ORACLE

Современные тенденции развития коммерческих СУБД

и некоторые заблуждения

Марк Ривкин
SC Director
Oracle CIS
24 февраля 2021

О чем будем говорить

Сначала намечались торжества, потом аресты, потом решили совместить

Фильм “Тот самый Мюнхаузен”

Тенденции развития современных коммерческих СУБД

Недопонимание терминологии и состояния технологий Oracle

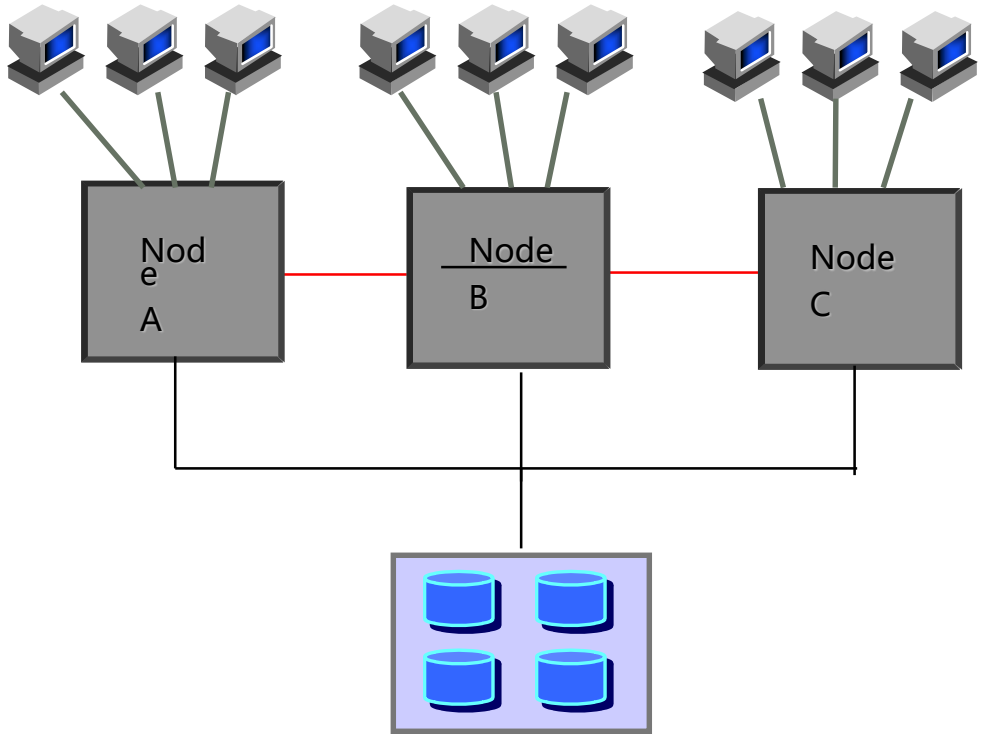
- RAC vs Exadata
- 6 In-memory технологий Oracle и их назначение

Real Application Cluster (RAC) – это не Exadata

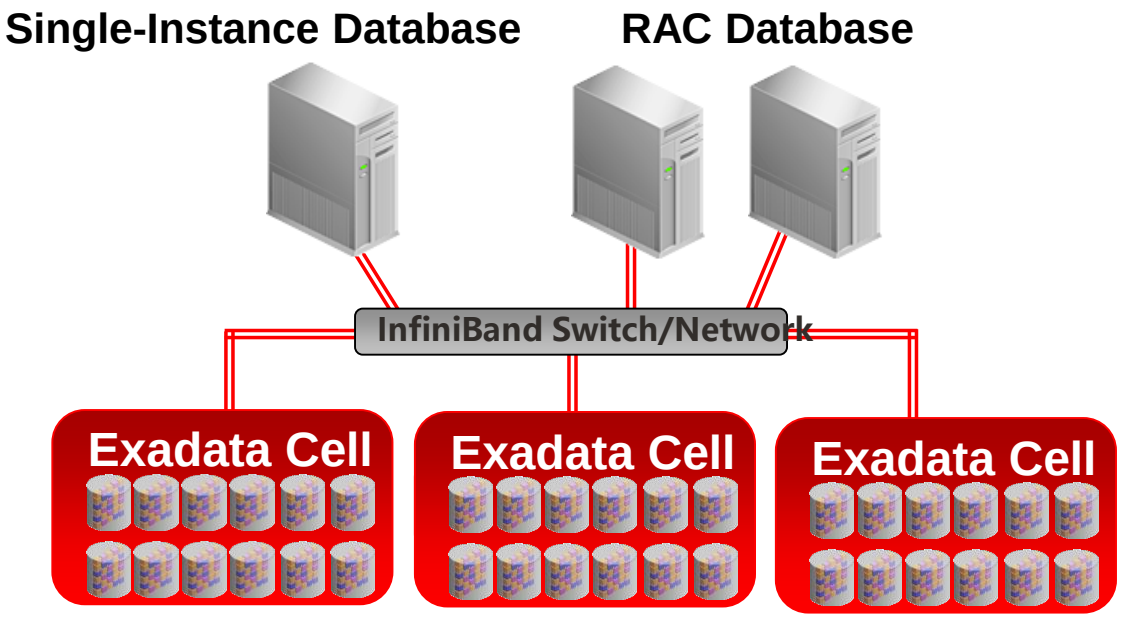
- Exadata – машина баз данных для OLTP и аналитики, специальная архитектура, сплав HW и специального SW, умные ячейки
- Узлы СУБД могут использоваться в кластере, а могут и для разных БД

- RAC – архитектура СУБД shared everything, может работать на ЛЮБОЙ платформе (x86, AIX, SPARC, HP UX, Exadata) – ей уже > 25 лет

- Для чего:
  - Надежность
  - Масштабируемость
  - Производительность
  - Распараллеливание
  - Кэш фьюжен, RDMA
  - Гибкость
  - Балансировка нагрузки
  - TAF и Application Continuity
  - Сервисы

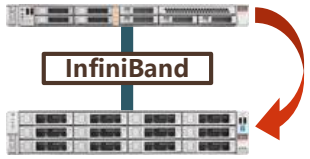


Машина баз данных Exadata



- Каждая ячейка Exadata – самостоятельный сервер с установленными дисками и ПО Exadata
- Данные «размазаны» между многими ячейками Exadata
- Толстый канал, 2 у каждой ячейки, offloading SQL
- Сейчас infiniband заменяем на ROCE
- В ячейках Flash и PMEM для OLTP

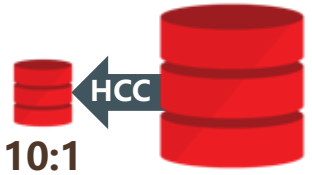
Технологический фундамент Exadata



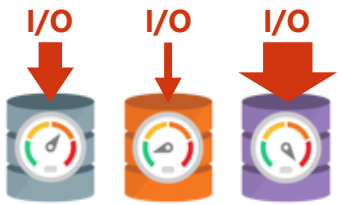
Перенос вычислений на СХД

Индексы на СХД

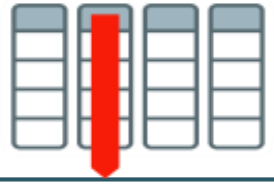
Кэширование во ФЛЭШ и PMEM



Hybrid Columnar Compression

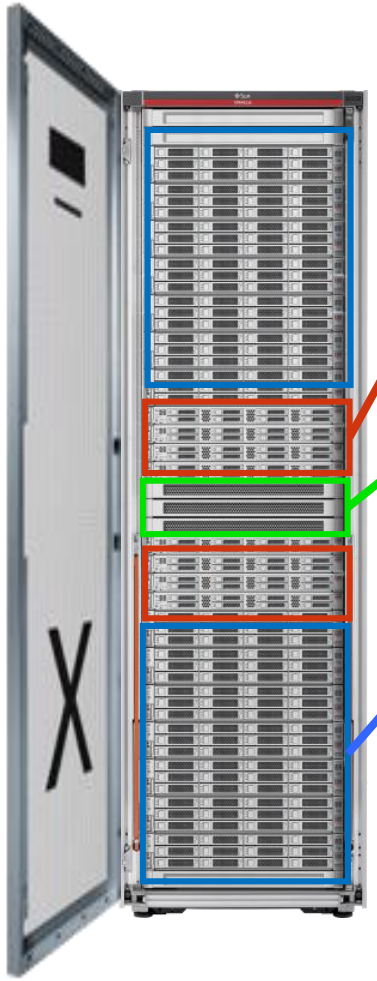


Менеджер ресурсов



Технологии СУБД In-Memory

Exadata X8M



Серверы БД 2 или 8 сокетные (2x8 или 8x2)

24 core Intel Cascade Lake процессоры

100Gb RDMA over Converged Ethernet (RoCE) Internal Fabric

В ячейке 2-сокетные Storage Servers

1.5 TB Persistent Memory per storage server

Три слоя хранения: PMEM, NVMe, HDD

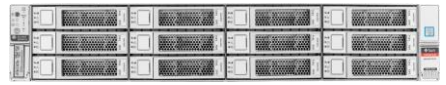
X8M-2 полная:

27 Tb PMEM, 920 Tb Flash, 3Pb HDD

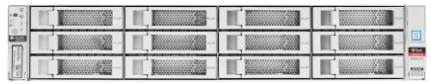
912 ядер, 28,5 Tb памяти

Database Server

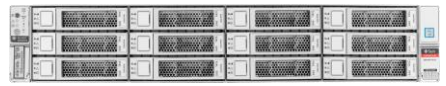
High-Capacity (HC) Storage



Extreme Flash (EF) Storage



Extended (XT) Storage



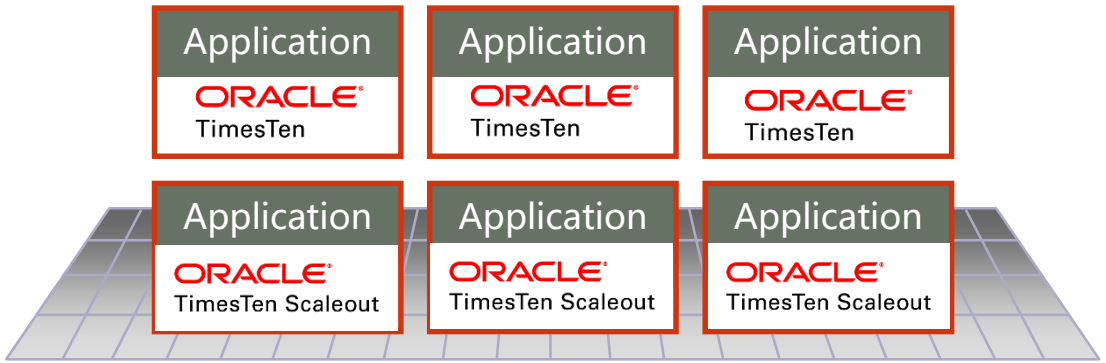
In-memory в Oracle

- **Oracle TimesTen**
  - **встроенная СУБД**
  - **зонтик над Oracle DB – для OLTP**
- **TimesTen Scaleout - масштабирование, OLTP**
- **In-memory option в Oracle DB – Аналитика**

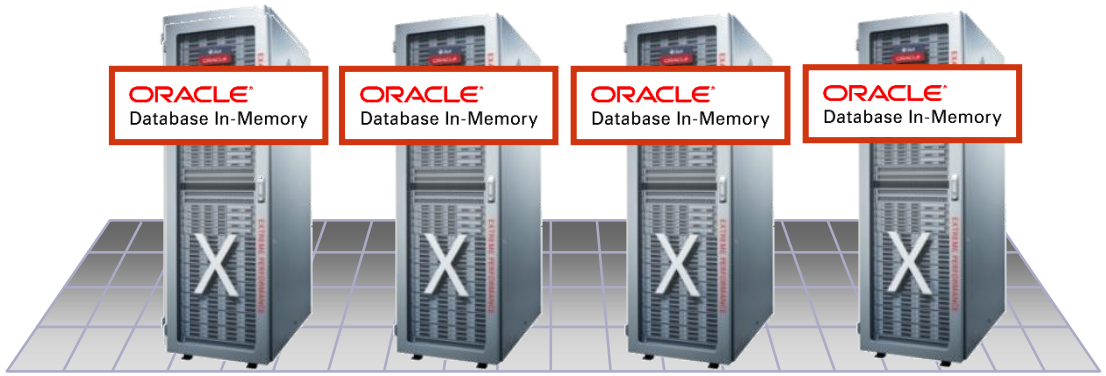
- **PMEM как расширение оперативной памяти**
- **PMEM для DB**
- **PMEM кэш в ячейках Exadata**

4 варианта In-Memory БД (пока DRAM): Для OLTP и Аналитики

In-Memory для OLTP



In-Memory для Аналитики



Oracle TimesTen In-Memory Database

- Легковесная, высокодоступная IMDB
- Предназначение: **Экстремальные OLTP нагрузки**
- **Микросекундное** время отклика
- **Миллионы TPS** на обычном оборудовании

Oracle Database In-Memory Option

- IMDB с двойным форматом хранения данных
- Предназначение: **Real Time Analytics**
- Скорость сканирования **Миллиарды строк в секунду**
- Ускорение промышленных OLTP систем со смешанной нагрузкой
  - Требуется меньше индексов для поддержки

Oracle TimesTen In-Memory Database

Несколько вариантов развёртывания

TimesTen Classic

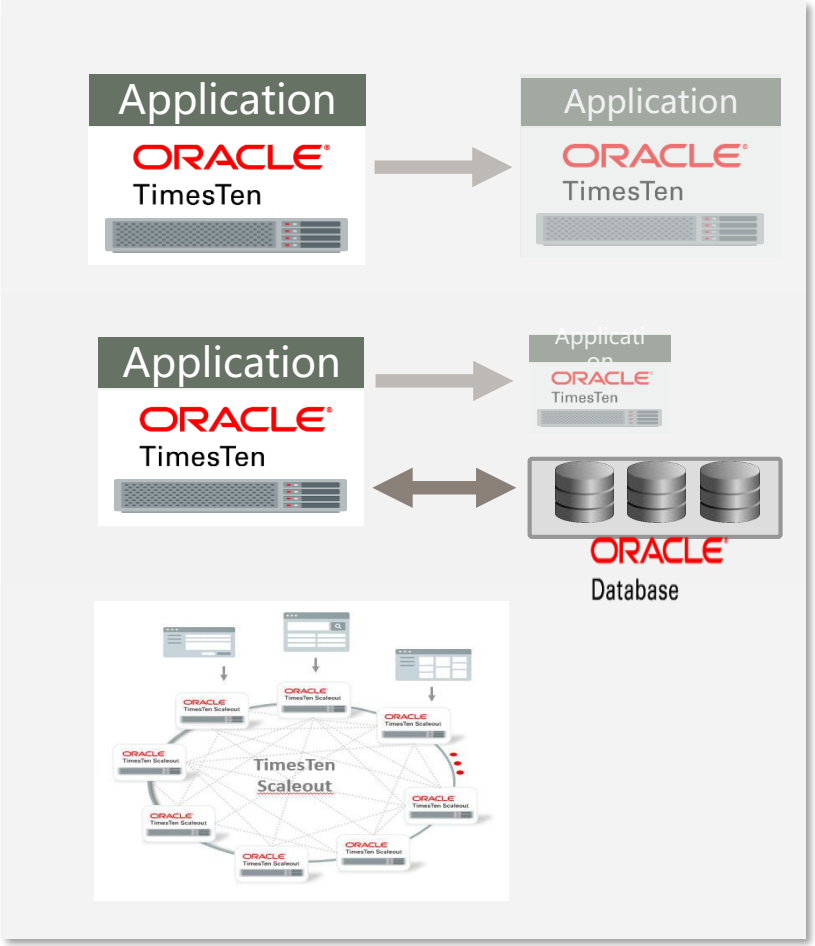
1. Одиночная / С Репликами реляционная IMDB
   - Приложения с низкой задержкой
   - ISV/OEM встраиваемые решения
2. Кэш для Oracle Database
   - Ускоряет Oracle Database OLTP приложений
   - HA с помощью Replication

Микросекундное время отклика, пропускная способность - миллионы TPS

TimesTen Scaleout

3. Распределённая реляционная IMDB
   - Высокая пропускная способность и большой объём данных
   - Прозрачное распределение данных
   - Эластичная масштабируемость
   - Отказоустойчивая

Сотни миллионов транзакций в секунду



Отличный от дисковых СУБД механизм хранения, индексирования, доступа, обработки

TimesTen Classic

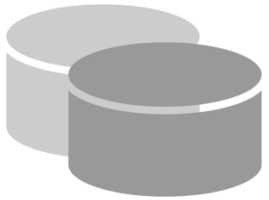
Реляционная База Данных

- Чисто in-memory база данных
- Поддержка ACID
- Стандартный SQL, PL/SQL
- Вся база данных в RAM
- JDBC, ODBC, OCI

Чрезвычайно Быстрая

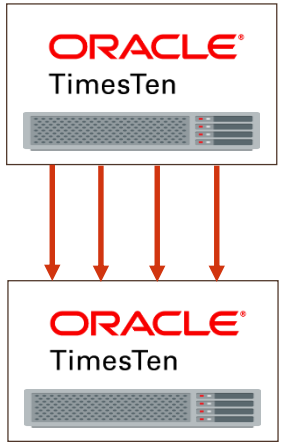
- Микросекундное время отклика
- Очень высокая пропускная способность

Персистентность и Восстановление



- База данных и журналы транзакций сохраняются на локальном диске или на флэш дисках
- Автоматическое восстановление после сбоя

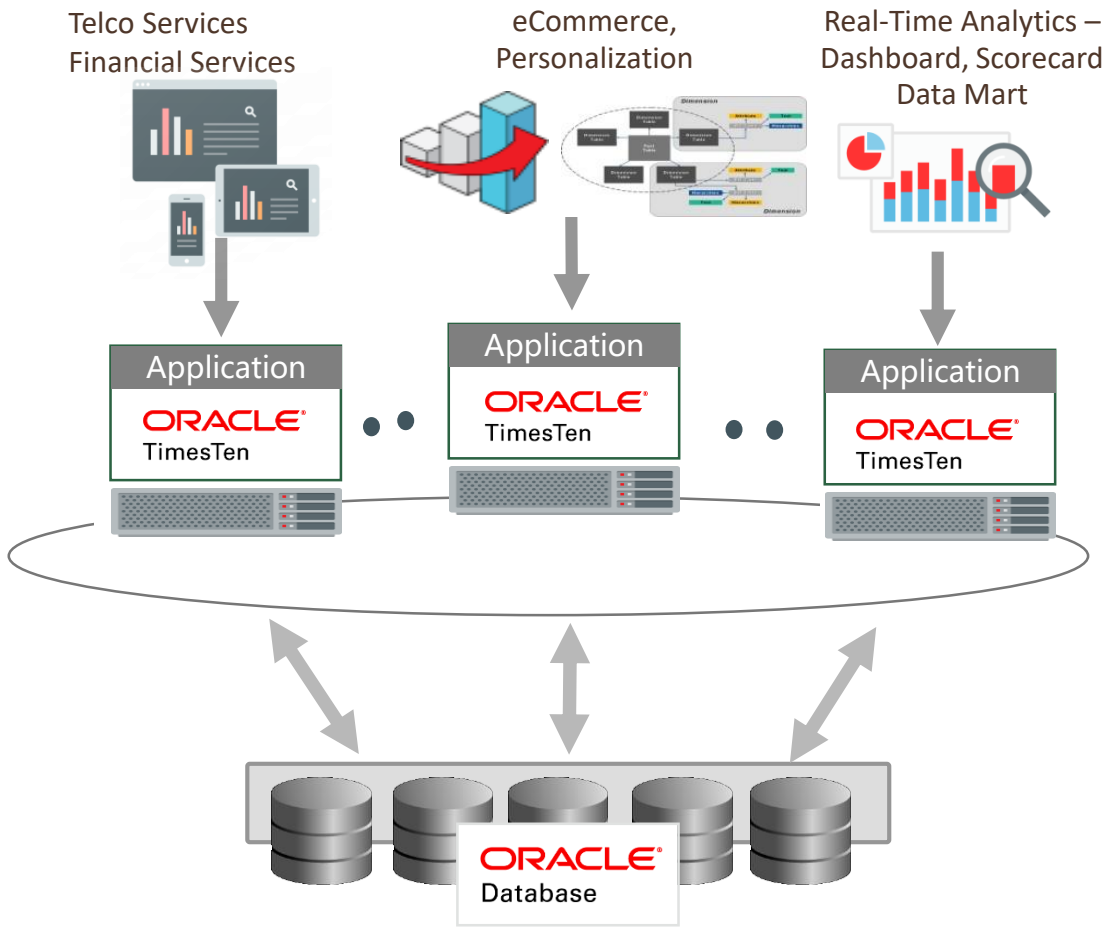
Высокодоступная



- Active-Standby и multi-master репликация
- Очень высокая производительность параллельной репликации
- HA и восстановление после аварий

TimesTen Application-Tier Database Cache

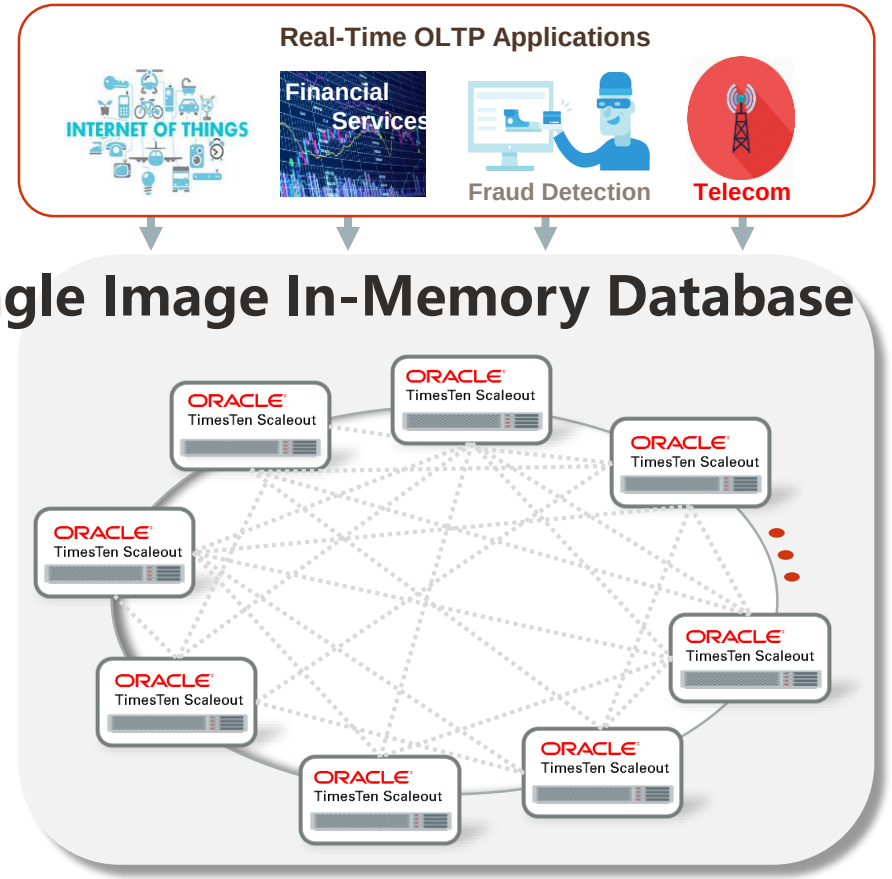
Для Oracle Database



- Кэширует подмножество таблиц Oracle Database в TimesTen для уменьшения времени отклика
  - С полной персистентностью на локальных дисках
- Read-write кэширование
  - Транзакции выполняются и сохраняются в TimesTen
- Read-only кэширование
  - Транзакции выполняются в Oracle Database
- Такая же архитектура, как у TimesTen Classic
  - Поддерживает кэш таблицы и нативные TimesTen таблицы
- HA и отказоустойчивость на уровне приложений

TimesTen Scaleout

Создана на основе проверенной технологии TimesTen

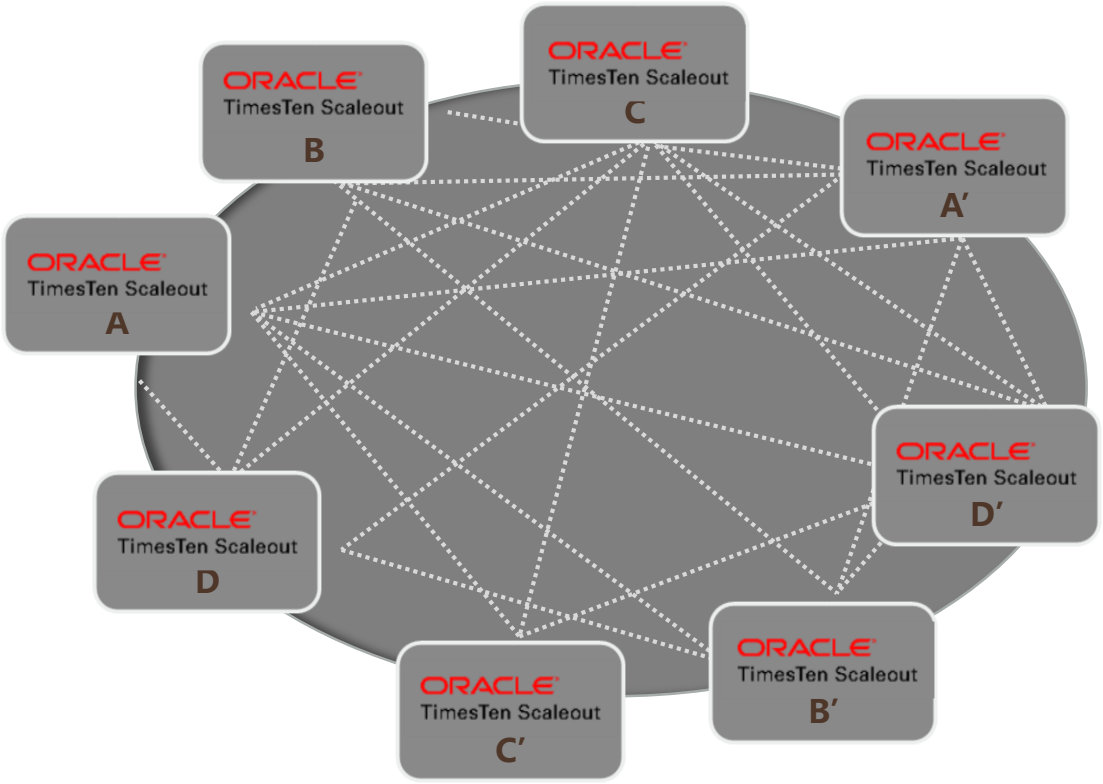


- Для высокоскоростных **OLTP** приложений с **Экстремальной** нагрузкой
  - IOT, биржевая торговля, обнаружение мошенничества, отслеживание местоположения, кликстрим, биллинг, заказы, …
- Передовая архитектура:
  - Только In-Memory, Полный SQL, Полная поддержка ACID транзакций
    - Горизонтально масштабируемая архитектура без разделения ресурсов
    - Несколько копий данных для обеспечения HA (K-safety)
      - Все копии активны для read/writes
    - Глобальные вторичные индексы
  - Комплексные SQL и Parallel SQL для отчётов и пакетной обработки
- Централизованное управление и администрирования

Распределённая, Shared Nothing, In-Memory БД

Единый образ базы данных, высокая доступность и эластичность

- Приложения видят её как одну базу данных
  - НЕ как набор шардов
- Горизонтальное масштабирование вверх и вниз
  - -Данные автоматически перераспределяются
  - -Нагрузка автоматически использует новые элементы
- Встроенная HA с помощью нескольких полностью активных копий
  - Копии автоматически синхронизируются
- -Высокая совместимость с Oracle Database
  - -Типы данных, APIs, SQL & PL/SQL



Структура Грида

Каждый грид содержит:

1 или 2 management instances

Заданное число data instances

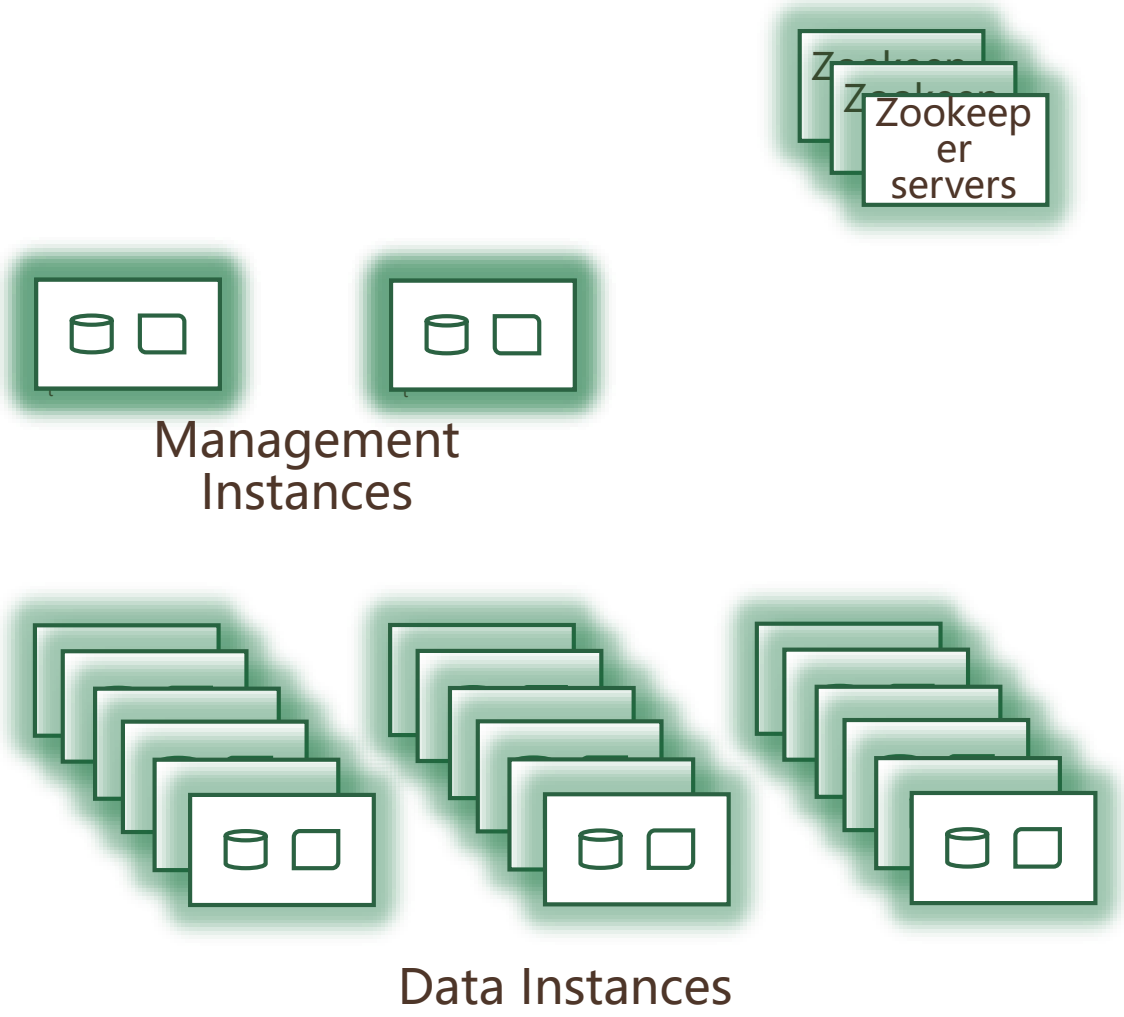
Каждый грид также использует:

- Набор membership серверов, на которых запущен ZooKeeper (обычно 1, 3 или 5)

Узел = instance TT = элемент <= 64

В элементе полный словарь + шард

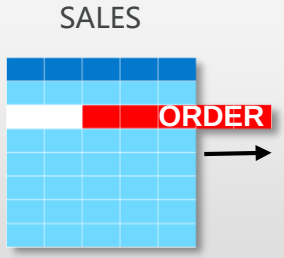
Хэш алгоритм, справочники, family



Оптимизация производительности запросов и транзакций (In-memory опция Oracle DB)

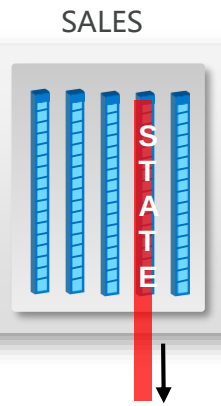
Строчный формат против поколоночного

Row



- **Транзакции работают быстрее со строчным форматом**
  - **Вставка или запрос sales order**
  - **Быстрая обработка нескольких строк, много колонок**

Column



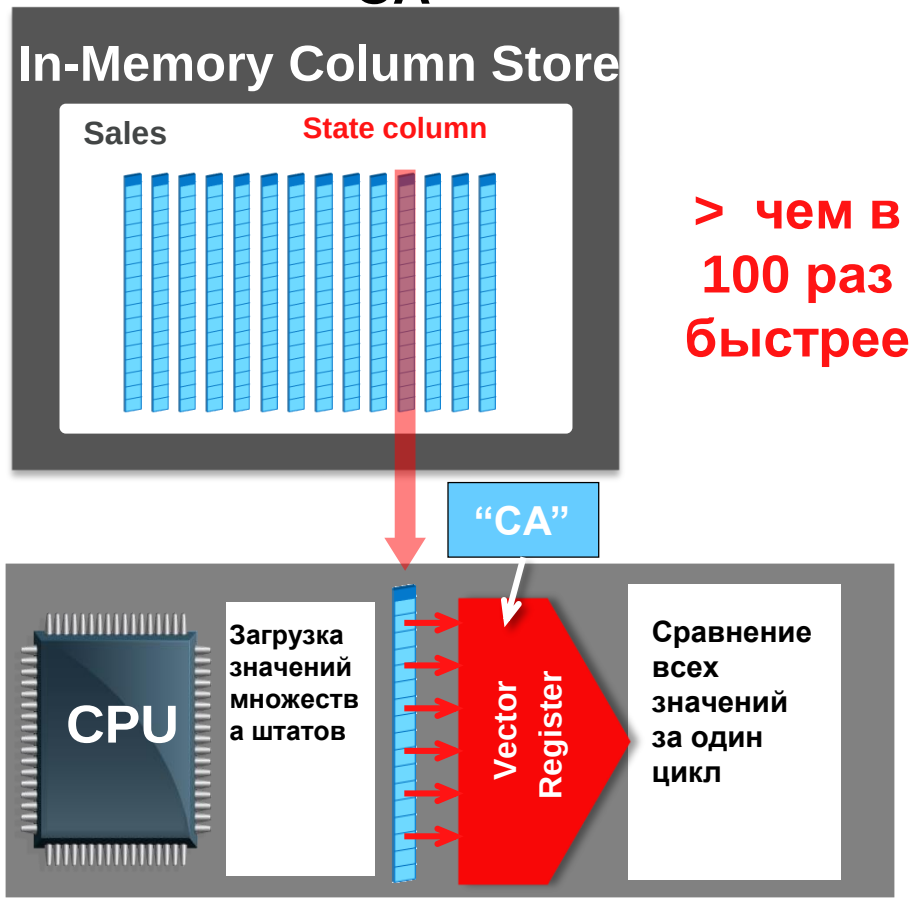
- **Аналитика работает быстрее с колоночным форматом**
  - **Отчет о сумме продаж по штату**
  - **Быстрая обработка нескольких колонок, много строк**

Oracle DB: Хранит данные в обоих форматах одновременно

Сканирование миллиарда строк в секунду на процессорном ядре

- Каждое процессорное ядро сканирует одну колонку
- При сканировании используются быстрые SIMD векторные инструкции
- Миллиарды строк в сек сканируются одним ядром

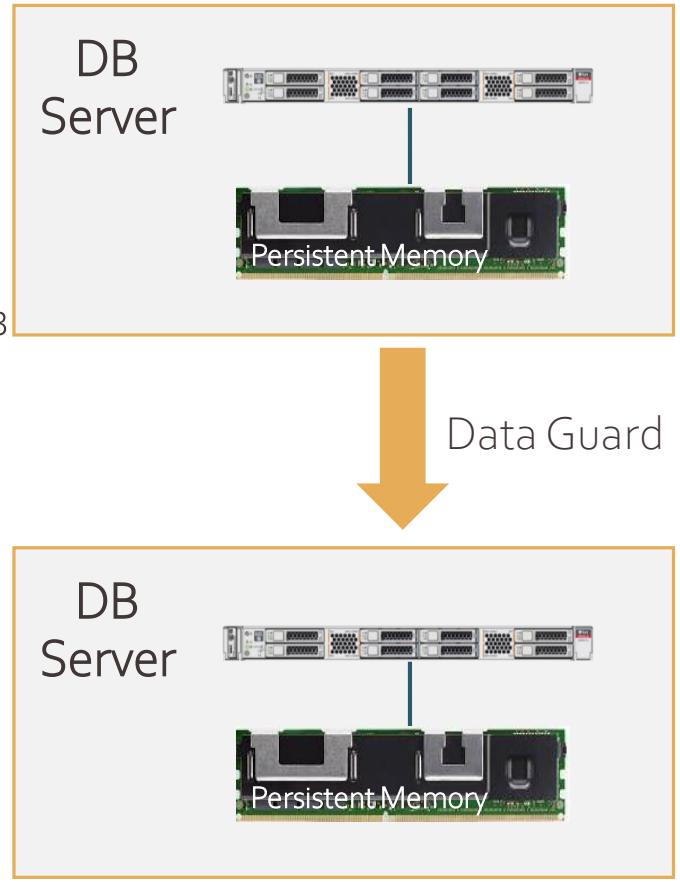
Пример: Найти все продажи в штате CA



> чем в 100 раз быстрее

Поддержка Persistent Memory !!!!

- Хранение данных и redo в локальной постоянной памяти (Persistent Memory - PMEM)
  - Для видов нагрузок, которым необходимы времена отклика меньше, чем может дать обычный флэш-диск
- SQL-запросы выполняются напрямую над данными в “файловой системе” в постоянной памяти
  - Не нужно управлять буферным кэшем – нет накладных расходов
  - Новый алгоритм в БД для предотвращения несогласованности данных в постоянной памяти
- Необходим Data Guard для защиты от отказа сервера или сбоя постоянной памяти



Прямая обработка без буферного кэша

Блоки БД на PMEM мэпятся в SGA

- Блоки БД в PMEM обрабатываются на месте а не читаются в SGA
- Исключаются операции I/O и memcpy() данных

Повышение производительности

- Разогрева кэша нет
- Доступ быстрее (PMEM доступ, нет I/O)

SGA мэпит PMEM

- ~384GB DRAM достаточно для отображения 6TB PMEM

Автоматическая миграция блоков в DRAM

DRAM в ~3X раза быстрее PMEM

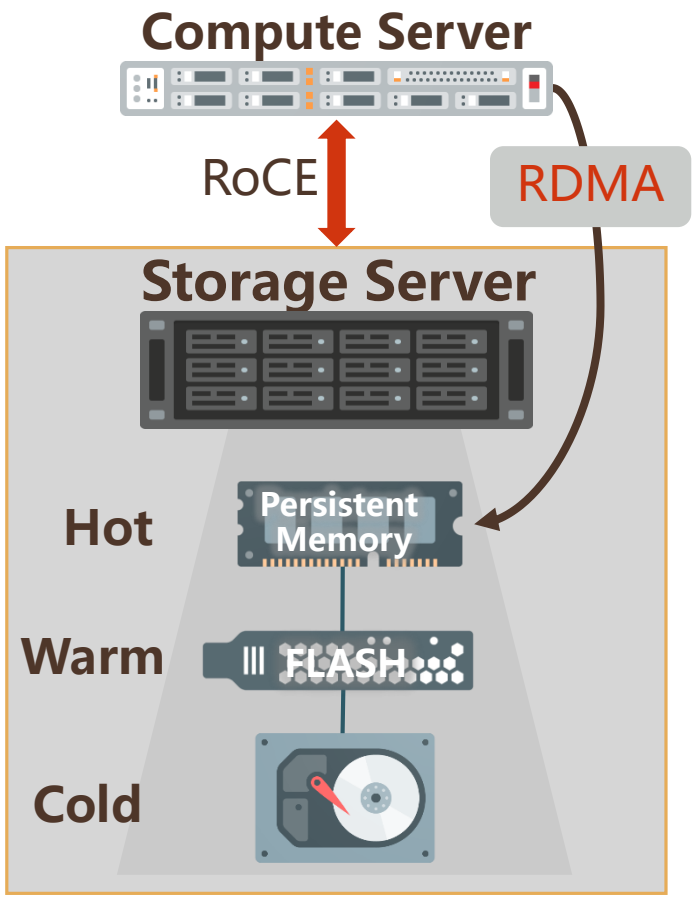
- Блоки обычно обрабатываются напрямую в PMEM
- Часто используемые блоки автоматически перемещаются в DRAM
- Настройка не нужна

Изменяемые блоки перемещаются в DRAM

- Блоки перемещаются в SGA для изменения
- INSERT, UPDATE, DELETE, и т д вызывают перемещение в DRAM
- Блоки пишутся в БД полностью с помощью DBWR
- PMEM Filestore обеспечивает СОГЛАСОВАННОСТЬ блока БД

Exadata X8M с Persistent Memory

Первая в мире и единственная оптимизированная для СУБД постоянная память (Persistent Memory)



19 µsec IO latency

- Exadata Storage Servers прозрачно добавляют Persistent Memory Accelerator перед флэш памятью
- СУБД использует RDMA для доступа к persistent memory
  - Исключение сетевого и IO стека ПО снижает задержку в десятки раз
- Persistent Memory автоматически используется разными БД
  - Только для горячих данных – увеличивает емкость в десятки раз
- Persistent Memory зеркалируется между storage servers для HA
- СУБД пишет журналы в persistent memory используя RDMA для очень быстрой фиксации транзакций

10 тенденций развития СУБД

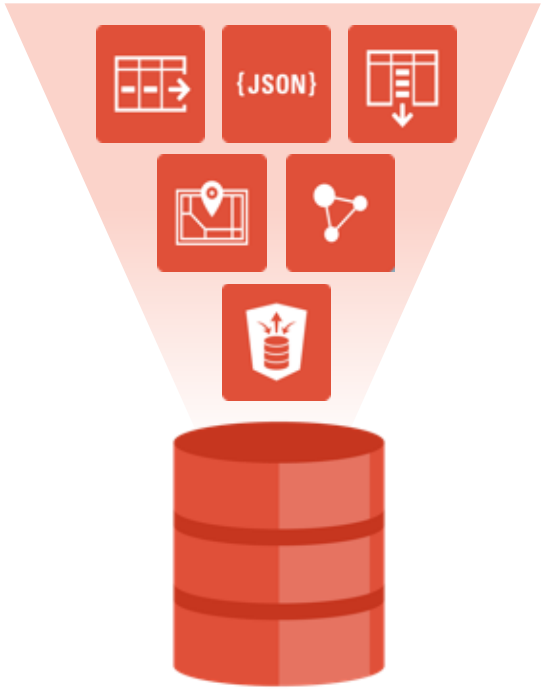
- **Конвергентные СУБД (поддержка одной СУБД множества типов данных - реляционные, гео, текст, JSON, NO SQL, XML, Hadoop, Spark, аудио, видео, изображения и т д) и множества разных нагрузок (OLTP, DSS, IoT, DW, Blockchain, key-value и т д)**
- **Создание мощных коммерческих СУБД, умеющих одинаково работать в разных средах (ЦОД заказчика, частное облако, публичное облако, гибридное облако, машина баз данных, кусок публичного облака в ЦОД заказчика – Cloud&Customer), т е БД как СУБД и как сервис**
- **Симбиоз БД и машин баз данных**
- **Работа с энергонезависимой памятью (Persistent memory)**
- **In-memory вычисления, использование векторных команд процессоров**
- **Самоуправляемые, автономные БД, встраивание алгоритмов искусственного интеллекта в СУБД, облегчение управления множеством СУБД**
- **Шардинг (параллельное хранение и обработка частей таблиц/групп таблиц на различных компьютерах)**
- **Встраивание новых технологий в СУБД (Blockchain, Machine Learning, IoT, JSON и т д) и языков**
- **Продолжение увеличения надежности, производительности, безопасности, масштабируемости и управляемости СУБД, Cloud, BigData …**

Специализированная vs. Многоцелевая

Вместо

**Телефон,
Пейджер,
Фотоаппарат,
Календарь,
Музыка,
Навигатор
Записная книжка
Калькулятор
Фонарь
Часы
…**

Смартфон



Вместо

Relational, No-SQL, JSON, XML, OLTP, Analytics, DW, In-Memory, Key-Value, IoT, ML, Blockchain, Spatial, Sharding…

Конвергентная СУБД

Самоуправляемые, автономные БД,

- встраивание алгоритмов искусственного интеллекта в СУБД,

облегчение управления множеством СУБД

Как в автомобиле без водителя

- Мы определяем политики и уровни сервиса, которые нужно обеспечить - определяем цели
- Система обеспечивает достижение целей
- Но руль пока останется

Традиционно DBA отвечает за:

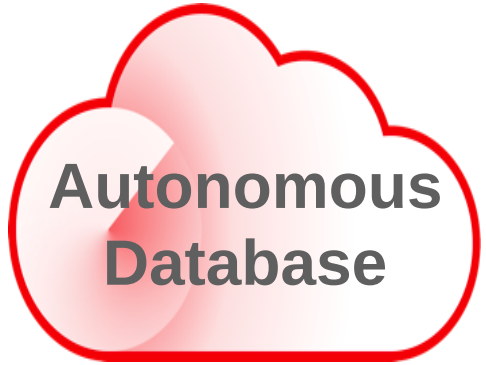
Основные задачи

- Конфигурация операционной системы, сети, подсистемы хранения данных
- Установка Oracle Database, установка новых версий, установка патчей
- Резервное копирование, восстановление
- Развертывание новых сред (для тестирования, разработки и т д)
- Построение отказоустойчивого решения
- Защита данных, устранение дыр в безопасности
- Оптимизация работы БД, настройка SQL
- Управление жизненным циклом данных
- Выявление багов и их устранение
- Выделение дополнительных вычислительных ресурсов (эластичность)

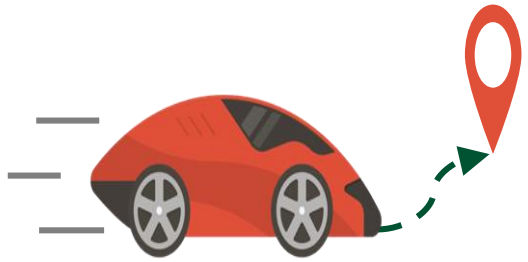
Автономная БД выполняет сама рутинные задачи

Основные задачи

- ~~Конфигурация операционной системы, сети, подсистемы хранения данных~~
- ~~Установка Oracle Database, установка новых версий, установка патчей~~
- ~~Резервное копирование, восстановление~~
- ~~Развертывание новых сред (для тестирования, разработки и т д)~~
- ~~Построение отказоустойчивого решения~~
- ~~Защита данных, устранение дыр в безопасности~~
- ~~Оптимизация работы БД, настройка SQL~~
- ~~Управление жизненным циклом данных~~
- ~~Выявление багов и их устранение~~
- ~~Выделение дополнительных вычислительных ресурсов (эластичность)~~



Автономные БД Oracle



Self-Driving

Автоматизирует все процессы по созданию, управлению, монито-рингу и настройке БД и IT инфраструктуры

Self-Securing

Защищает от внешних атак и несанкциониро-ванного вмешательства внутренних пользователей

Self-Repairing

Защищает от простоев, включая плановые остановки

Дешевле, меньше риски, больше инноваций

Архитектура СУБД Oracle Database

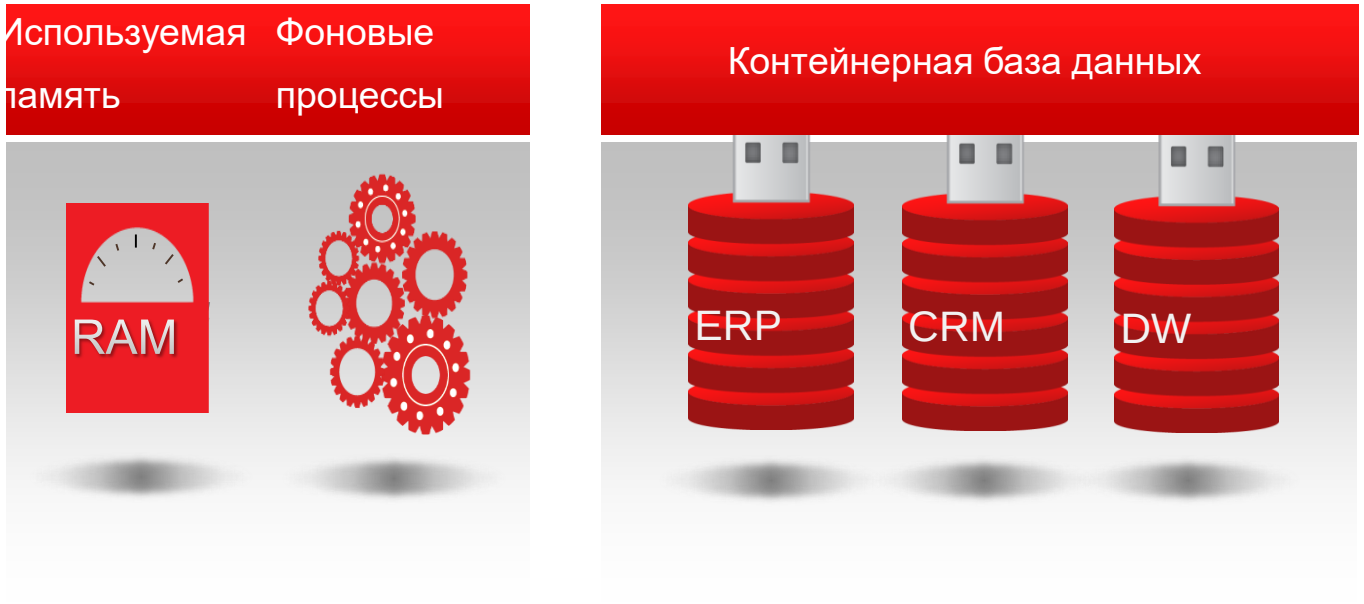
Для каждой БД требуется отдельная память и фоновые процессы

Новая архитектура СУБД

Память и процессы общие для всех БД в контейнере

Новая архитектура СУБД

Память и процессы общие для всех БД в контейнере



Многоарендная СУБД

Multitenancy

Больше БД на одном компьютере (консолидация)

Проще управление

Единый Backup/Restore

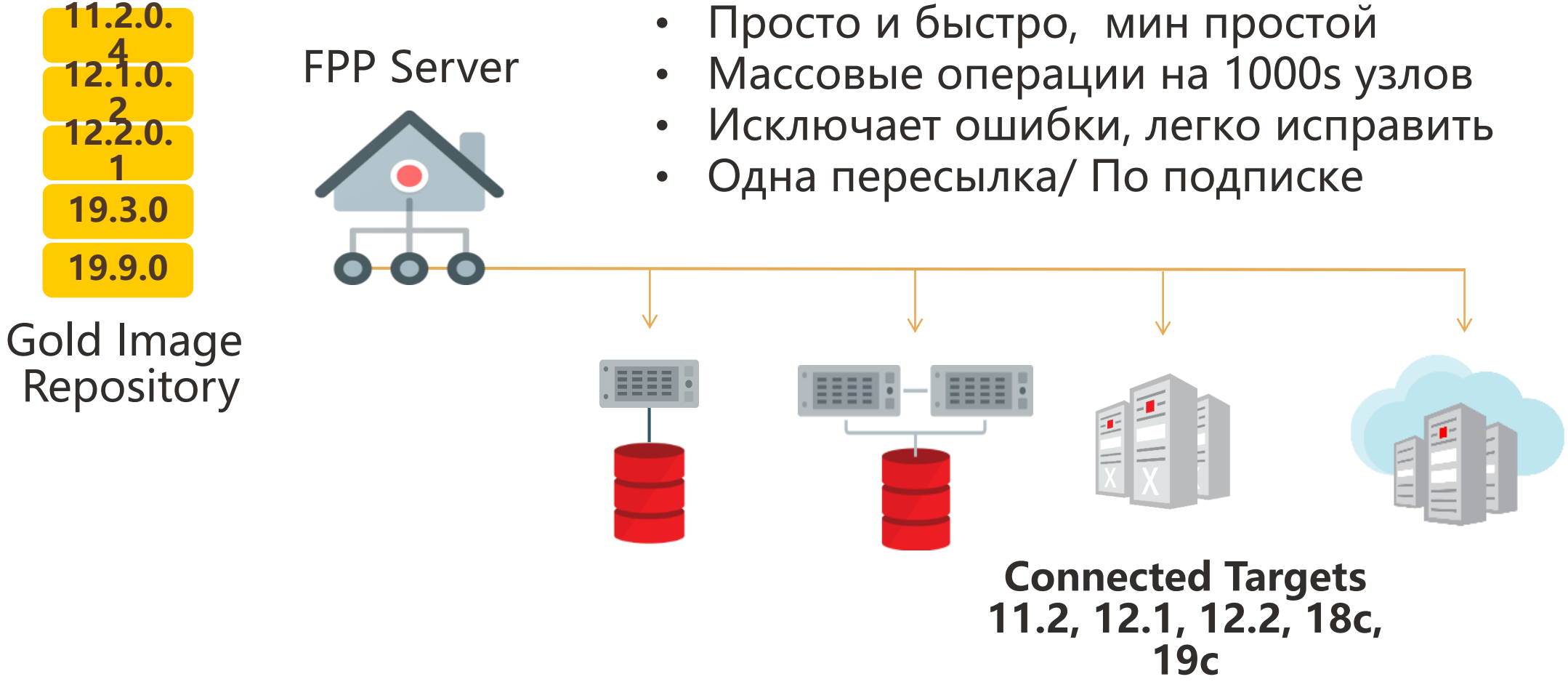
Единый Standby

Единый upgrade

Клонирование, добавление, перенос, вынос, удаление PDB

Каждая PBD – изолированная БД

Fleet management & provisioning



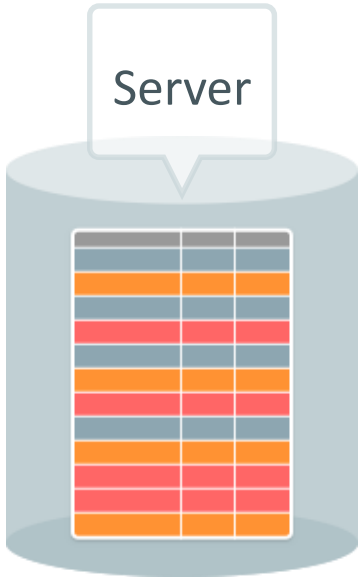
- Просто и быстро, мин простой
- Массовые операции на 1000s узлов
- Исключает ошибки, легко исправить
- Одна пересылка/ По подписке

Шардинг

Oracle Sharding - Архитектура

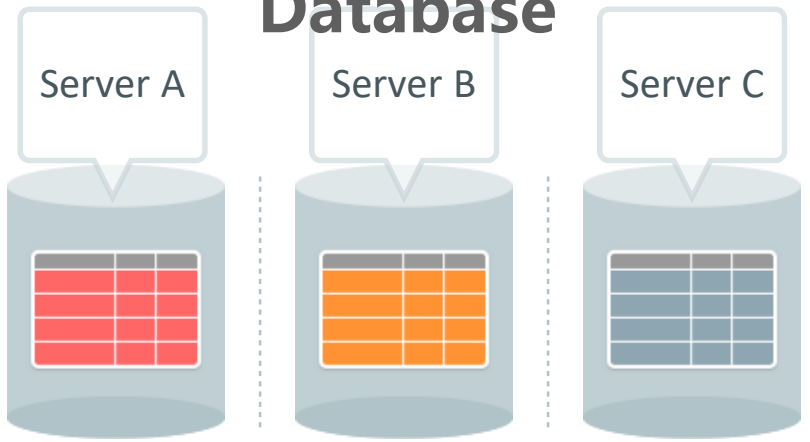
NEW IN 12.2

A Non-sharded Database



Одна единая физическая БД

A Sharded Database



Таблицы разделяются на секции распределенные по трем независимым базам данных (shards)

- Каждый шард имеет собственные CPU, память и диски
- Данные распределяются по базам с помощью ключа – (sharding key), напр: столбец account_id
- Приложение “видит” одну логическую БД

Схема БД с шардингом данных

NEW IN 12.2

Таблицы БД в схеме

Customers

Customer	Name
123	Mary
456	John
999	Peter

Orders

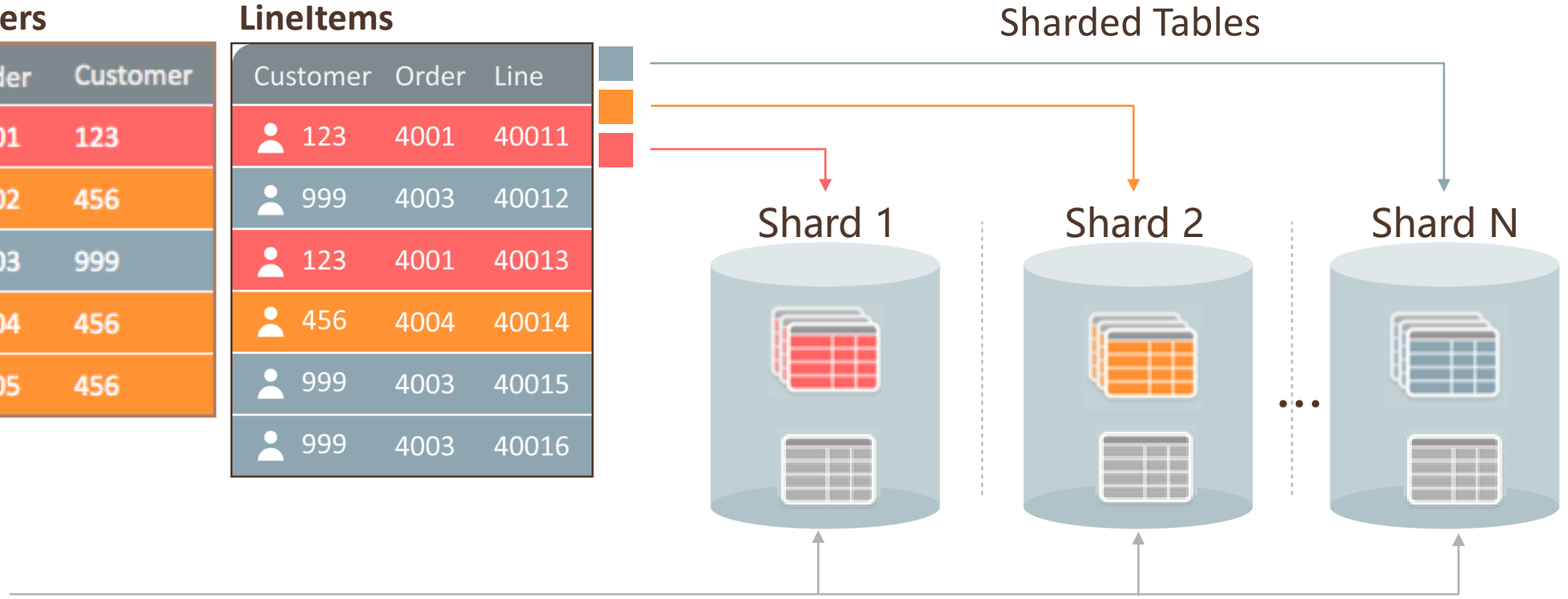
Order	Customer
4001	123
4002	456
4003	999
4004	456
4005	456

LineItems

Customer	Order	Line
123	4001	40011
999	4003	40012
123	4001	40013
456	4004	40014
999	4003	40015
999	4003	40016

Products

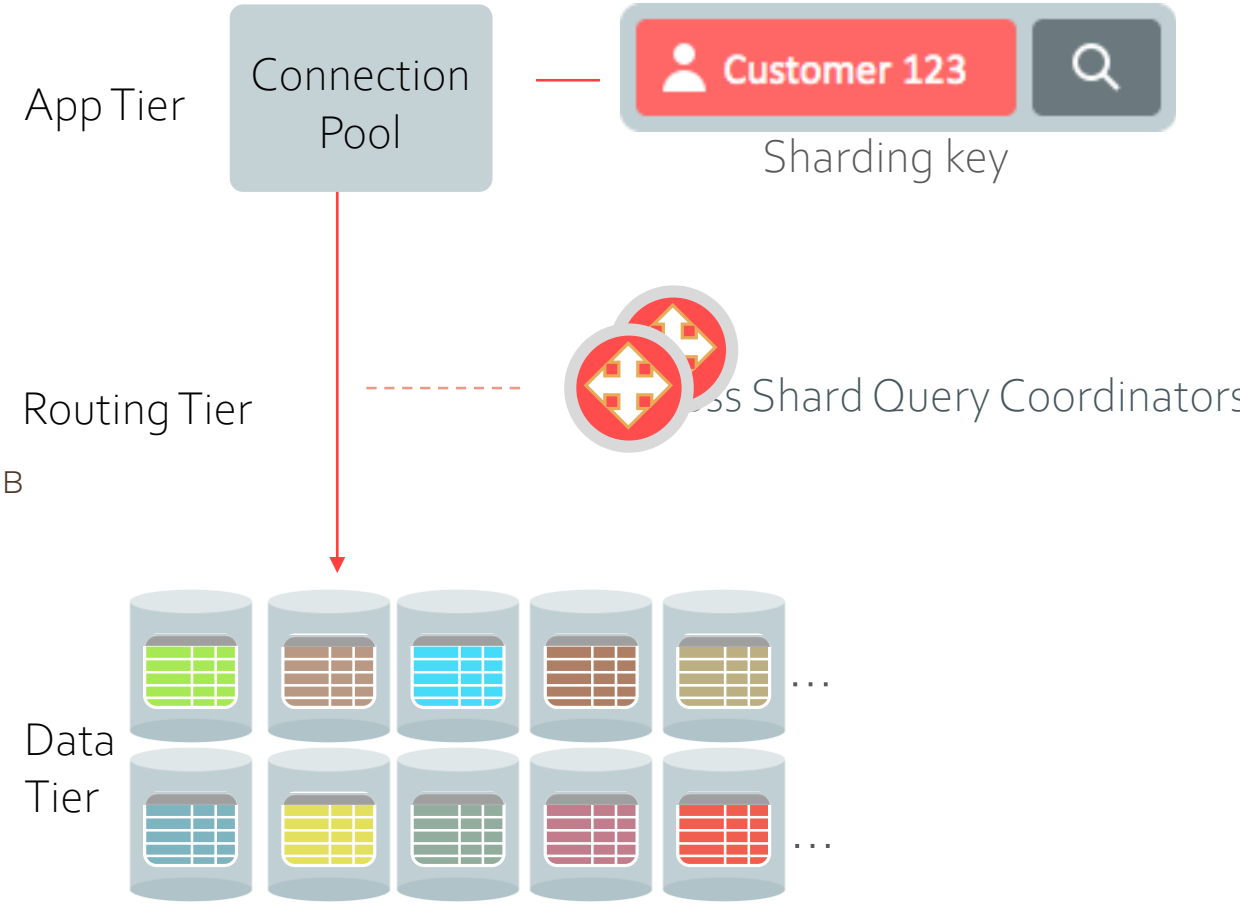
SKU	Product
100	Coil
101	Piston
102	Belt



ORACLE

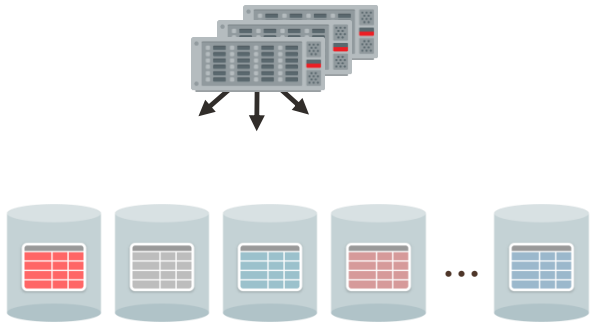
Архитектура

- Клиент (JDBC, OCI, UCP, ODP.NET)
  - Прямая маршрутизация из Connection pools
  - Прокси маршрутизация для многошардовых запросов
- Shard Catalog
  - Хранит метаданные
  - Работает как координатор для многошардовых запросов
  - Хранит инфо о схеме
- Shard Director
  - Глобальный сервис менеджер для прямой маршрутизации коннекта к шардам



Oracle Database Sharding - преимущества

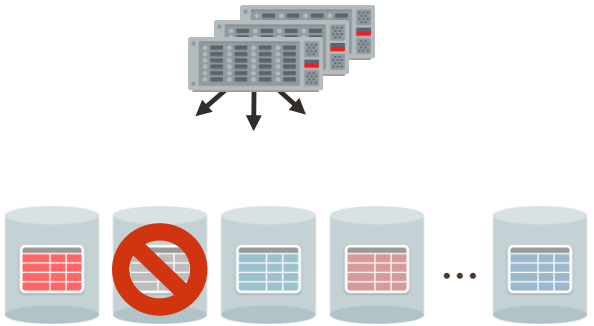
Линейная масштабируемость



Добавление новой БД "на лету" увеличивает пропускную способность приложения.

Балансировка и разбиение шардов “на лету”.

Отказоустойчивость



Сбой одного шарда не приводит к отказу “БД”. Шарды могут запускаться на разных релизах СУБД

Географическое распределение

Определяемое пользователем размещение данных для производительности, доступности, DR.

Встраивание новых технологий и языков

- Блокчейн таблицы
- Memory optimized for read таблицы (key-value)
- Memory optimized for writr таблицы (IoT)
- Auto ML
- Graal VM (Полиглот)

Q & A