

# 京东MySQL数据库Docker化最佳实践

刘风才



# 目录 CONTENTS

- 一 . 京东Docker技术发展历程
- 二 . MySQL数据库为何要Docker化
- 三 . MySQL数据库Docker化前准备工作
- 四 . 遇到的问题与解决方案
- 五 . 总结与展望

## Step 1: 初出茅庐

### 探索规划

京东从2013年开始规划虚拟化平台项目，组建团队迅速掌握OpenStack核心代码，采用OpenStack+KVM架构

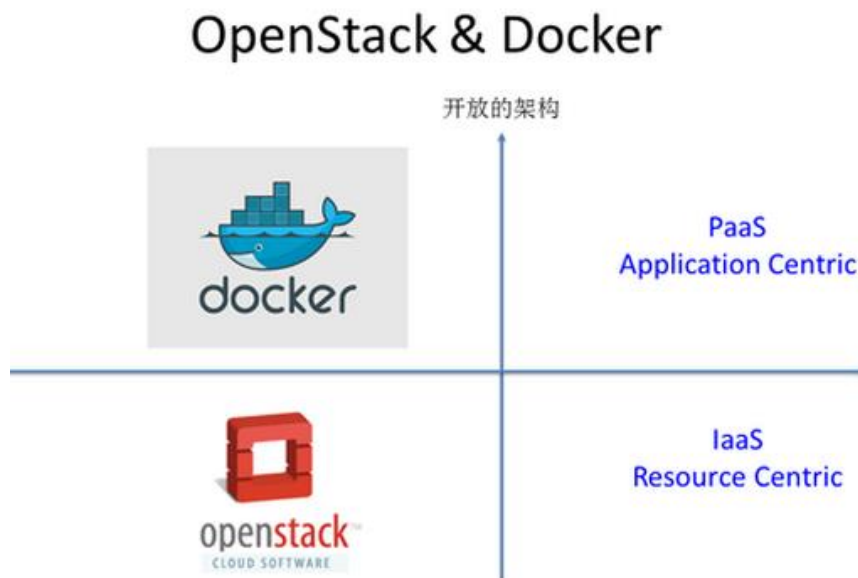
### 性能瓶颈

业务要求tp99保持在40 ms以内，但OpenStack在并发非常大的时候，无法满足性能要求，始终无法达到物理机的水平

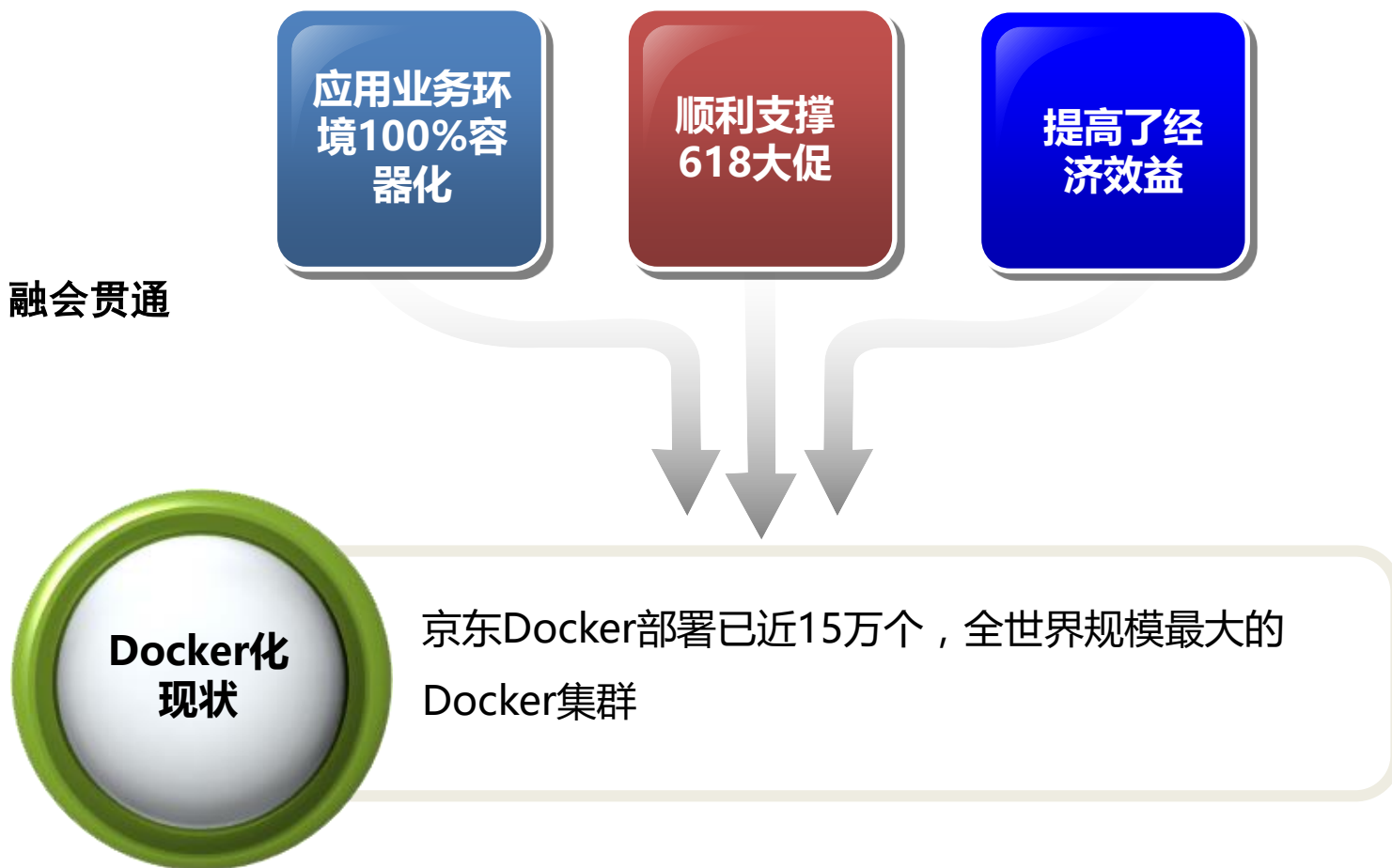
## Step 2: 小试牛刀

2014年9月，Docker技术进入京东，容器技术看到曙光

京东第一代容器引擎平台JDOS1.0(JD DataCenter OS)上线



## Step 3: 融会贯通



## 选择Docker技术收益

快速部署

采用Docker后，  
数据库实例创建  
时间只需要1分钟

动态扩容

Docker 可以在线  
扩容，方便快捷，  
更安全

资源利用率

解决多库合用或  
多实例部署都无  
法避免物理资源  
竞争、相互影响  
问题

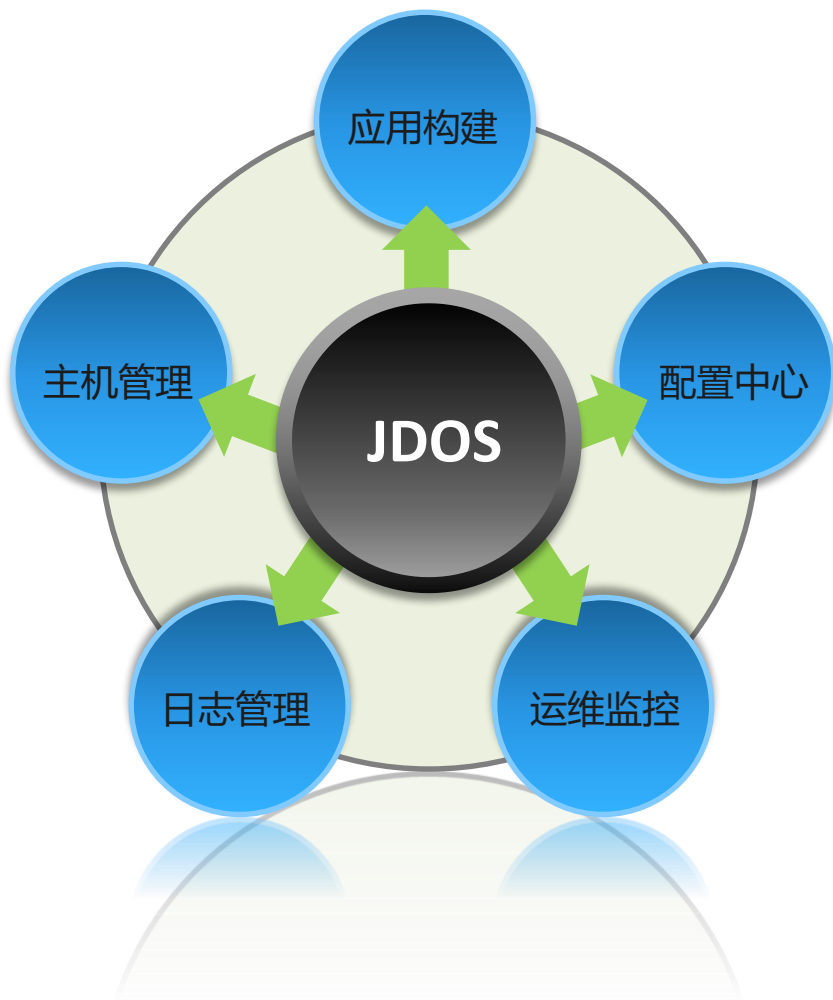
降低成本

节约了服务器资  
源、机架位资源  
等，提升经济效  
益

Docker技术

京东在Docker技  
术上的应用已经  
成熟，应用100%  
容器化，并经过  
618、双11大促的  
考验

## JDOS个性化开发-管理页面开发



从资源池的管理，到容器的申请、计算、调度、分配以及后续的监控运维等全生命周期管理

## JDOS个性化开发

### 分配算法

#### 标准原则

根据宿主机的状态（是否正常可用）、宿主机上的可用资源（cpu、内存、磁盘容量）是否满足申请的容器规格，将集群中所有的服务器都筛选完后，会计算服务器的权值，从中选择最适合的服务器作为宿主机

#### 分区实例

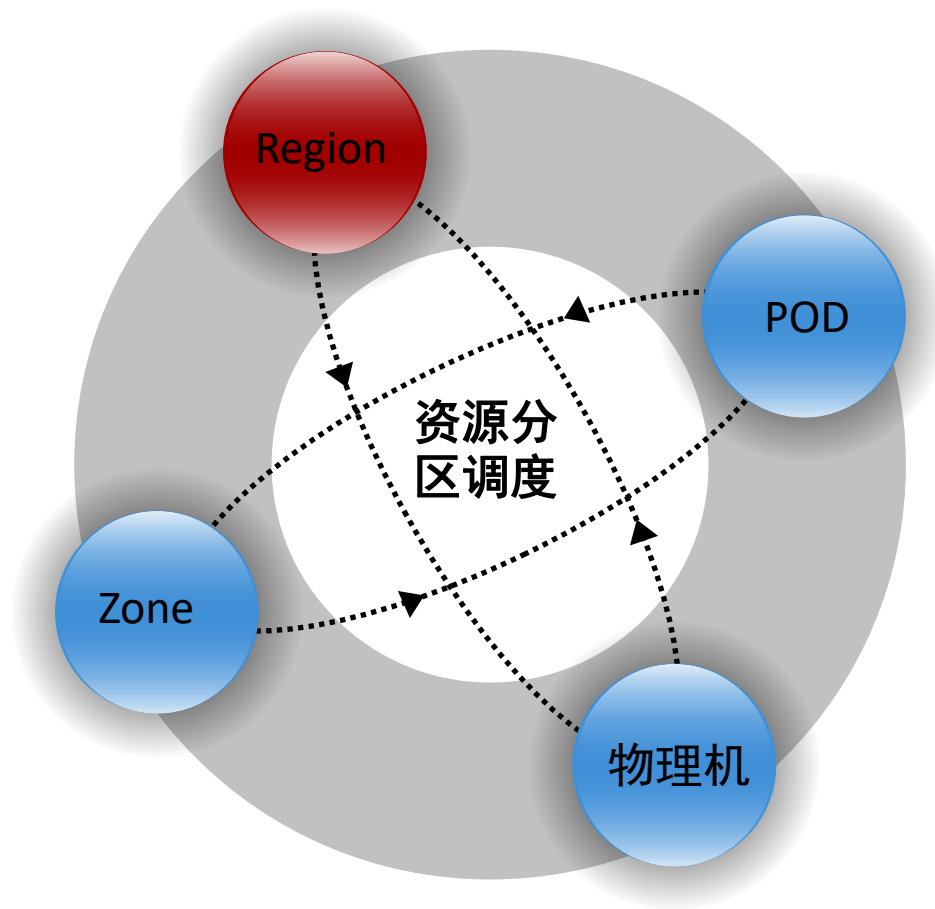
创建全新数据库集群实例如一主一从或者多从，JDOS调度器会把这2个或者多个MySQL容器实例分布到不同的机架

#### 多从服务

多从库需求且从库是高性能读，JDOS调度器会根据 `performance_slave=True` 标记，尽可能把从库创建到网络繁忙度较小的机架，并与主在相同的物理POD，甚至同一个Tor交换机下面的不同物理机上

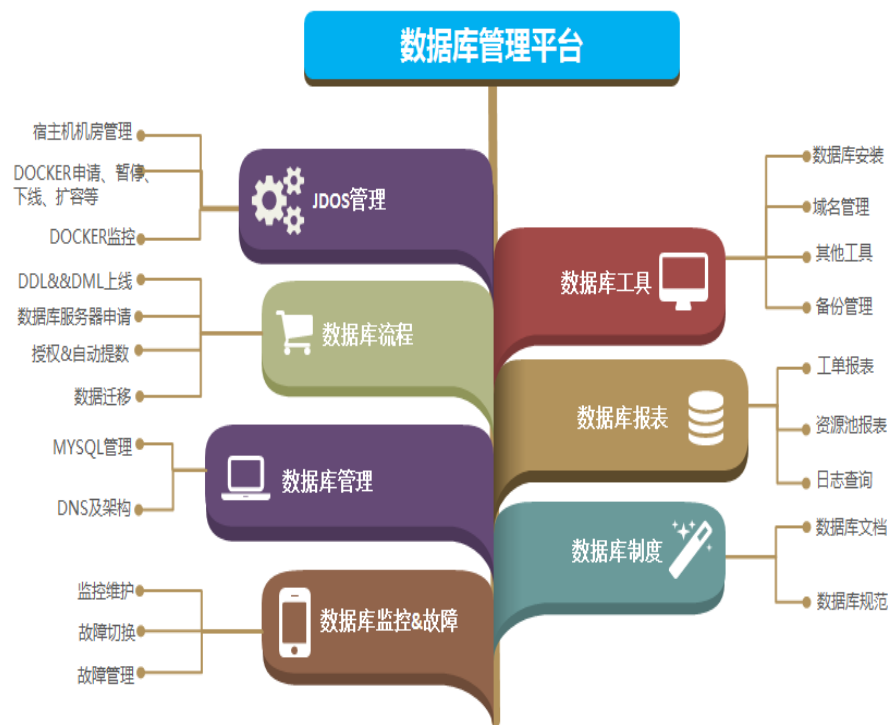


## JDOS个性化开发-磁盘调度算法优化



调度算法灵活可配确保业务特点与计算硬件环境特点结合适配。如果需要高io的，调度算法会权重选择负载最低的宿主机

## DBS管理平台开发



- 根据宿主机IP查询该宿主机下所有Docker信息
- 根据Docker IP 查询宿主机IP 以及该宿主机下所有Docker信息

## Zabbix监控项优化

在Docker上我们无法通过os系统命令抓取系统负载情况，在Docker宿主机上专门程序进行计算存储到redis中，zabbix从redis中读取对应监控值

## OpenStack集群规模受限

### 遇到的问题

创建容器消息在MQ传输过程丢失，容器状态挂起，DB连接数过大，计算节点各种agent定时任务hang，部署升级无法核对升级结果

### 解决方案

设计目标单个集群1万台物理节点，实现一个python版本的RPC ( brood ) 框架，解除对MQ依赖。特别是依赖MQ操作DB的全部替代使用京东自研的python版本RPC框架，对数据库的全部操作均使用RPC自带支持的京东JIMDB ( 内存缓存集群 )

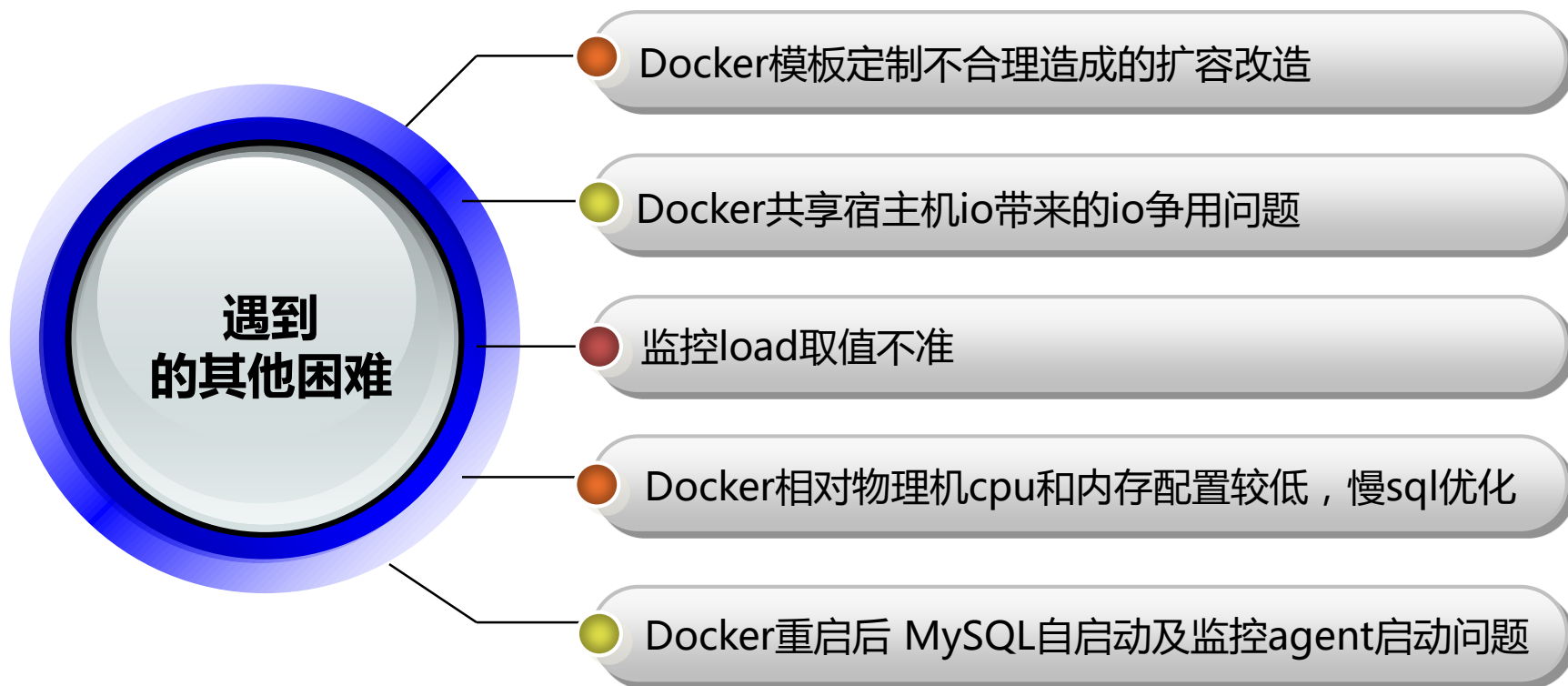
## 遇到的问题

规模大之后，遇到很多低概率但是实实在在发生了性能和稳定性问题，mac表满导致无法网络通信，UDP大报文硬塞，丢包，中断异常，系统slab集中回收性内存申请锁住时间过长

## 性能、稳定性问题

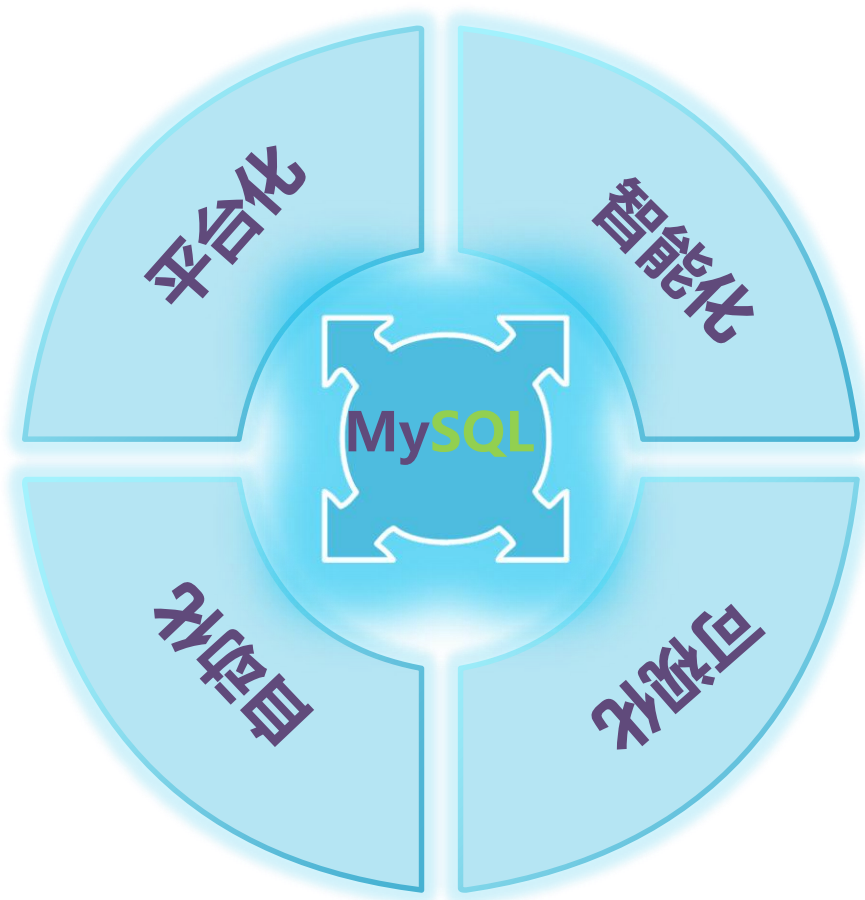
## 解决方案

问题的根源在于Linux kernel，意识到做容器就是做Linux kernel，即刻组建了Linux Kernel 团队，修改内核代码，最终解决并维护了京东自己的Linux Kernel分支



## 总结

目前生产环境中70%的MySQL数据库运行在Docker容器上。顺利支撑了两个618、一个双11大促，经住了大流量的考验。



谢谢大家！

