#SHAREorg

Diagnosing Problems on my z/OS system – New technologies

Anuja Deedwaniya
IBM Corp.
Anujad@us.ibm.com

August 2013
Session Number **14253**

SHARE in Boston

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.
Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are 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.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

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.

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

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 subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Notice Regarding Specialty Engines (e.g., zIIPs, zAAPs and IFLs):

Any information contained in this document regarding Specialty Engines ("SEs") and SE eligible workloads provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIIPs, zAAPs, and IFLs). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”).

No other workload processing is authorized for execution on an SE.

IBM offers SEs at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

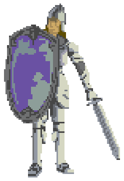
Agenda

- Problem determination pain points and Soft Failures
- Runtime Diagnostics
- Predictive Failure Analysis
- z/OSMF Incident Log
- IBM zAware
- Comparison of Runtime Diagnostics, Predictive Failure Analysis, and IBM zAware
- Interaction with other software management products
- Summary

Thanks to Karla Arndt and Robert Abrams for their contributions.

Types of Failures on System z and z/OS

Masked Failure



Software/Hardware detects failure

Software/Hardware corrects failure

No impact to business

Example: Hardware power supply failure: switch to alternate, IBM alerted, concurrent replacement

Hard Failure

Software/Hardware detects failure

Automations and operations restart the failing component

Minimal impact to business

Example: Application terminates but is restarted by ARM

Soft Failure

User detects failure, impact to business

Difficult to determine recovery actions

Example: component is failing, holds resources (locks, enqueues) required by other components, causes sysplex wide stall, leads to sysplex wide IPL

The Issue with Soft Failures

"Your systems don't break. They just stop working and we don't know why."

"Sick, but not dead" or Soft failures

Symptoms

- 80% of the business impact, but only 20% of the problems
- Long duration
- Infrequent
- Unique
- In software or hardware
- Cause creeping failures and sympathy sickness
- Hard to determine how to isolate
- Hard to determine how to recover
- Hard for software to detect internally
- Probabilistic, not deterministic

Manifested as

- Stalled or hung processes
  - Single system or sysplex members
  - Sympathy sickness
- Resource contention
- Storage growth
- CF, CDS growth
- I/O issues (paths, response times)
- Repetitive errors
- Queue growth
- Configuration issue
  - Single point of failure, threshold, cache structure size, not enabling new features

Problem Determination in a complex environment

Installation Pain Points

Risk to the business

- The impact of the symptoms
- Risk of recurrence
- Impact in getting system stabilized
- Mean time to recovery too long

Complexity of performing the task

Troubleshooting a live system and recovering from an apparent failure

Data collection very time-consuming

Significant skill level needed to analyze problems, interact with IBM and ISVs to obtain additional diagnostic info

Requirement Areas

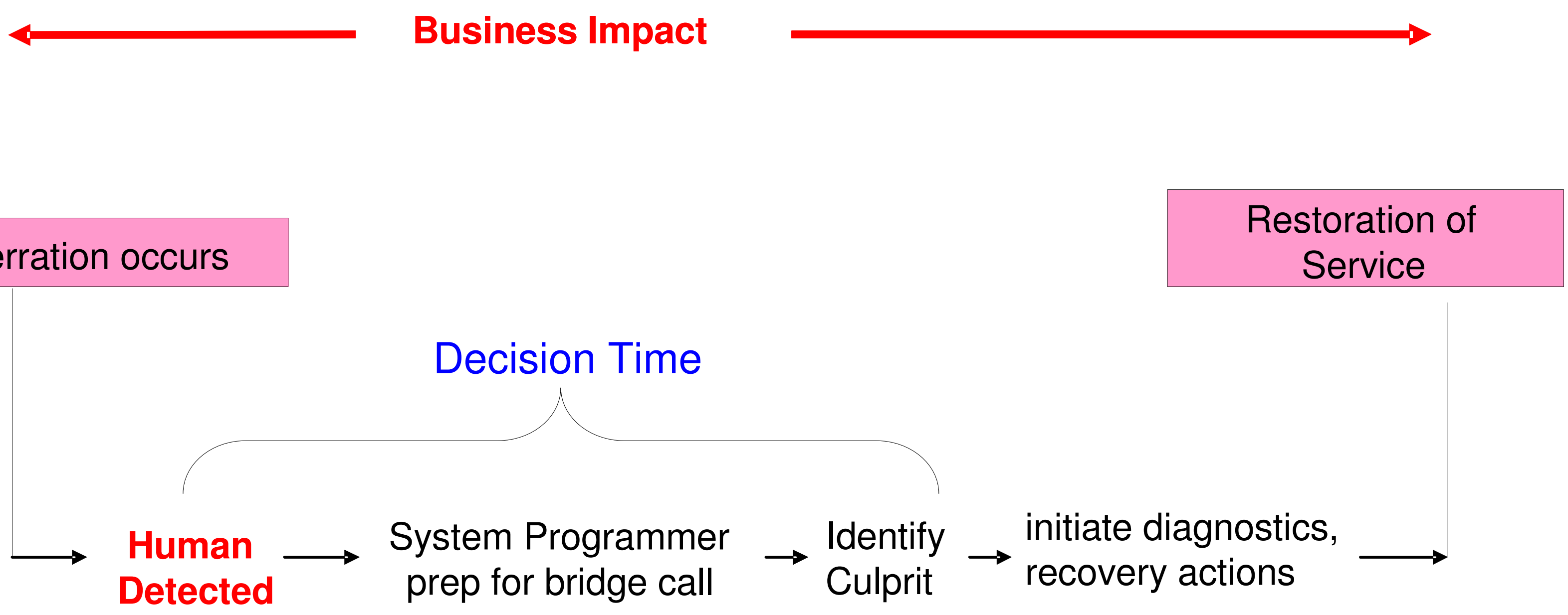
Detect “sick, but not dead” event **BEFORE** it causes problems

Diagnose the cause in **real time** to allow operations to mitigate event inquiries

Manage / **capture** data to determine cause of problem

- Allow problem to be fixed to prevent recurrence

Anatomy of an Outage



Business Impact

Aberration occurs

Restoration of Service

Decision Time

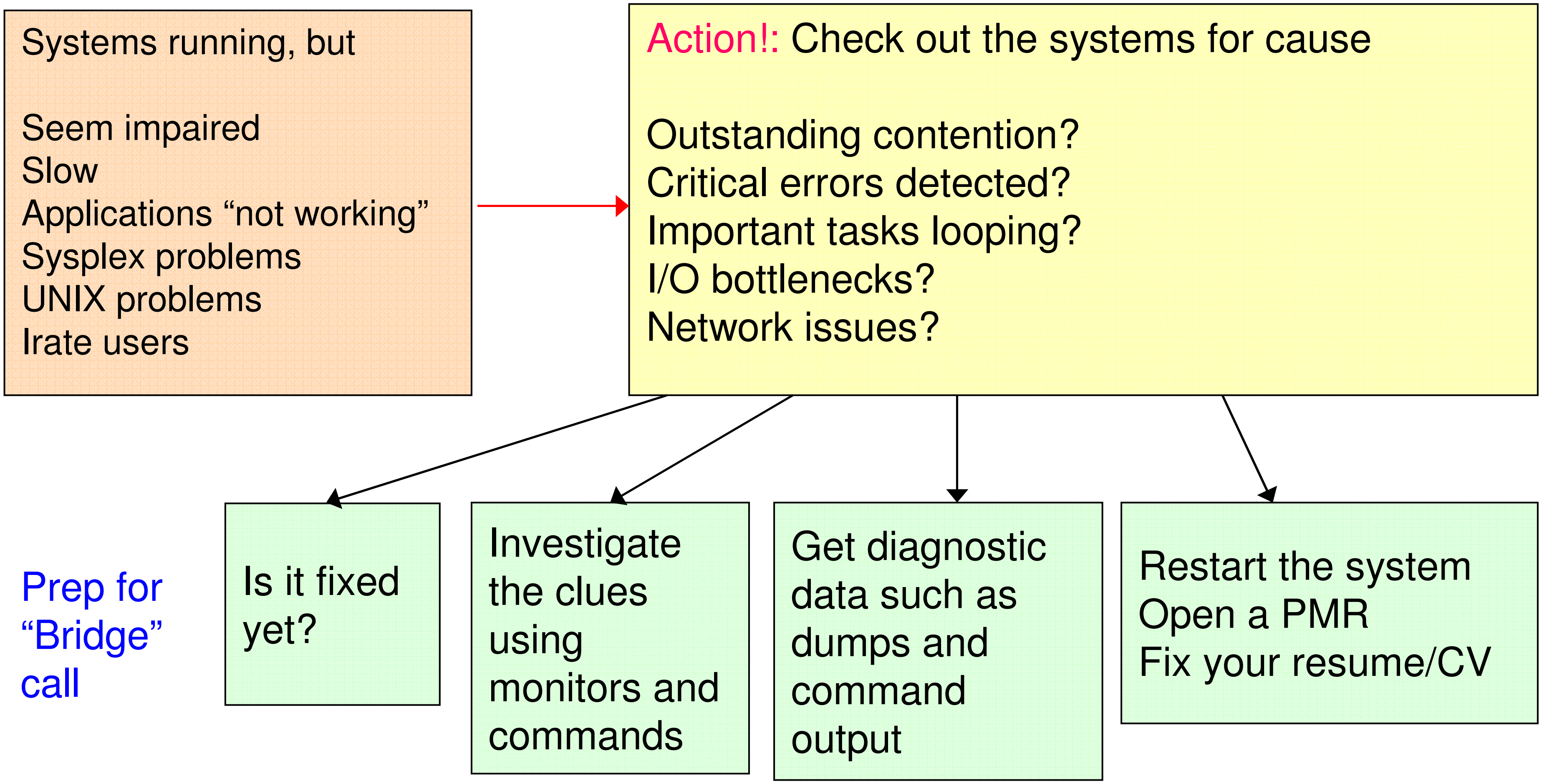
Human Detected → System Programmer prep for bridge call → Identify Culprit → initiate diagnostics, recovery actions →

***Customer Pain Points**:*

- *Fault occurs long before anyone notices*
- *Difficult to identify where the problem is coming from*

  → Leads to long decision time before recovery actions

Dealing with Soft Failures when they occur



Systems running, but

Seem impaired
Slow
Applications “not working”
Sysplex problems
UNIX problems
Irate users

Action!: Check out the systems for cause

Outstanding contention?
Critical errors detected?
Important tasks looping?
I/O bottlenecks?
Network issues?

Prep for “Bridge” call

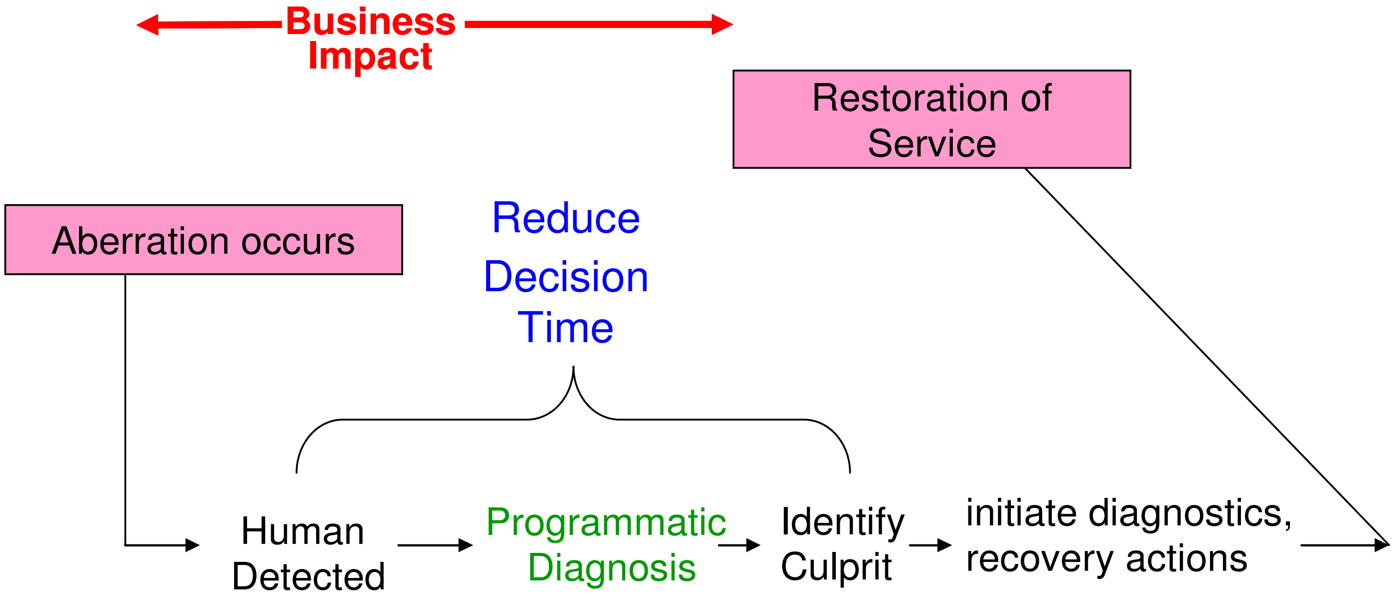
Is it fixed yet?

Investigate the clues using monitors and commands

Get diagnostic data such as dumps and command output

Restart the system
Open a PMR
Fix your resume/CV

Earlier Problem Diagnosis: Reduce Decision Time



Business Impact

Restoration of Service

Aberration occurs

Reduce Decision Time

Human Detected → Programmatic Diagnosis → Identify Culprit → initiate diagnostics, recovery actions →

*RAS Innovation --- **Runtime Diagnostics**:*

- *Machine-speed understanding*
- *Better tooling to identify the culprit*
- *Enables faster, correct recovery actions*

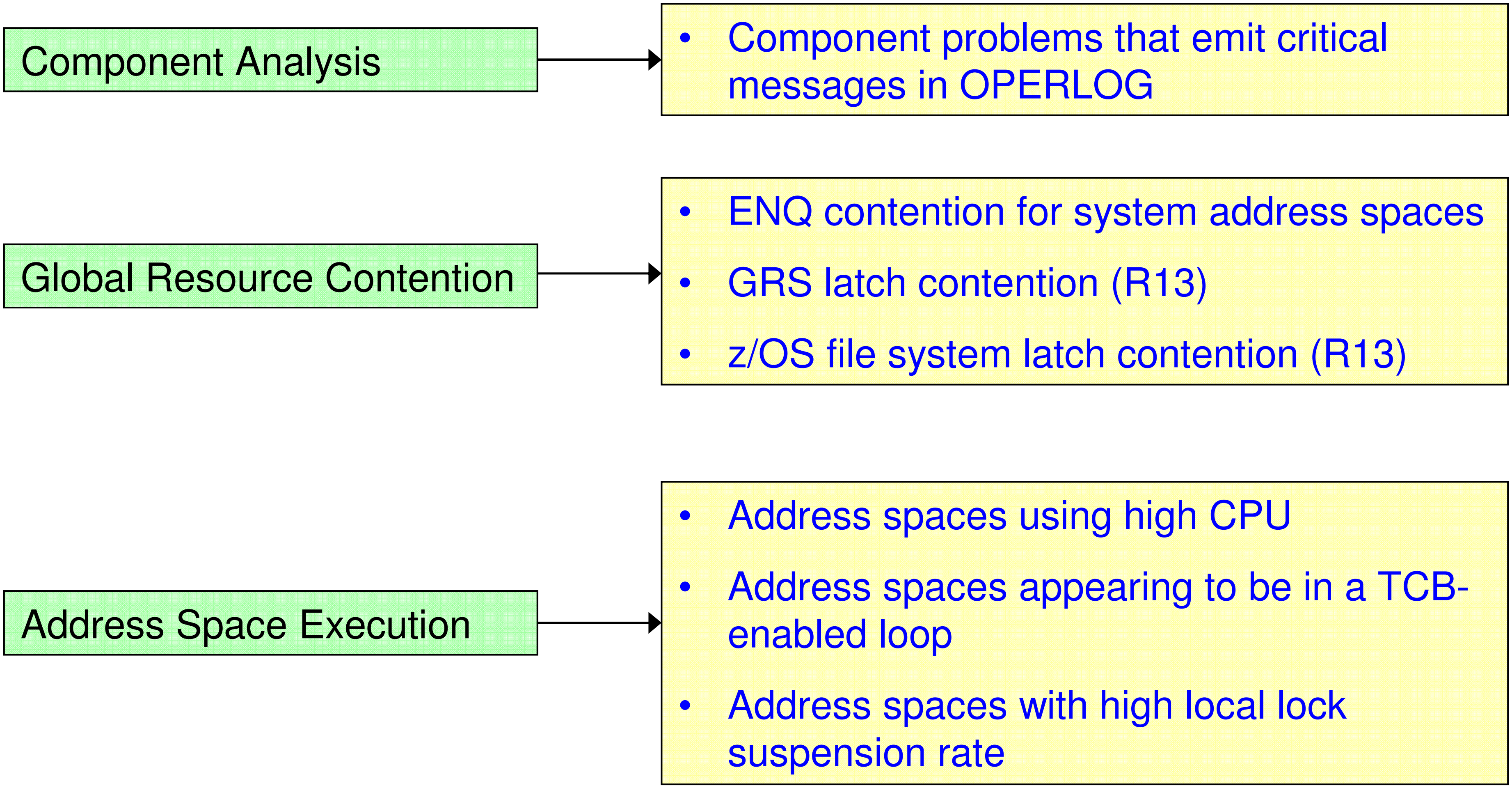
Runtime Diagnostics Usage and Benefits

- Runtime Diagnostics performs diagnostics on a “sick, but not dead” system in a timely manner
  - Reduces the skill level needed by a system programmer for investigating soft failures
  - Provides timely, comprehensive analysis at a critical time period
  - Performs analysis similar to that of a very experienced system programmer
    - But, more comprehensive and faster – goal of 60 seconds or less
    - great productivity aid for experienced system programmers!
  - Looks for specific evidence of “soft failures” occurring RIGHT NOW!

Runtime Diagnostics Usage and Benefits

- Runtime Diagnostics is not automation or a monitor and takes no corrective action,
- Runtime Diagnostics helps recommend next steps; Allows you to quickly discover next actions to take such as
  - Which jobs to cancel
  - What to investigate further such as classes of resources or a single address space using a monitor like RMF or Tivoli Omegamon
- There is no background processing and minimal dependencies on system services
- Use Runtime Diagnostics…
  - when the help desk or operations reports a problem on the system
  - to get ready for the “bridge call”
  - when PFA detects abnormal behavior

Runtime Diagnostics Analysis Types



Component Analysis

- Component problems that emit critical messages in OPERLOG

Global Resource Contention

- ENQ contention for system address spaces
- GRS latch contention (R13)
- z/OS file system latch contention (R13)

Address Space Execution

- Address spaces using high CPU
- Address spaces appearing to be in a TCB-enabled loop
- Address spaces with high local lock suspension rate

Runtime Diagnostics Invocation

- z/OS 1.12 – Started task – “Run” the analysis via a START command
  - START HZR,SUB=MSTR
  - Invokes HZR PROC
  - Will only run on R12 system, but other systems in the Sysplex do not need to be R12
  - Can override HZROUT to specify a data set, for example:
    - //HZROUT DD DISP=SHR,DSN=MY.DATA
    - START HZR,SUB=MSTR,DSN=MY.DATA,DISP=SHR
- z/OS 1.13 – Address space – started with the START command above
  - Address space needs to be available for PFA integration
    - Recommend to start address space at IPL
  - “Run” the analysis via a MODIFY command
    - f hzr,analyze
  - Migration Action: If you used Runtime Diagnostics in z/OS 1.12, ensure you update the hzrproc to point to PGM=HZRINIT instead of PGM=HZRMAIN.

Runtime Diagnostics Output

- A mutli-line WTO issued to the console that ran the command
  - If the MCS console has an out-of-line display area setup via K A,xx), the output will be displayed in the display area.

- The output can be directed to a sequential data set.
  - Use DISP=SHR to view output without stopping HZR started task.
    - //HZROUT DD DSN=MY.DATA,LRECL=121,BLKSIZE=0,RECFM=FB,DISP=SHR
    - START HZR,SUB=MSTR,DSN=MY.DATA,DISP=SHR

Runtime Diagnostics Example Output

Success → Events detected

```
HZR0200I RUNTIME DIAGNOSTICS RESULT 581
SUMMARY: SUCCESS
REQ: 004 TARGET SYSTEM: SY1 HOME: SY1 2010/12/21 - 13:51:32
INTERVAL: 60 MINUTES
EVENTS:
FOUND: 02 - PRIORITIES: HIGH:02 MED:00 LOW:00
TYPES: HIGHCPU:01
TYPES: LOOP:01
----------------------------------------------------------------------
EVENT 01: HIGH - HIGHCPU - SYSTEM: SY1 2010/12/21 - 13:51:33
ASID CPU RATE:99% ASID:002E JOBNAME:IBMUSERX
STEPNAME:STEP1 PROCSTEP: JOBID:JOB00045 USERID:IBMUSER
JOBSTART:2010/12/21 - 11:22:51
ERROR: ADDRESS SPACE USING EXCESSIVE CPU TIME. IT MIGHT BE LOOPING.
ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
----------------------------------------------------------------------
EVENT 02: HIGH - LOOP - SYSTEM: SY1 2010/12/21 - 13:51:14
ASID:002E JOBNAME:IBMUSERX TCB:004FF1C0
STEPNAME:STEP1 PROCSTEP: JOBID:JOB00045 USERID:IBMUSER
JOBSTART:2010/12/21 - 11:22:51
ERROR: ADDRESS SPACE MIGHT BE IN A LOOP.
ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
```

Example: When both a HIGHCPU event and a LOOP event are detected for the same system the job is very likely looping.

- The normal corrective action is to cancel the job

Runtime Diagnostics Events

z/OS 1.12

- Component-specific, critical messages in OPERLOG
  - Looks one hour back, if available
  - Additional analysis for some msgs
  - Message summary found in output
  - Can analyze messages in other systems in sysplex
- Enqueue Contention Checking
  - Looks for system address space waiting > 5 seconds
  - Lists both waiter and blocker
  - Can detect contention in other system in sysplex
- Local Lock Suspension
  - Any address space whose local lock suspension time is > 50%

z/OS 1.12 (continued)

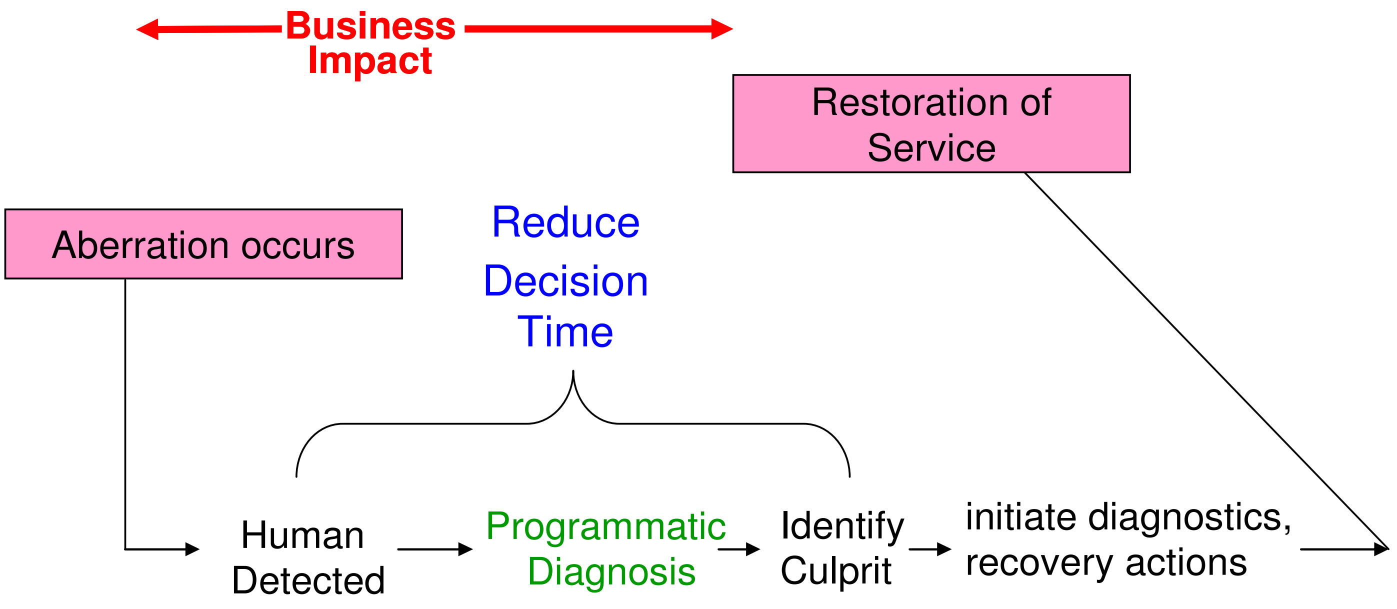
- CPU Analysis
  - Takes 2 samples over 1 sec. interval
  - Any task using > 95% is considered a potential problem
- Loop Detection
  - Investigates all tasks in all address spaces looking for TCB loops

z/OS 1.13

- z/OS UNIX Latch Contention
  - Looks for z/OS UNIX latch contention or waiting threads that exit for > 5 minutes.
- GRS Latch Contention
  - Obtains latch contention info from GRS
  - Omits z/OS UNIX file system latch contention
  - Returns longest waiter for each latch set

Runtime Diagnostics is simple to set up, simple to use, and ready to go when you need it most!

Faster Problem Diagnosis: Reduce Decision Time



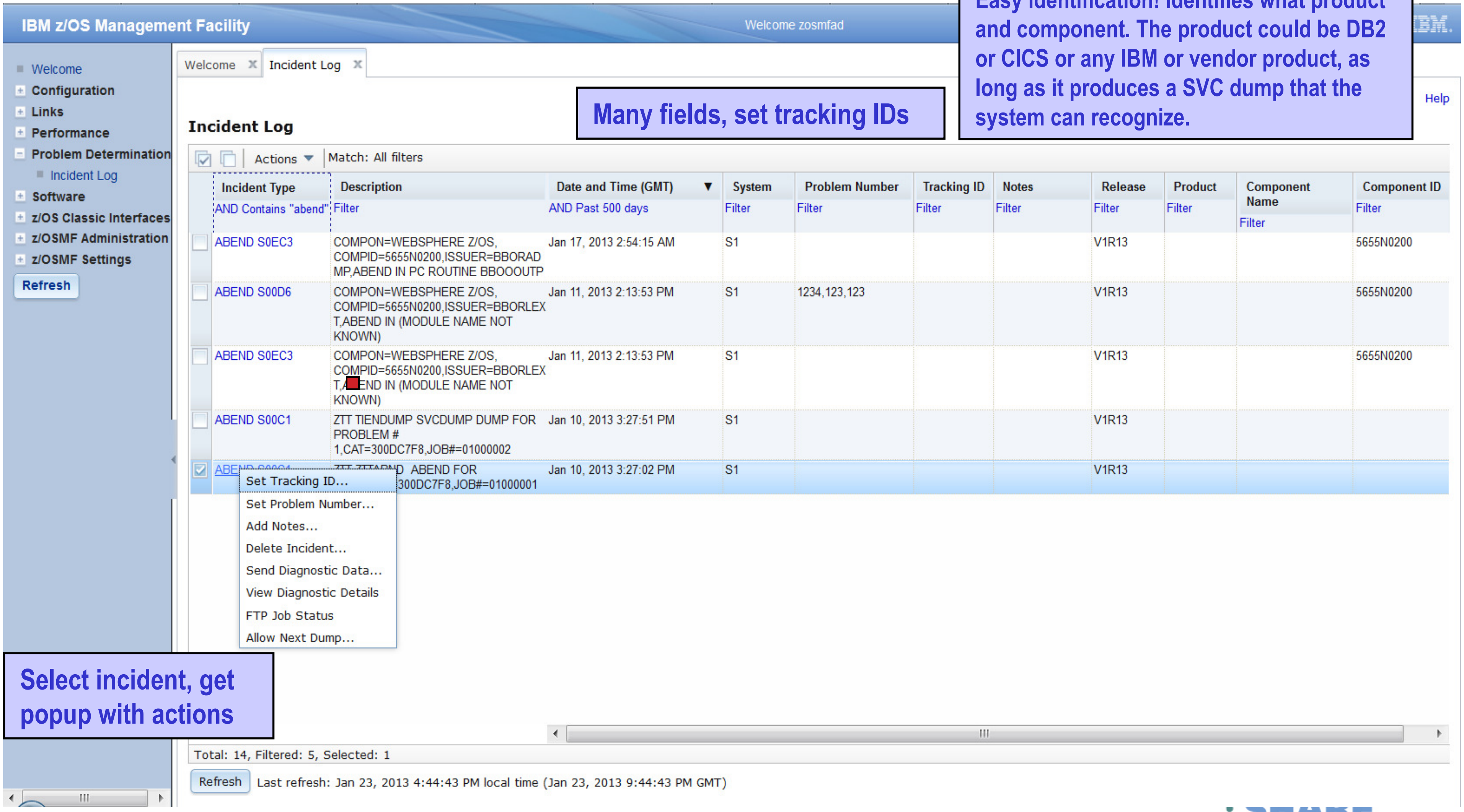
*RAS Innovation --- **z/OSMF Incident Log***

- *Identify system related problems*
- *First failure data capture and management of logs*
- *Easier and faster sending of diagnostic data for faster diagnosis*

z/OSMF Incident Log

- **Auto-capture basic diagnostic materials, triggered when the dump is written to a data set, managed via PARMLIB member**
  - **focus is Abend and user initiated SVC dumps**
  - Improved FFDC for system-detected problems
  - Diagnostic data “snapshots” for transient data: Snapshots of 30 min Operlog or Syslog, 1 hr Logrec detail, and 4-hour Logrec summary
  - Allow doc to be tersed and FTP'd to IBM (or ISV) without having to keep track of where logs are archived via easy to use interface
  - Simplify informing DAE to take the next dump for the incident's symptom string
- **z/OSMF Browser interface functions include:**
  - Display list of incidents across sysplex(Filter/ sort/ configure/ delete)
  - Display properties – view list of diagnostic data, logs
  - Set properties: associate problem number and tracking id , new fields and more customization capabilities
  - Send diagnostic data via FTP: Manage FTP jobs status and define FTP Profiles (firewall), support for encrypted and parallel FTP
  - Send additional user-defined diagnostic data
  - Allow next dump

Incident Log – Summary Information



Incident Log

Match: All filters

	Incident Type	Description	Date and Time (GMT) ▼	System	Problem Number	Tracking ID	Notes	Release	Product	Component Name	Component ID
	AND Contains "abend"	Filter	AND Past 500 days	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
□	ABEND S0EC3	COMPON=WEBSPHERE Z/OS, COMPID=5655N0200,ISSUER=BBORADMP,ABEND IN PC ROUTINE BBOOOUTP	Jan 17, 2013 2:54:15 AM	S1				V1R13			5655N0200
□	ABEND S00D6	COMPON=WEBSPHERE Z/OS, COMPID=5655N0200,ISSUER=BBORLEXT,ABEND IN (MODULE NAME NOT KNOWN)	Jan 11, 2013 2:13:53 PM	S1	1234,123,123			V1R13			5655N0200
□	ABEND S0EC3	COMPON=WEBSPHERE Z/OS, COMPID=5655N0200,ISSUER=BBORLEXT,A■END IN (MODULE NAME NOT KNOWN)	Jan 11, 2013 2:13:53 PM	S1				V1R13			5655N0200
□	ABEND S00C1	ZTT TIENDUMP SVCDUMP DUMP FOR PROBLEM # 1,CAT=300DC7F8,JOB#=01000002	Jan 10, 2013 3:27:51 PM	S1				V1R13			
☑	ABEND S00C1	ZTT ZTTABND ABEND FOR 300DC7F8,JOB#=01000001	Jan 10, 2013 3:27:02 PM	S1				V1R13			

Total: 14, Filtered: 5, Selected: 1

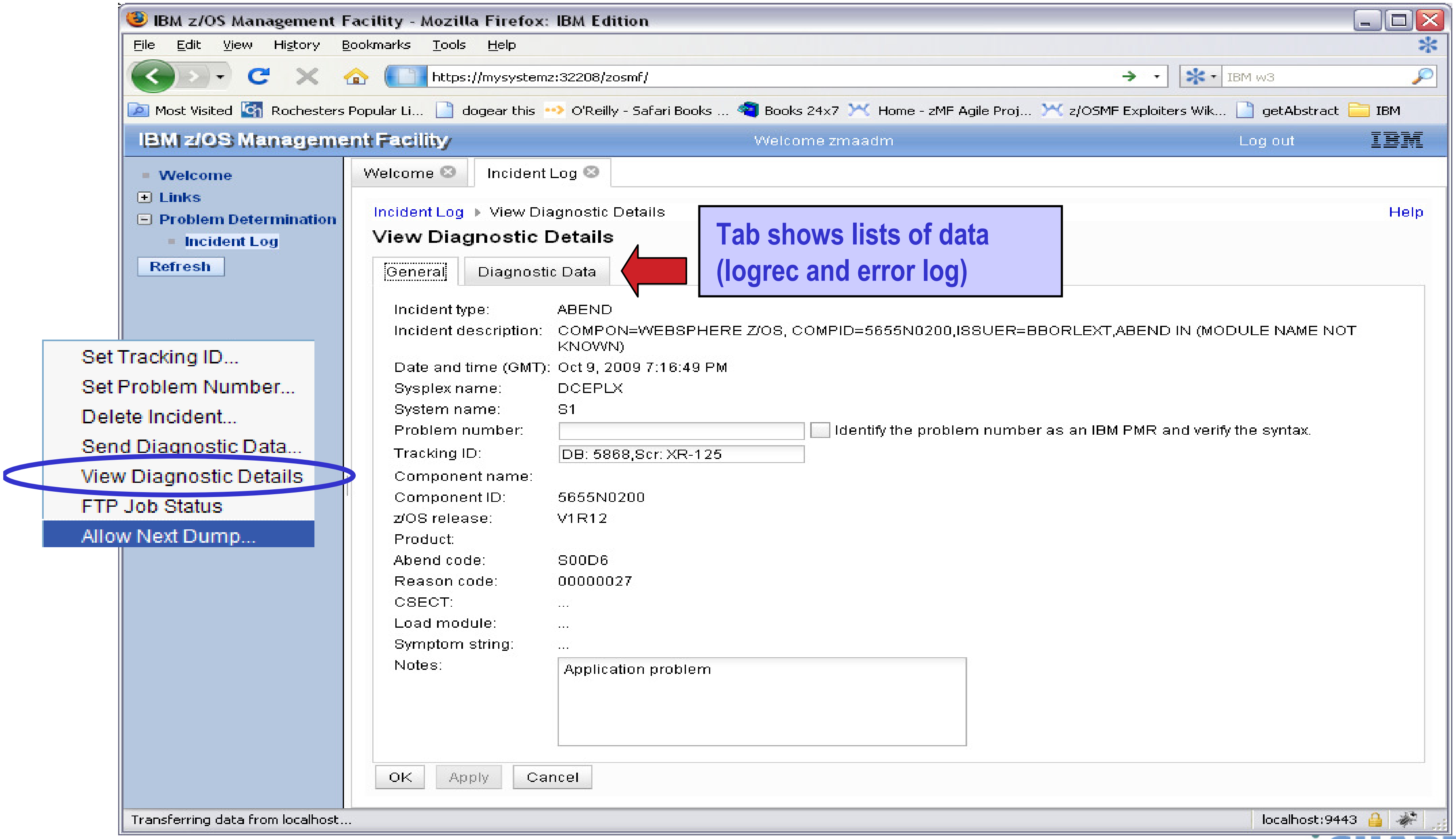
Refresh Last refresh: Jan 23, 2013 4:44:43 PM local time (Jan 23, 2013 9:44:43 PM GMT)

Many fields, set tracking IDs

Easy identification! Identifies what product and component. The product could be DB2 or CICS or any IBM or vendor product, as long as it produces a SVC dump that the system can recognize.

Select incident, get popup with actions

Incident Log – Incident Details



View Diagnostic Details

Incident type: ABEND

Incident description: COMPON=WEBSPHERE Z/OS, COMPID=5655N0200,ISSUER=BBORLEXT,ABEND IN (MODULE NAME NOT KNOWN)

Date and time (GMT): Oct 9, 2009 7:16:49 PM

Sysplex name: DCEPLX

System name: S1

Problem number: □ Identify the problem number as an IBM PMR and verify the syntax.

Tracking ID: DB: 5868,Scr: XR-125

Component name:

Component ID: 5655N0200

z/OS release: V1R12

Product:

Abend code: S00D6

Reason code: 00000027

CSECT: ...

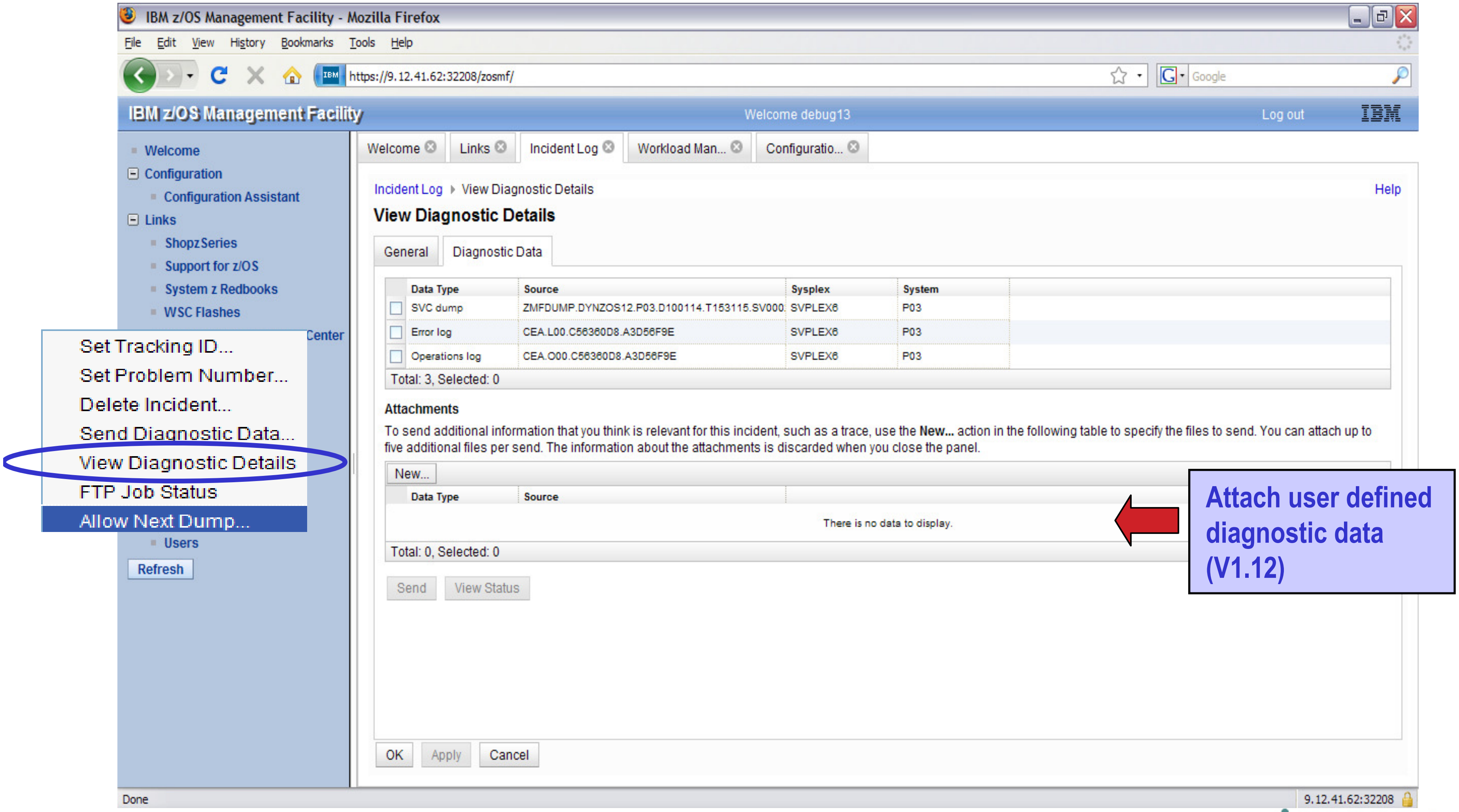
Load module: ...

Symptom string: ...

Notes: Application problem

Tab shows lists of data (logrec and error log)

Incident Log – Diagnostic Data



View Diagnostic Details

	Data Type	Source	Sysplex	System
□	SVC dump	ZMFDUMP.DYNZOS12.P03.D100114.T153115.SV000.	SVPLEX6	P03
□	Error log	CEA.L00.C56360D8.A3D56F9E	SVPLEX6	P03
□	Operations log	CEA.O00.C56360D8.A3D56F9E	SVPLEX6	P03

Total: 3, Selected: 0

Attachments

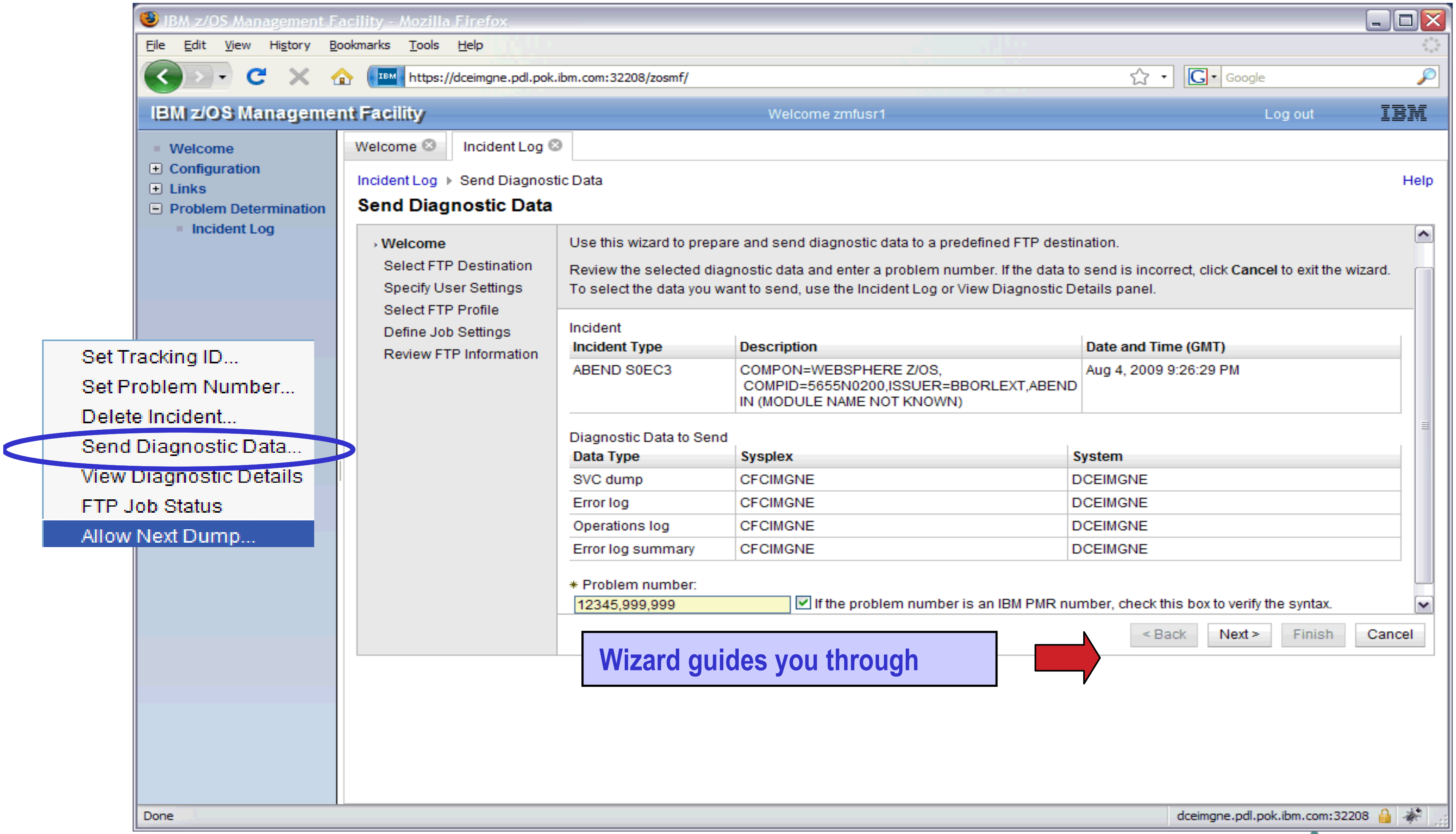
To send additional information that you think is relevant for this incident, such as a trace, use the **New...** action in the following table to specify the files to send. You can attach up to five additional files per send. The information about the attachments is discarded when you close the panel.

Data Type	Source
There is no data to display.	

Total: 0, Selected: 0

Attach user defined diagnostic data (V1.12)

Incident Log – Send Diagnostic Data



Send Diagnostic Data

Use this wizard to prepare and send diagnostic data to a predefined FTP destination.

Review the selected diagnostic data and enter a problem number. If the data to send is incorrect, click **Cancel** to exit the wizard. To select the data you want to send, use the Incident Log or View Diagnostic Details panel.

Incident

Incident Type	Description	Date and Time (GMT)
ABEND S0EC3	COMPON=WEBSPHERE Z/OS, COMPID=5655N0200,ISSUER=BBORLEXT,ABEND IN (MODULE NAME NOT KNOWN)	Aug 4, 2009 9:26:29 PM

Diagnostic Data to Send

Data Type	Sysplex	System
SVC dump	CFCIMGNE	DCEIMGNE
Error log	CFCIMGNE	DCEIMGNE
Operations log	CFCIMGNE	DCEIMGNE
Error log summary	CFCIMGNE	DCEIMGNE

* Problem number: 12345,999,999

☑ If the problem number is an IBM PMR number, check this box to verify the syntax.

Wizard guides you through

Complete your sessions evaluation online at SHARE.org/BostonEval

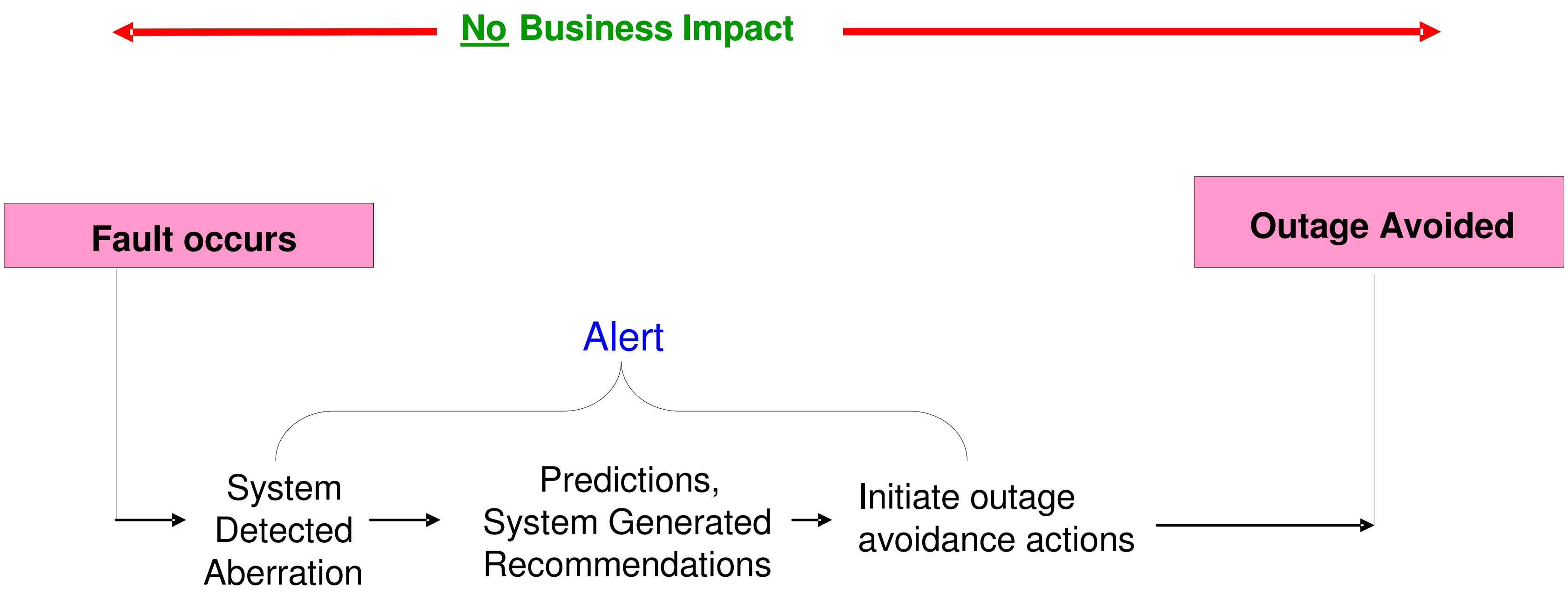
SHARE in Boston

z/OSMF Problem Determination – Incident log *Benefits*

	Without z/OSMF Incident Log **	With z/OSMF Incident Log **
Recognizing a system-detected (dumped) problem occurred	Requires 5 to 7 manual steps, plus skill on effective use of IPCS to extract data from each of the dumps. **Up to 5-6 minutes**	Display in 1 click. Greatly reduced skill required **As little as 5 seconds**
Allow new dump to be taken for the same symptom	Requires 7 to 12 manual steps, plus skill on effective use of IPCS to locate the dump data set, obtain the symptom string, get into the IPCS DAE display, locate the matching symptom string (could be non-trivial) and indicate TakeNext on the IPCS display **Up to 15 minutes**	Make the update happen in 3 mouse clicks **As little as 10 seconds**
Collecting and sending diagnostic data	Requires 7 to 15 manual steps, plus skill to locate the right log files, build and run jobs, rename the output datasets, and use an FTP job to send the different data sets to the target destination. **Up to 20 minutes** **Up to 30 minutes for sysplex components**	Send the material in 8 clicks: •Select the incident materials •Specify the FTP destination information •Send the material •Check whether the information was FTP'd successfully **As little as 30 seconds**

** Based on IBM laboratory results, your results may vary

Early Detection → Outage Avoidance



No Business Impact

Fault occurs

Outage Avoided

Alert

System Detected Aberration → Predictions, System Generated Recommendations → Initiate outage avoidance actions

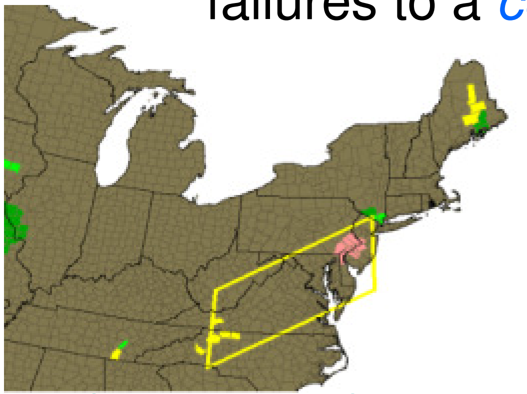
*RAS Innovation: **Predictive Failure Analysis***

- *Machine Learning - Convert diagnostic data to knowledge in real time*
- *Convert soft failures to correctable incidents*

Predictive Failure Analysis

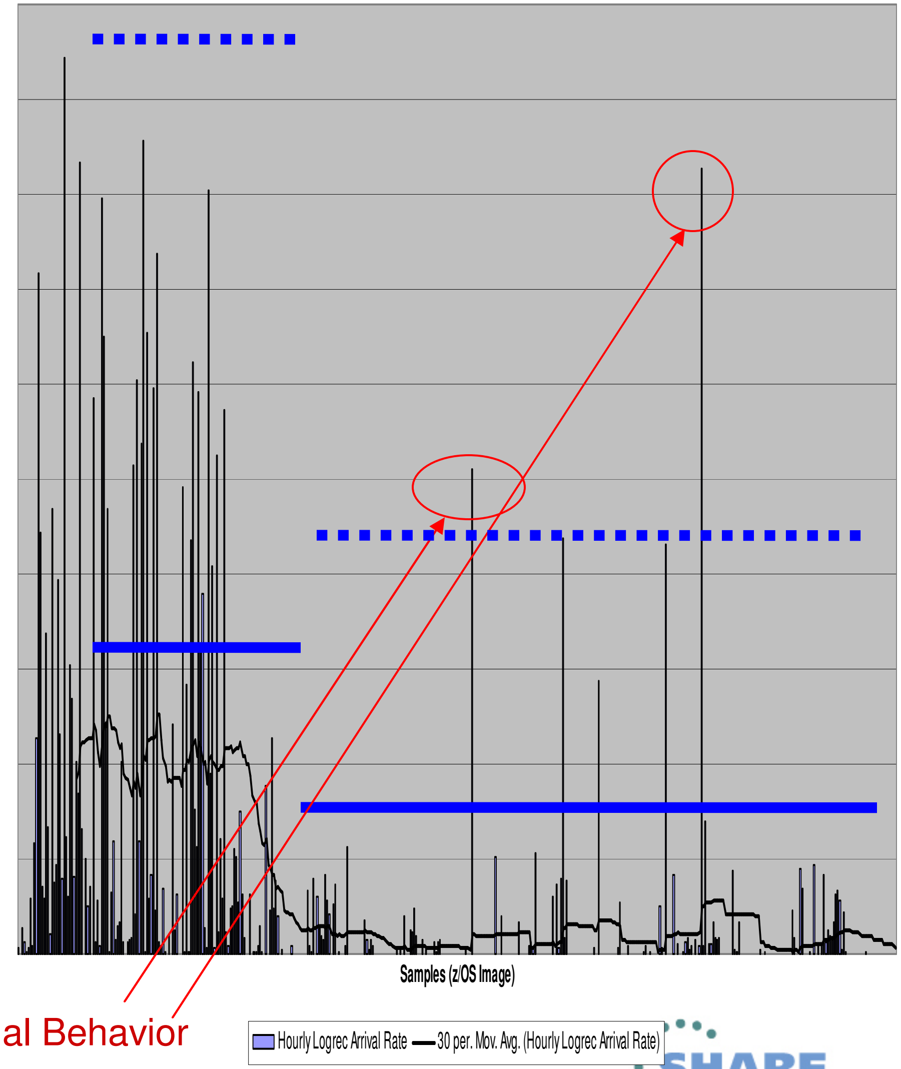
- Causes of soft failures
  - *Damaged systems*
    - Recurring or recursive errors anywhere in the software stack
  - Serialization
    - Priority inversion, classic deadlocks, owner gone
  - *Resource exhaustion*
    - Physical and software resources
  - Indeterminate or unexpected states

- PFA uses
  1. Historical data from each LPAR
  2. Machine learning and mathematical modeling
  - To *detect and alert* you to *abnormal behavior* and its potential causes
  - To *enable you* to convert soft failures to a *correctable incident*



PFA → Damaged Systems

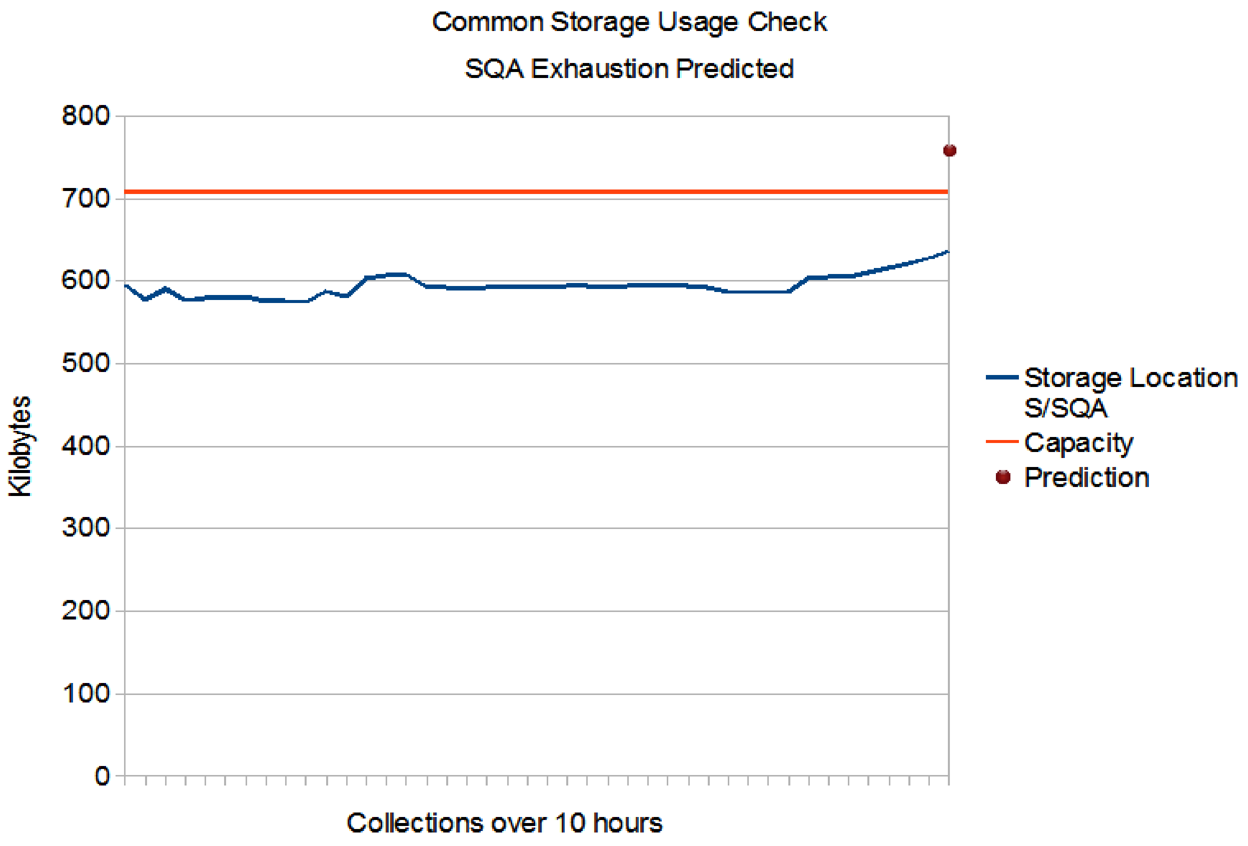
- Does *trend analysis* and models behavior to create *expected* value or rate
- Determines workload change vs. abnormal condition
  - Clusters current and past trends
  - Uses multiple models over time ranges when needed
    - 1 hour, 24 hours, 7 days
  - Uses CPU normalization and calculations when needed
    - Rates = Metric / CPU
- Five metrics
  - LOGREC arrival rate, Message arrival rate, SMF arrival rate, Enqueue request rate, JES spool usage



Abnormal Behavior

PFA → Resource Exhaustion

- Projects *current trend into the future*
- One metric: Common storage exhaustion



Capacity = 708
Current usage = 644
Prediction = 759

Exception issued when current trend was at 91% of capacity.

What happens when PFA detects a problem?

- *Health check exception* written to console
  - New exceptions suppressed until new model is available

- *Prediction report* available in SDSF (s.ck)
  - *"Top address spaces"* = potential villains
  - *Address spaces causing exception*
  - *Current and predicted values* provided
  - Reports also available when no problem occurs

- *Modeling automatically runs* more frequently

- *Logs and data files* for service written to EXC_*timestamp* directories

- *Best practices and more information* in *z/OS Problem Management*

Example report: Message Arrival Rate

```
                       Message Arrival Rate Prediction Report

Last successful model time       :  04/05/2012 07:08:01
Next model time                  :  04/05/2012 19:08:04
Model interval                   :  720
Last successful collection time  :  04/05/2012 08:22:15
Next collection time             :  04/05/2012 08:37:16
Collection interval              :  15

Message arrival rate
   at last collection interval      :       83.52
Prediction based on 1 hour of data  :       98.27
Prediction based on 24 hours of data:       85.98
Prediction based on 7 days of data  :      100.22

Top persistent users:
                                           Predicted Message
                      Message                Arrival Rate
  Job                 Arrival
  Name     ASID           Rate        1 Hour       24 Hour         7 Day

  -------- ----  ------------- ------------- ------------- -------------
  TRACKED1 001D          58.00         23.88         22.82         15.82
  TRACKED2 0028          11.00          0.34         11.11         12.11
  TRACKED3 0029          11.00         12.43          2.36          8.36
...
```

Heading information

System-level information

Address space information

Example report: Common Storage Usage

- Top predicted users
  - Tries to pinpoint potential villains
  - Those whose usage has *increased* the most in the last hour
- Other information
  - Expansion information
  - IBM Health Checker for z/OS message in its entirety

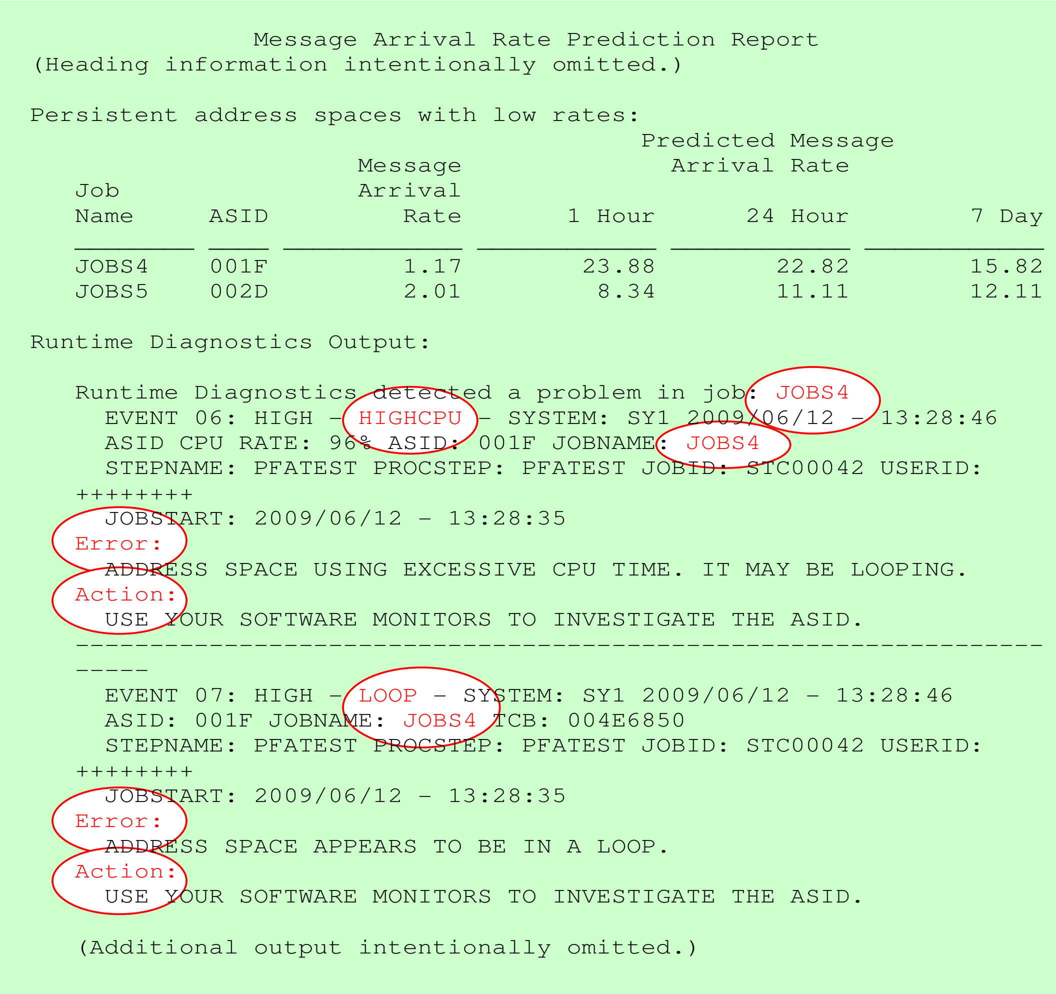
```
                 Common Storage Usage Prediction Report
(heading information intentionally omitted)
                                               Capacity When    Percentage
Storage      Current Usage     Prediction        Predicted      of Current
Location      in Kilobytes   in Kilobytes    in Kilobytes    to Capacity
__________   _____________   _____________   _____________   _____________
*CSA                  2796            3152            2956             95%
SQA                    455             455            2460             18%
CSA+SQA               3251            3771            5116             64%
ECSA                114922          637703          512700             22%
ESQA                  8414            9319           13184             64%
ECSA+ESQA           123336          646007          525884             23%

Storage requested from SQA expanded into CSA and is being included in
CSA usage and predictions. Comparisons for SQA are not being
performed.

Address spaces with the highest increased usage:
  Job          Storage     Current Usage     Predicted Usage
  Name         Location    in Kilobytes      in Kilobytes
  __________   ________   _______________   _______________
  JOB3         *CSA                  1235              1523
  JOB1         *CSA                   752               935
  JOB5         *CSA                   354               420
  JOB8         *CSA                   152               267
  JOB2         *CSA                    75                80
  JOB6         *CSA                    66                78
  JOB15        *CSA                    53                55
  JOB18        *CSA                    42                63
  JOB7         *CSA                    36                35
  JOB9         *CSA                    31                34

* = Storage locations that caused the exception.
```

The PFA Checks and Enhancements

- z/OS 1.10 SPE
  - Common storage exhaustion check
    - CSA+SQA → below the line
    - ECSA+ESQA above the line
  - LOGREC arrival rate check
    - Groups arrivals by key
    - Four time ranges
- z/OS 1.11
  - Message Arrival Rate check
    - (WTO + WTOR) / CPU = Rate
    - Tracked, persistent address spaces
    - Other persistent address spaces
    - Non-persistent address spaces
    - Total system
  - ~~Frames and Slots usage check~~
    - Removed from all releases in 2012 with OA40065
- z/OS 1.12
  - SMF arrival rate check
    - SMF arrival count / CPU = Rate
    - Same categories as Message Arrival Rate
  - Common storage exhaustion = 6 locations
  - Dynamic modeling improvements
  - Supervised learning (exclude jobs)
  - Performance and serviceability
- z/OS 1.13
  - JES spool usage check
    - JES2 only
    - Tracks all persistent address spaces
  - Enqueue request rate check
    - Tracked, persistent address spaces
    - Total system
  - Integration with Runtime Diagnostics to detect “too low”

PFA and Runtime Diagnostics Integration

- **Runtime Diagnostics output** included in PFA report
- Prediction report and result message **available in SDSF** (sdsf.ck)
- **PFA current rates and predictions** relevant to category causing exception



```
               Message Arrival Rate Prediction Report
(Heading information intentionally omitted.)

Persistent address spaces with low rates:
                                         Predicted Message
                      Message              Arrival Rate
   Job                Arrival
   Name     ASID         Rate       1 Hour      24 Hour        7 Day
   ________ ____ _____________ ____________ ____________ ____________
   JOBS4    001F         1.17        23.88        22.82        15.82
   JOBS5    002D         2.01         8.34        11.11        12.11

Runtime Diagnostics Output:

   Runtime Diagnostics detected a problem in job: JOBS4
     EVENT 06: HIGH - HIGHCPU - SYSTEM: SY1 2009/06/12 - 13:28:46
     ASID CPU RATE: 96% ASID: 001F JOBNAME: JOBS4
     STEPNAME: PFATEST PROCSTEP: PFATEST JOBID: STC00042 USERID:
   ++++++++
     JOBSTART: 2009/06/12 - 13:28:35
   Error:
     ADDRESS SPACE USING EXCESSIVE CPU TIME. IT MAY BE LOOPING.
   Action:
     USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
   -----------------------------------------------------------------
   -----
     EVENT 07: HIGH - LOOP - SYSTEM: SY1 2009/06/12 - 13:28:46
     ASID: 001F JOBNAME: JOBS4 TCB: 004E6850
     STEPNAME: PFATEST PROCSTEP: PFATEST JOBID: STC00042 USERID:
   ++++++++
     JOBSTART: 2009/06/12 - 13:28:35
   Error:
     ADDRESS SPACE APPEARS TO BE IN A LOOP.
   Action:
     USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.

   (Additional output intentionally omitted.)
```

How to Get the Most Out of PFA

- Use a zAAP to offload PFA's Java Processing
- Start z/OS Resiliency functions at IPL
  - IBM Health Checker for z/OS
  - PFA
  - Runtime Diagnostics (z/OS 1.13)
- Automate the PFA IBM Health Checker for z/OS exceptions
  - Simplest: Add exception messages to existing message automation product
  - More complex: Use exception messages and other information to tailor alerts
  - See *z/OS Problem Management* for exceptions issued for each check
- Create a policy in an HZSPRMxx member for persistent changes
  - Not all check-specific parameters are required on an UPDATE of PFA checks!
    - UPDATE CHECK=(IBMPFA,PFA_COMMON_STORAGE_USAGE)   **PARM**('THRESHOLD(3)')
- Get the latest PTFs
- See backup charts for more

zAware - background

- Class of problems with no clear failure,
- system may just fail due to systemic problems occurring over time
  - the ‘sick but not dead’ phenomenon
- Often clues are in the logs,
  - it is impossible to manually mine the system log for possible abnormal behaviors
  - Need some analytics to perform that function.
- The analytics should ideally be run out of band - outside the main OS,
  - so it is not impacted due to the system problems it is trying to detect
- Any analytics solution should proactively indicate when an anomaly (departure from routine behavior) is perceived.

What is zAware?

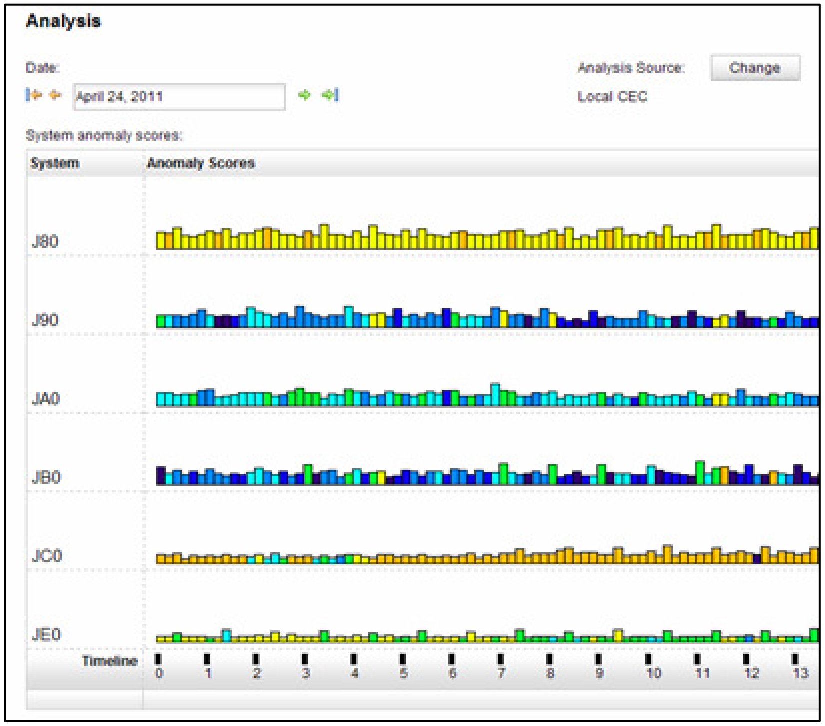
- **zAware is a classic deep analytics delivery which contains sophisticated analytics, IBM insight into the problem domain, and web browser based visualization.**
- **zAware runs as a firmware partition on zEnterprise EC12**
- **Uses modeling and machine learning to perform near real time high speed analytics on log data**
  - Uses historical data, pattern recognition, mathematical modeling to pinpoint deviations (anomalies) unique to the customer system
- **Requires z/OS 1.13 (with ptfs)**
- **Provides REST apis for monitoring and problem determination products to consume the anomaly information**
- **Uses role based authentication; Authentication by LTPA token or user supplied userid and password**

zAware characteristics

- The analytics creates a statistical model of the normal message traffic generated by each individual z/OS.
- Using this model which is stored in a database unexpected messages and patterns of messages are identified.
- Using a sliding ten minute interval which is updated every two minutes, a current score for the interval is created based on how unusual the message traffic is.
- For each interval zAware provides details of all of the unique message ids found within interval including how many, how rare, how much they contributed to the intervals score, when they first appeared.
- For IBM messages there is a link to the message description which often includes a recommended action to correct the issue highlighted by the message.
  ◄Output can be queued up to existing monitoring systems.
  Early detection and focused diagnosis can help improve time to recovery

zAware output

- **Monitors z/OS OPERLOG including all messages written to z/OS console, including ISV and application generated messages**
- **Detects things typical monitoring systems miss due to:**
  - Message suppression (message too common) - Useful for long-term health issues
  - Uniqueness (message not common enough) - Useful for real-time event diagnostics
- **Color coded easy to use GUI via web browsers**
- **Output can be queued up to existing monitoring systems.**
- **Early detection and focused diagnosis can help improve time to recovery**

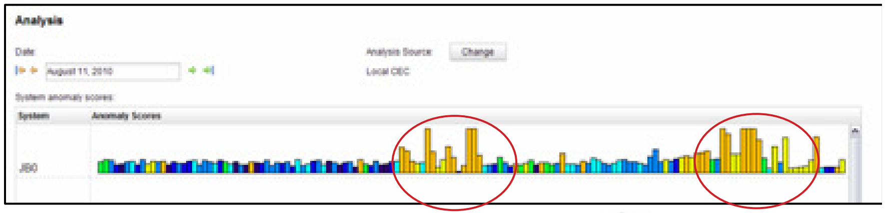


Analysis

Date: April 24, 2011

Analysis Source: Local CEC [Change]

System anomaly scores:



Analysis

Date: August 11, 2010

Analysis Source: Local CEC [Change]

System anomaly scores:

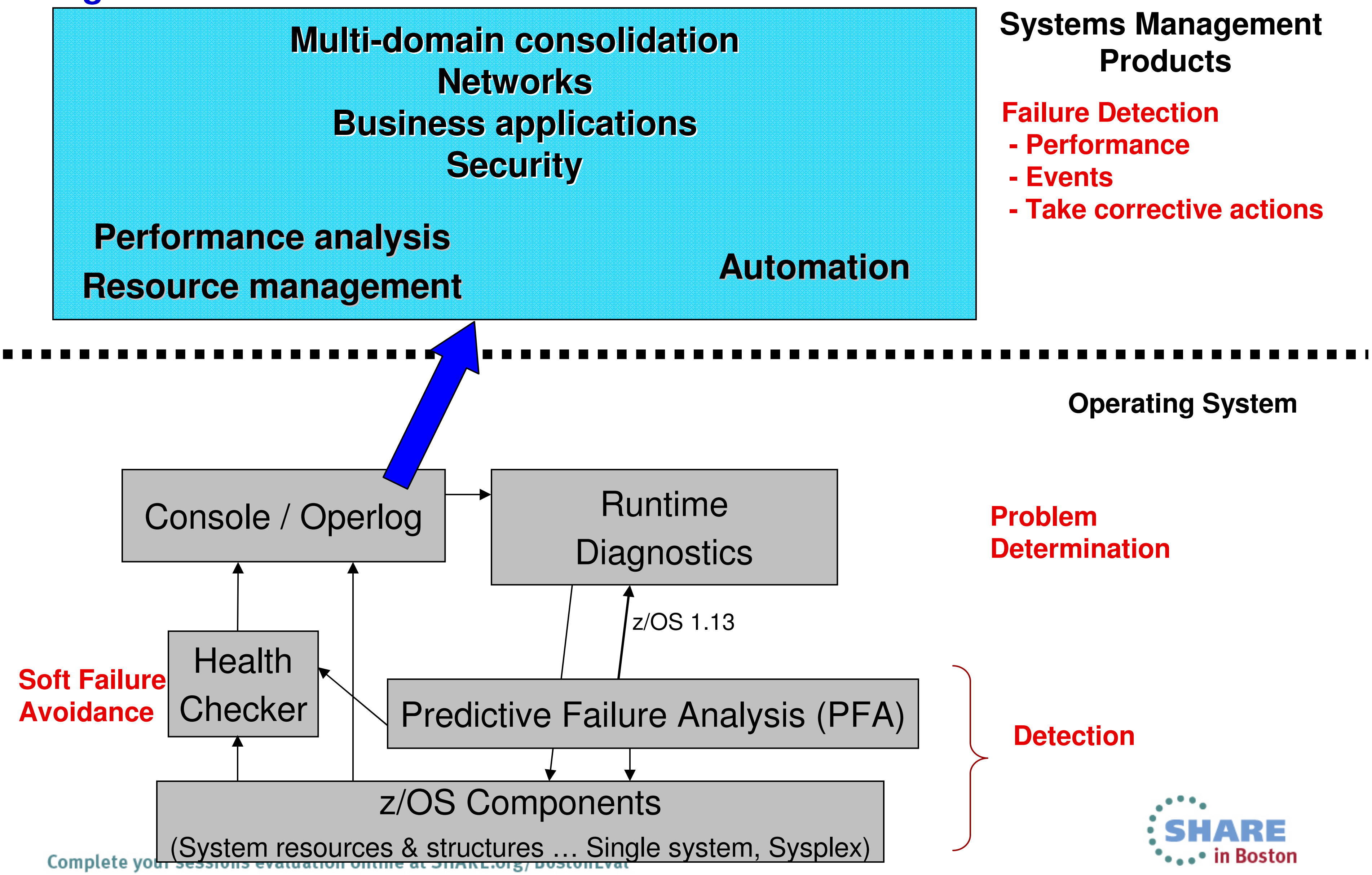
IBM z/OS Solutions Address Problem Determination

Solutions Available:		Rules-based	Analytics / Statistical model	Examines message traffic	Self Learning	Method
z/OS Health Checker (Avoid)	• Checks configurations • Programmatic, applies to IBM and ISV tools • Can escalate notifications • Part of z/OS	✓				Screen for conditions
z/OS Runtime Diagnostics (Diagnose)	• Real time diagnostics of specific z/OS system issues • 7 types of diagnostics • Part of z/OS	✓		By specific, IBM-defined messages back 1 hour		Diagnostics right after an incident
z/OS PFA (Predict and Detect)	• Trending analysis of z/OS system resources • Can invoke z/OS Runtime Diagnostics • 6 types of analysis • Part of z/OS		✓	By count of arrivals / CPU	✓	Early detection and alerting
IBM zAware (Avoid (when changes made to system) and Diagnose)	• Pattern-based message analysis • Provides aid in diagnosing complex z/OS problems including cross sysplex problems that may or may not bring the system down • Resides in own partition		✓	By message patterns, unusual messages, burst of specific message	✓	Diagnosis useful before or after an incident

Extending to Systems Management Products

- Many (ISV) Systems Management products support
  - Actions based on WTO message events
  - Automation of Health Check events
    - PFA Health Check events = soft failures
  - Performance analysis
  - Integration of Alert displays, performance exceptions, event based actions

Integrated z/OS Problem Detection & Prevention



Multi-domain consolidation
Networks
Business applications
Security

Performance analysis
Resource management

Automation

Systems Management Products

Failure Detection
- Performance
- Events
- Take corrective actions

Operating System

Console / Operlog

Runtime Diagnostics

Problem Determination

z/OS 1.13

Soft Failure Avoidance

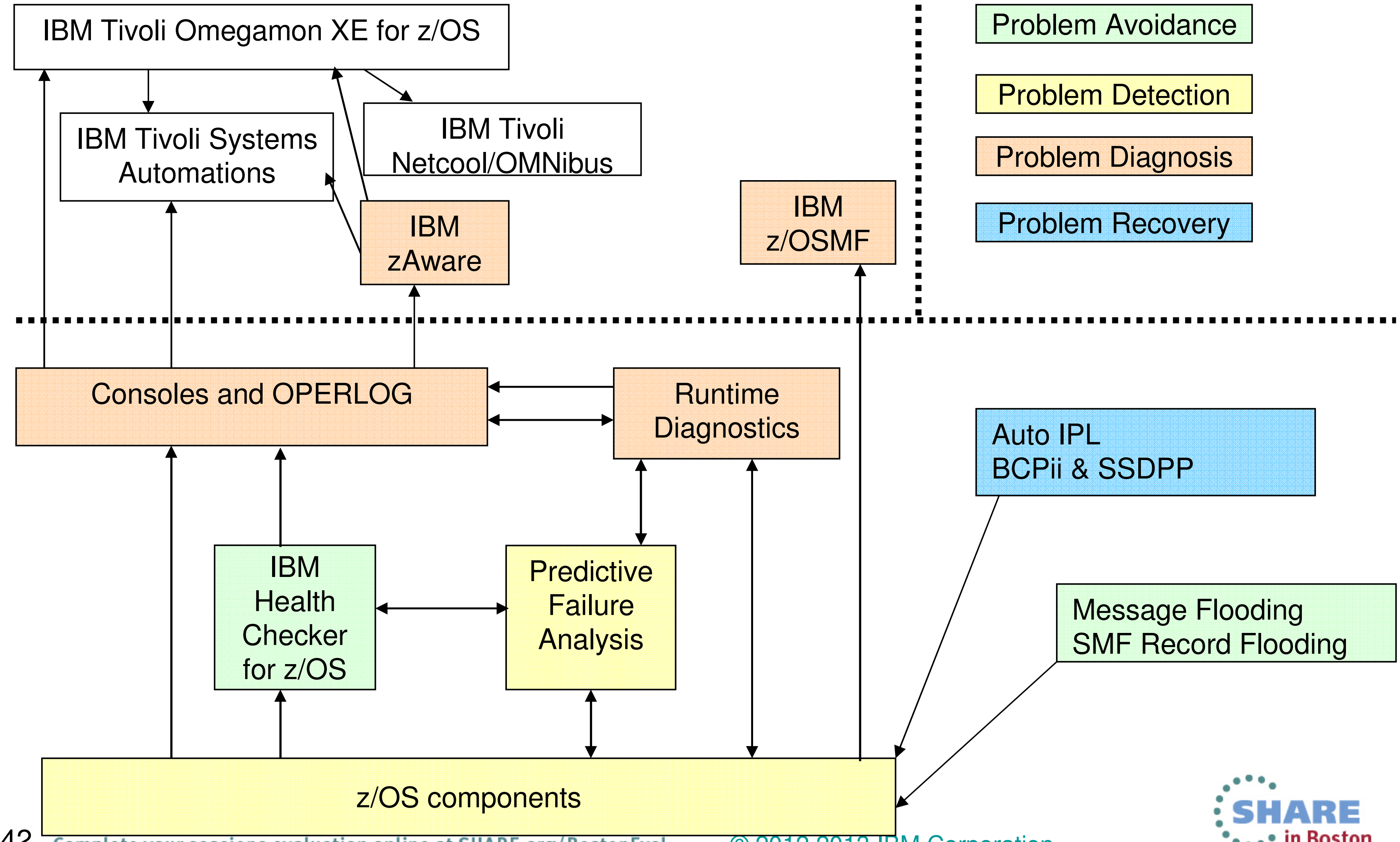
Health Checker

Predictive Failure Analysis (PFA)

Detection

z/OS Components
(System resources & structures … Single system, Sysplex)

Problem Avoidance, Detection, Diagnosis and Recovery



Which one should I use at which time?

Situation…

1. The system is unresponsive.
   - Use IBM zAware’s GUI to see if anomalous messages were issued on the failing system.

2. A Sysplex problem exists where all LPARs are affected.
   - Use IBM zAware’s GUI to see if anomalous messages were issued prior to the problem occurring and to determine on which LPAR the problem originated.

3. Software changes have been made to your system.
   - Use IBM Health Checker for z/OS to verify configuration settings and to detect if migration actions have been performed.
   - Use IBM zAware’s GUI to see if new, unusual message are being issued or if more messages are being issued than expected.

Which one should I use at which time?

Situation…

4. An exception was issued by PFA's message arrival rate check for a rate that is *too high*.
   - Use PFA's report in SDSF to identify address space(s) likely causing exception.
   - Use IBM zAware's GUI to identify messages by ID for the PFA interval and to identify any other anomalous behavior.
   - Use Runtime Diagnostics to further investigate the problem (but, do so quickly!)

5. A PFA exception was issued for any rate that is *too low*.
   - Use PFA's report in SDSF to view the Runtime Diagnostic's events and to view the address space(s) likely causing the exception.
   - Use IBM zAware's GUI to determine if there was a unique message prior to the PFA exception that could identify the source.

Which one should I use at which time?

Situation…

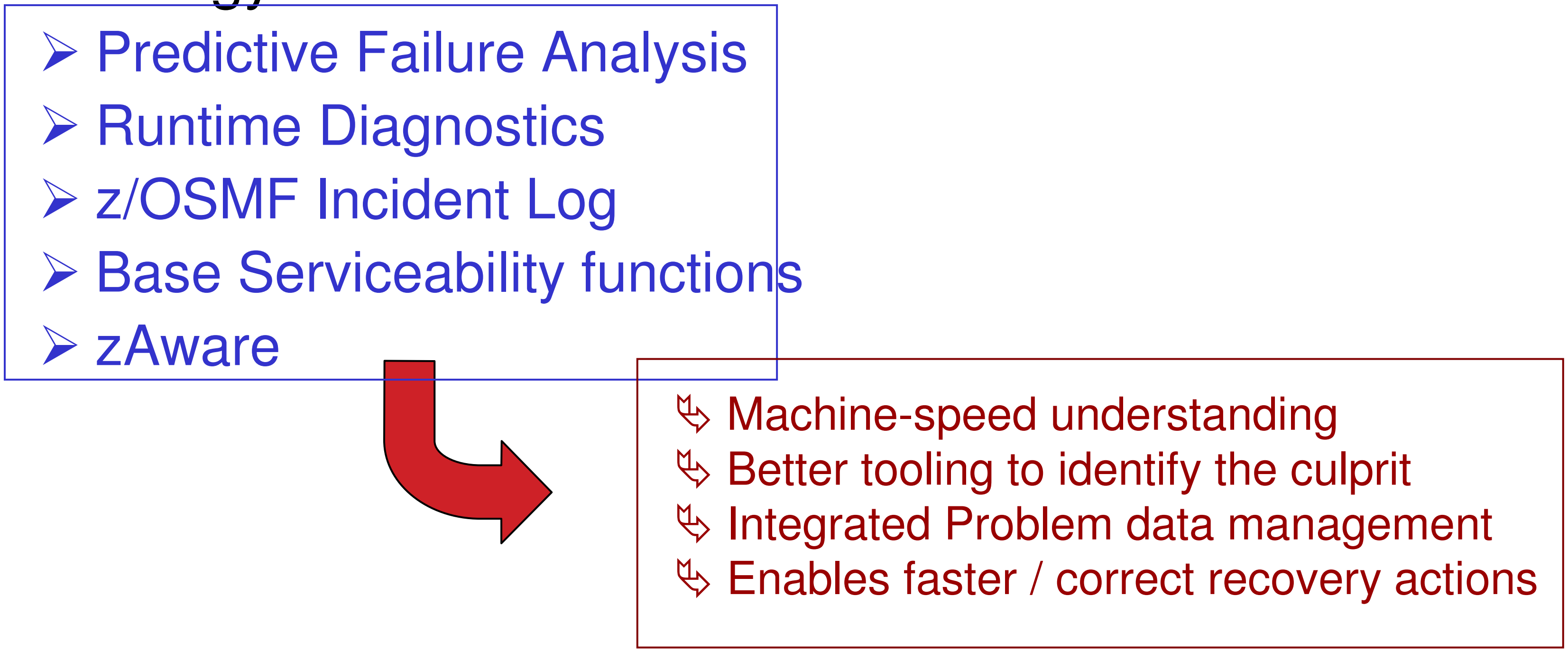
6. IBM zAware detected an anomaly.
   - Investigate the messages with the largest interval contribution score.
   - Use Runtime Diagnostics to further investigate the problem (but, do so quickly!)

7. An exception was issued by PFA's SMF arrival rate or Enqueue request rate checks for a rate that is *too high*.
   - Use PFA's report in SDSF to view the address space(s) likely causing the exception.
   - Use Runtime Diagnostics to see if there are any events (but, do so quickly!)
   - Use IBM zAware's GUI to determine if there is any anomalous message behavior prior to the PFA exception.

Summary

Problem Determination Simplification

Built on z/OS's robust (and continually evolving) RAS technology



- Predictive Failure Analysis
- Runtime Diagnostics
- z/OSMF Incident Log
- Base Serviceability functions
- zAware

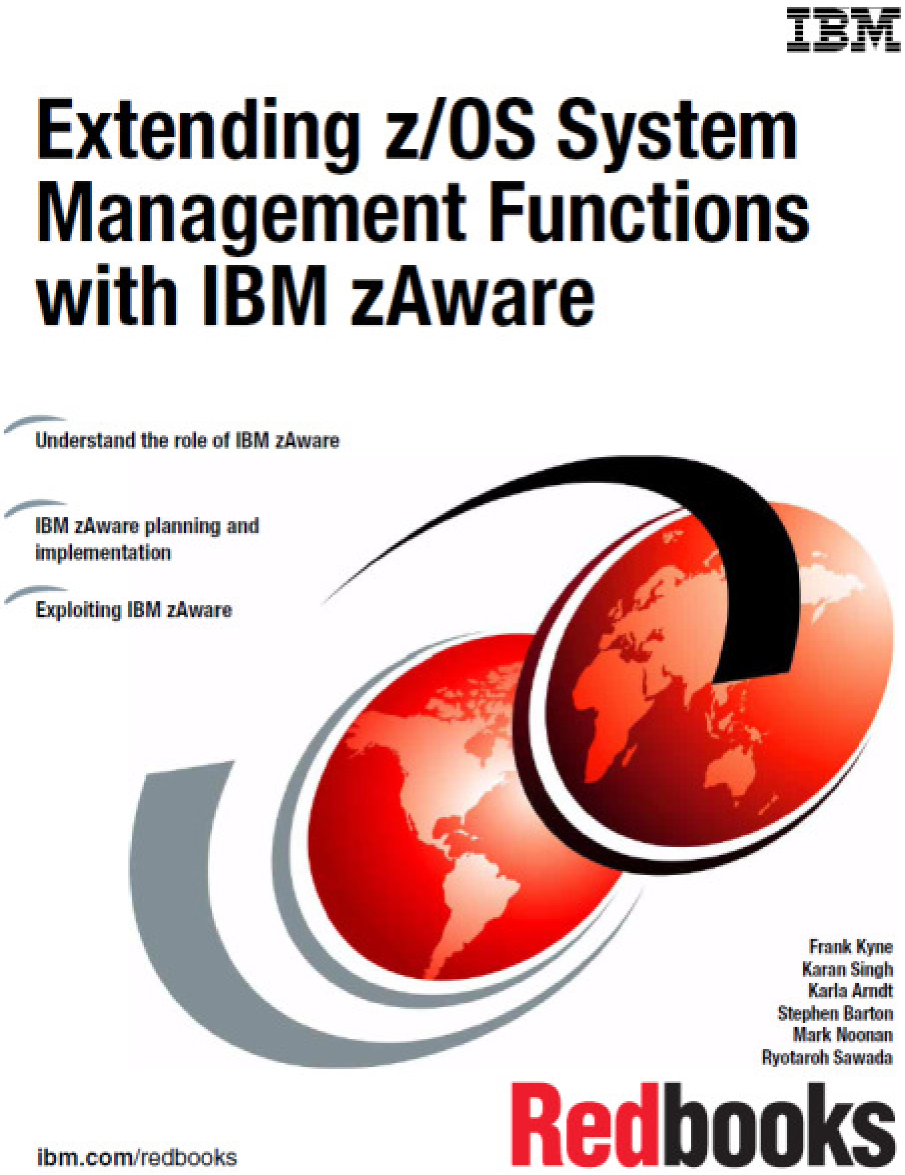
- Machine-speed understanding
- Better tooling to identify the culprit
- Integrated Problem data management
- Enables faster / correct recovery actions

Thank You!

BACKUP

Redbook

SG24-8070



IBM

Extending z/OS System Management Functions with IBM zAware

- Understand the role of IBM zAware
- IBM zAware planning and implementation
- Exploiting IBM zAware

Frank Kyne
Karan Singh
Karla Arndt
Stephen Barton
Mark Noonan
Ryotaroh Sawada

Redbooks

ibm.com/redbooks

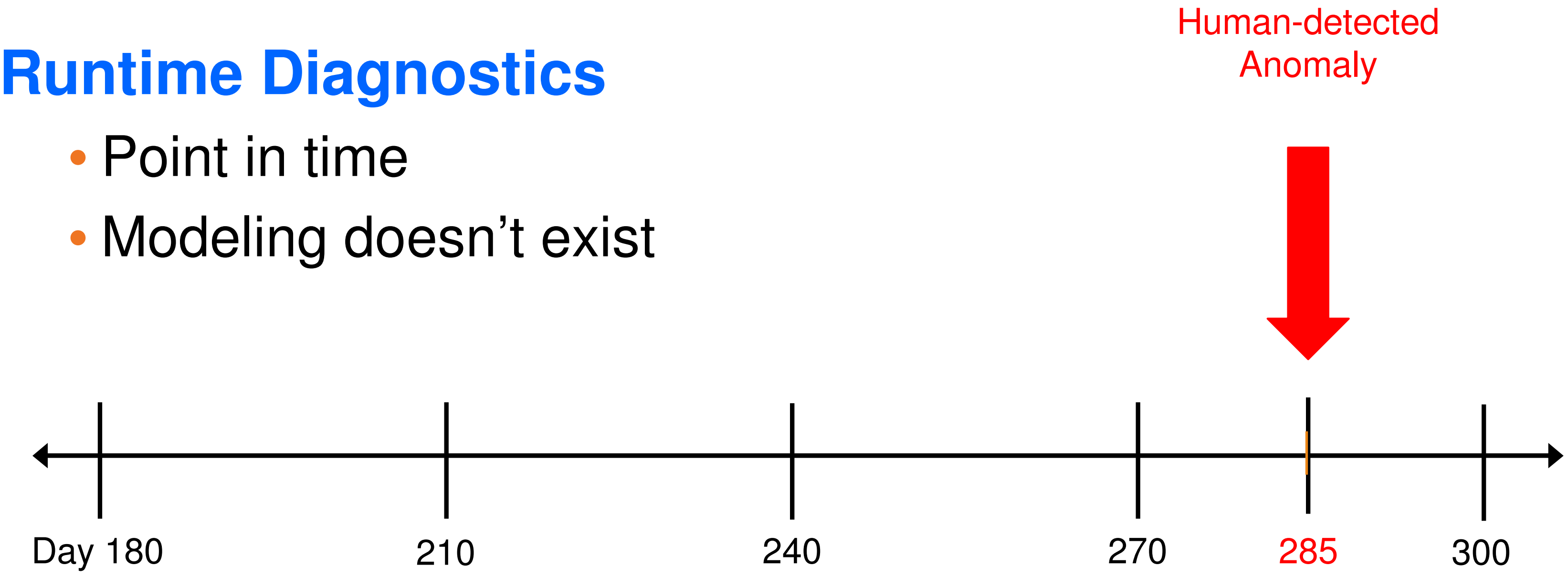
Additional Resources

- One main source of information: *z/OS Problem Management* G325-2564-XX
- PFA IEA presentations
  - http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp?topic=/com.ibm.iea.zos/zos/1.11/Availability/V1R11_PFA/player.html
  - http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp?topic=/com.ibm.iea.zos/zos/1.12/Availability/V1R12_Availability_PFA_Enhancements/player.html
- *z/OS Hot Topics* Newsletters: http://www.ibm.com/systems/z/os/zos/bkserv/hot_topics.html
  - #20 (GA22-7501-16) -- *Fix the Future with Predictive Failure Analysis* by Jim Caffrey, Karla Arndt, and Aspen Payton
  - #23 (GA22-7501-19) – *Predict to prevent: Let PFA change your destiny* by Jim Caffrey, Karla Arndt, and Aspen Payton
  - #23 (GA22-7501-19) – *Runtime to the Rescue! Using Runtime Diagnostics to find out your problems fast* by Bob Abrams, Don Durand, and Dave Zingaretti
- *IBM Systems Magazine - Mainframe Edition*
  - PFA *A Soft Touch* by Karla Arndt, Jim Caffrey, and Aspen Payton
  - http://www.ibmsystemsmagmainframedigital.com/nxtbooks/ibmsystemsmag/mainframe_20101112/index.php#/48

Message Analysis Timeline

Runtime Diagnostics

- Point in time
- Modeling doesn't exist



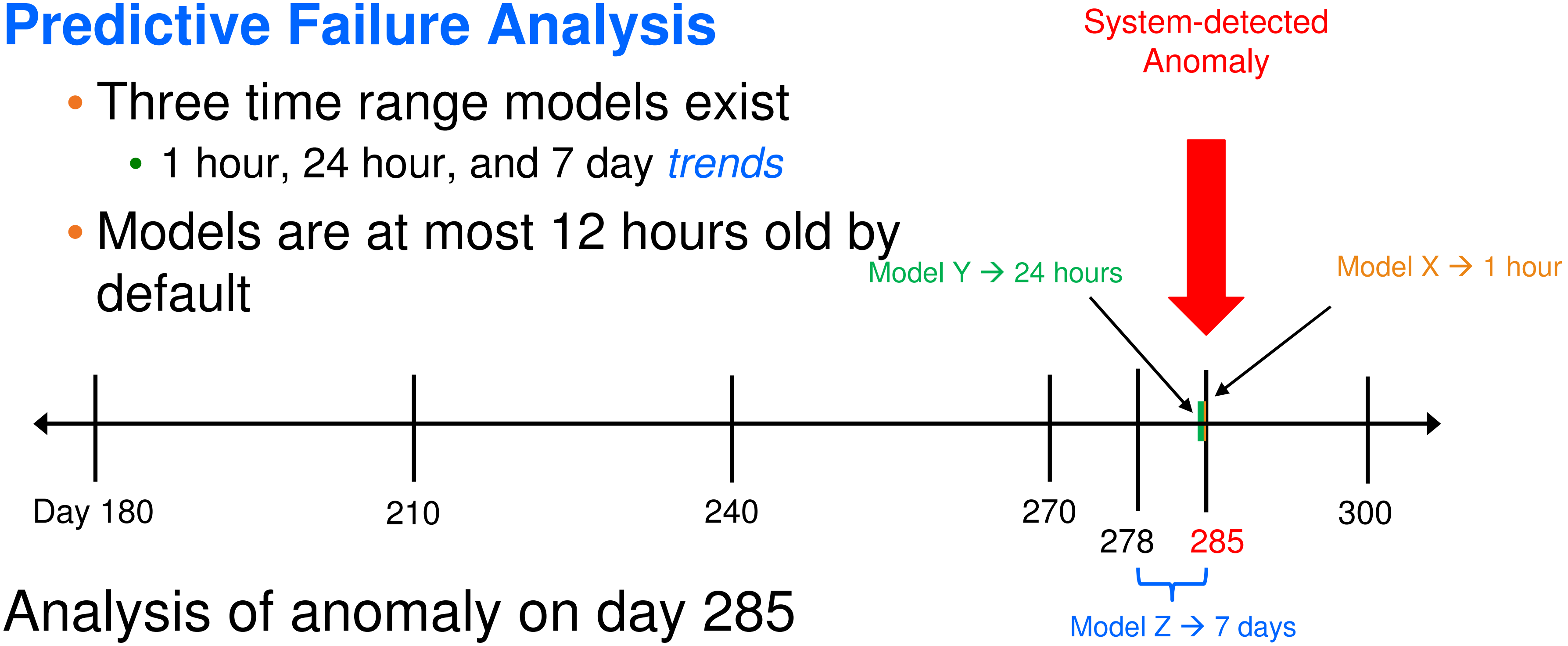
Analysis of anomaly on day 285

- Searches OPERLOG for messages in IBM-defined list
- Looks 1 hour back if available
- If you wait > 1 hour to use Runtime Diagnostics, it will not find the message!

Message Analysis Timeline

Predictive Failure Analysis

- Three time range models exist
  - 1 hour, 24 hour, and 7 day *trends*
- Models are at most 12 hours old by default



System-detected Anomaly

Model Y → 24 hours

Model X → 1 hour

Day 180

210

240

270

278

285

300

Model Z → 7 days

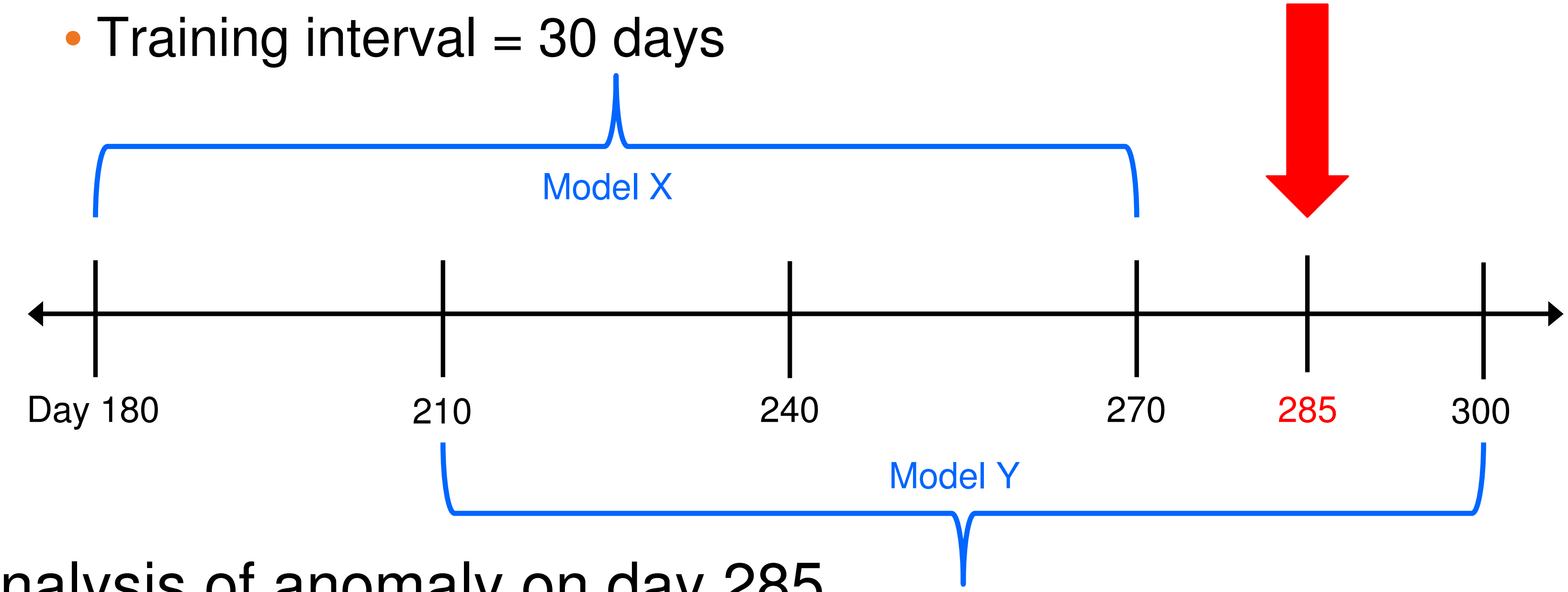
Analysis of anomaly on day 285

- Uses one or more time range models
  - Number depends on “workload change detection logic” in PFA
- Anomaly detected if total *rate* of messages “too high” or “too low”
  - Rate = (WTO + WTORs) / CPU
- New model created immediately if models deemed “too old”
- Can exclude address spaces from PFA’s processing

Message Analysis Timeline

IBM zAware

- Training period = 90 days
- Training interval = 30 days



Analysis of anomaly on day 285

- Uses Model X
- Anomaly detected if message not found in model, message rarely seen in model, message ID too frequent compared to model, or message out of context of known pattern
- New model Y created on day 300 (can drive new model earlier if desired)
  - Day 285 can be excluded by date for future model

Runtime Diagnostics Example Output

Success → Diagnostics run on **Home** system with no events found

```
F HZR,ANALYZE
HZR0201I RUNTIME DIAGNOSTICS SUCCESS. TIME (2012/09/25 - 12:55:19).
NO EVENTS WERE FOUND FOR SYSTEM: #@$A
```

Success → Diagnostics run on **Target** system with no events found

```
F HZR,ANALYZE,SYSNAME=#@$2
HZR0200I RUNTIME DIAGNOSTICS RESULT 830
SUMMARY: SUCCESS - NO EVENTS FOUND
 REQ: 045 TARGET SYSTEM: #@$2     HOME: #@$A     2012/09/25 - 13:07:37
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 00 - PRIORITIES: HIGH:00  MED:00  LOW:00
  PROCESSING BYPASSED:
   OMVS.......SPECIFIED TARGET SYSTEM IS NOT THE HOME SYSTEM.
   LATCHES....SPECIFIED TARGET SYSTEM IS NOT THE HOME SYSTEM.
   LOOP.......SPECIFIED TARGET SYSTEM IS NOT THE HOME SYSTEM.
   HIGHCPU....SPECIFIED TARGET SYSTEM IS NOT THE HOME SYSTEM.
   LOCK.......SPECIFIED TARGET SYSTEM IS NOT THE HOME SYSTEM.
```

Runtime Diagnostics Example Output

Qualified Success → Processing for one or more event types failed

```
f hzr,analyze
HZR0200I RUNTIME DIAGNOSTICS RESULT 751
SUMMARY: QUALIFIED SUCCESS - SOME PROCESSING FAILED
REQ: 001 TARGET SYSTEM: SY1 HOME: SY1 2010/12/21 - 11:25:55
INTERVAL: 60 MINUTES
EVENTS:
FOUND: 01 - PRIORITIES: HIGH:01 MED:00 LOW:00
TYPES: HIGHCPU:01
PROCESSING FAILURES:
OPERLOG....IXGCONN REQ=CONNECT ERROR.......RC=00000008 RS=0000080B
----------------------------------------------------------------------
EVENT 01: HIGH - HIGHCPU - SYSTEM: SY1 2010/12/21 - 11:25:56
ASID CPU RATE:99% ASID:002E JOBNAME:IBMUSERX
STEPNAME:STEP1 PROCSTEP: JOBID:JOB00045 USERID:IBMUSER
JOBSTART:2010/12/21 - 11:22:51
ERROR: ADDRESS SPACE USING EXCESSIVE CPU TIME. IT MIGHT BE LOOPING.
ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
```

Example: Unable to connect to OPERLOG

- OPERLOG may not be set up
- Runtime Diagnostics continues to analyze other types

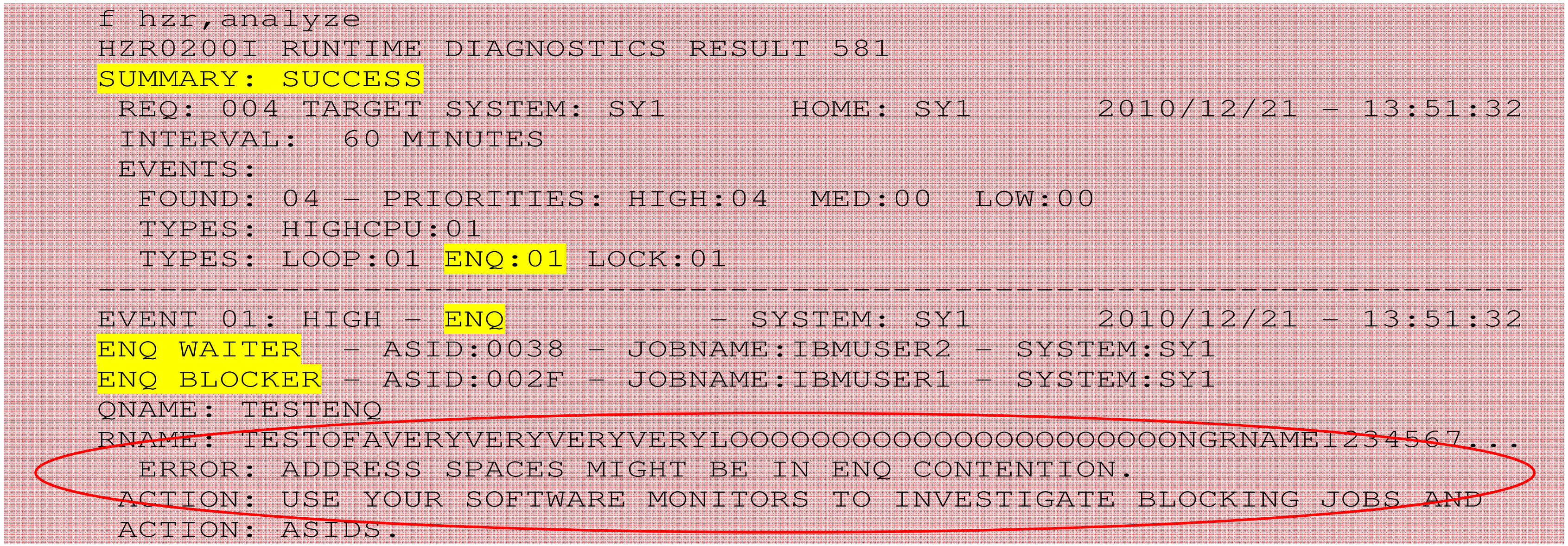
Runtime Diagnostics: Critical Message Analysis

- Component-specific, critical messages in OPERLOG
  - “Needles in a haystack”
  - Looks one hour back, if available
  - For some messages, additional analysis done
    - Groups related messages into a single event
    - Weeds out shortage and relieved critical messages
    - In some cases, will only show last message if a critical message for the same resource name is repeated, say every 10 minutes
  - Message summary found listed in Runtime Diagnostics output

```
EVENT 02: HIGH - CF           - SYSTEM: SY1      2011/02/15 - 14:47:03
IXC585E STRUCTURE LIST01 IN COUPLING FACILITY TESTCFN,
PHYSICAL STRUCTURE VERSION C7565A8D E48F6410,
IS AT OR ABOVE STRUCTURE FULL MONITORING THRESHOLD OF  80%.
ENTRIES:  IN-USE:       491 TOTAL:        583,  84% FULL
ELEMENTS: IN-USE:       508 TOTAL:       1167,  43% FULL
  ERROR: INDICATED STRUCTURE IS APPROACHING FULL MONITORING THRESHOLD.
 ACTION: D XCF,STR,STRNAME=strname TO GET STRUCTURE INFORMATION.
 ACTION: INCREASE STRUCTURE SIZE OR TAKE ACTION AGAINST APPLICATION.
```

Runtime Diagnostics: ENQ Contention Checking

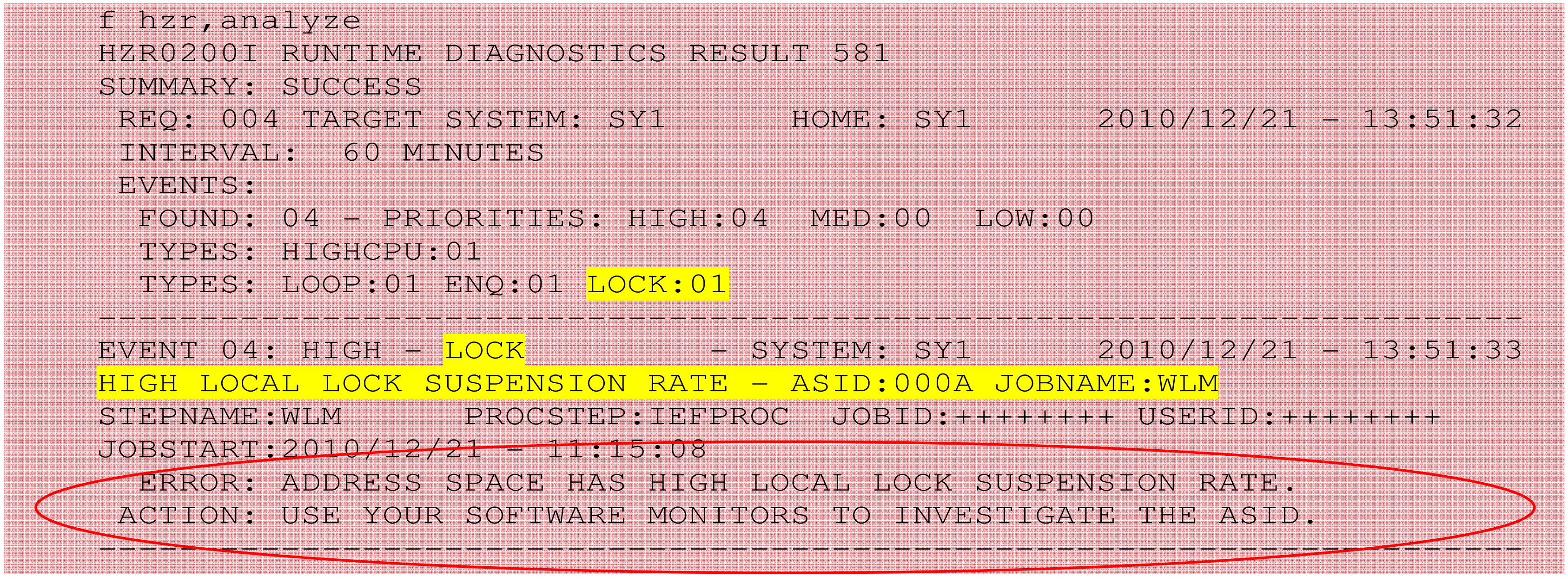
- Looks for a system address space that is an ENQ “waiter” for over 5 seconds
- Lists both waiter and blocker
- Equivalent to D GRS,AN,WAITER



```
f hzr,analyze
HZR0200I RUNTIME DIAGNOSTICS RESULT 581
SUMMARY: SUCCESS
 REQ: 004 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 13:51:32
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 04 - PRIORITIES: HIGH:04  MED:00  LOW:00
  TYPES: HIGHCPU:01
  TYPES: LOOP:01 ENQ:01 LOCK:01
----------------------------------------------------------------------
EVENT 01: HIGH - ENQ          - SYSTEM: SY1      2010/12/21 - 13:51:32
ENQ WAITER  - ASID:0038 - JOBNAME:IBMUSER2 - SYSTEM:SY1
ENQ BLOCKER - ASID:002F - JOBNAME:IBMUSER1 - SYSTEM:SY1
QNAME: TESTENQ
RNAME: TESTOFAVERYVERYVERYVERYLOOOOOOOOOOOOOOOOOOOOOOONGRNAME1234567...
  ERROR: ADDRESS SPACES MIGHT BE IN ENQ CONTENTION.
 ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE BLOCKING JOBS AND
 ACTION: ASIDS.
```

Runtime Diagnostics: Local Lock Suspension

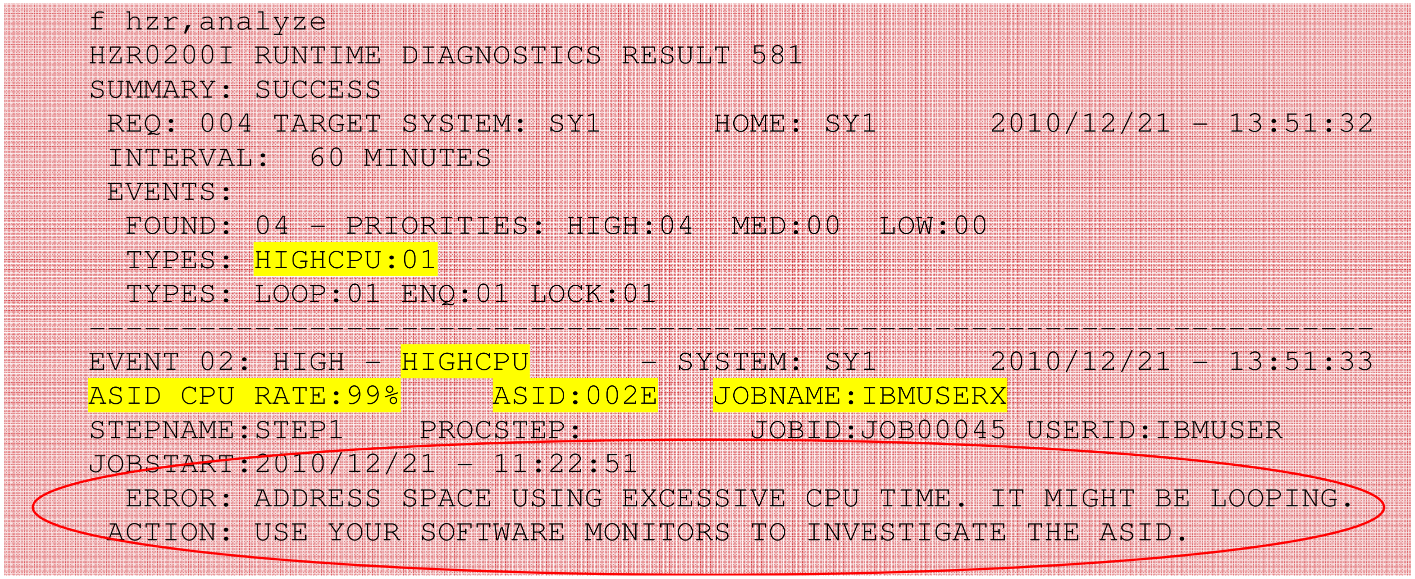
- Lists any address space where its local lock suspension time is over 50%



```
f hzr,analyze
HZR0200I RUNTIME DIAGNOSTICS RESULT 581
SUMMARY: SUCCESS
 REQ: 004 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 13:51:32
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 04 - PRIORITIES: HIGH:04  MED:00  LOW:00
  TYPES: HIGHCPU:01
  TYPES: LOOP:01 ENQ:01 LOCK:01
----------------------------------------------------------------------
EVENT 04: HIGH - LOCK           - SYSTEM: SY1      2010/12/21 - 13:51:33
HIGH LOCAL LOCK SUSPENSION RATE - ASID:000A JOBNAME:WLM
STEPNAME:WLM      PROCSTEP:IEFPROC  JOBID:++++++++ USERID:++++++++
JOBSTART:2010/12/21 - 11:15:08
  ERROR: ADDRESS SPACE HAS HIGH LOCAL LOCK SUSPENSION RATE.
 ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
----------------------------------------------------------------------
```

Runtime Diagnostics: CPU Analysis

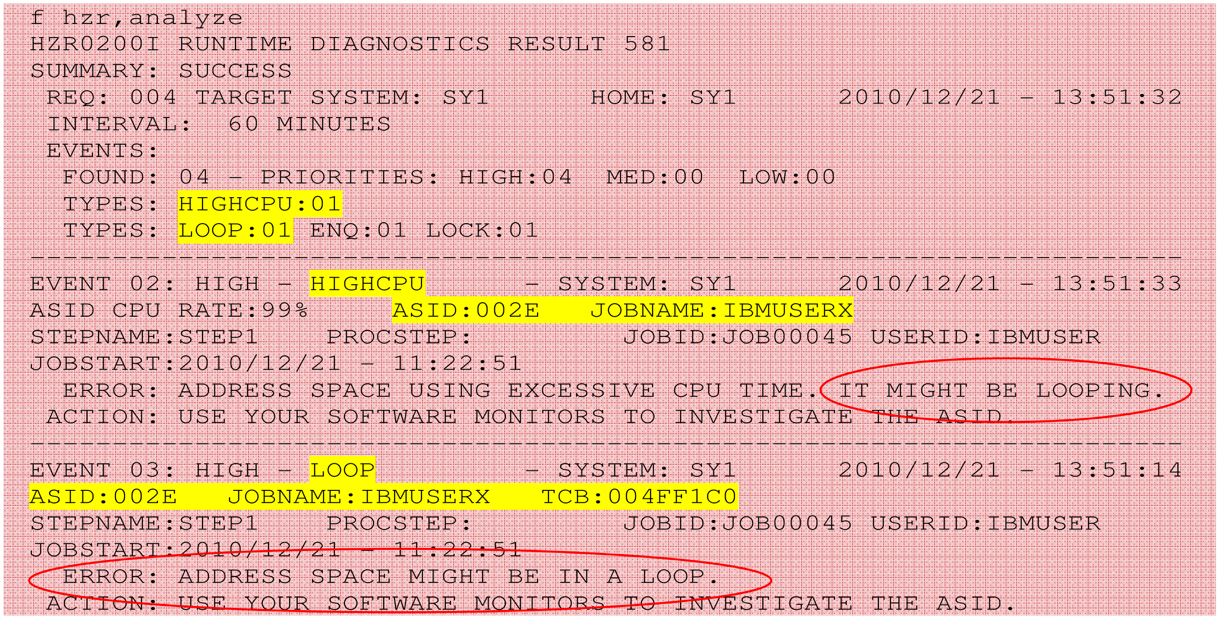
- Takes two quick samples over 1 second interval
- Any task using > 95% of a single CPU is considered a potential problem
- The usage reported might be > 100% if an address space has multiple TCBs and several are using a high percentage of the capacity of a CPU



```
f hzr,analyze
HZR0200I RUNTIME DIAGNOSTICS RESULT 581
SUMMARY: SUCCESS
 REQ: 004 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 13:51:32
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 04 - PRIORITIES: HIGH:04  MED:00  LOW:00
  TYPES: HIGHCPU:01
  TYPES: LOOP:01 ENQ:01 LOCK:01
----------------------------------------------------------------------
EVENT 02: HIGH - HIGHCPU      - SYSTEM: SY1      2010/12/21 - 13:51:33
ASID CPU RATE:99%     ASID:002E   JOBNAME:IBMUSERX
STEPNAME:STEP1    PROCSTEP:         JOBID:JOB00045 USERID:IBMUSER
JOBSTART:2010/12/21 - 11:22:51
  ERROR: ADDRESS SPACE USING EXCESSIVE CPU TIME. IT MIGHT BE LOOPING.
 ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
```

Runtime Diagnostics: Loop Detection

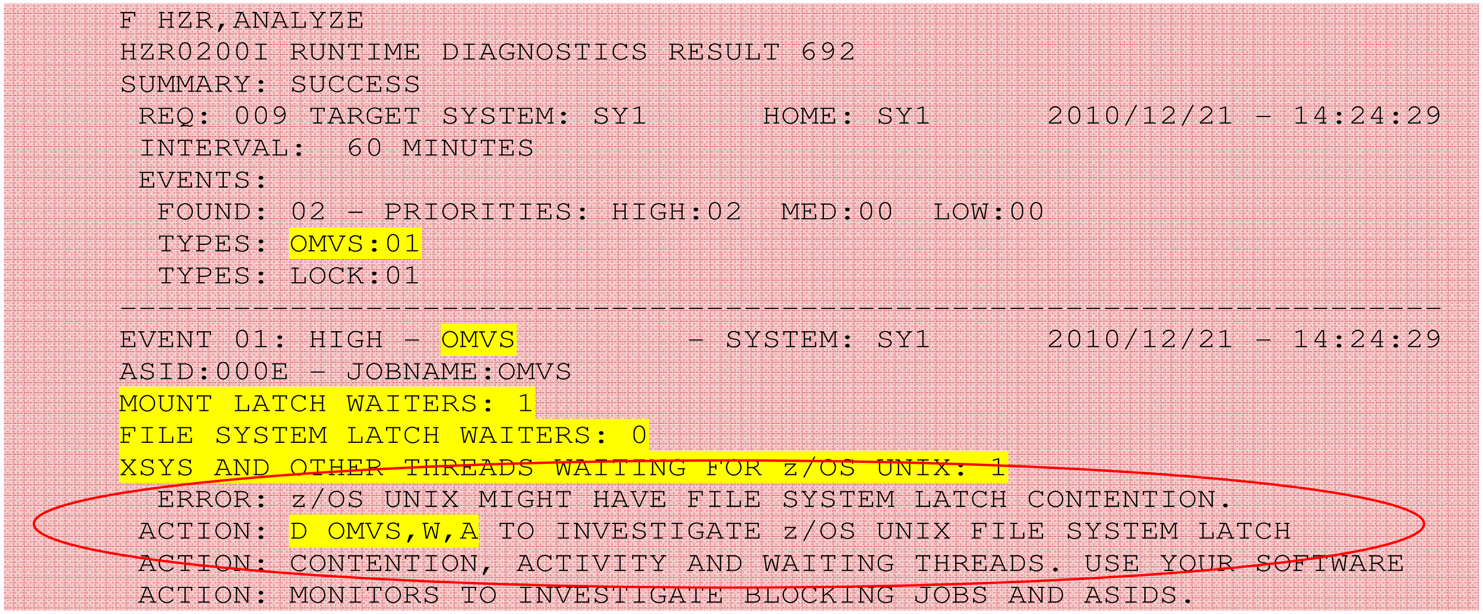
- Investigates all tasks in all address spaces looking for TCB loops
  - Takes a snapshot of the system trace
  - Looks for consistent, repetitive activity that typically indicates a loop
- When both HIGHCPU and LOOP events occur for the same job, there is a high probability that the task in the job is in a loop.
- Normal, corrective action is to cancel the job.



```
f hzr,analyze
HZR0200I RUNTIME DIAGNOSTICS RESULT 581
SUMMARY: SUCCESS
 REQ: 004 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 13:51:32
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 04 - PRIORITIES: HIGH:04  MED:00  LOW:00
  TYPES: HIGHCPU:01
  TYPES: LOOP:01 ENQ:01 LOCK:01
----------------------------------------------------------------------
EVENT 02: HIGH - HIGHCPU      - SYSTEM: SY1      2010/12/21 - 13:51:33
ASID CPU RATE:99%     ASID:002E   JOBNAME:IBMUSERX
STEPNAME:STEP1    PROCSTEP:         JOBID:JOB00045 USERID:IBMUSER
JOBSTART:2010/12/21 - 11:22:51
  ERROR: ADDRESS SPACE USING EXCESSIVE CPU TIME. IT MIGHT BE LOOPING.
 ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
----------------------------------------------------------------------
EVENT 03: HIGH - LOOP         - SYSTEM: SY1      2010/12/21 - 13:51:14
ASID:002E   JOBNAME:IBMUSERX   TCB:004FF1C0
STEPNAME:STEP1    PROCSTEP:         JOBID:JOB00045 USERID:IBMUSER
JOBSTART:2010/12/21 - 11:22:51
  ERROR: ADDRESS SPACE MIGHT BE IN A LOOP.
 ACTION: USE YOUR SOFTWARE MONITORS TO INVESTIGATE THE ASID.
```

Runtime Diagnostics: z/OS UNIX Latch Contention

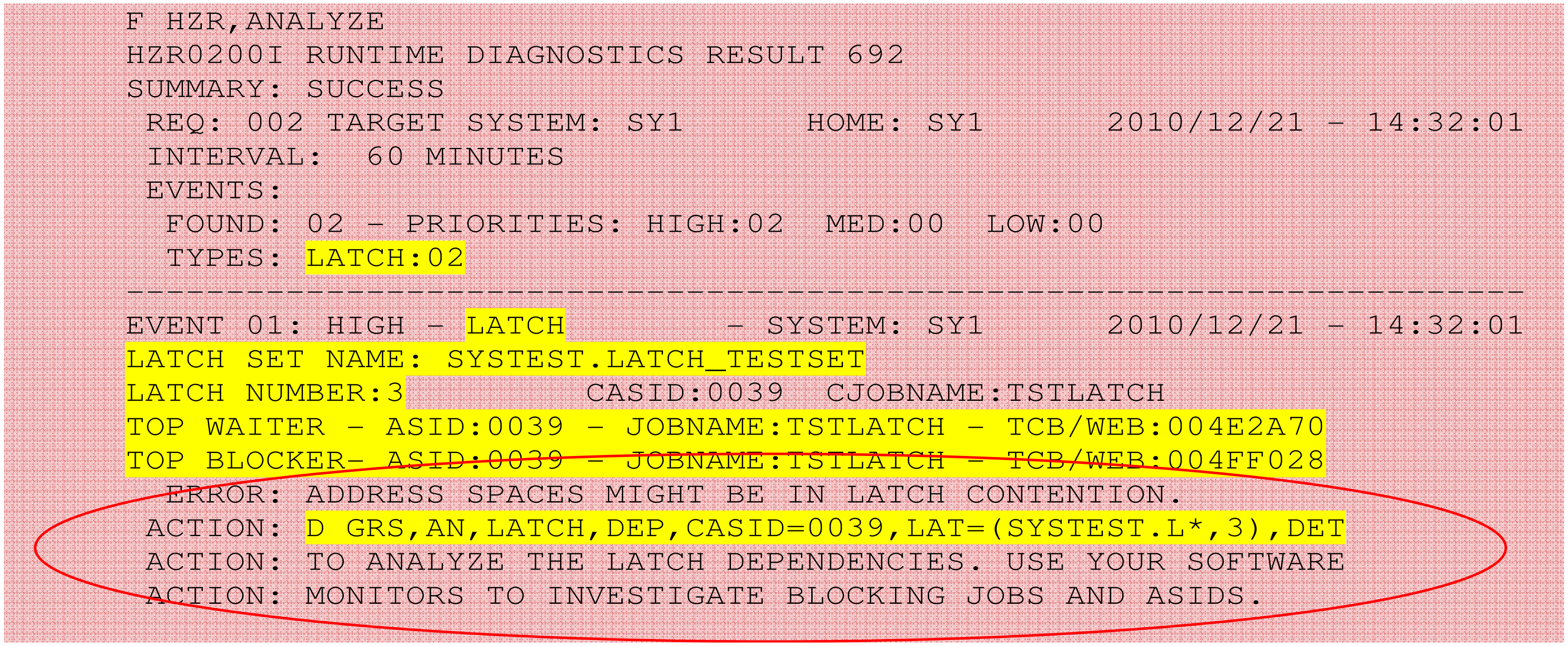
- New in z/OS 1.13
- If z/OS UNIX latch contention or waiting threads exist for > 5 minutes in z/OS UNIX, a Runtime Diagnostics OMVS event is created.
- Normal action is to issue D OMVS,W,A to get the ASID and job names of the waiters



```
F HZR,ANALYZE
HZR0200I RUNTIME DIAGNOSTICS RESULT 692
SUMMARY: SUCCESS
 REQ: 009 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 14:24:29
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 02 - PRIORITIES: HIGH:02  MED:00  LOW:00
  TYPES: OMVS:01
  TYPES: LOCK:01
----------------------------------------------------------------------
EVENT 01: HIGH - OMVS          - SYSTEM: SY1      2010/12/21 - 14:24:29
ASID:000E - JOBNAME:OMVS
MOUNT LATCH WAITERS: 1
FILE SYSTEM LATCH WAITERS: 0
XSYS AND OTHER THREADS WAITING FOR z/OS UNIX: 1
  ERROR: z/OS UNIX MIGHT HAVE FILE SYSTEM LATCH CONTENTION.
 ACTION: D OMVS,W,A TO INVESTIGATE z/OS UNIX FILE SYSTEM LATCH
 ACTION: CONTENTION, ACTIVITY AND WAITING THREADS. USE YOUR SOFTWARE
 ACTION: MONITORS TO INVESTIGATE BLOCKING JOBS AND ASIDS.
```

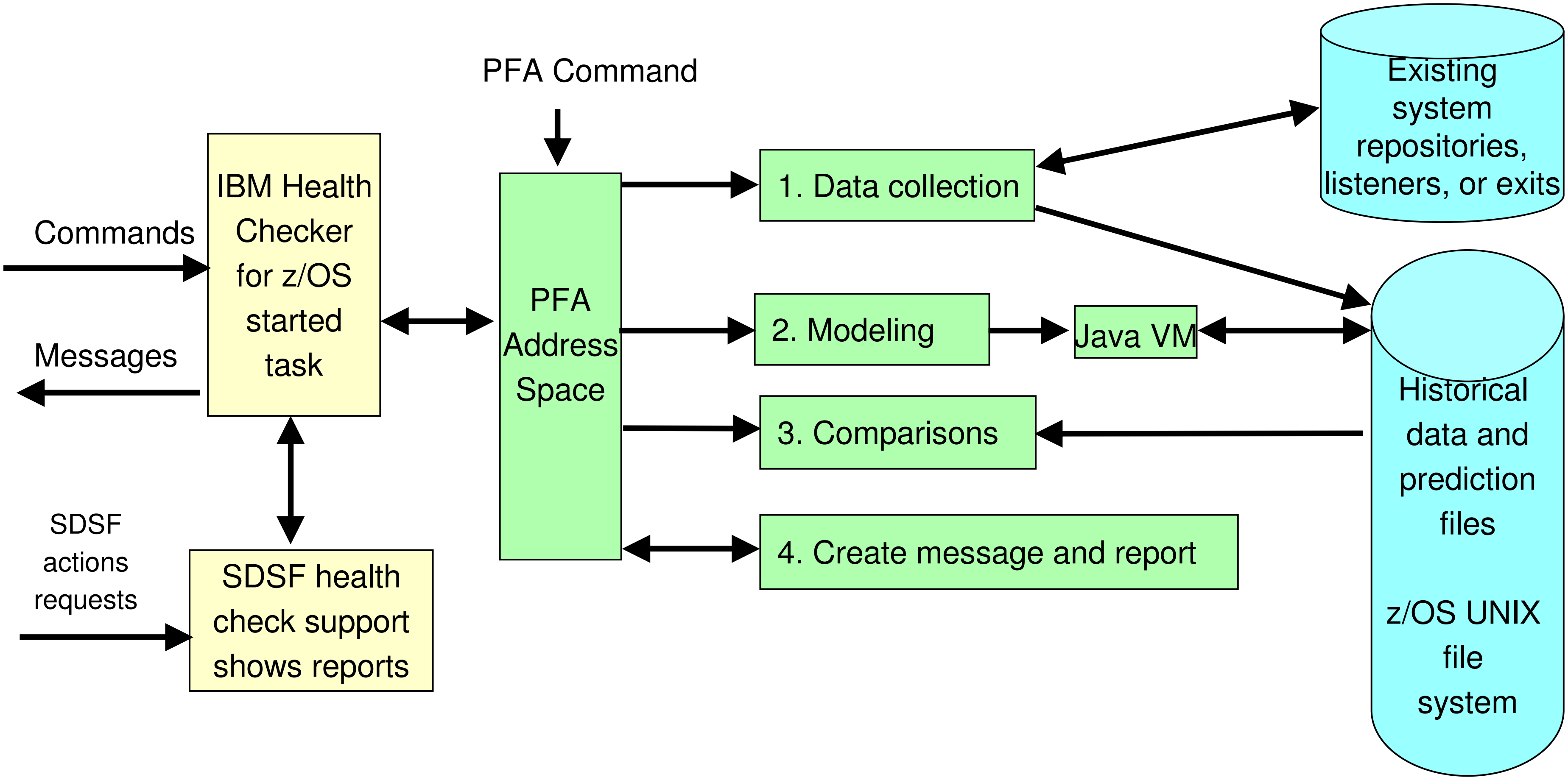
Runtime Diagnostics: GRS file system Latch Contention

- New in z/OS 1.13
- Obtains latch contention information from GRS
- Omits z/OS UNIX file system latch contention
- Returns the longest waiter for each latch set

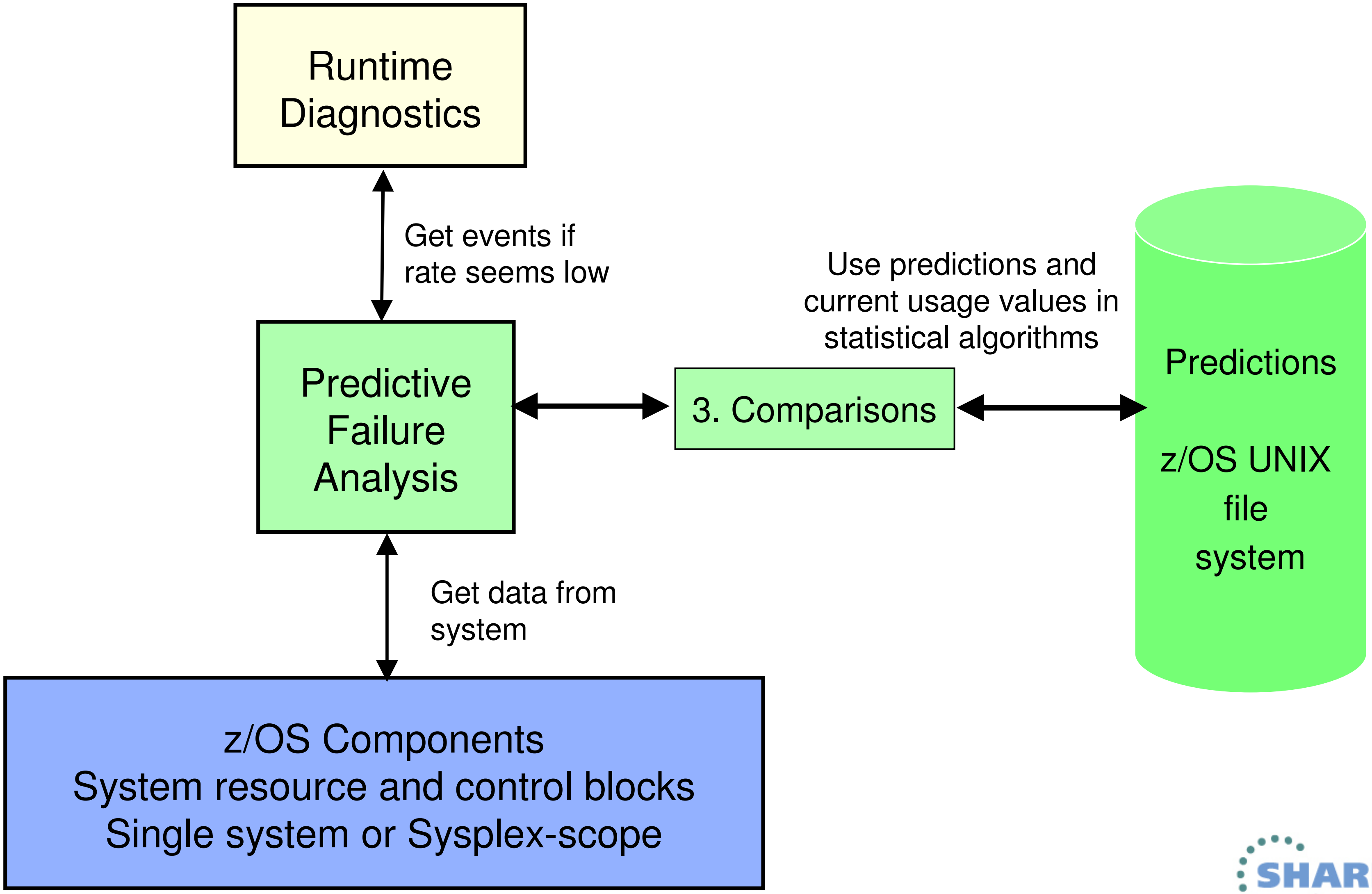


```
F HZR,ANALYZE
HZR0200I RUNTIME DIAGNOSTICS RESULT 692
SUMMARY: SUCCESS
 REQ: 002 TARGET SYSTEM: SY1      HOME: SY1      2010/12/21 - 14:32:01
 INTERVAL:  60 MINUTES
 EVENTS:
  FOUND: 02 - PRIORITIES: HIGH:02  MED:00  LOW:00
  TYPES: LATCH:02
----------------------------------------------------------------------
EVENT 01: HIGH - LATCH        - SYSTEM: SY1      2010/12/21 - 14:32:01
LATCH SET NAME: SYSTEST.LATCH_TESTSET
LATCH NUMBER:3         CASID:0039  CJOBNAME:TSTLATCH
TOP WAITER - ASID:0039 - JOBNAME:TSTLATCH - TCB/WEB:004E2A70
TOP BLOCKER- ASID:0039 - JOBNAME:TSTLATCH - TCB/WEB:004FF028
  ERROR: ADDRESS SPACES MIGHT BE IN LATCH CONTENTION.
 ACTION: D GRS,AN,LATCH,DEP,CASID=0039,LAT=(SYSTEST.L*,3),DET
 ACTION: TO ANALYZE THE LATCH DEPENDENCIES. USE YOUR SOFTWARE
 ACTION: MONITORS TO INVESTIGATE BLOCKING JOBS AND ASIDS.
```

PFA's Implementation


PFA Command
Commands
Messages
IBM Health Checker for z/OS started task
SDSF actions requests
SDSF health check support shows reports
PFA Address Space
1. Data collection
2. Modeling
Java VM
3. Comparisons
4. Create message and report
Existing system repositories, listeners, or exits
Historical data and prediction files
z/OS UNIX file system

PFA and Runtime Diagnostics Integration


Runtime Diagnostics
Get events if rate seems low
Predictive Failure Analysis
3. Comparisons
Use predictions and current usage values in statistical algorithms
Predictions
z/OS UNIX file system
Get data from system
z/OS Components
System resource and control blocks
Single system or Sysplex-scope

How to Get the Most Out of PFA

- Use PFA’s modify command to display parameters
  - Modify command in IBM Health Checker for z/OS does not display cumulative values

- Change the WTOTYPE or SEVERITY of a check if default is not appropriate for your installation
  - Default is SEVERITY(MED) which issues an eventual action WTO
  - `f hzsproc,update,check(ibmpfa,pfa_j*),wtotype=info`

- Quiesce rather than delete PFA checks
  - Optional step (only if you want to stop collections and modeling):
    - `f hzsproc,update,check(ibmpfa,pfa_j*),parm(‘collectinactive(0)’)`
  - `f hzsproc,deactivate,check(ibmpfa,pfa_j*)`

How to Get the Most Out of PFA

- Implement supervised learning to exclude address spaces that cause false exceptions
  - Exclude test programs
  - Exclude address spaces that are inconsistent or spikey in their behavior – “normally abnormal”
- Use check-specific tuning parameters to adjust sensitivity of comparisons if needed
- Use check-specific parameters to affect other behavior

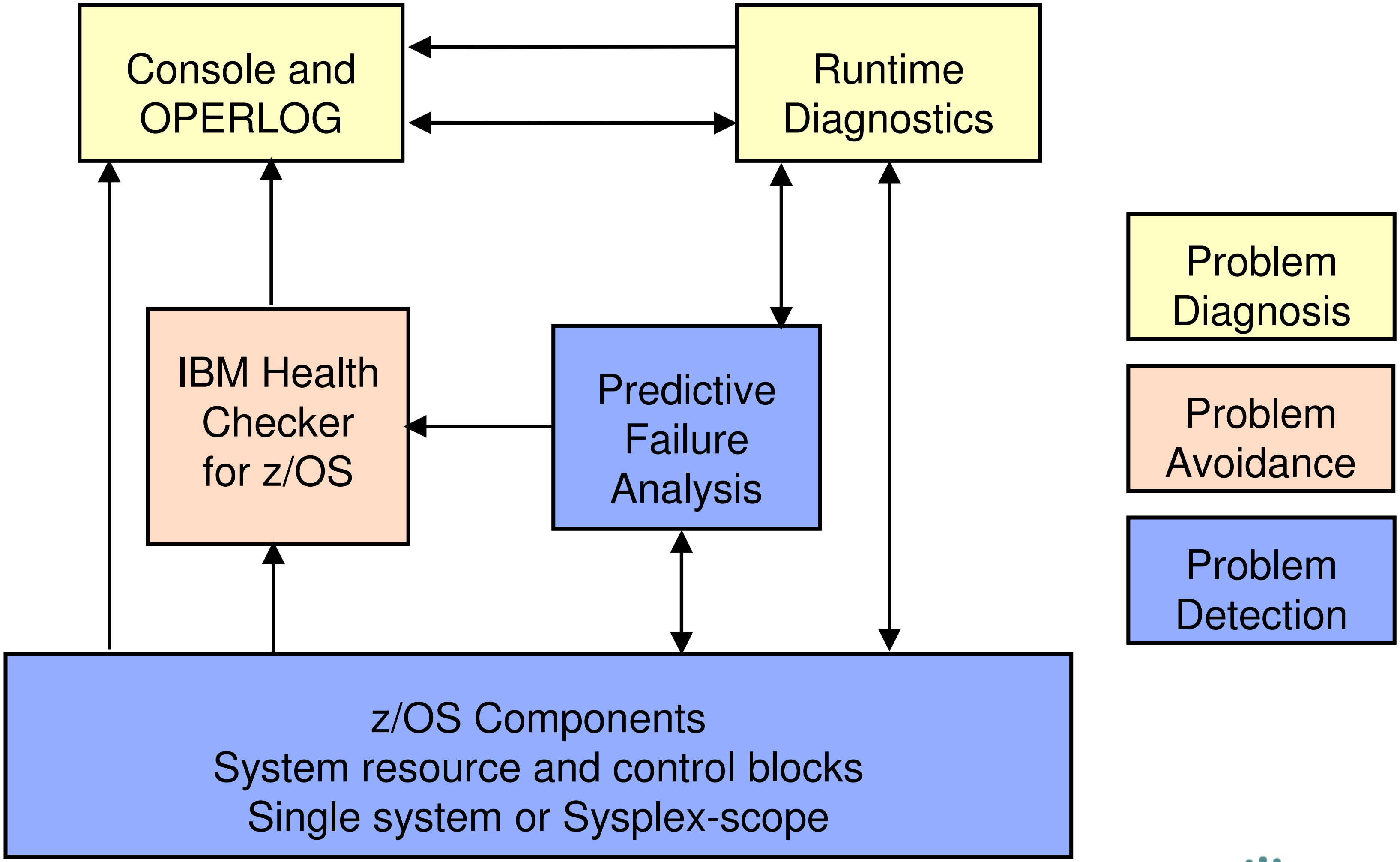
How to Get the Most Out of PFA

- **GET THE LATEST PTFs!!!**

- Joint effort by customers and PFA
  - Many changes over the last year to avoid false positives
  - Removed the PFA_FRAMES_AND_SLOTS_USAGE check
  - Changes to reduce exceptions for “too low” in progress
  - Get ALL PFA PTFs
  - Most fixes targeted for z/OS 1.13 and up

- Help us to make PFA’s results better for everyone!

Early Detection → Outage Avoidance


Console and OPERLOG
Runtime Diagnostics
IBM Health Checker for z/OS
Predictive Failure Analysis
Problem Diagnosis
Problem Avoidance
Problem Detection
z/OS Components
System resource and control blocks
Single system or Sysplex-scope