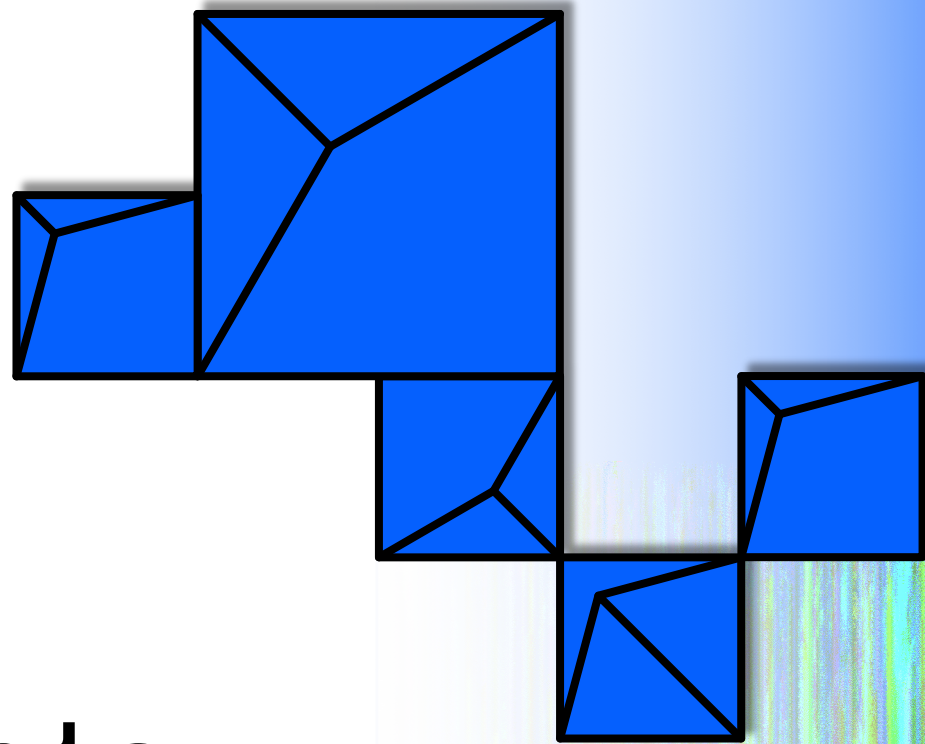


Java Enhancements

JT Plotzke



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.

Agenda

Recent Deliverables

JRE quarterly refreshes

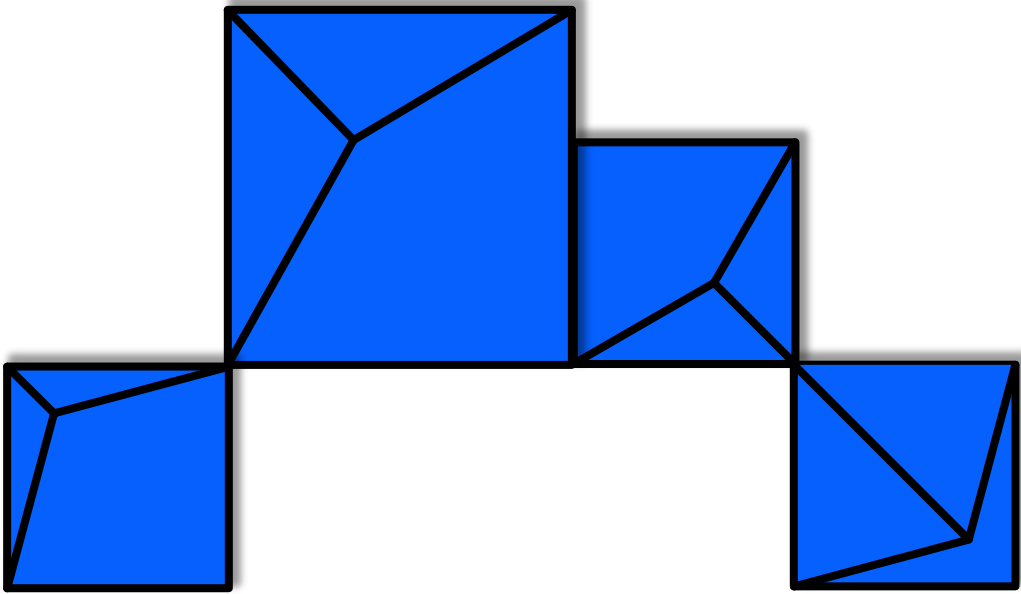
Guaranteed delivery model for Java

Future Deliverables

System-wide JVM monitoring

JRE performance

OpenJDK migration to Java 11



Recent Deliverables

Background

Recent Deliverables

- 2Q 2019 Java refresh (PJ45753 – Jun 2019)
- 3Q 2019 Java refresh (PJ45880 – Oct 2019)
- 4Q 2019 Java refresh (PJ45880* – Dec 2019)
- Guaranteed delivery model for Java (PJ45923 – Mar 2020)
- 1Q 2020 Java refresh (PJ45880* – Mar 2020)
- 2Q 2020 Java refresh (PJ46161 – Jul 2020)

Background

Java runtime environment (JRE) quarterly refreshes

PJ45753 – delivered Jun 2019

PJ45880 – delivered Oct 2019

PJ45880* – refresh replaced Dec 2019, Mar 2020

PJ46161 – delivered Jul 2020

- Stay current and pick up important security fixes from Oracle
- Does not always require a TPF APAR
 - Refreshes with no new TPF prerequisites replace prior refreshes with same prerequisites
 - Automatically pulled in by APAR download tools
- <https://www.ibm.com/support/pages/ibm-64-bit-runtime-environment-ztpf-java-technology-edition-version-8>

Background

JRE quarterly refreshes – TPF updates

PJ45753 – Jun 2019

- GCC 7 support (PJ45408) delivered Mar 2019
- Java OCO libraries now statically link in a GCC 7-built CPP1
- Decouples JRE from requiring rest of z/TPF to be at a GCC 7 level

PJ45880 – Oct 2019

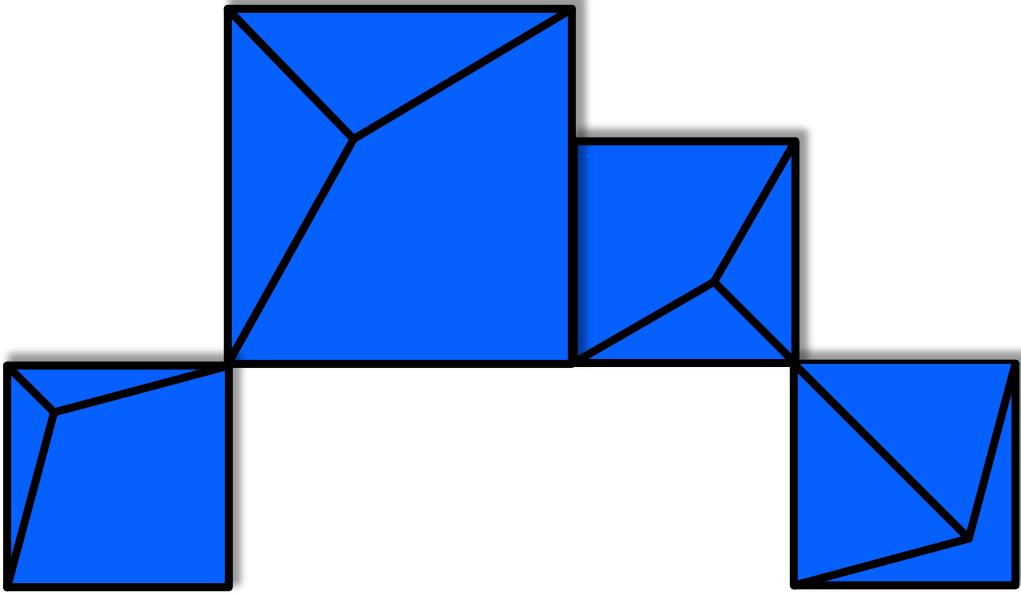
- zlib support for z15 hardware compression (PJ45872) delivered Oct 2019
- Java now incorporates the system version of zlib
- Automatically utilizes hardware compression on a z15

Background

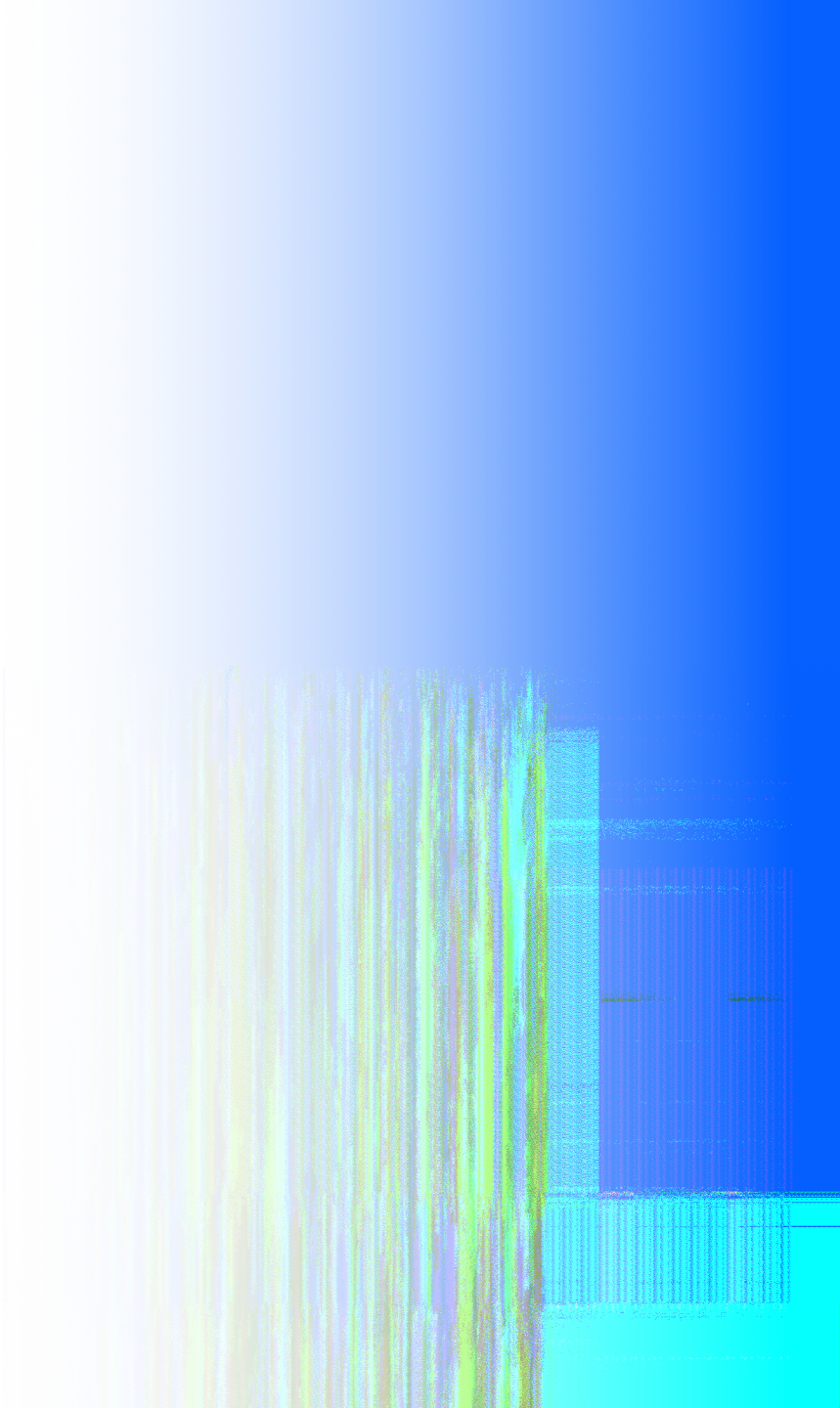
Guaranteed delivery model for Java

PJ45923 – delivered Mar 2020

- Allows a TPF application to submit work to a JVM
- Ensures delivery of TPF data with an “at least once” guarantee
- Supports standard (e.g. Kafka) and customizable solutions (e.g. TPF mail)
- More details to follow



Future Deliverables



What's next?

Future Deliverables

- System-wide JVM monitoring
- JRE performance
- OpenJDK migration to Java 11

What's next?

System-wide JVM monitoring

- Pain point: Existing Health Center support monitors a single JVM at a time
 - Must abort current connection before connecting to a different JVM
- Push monitoring data off z/TPF via real-time RTMC
- No performance impact to z/TPF system
- Comparison of JVMs within the same Application Manager for Java (JAM)

What's next?

As-is: IBM Health Center for Java

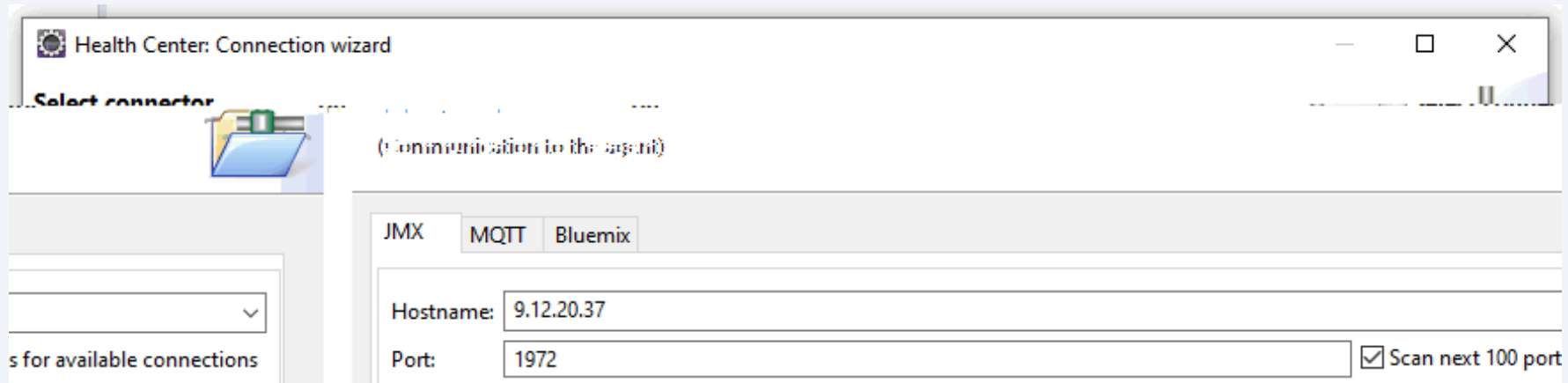
PJ45580 – delivered Mar 2019

- A diagnostic tool for monitoring the status of a Java application
- Can identify performance, bottleneck, and garbage collection issues before going into production
- Can be used in production with no impact to performance
- Separate, free Eclipse plugin required

What's next?

As-is: IBM Health Center – JVM Configuration

- Attach to JVMs configured with -Xhealthcenter option
 - For 1 JVM: add to command line at startup
 - For all JVMs: add to options.default or IBM_JAVA_OPTIONS env variable
 - For JAMs: add to <tns:OtherCommandLineOptions> section of the jam.xml config file
 - For already-running JVMs: use late attach feature
 - `zfile java -jar /sys/tpf_pbfiles/.../healthcenter.jar ID=<PID>`



What's next?

As-is: IBM Health Center – Late Attach Method

```
AAES0008I 00 ==> ZFILE java -jar /sys/tpf_pbf_files/opt/ibm/java-s390x-80/jre/lib/ext/healthcenter.jar
```

```
CSMP0097I 07.36.27 CPU-B SS-BSS SSU-HPN IS-01
```

```
FILE0001I 07.37.27 START OF DISPLAY FROM java -jar
```

A Health Center agent may be attached to one of the following Java Virtual Machines:

1:

```
1073744810: ID=1073744810
```

2:

```
com.sun.javatest.agent.AgentMain -active -activeHost 9.56.224.21 -activePort 1913:  
ID=1073747724
```

3:

```
/sys/tpf_pbf_files/opt/ibm/java-s390x-80/jre/lib/ext/healthcenter.jar: ID=1073879174
```

Please select the VM (enter number between 1 and 3) in which to enable the Health Center agent, or blank line to exit.

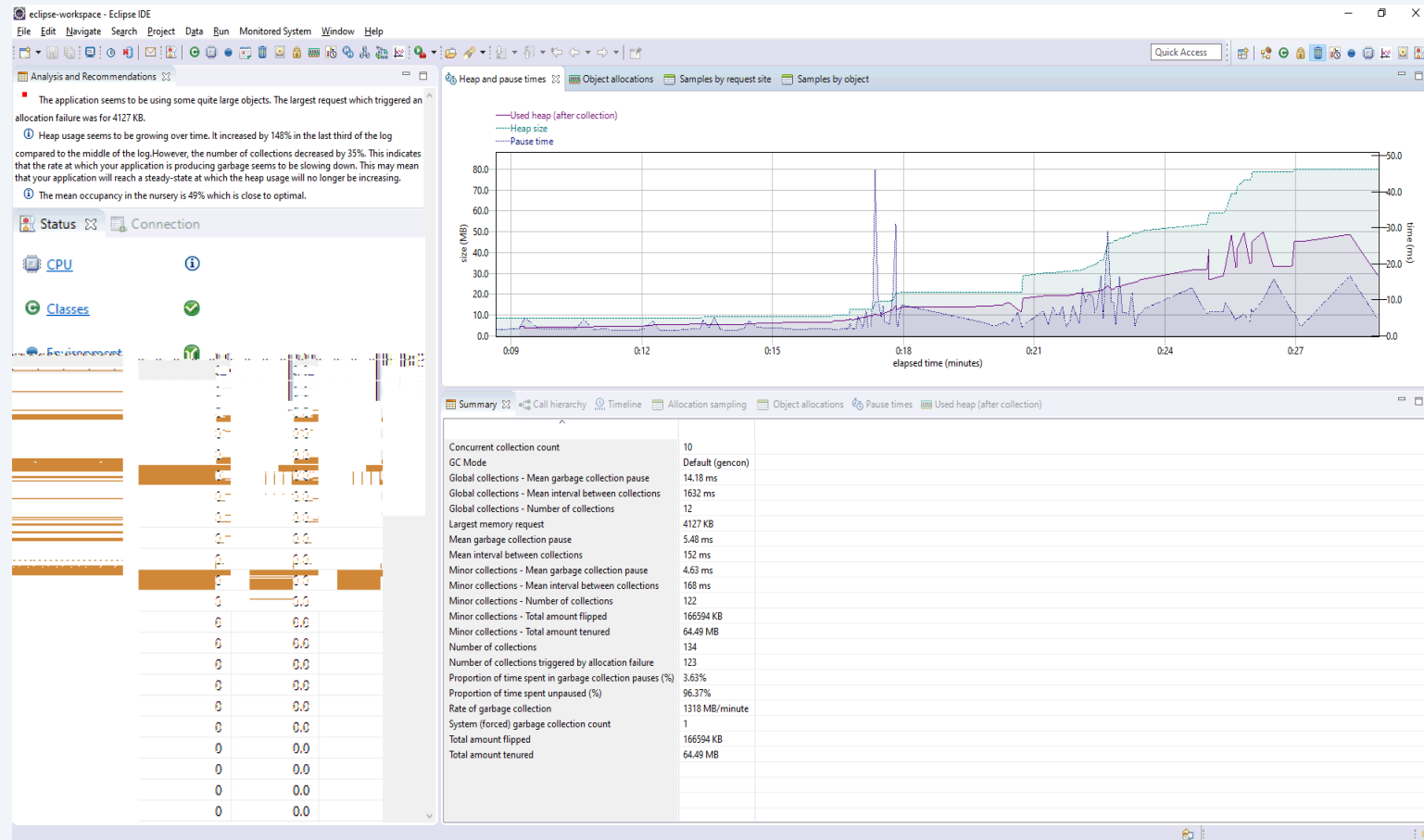
Empty VM number entered, so exiting.

```
END OF DISPLAY+
```

What's next?

As-is: IBM Health Center – Sample GUI

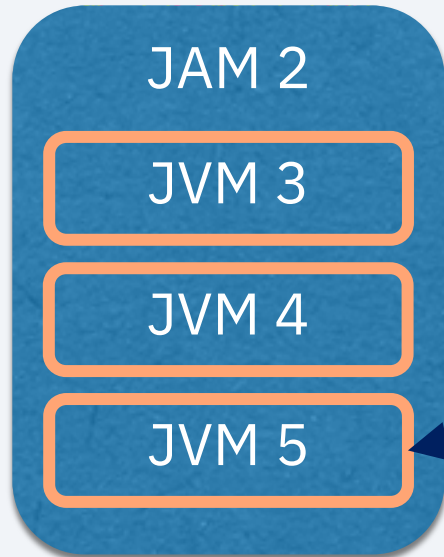
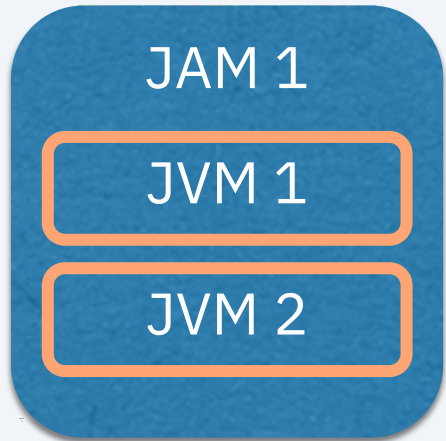
- Various JVM statistics can be monitored in real-time
 - CPU usage (total overall, per Java class)
 - Environment settings (classpath, runtime config options, env variables)
 - Garbage collection (heap usage, pause time)



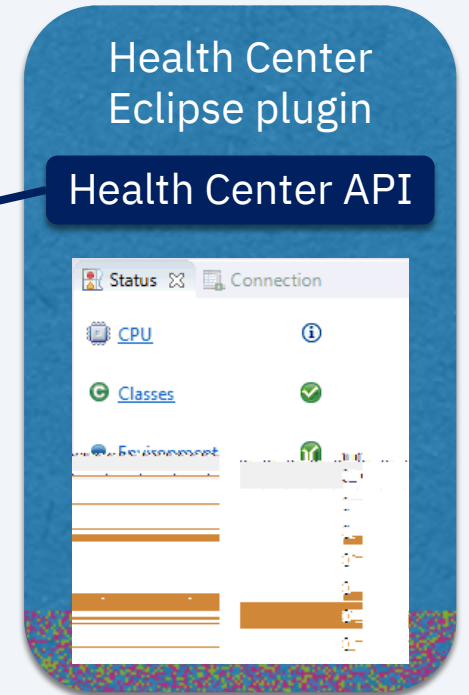
What's next?

As-is: z/TPF

Remote System



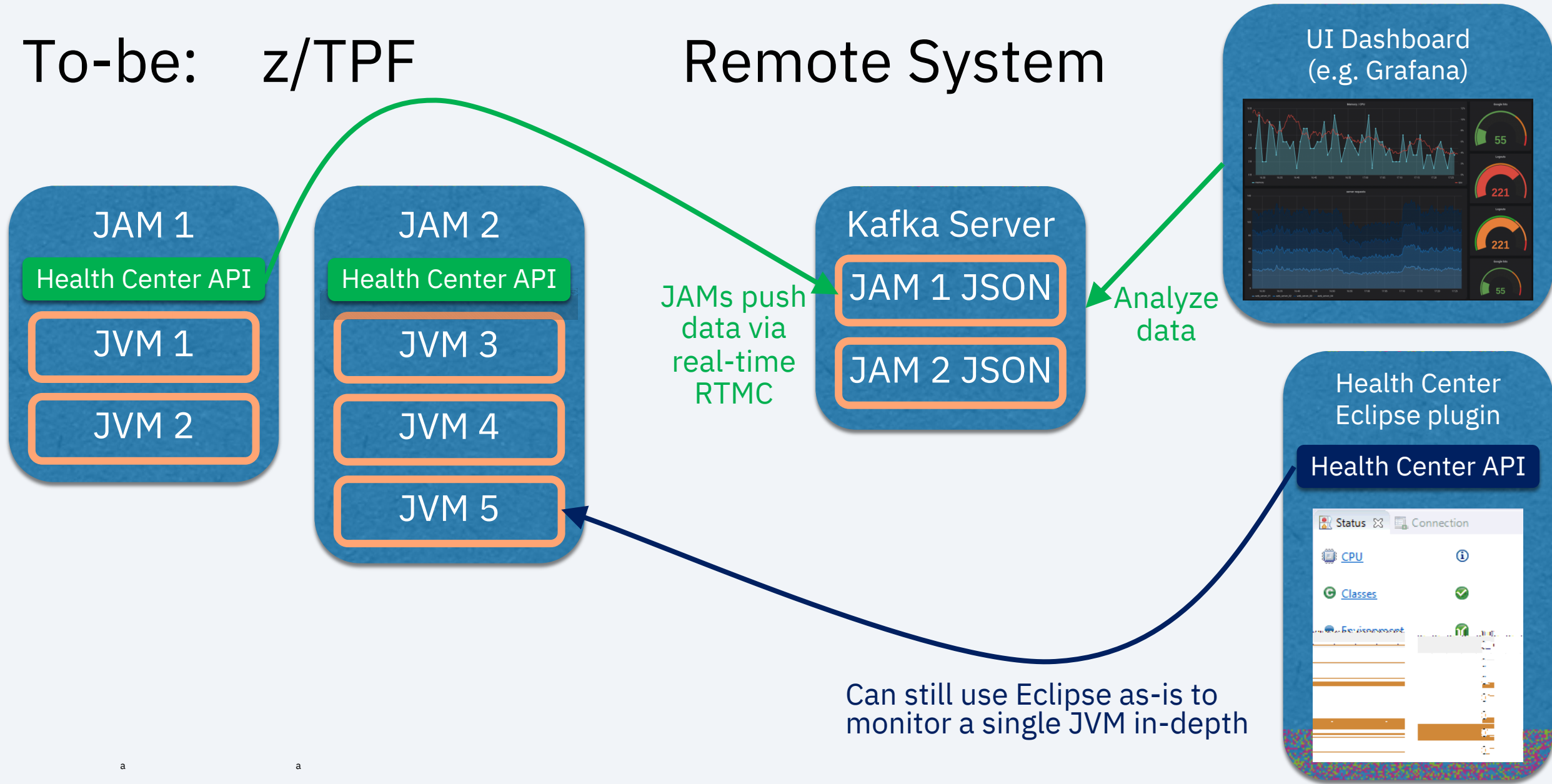
Establishes connection to a single JVM
Must then abort current connection
before connecting to a
different JVM



What's next?

To-be: z/TPF

Remote System



What's next?

System-wide JVM monitoring

- Enhance existing JAM support
 - Incorporate Health Center APIs as a listening application
 - Push monitoring data off z/TPF via real-time RTMC
 - Up-to-date information (< 1 sec refresh rate, as necessary)
- Analyze data on a remote system
 - No performance impact to z/TPF system
 - Allows for highly customizable, visual dashboard representation
 - Comparison of JVMs within the same JAM
- Continue to use existing Health Center support
 - Eclipse plugin attaches to a single JVM
- Proposal:
 - End of 2020: Release support for system-wide JVM monitoring
 - Sample GUI dashboard to be available for download

What's next?

JRE performance

- Pain point: Java startup time on TPF is measurably slower than on Linux.
- Ongoing effort to improve JVM startup time
 - No specific deliverable, incorporated into each quarterly refresh
- Optimize JAM recovery time in case of an outage
- Incorporate pause-less garbage collection into JRE

What's next?

As-is: JAM startup and recovery time

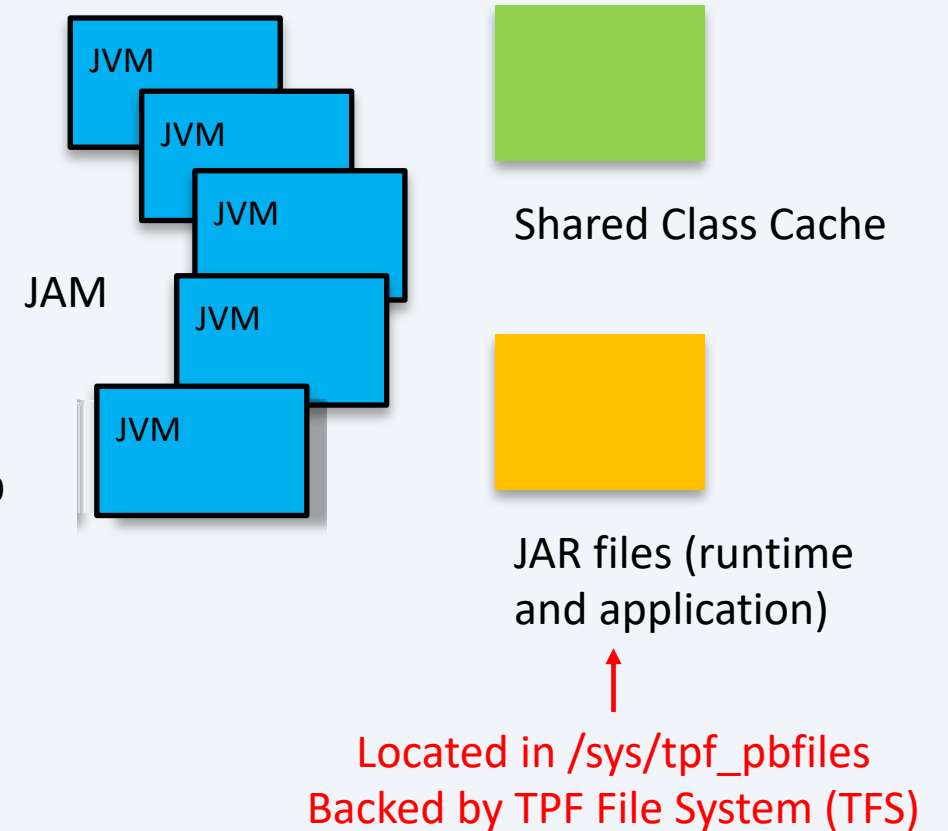
Before IPL

```
FILE0001I 07.37.27 START OF DISPLAY FROM java -jar
zfile java -Xshareclasses:listAllCaches
Listing all caches in cacheDir /tmp/java/sharedresources/
```

Cache name	level	cache-type	last detach time
MyCache	Java8 64-bit	non-persistent	Wed Feb 26 11:37:47 2020

END OF DISPLAY+

Today z/TPF only supports
non-persistent caches



What's next?

As-is: JAM startup and recovery time

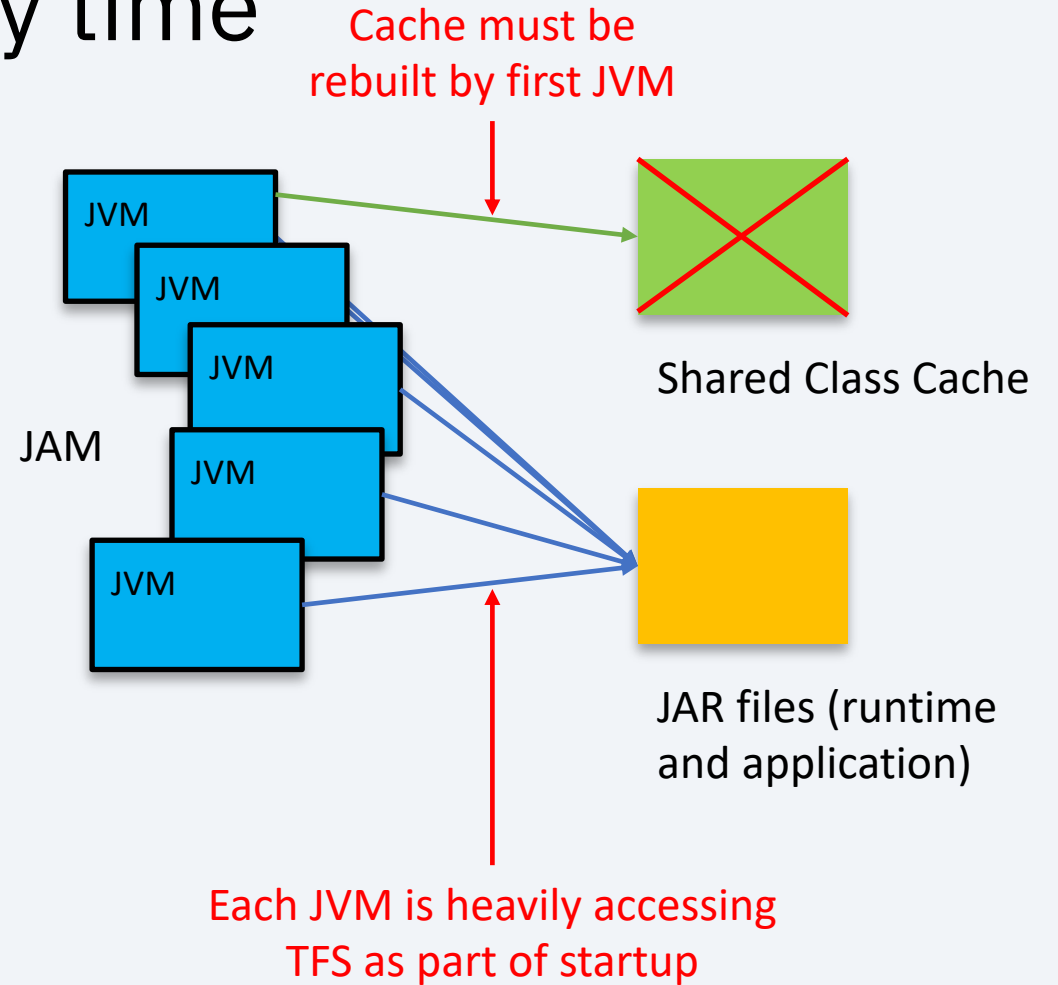
After IPL

JAM startup time with 5 JVMs (8 I/S, z14): **55 seconds**

```
FILE0001I 07.37.27 START OF DISPLAY FROM java -jar  
zfile java -Xshareclasses:listAllCaches  
JVMSHRC005I No shared class caches available
```

END OF DISPLAY+

z/TPF reports no caches exist
after an IPL



What's next?

To-be: JAM startup and recovery time

After IPL

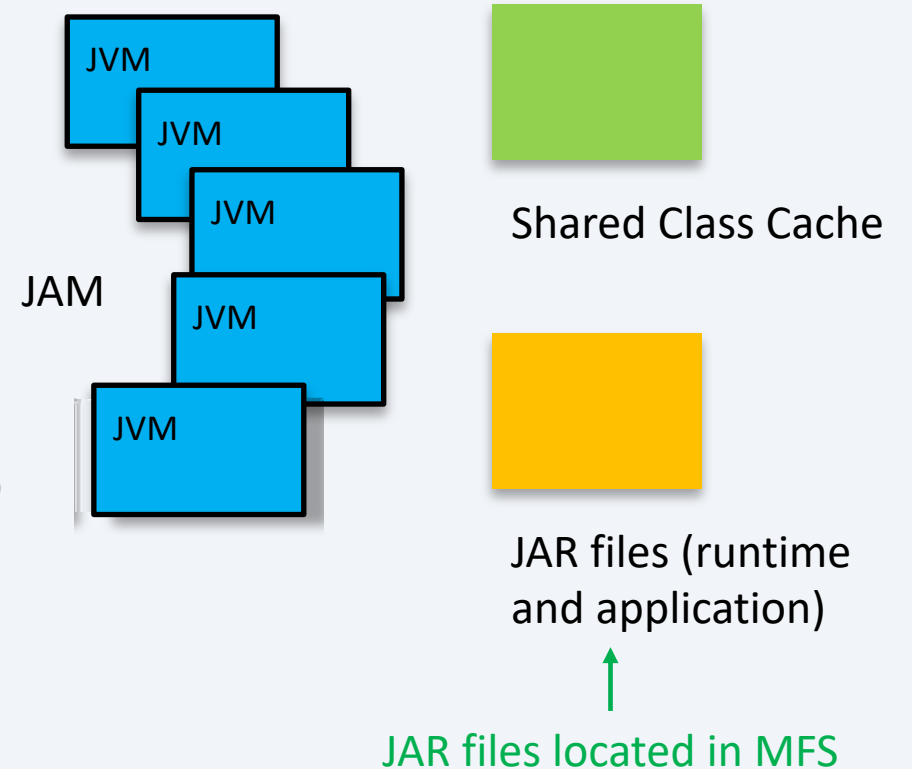
JAM startup time with 5 JVMs (8 I/S, z14): < 10 seconds

```
FILE0001I 07.37.27 START OF DISPLAY FROM java -jar
zfile java -Xshareclasses:listAllCaches
Listing all caches in cacheDir /tmp/java/sharedresources/
```

Cache name	level	cache-type	last detach time
MyCache	Java8 64-bit	recoverable	Wed Feb 26 11:37:47 2020

END OF DISPLAY+

New shared class cache type
ready for use across IPL



What's next?

Pause-less garbage collection

- Garbage collection (GC) policy specified `-Xgcpolicy=...` parameter
 - Default: `-Xgcpolicy=gencon` (generational concurrency)
- Response times for different GC events vary widely depending on GC pausing currently on z/TPF
- *gencon* uses a unique GC event named “scavenge” which accounts for a significant amount of total GC pause time
- Requires z13 or later hardware

What's next?

Pause-less garbage collection

DayTrader Benchmark on z/TPF, without (as-is) and with (to-be) pause-less



What's next?

OpenJDK migration to Java 11

- Java 8 released 2015
 - Still supported, end of service date unknown but potentially coming
- z/TPF support for Java (PJ43892) delivered Jan 2017
 - Based on IBM Java SDK 8.0
 - Proprietary IBM Java Virtual Machine (JVM)
 - Shared Java Class Libraries (JCL) with Oracle
- Beginning with Java 9, IBM's JVM (OpenJ9) is now open sourced
- Java 11 released end of 2018
 - Long-term service release (guaranteed 5 years of support)

What's next?

OpenJDK migration to Java 11

Features added since Java 8

OpenJDK 9

- Java Platform Modular System (Jigsaw Project)
- XML Catalogs
- Concurrency Updates to support Reactive Streams (new Flow Class)
- Compact Strings

OpenJDK 10

- Local-variable type inference
- Time-based release versioning
- Additional Unicode language-tag extensions
- Root certificates
- Thread-local handshakes
- Heap allocation on alternative memory devices
- Remove the native-header generation tool – javah

OpenJDK 11 (Long Term Service)

- Dynamic class-file constants
- Epsilon: a no-op garbage collector
- Local-variable syntax for lambda parameters
- Low-overhead heap profiling
- HTTP client (standard)
- Transport Layer Security (TLS) 1.3
- Flight recorder
- JavaFX, Java EE and CORBA modules have been removed from JDK
- Deprecated the Nashorn JavaScript engine
- Unicode 10.0.0 support (while current version is Unicode 11.0.0, it's only in Java 12)
- A number of features from previous releases were dropped; in particular, Java applets and Java Web Start are no longer available.

OpenJ9

← Same version can be compatible with
Multiple versions of OpenJDK

- JITaaService
- Shared Class Cache Enhancements (e.g, layering caches, optimizing entry validation, adding hints from previous runs to the cache)
- Container Aware related updates

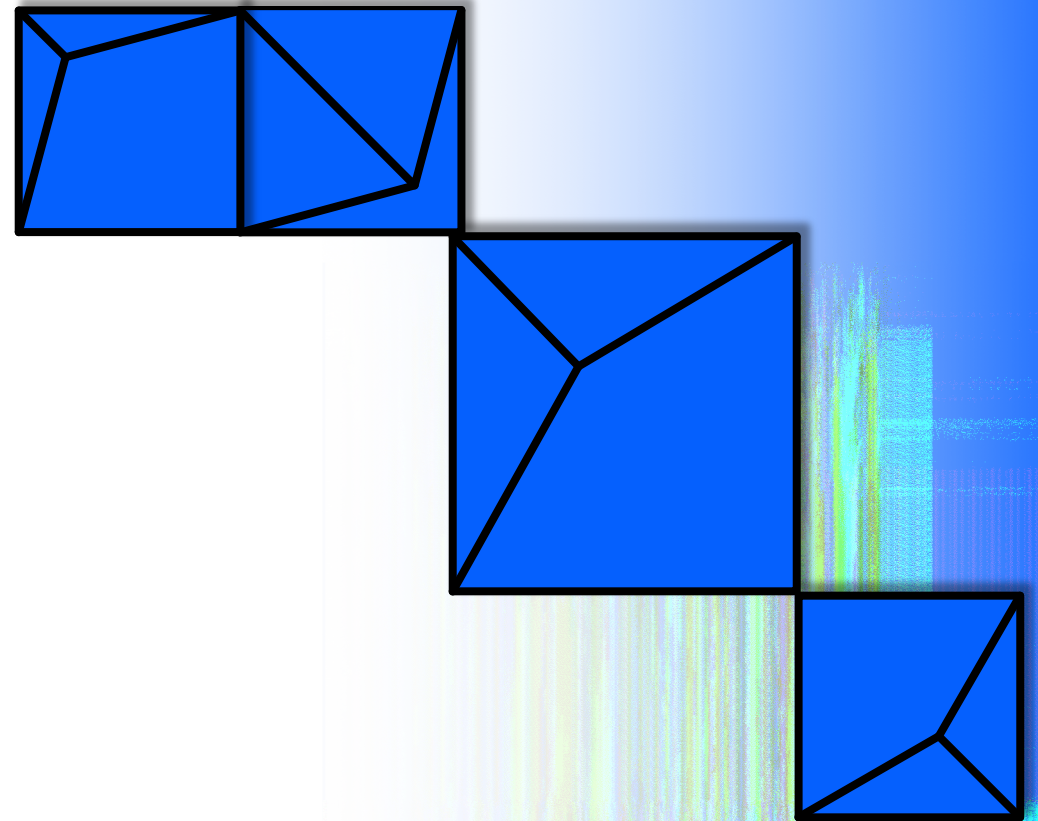
What's next?

OpenJDK 11 migration considerations

- Moving to OpenJDK 11 may cause backward compatibility issues
 - Need to replace Java EE modules with third-party versions
 - Increased security closes access to internal APIs
 - Some previously deprecated methods are removed
- May result in new compilation errors in existing Java applications
- Factor in OpenJDK 11 compatibility as you adopt new packages
- Proposal:
 - End of 2020: Release support for OpenJDK 11, announce deprecation of support for Java 8
 - End of 2021: Discontinue support for Java 8

Thank You

Questions? Comments?



Virtual TPFUG Q&A

Summary of Q&A from the virtual TPFUG event:

Question	Answer
Q: Any TPF resource considerations for the implementation of PJ45923, such as memory or file system?	A: There are no additional resource considerations (over and above the existing CTKA/JVM requirements)
Q: Can we log TPF REST/SOAP request/responses using Log4j2?	A: Yes, this could be done as there are user exits in our z/TPF REST provider support. The user exits can be used to log the data. However it may not be an optimal solution
Q: With Apache support removed, does the lab have any plans to provide an education session or an example of how to build Tomcat for z/TPF and provide experiences on that?	A: The TPF Lab is still investigating alternatives to the existing Apache support. We can provide some sessions on what's next if there is sufficient interest.



Trademarks

The logo, IBM, COM and Rational are trademarks or registered trademarks of International Business Machines Corporation registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. Current list of IBM trademarks is available on the web at [Copyright and trademark information](#) at www.ibm.com/legal/copytrade.shtml

Notes

Performance is in terms of internal throughput rate ratio based on measurements and projections using standard benchmarks in a controlled environment. The actual throughput that an user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the 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.

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 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 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.

This presentation and the claims outlined in it were reviewed for compliance with local law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.