The VMKit project: Java (and .Net) on top of LLVM

Nicolas Geoffray
Université Pierre et Marie Curie, France

nicolas.geoffray@lip6.fr

What is VMKit?

- Glue between existing VM components
  - LLVM, GNU Classpath, BoehmGC, Mono and Pnet class libraries
- A drop-in replacement to Java and .Net
- Usage:
  - vmkit -java HelloWorld
  - vmkit -net HelloWorld.exe

This talk:
Internals of VMKit (for Java)

1. The design of VMKit
2. VMKit's performance
3. Limitations
4. Research projects

The design of VMKit

Compiler

Two strategies:

- LLVM JIT --> Emits in-memory
- LLVM AOT --> Emits dynamic library

Same code base, except:

- JIT: Direct reference to objects
- AOT: Emission of global variables

Threads

Uses POSIX threads

- Fat locking and scheduler

With fast thread local storage access

- User-level stacks
- @TLS = Mask on the frame pointer
- Thin locks with biased locking

Garbage Collector

The weakest part

- Boehm GC and Mmap2
- Unprecise, non-copying, non-generational

But not impossible

- LLVM has all the bits for GC-support in JIT

Exceptions

Uses GCC unwinding runtime

- Same as gcj
- But __frame_register ***is slow***

In development: exception checks

- There will be checks for GC anyway
- No more dwarf, cxa, libgcc_s

Runtime

- Bootstrap class loader
- Parser for .class and .jar files
- Internal representation of classes
- JIT Callbacks
- Stack inspection
- GNU Classpath native methods
- JNI methods

Execution overview

- **Load** .class and .jar
- **JIT or dlopen** main()
  - In case of JIT, insert stubs for methods (lazy compilation) and fields
- **Run** main()
  - Stubs call JIT
  - Load class dynamically

Portability

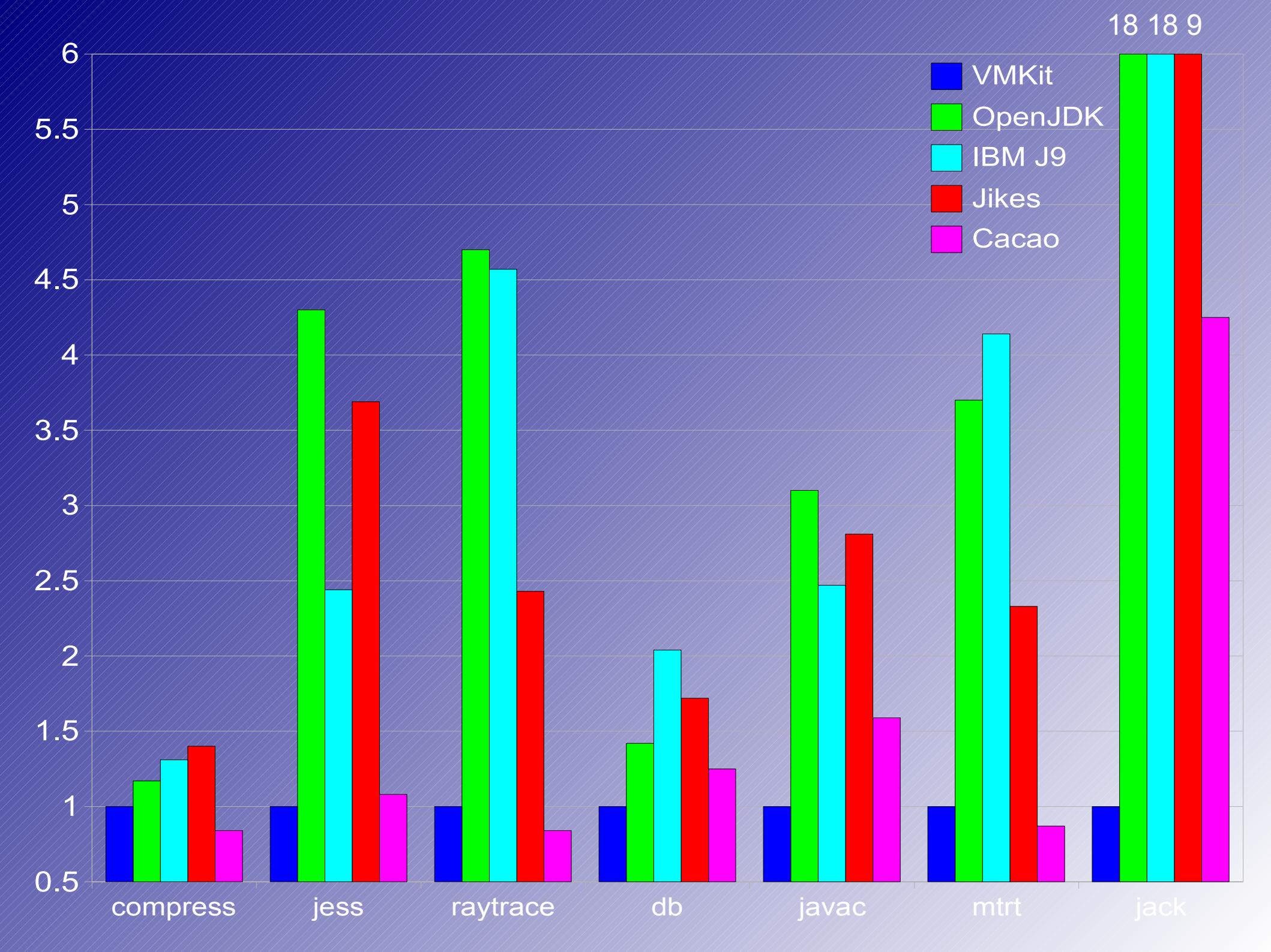
OS/Arch	JIT	AOT	GC
Linux/x86	✓	✓	Boehm or Mmap2
MacOSX/x86	✓	✓	Boehm or Mmap2
Linux/x64	✓	✗	Boehm
Linux/ppc32	✓	✓	Boehm or Mmap2
MacOSX/ppc32	✓	✓	Boehm or Mmap2
../ARM, PPC64	?	?	?
.../IA64, Mips, ...	✗	?	?

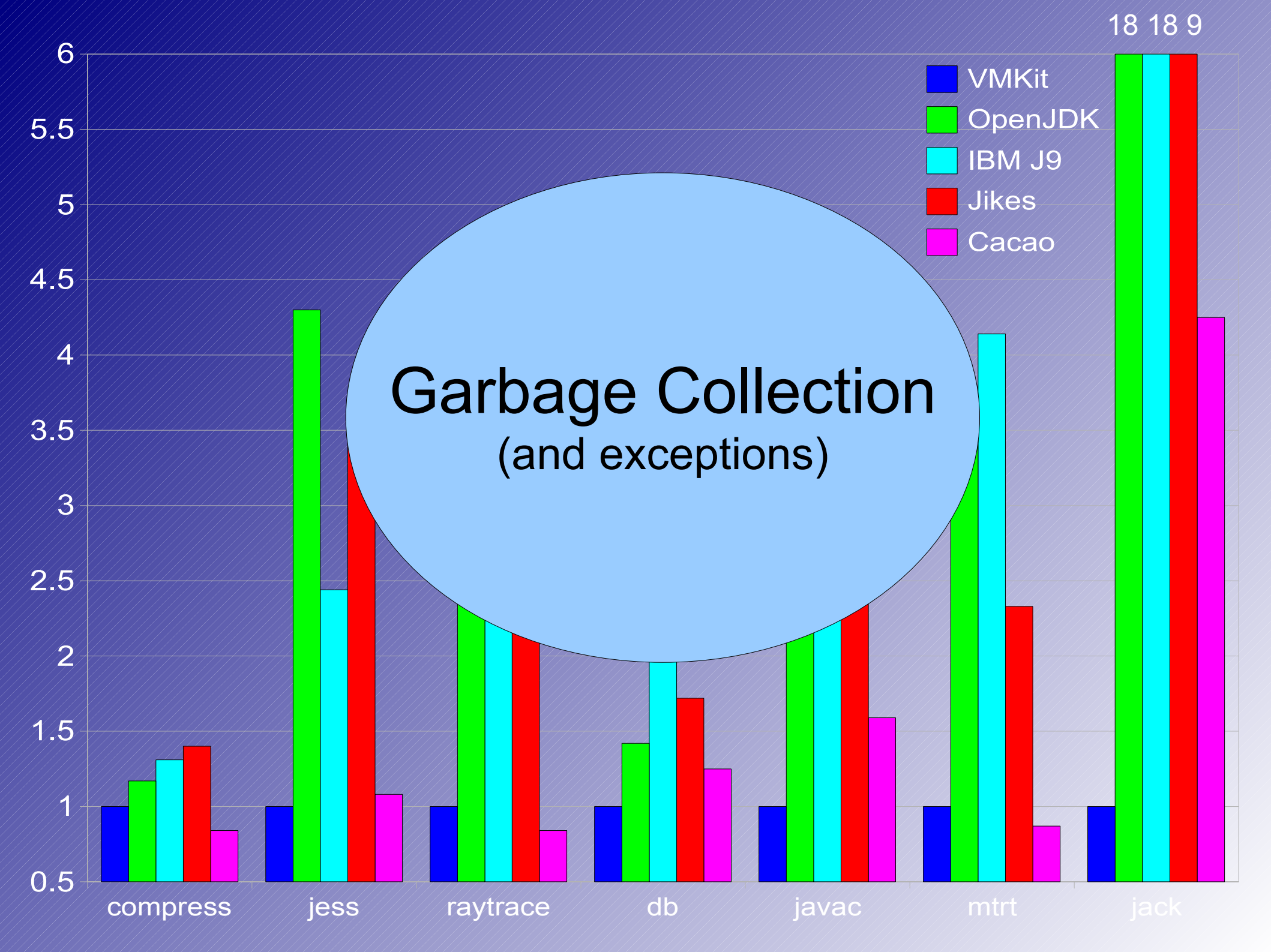
Where do we stand?

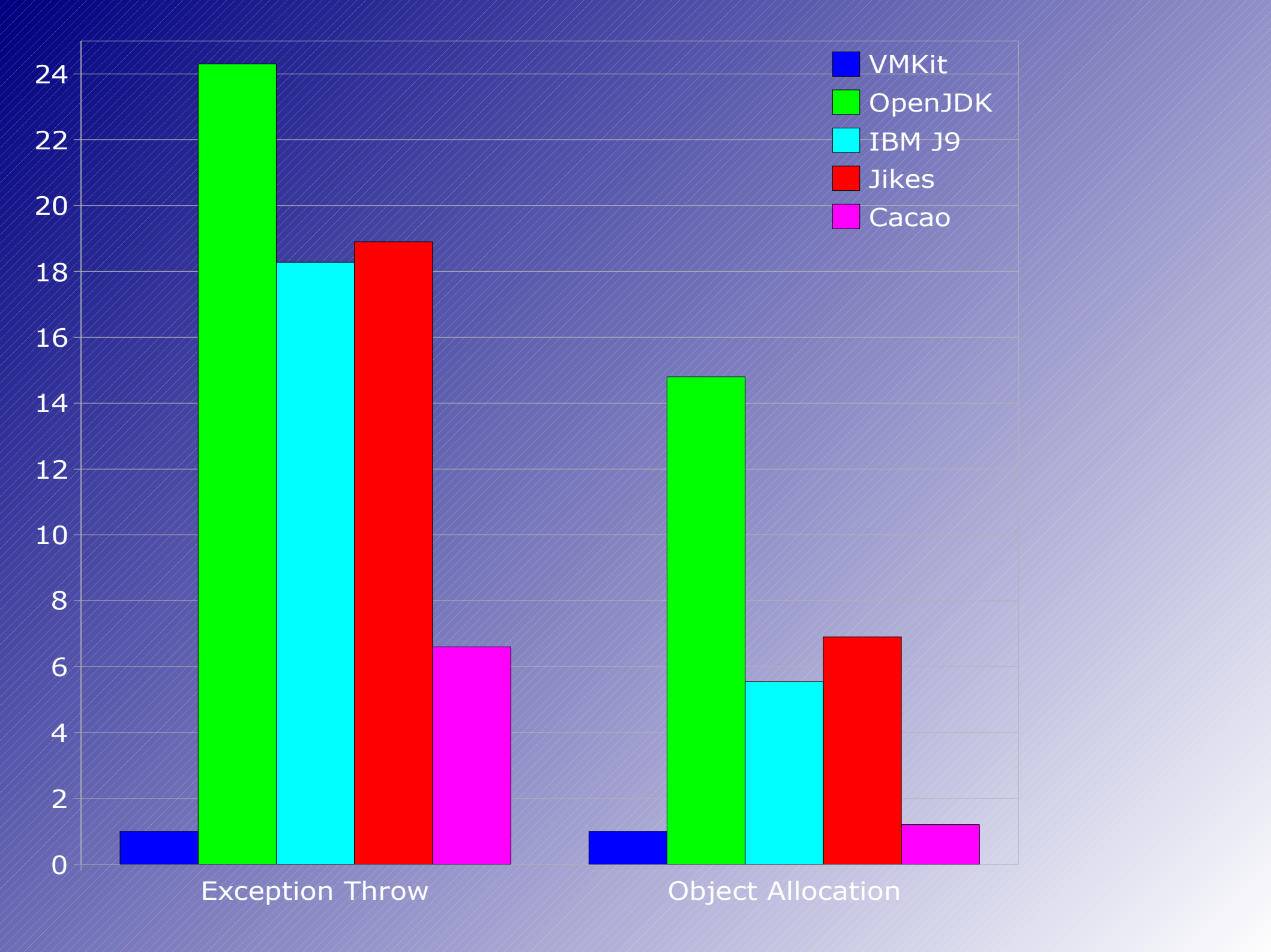
JVM Benchmarks

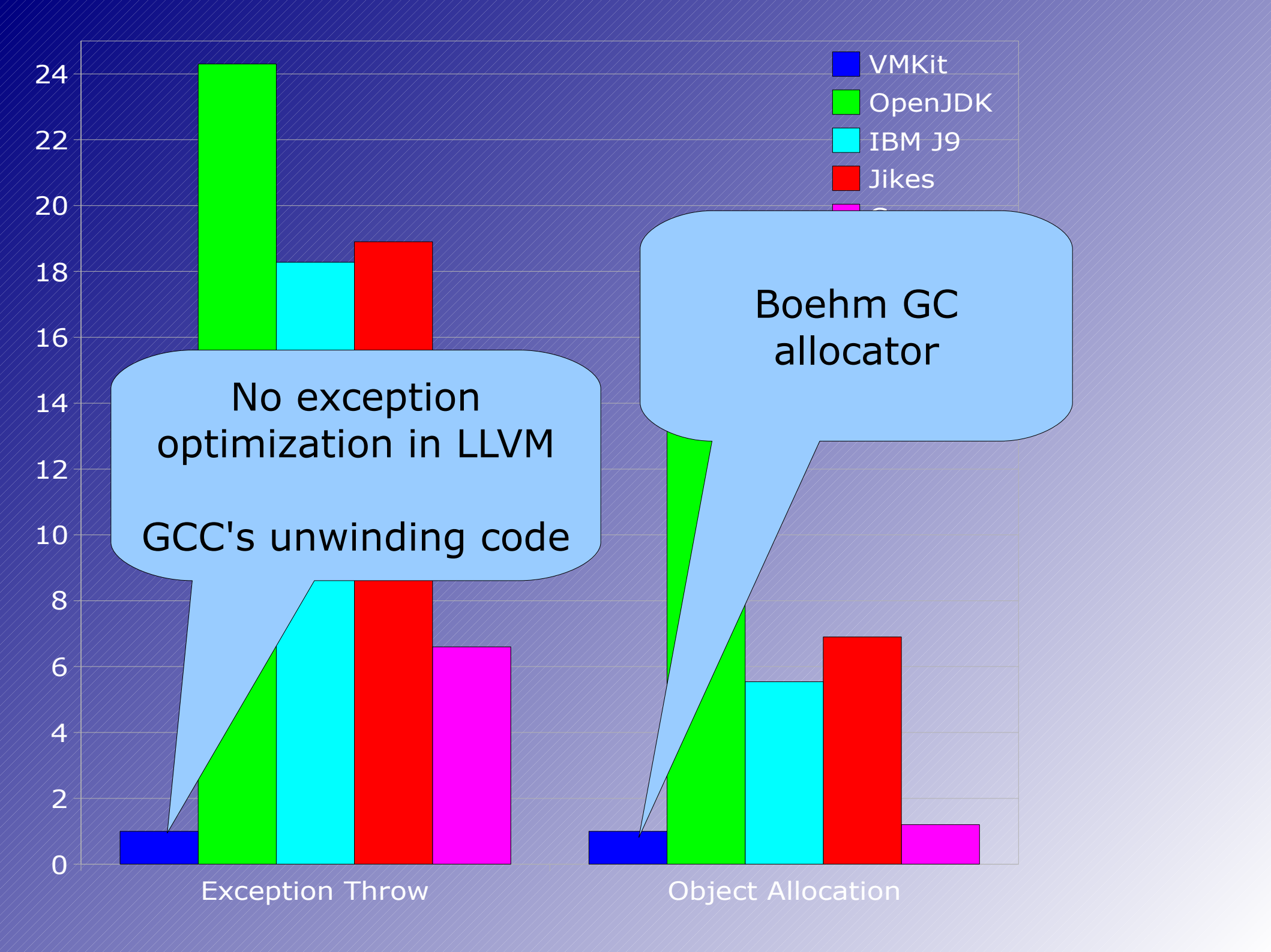
- Athlon XP 1800+, 512M, Linux
- 4 JVMs
  - OpenJDK, IBM J9, Jikes RVM, Cacao
- Java Grande Forum Benchmark
  - Section2: scientific benchmarks
- SPEC JVM98
  - Real-world applications


2.75
2.5
2.25
2
1.75
1.5
1.25
1
0.75
0.5
0.25
0
Series
LUFact
HeapSort
Crypt
FFT
SOR
SparseMatmult
VMKit
OpenJDK
IBM J9
Jikes
Cacao
Escape analysis
Array bounds checks
Relative performance to VMKit (higher is better)


18 18 9
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
VMKit
OpenJDK
IBM J9
Jikes
Cacao
compress
jess
raytrace
db
javac
mtrt
jack


Garbage Collection
(and exceptions)
VMKit
OpenJDK
IBM J9
Jikes
Cacao
18 18 9
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
compress
jess
raytrace
db
javac
mtrt
jack


VMKit
OpenJDK
IBM J9
Jikes
Cacao
0
2
4
6
8
10
12
14
16
18
20
22
24
Exception Throw
Object Allocation


VMKit
OpenJDK
IBM J9
Jikes
24
22
20
18
16
14
12
10
8
6
4
2
0
No exception optimization in LLVM
GCC's unwinding code
Boehm GC allocator
Exception Throw
Object Allocation

What's next

Garbage collection!

We all had fun writing a VM/JIT…

- Cacao, Mono, VMKit, Pnet, Harmony, JamVM, etc

… But now we need a better GC!

- Cacao, Mono, Harmony are developing ones

Garbage collection!

VMKit approach: let's take MMTK

- GC Framework used in JikesRVM
- Well-mature, state of the art GCs
- Apparently designed to be portable across VMs (but written in Java!)

GSoc, student, PhD, anyone: Help ;-)

Adaptive Optimization System!

Second missing feature of VMKit

- LLVM has an interpreter, JIT and AOT
- Even a “fast” JIT!
- But nothing to move from one to another (hotspot detection, on stack replacement, etc)

Oh, well... :)

Research projects

Isolation and Sharing

Isolation:

- Run multiple applications in the same JVM
- Run multiple virtual machines in VMKit (currently Java and .Net)

Sharing

- VM code: Mostly JIT and GC
- Application code: between applications and between class loaders

OSGi Security

What is OSGi?

- A Service-oriented architecture platform
- dummies.com: a better class loading model: one module = one class loader
- Modules execute in the same JVM and communicate with method calls.

Problem: ***for an open-service and universal gateway, what about malicious modules?***

OSGi Security

Our contribution:

- Combine isolate and class loader approach
- Resource accounting *per* module
- Being able to stop a module

Future work:

- Automated resource policies

For more information

http://vmkit.llvm.org

Thank you: Sun Microsystems