virtualLogix™

Real-Time Virtualization™

WHITE PAPER

Carrier Grade Virtualization

Leveraging virtualization in Carrier Grade Systems

Abstract

Network Equipment Providers (NEPs) have been building networking infrastructure equipment able to deliver “carrier grade” services, typically mission-critical services such as voice telephony. In decades past, NEPs have achieved high degrees of availability through purpose-built hardware and software implementations. Today they increasingly build on COTS (Commercial Off The Shelf) hardware and Open Source Software (OSS), freeing their engineering resources to focus on core telephony competencies. The move to COTS and OSS requires that these hardware and software components be available from an ecosystem of suppliers, and that they interoperate seamlessly. Bodies such as The Linux Foundation (LF), the Service Availability Forum (SA Forum) and PICMG have defined standards and specifications such as carrier grade OSes (CGLinux), Service Availability Forum APIs and AdvancedTCA hardware to target carrier grade applications.

Recent advances have made virtualization appealing for carrier class equipment by permitting significant cost reduction through consolidation of workloads and of physical hardware. Virtualization also transparently lets NEPs and other OEMs (Original Equipment Manufacturers) leverage multi-core processors to run legacy software designed for uniprocessor hardware. However, virtualization needs to meet specific requirements to enable network equipment deploying this technology to meet industry expectations for carrier grade systems.

This white paper introduces the concept of “Carrier Grade Virtualization” (CGV) – a strategic software component whose integration into carrier grade platforms must preserve existing carrier grade performance and attributes.

CGV targets more than core network equipment and can be applied to a much wider range of equipment types (network appliances, firewalls, gateways, mid-tier routers, PBXs, office equipment, etc.) and lets OEMs realize benefits of higher availability in an *evolutionary* fashion, without need for *revolutionary* re-architecture of hardware and software platforms.

Introduction

Networking infrastructure equipment enjoys high reliability by architecting for high availability (HA) and deploying a mix of commercial off-the-shelf (COTS) hardware and commercial and open source software (OSS) components. In particular, carrier grade, hardware and software components, such as ATCA platforms and versions of the Linux OS have helped revolutionize the design and integration of highly available core and edge equipment.

The advent of multicore chipset designs and embedded virtualization has also turned a new page in system design practices, rendering silicon more efficient and lowering costs for higher performance commodity hardware but has also presented new challenges to traditional carrier grade middleware in meeting the requirements of next generation platforms.

This white paper introduces a new but evolutionary concept -- “Carrier Grade Virtualization” (CGV). Carrier Grade Virtualization defines the carrier grade properties expected from a virtualization platform to enable its deployment in carrier grade applications. Operationally, CGV aims to avoid incremental while enhancing the availability and other carrier grade attributes of the overall system.

Defining Carrier Grade Virtualization is an incremental process. As a precedent, consider that Carrier Grade Linux (CGL) specifications have iterated through 4 revisions, and are now influenced by profiles published by the Scope Alliance. As with CGL, Carrier Grade Virtualization aims to be a high-level definition, mostly independent of any given implementation.

The last section illustrates how VirtualLogix VLX for Network Infrastructure and vHA products start to address Carrier Grade Virtualization properties thus making it a concrete notion.

Definitions: Carrier Grade, High Availability and Virtualization

Defining Carrier Grade

It is useful to establish a shared understanding of core carrier grade terminology and instructive to examine the genesis of requirements addressed by Carrier Grade Virtualization. The following list summarizes the characteristics defined by the SCOPE Alliance[1]:

- From 5 to 7 nines of availability.
- High-performance scaling with the amount of hardware, and supporting large number of transactions and simultaneous sessions.
- Small and controlled error recovery domains supporting effective error isolation.
- Real-time behavior for signaling and call control domains.
- Failover capabilities to let user sessions survive failures.
- Hardware and software upgrade and error correction capabilities with minimal service interruption.
- Configurable security to provide high level of safety and protection.

[1] SCOPE is an industry alliance whose mission is to help, enable and promote availability of open carrier grade base platforms based on COTS)hardware / software and FOSS building blocks, and to promote interoperability.

- Controlled and extended life cycle of hardware and software components.
- Efficient and uniform management interfaces.
- Cost effective operation and maintenance.
- Speedy application development and testing.
- Easy to learn and well-documented.

Note that carrier grade properties are defined for a full solution from hardware to application software, By extension components are termed “carrier grade” if they can be used to build carrier grade solutions. Carrier grade Linux is an example of such a component. Carrier grade Virtualization is labeled in the same spirit.

Defining and Measuring Availability

Although not the only characteristic of carrier grade systems, High Availability (HA) is one essential pillar. Availability is commonly expressed as the ratio of useful system uptime[2] to the total time in a given period, most often in one year. If an installation can tolerate a half day of downtime in the course of 365, then required availability equals 364.5/365 or 99.863%.

Suppliers rate systems offering high degrees of availability in terms of the number of “nines” supported. Highly available, Carrier Grade systems boast five or six nines.

Nines	Application	Up time %	Actual down time
2	Office equipment	99%	3 days, 15.6 hours
3	Most IT infrastructure	99.9%	8.76 hours
4	Internet infrastructure	99.99%	52 minutes 34 seconds
5	PSTN and other business critical	99.999%	5 minutes, 16 seconds
6	Carrier class core/edge	99.9999%	32.56 seconds

In the real world, downtime is expressed from historically and statistically derived values for Mean Time To Failures (MTTF). As important as downtime is the time needed to repair a fault – Mean Time To Repair (MTTR). Availability, then, is calculated as:

$$\text{Availability} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}$$

If a system or component offers 75,000 hours MTTF, and repairing or replacing it requires an average of 45 minutes, then calculated availability for that system equals 99.999%, or five nines.

Using this formula, it is easy to see how architects can enhance total availability by using more reliable hardware and software components – thereby increasing MTTF –, by avoiding service interruptions during system upgrades, and/or by reducing the duration and impact of faults – by decreasing MTTR.

Virtualization

Virtualization is a popular technology deployed across different domains ranging from Java Virtual Machines to hardware based virtualization through application virtualization and resource aggregation. In the context of

[2] It is important to qualify “useful uptime”, in that a computer system may be running normally but not providing the service for which it was deployed.

this white paper, the virtualization type of interest is the one depicted in The SCOPE Alliance Technical Position Paper as part of its reference architecture.

"Hardware-based" virtualization enables simultaneous execution of more than one workload or operating systems instance on a single hardware platform. This feat is achieved by running a hypervisor (also called Virtual Machine Monitor or VMM) directly over the CPU silicon (on "bare metal"). The hypervisor provides guest OSes with abstracted execution environments called "Virtual Machines" (VMs). Each VM is usually configurable in terms of memory, CPU, and physical and virtual devices it can access. Each VM is independent and isolated from other VMs collocated on the same physical platform.

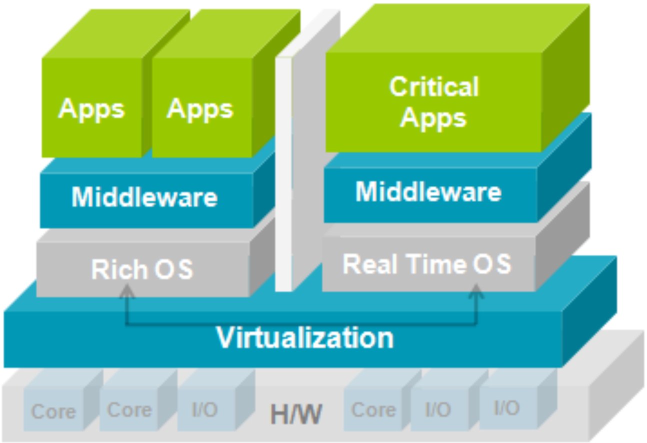

Figure 1. VirtualLogix Trusted Real-Time Virtualization

Virtualization finds frequent application in data centers and on the desktop, primarily on x86/Intel architecture CPUs[3] . More recently, virtualization has spread to different processors, and platforms including Power Architecture, ARM and DSP[4] based processors, cPCI[5] and ATCA[6] boards, and to domains such as network infrastructure equipments and mobile devices.

It is worthwhile to examine virtualization from a carrier grade point of view to determine which properties by virtualization match carrier grade expectations, and to draw perspectives in areas where virtualization could be enhanced to increase such properties to fit into carrier grade solutions as envisioned by the Scope reference architecture. In the absence of a formal definition of carrier grade properties of virtualization, it is worth taking a look at carrier grade operating systems and in particular at Carrier Grade Linux (CGL), which represents the most open and off-the-shelf example.

[3] Virtualization has been deployed for three decades on minicomputers and mainframes as well.

[4] Digital Signal Processor: A specialized digital microprocessor used to efficiently and rapidly perform calculations on digitized signals.

[5] CompactPCI: A standard for PCI industrial computers.

[6] Advanced Telecommunications Computing Architecture: A series of specifications regarding the design elements required for next-generation carrier-grade communications equipment.

Carrier Grade Linux

Starting in 2002, members of the Open Source Development Lab (now the Linux Foundation[7]) began an initiative to define requirements for a Carrier Grade version of the Linux OS (CGL), and later to vet and brand implementations of it. The Carrier Grade Linux definition has passed through multiple iterations: Version 4.0 was published in early 2007 and includes seven focus areas[8], as follows:

Availability Facilities for enhancing single node availability and recovery (e.g., monitoring free memory and CPU utilization).

Clustering Components to build clusters of individual systems (e.g., cluster management and cluster-level SA Forum APIs). Focus on high availability, with load balancing and performance as secondary aims.

Serviceability Features for servicing and maintaining a system and tools to support serviceability (e.g., IPMI[9] and HPI[10] chassis monitoring; power, fan, media, CPU and network).

Performance Features to sustain adequate system performance, especially at the base OS level (e.g., real-time responsiveness/latency, synchronization, communication).

Standards APIs, specifications, and standards necessary for CG operation (e.g. POSIX, IETF, SA Forum, AIS, AMF).

H/W Support Hardware-specific support for CG environments (Required ATCA, cPCI and BladeCenter platforms, chassis management, iSCSI, Rapid-I/O, etc.).

Security Individual features for building secure systems (access control, auditing, authentication, etc.).

We will return to these seven areas as a “lens” for focusing on Carrier Grade Virtualization.

[7] OSDL and its successor the Linux Foundation are non-profit organizations that support Linux kernel development and standardization of application APIs.

[8] Alliance CGL profile also played a role in the formation of the CGL 4.0 Requirements Definitions.

[9] Intelligent Platform Management Interface: A hardware level interface specification that defines a common, abstracted, message-based interface to platform monitoring and control functions.

[10] Hardware Platform Interface: An abstracted interface to managing computer hardware, typically for chassis and rack based servers.

Carrier Grade Virtualization

The topic of virtualization entered the ongoing CG ecosystem conversation in 2007 with the SCOPE Alliance introducing virtualization within its reference architecture.

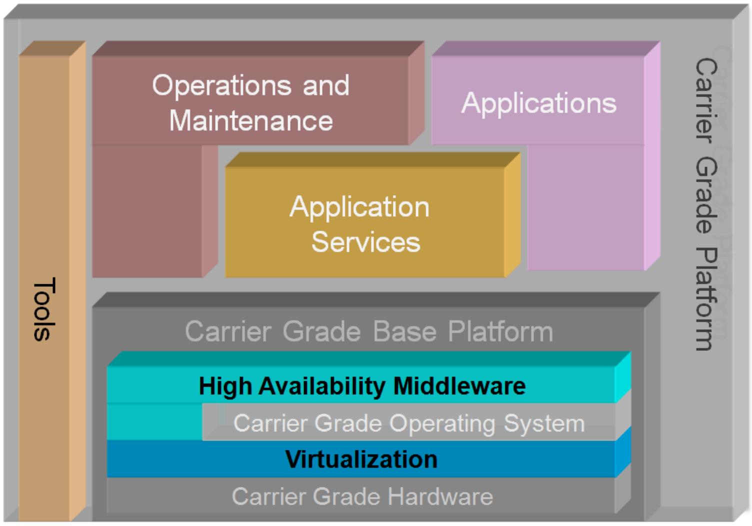
Operations and Maintenance
Applications
Application Services
Carrier Grade Platform
Tools
Carrier Grade Base Platform
High Availability Middleware
Carrier Grade Operating System
Virtualization
Carrier Grade Hardware

Figure 2. Simplified SCOPE Alliance Carrier Grade Platform architecture diagram

Defining Carrier Grade Virtualization

Carrier Grade Virtualization (CGV) may be defined as virtualization services that fulfill some or all expected properties existing in carrier grade solutions. Let's examine these properties and the requirements it imposes on Carrier Grade Virtualization. The list is not an exhaustive definition, but is comprehensive enough to provide an understanding of Carrier Grade Virtualization.

- **Availability:** Virtualization can increase overall availability of carrier grade systems, so that the 5 to 7 nines expected and provided by existing systems be preserved, or even enhanced (through lengthened MTTF or shortened MTTR).
- **High-Performance scaling:** Carrier Grade Virtualization must contribute to scalability, both through high efficiency with minimum overhead, and by enabling capabilities of new hardware, especially multicore processors.
- **Small error recovery domains:** Virtualization by itself isolates virtual machines and their guest OSes from each other; failure of a VM or a guest OS does not impact other collocated VMs. Carrier Grade Virtualization leverages this "natural" characteristic and extends it by dedicating VMs to specific functions such as providing device access.
- **Real-time behavior:** Since software components used in carrier grade systems do have real-time requirements, Carrier Grade Virtualization must be able to support real-time workloads and systems and so needs to offer real-time and deterministic properties.
- **Upgrade capabilities:** Virtualization changes the way software components (OSes, middleware, applications) are instantiated on a hardware platform. Carrier Grade Virtualization must ease software upgrades of such components, and must also support self upgrade.

- **Configurable security:** Virtualization isolates virtual machines and their workloads from one another. Carrier Grade Virtualization predicates more than "one size fits all" approach to security. CGV must offer capabilities to define and apply appropriate security policies among virtual machines and to control access by VMs to physical resources (CPU, memory, devices…).
- **Efficient and Uniform Management Interfaces:** Virtualization introduces new objects (virtual machines) which must be managed. Carrier Grade Virtualization must provide efficient and uniform management interfaces.

Let us examine these areas more in depth to understand the challenges and benefits of Carrier Grade Virtualization.

Availability

Traditional HA assumes a one-to-one correspondence between middleware, operating system and hardware, unaware of virtualization layer.

CGV manages traditional resources including virtual machines and the hypervisor.

Traditionally, highly available systems depend on middleware to enable application code to run in active / standby mode. HA middleware typically offers a range of redundancy models: 2N, N+1 and N+M. HA middleware architectures typically include) supervision agents which track errors and fault notifications, and trigger fail-over between active and standby application instances. Such middleware can monitor the health and functioning of practically single component participating in a carrier grade system, including hardware components as well. Traditional designs assumed that a single operating system supported a unique application instance on a given hardware board. As a result, failure of a physical board was corrected by performing a failover of the application running on the OS to a comparable stand-by instance. Conversely, OS failure triggered recovery by hardware reset of the underlying board.

Virtualization significantly changes this paradigm. The one to one correspondence among boards, OS and application is transformed by hardware and software consolidation induced by virtualization.

Carrier Grade Virtualization must accommodate this new situation: OS fault recovery entails resetting a virtual machine rather than a physical board. Moreover, just as HA middleware is central to Carrier Grade Linux implementation, Carrier Grade Virtualization must also integrate HA middleware. Such HA middleware must be able to manage new components introduced by virtualization: virtual machines and the hypervisor beneath them; dependencies among pre-existing components (physical boards, OS, middleware, applications) and new ones (hypervisor and virtual machines) must be carefully defined.

High-Performance Scaling

Application software and operating systems cannot scale well to large number of cores.

Embedded virtualization relieves OS and applications from multicore scaling burdens.

In the last three years, microprocessor design evolution has taken a new turn: the main focus is no longer on advancing clock speed but instead increasing circuit density and introducing multicore processors. Multi-processor systems are now the de facto common hardware platform, but legacy carrier grade software (applications, middleware and OSes) were designed for uniprocessor architectures.

While enhancing existing software to support multi-processor configurations is feasible, it is time consuming with uncertain results. Delivering an OS (and applications) to exploit two, four, eight, sixteen or more CPU cores is not a simple endeavor. Linux itself faces scaling challenges: version 2.4 of the kernel scaled to dual processor systems, but not beyond;. Linux 2.6 kernels scale well to four cores but degrade in performance with eight or sixteen.

Conversely, virtualization relieves OS and applications from explicit scaling burdens, presenting each guest OS and stack with uni- (or small number of) processor virtual machines. The simplicity of hypervisors allows them to scale easily to address increasing numbers of CPU cores .

A Carrier Grade Virtualization solution must also ensure that I/O access (disks, network interfaces) scales efficiently when the number of virtual machines increases on a physical platform.

Small Error Recovery Domains

Virtualization supports execution of independent virtual machines side by side on the same hardware. Failure of one virtual machine (or of the workload running on it) is contained and does not impact other VMs collocated on the same hardware.

Carrier Grade Virtualization leverages this intrinsic property by enabling specialization of VMs and dedicating them to run native device drivers and exporting (para)virtualized devices to other VMs. Isolation of device drivers in VMs limits the impact of a device driver failure to only virtual machines actually using a virtualized device.

While virtualization shrinks error recovery domains from a software standpoint, but it has no effect on hardware domains. Failure of a hardware domain still leads to the failure of several (small) software domains. However, smart configuration of HA middleware and applications failover can help spread induced load to several other hardware and software domains and not to a single backup hardware / software domain.

Real-time behavior

Carrier Grade Virtualization supports real-time software without altering its deterministic characteristics.

A Carrier Grade Virtualization component must be able to support legacy, proprietary real-time systems and workloads without compromising their real-time properties. This requires that the virtualization platform provide real-time scheduling of virtual machines and their loads to minimize interrupt latencies and behave in deterministic fashion.

In addition, when devices are virtualized it must be possible to assign Quality of Service (QoS) properties to VMs sharing access to underlying physical devices.

Upgrade Capabilities

In-service upgrade is essential to carrier grade equipment to minimize service interruption.

CGV allows the upgrade of an individual VM or a cluster of VMs as well as the hypervisor layer itself.

Carrier grade systems require capabilities for in situ upgrade without interruption of service. From a Carrier Grade Virtualization standpoint, this requirement has several aspects. CGV must support upgrading the VM contents (guest OS and applications) without impacting the overall system. CGV must support upgrading the virtualization layer itself without loss of service. Lastly, it must be possible to perform a global update of the (physical and virtual) cluster using appropriate mechanisms (split-mode upgrade, rolling upgrade, etc.).

Upgrading virtual machine content is a less strenuous task, as it relies on the same mechanisms as those used to upgrade physical (non-virtualized) machines. Upgrading the virtualization layer is really a special case of upgrading the system of a physical machine: as long as services running on the physical hardware can be failed over to redundant spare(s), the virtualization platform can be upgraded in situ followed by hardware reboot. Virtualization software upgrade and also cluster upgrade rely on enhanced management software that comprehends the presence of virtualization software on a given node.

Configurable Security

Configurable Security is a must-have for networking equipment. CGV security must protect critical resources from malicious access by other VMs.

In the context of virtualization, security issues can differ slightly from those on physical systems. For example, multi-tenants configurations entail sharing of physical network equipment by multiple operators (tenants), each having access to a subset of the physical equipment and being entitled to use equipment capacity up to a given level.

Carrier Grade Virtualization must provide each tenant with rights to administer a subset of the system while preventing them from accessing other tenants' resources or information flows. Overall physical platform management must also be handed over to a platform administrator and not to any of the tenants.

Efficient and Uniform Management Interfaces

Carrier Grade Virtualization introduces a new software layer and new objects in carrier grade systems. These objects (hypervisor, virtual machines, virtual devices…) require management and control. As equipment builders increasingly turn to COTS, such mechanisms must be based standards-based interfaces, especially to avoid vendor lock-in.

Table 2. Virtualization's impact on seven specification focus areas in Carrier Grade Linux

Carrier Grade OS Focus	Carrier Grade Virtualization Capabilities
Availability CG Virtualization provides an external perspective of guest OS health/resources and a generalized mechanism for managing guest OS lifecycle.	• Guest OS fault isolation • Fast VM / Guest OS reboot • Per VM Virtual watchdogs • Guest System upgrade
Clustering CGL clustering requirements focus on cluster management, not architecture. CGV augments the spec with virtual clustering.	• Virtual hosting of clustered systems • Independence from any single (proprietary) clustering scheme
Serviceability CGV extends monitoring and management to virtual nodes and elements; can remove dependencies on physical particulars of particular hardware type/integration.	• Monitoring, including VM life-cycle events • VM control (pause, resume, reboot, scheduling properties) • Dynamic VM management • Dumps, logs
Performance Real-time virtualization preserves performance attributes of legacy guest.	• Guest RTOS with deterministic, RT attributes • Native drivers • Low overhead
Standards Standards-compliance, while	• SAF AIS must be augmented with virtualization.

mainly an attribute of guest OSes and hosted m/w, can encompass APIs for managing VMs and virtual driver interfaces.	• OVF is a standard for virtual machine format. CGV must support it.
H/W Support CGV extends the notion of h/w support to include virtual nodes, blades, etc.	• CGV must be processor architecture agnostic • Transparent/efficient management / allocation of multi-core CPUs across VMs • Support of HPI / IPMI
Security CGV unifies diverse CGL security requirements and adds options for reducing TCB, enhancing isolation.	• Strong isolation of guest OSes • Physical and virtual resources access control • Secure boot and VM loading • Run-time integrity checking

Impacts and Benefits of Carrier Grade Virtualization

Carrier Grade Virtualization is not just a “nice idea”, it confers concrete benefits, both technical and financial.

Technical

On the technical side, CGV can positively impact fault granularity, detection and resolution, as well as overall security.

Scope and Impact of Faults

Purpose-built carrier-class systems take pains to minimize the granularity of failure so as to optimize fault detection, isolation, and resolution. Carrier Grade Virtualization offers at least as fine-grained fault resolution as purpose-built / integrated systems, and can actually go a step further.

- Integration of HA middleware preserves options for monitoring health of applications, hardware components, etc. and adds ability to monitor VM and hypervisor as managed objects
- Abstraction of device drivers, protocol stacks, etc. into lightweight VM contexts shrinks trusted computing base and helps to isolate and resolve faults in those components
- With CGV, most software faults are isolated in virtual machines. As a result, worst-case fault recovery entails resetting the containing VM; faster than physical board reset, it decreases MTTR.

Flexible Redundancy Models

This change in resolution and capability streamlines failover by

- Decoupling the redundancy of software fault domains and the redundancy hardware domains.
- Providing more choices for overall system architecture with more flexible sparing typology.

Real-time Support

Carrier Grade Virtualization must support legacy applications and OSes, many of which offer Quality of Service and real-time scheduling properties. As such, Carrier Grade Virtualization must

- Support multiple VMs with real-time workloads.
- Enable execution of general purpose OSes lacking real-time properties without degrading the behavior of real-time workloads running in other VMs.
- Guarantee that a VM won't face cycle starvation or be denied minimal access to available CPUs.

Enhanced Scalability

Carrier Grade Virtualization can address a greater range of applications than purpose-built HA systems by

- Targeting a wider range of embedded hardware, not just enterprise processors and blades, ATCA, etc.
- Enabling legacy OSes and workloads to run with scaling performance on multicore platforms without having to modify or tune them for that purpose.
- Supporting varying degrees of application integration with HA infrastructure to avoid force-fit of under-powered and overkill technology.

Improved Security

While specifications like Carrier Grade Linux specify a range of security mechanisms required by carrier class systems, they do so in a vacuum without consideration for viable system architecture context. Carrier Grade Virtualization offers a mix of implicit and explicit security improvements over most software deployed in purpose-built fault-resilient systems

- Strict isolation of all guest OSes and option to further isolate individual interfaces.
- More isolation / segmentation options for smaller Trusted Computing Bases (TCB).
- Scheduling of VMs and monitoring of CPU utilization in VMs / guest OSes to combat Denial of Service (DoS) attacks.

Financial

Financial benefits that derive from CGV are many and varied, and span the gamut from acquisition (capital expenditure) to streamlined performance and availability (operational expense):

CAPEX – Capital Expenditure

Compared to both legacy fault tolerant systems and to hardware-intensive highly available configuration of COTS hardware (CPCI, ATCA, BladeCenter, etc.), Carrier Grade Virtualization offers OEMs, integrators and service providers significant benefits:

- Run on both commodity blades and custom embedded hardware (including Power Architecture, ARM and DSP).
- Ability to provision with virtual spares improves hardware utilization, lowers cost of most configurations.
- Options for using commercial and open source HA middleware allows designers and integrators to find best-fit at best-cost for a given application.

OPEX – Operational Expenditure

Telecommunications and networking infrastructure systems are notoriously expensive to maintain and update. By designing with and deploying CGV, OEMs, integrators and end-deployers can optimize operational expenses through

- Better segmentation / smaller TCBs result in improved up-time and availability to meet stringent Service Level Agreements (SLA).
- Lower energy use from fewer active, hot and warm stand-by nodes.
- Better overall performance from more efficient multicore utilization and load balancing.

- More flexible hosting offers more plentiful and cost-effective upgrade options.

VirtualLogix Carrier Grade Virtualization

In line with industry standards, VirtualLogix is committed to providing Carrier Grade virtualization as outlined in the sections above. VirtualLogix introduces the first commercial implementation of a Carrier Grade Virtualization platform by building and delivering add-on virtualization components for embedded and real-time applications market. VirtualLogix CGV extends the merits of virtualization and high availability in a tightly integrated solution that is applicable to Network Equipment Providers and Telecom Equipment Manufacturers.

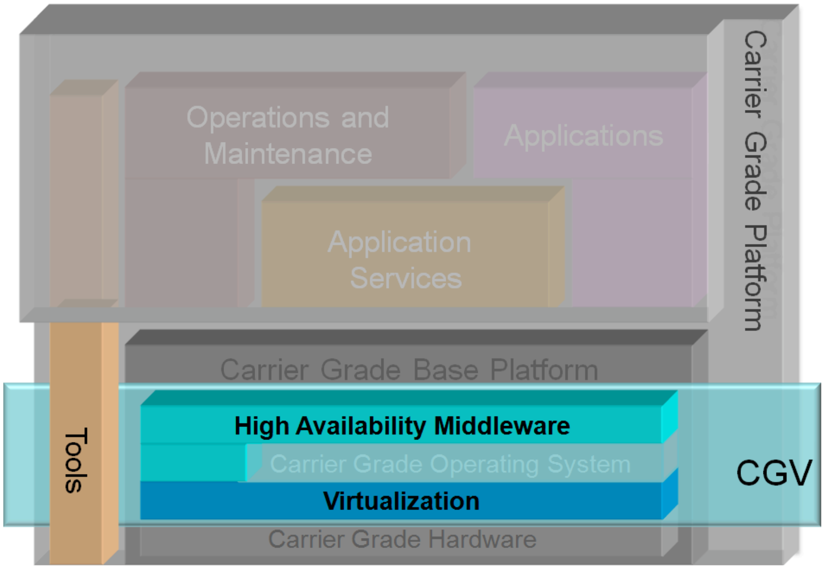
Carrier Grade Platform
Operations and Maintenance
Applications
Application Services
Carrier Grade Base Platform
Tools
High Availability Middleware
Carrier Grade Operating System
Virtualization
CGV
Carrier Grade Hardware

Figure 3. VirtualLogix VLX overlay on Carrier Grade Platform

VirtualLogix products include the following components of interest to Carrier Grade Virtualization:

- VLX for Network Infrastructure – the virtualization layer itself.
- vHA – a Virtualization enabled High Availability middleware.

VLX for Network Infrastructure

The VLX hypervisor is a bare metal virtualization layer which runs on many different processors, including Intel processors, Power architecture and DSP, in both single and multicore configurations. VLX for Network Infrastructure supports carrier grade platforms running on ATCA or blade-based and proprietary system hardware. VLX was designed to let real-time operating systems and workloads run neatly in virtual machines; as such it supports commercial real-time operating systems, legacy and proprietary carrier grade OSes as well as implementations of Carrier Grade Linux.

VLX also permits dedicating virtual machines to individual devices. As a result, I/O load can be spread across VMs to enhance overall I/O efficiency and scalability. Also, by encapsulating device drivers in different VMs device driver failure (a key source of faults) only disrupts a limited subset of applications, especially compared to traditional monolithic systems.

VLX for Network Infrastructure is integrated with VLX Developer, an Eclipse™ based suite of tools, to provide VM control and monitoring, including behavior of a VM and of its guest OS. This capability enables management software to monitor whether a VM or guest has reset a watchdog or is consuming a percentage of physical CPU resources. This monitoring information can be made available through a guest OS to an application running in a VM. This facility is used by the vHA component delivered by VirtualLogix.

vHA

VirtualLogix vHA provides fault recovery and system management services across virtual and physical domains. vHA features an open architecture supporting standard Service Availability Forum (SA Forum) software and interfaces. vHA scalability lets developers choose capabilities from hardware and virtual machines management to the restart of applications from a known state.

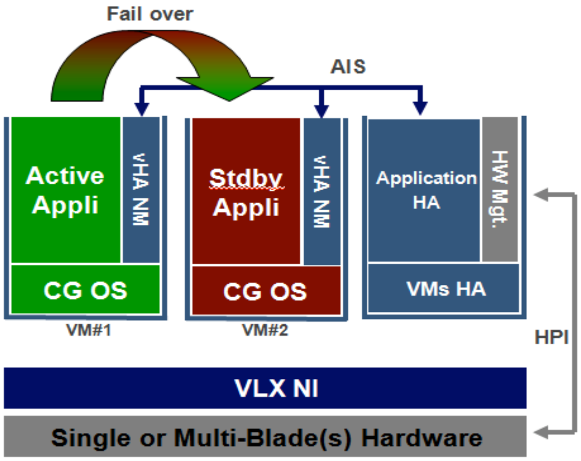

Figure 4. VirtualLogix Virtualization Enabled High Availability

vHA allow for three main deployable configurations:

Virtual Machine Controller

This configuration manages health of entire VMs and can work on single board computing systems. In this configuration, one VM is dedicated to run the vHA Manager (alone mode, without need for hardware or software redundancy.

Stateless Applications Controller

In addition to the service provided by the Virtual Machine Controller, this configuration can monitor health of applications lacking HA awareness, running on single or multiple boards. The vHA Manager may run alone or in redundant configuration (active/standby mode). The vHA Node Manager is installed in VMs where applications are monitored.

Stateful Applications Controller

In addition to the service provided by the Virtual Machine Controller and the Stateless Applications Control, this configuration can monitor health of HA-aware applications, providing services like checkpointing, events, messaging, logging or cluster membership. This configuration fits the needs of high-end equipment based on multi-board computing systems. The vHA Manager may run alone or in redundant (active/standby) mode. The vHA Node Manager must be installed in all the VMs with monitored applications.

For each possible configuration, hardware management using the Hardware Platform Interface (HPI) can complement the solution.

Conclusion

Carrier Grade Virtualization represents an important evolutionary step in the design and manufacture of networking equipment. This white paper has defined the concept of Carrier Grade Virtualization as a solution for a wide range of networking infrastructure applications. CGV is not presented as a panacea, but as a realistic set of requirements for, and capabilities of a platform that emerges from combining real-time virtualization and high availability middleware.

This white paper highlighted the virtualization platforms from VirtualLogix as real-world implementations of the emerging definition of Carrier Grade Virtualization. To learn more about VirtualLogix virtualization products, visit http://www.virtuallogix.com.

References

HADDAD, Ibrahim [2004]. “Towards Carrier Grade Linux Platforms”. In the *Proceedings of the USENIX Technical Conference*.

LINUX FOUNDATION, The [2008]. “Carrier Grade Linux 4.0 Specifications”.

SCOPE Alliance, The. [2008]. “SCOPE Technical Position Paper”.

SCOPE Alliance, The. [2008]. “Virtualization: State of the Art”.

SCOPE Alliance, The. [2008]. “Virtualization Requirements 1.0”.

SERVICE AVAILABILITY FORUM, The. [2008]. Applications Interface Specification 5.0”.

WEINBERG, William [2008]. “Achieving High Availability with Virtualization”. On *SearchServerVirtualization.com*