FerbJmon Tools

Visualizing Thread Access on Java Objects using Light-weight Runtime Monitoring

Dr. Marvin Ferber
Technical University Bergakademie Freiberg, Germany

1st European Workshop on Parallel and Distributed Computing Education for Undergraduate Students (Euro-EDUPAR)
Vienna, Austria, August 24, 2015

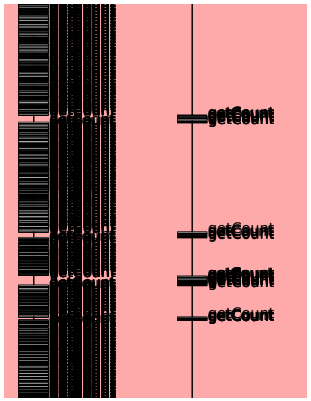
Multi-Threaded programming in Java

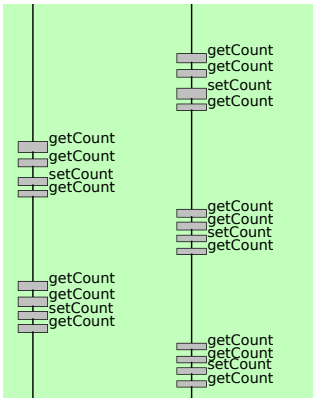
- **Threads in Java**
  - Class Thread (java.lang.Thread)
  - Interface Runnable (java.lang.Runnable)
- **Synchronization of Threads**
  - Synchronized Blocks (mutex)
  - Signals via wait() and notify()

Motivating Example

Task

- **alternating incrementation of an int value**
  - EvenThread
  - OddThread





Solution

- **mutual exclusion on int value update**
  - ➔ lock to determine value
  - ➔ many unsuccessful lock attempts
- **add signals to hand over control**
  - ➔ only lock if necessary
- **both version produce correct result value**

Motivation – Verify correct behavior?

Q

- **monitor and visualize**
  - concurrent access to Java objects?
  - minimal changes to the code?

A

- **static code analysis not useful**
- **monitor access to objects at runtime**
  - capture temporal alignment of access
  - detect changes to the object

FerbJmon Tools use Java Bytecode instrumentation

- add/modify Bytecode of user classes at runtime → inject monitoring
- transparent to the program (Java agent)

```
java -cp <classpath> my.package.MyClass
```

```
java -javaagent:<agent.jar> -cp <classpath> my.package.MyClass
```

- modified system classes must be loaded via Bootstrap classloader

```
java -Xbootclasspath/p:<sys.jar> -javaagent:<agent.jar>
-cp <classpath> my.package.MyClass
```

How to use FerbJmon Tools

- **wrapper scripts for all FerbJmon Tools**

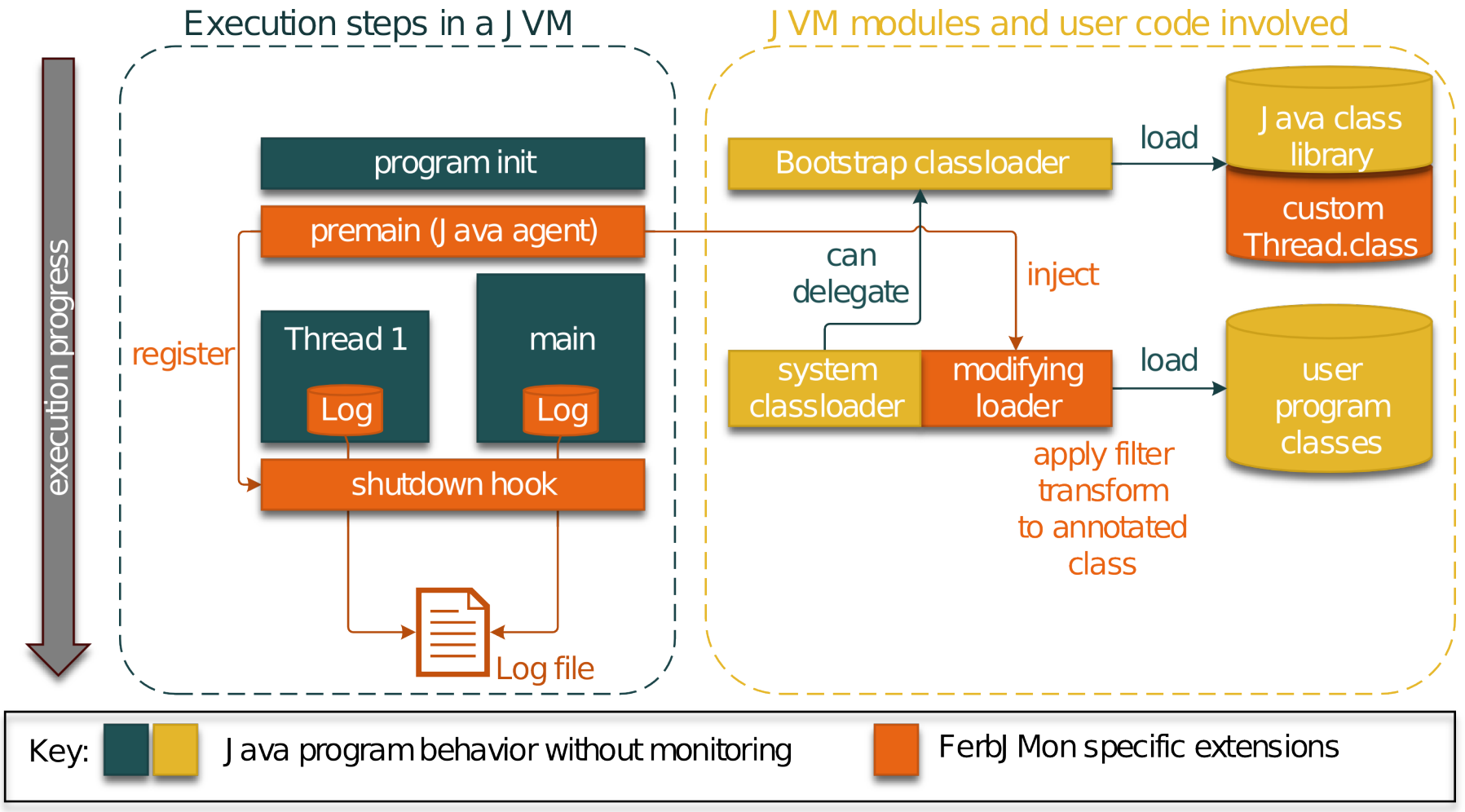
```
java -cp <classpath> my.package.MyClass
```

```
java_callgraph -cp <classpath> my.package.MyClass
java_threadorder -cp <classpath> my.package.MyClass
```

- **monitored class needs `@Monitored` annotation**

```
@de.ubt.ferbjmon.annotation.Monitored
public class MyStruct {
    public MyStruct(){ // some constructor }
    public void perform(){ // perform some action}
    public float internal(){ // some internal func.}
}
```

FerbJmon Tools Implementation

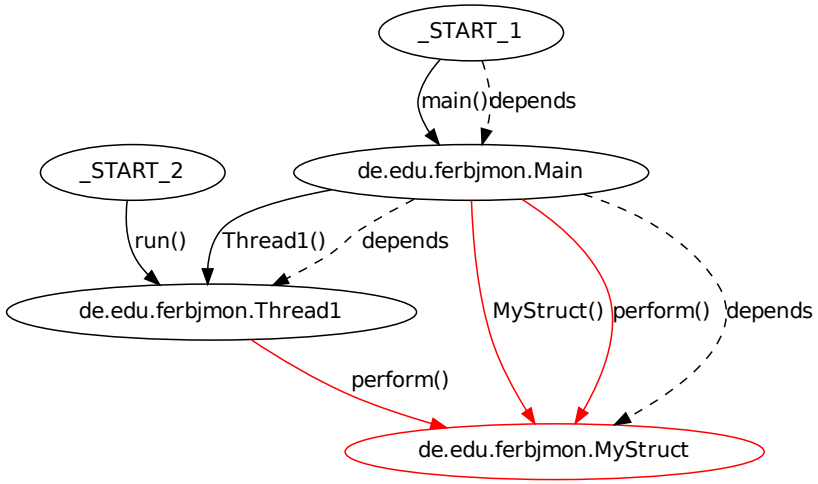


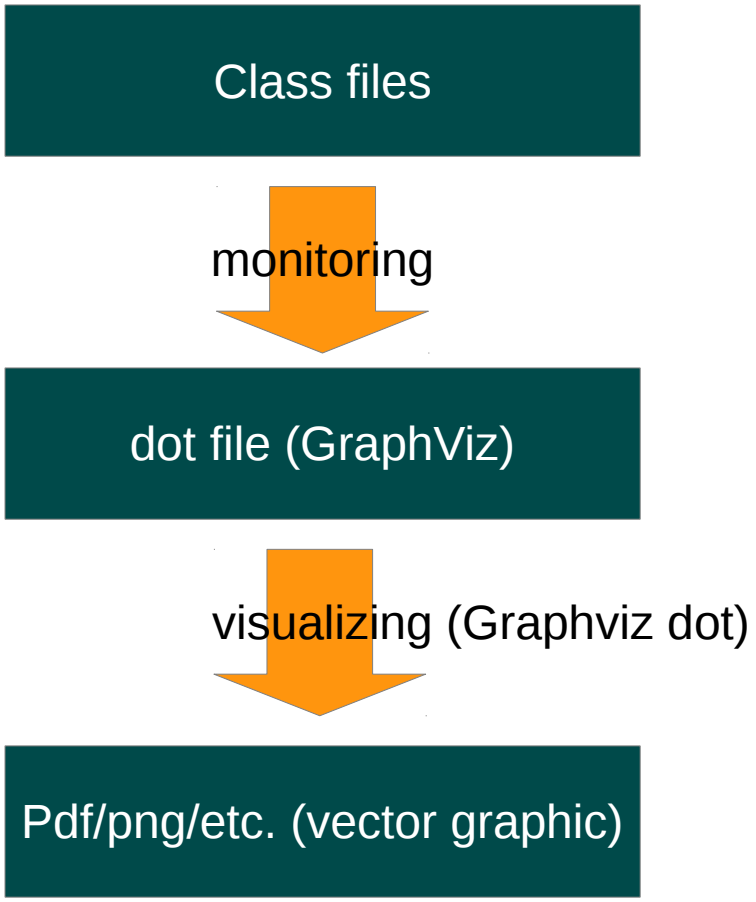
FerbJmon Logging Mechanisms

- **MemArray Logger**
  - main memory, fixed size array
  - fast and steady, but amount of logs limited
- **StringBuilder Logger**
  - main memory, variable size (StringBuilder)
  - restricted by size of RAM
- **File Logger**
  - file on disk, variable size (BufferedWriter)
  - slow, but good start for unknown behavior

FerbJmon Call Graph

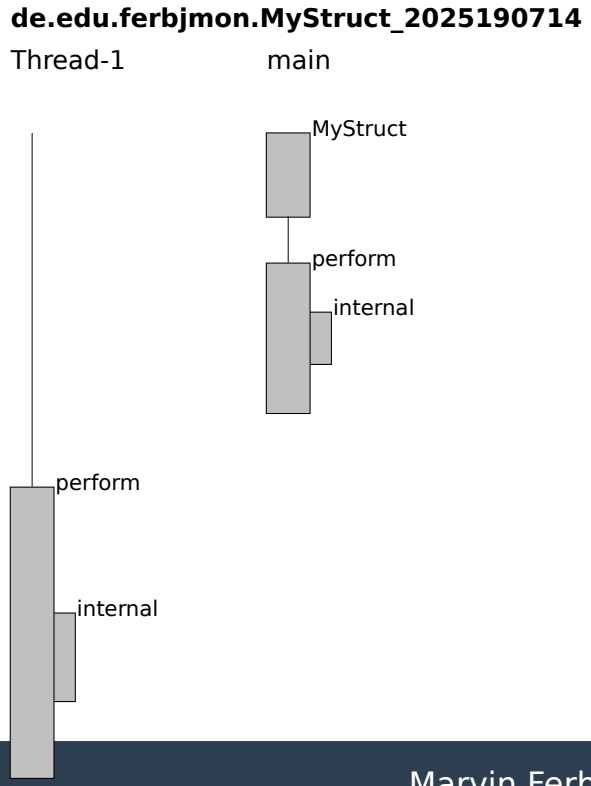
- **Graph view of program run:**
  - Method dependencies
  - Variable dependencies
- **Overview of involved user classes**

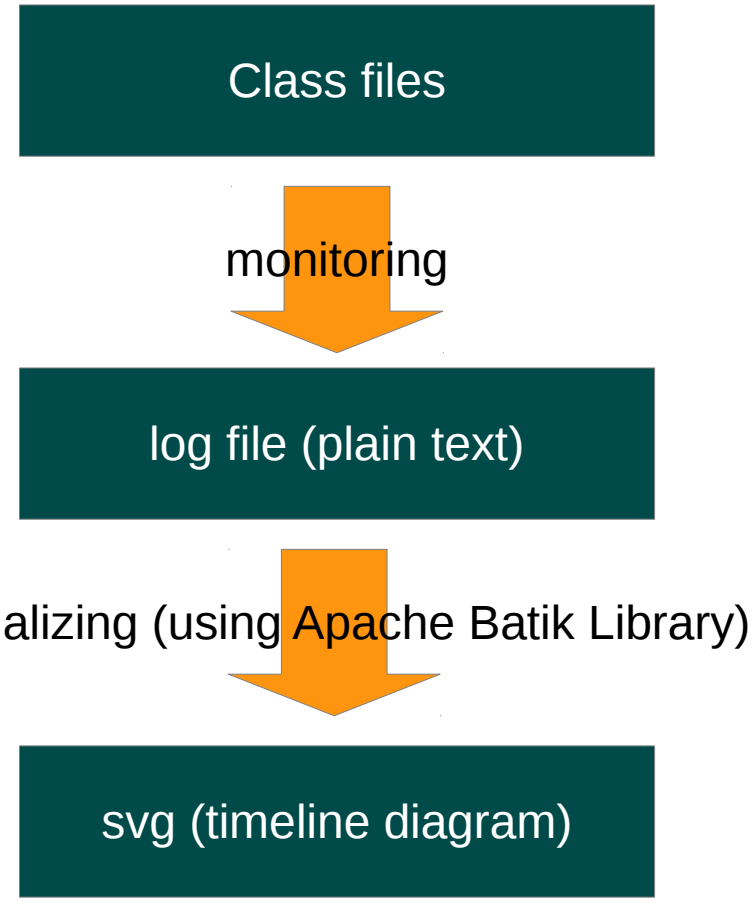




FerbJmon Thread Order

- **timeline diagram**
  - access to each instance
  - similar sequence diagram





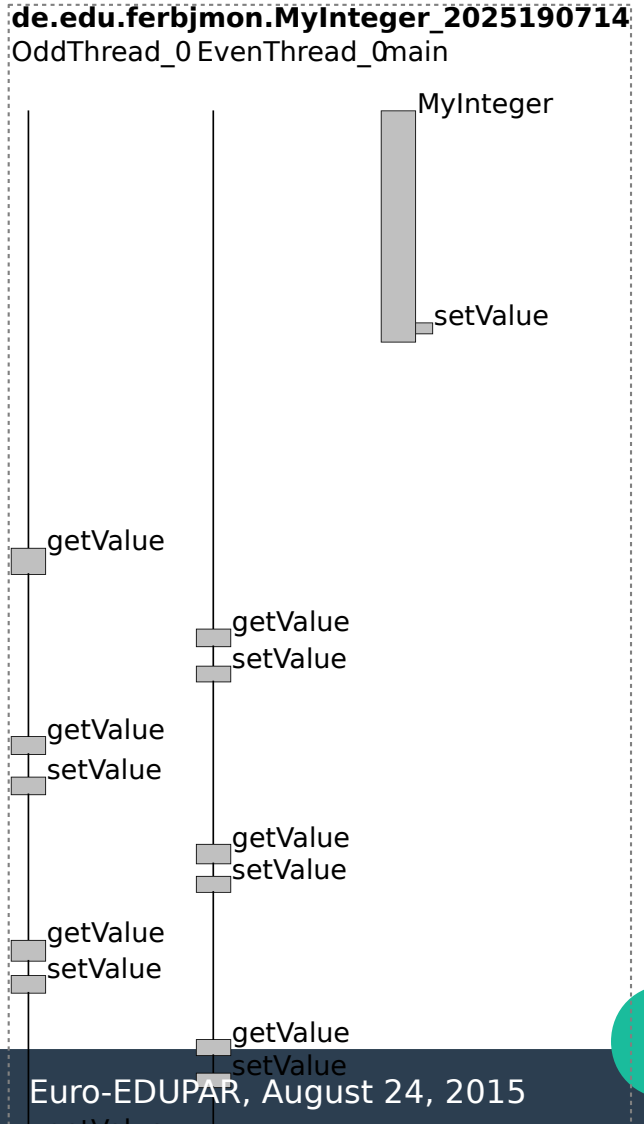
2nd Example - alternating incrementation of an int value

example code OddThread

```
import de.edu.ferbjmon.MyInteger;

public class OddThread extends Thread {
        private MyInteger myinteger;
        private int threshold;
        // constructor hidden here
        public OddThread(MyInteger my, int threshold, int i) {..}
        public void run() {
                while (true) {
                        // computation here
                        synchronized (myinteger) {
                                int value = myinteger.getValue();
                                if (value % 2 == 1) {
                                // EvenThread (value % 2 == 0) instead
                                        myinteger.setValue(value + 1);
                                        if ((value + 1) > threshold) break;
                                }
                                myinteger.notify();
                                try {myinteger.wait(); }
                                catch (InterruptedException e) {}
                        }
                }
                synchronized (myinteger) {
                        myinteger.notifyAll();
}}}
```

timeline diagram MyInteger



Thank you

Questions?

Send E-Mail?

marvin.ferber@informatik.tu-freiberg.de

Wanna try?

https://www.dropbox.com/s/o5shspqj4wlnmx9/ferbjmon.tar.gz

Dr. Marvin Ferber
Technical University Bergakademie Freiberg, Germany