

ORACLE®

ORACLE®

達人に聞く！データベースアップグレード成功の極意

Oracle Database
安定運用のためのベスト・プラクティス

パッチ適用とアップグレード

日本オラクル株式会社
テクノロジー製品事業統括本部
阿部 拓也



以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント(確約)するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

パッチ適用およびアップグレードの 計画指針

運用フェーズでのベスト・プラクティス

なぜパッチを適用しなければならないか？

安定稼働の要件とは？

「計画外で停止しない」

「パフォーマンスが劣化しない」

「データが破損・漏洩しない」

「不具合が発生しない」

- パッチ適用を運用に組み込むことで、結果不正を始めとする重大な問題に予め対応することができ、システムの安定性を向上させることができます
 - 問題発生前にメンテナンスを行っておくことで、パッチを適用せずに発生した問題に対応する場合と比較して原因調査にかかる時間やコストを少なく抑えることができます
- 最新のセキュリティパッチの適用はシステムやサービス、データを攻撃から守るために必須です
 - セキュリティパッチがリリースされると、セキュリティに関する問題や研究内容も同時に世界に向けて発信されますので、パッチを適用しないということは攻撃者にとって明らかな問題を放置することになります

運用フェーズでのベスト・プラクティス

安定稼働のためのパッチ適用とアップグレードの、ベスト・プラクティス・メッセージ

1 年に2～4回、PSU (BP) を適用する

- 実行計画、設定変更の必要な修正は含まれない（アプリテスト不要）
- 広く該当するまたは致命的な不具合修正と、最新セキュリティパッチを含む

2 ライフサイクルにアップグレードを組み込む

- PSU を適用し続けるためにはパッチ提供期間中のPSRに保つ必要がある
- 製品ライフサイクルから、2-3年に一度のPSR適用を計画しておく

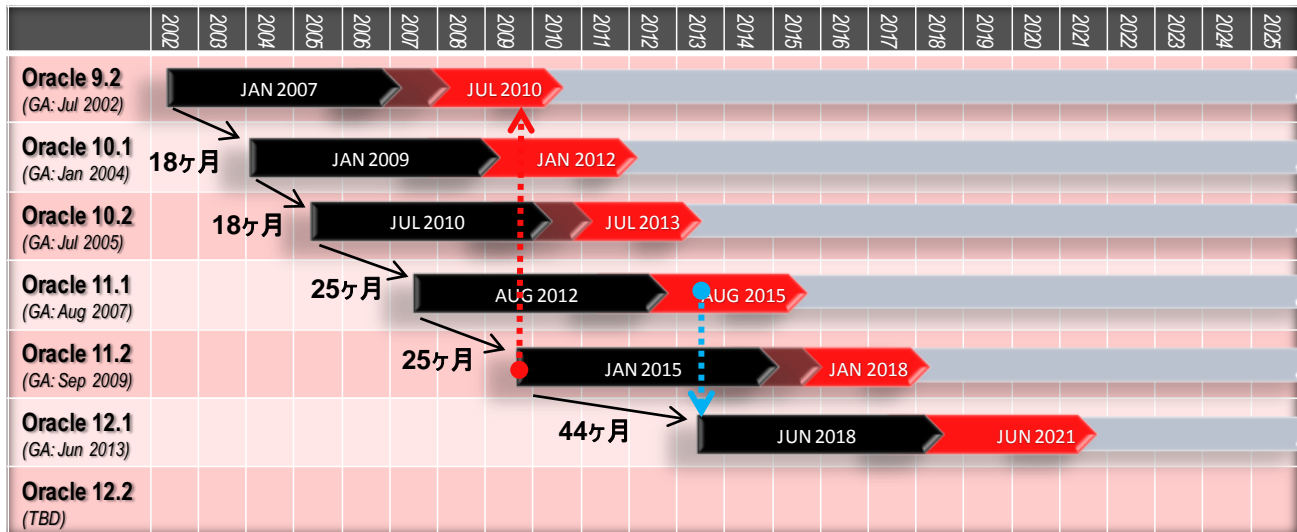
3 定期的な作業が可能なシステム構成と手順を採用する

- 繰り返しの実行に耐えられるよう、手順・バージョンは標準化する
- 定期的なアップグレードができるダウンタイム・コントロール

計画の定義 #1: サポート提供期間の確認

メジャー・リリースをスキップしない

- 製品リリース・サイクルの変化により、平行してサポートされるリリースの種類は少なくなる傾向にあります
 - 11gR2 がリリースされたタイミングでは4つ前のメンテナンス・リリースである 9iR2 もサポート期間中でした
 - 12cR1 がリリースされたタイミングでサポート期間中であるのは2つ前のメンテナンス・リリースまでです
 - サポート期間終了後も運用を継続する場合には、次のメンテナンス・リリースに移行するタイミングを予定しておく必要があります



11gR2をリリースした時点では、9iR2以降の5つのリリースがサポート期間中

2013年11月現在サポート期間中のリリースは3つ

計画の定義 #2: PSR の選定とアップグレードのタイミング

パッチが提供されるPSR に常に保つ

- 各 PSR に対するパッチ提供は、その次のPSRのリリースから2年後（ベース・リリースは1年後）に終了します
- PSRは、1-2年に1度の頻度でリリースされているため、2年ないし3年に1度は新しいPSRの適用が必要です
- 現時点でパッチ提供が行われているのは、11.1.0.7、11.2.0.3、11.2.0.4、12.1.0.1 の4種類のPSRです（2014年12月現在）

	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
11g R1	Premier Support (2012年8月 末まで)		Extended Support (2015年8月 末まで)			Sustaining Support					
11.1.0.6	パッチ提供期間 (2009年9月18日まで)										
11.1.0.7	パッチ提供期間 (2008年9月-2015年8月 末まで)										
11g R2	Premier Support (2015年1月 末まで: 例外的に5年以上に設定)				※	Extended Support (2018年1月 末まで)		※HP-UXのみ (2020年1月 末まで)		Sustaining Support	
11.2.0.1	パッチ提供期間 (2011年9月13日まで)										
11.2.0.2	パッチ提供期間 (2013年10月 末まで)			※11.2.0.4 の出荷をカバーする期間まで延長済み							
11.2.0.3	パッチ提供期間 (2011年9月-2015年8月27日まで)										
11.2.0.4	パッチ提供期間 (2013年8月-2018年1月 末まで)										
12c R1	Premier Support (2018年7月 末まで)						Extended Support (2021年7月 末まで)				
12.1.0.1	パッチ提供期間 (2013年6月-TBD)										

Free Extended Support 提供期間 (Extended Support を無償で提供)

計画の定義 #3: パッチ適用の頻度

四半期ごとにPSU を適用する

- Interim Patch より PSU が推奨

- オラクル社の開発部門でリリース前に行っているテストの種類・量ともに大きな違いがあるため
 - Interim Patch は個別環境での特定の不具合を修正するためのパッチなので不具合修正テストのみ
 - PSUは、リリースより1ヶ月前にコードをFIXし、機能テスト、システムテスト、パフォーマンステストなど、3000時間を超えるテストを経て出荷される

- PSU は四半期ごとに適用する（ Exadata/Database Applianceの場合は四半期ごとのBundle Patchが推奨）

- セキュリティと不具合の予防のために、四半期ごとの適用を想定して提供されるパッチセット
- 最新のセキュリティ・パッチである SPU に加えて、広く該当する可能性がある不具合の修正が提供されおり、セキュリティと不具合の両方に対して問題が起きる前に対応することができる
- 更に、PSU には実行計画に影響する修正、製品設定を変更する修正は含まれないため、負荷が高いパフォーマンステストを行う必要がない
- 累積パッチであるため四半期ごとの適用が難しい場合には半期ごとの適用も可能(それ以下の頻度は推奨しない)
- 個別パッチとの競合を解消するパッチも提供される
- SPU (セキュリティパッチ)は、12.1.0.1 からは個別での提供は無く、PSUに含まれた形式でのみ提供される

(参考) オラクルが提供するデータベース・パッチの種類

Database Patchには大きく6つの種類があり、いつどれを適用するかの戦略が必要

パッチ名称		適用対象コンポーネント	リリースサイクル
Interim Patch (One-off, PSE)		Oracle Database	不定期
Security Patch Update (SPU)*1			四半期毎
Patch Set Updates (PSU)		Oracle Database, Grid Infrastructure	四半期毎
Patch Set Release (PSR)			年次またはそれ以上
Exadata Storage Server patch		Exadata Storage Server (SW/OS/FW) Database Server (OS/FW)	四半期～半期毎
Bundle Patch	Quarterly Database Patch for Exadata (QDPE)*2	Oracle Database, Grid Infrastructure	四半期毎
	Interim Database Patch for Exadata (Interim BP) *3		月次または2ヶ月毎

- PSU, Bundle Patch は累積型です
- 上表の4種類のパッチに加え、Windows環境ではBundle Patch が提供されます
- *1: これまで CPU (Critical Patch Update) と呼ばれていました。12.1.0.1以降は提供されません
- *2: 推奨バンドルパッチは「Quarterly Database Patch for Exadata (QDPE)」と呼ばれ、SPUやPSUを含むように四半期ごとにリリースされます
- *3: QDPE以外にも、月次もしくは2ヶ月毎に Bundle Patch (Interim BP) がリリースされます (Interim BPのリリース周期は変更されることがあるので、最新の情報は note 888828.1 を参照してください)
- QDPEは多くのお客様に適用いただくBundle Patchです。不具合にヒットして修正が必要で次のQDPEを待てない場合にはInterim BPの適用を検討してください。

計画の定義 #4: 長期的な戦略を立てる

少数に絞った手順を繰り返し行うことで洗練させる

- 原則1: 自社内のバージョンの種類を少なくし、共通バージョンを利用する
 - インストールテストや基本テストの重複する作業の数を減らすことができ、既知の不具合などの情報を共通化することができる
 - パッチ適用やアップグレードのタイミングを管理しやすくし、見落としが少ない
- 原則2: アップグレード要件をパターン分けして、少数の方法に振り分ける
 - なるべく少ない方法に絞ることでスキルと知見を蓄積することができる
 - 絞り込んだ方法を繰り返し実施することでプロセスの改善を続ける
- 原則3: 複数のデータベースをアップグレードする場合、どこからプロジェクトを開始するかルールを決めておく
 - 最も大変なプロジェクトから始めるか、最も簡単なプロジェクトから始めるか
- 原則4: 隣接するメンテナンス・リリースや PSR へのアップグレードを基本にする
 - 新しいリリースでは求められるデータやトランザクション量に見合ったデータ移行方法やアップグレードツールが提供されている
 - 要件やデータ量が進化して、データベースが古いままの場合、要件と選択肢のギャップが広がり、結果的に想定外の負荷がかかる

計画の定義 #5: アップグレード手法の選択

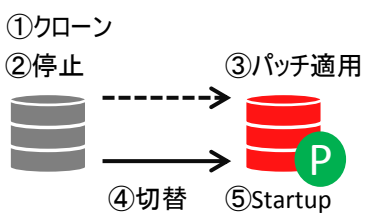
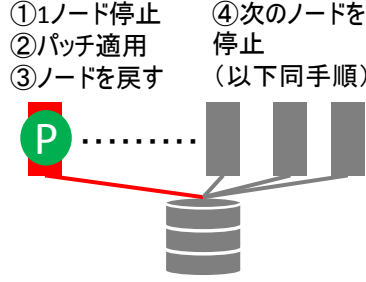
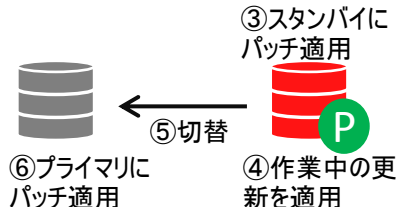
選択できる手段のうち、ダウンタイムとコストを最小限に抑えられるものを選択する

- データ移行、アップグレード後の切替、テスト方法について手法を選択します
 - 前述の原則に従って、いくつかのパターンをあらかじめ策定しておきます
- アップグレードに関する情報についての Magic Question などによりプロジェクトを整理することも有用です
 - Magic Question は、アップグレード手法の選定に影響を及ぼす要素についての質問で構成され、それに応じて適切な手法もご紹介できます
 - 移行元・移行先のバージョン
 - データベースサイズ（データ量、REDOのサイズなど）
 - HW移行の有無
 - OS 変更の有無
 - Endian の変更の有無
 - 移行するデータベース数
 - DataGuard、RACの利用有無
 - ダウンタイム要件
 - ネットワーク転送速度、等

アップグレードと アプリケーションテストの手法

ダウンタイムを最小化するパッチ適用の手法

RAC/DataGuardなどの高可用性構成はパッチ適用・アップグレードの際にも有効

名称	Out-of-Place Patching	Rolling Real Application Cluster Patching	Patch the Standby First
特徴	別Oracleホーム(クローン)を作成してパッチを適用し、稼働を切り替える方法	ダウンタイム無しの縮退運転のみで作業を行うことができる	パッチ適用後のスタンバイ環境でテストを行うこともできる
適した用途	シングル構成や、予備HWを用意できない場合でも利用できる	個別パッチ適用またはPSU適用 (RollingPatchに対応済のもの)	比較的頻繁に行うパッチ適用 (PSU等)
ダウンタイムの目安	ShutdownしてからStartupして切替するまで発生する	無し	数分 (DGのSwitchover)
ソフトウェア要件	無し	Enterprise Edition Real Application Clusters	11.2.0.2以上のEnterprise Edition (Data Guard設定)
方法と構成	 <p>①クローン ②停止 ③パッチ適用 ④切替 ⑤Startup</p>	 <p>①1ノード停止 ②パッチ適用 ③ノードを戻す ④次のノードを停止 ⑤パッチ適用 (以下同手順)</p>	 <p>①Data Guard運用 ②同期ストップ ③スタンバイにパッチ適用 ④作業中の更新を適用 ⑤切替 ⑥プライマリにパッチ適用</p>

データ移行ユーティリティ/機能とバージョンの対応

Oracle Databaseは格納されるデータ量が増えるに従って効率的なデータ移行機能を実装

- ・バージョンやダウンタイムなどの要件に応じて適切なデータ移行の方法を選択する
- ・データ移行、アップグレード方法を組み合わせることでダウンタイムを最小に抑えることができる

	方式	異なる			対応 Version	切り 戻し	データ量と Down-timeの 依存度	Down- time	作業 負荷
		OS	Block size	Character set					
データベースの アップグレード	Database Upgrade Assistant (DBUA)	×	×	×	9.2.0.8 - (To 11.2)	○	低	小	小
	コマンドラインアップグレード (CLI)	×	×	×	10.2.0.5 - (To 12.1)	○	低	小	小
移行	Export/Import	○	○	○	8 -	▲	高	大	小
	DataPump (expdp/impdp)	○	○	○	10.1 -	▲	中※1	小～中	小
	Transportable Tablespace (TTS)	×	×	×	8i -	○	中※1	小～中	中
	Cross Platform TTS	○	×	×	10.1 -	○	中※1	小～中	大
	GoldenGate	○	○	○	※2	◎	低	極小	中

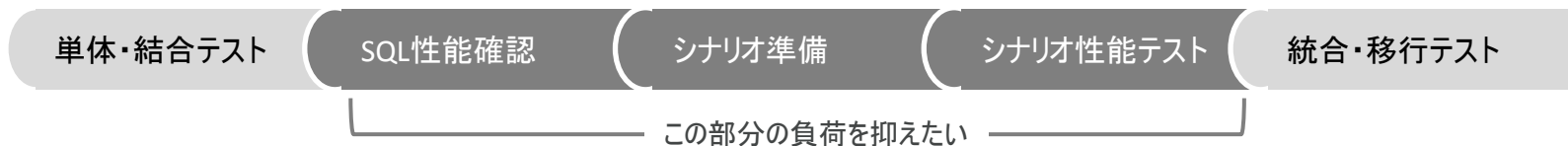
※1 目安として数TBなら Data Pump、数十TBなら TTSを使用することが考えられます。(参考レベル)

※2 移行元のバージョンに依存するため、日本オラクル社にご相談ください。

アプリケーション・テストの負荷を下げる

パフォーマンスに問題の出るSQLを特定してチューニングする

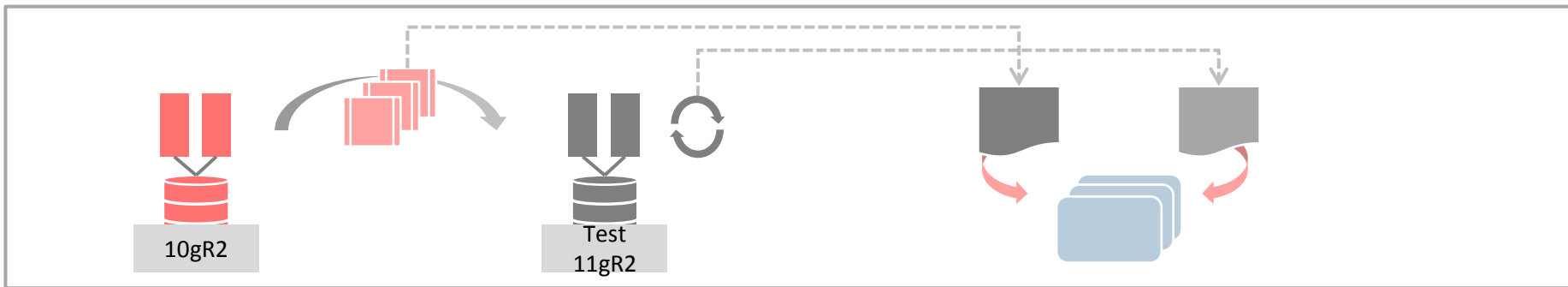
アプリケーション・テストの負荷



- 「全てのアプリケーションの全てのSQLをテストでチェックすることは出来ない」
 - アップグレード等で環境が変わる場合、全てのSQLの性能が劣化するわけではなく、逆に殆どのSQLで性能は向上する
 - 性能が劣化するSQLだけを簡易かつ正確に抽出することができれば、チューニングを行うSQLの数を減らすことができる
- 「本番環境と同じ環境を再現するだけのテストシナリオを用意することは無理だ」
 - 本番環境で実行されたSQLをそのまま実行することができれば、実際の環境と同じユースケースと実行負荷が再現できる
 - 更にシナリオの検討とスクリプトの準備に費やす時間が不要になる

アプリケーション・テスト: SQLパフォーマンス・チェック & チューニング

SQL Performance Analyzer (SPA)によるSQLの性能劣化の抽出とSQL Tuning Advisorによるチューニング
(例: 10gR2から11gR2のアップグレード)



STEP-1

- SQL ワークロードやパフォーマンス統計を、SQL Tuning Set (STS) に格納する
- フィルタリング、追加も可能 (9i/10gR1はSQLトレースからSTSに変換する必要がある)

STEP-2

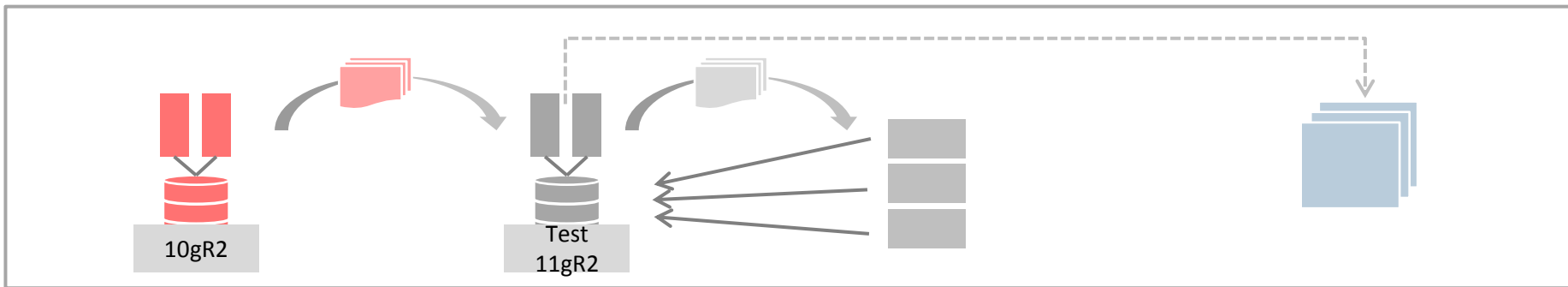
- 11gR2のテスト環境へSTSをインポート
- SPAからSTSを使ってSQLをシリアルに1回ずつ実施して、実行計画とパフォーマンス統計を取得する

STEP-3

- 10gR2と11gR2での違いをレポートとして出力 (実行時間、オプティマイザコスト、読み取りブロック数など)
- 変更前後で影響のあったSQLをリストして表示
- SQL Tuning Advisorを使って劣化したSQLをチューニング
- 10gR2の時の実行計画を採用したいときはSQL Plan Managementを使用

アプリケーション・テスト: スループット負荷テスト

SQLベースでのチューニングが完了したプログラムをDatabase Replayを使って本番環境の状態で行
(例: 10gR2から11gR2のアップグレード)



STEP-1

- 移行元環境でのトランザクションを1ヶ月～1週間前からキャプチャしておく。
(統計情報も取得しておく)

STEP-2

- テスト環境を用意する
(できれば本番環境と同じ環境)
- キャプチャしたファイルの本番環境からテスト環境にコピー
- テストを実行する環境で、キャプチャ・ファイルからリプレイ・ファイルに変換

STEP-3

- リプレイ・クライアントから、本番環境でキャプチャされたワークロードを実行時間・並列度・コミット順を再現するリクエストを送信
- 実行並列度や、処理が起きていない時間の圧縮は可能
- 11.2.0.3+Patch以上で Consolidated Replayも可能

STEP-4

- キャプチャ時とリプレイ時の違いをレポートとして出力
- パフォーマンスの違い
- エラー発生の違い
- データ処理の違い(行数など)
- 違いの発生したSQLを特定することができる

アップグレードをしない理由

主な理由は「テストの負荷」と「システムダウンやパフォーマンス劣化のリスク」

アプリケーションテスト/パフォーマンステストの負荷が高い

- 影響の起きうる範囲が調べられない
- パフォーマンスが劣化する
- 本番環境と同じトランザクションを再現できない



テスト・ツールの利用により負荷を削減することができます

- Database Replay により実運用環境のキャプチャ&リプレイ機能による実際の運用環境を再現
- SQL Plan Management によるアップグレード前後のパフォーマンスをSQLごとに比較し、影響を抽出

ダウンタイムが許容できない、サービス停止のリスクが高い

- システムが止められないのでアップグレードできない
- 今問題のないシステムに手を加えるリスクが取れない



MAA を含むアップグレード手法は確立されています

- ダウンタイムが極めて少ないアップグレードの事例が数多くあり、その中で確立されたダウンタイム最小化の手段や手順をご紹介します

アップグレードをする理由が見つからない

- インターネットに接続しない社内システムなのでセキュリティ対策はそれほど重要ではない
- アップグレードをするメリットを金額換算できない



アップグレードをしないリスクは想定できます

- セキュリティ事故の大半は社内発生しています
- セキュリティ事故を含め、アップグレードをしないリスクは金額で想定することができます

ケーススタディ

ケース・スタディ

1. 同一OSでのアップグレードの場合: コマンドラインアップグレード
2. 異なるOSでのアップグレードの場合: Data Pump
3. ニア・ゼロ・ダウンタイムのアップグレード: GoldenGate

	方式	異なる			対応 Version	切り 戻し	データ量と Down-timeの 依存度	Down- time	作業 負荷
		OS	Block size	Character set					
データベースの アップグレード	Database Upgrade Assistant (DBUA)	×	×	×	9.2.0.8 - (To 11.2)	○	低	小	小
	① コマンドラインアップグレード (CLI)	×	×	×	10.2.0.5 - (To 12.1)	○	低	小	小
移行	Export/Import	○	○	○	8 -	▲	高	大	小
	② DataPump (expdp/impdp)	○	○	○	10.1 -	▲	中※1	小～中	小
	Transportable Tablespace (TTS)	×	×	×	8i -	○	中※1	小～中	中
	Cross Platform TTS	○	×	×	10.1 -	○	中※1	小～中	大
	③ GoldenGate	○	○	○	※2	◎	低	極小	中

ケース・スタディ① 同一OSでのアップグレードの場合

Data Guard Physical Standby を利用したデータコピーでダウンタイムを短縮

■ 移行元:

- 2-node RAC × 6
- 10.2 / RAW
- RH Linux 32bit

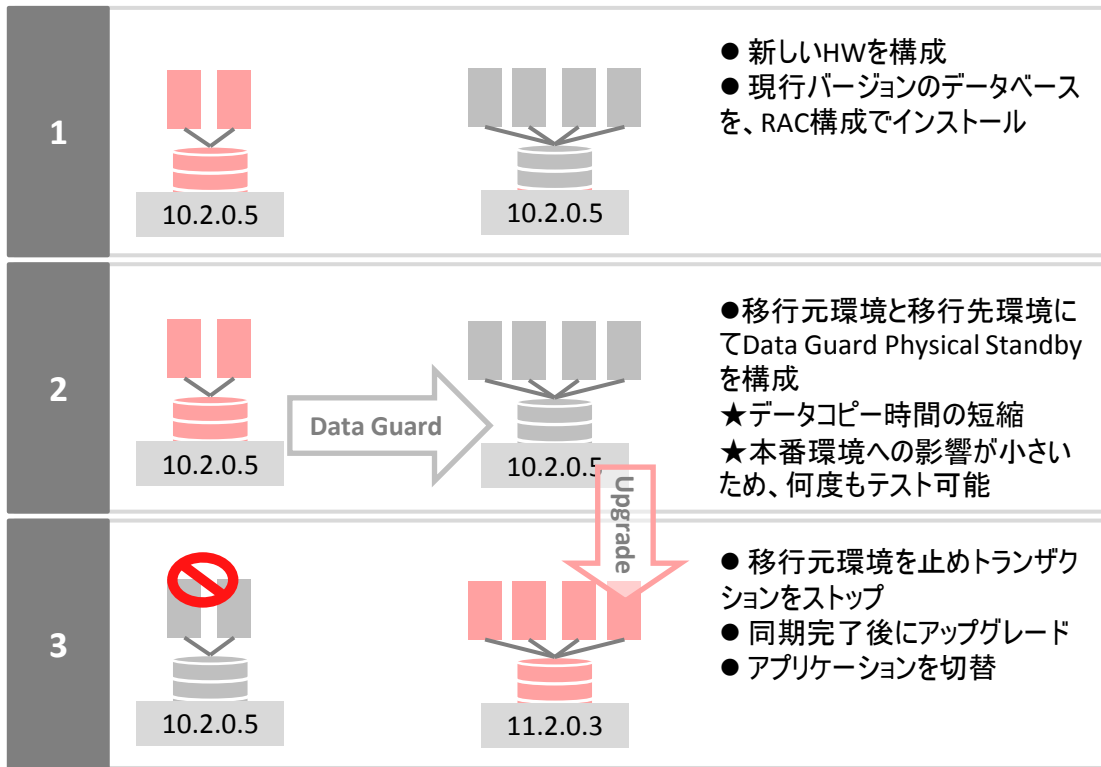
■ 移行先:

- 4-node RAC × 6
- 11.2 / ASM
- RH Linux 64bit

● 要件と補足説明

• ダウンタイムは4時間しか許容されていなかったため、6系統あるシステムを一度にアップグレードすることはできず、順次1つずつ実施

• ネットワークが細く、更にLOB型データを使っていたため、Data Pumpでは許容されるダウンタイム内のデータ移行が不可能だった。従ってよりデータ移行を短時間でできる方式を選択した



ケース・スタディ① 同一OSでのアップグレードの場合



顧客名

プロジェクト

制約

準備

アップグレード

備考

Success?

■ 顧客名:

Interhyp AG

- ドイツ ミュンヘンに本社を置く金融機関
- 住宅と開発金融の銀行
- ドイツの主要銀行へ銀行業務を提供
- オランダのING銀行の100%子会社



ケース・スタディ① 同一OSでのアップグレードの場合



顧客名

プロジェクト

制約

準備

アップグレード

備考

Success?

■ プロジェクト・スコープ:

- Oracle 10.1.0.5 (RH Linux 32bit) 2 Node RACの6システムをアップグレード
- アップグレード先:
 - Oracle RAC 11.2.0.2 with ASM
 - RH Linux 64bit
- 主要システムのハードウェア交換
2ノード・クラスター⇒ 4ノード・クラスター

ケース・スタディ① 同一OSでのアップグレードの場合



顧客名

プロジェクト

制約

準備

アップグレード

備考

Success?

■ 制約:

- 各データベースのダウンタイムは4時間以内
 - 同時並行ではなく、順番に移行する
- ネットワーク帯域はData Pumpで移行するのに十分ではない
- 移行元のデータベースにはLOBが存在

ケース・スタディ① 同一OSでのアップグレードの場合



顧客名

プロジェクト

制約

準備

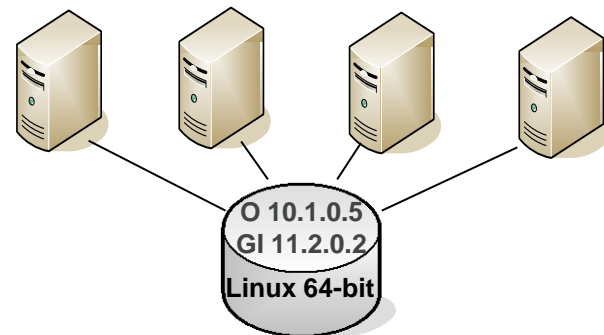
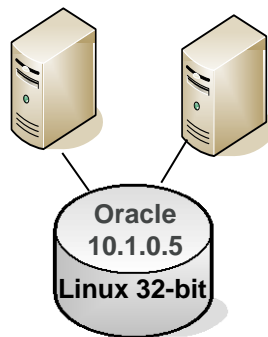
アップグレード

備考

Success?

■ 新しいクラスタの準備

- Oracle Grid Infrastructure 11.2をインストールし、パッチを適用
- アップグレード時間を30分以内まで短縮
 - 未使用コンポーネントを本番データベースから削除



ケース・スタディ① 同一OSでのアップグレードの場合



顧客名

プロジェクト

制約

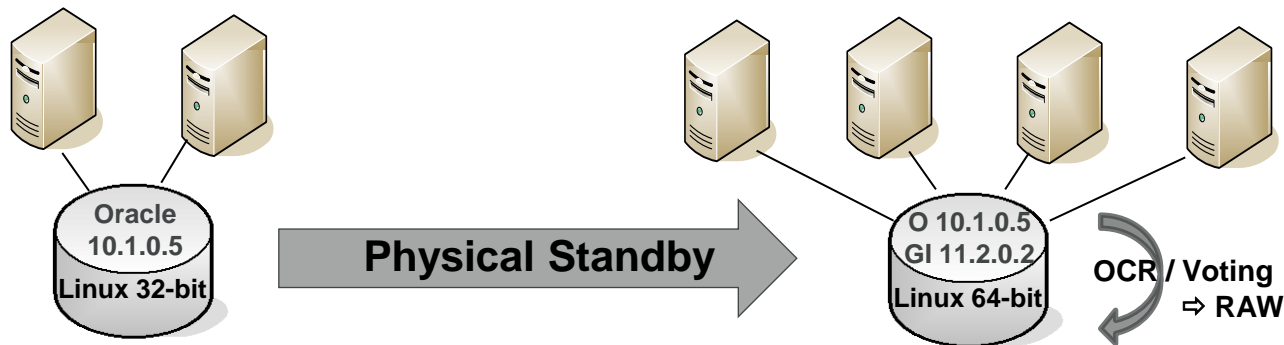
準備

アップグレード

備考

Success?

- 移行方法としてフィジカル・スタンバイを使用
 - データコピーのダウンタイムを解消
 - Oracle 10.1.0.5 ⇨ Oracle 10.1.0.5 (11.2 ASM上に)
注意: この構成は正式には未サポートだが、動作可能
 - スタンバイ・データベースを起動しアップグレード
 - 本番環境に影響を与えることなく、何度もテスト可



ORACLE

ケース・スタディ① 同一OSでのアップグレードの場合



顧客名

プロジェクト

制約

準備

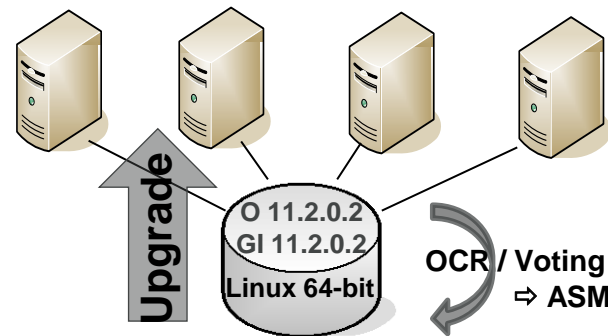
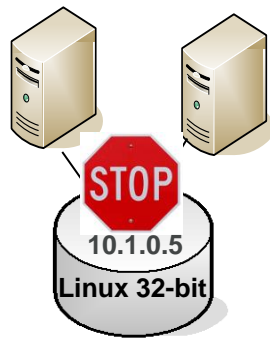
アップグレード

備考

Success?

■ アップグレード

- スタンバイ環境を稼働しSTARTUP UPGRADEモードで起動
 - 全てのパッケージ、コード(32bit ⇒ 64bit!)の無効化と再コンパイル
- データベースをクラスタウェアに登録し、OCRと投票ディスクをASMへ移動



ORACLE

ケース・スタディ① 同一OSでのアップグレードの場合



顧客名

プロジェクト

制約

準備

アップグレード

備考

Success?

- オプティマイザ

- オプティマイザに関するいくつかの問題が見つかった

- レポート処理に影響があった

- 改善方法: ヒント追加、SQL書き直し、パッチ適用、そしてSQLプロファイルの追加

ケース・スタディ① 同一OSでのアップグレードの場合



顧客名

プロジェクト

制約

準備

アップグレード

備考

Success?

▪ Live?And alive ?

- Yes!!! : 2010/11/27稼働
- 全体のダウンタイム: 2時間以内
- データベースのアップグレードに要した時間:
24分 + 5分(再コンパイル)
- Oracleソフトウェア・スタック全体を利用することで
とても強固な構成に

ケース・スタディ② 異なるOSでのアップグレードの場合

中間システムを介した Upgrade + Data Pumpの2段階移行

■ 移行元:

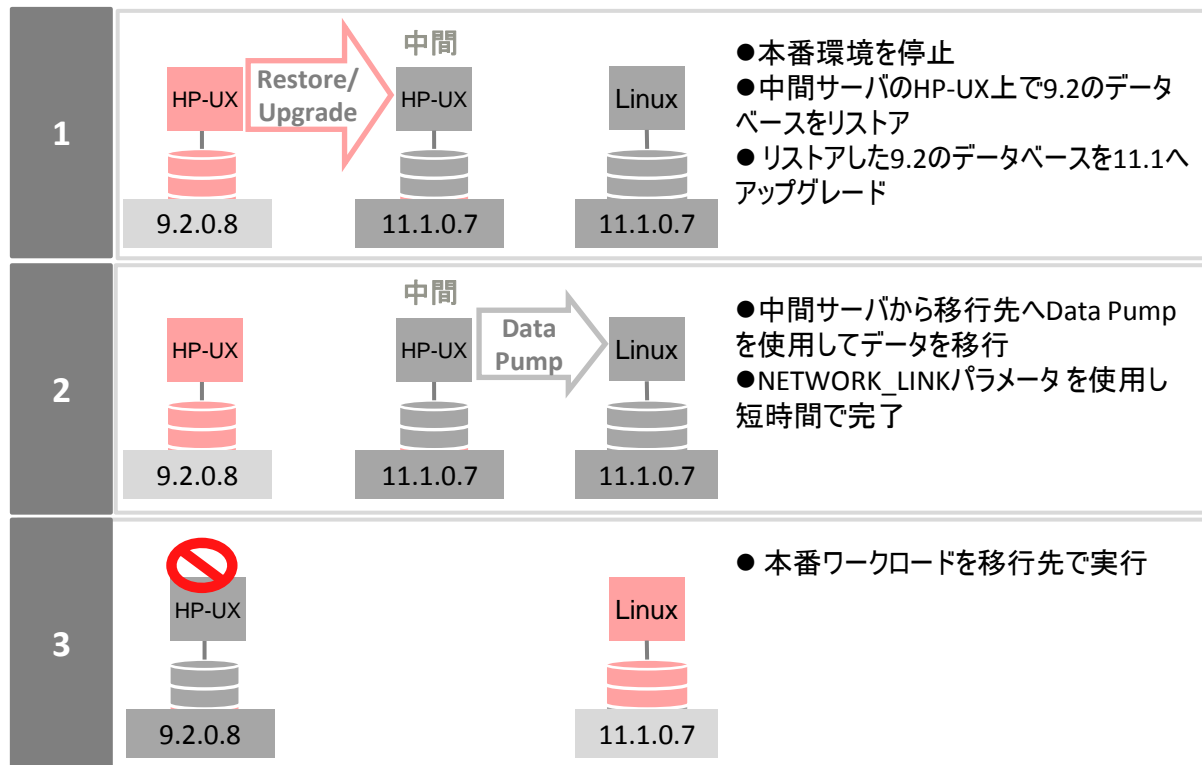
- 9.2
- HP-UX

■ 移行先:

- 11.1
- Oracle Enterprise Linux 64bit

● 要件と補足説明

- 24時間の移行時間の中で、8TBのデータを移行、OSおよびエンディアンの変更を伴うアップグレード事例
- 中間システム上で11.1へアップグレードした後、Data Pumpを使用して移行先へデータを高速に移行



ケース・スタディ② 異なるOSでのアップグレードの場合



顧客

プロジェクト

制約

準備

アップグレード

備考

成功?

■ Payback GmbH

- 本社はドイツ ミュンヘン
- カスタマイズされたITソリューションをベースとした専門の顧客ロイヤルティ・プログラムの開発と運営
- ヨーロッパで最大のボーナス・プログラムであるPayback のプロバイダ

ケース・スタディ② 異なるOSでのアップグレードの場合



顧客

プロジェクト

制約

準備

アップグレード

備考

成功?

- プロジェクト・スコープ:
 - 7 TBと1.5 TBをHP-UXからExadata V1へ移行
 - 異機種間, クロス・エンディアン, クロス・バージョン
 - Oracle 9.2.0.7 on HP-UXからOracle 11.1.0.7 on OELへ
 - 4か月間の計画と移行フェーズ
 - 2009年8月から11月
 - 稼動予定日
 - 2009年11月15日

ケース・スタディ② 異なるOSでのアップグレードの場合



顧客

プロジェクト

制約

準備

アップグレード

備考

成功?

■ 制約:

- 全てを24時間以内に移動
- ネットワーク・ボトルネック
 - 移行元システムにInfiniBand ハードウェアを実装
⇒ ~ 3GB/sec スループット!



ケース・スタディ② 異なるOSでのアップグレードの場合



- 顧客
- プロジェクト
- 制約
- 準備
- アップグレード
- 備考
- 成功?

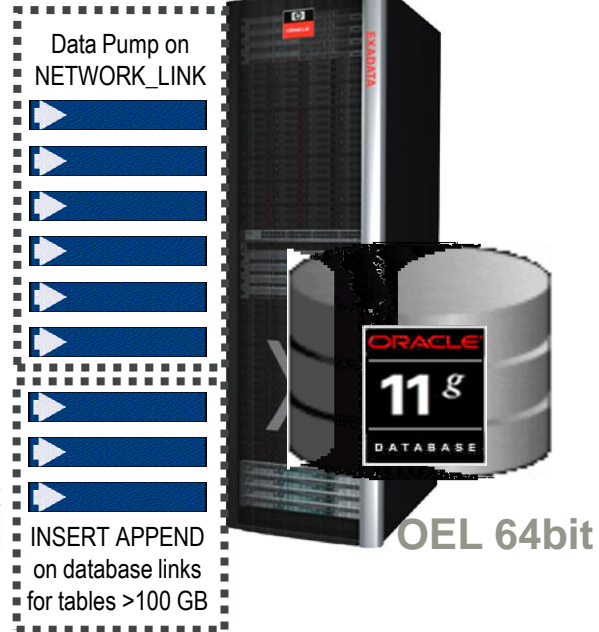
■ セットアップ:



ケース・スタディ② 異なるOSでのアップグレードの場合

- 顧客
- プロジェクト
- 制約
- 準備
- アップグレード
- 備考
- 成功?

■ 3回の移行テスト:



ケース・スタディ② 異なるOSでのアップグレードの場合



- 顧客
- プロジェクト
- 制約
- 準備
- アップグレード
- 備考
- 成功?

■ 並行稼動: パフォーマンス・テスト



ケース・スタディ② 異なるOSでのアップグレードの場合

- 顧客
- プロジェクト
- 制約
- 準備
- アップグレード
- 備考
- 成功?

■ 実アップグレード/移行



ケース・スタディ② 異なるOSでのアップグレードの場合

顧客

プロジェクト

制約

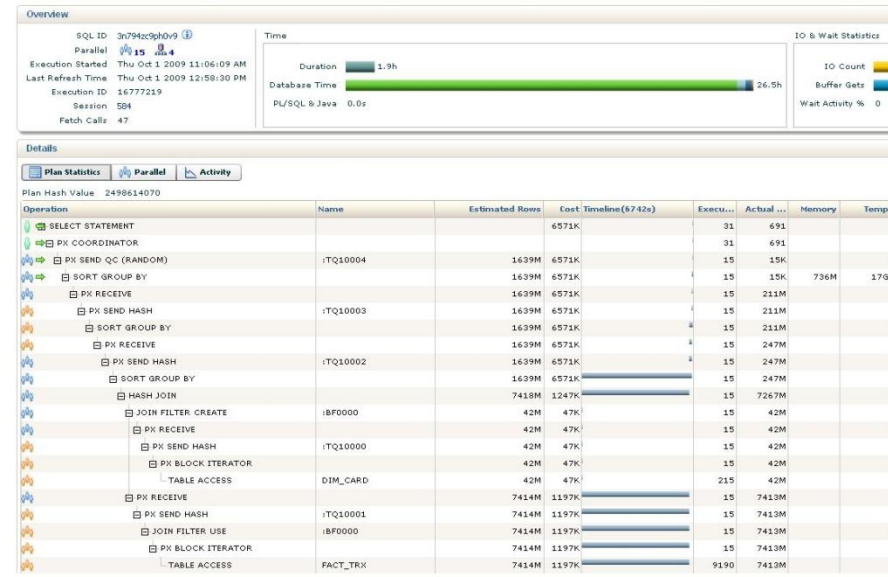
準備

アップグレード

備考

成功?

- 例: ジョブ実行時間が30時間から2時間以下へ
– SQL単体の変更なし



顧客

プロジェクト

制約

準備

アップグレード

備考

成功?

- Live? And alive?
 - Yes! 2009年11月初旬稼動
 - 予定より2週間早く稼動
 - リストア, リカバリ, アップグレードと最終移行は~20時間以内に完了
 - 劇的なパフォーマンス改善
 - ジョブ実行時間を80%近く減少
 - 実際に、とても速いパフォーマンスについてユーザーからのクレームが!!

ケース・スタディ② 異なるOSでのアップグレードの場合

顧客

プロジェクト

制約

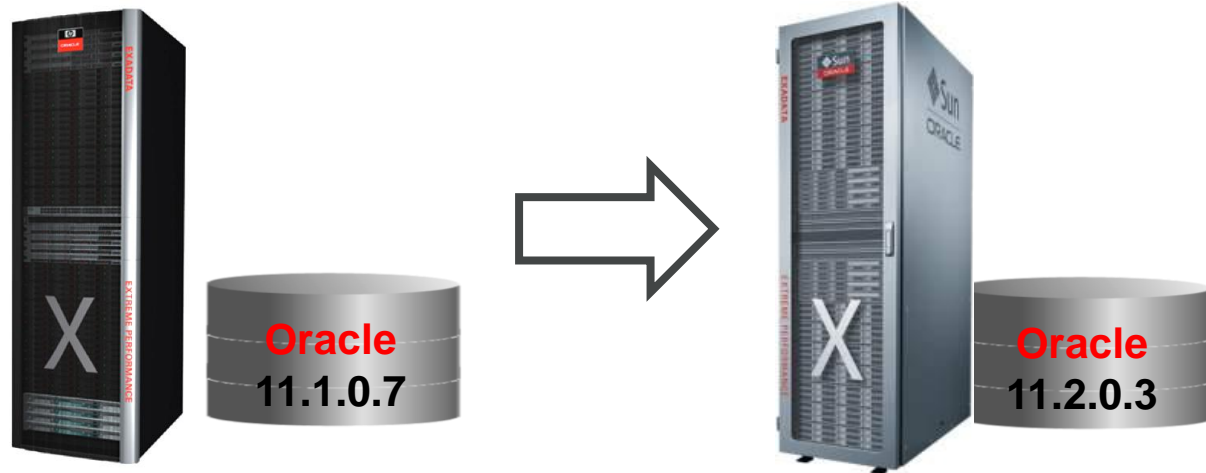
準備

アップグレード

備考

成功?

- 2012年: Exadata X2-2へアップグレード
 - RMANを使用して新環境へDBをリストア
 - アップグレードスクリプトで 11.2.0.3へアップグレード



ケース・スタディ③ ニア・ゼロ・ダウンタイムのアップグレード

GoldenGate を利用してアプリケーション停止時間を極小化

■ 移行元:

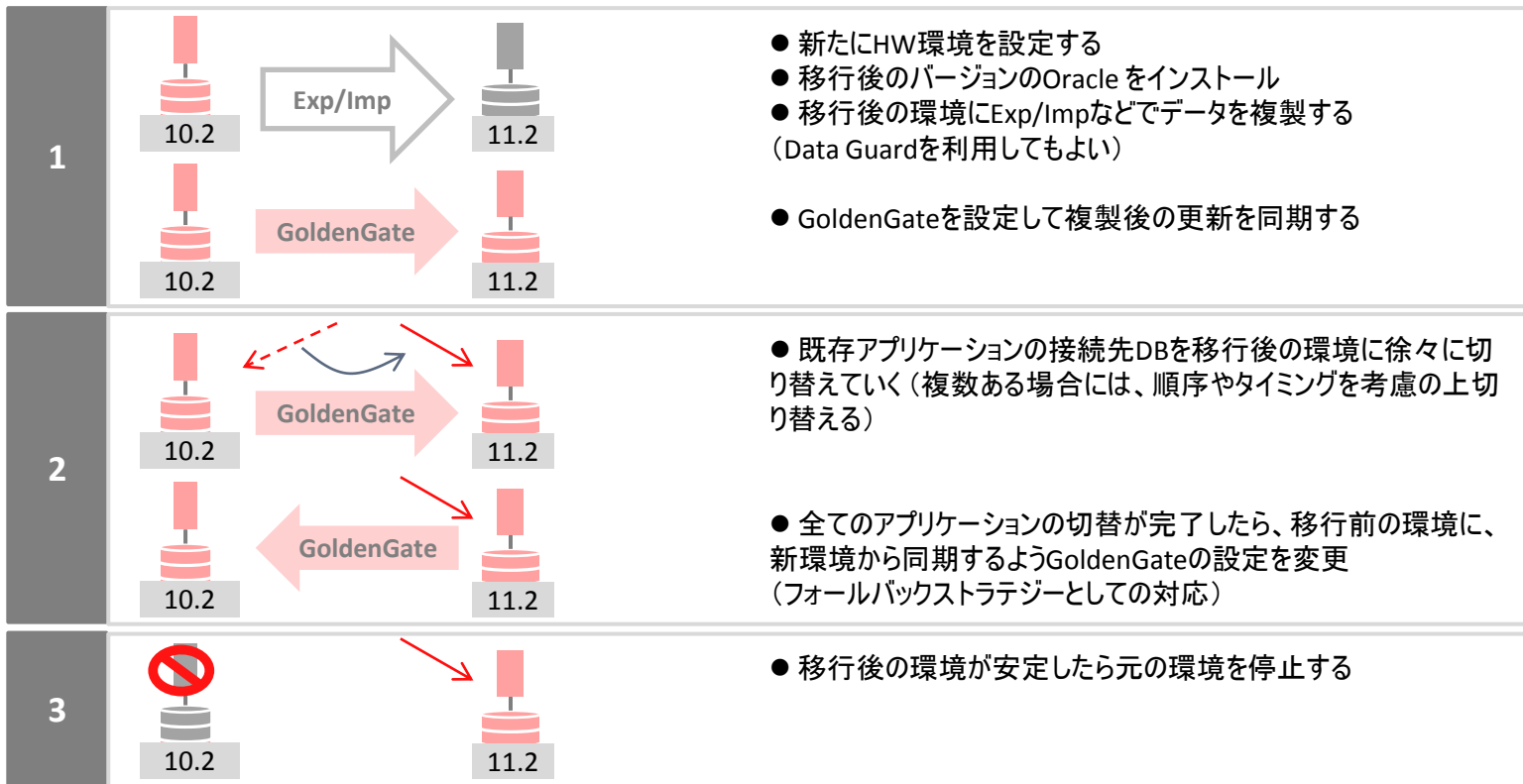
• 10.2

■ 移行先:

• 11.2

■ その他要件

- アプリケーション利用者から見た時のダウンタイムの極小化
- ミッション・クリティカルシステム



ケース・スタディ③ ニア・ゼロ・ダウンタイムのアップグレード

amadeus
Your technology partner

Customer

Project

Constraints

Preparation

Migration

Success?

Remarks

顧客名: Amadeus

- 世界の観光旅行業界に先進のテクノロジー及びソリューションを提供しているリーディング・カンパニー

DISTRIBUTION
BUSINESS

711 airlines
110,000+ hotel properties
30 car rental companies
50+ cruise and ferry lines
207 tour operators
24 insurance companies
95 railways

IT SOLUTIONS

Inventory
Departure Control
e-Commerce

Airlines
Airports
Hotels
Rail



20,000+ tx/sec (peak)
< 0.3 sec response time
10 Petabytes of storage
3+ million net bookings/day
> 1 billion tx/day

ORACLE

ケース・スタディ③ ニア・ゼロ・ダウンタイムのアップグレード

aMADEUS
Your technology partner

Customer

Project

Constraints




Preparation

Migration

Success?

Remarks

- 本番環境10gから新HW／OSの11g環境へ移行

Source		Target
Oracle 10.2 RAC HP-UX v2		Oracle 11.2.0.2/3 RAC HPUX v3
		Oracle 11.2.0.2/3 RAC RHE Linux
Oracle 10.2 Single Instance HP-UX v2		Oracle 11.2.0.2/3 RAC One RHE Linux

ORACLE

ケース・スタディ③ ニア・ゼロ・ダウンタイムのアップグレード

amADEUS
Your technology partner

Customer

Project

Constraints

Preparation

Migration

Success?

Remarks

- 四半期毎にメンテナンス実施: 停止時間
- データベースの停止時間は最大5分
- 停止時間以外での業務影響は無し
- インディアン変換: HP-UX ⇔ Linux (big⇔little endian)
- 停止時間中もしくはメンテナンス後に切り戻す可能性あり
- DBへの大量更新
 - Redo生成量: 20MB/sec
- 大規模DB (約14TB)
- 物理構成を再構成する可能性あり
 - データ・ディクショナリをフレッシュ
 - 表領域とパーティションを再設計

ORACLE

ケース・スタディ③ ニア・ゼロ・ダウンタイムのアップグレード



Customer

- 徹底的なPoCを実施 (Oracle支援)

- 機能面とデータ量にフォーカス

Project

- スケジュールに基き移行プロセスを標準化

Constraints

- 設定、監視、チューニング、スイッチオーバー用の作り込みスクリプトと手順を確立

Preparation

- DBAを支援する社内スペシャリストのトレーニング

Migration

Success?

Remarks

ケース・スタディ③ ニア・ゼロ・ダウンタイムのアップグレード

amaDEUS
Your technology partner

Customer

Project

Constraints

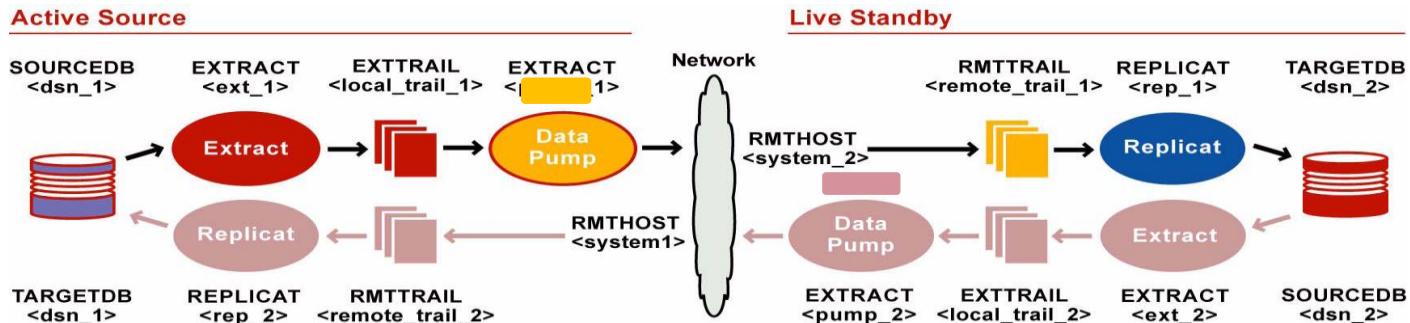
Preparation

Migration

Success?

Remarks

- 11g のインストール
 - フィジカル・スタンバイと Data Pump
- GoldenGateのインストール、設定、適用のチューニング



- 移行元と移行先のデータ確認 (Veridata)
- スイッチオーバーと切り戻しをリハーサル
- スイッチオーバー: 適用を停止して、逆方向への適用開始

ORACLE

ケース・スタディ③ ニア・ゼロ・ダウンタイムのアップグレード

Customer

Project

Constraints

Preparation

Migration

Success?

Remarks

■ データベース移行: 成功

- スイッチオーバーの時間: 2 - 6分
- 切り戻し無し

Source	Target	Migrated
Oracle 10.2 RAC HP-UX v2	Oracle 11.2.0.2/3 RAC HPUX v3	6
	Oracle 11.2.0.2/3 RAC RHE Linux	3
Oracle 10.2 Single Instance HP-UX v2	Oracle 11.2.0.2/3 RAC One RHE Linux	6

ケース・スタディ③ ニア・ゼロ・ダウンタイムのアップグレード

amADEUS
Your technology partner

Customer

Project

Constraints

Preparation

Migration

Success?

Remarks

- 異なるHW/OS、異なるOracle DBのバージョンへ円滑・安全に移行できるというコンセプトを証明



■ 考慮事項

- 移行先DBのインスタンス作成 (プラン・スタビリティを含む)
- データベース毎にGoldenGate設定をカスタマイズ
- サポートされないデータ型の対処 (例: ANYDATA)
- 移行元DBへのサプリメンタル・ロギングの影響
- DMLが多いDBにはGoldenGateをチューニング(適用プロセスを平行化)

ORACLE

まとめ

- パッチ適用・アップグレードの方針を策定して、定期的に実施
 - 計画、手順を確立して標準化する
- 移行時のアプリケーション・テストにはツールを活用
 - 負荷・工数を削減し、テストの網羅性を高めることができる
- 要件に応じた移行方法を検討
 - 各方法における手順や事例を確認して要件に適した移行方法を検討する

Hardware and Software

ORACLE®

Engineered to Work Together

ORACLE®

ORACLE®