z/TPF Anomaly Detection Support

Operations and coverage

Josh Wisniewski (on behalf of Mike Shershin!)

2025 TPF Users Group Conference
May 4-7, Austin, TX

IBM Z

IBM

Disclaimer

Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

The Sky is Falling

- The z/TPF system is overloaded for an extended period of time, such as 100% CPU busy or sustained input list shutdown (ILS) due to shortage of system resources.
- Most, if not all, transactions will be negatively affected with increased response times, likely impacting SLAs.
- System wide monitors, such as continuous data collection (CDC), alert you to system health problems like these.

An Isolated Rain Shower on the Radar

- The z/TPF system itself is healthy with plenty of CPU and system resources available.
- A small subset of transactions is experiencing poor response times.
- RTMC can show you differences in behavior to quickly pinpoint the cause. For example:
  - **Booking** messages across the board are doing 30% more I/O than they used to.
  - **Availability** messages from **Mom_and_Pop_Travel** are consuming 50% more CPU than they used to.

A Stealth Rain Shower

- The z/TPF system itself is healthy with plenty of CPU and system resources available.
- A small subset of transactions is experiencing poor response times.
- **RTMC shows no differences in behavior for that message type:**
  - No difference in CPU consumed per message, I/O operations per message, and so on now versus yesterday or a week ago.
- No new version of the application has been deployed recently.
- **What do you do?**

As-Is User Story

- A subset of messages have growing existence times.
- RTMC and other tools are not able to explain why.
- You look at:
  - The record hold table by using ZDRHT and ZRHLD, but locks can be released so quickly you cannot see the cause of the resource contention.
  - File access rates and service times.
  - Manual dump of a booking message.
  - Message analysis tool results, which show the macros with long wait times but without the context of the contention on the system.
  - The recent loadsets activated on the system.
- You guess your way to the solution over the course of several hours, potentially days.

To-Be User Story

- A subset of messages have growing existence times.
- Your **application performance monitoring (APM) tool** provides samples of these messages with large existence times including details for the file, DASD record lock, core hold, and so on with the largest anomaly times for that message.
- Traditional RTMC data does not show differences in message profile behavior (other than response time).
- New RTMC **anomaly detection** dashboards provide insights on events that historically are the most likely causes of increased response time for transactions, such as excessive lock contention (like DASD record locking and core holds) and slow DASD I/O.
- You can pinpoint the cause in a few minutes.

Anomaly Detection Usage Story

- Sarah, the site reliability engineer (SRE), is alerted by the enterprise APM tool that **booking** message response time is larger than normal on z/TPF.

EventId:
PDsh8nkDTuef2KgbmhA3Eg

Link:
https://instana.fake.com/#/events:eventID=PDsh8nkDTuef2KgbmhA3Eg&incidenteTo=16345989153

Incident started with:
Booking messages on z/TPF existence time exceeded “normal” threshold by three standard deviations.
These violations are occurring continuously for booking messages.
Booking SLAs may be violated within the next 5 minutes.

Detail:
Booking messages at edge server are completing in 31ms on average exceeding the 25ms SLA threshold for the past 10 seconds.
Existence time for booking messages on all servers is within “normal” threshold by 2 standard deviations except z/TPF.
Sampled booking messages from z/TPF include anomalies like “DASD record lock anomaly time of 21ms for the Inventory database”.

Metric name: latency (normal 3 stddev threshold)

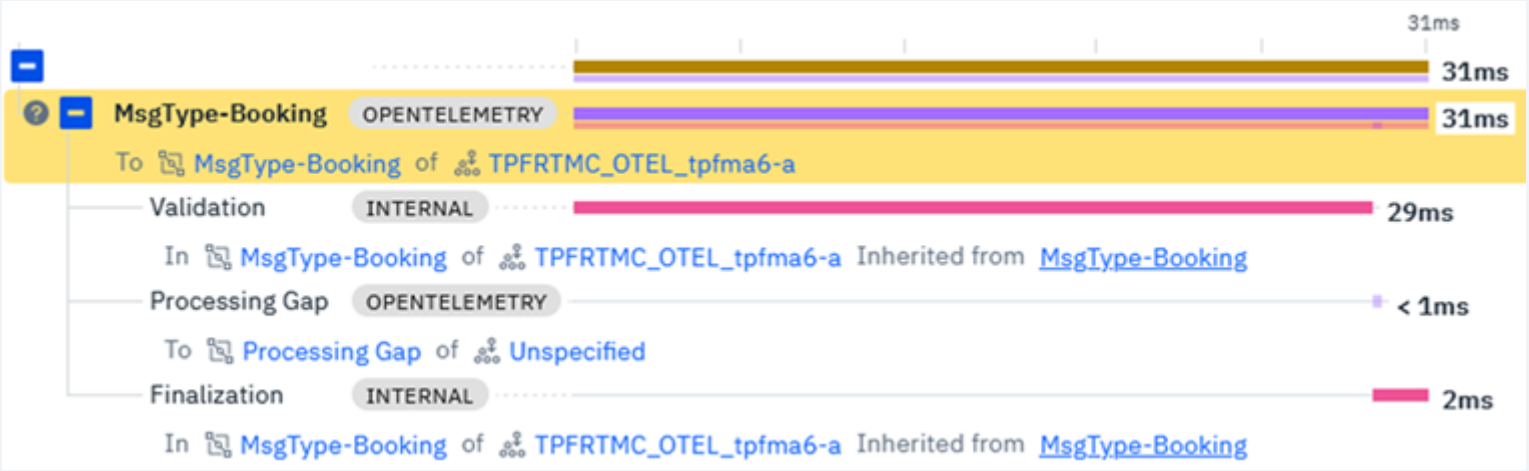
Severity: Critical

Sarah
site reliability engineer

*If you don’t have an APM tool, you could build an RTMC based alert to monitor existence time or deviations of detected anomalies.

Anomaly Detection Usage Story

- Sarah inspects a few **booking** messages in the APM tool and sees CPU consumed per message (cpu_used) is normal, but response time (exist_time) is larger than normal and violating SLAs.



Sarah
site reliability engineer

Anomaly Detection Usage Story

- Some **booking** messages have anomalies reported as the APM tool alert indicated: “DASD record lock anomaly time of 25ms for the Inventory database”.

EXIST_TIME	30978
FIND_DASD	0
MESSAGE_LIFETIME	30996
TPF_OTEL_rc	IValueUnassigned
CPU_USED	833
MsgType	Booking
TPF_OTEL_rc_msg	IValueUnassigned
FILE_DASD	0
Anomaly Detected	DASD record lock

Sarah
site reliability engineer

Anomaly Detection Usage Story

- Sarah needs help from a z/TPF expert.
- She contacts Carol the z/TPF coverage programmer.

Sarah
site reliability engineer

Carol
z/TPF coverage programmer

Anomaly Detection Usage Story

- Carol was also notified by the APM tool (or by RTMC) and is already investigating the issue.
- Looking at the RTMC name-value pair metrics dashboard, she has identified a point of inflection when the existence time suddenly rose for **booking** messages.

Carol
z/TPF
coverage
programmer

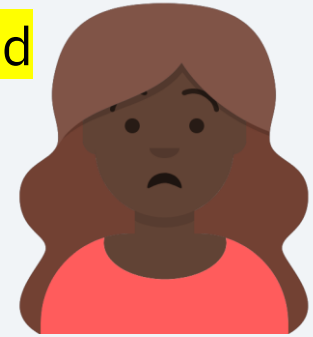
Anomaly Detection Usage Story

- Carol looks at the logs in that time frame and sees a loadset was recently activated in production for a new **availability** feature.
- Existence time also rose for **availability** messages and they're doing more I/O, but that was expected.
- That loadset did **not** change **booking** processing!

Carol
z/TPF
coverage
programmer

Anomaly Detection Usage Story

- Carol digs into other **booking** name-value pair metrics:
  - CPU consumed (cpu_used) is unchanged
  - Database activity (number of I/O operations) is unchanged
  - z/TPFDF database metrics are unchanged
  - Copy-on-write rate is unchanged
  - Even the message rate (est_msg_rate) is unchanged



Carol
z/TPF coverage programmer

Anomaly Detection Usage Story

- Carol looks at metrics for name-value pair combinations:
  - **Booking** message and code package are unchanged
  - **Booking** message and return code are unchanged
    - **Booking** error rates are unchanged
  - **Booking** message and ECB purpose are unchanged

Carol
z/TPF
coverage
programmer

Anomaly Detection Usage Story

- Carol opens the name-value pair correlation analysis dashboard to see if it can provide any insights by channel or other name-value pair combinations.
- It shows that all **booking** messages are affected regardless of name-value pair value combinations.

Message Type, SubType, Origin Rate Correlated to Inuse System ECBs

Message Type, SubType, Origin	Coef-Lt	Coef-Rt	MatDelta	Insuf-Lt	Insuf-Rt	%Trfc-Lt
[Availability, Hotel, Mobile]	-0.13			-	Variation	35.6
[Shopping, Hotel, Web]	-0.03			-	Variation	3.6
[Shopping, Air, Web]	0.12			-	Variation	19.8
[Booking, Hotel, Mobile]	-0.04			-	Variation	15.4
[Shopping, Hotel, Mobile]	0.12			-	Variation	16.0
[IValueUnassigned, IValueUnassig	0.10			-	Variation	4.6

Carol
z/TPF coverage programmer

Anomaly Detection Usage Story

- Everything about booking messages is the same but the existence time is larger. WHAT'S GOING ON?!?!?!
- Carol does not want to deactivate the **availability** loadset without careful consideration because the new features have executive scrutiny.

Carol
z/TPF
coverage
programmer

Anomaly Detection Usage Story

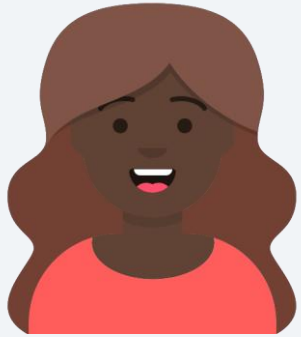
- Carol uses the new **anomaly detection** feature and opens the new RTMC dashboard.
- At a glance, she can see what the problem is!

Summary by Anomaly Type

Anomaly Type	Name	File ID	Anomaly Count	Average Anomaly Time	MAX Anomaly Time	Average Wait Queue	MAX Wait Queue
DASD Record Lock-Summary			523	30ms	100ms	10	50
DASD IO-Summary			2	7ms	30ms	0	3
Core Hold-Summary			1	10ms	10ms	1	7
DASD Record Lock	Inventory	FCD0	523	30ms	100ms	10	50
DASD IO	FFlyer	FEA0	1	2ms	12ms	0	2
DASD IO	PNR	FDD4	1	10ms	30ms	0	3
Core Hold	MyCoreHold		1	10ms	10ms	1	7

Filter Type Summary

Filter Values	Anomaly Type	Name	File ID	Count	Average Anomaly Time	MAX Anomaly Time	Average Wait Queue	MAX Wait Queue
Availability	DASD Record Lock-Summary			475	29ms	89ms	10	48
Booking	DASD Record Lock-Summary			48	33ms	100ms	9	50
Availability	DASD Record Lock	Inventory	FCD0	475	29ms	89ms	10	48
Booking	DASD Record Lock	Inventory	FCD0	48	33ms	100ms	9	50



Carol
z/TPF coverage programmer

*Screen shots are conceptual mockups of what the dashboard might look like. They will be refined with sponsor user feedback.

Anomaly Detection Usage Story

- The summary table shows lots of **DASD record lock anomalies** being reported and almost all are for the **inventory database**.
- The anomaly time is the time spent waiting to get the lock.

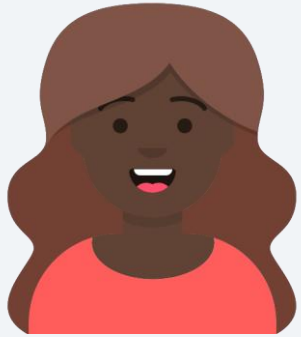
Summary by Anomaly Type

Anomaly Type	Name	File ID	Anomaly Count	Average Anomaly Time	MAX Anomaly Time	Average Wait Queue	MAX Wait Queue
DASD Record Lock-Summary			523	30ms	100ms	10	50
DASD IO-Summary			2	7ms	30ms	0	3
Core Hold-Summary			1	10ms	10ms	1	7
DASD Record Lock	Inventory	FCD0	523	30ms	100ms	10	50
DASD IO	FFlyer	FEA0	1	2ms	12ms	0	2
DASD IO	PNR	FDD4	1	10ms	30ms	0	3
Core Hold	MyCoreHold		1	10ms	10ms	1	7

Carol
z/TPF coverage programmer

Anomaly Detection Usage Story

- Carol filters based on anomaly type “DASD Record Lock”
- The summary shows **booking** AND **availability** messages are locking the **inventory database** and experiencing high levels of lock contention
- Carol knows that **availability** messages **are not supposed to lock** the **inventory database or any database**!



Carol
z/TPF coverage programmer

Filter Type Summary

Filter Values	Anomaly Type	Name	File ID	Count	Average Anomaly Time	MAX Anomaly Time	Average Wait Queue	MAX Wait Queue
Availability	DASD Record Lock-Summary			475	29ms	89ms	10	48
Booking	DASD Record Lock-Summary			48	33ms	100ms	9	50
Availability	DASD Record Lock	Inventory	FCD0	475	29ms	89ms	10	48
Booking	DASD Record Lock	Inventory	FCD0	48	33ms	100ms	9	50

Anomaly Detection Usage Story

- The **anomaly details table** shows information for each anomaly detected.
- Lock contention anomalies for the **inventory DB** for **availability** messages are coming from program WXYZ in loadset PROJECT1, which was activated 5 minutes ago!

Anomaly Details

Target	Anomaly Type	Name	File ID	Anomaly Time	Wait Queue	exist_time	cpu_used	NVP Value	Loadset	Program	Object	Macro	Offset	File Address	ECB SVM	ECB Time
■	DASD Record Lock	Inventory	FCD0	100ms	50	321ms	10ms	Booking	BASE	ABCD	abcd.o	FIWHC	0xf00	0x12345678	0x104000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	89ms	42	555ms	120ms	Availability	PROJECT1	WXYZ	wxyz.o	FIWHC	0xab2	0x87654320	0x10c000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	80ms	47	303ms	10ms	Booking	BASE	ABCD	abcd.o	FIWHC	0xf00	0x55909090	0x100000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	79ms	48	287ms	10ms	Booking	BASE	ABCD	abcd.o	FIWHC	0xf00	0x70454360	0x10e000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	79ms	35	291ms	10ms	Booking	BASE	ABCD	abcd.o	FIWHC	0xf00	0x12345678	0x114000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	70ms	40	501ms	121ms	Availability	PROJECT1	WXYZ	wxyz.o	FIWHC	0xab2	0x87654320	0x124000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	70ms	46	490ms	123ms	Availability	PROJECT1	WXYZ	wxyz.o	FIWHC	0xab2	0x87654320	0x12C000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	70ms	25	481ms	119ms	Availability	PROJECT1	WXYZ	wxyz.o	FIWHC	0xab2	0x12345678	0x110000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	68ms	41	497ms	122ms	Availability	PROJECT1	WXYZ	wxyz.o	FIWHC	0xab2	0x55909090	0x118000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	68ms	44	486ms	120ms	Availability	PROJECT1	WXYZ	wxyz.o	FIWHC	0xab2	0x70454360	0x134000	2/25/2025 12:54

Anomaly Detection Usage Story

- Based upon the **file address**, Carol can see that **excessive lock contention is occurring on different subfiles** in the **inventory** database.
- But, **booking** and **availability** messages are locking the **same subfiles**.

Anomaly Details

Target	Anomaly Type	Name	File ID	Anomaly Time	Wait Queue	exist_time	cpu_used	NVP Value	Loadset	Program	Object	Macro	Offset	File Address	ECB SVM	ECB Time
■	DASD Record Lock	Inventory	FCD0	100ms	50	321ms	10ms	Booking	BASE	ABCD	abcd.o	FIWHC	0xf00	12345678	0x104000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	89ms	42	555ms	120ms	Availability	PROJECT1	WXYZ	wxyz.o	FIWHC	0xab2	87654320	0x10c000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	80ms	47	303ms	10ms	Booking	BASE	ABCD	abcd.o	FIWHC	0xf00	55909090	0x100000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	79ms	48	287ms	10ms	Booking	BASE	ABCD	abcd.o	FIWHC	0xf00	70454360	0x10e000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	79ms	35	291ms	10ms	Booking	BASE	ABCD	abcd.o	FIWHC	0xf00	12345678	0x114000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	70ms	40	501ms	121ms	Availability	PROJECT1	WXYZ	wxyz.o	FIWHC	0xab2	87654320	0x124000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	70ms	46	490ms	123ms	Availability	PROJECT1	WXYZ	wxyz.o	FIWHC	0xab2	87654320	0x12C000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	70ms	25	481ms	119ms	Availability	PROJECT1	WXYZ	wxyz.o	FIWHC	0xab2	12345678	0x110000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	68ms	41	497ms	122ms	Availability	PROJECT1	WXYZ	wxyz.o	FIWHC	0xab2	55909090	0x118000	2/25/2025 12:54
☐	DASD Record Lock	Inventory	FCD0	68ms	44	486ms	120ms	Availability	PROJECT1	WXYZ	wxyz.o	FIWHC	0xab2	70454360	0x134000	2/25/2025 12:54

Anomaly Detection Usage Story

- Carol compares the current anomalies to the anomalies recorded an hour ago, which is before the loadset was activated.
- Before the loadset was activated, there was only a trivial number of anomalies reported for DASD lock contention anomalies on the **inventory database**.

Summary by Anomaly Type (previous 2/24/2025 12:00-2/24/2025 12:05 current 2/24/2025 13:00-2/24/2025 13:05)

Anomaly Type	Name	File ID	Previous Anomaly Count	Current Anomaly Count	Delta Count	Previous Average Anomaly Time	Current Average Anomaly Time	Delta Average Anomaly Time	Previous MAX Anomaly Time	Current MAX Anomaly Time	Delta MAX Anomaly Time	Previous Average Wait Queue	Curr Aver Wai Que
DASD Record Lock-Summary			1	523	522	2ms	30ms	28ms	2ms	100ms	98ms	2	10
DASD IO-Summary			1	2	1	3ms	7ms	4ms	3ms	30ms	27ms	1	0
Core Hold-Summary			2	1	-1	4ms	10ms	6ms	4ms	10ms	6ms	1	1
DASD Record Lock	Inventory	FCD0	1	523	522	2ms	30ms	28ms	2ms	100ms	98ms	2	10
DASD IO	FFlyer	FEA0	0	1	1		2ms			12ms			0
DASD IO	PNR	FDD4	1	1	0	3ms	10ms	7ms	3ms	30ms	27ms	1	0
Core Hold	MyCoreHold		2	1	-1	4ms	10ms	6ms	4ms	10ms	6ms	1	1

Anomaly Detection Usage Story

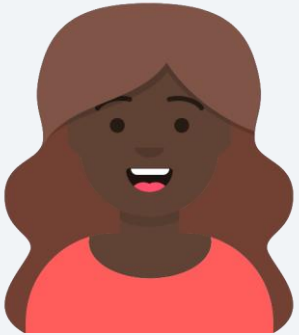
- Carol has enough evidence to conclude loadset PROJECT1 is the cause of the problem impacting **booking** messages and deactivates that loadset.
- **Booking** message existence time returns to normal.
- Carol checks the anomalies detection dashboards to confirm that DASD record lock anomalies are back to normal for the **booking** messages.

Carol
z/TPF
coverage
programmer

Anomaly Detection Usage Story

- Anomaly detection was the key to determining the source of the problem in a timely manner and minimizing impacts to SLAs.
- Sarah and Carol discuss how this was resolved in minutes when these types of problems used to take hours to figure out.

Sarah
site reliability engineer



Carol
z/TPF coverage programmer

z/TPF Anomaly Detection Summary

- Reactive use case:
  - Quickly identify what is impacting transaction response time for a subset of messages in production when the application itself is not exhibiting any other changes in behavior.
- Proactive use case:
  - Use during development process to ensure changes are not introducing contention, especially for tests and workload at scale.
  - Identify current minor impacts (contention) that would likely become a problem if the usage of that service (message) were to increase significantly. This allows you to act now to prevent it from impacting production workloads.

Be a sponsor user

Sponsor users assist in design and implementation, and your feedback drives our development cycle.

Target personas

- Coverage Programmer
- Application Developer

Starting in

3Q2025

Interested? Contact

Mike Shershin (shershin@us.ibm.com)
Josh Wisniewski (jwisniew@us.ibm.com)

Thank you