Matrix集群操作系统

百度-INF-吕斯哲

Baidu 百度

About

部门
- 基础架构部
- 大规模分布式系统

团队
- COS: Cluster Operating System
- Matrix/Normandy

个人
- 吕斯哲
- 阅读、慢跑、徒步

招聘: inf-jobs@baidu.com

演讲内容

Matrix	宏观结构	容器与混布	工程实施
• 背景 • 定位	• 分层结构 • 关键概念	• 应用部署 • 隔离、避让 • 调度、超发	• 系统工程 • 推进思路

Matrix是什么？

Matrix

历 史

提升资源利用率

- CPU利用率
- 资源交付与流转效率

抽象资源操作接口

- PaaS、运维平台
- DevOps/CI/CD

Matrix

- 2012年启动
- 获百度最高奖

现 状

规模

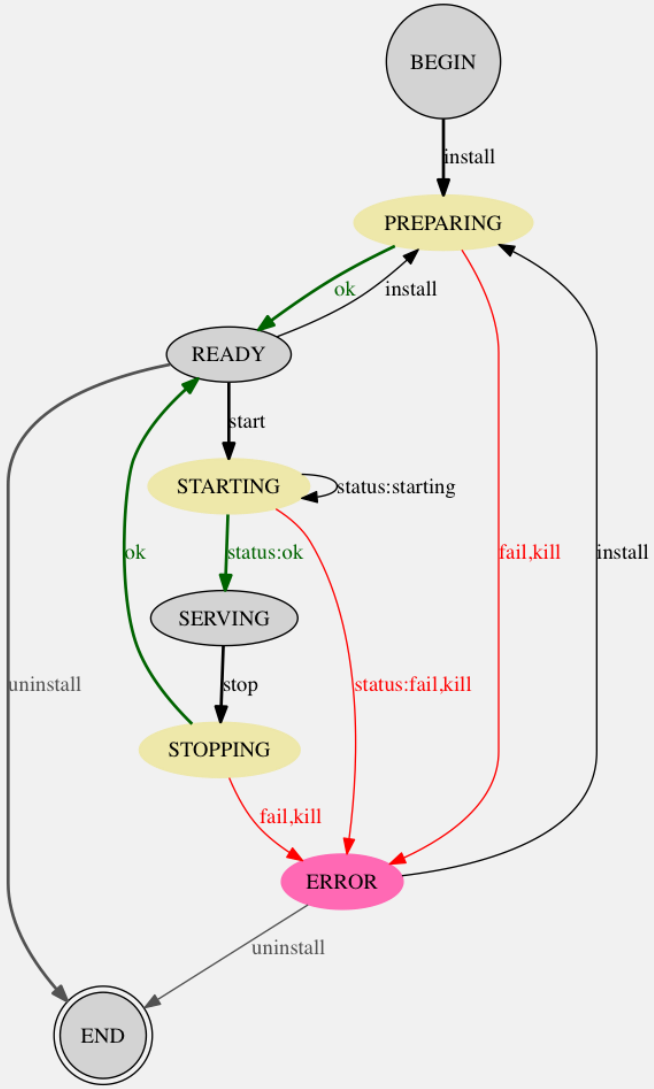
- 全集群：几十万台
- 单集群：几万台
- 容器数：几千万个

业务

- almost all
- 在线&离线
- 虚拟化战略

Matrix分层架构

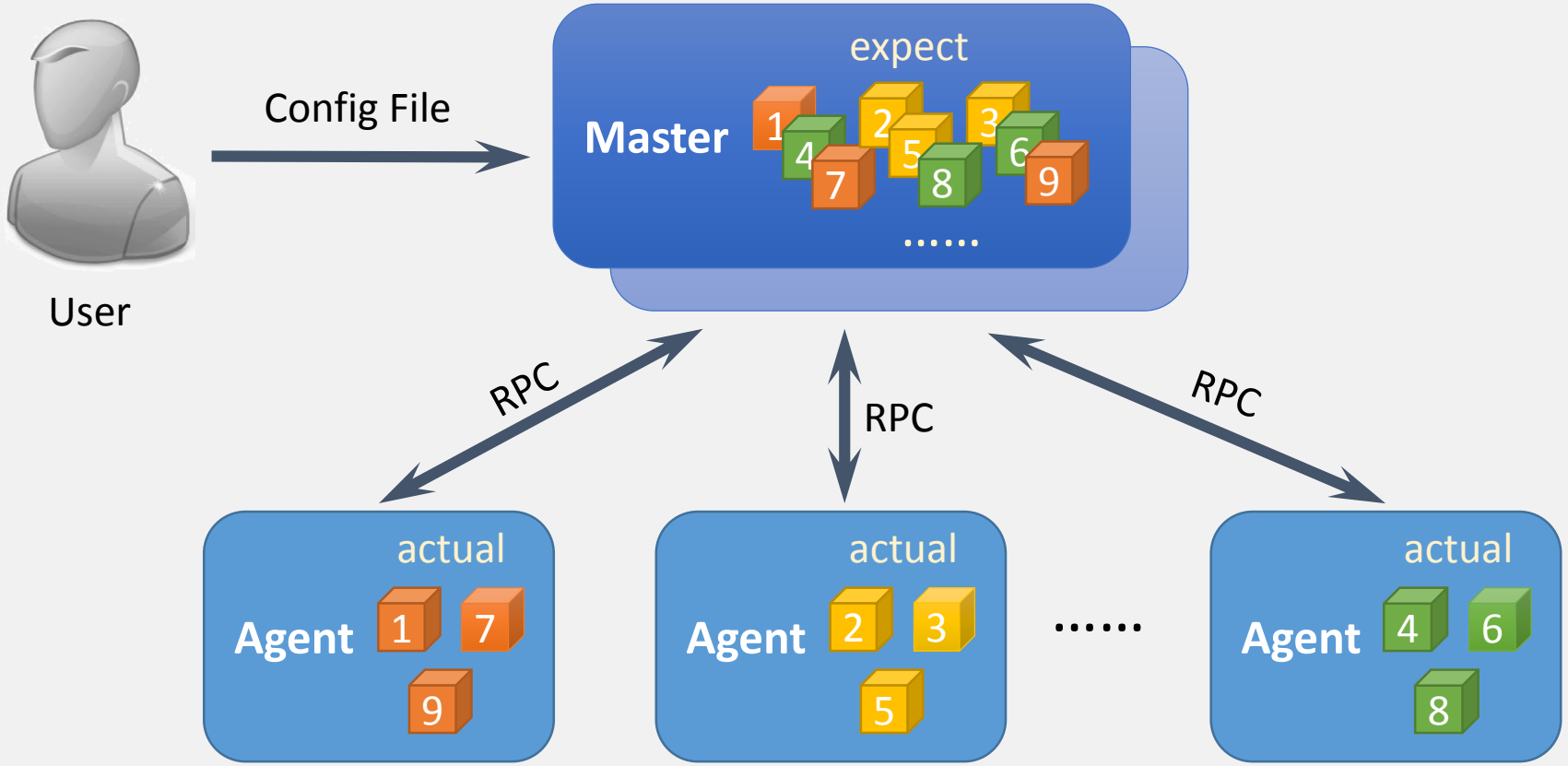
原理: 状态机设计



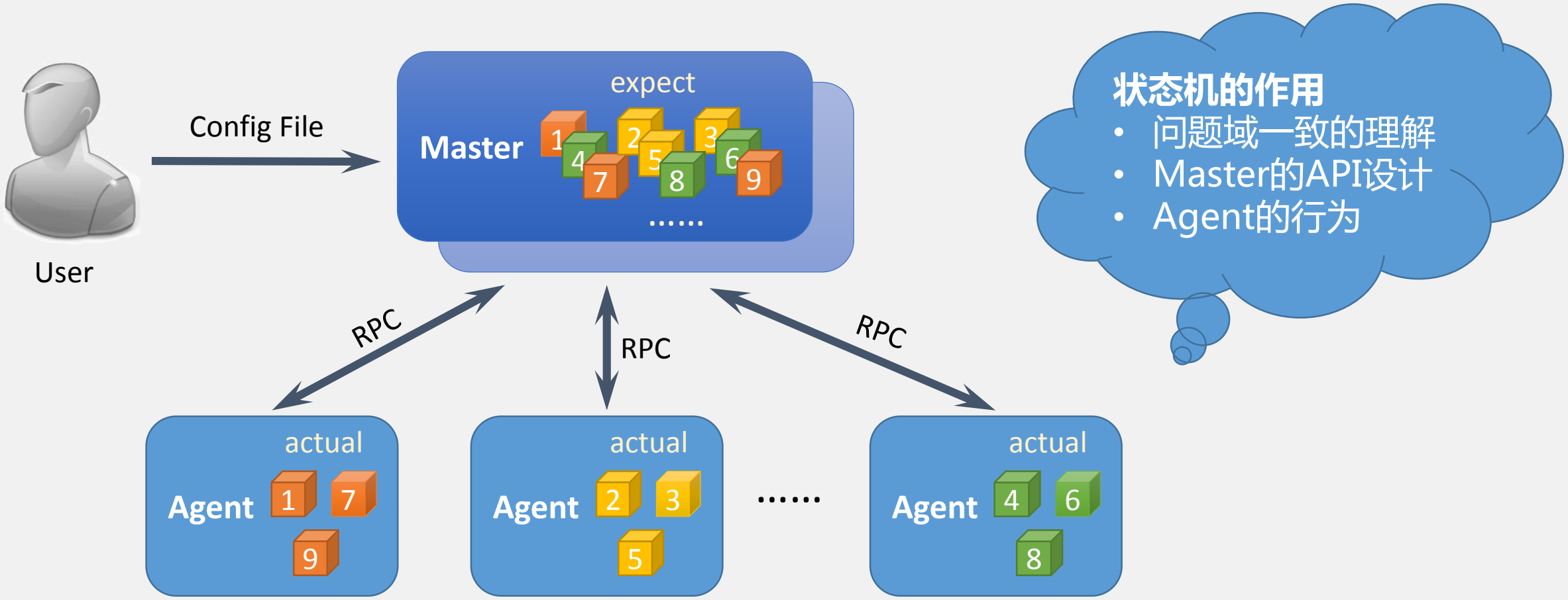
状态转移表

actual \ expect	-	READY	SERVING
-	-	install	install
READY	uninstall	-	start
SERVING	stop	stop	-

原理: 状态机设计


User
Config File
Master
expect
1
2
3
4
5
6
7
8
9
.....
RPC
RPC
RPC
Agent
actual
1
7
9
Agent
actual
2
3
5
.....
Agent
actual
4
6
8

原理: 状态机设计


User
Config File
Master
expect
1
4
7
2
5
8
3
6
9
.....
RPC
RPC
RPC
Agent
actual
1
7
9
Agent
actual
2
3
5
.....
Agent
actual
4
6
8
状态机的作用
• 问题域一致的理解
• Master的API设计
• Agent的行为

哪些对象？

Host
- 机器流转
- 自动维修
- 环境一致化

机器管理
Ultron/Bios

Container
- 资源的集合
- 隔离技术的载体
- 资源调度的单位

Matrix-RM

Instance
- 资源的使用
- “进程”
- 应用部署

Matrix-IM

Service/Server
- 常驻”进程”
- 在线服务类

Solaria

Job/Command
- 自结束”进程”
- 离线作业类

Solaria/Neptune

分层结构


WebUI
RPC
CLI
配置中心
数据系统
采集
存储
传输
计算
权限管理
Quota管理
作业编排
服务
基础功能
服务描述
灰度更新
Plugins
配置管理
日志系统
Tracing
服务保活
服务监控
Naming
自动扩缩容
自动重分布
RPC
简单作业
MPI
.....
MR
TM
任务管理
实例生命周期
任务执行
实例元数据
部署控制
任务调度
任务队列
资源管理
资源模型
资源分配
Container 元数据
Container 生命周期
单机技术
资源隔离
资源监控
执行器
内核特性
机器管理
环境管理与一致化
故障检测
自动维修

分层结构


WebUI
RPC
CLI
solacfg DSL
配置中心
作业编排
Compass
数据系统
采集
存储
传输
计算
服务
基础功能
服务描述
灰度更新
Plugins
配置管理
日志系统
Tracing
服务保活
服务监控
Naming
RPC
Solaria
简单作业
Neptune
MPI
.....
MR
TM
各计算框架
CASIO
任务管理
实例生命周期
任务执行
Matrix-IM
署控制
任务调度
任务队列
Normandy
权限管理
资源管理
资源模型
资源分配
Container 元数据
Container 生命周期
Matrix-RM
单机技术
资源隔离
资源监控
执行器
内核特性
Quota管理
机器管理
环境管理与一致化
故障检测
自动维修
ultron/bios

分层结构

- **分配资源：Container**

```
buf = malloc(…)
```

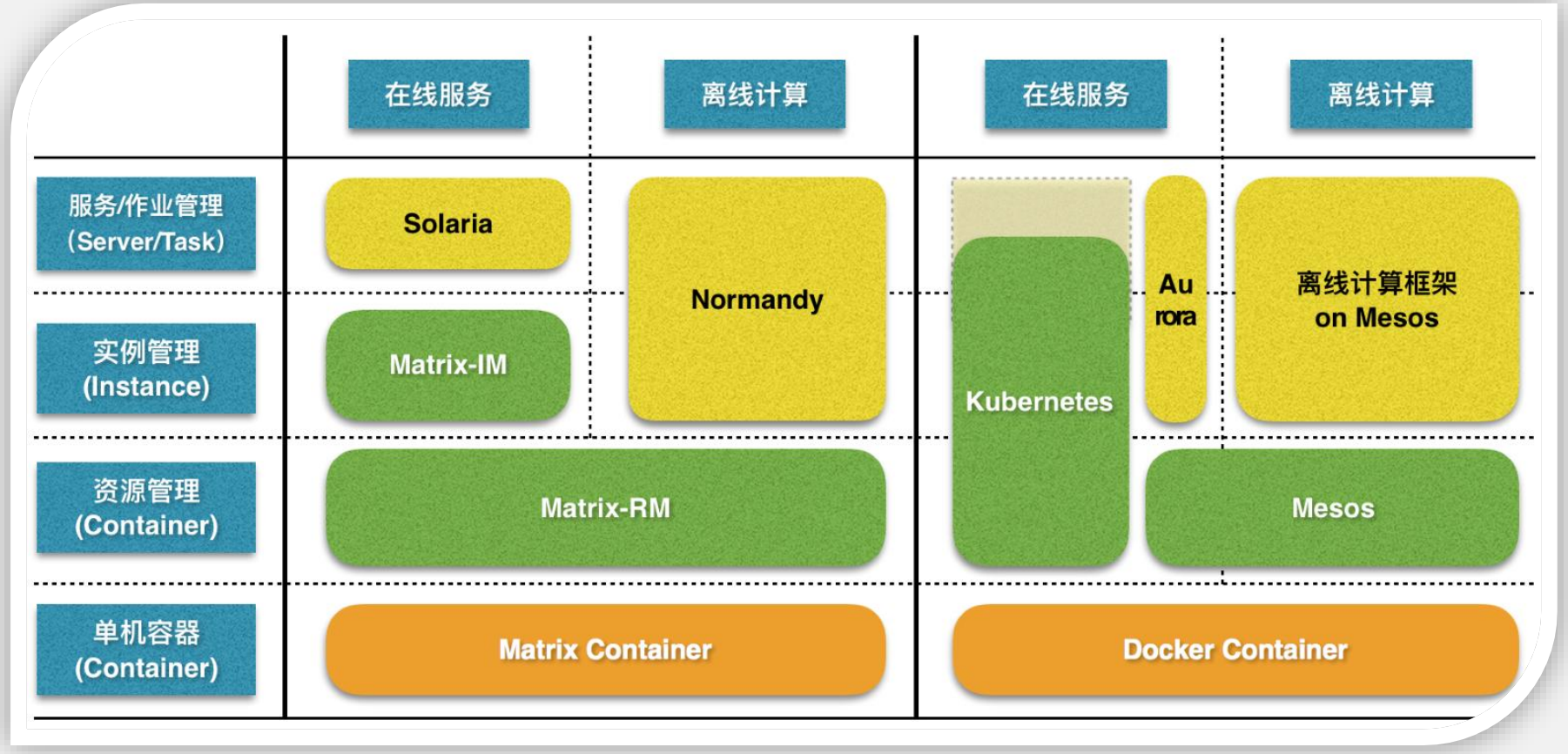
- **实例部署：Instance**

```
instance = new (buf) CTOR(package)
```

- **模型分化**
  - **long-running: service/server**
  - **routine: job/command**

```
server = (Server) instance
server.start()
```

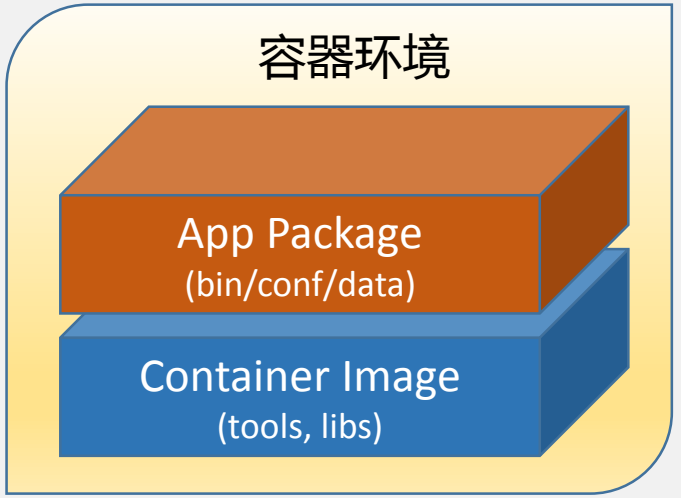
社区对比


在线服务
离线计算
在线服务
离线计算
服务/作业管理
(Server/Task)
实例管理
(Instance)
资源管理
(Container)
单机容器
(Container)
Solaria
Normandy
Matrix-IM
Matrix-RM
Matrix Container
Kubernetes
Au
rora
离线计算框架
on Mesos
Mesos
Docker Container

容器与混布

应用部署

- **Docker**: Container & Image
- **Matrix**：
  - 保留Matrix-Container
    - ✓ 保留自研混布、调度技术
    - ✓ 内部生态对接：可继续大规模部署
  - 支持Docker-Image部署
    - ✓ 两层结构：应用+容器
    - ✓ 现有部署系统兼容
    - ✓ 工具链升级: gcc, java, tomcat etc.



双层结构

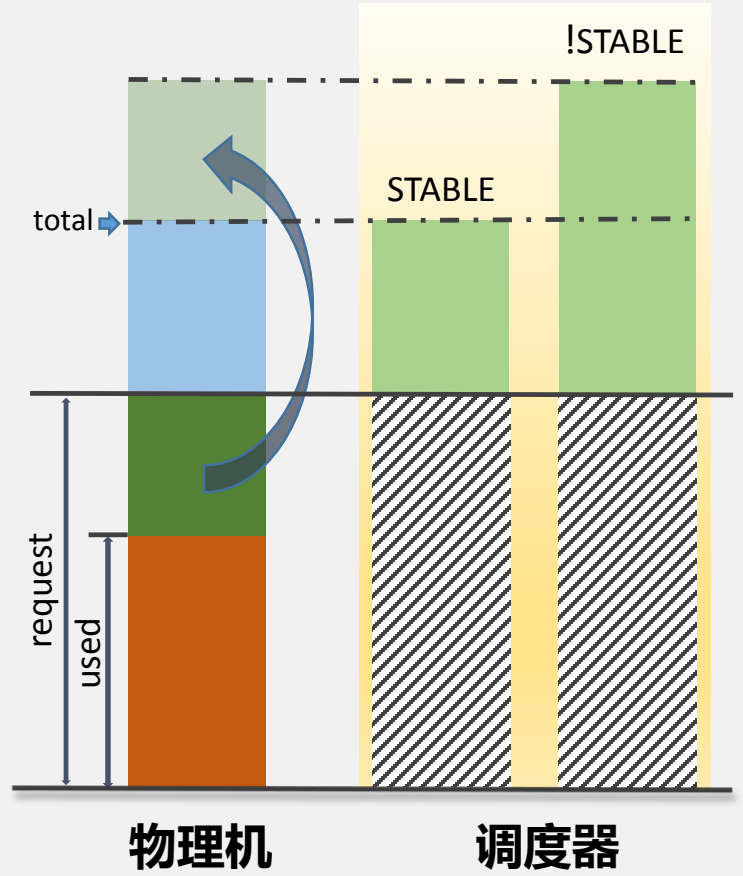
优先级与超发

优先级

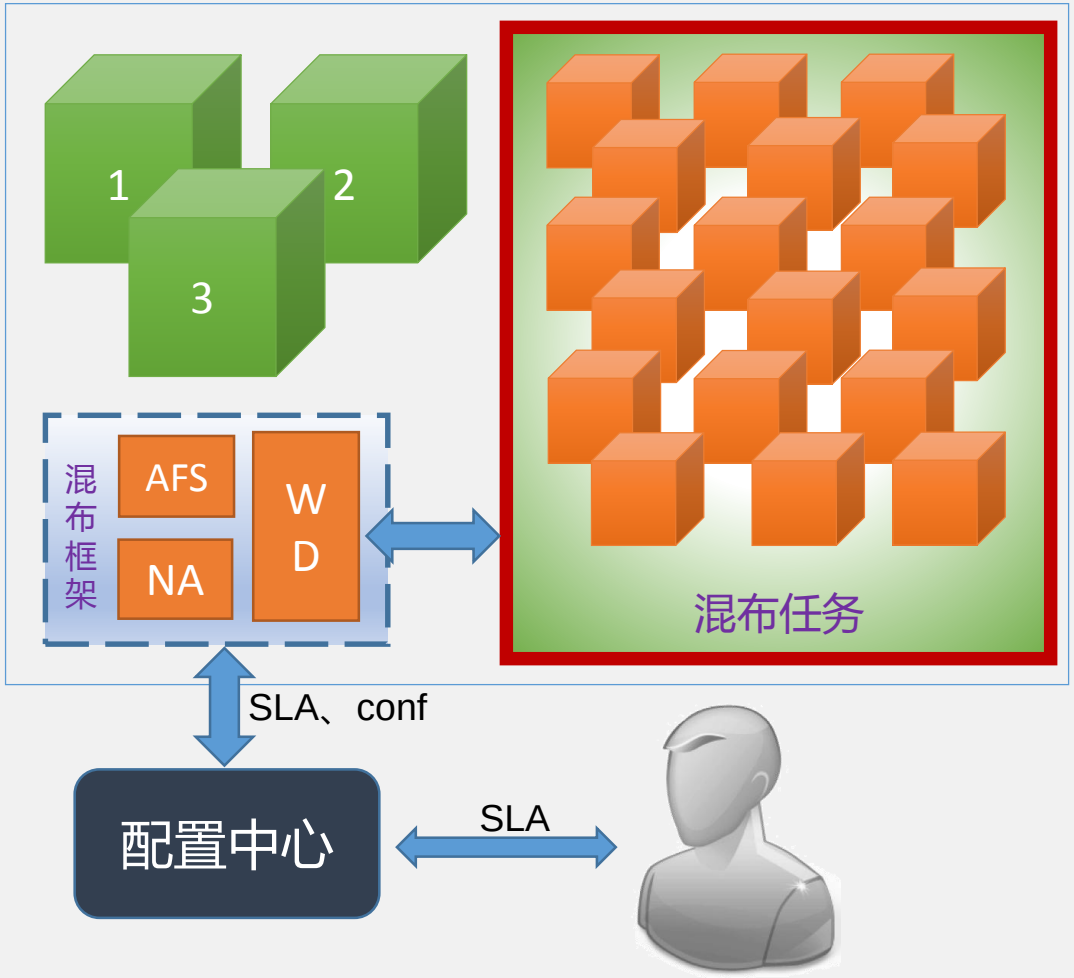
- STABLE > NORMAL > BESTEFFORT
- 单机：可否抢占
- 集群：是否有Quota

超发

- 用户行为：request – used > 0
- 超发：total’ = total + sum(request – used)
- 约束：sum(STABLE) < total



隔离与避让



混布框架
- 存储：AFS(hdfs)
- 计算：Normandy-Agent
- 策略：watchdog

混布策略
- 确保非混布任务资源质量
- 本地监控：SLA、策略配置
- 本地即时处理：压缩 or kill

用户运营
- 签订SLA
- 资源结算模式调整(ing)

集群混布

离线间

- 跨计算模型
- 计算模型适配
- 集中调度
- 兼容YARN
- 2014 - 2015

在离线间

- 跨业务
- 离线数据
- 业务SLA保证
- 2015 - Now

无差别

- 优美
- 调度策略优化
- 背包问题
- next...

大规模推广

为什么自研？

- 启动时间：2012年
- 重心：集群资源管理 vs 运维效率提升
- 降低业务接入成本：
  - 适配、脏逻辑
  - 内部生态对接: 机器维修、预结算、众多系统
  - 目标：扩大覆盖规模、业务多样性
- 紧盯社区：拆解替换、降低维护成本、反馈

推进过程


2012H1
• Matrix-1.0
• 运维系统升级
• FAILED
2012H2~2014H1
• Matrix-2.0
• Normandy立项
• 推广、稳定期
• 大搜、社区、商搜
2014H2~2015H1
• Matrix-3.0
• Normandy推广
• 大规模混布
• 覆盖率>50%
2015H2~Now
• PaaS组件建设
• Solaria
• 预结算、权限
• 系统整合

THANKS!

Baidu 百度