

JSF实现按APP实现调用次数限制预研

工具

被章耿添加, 被章耿最后更新于二月 06, 2015

需求:

- 一个接口下指定App在指定时间内限制其调用次数。
- 分析: 目前系统中没有业务系统这个概念, 只有IP。
- 调用次数限制: 有的要求精确限制调用次数 (即配多少就是多少), 有的是要求粗粒度限制调用次数 (配多少可以上下浮动)。
- 指定时间段: 有的是按分钟级限制, 有的是按天限制。

方案:

增加业务系统 (APP) 的概念

- 1. 用户在文件中配置标签, 每次请求带上APP, 缺点就是不可控
- 或者
- 1. JSF管理端开发接口下IP和App的关系映射界面。在JSF管理服务页面下可以勾选consumer进行设置app, 保留 接口+Consumer IP<--> App的关系。
- 2. Consumer在从注册中心订阅接口的Provider的时候, 注册中心附带返回App信息。此App信息会在Consumer每次调用的时候通过隐式传参带上, 发给服务端。

或者

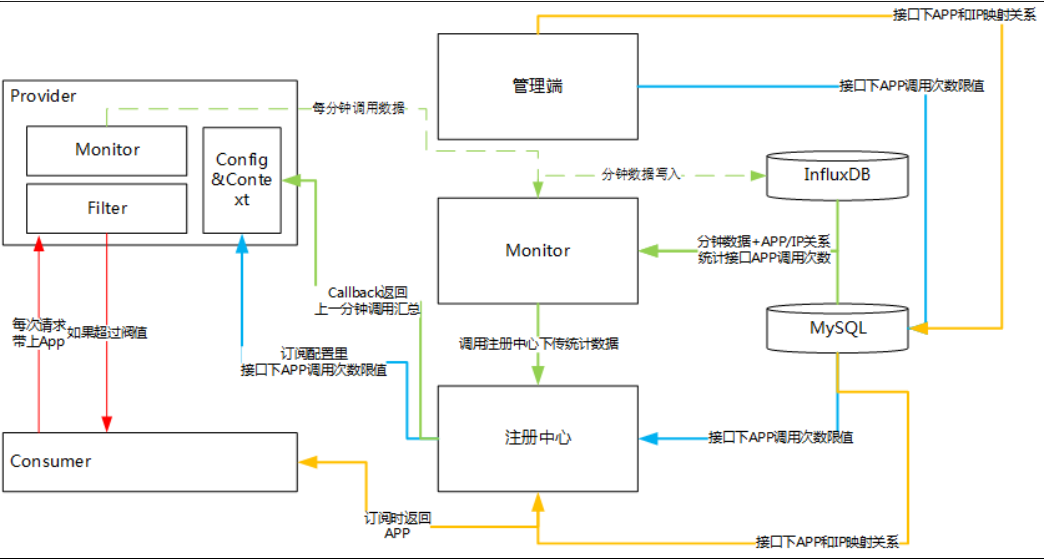
根据自动部署的启动脚本获取app_id, app_name, app_ins_id.

同时需要开发JSF管理端开发接口的方法下App的调用次数限额设置界面。保留 接口+App+方法<-->时间+次数 的数据。

调用次数限制方案一: 通过JSF Monitor Service收集数据

主要针对时间跨度较长且粗粒度限制的情况, 例如15分钟内调10万次可超出10%、或者一天调100w次限制这种。

- 1. (已有) 客户端每分钟的远程IP, 本地IP, 接口, 方法, 调用次数 发送给monitor server
- 2. Monitor server自动计算统计值 (根据数据库中远程IP和App的映射关系, 和Influxdb中的数据, 统计该接口该App的调用次数)
- 3. 注册中心和Provider增加统计数据下传Callback事件, 并且暴露出数据下传服务
- 4. Monitor server调用注册中心的数据下传服务, 将统计数据通过Callback下传到接口的全部Provider。
- 5. Provider从注册中心订阅接口配置的时候, 增加返回App的调用次数限额数据。
- 6. Provider开发filter或者在JSFTask中, 根据调用次数和限额返回结果拦截。



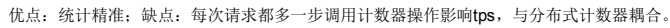
优点: 性能损耗少; 缺点: 不是很实时, 最近一分钟的数据可能超出限制。

调用次数限制方案二: 通过分布式计数器 (redis等) 收集数据

主要针对时间跨度较短且精确的情况, 例如1分钟内只调1w次, 不能多调。

- 1. 业务系统需要引入一个外部的分布式计数器
- 例如使用redis (结合incr和expire), key是接口+app, 来请求就incr, 这个key的过期时间是一分钟。
- 2. 业务系统自己维护限额数据, 或者 Provider从注册中心订阅接口配置的时候, 增加返回App的调用次数限额数据。
- 3. Provider开发filter, 在invoke方法中, 每次调用之前调用下计数器, 根据请求ip, app, 接口等信息, 判断是否超过限额, 进行拦截。





The diagram illustrates the architecture of the service call monitoring system, showing the flow of data and control between various components.

Core Components:

- InfluxDB:** A time-series database for storing monitoring data.
- MySQL:** A relational database for storing application and instance information.
- Monitor:** The central monitoring component, containing a **调用次数计算模块** (Call Count Calculation Module).
- 注册中心 (Registration Center):** Manages application and instance registration.
- 管理端 (Management End):** The user interface for the system.
- Provider:** Contains **Config&Context** and **Filter** modules.
- Consumer:** Contains **Filter** and **Config&Context** modules.
- 下发模块 (Distribution Module):** Manages the distribution of monitoring data.
- APP限制维护模块 (APP Restriction Maintenance Module):** Manages application restrictions.
- 自动部署系统 (Automatic Deployment System):** Automates the deployment of applications.

Data Flow and Interactions:

- 数据写入 (Data Writing):** The **Monitor** writes monitoring data to **InfluxDB** (分钟数据写入) and stores application/instance relationship statistics in **MySQL** (分钟数据+APP/IP关系统计接口APP调用次数).
- 接口限制 (Interface Restrictions):** The **管理端** interacts with the **APP限制维护模块** to set restrictions (保存接口+APP+间隔+次数限值) and with **MySQL** to find instance keys (根据app_id, app_ins_id找到ins_key下线).
- 注册与调用 (Registration and Invocation):** Applications and instances register with the **注册中心**. The **注册中心** interacts with **MySQL** to store instance keys and IP/application names. The **Monitor** sends monitoring data to the **注册中心** (调用注册中心下传统计数据).
- 回调与统计 (Callback and Statistics):** The **注册中心** sends callbacks to the **Monitor** (回调返回上一分钟调用汇总/或者是否可调用) and manages the distribution of data (下发模块).
- 配置与过滤 (Configuration and Filtering):** The **Provider** and **Consumer** modules interact with the **注册中心** to receive configuration and context information (app.id, app.name, app.ins.id). The **Filter** modules in both Provider and Consumer interact with each other to enforce restrictions (如果超过阈值, 每次请求带上App_id, Ins_id).
- 部署与调用 (Deployment and Invocation):** The **自动部署系统** deploys applications into the **JVM环境 (自动部署)**. The **Provider** and **Consumer** modules interact with the **JVM环境** to manage application and instance information.

- 无