

维多利亚的秘密 — 发现隐藏在互联网架构冰山下的性能问题

演讲人：廖雄杰



互联网架构下的性能问题
互联网架构下的性能发现
应用性能分析与优化

互联网架构下的性能问题

duǎng

甍

“悦耳”的运维警报响起!!!





应用前端响应缓慢☹️

COMPLAINT CENTER

— 投诉中心 —



用户开始投诉☹️



怎么办？

穿上滑板鞋，各种查监控!!!





- 应用系统：CPU/内存使用正常！日志无异常！
- 缓存：QPS、命中率正常！
- 数据库：CPU、IO正常！无明显慢查询！

怎么办??

打电话给可敬又可爱的程序猿哥哥...吧



运维：喂！兄弟！

程序猿：什么事，兄弟？系统又出问题了？

运维：是啊，可以帮忙加几行调试日志吗？

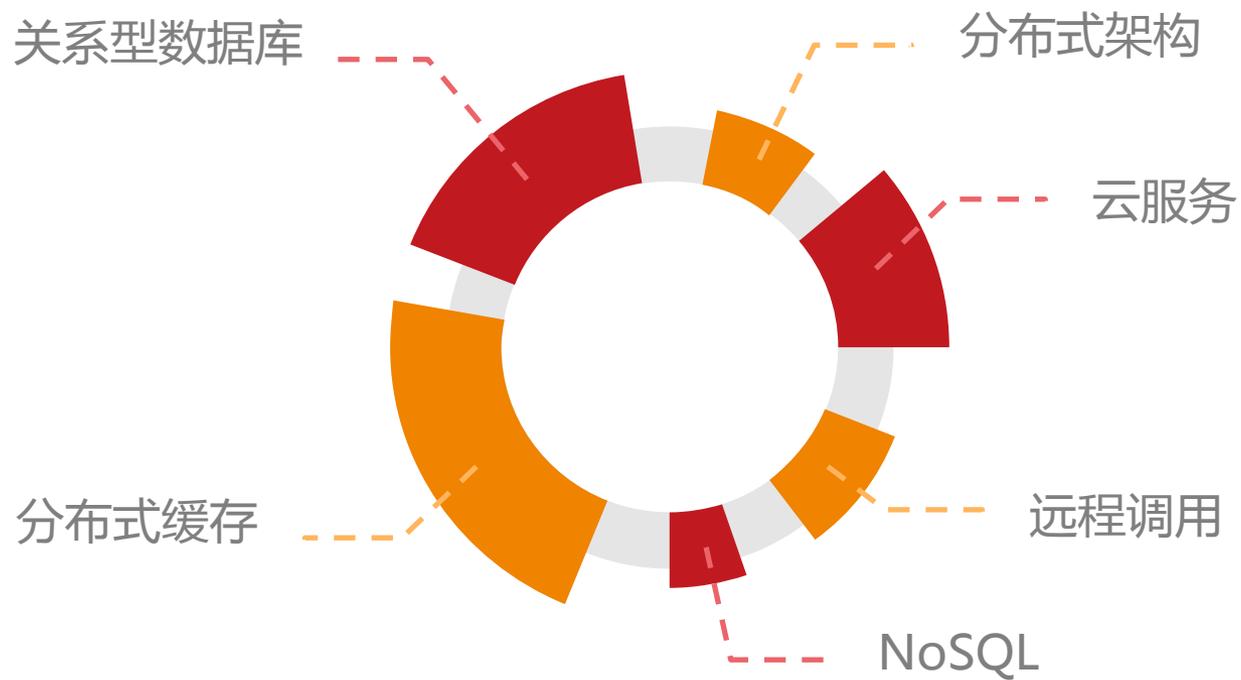
程序猿：好(Zhe)的(Jiao)，没(Shen)问(Me)题(Shi)！

10分钟过去了。

半小时过去了。。

1小时过去了。。。。

一定是哥查问题的姿势不对!!!



庞杂的监控指标（此处省略100项.....）

- 1 CPU/内存/IO
- 2 网络延时
- 3 数据库响应
- 4 缓存命中率

谁是导致应用性能问题的元凶？

它真的影响到了应用性能吗？

自下而上？OR 自上而下？

姿势很重要！

用户、应用、性能之间的关系就像.....



那么问题来了，如何与应用关联？

互联网架构下的性能发现

APM

(Application Performance Management)

深入**应用代码**的性能监控

帮助我们自上而下地发现性能问题

```
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

自动嵌入监控代码！

Agent 嵌码技术

Java

Bytecode/Instrumentation/Classloader

PHP

Opcode/Zend/Extensions/Xhprof

iOS

Hook/Swizzle

Android

Dalvik/Class rewriting

哪些地方需要嵌码？

原则

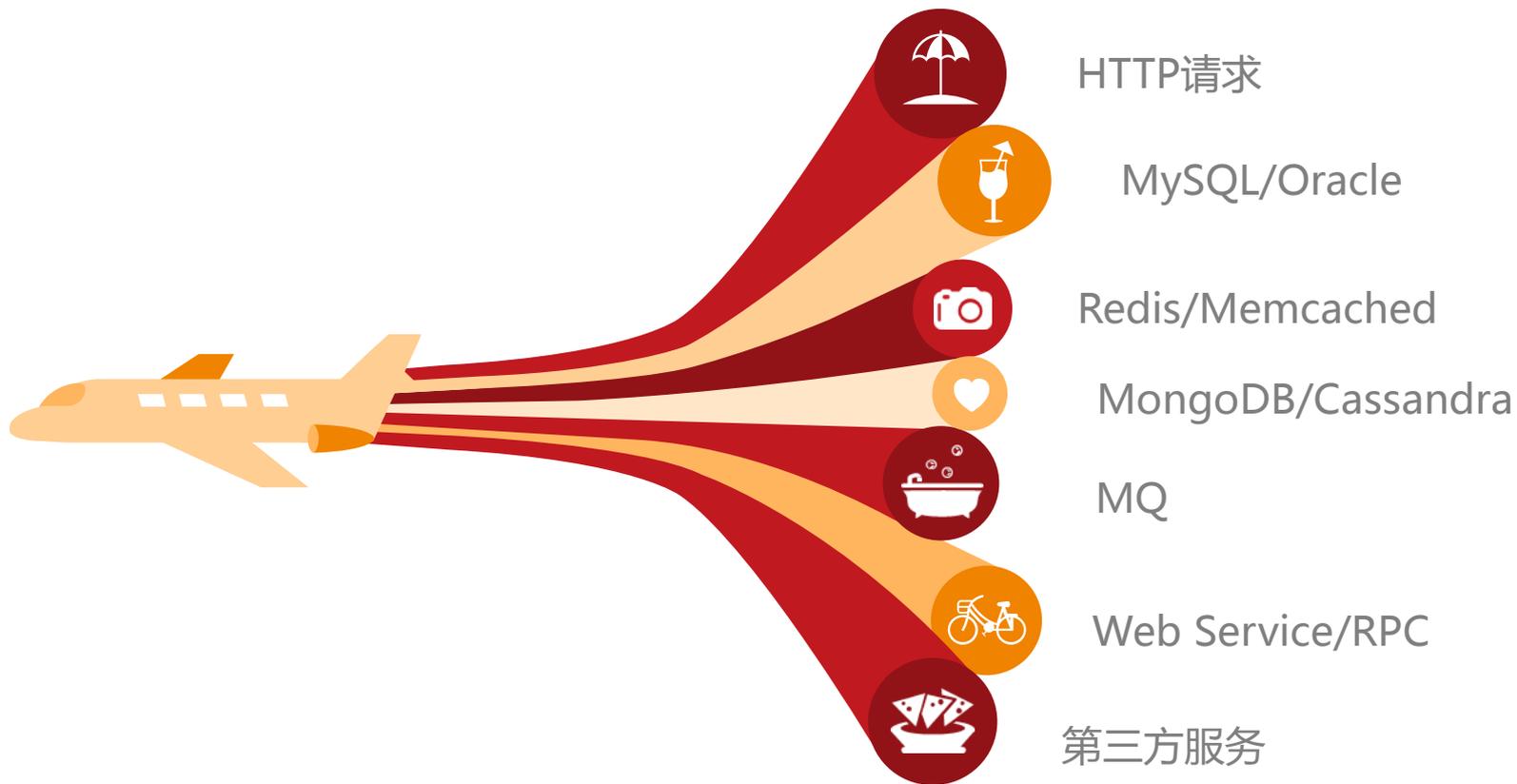
- 最少嵌入！
- 只在必要的地方嵌码！



必要

- 可能形成性能瓶颈，嵌码！
- 可能诱发其它性能问题，嵌码！





自上而下的性能发现

➤ MySQL slowlog

```
mysql> set global slow_query_log='on';
```

```
mysql> show variables like '%slow%';
```

Variable_name	Value
log_slow_queries	ON
slow_launch_time	2
slow_query_log	ON
slow_query_log_file	/usr/local/mysql/data/██████-slow.log

```
mysql> show variables like 'long%';
```

Variable_name	Value
long_query_time	10.000000

➤ MySQL slowlog

```
ken tmp # mysqldumpslow /tmp/slow-log
Count: 414 Time=3.51s (1454s) Lock=0.00s (0s) Rows=21974.9 (9097604), arch[arc
select post_title, post_content from wp_posts limit N,N
Count: 200 Time=3.40s (681s) Lock=0.00s (0s) Rows=15428.0 (3085600), arch[arc
select post_title, post_content from wp_posts where post_date < sysdate() lim
Count: 207 Time=3.16s (654s) Lock=0.00s (0s) Rows=30000.0 (6210000), arch[arc
select post_title, post_content from wp_posts where post_content like 's' lim
Count: 195 Time=3.09s (603s) Lock=0.00s (0s) Rows=25428.0 (4958460), arch[arc
select post_title, post_content from wp_posts where post_category = N limit
```

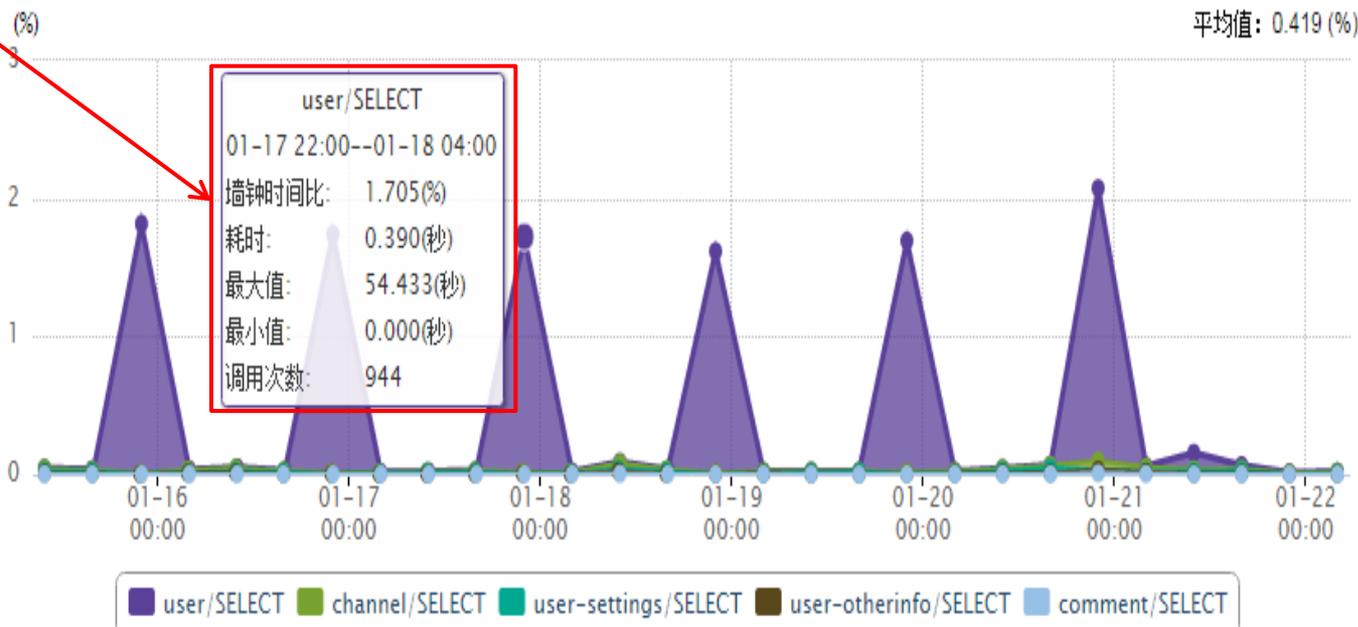
SQL性能分析：User/SELECT最长

排序 SQL耗时 显示Top20

数据库一览

最耗时SQL操作堆叠图

user/SELECT	2308.480秒
channel/SELECT	96.874秒
user-settings/SELECT	58.862秒
user-otherinfo/SELECT	40.054秒
comment/SELECT	10.533秒
follow/SELECT	5.257秒
gold/SELECT	5.247秒
sinauserinfo/SELECT	3.117秒
people/SELECT	1.102秒
user-sinapaik/SELECT	0.494秒
sku_order/SELECT	0.364秒
sku_list/SELECT	0.242秒



SQL性能分析：User/SELECT

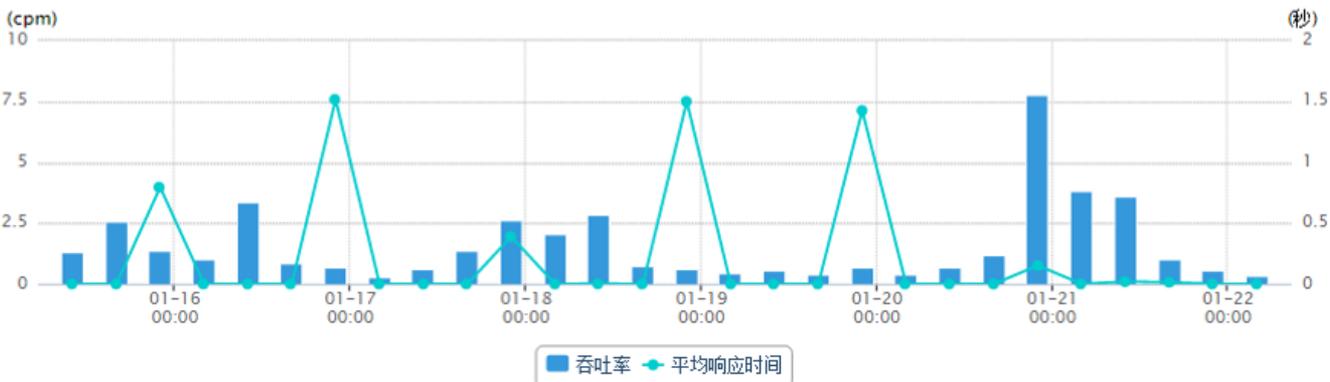
排序 平均响应时间 显示Top20

数据库一览

people/SELECT	0.157秒
user/SELECT	0.146秒
follow/SELECT	0.013秒
news/SELECT	0.005秒
weibojob/SELECT	0.003秒
comment/INSERT	0.002秒
sku_order/UPDATE	0.002秒
channel_extra/INSERT	0.002秒
user/UPDATE	0.002秒
comment/UPDATE	0.002秒
user-settings/SELECT	0.002秒
user-sinapaik/SELECT	0.002秒
sinauserinfo/SELECT	0.002秒
op_stat/SELECT	0.002秒
comment_warehouse/SELECT	0.002秒
news/INSERT	0.001秒

返回

数据库响应时间和吞吐率曲线图



TINGYUN

调用者耗时图表



- CI/.../index_api
- CI/integral/cashList
- Script/var/www/.../www/htdocs/index.php
- CI/robcomment/add
- CI/.../topic
- CI/snsforrob/be_followed
- CI/integral/exchangeList
- CI/snsforrob/batch_mark
- CI/show/htm
- Script/var/www/.../www/htdocs/index.php
- CI/u/feed
- CI/m/rpt_ios_devid
- CI/integral/exchangeInfo
- CI/user/loginhtm
- CI/integral/cashInfo
- CI/robcomment/com
- CI/robuser/index

SQL分析

慢SQL追踪

执行计划分析:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	user	range	Index_status	Index_status	4	null	3389003	Using where; Using filesort

Trace详情

rows条数过多

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/.../m/www/htdocs/application/controllers/yxsn/robuser.php:68)
call_user_func_array	(/var/www/.../m/www/htdocs/system/core/CodeIgniter.php:359)
require_once	(/var/www/.../m/www/htdocs/index.php:205)

```
select id from user where user.status in (?, ?, -?) order by user.createTime desc limit ?, ?
```

SQL :

```
SELECT id from user  
WHERE status in (?,?,-?)  
ORDER BY createTime DESC  
LIMIT ?,?
```

优化 :

1. 单纯从DBA的角度优化很困难
2. 建议应用层配合，where中增加createTime条件

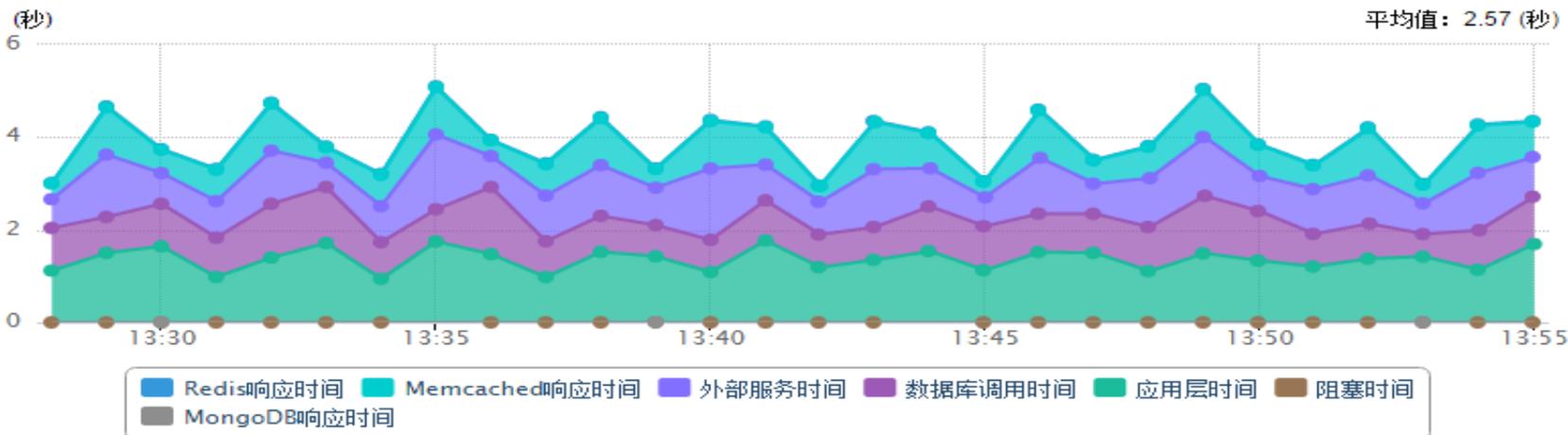
应用性能分析与优化

用户的期望是移动平台的速度能和PC上一样快

如果页面在**3秒**内打不开，**40%**的用户选择离开

摘自：11 Reasons to Care About Mobile Performance in 2015

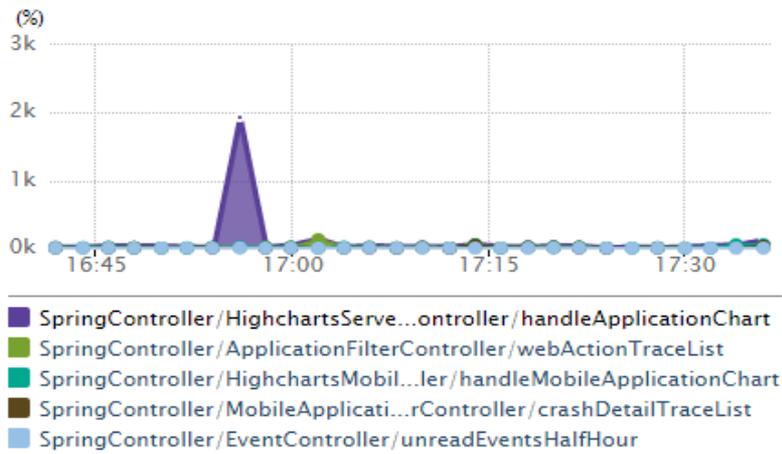
应用服务器响应时间



错误率



最耗时Web应用过程 (Web Action) 图表

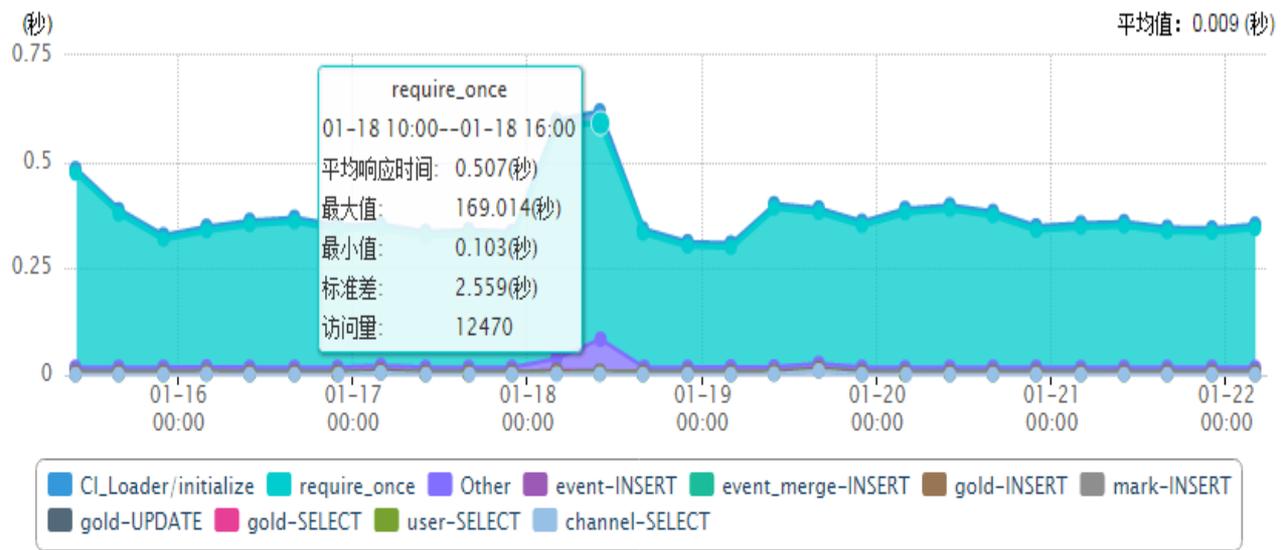


应用性能分析：CI/snsforrob/batch_mark耗时百分比达到 21%



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

应用性能分析：慢应用过程追踪

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

应用性能分析：慢应用过程追踪

摘要 | **追踪详情** | 相关SQL

StackTrace

mysql_query	(/var/www/.../www/htdocs/s/...)
CI_DB_mysql_driver_execute	(/var/www/.../www/htdocs/s/...)
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

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

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	调用次数	耗时(s)	总耗时(s)
mysql_query	1/828	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



NETWORKBENCH

基调网络 质量可视