

QCon 全球软件开发大会 【北京站】2016

分布式应用无银弹

分布式应用架构核心要素的设计方法探讨

董 健

1

互联网应用/企业应用那些事

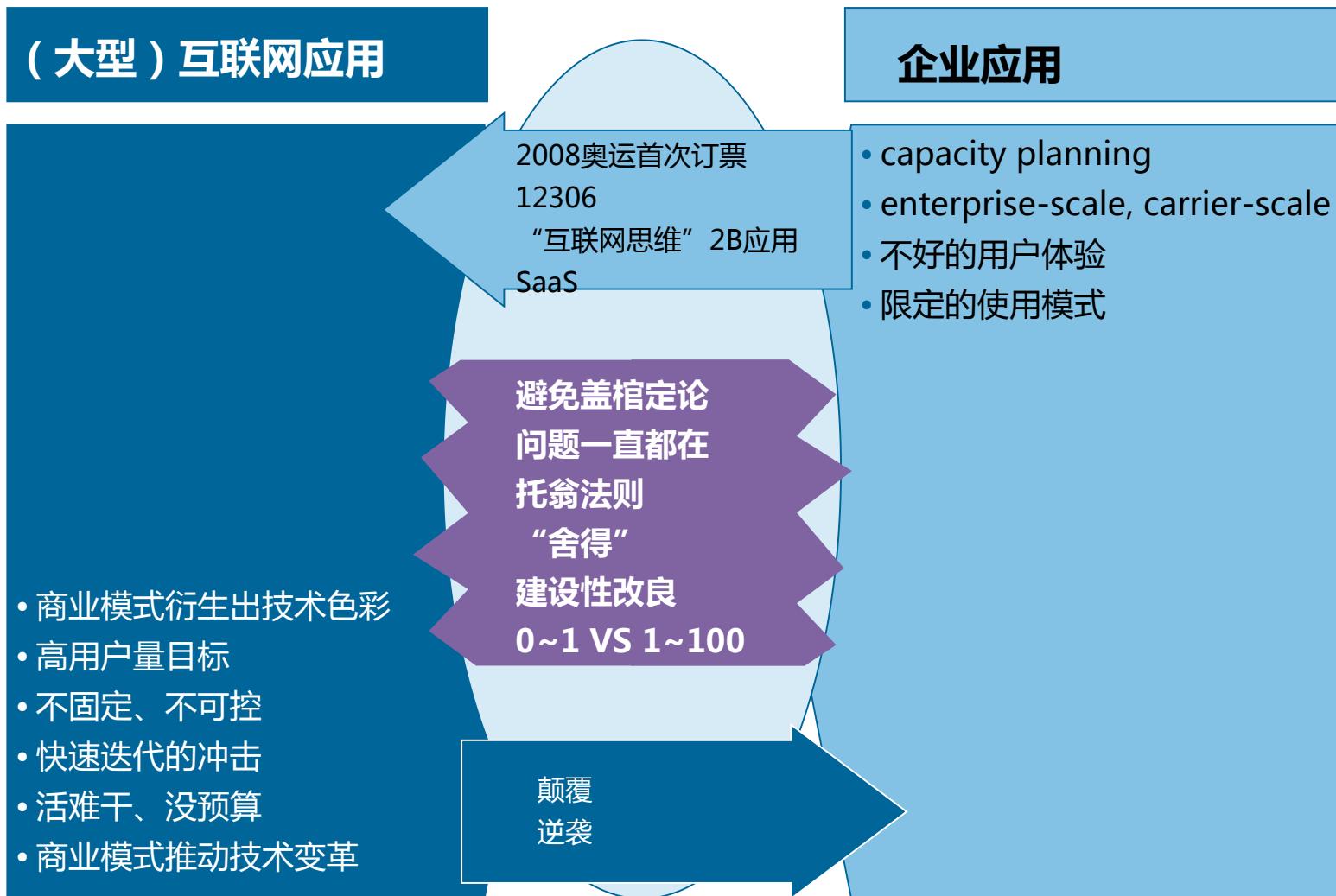
2

分布式为弹性应用解困

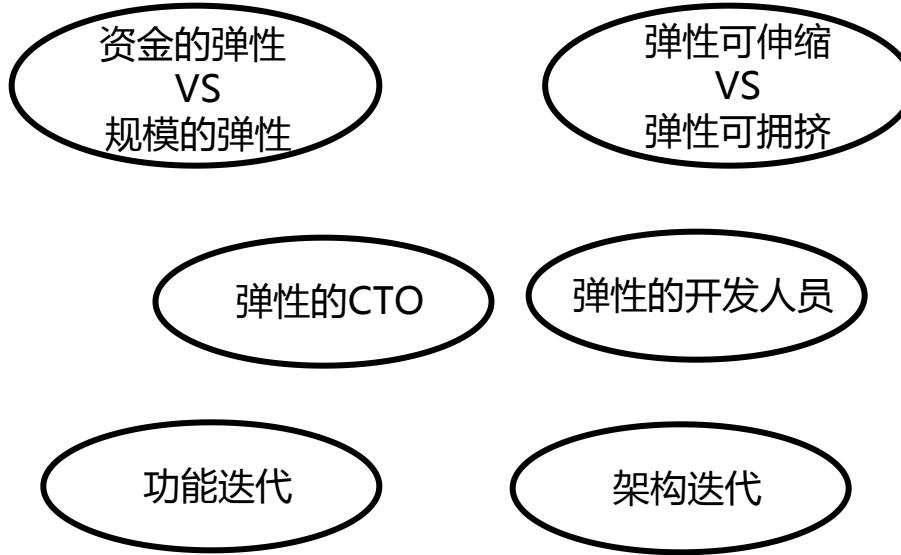
3

分布式平台的核心要素设计剖析

(大型) 互联网应用 VS 企业应用



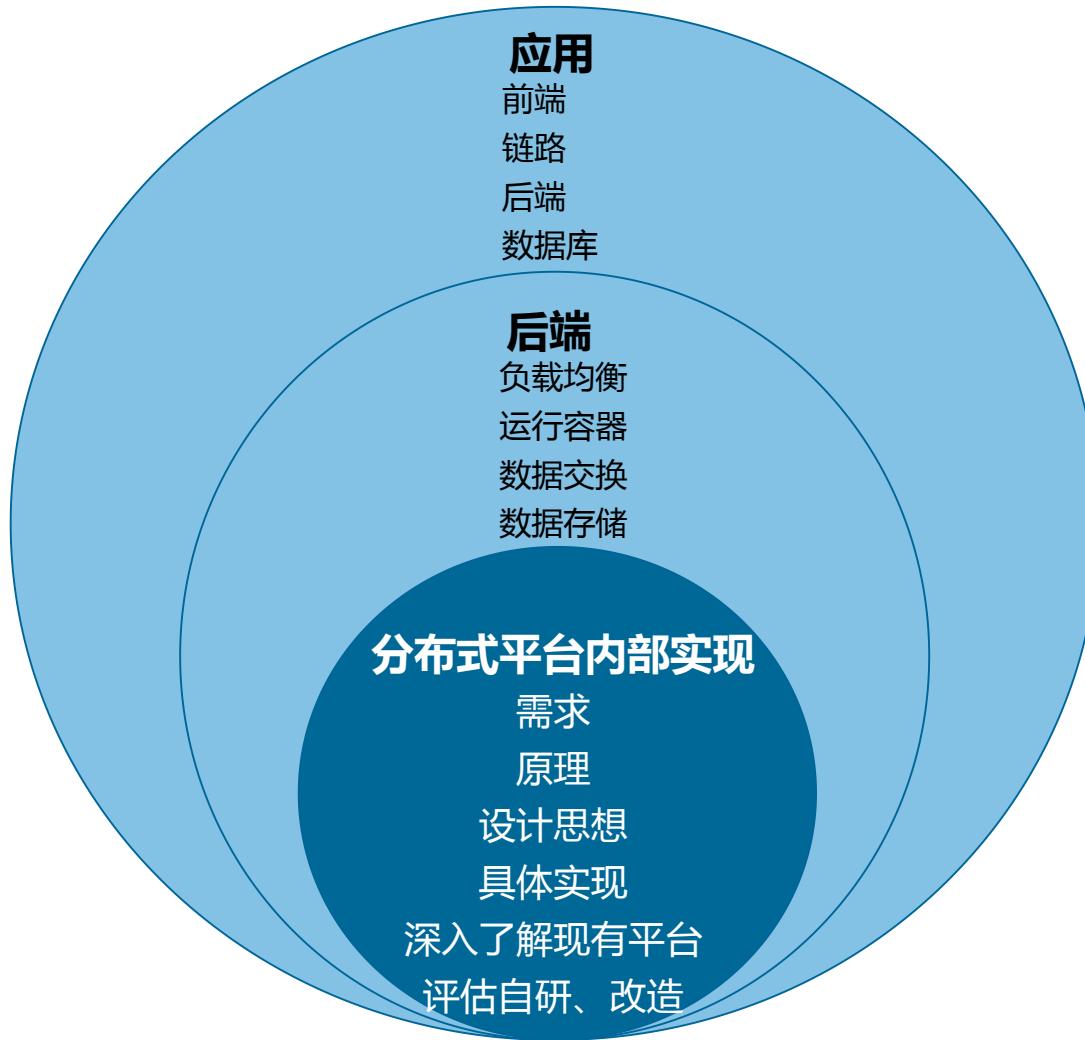
弹性应用



弹性架构

完善、灵巧
单体高效
值得尊敬的兼容性
提前布局

弹性应用的范畴



1

互联网应用/企业应用那些事

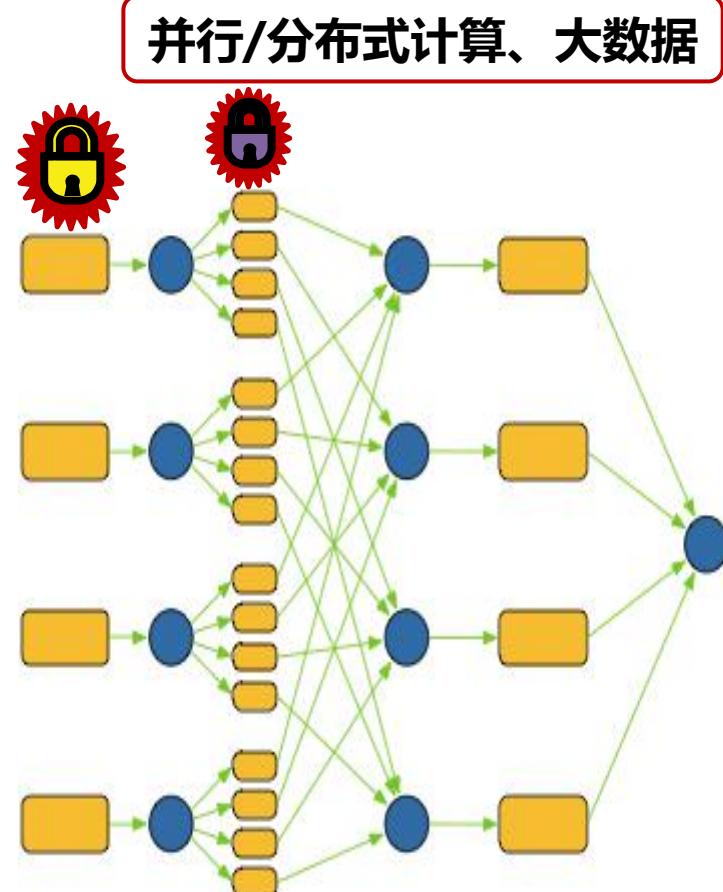
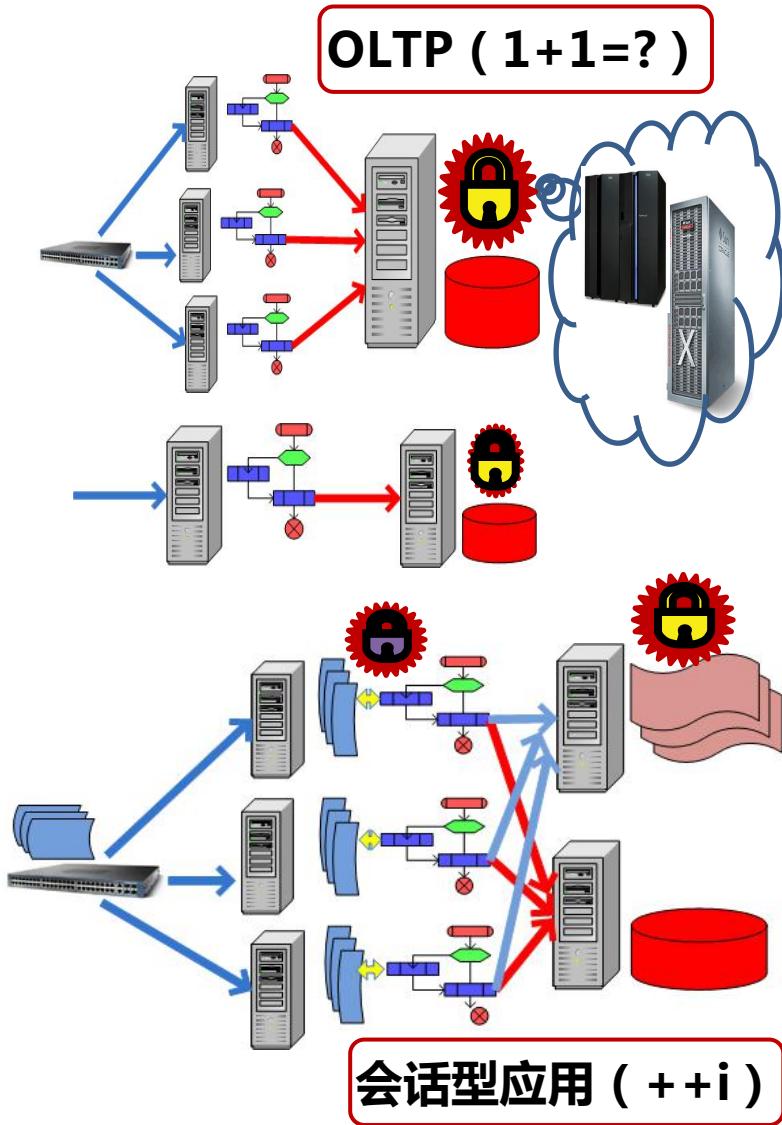
2

分布式为弹性应用解困

3

分布式平台的核心要素设计剖析

弹性应用面临的困境



困境中的答案



理想的分布式平台该做什么？

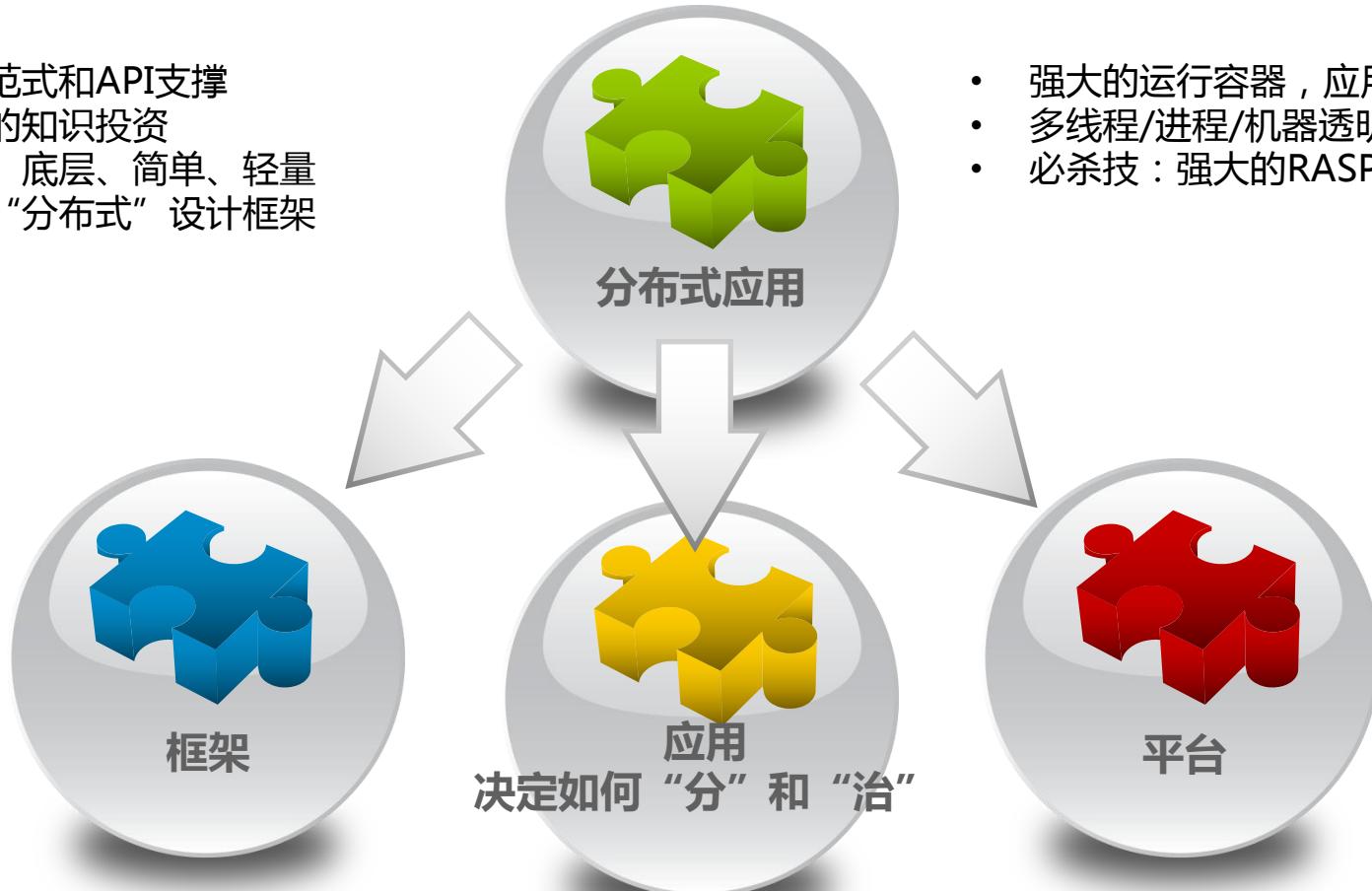
“最高深的技术是那些令人无法察觉的技术，这些技术不停地把他们自己编织进日常生活，直到你无从发现为止”
——Mark Weiser

开发态

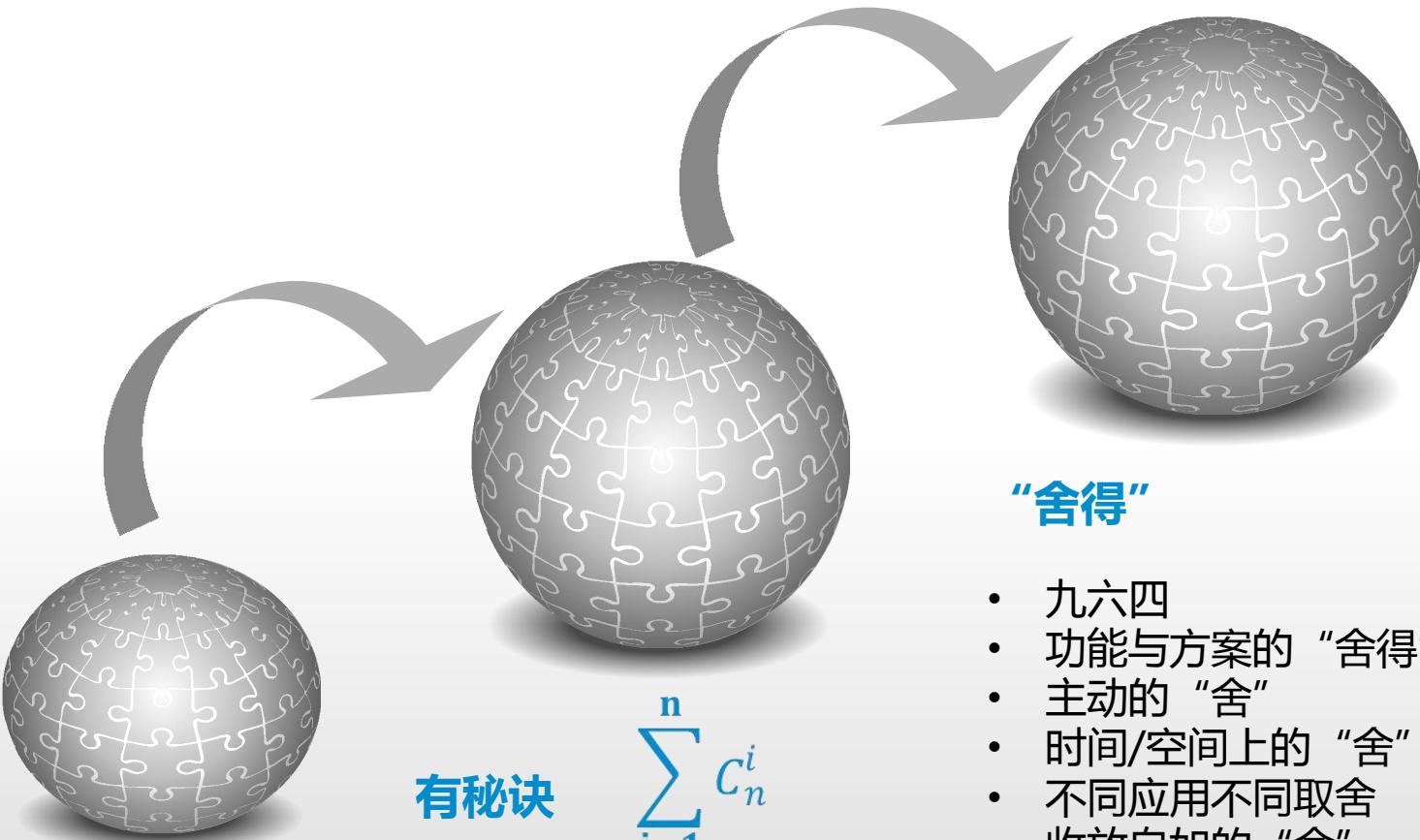
- 编程范式和API支撑
- 最少的知识投资
- 通用、底层、简单、轻量
- 杜绝“分布式”设计框架

运行态

- 强大的运行容器，应用透明
- 多线程/进程/机器透明并发
- 必杀技：强大的RASP



分布式平台设计的核心方法论



$$\sum_{i=1}^n C_n^i$$

- 功能和RASP的N
- 加法变乘法

“舍得”

- 九六四
- 功能与方案的“舍得”
- 主动的“舍”
- 时间/空间上的“舍”
- 不同应用不同取舍
- 收放自如的“舍”
- 不光是智商的问题
- “改变世界”的发明？

1

互联网应用/企业应用那些事

2

分布式为弹性应用解困

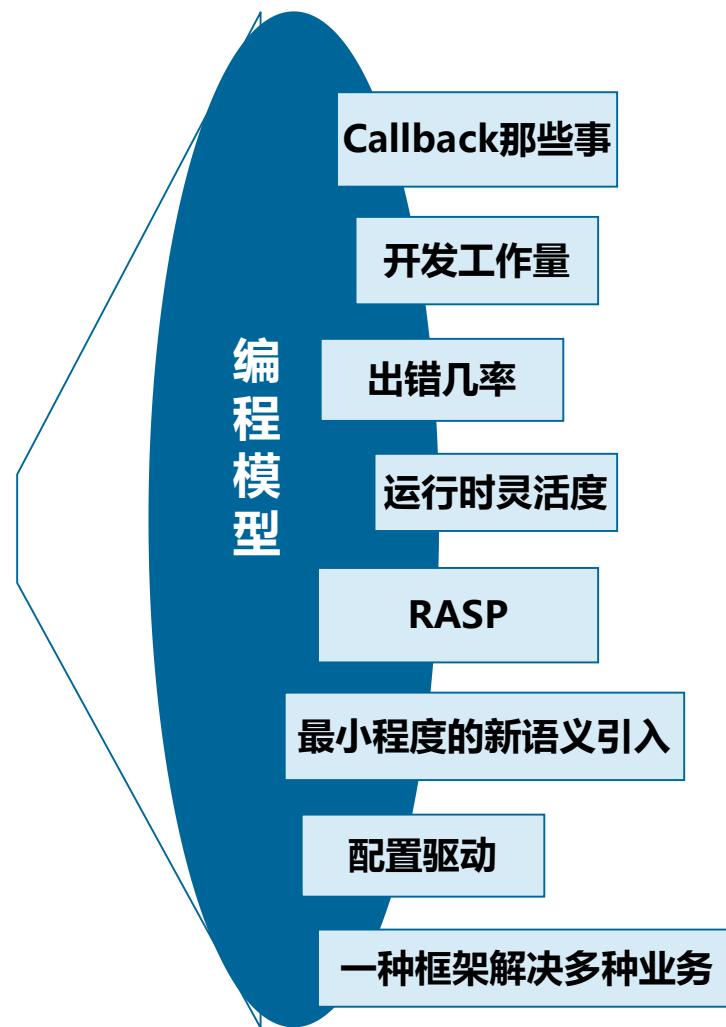
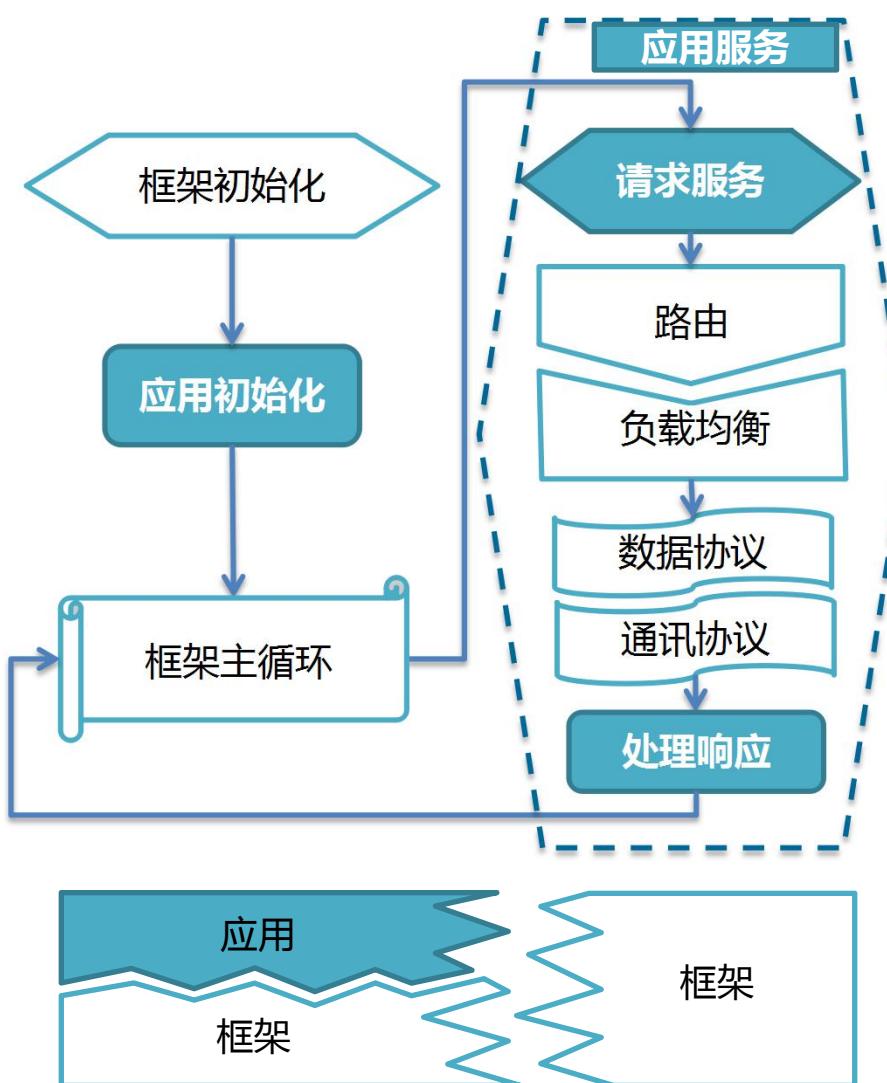
3

分布式平台的核心要素设计剖析

开发态（框架）的核心要素



便捷的编程模型



典型的分布式编程模型

请求式分布式应用

平台和应用有条件隔离

- HTTP Session
- 容器+运行时 VS 统一容器

请求成本高、并发弱

必备负载均衡

请求间同步需第三方支持

适合OLTP

Map-Reduce

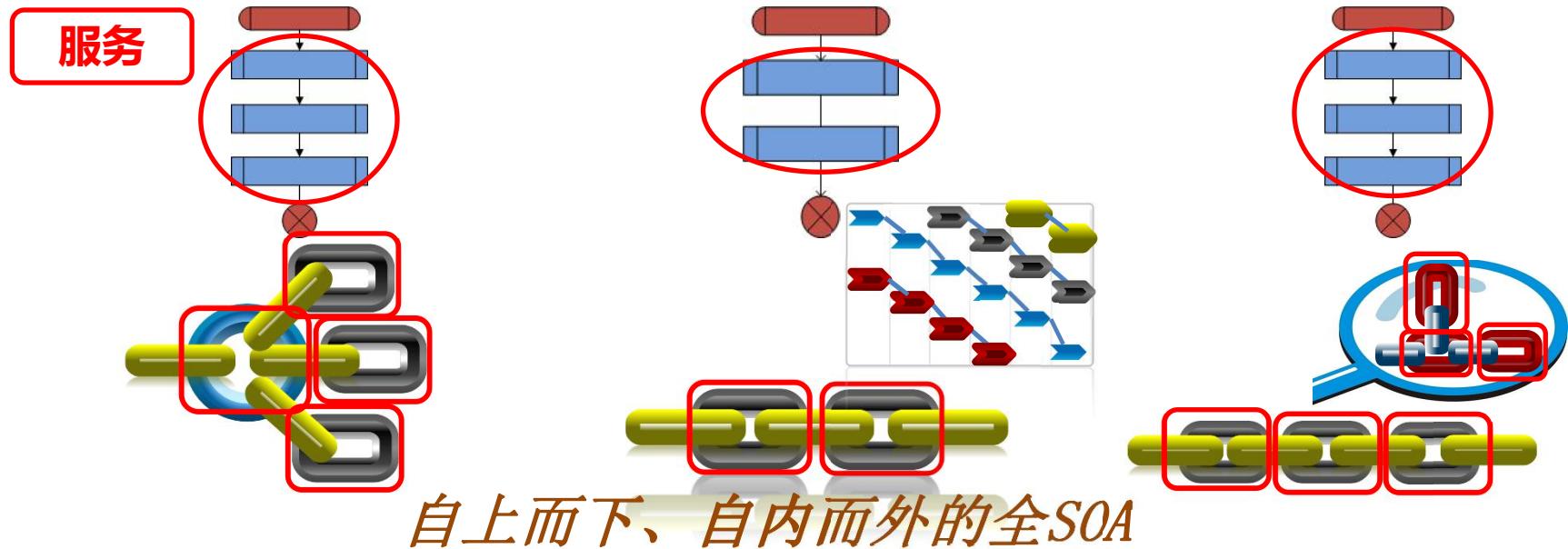
Buzzword

分布式编程框架完全入侵应用
逆向思维，本末倒置

合适时机出现的伟大的“廉价”创新
不等于分布式，不是银弹

简单粗暴的“分而治之”
适合批处理

服务型分布式计算框架——分布式遇上SOA



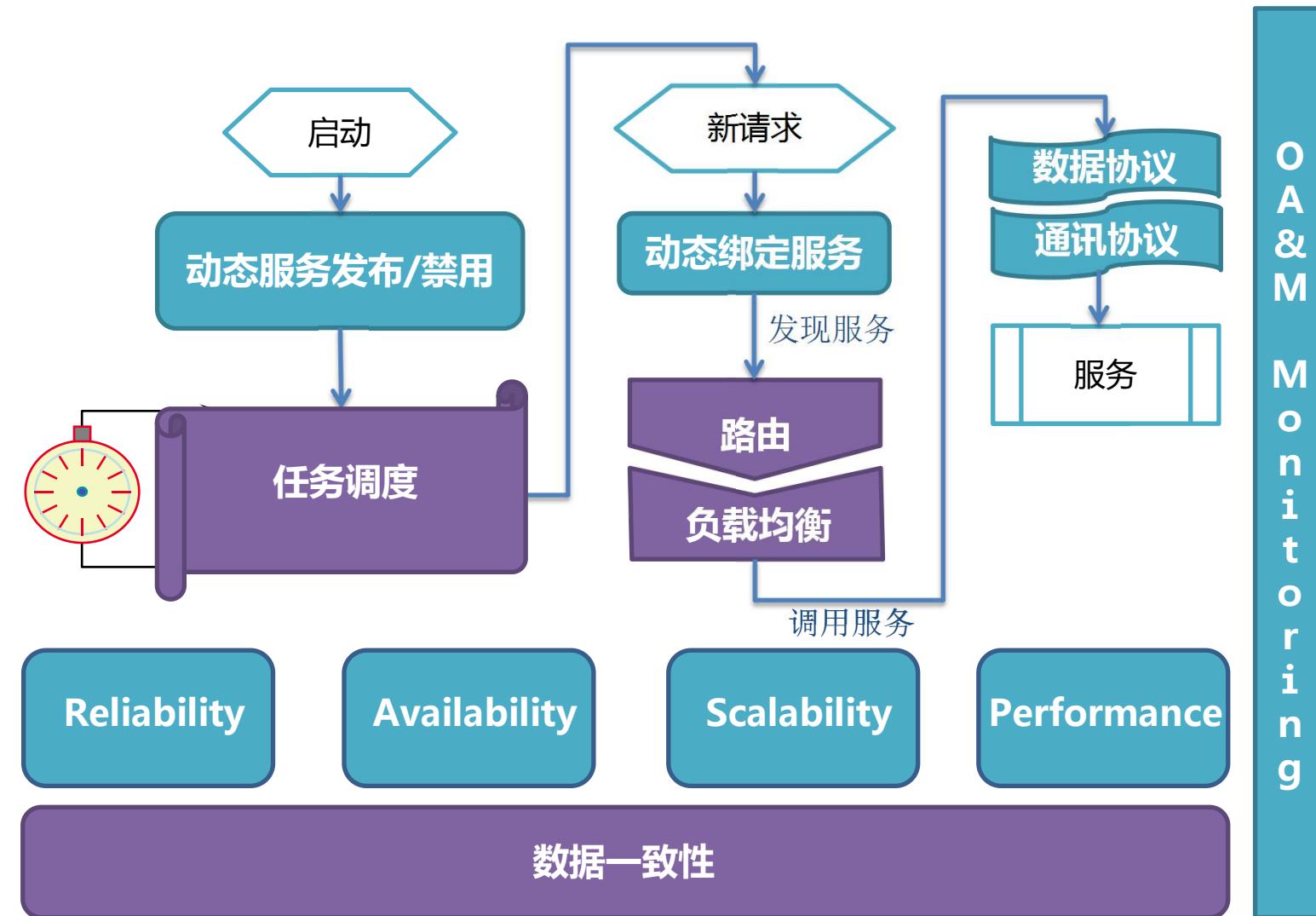
服务化的分布式应用开发

便捷开发
重要而简单的异步
顺向思维，业务导向
框架最小程度入侵
开发态运行态隔离

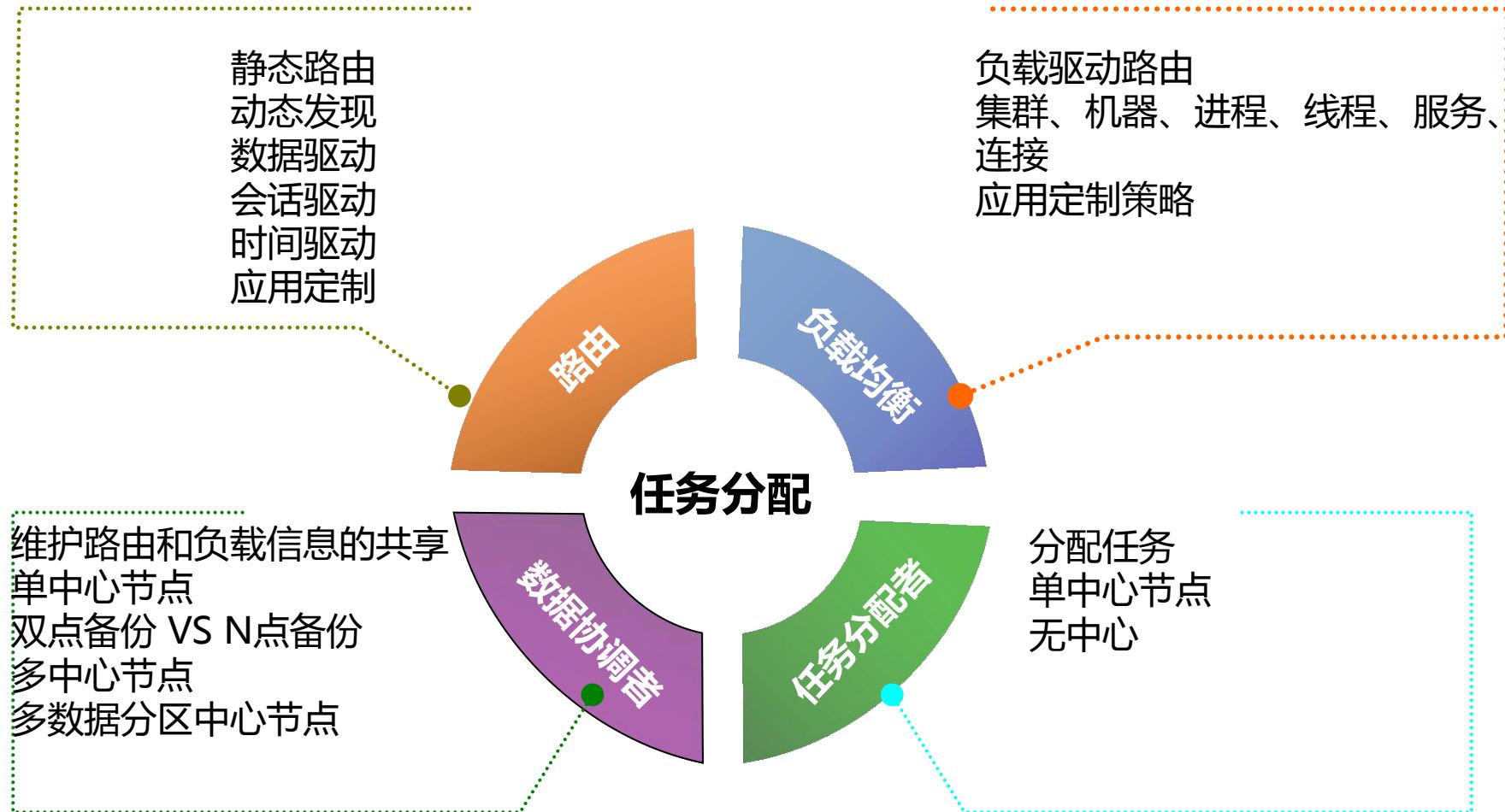
简单、灵活、高效、个性化的分布式应用架构

专注业务
内外统一，便捷发布
灵活的应用架构
高效的运行效率
个性化的分布式模型

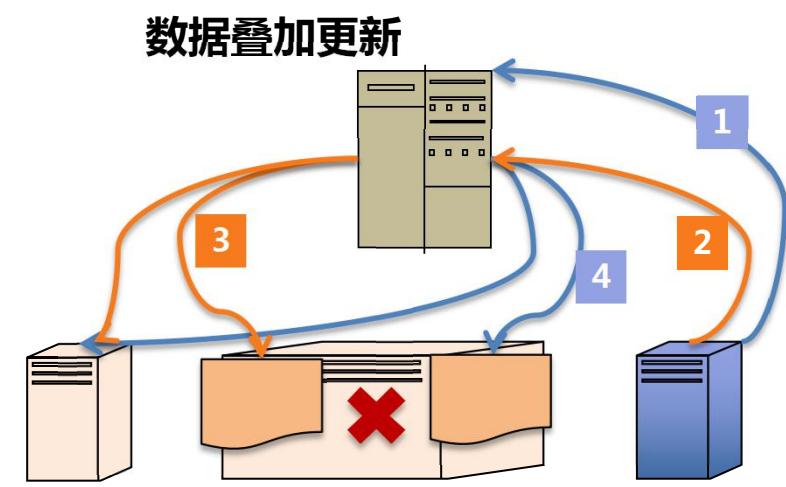
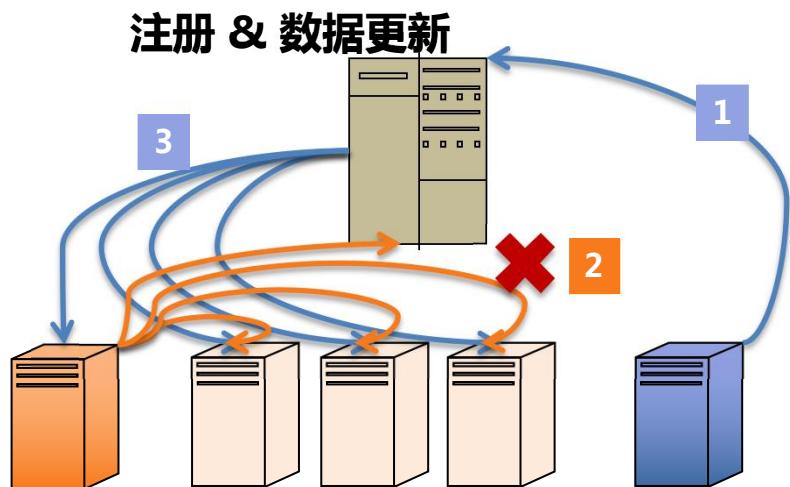
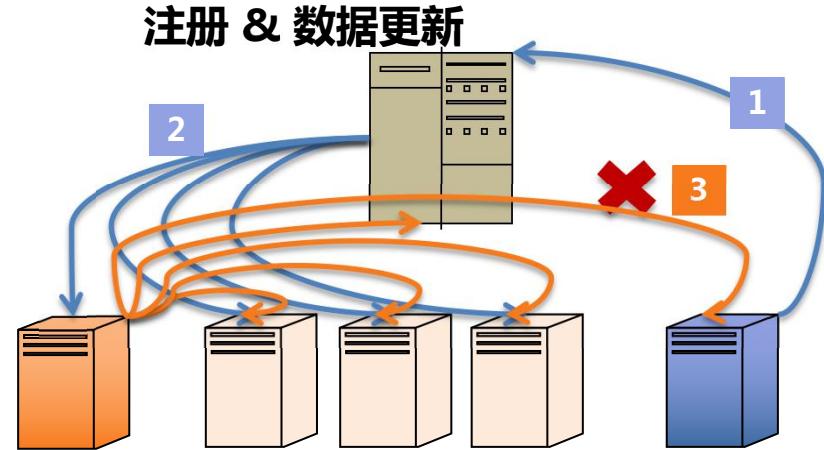
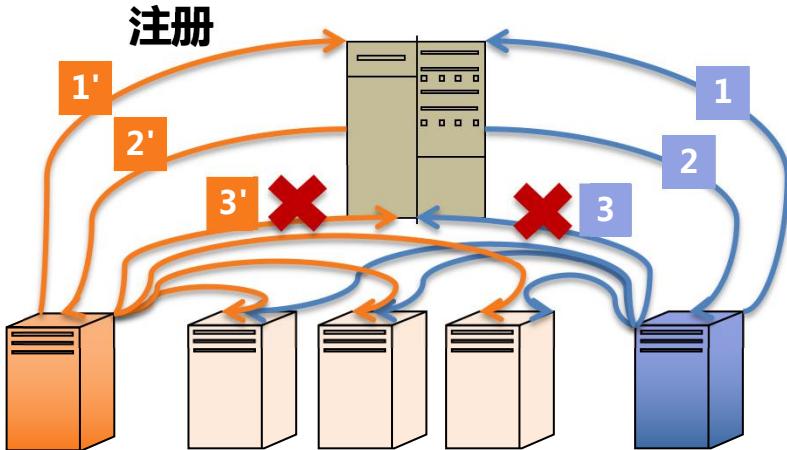
运行态（平台）的核心要素



任务分配：路由和负载均衡



典型的数据一致性问题



数据的一致性



任务调度

任务粒度

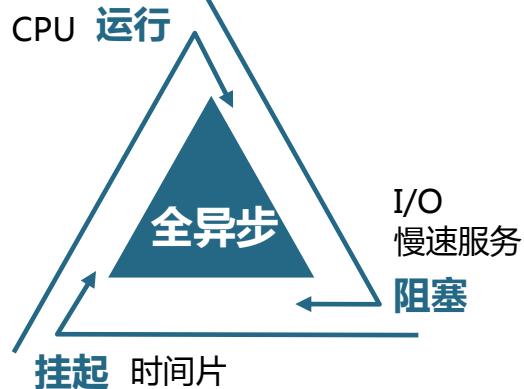
编程模型决定
调度灵活性
系统性能



任务虚拟化

调度器

依赖于任务粒度
尽力运算
CPU 100%
操作系统内核



服务型分布式计算框架

计算资源/ 能力虚拟化

计算资源 统一调度



1

互联网应用/企业应用那些事

2

分布式为弹性应用解困

3

分布式平台的核心要素设计剖析

THANKS!

董 健

博晓通

email: dongjian@inter3i.com

Tel: +86 18611886922