

# PHP应用性能管理与实践

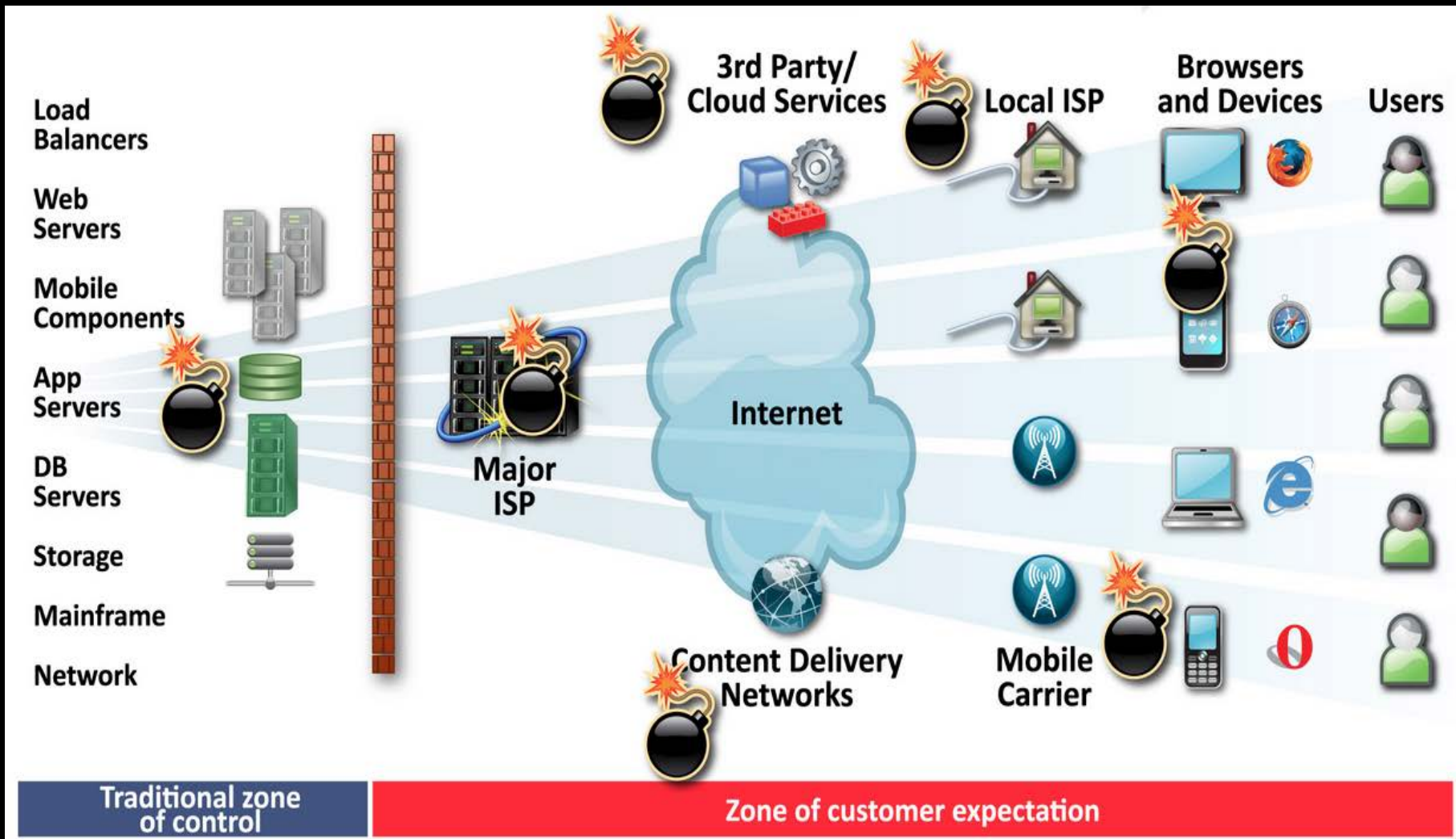
听云 甄志会



# 目录

- 应用性能管理简介
- PHP应用性能管理的实现原理
- 与常规性能工具的区别
- 图表演示
- 案例实践

# 复杂的应用交付链



# 应用性能管理？

## APM

### Application Performance Management

对软件应用的性能和可用性进行监控和管理，  
致力于发现和定位性能瓶颈和故障，  
以保证应用达到预期的服务水平（SLA）

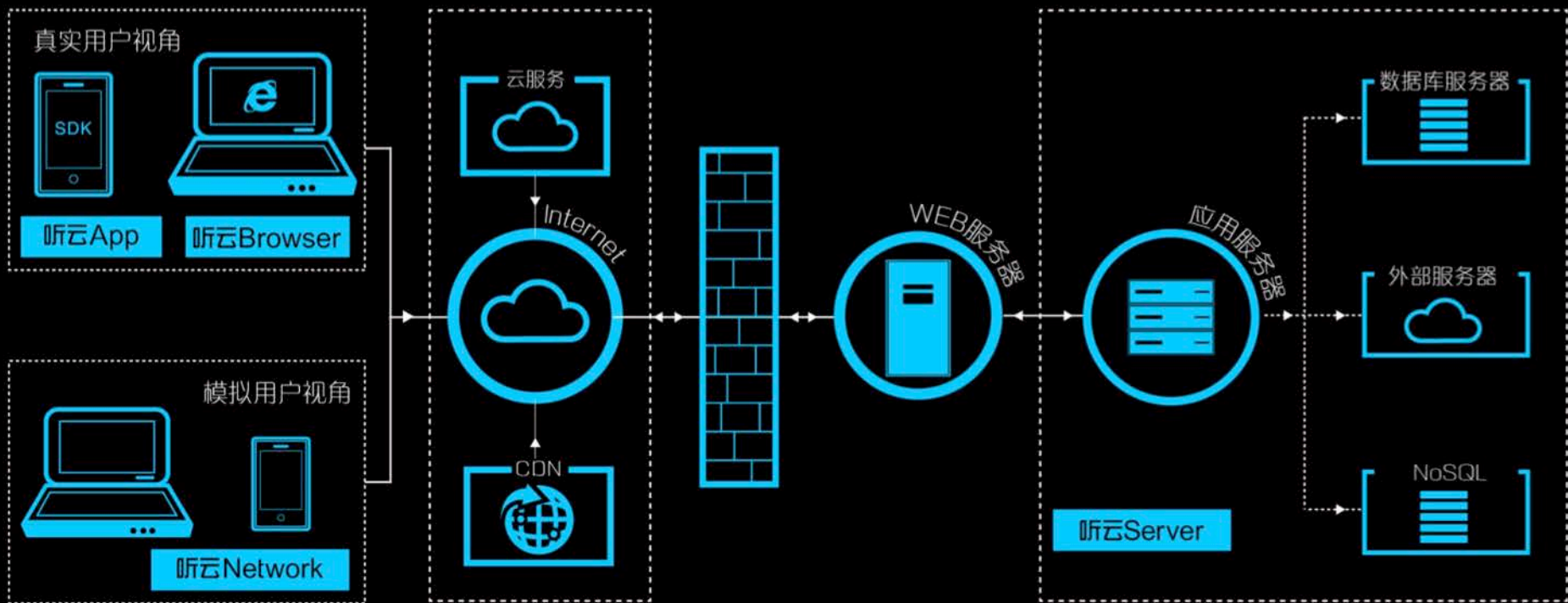
# 为什么需要APM

- 应用性能监测
  - 应用运营阶段
  - 复杂的生产环境
  - 发布后



- 应用性能测试
  - 开发测试阶段
  - 实验室测试环境
  - 发布前

# APM全景图



# SERVER端关注点

- 运行性能
- 服务质量（异常，错误）
- 应用拓扑
- 跨应用追踪

# PHP应用性能管理实现原理



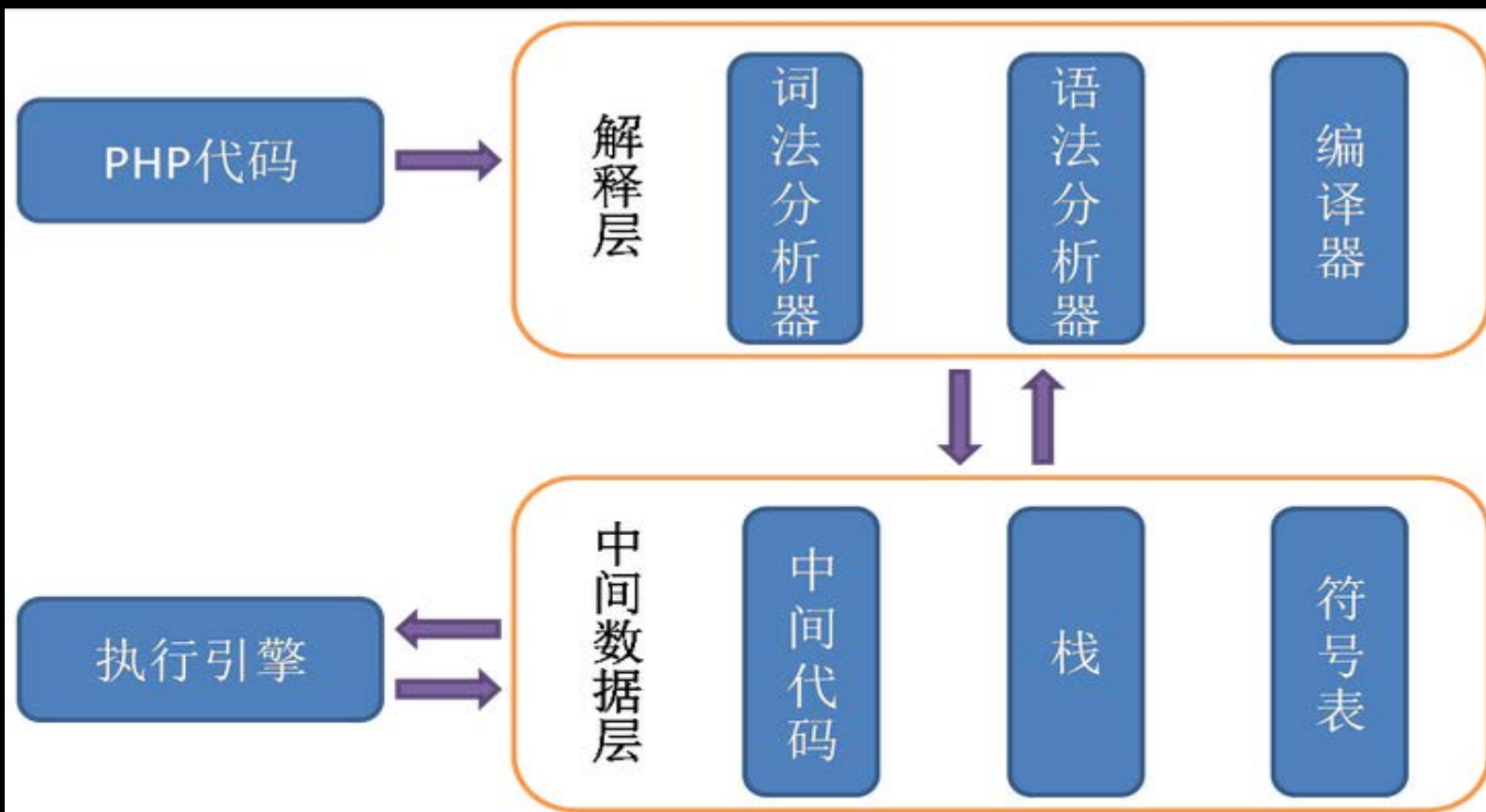
# 基本原理

## Zend/Opcache/Extensions

PHP有两大类模块：Zend引擎、扩展层

1. Zend引擎将PHP源文件解释为一系列操作码（opcode），然后循环执行操作码；
2. 扩展层是一组函数、类库和流，PHP使用它们来执行一些特定功能的操作，例如数据库功能、网络功能。

<http://www.php-internals.com/book/?p=chapt07/07-01-zend-vm-overview>



Zend 虚拟机体系结构图

# PHP扩展

利用扩展机制，增加特定功能，跟踪PHP引擎执行过程

相关著名的扩展：`xdebug` / `xhprof`

听云的PHP APM功能也是一个扩展，实现了跟踪请求及各种组件性能的功能。

# 嵌码原则

1. 可操作性：自动嵌码
2. 效率：只在必要的位置嵌入尽量简洁的代码

可能形成  
性能瓶颈  
的代码

可能诱发  
性能问题  
的服务

必要  
的位置



# 对应关系

关注点		名称
Http Request 性能	扩展	Rinit/Rshutdown
函数性能	内核	zend_execute
服务质量	内核/ 扩展	zend_error/ SAPI. http_response_code/ Mysql_query result/ Curl result/...
应用拓扑	扩展	Curl/Mysql/Memcached/ Redis/Thrift/SOAP/...
跨应用追踪	扩展	Curl/SOAP/...

# 与常规性能工具的区别

# PHP-FPM SlowLog

```
1  <?php
2
3  sleep(9);
4
5  $t = time();
6  while( $t + 3 > time() )
7  {
8      $x = strlen("abc");
9  }
10
11  ?>
```



# PHP-FPM SlowLog

## php-fpm.conf 配置

```
request_slowlog_timeout = 10
```

- PHP将执行时间超过10秒的脚本记录到慢日志文件
- 同时记录到日志中的还有在第10秒正在运行的函数

## 运行结果

```
[root@localhost usr]# tail -f /usr/local/log/www.log.slow  
[06-Mar-2015 17:40:37] [pool www] pid 27930  
script_filename = /var/www/html/slow.php  
[0x00002b118e098188] strlen() /var/www/html/slow.php:8
```

# APM慢过程跟踪效果

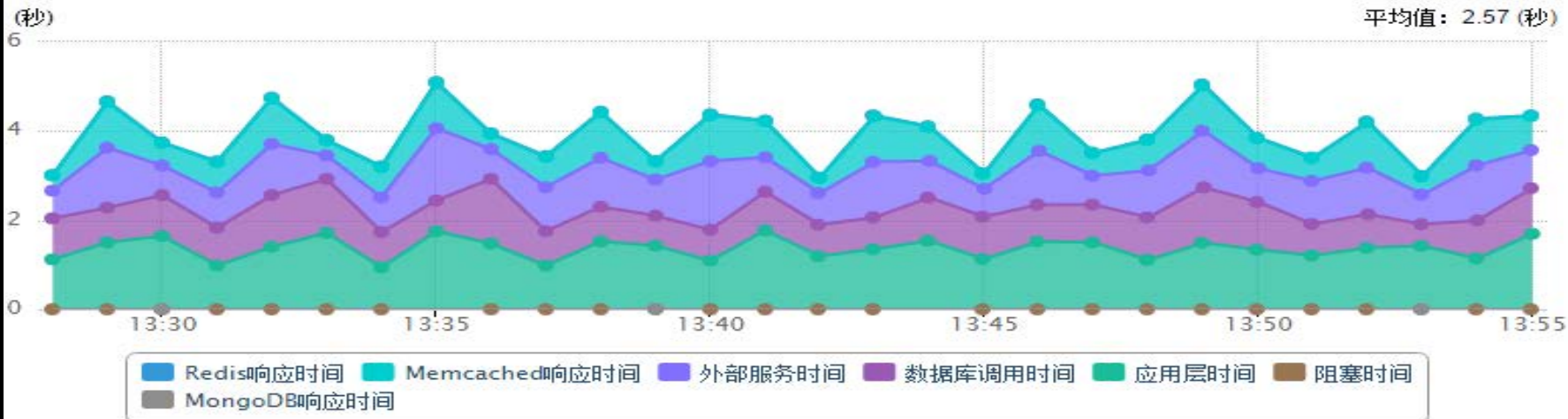
分类	持续时间(ms)	时间占比(%)
▼ PHP.execute	12024	100.00
▼ sleep	9002	75.00

自动采集实际性能占比最高的函数

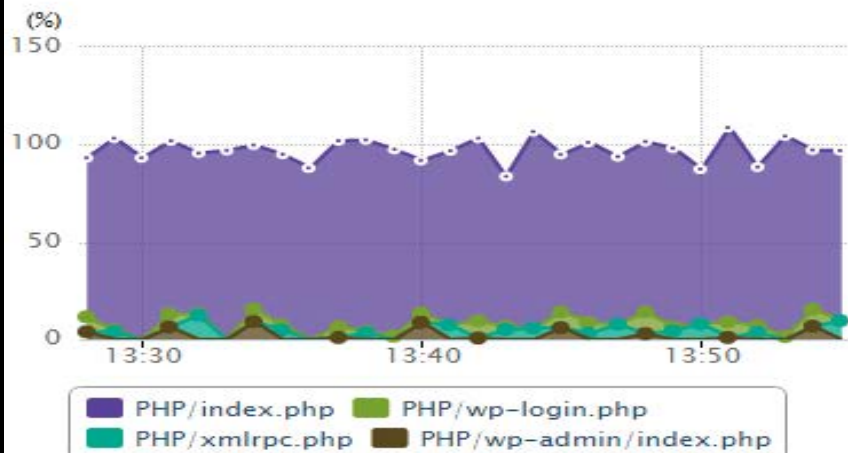
	xhprof	听云APM
嵌码方式	手工嵌入代码或 auto_prepend_file	无需开发人员参与
异常追踪	仅采集代码的性能	除性能信息还包括执行异常，sql错误等
函数归类	无	自动归纳为数据库、memcache、redis、web-service等多个分类
数据展现	手工录入run id，或二次开发汇总报表	各种性能报表和汇总数据：堆叠图、曲线图、邮件警告、手机短信报警

# APM图表

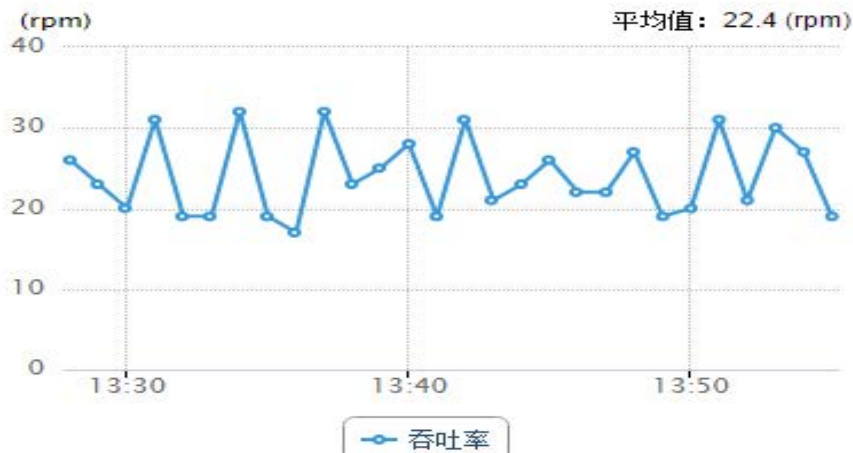
## ? 应用服务器响应时间



## ? 最耗时Web应用过程 (Web Action) 图表



## ? 吞吐率

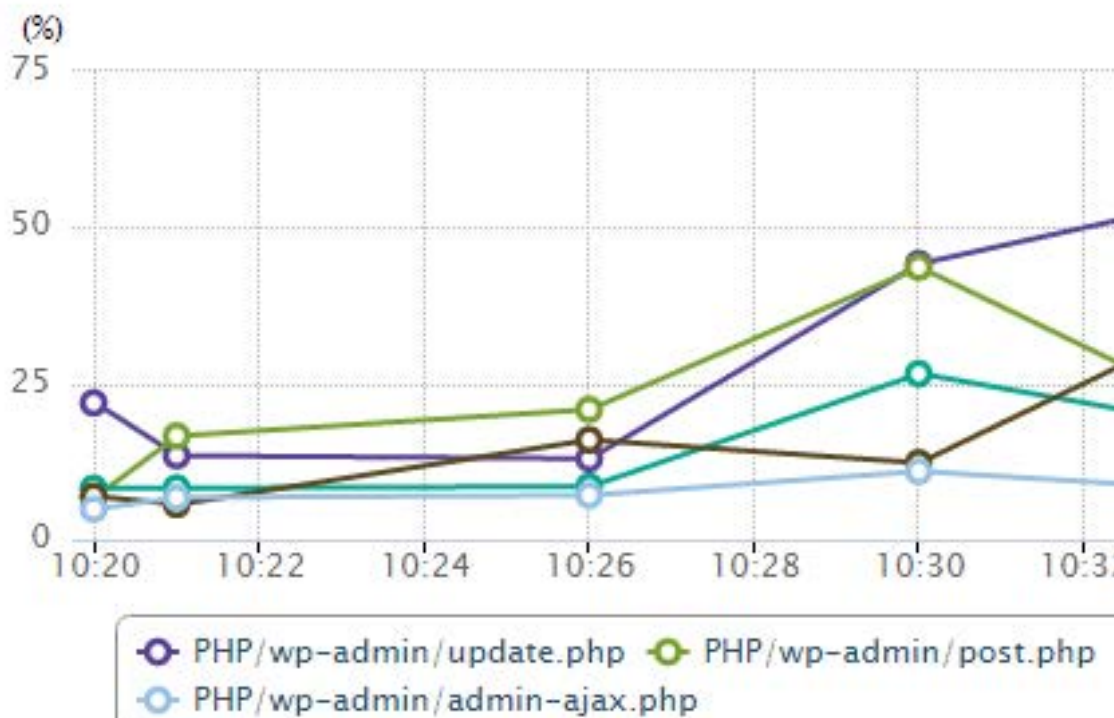


# 应用过程性能

## Web应用过程一览

PHP/wp-admin/update.php	18.41%
PHP/wp-admin/post.php	13.34%
PHP/wp-admin/plugin-install.php	9.57%
PHP/wp-admin/load-scripts.php	9.26%
PHP/wp-admin/admin-ajax.php	6.33%
PHP/wp-admin/edit.php	6.21%
PHP/wp-admin/index.php	5.88%
PHP/wp-admin/post-new.php	5.41%
PHP/wp-comments-post.php	5.07%
PHP/wp-cron.php	4.63%
PHP/wp-admin/load-styles.php	4.59%
PHP/wp-login.php	4.54%

### ? TOP5 最耗时Web应用过程(墙钟时间比)曲线图



# 慢过程追踪列表

## ? 慢应用过程追踪列表

Web应用过程:

查询

时间	Web应用过程	服务器响应时间(ms)
2015-03-09 12:11	PHP/wp-cron.php	6479
2015-03-09 11:04	PHP/index.php	6200
2015-03-09 13:12	PHP/index.php	5998

# 慢过程追踪详情

分类	持续时间(ms)	时间占比(%)	时间偏移量(ms)
▼ PHP.execute	6479	100.00	0
▶ require	67	1.03	0
▼ show_loginbar_new	6362	98.19	68
▼ getNewHeadHtml_2014	6362	98.19	68
▼ NewUC.get_system_msg_unread_count	6279	96.91	91
▼ MyDB.readone	6081	93.86	102
▼ DB_MySQL.query	6081	93.86	102
mysql_db_query	6081	93.86	102
<b>SQL</b> <span>⊗</span>			
SELECT count(*) as total FROM system_msg Where to_uid='?????????' and status<>? and addtime>???????????? and addtime			
▼ MyDB.readone	187	2.89	6183
▼ DB_MySQL.query	187	2.89	6183
mysql_db_query	187	2.89	6183
▶ get_vip_score	39	0.60	6370
▶ my_session_write	49	0.76	6430



# 跨应用分析

## 应用过程慢追踪

应用: [fvt\\_php5.6\\_zts](#)

应用过程: [URI/wordpress/index.php](#)

追踪时间: 2016-05-09 11:28:19

服务器响应时间: 3.745 (s)

实例信息: PHP:php5.6zts

摘要	追踪详情	相关SQL	拓扑图
<div style="display: flex; justify-content: space-between;"> <span>展开所有</span> <span>全部关闭</span> </div>			
分类			
▼ PHP.execute			
▼ require			
▼ require_once			
▼ preg_replace_callback			
▼ do_shortcode_tag			
▼ call_user_func			
▼ php_include			
▼ include			
▼ curl_exec			

## 应用过程慢追踪

应用: [fvt\\_php5.4\\_test](#)

应用过程: [URI/ob.php](#)

追踪时间: 2016-05-09 11:28:19

服务器响应时间: 3.013 (s)

实例信息: PHP:php5.4zts

摘要	追踪详情	相关SQL
<div style="display: flex; justify-content: space-between;"> <span>展开所有</span> <span>全部关闭</span> </div>		
分类		
▼ PHP.execute		
▼ sleep		

分类	持续时间(ms)	时间占比(%)	时间偏移量(ms)
▼ PHP.execute	3013	100.00	0
sleep	3001	99.60	3

分类	持续时间(ms)	时间占比(%)	时间偏移量(ms)
▼ php_include	3023	80.72	546
▼ include	3022	80.69	547
curl_exec	3022	80.69	547

外部应用: [fvt\\_php5.4\\_test](#) 实例信息: [php5.4zts](#)

Web应用过程: [WebAction/URI/ob.php](#)

URL: <http://192.168.2.107/ob.php>



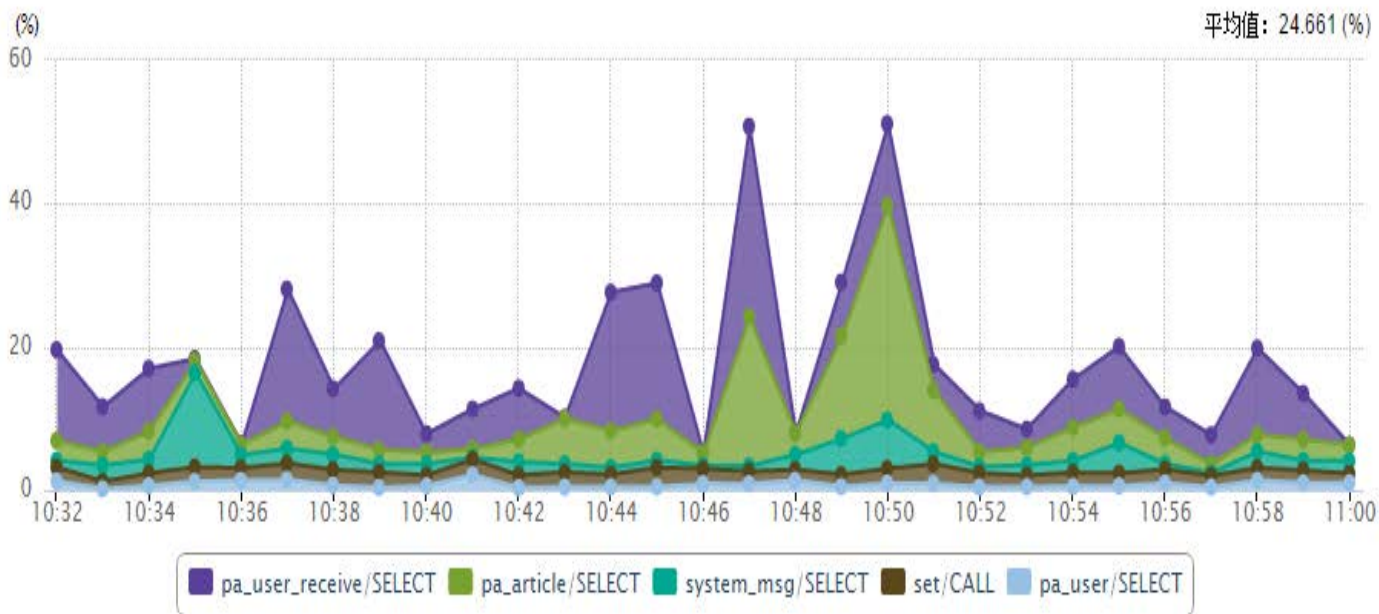
应用层时间 阻塞时间

# 数据库性能汇总

## 数据库一览

pa_user_receive/SELECT	133.011秒
pa_article/SELECT	88.803秒
system_msg/SELECT	38.485秒
set/CALL	32.524秒
pa_user/SELECT	14.382秒
pa_articletext/SELECT	12.671秒
pa_article_receive/SELECT	11.415秒
user_prize2_book/SELECT	10.147秒
bonus_count/SELECT	5.927秒
user_info/SELECT	5.674秒
pa_favorite/SELECT	5.543秒

? 最耗时SQL操作堆叠图



# 慢SQL列表

## ? 慢SQL追踪列表

SQL操作:

查询

SQL操作	追踪次数	平均执行时间(ms)
wp_options/UPDATE	21	1064
wp_options/DELETE	7	815
wp_options/SELECT	3	806
wp_options/INSERT	7	653
wp_posts/SELECT	1	501

## 完整SQL语句:

```
SELECT ID, post_name, post_parent, post_type FROM wp_posts WHERE post_name IN ('tingyun_cdn') AND (post_type = 'page' OR post_type = 'attachment')
```

## Trace详情

### 分类

mysql_query	(/opt/wwwroot/www.tingyun.com/wp-includes/wp-db.php:1213)
wpdb.query	(/opt/wwwroot/www.tingyun.com/wp-includes/wp-db.php:1538)
wpdb.get_results	(/opt/wwwroot/www.tingyun.com/wp-includes/post.php:3508)
get_page_by_path	(/opt/wwwroot/www.tingyun.com/wp-includes/class-wp.php:209)
WP.parse_request	(/opt/wwwroot/www.tingyun.com/wp-includes/class-wp.php:603)
WP.main	(/opt/wwwroot/www.tingyun.com/wp-includes/functions.php:808)
wp	(/opt/wwwroot/www.tingyun.com/wp-blog-header.php:14)
require	(/opt/wwwroot/www.tingyun.com/index.php:17)

## 外部应用一览

全部 自身服务 第三方

▶ cn.engadget.com 99.21%

来源: 自身服务

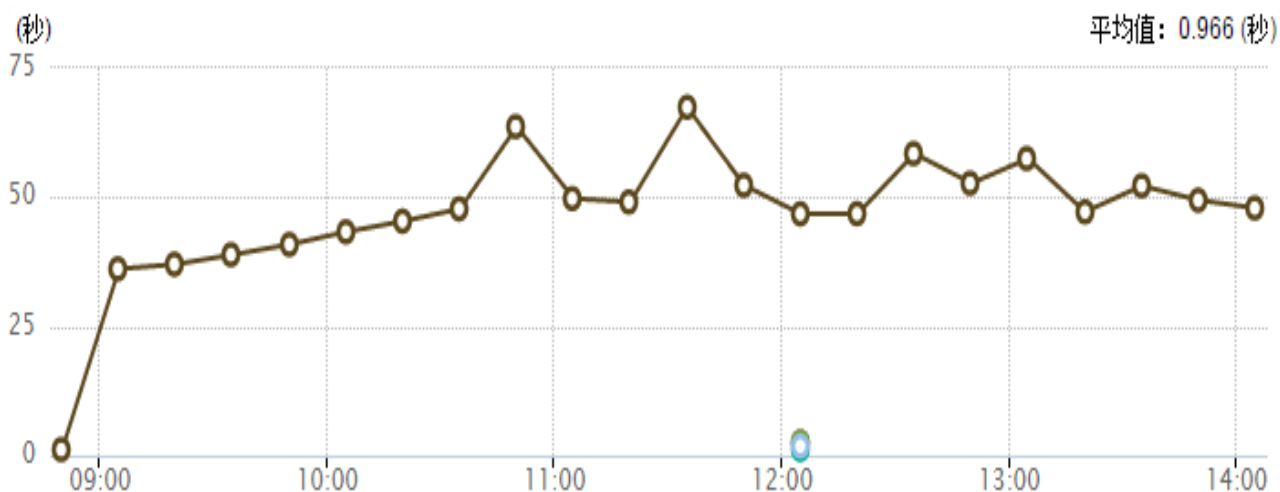
▶ api.wordpress.org 0.61%

来源: 自身服务

▶ 192.168.2.30 0.18%

来源: 自身服务

### ? Top5总耗时最慢的外部应用曲线图

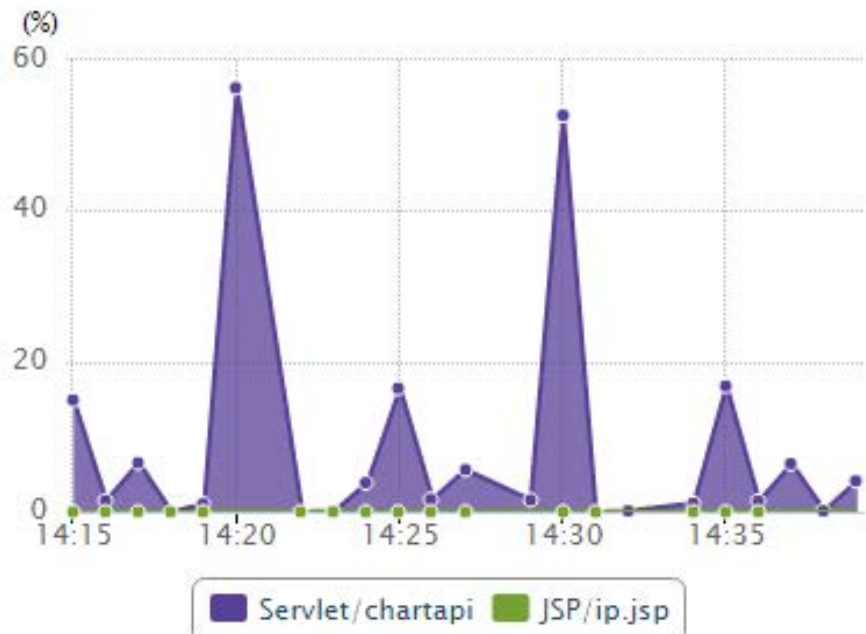


- api.wordpress.org/plugins/update-check/1.1/ (https)
- api.wordpress.org/themes/update-check/1.1/ (https)
- api.wordpress.org/core/version-check/1.7/ (https)
- cn.engadget.com/\* .xml
- 192.168.2.30/wp-cron.php

? 错误率



? 最耗时Web应用过程 (Web Action) 图表



# 错误列表

## ? 错误列表

开始出现时间	最后发生时间	持续时间	应用过程/错误类型	错误信息	统计次数
03-04 10:39	03-04 11:04	25分钟	PHP/web/user.php ERROR	Uncaught exception 'Exception' ...	3
03-04 10:38	03-04 10:54	16分钟	PHP/web/book.php ERROR	Uncaught exception 'Exception' ...	2
03-04 10:46	03-04 10:46	小于1分钟	PHP/web/store.php ERROR	Uncaught exception 'Exception' ...	1

# 错误详情1

## 错误信息

```
Uncaught exception 'Exception' with message '<br><b>Database error:</b> pconnect(121.18.211.81:3306, read, $DBPassword) failed.<br><b>MySQL Error</b>: 0 ()<br>' in /var/www/opendata/lib/db/db_mysql.php:322 Stack trace: #0 /var/www/opendata/lib/db/db_mysql.php(311): DB_MySQL->haltmsg('pconnect(121.18...') #1 /var/www/opendata/lib/db/db_mysql.php(108): DB_MySQL->halt('pconnect(121.18...') #2 /var/www/opendata/web/helper/class/MyDB.class.php(196): DB_MySQL->connect() #3 /var/www/opendata/web/helper/class/MyDB.class.php(116): MyDB->_getConnection(Object(DB_MySQL), Array) #4 /var/www/opendata/model/Mobile.class.php(204): MyDB->readone('select addtime ...') #5 /var/www/opendata/web/control/CBook.class.php(231): Mobile->get_app_old_start_log_time('60.173.10.108') #6 /var/www/opendata/web/helper/class/CAction.class.php(74): CBook->on_one_book() #7 /var/www/opendata/web/book.php(16): CAction->do_action('one_book') #8 {main} thrown
```

连接数据库 121.18.211.81:3306 失败



# 错误详情2

## 错误信息

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near " at line 1.

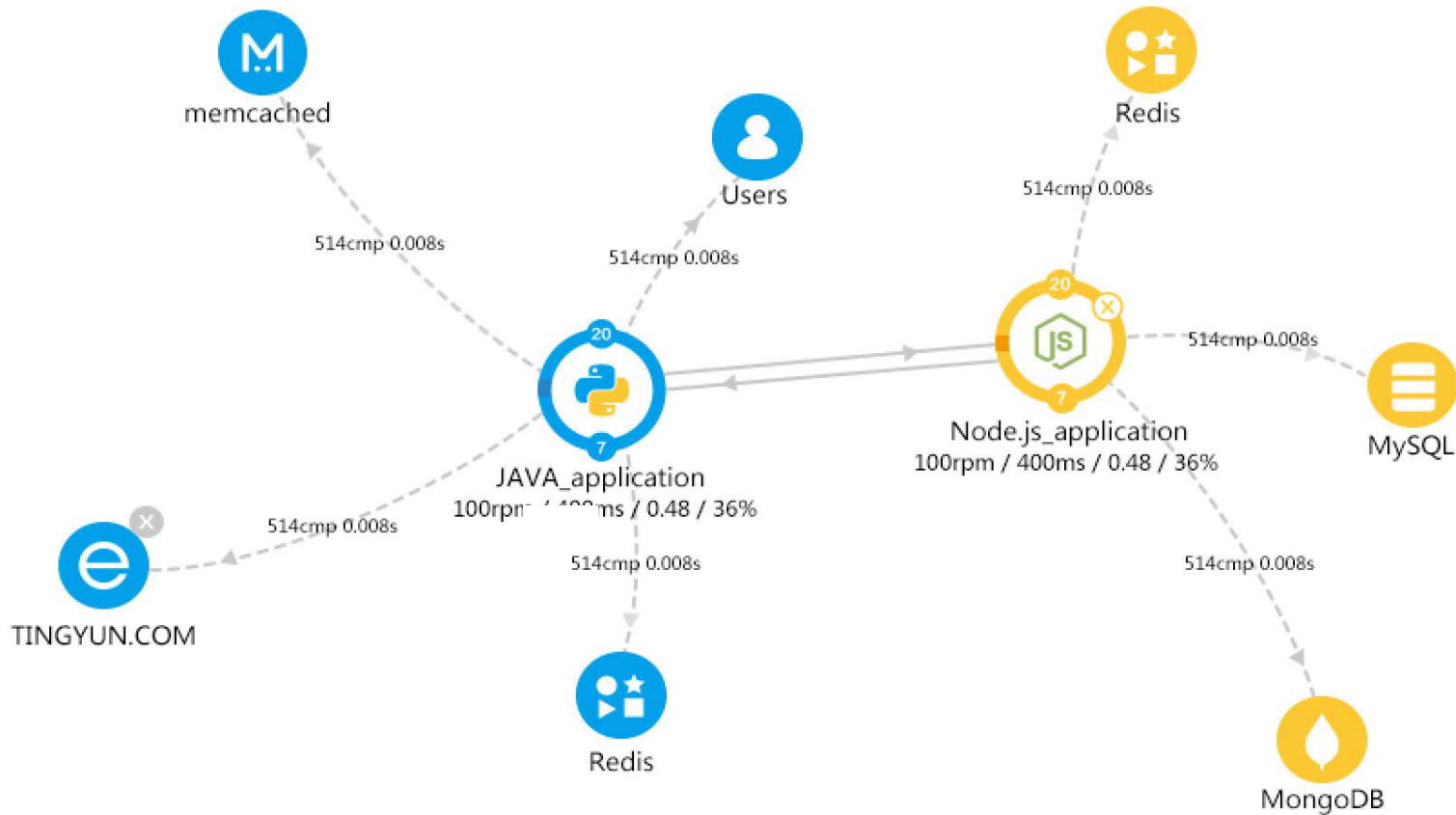
```
sql=[select * from 'diyoun_borrow' where borrow_nid=]
```

SQL语法错误

where条件缺少borrow\_nid数值

# 应用拓扑

正常 警报 严重警报 无数据 图例



# 案例实践

-某电商重大故障线索的快速发现

# 故障情况

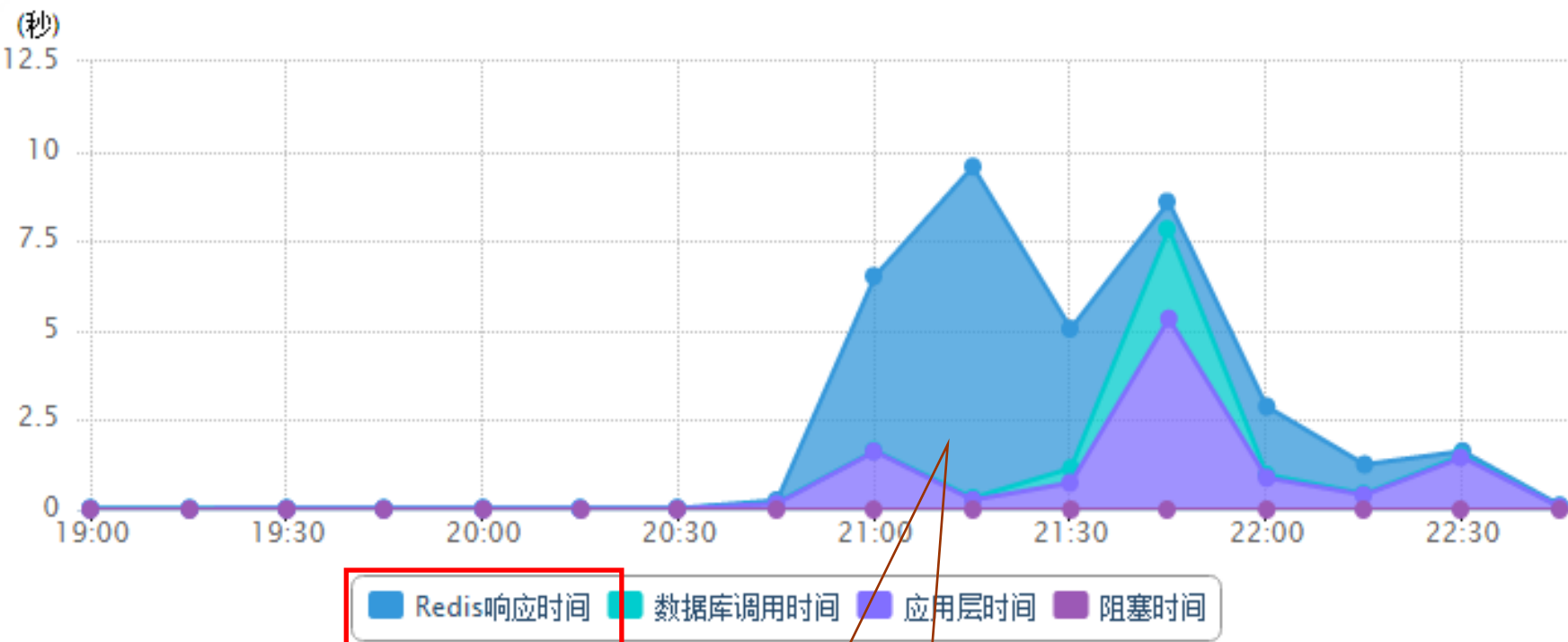
- 某电商某一产品线在2016年3月大促期间出现应用性能缓慢的问题。
- 架构情况：所有应用均负载均衡到几台服务器，这几台服务器每台上面均有所有应用的代码，每个应用依赖多个服务。
- 由于业务逻辑关系复杂，存在多个应用相互调用的情况，无法直接定位问题出现在哪个应用的哪个接口上。

# 分析过程 - 应用性能分解

ckapi\_50\_176



应用服务器响应时间



TINGYUN

瓶颈：Redis

ckapi\_50\_176

开始: 2016-02-29 19:00

结束: 2016-02-29 23:00

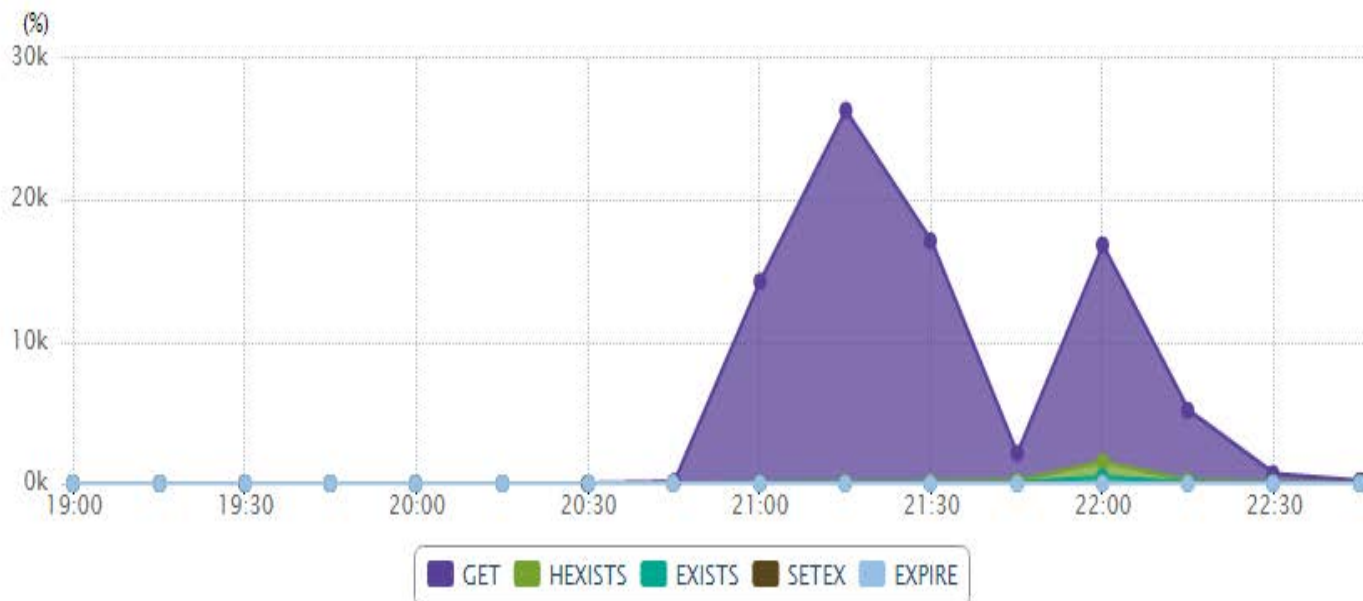
排序 操作总耗时 显示Top20

Redis

Redis—览

GET	725308.832秒
HEXISTS	14112.935秒
EXISTS	5720.131秒
SETEX	591.823秒
EXPIRE	18.124秒

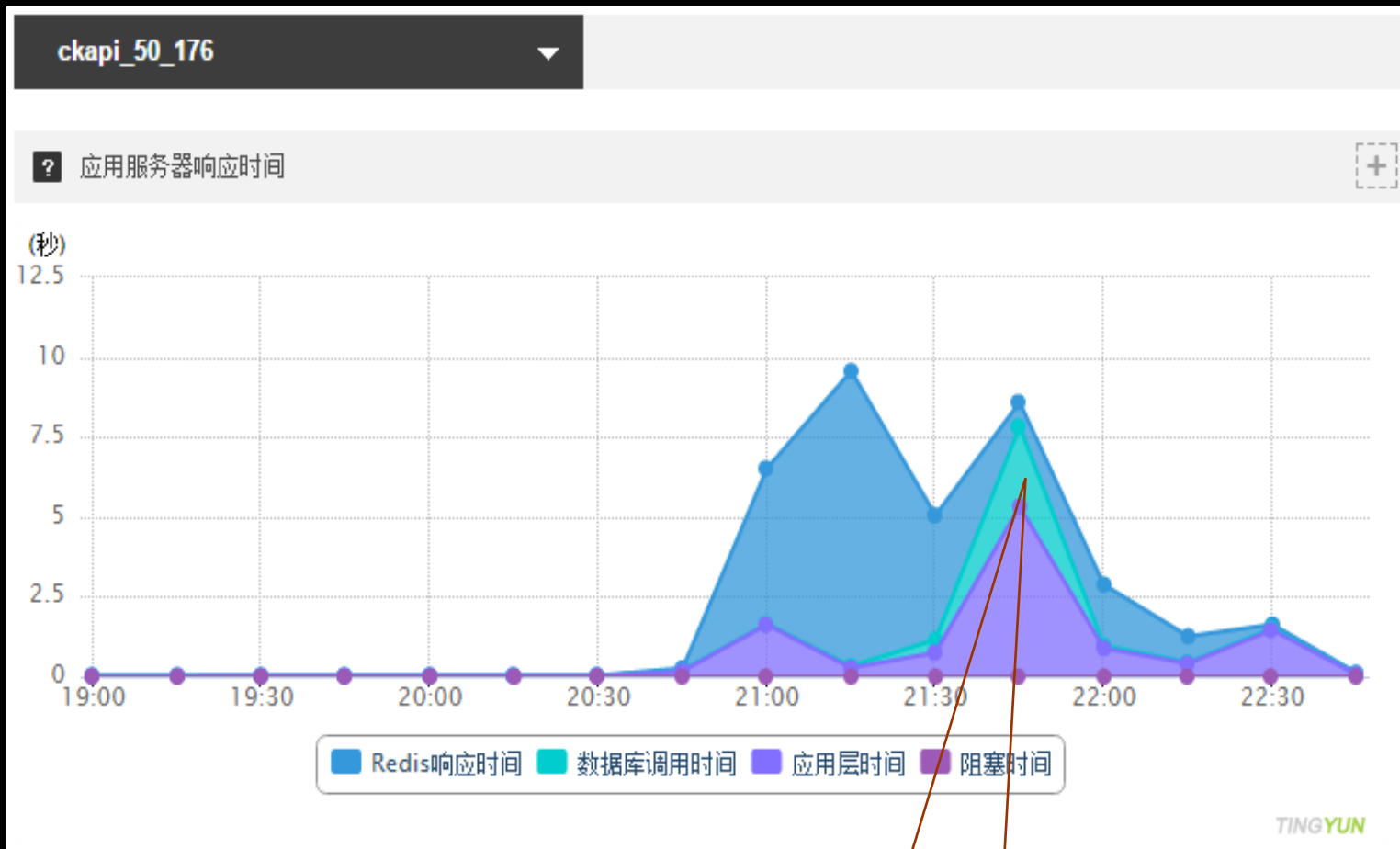
? 最耗时Redis操作堆叠图



# 处理过程

- 通过性能分解曲线，21：00左右发现是redis是性能瓶颈
- 查看慢过程跟踪堆栈、代码及相应配置发现是商品在redis集群内分布不均。虽然redis集群有十几台服务器，但大部分请求的商品集中在其中某一台Redis上，导致Redis负载过高，响应变慢。
- 客户快速调整商品在Redis集群内的分配策略，并做Redis设备调整。
- 经过大约半小时的调整，解决了Redis问题

# 分析过程 - 应用性能分解



瓶颈：数据库



▼ M_coupon.get_coupon_supplier_pro	☰	300031	91.69	3885
▼ get_mia_read	☰	300020	91.69	3885
▼ mia_db_pool	☰	300020	91.69	3885
▼ CI_Loader.database	☰	300018	91.69	3887
▼ DB	☰	300018	91.69	3887
▼ CI_DB_driver.initialize	☰	300018	91.69	3887
▼ CI_DB_mysqli_driver.db_connect	☰	300016	91.69	3887
mysqli_connect	☰	300016	91.69	3887

### StackTrace

mysqli_connect	(/opt/webroot/cksr/system/database/drivers/mysqli/mysqli_driver.php:72)
----------------	---

数据库连接超时

# 总结

- 快速缩小问题范围
- 优化redis商品分布策略
- 优化数据库策略



听云免费测试账号申请交流群