

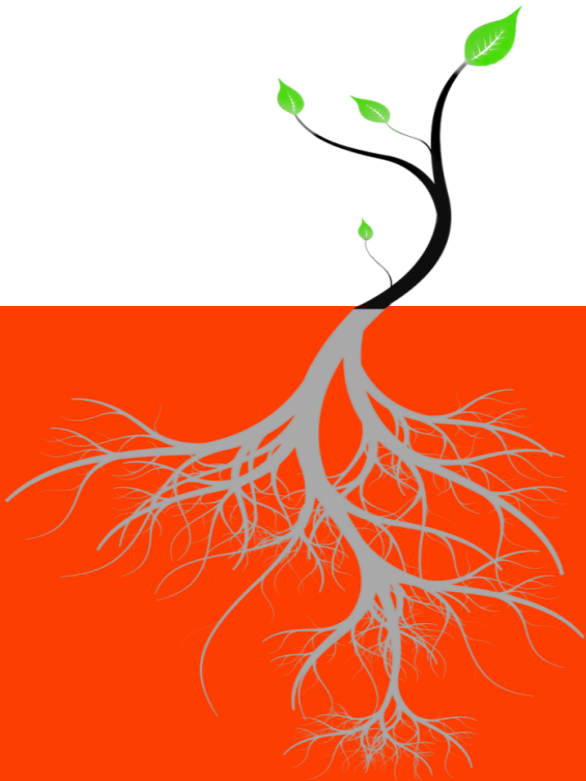


PHPCon 2015

北京站

用PHP在SAE创业

丛磊 @kobe



CONTENTS

- 01 SAE介绍以及发展历史
- 02 PHP Runtime的介绍
- 03 使用SAE PHP Runtime的优势
- 04 小技巧分享
- 05 典型的案例分享



SAE的发展历史



SAE发展历程



视频介绍



2

SAE PHP Runtime介绍



支持的PHP版本

php 5.6.8

php 5.3.29



运行环境的一些限制

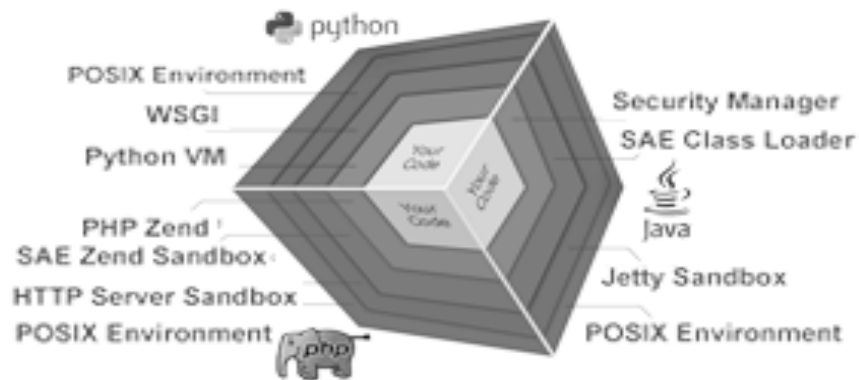
- ↻ 变相本地读写
- ↻ 不允许绑定端口
- ↻ 限制了一些函数和类
- ↻ `ini_set`使用白名单机制



应用之间的隔离

☞ 代码层面的隔离

- ✓ 第一层：PHP Sandbox
- ✓ 第二层：Linux Container
- ✓ 第三层：Apache Sandbox



☞ 网络层面的隔离

- ✓ 转发L7到FetchURL
- ✓ 转发L4到SocketProxy



代码的迁移



百分百的兼容性？



代码的迁移

- ✧ 使用stream wrapper代理本地的文件读写操作
saekv://
saemc://
saestor://
- ✧ 上传的文件需要存储到storage或者云存储中
- ✧ 使用到Memcache的部分只用把连接的ip:port任意指定即可
- ✧ 留意应用的错误日志是否有warning，解决warning

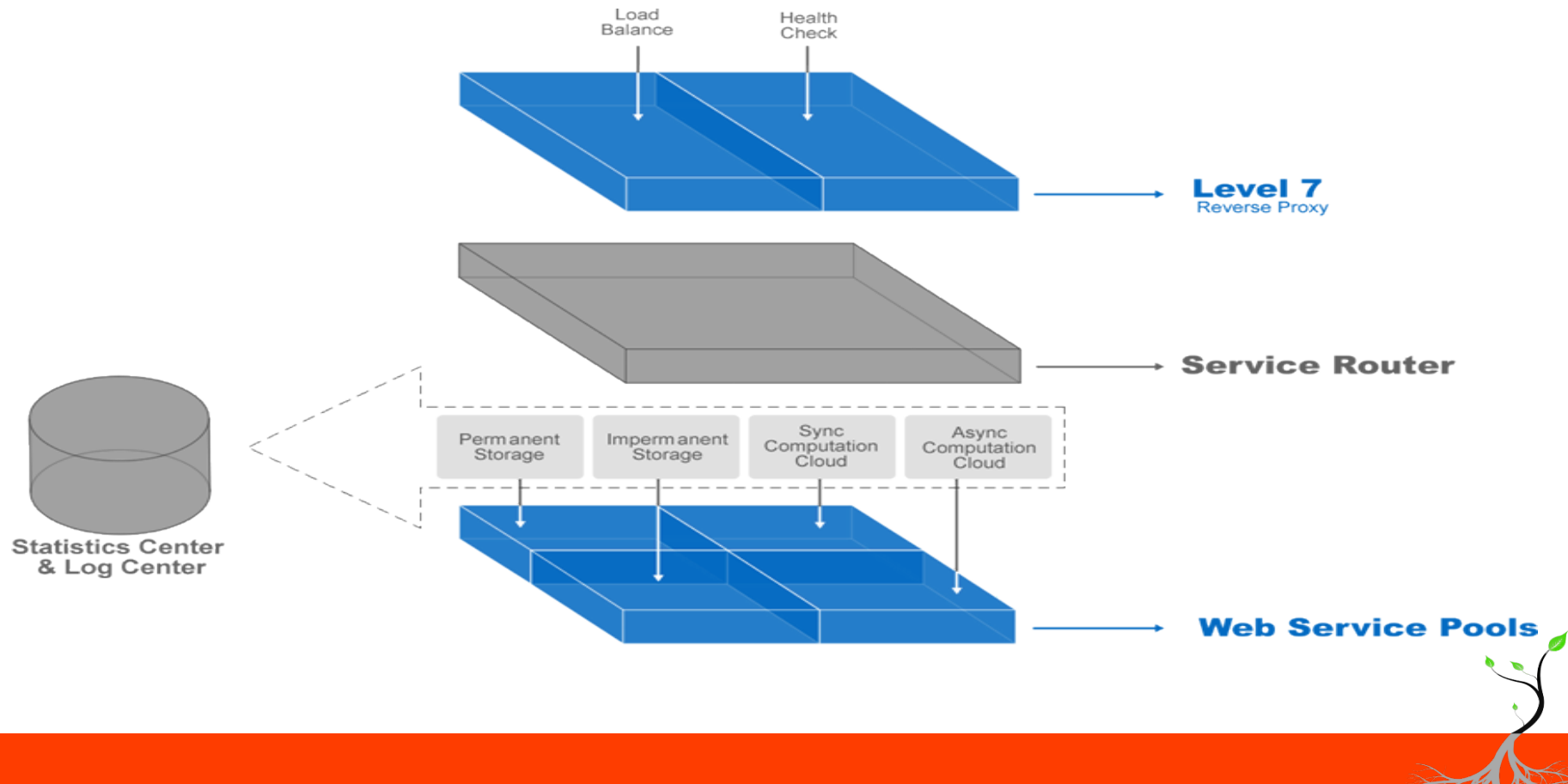




SAE PHP Runtime优势

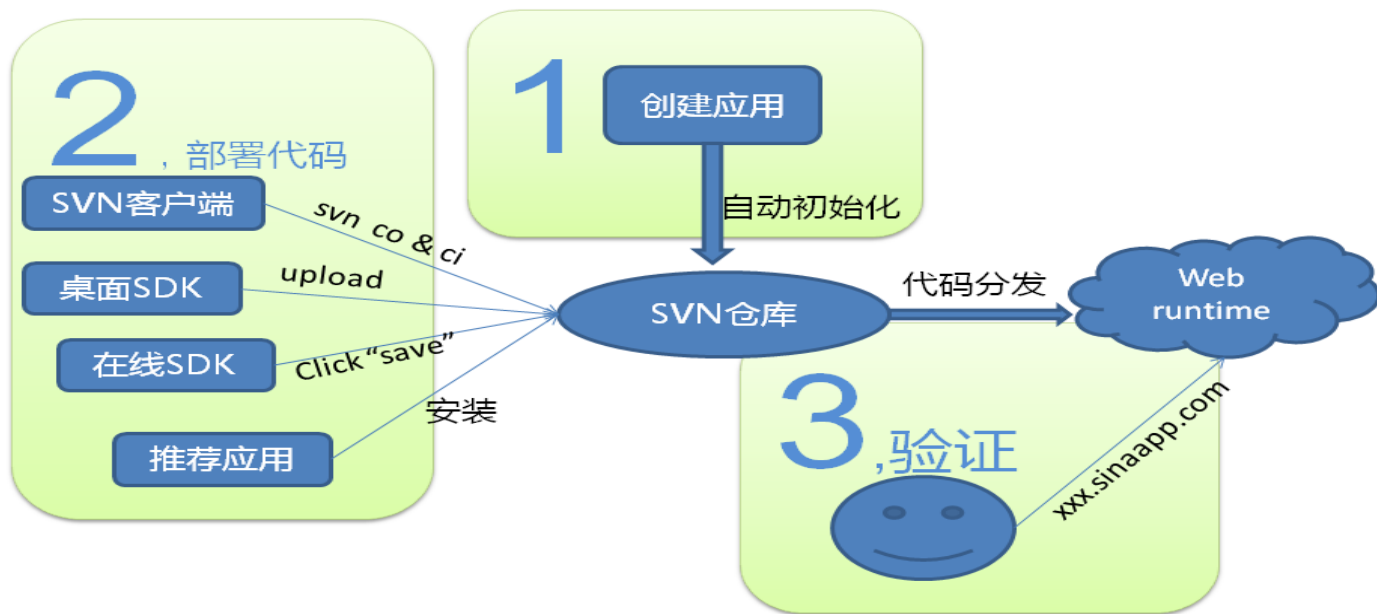


天生的分布式



天生的代码管理系统

SAE代码部署原理图



天生的多人协作

管理员

参与者

观察者

自定义

灵活分配团队成员 管理代码、管理服务、设置配额、设置应用信息权限



自带调优工具

Report

Run #1426761913:

Click a function name below to drill down.

[View Top Level Run Re](#)

Overall Summary
Total Incl. Wall Time 2,983
(microsec): microsecs
Number of Function Calls: 237

[\[View Full Callgraph\]](#)

Displaying top 100 functions: Sorted by Incl. Wall Time (microsec) [\[display all \]](#)

Function Name	Calls	Calls%	Incl. Wall Time (microsec)	IWall%	Excl. Wall Time (microsec)
main()	1	0.4%	2,983	100.0%	8
run_init::lp/lp_init.php	1	0.4%	1,938	65.0%	11
radioController::construct	1	0.4%	1,277	42.8%	
appController::construct	1	0.4%	1,273	42.7%	3
file_exists	8	3.4%	865	29.0%	86
session_start	1	0.4%	834	28.0%	83
SaeIswaf::init	1	0.4%	790	26.5%	6

- ✓ 管理后台一键开启
- ✓ 由函数sae_xhprof_start()和sae_xhprof_end()自行控制调试开关



主流的PHP开发框架都有SAE版本



yar、yaf支持

yar

yar support	enabled
Version	1.2.3

yaf

yaf support	enabled
Version	2.2.9
Supports	http://pecl.php.net/package/yaf

- ✎ 基于yaf可以在SAE上开发高性能应用
- ✎ 基于yar可以基于SAE提供RPC服务



支持定时任务、异步任务

- ☞ SAE提供粒度最低到分钟的分布式定时任务
- ☞ Cron的执行是以HTTP方式触发的，触发后真正执行的是用户在应用的HTTP的回调函数
- ☞ Cron服务是分布式环境部署的，具有高可靠性，多点之间相互隔离且同时触发，并且通过分布式锁进行选举并最终由一个健康节点执行
- ☞ SAE也提供分布式任务队列服务，用于执行异步的HTTP调用方式的任務
- ☞ 提供阻塞度长的顺序队列和并发度高的并发队列



长时间的cron

max_execution_time不够用怎么办？

- ↻ SAE将Cron和WebRuntime进行分离
- ↻ 允许执行cron的脚本最长运行30分钟



该有的服务都有

共享型MySQL

独享型MySQL

KVDB

Memcache

Storage

全文检索

计数器



支持更多的个性化服务

邮件群发服务

地理位置服务

语音服务

短信服务

苹果、安卓推送

应用体检

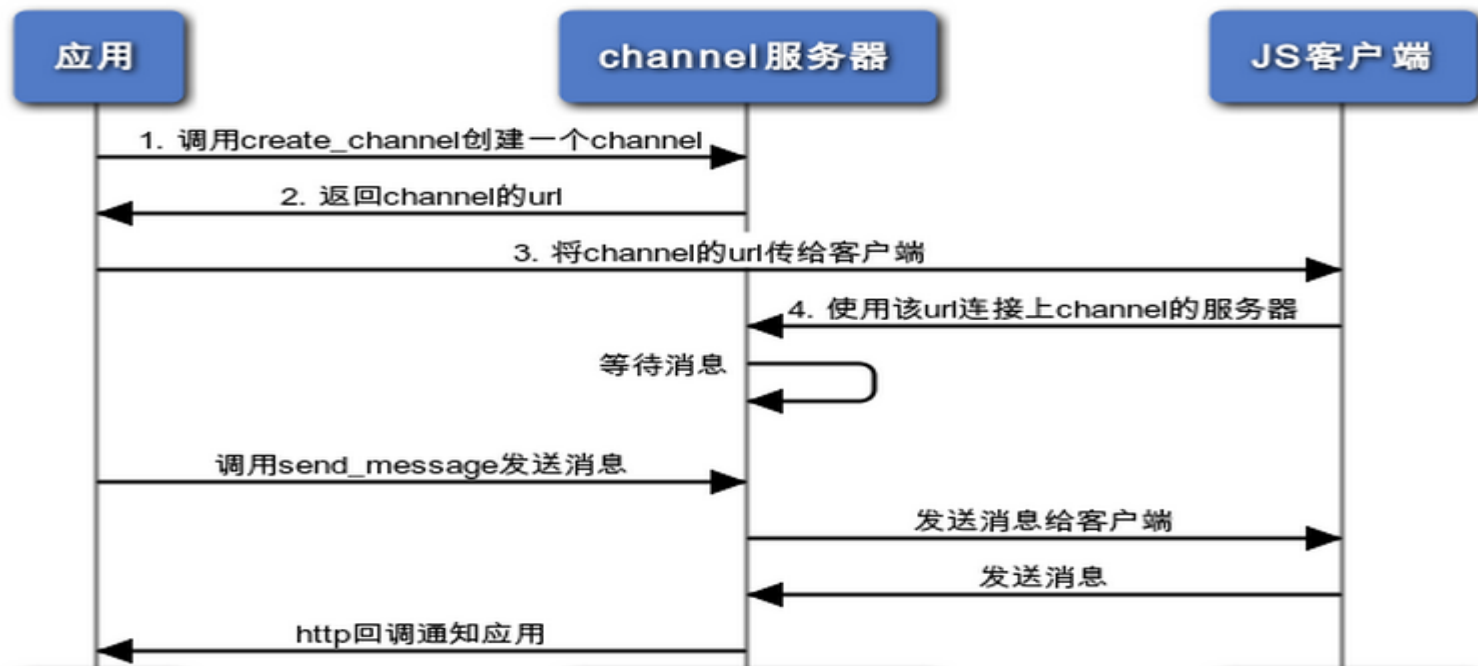
数据恢复服务

数据库备份服务

... ..



WebSocket支持



Channel是SAE提供的实时消息推送服务。通过在浏览器和SAE服务端之间建立长连接，使得应用可以方便的向JS客户端实时的推送消息



紧随PHP社区版本更新，无需关心底层安全问题

- ✧ SAE本身提供多层次的底层沙箱机制，保证应用的安全
- ✧ 紧随PHP官方的版本，及时更新小版本



应用防火墙

应用防火墙是运行在SAE负载均衡处的过滤模块，实现对恶意抓取的行为有效拦截

- ☞ 黑白名单机制
- ☞ 频率/流量限制机制
- ☞ 访问速度限制机制



低成本的使用CDN服务

- 一键开启
- 实现对网站中的静态资源和云存储中的静态文件的加速
- 只需按正常的方式进行部署，即可将资源部署至CDN源站，然后SAE的CDN服务会自动将资源同步至各个节点，随后您的用户在访问资源时，将就近获取，从而获得更好的访问体验
- 通过CDN服务产生的流量相对较便宜，会为您节约30%左右的流量费



4

小技巧分享



PHP完成密集型计算

PHP的慢是相对于**C/C++**级别的语言来说，事实上，**PHP**语言最初的设计，就不是用来解决计算密集型的应用场景。我们可以这样粗略理解为，**PHP**为了提升开发效率，而牺牲了执行效率。

?

那如何利用PHP完成密集型的计算呢，例如“撞库”？

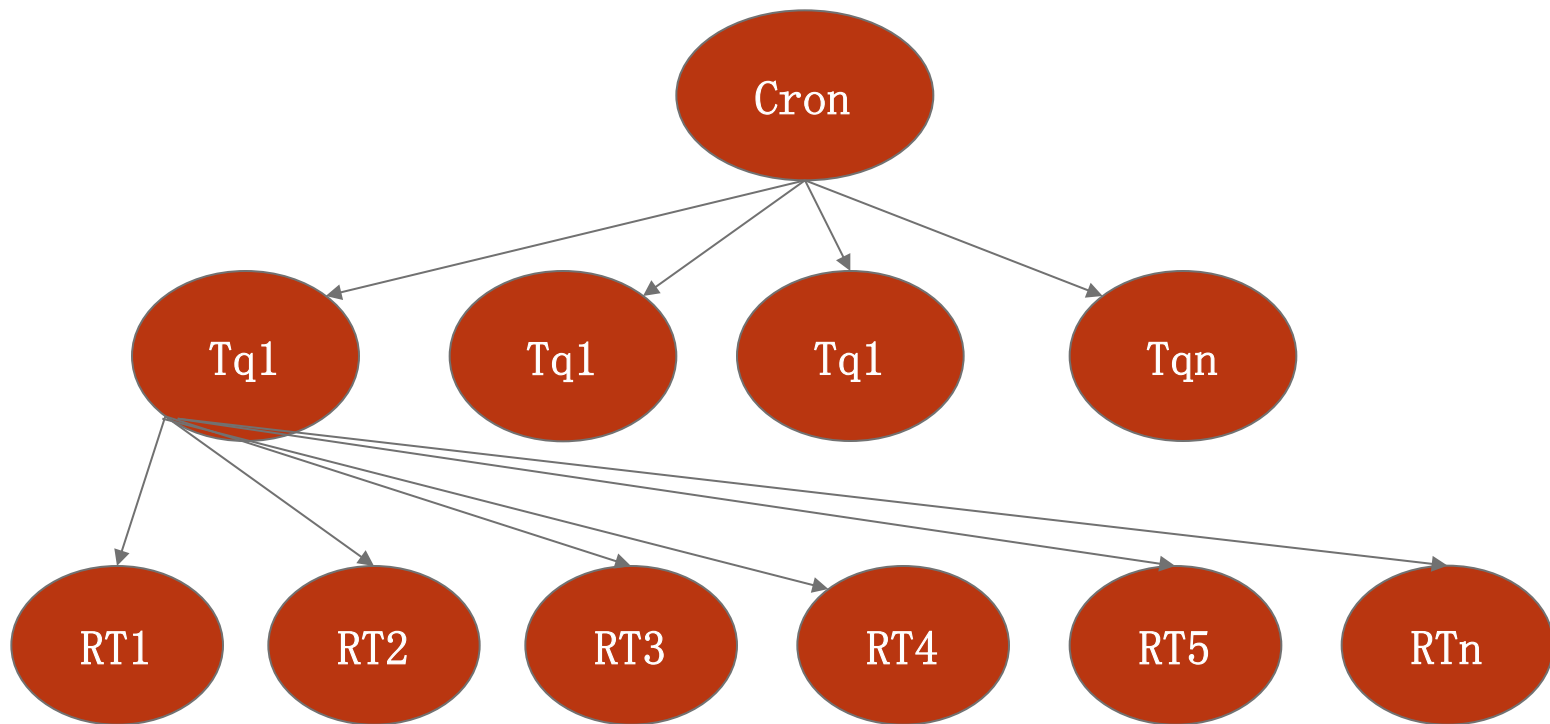


PHP实现密集型计算

- ✧ 写PHP脚本，以进程的方式运行
 - ✧ 在多台机器上都运行这个脚本
 - ✧ 实现一种机制防止竞争发生，避免计算资源浪费
- 存在一些缺点
- 这种调度机制不是很容易部署
 - 很难监控某台机器的某一个进程还在不在
 - 不具有通用性，换一种计算模型还得重头来过



在SAE由PHP实现密集型的计算



在SAE由PHP实现密集型的计算

- ✓ 本身实现是无锁的，任务的分配由Cron完成，插入到按任务hash的TaskQueue中
- ✓ TaskQueue并发度由TaskQueue的总数和单个TaskQueue的并发度乘积得出
- ✓ 计算节点无状态，执行结果可以更新到NoSQL中或者分表存储可以避免压力



PHP的并行

☞ 一个处理要调用几十个接口才能完成，页面加载很慢，如何优化？

可能的解决思路

- ✓ 尽可能的优化接口提高响应时间
- ✓ 合并可以合并的请求？
- ✓ 借助curl的multi簇函数并发？



SAE上划串行为并行的方式

```
1  <?php
2  $all_task = array(
3      'a' => 'http://www.baidu.com',
4      'b' => 'http://weibo.com',
5      'c' => 'http://www.sohu.com',
6  );
7  $taskqueue = new SaeTaskQueue('test');
8  $push = array();
9  foreach ($all_task as $key => $value) {
10     $post_data = sprintf("url=%s", $value);
11     $push[] = array('url'=>"/fetch.php", "postdata"=>$post_data);
12 }
13 $queue->addTask($array);
14 $ret = $queue->push();
15 $mc_instance = memcache_init();
16 if ($ret === false) {
17     // 失败时的处理
18 } else {
19     while (1) {
20         // 循环等待
21         foreach ($all_task as $key => $value) {
22             $ret &= memcache_get($mc_instance, $key);
23         }
24         if ($ret) {
25             // 任务完成
26             break;
27         }
28     }
29 }
```



SAE上划串行为并行的方式

- ↪ 如果接口之间有依赖关系？那也只好分组拆了
- ↪ 并发度还不够？异步的队列一样可以hash，按之前的说法，并发度是叠加的
- ↪ 有没有实际的案例？SAE本身之前有个业务是运行在自身的Runtime上的，通过此方式成功提速10倍



常规的快速上线和回滚方式

- ✧ 通过RPM包的方式？给新包和回滚备份包，出问题时立马用旧的包覆盖
- ✧ 通过git或者svn hook的方式？用代码控制器的方式回滚
- ✧ 其他方式？

存在的问题？

- 还是有个操作的时间，有时太紧急会出乱子
- 造一个测试的环境可能成本巨大，特别是系统的组件很多很多的时候



SAE上实现快速上线和回滚

- SAE提供基于版本发布的机制，可以任意选择一个数字版本当线上的版本，默认的应用名 `a.sinaapp.com` 或者绑定的域名会访问默认的版本。
- 可以选择任何其他的版本当开发版本，共享数据库、缓存等等任何服务数据。
- 仍然想用测试数据库？可以创建另外的应用把数据库授权给这个应用
- 独立域名绑定到非默认版本？SAE提供一种方式绑定域名到非默认版本，通过 `sae-cname-1.{应用名}.sinaapp.com` 就可以访问版本1
- 通过管理面板一键切换默认版本就可以实现快速上线和回滚。



5

使用案例分享



国家地震局

项目	描述
访问量级	跟地震波动很大
使用服务	MySQL、KVDB、Memcache、存储等



某广告平台

项目	描述
访问量级	超过1亿PV每天
使用服务	MySQL、KVDB、Memcache、CDN、存储、应用防火墙等
每天花费	约74人民币



掌上贵金属

项目	描述
访问量级	单个应用每天约5000万PV
使用服务	MySQL、KVDB、Memcache、CDN、存储、应用防火墙等
每天花费	约80人民币



Q & A

提问答疑

