

PAPI-V VMware Component Document

Matthew R. Johnson John Nelson
mrj@eecs.utk.edu jnelso37@utk.edu
21 November 2011
Revised: 23 January 2012

This document is intended to detail the features of the PAPI-V VMware component, and more specifically the installation, usage, and pseudo performance counters available. In order to make this component possible, extensive research into the actual counters available, as well as the leveraging of the VMware Guest API¹, was needed. As this is the first of the PAPI-V components, we seem to be stepping into a new realm of performance measurements that, previously, has been a new frontier, or unexplored all-together.

Installation:

To make PAPI with the VMware component you must go to the PAPI_ROOT/papi/src/components/vmware directory and configure with the flag: `--with-vmware_incdir=<path>`, where `<path>` is the path to the VMware Guest SDK for your machine.

NOTE: The VMware Guest SDK is normally found in the following default vmware-tools path:

```
/usr/lib/vmware-tools/GuestSDK
```

or:

```
/opt/GuestSDK
```

e.g.: `./configure --with-vmware_incdir=/usr/lib/vmware-tools/GuestSDK`

After running configure in the vmware directory, go to PAPI_CVS_ROOT/papi/src and configure again using the flag: `--with-components=vmware`

```
e.g.: ./configure --with-components=vmware
```

After running the main configure script you can then type make, the Makefiles have been automatically generated. If at any point you would like to uninstall PAPI and the VMware component, from the PAPI_ROOT/papi/src directory, just type:

```
make clean clobber
```

To make use of VMWare timekeeping pseudo-performance counters, the following configuration must be added through the vSphere client:

```
monitor_control.pseudo_perfctr = TRUE
```

As well as adding the `--with-vmware_pseudo_perfctr` flag during component configure in the `vmware` component directory.

WARNING: If you do not enable the `monitor_control.pseudo_perfctr` on the host side, and give configure the `--with-vmware_pseudo_perfctr`, you will get a segmentation fault upon `readpmc` trying to access protected memory without privileged access. This is expected behavior.

Available Performance Counters:

Below is the list of available performance metrics available to PAPI through the VMware component. If at any time you would like to see a full list of counters available to PAPI type `./papi_native_avail` from within the `utils` directory.

It is important to know that the counters `VMWARE_HOST_TSC`, `VMWARE_ELAPSED_TIME`, and `VMWARE_ELAPSED_APPARENT` are currently the only true to name register counters available from within a VMware virtual machine. Any Guest OS running on a VMware host must have the access enabled from within the VMware vSphere client managing the system for each virtual machine that wishes to use the VMware component, this exposes the counters to the virtual machine. All other counters you will see in the following list are software counters that are being exposed through the use of the VMware API¹.

Event Code	Symbol	Long Description
0x44000000	VMWARE_HOST_TSC	Physical host TSC
0x44000001	VMWARE_ELAPSED_TIME	Elapsed real time in ns.
0x44000002	VMWARE_ELAPSED_APPARENT	Elapsed apparent time in ns.
0x44000003	VMWARE_CPU_LIMIT	Retrieves the upper limit of processor use in MHz available to the virtual machine.
0x44000004	VMWARE_CPU_RESERVATION	Retrieves the minimum processing power in MHz reserved for the virtual machine.
0x44000005	VMWARE_CPU_SHARES	Retrieves the number of CPU shares allocated to the virtual machine.
0x44000006	VMWARE_CPU_STOLEN	Retrieves the number of milliseconds that th

```

| e virtual machine was in a ready state (able to transition to a r
| un state), but was not scheduled to run.
-----
-
0x44000007 | VMWARE_CPU_USED | Retrieves the number of milliseconds during wh
| ich the virtual machine has used the CPU. This value includes the
| time used by the guest operating system and the time used by vir
| tualization code for tasks for this virtual machine. You can comb
| ine this value with the elapsed time (VMWARE_ELAPSED) to estimate
| the effective virtual machine CPU speed. This value is a subset
| of elapsedMs.
-----
-
0x44000008 | VMWARE_ELAPSED | Retrieves the number of milliseconds that have
| passed in the virtual machine since it last started running on th
| e server. The count of elapsed time restarts each time the virtua
| l machine is powered on, resumed, or migrated using VMotion. This
| value counts milliseconds, regardless of whether the virtual mac
| hine is using processing power during that time. You can combine
| this value with the CPU time used by the virtual machine (VMWARE_
| CPU_USED) to estimate the effective virtual machine xCPU speed. c
| puUsedMS is a subset of this value.
-----
-
0x44000009 | VMWARE_MEM_ACTIVE | Retrieves the amount of memory the virtual m
| achine is actively using in MB its estimated working set size.
-----
-
0x4400000a | VMWARE_MEM_BALLOONED | Retrieves the amount of memory that has b
| een reclaimed from this virtual machine by the vSphere memory bal
| loon driver (also referred to as the "vmmemctl" driver) in MB.
-----
-
0x4400000b | VMWARE_MEM_LIMIT | Retrieves the upper limit of memory that is a
| vailable to the virtual machine in MB.
-----
-
0x4400000c | VMWARE_MEM_MAPPED | Retrieves the amount of memory that is alloc
| ated to the virtual machine in MB. Memory that is ballooned, swap
| ped, or has never been accessed is excluded.

```

```

-----
-
0x4400000d | VMWARE_MEM_OVERHEAD | Retrieves the amount of overhead mem
|
| ory associated with this virtual machine that is currently consum
|
| ed on the host system in MB. Overhead memory is additional memory
|
| that is reserved for data structures required by the virtualizat
|
| ion layer.
-----
-
0x4400000e | VMWARE_MEM_RESERVATION | Retrieves the minimum amount of memory
|
| that is reserved for the virtual machine in MB.
-----
-
0x4400000f | VMWARE_MEM_SHARED | Retrieves the amount of physical memory asso
|
| ciated with this virtual machine that is copy on write (COW)
|
| shared on the host in MB.
-----
-
0x44000010 | VMWARE_MEM_SHARES | Retrieves the number of memory shares alloca
|
| ted to the virtual machine.
-----
-
0x44000011 | VMWARE_MEM_SWAPPED | Retrieves the amount of memory that has bee
|
| n reclaimed from this virtual machine by transparently swapping g
|
| uest memory to disk in MB.
-----
-
0x44000012 | VMWARE_MEM_TARGET_SIZE | Retrieves the size of the target memory
|
| allocation for this virtual machine in MB.
-----
-
0x44000013 | VMWARE_MEM_USED | Retrieves the estimated amount of physical hos
|
| t memory currently consumed for this virtual machine's physical
|
| memory.
-----
-
0x44000014 | VMWARE_HOST_CPU | Retrieves the speed of the ESX system' s phys
|
| ical CPU in MHz.

```

Timekeeping Counters:

The pseudo-performance counter feature uses a trap to catch a privileged machine instruction issued by software running in the virtual machine and therefore has more overhead than reading a performance counter or the TSC on physical hardware. The feature will only trap correctly if the configuration setting is applied as described in Installation.

The timekeeping counters behave as follows:

VMWARE_HOST_TSC - Provides access to the Time Stamp Counter on the host machine which counts ticks since reset.

VMWARE_ELAPSED_TIME - Provides access to the elapsed time in ns since reset as measured on the host machine.

VMWARE_ELAPSED_APPARENT - Apparent time is the time visible to the Guest OS using virtualized timer devices. This timer may fall behind real time and catch up as needed.

Usage:

After installation of the VMware Component, you may use the `papi_command_line` interface, found in `PAPI_ROOT/papi/src/utils` to read out an instantaneous value from a certain counter from the above list. If you would like to read out a counter, type: `./papi_command_line COUNTER_SYMBOL_NAME`.

e.g.: To read out the value of the `VMWARE_MEM_USED` counter

```
user@vm1:~/papi/src/utils$ ./papi_command_line VMWARE_MEM_USED
Successfully added: VMWARE_MEM_USED
```

```
VMWARE_MEM_USED :    13074
```

Verification: Checks for valid event name.

This utility lets you add events from the command line interface to see if they work.

```
command_line.c          PASSED
```

For further usage of PAPI and its API on how to incorporate these counters into a program of your own please see the [PAPI Documentation](#)².

References :

- [1] VMware: <http://www.vmware.com/support/developer/guest-sdk>. Last accessed November 28, 2011
- [2] PAPI : http://icl.cs.utk.edu/projects/papi/wiki/Main_Page. Last accessed November 28, 2011