

Gdevops

Global DevOps Summit

全球敏捷运维峰会

维稳618和双11：
京东科技数据库海量运维的生态体系建设与实践

演讲人：高新刚



目录

海量运维概述

海量运维的高可用体系

海量资源管理

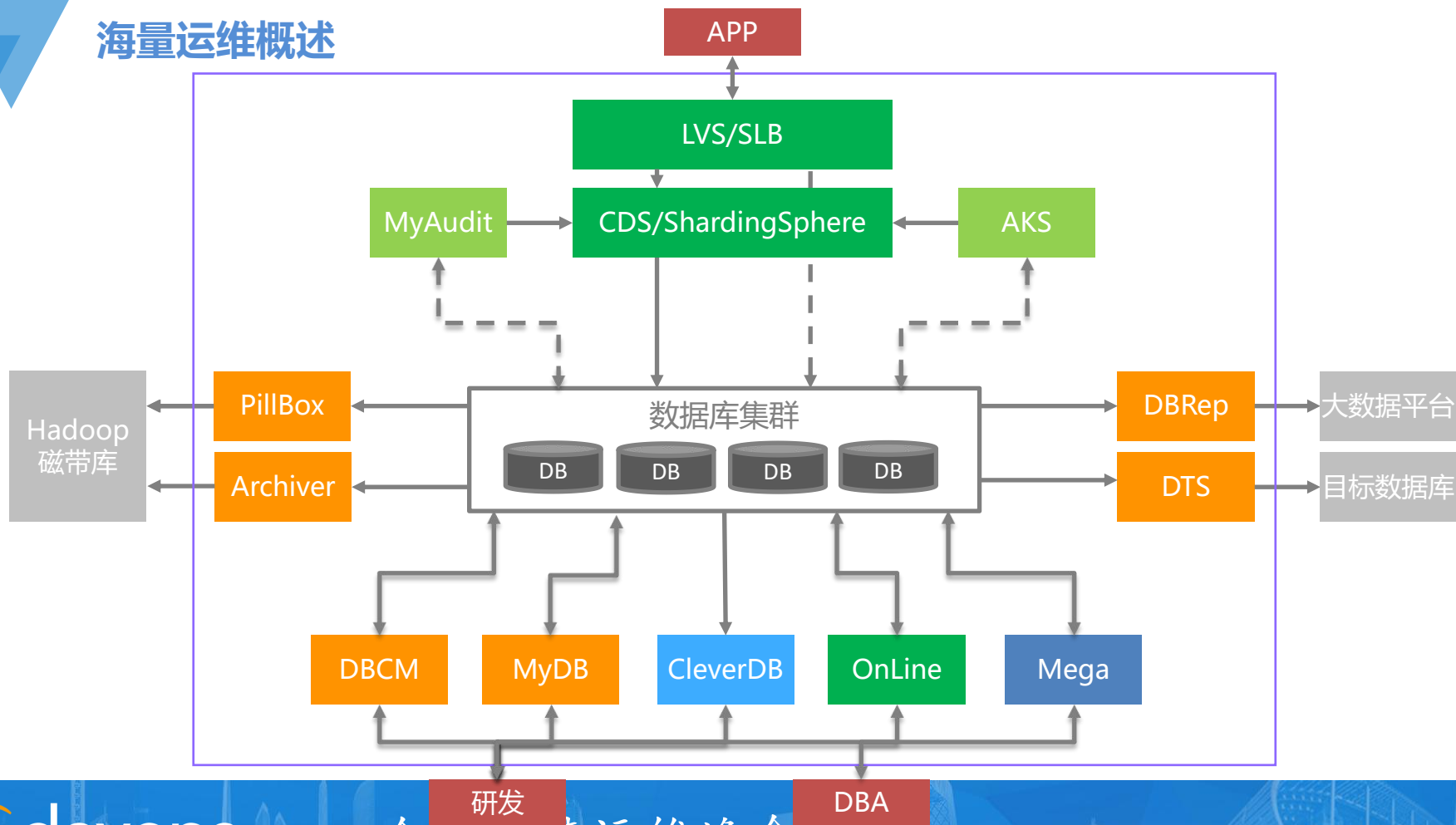
海量告警管理

大促备战分享





海量运维概述



A

优质的容灾服务质量

RTO < 10S
秒级监控
哨兵探测
高效流程快速切换

B

可靠的切换决策模型

自动识别异常高效监控
极端场景判别
故障自动决策避免误切

C

平台自动化管理

CMDB上下线一站式配置服务
进度状态、历史查询、主动通知
可插拔模块、系统融合
丰富的API接口
可视化信息统计和管理报表

D

良好的兼容性和适配性

不同环境：跨IDC、跨ZONE、跨主机
不同架构：读写分离、延迟同步、共享存储
不同中间件：SLB、LVS、CDS、DNS、PROXY

E

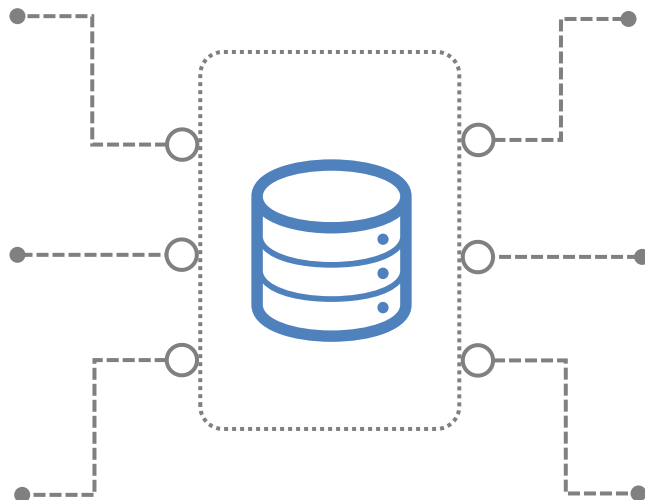
服务自身的可用性保障

容灾服务组件无状态
跨机房部署
配置库双活部署
服务多活无单点问题

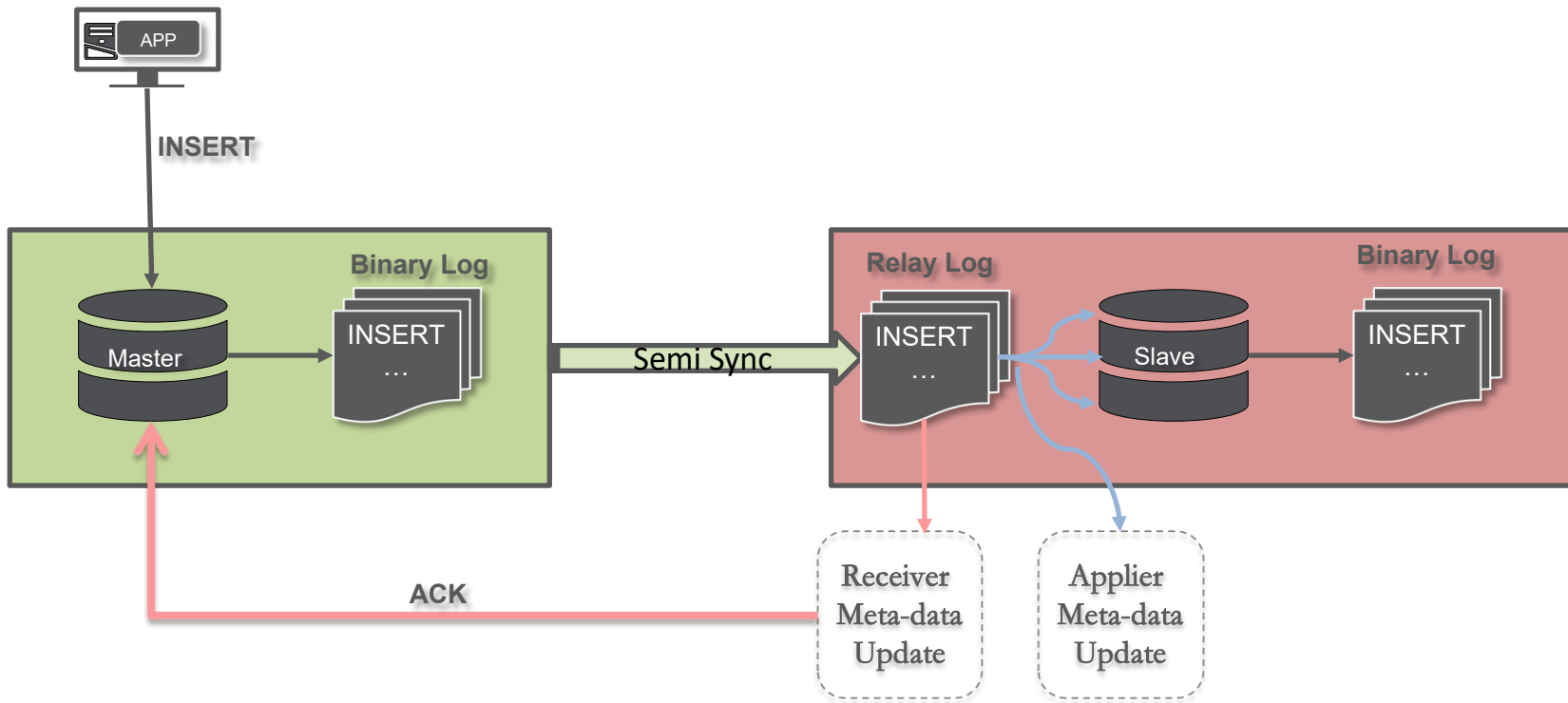
F

丰富的容灾类型

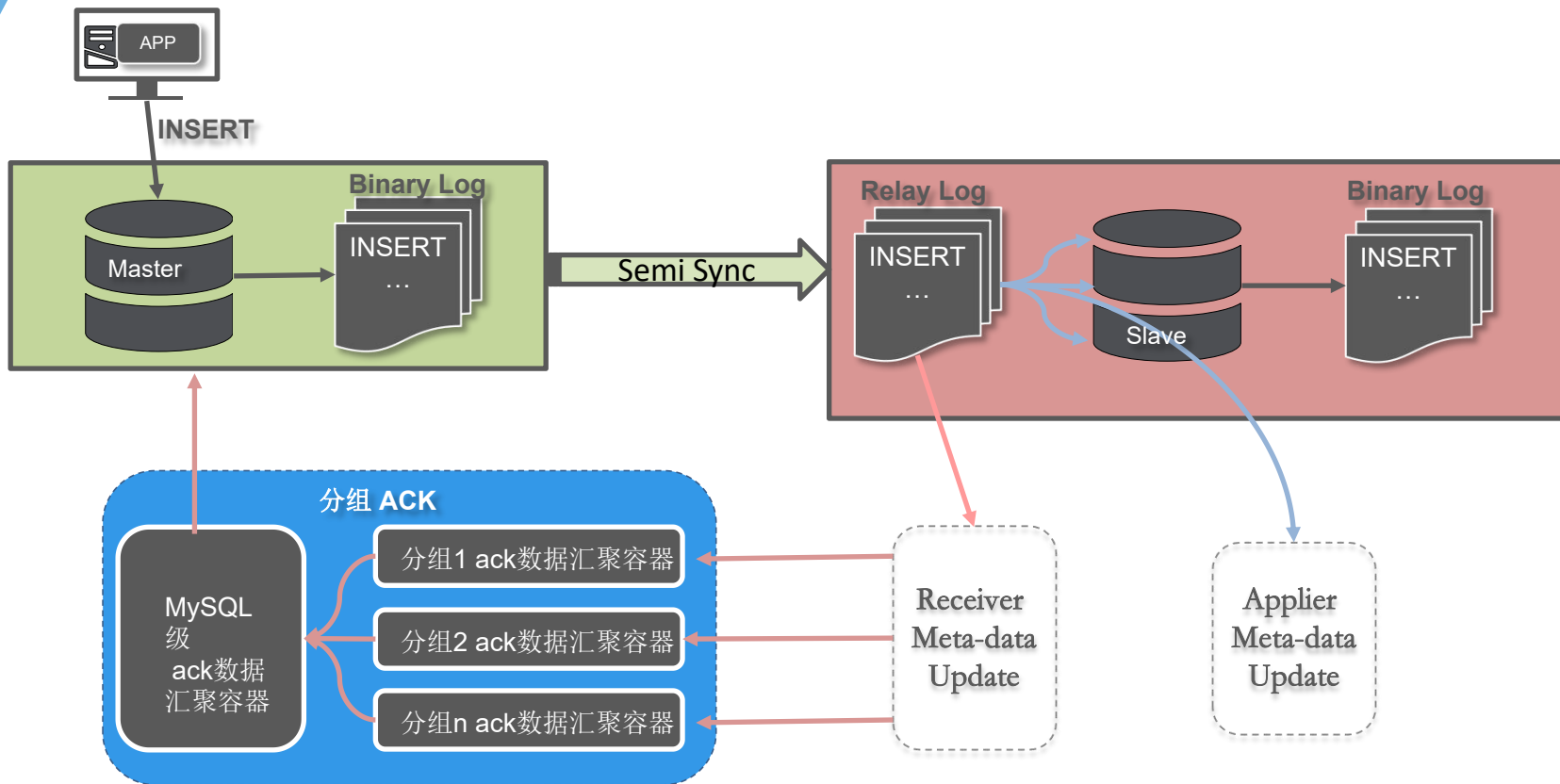
主库容灾、从库容灾
手动切换、自动切换、权重切换
Online switch & offline switch
批量切换、机房容灾



高可用体系的数据一致性保障



高可用体系的数据一致性保障



高可用体系的数据一致性保障

类别	主机房从库数量	同城机房从库数量	异地机房从库数量	机房级灾备ACK数量	地域级灾备ACK数量
半同步 强一致	1	1	1	>1	>2
	2	1	1	>2	>3
	3	1	1	>3	>4
	x	y	z	>x	>x+y
分组半 同步	1	1	1	>1	>2
	2	2	2	>1	>2
	3	2	2	>1	>2
	x	y	z	>1	>2

- 从库宕机会影响ack应答，导致业务夯住
- 从库节点越多，ACK应答数量多，影响数据库事务处理能力
- 主库宕机，容灾切换后，ack数量需要手动调整

- 解决从库宕机，网络抖动对ACK应答数量的影响
- 使用最少的ACK应答数量，实现数据一致性保障
- 避免主库宕机，对容灾切换的影响。
- 需要更多的从库节点



服务器

机房、机柜

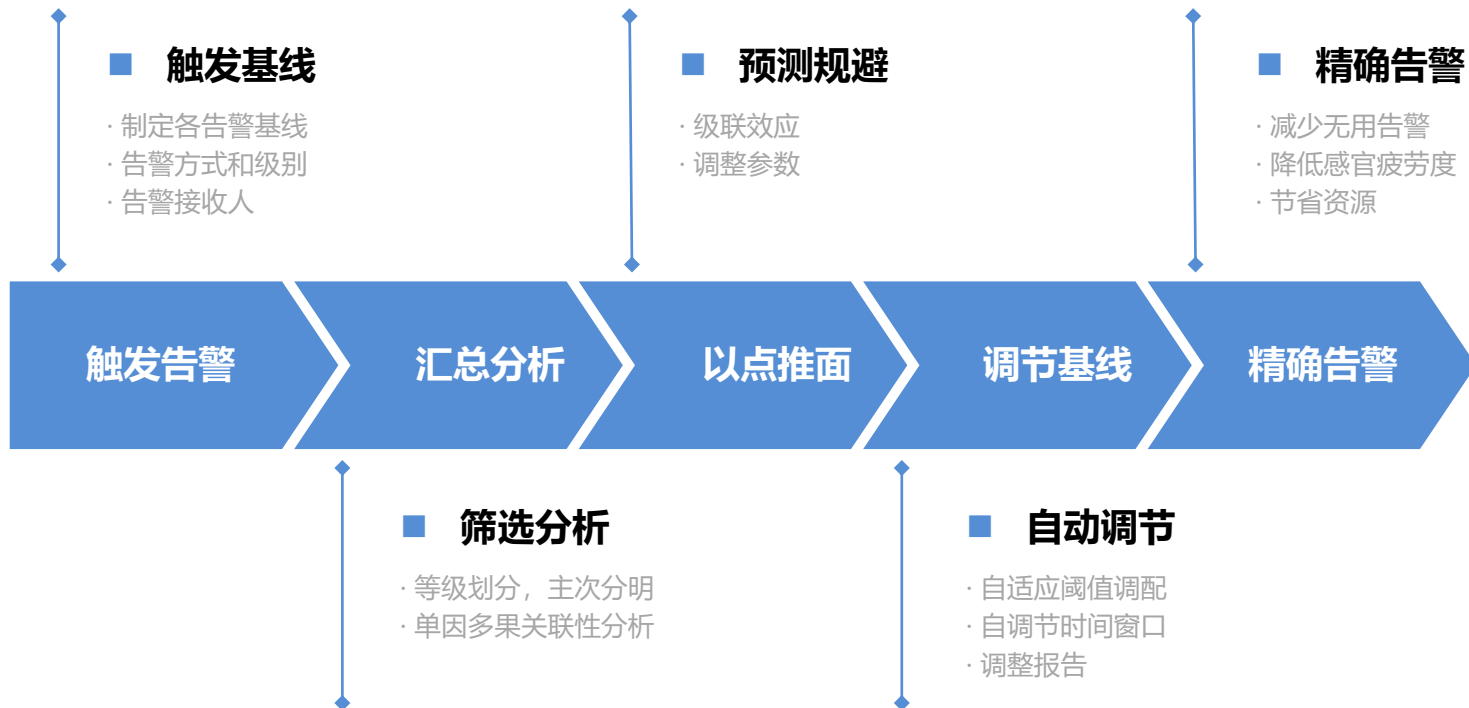
网段、ip

实例

业务线

数据库

- 资源自动上报
- 服务器使用状态管理
- 数据库与应用信息匹配
- 数据库与业务研发匹配
- 业务访问链路拓扑信息
- 历史信息追溯
- 服务器、实例、库、业务线血缘关系
- 元数据变更管理
- 资源使用视图
- 资源使用率报表
- API服务



大促备战分享

研发赋能、抓核心链路
监控告警、高可用切换、备份恢复、容量管理、变更管控、应急方案

备战准备

- 备战巡检
- 容量评估
- 优化改造
- 数据归档
- 压力测试
- 切换演练
- 变更管控

大促

- 大促调整
- 监控大屏
- 应急处理

案例复盘

- 事件管理
- 案例分享
- 经验总结

大促备战分享

自增主键

单表21亿
全局自增主键

磁盘空间

系统相关
DB相关

表分区

时间维度的分区
时间维度的拆分

连接数

活跃连接数
max_connections

备份

备份作业
有效性检查

热点表

数据量大
拆分不均匀
影子表

Top SQL

SQL 频次
SQL 效率

慢查询

SQL 消耗
优化改造

定时调度

业务跑批 ETL抽取
信息采集 备份调度

容量评估

硬件容量
性能容量

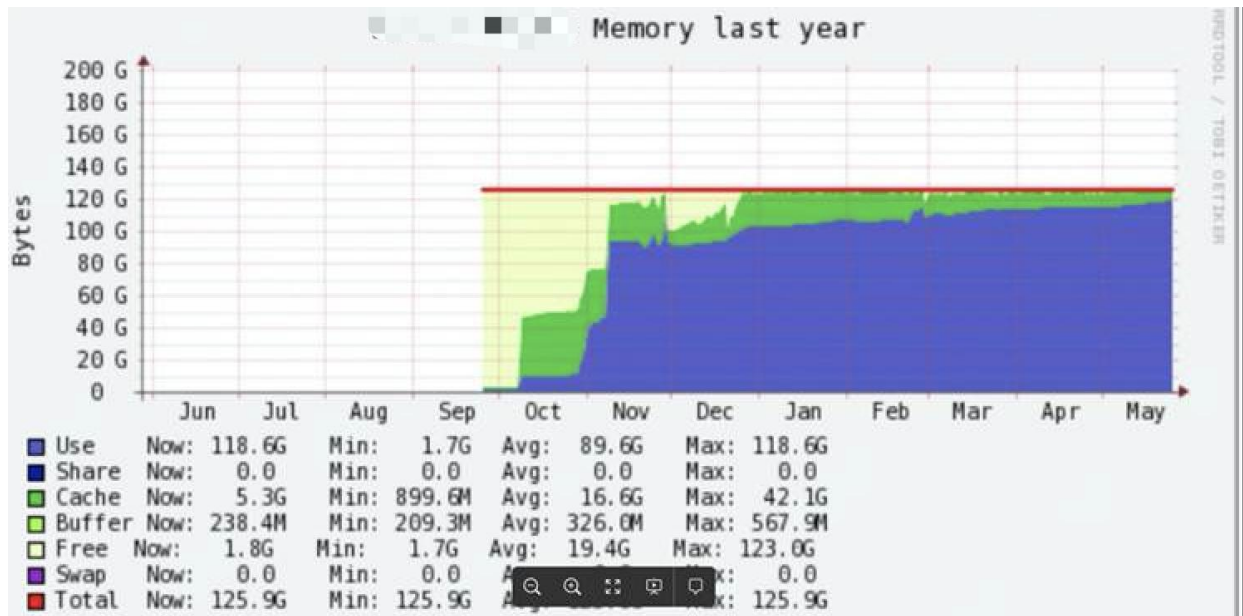
硬件&机房

磁盘 Raid卡
出口带宽 机柜超电
双电链路 温湿度

业务梳理

数据库依赖程度
事务读写逻辑
上下游调用逻辑

大促备战分享



总内存: 128G innodb_buffer_pool_size: 80G 实际使用内存: 118G connections: 2000

实际使用内存

118G

—

Innodb_buffer_pool

80G

—

Session级别*conn

20G

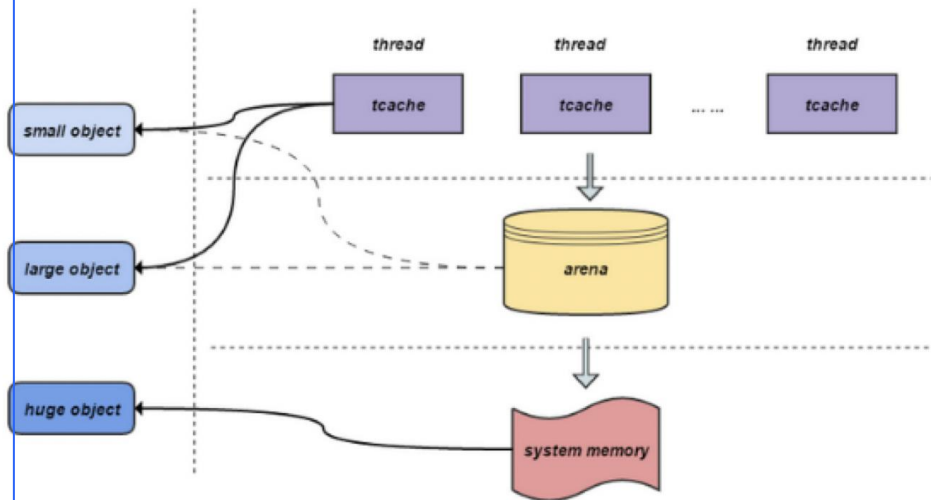
=

泄漏内存?

28G

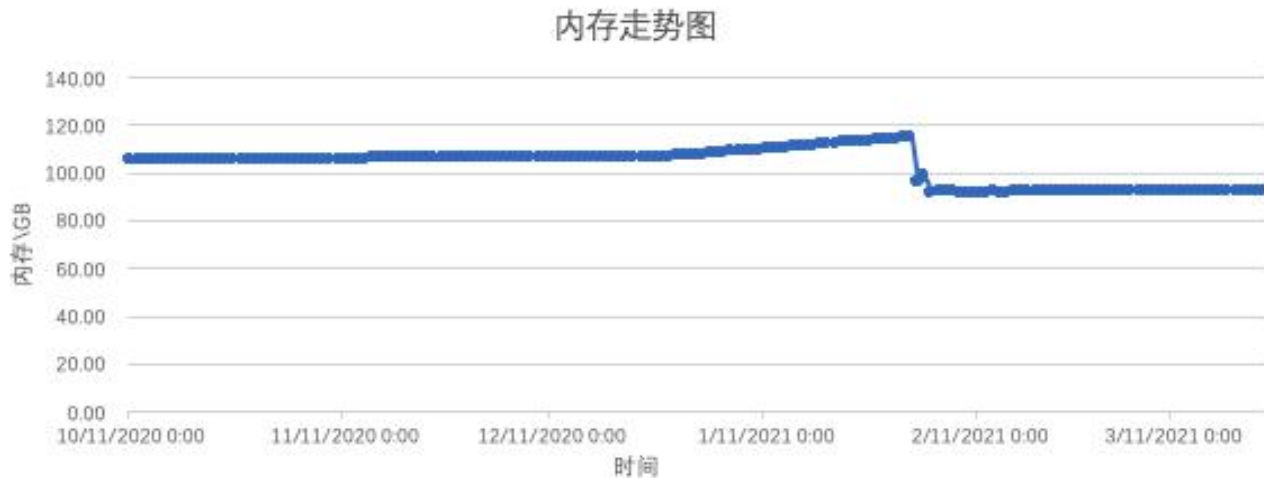
Ptmalloc:

- glibc 实现的内存分配器
- 标准实现
- per thread独享arena
- 提升多线程性能
- 线程结束前不主动回收
- 内存浪费严重

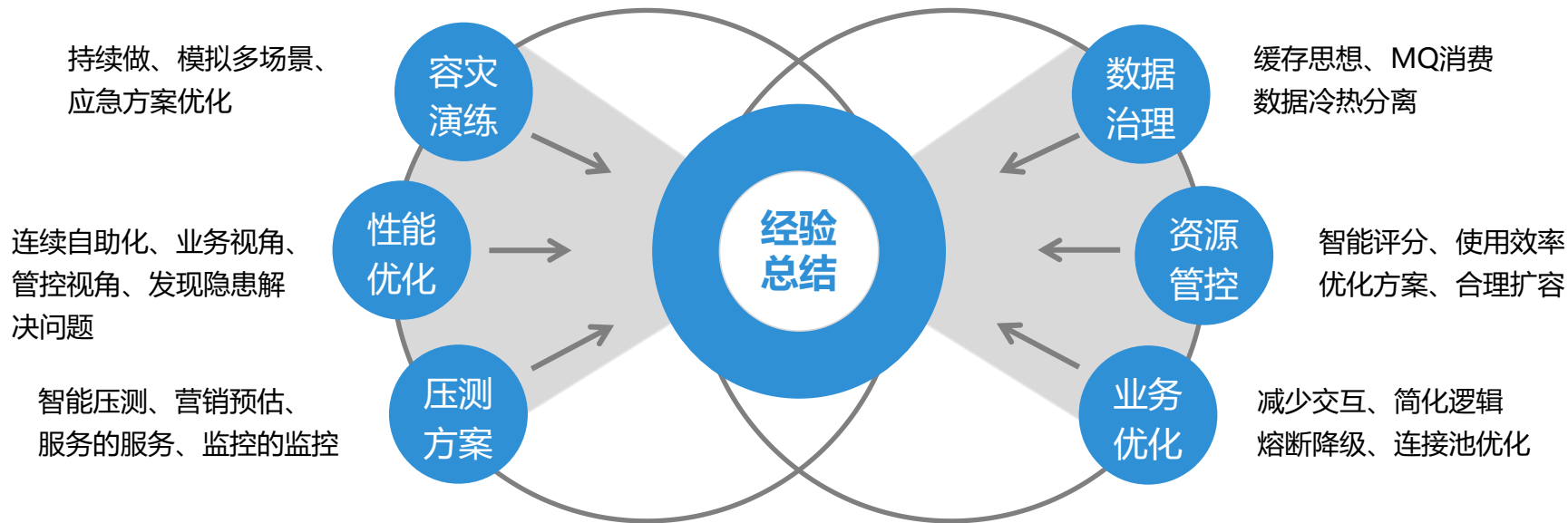


Jemalloc:

- 共享arena
- 线程缓存
- 进一步提升多线程性能
- 内存主动回收
- 减少内存碎片
- 灵活api,可审计和回收



- 某系统承载了京东科技业务的流水明细，每日增量5000w+，按日期做的分库分表
- 不定期OOM
- 表数量多\空表较多\开启information_schema采集\定期回收表碎片，内存增速更快
- 加载jemalloc插件后（my_jemalloc.so），内存保持平稳，不持续增长，避免发生oom重启





全球敏捷运维峰会

THANK YOU !

