性能日志分析表格各列含义

* Self = Time spend inside the code of that function itself.
* 在该函数本身的代码内花费的时间。
* Cum. = Time spend in functions Self calls with it's self time (short for Cumulative).

函数上花费的时间带有自身时间的Self调用（累积时间的缩写）。

* Avg vs. Total: Average is average time per call, Total is the total time spend in all calls.

[share](https://stackoverflow.com/a/10091979" \o "short permalink to this answer)  [improve this answer](https://stackoverflow.com/posts/10091979/edit" \o ")  follow

三种不同的profile

# Profiler：为什么需要这三种不同的类型

三种不同的Java分析器：

1. ****标准JVM探查器****，可[跟踪](https://stackify.com/jvm-metrics/)****JVM的****[每个细节](https://stackify.com/jvm-metrics/)（CPU，线程，内存，垃圾回收等）。
2. ****轻量级的探查器****，通过一些抽象突出显示您的应用程序。
3. 用于监视生产环境中的应用程序的应用****程序性能管理（**[APM](https://stackify.com/what-is-apm/)**）工具****。

## 轻量级Java事务分析器

XRebel和Stackify **[Prefix等产品](https://stackify.com/prefix/)**。

轻量级探查器通过将自身直接注入代码中来采用不同的方法来跟踪您的应用程序。

* Aspect Profilers使用面向方面的编程（AOP）将代码注入指定方法的开头和结尾。注入的代码可以启动计时器，然后在方法完成时报告经过的时间。这些分析器很容易设置，但是您需要知道要分析的内容。有关示例，请参见[Spring AOP方法分析](http://www.journaldev.com/7801/spring-aop-method-profiling-beginners-example" \t "https://stackify.com/java-profilers-3-types/_blank)。
* Java代理探查器使用[Java Instrumentation API](https://docs.oracle.com/javase/7/docs/api/java/lang/instrument/package-summary.html" \t "https://stackify.com/java-profilers-3-types/_blank)将代码注入您的应用程序。由于代码是在字节码级别重写的，因此该方法对您的应用程序具有更大的访问权限。这允许对应用程序中运行的所有代码进行检测-无论是您编写的代码还是应用程序所依赖的第三方库。查看[Java代理简介，](https://stackify.com/what-are-java-agents-and-how-to-profile-with-them/" \t "https://stackify.com/java-profilers-3-types/_blank)以了解其工作原理。

## 低开销的Java JVM生产环境分析（APM）

到目前为止，所有分析器都非常适合开发，但是跟踪系统在生产中的性能至关重要。

生产始终是不同的环境-开发和暂存设置通常没有相同的数据集和负载。

Java APM工具通常使用Java Agent Profiler方法，但具有不同的检测规则，以使其运行而不会影响生产性能。这些探查器的窍门是以一种聪明的方式提供正确的信息，以不占用CPU周期。

## 为什么您的应用程序需要所有3个Profiler？

标准分析器擅长在开发阶段发现性能问题。但是生产是另一种情况。您的应用程序的行为可能会根据传入流量，服务器请求，响应以及许多其他情况而发生变化。



那么，生产环境的解决方案是什么？APM工具就是答案。APM工具针对生产环境，并提供有关应用程序性能的报告。

因此，标准分析器有助于开发，而APM工具则有助于生产。您可能想知道轻量级事务分析器的需求是什么。

好吧，轻量级的探查器采用了不同的代码配置方法。他们直接将自己插入代码中，尤其是在方法的开始和结束处。

它们还易于设置，并且消耗相对较少的资源。这对于使用硬件事务存储器的应用程序非常有用。这些应用程序需要完善的分析工具，这些工具必须准确指出导致性能问题的方法或功能。

毫无疑问，我们可以说，如果您要开发复杂的应用程序，则将需要所有三个事件探查器。每种探查器类型都有一种独特的方法来检查应用程序的性能问题。

[RebelLab的调查](https://jrebel.com/rebellabs/developer-productivity-report-2015-java-performance-survey-results/" \t "https://stackify.com/java-profilers-3-types/_blank)还显示，大多数公司使用多个代码分析器来查找其应用程序中的性能问题。