FOSDEM 2007
Java HotSpot™ Virtual Machine

Peter B. Kessler
Senior Staff Engineer
Sun Microsystems, Inc.

What Does the Virtual Machine Do?

- Bytecode execution
  - Interpreter (also does profiling)
  - 2 runtime compilers: **`-client`** and **`-server`**
  - On-stack replacement (OSR)
- Storage allocation and garbage collector
  - Fast inline allocation; concurrent, parallel collectors
- Runtimes
  - Start up, shut down, class loading, threads, synchronization, safepoints, interactions with operating system, *etc*.

Interactions with the Platform

- Few API's
  - jni.h
  - JVM/TI
- “Intrinsification” of hot, or easy, or critical methods
  - java.lang.Math.sin
  - java.lang.String.indexOf
  - sun.misc.Unsafe.compareAndSwapInt
- Mostly implementation
  - Can be on a different release cycle than the JRE

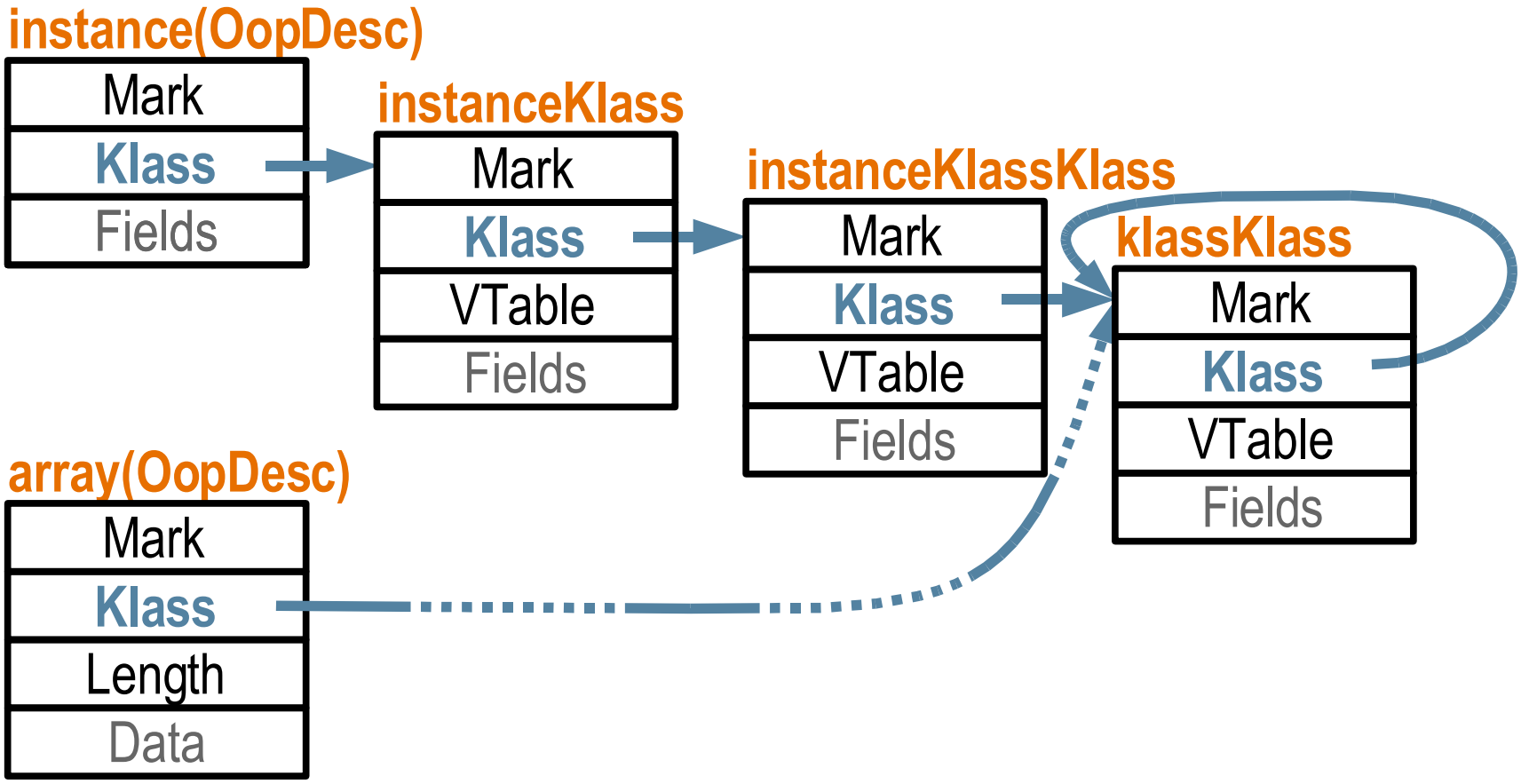
Some Things to Know (1)

Mark word in every object

Bitfields			Tag	State
Hashcode	Age	0	01	Unlocked
Lock record address			00	Light-weight locked
Monitor address			10	Heavy-weight locked
Forwarding address, etc.			11	Marked for GC
Thread ID	Age	1	01	Biased / biasable

Some Things to Know (2)

Klass in every object



Process

- Plan features, or get bug reports
- Discuss with group, *etc*.
- Write code
- Test for correctness, performance impact
- Code review by 2 other engineers
- Nightly testing in group workspace
- Integration testing in VM workspace
- “Big Apps” tests, cross-platform performance tests
- Backport (if necessary)

Actual Code!

- Coming up on 10 years of coding
- 188K lines of code
- 136 contributors + sponsored changes
  - *You could be next!*
- The virtual machine is already out under GPLv2
- Competition drives innovation
  - So do real applications and customers
- What's fun about it?
  - Complex multithreaded program that has to work perfectly!

Outside Projects

- University of Linz, Austria
  - "Escape Analysis in the Context of Dynamic Compilation and Deoptimization", Thomas Kotzmann, Ph.D.
  - "Linear Scan Register Allocation for the Java HotSpot? Client Compiler", Christian Wimmer, M.S., 2004
  - Others papers with Hanspeter Mössenböck
- Doug Lea, SUNY Oswego
  - Changes for JSR-133: memory model and threads

Further Reading

- http://java.sun.com/javase/technologies/hotspot/gc/memorymanagement_whitepaper.pdf
- T. Printezis and D. L. Detlefs. “A Generational Mostly-Concurrent Garbage Collector”. In A. L. Hosking, editor, P*roceedings of the 2000 International Symposium on Memory Management (ISMM 2000)*, pages 134-154, Minneapolis, MN, USA, October 2000. ACM Press.

Blogs

- http://blogs.sun.com/jonthecollector
- http://blogs.sun.com/dagastine
- http://blogs.sun.com/tony