mainstorconcept

z/IRIS®

YOUR MAINFRAME-INCLUSIVE OBSERVABILITY SOLUTION

KEY FEATURES

REAL-TIME TELEMETRY

Performance telemetry for mainframe applications is streamed near real-time to provide users with the latest state of system health.

OPENTELEMETRY SUPPORT

This CNCF project is an observability standard that defines the creation and exchange of application performance telemetry, such as traces and metrics. This framework enables sustainable and vendor-neutral integration capabilities that can be configured to address your enterprise observability requirements.

SEAMLESS INTEGRATION

Application telemetry from mainframe is easily ingested and processed by AIOps engines to provide teams with predictive analytics for mainframe integration.

SCALABLE ARCHITECTURE

A robust data streaming design that is effortlessly scaled up and out to easily meet the throughput of mainframe performance data in your enterprise environment.

Organizations implement observability tools (a.k.a. APM tools) to identify application bottlenecks and improve both availability and reliability. Insufficient coverage of all the systems that provide digital services for the business leads to skewed results due to only a portion of the data being available for analysis.

Industry-leading observability tools do not provide support for applications hosted on mainframe infrastructure, even though mainframes typically host core business services that manage customer crucial data. This is an immense problem for mainframe-backed enterprises, as it impacts their ability to achieve comprehensive and effective end-to-end application observability.

z/IRIS is a flexible software solution that adds performance data from mainframe applications to observability tools to enable end-to-end transaction workflow monitoring and performance management in real-time.

With z/IRIS, enterprise service graphs are extended to depict z/OS services like CICS, MQ and Db2 along with performance indicators like called and error rate and average latency. Depending on the enterprise tool implemented, users will be able to drill down to each request trace and analyze response-times and performance from front-end to IBM Z and AIOps engines will ingest mainframe data to detect and correlate anomalies.

MAINFRAME-INCLUSIVE OBSERVABILITY

z/IRIS creates traces for mainframe application activity that is triggered by distributed application requests. These traces are appended to distributed application traces in APM tools to extend the view of a request's workflow so that interactions with mainframe services becomes visible. This end-to-end observability ensures the comprehensive analysis of digital business services and the ability to identify bottlenecks and their impact to end-user experience.

ARCHITECTURE

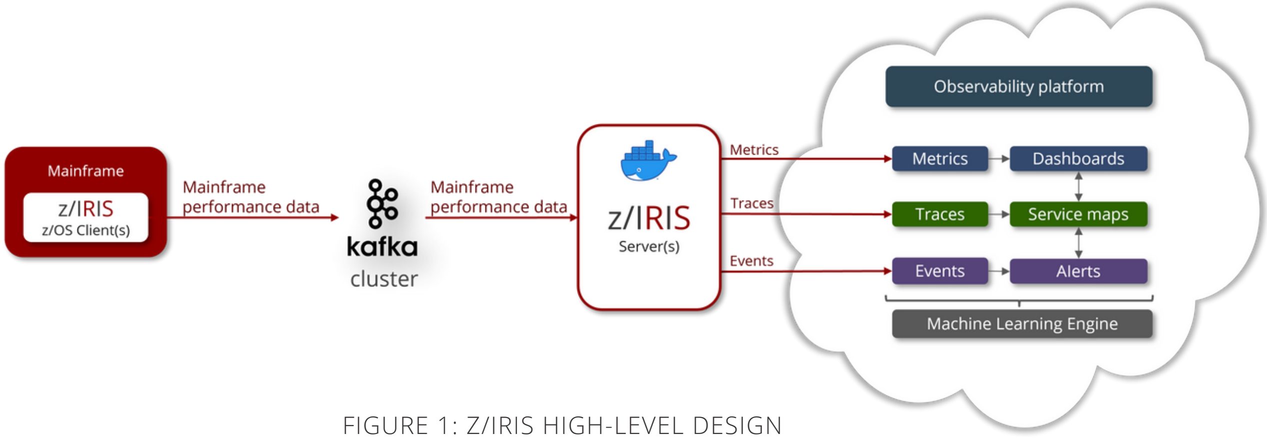


FIGURE 1: Z/IRIS HIGH-LEVEL DESIGN

z/IRIS is composed of one or more clients and one or more servers. A z/IRIS client is a z/OS application that runs as a started task on a z/OS LPAR. Users can run multiple z/OS clients to meet each environment's throughput demands of mainframe performance data. Each client is eligible to run on zIIP engines to offload their workloads to more cost-efficient mainframe processors. The clients read SMF data from predefined and customizable SMF in-memory resources and stream this data in real-time to a network accessible Apache Kafka cluster.

One or many z/IRIS servers read SMF data from the Kafka cluster near real-time and produce performance telemetry like traces and metrics by transforming, formatting and normalizing the SMF data into formats used by your chosen observability platform. An ever more popular format is OpenTelemetry (OTEL), which is supported by all industry-leading observability tools and ensures mainframe performance is available in any tool your organization implements for overall system health monitoring. z/IRIS mainframe telemetry is streamed near real-time to your observability tool and will be visible using all its relevant user interfaces and dashboards.

z/IRIS mainframe telemetry will be used to extend service maps to include mainframe applications, expand trace trees or flame graphs to include mainframe calls performed by a single request and enrich time-series databases with mainframe metrics to enable holistic monitoring of live systems.

ABOUT APACHE KAFKA

- More than 80% of all Fortune 100 companies trust, and use Kafka.
- Stream data at network limited throughput using a clustered deployment for the lowest latency.
- Clusters can be geographically dispersed or deployed across availability zones to enable high availability.
- Integrates with multiple sources and sinks using wide programming language support and open-source tools.
- On-premise, hybrid and public cloud environments fully supported.

OBSERVABILITY USE CASES

TRACES

A **distributed trace**, more commonly referred to as a **trace**, records how requests flow through mixed architecture services. A trace is composed of one or more **spans** that capture the performance of applications and services that process requests. Spans enable debugging, root-cause analysis and system-wide mapping of complex service dependencies. Spans contain additional metadata in the form of name:value tags that describe the system, request identification, resource consumption, latency and more.

CORRELATED REQUEST WORKFLOWS

z/IRIS generates **spans** for application and system tasks processed on z/OS. Depending on the request's workflow into mainframe applications, z/IRIS will enrich spans with trace context so that the mainframe application spans are correctly linked (i.e. correlated) to the request's trace. The graphic below details which workflows cause z/IRIS spans to be correlated to traces.

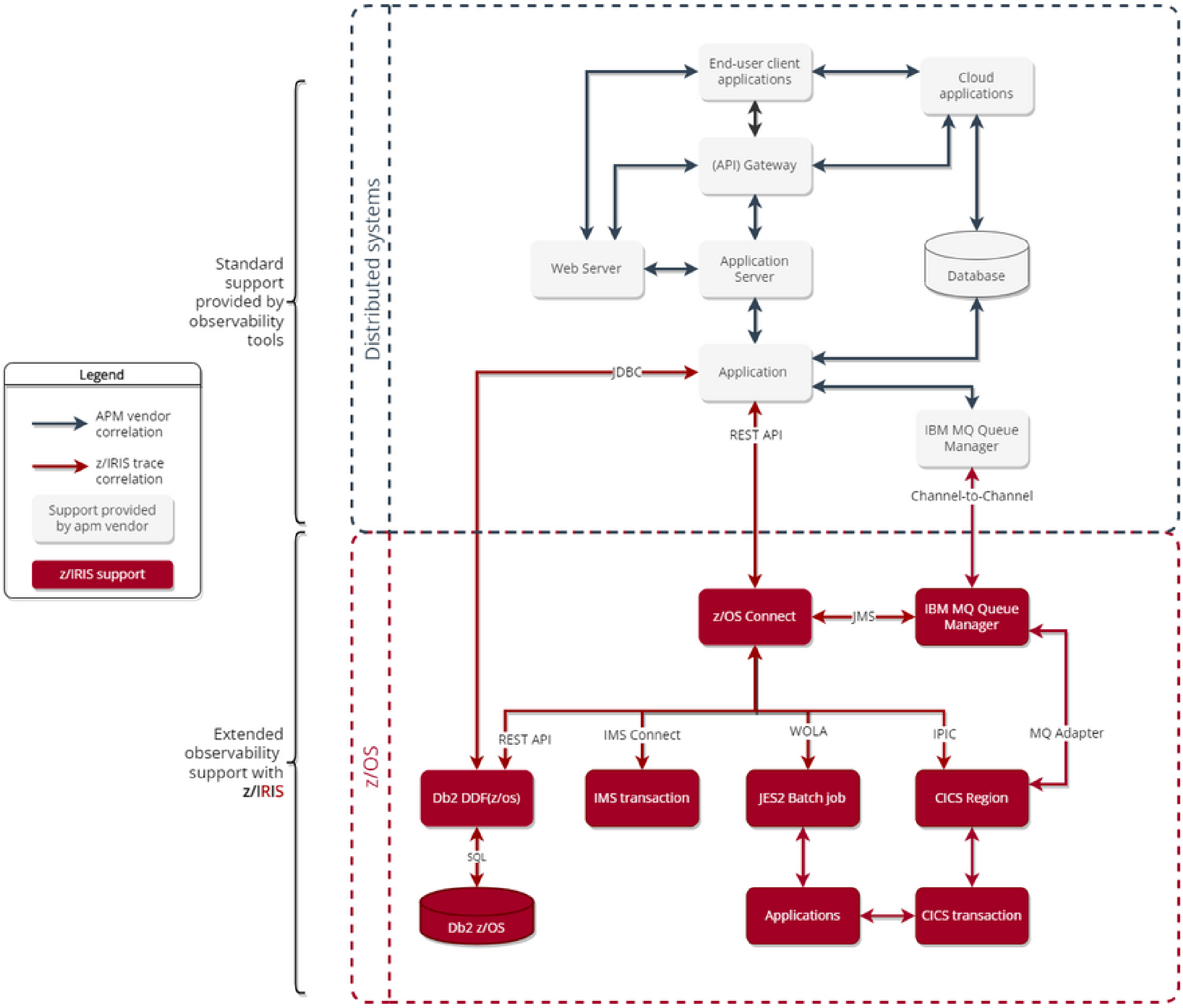
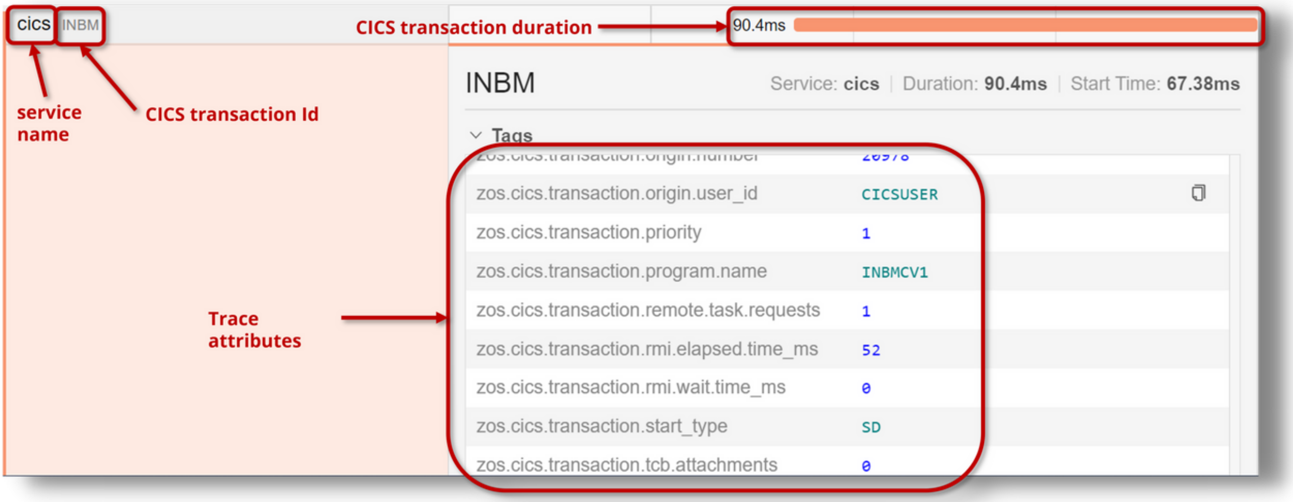


FIGURE 2: CORRELATED REQUEST WORKFLOWS

CICS TRANSACTION SPANS

z/IRIS creates OpenTelemetry spans that describe CICS transactions that executed on z/OS using SMF type 110 records.

Where possible, z/IRIS will provide additional correlation context that informs observability platforms which upstream application traces to append the CICS spans to so that user gain end-to-end visibility into how CICS transactions are utilized by business applications.



service name

CICS transaction Id

CICS transaction duration

Trace attributes

INBM

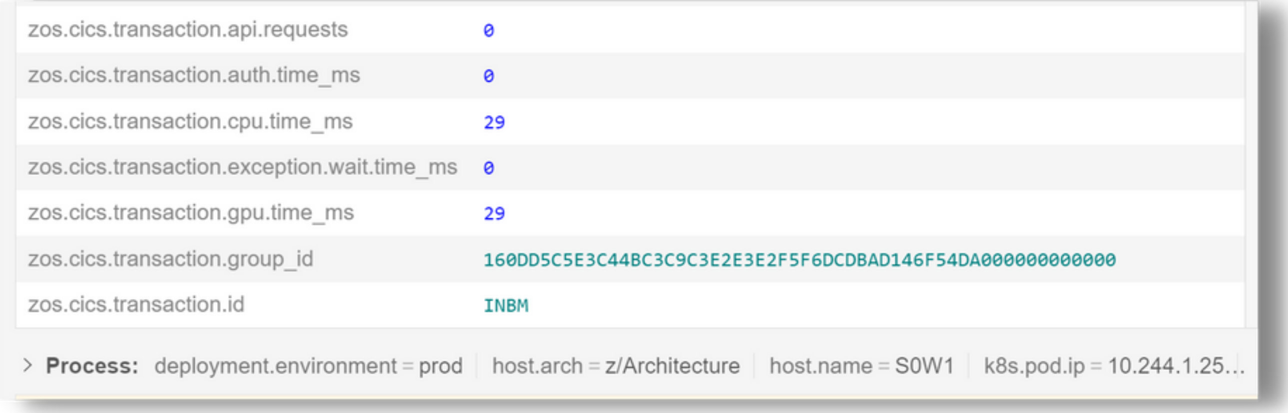
Service: **cics** | Duration: **90.4ms** | Start Time: **67.38ms**

Tags

zos.cics.transaction.origin.number	20978
zos.cics.transaction.origin.user_id	CICSUSER
zos.cics.transaction.priority	1
zos.cics.transaction.program.name	INBMCV1
zos.cics.transaction.remote.task.requests	1
zos.cics.transaction.rmi.elapsed.time_ms	52
zos.cics.transaction.rmi.wait.time_ms	0
zos.cics.transaction.start_type	SD
zos.cics.transaction.tcb.attachments	0

FIGURE 3: ANNOTATED CICS SPAN IN JAEGER

Each span identifies the transaction, the duration and the error status of the transaction's execution. Resource attributes describe the CICS subsystem, the z/OS system and any CICS resources utilized during execution. Trace attributes describe how the transaction was started, the peer/Client application that initiated the transaction, transaction properties, performance and resource consumption.



zos.cics.transaction.api.requests	0
zos.cics.transaction.auth.time_ms	0
zos.cics.transaction.cpu.time_ms	29
zos.cics.transaction.exception.wait.time_ms	0
zos.cics.transaction.gpu.time_ms	29
zos.cics.transaction.group_id	160DD5C5E3C44BC3C9C3E2E3E2F5F6DCDBAD146F54DA000000000000
zos.cics.transaction.id	INBM

> **Process:** deployment.environment = prod | host.arch = z/Architecture | host.name = S0W1 | k8s.pod.ip = 10.244.1.25...

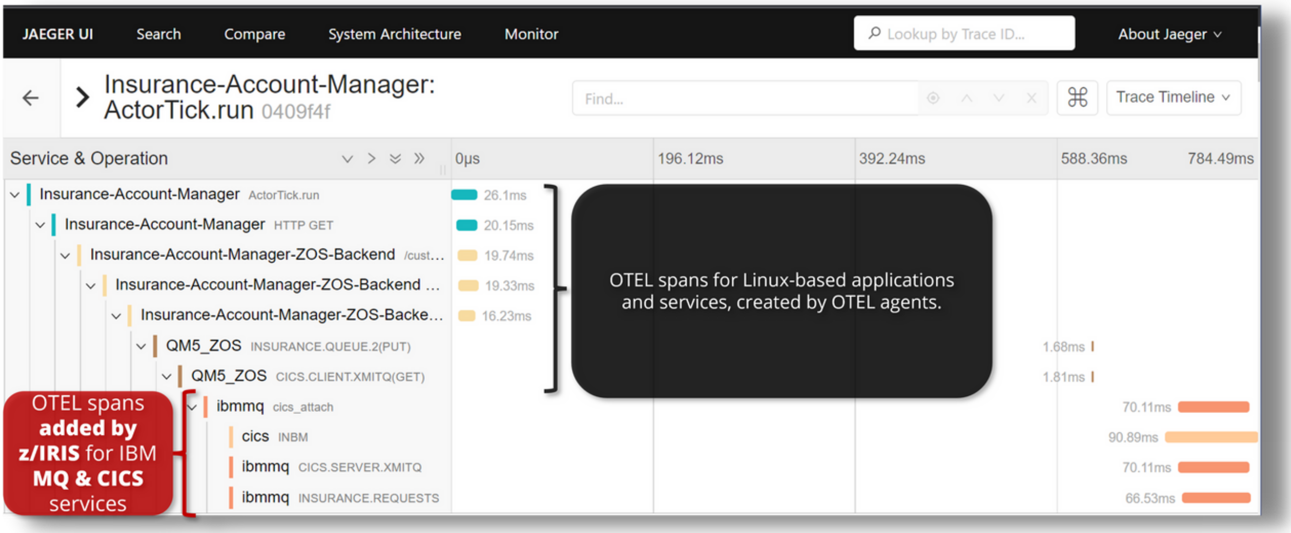
FIGURE 4: SAMPLE OF CICS ATTRIBUTES

MAINFRAME WORKFLOWS WITH CICS SPANS

CICS spans are appended to existing distributed traces under the following application workflows:

- MQ to CICS Workflow Tracing.

Contact z/IRIS engineering to find out about additional workflow support.



Insurance-Account-Manager: ActorTick.run 0409f4f

Service & Operation	0μs	196.12ms	392.24ms	588.36ms	784.49ms
Insurance-Account-Manager ActorTick.run	26.1ms				
Insurance-Account-Manager HTTP GET	20.15ms				
Insurance-Account-Manager-ZOS-Backend /cust...	19.74ms				
Insurance-Account-Manager-ZOS-Backend ...	19.33ms				
Insurance-Account-Manager-ZOS-Backe...	16.23ms				
QM5_ZOS INSURANCE.QUEUE.2(PUT)				1.68ms	
QM5_ZOS CICS.CLIENT.XMITQ(GET)				1.81ms	
ibmmq cics_attach					70.11ms
cics INBM					90.89ms
ibmmq CICS.SERVER.XMITQ					70.11ms
ibmmq INSURANCE.REQUESTS					66.53ms

OTEL spans for Linux-based applications and services, created by OTEL agents.

OTEL spans **added by z/IRIS** for IBM **MQ & CICS** services

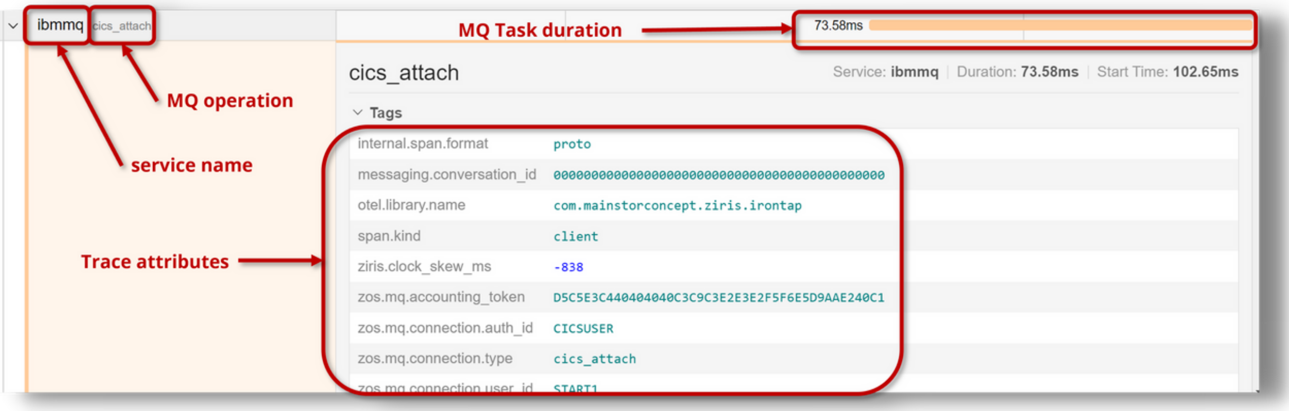
FIGURE 5: REQUEST TRACE WITH A CICS SPAN

IBM MQ FOR Z/OS SPANS

z/IRIS creates MQ traces for application observability using SMF record type 116 subtype 1 records. Each MQ trace details the performance and resource consumption, grouped by each queue or topic used by the task. Where possible, z/IRIS will provide additional correlation context that inform observability platforms which distributed, or cloud application initiated the MQ task workload on z/OS so that users gain end-to-end visibility into how IBM MQ on mainframe is utilized by business applications.

MQ FOR Z/OS PARENT SPAN

The parent span identifies the task, the MQ Queue Manager, connection and channels used as well as mainframe-specific identifiers for the unit of work.



cics_attach

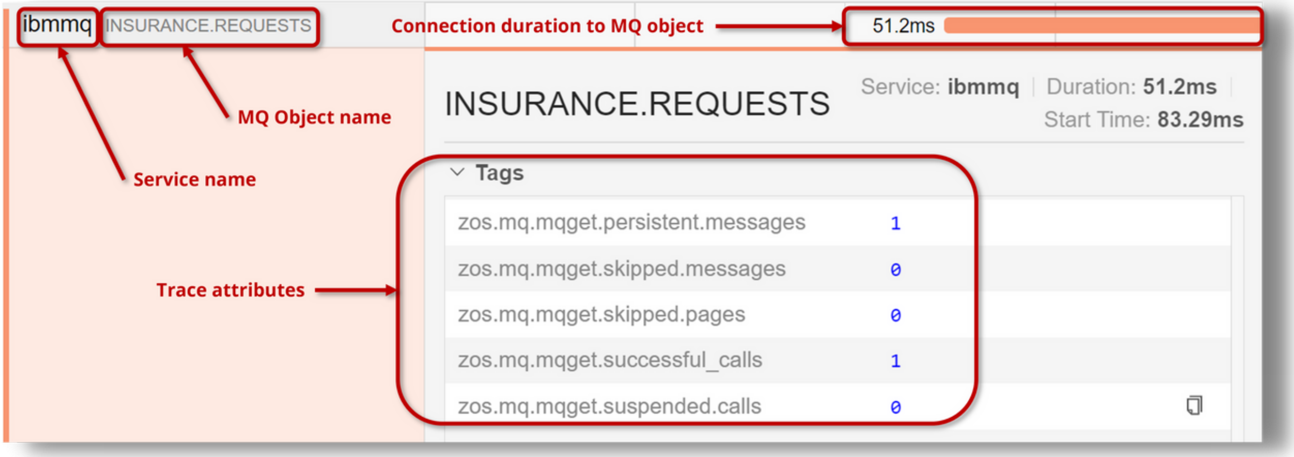
Service: **ibmmq** | Duration: **73.58ms** | Start Time: **102.65ms**

internal.span.format	proto
messaging.conversation_id	00000000000000000000000000000000000000000000
otel.library.name	com.mainstorconcept.ziris.irontap
span.kind	client
ziris.clock_skew_ms	-838
zos.mq.accounting_token	D5C5E3C440404040C3C9C3E2E3E2F5F6E5D9AAE240C1
zos.mq.connection.auth_id	CICSUSER
zos.mq.connection.type	cics_attach
zos.mq.connection.user_id	START1

FIGURE 6: ANNOTATED MQ PARENT SPAN IN JAEGER

MQ FOR Z/OS CHILD SPAN

Each child span identifies the MQ object, the duration for which the MQ task was connected to the object, and signals if any errors were detected by the queue manager during processing. The resource attributes describe the MQ object, the MQ subsystem and the MQ resources that were used. The trace attributes provide a breakdown of the MQ API calls the task executed against the MQ object, the processor utilization and internal latency incurred due to logging.



Connection duration to MQ object

MQ Object name

Service name

Trace attributes

INSURANCE.REQUESTS

Service: **ibmmq** | Duration: **51.2ms**
Start Time: **83.29ms**

Tags

zos.mq.mqget.persistent.messages	1
zos.mq.mqget.skipped.messages	0
zos.mq.mqget.skipped.pages	0
zos.mq.mqget.successful_calls	1
zos.mq.mqget.suspended.calls	0

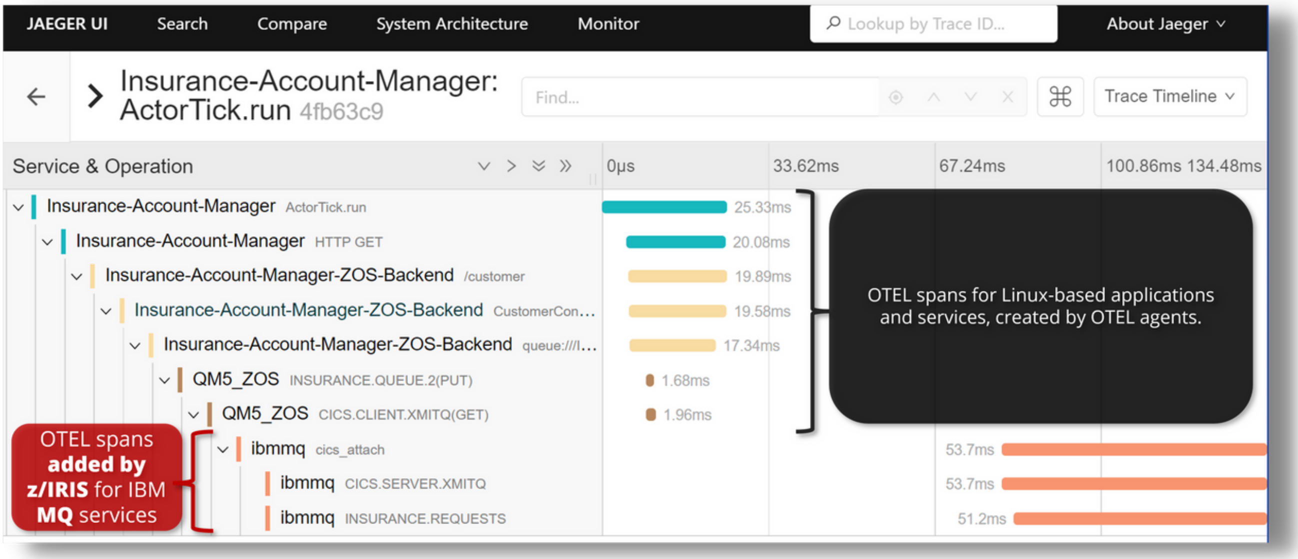
FIGURE 7: ANNOTATED MQ CHILD SPAN

MAINFRAME WORKFLOWS WITH MQ SPANS

MQ spans are appended to existing distributed traces under the following application workflows:

- Distributed MQ(JMS) to MQ for z/OS Queue Managers.

Contact z/IRIS engineering to find out about additional workflow support.



Insurance-Account-Manager: ActorTick.run 4fb63c9

Insurance-Account-Manager	ActorTick.run	25.33ms
Insurance-Account-Manager	HTTP GET	20.08ms
Insurance-Account-Manager-ZOS-Backend	/customer	19.89ms
Insurance-Account-Manager-ZOS-Backend	CustomerCon…	19.58ms
Insurance-Account-Manager-ZOS-Backend	queue:///I…	17.34ms
QM5_ZOS	INSURANCE.QUEUE.2(PUT)	1.68ms
QM5_ZOS	CICS.CLIENT.XMITQ(GET)	1.96ms
ibmmq	cics_attach	53.7ms
ibmmq	CICS.SERVER.XMITQ	53.7ms
ibmmq	INSURANCE.REQUESTS	51.2ms

OTEL spans for Linux-based applications and services, created by OTEL agents.

OTEL spans **added by z/IRIS** for IBM **MQ** services

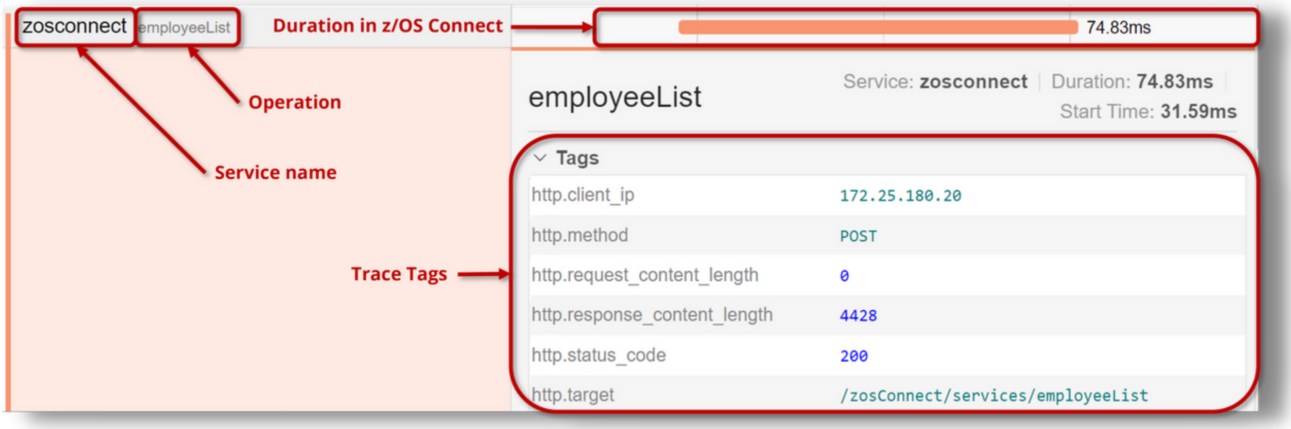
FIGURE 8: REQUEST TRACE WITH MQ SPANS

Z/OS CONNECT SPANS

z/IRIS creates traces from SMF 123 version 2 records that provide performance and system information for each REST-API requests processed on monitored z/OS Connect servers. Trace metadata/attributes can be used in capacity planning or application tuning queries to achieve end-to-end application analysis results.

Z/OS CONNECT PARENT SPAN

The parent span describes the execution of a REST API request in the z/OS Connect server. It identifies the API called, provides the duration of the request in the z/OS Connect server, and indicates whether an error was detected during processing.



employeeList

Service: **zosconnect** | Duration: **74.83ms** | Start Time: **31.59ms**

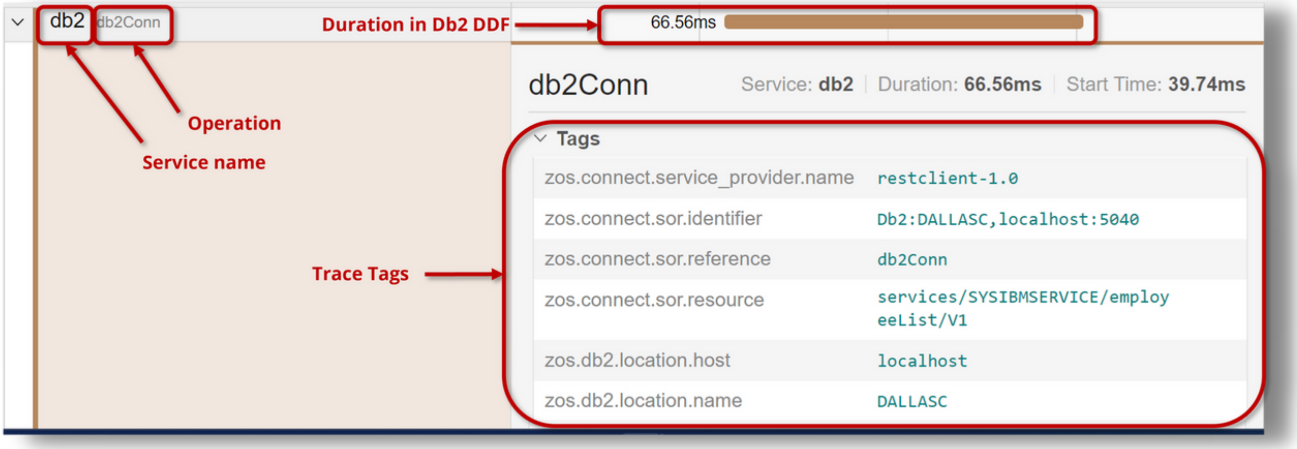
Tags

http.client_ip	172.25.180.20
http.method	POST
http.request_content_length	0
http.response_content_length	4428
http.status_code	200
http.target	/zosConnect/services/employeeList

FIGURE 9: ANNOTATED ZOSCONNECT PARENT SPAN

Z/OS CONNECT CHILD SPAN

The child span describes the processing of the REST API request in the System of Record (SOR). that could be CICS TS, IMS TM, MQ or Db2 on z/OS. The child span identifies the SOR, provides the duration the request spent within the SOR and indicates whether an error was detected during processing.



db2Conn

Service: **db2** | Duration: **66.56ms** | Start Time: **39.74ms**

Tags

zos.connect.service_provider.name	restclient-1.0
zos.connect.sor.identifier	Db2:DALLASC,localhost:5040
zos.connect.sor.reference	db2Conn
zos.connect.sor.resource	services/SYSIBMSERVICE/employeeList/V1
zos.db2.location.host	localhost
zos.db2.location.name	DALLASC

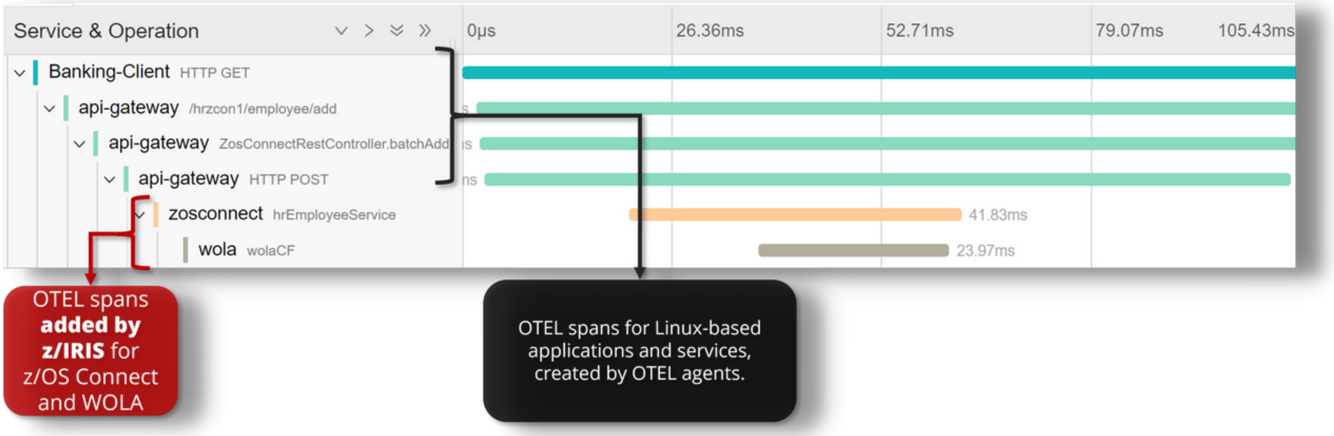
FIGURE 10: ANNOTATED ZOSCONNECT SOR CHILD SPAN

MAINFRAME WORKFLOWS WITH ZOSCONNECT SPANS

z/OS Connect spans are appended to existing distributed traces under the following application workflows:

- REST API requests processed by z/OS Connect Servers and its SORs.

Contact z/IRIS engineering to find out about additional workflow support.



OTEL spans **added by z/IRIS** for z/OS Connect and WOLA

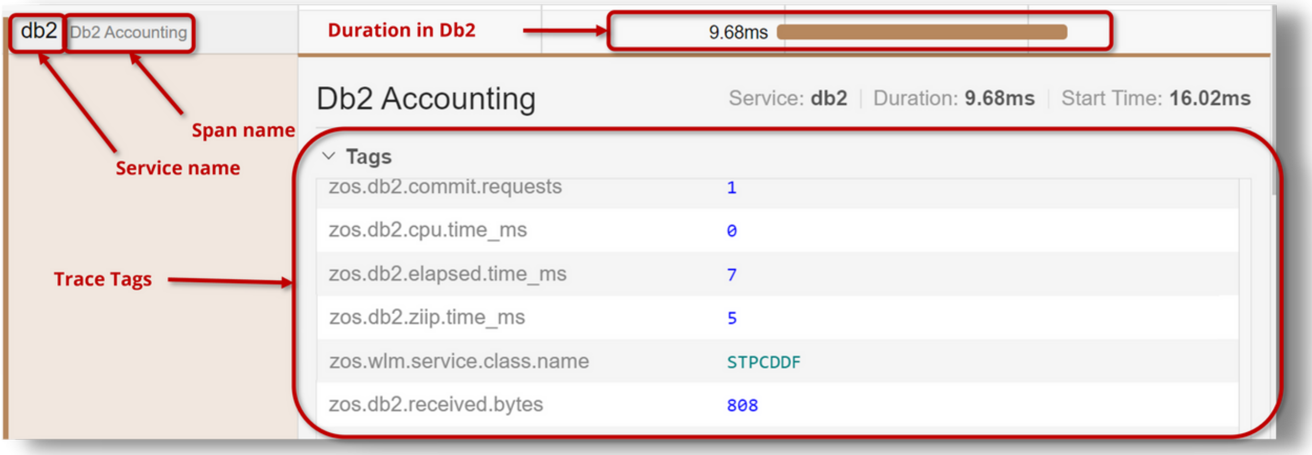
OTEL spans for Linux-based applications and services, created by OTEL agents.

FIGURE 11: REQUEST TRACE WITH ZOSCONNECT & SOR SPANS

DB2 FOR Z/OS

DB2 ACCOUNTING SPANS

z/IRIS streams Db2 for z/OS Accounting spans, created using SMF 101 records with IFCID 3 data. Users gain access to performance metrics and processing information related to the Db2 call processing. z/IRIS will also create events with information about deadlocks that occurred on the z/OS that impact the requests' response time or availability.



Db2 Accounting

Service: **db2** | Duration: **9.68ms** | Start Time: **16.02ms**

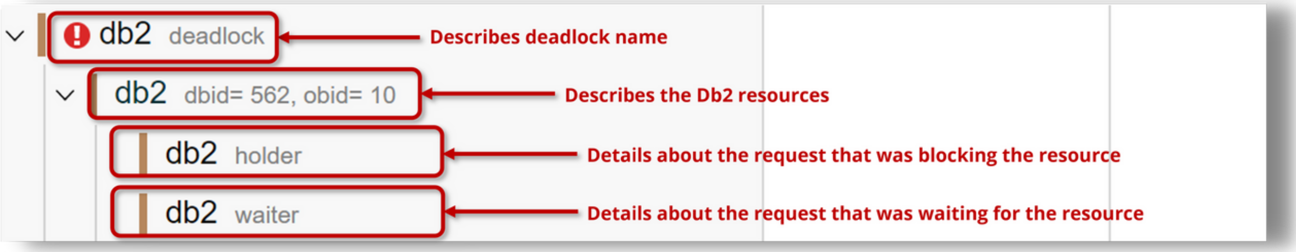
Tags

zos.db2.commit.requests	1
zos.db2.cpu.time_ms	0
zos.db2.elapsed.time_ms	7
zos.db2.ziip.time_ms	5
zos.wlm.service.class.name	STPCDDF
zos.db2.received.bytes	808

FIGURE 9: ANNOTATED SCREENSHOT OF A DB2 SPAN

DB2 DEADLOCK SPANS

z/IRIS will signal that a request deadlocked on a Db2 for z/OS resource using the Deadlock span, created from SMF Type 101 with IFCID 172 data. The deadlock span contains a parent that summarizes the deadlock, a child span that describes the database resource, and 2 further child spans describing either the blocking request or the waiting request.



db2 deadlock
db2 dbid= 562, obid= 10
db2 holder
db2 waiter

Describes deadlock name

Describes the Db2 resources

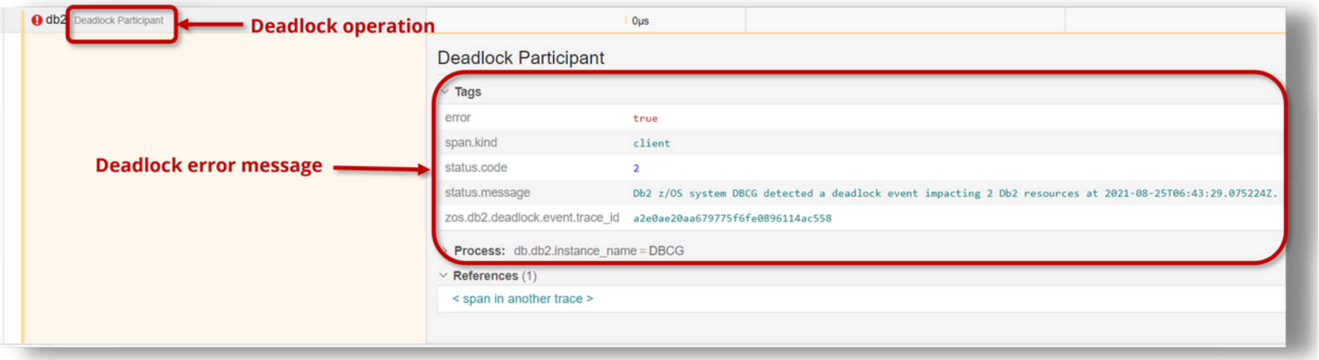
Details about the request that was blocking the resource

Details about the request that was waiting for the resource

FIGURE 10: ANNOTATED DB2 DEADLOCK SPANS

DEADLOCK PARTICIPANT SPANS

A span is added to request traces that deadlocked on Db2 for z/OS resources. The span indicates the Db2 location where the deadlock occurred and provides information about other requests that deadlocked on the same Db2 for z/OS resources



db2 Deadlock Participant

0μs

Deadlock operation

Deadlock error message

Deadlock Participant

Tags

error	true
span.kind	client
status.code	2
status.message	Db2 z/OS system DBCG detected a deadlock event impacting 2 Db2 resources at 2021-08-25T06:43:29.075224Z.
zos.db2.deadlock.event.trace_id	a2e0ae20aa679775f6fe0896114ac558

Process: db.db2.instance_name = DBCG

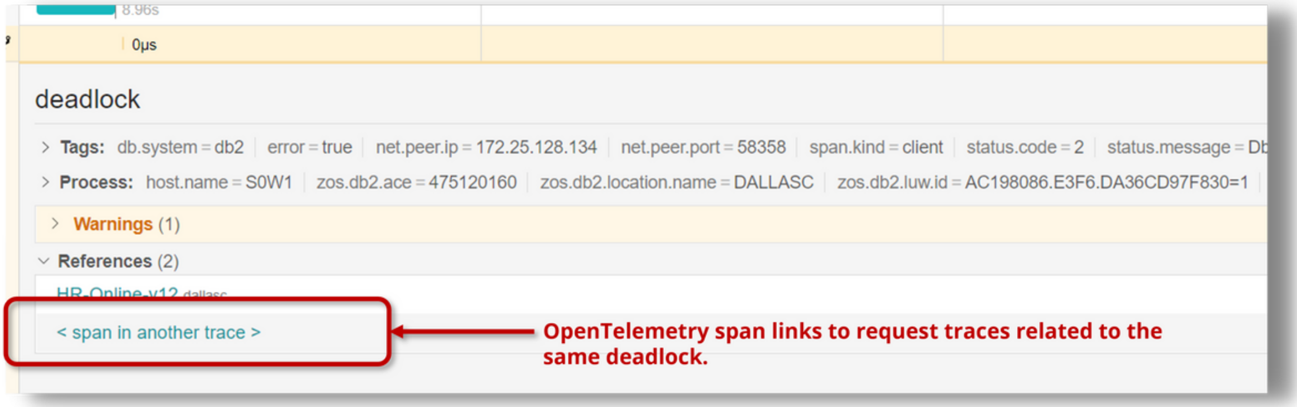
References (1)

< span in another trace >

FIGURE 11: ANNOTATED DB2 PARTICIPANT SPAN

SPAN LINKS

Using OpenTelemetry, z/IRIS can add hyperlinks to simplify navigating between requests traces that were impacted by the same that deadlock.



8.96s

0μs

deadlock

Tags: db.system = db2 | error = true | net.peer.ip = 172.25.128.134 | net.peer.port = 58358 | span.kind = client | status.code = 2 | status.message = Db

Process: host.name = S0W1 | zos.db2.ace = 475120160 | zos.db2.location.name = DALLASC | zos.db2.luw.id = AC198086.E3F6.DA36CD97F830=1

Warnings (1)

References (2)

HR-Online-v12 dallasc

< span in another trace >

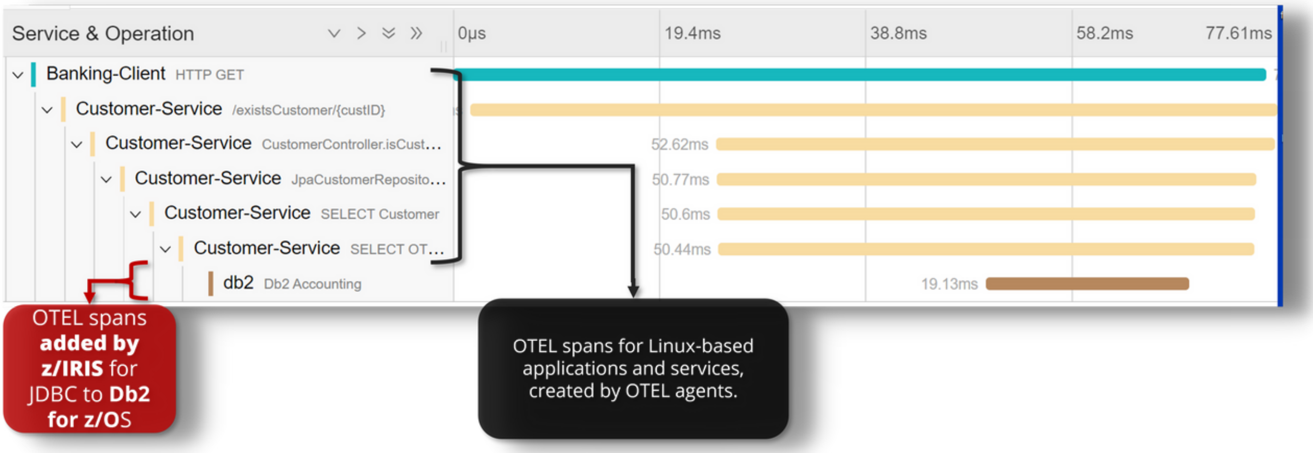
OpenTelemetry span links to request traces related to the same deadlock.

FIGURE 12: ANNOTATED DB2 DEADLOCK SPAN LINKS

MAINFRAME WORKFLOWS WITH DB2 FOR Z/OS SPANS

The following application workflows result in a Db2 for z/OS Span being added to a trace:

- JDBC from a Java application to Db2 for z/OS subsystem
- z/OS Connect APIs processed by the Db2 Native REST service


Service & Operation
0μs
19.4ms
38.8ms
58.2ms
77.61ms
Banking-Client HTTP GET
Customer-Service /existsCustomer/{custID}
Customer-Service CustomerController.isCust…
Customer-Service JpaCustomerReposito…
Customer-Service SELECT Customer
Customer-Service SELECT OT…
db2 Db2 Accounting
52.62ms
50.77ms
50.6ms
50.44ms
19.13ms

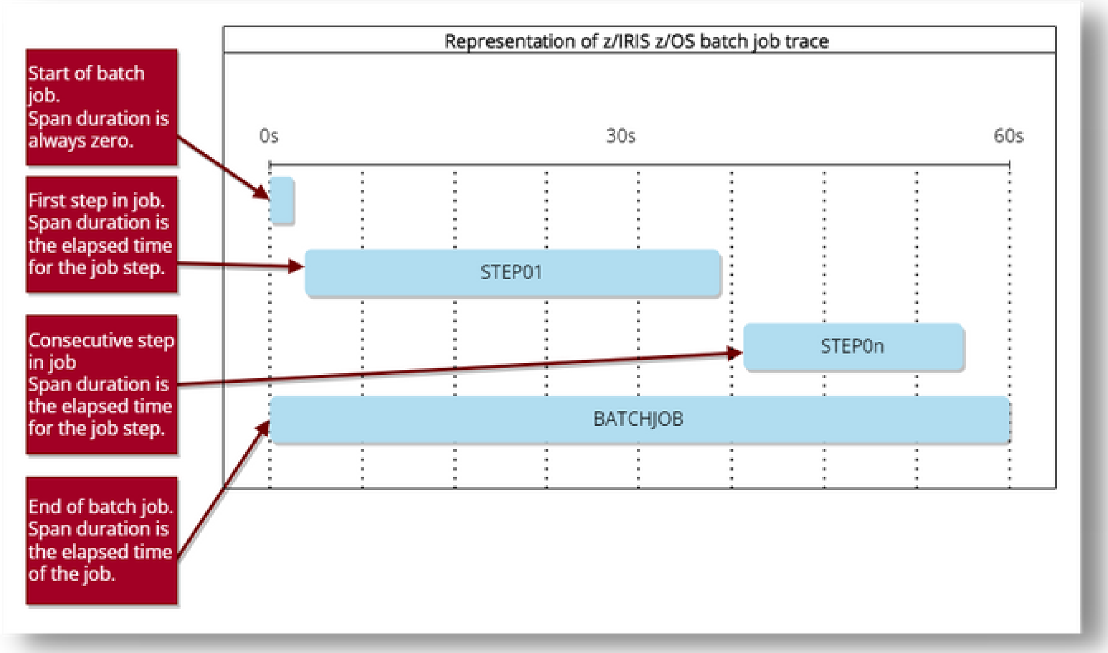
OTEL spans **added by z/IRIS** for JDBC to **Db2 for z/OS**

OTEL spans for Linux-based applications and services, created by OTEL agents.

FIGURE 12: REQUEST TRACE WITH DB2 SPAN

Z/OS WORK TRACES (SMF TYPE 30)

z/OS Work refers to JES2 batch jobs, TSO session activity and Unix processes. These traces enable users to monitor z/OS Work latency, frequency and error rate as well as consumption of mainframe resources like processor and tape mounts. Users can identify long-running tasks, CPU-intensive batch jobs and processes and have direct access to errors and error messages directly in the organization's strategic APM tools.



Start of batch job.
Span duration is always zero.

First step in job.
Span duration is the elapsed time for the job step.

Consecutive step in job
Span duration is the elapsed time for the job step.

End of batch job.
Span duration is the elapsed time of the job.

FIGURE 13: REPRESENTATION OF A BATCH JOB TRACE

METRICS AND DASHBOARDS

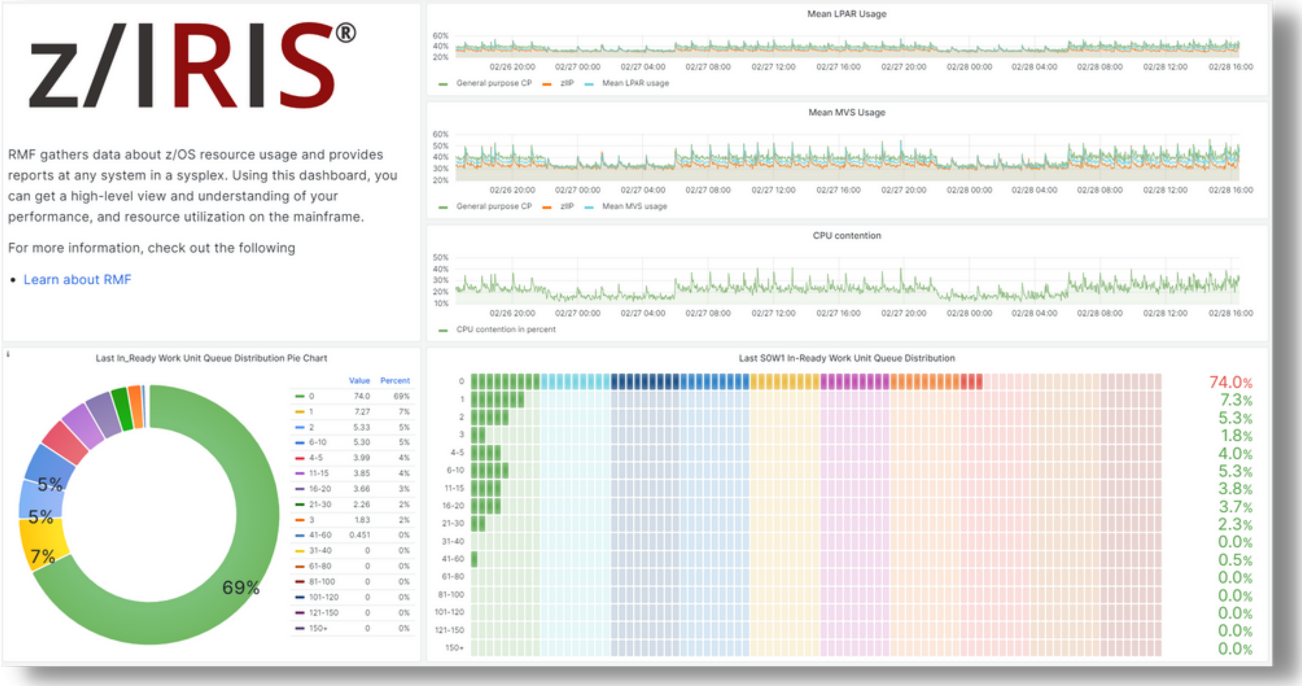
Metrics are time-based measurements captured during runtime from applications and systems. These capture important measurements that can either indicate workload, or signal that an error has occurred. In some cases, a hiatus of measurements may be indicative of a service failure.

Typically, metrics are used to monitor availability and overall performance by using dashboards. The z/IRIS service includes importable Grafana Dashboards to help users obtain value faster from z/IRIS metrics. All dashboards can be changed and configured before and after importing to better meet your organization's monitoring needs.

MAINFRAME INFRASTRUCTURE METRICS AND DASHBOARD

Measurements include for CPU load per processor type (i.e. CPU vs zIIP) and CPU contention and monitors how many work units are waiting for processor time. Infrastructure metrics are created from RMF SMF type 70 records.

The dashboard succinctly visualizes the 3 Infrastructure measurements. Additionally, an IBM recommended CPU contention threshold is preconfigured and can be changed to send alerts to the relevant Operations teams and Platform specialists.



z/IRIS®

RMF gathers data about z/OS resource usage and provides reports at any system in a sysplex. Using this dashboard, you can get a high-level view and understanding of your performance, and resource utilization on the mainframe.

For more information, check out the following

- Learn about RMF

Last In_Ready Work Unit Queue Distribution Pie Chart

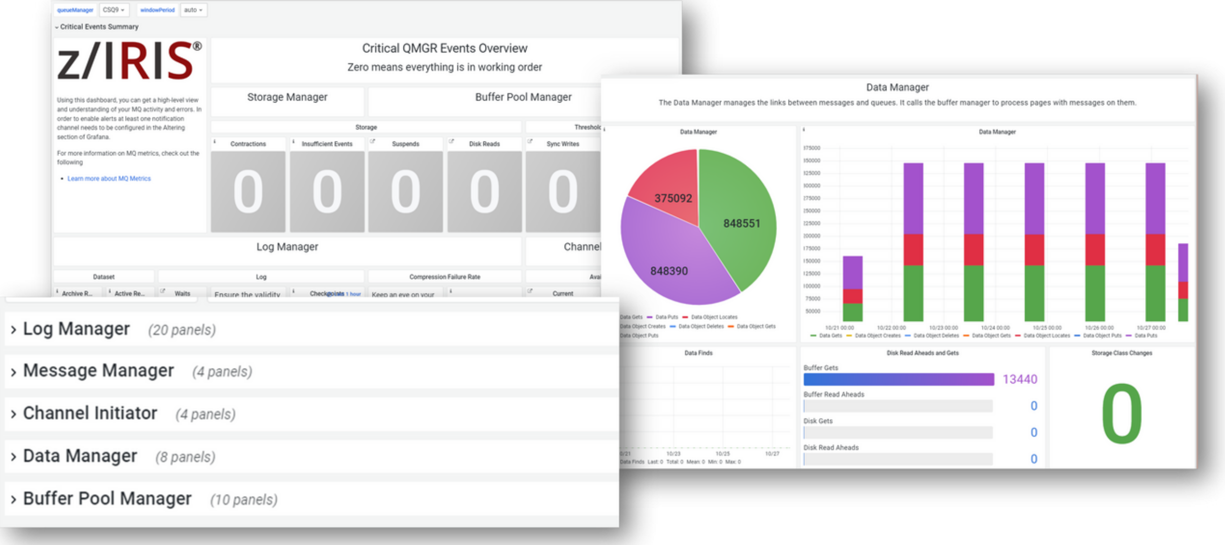
	Value	Percent
0	74.0	69%
1	7.27	7%
2	5.33	5%
6-10	5.30	5%
4-5	3.99	4%
11-15	3.85	4%
16-20	3.66	3%
21-30	2.26	2%
3	1.83	2%
41-60	0.451	0%
31-40	0	0%
61-80	0	0%
81-100	0	0%
101-120	0	0%
121-150	0	0%
150+	0	0%

FIGURE 14: INFRASTRUCTURE DASHBOARD

MQ QUEUE MANAGER METRICS AND DASHBOARD

Queue managers on z/OS provide critical pathways to mainframe applications. z/IRIS provides IBM MQ metrics for queue managers on z/OS that capture their health and activity and indicate crucial constraints in resources that affect their performance.

The importable dashboards are spearheaded by a overview of event metrics that signal whether critical shortages have occurred or thresholds were exceeded. Following the overview, users can find detailed visualization of activity and utiluation per queue manager.



Using this dashboard, you can get a high-level view and understanding of your MQ activity and errors. In order to enable alerts at least one notification channel needs to be configured in the Altering section of Grafana.

For more information on MQ metrics, check out the following

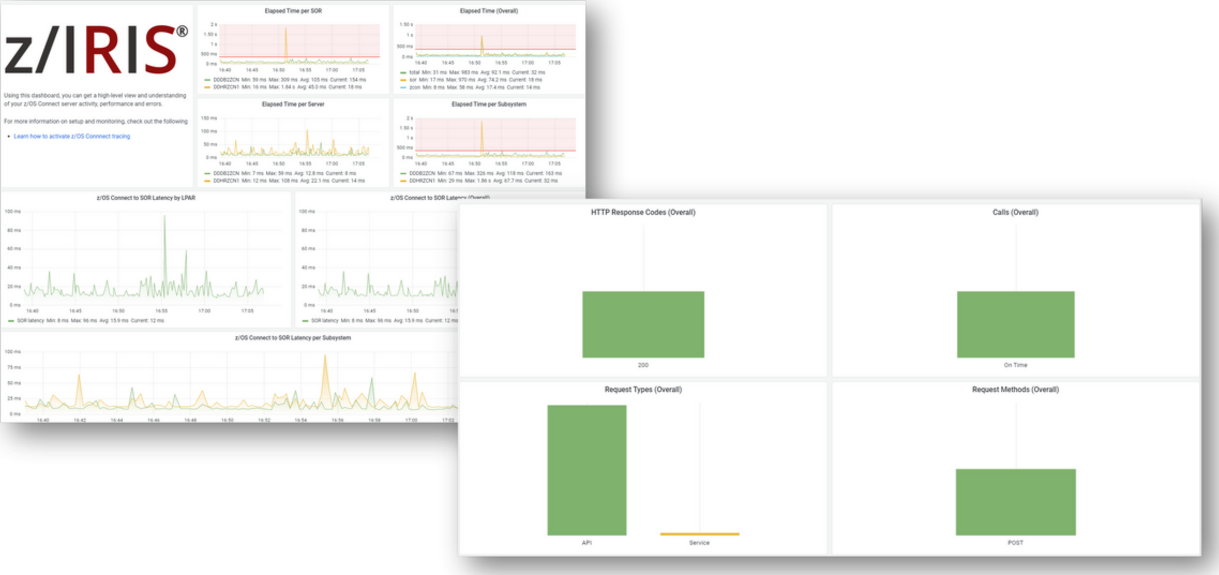
- Learn more about MQ Metrics

The Data Manager manages the links between messages and queues. It calls the buffer manager to process pages with messages on them.

FIGURE 15: MQ FOR Z/OS QUEUE MANAGER DASHBOARD

Z/OS CONNECT METRICS AND DASHBOARD

z/OS Connect metrics to monitor latency, workload and activity for active servers on mainframe. The importable dashboards can depict a single z/OS Connect server or group the measurements by LPAR or SYSPLEX.



Using this dashboard, you can get a high-level view and understanding of your z/OS Connect server activity, performance and errors.

For more information on setup and monitoring, check out the following

- Learn how to activate z/OS Connect tracing

z/IRIS®
Elapsed Time per SOR
Elapsed Time (Overall)
Elapsed Time per Server
Elapsed Time per Subsystem
z/OS Connect to SOR Latency by LPAR
z/OS Connect to SOR Latency (Overall)
z/OS Connect to SOR Latency per Subsystem
HTTP Response Codes (Overall)
200
Calls (Overall)
On Time
Request Types (Overall)
API
Service
Request Methods (Overall)
POST

FIGURE 16: MQ FOR Z/OS CONNECT SERVER DASHBOARD

INTEGRATIONS

DATADOG

INSTANA

JAEGER

OpenTelemetry

Grafana

influxdata®

new relic.

ABOUT MAINSTORCONCEPT

We provide enterprise solutions and services for mainframe and server technlogies. Our specialists are experts in storage, system administration, backup and disaster recovery as well as cyber security and resiliency. Our in-house solutions are catered to meet customer requirements and can be customized to meet your organizations individual needs and long-term strategies.

mainstorconcept was founded in 2005, as a consulting company in the mainframe sector. Additional services, such as software solutions, system management and enterprise servers and storage were added to our portfolio as the needs of the industry evolved. Our customers range from large enterprises to medium and small sized companies from the insurance, financial as well as the public sector, and rely on our extensive knowlegde, partnerships and innovation to support them in their technological and operational challenges.

CONTACT US

A z/IRIS demonstration is the perfect way to understand the product and its integration capabilities. Our experts will also provide professional consulting to understand your observability requirements and determine how z/IRIS will contribute to your enterprise strategy and goals.

www.mainstorconcept.com

public.mainstorconcept.com

ziris@mainstorconcept.com

Follow our updates and blogs on LinkedIn

mainstorconcept

mainstorconcept GmbH
Carl-Metz-Str. 15
76275 Ettlingen
Germany

Subscribe our YouTube cannel