Scheduler and Dispatcher

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

Objectives

- Understanding Scheduler / Dispatcher
- How SRM affects users
- How SHAREs affect users

What is important?

- When users / servers get dispatched
  - Prioritizing work (Share values)
- How long are they dispatched for (time slice)
- What happens when there are resource constraints (eligible list)

The Scheduler

- Maintains the lists of users
  - Eligible, Dispatch, Dormant
- Calculates “deadline” priorities
- Determines Eligibility to be Dispatchable

Note: CP's virtual processor management has been improved so that no user stays in the eligible list more than an instant before being added to the dispatch list. Therefore some functions intended to improve performance by managing the eligible list, such as the DSPBUF option, are now less meaningful.

The Dispatcher

- Selects a user to run
- Dispatches units of work

The Scheduler (z/vm 6.4)

```
q srm
IABIAS : INTENSITY=90%; DURATION=2   (interactive/cms only, impacts q1)
LDUBUF : Q1=100% Q2=75% Q3=60%       (no longer used)
STORBUF: Q1=300% Q2=300% Q3=300%     (no longer used)
DSPBUF : Q1=32767 Q2=32767 Q3=32767  (no longer used)
DISPATCHING MINOR TIMESLICE = 5 MS    (how long a looping user allowed)
MAXWSS : LIMIT=9999%                  (deadly to Linux)
...... : PAGES=46423259
XSTORE : ---                         (no longer used)
LIMITHARD METHOD: CONSUMPTION        (vs traditional “deadline”)
POLARIZATION: VERTICAL               (PARKING, vs Horizontal no parking)
GLOBAL PERFORMANCE DATA: ON          (input for parking available)
                                     (excessuse “HIGH” is aggressive)
EXCESSUSE: CP-MEDIUM  ZAAP-MEDIUM  IFL-MEDIUM  ICF-MEDIUM  ZIIP-MEDIUM
CPUPAD: CP-100%  ZAAP-100%  IFL-100%  ICF-100%  ZIIP-100%
DSPWDMETHOD: RESHUFFLE              (reshuffle pldv, vs rebalance (BAD))
UNPARKING: LARGE                    (???  Which causes less parking?)
Ready; T=0.01/0.01 15:16:06
```

The Scheduler – queues still maintained

Screen: ESAUSRQ Velocity Software - VSIVM4 ESAMON 4.3
2 of 3 User Queue and Load Analysis CLASS *

Time	UserID /Class	<----Average Number of Users-----> <--------in Dispatch List--------> Q0	Q1	Q2	Q3	Ldng	Limit List
15:29:00	System:	0.967	9.767	2.217	11.850	0	0
	GPFS	0	0	0	3.000	0	0
	KeyUser	0.850	0	0	0	0	0
	NCS	0	0	0	0	0	0
	ORACLE	**0**	**0.133**	**0.467**	**2.050**	0	0
	REDHAT	0	0.317	0	1.000	0	0
	Servers	0.050	0.017	0	0	0	0
	SUSE	0	1.283	0.067	3.000	0	0
	TheUsrs	0	5.083	0.750	1.767	0	0
	TEST	0.017	1.683	0.550	0	0	0
	UBUNTU	0	0	0	1.000	0	0
	Velocity	0.033	0.183	0.050	0	0	0
	Web	0.017	0	0	0	0	0
	ZPRO	0	1.067	0.333	0.033	0	0

"drop from queue"

Old oracle 10 did not poll, new one does

```
Screen: ESAORAC  Velocity Software - VSIVM4     ESAMON 4.330 06/24 15:30-15:3
2 of 2  Oracle Database Configuration           NODE *              2828 0414


                    Database          Database   <------Database------->
Time     Node       Instance          Version    <----Start-----> Status
-------- -------- ---------------- ---------- ---------------- ------
15:31:00 oracle   orcl             10.2.0.4.0 2018/06/21 12:41 OPEN
         sles12   db02             12.2.0.1.0 2018/05/31 16:47 OPEN
                  db01             12.2.0.1.0 2018/05/31 16:46 OPEN
         REDHAT6X db01             11.2.0.2.0 2018/06/21 16:11 OPEN
```

Scheduler affected by:

- SET SRM STORBUF (control storage utilization Not used)
- SET SRM DSPBUF (control processor utilization Not used)
- SET SRM LDUBUF (control paging device utilization Not used)
- SET SRM DSPSLICE (time slize, default 5ms)
- SET SRM IABIAS (bias interactive users no value)
- **SET SHARE (guarantee a share of CPU)**
- SET QUICKDSP (ignore STORBUF, DSPBUF, LDUBUF)

Dispatcher affected by:

- SET SRM DSPSLICE

SET SHARE

Shares are “normalized” to workload

- Absolute is fixed percent
- Relative is relative to other relative

Absolute vs Relative

- Absolute shares go up as workload increases
- Relative shares go down as workload increases

Use Absolute shares for: (Ignore IBM defaults)

- **Servers that need more resource as more users log on**
- **Examples: TCPIP, RACF, Database servers**

Use Relative shares for users

QUICKDSP does NOT impact share values!

VELOCITY SOFTWARE

Scheduler Lists Overview

Dormant List

- Idle users, those logging on, logging off
- No special order
- Any user idle for 300ms or more,
- Traditional CMS workloads

Eligible List (not used anymore)

- Contains users who want to consume resources
- Users not yet allowed to contend,
  - Short on storage
  - Short on paging devices
- Kept in priority order

Dispatch List

- Users contending for resources now
- Kept in priority order
- Linux always here

VELOCITY SOFTWARE

Important Terms

Dispatch Queue (Dispatch List)

- The list of virtual machines requesting resource (working)

Dispatch Time Slice

- maximum time virtual machine dispatched

Elapsed time slice

- Maximum Time in queue before q-drop

Queue Drop (Prior to z/vm 6.3)

- virtual machine is done working, or ETS has expired

Dormant List

- Idle users **(Idle for 300ms)**

Eligible List (Prior to z/vm 6.4)

- Virtual machines that want to do work, but are held back

Dispatch List Definition

Class 1 (Interactive)

- Entry from the Dormant List
- Initial Q1ETS (variable from .05 seconds to 16 seconds)
- IA (InterActive) Bias applies

Class 2 (Non–Interactive)

- Entry after one ETS in Class 1
- Q2ETS is 8x Class 1 ETS (fixed multiple)
- Long running user will get 1 Q2ETS stay in Q2 before demotion

Class 3 (Long–running - Polling, batch, guests)

- Entry after one stay (8x ETS) in Class 2
- Q3ETS is 48x the Class 1 ETS (fixed multiple)

Used to understand what server is polling….

Users start in class 1, graduate to class 2, then 3

Example, Linux servers (WAS, DB2…) users in Q3

Report: ESAUSRQ User Queue and Load Analysis

UserID /Class	<User Load> Logged on	Non-Idle	Active	Disc-conn	Total InQue	Tran /min	<Average Num Dispatch List> Q0	Q1	Q2	Q3
05:06:00	58.0	.	33.2	.	25.4	259	4.0	2.4	0.6	18.4
Hi-Freq:	58.0	34	33.2	56	23.7	233	3.3	0.6	1.5	18.3
***Key User Analysis ***										
VMSECURE	1.0	1	1.0	1	0	3.6	0	0	0	0
User Class Analysis										
Servers	16.0	9	9.0	14	0.1	20.0	0	0.1	0	0
KeyUsrs	2.0	2	2.0	2	1.3	106	1.3	0	0	0
ZVPS	9.0	5	5.0	9	0.1	37.2	0	0.1	0	0
Linux	13.0	12	12.0	13	20.1	35.6	0	0.3	1.5	18.3
TheUsers	15.0	4	3.2	15	2.0	30.4	2.0	0.0	0	0
Top User Analysis										
ZLNXB20	1.0	1	1.0	1	1.0	0	0	0	0	1.0
ZLNXB15	1.0	1	1.0	1	1.0	0	0	0	0	1.0
ZLNXB21	1.0	1	1.0	1	1.0	0	0	0	0	1.0
ZLNXB16	1.0	1	1.0	1	1.0	0	0	0	0	1.0
ZLNXB17	1.0	1	1.0	1	1.0	0	0	0	0	1.0
ZLNXB10	1.0	1	1.0	1	1.0	9.6	0	0.1	0.4	0.5
ZLNXB18	1.0	1	1.0	1	1.0	0	0	0	0	1.0

Fair Share Scheduler (Wheeler scheduler):

- Allows prioritization of work
- Determines work “Eligibility” (deprecated with z/vm 6.4)
- Protects workload from resource over commitment using the “eligible List” - no “Thrashing”
- Supports 1000’s of concurrent virtual machines
- Maintains dispatch list to create fair share
- Allows wide range of workloads to effectively utilize resource

Also called DEADLINE SCHEDULING

- Every inqueue user assigned a deadline

Question: What are we trying to control with Eligible?

- Fair share based on business requirements
- System responsiveness when resources constrained

Dispatch List Deadline Priority

Deadline priority is a “target” time of day

- Deadline = TOD + DelayFactor
- “Dispatch List” and “Eligible List” priority are of this type
- Based on ATOD (artificial time of day)

Dispatch list delay factor:

- Based on “Normalized” share
- Delay factor = DSPSLICE / (ncpus * normalized share)
  - 1% share will have 100 time slice delay (500ms)
- Subtract IABias (Interactive Bias – first n times enters Q1)
- Subtract PageBias (E2/E3 users with stolen pages) deprecated
- Deadline is calculated after every dispatch time slice is completed.

Setting Shares

Looping users (1991 survey done with vtam)

- Does a looping user affect other users?
- Do you have TCPIP at relative share 10000?
- Are TCPIP's high share and looping users affecting other users related?
- How much excess share does RELATIVE 10000 create?

Why set share to relative 10000 anyway???

- Recommendation from VM development without analysis? They don't recommend it now.
- Destroys scheduler ability to "fair share"

What is normalized share? (WHY IMPORTANT?)

Excess share created by giving TCPIP REL 3000

- REL 3000 on many systems: 50% normalized share, uses 1%

Starting with 3 looping users RELATIVE 100 share

- They all get equal share of the resources
- this is as we expected.

```
Screen: ESAUSP2  Velocity Software-Test VSIVM4  ESAMON 3.778
<-------Main Storage-------->
         UserID   <Processor> <Resident->  Lock <-WSSize-->
Time     /Class   Total  Virt Total  Actv   -ed Total  Actv
-------- -------- ----- ----- ----- ----- ----- ----- -----
00:11:00 ROBLNX1  32.39 32.38 15862 15862    11 15536 15536
         ROBLX2   32.12 32.11 66136 66136   259 78478 78478
         ROBLX1   32.02 32.01 38219 38219   176 37790 37790
         ROB2LV    0.01  0.00  2246  2246     0  2246  2246
```

We now give ROBLX2 a RELATIVE 200 share

- because that is a more important service
- (nothing with virtual 2-way).
- Not as expected, it gets the excess share

```
 Screen: ESAUSP2  Velocity Software-Test VSIVM4  ESAMON 3.778
1 of 3  User Percent Utilization                    CLASS * USER
                                <-------Main Storage-------->
         UserID   <Processor> <Resident->  Lock <-WSSize-->
Time     /Class   Total  Virt Total  Actv   -ed Total  Actv
-------- -------- ----- ----- ----- ----- ----- ----- -----
00:14:00 ROBLX2   68.71 68.68 66211 66211   258 78478 78478
         ROBLX1   14.00 14.00 38245 38245   256 37790 37790
         ROBLNX1  13.99 13.99 15879 15879    11 15536 15536
         ROB2LV    0.01  0.00  2246  2246     0  2246  2246
```

Now for the experiment – Set shares "correctly"

- we reduce the relative share for all idle but inqueue users down to 1
- Convert TCPIP from REL 3000 to ABS 2%
- (using the allocated share computation below and showing how much allocated / consumed share is).
- This ELIMINATES "EXCESS" bucket – allows perfect case scenario

```
Screen: ESAUSP2  Velocity Software-Test VSIVM4  ESAMON 3.778
1 of 3  User Percent Utilization                CLASS * USER
                                <-------Main Storage-------->
         UserID   <Processor> <Resident->  Lock <-WSSize-->
Time     /Class   Total  Virt Total  Actv   -ed Total  Actv
-------- -------- ----- ----- ----- ----- ----- ----- -----
00:20:00 ROBLX2   48.39 48.37 67141 67141   292 80047 80047
         ROBLNX1  24.19 24.19 16168 16168    11 15536 15536
         ROBLX1   24.19 24.18 39006 39006   241 37790 37790
         ROB2LV    0.01  0.00  2246  2246     0  2246  2246
```

Starting with 3 looping users RELATIVE 100 share

- They all get equal share of the resources
- this is as we expected.

```
Screen: SMART    Velocity Software        ESAMON 4.301 01/22 09:47-09:
1 of 1  Smart
-----------------------------------  -------------------------------
    <----------Top Users---------->  <----------Servers------------>
    Userid:     CPU%  IO/Sec Pg/Sec  Userid:     CPU%   IO/Sec Pg/Sec
 1) BART2       27.8       0      0  System:     88.0    16.12      0
 2) BART3       27.8    0.33      0  RACFVM       0.2     1.95      0
 3) BART1       27.2       0      0  TCPIP        0.2        0      0
 4) OPERATOR     1.1       0      0  TCPIP2       0.1        0      0
 5) ZVPS         1.1       0      0  RSCS         0.0        0      0
 7) VMSYSVPS     0.8   12.47      0
10) ZWRITE       0.3    1.00      0
```

Excess Share Analysis (6.4)

We now give BART2 a RELATIVE 200 share

- because that is a more important service
- With EXCESS SHARE, Better, definitely Not “perfect”

```
Screen: SMART    Velocity Software        ESAMON 4.301 01/22 09:53
1 of 1  Smart
--------------------------------  --------------------------------
    <----------Top Users---------->  <-----------Servers------------>
    Userid:     CPU%  IO/Sec Pg/Sec  Userid:     CPU%   IO/Sec Pg/Sec
 1) BART2       48.9       0      0  System:     89.9     1.00      0
 2) BART1       19.3       0      0  TCPIP        0.2        0      0
 3) BART3       19.3       0      0  TCPIP2       0.1        0      0
 5) ZWRITE       0.3    0.50      0
 7) ZVPS         0.2       0      0
 8) ZTCP         0.1       0      0
 9) VMSYSVPS     0.0    0.38      0
```

Share settings – WITH EXCESS SHARE 10000:

- Doesn't look right (But better than z/VM 6.3)
- Not different from when low excess share

```
 Screen: SMART    Velocity Software
-----------------------------------
    <----------Top Users---------->
    Userid:     CPU%  IO/Sec Pg/Sec
 1) BART3       41.5       0      0     REL SHARE 300   REASONABLE
 2) BART2       27.2       0      0     REL SHARE 200   REASONABLE
 3) BART1        9.8       0      0     REL SHARE 100   NOT RIGHT
 4) BLAKE001     6.8    0.47      0     REL SHARE 10000, excess
 5) ZALERT       0.8       0      0
 6) ZWRITE       0.6    5.43      0
 9) ZSERVE       0.1    0.07      0
10) ZTCP         0.1       0      0
```

Calculation of Normalized Share

All ABSOLUTE and RELATIVE shares “normalized”

- Sum the Absolute shares of all VMDBKs in Dispatch list (SRMABSDL)
- Sum the Relative shares of all VMDBKs in Dispatch List (SRMRELDL)

```
Report: ESASUM System Summary
Variable Average Minimum Maximum Description
-------- ------- ------- ------- ----------------------------------------
SRMBIASI      90                 Interactive bias intensity percent (SET SRM I
SRMBIASD       2                 Interactive bias duration (SET SRM IAB)

SRMTSLIC    5.00                 Minor time slice (ms) (SET SRM DSPSLICE)
SRMTSHOT    2.00                 Minor time slice (ms) for HOTSHOT users

SRMABSDL    52.0    48.0    55.0 Total absolute shares of VMDBKs in the dispat
SRMRELDL     818     550    1900 Total relative shares of VMDBKs in the dispat
```

VELOCITY SOFTWARE

Calculation of Normalized Share

If SRMABSDL is less than 100%

- Normalized share equals Absolute Share
- Relative Share users get:

  (100 - SRMABSDL) x (relative share / SRMRELDL)

If SRMABSDL is greater than 99,

- Absolute shares “normalized” to 99
- Relative users “share” 1 percent
- Very dangerous situation

Normalized shares are percentages of the CPU resource

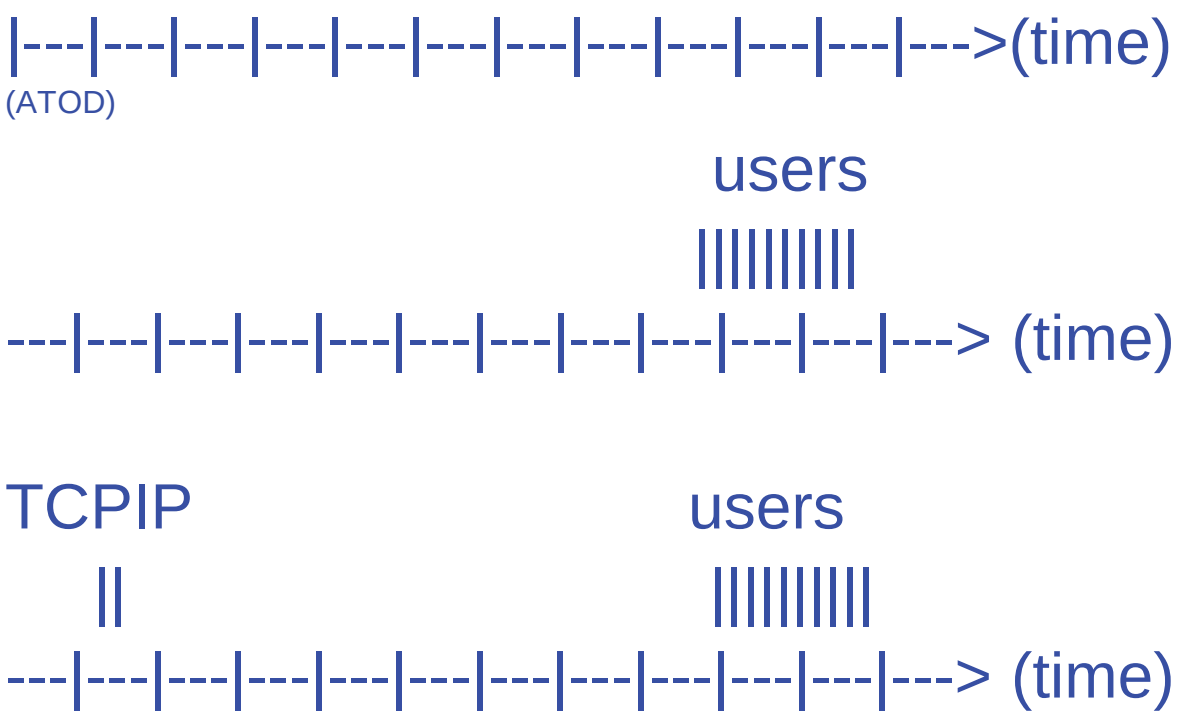
Delay factor (OFFSET) is then DSPSLICE / “normalized” share

VELOCITY SOFTWARE

Setting Deadline

Deadline time of day = current TOD + offset

Offset = (DSPSLICE / Normalized share) * bias



Dispatcher takes users in order by time from sorted deadline list

VELOCITY SOFTWARE

SHARE Impact on CPU Delivery Rate

CPU Delivery Rate for “one cpu system”

If normal share is 10%, user will have:

- Delivery rate = 1 dispatch time slice out of 10.
- Offset = 10 dispatch time slices.

If normal share is 50%, user will have:

- Delivery rate = 1 dispatch time slice out of 2.
- Offset = 2 dispatch time slices.

If normal share is 1%, user will have:

- Delivery rate = 1 dispatch time slice out of 100.
- Offset = 100 dispatch time slices.

Worst case seen – offset for general users:

- 30 minutes

Sample Deadlines

Example (50 users using IBM Defaults)

- RACF has relative share 10000
- TCPIP has relative share 10000
- User has relative share 100
- DSPSLICE = 5ms
- SRMRELDL = 25000 (typical)
- **(100 - SRMABSDL) x (relative share / SRMRELDL)**

Normalized share = 100 / 25000 = .004 (.4%)

- CPU Delivery rate = 5ms / .004
- = 5ms per 1.25 seconds
- Subsecond obviously NOT the design point

Sample Deadlines - Comparison

Example 1:

- TCPIP offset 2.5 dspslice (Share 10000)
- Users offset 250 dspslice (1.25 seconds)

Example 2: Change tcpip/racf share to ABSOLUTE 20

- TCPIP offset 5 dspslice
- Users offset 84 dspslice (.42 seconds)

VELOCITY SOFTWARE

Sample Deadlines - Comparison

Did it make a difference to RACF/TCPIP to reduce share?

- NO. Still number one always on dispatch list

Did it make a difference to users?

- Yes, they are guaranteed 3 times the amount of CPU when looping users are on the system

Does setting shares too high for some users impact other users?

- Only when large CPU consumers (including loopers) exist.
- IBM does not let looping users on their benchmark systems.

Recommend low ABS shares when appropriate for servers

SET SRM IABIAS pct nn

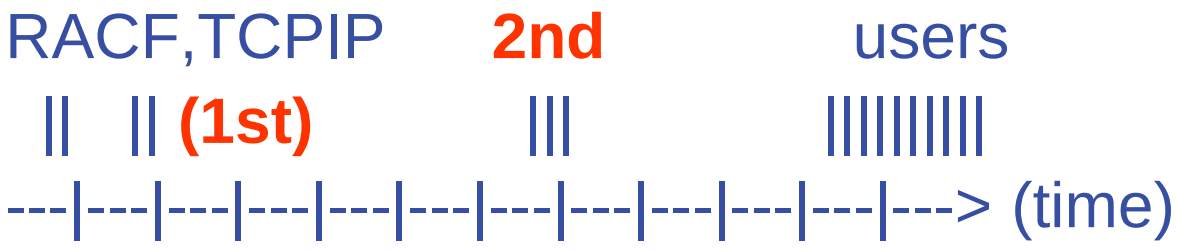
Improves deadline of first nn dispatch time slices.

- Default of 90 2 gives 90% boost on first time slice, 45% boost on 2nd dispatch time slice.
- Bias range is based on normalized share of highest current dispatchable user
  - If TCPIP is 10% share (scheduled at 10 time slices)
  - user is 1%, (scheduled at 100 time slices)
  - Moves user from 100 time slices delay to 18 time slice delay
- Use to improve performance of very interactive CMS users
- Linux polls, so is not considered “interactive”

Default IABIAS 90 2

- (RACF, tcpip rel share 10000, 10 users rel 100)
- (RACF, tcpip offset 21000/10000 -> 10.5ms)
- (user offset 21000/100 -> 1050 ms)

- 1st time slice offset = offset - (90% * delta) = 115 ms
- 2nd time slice offset = offset - (45% * delta) = 478ms
- 3rd time slice offset = offset = 1050 ms



Delta = difference of best deadline and offset

Analyzing Scheduler/Dispatcher

```
Report: ESASUM       System   z/VM   ESAMAP 4.1.1 01/16/1
Monitor initialized: 03/12/09 at st record analyzed: 03/12/09 05:01:00
--------------------------------------------------------------------------------
Variable Average Minimum Maximum Description
-------- ------- ------- ------- ------------------------------------------------
****************************SCHEDULER PARAMETERS************************
SRMBIASI      90                 Interactive bias intensity percent (SET SRM IAB
SRMBIASD       2                 Interactive bias duration (SET SRM IAB)
SRMTSLIC    5.00                 Minor time slice (ms) (SET SRM DSPSLICE)
SRMTSHOT    2.00                 Minor time slice (ms) for HOTSHOT users
SRMRSCTM  599.90  580.80  659.99 Reset interval (seconds)
SRMABSDL    52.0    48.0    55.0 Total absolute shares of VMDBKs in the dispatch
SRMRELDL     818     550    1900 Total relative shares of VMDBKs in the dispatch

SRMCDLDG       0       0       0 Loading users in dispatch list
SRMLDGUS       5                 Q1 page reads identifying loading user
SRMLDGCP       8                 Loading user capacity of system
SRMP1LDG     100                 Q1 loading user buffer percent (SET SRM LDUBUF)
SRMP2LDG      75                 Q2 loading user buffer percent (SET SRM LDUBUF)
SRMP3LDG      60                 Q3 loading user buffer percent (SET SRM LDUBUF)

SRMP1WSS     300                 Percent memory for E1/E2/E3 users (SET SRM STOR)
SRMP2WSS     300                 Percent memory for E2/E3 users (SET SRM STORBUF)
SRMP3WSS     300                 Percent memory for E3 users (SET SRM STORBUF)
SRMWSSMP    9998                 Maximum working set size percent (SET SRM MAXWSSIZ)

SRMXPCTG       0                 Percent Xstore used in SET SRM STORBUF calculation
SRML1DSP   32767                 Q1/Q2/Q3 Dispatch list size (SET SRM DSPBUF)
SRML2DSP   32767                 Q2/Q3 Dispatch list size (SET SRM DSPBUF)
SRML3DSP   32767                 Q3 Dispatch list size (SET SRM DSPBUF)

SRMEPNF1    2.00    2.00    2.00 E1 expansion factor
SRMEPNF2    2.00    2.00    2.00 E2 expansion factor
SRMEPNF3    2.00    2.00    2.00 E3 expansion factor
SRMLLCNT       0       0       0 Adds per minute to limit list
SRMCONLL       0       0       0 Count of users on limit list
```

VELOCITY SOFTWARE

ESAMON SHARE MACRO

```
/* calculate normalized share for user */
parse upper arg userid .

ADDRESS ESAMON 'EXTRACT FROM INTERVAL',
'FIELD RUNTIME NCPUS SYTSCG.SRMRELDL SYTSCG.SRMABSDL MTRSCH.SRMTSLIC'

ADDRESS ESAMON 'EXTRACT USER 'userid,
'FIELD USERDATA.VMDRELSH USERDATA.VMDABSSH'
mtrsch.srmtslic = mtrsch.srmtslic / 4096 / 1000  /* Convert to seconds */
sytscg.srmabsdl = sytscg.srmabsdl * 100 / 64 / 1024 /* Convert from internal
  format */

If SYTSCG.SRMABSDL > 99
  Then factor = 99 / sytscg.srmabsdl  ;   Else factor = 1
If userdata.vmdabssh > 0
 Then normshr = (userdata.vmdabssh * factor)
 Else Do; /* Absolute shares */
   If sytscg.srmreldl = 0 then sytscg.srmreldl = 100
   availshr = (100 – factor * sytscg.srmabsdl)
   normshr = (userdata.vmdrelsh / sytscg.srmreldl) * availshr
End;
say 'Share:' normshr'%'
say 'deadline:' mtrsch.srmtslic / (10 * normshr * ncpus ) 'Seconds'
```

```
ESAMON SHARE BARTON
Share: 1.90199309%
deadline: 0.262882133 Seconds Ready;
```

VELOCITY SOFTWARE

User Share Case Study

Installation had set TCPIP share from REL 3000 (default) to ABS 3%.

- **Good or bad?**

What would this do?

- **Relative share and absolute share normalized**
- **Need to know impact on normalized share**

What do we want?

- **TCPIP to have sufficient share to meet workload requirement**

TCPIP needs how much CPU?

45% of one CPU

during peak 15 minutes

```
Report: ESAUSP2      User Resource Rate Report
Monitor initialized: 02/07/07 at 00:00:05 on 2084 serial
-------------------------------------------------------
         <---CPU time--> <----Main Storage (pages)----->
UserID   <(Percent)> T:V <Resident> Lock <-----WSS----->
/Class   Total  Virt Rat Totl Activ  -ed Totl Activ  Avg
-------- ----- ----- --- ---- ----- ---- ---- ----- ----
13:05:00 188.8 178.4 1.1   2M 1559K 4782   2M 1753K  46K
 ***Key User Analysis ***
TCPIP     8.75  6.40 1.4 2722  2722  202  799   799  799
 ***User Class Analysis***
*Keys     0.36  0.32 1.1  527   527    3  558   558  186
*TheUsrs  4.42  4.18 1.1 141K  141K  339 165K  164K  13K
MPROUTE   0.20  0.19 1.1  319   319    1  315   315  472
-------------------------------------------------------
13:26:00 384.2 107.8 3.6   1M 1153K 4384   1M 1442K  37K
 ***Key User Analysis ***
TCPIP    44.83  6.20 7.2 2412  2412  202  621   621  621
 ***User Class Analysis***
*Keys    31.11  0.21 147  160   160    3  338   338  113
*TheUsrs 113.5  2.08  55  64K 64424  229  66K 66305 4973
DTCVSW1  17.69  0.00 .2M   17    17    0   16    16   24
DTCVSW2  16.02  0.00 .2M   17    17    0   16    16   24
```

Server Requirement Case Study

TCPIP used 45% of a processor at peak

- **LPAR has 10 processors**
- **TCPIP has a requirement of 5% of the system to meet peak requirement**
- **Is 3% absolute sufficient?**

What was 3000 relative in normalized terms?

- **Calculate normalizeShare =**
  - **(RelShare / SRMRELDL) * (100 – SRMABSDL) = ????**
- **Check ESASUM, Scheduler section**

```
Report: ESASUM       System Summary
------------------------------------------------------------------------------
                                      Minimum     Maximum      Std   Obs
Variable Average Minimum Maximum Date  Time  Date  Time     Dev Count Descript
-------- ------- ------- ------- ----- ----- ----- ----- ------ ----- --------
*********************************************SCHEDULER PARAMETERS********
SRMBIASI      90                                                 1394 Interact
SRMBIASD       2                                                 1394 Interact
SRMTSLIC    5.00                                                 1394 Minor ti

SRMTSHOT    2.00                                                 1394 Minor ti
SRMRSCTM  126.38    6.31  306.49 02/07 13:02 02/07 15:54  75.06  1368 Reset in
SRMABSDL      2.3       0     6.0 02/07 00:03 02/07 12:25   80.7  1370 Total ab
SRMRELDL     5296    1200    7070 02/07 15:53 02/07 24:00    523  1370 Total re
```

Dispatch / Eligible List Classes

There are three normal classes, one special class

- used to differentiate types of work
- Control thrashing based on queue
- Q1,Q2,Q3, and Q0

Each class has an associated Elapsed Time Slice (ETS),

- the amount of time a user may stay in the class

Trivial transactions defined as ending transaction in Q1

- ETS adjusted at every qdrop to maintain q1 levels
- Mostly meaningless unit of time (50ms-16sec)
- Defines queue stay, trivial transaction

Dispatch / Eligible List Classes - ETS

Elapsed time slice = .05 - 16 seconds.

- Varies dynamically,

ETS keeps 85% of INQUEUE users in Q1

- Q1 users: Inqueue < 1 ETS
- Q2 users: Inqueue < 7 ETS
- Q3 users: Inqueue > 7 ETS

Q1 size = (Q2 size / 6 + Q3 size / 48) / (.85 / .15)

ETS

- Does not keep '85% of the transactions trivial!
- is not useful to the performance analyst or for SLA!

Dispatch / Eligible List Class 0

Class 0 (Special case, Not held on E–List)

QUICKDSP: set by installation, ETS is the same as Class 2, dispatch priority reprioritized after 8 Q1ETS (1 Q2ETS)

Lockshot: User is holding a lock and stays in class 0 until lock is released. User is treated as QUICKDSP with regard to eligible list.

Hotshot: User is already in queue and interacts with the terminal. Dispatch time slice is a hotshot time slice. Hotshot bias is 90 or 95%

- (users that issue #CP Q T for example during long transaction

Dispatches System VMDBK first

Dispatches user with lowest dispatch deadline priority

- CP System Work
- CP User work
- Users/Servers

Gives a user one dispatch time slice

- Unit of time virtual machine is dispatched
- SET SRM DSPSLICE
- 1-99ms, Default 5ms

Does not care if user is Q1, Q2, or Q3

Processor Local Dispatch Vector

- One per each local processor
- One additional for master

Dispatchable users picked by dispatcher and put on PLDV

- Requires lock, so multiple users “picked”
- “steals” is the “pldv reshuffle”

Report: ESAPLDV Processor Local Dispatch Vector Activity Velocity Software, Inc.

Time	<----Users-----> Logged	Actv	In Q	Tran /sec	CPU	<VMDBK Moves/sec> Steals	To Master	<--------PLDV Lengths-------> Avg	Max	Mstr	MstrMax	%Empty	Dispatcher Long Paths
13:16:04	788	274	23.7	19.0	0	126.7	334.3	0.8	2.0	0.3	1.0	44.4	977.4
					1	69.5	0	0.1	2.0	.	.	92.5	357.8
					2	64.7	0	0.1	2.0	.	.	91.9	315.4
					3	69.9	0	0.1	2.0	.	.	91.1	340.6
					4	63.2	0	0.1	2.0	.	.	93.5	302.8
					5	74.5	0	0.1	2.0	.	.	91.6	383.3
System:						468.5	334.3	1.4	12.0	0.3	1.0	504.9	2677.2

Virtual Multi-processors:

- Both virtual processors must go idle for server to drop from queue
- Analysis required.

Current JDK polls every 10ms

Current polling issues impact:

- WAS/Java,
- DOMINO,
- Tivoli Applications
- Oracle
- SAP….

Storage management changes in 6.3 make polling less relevant

Tracing User

Enable Scheduler domain for user

Record Raw Monitor data for analysis interval

Run ESAMAP against raw data

Set ESAMAP Option:

- TRACE.USER = ’userid’

ESATUNA LISTING

- QDrops
- QAdds
- Transaction Details
- Seek Details

VELOCITY SOFTWARE

Transaction Time Line

When analyzing a performance problem, build a timeline

A CMS “short” transaction timeline

```
07:11:00.459272 Scheduler Data (SCLAEL), Add User to Eligible List: 1
07:11:00.459436 Scheduler Data (SCLADL), Add User to Dispatch List: 1
Dispatch lists: q0: 1 q1: 1 q2: 0 q3: 1
07:11:00.461404 Scheduler Data (SCLRDC), Read Complete From 0004
07:11:00.464087 Scheduler Data (SCLWRR), Write Response To 0004
07:11:01.924552 Scheduler Data (SCLDDL), Drop User from Dispatch List
```

1. Add user to Eligible List (SCLAEL)
2. Move user to dispatch list SCLADL)
3. Read input data from screen (SCLRDC)
4. Write input data back to screen (SCLWRR)
5. Drop user from dispatch list (SCLDDL)

ESATUNA Report

Very large

Time stamped

Details of activity

(Transactions cut at beginning of next transaction)

```
07:10:00.878347 Sample Data (USEACT), Resources used:
07:10:00.878506 Sample Data (USEINT), Delay Analysis
07:10:08.842449 Event Data (USETRE) response times:
Response time (seconds): 1.827
InQueue time (seconds): 2.224
Think time (seconds): 27.5
07:10:08.842501 Event Data (USEATE), Resources used:
07:10:08.842584 Event Data (USEITE), Wait Analysis:
07:11:00.459018 Event Data (USETRE) response times:
Response time (seconds): 0.122
InQueue time (seconds): 2.018
Think time (seconds): 49.6
07:11:00.459067 Event Data (USEATE), Resources used:
User operating in ESA mode.
User has Relative Share of: 100
Processor Consumption (CPU Seconds)
TotCPUTm 0.02020 VirtCPU 0.00269
Storage Consumption (Pages)
PagesRes 235.000 WSS Size 235.000 VM Size 2048.00
Paging Activity (Counts)
NonPfPgs 43.0000
Spooling Activity (Counts)
SplPages 55.0000
Non-DASD Virtual I/O (Counts)
Cons I/O 2.00000
07:11:00.459144 Event Data (USEITE), Wait Analysis:
InQueue State Sample Counts
InQueue 2.00000 TstIdle 2.00000
InQueue Percent State Analysis
InQueue 3.84615 TstIdle 100.000
Queue Analysis
Pct Q1 100.000
Time slice used up in Q1 1 times.
```

Tracing Linux User

```
17:57:45.583123 VCPUad: 00 Scheduler Data (SCLAEL), Add User to Eligible List: 1
17:57:45.583126 VCPUad: 00 Scheduler Data (SCLADL), Add User to Dispatch List: 1
 Dispatch lists: q0:   4 q1:   5 q2:   0 q3:  27
 Dispatch Priority(Original): 2833969.0000
  Dispatch Priority(Revised): 2833967.0000
 Elapsed time slice:       0.4658 Required thruput:     422.0000
  VMDIABIA: Interactive Bias in effect
17:57:45.773364 VCPUad: 01 Scheduler Data (SCLAEL), Add User to Eligible List: 1
17:57:45.773367 VCPUad: 01 Scheduler Data (SCLADL), Add User to Dispatch List: 1
 Dispatch lists: q0:   4 q1:   6 q2:   2 q3:  27
 Dispatch Priority(Original): 2833969.0000
  Dispatch Priority(Revised): 2833967.0000
 Elapsed time slice: 1799808.0000 Required thruput:     455.0000
  VMDIABIA: Interactive Bias in effect
17:57:45.773416 VCPUad: 01 Scheduler Data (SCLDDL), Drop User from Dispatch List
 User requires scheduler intervention, VMDSACTL = 00001000
  VMDIDROP: Drop from DISP Immediately
  VMDIABIA: Interactive Bias in effect
17:57:46.048896 VCPUad: 00 Scheduler Data (SCLDDL), Drop User from Dispatch List
 User requires scheduler intervention, VMDSACTL = 00000001
  VMDRSCEL: VMDBK exceeded limits of controlled resource
User requires scheduler intervention, VMDSACTX = 00010000
  VMDESEND: Elapsed Timeslice Exceeded
  VMDIABIA: Interactive Bias in effect
```