Performance Management for z/VM, Linux and RHOS

- Barton@VelocitySoftware.com
- HTTP://VelocitySoftware.com

"If you can't Measure it,

I am Just Not Interested ™"

Topics

Velocity Software overview

Performance Data collection technology

Container Architecture

- Docker,
- Openshift / kubernetes

Open Shift Performance Analysis

RHOS overhead analysis

Collecting container Performance data

z/OS zCX data, case study

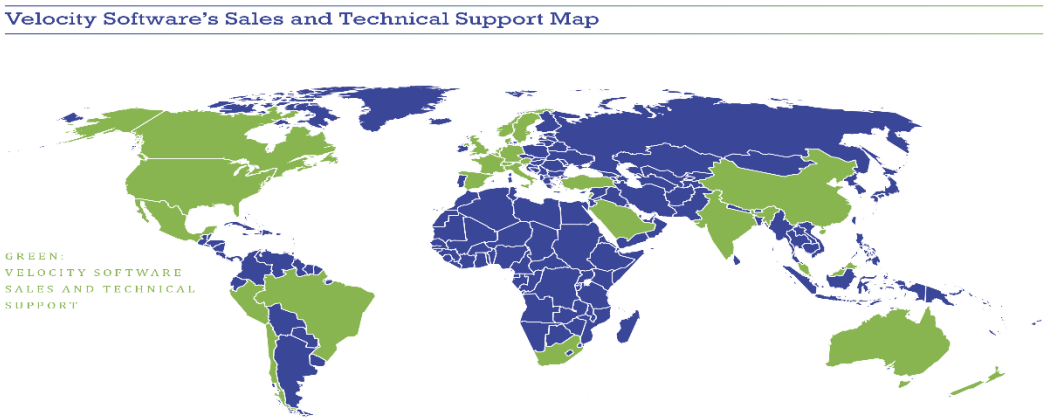
Who is Velocity Software

Who is VSI:

Since 1988 VSI has been the industry leader for z/VM performance management products, providing software, education and support to assist customers in optimizing their z/VM environment:
Currently managing 1000s of IFLs world wide in over 22 countries and 6 continents

Who are the customers and targets:

Financial Institutions, Banks, Governments, … but any client with z/VM installed on IBM Z (“large bank”,, “large bank”, utilizes zPRO).



Velocity Software

zVPS
PERFORMANCE SUITE
COMPLETE PERFORMANCE MANAGEMENT PACKAGE FOR Z/VM AND ALL SYSTEMS RUNNING UNDER Z/VM SUCH AS LINUX, Z/OS AND Z/VSE.

zPRO
CLOUD MANAGEMENT
VELOCITY SOFTWARE'S SOLUTION FOR IMPLEMENTING PRIVATE ON-PREM CLOUD AS WELL AS PROVIDING AN EASY-TO-USE WEB PAGE FOR SYSTEM PROGRAMMERS MANAGING Z/VM ENVIRONMENTS.

zTUNE
SUPPORT SUBSCRIPTION
VELOCITY SOFTWARE'S ELITE PERFORMANCE SUPPORT SERVICES. THE COMPONENT CREATES ADAILY REPORT THAT PROVIDES PERFORMANCE ADVICE AND GIVES ACCESS TO THE WORLD'S LEADING PERFORMANCE PROBLEM SOLVING EXPERIENCE AND EXPERTISE.

zVRM
RESOURCE MANAGER
REAL-TIME MANAGEMENT FACILITY TO MANAGE LINUX STORAGE/RAM RESOURCES AND VIRTUAL CPU COUNTS, BOTH TO MEET CURRENT WORKLOAD REQUIREMENTS.

zVPS: Performance Mgmt is a Process

Performance Analysis

- Understanding system, application performance
- Resolving current performance issues (z/VM, Linux, network)

Operational Alerts

- Supporting 100's/1000's of servers/containers in many locations
- Defining and automating operational support

Capacity Planning

- Providing input to the financial acquisition process

Accounting / Charge back

- Building a financial model for resource billing
- Who is consuming the resource?

Performance management can NOT be the performance problem
Black boxes are not managed by definition

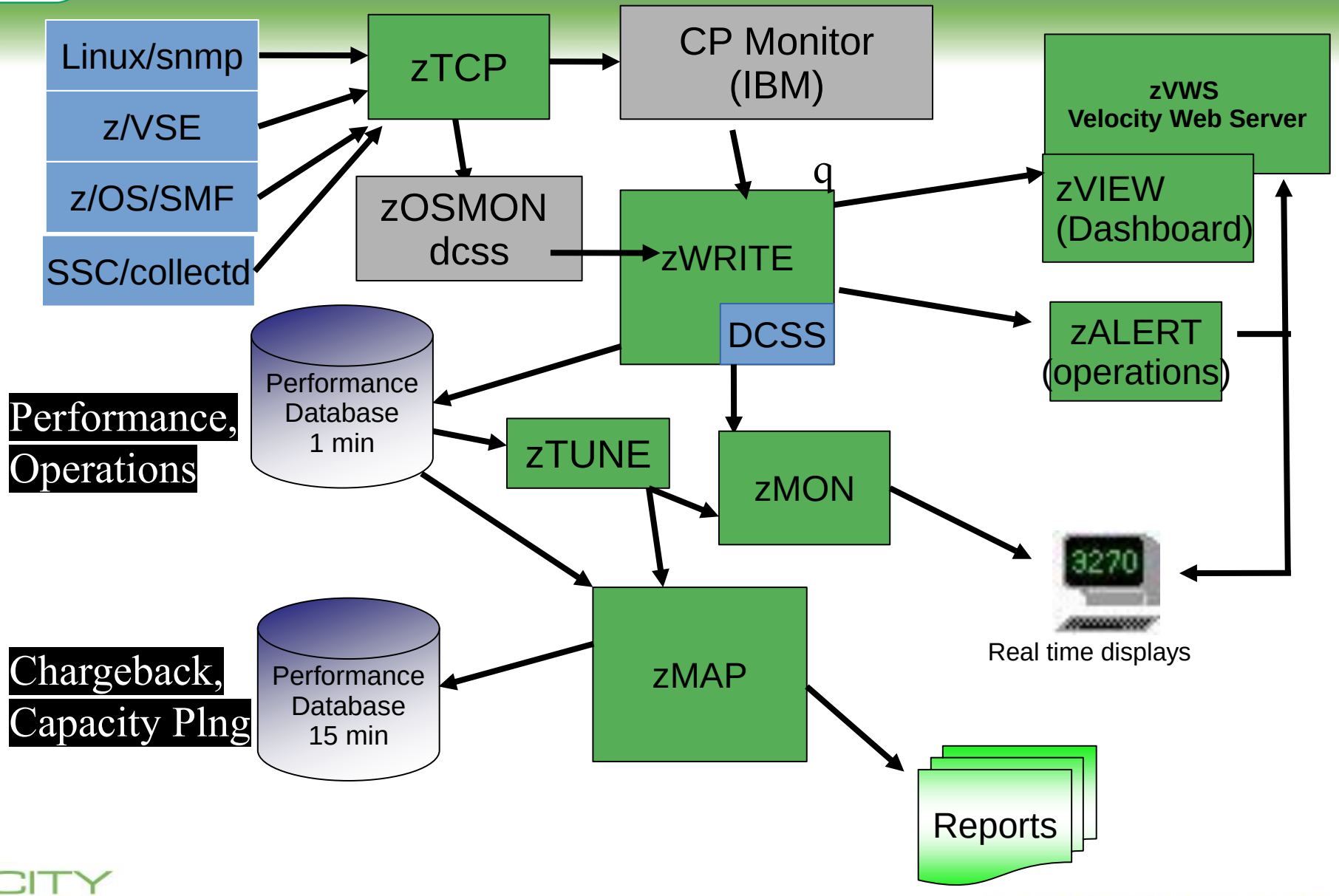
zVPS Components

- zMAP - Performance reporting, long term performance data base
- zMON – Real time interface, short term performance data base
- zVWS – Native webserver
- zTCP – Data collector (snmp, collectd, smf, dmf)

No charge components with graphical interface

- ESAEXTR – build your own reports, analyzer
- zVIEW – performance management dashboard
- zOPERATOR
  - Fully integrated operations console (replaces Ops Mgr)
- zALERT
  - Performance alerts and notifications (integrated)
  - Alerts to SNMP management console (NETCOOL, HPOpenView)
  - Email alerts, Cell phone text alerts

zVPS Technical Infrastructure



zTUNE – The z/VM Health Checker

Why zTUNE? (KI Based, Knowledge based)3

- Mainframes/LinuxOne environments are Complex
- Many things get overlooked even by experts
- Experts become experts by seeing many many performance problems
- What causes problems?
- When same problem multiple times, create a rule to look for it
- Checks for configuration “best practices”
- Inexpensive insurance to have the best skills when there is a problem

zTUNE Components

- zTUNE RULES – 100+ performance items that get checked
- ESATUNE report produced by zMAP, display on zVIEW
- Performance assistance on demand from Velocity Software experts
- Upload data for analysis at any time
- Ptrack ‘zTUNE’ sev1 alerts velocity management phones….

zTUNE – The z/VM Health Checker

Focus more now on simplifying problem resolution

User reports that applications complained about Linux on Z WAS performance:

```
Report: ESATUNE      Tuning Recommendation Report
------------------------------------------------------------
The following changes are suggestions by Velocity Software
 to enhance performance of this system.
However, Velocity Software takes no responsibility -
 all tuning is the responsibility of the installations.
Please call 650-964-8867 if you have any questions about
 these values, or suggestions on report enhancements.

USR2 User LINUX160 is paging excessively (75.0 per second)
     This user can be protected using SET RESERVED

SPL5 Spool utilization is 100% full.
     Perform Spool file analysis and purge large
     spool files, or force users currently writing
     excessively to spool.

*****zTUNE Evaluation   *************
XAC1 User total PROCESSOR WAIT excessive at 33 percent.
     Current reporting threshold set to 20.
     This is percent of inqueue time waiting for
     specific (PROCESSOR)resources to become available.
LPR3 LPAR share is too low, causing USER CPU Wait
     VM LPAR allocated share: 0.94 percent of total
     VM LPAR used 389 percent of allocated share
```

Another Velocity Software's Advantage

zVWS: Native generalized z/VM Webserver – base for modernization

- CMS based, Written in Assembler, very light weight
- Full function, Generalized server – completely eliminates need for SMAPI
- CGIs in rexx, assembler, pl1, etc (**Issue CP, CMS commands directly**)
- VERY EASY to develop web pages and applications

•VelocitySoftware.com (all runs on z/VM natively – Secure, Simple)

- VelocitySoftware.com, VelocitySoftware.de, VelocitySoftware.net, etc
- Linuxvm.org, MVMUA.org (and other user groups)
- VMWorkshop.org (greatest conference for z/VM)

•Many customers utilize zVWS for their own applications (govt, financial)

•Applications provided by Velocity Software

- zVIEW (Performance data presentation "dashboard"s)
- zPORTAL (GUI interface to managing zVPS)
- zPRO - on prem cloud, modernizing the platform in many ways
- No smapi, no java, No linux server requirements, no complexities

Monitoring Single (browser) pane of glass

All platforms provided, one technology

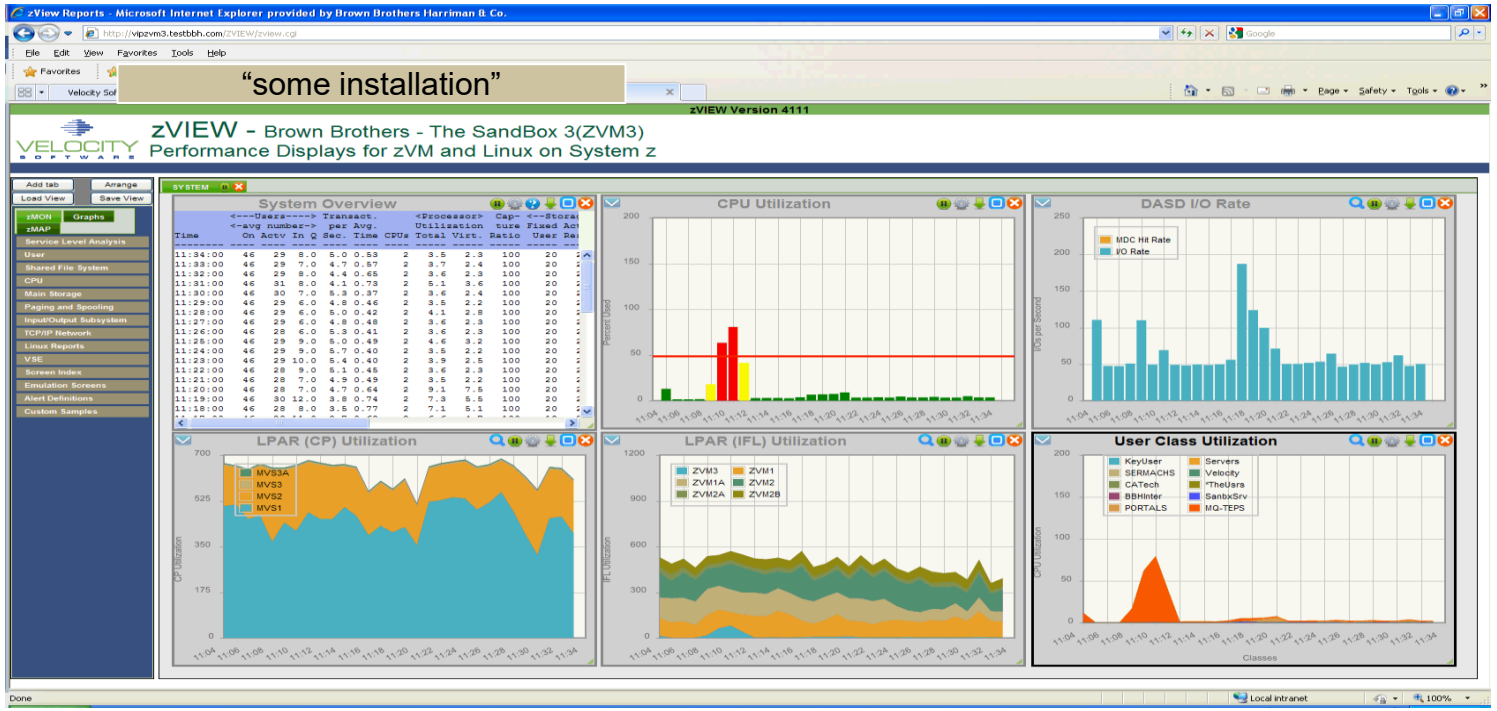
- **z/VM (CP monitor)**
- Networks (snmp)
- **Linux (“z” and “x”) (snmp)**
- z/VSE (VSEMON – no charge, snmp, DMF)
- z/OS (zOSMON: SMF record input)
- SSC (IBM Secure Software Container – collectd)
- Microsoft (snmp – no charge)
- VMWare/ ESX (snmp – no charge)

•Many applications

- Oracle (snmp)
- JVM (snmp)
- GPFS (snmp)
- Docker, **Kubernetes (OpenShift, Rancher)**
- MongoDB Enterprise

zVIEW – Graphical Interface

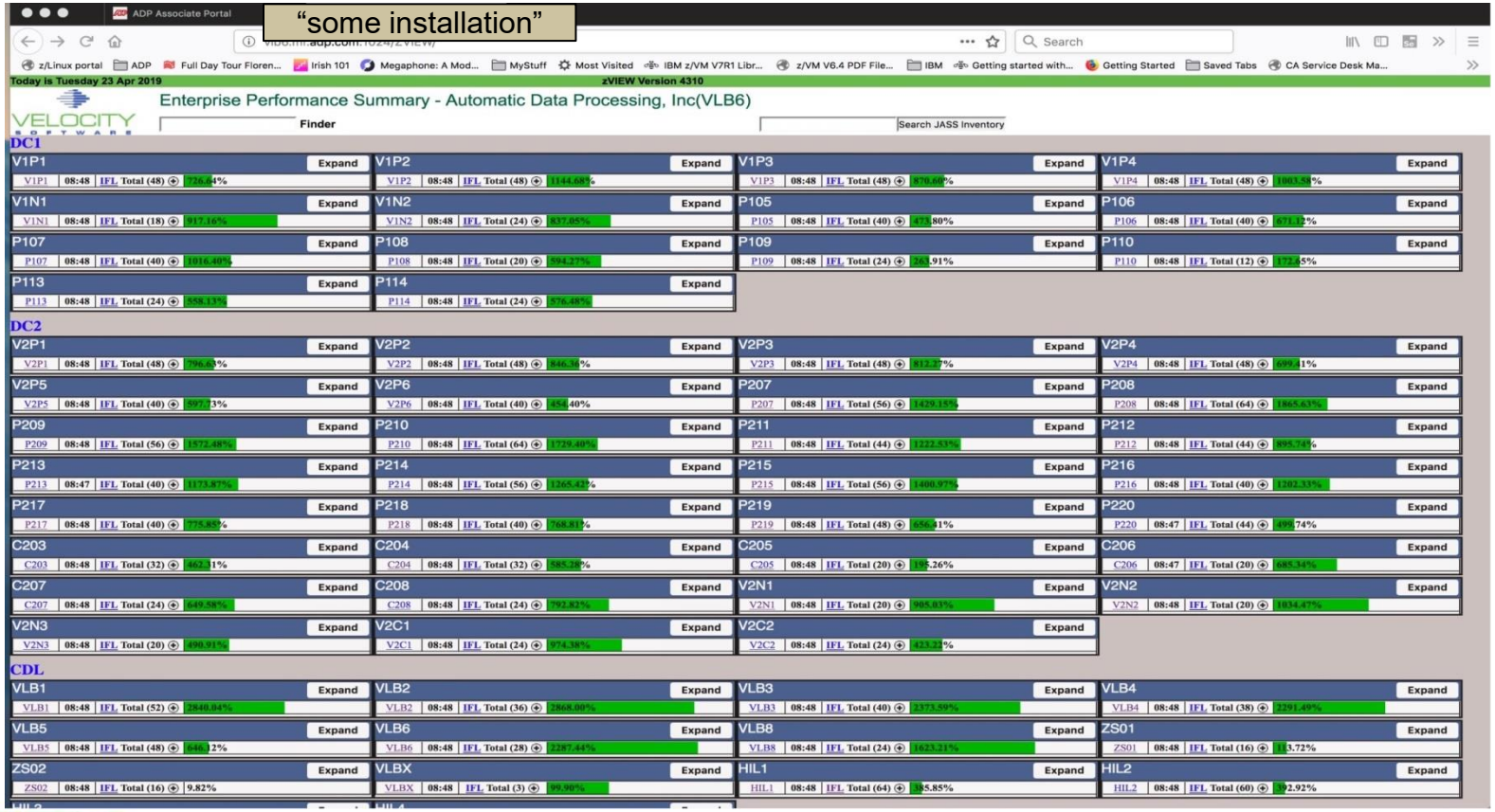
- End users define their dashboard(s)
- Many dashboards are provided (VSE, z/OS, Linux)
- Not just for Systems Programmers. (Applications, operations)
- Menu driven



zVPS Enterprise View is scalable

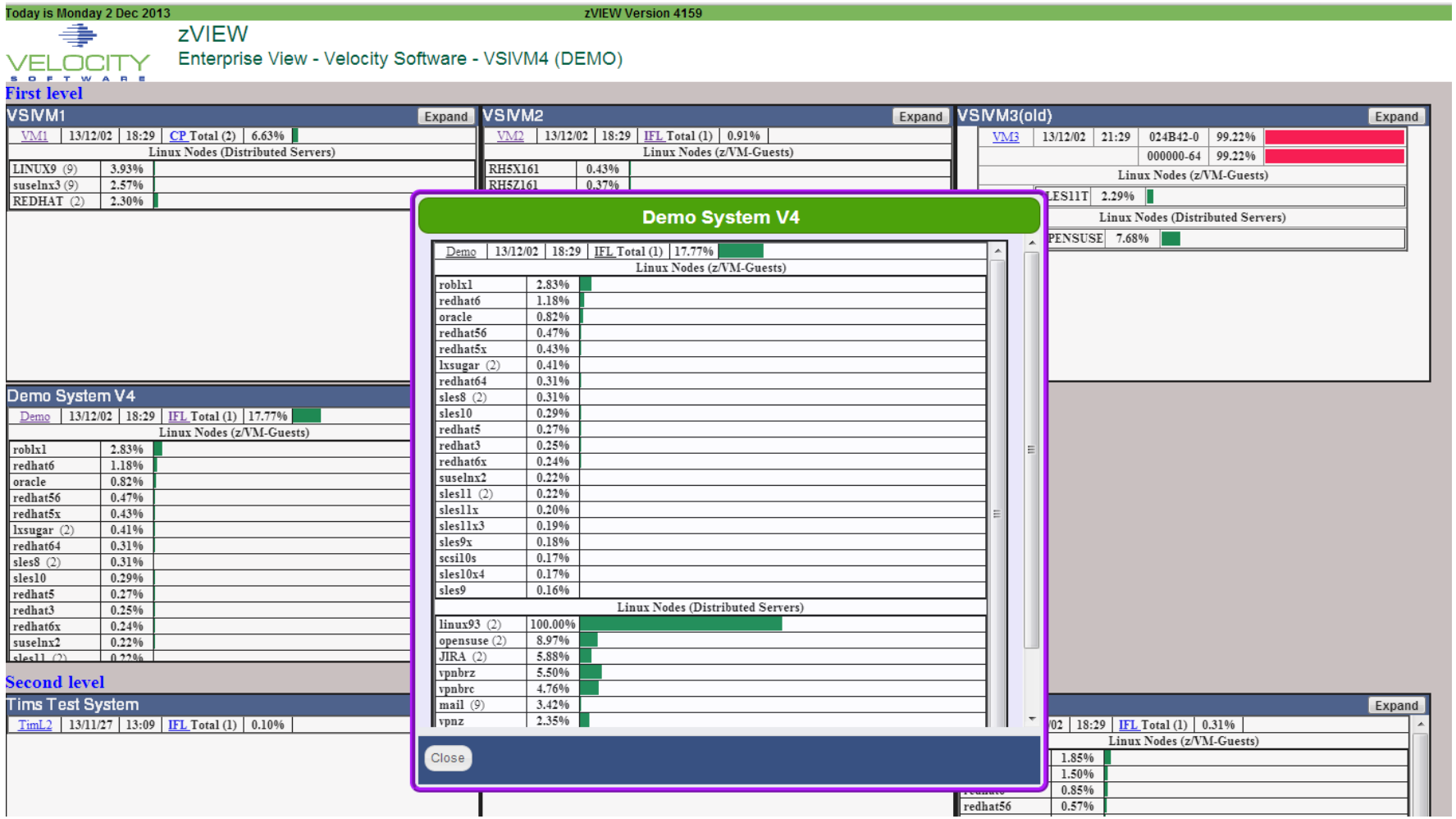
Single pane of glass (Very scalable)

- Data from “many” multiple LPARs(50) / geographies(3)



zVIEW Enterprise View: All LPARs

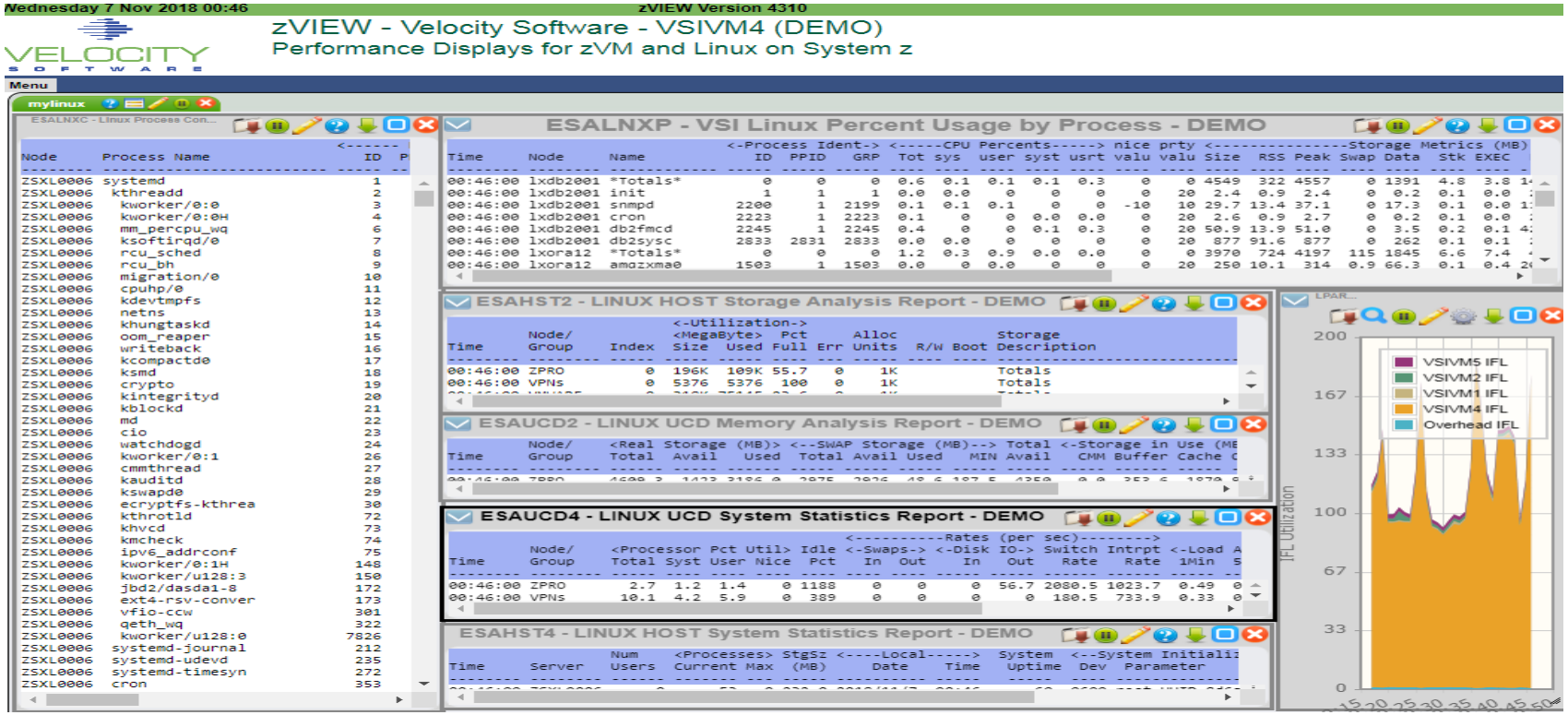
Tailorable, expandable, zoomable



One Click Linux Server performance

End users define their dashboard(s)

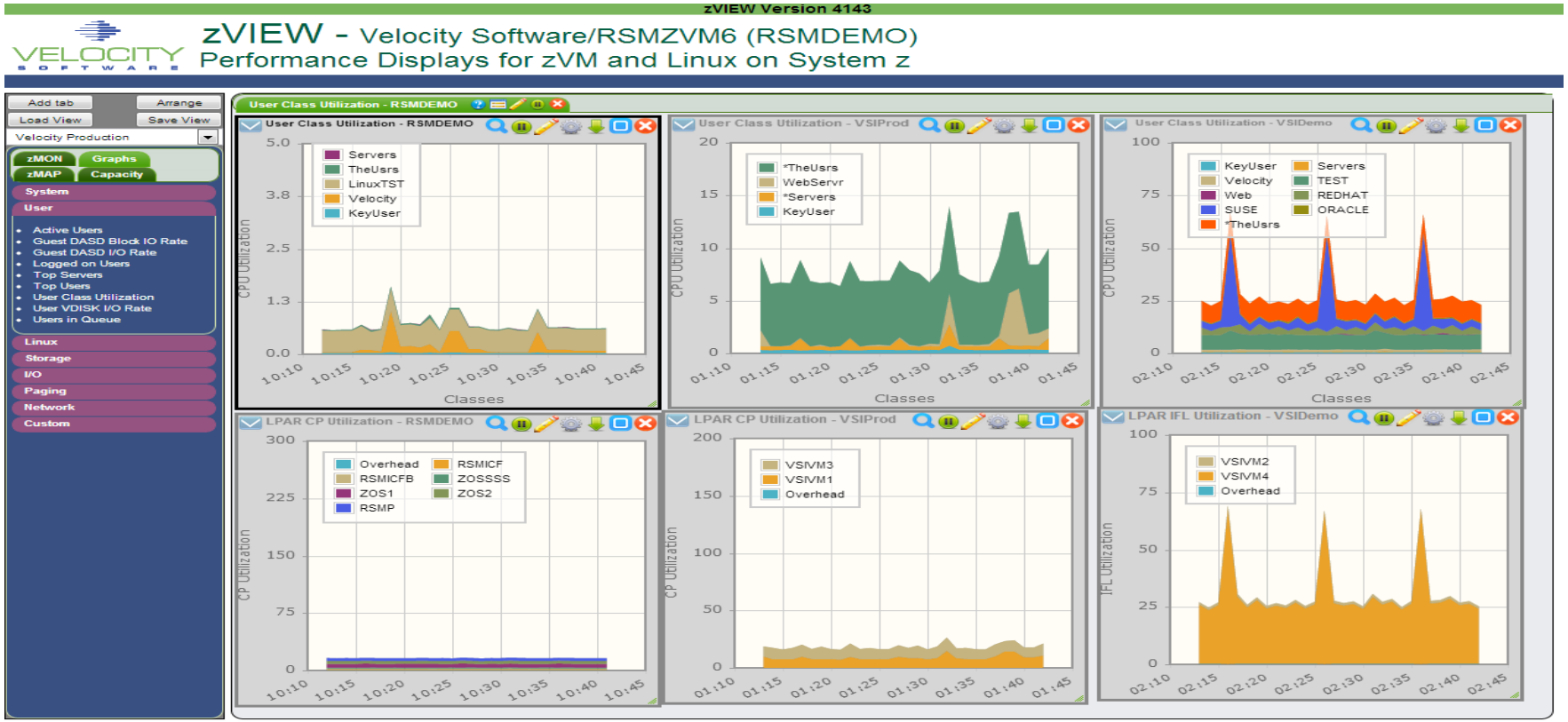
- Linux administrator dashboard provided, everything in one click
- Secure, no need for logon to Linux (no ssh, top)
- Fast and efficient, no restriction on numbers of viewers



Single Pane – Enterprise Wide

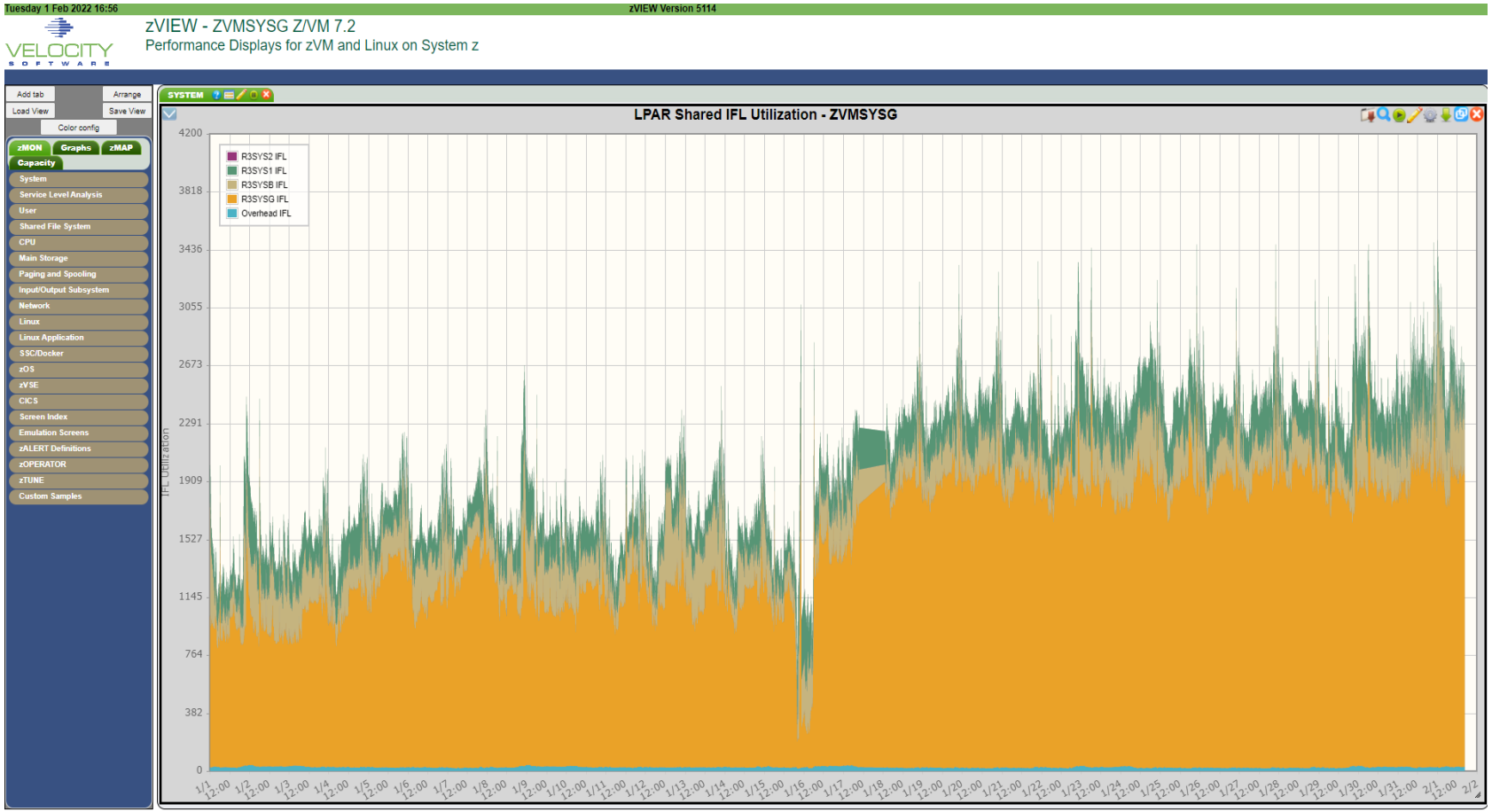
Single pane of glass

- Data from multiple LPARs / geographies
- Menu driven end user designed view



Dynamic Charts

- Data extracted from database dynamically to create graph, example last month



Operational Alerts: zALERT

3270 Style Alerts (50+ sample alerts provided)

```
Screen: LINALERT          Velocity Software          25 Mar 2015 06:42:29
------------------------ Exceptions Analysis Alerts ------------------------

   Type Description
   LNDX / area on oracle is 79.51% full
   LNDX /opt area on oracle is 82.24% full
   LNDX /home area on oracle is 59.02% full
   LNDX / area on RH5X161 is 32.54% full
   LNDX / area on S11R20RA is 81.56% full
   LNDX /boot area on S11R20RA is 2
   LNDX /opt area on S11R20RA is 95
   LNDX /mnt/oracle area on S11R20R
   LNSU Swap utilization for Linux
   LNSU Swap utilization for Linux
```

Today is Wednesday 25 Mar 2015 zVIEW Version 4174

zVIEW - Velocity Software - VSIVM4 (DEMO)

Performance Displays for zVM and Linux on System z

Menu

LINALERT

LINALERT - Exceptions Analysis Alerts - 15/03/25 at 06:47 - DEMO

Code	Alert Description
LNCP	CPU utilization on Linux node BlakeMC is 13.86%
LNDX	/ area on lxsugar is 90.74% full
LNDX	/usr area on lxsugar is 57.59% full
LNDX	/ area on opensuse is 39.71% full
LNDX	/home area on opensuse is 53.23% full
LNDX	/iso/sles11s area on opensuse is 100.00% full
LNDX	/iso/s11sp2- area on opensuse is 100.00% full
LNDX	/iso/s11sp2- area on opensuse is 100.00% full
LNDX	/iso/s11sp3- area on opensuse is 100.00% full
LNDX	/iso/s11sdk- area on opensuse is 100.00% full
LNDX	/iso/s10sp2 area on opensuse is 100.00% full
LNDX	/iso/r64 area on opensuse is 100.00% full
LNDX	/iso/r62 area on opensuse is 100.00% full
LNDX	/iso/s10v1 area on opensuse is 100.00% full
LNDX	/iso/r7 area on opensuse is 100.00% full
LNDX	/iso/sles11s area on opensuse is 100.00% full
LNDX	/iso/s12-1 area on opensuse is 100.00% full
LNDX	/iso/s12-2 area on opensuse is 100.00% full
LNDX	/iso/s12sdk1 area on opensuse is 100.00% full
LNDX	/iso/s12sdk2 area on opensuse is 100.00% full
LNDX	/ area on oracle is 79.51% full
LNDX	/opt area on oracle is 82.24% full
LNDX	/home area on oracle is 59.02% full
LNDX	/ area on redhat5 is 52.26% full
LNDX	/ area on redhat5x is 32.54% full
LNDX	/ area on redhat5G is 95.88% full
LNDX	/mnt area on redhat5G is 53.23% full
LNDX	/ area on redhat6 is 30.60% full
LNDX	/ area on redhat6x is 94.92% full
LNDX	/dev/shm area on redhat6x is 51.42% full
LNDX	/ area on redhat64 is 36.09% full
LNDX	/boot area on rhel7v is 23.79% full
LNDX	/ area on roblnx2 is 78.74% full

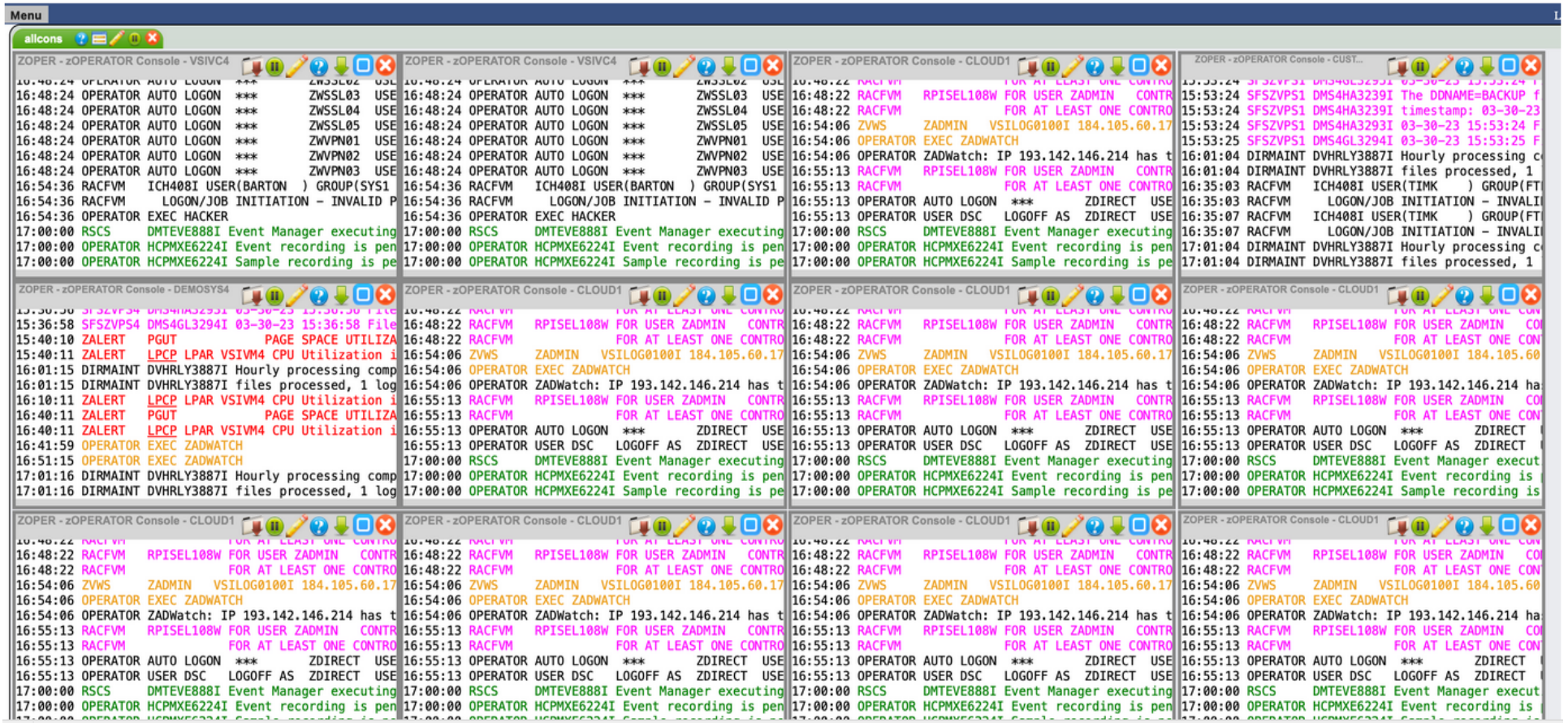
Or Browser based:

Click Thru

or SMS, email…

Operations Console for Enterprise

Single pane of glass – all LPARs console



RHOS Data Collection Technologies

RHOS Servers are “closed”

- Not open to using snmp directly
- No collectd
- Pushes data to Prometheus (is the data correct?)
- CPU data incorrect in smt environment.

Prometheus agents part of openshift (not efficient)

- HTTP interface
- “blob of data” (100k/pod, multiple megabytes per request) (yes, really “blob”)
  - https://github.com/google/cadvisor/blob/master/docs/storage/prometheus.md#prometheus-container-metrics
- Limited metrics
- Significant overhead in parsing every single metric of “blob”

Openshift performance data source

- **Data sources critical to performance management**
- **Snmp installed in a container**
  - **Snmp is very efficient for collecting performance data!!!**
  - Full access to systemwide metrics
  - **Additional metrics** collected to align process data with containers
  - Snmp is extremely efficient in every comparison
  - Snmp container installed on every node

Container Architecture

Containers are not magic

Each container managed by a “container manager”

- “virtual” Processes inside container are mapped to “real”
- File systems are remapped
- Container “limited to what it can see”’

Review standard Linux data (docker)

h

Standard Linux Process Tree for Docker

```
Report: ESALNXC       LINUX Process Configuration
-------------------------------------------------
Node/          <-------Process Ident-------> Appl
 Name            ID       PPID   GRP     Appl Name
------------ ------- ------- ----- ------- --------
21:32:00
DOCKER
 systemd           1       0     1   61176 systemd
  dockerd       1218       1  1218    1218 dockerd
   docker-c     1339    1218  1339    1218 dockerd
    docker-c   32289    1339 32289    1218 dockerd
     httpd     32312   32289 32312   32312 bouncer1
      httpd    32370   32312 32312   32312 bouncer1
      httpd    32371   32312 32312   32312 bouncer1
      httpd    32372   32312 32312   32312 bouncer1
    docker-c   32476    1339 32476    1218 dockerd
     httpd     32498   32476 32498   32498 bouncer2 <--
      httpd    32553   32498 32498   32498 bouncer2
      httpd    32554   32498 32498   32498 bouncer2
      httpd    32555   32498 32498   32498 bouncer2
```

Linux Process Tree

- Shows all processes
- "ppid" is parent process
- "1" is root
- Dockerd is manager
- Docker-c "container"

Result:

- CPU by process
- CPU by container

Capture ratios for Linux processes is 100%

- Docker (sles12)
- Rancher by Suse
- RHOSCP by Redhat
- zCX…. (docker only because of available storage)

Report: ESALNXV LINUX Virtual Processor Analysis Report

Node/ Name	VM ServerID	Node GroupID	<Linux Pct CPU> Total	Syst	User	<Process Data> Total	Syst	User	Capture Ratio
00:15:00									
rancha1	RANCHA1	TheUsrs	11.6	5.0	6.5	11.8	5.0	6.8	0.997
rancha2	RANCHA2	TheUsrs	11.6	4.9	6.6	11.8	4.9	6.9	0.999
ranchs1	RANCHS1	TheUsrs	19.4	6.0	13.4	19.4	6.0	13.4	1.000
rhoscp1	RHOSCP1	OpenShft	57.5	10.1	47.4	57.5	10.1	47.4	0.999
rhoscp2	RHOSCP2	OpenShft	46.8	10.1	36.7	46.8	10.1	36.7	1.000
rhoscp3	RHOSCP3	OpenShft	51.1	9.2	41.9	51.1	9.2	41.9	1.001
SLES12	SLES12	SUSE	24.1	0.8	23.3	24.1	0.8	23.3	1.002

OpenShift Process Table

Containers not magic

Standard Linux data provided for RHOS.

- “conmon” container manager

```
Report: ESALNXC       LINUX Process
Monitor initialized: 03/08/23 at 0
-------------------------------------
Node/          <-------Process Ident
 Name            ID      PPID   GRP
------------ ------- ------- -----
rhoscp1
 systemd           1       0     1
  systemd-       919       1   919
  systemd-       963       1   963
  auditd        1060       1  1060
  dbus-dae      1097       1  1097
  crio          1929       1  1929
  kubelet       1970       1  1970
  conmon        2387       1  2387
   kube-sch     2442    2387  2442
  conmon        2520       1  2520
   cluster-     2559    2520  2559
  conmon        2582       1  2582
   cluster-     2600    2582  2600
  conmon        2677       1  2677
   cluster-     2703    2677  2703
  conmon        8396       1  8396
   promethe     8455    8396  8455
  conmon       10850       1 10850
   grafana-    11073   10850 11073
  conmon      1509339       1  2011
   snmpd      1509401 1509339  2073
```

CPU by Process

By CPU Consumption by process.

- Just the active processes

```
Report: ESALNXP      LINUX HOST Process Statistics Report
-------------------------------------------------------------
node/     <Process Ident> Nice PRTY <------CPU Percents---->
 Name           ID    PPID Valu Valu  Tot  sys user syst usrt
--------- ------- ------- ---- ---- ---- ---- ---- ---- ----
07:02:00
rhoscp1          0       0    0    0  139 19.9  113 3.74 2.87 → node totals
 systemd         1       0    0   20 1.62 0.57 1.05    0    0
 crio         1929       1    0   20 6.48 0.15 0.58 3.24 2.51
 kubelet      1970       1    0   20 10.5 3.06 7.47    0    0
 etcd         3078    3061    0   20 15.4 5.91  9.5    0    0
 etcd         3131    3113    0   20 1.67 0.87 0.80    0    0
 cluster-     3959    3944    0   20 2.49 0.23 2.25    0    0
 cluster-     5072    4989    0   20 2.02 0.35 1.67    0    0
 promethe     8455    8396    0   20 64.1 3.47 60.6    0    0
 openshif    11221   11167    0   20 2.72 0.32 2.41    0    0
 kube-api 1245253 1245136    0   20 16.7 1.77 15.0    0    0
 oauth-ap 1830274 1830210    0   20 3.29 0.25 3.04    0    0
rhoscp2          0       0    0    0 84.2 18.5 59.4 3.88 2.52  → node totals
 systemd         1       0    0   20 2.15 0.67 1.18 0.22 0.08
 crio         2056       1    0   20 5.85 0.18 0.58 3.10 1.98
 kubelet      2091       1    0   20 12.5 3.72 8.81    0    0
 etcd         3185    3169    0   20 14.3 5.33 9.00    0    0
 etcd         3230    3210    0   20 1.35 0.68 0.67    0    0
 cluster-     3277    3258    0   20 2.50 0.23 2.27    0    0
 cluster-     9096    9068    0   20 1.47 0.22 1.25    0    0
 cluster-    10370   10281    0   20 1.02 0.20 0.82    0    0
 openshif    10951   10905    0   20 2.22 0.22 2.00    0    0
 cluster-    11973   11886    0   20 1.50 0.23 1.27    0    0
```

CPU realtime by Process

By CPU Consumption For “ NODE RHOS*”, sorted by cpu

Screen: ESALNXP Velocity Software - VSIVM4 ESAMON 5.134 03/03 :44-22:45
<--1 of 4 VSI Linux Percent Usage by Process **NODE RHOSCP***

			-Process Ident--->			<-----CPU Percents----->				
Time	Node	Name	ID	PPID	GRP	**Tot**	sys	user	syst	usrt
22:45:00	rhoscp1	*Totals*	0	0	0	84.5	12.1	67.5	2.9	2.1
	rhoscp3	*Totals*	0	0	0	73.2	9.4	60.7	1.8	1.3
	rhoscp2	*Totals*	0	0	0	53.7	10.4	33.7	2.2	7.5
	rhoscp1	**prometheus**	8455	8396	8455	**38.6**	1.6	37.0	0	0
	rhoscp3	prometheus	7807	7735	7807	**38.4**	1.5	36.9	0	0
	rhoscp1	etcd	3078	3061	3078	10.1	3.9	6.1	0	0
	rhoscp2	kube-apiserv	1464486	1464276	0	8.4	0.9	7.5	0	0
	rhoscp3	etcd	4129	4099	4129	8.2	3.1	5.1	0	0
	rhoscp1	kube-apiserv	1245253	1245136	65488	7.9	0.8	7.1	0	0
	rhoscp2	etcd	3185	3169	3185	7.7	2.9	4.8	0	0
		systemd	1	0	1	7.3	0.4	0.7	0.2	6.0
		kubelet	2091	1	2091	7.3	2.3	5.1	0	0
	rhoscp1	kubelet	1970	1	1970	6.9	2.0	4.9	0	0
	rhoscp3	kube-apiserv	1609641	1609516	36652	6.7	0.7	6.0	0	0
		kubelet	1880	1	1880	5.3	1.6	3.7	0	0
	rhoscp1	crio	1929	1	1929	4.7	0.1	0.3	2.5	1.8
	rhoscp2	crio	2056	1	2056	3.2	0.1	0.3	1.6	1.2
	rhoscp3	crio	1839	1	1839	2.8	0.1	0.2	1.5	1.1
	rhoscp1	oauth-apiser	1830274	1830210	60802	2.1	0.1	2.0	0	0
	rhoscp2	oauth-apiser	694375	694352	0	1.9	0.1	1.8	0	0
	rhoscp3	oauth-apiser	634417	634398	44593	1.9	0.1	1.8	0	0
	rhoscp1	openshift-ap	11221	11167	11221	1.8	0.2	1.6	0	0
		cluster-etcd	3959	3944	3959	1.6	0.2	1.5	0	0
	rhoscp2	openshift-ap	10951	10905	10951	1.6	0.1	1.5	0	0
		cluster-etcd	3277	3258	3277	1.5	0.1	1.3	0	0

CPU by Component/Function

Other components?

- Node “groups” are defined groups of associated servers
- Totals for all RHOS “OpenShif” servers **(Linux perspective)**
- CPU time as measured by Linux – correct???

Report: ESALNXA LINUX HOST Application Report

Node/ Date Time	Process/ Application name	ID	<---Processor Percent---> Total	<Process> sys	<Process> user	<Children> syst	<Children> usrt
11:25:00							
OpenShif	*Totals*	0	**351.0**	53.4	278	11.8	7.9
	conmon	0	293.1	39.9	251	1.5	1.1
	crio	0	18.68	0.4	1.3	10.3	6.8
	kernel	0	1.37	1.4	0	0	0
	kubelet	0	31.91	9.5	22.4	0.0	0.0
	ovs-vswi	0	1.27	0.7	0.6	0	0
	ovsdb-se	0	0.15	0.0	0.1	0	0
	systemd	0	4.50	1.5	3.0	0	0

Configuration: RHOS

Container Configuration RHOS much larger (snmp in container)…

```
Report: ESAK8S1       Kubernetes Configuration Report                Velocity Sof
Monitor initialized: 06/22/23 at 00:00:00 on 8562 serial 040F78     First record
-------------------------------------------------------------------------------
Linux     <---OpenShift Pod Configuartion-->  <----------Container Configuration
Node/                                                          <--Process Ide
Time      PodName                 PodIndex      Name                 ProcessID Pro
--------  ----------------------  ------------  -------------------- --------- ---
00:15:00
rhoscp1
          insights-operator-7f    ba92ef4e1b29  insights-operator         13075 ins
          multus-admission-con    bbb779ebae39  kube-rbac-proxy          14520 kub
                                                multus-admission-con     12865 web
          etcd-rhoscp1.vsi1.ve    c6088570034c  etcd                      2276 etc
                                                etcd-metrics              2389 etc
                                                etcd-readyz               2941 clu
                                                etcd-health-monitor       3594 clu
          prometheus-operator-    c7875e16a183  prometheus-operator-     11955 pro
          kube-state-metrics-5    d5e9800a6a6c  kube-state-metrics       11658 kub
                                                kube-rbac-proxy-main     12519 kub
                                                kube-rbac-proxy-self     13456 kub
          prometheus-k8s-1        45f9f5becfa0  prometheus               14548 pro
                                                thanos-sidecar           15191 tha
                                                prometheus-proxy         15372 oau
                                                kube-rbac-proxy-than     15931 kub
                                                kube-rbac-proxy          15508 kub
                                                config-reloader          14820 pro
          packageserver-5f99c6    662518f1bc49  packageserver            13044 pac
          vsi-snmpd-vk5vd         7e583397ff6e  vsi-snmpd                19285 snm
          multus-9fj4w            8e5c35c1b612  kube-multus               4922 /en
```

CPU by container (rhos)

By CPU Consumption by container for RHOS

```
Report: ESAK8S2      Kubernetes Resource Utilization Report
-------------------------------------------------------------------------
NODE/                          <---Container-->  <--Container CPU------>
Time/ PodName                  <--Process ID-->  <------CPU Percents---->
Date   ContainerName            ProcID ProcName   Tot  sys user syst usrt
----------------------         ------ --------  ---- ---- ---- ---- ----
00:15:00
rhoscp1
   etcd-rhoscp1.vsi1.ve
    etcd                         2276 etcd      7.94 2.81 5.13    0    0
    etcd-metrics                 2389 etcd      1.31 0.69 0.62    0    0
    etcd-readyz                  2941 cluster-  0.79 0.07 0.72    0    0
    etcd-health-monitor          3594 cluster-  1.33 0.11 1.22    0    0
   kube-controller-mana
    kube-controller-mana         8999 cluster-  0.61 0.11 0.49    0    0
   apiserver-d84c8f947-
    oauth-apiserver             24092 oauth-ap  1.81 0.08 1.72    0    0
   apiserver-5d795f8cd7
    openshift-apiserver         12688 openshif  1.79 0.13 1.65    0    0
   prometheus-k8s-1
    prometheus                  14548 promethe  13.4 0.45 12.9    0    0
   authentication-opera
    authentication-opera        13103 authenti  1.01 0.16 0.85    0    0
   packageserver-5f99c6
    packageserver               13044 package-  0.95 0.12 0.83    0    0
   kube-apiserver-rhosc
    kube-apiserver             574916 watch-te  10.3 0.79 9.48    0    0
   kube-controller-mana
    kube-controller-mana        22941 kube-con  0.78 0.21 0.57    0    0
```

Correct Data ?

Bank (very large bank) complains:

- Chargeback model is over charging when using SMT
- Using IBM monitor metrics
- All Linux numbers are “thread time” not “cpu time”

IBM CP Monitor provides 2 metrics

- Thread time (traditional measurement)
- SMT CPU prorated time (IBM’s estimated, incorrect)

If chargeback or capacity planning is important?

- Valid numbers would be appreciated?

CPU by Component/Function

Some even “better news”

- CPU numbers are traditional, measured by Linux
- **VSI Prorated** based on HMC data
  - Shows SMT is significantly better

```
Report: ESAUSP5       User SMT CPU Consumption Analysis
-------------------------------------------------------------------------
          <------CPU Percent Consumed   (Total)---->  <-TOTAL CPU-->
UserID    <Traditional> <MT-Equivalent> <IBM Prorate> <VSI Prorated>
/Class    Total  Virt   Total  Virtual  Total Virtual Total  Virtual
-------- ----- -----   -----  -------  ----- ------- -----  --------
07:02:00 414.9  408.0   322.7    317.3  239.7   235.8  208.2    204.7
 ***User Class Analysis***
OpenShif 355.0  350.3   276.0    272.3  204.9   202.2  178.1    175.7
 ***Top User Analysis***
RHOSCP1  142.4  140.8   110.1    108.9  82.93   82.01  71.43    70.65
RHOSCP3  125.2  123.8   97.38    96.34  72.35   71.60  62.80    62.14
RHOSCP2  86.79  85.04   68.00    66.64  49.31   48.30  43.55    42.67
```

Chargeback/capacity planning

Resource consumption model

- Correct the CPU first from SMT
- Aggregate containers into pods
- Aggregate pods by name
- Naming convention allows consumption model

What Else do we know?

RHOS Case study:

- Production environment
- 9 nodes, snmp enabled

```
Linux Release: RHELCoreOS 412.86.202310210217-0
   Linux s01vx9986726 4.18.0-372.76.1.el8_6.s390x
```

CPU by container and POD

Production (Bank) Container CPU Report, Room for tuning?

```
Report: ESAK8S2      Kubernetes Resource Utilization Report
Monitor initialized: 11/06/23 at 17:00:00 on 3931
-------------------------------------------------------------------------
NODE/                          <---Container-->  <--Container CPU------->
Time/ PodName                  <--Process ID-->  <------CPU Percents---->
Date   ContainerName            ProcID ProcName   Tot  sys user syst usrt
-----------------------  ------ --------  ---- ---- ---- ---- ----
RHOSDI1                         (Totals)         20.0 1.15 12.9 3.31 2.63
   s01vx9986726                                  3.75 0.01 0.01 1.93 1.80
    gpfs                         13524 sh        3.62    0    0 1.86 1.76
    logs                         13587 sh        0.12 0.01 0.01 0.07 0.03
   vsi-snmpd-qkhl7                               0.38 0.21 0.17    0    0
    vsi-snmpd                     4386 snmpd      0.38 0.21 0.17    0    0
   sdn-2l2xl                                     1.76 0.04 0.08 1.08 0.57
    sdn                          1997 openshif   1.76 0.04 0.08 1.08 0.57
   prometheus-k8s-1                              11.0 0.54 10.5    0    0
    prometheus                   6170 promethe   10.2 0.45  9.7    0    0
    thanos-sidecar               6632 thanos     0.58 0.08 0.50    0    0
    prometheus-proxy             6912 oauth-pr   0.31 0.01 0.30    0    0
   alertmanager-main-1                           0.16 0.01 0.14    0    0
    alertmanager-proxy           5850 oauth-pr   0.16 0.01 0.14    0    0
   node-exporter-88d8x                           0.17 0.04 0.12    0    0
    node-exporter                2402 node_exp   0.17 0.04 0.12    0    0
   ibm-spectrum-scale-p                             0    0    0    0    0
    sysmon                       7396 runConta      0    0    0    0    0
   prometheus-adapter-7                          0.38 0.04 0.33    0    0
    prometheus-adapter           5869 adapter    0.38 0.04 0.33    0    0
   ibm-spectrum-scale-g                             0    0    0    0    0
    liberty                      7171 sh            0    0    0    0    0
```

What Else do we know?

Case study:

Production environment

Storage over configured

- ESAUCD2 – 400Gb defined, 300Gb free.
- CMM works. (cmm container of course)
- Can manage

```
Report: ESAUCD2
-----------------------------
Node/     <-------------------
Time/     <--Real Storage-->
Date      Total   Avail Used
--------- ------ ----- -----
17:01:00
***Node Groups***
COREOS    410936  303K   98K
*** Nodes *******
RHOSDI1    24156 14201  9955
RHOSDI2    24156 13141 11015
RHOSDM1    24156 10483 13673
RHOSDM2    24156  8964 15192
RHOSDM3    24156 10811 13345
RHOSDWB    96731 90768  5964
RHOSDW1    64475 53776 10699
RHOSDW2    64475 52806 11670
RHOSDW3    64475 55161  9315
```

What Else do we know?

Case study:

Production environment

VCPUs overconfigured

- 6 VCPU
- 1 needed
- Extra overhead
- Pollutes cache

Corrected with zVRM

- Vary cpus offline

```
Report: ESALNXS LINUX VSI System
--------------------------------
Node/       CPU <Processor Pct Util>
Time        NBR Total Syst User Idle
--------  --- ----- ---- ---- ----
17:01:00
LIMEDI1     Tot  45.0 11.1 32.4  552
              1   8.1  2.0  5.6 91.3
              2   7.4  1.8  5.3 92.3
              3   6.6  1.7  4.7 93.0
              4   7.8  1.9  5.6 91.8
              5   9.4  1.9  7.3 90.1
              6   5.7  1.7  3.8 93.9
LIMEDI2     Tot  75.0 25.2 47.8  523
              1  12.7  4.1  8.0 86.8
              2  11.9  4.6  7.0 87.6
              3  12.2  4.1  7.8 87.5
              4  12.5  3.9  8.4 87.1
              5  14.0  4.4  9.3 85.6
              6  11.7  4.1  7.3 88.0
```

zVRM: Automate Server Sizing

What we have learned? Users want very large servers…

Server modification "happens", applications grow

- More CPU, RAM needed and must be added, requiring outage

Why Excessively large servers?

- That's they way they do it on Intel / VMWare

zVRM, Velocity Resource Manager automates management

- Centralized management facility for managing server resources
- CMM to reduce over sized storage when not needed
- CMM to return storage as workload increases
- Vary vcpu on/offline to meet demand
- Allows definitions of oversized servers to operate efficiently

Requires zPRO APIs, zVPS for data input and feedback

zCX on z/OS?

Snmp works with containerized snmp

- **Standard network data**

CPU analysis

- **Linux Process data**
- **Linux System data**
- **Docker / RHOS Container / pod**
- **SMF records System (70), Job (30)**

Storage / ram / Swap

- **System**
- **Process**
- **Container / pod**

zCX on z/OS?

Install zCX with docker

- “https://velocitysoftware.com/zcximpl.html”

Install snmp container

- And Yes, it works, and what is it?
- (Linux 5.4.0, Ubuntu Distribution for 390)

```
SNMP Server Configuration:
  Description:
    Linux 7f1752ffc4e3 5.4.0-146-generic
    #163-Ubuntu SMP Fri Mar 17 18:32:31 UTC 2023 s390x
  ObjectId: 01.03.06.01.04.01.BF.08.03.02.0A
```

Next Step: z/OS zCX

zCX, Openshift run on zip in z/OS address space

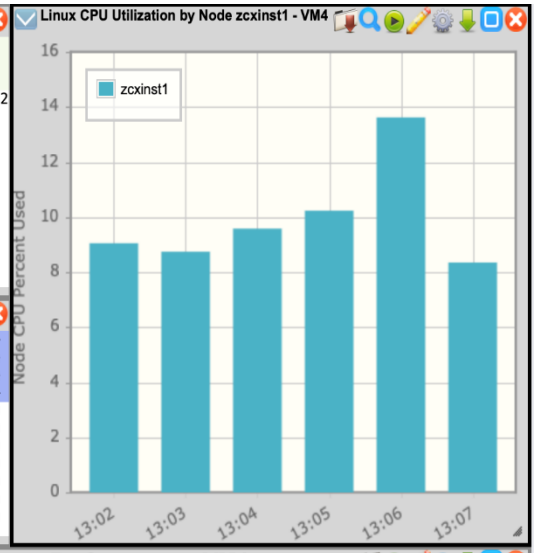
Linux dashboard: Looks like any other Linux server to us

ESALNXP - VSI Linux Percent Usage by Process - VM4

Time	Node	Name	ID	PPID	GRP	Tot	sys	user	syst	usrt	nice	prty	valu	valu	Size	RSS	Peak	Swap	Data	Stk	EXEC	Lib	Lck	PTbl	min	maj
13:08:00	zcxinst1	*Totals*	0	0	0	10.7	6.2	2.7	1.5	0.3	0	0	0	24K	714	25K	306	2834	12.7	169	1981	336	11.6	17	0	2
13:08:00	zcxinst1	systemd	1	0	1	0.1	0.0	0.0	0	0	0	20	164	7.4	226	0.7	20.5	0.1	1.3	13.7	0	0.1	0	0		
13:08:00	zcxinst1	ksoftirqd/0	10	2	0	0.1	0.1	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0		
13:08:00	zcxinst1	rcu_sched	11	2	0	0.1	0.1	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0		
13:08:00	zcxinst1	ksoftirqd/1	16	2	0	0.1	0.1	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0		
13:08:00	zcxinst1	dmcrypt_writ	603	2	0	0.0	0.0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0		
13:08:00	zcxinst1	btrfs-transa	682	2	0	0.0	0.0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0		
13:08:00	zcxinst1	systemd-jour	903	1	903	0.1	0.1	0.0	0	0	-1	19	89.1	29.7	111	0.1	18.7	0.1	0.1	13.6	0	0.2	2	0		
13:08:00	zcxinst1	multipathd	1029	1	1029	0.1	0.1	0.0	0	0	0	-100	336	16.1	400	0	10.0	0.1	0.1	6.4	336	0.1	0	0		
13:08:00	zcxinst1	zcxauthplugi	1083	1	1083	0.1	0.1	0.0	0	0	0	20	469	5.8	469	0.2	142	0.1	2.7	1.9	0	0.1	0	0		
13:08:00	zcxinst1	containerd	1085	1	1085	1.1	0.7	0.4	0	0	0	20	1453	19.4	1517	3.4	134	0.1	25.6	1.9	0	0.2	0	0		

ESALNXR - Linux Memory Analysis - VM4

Time	Node	Total	Free	Size	Activ	Swap	Total	Activ	Inact	Size	Size	Reclm	Total	Dirty	WrtBk	Total	Used	Chunk	table	Total	Free	Rsvd	Size
13:08:00	zcxinst1	1407	139.7	459.3	237.6	21.7	271.0	128.1	163.2	14.6	257.9	99.0	7.7	0.1	0	126.0	0.1	0	16.9	0	0	0	0
13:07:00	zcxinst1	1407	139.8	458.9	237.0	21.7	271.0	128.1	163.2	14.7	257.8	99.0	7.7	0.1	0	126.0	0.1	0	16.9	0	0	0	0
13:06:00	zcxinst1	1407	140.2	458.5	236.1	21.7	271.0	128.1	163.2	14.7	258.0	99.0	7.6	0.3	0	126.0	0.1	0	16.9	0	0	0	0
13:05:00	zcxinst1	1407	132.2	457.7	235.0	21.7	280.6	137.7	163.2	14.7	258.1	99.0	7.6	0.1	0	126.0	0.1	0	17.2	0	0	0	0
13:04:00	zcxinst1	1407	141.5	457.4	234.6	21.7	270.9	128.1	163.2	14.7	258.1	98.9	7.5	0.2	0	126.0	0.1	0	17.0	0	0	0	0
13:03:00	zcxinst1	1407	142.0	457.2	234.2	21.7	271.0	128.0	163.2	14.7	258.0	98.9	7.5	0.0	0	126.0	0.1	0	16.9	0	0	0	0
13:02:00	zcxinst1	1407	142.0	456.9	233.6	21.7	271.0	128.1	163.2	14.6	258.0	98.9	7.5	0.2	0	126.0	0.1	0	16.9	0	0	0	0



ESALNXV - LINUX Virtual Processor Analysis Repo - VM4

Time	Node	VM ServerID	Node GroupID	Linux Pct Total	CPU Syst	CPU User	Process Total	Data Syst	Data User	Capture Ratio	Prorate Factor	NVcpu	LPAR Name
13:08:00	zcxinst1	INST1	TheUsrs	10.7	7.7	3.0	10.7	7.7	3.0	1.000	1.000	2	ZOS25A
13:07:00	zcxinst1	INST1	TheUsrs	8.4	6.0	2.3	8.4	6.0	2.3	1.000	1.000	2	ZOS25A
13:06:00	zcxinst1	INST1	TheUsrs	13.6	9.3	4.4	13.6	9.3	4.4	1.000	1.000	2	ZOS25A
13:05:00	zcxinst1	INST1	TheUsrs	10.2	7.5	2.7	10.2	7.5	2.7	1.000	1.000	2	ZOS25A
13:04:00	zcxinst1	INST1	TheUsrs	9.6	6.8	2.8	9.6	6.8	2.8	1.000	1.000	2	ZOS25A
13:03:00	zcxinst1	INST1	TheUsrs	8.7	6.3	2.5	8.7	6.3	2.5	1.000	1.000	2	ZOS25A
13:02:00	zcxinst1	INST1	TheUsrs	9.1	6.5	2.5	9.0	6.5	2.4	0.991	1.000	2	ZOS25A

ESALNXF - VSI Disk File System Performance - VM4

Time	Node	Name	Type	Read I/O /Second	Read I/O Sctrs	Read I/O ms	Write I/O /Second	Write I/O Sctrs	Write I/O ms	I/O Mrgd	I/O /RdIO	I/O /IO	I/O Mrgd	I/O /WrIO	I/O /IO	Time(ms) Prog	Time(ms) IOQ	Time(ms) I/O	Device Path
13:08:00	zcxinst1	dm-0	lvm	0	0	0	0	3.3	0	23.0	9.34	0	2.40	9.34	dm-0				
13:08:00	zcxinst1	dm-1	lvm	0	0	0	0	3.3	0	23.0	40.7	0	2.65	40.7	dm-1				
13:08:00	zcxinst1	dm-2	lvm	0.0	0	32.0	0	0.5	0	88.0	5.16	0	3.13	5.00	dm-2				
13:08:00	zcxinst1	vda	disk	0	0	0	0	3.4	0.7	23.9	8.40	0	2.98	3.90	ccw-0.0.0001				
13:08:00	zcxinst1	vda1	part	0	0	0	0	0.4	0.4	15.4	2.77	0	5.00	0.38	ccw-0.0.0001-part1				
13:08:00	zcxinst1	vda2	part	0	0	0	0	3.0	0.3	25.2	9.21	0	2.68	4.41	ccw-0.0.0001-part2				
13:08:00	zcxinst1	vdc	disk	0.0	0	32.0	1.00	0.3	0.2	144	4.68	0	5.00	0.50	ccw-0.0.0004				
13:08:00	zcxinst1	vdc1	part	0.0	0	32.0	1.00	0.3	0.2	144	4.68	0	5.00	0.50	ccw-0.0.0004-part1				
13:07:00	zcxinst1	dm-0	lvm	0	0	0	0	2.9	0	16.6	17.4	0	4.38	17.4	dm-0				

Docker containers (configuration, CPU)

- Note docker assigns names if not provided

```
Report: ESADOCK1      DOCKER Configuration Report
-------------------------------------------------------------------
Time /    <-----------Container Configuration----------> Status
Node      ContainerName     ImageName  Index          Procid
--------  ----------------- ---------- -------------- ------ ------
15:14:00
zcxinst1
          clever_ramanujan  localhost/ 7f1752ffc4e3    5752 runn
          httpd1            httpd      2c9d7574ca87    6032 runn
          httpd2            httpd      1fca2a98a85e    5856 runn
          stress2           stresscpu  1c35d9534623    5518 runn
          stress1           stresscpu  aa927ba72ee6    5655 runn
          ibm_zcx_zos_ssh_  ibm_zcx_zo a3cfd896b4d7    3436 runn
```

zCX z/OS Data

From SMF 30 (job records), only one user of ziip

- Why is “zip on CP”, and why “GP CPU percent”?
- OpenShift is not “free” in terms of GP requirements

z/OS Job/Step CPU/Resources

SYSID	<------JOB---------->			<--CPU Percents-				<---ZIP Pct--->			
	Name	JobID	Step Nbr	Total	STD	SRB	I T	Tot	Enc	Dep	on CP
15:13:00 - 15:14:00											
V25A	Totals		.	8.92	4.45	3.95		59	0	0.2	1.6
	INST1	STC05196	1	2.88	1.43	1.10		59	0	0.1	1.4
	ZOSMNVM2	STC04764	1	0.02	0.02	0		0	0	0	0
	ZOSMNVM4	STC04762	1	0.02	0.02	0		0	0	0	0

zCX Performance Case Study

zCX has swap defined

- Linux Uses swap when short on storage
- When performance gets bad, is it swapping to disk?
- CMM is NOT an option for zCX
- IBM: Add storage to eliminate swap?
- What is zCX plan for alerts for swapping?
- Note that we get swap space per process, so look at process table

```
Report: ESAUCD2       LINUX UCD Memory Analysis Report
Monitor initialized: 10/27/23
-------------------------------------------------------
Node/    <----------------------Storage Sizes (MegaBytes)
Time/    <--Real Storage--> <-----SWAP Storage----> Total
Date     Total  Avail Used  Total Avail Used  MIN   Avail
-------- ------ ----- ----- ----- ----- ----- ----- -----
15:14:00
*** Nodes *******
zcxinst1 1407.0 312.5  1095  1980  1697 283.3  15.6  2009
```

zCX Performance Case Study

If there is a “bug” or other problem?

- “reboot”???
- Add resources and reboot to hide the problem (for a while)?
- Understand and fix the problem? (30 “uwsgi-core” processes sleeping)
- 81 zombie processes….

```
Report: ESALNXP      LINUXcess Statistics Report          Velocity
Softw
-------------------------------------------------------------------
-
node/      <Process Ident>  <------CPU Percents----> <--------------
Stor
 Name            ID    PPID   Tot  sys user syst usrt  Size RSS Peak Swap
--------- ------- ------- ---- ---- ---- ---- ---- ---- --- ---- ----
ZOSLP1           0       0 49.4 0.20 0.21 0.08 48.9  17K 332  17K  573
 uwsgi-co     1177    1121 0.01    0 0.01    0    0   79   4 79.1 20.9
 uwsgi-co     1178    1121    0    0    0    0    0   80   6 79.7 19.2
 uwsgi-co     1179    1121    0    0    0    0    0   78   4 78.2 19.3
 uwsgi-co     1180    1121    0    0    0    0    0   81   6 80.8 20.5
 uwsgi-co     1181    1121 0.01    0 0.01    0    0   79   4 79.5 20.8
 uwsgi-co     1182    1121    0    0    0    0    0   79   4 79.3 21.1
 uwsgi-co     1183    1121    0    0    0    0    0   78   5 78.5 19.1
 uwsgi-co     1184    1121    0    0    0    0    0   79   4 78.7 20.4
 uwsgi-co     1185    1121    0    0    0    0    0   79   4 79.5 21.2
 uwsgi-co     1186    1121    0    0    0    0    0   79   4 79.7 21.2
 uwsgi-co     1187    1121    0    0    0    0    0   79   5 79.4 19.9
 uwsgi-co     1188    1121 0.01    0 0.01    0    0   81   7 82.6 18.9
```

Velocity Software's Leadership Role

z/VM Performance Management since 1988 (first for VM/XA)

Linux Performance Management (Since 2001 – First on "mainframe")

- Lightweight, one click dashboards
- Full dashboard for all data (zVIEW)
- Provide data (VSIPump) to other dashboards
  - (Grafana, splunk, etc)
  - Linux, Oracle, MongoDB, Postgres, OpenShift
  - Secure container platform
  - Note: IBM "datapump" Limited to z/VM data only
- Manage performance (zVRM – Velocity Resource Manager)
  - Tailor Linux servers to meet current workload requirements
- Produce health reports (zTUNE)
- Alert management (zAlert, zOperator)

Summary

OpenShift / Kubernetes, Docker has the data for:

- Performance Analysis
- Operational Alerts (swapping)
- Capacity Planning
- Chargeback (accurate cpu?)

Data collection with snmp

- Inexpensive
- Validated
- Measurable by container
- z/OS data needs effective prorate technology

Black boxes will be problematic, RHOS is no longer a black box

Thank you.