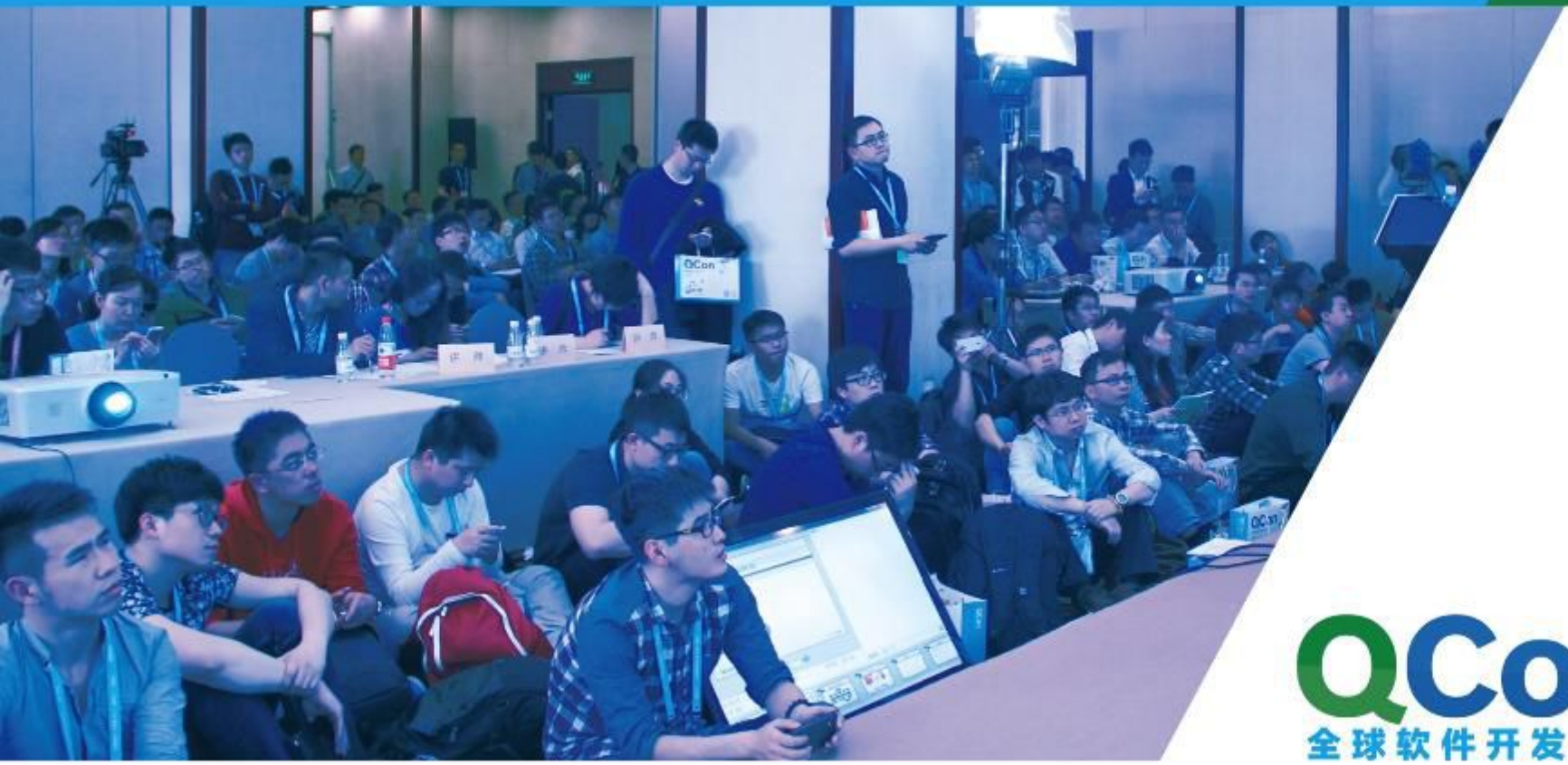
QCon全球软件开发大会

International Software Development Conference



QCon
全球软件开发大会

应用性能监测
Java Instrumentation技术实践

听云・廖雄杰

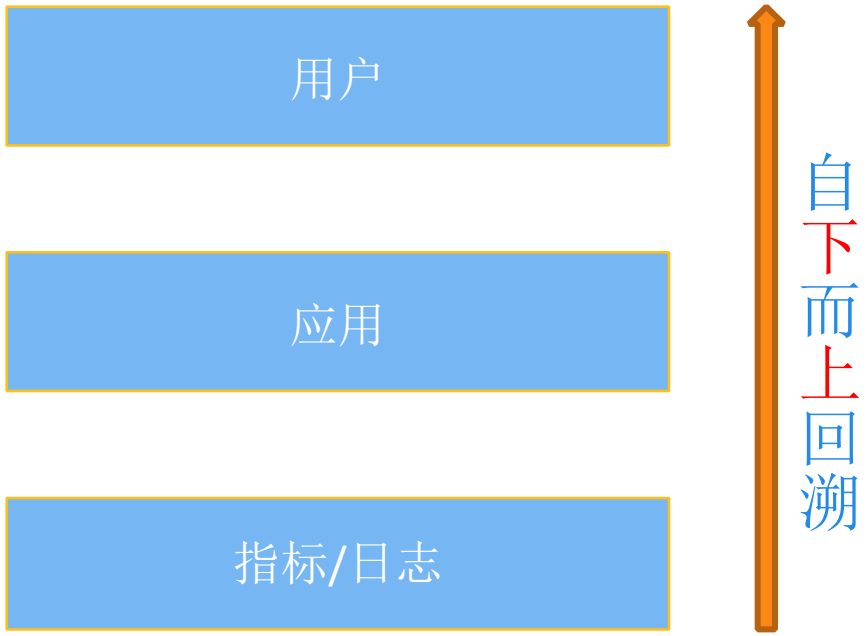
应用系统中的常见组件

- RDB
- NoSQL (Redis/Memcached/MongoDB)
- API/RPC
- 云服务
- MQ

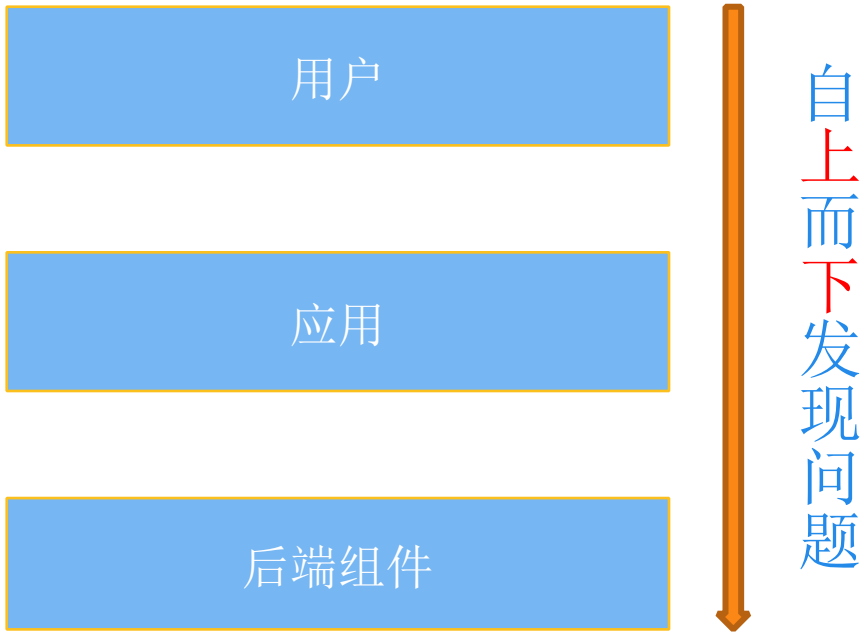
应用指标

- 系统指标
  - CPU
  - 内存
  - IO
  - 网络
- 应用组件指标
  - slow sql
  - 缓存命中率
  - qps
  - 应用日志

传统监控



APM：下一代监控技术



APM：下一代监控技术

Application as Monitoring

APM是什么？

Application Performance Management

深入应用代码的性能监控

1. 获取方法开始时间

```
public void xxoo() {
        long startTime = System.currentTimeMillis();

        try {
                doXX();
                doOO();

                long endTime = System.currentTimeMillis();
                long callTime = endTime - startTime;

                APM.reportMetric("xxoo", callTime);
        } catch(Exception ex) {
                APM.reportError("xxoo",
                                ex.getMessage(),
                                ex.getStacktrace());

                throw ex;
        }
}
```

2. 获取方法完成时间，并计算执行时间

3. 上报指标名及性能

4. 上报异常

自动化APM监控

- -javaagent:apm-agent.jar
- Java classloader
- premain()/Instrumentation/bytecode transforming

自动化APM监控

```
public class Xxoo {
        public static void main(String[] args) {
                Xxoo xo = new Xxoo();
                xo.xxoo();
        }

        public void xxoo() {
                doXX();
                doOO();
        }

        public void doXX() {
                System.out.println("xx!");
        }

        public void doOO() {
                System.out.println("oo!");
        }
}
```

自动化APM监控

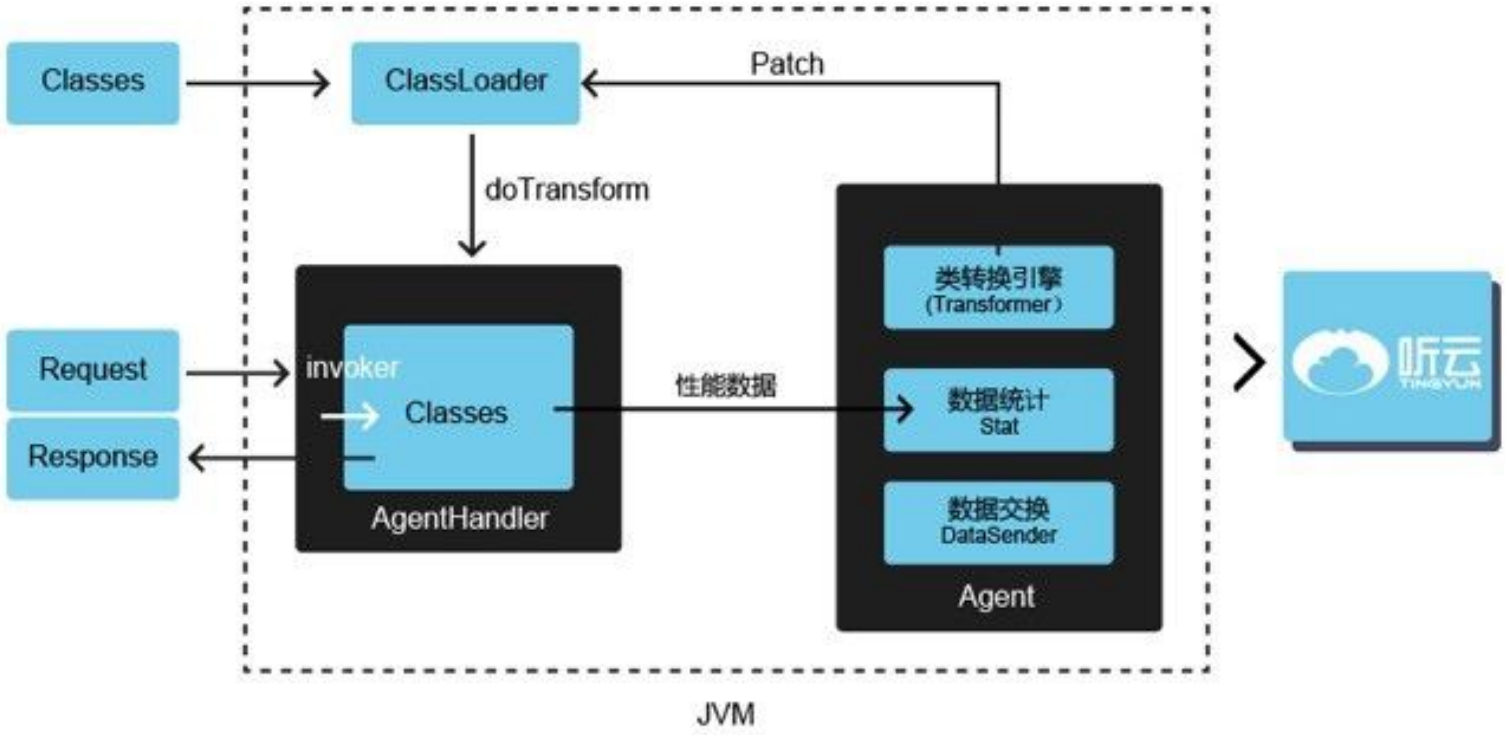
```
public class ApmAgent {
    public static void premain(String agentArgs, Instrumentation inst) {
        final ClassPool cp = ClassPool.getDefault();
        inst.addTransformer(new ClassFileTransformer() {
            public byte[] transform(ClassLoader loader,
                    String className, Class<?> classBeingRedefined,
                    ProtectionDomain protectionDomain,
                    byte[] classfileBuffer) throws IllegalClassFormatException {
                if("xxoo/demo/Xxoo".equals(className)) {
                    try {
                        String clazzName = className.replace('/', '.');
                        cp.insertClassPath(new ByteArrayClassPath(clazzName, classfileBuffer));
                        CtClass cc = cp.get(clazzName);
                        CtMethod cm = cc.getDeclaredMethod("xxoo");
                        cm.insertBefore("APM.start();");
                        cm.insertAfter("APM.end();");
                        cm.insertAfter("System.out.println(\"xxoo invoked in: \" + APM.elapsed() + \" ms.\");");
                        return cc.toBytecode();
                    } catch (Exception e) {
                    }
                }
                return classfileBuffer;
            }
        });
    }
}
```

自动注入监控代码

自动化APM监控

```
G:\Projects\dev\xxoo-java-agent-demo>java -cp bin -javaagent:target\apm-agent.ja
r xxoo.demo.Xxoo
xx!
oo!
xxoo invoked in: 2 ms.
```

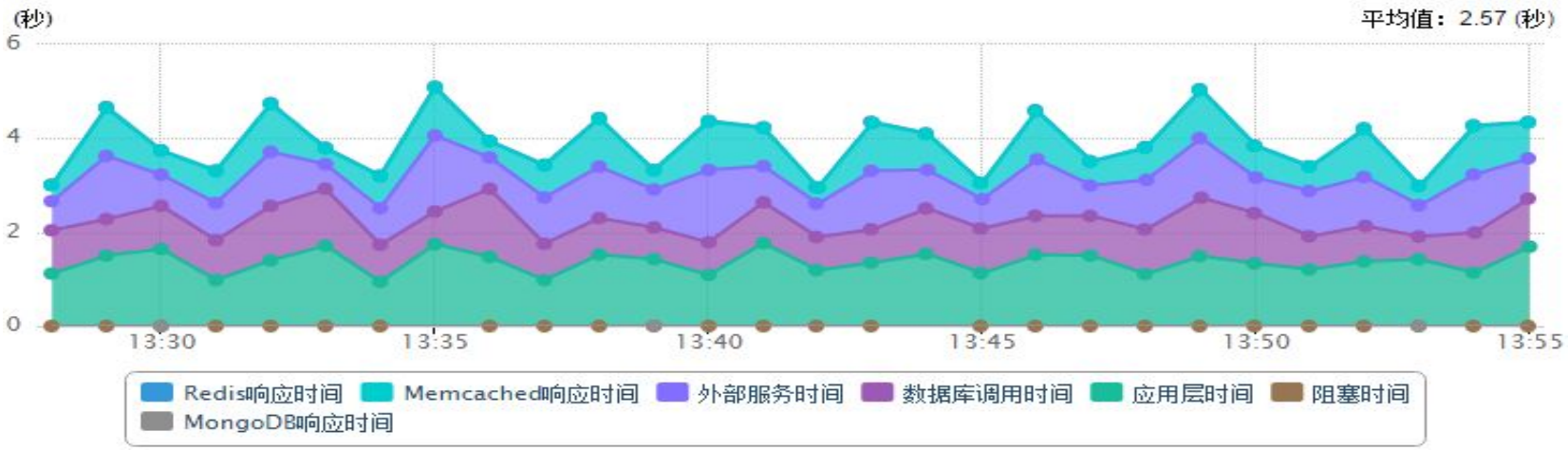
自动化APM监控


Classes
ClassLoader
Patch
doTransform
Request
Response
invoker
Classes
AgentHandler
性能数据
类转换引擎
(Transformer)
数据统计
Stat
数据交换
DataSender
Agent
听云
JVM

自动化APM监控

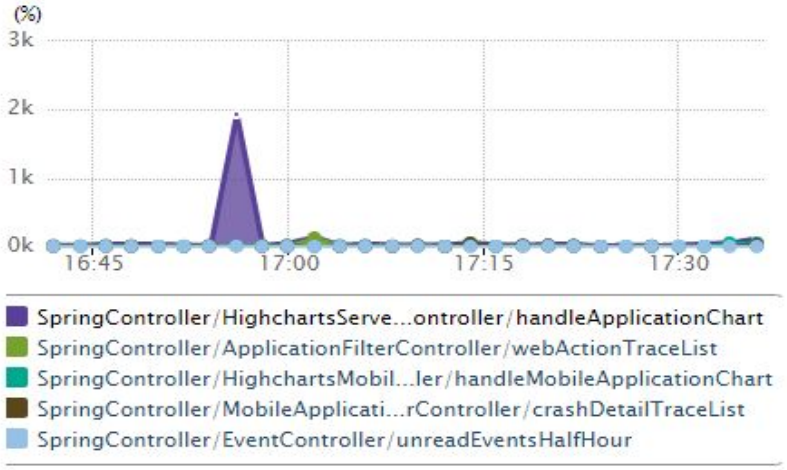
- 只在必要的位置嵌入监控代码
  - 数据库
  - NoSQL
  - HTTP
  - MQ
  - 可能存在性能瓶颈的业务代码

应用服务器响应时间



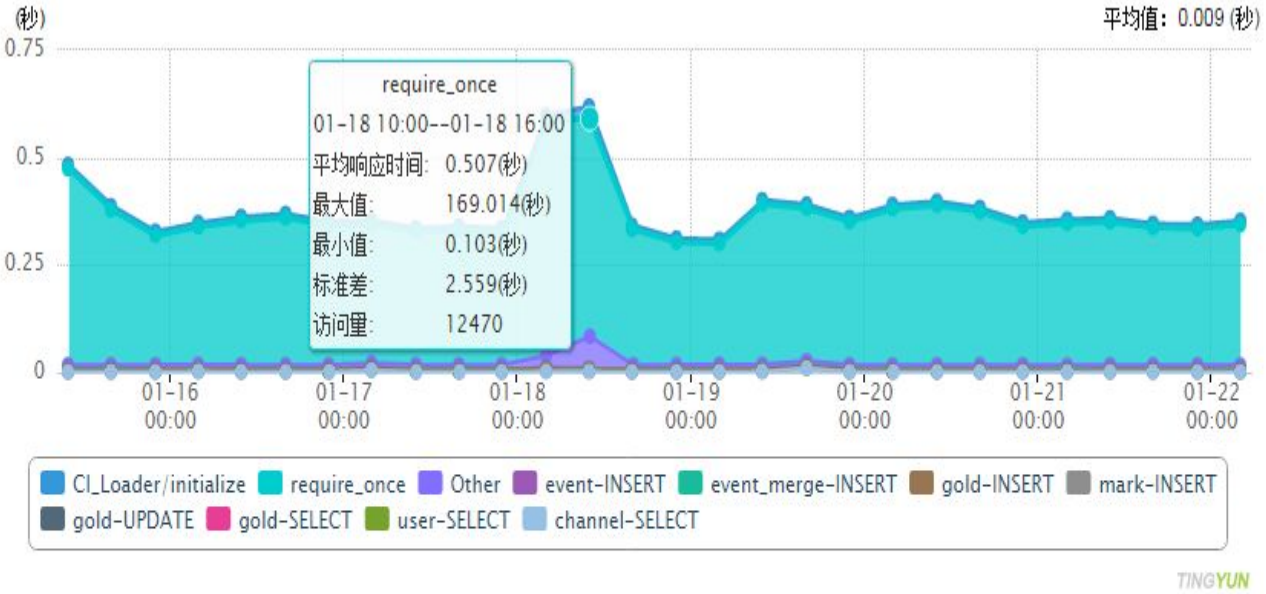
错误率

最耗时Web应用过程（Web Action）图表



CI/snsforrob/be_followed	75.67%
CI/snsforrob/batch_mark	20.88%
CI/u/feed	1.35%
CI/u/relation	1.02%
CI/show/htm	0.51%
CI/pages/index	0.33%
CI/m/topview	0.11%
CI/miaopai/topic	0.09%
CI/pages/favicon	0.01%
CI/pages/robots	0.01%
CI/m/gift_gif	0.01%
CI/backend	0.01%
CI/hongbao/up_pic	0.01%
CI/pages/aboutus	0.00%
CI/miaopai	0.00%
CI/miaopai/plaza	0.00%
CI/hongbao/star	0.00%
CI/hongbao	0.00%
CI/pages/hiring	0.00%
CI/admin_media/video	0.00%

应用过程性能分解堆叠图



应用过程分解表格

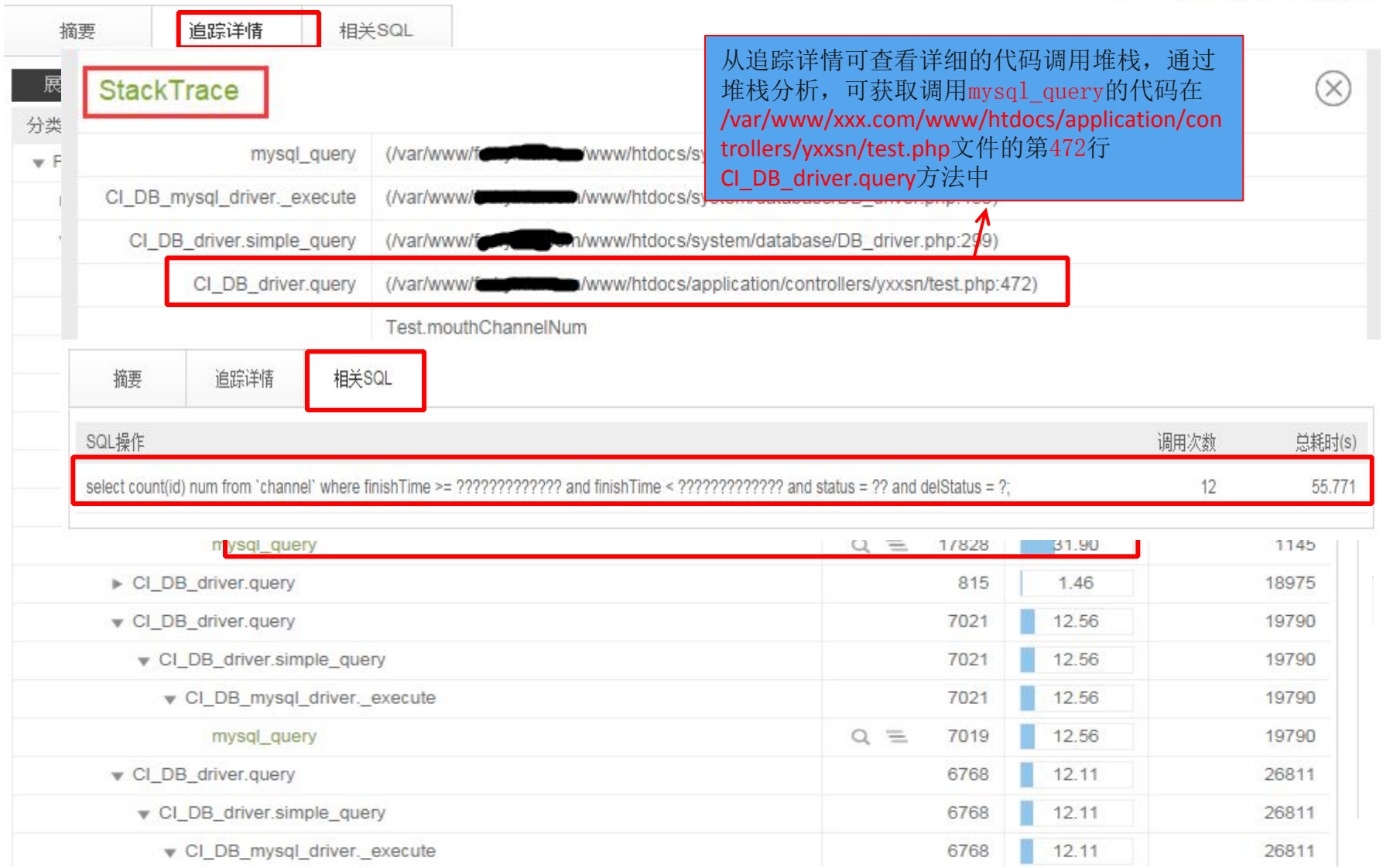
代码段	性能分类	耗时百分比(%)	平均调用次数	平均响应时间(ms)
require_once	PHP	87.5	344411	348
Another	Composite	11.7	14194756	0
Other	PHP	0.7	63066	16

摘要 | 追踪详情 | 相关SQL

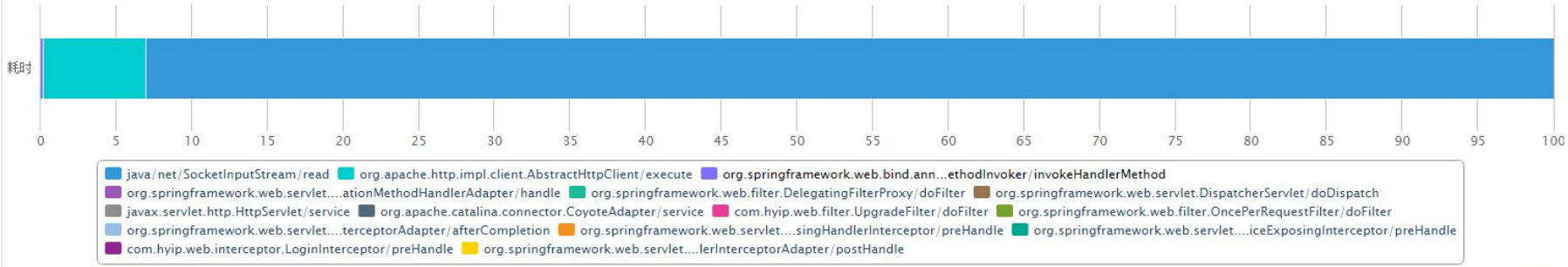
展开所有 | 全部关闭

分类	持续时间(ms)	时间占比(%)	时间偏移量(ms)
▼ PHP.execute	169042	100.00	0
▼ require_once	169040	100.00	2
▼ call_user_func_array	168995	99.97	47
▼ Snsforrob.batch_mark	168995	99.97	47
▼ Snsforrob.do_mark	168884	99.91	47
▼ CI_Loader.library	94798	56.08	47
CI_Loader._ci_load_class	94798	56.08	47
▼ CI_Loader.library	74060	43.81	94845
▼ CI_Loader._ci_load_class	74060	43.81	94845
▼ CI_Loader._ci_init_class	74057	43.81	94848
▼ FeedService.__construct	74057	43.81	94848
▼ FeedService.init	74057	43.81	94848
▼ CI_Loader.model	74034	43.80	94848
▼ CI_Loader.database	11322	6.70	94848
DB	11321	6.70	94849
▼ Channel_Model.__construct	36521	21.60	132361
▼ DB	36521	21.60	132361
▼ CI_DB_driver.initialize	36521	21.60	132361
▼ CI_DB_mysql_driver.db_connect	36519	21.60	132361
mysql_connect	36519	21.60	132361

FeedService/
CI_DB_mysql


摘要
追踪详情
相关SQL
StackTrace
mysql_query (/var/www/[redacted]/www/htdocs/s
CI_DB_mysql_driver._execute (/var/www/[redacted]/www/htdocs/s
CI_DB_driver.simple_query (/var/www/[redacted]/www/htdocs/system/database/DB_driver.php:299)
CI_DB_driver.query (/var/www/[redacted]/www/htdocs/application/controllers/yxxsn/test.php:472)
Test.mouthChannelNum
从追踪详情可查看详细的代码调用堆栈，通过堆栈分析，可获取调用mysql_query的代码在/var/www/xxx.com/www/htdocs/application/controllers/yxxsn/test.php文件的第472行CI_DB_driver.query方法中
摘要
追踪详情
相关SQL
SQL操作
调用次数
总耗时(s)
select count(id) num from `channel` where finishTime >= ????????????? and finishTime < ????????????? and status = ?? and delStatus = ?;
12
55.771
mysql_query 17828 31.90 1145
CI_DB_driver.query 815 1.46 18975
CI_DB_driver.query 7021 12.56 19790
CI_DB_driver.simple_query 7021 12.56 19790
CI_DB_mysql_driver._execute 7021 12.56 19790
mysql_query 7019 12.56 19790
CI_DB_driver.query 6768 12.11 26811
CI_DB_driver.simple_query 6768 12.11 26811
CI_DB_mysql_driver._execute 6768 12.11 26811

摘要 | 追踪详情 | 相关SQL



最慢组件	调用次数	持续时间(s)	占比(%)
java/net/SocketInputStream/read	69	86.975	93.007
org.apache.http.impl.client.AbstractHttpClient/execute	1	6.349	6.789
org.springframework.web.bind.annotation.support.HandlerMethodInvoker/invokeHandlerMethod	1	0.182	0.195
org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter/handle	1	0.003	0.003

服务器响应时间：93.513 (s)

实例信息：JAVA:host-35:8080

URL：/api/v30/account4Client/friendRecommend

httpStatus：200

threadName：http-nio-8080-exec-209

referer：

摘要 | 追踪详情 | 相关SQL

展开所有 | 全部关闭

分类	持续时间(ms)	时间占比(%)	时间偏移量(ms)
CoyoteAdapter.service	93514	100.00	0
CoyoteAdapter.service	93514	100.00	0
UpgradeFilter.doFilter	93513	100.00	0
OncePerRequestFilter.doFilter	93513	100.00	0
DelegatingFilterProxy.doFilter	93513	100.00	0
HttpServlet.service	93511	100.00	1
DispatcherServlet.doDispatch	93510	100.00	1
AbstractUrlHandlerMapping$PathExposingHandlerInterceptor.preHandle	0	0.00	2
ConversionServiceExposingInterceptor.preHandle	0	0.00	2
LoginInterceptor.preHandle	0	0.00	2
AnnotationMethodHandlerAdapter.handle	93509	100.00	2
HandlerMethodInvoker.invokeHandlerMethod	93506	99.99	3
AbstractHttpClient.execute	6349	6.79	4
java/net/SocketInputStream.read	86975	93.01	6354
HandlerInterceptorAdapter.postHandle	0	0.00	93511
HandlerInterceptorAdapter.postHandle	0	0.00	93511
HandlerInterceptorAdapter.postHandle	0	0.00	93511

API接口执行时间6.3秒

URL

http://user.qianbao666.com/api/getRecommendPeople.html

读取API返回结果花费86秒

应用过程慢追踪

应用：user.qianbao666.com

追踪时间：2015-05-19 07:30:59

服务器响应时间：7.715 (s)

实例信息：JAVA:host-19:8480

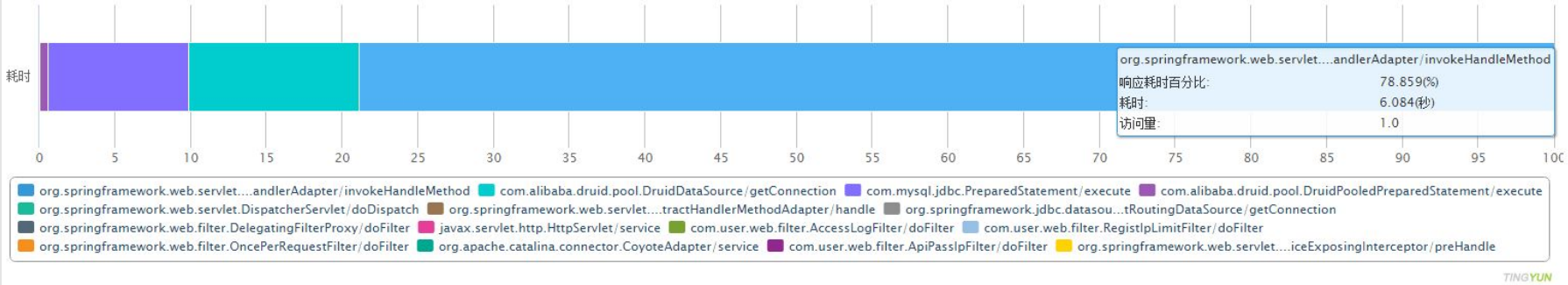
URL：/api/getRecommendPeople.html

httpStatus：200

threadName：http-nio-8480-exec-37

referer：

摘要 | 追踪详情 | 相关SQL



最慢组件	调用次数	持续时间(s)	占比(%)
org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter/invokeHandleMethod	1	6.084	78.859
com.alibaba.druid.pool.DruidDataSource/getConnection	996	0.866	11.225
com.mysql.jdbc.PreparedStatement/execute	996	0.719	9.32
com.alibaba.druid.pool.DruidPooledPreparedStatement/execute	997	0.039	0.506
org.springframework.web.servlet.DispatcherServlet/doDispatch	1	0.003	0.039

▼ DispatcherServlet.doDispatch		6331	99.97	0
ConversionServiceExposingInterceptor.preHandle		0	0.00	2
▼ AbstractHandlerMethodAdapter.handle		6329	99.94	2
▼ RequestMappingHandlerAdapter.invokeHandleMethod		6328	99.92	2
AbstractRoutingDataSource.getConnection		1	0.02	2
▶ com.alibaba.druid.pool.DruidPooledPreparedStatement.execute	🔍	20	0.32	4
DruidDataSource.getConnection		1	0.02	99
▶ com.alibaba.druid.pool.DruidPooledPreparedStatement.execute	🔍	0	0.00	101
DruidDataSource.getConnection		0	0.00	102
▶ com.alibaba.druid.pool.DruidPooledPreparedStatement.execute	🔍	0	0.00	103

类似的SQL调用1994次，累计时间约2秒

SQL

```
SELECT req.alias_name alias,
    hui.nick_name,
    us.username   user_name,
    req.user_id,
    us.id        friend_id,
    upload_avata_time
FROM   hyip_user us
    LEFT JOIN hyip_user_info hui
    ON us.id=hui.user_id
    LEFT JOIN im_friend_request req
    ON us.id=req.friend_id AND req.state=? AND req.user_id=?
WHERE  us.id = ?
```

THANKS