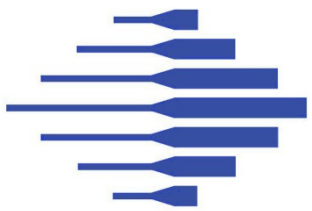


VELOCITY

SOFTWARE

Tailoring zVPS

ZVPS

- Data Flow
- Installation
- Operations Overview

ZTCP

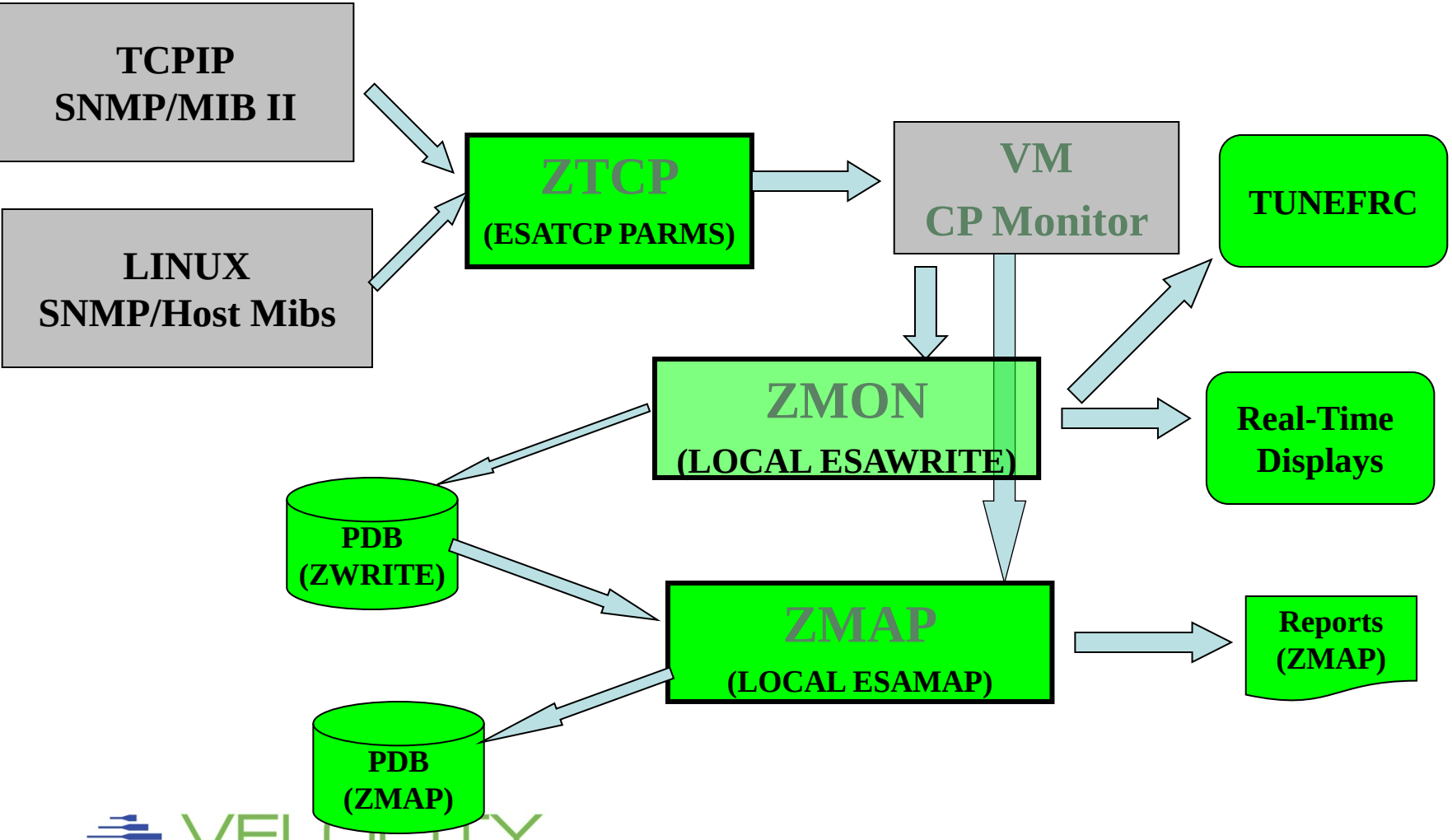
ZWRITE

ZMAP

ZMON

Accounting

Parameters are on the configuration disk



Data Flow

Data source:

- CP Monitor: Monitor DCSS (IBM)
- ZTCP – Linux Data

Current Data: ZWRITE 191

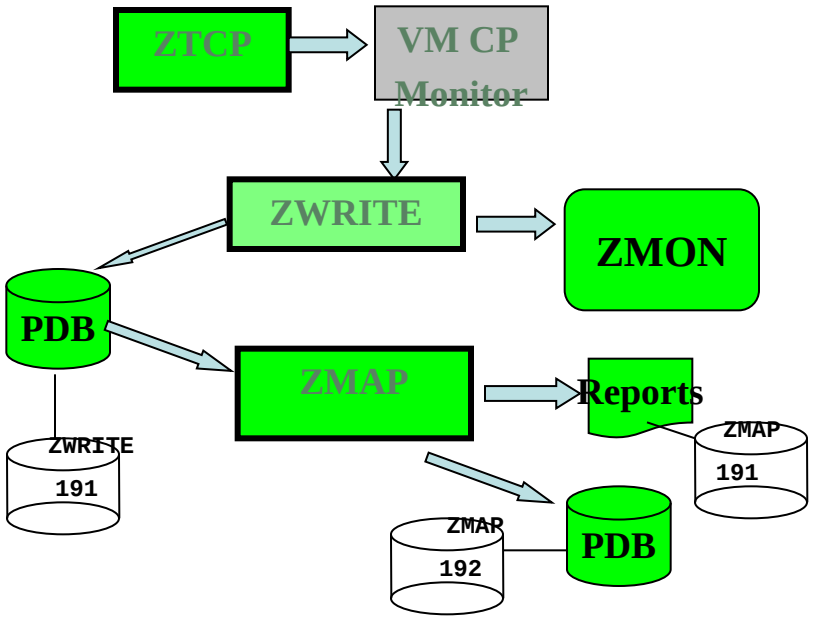
- One minute granularity
- Created by ZWRITE
- Used by ZMON
- Erased by ZMAP

Current Reports: ZMAP 191

- Work Disk (empty if all goes ok)
- OUT Reports
- 15-Minute Granularity

Archive data: ZMAP 192

- Monthly data



VELOCITY SOFTWARE

ZTCP

- Define Linux Nodes to collect data from

ZWRITE Functions

- Collect and process monitor data
- User classification

ZMAP Functions

- Automation (RUNAUTO, TRIM)
- ESAPRINT – create print format files
- ESAEXTR – extract data from performance data base

zMON Functions

- Extracts real time
- Alerts for operations

ESATCP PARMS

- sets defaults for community name, tcpip stack, port
- Read statements for including node files:
  - READ AUTHUSER ; read in file containing authorized users
  - READ PEERS ; read in file that defines peer ztcp ip addresses
  - READ DNS ; read in file that defines dns server ip address
  - READ LOCALLNX ; read in a node file that defines nodes

Node File Parameters – define nodes

- Node = ‘LINUX01’ Node name “
- Network_address = 10.129.13.3’ IP Address
- domain=‘linux01.VelocitySoftware.com'
- community = 'velocity'
- Ssiflag = ‘1’b

Parameter files for enterprise

```
TCPIP='TCPIP'
LPARNAME = 'PRODVM1'
peerport = 1998
peeraddr ='67.218.99.131' peerport = 1998   ;vsivm1
peeraddr ='67.218.99.132' peerport = 1998   ;vsivm2
peeraddr ='67.218.99.134' peerport = 1998   ;vsivm4

dnsport = 53
dnsIPADDR = '64.105.172.26'
```

ZTCP PARMS Sample File, sets defaults

```
;
TCPIP = 'TCPIP’
PORT = 161
COMMUNITY = ‘velocity’


READ VMLOCAL

READ LINUX1
```

Single Node Parameters

Parameters must be entered for each node

SSIFLAG tell ZTCP to measure node where it runs

```
Community = ‘vel5x4xxx’   ; different community
SSIFLAG = ‘1’b;    Nodes with this setting can be moved
NODEGRP = ‘APPS01’ ; tells zwrite to group these nodes

NODE = ‘LINUX001’  NETWORK_ADDRESS = '10.103.104.21‘
NODE = ‘LINUX002’  domain=‘linux002.VelocitySoftware.com'
```

SMSG command Interface

- “Smsg ztcp help”

Web interface – portal

Requires user authorization

```
;
authuser = 'MAINT'
authuser = 'ZMON'
authuser = 'ZTCP‘
Authuser = ‘zweb01’
```

SNMP alerts can be sent to any SNMP operations console

Create “SNMP TRAPDEST” file

```
* this file is the list of snmp trap destinations
* format is ip address, and community name
67.100.74.25 velocity

*  following is default
67.100.74.29 velocity 2B0601020102020101 TCPIP 162
```

Sending SNMP alerts (used by zALERT)

```
/* authorized user can send alerts */
parse arg msg
‘CP SMSG ZTCP ALERT’ msg
```

SMSG Interface Allows Dynamic config changes

Many functions for analyzing configuration

SMSG ZTCP HELP

```
ZTCP Command List
   CP command
   CMS command
   AUTHORIZE userid
   READ  filename
   SCAN ?
   SCAN x node
   START node
   STOP node
   ADDVMID node vmid
   RESET node
   QUERY [AUTH] | [DEADLINE]
   STATUS [node] [HOST | UCD | VMCF]
   TRACE [ON|OFF|VMCF|UCD|STORAGE|DEADLINE ]
   USER userid command
  End List
```

Status (from VM 3270 or via Web interface)

```
smsg ZTCP status
Ready; T=0.01/0.01 10:31:33

   zTCP Release 4.3.0.0

Status of monitored nodes at (time): 14:16:01 on LPAR: VSIDEV

NODE     Status  Actv lastResp Stack    Level    VirtMach NodeGrp  LPAR   SSI
-------- ------- ---- -------- -------- -------- -------- -------- -------- -
linux9   Started  Yes 14:15:51 Linux    UCD5.3.p LINUX9                     N
oracle   Started  Yes 14:15:57 Linux    VSI5.3.0 ORACLE           VSIVM4    N
OSA178   Started  Yes 14:15:53 Linux    VSI5.7.3 SLES12           VSIVM4    N
redhat6  Started  Yes 14:15:53 Linux    VSI5.5   REDHAT6          VSIVM4    N
redhat6M Started  Yes 14:15:50 Linux    VSI5.5   REDHAT6          VSIVM4    N
redhat6S Started  Yes 14:15:51 Linux    VSI5.5   REDHAT6          VSIVM4    N
redhat6x Started  Yes 14:15:53 Linux    VSI5.5   REDHAT6X         VSIVM4    N
redhat62 Started  Yes 14:15:53 Linux    VSI5.5   REDHAT6          VSIVM4    N
RH5X161  Started  Yes 14:15:50 Linux    VSI5.3.2 RH5X161                    N
sles11x2 Started  Yes 14:15:52 Linux    VSI5.4.2 SLES11X2         VSIVM4    N
s11s2ora Started  Yes 14:15:59 Linux    VSI5.4.2                            N
```

Add node command:

```
ADDNODE BOBS velocity 67.100.74.18

smsg ZTCP query
Ready; T=0.01/0.01 10:31:33

  Begin monitoring of following network addresses:
  NODE     IP_Address      Community        PORT VMCF TCPIP-VM Intervals Avail
  -------- --------------- ---------------- ---- ---- -------- --------- -----
  BOBHOME  208.60.43.66    bobtest1          161 OFF  TCPIP     60  3 60 10
  BOBLINX  10.0.0.109      bobtest1          161 OFF  TCPIP     60  3 60 10
  dslvsi   66.134.42.249   public            161 ON   TCPIP     60  3 60 10
  spamm83  168.103.222.83  public            161 ON   TCPIP     60 60 60 10
  VMLOCAL  67.100.74.19    velocity          161 ON   TCPIP     60 60 60 10
  BOBS     67.100.74.18    velocity          161 OFF  TCPIP     60 60 60 10
  End Display
```

Default parameter file: "ZMON ZWRITE"

- Do NOT modify this file
- Use "LOCAL ZWRITE" to modify

"LOCAL ZWRITE":

- Appended to "ZMON ZWRITE" before execution
- Should ONLY contain changes
- Use "%INCLUDE" statement for ease of management
- Most default settings are very appropriate
- Setting user classes most important

Sample "LOCAL ZWRITE":

```
Custname = 'Velocity Software'
tune_report = '1100'b;
%include MICS
%include USERCLAS
```

ZWRITE Output: User classes

User classes important

- Allows analysis of workloads
- Define “test servers” as a class
- Define “samba servers” as a class

Definitions in “USERCLAS ZWRITE”

- Many default classes, probably correct
- Users may be in multiple classes with nclass option
- Wildcards useful

```
Userclass(10,000) = ‘LNXTEST’   ; 000 is class name
Userclass(10,001) = ‘TST*’
Userclass(11,000) = ‘SAMBA’
Userclass(11,001) = ‘&&&SMB’
```

VELOCITY SOFTWARE

ZWRITE Output: History Files

File Types

- ESASYS (global data, dcss, address space, LPAR data)
- ESACPU (one record per engine, one total record)
- ESAUSER (users, classes, applications)
- ESADEV (devices, control units, seeks)
- ESATCP (Network, Linux)
- ESAOPER (configuration changes)

"MICS ESAWRITE"

```
mics_interval=10          ;3.1; Set interval for extract in
minutes
; Note that the time_zone parameter must be set for the MICS
extract = '0'b;                        ; MICS extract yes(1) or no(0)
extract_starttime = '00:00:00'     ; MICS extract start time
extract_stoptime  = '24:00:00'     ; MICS extract stop time
extract_record(0,1) = '1'b         ; MICS extract:  SYTSYP:
extract_record(0,2) = '1'b         ; MICS extract:  SYTPRP:
extract_record(0,3) = '1'b         ; MICS extract:  SYTRSG:
```

MXG

- look for 'XAMxxx' records

ZMAP Functions

- Report Writing
- Archive Creation and management
- Long term performance database management
- Zview graph definitions, create

Operation:

- Generate all reports, very little added cost
- Generate all history, very little added cost

Parameter files (LOCAL ESAMAP)

Automation (RUNAUTO PARMS)

ESAPRINT – All reports maintained by default

ESAEXTR – database interface language

ZMAP is Automated using RUNAUTO EXEC

- Profile calls RUNAUTO
- Runs ZMAP to create reports
- Copies history data to archive
- Runs ESAPRINT to create one “OUT” file with all reports
- Erases ZWRITE daytime files from ZWRITE 191
- Creates weekly/monthly “management reports”

Results:

- ZWRITE 191 clean
- Reports on ZMAP 191 (OUTxx files)
- Archive on ZMAP 192

RUNAUTO PARMS - ZMAP Automation

```
stdrept  LOCAL              /* Define LOCAL ESAMAP Parm file */
serve    ZSERVE             /* ZSERVE virtual machine userid*/

datadisk  100
datamode  J

ARC_DAYS 6                  /* Days of WRITE data to maintain*/

Mgtparm         MGTRPT      /* Management report parm file*/
Monthly_report  1           /* Monthly management reports */
WeekParm        MGTRPT      /* Weekly management parms    */
Weekly_report   1           /* Weekly management reports  */

MAPGOOD 1                   /* Enable User Exit           */

WeekStart Monday            /* Support different calander */
WeekDays  5                 /* Report full week  (7 days) */

ProcessRaw     0            /* Process raw data on WRT191 */

arcpct      90              /* Archive disk threshold     */
adiskblocks 10000           /* Trim A disk to available   */

SUPPORT BARTON@VELOCITYSOFTWARE.COM
SUPPORT SUPPORT@velocitysoftware.com

CHARTCNT 100 52 12          /* daily, weekly, monthly charts*/
```

RUNAUTO PARMS - ZMAP Automation – other parameters
Many choices - Most take defaults

```
exclude  Saturday Sunday  /* Exclude list, date(w) format*/

/***** Archive Disk: For longterm history files          */
arcdisk DATA:ZMAP.D192  /* SFS Example (Not recommended)*/
arcdisk 192               /* history archive disk       */
arcmode D                 /* enable archive to D disk.  */
```

MAPGOOD EXEC, Called from RUNAUTO when 'GOOD'

```
/* This exec is called by RUNAUTO at successful ZMAP execution. */
barton = 'barton at VelocitySoftware.com' ; sysprog= ''
Queue 'Input Subject: ZMAP Z/VM Reports'
files = 'ESASSUM ESATCP1'
Do i = 1 to words(files)
 Queue 'a3';Queue 'next 3'
 Queue 'PIPE < 'WORD(files,i)' listing | Spec 2-80 | Xedit'
end;
Queue 'COMMAND CMS SENDFILE ( NOTE'
'EXEC NOTE' barton sysprog '(NONOTEBOOK'
exit
```

MAPERROR (Phone Home) - 24 hours to fix problems...

```
/* This exec is called by RUNAUTO when there are problems */
Parse Arg maprc .
support = 'SUPPORT at velocity-software.com'
Queue 'Input Subject: ZMAP RUNAUTO Errors'
Queue 'a3';Queue 'next 3'
Queue 'get ZMAP errlog a'; Queue 'get local ZMAP  a'
Queue 'a3';Queue 'bot'
Queue 'pipe literal q disk | cms | xedit '
Queue 'COMMAND CMS SENDFILE ( NOTE'
'EXEC NOTE' support '(NONOTEBOOK'
exit
```

Other "Interesting Parameters"

DASD

- PAVBYDEV

User data

- classbyaci = '1'b
- report_user_rate = '1'b
- user_percent_of_total (out of 100, or percent of percent)
- report_user_rate (Rate vs Percent)

Other “Interesting Parameters”

Linux Support

- ProrateON=‘1’b

Network

- tcp_port_accum=80 (ignores ZTCP connections)
- tcp_port_accum=81 (ignores ZTCP connections)
- tcp_conn_thresh= 10 (bytes / second)
- tcp_threshold (datagrams)

First set parameters in the ZMAP/ZWRITE Parm file

```
; Following allow setting accounting charges in terms of service units
acct_report  = '1111'b        ;User accounting report(ESAACCT)
service_charge_factor = .00005;Set in terms of $$ per service unit
service_vcpu_factor = 2300    ;Service units per second of Virt CPU
service_tcpu_factor = 2300    ;Service units per second of Total CPU
service_diof_factor = .0001   ;Service units per I/O
service_stor_factor = .00003  ;Service units per page
```

Shows on heading

```
Report: ESAACCT      User Accounting Report
Monitor initialized:
------------------------------------------------------------
Service Unit Virtual CPU Factor:   2300.0
Service Unit Overhead CPU Factor:  2300.0
Service Unit I/O Factor:            0.000
Service Unit Storage Factor:        0.000
Charge per Service Units:            0.00005
```

And sample output showing linux servers:

```
Report: ESAACCT      User Accounting Report
Monitor initialized:                        on 9672 serial 205ED     First
-------------------------------------------------------------------------
22:30:48     41 31.0  1698.0   1528K     76.41  664.37     21822  5628K
 ***User Class Analysis***
*Servers     13  4.0    8.48  7633.3      0.38    3.32       789  11384
*LINUX        1  1.0    0.02   13.85      0.00    0.01         0    816
*TheUsrs     26 25.0  1647.8   1483K     74.15  644.71     21033  5612K
 ***Top User Analysis***
LAS122        1  1.0  242.73  218453     10.92   94.97      1756   492K
LAS114        1  1.0  238.72  214849     10.74   93.41       617   120K
LAS120        1  1.0  238.58  214725     10.74   93.35      1302   492K
LAS112        1  1.0  162.29  146066      7.30   63.50      1211   490K
LAS134        1  1.0  116.82  105136      5.26   45.71      1120   386K
LAS110        1  1.0  106.31   95677      4.78   41.59      1186   433K
```

For Linux application data, create file “LINAPP ESAEXTR”

```
extract:
x ='node'
y ='vsinap.name'
y ='vsinap.totcpu
```

Result:

```
*HDR Date    Time    NODE VSINAP.NAME VSINAP.TOTCPU
     20061029 221548 LAS118    *Totals*         711
     20061029 221548 LAS120    *Totals*         764
     20061029 221548 LAS122    *Totals*         745
     20061029 221548 LAS124    *Totals*        1547
     20061029 221548 LAS124    cron            1235
     20061029 221548 LAS126    *Totals*        1026
     20061029 221548 LAS126    perfadap         759
     20061029 221548 LAS128    *Totals*        1023
     20061029 221548 LAS128    cron             754
     20061029 221548 LAS130    *Totals*        1797
```

Database Extraction Using ESAEXTR

Input files may be:

- One minute data on zWRITE 191
- 15 minute data on zMAP 192

Format of command:

- ESAEXTR (WEEK 51 PARM LINAPP
- ESAEXTR (MONTH 12   PARM LINAPP

Or wild card:

- ESAEXTR 201604* (PARM LINAPP

VELOCITY SOFTWARE

ESAEXTR - Extraction from PDB

Write your own special purpose reports

Don't cut from reports….

Field names:

- History Keywords
- segment.fieldname

Record types

- Interval
- Summary

x, y: (no limit on number)

Criteria (>,<,=,^=,<=,>=)

Headings

Arithmetic functions (Left to Right)

Special Cases

Sample Extract

```
EXTRACT:

INTERVAL = 'INTERVAL'

X = 'NODE'
Y = 'UCDSYS.SWAPRATE'
Y = 'UCDSYS.SWAPPCT'
Y = 'UCDSYS.TOTCPU'
CRITERIA='UCDSYS.TOTCPU>1'


EXTRACT:
INTERVAL = 'INTERVAL'
X = 'NODE'
Y = 'HSTSFT.INDEX'
Y = 'HSTSFT.NAME'
Y = 'HSTSFT.CPUPCT'
Y = 'HSTSFT.PERFCPU'

CRITERIA='HSTSFT.CPUPCT>1'
```

VELOCITY SOFTWARE

ESAEXTR History Keywords

Format:

- Variable name, location on one line
- Variable description on next

```
* variable name  Offset DB  Dec/Len Type
DATE               0000 SYS   8     CHAR
'History interval date'
STARTTIME          0008 SYS   8     CHAR
'History interval start time'
STOPTIME           0016 SYS   8     CHAR
'History interval stop time'
SERIAL             0024 SYS   6     CHAR
'Processor serial number'
INTERVAL           0030 SYS   2     CHAR
'INterval or SUmmary'
QUALIFIER          0032       8     CHAR

'USERID, CPx, DEVNAME||DEVICE'
```

ESAEXTR: Headings

Titles

- TITLE (up to 4)
- TITLEC (title continuation)

Columns

- Default width 10

```
EXTRACT:            ;
TITLE = 'Stop     CU   Devc  <SSCH/sec-> <INTENSITIES> %DeV   '
TITLEC=' <----------Response times  (ms)--------->  MDisks'
TITLE = 'Time     No.  Type   avg  peak   I/O   Queue   Busy  '
TITLEC='  Resp   Queue   Pend  CtlQu  Disc  Conn   linked'
TITLE = '-------- ---- ----  ----- ----- ------ ------- ----- '
TITLEC=' ------ ------ ------ ------ ------ ------  -----'
COLUMNS = '8  5 5 7 6 7 8 6    8 7 7 7 7 7 7 6'
X = 'STOPTIME'
Y = 'DEVICE'
Y = 'DEVNAME'
Y = 'IODDEV.RATE'
Y = 'IODDEV.MAXSCMSSCH'    ; max ssch
Y = 'IODDEV.SCMDDTIM+IODDEV.SCMCNTIM*IODDEV.RATE'
Y = 'IODDEV.RESPTIME*IODDEV.RATE'
criteria = 'DEVNAME=3390'
criteria = 'IODDEV.RATE > 1'
.
```

ESAEXTR: Headings

Command:

"ESAEXTR FLOCK (PARM TEST"

Produces file "FLOCK EXTRACT"

*HDR Date	Time	Stop Time	CU No.	Devc Type	<SSCH/sec-> avg	peak	<INTENSITIES> I/O	Queue	%
20040114	010000	010100	152D	3390	5.6	5.6	25.2	26.3	
20040114	010000	010100	15AD	3390	6.5	6.5	27.4	28.5	
20040114	010000	010100	15ED	3390	7.3	7.3	34.2	35.7	
20040114	011500	011600	152D	3390	5.8	5.8	28.0	29.0	
20040114	011500	011600	15AD	3390	6.1	6.1	26.0	27.1	
20040114	011500	011600	15ED	3390	7.6	7.6	33.8	35.1	
20040114	013000	013100	152D	3390	5.4	5.4	23.2	24.1	
20040114	013000	013100	15AD	3390	6.6	6.6	26.4	27.6	
20040114	013000	013100	15ED	3390	7.6	7.6	31.7	33.0	
20040114	014500	014600	152D	3390	5.2	5.2	20.2	21.1	
20040114	014500	014600	15AD	3390	6.4	6.4	25.9	27.3	
20040114	014500	014600	15ED	3390	7.1	7.1	31.4	66.6	
20040114	015900	020000	15ED	3390	5.7	5.7	22.0	23.1	
20040114	020000	020100	152D	3390	5.6	5.6	21.7	22.9	
20040114	020000	020100	15AD	3390	6.6	6.6	27.9	29.5	
20040114	020000	020100	15ED	3390	7.4	7.4	34.3	35.8	

ESAEXTR: Arithmetic, Special Cases

- Arithmetic
- Calculated left to right
- No “parentheses’
- Variables or static values
- Special Cases
- Multiple Multiple segments don’t make sense
- LPAR data, HSTSFT, ASPACE, MDISK

```
EXTRACT:              ;
Y = 'IODDEV.RATE'
Y = 'IODDEV.MAXSCMSSCH'     ; max ssch
Y = 'IODDEV.SCMDDTIM+IODDEV.SCMCNTIM*IODDEV.RATE'
Y = 'IODDEV.RESPTIME*IODDEV.RATE'
```

Daily/Weekly/Monthly CSV files created, zview exposed….

```
; Chart Syntax: type days strt stop extr parm
;  Where type is DAILY/WEEK/MONTH
;   "days" is the previous "n" days
;   TimesT is called with these parms

;       type    days  strttime stop   extractname  parm
CHART  DAILY     1     00:00    24:00   CPULPAR
CHART  DAILY     1     00:00    24:00   USERCPU

CHART  MONTHLY   31    00:00    24:00   CPULPAR
CHART  MONTHLY   31    07:00    17:00   USERCPU
CHART  MONTHLY   31    *        *       USERWAIT    SUSELNX1

CHART  WEEKLY    7     07:00    17:00   USERCPU
CHART  WEEKLY    7     00:00    24:00   CPULPAR

; Charts will be kept up to 12 months,52 weeks, 31  days
; format of chartcnt is "chartcnt mm ww dd"
CHARTCNT 2 2 2
```

```
;LPAR Utilization over time
"STARTTIME","STOPTIME","LPARNAME","CPUUTIL"
"2010/12/13","02:00:00","Totals:","79.64"
"2010/12/13","02:00:00","VSIVM4","34.72"
"2010/12/13","02:00:00","VSIVM1","3.87"
"2010/12/13","02:00:00","VSIVM2","0.41"
```

```
ESAMAP   FILELIST A0  V 1
  Filename Filetype Fm
  TOTAL    CSVDC348 A1
  TOTAL    CSVDU348 A1
  TOTAL    CSVWC049 A1
  TOTAL    CSVWU049 A1
  TOTAL    CSVDC347 A1
  TOTAL    CSVDU347 A1
  TOTAL    CSVDC346 A1
  TOTAL    CSVDU346 A1
```

VELOCITY SOFTWARE

Set the report selection and order for OUTxxx

Executed by RUNAUTO

Create ESATOC, ESAINDEX

ESAMAP43 ESAPRINT:

```
*
ESAHDR ESATUNE
*Performance Summary
ESASSUM ESASUM
*Transaction Activity
ESAUSLA ESAXACT ESARATE
ESACLAS ESAEXCP
*User Activity
ESATUNA
ESASRVC ESASRV1 ESAUSRC ESAUSR1 ESAUSR2 ESAUSR3 ESAUSR4
ESAUSEK
ESAUSP2 ESAUSP3 ESAUSP4 ESAVECT
ESAWKLD ESAUSRQ ESASCED
ESAACCT
```

ESAPARM EXEC : setting up Operations

```
MONITORSTART = 'BLOCK 4'

/* The variable ZWRITEparms contains the parameter string*/
/* that is sent to the ZWRITE command                    */
ESAWRITEPARMS = 'NONE (PARM LOCAL'
/* Parameters for ESACLDSK execution                      */

ESACLDSK_PERCENT  = 80              /* zwrite 191 Target percent*/
ESACLDSK_DISKVDEV = 100             /* Virtual device addr */
ESACLDSK_DISKMODE = 'J'             /* Virtual device mode */
ESACLDSK_WRITEMOD = 'A'             /* ZWRITE's disk mode*/

Exit(0)
```

Create New screen file FREE ESAMNSCR

```
:TITLE.Free Storage Analysis
:HEADING
          <------Load-------> <Free Store Pgs>
:EHEADING
:HEADING
          <---Users----> Tran <Subpools>  Long
:EHEADING
:HEADING
Time        On Actv In Q /sec  Alloc Used  Term
:EHEADING
:HEADING
--------  ---- ---- ---- ----  ----- ---- -----
:EHEADING
:FORMAT
hh:mm:ss  nnnn nnnn nn.n nn.n  nnnnn nnnn nnnnn
:EFORMAT
:VIEWS
  1  30  31  48
:EVIEWS
:FIELDS
STOPTIME SYTUSR.SYSUSRS ACTIVE.VACTIVE
SYTSCG.SRMCDISP SYTUSR.CALTOTCT
SYTRSG.RSAFSTOR/512 SYTRSG.RSAFSYUD/512
SYTRSG.RSAXFREE-SYTRSG.RSAFSTOR/512
:EFIELDS
:SOURCE.INTERVAL
```

New screen:

- Following sample “FREE ESAMNSCR”
- Ensure the “ZMON CODE” disk is accessed
- Issue “ESAMONSC FREE (NOUPDATE”
- Issue “ZMON FREE
- Note: ZMON loads modules and does not re-load

```
Screen: FREE     Velocity Software                ESAMON 4.240
1 of 1  Free Storage Analysis

          <------Load-------> <Free Store Pgs>
          <---Users----> Tran <Subpools>  Long
Time        On Actv In Q /sec  Alloc Used  Term
--------  ---- ---- ---- ----  ----- ---- -----
16:04:00    41   18  1.0  1.9      0    0     4
16:03:00    41   18  2.0  2.3      0    0     4
16:02:00    41   18  1.0  2.1      0    0     4
16:01:00    41   18  2.0  1.9      0    0     4
16:00:00    41   21  1.0  1.8      0    0     4
15:59:00    41   17  1.0  2.0      0    0     4
15:58:00    41   17  3.0  1.9      0    0     4
15:57:00    41   17  2.0  2.0      0    0     4
15:56:00    41   17  1.0  1.9      0    0     4
15:55:00    41   21  2.0  2.3      0    0     4
15:54:00    41   15  2.0  1.8      0    0     4
```

Rexx FUNCTIONS

- :NEWDATA.CUSTOM
- Calls “C”screen-name ZMON to extract and manipulate output :
- Creates stem “newline.”

Example: ESALPARS ESAMNSCR

- Calls CESALPAR ZMON

```
:SOURCE.INTERVAL
:NEWDATA.CUSTOM
:ORDER.FORWARD
:SORT.1 DESCENDING
:PARM.LPAR * CPU ALL
:PARMGROUP.LPAR
:OPTIONS.DEDUP  3
```

Tailoring ZMON Displays (3270)

ESAMNDEF COPY

- Provides the default settings (do not modify)

```
alert.ESAMAIN    = 'ON'
alert.ESAMAIN.8  = '> 80 RED REVVIDEO 80*NCPUS'  /* t-cpu   */
alert.ESAMAIN.17 = '> 25 RED REVVIDEO'           /* DASDresp*/
alert.ESAMAIN.20 = '> 95 YELLOW REVVIDEO'        /* MDC     */
```

MONPROF COPY supercedes

- Tailored to meet installation/user requirements
- User defined screen alerts
- Following sets alert for CPU Utilization to 5%

```
alert.esamain = 'ON'
alert.esamain.8 = '> 05 RED REVVIDEO'          /* t-cpu   */
```

Criteria in “MONPROF COPY” override screen defaults:

```
:CRITERIA
UCDSYS.SYSTPCT > 0
:ECRITERIA
```

Override with:

```
Criteria.ESAUCD4 = ‘UCDSYS.SYSTPCT >= 0’
```

Sample MONPROF COPY:

```
alerts      = 'YES'
defaultscreen = 'SMART'
alert.esamain = 'ON'
alert.esamain.8 = '> 05 RED REVVIDEO'     /* t-cpu   */
criteria.esaucd4 = 'UCDSYS.SYSTPCT >= 0'
```

PROFILE EXEC

```
/* */
‘vmlink .dir SFSZVPS:ZMON.CODE'
Dsc = (Word(DiagRC(24,-1),2)¬=0)
if (dsc ¬= 'DSC')
 Then Do;
  say 'Userid' userid() 'is not disconnected'
  say 'not starting ZMON'
  Exit;
 End;
'CP DEF GRAF 202’ ; 'CP DEF GRAF 203’
‘CP DEF GRAF 8A1’ ;’ CP DEF GRAF 8A2'
'EXEC ZMON VPLOT'
'CP LOGOFF'
```

Terminal Definition file: “TERMINAL LIST” (see the trick?)

```
/*   Terminal definitions for ZMON screens
8A0 8A1 8A2 8A3 8A4 8A5   /* Dialable
/* Following are attached/dedicated devices for operations:
901 902
```

REXX “extract” macros

extract real time data and process

(Same interface Used for all data retrieval)

Variables defined in Performance Data Reference

```
/* */
'EXTRACT FROM 07:00 UNTIL 08:00 USER *',
   'WHILE USEACT.VMDTTIME > 0',
     'FIELDS STOPTIME USERDATA.USERID USEACT.VMDTTIME',
       'USEACT.ACTVWSS USERACTV.NUSERS‘

Do while RC = 0
   Say Stoptime '-' userdata.userid
     Say 'TTime: ' useact.vmdttime
     Say 'Avgwss:' useact.actvwss/useractv.nusers
  'RETRIEVE'
End
```

Real time alerts – sample "ZMON VMALERT"

```
 Screen: VMALERT          Velocity Software, Inc.         31 Oct 2006 08:04:42
------------------------ Exceptions Analysis Alerts ------------------------

   Type Description
   INQU 3.0  users in queue
   PGAL Page volume PAGEDE (0402) is     54.29% full
   SPFU User spool frames for TCPIP   : 102
   SPOL Spool space is   49.66% used
   VMCP User SNMPD    at      5.40% of processor
   VMCP User TCPIP    at      6.83% of processor
   VMCP User VMPOP    at      5.57% of processor
   VMID User EWEBADM  idle for   482 minutes
   VMID User EWEB001  idle for    22 minutes
   VMID User EWEB002  idle for   325 minutes
   VMID User EWEB003  idle for    22 minutes
   VMID User EWEB004  idle for    22 minutes
   VMID User EWEB005  idle for    22 minutes
   VMID User EWEB006  idle for    22 minutes
   VMID User EWEB007  idle for   325 minutes
   VMID User EWEB008  idle for    22 minutes
   VMID User EWEB012  idle for   325 minutes
   VMID User VMSERVR  idle for  1199 minutes
   VMID User VMSERVS  idle for 71451 minutes
   VMID User VMSERVU  idle for 71451 minutes
   VMID User VSIFTP   idle for    97 minutes
```

Real time alerts

- Either display or disconnected
- Alerts defined in file "**name** MONALERT"
- Alerts set for any variable in Performance Data Reference
- Executed with "ZMON **name**"
- Alerts sent to 3270 Display
  - level 15 yellow rev
  - text User spool frames for &userid: &user_spool
- Alerts call functions
  - level 15 yellow rev action **cp msg op** &userid running crazy
- Alerts for SNMP
  - level 80 red rev **action cp smsg ZTCP** add page volume
  - Level 80 red rev **ALERT PG01** add page volume

Real time alerts

- Additional functions created with REXX macros

```
/*============================>
extract
parms user *
criteria userdata.userid <> 'System:' & useact.vmdttime = 0
var     userid    | 8   | userdata.userid
var     last_date | 12  | userdata.lastactd
function alrtidle  | 5 0 | &last_date

alert alrtidle vmid
exclude userid vmid
level 10 blue
level 15 yellow
level 25 red
text User &userid idle for &alrtidle minutes
```

ALRTIDLE EXEC

```
/*-------------------------------------------*/
address ''

Parse upper arg Ldate logcnt .    /* yymmddhhmmss */

If logcnt = 0 Then Return 0

Numeric digits 16

Cdate = Date('B')       /* days since base date   */
Ctime = Time()          /* time since midnight    */
                        /* seconds since base date*/
Cdate = (Cdate*24)*60 + (Left(Ctime,2)*60) +
Substr(Ctime,4,2)

Ltime = Substr(Ldate,7,6)    /* hhmmss */
Ldate = Left(Ldate,2)'/'Substr(Ldate,3,2)'/' ||,
        Substr(Ldate,5,2)    /* yy/mm/dd */
Ldate = Date('B', Ldate, 'O')
Ldate = (Ldate*24)*60 + (Left(Ltime,2)*60) +
Substr(Ltime,3,2)

Return (Cdate - Ldate)  /* Returns idle minutes*/
```

Real time alerts – disk checking

```
extract
parms node *
criteria hstmem.used > 0
var    node    | 8   | tcpip.node
var    diskpct | 6 2 | (hstmem.used/hstmem.size)*100
var    desc    | 16  | hstmem.desc
function diskchck  | 6 2 | "&node &diskpct &desc"

alert diskchck abcd
level 1  red rev APPMS002 &node &desc is at &miskpct%
level 2  red rev APPMS003 &node &desc is at &diskpct%
level 3  red rev APPMS004 &node &desc is at &diskpct%
level 4  red rev APPMS005 &node &desc is at &diskpct%
text &desc area on &node is &diskpct% full
```

DISKCHCK EXEC

```
/* exec to suppport ZMON alerts on filesystems*/
parse arg node pct descr .
Select;
 /* Eliminate standard stuff*/
 when descr = 'Memory' Then rsp = 0;
 when descr = 'Real'   Then rsp = 0;
 when descr = 'Swap'   Then rsp = 0;
 When descr = '/temp' then rsp = 0

 /* put in special checks */
 When descr = '/boot         ' then rsp = 0
 When descr = '/tmp          ' then rsp = 0
 When descr = '/var' then rsp = 1

 when pct < 10 then rsp = 0

 when node ='linux93'
  Then Do;
    If descr = '/' & pct > 30 Then rsp = 2
  end;

 When SUBSTR(descr,1,4) = '/opt' Then rsp = 0
 When pct > 50 Then rsp = 2

 otherwise rsp = 0;
end
return rsp
```