DTCC 2014中国数据库技术大会
DATABASE TECHNOLOGY CONFERENCE CHINA 2014
大数据技术探索和价值发现

TimesTen内存数据库

— 架构扩展应用实践

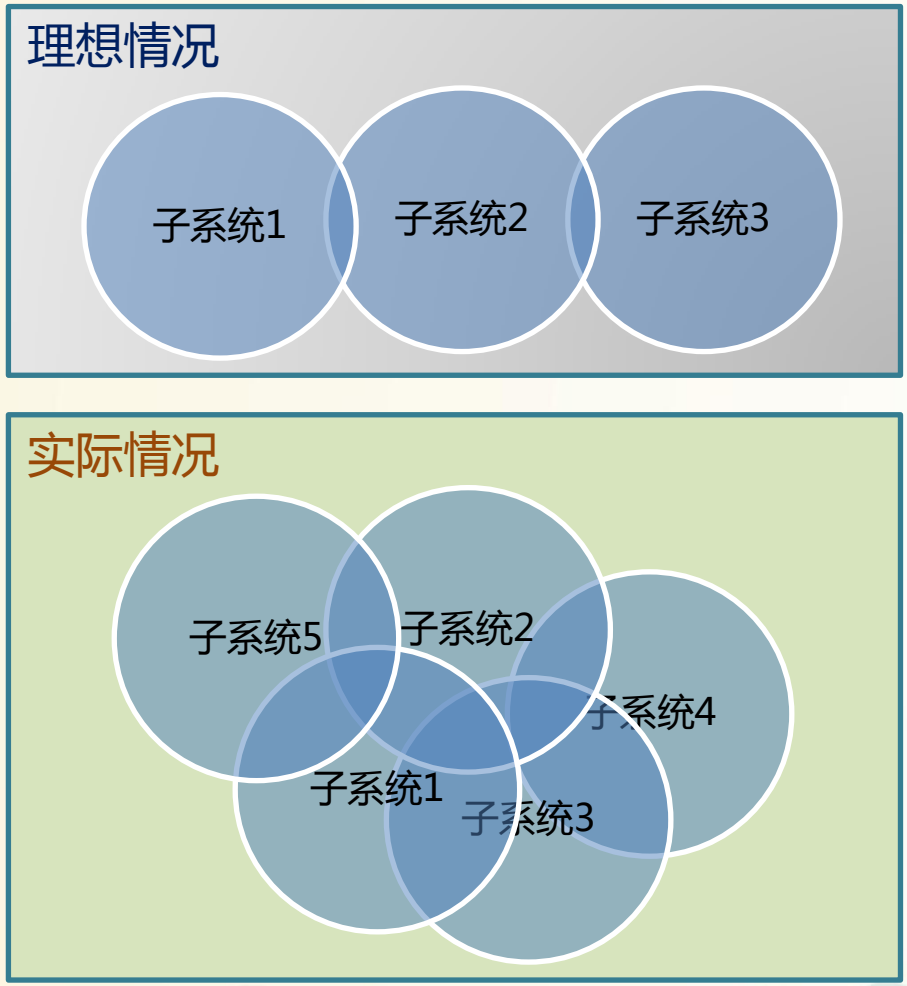
自我介绍

汪洋

- 从1998年开始接触Oracle数据库，并在1999年考获OCP 7.3证书，从此与Oracle数据库结下不解之缘，至今已有16年。
- 从2011年至今任职于平安科技（深圳）有限公司，当前负责数据库技术支持部的管理，负责生产、开发、测试数据库的运维工作，以及向开发部门提供应用系统的数据库架构设计方案，提供必要的数据库开发支持等。
- 在加入平安之前，供职于Oracle香港高级客户服务部门3年，为香港、澳门和深圳的客户提供Oracle数据库架构设计，升级方案制定，驻场支持等高级服务。期间也做为资深数据库专家参与解决香港和深圳重要客户的疑难问题分析和解决。

目录

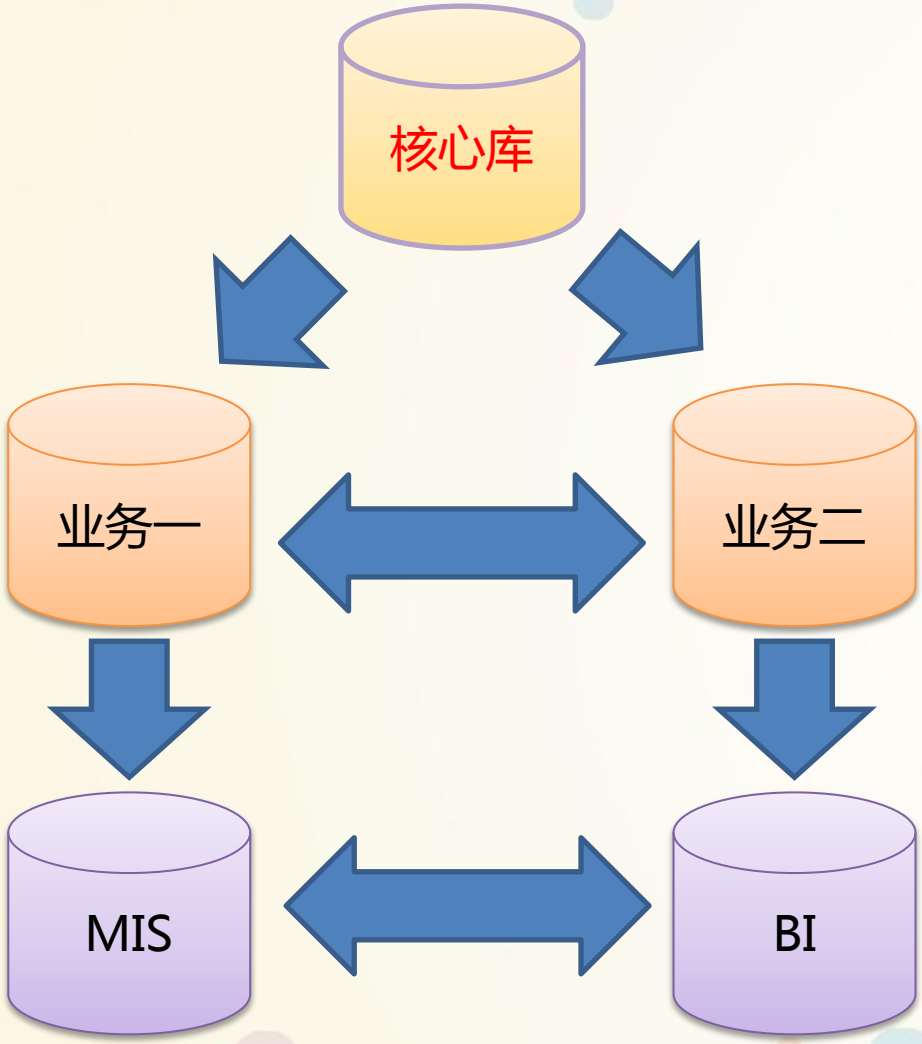
架构现状



应用系统架构

- 理想情况下的应用系统架构，子系统应该是相对比较独立的，子系统之间关联较少，而且相互关联的子字体数量相对较少。
- 实际情况往往是大相径庭的，子系统之间存在很高的耦合性。子系统内读写错综复杂，基本上不可能实现读写分离。
- 面对这样的现实，出于成本和风险的考虑，很难做到子系统的解耦，理想情况也只能是理想了。

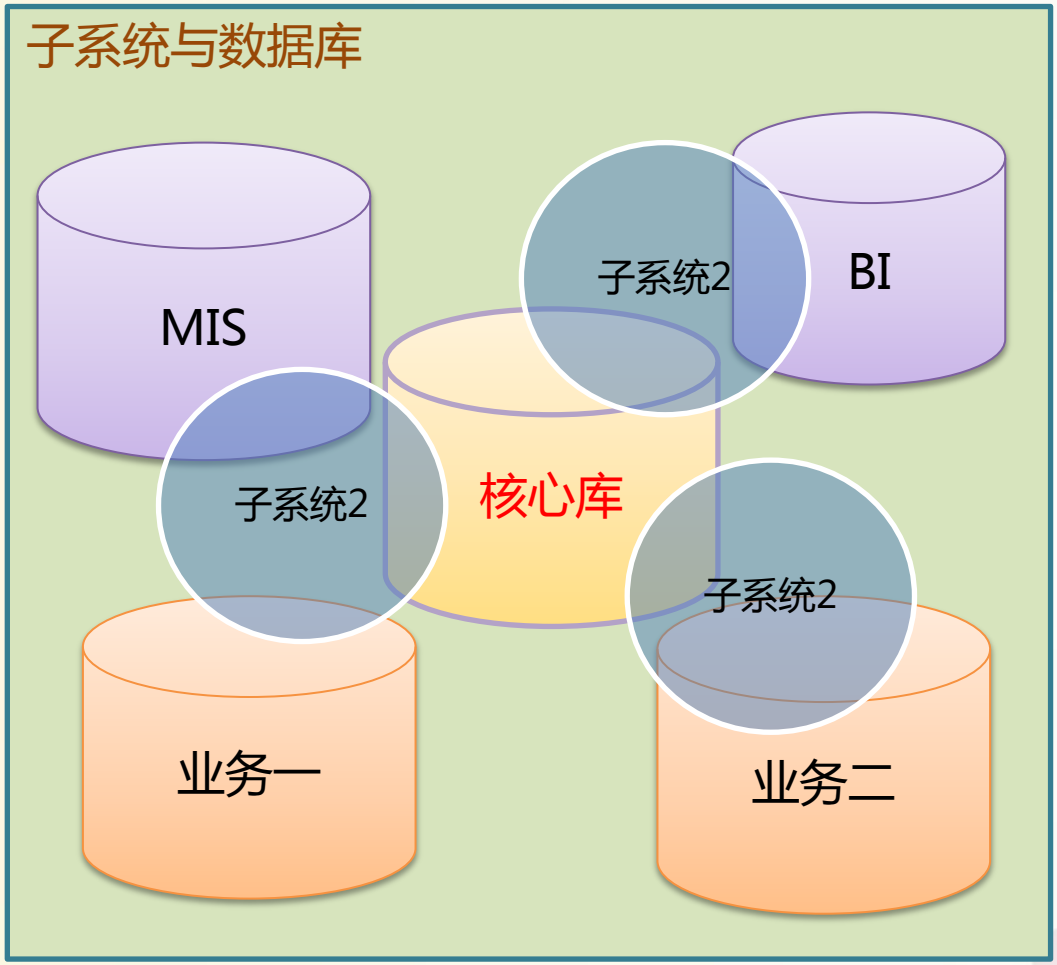
架构现状



数据库架构

- 对于一个系列的业务应用来说，肯定不可能通过单个数据库来实现的。需要的是一个数据库群。
- 包含核心库、业务应用库，以及各种功能性的库，根据重要性做层级划分。
- 数据库之间实现即时的数据同步，实现一套完整的数据库生态环境。
- 厚重的架构很难进行数据库体系的解耦。

架构现状



子系统与数据库的对应

- 在一套完整的数据库生态系统中，子系统和数据库也不是能独立开的。
- 理想情况是若干子系统分布在一个数据库中，数据库基本上是自包含的。
- 现实仍然是残酷的，往往是子系统和数据库出现多对多的关系。
- **现有架构做解耦、拆分、横向扩展只能是无动力驱动的高铁动车。**

架构现状

- ✓ 标准的三层架构
  - ➢ WEB服务器接受用户请求
  - ➢ 应用服务器封装业务逻辑
  - ➢ 数据库服务器负责数据查询和处理
- ✓ 数据库的负载
  - ➢ 应用过来的所有负载
  - ➢ 磁盘的IO问题
  - ➢ 高并发问题

架构现状

1. 数据库容量扩展
2. 高并发处理能力
3. 拒绝架构革命
4. 保证用户体验

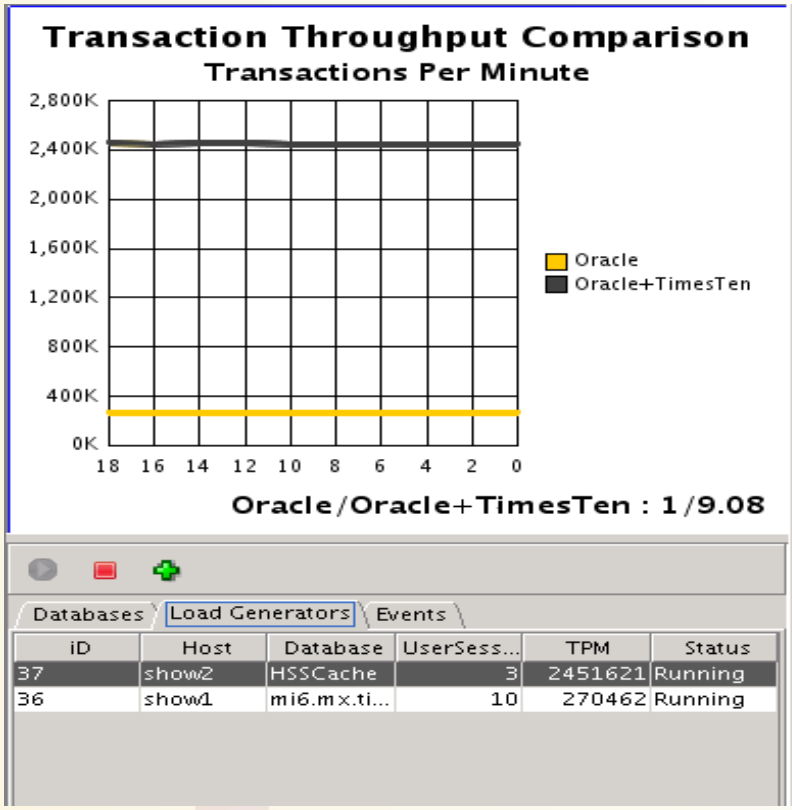
目录

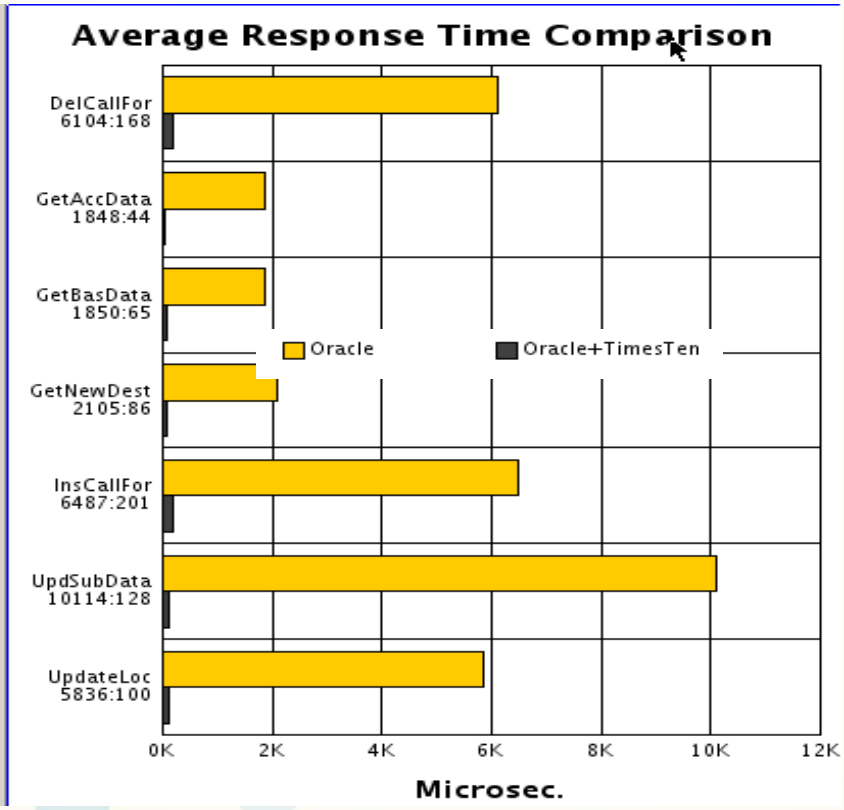
- 架构现状
- TimesTen
- 个性定制
- 应用效果
- 其 他

Timesten

Oracle VS. TimesTen

- 对于一些并发和响应速度要求较高的应用场景来说，TimesTen有明显的响应速度和即时吞吐量的优势；
- TimesTen小巧快捷，独立使用优势不大，应该尽可能与Oracle联合使用；
- TT支持Direct和C/S两种连接模式：Direct模式可作为中间件部署，C/S模式可作为独立数据库部署。





Timesten

In-Memory Database or Database In-Memory

内存数据库基本判断规则

- 规则 #1：内存数据库是整体装载在内存中的
- 规则 #2：内存数据库不存在复杂的代码去判断数据的位置并访问数据，因此可以获得更低的延迟和更快的响应时间

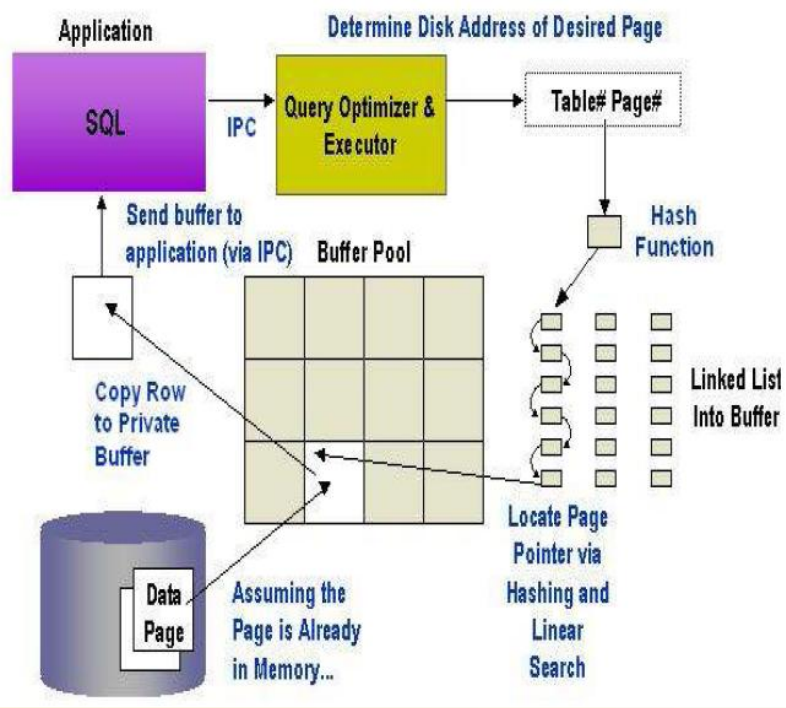
它们是内存数据库吗？

- 运行在DRAM中的数据库（Timesten）
- 运行在NAND Flash Memory中的数据库
- Oracle Exadata X3 Database In-Memory Machine

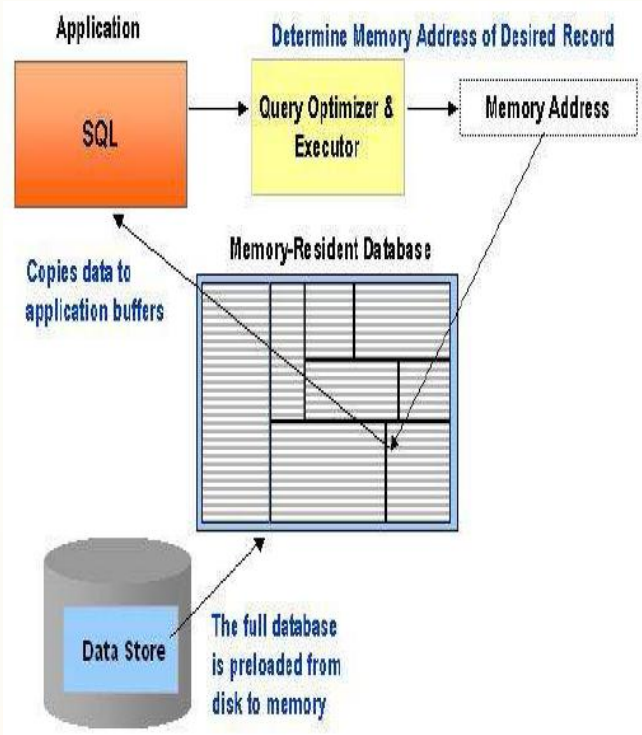
Timesten

TimesTen优势的秘密

- 完全内存存储和内存读写，没有缓冲缓存管理开销；
- 更短的SQL路径导致更快的性能，更少的CPU指令导致更少的CPU开销。

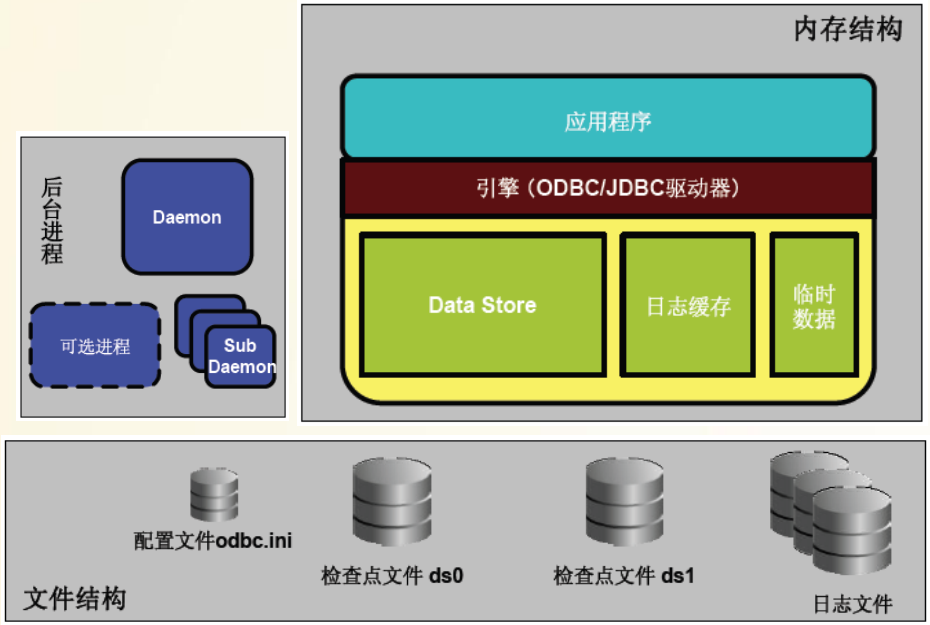


Oracle数据库



TimesTen数据库

Timesten



内存结构

- Data Store
  - 保存所有数据库数据的区域
- 日志缓存
  - 用于暂时存储记录Data Store变更的日志
- 临时数据区域
  - 临时存储执行计划等数据的共享区域
  - 排序等等操作临时使用。

文件结构

- 配置文件odbc.ini：用于记录各个DSN的参数
- 检查点(Checkpoint)文件
  - 保存于磁盘的数据库镜像。
  - 启动时，检查点文件的数据被装载到内存中
  - 运行时，隔一段时间进行一次检查点处理，仅保存改变的数据块，并删除无用的日志文件
  - 关闭时，用于保存Data Store内的数据
- 日志文件
  - 保存数据库的变更
  - 有文件超过一定的大小，自动生成新的日志文件
  - 与检查点文件一起用于数据库的恢复

后台进程

- 主进程(Daemon)：
  监听功能(Listener)；读取配置文件odbc.ini；分配和监视子进程。
- 子进程(Sub Daemon)：
  载入/卸载Data Store；将日志缓存写入日志文件；监视和解除死锁；执行检查点。

Timesten

TimesTen的Cache Group选择

Read only

1. 每隔一定时间，缓存代理将数据复制到TimesTen中；
2. 适合用于储存几乎没有更新的数据；
3. 能保证单一数据库源的写入，能更好控制数据一致性，重点推荐。

AWT

1. 异步写入，DML被抛给Oracle，不等待Oracle完成COMMIT；
2. 适合希望通过TimesTen加速DML操作的情况；
3. 尽可能只在TT单一数据库源进行写入，保证数据一致性，值得推荐。

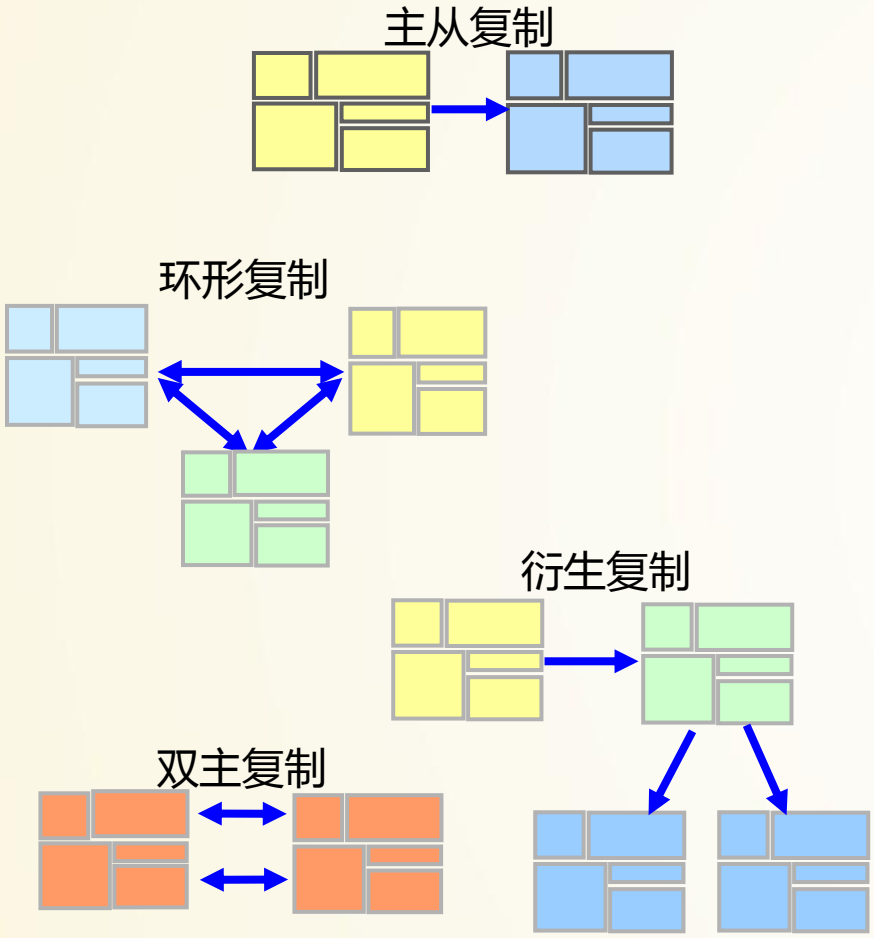
SWT

1. 同步写入，DML被抛给Oracle，等待Oracle完成COMMIT；
2. 适合仅希望通过使用TimesTen加速SELECT操作的情况；
3. SWT很难保证数据单一来源和数据一致性，甚至可能造成新的性能问题，不推荐。

自定义

1. 用户选择自动或手动控制Oracle与TimesTen之间的数据同步；
2. USERMANAGED不推荐。

Timesten



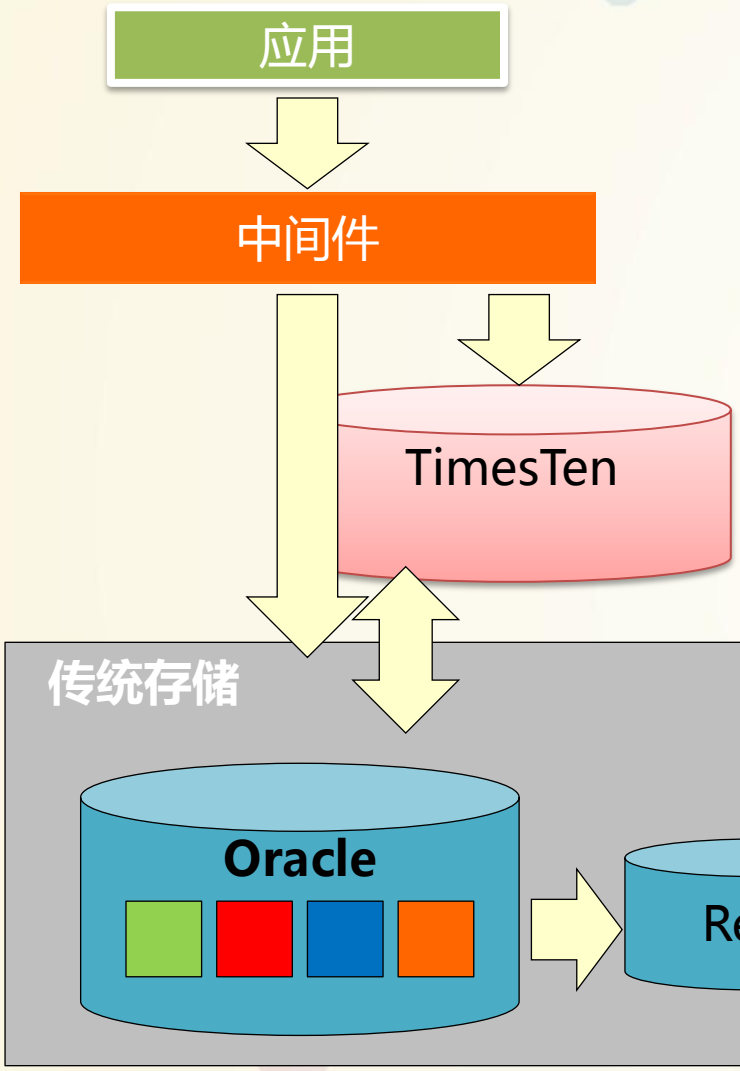
TimesTen高可用架构

- TimesTen支持多种容灾复制方式：
  - 主从复制（Active-Standby）
  - 双主复制（Active-Active）
  - 环形复制
  - 衍生复制
- 架构设计原则
  - 主从复制方式足够了；
  - TimesTen本身就是轻量级的，简单至上；
  - 不要期望TimesTen能像Oracle一样强大。

目录

- 架构现状
- TimesTen
- **个性定制**
- 应用效果
- 其 他

个性定制



架构演变

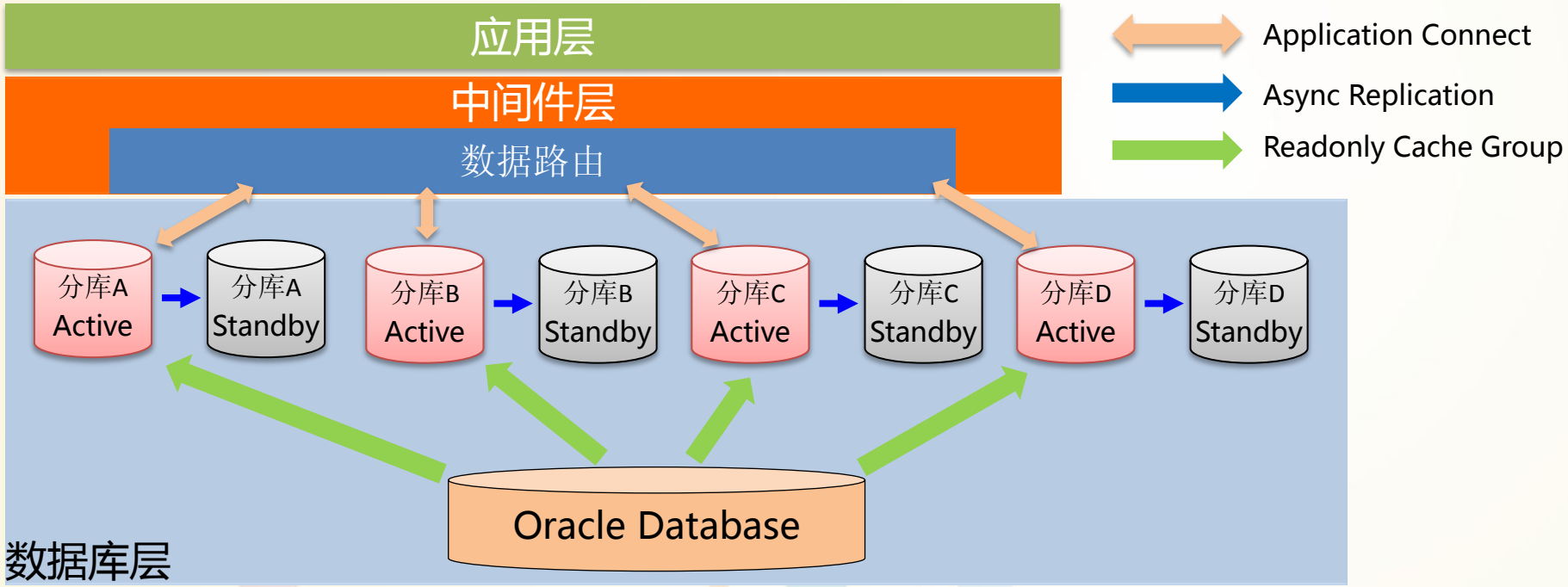
- 之前为Oracle数据库全部存储在SAN传统存储设备上。中间件直连Oracle数据库，对应用提供服务。
- 引进Flash存储设备，将Redo日志和部分热点数据文件转存其上，获取较好的IO性能。
- 引进TimesTen内存数据库，作为数据库中间件置于中间件与Oracle数据库之间，数据在TT和Oracle之间快速同步。
- 将部分需要快速响应的业务分流到TimesTen上，彻底缓解Oracle数据库压力。

Flash存储

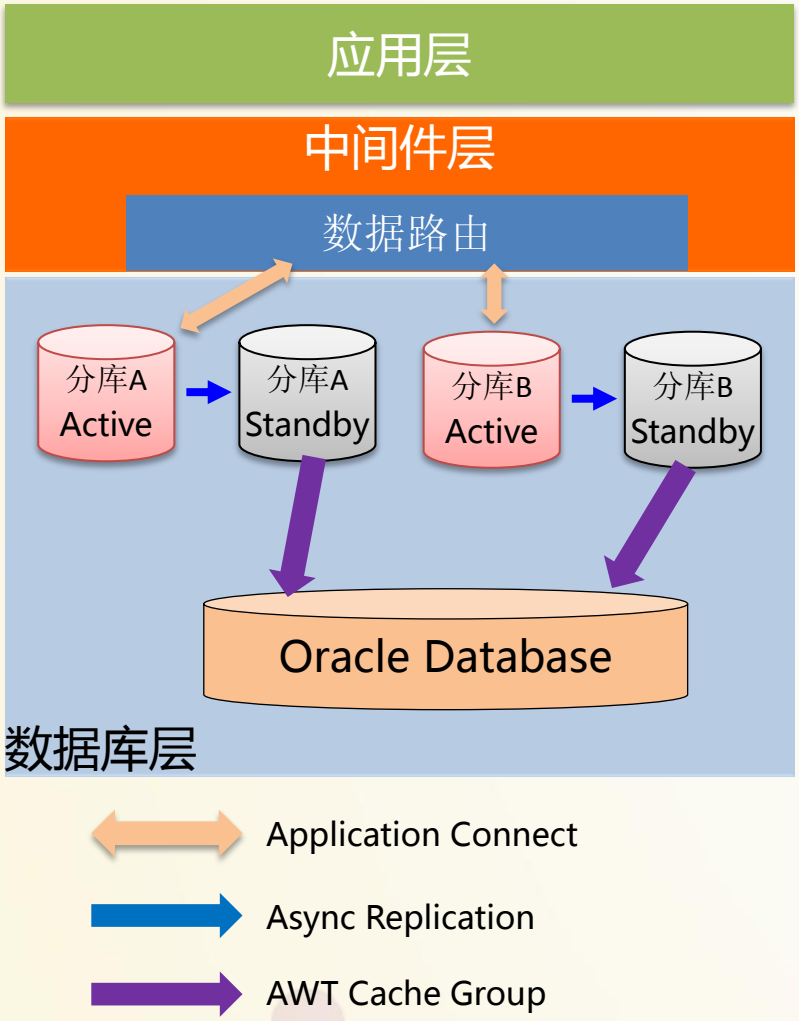
个性定制

TimesTen只读架构

- 采用**Client/Server**连接模式，将TimesTen数据库部署在数据库层；
- 采用**只读Cache Group**，Oracle作为唯一数据来源，**半即时（3秒）同步**数据到TimesTen，保证数据强一致性。
- TimesTen拆分为**四个分库**，在中间层架构一个数据路由组件，按照sharding键值找到对应数据位置。
- 每个分库控制数据量在**100GB**以内，且各有一组**ASP高可用**架构，实现应用的自动Switchover和Failover。



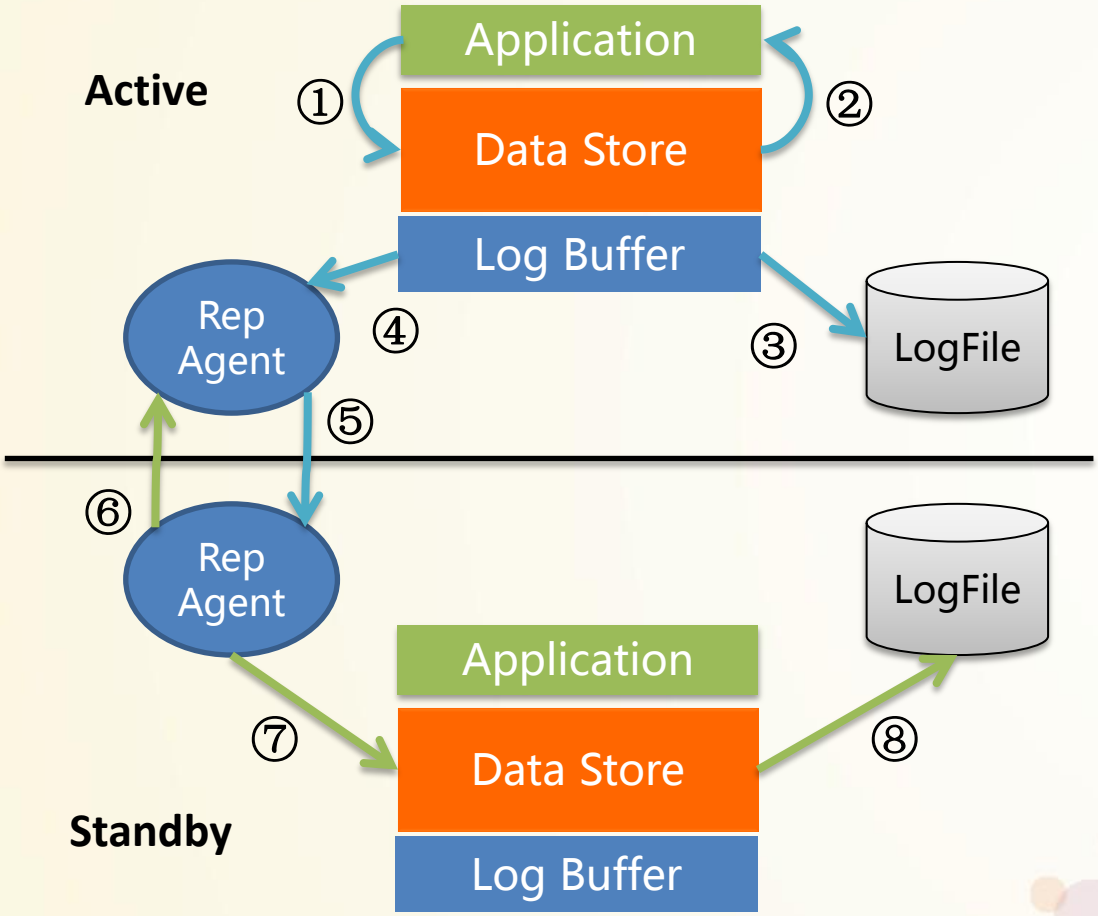
个性定制



TimesTen可读写架构

- 可读写架构和只读架构最大的区别在于采用的Cache Group模式和数据来源。
- 可读写架构仍保证数据单一来源，即此时TimesTen的表可读写，Oracle对应表为只读。
- 仍采用**Client/Server**连接模式，将TimesTen数据库部署在数据库层；
- 采用**AWT Cache Group**，TimesTen作为唯一数据来源，**半即时同步**数据到Oracle，保证数据强一致性。
- 中间层实现数据路由，按照sharding键值找到对应数据位置。
- 同样每个分库控制数据量在**100GB**以内，且各有一组**ASP高可用**架构，实现应用的自动Switchover和Failover。

个性定制



ASP架构原理

① 应用COMMIT提交；

② 立刻返回应用COMMIT提交成功；

③ 通过日志缓存写入日志文件；

④ 将更新信息发送给Replication Agent；

⑤ 将更新信息从Active发生到Standby的Replication Agent；

⑥ 确认更新信息接收成功；

⑦ 应用更新信息，并写入Data Store；

⑧ 记录Standby的日志文件。

个性定制

IO

- ✓ 内存数据库往往被误解和磁盘IO没有关系。
- ✓ TimesTen的数据文件和日志文件均存储在磁盘上，与内存保持同步，如果磁盘IO不佳会成为短板，尽可能使用SSD盘等高效IO的存储。
- ✓ 并发高， TimesTen更易导致CPU问题。

内存

- ✓ TimesTen中，表和索引均保存在内存中，容量（内存）估算很重要，单库尽可能控制在50~60GB，不超过100GB。
- ✓ 表字段不宜过多，字段不宜过长，TimesTen存储具有字段对齐的特点，比Oracle需要更多的存储空间，尽可能不超过128字节。

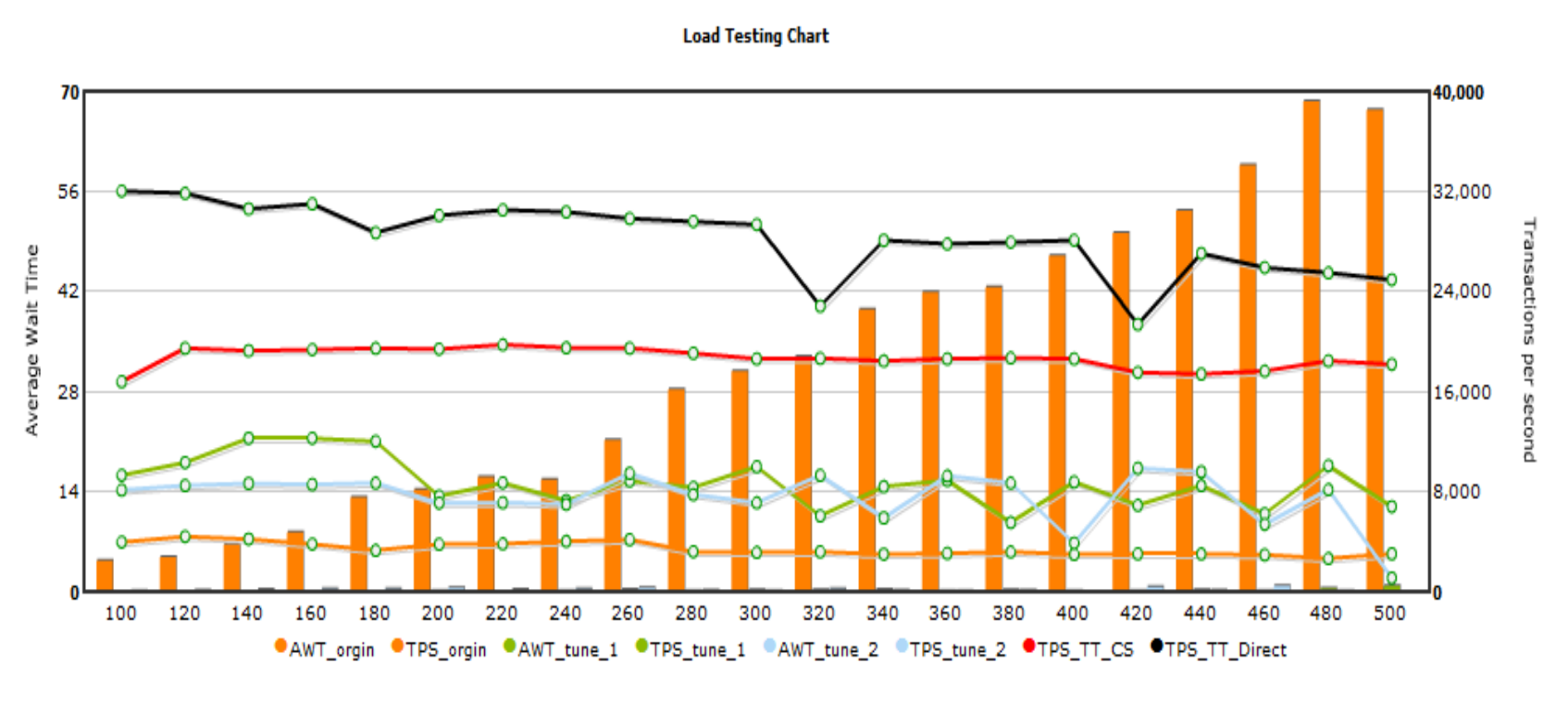
目录

- 架构现状
- TimesTen
- 个性定制
- **应用效果**
- 其 他

应用效果

单体压力测试

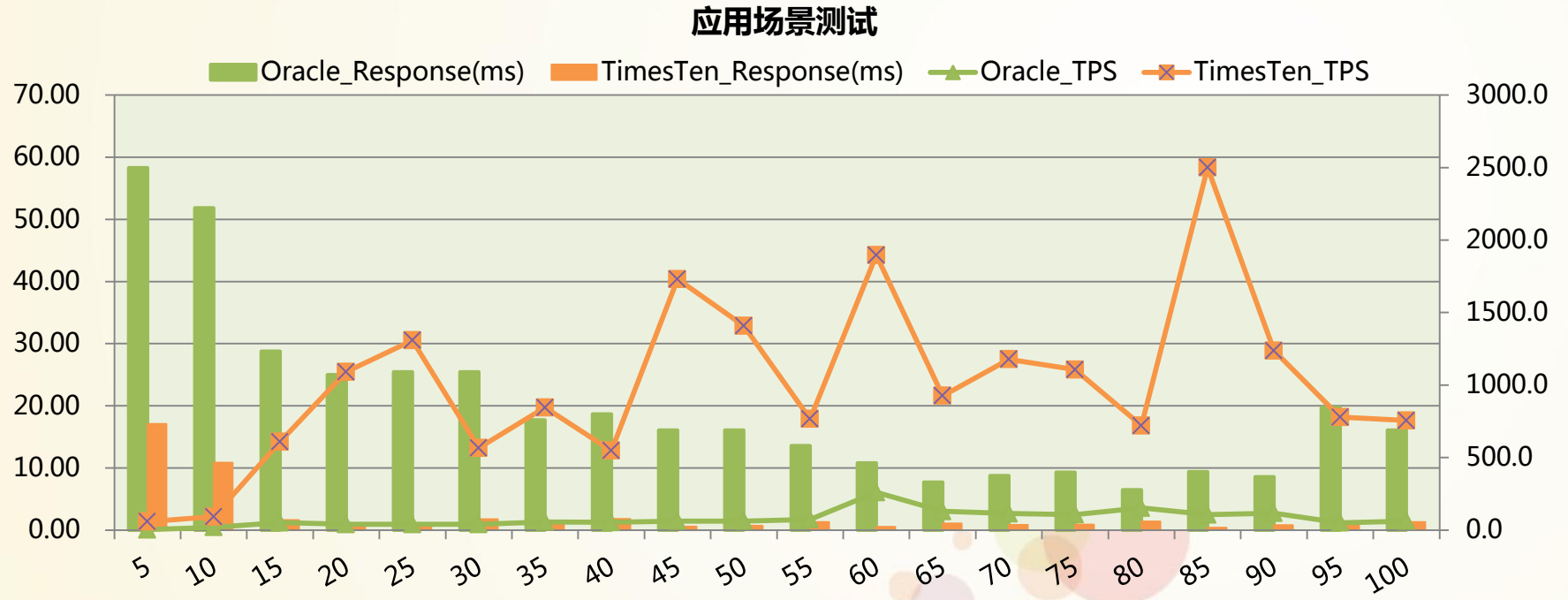
- TPS曲线从低到高依次为：橙色、绿色、蓝色、红色、黑色，分别对应Oracle原始、Oracle优化一、Oracle优化二、TimesTen的C/S模式、TimesTen的Direct模式。
- 与Oracle不同，TimesTen在进行单体的读写操作压力测试时，具有更高的吞吐量和更好的稳定性。



应用效果

TimesTen应用效果

- 将部分高并发场景独立到Timesten数据库进行POC测试，有**5~10倍**不等的性能提升。
- 按场景需求分流Oracle的压力，提高响应速度，标本兼治，一劳永逸。
- 值得注意：在实际使用中，TimesTen对CPU依赖很高。



目录

- 架构现状
- TimesTen
- 个性定制
- 应用效果
- **其 他**

Timesten使用注意事项

- ✓ Backing Storage SSD
- ✓ Checkpoint Frequency
- ✓ Simplified SQL statement
- ✓ Optimizer Statistics Collection
- ✓ Refresh frequency

Timesten的展望

- ✓ 分担负载，提高系统扩展性
- ✓ 隔离影响，提升系统可用性
- ✓ Timesten vs. 12c in-memory option
  - ➢ Timesten is desinged for OLTP system
  - ➢ As database cache close to APP server
  - ➢ 12c in-memory is Columnar format
  - ➢ 12c in-memory is a built-in feature in RDBMS

Timesten文档获取

- Timesten最佳实践

  http://download.oracle.com/otn_hosted_doc/timesten/1122/quickstart/html/best_practices/bp.html

- Timesten

  https://community.oracle.com/community/developer/english/oracle_database/timesten_in-memory_database

- Timesten博客

  https://blogs.oracle.com/timesten/

  http://ggsig.blogspot.com

- Timesten官方文档

  http://www.oracle.com/technetwork/database/database-technologies/timesten/documentation/index.html

- Timesten MOS Master Note

  INDEX : Oracle TimesTen In-Memory Database (Doc ID 1088128.1)

- 感谢：IT168、ITPUB、DTCC
- 感谢：中国平安DBA团队
- 感谢：在座的每一位伙伴

Q&A

THANKS