Alibaba Group 阿里巴巴集团

OceanBase
A Powerful Database System

破解数据库高可用难题

孙志东（解伦）
xielun.szd@alipay.com

杭州
2014-6-15

OceanBase
A Powerful Database System

Alibaba Group 阿里巴巴集团

Agenda

传统数据库的高可用性

OceanBase 的高可用架构

OceanBase 的分布式选举

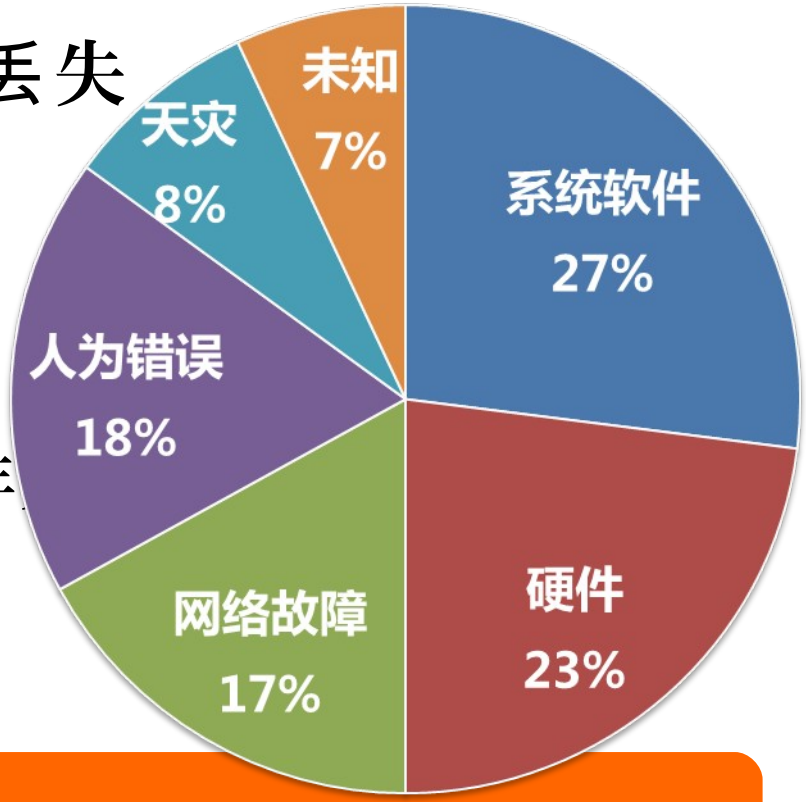
小结

OceanBase
A Powerful Database System

高可用的数据库系统

- 数据可用性：保证数据可访问
- 数据安全性：防止数据丢失

- 故障不可避免
  - 软件： Bug
  - 硬件：阿里数据中心每年
  - 天灾：致命的
  - 人为：误操作



竞猜题：按概率排序

Alibaba Group 阿里巴巴集团

几个“九”的认识

PERCENTAGE UPTIME	PERCENTAGE DOWNTIME	DOWNTIME PER YEAR	DOWNTIME PER WEEK
98%	2%	7.3 days	3 hours, 22 minutes
99%	1%	3.65 days	1 hour, 41 minutes
99.8%	0.2%	17 hours, 30 minutes	20 minutes, 10 seconds
99.9%	0.1%	8 hours, 45 minutes	10 minutes, 5 seconds
99.99%	0.01%	52.5 minutes	1 minute
99.999%	0.001%	5.25 minutes	6 seconds
99.9999% (“six 9s”)	0.0001%	31.5 seconds	0.6 seconds

OceanBase
A Powerful Database System

可用性的业务价值

- 5 个 9 可用性
- 5.25 分钟意味着什么？
  - = 580,000,000
  - = 17 万条内裤
  - = 伤百万用户的心

Alibaba Group 阿里巴巴集团

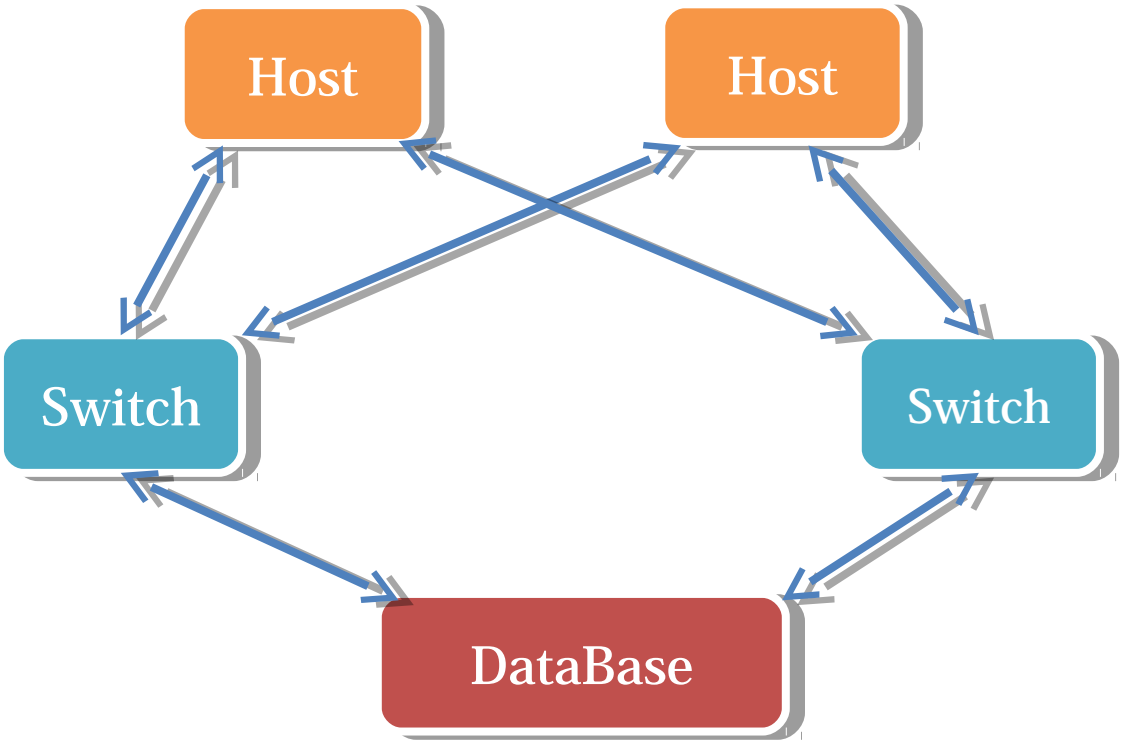
传统数据库的高可用

- 单机的高可用
  - 高可用的硬件
  - 组件级冗余
  - 无法避免单点故障？
- 集群的高可用
  - 高端的存储区域网（SAN）
  - 单机房部署，无法避免的“天灾”？
- 主备复制的高可用
  - 可以跨机房

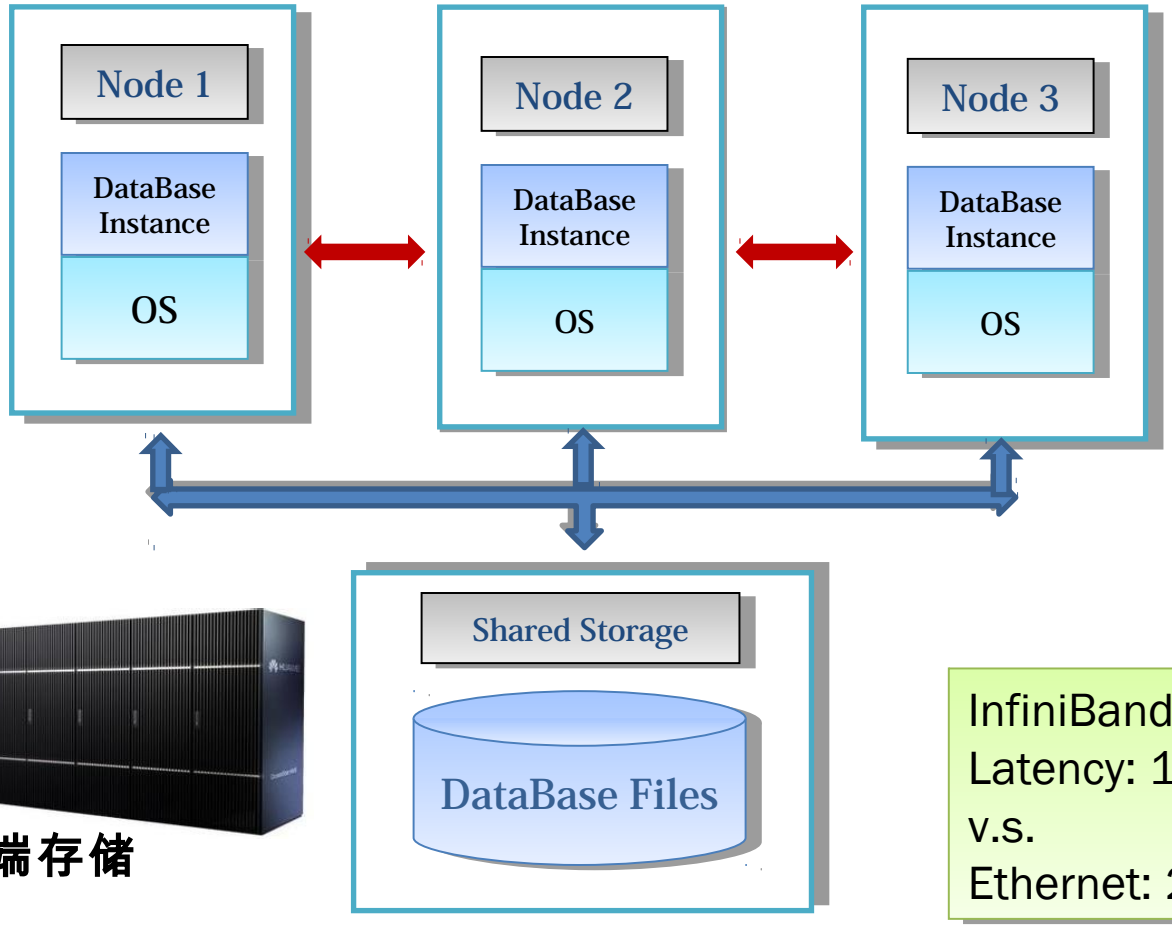
Alibaba Group 阿里巴巴集团

高可用的单机

- 双路冗余热交换电源
- 双路市电独立供电
- 双路冗余光纤交互专有网络



集群的高可用



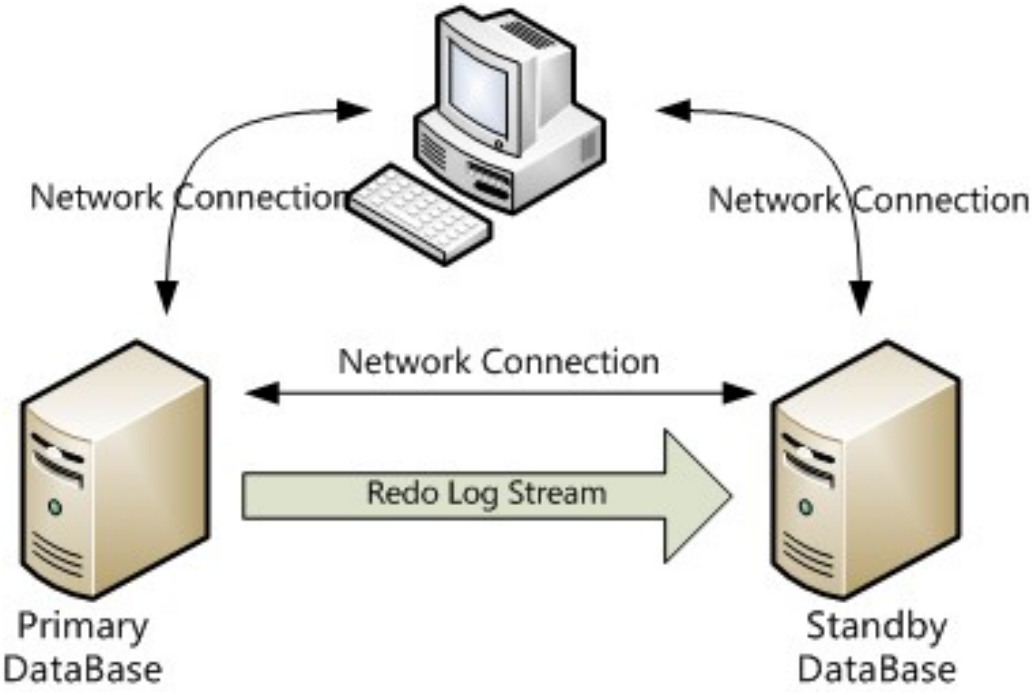
高档服务器

高端存储

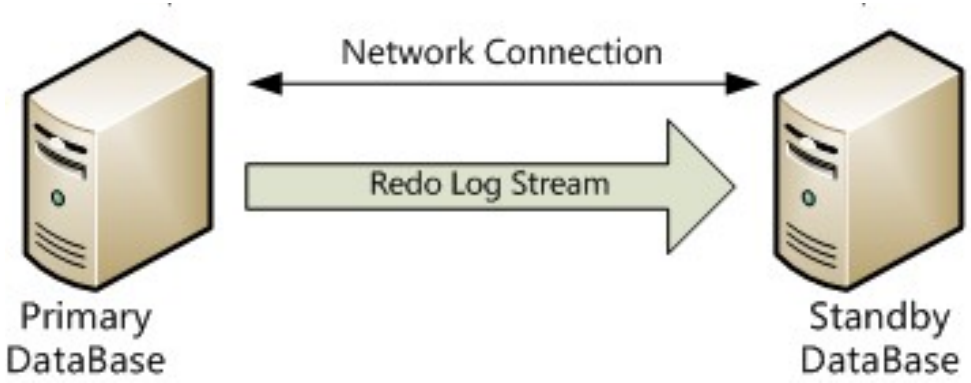
InfiniBand: 40Gbit/s
Latency: 1us
v.s.
Ethernet: 250 us

主备的高可用

- 主备复制
  - 主机提供读写，备机提供读服务
  - 主机宕机，把备机切换为主机
  - 一主N备



数据库主备复制

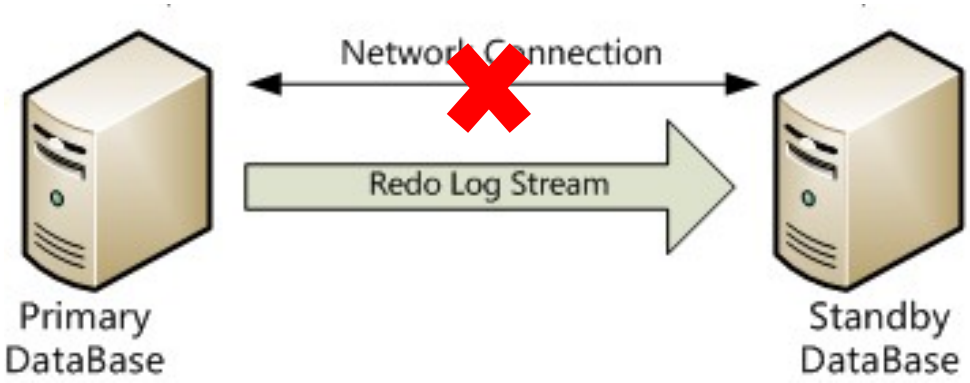


- 主机复制 redo 日志到备机
  - ➢同步模式（最大保护）
    - 备机是否写盘后应答主机？
  - ➢异步模式（最高性能）
    - 主机不等待备机的应答

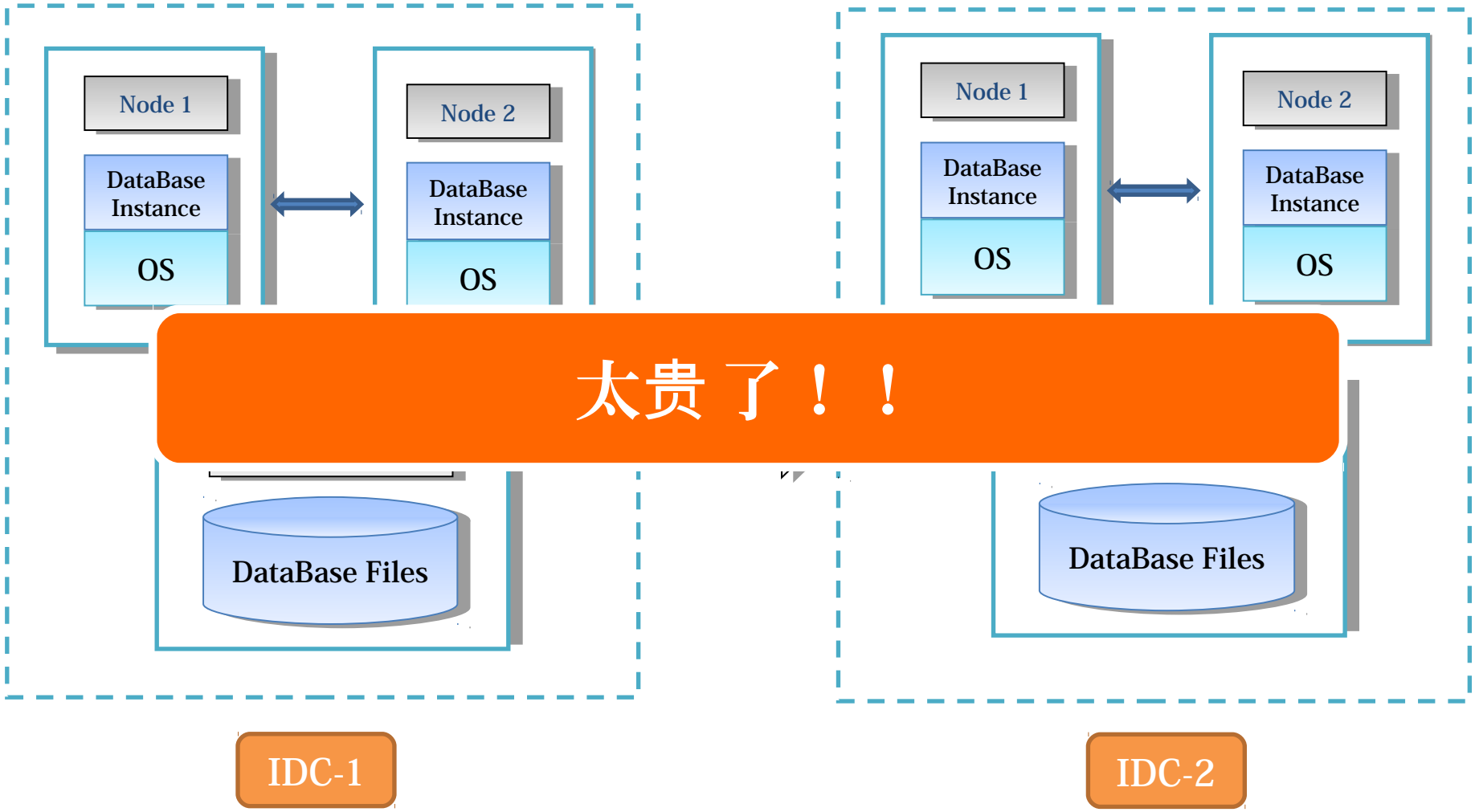
思考题：对比如上两种模式

数据库主备复制

- 主机复制 redo 日志到备机
  - 混合模式（最大可用）
    - 同步模式不可用时，转为异步模式
    - 权衡了最大性能和对数据的最大保护



高可用集群 + 主备复制

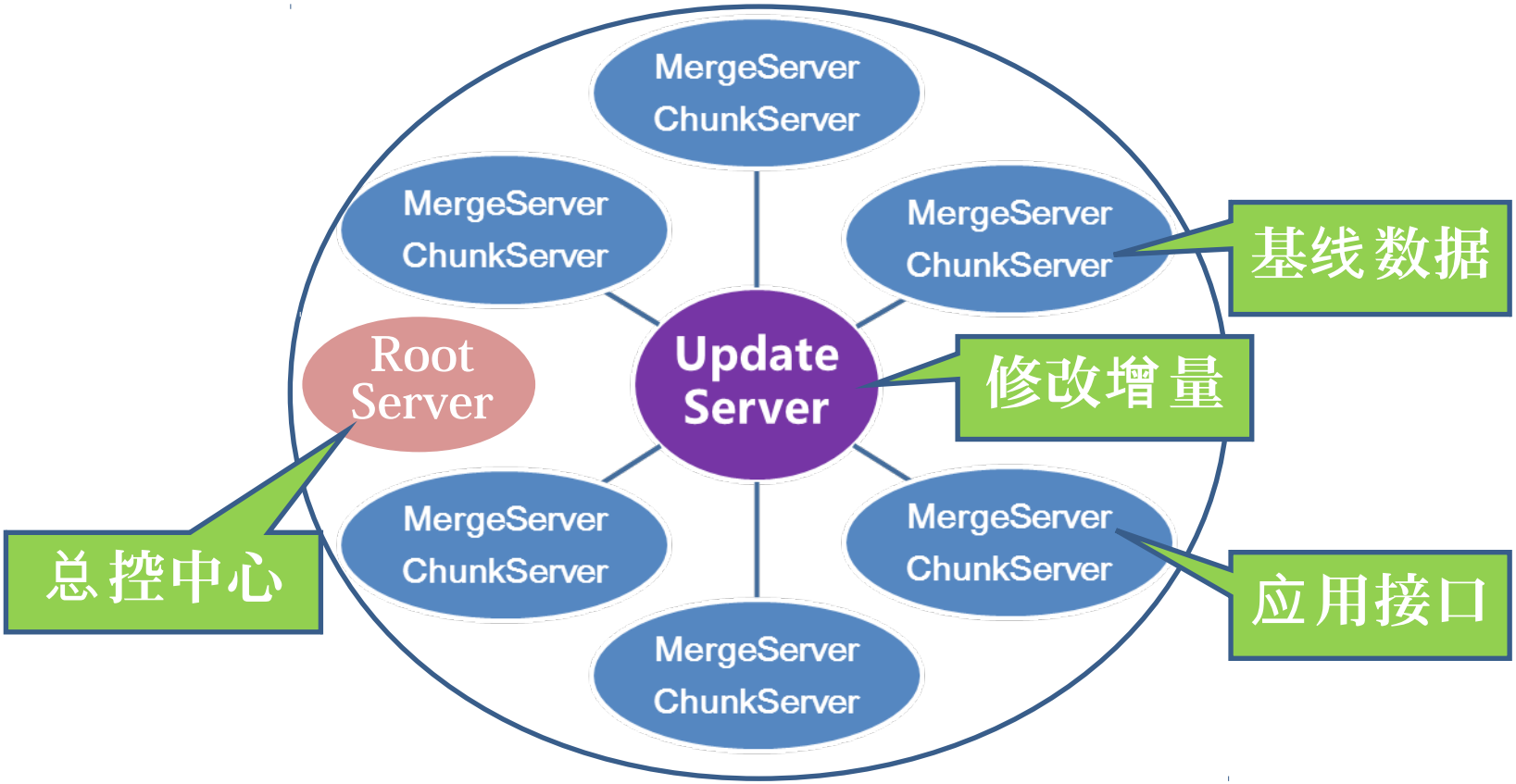

Node 1
DataBase Instance
OS
Node 2
DataBase Instance
OS
DataBase Files
Node 1
DataBase Instance
OS
Node 2
DataBase Instance
OS
DataBase Files
太贵了！！
IDC-1
IDC-2

Alibaba Group 阿里巴巴集团

OceanBase 的高可用性架构

OceanBase A Powerful Database System

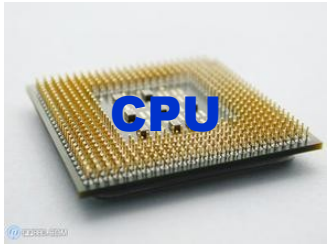
OceanBase 架构（单机群）



- 注：后续内容会将 OB 单集群作为一个 DataBase

分布式环境的困境

- 普通服务器 + 公用的网络环境
  - 故障成为常态
  - 跨机房部署是必需

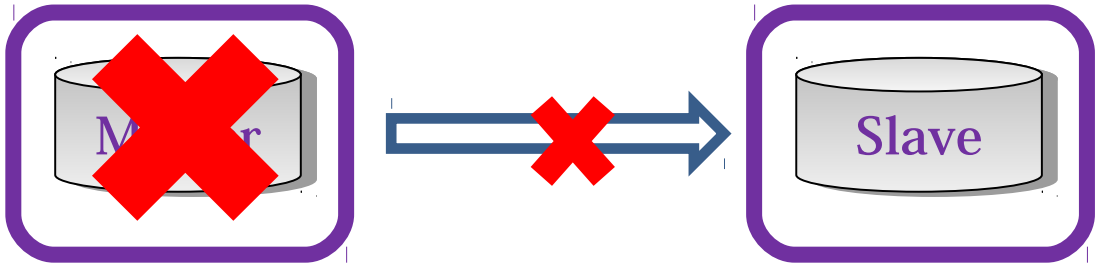


- OceanBase 只能走主备复制的路
  - 如何保证真正的数据零丢失？
  - 如何权衡性能、可用性、数据安全？

基于主备复制的困境

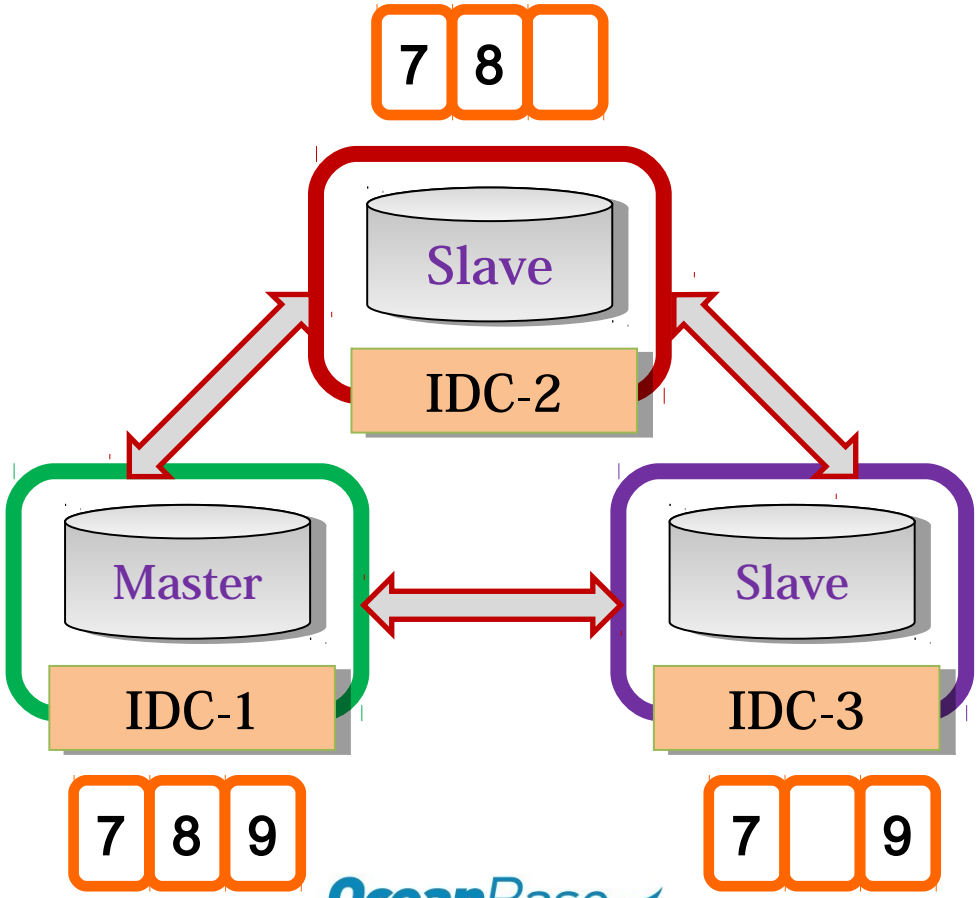
- 最大可用 = 最大保护 （同步） + 最大性能（异步）

- 最大可用模式的数据丢失



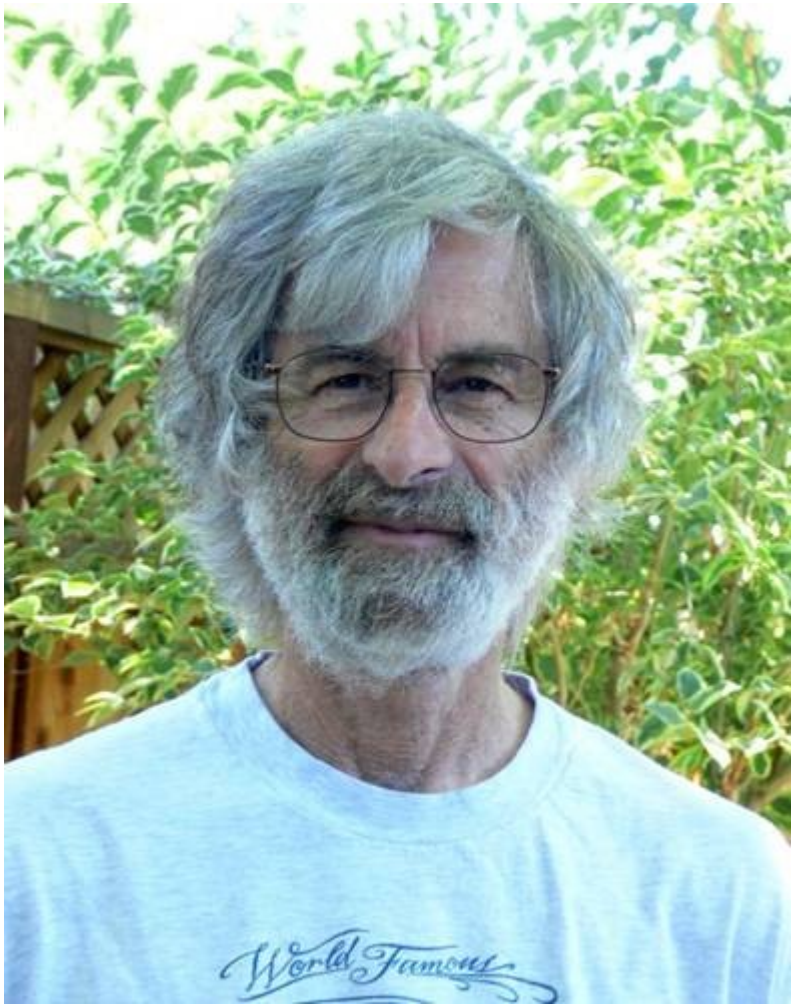
OceanBase 主备同步

- 主库执行写事务并同步 redo 日志到备库
  - 多数成功就认为写事务成功



Alibaba Group 阿里巴巴集团

Paxos



- "Time, Clocks, and the Ordering of Events in a Distributed System"
- Byzantine generals
- Paxos
- LaTeX
- 2013 ACM Turing Award

Alibaba Group 阿里巴巴集团

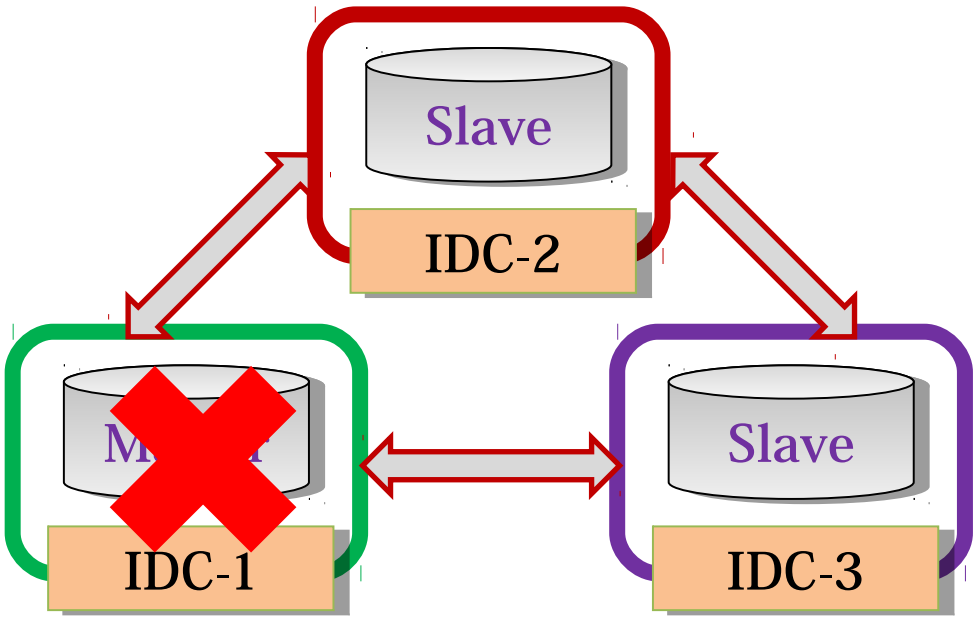
基于投票的同步复制

- 优点
  - 保证了数据安全性
    - redo 日志强同步写多份
  - 更大化系统的可用性
  - 单机房故障不影响读写服务

- 数据一致性 & 系统可用性：3/5 > 2/3 > 2/2

小结

- 同步 redo 日志写多份保证数据零丢失
- 多数写成功即成功提供更高的可用性



Alibaba Group 阿里巴巴集团

OceanBase 的分布式选举

OceanBase A Powerful Database System

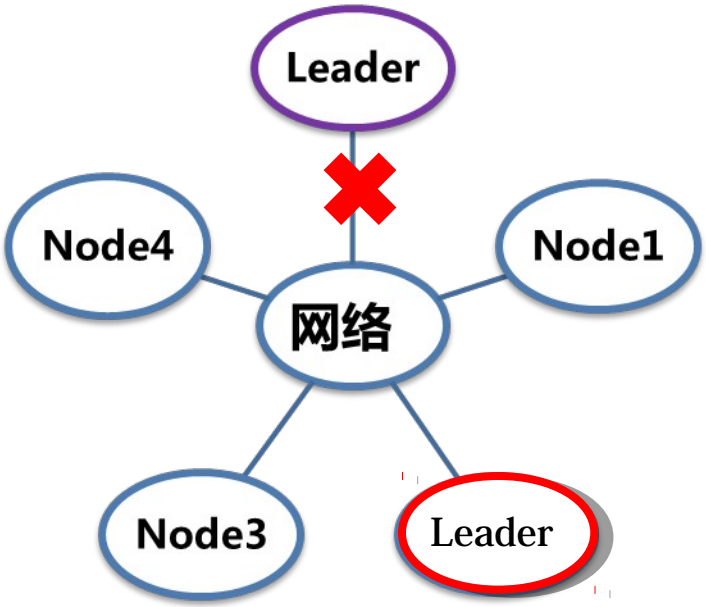
Alibaba Group 阿里巴巴集团

投票和选举

OceanBase
A Powerful Database System

分布式选举问题概述

- 原则：任意时刻最多只能有一个 Leader
- 投票协议，以不可靠成员提供可靠服务
  - 多数 (超过半数) 成员可用则服务可用
- 简单投票协议
  - 要容忍网络分区
  - Leader 租约（Lease）



Alibaba Group 阿里巴巴集团

分布式选举基本原理

- Paxos 协议的基本要求
  - 成员不说假话（非拜占庭式）
  - 单个成员说话不自相矛盾：投票给 A 了，就不能再投票给 B
  - 任何修改需要多数成员同意：多个成员投票的同步
- 单个成员说话不自相矛盾
  - 对投出的票进行持久化？
  - 记住自己在一个 lease 周期内的投票（分布式选举）
  - 重启后一个 lease 周期内不投票
- 多个成员投票的同步
  - 有协调者 (leader)
  - 无协调者？

OceanBase A Powerful Database System

Alibaba Group 阿里巴巴集团

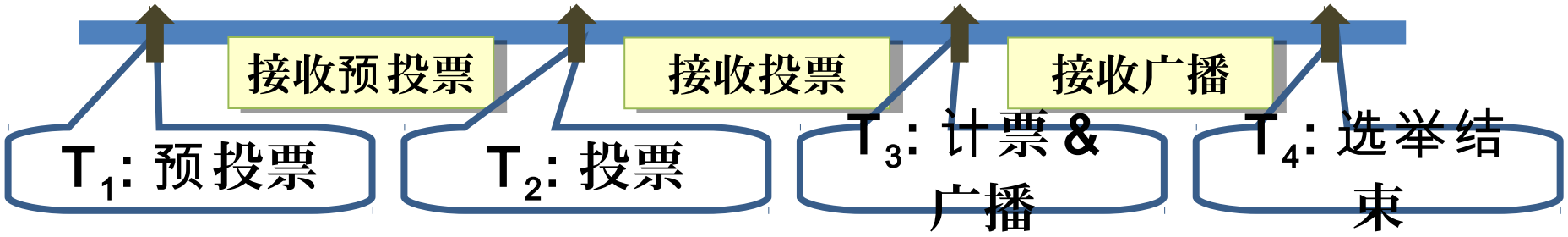
多个成员投票的同步

- 问题
  - 系统刚启动 或者原 leader 异常 lease 过期后，选举 leader 的时候，各个参与者的投票时间各不相同，每个参与者收到选票的时间各不相同
  - 投票后，参与者在一轮 lease 周期内不得再次投票 (“不得自相矛盾”)
- 已有方案
  - 各个参与者在发起投票时，延迟某个随机时间 (100ms~300ms)，最早发起者通常成为新的 leader
    - 时序逻辑与选举逻辑紧密耦合、业务规则难以融入选举中
    - 容易出现选举失败 (election split)，下次选主要等 Lease 过期！！

OceanBase A Powerful Database System

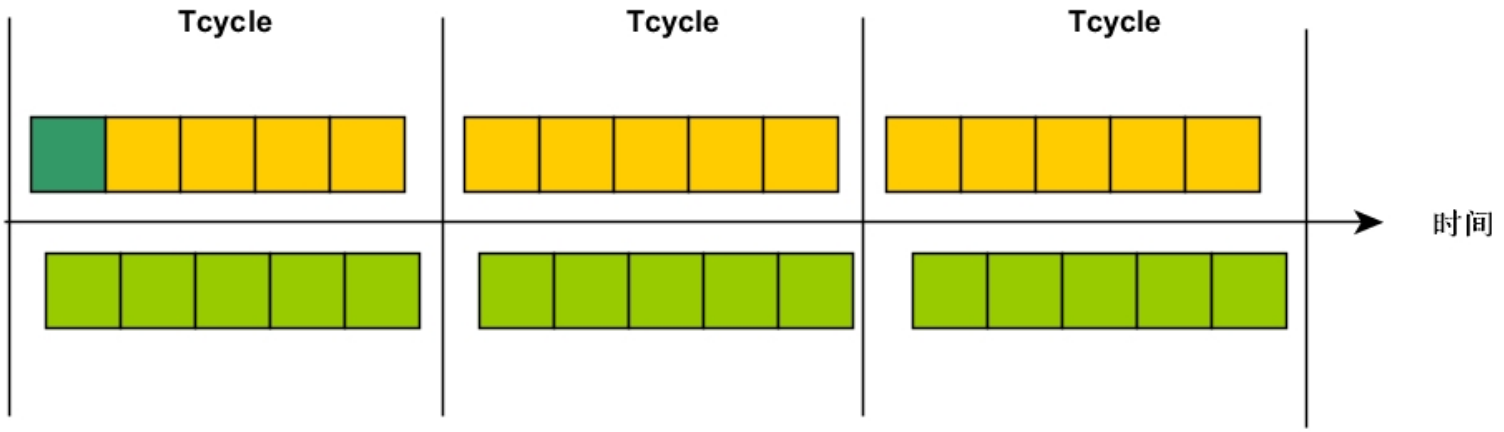
“同步”无主选举

- 按统一的规则(“投票权重”)选择新leader
- 所有成员在T_1时刻“同时”发起选举

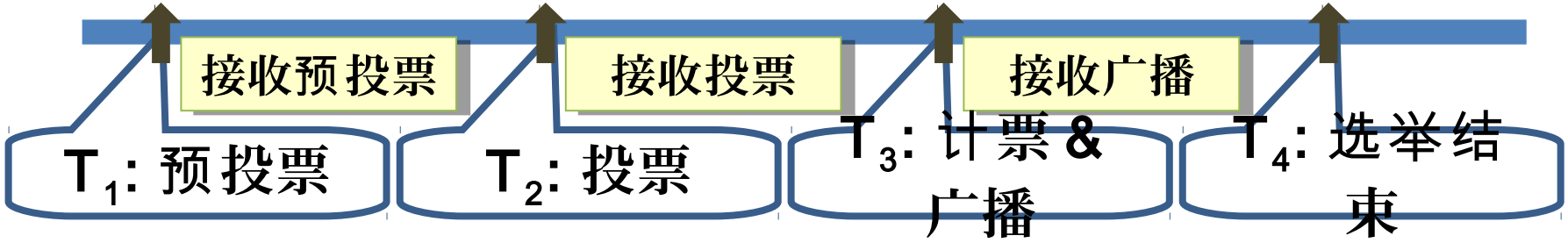


“同步”时钟

- 时钟充当无主选举的协调者
  - 每个进程的时间被均分为时间片
  - 每个时间片内只能进行一次无主选举
  - 在 T_{cycle} 整数倍时刻发起选举

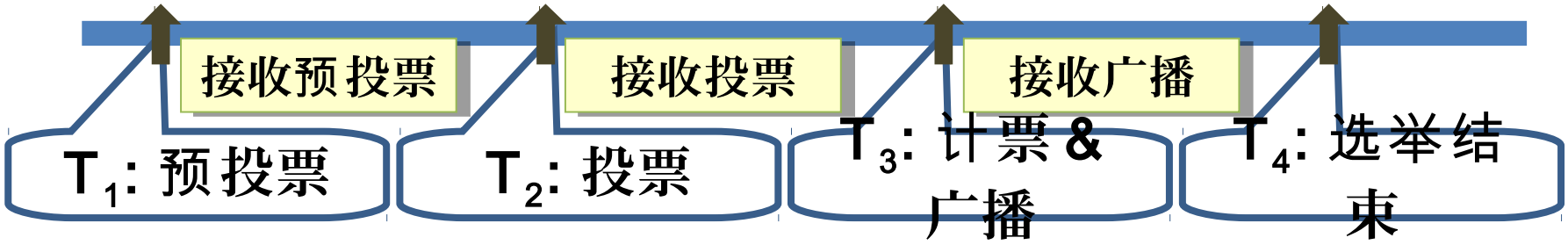


无主选举时序分析



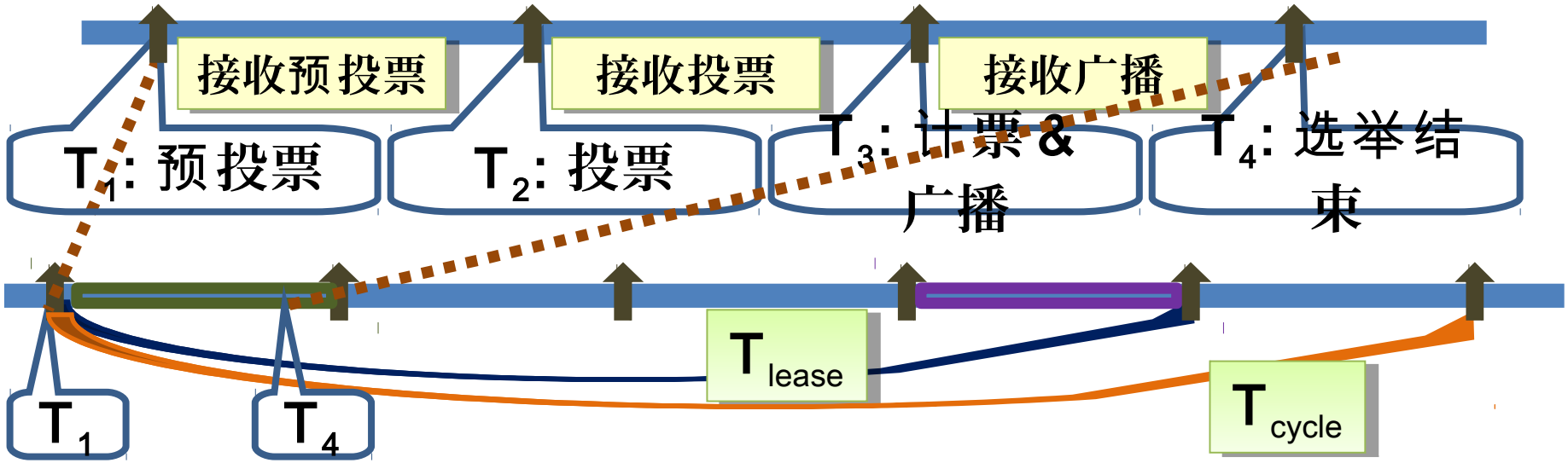
- 时钟偏差最大 T_{diff}，网络单程收发传输处理时间最长 T_{st}
- Step 1 ：T_1 时刻广播投票权重
- Step 2 ：接收投票权重并在 T_2 时刻向最大值者投票
  - 预投票到达时间：$[T_1-T_{diff}\times2，\ T_1+T_{diff}\times2+T_{st}=T_2]$

无主选举时序分析



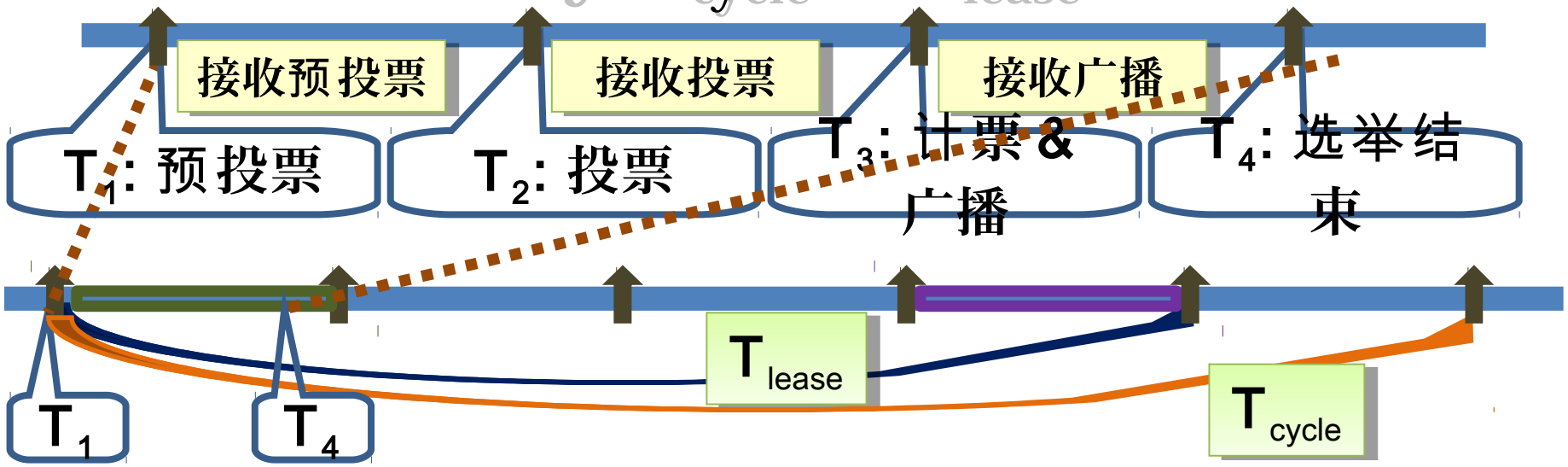
- Step 3 ：接收选票，T_3 时刻计票，得票过半者成功并广播
  - 投票到达时间：$[T_2-T_{diff}\times2，T_2+T_{diff}\times2+T_{st}=T_3]$
- Step 4 ：接收新任 leader 广播并在 T_4 时刻结束选举
  - 新任 leader 广播到达时间：$[T_3-T_{diff}\times2，T_3+T_{diff}\times2+T_{st}=T_4]$
- 选举耗时 $T_{elect}=T_4-T_1=T_{diff}\times6+T_{st}\times3$

Lease 及无主选举周期

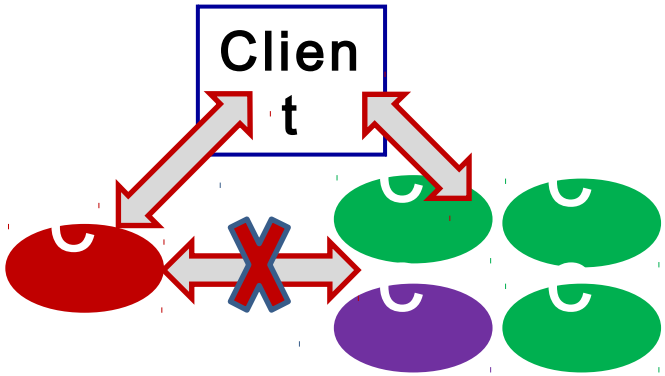


- 时钟偏差 T_{diff}=100ms，网络单程传输 T_{st}=200ms
- 选举耗时 $T_{elect}=T_{diff}\times6+T_{st}\times3$=1200ms
- 扩展的选举耗时 $T_{elect2}=T_{elect}+200$=1400ms
- $T_{lease}=4\times T_{elect2}$=5600ms，从 T_1 开始
- 无主选举周期 $T_{cycle}=5\times T_{lease}$=7000ms

Why $T_{cycle} > T_{lease}$?

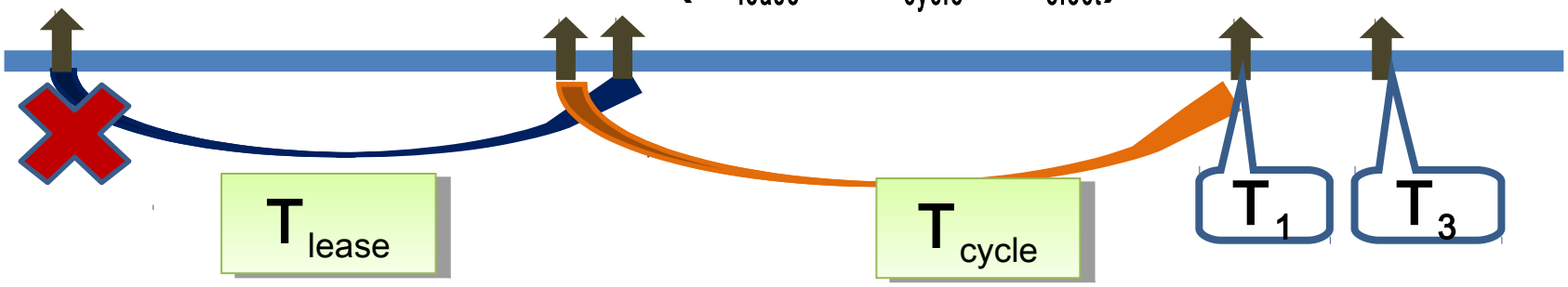


- 5 个成员 C_1~C_5 选举
  - T_1 时刻开始选举，选出新 leader C_1
  - T_4 时刻 C_2~C_5 未收到 C_1 新任 leader 广播
  - T_{cycle} 时刻 C_2~C_5 重新开始选举，选出新 leader C_4



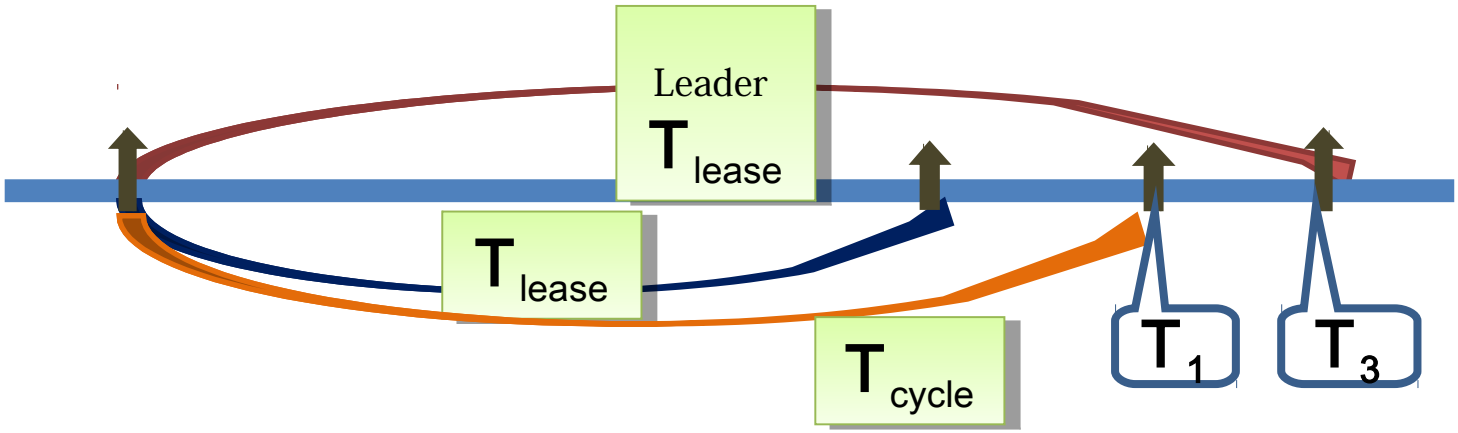
“同步”无主选举的优缺点

- 优点
  - 超过半数成员正常且参与，则选举一定成功
  - 实现简单：定长数据结构 + 新消息直接覆盖旧的 + 定时处理
- 缺点
  - 对最大时钟偏差及最大网络传输时间有要求
  - Leader 异常后，最长 (T_{lease} + T_{cycle} + T_{elect}) 选出新 leader



选举协议的鲁棒性

- 可能双主(脑裂)
  - 无主选举：若 $T_{diff} > (T_{elect2}+T3-T1)=2200ms$
  - 自动监控：A 在 T_a 时刻发包，B 在 T_b 时刻收到，则：$-2*T_{diff} <= T_b-T_a <= 2*T_{diff} + T_{st}$



Alibaba Group 阿里巴巴集团

总结

- 传统数据库的高可用性
  - 依赖昂贵的硬件设备
  - 主备同步的局限性
- OceanBase 的高可用性架构
  - 不可靠的 PC 服务器
  - 利用分布式投票实现的多机日志同步
    - 保证强一致的同时提供更大可用性
  - 利用分布式选举实现了可靠的选主
    - 主宕机后自动恢复保证写的可用性

Alibaba Group 阿里巴巴集团

Thanks

开源的分布式关系数据库

OceanBase

http://alibaba.github.io/oceanbase/

OceanBase A Powerful Database System

Alibaba Group 阿里巴巴集团

实际的分布式系统环境

OceanBase
A Powerful Database System

Alibaba Group 阿里巴巴集团

分布式CAP难题

- Any networked shared-data system can have at most two of three desirable properties:
  - consistency (C) equivalent to having a single up-to-date copy of the data;
  - high availability (A) of that data (for updates); and
  - tolerance to network partitions (P).
- 如果发生了网络分区，高可用性和数据一致性不可兼得

Alibaba Group 阿里巴巴集团

投票协议

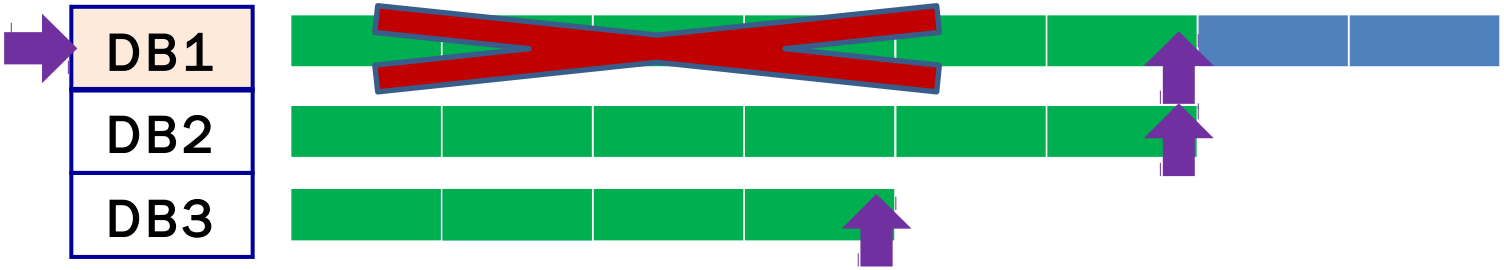
- 分布式投票和选举协议，以不可靠成员提供可靠服务
  - 多数(超过半数)成员可用则服务可用
- 协议本身比较复杂，但基本原理并不复杂
  1. 成员不说假话(非拜占庭式)
  2. 单个成员说话不自相矛盾
  3. 任何修改需要多数成员同意
- 基本做法
  - 通常选出一个leader作为协调者，但任何修改仍然需要多数成员同意
  - Leader在lease时间内有效，lease延长需多数成员同意

日志按事务顺序同步

- 主库执行事务，写磁盘，发送给备库，但不提交
- 备库按事务 commit 顺序应答主库
  - 收到 #5 及之前的事务日志，未收到 #6 号事务日志，但收到 #7,#8 号事务日志，只应答到 #5
  - 优点：备库日志中间不会有空洞，实现相对简单
  - 缺点：对网络抖动和服务器性能抖动的抵抗能力较弱，事务可能偶尔“顿住”

按事务顺序同步 - 故障恢复 (1.1)

- 故障恢复 ： 1. 日志最新者为主， 2. 事务日志到达超半数的库， 3. 恢复过程中不对外服务
- Case 1 ： DB1 故障

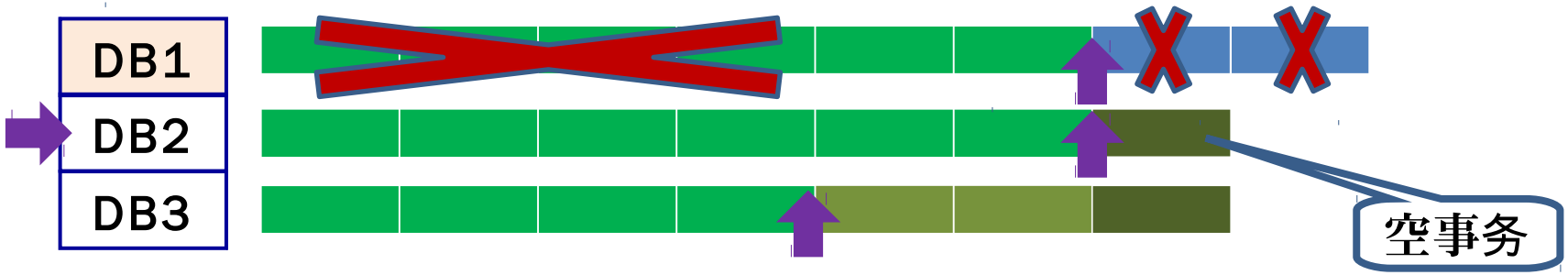


  - Case 1.1 ： DB1 又恢复了，成为主库

- 恢复被中断了， DB1 又故障了？
  - 恢复中只补全数据副本，不增删改业务数据，是可重入的

按事务顺序同步 - 故障恢复 (1.2)

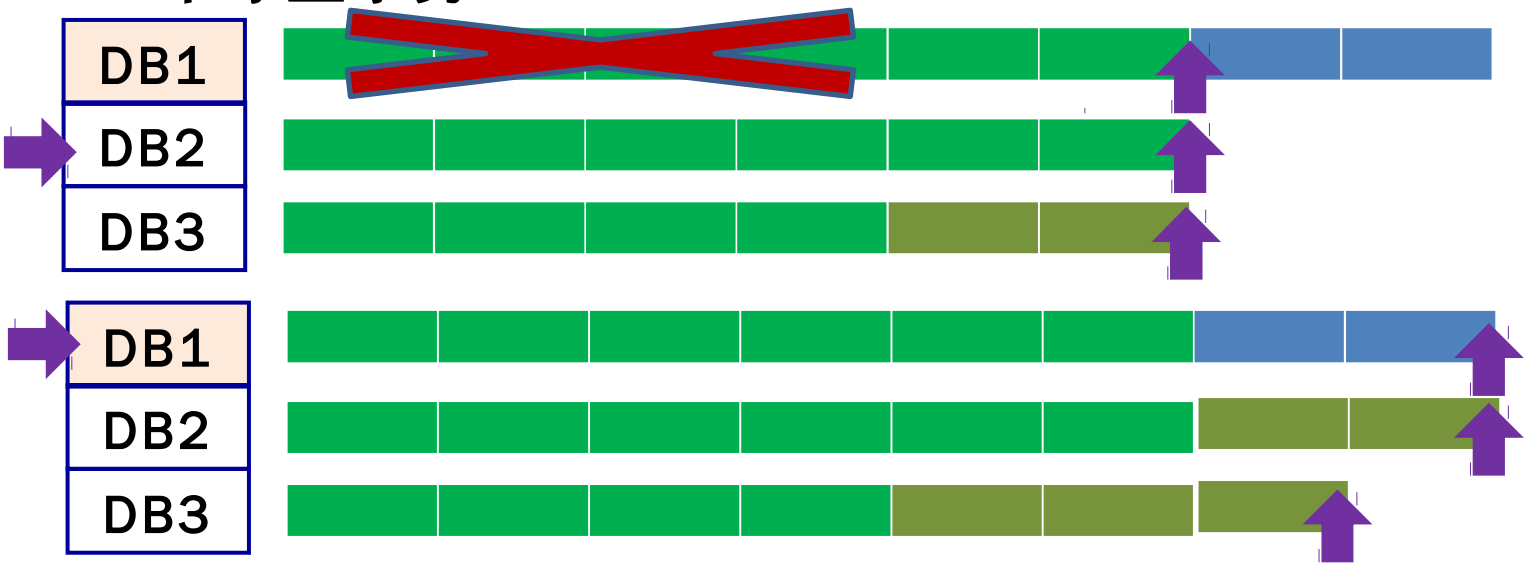
- 故障恢复： 1. 日志最新者为主， 2. 事务日志到达超半数的库， 3. 恢复过程中不对外服务
- Case 1 ： DB1 故障
  - Case 1.2 ： DB1 未恢复， DB2 成为主库



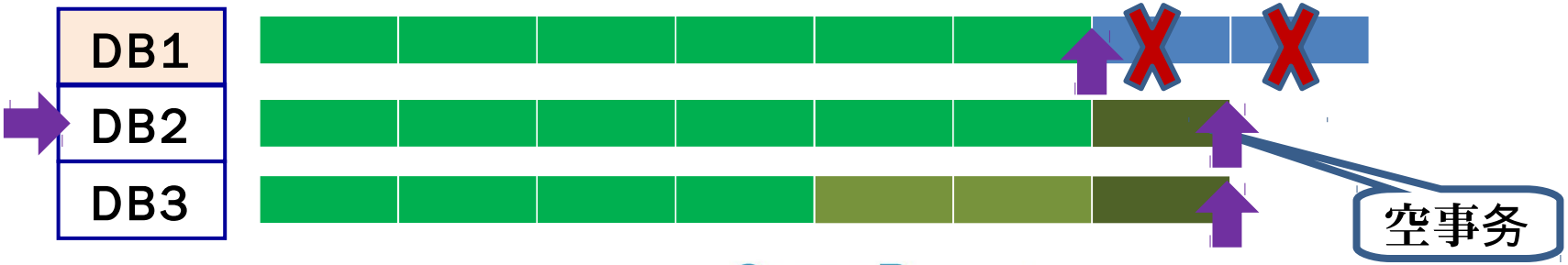
- 恢复过程可能被打断
- 未决事务：原主库 (DB1) 最后未同步的若干事务
  - 若原主库参与恢复，则未决事务被 commit ，否则未决事务被回滚

恢复末尾为什么要写空事务？

- DB1 故障， DB2 成主库，故障恢复后没有写入操作，再次故障，然后 DB1 上线
  - 不写空事务：

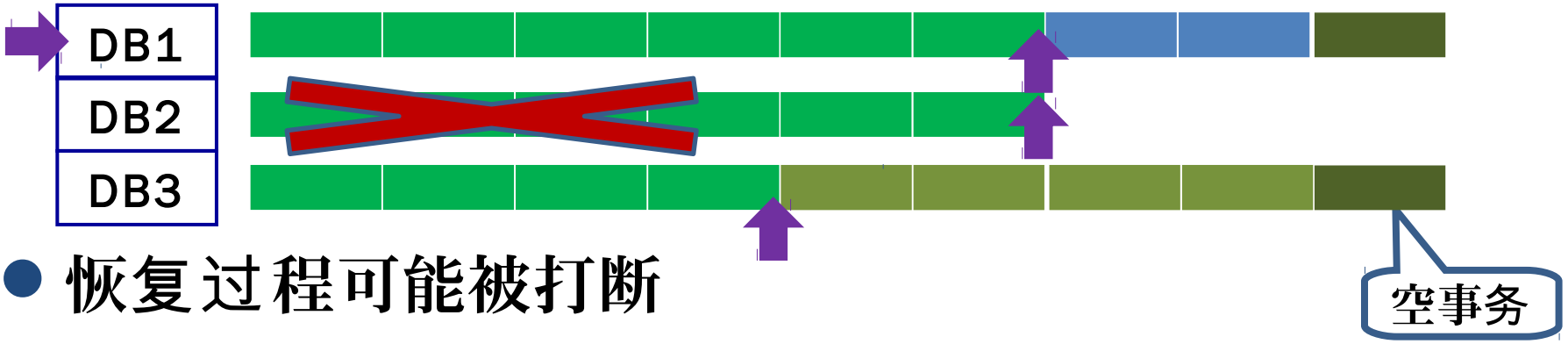


  - 写空事务：



按事务顺序同步 - 故障恢复 (2)

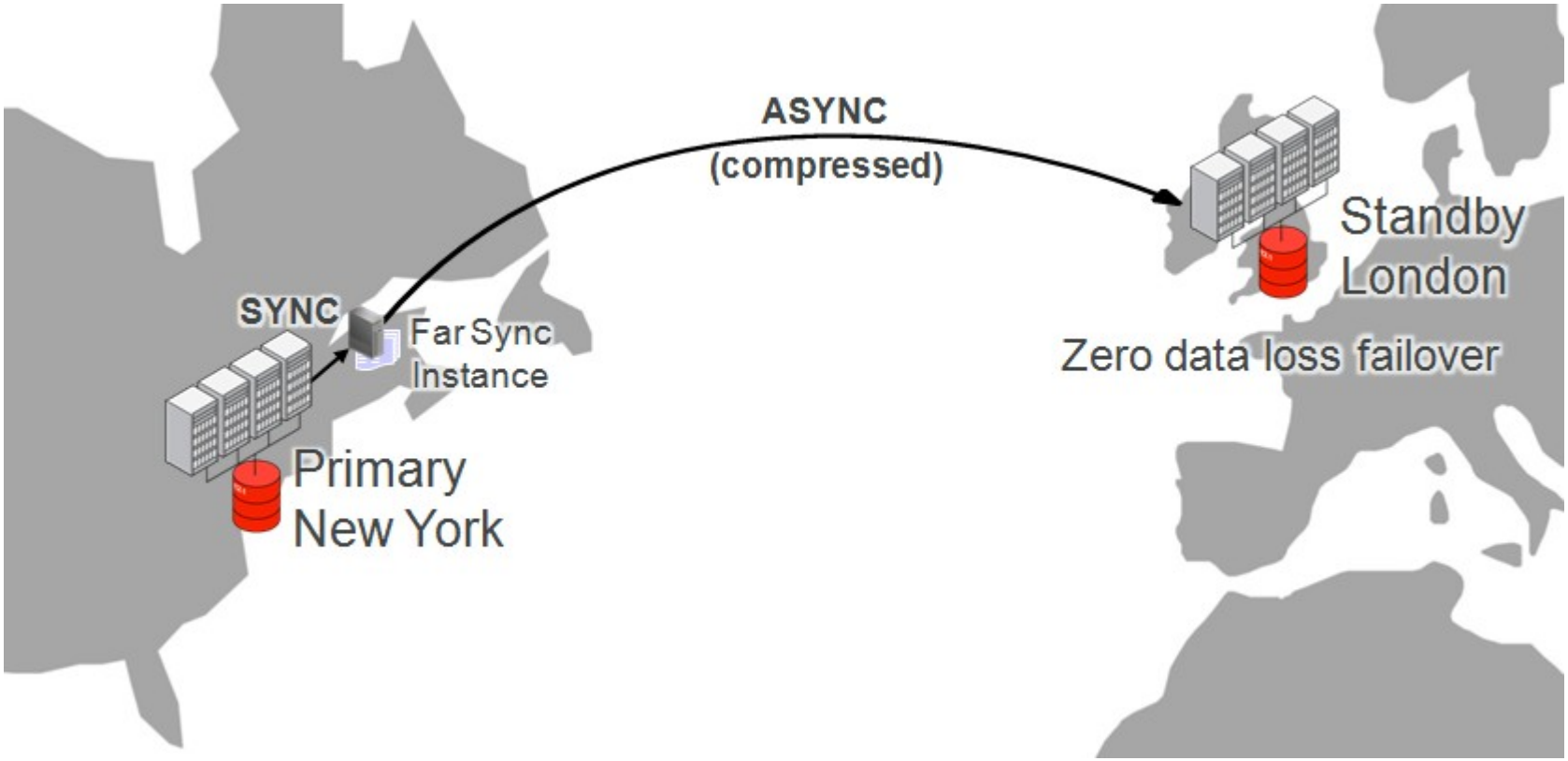
- 故障恢复 ： 1. 日志最新者为主， 2. 事务日志到达超半数的库， 3. 恢复过程中不对外服务
- Case 2 ：集群重启且 DB2 故障



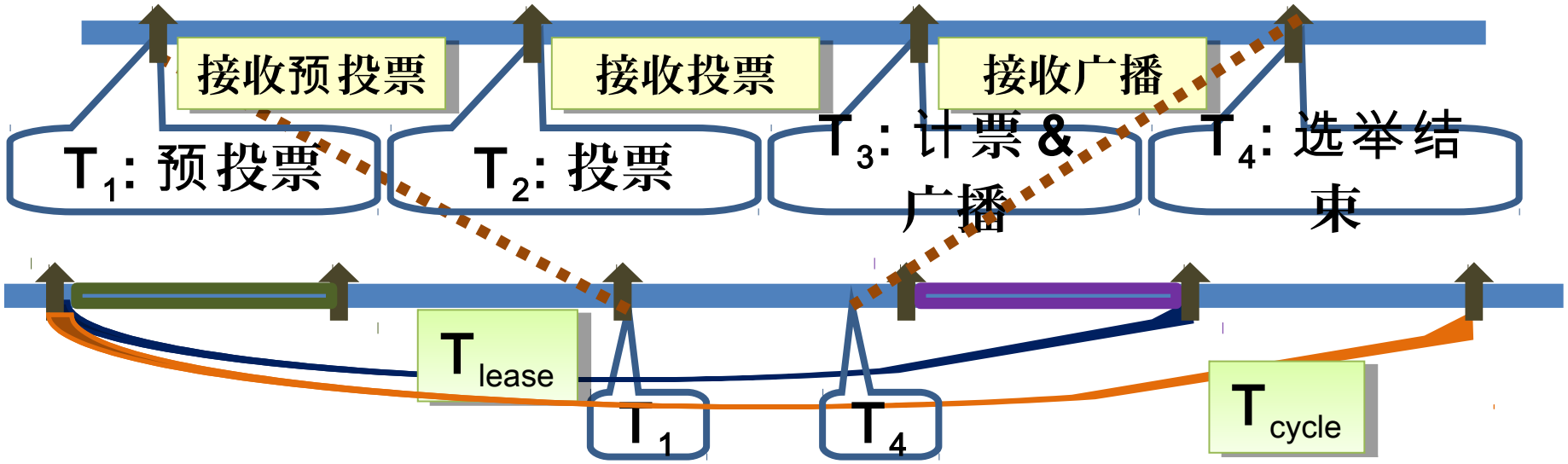
- 恢复过程可能被打断
- 未决事务 ：原主库 (DB1) 最后未同步的若干事务
  - 若原主库参与恢复，则未决事务被 commit ，否则未决事务被回滚

跨机房主备复制

- Oracle Far Sync ：主机实时复制日志到一台轻量服务器，后者再异步复制到远端备机
  - 提供同步模式的数据安全性，且性能更优

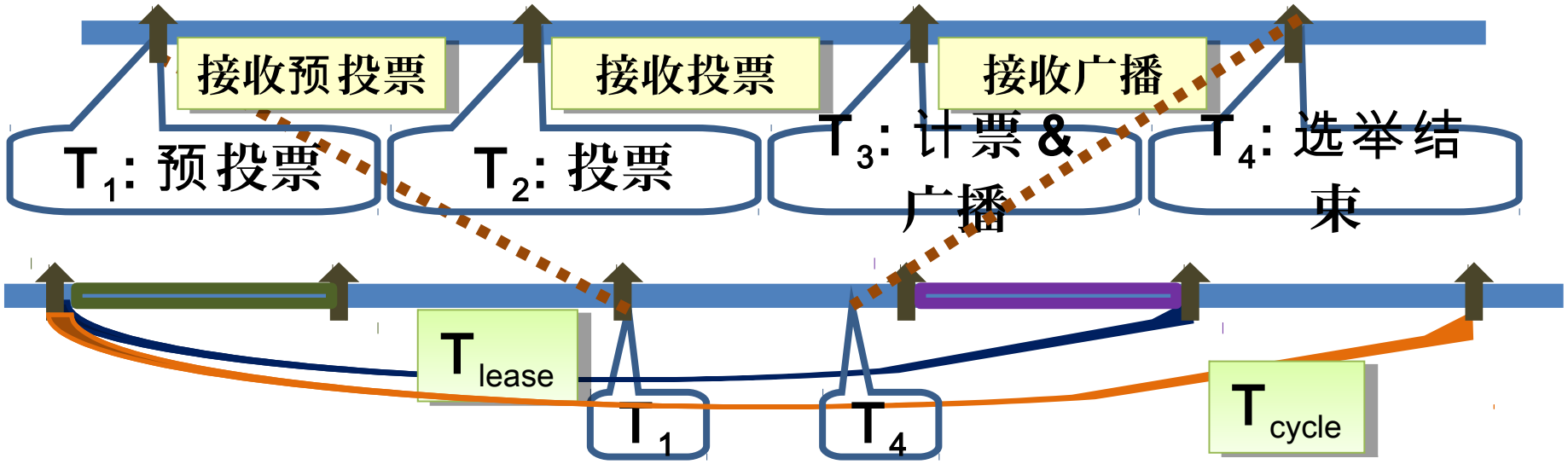


有主选举：连任



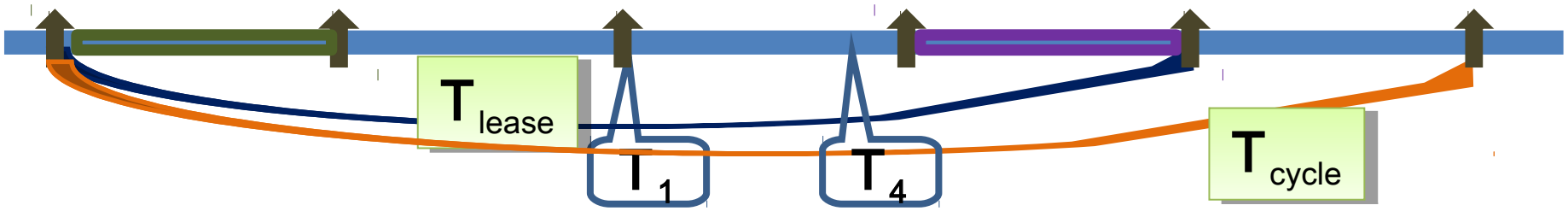
- Leader 连任
  1. 预投票：L1(当前 Leader) 在 T_1 时刻发起连任申请
  2. 投票：成员收到 leader 连任申请后在 T_2 时刻投票
  3. 计票 & 广播：T_3 时刻计票，得票超过半数则选举成功并广播，否则选举失败 (将在 lease 过期后进入无主选举)
  4. 选举结束：接收选举成功广播并在 T_4 时刻结束选举

有主选举：改选



- Leader 改选
  1. 预投票：L1(当前 Leader) 在 T_1 发起改选 leader 为 L2 的申请
  2. 投票：成员收到 leader 改选申请后在 T_2 时刻投票
  3. 计票 & 广播：T_3 时刻 L1 计票，若 L2 得票超过半数则 L1 卸任并广播 (含旧新 leader)，否则选举失败 (将在 lease 过期后进入无主选举)
  4. 选举结束：接收选举成功广播 (L2 收到广播后立刻把自己设置为 leader)，在 T_4 时刻结束选举

连任/改选时机



- Lease 从 T_1 时刻起，共 $m \times T_{elect2}(m>=4)$
  - Leader 自身扣除 lease 的最后一个 T_{elect2}
- Leader 在 lease 开始后 $2 \times T_{elect2}$ 时刻发起连任/改选