The Scheduler and Dispatcher

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

"If you can't Measure it,
I am Just Not Interested ™"

Scheduler and Dispatcher

Objectives

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

What is important?

- When users / servers get dispatched
- Prioritizing work (Share values)
- How long are they dispatched for (time slice)

Functional Overview

The Scheduler

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

The Dispatcher

- Selects a user to run
- Dispatches units of work

Command Overview

Scheduler affected by:

~~SET SRM STORBUF~~	~~(control storage utilization)~~
~~SET SRM DSPBUF~~	~~(control processor utilization)~~
~~SET SRM LDUBUF~~	~~(control paging device utilization)~~
SET SRM DSPSLICE	(time slize, default 5ms)
SET SRM IABIAS	(bias interactive users)
SET SHARE	**(guarantee a share of CPU)**
~~SET QUICKDSP~~	~~(ignore STORBUF, DSPBUF, LDUBUF)~~

Dispatcher affected by:

- SET SRM DSPSLICE
- Secret command

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 doc)

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

Use Relative shares for users

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~~

Dispatch List

- Users contending for resources now
- Kept in priority order
- Processor Local Dispatch Vector
- Linux always here

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

- Virtual machine is done working, or ETS has expired

Dormant List

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

Transaction Class 1 / 2 / 3

- Workload classification

Dispatch List Definition

Class 1 (Interactive)

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

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, batch, guests)

- Entry after one stay (8x ETS) in Class 2
- Q3ETS is 48x the Class 1 ETS (fixed multiple)
- All Linux guests run here all the time

Dispatch List Workloads

Class 1 (Interactive)

- CMS Users
- Idle Linux users with timer patch

Class 2 (Non–Interactive)

- Long running CMS users
- Medium Linux transactions with timer patch

Class 3 (Long–running, batch, guests)

- Z/OS, TPF
- Linux (“Idle” or active”)

Queue Analysis

Example, Linux users in Queue 3

```
Report: ESAUSRQ      User Queue and Load Analysis
------------------------------------------------------------------------
         <-----------User Load------------>      <-----------Average Num
UserID   Logged  Non-           Disc- Total  Tran <-------Dispatch List--
/Class       on  Idle  Active   conn InQue  /min    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
```

High Level Scheduler

Fair Share Scheduler (Wheeler scheduler):

- Allows prioritization of work
- ~~Determines work "Eligibility"~~
- ~~Protects workload from resource over commitment using the "Eligible List"~~
- 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~~

Excess Share Analysis

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
 1 of 3  User Percent Utilization                CLASS * USER
                              <-------Main Storage-------->
         UserID   <Processor> <Resident->  Lock <-WSSize-->
Time     /Class   Total  Virt Total  Actv   -ed Total  Actv
-------- -------- ----- ----- ----- ----- ----- ----- -----
00:11:00 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
```

Excess Share Analysis

We now give ROBLX2 a RELATIVE 200 share

- Because that is a more important service
- (There is 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
```

Excess Share Analysis

Now for the experiment – Set shares “correctly”

- 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

```
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
```

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)
- 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?

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
```

Calculation of Normalized Share

If SRMABSDL is less than 100%

- Normalized share equals Absolute Share
- Relative Share users get: (100 - SRMABSDL) * (relative share / SRMRELDL)

If SRMABSDL is greater than 99,

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

Normalized shares are percentages of the CPU resource

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

Setting Deadline

Deadline time of day = current TOD + offset

Offset = (DSPSLICE / Normalized share) * bias

```
|---|---|---|---|---|---|---|---|---|---|---|---|--->(time)
(ATOD)
                                   users
                                  ||||||||||
---|---|---|---|---|---|---|---|---|---|---|---> (time)

TCPIP                             users
    ||                             ||||||||||
---|---|---|---|---|---|---|---|---|---|---|---> (time)
```

Dispatcher takes users in order by time from sorted deadline list

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 30 minutes

Sample Deadlines

Sample Deadlines:

Example (50 users, 4 vcpu each)

- RACF has relative share 3000
- TCPIP has relative share 3000
- User has relative share 100
- DSPSLICE = 5ms
- SRMRELDL = 25000 (typical)

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

- Divide by 4 vCPU = .1%
- CPU delivery rate = 5ms / .001 or 5ms per 5 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)

```
RACF,TCPIP                  users
||                       ||||||||||
---|---|---|---|---|---|---|---|---|---|---|---> (time)
TOD
```

Example 2:

Change TCPIP/RACF share to ABSOLUTE 20

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

```
RACF,TCPIP    users
 ||          ||||||||||
---|---|---|---|---|---|---|---|---|---|---|---> (time)
TOD
```

Sample Deadlines - Comparison

Did it make a difference to TCPIP/ RACF 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 high for some users impact others?

- Only when large CPU consumers exist
- IBM does not let looping users on their benchmark systems

Recommend low ABS shares when appropriate for servers

Deadline – Behind Schedule?

Behind schedule?

```
behind  TCPIP                    users
|              ||                  ||||||||
--^-----+---|---|---|---|---|---|---|---|---|---|---|---> (time)
     TOD
```

Do users get behind?

- PLSEFRC1 for Q1
- PLSEFRC2 for Q2
- PLSEFRC3 for Q3

Reported on ESASUM - No it doesn't happen

Interactive Bias

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 users

DOES NOT IMPROVE LINUX Guests

Interactive Bias

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

```
RACF/TCPIP     2nd           users
  ||  || (1st)        |||           ||||||||||
---|---|---|---|---|---|---|---|---|---|---|---> (time)
TOD
```

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
```

zMON 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;
```

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

User Share Case Study

```
Report: ESAUSP2      User Resource Rate Report
---------------------------------------------------------
         <---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
---------------------------------------------------------
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
```

TCPIP needs how much CPU?

- 45% of one CPU
- During peak 15 minutes

Dispatcher

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 (with SMT – Default 10ms)

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

Dispatcher

Processor Local Dispatch Vector (PLDV)

- One per each local processor
- One additional for master

Dispatchable users picked by dispatcher and put on PLDV

- Requires lock, so multiple users “picked”
- Affinity blocks steals for 50ms

```
Report: ESAPLDV      Processor Local Dispatch Vector Activity      Velocity Software, Inc.
----------------------------------------------------------------------------------------------------
         <----Users----->   Tran       <VMDBK Moves/sec>  <--------PLDV Lengths-------> Dispatcher
Time     Logged Actv In Q   /sec  CPU  Steals  To Master   Avg  Max Mstr MstrMax %Empty 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
```

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

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)

Tracing User

```
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
Pct Q1 100.00
```

ESATUNA Report

- Very large
- Time stamped
- Details of activity
- (Transactions cut at beginning of next transaction)

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
```