

字节码之舞

互联网架构下Java性能优化

陆传胜

chuanshenglu@gmail.com

关于我

- 陆传胜
- 阿里巴巴JVM组
 - AJDK, GC, Virtualization, Profiling
- IBM Java Technology Center
 - OpenJDK, IBMJDK, Harmony

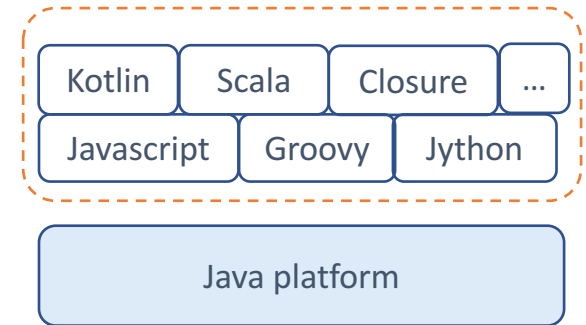


AJDK

Best JDK for cloud computing

尚能饭否？

Nov 2017	Nov 2016	Change	Programming Language	Rankings	Change
1	1		Java	13.231%	-6.52%
2	2		C	9.293%	+0.09%
3	3		C++	5.343%	-0.07%
4	5	▲	Python	4.482%	+0.01%
5	4	▼	C#	3.012%	-0.86%
6	8	▲	JavaScript	2.972%	+0.27%
7	6	▼	Visual Basic .NET	2.909%	-0.26%
8	7	▼	PHP	1.897%	-1.25%
9	16	▲	Delphi/Object Pascal	1.744%	-0.21%
10	9	▼	Assembly language	1.722%	-0.72%



做Java也要做互联网



这次分享将包括

- 性能和成本
- Java性能监控
- 分析和优化

一个JDK开发者的视角

从一段代码说起

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello world!");  
    }  
}
```

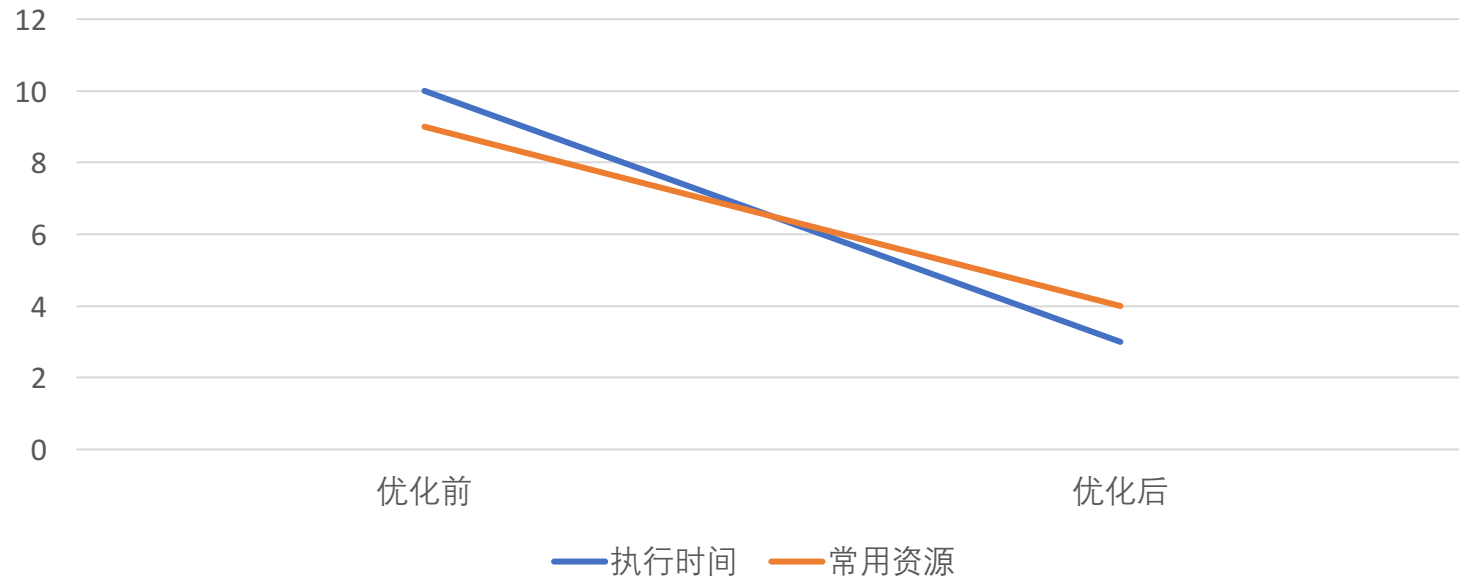
real	0m0.229s
user	0m0.079s
sys	0m0.036s

24 threads

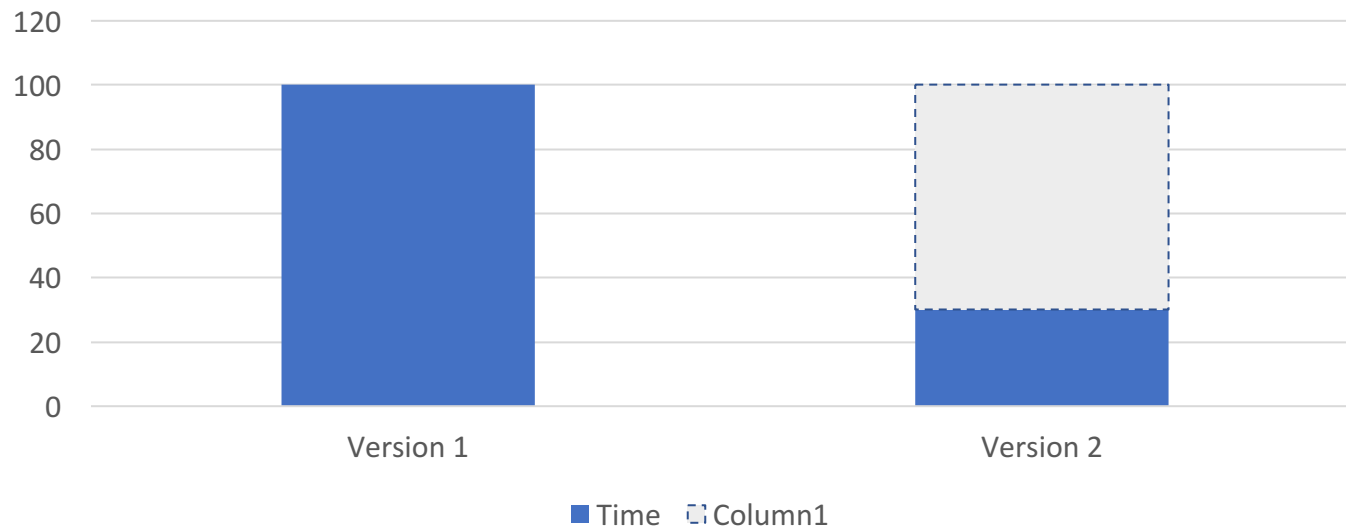
570 context-switches
6,594 page-faults

213,670,272 instructions
132.153222 task-clock
304,667,843 cycles
38,695,465 branches:


期待的性能优化



我最爱的性能优化



收益

- 假设优化一个方法：10ms \rightarrow 4ms
- 0.006秒 * 100次/分 * (60 * 24 * 365) = 315360秒
- * 10000 台 = 876000 小时
- * 135w / 1000 * 0.35 元 /kwh = 4,1391 元
- 一线城市0.5m² !
- 那么，如果我们每天优化掉一个方法.....

成本



网络 100 Mbps * M
¥ 7200/月

电费
0.35/kwh

DDR4 8g * N
¥ 800

Xeon E5 2620 8c * N
¥ 12499



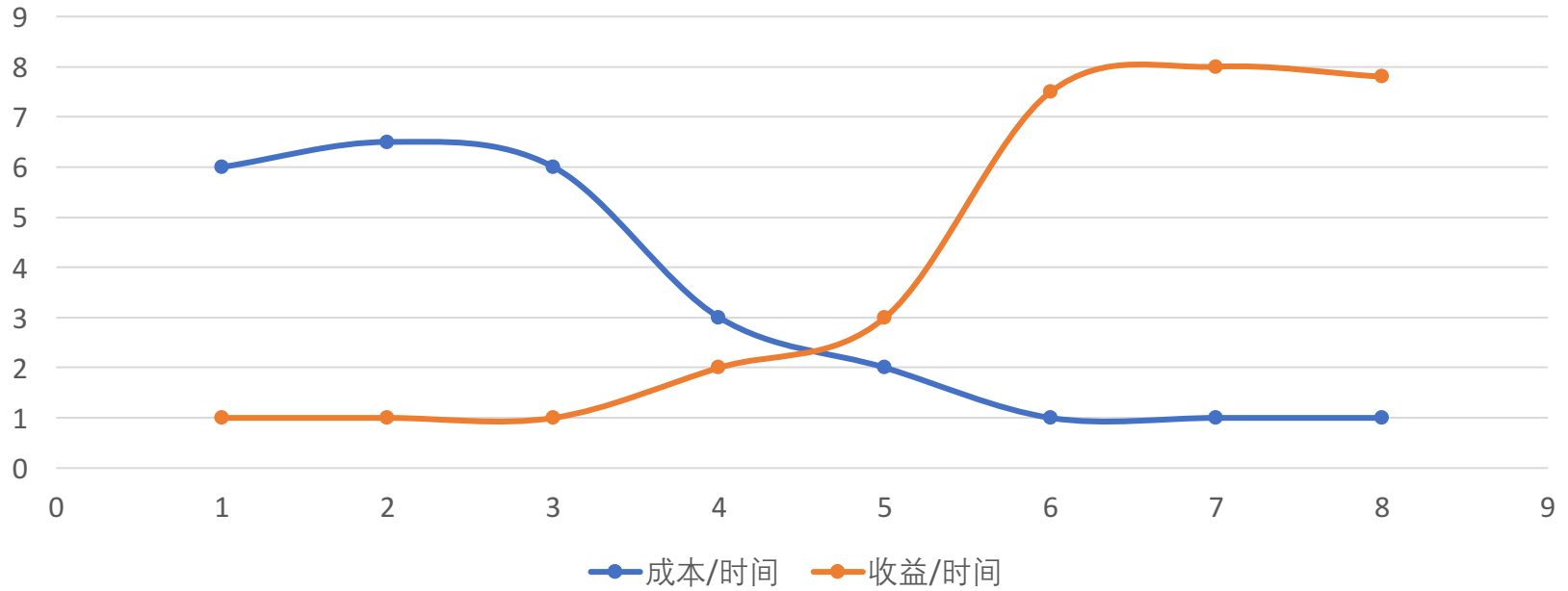
Salary \$48,905 -
\$102,053

Bonus \$0.00 - \$10,116

Profit sharing \$-0.48 -
\$10,022

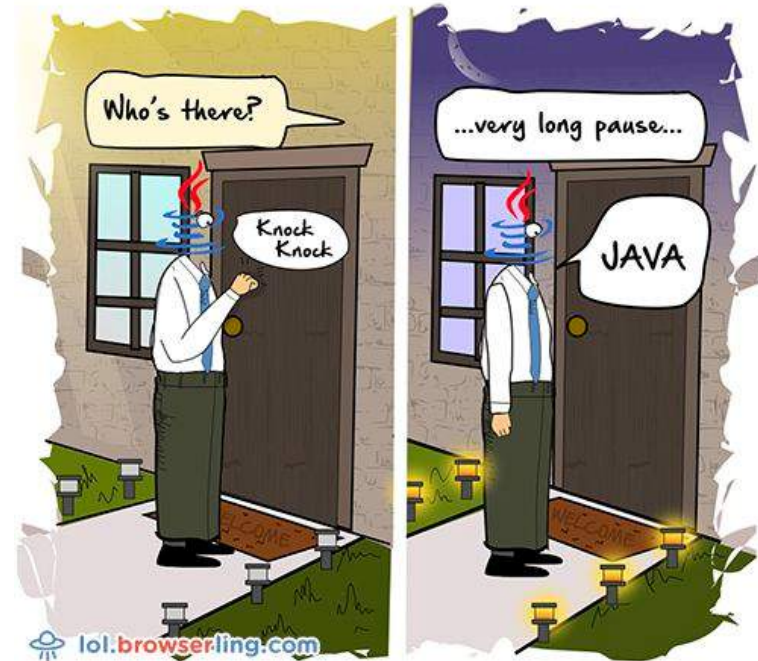


5个月内推到重写...
9-9-6工作节奏...
延期两个月上线...
推迟一年晋升...
提前半年跳槽...



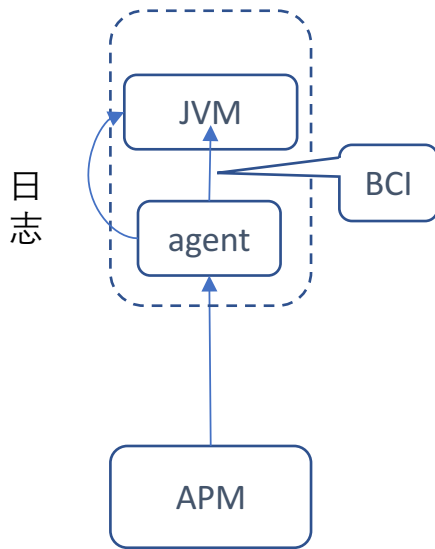
Java和性能

- 性能对任何语言都是关键
- 对于OpenJDK和Oracle
- 性能有很多内涵

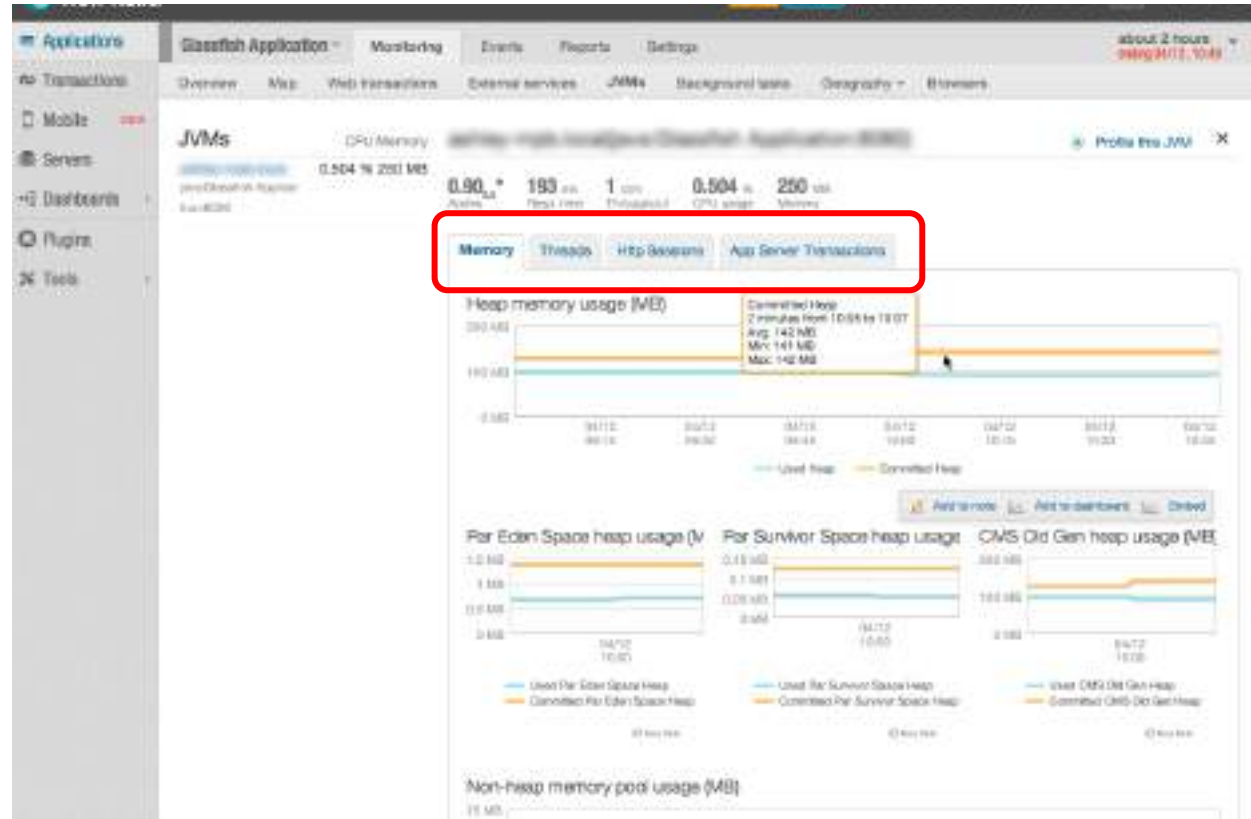


<https://www.pinterest.com/pin/433119689144259039/>

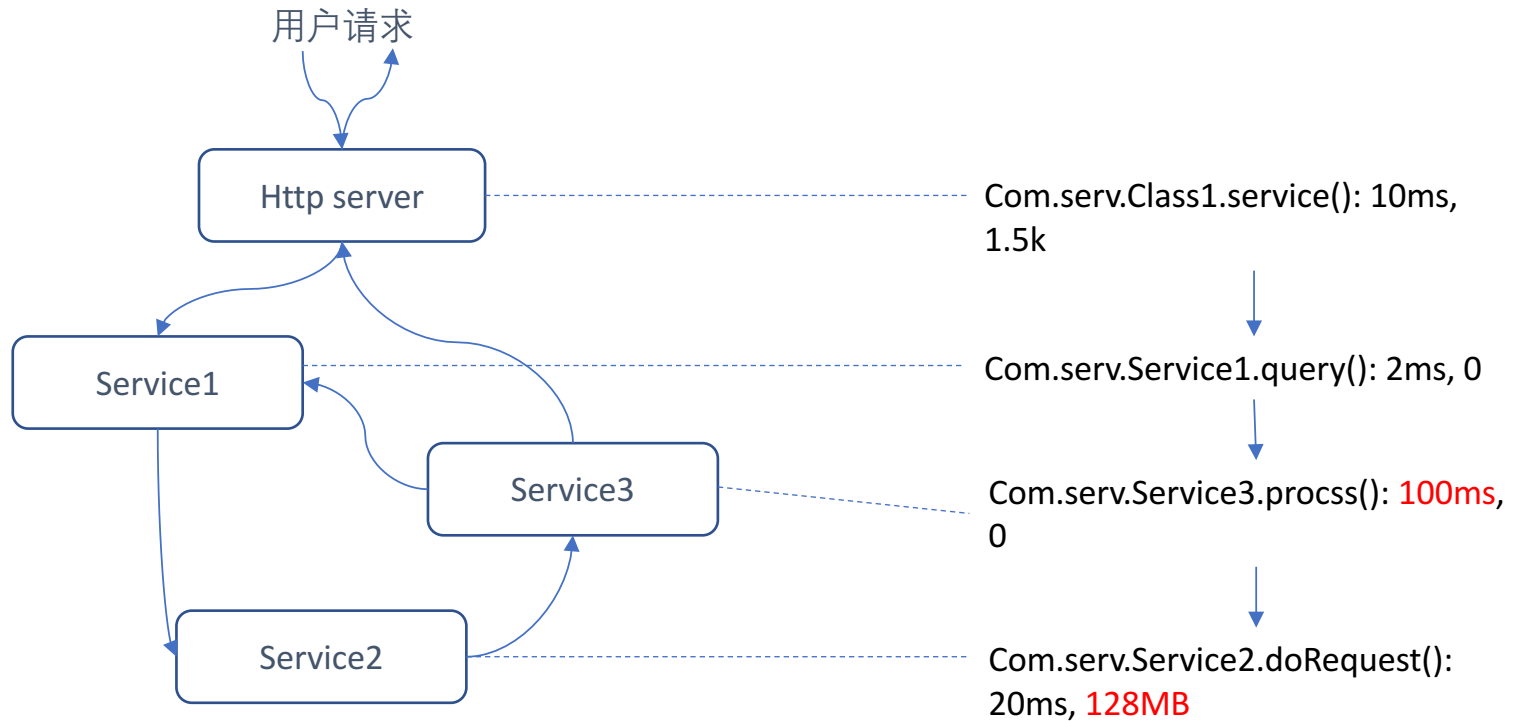
Java性能监控



- 垃圾回收器
- 类加载和回收
- 堆内、堆外内存分配
- 线程profiling
- 集群视图

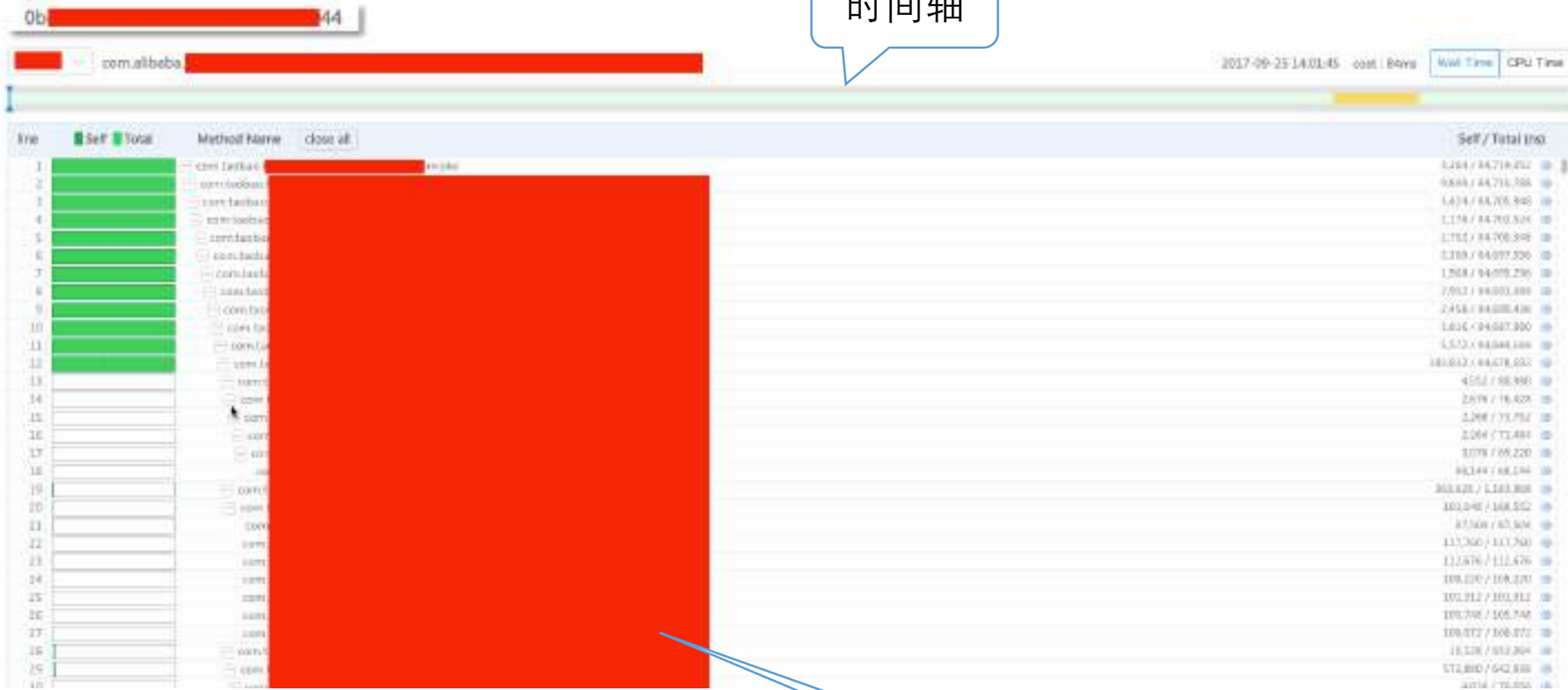


Java Live Profiler



业务ID

时间轴

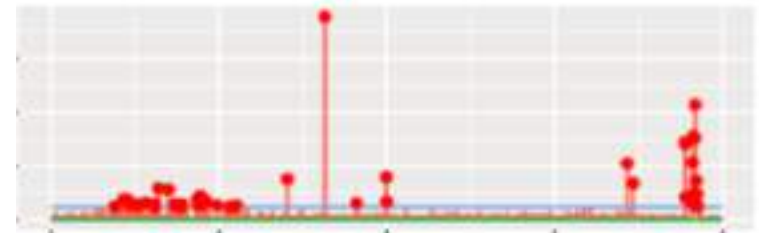
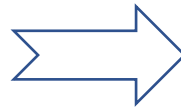


Java方法
调用链

水位



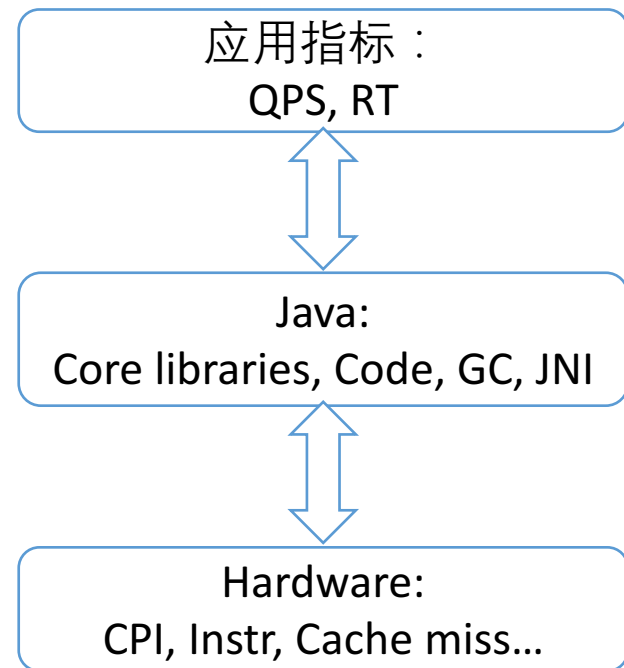
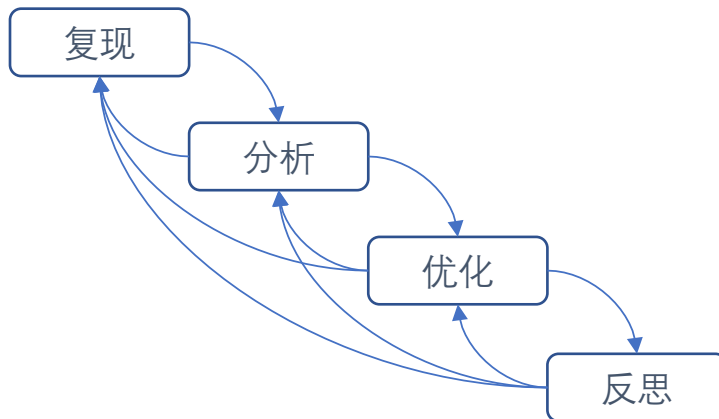
从报警到预警



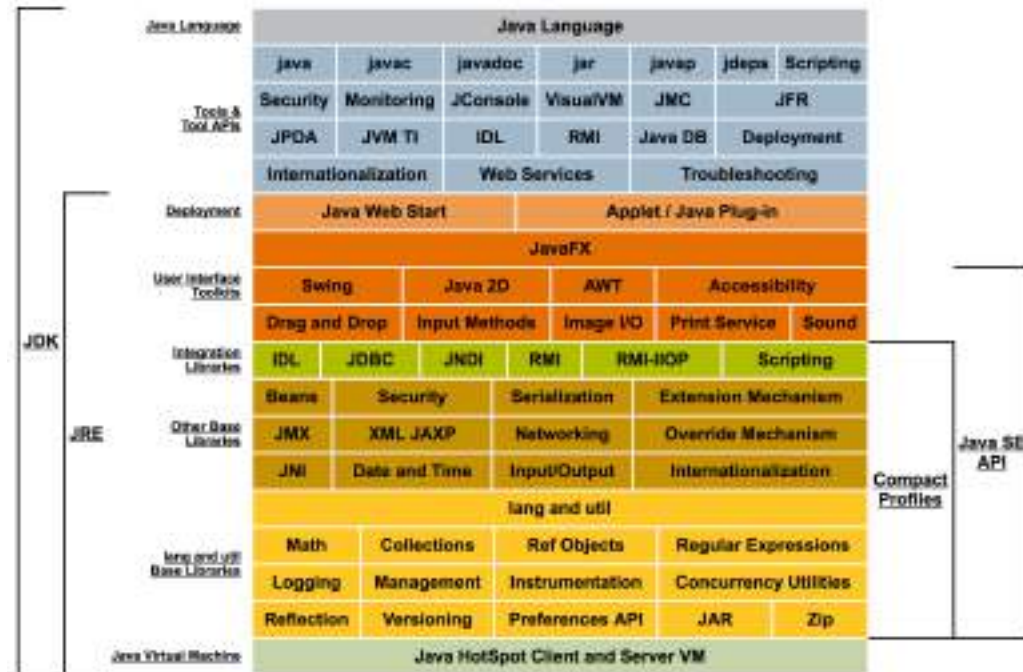
Q: 下一次Full GC会发生在什么时候?

Java性能优化

- 方法论的优化
- 性能优化
 - 主动
 - 被动



Java层级

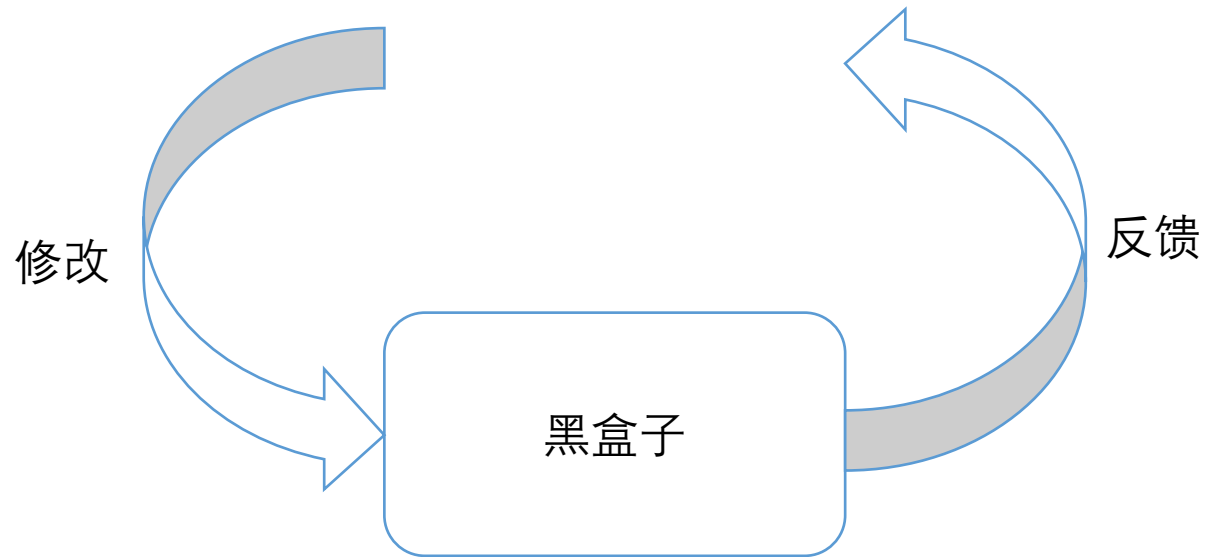


GC优化

- 典型互联网企业的Java Heap
 - 大量微服务，Heap不是特别大， < 100GB
 - 应用类型较多
 - 敏感型：Throughput敏感; Response Time 敏感
 - 不敏感型：什么都不敏感...
 - 业务变化较快
 - 一般会有一定容错性



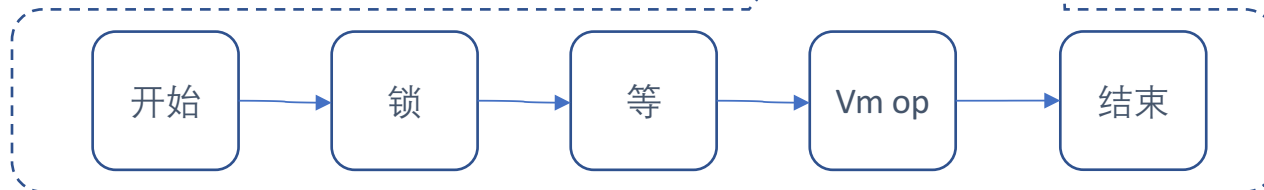
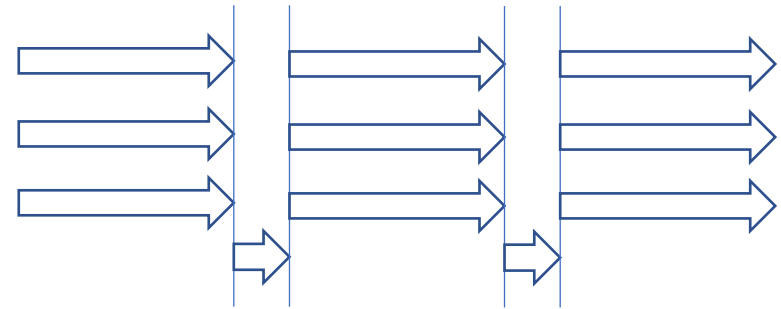
GC优化



Gc优化—停顿

- 不光GC才会导致暂停
- 但是大部分GC都会暂停
- HotSpot JVM的执行节奏
- Safepoint

-XX:+PrintGCApplicationStoppedTime
-XX:+PrintSafepointStatistics
-XX:+TraceSafepointCleanupTime



GC优化

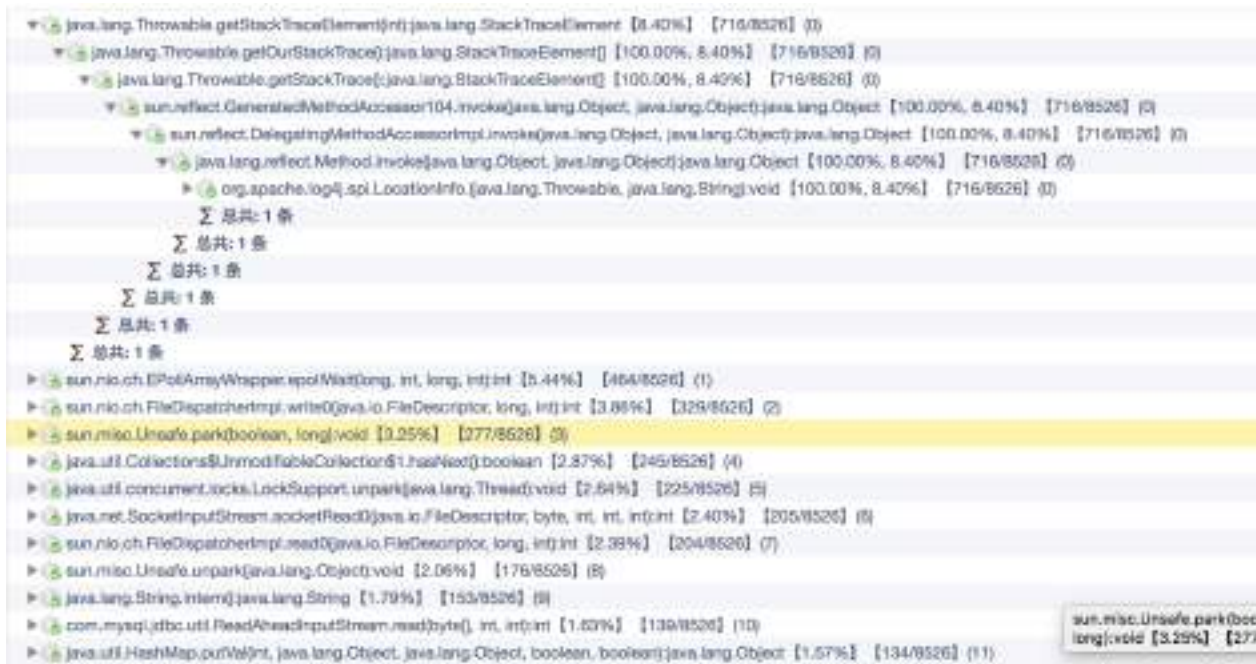
- Case study:

2015-12-20T16:09:04.088-0800: 95.743: [GC pause (G1 Evacuation Pause) (young) (initial-mark) 8258M->6294M(10G), 0.1343256 secs]

2015-12-20T16:09:08.257-0800: 99.912: Total time for which application threads were stopped: 4.1692476 seconds

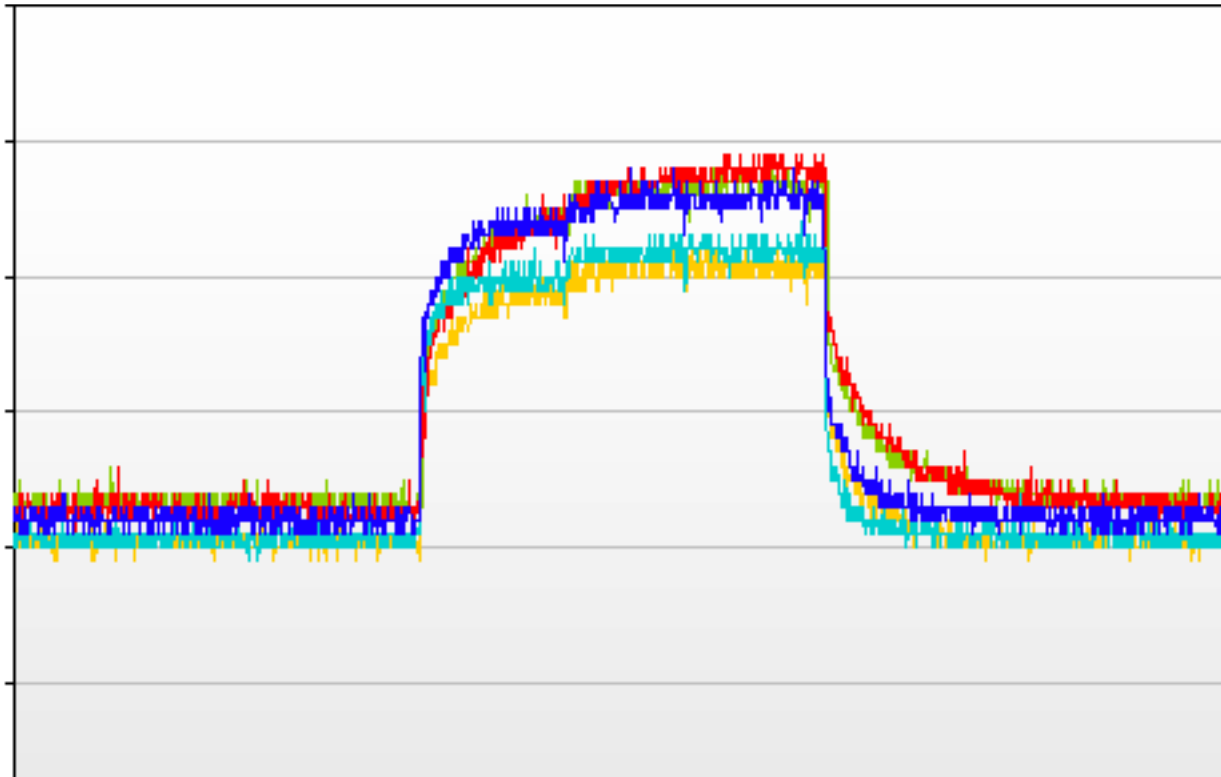
代码效率

- JIT (Just In Time Compiler)
- 热点方法



代码效率

- 编译的代价



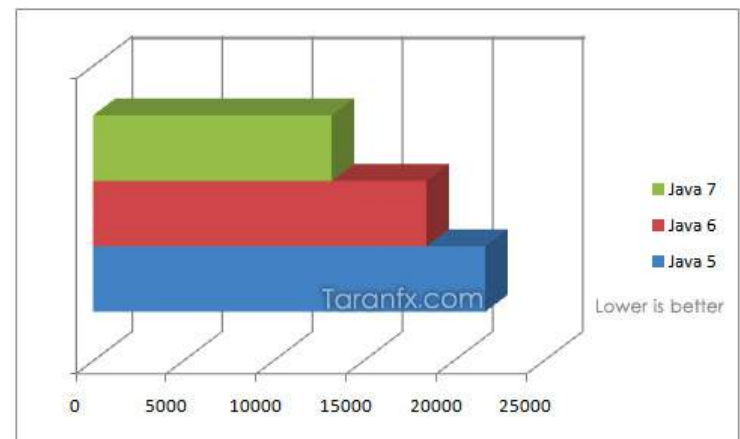
代码效率

- 编程模型的改进
 - 并发
 - 异步

```
com.alibaba.wisp.engine.WispEngine.dispatch()-> {  
    System.out.println("Hello world");  
}
```

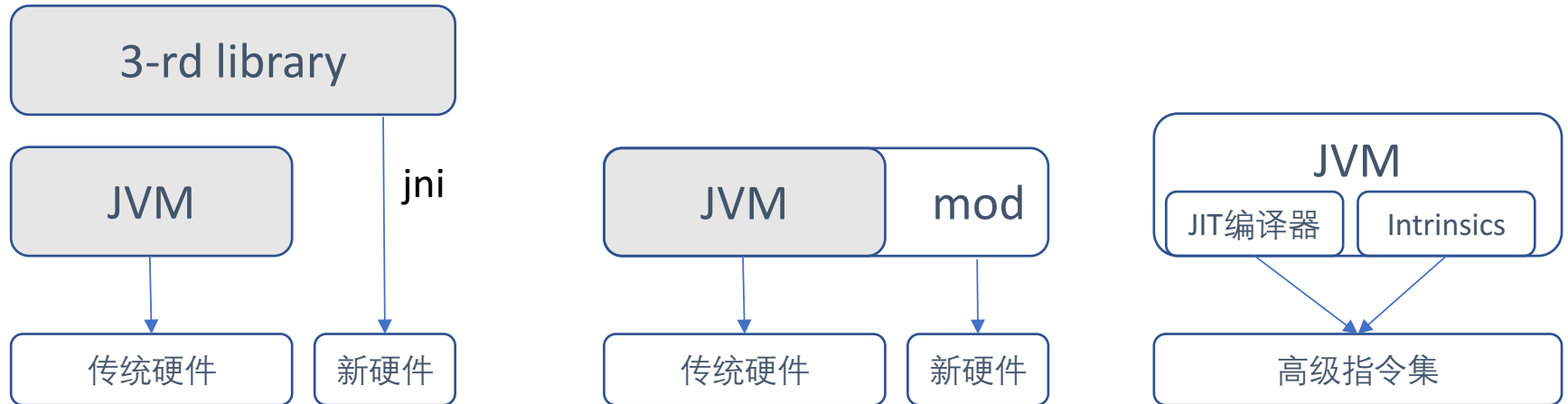
升级之路

- 收益
 - 性能提升
 - 稳定性
 - 新特性
- 代价
 - 开发
 - 踩坑



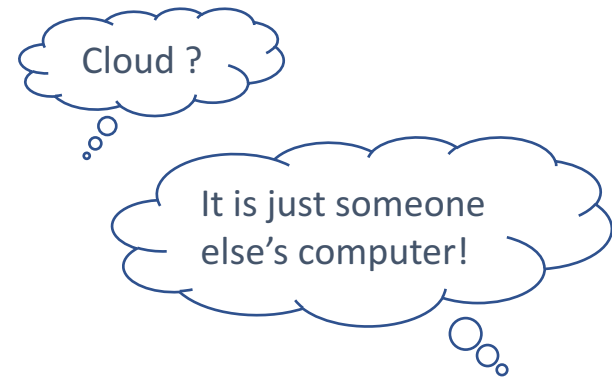
<http://geeknizer.com/java-7-whats-new-performance-benchmark-1-5-1-6-1-7/>

硬件辅助优化



面向基础设施

- Java的挑战——cloud
 - 不可控
 - 不透明

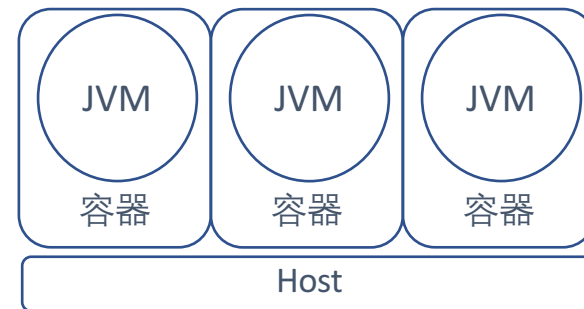


AJDK

Best JDK for cloud computing!

面向基础设施

- Java的挑战——容器
 - 对容器的感知
 - 调度导致的不确定性
 - 共享的资源和部署密度
- Java感知容器
- Java支援调度



JEP: container aware Java

AJDK

Best JDK for cloud computing!

展望未来

- 云上的Java
- 更快的Java平台迭代
- 新硬件新平台
- Java语言的更新

I had my time, believe me.
Sooner or later they will
call you slow, verbose,
old fashioned...



Daniel Stori (turnoff.us)

<https://dzone.com/articles/lang-buddies-comic>

Thank you!

AJDK

Best JDK for cloud computing!

