

Table of Contents

1 General remarks.....	2
1.1 Scope of this gatherer plugin	2
1.2 Memory-usage metrics.....	2
1.3 Long-term monitoring	2
2 Metrics associated to Xen_ComputerSystem instances.....	2
2.1 CPU related metrics	3
2.2 Memory related metrics.....	8
3 where “ <i>HostTotalMemory</i> ” is the size of physical memory available to the host.	
Appendix	11

Abstract

The [Common Information Model](#) (CIM)¹ is an object-oriented information model defined by the Distributed Management Task Force (DMTF), which provides a conceptual framework for describing management data.

Based on DMTF CIM Dynamic Metrics model, this document describes a first stage of metrics to be implemented for Xen. It is intended to be made a part of a future subprofile of the DMTF Virtual Server Model, for performance monitoring.

The metrics defined in this spec are implemented in open source as part of the SBLIM gatherer module. Please refer to <http://sblim.sourceforge.net/monitor.html> and <http://sourceforge.net/projects/sblim> for further details.

¹Common Information Model (CIM) Specification, V2.2, June 14, 1999,
http://www.dmtf.org/standards/standard_cim.php

1 General remarks

1.1 Scope of this gatherer plugin

This plugin is focused on metrics which are related to the Xen hypervisor. In the long run, this has to be abstracted to other virtualization platforms – like z/VM or PHYP – as well. The metrics of this plugin are associated to Xen_ComputerSystem resource class.

Metrics associated to the virtual server and the hosted operating system are not part of this model, but it might be useful to associate to those in-band implementations as well, metrics associated to e.g. Linux_OperatingSystem if using the SBLIM implementation.

1.2 Memory-usage metrics

For memory utilization, it would be useful to have a look at in-band metrics as well, like the Linux specific metric “ActiveMemory” in “/proc/meminfo”, or PageInRate as defined in eServer OS monitoring.

1.3 Long-term monitoring

As long-term monitoring is currently not implemented in SBLIM Gatherer infrastructure, we add two metrics beginning with “TenMinutes...” which will externalize the data based on 10-minute interval. All other metrics are based on SBLIM default gatherer interval.

1.4 Internal metrics (raw counters)

The metrics “_Internal_CPUTime” and “_Internal_TotalCPUTime” are not meant to be used directly but only for internal purposes of the gathering infrastructure. Those two metrics may go away in a future release or may change semantics. They are not documented in detail in this specification.

2 Metrics associated to Xen_ComputerSystem instances

The Xen_ComputerSystem instance represents the Virtual System.

The following metrics are associated to all instances of Xen_ComputerSystem (including domain 0).

2.1 CPU related metrics

<i>Name</i>	TotalCPUTime
<i>MeasuredElementClass</i>	Xen_ComputerSystem
<i>DataType</i>	uint64 (13)
<i>Calculable</i>	Summable (2)
<i>Units</i>	MilliSeconds
<i>IsContinuous</i>	TRUE
<i>ChangeType</i>	Gauge (4)
<i>TimeScope</i>	Interval (3)
<i>GatheringType</i>	Periodic (3)
<i>Based on existing eServer OS monitoring metric</i>	Yes

CPU time spent for the given domain / Xen_ComputerSystem for the reported interval. As it is expected that Xen behavior might change in future, the content of this metric is currently undefined if hyper-threading (SMT) is enabled.

This metric is based on the “*cpu_time*” attribute as displayed by “*xm list –long*”.

<i>Name</i>	TenMinuteTotalCPUTime
<i>MeasuredElementClass</i>	Xen_ComputerSystem
<i>DataType</i>	uint64 (13)
<i>Calculable</i>	Summable (2)
<i>Units</i>	MilliSeconds
<i>IsContinuous</i>	TRUE
<i>ChangeType</i>	Gauge (4)
<i>TimeScope</i>	Interval (3)
<i>GatheringType</i>	Periodic (3)
<i>Based on existing eServer OS monitoring metric</i>	No

Like *TotalCPUTime* metric, but with 10 minute interval length (that's the *Duration* attribute). This metric may go away in future after we have consistent implementation for long-term monitoring.

<i>Name</i>	ActiveVirtualProcessors
<i>MeasuredElementClass</i>	Xen_ComputerSystem
<i>DataType</i>	real32 (4)
<i>Calculable</i>	Non-Summable (3)
<i>Units</i>	Percent
<i>IsContinuous</i>	TRUE
<i>ChangeType</i>	Gauge (4)
<i>TimeScope</i>	Interval (3)
<i>GatheringType</i>	Periodic (3)
<i>Based on existing eServer OS monitoring metric</i>	Yes

Average number of virtual processors active in sampling interval.

This metric is based on the “*vcpus*” attribute as displayed by “*xm list –long*”.

<i>Name</i>	ExternalViewTotalCPUPercentage
<i>MeasuredElementClass</i>	Xen_ComputerSystem
<i>DataType</i>	real32 (4)
<i>Calculable</i>	Non-Summable (3)
<i>Units</i>	Percent
<i>IsContinuous</i>	TRUE
<i>ChangeType</i>	Gauge (4)
<i>TimeScope</i>	Interval (3)
<i>GatheringType</i>	Periodic (3)
<i>Based on existing eServer OS monitoring metric</i>	Yes

Percentage of CPU resources used by this domain. For example, if domain consumes on average 0.7 CPUs on a 4-way server (real cores, not SMP), this metric has a value of 17.5%.

If power management or hyper-threading are used, the content of this metric may be wrong.

This metric is derived from "TotalCPUTime" metric, "ActiveVirtualProcessors" metric and wall-clock time.

<i>Name</i>	TenMinuteExternalViewTotalCPUPercentage
<i>MeasuredElementClass</i>	Xen_ComputerSystem
<i>DataType</i>	real32 (4)
<i>Calculable</i>	Non-Summable (3)
<i>Units</i>	Percent
<i>IsContinuous</i>	TRUE
<i>ChangeType</i>	Gauge (4)
<i>TimeScope</i>	Interval (3)
<i>GatheringType</i>	Periodic (3)
<i>Based on existing eServer OS monitoring metric</i>	No

Like *ExternalViewTotalCPUPercentage* metric, but with 10 minute interval length (that's the *Duration* attribute). This metric may go away in future after we have consistent implementation for long-term monitoring.

2.2 Memory related metrics

<i>Name</i>	PhysicalMemoryAllocatedToVirtualSystem
<i>MeasuredElementClass</i>	Xen_ComputerSystem
<i>DataType</i>	uint64 (13)
<i>Calculable</i>	Non-Summable (3)
<i>Units</i>	KiloBytes
<i>IsContinuous</i>	TRUE
<i>ChangeType</i>	Gauge (4)
<i>TimeScope</i>	Point (2)
<i>GatheringType</i>	Periodic (3)
<i>Based on existing eServer OS monitoring metric</i>	No

Memory currently claimed by the given domain. Note that a domain may ask the hypervisor for more memory if under memory pressure.

This metric is based on the “*memory*” attribute as displayed by “*xm list –long*”.

<i>Name</i>	HostFreePhysicalMemory
<i>MeasuredElementClass</i>	PGS_ComputerSystem (Host System)
<i>DataType</i>	uint64 (13)
<i>Calculable</i>	Non-Summable (3)
<i>Units</i>	KiloBytes
<i>IsContinuous</i>	TRUE
<i>ChangeType</i>	Gauge (4)
<i>TimeScope</i>	Point (2)
<i>GatheringType</i>	Periodic (3)
<i>Based on existing eServer OS monitoring metric</i>	No, but FreePhysicalMemory is similar

Host physical memory currently not claimed by any partition.

This metric is based on “*xm info*” output. It is only associated to Xen_Computersystem “domain 0”.

<i>Name</i>	PhysicalMemoryAllocatedToVirtualSystemPercentage
<i>MeasuredElementClass</i>	Xen_ComputerSystem
<i>DataType</i>	real32 (4)
<i>Calculable</i>	Non-Summable (3)
<i>Units</i>	Percent
<i>IsContinuous</i>	TRUE
<i>ChangeType</i>	Gauge (4)
<i>TimeScope</i>	Point (2)
<i>GatheringType</i>	Periodic (3)
<i>Based on existing eServer OS monitoring metric</i>	No

Percentage of maximum memory size actually claimed, that's

$$\begin{aligned}
 &\text{PhysicalMemoryAllocatedToVirtualSystemPercentage} \\
 &= \text{PhysicalMemoryAllocatedtoVirtualSystem} / \\
 &\quad \text{MaxPhysicalMemoryAllocatedToVirtualSystem} \\
 &\quad * 100.0
 \end{aligned}$$

where MaxPhysicalMemoryAllocatedToVirtualSystem is the maximum resource allocation setting data for physical memory for the given virtual system.

<i>Name</i>	HostMemoryPercentage
<i>MeasuredElementClass</i>	Xen_ComputerSystem
<i>DataType</i>	real32 (4)
<i>Calculable</i>	Non-Summable (3)
<i>Units</i>	Percent
<i>IsContinuous</i>	TRUE
<i>ChangeType</i>	Gauge (4)
<i>TimeScope</i>	Point (2)
<i>GatheringType</i>	Periodic (3)
<i>Based on existing eServer OS monitoring metric</i>	No

Percentage of overall physical memory claimed by this domain, that's

HostMemoryPercentage

$$= \text{PhysicalMemoryAllocatedToVirtualSystem} / \text{HostTotalMemory} * 100.0$$

- where “HostTotalMemory” is the size of physical memory available to the host.

