

# ORACLE®

## *Oracle RAT: Снижение рисков при миграции со старых версий СУБД Oracle*



**Марк Ривкин**  
**Российское представительство Oracle**

ORACLE®

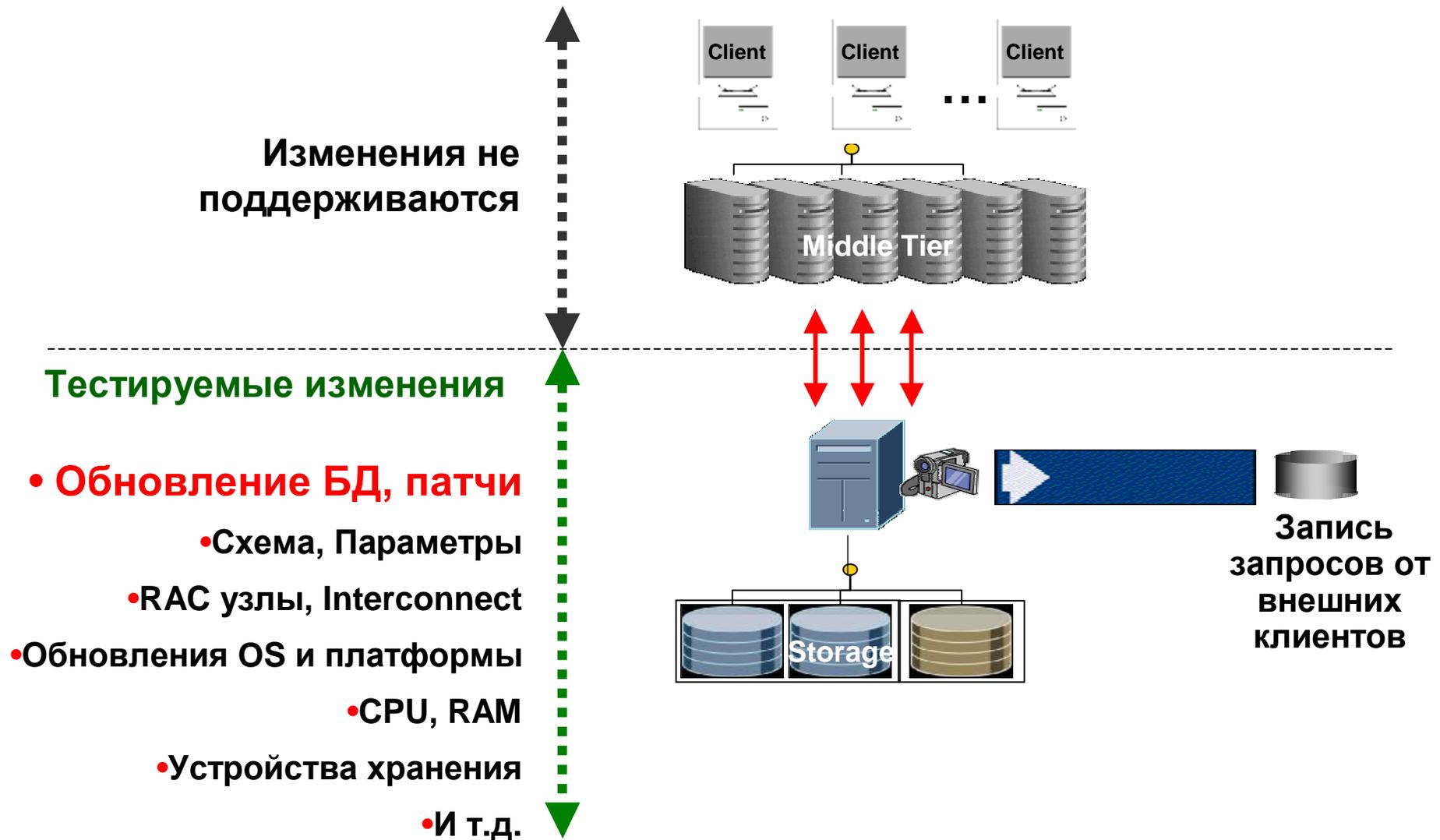
# Надо оценивать влияние изменений

- Изменения в ПО и оборудовании факт нашей жизни.
- Заказчики хотят оценить влияние изменений прежде, чем их осуществлять.
- Полноценное тестирование требует времени и денег.
- Результат может оказаться плохим несмотря на дорогое тестирование
  - Много проблем оказались не выявленными
  - Изменения негативно сказались на производительности и доступности системы
- Основная причина неудачного тестирования -
  - Неспособность воспроизвести реальную нагрузку
- RAT позволяет провести полноценное тестирование с **РЕАЛЬНОЙ** нагрузкой.

# Почему мы сейчас об этом говорим

- Огромное число заказчиков еще работает на 9i и 10g и боятся переходить на новые версии
- Эта опция нужна всем, т к изменения происходят все время
- Показать простоту использования и научить использовать
- В условиях кризиса
  - Простои недопустимы
  - Нужно более эффективно использовать оборудование
  - Downgrade или объединение приложений (RAC)

# RAT: Тестируемые изменения

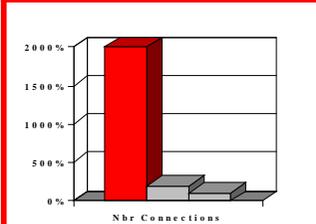


ORACLE

# Почему надо делать Upgrade?

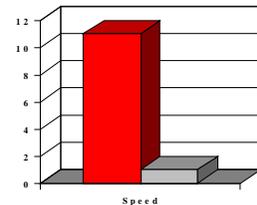
## Преимущества: Oracle Database 11g vs. 10g

### Database Resident Connection Pool



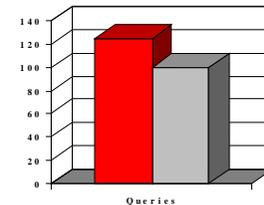
Up to  
20 x connections

### Java Just-In-Time Compiler



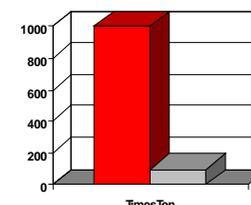
Up to  
11 x Faster

### Query Result Caching



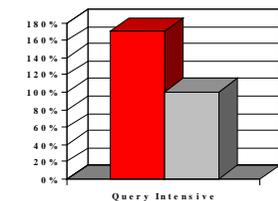
Up to  
25% Faster

### TimesTen Cache



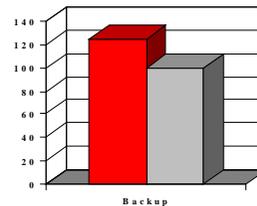
Up to  
10 x Faster

### RAC Performance Enhancements



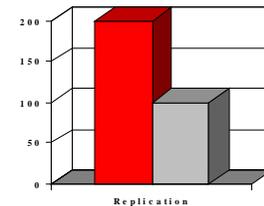
Up to  
70% Faster

### Oracle Secure Backup



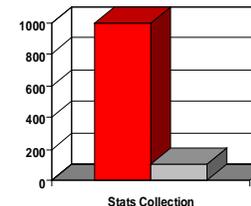
Up to  
25% Faster

### Oracle Streams Enhancements



Up to  
2 x Faster

### Optimizer Stats Collection



Up to  
10 x Faster

ORACLE

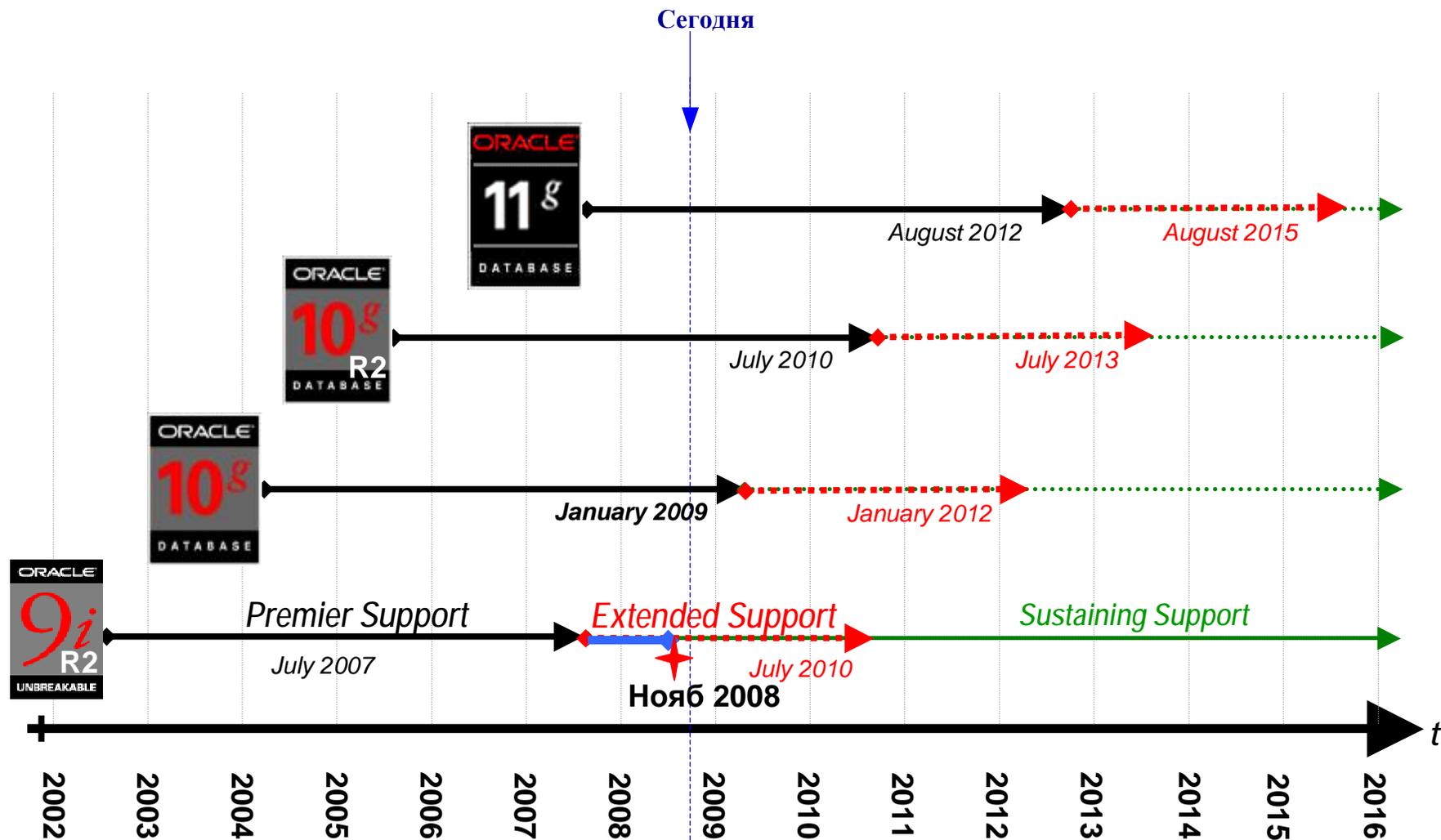
# НОВЫЕ ВОЗМОЖНОСТИ

- Сжатие
- ILM
- Измерение времени в БД (flashback, total recall)
- Active DataGuard
- GRID
- DataVault
- Самоуправляемость
- ASM
- SecureFiles (быстрые LOB)
- Advisors
- Streams
- И т д
- И т п
- .....

ORACLE

# Почему надо делать Upgrade?

## Политика технической поддержки



ORACLE

<http://www.oracle.com/support/library/data-sheet/oracle-lifetime-support-policy-datasheet.pdf>

# Real Application Testing

- ∅ Real Application Testing = SPA + Database Replay
- ∅ SPA + Database Replay дополняют друг друга
- ∅ Вместе они обеспечивают полное тестовое решение

## • Database Replay

- Предсказывает влияние изменений на весь workload
- Оценивает влияние изменений проигрывая весь workload с соблюдением одновременности и зависимостей
- Полное тестирование всего стека DB, например, память, RAC, параллелизм
- Можно использовать для миграции с
  - **9.2 or 10.2 à 11g**

## • SQL Performance Analyzer

- Предсказывает влияние изменений на производительность SQL
- Оценивает влияние изменений индивидуально выполняя query part of SQL (без одновременности)
- Тестирует влияние разных факторов на конкретный SQL, например, изменение статистики/конфигурации, drop index, и т д.
- Можно использовать для миграции с
  - **9.2/10.1 à 10.2 or 11g**
  - **10.2 à 10.2 or 11g**

ORACLE

# Real Application Testing для предыдущих релизов

- Вначале RAT сделали для 11.1
- Database Replay: Захват на старых релизах; проигрывание на 11.1 и выше
- SQL Performance Analyzer: Выполнение тестов на 10.2 и выше
- Нужны патчи на 9i и 10g

Компонента	Миграция с	Миграция на
Database Replay	10g R2	11g
	9i R2	11g
SQL Performance Analyzer	10g R2	10g R2 or 11g
	10g R1	10g R2 or 11g
	9i R2	10g R2 or 11g

ORACLE

† ML 560977.1: Real Application Testing for Earlier Releases

# Шаги миграции

1. Создание тестовой копии production system
2. Захват нагрузки/SQL на production system
3. Upgrade тестовой БД и ПО
4. Воспроизведение промышленной нагрузки на тестовой БД
5. Сравнение и анализ результатов миграции
6. Настройка тестовой СУБД
7. Миграция production system - делать не будем

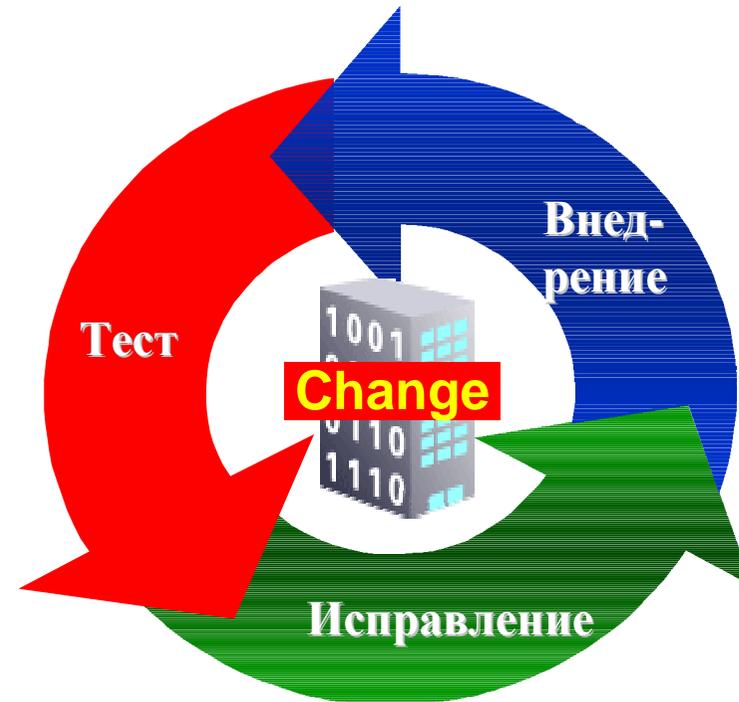
Каждый шаг имеет много вариантов выполнения и свои проблемы и особенности, о них и поговорим

# Программа

- Не только презентации но и демонстрация каждого шага
- Базовый сценарий – миграция с 9i на 11g
- Расскажем про особенности применения этого сценария для 10g
- Дано: БД Oracle 9i (9.2.0.8)
- Тестовую 11.1.0.7 создадим на ее основе и с ней будем работать
- Заготовили БД 10.2, чтоб показать захват/тестирование нагрузки (SPA) на 10g

# Real Application Testing

- **Ценность**
  - Ускорение внесения изменений
  - Более высокое качество тестирования
- **Польза для бизнеса**
  - Снижение стоимости
  - Снижение рисков



**Real Application Testing позволяет провести реалистичное тестирование реальных систем**

ORACLE

# Способы создания тестовой среды (БД и СУБД)

ORACLE

**Технологии, способы и  
проблемы UPGRADE  
(БД и СУБД)  
9.2.0.8 è 11.1.0.7**

ORACLE

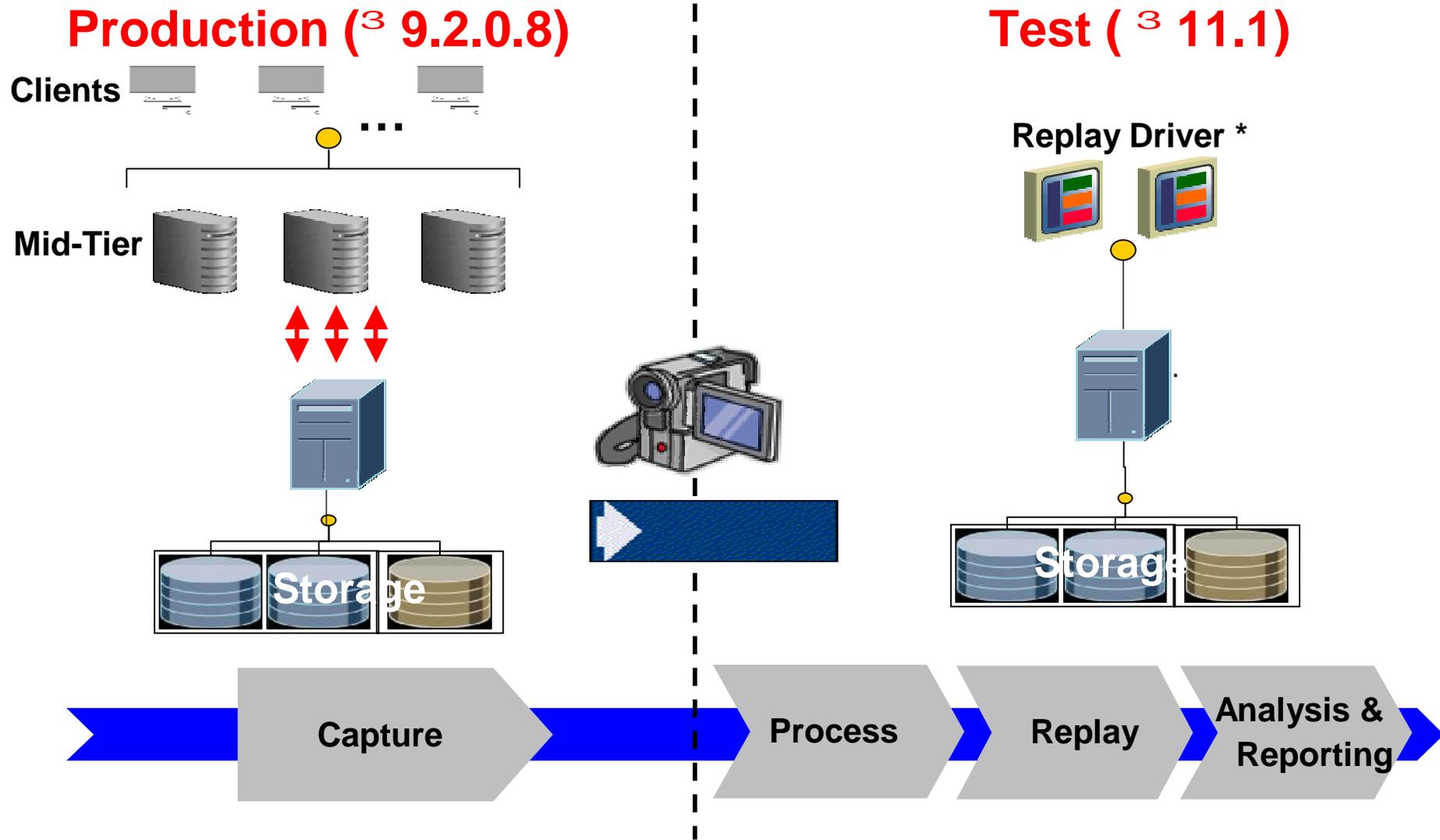
# Database Replay

ORACLE

# Database Replay

- Patch на 9.2.0.8 или 10.2
- Далее все стандартно
- DBConsole или API
- Replay только на 11g

# Database Replay Workflow



ORACLE

\* No middle/client tier setup required

# Database Replay

- Воспроизведение реальной нагрузки на базу данных на тестовой системе
- Выявление, анализ и исправление потенциальных проблем прежде, чем сделать изменения на рабочей системе

- **Захват нагрузки на рабочей БД**

- Захват полной рабочей нагрузки на рабочую БД, включая загрузку и параллельность работы
- Перемещение нагрузки на тестовую систему



- **Воспроизведение нагрузки на тестовой системе**

- Делаем планируемые изменения на тестовой системе
- Воспроизводим нагрузку и параллельность
- Сохраняем порядок транзакций



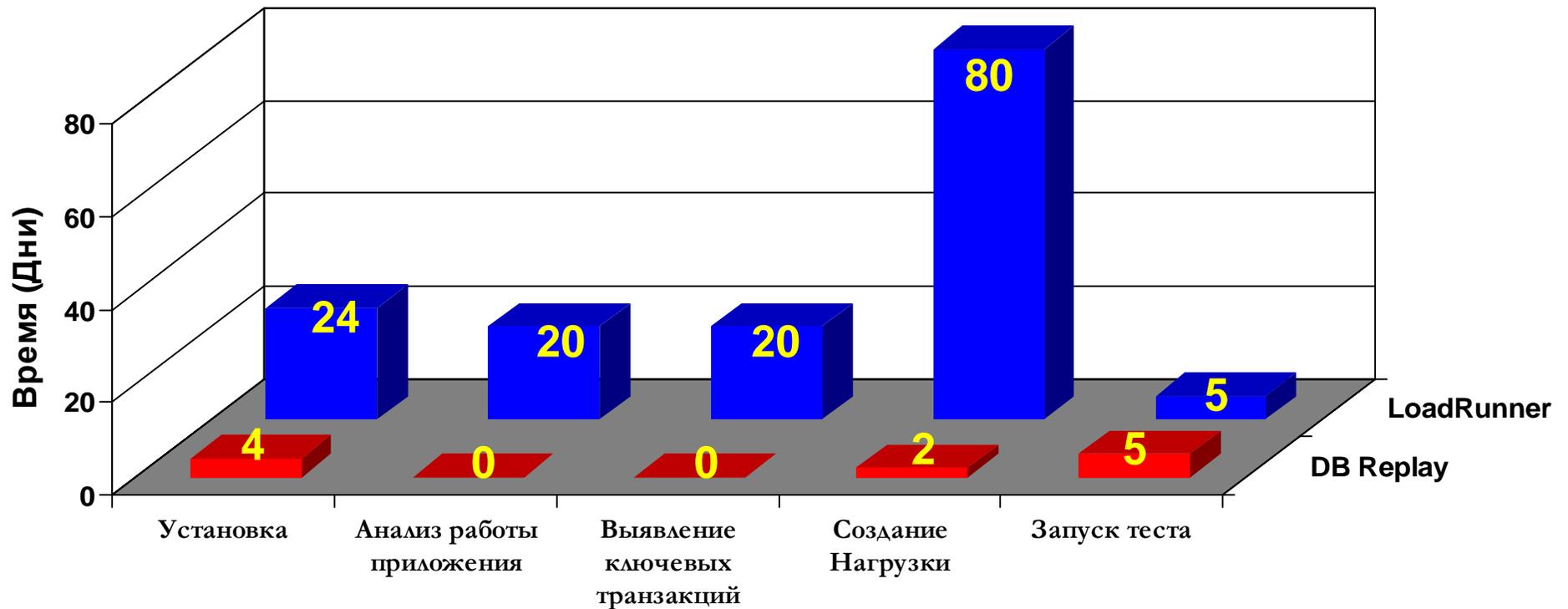
- **Анализ & Отчёты**

- Ошибки
- Расхождение в данных
- Расхождение в производительности

## Анализ & Отчёты

ORACLE

# Сравнение LoadRunner & DB Replay Тестирование e-Business Suite



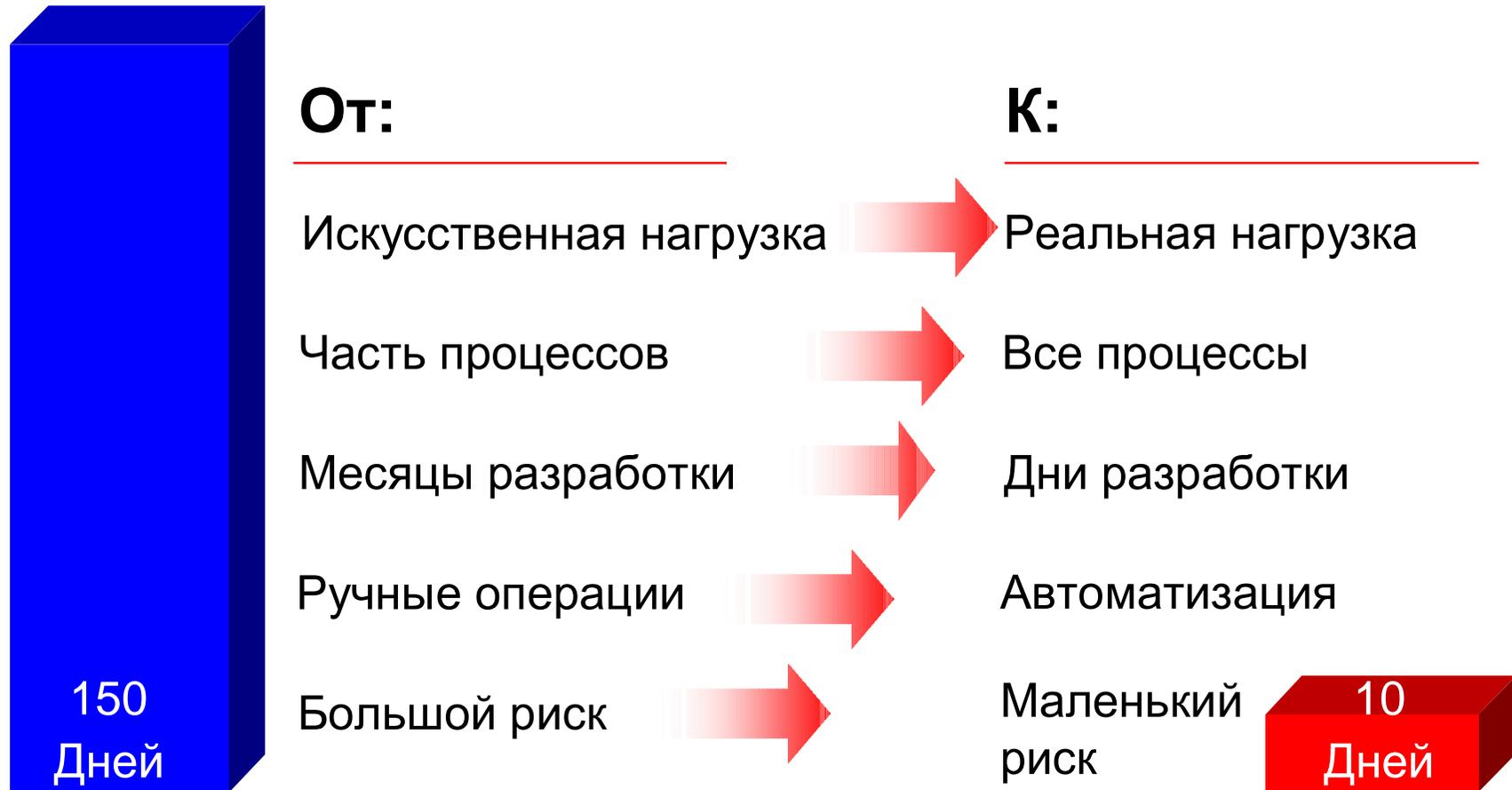
## Общее время тестирования

**DB Replay: 2 недели**

**LoadRunner: 30 недель**

ORACLE

# Почему DB Replay?



ORACLE

# Шаг 1: Захват Нагрузки

- Все запросы от внешних клиентов записываются в двоичные файлы
- Системные процессы и внутренние операции исключаются
- Минимальные накладные расходы (< 5% CPU)
- Для RAC поддерживаются разделяемая и локальные файловые системы
- Рекомендуется записывать нагрузку в момент пиковой загрузки или в другие важные для бизнеса периоды



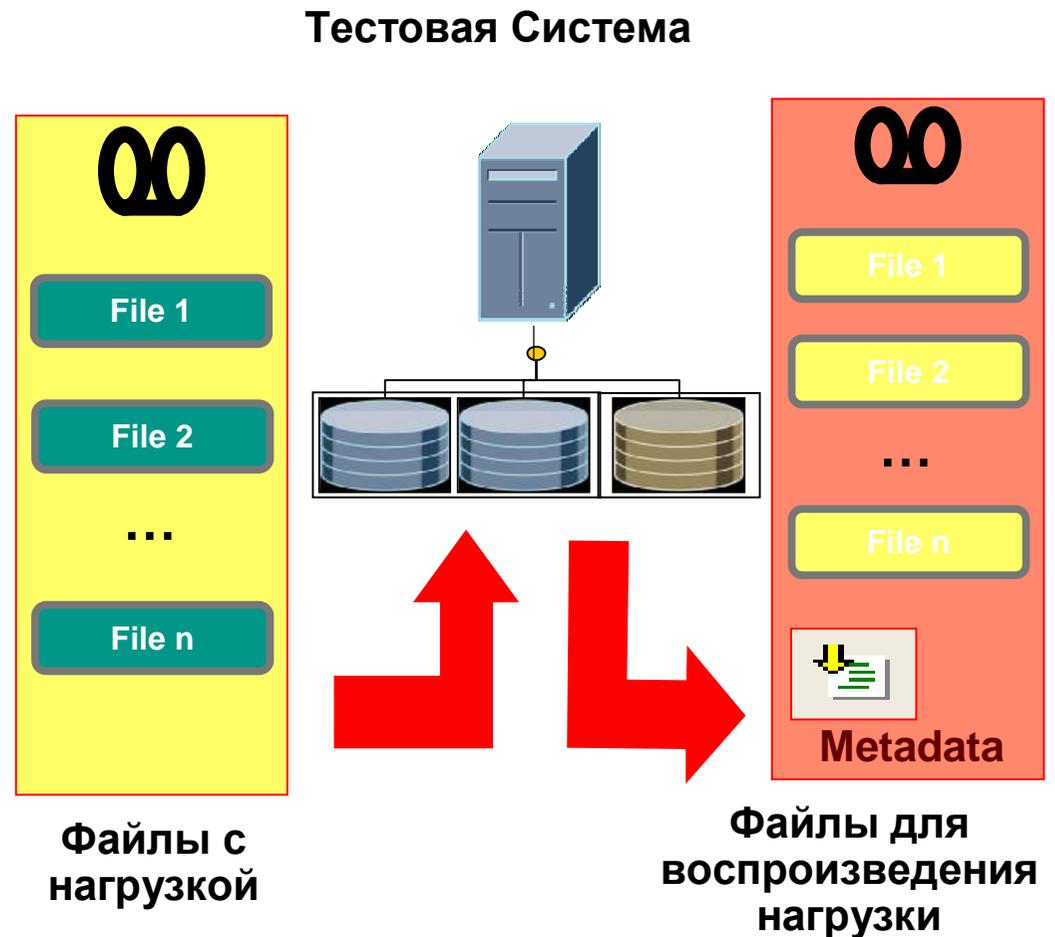
ORACLE

# Параметры захвата нагрузки

- **Для захвата нагрузки можно включить фильтры**
  - Типы фильтров
    - Включающие: Указывают, какие сессии будут записываться
    - Исключающие: Указывают сессии, которые НЕ будут записываться
  - Параметры фильтрации: Захват нагрузки может быть отфильтрован по следующим параметрам
    - Имя пользователя
    - Программа
    - Модуль
    - Операция
    - Сервис
    - ID сессии
- **Запись нагрузки может осуществляться немедленно или по расписанию через какое-то время**
- **На время или до прерывания**

## Шаг 2: Обработка записанной нагрузки

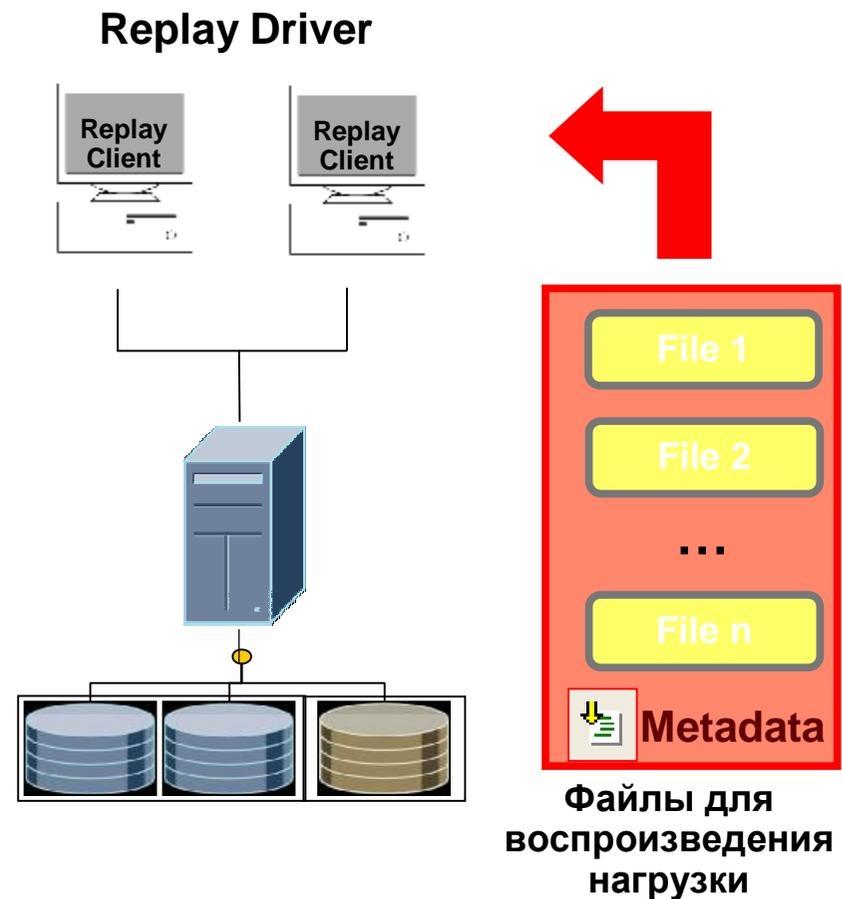
- Создать тестовую систему
  - Данные должны соответствовать рабочей системе
  - Можно использовать
    - RMAN для восстановления данных из резервной копии
    - Snapshot standby
    - imp/exp, Data Pump и т.д.
- Преобразовать файлы с записанной нагрузкой в файлы для воспроизведения, создав необходимые метаданные
- Обработка должна делаться на той же версии БД, где будет проводиться тестирование
- Рекомендуется использовать тестовую систему
- После обработки файлы могут проигрываться много раз
- Для RAC, если использовалась локальная файловая система, надо собрать все файлы с нагрузкой в одну директорию для обработки



ORACLE

# Шаг 3: Воспроизведение нагрузки

- Replay Driver – специальная клиентская программа, которая читает обработанные файлы с нагрузкой и посылает запросы тестовой системе
- При воспроизведении сохраняются время, параллельность и зависимости, которые были на рабочей системе
- Replay Driver состоит из одного или более клиентов. Для воспроизведения нагрузки от большого количества сессий может понадобиться стартовать несколько клиентов
- Воспроизводить нагрузку можно после старта всех клиентов



ORACLE

# Параметры воспроизведения нагрузки

- Синхронное воспроизведение
  - Нагрузка воспроизводится в полностью синхронном режиме
  - Те же самые параллельность и интервалы между операциями, что и на рабочей системе
  - Соблюдается порядок завершения транзакций
  - Минимальное отклонение в данных
- Несинхронное воспроизведение
  - Полезно для нагрузочного тестирования
  - Большое отклонение в данных
  - Три параметра определяют степень синхронизации
    - Think time – время между операциями
    - Commit order – соблюдение порядка транзакций
    - Connect (logon) time – скорость подключения сессий

# Параметры воспроизведения нагрузки

- Количество клиентов воспроизведения
  - Задаётся пользователем
  - Client Calibration Advisor рекомендует количество клиентов для заданной нагрузки
  - Клиенты многопоточковые программы и могут проигрывать запросы от нескольких сессий одновременно

# Анализ & Отчёты

- После воспроизведения нагрузки предоставляются детальные отчёты
- Публикуется 3 типа расхождений
  - **Расхождение в данных:** Количество строк сравнивается для каждого вызова



- **Расхождение в ошибках:**
  - Новые: ошибки, которых не было при записи нагрузки
  - Пропавшие: ошибки, которые не воспроизвелись
  - Изменившиеся: ошибка при воспроизведении отличается от той, которая была при записи нагрузки
- **Расхождение в производительности**
  - Захват и Воспроизведение Отчёт: Публикует общую информацию о производительности
  - ADDM Отчёт: Детальный анализ производительности
  - AWR, ASH Отчёты: Сравнительный анализ отклонений

# Поддерживаемые типы нагрузок

## Поддерживаемые

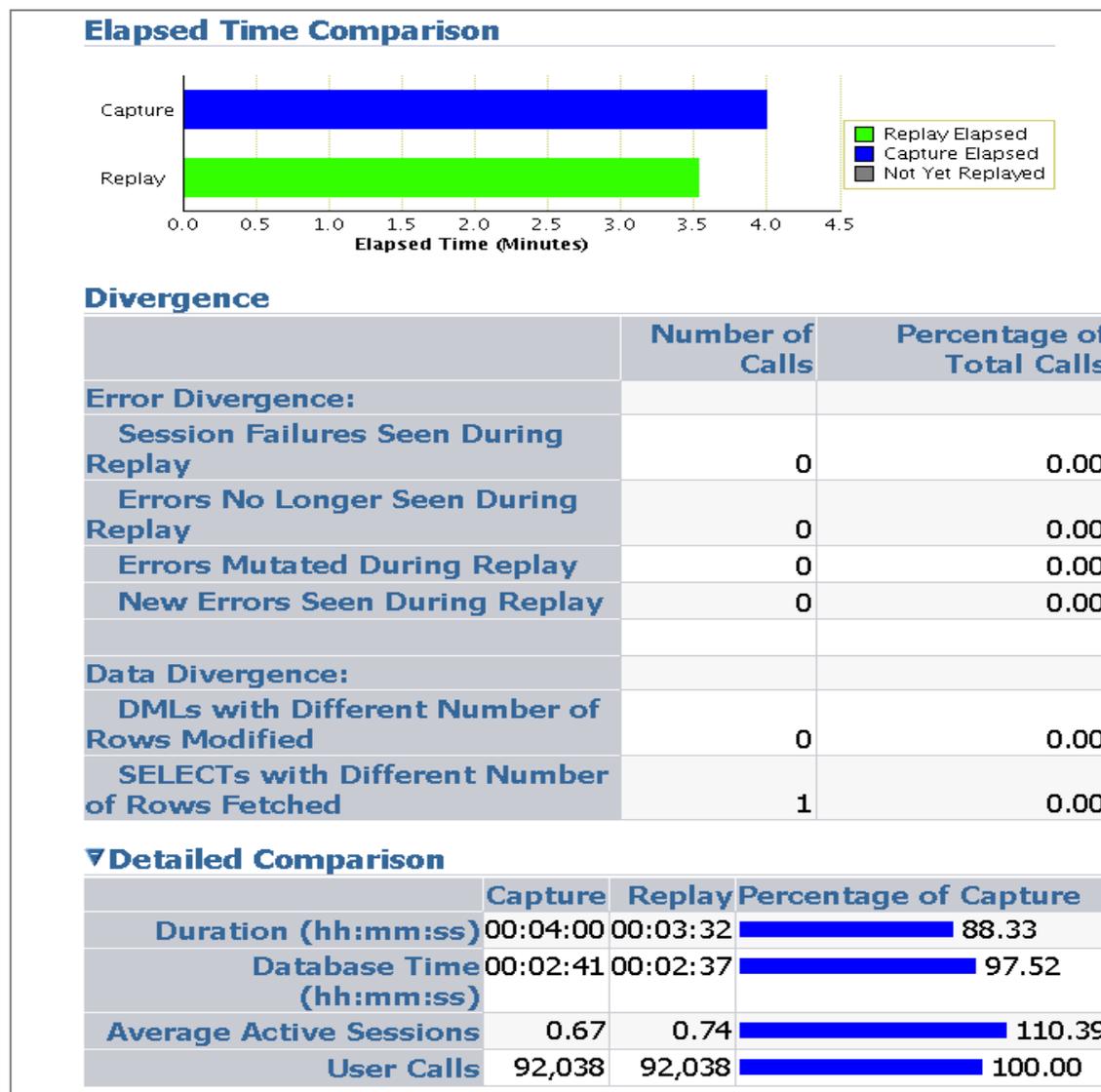
- Все типы SQL (DML, DDL, PLSQL) операций
- Полная поддержка LOB
- Локальные транзакции
- Login/Logoffs
- Переключение сессий
- Ограниченный набор PL/SQL удалённых вызовов процедур

## Ограничения

- Операции прямой загрузки данных (direct path load)
- OCI based object navigation (ADTs) и REF binds
- Streams, не PL/SQL AQ вызовы
- Распределённые транзакции, удалённые describe/commit
- Flashback
- Shared Server

ORACLE

# Database Replay: Анализ



# Отчет Replay:

## Ошибки и расхождение в данных:

Errors No Longer Seen During Replay	0
New Errors Seen During Replay	4
Errors Mutated During Replay	0
DMLs with Different Number of Rows Modified	4

### Sessions

Session ID	Session Serial	Capture Error	Replay Error	Count
291	224	Successful	ORA-00947	4

### SQL: Errors

SQL ID	Capture Error	Replay Error	Count
3vm628apyru8a	Successful	ORA-00947	1
5mjwfkq1s5sxn	Successful	ORA-00947	1
bvp25wxngrdhm	Successful	ORA-00947	1
g24b281jt7cyv	Successful	ORA-00947	1

### SQL: Divergence

SQL ID	Avg Rows Affected	Avg Absolute Rows Affected	Number of Distinct Sessions	Count
3vm628apyru8a	-1	1	1	1
5mjwfkq1s5sxn	-1	1	1	1
bvp25wxngrdhm	-1	1	1	1
g24b281jt7cyv	-1	1	1	1

ORACLE

ORA-947: Not enough values

# Отчет: Capture Analysis

- Capture Profile и отчет
- Подробная статистика о нагрузке
- Метрики production workload
  - Top Events
  - Top Service/Module
  - Top SQL
  - Top Sessions
- Непроигранные, отфильтрованные операции

# Workload Capture Report: Workload Captured

## Workload Captured

- [Top Events Captured](#)
- [Top Service/Module Captured](#)
- [Top SQL Captured](#)
- [Top Sessions Captured](#)

[Back to Top](#)

- Top Events
- Top Service/Module
- Top SQL
- Top Sessions

## Top Events Captured

Event	Event Class	% Event	Avg Active Sessions
CPU + Wait for CPU	CPU	80.81	5.33
db file sequential read	User I/O	5.05	0.33
SQL*Net message to client	Network	1.01	0.07

[Back to Workload Captured](#)

[Back to Top](#)

ORACLE

# Анализ результатов

- Базовая информация: Workload Profile
- Дополнительная информация
  - DB Replay Report
  - AWR Compare Period Report
  - ASH Report

# AWR Compare Period Report: Load Profile

## Pre-Change

Load Profile						
	1st per sec	2nd per sec	%Diff	1st per txn	2nd per txn	%Diff
DB time:	0.68	0.80	17.65	0.09	0.08	-11.11
CPU time:	0.36	0.42	16.67	0.05	0.04	-20.00
Redo size:	141,784.30	182,716.84	28.87	18,478.25	18,369.88	-0.59
Logical reads:	30,539.38	39,676.09	29.92	3,980.09	3,988.93	0.22
Block changes:	726.20	932.80	28.45	94.64	93.78	-0.91
Physical reads:	6,790.88	8,782.25	29.32	885.03	882.95	-0.24
Physical writes:	2.88	1.48	-48.61	0.38	0.15	-60.53
User calls:	338.11	435.87	28.91	44.06	43.82	-0.54
Parses:	15.58	18.65	19.70	2.03	1.88	-7.39
Hard parses:	0.83	0.96	15.66	0.11	0.10	-9.09
Sorts:	4.57	9.00	96.94	0.60	0.90	50.00
Logons:	0.09	0.10	11.11	0.01	0.01	0.00
Executes:	344.89	444.56	28.90	44.95	44.70	-0.56
Transactions:	7.67	9.95	29.73			
				1st	2nd	Diff
% Blocks changed per Read:				2.38	2.35	-0.03
Recursive Call %:				28.01	29.54	1.53
Rollback per transaction %:				0.96	0.76	-0.19
Rows per Sort:				51.54	12.64	-38.90
Avg DB time per Call (sec):				0.00	0.00	-0.00

## Post-Change

Load Profile						
	1st per sec	2nd per sec	%Diff	1st per txn	2nd per txn	%Diff
DB time:	0.68	0.30	-55.88	0.09	0.03	-66.67
CPU time:	0.36	0.20	-44.44	0.05	0.02	-60.00
Redo size:	141,784.30	186,369.33	31.45	18,478.25	18,542.89	0.35
Logical reads:	30,539.38	1,289.19	-95.78	3,980.09	128.27	-96.78
Block changes:	726.20	949.25	30.71	94.64	94.45	-0.20
Physical reads:	6,790.88	0.61	-99.99	885.03	0.06	-99.99
Physical writes:	2.88	1.68	-41.67	0.38	0.17	-55.26
User calls:	338.11	447.90	32.47	44.06	44.56	1.13
Parses:	15.58	17.39	11.62	2.03	1.73	-14.78
Hard parses:	0.83	0.24	-71.08	0.11	0.02	-81.82
Sorts:	4.57	9.52	108.32	0.60	0.95	58.33
Logons:	0.09	0.11	22.22	0.01	0.01	0.00
Executes:	344.89	449.90	30.45	44.95	44.76	-0.42
Transactions:	7.67	10.05	31.03			
				1st	2nd	Diff
% Blocks changed per Read:				2.38	73.63	71.25
Recursive Call %:				28.01	21.66	-6.35
Rollback per transaction %:				0.96	0.68	-0.28
Rows per Sort:				51.54	11.15	-40.39
Avg DB time per Call (sec):				0.00	0.00	-0.00

ORACLE

# AWR Compare Period Report: Top SQL by Elapsed Time

## Pre-Change

SQL Id	Elapsed Time % of DB time					Elapsed Time (ms) per Exec		#Exec/sec (DB time)		CPU Time (ms) per Exec		Physical Reads per Exec		#Rows Processed per Exec		#Executions		#Plans	SQL Text
	1st	1st Total	2nd	2nd Total	Diff	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st/2nd/Both	
<a href="#">1vu8j8vxpak4v</a>			4.88	4.88	4.88		2,048		0.02		855		47.25		1.00		4		BEGIN :1 := dbms_workload_repl...
<a href="#">b6v4z72bxvp2y</a>	39.21	39.21	35.03	39.92	- 4.18	1,967	1,588	0.20	0.22	893	762	50,001.08	50,001.16	3.00	3.00	37	37	1/1/1	SELECT count(pnum) ...
<a href="#">22x9qxj96n6vx</a>	35.09	74.31	33.15	73.06	- 1.95	7,236	6,177	0.05	0.05	3,282	3,055	129.11	128.67	31.00	31.00	9	9	1/1/1	SELECT /* DSS_Q54 */...
<a href="#">gmtgm98c05ag1</a>	3.73	78.04	5.33	78.39	1.60	0	0	467.45	517.23	0	0	0.00	0.00	1.00	1.00	86,748	86,748		INSERT into po values (:SYS_...
<a href="#">auu0bcrau5ff55</a>			1.47	79.86	1.47		615		0.02		179		2.50		1.00		4	/1/1	SELECT XMLCONCAT(:B1 , DBMS_...

## Post-Change

SQL Id	Elapsed Time % of DB time					Elapsed Time (ms) per Exec		#Exec/sec (DB time)		CPU Time (ms) per Exec		Physical Reads per Exec		#Rows Processed per Exec		#Executions		#Plans	SQL Text
	1st	1st Total	2nd	2nd Total	Diff	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st/2nd/Both	
<a href="#">b6v4z72bxvp2y</a>	39.21	39.21	0.23	0.23	- 38.98	1,967	4	0.20	0.59	893	1	50,001.08	0.05	3.00	3.00	37	37	1/1/2	SELECT count(pnum) ...
<a href="#">22x9qxj96n6vx</a>	35.09	74.31	2.98	3.22	- 32.11	7,236	207	0.05	0.14	3,282	134	129.11	1.56	31.00	31.00	9	9	1/1/2	SELECT /* DSS_Q54 */...
<a href="#">1vu8j8vxpak4v</a>			12.44	15.66	12.44		862		0.14		363		4.56		1.00		9		BEGIN :1 := dbms_workload_repl...
<a href="#">gmtgm98c05ag1</a>	3.73	78.04	13.99	29.65	10.26	0	0	467.45	1,391.37	0	0	0.00	0.00	1.00	1.00	86,748	86,748		INSERT into po values (:SYS_...
<a href="#">auu0bcrau5ff55</a>			5.48	35.13	5.48		380		0.14		80		1.11		1.00		9	/1/1	SELECT XMLCONCAT(:B1 , DBMS_...

ORACLE

# Рекомендации

- Заранее продумать стратегию создания тестовой системы
- Данные должны соответствовать рабочей системе
- Перезапустить Production DB перед захватом
- Откатиться к ICN
- Можно использовать
  - RMAN ... Duplicate ... для восстановления данных из резервной копии
  - Snapshot standby
  - imp/exp
  - Data Pump
  - Clone database
  - и т.д.
- В новом OEM будет специальный интерфейс
- Можно много раз делать Flashback test database

# Database Replay: Workflow B OEM

## Database Replay

The Database Replay feature allows database workload to be captured on one system and replayed later on a different system. Replaying a captured workload can be useful to compare two different systems.

Page Refreshed Sep 13, 2008 7:20:12 PM CDT [Refresh](#)

Task	Task Name	Description	Go to Task
1	Capture Workload	Choose this option to capture workload on this database.	
2	Preprocess Captured Workload	Preprocessing will prepare a captured workload for replay. This must be done once for every captured workload.	
3	Replay Workload	Choose this option to replay a preprocessed workload on this database.	

[View Workload Capture History](#)

### Active Capture and Replay

Select	Name	Type	Directory Object	Start Time
	No items found			

## Overview

The following are the typical steps to perform Database Replay:

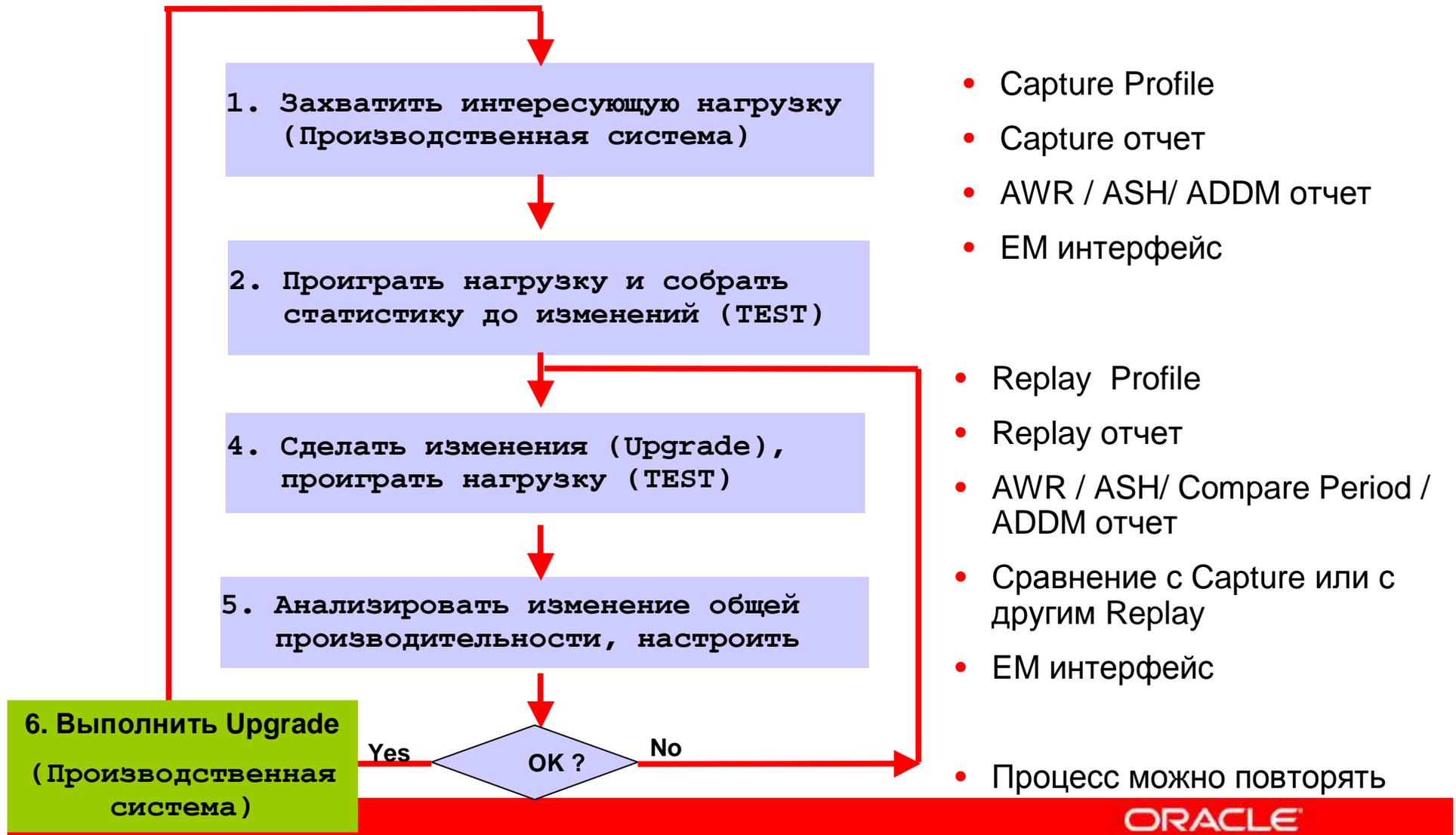
1. Capture the workload on a database. (Task 1)
2. Optionally export the AWR data. (Task 1)
3. Restore the replay database on a test system to match the capture database at the start of the workload capture.
4. Make changes (such as perform an upgrade) to the test system as needed.
5. Copy the workload to the test system.
6. Preprocess the captured workload. (Task 2)
7. Configure the test system for the replay.
8. Replay the workload on the restored database. (Task 3)

ORACLE

# Best Practices

- **Захват**
  - Проверьте наличие достаточного свободного места на диске для создаваемых бинарных файлов (captured workload)
  - Рестартуйте СУБД (Optional): Рекомендуется для идентичности результатов захвата и проигрывания
  - Для RAC, используйте разделяемую файловую систему
- **Создание тестовой среды**
  - Обеспечьте, что данные тестовой среды идентичны данным производственной среды в момент захвата. Это уменьшит data divergence
  - Используйте RMAN backup/restore или Snapshot Standby для создания тестовой среды
  - Для анализа производительности конфигурация HW и SW должны быть похожими
  - Переустановите системные часы так, чтобы они совпадали с часами производственной системы, если используется SYSDATE
- **Обработка файлов**
  - Обработка загружает процессор и может занять много времени
  - Выполняйте обработку на тестовой системе, а не на производственной
- **Воспроизведение**
  - Используйте Client Calibration Advisor, чтоб определить число replay клиентов, необходимых для воссоздания нагрузки

# Анализ влияния Upgrade на производительность



# Database Replay for Earlier Releases

- Облегчает переход на 11g
  - “Capture” функционал Database Replay backported to 9.2 & 10.2
  - **Replay only possible on Oracle Database 11g and above**
  - ML Note: 560977.1: Real Application Testing for Earlier Releases

Upgrade From	Upgrade To	Release /Patches needed	Comments
11.1.0.6	>= 11.1.0.7	Production Release	No one-off patch needed
10.2.0.4	>= 11.1.0.6	Patch-set Release	No one-off patch needed
10.2.0.3	>= 11.1.0.6	10.2.0.3 + One-off patch needed	-
10.2.0.2	>= 11.1.0.6	10.2.0.2 + One-off patch	-
9.2.0.8	>= 11.1.0.6	9.2.0.8 + One-off patch	-

ORACLE

\* For Windows see ML Note

## Надо учесть при Upgrade 9.2.0.8 à 11g

Step	9.2.0.8 à 11g
Захват	<ul style="list-style-type: none"><li>• Исключите мониторинг/фоновую активность (Statspack, etc.)</li><li>• Для RAC, стартуйте capture на каждом instance вручную</li><li>• Не надо устанавливать параметры</li></ul>
Capture Performance Baseline data	<ul style="list-style-type: none"><li>• Нет AWR, используйте Statspack</li><li>• Сделайте ручные snapshots до и после workload</li><li>• Экпортируйте данные</li></ul>
Проигрывание и анализ	<ul style="list-style-type: none"><li>• Ручное сравнение Statspack / AWR</li><li>• 11g: AWR, ASH, ADDM reports available</li></ul>
Исправление ухудшений	<ul style="list-style-type: none"><li>• 11g: Errors/Data Divergence: EM Support Workbench SQL Diag Advisor, SQL Test Case Builder</li><li>• 11g: Performance: ADDM recs, SQL Tuning / Access Advisors, Transport 9i Stored Outlines</li></ul>
UI	<ul style="list-style-type: none"><li>• 9.2 Capture: Command line</li><li>• 11g Replay: DB Control or 10.2.0.5 EM GC when available</li></ul>

ORACLE

## Надо учесть при Upgrade 10.2 à 11g

Step	10.2.0.4 à 11g	10.2.0.2, 10.2.0.3 à 11g
Захват	<ul style="list-style-type: none"> <li>Set init.ora param pre_11g_enable_capture=true</li> </ul>	<ul style="list-style-type: none"> <li>Параметр не нужен</li> </ul>
Capture Performance Baseline data	<ul style="list-style-type: none"> <li>Use AWR</li> </ul>	<ul style="list-style-type: none"> <li>Same</li> </ul>
Проигрывание и анализ	<ul style="list-style-type: none"> <li>11g: AWR, ASH, Compare Period, ADDM reports</li> </ul>	<ul style="list-style-type: none"> <li>Same</li> </ul>
Исправление ухудшений	<ul style="list-style-type: none"> <li>Errors/Data Divergence: 11g EM Support Workbench, SQL Diag Advisor, SQL Test Case Builder</li> <li>Performance: ADDM recs, SQL Tuning / Access Advisors, Transport 10g Stored Outlines</li> </ul>	<ul style="list-style-type: none"> <li>Same</li> </ul>
UI	<ul style="list-style-type: none"> <li>10.2.0.4 Capture: EM DB Control</li> <li>11g Replay: DB Control or 10.2.0.5 EM GC when available</li> </ul>	<ul style="list-style-type: none"> <li>Capture: Command line                             <ul style="list-style-type: none"> <li>11g Replay: Same</li> </ul> </li> </ul>

# EM GC 10.2.0.5 Database Replay

**Database Replay**

Database Replay allows workloads to be captured from production systems and re-executed with high fidelity on test copies of production databases. This enables detailed analysis of the effects that the proposed changes may have on production systems, for instance patching or upgrading database software.

Page Refreshed Aug 31, 2008 1:04:19 AM PDT [Refresh](#)

**Task List**

Task	Task Name	Description	Go to Task
▼ 1	Capture Production Workload	Initiate or schedule workload capture, export AWR data after capture, copy capture files to workload staging area.	
	Capture Workload	Capture workload from production environment. This can be scheduled to accommodate database restart if desired.	
	Export AWR Data	Exporting AWR data allows better performance comparison between captured and replayed workloads.	
	Copy to Workload Staging Area	Copy capture files away from production to the workload staging area for later preprocessing. For a cluster database, capture files from different database instances can be consolidated in the workload staging area.	
▼ 2	Prepare Test Database	Set up test database from production, upgrade or otherwise modify test database, isolate test database prior to replay.	
	Set Up Test Database	Clone the production database to a test environment. The test database should be restored to match the capture database at the start of capture. You may make any changes to the test environment as needed.	
	Isolate Test Database	Isolate the test system from production environment prior to workload replay. This task must be performed on the test database target.	
▼ 3	Prepare for Replay	Prepare workload capture files for replay (preprocess), copy preprocessed workload files to workload staging area, deploy Replay Clients, copy preprocessed workload files to Replay Client hosts.	
	Preprocess Workload	Preprocessing prepares a captured workload for replay. You must do this once for every captured workload. Preprocessing is best performed in the test database. The captured workload must be accessible from the test database.	
	Copy to Workload Staging Area	Copy preprocessed workload files to the workload staging area. The preprocessed workload files must be accessible by the database server and the Replay Clients during replay.	
	Deploy Replay Clients	Deploy the Replay Client to one or more host machines. Replay Clients are used to replay the preprocessed workload.	
	Copy Workload to Replay Client Hosts	Copy the preprocessed workload to one or more Replay Client host machines. Each Replay Client must be able to access the preprocessed workload during replay.	
▼ 4	Replay Workload on Test Database	Set up workload replay on test database, copy replay results to workload staging area, analyze results.	

**▼ Workload Capture History**

[View](#) [Delete](#) [Export AWR Data](#) [Copy to Workload Staging Area](#)

Select	Capture Name	Status	Directory Object	Start SCN	Duration (hh:mm:ss)	Start Time ▾	AWR Data Exported
<input checked="" type="radio"/>	CAPTURE-database-20080828190000	Completed	capture1	420607	00:07:40	Aug 28, 2008 7:01:52 PM PDT	x

# Демонстрация Database Replay 9.2.0.8 è 11.1.0.7

ORACLE

# Но

- Database Replay – только для 11g (а нужно ли на 10g мигрировать если все работает?)
- Многие боятся и ждут 11.2
- Хотят с 9i/10.1 на 10.2.0.4 или с 10.2.x на 10.2.y
- DB replay дает информацию о суммарной производительности, но возможно часть SQL ухудшилась и их можно настроить
- Можно выявить общие проблемы производительности, связанные с Upgrade, и исправить их

# SQL Performance Analyzer (SPA)

ORACLE

# Важность SQL Performance Analyzer (SPA)

- Бизнесу нужны системы, которые имеют необходимую производительность и соответствуют SLA
- Снижение быстродействия SQL – главная причина плохой производительности системы
- Не существовало решения для проактивного выявления регрессии всех SQL
- DBA используют неэффективные и сложные ручные скрипты для выявления проблем



**SPA выявляет все изменения в производительности SQL до того, как они начнут влиять на производительность системы**

ORACLE

# SQL Performance Analyzer (SPA)

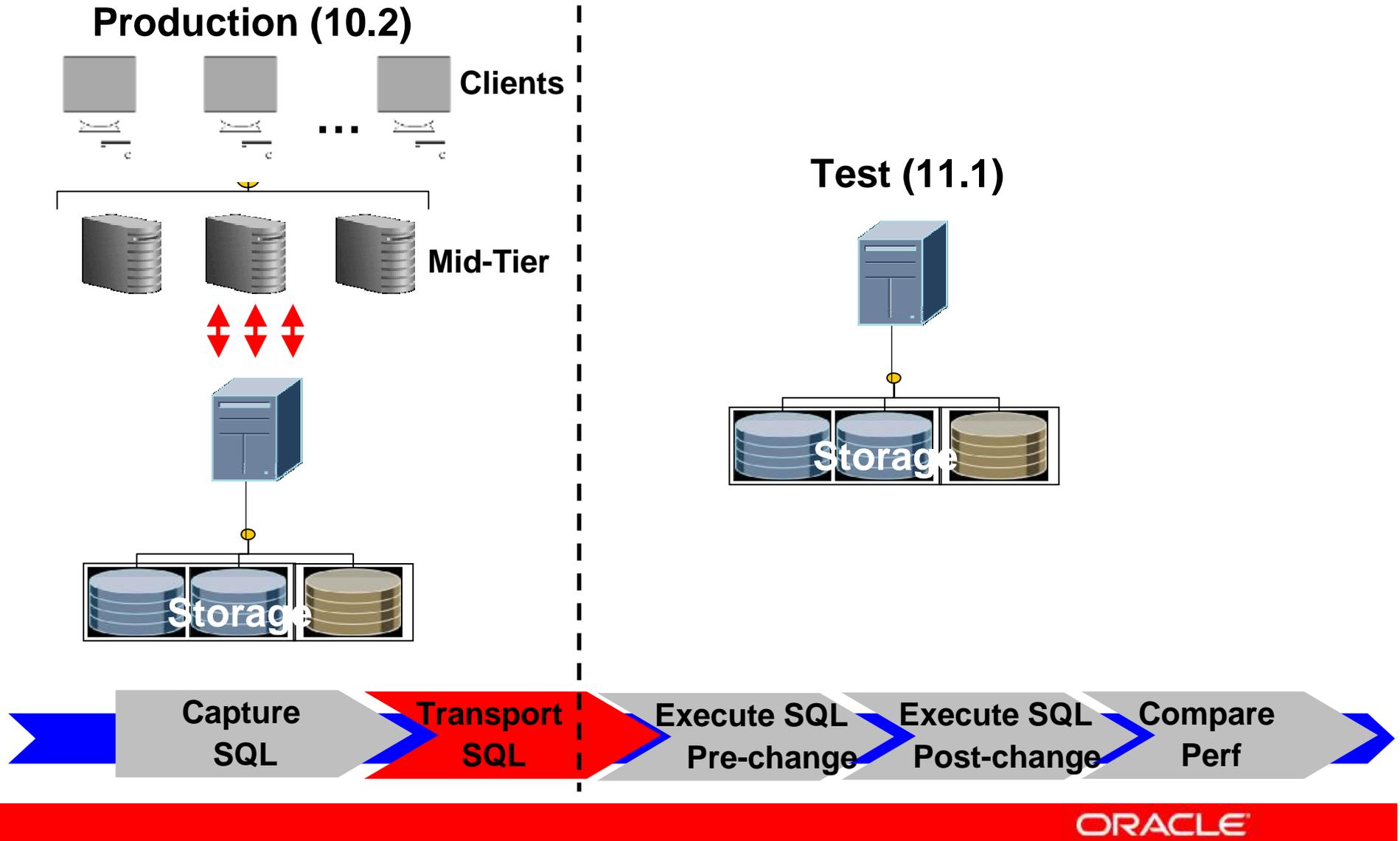
- Тестирует влияние изменений на производительность SQL
- Захват SQL нагрузки на рабочей системе, включая планы и bind переменные
- Выполнение SQL запросов на тестовой машине



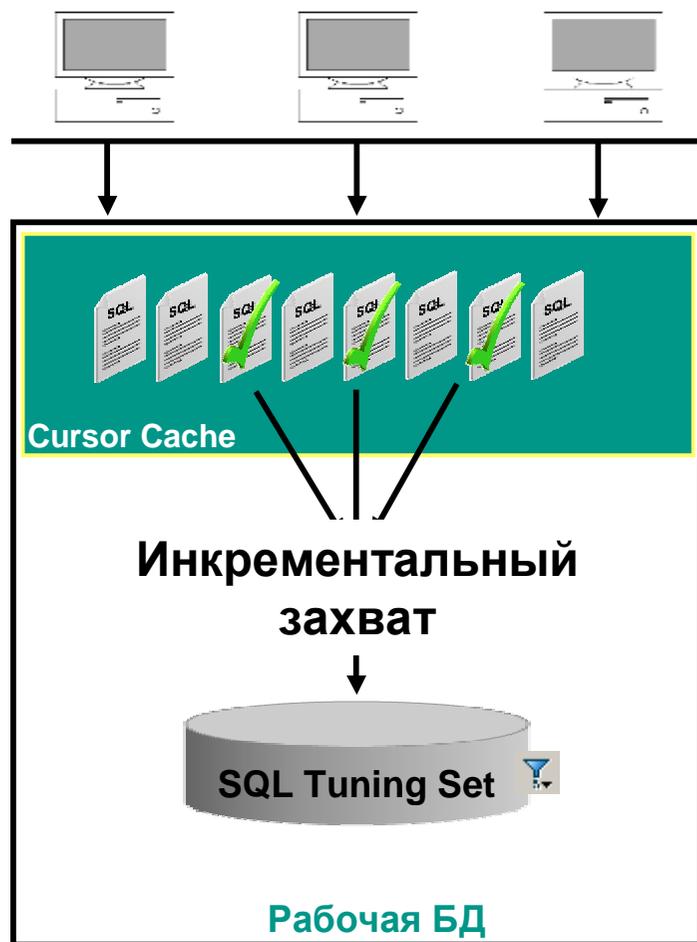
# Польза от SPA

- Обнаружение ухудшения производительности SQL прежде, чем это отразится на конечных пользователях
- SPA полезен при следующих изменениях
  - Обновление БД
  - Изменение статистики оптимизатора
  - Новые индексы, материализованные представления, секционирование и т.д.
- Автоматически отслеживает изменение производительности сотен и тысяч запросов – невозможно сделать вручную
- Минимальные накладные расходы при захвате SQL нагрузки
- Интегрирован с SQL Tuning Advisor и SQL Plan Baselines

# SQL Performance Analyzer Workflow



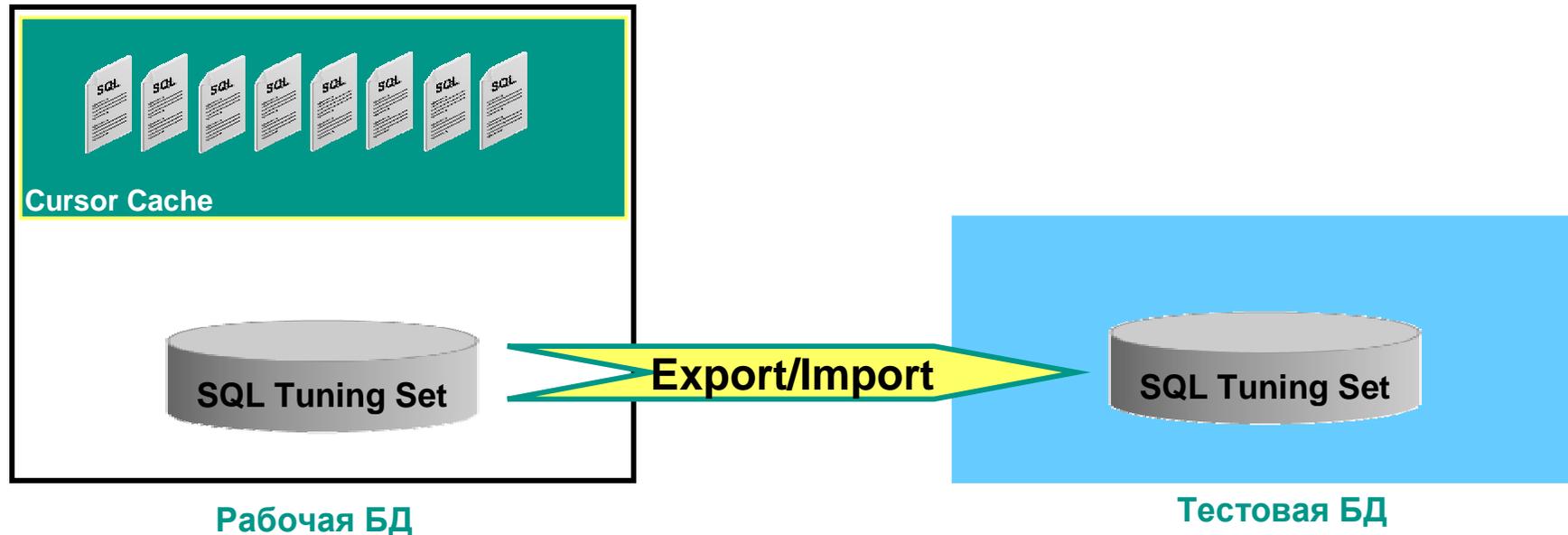
# Шаг 1: Захват SQL нагрузки



- Нагрузка хранится в виде SQL Tuning Set (STS)
- STS включает:
  - SQL текст
  - Bind переменные
  - Планы
  - Статистика
- Инкрементальный захват пополняет STS новыми SQL из кэша курсоров
- SQL можно фильтровать
- SQL нагрузка от 10.2.0.1 и выше может обрабатываться в SPA 11g
- Минимальное влияние на производительность (< 1%)

ORACLE

## Шаг 2: Перенос SQL нагрузки на тестовую систему



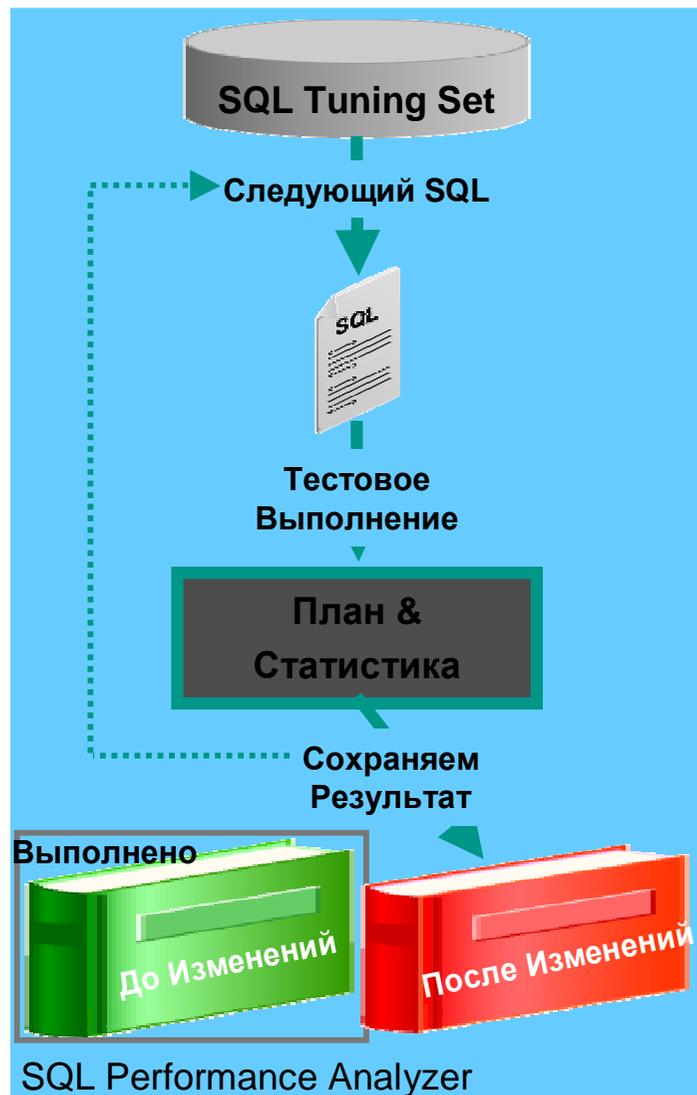
- Копируем SQL tuning set в таблицу (“пакуем”)
- Переносим таблицу на тестовую систему (datapump, db link, и т.д.)
- Копируем SQL tuning set из таблицы (“распаковываем”)

# Шаг 3: Выполняем SQL до изменений



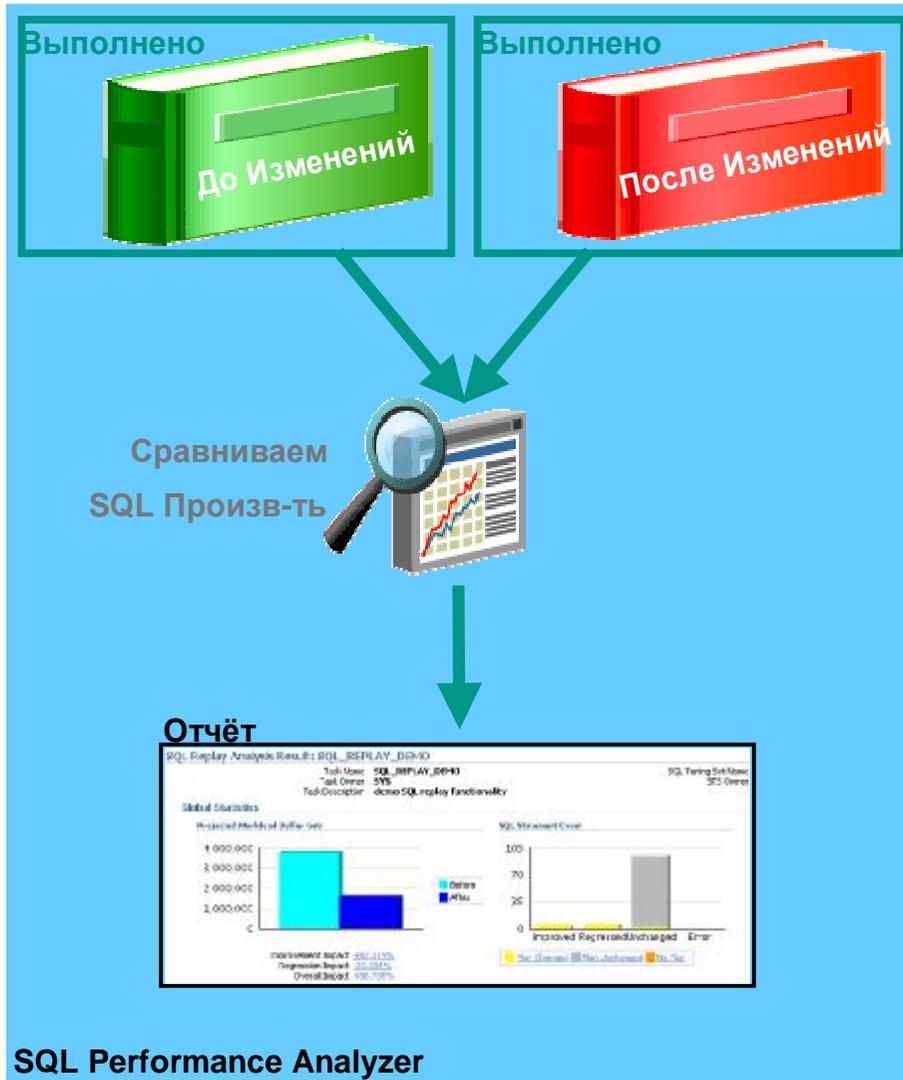
- Производительность SQL до изменений – базис для сравнения
- SQL Trail = планы + статистика выполнения
- Выполняем SQL из SQL tuning set:
  - Получаем планы и статистику
  - SQL выполняется последовательно
  - Каждый SQL выполняется только раз
  - DDL/DML пропускаются
- Можно отказаться от выполнения SQL и делать только Explain Plan анализ
- Для 9i/10.1 берем из trace file
- Для 10g можно выполнить удаленно

# Шаг 4: Выполняем SQL после изменений



- **Производим планируемые изменения**
  - Обновление БД, патчи
  - Сбор статистики для оптимизатора запросов
  - Изменения схемы
  - Изменение параметров БД
  - Выполнение рекомендаций по настройке, например, создание SQL Profiles
  - Обновление OS и оборудования
- **Выполняем SQL после изменений**
  - Получаем новые планы и статистику

## Шаг 5: Сравниваем & Анализируем Производительность



- Сравниваем производительность, используя различные метрики:
  - Elapsed Time
  - Parse Time
  - Execute Elapsed Time
  - Execute CPU Time
  - Buffer Gets
  - Disk Reads
  - Disk Writes
  - Optimizer Cost
- SPA отчёт показывает влияние изменений на каждый SQL
  - Улучшившиеся SQL
  - Ухудшившиеся SQL
  - Не изменившиеся SQL
  - SQL с ошибками
- Исправляем плохие SQL используя SQL Tuning Advisor или SQL Plan Baselines

ORACLE

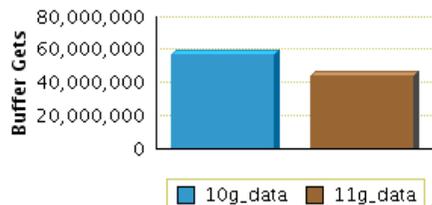
# SPA Report

## SQL Performance Analyzer Task Result: SYS.UPGRADE\_10G11G

Task Name	UPGRADE_10G11G	SQL Tuning Set Name	OOW_54G	Replay Trial 1	10g_data
Task Owner	SYS	STS Owner	SYS	Replay Trial 2	11g_data
Task Description	test upgrade to 11g	Total SQL Statements	54	Comparison Metric	Buffer Gets
		SQL Statements With Errors	0		

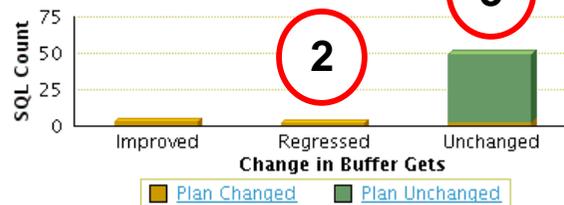
### Global Statistics

Projected Workload Buffer Gets



Improvement Impact **24%** ↑  
 Regression Impact **-2%** ↓  
 Overall Impact **22%** ↑

SQL Statement Count



### Recommendations

Oracle offers two options to fix regressed SQL resulting from plan changes:

Use the better execution plan from SQL Trial 1 by creating SQL Plan Baselines.

[Create SQL Plan Baselines](#)

Explore alternate execution plans using SQL Tuning Advisor.

[Run SQL Tuning Advisor](#)

### Top 10 SQL Statements Based on Impact on Workload

SQL ID	Net Impact on Workload (%)	Buffer Gets		Net Impact on SQL (%)	% of Workload		Plan Changed
		10g_data	11g_data		10g_data	11g_data	
↑ <a href="#">g4dzf4ak4rus2</a>	12.000	20,318,458.000	13,502,097.000	33.550	35.780	30.670	Y
↑ <a href="#">gfacm5jr3rz9j</a>	11.990	6,990,541.000	180,401.000	97.420	12.310	0.410	Y
↓ <a href="#">2ny751aat2vd9</a>	-0.820	12,973,052.000	13,440,825.000	-3.610	22.850	30.530	Y
↓ <a href="#">c2fb0ug5p7d4p</a>	-0.750	12,740,524.000	13,165,998.000	-3.340	22.440	29.910	Y
↑ <a href="#">2wtqxbjz6u2by</a>	0.050	244,678.000	218,533.000	10.690	0.430	0.500	Y

# SPA Report

## Regressed SQL Statements

Regressed SQL Statements								
SQL ID	Net Impact on Workload (%)	Buffer Gets		Net Impact on SQL (%)	% of Workload		Plan Changed	
		10g_data	11g_data		10g_data	11g_data		
<a href="#">2ny751aat2vd9</a>	-0.820	12,973,052.000	13,440,825.000	-3.610	22.850	30.530	Y	
<a href="#">c2fb0u...</a>	-0.750	12,						

**SQL Details: 2ny751aat2vd9**

Parsing Schema **DWH\_TEST**      Execution Frequency **1**      (Show as SQL Tuning Advisor)

▶SQL Text

Single Execution Statistics

Execution Statistic Name	Net Impact on Workload (%)	Execution Statistic Collected		Net Impact on SQL (%)	% of Workload	
		10g_data	11g_data		10g_data	11g_data
↓ Elapsed Time	-4.340	70.519	89.593	-27.050	16.060	24.170
↓ Parse Time	-13.830	0.207	0.312	-50.720	27.270	32.470
↓ CPU Time	-5.700	64.704	85.188	-31.660	18.010	24.200
↓ Buffer Gets	-0.820	12,973,052.000	13,440,825.000	-3.610	22.850	30.530
↑ Optimizer Cost	0.170	982.000	658.000	32.990	0.530	0.360
↑ Disk Reads	10.900	7,011.000	5.000	99.930	10.810	1.850
↑ Direct Writes	10.950	6,968.000	0.000	100.000	10.950	0.000
↓ Rows Processed	0.000	111.000	111.000	0.000	0.000	0.000

**Plan Comparison**

**10g\_data**

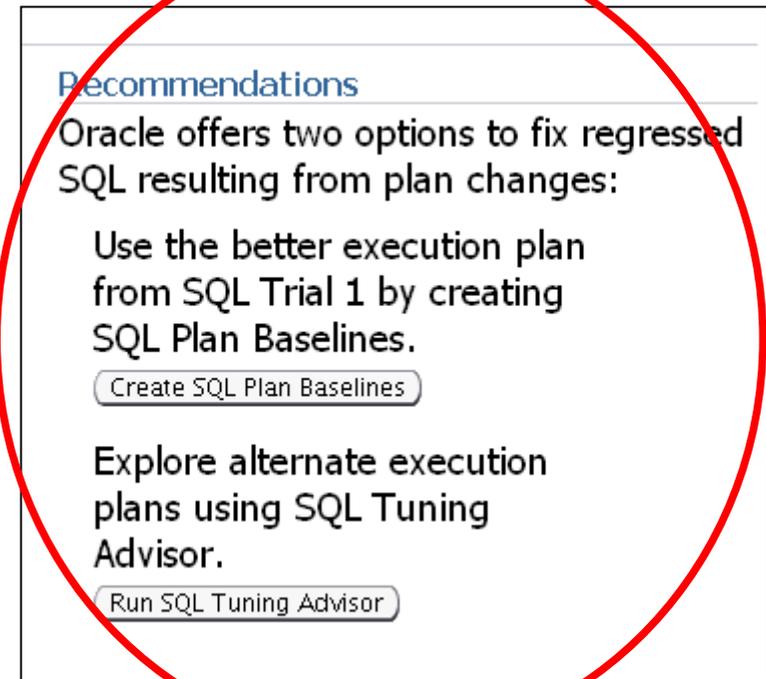
Plan Hash Value **393503022**

[Expand All](#) | [Collapse All](#)

Operation	Line ID	Object	Rows	Cost
▼ SELECT STATEMENT	0		1	967
▼ HASH	1		1	967
▼ TABLE ACCESS	2	<a href="#">FACT_PD_OUT_ITM_293</a>	1	966
▼ NESTED LOOPS	3		1	966
▼ MERGE JOIN	4		1	320
▼ SORT	5		90	315
▼ TABLE ACCESS	6	<a href="#">ADM_PG_FEATUREVALUE</a>	1	2
▼ NESTED LOOPS	7		90	314

# Отчет SPA – исправление регрессировавшего SQL

- **SQL Tuning Advisor:** Помогает улучшить план выполнения запроса
- Все ухудшившиеся SQL автоматически назначаются на улучшение
- **Создание SQL Plan Baselines:** Позволяет вернуться к старым, хорошим планам
- Для 10.2 STA выполнять на тестовой 10.2 БД
- Для 10.2 можно использовать Outlines



**Recommendations**

Oracle offers two options to fix regressed SQL resulting from plan changes:

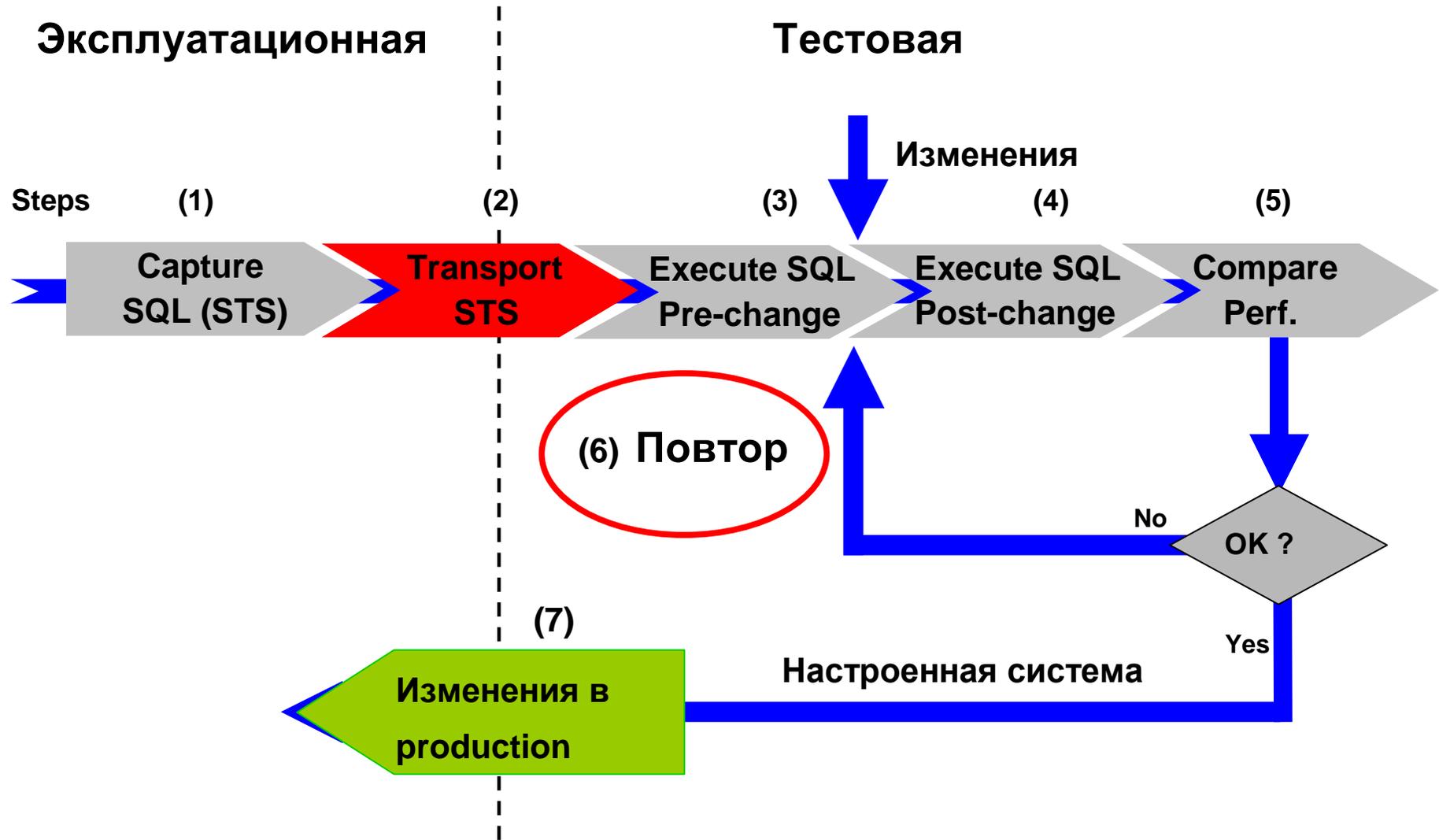
Use the better execution plan from SQL Trial 1 by creating SQL Plan Baselines.

[Create SQL Plan Baselines](#)

Explore alternate execution plans using SQL Tuning Advisor.

[Run SQL Tuning Advisor](#)

# SQL Performance Analyzer: Workflow



ORACLE

# Шаг 6: Повтор

- SPA workflow может быть повторен много раз и для различных изменений
- Сохраняется история изменений и сравнений

## Шаг 7: Изменение промышленной СУБД и ее настройка

- Выполните отлаженные изменения на пром системе
- Перенесите все проверенные настройки на пром систему
  - Export/Import SQL Plan Baselines
  - Export/Import SQL Profiles
  - Сделайте изменения в схеме (e.g., use DBMS\_REDEFINITION)
  - Соберите новую статистику для оптимизатора
  - Измените параметры

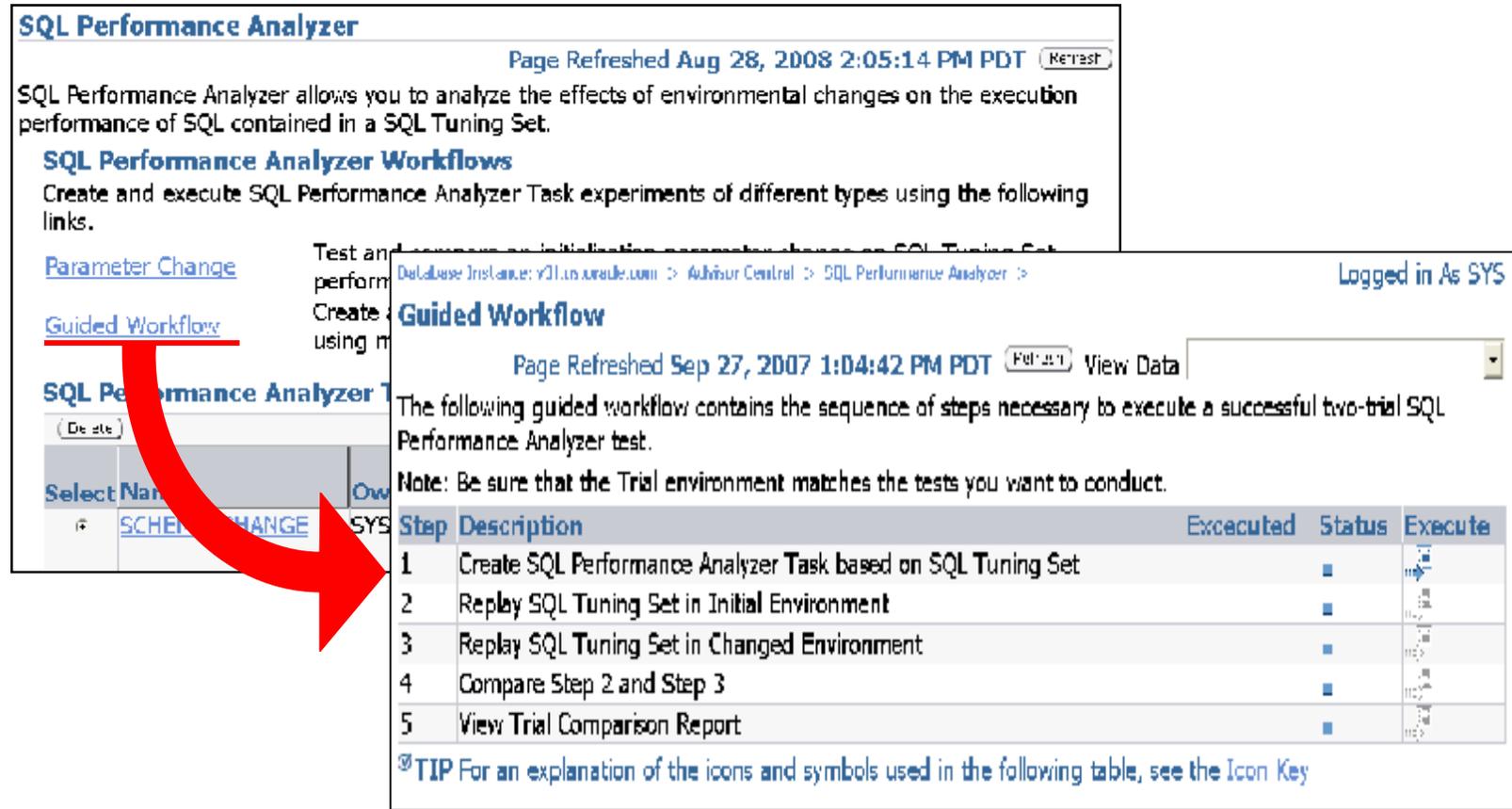
# Best Practices

- **SQL Workload Capture**
  - Используйте incremental STS capture, чтобы аккуратно захватить все важные SQL
- **Тестовая система**
  - Выполняйте SPA на тестовой системе, т к анализ загружает процессор
  - Ensure test system has similar configuration and comparable optimizer statistics as production
- **Сравнение производительности**
  - Для получения более надежных результатов используйте несколько различных метрик, например, elapsed time, CPU time, и т д., для сравнения pre- и post-change производительности
- **Исправление регрессии**
  - Используйте SQL Tuning Advisor и SQL Plan Baselines
- **Настройка промышленной СУБД**
  - Используйте тестирование на уровне схемы или отложенные режимы, чтоб снизить видимость Вашей работы
    - sqltune\_category, pending statistics, invisible indexes features
    - Maintenance window, resource throttling

# Улучшение Regressed SQL

- **Систематические проблемы**
  - Проверьте непроанализированные таблицы, PGA память, статистику, системную статистику
  - Читайте “Upgrading from Oracle x to y: What to expect from the Optimizer” на OTN
- **Для операторов, страдающих от изолированных проблем, используйте следующие приемы**
  - SQL Profiles: Создайте Profiles рекомендованные SQL Tuning Advisor (STA)
  - Stored Outlines: Если profile не были рекомендованы STA, тогда можно использовать Stored Outlines в 9i/10g для данных SQL операторов. Импортируйте stored outline в 10g.

# 2 Workflow SPA



**SQL Performance Analyzer**  
Page Refreshed Aug 28, 2008 2:05:14 PM PDT [Refresh](#)

SQL Performance Analyzer allows you to analyze the effects of environmental changes on the execution performance of SQL contained in a SQL Tuning Set.

**SQL Performance Analyzer Workflows**  
Create and execute SQL Performance Analyzer Task experiments of different types using the following links.

[Parameter Change](#)  
[Guided Workflow](#)

**SQL Performance Analyzer**  
(Delete)

Select Name	Owner
SCHEN CHANGE	SYS

Database Instance: v3111n.Oracle.com > Advisor Central > SQL Performance Analyzer > Logged in As SYS

**Guided Workflow**  
Page Refreshed Sep 27, 2007 1:04:42 PM PDT [Refresh](#) [View Data](#)

The following guided workflow contains the sequence of steps necessary to execute a successful two-trial SQL Performance Analyzer test.

Note: Be sure that the Trial environment matches the tests you want to conduct.

Step	Description	Executed	Status	Execute
1	Create SQL Performance Analyzer Task based on SQL Tuning Set		■	
2	Replay SQL Tuning Set in Initial Environment		■	
3	Replay SQL Tuning Set in Changed Environment		■	
4	Compare Step 2 and Step 3		■	
5	View Trial Comparison Report		■	

**TIP** For an explanation of the icons and symbols used in the following table, see the [Icon Key](#)

ORACLE

# Сценарии использования SPA при миграции

- Тестирование изменений в 11.g
- Тестирование Upgrade 10.2 à 11g
  
- Тестирование Upgrade 10.2.x à 10.2.y
- Тестирование Upgrade 9i/10.1 à 11g
- Тестирование Upgrade 9i/10.1 à 10.2

# Миграция из 10.2 в 11g

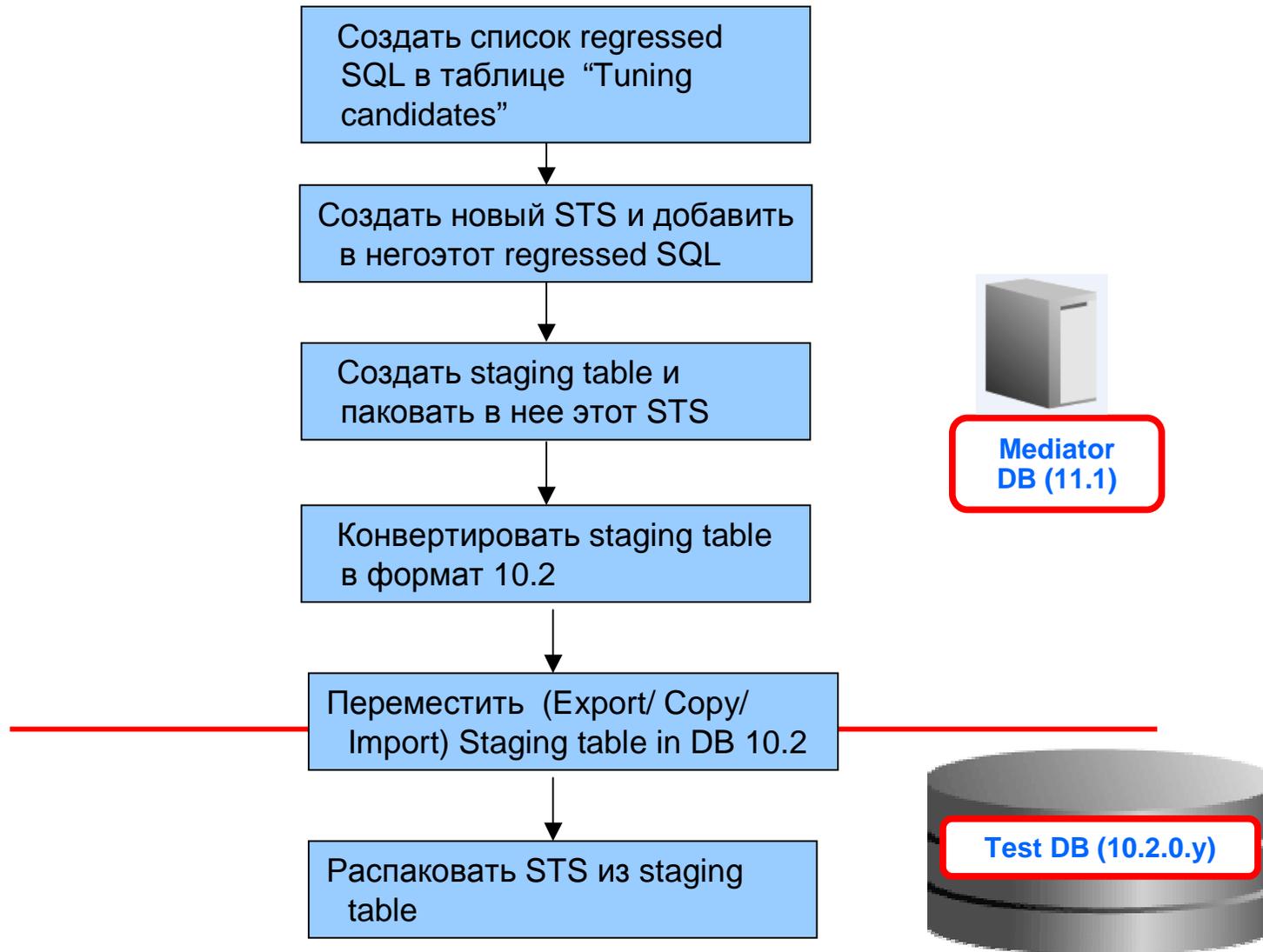
- Создать в 10.2 STS (из кэша, AWR, import)
- Создать тестовую копию 10.2
- Переместить в нее STS (OEM, scripts)
- Upgrade test DB в 11g (11.1.0.7)
- Создать Analysis Task (OEM)
- Выполнить STS удаленно в production 10.2 (patch)
- Выполнить STS локально в 11g
- Сравнить результат и использовать STA и SQL Plan Baseline для настройки regressed SQL

# Специфика миграции с 10.2.x на 10.2.y

- Можно мигрировать с/в Oracle 10.2.0.2/10.2.0.3/10.2.0.4
- Нужен посредник 11g для управления процессом и сравнения результатов - 11g SPA System (в 10g нет SPA)
- Посредник имеет маленькую БД
- Из посредника выполняется удаленное выполнение STS в 10.2 => нужен Database Link
- Patch для Oracle 10.2 зависит от ОС и версии СУБД (см metalink note 560977.1)
- Patch for Oracle 11.1.0.6 зависит от ОС и версии СУБД (или нужна 11.1.0.7)
- Если production DB - Oracle 10.2x надо:
  - Создать клон production database 10.2.x на тестовом узле
  - Создать STS в production database
  - Переместить этот STS в посредник 11.1
  - Создать remote pre-change trial для Oracle 10.2.x тестового узла
  - upgrade 10.2.x в 10.2.y
  - Продолжить традиционную работу с SPA
  - .....

ORACLE

# Перемещение regressed SQL в 10.2 с ПОМОЩЬЮ СКРИПТОВ



ORACLE

# Database Upgrade: 10.2.x -> 10.2.y

## Экспорт SQL profiles в production system

- Если STA создает STS profiles в тестовой 10.2, экспортируйте их, а затем импортируйте в НОВУЮ production system
- Используйте Expdp/Impdp для export/import stage table
- Распакуйте SQL profiles, импортированные в новую production 10.2.y, используя CLI

```
begin
  dbms_sqltune.create_stgtab_sqlprof(
    table_name => 'SQLPROF_TAB');

  -- pack all sql profiles, for export
  dbms_sqltune.pack_stgtab_sqlprof(
    profile_name => '%',
    staging_table_name => 'SQLPROF_TAB');
end;
```

```
begin
  -- pack all sql profiles, for export
  dbms_sqltune.unpack_stgtab_sqlprof(
    replace => TRUE,
    staging_table_name => 'SQLPROF_TAB');
end;
```

# Особенности перехода с 9i/10.1

- Там нет STS
- Данные для построения STS берем из Trace files
- Статистику pre-change берем из trace files
- Используем скрипты
- TWP на OTN
- Скрипты на OTN
- Для большей части работы используем OEM
- Сравнение в 11.1 SPA system, а настройка в тестовой 10.2 DB

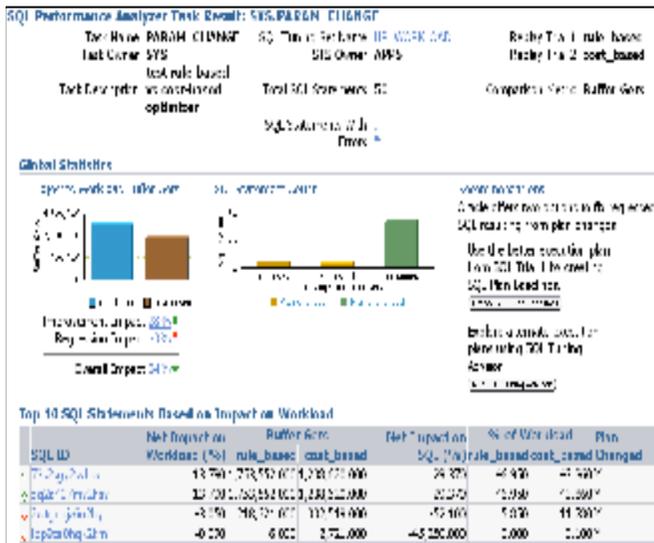
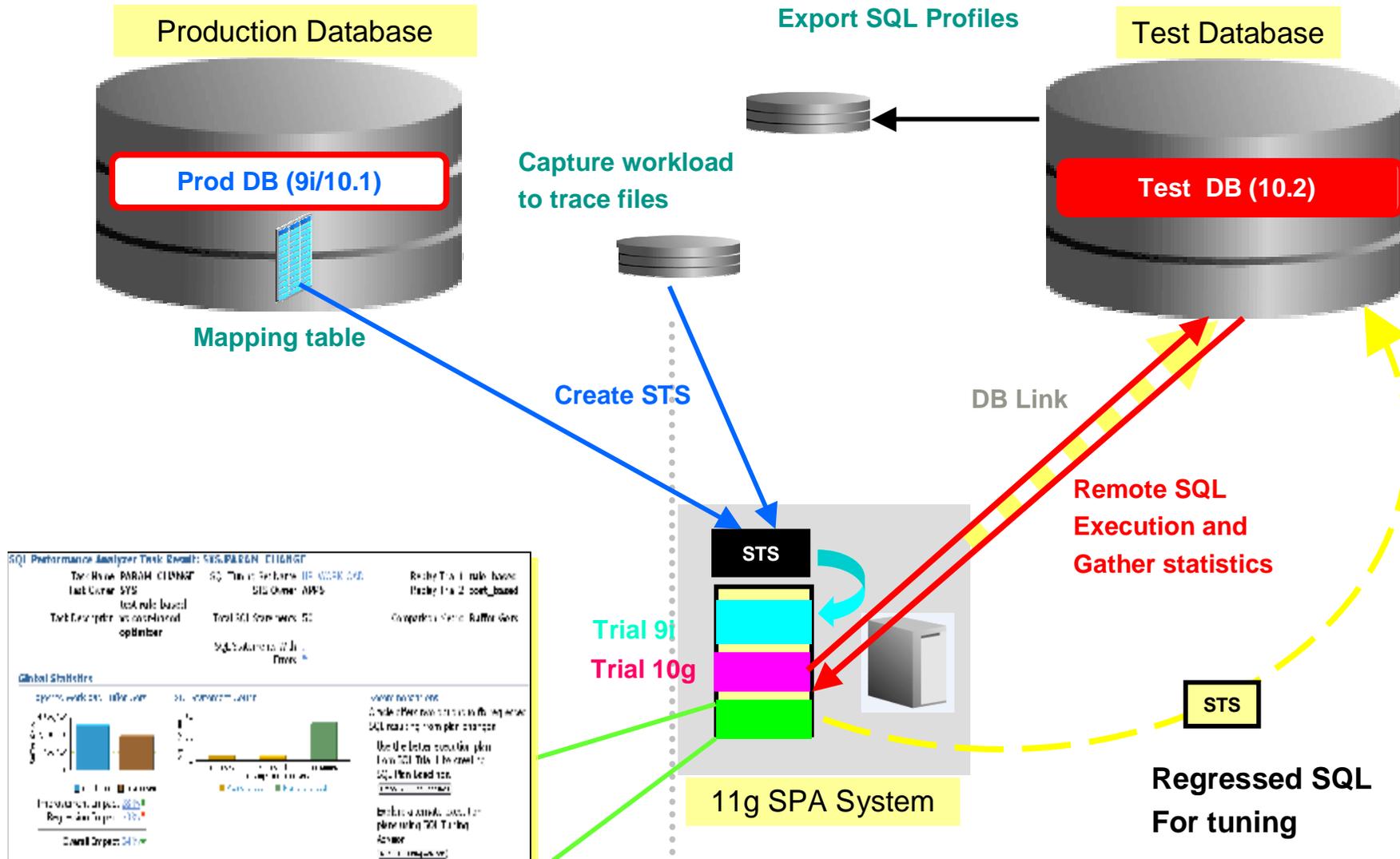
# Database Upgrade: с 9.2/10.1

## Захват SQL с помощью SQL Trace

- Выявите моменты интенсивной загрузки СУБД, такие как конец месяца, ежедневный пик и т.д.
- Захватывайте SQL trace по несколько сеансов за раз
- Используйте `dbms_support/dbms_monitor` package, он захватывает
  - bind value
  - Трассировку других сессий
- SQL trace настройка
  - `time_statistics=true`: Важно чтоб собрать статистику производительности
  - `user_dump_dest`
  - `max_dump_file_size`
  - `trace_file_identifier`
  - Накладные расходы: 10-15% для трассируемой сессии

ORACLE

# Database Upgrade: 9i/10.1 à 10.2.



ORACLE

# Демонстрация SQL Performance Analyser

**9.2.0.8 è 11.1.0.7**

**9.2.0.8 è 10.2.0.4**

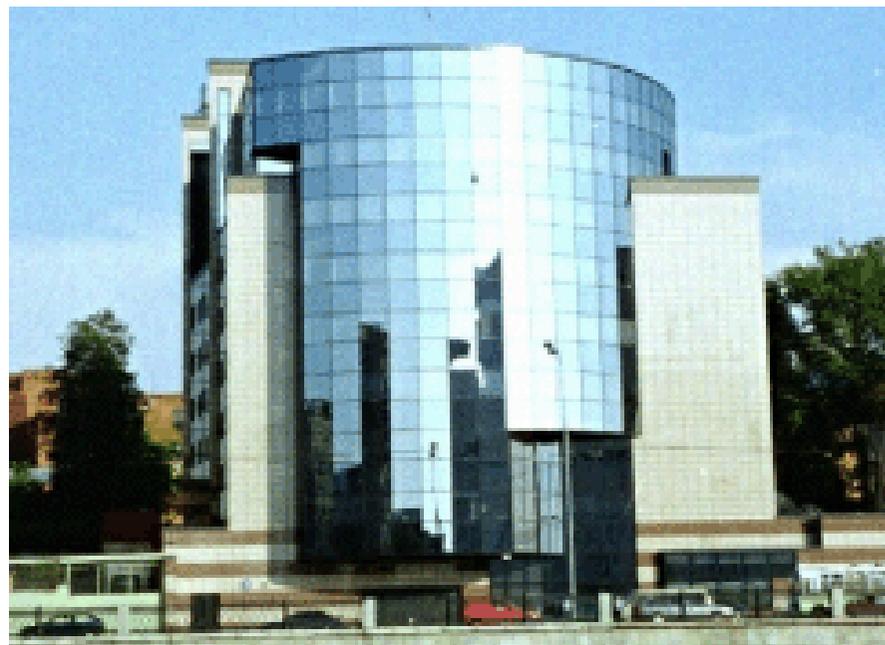
ORACLE

# Заключение

- RAC нужен не только при Upgrade, Вы будете использовать его постоянно, деньги не пропадут
- Он полезен:
  - При изменении HW и ОС
  - При изменении параметров
  - При построении индексов, MV и т д
  - При выполнении добрых советов от коллег, STA, экспертов и т д
  - При миграции на RAC
  - При использовании новых возможностей Oracle (например, автоматический сбор статистики)
  - И т д

# Как нас найти...

- **Телефон в Москве**  
**+ (7 495) 641-14-00**
- **[www.oracle.com/ru](http://www.oracle.com/ru)**
- **[www.oracle.ru](http://www.oracle.ru)**
- **Email**  
**[Mark.Rivkin@oracle.com](mailto:Mark.Rivkin@oracle.com)**

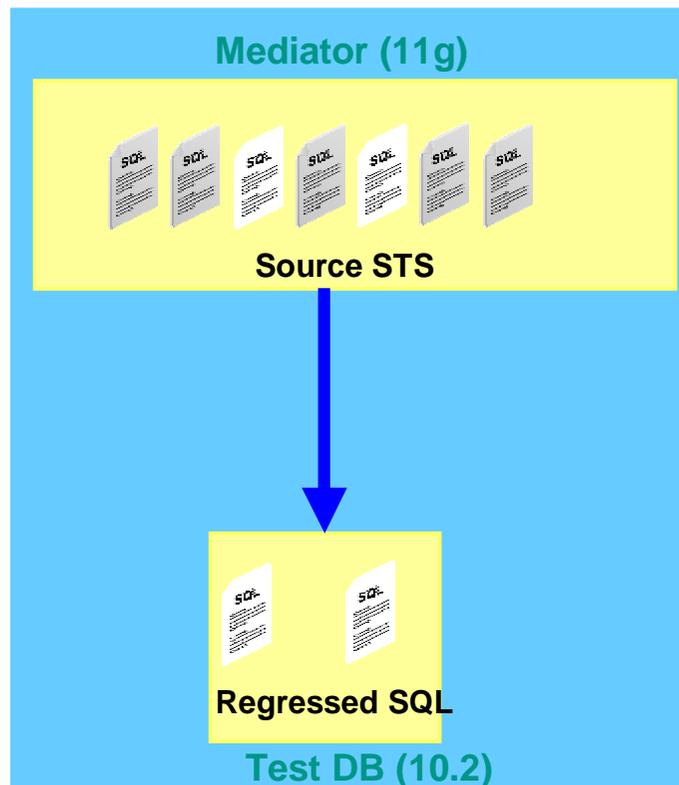


ORACLE

# Database Upgrade: 10.2.x à 10.2.y

## Step: Convert full STS (11g) to regressed STS (10.2.x)

- Create new STS (regressed STS) in Oracle 11.1 and move regressed SQL in this STS



```
1 Create a table tuning_candidates with a single column, sql_id,  
containing the list of SQL_IDs for regressed SQL
```

```
2 dbms_sqltune.create_sqlset('sqs_to_tune');
```

```
declare
```

```
mycur dbms_sqltune.sqlset_cursor;
```

```
basf varchar2(32767);
```

```
begin
```

```
-- put the subset of sqs we want to tune into another STS
```

```
basf := 'sql_id in (select sql_id from tuning_candidates)'
```

```
open mycur for
```

```
select value(p)
```

```
from table(dbms_sqltune.select_sqlset('prod_wkld',
```

```
basf, null, null, null, null, 1, null,'BASIC')) p;
```

```
dbms_sqltune.load_sqlset('sqs_to_tune', mycur);
```

```
close mycur;
```

```
end;
```

```
/
```

ORACLE

# Export/Import STS (11.g -> 10.2)

- Pack 11g STS and convert it in 10.2 staging table

```
begin
-- pack the sts into a staging table
dbms_sqltune.create_stgtab_sqlset('STGTAB');
dbms_sqltune.pack_stgtab_sqlset(
  sqlset_name => 'sqs_to_tune',
  staging_table_name => 'STGTAB');

exec dbms_sqltune.create_stgtab_sqlset('STGTAB_10R2');

insert into stgtab_10R2
(name, owner, description, sql_id, force_matching_signature, sql_text, parsing_schema_name, bind_data, bind_list, module, action, elapsed_time,
cpu_time, buffer_gets, disk_reads, direct_writes, rows_processed, fetches, executions, end_of_fetch_count, optimizer_cost, optimizer_env,
priority, command_type, first_load_time, stat_period, active_stat_period, other, plan_hash_value, spare2)
select name, owner, description, sql_id, force_matching_signature, sql_text, parsing_schema_name, bind_data, bind_list, module, action,
elapsed_time, cpu_time, buffer_gets, disk_reads, direct_writes, rows_processed, fetches, executions, end_of_fetch_count, optimizer_cost, null,
priority, command_type, first_load_time, stat_period, active_stat_period, other, plan_hash_value, 1
from stgtab;
commit;
end;
/
```

ORACLE

# Export/Import/unpack STS from 11g to 10.2

- Export stage table
  - expdp user/pwd tables=STGTAB\_10R2  
version=10.2.0.2 directory=WORK\_DIR  
dumpfile=stgtab10R2.dmp
- Copy dump file on test system 10.2, using FTP/Copy/etc
- Import and unpack stage table (you can use CLI or GUI OEM)

```
impdp user/pwd directory=WORK_DIR dumpfile=stgtab10R2.dmp
begin
  dbms_sqltune.unpack_stgtab_sqlset(
    sqlset_name => '%',
    sqlset_owner => '%',
    staging_table_name => 'STGTAB',
    replace => TRUE);
end;
/
```

# After first tuning regressed SQL in test DB

