August 2002

Console Performance Hints and Tips for a Parallel Sysplex Environment

Author: David Yackel
IBM Corporation
z/OS Software Service
Poughkeepsie, NY
Phone (external): (845)-435-4069
Phone (internal): 8-295-4069
email:DSYACKEL@US.IBM.COM

Table of Contents

Introduction

This paper contains information to assist customers with analyzing performance problems related to the CONSOLE address space in a Parallel Sysplex® environment. The basic concepts apply to non-parallel environments, as well. This paper assumes that the reader has a basic understanding of the z/OS™ Consoles and performance monitors used to analyze CPU usage.

The CONSOLE address space provides multiple functions related to console and message processing in a SYSPLEX environment. Sometimes these functions can use a large amount of CPU resources, thereby causing problems for other units of work. Some of these problems are known defects, and there are APARs/PTFs to correct these. Others are due to how the system or SYSPLEX is configured. Still others are due to how heavily CONSOLE functions are being used. The purpose of this paper is to help identify those scenarios that can adversly affect CPU utilization, thereby helping to avoid potential outages related to CONSOLE activities.

This paper is divided into multiple sections. The first section describes what data to obtain in order to determine if a CONSOLE performance problem you are experiencing is a known problem. The following sections describe performance problems that you may experience, with a non-defect resolution.

This paper is intended to be used by system programmers who are familiar with their SYSTEM/SYSPLEX set up, and how CONSOLEs and CONSOLE functions are generally used in their SYSTEM/SYSPLEX. The system programmer should know/understand what is "normal" CPU usage by the CONSOLE address space on their systems. In addition, examples that are given in this paper are used in a effort to demonstrate extremes of CONSOLE CPU utilization; they are not meant to describe what should be considered "normal" for each customers' SYSTEM/SYSPLEX.

Determining if CONSOLE Performance Problem is a Known Problem

As with all APARs, we try to identify performance problems related to CONSOLE functions with specific keywords.

The keyword *PERFORMANCE* is used to describe a negative impact on the SYSTEM due to the CONSOLE address space. These problems are accompanied by another primary symptom associated with the performance problem, such as *ENQ, COMMAND, or LOOP* . These APARs will also contain the secondary symptom that describes the primary one, such as the major resource name (ie. SYSZMCS), the specific command (ie. ROUTE), or the csect(s) that is/are looping (ie. IEAVMDOM).

The keywords *HIGH CPU UTILIZATION* are used to describe those problems in which the primary symptom is directly related to how much CPU the CONSOLE address space is using. Generally, these problems are detected by system monitors running on the z/OS system. The secondary symptom is generally the task name (ie. IEAVG617) that the monitor identifies as using the CPU, and the csect name (ie. CSRCPQRY) that the CPU is primarily running in.

Using the above keywords in the combinations described will give very specific hits, and help to find a known problem quickly. If you are in doubt as to whether an APAR describes the problem you are experiencing, obtain a dump of the CONSOLE address space and its dataspaces (see the DUMP command in the Commands and Responses section) and queue to the IBM Support Center, where Level2 will be able to confirm/recommend the specific APAR/PTF for the problem.

In addition to the above keywords, such problems are also marked *HIPER*. It is also IBM's recommendation that HIPER APARs be examined regularly, and CONSOLE maintenance be kept very current, so as to avoid serious known problems.

CPU Utilization SPIKES

There are times when the CONSOLE address space may use a lot of CPU but does so in "spikes" – the CONSOLE address space may normally be using 1–2 % CPU and then jump to 20–30 % for a period of time and then go back to normal. These "spikes" may come and go, and may be associated with additional symptoms – an increase/decrease in message traffic, specific jobs starting/ending, or resource contention. If such is the case, it may be "normal," and nothing to worry about. However, if these "spikes" are NOT associated with any of the above additional symptoms, then the "spike" might be related to the CONSOLE address space itself and its relationship to other address spaces which produce work (ie. messages, deleting messages, commands, etc.) for CONSOLE.

Spikes and Dispatching Priority

The CONSOLE address space should run at the highest dispatching priority (Compatibility mode) in the system (x"FF"). However, entries specified in your IEAICSxx and IEAIPSxx parmlib members may result in the CONSOLE address space having a dispatching priority less than x"FF." This allows for the possibility that other address spaces can have a higher DP, and can therefore run more/longer. If such address spaces perform functions that give work to the CONSOLE address space (messages, commands, automation, deleting messages), such work can "back up" since the CONSOLE address space may not get the CPU often enough to keep up concurrently with the given work. The end results are these CPU "spikes" where the CONSOLE address space now uses as much processor as possible for as long as possible in order to do all its work before being interrupted by a higher dispatchable unit of work. Additional symptoms to these "spikes" may be ENQ contention on SYSZMCS, and what appear to be "loops" in CONSOLE csects.

In order to resolve this condition, any reference to CONSOLE (TRXNAME) should be removed from the IEAICSxx parmlib member (This is also true of the GRS and XCFAS address spaces). If you are running OS/390® R10 or higher, then APAR OW49843 will prevent this problem by ensuring that these system address spaces' dispatching priority will be x"FF."

High Sustained or Gradually Increasing CPU Utilization

CONSOLE processing attempts to give a SYSPLEX view from any system in the SYSPLEX environment. This SYSPLEX view includes messages and consoles. In order to obtain this SYSPLEX view, CONSOLE data is *replicated* throughout the SYSPLEX. This means that a message issued on SYS1 and that needs to go to SYS2, is sent and re-issued via WTO on SYS2. A console defined on SYS1 has all of its data sent to all other systems in the SYSPLEX so that they each know of that console and its attributes. An action message retained on SYS1 via AMRF is sent to all other systems in the SYSPLEX for them to issue via WTO and save in their copies of AMRF. To summarize, when CONSOLE is using a lot of CPU, it needs to be determined what type of work CONSOLE is doing and where that work is originating - from this system, or from other systems in the SYSPLEX.

Message Rates and SYSLOG

The System Log (SYSLOG) is an excellent indication of the message work being done on this system. SYSLOG can be used to gauge the "normal" amount of message work, and can therefore be used to see if message work ever becomes abnormal. In addition, the normal message rate can then be compared between systems in a SYSPLEX to determine if there may be any system whose message work may be disproportionally higher.

Message rates may be disproportional between systems for a couple of reasons. One reason could be due to bursts of messages related to some periodic activity, ie. jobs starting/terminating, commands being issued, etc. Another reason could be a result of a difference in MIP rates between systems. If SYS1 runs at 900 MIPS, and SYS2 at 100 MIPS, then it is very possible that SYS1 could overrun SYS2's capability to keep up with the messages SYS1 is sending to it. This would also be validated through the message rate comparison of each system's SYSLOGs.

Message Rates and Console Tasks

Monitor applications can also give a clue about where message traffic is being generated. The IEAVMQWR task does the work of queuing messages to consoles defined on *this* system. The IEAVG617 task does the work of receiving messages sent from other systems, and then placing them on this systems' work queue. If the monitor indicates that the majority of CPU time is being spent under the IEAVMQWR task, then it is an indication that the message traffic is originating from this system. If the monitor shows the CPU time spent under the IEAVG617 task is close to or greater than the IEAVMQWR task, then the message traffic is originating from another system in the SYSPLEX environment.

Actions to address message rates

Once it has been determined where the message traffic has originated, you can use the "D C" and "D EMCS" commands to check message routing. If the messages are originating from elsewhere in the SYSPLEX, verify the MSCOPE values for the consoles. You may need to remove systems from your console's MSCOPE list in order to eliminate messages from that system. This is particularly useful when a system in the SYSPLEX is flooding other systems in the SYSPLEX, either due to the amount of messages it is issuing, or due to the disparity in MIP rates between systems. If the message traffic is originating on this system, then SYSLOG and/or a "D C,B" command can be used to determine whether a specific job(s) is/are flooding the system and allow you to take action against the job(s), such as canceling the job.

Other Console functions and Tasks

There are other functions that the CONSOLE address space performs, and some of these can also lead to high CPU usage. Following are some specific task names in the CONSOLE address space, their functions, and possible actions to take.

IEAVMQWR

This task performs multiple functions, and does the bulk of the work in the CONSOLE address space. It is common for this task to show using the most total CPU time. However, when the CONSOLE address space is using an abnormal amount of CPU, and this task is the highest, then check the following areas:

WTO BUFFER SHORTAGE – The IEAVMQWR task detects and will attempt to alleviate a WTO buffer shortage. It does so by processing DOMs and MCS console work in order to take WTOs off the chain. If there is a WTO buffer shortage that goes unresolved for a long period of time, or if the system keeps going into and out of a shortage condition, IEAVMQWR will accumulate a large amount of CPU. Actions to take to relieve a WTO buffer shortage condition are described in the "WTO Buffer Shortages" section.

AMRF – Action Message Retention Facility – The IEAVMQWR task will also RETAIN action messages in AMRF (if active). As this task processes DOMs, it runs the AMRF chain as well so as to remove the message from AMRF. If AMRF grows to be very large (> 500), then the amount of time taken to run the chain for each single DOM also increases (whether a match is found or not). You can determine the status of AMRF and how many of each type of action message is currently being retained with the following command:

D R

```
IEE112I 15.16.17 PENDING REQUESTS 233
RM=2    IM=30    CEM=25    EM=3421  RU=0    IR=0    AMRF
```

Generally, the IM (immediate action messages) and the CEM (critical eventual action messages) are DOM'd by the function or application that is issuing them. It is usually the EM (eventual action messages) that fill up AMRF (an example of an EM message is IEF234E, which is not highlighted and not DOM'd). To alleviate the amount of CPU being used that is directly related to the number of EM messages in AMRF, use the following command to DELETE the EM messages:

```
K C,E,n-nnnnnnnnn
```

This command will remove from AMRF the eventual action messages whose IDs fall into the specified range. To delete ALL EM messages, the range would be 1-9999999999. If there are many EM messages, this command may take a few minutes, and more CPU may be used by the CONSOLE address space during this time.

If, after deleting the action messages from AMRF, you wish to turn off AMRF, it can be done via the following command:

```
K M,AMRF=N
```

This command will not delete any messages already in AMRF; it will only prevent more messages from being added to AMRF.

It is also recommended to NOT RETAIN any EM messages. This can be done by coding

```
.DEFAULT RETAIN(I,CE)
.DEFAULT NO_ENTRY(I,CE)
```

in your MPFLSTxx parmlib member.

IEAVG617

This task does the work of taking messages sent from other systems and placing them on this system’s message work queue. If this task shows as using much of the CPU time, then it is due to one of the following reasons:

Abundant Foreign Message Traffic

If the number of messages being sent from other systems is very large, then CONSOLE can spend a lot of time processing these messages. It is even possible that this message traffic could overwhelm the CONSOLE address space to the point that it can not keep up. There are no external commands that will show the amount of messages that are on this queue, however, a "D C,B" command will give a summary of messages on the WQE chain, which should be reflective of messages coming from other systems.

```
d c,b
IEE889I 14.10.27 CONSOLE DISPLAY 379
MSG: CURR=1070 LIM=1500 RPLY:CURR=0    LIM=20   SYS=YAK1      PFK=01
 CONSOLE/ALT       ID  -------------- SPECIFICATIONS ---------------
 YAK1MSTR          11  COND=A      AUTH=MASTER       NBUF=1065 UD=N
  03E0                  AREA=Z          MFORM=T,S
  YAK1                  DEL=RD   RTME=1      RNUM=10   SEG=14   CON=N
                        USE=FC   LEVEL=ALL             PFKTAB=01
                        ROUTCDE=1-2,8-10
                        CMDSYS=YAK1
                        MSCOPE=*ALL
                        MONITOR=JOBNAMES
                        ALTGRP=ALTGRPK1
 WTO BUFFERS IN CONSOLE BACKUP STORAGE =        0
 ADDRESS SPACE WTO BUFFER USAGE
 NO ADDRESS SPACES ARE USING MORE THAN   500 WTO BUFFERS
 MESSAGES BEING HELD FOR OTHER SYSTEMS - WTO BUFFER USAGE
 NO WTO BUFFERS ARE BEING HELD FOR OTHER SYSTEMS
 MESSAGES COMING FROM OTHER SYSTEMS - WTO BUFFER USAGE
 SYSTEM = BAT2      NBUF =     1023
 SYSTEM = FOX3      NBUF =       10
```

In the above example, the display shows that most of the messages on system YAK1's message queue are from system BAT2.

If the display did not give any clues as to where the messages are coming from, then an examination of the SYSLOGs of each system would be necessary (see *Message Rates and SYSLOG* and *Message Rates and Console Tasks*).

To reduce or eliminate messages coming from other systems, you would need to determine ("D C,A,SYS=sysname" for MCS consoles, "D EMCS,I,SYS=sysname" for EMCS consoles) what Consoles are MSCOPEd to which systems, and either change the MSCOPE for those consoles, or change their queuing attributes (ie. Route codes, Automation, HC, Level, etc.).

```
d c,a,sys=yak1
IEE889I 11.59.06 CONSOLE DISPLAY 307
MSG: CURR=1070 LIM=1500 RPLY:CURR=0    LIM=20   SYS=YAK1      PFK=01
 CONSOLE/ALT       ID  --------------- SPECIFICATIONS ---------------
 SYSLOG                 COND=H      AUTH=CMDS        NBUF=0    UD=N
                        ROUTCDE=ALL
 YAK1MSTR          01  COND=M      AUTH=MASTER       NBUF=1065 UD=Y
  03E0                  AREA=Z          MFORM=T,S
  YAK1                  DEL=RD   RTME=1      RNUM=10   SEG=14   CON=N
                        USE=FC   LEVEL=ALL             PFKTAB=05
                        ROUTCDE=ALL
                        LOGON=OPTIONAL
                        CMDSYS=YAK1
                        MSCOPE=*ALL
                        MONITOR=JOBNAMES
                        ALTGRP=ALTGRPK1
```

In the above example, console YAK1MSTR has MSCOPE=*ALL, which means it will receive messages from all other systems. The messages it will receive are defined by its other queuing attributes, therefore this console will receive any message with ROUTE CODEs 1-128 and any undelivered message (UD=Y - action messages not sent to any console because there was no intersection with any console's queuing attributes). This could result in a large number of messages being sent to this system, depending on the number of systems in the SYSPLEX and the amount of message traffic on each system.

To change the MSCOPE value of this console, issue the following command:

```
v cn(yak1mstr),mscope=*
IEE712I VARY CN  PROCESSING COMPLETE
d c,cn=(yak1mstr)
IEE889I 12.24.59 CONSOLE DISPLAY 325
MSG: CURR=1    LIM=1500 RPLY:CURR=0    LIM=20   SYS=YAK1      PFK=01
 CONSOLE/ALT       ID  --------------- SPECIFICATIONS ---------------
 YAK1MSTR          01  COND=M      AUTH=MASTER       NBUF=1    UD=Y
  03E0                  AREA=Z          MFORM=T,S
  YAK1                  DEL=RD   RTME=1      RNUM=10   SEG=14  CON=N
                        USE=FC   LEVEL=ALL             PFKTAB=05
                        ROUTCDE=ALL
                        LOGON=OPTIONAL
                        CMDSYS=YAK1
                        MSCOPE=YAK1
                        MONITOR=JOBNAMES
                        ALTGRP=ALTGRPK1
```

Instead of changing the MSCOPE value, message traffic can be reduced by changing the queuing attribute, specifically the route codes. To change only the route codes for this console, issue the following command:

```
v cn(yak1mstr),rout=(1,2,8-10)
IEE712I VARY CN  PROCESSING COMPLETE
d c,cn=yak1mstr
IEE889I 12.33.11 CONSOLE DISPLAY 348
MSG: CURR=1    LIM=1500 RPLY:CURR=0    LIM=20   SYS=YAK1      PFK=01
 CONSOLE/ALT       ID  --------------- SPECIFICATIONS ---------------
 YAK1MSTR          01  COND=M      AUTH=MASTER       NBUF=1    UD=Y
  03E0                  AREA=Z          MFORM=T,S
  YAK1                  DEL=RD   RTME=1      RNUM=10   SEG=14  CON=N
                        USE=FC   LEVEL=ALL             PFKTAB=05
                        ROUTCDE=1-2,8-10
                        LOGON=OPTIONAL
                        CMDSYS=YAK1
                        MSCOPE=*ALL
                        MONITOR=JOBNAMES
                        ALTGRP=ALTGRPK1
```

You will need to check your EMCS consoles as well:

```
d emcs,i,sys=yak1
IEE130I 12.40.38 DISPLAY EMCS 351
DISPLAY EMCS,I,SYS=YAK1
NUMBER OF CONSOLES MATCHING CRITERIA: 3
CN=YAK1     STATUS=A    CNID=01000001 MIGID=100 KEY=SYSCONS
  SYS=YAK1     ASID=000A JOBNAME=-------- JOBID=--------
  UD=N HC=N AUTO=N DOM=ALL    TERMNAME=YAK1
  MONITOR=--------
  CMDSYS=YAK1
  ALTGRP=-------- LEVEL=ALL,NB       AUTH=MASTER
  MSCOPE=*ALL
  ROUTCDE=NONE
CN=*ROUT001 STATUS=A    CNID=02000001 MIGID=--- KEY=MVSROUTE
  SYS=YAK1     ASID=000A JOBNAME=-------- JOBID=--------
  UD=N HC=N AUTO=N DOM=NONE   TERMNAME=ROUTEALL
  MONITOR=--------
  CMDSYS=YAK1
  ALTGRP=-------- LEVEL=R,NB         AUTH=CONS
  MSCOPE=*ALL
  ROUTCDE=NONE
CN=EMCS0001 STATUS=A    CNID=03000001 MIGID=--- KEY=AUTOEMCS
  SYS=YAK1     ASID=0054 JOBNAME=MYAUTO   JOBID=STC10501
  UD=N HC=N AUTO=Y DOM=NONE   TERMNAME=EMCS0001
  MONITOR=--------
  CMDSYS=YAK1
  ALTGRP=-------- LEVEL=R,NB         AUTH=CONS
  MSCOPE=*ALL
  ROUTCDE=NONE
```

In the above examples, consoles YAK1 and *ROUT001 have MSCOPE=*ALL, but they do not have any other queuing attributes. Therefore, there are no unsolicited messages being sent to these consoles. Only messages being sent by console name/id are going to them (ie. as a result of a command response). Console EMCS0001 also has MSCOPE=*ALL, and a queuing attribute of AUTO=Y. Any messages with AUTO=YES specified from any system will be sent here. Most likely, this EMCS console is not automating messages from another system, so changing its MSCOPE would be the action to take to reduce foreign message traffic. This can be done with the commands described above.

In the above examples, changing MSCOPE to be just this system (MSCOPE=*) will have the greatest effect of reducing the amount of messages coming to this console. However, changing MSCOPE (not *ALL and not*) and ROUTCDE in combination will reduce message traffic – you will need to continue monitoring CPU usage to determine if the reduction was effective. NOTE - it is the sum total of all console's queuing attributes that determines whether messages are sent from other systems. Changing all consoles but still having one that is MSCOPE=*ALL and other queuing attributes will not reduce the amount of CPU that IEAVG617 is using.

Automation on the Subsystem Interface (SSI)

The IEAVG617 task takes messages sent from other systems and places them on this systems' work queue by issuing a WTO. When WTO processes, a copy of the message is presented to each subsystem on the subsystem chain. Any processing done by any of these subsystems adds to the time spent in WTO, and hence to the total CPU time of the task. It has been seen that subsystem time can account for > 90% of the total time spent in WTO, which equates to about the same amount of time for the task. A monitor that can break down CPU time under a task to a CSECT or CSECTs is the best way to determine where the time is being spent, and relate that time to actions being taken by a specific subsystem (ie. automating on messages coming from other systems in the SYSPLEX).

IEAVM613

This task is responsible for taking branch entry WTOs that have been issued, and placing them on the WQE chain. If this task is using a lot of CPU, then there are many branch entry WTOs being issued. Reducing or eliminating the BR Entry WTOs will reduce this task's CPU time. SYSLOG can be examined to see if there are a large number of the same messages being issued (with the assumption that the message is a BR Entry WTO). An example of a BR Entry WTO is BPXF024I – if many of these are seen in SYSLOG, determine the reason for its issuance and correct.

IEAVM605

This task does the work for the System Console (SYSCONS). Normally, the SYSCONS is in an active state as shown:

```
d emcs,i,cn=sysconk1
IEE130I 14.36.28 DISPLAY EMCS 413
DISPLAY EMCS,I,CN=SYSCONK1
NUMBER OF CONSOLES MATCHING CRITERIA: 1
CN=SYSCONK1 STATUS=A    CNID=01000002 MIGID=101 KEY=SYSCONS
  SYS=YAK1    ASID=000A JOBNAME=-------- JOBID=--------
  UD=N HC=N AUTO=N DOM=ALL    TERMNAME=YAK1
  MONITOR=--------
  CMDSYS=YAK1
  ALTGRP=-------- LEVEL=ALL,NB       AUTH=MASTER
  MSCOPE=*ALL
  ROUTCDE=NONE
```

However, when placed in Problem Determination mode (issue V CN(*), ACTIVATE from the SYSCONS itself) the state will be STATUS=A,PD as shown:

```
d emcs,i,cn=sysconk1
IEE130I 14.36.28 DISPLAY EMCS 413
DISPLAY EMCS,I,CN=SYSCONK1
NUMBER OF CONSOLES MATCHING CRITERIA: 1
CN=SYSCONK1 STATUS=A,PD CNID=01000002 MIGID=101 KEY=SYSCONS
  SYS=YAK1     ASID=000A JOBNAME=-------- JOBID=--------
  UD=N HC=N AUTO=N DOM=ALL    TERMNAME=YAK1
  MONITOR=--------
  CMDSYS=YAK1
  ALTGRP=-------- LEVEL=ALL,NB       AUTH=MASTER
  MSCOPE=*ALL
  ROUTCDE=NONE
```

Once in this mode, the SYSCONS can issue commands and receive responses. If this task shows as using high CPU, then check the queuing attributes (as shown above) and make any necessary adjustments (ie. change MSCOPE, ROUTECODE, or UD to reduce message traffic).

Another reason for high CPU time for this task can be directly related to how long the SYSCONS remains in PD mode. If it is left in PD mode for days, weeks, or months, and receives numerous *IEA124I and IEA125I* messages, then you could have the problem described by OW39405. Review the description of the APAR, and contact the Support Center if you have any questions.

It is recommended that the SYSCONS be used as an emergency console, activating it only to issue commands in an attempt to resolve other system problems when MCS/EMCS consoles do not appear to be functioning. Once the problem is resolved, the SYSCONS should then be deactivated.

IEAVM611

These tasks (3) do the work for the EMCS consoles defined on this system. If any of these tasks are using high CPU, then one or more EMCS consoles being handled by the task have queuing attributes that greatly increase the amount of messages the task needs to process. Check the routing attributes for your EMCS consoles by issuing a D EMCS,F,SYS=sysname, as shown:

```
d emcs,f,sys=yak1
IEE130I 15.57.00 DISPLAY EMCS 434
DISPLAY EMCS,F,SYS=YAK1
NUMBER OF CONSOLES MATCHING CRITERIA: 2
CN=SYSCONK1 STATUS=A,PD CNID=01000002 MIGID=101 KEY=SYSCONS
  SYS=YAK1     ASID=000A JOBNAME=-------- JOBID=--------
  UD=N HC=N AUTO=N DOM=ALL    TERMNAME=YAK1
  MONITOR=--------
  CMDSYS=YAK1
  ALTGRP=-------- LEVEL=ALL,NB       AUTH=MASTER
  MSCOPE=*ALL
  ROUTCDE=NONE
  ALERTPCT=100
  QUEUED=0          QLIMIT=10000
CN=*ROUT002 STATUS=A    CNID=02000002 MIGID=--- KEY=MVSROUTE
  SYS=YAK1     ASID=000A JOBNAME=-------- JOBID=--------
  UD=N HC=N AUTO=N DOM=NONE   TERMNAME=ROUTEALL
  MONITOR=--------
  CMDSYS=YAK1
  ALTGRP=-------- LEVEL=R,NB         AUTH=CONS
  MSCOPE=*ALL
  ROUTCDE=NONE
  ALERTPCT=100
  QUEUED=0          QLIMIT=2147483647
CN=HCTASK   STATUS=A    CNID=03000002 MIGID=--- KEY=HARDCOPY
  SYS=YAK1     ASID=000A JOBNAME=-------- JOBID=--------
  UD=N HC=Y AUTO=N DOM=NONE   TERMNAME=HCTASK
  MONITOR=--------
  CMDSYS=YAK1
  ALTGRP=-------- LEVEL=R,NB         AUTH=CONS
  MSCOPE=*ALL
  ROUTCDE=NONE
  ALERTPCT=100
  QUEUED=132        QLIMIT=2147483647
```

Look for MSCOPE, ROUTCDE, HC=Y (Hardcopy), AUTO=Y (present all messages queued for automation), and DOM=ALL (present all DOMs in the SYSPLEX). Limiting these attributes can reduce the amount of work being queued to the console, and hence the amount of work done by the IEAVM611 task(s). This can be done via the VARY CN command, as the following examples show:

```
V CN(HCTASK),MSCOPE=*
V CN(consname),AUTO=N
V CN(consname),UD=N
V CN(consname),ROUT=NONE
```

To change the HC and DOM attribute, the owning application must deactivate the EMCS console, and then activate it with the new values for those attributes.

In the above example, console HCTASK has attributes of HC=Y and MSCOPE=*ALL - this results in this EMCS console getting all the messages appearing in SYSLOG for all systems in the SYSPLEX. The recommendation here would be to change this console's MSCOPE value (MSCOPE=*).

If all of the tasks are using high CPU, and it does not appear to be related to queuing attributes, then it could be due to the total number of EMCS consoles defined. The command D EMCS,S,SYS=sysname,ST=L will show the total count of EMCS consoles for the SYSPLEX:

```
d emcs,s,sys=yak1,st=l

IEE129I 17.51.54 DISPLAY EMCS 890
DISPLAY EMCS,S,SYS=YAK1,ST=L
NUMBER OF CONSOLES MATCHING CRITERIA: 2486
SYSCONK1 AUTOTSK1 AUTOTSK2 *ROUT001 AUTOTSK3 AUTOTSK4
HCTASK01 HCTASK02 SUBSYSA1 SUBSYSA2 SUBSYSA3 SUBSYSA4
TSOUSER1 TSOUSER2 XYZ00001 XYZ00002 XYZ00003 AAAA0099
    .        .        .        .        .        .
    .        .        .        .        .        .
    .        .        .        .        .        .
```

If this number is large (ie. >500 per system in the SYSPLEX) then that could contribute to the amount of CPU the IEAVM611 tasks use. Each EMCS console defined, whether active or not, remains for the life of the SYSPLEX, and the IEAVM611 tasks must look at each one to determine if there are things (messages and/or DOMs) to queue. Therefore, if there is an application that defines a new EMCS console for each command it issues, the number of EMCS consoles can grow dramatically and will definitely affect the amount of CPU IEAVM611 consumes.

WTO Buffer Shortages

WTO buffer shortages occur once the number of outstanding messages goes above MLIM (message limit specified via CONSOLxx or "K M" command). At this point, a 100% buffer shortage condition exists, and message IEA404A is issued. Prior to this event occurring, message IEA405E is issued to indicate that the 80% buffer limit has been reached. At this same time, a "D C,B" (Display Consoles,Backlog) command is issued internally to show any "backups," including any consoles that have messages queued, any jobs that have messages outstanding, any systems that have sent messages to this system that are still outstanding, and how many messages this system has that need to be sent to other systems. Once we have hit the 100% limit, non-authorized problem program WTO issuers are placed into a WAIT for a buffer. Other WTO issuers will get a buffer, but it will be placed in a holding area of storage above the 16M line until a buffer below the lines frees up. Certain WTO issuers are considered "privileged," and will be allowed to get a buffer below the line all the time (ie. Console main task itself, the "D C,B" command processor, and the SYSLOG task).

Note now what the externals will be for a buffer shortage that does NOT get resolved. Almost all WTO issuers will either be placed into a WAIT, or their message buffer will be in the "holding" area above the line. Externally, this will appear as if the system is no longer processing messages. Until buffers free up below the line, no new messages can be displayed, and commands may appear to be "swallowed" and never processed. Consoles will appear "hung." Also, since no messages are being displayed, SYSLOG will also appear "hung," as no messages are hardcopied until after they are processed in buffers below the line. While CONSOLE is in its 100% limit, the main task will attempt to resolve the shortage by examining all of its buffers to see if any can be freed. This means that the IEAVMQWR task may use more processor, as it constantly checks to see if any buffers can be freed. DOM processing (Delete Operator Message) also occurs under this task, so each DOM in the system will cause the task to run the entire chain of messages to see if a match can be found and a buffer freed. The longer the chain of messages, the more work the task will do. Also, if there are a lot of DOMs being issued, the work will grow by a factor of MLIM, as the entire chain is searched for a match for each DOM.

While the external symptoms may appear to indicate that the system is dead or non-responsive, a buffer shortage itself should not cause a system to crash. If the shortage is not resolved, messages will be placed in the holding area above the line in order to prevent any storage problems for the CONSOLE address space. Once 40,000 messages are in this holding area, additional messages are thrown away, thereby preventing CONSOLE from abending due to an out of storage condition. The end result is that the CONSOLE address space will continue to run, and look to resolve the buffer shortage, but the buffer shortage itself will not cause a system to crash. The external manifestation of a buffer shortage gives the appearance of a dead system, however, proper actions can prevent an unnecessary IPL.

There are actions that can be taken to resolve a buffer shortage. The following messages and actions may be helpful in setting up automation to handle a WTO buffer shortage:

IEA653I JOBNAME= jjjjjj ASID= aaaa HAS REACHED 50% OF THE WTO BUFFER LIMIT
This message will be issued for any job that is currently using 1/2 of the total WTO buffers (1/2 of MLIM). Buffers can be freed by CANCEL'ing the job, but care should be taken as it may not be appropriate to CANCEL any or every job.

IEA099A JOBNAME= jjjjjj ASID= aaaa HAS REACHED THE WTO BUFFER LIMIT
This message will be issued for any job that is currently using WTO buffers equal to the current MLIM value. Note that the total number of WTO buffers being used will most likely be greater than MLIM, as other jobs are also using buffers "normally". These buffers can be freed by CANCEL'ing the job, but care should be taken as it may not be appropriate to CANCEL any or every job.

IEA405E WTO BUFFER SHORTAGE - 80% FULL
At the time this message is issued, CONSOLE will also issue a "D C,B" command and produce message IEA889I. (See the "D C,B" example in Commands and Responses). Automation can be used to look for the IEA889I and perform any or all of the following actions:

K Q,L=consname - issue a K Q command for every console listed in the output, as only consoles with buffers backed up will be listed.

CANCEL jobname - CANCEL any job listed in the output; care should be taken as you may not want certain jobs ever to be CANCEL'd.

*V CN(consname),MSCOPE=** - if the display shows many messages coming from other systems, remove other systems from each consoles MSCOPE value, thereby preventing messages from other systems from being sent to this one.

K M,MLIM=nnnn - use this command to increase the current WTO buffer limit up to 9999. This may be provide only temporary relief from the buffer shortage, depending on what the cause is.

IEA404A SEVERE WTO BUFFER SHORTAGE - 100% FULL

Once this message is issued, the system is going above the WTO buffer limit. Commands that can be issued are:

D C,B - use this command to check for a backlog of messages queued to MCS consoles, and automate on the IEE889I as described above.

K S,DEL=R,L=consname - use this command to place all consoles into ROLL mode that are listed in the IEE889I output.

K Q,L=nn - If there is a backlog of buffers above the line, this command will free below the line buffers that are backed up to consoles. Once freed, the above the line buffers will be moved below the line and queued to consoles, possibly causing a shortage again for which repetitive automation actions are required. Therefore, it may be necessary to issue this command multiple times.

D EMCS,ST=B - use this command to check for a backlog of messages queued to EMCS consoles. Message IEE130I is issued. It may be that certain applications that make use of EMCS consoles could be backlogged for a number of reasons: low dispatching priority or it is in a WAIT. It may be necessary to take specific actions related to the application, and/or CANCEL the application so that the EMCS console is deactivated and the buffers freed.

IEE766E BUFFER SHORTAGE FOR SYSTEM LOG - 60% FULL

This message is issued when the SYSTEM LOG buffers, LOGLIM, have reached 60% full. An attempt should be made to identify and resolve the shortage. The SYSTEM LOG is a SPOOL dataset, so the shortage could be related to JES SPOOL problems.

IEE767A SEVERE BUFFER SHORTAGE FOR SYSTEM LOG - 100% FULL

Once this message is issued, the SYSTEM LOG buffers are now full (equal to LOGLIM). From this point on, messages that need to be hardcopied are not placed in a SYSLOG buffer, but are kept on the WQE chain. If the SYSLOG buffer shortage is not resolved, this will lead to a WTO buffer shortage.

K M,LOGLIM=nnnnn - use this command to increase the number of SYSLOG buffers. This is just a temporary action until the cause of the shortage is resolved, however, it will provide some additional time before the shortage adversly affects message processing (leading to a WTO buffer shortage).

K M,LOGLIM=0 - use this command to delete the current SYSTEM LOG buffers in storage. It will immediately resolve the buffer shortage, but not the root cause. If there were messages held on the WQE chain, they will be immediately placed into a SYSTEM LOG buffer. This could result in another SYSLOG buffer shortage. It is possible that this command would need to be issued multiple times to clear both the SYSLOG buffers and the WQEs being held because of the shortage. Note that all messages in the SYSTEM LOG buffers at the time this command is issued will never be hardcopied to SYSLOG.

Summary

There are many factors that can affect the work, and therefore the amount of CPU, that the CONSOLE address space uses. It is important to understand how each system's consoles are configured, for both ease of operations, and CPU consumption. Factors to consider are:

- *Average message traffic of this system*
- *Average message traffic coming from other systems*
- *Message traffic spikes*
- *WTO buffer shortages*
- *Role of Automation in console operations*
- *Role of AMRF in the SYSPLEX*
- *Role of MPF suppression for each system in the SYSPLEX. It is the hope that the use of system monitors and recommendations of this paper will assist you in establishing a console configuration that fits your operations, and does not affect the overall processing of your system or SYSPLEX.*

Commands and Responses

The following are commands and responses referenced throughout this paper and are here for illustration/examples:

D C,A,SYS=thissysname - check routing attributes for this system

This command shows how one can determine the routing attributes for consoles on this system. Check the MSCOPE value and the ROUTCDE value. MSCOPE indicates from what other systems this console should receive messages, and ROUTCDE indicates the ROUTE CODEs those messages must be to receive from those systems. One would use this command when there is high CPU for either IEAVG617 or IEAVMQWR or both:

```
d c,a,sys=yak1

IEE889I 13.31.04 CONSOLE DISPLAY 484
MSG: CURR=3    LIM=1500 RPLY:CURR=2    LIM=20   SYS=YAK1      PFK=01
 CONSOLE/ALT       ID  --------------- SPECIFICATIONS ---------------
 SYSLOG                 COND=H      AUTH=CMDS        NBUF=0    UD=N
                        ROUTCDE=ALL
 YAK1MSTR          01  COND=M      AUTH=MASTER       NBUF=1    UD=Y
  03E0                  AREA=Z          MFORM=M
  YAK1                  DEL=RD   RTME=1      RNUM=10   SEG=14   CON=N
                        USE=FC   LEVEL=ALL             PFKTAB=01
                        ROUTCDE=1-2,8-10
                        CMDSYS=YAK1
                        MSCOPE=*ALL
                        MONITOR=JOBNAMES
                        ALTGRP=ALTGRPK1
```

D EMCS,F,SYS=sysname - check routing attributes for EMCS Consoles

This command will show the routing attributes for the EMCS consoles on this system. Check the MSCOPE value - this value indicates from what other systems this console should receive messages. Then check the ROUTCDE, AUTO, and HC values for these consoles, as they indicate what messages will be sent to this console from the other systems specified via MSCOPE. One would use this command when there is high CPU for IEAVG617, IEAVMQWR, or IEAVM611:

```
d emcs,f,sys=yak1
IEE130I 15.57.00 DISPLAY EMCS 434
DISPLAY EMCS,F,SYS=YAK1
NUMBER OF CONSOLES MATCHING CRITERIA: 2
CN=SYSCONK1 STATUS=A,PD CNID=01000002 MIGID=101 KEY=SYSCONS
  SYS=YAK1     ASID=000A JOBNAME=-------- JOBID=--------
  UD=N HC=N AUTO=N DOM=ALL    TERMNAME=YAK1
  MONITOR=--------
  CMDSYS=YAK1
  ALTGRP=-------- LEVEL=ALL,NB       AUTH=MASTER
  MSCOPE=*ALL
  ROUTCDE=NONE
  ALERTPCT=100
  QUEUED=0          QLIMIT=10000
CN=*ROUT002 STATUS=A    CNID=02000002 MIGID=--- KEY=MVSROUTE
  SYS=YAK1     ASID=000A JOBNAME=-------- JOBID=--------
  UD=N HC=N AUTO=N DOM=NONE   TERMNAME=ROUTEALL
  MONITOR=--------
  CMDSYS=YAK1
  ALTGRP=-------- LEVEL=R,NB         AUTH=CONS
  MSCOPE=*ALL
  ROUTCDE=NONE
  ALERTPCT=100
  QUEUED=0          QLIMIT=2147483647
CN=HCTASK   STATUS=A    CNID=03000002 MIGID=--- KEY=HARDCOPY
  SYS=YAK1     ASID=000A JOBNAME=-------- JOBID=--------
  UD=N HC=Y AUTO=N DOM=NONE   TERMNAME=HCTASK
  MONITOR=--------
  CMDSYS=YAK1
  ALTGRP=-------- LEVEL=R,NB         AUTH=CONS
  MSCOPE=*ALL
  ROUTCDE=NONE
  ALERTPCT=100
  QUEUED=132        QLIMIT=2147483647
```

D C,B - check for messages originating from elsewhere

This command shows how one can determine from what other system(s) messages are coming from, causing high CPU for either IEAVMQWR or IEAVG617 or both:

```
d c,b

IEE889I 13.19.10 CONSOLE DISPLAY 475
MSG: CURR=450  LIM=1500 RPLY:CURR=2    LIM=20   SYS=YAK1      PFK=01
 CONSOLE/ALT        ID  --------------- SPECIFICATIONS ---------------
 YAK1MSTR           01  COND=M      AUTH=MASTER       NBUF=1  UD=Y
  03E0                  AREA=Z          MFORM=M
  YAK1                  DEL=RD   RTME=1      RNUM=10   SEG=14   CON=N
                        USE=FC   LEVEL=ALL            PFKTAB=01
                        ROUTCDE=1-2,8-10
                        CMDSYS=YAK1
                        MSCOPE=*ALL
                        MONITOR=JOBNAMES
                        ALTGRP=ALTGRPK1
 WTO BUFFERS IN CONSOLE BACKUP STORAGE =        0
 ADDRESS SPACE WTO BUFFER USAGE
 NO ADDRESS SPACES ARE USING MORE THAN   500 WTO BUFFERS
 MESSAGES BEING HELD FOR OTHER SYSTEMS - WTO BUFFER USAGE
 NO WTO BUFFERS ARE BEING HELD FOR OTHER SYSTEMS
 MESSAGES COMING FROM OTHER SYSTEMS - WTO BUFFER USAGE
 SYSTEM = BAT2      NBUF =      439
 SYSTEM = FOX3      NBUF =        1
```

D EMCS,S,ST=L - determine the number of EMCS consoles in the SYSPLEX

This command shows how one can determine the total number of EMCS consoles defined in the SYSPLEX. The total number, whether active or not, can affect the amount of CPU used by the IEAVM611 tasks:

```
d emcs,s,st=l

IEE129I 17.51.54 DISPLAY EMCS 890
DISPLAY EMCS,S,ST=L
NUMBER OF CONSOLES MATCHING CRITERIA: 6
SYSCONK1 SYSCONX3 SYSCONT2 *ROUT001 *ROUT002 *ROUT003
```

D R - check for AMRF active, and count of action messages

This command shows how one can determine whether AMRF is active, and how many of each type of action messages there are:

```
d r
```

```
IEE112I 15.16.17 PENDING REQUESTS 233
RM=2    IM=30    CEM=25    EM=3421  RU=0    IR=0    AMRF
```

RM = Reply Messages - the number of outstanding WTORs
IM = Immediate action Messages - highlighted, descriptor code 1 or 2
CEM = Critical Eventual action Messages - highlighted, desc code 11
EM = Eventual action Messages - non-highlighted, desc code 3

K C,E - delete Eventual Actions messages from AMRF

The following command will delete all eventual action message from AMRF. Using the example from the "D R" command, the command and response is:

k c,e,1-9999999999

IEE146I K COMMAND ENDED-3421 MESSAGE(S) DELETED

DUMP - obtaining a dump of the CONSOLE address space

This command shows how one can dump the CONSOLE address space and its dataspaces:

```
dump comm=(console asid)
```

```
R nn,j=console,dspname=('console'.*),sdata=(rgn,allnuc,psa,sqa,csa,
trt,grsq),end
```

IBM Corporation
Marketing Communications, Server Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
08/02

Information concerning non-IBM products was obtained from the suppliers of their products or their published announcements. Questions on the capabilities of the non-IBM products should be addressed with the suppliers.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

GM13-0166-00
ZSW03334-USEN-00