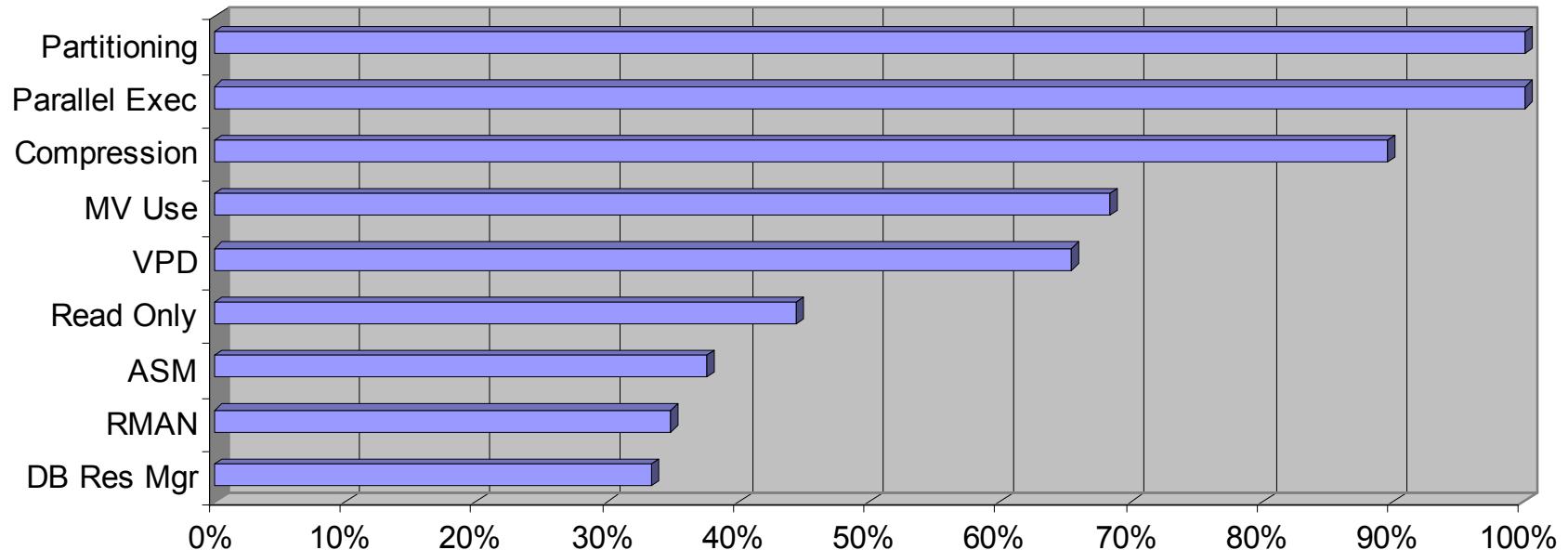


# **Новые возможности базы данных для работы с VLDB**

**ORACLE**

Copyright © 2008, Oracle. All rights reserved.

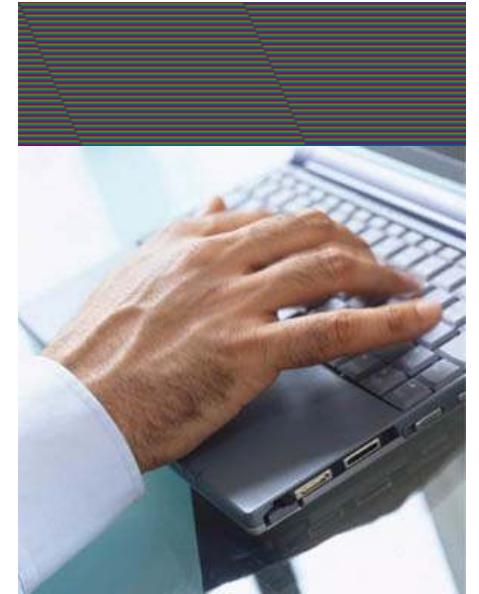
# Feature Usage for Large-Scale Data Warehouses



Partitioning, parallelism, and compression are the foundation for large-scale data warehousing

# Содержание

- Секционирование (partitioning)
- Новые возможности секционирования в 11g
- Advanced Compression
- Parallel
- Q&A



# **Секционирование**

**ORACLE**

Copyright © 2008, Oracle. All rights reserved.

# Oracle Partitioning

	Core functionality	Performance	Manageability
Oracle8	Range partitioning Global range indexes	Static partition pruning	Basic maintenance operations: add, drop, exchange
Oracle8 <i>i</i>	Hash and composite range-hash partitioning	Partitionwise joins Dynamic pruning	Merge operation
Oracle9 <i>i</i>	List partitioning		Global index maintenance
Oracle9 <i>i</i> R2	Composite range-list partitioning	Fast partition split	
Oracle10g	Global hash indexes		Local Index maintenance
Oracle10g R2	1M partitions per table	Multidimensional pruning	Fast drop table
Oracle Database 11g	More composite choices REF Partitioning Virtual Column Partitioning		Interval Partitioning Partition Advisor

# Что такое секционирование (Partitioning)?

## Это

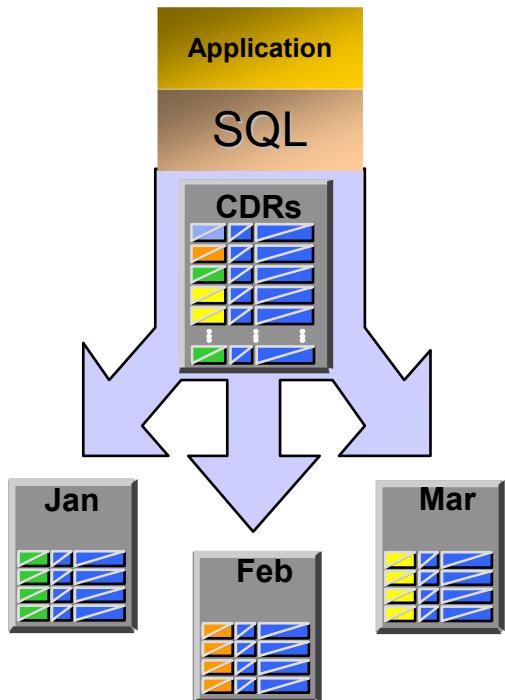
- функционал, позволяющий **логически** разбить объект на более мелкие части
- Выполняется, исходя из требований бизнеса
- Секционирование позволяет повысить производительность, управляемость, доступность

## Это не

- Только способ физически разделить большой объем данных на более мелкие управляемые части

ORACLE

# Секционирование - философия



**С точки зрения приложения:**

Одна большая таблица

Прозрачна для приложения  
оптимизация

Производительность

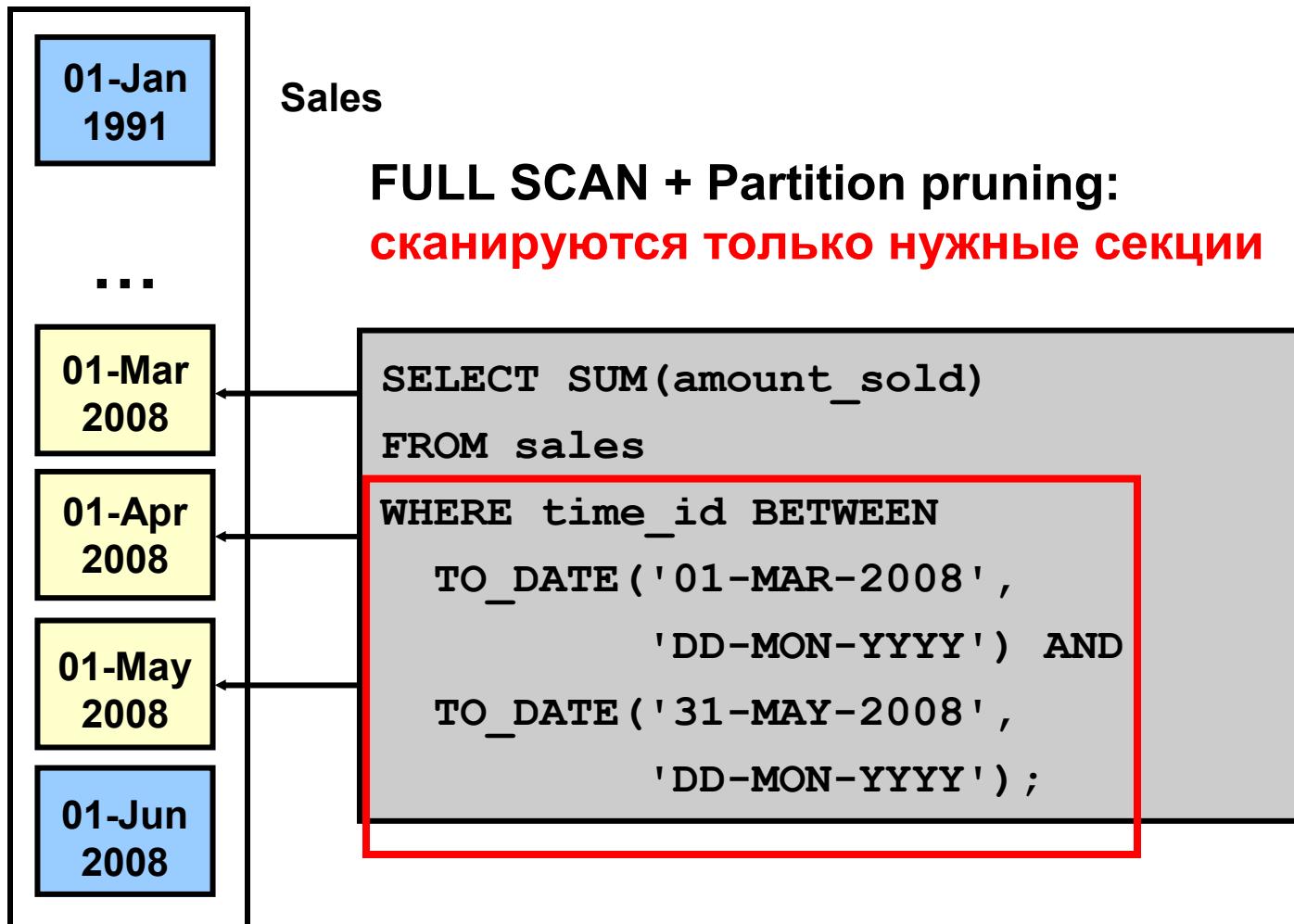
**С точки зрения DBA:**

Маленькие секции

Управление секциями по  
отдельности или несколькими за раз

Управляемость/Доступность

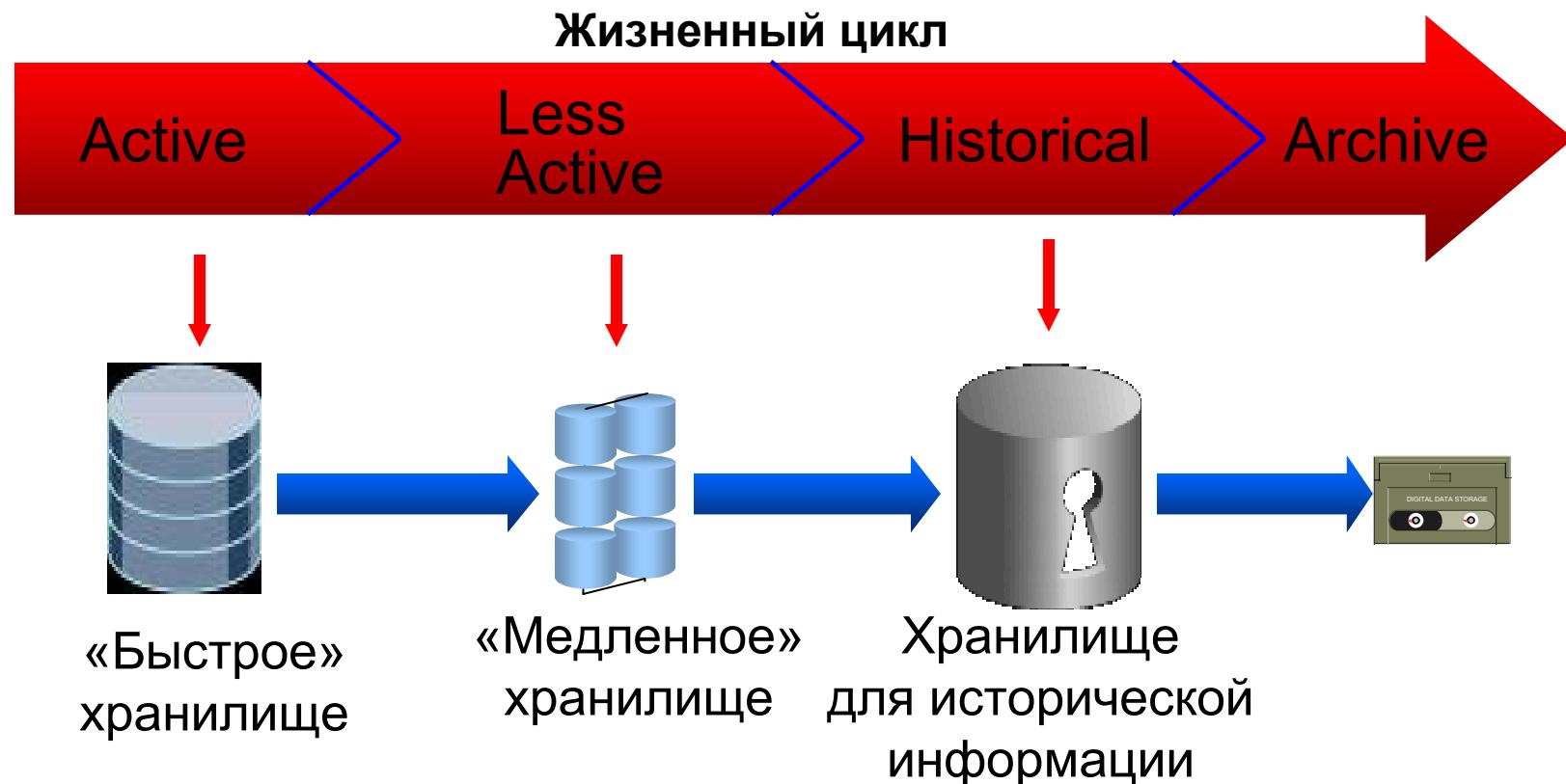
# Повышение производительности: Partition Pruning



# Жизненный цикл данных



# Кризис: оптимизируем хранение данных



# Управляемость: управление на уровне секций

- Секция может быть:
  - перемещена между табличными пространствами
  - разделена
  - переведена в оффлайн
  - перестроен индекс
  - быстро удалена из таблицы
- SELECT, UPDATE, INSERT, и DELETE могут работать на уровне секций вместо уровня таблицы.

# **Новые возможности по секционированию в 11g**

**ORACLE**

Copyright © 2008, Oracle. All rights reserved.

# Creating Partitions with Enterprise Manager

The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. The top navigation bar includes links for Setup, Preferences, Help, Logout, and a Database button. The main title is "Database Instance: orcl.us.oracle.com > Tables > Create Table > Create Partitions: Partitioning Method". The user is logged in as SYS. The page displays five partitioning methods: Range, Hash, List, Range-Hash, and Range-List. The "Range" method is selected. A note states: "Range partitioning maps rows to partitions based on ranges of column values. This type of partitioning is useful when dealing with data that has logical ranges into which it can be distributed; for example, months of the year or financial quarters. Performance is best when the data evenly distributes across the range." The "Hash" method is described as evenly distributing data across partitions based on a hash value. The "List" method allows explicit control over row-to-partition mapping. The "Range-Hash" and "Range-List" methods combine range and list partitioning respectively. At the bottom are "Cancel" and "Continue" buttons, and a footer with links for Database, Setup, Preferences, Help, and Logout.

ORACLE Enterprise Manager 11g Database Control

Database Instance: orcl.us.oracle.com > Tables > Create Table > Create Partitions: Partitioning Method

Logged in As SYS

SYS

Cancel Continue

Select the method to use to partition the table.

Range

Range partitioning maps rows to partitions based on ranges of column values. This type of partitioning is useful when dealing with data that has logical ranges into which it can be distributed; for example, months of the year or financial quarters. Performance is best when the data evenly distributes across the range.

Hash

Hash partitioning provides a method of evenly distributing data across a specified number of partitions. Rows are mapped into partitions based on a hash value of the partitioning key. Creating and using hash partitions gives you a highly tunable method of data placement, because you can influence availability and performance by spreading these evenly sized partitions across I/O devices (striping).

List

List partitioning gives you explicit control over how rows map to partitions. List partitioning is specifically designed for modeling data distributions that follow discrete values. The advantage of list partitioning is that you can group and organize unordered and unrelated sets of data in a natural way.

Range-Hash

Range-Hash composite partitioning partitions data using the range method, and within each partition, subpartitions it using the hash method. Range-Hash partitions provide the improved manageability of range partitioning combined with the even data distribution and parallelism advantages of hash partitioning.

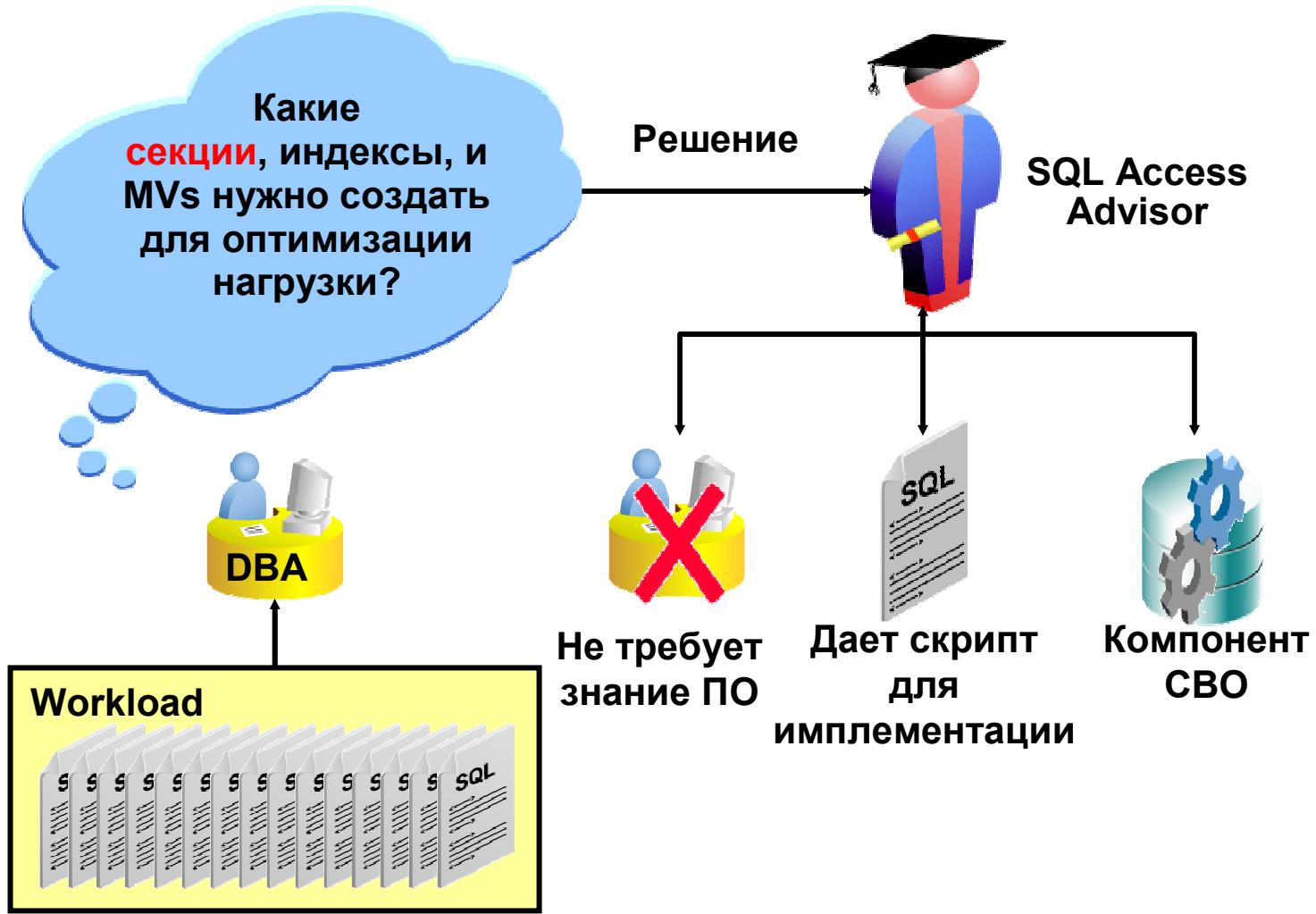
Range-List

Range-List composite partitioning partitions data using the range method, and within each partition, subpartitions it using the list method. Range-List partitions provide the improved manageability of range partitioning combined with the explicit placement control of list partitioning.

Cancel Continue

[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

# SQL Access Advisor: Обзор



# SQL Access Advisor: Рекомендации

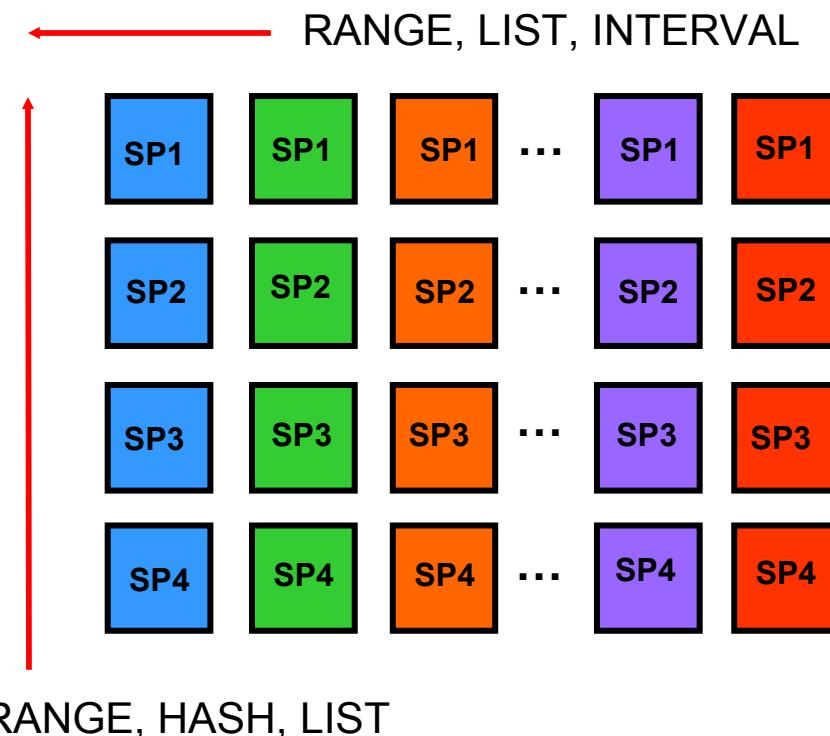
Рекомендация	Исчерпывающ	Огранич
Добавить новый ( <b>секционированный</b> ) индекс на таблицу или materialized view.	Да	Да
Удалить неиспользуемый индекс	Да	Нет
Изменить тип индекса	Да	Нет
Добавить в индекс дополнительные столбцы	Да	Да
Добавить новую ( <b>секционированную</b> ) materialized view.	Да	Да
Удалить неиспользуемую materialized view (журнал).	Да	Нет
Добавить новый materialized view журнал.	Да	Да
Изменить существующий materialized view журнал добавлением столбцов или ограничений	Да	Да
<b>Секционировать существующую несекционированную таблицу или индекс</b>	Да	Да

# Стратегии секционирования: Одноуровневые секции

- **Диапазон** распределяет данные, основываясь на значении, попадающем в диапазон секции
- **Hash** распределяет данные, основываясь на вычислении хэша для ключа секционирования
- **Список** распределяет данные на основе сопоставления дискретного набора значений каждой секции.
- **Интервал** то же, что и диапазон, но можно автоматизировать процедуру создания новых диапазонов.
- **Системная** позволяет приложению явно указать, в какую секцию попадут данные.

# Стратегии секционирования: Сложное секционирование

- Диапазон - верхний уровень:
  - Диапазон-Диапазон
  - Диапазон-Хэш
  - Диапазон-Список
- Список - верхний уровень:
  - Список-Диапазон
  - Список-Хэш
  - Список-Список
- Интервал - верхний уровень:
  - Интервал-Диапазон
  - Интервал-Хэш
  - Интервал-Список



# Новые возможности секционирования в 11g

- **Секционирование «по ссылке» (Reference):**
  - Секционирование «по ссылке» используется в таблицах со связями по внешнему включу
  - Ключ секционирования извлекается через активное ограничение целостности
  - Ключ секционирования может не существовать в дочерней таблице.
- **Секционирование по виртуальному столбцу:**
  - Виртуальный столбец – это выражение, которое использует один или несколько столбцов таблицы
  - Виртуальный столбец существует только как метаданные
  - Вы можете строить ключ секционирования, основываясь на значениях выражения.

# Oracle Database 11g Advanced Compression

ORACLE

Copyright © 2008, Oracle. All rights reserved.

# Вызовы нашего времени



Взрывной рост объема данных

- Правительственные акты (Sarbanes-Oxley, 77-ФЗ, Перечень типовых управлеченческих документов, образующихся в деятельности организаций, с указанием сроков хранения, Примерный перечень документов, образующихся в деятельности кредитных организаций, с указанием сроков хранения... )
- Web 2.0
- Консолидация приложений

Бюджет меньше – данных больше

- Необходимо оптимизировать использование хранилищ
- Приложения должны работать быстро

**Умная и эффективная компрессия может помочь**

# Введение

Oracle Database 11g предлагает исчерпывающий набор возможностей компрессии

- Табличных данных
- Неструктурированных данных
- Резервных копий
- Сетевого трафика

Уменьшает стоимость и требования к ресурсам

- Системы хранения
- Пропускная способность сетевых каналов
- Объем памяти



# Сжатие таблиц

Появилось в Oracle Database 9i Release 2

- Сжатие происходит во время bulk load операций (Direct Load, CTAS)
- Данные модифицируемые обычным DML не сжимаются

Алгоритм компрессии оптимизирован для реляционных данных

Запросы, читающие большой объем

- Совершают меньше операций ввода-вывода
- Имеют больший коэффициент попаданий в буфер данных

Данные компрессируются на уровне блоков

Включается на уровне таблиц **или секций**

Для приложения полностью прозрачно



# OLTP Table Compression

ORACLE<sup>®</sup>  
D A T A B A S E 11g

Oracle Database 11g расширяет поддержку компрессии для данных OLTP

- Поддерживаются обычные DML операции (INSERT, UPDATE)

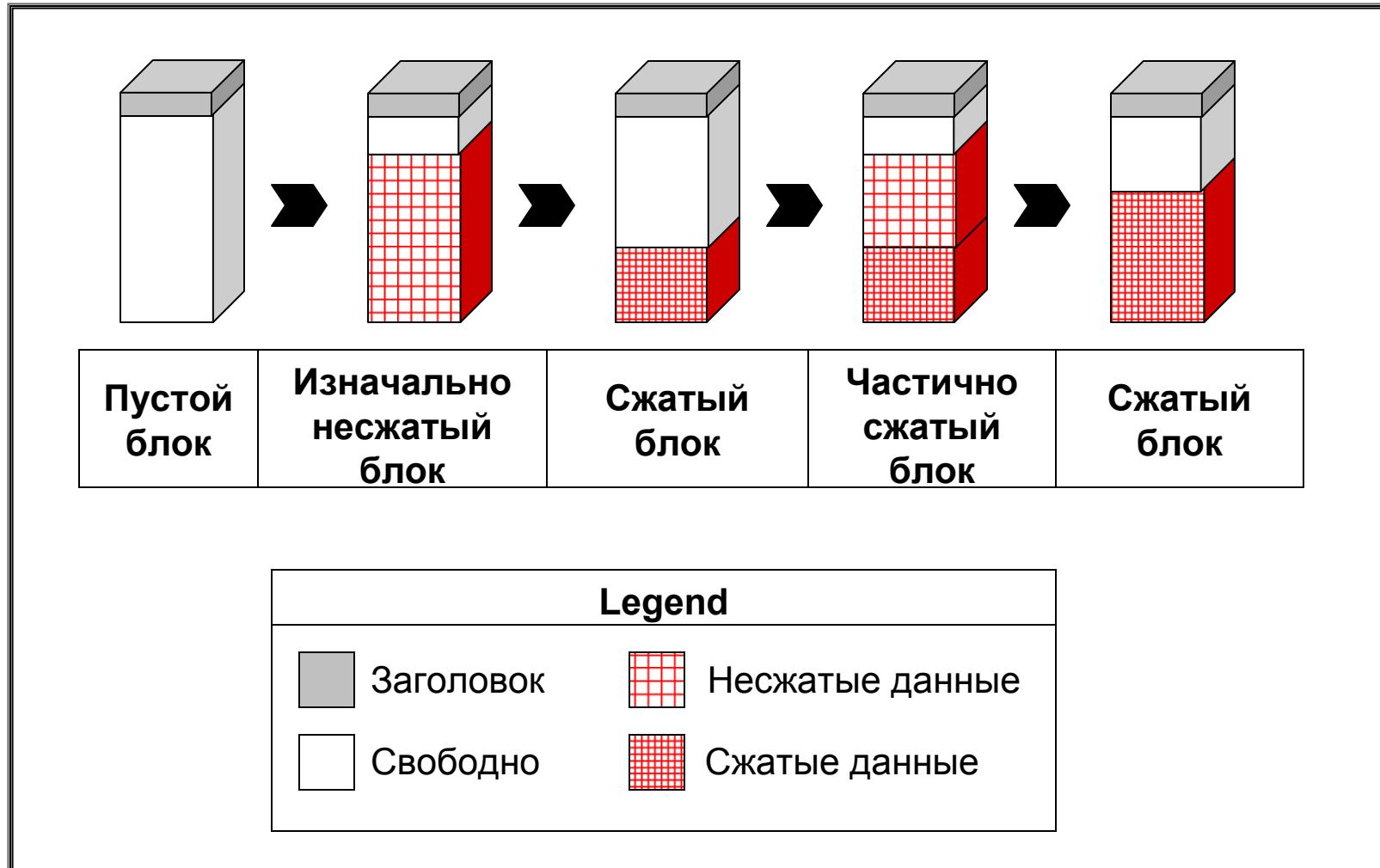
Новый алгоритм значительно снижает доп. нагрузку

- Batched compression ensures no impact for most OLTP transactions

Обычно не влияет на чтение

- Читатель можно даже получить прирост производительности за счет меньшего объема IO и большей эффективности использования памяти

# Процесс сжатия OLTP таблиц



# Сжатие OLTP таблиц

**Employee Table**

ID	FIRST_NAME	LAST_NAME
1	John	Doe
2	Jane	Doe
3	John	Smith
4	Jane	Doe
5	Jack	S



**Сжатый блок**

Заголовок			
John=①	Doe=②	Jane=③	Smith=④
1•①•①	2•②•①	3•③•④	4•②
•①	5•Jack•③		
		Свободно	

## Синтаксис для сжатия

- Сжатие для всех операций

```
CREATE TABLE emp () COMPRESS FOR ALL OPERATIONS;
```

- Сжатие для операций прямой загрузки:

```
CREATE TABLE emp () COMPRESS  
[FOR DIRECT_LOAD OPERATIONS];
```

- Сжатие существующей таблицы:

```
ALTER TABLE emp () MOVE COMPRESS;
```

- Табличное пространство (сжатие по-умолчанию)

```
CREATE TABLESPACE test DATAFILE 'test01.dbf'  
SIZE 100m DEFAULT COMPRESS;
```

# Сжатие OLTP таблиц

## *Рекомендации*

Сжимайте Ваши 10 самых больших таблицы

- Правило 80/20 – 20% таблиц занимают 80% диска

Чем больше блок, тем больше сжатие

- Большая вероятность дублирования данных

Сжатие B-Tree индексов

```
CREATE INDEX ord_customer_ix_demo  
ON orders (customer_id, sales_rep_id) COMPRESS;
```

- Степень сжатия можно увидеть в **INDEX\_STATS**
  - **INDEX\_STATS.OPT\_CMPR\_COUNT**
  - **INDEX\_STATS.OPT\_CMPR\_PCTSAVE**

Bitmap Indexes компрессируются по-умолчанию

## Увеличение степени сжатия

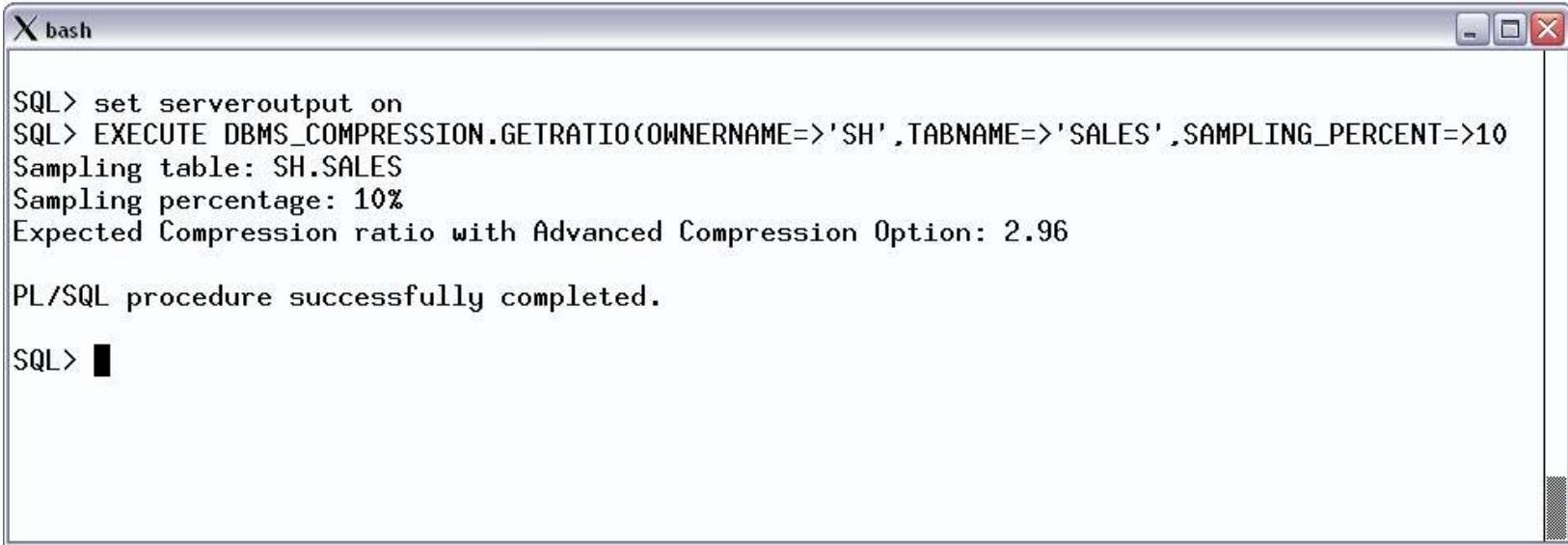
- Сжатые данные могут быть отсортированы повышения степени дублируемости данных в рамках блока
- Если используем внешние таблицы, то можно указать фразу ORDER BY.
- Использовать больший объем блока.

# Сжатие OLTP таблиц

## *Начало работы*

### Compression Advisor (Доступен!)

- Поддержка Oracle Database 9i Release 2 и более поздних
- Позволяет оценить сжатие для несжатых таблиц
- В 11g показывает сжатие для сжатых



The screenshot shows a terminal window titled "bash" running on an Oracle database. The user has run the following PL/SQL command:

```
SQL> set serveroutput on
SQL> EXECUTE DBMS_COMPRESSION.GETRATIO(OWNERNAME=>'SH',TABNAME=>'SALES',SAMPLING_PERCENT=>10
Sampling table: SH.SALES
Sampling percentage: 10%
Expected Compression ratio with Advanced Compression Option: 2.96
PL/SQL procedure successfully completed.

SQL> ■
```

# Сжатие таблиц. Тестирование

Данные по внедрению компрессии в Oracle Applications

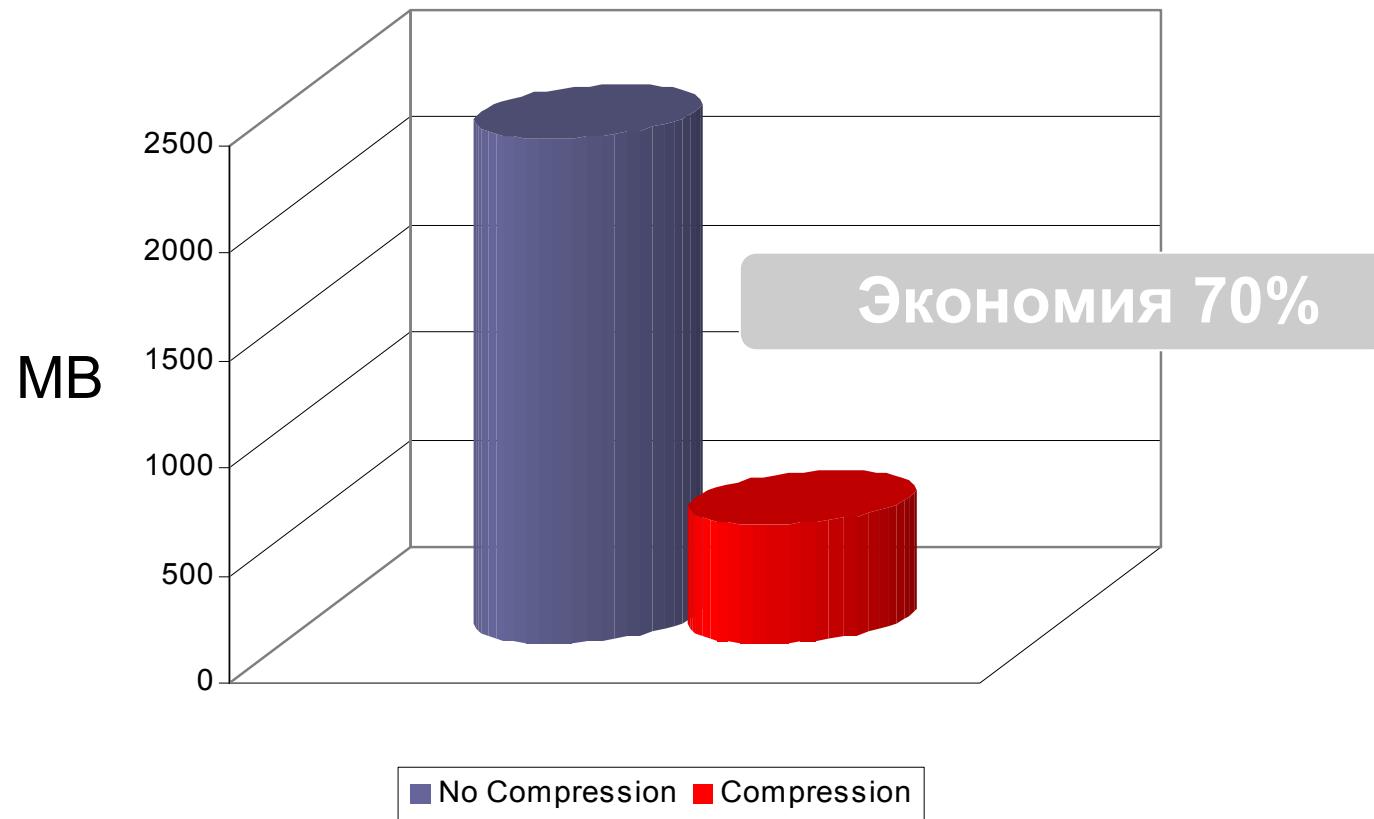
- Сжаты 10 самых больших таблицы
  - 2 копии каждой таблицы: сжатая и несжатая
- Oracle Enterprise Linux
- Oracle Database 11g Release 1

Тесты

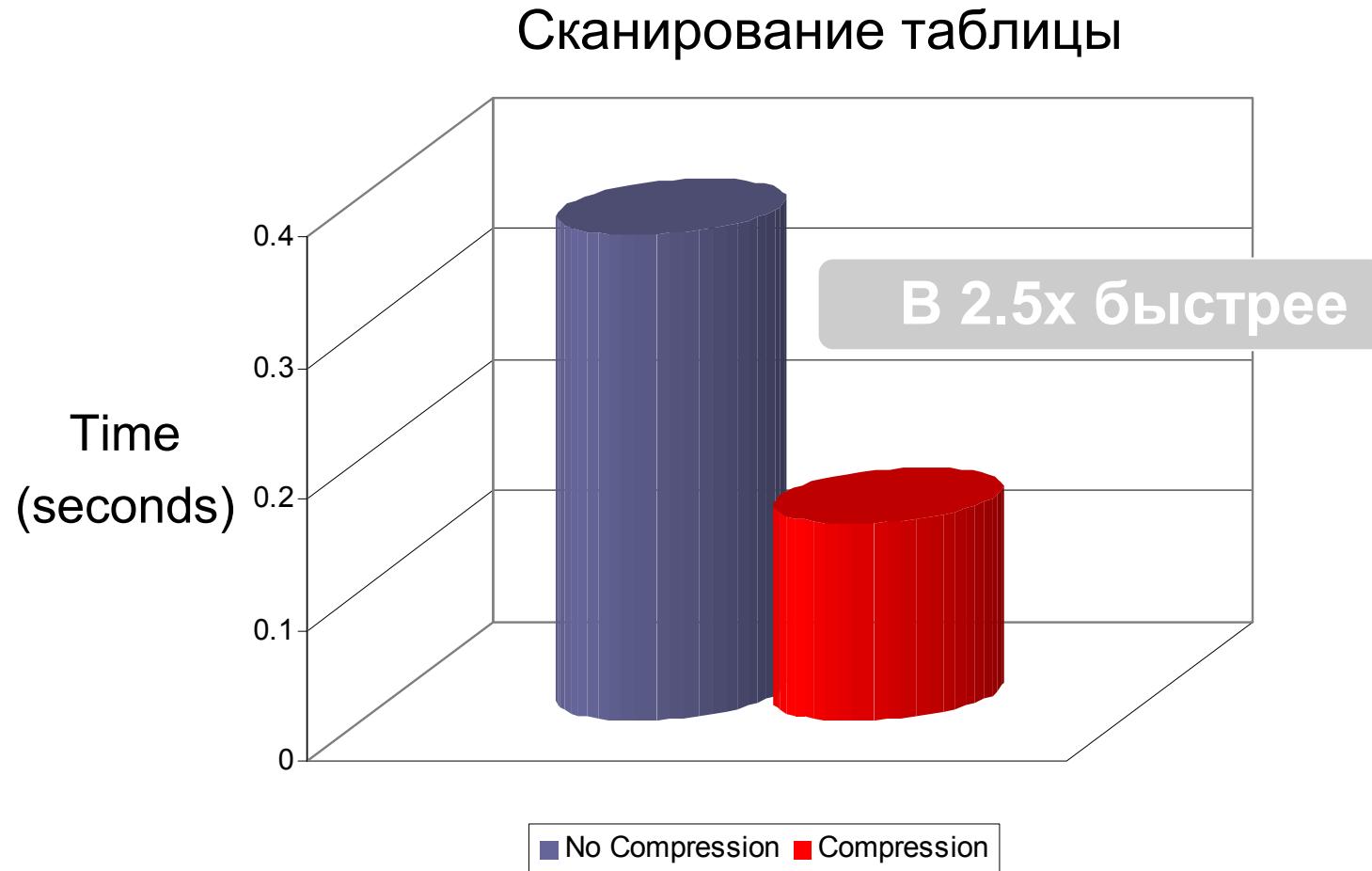
- Full Table Scan
- Index Range Scan
- DML операции (Insert, Update, Delete)

# Сжатие таблиц. Тестирование

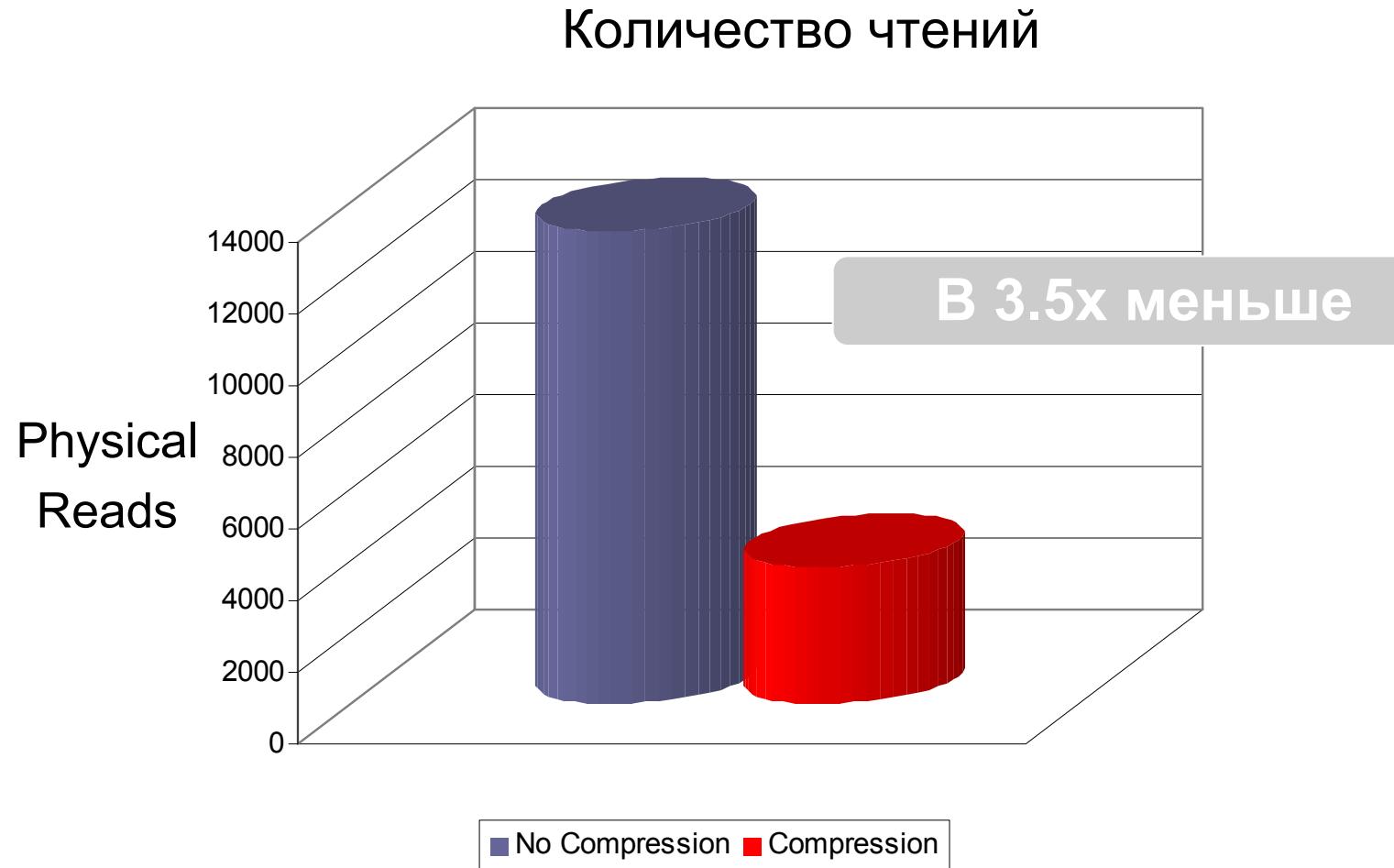
Использование пространства



# Сжатие таблиц. Тестирование

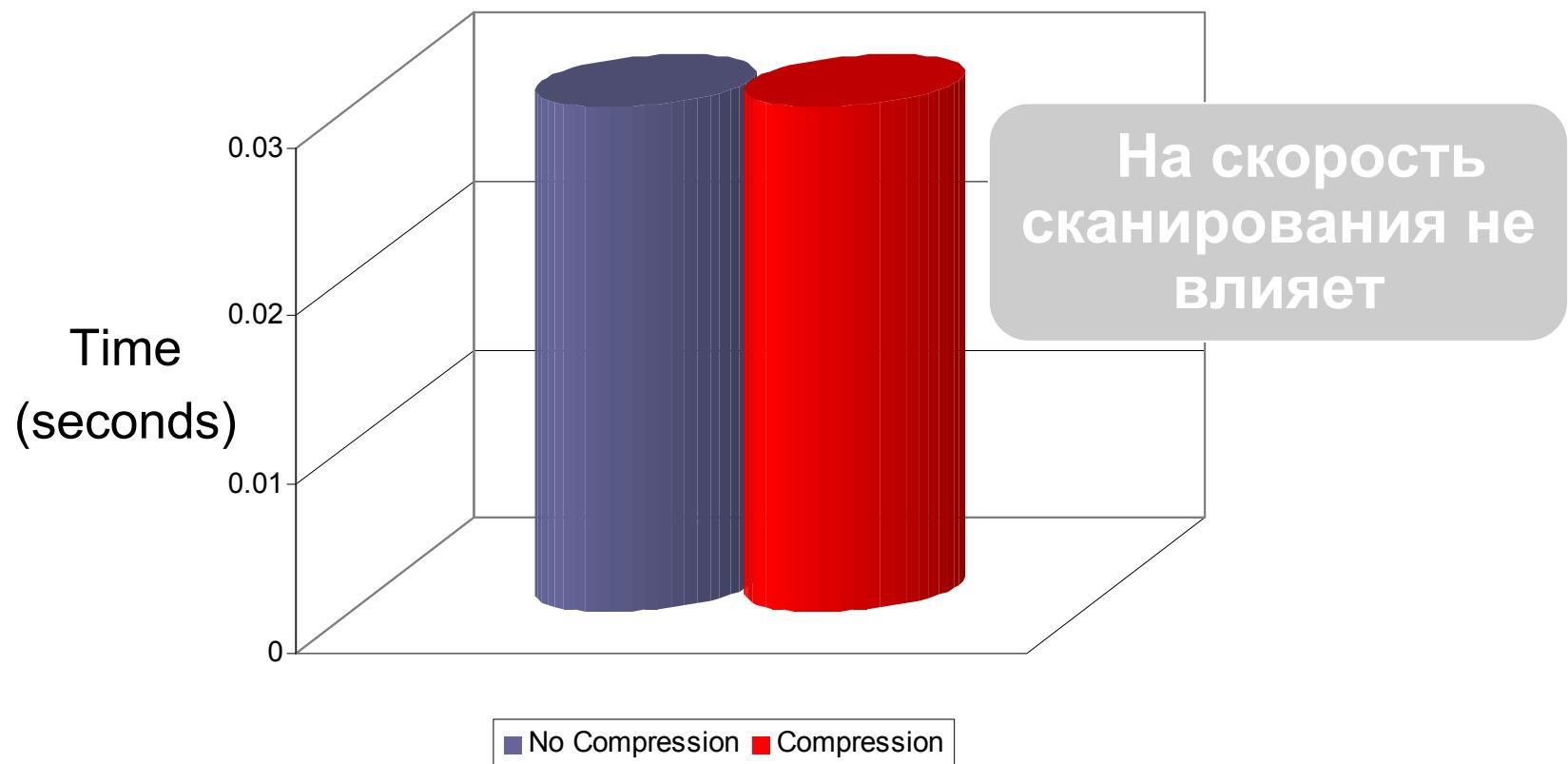


# Сжатие таблиц. Тестирование



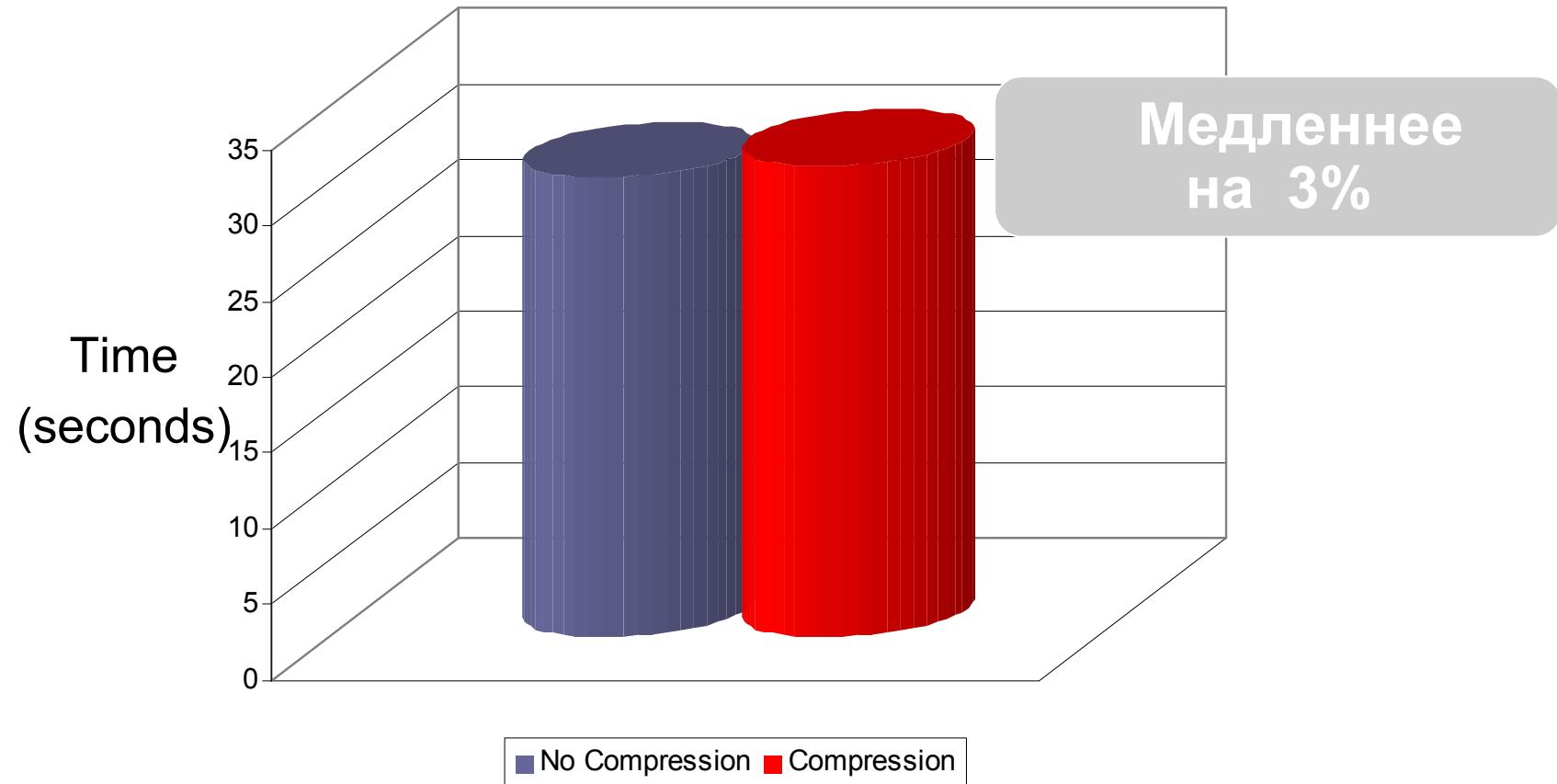
# Сжатие таблиц. Тестирование

Сканирование по индексу



# Сжатие таблиц. Тестирование

## DML операции



# Сжатие Data Pump

Сжатие метаданных доступно с Oracle Database 10g

Oracle Database 11g расширяет сжатие на табличные данные во время экспорта

- Не требуется декомпрессия перед импортом

Компрессия данных и метаданных в один шаг

- Компрессированные данные сохраняются напрямую в файл дампа – требует меньше места на диске
- Внутренние тесты показывают сжатие до 75%

Степень сжатия сравнима с GNU gzip

Прозрачна для приложений

- Вся функциональность Data Pump доступна на сжатых файлах



# Сжатие Data Pump

Стоимость сжатия: около ~10%

Степень сжатия : сравнимо с gzip

Метод сжатия	OE/SH Schemas	Spatial Table	Spatial Schema
<code>expdp compression=none</code>	6.0 MB	26.6 MB	443 MB
<code>expdp compression=all</code>	1.5 MB (74.7%)	<b>9.9 MB (62.7%)</b>	<b>140 MB (68.4%)</b>
<code>gzip -cv1</code>	1.1 MB (82.7%)	11.4 MB (57.1%)	162 MB (63.4%)
<code>gzip -cv6</code>	835 KB (86.2%)	10.2 MB (61.7%)	142 MB (68.0%)
<code>gzip -cv9</code>	<b>818 KB (86.5%)</b>	10.1 MB (62.0%)	141 MB (68.2%)
<code>compress</code>	1.6 MB (74.2%)	13.8 MB (48.1%)	198 MB (55.3%)

# Backup Compression

## Быстрое сжатие RMAN

- Сжимает содержимое backup set перед записью на диск или ленту
- Не требует дополнительных шагов
- Высокая производительность, стандартные алгоритмы
  - 40% быстрее сжатия в Oracle Database 10g
- Подходит для быстрых, инкрементальных резервных копий
- Уменьшает сетевую нагрузку



# Backup Compression Syntax

## FAST RMAN Compression Configuration

```
RMAN> CONFIGURE COMPRESSION ALGORITHM 'zlib';
```

## RMAN Compression Syntax

```
RMAN> backup as COMPRESSED BACKUPSET database  
archivelog all
```

---

## DataPump Syntax

```
PROMPT> expdp hr FULL=y  
DUMPFILE=dpump_dir:full.dmp COMPRESS
```



# Сжатие RMAN. Тестирование

Данные Oracle Applications

- 3.5 GB Database
- Oracle Enterprise Linux
- Oracle Database 11g Release 1

Test 1: Slow I/O (16 MB/s)

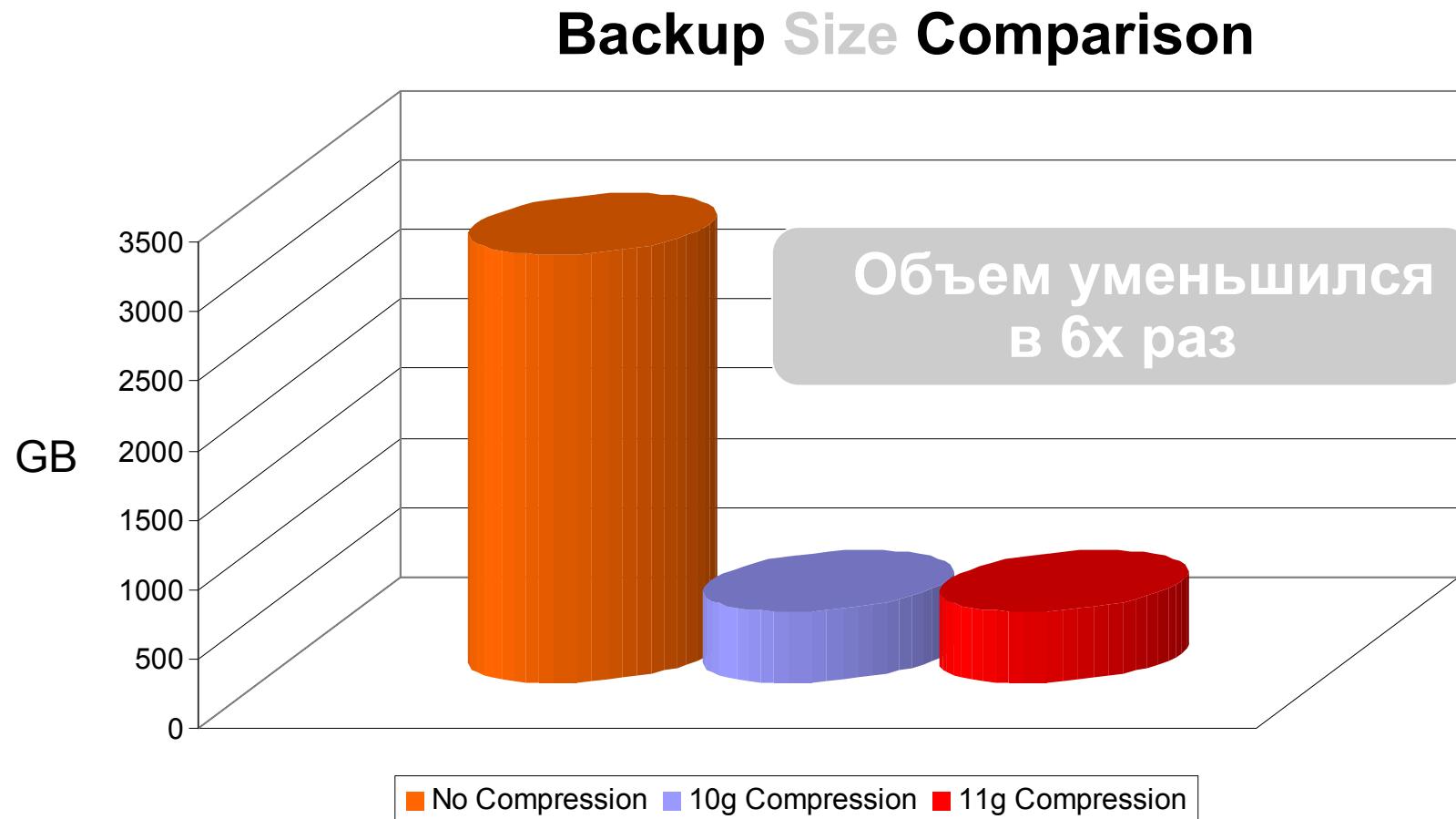
- 11g RMAN без сжатия
- 10g RMAN со сжатием
- 11g RMAN с БЫСТРЫМ сжатием

Test 2: Fast I/O (200 MB/s)

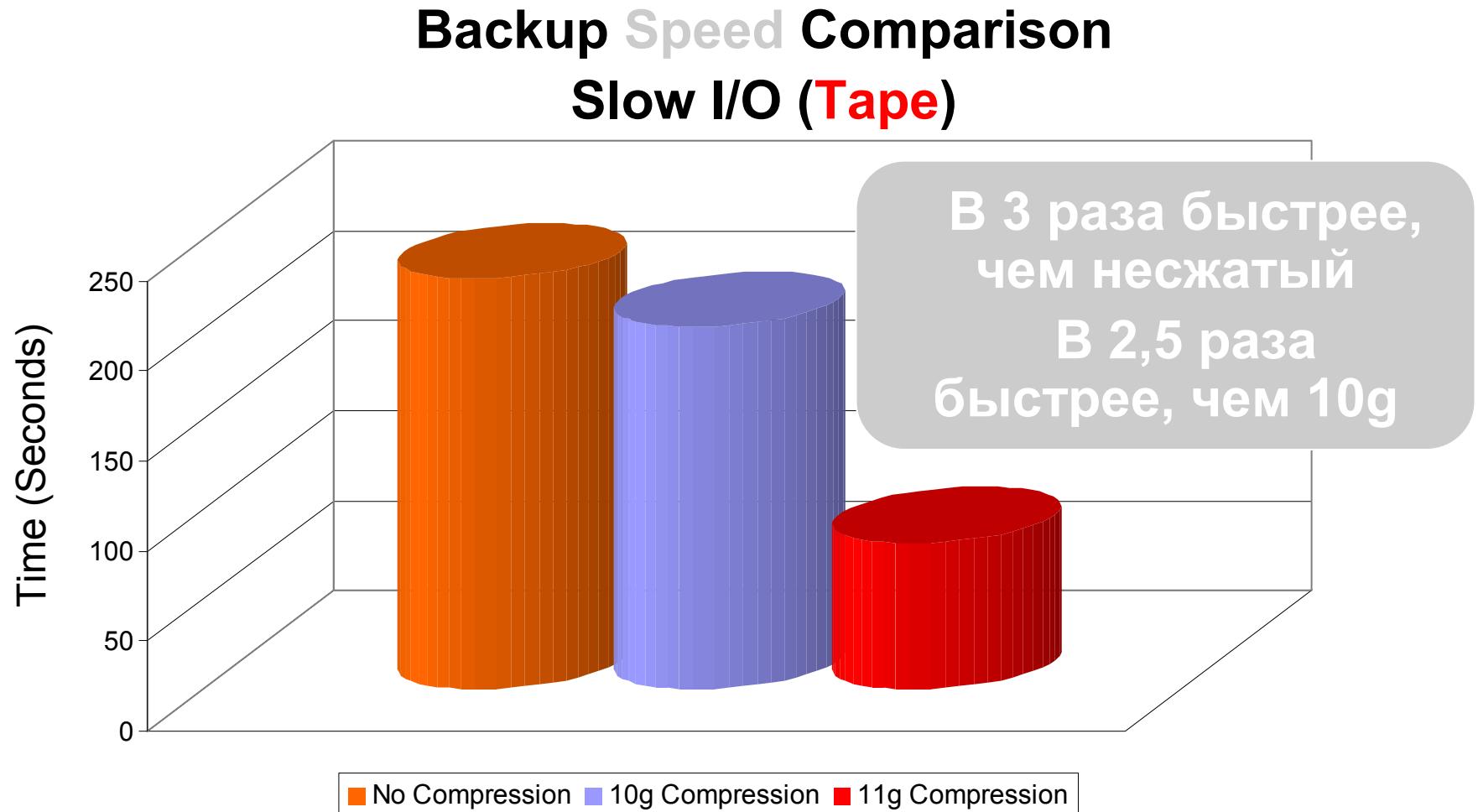
- 11g RMAN без сжатия
- 10g RMAN со сжатием
- 11g RMAN с БЫСТРЫМ сжатием



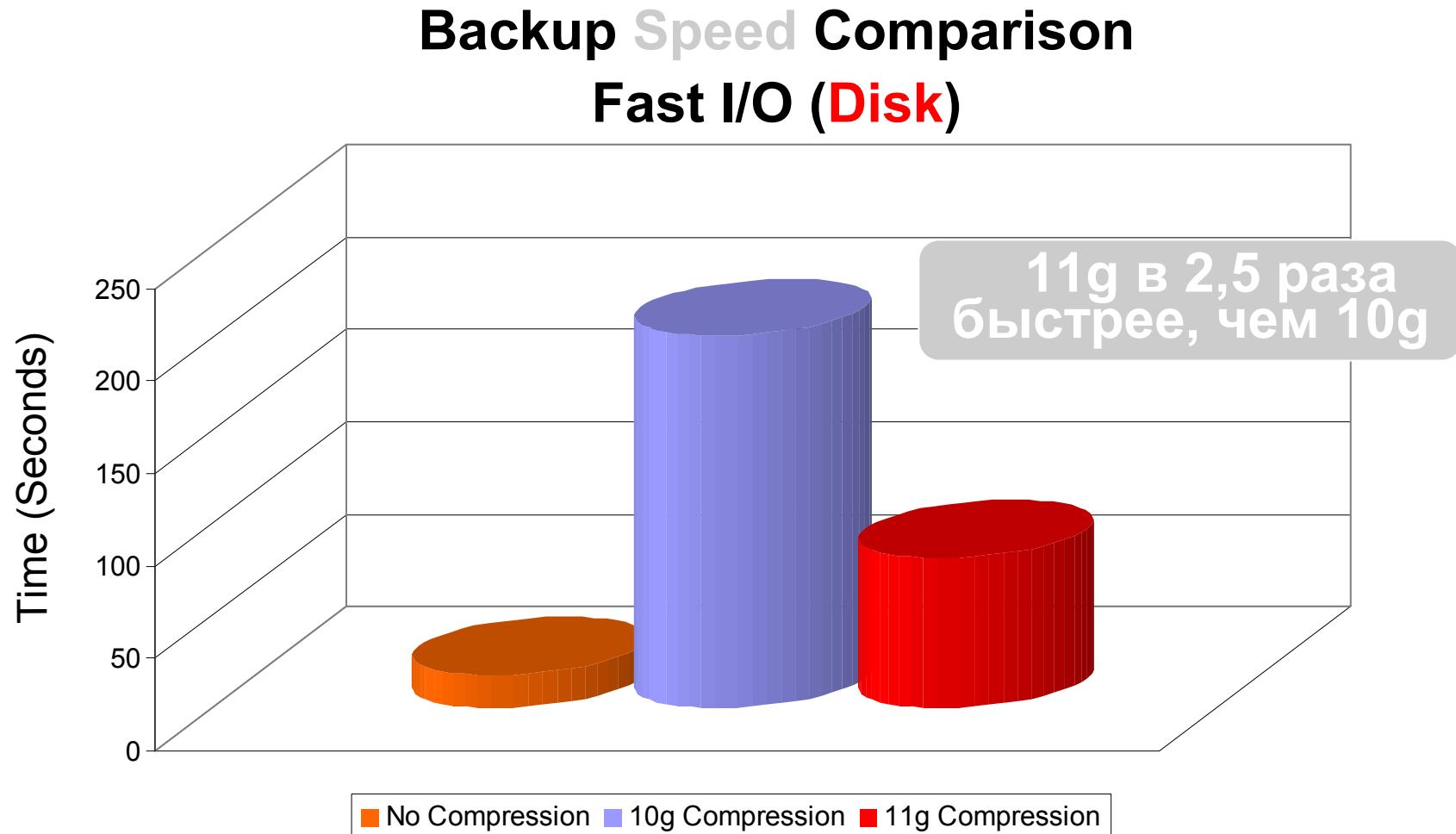
# Сжатие RMA. Тестирование



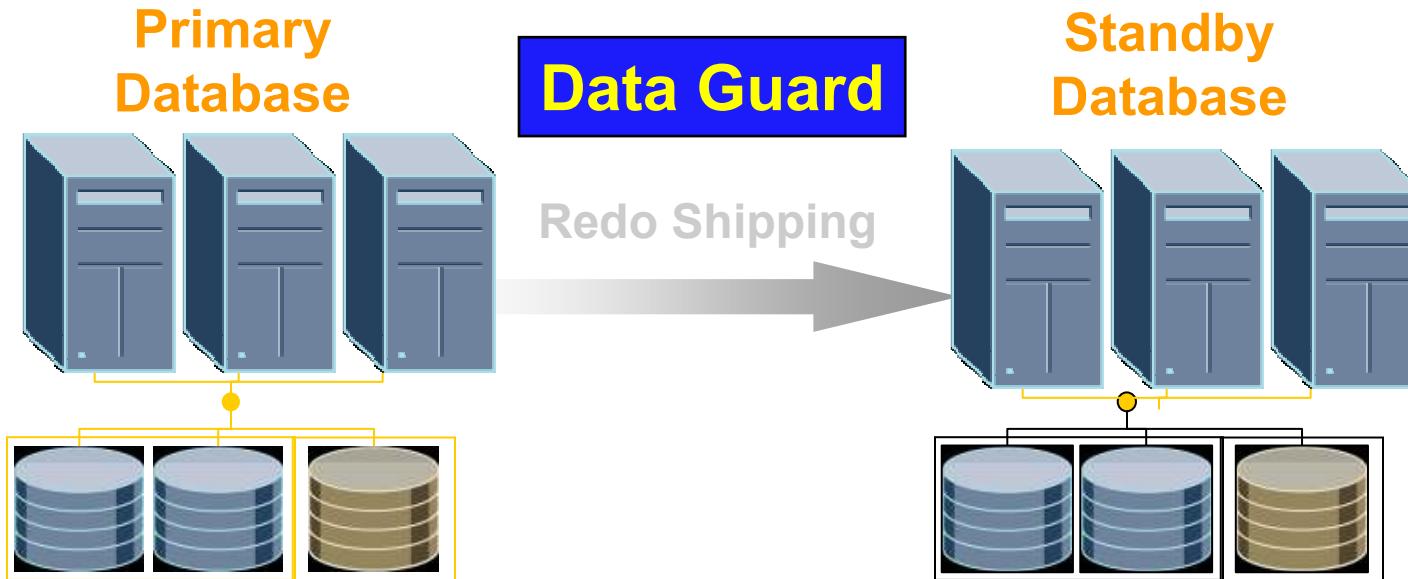
# Сжатие RMA. Тестирование



# Сжатие RMA. Тестирование



# Oracle Data Guard



## Redo Shipping

- Посыпает redo данные по сети с primary на standby
- Размер redo данных обычно небольшой (transactional) и сеть не является узким местом

## Gap Resolution

- После перебоев в сети – ресинхронизация standby
- Размер redo данных гораздо больше

ORACLE

## **Сетевое сжатие**

*Oracle Data Guard Redo Transport Services*

**Быстрая синхронизация standby после перебоев в сети**

**Меньше нагрузка на сеть (<100Mbps)**

- 15-35% меньше времени для передачи 1 GB данных
- Загрузка сети меньше до 35%

**Высокоскоростные сети (>100 Mbps)**

- Компрессия не уменьшает время передачи
- Загрузка сети по прежнему меньше до 35%

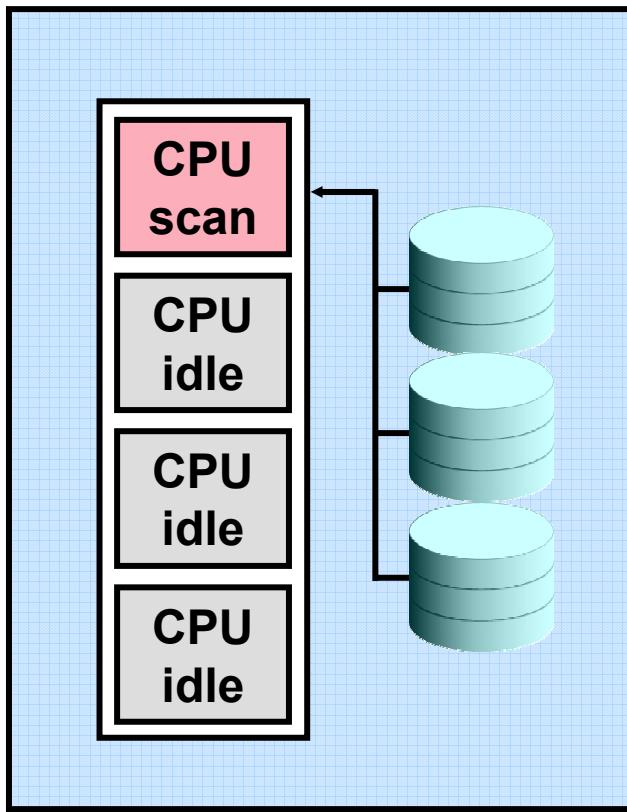


# Parallel

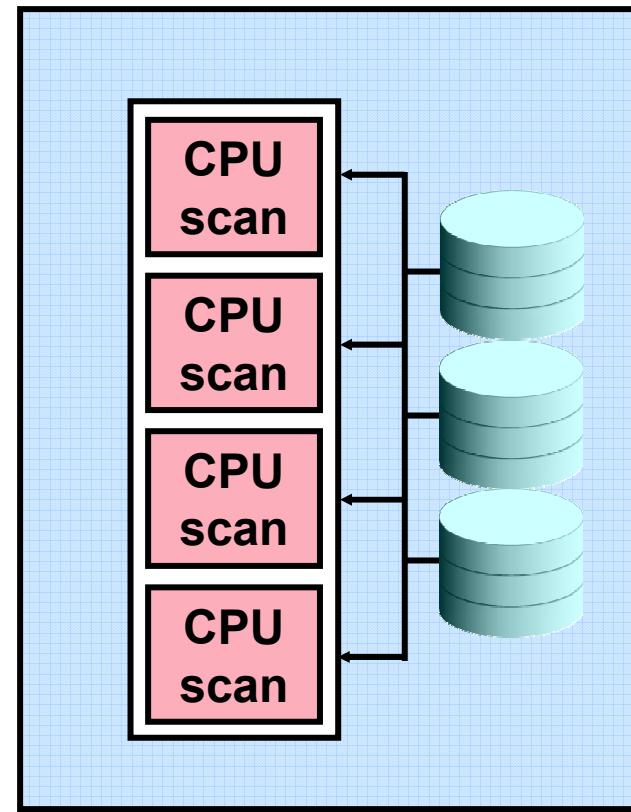
ORACLE

Copyright © 2008, Oracle. All rights reserved.

# Введение в параллельное исполнение



Сервер без параллелизма



Сервер с параллелизмом

# Распараллеливание по-умолчанию

По-умолчанию DOP (degree of parallelism):

- В параллельных операциях, явно не указавших степень
- Вычисляется динамически
- Зависит от:
  - Количество CPU
  - PARALLEL\_THREADS\_PER\_CPU
- Может быть уменьшена, если недостаточно процессов



# Параметры

- PARALLEL\_MIN\_SERVERS.
- PARALLEL\_MAX\_SERVERS.
- PARALLEL\_MIN\_PERCENT
- PARALLEL\_ADAPTIVE\_MULTI\_USER

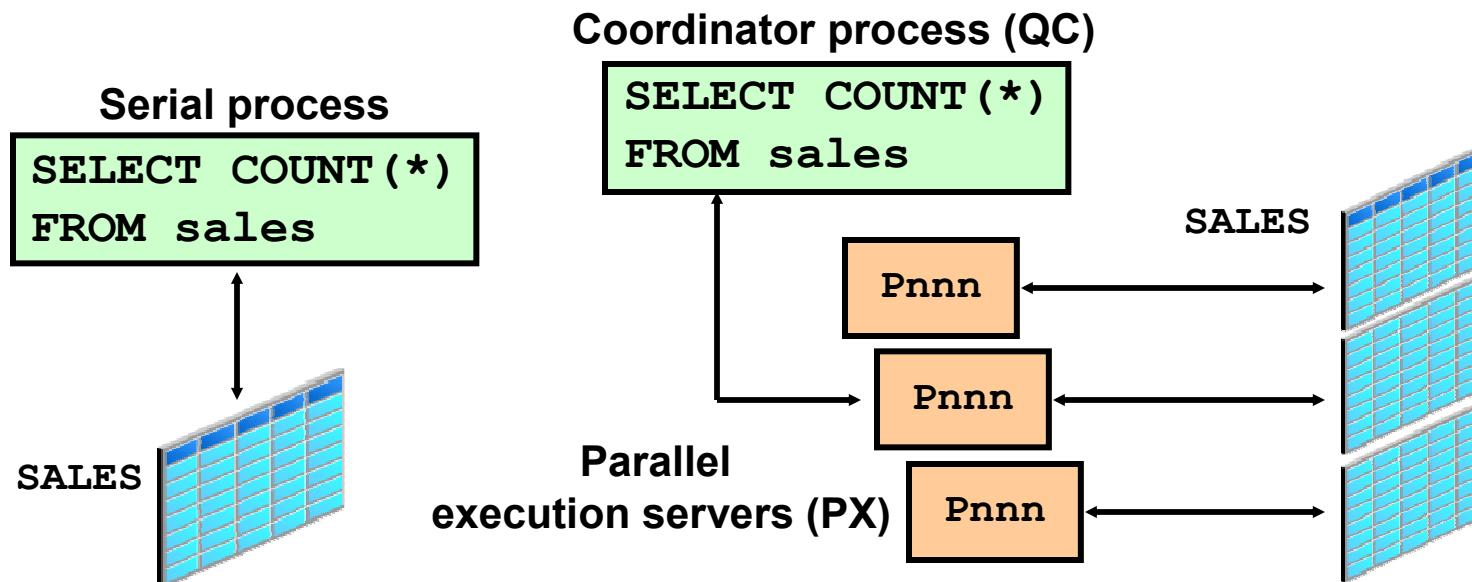


# Распараллеливаемые операции

- Метод доступа
  - Table scans, fast full index scans
  - Partitioned index range scans
- Соединения:
  - Nested loop, sort merge
  - Hash, star transformation, partitionwise join
- DDL выражения:
  - CTAS, CREATE INDEX, REBUILD INDEX [PARTITION]
  - MOVE, SPLIT, COALESCE PARTITION
- DML выражения :
  - INSERT SELECT, UPDATE, DELETE, MERGE

# Параллельное сканирование таблицы

- Один координатор (QC)
- Несколько серверных процессов
- Таблица динамически делится на гранулы



# Включение параллельных DML, DDL и запросов

- Фраза ALTER SESSION включает режим параллельного исполнения команд:

```
ALTER SESSION ( ENABLE | DISABLE | FORCE )
PARALLEL ( DML | DDL | QUERY ) ( PARALLEL n );
```

- FORCE ... PARALLEL n перекрывает настройки DOP
- Можно перекрыть на уровне запроса

# Подсказка, выражение PARALLEL

- PARALLEL (*table\_name*, *DOP\_value*)

```
SELECT /*+PARALLEL(sales,9)*/ *
FROM sales;
```

- NOPARALLEL (*table\_name*)
- PARALLEL\_INDEX (*table\_name*, *index*, *DOP\_value*)
- Фраза PARALLEL

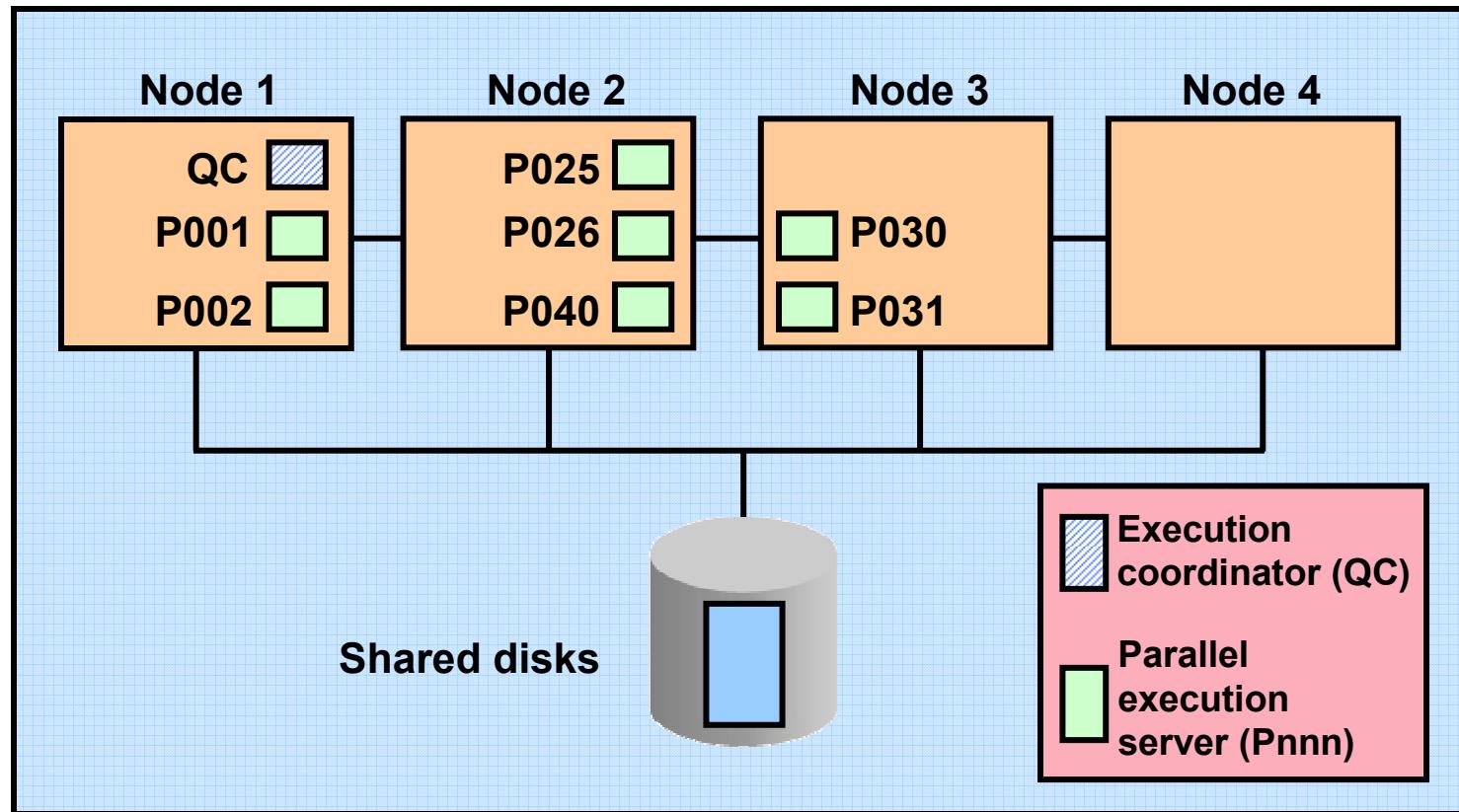
```
CREATE INDEX ord_customer_ix ON oe.orders
(customer_id) NOLOGGING PARALLEL;
```

```
ALTER TABLE customers PARALLEL 5;
```



# Parallel Execution with Real Application Clusters (RAC)

Execution slaves have node affinity for the execution coordinator, but they expand if needed.



# Demo