

利用Tokio实现一个高性能Mini-Http-Server

Web 框架建议选用 Axum

苏林

自我介绍

- 前折800互联网研发团队负责人, 10余年一线研发经验
- 目前是多点Dmall技术Leader
- 具有多年的软件开发经验, 熟悉Ruby、Java、Rust等开发语言
- 同时也参与过Rust中文社区日报维护工作.

分享内容

- 高性能Http Server实战课程开篇
- 回顾Tokio是什么
- 利用Tokio实现一个Mini Http Server
- 压测Mini Http Server
- Rust web框架生态介绍及推荐使用Axum框架

期望公开课达到的目的

- 1、实战: 学会从0开始设计架构短链服务.
- 2、Http Server的本质是什么.
- 3、了解Rust Web框架生态, 为大家技术选型提供一些参考.

高性能Http Server课程开篇

用Tokio来实现一个短链系统

业务背景

<http://gk.link/a/xxx>

<http://gk.link/xkjfsljk/>

[32839283?utm_source=xxxxxx&kfds&jfds](http://gk.link/32839283?utm_source=xxxxxx&kfds&jfds)

涉及的技术点

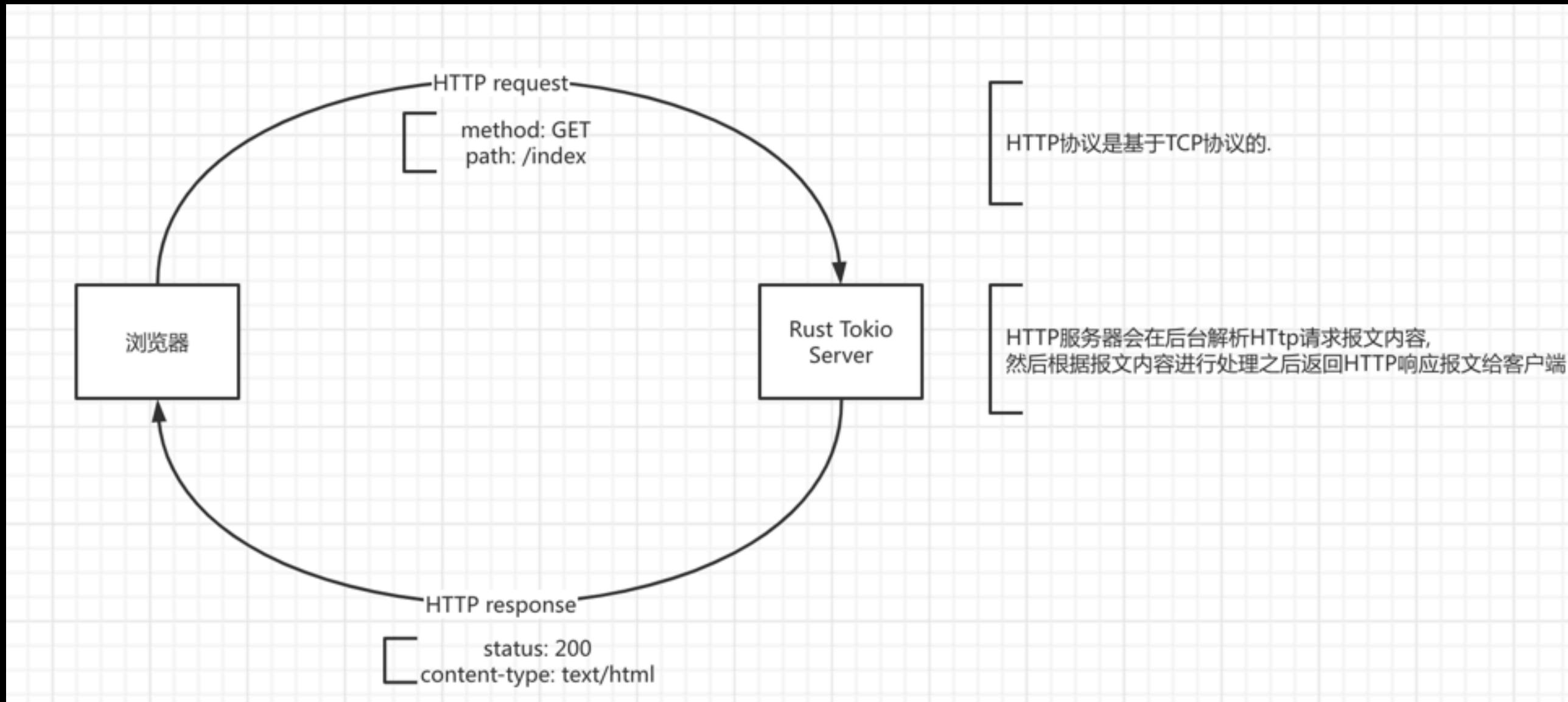
Mysql、Redis、RPC

学完后可以举一反三, 用Rust构建更多的应用

回顾Tokio是什么

Tokio 是一个 Rust 异步运行时库，底层基于 epoll/kqueue 这样的跨平台多路复用 IO 以及 event loop，目前正在支持 io_uring。它的 scheduler 和 Erlang/Go 实现的 N:M threads 类似，线程会执行 Task，可以充分利用多核。Task 是 Rust 基于 Future 抽象出的一种绿色线程，因为不需要预先分配多余的栈内存，可以创建大量 task，很适合做 IO 密集型应用。

利用Tokio实现一个Mini Http Server



利用Tokio实现一个高性能Mini Http Server | Web 框架建议选用 Axum

压测Mini Http Server

Rust web框架生态介绍

为何需要使用Web框架？

Rust生态当中的异步框架

Rocket 比较早期 -> Tokio

tide -> async_std 目前感觉维护不活跃

Axum -> Tokio官方出的, 充分利用了tokio生态,
Rust在web生态开发的里程碑

QA环节

加群一起交流Rust & Datafuse

