

NJSD

中国（南京）软件开发者大会

China (Nanjing) Software Developers Conference

2016



NJSD

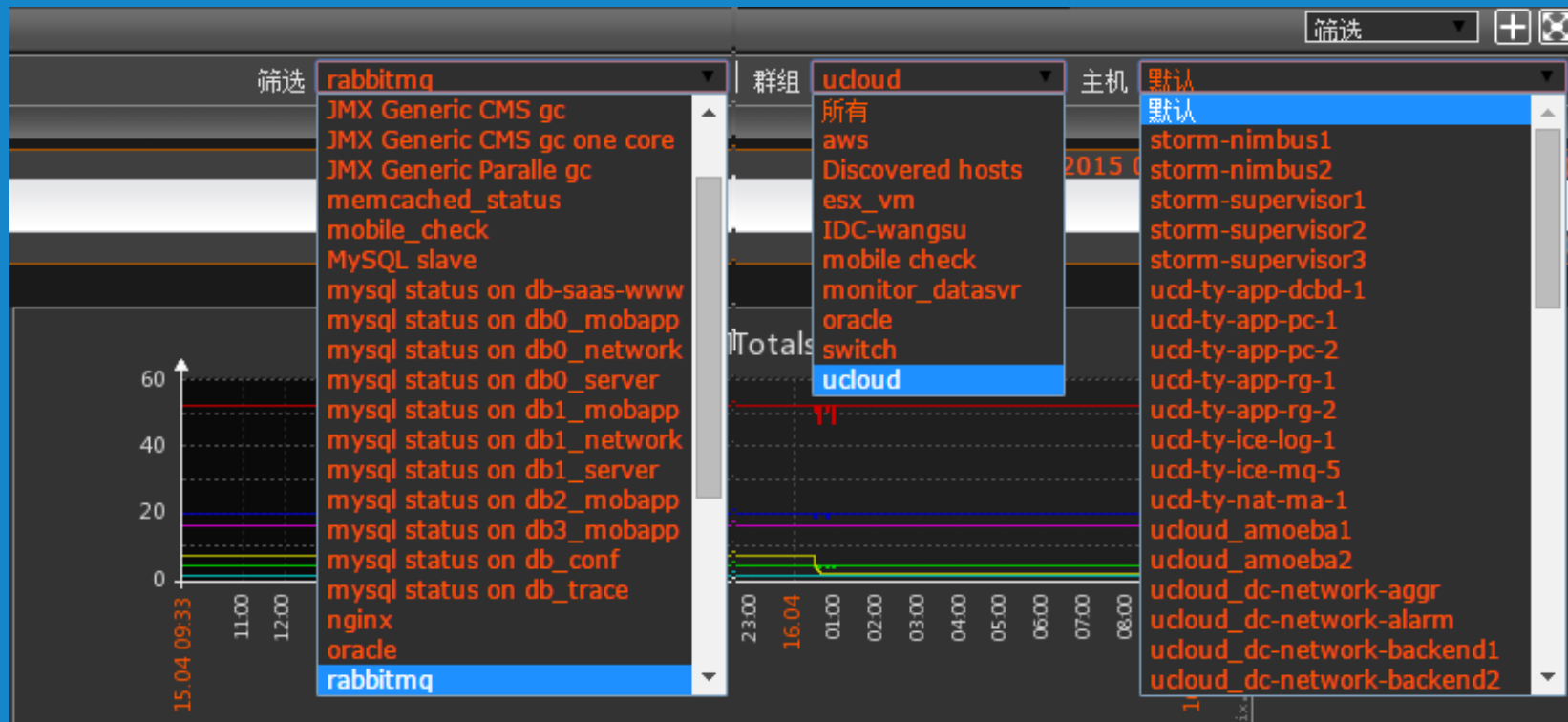
中国（南京）软件开发者大会
China (Nanjing) Software Developers Conference

2016

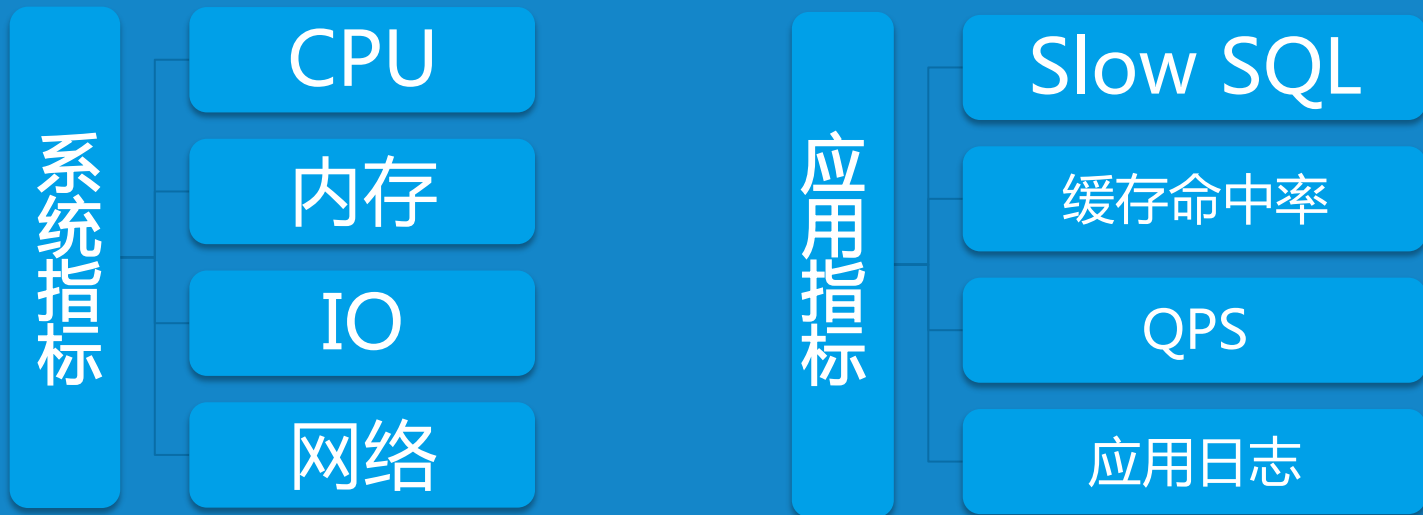


快速迭代中的 精益应用性能管理

廖雄杰



传统应用性能监控



快速迭代下传统监控的挑战

用户向移动端转移

应用向云端转移

产品迭代快

复杂的技术结构

快速迭代下传统监控的挑战

如何监控应用？

RDB

NoSQL
(Redis
MC
Mongo)

API/
RPC

云服务

MQ

精益化绩效管理



精益化绩效管理

应用性能
是可以精确度量的！



精益化绩效管理

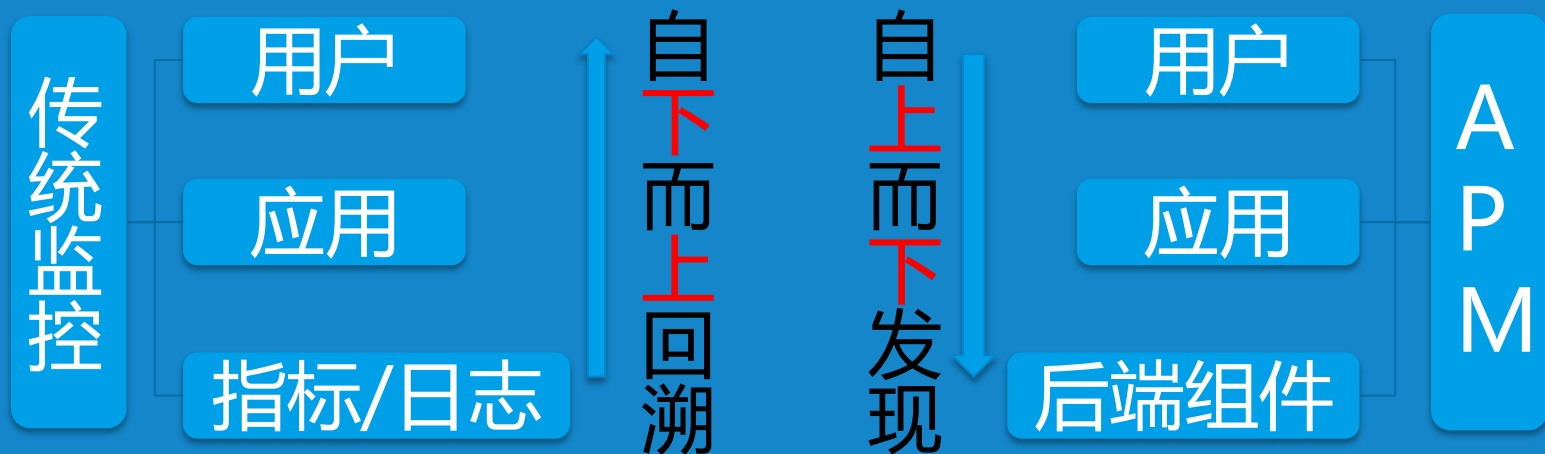
APM

(Application Performance Management)

精益化绩效管理

Application as Monitoring

精益化性能管理



APM：深入应用代码的性能监控

```
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;  
    }  
}
```

1. 获取方法开始时间

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

1

`-javaagent:apm-agent.jar`

2

Java classloader

3

Instrumentation/bytecode

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 clazName = className.replace('/', '.');  
                        cp.insertClassPath(new ByteArrayClassPath(clazName, classfileBuffer));  
                        CtClass cc = cp.get(clazName);  
                        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.jar  
r xxoo.demo.Xxoo  
xx!  
oo!  
xxoo invoked in: 2 ms.
```


精益化性能管理

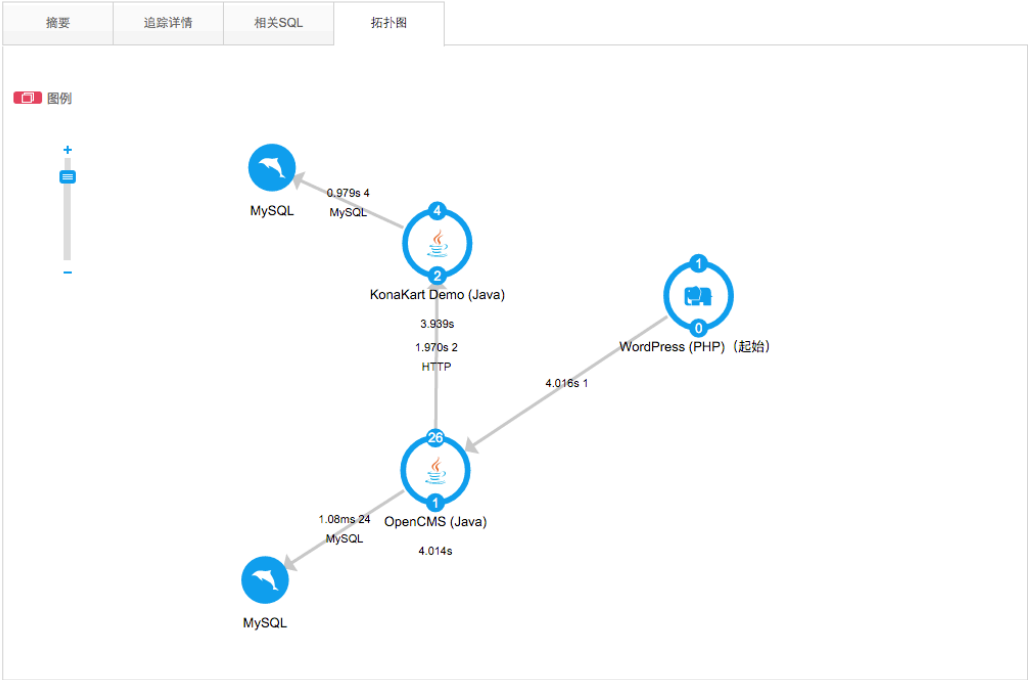
应用过程: [URL/index.php?p=68](#)

追踪时间: 2016-03-08 09:45:21

服务器响应时间: 6.29 (s)

实例信息: PHP:centos-php

共有 3 个应用追踪信息



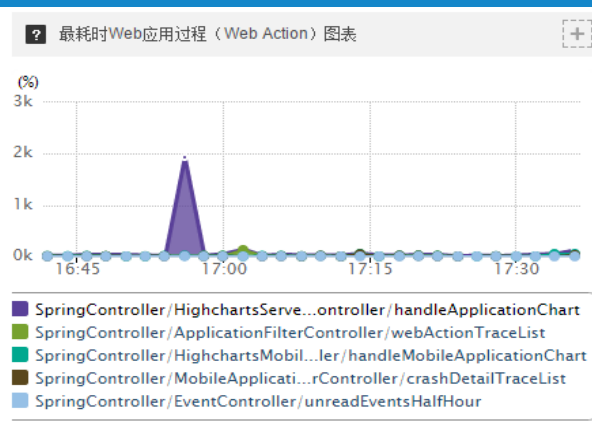
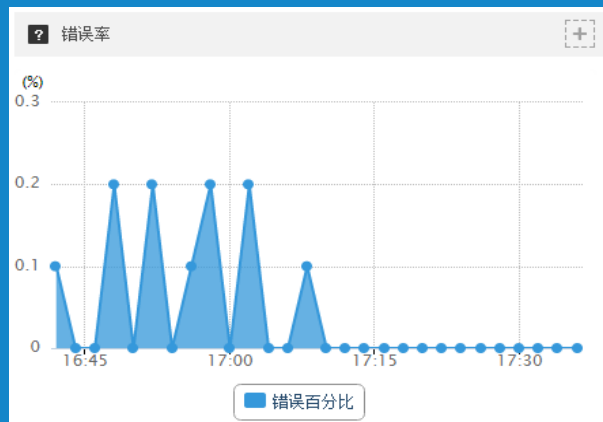
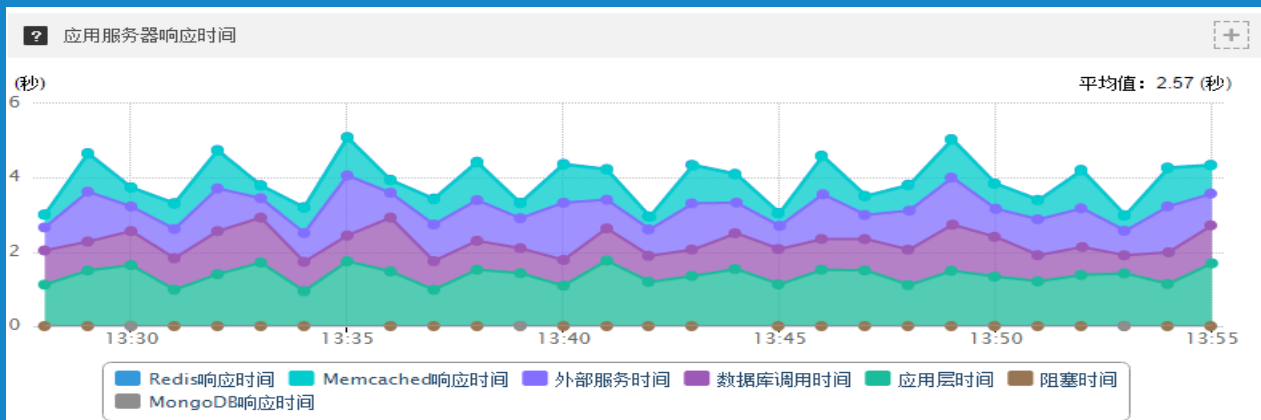
请求信息

请求URL: /?p=68
线程名称: pid-29875
HTTP响应: 200
referer: http://123.103.75.25/
user-agent: Wget/1.12 (linux-gnu)

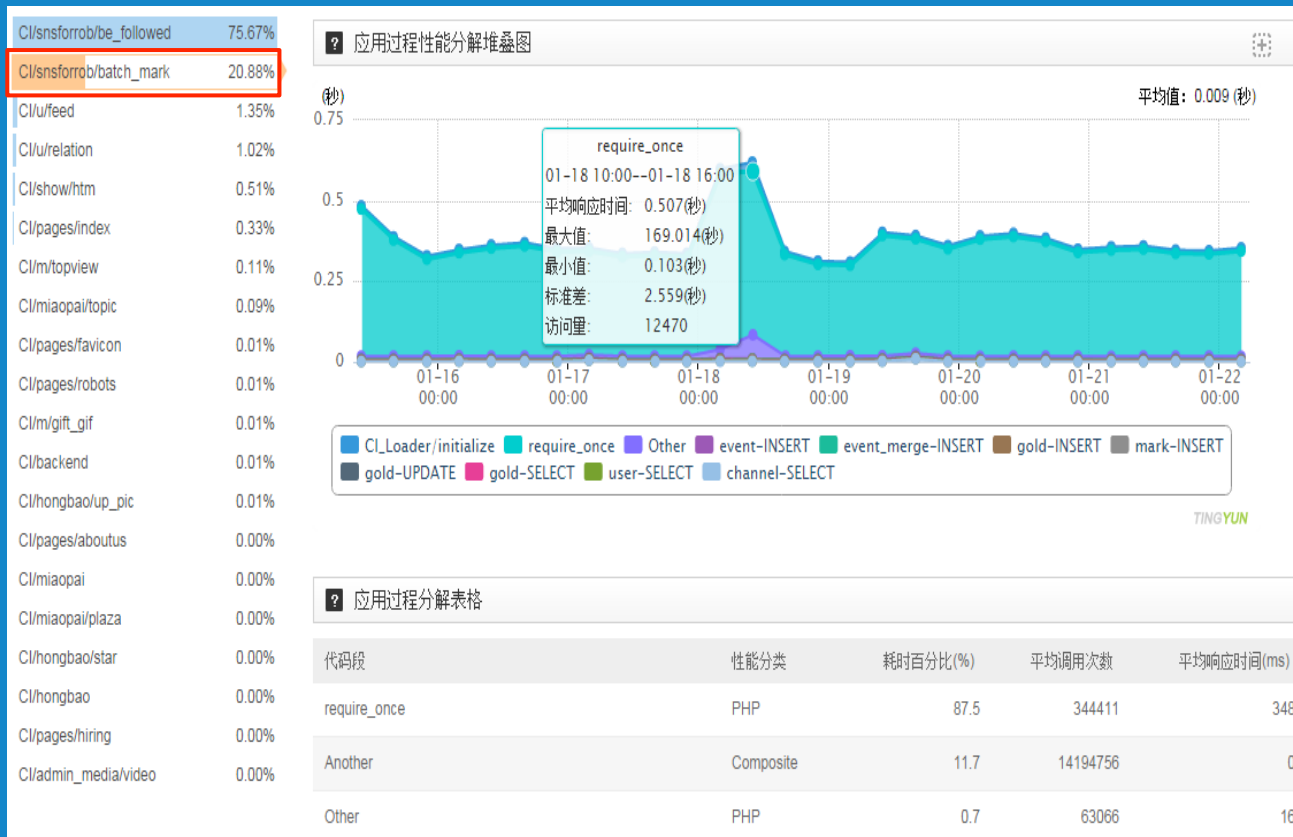
请求参数

p: 68

精益化性能管理



精益化性能管理



精益化性能管理

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

CI_Loader/

FeedService/

CI_DB_mysql

精益化性能管理

摘要 追踪详情 相关SQL

StackTrace

堆栈/代码行

分类

mysql_query (/var/www/.../www/htdocs/system/database/drivers/mysql/mysql_driver.php:194)

CI_DB_mysql_driver_execute (/var/www/.../www/htdocs/system/database/DB_driver.php:453)

CI_DB_driver.simple_query (/var/www/.../www/htdocs/system/database/DB_driver.php:299)

CI_DB_driver.query (/var/www/.../www/htdocs/application/controllers/yxxsn/test.php:472)

Test.mouthChannelNum

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 = ?;

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

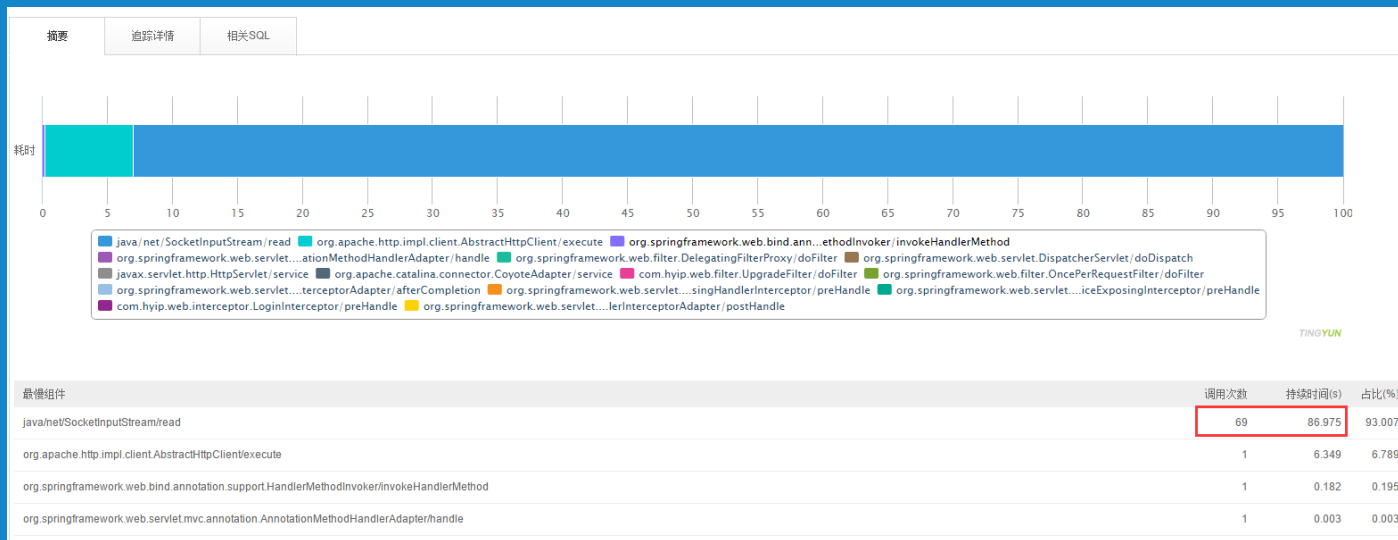
mysql_query 7019 12.56 19790

CI_DB_driver.query 6768 12.11 26811

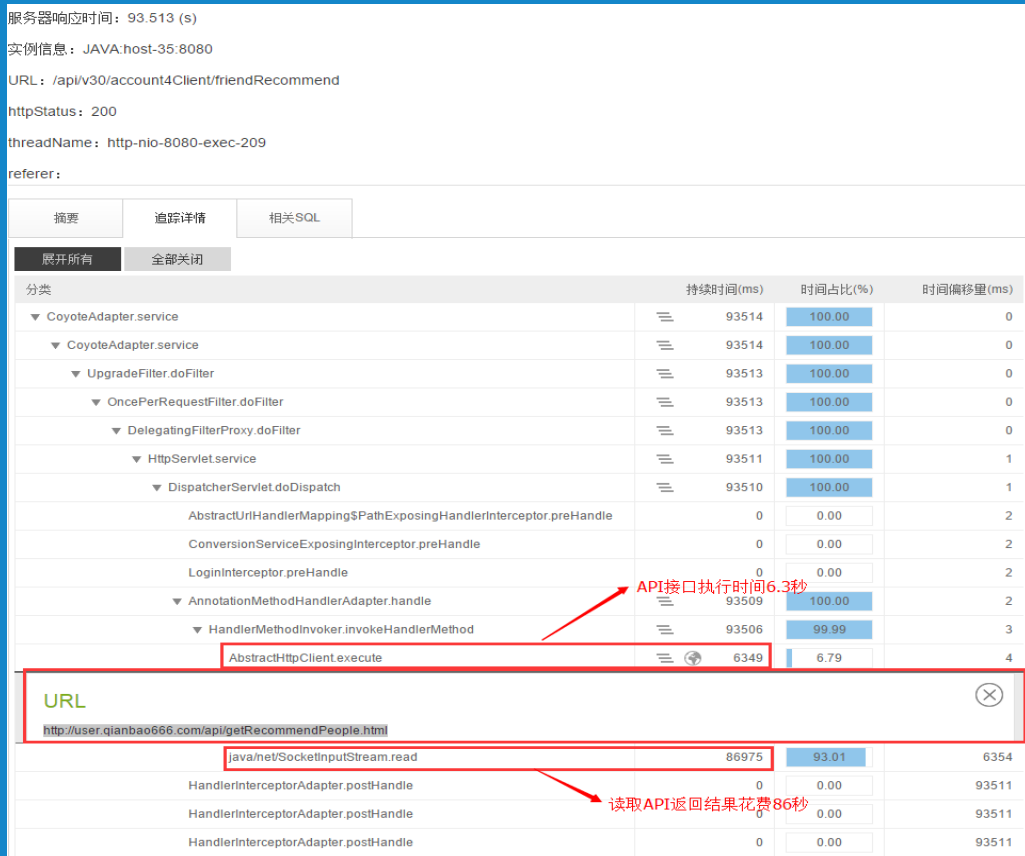
CI_DB_driver.simple_query 6768 12.11 26811

CI_DB_mysql_driver_execute 6768 12.11 26811

服务端调用链追踪



服务端调用链追踪



服务端调用链追踪

▼ 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	Q	20	0.32	4
DruidDataSource.getConnection		1	0.02	99
▶ com.alibaba.druid.pool.DruidPooledPreparedStatement.execute	Q	0	0.00	101
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 = ?				
DruidDataSource.getConnection		0	0.00	102
▶ com.alibaba.druid.pool.DruidPooledPreparedStatement.execute	Q	0	0.00	103

类似的SQL调用1994次，累计时间约2秒

APM嵌码中的那些坑儿

APM嵌码中的那些坑儿

并非所有的逻辑都需要嵌码!

RDB ✓

系统函数 X

NoSQL ✓

大部分业务代码 X

HTTP ✓

MQ ✓

APM嵌码中的那些坑儿

系统高压下嵌码逻辑雪上加霜

解决办法：

高压下监控自动降级，转采样！

极端情况下自动禁用！

APM嵌码中的那些坑儿

不确定业务代码是否有性能瓶颈，
怎么破？

解决办法：

让监控在运行过程中自我学习，自动监
控热点代码

THANK YOU