Case Study 1

Linux Server Experiencing Timeouts

Case Study Summary

Velocity Software solves performance problems.

- **As a valued customer, we want to pass this knowledge on to you.**
- **The following is a case study of a solved real-life performance issue.**
- **This case study will show:**
    - **The problem as reported by users**
    - **The problem observations**
    - **What was found in the Velocity Software data**
    - **What was suggested to the customer**
    - **If provided, follow up from the customer**

The Problem

The Problem:

A real-time transaction system running on a Linux server was experiencing timeouts

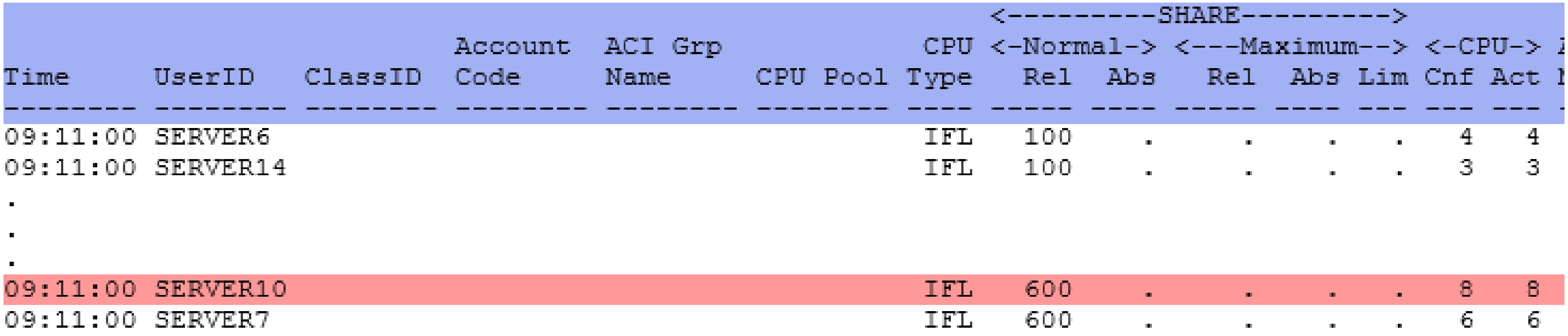
Problem Observations:

- SERVER10 running on LPAR2 was showing timeouts
- LPAR2 had 7 real engines/14 threads with SMT enabled
- SERVER10's virtual machine had 8 virtual CPUs with a relative share of 600
- When SMT is enabled, the default dispatch time slice changed from 5ms to 10ms

What the Data Showed (Configuration data)

ESAUSRC – User Configuration showed:

- SERVER10 had 8 configured and active virtual CPUs
- SERVER10 had a Relative Share setting of 600



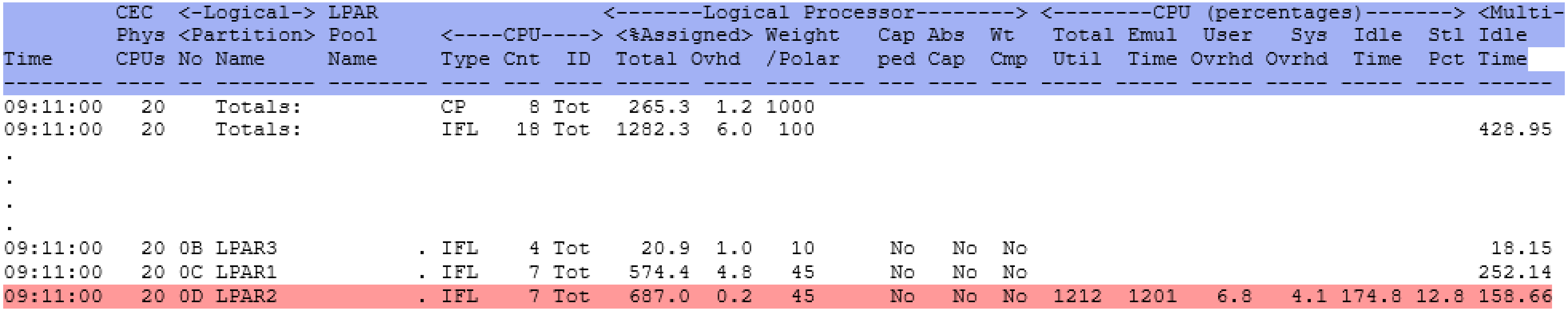
```
                                                           <---------SHARE--------->
                           Account  ACI Grp                 CPU <-Normal-> <---Maximum--> <-CPU-> A
Time     UserID   ClassID  Code     Name     CPU Pool Type   Rel  Abs   Rel  Abs Lim Cnf Act I
-------- -------- -------- -------- -------- -------- ---- ----- ---- ----- ---- --- --- --- -
09:11:00 SERVER6                                      IFL    100    .     .    .   .   4   4
09:11:00 SERVER14                                     IFL    100    .     .    .   .   3   3
.
.
.
09:11:00 SERVER10                                     IFL    600    .     .    .   .   8   8
09:11:00 SERVER7                                      IFL    600    .     .    .   .   6   6
```

Relative share is divided by active vCPUs so for SERVER10, each vCPU only got a share of 75 instead of 100 (default) or 600 (desired).

What the Data Showed (Configuration data)

ESALPAR – Logical Partition Analysis showed:

- LPAR2 had 7 IFLs assigned
- LPAR2 had a weight of 45 (out of 100)
- LPAR2 had SMT enabled



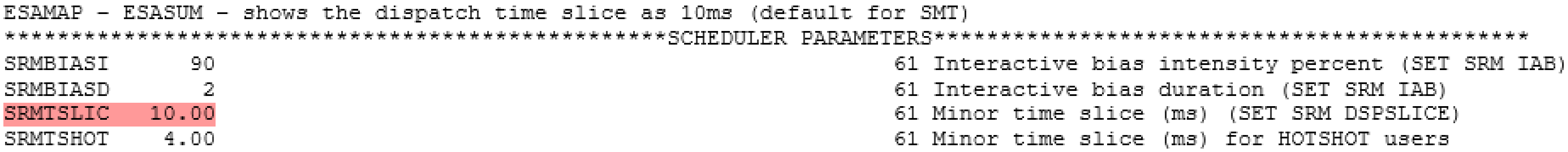
```
         CEC  <-Logical-> LPAR                      <-------Logical Processor---------> <---------CPU (percentages)--------> <Multi-
         Phys <Partition> Pool     <----CPU----> <%Assigned> Weight   Cap Abs  Wt   Total Emul  User    Sys  Idle  Stl Idle
Time     CPUs No Name     Name     Type Cnt   ID Total  Ovhd /Polar   ped Cap  Cmp  Util  Time Ovrhd Ovrhd  Time  Pct Time
-------- ---- -- -------- -------- ---- --- ---- ------ ---- ------ --- --- ---- --- ----- ----- ----- ----- ----- ---- ------
09:11:00   20    Totals:           CP     8 Tot  265.3   1.2   1000
09:11:00   20    Totals:           IFL   18 Tot 1282.3   6.0    100                                                      428.95
.
.
.
.
09:11:00   20 0B LPAR3           . IFL    4 Tot   20.9   1.0     10      No   No   No                                   18.15
09:11:00   20 0C LPAR1           . IFL    7 Tot  574.4   4.8     45      No   No   No                                  252.14
09:11:00   20 0D LPAR2           . IFL    7 Tot  687.0   0.2     45      No   No   No  1212  1201   6.8   4.1 174.8 12.8 158.66
```

The LPAR2 processor had 7 IFLs that were approximately 98% busy.

What the Data Showed (Configuration data)

ESASUM – System Summary showed:

- The Dispatch Time Slice was 10ms



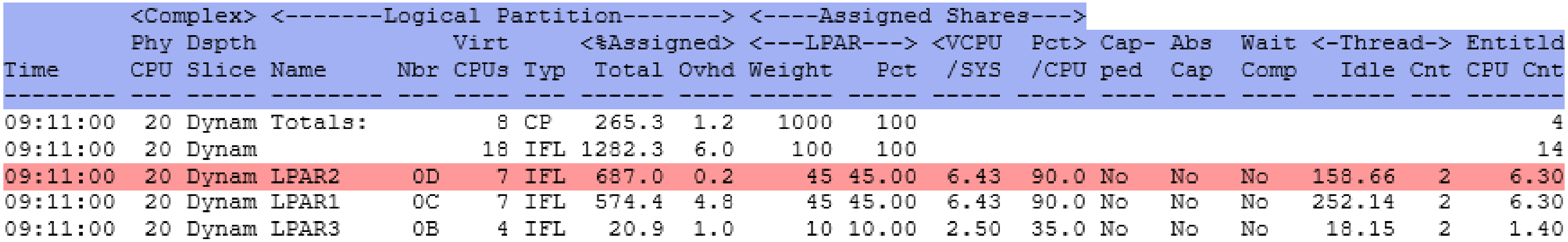
```
ESAMAP - ESASUM - shows the dispatch time slice as 10ms (default for SMT)
**************************************************SCHEDULER PARAMETERS*********************************************
SRMBIASI       90                                                  61 Interactive bias intensity percent (SET SRM IAB)
SRMBIASD        2                                                  61 Interactive bias duration (SET SRM IAB)
SRMTSLIC    10.00                                                  61 Minor time slice (ms) (SET SRM DSPSLICE)
SRMTSHOT     4.00                                                  61 Minor time slice (ms) for HOTSHOT users
```

The Dispatch Time Slice has a default setting of 5ms. When SMT is enabled (which it was here) it becomes 10ms. This works better for batch environments, not online transaction environments.

What the Data Showed (Utilization data)

ESALPARS – Logical Partition Analysis Summary showed:

- LPAR2 is entitled to 6.3 engines but was using more (6.8)



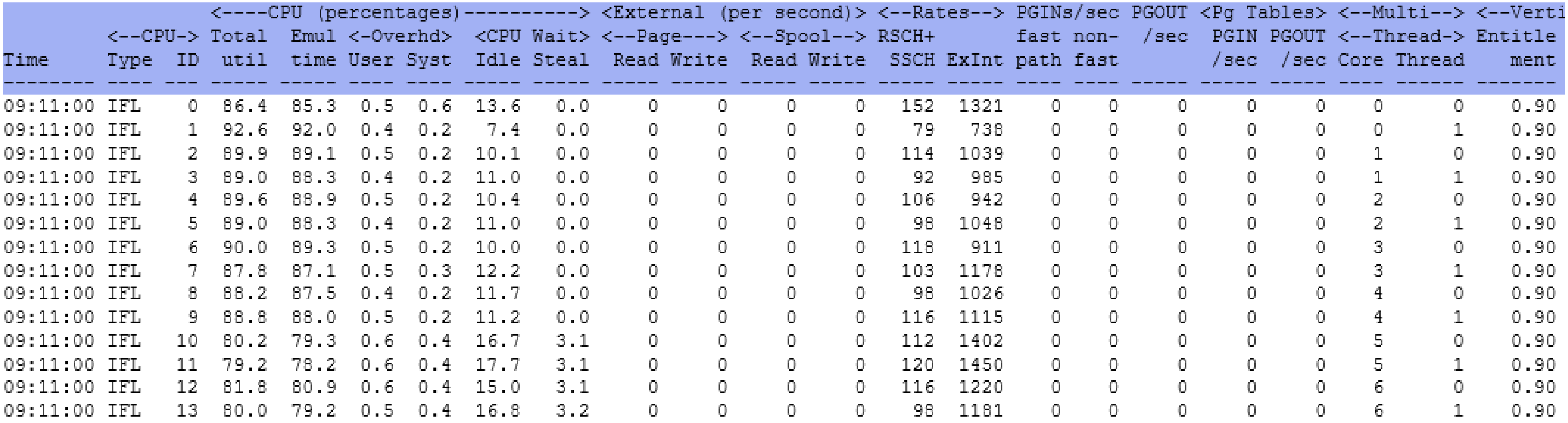
```
         <Complex> <-------Logical Partition-------> <----Assigned Shares--->
         Phy Dspth                 Virt     <%Assigned> <---LPAR---> <VCPU  Pct> Cap- Abs  Wait <-Thread-> Entitld
Time     CPU Slice Name      Nbr CPUs Typ  Total Ovhd Weight   Pct  /SYS  /CPU ped  Cap  Comp   Idle Cnt CPU Cnt
-------- --- ----- --------- --- ---- --- ------ ---- ------ ----- ----- ----- ---- ---- ---- ------ --- -------
09:11:00  20 Dynam Totals:          8 CP   265.3  1.2   1000   100                                             4
09:11:00  20 Dynam                 18 IFL 1282.3  6.0    100   100                                            14
09:11:00  20 Dynam LPAR2      0D    7 IFL  687.0  0.2     45 45.00  6.43  90.0 No   No   No   158.66   2    6.30
09:11:00  20 Dynam LPAR1      0C    7 IFL  574.4  4.8     45 45.00  6.43  90.0 No   No   No   252.14   2    6.30
09:11:00  20 Dynam LPAR3      0B    4 IFL   20.9  1.0     10 10.00  2.50  35.0 No   No   No    18.15   2    1.40
```

The LPAR2 processor was running at approximately 98% during the time of the issue.

What the Data Showed (Utilization data)

ESACPUU – CPU Utilization Analysis showed:

- LPAR2 had 14 threads that all had high utilization

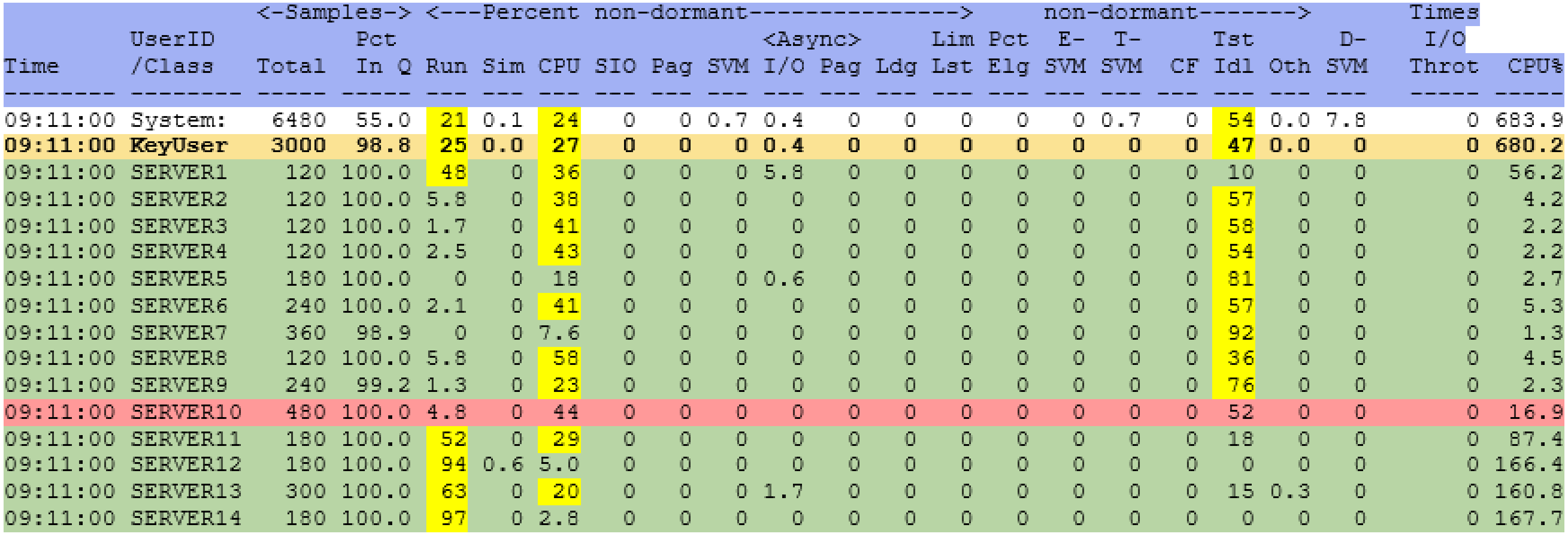


```
                  <----CPU (percentages)-----------> <External (per second)> <--Rates--> PGINs/sec PGOUT <Pg Tables> <--Multi--> <--Verti
         <--CPU-> Total  Emul <-Overhd>  <CPU Wait> <--Page---> <--Spool--> RSCH+       fast non-  /sec  PGIN PGOUT <--Thread-> Entitle
Time     Type  ID  util  time User Syst  Idle Steal  Read Write  Read Write  SSCH ExInt path fast        /sec  /sec Core Thread     ment
-------- ---- --- ----- ----- ---- ---- ----- ----- ----- ----- ----- ----- ----- ----- ---- ---- ----- ----- ----- ---- ------ --------
09:11:00 IFL    0  86.4  85.3  0.5  0.6  13.6   0.0     0     0     0     0   152  1321    0    0     0     0     0    0      0     0.90
09:11:00 IFL    1  92.6  92.0  0.4  0.2   7.4   0.0     0     0     0     0    79   738    0    0     0     0     0    0      1     0.90
09:11:00 IFL    2  89.9  89.1  0.5  0.2  10.1   0.0     0     0     0     0   114  1039    0    0     0     0     0    1      0     0.90
09:11:00 IFL    3  89.0  88.3  0.4  0.2  11.0   0.0     0     0     0     0    92   985    0    0     0     0     0    1      1     0.90
09:11:00 IFL    4  89.6  88.9  0.5  0.2  10.4   0.0     0     0     0     0   106   942    0    0     0     0     0    2      0     0.90
09:11:00 IFL    5  89.0  88.3  0.4  0.2  11.0   0.0     0     0     0     0    98  1048    0    0     0     0     0    2      1     0.90
09:11:00 IFL    6  90.0  89.3  0.5  0.2  10.0   0.0     0     0     0     0   118   911    0    0     0     0     0    3      0     0.90
09:11:00 IFL    7  87.8  87.1  0.5  0.3  12.2   0.0     0     0     0     0   103  1178    0    0     0     0     0    3      1     0.90
09:11:00 IFL    8  88.2  87.5  0.4  0.2  11.7   0.0     0     0     0     0    98  1026    0    0     0     0     0    4      0     0.90
09:11:00 IFL    9  88.8  88.0  0.5  0.2  11.2   0.0     0     0     0     0   116  1115    0    0     0     0     0    4      1     0.90
09:11:00 IFL   10  80.2  79.3  0.6  0.4  16.7   3.1     0     0     0     0   112  1402    0    0     0     0     0    5      0     0.90
09:11:00 IFL   11  79.2  78.2  0.6  0.4  17.7   3.1     0     0     0     0   120  1450    0    0     0     0     0    5      1     0.90
09:11:00 IFL   12  81.8  80.9  0.6  0.4  15.0   3.1     0     0     0     0   116  1220    0    0     0     0     0    6      0     0.90
09:11:00 IFL   13  80.0  79.2  0.5  0.4  16.8   3.2     0     0     0     0    98  1181    0    0     0     0     0    6      1     0.90
```

When SMT is enabled, z/VM shows two threads for every CPU so 7 CPUs show as 14 threads, all of which were highly utilized.

ESAXACT – Transaction Delay Analysis showed:

- SERVER10 is waiting on CPU
- Other servers are also waiting on CPU

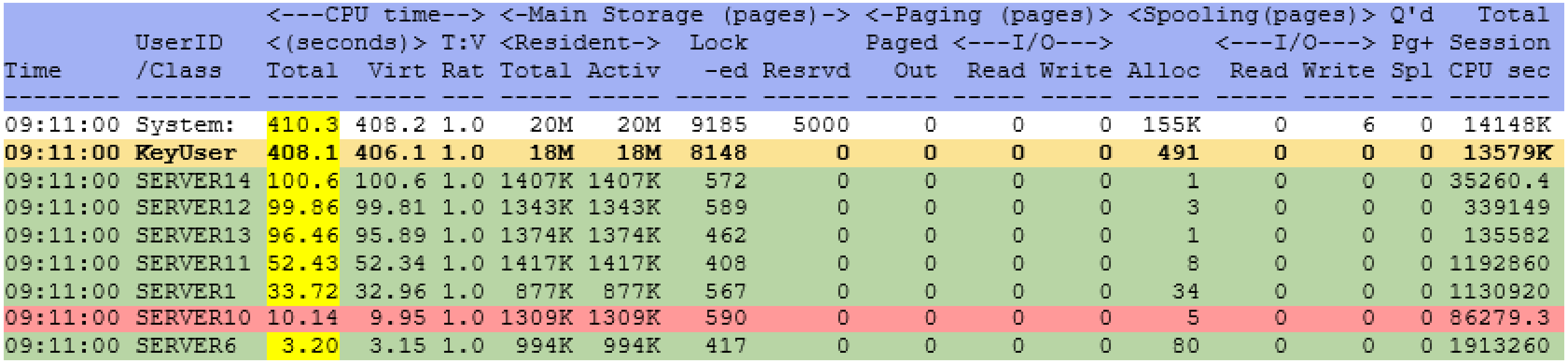


```
                   <-Samples-> <---Percent non-dormant--------------->    non-dormant------->         Times
         UserID           Pct                          <Async>         Lim Pct  E-  T-     Tst      D-      I/O
Time     /Class    Total  In Q Run Sim CPU SIO Pag SVM I/O Pag Ldg Lst Elg SVM SVM  CF Idl Oth SVM     Throt   CPU%
-------- --------- ----- ----- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---    ------ ------
09:11:00 System:    6480  55.0  21 0.1  24   0   0 0.7 0.4   0   0   0   0   0 0.7   0  54 0.0 7.8         0  683.9
09:11:00 KeyUser    3000  98.8  25 0.0  27   0   0   0 0.4   0   0   0   0   0   0   0  47 0.0   0         0  680.2
09:11:00 SERVER1     120 100.0  48   0  36   0   0   0 5.8   0   0   0   0   0   0   0  10   0   0         0   56.2
09:11:00 SERVER2     120 100.0 5.8   0  38   0   0   0   0   0   0   0   0   0   0   0  57   0   0         0    4.2
09:11:00 SERVER3     120 100.0 1.7   0  41   0   0   0   0   0   0   0   0   0   0   0  58   0   0         0    2.2
09:11:00 SERVER4     120 100.0 2.5   0  43   0   0   0   0   0   0   0   0   0   0   0  54   0   0         0    2.2
09:11:00 SERVER5     180 100.0   0   0  18   0   0   0 0.6   0   0   0   0   0   0   0  81   0   0         0    2.7
09:11:00 SERVER6     240 100.0 2.1   0  41   0   0   0   0   0   0   0   0   0   0   0  57   0   0         0    5.3
09:11:00 SERVER7     360  98.9   0   0 7.6   0   0   0   0   0   0   0   0   0   0   0  92   0   0         0    1.3
09:11:00 SERVER8     120 100.0 5.8   0  58   0   0   0   0   0   0   0   0   0   0   0  36   0   0         0    4.5
09:11:00 SERVER9     240  99.2 1.3   0  23   0   0   0   0   0   0   0   0   0   0   0  76   0   0         0    2.3
09:11:00 SERVER10    480 100.0 4.8   0  44   0   0   0   0   0   0   0   0   0   0   0  52   0   0         0   16.9
09:11:00 SERVER11    180 100.0  52   0  29   0   0   0   0   0   0   0   0   0   0   0  18   0   0         0   87.4
09:11:00 SERVER12    180 100.0  94 0.6 5.0   0   0   0   0   0   0   0   0   0   0   0   0   0   0         0  166.4
09:11:00 SERVER13    300 100.0  63   0  20   0   0   0 1.7   0   0   0   0   0   0   0  15 0.3   0         0  160.8
09:11:00 SERVER14    180 100.0  97   0 2.8   0   0   0   0   0   0   0   0   0   0   0   0   0   0         0  167.7
```

The ESAXACT data/report is one of the best ways to see what resources are holding up system activity.

ESAUSR2 – User Resource Utilization showed:

- SERVER10 is getting less CPU than other servers
- (The customer said these other servers were running more batch-like applications)



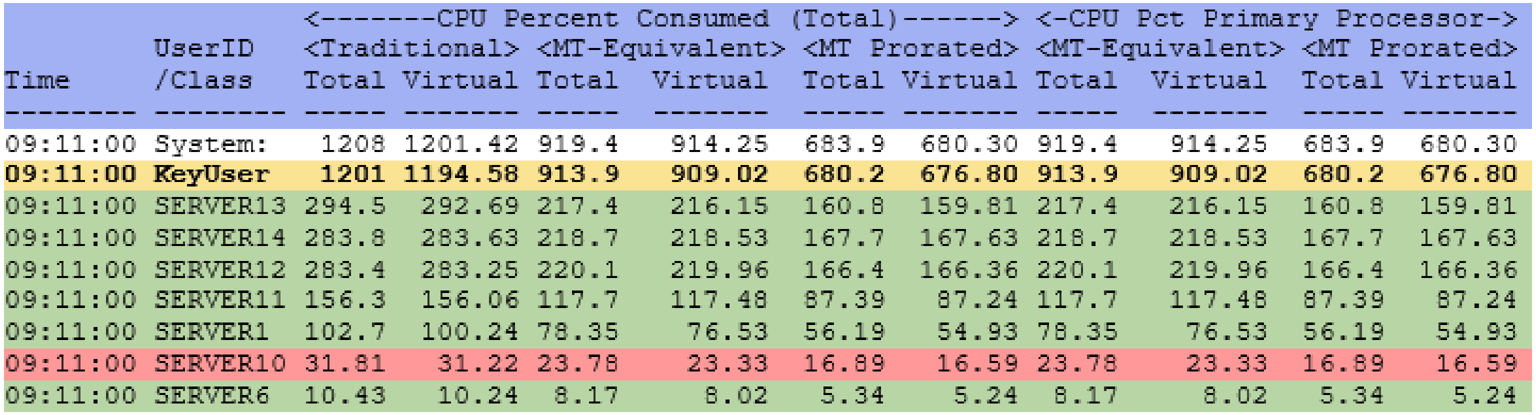
```
                   <---CPU time--> <-Main Storage (pages)-> <-Paging (pages)> <Spooling(pages)> Q'd   Total
         UserID    <(seconds)> T:V <Resident->  Lock        Paged <---I/O--->       <---I/O---> Pg+ Session
Time     /Class    Total  Virt Rat Total Activ   -ed Resrvd   Out  Read Write Alloc  Read Write Spl CPU sec
-------- --------- ----- ----- --- ----- ----- ----- ------ ----- ----- ----- ----- ----- ----- --- -------
09:11:00 System:   410.3 408.2 1.0   20M   20M  9185   5000     0     0     0  155K     0     6   0  14148K
09:11:00 KeyUser   408.1 406.1 1.0   18M   18M  8148      0     0     0     0   491     0     0   0  13579K
09:11:00 SERVER14  100.6 100.6 1.0 1407K 1407K   572      0     0     0     0     1     0     0   0 35260.4
09:11:00 SERVER12  99.86 99.81 1.0 1343K 1343K   589      0     0     0     0     3     0     0   0  339149
09:11:00 SERVER13  96.46 95.89 1.0 1374K 1374K   462      0     0     0     0     1     0     0   0  135582
09:11:00 SERVER11  52.43 52.34 1.0 1417K 1417K   408      0     0     0     0     8     0     0   0 1192860
09:11:00 SERVER1   33.72 32.96 1.0  877K  877K   567      0     0     0     0    34     0     0   0 1130920
09:11:00 SERVER10  10.14  9.95 1.0 1309K 1309K   590      0     0     0     0     5     0     0   0 86279.3
09:11:00 SERVER6    3.20  3.15 1.0  994K  994K   417      0     0     0     0    80     0     0   0 1913260
```

When an important server that is running online transactions is waiting on servers running batch, the user's performance will suffer.

ESAUSR5 – User SMT CPU Percent Utilization showed:

The ESAUSR5 information showed:

- The same information as ESAUSR2 but from an SMT perspective.



```
                   <--------CPU Percent Consumed (Total)------> <-CPU Pct Primary Processor->
         UserID    <Traditional> <MT-Equivalent> <MT Prorated> <MT-Equivalent> <MT Prorated>
Time     /Class    Total Virtual Total  Virtual  Total Virtual Total  Virtual  Total Virtual
-------- --------  ----- ------- -----  -------  ----- ------- -----  -------  ----- -------
09:11:00 System:    1208 1201.42 919.4   914.25  683.9  680.30 919.4   914.25  683.9  680.30
09:11:00 KeyUser    1201 1194.58 913.9   909.02  680.2  676.80 913.9   909.02  680.2  676.80
09:11:00 SERVER13 294.5   292.69 217.4   216.15  160.8  159.81 217.4   216.15  160.8  159.81
09:11:00 SERVER14 283.8   283.63 218.7   218.53  167.7  167.63 218.7   218.53  167.7  167.63
09:11:00 SERVER12 283.4   283.25 220.1   219.96  166.4  166.36 220.1   219.96  166.4  166.36
09:11:00 SERVER11 156.3   156.06 117.7   117.48  87.39   87.24 117.7   117.48  87.39   87.24
09:11:00 SERVER1  102.7   100.24 78.35    76.53  56.19   54.93 78.35    76.53  56.19   54.93
09:11:00 SERVER10 31.81    31.22 23.78    23.33  16.89   16.59 23.78    23.33  16.89   16.59
09:11:00 SERVER6  10.43    10.24  8.17     8.02   5.34    5.24  8.17     8.02   5.34    5.24
```

This has the same outcome, when an important server that is running online transactions is waiting on servers running batch, the user's performance will suffer.

Performance Enhancement Suggestions:

1 – Change the SHARE setting for SERVER10

- This server is running online transactions
  - It needs to have priority over batch

- The current setting was REL 600 (for 8 vCPUs)
  - That only gave each vCPU REL 75 (the default is 100)

- Update the setting to REL 1200 – would double its current SHARE and make it 50% better than batch
  - If not using all of its SHARE, the CPU would be free for others to use but would allow SERVER10 more processing power when needed

Performance Enhancement Suggestions:

2 – Use Resource Pools

- Resource pools can be used to set resource restrictions by group
  - Batch and online groups can be created (for example)

- Resource pools can be scheduled to allow resource cooperation
  - Resource pools can be scheduled to allow online transaction servers more processing power during the day and batch more processing power at night

Velocity Software's z/PRO is a very convenient way to schedule resource pool actions

Performance Enhancement Suggestions:

3 – Change the dispatch time slice

- The default dispatch time slice without SMT enabled is 5ms
- When enabling SMT, the dispatch time slice default becomes 10ms which is more conducive to batch transactions than online
- Set the dispatch time slice to 1ms
  - Online transactions do much better with this setting
  - CP SET SRM DSPSlice 1

Customer Feedback

What the customer reported:

- The dispatch time slice was set to 1ms and is working well

- Resource pools are being created/updated

- The SHARE for SERVER10 was set to REL 1200
- Another slowdown was seen due to a hot-running process
- The SHARE was then set to REL 2400

- No other issues have been reported