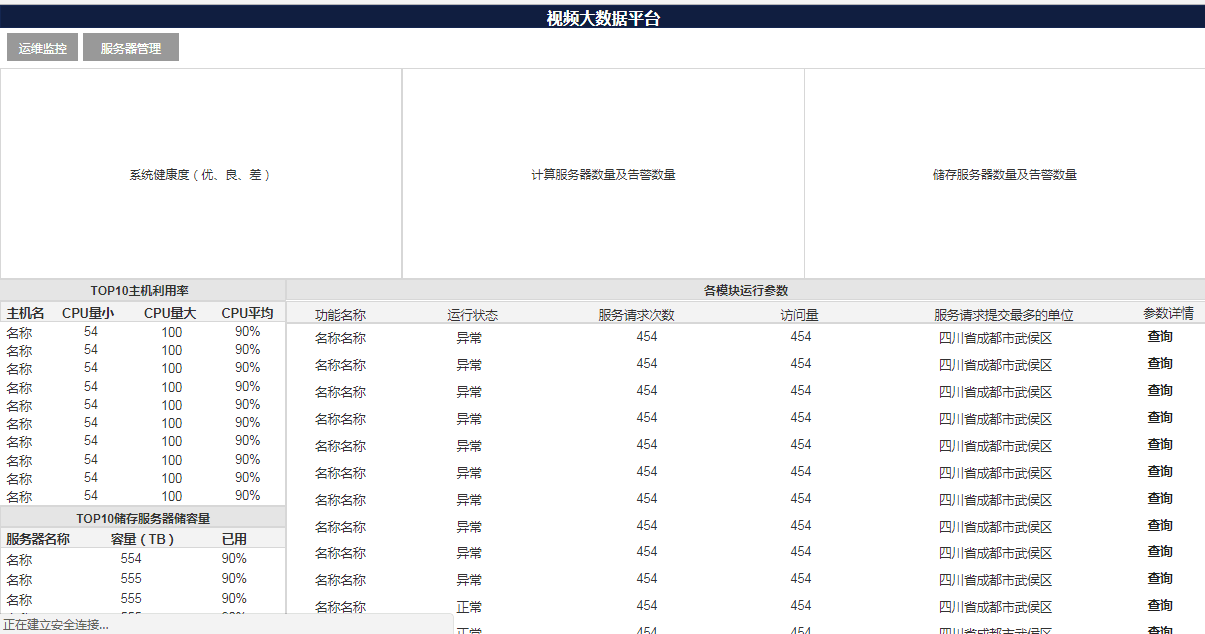
关于运维模块设计的思考

## 需求

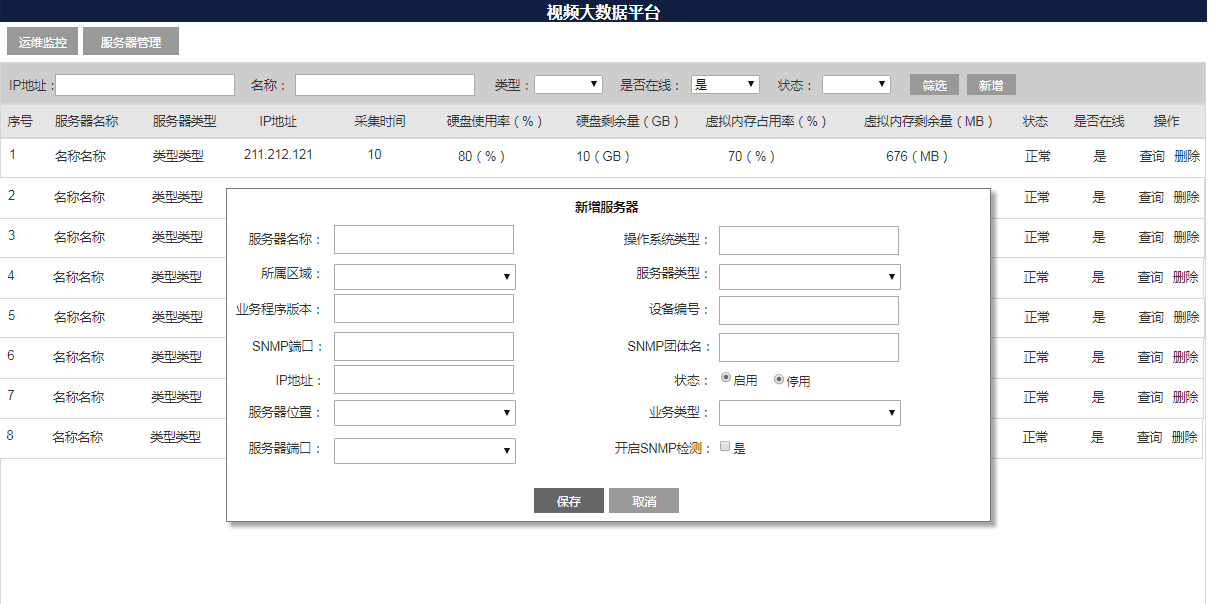
1. 对集群机器进行资源监控（包括但不仅限于CPU，内存，磁盘空间，磁盘IO，网络IO等）
2. 对应用平台spring cloud微服务进行监控（服务访问次数）
3. 对大数据平台的服务资源进行监控（Hadoop，spark，storm，kafka，但就目前的原型上并没看到此类需求）

## 相应原型

## 运维监控



## 服务器管理



## 目前对产品需求存在的疑问

1. 不确定该模块的页面设计基于原型自己做还是使用使用监控工具自带的监控WEB页面
2. 就原型而言似乎没看到关于大数据平台的相关监控需求（如对kafka,storm，hbase等），是未完善还是借用CDH5 管理工具。

## 可行性方案思考

### 方案一：基于ganglia

优点：

1. 其本身就具有分布式系统资源监控功能，并且提供了可视化web界面，对于需求1可以满足
2. Ganglia支持扩展，在理论上我们可以让其支持对微服务的监控已晚上对需求二的支持
3. 目前Hadoop对ganglia有原生的支持，应该可以适应对大数据平台监控的需求。
4. 目前ganglia似乎提供了restful api 这为我们自己写web前面提供了可能

缺点：

1. 原生界面为英文版，需要一定工作量来汉化，当然如果不用自带的web界面，自行开发也是可以考虑的，毕竟提供了restful api。
2. Spring cloud似乎没有支持ganglia，需要开发扩展，而且扩展程序基于python，c，不是所有的研发都对熟悉这两个语言
3. 缺乏报警。顺然有监控，但据说不具有报警功能，如某资源超负荷后，不会报警，只能人为判断。

**总结：系统资源的监控应该是OK的，对于微服务的监控可能需要评估一下。对于页面是否需要汉化看需求到底是自己写页面还是用原生监控页面。邮件报警和短信报警似乎不支持。**

### 方案二：基于ganglia + spring cloud 监控全家桶

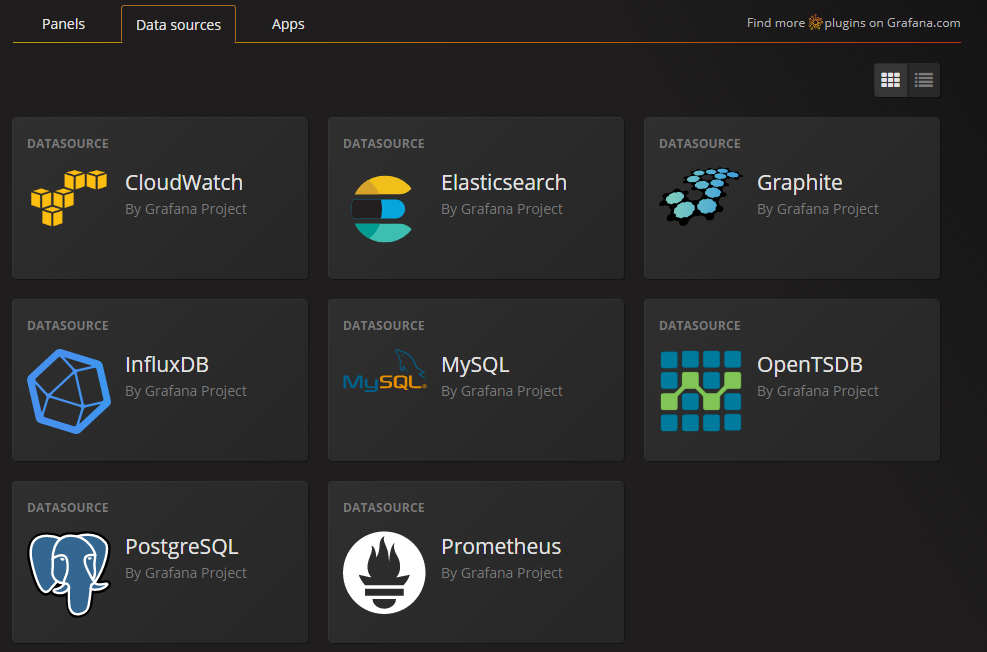
Spring cloud监控全家桶可以做到服务是否正常，调用次数统计，调用链跟踪。问题在于如何处理spring cloud和ganglia两种监控数据源。

1. 通过扩展的方式让ganglia搜集spring cloud数据，这样前端只需要从ganglia请求数据即可
2. 按部就班，系统资源等数据从ganglia收集，微服务部分从spring cloud暴露的api收集

总结：感觉上是解决了微服务的问题，但决定

### 方案三：基于granfana

Granfana本身不具有监控功能，他更像面向运维监控而抽象出来的一层。利用它就像堆积木，想要监控什么只要组合起来就可以，Granfana负责展示就好。Granfana实现这一点的方式在于支持多种数据源



优点：

1. 几乎对目前所能想到的监控对象都可以利用搜索引擎找到相应的解决方案，也就说监控方案较齐全，微服务，hadoop集群，各种数据库监控。网上资源较多，学习资料也多。
2. 支持报警功能
3. 前端页面可进行二次开发

缺点：

1. 不确定是否提供api，如果我们想自己开发界面，可能有所难度。但是若对其有更多的了解，可能这并不是一个复杂的问题。

## 总结

就目前经验和了解来看，似乎方案一二的可行性相对更高，但个人直觉上更看好方案三。目前没有一个监控工具可以满足当前需求，组合拳是肯定要打的。这些方案中信息多来于互联网，很多仍需要技术验证