

VDCF - Virtual Datacenter Control Framework for the Solaris™ Operating System

Monitoring

Version 2.1
22. December 2011

Copyright © 2005-2011 JomaSoft GmbH
All rights reserved.



Table of Contents

1 Introduction.....	3
1.1 Overview.....	3
1.2 Hardware Monitoring.....	3
1.2.1 Alarming.....	3
1.2.2 Requirements.....	3
1.3 High Availability (HA) Monitoring.....	4
1.3.1 Components.....	4
1.3.2 Node failure detection.....	4
1.3.3 Node Evacuation sequence.....	5
1.3.4 Requirements.....	6
1.4 Resource Monitoring.....	7
1.4.1 Requirements.....	7
2 Installation and Configuration	8
2.1 Prerequisites.....	8
2.2 Installation.....	8
2.3 Configuration.....	9
2.3.1 Granting User Access.....	9
2.3.2 Customizing Hardware Monitoring.....	9
2.3.3 Customizing High Availability (HA) Monitoring.....	11
2.3.4 Customizing Resource Monitoring.....	14
3 Usage.....	15
3.1 Hardware Monitoring.....	15
3.1.1 Enabling / Disabling.....	15
3.1.2 Check node manually.....	15
3.1.3 System Locator LED.....	15
3.1.4 Display Hardware state	16
3.1.5 Clear hardware state history.....	18
3.2 High Availability (HA) Monitoring.....	19
3.2.1 Enabling / Disabling.....	19
3.2.2 Display Node State.....	20
3.2.3 Suspending Nodes.....	21
3.2.4 Fallback after Evacuation.....	21
3.3 Resource Monitoring.....	22
3.3.1 Enable resource monitoring.....	22
3.3.2 Usage Collector.....	22
3.3.3 Usage Aggregator.....	22
3.3.4 Disable resource monitoring.....	23
3.3.5 Update node data manually.....	23
3.3.6 Show resource consumption data.....	24
4 Appendixes.....	26
4.1 Node failover detection details.....	26

1 Introduction

This documentation describes the Monitoring features of the Virtual Datacenter Control Framework (VDCF) for the Solaris Operating System and explains how to use this features.

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

<i>VDCF Base – Administration Guide</i>	for information about VDCF base usage
<i>VDCF vServer – Administration Guide</i>	for information about VDCF vServer usage
<i>VDCF – Resource Management</i>	for information about VDCF Resource Management usage

1.1 Overview

VDCF Monitoring is a VDCF Enterprise extension available to VDCF Enterprise customers.

This extension consists of three separate components:

- Hardware Monitoring to detect hardware failures
- Resource Monitoring to collect and display resource usage of global and local Solaris zones
- High Availability (HA) Monitoring to automatically failover, if a data center or hardware fails

While VDCF Resource Monitoring collects and displays resource usage, VDCF Resource Management is used to configure resource limits.

1.2 Hardware Monitoring

The VDCF Hardware Monitoring connects periodically to the system controller of all nodes defined in the VDCF repository and checks for hardware and OS state.

1.2.1 Alarming

If the VDCF Hardware Monitoring detects hardware failures the user may be informed in two way

- sending e-Mails
- executing a script to integrate other software products

1.2.2 Requirements

As the Hardware monitor is based on information from the system controller it's required to configure a 'console' for each node within VDCF.

1.3 High Availability (HA) Monitoring

The VDCF High Availability feature is used to monitor the health of nodes. If a failed node is discovered the node may be stopped and/or the node evacuation logic is called to failover all vServers to other nodes. This evacuation is based on resource usage information to avoid overloading the remaining nodes.

This solution is positioned between manual failover initiated by a System Administrator and a full-featured failover solution using Cluster software. This VDCF HA feature is able to handle the typical node failures, like boot disk issues, network outages, platform errors like CPU, memory problems or power supply failures. The goal is to keep this solution as simple and usable as possible, therefore it doesn't require cluster interconnect between the nodes and doesn't check and handle issues with SAN connections like a Cluster software does.

1.3.1 Components

The HA monitor is built from several components:

Each node participating has a daemon (SMF service `vdcf_keep_alive`) installed that calls periodically into the management server. These keep-alive messages are stored within the `/var/opt/jomasoft/vdcf/keepalive` directory.

The second component is the monitoring daemon (`hamon_watchd`) on the VDCF management server. This daemon consists of two processes. One (`hamon_monitord`) is used to monitor for keep-alive messages at the interval `HAMON_KEEP_ALIVE_INTERVAL` seconds from all participating Nodes. The second process (`hamon_checkd`) is used to check and act upon a failed node was detected.

1.3.2 Node failure detection

A node is considered as failed if the following rules are met

- no keep-alive messages are received within the defined Threshold (`HAMON_KEEP_ALIVE_ACTION_THOLD`)
- a ssh connection from VDCF to the node fails
- node's system controller / console does not respond or node is at the OK prompt or powered off

An optional network probing rule may be activated by setting `HAMON_CHECK_NETWORK_PROBES="true"`. If the node system controller is not reachable, the reason may be network-related or the Node has no power at all. If this setting is true, VDCF tries to connect to configured intermediate network equipment. If the network equipment is reachable, VDCF considers its network connection as good and therefore the nodes as failed.

For more details about this failure detection consult the Appendix 4.1 Node failure detection details.

Based on the description above, the VDCF HA monitor is able to detect the following failures:

- complete hardware failure of the Node
- accidentally shutdown of Node by a System Administrator
- failure of network interfaces of the Node

The following failures are detected if Network Probing is activated and properly configured:

- complete power-failure of the Node (system controller not reachable)
- complete data center failure, as long as the network is still reachable (depends on configuration)

The following failures are **NOT** detected:

- failure or config issues of SAN components
- complete data center failure, if the network is affected (depends on configuration)
- accidentally network interface miss configuration by a System Administrator

For setting up and configuring your HA environment consulting services from JomaSoft is available.

1.3.3 Node Evacuation sequence

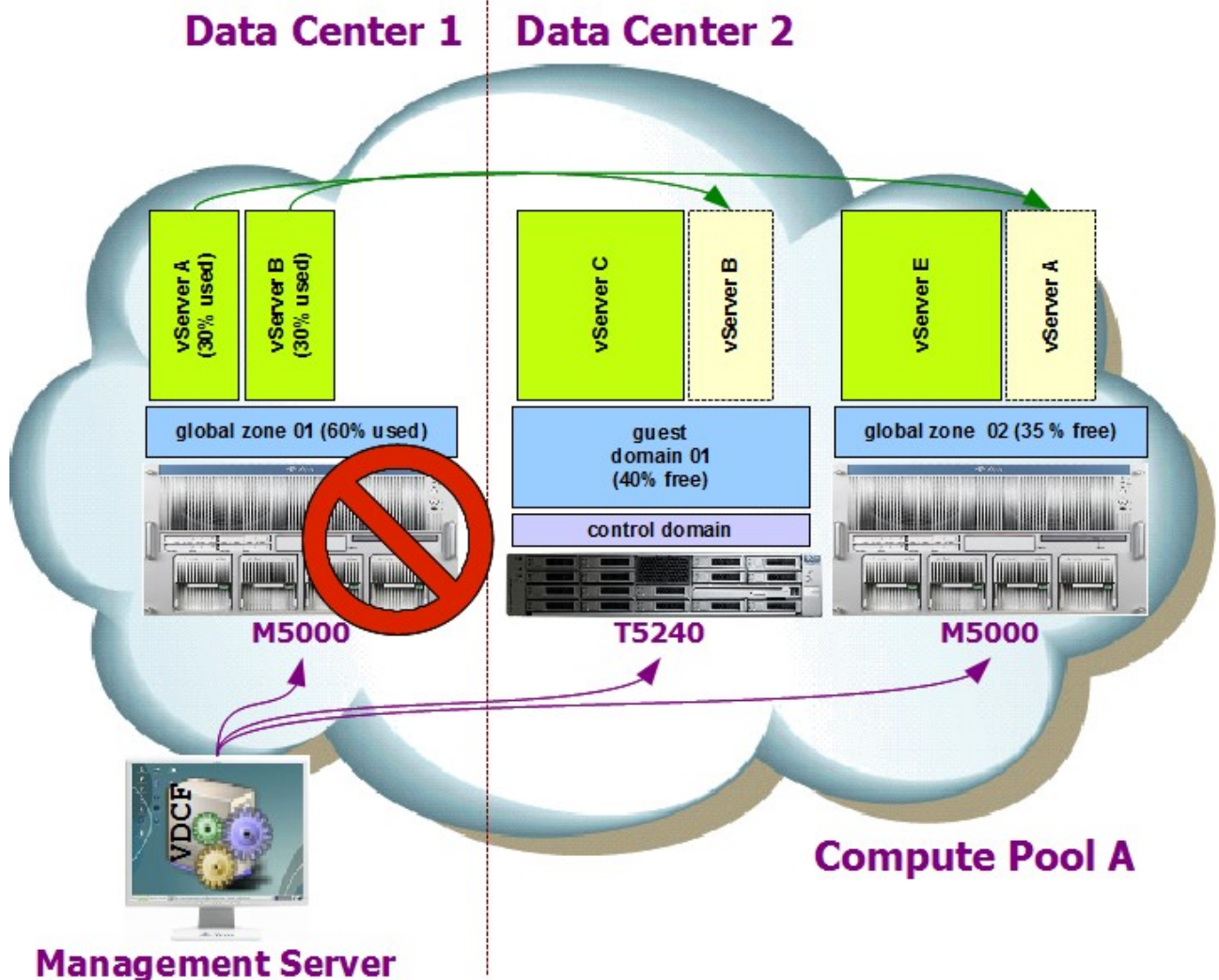
If Node Evacuation is configured, all vServer of a faulted node are evacuated (failed over) to other active nodes in the same compute pool. The procedure to detect the possible target nodes looks as follows:

1. For each vServer we get a list of candidate nodes (using `vserver -c show candidates`).
2. Based on the resource usage data reported from resource monitoring we select a possible target node for each vServer.
3. Because the source node isn't reachable anymore we do a `vserver detach force`.
4. Then we try to attach and boot the vServer on the new node.
5. If attach has failed we try the same procedure on the next possible target node until all vServers are evacuated or no more target nodes are left.

Upgrade on attach is supported by setting the value `HAMON_EVACUATE_UPGRADE` to true in the `customize.cfg` file.

The sequence of vServer migration is ordered by the vServer category and/or priority. See configuration items for more details.

The following picture illustrates the migrations if the M5000 in Data Center 1 fails.



The Node Evacuation can be started manually using the command `node -c evacuate`.

1.3.4 Requirements

As the HA monitor is monitoring the console and is trying to shutdown a failed node through the system controller, it's required to configure a 'console' for each node within VDCF.

The node evacuation logic is based on resource information from Resource Monitoring. Activate VDCF Resource Monitoring on all participating nodes is therefore required.

1.4 Resource Monitoring

Resource Monitoring may be enabled (and disabled) individually for each node. A usage collector service is then started on the node. This service is recording the resource usage (CPU and memory) of the Node and all installed vServers. Periodically each node is pushing the recorded data onto the VDCF management server.

A cron job called 'Usage Data Collector' on the management server is importing the collected data periodically into the VDCF database.

A second cron job 'Usage Data Aggregator' is used to generate aggregated resource information. The aggregated data can be displayed on a daily, weekly, monthly or yearly base.

A third cron job is started / stopped together with the the 'Usage Data Collector' cron job. This cron job is evaluating the current average resource usage of nodes and vServers in the last 24 hours. This information may be used later by the HA monitor Node evacuation feature.

1.4.1 Requirements

The VDCF Resource Monitoring 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.

2 Installation and Configuration

2.1 Prerequisites

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

- JSvdcf-base 4.0.0 or later
- JSvdcf-vserver 4.0.0 or later

2.2 Installation

a) sparc platform

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

b) i386 platform

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

2.3 Configuration

2.3.1 Granting User Access

The VDCF Monitoring package introduces two new RBAC Profiles “VDCF hwmonitor Module” for the Hardware and Resource Monitoring and “VDCF hamonitor Module” for the HA feature.

Assign the RBAC profiles to your admin users.

2.3.2 Customizing Hardware Monitoring

Check Interval

By default the Hardware Monitoring cronjob is executed once an hour to check the state of all nodes.

You may display the current setting with this command:

```
$ hwmon -c status verbose
                          Central Monitor Component Status
                          HW Monitor: enabled

                          Central Monitor Component Timespec
Crontab timespec for HW Monitor: '15 * * * *'
```

To change this setting configure the cron timespec in customize.cfg using this variable:

```
export MONITOR_HW_INTERVAL="15 * * * *"
```

If the Hardware Monitor was already enabled before, you have to re-enable the cron job using these commands:

```
$ hwmon -c disable
HW Monitor: disabled
```

```
$ hwmon -c enable
HW Monitor: enabled
```

Alarming

The Hardware Monitor is able to send e-Mails, if a Hardware fault is detected. To enable this feature you have to set the the following variables in VDCF's customize.cfg:

```
export MONITOR_EVENT=true
export MONITOR_EVENT_EMAIL_LIST="user1@company.ch,user2@company.ch"
export MONITOR_EVENT_EMAIL_FROM="root@system.domain.ch"
```

Additionally it is supported to configure a script, which is called for every event. This feature allows you to forward events to your event management or ticketing system.

```
export MONITOR_EVENT_SCRIPT=/opt/company/bin/my_vdcf_hwmon_script
```

The 'MONITOR_EVENT_SCRIPT' will be executed if an monitor event occurs. The script may use the following 5 input arguments:

<node>	<new_state>	<date>	<time>	<logfile name>
<node>				Node name where the event occurred
<new_state>				Hardware and OS state after the event occurred e.g. OK:OS-RUN FAULTED:ON-OBP N/A:N/A
<date/time>				Date and time when the event was recorded
<logfile name>				Logfile on the management server where detailed information is stored

2.3.3 Customizing High Availability (HA) Monitoring

Keep Alive Interval

At each HAMON_KEEP_ALIVE_INTERVAL (default: 60 seconds) the Node is posting a keep-alive message to the management server.

Warning Threshold

After a number of missing keep-alive messages (HAMON_KEEP_ALIVE_WARN_THOLD (default 10) a e-Mail is sent if requested.

Define your e-Mail addresses as follows.

```
export HAMON_EVENT_EMAIL_LIST="user1@company.ch,user2@company.ch"
```

Action Threshold

A Node is considered as suspect if for HAMON_KEEP_ALIVE_ACTION_THOLD (default 20) intervals no keep-alive message has been posted.

You may display the current setting with the status command:

```
$ hamon -c status
      HA Monitor Information
      Interval: 60s
Warning Threshold: 10
Action Threshold: 20
      Watch Daemon: online
```

Actions on failure

Set HAMON_POWEROFF_ON_FAILURE to 'true' for a Node poweroff after failure detection. This setting is highly recommended. If this setting is false, you risk to corrupt your data if the filesystems are mount twice ...

And HAMON_EVACUATE_ON_FAILURE if all vServers of failed Nodes must be migrated to other running Nodes.

Node evacuation

A node is set to INACTIVE after an evacuate by default. Set HAMON_EVACUATE_INACTIVATE to 'false' to leave the node in ACTIVE state.

vServer do not upgrade on attach by default. Therefore nodes with an higher patch-levels aren't potential targets for the evacuated vServers. Set HAMON_EVACUATE_UPGRADE to 'true' to enable the upgrade on attach feature.

vServer target detection

First of all you have to categorize/prioritize your vServer using the `vserver -c modify` command. You may use categories to identify important or less important vServers and the priority to order within a category.

Then customize the evacuation variables in your `customize.cfg`. Use `VIRTUAL_EVACUATION_CATEGORY_ORDER` to identify the most important categories to be migrated first. Identify categories which you don't want to evacuate at all in `VIRTUAL_EVACUATION_IGNORE_CATEGORIES`.

By default CPU_Share resource definitions aren't used for target node detection. Set the `NODE_EVACUATION_USE_CPUSHARES` to 'true' to enable a check if the target node has enough free CPU_Shares available.

Network reachability check

To enable the network reachability check you have to configure the `HAMON_CHECK_NETWORK_PROBES` to true and the `HAMON_KEEP_ALIVE_NET_PROBE` variable. The monitor selects the target probe address based on the nodes MNGT interface and derives the network number from it. With this network number a search is done in `HAMON_KEEP_ALIVE_NET_PROBE` to find an associated probe address. If no match is found the default address is used if it is not set to 0.0.0.0. The variable `HAMON_KEEP_ALIVE_NET_PROBE` has the following format: "net_number:probe_ip default:probe_ip net_number:probe_ip"

The following are recommended settings for your customize.cfg:

```
export HAMON_EVENT_EMAIL_LIST="user1@company.ch,user2@company.ch"
export HAMON_POWEROFF_ON_FAILURE="true"
export HAMON_EVACUATE_ON_FAILURE="true"
export HAMON_EVACUATE_UPGRADE="true"

# migration category order (comma separated categories)
export VIRTUAL_EVACUATION_CATEGORY_ORDER="PROD,ACC,BANK1"
# migration ignore categories (comma separated categories)
export VIRTUAL_EVACUATION_IGNORE_CATEGORIES="TEST,MAINT"
```

Optional settings

1. To lower the reaction times (Warn after 5 Mins, instead of 10 / Action 20 → 10)

```
export HAMON_KEEP_ALIVE_WARN_THRESHOLD="5"
export HAMON_KEEP_ALIVE_ACTION_THRESHOLD="10"
```

2. To take CPU_Shares into account for the check of free resources on target nodes.

```
export HAMON_EVACUATE_USE_CPUSHARES="true"
```

3. To enable Network Probing (depends on your network infrastructure)

```
export HAMON_CHECK_NETWORK_PROBES="true"
export HAMON_KEEP_ALIVE_NET_PROBE="192.168.0.0:192.168.0.1 10.1.1.0:10.1.1.1"
```

2.3.4 Customizing Resource Monitoring

You may customize some aspects of the resource monitoring by overwriting this VDCF variables using the customize.cfg.

Usage interval

With this variable you may set the interval used to get zone usage information on the compute node in seconds. Using the default value of 60 produces a usage record every minute.

```
export MONITOR_ZONE_USAGE_INTERVAL=60
```

Usage delivery

The number of samples accumulated before delivery to the VDCF management server happens. The actual time between delivery of zone usage information is computed by `MONITOR_ZONE_USAGE_INTERVAL * MONITOR_ZONE_USAGE_DELIVERY`.

```
export MONITOR_ZONE_USAGE_DELIVERY=60
```

Collector and aggregator interval

You may display the current cron timespec setting with this command:

```
$ rcmon -c status verbose

                          Central Monitor Component Status
                          Usage Data Collector: enabled
                          Usage Data Aggregation: enabled

                          Central Monitor Component Timespec
                          Crontab timespec for Usage Data Collector: '5,25,45 * * * *'
                          Crontab timespec for Usage Data Aggregation: '0 6 * * *'
                          Crontab timespec for Usage Data 24h average: '0 23 * * *'
```

To change this settings configure the cron timespec in customize.cfg using these variables:

```
export MONITOR_USAGE_TX_INTERVAL="5,25,45 * * * *"
export MONITOR_AGGR_INTERVAL="0 6 * * *"
export CURRENT_RES_USAGE_UPDATE_INTERVAL="0 23 * * *"
```

If resource monitoring was already enabled before, you have to re-enable the cron jobs using these commands. (The 24h average cron job is controlled together with the collector cron job):

```
$ rcmon -c disable aggregator
$ rcmon -c enable aggregator

$ rcmon -c disable collector
$ rcmon -c enable collector
```

3 Usage

3.1 Hardware Monitoring

3.1.1 Enabling / Disabling

The hardware monitoring feature can be enabled/disabled globally.

```
$ hwmon -c enable  
$ hwmon -c disable
```

Use the status command to display the current state of hardware monitoring:

```
$ hwmon -c status verbose  
Central Monitor Component Status  
HW Monitor: enabled  
  
Central Monitor Component Timespec  
Crontab timespec for HW Monitor: '15 * * * *'
```

3.1.2 Check node manually

If the hwmon is enabled a cron job is checking periodical the state of all nodes.

To check a node manually you may issue this command:

```
$ hwmon -c update all | node=<node name>
```

3.1.3 System Locator LED

The hardware monitoring feature let you also control the system locator LED.

Displays the current state of the Locator LED as either on or off:

```
$ hwmon -c show_locator node=<node name>  
Locator led is OFF
```

Turns the locator LED on:

```
$ hwmon -c set_locator node=<node name>
```

Turns the locator LED off:

```
$ hwmon -c clear_locator node=<node name>
```



3.1.4 Display Hardware state

Using the show operation an overview about all Nodes is displayed.

```
$ hwmon -c show
```

```
Current Hardware State
  Node  Soft State  HW State  Last Change  Last Update
s0003  OS-RUN      OK        2010-08-18 09:15:01  2010-08-26 18:15:02
s0004  OS-RUN      OK        2010-01-30 15:38:18  2010-08-26 18:15:02
```

Using the node attribute and/or verbose flag the state history and details from the system controller is shown.

```
$ hwmon -c show node=s0003
```

```
Current Hardware State (SUNW,Sun-Fire-T1000/ALOMCMT)
  Node  Soft State  HW State  Last Change  Last Update
s0003  OS-RUN      OK        2010-08-18 09:15:01  2010-08-26 18:15:02
```

```
State Change History
  Node  Soft State  HW State  Event Date
s0003  OS-RUN      OK        2010-08-18 09:15:01
s0003  PWR-OFF     OK        2010-05-25 17:15:02
```

```
$ hwmon -c show node=s0003 verbose
```

```
Current Hardware State (SUNW,Sun-Fire-T1000/ALOMCMT)
  Node  Soft State  HW State  Last Change  Last Update
s0003  OS-RUN      OK        2010-08-18 09:15:01  2010-08-26 18:15:02
```

```
State Change History
  Node  Soft State  HW State  Event Date
s0003  OS-RUN      OK        2010-08-18 09:15:01
s0003  PWR-OFF     OK        2010-05-25 17:15:02
```

```
System Locator Status
Locator led is OFF
```

System Specific Status Informations

```
===== Environmental Status =====
```

```
-----
System Temperatures (Temperatures in Celsius):
-----
```

Sensor	Status	Temp	LowHard	LowSoft	LowWarn	HighWarn	HighSoft	HighHard
MB/T_AMB	OK	24	-10	-5	0	45	50	55
MB/CMP0/T_TCORE	OK	40	-10	-5	0	85	90	95
MB/CMP0/T_BCORE	OK	39	-10	-5	0	85	90	95
MB/IOB/T_CORE	OK	37	-10	-5	0	95	100	105



System Indicator Status:

```

-----
SYS/LOCATE          SYS/SERVICE          SYS/ACT
OFF                 OFF                   ON
-----
  
```

Fans (Speeds Revolution Per Minute):

```

-----
Sensor              Status              Speed  Warn  Low
-----
FT0/F0              OK                  9166  2240  1920
FT0/F1              OK                  8776  2240  1920
FT0/F2              OK                  8967  2240  1920
FT0/F3              OK                  8967  2240  1920
-----
  
```

Voltage sensors (in Volts):

```

-----
Sensor              Status              Voltage LowSoft LowWarn HighWarn HighSoft
-----
MB/V_VCORE          OK                  1.32   1.20   1.24   1.36   1.39
MB/V_VMEM           OK                  1.78   1.69   1.72   1.87   1.90
MB/V_VTT            OK                  0.87   0.84   0.86   0.93   0.95
MB/V_+1V2           OK                  1.18   1.09   1.11   1.28   1.30
MB/V_+1V5           OK                  1.48   1.36   1.39   1.60   1.63
MB/V_+2V5           OK                  2.50   2.27   2.32   2.67   2.72
MB/V_+3V3           OK                  3.29   3.06   3.10   3.49   3.53
MB/V_+5V            OK                  4.99   4.55   4.65   5.35   5.45
MB/V_+12V           OK                  12.18  10.92  11.16  12.84  13.08
MB/V_+3V3STBY      OK                  3.31   3.13   3.16   3.53   3.59
-----
  
```

System Load (in amps):

```

-----
Sensor              Status              Load    Warn Shutdown
-----
MB/I_VCORE          OK                  23.360  80.000  88.000
MB/I_VMEM           OK                  6.420   60.000  66.000
-----
  
```

Current sensors:

```

-----
Sensor              Status
-----
MB/BAT/V_BAT       OK
-----
  
```

Power Supplies:

```

-----
Supply  Status          Underspeed  Overtemp  Overvolt  Undervolt  Overcurrent
-----
PS0     OK                 OFF         OFF       OFF       OFF       OFF
-----
  
```

Last POST run: WED AUG 18 05:52:20 2010
 POST status: Passed all devices

No failures found in System

3.1.5 Clear hardware state history

A history record is generated for every hardware state change discovered by the periodical (or manually initiated) system check.

To clear all history records of a node:

```
$ hwmon -c clear_history node=<node name>
```

3.2 High Availability (HA) Monitoring

3.2.1 Enabling / Disabling

The HA monitoring feature can be enabled/disabled globally.

```
$ hamon -c enable daemon  
$ hamon -c disable daemon
```

Then each participating node has to be enabled too:

```
$ hamon -c enable node=<node name>  
$ hamon -c disable node=<node name>
```

Please notice that only non-cluster nodes may be enabled for HA monitoring.

To display the status of HA monitoring use this command:

```
$ hamon -c status  
  
    HA Monitor Information  
        Interval: 60s  
Warning Threshold: 10  
    Action Threshold: 20  
        Watch Daemon: online
```

3.2.2 Display Node State

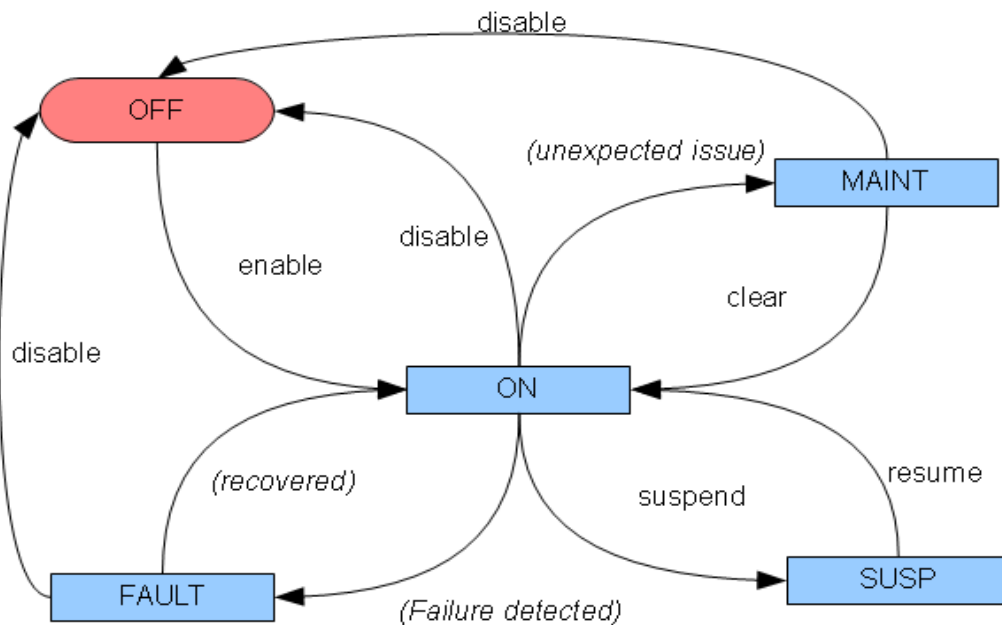
Using the show operation an overview about all Nodes is displayed.

```
$ hamon -c show
```

Node	Mon State	Ops State	Date	Details
s0003	ON	PROBING	2011-02-16 16:37:48	normal operation
s0009	ON	PROBING	2011-02-16 16:32:43	normal operation
s0010	ON	PROBING	2011-02-16 16:38:19	normal operation
s0004	FAULT	-	2011-02-16 16:45:22	console did not respond / not powered off

Each node has a Mon(itoring) State, which is influenced by the System Administrator using hamon operations and by the VDCF HA monitor.

The following diagram explains the possible states and actions:



3.2.3 Suspending Nodes

To avoid unnecessary failovers, it is required to suspend the Node from Monitoring if Maintenance is done on the Node. Suspend the Node before you shutdown the Node, for example to add more Memory using

```
$ hamon -c suspend node=s0003  
HA monitor suspended on Node s0003
```

```
$ hamon -c show node=s0003  
Node Mon State Ops State Date Details  
s0003 SUSP - 2011-02-16 16:57:19 -
```

```
$ hamon -c resume node=s0003  
HA monitor resumed for Node s0003
```

```
$ hamon -c show node=s0003  
Node Mon State Ops State Date Details  
s0003 ON PROBING 2011-02-16 16:57:33 normal operation
```

3.2.4 Fallback after Evacuation

Using the VDCF recommended settings, if a Node fails, the vServers are evacuated and the Node is set to state INACTIVE. This is done to avoid usage of that Node for new vServers.

You boot the Node when the issues, that caused the Node to fail, are solved, The HA Monitoring is then re-activated automatically. To use the Node for vServers again, you need to activate the Node again:

```
$ node -c activate name=mynode
```

The vServers are NOT automatically migrated back to the Node. You need to migrate the vServers manually back to your node using the migrate operation.

```
$ vserver -c migrate name=myvserver node=mynode shutdown
```

3.3 Resource Monitoring

3.3.1 Enable resource monitoring

The recording of resource usage information may be activated individually for each node. By enabling a node a `usage_collect` service is started on the Node. After the defined interval (`MONITOR_ZONE_USAGE_INTERVAL`) a usage record is saved locally on the node. After a defined number of records (`MONITOR_ZONE_USAGE_DELIVERY`) are saved the `usage_collect` service transfers the data to the VDCF management server.

To enable usage collection on nodes use this command:

```
$ rcmon -c enable      node=<node name> | node all
```

To display the status of resource monitoring for all nodes use this command:

```
$ rcmon -c status node
                                Central Monitor Component Status
                                Usage Data Collector: enabled
                                Usage Data 24h average: enabled
                                Usage Data Aggregation: enabled

                                Node Monitor Component Status
                                Usage Data Collection on s0002: enabled
                                Usage Data Collection on s0003: enabled
```

3.3.2 Usage Collector

The usage data transferred from the nodes is imported periodically into the VDCF repository using the 'Usage Data Collector' cron job.

You enable this collector using:

```
$ rcmon -c enable collector
```

When enabling the collector a further cron job is enabled: The 'Usage Data 24h average' cron job is a summary function to calculate the average resource usage of all nodes and vServers in the last 24 hours. To display that average data use the `rcmon -c summary` command.

3.3.3 Usage Aggregator

To avoid using up too much space on the VDCF management server VDCF offers a 'Usage Data Aggregation'. This cron job aggregates old data.

```
$ rcmon -c enable aggregator
```

Usage records older than a week are aggregated to a record per hour.

Usage records older than a month are aggregated to a record per day.

3.3.4 Disable resource monitoring

Same procedure as for enabling the resource monitoring components

Disable collection on nodes:

```
$ rcmon -c disable node=<node name> | node all
```

Disable Usage Data Collector:

```
$ rcmon -c disable collector
```

Disable Usage Data Aggregation:

```
$ rcmon -c disable aggregator
```

3.3.5 Update node data manually

You may request an update of the database with the newest usage data available.

This command restarts the usage collector service on the node and transfers back the current usage data file to the VDCF management server. Followed by an import into the VDCF repository.

```
$ rcmon -c update node=<node name> | node all
```



3.3.6 Show resource consumption data

To show the collected usage information for a vservers or a node use the show operation.

```
rcmon -c show          cpu | memory | memory_extended
                        hourly | daily | monthly | yearly
                        server=<server name>
                        [ verbose ]

                        [ gz_total | gzt ]

rcmon -c show          cpu | memory | memory_extended
                        from=<'time-spec'>
                        server=<server name>
                        [ to=<'time-spec'> ]
                        [ aggr=<aggr-spec> ]
                        [ verbose ]
                        [ gz_total | gzt ]
```

For explanation of the command flags and output, please see manpage 'rcmon -H show' for detailed information. Some examples:

The following command lists the available CPU usage information of the last hour with no further aggregation:

```
$ rcmon -c show server=s0180 cpu hourly

----- Pool -----   --- CpuShr ---   --- CpuSys ---   --- CpuUsr ---
--- CpuAll ---   -- CpuSAll --
DateTime          ID/Type  Max   Cur   All   Min /Avg /Max   Min /Avg /Max   Min /Avg /Max
Min /Avg /Max   Min /Avg /Max   Name
2010-08-26 18:48:18  30/priv 15   2     8.3% - 100% -   -   0.0% -   -   0.0% -
-   0.0% -   -   0.0% -   s0180
2010-08-26 18:49:19  30/priv 15   2     8.3% - 100% -   -   0.0% -   -   0.0% -
-   0.0% -   -   0.0% -   s0180
...
```

This command lists a nodes memory consumption during the last month. It includes summed up resource values of the global and the non global zones:

```
$ rcmon -c show server=s0003 memory monthly gzt

----- RamTot -----   ---- RamKern ----   ---- RamFree ----   ---- RamUse ----
---- RamUtil ----   ----- VmUse -----   ----- VmUtil -----
DateTime          Min / Avg / Max   Min / Avg / Max   Min / Avg / Max   Min / Avg / Max
Min / Avg / Max   Min / Avg / Max   Name
2010-07-26 23:59:07  -   1920M -   -   1625M -   -   427M -   -   455M -
-   24% -   -   367M -   -   18% -   s0003
2010-07-27 23:59:36  -   1920M -   -   1628M -   -   423M -   -   456M -
-   24% -   -   367M -   -   18% -   s0003
...
```



The following command lists the used memory resources of a vServer of the last 5 hours:

```
$ rcmon -c show server=s0180 memory from="-5 hours" aggr=hour
```

----- VmUtil -----			----- RamKern -----			----- RamUse -----			----- RamUtil -----			----- VmUse -----		
DateTime	Min / Avg / Max	Name	Min / Avg / Max	Min / Avg / Max	Min / Avg / Max	Min / Avg / Max	Min / Avg / Max	Min / Avg / Max	Min / Avg / Max	Min / Avg / Max	Min / Avg / Max	Min / Avg / Max		
2010-08-26 14:59:51	1614M 1614M 1615M	s0180	-	48M	-	-	12%	-	-	42M	-	2.0% 2.0% 2.0%		
2010-08-26 15:59:37	1614M 1615M 1615M	s0180	-	48M	-	-	12%	-	-	42M	-	2.0% 2.0% 2.0%		
2010-08-26 16:59:23	1615M 1615M 1616M	s0180	48M	48M	48M	12%	12%	12%	-	42M	-	2.0% 2.0% 2.0%		
2010-08-26 17:59:39	1615M 1616M 1618M	s0180	48M	49M	55M	12%	12%	14%	42M	43M	49M	2.0% 2.0% 2.3%		
2010-08-26 18:59:25	1617M 1617M 1618M	s0180	49M	49M	49M	12%	12%	12%	-	42M	-	- 2.0% -		
2010-08-26 19:47:04	1617M 1618M 1618M	s0180	-	49M	-	-	12%	-	-	42M	-	- 2.0% -		

Use this summary operation to display the average resource usage data of the last 24 hours. Results may be ordered by ram, cpu or server name in ascending or descending order. Default ordering is ram descending:

```
$ rcmon -c summary sortkey=cpu
```

24h resource usage average ordered by cpu/desc:

Node	Total RAM	Free RAM	Total CPU	Free CPU	LastUpdate	Comment
s0003	768	40	800	795	2011-10-11 23:00:28	Sol 11
s0006	2048	135	658	612	2011-12-06 23:00:20	Sol 10
s0009	1024	615	193	188	2011-12-06 23:00:17	Bank01

vServer	Used RAM	Used CPU	CPU Pool	LastUpdate	Comment
v0104	25	1	0	2011-11-30 23:00:33	Exkl IP im AccessNet
v0100	50	1	0	2011-12-04 23:00:19	ZFS vServer
v0101	50	1	0	2011-12-06 23:00:20	on Diskset
v0103	50	1	0	2011-12-06 23:00:21	Virtual Server v0103
v0105	50	1	0	2011-12-06 23:00:23	ufs to zfs
v0106	50	1	0	2011-12-06 23:00:26	VDCF Zone

4 Appendixes

4.1 Node failover detection details

A node is considered as failed if for a defined number of intervals no probe message has been posted from a node. The monitor will kick off an action after $(\text{HAMON_KEEP_ALIVE_ACTION_THOLD}+1) * \text{HAMON_KEEP_ALIVE_INTERVAL}$ seconds after a node is no longer submitting its keep alive messages.

The action part of the hamon_check goes through several steps until it considers a node as failed:

1. First of all network connectivity is verified by trying to check the status of the vdcf_keep_alive service on the suspect node. If the node can be reached and the check returns a service state other than enabled, the monitor tries to reestablish the vdcf_keep_alive service. If this succeeds, the monitor returns to normal operation and awaits the keep alive probe for this node. If the service state already was enabled and the monitor was able to query its state, it also returns to normal operation, assuming the probe failure was of temporary nature.
2. If network reachability of the suspect node is not given, the monitor tries to access the nodes system controller. If we successfully reach the system controller the monitor checks the node's console for a running operating system. In this case the monitor resumes normal operation, assuming a healthy node with keep-alive failures due to temporary network problems. If the console check returns no signs of live the node will be powered off, if configured so and its workload will be evacuated.
3. If the monitor is not able to reach the system controller and HAMON_CHECK_NETWORK_PROBES is true, the network will be checked. This is done by trying to reach intermediate network equipment as defined in HAMON_KEEP_ALIVE_NET_PROBE. If, based on this check, the network is considered as healthy, the suspect node will be assumed as failed and the workload is evacuated. If the network is considered as failed, the monitor resumes normal operation without acting on the suspect node.