Intro on Hypervisors

This Book is a Guide on Hypervisors

What is a Hypervisor

This section is a small guide on Hypervisors

Welcome

Welcome to the Educational IT Knowledge Base! In this section, we will provide you with a comprehensive understanding of the concept of a "Hypervisor." This page covers everything from the definition and types of hypervisors to their key features, popular solutions, use cases, management best practices, and future trends. Let's dive in!

Table of Contents:

Use Cases of Hypervisors

- Data Centers and Server Consolidation
- Software Development and Testing
- Cloud Computing
- Desktop Virtualization

Choosing the Right Hypervisor

- Considerations for Selection
- Licensing and Cost
- Integration with Existing Infrastructure

Best Practices for Hypervisor Management

- Performance Optimization
- Backup and Disaster Recovery
- Security Best Practices

Future Trends in Hypervisor Technology

- Containerization and Hypervisors
- Unikernels and Microvisor

Conclusion

- Recap of Hypervisor Concepts
- Importance in Modern IT Environments

1. Introduction to Hypervisor

Definition and Purpose: A hypervisor is a software layer that allows the creation and management of multiple virtual machines (VMs) on a single physical computer or server. It acts as an intermediary between the physical hardware and the virtual machines, providing each VM with a virtualized environment that emulates the underlying hardware components, including CPU, memory, storage, and networking devices.

How Hypervisors Work: Hypervisors manage the allocation of physical resources to virtual machines, ensuring that each VM operates independently and securely. They schedule the usage of CPU time, memory, and other resources, preventing conflicts between VMs and optimizing overall system performance.

Benefits of Using Hypervisors:

- Server Consolidation: Hypervisors enables the consolidation of multiple servers onto a single physical machine, reducing hardware costs and power consumption.
- Isolation: VMs are isolated from each other and the host system, ensuring that failures or issues in one VM do not affect others.
- Resource Allocation: Hypervisors allow dynamic allocation of resources to VMs, enabling efficient utilization of hardware.
- Testing and Development: Virtualization simplifies software testing and development by providing isolated environments for experimentation.
- High Availability: Hypervisors support features like live migration, which facilitates moving VMs between physical hosts without downtime, ensuring continuous operation.

2. Types of Hypervisors

Type 1 Hypervisor (Bare-Metal Hypervisor): Type 1 hypervisors run directly on the physical hardware and do not require an underlying operating system. They have direct access to hardware resources, making them more efficient and suitable for enterprise-level virtualization deployments.

Type 2 Hypervisor (Hosted Hypervisor): Type 2 hypervisors run on top of a host operating system. They are typically used for desktop virtualization and are easier to set up and use for testing and development purposes.

3. Key Features of Hypervisors

Hardware Virtualization: Hypervisors abstract physical hardware, presenting virtual hardware to each VM, enabling the use of multiple operating systems and applications on a single physical machine.

Resource Management: Hypervisors efficiently allocate CPU, memory, storage, and networking resources to VMs, ensuring fair distribution and optimal utilization.

Isolation and Security: Hypervisors provide strong isolation between VMs, preventing one VM from accessing or affecting the data and processes of another, enhancing security and stability.

Snapshot and Cloning: Hypervisors offer snapshot and cloning features, allowing administrators to create point-in-time backups of VMs or duplicate VM instances for testing or scaling purposes.

Live Migration: Hypervisors support live migration, enabling VMs to be moved between physical hosts with little to no downtime, facilitating load balancing and hardware maintenance.

4. Popular Hypervisor Solutions

VMware vSphere/ESXi: VMware vSphere, with ESXi as its hypervisor component, is a leading enterprise virtualization solution known for its robust features, scalability, and management capabilities.

Microsoft Hyper-V: Hyper-V is Microsoft's hypervisor, integrated into Windows Server and available as a standalone product, offering comprehensive virtualization capabilities.

KVM (Kernel-based Virtual Machine): KVM is an open-source Linux hypervisor that leverages the Linux kernel for virtualization, providing a cost-effective and flexible virtualization solution.

Xen: Xen is an open-source hypervisor that supports both paravirtualization and hardware virtualization, widely used in cloud computing environments.

5. Use Cases of Hypervisors

Data Centers and Server Consolidation: Hypervisors are extensively used in data centers to consolidate multiple physical servers onto a single host, optimizing resource utilization and reducing hardware costs.

Software Development and Testing: Hypervisors provide isolated environments for software developers and testers to work with different operating systems and test their applications in controlled settings.

Cloud Computing: Cloud providers use hypervisors to create and manage virtual machines, enabling customers to deploy applications in the cloud without worrying about the underlying infrastructure.

Desktop Virtualization: Hypervisors enable virtual desktop infrastructure (VDI), allowing multiple virtual desktops to run on a single physical machine, offering centralized management and enhanced security.

6. Choosing the Right Hypervisor:

Considerations for Selection: Factors to consider when choosing a hypervisor include performance, scalability, management capabilities, and compatibility with the existing infrastructure.

Licensing and Cost: Different hypervisor solutions come with varying licensing models, and it's essential to evaluate the costs based on your organization's needs and budget.

Integration with Existing Infrastructure: Ensure the selected hypervisor seamlessly integrates with your network, storage, and management systems to avoid compatibility issues.

7. Best Practices for Hypervisor

Management: Performance Optimization: Implement best practices to optimize the performance of VMs and hypervisor hosts, including proper resource allocation and periodic performance monitoring.

Backup and Disaster Recovery: Set up regular backups of VMs and establish disaster recovery plans to ensure data integrity and business continuity.

Security Best Practices: Secure the hypervisor environment by employing measures like network segmentation, access controls, and timely patching to mitigate potential security risks.

8. Future Trends in Hypervisor Technology

Containerization and Hypervisors: The integration of containers and hypervisors allows organizations to leverage the benefits of both technologies for application deployment and management.

Unikernels and Microvisor: Unikernels are specialized, single-address-space machine images that run directly on a microvisor, reducing the attack surface and improving performance for specific applications.

9. Conclusion

In conclusion, hypervisors play a vital role in modern IT environments, enabling efficient resource utilization, enhanced security, and flexible management of virtualized infrastructure. Whether it's server consolidation, software development, cloud computing, or desktop virtualization, hypervisors continue to be at the forefront of transforming IT operations.

We hope this knowledge base has provided you with a comprehensive understanding of hypervisors and their significance in the world of virtualization and IT management. Should you have any further questions or require additional assistance, feel free to explore other sections of our Educational IT Knowledge Base or seek expert advice from IT professionals. Happy virtualizing!

10. Hypervisor Documentation

VMware vSphere/ESXi:

- Documentation: https://docs.vmware.com/en/VMware-vSphere/index.html

Microsoft Hyper-V:

- Documentation: https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/hyper-v-on-windows-server?view=windows-server-2022

KVM (Kernel-based Virtual Machine):

- Documentation: https://www.linux-kvm.org/page/Documentation

Xen:

- Documentation: https://xenproject.org/help/documentation/

XCP-ng Documentation:

- XCP-ng:
  - Documentation: https://xcp-ng.org/docs/

Please note that the provided links are based on the information available up to September 2021. It's always a good practice to visit the official websites of the respective technologies to ensure you have the most up-to-date and accurate documentation.

General Hypervisor How-To's

How to identify plugged-in SFP module model, manufacturer, and hardware details

Summary:

Identify or verifying SFP/SFP+ transceiver hardware details for hypervisor cli, such as; vendor, model, product code, transceiver type, cable length, and more without physically disconnecting it for visual inspection.

Versions affected:

ALL Hyper-V Versions, ALL ESXI Versions, ALL AHV Versions

Description:

An SFP/SFP+ (Small Form-factor Pluggable) module or transceiver is the physical hardware component that terminates the physical network media and provides the Layer 1 interface to a physical NIC installed in a host. For example, the SFP(+) module may interface SX/LX/SM/MM types of optical fiber, or varying lengths of Twinax copper cabling, between the host and network switch. The host NIC and switch don't need to worry about the physical media type, only the bits being transmitted as 1s and 0s. Whether that is done as pulses of light at varying wavelengths, over solid glass fiber, strands of plastic fiber, or over shielded twisted bits of copper as electrical pulses, the NIC and switch don't care ... that's the SFP transceivers problem!

In certain circumstances, such as troubleshooting or during maintenance and network changes or upgrades, the hardware details of the SFP(+) module plugged into a host NIC may need to be identified and verified. Typically the useful information is printed on the SFP(+) label, however, if the SFP(+) transceiver plugged in and is in use, or is in a remote location where you are not, it may be impractical or impossible to pull it out for inspection.

To overcome the inability to physically inspect the SFP(+) module, certain hypervisor OS utilities can be used to poll and expose useful SFP(+) details via console/CLI.
Some useful SFP(+) module information may include: Vendor, model, part number, serial number, transceiver type, cable length, connector type, signal quality, and more.

Solution:

AHV

```
ethtool -m <interfaceX>
```

> Example usage and output from an Arista Twinax (1 x 40g -> 4 x 10g) breakout copper SFP+:
> (where "ethX" is the specific NIC you want to poll, such as "eth2")

```
[root@AHV ~]# ethtool -m ethX
    Identifier                          : 0x03 (SFP)
    Extended identifier                      : 0x04 (GBIC/SFP defined by 2-wire interface ID)
    Connector                             : 0x21 (Copper pigtail)
    Transceiver codes                         : 0x01 0x00 0x00 0x00 0x00 0x04 0x80 0x00
    Transceiver type                         : Infiniband: 1X Copper Passive
    Transceiver type                         : Passive Cable
    Transceiver type                         : FC: Twin Axial Pair (TW)
    Encoding                              : 0x06 (64B/66B)
    BR, Nominal                            : 10300MBd
    Rate identifier                        : 0x00 (unspecified)
    Length (SMF,km)                           : 0km
    Length (SMF)                            : 0m
    Length (50um)                            : 0m
    Length (62.5um)                           : 0m
    Length (Copper)                          : 5m
    Length (OM3)                            : 0m
    Passive Cu cmplnce.                        : 0x00 (unspecified) [SFF-8472 rev10.4 only]
    Vendor name                              : Arista Networks
    Vendor OUI                             : 00:1c:73
    Vendor PN                          : CAB-Q-S-5M
    Vendor rev                         : 20
    Option values                       : 0x00 0x00
    BR margin, max                        : 0%
    BR margin, min                       : 0%
    Vendor SN                          : xxxxxxxxxx
    Date code                          : 180907
```

Vmware ESXi

Unfortunately, there does not appear a way to poll the SFP transceiver module from an ESXi host. Whilst "*ethtool*" exists, it is limited and does not allow the "*-m*" switch to poll the SFP 'module eeprom'.

Many similar functions exist in "*esxcli*", such as "*esxcli hardware pci list*" and "*esxcli network nic get -n vmnicX*", however, relevant to the SFP they only provide Layer 1 speed and connectivity status but no transceiver-specific data.

XenServer

```
[root@XenServer ~]# ethtool -m ethX
```

Example usage and output from an Arista Twinax (1 x 40g -> 4 x 10g) breakout copper SFP+: (where "ethX" is the specific NIC you want to poll, such as "eth2")

```
[root@XenServer ~]# ethtool -m ethX
Identifier                              : 0x03 (SFP)
Extended identifier                       : 0x04 (GBIC/SFP defined by 2-wire interface ID)
Connector                               : 0x21 (Copper pigtail)
Transceiver codes                         : 0x01 0x00 0x00 0x00 0x00 0x04 0x80 0x00
Transceiver type                          : Infiniband: 1X Copper Passive
Transceiver type                          : FC: Copper Passive
Transceiver type                          : FC: Twin Axial Pair (TW)
Encoding                                : 0x06 (64B/66B)
BR, Nominal                              : 10300MBd
Rate identifier                          : 0x00 (unspecified)
Length (SMF,km)                           : 0km
Length (SMF)                             : 0m
Length (50um)                             : 0m
Length (62.5um)                           : 0m
Length (Copper)                           : 1m
Length (OM3)                             : 0m
```

```
	Passive Cu cmplnce.                   : 0x00 (unspecified) [SFF-8472 rev10.4 only]
	Vendor name                           : Arista Networks
	Vendor OUI                            : 00:1c:73
	Vendor PN                             : CAB-Q-S-1M
	Vendor rev                            : 20
```

Hyper-V

Unfortunately, there does not appear to be a *Powershell* or *cmd* utility available to poll and expose SFP+ vendor make/model/serial details. In certain circumstances, some SFP+ manufacturers may offer a 3rd party utility to install on the host and run to gather this info. Please reach out to the SFP+ manufacturer for further guidance if required.

Note that for ESXi and Hyper-V, if using '*Twinax*' SFP+ modules, then consider polling for the transceiver details from the network switch instead. Twinax cables are typically hard-wired with the same transceiver at each end by the manufacturer, so the network switch transceiver may shed light by assumption on the host-side module also. Consider using host NIC MAC addresses against the switch MACC address table to confirm which switch port to interrogate.
Even for fiber optic connections, the switch-side SFP module metrics may also provide certain useful signal health and connection speed/status/type information, stopping short though of confirming the vendor/model module at the remote host-end.

Cabling Issues

If ethtool returns error strings as shown below the SFP should be confirmed to be fully seated in the correct orientation. You should also validate on the NIC's Compatibility Matrix to ensure the cables have been tested and validated to work with the specific NIC in use.

```
[root@AHV ~]# ethtool -m eth4
Cannot get module EEPROM information: Input/output error


[root@AHV ~]# ethtool --show-fec eth4
Cannot get FEC settings: Invalid argument
```