

## **Resource metering, a new feature in Hyper-V in Windows Server 2012.**

As you've probably heard by now, Windows Server 2012 is a great platform for the private cloud. When we began planning this release, we realized that one of the things you need in order to run a cloud is to be able to charge your users for the resources they use.

This is the need resource metering fills. It allows you to measure the resource utilization of your virtual machines. You can use this information as a platform for your own dynamic chargeback solutions, where you can charge customers based on the resources they use instead of a flat upfront cost, or to plan your hosting capacity appropriately.

There are four resources that you can measure: your CPU, memory, network, and storage utilization. We measure these resources over the period of time between when you measure and when you last reset metering.

**CPU (MHz):** We report the average utilization in megahertz.

Now, you're probably wondering why we don't report this as a percentage. After all, that's what we do in Hyper-V Manager. Well, we know that you like to move your virtual machines. With Windows Server 2012, you can live migrate your virtual machines all over the place. Naturally, the record of how much resources your virtual machine has used moves with it.

We want the virtual machine's CPU utilization to make sense across multiple machines. If we report a percentage and you move the virtual machine to a host with different processing capabilities, it's no longer clear what the percentage refers to.

Instead, we report in megahertz. For example, if your virtual machine had an average CPU utilization of 50% over the past billing cycle on a host with a CPU running at 3GHz, we would report 1500MHz.

If your virtual machine spent one hour on a host with a 3GHz CPU and used 50% and another hour on a host with 1GHz CPU and used 75%, we would report the following:

$$(3\text{GHz} * 1000\text{MHz}/1\text{GHz} * .5 * 1\text{hr}) + (1\text{GHz} * 1000\text{MHz}/1\text{GHz} * .75 * 1\text{hr}) = 2250\text{MHz-Hr}$$

Here I am converting the CPU capacity from GHz to MHz and figuring out how much of that capacity was used over each hour.

$$2250\text{MHz-Hr} / 2 \text{ Hours} = 1125 \text{ MHz.}$$

Then, I simply divide over the two hours to get this value.

One final note: we don't report minimum and maximum utilization values for CPU. If you think on it a moment, you'll come to the same realization we did: it is very likely that the minimum will be 0 and the maximum will be the full capacity of the hosts' CPU at some point over the timespan you're measuring. Since that's not very useful, we don't report it.

**Memory (MB):** We report the average, maximum, and minimum utilization here, in megabytes.

The minimum memory utilization captures the least memory used over the timespan measured. Since it's not very useful to know that the minimum memory usage was zero if the virtual machine was ever turned off, we only look at the minimum memory utilization when the virtual machine is running.

We do include the offline time of the virtual machine when calculating the average memory utilization. This provides an accurate view of how much memory the virtual machine was using over that billing cycle, so that you can charge your users accurately.

**Network (MB):** We report network utilization in megabytes. Of course, we want this metric to be useful, so we considered how you would want to see this information broken down. One way you might want to distinguish between network traffic is to see how much traffic is inbound to the virtual machine, and how much is outbound.

The most important breakdown you will want is how much traffic does the virtual machine send or receive from the internet, which costs you money, and how much is just communication between virtual machines you host, which costs you nothing since it is just using your intranet. With this breakdown, you can charge your user accurately for their internet usage.

So how do we provide these breakdowns? We use ACLs set on the virtual machine's network adapter. Each ACL has

- Direction
  - “Inbound” or “Outbound”
- Remote IP Address
  - The source or destination of the network packet, depending on direction
  - For example, 10.0.0.0/8
- Action
  - Allow, Deny, or Meter

These ACLs are used for more than just resource metering; note the Allow and Deny actions. For our purposes, you set the action to “Meter”.

Enabling resource metering creates two sets of default metering ACLs, provided none are already configured. The first set of ACLs, one inbound and one outbound, has a remote IP address of \*.\*; this wildcard notation indicates that it will meter all IPv4 traffic that is not covered by another ACL. The second set has an IP address of \*.\*:, which meters all IPv6 traffic.

With these metering ACLs, you can measure the total network traffic sent and received by the virtual machine, in megabytes. You can configure your own ACLs to count intranet traffic separately from internet traffic, and charge accordingly.

**Disk (MB):** As we spoke with customers, we realized that for chargeback purposes, they were only interested in the total disk allocation for a virtual machine. So, here we report that in megabytes.

The total value is the capacity (not the current size on disk) of the VHDs attached to the virtual machine plus the size of the snapshots. Take the following examples:

Fixed size disk:

VM with a single 100GB fixed size VHDs attached

-----  
Total Disk Allocation reported: 100GB

Dynamic disk:

VM with a single dynamic VHD attached,  
Current size 30GB, maximum size 100GB

-----  
Total Disk Allocation reported: 100GB

With snapshots:

VM with a single dynamic size VHDs attached,  
Current size 30GB, maximum size 100GB,  
Plus a 20GB snapshot

-----  
Total Disk Allocation reported: 120GB

Pass-through disks, DAS disks, guest iSCSI connections, and virtual Fibre Channel disks are not included in the total disk allocation metric.

Once you enable resource metering, Hyper-V will begin collecting data. You can reset metering at any time. We will then discard the data we have collected up to that point and start fresh. So, you will typically measure the utilization first, and then reset. When you measure, you are measuring the utilization over the timespan since you last reset metering. Metering is designed to collect this data over long periods of time. If you need greater granularity, you can look at performance counters.

Having resource metering enabled and just capturing utilization data per your billing cycle has no noticeable performance impact. There will be some negligible disk and CPU activity as data is written to the configuration file.