VELOCITY SOFTWARE

zVRM
The Velocity Resource Manager

Velocity Software Inc.
196-D Castro Street
Mountain View CA 94041
650-964-8867

Velocity Software GmbH
Max-Joseph-Str. 5
D-68167 Mannheim
Germany
+49 (0)621 373844

Barton Robinson,
barton@velocitysoftware.com

If you can't measure it, I'm just not interested….

The point of zVRM: Dynamically size Linux servers to meet current workload requirements

Velocity Software's mission:

- Enhancing z/VM Platform Acceptance

Agenda:

- Processor cache: CPU Case Study
- Memory options
- CPU options
- Cpuplugd issues
- VMRM issues
- zVRM Overview

The "Large Server" Problem

Servers moving from x86 oversized

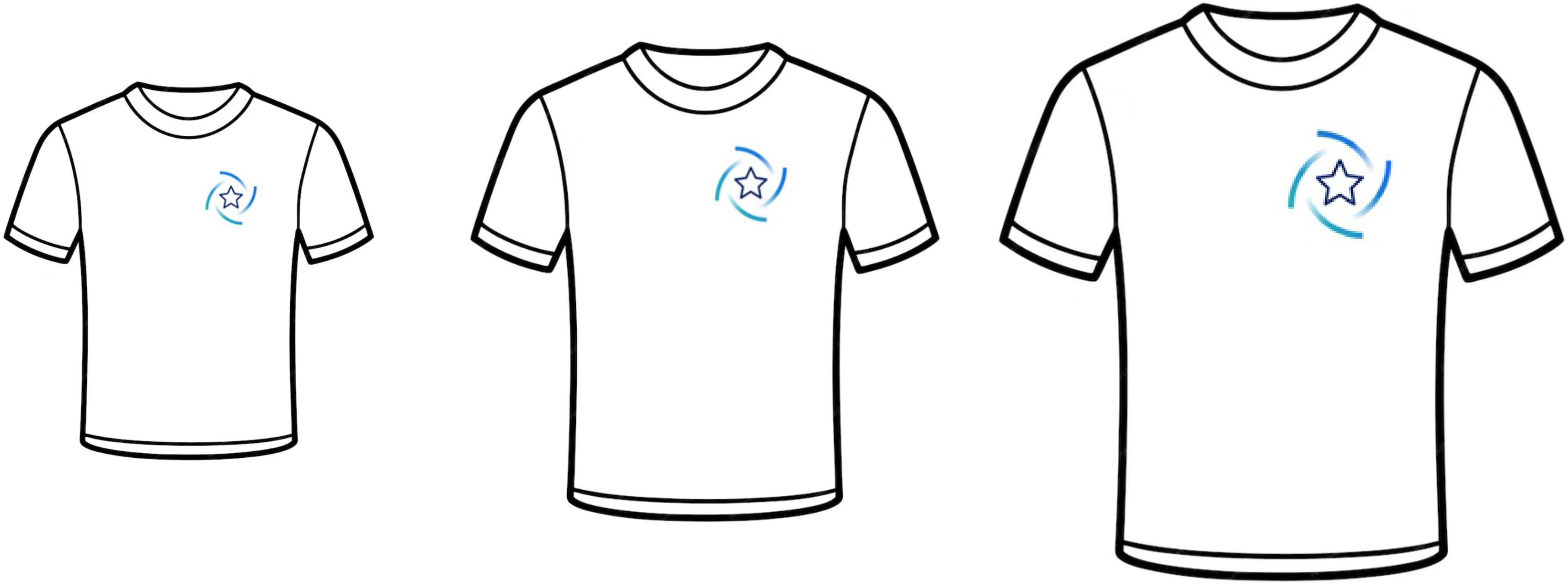
- Typically more (inexpensive) storage on x86
- More (less efficient) processors on x86
- Education and trust in z

Why large virtual machines?

- Intel servers require and outage to add resources
- Intel hardware is much less expensive
- The standard is to oversize everything

Cookie cutter virtual machines make life easier

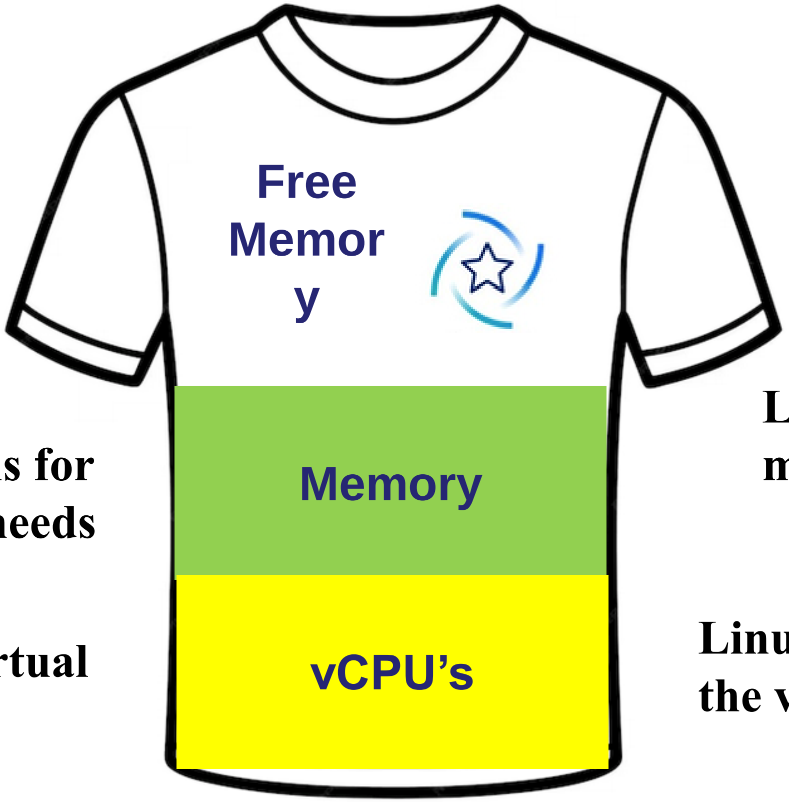
- Cloning is easier
- Requires little planning
- Easy to provide "small, medium and large"



z/VM can give the system back “free memory”

z/VM provides great feedback mechanisms for memory and vCPU needs

z/VM manages all virtual to real processors assignments



Linux touches all memory assigned to it

Linux can grow and shrink memory and vCPU’s

Linux spreads work out over all the vCPU’s assigned to it

The Result

Expensive Real storage is overcommitted

- Workloads variable
- Idle servers consume storage
- Storage requirement larger

Multiple processors result is spinlocks

- The more vCPUs, the more spinlocks (DIAG 44, 9C)
- Spinlocks cause both system overhead and delays
- Overhead and higher CPI results in more IFLs

Case Study

Cookie cutter servers

- 85gb
- 16 VCPU
- 100+ servers

What should server size be?

- What should CPU busy be?
- What should free storage be?

Linux Configuration Guideline Summary

Virtual machine size

- Minimize until some swap (swap out initialization pages)
- **Minimize vCPU counts to avoid overhead**

Swapping

- swap to virtual disk
- Define 2 virtual disks,
  - One to meet the average requirement
  - Second one for overflow - Insurance
- Use DIAG driver instead of FBA
  - Reduces I/O by factor of 8

Virtual processors

- **Minimize to meet the workload/application requirement**
- Ensure diag 9c, not 44

Infrastructure costs

- Minimize – shared resource architecture

Cpuplugd Operational Details

```
Report: ESAOPER      Operator/System Log
Monitor initialized:          at         on
10:15:00 LNXS3J2  vCPU stopped:       1
10:15:00 LNXS3J2  vCPU stopped:       2
10:15:00 LNXS3J2  vCPU stopped:       3
10:15:00 LNXS3J2  vCPU stopped:       4
10:15:00 LNXS3J2  vCPU stopped:       5
10:17:00 LNXS3J2  vCPU started:       1
10:17:00 LNXS3J2  vCPU started:       2
10:17:00 LNXS3J2  vCPU started:       3
10:17:00 LNXS3J2  vCPU started:       4
10:17:00 LNXS3J2  vCPU started:       5
10:17:00 LNXS3J2  vCPU started:       6
10:17:00 LNXS3J2  vCPU started:       7
10:17:00 LNXS3J2  vCPU started:       8
10:17:00 LNXS3J2  vCPU started:       9
10:17:00 LNXS3J2  vCPU started:      16
10:17:00 LNXS3J2  vCPU started:      17
10:17:00 LNXS3J2  vCPU started:      18
10:17:00 LNXS3J2  vCPU started:      19
10:17:00 LNXS3J2  vCPU started:      20
10:17:00 LNXS3J2  vCPU started:      21
10:29:00 LNXS3J2  vCPU stopped:      20
10:29:00 LNXS3J2  vCPU stopped:      21
10:30:00 LNXS3J2  vCPU started:      20
10:30:00 LNXS3J2  vCPU started:      21
10:36:00 LNXS3J2  vCPU stopped:      19
```

Operational changes are logged

- Evaluated at monitor start
- vCPU start/stops?

Cpuplugd at work

- Is it effective?

Virtual Machine Storage : ESAUCD2 (again)

```
Report: ESAUCD2      LINUX UCD Memory Analysis Report
-----------------------------------------------------------------
Node/    <-------------------------Storage Sizes (in MegaBytes)--
Time/    <--Real Storage--> <-----SWAP Storage----> Total <------
Date     Total  Avail Used  Total Avail Used  MIN   Avail CMM
-------- ------ ----- ----- ----- ----- ----- ----- ----- ------
10:30:00
LNXS1J2   85304 41209 44095  2810  2810     0  15.6 44019      0
LNXS1J4   85304 39480 45824  2810  2810     0  15.6 42290      0
LNXS2J2   85304 29881 55423  2810  2810     0  15.6 32691      0
LNXS3J2   85304 31377 53927  2810  2810     0  15.6 34187      0
```

Linux storage analysis (“85 GByte”)

- Swap Unused
- Available storage 140GB
- More if Linux was slightly constrained
- CMM not being utilized

“other” resource managers

VMRM - IBM

- No feedback mechanism -> no insight into application requirements
- No storage metrics available
- Would arbitrarily take storage away from servers
- Servers crashed for lack of storage
- Relative shares set “ridiculous”….
- Many controls added for manual control

Cpuplugd – opensource

- Each server individually manually configured
- Turning off vCPUs gives remaining vCPUs very high priority

Some history

Cooperative Memory Management (CMM1, z/VM 5.2)

- Provided command support for Linux to give up ram
- Builds the "CMM Balloon" and tells CP to re-use the storage
- Still available

IBM's VMRM Cooperative Memory Management (2007)

- CP XAUTOLOG VMRMSVM
- Used CMM based on external sizing
- Zero ability to look inside Linux for "free storage"
- Attempts to utilize resulted in bad things
- Adjusted SHARES based on business requirements
- "I saw some relative shares of 1 which was a bit of shock"

Collaborative Memory Management Assist (CMM2)

- Hardware assist, seemed too complicated

zVRM Overview

Centrally managed via zPRO

- By LPAR defaults
- By node group
- By node

zVPS provides feedback and performance metrics

CMM “balloon” used for storage management

- Small increments every interval
- Swapping causes immediate balloon pop
- Will minimize residency of stale storage

CPU vary on / off

- Uses the zPRO command interface
- Threshold to ensure minimum vCPU counts
- Target utilization controlled by zVRM

zVRM Controls

CMM “balloon” used for storage management

- Small (defined) increments every interval
- Swapping causes immediate balloon pop
- Will minimize residency of stale storage
- Maintains target percent of available storage

CPU vary on / off

- Uses the zPRO command interface
- Threshold to ensure minimum vCPU counts
- Target utilization controlled by zVRM
  - (higher for batch, lower for realtime)

zVRM Functions

Centralized control via zPRO interface

- One screen, all LPARs

All data sourced on one minute interval

- Standard zVPS interval
- Decisions based on Linux metrics

Storage:

- Reduces free storage incrementally

CPU Counts online managed to CPU utilization

- Requires zPRO API

Linux Server Requirements

Storage / CMM

- ‘modprobe cmm’

CPU Command interface

- zPRO command interface as part of zPRO

Control Parameters

Control by server, by user class

- Parameter settings
- SMSG interface with same format

ZVRM PARMS

- Provides default settings

zPRO -> z/VM Administration -> zVRM

- Portal to manage zVRM for enterprise

Note: zVRM runs on ALL managed LPARs

Control Parameters

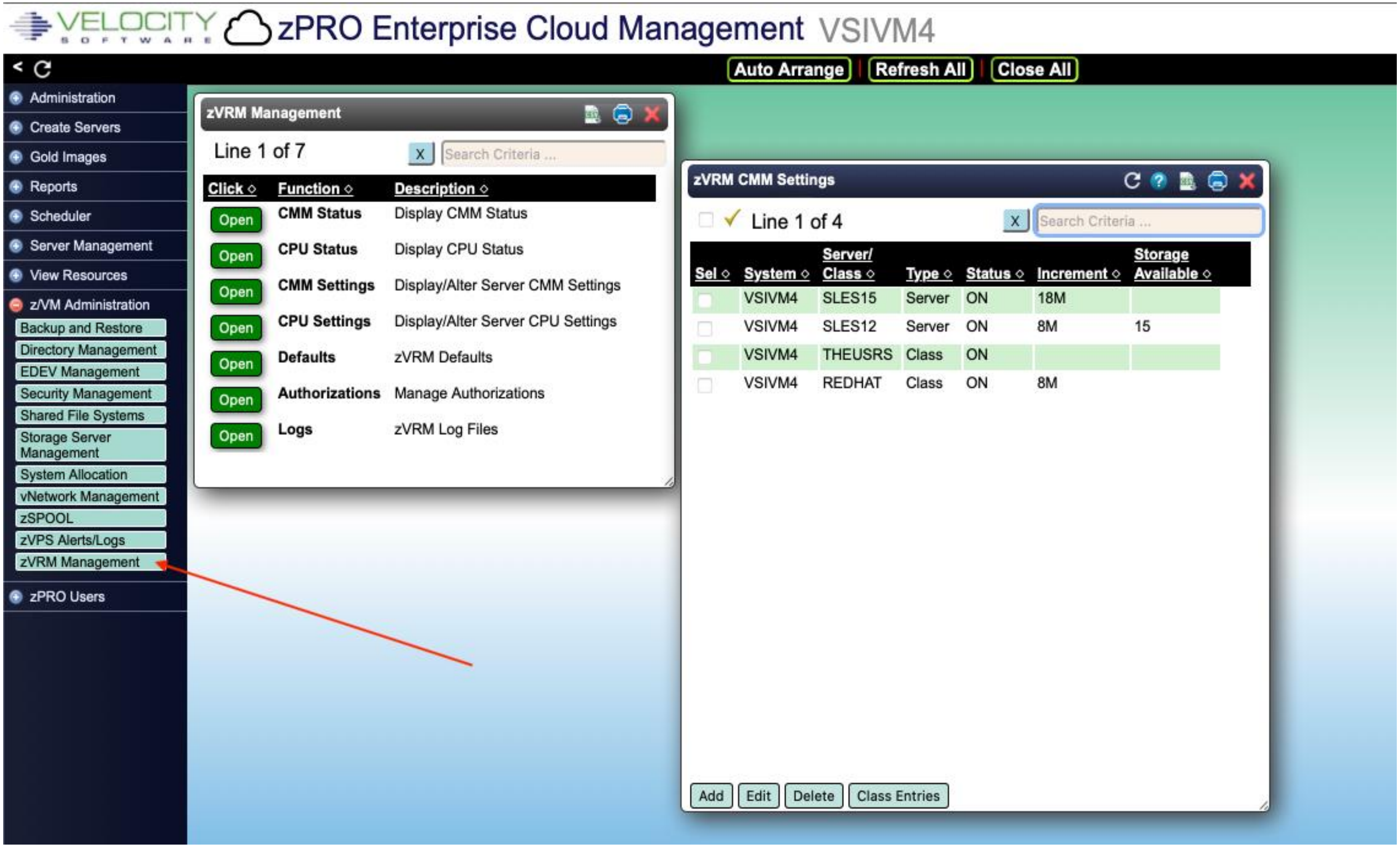
Storage control:

- CMM DEFAULT ON INCREMENT 32M
- CMM DEFAULT STGAVAIL   20   ;Minimum storage available
- CMM SRVR SLES15 ON INCREMENT 64M
- CMM SRVR SLES12 ON INCREMENT 128M
- CMM CLAS THEUSRS OFF
- CMM CLAS SERVERS ON

CPU Control

- CPU DEFAULT MINCPU 4
- CPU DEFAULT CPUPCT 25
- CPU SRVR   sles15 ON MINCPU 2
- CPU SRVR   sles15 ON CPUPCT 30

zPRO Enterprise Portal for zVRM



zVRM Management

Line 1 of 7

Click ◇	Function ◇	Description ◇
Open	CMM Status	Display CMM Status
Open	CPU Status	Display CPU Status
Open	CMM Settings	Display/Alter Server CMM Settings
Open	CPU Settings	Display/Alter Server CPU Settings
Open	Defaults	zVRM Defaults
Open	Authorizations	Manage Authorizations
Open	Logs	zVRM Log Files

zVRM CMM Settings

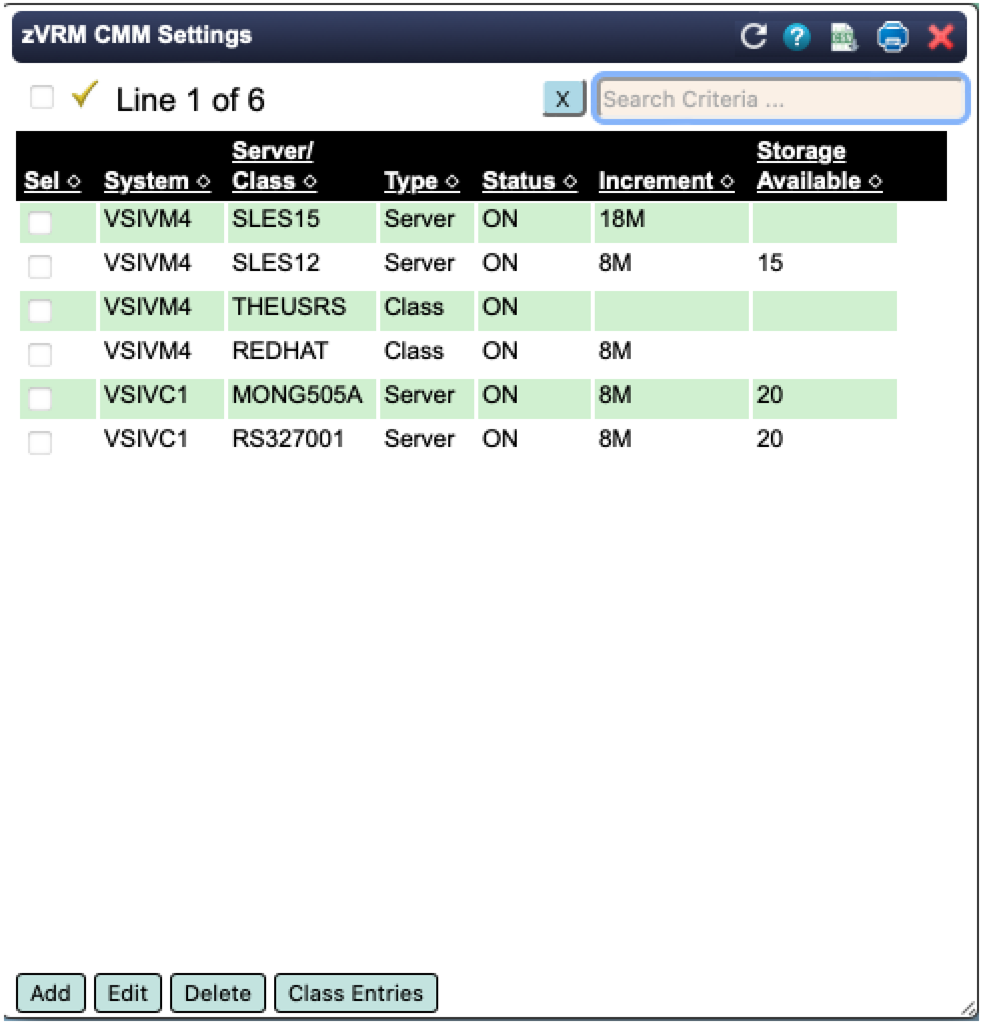
□ ✓ Line 1 of 4

Sel ◇	System ◇	Server/ Class ◇	Type ◇	Status ◇	Increment ◇	Storage Available ◇
□	VSIVM4	SLES15	Server	ON	18M	
□	VSIVM4	SLES12	Server	ON	8M	15
□	VSIVM4	THEUSRS	Class	ON		
□	VSIVM4	REDHAT	Class	ON	8M	

zPRO Enterprise Portal for zVRM

Manage across LPARs

- (VSIVM4,VSIVC1)
- Manage classes
- Manage servers



zVRM CMM Settings

□ ✓ Line 1 of 6

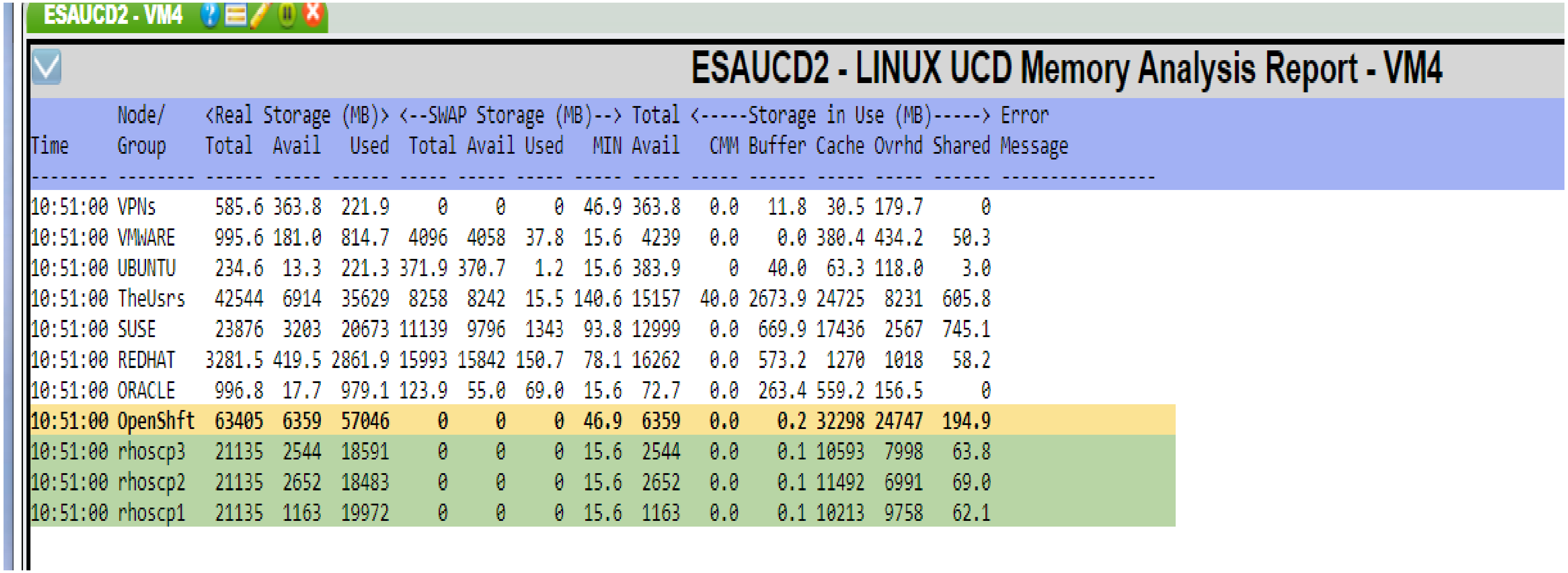
Sel ◇	System ◇	Server/ Class ◇	Type ◇	Status ◇	Increment ◇	Storage Available ◇
□	VSIVM4	SLES15	Server	ON	18M	
□	VSIVM4	SLES12	Server	ON	8M	15
□	VSIVM4	THEUSRS	Class	ON		
□	VSIVM4	REDHAT	Class	ON	8M	
□	VSIVC1	MONG505A	Server	ON	8M	20
□	VSIVC1	RS327001	Server	ON	8M	20

Management Requirements

Don’t drive a car without

- A speedometer….
- A gas gauge
- Headlights….

zVPS provides Linux metrics (by server)



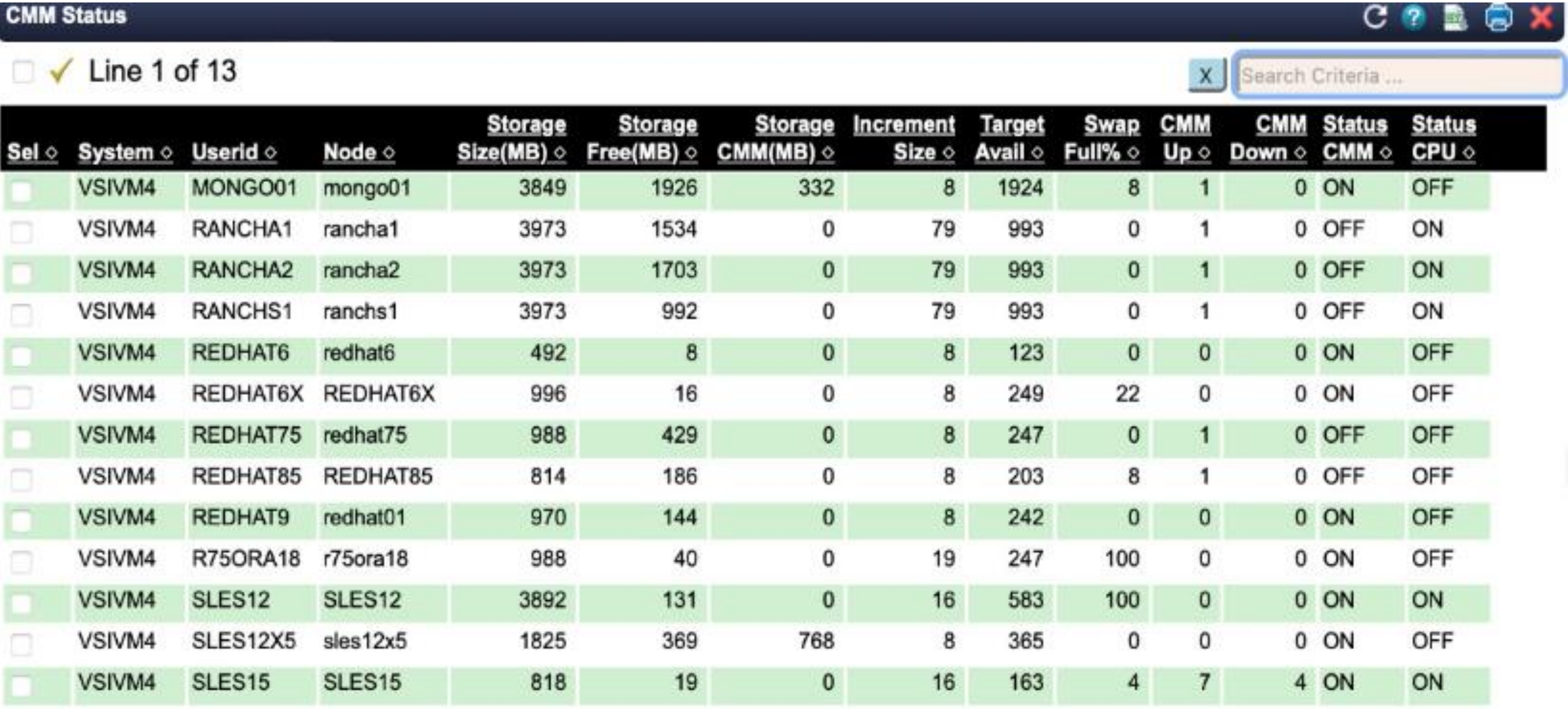
ESAUCD2 - LINUX UCD Memory Analysis Report - VM4

```
         Node/    <Real Storage (MB)> <--SWAP Storage (MB)--> Total <-----Storage in Use (MB)-----> Error
Time     Group    Total  Avail   Used Total Avail Used   MIN Avail   CMM Buffer Cache Ovrhd Shared Message
-------- -------- ------ ----- ------ ----- ----- ----- ----- ----- ----- ------ ----- ----- ------ ----------------
10:51:00 VPNs      585.6 363.8  221.9     0     0     0  46.9 363.8   0.0   11.8  30.5 179.7      0
10:51:00 VMWARE    995.6 181.0  814.7  4096  4058  37.8  15.6  4239   0.0    0.0 380.4 434.2   50.3
10:51:00 UBUNTU    234.6  13.3  221.3 371.9 370.7   1.2  15.6 383.9     0   40.0  63.3 118.0    3.0
10:51:00 TheUsrs   42544  6914  35629  8258  8242  15.5 140.6 15157  40.0 2673.9 24725  8231  605.8
10:51:00 SUSE      23876  3203  20673 11139  9796  1343  93.8 12999   0.0  669.9 17436  2567  745.1
10:51:00 REDHAT   3281.5 419.5 2861.9 15993 15842 150.7  78.1 16262   0.0  573.2  1270  1018   58.2
10:51:00 ORACLE    996.8  17.7  979.1 123.9  55.0  69.0  15.6  72.7   0.0  263.4 559.2 156.5      0
10:51:00 OpenShft  63405  6359  57046     0     0     0  46.9  6359   0.0    0.2 32298 24747  194.9
10:51:00 rhoscp3   21135  2544  18591     0     0     0  15.6  2544   0.0    0.1 10593  7998   63.8
10:51:00 rhoscp2   21135  2652  18483     0     0     0  15.6  2652   0.0    0.1 11492  6991   69.0
10:51:00 rhoscp1   21135  1163  19972     0     0     0  15.6  1163   0.0    0.1 10213  9758   62.1
```

zVRM Controls

Storage controls

- Free storage (percent)
- Increment size



CMM Status

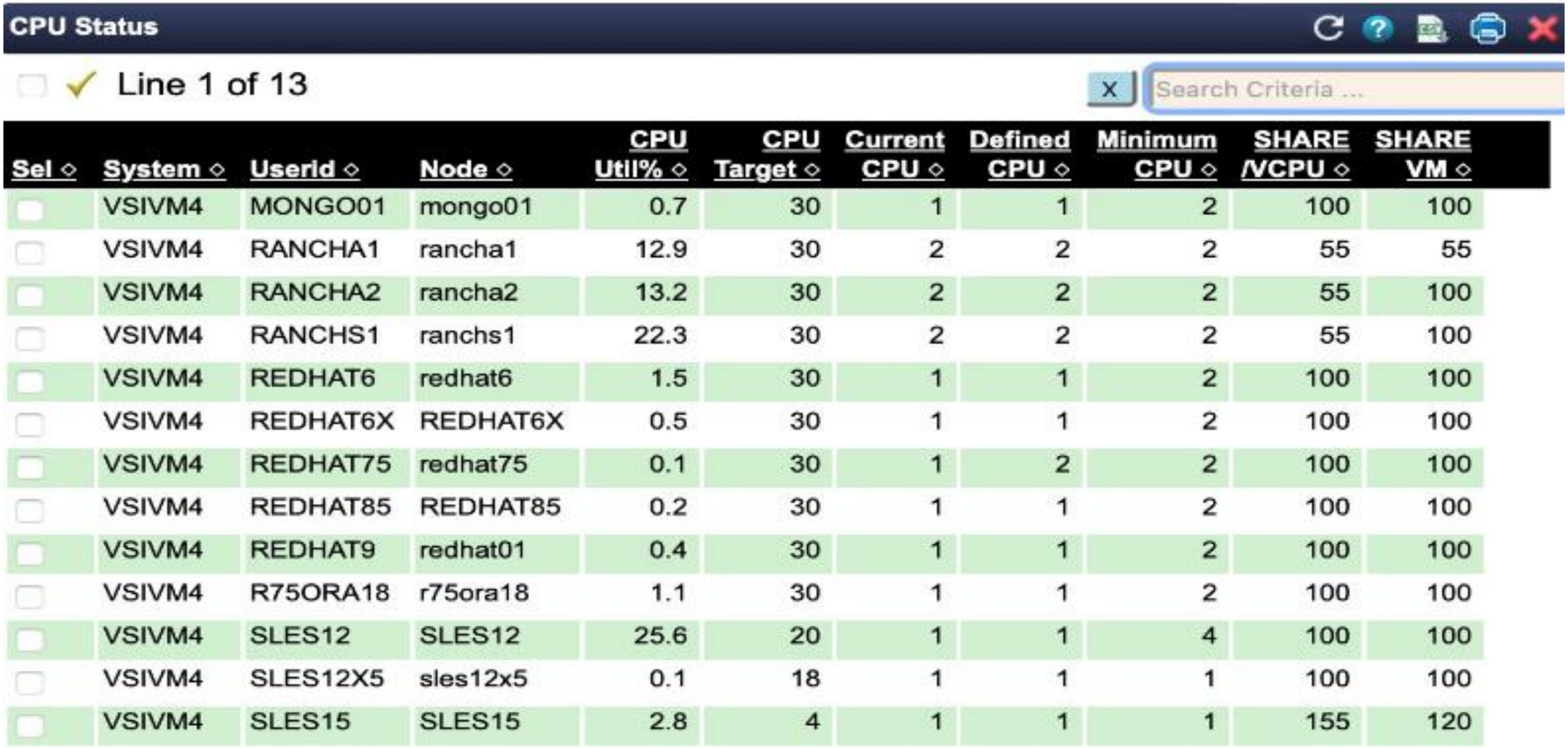
☐ ✓ Line 1 of 13

Sel ◇	System ◇	Userid ◇	Node ◇	Storage Size(MB) ◇	Storage Free(MB) ◇	Storage CMM(MB) ◇	Increment Size ◇	Target Avail ◇	Swap Full% ◇	CMM Up ◇	CMM Down ◇	Status CMM ◇	Status CPU ◇
☐	VSIVM4	MONGO01	mongo01	3849	1926	332	8	1924	8	1	0	ON	OFF
☐	VSIVM4	RANCHA1	rancha1	3973	1534	0	79	993	0	1	0	OFF	ON
☐	VSIVM4	RANCHA2	rancha2	3973	1703	0	79	993	0	1	0	OFF	ON
☐	VSIVM4	RANCHS1	ranchs1	3973	992	0	79	993	0	1	0	OFF	ON
☐	VSIVM4	REDHAT6	redhat6	492	8	0	8	123	0	0	0	ON	OFF
☐	VSIVM4	REDHAT6X	REDHAT6X	996	16	0	8	249	22	0	0	ON	OFF
☐	VSIVM4	REDHAT75	redhat75	988	429	0	8	247	0	1	0	OFF	OFF
☐	VSIVM4	REDHAT85	REDHAT85	814	186	0	8	203	8	1	0	OFF	OFF
☐	VSIVM4	REDHAT9	redhat01	970	144	0	8	242	0	0	0	ON	OFF
☐	VSIVM4	R75ORA18	r75ora18	988	40	0	19	247	100	0	0	ON	OFF
☐	VSIVM4	SLES12	SLES12	3892	131	0	16	583	100	0	0	ON	ON
☐	VSIVM4	SLES12X5	sles12x5	1825	369	768	8	365	0	0	0	ON	OFF
☐	VSIVM4	SLES15	SLES15	818	19	0	16	163	4	7	4	ON	ON

zVRM Controls

vCPU controls

- Share controls by vCPU
- Target vCPU utilization



CPU Status

□ ✓ Line 1 of 13

Sel ◇	System ◇	Userid ◇	Node ◇	CPU Util% ◇	CPU Target ◇	Current CPU ◇	Defined CPU ◇	Minimum CPU ◇	SHARE /VCPU ◇	SHARE VM ◇
□	VSIVM4	MONGO01	mongo01	0.7	30	1	1	2	100	100
□	VSIVM4	RANCHA1	rancha1	12.9	30	2	2	2	55	55
□	VSIVM4	RANCHA2	rancha2	13.2	30	2	2	2	55	100
□	VSIVM4	RANCHS1	ranchs1	22.3	30	2	2	2	55	100
□	VSIVM4	REDHAT6	redhat6	1.5	30	1	1	2	100	100
□	VSIVM4	REDHAT6X	REDHAT6X	0.5	30	1	1	2	100	100
□	VSIVM4	REDHAT75	redhat75	0.1	30	1	2	2	100	100
□	VSIVM4	REDHAT85	REDHAT85	0.2	30	1	1	2	100	100
□	VSIVM4	REDHAT9	redhat01	0.4	30	1	1	2	100	100
□	VSIVM4	R75ORA18	r75ora18	1.1	30	1	1	2	100	100
□	VSIVM4	SLES12	SLES12	25.6	20	1	1	4	100	100
□	VSIVM4	SLES12X5	sles12x5	0.1	18	1	1	1	100	100
□	VSIVM4	SLES15	SLES15	2.8	4	1	1	1	155	120

zVRM Summary

Cookie cutter servers manageable

- Storage / ram reduced to meet workload requirements
- vCPU counts managed to meet workload requirements
- Share settings, dispatch priorities managed

Centralized management

- zPRO function
- By node class, node

Optimize over commitment of resources

- Full feedback mechanisms
- Data driven decisions

Recap and Thankyou

zVRM

- Centralized resource management
- Will reduce memory requirements
- Will reduce CPU requirements
- Will make your machine faster
- Allows large “cookie cutter servers”
- Future opportunities

Questions and suggestions can be sent to ‘barton@velocitysoftware.com’