18th International Conference on Space Operations, Montreal, Canada, 26 - 30 May 2025.

SpaceOps-2025, ID # 1 (101)

The resilience of the Space system by the virtualisation

Jamel Metmati[a]

[a] *Cyber Science Institut, ISAE-SUP AERO, France*, *Djamel.METMATI@ext.isae-supaero.fr*

Abstract

The virtualisation concerns the re-configuration of an equipment itself for another purpose providing a virtual machine able to use an application or a function and the capacity to use the memory capacity between the ground and the Space segment. The processors and the memory management provide the capacity to manage the computer on board in allocating the ressources for the sensors, the system and the subsystems as well the latency, the jitter, the packet loss. These elements concern the watt management from inputs for need ressources, optimization and the on line operations with less interface human-machine in the mission control. As the re-configuration of the system from the ground to new functionalities of an equipment on the ISS station for computer in Space demonstrates the applicability to use the memory potential of an equipment, some functions should be re-used differently. Moreover, the generation of satellites incoming uses processors with much memory providing the functionalities for data in Space and the data management on the ground. This reinforces the resilience of Space system in the conception and the sustainability allowing a re-use of equipments in Space.

Keywords: virtualisation, container, processor

Acronyms/Abbreviations : MBSE, SAN, NAS, API, LUN, KVM, QEMU

1. Introduction

The objective is to present an architecture to improve the on board computer management by the functionalities of the virtualisation through the simulated model of the processors. Then, with the micro-services the way to share by the stack solution the capacity to support the resources on board and to manage computation between the ground segment and the Space segment through the virtualisation. This to improve the re-configuration for priorities on board and to manage the functionalities from the re-configuration of the equipment by the capacity to re-program on remote thanks to the virtualisation function.

2. Material and methods

In order to reach the re-configuration and the management of functionalities, the system approach requires to consider an use case for an embebbed processor design to support several generic sub-systems for Earth observation. The Model Base Engineering is used from the optimized approach to be closer for the requirements of the virtualisation processing [1].

The features of the processor as the sub-systems are simulated from their industrial technical characteristics. From this, the materials used are the linux environment with micro-service architecture to simulate the on-board computer, the internal connections, and the connections between the Space segment and the ground segment.

The ground segment shall be considered by a virtual machine to test the functionalities of re-configuration and the management of functionalities to the processor. The virtual machine gets the command and control function with a connection to the processor on board. It can complete by the server function to store the logs, to share the memory of the processor, to deploy an agent to automate the inputs status from the processor. This last point should act directly on the operation of the processes in the processor memory. All is to be integrated in the NAS, SAN workspace for the management of the processor [2].

3. Theory and calculation

The architecture is built from the simulated model linked to MBSE approach thanks to the capacity to micro-service solutions. The software containers contains the function and the parameters of the systems and the subsystems and each one is connected with each others to simulate the features of bus for the embebbed system.

This point shall provide the inputs for the memory management through the linux software environment used through the PID to identify the jobs, the building of the packets making possible, the observability of the real time and shared time of the processor. The ground segment and the function of virtualisation between the on board computer and the ground are built through an API from the same computer with virtual machine or it can integrate in a provider which uses the functions of the ground segment. In this case, the function of virtualisation shall be in the packages available from the providers.

The data flow is simulated through the architecture and the functions to be underlined for the on-board management computer. The functions of the virtualisation tested for the proof of concept is to able to share the memory between the ground and the embebbed system. This for the handlers and priority management in doing in the board computer through check, security and safety of the systems as sub-systems. The data features is produced by small bots to test the connections and the behaviors of the systems and sub-systems.

The calculation for the architecture and the data is taken account following the MBSE in the blocks before to code it in the software environment. The computation shall be closed with the features of the processors, the bus, the configuration already working and used in Space. The use case provides accurate data to built the micro-service architecture.

The on-board computer manages the sub-systems (figure 1) with the data flow and the bus power. The ground segment considers to get an access to the OBC memory to follow the parameters and check the anomalies from the other systems [3].

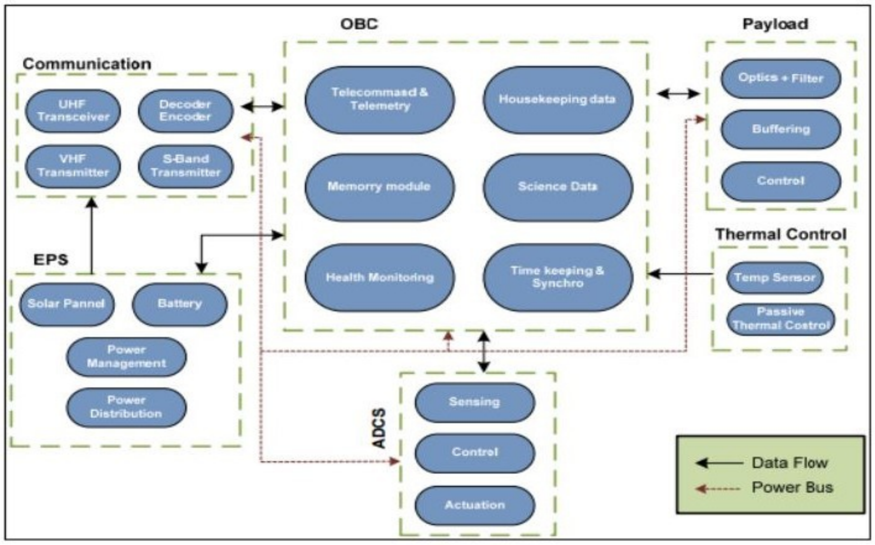


Figure 1 : on-board computer and its subsystems

The ground segment architecture shall be considered as the data system to get down the data by its physical architecture with the function of virtualisation include on it. It means thanks to a server and a virtual machine to capacity to share through the SAN/LUN/NAS functionalities the memory of the processor in a pipeline. This is, then used in a simulated model to provide the capacities of management by emulating the embedded system without expecting a timer slot for a link to Space.

The function of virtualisation is implemented on the ground segment facilities and the use is done through a virtual machine. The effects of the function can provide the check and the follow up of the on board computer with the

machine learning options for the treatment on remote on the embebbed system. Through the Model Base System engineering linked with the data model from an use case, the simulated model shall be considered by the figure 2 .
At last, the point is to emulate the processor of the on-board computer on remote able to be manage by the virtual function from the ground. The main purpose is to able to manage the system on board rapidly to ensure the safety of the payload.
The simulated model is built from the DO178C, MISRA-C regulations to ensure to be closer for the standard for a mission context and the system requirements [4]. The LEON processor is used for the model. The software environment considers a workstation with linux distribution with the packages for the virtualisation QEMU and KVM.
Once installed, it considers the micro-services to align the function of on computer on board with the parameters of the mission planned on orbit. It means the OBC is constructed by a docker container and the sub-system by other containers linked with the need to check the process on board.
The examples of energy is taken account for the function is critical for the embedded system on orbit. The bus and the data flow are represented by the API which is configured with the capacity of a bus to transmit data. The virtual machine is installed on the workstation to send and to receive data from the OBC with the test planned.
The extension of the simulated model is to use an agent or a bot to test the routine of the processor on board on the simulated model. The API makes the job between the digital model and the physical data process on the infrastructure. This function is double for the management of the simulated model and to be able to be connected with a SAN/LUN/NAS shared with the physical infrastructure. This storage provides the emulation of the OBC on the way on orbit.
It gives the inputs to the simulated model for the extension of the management thanks to the virtualisation function. For this case, it shall be considered that the virtualisation is integrated in the physical processor on orbit with Intel VT-x or AMD-V and available on the BIOS.
The architecture of the virtual function for the OBC from the ground with simulated model and the potential of the extension provides the possibility to test the OBC without breaking down the physical OBC on orbit. The tests would boil down by the capacity to use several applications, operating system, the CPU and memory management. The scenarios taken account are linked with the off-line operation and the on-line operation. The off-line operation considers the function of virtualisation from the simulated model. The on-line operation considers the work of the function between the simulated model and the physical connections ground-to-Space.
The paper will process the simulated model till the workplace shared with the physical model to view the time slots, the service duration, the task deadline. The principle adopted is the Kraken CI with the minimum of storage and computation. The block diagram (figure 2) includes :

- The OBC block with the parameters of LEON processors in which the emulating function is active to display the parameters on the ground
- The communication module for the virtual link with the ground-to-Space
- The workplace shared configured as SAN/NAS/LUN for the gateway with physical architecture
- The LUN is simulated from the iSCSI local server on the VM

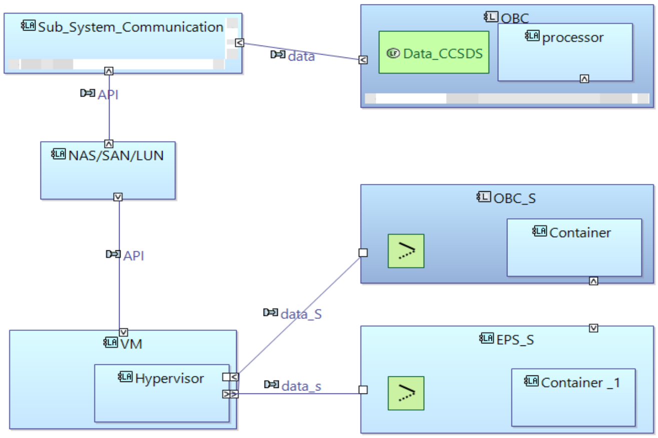


Figure 2 : Virtualisation architecture

4. Results and Discussion

As the results considers the use of the virtual function in the context of the LUN. The model can be extended through the NAS/SAN in the operational context. Then, the LUN parameters is configured from the virtual machine by the local server. It considers the use of the virtual function to the data available on a server shared with the operational missions. Moreover, the model combines the emulation in the virtual machine to reproduce the on-board computer in the virtual environment through the containers. The results should be understood by the model between the virtual machine and the containers, the OBC_S and the EPS_S susb-system.

The model uses the LEON processor with the following parameters available from the satellite catalogs. The on-board management takes the data from the API in the LUN and inside the virtual function, the emulation of the processor with the containers. The virtual machine plays only the function of command and control. The parameters can be built with a parametric diagram to make the off-line or on-line operation depending of the scenarios planned for the on-board computer management.

The LEON processor is linked with the SPARC architecture, the choice of the cycle clock is the 100MHz [5]. The environment can test the values between 50Mhz and 200 MHz depending on the scenarios. The memory cache from the 16 to 32Ko provide the variation to check the behavior of the processor. The choice used is the 20Ko.

The communication interfaces like UART, SPI, I2C, Ethernet are the bridges and the API. The model considers the SpaceWire and the Ethernet for the data CCSDS. The rate of data signaling displays 2Mbits/s minimum beyond 100 Mbits/s. The LVDS point provides the tension which must be produced to transmit *n*Ko bits of data depending of the Watt value. For the 10 Ko bits of data with the rate of signaling at 2Mbits/s with 0,5 A, the times to send the datas would be 0,00512 seconds for a 3 V tension as described by V = P/I

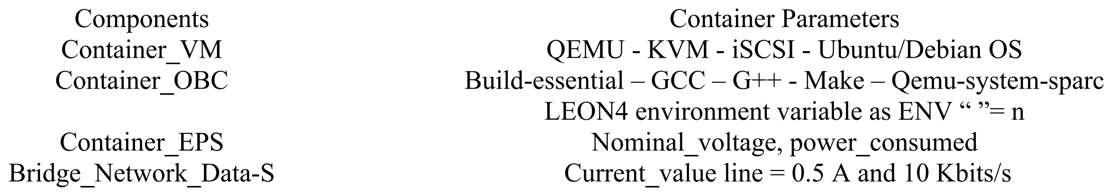
$$V = P / I \wedge T = data / tx_s \left(10 * 1024 \right) / 2 * 10 \wedge 6 \, bps \,\vert$$

It considers the bus parameters on terms of the bandwith and the delays. The radiation tolerance are not taken account for the processor requirements integrate already this constraints in its design. The energy use ranges with 1 Watt and 3 Watt for the LEON4 processor. The parameters choosen for the model is 1,5 Watt producing 0,5 A. The internal bus called AMBA 2.0 gets a large band between 64 bits and 128 bits providing performance of data for the on-board computer and the subsystem EPS. It can be considered that the value is 80 bits. The parameters complete the model with the parametric table (figure 2- figure 3):

Parameters	Components	Bus
10 Ko bits, 3V	OBC	Bridge Network
3 V	EPS	Bridge_Network
10 Ko bits, 3V	VM-OBC-EPS	Bridge_Network
CPU 1, 1GHz, RAM 512 Mo, 10 Go storage	VM-LUN	VM Kernel, static IP, port

Figure 2 : The parameter model

The simulated model with the components, the parameters, the bus value for the bridge between the virtual machine and the sub-systems OBC plus EPS shall be considered by the figure 2. The code for the containers OBC and EPS could be oriented through the value on below for the OBC LEON4 in the context of docker and Python technology [6].



Components	Container Parameters
Container_VM	QEMU - KVM - iSCSI - Ubuntu/Debian OS
Container_OBC	Build-essential – GCC – G++ - Make – Qemu-system-sparc LEON4 environment variable as ENV " "= n
Container_EPS	Nominal_voltage, power_consumed
Bridge_Network_Data-S	Current_value line = 0.5 A and 10 Kbits/s

Figure 3 : the container parameters

The result for the data flow from the virtual function from the Virtual Machine shows several features for OBC management considering data available from LUN with iSCSI and the emulation inside the virtualisation function with the connection to the OBC container.

The off-line operation considers the data of OBC from data of the on-line operation. The use of the data available provides a software environment to test the scenarios of management for the OBC. The scenarios applicable should be once the initial data from the OBC received on the LUN :

- The first scenario would be to change the parameters of the bridge to notice the behavior of data transfer on the simulated model,
- The second scenario is to consider the management of the real-time and the interruptions in the data process. For example, the PicSat recorded a shutdown in 2018 after its launch and a wake-up of the signal from the on-board computer in 2022. To consider the reverse monitoring of the system and the sub-system, the scenario for this example means to be able to get the memory logs and use the commands to manage the priorities plus the resources. From the virtual machine, the connection through the virtual pipeline from LUN/SAN would get the steps following : the access of the interruptions logs is used through the command line "watch -n/proc/interrupts" to display the row expected. The "irq" provide the number of the interruption. Depending on the causes of the problem to avoid the processor put in safe mode, the other steps would be to introduce POSIX to limit the effects of one process on another one by the mutex sequences. Then, to further, the shared memory linked with virtual pipeline through RTAI and Xenoami could be managed. The "mmap" command should provide the path to share the physical memory between the process and the kernel of the processor on board. The "mlockall" would lock the virtual memory to physical memory. The way is to use the shared memory between the on-board computer in the satellite and the ground segment through the LUN and the virtual machine connected with this workspace. For this item, the embebbed processor working with the virtualisation process with its sub-system.
- The third scenario is to implement an agent with a low level of CPU computation to get the data stored in the SAN from the LUN by the agent from the virtual machine. And then, the purpose would be to re-inject the data in the simulated model. The variation of the scenarios depends on the architecture and the capacity for the ground segment to virtualize the connections from the physical infrastructure.

5. Conclusions

The resilience of Space system provides the inputs to re-code the function of the assets by the networks ground-to-Space by the direct methods on the materials or from the shared workspace to test the configuration before to inject the data to Space. The virtualisation function can be extended in a cloud system to provide the computation with the other functionalities for application.

References

Reference to a conference:
[1] "Generic approach for systems design optimization in MBSE,". In : The 22nd Annual INCOSE International Symposium. Broodney, H., Dotan, D., Greenberg, 2012.

Reference to a book:
[2] La virtualisation avec VMWare vSphere 8, notions fondamentales, Luc Breton, 2023.
[3] Small Satellite Mission for Earth observation : new developments and trends, Rainer Sandau, 2024.
[4] "Software product line engineering : Foundations, principles and techniques.". s.l. : Springer, Pohl, K., Böckle, G., & van Der Linden, F. J., 2005.

Reference to a journal publication :
[5] Van Huong, P., & Binh, N. N. 2012. "Embedded system architecture design and optimization at the model level". International Journal of Computer and Communication Engineering. 2012, Vol. 1, 4, p. 345.

Reference to a website:
[6] https://github.com/CSI-Space