



VMWARE VSPHERE 4 ADVANCED TROUBLESHOOTING

ERIC SLOOF - NTPRO.NL



INTRODUCTION



- VMware Certified Instructor
- Blogger @ NTPRO.NL



BLOGGER @ NTPRO.NL



the official dutch vmware user group



AGENDA

- 1. Introduction by Scott Drummonds**
- 2. CPU troubleshooting**
- 3. Memory troubleshooting**
- 4. Storage troubleshooting**
- 5. Network troubleshooting**
- 6. Troubleshooting tools**



INTRODUCTION



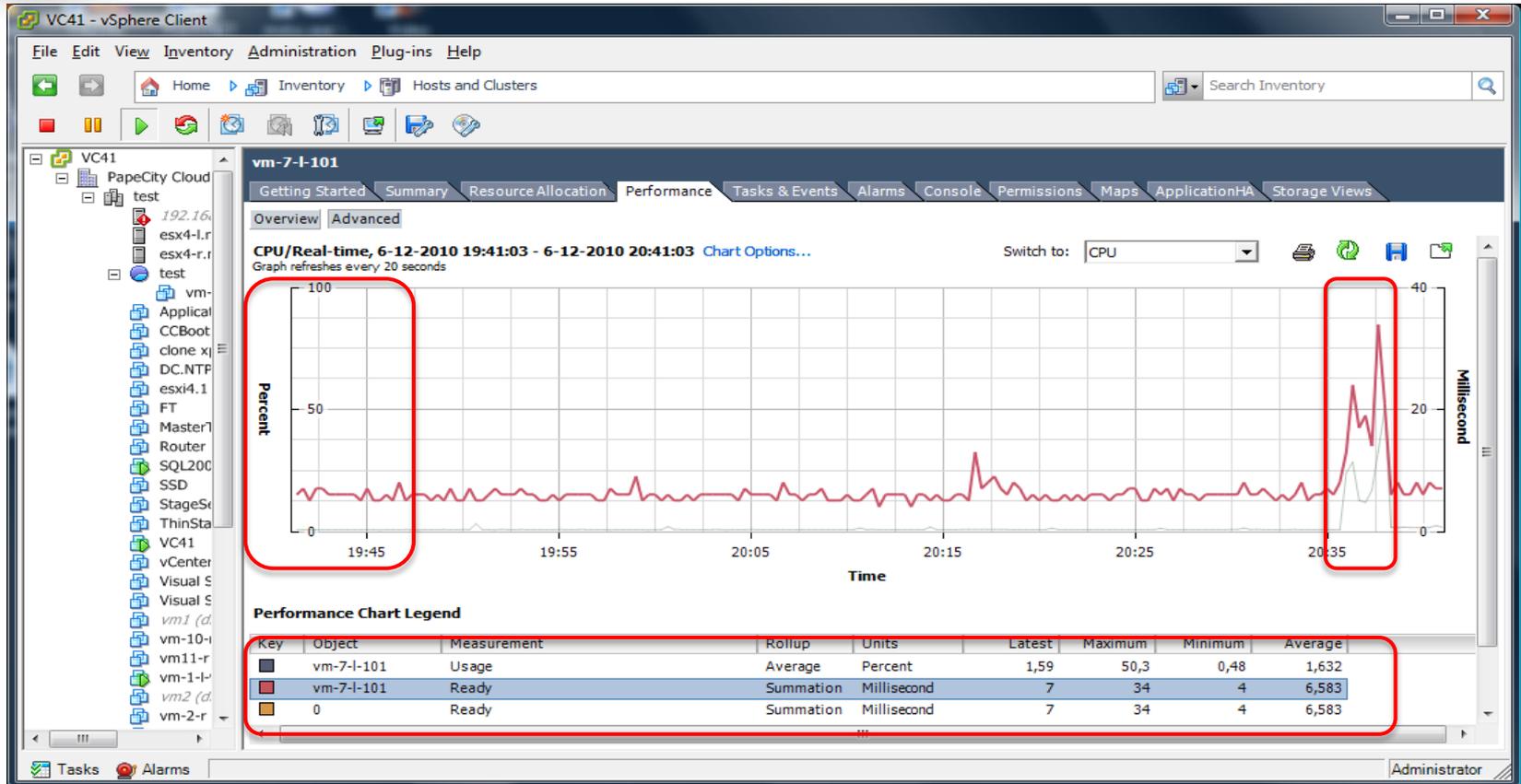
Scott Drummonds

Technical Director, vSpecialists, APJ at EMC

The performance space is massive. It's nearly impossible to keep up with everything that is happening in this space. With the benefit of close contact with VMware's performance engineering team I was barely able to hold the reins on that massive beast. The secret is not to try and learn every little thing out there, but to develop a strong handle on troubleshooting using esxtop, vCenter and vscsiStats. Everything comes from there.



CPU TROUBLESHOOTING – CPU READY TIME



The vSphere Client Graph refreshes every 20 seconds

$1000 \text{ Milliseconds} / 20.000 \text{ Milliseconds} = 5 \%$

$34 \text{ Milliseconds} / 20.000 \text{ Milliseconds} = 0,17 \%$ <~ no worries 😊



CPU TROUBLESHOOTING – CPU READY TIME

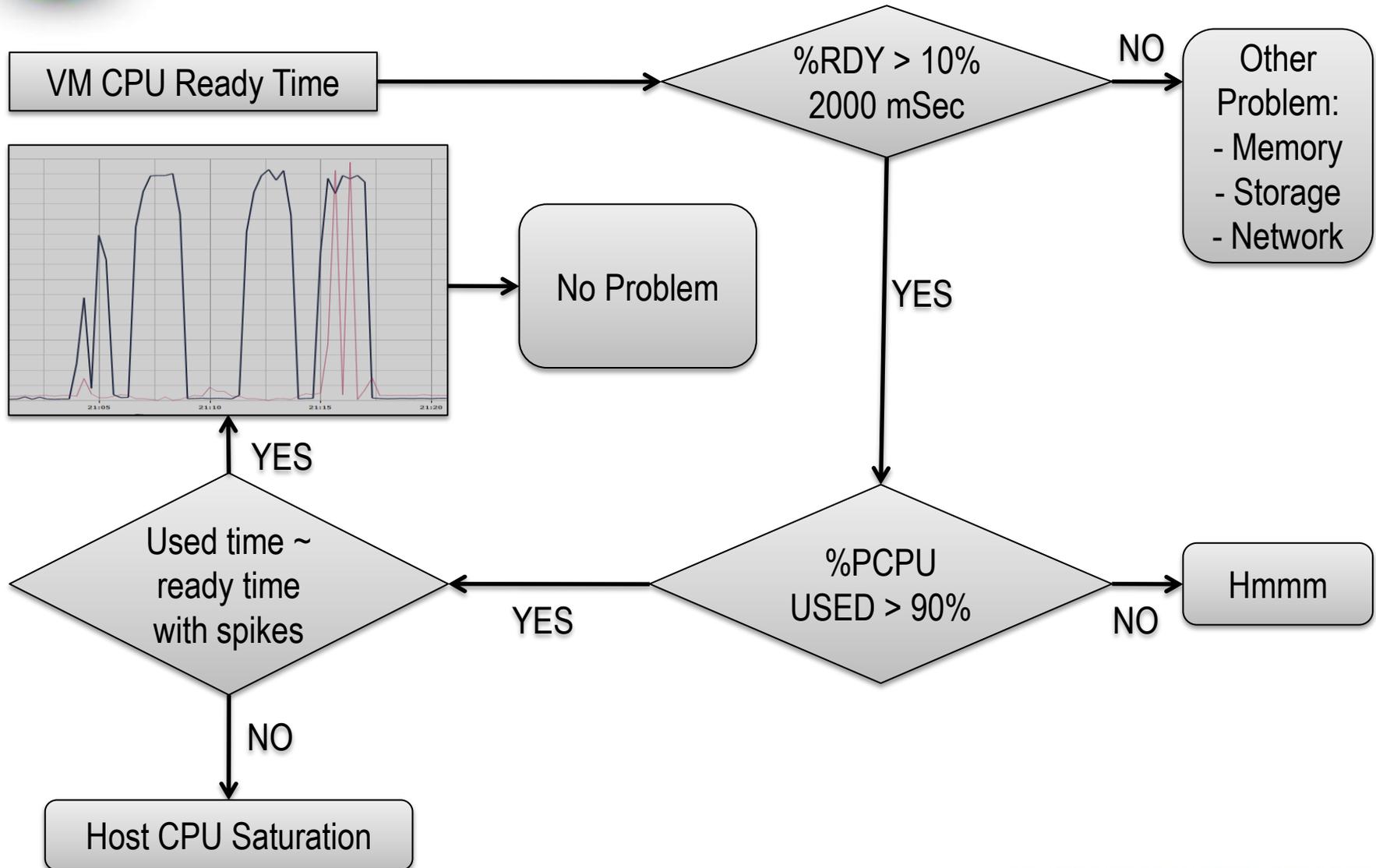
```
esx4-r.ntpro.local - KITTY
11:58:42am up 39 min, 216 worlds; CPU load average: 1.97, 0.64, 0.22
PCPU USED(%): 99 99 99 100 AVG: 100
PCPU UTIL(%): 100 99 100 100 AVG: 99
CCPU(%): 0 us, 3 sy, 0 id, 97 wa ; cs/sec: 271
```

ID	GID	NAME	NWLD	%USED	%RUN	%SYS	%WAIT	%RDY	%IDLE	%OVRLP	%CSTP	%MLMTD	%SWPWT
75	75	Visual Studio 1	5	82.59	83.93	0.03	397.77	17.97	96.61	1.36	0.37	0.00	0.00
69	69	vmVisor	5	81.80	82.97	0.09	397.47	19.43	98.25	1.25	0.18	0.00	0.00
71	71	clone xp	4	54.18	56.31	0.05	299.80	43.93	0.00	2.19	0.00	0.00	0.00
64	64	Visual Studio 2	4	51.45	52.98	0.03	299.87	47.19	0.00	1.57	0.00	0.00	0.00
66	66	SSD	4	51.31	52.15	0.11	301.49	46.40	0.00	0.98	0.00	0.00	0.00
61	61	vm-5-1-130	7	45.31	46.01	0.20	625.78	11.00	193.07	1.02	17.29	0.00	19.39
74	74	FT	4	6.34	3.52	2.94	393.39	3.12	93.54	0.13	0.00	0.00	0.00
58	58	VMware Capacity	4	2.79	2.75	0.04	393.96	3.35	94.06	0.05	0.00	0.00	0.00
67	67	Zenoss	4	2.76	2.67	0.08	394.99	2.38	95.22	0.04	0.00	0.00	0.00
63	63	MasterTestVM	4	2.76	2.74	0.01	395.26	2.05	95.43	0.04	0.00	0.00	0.00
68	68	vSphere Managem	4	2.72	2.69	0.02	395.18	2.17	95.45	0.06	0.00	0.00	0.00
56	56	vm-6-r	7	2.14	2.16	0.00	695.40	2.27	395.44	0.07	0.23	0.00	0.00
60	60	vm-10-r	5	1.96	1.74	0.24	497.68	0.65	97.85	0.02	0.00	0.00	0.00
72	72	ThinStation	5	1.07	1.11	0.00	494.48	4.48	94.60	0.06	0.00	0.00	0.00
59	59	ApplicationHA	5	0.90	0.92	0.01	498.26	0.89	198.47	0.05	0.00	0.00	0.00
70	70	Router	5	0.76	0.77	0.00	498.85	0.43	98.86	0.02	0.00	0.00	0.00
57	57	vm-4-r	4	0.62	0.62	0.00	399.10	0.33	99.13	0.01	0.00	0.00	0.00
55	55	vm-2-r	4	0.62	0.65	0.01	399.23	0.17	99.28	0.04	0.00	0.00	0.00
62	62	vm11-r	4	0.16	0.17	0.00	399.66	0.21	99.83	0.02	0.00	0.00	0.00
73	73	DC.NTPRO.LOCAL	2	0.11	0.03	0.08	199.98	0.01	0.00	0.00	0.00	0.00	0.00

A %RDY figure of 17.97% means that the virtual machine spent 17.97% of its last sample period waiting for available CPU resources. Esxtop's default refresh interval is 5 seconds. The PCPU AVG value in this example is 100%.



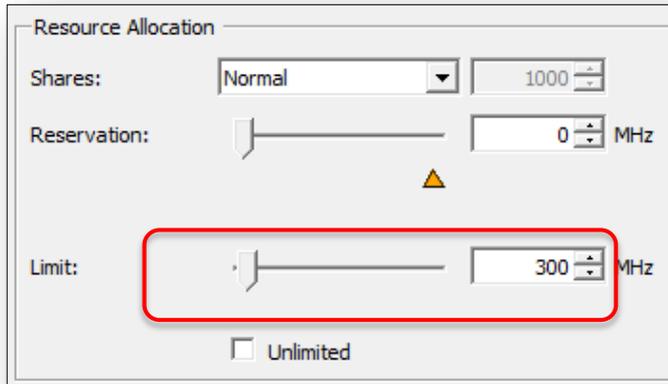
CPU TROUBLESHOOTING - FLOWCHART





CPU TROUBLESHOOTING - MAX LIMITED

%MLMTD - The max limited time is the percentage of time the VM world was ready to run but deliberately wasn't scheduled because that would violate the VM's "CPU limit" settings.



%RDY includes %MLMTD

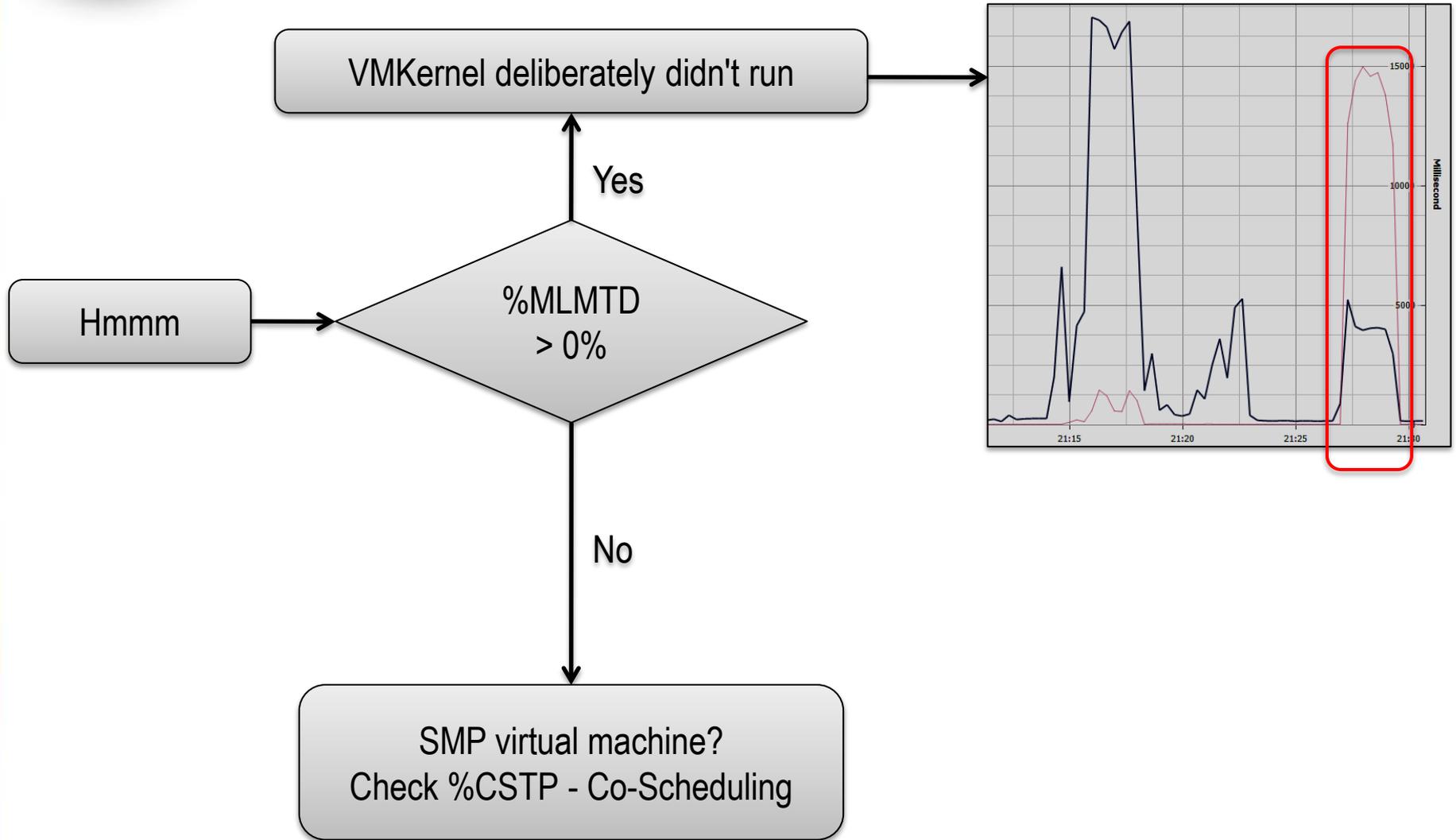
For CPU contention, use "**%RDY - %MLMTD**". $99.75 - 99.73 = 0.02$
So there's no contention despite of the high ready time.

```
esx4-r.ntpro.local - KITTY
10:09:50pm up 5:48, 126 worlds; CPU load average: 0.02, 0.03, 0.02
PCPU USED(%): 89 0.1 0.2 0.1 AVG: 22
PCPU UTIL(%): 100 100 100 100 AVG: 100
CCPU(%): 0 us, 100 sy, 0 id, 0 wa ; cs/sec: 81
```

ID	GID	NAME	NWLD	%USED	%RUN	%SYS	%WAIT	%RDY	%IDLE	%OVRLP	%CSTP	%MLMTD
56	56	vm-2-r	4	0.15	0.15	0.00	299.03	99.75	0.00	0.00	0.00	99.73

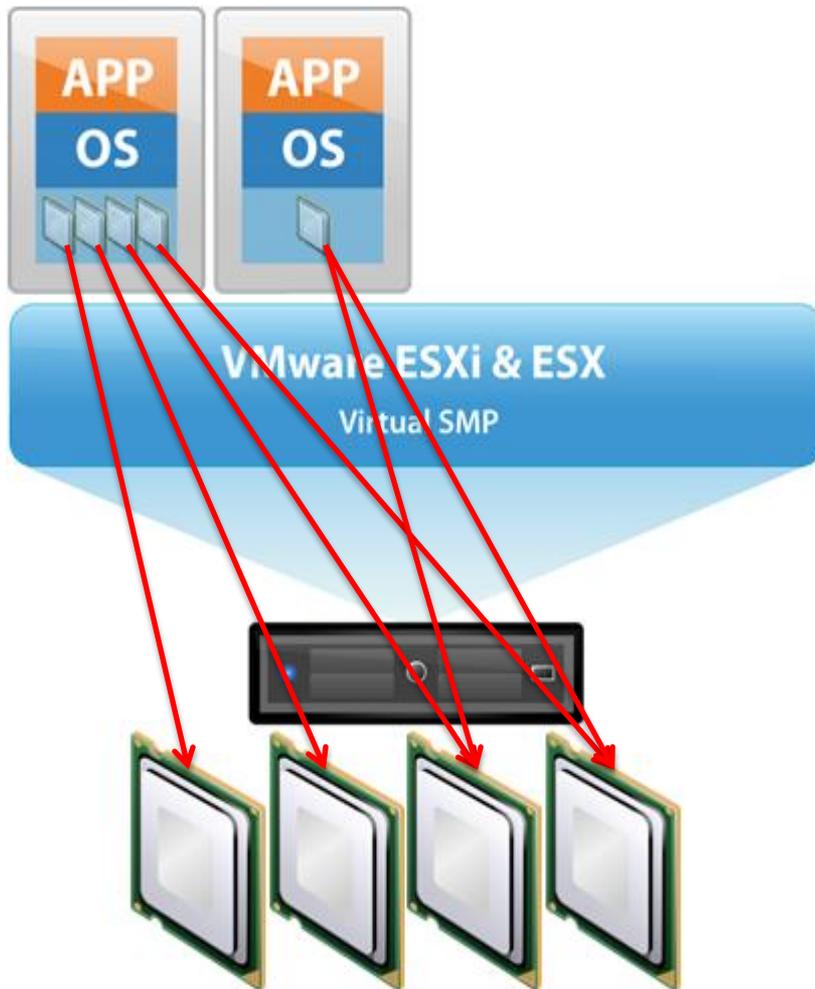


CPU TROUBLESHOOTING - MAX LIMITED





CPU TROUBLESHOOTING – CO SCHEDULING



At any particular point in time, each virtual cpu may be scheduled, descheduled, preempted, or blocked waiting for some event.

Without co scheduling, the VCPUs associated with an SMP VM would be scheduled independently, breaking the guest's assumptions regarding uniform progress.

VMware uses the term "skew" to refer to the difference in execution rates between two or more VCPUs associated with an SMP VM.



CPU TROUBLESHOOTING – CO SCHEDULING

```
root@esx4-r:~
1:41:32pm up 1:15, 137 worlds; CPU load average: 1.07, 1.02, 0.88
PCPU USED(%): 92 93 94 94 AVG: 93
PCPU UTIL(%): 92 93 94 94 AVG: 93
CCPU(%): 1 us, 2 sy, 98 id, 0 wa ; cs/sec: 187
```

ID	GID	NAME	NWLD	%USED	%RUN	%SYS	%WAIT	%RDY	%IDLE	%OVRLP	%CSTP	%MLMTD	%SWPWT
63	63	vm-6-r	7	296.84	297.50	0.09	311.88	76.22	4.72	0.65	14.23	0.00	0.00
61	61	vm-10-r	5	74.53	74.68	0.05	398.76	26.44	0.00	0.15	0.00	0.00	0.00

Type “e” to show all the worlds associated with a single virtual machine. The %CSTP metric indicates co scheduling.

```
root@esx4-r:~
2:02:25pm up 1:36, 135 worlds; CPU load average: 1.25, 0.98, 0.86
PCPU USED(%): 90 93 93 93 AVG: 92
PCPU UTIL(%): 90 93 93 93 AVG: 92
CCPU(%): 0 us, 2 sy, 97 id, 1 wa ; cs/sec: 402
```

ID	GID	NAME	NWLD	%USED	%RUN	%SYS	%WAIT	%RDY	%IDLE	%OVRLP	%CSTP	%MLMTD	%SWPWT
4375	63	vmware-vmx	1	0.07	0.07	0.00	100.00	0.03	0.00	0.00	0.00	0.00	0.00
4377	63	vmassistant.437	1	0.05	0.05	0.00	100.00	0.10	0.00	0.00	0.00	0.00	0.00
4381	63	mks:vm-6-r	1	0.56	0.56	0.00	98.91	0.68	0.00	0.01	0.00	0.00	0.00
4382	63	vcpu-0:vm-6-r	1	71.45	71.53	0.10	3.36	21.77	1.03	0.19	3.50	0.00	0.00
4383	63	vcpu-1:vm-6-r	1	73.68	74.00	0.00	7.12	16.21	0.87	0.25	2.82	0.00	0.00
4384	63	vcpu-2:vm-6-r	1	82.41	82.58	0.00	2.14	12.36	0.67	0.19	3.07	0.00	0.00
4385	63	vcpu-3:vm-6-r	1	65.89	66.11	0.00	6.43	24.55	1.65	0.24	3.07	0.00	0.00
4345	61	vmware-vmx	1	0.10	0.10	0.00	99.84	0.08	0.00	0.00	0.00	0.00	0.00
4347	61	vmassistant.434	1	0.01	0.01	0.00	99.54	0.60	0.00	0.00	0.00	0.00	0.00
4348	61	mks:vm-10-r	1	0.01	0.01	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
4349	61	vcpu-0:vm-10-r	1	73.33	73.54	0.00	0.00	26.60	0.00	0.15	0.00	0.00	0.00
4353	61	Worker#0:vm-10-	1	0.00	0.00	0.00	100.00	0.02	0.00	0.00	0.00	0.00	0.00

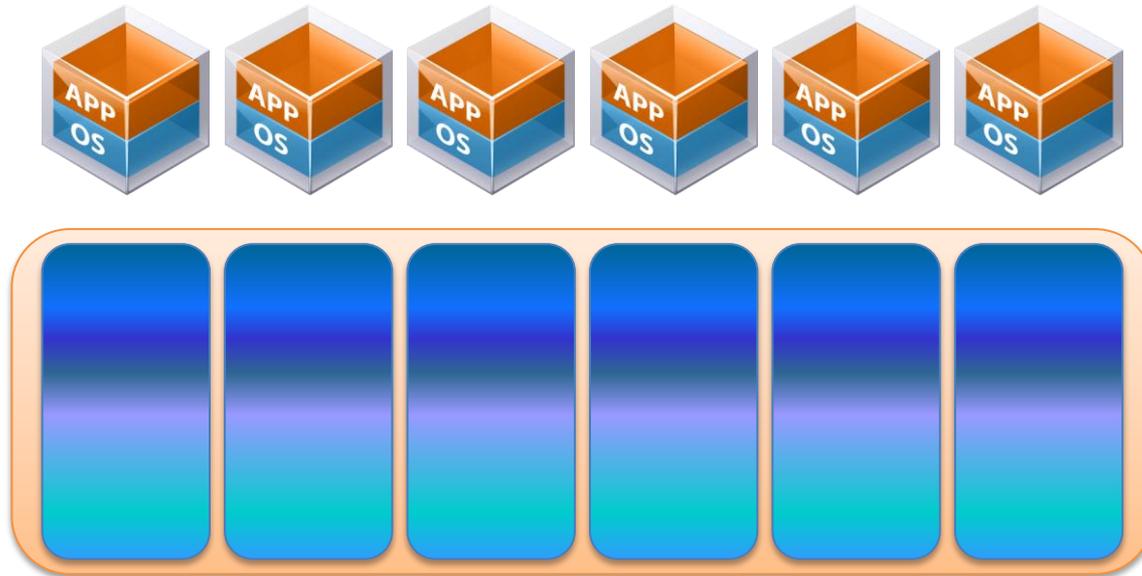


CPU TROUBLESHOOTING - RECAP

- If ready time $\leq 5\%$, there's no problem.
- If ready time is $5\% \leq 10\%$, there might be an issue.
- If ready time is $\geq 10\%$ there's a performance issue.
- Check if the virtual machine's CPU is not limited.
- Check if there's CPU over commitment all the time, occasional spikes are no problem.
- If it's an SMP virtual machine check if the application is multithreading and actually using the resources.
- If the ESX host is saturated reduce the number of virtual machines.



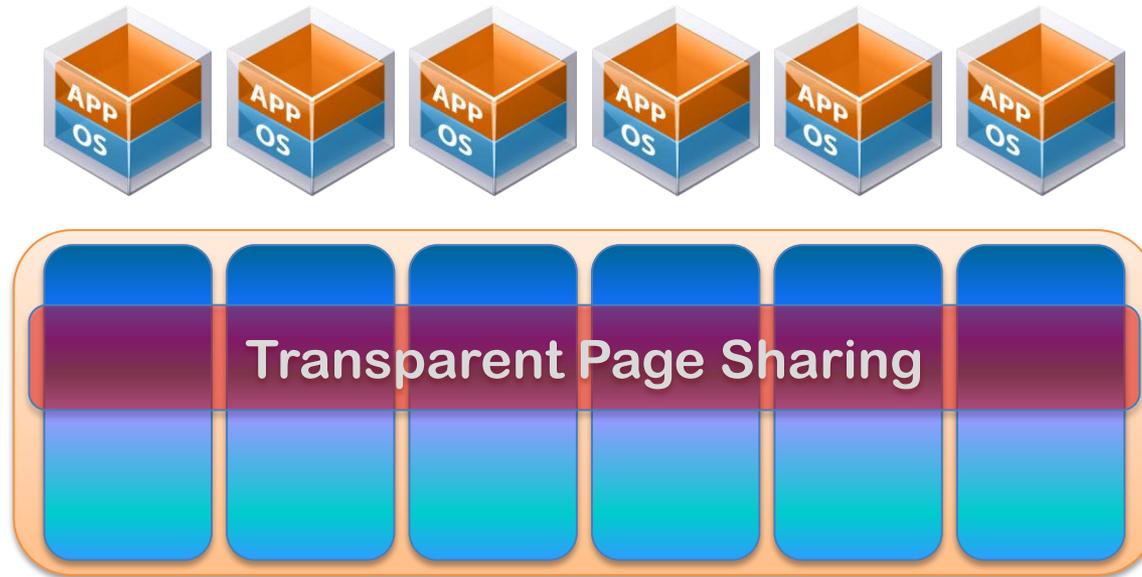
MEMORY TROUBLESHOOTING



For each running virtual machine, the ESX host reserves physical memory for the virtual machine's reservation (if any) and for its virtualization overhead. Because of the memory management techniques the ESX host uses, your VMs can use more memory than there's physically available...



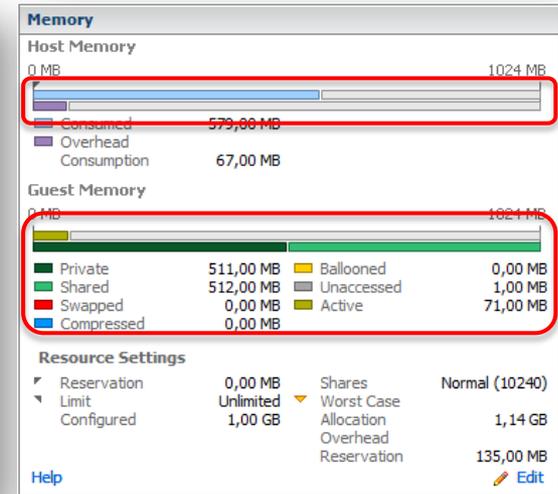
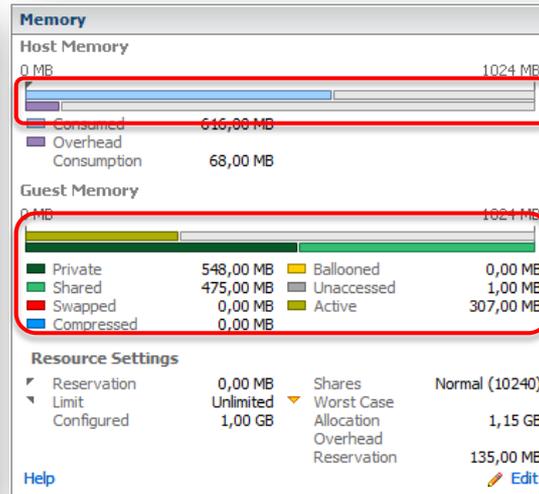
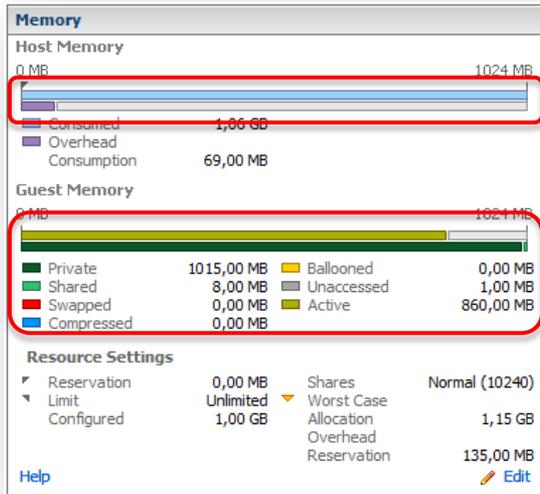
MEMORY TROUBLESHOOTING – PAGE SHARING



Transparent page sharing (TPS) reclaims memory by consolidating redundant pages with identical content. This helps to free memory that a virtual machine would otherwise (not) be using. Page sharing will show up in esxtop at modern Intel/AMD processors only when host memory is overcommitted.



MEMORY TROUBLESHOOTING – PAGE SHARING

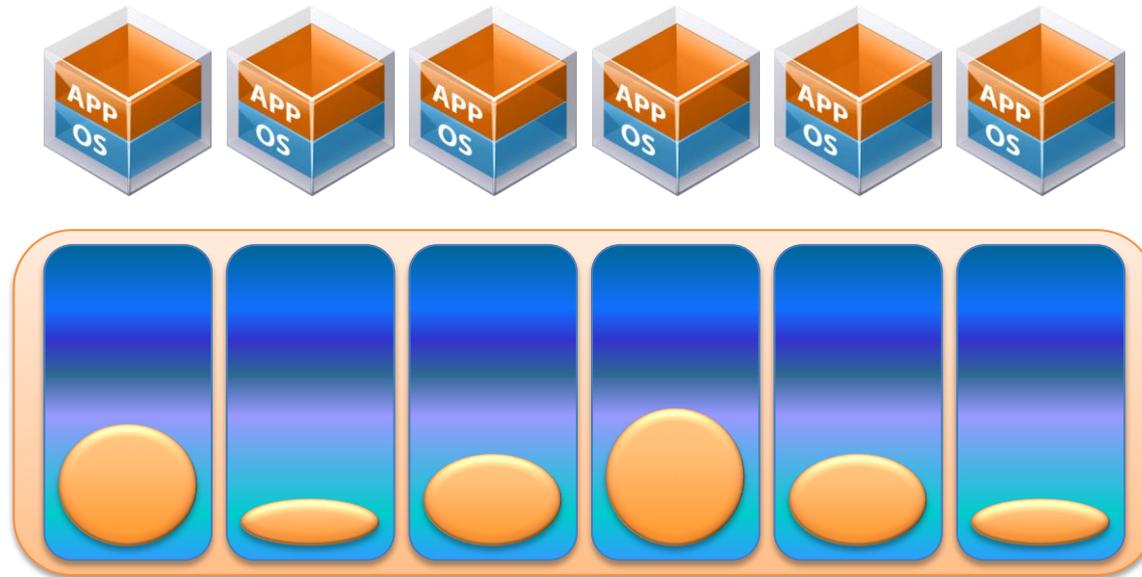


Guest physical memory is not “freed”, the memory is moved to the “free” list. The ESX host has no access to the guest’s “free” list and the ESX host cannot “reclaim” the memory freed up by the guest.

Sharing happens with other virtual machines on the same host but also within virtual machines.



MEMORY TROUBLESHOOTING - BALLOONING



Ballooning reclaims memory by artificially increasing the memory pressure inside the guest and will become a performance issue when the guest OS is paging active memory to its own page file. Ballooning offers a better performance than ESX swapping or ESX memory compression.



MEMORY TROUBLESHOOTING - BALLOONING

```
esx4-r.ntpro.local - KiTTY
8:49:00pm up 2:35, 151 worlds; MEM overcommit avg: 0.21, 0.78, 1.57
PMEM /MB: 8182 total: 300 cos, 558 vmk, 5295 other, 2028 free
VMKMEM/MB: 7730 managed: 463 minfree, 2991 rsvd, 4738 ursvd, high state
COSMEM/MB: 5 free: 596 swap_t, 594 swap_f: 0.00 r/s, 0.00 w/s
PSHARE/MB: 1507 shared, 30 common: 1477 saving
SWAP /MB: 275 curr, 312 rclmtgt: 0.01 r/s, 0.00 w/s
ZIP /MB: 74 zipped, 45 saved
MEMCTL/MB: 1492 curr, 1565 target, 5323 max
```

NAME	MEMSZ	GRANT	SZTGT	TCHD	TCHD W	MCTL?	MCTLSZ	MCTLTGT	MCTLMAX
Visual Studio 1	4096.00	4094.12	3873.04	1720.32	1597.44	Y	0.00	0.00	2662.11
vm-5-l-130	2048.00	1831.79	1137.02	1753.96	1414.48	Y	162.02	235.69	1331.20
SSD	1024.00	193.00	100.04	10.77	3.59	Y	665.02	665.02	665.02
vm-10-r	1024.00	208.16	95.96	17.95	7.18	Y	665.02	665.02	665.02
vm-2-r	1024.00	5.00	36.34	768.00	768.00	N	0.00	0.00	0.00

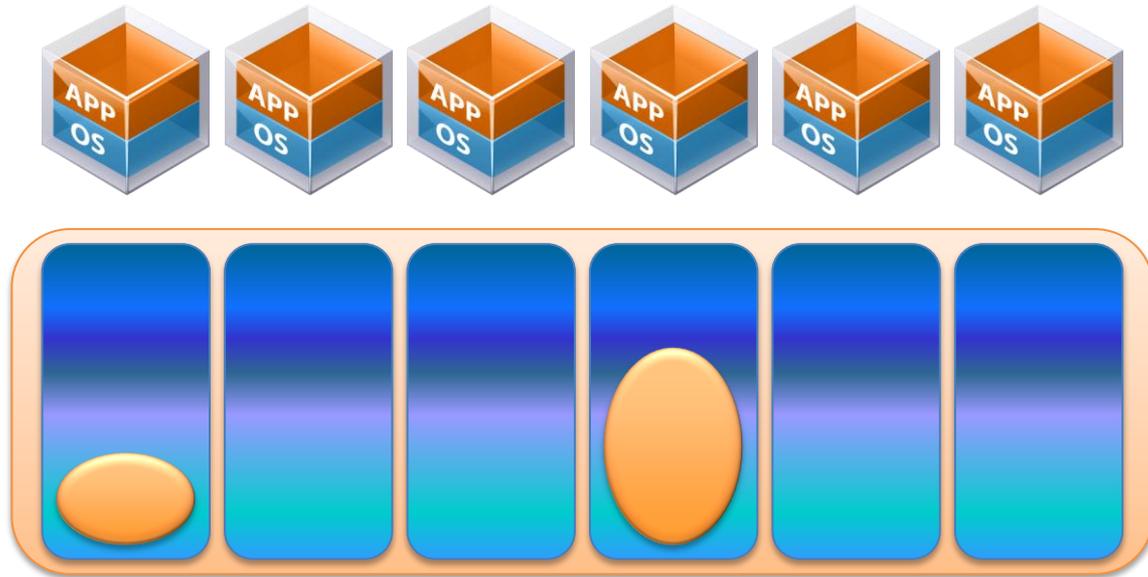
The **MCTLTGT** (target) value set by VMkernel for the VM's memory balloon size, in conjunction with **MCTLSZ** (size) metric, is used by VMkernel to inflate and deflate the balloon for a virtual machine.

If **MCTLTGT > MCTLSZ** the VMkernel inflates the balloon.

If **MCTLTGT < MCTLSZ** the VMkernel deflates balloon.



MEMORY TROUBLESHOOTING - LIMIT

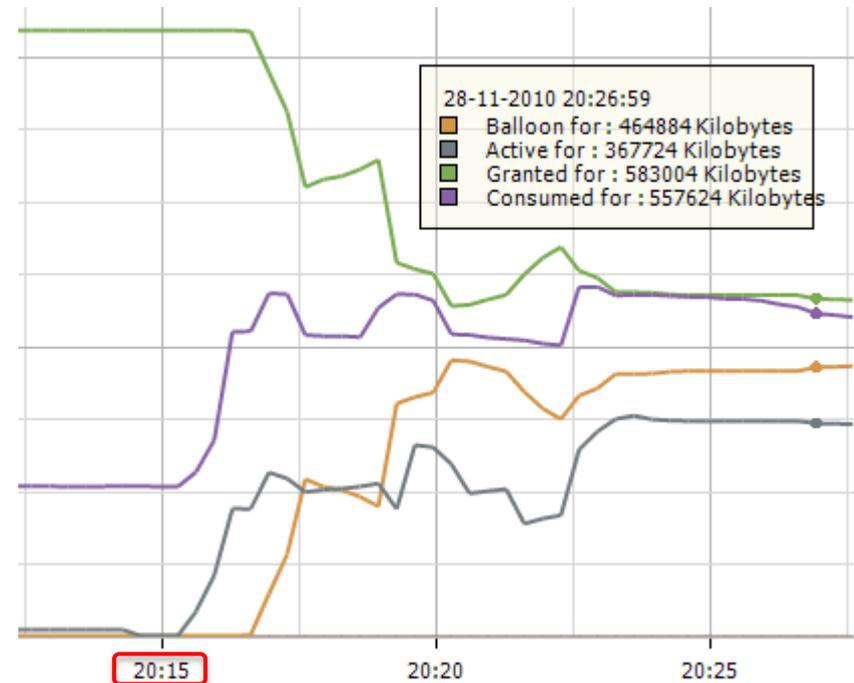


Don't configure VM memory limits, set an appropriate VM memory size instead! Virtual machines deployed from a template with a configured memory limit will become ballooning ghosts after adding more configured memory. Even though there's enough memory available at host level you will see ballooning with a maximum of 65%.



MEMORY TROUBLESHOOTING - LIMIT

This is an example of a virtual machine configured with 1024 MB of memory and no limit. Before 20:15 there's no memory limit configured after 20:15 the limit is set 512 MB. As soon as the VM is trying to access memory above 512 MB - ballooning kicks in.

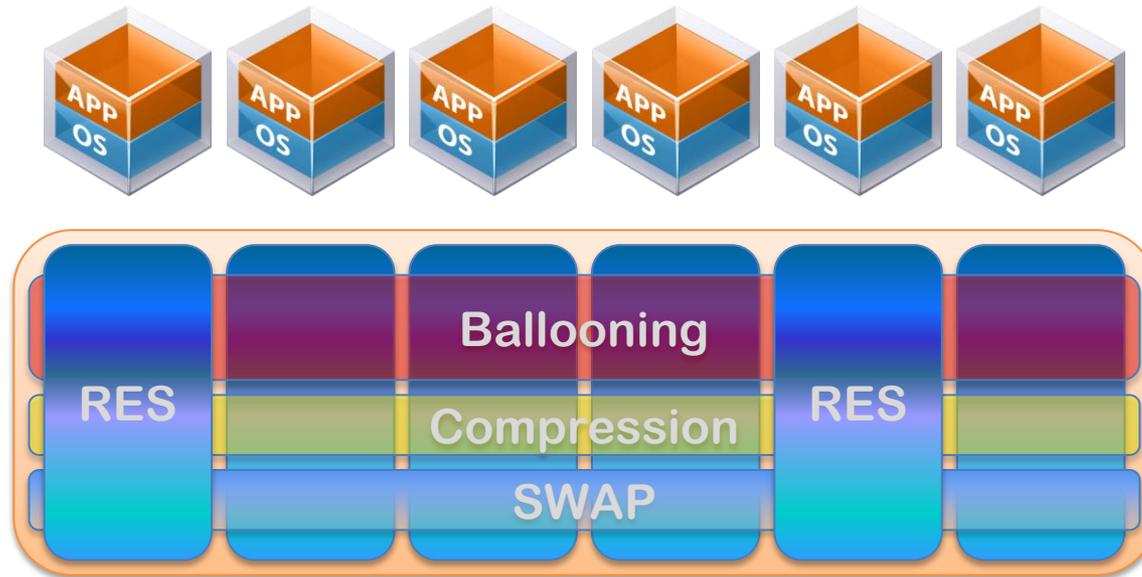


```
esx4-r.ntpro.local - KITTY
8:24:32pm up 5:41, 128 worlds; MEM overcommit avg: 0.09, 0.24, 0.30
PMEM /MB: 8182 total: 300 cos, 550 vmk, 665 other, 6666 free
VMKMEM/MB: 7730 managed: 463 minfree, 1308 rsvd, 6421 ursvd, high state
COSMEM/MB: 7 free: 596 swap_t, 589 swap_f: 0.00 r/s, 0.00 w/s
PSHARE/MB: 22 shared, 20 common: 2 saving
SWAP /MB: 0 curr, 0 rclmtgt: 0.00 r/s, 0.00 w/s
ZIP /MB: 0 zipped, 0 saved
MEMCTL/MB: 446 curr, 508 target, 665 max
```

GID	NAME	MEMSZ	GRANT	SZTGT	TCHD	TCHD W	MCTL?	MCTLSZ	MCTLTGT	MCTLMAX
55	vm-2-r	1024.00	576.40	592.08	363.55	346.24	Y	446.93	508.58	665.02



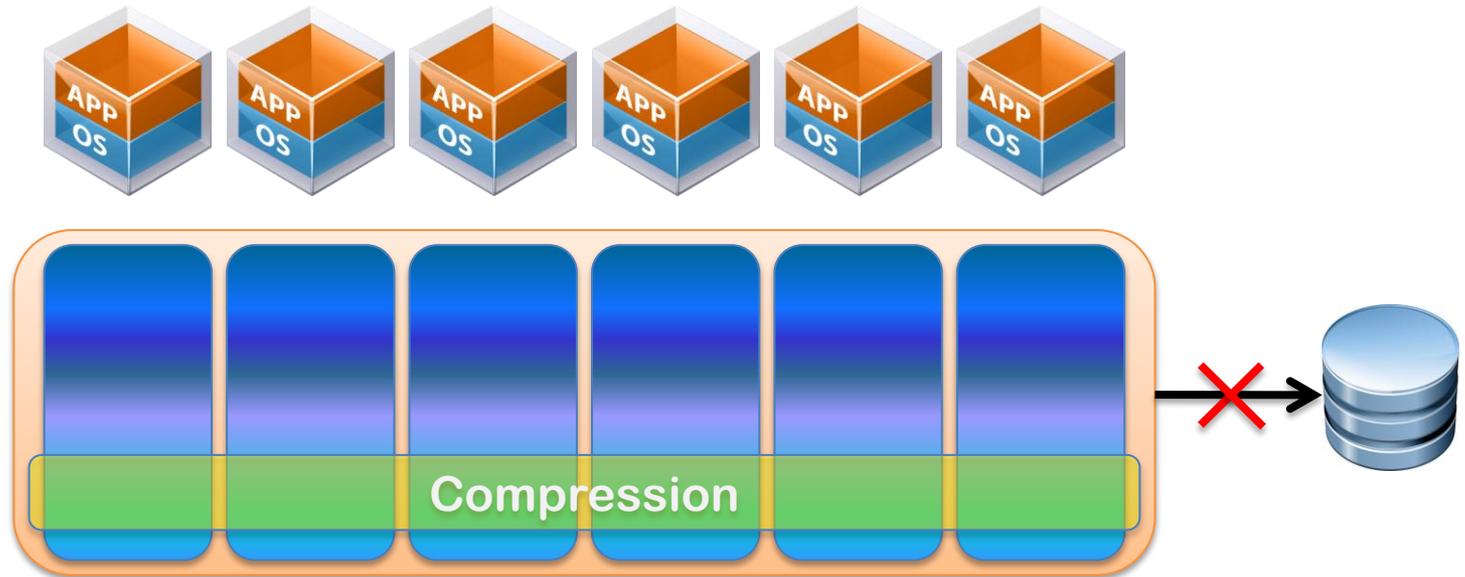
MEMORY TROUBLESHOOTING - RESERVATION



Be careful with configuring a high VM reservation. As soon as a virtual machine has used or touched it's reserved memory, the other virtual machines can't use it anymore. The VM reservation is also used for calculating the slot size in an HA cluster with "number host failures allowed". Only reserve what is really used and needs to be guaranteed.



MEMORY TROUBLESHOOTING – COMPRESSION

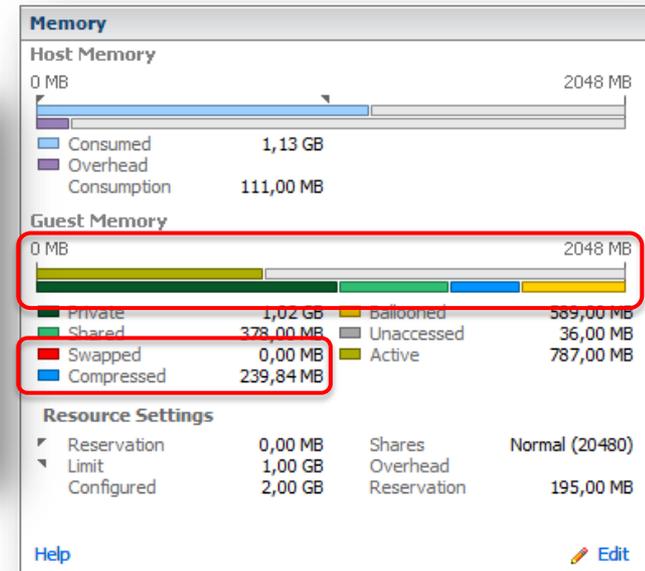


Memory compression reclaims memory by compressing the pages that need to be swapped out. If the swapped out pages can be compressed and stored in a compression cache located in the main memory, the next access to the page only causes a page decompression, which can be an order of magnitude faster than the disk access. This means the number of future synchronous swap-in operations will be reduced. The compression ratio must be + 50%.



MEMORY TROUBLESHOOTING – COMPRESSION

SWCUR	SWTGT	SWR/s	SWW/s	CACHESZ	CACHEUSD	ZIP/s	UNZIP/s
219.30	257.37	0.12	0.00	27.45	26.61	0.00	0.16
134.54	167.62	0.00	0.00	45.26	44.49	0.00	0.00
177.72	576.92	0.16	3.02	163.70	163.69	10.97	0.16
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
57.23	116.25	0.00	0.00	42.74	42.71	0.00	0.00
48.24	46.52	0.00	0.00	29.93	28.17	0.00	0.00
66.40	3.45	0.00	0.00	5.30	3.72	0.00	0.00
54.93	118.16	0.00	0.00	30.66	29.86	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
93.36	119.31	0.00	0.00	18.71	18.37	0.00	0.00

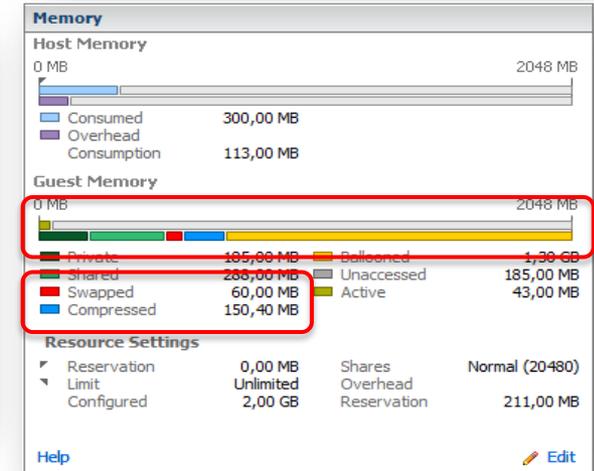


- The **CACHESZ** value (10% of the VM memory) is the compression cache size.
- The **CACHEUSD** value is the compression cache currently used.
- **ZIP/s** and **UNZIP/s** are the compressions and uncompressing actions per second.



MEMORY TROUBLESHOOTING – SWAP

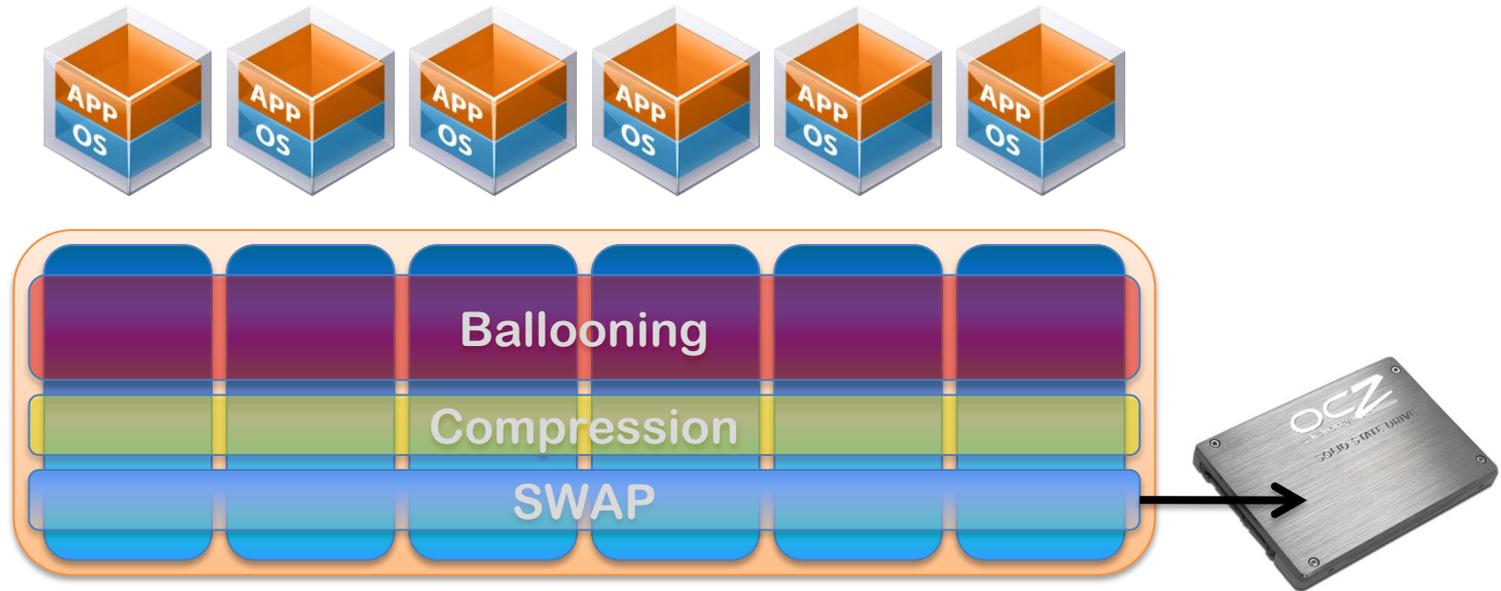
SWCUR	SWTGT	SWR/s	SWW/s	CACHESZ	CACHEUSD	ZIP/s	UNZIP/s
219.30	257.37	0.12	0.00	27.45	26.61	0.00	0.16
134.54	167.62	0.00	0.00	45.26	44.49	0.00	0.00
177.72	576.92	0.16	3.02	163.70	163.69	10.97	0.16
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
57.23	116.25	0.00	0.00	42.74	42.71	0.00	0.00
48.24	46.52	0.00	0.00	29.93	28.17	0.00	0.00
66.40	3.45	0.00	0.00	5.30	3.72	0.00	0.00
54.93	118.16	0.00	0.00	30.66	29.86	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
93.36	119.31	0.00	0.00	18.71	18.37	0.00	0.00



- **SWCUR** is the current amount of guest physical memory swapped out to the virtual machine's swap file by the VMkernel. Swapped memory stays on disk until the virtual machine needs it.
- If **SWTGT** > **SWCUR**, the VMkernel can start swapping when necessary.
- If **SWTGT** < **SWCUR**, the VMkernel stops swapping memory.



MEMORY TROUBLESHOOTING - SWAP



High swap-in latency, which can be tens of milliseconds, can severely degrade guest performance. If available configure local SSD storage for your virtual machine swap file location. There's a -12% degradation with local SSD versus -69% for Fiber Channel and -83% for local SATA storage.



MEMORY TROUBLESHOOTING – SWAP

```
esx4-r.ntpro.local - KITTY
10:35:05pm up 11:16, 151 worlds; CPU load average: 0.07, 0.03, 0.13
PCPU USED(%): 4.5 3.3 10 10 AVG: 7.0
PCPU UTIL(%): 4.9 3.8 10 10 AVG: 7.5
CCPU(%): 0 us, 2 sy, 96 id, 2 wa ; cs/sec: 173
```

ID	GID	NAME	NWLD	%USED	%RUN	%SYS	%WAIT	%RDY	%IDLE	%OVRLP	%CSTP	%MLMTD	%SWPWT
66	66	SSD	4	21.76	22.28	1.37	377.67	0.09	35.45	1.18	0.00	0.00	17.09
59	59	ApplicationHA	5	0.98	1.08	0.01	498.87	0.09	198.56	0.02	0.00	0.00	0.00
70	70	Router	5	0.58	0.69	0.00	499.31	0.04	99.32	0.03	0.00	0.00	0.00
71	71	clone xp	4	0.55	0.63	0.00	399.36	0.04	99.27	0.01	0.00	0.00	0.00
73	73	DC.NTPRO.LOCAL	4	0.49	0.57	0.00	399.41	0.04	99.61	0.01	0.00	0.00	0.00
74	74	FT	4	0.47	0.56	0.00	399.43	0.04	99.38	0.02	0.00	0.00	0.00

SWR/s	SWW/s
0.12	0.00
0.00	0.00
0.16	3.02
0.00	0.00
0.00	0.00
0.00	0.00
0.00	0.00
0.00	0.00
0.00	0.00
0.00	0.00
0.00	0.00
0.00	0.00

- **SWPWT** is the percentage of time that the virtual machine is waiting for memory to be swapped in. This value shouldn't be above 5%.
- **SWR/s** is the rate at which memory is swapped from (SSD) disk into active memory.
- **SWW/s** is the rate at which memory is being swapped from active memory and written to (SSD) disk.

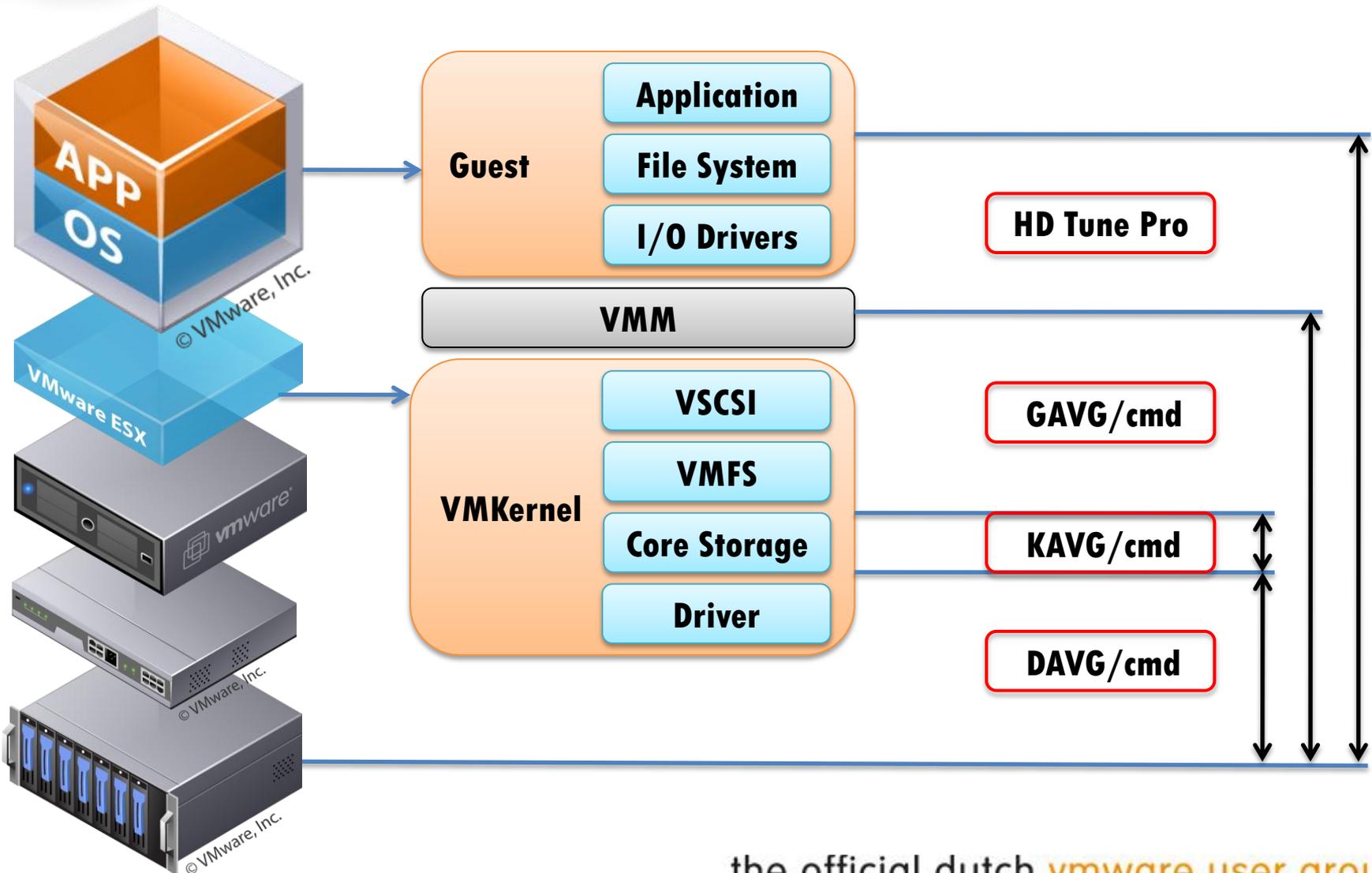


MEMORY TROUBLESHOOTING - RECAP

- **Be careful with setting virtual machine memory reservations. When memory is touched by the VM, the other virtual machines can't use the memory anymore. Only configure what the virtual machine really needs.**
- **Don't set memory limits, set an appropriate virtual machine memory size instead.**
- **Do not disable page sharing or the balloon driver. Ballooning is OK as long as the guest OS isn't using it's own page file for active memory swapping.**
- **The use of large pages results in reduced memory management overhead and can therefore increase hypervisor performance. But take into consideration that using large pages (2MB) TSP might not occur until memory over commitment is high enough to require the large pages to be broken into small pages.**



STORAGE TROUBLESHOOTING – THE STACK



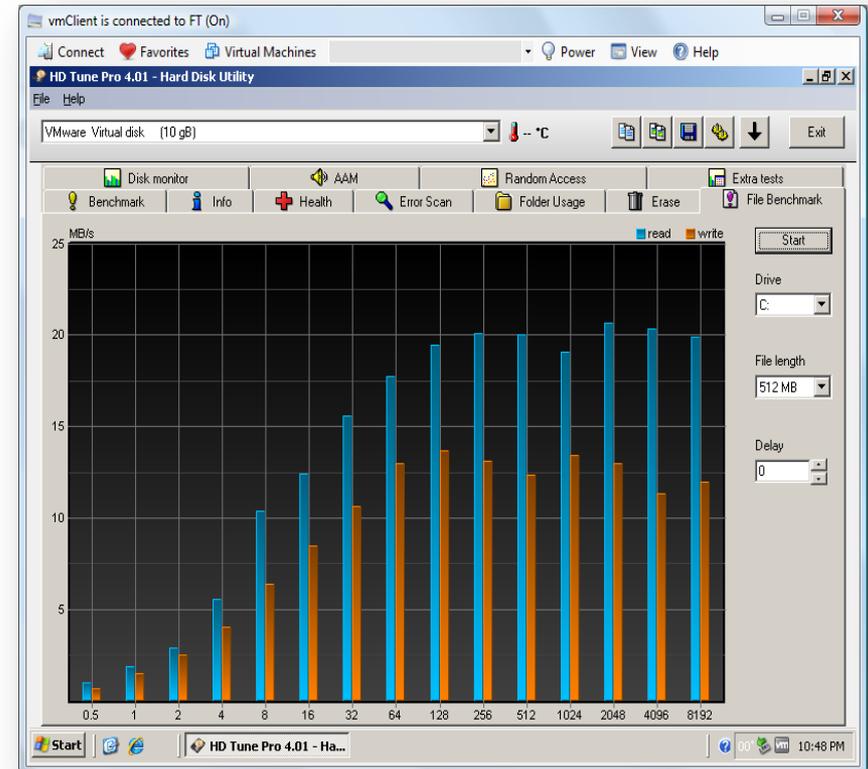


STORAGE TROUBLESHOOTING – THE METRICS

DAVG - This is the latency seen at the device driver level. It includes the roundtrip time between the HBA and the storage.

KAVG - This counter tracks the latency due to the ESX Kernel's command.

GAVG - This is the round-trip latency that the guest sees for all IO requests sent to the virtual storage device.



```
esx4-r.ntpro.local - KITTY
10:52:49pm up 7:16, 132 worlds; CPU load average: 0.03, 0.01, 0.01
```

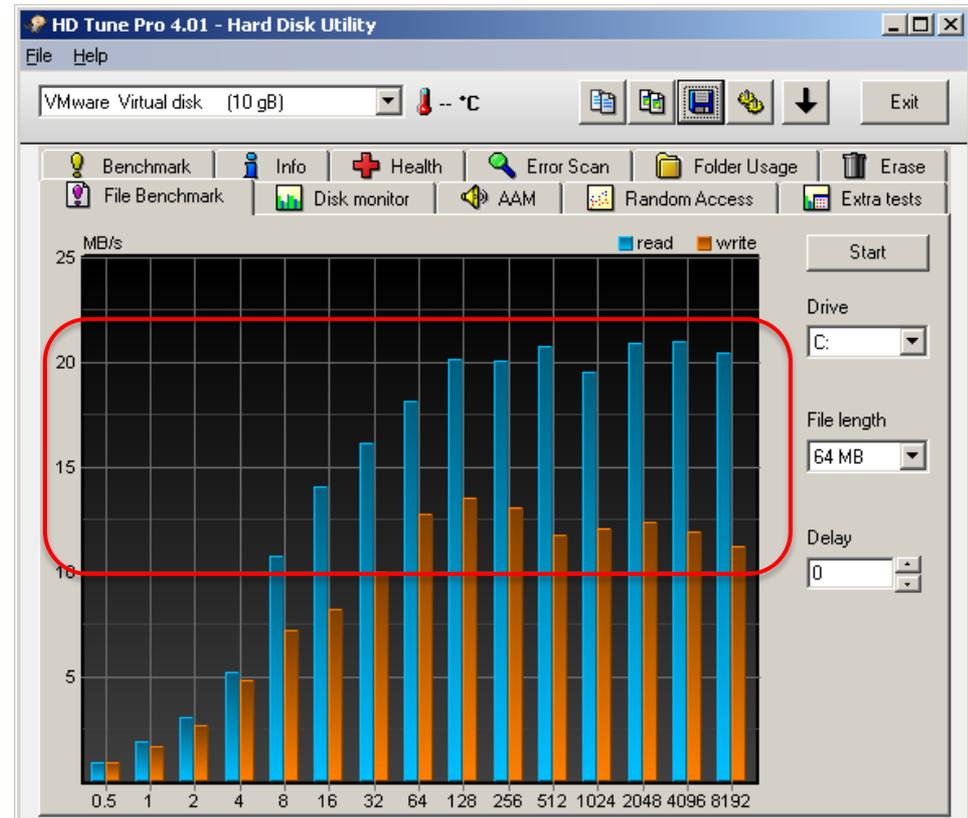
ADAPTR	PATH	NPTH	CMDS/s	READS/s	WRITES/s	MBREAD/s	MBWRTN/s	DAVG/cmd	KAVG/cmd	GAVG/cmd	QAVG/cmd
vmhba0	-	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
vmhba1	-	1	57.79	11.26	41.63	0.02	0.11	0.75	0.23	0.98	0.00
vmhba33	-	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
vmhba34	-	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
vmhba35	-	1	323.73	321.77	1.96	20.01	0.00	157.18	0.00	157.18	0.00



STORAGE TROUBLESHOOTING – IOMEGA IX2

Iomega StorCenter ix2
with 500 GB - RAID 1

1 Gigabit Ethernet
Jumbo frame support
iSCSI target or CIFS/NFS



```
esx4-lntpro.local - KITTY
10:14:54pm up 7:01, 149 worlds; CPU load average: 0.06, 0.05, 0.05
```

ADAPTR	PATH	NPTH	CMDS/s	READS/s	WRITES/s	MBREAD/s	MBWRTN/s	DAVG/cmd	KAVG/cmd	GAVG/cmd	QAVG/cmd
vmhba0	-	1	4.36	0.20	4.16	0.00	0.04	3.91	1.31	5.23	0.00
vmhba1	-	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
vmhba32	-	1	1.19	0.00	1.19	0.00	0.00	1.66	0.02	1.68	0.01
vmhba33	-	1	1.39	0.00	1.39	0.00	0.00	0.09	0.02	0.11	0.01
vmhba34	-	1	263.06	149.95	113.11	9.26	6.64	31.26	0.00	31.26	0.00



STORAGE TROUBLESHOOTING - (CONS/s)



```
esx4-r.ntpro.local - KITTY
[root@esx4-r ISCSI]# cat loop.sh
for i in {0..100000}
do
    touch test.csv
done
[root@esx4-r ISCSI]#
```

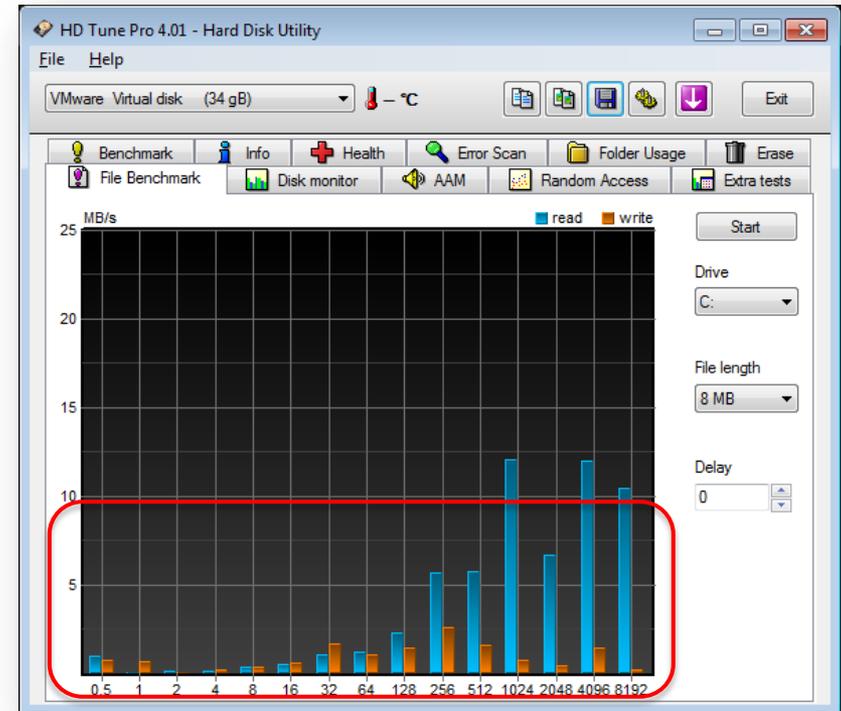
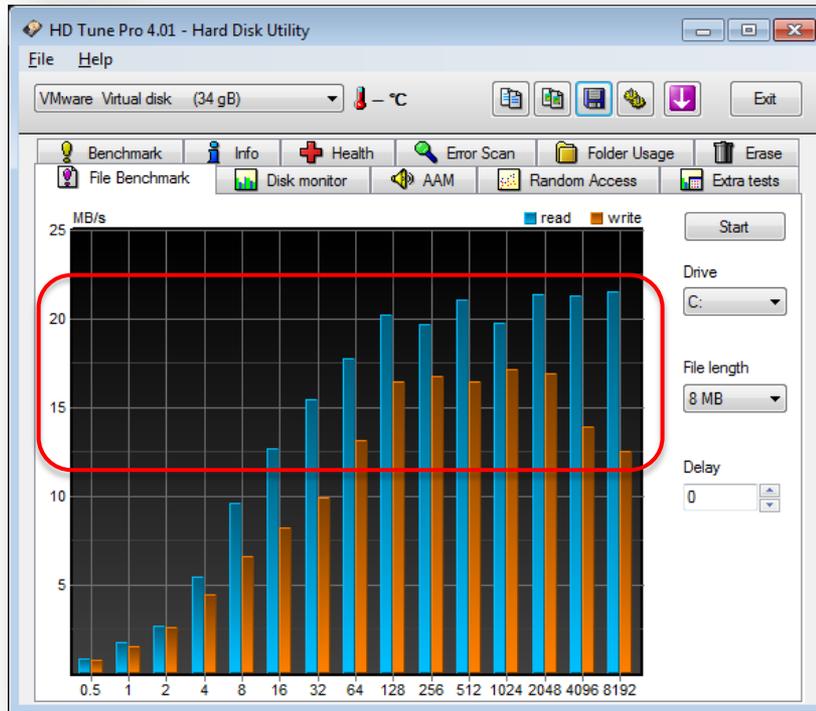
The SCSI reservation conflict counter - **CONS/s** will become non-zero when a host tries to do SCSI reservation on a LUN which has a SCSI reservation in progress. This happens only when two hosts try to do metadata operation on the same LUN at the same exact time.

```
esx4-l.ntpro.local - KITTY
9:37:31pm up 1:50, 136 worlds; CPU load average: 0.04, 0.04, 0.04
```

DEVICE	DQLEN	WQLEN	ACTV	QUED	%USD	LOAD	CMDS/s	READS/s	WRITES/s	MBREAD/s	MBWRN/s	RESV/s	CONS/s	D AVG/cmd
t10.54D4340000	32	-	0	0	0	0.00	11.72	0.00	11.72	0.00	8.80	0.00	8.79	230.35
t10.ATA__INT	1	-	0	0	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
t10.ATA__SAM	1	-	0	0	0	0.00	12.69	0.00	12.69	0.00	0.07	0.00	0.00	0.31
t10.ATA__SAM	1	-	0	0	0	0.00	0.49	0.00	0.49	0.00	0.00	0.00	0.00	1.83



STORAGE TROUBLESHOOTING - (CONS/s)



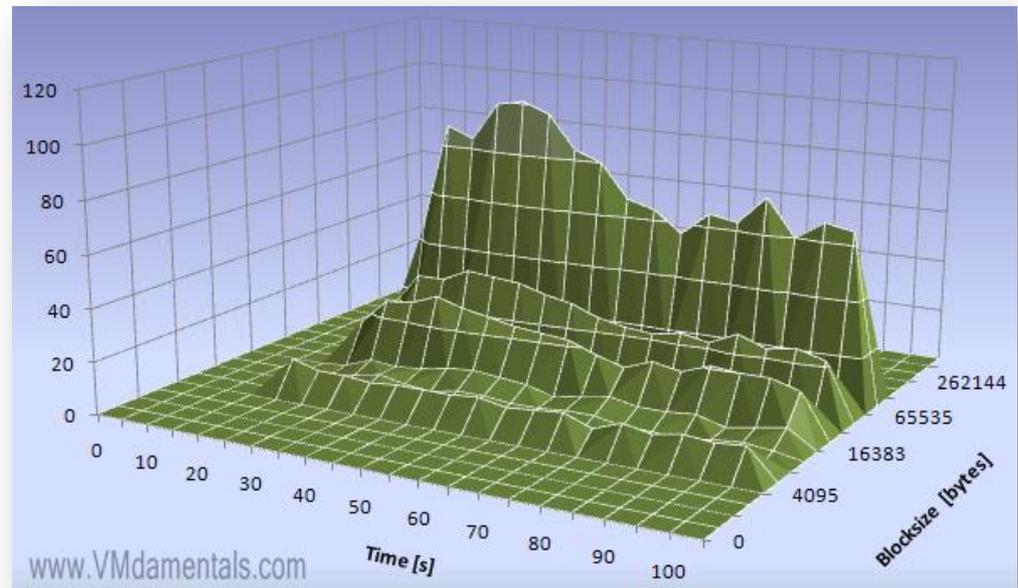
SCSI reservation is held for a very short period (few hundred microseconds) so the chances of getting a conflict is very less on a small cluster. However as the number of hosts that shares the LUN increases conflicts could arise more frequently.



STORAGE TROUBLESHOOTING - VSCSI STAT

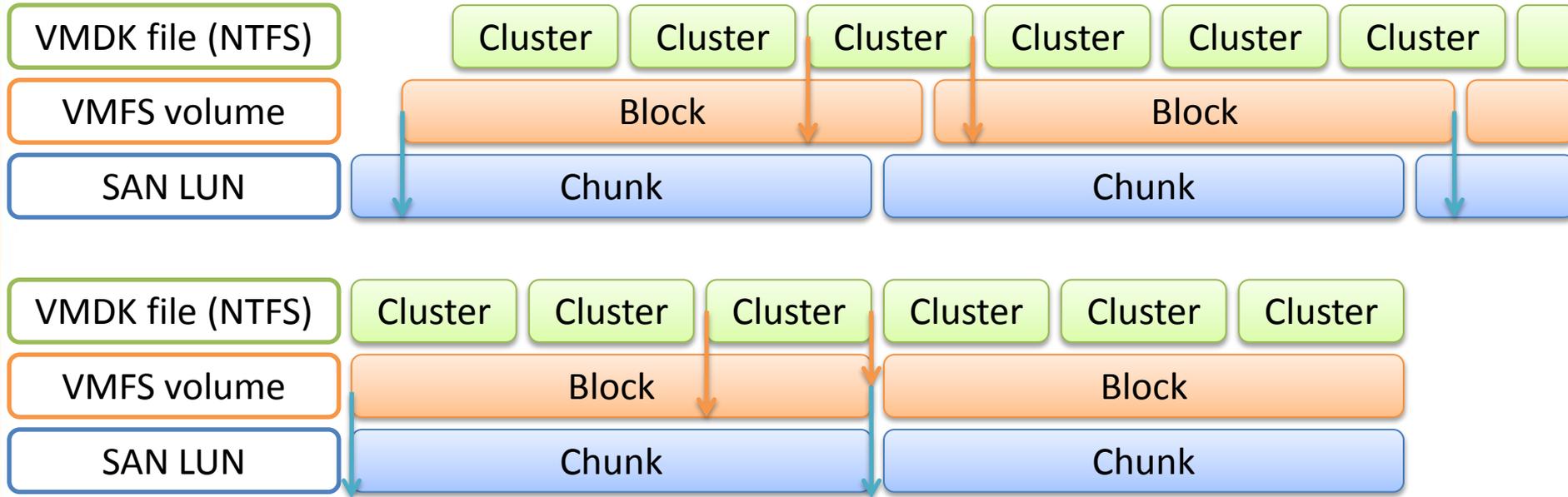
vscsiStats collects and reports counters on storage activity. Its data is collected at the virtual SCSI device level in the kernel. This means that results are reported per VMDK (or RDM) irrespective of the underlying storage protocol. The following data are reported in histogram form:

- IO size
- Seek distance
- Outstanding IOs
- Latency (in mSecs)





STORAGE TROUBLESHOOTING - ALIGNMENT



Like other known disk based file systems, VMFS suffers a penalty when the partition is unaligned. Use the vSphere client to create VMFS partitions since the vSphere client automatically aligns the partitions along the 64 KB boundary.



STORAGE TROUBLESHOOTING – ALIGNMENT

- Guest OS alignment is important for Microsoft Windows Server 2003, XP and 2000. When a partition is created on Windows 2008 or Windows 7 the newly created partition is automatically aligned.
- Windows uses a factor of 512 bytes to create volume clusters. This behavior causes a misaligned partition.
- To resolve this issue, use the Diskpart.exe tool to create the disk partition and to specify a starting offset of 128 sectors (64 kilobyte).
- Create partition primary align=64

$((\text{Partition offset}) * (\text{Disk sector size})) / (\text{Stripe unit size})$

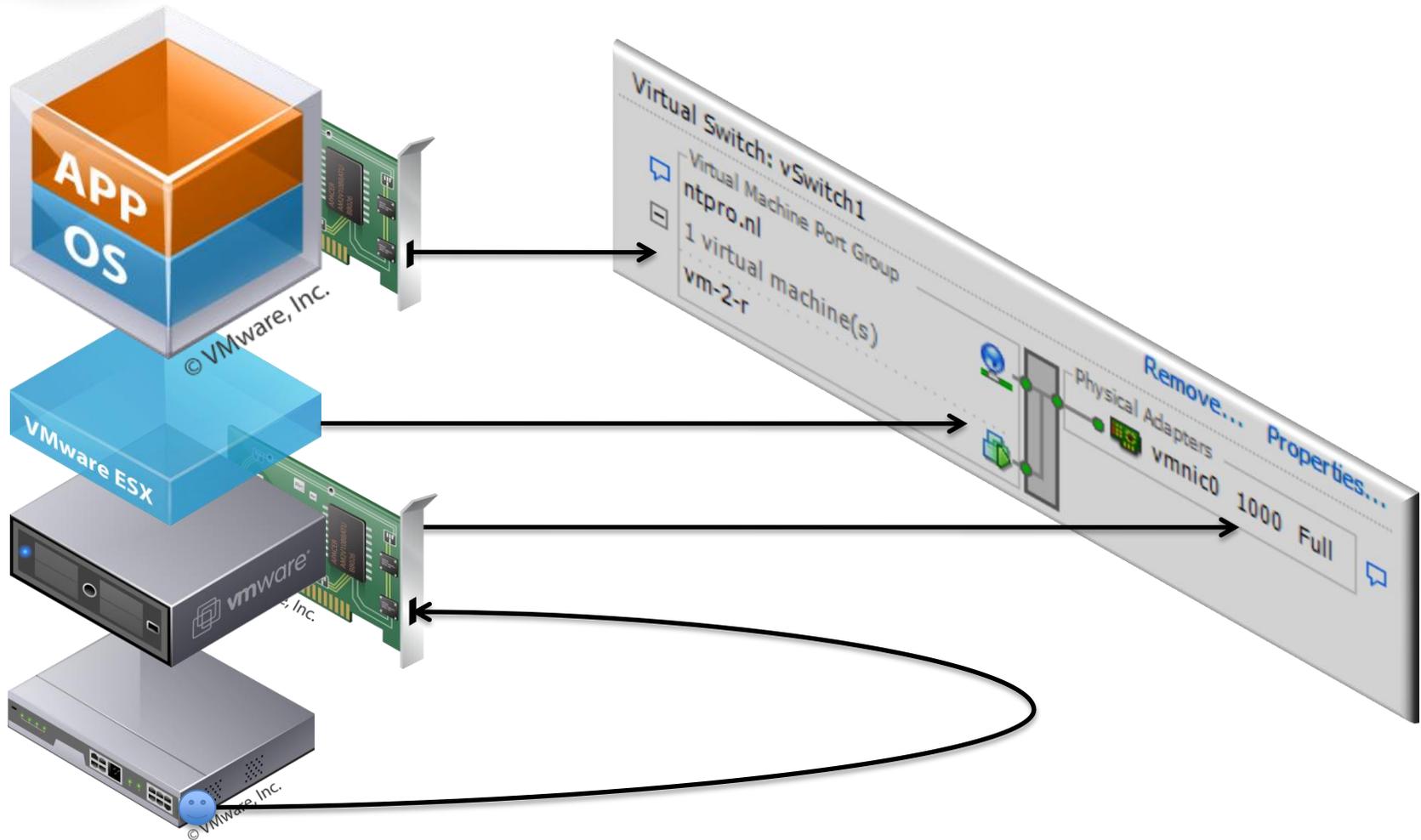


STORAGE TROUBLESHOOTING - RECAP

- If $KAVG/cmd > 3$ mSec or $DAVG/cmd > 20$ mSec there might be a storage performance problem.
- Check alignment on the array, VMFS and in the guest OS.
- Monitor the number of reservation conflicts per second and be careful with snapshots.
- Pay attention to drive types, the more drives you use the more IOPS you will get.
- When creating an VMFS, give it the right size and keep in mind how many virtual machines you want to host on that datastore.
- When choosing a block size, stick to it.



NETWORK TROUBLESHOOTING – THE NETWORK STACK





NETWORK TROUBLESHOOTING – DROPPED PKT

Receive packets might be dropped at the virtual switch if the virtual machine's network driver runs out of receive (Rx) buffers, that's a buffer overflow.

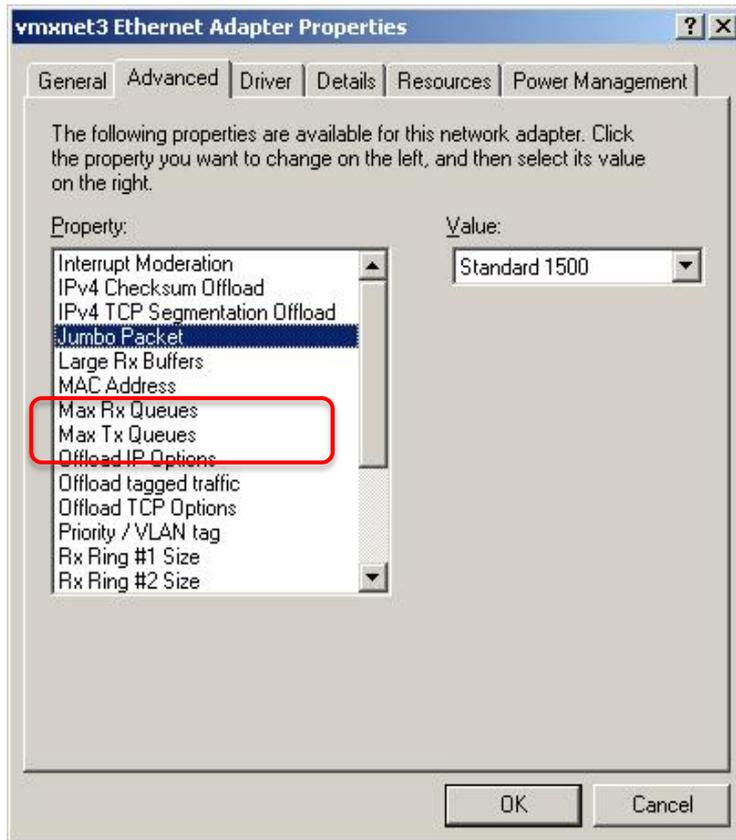
The dropped packets (%DRPR) may be reduced by increasing the Rx buffers for the virtual network driver.

```
esx4-r.ntpro.local - KITTY
11:29:32pm up 7:09, 139 worlds; CPU load average: 0.01, 0.10, 0.05
```

PORT-ID	USED-BY	TEAM-PNIC	DNAME	PKTTX/s	MbTX/s	PKTRX/s	MbRX/s	%DRPTX	%DRPR
16777217	Management	n/a	vSwitch0	0.00	0.00	0.00	0.00	0.00	0.0
16777218	vmnic3	-	vSwitch0	0.40	0.00	4.36	0.01	0.00	0.0
16777219	vmnic4	-	vSwitch0	5.94	0.03	7.72	0.01	0.00	0.0
16777220	vmk0	vmnic3	vSwitch0	0.40	0.00	1.98	0.00	0.00	0.0
16777221	4096:vswif0	vmnic4	vSwitch0	5.94	0.03	3.56	0.00	0.00	0.0
33554433	Management	n/a	DvsPortset-0	0.00	0.00	0.00	0.00	0.00	0.0
33554434	vmnic1	-	DvsPortset-0	0.00	0.00	2.97	0.00	0.00	0.0
33554435	vmnic2	-	DvsPortset-0	0.40	0.00	3.17	0.00	0.00	0.0
33554436	4255:vm-2-r ethernet	vmnic2	DvsPortset-0	7304.02	5.06	7310.75	3.50	0.00	93.8
33554438	4350:vm-6-r.eth0	vmnic2	DvsPortset-0	119569.51	57.18	7306.79	5.28	0.00	0.0



NETWORK TROUBLESHOOTING – NIC SETTINGS



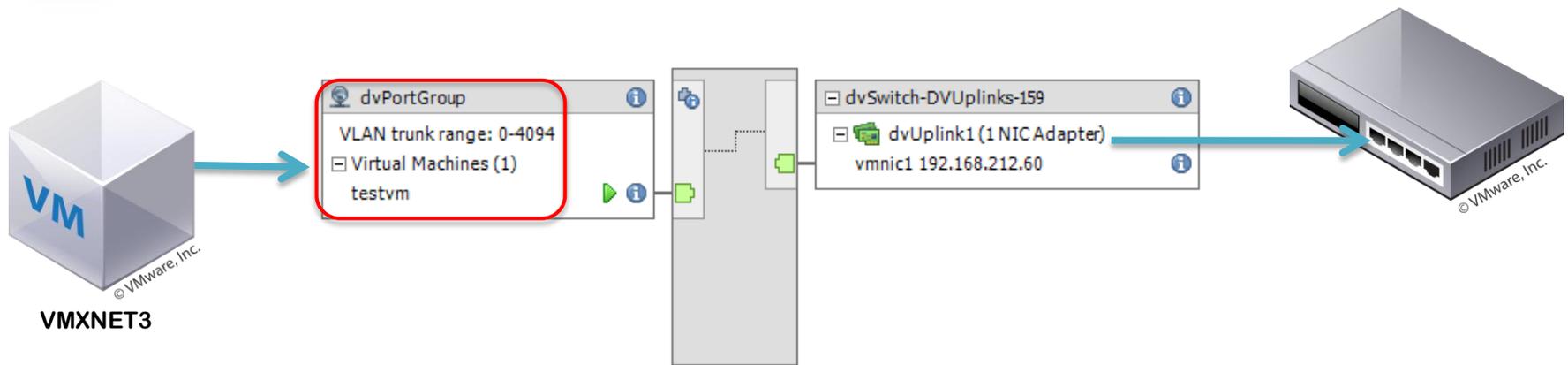
In ESX 4.1, you can configure the advanced VMXNET3 parameters from the Device Manager in the Windows guest OS.

It's possible to increase the Rx buffers for the virtual network driver here.

This also works on an Intel E1000 with the native driver installed in the guest OS.



NETWORK TROUBLESHOOTING – VLAN ID

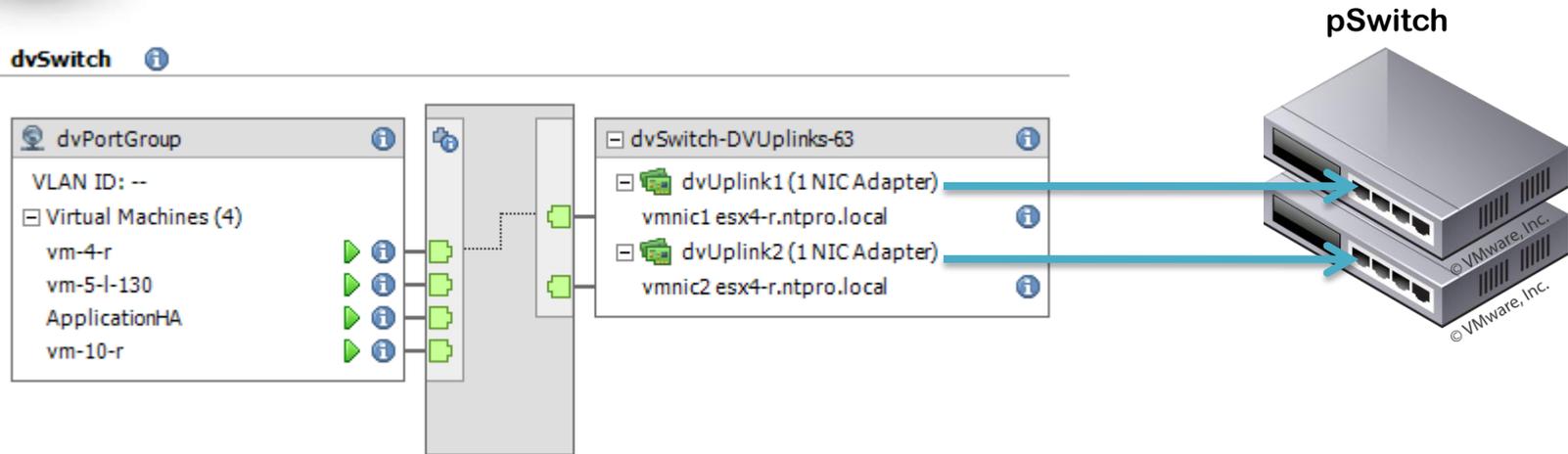


For VLAN troubleshooting, you have to create a new dvPortgroup with a VLAN trunk. This way the network traffic is delivered with a VLAN tag in the guest OS.

Now you can configure the VLAN advanced parameters for an Intel E1000 or an VMXNET3 adapter in the guest OS and specify a VLAN ID. This allows you to hop between VLANs.



NETWORK TROUBLESHOOTING – LOAD BASED TEAMING



LBT reshuffles port binding dynamically based on load and dvUplinks usage to make an efficient use of the available bandwidth.

When Load Based Teaming reassigns ports, the MAC address change to a different pSwitch port. The pSwitch must allow for this.



NETWORK TROUBLESHOOTING – LOAD BASED TEAMING

```
esx4-r.ntpro.local - KITTY
9:36:30pm up 6:23, 153 worlds; CPU load average: 0.14, 0.17, 0.09
```

PORT-ID	USED-BY	TEAM-PNIC	DNAME	PKTTX/s	MbTX/s	PKTRX/s	MbRX/s	%DRPTX	%DRPR
16777217	Management	n/a	vSwitch0	0.00	0.00	0.00	0.00	0.00	0.0
16777218	vmnic3	-	vSwitch0	0.40	0.00	4.74	0.00	0.00	0.0
16777219	vmnic4	-	vSwitch0	9.29	0.07	12.45	0.02	0.00	0.0
16777220	vmk0	vmnic3	vSwitch0	0.40	0.00	1.19	0.00	0.00	0.0
16777221	4096:vswif0	vmnic4	vSwitch0	9.29	0.07	7.12	0.02	0.00	0.0
33554433	Management	n/a	vSwitch1	0.00	0.00	0.00	0.00	0.00	0.0
33554434	vmnic0	-	vSwitch1	0.00	0.00	1.58	0.00	0.00	0.0
33554436	5312:vm-2-r	vmnic0	vSwitch1	0.00	0.00	0.99	0.00	0.00	0.0
50331649	Management	n/a	DvsPortset-0	0.00	0.00	0.00	0.00	0.00	0.0
50331650	vmnic1	-	DvsPortset-0	123043.52	90.96	53780.92	44.68	2.29	0.0
50331651	vmnic2	-	DvsPortset-0	112180.35	73.94	30960.39	324.65	21.16	0.0
50331653	4351:vm-4-r ethernet	vmnic2	DvsPortset-0	142389.93	93.85	1.98	0.00	0.00	0.0
50331654	4328:ApplicationHA e	vmnic1	DvsPortset-0	124522.00	77.71	51870.69	43.45	0.00	1.8
50331655	4368:vm-10-r etherne	vmnic1	DvsPortset-0	1431.44	15.12	954.62	0.44	0.00	0.0
50331657	5287:vm-5-1-130.eth0	vmnic2	DvsPortset-0	0.40	0.00	30895.75	324.92	0.00	0.2

LBT will only move a flow when the mean send or receive utilization on an uplink exceeds 75 percent of capacity over a 30-second period. LBT will not move flows more often than every 30 seconds. Enable PortFast mode for the physical switch ports facing the ESX Server.



NETWORK TROUBLESHOOTING - RECAP

- **Enable PortFast mode for the physical switch ports facing the ESXi Server.**
- **Disable STP for the physical switch ports facing the ESX Server.**
- **Use the VMXNET3 virtual network card wherever possible.**

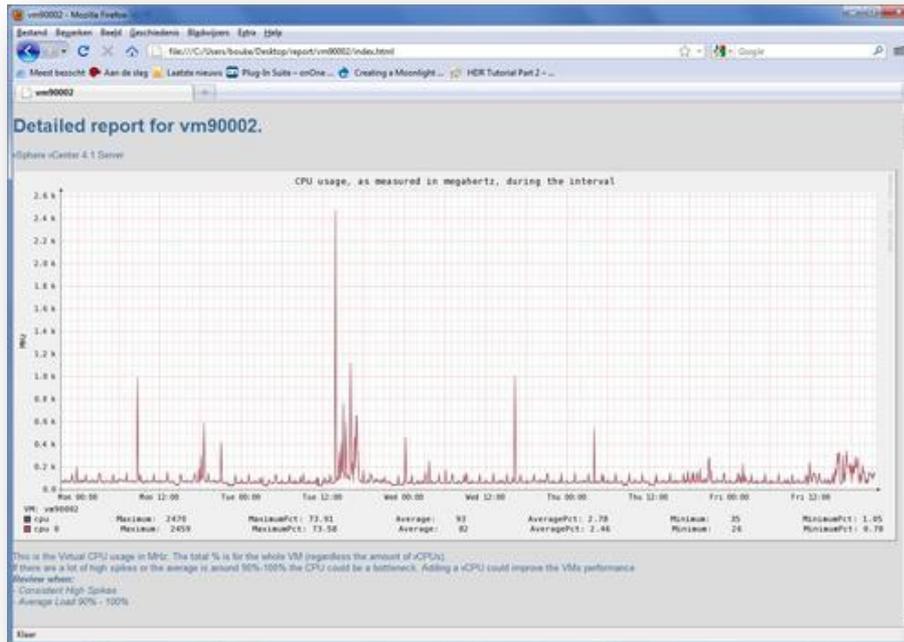


TROUBLESHOOTING TOOLS

- **Veeam Monitor**
- **VMTurbo Watchdog**
- **Quest vFoglight**
- **VKernel Capacity Analyzer**
- **VESI VMware Community PowerPack**
- **VMware Health Check Analyzer**
- **Bouke Groenescheij -> Graph-VM**
- **Esxplot and perfmon**
- **Rob de Veij - RVTools**
- **Xangati for ESX**



TROUBLESHOOTING TOOLS – GRAPH-VM

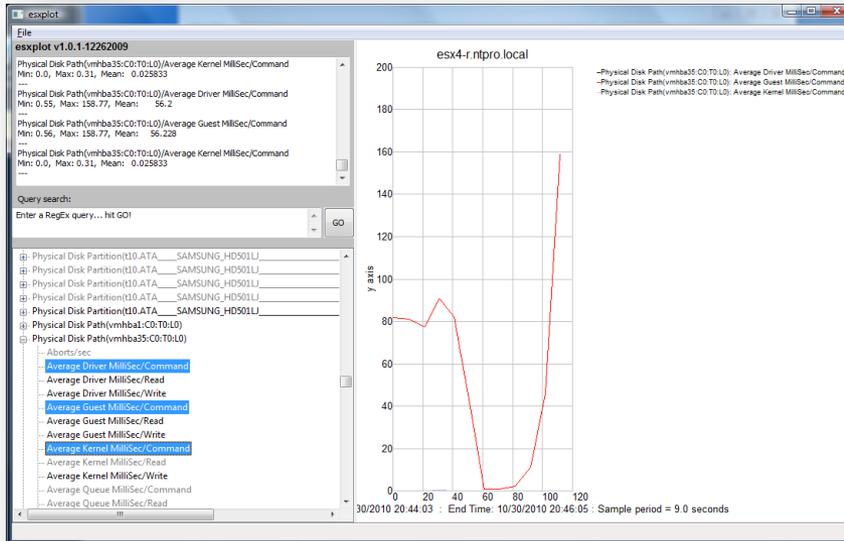


<http://www.jume.nl>

Bouke Groenescheij has created a framework of scripts which are able to produce some real nice graphs. Graph-VM uses PowerShell to gather the information and creates reports with the RDDTool.



TROUBLESHOOTING TOOLS – ESXPLOT



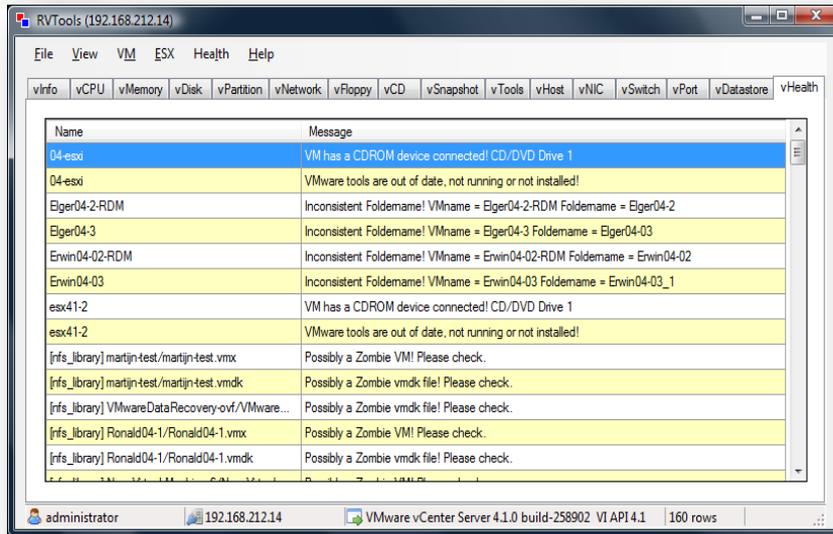
<http://labs.vmware.com>

The following command would run esxtop in batch mode, updating all statistics to the file perfstats.csv every 10 seconds for 360 iterations (a total of 60 minutes) before exiting:

```
esxtop -a -b -d 10 -n 360 > perfstats.csv
```



TROUBLESHOOTING TOOLS - RVTOOLS

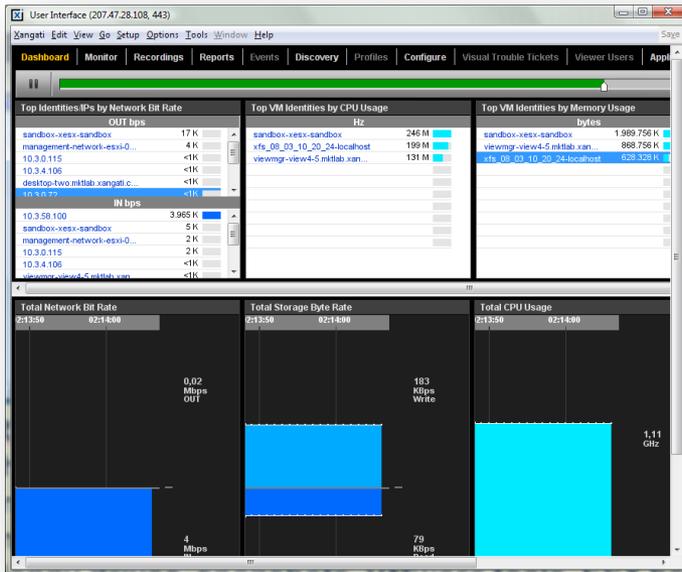


<http://www.robware.net>

RVTools is a windows .NET 2.0 application which uses the VI SDK to display information about your virtual machines and ESX hosts. RVTools is able to list information about cpu, memory, disks, nics, cd-rom, floppy drives, snapshots, VMware tools, ESX hosts, nics, datastores, switches, ports and health checks.



TROUBLESHOOTING TOOLS - XANGATI



<http://xangati.com>

Xangati for ESX is a Free tool designed for smaller scale environments with only a few ESX/ESXi hosts. It offers continuous, real-time visibility into over 100 metrics on an ESX/ESXi host and its VMs activity, including communications, CPU, memory, disk, and storage latency.



THANK YOU - QUESTIONS

This presentation is available for download at <http://www.ntpro.nl> and <http://www.vmug.nl>

Don't forget to fill out the Session Evaluation.

