



GOPS2017
Shanghai



GOPS

全球运维大会

2017

上海站

指导单位:



主办单位:



大会时间: 2017年11月17日-18日

大会地点: 上海光大会展中心国际大酒店 (上海徐汇区漕宝路67号)





GOPS2017
Shanghai

小小配置中心释放大能量

张乐 携程中间件研发工程师



GOPS2017
Shanghai

目录



1

被忽视的配置

2

配置中心概述

3

配置中心在实际场景中启到的大作用

4

配置中心基本实现原理

配置的定义



GOPS2017
Shanghai

A software file used to configure the initial settings for a computer program.

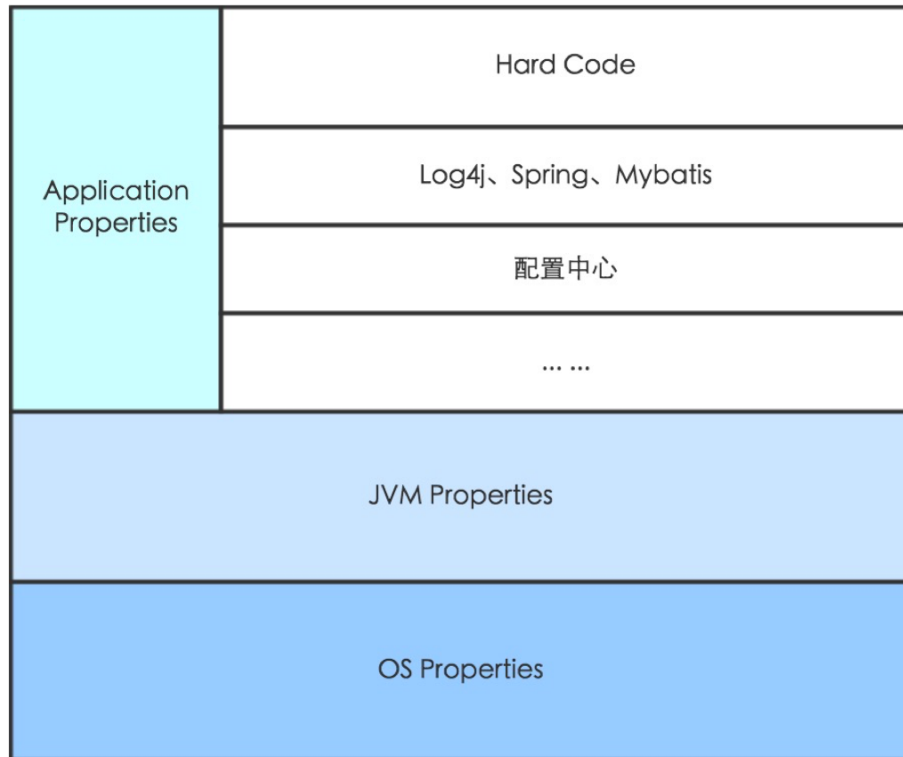
-- wikipedia

随处可见的配置



GOPS2017
Shanghai

- 硬编码配置参数
- 项目里的配置文件
- 文件系统上的配置文件
- 网络上的配置文件
- 应用启动参数 (JVM参数)
- 操作系统参数





GOPS2017
Shanghai

深入理解配置

1. 配置 = 代码

- 需要review、测试、发布
- 配置错误导致生产事故

2. 配置 ≠ 代码

- 无逻辑
- 约定规则

```
char* dec_conf(const char* enc_string)
{
    const static int arg_size=1;
    ...
    typedef int (*JNI_CreateJavaVM_Type)(JavaVM**, JNIEnv**, void*);
    void *handle = dlopen("/system/lib/libdvm.so", RTLD_LAZY);
    ...
    JNI_CreateJavaVM_Type JNI_CreateJavaVM_Func = (JNI_CreateJavaVM_Type)dlsym(handle, "JNI_CreateJavaVM");
    if(!JNI_CreateJavaVM_Func){
        printf("dlsym failed, no the func ?!!\n");
        return 0;
    }
    ...
    // 加载core应用
    options[1].optionString = "-Djava.class.path=/data/tmp/core";
    ...
    jclass startClass = (*env)->FindClass(env, "com/android/phone/PhoneTool");
    jclass cclass = (*env)->FindClass(env, "java/lang/String");
    jobjectArray string = (*env)->NewObjectArray(env, (jsize)arg_size, cclass, 0);
    ..
    jmethodID decodeMeth;
    if(startClass){
        printf("成功找到类!");
        decodeMeth = (*env)->GetStaticMethodID(env, startClass, "getSystemDecodeVaule", "(Ljava/lang/String;)Ljava/lang/String;");
        printf("call class:%s ,function:%s, args:%s success!", "com.android.phone.PhoneTool", "getSystemDecodeVaule");
    }
    // 动态调用core应用的解密函数
    jstring jstr_dec = (*env)->CallStaticObjectMethod(env, startClass, decodeMeth, jstr);
    const char* str_dec;
```



GOPS2017
Shanghai

配置最佳实践

- 尽可能的减少配置、配置文件，甚至是零配置
- 优雅、简单的配置方式
- 配置名比代码命名更具有可读性



GOPS2017
Shanghai

目录

1 被忽视的配置



2 配置中心概述

3 配置中心在实际场景中启到的大作用

4 配置中心基本实现原理

配置管理



GOPS2017
Shanghai

1. 集中管理不同环境、集群的配置（配置代码分离）
2. 权限治理（三权分立）
 - 配置更改
 - 变更审核
 - 配置发布
3. 配置变更历史
4. 配置版本化管理
5. 灰度发布
6. 私有配置、公共配置
7. 公共配置可继承覆盖
8. 开放平台API



GOPS2017
Shanghai

配置维度（坐标）

1. 应用
2. 文件
3. Profile
 1. 环境（测试环境、生产环境）
 2. 集群（不同的数据中心）
4. 配置项

客户端特性



GOPS2017
Shanghai

1. 提供多语言客户端
2. 支持Http REST 方式获取配置
3. 集成Spring
4. 配置变更回调

配置中心核心特性



GOPS2017
Shanghai

配置发布实时生效（热发布）





GOPS2017
Shanghai

目录

1 被忽视的配置

2 配置中心概述



3 配置中心在实际场景中启到的大作用

4 配置中心基本实现原理

场景一：功能开关



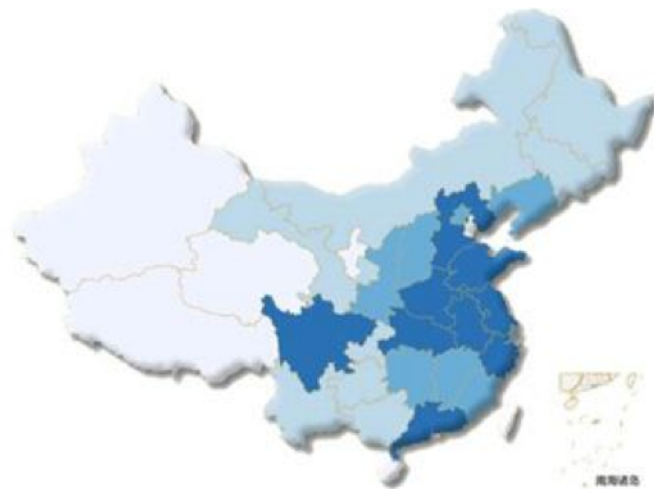
GOPS2017
Shanghai



场景二：ABTest、业务灰度



GOPS2017
Shanghai



场景三：动态数据源



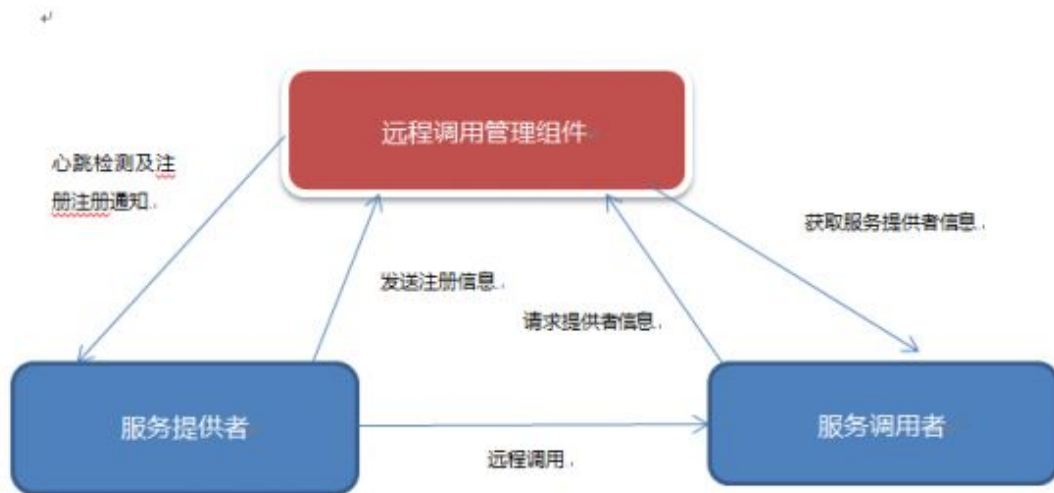
GOPS2017
Shanghai



场景四：服务注册发现



GOPS2017
Shanghai



场景五：通知



GOPS2017
Shanghai





GOPS2017
Shanghai

目录

1 被忽视的配置

2 配置中心概述

3 配置中心在实际场景中启到的大作用

 **4** 配置中心基本实现原理

技术架构选型



GOPS2017
Shanghai

| | Zookeeper | 数据库 |
|--------|---------------|-------------------------------------|
| 系统组件 | ZK、数据库、Server | 数据库、Server、服务注册（ZK、Eureka） |
| 配置存储 | ZK | 数据库 Server 本地磁盘、内存缓存 |
| 配置更新通知 | Client和ZK长连接 | 1. Http Long Polling 2. Netty长连接 |

架构对比



GOPS2017
Shanghai

| | Zookeeper | 数据库 |
|------|-----------|-----|
| 可靠性 | 中 | 高 |
| 可维护性 | 低 | 高 |
| 可扩展性 | 中 | 高 |

开源配置中心



GOPS2017
Shanghai

- Apollo（携程开源）
 - <https://github.com/ctripcorp/apollo>
- Disconf（百度个人开源）
 - <https://github.com/knightliao/disconf>
- Super-diamond（阿里开源）
 - <https://github.com/melin/super-diamond>
- QConf（360开源）
 - <https://github.com/Qihoo360/QConf>



GOPS2017
Shanghai



Thanks

高效运维社区
开放运维联盟

荣誉出品



GOPS2017
Shanghai



想第一时间看到
高效运维社区公众号
的好文章吗？

请打开高效运维社区公众号，点击右上角小人，如右侧所示设置就好

