ORACLE
SOLARIS

Virtualization Monitoring in Oracle Solaris Zones

ORACLE WHITE PAPER | AUGUST 2015

ORACLE

Table of Contents

Introduction

Modern operating systems like Oracle Solaris offer multiple forms of virtualization. Oracle has the following virtualization products for Oracle Solaris: Oracle VM Server for SPARC, Oracle VM Server for x86, and Oracle Solaris Zones, a feature of Oracle Solaris 11. Since Oracle Solaris 11.2, the Oracle Solaris Zones framework offers three types of zones: **solaris**, **solaris10**, and **solaris-kz**.

- solaris is the default zone of Oracle Solaris Zones. Such a zone shares the kernel with the global zone.
- solaris10 is used for a zone containing an Oracle Solaris 10 operating environment.
- solaris-kz is used for a zone that has its own kernel, which can have a version different from the kernel in the global zone.

Often it is necessary to restrict and monitor resource utilization of the virtual machines. This paper describes the possibilities to restrict resource usage of Oracle Solaris Zones, monitor the different zone types of Oracle Solaris Zones from within the nonglobal zone and from within the global zone. There are different tools that help you to observe the resources of the complete system. This paper focuses on the monitoring of two resources: physical memory and CPUs. The following topics are discussed:

- Use of resource controls to control CPU resource usage and memory usage of zones
- Oracle Solaris tools like the `zonestat`, `poolstat`, `kstat`, `rcapstat`, or `prstat` commands for monitoring zones
- Monitoring of different zone types from within global and nonglobal zones
- Differences in monitoring zones with CPU caps or dedicated CPUs
- Use of the OpenStack dashboard to get an overview of your zones

Definitions and Terms

The following terms are used throughout this paper.

Term	**Definition**
Default zone	The default type of Oracle Solaris Zones used in Oracle Solaris 11. Such a zone shares the kernel with the global zone.
Oracle Solaris 10 zone	This type of Oracle Solaris Zones contains an Oracle Solaris 10 operating environment running within an Oracle Solaris 11 global zone.
Kernel Zones	This feature of Oracle Solaris contains its own kernel and therefore can run a different kernel version than the global zone.
CPU	A hardware thread or strand.

Resource Controlling of Oracle Solaris Zones

Modern processors usually consist of multiple cores, where each core has a number of hardware threads. A hardware thread is also known as a strand. For example, Oracle's SPARC T5 processor has 16 cores, and each core has 8 hardware threads or 128 hardware threads per processor. Oracle Solaris monitoring tools have commands like `mpstat` and `kstat` that show each hardware thread as a single CPU.

Oracle Solaris offers the possibility to assign resources to Oracle Solaris Zones. If none of the following options is used, a zone can use all the memory and CPUs that are available to the global zone.

- **Dedicated CPUs.** Assign a number or range of CPUs exclusively to a zone.
- **Capped CPU.** Restrict the usage of CPU resources for the zone.
- **Capped memory.** Restrict the usage of memory for the zone.
- **Virtual CPU.** Dedicate a number of CPUs to the kernel zone.
- **CPU shares.** Assign a quantity of shares to a zone which the fair share scheduler (FSS) uses to assign appropriate CPU resources to the zone.

This paper covers both dedicated and capped CPUs as well as capped memory. For further information about virtual CPUs or CPU shares, please refer to the Oracle Solaris documentation.[12]

Dedicated CPUs

1 Oracle Corporation, "solaris-kz Only: virtual-cpu Resource," December 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36848/gnzme.html.

2 Oracle Corporation, "Setting Zone-Wide Resource Controls," December 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36848/z.config.ov-13.html.

With dedicated CPUs you can assign a fixed number or a range of dedicated CPUs to a zone. If a fixed number of CPUs is assigned to the zone, the zone has exclusive access to these CPUs. When booting the zone, a resource pool is created that is assigned exclusively to the zone. A processor set is bound to this resource pool, which includes the chosen number of CPUs. In case not enough CPUs are available, the zone won't boot. As the CPUs are put in a resource pool exclusively assigned to the zone, no other zones or the global zone can use these CPUs.

The following example shows how to assign one dedicated CPU to the zone (myzone) with the `zonecfg` command.

```
zonecfg:myzone> add dedicated-cpu
zonecfg:myzone:dedicated-cpu> set ncpus=1
zonecfg:myzone:dedicated-cpu> end
zonecfg:myzone> verify
zonecfg:myzone> info dedicated-cpu
dedicated-cpu:
       ncpus: 1
       cpus not specified
       cores not specified
       sockets not specified
zonecfg:myzone>
```

It is also possible to assign a range of CPUs to a zone. In this case Oracle Solaris creates a resource pool exclusively assigned to the zone with a processor set bound to it. The processor set will at least contain the number of CPUs defined in the range as minimum. In case the zone needs more CPU resources, Oracle Solaris can put additional CPUs in the processor set. In the opposite case Oracle Solaris can remove CPUs from the processor set until the minimum number is reached.

Dedicated CPUs are set by using the `zonecfg` command and by adding the dedicated-cpu resource to the zone.

The following example demonstrates how to assign a range from one to two dedicated CPUs to the zone (myzone) with the `zonecfg` command. If you assign a range with the dedicated-cpu resource, it is necessary to enable the service **system/pools/dynamic**.

```
zonecfg:myzone> add dedicated-cpu
zonecfg:myzone:dedicated-cpu> set ncpus=1-2
zonecfg:myzone:dedicated-cpu> end
zonecfg:myzone> verify
zonecfg:myzone> info dedicated-cpu
dedicated-cpu:
       ncpus: 1-2
       cpus not specified
       cores not specified
       sockets not specified
zonecfg:myzone>
```

The dedicated-cpu resource can be combined with the capped-memory resource, but not with the capped-cpu resource. When using the dedicated-cpu resource for Oracle's Kernel Zones, a feature of Oracle Solaris 11.2, defining a range is not possible. Only a fixed number of CPUs can be assigned to an instance of Kernel Zones with the dedicated-cpu resource. You can the assign a specific number of CPUs, CPU cores, or sockets to the zone.

Refer to the Oracle Solaris documentation[3] for more information about the dedicated-cpu resource.

Capped CPU

This means the CPU usage of the zone is capped at the number of CPUs specified in the capped-cpu resource. The number of CPUs set in the capped-cpu resource can be a non-integer number. This can be interpreted as percentage of CPUs, where 1 means 100 percent of a CPU. A value of 1.5 means 150 percent. The full usage of this cap can be achieved by having one CPU fully utilized and another CPU with 50 percent utilization. Having two CPUs fully utilized isn't possible in this case, as this exceeds the cap of 1.5.

Every zone is assigned to a resource pool. If the dedicated-cpu resource is not used and the zone is not explicitly assigned to a resource pool, the zone is assigned to the default resource pool. All CPUs in the assigned resource pool are visible from within the zone. When using capped CPU, Oracle Solaris restricts the CPU usage of the zone to the number of CPUs set by the capped-cpu resource.

The capped CPU is set using the `zonecfg` command by adding the capped-cpu resource to the zone.

The following example shows how a CPU cap of 1.5 is set for the zone (myzone) using the `zonecfg` command.

```
zonecfg:myzone> add capped-cpu
zonecfg:myzone:capped-cpu> set ncpus=1.5
zonecfg:myzone:capped-cpu> end
zonecfg:myzone> verify
zonecfg:myzone> info capped-cpu
capped-cpu:
        [ncpus: 1.50]
zonecfg:myzone>
```

The capped-cpu resource can be combined with the capped-memory resource but not with the dedicated-cpu resource.

Refer to the Oracle Solaris documentation[4] for more information about the capped-cpu resource.

Capped Memory

In addition to restricting CPU usage in the zone, it is also possible to restrict memory usage. This is done via the capped-memory resource. The different Oracle Solaris Zones types use this resource in a different way.

In default and Oracle Solaris 10 zones, the usage of physical memory is restricted to the amount specified in the capped-memory resource, and if the zone reaches its physical memory limit, paging will occur. Memory is not exclusively assigned to the zone, but it is also available for the global zone and other zones.

With Kernel Zones, the amount specified in the capped-memory resource is statically assigned to an instance of the kernel zone. This memory is not available for the global zone or other zones.

The capped memory is set using the `zonecfg` command by adding the capped-memory resource to the zone. It is also possible to set caps for swap and locked memory.

Swap in the resource capped-memory means physical memory plus swap space (e.g., disk swap). If a zone reaches the swap limit, memory allocations inside the zone will fail.

3 Oracle Corporation, "dedicated-cpu Resource," December 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36848/gejlw.html.

4 Oracle Corporation, "capped-cpu Resource," December 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36848/gepzc.html.

The following example shows how a limit of 2 GB is set as a physical memory cap for the zone (myzone) using the `zonecfg` command. Additionally, the swap usage is capped at 3 GB.

```
zonecfg:myzone> add capped-memory
zonecfg:myzone:capped-memory> set physical=2G
zonecfg:myzone:capped-memory> set swap=3G
zonecfg:myzone:capped-memory> end
zonecfg:myzone> verify
zonecfg:myzone> info capped-memory
capped-memory:
       physical: 2G
       [swap: 3G]
zonecfg:myzone>
```

The capped-memory resource can be either combined with the dedicated-cpu resource or with the capped-cpu resource.

Refer to the Oracle Solaris documentation[5] for more information about the capped-memory resource.

Overview

The following tables summarize the preceding sections of this whitepaper.

AVAILABLE RESOURCES FOR DIFFERENT ZONE TYPES

Zone Type	**Dedicated CPU**	**Capped CPU**	**Capped Memory**
Default Oracle Solaris Zones (solaris)	Yes	Yes	Yes
Oracle Solaris 10 zones (solaris10)	Yes	Yes	Yes
Kernel Zones (solaris-kz)	Partially (only fixed number of CPUs)	Yes	Yes (required)

This table shows which zone type supports which resource types.

RESOURCES USED FOR THE SAME ZONE

	Dedicated CPU	Capped CPU	Capped Memory
Dedicated cpu	-	No	Yes
Capped cpu	No	-	Yes
Capped memory	Yes	Yes	-

This table shows which resources can be used together in the same zone.

5 Oracle Corporation, "Physical Memory Control and the capped-memory Resource," December 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36848/gejkz.html.

Monitoring Default Zones

Oracle Solaris 11 has a new tool that can be used to monitor the resource usage of zones. It is not only available in the global zone but also in the nonglobal zones.

The command `zonestat` offers many reports to choose from. For example, reports about processes, memory, CPUs, network, and shared memory. This paper discusses two reports that monitor the usage of the resources discussed in the previous chapter. These reports are processor set and memory.

Other reports, such as network, limits, or sysv (shared memory, semaphores, etc.), are available by using the `zonestat` command. Additionally, the reports *all* and *summary* are available; *summary* displays a summary report of CPUs, physical memory, virtual memory, and networks. The default report of the `zonestat` command is the summary report.

There is also an API available to use the `zonestat` command functionality within programs. This API is described in the man page for the command `libzonestat`[6]. Further information can also be found[7]

When using the `zonestat` or `libzonestat` commands, ensure that the service **system/zones-monitoring:default** is running in the global zone, so that the `zonestat` or `libzonestat` commands are able to monitor the zones.

The following section outlines the differences of the `zonestat` command when running from global zone and from within a default zone. The report in the nonglobal zone only shows the statistics for the current zone and of the complete system, or in case of CPUs, of the complete processor set the zone is using[8]. The report in the global zone shows the statistics for each zone and the global zone separately.

A complete overview of the `zonestat` command options is available in the man page[9].

Monitoring Default Zones with Dedicated CPU

In this example, a zone is created with one dedicated CPU. A Perl script is running in this zone. The output of the `prstat` command shows that Perl is using 100 percent of the CPU.

```
-bash-4.1$ prstat -c -n 1 1 1
Please wait...
   PID USERNAME  SIZE   RSS STATE   PRI NICE      TIME  CPU PROCESS/NLWP
15334 sun        11M 8292K run      50    0   1:29:29 100% perl/1
Total: 42 processes, 160 lwps, load averages: 1.00, 0.99, 0.98
```

From the nonglobal zone

The following output shows how the `zonestat` command is invoked to display the report processor set once after 60 seconds.

6 Oracle Corporation, "libzonestat man page," July 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36873/libzonestat-3lib.html.

7 Oracle Corporation, "Resource Management and Oracle® Solaris Zones Developer's Guide," July 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36849/index.html.

8 Oracle Corporation, "Using the zonestat Utility in a Non-Global Zone," May 2015. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E37628/gklcu.html.

9 Oracle Corporation, "zonestat man page," July 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36870/zonestat-1.html.

```
-bash-4.1$ zonestat -r processor-set 60 1
Collecting data for first interval...
Interval: 1, Duration: 0:01:00
PROCESSOR_SET                   TYPE  ONLINE/CPUS     MIN/MAX
myzone                 dedicated-cpu          1/1         1/1
                                ZONE  USED %USED   CAP  %CAP   SHRS  %SHR %SHRU
                             [total]  0.99 99.9%     -     -      -     -     -
                            [system]  0.00 0.00%     -     -      -     -     -
                              myzone  0.99 99.9%     -     -      -     -     -
```

The report for processor set within the nonglobal zone only shows the processor set that is used by the zone. Because the dedicated-cpu resource was added to this zone, Oracle Solaris created a resource pool with a processor set exclusively assigned to the zone. You can see the minimum, maximum, and currently assigned CPUs to this processor set, which is 1. Then three lines starting with [total], [system], and myzone follow, and they display the CPU usage (in percentage and number of CPUs). The line starting with [system] usually describes the usage of this resource (CPU) by other zones or the global zone (in case the `zonestat` command is called from a nonglobal zone). But, because the dedicated-cpu resource is used, the processor set is exclusively assigned to the zone. No other zones or the global zone use this processor set, and [system] is 0 in this case. After the line beginning with [system] there is a line for each zone using this resource. Again, this processor set is used only by this zone, so only the zone (myzone) is displayed here, showing that it uses 0.99 CPUs or almost 100 percent of the CPUs in the processor set. The [total] line is the sum of the [system] and myzone lines.

Then the zone configuration is changed so it uses one to two dedicated CPUs. The same Perl script runs on the zone and `zonestat` command, and it is invoked again.

```
-bash-4.1$ zonestat -r processor-set 60 1
Collecting data for first interval...
Interval: 1, Duration: 0:01:00
PROCESSOR_SET                   TYPE  ONLINE/CPUS     MIN/MAX
myzone                 dedicated-cpu          2/2         1/2
                                ZONE  USED %USED   CAP  %CAP   SHRS  %SHR %SHRU
                             [total]  1.00 50.1%     -     -      -     -     -
                            [system]  0.00 0.00%     -     -      -     -     -
                              myzone  1.00 50.1%     -     -      -     -     -
```

The difference is that, in the example above, MIN is 1 and MAX is 2, showing that the zone that exclusively uses this processor set can use at least one and at most two CPUs. The ONLINE column shows that currently two CPUs are assigned.

In the line starting with myzone you also can see that the zone utilizes one CPU or about 50 percent of the resources currently assigned to the zone.

From the global zone

The following output shows the same command invoked from the global zone when the zone (myzone) has only one dedicated CPU assigned.

```
-bash-4.1$ zonestat -r processor-set 60 1
Collecting data for first interval...
```

```
Interval: 1, Duration: 0:01:00
PROCESSOR_SET                    TYPE  ONLINE/CPUS     MIN/MAX
pset_default            default-pset          3/3         1/-
                                ZONE  USED %USED   CAP  %CAP   SHRS  %SHR %SHRU
                             [total]  0.14 4.81%     -     -      -     -     -
                            [system]  0.00 0.27%     -     -      -     -     -
                              global  0.13 4.53%     -     -      -     -     -

PROCESSOR_SET                    TYPE  ONLINE/CPUS     MIN/MAX
myzone                  dedicated-cpu          1/1         1/1
                                ZONE  USED %USED   CAP  %CAP   SHRS  %SHR %SHRU
                             [total]  0.99 99.9%     -     -      -     -     -
                            [system]  0.00 0.00%     -     -      -     -     -
                              myzone  0.99 99.9%     -     -      -     -     -
```

Here the report displays two processor sets. The first processor set is the default processor set and currently has three CPUs. The second processor set is the processor set used by the zone (myzone), and it shows the same output as from within the zone. If you have a look at the output for pset_default (the default processor set) you see that currently three CPUs are used by it. The [system] line shows the CPU usage by the kernel. The global line shows the CPU usage by the global zone. Any additional zones using this processor set can follow here. [total] is the sum of all the lines in this processor set.

Now take a look at the same output after the zone (myzone) is changed to use one to two dedicated CPUs.

```
-bash-4.1$ zonestat -r processor-set 60 1
Collecting data for first interval...
Interval: 1, Duration: 0:01:00
PROCESSOR_SET                    TYPE  ONLINE/CPUS     MIN/MAX
pset_default            default-pset          2/2         1/-
                                ZONE  USED %USED   CAP  %CAP   SHRS  %SHR %SHRU
                             [total]  0.12 6.30%     -     -      -     -     -
                            [system]  0.00 0.38%     -     -      -     -     -
                              global  0.11 5.92%     -     -      -     -     -

PROCESSOR_SET                    TYPE  ONLINE/CPUS     MIN/MAX
myzone                  dedicated-cpu          2/2         1/2
                                ZONE  USED %USED   CAP  %CAP   SHRS  %SHR %SHRU
                             [total]  1.00 50.1%     -     -      -     -     -
                            [system]  0.00 0.00%     -     -      -     -     -
                              myzone  1.00 50.1%     -     -      -     -     -
```

The block showing the processor set of the zone (myzone) is the same as within the zone. There's only one difference with the pset_default. Now it only uses two CPUs. This is because the server has four CPUs and two of them are currently assigned to the processor set of myzone.

Monitoring Default Zones with Capped CPU

In this example a zone is created with a CPU cap set to 1.5. A Perl script is running in this zone, which is almost fully utilizing one CPU.

From the nonglobal zone

The following output shows the `prstat` command invoked in the zone.

```
-bash-4.1$ prstat -c -n 1 1 1
Please wait...
   PID USERNAME  SIZE   RSS STATE   PRI NICE      TIME  CPU PROCESS/NLWP
  7590 sun        11M 8780K cpu1      0    0   0:01:15  24% perl/1
Total: 42 processes, 158 lwps, load averages: 0.82, 0.40, 0.24
-bash-4.1$
```

Remember that when using capped CPU the zone is in the default processor set. The CPU utilization displayed by the command `prstat` relates to all CPUs in the processor set. There are four CPUs in this processor set, and one CPU is almost fully utilized, so the `prstat` command shows 24 percent CPU utilization.

Invoking the `zonestat` command will display a report about the CPU usage in the processor set to which the zone belongs.

```
-bash-4.1$ zonestat -r processor-set 60 1
Collecting data for first interval...
Interval: 1, Duration: 0:01:00
PROCESSOR_SET                   TYPE  ONLINE/CPUS     MIN/MAX
pset_default            default-pset          4/4         1/-
                                ZONE  USED %USED   CAP  %CAP   SHRS  %SHR %SHRU
                             [total]  1.20 30.0%     -     -      -     -     -
                            [system]  0.20 5.10%     -     -      -     -     -
                              myzone  0.99 24.9%  1.50 66.5%      -     -     -
```

As the zone uses the default processor set, the CPU usage within the default processor set is displayed. You see that this processor set currently has four CPUs assigned to it. In the line [system] you see that the global zone and all other zones use 0.2 CPUs or 5.1 percent of this processor set. The zone itself uses almost one CPU, which is almost 25 percent. A cap is set to 1.5, and the zone uses 66.5 percent of this cap. The line [total] displays the sum of all lines. Although this processor set may contain other zones, you see only the usage by this zone. The usage of all other zones and the global zone is aggregated in the line [system].

From the global zone

This server has two zones installed, the default zone (myzone) and the Oracle Solaris 10 zone **solmarc10**. From within other zones the `zonestat` command doesn't show that other zones are on the server or in the same processor set. But executing the `zonestat` command from the global zone will show all zones.

```
-bash-4.1$ zonestat -r processor-set 60 1
Collecting data for first interval...
Interval: 1, Duration: 0:01:00
PROCESSOR_SET                   TYPE  ONLINE/CPUS     MIN/MAX
pset_default            default-pset          4/4         1/-
                                ZONE  USED %USED   CAP  %CAP   SHRS  %SHR %SHRU
```

```
                          [total]  1.20 30.1%     -     -      -     -     -
                        [system]  0.01 0.33%     -     -      -     -     -
                          myzone  0.99 24.9%  1.50 66.5%      -     -     -
                          global  0.14 3.63%     -     -      -     -     -
                       solmarc10  0.04 1.19%  1.50 3.19%      -     -     -
```

In the zonestat report, you see that four CPUs are currently online in the default processor set. In the line [system] you see that the system just uses 0.33 percent of the processor set. The output shows that both zones (myzone and solmarc10) have a CPU cap of 1.5. It also shows the CPU usage of each zone and the global zone in separate lines.

Monitoring Default Zones with Capped Memory

A capped-memory resource with 2 GB is added to the myzone. Previously only the zonestat report for processor sets was discussed. This section takes a look at the memory report of the `zonestat` command. The memory report shows usage and caps of physical memory, locked memory, and virtual memory (swap). If you are interested in only one of the memory types, you can specify `physical-memory`, `virtual-memory`, or `locked-memory` on the command line for the appropriate report.

From the nonglobal zone

If the physical-memory report is executed from the nonglobal zone, the following output is shown.

```
-bash-4.1$ zonestat -r memory 60 1
Collecting data for first interval...
Interval: 1, Duration: 0:01:00
PHYSICAL-MEMORY              SYSTEM MEMORY
mem_default                         15.7G
                                ZONE  USED %USED   CAP  %CAP
                             [total] 4059M 25.1%     -     -
                           [system] 3865M 23.9%     -     -
                             myzone  193M 1.20% 2048M 9.45%

VIRTUAL-MEMORY               SYSTEM MEMORY
vm_default                          23.7G
                                ZONE  USED %USED   CAP  %CAP
                             [total] 5933M 24.3%     -     -
                           [system] 5763M 23.7%     -     -
                             myzone  169M 0.69% 3072M 5.53%

LOCKED-MEMORY                SYSTEM MEMORY
mem_default                         15.7G
                                ZONE  USED %USED   CAP  %CAP
                             [total] 2727M 16.9%     -     -
                           [system] 2727M 16.9%     -     -
                             myzone     0 0.00%     -     -
```

For each memory type a block is displayed. The first block shows statistics about physical memory.

The column SYSTEM MEMORY shows the physical memory installed in the system. Then for the lines [total], [system], and the zone (myzone), the values are displayed. The line [system] shows the memory usage of the kernel and all zones except the current zone. In the line below you can observe the memory usage of the current zone. As a memory cap is set, the memory cap is shown in the column CAP. Also the percentage of memory used related to the CAP is shown.

The next block shows the same statistics for virtual memory. The CAP is at 3,072 MB for this zone, which was specified earlier in the capped-memory resource. You also can see the complete amount of virtual memory for the system (physical memory + swap space) in the column SYSTEM MEMORY.

The last block shows all the statistics for locked memory. In this case no cap is defined, so no cap is shown for the zone.

From the global zone

As you see, the report shows all zones (including Oracle Solaris 10 zones) running on the system. Monitoring Oracle Solaris 10 zones is discussed in the next chapter.

```
-bash-4.1$ zonestat -r memory 60 1
Collecting data for first interval...
Interval: 1, Duration: 0:01:00
PHYSICAL-MEMORY              SYSTEM MEMORY
mem_default                         15.7G
                                ZONE  USED %USED   CAP  %CAP
                             [total] 4059M 25.1%     -     -
                            [system] 2570M 15.9%     -     -
                              global 1149M 7.12%     -     -
                              myzone  193M 1.20% 2048M 9.45%
                            solmarc10  146M 0.90% 1024M 14.2%

VIRTUAL-MEMORY               SYSTEM MEMORY
vm_default                          23.7G
                                ZONE  USED %USED   CAP  %CAP
                             [total] 5933M 24.3%     -     -
                            [system] 2911M 11.9%     -     -
                              global 2710M 11.1%     -     -
                              myzone  169M 0.69% 3072M 5.53%
                            solmarc10  141M 0.58% 1536M 9.23%

LOCKED-MEMORY                SYSTEM MEMORY
mem_default                         15.7G
                                ZONE  USED %USED   CAP  %CAP
                             [total] 2727M 16.9%     -     -
                            [system] 2719M 16.8%     -     -
                              global 7408K 0.04%     -     -
                              myzone     0 0.00%     -     -
                            solmarc10     0 0.00%     -     -
```

The memory report has the same structure as the one in the nonglobal zone. Three blocks for the memory types are displayed.

For each memory type there is a line for each zone and the global zone describing the memory usage, the memory cap, and the usage of the memory cap (if any). Within the global zone, the line [system] means memory used by the kernel. [total] is the sum of [system], the global zone, and the nonglobal zones.

Monitoring Oracle Solaris 10 Zones

In the previous chapter, the `zonestat` command was introduced. This command is useful for monitoring the resources used by zones. It can be used in the global zone to monitor the resource usage and limits of all zones on the system, and it can be used from within default zones to monitor the resource usage and limits of the current zone. As this tool is only available with Oracle Solaris 11, it is not present in Oracle Solaris 10 zones. Nevertheless, the `zonestat` command still can be used within the global zone to monitor Oracle Solaris 10 zones. It is possible to monitor resource usage and limits from within the Oracle Solaris 10 zone without the `zonestat` command

. Other tools can be used for this purpose, and they are discussed in this chapter. These tools are discussed only for usage from within the zone as the `zonestat` command can be used in the global zone.

Monitoring Oracle Solaris 10 Zones with Dedicated CPU

As previously explained, Oracle Solaris automatically creates a resource pool with a processor set for the zone when the dedicated-cpu resource is used. It is possible to use the `poolstat` command to monitor an Oracle Solaris 10 zone with dedicated CPUs[10]. This zone has one to two dedicated CPUs, and a process is running that is utilizing 50 percent of one CPU. To monitor the processor set it is necessary to invoke the `poolstat` command with the option (`-r pset`). It prints two reports in a 60-second interval.

```
bash-3.2$ poolstat -r pset 60 2
 id pool                 type rid rset                 min  max size used load
  2 SUNWtmp_solmarc10    pset   1 SUNWtmp_solmarc10       1    2    1 0.00 0.49

 id pool                 type rid rset                 min  max size used load
  2 SUNWtmp_solmarc10    pset   1 SUNWtmp_solmarc10       1    2    1 0.50 0.50
```

The second report is created after the `poolstat` command runs for 60 seconds. The output contains the name of the pool and processor set, which is assigned to the zone. You also can see the columns *min* and *max*, which show the minimum and maximum size of the pool. The column size shows how many CPUs are assigned currently to the processor set. The column *used* shows the utilization of the processor set.

Now an additional process is started that causes almost 100 percent CPU utilization. Together with the already-running process, you can expect 150 percent CPU utilization (related to one CPU).

```
bash-3.2$ poolstat -r pset 60 2
 id pool                 type rid rset                 min  max size used load
  2 SUNWtmp_solmarc10    pset   1 SUNWtmp_solmarc10       1    2    2 0.00 1.38
```

10 Oracle Corporation, "Using poolstat to Monitor the Pools Facility and Resource Utilization," July 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E37631/rmpool-107.html.

```
 id pool                 type rid rset                  min  max size used load
  2 SUNWtmp_solmarc10    pset   1 SUNWtmp_solmarc10       1    2    2 1.50 1.45
```

You can see in the column size that an additional CPU is assigned to the processor set. The column *load* displays that the usage is 1.5 (which is 150 percent CPU utilization). The load of the processor set also increased.

Further information about the `poolstat` command can be found on its man page[11] or at[12]. The library libpool contains an API that can be used to gather the information displayed via the `poolstat` command. For further information, read the libpool man page[13].

Monitoring Oracle Solaris 10 Zones with Capped CPU

No information about CPU cap is displayed by the `poolstat` command, so you have to use a different tool to monitor an Oracle Solaris 10 zone with capped CPU. In this example the zone has a CPU cap of 1.5, and a process is running that causes 50 percent CPU utilization.

Statistics about CPU caps can be found by using the `kstat` command. From within a zone only the statistics of the current zone are displayed. The statistics about the CPU cap (as defined in the `zonecfg` command) are stored in the statistic with name `cpucaps_zone_<zone ID>`. You can either find out the zone ID of your zone or just use a wildcard when calling the `kstat` command. If no CPU cap is set in the zone this statistic is not available. In the following output the wildcard "?" is used to get the CPU caps statistic for the current zone. Two reports are displayed with an interval of 60 seconds.

```
bash-3.2$ kstat -n 'cpucaps_zone_?' 60 2
module: caps                            instance: 4
name:   cpucaps_zone_4                  class:    zone_caps
        above_sec                       0
        below_sec                       1177
        crtime                          10249.142302345
        maxusage                        151
        nwait                           0
        snaptime                        11426.764310172
        usage                           52
        value                           150
        zonename                        solmarc10


module: caps                            instance: 4
name:   cpucaps_zone_4                  class:    zone_caps
        above_sec                       0
        below_sec                       1237
        crtime                          10249.142302345
        maxusage                        151
```

11 Oracle Corporation, "poolstat man page," July 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36871/poolstat-1m.html.

12 Oracle Corporation, "Resource Pools Used in Zones," July 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E37631/rmpool-114.html.

13 Oracle Corporation, "libpool man page," July 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36873/libpool-3lib.html.

```
        nwait                           0
        snaptime                        11486.772663912
        usage                           50
        value                           150
        zonename                        solmarc10
```

The two important statistics here are *usage* and *value*. The statistic *value* shows the value of the CPU cap multiplied by 100. As this zone is defined with a CPU cap of 1.5, the statistic value shows 150. The statistic *usage* shows 50, which is exactly the CPU utilization that is expected.

Now the process that causes 50 percent CPU utilization is stopped. Two new processes are started, which each cause 100 percent CPU utilization. Then you can use the `kstat` command again to monitor if the cap is enforced.

```
bash-3.2$ kstat -n 'cpucaps_zone_?' 60 2
module: caps                            instance: 4
name:   cpucaps_zone_4                  class:    zone_caps
        above_sec                       56
        below_sec                       3938
        crtime                          10249.142302345
        maxusage                        152
        nwait                           0
        snaptime                        14242.963750635
        usage                           150
        value                           150
        zonename                        solmarc10


module: caps                            instance: 4
name:   cpucaps_zone_4                  class:    zone_caps
        above_sec                       73
        below_sec                       3981
        crtime                          10249.142302345
        maxusage                        152
        nwait                           0
        snaptime                        14302.972309689
        usage                           150
        value                           150
        zonename                        solmarc10
```

As you see the cap of 1.5 is enforced, because the statistic value is 150.

If you want to know how many CPUs are available for the zone, you can use the `poolstat` command for this. Remember that by default the zone uses CPUs from the default processor set, so you can monitor the usage of this processor set with the command `poolstat`.

```
bash-3.2$ poolstat -r pset 60 2
 id pool                 type rid rset                 min  max size used load
```

```
  0 pool_default         pset  -1 pset_default            1  66K    4 0.00 2.68

 id pool                 type rid rset                 min  max size used load
  0 pool_default         pset  -1 pset_default            1  66K    4 2.54 2.67
```

In this case you already know from the `kstat` command output that the zone uses 1.5 CPUs (as limited by the cap). The processor set that the zone is using is utilizing 2.54 CPUs. The output of the `poolstat` command shows that four CPUs are assigned to the zone, and you can calculate that approximately 1.5 CPUs are still available.

Further information about the command `kstat` can be found on its man page[14]. The library libkstat contains an API to gather the information displayed via the command `kstat`. For further information, read the man page of libkstat.[15]

Monitoring Oracle Solaris 10 Zones with Capped Memory

To monitor the memory usage from within Oracle Solaris 10 zones, you can use a combination of the `rcapstat`, `prstat`, and `kstat` commands.

There are some statistics in the `kstat` command for zone caps that have the class **zone_caps**. All these statistics can be seen with the command `kstat -c zone_caps`. For observation of the usage of the capped-memory resource, there are only two statistics available. The **lockedmem_zone_<zone id>** statistic displays usage and cap of locked memory in the zone, and the **swapresv_zone_<zone id>** statistic displays usage and cap of virtual memory in the zone.

In this case the `kstat` command is invoked to display the lockedmem_zone and swapresv_zone statistic twice within an interval of 60 seconds.

```
bash-3.2$ kstat -n '/lockedmem_zone|swapresv_zone/' 60 2
module: caps                            instance: 4
name:   lockedmem_zone_4                class:    zone_caps
        crtime                          98343.245904062
        snaptime                        100820.223739768
        usage                           0
        value                           18446744073709551615
        zonename                        solmarc10

module: caps                            instance: 4
name:   swapresv_zone_4                 class:    zone_caps
        crtime                          98343.245911588
        snaptime                        100820.224596904
        usage                           152866816
        value                           1610612736
        zonename                        solmarc10


module: caps                            instance: 4
```

14 Oracle Corporation, "kstat man page," July 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36871/kstat-1m.html).

15 Oracle Corporation, "libkstat man page," July 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36873/libkstat-3lib.html.

```
name:   lockedmem_zone_4                class:    zone_caps
       crtime                           98343.245904062
       snaptime                         100880.234017332
       usage                            0
       value                            18446744073709551615
       zonename                         solmarc10

module: caps                             instance: 4
name:   swapresv_zone_4                 class:    zone_caps
       crtime                           98343.245911588
       snaptime                         100880.234037401
       usage                            153268224
       value                            1610612736
       zonename                         solmarc10
```

The last statistic, **swapresv_zone_4**, is for the virtual memory in the zone. The statistic *value* is the actual value of the swap cap in byte, set in the capped-memory resource. The statistic *usage* is the current usage of the virtual memory by this zone in byte. The statistic above, lockedmem_zone_4, is for the locked memory in the zone. As no cap is set for locked memory, the line value is set to 18446744073709551615, which is 2^64 -1 (the theoretical limit of physical memory on a 64-bit system). The statistic *usage* shows that 0 byte is currently used as locked memory by the zone.

Unfortunately, the `kstat` command doesn't have any statistics about the physical memory cap. Some information on the physical memory cap can be found via the command `rcapstat`. To use the `rcapstat` command, the service (`system/rcap`) has to be enabled in the zone.

```
bash-3.2$ rcapstat -z 1 1

    id zone             nproc    vm   rss   cap    at avgat    pg avgpg
    4  solmarc10            -    0K    0K 1024M    0K    0K    0K    0K
```

The output of the command `rcapstat` shows that the physical memory cap is set to 1 GB. This is set in the capped-memory resource in the `zonecfg` command. In the Oracle Solaris 10 zone, the command `rcapstat` only displays the cap but not the usage of the cap. To monitor the usage of the physical memory cap you can use the `prstat` command with the option `-Z`. The option `-n 1` is used to reduce the number of processes that are shown. Only the memory usage statistic is important here.

```
bash-3.2$ prstat -n 1 -Z 60 2
   PID USERNAME  SIZE   RSS STATE  PRI NICE      TIME  CPU PROCESS/NLWP
  1585 daemon   9008K 3200K sleep   59    0   0:00:11 0.0% rcapd/1
ZONEID    NPROC  SWAP   RSS MEMORY      TIME  CPU ZONE
     4       38  145M  158M    15%   0:00:32 0.0% solmarc10
Total: 38 processes, 145 lwps, load averages: 0.18, 0.86, 1.31
```

The memory usage statistic can be found on the second to last line of the output. The value for swap (which you also can find via the `kstat` command) and the value for the resident-set size of all processes running in the zone are visible in this line. Additionally, the percentage of memory used by the zone is shown. This is related to the memory cap set for the zone.

Further information about the `prstat` and `rcapstat` commands can be found on their man pages[16] and[17].

Monitoring Kernel Zones

Kernel Zones is the latest zone type introduced with Oracle Solaris 11.2. While the default zones share their kernel with the global zone, each instance of Kernel Zones has its own kernel. This means that a kernel zone can have a different Oracle Solaris release than the global zone. Another difference is that it is possible to install default zones or Oracle Solaris 10 zones inside a kernel zone. Such zones are called nested zones.

To retrieve the name of the kernel zone, you have to use the `virtinfo` command. The `zonename` command only returns *global* in a kernel zone.

```
root@scs-lh:~# zonename
global
root@scs-lh:~# virtinfo -c current get zonename
NAME        CLASS    PROPERTY VALUE
kernel-zone current zonename kzone1
```

The `virtinfo` command displays the zone name of the kernel zone as it is defined in the global zone. You can check with the `zoneadm` command in the global zone to see the name of the kernel zone, which is kzone1 in this example.

```
root@t44:~# zoneadm list -v
  ID NAME             STATUS      PATH                           BRAND      IP
   0 global           running     /                              solaris    shared
   2 kzone1           running     -                              solaris-kz excl
```

In this example, the global zone is configured to use a dedicated-cpu resource with 100 CPUs. This means that a processor set with 100 CPUs is created and is assigned to the resource pool that is exclusively assigned to the kernel zone. The capped-memory resource has a different meaning with Kernel Zones than with default zones or Oracle Solaris 10 zones. In this case the physical memory specified in the capped-memory resource is statically assigned to the kernel zone. This memory is not available for the global zone anymore.

```
root@t44:~# zonecfg -z kzone1
zonecfg:kzone1> info dedicated-cpu
dedicated-cpu:
        ncpus: 100
        cpus not specified
        cores not specified
        sockets not specified
zonecfg:kzone1> info capped-memory
capped-memory:
        physical: 64G
zonecfg:kzone1>
```

16 Oracle Corporation, "prstat man page," July 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36871/prstat-1m.html.

17 Oracle Corporation, "rcapstat man page," July 2014. [Online]. Available: http://docs.oracle.com/cd/E36784_01/html/E36870/rcapstat-1.html.

Monitoring Kernel Zones with Capped Memory

Kernel Zones also can be monitored with the `zonestat` command.

From the kernel zone

In contrast to default zones, the `zonestat` command only reports the statistics of the kernel zone. Statistics of the complete server are not visible here. You can see that the 64 GB defined in the `zonecfg` command are visible as physical memory of the kernel zone. If further zones are created within the kernel zone, the statistics about these zones are visible here.

```
root@scs-lh:~# zonestat -r memory 60 1

Collecting data for first interval...
Interval: 1, Duration: 0:01:00
PHYSICAL-MEMORY              SYSTEM MEMORY
mem_default                         64.0G
                                ZONE  USED %USED   CAP  %CAP
                             [total] 3375M 5.15%     -     -
                            [system] 3076M 4.69%     -     -
                              global  298M 0.45%     -     -

VIRTUAL-MEMORY               SYSTEM MEMORY
vm_default                          64.0G
                                ZONE  USED %USED   CAP  %CAP
                             [total] 4789M 7.30%     -     -
                            [system] 4233M 6.46%     -     -
                              global  555M 0.84%     -     -

LOCKED-MEMORY                SYSTEM MEMORY
mem_default                         64.0G
                                ZONE  USED %USED   CAP  %CAP
                             [total] 4064M 6.20%     -     -
                            [system] 4064M 6.20%     -     -
                              global     0 0.00%     -     -
```

From the global zone

Using the `zonestat` command from the global zone shows you the statistics about all zones on the system including the kernel zone. The memory statistics for the kernel zone show that it has a cap of 64 GB and all of it is used. As you know from the zonestat report within the kernel zone, the kernel zone only uses about 3.3 GB of memory. This means that the `zonestat` command in the global zone doesn't have detailed statistics about the memory usage of the kernel zone.

```
root@t44:~# zonestat -r memory 60 1
Collecting data for first interval...
Interval: 1, Duration: 0:01:00
PHYSICAL-MEMORY              SYSTEM MEMORY
```

```
mem_default                             512G
                                  ZONE  USED %USED   CAP  %CAP
                                [total] 81.2G 15.8%     -     -
                               [system] 16.5G 3.22%     -     -
                                 kzone1 64.0G 12.5% 64.0G  100%
                                 global  692M 0.13%     -     -

VIRTUAL-MEMORY                 SYSTEM MEMORY
vm_default                              512G
                                  ZONE  USED %USED   CAP  %CAP
                                [total] 89.5G 17.4%     -     -
                               [system] 88.4G 17.2%     -     -
                                 global  965M 0.18%     -     -
                                 kzone1  192M 0.03%     -     -

LOCKED-MEMORY                  SYSTEM MEMORY
mem_default                             512G
                                  ZONE  USED %USED   CAP  %CAP
                                [total] 25.1G 4.90%     -     -
                               [system] 25.1G 4.90%     -     -
                                 global     0 0.00%     -     -
                                 kzone1     0 0.00%     -     -
```

Monitoring Kernel Zones with Dedicated CPU

From the kernel zone

Because the kernel zone uses the dedicated-cpu resource, a processor set is exclusively assigned to the kernel zone. When calling the `zonestat` command within the kernel zone, the statistics of this processor set are shown. If further zones are created inside the kernel zone, the processor sets or the CPU caps of these zones are also displayed here. The output shows that 100 CPUs are part of this processor set and therefore of the kernel zone, and that currently one CPU is fully utilized.

```
root@scs-lh:~# zonestat -r processor-set 60 1
Collecting data for first interval...
Interval: 1, Duration: 0:01:00
PROCESSOR_SET                   TYPE  ONLINE/CPUS     MIN/MAX
pset_default           default-pset      100/100         1/-
                                  ZONE  USED %USED   CAP  %CAP   SHRS  %SHR %SHRU
                                [total]  1.01 1.01%     -     -      -     -     -
                               [system]  0.01 0.01%     -     -      -     -     -
                                 global  1.00 1.00%     -     -      -     -     -
```

From the global zone

In the global zone, the command `zonestat` displays statistics about all processor sets on the server. Also, the statistics of the processor sets used by Kernel Zones are displayed. In this case you can see both the default

processor set and the processor set kzone1, which is used by the kernel zone. The USED column for the kzone1 processor set shows that one CPU is fully utilized. The same is also observed from within the kernel zone.

```
root@t44:~# zonestat -r processor-set 60 1
Collecting data for first interval...
Interval: 1, Duration: 0:01:00
PROCESSOR_SET                    TYPE  ONLINE/CPUS     MIN/MAX
pset_default             default-pset        28/28         1/-
                    ZONE  USED %USED  STLN %STLN   CAP  %CAP   SHRS  %SHR %SHRU
                 [total]  0.81 2.89%  0.00 0.00%     -     -      -     -     -
                [system]  0.05 0.19%     -     -     -     -      -     -     -
                  global  0.75 2.70%     -     -     -     -      -     -     -

PROCESSOR_SET                    TYPE  ONLINE/CPUS     MIN/MAX
kzone1                  dedicated-cpu     100/100     100/100
                    ZONE  USED %USED  STLN %STLN   CAP  %CAP   SHRS  %SHR %SHRU
                 [total]  1.06 1.06%  0.00 0.00%     -     -      -     -     -
                [system]  0.02 0.02%     -     -     -     -      -     -     -
                  kzone1  1.04 1.04%  0.00 0.00%     -     -      -     -     -
```

Beginning with Oracle Solaris 11.2 SRU 8, the `zonestat` command also reports stolen time. Stolen time is the CPU time the zone is not able to use, because the server is using the CPU for other purposes. You can see the stolen time in the STLN column. The column *%STLN* is the percentage of stolen time related to the total CPU time available.

Monitoring Nested Zones

From the perspective of a nested zone, the kernel zone is its global zone. This means it is possible to use the commands `zonestat`, `poolstat`, `rcapstat`, and `kstat` to monitor the nested zones from within the nested zones or from the kernel zone.

Please be aware that you cannot monitor nested zones within a kernel zone from the global zone. For example, if the kernel zone kzone1 contains nested zones, you cannot observe or monitor them from the global zone.

Using OpenStack for Monitoring Zones

Beginning with Oracle Solaris 11.2, an OpenStack distribution is delivered and integrated into Oracle Solaris. The OpenStack components are the following:

- Nova is the compute engine and is the component responsible for creating, starting, and stopping virtual machines. It communicates with the Oracle Solaris Zones framework.
- Neutron is the networking component that creates virtual networks by communicating with the Oracle Solaris elastic virtual switches.
- Glance manages images that can be used by Nova to launch virtual machines from it. Glance makes use of Unified Archive, a feature of Oracle Solaris.
- Cinder and Swift are services for accessing storage. Both make use of ZFS in Oracle Solaris.
- Horizon is a dashboard that is accessible via a web browser.
- Keystone manages users and access rights within the OpenStack components.

Each component has its own command-line tools, and the most important functions are accessible via the OpenStack dashboard.

In the Admin tab in the OpenStack dashboard, there are different menu entries. Hypervisors and Instances, for example, are discussed below.

Hypervisors gives you an overview of all hypervisors managed by OpenStack. It shows each host in the cloud and the type of hypervisor.

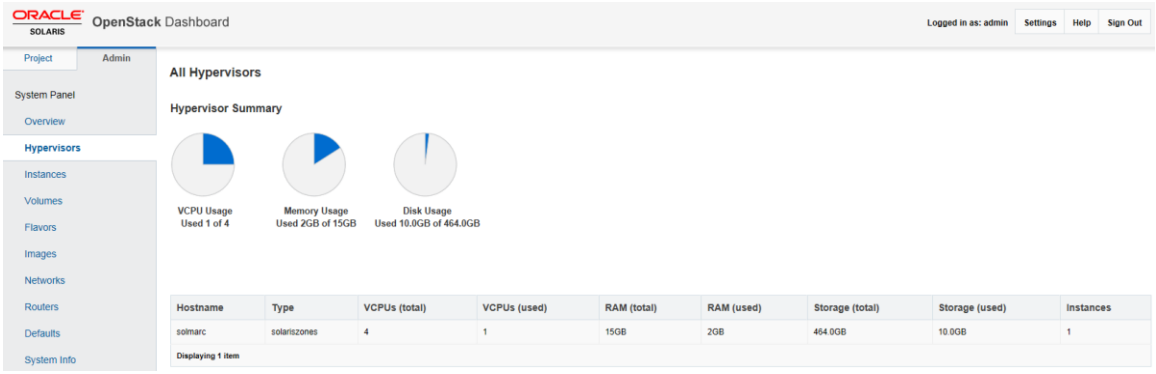

Figure 1. Overview of the managed hypervisors and summary of CPU, memory, and disk usage in the OpenStack dashboard

In this screenshot there is only one host managed by OpenStack, and it uses the Oracle Solaris Zones framework as the hypervisor. You can see statistics of memory, CPUs, and storage per hypervisor and as summary overall hypervisors. It is important to know that the statistics only take zones into account that are created with OpenStack. OpenStack is not aware of zones that are created in other ways (e.g., via `zonecfg` from the command line). In this example the hypervisor overview shows that only one virtual machine (VM) instance exists. If you choose Instances, you can see which instances are created via OpenStack.

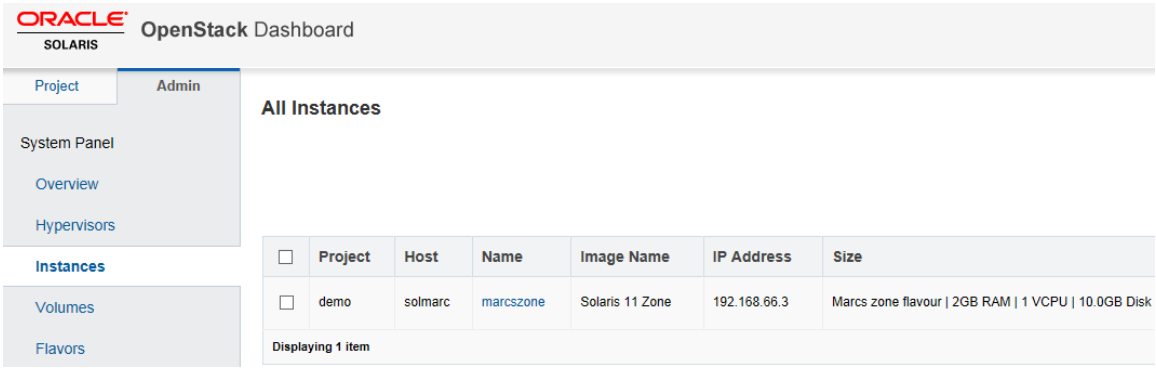

Figure 2. Overview of the instances created with OpenStack. In this case the only instance that is created is the zone **marcszone**.

In this case only one instance (a default zone) is created. It has 2 GB RAM and one CPU. If you execute the command `zoneadm` list on the server, the zone created by OpenStack that is named instance-00000016, and another zone that is created with the `zonecfg` command, are displayed.

```
root@solmarc:/# zoneadm list -v
  ID NAME             STATUS      PATH                          BRAND      IP
   0 global           running     /                             solaris    shared
```

```
   3 instance-00000016 running     /system/zones/instance-00000016 solaris    excl
   4 myzone            running     /system/zones/myzone           solaris    excl
```

Only the zone created by OpenStack is shown in the dashboard. If you examine the settings of the zone, which is created by OpenStack, you see that the capped-cpu resource is used to limit the number of CPUs used by the zone. The capped-memory resource is used, too. Instead of setting a physical cap, OpenStack sets a cap for swap. This limits the usage of physical memory to this amount, as swap means physical memory + swap space.

```
root@solmarc:/# zonecfg -z instance-00000016
zonecfg:instance-00000016> info capped-memory
capped-memory:
       [swap: 2G]
zonecfg:instance-00000016> info capped-cpu
capped-cpu:
       [ncpus: 1.00]
zonecfg:instance-00000016>
```

Back to the hypervisor view in the OpenStack dashboard—it shows that currently 2 GB of memory are used. If you compare this with the output of the command `zonestat` from the global zone, you see that the usage in OpenStack means that this amount of memory could be used by a zone—like in this case where the zone has a 2 GB cap.

```
root@solmarc:/# zonestat -r physical-memory,virtual-memory 60 1
Collecting data for first interval...
Interval: 1, Duration: 0:01:00
PHYSICAL-MEMORY              SYSTEM MEMORY
mem_default                         15.7G
                                ZONE  USED %USED   CAP  %CAP
                             [total] 6255M 38.7%     -     -
                            [system] 3114M 19.3%     -     -
                              global 2767M 17.1%     -     -
                  instance-00000016  187M 1.16%     -     -
                              myzone  185M 1.15% 2048M 9.07%

VIRTUAL-MEMORY               SYSTEM MEMORY
vm_default                          23.7G
                                ZONE  USED %USED   CAP  %CAP
                             [total] 6927M 28.4%     -     -
                            [system] 3413M 14.0%     -     -
                              global 3162M 13.0%     -     -
                  instance-00000016  180M 0.74% 2048M 8.82%
                              myzone  170M 0.70% 3072M 5.56%
```

Memory used by other zones or by the system is not taken into account. You can see that the system currently uses approximately 6 GB of memory and not 2 GB as OpenStack shows. The same is valid for CPU usage.

```
root@solmarc:/# zonestat -r processor-set 60 1
Collecting data for first interval...
```

```
Interval: 1, Duration: 0:01:00
PROCESSOR_SET                    TYPE  ONLINE/CPUS     MIN/MAX
pset_default            default-pset          4/4         1/-
                     ZONE  USED %USED  STLN %STLN   CAP  %CAP   SHRS  %SHR %SHRU
                  [total]  2.13 53.4%  0.00 0.00%     -     -      -     -     -
                 [system]  0.00 0.20%     -     -     -     -      -     -     -
                   global  2.13 53.2%     -     -     -     -      -     -     -
        instance-00000016  0.00 0.00%     -     -  1.00 0.03%      -     -     -
                   myzone  0.00 0.01%     -     -  1.50 0.02%      -     -     -
```

In this case, the OpenStack zone (instance-00000016) has a CPU cap of 1, but the current usage is 0. The overall usage of the system is 2.13, but the OpenStack dashboard shows only the CPU usage is 1.

Therefore, if you use OpenStack on some servers, the zones only should be provided via OpenStack and not manually via the command `zonecfg`. This helps you avoid the problem caused because OpenStack isn't aware of zones not provided by OpenStack. Furthermore, OpenStack shouldn't be used to monitor the exact usage. It is useful for getting an overview of how many CPUs and how much memory the machine has, what the caps are, and if the zones are overprovisioned. For an exact monitoring of zones with exact information about memory usage and CPU utilization, it is better to rely on the command `zonestat` and the other tools discussed in this white paper.

Conclusion

Oracle Solaris offers different methods to control the resource usage of the zones. To control CPU resource usage you can configure your zone with dedicated CPUs, capped CPU, or CPU shares. In case the zone is a kernel zone, you also can use virtual CPUs. To restrict memory usage in zones you can use capped memory.

Monitoring Oracle Solaris Zones works best from the global zone. The `zonestat` command offers many statistics to choose from and is able to monitor the usage of the system's resources by all zone types. Monitoring from within the zone is also possible.

In default zones, the command `zonestat` is the tool of choice, and it is also accessible via the zonestat API in libzonestat (e.g., if you want to integrate zones monitoring in a third-party application running on zones). In default zones it is possible to monitor resource usage of the rest of the system. If the default zone uses dedicated CPUs, only the CPU resource usage within its own processor set can be monitored. In all cases, no details about other zones can be monitored from within the zone. Only a combined value for global and other zones is shown within the zone.

The command `zonestat` is not available on Oracle Solaris 10. In Oracle Solaris 10 zones, other tools have to be used. The commands `poolstat` and `kstat` also have APIs that can be used for zones monitoring in a third-party application.

The command `zonestat` is available in Kernel Zones, but it isn't able to monitor resources used by the global zone and other zones. Usually this isn't a problem because the CPUs and memory are assigned exclusively to a kernel zone and won't be used by other zones or the global zone.

The following table suggests different monitoring tools (commands) that should be used depending on the zone type and resource type to monitor.

OVERVIEW OF SUGGESTED ZONE MONITORING TOOLS

Resource Type	Zone Type	Monitoring Tool Within the Zone	Monitoring Tool from the Global Zone
Dedicated CPU	Default Oracle Solaris Zones (solaris)	`zonestat -r processor-set <interval> <count>`	`zonestat -r processor-set <interval> <count>`
	Oracle Solaris 10 zones (solaris10)	`poolstat -r pset <interval> <count>`	
	Kernel Zones (solaris-kz)	`zonestat -r processor-set <interval> <count>`	
Capped CPU	Default Oracle Solaris Zones (solaris)	`zonestat -r processor-set <interval> <count>`	`zonestat -r processor-set <interval> <count>`
	Oracle Solaris 10 zones (solaris10)	`kstat -n 'cpucaps_zone_?' <interval> <count>` and `poolstat -r pset <interval> <count>`	
Capped memory	Default Oracle Solaris Zones (solaris)	`zonestat -r memory <interval> <count>`	`zonestat -r memory <interval> <count>`
	Oracle Solaris 10 zones (solaris10)	**`kstat -n '/lockedmem_zone|swapresv_zone/' <interval> <count>`** and **`prstat -z <interval> <count>`** and **`rcapstat -z <interval> <count>`**	
	Kernel Zones (solaris-kz)	`zonestat -r memory <interval> <count>`	

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

ORACLE

CONNECT WITH US


blogs.oracle.com/oracle


facebook.com/oracle


twitter.com/oracle


oracle.com

Integrated Cloud Applications & Platform Services

Virtualization Monitoring in Oracle Solaris Zones
August 2015
Author: Marc Nesello


Oracle is committed to developing practices and products that help protect the environment