# 大型业务系统数据库升级到11gR2实践

# 个人介绍

杨志洪：

- 10年ORACLE服务经验
- 服务过的行业：电信、联通、移动、银行、保险、证券、基金、制造、政府
- 服务于国内最大的数据库第三方服务厂商**上海新炬**

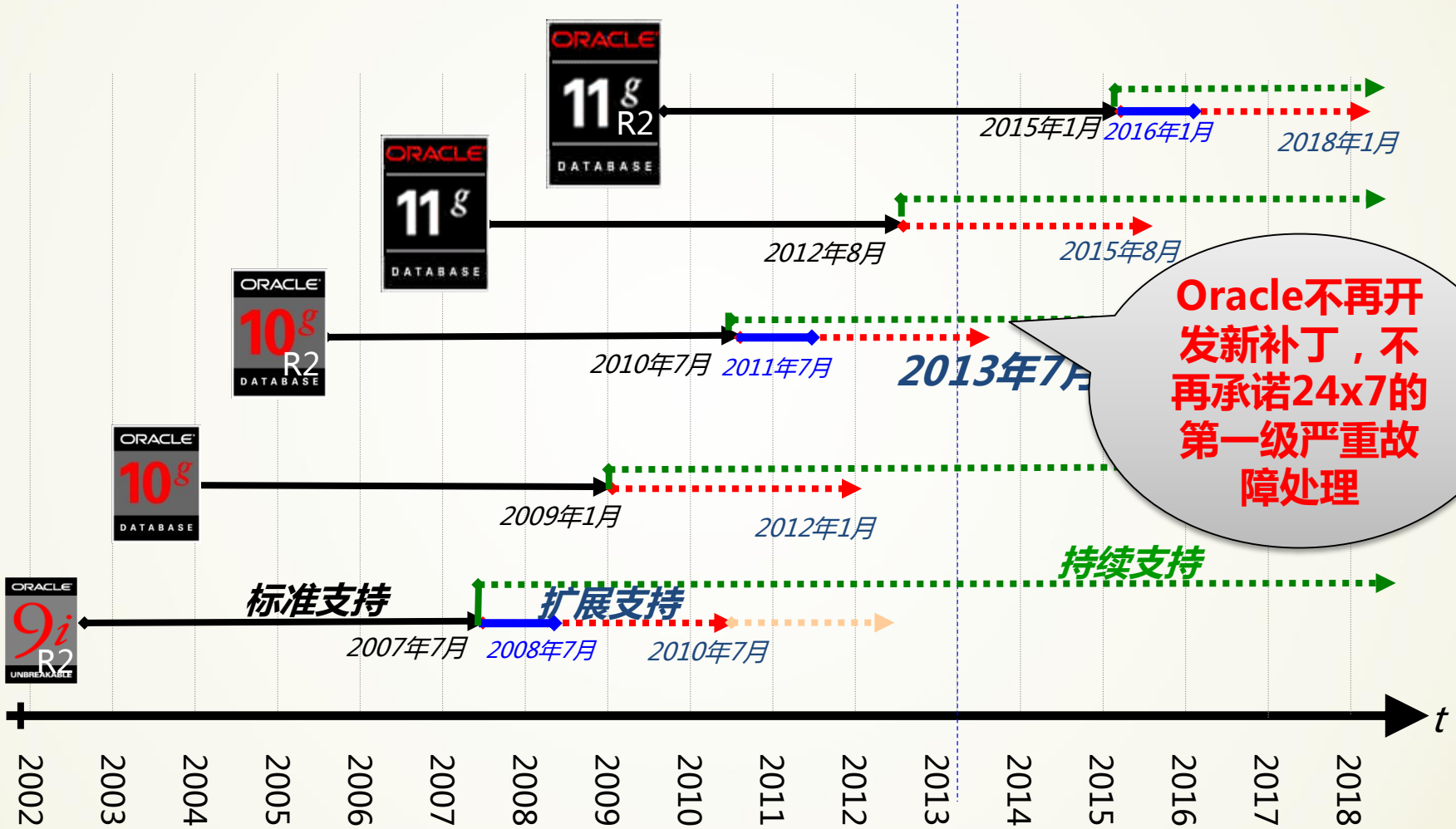
联系方式：

- yangzhihong@shsnc.com
- 18621566151http://weibo.com/boypoo

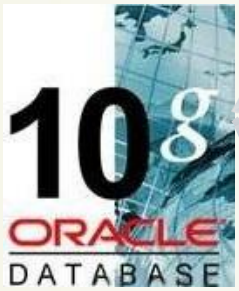
# 分享大纲

- 升级到11gR2的必要性
- 升级到11gR2的技术路径
- 大型业务系统升级到11gR2的综合因素
- 升级后的系统稳定性保障方案

# ORACLE数据库版本生命周期策略，推动升级步伐



# 上海新炬已升级完多个客户多个核心系统并稳定运行




1. ➢目前上海新炬截止目前，总共运维ORACLE数据库近2000套，目前新增系统均首选11G，已有核心系统大部份已经完成或计划进行升级

2. ➢目前上海新炬运维的核心生产系统11G版本稳定性及可维护性方面均优于10G.在10G版本影响较大的一些BUG或缺陷在新版中基本得到修复

3. ➢运维的核心生产系统多次遇到全球首发的BUG，由于ORACLE停止SUPPORT，后通过ORACLE协调美国研发部门特事特办解决，费尽周折

4. ➢当前许多核心系统数据库已安装了多达几十甚至几百的Patch，这些用以规避bug的临时补丁已严重威胁系统的稳定性

某用户核心业务系统在10G版本因为BUG打的补丁已达到**118个**，11G环境运行近2年修复BUG的补丁只有**10个**

# 充分利用ORACLE DB 11g新特性提升系统运行性能与稳定性

## 可管理性

- 计划管理
- 自动诊断知识库
- 事件打包
- 基本特性补丁
- 访问建议器
- 资源管理器
- ADDM
- **更多的表分区类型**
- **在线应用升级**
- **完整的云管理**

## 开发及性能

- ORACLE SECUREFILES
- OLTP 表压缩
- 内存并行执行
- 更多的分区选项支持
- 结果集缓存
- 只读表
- 增强压缩技术
- 自动SQL优化
- 自动内存优化
- **数据库重演**
- **SQL重演**

## 高可用性及安全

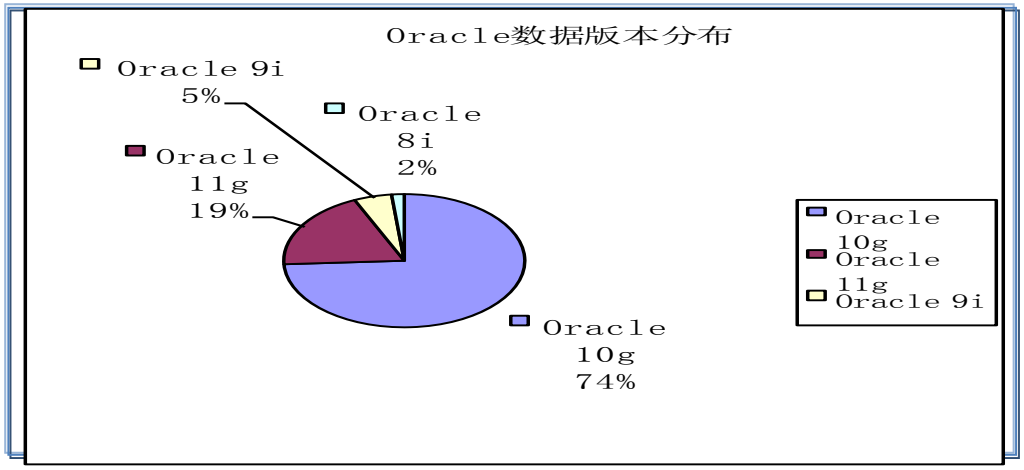
- **ACTIVE DATAGUARD**
- 数据库服务器池
- FLASHBACK DATA ARCHIVE（全面回忆数据变化）
- 数据库防火墙和 Database vault
- Audit vault
- 备份恢复增强

相对于10G，11G版本有非常多的新特性及在性能方面有很大的提升，可以更好的满足用户日益增长的业务需求

# 分享大纲

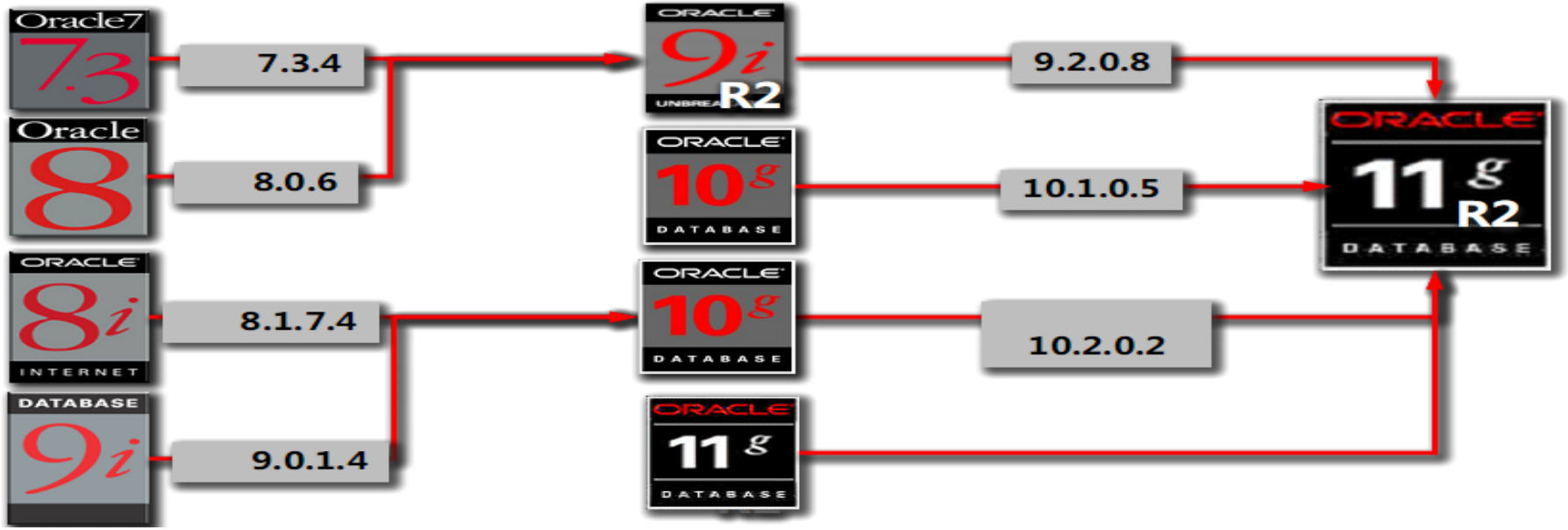
- 升级到11gR2的必要性
- 升级到11gR2的技术路径
- 大型业务系统升级到11gR2的综合因素
- 升级后的系统稳定性保障方案

# 用户需求：多个业务系统/多版本/多平台的升级




- 用户大多有数十甚至上百套ORACLE数据库产品
- 可能涉及8i/9i/10g/11gr1等多个需要升级的版本
- 如何规划升级路径及选择合适的升级方式非常关键
- 各系统如何排定升级顺序，如何组织硬件、开发、测试资源

## 升级到 Oracle Database 11.2

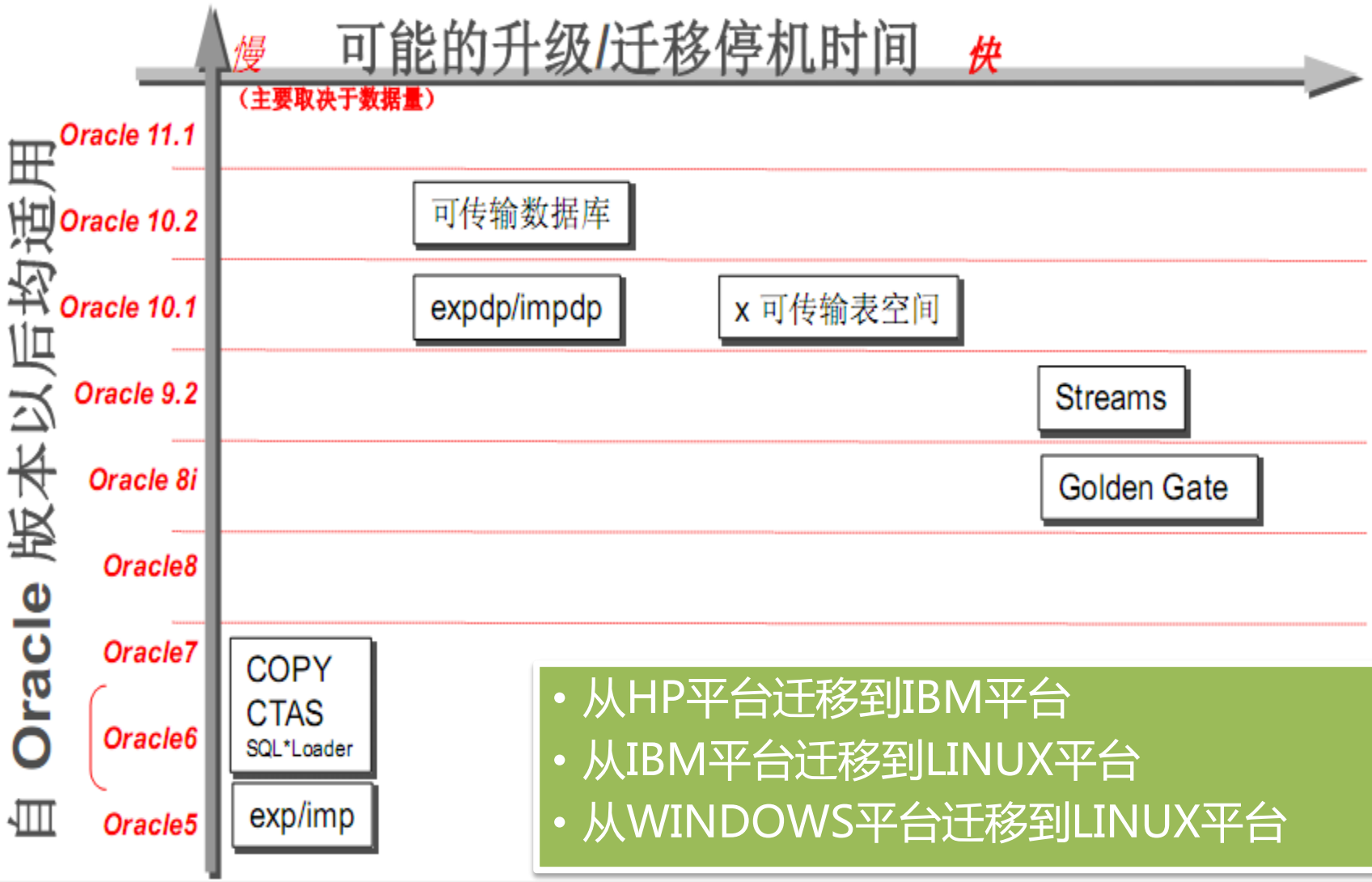


# 我们需要考虑的问题远不止这些

- 您将升级多少个数据库？
- 您将更换操作系统吗？
- 您将迁移到新硬件吗？
- 您可以承受多长时间的停机？
- 您的回退要求是什么？
- 您如何测试应用程序？
- 您将更改字符集吗？
- 您将启用新特性吗？

- 导入导出
- 直接升级
- 可传输表空间
- CTAS
- STREAM/GOLDENGATE
- .....

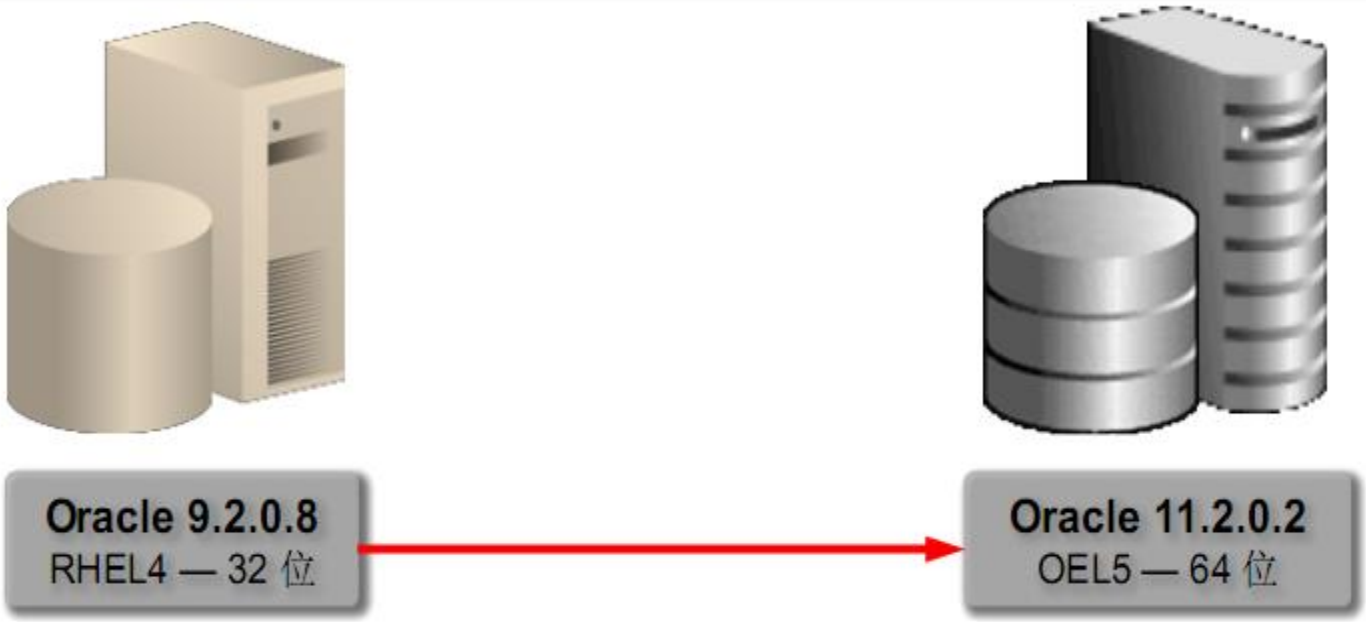
# 场景示例1：变更为新的操作系统和硬件



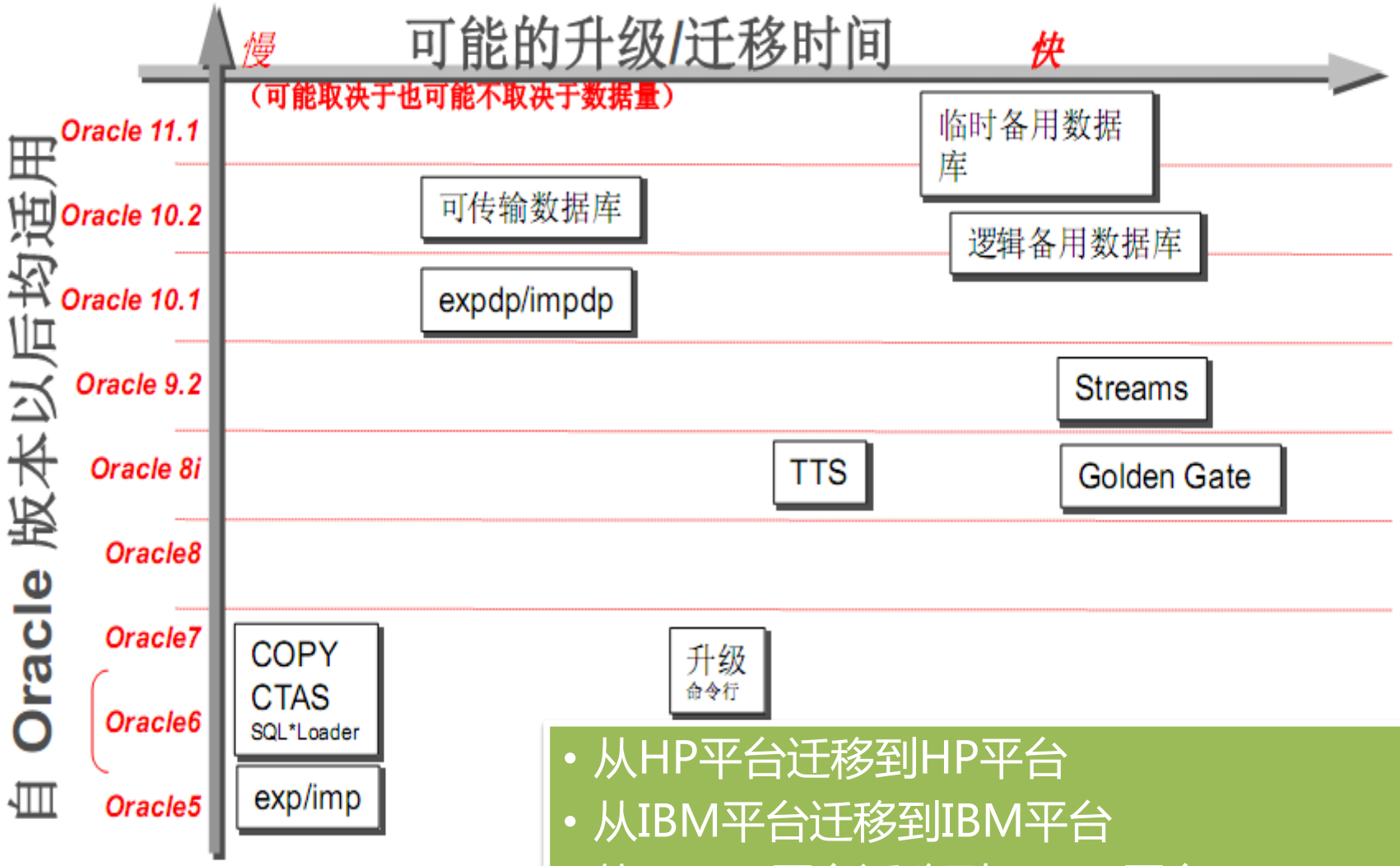

- 从HP平台迁移到IBM平台
- 从IBM平台迁移到LINUX平台
- 从WINDOWS平台迁移到LINUX平台

# 场景案例1：变更为新的操作系统和硬件

| 事实与说明 |
| --- |
| 1. 硬件迁移 |
| 2. 数据库从 9.2 升级到 11.2（32 位到 64 位） |
| 3. 容许的最长停机时间：24 小时 |
| 4. 数据库大小：1TB |
| 5. 可行方法： |
| exp/imp、RMAN 恢复和升级、Golden Gate 或 Streams、卸载/加载 |



# 场景示例2：保留操作系统，升级和更改硬件




- 从HP平台迁移到HP平台
- 从IBM平台迁移到IBM平台
- 从LINUX平台迁移到LINUX平台

# 场景案例2：存储方式变更为ASM

## 升级条件说明

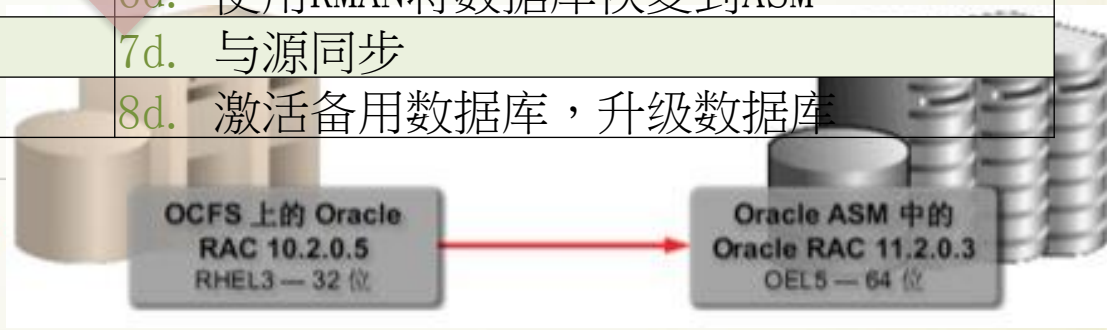
1、升级到Oracle 11.2 并将数据库迁移到ASM

2、硬件迁移到新集群

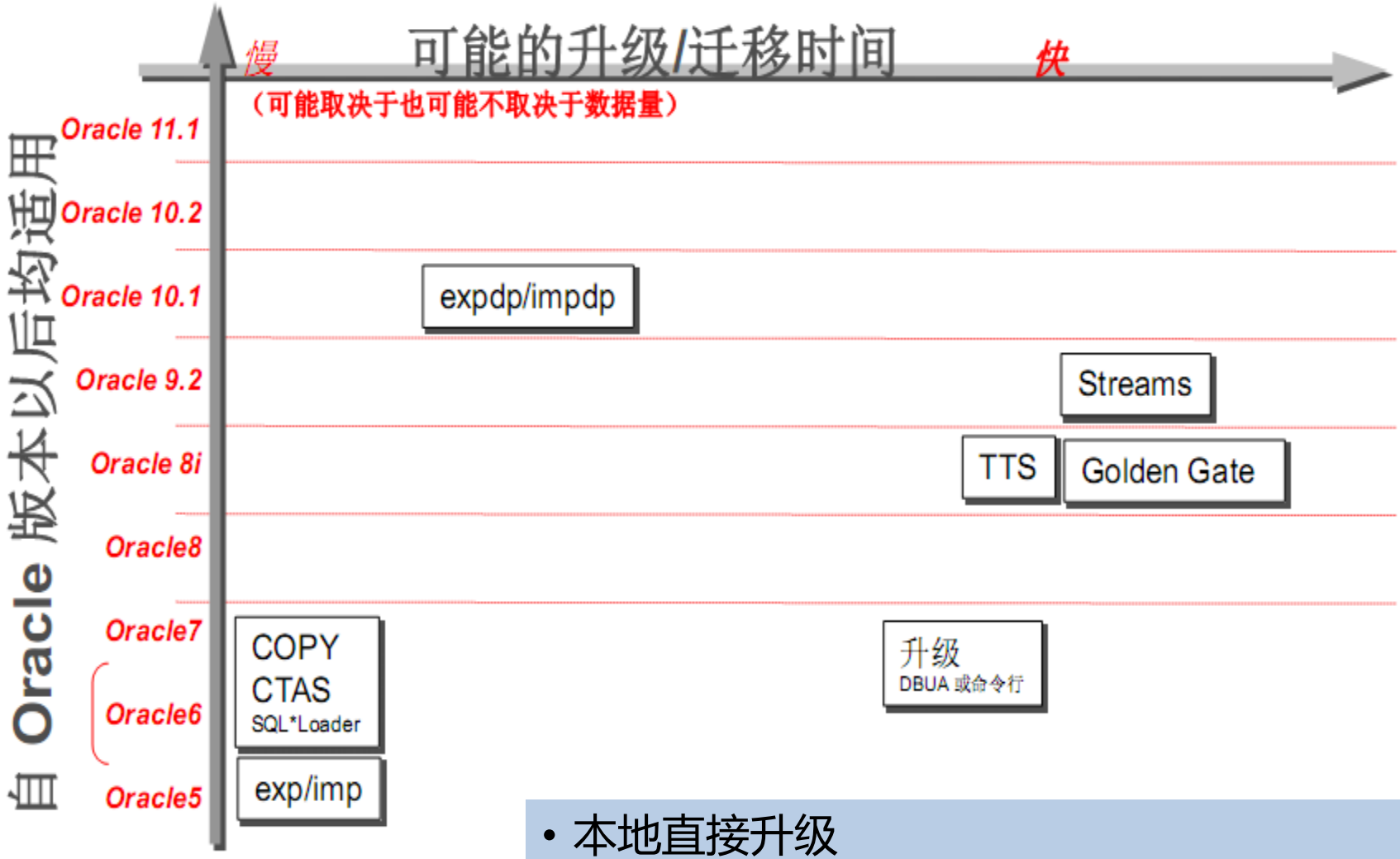
3、数据库大小：5TB

4、可容许的最长停机时间：4小时

| 源 | 目标 |
| --- | --- |
| 1s. 进行完整联机备份 | 1d. 安装GI 11.2.0.3 |
| 2s. 将备份复制到目标服务器 | 2d. 安装Oracle Database 11.2.0.3 |
| | 3d. 安装Oracle10.2+10.2.0.5 |
| | 4d. 配置并启动ASM实例 |
| 5s. 为备用数据库准备init.ora /tnsnames.ora | 5d. 为备用数据库准备init.ora /tnsnames.ora |
| | 6d. 使用RMAN将数据库恢复到ASM |
| 7s. 启动日志传输 | 7d. 与源同步 |
| 8s. 停止应用程序 | 8d. 激活备用数据库，升级数据库 |



# 场景示例3：保留硬件,仅升级数据库




- 本地直接升级

# 分享大纲

- 升级到11gR2的必要性
- 升级到11gR2的技术路径
- 大型业务系统升级到11gR2的综合因素
- 升级后的系统稳定性保障方案

# 升级前明确需要做的工作及组建团队

- **调研分析**
  - 接口
  - 应用连接方式
  - 硬件环境
  - 应用代码改造
  - 应急方式
  - 运维
- **方案制定**
  - 升级路径
  - 升级方法
  - 回退方案
  - 应急方案
  - 接口应用调整方案
- **升级测试**
  - 硬件环境测试
  - 软件升级测试
  - 数据升级测试
  - 应用功能测试
  - 应用性能测试
  - 联调测试
- **正式升级**
  - 升级前环境准备
  - 软件升级
  - 参数调整
  - 应用验证
  - 备份
  - 容灾

**➢项目管理**

•整体负责升级工作

•协调相关开发、软硬件资源

•组织相关人员协同工作

**➢环境小组**

•软硬件环境准备

•网络环境准备

**➢应用测试小组**

➢功能测试

➢性能测试

➢接口测试

**➢升级小组**

➢调研分析

➢升级方案制定

➢升级测试

➢正式升级

**➢监控小组**

➢业务监控

# 数据库层面需要考虑的问题

- ORACLE11g R2数据库特性变化
- 数据库补丁评估
- 禁用的10g/9i参数
- 过期/废除的参数
- 特殊的数据库参数
- 升级测试、功能测试、性能测试
- 11g新特性启用评估
  - 自动任务
  - 自适应游标共享
  - OLTP表压缩
  - SCAN IP
  - runcate段清除
  - 闪回数据归档
  - 新分区技术
  - ADG使用
- 客户端服务端兼容性列表
- 11g GI集群软件OCR、VOTING DISK存放位置选择
- 数据库升级方式的考虑
- 统计信息的收集
- 历史执行计划的保存
- 11gR2 GI/RDBMS安装准备
  - 硬件要求
  - 网络要求
  - 软件要求
  - 数据库版本要求
  - 介质要求
  - 用户和组
  - ORACLE软件安装目录
- 往次升级遇到的问题总结
- 版本兼容性考虑

# 开始升级前的准备工作：数据库对象状态确认

- 检查/处理无效对象
  - SQL> select unique OBJECT_NAME, OBJECT_TYPE, OWNER from DBA_OBJECTS where STATUS= ‘INVALID’;
  - utlrp.sql 确认并移除无用对象
- 检查/处理无效组件
  - SQL> select * from DBA_REGISTRY where STATUS<>’VALID’;
- 移除过期参数\Event参数
  - #_always_semi_join=off
  - #_unnest_subquery=false
  - ……
- 清空垃圾回收站和aud$表
  - SQL> purge recyclebin;
  - SQL> truncate table sys.aud$;

# 建议了解的初始化参数

- **1. OPTIMIZER_USE_INVISIBLE_INDEXES**
  - 11gR1版本中引入,启用或禁用不可见索引的使用，默认值为FALSE.
- **2. OPTIMIZER_USE_PENDING_STATISTICS**
  - 11gR1版本中引入,决定编译sql语句时优化器是否使用pending的统计信息,默认值为FALSE.
- **3. OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES**
  - 11gR1版本中引入,在生成sql执行计划基线是，启用或禁用自动识别重复的sql语句 ,默认为FALSE.
- **4. OPTIMIZER_USE_SQL_PLAN_BASELINES**
  - 11gR1版本中引入, 启用或禁用使用SMB(sql Management Base)中sql执行计划基线，当启用时，优化器将会为正在编译的sql语句查找基线，如果从SMB中找到一条记录，那么优化器将会计算每条基线执行计划的成本，并会挑出其中成本最低的一个，默认值为TRUE.
- **5. deferred_segment_creation**
  - **11gR2版本中引入,**在使用exp/expdp导出时可能遇到错误.
- **6. optimizer_secure_view_merging**
  - **10gR2引入，防止复杂视图结构钟出现优化器问题，生成过多执行计划。建议设置为FALSE.**
- **7. memory_target**
  - **11gR1版本中引入,内存自动管理.**
- **8. _gc_policy_time**
  - **11g引入，DRM特性.建议关闭，设置为0.**
- **9. session_cached_cursors**
  - **自10.2.0.4后，定义该参数确保pl/sql游标缓存，基础建议为300。查询v$sysstat检验游标命中率设置**

# 分享大纲

- 升级到11gR2的必要性
- 升级到11gR2的技术路径
- 大型业务系统升级到11gR2的综合因素
- 升级后的系统稳定性保障方案

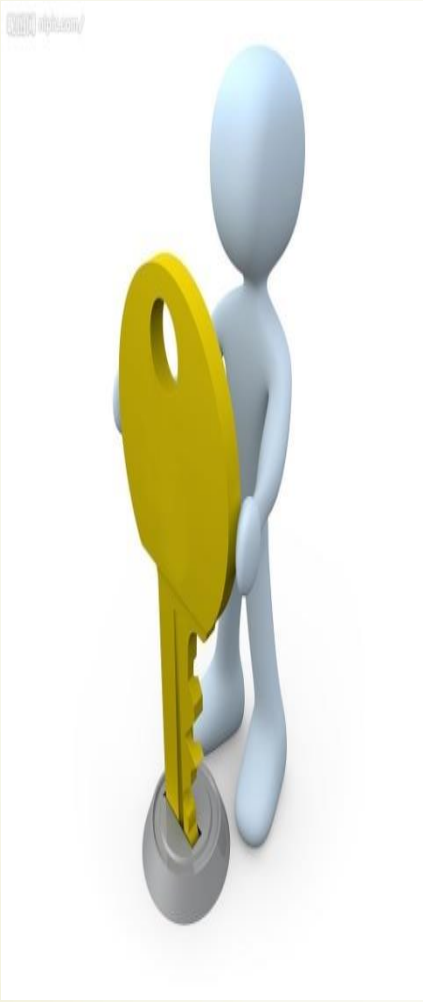
# 数据库升级，DBA最担心的问题

我们的性能测试准确吗？能不能反映真实负载？

升级后系统性能不会变差吧？

性能变差了怎么办吧？

# 我们的应对之策



- 90 percent of the time the default sample works
- 9 percent of the time a larger sample works
- 1 percent of the time the sample size is irrelevant

Andrew Holdsworth
Senior Director, Real World Performance, Oracle

传统方法＋11g新方法

# 传统方法

- 优化器 - 阻止执行计划发生变化:
  - 使用基于规则的优化器模式(RBO,expdp性能有影响)
  - 设置老的optimizer_features_enabled参数
  - 使用存储大纲(Stored Outlines)
  - 使用提示(Hint)
  - 重写SQL 语句
  - 设置特定的优化器参数
  - 搜集保存统计信息，导入并固定统计信息

# 结合11g新方法确保99%的性能问题

# 用SQL计划管理（SPM）来帮助稳定系统性能

SQL计划管理通过3个步骤来实现可演进的SQL计划稳定功能：

- 捕获
  - **设置OPTIMIZER_CAPTURE_SQL_PLAN_BASELINE=TRUE**
  - 创建一个代表了可信任的执行计划的基准线
  - 基准信息保存在SYSAUX表空间
- 选择
  - 只使用已接受的SQL执行计划
  - 新的执行计划记录在执行计划历史中
- 演化
  - 评估所有未经验证的执行计划
  - 根据评估结果接受或者拒绝未经验证的执行计划
  - 用户可手动接受或者拒绝执行计划

# 数据库升级中使用SQL计划管理

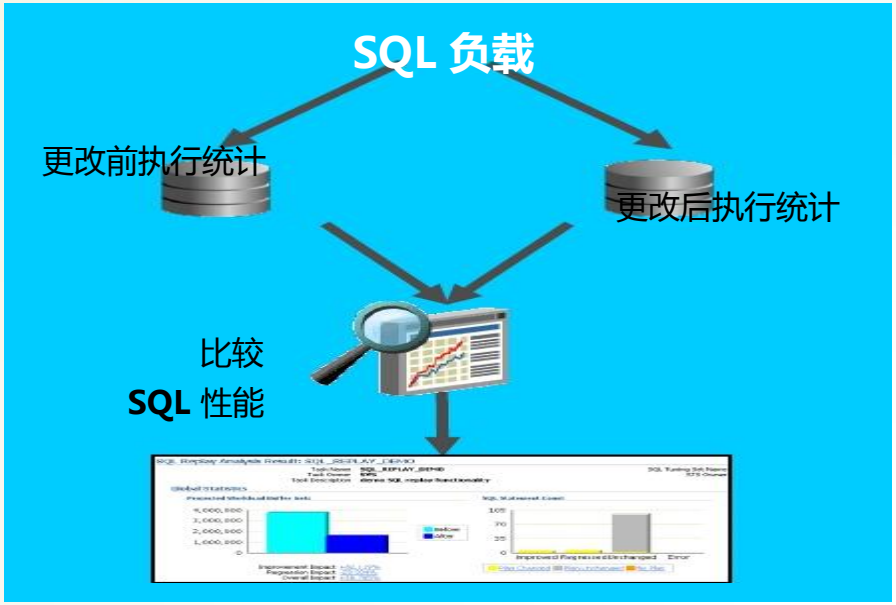
- 通过SQL 优化集中转
  - 10g中创建SQL优化集
  - 11g中从SQL优化集创建SQL计划基准
- 通过存储大纲中转
  - 10g中创建存储大纲
  - 11g中从存储大纲迁移到SQL计划基准
- 通过10g的优化器中转
  - 升级后11g中设置优化器为10g模式
  - 启用自动获取模式
  - 试运行一段时间后，再改优化器为11g模式
- 通过11g测试环境中转
  - 测试环境优化并抓取好的SQL
  - 通过中间表导入/导出到生产环境

# 数据库重放(DB Replay）帮助稳定系统性能

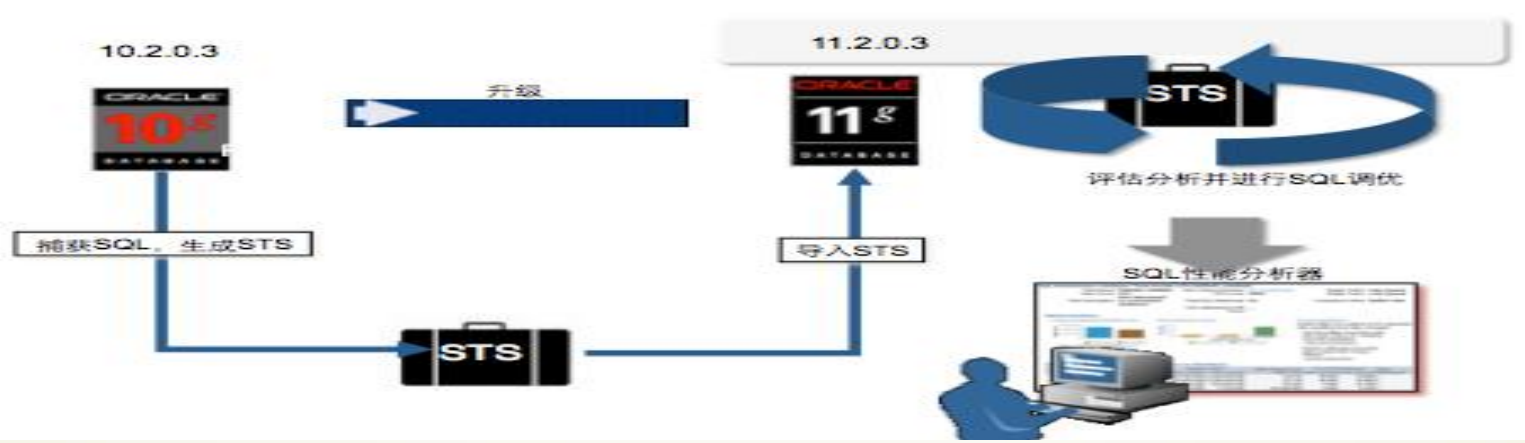
**在测试环境中重放实际生产数据库的负载**
**在实际对生产环境作改动之前识别，分析和解决潜在的不稳定因素**

- 在生产库中记录
  - 记录完整的生产库的负载，包括实际负荷，时间和并发情况
  - 将生成的负载记录文件转移到测试环境
- 在测试环境中重放
  - 首先在测试环境中作出计划要做的改动
  - 完全按照生产库的负载情况进行重放
  - 严格的事务提交顺序
- 分析和报告
  - 错误
  - 数据差异
  - 性能差异

# SQL性能分析器(SPA)帮助稳定系统性能




- 提供了快速假设分析以显示系统在不同设置时的表现
- 提供对个别 SQL 的细粒度性能分析
- 能够在性能问题影响最终用户前将其识别并解决



# SQLT:用SQLT来解决部分“遗漏的”SQL问题

作者：Carlos Sierra  Consulting Technical Advisor  Center of

下载：MOS 215187.1

用途：

1. 收集单条SQL语句的信息,包括: SQL文本/执行计划(explain)/真实执行计划(row source operation) 运行时的信息(时间,记录数等等)/10046/10053 trace
2. 收集影响optimizer的统计信息和可能会影响optimizer计算的参数;
3. 自动调用SQL Tuning Advisor来生成优化报告;
4. 自动导出相关数据以方便客户/Oracle Support创建test case来重现客户的问题,加速 问题的处理;
5. 自动生成SQL Profile的脚本,来固定SQL的执行计划。

```
SQL>  @coe.sql   5bhps37ha4k56        2525682227
......
SQL>-- generates script that creates sql profile in target system:
SQL>SET ECHO OFF;
coe_xfr_sql_profile_5bhps37ha4k56_2525682227.sql.
```

```
force_match => FALSE /* TRUE:FORCE (match even when different literals in
```

# 剩下的1%性能问题怎么办？扩展统计信息来帮忙

```
SQL> exec dbms_stats.gather_table_stats(null,'customers', -
>           method_opt => 'for all columns size skewonly');

PL/SQL procedure successfully completed.
```

```
-------------------------------------------------------------------------------------
| Id  | Operation                    | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |           |       |       |   278 (100)|          |
|   1 |  SORT AGGREGATE              |           |     1 |    16 |            |          |
|*  2 |   TABLE ACCESS STORAGE FULL| CUSTOMERS |  1090 | 17440 |   278   (1)| 00:00:01 |
```

```
SQL> SELECT DBMS_STATS.CREATE_EXTENDED_STATS(null,'customers', '(country_id, cust_state_province)')
  2  FROM   dual;
```

```
-------------------------------------------------------------------------------------
| Id  | Operation                    | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |           |       |       |   278 (100)|          |
|   1 |  SORT AGGREGATE              |           |     1 |    16 |            |          |
|*  2 |   TABLE ACCESS STORAGE FULL| CUSTOMERS |  3531 | 56496 |   278   (1)| 00:00:01 |
-------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - storage(("CUST_STATE_PROVINCE"='CA' AND "COUNTRY_ID"=52790))
       filter(("CUST_STATE_PROVINCE"='CA' AND "COUNTRY_ID"=52790))
```

# 欢迎莅临

# 2013中国数据库技术大会