JVM优化工具

1. 打印对象内存

```
Integer i = new Integer(23);
System.out.println(ClassLayout.parseInstance(i)
            .toPrintable());
```

需要引入jar包：

```
<dependency>
    <groupId>org.openjdk.jol</groupId>
    <artifactId>jol-core</artifactId>
    <version>0.10</version>
</dependency>
```

打印结果：

```
Running 64-bit HotSpot VM.
Objects are 8 bytes aligned.

java.lang.Integer object internals:
OFFSET  SIZE   TYPE DESCRIPTION                    VALUE
  0     4        (object header)                01 00 00 00 (00000001 00000000 00000000 00000000) (1)
  4     4        (object header)                00 00 00 00 (00000000 00000000 00000000 00000000) (0)
  8     4        (object header)                48 33 36 97 (01001000 00110011 00110110 10010111) (-1758055608)
 12     4        (object header)                01 00 00 00 (00000001 00000000 00000000 00000000) (1)
 16     4    int Integer.value                  23
 20     4        (loss due to the next object alignment)

Instance size: 24 bytes (reported by Instrumentation API)
Space losses: 0 bytes internal + 4 bytes external = 4 bytes total
```

2. Jconsole

3. JOL

Code Tools: jol

更多

JVM 监控以及内存分析 | jcmd | GC时间