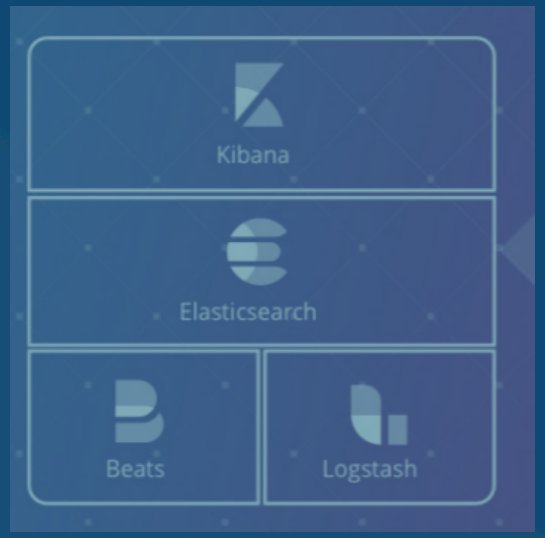


基于Elastic Stack+kafka构建日志监控平台实践

彭　科

CONTENTS

01 Part One
架 构 演 进

02 Part Two
监 控 运 维

03 Part Three
集 群 优 化

04 Part Four
业 务 案 例

CONTENTS

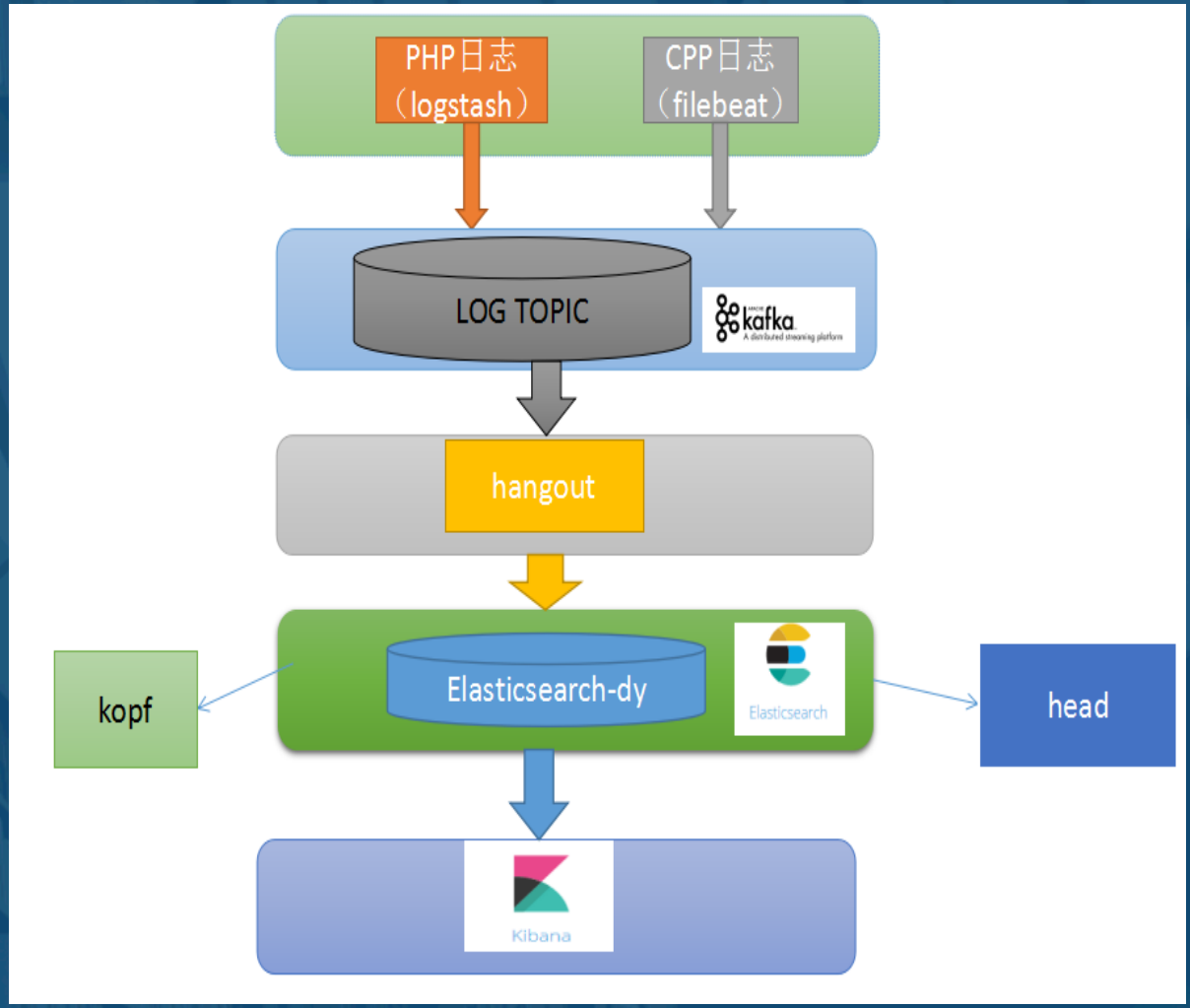
01 *Part One*
统一日志平台架构演进





统一日志平台V1

- **基础架构**
  - ●数据采集agent
  - ●日志中心
  - ●数据解析
  - ●日志存储
  - ●kibana展示

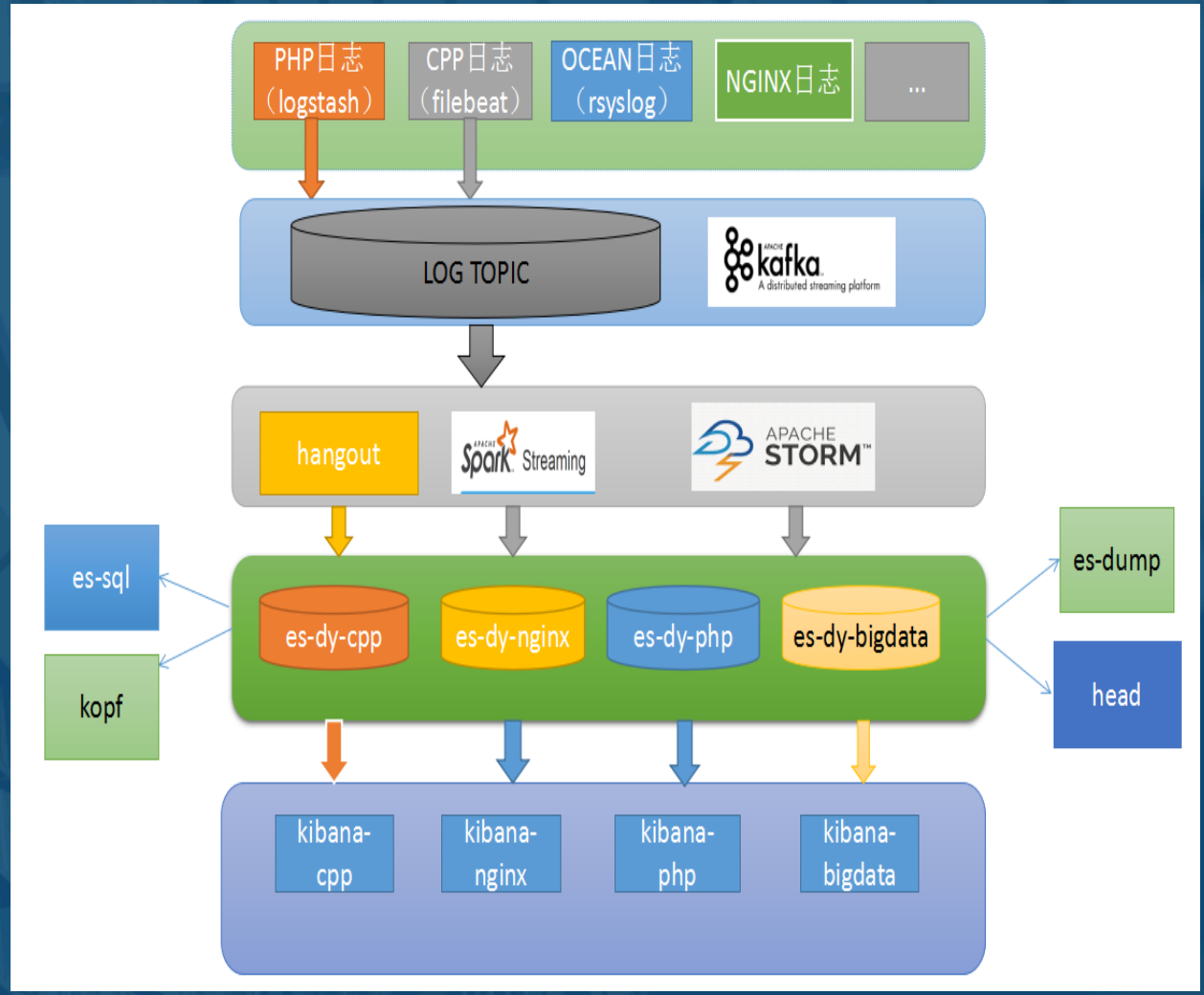


存在不足：
1.一个服务出现问题影响整个cluster
2.网络饱和
3.监控不足、节点频繁掉线
4.索引和搜索慢

统一日志平台V2

- **基础架构**
  - 引入数据归档插件
  - 引入sql查询插件
  - 集群按业务隔离
  - 引入实时流计算框架统计聚合后写入ES

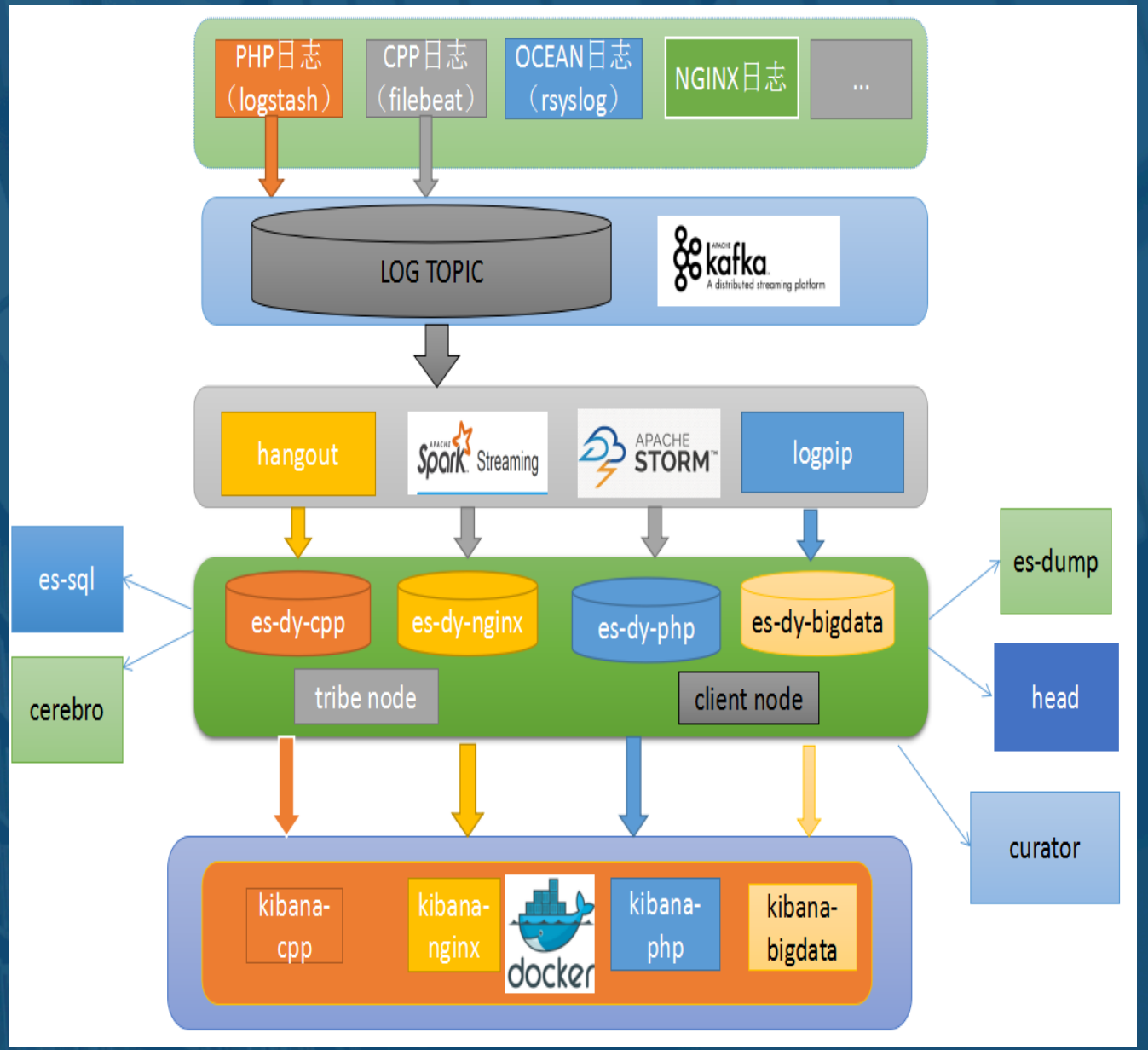
存在不足：
1.索引多，管理困难
2.hangout正则解析慢导致索引速度慢
3.管理员查询不同日志麻烦
4.索引分片不合理，段文件多，搜索慢



统一日志平台V3

- **基础架构**
  - 引入索引管理插件
  - 引入tribe节点和client节点
  - kibana容器化
  - 自写程序解析日志

待改进：
1.更细粒度权限管理
2.ES容器化
3.冷热分离



集群现状

基础运维

- 四大集群，30台物理机器， 60+节点
- 单日索引数据条数60亿+，单日新增索引文件大小10TB+
- 业务高峰期单个索引速度在10w/s以上，集群20w/s
- 历史数据保留时长根据业务需求制定，从3天至365天不等，归档重要日志
- Kibana用户从开发人员到数据分析人员和安全人员
- 对接日志种类包含Nginx日志、PHP业务日志和Tomcat容器日志、C++统计日志及业务日志和RPC日志、接口切面日志和GC日志、Mysql慢查询日志、应用日志等
- 集群管理监控工具丰富、升级快、扩展性强，下线与增加节点迅速

Elasticsearch: You Know, For Logs

CONTENTS

02 *Part Two*

监　控　运　维

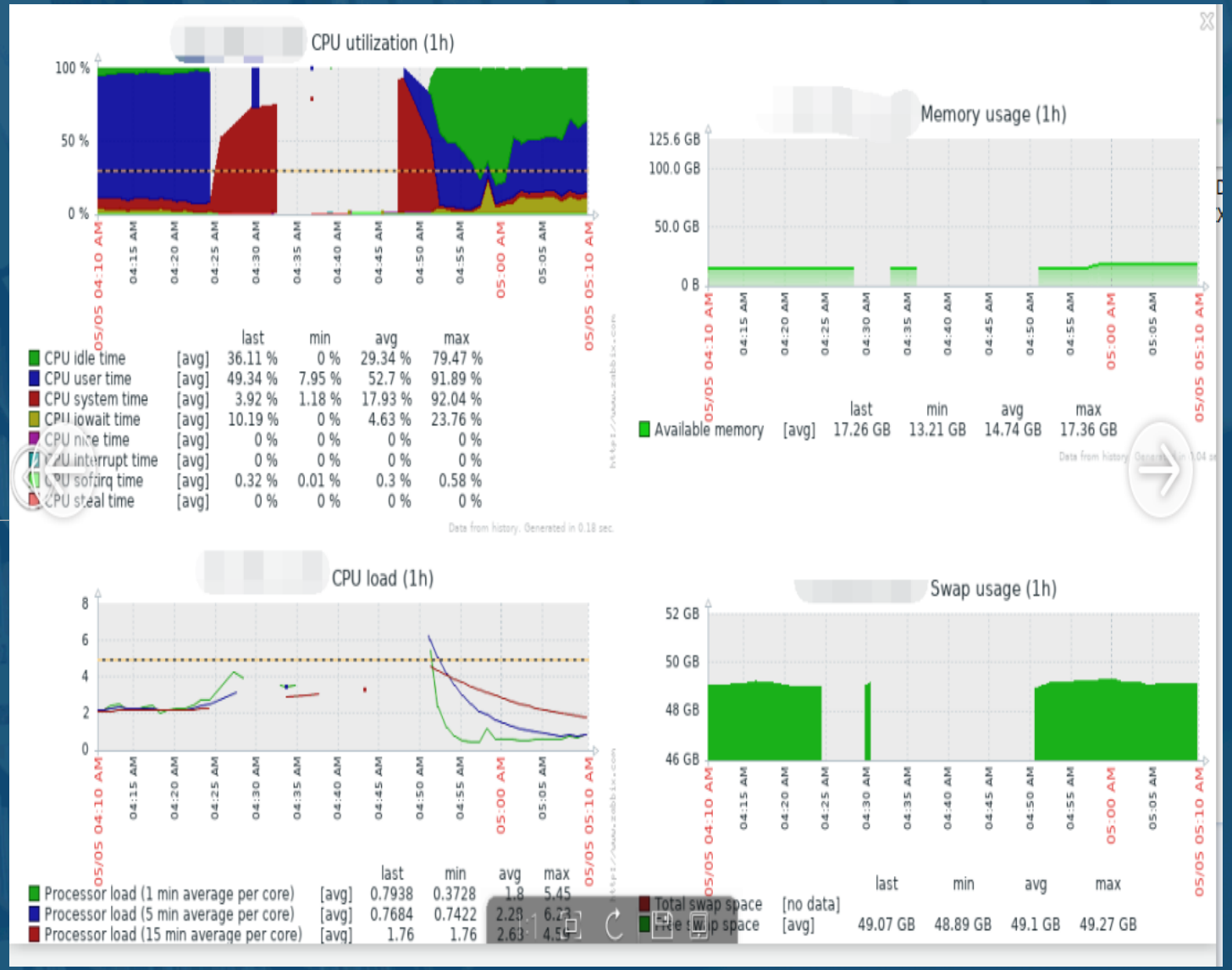
基于zabbix监控基础设施

内存

负载

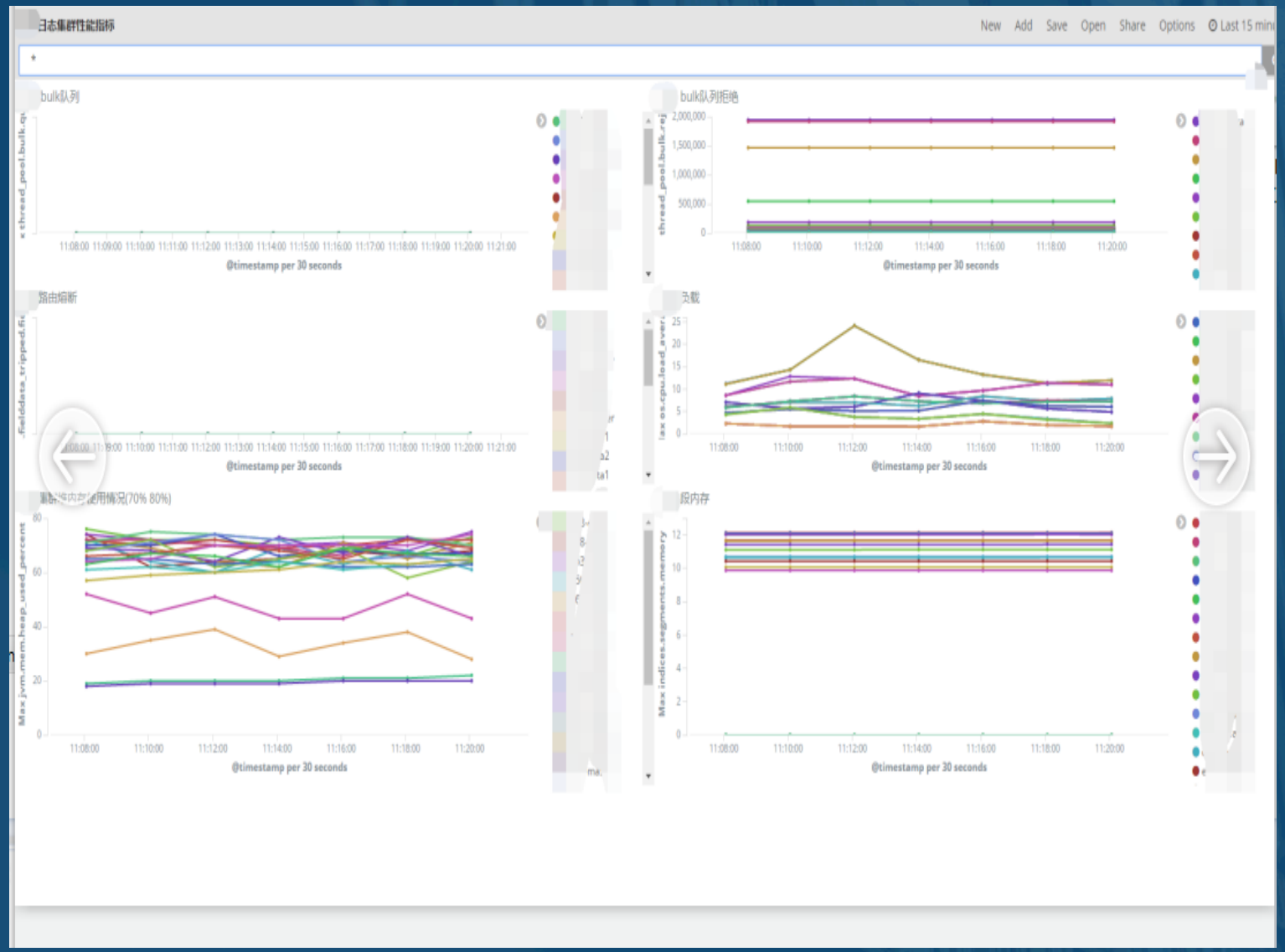
io wait

磁盘读写状态



ES集群监控

集群关键性能指标

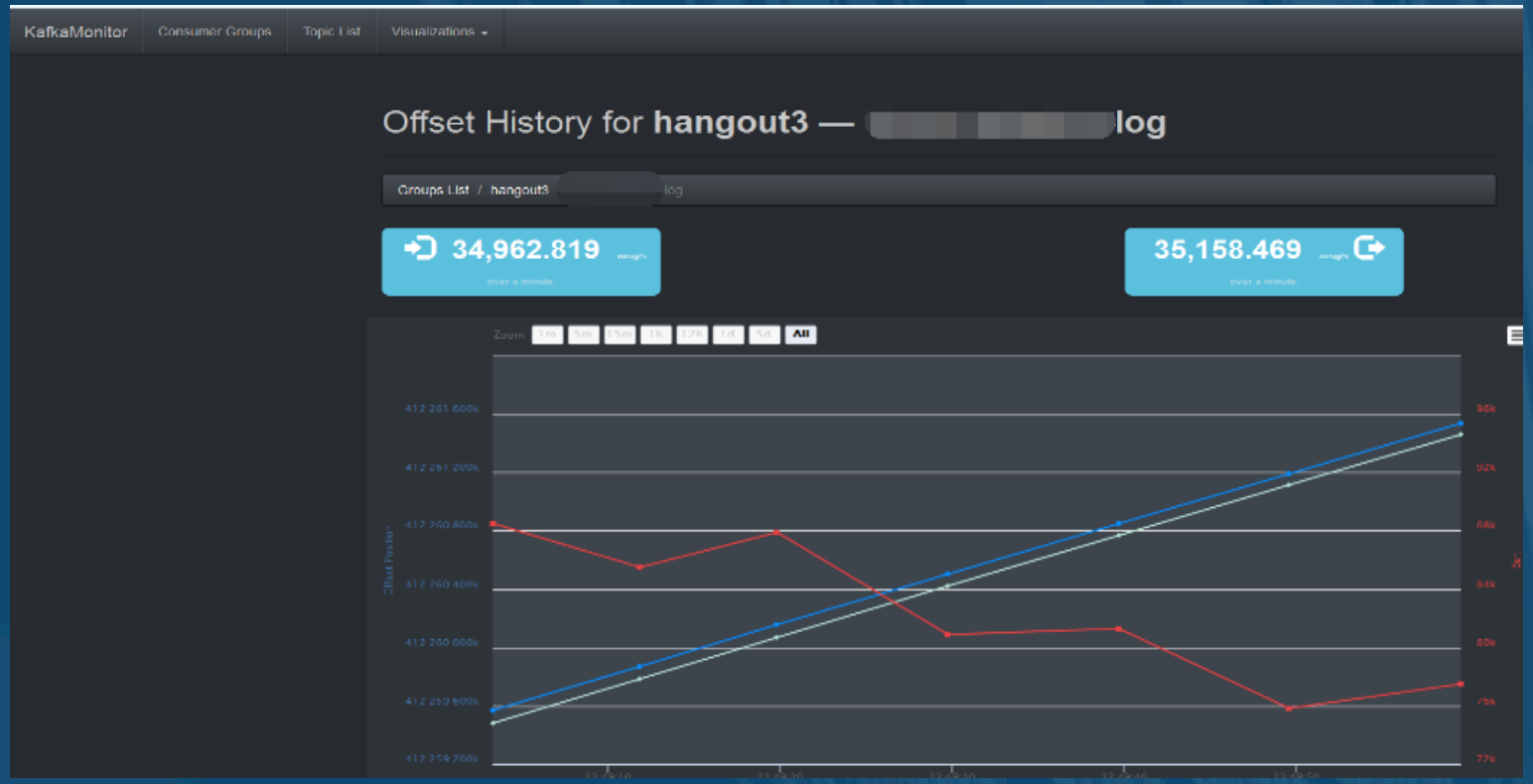


节点在线状态

【报警名称】:监控ES状态是否正常
【报警时间】:2017-05-05 13:23:00
【故障发生时间】:2017-05-05 13:23:00
【报警次数】:1
【报警级别】:警告
【故障信息】:ES状态异常， 错误信息：cluster nodes is unhealthy: the node es108-data1 is lost
【屏蔽报警】ocean-monitor/pauseAlarm/item/4?token=9f98742a3d5294cd19a57f5803bdb675&isWxGroup=false&concatId=2">屏蔽报警

故障已修复
item_validator：监控ES状态是否正常
故障时间：2017-05-05 13:23:00
修复时间：2017-05-05 13:26:00

kafka 监控



Kafka堆积监控

【报警名称】:Kafka offset -
log监控
【报警时间】:2017-05-04 21:50:00
【故障发生时间】:2017-05-04 21:50:00
【报警次数】:1
【报警级别】:警告
【故障信息】:Kafka offset-
nginx-log监控堆积，表达式：5888138.0, 阈值：{"threshold":"3000000"} 警告
【屏蔽报警】
0/ocean-monitor/pauseAlarm/item/119?token=ddc93f6cd9c80bff2c274adc77874192&isWxGroup=false&concatId=2">屏蔽报警

运维之索引管理



定期删除无用日志数据索引

定期关闭老的索引，节省内存

定期进行段合并，提升搜索速度

CONTENTS

03 *Part Three*

集　群　优　化



日志集群优化

- **集群部署优化**：角色分离，根据服务器内存合理部署实例。例如120g内存，可以选择部署两个30g的data节点和一个10g的master节点。
- **索引数据存储路径优化**：不同data实例，索引存储路径不要在同一个盘目录下，否则会引起io资源竞争激烈，服务器负载高，reject线程池提交，服务器响应慢。曾今在CentOS6上踩过这个坑。
- **避免主和备份副本在一台机器上**：多个data实例部署下，设置cluster.routing.allocation.same_shard.host: true。
- **集群ping参数优化**：增大集群ping连接超时时间，防止集群因为gc时间过长而导致的节点掉线。

```
PUT /_all/_settings
{
  "settings": {
    "index.unassigned.node_left.delayed_timeout": "5m"
  }
}
```

```
discovery.zen.ping_timeout: 200s
discovery.zen.fd.ping_timeout: 200s
discovery.zen.fd.ping_interval: 30s
discovery.zen.fd.ping_retries: 5
```

日志集群优化

- **合理设置translog flush时间**：减少io压力。
- **索引体积优化**：禁用_all减少索引体积，提升搜索效率，针对不同索引数据量设置不同分片，保证单个分片在1g-10g之内。
- **日志格式优化**：尽量使用json格式，避免使用grok解析造成索引慢，例如Nginx日志。所有Field类型尽量在模板定义好，曾今因为一个业务日志格式不当，使用key=value\u0001key=value……造成field不断新增，集群不断uptate mappings，数据延迟写入，最后集群雪崩。
- **多线程多进程提交数据**：使用bulk多线程提交数据，尽量保证一个批次提交在15mb以内，避免索引速度慢，避免触发路由熔断。
- **收集用户访问kibana日志**：一方面从架构安全考虑，另一方面根据访问日志优化搜索。

```
        at java.lang.Thread.run(Thread.java:745)
xception in thread "pool-3-thread-8" CircuitBreakingException[[parent] Data too large, data for [<transport_request>] would be larger than limit of [948313128
8.8gb]]
        at org.elasticsearch.indices.breaker.HierarchyCircuitBreakerService.checkParentLimit(HierarchyCircuitBreakerService.java:211)
        at org.elasticsearch.common.breaker.ChildMemoryCircuitBreaker.addEstimateBytesAndMaybeBreak(ChildMemoryCircuitBreaker.java:128)
        at org.elasticsearch.transport.TcpTransport.handleRequest(TcpTransport.java:1332)
        at org.elasticsearch.transport.TcpTransport.messageReceived(TcpTransport.java:1242)
        at org.elasticsearch.transport.netty4.Netty4MessageChannelHandler.channelRead(Netty4MessageChannelHandler.java:74)
        at io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:373)
        at io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:359)
        at io.netty.channel.AbstractChannelHandlerContext.fireChannelRead(AbstractChannelHandlerContext.java:351)
        at io.netty.handler.codec.ByteToMessageDecoder.fireChannelRead(ByteToMessageDecoder.java:293)
```

CONTENTS

04 *Part Four*

业　务　案　例

日志监控

业务接口接口

耗时中位数
最大耗时
请求量
调用链
GC耗时

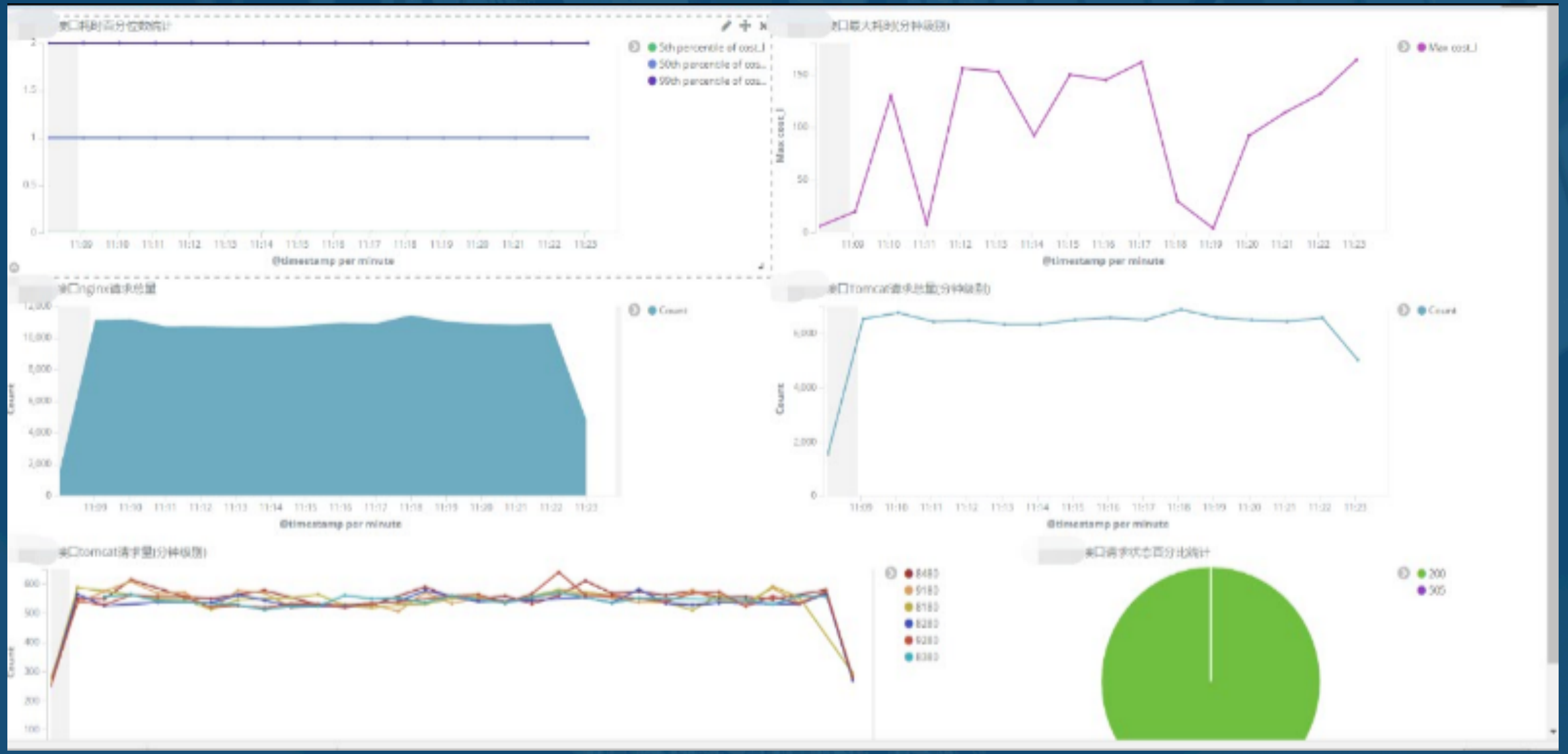
php-fpm日志

500 错误
消耗时间
消耗cpu
消耗mem
错误日志

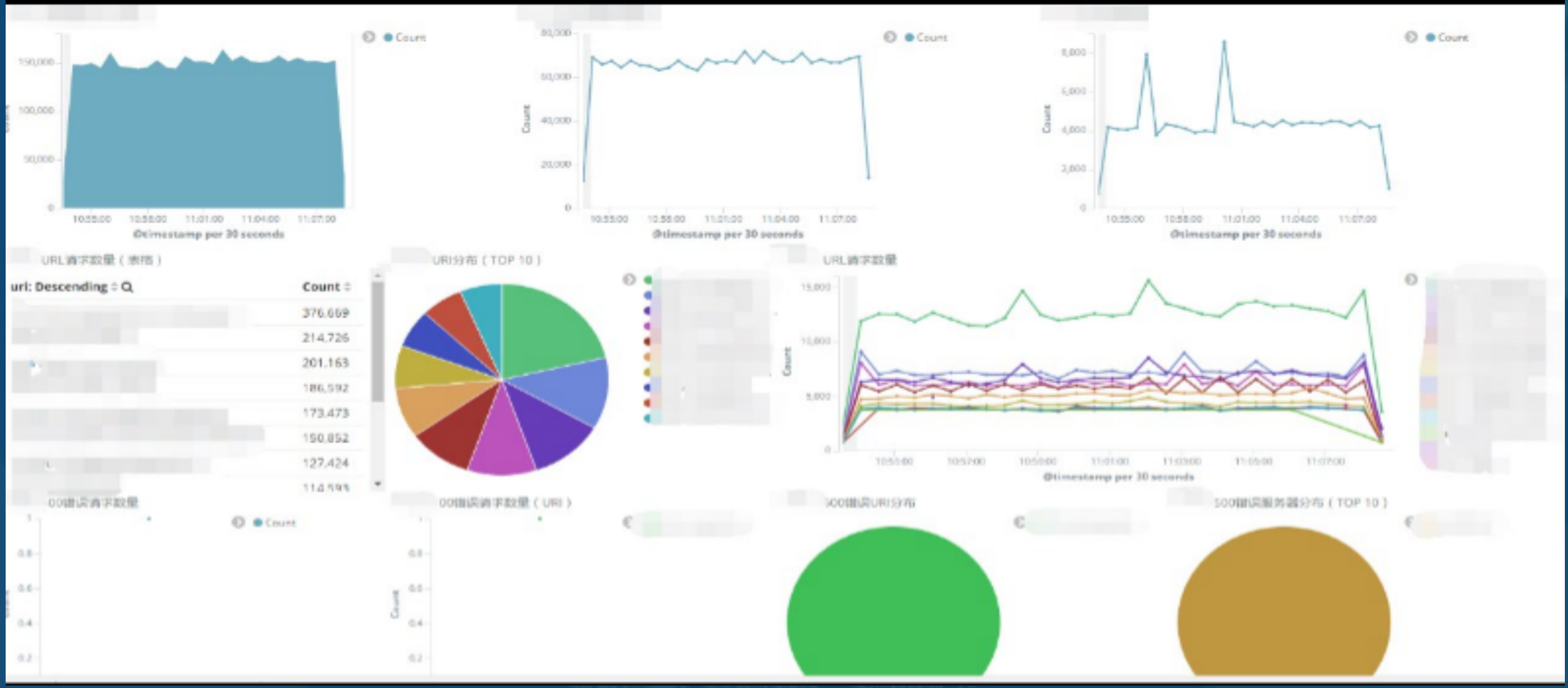
其他日志

Mysql慢查询日志
系统审计安全日志
RPC服务日志
kibana访问日志
应用日志
业务日志
……

接口耗时监控


@timestamp per minute
Count
Max cost
@timestamp per minute
Count
@timestamp per minute
Count
@timestamp per minute
Count
200
505

php-fpm日志监控


Count
@timestamp per 30 seconds
URL请求数量（表格）
url: Descending
Count
376,669
214,726
201,163
186,592
173,473
150,852
127,424
URI分布（TOP 10）
URL请求数量
@timestamp per 30 seconds
400错误请求数量
400错误请求数量（URI）
500错误URI分布
500错误服务器分布（TOP 10）

Next

日志指标系统

基于es-sql实现指标web分析系统，让除了开发人员，更多业务人员可以直接分析统计日志。

01

日志平台容器化

利用Docker和k8s搭建日志平台，做资源隔离和服务封装。

02

ELK运维系统

除了能够实现基本的监控告警，索引管理和优化以外，还能够实现常见故障的自愈处理、节点的自动踢除、节点的启动、与资产管理系统对接实现弹性扩容，agent端的可视化页面配置。

03

日志业务分析平台

除了基本的运维监控以外，能够处理常见的业务分析需求。如安全审计、多维度异常分析、动态趋势检测。

04

THANK YOU

