

—— World Of Tech 2017 ——

全球架构与运维技术峰会

2017年4月14日-15日 北京富力万丽酒店

ARCHITECTURE



出品人及主持人：

来 炜

滴滴出行基础平台部
技术总监

创新运维探索

分布式监控CAT美团 点评实践



尤勇

美团点评
技术工程部技术专家

分享主题：

美团点评分布式监控 CAT 系统架构演进

自我介绍

- 尤勇 硕士 南京大学
 - 2010年加入美团点评 基础架构组至今
- 资深技术专家
 - 目前主要负责CAT监控、移动接入层

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以及性能诊断工具，主要包括移动端监控、前端监控、服务端监控等。
- 今天侧重服务端监控体系介绍。

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

CAT服务端监控系统

- 1、客户端日志不落地
- 2、服务端实时处理
 - 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**
- 3、客户端接近全量数据采集



CAT的Logview

- 消息头

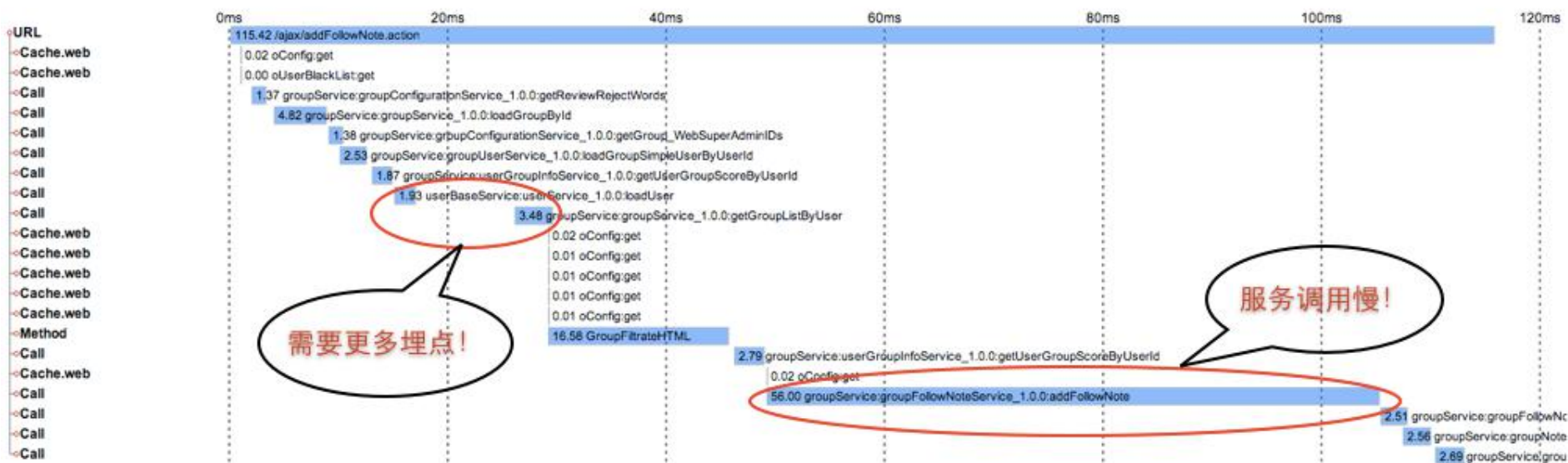
t: Transaction Start
E: Event
T: Transaction End
A: Atomic Transaction

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= [REDACTED] &Referer=http://cat.dianping	
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:// [REDACTED]:8080/cat/r/model/transact	
A14:38:56.596	ModelService	RemoteTransactionService	0.86ms http:// [REDACTED]:8080/cat/r/model/transac	
A14:38:56.596	ModelService	RemoteTransactionService	1.89ms http:// [REDACTED]:8080/cat/r/model/transac	
A14:38:56.596	ModelService	RemoteTransactionService	1.79ms http:// [REDACTED]:8080/cat/r/model/transac	
A14:38:56.596	ModelService	RemoteTransactionService	27ms http:// [REDACTED]: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



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/addVote
A15:00:44.023	Cache.web	oConfig:get	0.02ms finalKey=oConfig.1
A15:00:44.023	Cache.web	oUserBlackList:get	0.00ms finalKey=oUserBlac
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=U	
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=U	
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、TP95、TP99、\
Event	一行代码的执行次数
Problem	系统出现异常，包括访问较慢的重要节点
Business	多维度的业务指标监控
Hearbeat	JVM内部一些状态信息，Memory，Thread等
RPC	SOA系统用关于RPC调用的报表
Cache	缓存实时分析统计
Database	数据库实时分析统计
API	API依赖分析报表

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

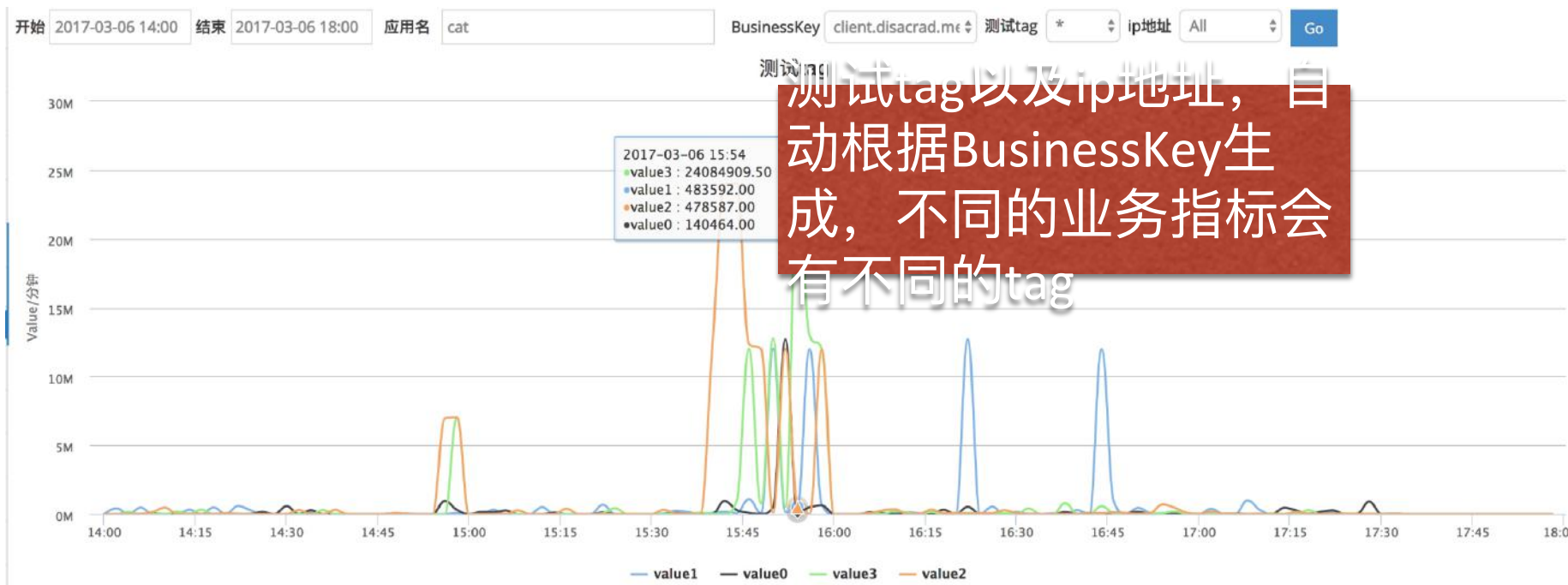
项目: Cat [切换] [常用] 【报表时间】 From 2013-09-16 12:00:00 to 2013-09-16 12:59:59 【切												
Machines: [All] [101.84] [.102] [.108] [126] [5.128] [6.145] [6.37] [8.64]												
Type	Total Count	Failure Count	Failure%	Sample Link	Min(ms)	Max(ms)	Avg(ms)	95Line(ms)	99.9Line(ms)	Std(ms)	QPS	
[:: show ::] 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

[illegible]

Business报表



RPC报表

调用其他RPC服务						服务端数据					
客户端数据						服务端数据					
Remote	Total	Failure	Failure%	Avg(ms)	QPS	Total	Failure	Failure%	Avg(ms)	QPS	
[:: show ::] All	15,480	0	0.0000%	3.67	6.66	15,433	0	0.0000%	2.60	6.64	
[:: show ::] tp-dock-facade-service	58	0	0.0000%	21.93	0.02	58	0	0.0000%	20.95	0.02	
[:: show ::] merchant-member-rpc-service	720	0	0.0000%	9.78	0.31	720	0	0.0000%	8.37	0.31	
[:: show ::] shop-server	44	0	0.0000%	6.89	0.02	3	0	0.0000%	7.00	0.00	
[:: show ::] hui-couponoffer-rpc-service	8,391	0	0.0000%	3.54	3.61	8,391	0	0.0000%	2.64	3.61	
[:: show ::] tp-dock-business-service	2,231	0	0.0000%	3.01	0.96	2,230	0	0.0000%	1.79	0.96	
[:: show ::] Swallow	4,031	0	0.0000%	2.91	1.73	4,031	0	0.0000%	1.68	1.73	
[:: show ::] Unknown	5	0	0.0000%		0.00	0	0	0.0000%		0.00	

调用的RPC服务应用列表

数据来自

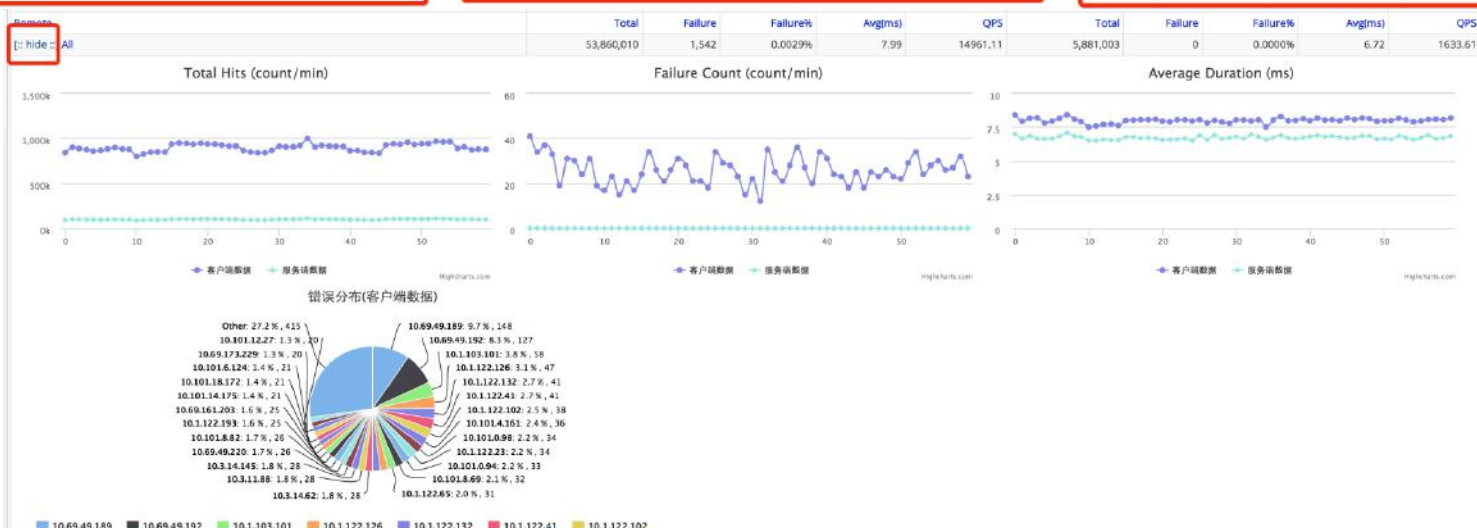
数据来自

提供RPC服务						服务端数据					
客户端数据						服务端数据					
Remote	Total	Failure	Failure%	Avg(ms)	QPS	Total	Failure	Failure%	Avg(ms)	QPS	
[:: show ::] All	150,142	2	0.0013%	3.18	64.60	150,193	0	0.0000%		64.63	
[:: show ::] Unknown	0	0	0.0000%	0.00	0.00	10	0	0.0000%	4045.80	0.00	
[:: show ::] hui-publish-service	4,773	2	0.0419%	12.75	2.05	4,780	0	0.0000%	10.01	2.06	
[:: show ::] hui-open-service	367	0	0.0000%	5.93	0.16	367	0	0.0000%	4.73	0.16	
[:: show ::] hui-business-service	5,459	0	0.0000%	5.09	2.35	5,461	0	0.0000%	3.80	2.35	
[:: show ::] mopay-service	3,209	0	0.0000%	4.24	1.38	3,209	0	0.0000%	3.14	1.38	
[:: show ::] hui-settle-service	2,953	0	0.0000%	3.85	1.27	2,954	0	0.0000%	2.87	1.27	
[:: show ::] hui-open-business-web	88,173	0	0.0000%	2.77	37.94	88,191	0	0.0000%	1.75	37.95	
[:: show ::] hui-mt-mis-web	396	0	0.0000%	2.69	0.17	397	0	0.0000%	1.46	0.17	
[:: show ::] mt-merchant-mopay-web	42,132	0	0.0000%	2.63	18.13	42,132	0	0.0000%	1.35	18.13	
[:: show ::] hui-merchant-service	2,680	0	0.0000%	2.26	1.15	2,692	0	0.0000%	1.09	1.16	

被这些应用调用RPC服务

RPC客户端(调用方)打点

RPC服务端(被调用方)打点



大纲

- CAT历程
- CAT设计
- 最佳实践

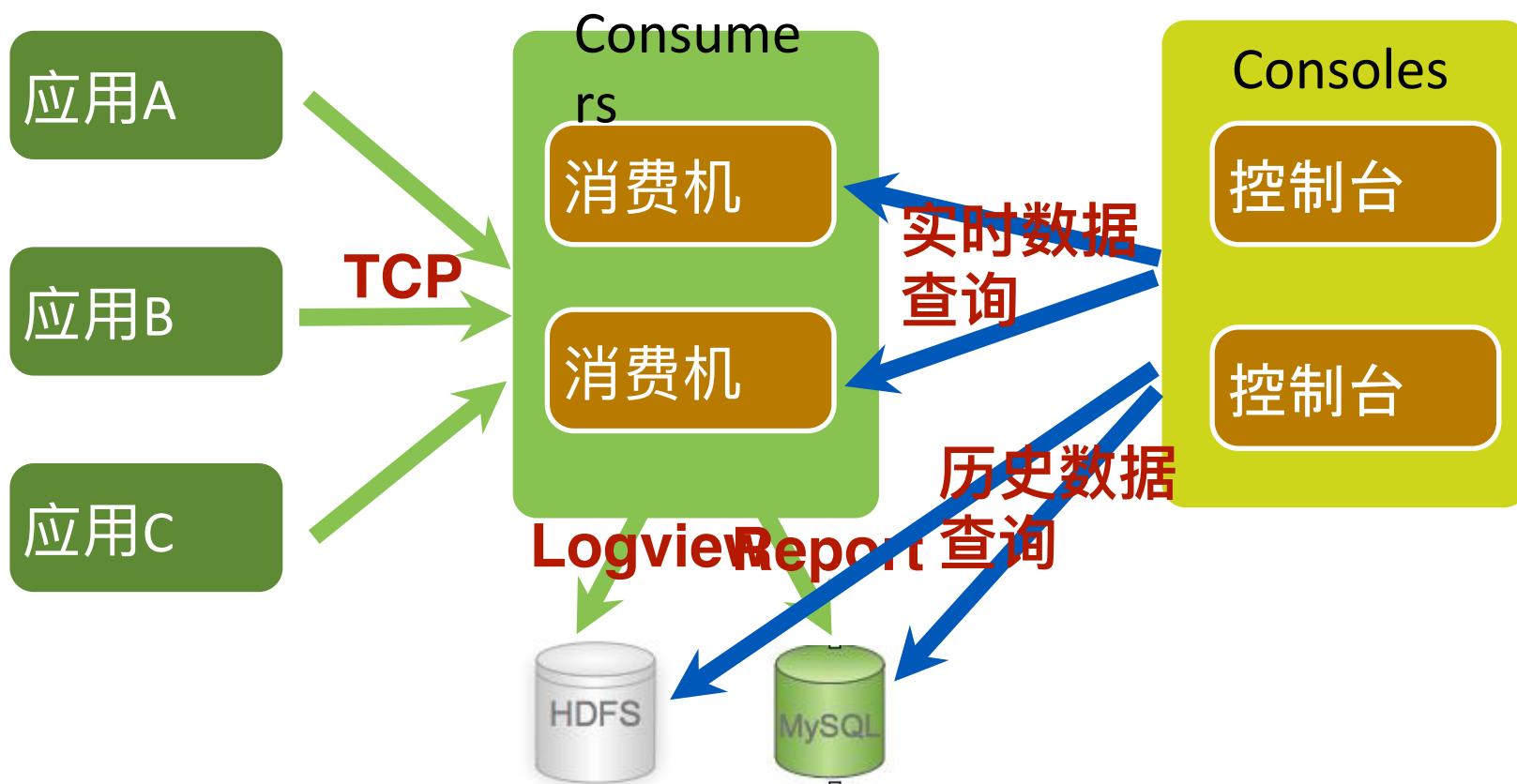
CAT设计

- 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



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



MySQL

home

机房2

应用A

应用B

监控消息
异步发送

consumer1
consumer2
consumer3
....

存储



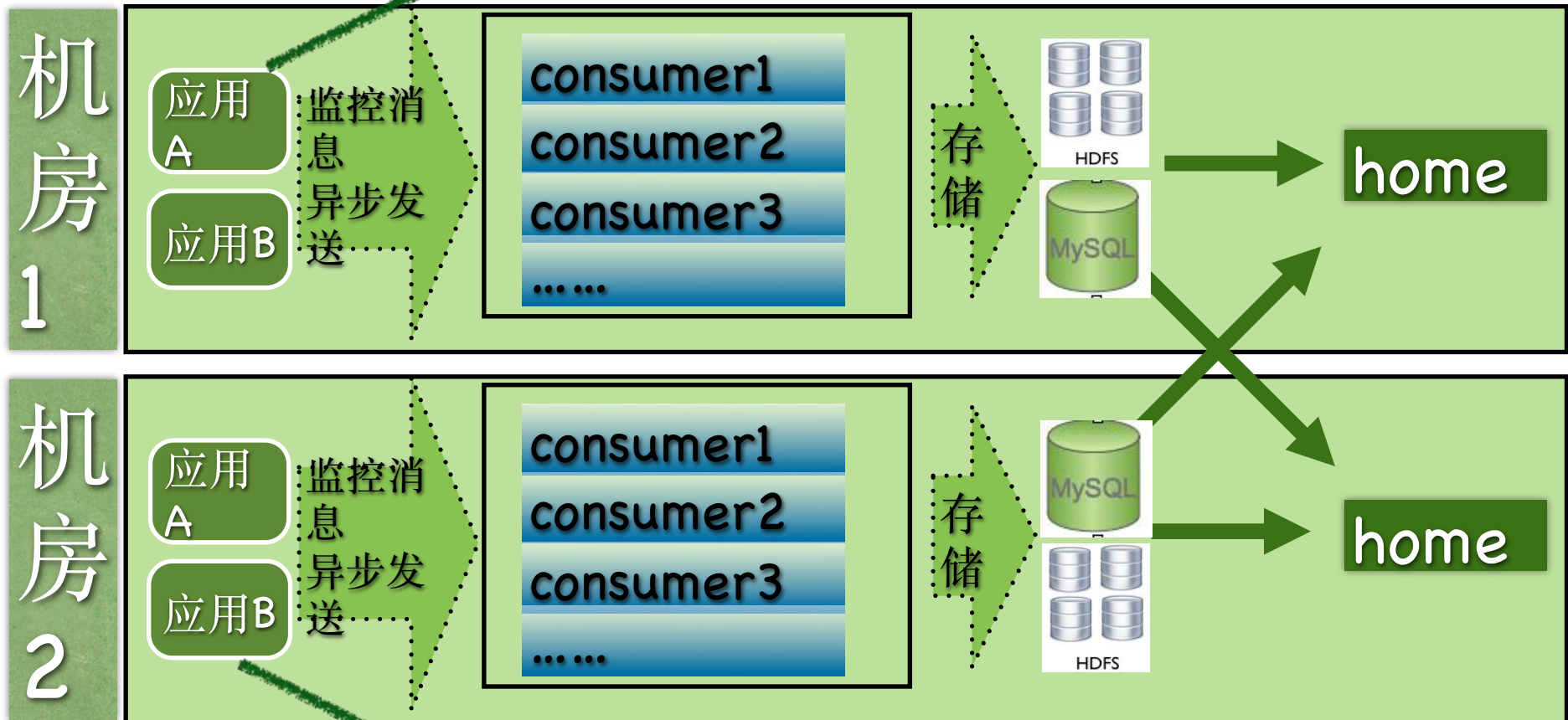
MySQL



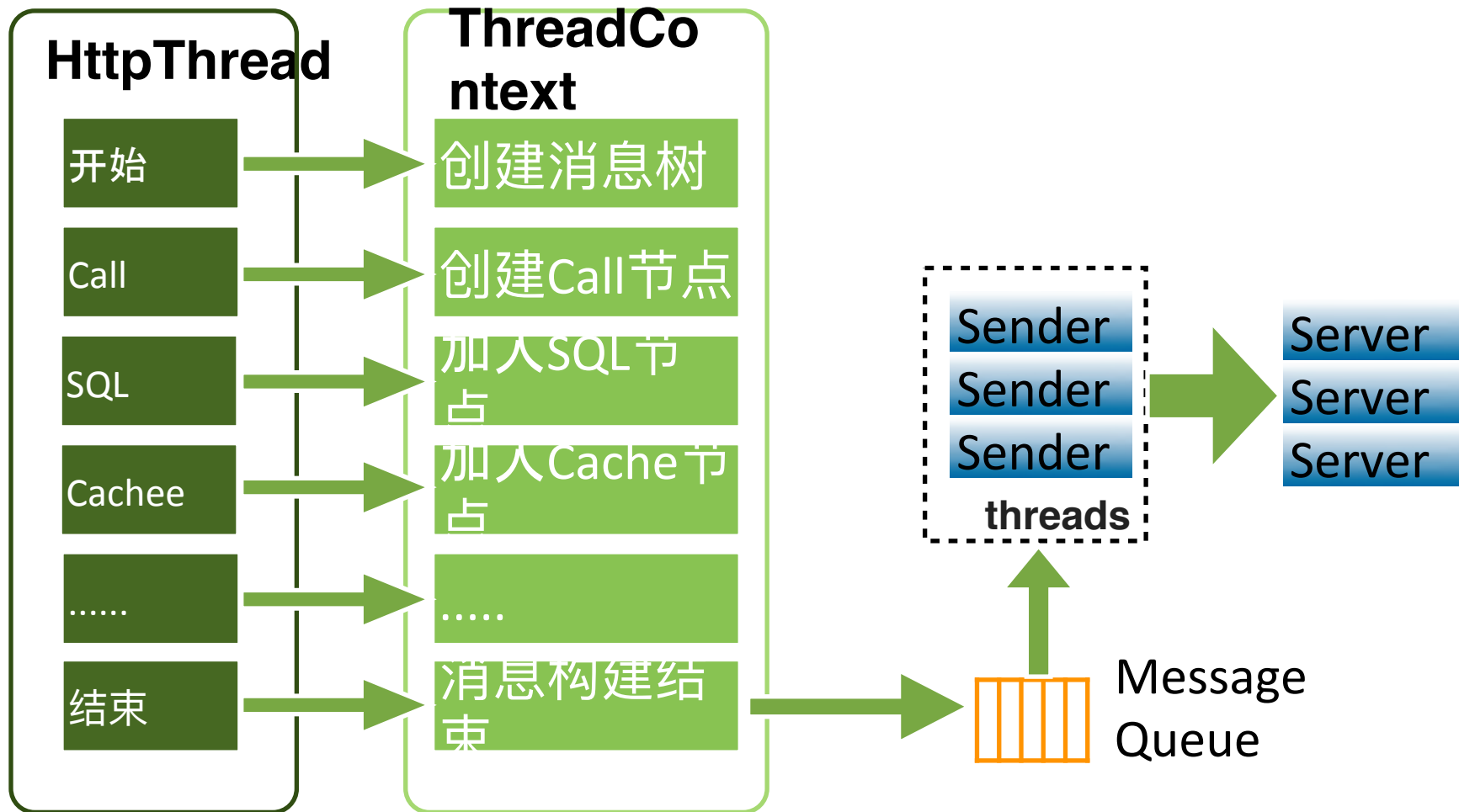
HDFS

home

启动时访问路由中心



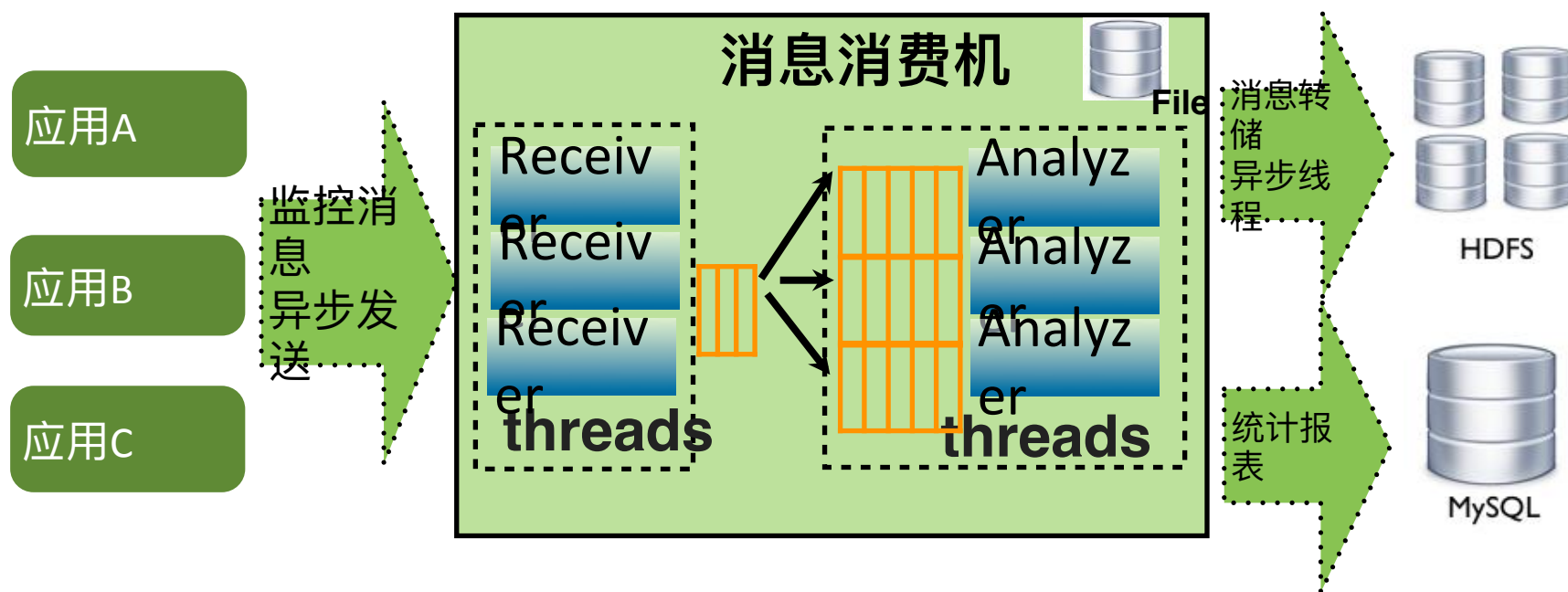
客户端设计



客户端重点

- 不影响业务，任何极端场景
- 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%
- 自定义序列化，文本协议 -> 二进制协议
- 客户端采样和聚合
- 基于netty实现消息传输

服务端设计



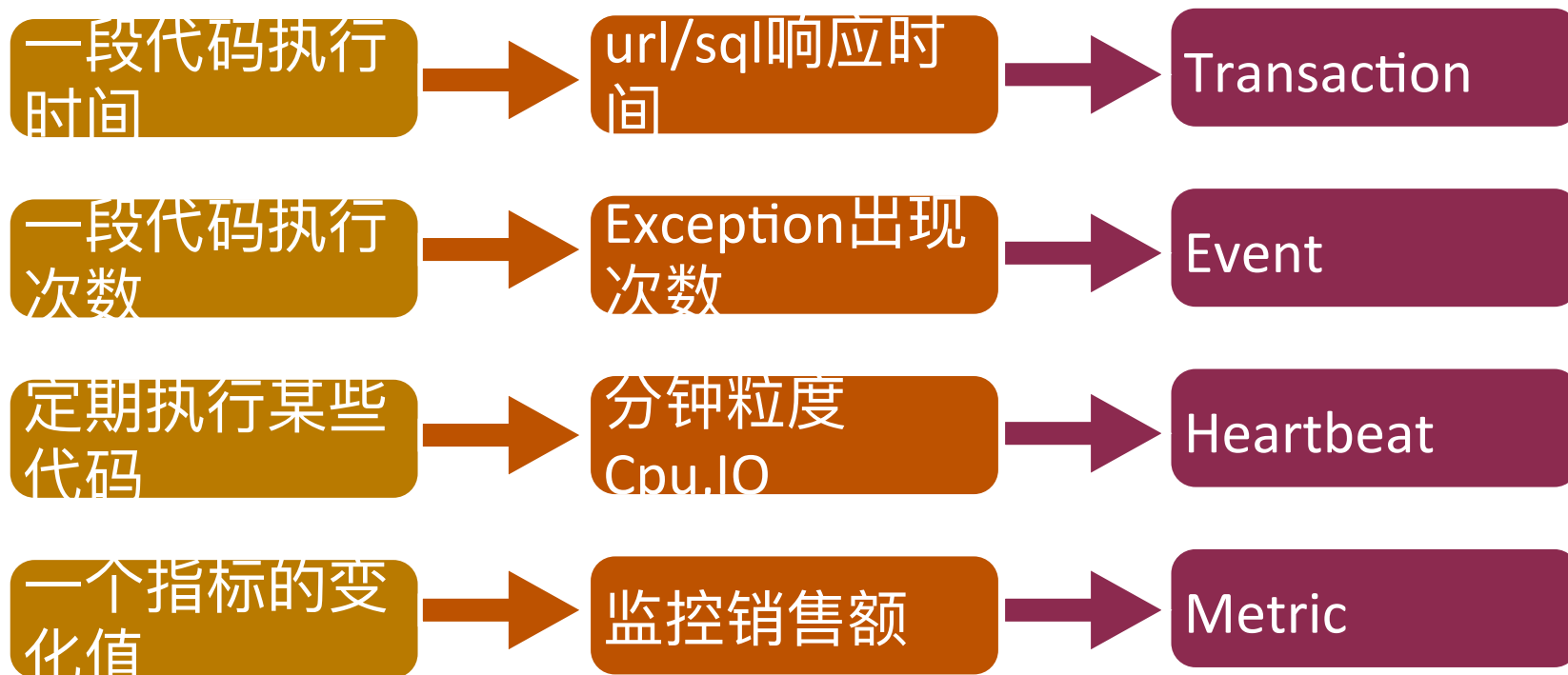
服务端重点

- 监控建模
- 报表建模
- CPU优化
- 数据存储

建模

- 监控领域模型
- 数据报表模型

监控建模



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 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>
```

```
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.get Durations().values()) {  
            visitDuration(duration);  
        }  
  
        for (AllDuration allDuration : name.getAll Durations().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性能优化

- 减少重复复杂的计算
- 减少重复对象的创建
- jprofile

数据存储

- 顺序写、随机读
- 批量压缩（大规模数据）

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）

领域建模

- 监控领域监控
- 后端报表模型建模

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

```
1 |
2 | @RunWith(JUnit4.class)
3 | public class TestServer extends JettyServer {
4 |     public static void main(String[] args) throws Exception {
5 |         TestServer server = new TestServer();
6 |         System.setProperty("devMode", "true");
7 |         server.startServer();
8 |         server.startWebApp();
9 |         server.stopServer();
10 |    }
11 |
12 |    @Before
13 |    public void before() throws Exception {
14 |        System.setProperty("devMode", "true");
15 |        super.startServer();
16 |    }
17 |
18 |    @Override
19 |    protected String getContextPath() {
20 |        return "/cat";
21 |    }
22 |
23 |    @Override
24 |    protected int getServerPort() {
25 |        return 2281;
26 |    }
27 |
28 |    @Override
29 |    protected void postConfigure(WebAppContext context) {
30 |        context.addFilter(GzipFilter.class, "/*", Handler.ALL);
31 |    }
32 |
33 |    @Test
34 |    public void startWebApp() throws Exception {
35 |        // open the page in the default browser
36 |        display("/cat/r");
37 |        waitForAnyKey();
38 |    }
39 | }
```

站在巨人肩膀上

- Ebay的CAL系统
- 商业产品NewRelic
- 腾讯的移动端监控
-

最难的事情

- 项目上线推动
 - 如何推动整个项目（2-3人）
 - 部门之间沟通问题
 - 跨地域支持问题
 - 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99.9%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	王小明	25,564,634	25,434,559	99.491%
交易前台-支付	李小红	14,381	13,960	97.073%
交易前台-玩乐	张三	2,073,438	2,071,467	99.905%
交易前台-电影	王小明	10,567,622	10,523,487	99.582%
交易前台-运营	李小红	124	124	100.0%
交易前台-汇总	王小明	38,220,199	38,043,597	99.538%
基础垂直-丽人	王小明	2,476,769	2,466,669	99.592%
基础垂直-结婚亲子家装	王小明	4,347,402	4,342,036	99.877%
基础垂直-酒店	王小明	3,910,053	3,900,772	99.763%
基础垂直-汇总	王小明	10,734,224	10,709,477	99.769%
大众微生活-微生活	王小明	766,481	765,276	99.843%
大众微生活-汇总	王小明	766,481	765,276	99.843%



CAT历程

- 2011-11月份 启动
- 2012-3月份 MVP模型
- 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
- 2017-4月份 5000+应用 34000+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

- . 2017年3月
- . Apache License, 国内上百家公司在使用和评估

. <http://github.com>

Unwatch ▼

560

★ Unstar

2,821

Fork

1,444

- . 每天200TB数据, 5000应用, 32000+机器



Thank you !