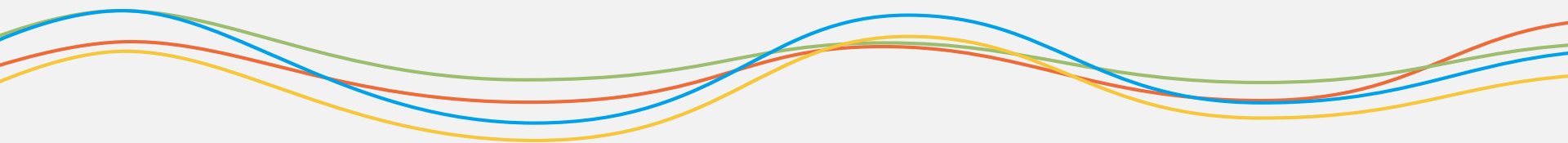


# “静态调用链路发现”在APM中的应用场景分析及实践探索

天弘基金 李鑫



## ◆ APM的发展与现状

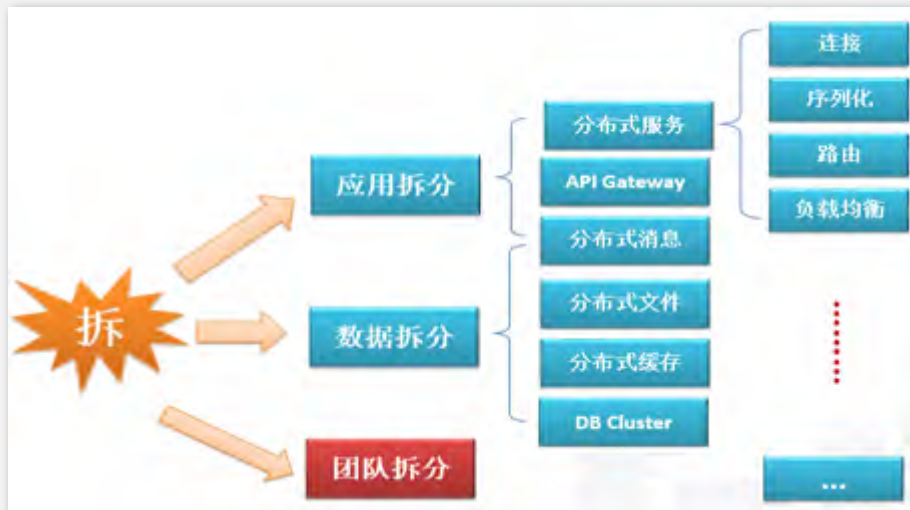
- APM发展历程
- 动态调用链路的模式及“优缺点”

## ◆ 静态调用链路

- 原理
- 实现技术及手段
- 与动态调用链路（及监控）的结合
- 应用场景分析

## ◆ 总结



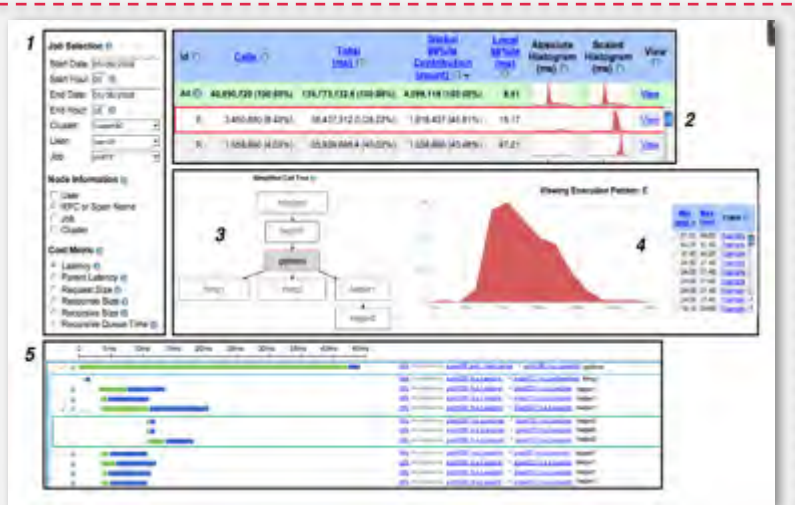
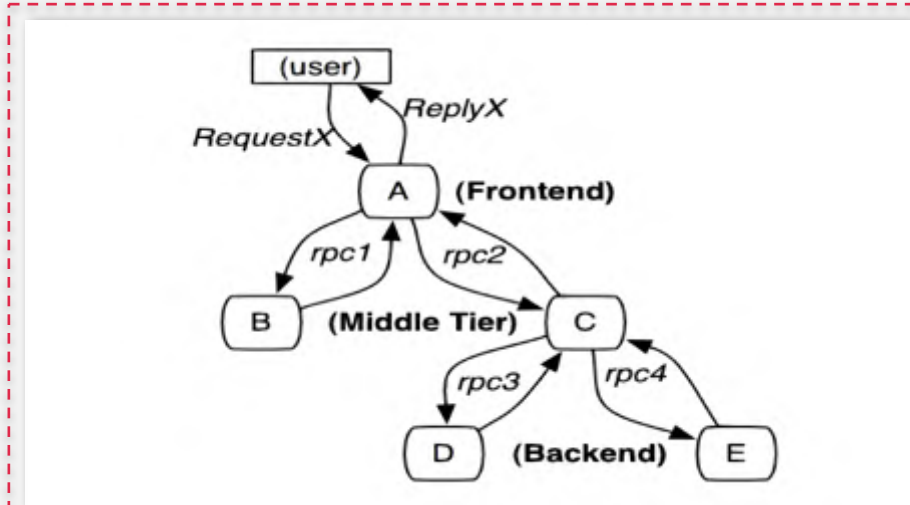


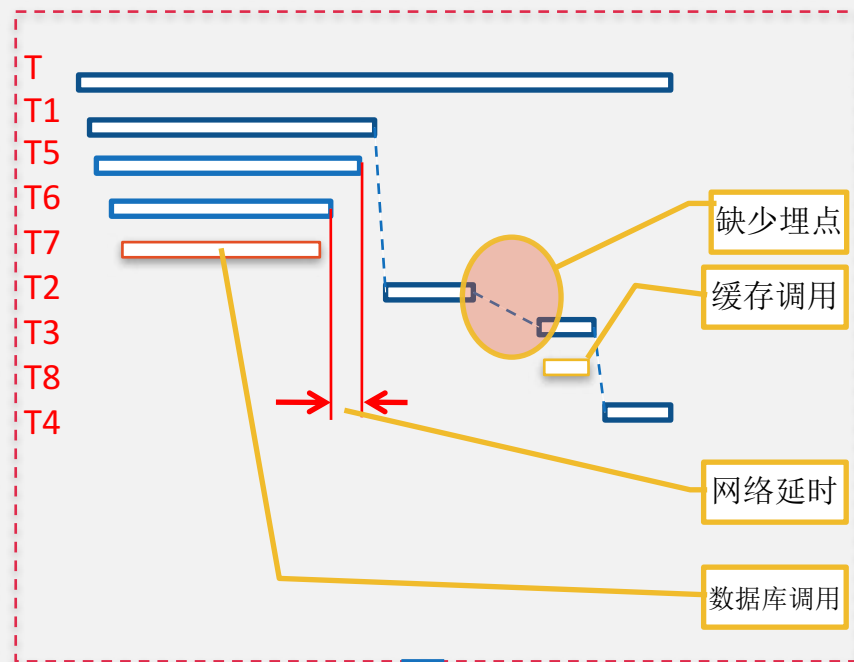
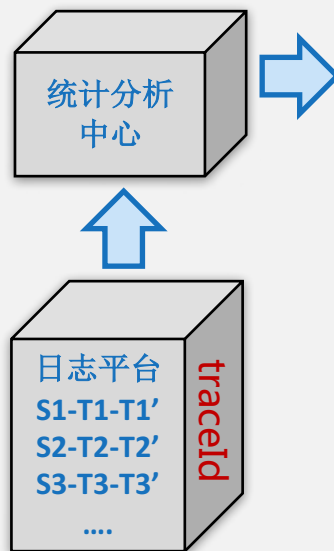
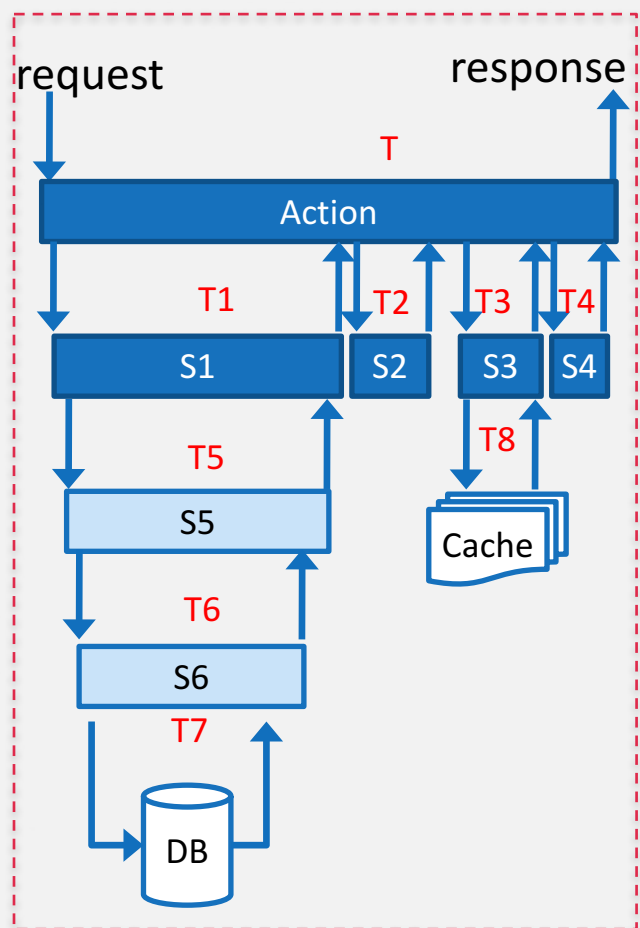
服务化拆分是系统发展的必然趋势

故障在哪?  
性能瓶颈在哪?  
调用关系?

A sad green character with a question mark above its head is shown, representing the challenges of debugging and performance analysis in a distributed system.

服务化后系统运维的新难题





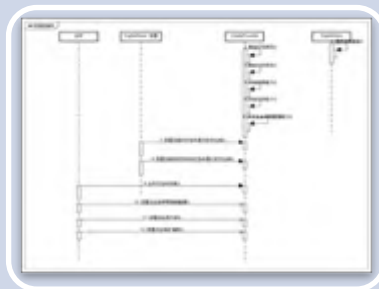
		调用链A	调用链B	调用链C	调用链D	Total
DB1	Read/m	28453	544	12821	9987	51805
	write/m	4421	22	0	2233	6676
DB2	Read/m	19611	500	12821	5521	38453
	write/m	43643	1022	25642	13276	83582



基于埋点、或基于线程堆栈技术，只能发现实际调用的业务逻辑。大量没有埋点或未被触发的调用逻辑则成了“失落的世界”，无法触碰。



**我们看到的，只是冰山的一角**



业务需求

产品设计

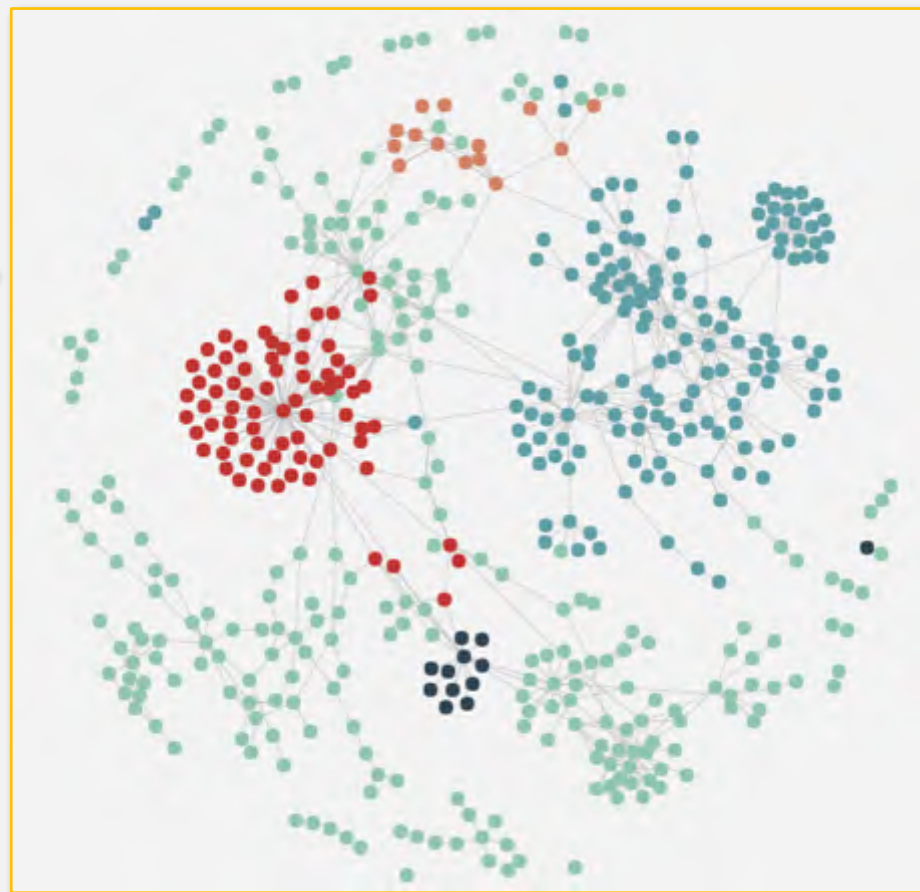
架构设计



软件开发是一项协作性的智力行为，所有的思考、意图、策略最终都体现在了代码上。因此，一个系统的代码就是一本“书”，读懂它，我们就能知道系统的“前世今生”。

问题是，我们如何“读懂”这本“书”。

# 一切源于源代码...



代码关系



链路关系



## Eclipse JDT

- 引用 (import)
- 变量
- 方法
- 继承、实现
- **注释**

利用JDT获得源代码的基本结构

## Method代码扫描

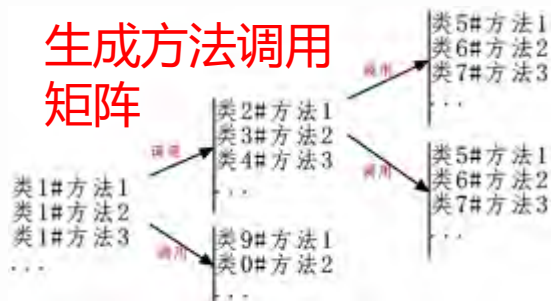
- 方法内变量
- 其它类方法调用
- 本类方法调用
- 方法内Block (IF-END IF、FOR-END FOR、WHILE...)

通过逐行解析获取代码块对外部方法的调用详情

## 调用关系 (层级)

- 主动调用关系
- 被动调用关系
- **过滤**

生成方法调用矩阵



# “静态调用链路”的基本形态

## 正向调用关系：

```

com.thfund.client.communitycommonlib.CommunityManager#requestCookie
-->com.thfund.client.thfundlib.rpcService.userRpcService.UserInfoManagerRpcService#requestCookie
-->com.thfund.app.ifinance.UserInfoManager#readDZCookie
   =com.thfund.app.ifinance.biz.UserInfoManagerImpl#readDZCookie
-->com.thfund.app.ifinance.dao.UserMapper#queryUserByUserId
   -->com.yl.authcode.util.AuthCode#encode
   -->android.content.SharedPreferences#edit
   -->android.content.SharedPreferences#edit
   -->android.content.SharedPreferences#edit
-->com.thfund.client.communitycommonlib.CommunityManager#checkPostAllowlance
->com.thfund.client.communitycommonlib.CommunityManager#isSetCookieInfo
   -->com.thfund.client.communitycommonlib.CommunityManager#isSetCookie[IF]
   -->com.thfund.client.communitycommonlib.CommunityManager#requestCookie[deadLoop][END IF]
   -->com.android.volley.toolbox.Volley#newRequestQueue
  
```

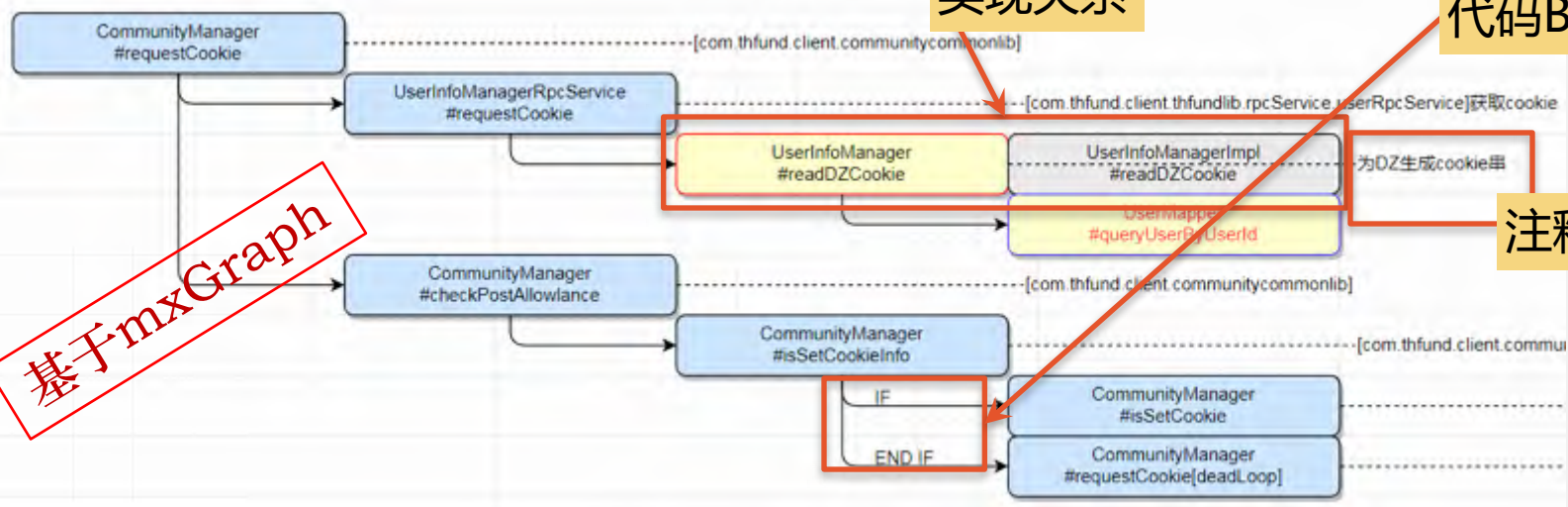
基于文本

实现关系

代码Block

注释自动提取

基于mxGraph



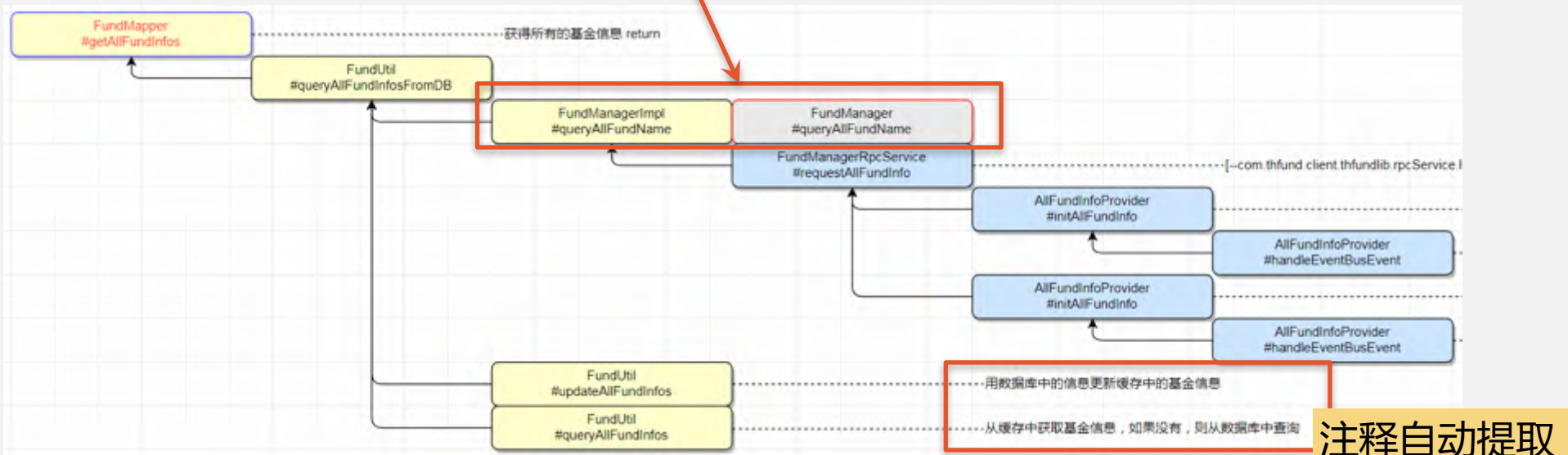
# “静态调用链路”的基本形态

## 逆向调用关系：

```

com.thfund.app.ifinance.dao.FundMapper#getAllFundInfos
<--com.thfund.app.ifinance.utils.FundUtil#queryAllFundInfosFromDB
<--com.thfund.app.ifinance.biz.FundManagerImpl#queryAllFundName
    =com.thfund.app.ifinance.FundManager#queryAllFundName
<--com.thfund.client.thfundlib.rpcService.launcherRpcService.FundManagerRpcService#requestAllFundInfo
    <--com.thfund.client.thfundlib.dataManager.AllFundInfoProvider#initAllFundInfo
        <--com.thfund.client.thfundlib.dataManager.AllFundInfoProvider#handleEventBusEvent
    <--com.thfund.client.thfundlib.dataManager.AllFundInfoProvider#initAllFundInfo
        <--com.thfund.client.thfundlib.dataManager.AllFundInfoProvider#handleEventBusEvent
<--com.thfund.app.ifinance.utils.FundUtil#updateAllFundInfos
<--com.thfund.app.ifinance.utils.FundUtil#queryAllFundInfos
    
```

### 实现关系



## 多种日志收集方式并存的采集端架构

### ◆ 日志文件拉取

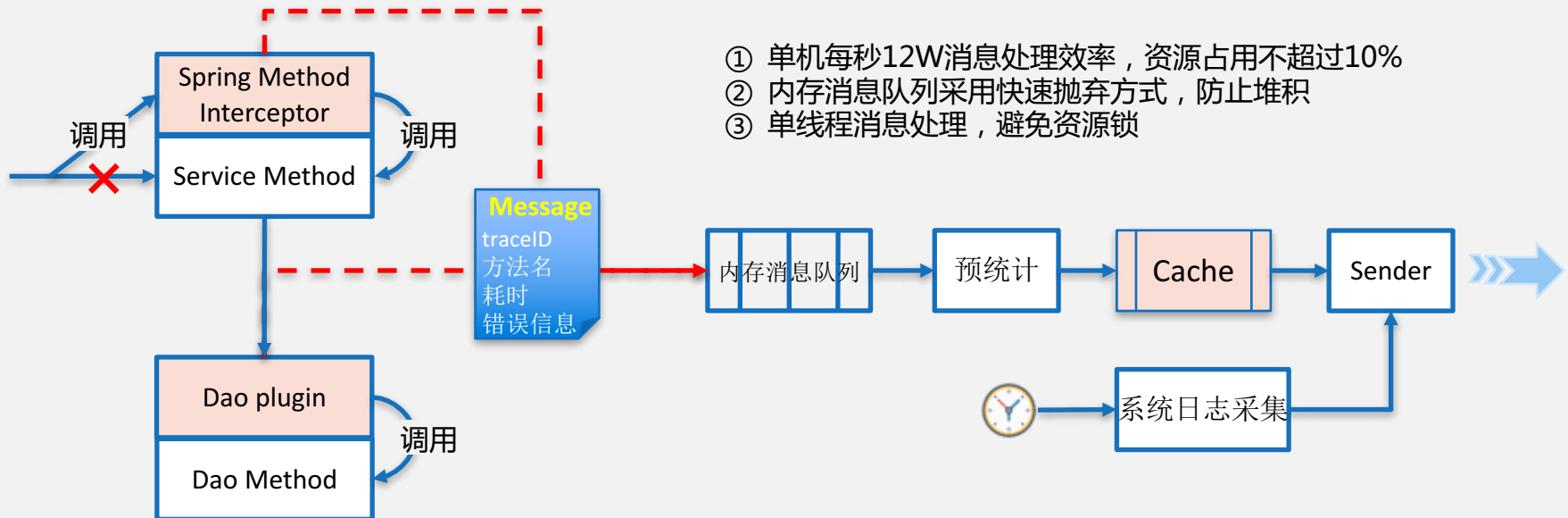
- 通过log4j的MDC添加traceID
- Logstash增量拉取日志数据

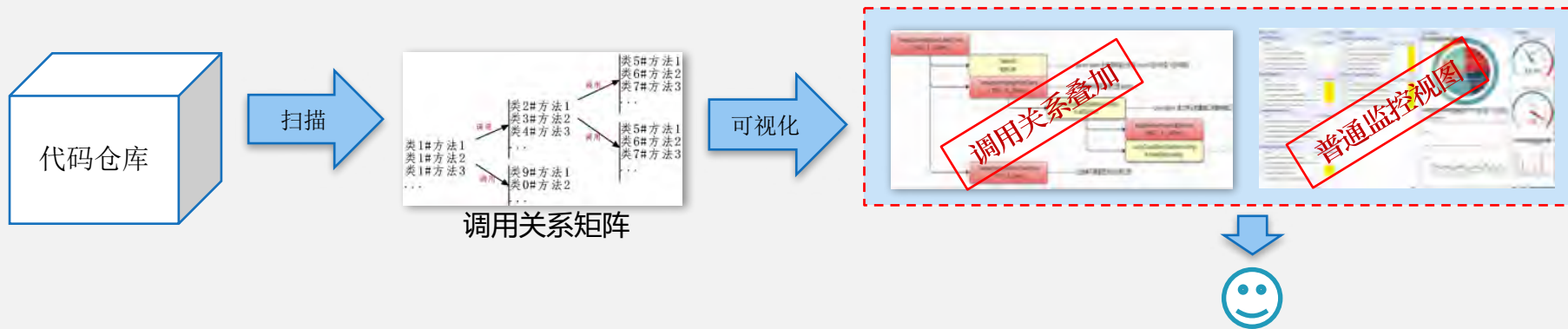
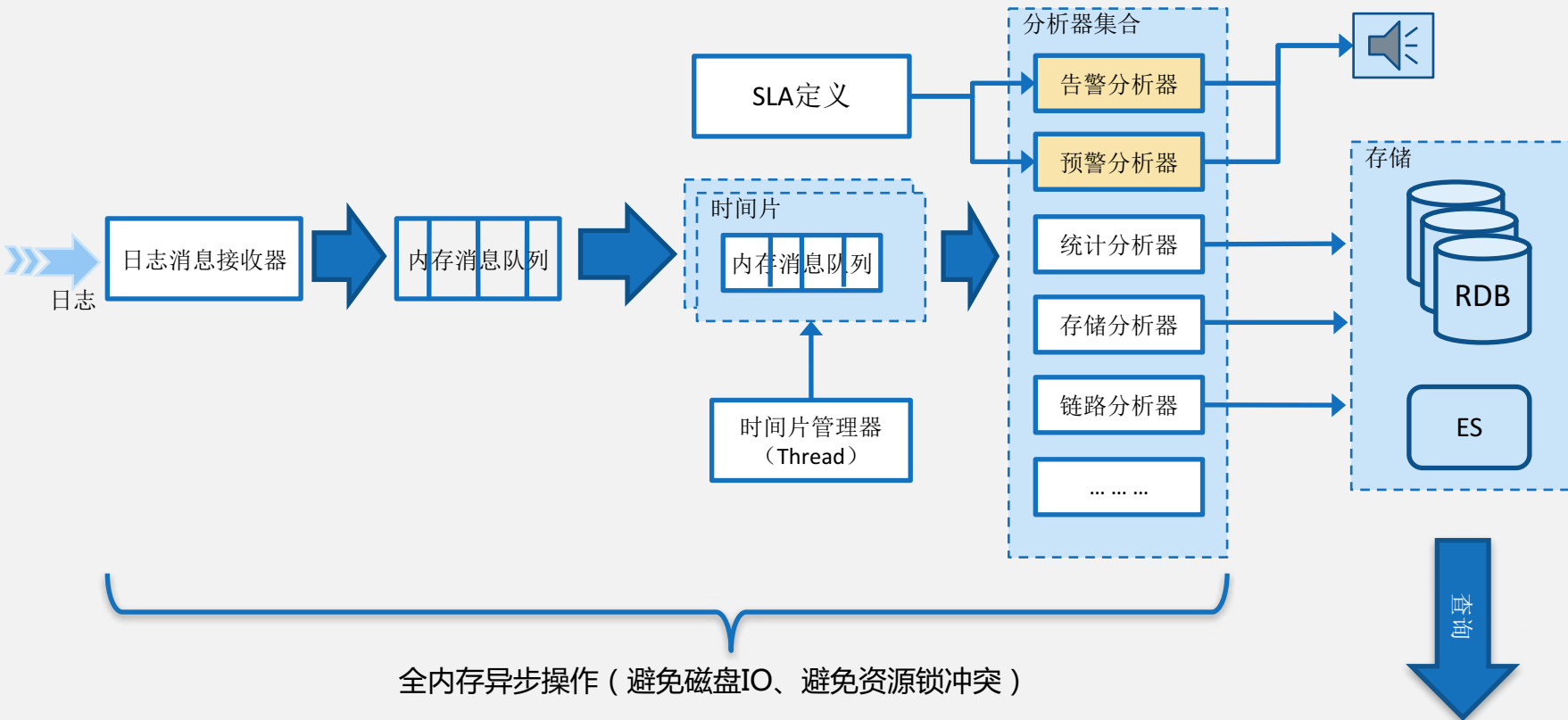
### ◆ 批量消息推送

- 纯内存操作，无磁盘IO
- 预统计
- 批量发送

### ◆ 定期采集系统性能日志

- CPU、磁盘、内存、SWAP、系统负载
- JVM堆栈、线程、GC



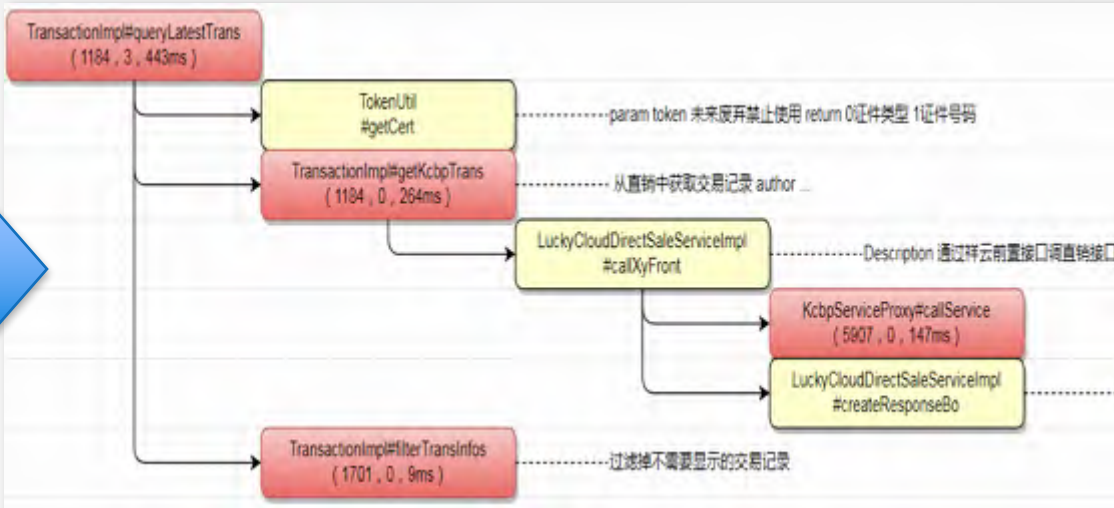


## 与普通监控数据结合（可视化）

## • 离散监控数据关系化

Top 20 的慢方法

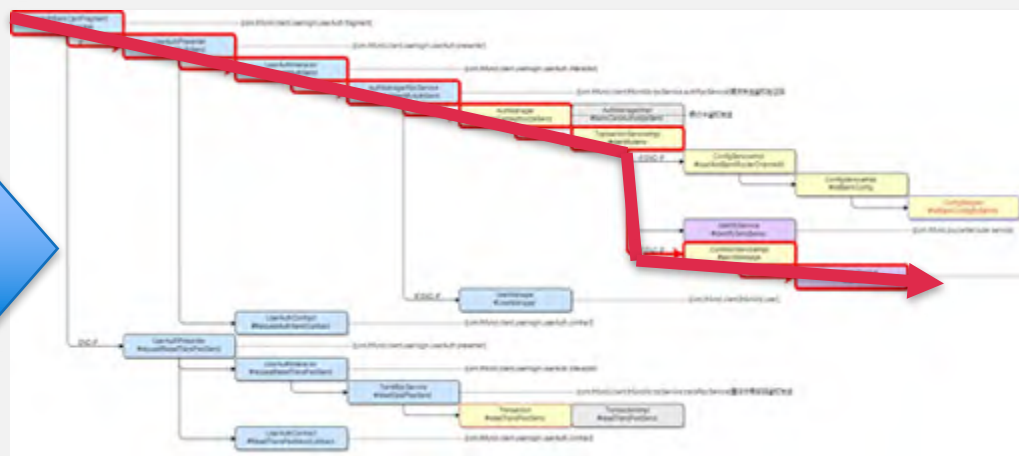
应用名称	方法名称	调用次数	总耗时 (ms)	平均耗时 (ms)
com.ibm.app	com.ibm.app	1000	10000	10
com.ibm.app	com.ibm.app	500	5000	10
com.ibm.app	com.ibm.app	300	3000	10
com.ibm.app	com.ibm.app	200	2000	10
com.ibm.app	com.ibm.app	150	1500	10
com.ibm.app	com.ibm.app	100	1000	10
com.ibm.app	com.ibm.app	80	800	10
com.ibm.app	com.ibm.app	60	600	10
com.ibm.app	com.ibm.app	50	500	10
com.ibm.app	com.ibm.app	40	400	10
com.ibm.app	com.ibm.app	30	300	10
com.ibm.app	com.ibm.app	20	200	10
com.ibm.app	com.ibm.app	15	150	10
com.ibm.app	com.ibm.app	10	100	10
com.ibm.app	com.ibm.app	8	80	10
com.ibm.app	com.ibm.app	6	60	10
com.ibm.app	com.ibm.app	5	50	10
com.ibm.app	com.ibm.app	4	40	10
com.ibm.app	com.ibm.app	3	30	10
com.ibm.app	com.ibm.app	2	20	10
com.ibm.app	com.ibm.app	1	10	10



## 与调用链监控数据结合（完善）

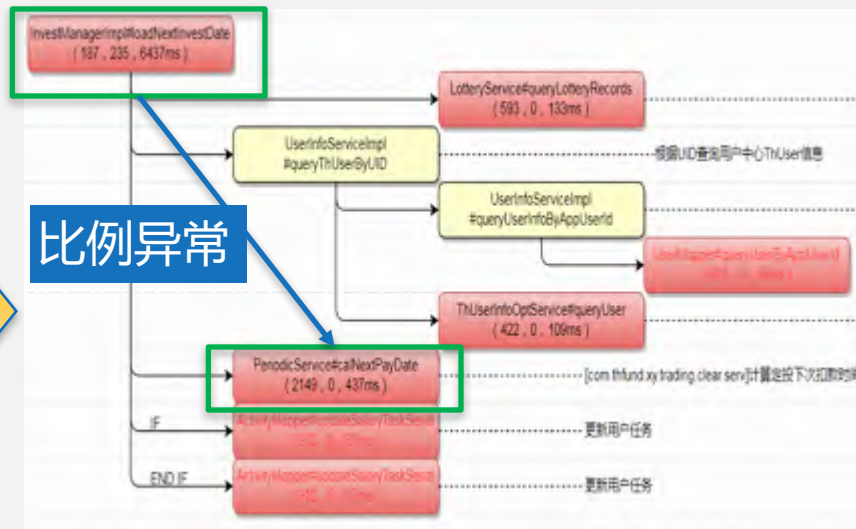
## • 调用关系完整化

应用名	类型	状态	大小	服务/方法	时间轴
tt_buy	TRACE	OK	-	http://www.ibm.com	23ms
tradesplitoms	HSP	OK	670B	tt.TcTradeService@getOrderSvc(58@user@-)	0ms
tradesplitoms	HSP	OK	6.81KB	trade.IOpatingOrderService@createOrderPor(0@0@-L)	40ms
itemcenter	HSP	OK	2.11KB	ItemLItemQueryService@queryItemAndSKU(10@PYToText-L)	5ms
itemcenter	HSP	OK	2.11KB	ItemSkuService@getSku-I	3ms
(notify)	NOTIFY	OK	-	Notify@send	6ms
timecenter	NOTIFY	OK	-	Notify@recv	10ms
tradebgs	NOTIFY	OK	-	Notify@recv	10ms
tradeaccessonip	NOTIFY	OK	-	Notify@recv	12ms
tradebgs	NOTIFY	OK	-	Notify@recv	12ms
tradebgs	NOTIFY	OK	-	Notify@recv	13ms
tradeesuid	NOTIFY	OK	-	Notify@recv	13ms
tie	NOTIFY	OK	-	Notify@recv	13ms
stoa	NOTIFY	OK	-	Notify@recv	15ms
trade_sub2_notify	NOTIFY	OK	2.11KB	Notify@recv-SystemMessage-TRADE-100-trade-created-done-by-server-group-2	6ms
(db tradesub2)	TDDL	OK	-	TDDL_UPDATE@tradesub2	1ms
trade_sub_notify	NOTIFY	OK	2.11KB	Notify@recv-SystemMessage-TRADE-100-trade-created-done-by-server-group-2	10ms
(db notify_trade)	TDDL	OK	-	TDDL_UPDATE@notify_trade	1ms
misccenter	HSP	OK	5.11KB	misccenter.EcardOrderService@desertEcardOrder-E	3ms
(db ecard01)	TDDL	OK	-	TDDL_UPDATE@ecard01	1ms
tradesplitoms	HSP	OK	1.11KB	trade.IOrderingOrderService@enableOrder-L	16ms

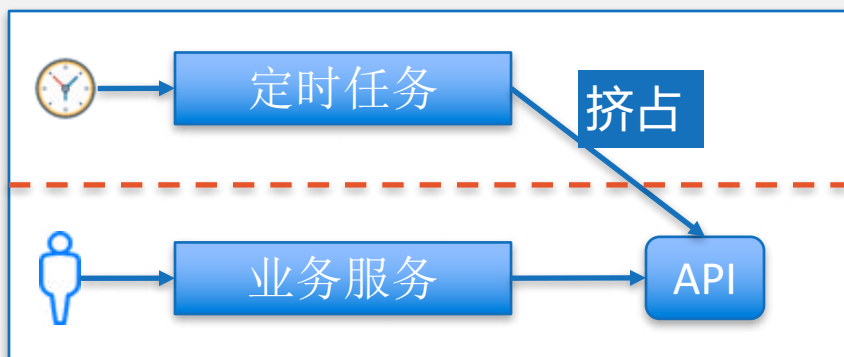


URL	请求数	响应时间	成功率
com.thfund.app.ifinance.biz.InvestManagerImpl.loadNextInvestDate	230	187	0.5600
com.thfund.uc.thuser.service.ThUserInfoOptService.queryUserBase	24	5468	0.0151
com.thfund.app.ifinance.biz.HomePageManagerImpl.homepageInfoSingleV270	51	6131	0.0002
com.thfund.app.ifinance.biz.TransactionImpl.queryTransButtonStatus	30	1822	0.0178
com.thfund.app.ifinance.biz.MsgCenterManagerImpl.queryMyMessageAlert	32	8105	0.0039
com.thfund.app.ifinance.biz.LotteryCardManagerImpl.queryUserActivityLotteryCard	28	198	0.1250
com.thfund.xy.trading.bail.service.CombineQueryService.combineOrderQuery	29	318288	0.0001
com.thfund.app.ifinance.biz.MobilePagesImpl.getMyPageDataSinceV270	20	8866	0.0334
com.thfund.app.ifinance.biz.UserInfoManagerImpl.addBankCardRetrieve	19	63	0.2025
com.thfund.app.ifinance.biz.TransactionImpl.queryTrans	15	12257	0.0312
com.thfund.app.ifinance.biz.AuthManagerImpl.addYinZiSign	13	4	0.7805
com.thfund.uc.thuser.service.ThUserInfoOptService.updateLoginPwd	13	66	0.1805
com.thfund.app.ifinance.biz.TransactionImpl.applyByEncryptTransPwd	11	371	0.0388
com.thfund.dc.service.UserQueryService.getResult	9	0	1.0000
com.thfund.app.ifinance.biz.CombineProdManagerImpl.holdHomepage	6	1297	0.0001
com.thfund.app.ifinance.biz.UserInfoManagerImpl.queryBankCards	7	1412	0.0000
com.thfund.app.ifinance.biz.InvestManagerImpl.saveOrUpdateInvestPlan	6	264	0.6500

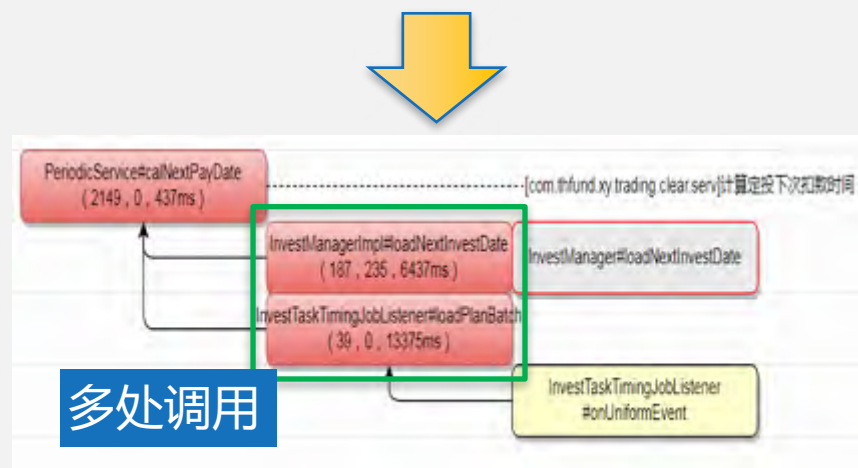
## 1.故障现象



## 2.异常分析

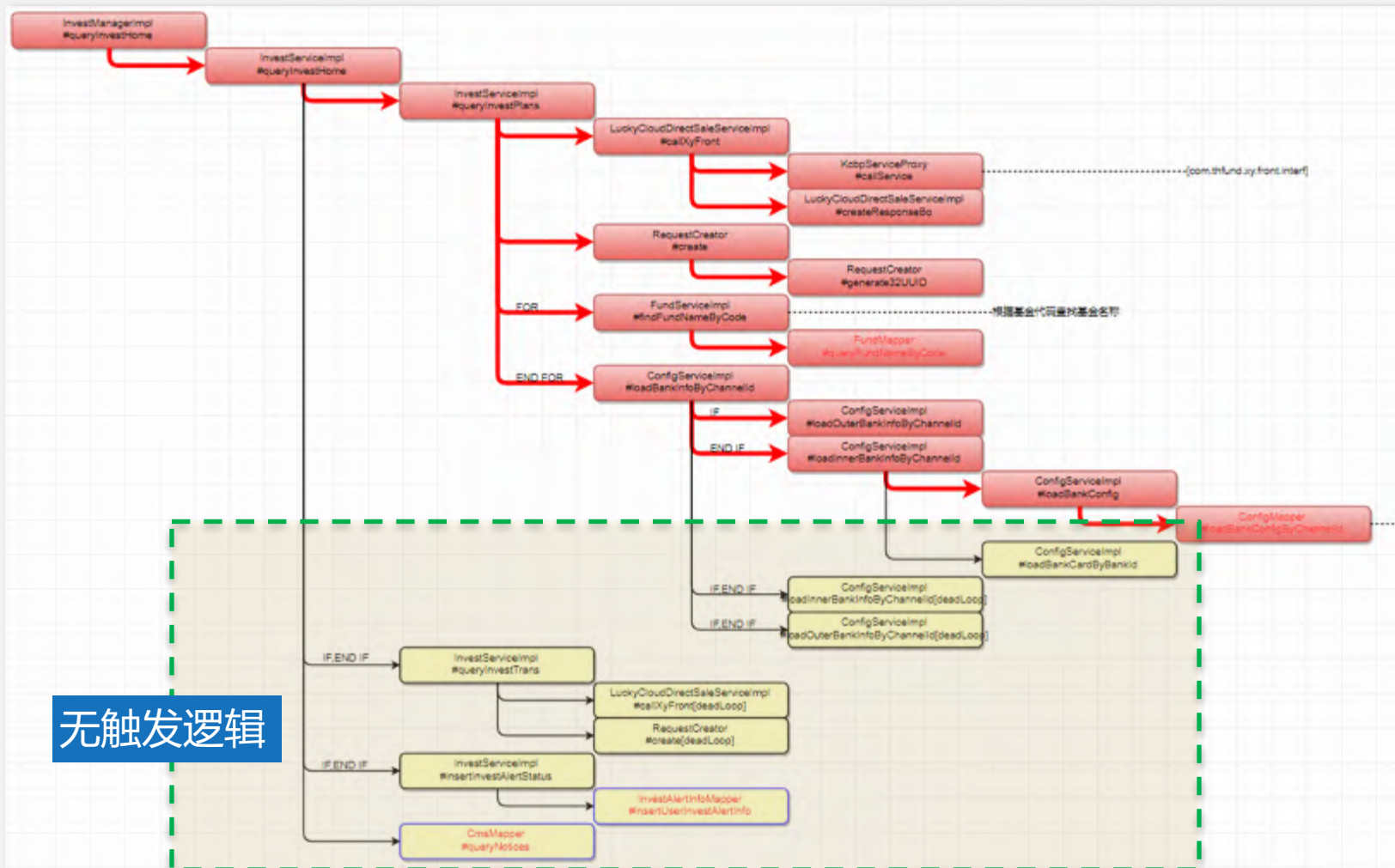


## 4.原因定位



## 3.逆向排查

1. 多余的业务逻辑链路
2. 异常处理链路
3. 旧版本兼容





# 静态调用链路的额外应用场景（工程效能）

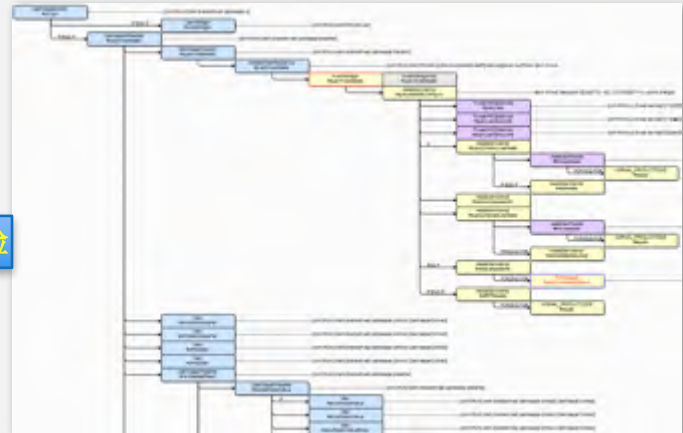
设计稿

设计稿校验



设计稿

校验



实际链路

文档

接口文档  
调用关系文档  
...

1. com.thfund.app. [redacted] web.OrgManageAc

描述: 配置管理  
method: [redacted]

1.1. queryEmployee

1.1.1. 方法描述: [redacted]

1.1.2. 访问地址: [redacted]

1.1.3. 方法说明: 通过对多种条件组合获取员工信息

四、入参:

字段	字段名称	类型	长度	非空	备注
condition	描述: 查询条件	com.thfund.infobase.EmployeeInfoView			
employeeId	描述: 员工ID	java.lang.Long			
employeeName	描述: 员工姓名	java.lang.String			
employeeStatus	描述: 员工状态	java.lang.String			
multi	描述: 手机号	java.lang.String			
orgId	描述: 分支机构ID	java.lang.Integer			
orgName	描述: 机构名称	java.util.Date			
updateTime	描述: 更新时间	java.util.Date			
updateTimeStart	描述: 最近更新时间	java.util.Date			
updateTimeEnd	描述: 最近更新时间	java.util.Date			
updateTimeStart	描述: 最近更新时间	java.util.Date			
updateTimeEnd	描述: 最近更新时间	java.util.Date			
updateTimeStart	描述: 最近更新时间	java.util.Date			
updateTimeEnd	描述: 最近更新时间	java.util.Date			

五、出参:

数据标识	名称	类型	长度	非空	备注
list	员工列表	员工列表			

1.2. querySalaryManager

1.2.1. 方法描述: [redacted]

1.2.2. 访问地址: [redacted]

接口文档



调用关系文档

1. 回顾了APM发展的3个阶段；
2. 分析了动态调用链路的实现模式及“优缺点”；
3. 提出了扫描代码库来获取方法级别的调用链路的思路；
4. 介绍了基于Eclipse JDT的代码结构解析及调用关系分析的技术及手段；
5. 介绍了静态调用链路的两种基本模式（正向调用及逆向调用）；
6. 应用场景介绍（监控、故障定界定位、冗余分析、文档...）。

# Q/A

## 联系方式:

李鑫

天弘基金(余额宝)·移动平台·首席架构师

E-mail:lixin.storm@foxmail.com

QQ:25893288


## 欢迎交流:

专注于大规模分布式应用及治理、中间件云化及服务化(PaaS)、APM监控、基础开发平台、移动应用平台、企业架构等技术领域。



个人公众号

**天弘基金**  
TIANHONG  
ASSET MANAGEMENT  
■ 稳健理财 值得信赖

 天弘爱理财

**THANK YOU**

