jpf-concurrent: An Extension of JPF for Java.util.concurrent

Mateusz Ujma
University of Oxford
Oxford, United Kingdom

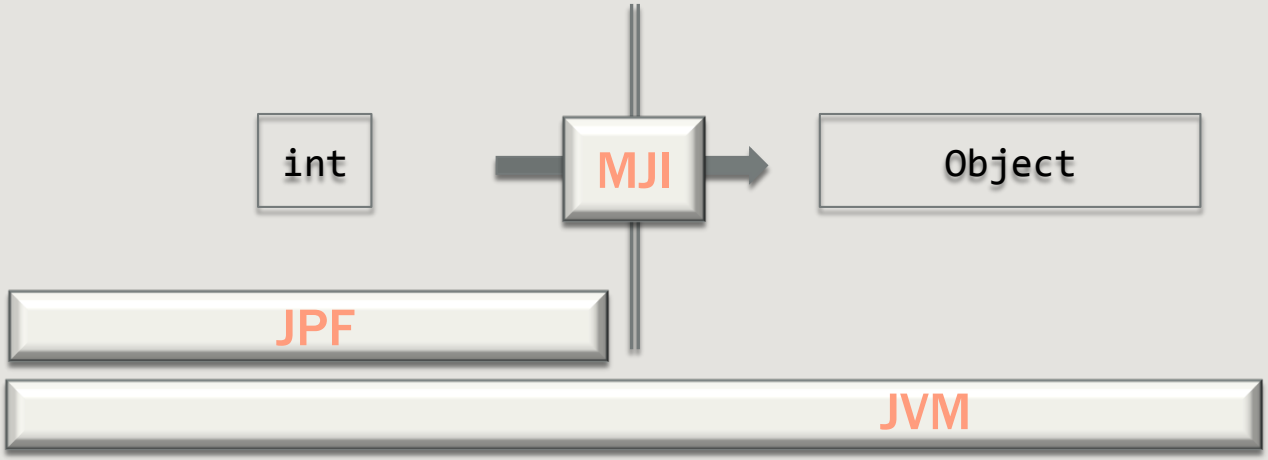
Nastaran Shafiei
York University
Toronto, Canada

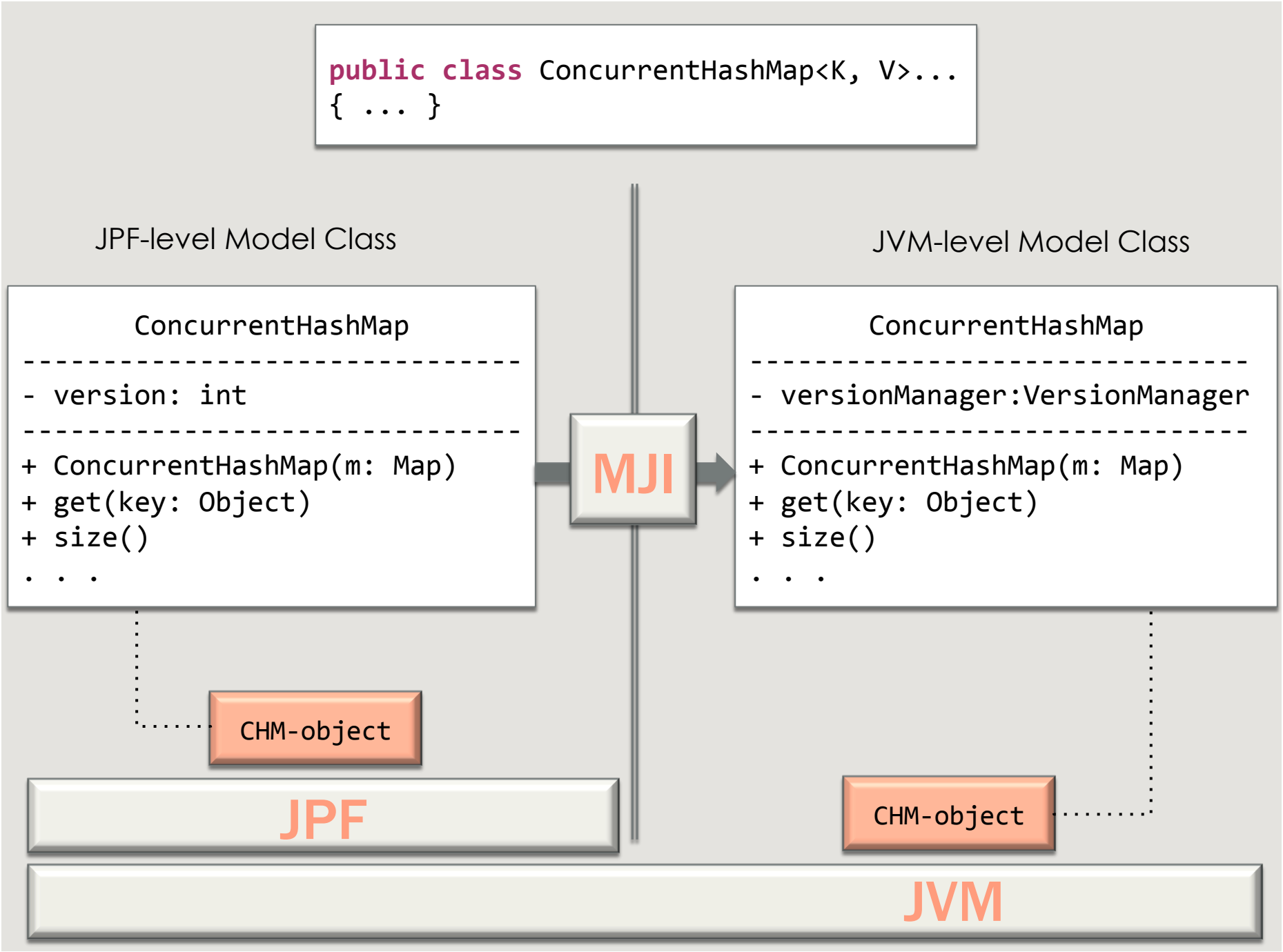
INTRODUCTION

- Java concurrency utilities simplify the development of multi-threaded Java programs
- Using JPF to model check such programs gives rise to several problems
  - Unhandled native calls
  - Large state spaces

OUR SOLUTION

- Created **jpf-concurrent**
  - Modeled `java.util.concurrent` based on Java APIs
- The method is based on:
  - Abstracting the state of the application in JPF
  - Using the MJI of JPF to delegate the execution to JVM

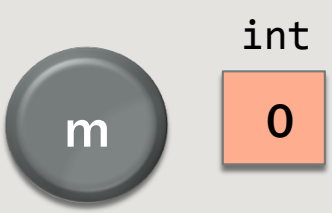



public class ConcurrentHashMap<K, V>...
{ ... }
JPF-level Model Class
JVM-level Model Class
ConcurrentHashMap
- version: int
+ ConcurrentHashMap(m: Map)
+ get(key: Object)
+ size()
. . .
MJI
ConcurrentHashMap
- versionManager:VersionManager
+ ConcurrentHashMap(m: Map)
+ get(key: Object)
+ size()
. . .
CHM-object
JPF
CHM-object
JVM

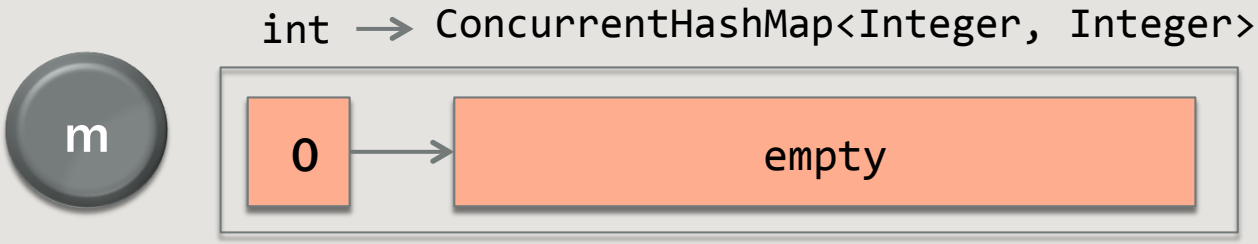
EXAMPLE

```
• ConcurrentHashMap<Object,Object> m =
        new ConcurrentHashMap<Object,Object>();
• m.put(key,value);
• m.clear();
```

JPF



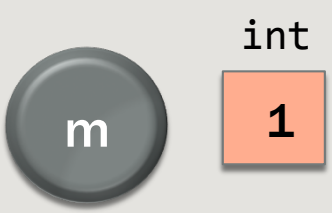
JVM



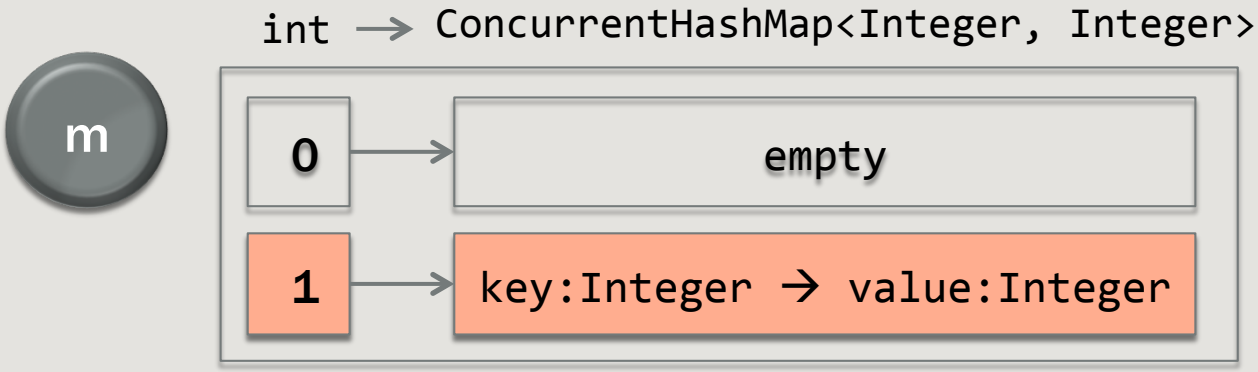
EXAMPLE

```
• ConcurrentHashMap<Object,Object> m =
        new ConcurrentHashMap<Object,Object>();
• m.put(key,value);
• m.clear();
```

JPF



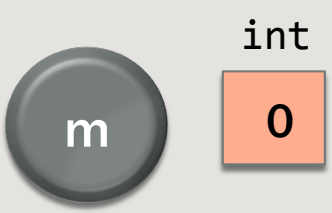
JVM



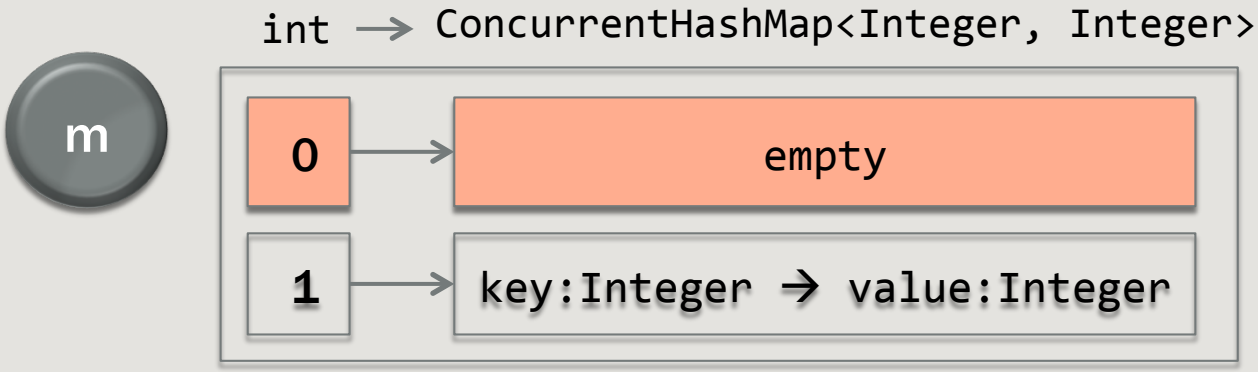
EXAMPLE

```
• ConcurrentHashMap<Object,Object> m =
        new ConcurrentHashMap<Object,Object>();
• m.put(key,value);
• m.clear();
```

JPF



JVM



DISCUSSION

- We cannot delegate the executions of all the methods to the JVM
- Only the objects created at JPF level can be garbage collected
- By moving the implementation to the JVM certain interleaving may not be considered

RESULTS

Class(# of threads)	jpf-concurrent disabled		jpf-concurrent enabled			
	states	time (sec)	states	time (sec)	state-space reduction	speedup
ReentrantLock(5)	620,932	229	176,775	47	3.5x	4.8x
ReentrantLock(6)	9,973,504	4472	1,940,134	850	5.1x	5.2x
ConurrentHashMap(11)	1,054,651	732	753,727	347	1.3x	2.1x
ConurrentHashMap(12)	4,640,047	3502	3,153,398	1630	1.4x	2.1x

SUMMARY & FUTURE WORK

- Proposed a new method to model Java Concurrency Utilities
  - The method is based on state abstraction
- Future work
  - Providing support for classes added in Java 6 and 7
  - Supporting operations that rely on time

Thank You