# 一、Host Monitor需求

## 1. 项目介绍

## 2. UI原型图

### 2.1总览界面



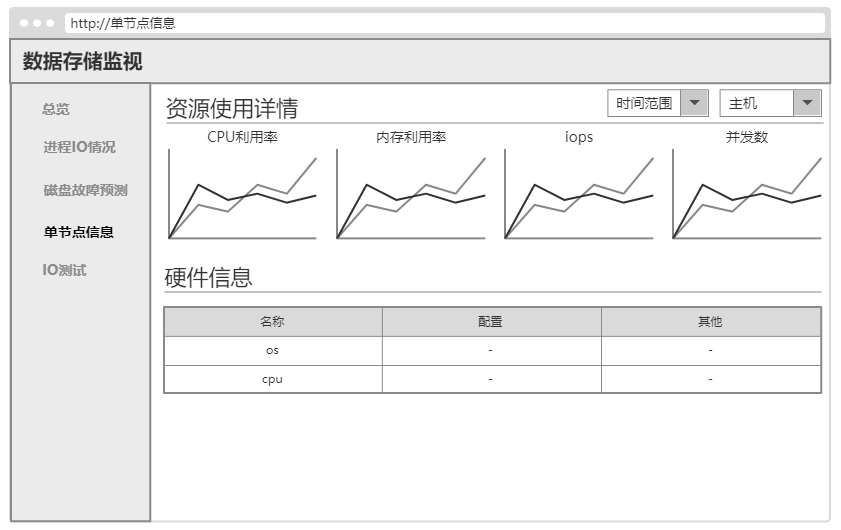
### 2.2进程IO界面



### 2.3磁盘故障预测界面



### 2.4节点信息界面



### 2.5 IO测试界面



## 3. 功能需求

## 4. 非功能需求

# 二、Host Monitor设计

## 1. 架构设计

项目整体使用Spring Cloud微服务架构，项目以用户交互为划分依据分为前后端两部分，使用B/S web结构。用户直接与前端进行交互。根据需求，后端由多个子模块/微服务组成。模块之间会相互通信合作完成业务流程。

项目分为5个模块：

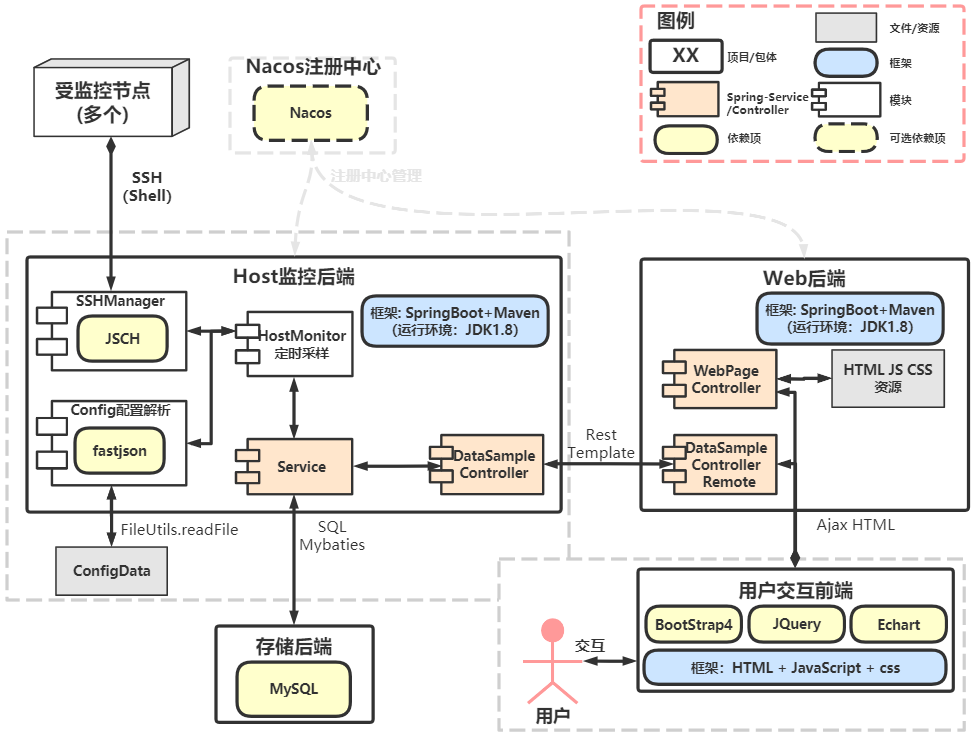
（1）Host监控后端

（2）Web后端

（3）存储后端

（4）用户交互前端

（5）Nacos注册中心(可选)



## 2. Host监控后端

**项目框架：**Spring Boot，Maven。

**运行环境：**JDK1.8

### 2.1主要功能

HostMonitor模块，通过使用JSCH向所有被监控Host进行SSH连接，通过定时远程执行Shell脚本的方式进行数据采样，获取采样数据。

Service，用于管理HostMonitor，并且会定时将数据存储到数据库中。遇到部分请求会通过查询操作从数据库读取并返回数据。此外也包含部分数据缓存，减少频繁的数据库读取操作。

DataSampleController为Spring的控制器，用于对外暴漏服务接，即可以通过URL获取数据。为Web后端模块提供服务支持。

### 2.2其他模块

SSHManager，使用JSCH与配置文件给出的Host建立连接，便于后续执行Shell。

Config配置解析，读取并解析ConfigData文件夹下的文件。使用fastjson解析json配置文件。

### 2.3配置数据

需在Jar文件同级目录下包含ConfigData文件夹及其下所有文件。

（1）SampleCommand.sh

采样指令，每次都会执行文件内所有指令。

（2）StorageDeviceInfo.json

主配置文件。用于配置被监控的Host信息，用于SSH连接的用户名密码。以及每次采样所记录的JSON数据格式。

（3）InitEnvironment.sh

Host环境初始化指令文件。Host采样指令需要使用多个外部依赖库，因此需要在第一次采样前安装环境。

（4）IOTest.sh

IO测试指令文件。当对Host进行IO测速时会执行一次。

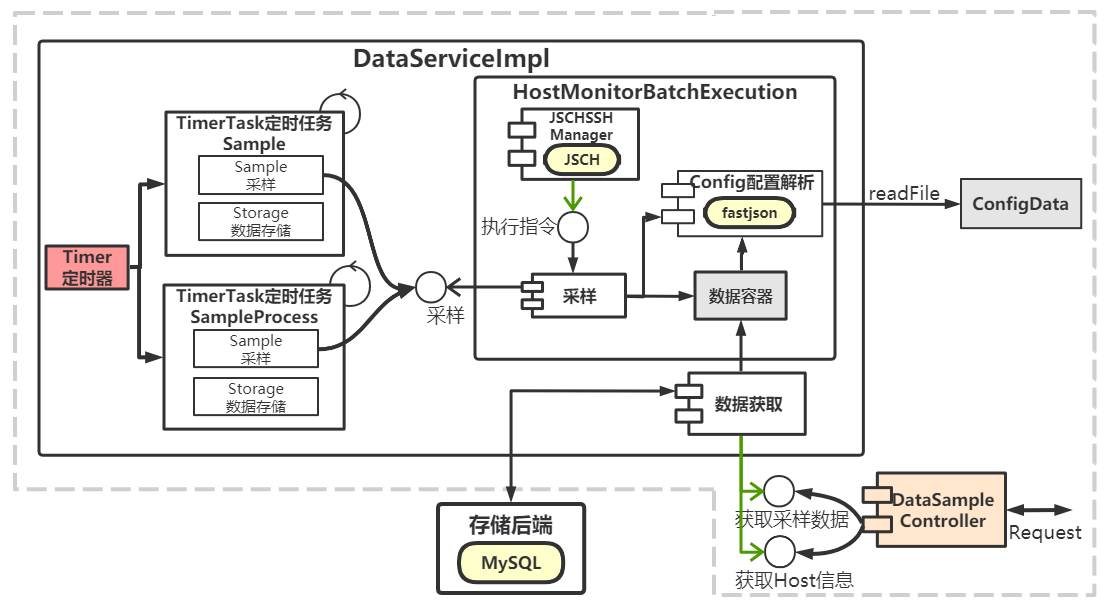
（5）ProcessSampleCommand.sh

进程采样指令。对Host实时运行的进程进行采样，获取进程io等信息。

（6）test.sh

用于项目功能测试的脚本。（生产环境中会使用到，实际工作环境不会使用到）

### 2.4结构



本模块基于Spring Boot框架。主体为MVC三层结构：Service，DAO（Data Access Object），Controller。

Controller层为控制层，用于接收外部请求并返回数据。DAO层为数据持久化层，主要使用Mybatis框架通过SQL查询数据库获取数据并直接转换成Java对象。Service层为中间的服务层，主要为控制层提供各种服务，并且根据控制层需求调用DAO层获取所需数据。

本模块的主控制类为DataSampleController。主服务类接口为DataService，其中DataServiceImpl为当前版本的服务实现类。

（1）DataServiceImpl

内部使用TimerTask执行两个定时任务，负责向所有Host周期性采样，然后存储新的数据。这些定时任务会在Service构造后直接开始执行。

其中，内部使用HostMonitorBatchExecution所提供的接口进行采样。为Controller提供服务接口，根据需求返回所需数据。数据可能来自HostMonitorBatchExecution类内的数据容器，也可能来自数据库。

（2）HostMonitorBatchExecution

本类使用单例模式。

包含一个配置解析类Config，用于读取各种配置文件，以及获取采样所需的Shell脚本。

包含采样数据容器，存储所有Host的采样数据，且每次采样后都会更新。

异步采样：由于Host可能会很多，因此与顺序采样相比，使用异步采样能够有效减少等待时间。并且使用了线程池防止线程频繁创建与调用。每次采样都是使用JSCHSSHManager对所有Host执行一个采样脚本指令，然后返回结果，之后对结果进行解析并存入数据容器中。

对外提供多个接口，用于获取容器中的采样数据，以及获取配置文件解析后的数据。

（3）Config配置解析

用于读取各种配置文件，以及获取采样所需的Shell脚本。

配置文件位于ConfigData文件夹下。开发环境中，此文件位于父工程目录下。实际运行时需位于本模块Jar包同级目录下。

Config类目前仅被HostMonitor类所管理与直接使用。

（4）JSCHSSHManager

使用JSCH库，根据提供的Host IP、用户、密码等（代理IP与端口为可选项）信息，与对应Host进行SSH连接并执行指令（Shell脚本），然后返回指令执行结果。

类内部缓存多个Session，防止每次执行指令后Session重复的创建与删除。因此如无网络错误，基本上是第一次对所有Host采样时建立连接，之后每次采样都会重用连接执行脚本。当然如果Session出现网络连接错误，则会造成当此采样失败并抛出异常，这样会移除异常的Session，从而当下次采样时会重新建立连接。

JSCHSSHManager实现了SSHManager接口runCommand方法。

（5）DataSampleController

提供对外接口，并使用DataServiceImpl提供的接口处理请求并返回所需数据。

## 3. Web后端

**项目框架：**Spring Boot，Maven。

**运行环境：**JDK1.8

**主要功能：**

用户通过URL访问并显示Web界面。

远程调用Host监控后端中的服务，获取采样信息数据。

**依赖资源/库：**

前端所依赖的库均下载到项目内，

## 4. 存储后端

部署MySQL数据库，用于采样数据存储。

## 5. 用户交互前端

是与用户交互的前端部分。用户通过浏览器输入URL访问到项目交互界面。

**依赖库：**

（1）BootStrap4：界面UI控件库。

（2）EChart：界面图表库

（3）JQuery：使用ajax与后端进行数据收发。

## 6. Nacos注册中心(可选)

Nacos主要用途是作为SpringCloud的微服务注册中心，监控多个微服务节点。同时方便后续扩展与管理，方便微服务分布式运行与配置。

Nacos注册中心为可选模块项，本项目可不适用注册中心也可实现正常运行。