Managing z/VM, Linux and Openshift

Barton Robinson, CTO
Velocity Software, Inc.
Barton@velocitySoftware.com

Who is Velocity Software

Founded in 1988 in Mountain View California.

Constant Delivery since 1988, with a focus on these key areas:

- ❑ Performance Management Suite: **zVPS** a single pane of glass for:
  - z/VM
  - Linux
  - Network
  - zVSEMON
  - zOSMON
  - Applications (Oracle, Docker, MongoDB, **OpenShift**)
- ❑ Modernizing z/VM with: **zPRO**
- ❑ Performance Support and Tuning with: **zTUNE** and **zVRM**
- ❑ **(Ansible support will be announced shortly)**

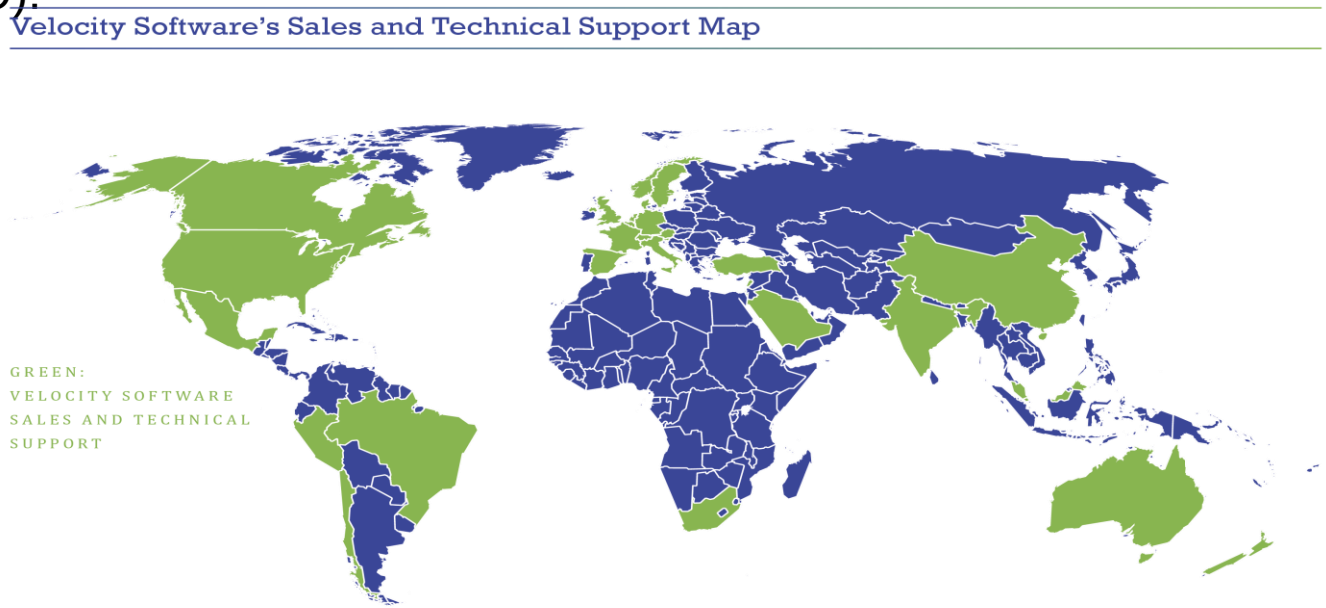
Who is Velocity Software

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", Wells, ATPCO, UHG, BCBSSC utilize zVPS, "large bank", Wells, ATPCO utilizes zPRO).



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.

zLOV
LINUX ON VELOCITY

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.

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

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 an accurate financial model for resource billing
- Who is consuming the resource?

Performance management can NOT be the performance problem

Black boxes are not managed

Numbers must be reliable

VELOCITY SOFTWARE

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)

Linux Performance data source

Data sources critical to performance management

Snmp is very lightweight

- **Used by Velocity Software for Linux, network, applications**
- **Snmp is very efficient for collecting performance data!!!**
- **Containerized** for Kubernetes architectures
- 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

Linux Data Collection every 60 seconds for ALL nodes

- Overhead – target is .1% of one CPU per node
- Full process table

Container Architecture

Containers are not magic – and not a black box with zVPS

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”’

OpenShift Process Table

Openshift is not rocket science

Standard Linux data provided for RHOS.

- “conmon” **container manager (six containers in sample)**

```
Report: ESALNXC       LINUX Process
-----------------------------------
Node/          <-------Process Ident
 Name            ID       PPID   GRP
------------ ------- ------- -----
rhoscp1
 systemd           1       0     1
  systemd-       919       1   919
  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        8396       1  8396   ←
   promethe     8455    8396  8455
  conmon       10850       1 10850   ←
   grafana-    11073   10850 11073
  conmon      1509339       1  2011   ←  (Velocity Software)
   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
```

VELOCITY SOFTWARE

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/     Process/     ID      <---Processor Percent--->
Date      Application                <Process><Children>
Time      name                 Total sys  user syst 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
```

VELOCITY SOFTWARE

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 "better news" from z/VM based measurements

- CPU numbers are traditional, measured by Linux (Thread time)
- **Virtual Machine** with SMT "Prorate" are lower
- IBM SMT numbers do not match reality…. (Same in zCX)
- https://VelocitySoftware.com/smt.html for understanding

```
Report: ESAUSP5        User SMT CPU Consumption Analys
-------------------------------------------------------
            <------CPU Percent Consumed    (Total)---->
UserID      <Traditional> <MT-Equivalent> <IBM Prorate>
/Class      Total  Virt    Total  Virtual  Total Virtual
--------- ------ ------   ------ -------- ------ -------
07:02:00  414.9   408.0   322.7    317.3   239.7   235.8

  ***User Class Analysis***
OpenShif  355.0   350.3   276.0    272.3   204.9   202.2

  ***Top User Analysis***
RHOSCP1   142.4   140.8   110.1    108.9   82.93   82.01
RHOSCP3   125.2   123.8   97.38    96.34   72.35   71.60
RHOSCP2   86.79   85.04   68.00    66.64   49.31   48.30
```

VELOCITY SOFTWARE

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
```

CPU by Component/Function

Production environment (Bank) – large discrepancy

```
Report: ESAUSP5       Consumption Analysis
-------------------------------------------------
         <-CPU PERCENTS (Total)----> <-TOTAL CPU-->
UserID   <Traditional> <IBM Prorate> <VSI Prorated>
/Class   Total  Virt   Total Virtual Total  Virtual
-------- ----- -----   ----- ------- ----- --------
11/06/23
17:01:00 430.6  425.0  271.0   267.6  216.7   213.9
***User Class Analysi
Velocity  0.58   0.55   0.35    0.33   0.29    0.28
TheUsrs   3.53   3.53   3.12    3.12   1.78    1.78
RHEL      4.55   4.31   2.85    2.71   2.29    2.17
COREOS   421.8  416.5  264.6   261.4  212.3   209.6
 ***Top User Analysis*
RHOSDW2  194.6  193.1  122.5   121.6  97.92   97.18
RHOSDW1  116.0  114.9  73.77   73.05  58.39   57.80
RHOSDM1  61.53  59.64  38.21   37.05  30.96   30.01
```

Chargeback/capacity planning

Resource consumption model

- Correct the CPU (by server)first from SMT
- Prorate the process data
- Aggregate process data into containers
- Aggregate containers into pods
- Aggregate pods by name
- (Naming convention allows consumption model)

Added metrics for Kubernetes

Kubernetes (openshift, rancher) Container Configuration includes

(Using Rancher for simple perspective)

- Container index, name,
- Pod index, name
- ProcessID – to connect process data to container

```
Report: ESAK8S1       Kubernetes Configuration Report
-------------------------------------------------------------------------------
Linux    <---OpenShift Pod Configuration-->  <--------Container Configuration
Node/                                                    <--Process Ide
Time     PodName                 PodIndex      Name                  ProcessID Pro
-------- ----------------------  ------------  --------------------  --------- ---
06/22/23
00:15:00
rancha1
         rke2-canal-5n722        b815f6bd4ba1  calico-node                3725
                                               kube-flannel               3763 fla
         cert-manager-webhook    c317030510d4  cert-manager               2432 web
         rancher-7c5dbf46fc-z    02ad1e9d7318  rancher                    3497
         rke2-coredns-rke2-co    22a8cbf9d51f  coredns                    3349 cor
         rke2-ingress-nginx-c    31a310f130ed  rke2-ingress-nginx-c       3555
         rancher-webhook-577b    5a81e5c27074  rancher-webhook            3105 web
         kube-proxy-rancher-a    545065ae69ff  kube-proxy                 1697 kub
```

Configuration: RHOS

K8S Metrics added for: Pod, container, processID

```
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
```

VELOCITY SOFTWARE

Customer Case Study

RHOS Case study (Large Bank)

- Production environment, Growing!
- 9 rhos nodes (COREOS)
- Velocity containerized snmp
- standard snmp enabled
- RHOS no longer a black box

All standard Linux metrics captured via snmp

- Storage metrics show “opportunity”

```
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
```

VELOCITY SOFTWARE

What Else do we know?

Case study:

Production environment Storage over configured

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

```
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
```

VELOCITY SOFTWARE

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
--------    --- ----- ---- ---- ----
11/06/23
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
LIMEDL1     Tot   4.2  1.6  2.3 95.4
```

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**

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.