flexiant™

your cloud simplified

Common considerations when selecting your hypervisor

A review of four leading hypervisors and Flexiant Cloud Orchestrator

Contents

I. Introduction

When engaging with Flexiant, customers often ask us which hypervisor they should select. Flexiant Cloud Orchestrator is a hypervisor neutral cloud orchestration platform. By that we mean it is not tied to a particular hypervisor; the licensee can choose, subject to the software edition they have purchased, between any of our supported hypervisors. This deliberate policy of hypervisor neutrality allows our customers to make the right choice for their business, as well as support multiple hypervisors within the same deployment. Whilst we make every attempt to ensure the same feature set is available on all hypervisors, there are important differences in their integration.

This white paper sets out to help you select the appropriate hypervisor for your business and includes an analysis of how Flexiant Cloud Orchestrator integrates with each hypervisor.

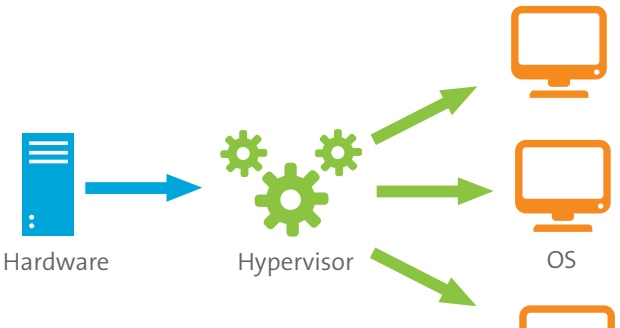
II. What does a hypervisor do?

A hypervisor is one of two main ways to virtualize a computing environment. By 'virtualize', we mean to divide the resources (CPU, RAM etc.) of the physical computing environment (known as a host) into several smaller independent 'virtual machines' known as guests. Each guest can run its own operating system, to which it appears the virtual machine has its own CPU and RAM, i.e. it appears as if it has its own physical machine even though it does not. To do this efficiently, it requires support from the underlying processor (a feature called VT-x on Intel, and AMD-V on AMD).

One of the key functions a hypervisor provides is isolation, meaning that a guest cannot affect the operation of the host or any other guest, even if it crashes. As such, the hypervisor must carefully emulate the hardware of a physical machine, and (except under carefully controlled circumstances), prevent access by a guest to the real hardware. How the hypervisor does this is a key determinant of virtual machine performance. But because emulating real hardware can be slow, hypervisors often provide special drivers, so called 'paravirtualized drivers' or 'PV drivers', such that virtual disks and network cards can be represented to the guest as if they were a new piece of hardware, using an interface optimized for the hypervisor. These PV drivers are operating system and (often) hypervisor specific. Use of PV drivers can speed up performance by an order of magnitude, and are also a key determinant to performance.

Type 1 and Type 2 hypervisors – appearances can be deceptive

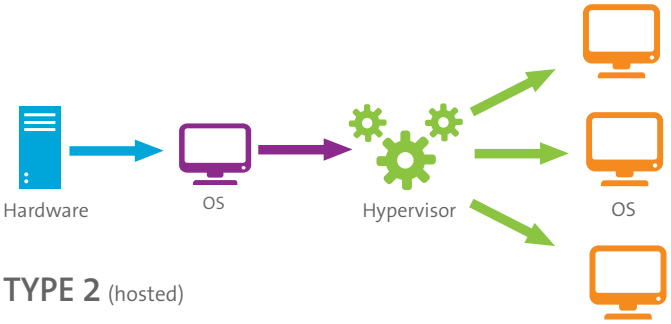
Hypervisors are often divided between Type 1 and Type 2 hypervisors.



TYPE 1 native (bare metal)

A Type 1 hypervisor (sometimes called a 'Bare Metal' hypervisor) runs directly on top of the physical hardware. Each guest operating system runs atop the hypervisor. Xen is perhaps the canonical example.

One or more guests may be designated as special in some way (in Xen this is called ‘dom-0’) and afforded privileged control over the hypervisor.



TYPE 2 (hosted)

A Type 2 hypervisor (sometimes called a ‘Hosted’ hypervisor) runs inside an operating system which in turn runs on the physical hardware. Each guest operating system then runs atop the hypervisor. Desktop virtualization systems often work in this manner.

A common perception is that Type 1 hypervisors will perform better than Type 2 hypervisors because a Type 1 hypervisor avoids the overhead of the host operating system when accessing physical resources. This is too simplistic an analysis. For instance, at first glance, KVM is launched as a process on a host Linux operating system, so appears to be a Type 2 hypervisor.

In fact, the process launched merely gives access to a limited number of resources through the host operating system, and most performance sensitive tasks are performed by a kernel module which has direct access to the hardware. Hyper-V is often thought of as a Type 2 hypervisor because of its management through the Windows GUI; however, in reality, a hypervisor layer is loaded beneath the host operating system.

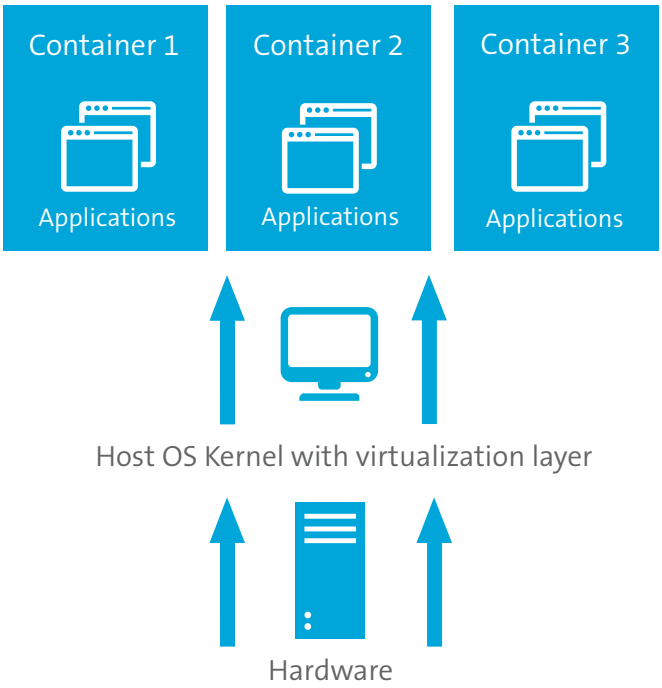
Another wrinkle is that the term ‘bare metal’ (often used to signify a Type 1 hypervisor) is often used to refer to a hypervisor that loads (with or without a small embedded host operating system, and whether or not technically a Type 1 hypervisor) without installation on an existing platform, rather like an appliance. VMware describes ESXi as a ‘bare metal’ hypervisor in this context. Flexiant Cloud Orchestrator’s deployment of both Xen and KVM also fit into this category: we PXEboot a tiny operating system image dedicated to the running of the hypervisor. However, both hypervisors could be installed in a conventional server environment.

Appearances can thus be deceptive.

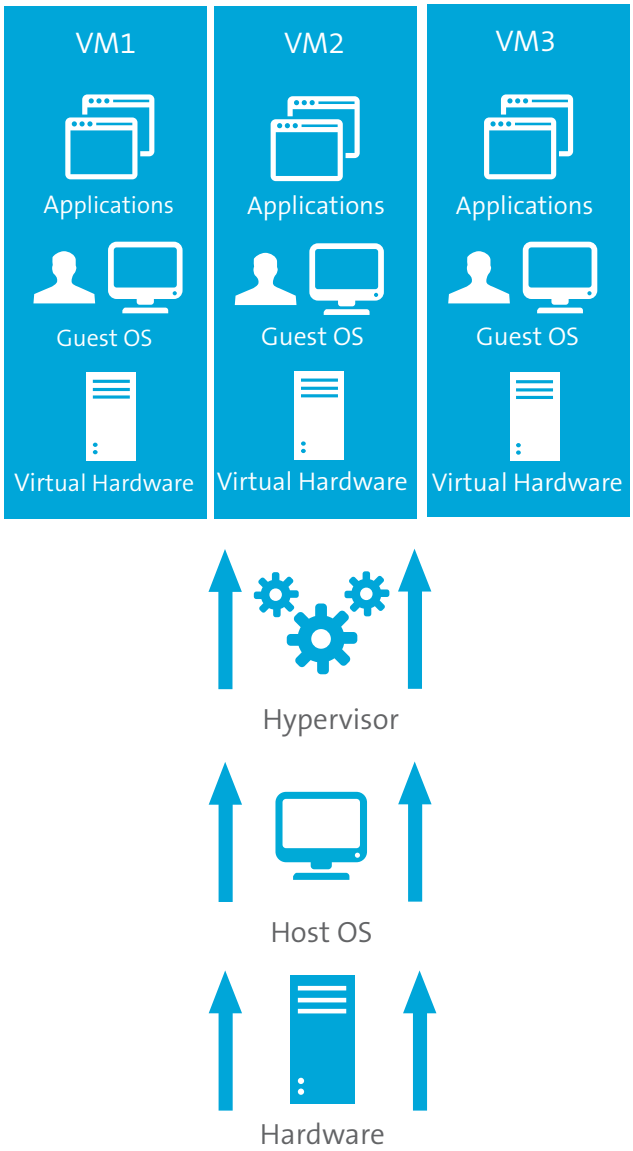
Hypervisors versus Containers

It is mentioned that a hypervisor was one of two main ways to segment a physical machine into multiple virtual machines; the other significant method is to use containers. A hypervisor segments the hardware by allowing multiple guest operating systems to run on top of it. In a container system, the host operating is itself divided into multiple containers, each running a virtual machine. Each virtual machine thus not only shares a single type of operating system, but also a single instance of an operating system (or at least a single instance of a kernel).

Virtualization using containers



Virtualization using hypervisors



Containers have the advantage of providing lower overhead (and thus increased virtual machine density), and are often more efficient particularly in high I/O environments. However, they restrict guest operating systems to those run by the host (it is not possible, for instance, to run Windows inside a container on a Linux operating system), and the isolation between virtual machines is in general poorer. Further, if a guest manages to crash its operating system (for instance due to a bug in the Linux kernel), this can affect the entire host, because the operating system is shared between all guests.

III. Considerations when selecting a hypervisor

Clearly from the above, the performance and maturity of the hypervisor are going to be important considerations. What are the others?

So far we've described the hypervisor as simply the software that segments the physical hardware to run workloads in the virtual machine. Technically that is true, but attached to the choice of hypervisor come four other considerations.

1 In Flexiant Cloud Orchestrator's case, the choice of hypervisor controls how the hypervisor is integrated. With Xen, KVM and Hyper-V, Flexiant Cloud Orchestrator communicates directly with the physical server (in Flexiant Cloud Orchestrator terminology, a 'node'), using an agent installed on the node. However, with VMware, we communicate with VMware's own management plane, which in turn communicates with the VMware hypervisor, ESXi – incidentally that's why we refer to the hypervisor as 'VMware' rather than ESXi. This means that VMware users can take advantage of the rest of the VMware ecosystem. However, it also means that VMware users must put in place the normal control stack associated with a VMware deployment with consequent implications for hardware requirements.

2 There are commercial implications to software choices. For instance, some hypervisor choices will require chargeable software licenses (VMware and Hyper-V), whereas KVM and Xen are open source and are included within Flexiant Cloud Orchestrator. VMware produced the first enterprise virtualization software and remains a market leader in that segment, and thus has a brand that is relevant to licensee's customers, whatever licensees may think of it. Microsoft gives favorable licensing terms to Windows operating systems running on Hyper-V.

3 Different hypervisors have different degrees of guest and functionality support. For instance, Hyper-V's support for Windows is (unsurprisingly) second to none. However, as KVM and Xen are more closely coupled to Flexiant Cloud Orchestrator, the variety of network support options is greater on these hypervisors. Apart from guest OS integration, the two main areas affected by hypervisor choice are network functionality and storage functionality.

4 There's the breadth and maturity of Flexiant Cloud Orchestrator's integration with the hypervisor concerned.

As a result, selecting a hypervisor is a multifactorial decision. After all, if one hypervisor always turned out best in all situations, we would not have provided support for four.

IV. Four hypervisors under review

KVM is a Linux based open source hypervisor. First introduced into the Linux kernel in February 2007, it is now a mature hypervisor and is probably the most widely deployed open source hypervisor in an open source environment. KVM is used in products such as Redhat Enterprise Virtualization (RHEV).

As KVM is open source and packaged as part of Flexiant Cloud Orchestrator, it is available under all Flexiant Cloud Orchestrator licenses at no additional cost to the licensee. We find it robust and reliable, and is our most heavily tested architecture. It supports the largest number of features of any hypervisor. This means it takes full advantages of all Flexiant Cloud Orchestrator's native storage and networking modes, including PVIP (see page 13).

However, it does not benefit from the wide integration to enterprise storage arrays that VMware does. It is also not a well-known brand amongst enterprise customers.

KVM is the most popular choice of our hosting customers who are typically concerned about cost per virtual machine and are less interested in enterprise style features.

KVM's place in the history of hypervisors

Whilst Flexiant's policy is to remain agnostic about hypervisors, we believe there are some specific considerations worthy of note concerning KVM and the history of hypervisor development.

When the VMware hypervisor started to gain market traction more than 10 years ago, it came with a large overhead as it was performing binary translation of each instruction; this overhead might consume almost half of the machine's CPU power.

Whilst VMware has long since addressed this issue, the elimination of this overhead was the main pain point that Xen helped to address. Xen also introduced para-virtualization, a way to make the guest kernel collaborate with the host kernel (thus being aware of each other) in order to minimize I/O overhead, bringing virtualization overhead down to 5-10%. However, despite its performance advantages, due to the requirement for a modified kernel images (again, an issue long since addressed), Xen was not fully accepted by the open-source community as a clean solution. Then came KVM.

KVM was born soon after Intel and AMD just released their extensions to allow the processor to virtualize guests without binary translation. KVM thus became a simple lightweight driver to make use of those instructions. If VMware's hypervisor was built using more than 6 millions of lines of code, and Xen reduced that to 500,000, the first stable version of KVM had little more than 10,000 lines. Of course, nowadays both Xen and VMware use the hardware extensions to virtualize, but Xen's original codebase has never been completely rewritten, leading to a more complex code base. As VMware is closed source, we can't comment on what happened there.

This history is why most Linux kernel developers will say KVM is “the way to do it”. This is because KVM was very quickly incorporated into the Linux kernel (whereas the Xen kernel modifications lived outside the kernel tree for many years), and by the fact that Red Hat, SUE and other distributors soon adopted KVM in preference to Xen. When Illumos (the free operating system derived from OpenSolaris) added support for hardware virtualization support, they chose to port KVM as opposed to any other hypervisor, because that was, in their words, the “right way to do it”.

Amazon and a few other cloud providers use Xen in a different way. Firstly, they use an old version of Xen (version 3), forked and adapted and optimized internally for their use case. Secondly, they use Xen in paravirtualized (PV) mode, which requires running Xen specific (and indeed Xen PV specific) kernels. This limits interoperability. This is for historical reasons. Xen hardware virtualization mode (HVM), which produces a guest that looks like conventional hardware (as does KVM, VMware and Hyper-V) was very new at the time EC-2 launched. This difference in codebase and in virtual machine configuration produces a cost overhead for the licensee and for end users.

KVM is open-source and as such has the potential to reduce the overall cost of virtualization. IBM did some calculations on the cost of running KVM against VMware and Microsoft, showing a 39% reduction of TCO when using KVM*, though clearly TCO is dependent on workload and the operational setting.

Whilst choice of hypervisor depends on a multitude of factors, for these reasons we believe KVM will often find a place on at least one cluster of the service provider’s Flexiant Cloud Orchestrator deployment.

* Reference to content: http://www-03.ibm.com/systems/virtualization/kvm/ and https://events.linuxfoundation.org/images/stories/pdf/lcna_co2012_hsu.pdf

VII. Considerations for hypervisor selection

	KVM	Xen4	VMware	Hyper-V
Hypervisor Features & Versions				
Hypervisor Version	3.8.0-32 / 1.0	4.3	VMware vSphere ESXi (5.0 / 5.1 / 5.5)	Hyper-V 2012 Datacenter Edition
Hypervisor provided by	Flexiant		Customer	Customer
Storage controlled by	FCO		vSphere, orchestrated by FCO	Hyper-V, orchestrated by FCO
Network controlled by	FCO		vSphere, orchestrated by FCO	Hyper-V, orchestrated by FCO
FCO Network Matrix				
Networking modes				
PVIP	✓		✗	✗
VLAN modes (public, private, interworking)	✓		✓	✓
Routing protocol between node and upstream router(s)				
Static+ARC	✓		✓	✓
BGP	⚙		⚙	⚙
OSPF	⚙		⚙	⚙
FCO Storage Matrix				
NFS v4 NAS	✓		✓	✗
Nexenta SAN (iSCSI)	✓		✓	✗
Shared SAN (iSCSI)	✓		✗	✗
Local storage	✓		✗	✗
Distributed Storage (Ceph)	⚙		✗	✗
Other distributed storage (e.g. Gluster)	⚙		✗	✗
Windows Server	⚙		Unknown	✓
iSCSI / Fibrechannel SAN (other than 'Shared SAN')	⚙		✓	✗
Anything else vSphere supports for storage	✗		⚙	✗

Out the box ✓ Requires configuration ⚙ N/A ✗

Note that the customer's own internal routing protocol is irrelevant. This is because the routing protocol running between the node and the upstream router should have no connection to the routing protocol used in the remainder of the customer's network.

VIII. About Flexiant

Flexiant provides cloud orchestration software focused solely to the global service provider market. Flexiant Cloud Orchestrator is a cloud management software suite that arms service providers with a customizable platform to help them turn innovative ideas into revenue generating services quickly and easily. With Flexiant, service providers can generate more revenue and accelerate growth, compete more effectively and lead the market through innovation. Vendor agnostic and supporting multiple hypervisors, Flexiant Cloud Orchestrator offers a customizable platform, a flexible interface, integrated metering and billing, reseller capabilities and application management. Flexiant gives service providers the ability to develop, launch and bill for new cloud services quickly.

Flexiant has been named a Gartner Cool Vendor in Cloud Management, received the Info-Tech Research Group Trendsetter Award and called an industry double threat by 451 Group. Flexiant is now a **Dell certified technology partner**. Customers include **Computerlinks, ITSONNET, FailProof Technologies, ITEX** and **NetGroup.** Flexiant is also a key participant in the **FP7 Consortium**.

For more information visit **www.flexiant.com**.