

ORACLE®



Mission Possible - Near zero overhead profiling

Klara Ward
Principal Software Developer
Java Mission Control team, Oracle
February 6, 2018



Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

About the speaker



- Developer in the Java Mission Control team
- @klaraward

Program Agenda

- 1 Overview of Java Flight Recorder
- 2 Overview of Java Mission Control
- 3 Creating Recordings
- 4 Analyzing Recordings
- 5 Q&A

Production Time Profiling and Diagnostics

“The big challenge is no longer really performance. The big challenge is profiling, and especially profiling in production.”

- Tony Printezis, JVM engineer, Twitter
(Devoxx 2015, “Life of a Twitter JVM Engineer”, 49:49)

"Once more @javamissionctrl is saving my day! "

- *Michael Nitschinger, SDK Engineer, Couchbase*

"Java Mission Control is the best profiler by far."

- *T Jake Luciani, PMC Cassandra*

"JMC not only saves time trying to resolve performance issues and bugs, it can give you a detailed view on your application you cannot get with other commercial profilers"

- *Peter Lawrey, CEO, Chronicle Software*

"... Our real-time messaging products can publish millions of messages a second to many thousands of connections - only JMC can keep up with this level of load."

- *Phil Aston, Product Architect, Push Technology*

"For the record: Java Mission Control is the best profiler ever, I use it daily, and so should you"

- *Marcus Lagergren, Lead Architect, Nasdaq*

"I am ACS engineer since 2008, delivering local Middleware support to several customers. Since I started to work with Java/JRockit Mission Control, it became a key tool for my work, helping me to troubleshooting, identifying root causes and bottlenecks, and also for doing proactive follow up services to customers.

Without it, I would be blind."

- Iratxe Etxebarria, Oracle (ACS)

"In Fusion we create hundreds of thousands of Flight Recordings, and we can **figure out 95% of the issues using nothing but the recordings."**

- Joe Albowicz, Oracle (Fusion Application Development)

”Java Mission Control profiler”

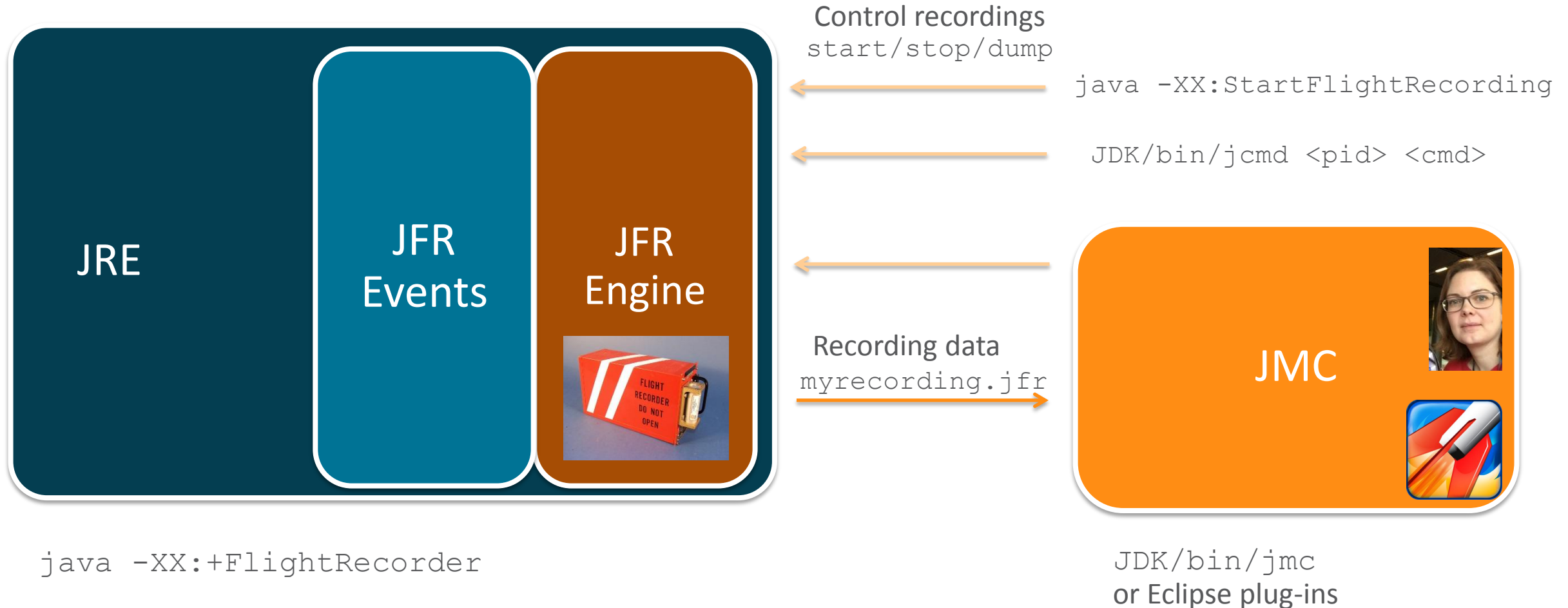
What do they mean?

Usually:

- Data from Java Flight Recorder
- Visualized in Java Mission Control



Java Flight Recorder (JFR) & Java Mission Control (JMC)



`java -XX:+FlightRecorder`

Overview of Java Flight Recorder

Low overhead profiling



Java Flight Recorder

- Event Recorder
- Build into the JVM
- Extremely low overhead
- Can keep it always on and dump when necessary
- Free for development use
 - Commercial use still requires a license
- Open source !

To be open sourced

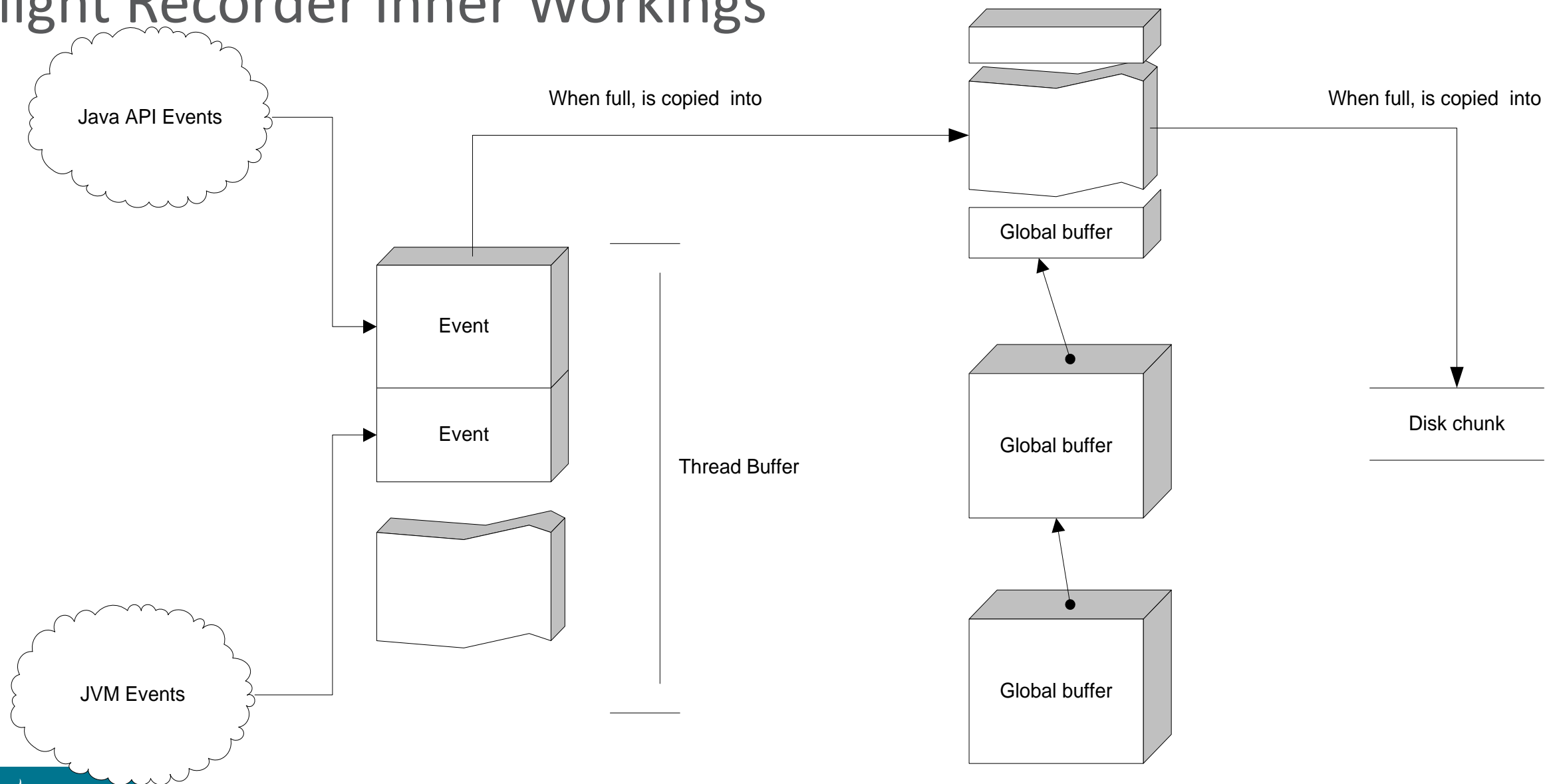


Low overhead

- 1-2 % depending on settings
 - Enabled events, thresholds, periods
- Thread local buffers
- Using JVM internal data structures
- Avoids recompiling code or disabling optimizations
- Sampled method and allocation profiling
- Fast timestamps – invariant TSC



Flight Recorder Inner Workings



Method sampling

- Sampling profiler
 - Periodic samples
 - Not sampling all threads every time
 - Does not require threads to be at safepoints*
 - (Flags currently needed to give more accurate non-safepoint data)
 - `-XX:+UnlockDiagnosticVMOptions -XX:+DebugNonSafepoints`
 - Not sampling threads in native
 - Will do in coming releases
- * More on safepoints - <http://psy-lob-saw.blogspot.co.uk/2016/02/why-most-sampling-java-profilers-are.html>



Profilers Are Lying Hobbitises

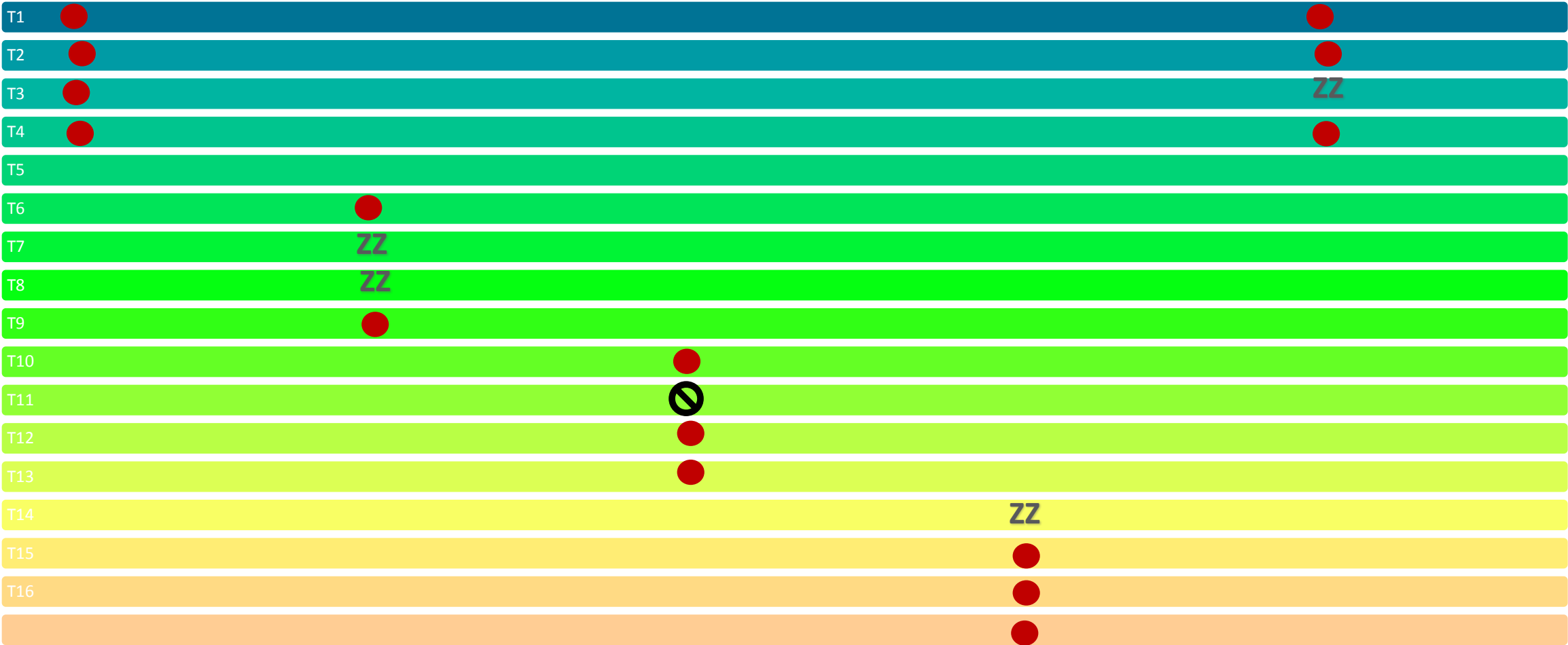


Filmed at
QCon San Francisco 2015

brought to you by
InfoQ

Nitsan Wakart (@nitsanw)
Lead Performance Engineer, Azul Systems

Method sampling



Allocation profiling

- Sampling profiler
- "TLAB" - Thread Local Allocation Buffer
 - Creates an event for each new TLAB
 - Creates an event for each object outside a TLAB

Allocation profiling



"Apple 150 g, total 2 kg"



"Banana 100 g, total 3 kg"



"Apple 5 kg"

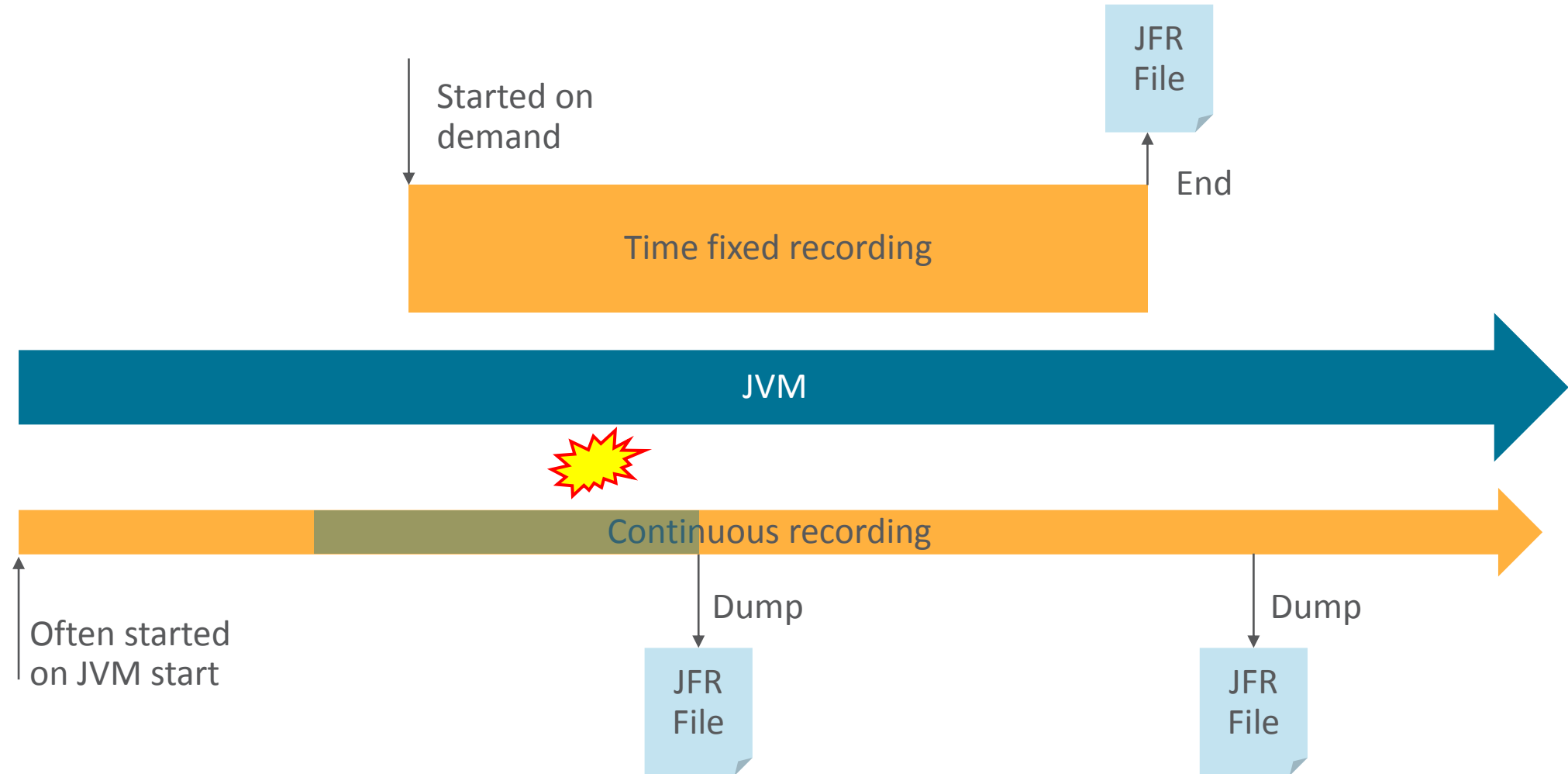
“Flight Recorder”

✓ Profiler

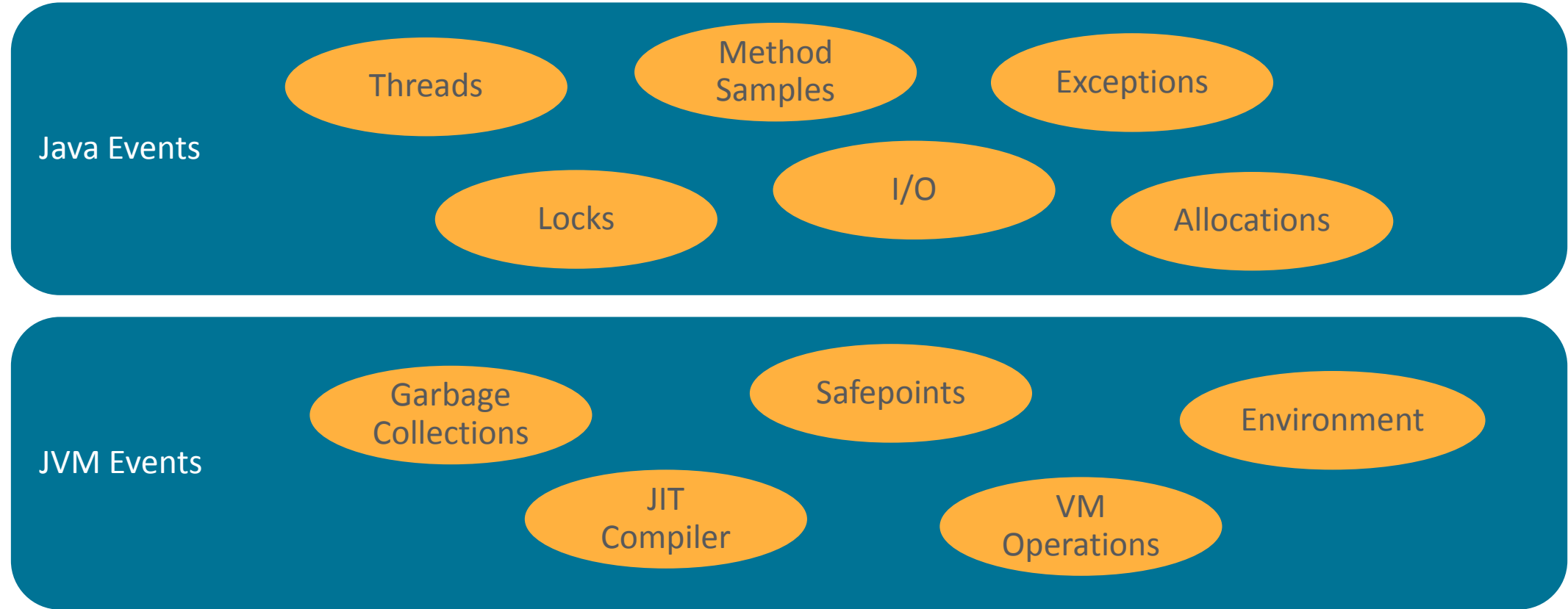
✓ Logger



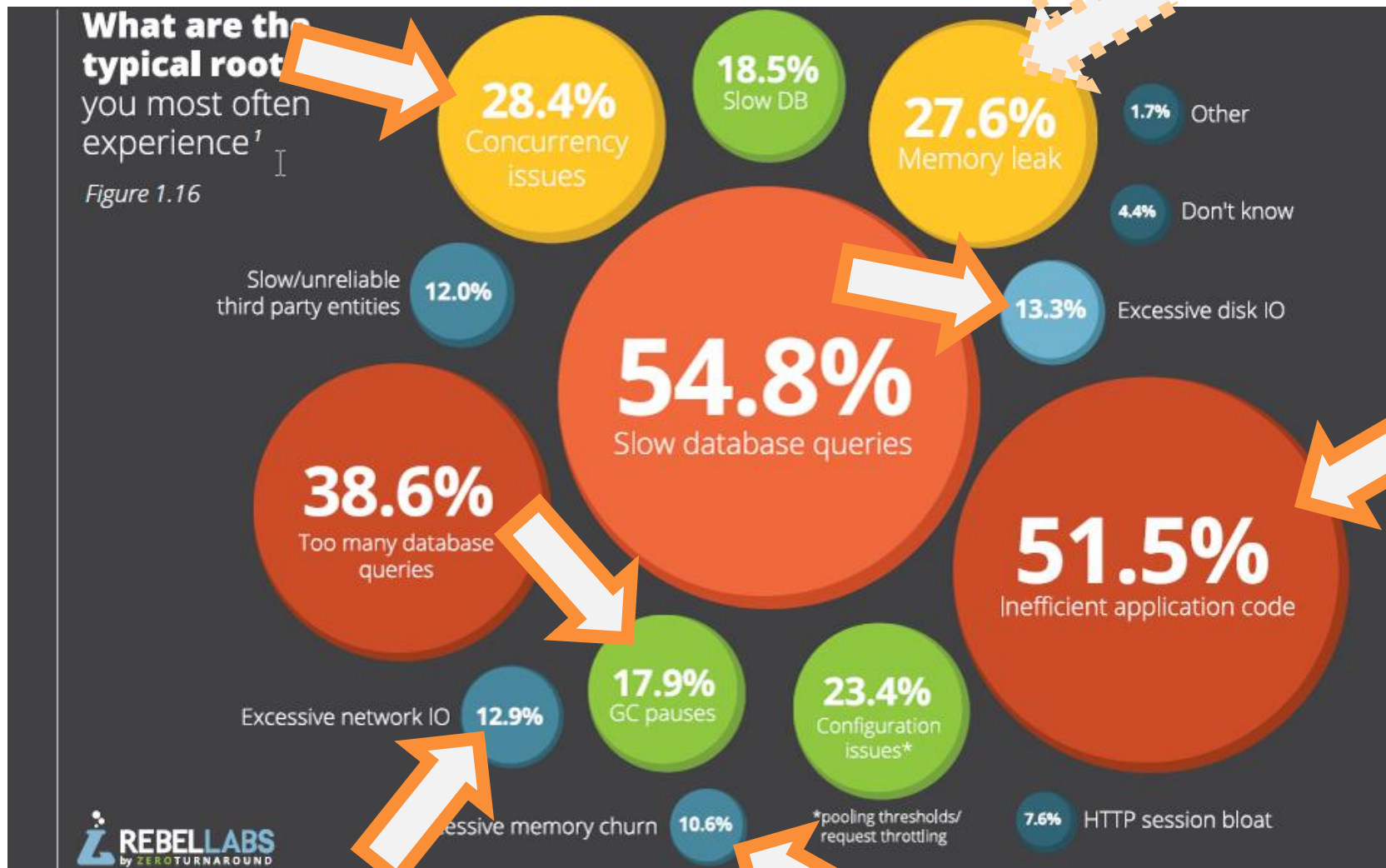
JFR Recording Types



Data Collected by JFR

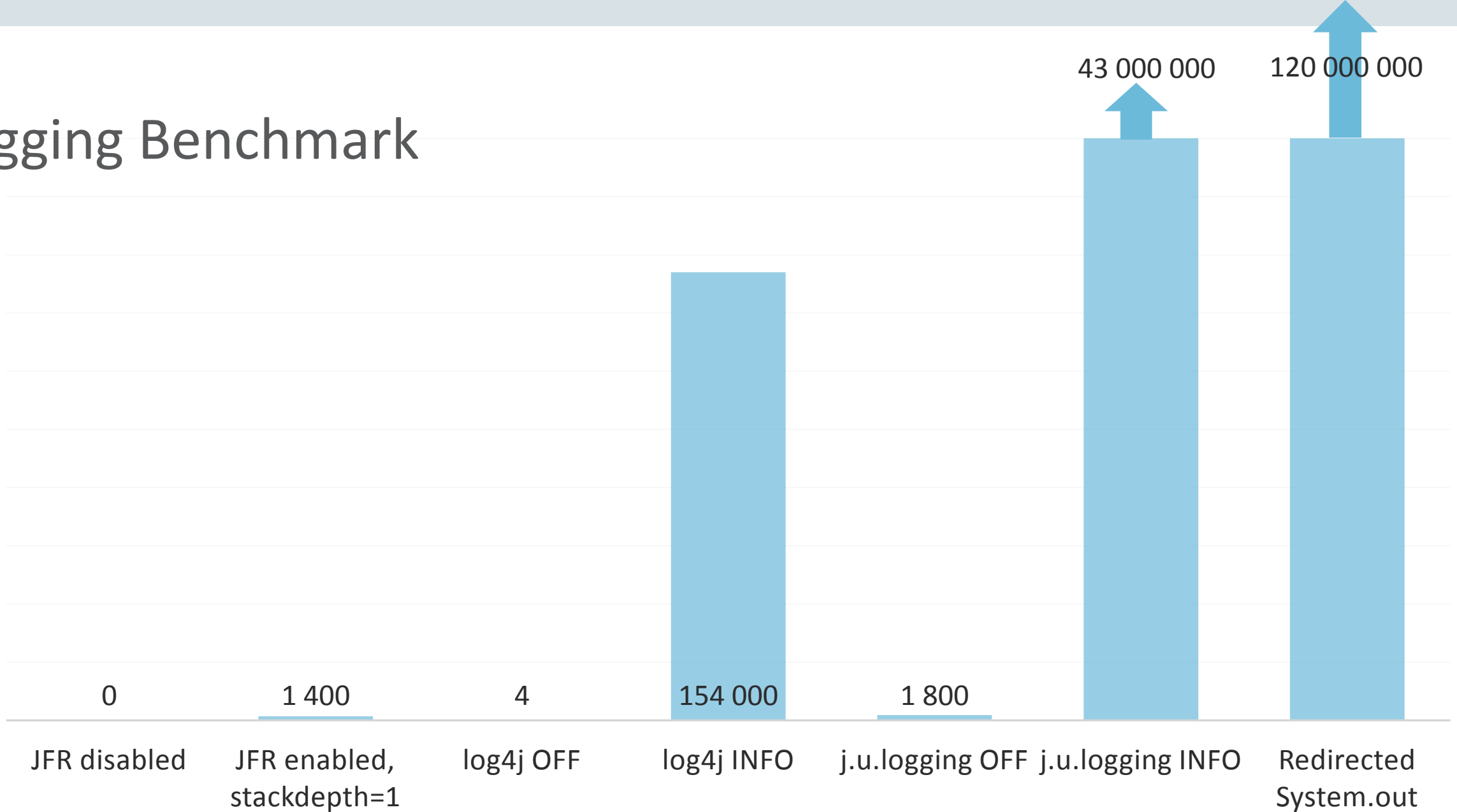


RebelLabs Developer Productivity Report 2015, Java Performance Survey



Logging Benchmark

NANOS / OPERATION



JMC, 8 THREADS

Java Flight Recorder in Java 9

- New supported APIs
 - Create your own events
 - Create and read recordings
 - Google “jdk.jfr api” for documentation
- Performance enhancements
- Improved ability to emit data in bad situations
 - Can dump on crashes and OOM
- More event types



Java Flight Recorder in Java 9

New API for custom events

```
import jdk.jfr.Event;
import jdk.jfr.Label;

public class Hello {

    @Label("Hello World")
    static class HelloWorldEvent extends Event {
        @Label("Message")
        String message;
    }

    public static void main(String... args) {
        HelloWorldEvent event = new HelloWorldEvent();
        event.message = "Hello World event message!";
        event.commit();
    }
}
```

Java Flight Recorder in Java 9

New API for starting recordings

```
import jdk.jfr.Recording;

public class Record {

    public static void main(String... args) {
        Recording recording = new Recording();
        recording.enable>HelloWorldEvent.class);
        recording.start();
        // Do stuff and commit events
        recording.stop();
        recording.dump(path);
    }
}
```

Java Flight Recorder in Java 9

New API for starting recordings

```
import jdk.jfr.consumer.RecordedEvent;
import jdk.jfr.consumer.RecordingFile;

public class ParseRecording {

    public static void main(String... args) {
        RecordingFile.readAllEvents(Path.get(args[0])).stream()
            .filter((RecordedEvent e) ->
                e.getEventType().getName().equals("HelloWorldEvent"))
            .map(e -> e.getValue("message"))
            .forEach(System.out::println);
    }
}
```

Overview of Java Mission Control

The graphical client



Java Mission Control

- A tool suite for monitoring JVM behavior
 - JMX Console
 - Real time monitoring
 - Flight Recorder UI
 - Visualization of Flight Recorder data
 - Completely reworked in JMC 6
 - Automatic Recording Analysis
 - Detects potential problems and provides recommendations on how to solve them
 - Supports Flight Recordings from JDK 7, 8 and 9

To be open sourced



JMC Installation and Startup

- Bundled with JDK

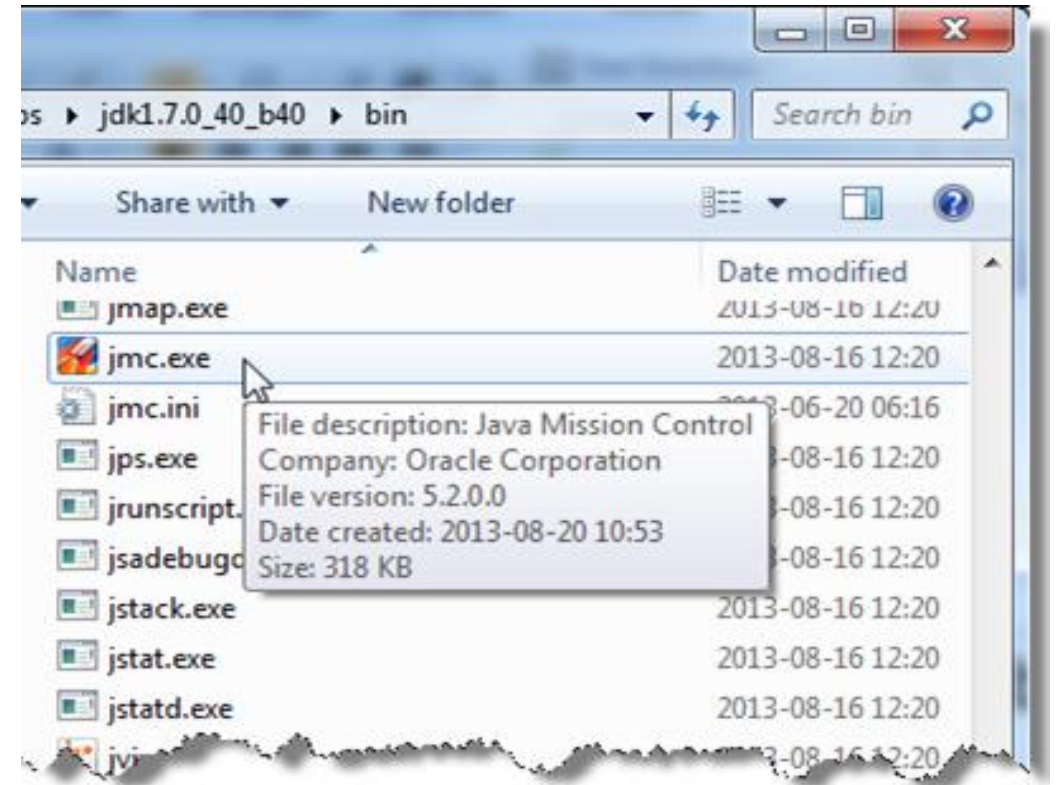
Windows/Linux: <JDK>/bin/jmc

Mac: (/usr/bin/) jmc

- Eclipse plugins

– Install from update site

<http://oracle.com/missioncontrol>, Eclipse Update Site



Creating Recordings

Enable Flight Recorder in the Java VM

- Start the JVM that you want to record from
 - XX:+UnlockCommercialFeatures
 - XX:+FlightRecorder

Enable Flight Recorder in the Java VM

- Start the JVM that you want to record from

-XX:+UnlockCommercialFeatures

-XX:+FlightRecorder

- Java 8u40 and later: Enable at runtime if needed
 - Using Java Mission Control
 - Using jcmd

Enable Flight Recorder in the Java VM

- Start the JVM that you want to record from

`-XX:+UnlockCommercialFeatures`

~~`-XX:+FlightRecorder`~~

- Java 9 and later

Enable Flight Recorder in the Java VM

- Start the JVM that you want to record from

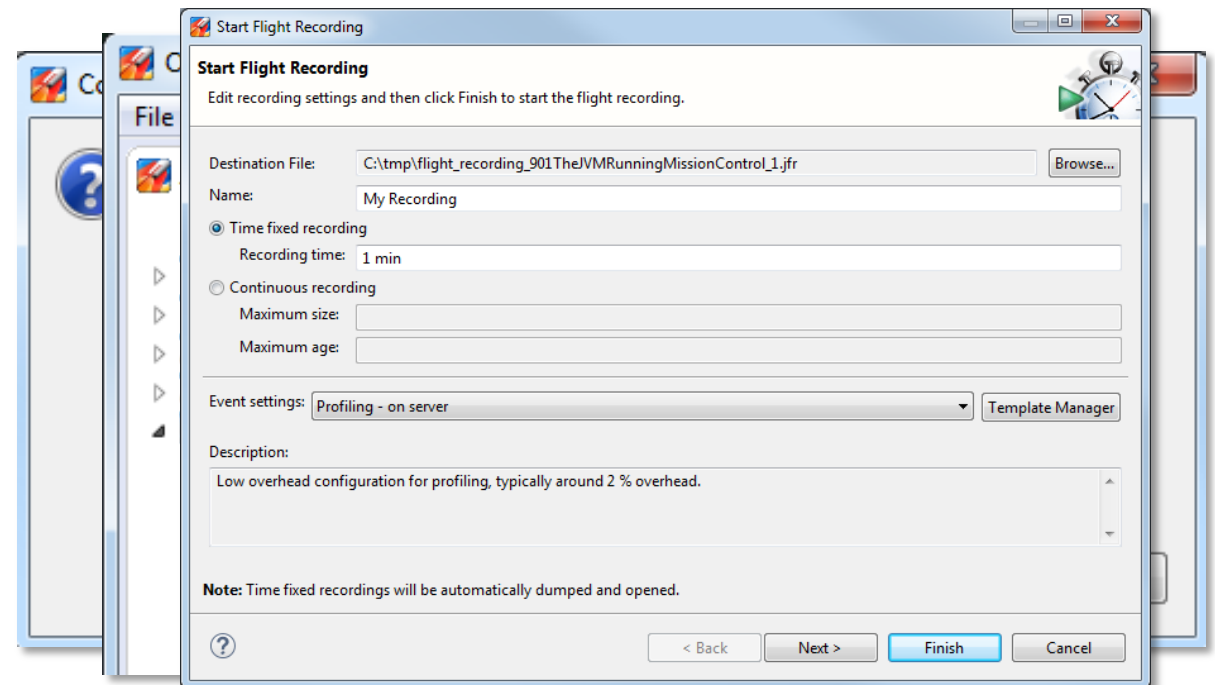
~~-XX:+UnlockCommercialFeatures~~

~~-XX:+FlightRecorder~~

- Future version of Java

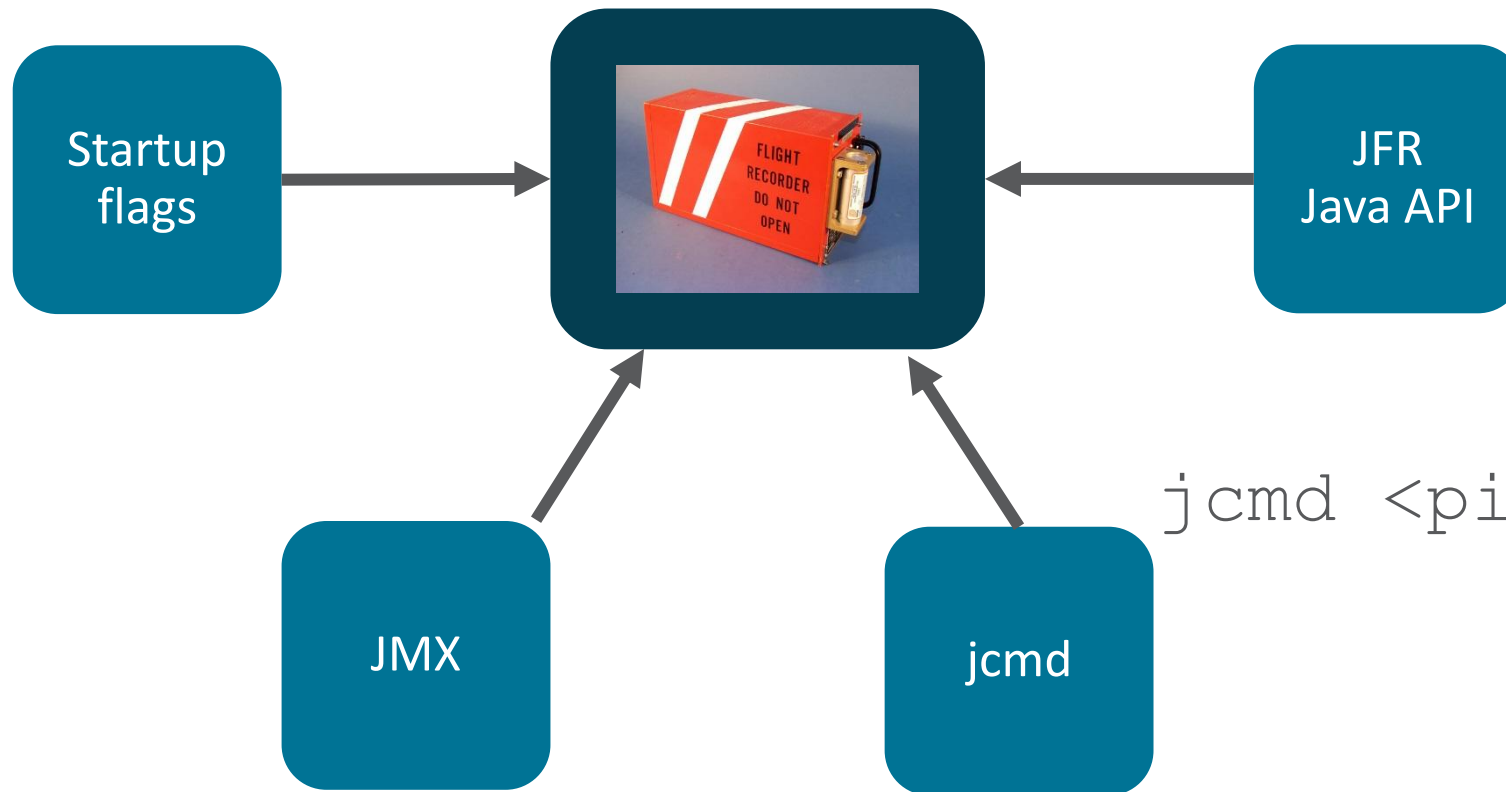
Creating Recordings Using Mission Control

- Easy and intuitive
1. Find a JVM to record from in the JVM Browser
 2. Double click the Flight Recorder node under the JVM
 3. Follow the wizard



Creating Recordings in other ways

`-XX:StartFlightRecording`



`jcmd <pid> JFR.start`

Analyzing Recordings

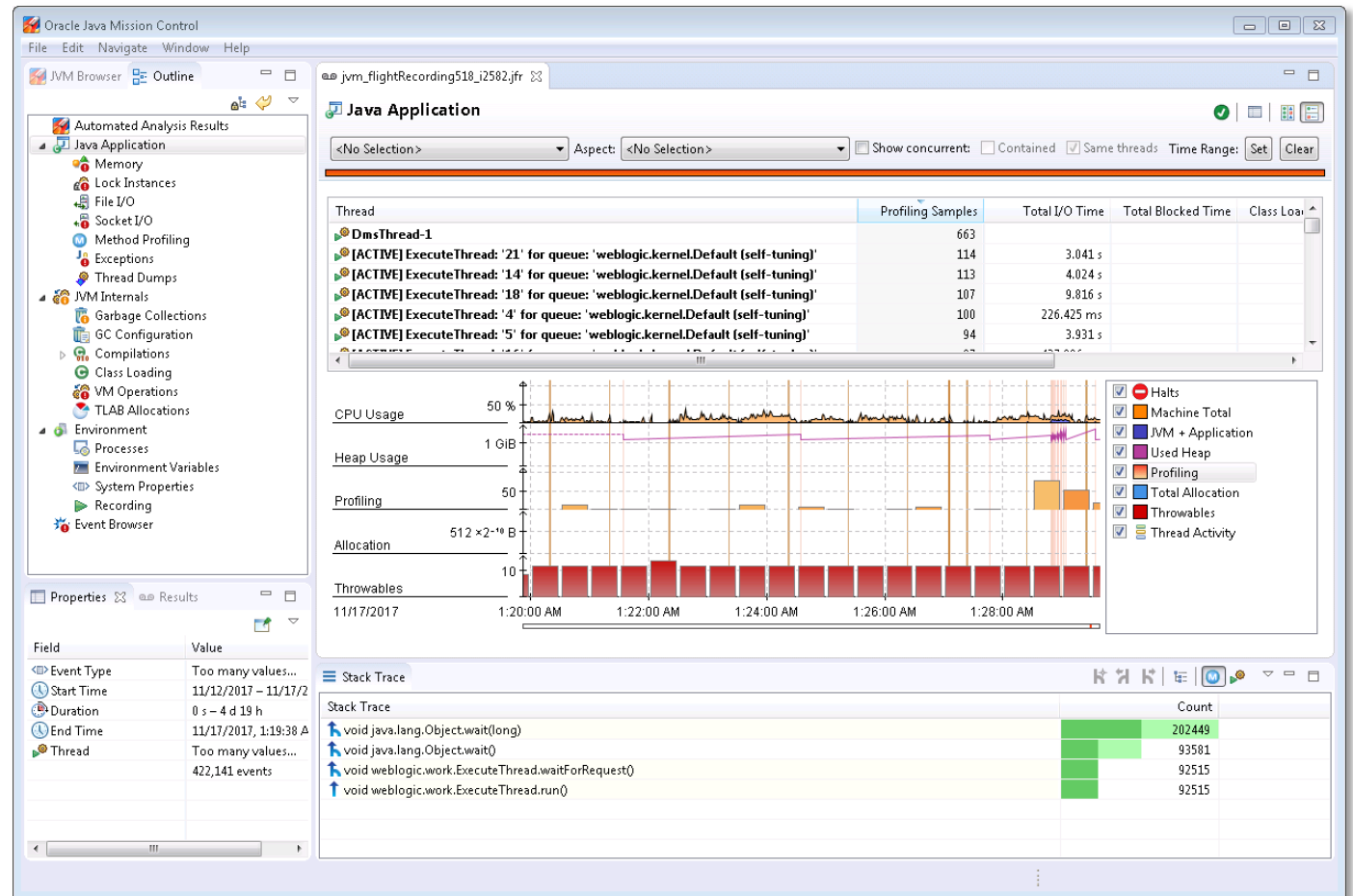


What the data can tell you

- Investigate bad behavior
- Find performance bottlenecks
- Some things are always bad
- CPU?
 - Batch job or real time trading

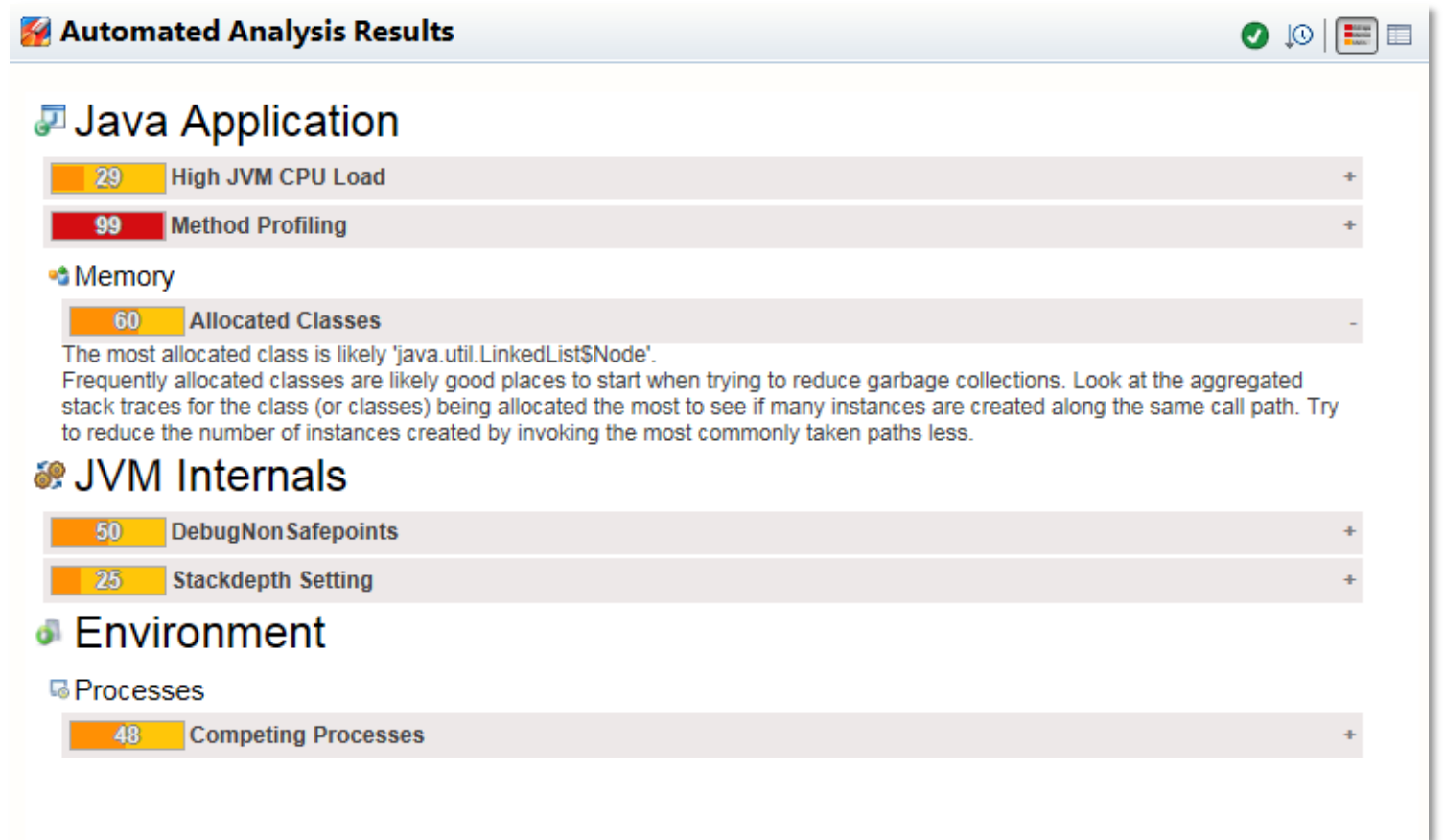
Analyzing Flight Recordings in JMC

- Data shown on pages with different themes
 - Memory
 - Locks
 - Method profiling
 - etc.



Automated Analysis of Recordings

- Recordings contain lots of data
- JMC rules attempt to:
 - Highlight relevant information
 - Explain detected problems and how to solve them
- Can be run from POJO using unsupported API



JFR and JMC Demo

Resources

- Home page
 - <http://oracle.com/missioncontrol> (Click Discussion to find the forum)
 - <https://docs.oracle.com/javase/9/troubleshoot/toc.htm>
- Twitter
 - @javamissionctrl
 - @hirt
- Blog
 - <http://hirt.se/blog/>

Open source

- Open sourcing
 - Java Flight Recorder
 - Mission Control
- Free for development now



Butterfly by Beverly is licensed under CC BY-NC-ND 2.0

Q&A



Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



ORACLE®