

Scheduler and Dispatcher

- Barton@VelocitySoftware.com
- [HTTP://VelocitySoftware.com](http://VelocitySoftware.com)

“If you can’t Measure it,
I am Just Not Interested™”

Objectives

- Understanding Scheduler / Dispatcher
- How SHAREs affect users

What is important?

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

The Scheduler

- Calculates “deadline” priorities
- Sorts the dispatch list accordingly

The Dispatcher

- Selects a user to run
- Dispatches units of work

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

Fair Share Scheduler (Wheeler scheduler):

- Allows prioritization of work
- 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

Deadline priority is a “target” time of day

- Deadline = TOD + **DelayFactor**
- 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)
- Deadline is calculated after every dispatch time slice is completed.

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 it was 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 TSTLN1  32.39  32.38  15862  15862    11  15536  15536
          TSTLN2  32.12  32.11  66136  66136   259  78478  78478
          TSTLN1  32.02  32.01  38219  38219   176  37790  37790
          TST2LV   0.01   0.00   2246   2246    0    2246   2246
```

We now give TSTLX2 a RELATIVE 200 share

- Because that is a more important service
- 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 TSTLX2  68.71 68.68 66211 66211 258 78478 78478
        TSTLX1  14.00 14.00 38245 38245 256 37790 37790
        TSTLNX1 13.99 13.99 15879 15879 11 15536 15536
        TST2LV  0.01 0.00 2246 2246 0 2246 2246
  
```

Now for the experiment – Set shares “correctly”

- Convert TCPIP, SSL, RACF 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----->								
Time	UserID /Class	<Processor> Total	<Resident-> Virt	Lock -ed	<-WSSize--> Total	Actv	Actv	Actv
00:20:00	TSTLX2	48.39	48.37	67141	67141	292	80047	80047
	TSTLNX1	24.19	24.19	16168	16168	11	15536	15536
	TSTLX1	24.19	24.18	39006	39006	241	37790	37790
	TST2LV	0.01	0.00	2246	2246	0	2246	2246

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

Variable	Average	Minimum	Maximum	Description
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

If SRMABSDL is less than 100%

- Normalized share equals Absolute Share
- Relative Share users get:
 $(100 - \text{SRMABSDL}) \times (\text{relative share} / \text{SRMRELDL})$

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

Deadline time of day = current TOD + offset

Offset = (DSPSLICE / Normalized share) * bias



users



TCPIP

users

"new users"



Dispatcher takes users in order by time from sorted deadline list

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.

Example 1:

- TCPIP offset 2.5 dspslice (Share 3000)
- Users offset 250 dspslice (1.25 seconds)



Example 2: Change TCPIP/RACF share to ABSOLUTE 10

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



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