360病毒样本大规模异构实时扫描平台

2017年

个人简介

- **2008年 至 今　360公司**

  现在所在岗位：

  核心安全事业部/服务端部门/样本鉴定流程组

  程序猿一枚

- **参与项目**

  云查杀

  样本存储

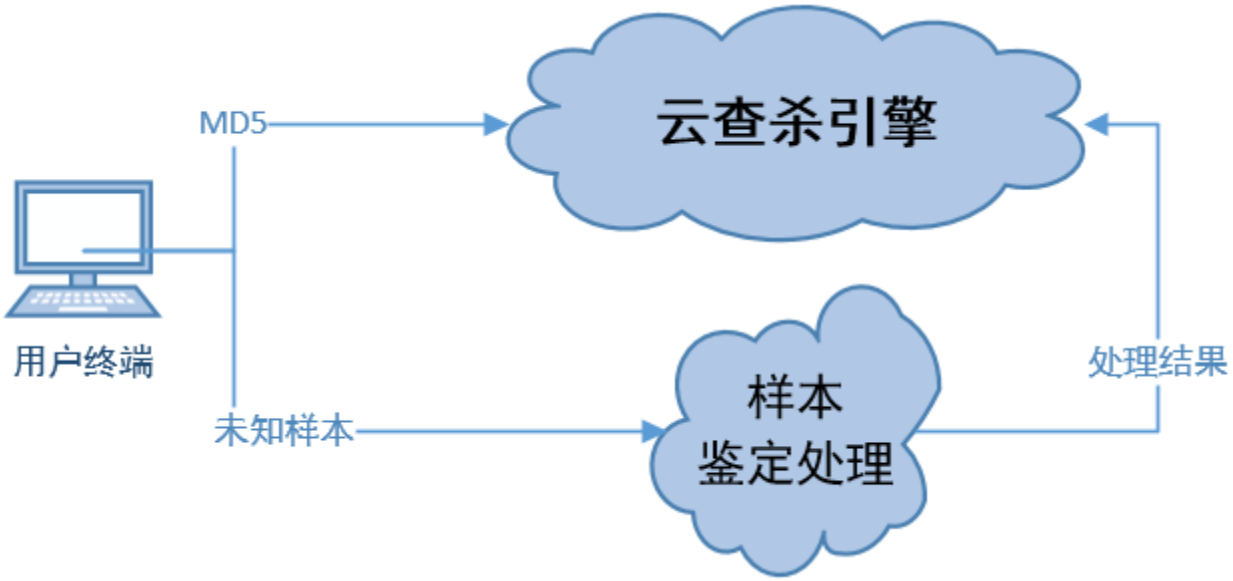
  样本自动化处理平台

  360搜索

大纲

- 背景介绍
- 发展历程
- 细节剖析
- 总结
- 未来展望

背景介绍



业务背景

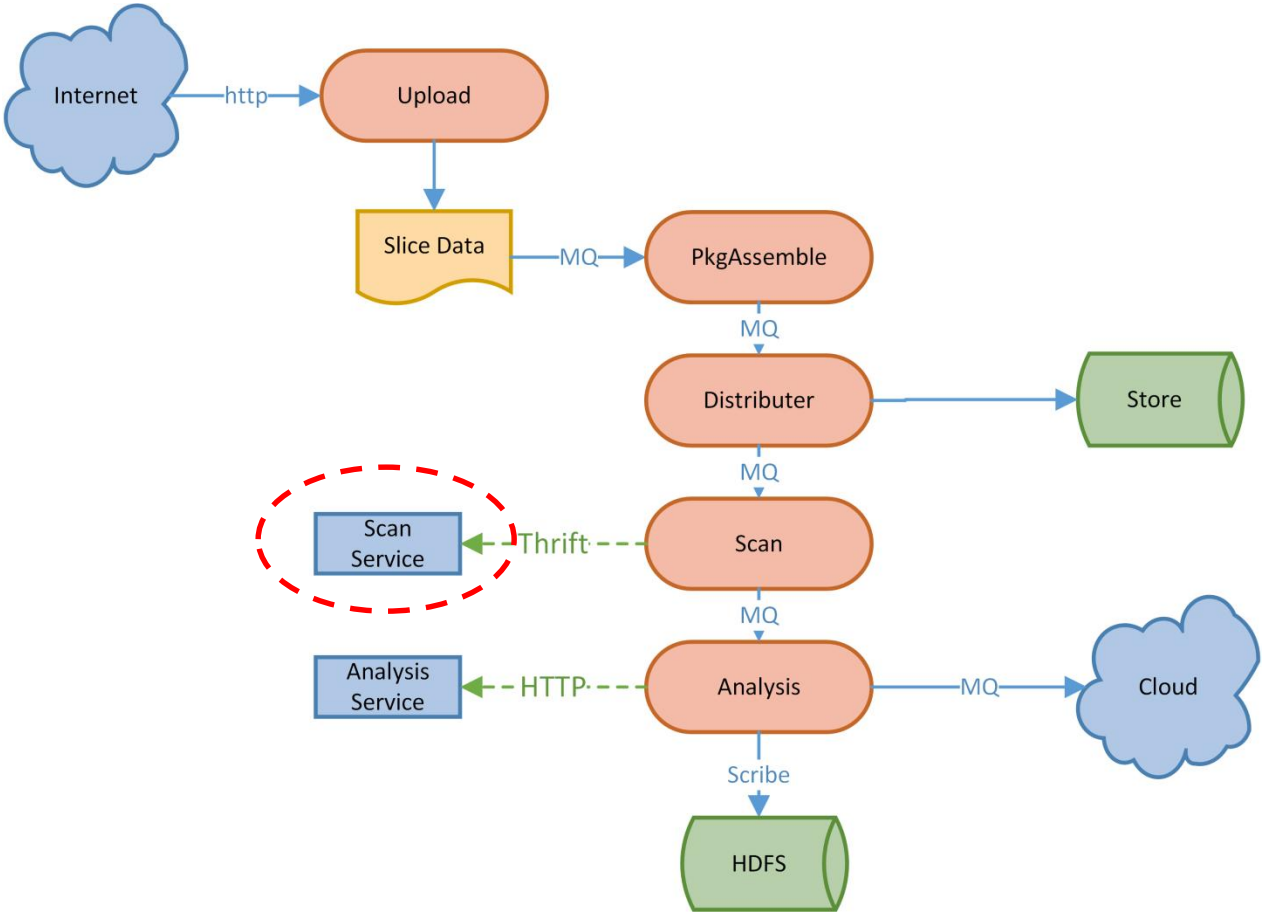
- **系统能够做到实时的流式处理**

  第一时间对木马病毒样本进行鉴定，并给予有效处理

- **进行高吞吐量的回归扫描**

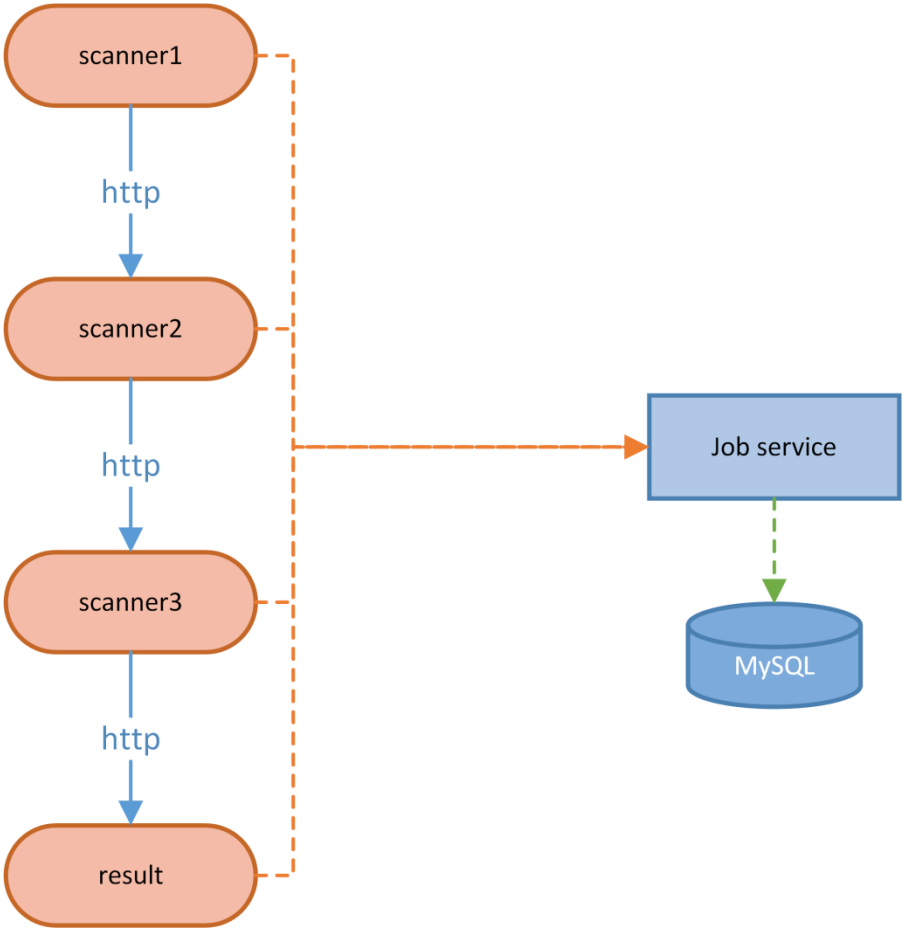
  要在算法更新特征库升级的情况下对历史样本进行高吞吐量的回归扫描

背景介绍


Internet
http
Upload
Slice Data
MQ
PkgAssemble
MQ
Distributer
Store
MQ
Scan Service
Thrift
Scan
MQ
Analysis Service
HTTP
Analysis
MQ
Cloud
Scribe
HDFS

发展历程-v1版本

初期版本



发展历程-v1版本

业务指标：

60秒出鉴定结果

解决方案：

基于内存方式减少磁盘IO

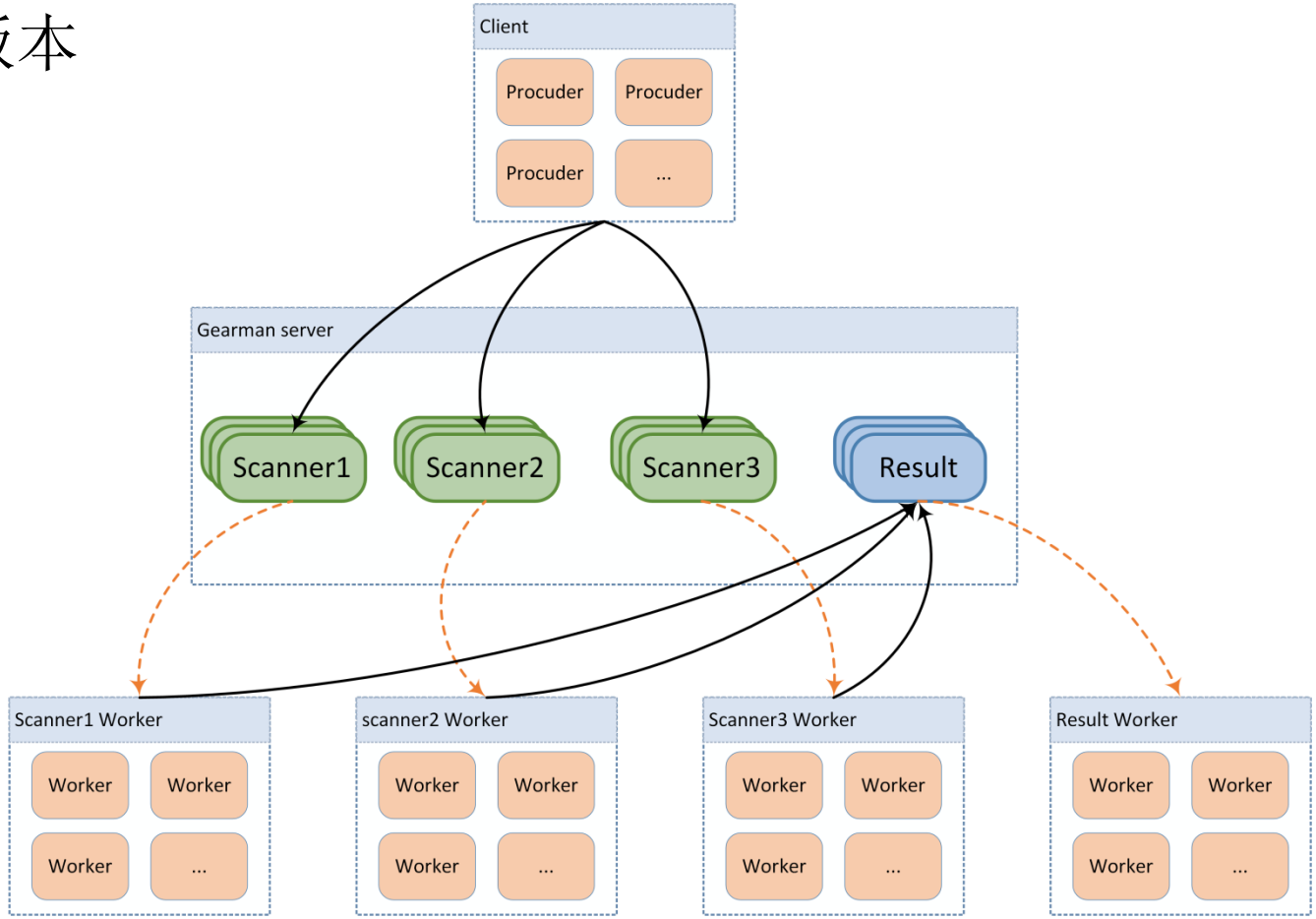
扫描模块串行改为并行

引入相关技术：

分布式队列Gearman

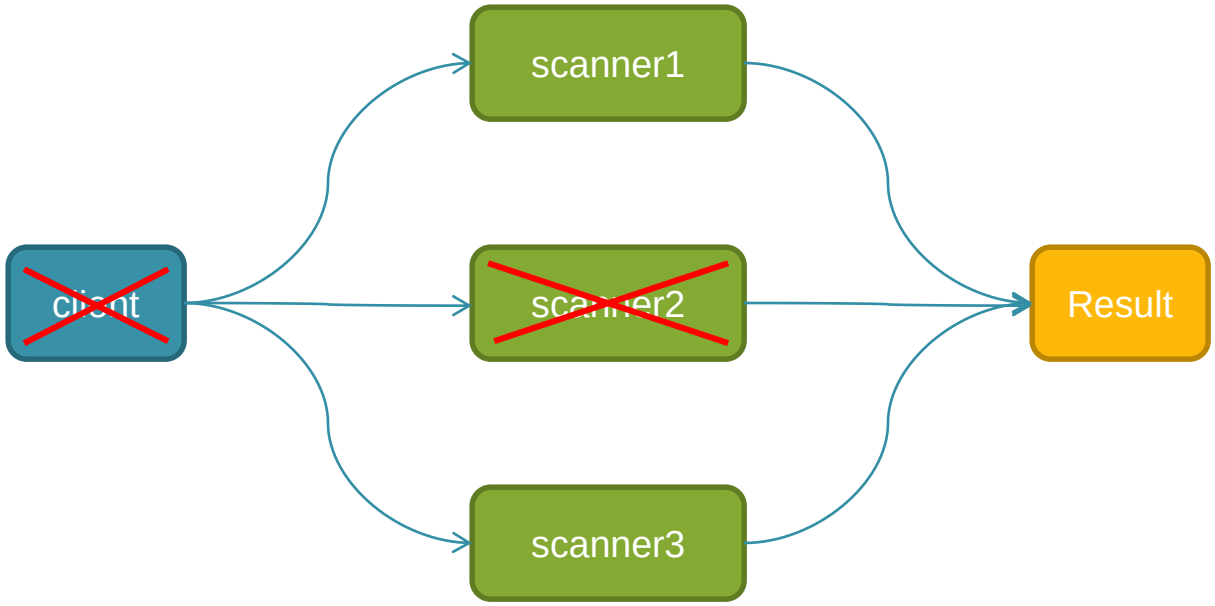
发展历程-v2版本

初期版本



发展历程-v2版本

队列阻塞具有传递性，任何一款扫描模块故障都会导致系统崩溃



发展历程

业务指标：

提高**可靠的保证**

问题：

单款扫描器故障导致系统无法工作

扫描器越多整体故障率越高

运维成本高

细节剖析

理论上的通用需求：

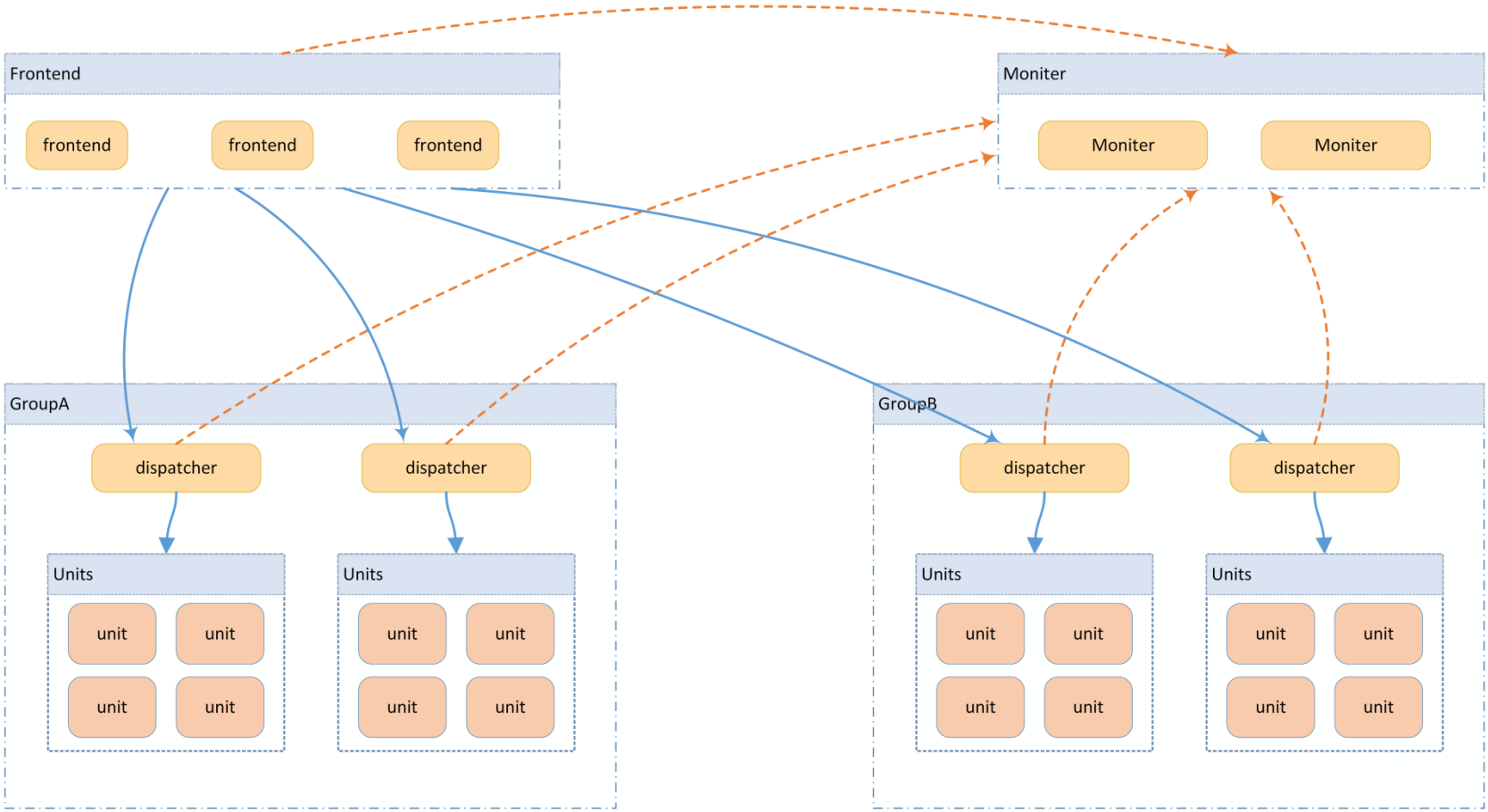
- 低延迟
- 高性能
- 可扩展
- 容错

细节剖析

需求分析：

- 跨平台支持
- 需要多语言
- 需要现成的网络通信模块
- 计算节点可扩展并且支持负载均衡调节
- 故障情况下可以降级服务
- 单样本扫描的业务可靠性要求不高
- 低优先级的任务不要求时效性
- 尽量用满所有计算资源
- 运维操作尽量简洁

细节剖析


Frontend
frontend
frontend
frontend
Moniter
Moniter
Moniter
GroupA
dispatcher
dispatcher
Units
unit
unit
unit
unit
Units
unit
unit
unit
unit
GroupB
dispatcher
dispatcher
Units
unit
unit
unit
unit
Units
unit
unit
unit
unit

细节剖析

Monitor

作用：

配置中心化
心跳检测
dispatcher 负载调度

实现：

mysql做配置持久化存储
心跳检查，以及负载数据，用内存临时存储
thrift网络RPC

细节剖析

Frontend

作用：

用户接口
扫描开始将task推送至合适的dispatcher
扫描结束将task结果缓存，或者推送到异步队列中

实现：

提交任务时候请求Monitor获取dispatcher负载并作出选择
ScanFlag参数实现降级服务

细节剖析

接口定义示例：

```
/**
 * Commit a scan task to the platform, this call will block
 * until the server returns
 */
map<string,Result>
    ScanWait(
            1:Sample sample,
            2:list<string> methods,
            3:i32 timeout,                  /** seconds **/
            4:map<string, ScanFlag> flags)
    throws (1:InvalidOperation io,
            2:ScanWaitTimedOut swto);

/**
 * Commit a scan task with ext, return immediately, client should use
 * the `Query` method to get the result.
 */
i64
    ScanWithExt(
            1:Sample sample,
            2:list<string> methods,
            3:Priority priority,
            4:map<string, ScanFlag> flags
            5:string ext)
    throws (1:InvalidOperation io);

/**
 * Get result of a task.
 */
ScanRst
    QueryWithExt(1:i64 task_id)
    throws (1:InvalidOperation io);
```

细节剖析

接口定义示例：

```
/** Sample file description */
struct Sample {
    1: required Protocol protocol = Protocol.NONE;
    2: optional string uri;              // Sample location description
    3: optional string md5;              // Sample md5
    4: optional i64    size;               // Sample file size
    5: optional string name;             // Sample file name, must be provided when protocol == NONE
    6: optional string content;
}

/** Scan flag */
struct ScanFlag {
    1: optional i32 omit_if_no_resource = 0;    // if no resource, 0: can't omit , 1: omit, 2: at least one can't omit
    2: optional bool retry_if_failed = 1;        // need retry if scan job failed
    3: optional i32 max_retry_times = 3;         // max retry times when scan job failed
}


/** Single task result for scan */
struct Result {
    1: optional ErrorCode ec = ErrorCode.OK;
    2: optional string msg;
    3: optional string name;
    4: optional string type;
    5: optional string desc;
}

struct ScanRst {
    1: map<string,Result> res;
    2: optional string ext;
}
```

细节剖析

Dispacher

作用：

推送task到空闲unit上
实现batch和single 两种模式
重试机制
优先级调度

实现：

优先级队列
unit调度时候对空闲资源采取轮询策略

细节剖析

Unit

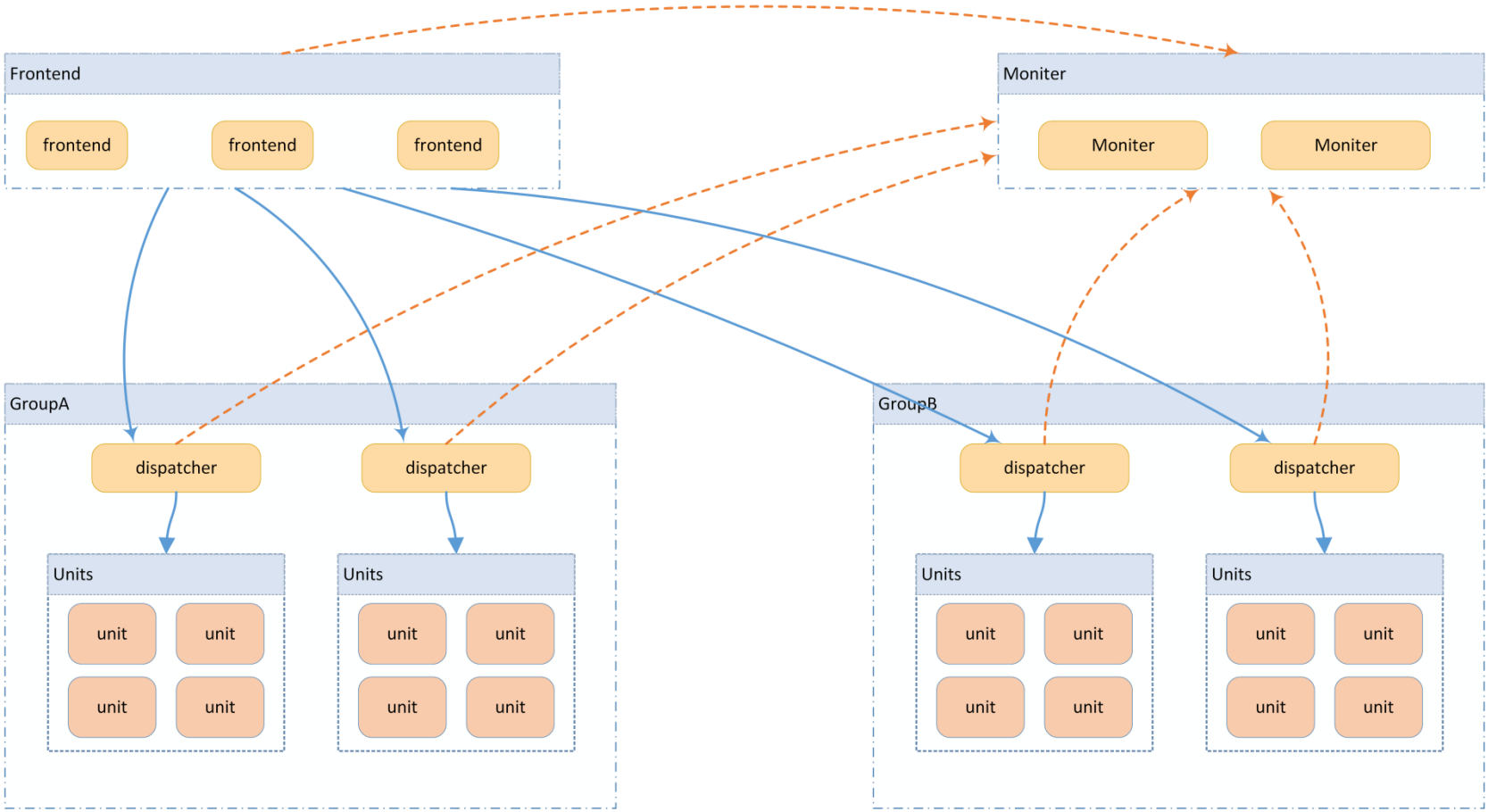
作用：

样本的扫描
支持windows / linux / andriod 多平台
扫描模块支持SDK，命令行，管道输入 等模式

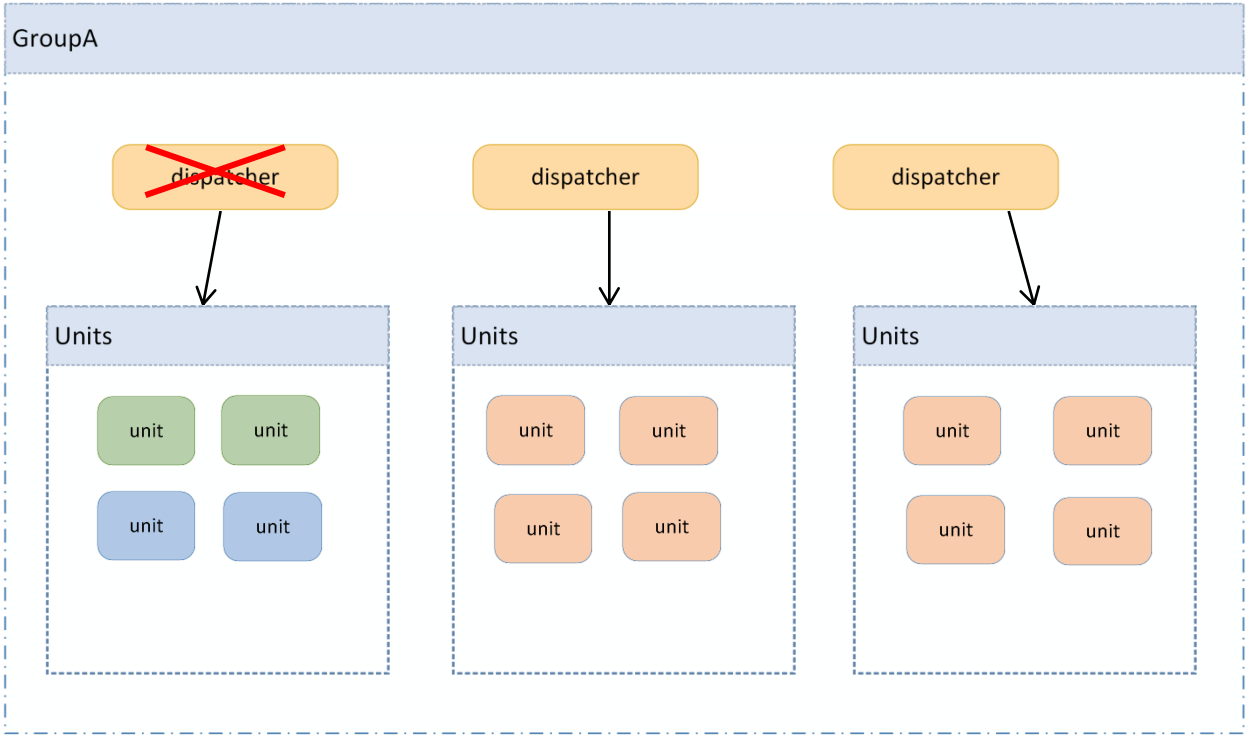
实现：

虚拟机使用xen,kvm
ramdisk作为样本落地目录加速扫描
windows用到了cygwin
整体以python实现，配合了c的py扩展

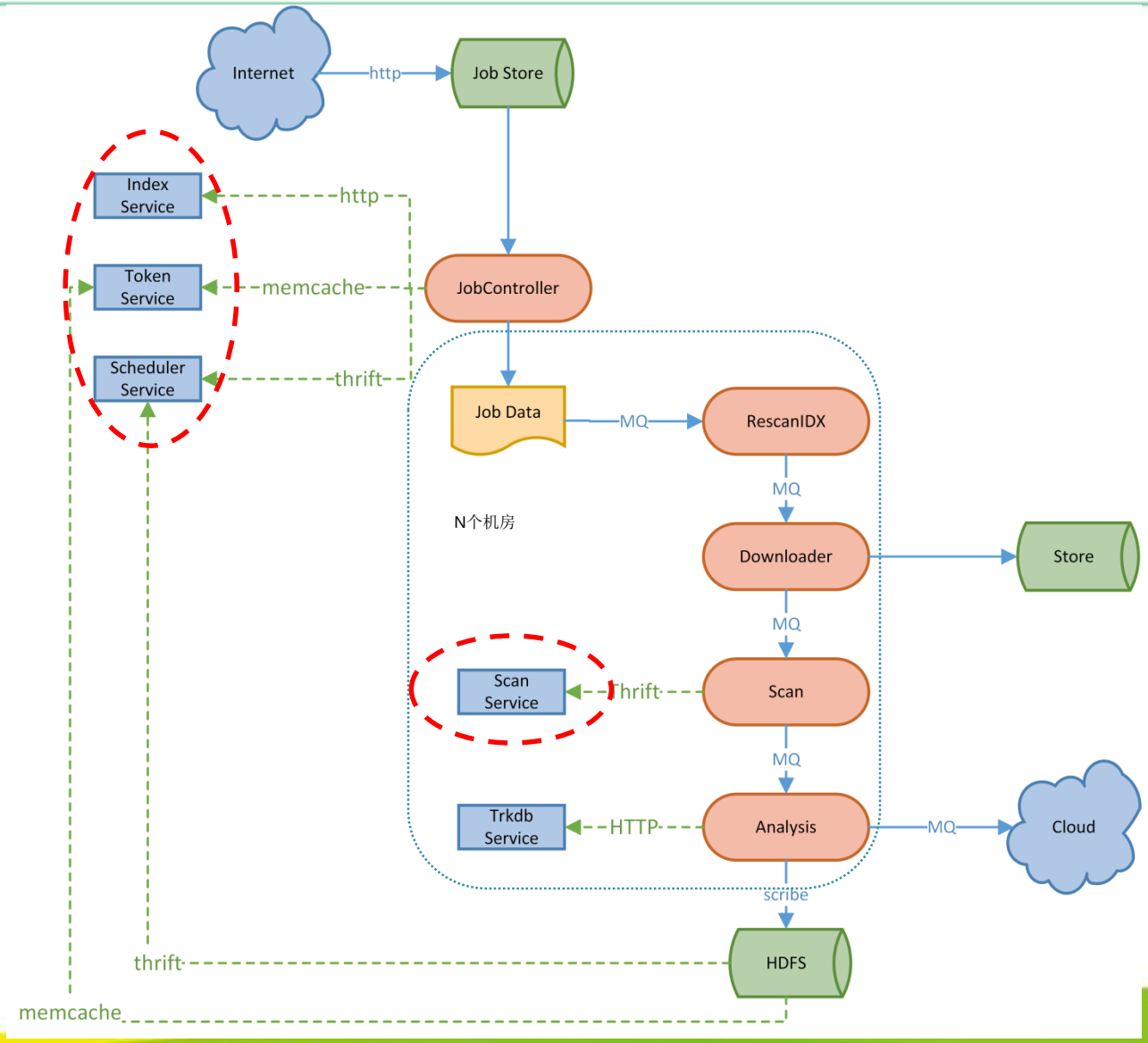
细节剖析


Frontend
frontend
frontend
frontend
Moniter
Moniter
Moniter
GroupA
dispatcher
dispatcher
Units
unit
unit
unit
unit
Units
unit
unit
unit
unit
GroupB
dispatcher
dispatcher
Units
unit
unit
unit
unit
Units
unit
unit
unit
unit

通过monitor方便实现故障自动调节功能

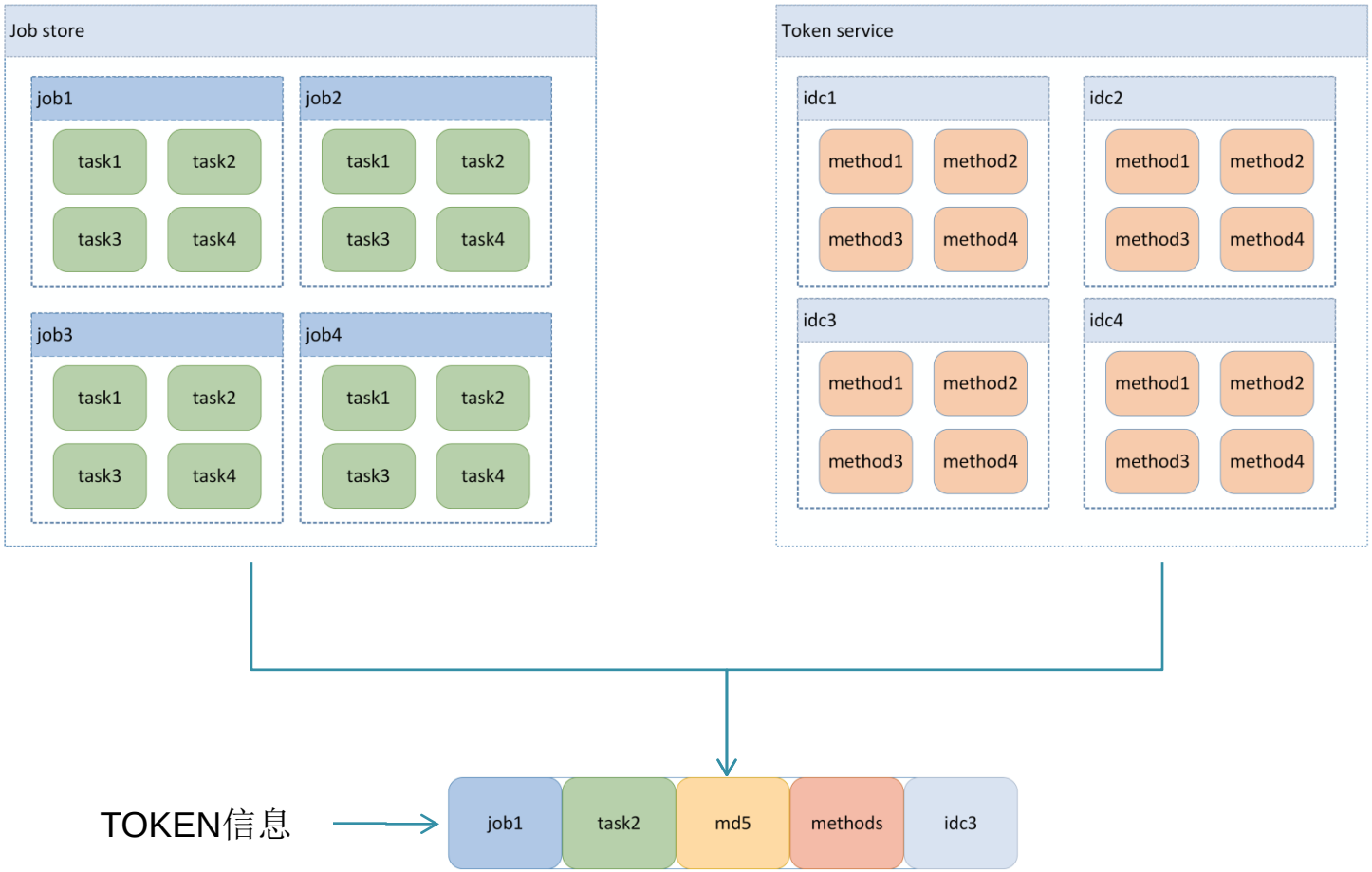


细节剖析-迭代改进


Internet
http
Job Store
Index Service
Token Service
Scheduler Service
http
memcache
thrift
JobController
Job Data
MQ
RescanIDX
MQ
N个机房
Downloader
Store
MQ
Scan Service
Thrift
Scan
MQ
Trkdb Service
HTTP
Analysis
MQ
Cloud
scribe
HDFS
thrift
memcache

细节剖析-迭代改进

Job管理以及Token服务



细节剖析-迭代改进

Job管理以及Token服务作用：

1. 限制流量
2. 多个集群负载均衡
3. 数据异常丢失Redo
4. 接受用户需求更友好

细节剖析-迭代改进

调度服务

作用：

减少重复扫描
合理的利用资源

实现：

kv作为数据存储
thrift网络RPC封装策略

总结

每日处理增量千万级样本量

每日处理增量样本大小在TB级

每日处理回扫样本数上亿

每日处理样本大小PB级

增量样本处理平均响应时间为10秒

总结

高可用性和扩展性

调试接入新的扫描模块比较便捷（类似调试MR的streaming）

必要时可以自动降级服务

总结

业务系统要有所舍弃

用最简单最可靠的技术

硬件的提升有时候可以更快速更廉价的解决问题

未来展望

未来改进点：

1. 评估服务容量极限
2. 更加人性化的全链路追踪系统
3. 虚拟机自动按需扩容和回收资源

谢 谢！