DDM Inventory Virtualized Support

Where to find virtual data

HP Discovery and Dependency Mapping Inventory

Disclaimer

This paper assumes knowledge of SQL scripting and Database Mining for a MySQL database. There is many education avenues for working with SQL Statements and should be considered before attempting to create scripts from the DDM Inventory Database.

The following SQL statements can be modified to include other items that you wish to include in your report. The objective of the statements in this paper is to give ideas and provide a starting point in developing the custom report.

Additional training can be found at

http://www.mysql.com/training/

DDM Inventory Virtualized Support

DDM Inventory supports many different technologies in the Virtualized space. Part of that support is making a relationship of the virtual server that the physical host has configured. This support is only with VMware and Solaris. Below is a complete list of supported technologies.

DDMI 7.70 Supported Technologies

Virtualization Technology	Host Inventory	Scanner Detection	Host/VM Relationship Determination	Comments
Microsoft Virtual PC 2004/2007	Yes	Yes	No	Host and VMs need to be inventoried separately.
Microsoft Virtual Server 2005, 2005 R2	Yes	Yes	No	Host and VMs need to be inventoried separately.
Microsoft Windows Server 2008 Hyper-V	Yes	No	No	Host and VMs need to be inventoried separately.
Microsoft Hyper-V Server	No	No	No	VMs need to be inventoried separately.
VMware Workstation 6,7, VMware Player 2.5, 3	Yes	Yes	No	Host and VMs need to be inventoried separately.
VMware ESX/ESXi 3.5, 4.0	Yes	Yes	Yes (through web services interface)	VMs need to be inventoried separately. The VMware host can be inventoried remotely through a web service interface. HP does not recommend installing a Linux agent into the VMWare ESX host.
VMware VirtualCenter 2.5	N/A	N/A	Yes (through web services interface)	VMs need to be inventoried separately. Multiple VMware ESX servers managed by VirtualCenter can be discovered through its web services interface.
VMware vCenter Server 4.0	N/A	N/A	Yes (through web services interface)	VMs need to be inventoried separately. Multiple VMware ESX servers managed by VirtualCenter can be discovered through its web services interface.
Solaris 10 Zones	Yes	Yes	Yes (by the scanner)	Zones can be inventoried separately or global zone can be inventoried.
AIX LPAR	No	Yes	No	LPARs need to be inventoried separately.
HP-UX vPar	No	Yes	No	vPars need to be inventoried separately.
HP-UX nPartition	No	Yes	No	nPartitions need to be inventoried separately.

Development is actively looking for ways to improve and add features to the various technologies. What this paper will attempt to do is show you where the data is located for various aspect of using the virtual data.

Information Gathered by the Scanner

The DDM Inventory scanner now has support for collecting information about the Virtual Image running for IBM AIX LPAR and HP HPUX VPAR technologies. The virtual image must be scanned and processed by the DDM I database for the relationship to work. The information that is gathered from the Virtual device contains data that can be used to join the devices. The following is what is gathered and stored for each of the support technologies.

IBM AIX LPAR

The DDM Inventory Scanner will run a command inside the LPAR partition and collect the following information

```
Command: lparstat -i
List of attributes
Node Name  (Hostname)
Partition Name
Partition Number
Type
Mode
Entitled Capacity
Partition Group-ID
Shared Pool ID
Online Virtual CPUs
Maximum Virtual CPUs
Minimum Virtual CPUs
Online Memory
Maximum Memory
Minimum Memory
Variable Capacity Weight
Minimum Capacity
Maximum Capacity
Capacity Increment
Maximum Physical CPUs in system
Active Physical CPUs in system
Active CPUs in Pool
Unallocated Capacity
Physical CPU Percentage
Unallocated Weight
```

HP HPUX

With HPUX the scanner collects information from two places. The virtual image and the host. The scanner will detect that it's a Host or virtual system and run the commands below. The following is the commands that are executed and collected by the scanner.

```
Command: vparstatus -M

List of Attributes
ComplexName
ComplexSerialNumber
PartitionNummber
VparName
State
Attribute
KernelPath
```

```
KernelOpt
CpuMinMax
CpuBound
CpuUnBound
CpuBoundList
CpuUnBoundList
Memory
NumIO


Command: parstatus -C -M

List of Attributs
ComplexName
ComplexSerialNumber
CabinetNumber
HardwareLocation
ActUse
CPU
Memory
ConnTo
CoreCellCap
UseOnNextBoot
ParNum
```

Database Changes

A table has been created in the DDM I Database to store the Properties of the data collected with the above commands. This is named hwOSContainerProperties. We also have added some data to existing fields in the database for some of the more common information between the three UNIX technologies. The following fields are now being updated with HPUX and AIX information and will continue to update the Solaris Zone information.

hwField	General meaning	Values
column: hwOSContainers.hwOSContainerType	VM Container type	Constant: Solaris Zones/VPAR/LPAR
column: hwOSContainers.hwOSContainerName	Local name of VM	Zone name, vParName, Partition Name
column: hwOSContainers.hwOSContainerID	Local id of VM	Zone ID / PartitionNumber / Partition Number
table: hwOSContainerProperties	Extra container information	properties captured from lparstat / vparstatus

Is it a physical or a virtual machine?

More and more we hear this question from our install base. With the vast amount of virtual technologies we are challenged on figuring out what is a physical system and what a virtual system is. DDMI does collect information on the type of virtual system. The location for the different technologies within the DDMI database is where it becomes challenging. For the most part we can identify what type of virtual system it is from one table in the database. The fields that we are using in the examples below are collected with the inventory process. Therefore the Inventory process must be run on all devices that you wish to collect Virtual data from. Not doing so will result in no data being returned.

Extracting the data

```
SELECT
Device.Device_PreferredIPAddress,
Device.Device_PreferredDNSName,
hwVirtualMachine.hwVirtualMachineType
FROM
Device
INNER JOIN hwVirtualMachine ON Device.Device_ID =
hwVirtualMachine.Device_ID AND ''= ''
GROUP BY
Device.Device_PreferredDNSName
order by device_preferredDNSname desc
```

This will produce a report that will identify all the devices that are part of a virtualized technology. While VMWare host servers are not part of this list, there are physical devices listed from the UNIX technologies. What we need to do now is filter those devices out by adding in an extra piece of information that only the Physical UNIX devices would have.

To make things easier you probably want to have a separate report for all the different virtualized technologies. With the above SQL statement we can add a where clause to filter out the devices that have the matching value of VMware under the field <hwVirtualMachineType>.

```
SELECT
Device.Device_PreferredIPAddress,
Device.Device_PreferredDNSName,
hwVirtualMachine.hwVirtualMachineType
FROM
Device
INNER JOIN hwVirtualMachine ON Device.Device_ID =
hwVirtualMachine.Device_ID AND ''= ''
WHERE
hwVirtualMachine.hwVirtualMachineType = 'VMWare'
GROUP BY
Device.Device_PreferredDNSName
order by device_preferredDNSname desc
```

Each virtual device needs to be inventoried in order for you to collect and report with the data above.

VMware ESX Technology

As mentioned in the table above DDMI supports native VMWare discovery and is not dependent on the inventory process to make the relationship. data from Host servers and Virtual Centers. In these cases we can see that there is a virtual server without the inventory server and to tell what is physical and what is virtual a new set of SQL Statements must be used to answer the question. By default DDMI will draw the relationship and produce a report for you in the reports section.

Solaris Zones Report

📄	Solaris Zones	Solaris global zones and the zones that they host [Export]

VMware Reports

📄	VMware Hosts	VMware host device details [Export]
📄	VMware Virtual Machines	VMware host devices and the VMs that they host [Export]
📄	VMware VirtualCenter	VMware VirtualCenter management servers and the VM hosts they manage [Export]

If there is a need to see more information that are considered virtual machines from a VMware ESX server, without having to do inventory, then the following tables could be used to create your own report.

Table	Location
Virtual Device	Aggregate.VirtualDevice
Host Servers	Aggregate.Hosts
Virtual Link	Aggregator.VirtualLink
Device Information	Aggregate.Device

The following example pulls out information about the IP Address and DNS Name for all the virtual devices within the Virtual Device table.

```
SELECT
Device.Device_PreferredDNSName,
VirtualDevice.VirtualDevice_Name,
VirtualDevice.VirtualDevice_OS,
VirtualDevice.VirtualDevice_VMPath,
VirtualDevice.VirtualDevice_VMUUID,
Device.Device_PreferredIPAddress,
Device.Device_Tag,
VirtualDevice.VirtualDevice_Status,
Device.Device_ID
FROM
VirtualDevice
INNER JOIN Device ON Device.Device_ID = VirtualDevice.Device_ID
ORDER BY
Device.Device_PreferredDNSName ASC
```

While the reports above will give you information on the relationship the above statement will pull out individual device information of each virtual device that DDMI has found in VMware servers. When Host information is needed we could use a different statement with the Host Servers table to pull out the same report about the Physical devices that host the virtual devices.

```
SELECT
Device.Device_PreferredDNSName,
Device.Device_PreferredIPAddress,
`Host`.Host_OS,
`Host`.Host_Type,
`Host`.Host_ServerType,
`Host`.Host_Status
FROM
Device
INNER JOIN `Host` ON Device.Device_ID = `Host`.Device_ID
ORDER BY
Device.Device_PreferredDNSName ASC
```

Solaris Technology

Along with the VMware features, DDMI will also make a relationship on the virtual devices in Solaris Containers. The feature that exists for Solaris relationship creation is done quite differently than with VMware. In Solaris we need to inventory the Host to gain knowledge of the virtual partition that make up the local zones. The Host server, generally known as the Global Zone, is scanned and the information produced is sent through the enrichment process. The inventory file is then split into several, inventory files depending on the number of partition, local zones, which were created.

```
SELECT
Device.Device_PreferredDNSName,
Device.Device_PreferredIPAddress,
hwOSContainers.hwOSContainerName,
hwOSContainers.hwOSContainerType
FROM
Device
INNER JOIN hwOSContainers ON Device.Device_ID = hwOSContainers.Device_ID
WHERE
hwOSContainers.hwOSContainerType = 'Solaris Zone'
```

The field hwOSConainerName will display the name of the individual zones. The devices marked with Global as the value will be the Host servers.

With some customers they want to scan the individual Local Zone instead of allowing DDMI to create the inventory record. Although this is a configurable option, it is not recommended by HP to run the inventory process with the following credentials as it breaks the relationship between the Host and Virtual server. Global/Local relationship.

The configuration items that need to change in this scenario are as follows
Administration > System Configuration > Scan Processing > Generate Solaris local zone inventory from the global zone – This option must be set to no to avoid the automatic generation.
Scanner Generator > Scanner Options > Miscellaneous > Terminate the scanner if the following environments are detected – You must deselect the option for Non Global Zones (Solaris), meaning when executing the inventory scanner the scanner will terminate when it sees a Local Zone.

Again this is not a recommended approach to Solaris discovery/inventory collection. If you have setup your DDMI Server to collect this type of data in this manner then you would need to work with the following information to create a report that describes the type of Solaris Virtual sever it is.

```
SELECT
hwOsData.hwOSHostOsCategory,
Device.Device_PreferredIPAddress,
hwVirtualMachine.hwVirtualMachineType,
hwBiosData.hwBiosSource,
hwBiosData.hwBiosMachineId,
hwBiosData.hwBiosSerialNumber,
hwOSContainers.hwOSContainerName,
hwBiosData.hwBiosMachineModel,
hwBiosData.hwBiosManufacturer,
```

```
Device.Device_PreferredDNSName,
hwOsData.hwHostOS
FROM
hwOsData
INNER JOIN Device ON Device.Device_ID = hwOsData.Device_ID
INNER JOIN hwVirtualMachine ON Device.Device_ID =
hwVirtualMachine.Device_ID
INNER JOIN hwBiosData ON Device.Device_ID = hwBiosData.Device_ID
INNER JOIN hwOSContainers ON Device.Device_ID = hwOSContainers.Device_ID
```

LPAR

We can use the same statement as above to pull out the devices that have
the attribute lpar and use this to build upon a relationship. For LPAR
information we need to gather many pieces to make the relationship work.
The script below uses a few pieces of information to determine what
devices are related together. Some fields are only informational data.
Using <hwBiosSerialNumber> we can identify devices that match this value.
We can then use this to match against <hwOSContainerName>. The 'host
server will have the matching value making it the key to creating the
relationship.

```
SELECT
hwOsData.hwOSHostOsCategory,
Device.Device_PreferredIPAddress,
hwVirtualMachine.hwVirtualMachineType,
hwBiosData.hwBiosSource,
hwBiosData.hwBiosMachineId,
hwBiosData.hwBiosSerialNumber,
hwOSContainers.hwOSContainerName,
hwBiosData.hwBiosMachineModel,
hwBiosData.hwBiosManufacturer,
Device.Device_PreferredDNSName,
hwOsData.hwHostOS
FROM
hwOsData
INNER JOIN Device ON Device.Device_ID = hwOsData.Device_ID
INNER JOIN hwVirtualMachine ON Device.Device_ID =
hwVirtualMachine.Device_ID
INNER JOIN hwBiosData ON Device.Device_ID = hwBiosData.Device_ID
INNER JOIN hwOSContainers ON Device.Device_ID = hwOSContainers.Device_ID
WHERE
hwOsData.hwOSHostOsCategory = 'UNIX' AND
hwVirtualMachine.hwVirtualMachineType = 'lpar'
GROUP BY
Device.Device_PreferredIPAddress
ORDER BY
hwBiosData.hwBiosSource ASC
```

HPUX

We can identify the difference between a Host and a Virtual server by
some of the information available. Making a relationship will be a little
different depending on your platform you are running the virtual
technology. Superdomes are not supported with inventory collection. We
can identify the virtual systems using the following statement and

relationship will be something that will need to be looked at to get the data correct.

```
SELECT
hwOsData.hwOSHostOsCategory,
Device.Device_PreferredIPAddress,
hwVirtualMachine.hwVirtualMachineType,
hwBiosData.hwBiosSource,
hwBiosData.hwBiosMachineId,
hwBiosData.hwBiosSerialNumber,
hwOSContainers.hwOSContainerName,
hwBiosData.hwBiosMachineModel,
hwBiosData.hwBiosManufacturer,
Device.Device_PreferredDNSName,
hwOsData.hwHostOS
FROM
hwOsData
INNER JOIN Device ON Device.Device_ID = hwOsData.Device_ID
INNER JOIN hwVirtualMachine ON Device.Device_ID =
hwVirtualMachine.Device_ID
INNER JOIN hwBiosData ON Device.Device_ID = hwBiosData.Device_ID
INNER JOIN hwOSContainers ON Device.Device_ID = hwOSContainers.Device_ID
WHERE
hwVirtualMachine.hwVirtualMachineType = 'vpar'
```

Summary

Virtualization is becoming more and more common to reduce cost of physical devices. With it brings challenges on how to control the licensing costs associated to it. DDMI helps collect and store information on the virtual environments for better management and provide cost savings for the software that is installed. Knowing where specific information exists helps create processes that will enable better management of virtual devices and create processes to charge back the use of those applications. This paper talked about where the data exist in DDMI and how to identify the virtual devices. The next revision of this paper will include information on creating relationships between the supported technologies.

The Objective us this paper is not to teach you how to write SQL Statements but rather give you a starting point of where the data exist in DDMI and how you might be able to use this data within other systems. It provides outputs that can be reviewed and manipulated to control the use of the virtualized technologies.

For more information

Please visit the HP OpenView support web site at:
http://www.hp.com/managementsoftware/support

This web site provides contact information and details about the products, services, and support that HP OpenView offers.

HP OpenView online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valuable support customer, you can benefit by being able to:

- Search for knowledge documents of interest
- Submit and track progress on support cases
- Submit enhancement requests online
- Download software patches
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

Note: Most of the support areas require that you register as an HP Passport user and sign in. Many also require an active support contract.

To find more information about support access levels, go to the following URL:

http://www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to the following URL:

http://www.managementsoftware.hp.com/passport-registration.html

Please visit the HP OpenView support web site at: