

Hybrid Cloud Monitoring with Instana

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.

What we heard from you



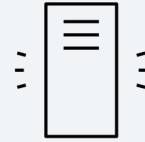
Middleware is always blamed when something goes wrong



How to know that there is a problem with a specific service on z/TPF



z/TPF is always blamed when something goes wrong



z/TPF needs to be regarded the same as other servers

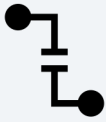
“I want to quickly figure out *what* the problem is and connect with the right SME to fix it.”

“I don’t want to spend a lot of time proving my system *is not* the problem”

- Coverage programmer



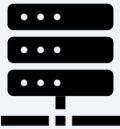
z/TPF Specific Use cases



Messages are not getting to the z/TPF system



Services z/TPF applications call, are taking a long time to respond



The z/TPF system itself is ok, but there is a problem with a specific service

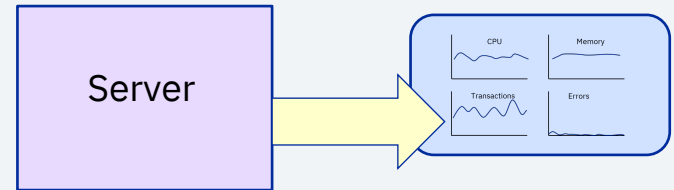
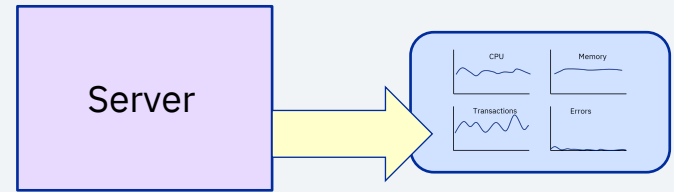
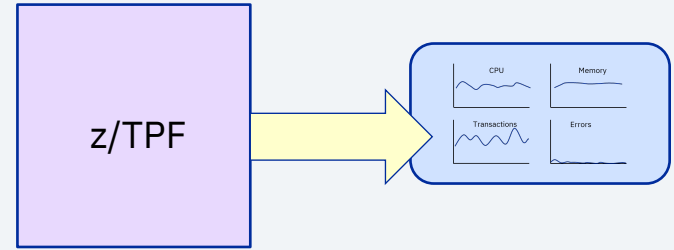
In all these cases, the problem surfaces as an impact to one or more end users

As-is monitoring

When the various components are monitored each within their own silo, a small problem may go unnoticed, or may not understand the impact it has on the end users

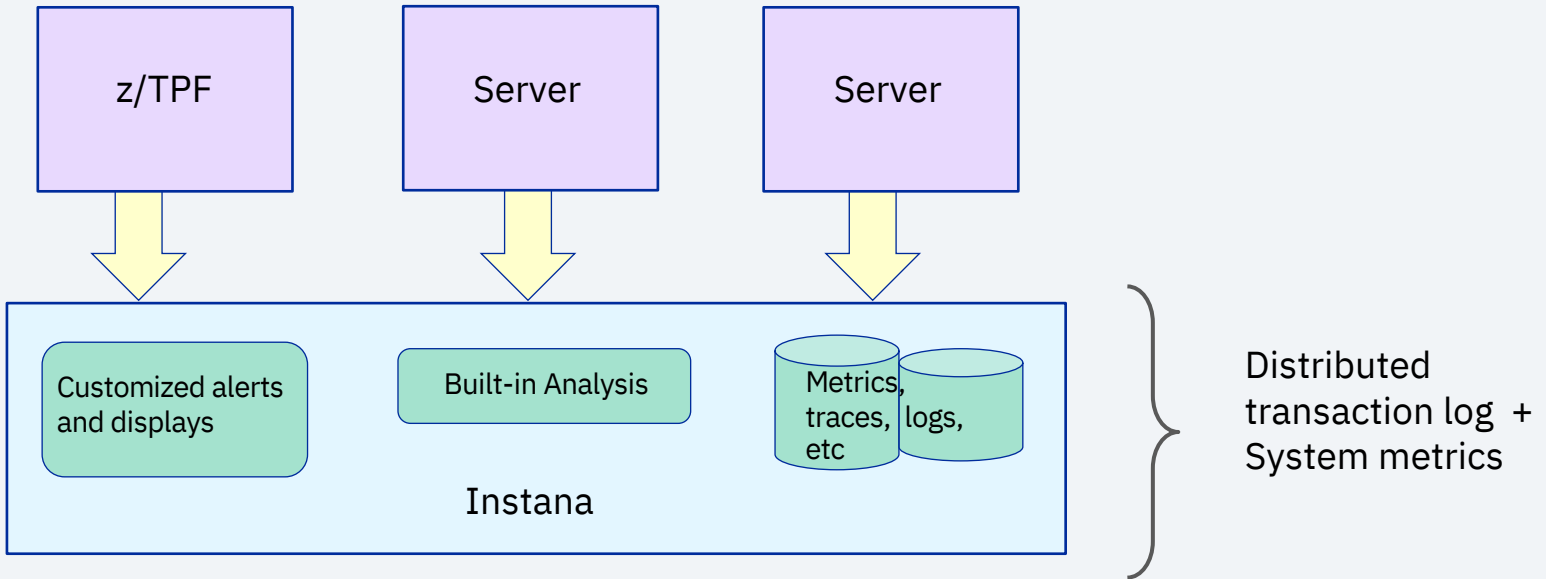
Example: one service is taking errors for a subset of calls

From a monitoring perspective the component where that service is running may seem to be operating normally.



To-be Monitoring

A single platform that is monitoring all systems that can correlate events across the various components, and provide an end-to-end view of specific transaction flows both at the service level as well as the machine level

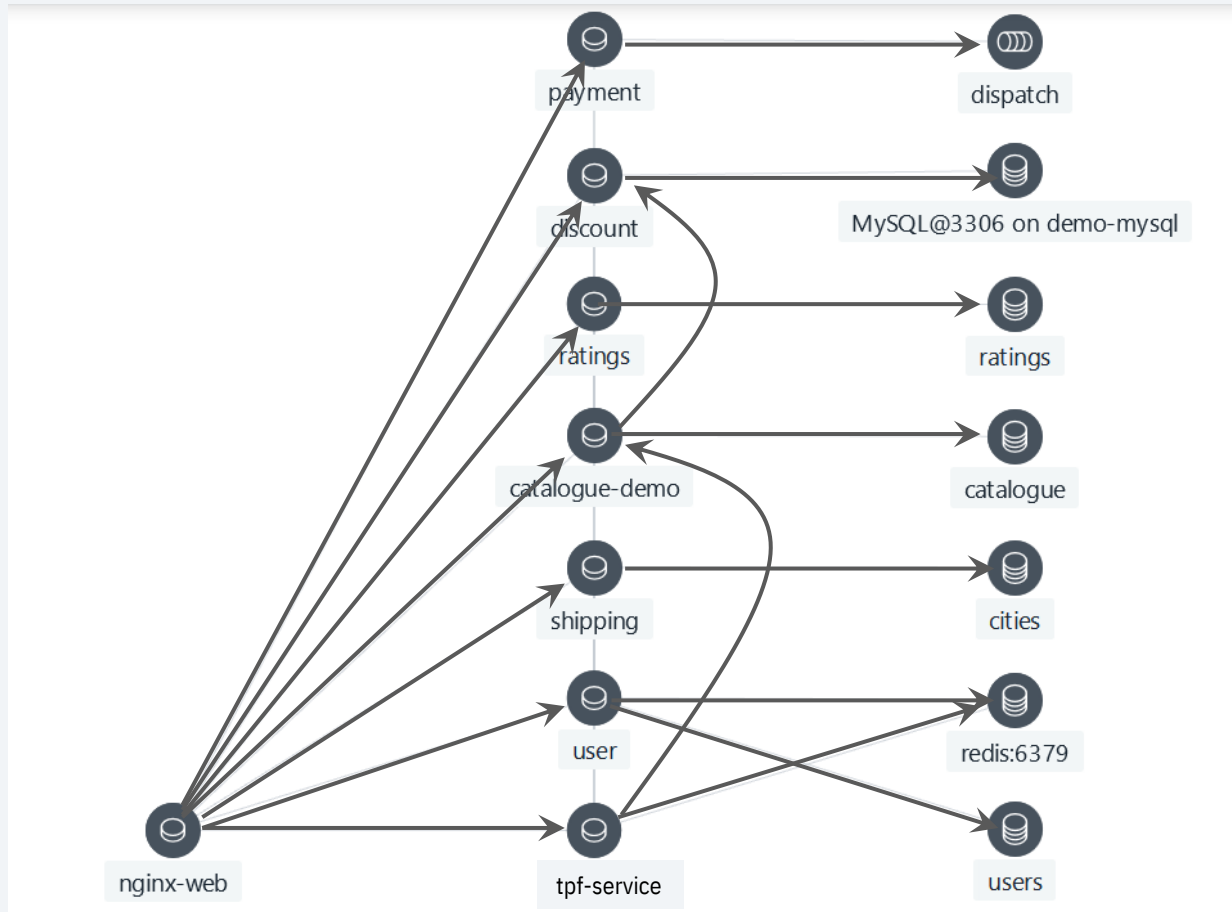


A Simple Example

Robot-Shop consists of a front end process that makes calls to several service calls, one of which calls a service on z/TPF which in turn makes outbound calls to another service.

All components are sending data to the monitoring platform where the information is analyzed and correlated across all services. Data collected includes [latency](#) and [error rates](#).

Alerts have been set up to notify the Site Reliable Engineer (SRE) of any incidents.



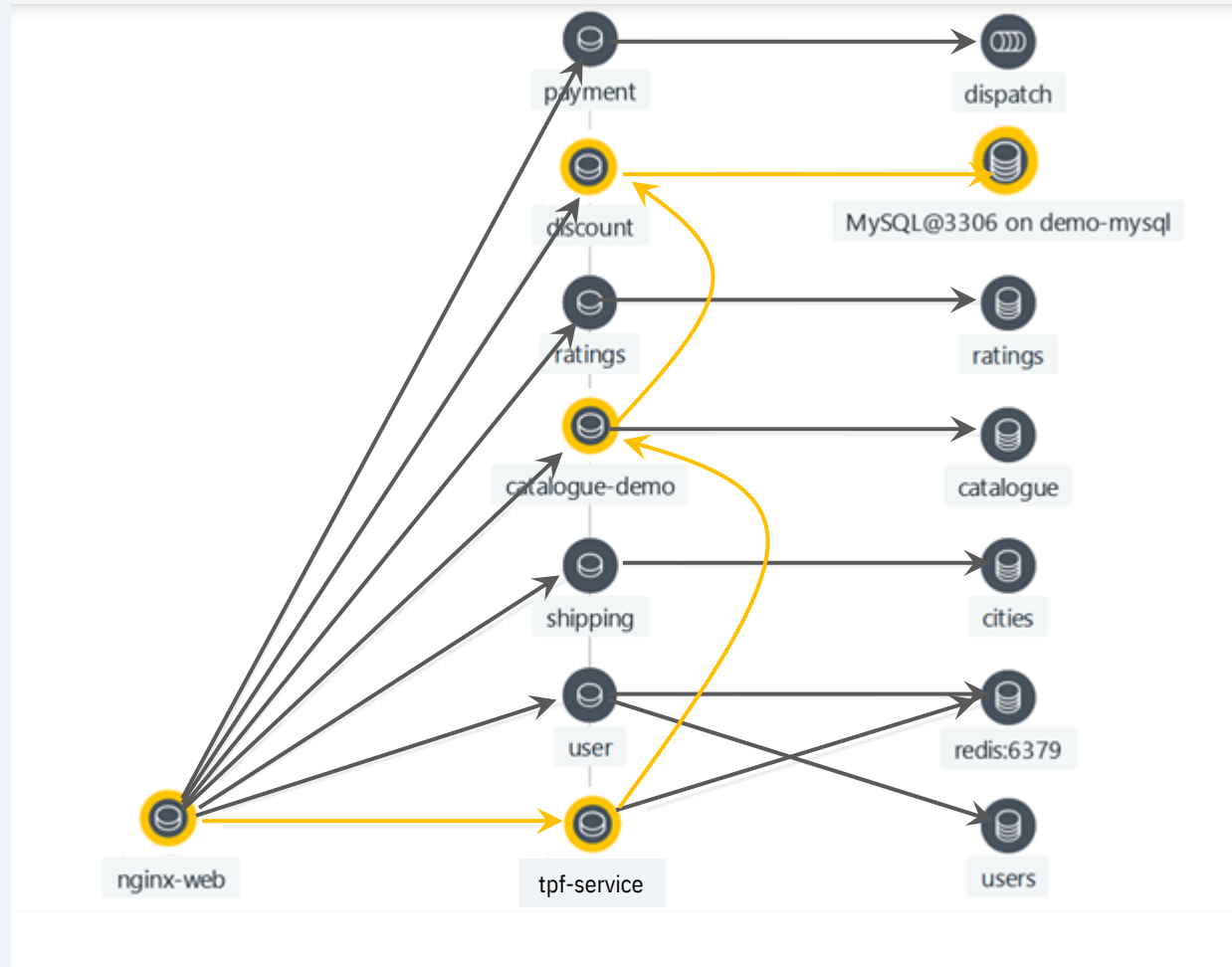
The Situation

The Site Reliability Engineer has set up configured incidents for the Robot-Shop to be sent to PagerDuty.

This will allow him to quickly respond if something in the environment is impacting the experience of the end users.

In our example, the termination of the backend service MySQL causes an impact to only a portion of the end user requests.

Let's see how the monitoring solution helps to quickly identify the problem.



Incident Alert

The SRE gets an incident alert and is taken to the event page to see what is the problem and when did it happen.

Triggering Event

Sudden increase in the number of erroneous calls in service discount

Event Mar 12 Last hour Live

All Incidents Issues Changes Monitoring issues Filters

Sudden increase in the number of erroneous calls

Triggered 2021-03-12 15:30:40	Ended 2021-03-12 15:41:50	Duration 11m 10s	Active 0/8	Changes 1	Affected entities 3
-------------------------------------	---------------------------------	---------------------	---------------	--------------	------------------------

Incident Timeline Show Changes

2021-03-12 15:27:40 | 15:29 | 15:31 | 15:33 | 15:35 | 15:37 | 15:39

Triggering Event

SERVICE IMPACT **TRIGGERING EVENT**

15:30:40 • Sudden increase in the number of erroneous calls ENDED 2021-03-12 15:40:40 DURATION 10m
On: discount

Related Events

In addition to seeing the event that triggered the incident, all related events are displayed to bring the incident into context.

Related Events (7)
Abnormal termination
/usr/sbin/msqld

The screenshot shows a web-based incident management interface. At the top, there's a navigation bar with a home icon, a warning icon, and the word "Event". To the right, there are buttons for "Share", "Mar 12 Last hour", and "Live". Below this is a filter bar with tabs for "All", "Incidents", "Issues", "Changes", and "Monitoring issues", along with a search box and a "Filters" dropdown. The main content area displays an incident titled "Sudden increase in the number of erroneous calls". Below the incident title, a section titled "Related Events (7)" lists several events. A blue oval highlights the first two events: "Abnormal termination" at 15:30:01 and "offline" at 15:30:37. The "Abnormal termination" event is the one mentioned in the text on the left. Other events include "Sudden increase in the number of erroneous calls" (15:30:40), "Sudden increase in average latency" (15:31:50), "Erroneous call rate is too high" (15:33:00), and another "Sudden increase in average latency" (15:31:50) and "Erroneous call rate is too high" (15:33:00) event.

Time	Event Type	Duration
15:30:01	Abnormal termination	1m
15:30:37	offline	
15:30:40	Sudden increase in the number of erroneous calls	10m
15:31:50	Sudden increase in average latency	10m
15:33:00	Erroneous call rate is too high	5m
15:31:50	Sudden increase in average latency	10m
15:33:00	Erroneous call rate is too high	5m

Upstream / Downstream Calls

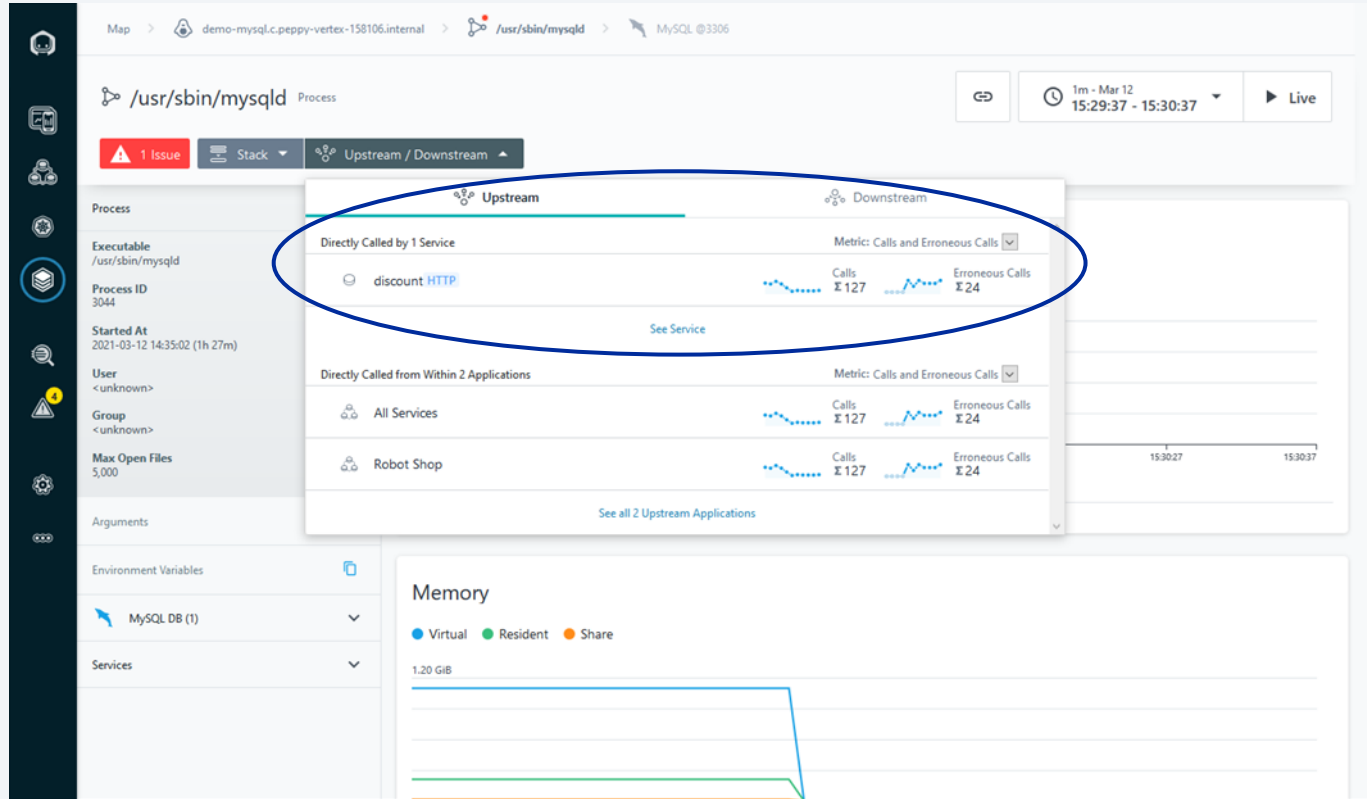
Clicking on the abnormal termination event, the SRE can see both the upstream and downstream callers.

Upstream

Directly called by 1 Service

discount

This is the most likely cause of the problem

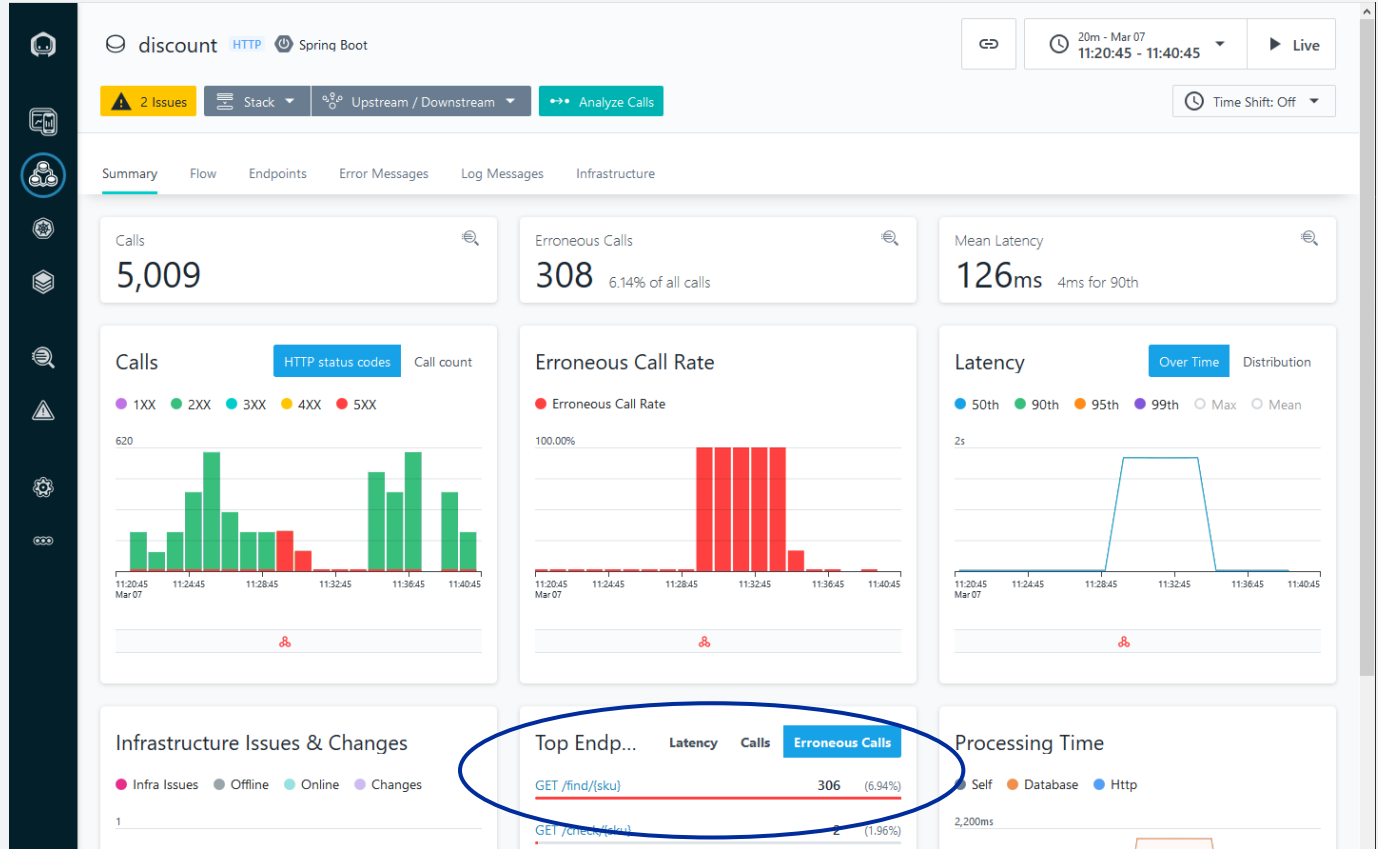


Service Details

To gather more information to confirm the issue can go into the details of the “discount” service.

Since the incident involved increase in erroneous calls, can see the top endpoints taking errors

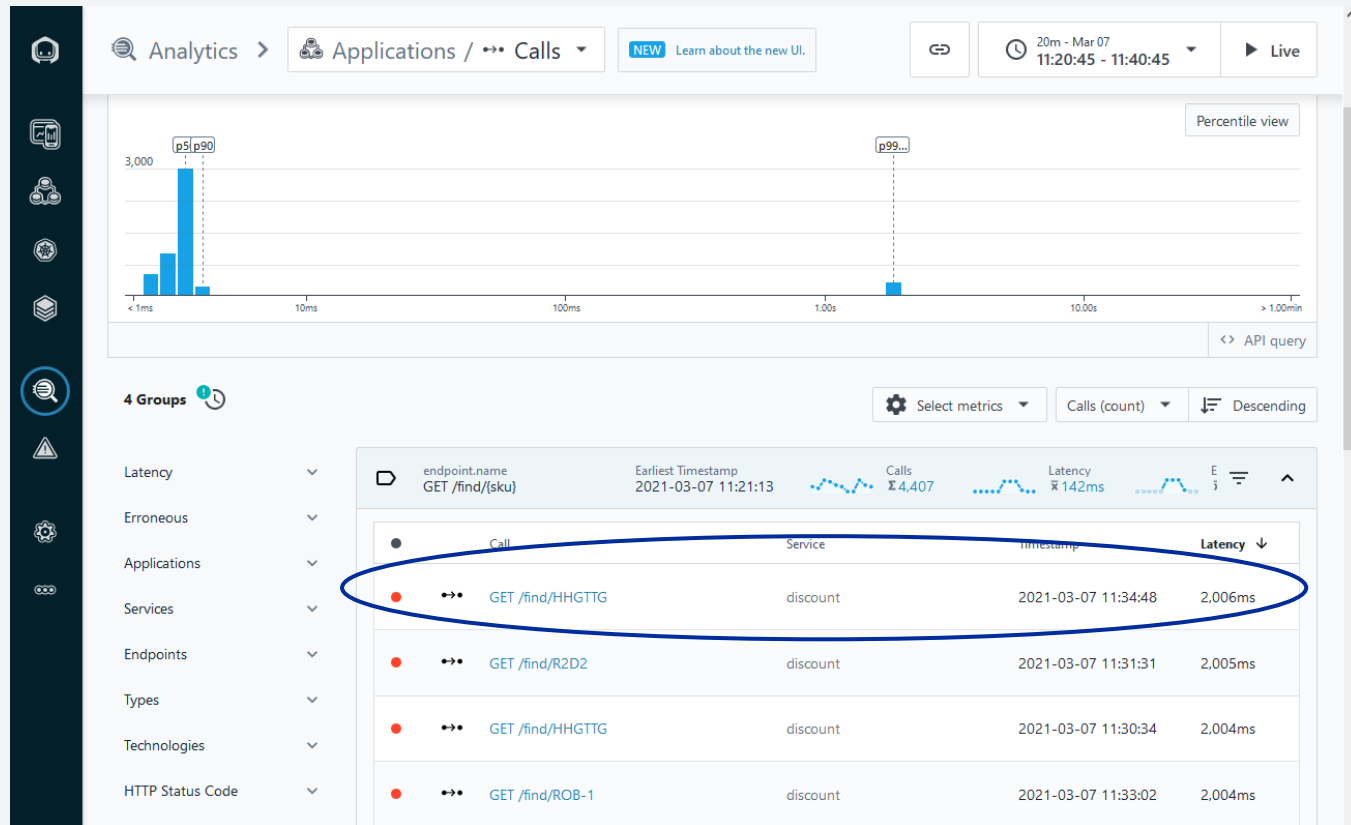
GET/find/{sku}



Analyze Calls

Shows the different APIs that make up the “discount” service, and groups together individual calls to a particular endpoint.

Here we see all the instances of calls to GET/find/{sku}



Drill Down on a Specific Request

Shows the end-to-end flow of the request showing the break-down of all the calls ending with the timeout of the call to the database.

The front end process nginx-web is calling the tpf-service, which is not the problem.

It's the backend service discount and it's call to the MYSQL database that is the source of the problem.

Analytics > Calls

20m - Mar 07 11:20:45 - 11:40:45 Live

GET /api/tpf-service/add/anonymous-198071/HHGTTG/1 Trace ID: 7425d5766aaB5bbe

Download

Service	Method	Duration	Count
discount	GET /fi	2,006ms	1
MySQL@3306 on demo-mysql	CONN	2,000ms	1

1 2

Calls

Colorize by Endpoint Technology

- GET /api/cart/add/an... HTTP 2.02s
 - To upstream cart of nginx-web 2,021ms
 - GET /add/anonymous-198071/HHGTTG HTTP 2,021ms
 - To GET /add/id/sku/qty of tpf-service 2,019ms
 - GET /product/sku/198071 HTTP 2,019ms
 - To GET /product/sku of catalogue-demo
 - com.mysql.jdbc.PreparedStatement.executeUpdate() VENDOR DATABAS 1ms
 - To catalogue.products of catalogue
 - GET /find/sku of discount 2,006ms
 - com.mysql.jdbc.PreparedStatement.executeUpdate() VENDOR DATABAS 2,000ms
 - To CONNECT of MySQL@3306 on demo-mysql

Details & Stack Trace

Type HTTP Call

Category http

Host discount-svc:8080

Request Path /find/HHGTTG

URL http://discount-svc:8080/find/HHGTTG

Method GET

Status Code 500 - Internal Server Error

Error

Content Length 0

Stack Trace

```
execute in org.springframework.http.client.AbstractClientHttpRequest
doExecute in org.springframework.web.client.RestTemplate:734
execute in org.springframework.web.client.RestTemplate:659
exchange in org.springframework.web.client.RestTemplate:578
getDiscount in com.instana.demo2.catalogue.service.Integrator
getProductBySku in com.instana.demo2.catalogue.CatalogueClient
doInvoke in org.springframework.web.method.support.Invocable
invokeForRequest in org.springframework.web.method.support
invokeAndHandle in org.springframework.web.servlet.mvc.method
invokeHandlerMethod in org.springframework.web.servlet.mvc.method
handleInternal in org.springframework.web.servlet.mvc.method
handle in org.springframework.web.servlet.mvc.method
doDispatch in org.springframework.web.servlet.DispatcherServlet
doService in org.springframework.web.servlet.DispatcherServlet
processRequest in org.springframework.web.servlet.FrameworkServlet
```

Value

The next level of tooling is about providing insights, not just collecting data.

- Built-in correlation between services and infrastructure to give context to issues
- Alerts based on rich set of rules including dynamic baselines and anomaly detection
- AI engine to group related events to quickly get to root cause

z/TPF is no longer a black box or black hole!



Why Instana?



APM Experts

Ranking the Observability Offerings

Capability	Splunk	Elastic	AppDynamics	Dynatrace	New Relic	Instana	Datadog	Honeycomb	Lightstep
Comprehensive Log Collection	●	●	○	◐	◐	◐	◐	◐	○
Comprehensive Metric Collection	◐	◐	◐	◐	◐	●	◐	◐	◐
Comprehensive Tracing Collection	◐	◐	◐	●	●	●	●	◐	●
Comprehensive Dependency Collection	○	○	◐	◐	◐	●	◐	○	○
Comprehensive Relating of Logs, Metrics, and Dependencies	○	○	◐	◐	◐	●	◐	○	○
Automated and Instant Instrumentation	○	○	○	●	○	●	○	○	○
High Cardinality Analytics	◐	◐	○	◐	◐	◐	◐	●	●
Dependency Map and AI Based Root Cause	◐	◐	◐	◐	◐	●	◐	○	○
Rank	5 th	5 th	8 th	2 nd	3 rd	1 st	4 th	5 th	8 th

● – Strongly Differentiated Capability
 ◐ – Differentiated Capability

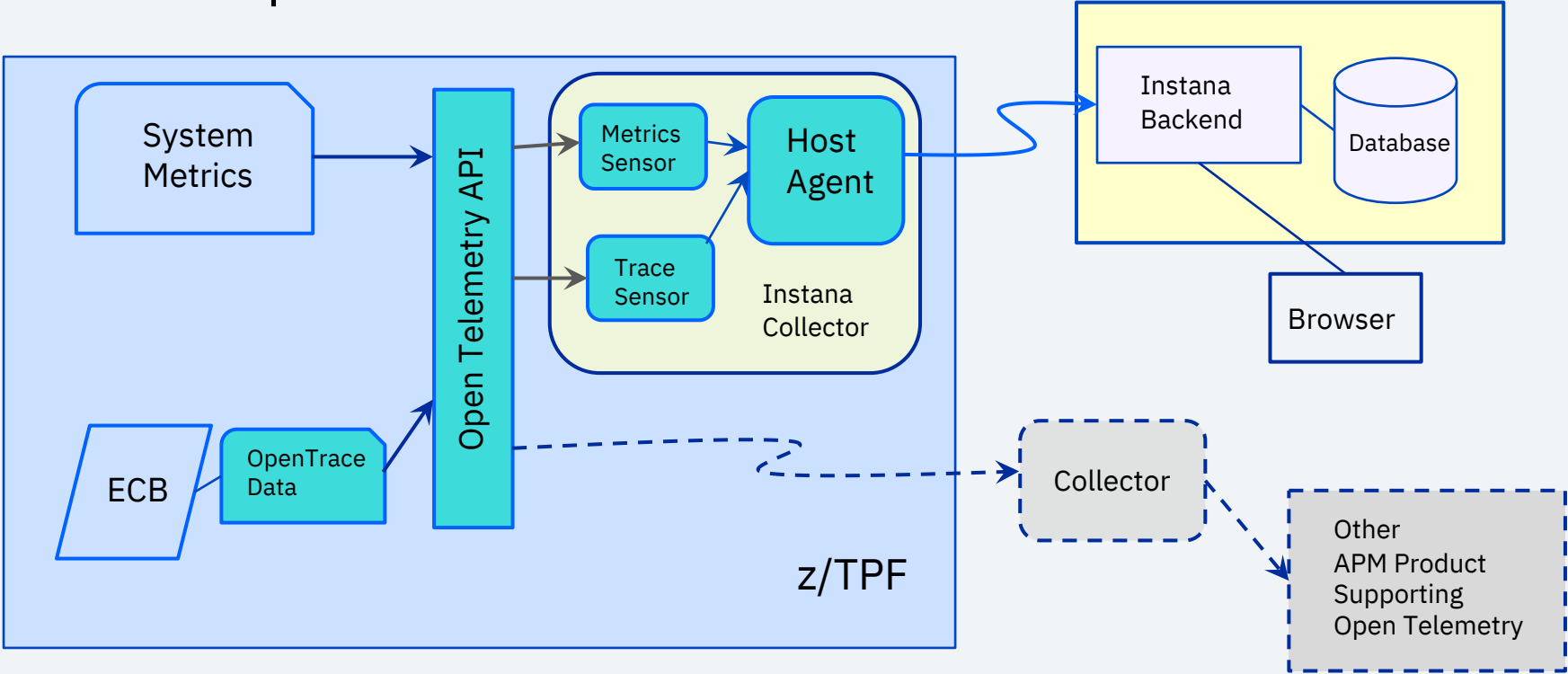
◐ – Average Value
 ◐ – Less Value than Normal

○ – Feature or Benefit not Present

Value Statement

A coverage programmer can view all the different components of an end-to-end transaction on a single dashboard, such that the component causing the problem can be identified in a matter of minutes.

Possible Implementation



Note: Overhead to z/TPF is expected to be minimal and will support sampling mode. Collection process would be TE eligible

Call for Sponsor Users

Will be looking for Sponsor Users to assist in design and implementation, targeting the following personas:

- Application Architects
- Enterprise operators and coverage
- z/TPF developers

If you would like to be involved, contact:

Colette Manoni (cmast@us.ibm.com) or Dan Gritter (dgritter@us.ibm.com)

Thank you !

© Copyright IBM Corporation 2020. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and ibm.com are trademarks of IBM Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at [Copyright and trademark information](#).

