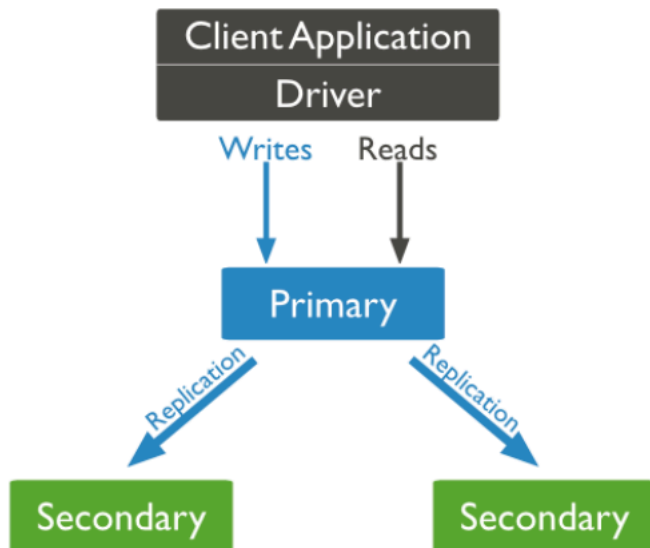




高德单元化改造 ——实践与思考



从历史1.0说起





2.0的演变，同城双机房

同城双机房—单实例跨双机房

✓ 高可靠，集群高可用

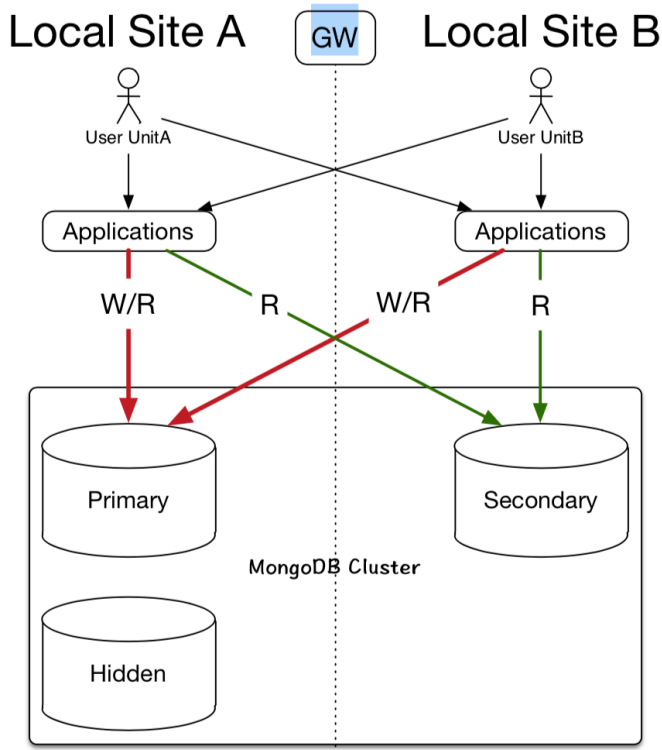
- 三份数据存储
- 任意节点失效不影响服务

✓ 主机房写高可用，次机房读高可用

- 主机房含有大多数节点，满足选举条件
- 次机房只还有少数节点，无法成为Primary
- 强制改变节点角色？影响集群恢复

✓ 业务普适，适合云基础用户

- 普遍物理环境
- 使用接口简单





2.0的演变，同城三机房

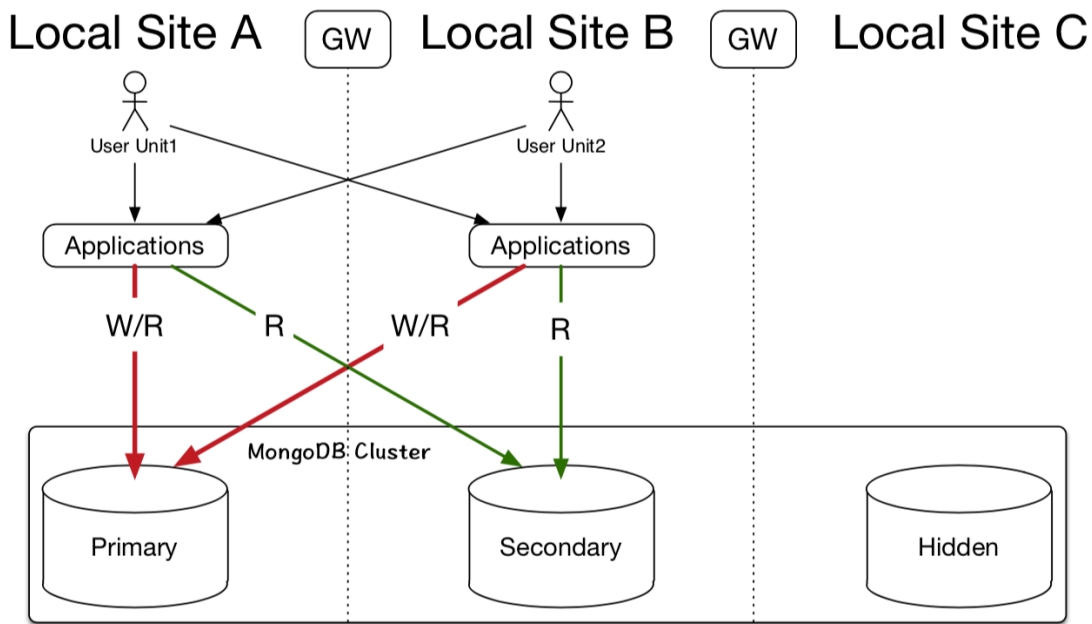
同城三机房—单实例跨三机房

✓ 高可靠，集群高可用，机房高可用

- 三份数据存储
- 任意节点失效不影响服务

✓ 业务普适，但机房环境要求苛刻

- 已具备，上海（杭州），深圳



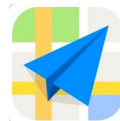


业务发展面临的挑战

资源扩容

就近接入

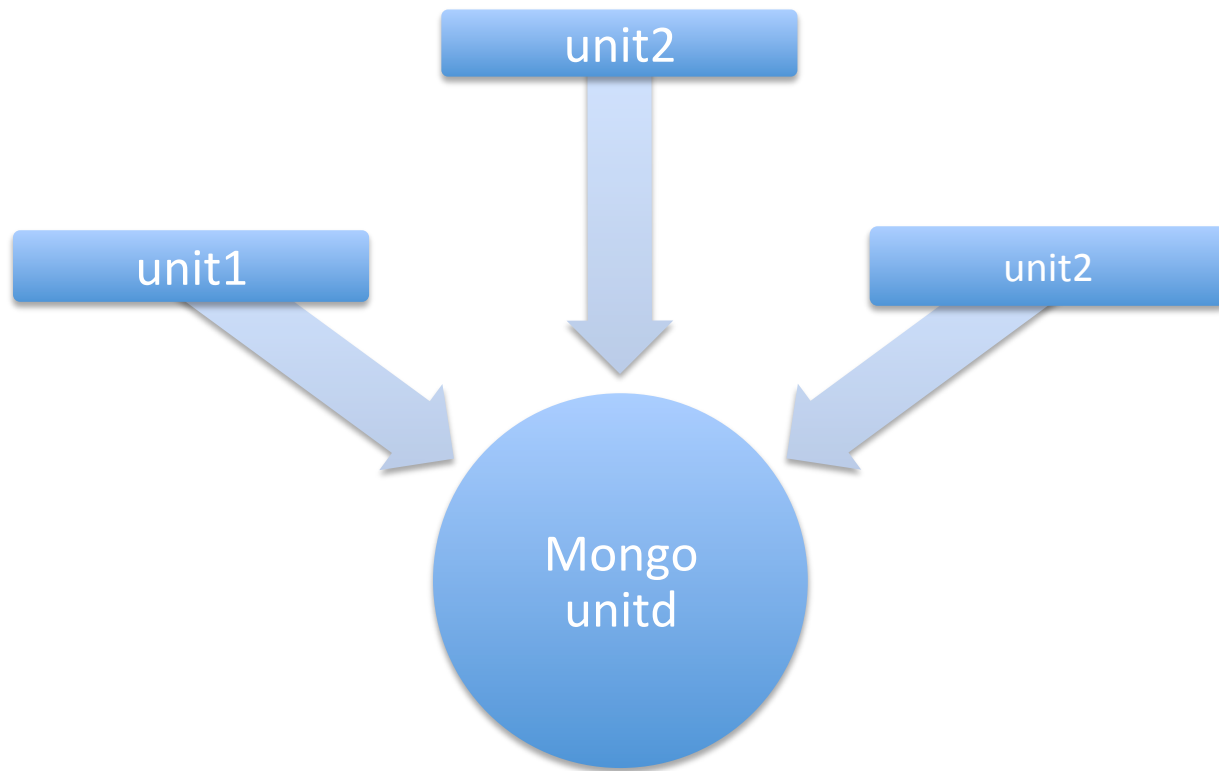
异地容灾





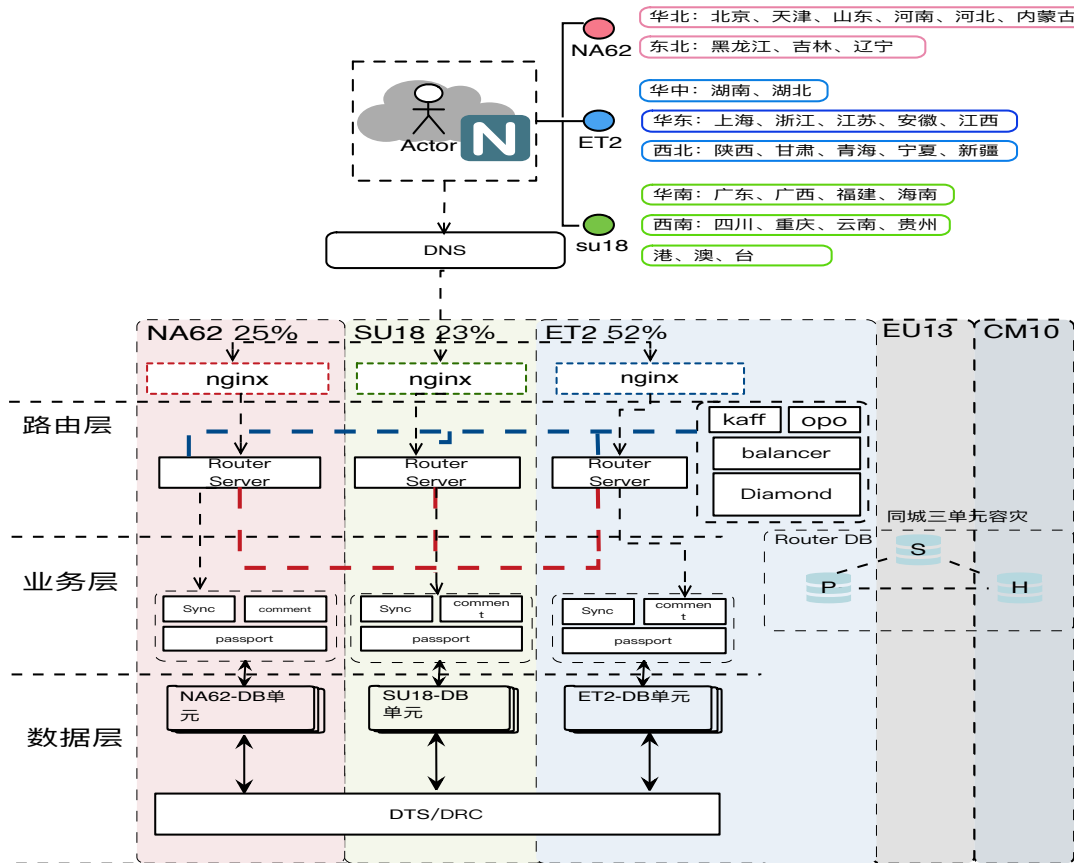
解决思路：单元化改造

- 共同承担
- 高度自治
- 互帮互助





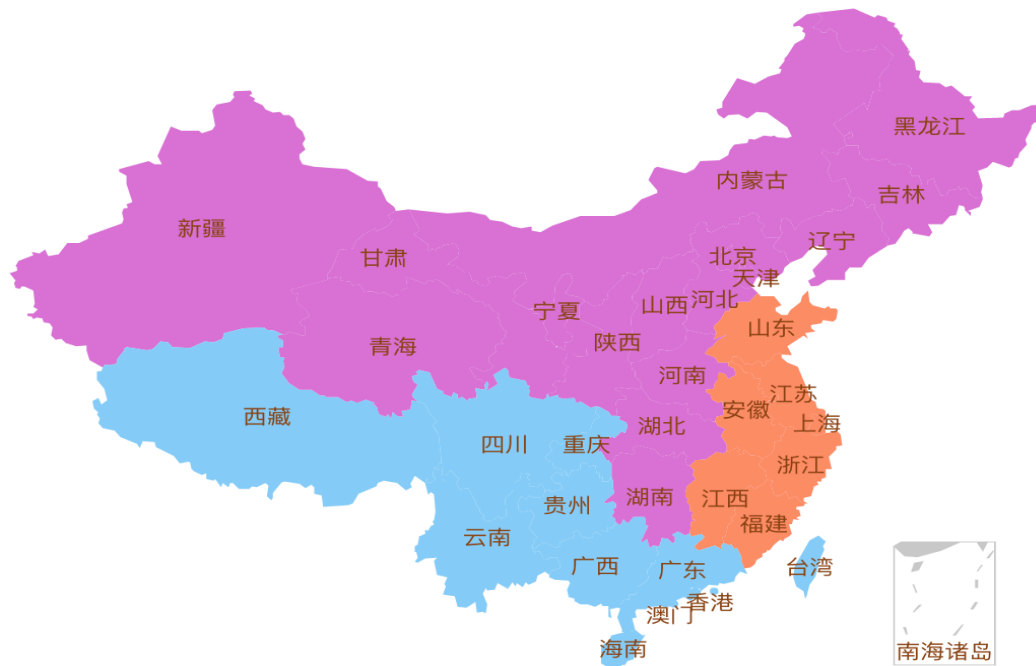
单元化业务架构1.0





单元化后业务流量分布

单元省份分布





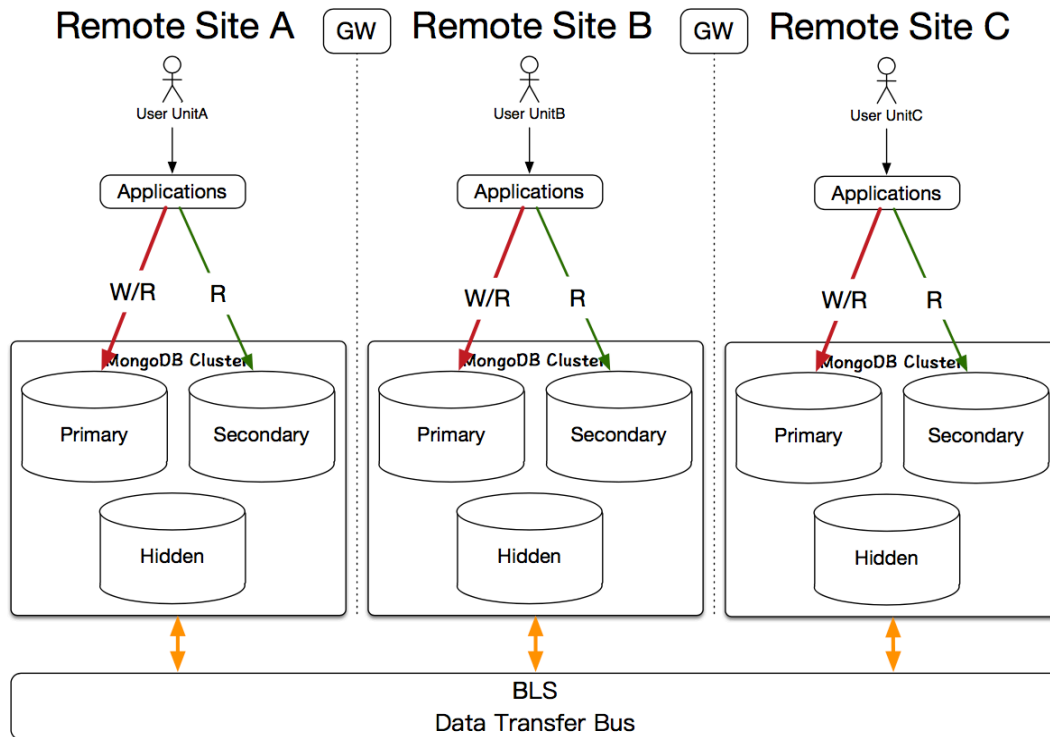
路由层

- 分组策略（用户维度）
 - 地域（就近接入）
 - 模除（流量均分）
 - 其它（结合业务特性）
- 其它
 - 热度调整
 - 容灾切流



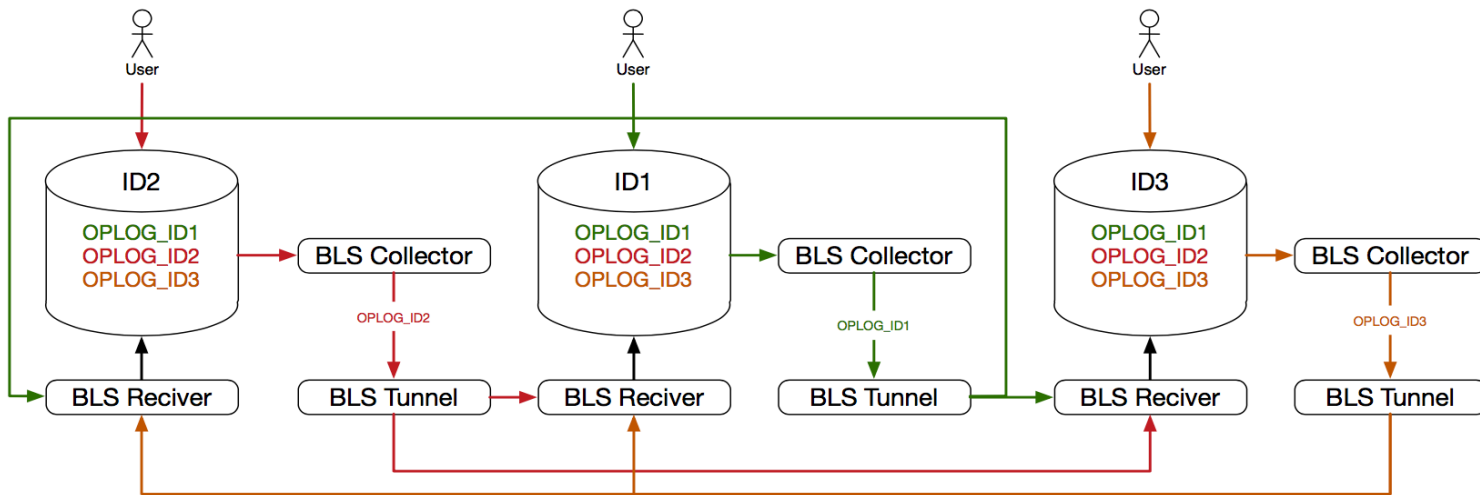
DB层改造

异地多机房—多实例多单元





MongoDB BLS数据同步—多单元全量同步

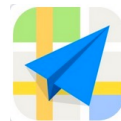


- ✓ OPLOG记录实例ID
- ✓ Collector & Receiver根据ID过滤相应OPLOG
- ✓ 实例ID全局唯一不变



容灾&单元化总结

	高可用	高可靠	单元化	关注
单实例&同城&双机房	主机房读写高可用， 次机房读高可用	实例数据高可靠	交叉访问	次机房能力弱
单实例&同城&三机房	读写高可用	实例数据高可靠	交叉访问	三机房物理条件苛刻，上海（杭州），深圳满足
多实例&异地&多机房 ★	机房内高可用，容灾 切换单元流量	实例之间存在同步延迟	单元内流转	BLS同步数据
双实例&同城&双机房	机房内高可用，容灾 切换数据库访问流量	实例之间存在同步延迟	交叉访问	BLS同步数据





问题与思考

就近接入：区
域用户数据热
度不同

躲不开的CAP，
需要根据业务
做出取舍

人类的迁徙，
如果才能快速
拥抱变化



Thanks

高亮