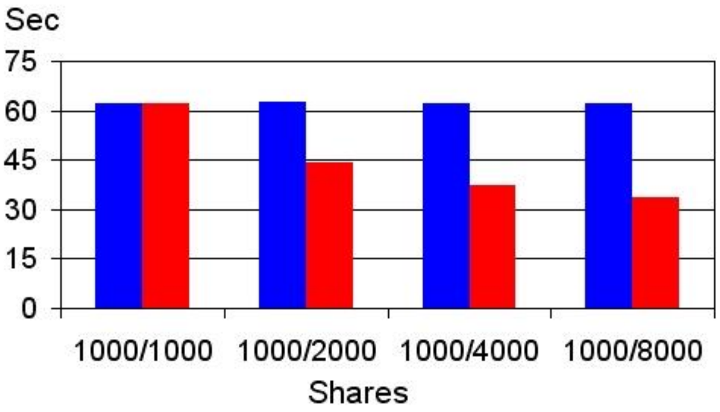

Figure 10. Performance isolation test

Performance isolation (by total percentage of time)

Figure 11. Performance isolation test

The results on figure 11 – which shows how the CPU time for each execution is always constant - were inferred, being the logical explanation for the results shown in figure 10.

C. Fault isolation

The setup of this test is shown in figure 12. One VM is subject to the introduction of failures which eventually leads it to crash. Two other machines (passive victims) run on the same physical hardware and are observed to determine if they suffer undesired failures themselves.

A total of 59000 failures were introduced in this test during 300 hours. The target machine failed (as expected) 3960 times. However, the two passive victims only produced 3 errors, none of which could be reproduced again. This shows that performance isolation is very good.

Figure 12. Performance isolation test

VI. Conclusions

The work described in this paper has analyzed the ESOC environment, extracting requirements for a new ground data systems architecture based on virtualization. The different virtualization technologies and products have been tested in detail, with good overall results.

A new architecture based on virtualization was proposed. This architecture takes into account the special requirements needed in an operational environment. A prototype was produced to assess the viability of the approach.

In overall, the results of the work show that significant improvements would be gained using virtualization. The following benefits in particular are expected:

- Efficient usage of the hardware resources;
- Isolation of applications from the hardware;
- Simplification of the management of the environment;
- Simplification of application deployment.

Some open issues were identified and will be further analyzed:

- Integration with existing networks;
- Data backups;
- Out of band monitoring;
- Access to disk for disk intensive applications such as databases.

The work is currently being continued to complete the assessment and address all the outstanding points. A complete test infrastructure for the development environment (as opposed to the operational environment) is expected to be ready by Q3 2008. New tests will be performed using this infrastructure which may lead to a first deployment of virtualization in the operational environment.

In addition to this the virtualization market is being carefully monitored as new products and techniques appear regularly.