CAT
(Central Application Tracking)

美团点评基础架构中心 尤勇

自我介绍

- 尤勇 南京大学 资深技术专家
- 2010年加入美团点评 基础架构组
- 主要负责
  - CAT统一监控监控
  - 移动长连接接入层shark
  - 全链路压测平台

大纲

- **CAT介绍**
- CAT设计
- 最佳实践

CAT介绍

- CAT(Central Application Tracking)是基于Java开发的实时监控平台，主要包括移动端监控、应用侧监控等。

- CAT是一个给提供实时监控告警，移动以及后端应用性能分析诊断的工具。

监控分层

移动端监控
（返回码、城市、地区、运营商、版本、系统等多维度）

前端监控

业务监控
（业务核心指标监控，支付、验券）

应用层监控
（url、service、sql、cache 可用率、响应时间、qps等）

系统层监控（物理机，虚拟机）
（cpu memory network disk等）

基础设施监控（网络，交换机）
（网络流量，丢包，错包，连接数等）

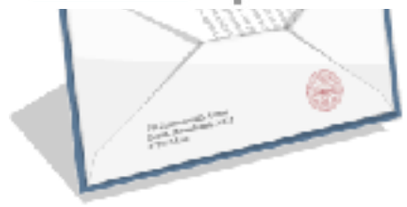
CAT

zabbix
falcon
等

实时系统

- 1、客户端日志不落地
- 2、服务端实时处理
- 3、客户端全量数据采集
- 整个系统从客户端产生消息到服务端产生实时报表延迟在**毫秒级别**

CAT的Logview



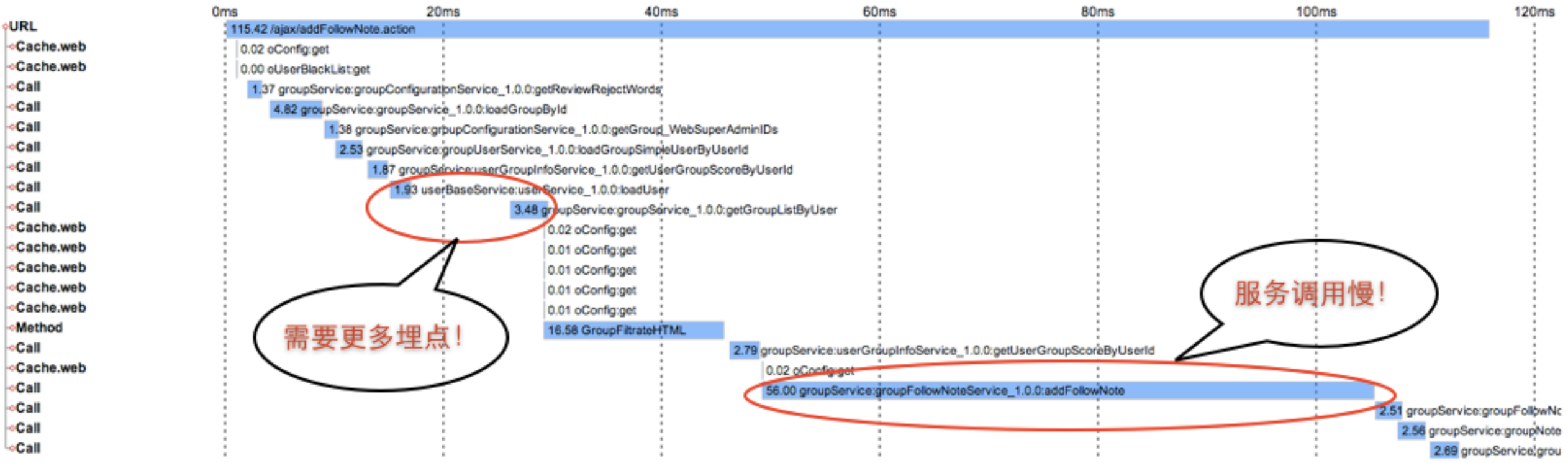
- 消息头
  - 版本号，消息ID，所属业务，IP，所在线程，根消息ID

```
t: Transaction Start
E: Event
T: Transaction End
A: Atomic Transaction
```

```
Transaction: 可嵌套
Event: 不可嵌套
Heartbeat: 不可嵌套
```

Type & timestamp	1st Category	2nd Category	Status	Duration & Attributes
t14:38:56.595	URL	t		
E14:38:56.595	URL.Server	cat.dianpingoa.com		RemoteIP= &Referer=http://cat.dianpin
E14:38:56.595	URL.Method	HTTP/GET		/cat/r/t?domain=&date=2012101314&reportType=
A14:38:56.595	MVC	InboundPhase		0.06ms
A14:38:56.595	MVC	TransitionPhase		0.00ms
t14:38:56.595	MVC	OutboundPhase		
t14:38:56.595	ModelService	CompositeTransactionService		
A14:38:56.596	ModelService	RemoteTransactionService		1.06ms http:// :8080/cat/r/model/transact
A14:38:56.596	ModelService	RemoteTransactionService		0.86ms http:// :8080/cat/r/model/transac
A14:38:56.596	ModelService	RemoteTransactionService		1.89ms http:// :8080/cat/r/model/transac
A14:38:56.596	ModelService	RemoteTransactionService		1.79ms http:// :8080/cat/r/model/transac
A14:38:56.596	ModelService	RemoteTransactionService		27ms http:// :8080/cat/r/model/transacti
T14:38:56.622	ModelService	CompositeTransactionService		27ms request=ModelRequest[domain=Cat, period
T14:38:56.628	MVC	OutboundPhase		33ms
T14:38:56.628	URL	t		33ms module=r&in=t&out=t

可视化Logview


0ms
20ms
40ms
60ms
80ms
100ms
120ms
URL
Cache.web
Cache.web
Call
Call
Call
Call
Call
Cache.web
Cache.web
Cache.web
Cache.web
Cache.web
Method
Call
Cache.web
Call
Call
Call
115.42 /ajax/addFollowNote.action
0.02 oConfig:get
0.00 oUserBlackList:get
1.37 groupService:groupConfigurationService_1.0.0:getReviewRejectWords
4.82 groupService:groupService_1.0.0:loadGroupById
1.38 groupService:groupConfigurationService_1.0.0:getGroup_WebSuperAdminIDs
2.53 groupService:groupUserService_1.0.0:loadGroupSimpleUserByUserId
1.87 groupService:userGroupInfoService_1.0.0:getUserGroupScoreByUserId
1.93 userBaseService:userService_1.0.0:loadUser
3.48 groupService:groupService_1.0.0:getGroupListByUser
0.02 oConfig:get
0.01 oConfig:get
0.01 oConfig:get
0.01 oConfig:get
0.01 oConfig:get
16.58 GroupFiltrateHTML
2.79 groupService:userGroupInfoService_1.0.0:getUserGroupScoreByUserId
0.02 oConfig:get
56.00 groupService:groupFollowNoteService_1.0.0:addFollowNote
2.51 groupService:groupFollowNc
2.56 groupService:groupNote
2.69 groupService:grou
需要更多埋点!
服务调用慢!

分布式Logview

t15:00:44.023	URL	/ajax/addVote.action	
E15:00:44.023	URL	ClientInfo	RemoteIP=180.175.162.12
E15:00:44.023	URL	Payload	HTTP/POST /ajax/addVot
A15:00:44.023	Cache.web	oConfig:get	0.02ms finalKey=oConfig.(
A15:00:44.023	Cache.web	oUserBlackList:get	0.00ms finalKey=oUserBla
t15:00:44.026	Call	groupService:groupSurveyService_1.0.0:addVote	

[:: hide ::]

t15:00:43.967	Service	groupService:groupSurveyService_1.0.0:addVote	
E15:00:43.967	PigeonRequest	Payload	
t15:00:43.967	SQL	GroupSurvey.loadSurvey	
E15:00:43.967	SQL.Method	Select	
E15:00:43.968	SQL.Database	jdbc:mysql://[redacted]?characterEncoding=(	
T15:00:43.967	SQL	GroupSurvey.loadSurvey	
t15:00:43.968	Call	userBaseService:userService_1.0.0:loadUser	

[:: hide ::]

t15:00:44.089	Service	userBaseService:us	
E15:00:44.089	PigeonRequest	Payload	
A15:00:44.089	Cache.memcached	eUserAtUC:get	
T15:00:44.089	Service	userBaseService:us	

[:: show ::]

T15:00:43.970	Call	userBaseService:userService_1.0.0:loadUser	
A15:00:43.970	Cache.memcached	oUserGroupScore:get	
t15:00:43.975	SQL	GroupSurvey.addVote	
E15:00:43.975	SQL.Method	Execute	
E15:00:44.244	SQL.Database	jdbc:mysql://[redacted]?characterEncoding=(	
T15:00:44.243	SQL	GroupSurvey.addVote	
T15:00:44.244	Service	groupService:groupSurveyService_1.0.0:addVote	

[:: show ::]

T15:00:44.305	Call	groupService:groupSurveyService_1.0.0:addVote	279ms CallType=sync
T15:00:44.307	URL	/ajax/addVote.action	284ms

应用监控报表（APM）

报表	说明
Transaction	一段代码运行时间、次数
Event	一行代码的执行次数
Problem	系统可能出现的异常，包括访问较慢的程序等
Business	多维度业务指标报表
Hearbeat	JVM内部一些状态信息，Memory，Thread等
API	一个请求调用链路统计
RPC	SOA系统用关于RPC调用的报表
Dependency	项目依赖关系视图
…	…

Transaction报表

- 支持项目、IP、TYPE、NAME 四层统计
- 框架层面统一接入了URL、RPC、SQL、Cache、Message等

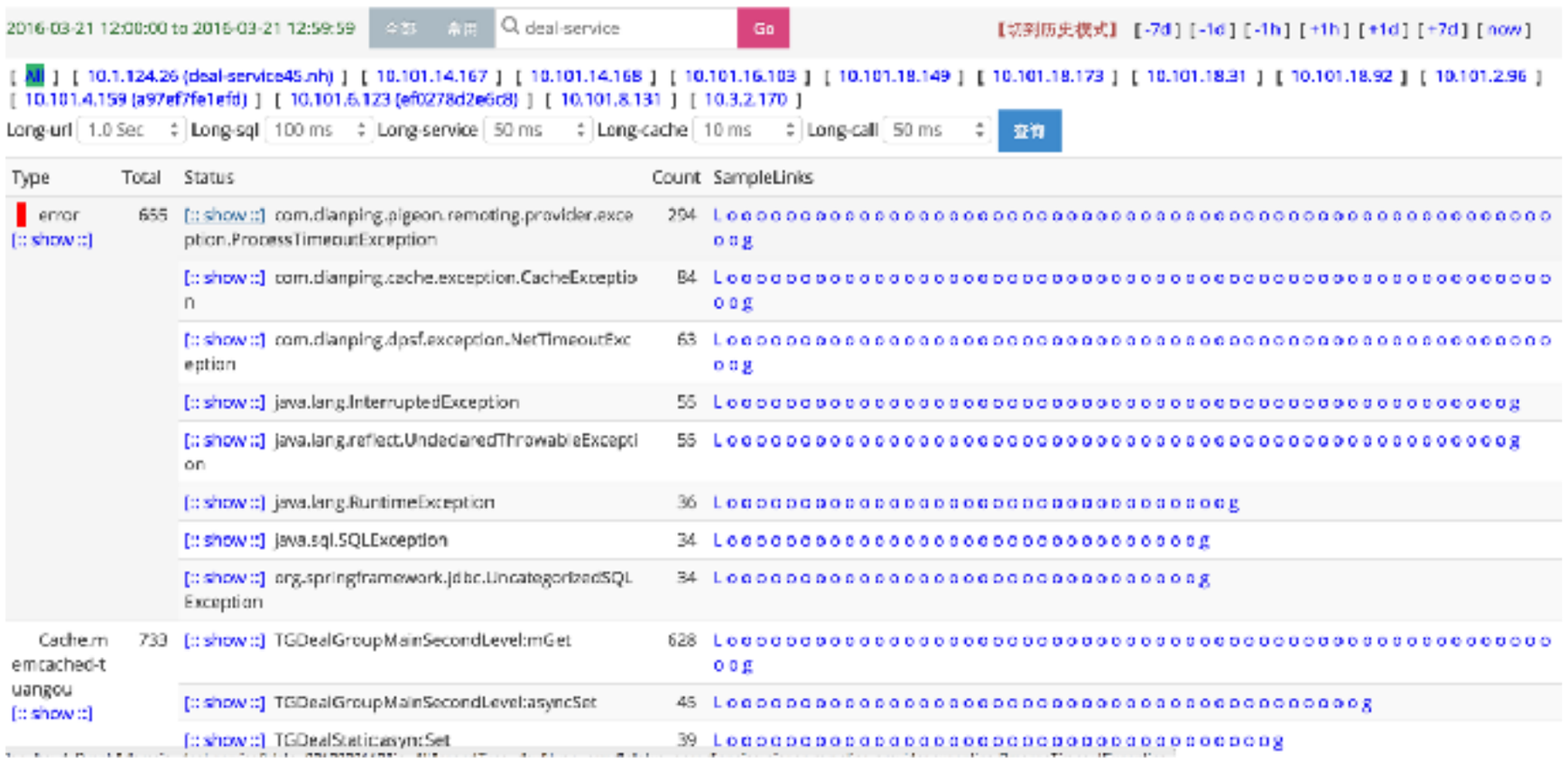
项目: Cat [切换][常用] 【报表时间】From 2013-09-16 12:00:00 to 2013-09-16 12:59:59 【切

Machines: [All] [[redacted].101.84] [[redacted].102] [[redacted].108] [[redacted]126] [[redacted].128] [[redacted].6.145] [[redacted]6.37] [[redacted]8.64]

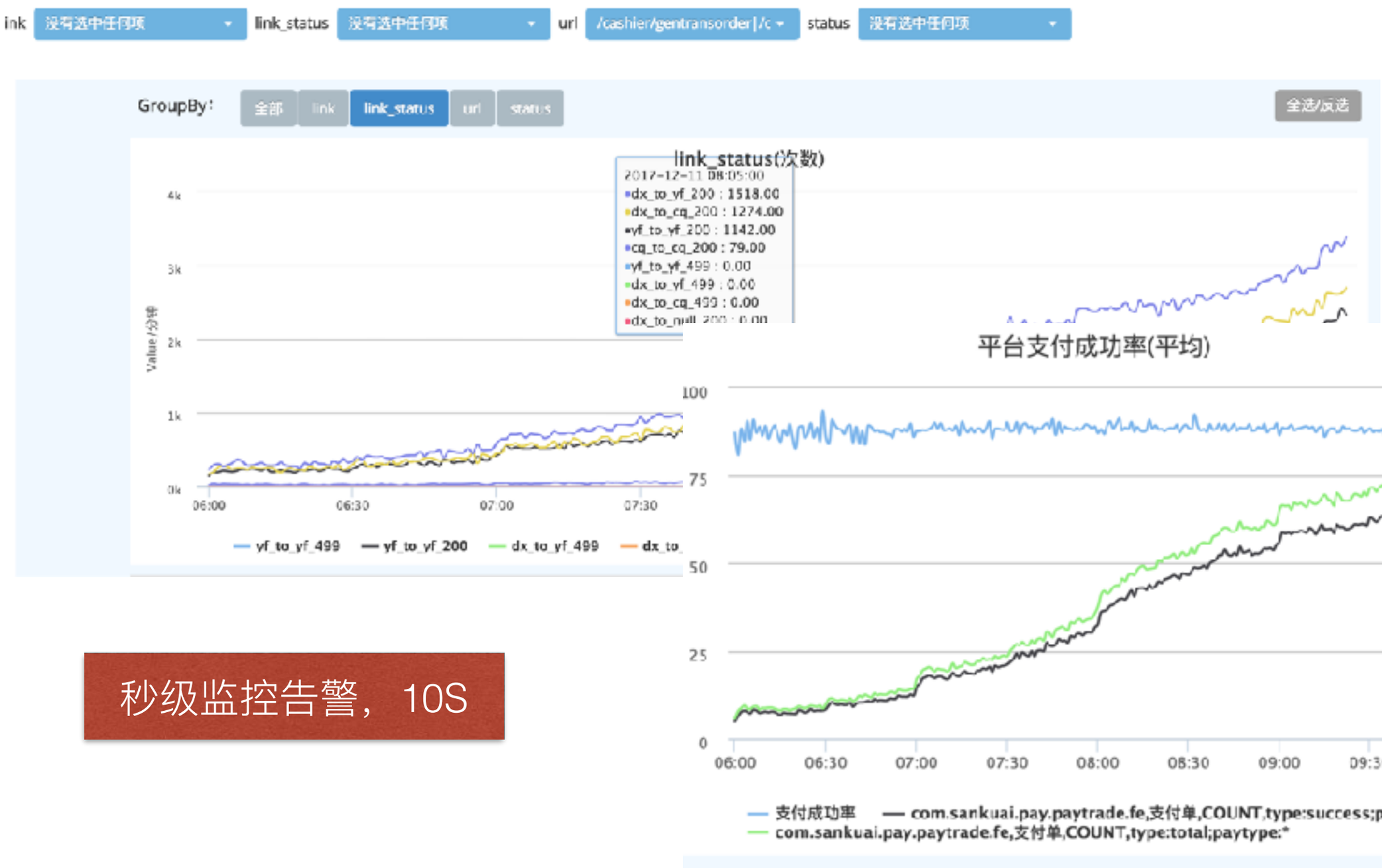
Type		Total Count	Failure Count	Failure%	Sample Link	Min(ms)	Max(ms)	Avg(ms)	95Line(ms)	99.9Line(ms)	Std(ms)	QPS
[:: show ::]	Checkpoint	113	0	0.0000%	Log View	0	88489.3	4021.8	35592.1	35592.1	11732.9	0.0
[:: show ::]	System	4,018	0	0.0000%	Log View	7.1	469587.7	1418.9	2458.0	304513.4	15837.9	1.3
[:: show ::]	Dependency	636	0	0.0000%	Log View	3.2	5430	975.3	2112.0	5430.0	767.0	0.2
[:: show ::]	Task	775	0	0.0000%	Log View	12.2	27520.4	173.4	373.0	27520.0	1281.6	0.2
[:: show ::]	BucketService	62	0	0.0000%	Log View	0.1	1079.5	167.1	580.0	1079.0	221.8	0.0
[:: show ::]	MetricAlert	53	0	0.0000%	Log View	28.1	130.9	49.1	115.0	130.0	25.0	0.0
[:: show ::]	SQL	21,855	0	0.0000%	Log View	0	3268.7	8.4	24.1	929.3	55.5	6.9
[:: show ::]	ModelService	581,564	142	0.0244%	Log View	0	1424	6.0	9.9	664.8	39.5	183.1
[:: show ::]	URL	530,411	0	0.0000%	Log View	0.1	3844.4	4.4	6.5	540.7	40.7	167.0
[:: show ::]	ABTest	424	0	0.0000%	Log View	1.5	22	4.0	6.1	11.5	1.9	0.1
[:: show ::]	MVC	1,591,233	0	0.0000%	Log View	0	3844.1	1.4	1.0	433.7	23.6	500.9
[:: show ::]	URL.Forward	530,411	0	0.0000%	Log View	0	3283.8	0.2	0.0	23.5	5.4	167.0
[:: show ::]	Gzip	22,404	0	0.0000%	Log View	0	85.8	0.1	0.0	2.7	0.9	7.1
[:: show ::]	Decode	22,403	0	0.0000%	Log View	0	155.6	0.1	0.0	1.9	1.2	7.1

Problem报表

- exception
- long-url
- long-sql
- long-service
- long-cache
- long-call
- Transaction fail

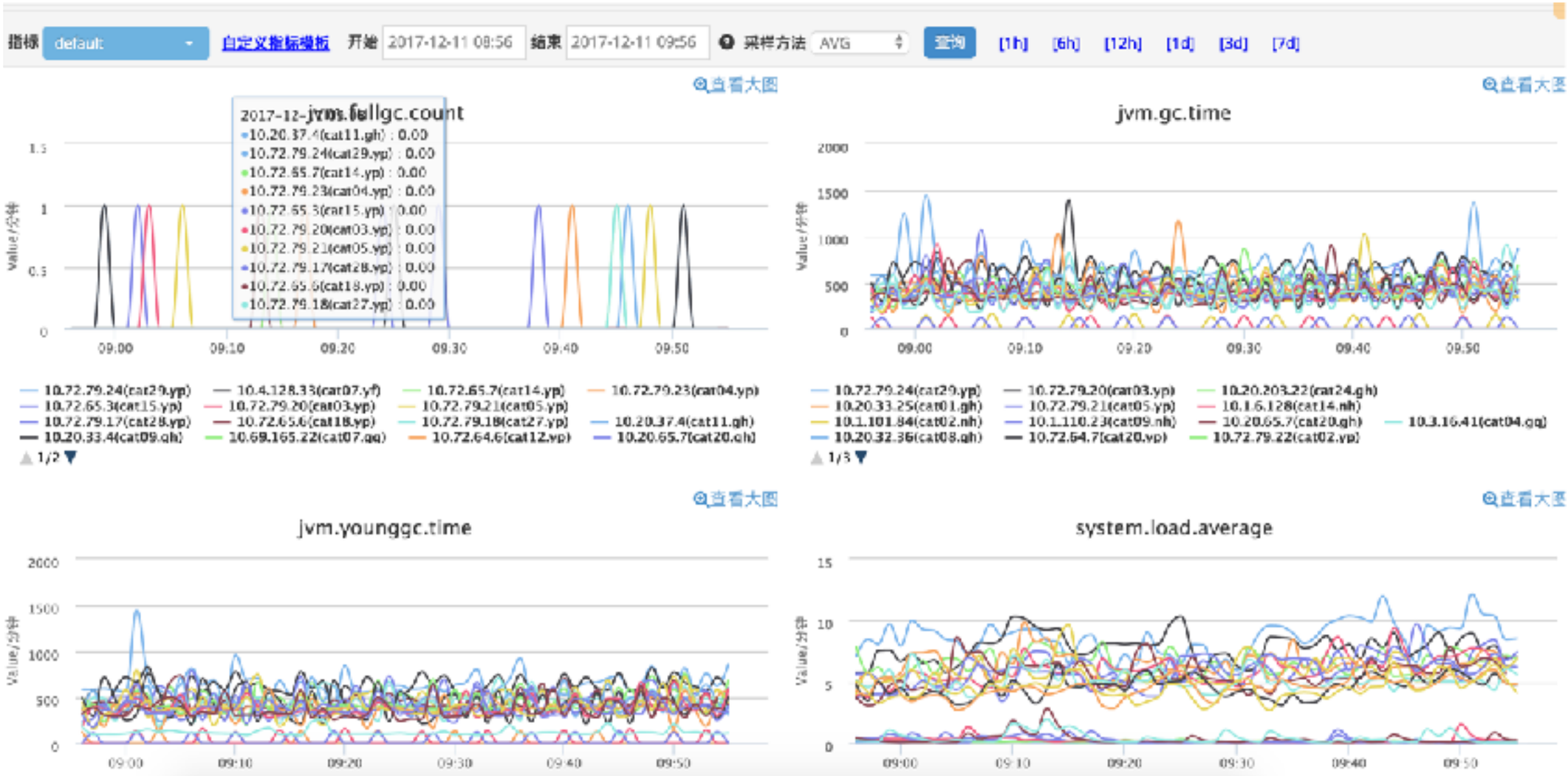


Business报表



秒级监控告警，10S

Heartbeat报表



大纲

- CAT历程
- CAT设计
- 最佳实践

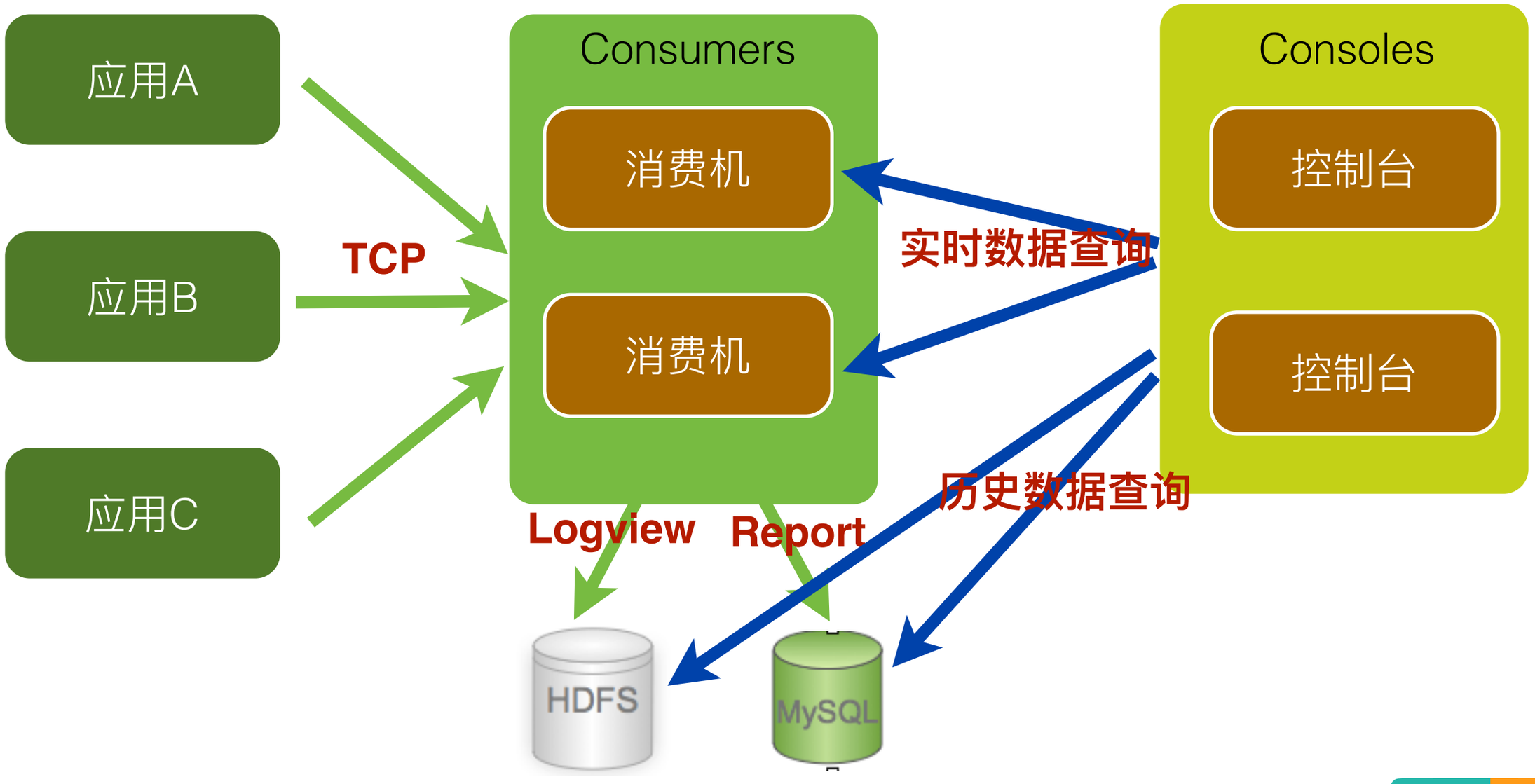
CAT设计

- 整体设计
- 客户端设计
- 服务端设计

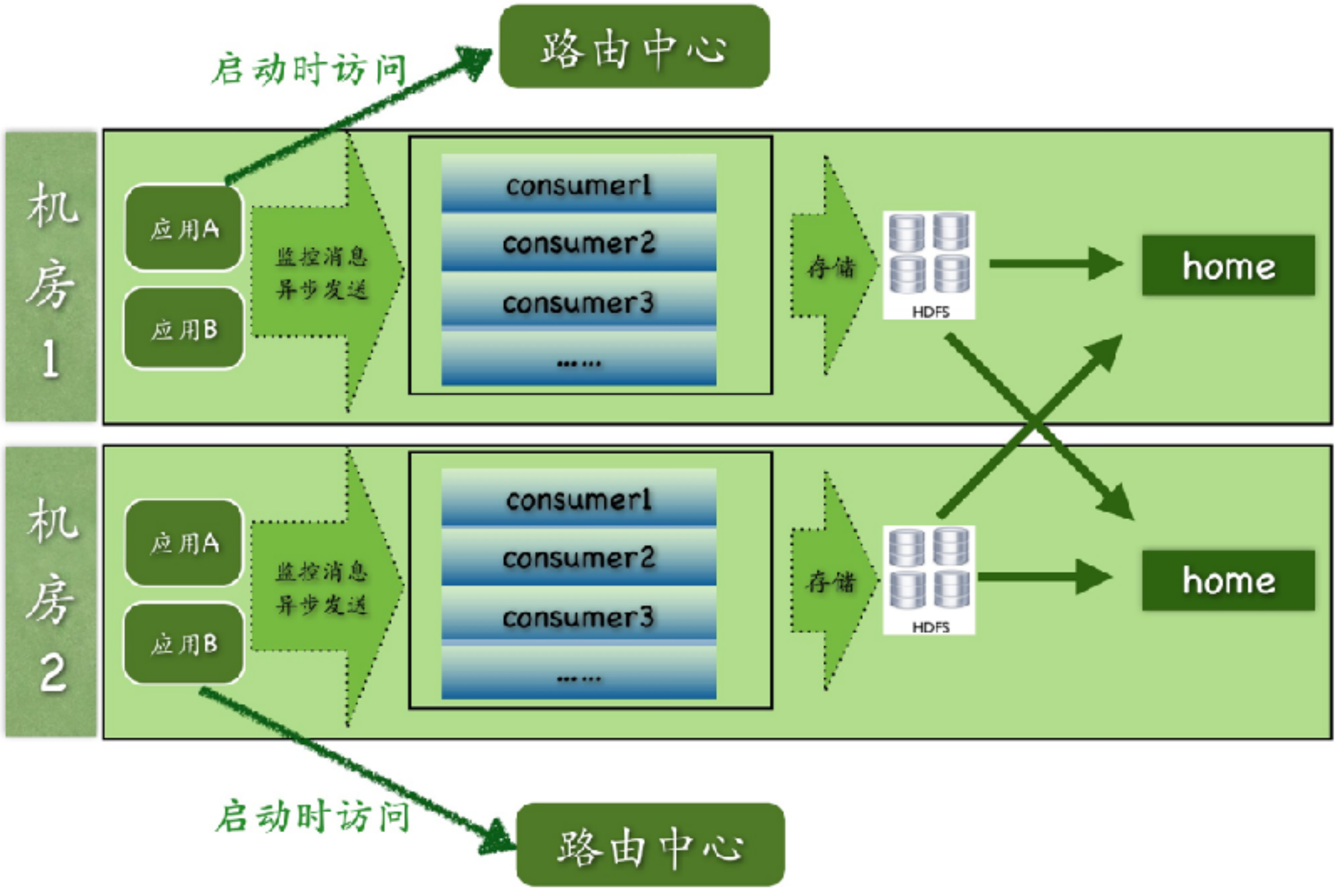
监控系统指标

- 对应用无影响（服务端上线、宕机等）
- 实时性（消息尽快到达服务端）
- 吞吐量（服务端高的吞吐量）
- 开销低（客户端尽可能开销低）（开销2%以内）
- 可靠性（消息100%到达服务端）
- 服务端处理100%的到达消息

整体设计1.0


应用A
应用B
应用C
TCP
Consumers
消费机
消费机
Consoles
控制台
控制台
实时数据查询
历史数据查询
Logview
Report
HDFS
MySQL

整体设计2.0

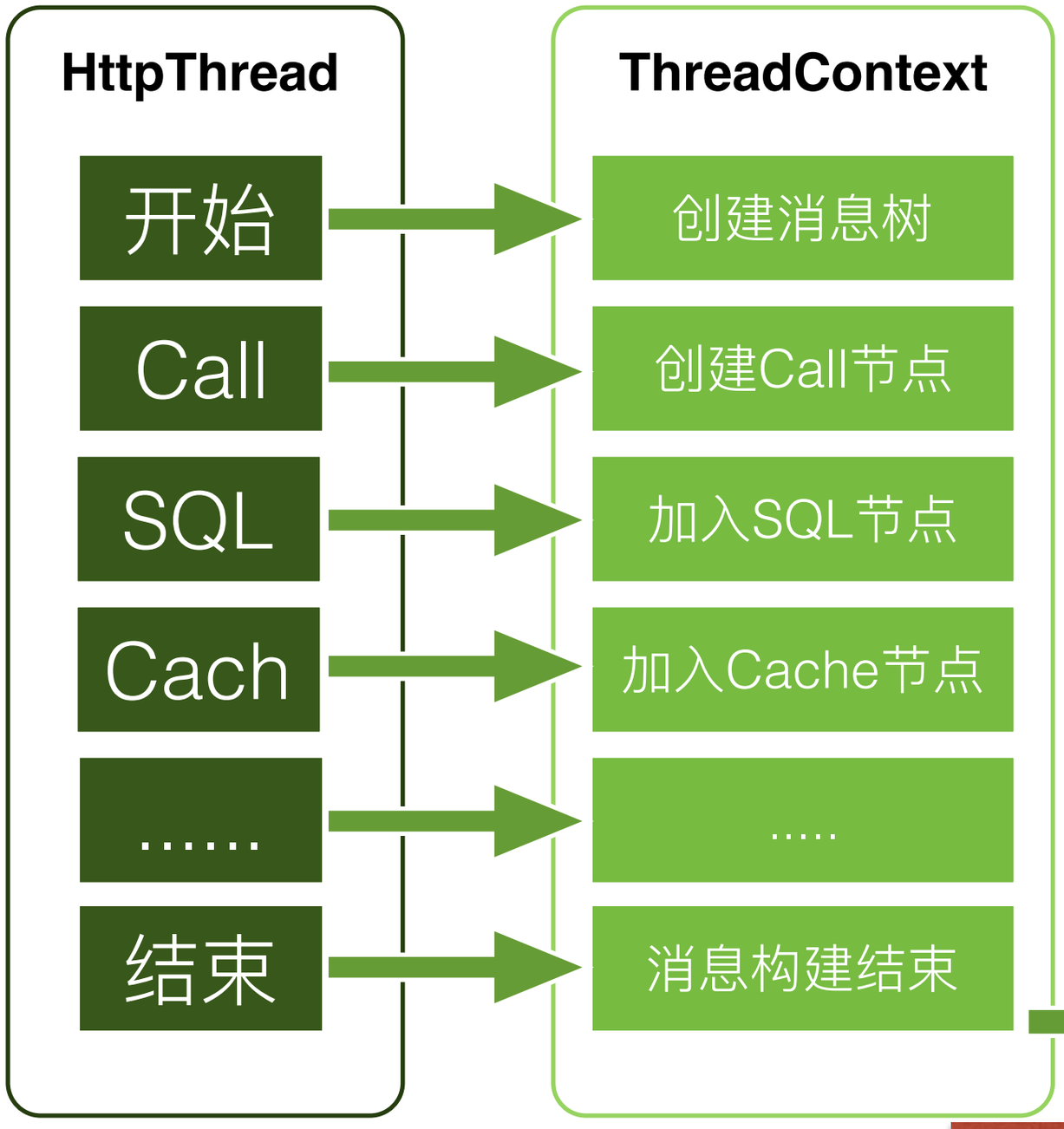

启动时访问
路由中心
机房1
应用A
应用B
监控消息
异步发送
consumer1
consumer2
consumer3
……
存储
HDFS
home
机房2
应用A
应用B
监控消息
异步发送
consumer1
consumer2
consumer3
……
存储
HDFS
home
启动时访问
路由中心

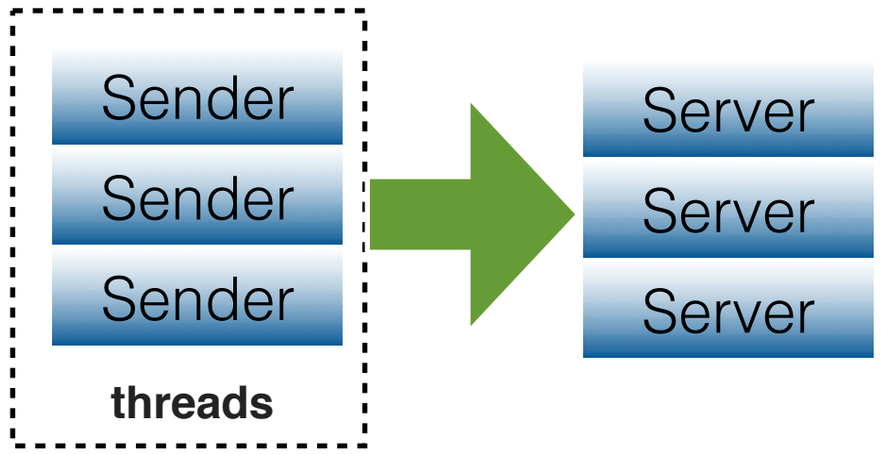
客户端设计

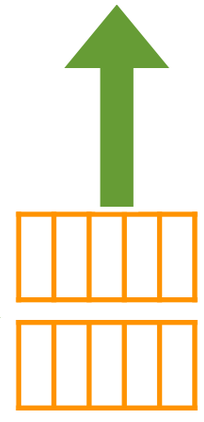
采样聚合

全链路压测隔离

异步线程聚合







优先级队列

自定义二进制协议

客户端重点

- 内存开销
  - 由于埋点问题，消息足够大
- CPU开销
  - 构建消息足够轻量，开销减低在2%
- 客户端没有做压缩
- 自定义协议序列化
- 基于netty实现消息传输

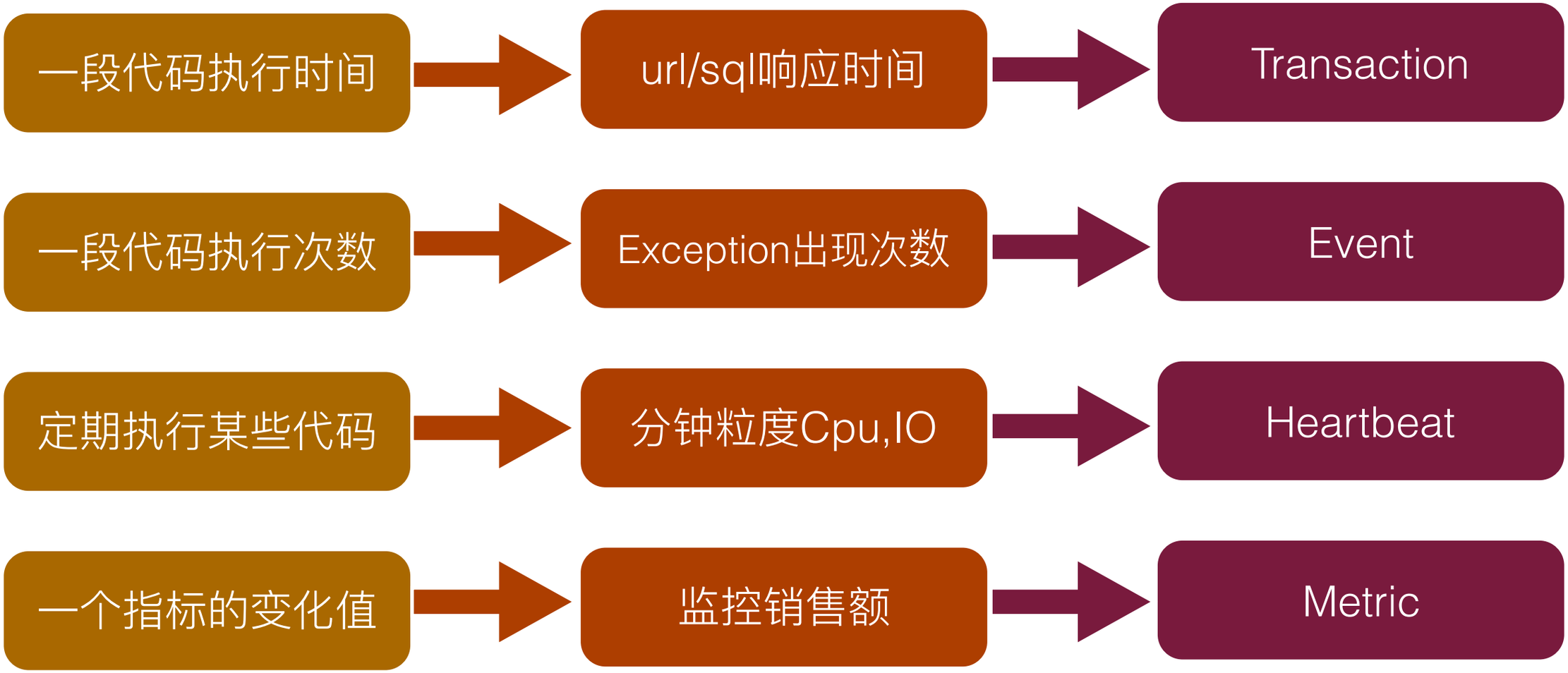
服务端重点

- 监控建模
- 报表建模
- CPU优化
- 数据存储

建模

- 监控领域数据模型
- 数据报表模型

监控建模


一段代码执行时间
url/sql响应时间
Transaction
一段代码执行次数
Exception出现次数
Event
定期执行某些代码
分钟粒度Cpu,IO
Heartbeat
一个指标的变化值
监控销售额
Metric

KeyValue的方式

- 后续扩展性较好
- 后续配置成本很高
- 后续计算成本很高

报表

- Transaction
- Event
- Problem
- Heartbeat
- ……

报表建模

- 目标模型定义
- 访问、转换和合并
- 模型持久化
- XML, JSON, Binary..
- 代码生成

```xml
<?xml version="1.0" encoding="UTF-8"?>
<model>
      <entity name="transaction-report" root="true">
            <attribute name="domain" value-type="String" key="true" />
            <attribute name="startTime" value-type="Date" />
            <attribute name="endTime" value-type="Date" />
            <entity-ref name="machine" type="map" names="machines" />
      </entity>
      <entity name="machine">
            <attribute name="ip" value-type="String" key="true"/>
            <entity-ref name="type" type="map" names="types" />
      </entity>
      <entity name="type">
            <attribute name="id" value-type="String" key="true" />
            <attribute name="total-count" value-type="int" />
            <attribute name="fail-count" value-type="int" />
            <attribute name="min" value-type="double" />
            <attribute name="max" value-type="double" />
            <attribute name="sum" value-type="double" />
            <attribute name="sum2" value-type="double" />
            <element name="success-message" value-type="String" />
            <element name="fail-message" value-type="String" />
            <entity-ref name="name" type="map" names="names" />
      </entity>
      ...
</model>
```

```java
public interface IVisitor {
   public void visitTransactionReport(TransactionReport transactionReport);
   public void visitMachine(Machine machine);
   public void visitType(TransactionType type);
   public void visitName(TransactionName name);
   public void visitRange(Range range);
   public void visitDuration(Duration duration);
}
```

模型遍历

```
public abstract class BaseVisitor implements IVisitor {
    @Override
    public void visitAllDuration(AllDuration allDuration) {
    }

    @Override
    public void visitDuration(Duration duration) {
    }

    @Override
    public void visitMachine(Machine machine) {
        for (TransactionType type : machine.getTypes().values()) {
            visitType(type);
        }
    }

    @Override
    public void visitName(TransactionName name) {
        for (Range range : name.getRanges().values()) {
            visitRange(range);
        }

        for (Duration duration : name.getDurations().values()) {
            visitDuration(duration);
        }

        for (AllDuration allDuration : name.getAllDurations().values()) {
            visitAllDuration(allDuration);
        }
    }

    @Override
    public void visitRange(Range range) {
    }
```

模型合并

```
public class TransactionReportMerger extends DefaultMerger {
    public TransactionReportMerger(TransactionReport transactionReport) {
        super(transactionReport);
    }

    @Override
    public void mergeDuration(Duration old, Duration duration) {
        old.setCount(old.getCount() + duration.getCount());
        old.setValue(duration.getValue());
    }

    @Override
    public void mergeMachine(Machine old, Machine machine) {
    }

    @Override
    public void mergeName(TransactionName old, TransactionName other) {
        long totalCountSum = old.getTotalCount() + other.getTotalCount();
        if (totalCountSum > 0) {
            double line95Values = old.getLine95Value() * old.getTotalCount() + other.getLine95Value()
                    * other.getTotalCount();
            double line99Values = old.getLine99Value() * old.getTotalCount() + other.getLine99Value()
                    * other.getTotalCount();

            old.setLine95Value(line95Values / totalCountSum);
            old.setLine99Value(line99Values / totalCountSum);
        }

        old.setTotalCount(totalCountSum);
        old.setFailCount(old.getFailCount() + other.getFailCount());
        old.setTps(old.getTps() + other.getTps());

        if (other.getMin() < old.getMin()) {
            old.setMin(other.getMin());
        }

        if (other.getMax() > old.getMax()) {
            old.setMax(other.getMax());
        }
```

cpu优化

```
protected static class DateHelper {
   private BlockingQueue<SimpleDateFormat> m_formats = new ArrayBlockingQueue<SimpleDateFormat>(20);

   private Map<String, Long> m_map = new ConcurrentHashMap<String, Long>();

   public String format(long timestamp) {
      SimpleDateFormat format = m_formats.poll();
      if (format == null) {
         format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss.SSS");
         format.setTimeZone(TimeZone.getTimeZone("GMT+8"));
      }

      try {
         return format.format(new Date(timestamp));
      } finally {
         if (m_formats.remainingCapacity() > 0) {
            m_formats.offer(format);
         }
      }
   }
```

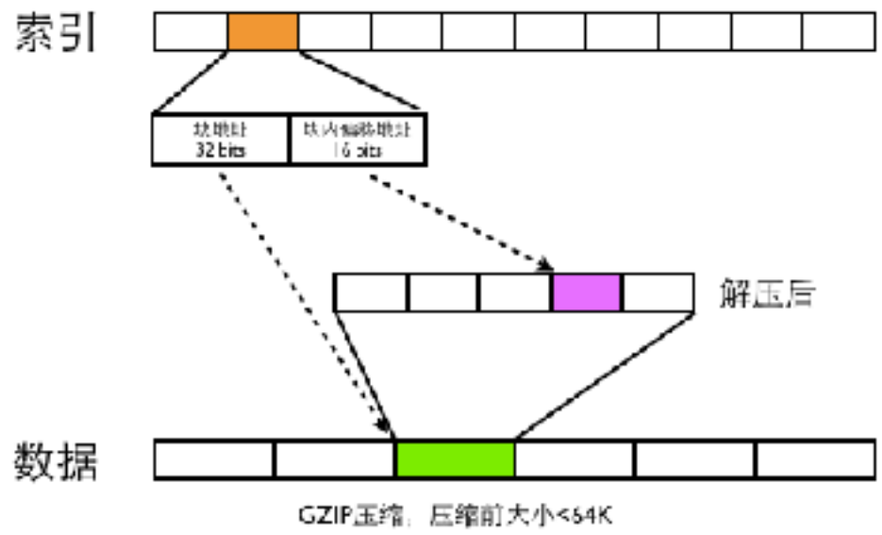
```
public long parse(String str) {
   int len = str.length();
   String date = str.substring(0, 10);
   Long baseline = m_map.get(date);

   if (baseline == null) {
      try {
         SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");

         format.setTimeZone(TimeZone.getTimeZone("GMT+8"));
         baseline = format.parse(date).getTime();
         m_map.put(date, baseline);
      } catch (ParseException e) {
         return -1;
      }

      long time = baseline.longValue();
      long metric = 1;
      boolean millisecond = true;

      for (int i = len - 1; i > 10; i--) {
         char ch = str.charAt(i);

         if (ch >= '0' && ch <= '9') {
            time += (ch - '0') * metric;
            metric *= 10;
         } else if (millisecond) {
            millisecond = false;
         } else {
            metric = metric / 100 * 60;
         }
      }
      return time;
   }
```


数据存储

- 顺序写、随机读
- 批量压缩提高压缩率

数据存储

- 消息ID：**ShopWeb-0a010680-375030-2**
  - 消息可能的存储路径
    - /2012/10/13/14/ShopSerivce-ShopWeb-10.1.6.1
    - /2012/10/13/14/ShopService-ShopWeb-10.1.6.2
    - 375030 => 2012-10-13 14:00:00
    - ShopService => 消息被记录的domain
    - 10.1.6.1/2 => 消息被处理的机器IP
    - 0a010680 => 10.1.6.128 用于保证消息ID唯一性



大纲

- CAT介绍
- CAT设计
- 最佳实践

MVP版本

- Demo 1个月
- MVP 3个月
- 重点解决最急迫的一个问题

一些不和谐的声音

- 客户端
- 业务的挑战（可靠，性能）
- 领导的挑战（当***时候，加一个动态开关）

数据质量

- 数据质量

- sql框架、cache框架、rpc框架、web框架

- 数据质量决定了监控质量

单机开发环境

- jetty server
- hdfs依赖
- mysql依赖

```
@RunWith(JUnit4.class)
public class TestServer extends JettyServer {
    public static void main(String[] args) throws Exception {
        TestServer server = new TestServer();
        System.setProperty("devMode", "true");
        server.startServer();
        server.startWebApp();
        server.stopServer();
    }

    @Before
    public void before() throws Exception {
        System.setProperty("devMode", "true");
        super.startServer();
    }

    @Override
    protected String getContextPath() {
        return "/cat";
    }

    @Override
    protected int getServerPort() {
        return 2281;
    }

    @Override
    protected void postConfigure(WebAppContext context) {
        context.addFilter(GzipFilter.class, "/*", Handler.ALL);
    }

    @Test
    public void startWebApp() throws Exception {
        // open the page in the default browser
        display("/cat/r");
        waitForAnyKey();
    }
}
```

最难的事情

- 项目上线推动
  - 如何推动整个项目上线（2-3人）
  - 部门之间沟通问题
  - 后续的支持和培训

开放生态

- 产品的scope
- 各种需求
- 系统开放生态

业务线汇总统计

业务线	业务线负责人	应用数	可用性低于99.9%的数量	可用性低于99.5%所占比例
到店综合用户与营销	hongwei.xia	147	4	2.721%
技术工程及基础数据平台	None	39	5	12.821%
人力资源及服务保障平台	None	22	0	0.0%
到店餐饮事业群	None	49	0	0.0%
外卖配送事业群(上海)	None	5	0	0.0%
未知bu		19	1	5.263%
酒店旅游事业群	None	26	2	7.692%

业务线	责任人	总数	成功数	成功率
交易前台-团购	[redacted]	25,564,634	25,434,559	99.491%
交易前台-支付	[redacted]	14,381	13,960	97.073%
交易前台-玩乐	[redacted]	2,073,438	2,071,467	99.905%
交易前台-电影	[redacted]	10,567,622	10,523,487	99.582%
交易前台-运营	[redacted]	124	124	100.0%
交易前台-汇总	[redacted]	38,220,199	38,043,597	99.538%
基础垂直-丽人	[redacted]	2,476,769	2,466,669	99.592%
基础垂直-结婚亲子家装	[redacted]	4,347,402	4,342,036	99.877%
基础垂直 酒店	[redacted]	3,910,053	3,900,772	99.763%
基础垂直 汇总	[redacted]	10,734,224	10,709,477	99.769%
大众微生活-微生活	[redacted]	766,481	765,276	99.843%
大众微生活 汇总	[redacted]	766,481	765,276	99.843%

CAT历程

- 2011-11月份　启动
- 2012-3月份　MVP模型
- 2012-6月份　正式上线
- 2012-12月份　150+应用　500+服务器
- 2013-12月份　400+应用　1500+服务器
- 2014-12月份　800+应用　3000+服务器
- 2015-9月份　1500+应用 7000+服务器
- 2016-6月份　2600+应用 12000+服务器
- 2017-6月份　6000+应用　50000+服务器

CAT总结

- 近5年时间，2-3个人
- 先做小做精，再做大做全
- 持续集成，持续发布，不断监控
- 单机开发和调试
- Everything Fails
- 关注客户，快速响应
- 站在巨人的肩膀上

CAT总结

- 2017年12月

- Apache License，国内超过百家公司在使用和评估
- 早期用户 **http://github.com/dianping/cat**

每天200TB日志，5000应用，50000+机器

每天60TB日志，3000应用，8000+机器

QA

- thank you