



链滴

Pepper Metrics - Spring/Spring Boot 应用性能监控利器

作者: [Lord-X](#)

原文链接: <https://ld246.com/article/1567742861333>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



关于项目

Pepper Metrics是我与同事开发的一个开源工具(<https://github.com/zrbcool/pepper-metrics>), 通过收集jedis/mybatis/httpServlet/dubbo/motan的运行性能统计, 并暴露成prometheus等主流序数据库兼容数据, 通过grafana展示趋势。其插件化的架构也非常方便使用者扩展并集成其他开源件。

请大家给个star, 同时欢迎大家成为开发者提交PR一起完善项目。

Architecture

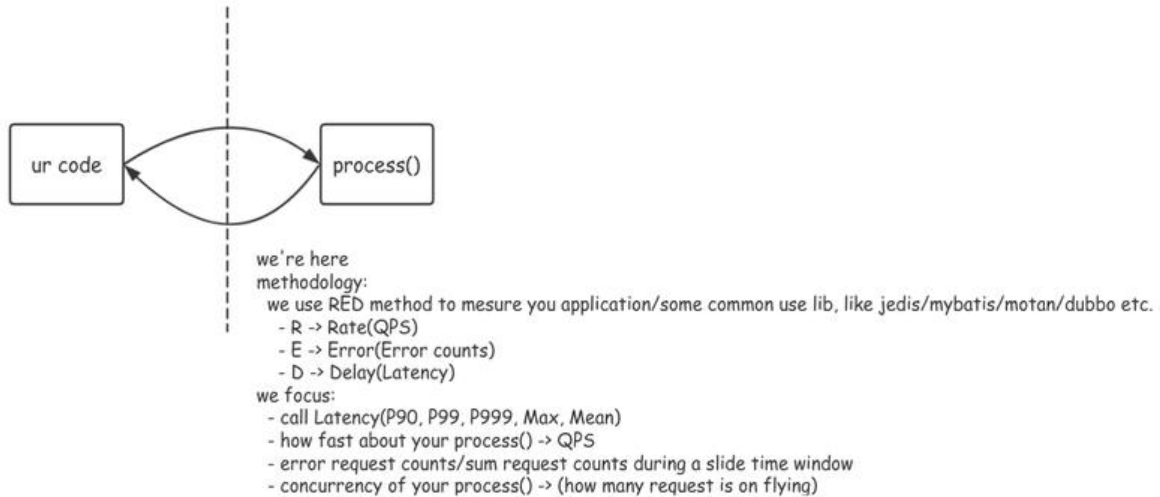
Pepper Metrics项目从核心上来说, 基于Tom Wilkie的RED理论, 即对每个服务 (这里的服务特指进程中的某种调用, 比如调用一次数据库查询) 进行RED指标收集, 包括:

- Rate (请求速率一般指QPS)
- Errors (错误数或单位时间窗口内的错误率)
- Duration (请求消耗的时间一般以PXX的百分位时间表示, 比如P99=100ms代表百分之九十九的请求耗时在X毫秒内)

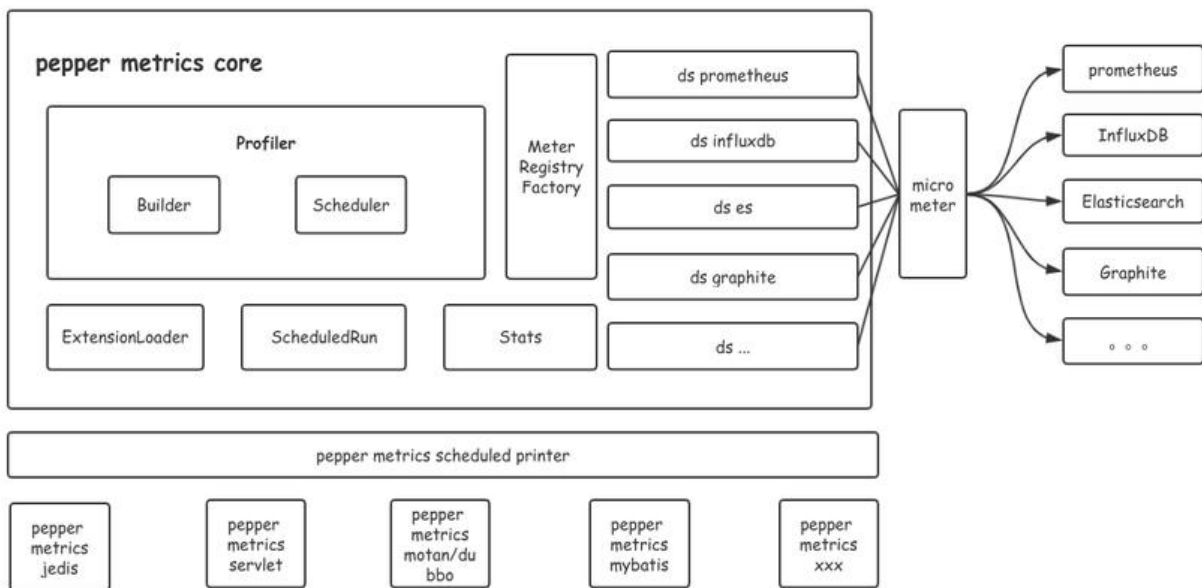
上面简述了Pepper Metrics项目的核心思想及方法论依据, 而从技术上来说, Pepper Metrics项目建了一套完整的可插拔插件体系, 使应用可以基于选用的组件 (如RPC通信框架dubbo, motan、OM对象模型关系映射框架mybatis、标准的HTTP Servlet组件、Redis操作库jedis、等) 选择现有的件扩展直接具备上述指标的:

- 收集
- 打印 (基于标准格式设计并基于slf4j定时输出于日志)
- 输出 (针对多种数据库, 默认以prometheus实现, 将指标输出到prometheus中)
- 可视化 (基于grafana开发的dashboard, 默认以prometheus作为数据源)

Concept



Architecture



各个组件说明

- Profiler, 核心部分, 用于启动定期调度任务, 并通过ExtensionLoad加载所有的ScheduledRun扩展, 按照指定周期发起调度。同时内部维护Stats的构造器Profiler.Builder
- Scheduler, 虚拟概念, 在Profiler作为一个定时任务存在
- ExtensionLoader, 非常重要的组件, 通过Java SPI机制加载插件, 使项目的各个模块可以灵活插拔, 也是项目架构的基石
- ScheduledRun, 扩展点: pepper metrics core会定时调度, 传递所有的Stats, 实现插件可使用Stats当中收集到的性能数据, 目前已实现的为scheduled printer组件
- MeterRegistryFactory, 扩展点: 基于不同的micrometer的Registry实现抽象并屏蔽各个数据的差异

- Pepper Metrics X, 具体的集成, 我们的目标是度量一切, 目前计划实现的为: jedis, motan, d bbo, servlet, mybatis等最常用组件

写在最后

- 项目WIKI

<https://github.com/zrbcool/pepper-metrics/wiki>

- 快速入门

<https://github.com/zrbcool/pepper-metrics/wiki/ZH-Quick-Start>

- 快速DEMO(十分钟本地启动全套模拟应用+grafana+prometheus+内置dashboard)

<https://github.com/zrbcool/pepper-metrics-demo>

- 在线DEMO

<http://pepper-metrics.zrbcool.top>