Virtualization: Requirements

Version 1.0, April 14, 2008

Questions pertaining to this document, or the terms or conditions of its provision, should be addressed to:

SCOPE Alliance,
c/o IEEE-ISTO
445 Hoes Lane
Piscataway, NJ 08854
Attn: Board Chairman

Or

For questions or feedback, use the web-based forms found under the Contacts tab on www.scope-alliance.org

1. INTRODUCTION

This document of the SCOPE Alliance Virtualization Working Group aims to provide a set of hardware and software virtualization requirements that describe functional and non-functional aspects of virtualization features required by the NEPs/TEMs in support of their carrier grade telecommunications solutions. The initial focus of the Virtualization Working Group is on the core networking area, with emphasis on the control and data planes. Other networking areas might be considered in the future.

This document relies on the SCOPE Alliance Virtualization: State of the Art document [2], which classifies the various virtualization approaches along with their goals, advantages and drawbacks. The Virtualization: State of the Art document provides a glossary of terms and definitions related to virtualization.

The virtualization requirements, presented herein, have been generated based on a set of business and technical use cases provided by the NEPs/TEMs. The use cases collectively describe the areas where virtualization might be of value to them and are described in the SCOPE Alliance Virtualization: Use Cases document [3].

The virtualization requirements are intended to be used by the ecosystem providers for delivery of carrier grade virtualization features and solutions for use by the NEPs/TEMs. An ecosystem provider may choose to address all or part of the virtualization requirements, depending on the elements of the ecosystem that it aims to provide and the use cases it aims to address.

2. REFERENCES

1. DMTF, DSP 1057, Virtual System Profile, http://www.dmtf.org/standards/published_documents/DSP1057.pdf

2. SCOPE Alliance, Virtualization: State of the Art, http://www.scope-alliance.org

3. SCOPE Alliance, Virtualization: Use Cases, http://www.scope-alliance.org

3. VIRTUALIZATION REQUIREMENTS

The virtualization requirements, identified by the SCOPE Alliance Virtualization Working Group, are grouped into the following categories:

- VM Definition
- VM Activation
- VM Operation
- VM Fault Handling

- VM Administration
- VF Deployment
- VF Operation
- Security
- Performance
- Availability.

The requirements in the above categories are generally considered to be of higher priority. In addition to the above categories, the Virtualization Working Group has identified another category:

- Live Migration.

The requirements in this category are considered to be of lower priority. They are to be implemented after the requirements in the other categories have been implemented.

The requirements are split into two main classes: Virtual Machine (VM) requirements and Virtualization Facilities (VF) requirements. See the Virtualization: State of the Art document [2] and Section 2.2 for the definitions of these terms. To determine how to assign requirements to these two classes, the Virtualization Working Group ruled that VM requirements apply to a single VM, while VF requirements apply to the VF software.

Many of the VM requirements have counterpart VF requirements. The Virtualization Working Group has chosen not to explicitly state (or duplicate) such requirements. This choice was based on the implicit generic requirement – or assumption – that all VM requirements must be supported by the underlying VF.

The Security, Performance and Availability requirements are orthogonal to the previous two classes and might apply to both the VM and the VF.

3.1 Methodology Used

The SCOPE Alliance Virtualization Working Group developed a set of requirements for each of the use cases presented in the Virtuallization: Use Cases document [3], and then aggregated them into the above 10 higher priority categories. They determined that some of those requirements -- the requirements that pertain to Live Migration -- are of lower priority and created a separate category for them.

Each of the categories was then reviewed. Requirements were reworded, split, removed, postponed, moved to another category or coalesced with other requirements when they were seen as duplicates.

The requirements were then associated with their original use cases, so that they can be traced back to those use cases. Some of the requirements given below do not make reference to any use case.

3.2 VM Lifecycle

Virtual Machines (VM) are hosted on software/firmware platforms that are named Virtualization Facilities (VF). The term Virtualization Facilities is used to remain as generic as possible and to avoid reference to specific implementation details.

Virtual Machines have a lifecycle. A brief overview of this lifecycle of a VM is given below. It is based on the description provided in the DMTF DSP 1057 Virtual System Profile [1], and is illustrated in Figure 1.

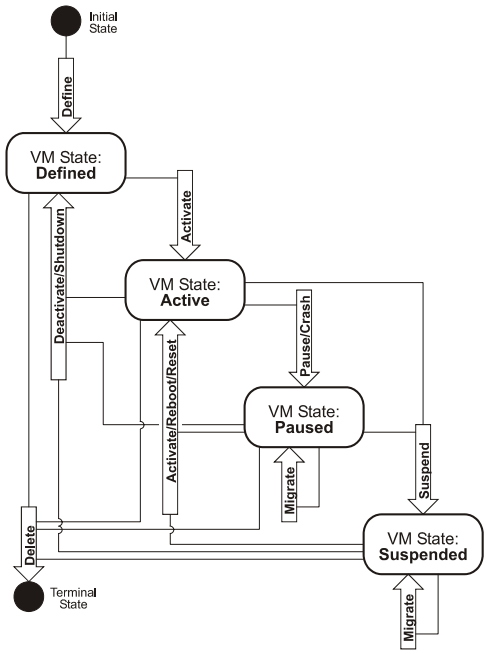

Figure 1. VM lifecycle.

The first step in the lifecycle is that the Virtual Machine needs to be **defined**. This definition is a description of relevant VM properties, characteristics and needs. A definition does not create or instantiate an active running Virtual Machine on a Virtualization Facility. Requirements related to this step are covered in Section 3.4.

The second step in a VM lifecycle is to **activate** the VM. At this point, the VM is created and becomes an active entity that consumes resources. As part of the activation process, abstract properties and needs of the VM defined in the first step must be **mapped** onto the real resources of the target system. Requirements related to this step are covered in Section 3.5.

When a Virtual Machine is active, **operations** can be applied to the VM. As part of this step, a VM can be paused or suspended and activated again. A VM is also subject to shutdown, reset, and reboot operations. Requirements related to this step are covered in Section 3.6.

The normal operation of a VM may be affected by **faults**. Requirements related to such events are covered in Section 3.7.

Finally, the VM must be **administered**. Related requirements are covered in Section 3.8.

The VM might also undergo **live migration**. Requirements related to live migration are given in Section 3.14.

In addition to the VM requirements, there are requirements for Virtualization Facilities (VF). Section 3.9 covers VF deployment requirements and Section 3.10 covers VF operation requirements.

The Security, Performance and Availability requirements, which are orthogonal to the VM and VF requirements and might apply to both VMs and VFs, are given in Section 3.11, Section 3.12 and Section 3.13, respectively.

3.3 Implementation Considerations

It is expected that virtualization will be available on commonly used processor architectures that are used in the NEPs/TEMs equipment. Depending on the platform and the equipment, different kinds of processors may be used. In particular, it is expected that

- Virtualization will be available on commonly used processor architectures in the NEPs/TEMs equipment. Depending on the platform and the equipment, different kinds of processors may be used.
- Virtualization will be supported on both 32 bit and 64 bit processor architectures, wherever this is meaningful.
- Virtualization will run on unicore, multithreaded, multicore processors, as well as on SMP and NUMA architectures.
- Processors may or may not be fully virtualizable, and they may provide hardware-based memory protection mechanisms.

Moreover, it is expected that virtualization will be provided through different means. In particular,

- It may be necessary to modify the guest OSes, and/or to configure them with appropriate modules and/or device drivers to run inside a VM.
- Native binary guest OSes may be supported without required adaptations.
- Virtualization may be provided through firmware, software, or a combination of firmware and software.
- Virtualization may be provided by a Type I hosted approach or a Type II native approach, as well as by OS virtualization.

The requirements expressed hereafter are as far as possible implementation independent, with a few exceptions where OS virtualization is explicitly addressed. They are shown below in tabular format grouped into the 10 higher priority categories. In addition, the requirements for the lower priority Live Upgrade category are given.

3.4 VM Definition

Requirement Number	Requirement Description	Use Case
VMDEF1	As part of the VM definition, it SHALL be possible to define a boot sequence for all VMs (i.e., sequence VM booting).	6
VMDEF2	The VM definition SHALL enable specification of the following VM attributes: - Scheduling properties and quality of service (QoS) including interrupt latency and minimum timeslot allocation - Physical devices and their required QoS, including bandwidth, latency (e.g., caused by sharing) and exclusive access - Number of virtual processors and associated QoS requirements - Memory isolation requirements - Communication isolation requirements - Console - Memory requirements (see VMDEF6, VMDEF7, VMDEF8, VMDEF9) - Boot devices (see VMDEF4, VMDEF5)	2,3,5,6,11
VMDEF3	The VM definition SHOULD enable specification of the following VM attributes: - Number of physical cores and threads of the virtual CPUs including affinity and exclusive access requirements - Arbitrary parameter passing to the VM on initialization - Fault handling policy - Logging requirement - Endian configuration requirement	2,3,5,6,11
VMDEF4	The VM definition SHALL support the specification of a first executable or boot image to be loaded on initialization of a VM. It is up to this first executable to provide appropriate ways of booting. The VM definition SHALL support the use of the following boot methods: boot from disk, boot from network (PXE, ...), boot from local and/or network storage ...	2,3,5,6,11
VMDEF5	The VM boot mechanism SHALL support commonly used PROMs (OpenBoot, BIOS, EFI, ...)	2,3,5,6,11
VMDEF6	The VM definition SHALL enable the specification of a minimum memory allocation required for a VM.	2,3,5,6,11
VMDEF7	The VM definition SHOULD enable the specification of the page size for a VM.	2,3,5,6,11
VMDEF8	The VM definition SHALL enable the stipulation that a VM requires static memory management.	2,3,5,6,11
VMDEF9	The VM definition SHOULD enable the stipulation that a VM requires dynamic memory management.	2,3,5,6,11

3.5 VM Activation

Requirement Number	Requirement Description	Use Case
VMACT1	The physical resources required by a VM (e.g., CPU, memory, other physical devices) SHALL be assigned to the VM at activation time through one of the following mechanisms: (1) A dynamic, automatic, implicit mapping performed by the VF without any information other than the VM definition (2) A mapping based on precise constraints given either as part of the VM definition or as part of the VM activation (3) A mapping using pre-reserved resources for the VM. When this mechanism is used, pre-reservation and activation may be performed at different moments in time. Not all physical resources are candidates for mechanisms (2) and (3).	5
VMACT2	The VFs SHOULD provide administrative mechanisms to allocate specific physical resources (e.g., CPU, memory, etc.) dynamically to a VM. This requirement might not be meaningful for all physical resource types.	5

3.6 VM Operation

Requirement Number	Requirement Description	Use Case
VMOPE1	The failure of a VM, or any software within a VM, SHALL NOT affect any other VMs. Application behavior SHALL be preserved in the virtualized environment. Application behavior SHALL be independent of the underlying VFs.	1,3,5,11
VMOPE3	Hardware-based VFs SHALL support concurrent operation of multiple (two or more) VMs, each of which may run different workloads. There SHOULD NOT be an arbitrary limitation (upper bound) imposed by the VF on the number of VMs that can be concurrently supported.	2,3,4,6,8
VMOPE4	The VFs SHALL provide an administrative means to activate, deactivate, pause, suspend, shutdown, reboot, and reset one VM without impact to any other VMs defined and/or operational on the underlying VF.	2,8,11
VMOPE5	The VFs SHOULD provide a programmatic means to activate, deactivate, pause, suspend, shutdown, reboot, and reset one VM without impact to any other VMs defined and/or operational on the underlying VF.	2,8,11
VMOPE6	All dynamic resources of a VM SHALL be reclaimable, i.e., there SHALL be no "leakage" of system resources.	5

3.7 VM Fault Handling

Requirement Number	Requirement Description	Use Case
VMFH1	The crash and reboot of a VM SHOULD be detected by the VF. This detection SHOULD be provided by a watchdog function. The VF SHALL provide restart of the VM on detection of VM workload failure. The VF SHOULD provide mechanisms to configure which actions to take when such an event occurs. These actions SHOULD include the delivery of an event notification to predetermined addresses, as well as dump, reboot and restart policies and other policy-based actions.	5
VMFH2	In the event of a CPU failure on a physical server with multiple CPUs, the virtualized CPUs of all active VMs SHOULD be remapped to the remaining physical CPUs on the underlying physical server. The VF SHOULD then degrade the QoS of the active VMs following appropriately defined policies. Notifications and categories of failure are to be generated. Failure of a memory or a network device is to be handled by the VF in a similar way to CPU failure.	

3.8 VM Administration

Requirement Number	Requirement Description	Use Case
VMADM1	The VF SHALL provide Key Performance Indicators (KPIs) to facilitate the load status of individual VMs. Among the KPIs required are CPU and memory usage, I/O throughput, number of context switches, storage allocation, and latency. These indicators SHALL be provided on a single VM basis. In addition, a platform aggregated view SHALL be possible.	8
VMADM2	The VF SHOULD provide mechanisms to report if the configuration requirements of a particular VM can be met by a given physical server.	11
VMADM3	The VF MAY provide mechanisms to report whether the set of all configuration requirements of a set of VMs can be met by a given physical server.	11
VMADM4	The VF MAY provide mechanisms to reserve (prior to VM creation time) resources that will be needed to create a given VM.	11
VMADM5	If VF mechanisms for resource reservations are provided (as in VMADM4), it SHALL be possible to remove such reservations without having to step through the VM creation and VM removal phases.	11

3.9 VF Deployment

Requirement Number	Requirement Description	Use Case
VFDEP1	The VF SHALL support in-service/out-of-service operator-initiated upgrades from the currently deployed release N to a new release N+1.	13,14
VFDEP2	The VF SHOULD support in-service/out-of-service operator-initiated upgrades from the currently deployed release N to a new release N+M, where M is greater than or equal to 1 and less than or equal to 3.	13,14
VFDEP3	The VF SHALL support in-service/out-of-service automated upgrades from the currently deployed release N to a new release N+1.	13,14
VFDEP4	The VF SHOULD support in-service/out-of-service automated upgrades from the currently deployed release N to a new release N+M, where M is greater than or equal to 1 and less than or equal to 3.	13,14
VFDEP5	The in-service/out-of-service VF upgrade process SHALL support execution of pre- and post-upgrade procedures, written in a shell programming language of the vendor's choice.	13,14
VFDEP6	The in-service/out-of-service VF upgrade process MAY be initiated by the VF itself.	13,14
VFDEP7	It SHALL be possible to initiate the in-service/out-of-service VF upgrade process from an on- or off-site administrative access point external to the VF.	13,14
VFDEP8	The in-service/out-of-service VF upgrade process SHALL provide a means of local and remote progress indication.	13,14
VFDEP9	The VF SHALL support backup and restoration of VF configuration data, necessary for continued normal operation of the VF.	13,14
VFDEP10	The VF SHALL provide an administrative interface for export of backed up VF configuration data.	13,14
VFDEP11	The VF SHALL provide an administrative interface for import of backed up VF configuration data.	14
VFDEP12	The VF SHALL provide an administrative interface to enable automatic initiation of the backup and restoration operations during the in-service/out-of-service VF upgrade process.	14
VFDEP13	The VF configuration data, backed by release N of the VF, SHALL be consumable by VF release N+M, where M is greater than or equal to 1.	14
VFDEP14	The VF SHALL provide the capability to suspend and resume the execution of the hosted VMs during an out-of-service upgrade. The suspend and resume operations SHALL be VF release agnostic.	
VFDEP15	The in-service/out of-service upgrade process SHALL support roll-back to the previous version of the VF. If the upgrade process contains steps that are not easily undone, the VF SHALL clearly identify such steps via an explicit commit operation.	

VFDEP16	It SHALL be possible to complete one in-service/out-of-service upgrade, in its entirety, during a single 4 hour maintenance window.	
VFDEP17	An in-service VF upgrade SHALL have no impact on the availability of the operational VMs hosted by the VF that is being upgraded.	
VFDEP18	An in-service VF upgrade SHOULD have minimal measurable impact on the performance of operational VMs hosted by the VF that is being upgraded.	
VFDEP19	Changes that impact the configuration and/or operation of the VMs, resulting from an in-service/out-of-service upgrade of the VF, SHALL NOT negatively impact the VMs.	
VFDEP20	The VF SHALL support in-service/out-of-service operator-initiated patching from the currently deployed release N to a new release N+1.	
VFDEP21	The in-service/out-of-service VF patch process SHALL support execution of pre- and post-patch procedures, written in a shell programming languages of the vendor's choice.	
VFDEP22	The VF SHOULD be able to self-initiate the in-service/out-of-service patch process.	
VFDEP23	It SHALL be possible to initiate the in-service/out-of-service VF patch process from an on- or off-site administrative access point external to the VF.	
VFDEP24	The in-service/out-of-service VF patch process SHALL provide a means of local and remote progress indication.	
VFDEP25	The VF SHALL provide an administrative interface to enable automatic initiation of the backup and restoration operations during the in-service/out-of-service VF patch process.	
VFDEP26	The VF SHALL provide the capability to suspend and resume the execution of the hosted VMs during the out-of-service patch process. The suspend and resume operations SHALL be VF release agnostic.	
VFDEP27	The in-service/out of-service process SHALL support rollback to the previous version of the VF. If the patch process contains steps that are not easily undone, the VF SHALL clearly identify such steps via an explicit commit operation.	
VFDEP28	It SHALL be possible to complete one in-service/out-of-service patch process, in its entirety, during a single 4 hour maintenance window.	
VFDEP29	The in-service VF patch process SHALL have no impact on the availability of the operational VMs hosted by the VF that is being patched.	
VFDEP30	The in-service VF patch process SHOULD have a minimal measurable impact on the performance of operational VMs hosted by the VF that is being patched.	
VFDEP31	Changes that impact the configuration and/or operation of the VMs, resulting from the in-service/out-of-service patch process of the VF, SHALL NOT negatively impact a VM.	
VFDEP32	VF configuration mechanisms SHALL be provided to specify how the VF is booted on the target platform.	

VFDEP33	VF configuration mechanisms SHALL be provided to define an ordered list of devices from which the VF will be booted. Example boot mechanisms to be supported include local disk, PXE, …	

3.10 VF Operation

Requirement Number	Requirement Description	Use Case
VFOPE1	The VF SHOULD support definition of VM scheduling policies.	
VFOPE2	Each VM scheduling policy SHOULD support simultaneous operation of real and non-real time VMs.	
VFOPE3	The VF SHOULD support dynamic changes to VM scheduling policy attributes at any time prior to or following VM activation.	
VFOPE4	The VF SHOULD support VM monitoring on a per-VM basis.	
VFOPE5	VM monitoring data, provided by the VF, SHOULD at minimum include CPU activity, interrupt activity, and memory usage information.	
VFOPE6	The VF SHOULD provide a means of exporting VM monitoring data using a portable format.	

3.11 Security

Requirement Number	Requirement Description	Use Case
VMSEC1	The VF SHOULD provide a means of relaxing strong memory isolation between hosted VMs.	5

3.12 Performance

Requirement Number	Requirement Description	Use Case
VMPERF1	The run-time VF overhead SHALL be bounded within an advertised range.	
VMPERF2	The run-time VF overhead SHALL be tunable in terms of VF CPU utilization, VF memory utilization, VF storage, and network I/O utilization.	
VMPERF3	The VF static memory and storage footprint SHOULD be kept to a minimum.	
VMPERF4	The memory footprint, required to store per-VM metadata within the VF, SHOULD be kept to a minimum.	
VMPERF5	The VF startup and shutdown time SHALL be within customer-specified bounds.	
VMPERF6	The VF SHALL support operation of real-time workloads within VMs.	
VMPERF7	The VF SHALL enable a real-time workload response guarantee.	
VMPERF8	The VF SHALL ensure that VMs with higher execution priority SHALL NOT be negatively impacted by VMs with lower execution priority.	
VMPERF9	The impact of the VF on the VM interrupt latency SHALL remain within advertised bounds regardless of the number of VMs being hosted by the VF.	

VMPERF10	The VM context switch latency imposed by the VF SHALL remain within advertised bounds regardless of the number of VMs being hosted by the VF.	
VMPERF11	The VF SHALL support per-VM access to underlying storage and network I/O resources, such that a VM with a higher priority access SHALL NOT be negatively impacted by a VM with similar lower priority.	
VMPERF12	The VF SHALL support dynamic reallocation of system resources across running VMs (e.g., I/O devices, CPU, memory).	
VMPERF13	Running VMs, that are affected by dynamic reallocation of resources, SHALL remain fully operational while resource reallocation takes place.	
VMPERF14	The dynamic reallocation of system resources across VMs SHALL be performed within a bounded time limit.	
VMPERF15	The VF SHALL be able to start, stop or suspend a VM in a bounded amount of time and without affecting the performance of the other VMs and the VF itself.	
VMPERF16	The VF SHALL be capable of detecting a VM failure within a defined tunable time limit.	
VMPERF17	The VF SHALL NOT negatively impact the QoS of running VMs while collecting VM statistics and diagnostic information.	
VMPERF18	The VF SHALL enable on-demand display of state information of a given VM without negative performance impact on other VMs.	
VMPERF19	The upgrade of the workload of a VM SHALL NOT have negative impact on the performance of other running VMs.	

3.13 Availability

Requirement Number	Requirement Description	Use Case
VMAVAIL1	The VF SHALL enable periodic snapshots of VM data, necessary for restart of a VM at a later time.	
VMAVAIL2	The amount of time between VM snapshots SHALL be configurable through the VF.	
VMAVAIL3	The VF SHOULD preserve VM network context during VM failover.	

3.14 Live Migration

Requirement Number	Requirement Description	Use `Case
VMLIVMIG1	The VF SHOULD support initiation of automatic live migration of VMs to another physical server when physical failure detection on the originating physical server occurs. The covered failures are typically local I/O errors that don't affect the ability of the VMs to run and to communicate with remote servers.	9
VMLIVMIG2	The live migration of a VM SHALL NOT result in loss of service provided by the workload hosted inside that VM.	
VMLIVMIG3	The migration of a VM SHALL NOT have negative impact on the performance of other running VMs or the VF.	
VMLIVMIG4	Any latency incurred during live migration of a VM SHALL be bounded.	
VMLIVMIG5	The VF operational overhead SHALL remain within predefined bounds during the live migration of VMs.	
VMLIVMIG6	The VF SHALL generate an event at VM migration start time.	13,14
VMLIVMIG7	The VF SHALL generate an event at VM migration completion time.	13,14
VMLIVMIG8	The VF SHALL support definition of the maximum time allotted for VM migration.	
VMLIVMIG9	The VF SHOULD enable best-effort preservation of QoS during VM migration.	
VMLIVMIG10	The VF SHALL be able to perform simultaneous migration of two or more VMs.	

4. SUMMARY

This document has presented virtualization requirements that have been identified by the SCOPE Alliance Virtualization Working Group as being necessary for the NEPs/TEMs networking/telecommunications systems. These virtualization requirements have been associated with the use cases from which they were derived and have been aggregated into categories. The higher priority requirements comprise 86 requirements that fall into the 10 categories listed below, each with the indicated numbers of requirements:

- VM Definition – 9 requirements
- VM Activation – 2 requirements
- VM Operation – 5 requirements
- VM Fault Handling – 2 requirements
- VM Administration – 5 requirements
- VF Deployment – 33 requirements
- VF Operation – 6 requirements
- Security – 1 requirement
- Performance – 19 requirements
- Availability – 3 requirements.

In addition to the above categories, the following category of lower priority requirements was identified, with the indicated number of requirements.

- Live Migration – 10 requirements.

The SCOPE Alliance Virtualization Working Group intends that the virtualization requirements presented in this document will be used by the ecosystem providers in delivering carrier grade virtualization features and solutions for the NEPs/TEMs.