

CAT

(Central Application Tracking)

Geekbang>

极客邦科技

全球领先的技术人学习和交流平台

InfoQ^{ueue}

专注中高端技术
人员的社区媒体

EGO EXTRA GEEKS' ORGANIZATION
NETWORKS

高端技术人员
学习型社交网络

StuQ^{ueue}

实践驱动的IT职业
学习和服务平台

扫我，码上开启新世界



Geekbang>

InfoQ! | EGO NETWORKS | StuQ!



促进软件开发领域知识与创新的传播



实践第一 案例为主

时间：2015年12月18-19日 / 地点：北京·国际会议中心

欢迎您参加ArchSummit北京2015，技术因你而不同



ArchSummit北京二维码



【北京站】

2016年04月21日-23日



关注InfoQ官方信息
及时获取QCon演讲视频信息

自我介绍

- 尤勇
- 大众点评 基础架构组

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、核心网络层监控、系统层监控等。
- CAT是一个给提供实时监控告警，应用性能分析诊断的工具。

实时系统

- ① 1、客户端日志不落地
- ① 2、服务端流处理
- ① 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
- 消息体

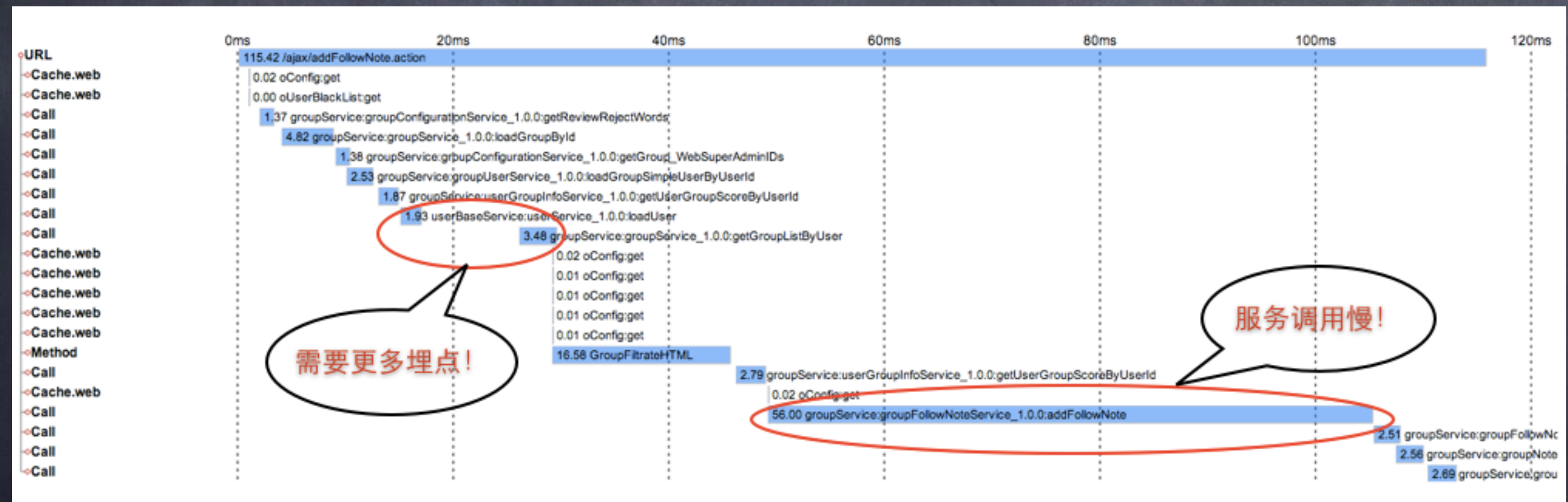
t: Transaction Start
E: Event
T: Transaction End
A: Atomic Transaction

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 | RemotelP=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.i |
| A15:00:44.023 | Cache.web | oUserBlackList:get | 0.00ms finalKey=oUserBl |
| 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=l | |
| 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=l | |
| 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 |
| Hearbeat | JVM内部一些状态信息，Memory，Thread等 |
| Matrix | 一个请求调用链路统计 |
| Cross | SOA系统用关于RPC调用的报表 |
| Cache | 缓存使用分析统计 |
| Dependency | 系统之间实时调用数据信息，远程服务、数据库、缓存等 |
| ... | ... |

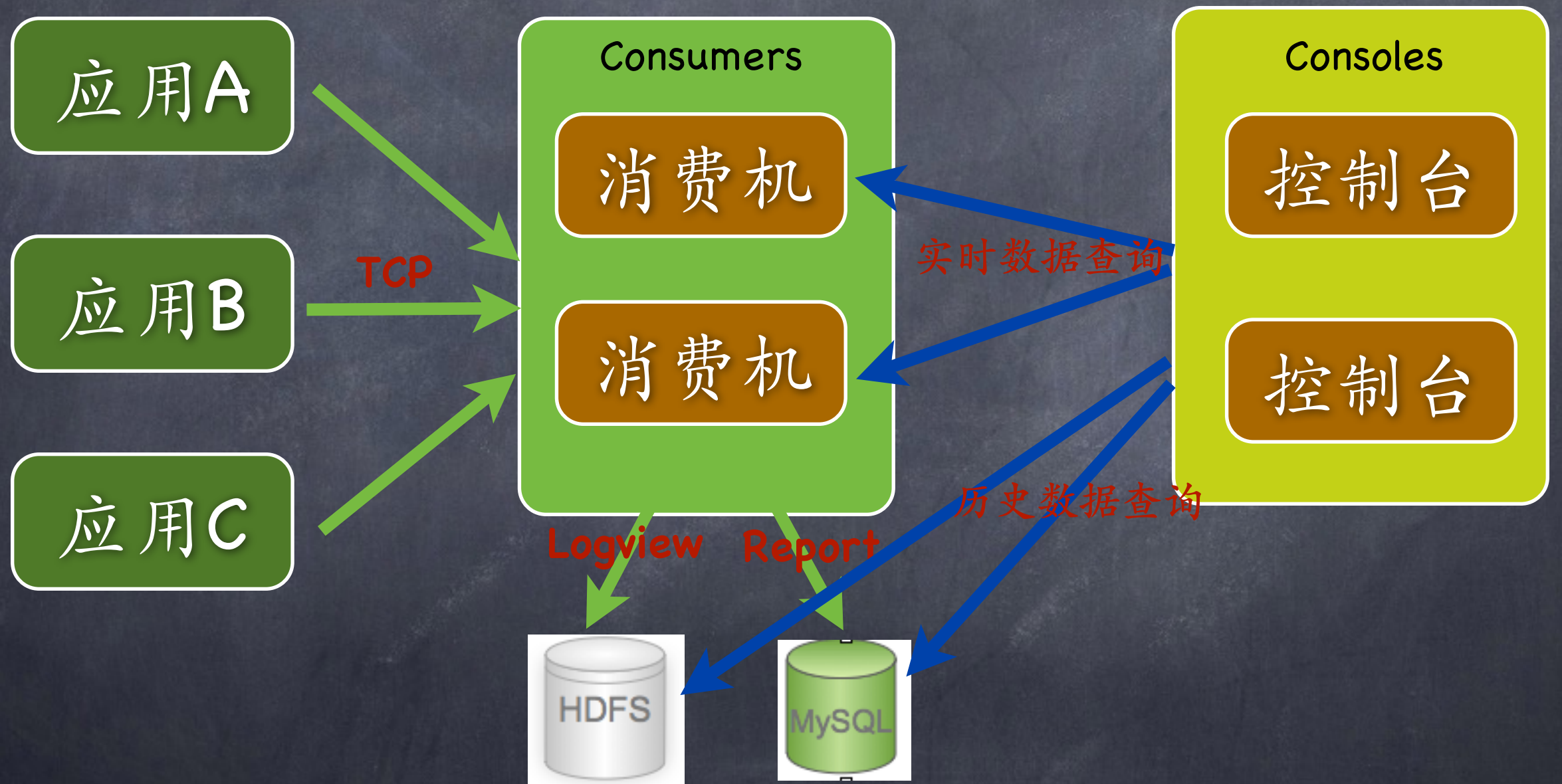
大纲

- CAT 历程
- CAT 设计
- 最佳实践

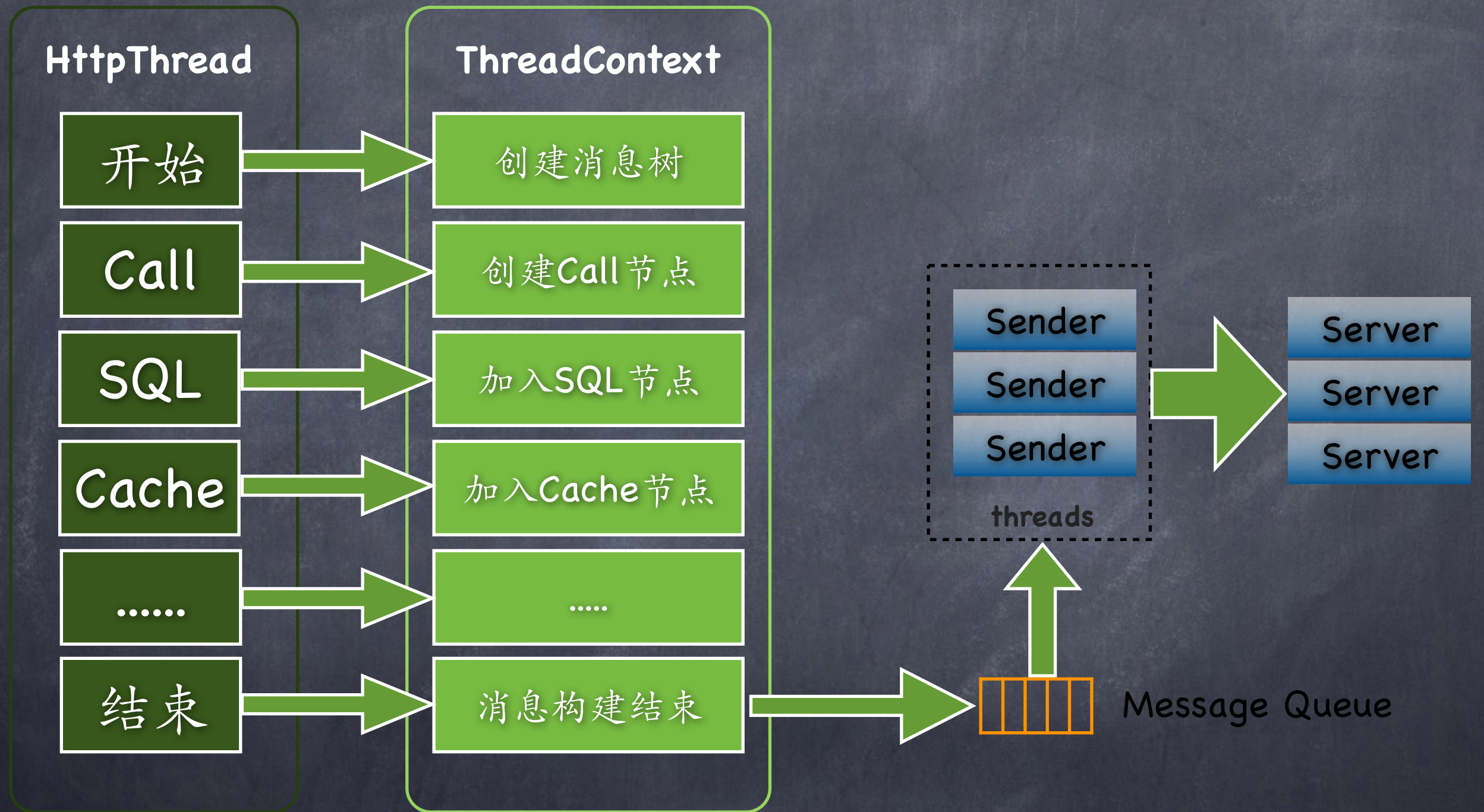
CAT设计

- ① 整体设计
- ② 客户端设计
- ③ 服务端设计

整体设计



客户端设计



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
- 基于netty实现消息传输

遇到问题-IO

- java message tree id的生成, java MappedByteBuffer需要做持久化
- 业务主线程的使用
- 在任何时候客户端都是需要考虑极端情况cpu或者io的开销

遇到问题-Memory

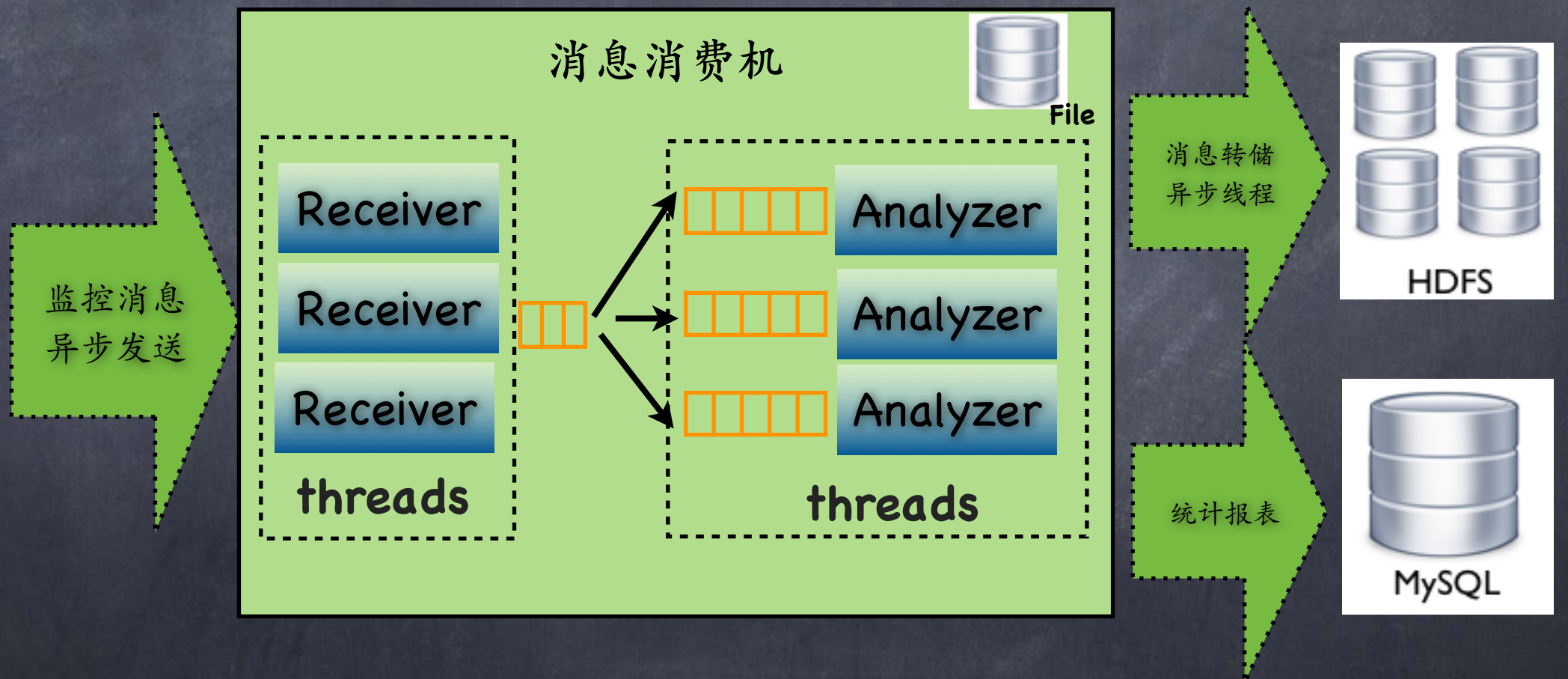
- ⑥ MessageTree的内存占用太大，极端情况下，一个messageTree里面上万个节点
- ⑥ 在任何时候客户端都是需要考虑极端情况内存的开销

服务端设计

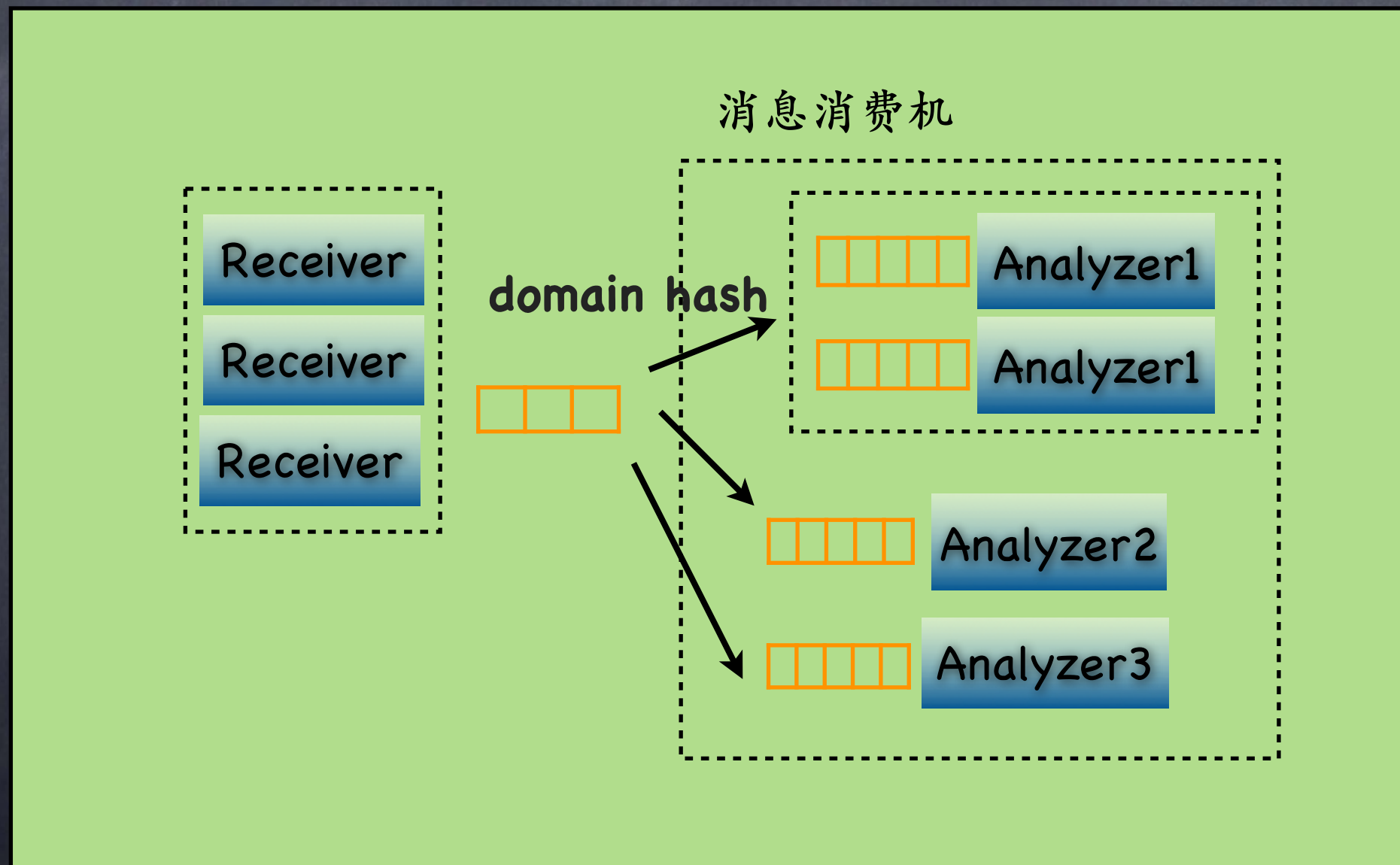
应用A

应用B

应用C



当Analyzer处理来不及



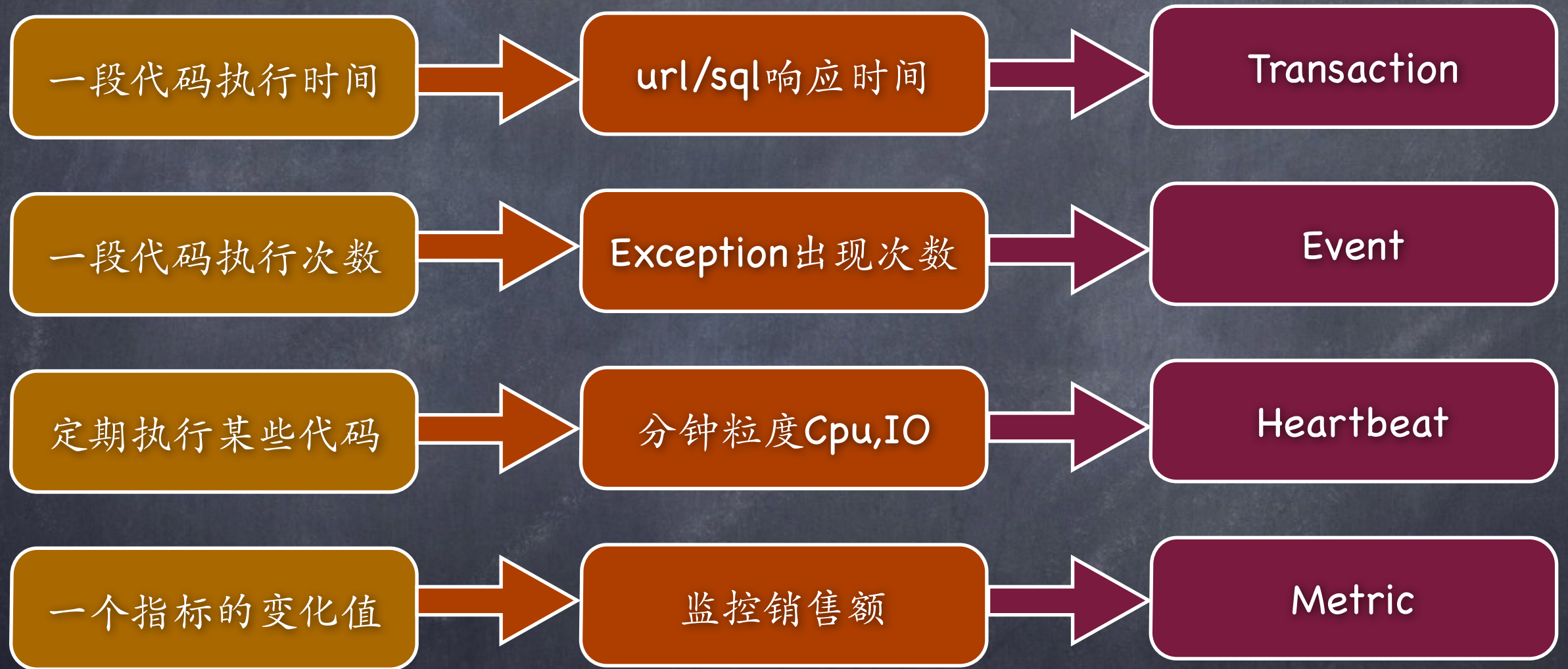
服务端重点

- ◉ 监控建模
- ◉ 报表建模
- ◉ CPU优化
- ◉ 负载均衡
- ◉ 数据存储
- ◉ 内存以及系统问题

建模

- 监控领域数据模型
- 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);  
            }  
        }  
    }  
}
```

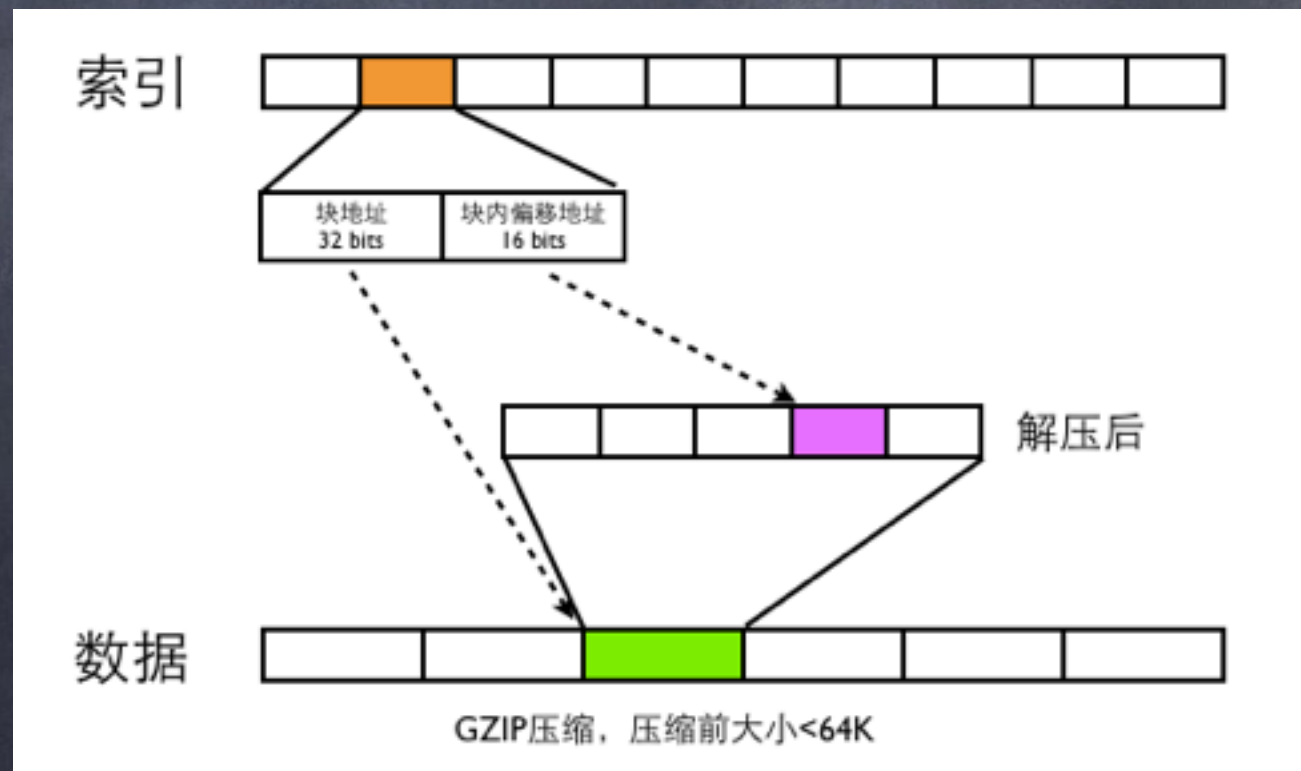
```
public long parse(String str) {  
    int len = str.length();  
    String date = str.substring(0, 10);  
    Long baseline = m_map.get(date);  
  
    if (baseline == null) {  
        try {  
            SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");  
  
            format.setTimeZone(TimeZone.getTimeZone("GMT+8"));  
            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/ShopService-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



内存困惑

- 1、 transaction report, event report 里面的 name 无限个节点
- 2、 swap off
- 3、 numactl --interleave=all
- 4、 jvm

系统问题

- ① 1、机器上非jvm线程的影响
- ① 2、`ethtool -K eth0 tso off`
- ① 有些“tg3网卡”因为硬件缺陷的原因，在启用TSO时，有时会要求报文必须存储在物理连续的内存中，而驱动程序如果发现报文不满足要求就会复制一份报文出来，如果这时内存不足，或者因为内存碎片问题，驱动可能分配不到内存。

大纲

- CAT介绍
- CAT设计
- 最佳实践

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

MVP版本

- Demo 1个月
- MVP 3个月
- 重点解决最急迫的一个问题

小白鼠客户

- 典型客户
- vip服务

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）

上线以及后续

- ⑥ 独立快速发布（项目初期）
- ⑥ 灰度发布（项目中后期）
- ⑥ 问题排查（mat）

不仅仅是code

- 不同角色如何使用系统
- 系统如何运维
- 系统如何推广

数据质量

- 数据质量
- sql框架、缓存框架、rpc框架、web框架
- 数据质量决定了监控质量

单机开发环境

```
@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();
    }
}
```

```
protected boolean isActive() {
    synchronized (this) {
        return m_active;
    }
}

protected boolean isLocalMode() {
    return m_serverConfigManager.isLocalMode();
}
```

• jetty server

• hdfs依赖

• mysql依赖

最难的事情

- ① 项目上线推动
 - ① 如何推动整个项目上线（2-3人）
 - ① 部门之间沟通问题
 - ① 后续的支持和培训

其他需求

- 其他的需求
- 系统开放生态

Open Source

- 代码依赖
- doc的完善
- code clean
- <http://github.com/dianping/cat>

项目合作

- <https://github.com/dianping/cat/issues/753>



QA

- Thanks