vSphere Performance Troubleshooting and Root Cause Analysis (Compute)

Brett Guarino, VMware, Inc.
Rebecca Fitzhugh, Rubrik
Disclaimer

This presentation may contain product features or functionality that are currently under development.

This overview of new technology represents no commitment from VMware to deliver these features in any generally available product.

Features are subject to change, and must not be included in contracts, purchase orders, or sales agreements of any kind.

Technical feasibility and market demand will affect final delivery.

Pricing and packaging for any new features/functionality/technology discussed or presented, have not been determined.
Brett Guarino

@Brett_Guarino

Principal Instructor at VMware

VCDX #256
esxtop Overview
esxtop

- **esxtop** is the primary real-time performance monitoring tool for vSphere
  - It can be run from an ESXi host local command line as `esxtop`
  - It can be run remotely from vCLI as `resxtop`
- Designed to work like the **top** performance utility in Linux
- The key performance indicators are viewed on individual resource screens by entering the appropriate keys. Commands are case sensitive

<table>
<thead>
<tr>
<th>Key</th>
<th>Screen Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>CPU screen (default)</td>
</tr>
<tr>
<td>m</td>
<td>Memory screen</td>
</tr>
<tr>
<td>d</td>
<td>Disk (adapter) screen</td>
</tr>
<tr>
<td>u</td>
<td>Disk (device) screen</td>
</tr>
<tr>
<td>v</td>
<td>Virtual disk view (lowercase v)</td>
</tr>
<tr>
<td>n</td>
<td>Network screen</td>
</tr>
<tr>
<td>f/F</td>
<td>Add or remove statistic columns</td>
</tr>
<tr>
<td>v</td>
<td>Virtual machine view (uppercase V)</td>
</tr>
<tr>
<td>h</td>
<td>Help</td>
</tr>
<tr>
<td>q</td>
<td>Quit</td>
</tr>
</tbody>
</table>
CPU Key
Performance Indicators
CPU Key Performance Indicators

**CPU Key Performance Indicators for ESXi Hosts**
- Ready Time
- Utilization
- Load Average

**CPU Key Performance Indicators for Virtual Machines**
- Ready Time (%RDY)
- Co-Stop (%CSTP)
- Swap Wait (%SWPWT)
- Memory Limited (%MLMTD)
Using `resxtop` Interactively

**To use `resxtop` in interactive mode:**

- Log in to a system installed with VMware vSphere Command-Line Interface.
- Run `resxtop` with one or more connection parameters.
  ```
  # resxtop --server vc01.vmeduc.com --username administrator -- vihost esxi01.vmeduc.com
  ```
- In the `resxtop` window, enter a character to change the screen or behavior. Commands are case-sensitive.

**Sample `resxtop` commands:**

- Enter `c` for CPU view (default).
- Enter `m` for Memory view.
- Enter `n` for Network view.
- Enter `u` for Storage (disk) device view.
- Enter `f` to specify columns to display.
- Enter `o` to order the columns displayed.
Example: Identifying CPU Constraint

The %USED and %RDY columns of the resxtop command output indicate CPU overcommitment.
The CPU scheduler is crucial to providing good performance in a consolidated environment.

The CPU scheduler has multiple functions:
- Schedules virtual CPUs (vCPUs) on physical CPUs
- Enforces the proportional-share algorithm for CPU usage
- Supports symmetric multiprocessing (SMP) virtual machines
- Uses relaxed co-scheduling for SMP virtual machines

The CPU scheduler is aware of NUMA, processor, and cache topology.
Host CPU Co-Scheduler World
CPU Scheduler Features

The CPU scheduler allocates CPU resources and coordinates CPU usage.

The CPU scheduler uses dynamic and transparent CPU resource allocation:
- Schedules vCPUs on physical CPUs.
- Checks physical CPU utilization every 2 to 40 milliseconds and migrates vCPUs as necessary.

The CPU scheduler enforces the proportional-share algorithm for CPU usage:
- Hosts time-slice physical CPUs across all virtual machines when CPU resources are overcommitted.
- Prioritizes each vCPU by resource allocation settings: shares, reservations, and limits.
About Worlds

A world is an execution context that is scheduled on a processor. A world is like a process in conventional operating systems.

A virtual machine is a collection, or group, of worlds:
- One for each vCPU
- One for the virtual machine’s mouse, keyboard, and screen (MKS)
- One for the virtual machine monitor (VMM)

The CPU scheduler chooses which world to schedule on a processor.
Right Size Virtual Machine vCPUs

What is Co-Stop?
When a VM with multiple vCPUs must stop processing on one or more vCPUs.

Why does Co-Stop Occur?
The fastest sibling vCPU stops itself when it’s slowest sibling vCPU on the VM violates a threshold. This is due to “skew” between sibling vCPUs. The vCPUs co-start when the slowest sibling begins to make progress. It progresses because scheduling opportunities are available once the fastest vCPUs are co-stopped.

How do I resolve CPU Co-Stop issues?
Right size your VM’s vCPUs. When in doubt, mimic the physical host CPU topology to take advantage of physical/virtual NUMA.

Consider “Wide and Flat” vCPU allocations.
For configurations for VMs with greater than 8 vCPUs, allocate “X” number of virtual sockets and a single virtual core.
Many factors contribute to CPU performance:

**Idling virtual machines**: Consider the overhead of delivering guest timer interrupts.

**CPU affinity**: CPU affinity constrains the scheduler and can cause an imbalanced load.

**SMP virtual machines**: Some co-scheduling overhead is incurred.

**Insufficient CPU resources for demand**: If CPU contention exists, the scheduler forces vCPUs of lower-priority virtual machines to queue their CPU requests in deference to higher-priority virtual machines.
CPU Performance Analysis Using resxtop

The resxtop command provides much of the same information as the vSphere Web Client but supplements the output with more detailed information.

In resxtop, a group refers to a resource pool, a running virtual machine, or a non-virtual machine world. For worlds belonging to a virtual machine, statistics for the running virtual machine are shown.

Per-group statistics shown by resxtop include:

- PCPU USED(%)  
- %USED  
- %SYS  
- %RDY  
- %WAIT  
- %CSTP  
- %MLMTD  
- NWLD
Using `resxtop` to View CPU Metrics per Virtual Machine

- `resxtop` includes commands used in interactive mode to customize command output.
- Enter uppercase `v` to filter the output to show only virtual machines.
Warning Sign: Ready Time

Ready time is the amount of time that the vCPU waits for the physical CPU to become available.

vCPUs are allocated CPU cycles on an assigned physical CPU based on the proportional-share algorithm enforced by the CPU scheduler:

- If a vCPU tries to execute a CPU instruction while no cycles are available on the physical CPU, the request is queued.
- A physical CPU with no available cycles might be due to high load on the physical CPU or a higher-priority vCPU receiving preference.

Ready time can affect performance of the guest operating system and its applications in a virtual machine.
Example: Identifying CPU Overcommitment

The %USED and %RDY columns of the `resxtop` command output indicate CPU overcommitment.

<table>
<thead>
<tr>
<th>ID</th>
<th>GID NAME</th>
<th>VMEM</th>
<th>%USED</th>
<th>%SYS</th>
<th>%CPU/L</th>
<th>%RUN</th>
<th>%WAIT</th>
<th>%RDY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>idle</td>
<td>2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>system</td>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>console</td>
<td>1</td>
<td>4.78</td>
<td>0.00</td>
<td>0.01</td>
<td>4.79</td>
<td>93.80</td>
<td>1.47</td>
</tr>
<tr>
<td>7</td>
<td>helper</td>
<td>13</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
<td>130.00</td>
<td>0.02</td>
</tr>
<tr>
<td>8</td>
<td>drivers</td>
<td>7</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
<td>70.00</td>
<td>0.02</td>
</tr>
<tr>
<td>12</td>
<td>vmware-vmkauthd</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
</tr>
<tr>
<td>15</td>
<td>memhog-linux-sm</td>
<td>7</td>
<td>64.68</td>
<td>0.00</td>
<td>0.16</td>
<td>64.84</td>
<td>477.30</td>
<td>158.32</td>
</tr>
<tr>
<td>17</td>
<td>kernelcompile-u</td>
<td>5</td>
<td>64.44</td>
<td>0.01</td>
<td>0.11</td>
<td>64.74</td>
<td>365.69</td>
<td>69.51</td>
</tr>
<tr>
<td>18</td>
<td>memhog-linux</td>
<td>5</td>
<td>65.62</td>
<td>0.00</td>
<td>0.14</td>
<td>65.76</td>
<td>378.65</td>
<td>55.93</td>
</tr>
</tbody>
</table>
Viewing CPU Metrics in vSphere Web Client

Use vSphere Web Client CPU performance charts to monitor CPU usage for hosts, clusters, resource pools, virtual machines, and VMware vSphere vApps.
Using resxtop to View Single CPU Statistics

Enter lowercase e to show all the worlds associated with a single virtual machine.
Identifying host CPU issues requires a step-by-step process.

1. Measure host’s CPU usage.
2. Is average usage > 75% or peak usage > 90%? 
   - Yes: Locate VM with highest CPU usage.
   - No: Return to basic troubleshooting flow.
3. Is VM’s Ready value > 200ms (10%) for any vCPU? 
   - Yes: Host CPU saturation exists
   - No: Return to basic troubleshooting flow.
Several options are available to resolve host CPU saturation

1. Reduce the number of virtual machines running on the host.

2. Increase the available CPU by adding resources to a VMware vSphere DRS cluster.

3. Increase the efficiency with which virtual machines use CPU resources.

4. Use resource controls to direct available resources to critical virtual machines.
Memory Key Performance Indicators
Using `esxtop` to Monitor Memory Usage

`esxtop` offers several options for monitoring memory usage.

Enter `m` to display the memory screen.

Possible States: High, Clear, Soft, Hard, Low

PMEM: installed MB
VMKMEM: managed MB
MINFREE: calculated MB

MB of free RAM
### esxtop Memory State Thresholds

What is MinFree?

<table>
<thead>
<tr>
<th>Memory State</th>
<th>Threshold</th>
<th>Actions Performed</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>300% of minFree</td>
<td>Break Large Pages (wait for next TPS run)</td>
</tr>
<tr>
<td>Clear</td>
<td>100% of MinFree</td>
<td>Break Large Pages and active call TPS to collapse pages</td>
</tr>
<tr>
<td>Soft</td>
<td>64% of minFree</td>
<td>TPS + Balloon</td>
</tr>
<tr>
<td>Hard</td>
<td>32% of minFree</td>
<td>TPS + Compress + Swap</td>
</tr>
<tr>
<td>Low</td>
<td>16% of minFree</td>
<td>Compress + Swap + Block</td>
</tr>
</tbody>
</table>
The memory key performance indicators for hosts are based on:

- Ballooning
- Memory compression
- Host-level (hypervisor) swapping
Host Memory Shortage: Ballooning in the Guest Operating System

The memory balloon driver (*vmmemctl*) collaborates with the virtual machine to reclaim pages that are considered least valuable by the guest operating system. *vmmemctl* is the only way to reclaim unused memory from a guest operating system.

Initially the Balloon driver is inactive

Host memory pressure calls the Balloon driver to reserve memory in the guest (inflating) and forces the guest to page out memory to its swap space

Host memory pressure subsides and the Balloon driver stops reserving memory in the guest (deflating). The guest will page in from the swap space as needed
The goal of ballooning is to make the guest operating system aware of the low memory status of the host so that the guest operating system can free some of its memory.

Ballooning preferentially selects free or idle virtual machine memory.

Maximum 65% of VM memory can be ballooned
With memory compression, the VMware ESXi host stores pages in a compression cache rather than swapping them out to disk.
ESXi uses host-level swapping when ballooning, and memory compression are not sufficient to reclaim memory.

Host-level swapping randomly selects guest physical memory to reclaim, potentially including virtual machine active memory.
In the `esxtop` memory screen add the `J` field for virtual machine ballooning activity.

**Memory Balloon Statistics for the Host**

<table>
<thead>
<tr>
<th>GID</th>
<th>NAME</th>
<th>MEM2</th>
<th>GRANT</th>
<th>TOTAL</th>
<th>TCHD</th>
<th>TCHD_W</th>
<th>MCTL2</th>
<th>MCTLTGT</th>
<th>MCTLMAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>5223</td>
<td>VMTest01</td>
<td>988.00</td>
<td>978.31</td>
<td>1007.99</td>
<td>484.12</td>
<td>19.76</td>
<td>Y</td>
<td>0.00</td>
<td>633.95</td>
</tr>
<tr>
<td>5226</td>
<td>VMTest02</td>
<td>988.00</td>
<td>978.31</td>
<td>1007.99</td>
<td>484.12</td>
<td>19.76</td>
<td>Y</td>
<td>0.00</td>
<td>633.95</td>
</tr>
<tr>
<td>387205</td>
<td>Workload1</td>
<td>256.00</td>
<td>215.77</td>
<td>249.90</td>
<td>238.08</td>
<td>238.08</td>
<td>H</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>387206</td>
<td>Workload2</td>
<td>256.00</td>
<td>215.41</td>
<td>250.36</td>
<td>250.88</td>
<td>250.88</td>
<td>N</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>387207</td>
<td>Workload3</td>
<td>256.00</td>
<td>215.95</td>
<td>249.55</td>
<td>253.41</td>
<td>253.41</td>
<td>N</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>387208</td>
<td>Workload4</td>
<td>256.00</td>
<td>215.95</td>
<td>249.55</td>
<td>253.41</td>
<td>253.41</td>
<td>N</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>387209</td>
<td>Workload5</td>
<td>256.00</td>
<td>215.03</td>
<td>249.91</td>
<td>250.66</td>
<td>250.66</td>
<td>N</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>387210</td>
<td>Workload6</td>
<td>256.00</td>
<td>215.90</td>
<td>249.98</td>
<td>250.86</td>
<td>250.86</td>
<td>N</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
In the `esxtop` memory screen add the `Q` field for virtual machine compression activity.

<table>
<thead>
<tr>
<th>GID</th>
<th>NAME</th>
<th>MEMSZ</th>
<th>GRANT</th>
<th>CNSTM</th>
<th>SZTGT</th>
<th>TCHD</th>
<th>TCHD W</th>
<th>Cache Sz</th>
<th>Cache Used</th>
<th>ZIP/s</th>
<th>UNZIP/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>9198306</td>
<td>Nostalgia05</td>
<td>2048.00</td>
<td>8.00</td>
<td>8.00</td>
<td>20.38</td>
<td>348.16</td>
<td>348.16</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>9198584</td>
<td>Nostalgia07</td>
<td>2048.00</td>
<td>8.00</td>
<td>8.00</td>
<td>20.36</td>
<td>471.04</td>
<td>471.04</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>9201763</td>
<td>Workload01</td>
<td>512.00</td>
<td>6.00</td>
<td>6.00</td>
<td>18.32</td>
<td>286.72</td>
<td>286.72</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>9202021</td>
<td>Workload02</td>
<td>512.00</td>
<td>5.04</td>
<td>4.07</td>
<td>16.20</td>
<td>286.72</td>
<td>286.72</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

- **Memory Compression Statistics for the Host**
- **Calculated compression cache size**
- **Actively compressing memory per VM**
- **Memory compressed in MB per VM**
- **Accessing compressed memory per VM**
Host Memory Shortage KPI: Swapping Activity in esxtop (Memory screen)

In the esxtop memory screen add the K field for virtual machine swapping activity.

- Total Memory Swapped for All Virtual Machines on Host
- Total Memory Swap Rate for All Virtual Machines on Host
- Swap Reads per Second
- Swap Writes per Second
- Swap Space Currently Used
- Swap Space Target
Host Memory Shortage KPI: Swapping Activity in esxtop (CPU Screen)

In esxtop, the CPU screen can indicate memory swapping is occurring.
Host Memory Shortage KPI: Ballooning and Swapping Activity Using vCenter Charts

- Ballooning activity for a host
- Swapping and Active Memory during same period
Host Memory Shortage Solutions

Host memory shortages can be resolved in the following ways:

- Reduce the level of memory overcommitment → vMotion
- Enable the balloon driver in all virtual machines → Install VMware Tools
- Add memory to the host
- Reduce memory reservations → Configured VM memory can be overcommitted, not Reserved VM memory
- Use resource controls to dedicate memory to critical virtual machines → Only as a last resort
Questions?
PLEASE FILL OUT YOUR SURVEY.

Take a survey and enter a drawing for a VMware company store gift card.
POSSIBLE BEGINS WITH YOU

THANK YOU!

#vmworld  #VIN1372BU