JomaSoft GmbH
Falkensteinstr. 54a
9000 St. Gallen
Switzerland

VDCF - Virtual Datacenter Cloud Framework

for the Solaris™ Operating System

Resource Management

Version 3.3
31. March 2016

JomaSoft GmbH
Falkensteinstr. 54a
9000 St. Gallen
Switzerland

Contents

JomaSoft GmbH
Falkensteinstr. 54a
9000 St. Gallen
Switzerland

1 Introduction

This documentation describes the Resource Management (RM) feature of the Virtual Datacenter Cloud Framework (VDCF) for the Solaris Operating System and explains how to use this feature.

See these documents for more information about the related VDCF products:

VDCF – Administration Guide	for information about VDCF Usage
VDCF – Monitoring	for information about VDCF Monitoring usage

1.1 Overview

VDCF Resource Management is a VDCF Enterprise extension available to VDCF Enterprise and Standard customers.

VDCF Resource Management is based on Solaris 10 and 11 Resource Manager. It delivers central control of resources that can be assigned to a particular Zone. Extended Zone Resource Management has been integrated into Solaris 10 Update 4 (8/07). With Solaris 10 Update 5 (5/08) the Resource Manager got a new resource control 'cpu-cap'.

The basic functionality of VDCF Resource Management relies on this Solaris feature set.

While VDCF Resource Management is used to configure resource limits, VDCF Monitoring offers features to collect and display the resource usage.

1.1.1 CPU Resource Management

Solaris Resource Manager delivers two distinct technologies to control the amount of CPU being used by a Zone or as we call it in VDCF terminology: vServer. One of these technologies are Dynamic Resource Pools. Such a pool can be configured to contain a number of CPU's as seen by the Solaris scheduler. A pool can be configured with a static number or a range of CPU's. If a range has been specified, the number of CPU's can and will vary based on the defined importance for that pool. As of Solaris 10 8/07 the pool can be configured using the Zone configuration. As soon as a Zone boot up, the pool gets created automatically. The Zone can only boot up if sufficient CPU's are available. Note that a CPU can not be shared between different pools. The granularity for the assignment of CPU's to vServer therefor is 1 CPU.

The other way to control the amount of CPU being used by a Zone is the Fair Share Scheduler (FSS). It allows for fine granular assignment of CPU resources in terms of shares. FSS allows for sub CPU granularity.

To fully support the approach of utility based computing, we need to base the CPU Resource Management on a performance metric. This allows an administrator to set up a vServer with a particular performance in terms of CPU power. The metric used to specify the power of a particular vServer, should be generic and automatically translated into the number of CPU shares. This would allow the migration of vServer from one node to another, even if this new node has more or less total CPU power available. A migration of a vServer requires an adaption of its assigned shares so, that its available CPU power remains the same.

VDCF Resource Management supports both technologies. However, true utility based computing can only be achieved using the FSS based mechanism.

JomaSoft GmbH
Falkensteinstr. 54a
9000 St. Gallen
Switzerland

1.1.2 Memory Resource Management

Memory can be controlled on different levels using the Solaris Resource Manager. As of Solaris 10 8/07 it is possible to control the amount of physical memory (RAM), the amount of virtual memory (SWAP), the amount of shared memory and the amount of locked down memory allowed for a particular vServer.

VDCF Resource Management support all mechanisms for controlling resources related to a Solaris Zone. More information about the Solaris 10 Resource Manager and its abilities can be found on http://download.oracle.com/docs/cd/E19044-01/sol.containers/817-1592

1.2 Requirements / Patches

The VDCF Resource Management implementation is based on Solaris 10 8/07 (Update 4) features. To use this feature the target Nodes must run Solaris 10 8/07 or later. It is supported to use an older Solaris 10 Release (Update1,2,3) with Kernel Patch 120011-14 (sparc) or 120012-14 (i386) or later.

JomaSoft GmbH
Falkensteinstr. 54a
9000 St. Gallen
Switzerland

2 Installation and Configuration

2.1 Prerequisites

The JSvdcf-rm package requires the following VDCF packages to be installed on the VDCF Management Server:

- JSvdcf-base 5.7.0 or later

2.2 Installation

a) sparc platform

```
cd <download-dir>
pkgadd -d ./JSvdcf-rm_<version>_sparc.pkg
```

b) i386 platform

```
cd <download-dir>
pkgadd -d ./JSvdcf-rm_<version>_i386.pkg
```

JomaSoft GmbH
Falkensteinstr. 54a
9000 St. Gallen
Switzerland

2.3 Configuration

2.3.1 Granting User Access

The VDCF Resource Management package introduces the RBAC Profile “VDCF resource Module”. Assign this RBAC profile to your admin users.

2.3.2 Base Configuration

VDCF Resource Management (RM) uses a database table to define the relative performance of a specific CPU type. This performance metric will be used to define a servers relative CPU power. To list all defined CPU's and there performance metrics use the following command:

```
-bash-4.1$ rcadm -c show_perf cpu

         Performance Base Units set to: 100

           CPU-Model    CPU-Freq      Perf-Value    Base-Units
          SPARC64-X+    3700          2229          22.29
          SPARC64-X+    3400          2011          20.11
          SPARC64-X+    3200          1910          19.1
           SPARC64-X    2800          1576          15.76
     UltraSPARC-IV+     2100          617           6.17
            SPARC-M7    4133          612           6.12
          SPARC64-V     1650          606           6.06
     UltraSPARC-IV+     1800          570           5.7
     UltraSPARC-IV+     1950          554           5.54
     UltraSPARC-IV+     1500          502           5.02
                x86     1667          500           5
       SPARC64-VII+     3000          484           4.84
            SPARC-T5    3600          477           4.77
       SPARC64-VII+     2860          470           4.7
            SPARC-T4    3000          422           4.22
        SPARC64-VII     2880          413           4.13
        SPARC64-VII     2750          411           4.11
            SPARC-T4    2848          401           4.01
    UltraSPARC-IIIi     1600          389           3.89
    UltraSPARC-IIIi     1593          387           3.87
    UltraSPARC-IIIi     1504          378           3.78
    UltraSPARC-IIIi     1500          377           3.77
    UltraSPARC-IIIi     1503          377           3.77
       SPARC64-VII+     2660          368           3.68
         SPARC64-VI     2400          368           3.68
         SPARC64-VI     2280          360           3.6
    UltraSPARC-III+     1350          360           3.6
         SPARC64-VI     2150          352           3.52
    UltraSPARC-IIIi     1336          342           3.42
      UltraSPARC-IV     1350          337           3.37
    UltraSPARC-IIIi     1280          329           3.29
    UltraSPARC-III+     1200          323           3.23
      UltraSPARC-IV     1200          315           3.15
        SPARC64-VII     2530          314           3.14
        SPARC64-VII     2520          313           3.13
    UltraSPARC-III+     1050          301           3.01
      UltraSPARC-IV     1050          293           2.93
    UltraSPARC-III+     900           269           2.69
    UltraSPARC-IIIi     1002          269           2.69
        SPARC64-VII     2400          257           2.57
     UltraSPARC-IIe     648           229           2.29
     UltraSPARC-IIe     650           229           2.29
```

JomaSoft GmbH
Falkensteinstr. 54a
9000 St. Gallen
Switzerland

UltraSPARC-III+	750	228	2.28
UltraSPARC-IIe	548	193	1.93
UltraSPARC-IIe	500	176	1.76
UltraSPARC-IIe	502	176	1.76
UltraSPARC-II	480	171	1.71
UltraSPARC-II	450	165	1.65
UltraSPARC-T2	1582	164	1.64
UltraSPARC-T2+	1582	156	1.56
UltraSPARC-T2+	1596	153	1.53
SPARC64-IV	603	150	1.5
UltraSPARC-T2	1415	142	1.42
UltraSPARC-T2	1417	142	1.42
SPARC-T3	1649	141	1.41
UltraSPARC-II	400	136	1.36
UltraSPARC-T1	1400	135	1.35
UltraSPARC-T2+	1415	128	1.28
UltraSPARC-T2	1165	124	1.24
UltraSPARC-T2	1167	124	1.24
UltraSPARC-T1	1200	118	1.18
UltraSPARC-T2+	1162	112	1.12
UltraSPARC-T2+	1165	112	1.12
UltraSPARC-II	296	111	1.11
UltraSPARC-T1	1000	100	1

This command lists all the known SPARC CPU types available on the market and the frequencies. Assigned to each CPU there is a performance metric that indicates the relative performance of that particular CPU (third column). The fourth column indicates the performance of a CPU type in so called Base Unit values. Base Units are used to define a vServers CPU power.

As indicated in the first line of the output, the Base Units are set to 100 in this example. This default value can be modified by adding another value in `conf/customize.cfg`.

```
#
# PERFORMANCE METRIC BASE SETTING
export PERF_BASE_UNIT="10"
```

It allows a customer to set its preferred CPU type to be represented as 1 Base Unit. In this example, an `UltraSPARC-T1` CPU (as represented as a dispatchable unit in Solaris) is defined as 1 Base Unit. To define the relative CPU power of a particular system, this value will be multiplied by the number of CPU's visible by the OS. Discovering a new physical server using the `nodecfg` command will automatically set up the appropriate metrics for it. To list the defined values use:

```
# rcadm -c show_perf node

   Name ... CPU-Model           CPU-Freq  CPUs   Mem MB    Base-Units
  s0003 ... UltraSPARC-IIIi     1002      2      2048      11.1
  s0004 ... UltraSPARC-IIIi     1280      2      2048      14.08
  s0005 ... UltraSPARC-T1       1000      24     8184      24
```

To stick with the above example, the output for the `UltraSPARC-T1` shows a total computing power of 24 Base Units. These Base Units can now be allocated to vServers. It guarantees the requested CPU power in case multiple vServers are contending for CPU resources on that system. However, it is not possible to cap CPU consumption on that defined value. This is a Solaris 10 Resource Manager rather then a VDCF RM restriction. The implementation of CPU capping will be available in a future Solaris 10 Update. As soon as this feature will be made available, VDCF RM will take advantage of it.

2.3.1 Resource Rules

New since VDCF Resource Management 3.1

To avoid assigning too much resources to vServers, VDCF Resource Management allows to define Minimum Memory and CPU required for the Node.

Default values:

```
# RESOURCE RULES
# - Minimum RAM required/reserved for NODE in %
export RESOURCE_NODE_RAM_MIN=10
# - Minimum CPU required/reserved for NODE in %
export RESOURCE_NODE_CPU_MIN=0
```

Defining too much Memory to a vServer is rejected by the rcadm set operation. This check may be disabled by using the '`force`' flag.

```
$ rcadm -c set vserver=s0152 ram=1024
setting rctl properties for vServer s0152
ERROR: Not enough Minimum Memory (192 MB / 10%) for Node s0006, Available 704 MB
ERROR: failed to set rctl properties
```

If you define too much CPUs to a vServer the rcadm set operation will warn you.

```
$ rcadm -c set cpu_shares=4 vserver=s1052
setting rctl properties for vServer s1052
WARN: Not enough free CPU_Shares (0 / 10%) for Node s0006, Available 1
setting rctl properties successful
```

JomaSoft GmbH
Falkensteinstr. 54a
9000 St. Gallen
Switzerland

3 Usage

The whole concept of VDCF RM is based on the idea that vServers are used to consolidate applications. This consolidation uses a pool of compute resources where a server is called a node. vServers are hosted on that pool with great flexibility. A vServer can be migrated from one physical node to another. VDCF takes care of all the required operations. VDCF RM can be used to ensure that a particular vServer gets enough resources while it is running in parallel with other vServers on the same node.

VDCF RM uses the `rcadm` command to control resource assignments centrally for all vServers managed by VDCF.

3.1 Available Resource Controls

`rcadm` manages Solaris Resource Manager Resource Controls as properties. CPU Resources are either controlled by Fair Share Scheduler (FSS) or by Dynamic Resource Pool (DRP) feature. For information about this two technologies see Chapter 1.1. Overview. The two technologies are mutually exclusive. That is, one either specifies CPU resources by FSS or DRP.

The following properties are available to activate FSS based CPU control

		Solaris Zone Property
'CPU_Shares'	Number of Base Units. If CPU_Shares are defined 'CPUs' and 'Importance' are not allowed and FSS based RM is activated	cpu-shares
'CPU_cap'	CPU capping in Base Units. 'CPU_Shares' and 'CPU_cap' can be but must not be specified together. They indicate a guaranteed and a maximum CPU entitlement.	cpu-cap

If DRP should be activated the following properties needs to be set:

'CPUs'	Number of CPU's for vServer. Can be a range.	dedicated-cpu
'Importance'	Relative importance of vServer.	importance

VDCF RM also supports the management of all other supported zone resource controls. These are:

'RAM'	Physical RAM in K,M,G,T	capped-memory
'SWAP'	Virtual Memory in K,M,G,T	max-swap
'Locked'	Maximum locked down Memory in K,M,G,T	max-locked-memory
'LWP'	Maximum number of LWPs	max-lwps
'MSG_ids'	Maximum number of Message Queues	max-msg-ids
'SEM_ids'	Maximum number of Semaphores	max-sem-ids
'SHM_ids'	Maximum number of Shared Memory Segments	max-shm-ids
'SHM_Size'	Maximum size of all Shared Memory Segments in K,M,G,T	max-shm-memory

Sizes are specified as n, n[bB], n[kK], n[mM], n[gG], n[tT] where n is megabytes. The minimum values are: 1048576b, 1024k, 1 or 1m, 1g, 1t.

Properties are not case sensitive!

For more information about the `rcadm` command use `rcadm -H`.

JomaSoft GmbH
Falkensteinstr. 54a
9000 St. Gallen
Switzerland

3.2 Basic vServer Operation

Properties are either set or unset for a particular vServer. Additionally, each property, set or unset, is either current or uncommitted. Current means that they are currently activated for the vServer. Uncommitted properties are defined but not activated for a vServer. However, the next `commit` operation will activate those settings. Uncommitted settings might be abandoned using the `revert` operation.

To check property settings and state for a vServer called s1022 use the following command:

```
# rcadm -c show vserver=s1022

       Resource Management settings for: s1022

  Property Name   Current Setting       Uncommitted Setting   Units
     CPU_Shares:  6                     -                     BaseUnits
           CPUs:  -                     -                     CPUs
     Importance:  -                     -                     Weight
            RAM:  1024                  -                     MB
           SWAP:  2048                  -                     MB
         Locked:  512                   -                     MB
            LWP:  -                     -                     Units
        MSG_ids:  -                     -                     Units
        SEM_ids:  10                    -                     Units
        SHM_ids:  10                    -                     Units
       SHM_Size:  800                   -                     MB
```

Various resource controls have been set and activated for that vServer. There are currently no uncommitted settings for it. To change a current, activate a new or abandon an existing setting use the following commands:

```
# rcadm -c set   vserver=s1022 lwp=1000 cpu_shares=4
# rcadm -c unset vserver=s1022 sem_ids
```

Check your changes with:

```
# rcadm -c show vserver=s1022

       Resource Management settings for: s1022

  Property Name   Current Setting       Uncommitted Setting   Units
     CPU_Shares:  6                     4                     BaseUnits
           CPUs:  -                     -                     CPUs
     Importance:  -                     -                     Weight
            RAM:  1024                  1024                  MB
           SWAP:  2048                  2048                  MB
         Locked:  512                   512                   MB
            LWP:  -                     1000                  Units
        MSG_ids:  -                     -                     Units
        SEM_ids:  10                    unset                 Units
        SHM_ids:  10                    10                    Units
       SHM_Size:  800                   800                   MB
```

The requested changes are now stored in the database but no changes have been made on the actual running configuration for the vServer. To activate those changes, submit the following command:

```
# rcadm -c commit vserver=s1022
```

If one prefers not to implement the changes, the revert operation can be used:

```
# rcadm -c revert vserver=s1022
```

JomaSoft GmbH
Falkensteinstr. 54a
9000 St. Gallen
Switzerland

3.3 Observing Node Settings

Once vServers have been configured to be resource controlled, it might be interesting to see what the impact on the most constrained resources are. To generate an overview list of all resource managed vServer use the following command:

```
# rcadm -c show

vServer  RCTL     ---- Base  Units ----       Memory   Node   vServer    vServer
Name     State    Total      CPU Shares  CPUs Size MB  Name   CPU Usage  Mem Usage
s1022    ACTIVE   11.1        6          -     1024    s0003  54.05%     50.00%
s1023    ACTIVE   11.1        3          -      512    s0003  27.02%     25.00%
s1024    ACTIVE   20.0       16          -     4096    s0005  80.00%     75.00%
```

To see what a particular Node has in terms of resource management settings and what its current occupation is, issue the following command:

```
# rcadm -c show name=s0003
```

The `name` argument takes either a Node or a vServer name. The output triggered by the `name` argument, always lists the respective Node and its assigned vServers.

```
   Node  --- Base Units ---      Total      Total      Total
   Name  Total       Used       Memory MB  CPU Usage  Mem Usage
  s0003  11.1        9          2048        81.07%     75.00%

vServer  RCTL       Base Units  Used       Memory     vServer    vServer
   Name  State      CPU Shares  CPUs       Size MB    CPU Usage  Mem Usage
  s1022  ACTIVE     6           -          1024        54.05%     50.00%
  s1023  ACTIVE     3           -           512        27.02%     25.00%
```

You may set the VDCF configuration variable: VIRTUAL_RCADM_SHOW_IGNORE_CSTATES if not all vServers should be displayed in the show command output, for example detached vServers.
To list all vServer use the additional 'all-states' flag.

JomaSoft GmbH
Falkensteinstr. 54a
9000 St. Gallen
Switzerland

3.4 Converting CPU Pools

New since VDCF Resource Management 3.2

VDCF uses the zonecfg resource controls to define dynamic CPU pools, when the rcadm `CPUs` and `Importance` attributes are used. If existing vServers are imported into VDCF, there may exist static CPU pools defined using the Solaris poolcfg command.

When importing a vServer into VDCF, such CPU pools are listed.

```
-bash-4.1$ vserver -c import name=v0140 node=g0050

vServer v0140 (imported vServer) is created.
Convertable CPU Pool v0140_pool detected
CPU Pool updated to 'v0140_pool (convertable)' for vServer v0140
Filesystem /zones/v0140 defined as local directory.
Network management for vServer v0140 defined.
Network public for vServer v0140 defined.
Checking PatchLevel of Node g0050 ...
Analyzing PKGs of vServer v0140 ...
```

`rcadm -c convert_pool` converts such static CPU pools into the dynamic CPU pools used by rcadm. The conversion is executed while the vServer is running. The static CPU pool will be renamed and the attributes updated.

Benefits
In contrast to the static CPU pool a dynamic CPU pool is destroyed when a vServer stops and is recreated when the vServer is started again. This allows to migrate a vServer to other Nodes and construct the CPU pool on the new Node dynamically.

```
-bash-4.1$ rcadm -c convert_pool vserver=v0140

converting cpu pool of vServer v0140
vserver v0140 uses pool v0140_pool
Resource set: CPUS=2 Importance=1


-bash-4.1$ rcadm -c show vserver=v0140

            Resource Management settings for: v0140

       Property Name   Current Setting       Uncommitted Setting   Units
          CPU_Shares:  -                     -                     BaseUnits
             CPU_Cap:  -                     -                     BaseUnits
                CPUs:  2                     -                     CPUs
          Importance:  1                     -                     Weight
                 RAM:  -                     -                     MB
                SWAP:  -                     -                     MB
              Locked:  -                     -                     MB
                 LWP:  -                     -                     Units
             MSG_ids:  -                     -                     Units
             SEM_ids:  -                     -                     Units
             SHM_ids:  -                     -                     Units
            SHM_Size:  -                     -                     MB
```

The convert will work if the following requirements are met
- Existing CPU pool is not shared with other Zones/vServers
- Existing CPU pset is not shared with other CPU pools
- No CPUs are pinned and no non-default objectives are defined

JomaSoft GmbH
Falkensteinstr. 54a
9000 St. Gallen
Switzerland

3.5 Customization of the VDCF environment

These VDCF configuration values can be changed to adjust VDCF for a customer environment.
To overwrite a VDCF variable add the appropriate value to `customize.cfg`:

Variable name	Description
`RESOURCE_NODE_RAM_MIN`	% of physical RAM reserved for the global zone. Defaults to 10%. You can't assign more than 90% of the RAM to your vServers.
`RESOURCE_NODE_CPU_MIN`	% of CPUs reserved for the global zone. Defaults to 0%.
`VIRTUAL_RCADM_SHOW_IGNORE_CSTATES`	vServer to ignore in the rcadm -c show Default: DETACHED Valid: ATTACHING|DEFINED|DETACHED|DETACHING