JMC (& JFR) – 2020 Vision

Looking ahead

Jie Kang

Senior Software Engineer

Red Hat

Marcus Hirt

Director of Engineering

Datadog

Red Hat

What we'll discuss today

Intro to JMC & JFR

Development Updates

Roadmap

Demo

Red Hat

Intro to JMC & JFR

JDK Flight Recorder (JFR)
Events-based JVM technology for production time profiling and diagnostics

JDK Mission Control (JMC)
Desktop application for JMX browsing and JFR visualization and analysis

JDK Flight Recorder

Low overhead recording
Compact, self-describing data
Extensible

Like a Flight Recorder in an airplane

JDK Mission Control

JMX Browser
JFR Visualization & Analysis

Core libraries for parsing and analyzing JFR data

JDK Flight Recorder
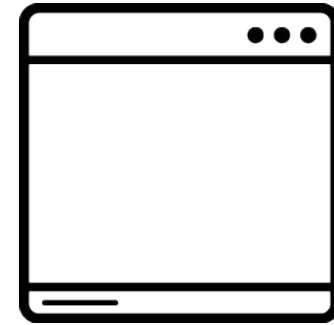# Toolchain

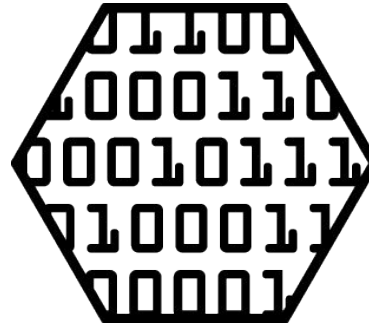Control JFR
> CLI, POJO, JMX, jcmd tool, JMC

Add data
> Java API, JMC Agent, 3rd-Party Integrations

Use data
> JMC, jfr tool, 3rd-Party Applications

Red Hat

General Workflow

Run JVM with JFR enabled   **Acquire JFR file**   **Analyze JFR data with JMC**

Mini Demo! (JMC)
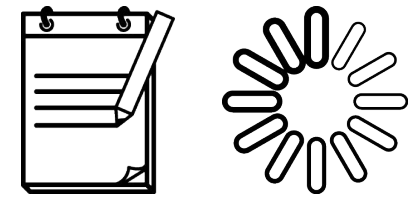
Red Hat

Development Updates

Red Hat

JMC project is now on GitHub via Skara

The contribution process can now be started with a GitHub pull request
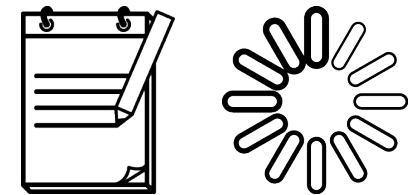
https://github.com/openjdk/jmc

JMC 7.0.0 ga tagged 2019-06-02

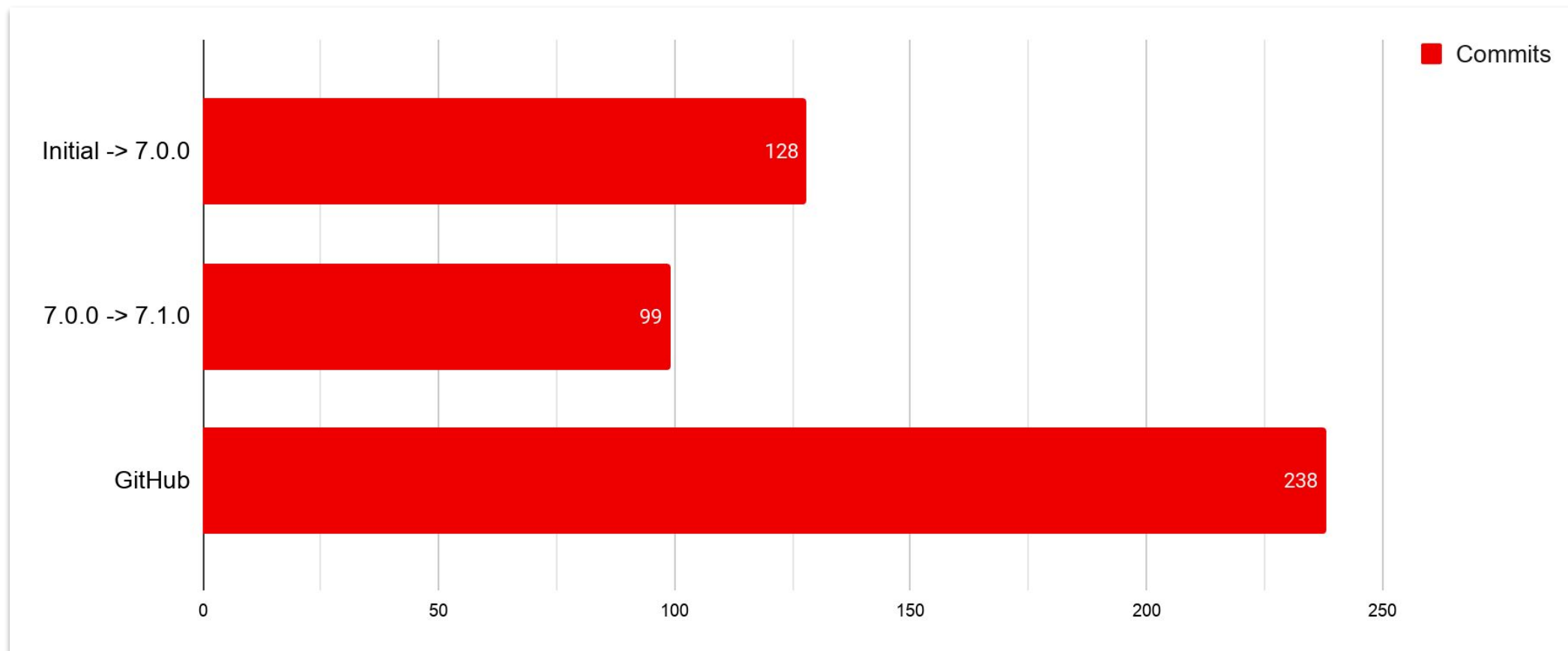First open-source release of JDK Mission Control

Red Hat

JMC 7.1.0 ga tagged 2019-12-06

Rule optimizations
SWT-based JOverflow View
Flamegraph View
HdrHistogram View

JMC Project Commits

JMC Distributions

AdoptOpenJDK (Binary)
https://adoptopenjdk.net/jmc.html

Oracle (Binary & Eclipse Update Site)
Red Hat (RPMs for Fedora 29+ and RHEL 7+)

15

Contributing to JMC

Repo: https://github.com/openjdk/jmc
Mailing List: https://mail.openjdk.java.net/mailman/listinfo/jmc-dev
Slack: https://jdkmissioncontrol.slack.com
Bug tracker: https://bugs.openjdk.java.net/projects/JMC/issues
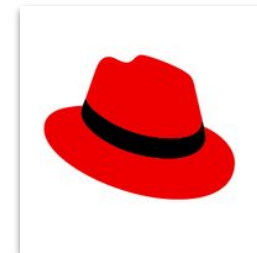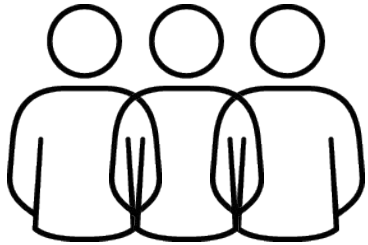
JMC Tutorial: https://github.com/thegreystone/jmc-tutorial

Roadmap

Red Hat

JFR in OpenJDK 8

CSR finalized and closed 2019-12-20
https://hg.openjdk.java.net/jdk8u/jdk8u-jfr-incubator

Todo: Merge into mainline
https://mail.openjdk.java.net/pipermail/jdk8u-dev/2020-January/011063.html

Red Hat

JMC 8.0.0

Major release to target OpenJDK 15

OpenJDK 15: September 2020
JMC 8: ~November 2020

Red Hat

JMC 8.0.0

JMC Agent

Rules 2.0

Update core libraries

Red Hat

Dynamic JFR Events
JMC Agent

Declarative insertion of JFR events at runtime
Can be redefined
Optimized for the job

Red Hat

Automated Analysis
Rules 2.0

Improve rule execution performance
Redesign rules for reuse
Add more typed information for visualization
Remove HTML from rule results

Red Hat

Updates!
Core API

Use JDK 8 language features (previously 7)
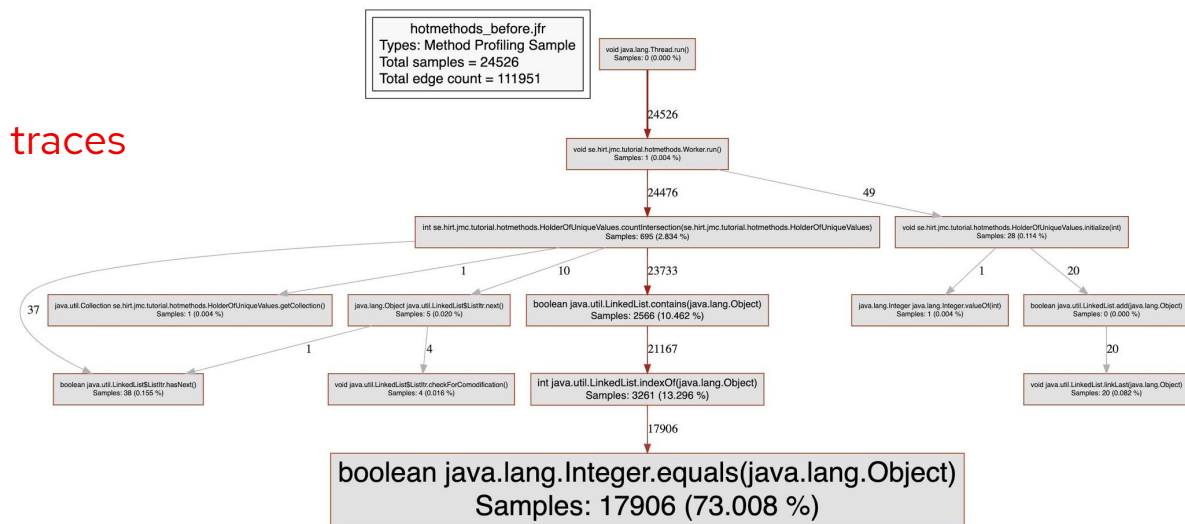Move some application API to Core for reuse by third-party applications

JMC 8.0.0

New aggregated stack trace visualizations
Thread graph improvements
Flame Graph 2.0

Graph visualizations of stack traces

Demo: JFR management in OpenShift with container-jfr

Red Hat

One flightrecorder
resourcekind is
auto-managed per
monitorable instance in
the project

flightrecorder

flightrecorder

One container-jfr-operator
per project

flightrecorder

container-jfr-operator

flightrecorder

User sends
request for
robot. Order,
factory and
customer
services work
to fulfill order

containerjfr

One containerjfr
per project

robotshop-factory-service

container-jfr
container-jfr-web
jfr-datasource
grafana

container-jfr can
connect to java
applications in other
pods

user

robotshop-order-service

robotshop-customer-service

jaeger

One jaeger-operator per
project OR per cluster

Services send traces to
Jaeger

One jaeger per
project OR per
cluster

jaeger-operator

Setup Instructions:
https://github.com/jiekang/fosdem2020-demo

Red Hat

container-jfr

Controlling JFR on JVMs in containers (Docker, Kubernetes, OpenShift)

GitHub Organization:
https://github.com/rh-jmc-team

container-jfr features

Start/Stop recordings

Archive recordings, download recordings

View automated analysis immediately

View events available for the JVM

Manipulate via web UI or oc CLI

Link to Grafana

container-jfr

JFR management service
k8s/oc API for JFR management

container-jfr-core

Core libraries for JFR management

container-jfr-web

Web front-end to ease use of container-jfr

container-jfr-operator

Kubernetes/OpenShift Operator to ease deployment of container-jfr

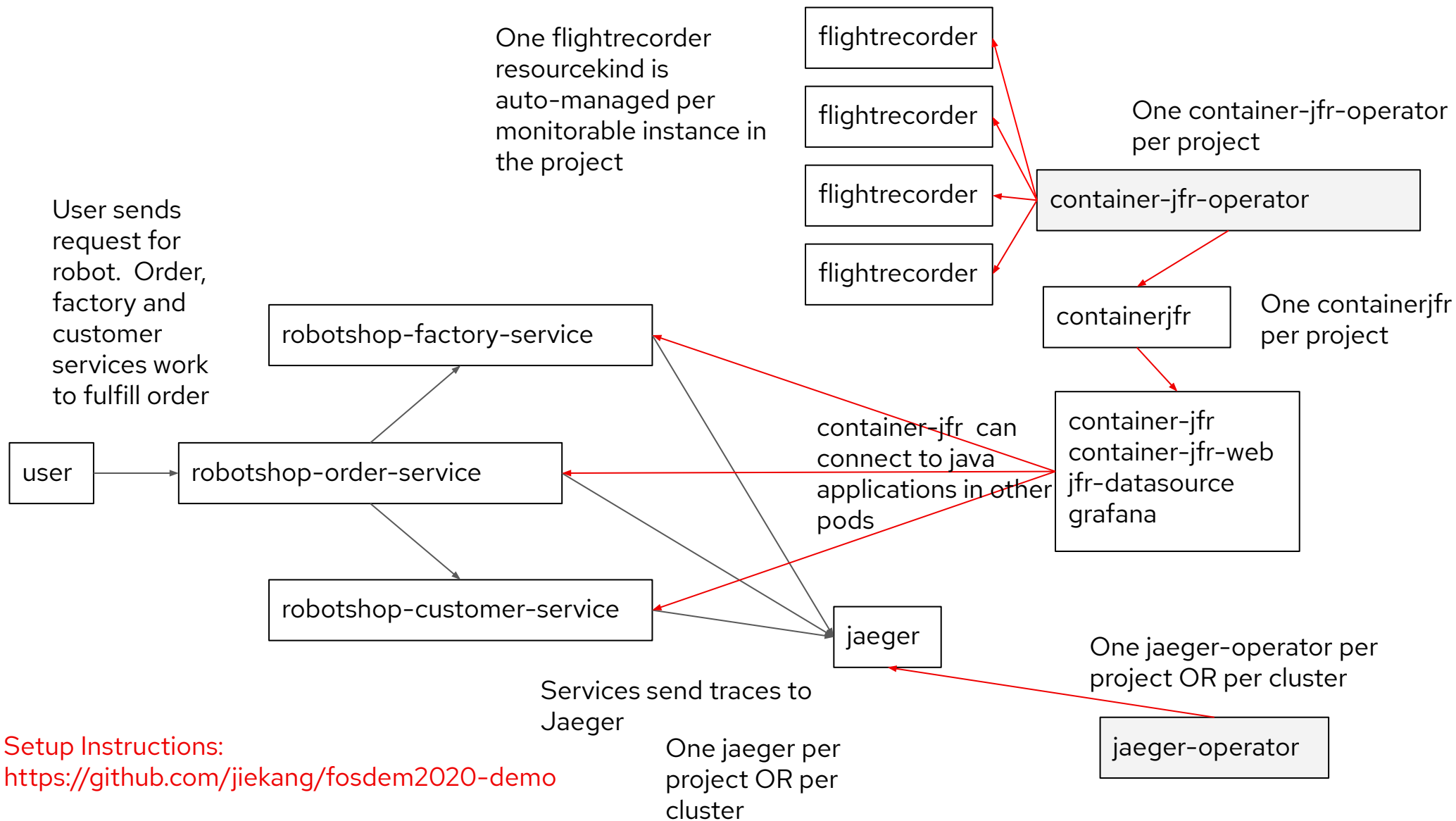# jfr-datasource

Grafana datasource for JFR files

https://github.com/rh-jmc-team/jfr-datasource

Jaeger

Open-source, end-to-end distributed tracing
Monitor and troubleshoot transactions in complex distributed systems

https://www.jaegertracing.io/
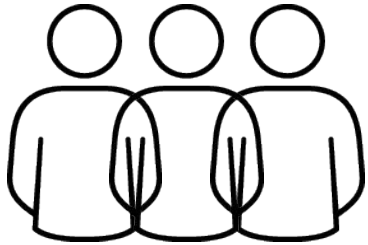
One flightrecorder
resourcekind is
auto-managed per
monitorable instance in
the project

flightrecorder

flightrecorder

One container-jfr-operator
per project

flightrecorder

container-jfr-operator

User sends
request for
robot. Order,
factory and
customer
services work
to fulfill order

flightrecorder

containerjfr

One containerjfr
per project

robotshop-factory-service

container-jfr
container-jfr-web
jfr-datasource
grafana

container-jfr can
connect to java
applications in other
pods

user

robotshop-order-service

robotshop-customer-service

jaeger

One jaeger-operator per
project OR per cluster

Services send traces to
Jaeger

jaeger-operator

One jaeger per
project OR per
cluster

Red Hat

Demo: JFR management in OpenShift with container-jfr

Red Hat

Contributing to JMC

Repo: https://github.com/openjdk/jmc
Mailing List: https://mail.openjdk.java.net/mailman/listinfo/jmc-dev
Slack: https://jdkmissioncontrol.slack.com
Bug tracker: https://bugs.openjdk.java.net/projects/JMC/issues

JMC Tutorial: https://github.com/thegreystone/jmc-tutorial

Thank you

Special thanks to Marcus Hirt for providing content

for the JMC project, updates and roadmap, and the

deep distributed tracing demo

Red Hat