



# Matrix集群操作系统

百度-INF-吕斯哲

# About

---



招聘: [inf-jobs@baidu.com](mailto:inf-jobs@baidu.com)

# 演讲内容

---

## Matrix

- 背景
- 定位

## 宏观结构

- 分层结构
- 关键概念

## 容器与混布

- 应用部署
- 隔离、避让
- 调度、超发

## 工程实施

- 系统工程
- 推进思路

Matrix是什么？



# Matrix

百度业务

搜索

广告

移动

社区

开放云

PAAS

在线

离线

CI

....

Matrix

统一资源池

服务器集群



# 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

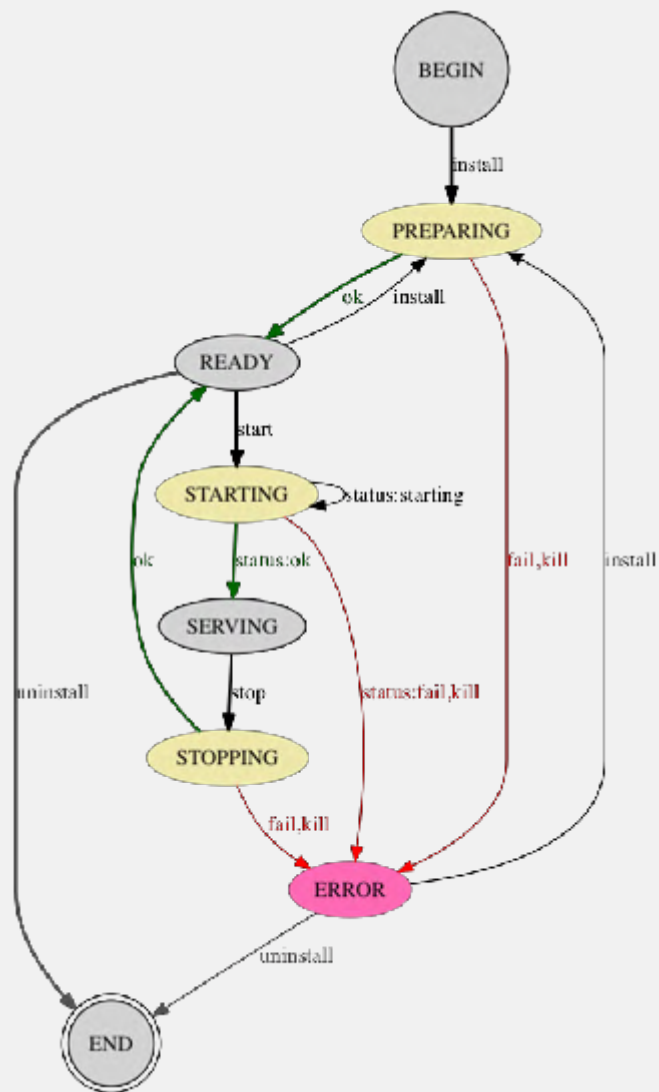
虚拟化战略

# Matrix分层架构





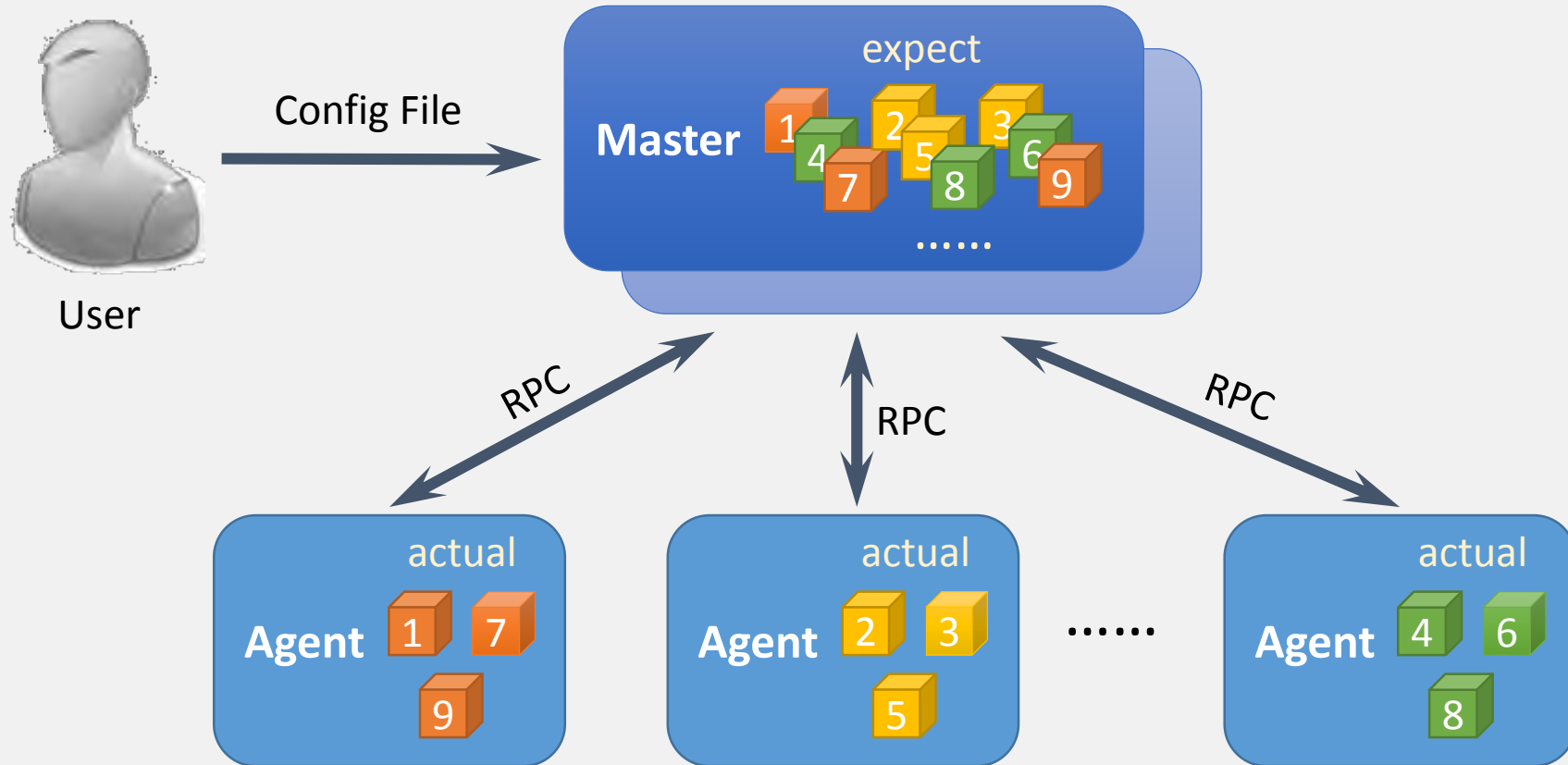
# 原理: 状态机设计



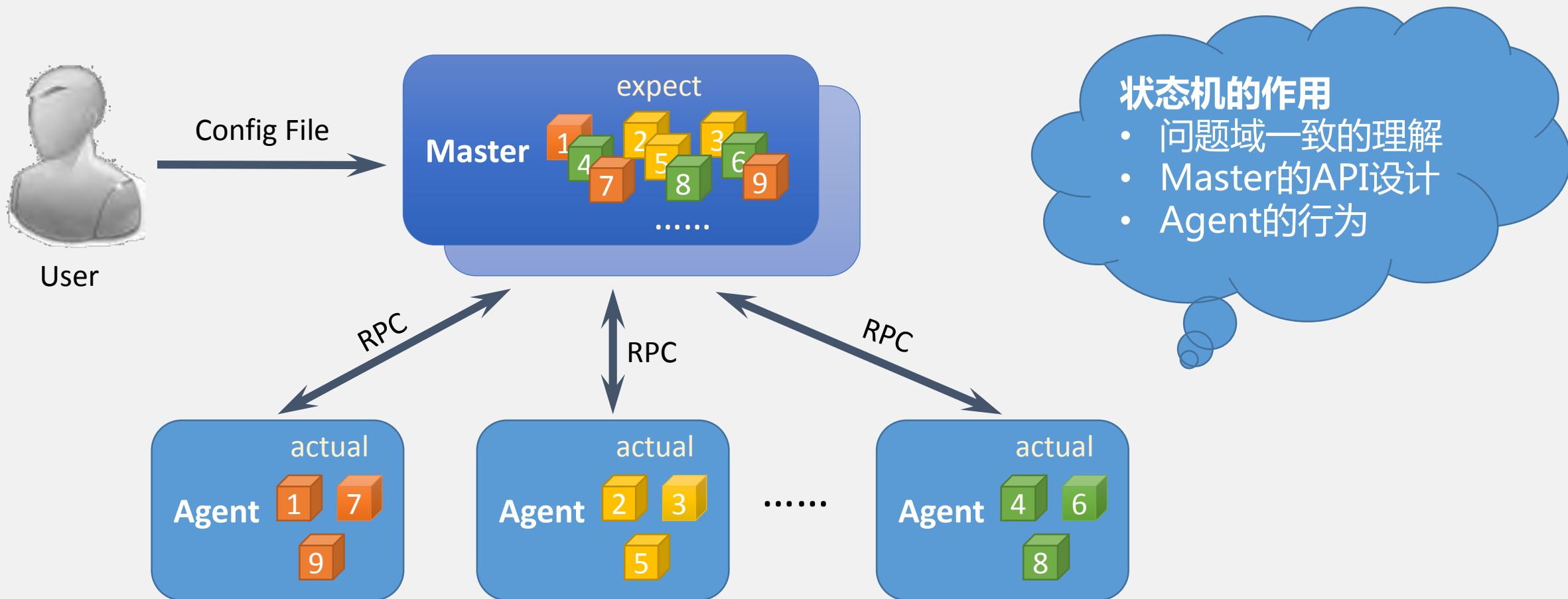
状态转移表

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

# 原理: 状态机设计



# 原理: 状态机设计



# 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

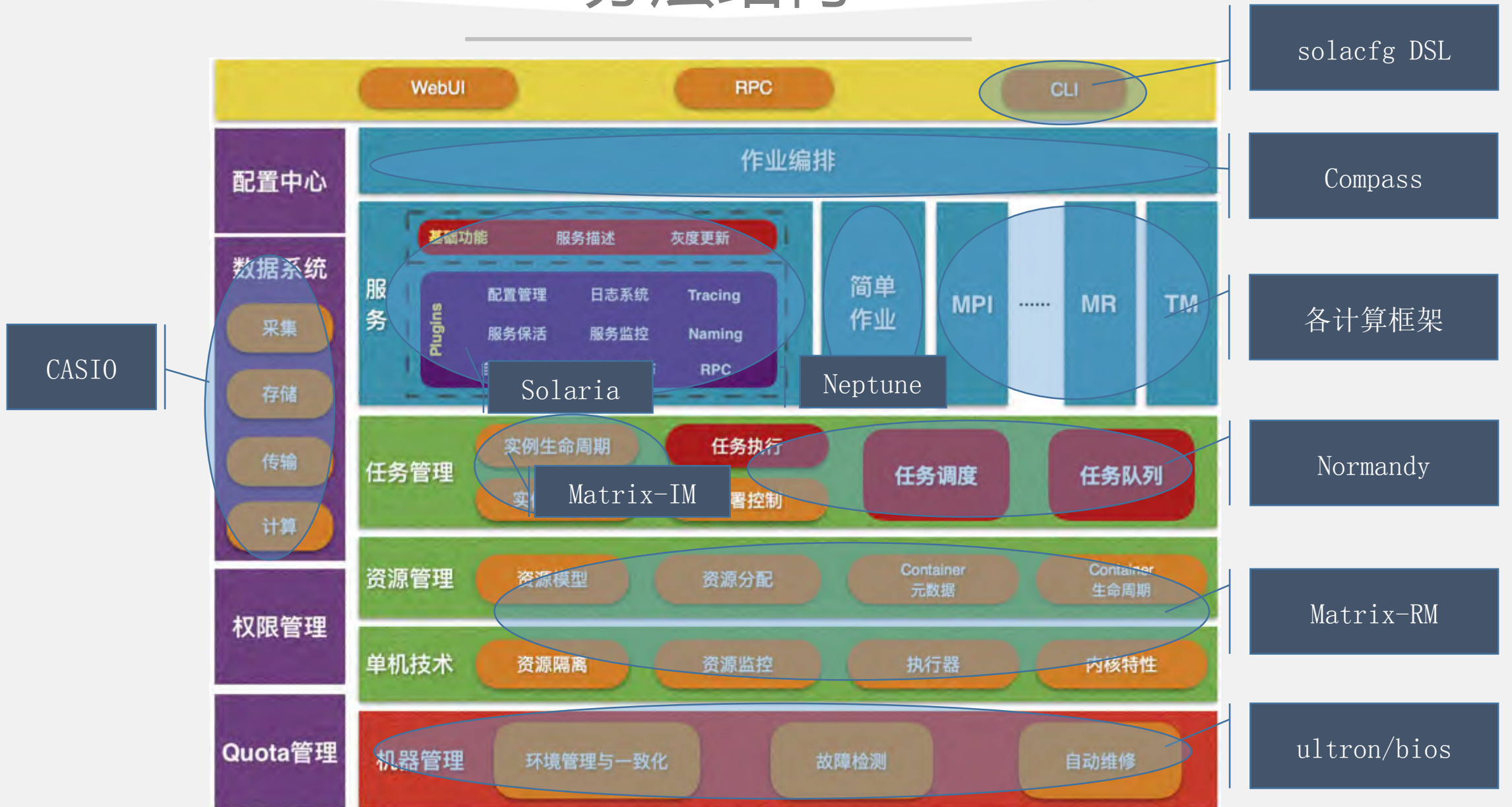
- 自结束“进程”
- 离线作业类

Solaria/Neptune

# 分层结构



# 分层结构





# 分层结构



- 分配资源 : Container

```
buf = malloc(...)
```

- 实例部署 : Instance

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

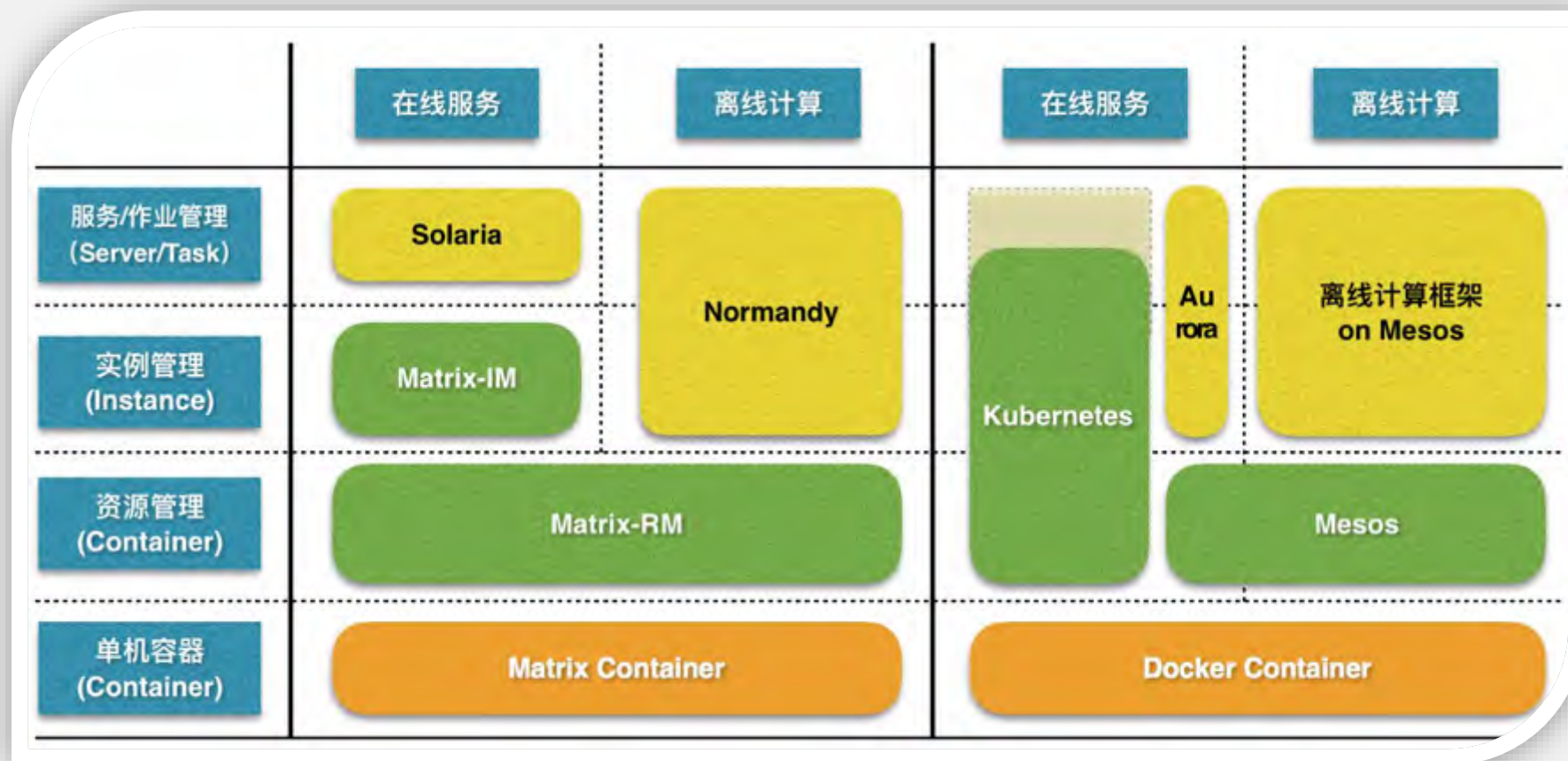
- 模型分化

- long-running: service/server

- routine: job/command

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

# 社区对比





# 容器与混布



# 应用部署

- **Docker: Container & Image**

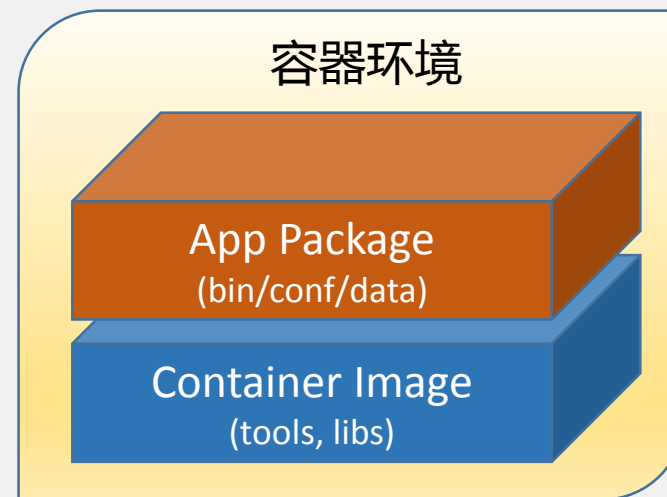
- **Matrix :**

- 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

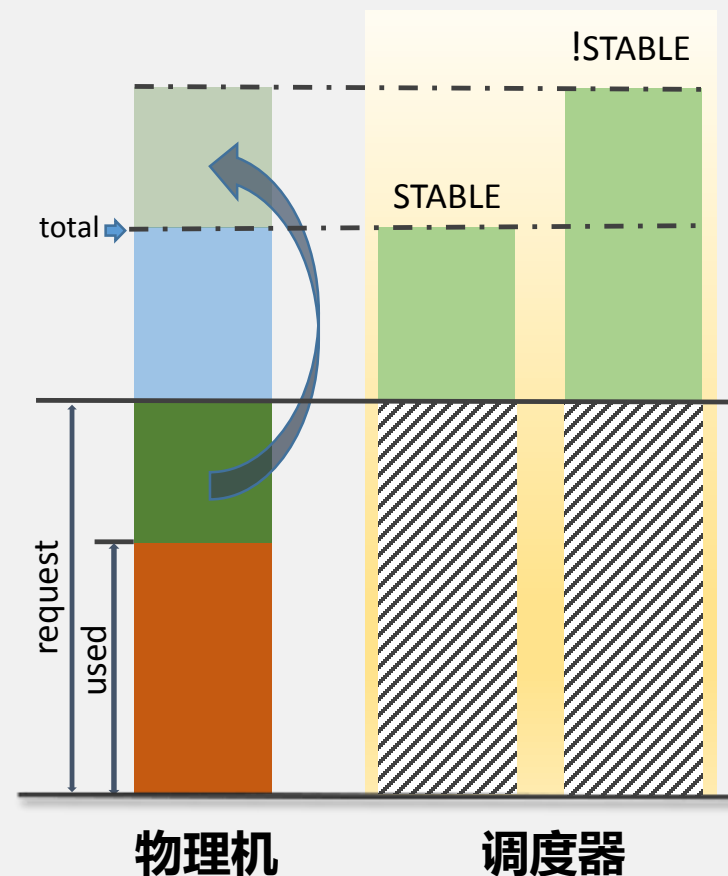
# 优先级与超发

## 优先级

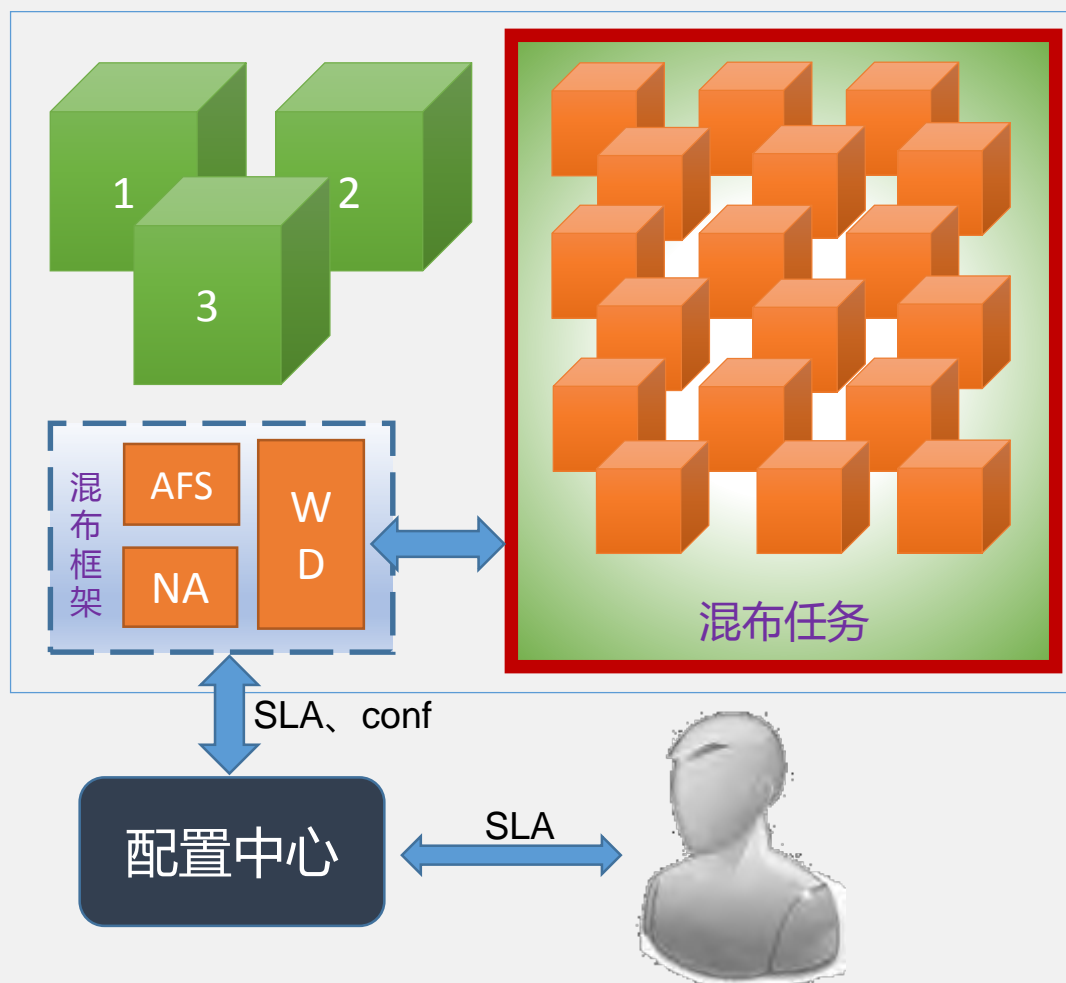
- STABLE > NORMAL > BESTEFFORT
- 单机：可否抢占
- 集群：是否有Quota

## 超发

- 用户行为： $\text{request} - \text{used} > 0$
- 超发： $\text{total}' = \text{total} + \text{sum}(\text{request} - \text{used})$
- 约束： $\text{sum}(\text{STABLE}) < \text{total}$



# 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







THANKS!