ArchSummit全球架构师峰会深圳站2016

Eden
百度搜索引擎的PaaS架构设计和实践

百度搜索架构师 郑然

自我介绍

郑然

- 百度网页搜索架构部 – 搜索架构师
- 七年搜索引擎架构工作经验
  - 流式索引构建系统, 离线计算平台架构
  - 服务化组件开发平台SOFA
  - 在线服务PaaS平台建设
  - 服务治理 & 高可用架构 & DevOps

目录
CONTENTS

百度搜索引擎的挑战

Eden的前世今生

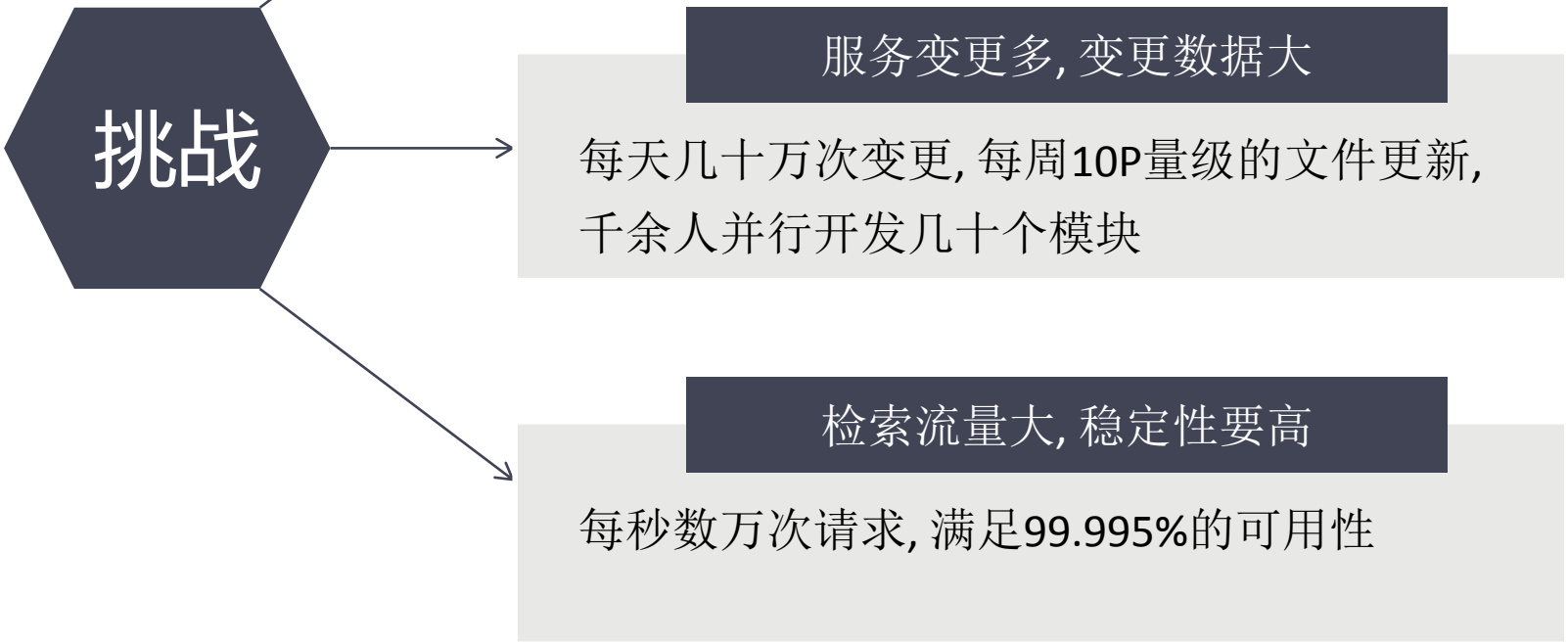
软件包的标准化

Eden架构设计和实践

经验教训

百度搜索引擎的挑战

百度搜索引擎的挑战

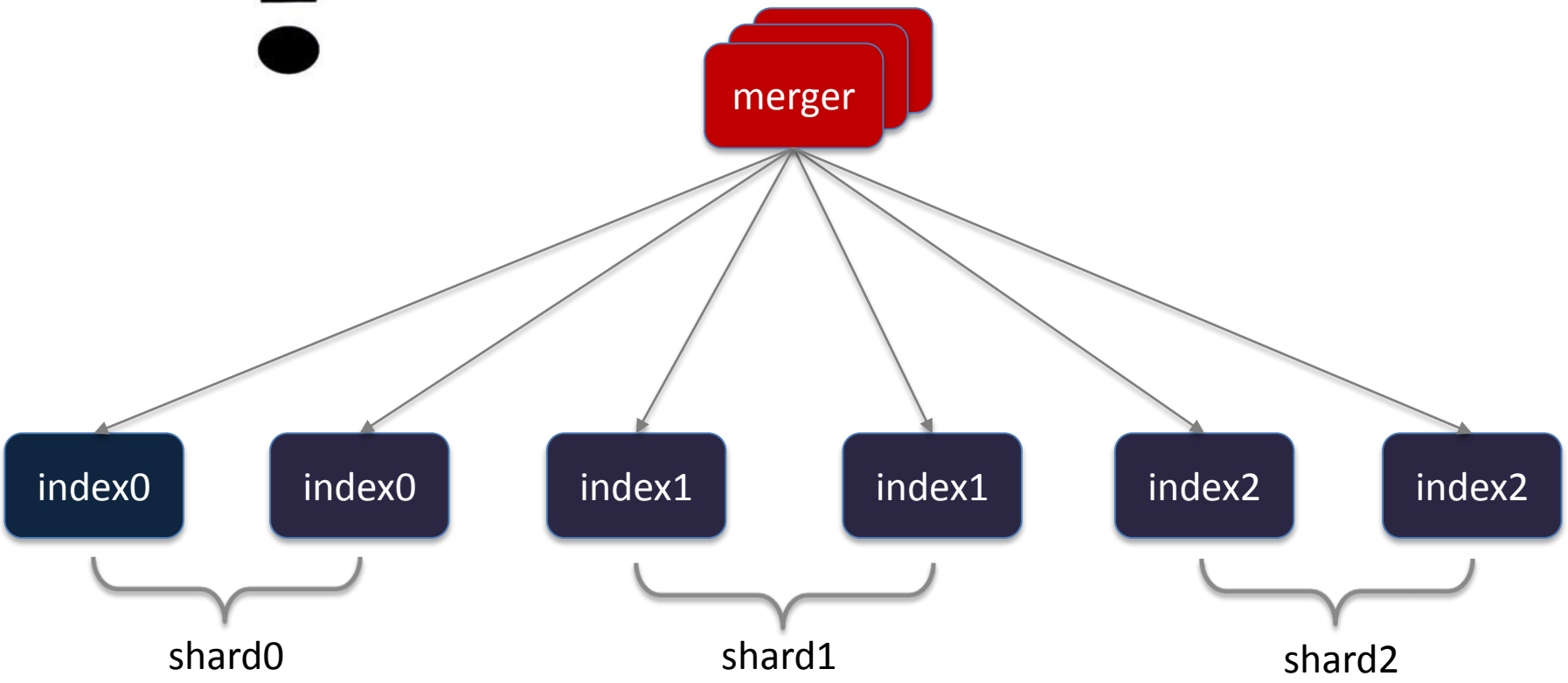

挑战
机器数量多，服务数量大
数万台服务器，数十万个服务，分布在多个IDC
服务变更多，变更数据大
每天几十万次变更，每周10P量级的文件更新，
千余人并行开发几十个模块
检索流量大，稳定性要高
每秒数万次请求，满足99.995%的可用性

Eden的前世今生

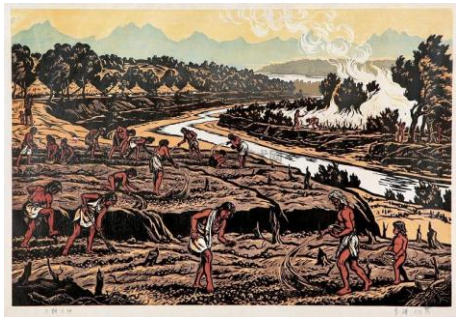
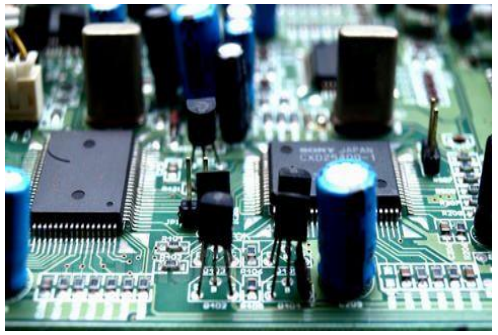
从搜索引擎的基础架构说起

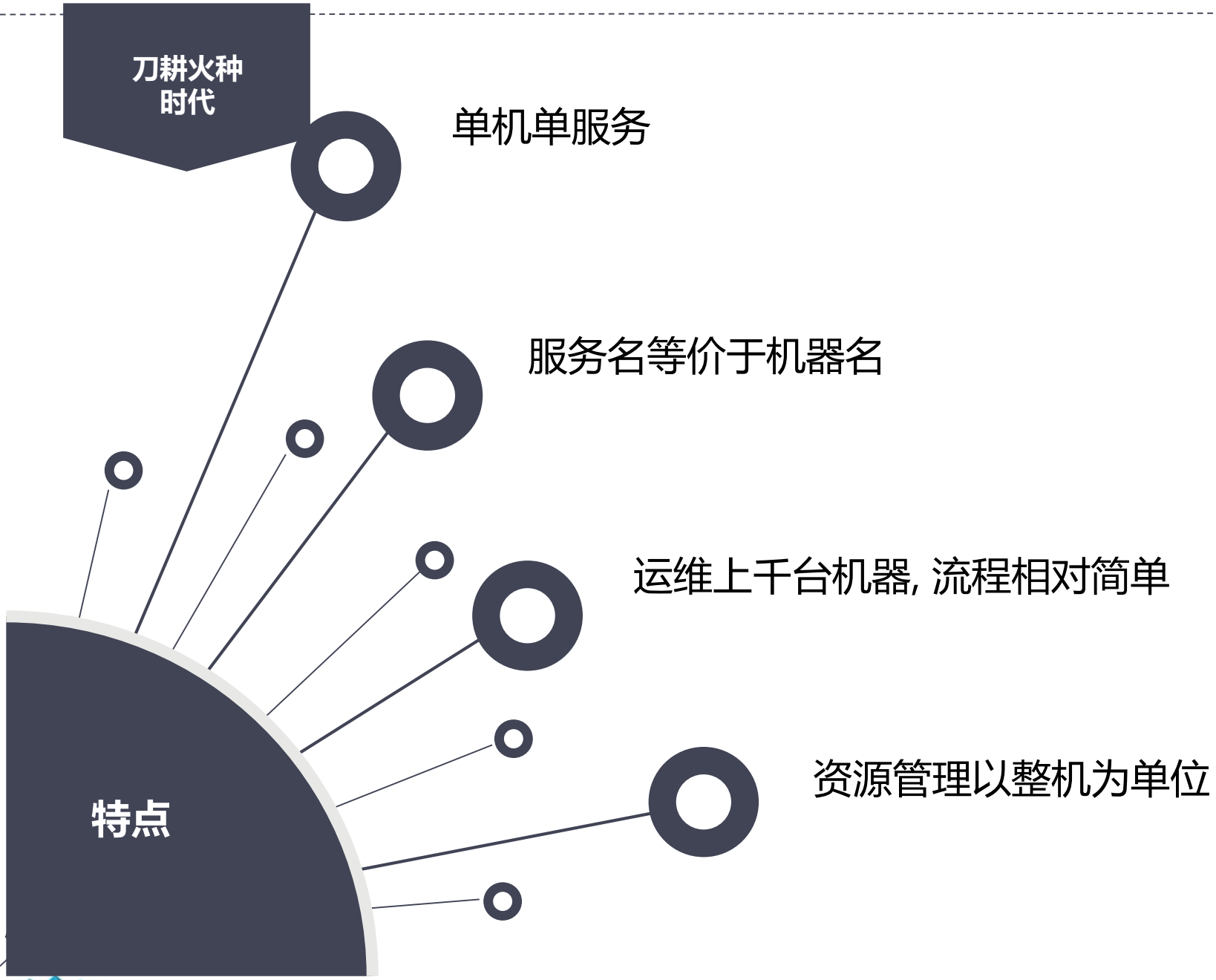
?

海量服务分布 吞吐和延迟最优



服务治理的三个阶段

2009年以前	2010 ~ 2013	2014 ~ 2016
		
刀耕火种时代	蒸汽机车时代	集成电路时代
粗暴 简单	半自动化	精密 高效


刀耕火种
时代
单机单服务
服务名等价于机器名
运维上千台机器, 流程相对简单
资源管理以整机为单位
特点

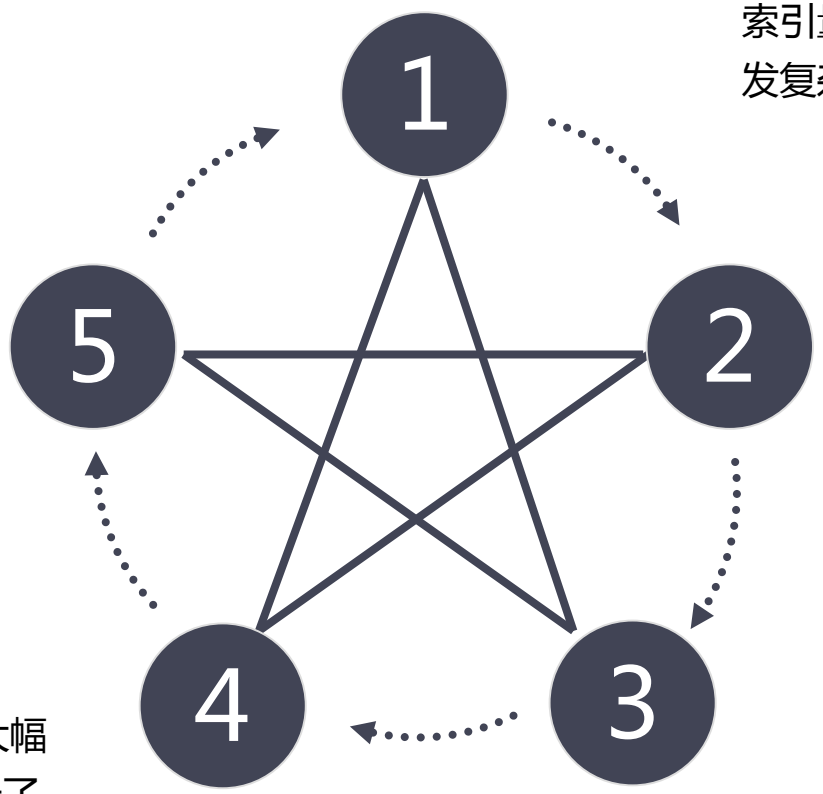
蒸汽机车
时代

业务驱使

索引量增长, 相关性算法越发复杂, 资源消耗增加

缺陷

- 没有把服务治理系统作为一个有机整体



资源优化?

单机单服务, 机型差异大, 资源使用不均衡, 呈现严重的木桶效应

服务治理问题

开发了DOP系统, 大幅度节省了资源, 提升了运维效率

单机多服务

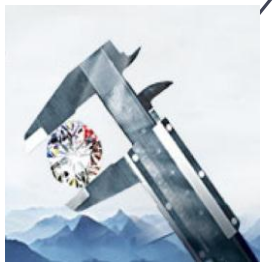
单个索引shard瘦身, 优化索引分布

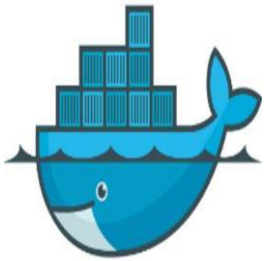
集成电路
时代

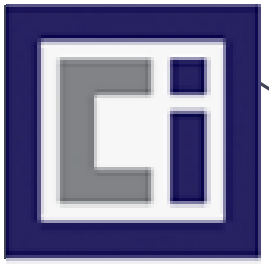
1	生龙活虎	服务存活率 99.5%
2	弹性伸缩	弹性的服务扩容和缩容
3	海量变更	几十个模块, 上千路数据, 每天几十万次变更
4	生态建设	以Eden为核心构建DevOps生态

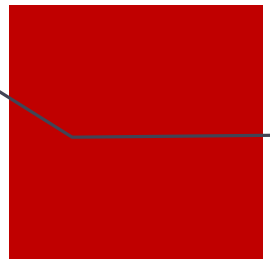
软件包的标准化

部署系统的核心









- 标准化是自动运维的基础
- 部署系统的核心是包的标准化
- 本质上希望自给自足
- 容器的发展更说明包的重要性
- OCI定义了容器的标准

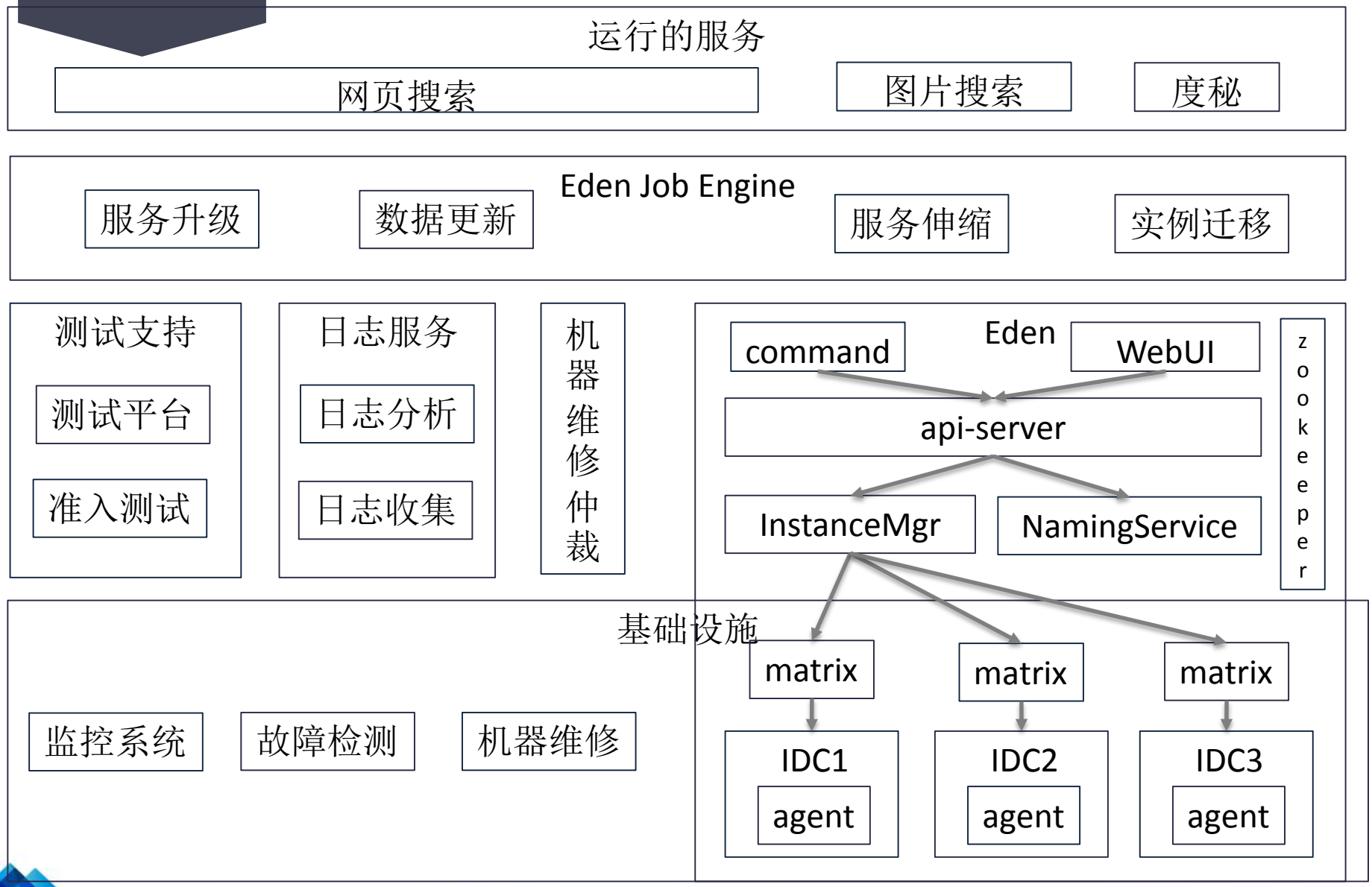
Eden
标准化包

```
"package":{
    "dataSource":"hdfs://hdfs.baidu.com:
    "pre_action":"stop",
    "post_action":"start",
    "transaction_id":14807
},
"data":{
    "resource":{
        "cpu":{
            "numCores":324,
            "overUseEnabled":true
        },
        "memory":{
            "sizeMB":13312,
            "overUseEnabled":false
        },
        "disks":{
            "workspace":{
                "sizeMB":5120,
                "numInodes":100000
            }
        }
    },
    "mask":{
        "dataSource":"hdfs://hdfs.baidu.com:54310/
        "pre_action":"",
        "post_action":"restart",
        "transaction_id":15900
    },
    "click":{
        "dataSource":"hdfs://hdfs.baidu.com:54310/
        "pre_action":"",
        "post_action":"restart",
        "transaction_id":15900
    }
}
    "naming":{
        "dependency":[
            {
                "app_id":"app1",
                "attribute":{
                    "MIN_USABLE_RATIO":"50"
                }
            },
            {
                "app_id":"app2"
                "attribute":{
                    "MIN_USABLE_RATIO":"70"
                }
            }
        ]
    }
```

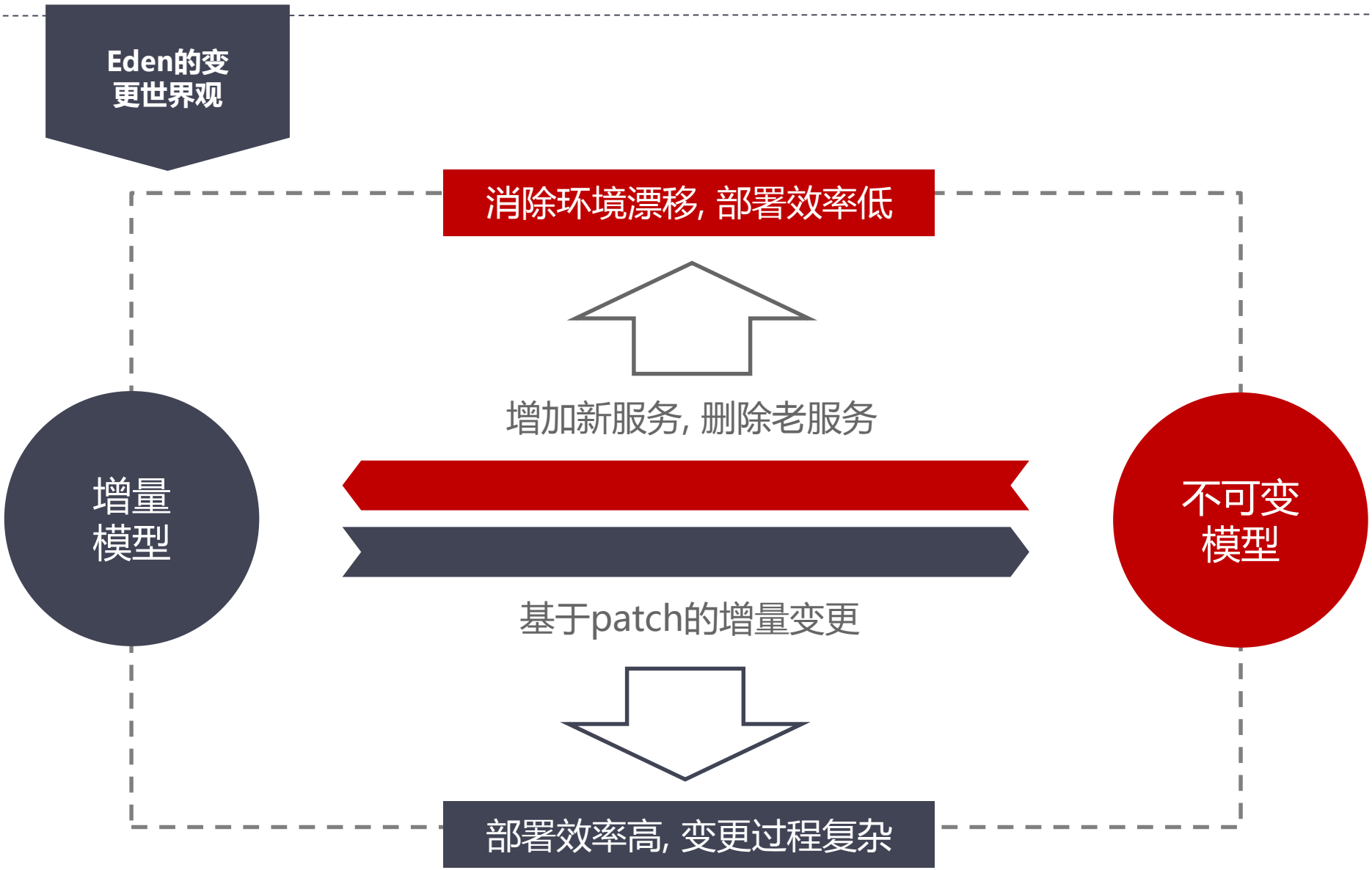
架构设计和实践

- 架构蓝图
- Eden的变更世界观
- 故障和高可用

架构蓝图


运行的服务
网页搜索
图片搜索
度秘
Eden Job Engine
服务升级
数据更新
服务伸缩
实例迁移
测试支持
测试平台
准入测试
日志服务
日志分析
日志收集
机器维修仲裁
command
Eden
WebUI
api-server
InstanceMgr
NamingService
zookeeper
基础设施
监控系统
故障检测
机器维修
matrix
matrix
matrix
IDC1
agent
IDC2
agent
IDC3
agent

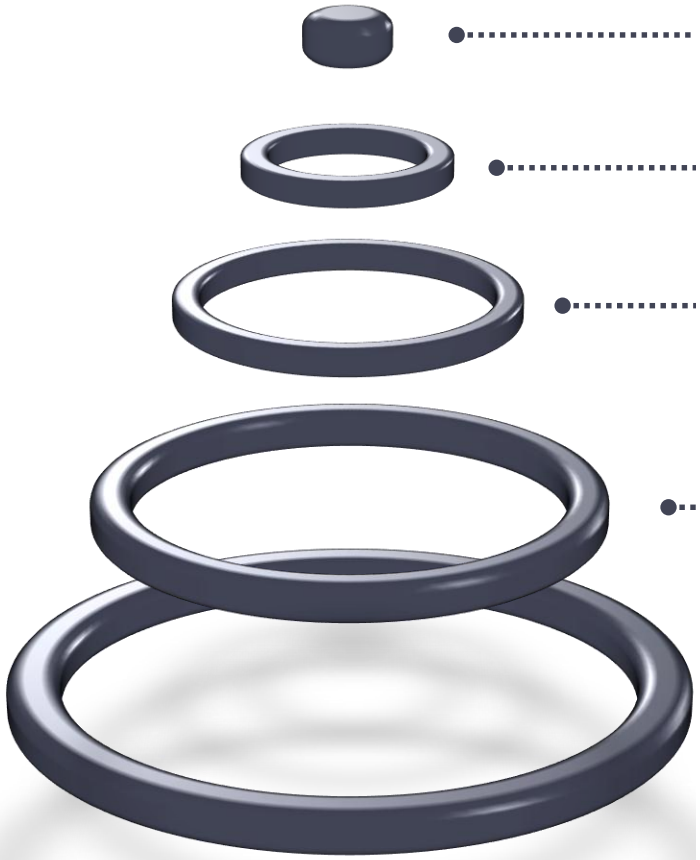
Eden的变更世界观

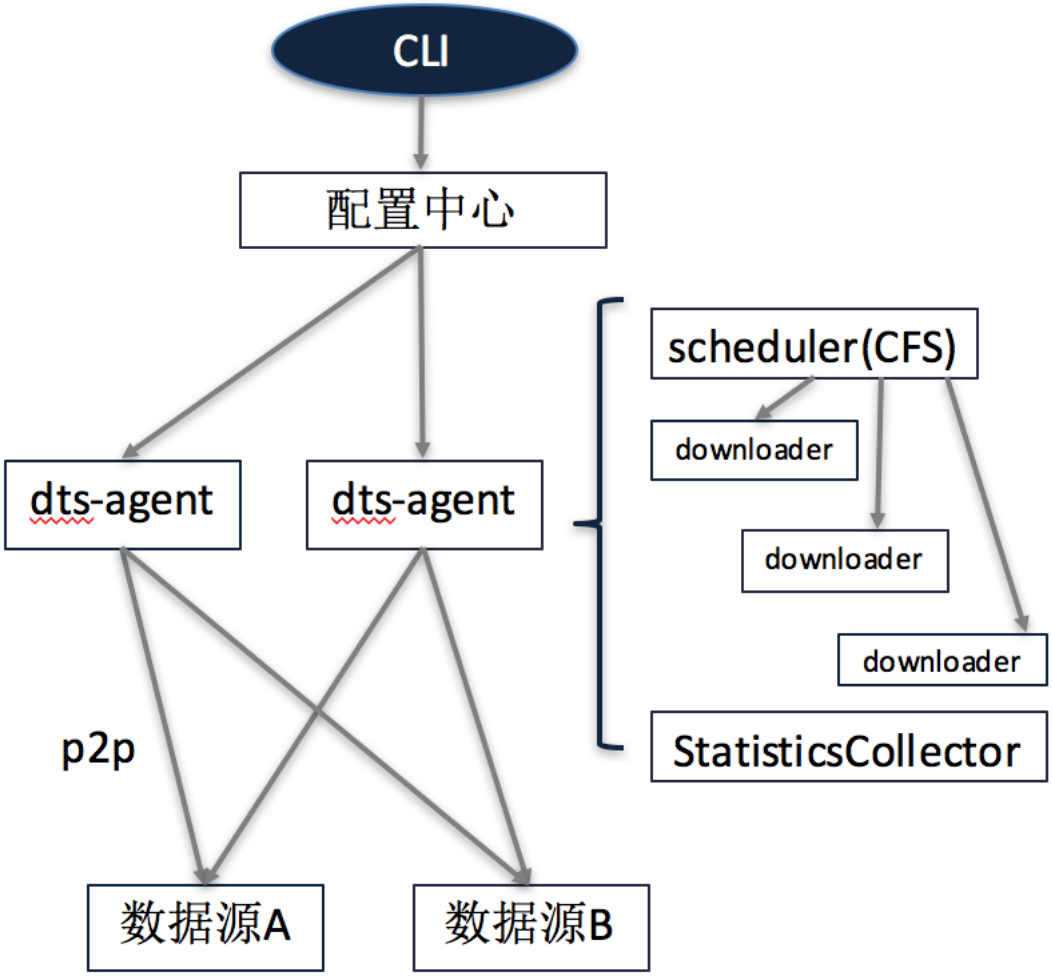

消除环境漂移, 部署效率低
增加新服务, 删除老服务
增量模型
不可变模型
基于patch的增量变更
部署效率高, 变更过程复杂

Eden的变更世界观

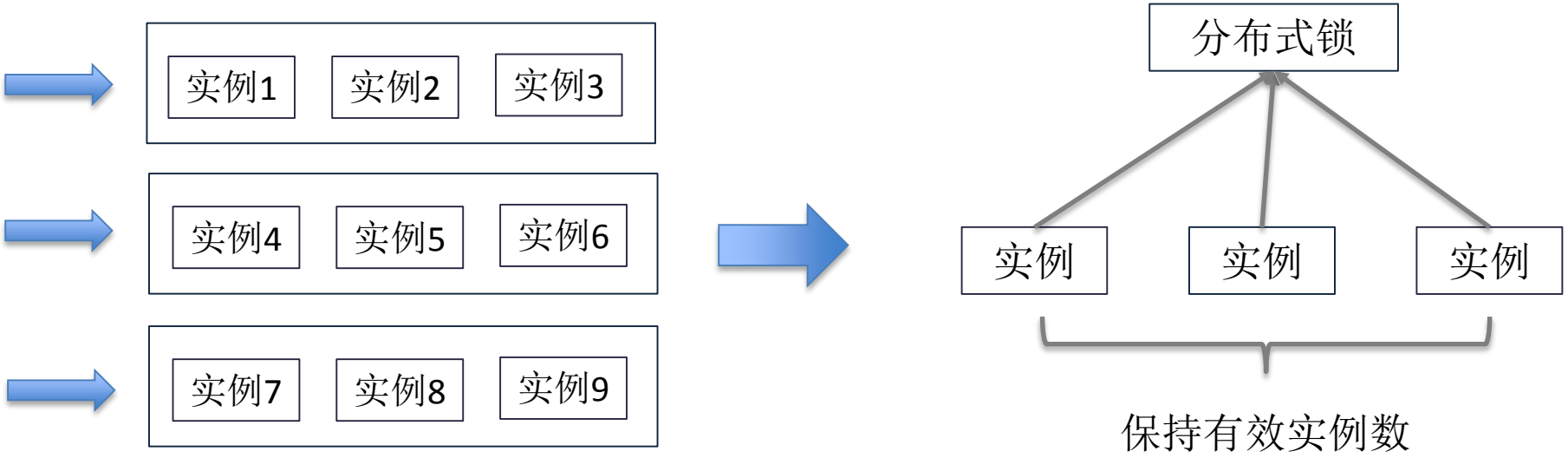

状态对比
记录实例的目标状态和当前状态
darcs理论
文件下载
文件下载后执行前置和后置命令
执行命令
事务
改变状态

Eden的变更世界观




CLI
配置中心
dts-agent
dts-agent
scheduler(CFS)
downloader
downloader
downloader
StatisticsCollector
p2p
数据源A
数据源B

Eden的变更世界观



Eden的变更世界观

STEP 1

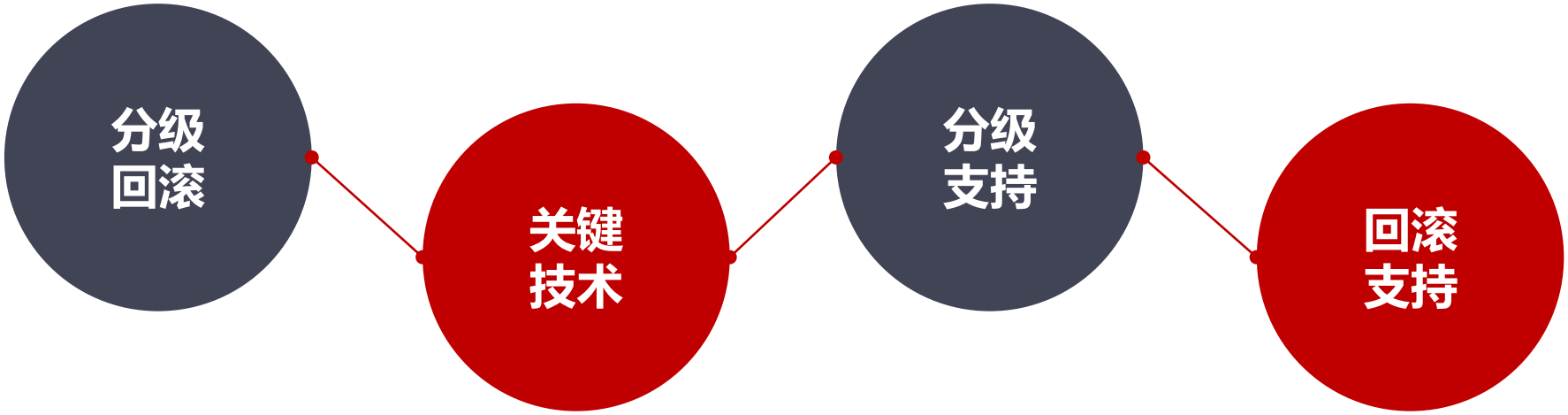
常规部署

1. 文件传输
2. 执行前置命令
3. 切换新文件
4. 执行后置命令

```
"dataStore":{
    "binary":{
        "version1":{
            "dataSource":"hdfs://hdfs.baidu.com:54310/product/ps/se/bs/v1",
            "priority":100,
            "md5sum_file":"md5"
        },
        "version2":{
            "dataSource":"hdfs://hdfs.baidu.com:54310/product/ps/se/bs/v2",
            "priority":100,
            "md5sum_file":"md5"
        }
    },
    "deploy"{
        "package":{
            "data_id":"dataStore.binary.version1",
            "pre_action":"stop",
            "post_action":"start",
            "transaction_id":14807
        },
```

文件预分发实现了变更效率质的飞跃

Eden的变更世界观



变更的原则

分级和可回滚是任何一个变更操作的基本原则

关键机制

实例级变更的能力和备份的能力

分级粒度

实例级的数据, 连接, 资源配额的变更

数据备份

元数据的版本化管理和文件的版本化管理

没有分级和回滚的变更就是一枚炸弹!

故障

- 硬件
  - 主板, CPU, 内存, 风扇, 网卡, Raid卡
  - 磁盘(挂载点缺失, 扇区损坏, SMART, 设备文件故障)
- 软件
  - 文件系统(文件损坏, inode满, 磁盘满)
  - ssh登录失败
  - agent假死
- 全靠人的时代: 服务存活率 96%, 机器故障率 4%

HOW

- 迁移实例?
  - 无冗余资源, 副本数多
  - 有冗余资源, 副本数少
- 故障机器多, 全部送修?
  - 死机 → 送修
  - 带伤 → 分批送修. 优先级呢?
- 机器修复之后?
- 重启 or 重装 ?

故障
高可用

Parallel

agent保活

进程存活

版本一致

服务维修

原地维修

迁移服务

机器维修

优 先 级

漏斗分类

副本状态

故障重装

我们做到了

机器健康率

98.5%

服务存活率

99.5%

故障自愈能力

故障
高可用

服务的中枢

- 2014年的一次误操作, 瞬间删除了一个服务单元的摘要服务
- 2015年的一次错误配置, 一天内缓慢删除了2000个服务, 第二天才发现
- Eden不可用, 搜索服务不可变更, 影响时效性结果, 阻碍近千人的开发团队上线

高可用机制

- 多维度全局安全阈值
- app粒度的权限控制机制
- agent不可用, 不影响正在运行的服务
- 版本化服务描述文件的变更, 做到变更可追溯

经验教训

经验教训

经验

- 故障检测和维修切记太敏感
- 任何变更都要分级
- 故障不可怕, 关键是自愈能力
- 放牛式思维

教训

- 可视化是自动化的前提
- 自助化+平台级的约束能力
- 接口幂等化
- safe mode机制保底

深耕运维 精益求精

服务治理 匠心精神

Thanks!