

PHP在金融股票项目中的应用

超群@phpconchina

2016



关于我

- 早年在新浪音乐
- 原腾讯财经后台技术负责人
- 现在和几个朋友run一个股票相关startup
- 量化投资学徒

财经类PHP应用

- 腾讯财经
- 新浪财经
- 凤凰财经
- 同花顺
- 雅虎财经
- 钱升钱
- ...

目录

- ◆ 数据准确性
- ◆ 解耦：一切皆接口
- ◆ 几个性能问题
- ◆ lxc容器部署
- ◆ 全技术栈

数据准确性

关于上海证券交易所行情指数成交金额数值显示有关事宜的通知

各市场参与者：

本所 2015 年 4 月 17 日新闻发布会公布了 FAST 行情切换相应安排，将分步以 FAST 行情、产品非交易基础信息文件替代现行的 SHOW2003 行情文件，其中改进之一即是解决 SHOW2003 行情文件指数成交金额字段仅支持 1 万亿以下数值的情况。

今日，上交所市场成交金额超过 1 万亿，现有 SHOW2003 行情文件中指数成交金额字段显示为全“9”数字，交易及个股行情不受影响，均正常显示。此为软件设置原因，并非技术故障。该字段数值以 FAST 行情文件的相应字段为准。在 FAST 行情文件正式启用并全面替代 SHOW2003 之前，出现类似情形均以 FAST 行情显示的数值为准。FAST 文件行情替换 SHOW2003 的具体时间将

数据准确性：类型

- 问题：PHP动态语言，弱类型

"0" == 0

"1000" == 1000

false == 0

null == 0

null == false

数据准确性：类型

- 解决方案：配置化的validation

```
protected $_rules = [  
    'id'      => ['type' => 'int', 'msg' => '用户ID必须为整数'],  
    'email'   => ['type' => 'email', 'range' => [5, 32]],  
    'mobile'  => ['func' => 'ValidateUtils::checkMobile'],  
    'name'    => ['type' => 'string', 'range' => [3, 16]],  
    'passwd'  => ['type' => 'string', 'min' => 6, 'required' => true],  
    'meta'    => ['rules' => [  
        'time'    => ['type' => 'float'],  
        'foo'     => ['regex' => '^\\d+$'],  
        'gender'  => ['in' => ['M', 'F']]  
    ]],  
    'status' => ['type' => 'int']  
];
```


数据准确性：浮点

```
<?php  
  
$f = 0.58;  
echo intval($f * 100);  
// output 57
```

- 原因：IEEE 754，mysql也有浮点问题→ decimal

数据准确性：浮点

- 解决方法：

1. 全部*1000000，bigint $-2^{63} \sim 2^{63}-1$
2. 字符串化，bcmath/gmp运算

沪深	环球	外汇
美元人民币	美元人民币离岸	美元指数
▲ 6.5828 +0.0088 +0.0013%	▲ 6.5983 +0.0183 +0.0028%	▲ 94.0838 +1.0044 +0.0108%
基本汇率		
英镑美元	1.4493	-0.0342%
欧元美元	1.1284	-0.0123%
新西兰元美元	0.7184	-0.0088%
美元瑞郎	0.9629	-0.0004%

数据准确性：校验

- 接口正确

`{"code": 0}`

- 格式正确

规则化validation

- 数据正确

对比监控

解耦：一切皆接口

解耦：一切皆接口

- 问题：
 - a. All in one , 既处理数据又对外服务
 - b. 直接读库/写库

解耦：一切皆接口

- 方案：
一切皆接口，包括所有后台服务，一个程序只干一件事情
- 好处：
 - a. 容灾/多数据源
 - b. 统一处理
 - c. 有损服务/降级服务
 - d. 便于重构

解耦：行情服务

1. 获取股票代码
2. 拉取行情并解析
3. 更新行情 (入库&pub)
4. 对外服务

几个性能问题

几个性能问题：缓存

- 高速缓存 VS 大量数据 → redis & ssdb
- 被动缓存 VS 主动缓存
- 特殊情况 dailyCache
- 正向缓存 VS 反向缓存



几个性能问题：行情

- 请求最频繁的接口

- 几种方案：

a. 专属自研 7w+ -> 3w+ -> 4.32亿

b. ngx+lua 5w+ -> 2.5w+ -> 3.60亿

c. php-fpm 15k+ -> 8k+ -> 1.15亿

d. swoole 3.5w+ -> 2w+ -> 2.88亿

几个性能问题：kql



问题：

一个功能页面需要请求N个不同数据接口

思考：

a. 服务器端根据产品需求合并接口

b. 客户端根据产品需求自组织接口

方案：

kql, 基于yar的自组织合并请求接口

几个性能问题：kql

```
/kql?q= { ... }
```

一个典型的请求：

```
{
  "foo": [
    "quote",
    {"symbols": ["sz000001"]}
  ],
  "bar": [
    "mline.d1",
    {"symbol": "sz000001"}
  ],
  "news": [
    "news.symbols",
    {"symbols": ["sz000001"], "mtime": 1465788897}
  ]
}
```

一个典型的响应：

```
{
  "foo": { ..., "mtime": 1465788897},
  "bar": { ..., "mtime": 1465758837}
}
```

几个性能问题：实时计算

- 问题：
 - a. 数据项多 b. 格式复杂 c. 数据量大 d. 计算指标多
- 需求：
 - a. 准实时 b. 计算离数据近 c. 方便，尽量少写代码
- 方案：
 - a. mysql b. spark c. aerospike d. tarantool e. postgre f. mongodb

几个性能问题：实时计算

- 排行榜：
涨幅榜/跌幅榜/换手率榜/资金流向榜单/....
- 汇总/均值：
aggregate/\$group/\$sum/\$avg/\$max/\$min
- map-reduce：
基于mongodb服务器端js编程
- 除复权
python -> php

lxc容器部署

lxc容器部署

- 问题：
模块多，依赖多，环境复杂
- 容器：
docker：进程容器
lxc：环境容器，docker他爹
- 方案：
lxc容器部署，git代码发布

lxc容器部署

- 复制：

```
lxc-clone -o ubuntu -n foobar
```

- 备份：

```
tar --numeric-owner -czf ./foobar.tgz /path/to/lxc/foobar
```

- 迁移：

```
tar --numeric-owner -xzf ./foobar.tgz
```

全技术栈

全技术栈

- nginx+php-fpm : web接入
- mysql : 线上服务数据(在线) , 5.7.x , 大量使用json字段
- mongodb : 计算用数据(离线) , 3.2.x , 数据仓库
- sphinx : 股票搜索
- python : 数据处理&机器学习
- redis : 高速缓存&pubsub
- ssdb : 普通缓存

寻人

地点：北京，人大附近

技能树：php/python/js/css

待遇：略高于原公司，不加班

团队：标配BAT

邮箱：fuchaoqun#gmail.com



看盘宝，安静好用的股票App