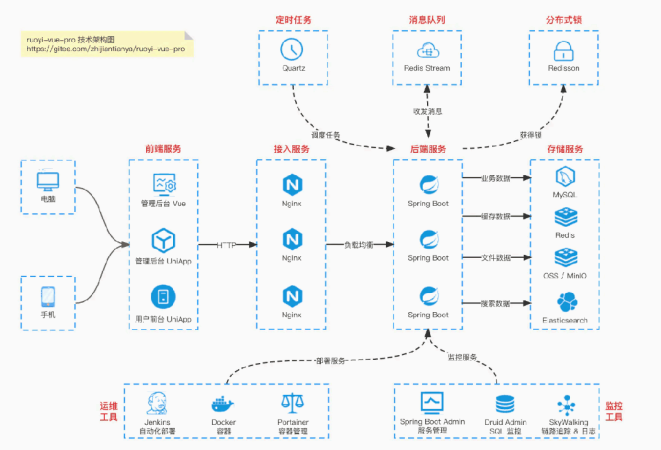
# 高性能、无侵入的 Java 性能监控神器

Java基基 2024年10月29日 11:55 上海








来源：github.com/LinShunKang/MyPerf4J

- **背景**

## 背景

随着所在公司的发展，应用服务的规模不断扩大，原有的垂直应用架构已无法满足产品的发展，几十个工程师在一个项目里并行开发不同的功能，开发效率不断降低。

于是公司开始全面推进服务化进程，把团队内的大部分工程师主要精力全部都集中到服务化中。服务化可以让每个工程师仅在自己负责的子项目中进行开发，提高了开发的效率，但是服务化同时也带来了其他问题：

- 无法知道每个服务的运行情况，例如，某一台服务它目前的 QPS 是多少？它的平均延迟是多少，99% 的延迟是多少，99.9% 的延迟又是多少？
- 某一个接口响应时间慢，如何定位是哪个方法引起的？
- 每个服务的负载是否均衡？
- 当服务出现抖动时，如何判断是 DB、Cache 还是下游服务引起的？
- DB 和 Cache 响应延迟是多少？
- 如何评估服务的容量，随着服务的调用量越来越大，这个服务需要多少机器来支撑？什么时候应该加机器？

针对以上开发中的烦恼，今天我们介绍一个针对高并发、低延迟应用设计的高性能 Java 性能监控和统计工具——**MyPerf4J** 。

基于 Spring Boot + MyBatis Plus + Vue & Element 实现的后台管理系统 + 用户小程序，支持 RBAC 动态权限、多租户、数据权限、工作流、三方登录、支付、短信、商城等功能

- 项目地址：https://github.com/YunaiV/ruoyi-vue-pro
- 视频教程：https://doc.iocoder.cn/video/

## 使用场景

- 在开发环境中快速定位 Java 应用程序的性能瓶颈
- 在生产环境中长期监控 Java 应用程序的性能指标

基于 Spring Cloud Alibaba + Gateway + Nacos + RocketMQ + Vue & Element 实现的后台管理系统 + 用户小程序，支持 RBAC 动态权限、多租户、

数据权限、工作流、三方登录、支付、短信、商城等功能

- 项目地址：https://github.com/YunaiV/yudao-cloud
- 视频教程：https://doc.iocoder.cn/video/

## 特性

- 高性能：单线程支持每秒 1000万次 响应时间的记录，每次记录只花费 73纳秒
- 无侵入：采用 JavaAgent 方式，对应用程序完全无侵入，无需修改应用代码
- 低内存：采用内存复用的方式，整个生命周期只产生极少的临时对象，不影响应用程序的GC
- 高精度：采用纳秒来计算响应时间
- 高实时：支持秒级监控，最低 1 秒!

## 监控指标

MyPerf4J 为每个应用收集数十个监控指标，所有的监控指标都是实时采集和展现的。

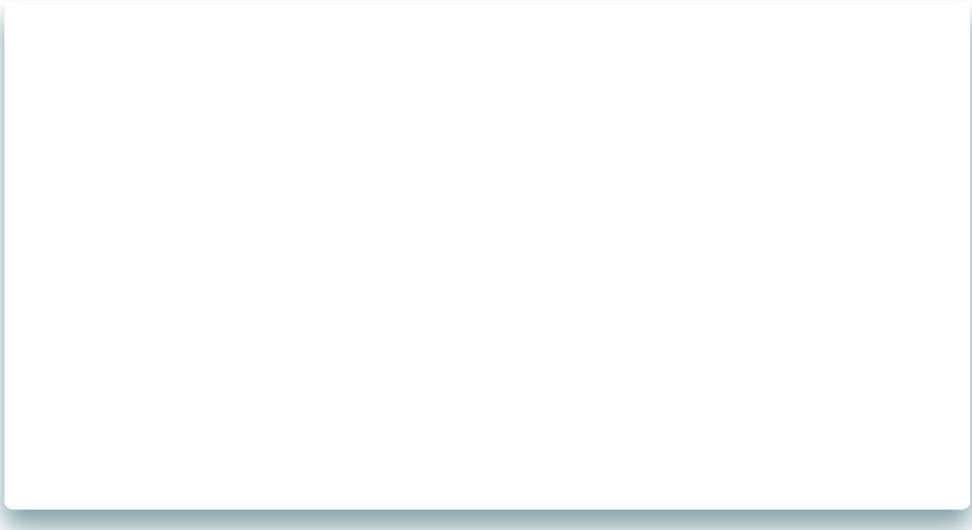
下面是 MyPerf4J 目前支持的监控指标列表：

### Method

RPS，Count，Avg，Min，Max，StdDev，TP50，TP90，TP95，TP99，TP999，TP9999，TP99999，TP100

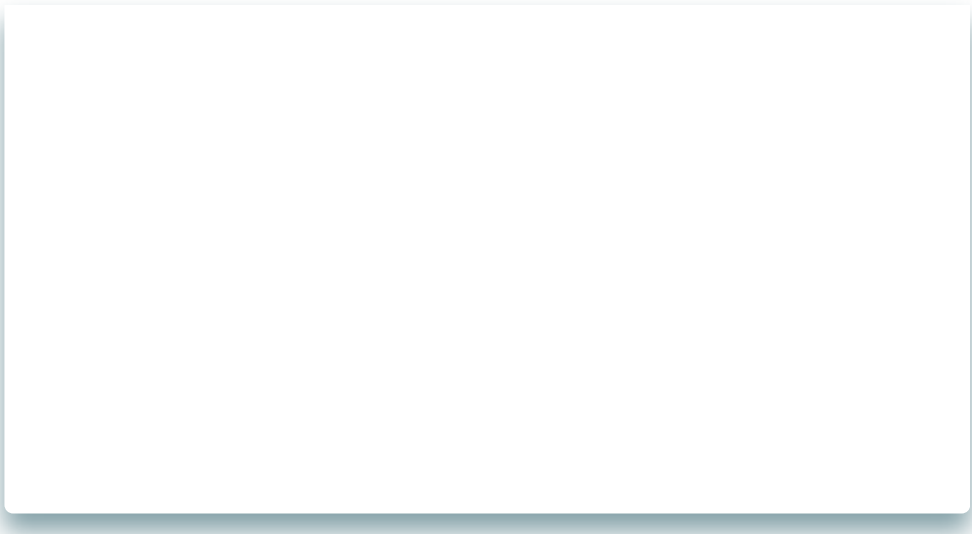
### Memory

HeapInit，HeapUsed，HeapCommitted，HeapMax，NonHeapInit，NonHeapUsed，NonHeapCommitted，NonHeapMax



## JVM GC

```
CollectCount, CollectTime
```



## JVM Class

```
Total, Loaded, Unloaded
```

## 快速启动

MyPerf4J 采用 JavaAgent 配置方式，透明化接入应用，对应用代码完全没有侵入。

### 下载

- 下载并解压 MyPerf4J-ASM.zip
- 阅读解压出的 README 文件
- 修改解压出的 MyPerf4J.properties 配置文件中 `app_name` 、 `metrics.log.xxx` 和 `filter.packages.include` 的配置值

- **MyPerf4J-ASM.zip包**：https://github.com/LinShunKang/Objects/blob/master/zips/CN/MyPerf4J-ASM-3.3.0-SNAPSHOT.zip?raw=true

### 配置

在 JVM 启动参数里加上以下两个参数

```
-javaagent:/path/to/MyPerf4J-ASM.jar
-DMyPerf4JPropFile=/path/to/MyPerf4J.properties
```

### 运行

启动应用，监控日志输出到 `/path/to/log/method_metrics.log`：

```
MyPerf4J Method Metrics [2020-01-01 12:49:57, 2020-01-01 12:49:58]
Method[6]                              Type        Level  TimePercent      RPS  Avg(ms)  Min(m
DemoServiceImpl.getId2(long)        General      Service      322.50%     6524     0.49
DemoServiceImpl.getId3(long)        General      Service      296.10%     4350     0.68
DemoServiceImpl.getId4(long)        General      Service      164.60%     2176     0.76
DemoServiceImpl.getId1(long)        General      Service        0.00%     8704     0.00
DemoDAO.getId1(long)             DynamicProxy        DAO        0.00%     2176     0.00
DemoDAO.getId2()                 DynamicProxy        DAO        0.00%     2176     0.00
```

### 卸载

在 JVM 启动参数中去掉以下两个参数，重启即可卸载此工具。

```
-javaagent:/path/to/MyPerf4J-ASM.jar
-DMyPerf4JPropFile=/path/to/MyPerf4J.properties
```

## 开源地址

- https://github.com/LinShunKang/MyPerf4J

欢迎加入我的知识星球，全面提升技术能力。

👉 加入方式，**“长按”或“扫描”下方二维码噢**：

星球的**内容包括**：项目实战、面试招聘、源码解析、学习路线。

**文章有帮助的话，在看，转发吧。**
**谢谢支持哟（*^__^*）**

阅读原文