



| z/OS Software

Getting Started With Message Flood Automation

| Kevin Kelley
z/OS Core Components Development
kkelley@us.ibm.com

© 2008 IBM Corporation

Trademarks and Disclaimers

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml; AIX, AIX SL, BladeCenter, Blue Gene, DB2, e-business logo, eServer, IBM, IBM Logo, Infoprint, IntelliStation, iSeries, pSeries, OpenPower, POWER5, POWER5+, Power Architecture, TotalStorage, Websphere, xSeries, z/OS, zSeries

The following are trademarks or registered trademarks of other companies:

Java and all Java based trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries or both Microsoft, Windows, Windows NT and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

NOTES:

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Users of this document should verify the applicable data for their specific environment.

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

Information is provided "AS IS" without warranty of any kind.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. 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.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices are suggested US list prices and are subject to change without notice. Starting price may not include a hard drive, operating system or other features. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use.

The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The

materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM makes no representation or warranty regarding third-party products or services including those designated as ServerProven, ClusterProven or BladeCenter Interoperability Program products. Support for these third-party (non-IBM) products is provided by non-IBM Manufacturers.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquiries, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.

z/OS Message Flood Automation

- **Delivered as an SPE (OA17514) for z/OS R6, R7 and R8.**
- **Delivered in the base of z/OS R9.**
 - Further integration to come
- **Part of z/OS Consoles component.**
- **Shipped by GDPS since 1Q2003 – installed at some of their largest customers. Has successfully dealt with a number of message flooding situations.**

The Message Flood Automation SPE was made available November 29, 2006.

GDPS has shipped Message Flood Automation since 1Q2003. It has been a formal part of GDPS 3.2 and GDPS 3.3. Message Flood Automation has been installed at several of the very largest GDPS customers and has successfully dealt with the message floods that occur when a HyperSwap operation is performed.

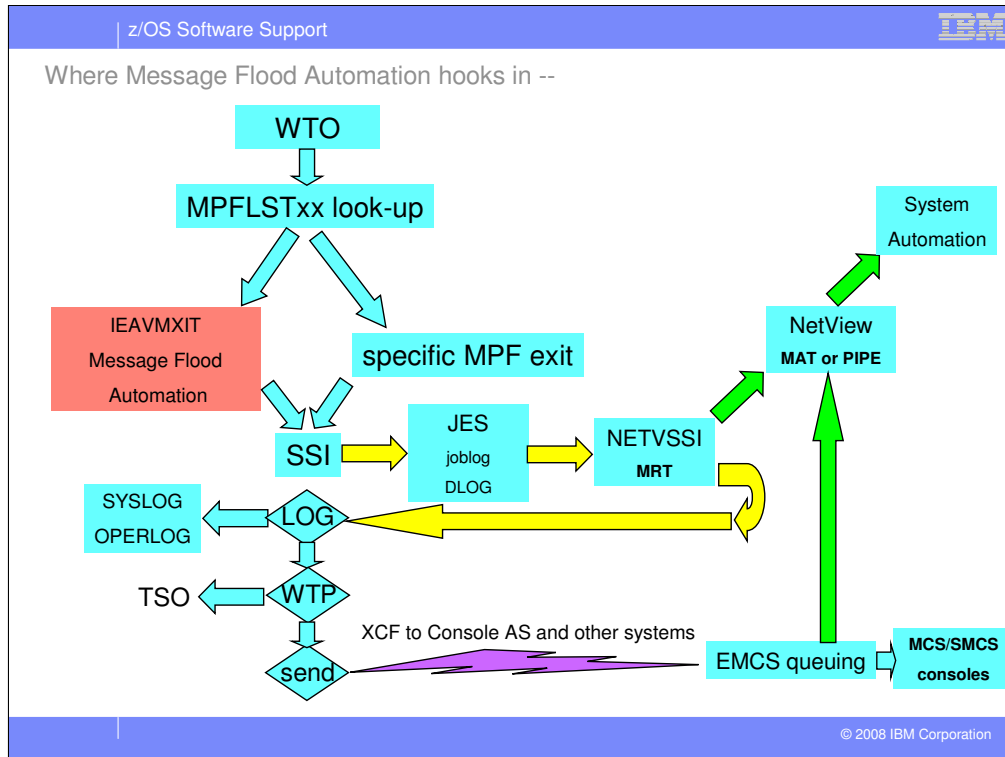
z/OS Message Flood Automation - Implementation

- **Implemented as an MPF IEAVMXIT exit and as a Command Exit.**
 - Several historical reasons for this.
 - Shortcomings of this approach are understood.
 - Customer must take explicit steps to install and enable.

The current implementation is a rework of a prior IEAVMXIT-based implementation done in the United Kingdom in 1990 for one of the large banks. Initially, it was expected that Message Flood Automation would be made available in source form as a web-download. Over time, this decision was changed to a web-download in object form of a new component. Literally, at the last minute, the decision was made to make Message Flood Automation available as an SPE on the Consoles component. This last minute decision provided little time to make all but the most rudimentary changes.

IBM is well aware of the problems that this implementation cause. Based on our experience in the GDPS environment, it was viewed as more important to make the function available than worry immediately about the implementation issues.

Use of Message Flood Automation is optional. You must take explicit actions to install and enable Message Flood Automation.



This diagram shows where Message Flood Automation receives control during message processing. It also shows where Message Flood Automation operates relative to other automation, such as the NetView Message Revision Table (MRT) and NetView (Message) Automation Table (MAT) and Tivoli System Automation.

z/OS Message Flood Automation – What Does It Do?

- **Uses thresholding technology to determine if a message flooding situation is underway, and if so, takes installation-specified actions against the offending message or address space, *before* messages fill buffers or get queued to consoles.**
- **Policy specifications**
 - Built-in for use prior to reading PARMLIB
 - New MSGFLDxx PARMLIB member
 - SET MSGFLD=xx command
 - New DISPLAY MSGFLD and SETMF commands

The whole point of Message Flood Automation is to take action against messages *before* the messages have an opportunity to be written to the log, sent to other systems in the sysplex, fill buffers or be queued to consoles or automation. Rather than clean-up after the mess has occurred, Message Flood Automation attempts to prevent most of the symptoms of a message flood from ever occurring.

Default, built-in policy is provided to handle message floods that occur before the system services exist to read PARMLIB. Individual built-in policy specifications are used if the MSGFLDxx PARMLIB member does not provide an overriding policy specification.

The SET MSGFLD=xx command (which can be abbreviated T MSGFLD=xx) behaves like any other SET command but it is not implemented as part of the SET command processor and cannot be invoked as part of the invocation of other SET commands. The MSGFLDxx PARMLIB member must reside in the PARMLIB concatenation like any other PARMLIB member.

The DISPLAY MSGFLD command (which can be abbreviated D MF) is not implemented as part of the DISPLAY command processor but otherwise behaves like any other DISPLAY command. All of the Message Flood Automation policy can be displayed using the DISPLAY MSGFLD command.

The SETMF command allows all of the Message Flood Automation policy to be altered dynamically. Any changes made are retained until the next SET MSGFLD=xx command is issued or an IPL occurs.

z/OS Message Flood Automation – How Does It Do It?

- **Two step process**

- *Very* low overhead process to determine if a flooding situation is underway;
- More intensive process to determine the specific culprit and to determine when the flood has ended.
- Initial flooding determination made based on messages / unit of time
 - MSGTHRESH / INTVLTIME
- Identity of flooder also determined by messages / unit of time
 - JOBTHRESH / INTVLTIME
 - MSGLIMIT / INTVLTIME

Normal Message Flood Automation processing has very low overhead because no working storage is obtained and no recovery environment is set-up.

“Intensive mode” processing does obtain working storage and does set-up a recovery environment.

The smallest unit of time that may be specified is one second.

z/OS Message Flood Automation – How Does It Do It?

- **Two step process, continued**
 - The time between successive messages is used to determine when a flooding situation has ended
 - SYSIMTIME
 - JOBIMTIME
- **Message Flood Automation provides audit-trail messages in the SYSLOG or OPERLOG**
 - when it detects a message flood
 - when it determines what is causing the flood
 - when it ceases taking action on a flood

The various “inter-message” times are real numbers and can have fractional values as small as 1 millionth of a second.

Message Flood Automation provides an audit trail of its activities in the SYSLOG and/or OPERLOG. It provides messages whenever it transitions from one mode of operation to another. When a flood ceases, Message Flood Automation will summarize how many messages it took action against.

z/OS Message Flood Automation – How Does It Do It?

- **Actions that can be taken:**
 - LOG | NONLOG – writing of the message to SYSLOG/OPERLOG
 - DISPLAY | NODISPLAY – queuing of the message to consoles
 - AUTO | NOAUTO – queuing of the message for automation
 - RETAIN | NORETAIN – (action messages only) retention in AMRF
 - CMD | NOCMD – issuing a command
 - IGNORE | NOIGNORE – whether to take action or not
- **Specifying NOLOG, NODISPLAY and NOAUTO will delete the message.**
- **If the message is logged, the actions taken are visible in the MPF flags.**

Not all of the actions can be taken against all messages. For example, the RETAIN and NORETAIN actions only apply to messages that have action message descriptor codes set.

Deletion of a message occurs immediately after the Subsystem Interface (SSI) and will prevent the message from being sent to other systems in the sysplex, being written to the SYSLOG or OPERLOG and being sent to the Consoles address space for caching and queuing to consoles.

z/OS Message Flood Automation – How Does It Do It?

- **Granularity:**

- Messages are divided into three classes, with independent policy and independent processing for each class:
 - SPECIFIC messages – messages by message ID
 - ACTION messages – messages with action descriptor codes
 - REGULAR messages – all other messages
- For each class, the class policy can be overridden for specific message IDs (SPECIFIC messages) or JOBS (ACTION and REGULAR messages)

ACTION messages are messages that have descriptor codes 1, 2, 3 or 11 set.

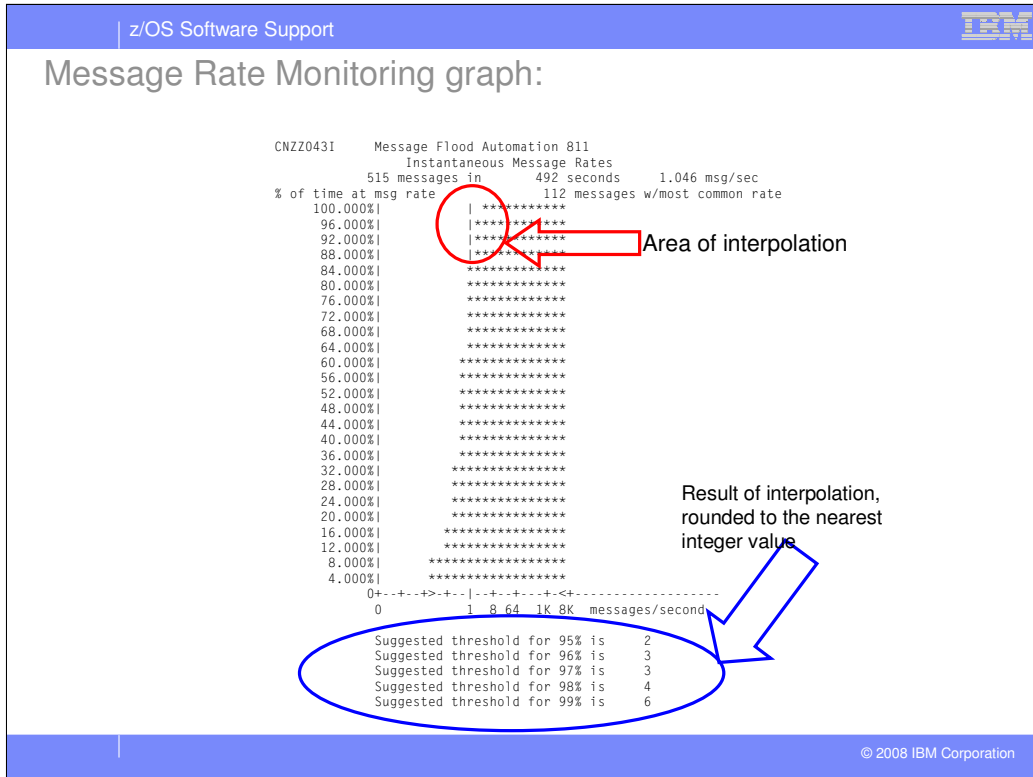
z/OS Message Flood Automation – How Does It Do It?

- **The key to setting up Message Flood Automation is determining the REGULAR MSGTHRESH value.**
 - Should be set high enough that Message Flood Automation is not constantly “tripping” in and out –
 - *But* low enough to catch real message floods in their infancy
- **Message Rate Monitoring function provided to measure the message rate as Message Flood Automation sees it**
 - Message Rate Monitoring provides suggested values for REGULAR MSGTHRESH
 - Message Rate Monitoring should be run periodically to readjust REGULAR MSGTHRESH value

Most of the time, Message Flood Automation should be invisible. It should only be visible – and take action – when it is really needed.

The Message Rate Monitoring function can be turned on and turned off by operator command to obtain information about typical message rates within the system. It uses the same filtering as the rest of Message Flood Automation and therefore “sees” messages at the same rate that Message Flood Automation “sees” them. The message rate information captured by the Message Rate Monitoring function can be displayed in the form of a graph that is produced in response to a DISPLAY MSGFLD,MSGRATE command. The graph is a multi-line WTO and by default has 25 lines. The number of lines produced can be varied from 8 to 200 through a parameter on the command. To make the graph compact, a logarithmic scale is used on the X-axis. Since this can make reading values from the graph somewhat more difficult, built-in linear interpolation is provided for the values most likely to be of interest in setting the REGULAR MSGTHRESH value.

IBM recommends that the Message Rate Monitoring function be used to obtain at least a 24 hour message rate sample *from the system that the policy is being prepared for*. To be the most useful, the sample should include the busiest part of the day on the busiest day of the year. You should run the Message Rate Monitoring function and readjust your policy thresholds every few months or whenever the load on the system substantially increases.



This chart contains an actual output produced in response to a D MF,MSGRATE command.

The X-axis scale is logarithmic – each position is a power-of-2 greater than or less than its neighbor.

The Y-axis is the percent of time at a particular message rate (*or less*). For example, picking the 1 message per second rate on the X-axis, it can be seen that a message rate of 1 message per second or less occurs 84% of the time.

The '>' and '<' symbols on the X-axis bracket the instantaneous message rates that were observed in the sample. The '|' symbol shows the most commonly occurring message rate (this is not necessarily the same as the simple average message rate that is shown in the header).

You can see how difficult it would be to manually derive message threshold values from the graph. Although the Y-axis can be extended up to 200 lines, it would still be very difficult to figure out the corresponding X values. The built-in interpolation does this automatically for the user and provides the X values rounded to the nearest integer value (since Message Flood Automation message thresholds are integer values). The suggested thresholds are interpreted as follows: "98% of the time, the instantaneous message rate will not exceed 4 messages per second". Notice that the threshold for 99% is considerably less than the peak instantaneous message rate that was observed which was approximately 4096 messages per second! A high instantaneous message rate does not imply that there was a flood; it merely indicates that two successive messages occurred very close in time to one another (in this case, within 1/4096th of a second of one another). Sporadic high instantaneous message rates will not trigger Message Flood Automation; Message Flood Automation is triggered only when a large number of messages occur within a short period of time.

Note the characteristic "S" shape of the graph: there are few messages occurring at very low message rates and few messages occurring with very high message rates.



| z/OS Software

Message Flood Automation in Action



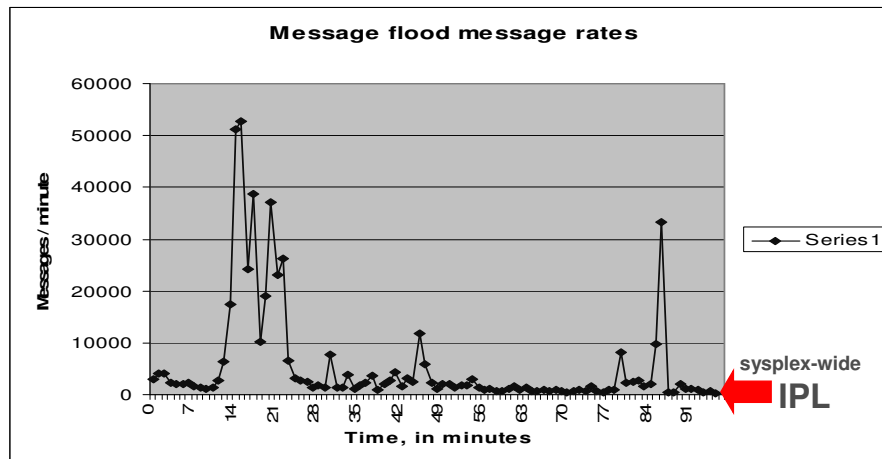
© 2008 IBM Corporation

Anatomy of a message flood

- **314971 total messages in approximately 14 minutes**
 - 109888 IOS450E 0329,ED, NOT OPERATIONAL PATH TAKEN OFFLINE
 - 41512 IOS444I DYNAMIC PATHING NOT OPERATIONAL ON PATH (0329,ED)
 - 32199 IOS001E 563F,INOPERATIVE PATH 3E
 - 24860 IST663I INIT OTHER REQUEST FAILED, SENSE=087D0001 179
 - 14731 IOS071I 4269,FA,XCFAS, HALT SUBCHANNEL INTERRUPT MISSING
 - 9831 IOS251I PATH 0B TAKEN OFFLINE FROM DEVICE E815
 - 7861 IEE103I UNIT 5238 NOT BROUGHT ONLINE 758
 - 3697 IOS051I INTERFACE TIMEOUT DETECTED ON 0329,ED,DB,**02,PCHID=0530
 - 1037 IST1494I PATH SWITCH STARTED FOR RTP CNR001E0 139
 - 893 IOS050I CHANNEL DETECTED ERROR ON 900D,60,E7,**02,PCHID=0541
 - 891 IOS002A 5248,NO PATHS AVAILABLE
 - 880 IEE025I UNIT 5278 HAS NO LOGICAL PATHS

Here are some of the most frequently occurring messages during this particular message flood.

Typical message flooding situation without Message Flood Automation



The graph shows a message flooding situation that occurred at a real z/OS parallel sysplex customer.

Within a very few minutes, a malfunctioning I/O device inundated the sysplex with tens of thousands of error messages. The customer struggled to regain control of their sysplex, but the volume of messages that were buffered and queued made recovery all but impossible. Ultimately, the customer re-IPL'd the entire sysplex. Unfortunately, a similar malfunction took most of their sysplex down again a few days later.

Properly set-up, Message Flood Automation is rarely invoked --

```

13:53:39.04 STC00027 00000010 +FLDR060I -- exponential distribution message: 221
13:53:39.06 STC00027 00000010 +FLDR060I -- exponential distribution message: 222
13:53:39.10 STC00027 00000010 +FLDR060I -- exponential distribution message: 223
13:53:39.17 STC00027 00000010 CNZZ007E MESSAGE RATE EXCEEDED      10 MESSAGES IN <1 SECONDS.
13:53:39.17 STC00027 00000010 +FLDR060I -- exponential distribution message: 224
13:53:39.29 STC00027 00000010 +FLDR060I -- exponential distribution message: 225
13:53:39.51 STC00027 00000010 +FLDR060I -- exponential distribution message: 226
13:53:39.97 STC00027 00000010 CNZZ002E MESSAGE THRESHOLD REACHED FOR JOB WTOXDSPK
13:53:41.70 STC00027 00000010 CNZZ008E REGULAR MESSAGE RATE ACCEPTABLE.      1 MESSAGES ACTED UPON.
13:53:41.70 STC00027 00000010 +FLDR060I -- exponential distribution message: 228
13:53:42.29 STC00027 00000010 +FLDR060I -- exponential distribution message: 229
13:53:42.39 STC00027 00000010 +FLDR060I -- exponential distribution message: 230

```

MSGTHRESH value

Note that message 227 has been acted upon.

...unless a flood occurs.

With the parameters set from the previous chart, and under the same message rate conditions that were observed by the Message Rate Monitoring function, Message Flood Automation is rarely triggered unnecessarily.

You can of course set the thresholds slightly higher and avoid even these rare activations.

When a flood occurs, Message Flood Automation acts quickly --

```

14:00:33.47 STC00027 00000010 +FLDR060I -- exponential distribution message:    998
14:00:33.69 STC00027 00000010 +FLDR060I -- exponential distribution message:    999
14:00:34.13 STC00027 00000010 +FLDR060I -- exponential distribution message:   1000
14:00:35.70 STC00027 00000010 +FLDR061I -- SPIKE message:          1
14:00:35.70 STC00027 00000010 +FLDR061I -- SPIKE message:          2
14:00:35.71 STC00027 00000010 +FLDR061I -- SPIKE message:          3
14:00:35.71 STC00027 00000010 +FLDR061I -- SPIKE message:          4
14:00:35.71 STC00027 00000010 +FLDR061I -- SPIKE message:          5
14:00:35.71 STC00027 00000010 +FLDR061I -- SPIKE message:          6
14:00:35.71 STC00027 00000010 +FLDR061I -- SPIKE message:          7
14:00:35.71 STC00027 00000010 +FLDR061I -- SPIKE message:          8
14:00:35.71 STC00027 00000010 +FLDR061I -- SPIKE message:          9
14:00:35.71 STC00027 00000010 +FLDR061I -- SPIKE message:         10
14:00:35.71 STC00027 00000010 +FLDR061I -- SPIKE message:         11
14:00:35.71 STC00027 00000010 +FLDR061I -- SPIKE message:         12
14:00:35.71 STC00027 00000010 +FLDR061I -- SPIKE message:         13
14:00:35.71 STC00027 00000010 +FLDR061I -- SPIKE message:         14
14:00:35.71 STC00027 00000010 +FLDR061I -- SPIKE message:         15
14:00:35.72 STC00027 00000010 CNZZ007E MESSAGE RATE EXCEEDED      10 MESSAGES IN <1 SECONDS.
14:00:35.72 STC00027 00000010 +FLDR061I -- SPIKE message:         16
14:00:35.72 STC00027 00000010 +FLDR061I -- SPIKE message:         17
14:00:35.72 STC00027 00000010 +FLDR061I -- SPIKE message:         18
14:00:35.72 STC00027 00000010 CNZZ002E MESSAGE THRESHOLD REACHED FOR JOB WTOX00PK
14:00:35.72 STC00027 00000010 CNZZ008E REGULAR MESSAGE RATE ACCEPTABLE.    85 MESSAGES ACTED UPON.
14:00:38.43 STC00027 00000010 +FLDR060I -- exponential distribution message:          4
14:00:41.59 STC00027 00000010 +FLDR060I -- exponential distribution message:          5
14:00:41.60 STC00027 00000010 +FLDR060I -- exponential distribution message:          6

```

Effective handling
of the flood --

Minimal loss of
uninvolved
messages

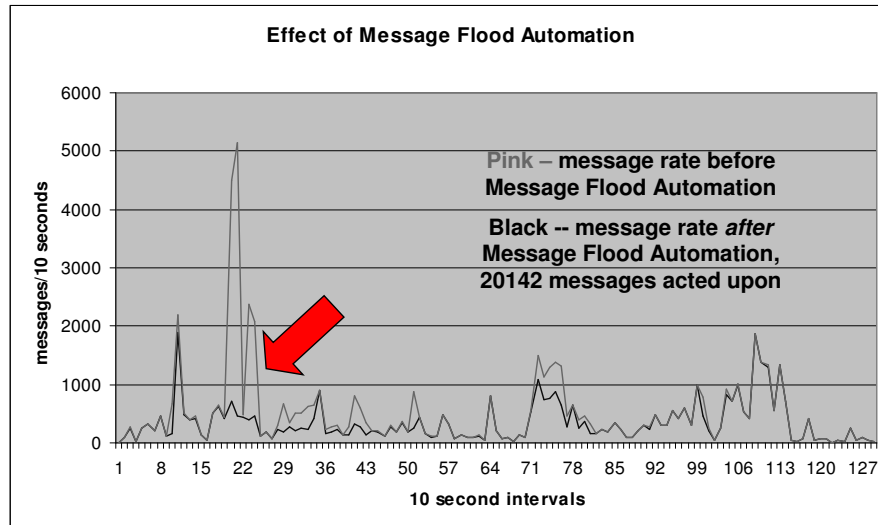
© 2008 IBM Corporation

When a message flood occurs, Message Flood Automation reacts quickly to take action, in this case in about 2/100ths of a second!

In this case, the policy causes messages from the offending program to be deleted, and 85 messages do not appear in the SYSLOG.

Note also, that when the flood ceases, Message Flood Automation quickly stops taking action, minimizing “collateral damage.”

Typical message flooding situation *with* Message Flood Automation



© 2008 IBM Corporation

The graph shows a message flooding situation that occurred at a real z/OS parallel sysplex customer. The pink line shows the overall message rate; the black line shows the message rate after Message Flood Automation has acted on the messages. Note that these message rates include messages which were being written to hardcopy-only and were not visible on a console (which is why Message Flood Automation seems to have little effect on the left-most peak – $\frac{3}{4}$ of these messages were already being suppressed). 20142 messages were acted upon within the interval shown.



z/OS Message Flood Automation - Recent News

➤ ***Message Flood Automation User's Guide***

- <http://publibz.boulder.ibm.com/zoslib/pdf/mfaguide.pdf>

- **OA19764 fixes a PARMLIB allocation bug**
- **OA21122 fixes a common storage management problem**
- **OA23112 fixes inadvertently taking action**

➤ **IBM Education Assistant –**

Charts (PDF) and audio presentation

<http://publib.boulder.ibm.com/infocenter/ieduasst/stqv1r0/index.jsp?>

Select: z/OS Operating System

Select: General

Select: Optimization and management capabilities

Select: Getting started with message flood automation

➤ **National Language Support for messages available in z/OS R9**



| z/OS Software

QUESTIONS?

© 2008 IBM Corporation