



表4–2 jstat工具主要选项

选 项	作 用
-class	监视类加载、卸载数量、总空间以及类装载所耗费的时间
-gc	监视 Java 堆状况，包括 Eden 区、2 个 Survivor 区、老年代、永久代等的容量，已用空间，垃圾收集时间合计等信息
-gccapacity	监视内容与 -gc 基本相同，但输出主要关注 Java 堆各个区域使用到的最大、最小空间
-gcutil	监视内容与 -gc 基本相同，但输出主要关注已使用空间占总空间的百分比
-gccause	与 -gcutil 功能一样，但是会额外输出导致上一次垃圾收集产生的原因
-gcnew	监视新生代垃圾收集状况
-gcnewcapacity	监视内容与 -gcnew 基本相同，输出主要关注使用到的最大、最小空间
-gcold	监视老年代垃圾收集状况
-gcoldcapacity	监视内容与 -gcold 基本相同，输出主要关注使用到的最大、最小空间
-gcpermcapacity	输出永久代使用到的最大、最小空间
-compiler	输出即时编译器编译过的方法、耗时等信息
-printcompilation	输出已经被即时编译的方法

表4–3 jmap工具主要选项

选 项	作 用
-dump	生成 Java 堆转储快照。格式为 -dump:[live,]format=b,file=<filename>，其中 live 子参数说明是否只 dump 出存活的对象
-finalizerinfo	显示在 F-Queue 中等待 Finalizer 线程执行 finalize 方法的对象。只在 Linux/Solaris 平台下有效
-heap	显示 Java 堆详细信息，如使用哪种回收器、参数配置、分代状况等。只在 Linux/Solaris 平台下有效
-histo	显示堆中对象统计信息，包括类、实例数量、合计容量
-permstat	以 ClassLoader 为统计口径显示永久代内存状态。只在 Linux/Solaris 平台下有效
-F	当虚拟机进程对 -dump 选项没有响应时，可使用这个选项强制生成 dump 快照。只在 Linux/Solaris 平台下有效

表4–4 jstack工具主要选项

选 项	作 用
-F	当正常输出的请求不被响应时，强制输出线程堆栈
-l	除堆栈外，显示关于锁的附加信息
-m	如果调用到本地方法的话，可以显示 C/C++ 的堆栈

```
<%@ page import="java.util.Map"%>

<html>
<head>
<title>服务器线程信息</title>
</head>
<body>
<pre>
<%
for (Map.Entry<Thread, StackTraceElement[]> stackTrace : Thread.getAllStack-Traces().entrySet()) {
    Thread thread = (Thread) stackTrace.getKey();
    StackTraceElement[] stack = (StackTraceElement[]) stackTrace.getValue();
    if (thread.equals(Thread.currentThread())) {
        continue;
    }
    out.print("\n线程： " + thread.getName() + "\n");
    for (StackTraceElement element : stack) {
        out.print("\t"+element+"\n");
    }
}
%>
</pre>
</body>
</html>
```