オラクルコンサルが語る データベース・アップグレード時の SQL性能比較術

日本オラクル株式会社
クラウド・テクノロジーコンサルティング統括本部
シニアプリンシパルコンサルタント
加藤　健

2015.8.19

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。

アジェンダ

1. DBアップグレードの課題
2. 一般的なテストとSPAを使用したSQL単体性能テストの違い
3. SPAを利用したSQL単体性能テスト手順概要
4. SPAを使用したSQL単体性能の評価事例(screen only)
5. まとめ
6. Q&A
7. Appendix
    1. SPAを利用したSQL単体性能テスト手順詳細
    2. SPAのTIPS

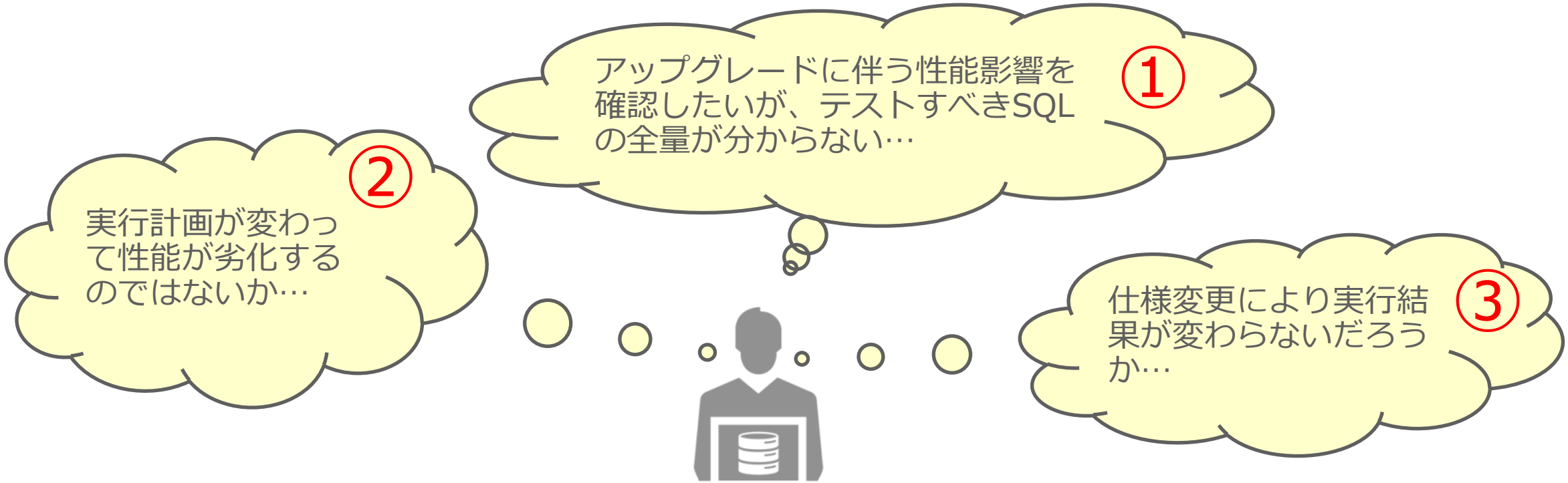
アジェンダ

1. DBアップグレードの課題
2. 一般的なテストとSPAを使用したSQL単体性能テストの違い
3. SPAを利用したSQL単体性能テスト手順概要
4. SPAを使用したSQL単体性能の評価事例(screen only)
5. まとめ
6. Q&A
7. Appendix
   1. SPAを利用したSQL単体性能テスト手順詳細
   2. SPAのTIPS

DBアップグレードの課題

DBアップグレードを躊躇したり、複雑にする要因

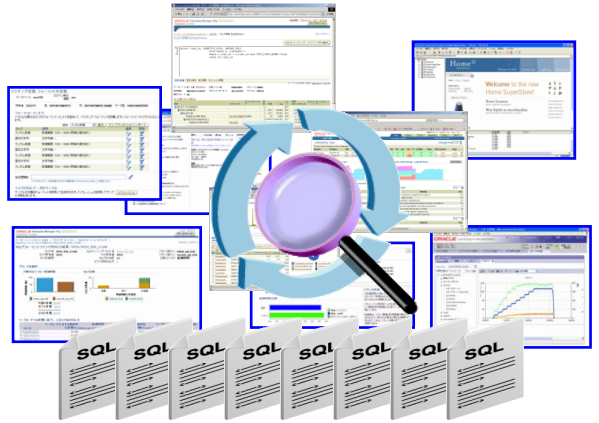
　DBアップグレードに伴い、多くのお客様が下記のような課題に直面します。DBアップグレードを成功させるには、これらの課題を全て解決する必要があります。



DBアップグレードの課題

①テストすべきSQL数が不明

どれだけのSQLがあるんだろう・・・
正確に調査するのは大変だ・・・



アプリケーション

システム規模が大きい場合、全てのアプリを調査して漏れなくSQLを抽出することは困難で、現実的ではありません。
例え出来たとしても、多くの工数・期間が掛かることが予想されます。
このため、本番機で実行されたSQLをキャプチャする方法をお勧めします。

DBアップグレードの課題

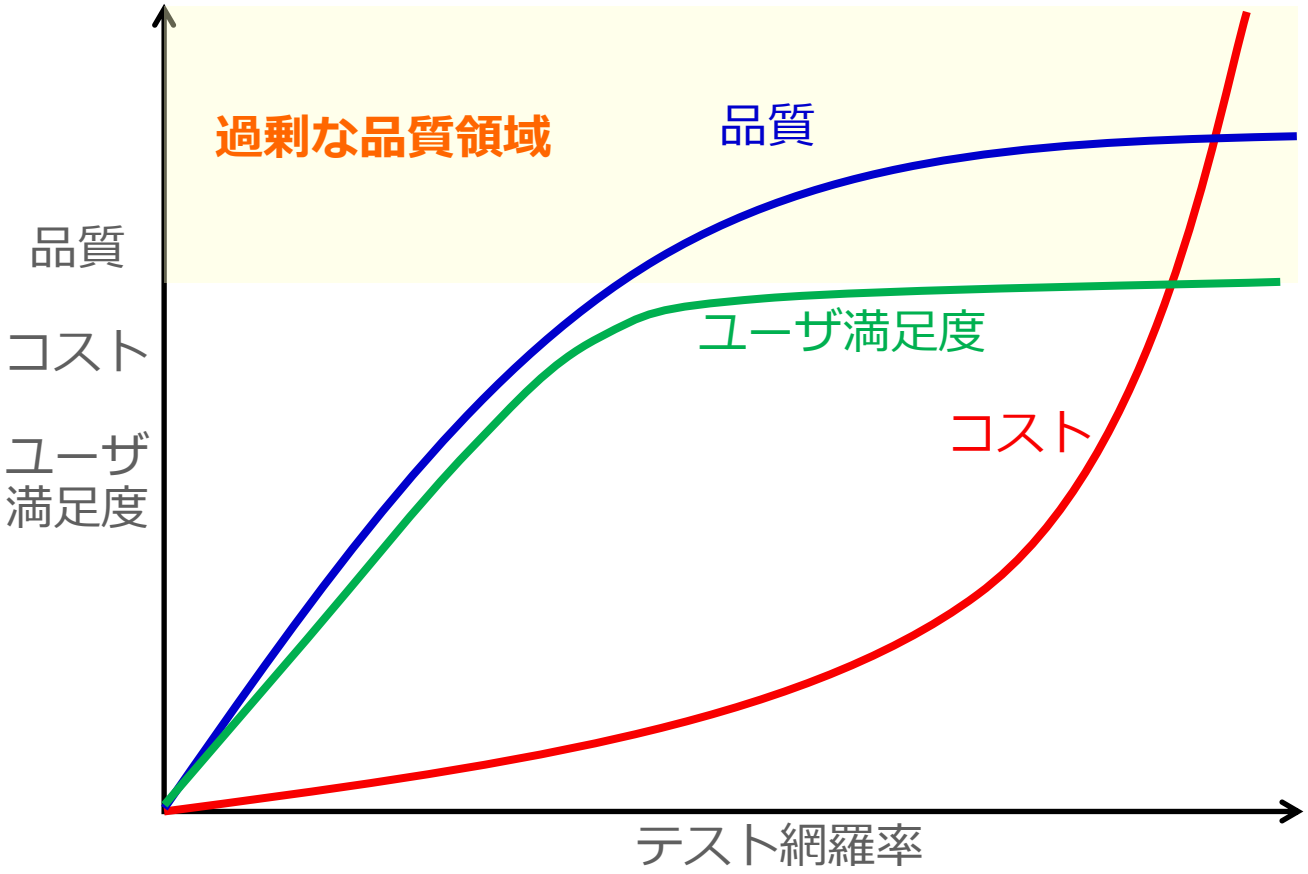
①テストすべきSQL数が不明

	アプリのソースを調査して SQLを取得する方法	本番機で実行されたSQLを キャプチャーする方法
網羅性	**中～高い。** ・全ソースに対して調査を行えば網羅性は高まるが、実際には重要な機能などに絞って実施することが多く、その場合の網羅性は中程度になる。 ・入力値のバリエーションを調べるのは困難。	**中～高い。** ・キャプチャー期間中に行われたアプリ操作の網羅性に依存する。キャプチャー期間中に全ての機能が実行された場合は網羅性は高くなる。 ・入力値のバリエーションを調べるのは容易。
コスト	**高い。** ・アプリ調査に多くの工数が掛かる。 ・アプリの数に比例して増大する。	**低い。** ・アプリ調査は不要。 ・SQL数が多くても工数はあまり増えない。
期間	**長い。** ・アプリ調査に多くの期間が掛かる。	**短い。** ・本番機でキャプチャーする期間のみ。

DBアップグレードの課題

①テストすべきSQL数が不明

【テスト網羅率と品質・コスト・ユーザ満足度の関係】



テスト網羅率を上げると品質は向上しますが、ある点を境にユーザ満足度は上がらなくなります（例えば、**ほとんど使われない機能を網羅的にテストして品質を上げてもユーザ満足度は上がりにくい**）。

過剰な品質によるコスト増を避け、低コストでテストを実現するには、網羅率に対して割り切ることも必要。

※テスト網羅率の向上は品質をあげる一要因となります。
試験環境や使用データなども品質を左右する要因となります。

DBアップグレードの課題

①テストすべきSQL数が不明

どちらの方法を採用してSQLを調査すべきかはプロジェクトの特性によって異なります。

コスト度外視、品質最優先の場合	「アプリのソースを調査してSQLを取得する方法」の採用を検討。※全ソースを対象
コスト、品質のバランス重視	「本番機で実行されたSQLをキャプチャーする方法」の採用を検討。
コスト最優先、品質は妥協可	

DBアップグレードの課題

②実行計画の変化により性能が変わる

実行計画が変化し、性能が変わる可能性があるのではないか。
性能が劣化する可能性もあるのではないか。

アップグレード前

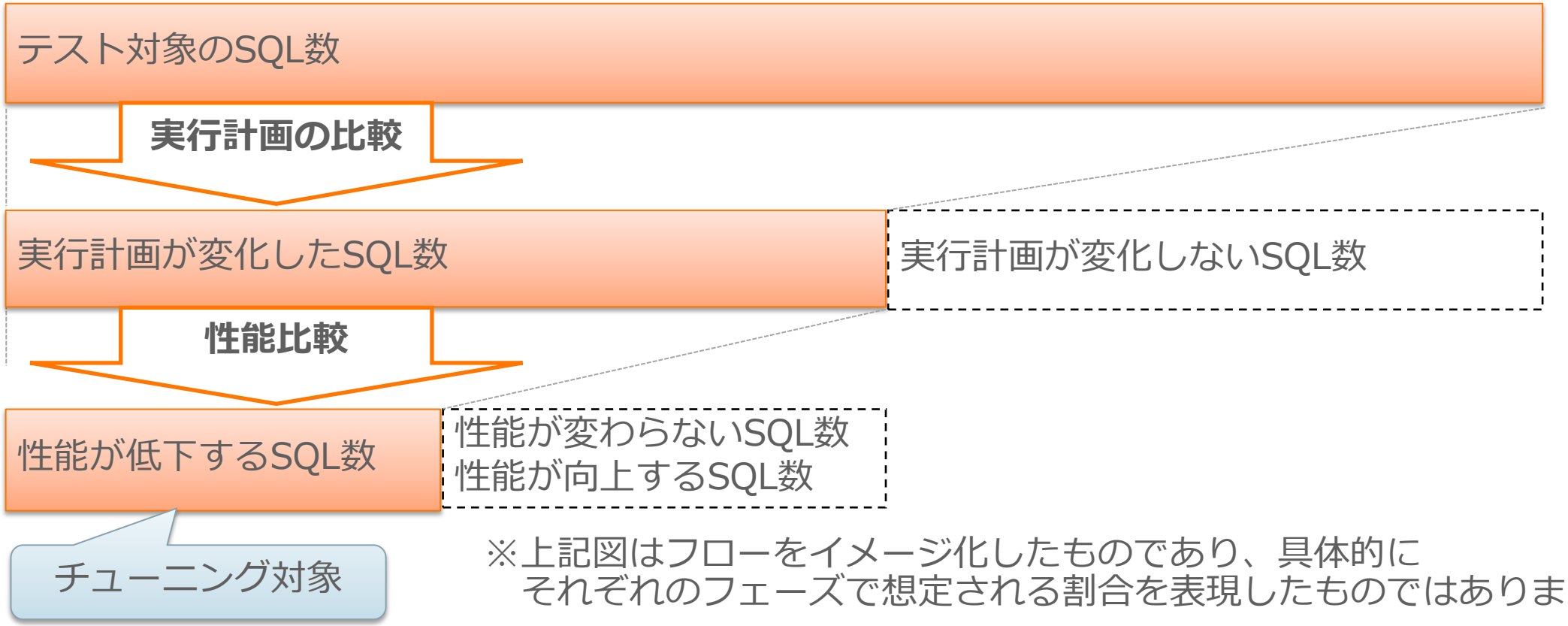
アップグレード後

SQLをアップグレード後の環境で実行し、実行計画が変化していないかを確認しましょう。もし、実行計画が変化している場合は実際に性能が劣化していないかの確認も行ってください。

DBアップグレードの課題

②実行計画の変化により性能が変わる

性能が劣化したSQLを特定することが重要。

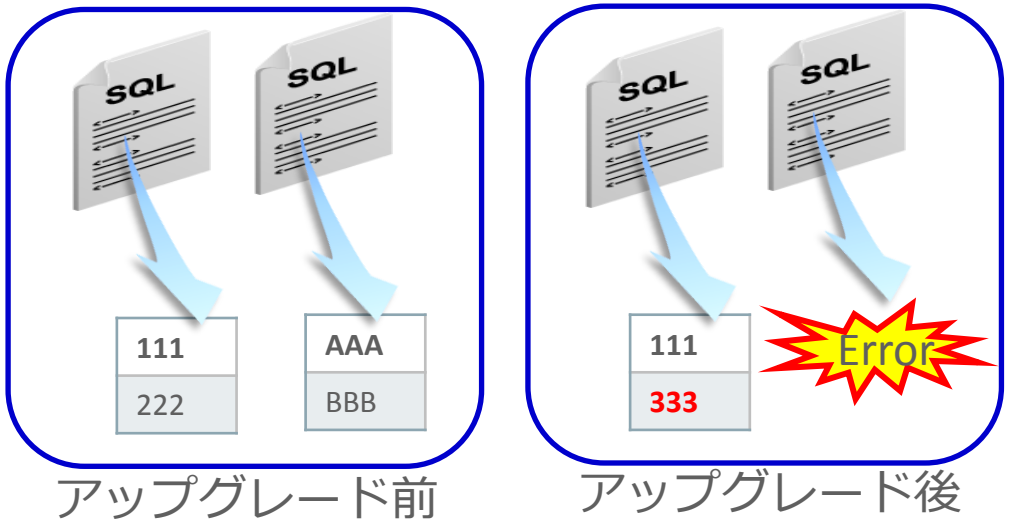


※上記図はフローをイメージ化したものであり、具体的に
それぞれのフェーズで想定される割合を表現したものではありません。

DBアップグレードの課題

③SQLの実行結果が変わる

今までエラーにはならなかったSQLがエラーになる可能性は？
不具合が解消されて、SQLの実行結果が変わる可能性は？

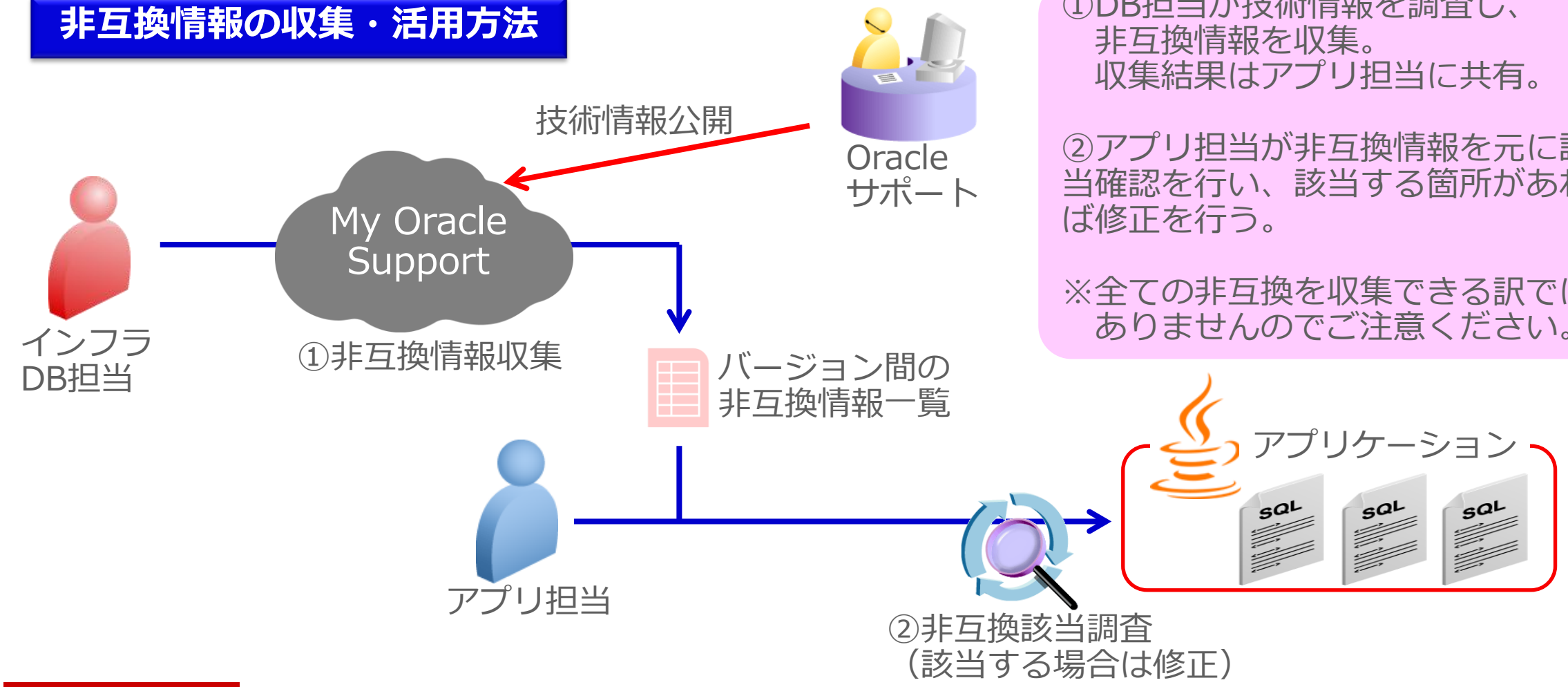


不具合の解消などにより実際に両方とも起こりえますが、全てのSQLに対して実行結果の比較を行うことはコスト的に困難な場合が多いと思われます。
このため、**弊社サポート部門が公開している技術情報から非互換に関する情報を収集し、アプリ側で該当調査することをお勧めします。**
なお、SPAを利用した場合、構文エラー等は実行時に確認できます。

DBアップグレードの課題

③SQLの実行結果が変わる

非互換情報の収集・活用方法



①DB担当が技術情報を調査し、
非互換情報を収集。
収集結果はアプリ担当に共有。

②アプリ担当が非互換情報を元に該当確認を行い、該当する箇所があれば修正を行う。

※全ての非互換を収集できる訳ではありませんのでご注意ください。

DBアップグレードの課題

課題と対処策のまとめ

	課題	対処策
①	テストすべきSQL数が不明	SQLをキャプチャーし、テストすべきSQLを把握することを推奨
②	実行計画の変化により性能が変わる	SPAを活用し、実行計画や性能の変動を確認することを推奨
③	SQLの実行結果が変わる	非互換情報を収集し、アプリ側で該当確認することを推奨

以降のページではSPAを活用してSQL単体性能を簡単に比較する方法と、その事例についてご紹介いたします。

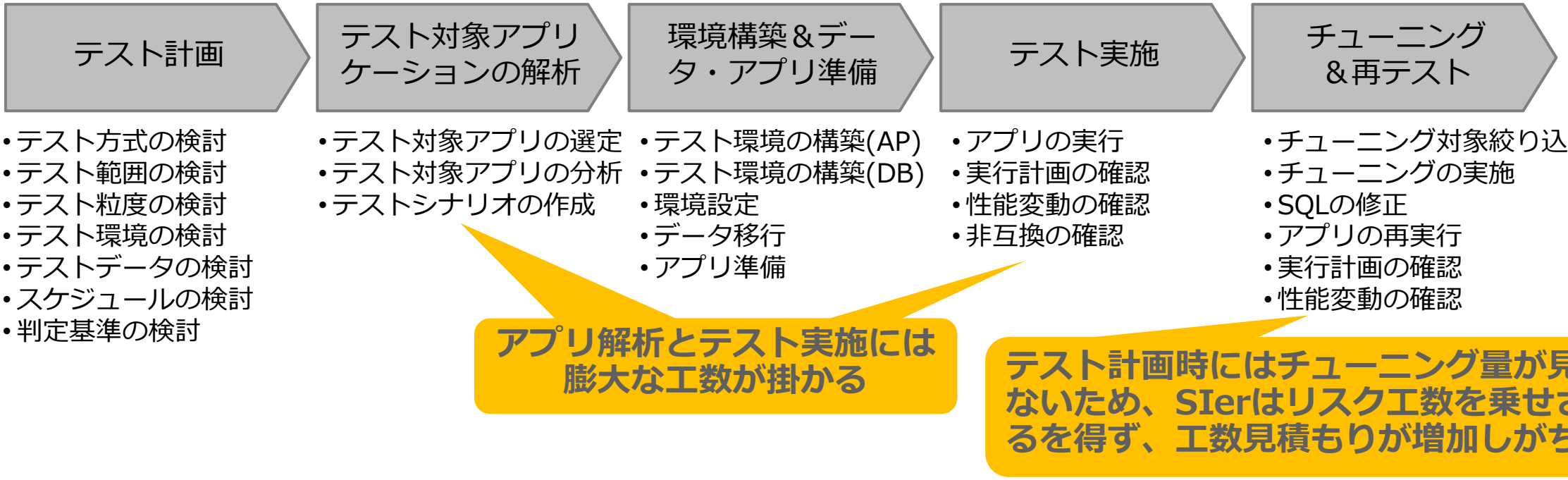
アジェンダ

1. DBアップグレードの課題
2. 一般的なテストとSPAを使用したSQL単体性能テストの違い
3. SPAを利用したSQL単体性能テスト手順概要
4. SPAを使用したSQL単体性能の評価事例(screen only)
5. まとめ
6. Q&A
7. Appendix
   1. SPAを利用したSQL単体性能テスト手順詳細
   2. SPAのTIPS

一般的なテストとSPAを使用したSQL単体性能テストの違い

SQL単体性能テストのプロセス比較（一般的な方法 vs SPA）

一般的なSQL単体性能テストのプロセス



全てエンジニアによる手作業で行われるためコストが掛かる。

一般的なテストとSPAを使用したSQL単体性能テストの違い

SQL単体性能テストのプロセス比較（一般的な方法 vs SPA）

SPAを利用したSQL単体性能テストのプロセス

テスト計画	テスト対象アプリケーションの解析	環境構築＆データ・アプリ準備	テスト実施	チューニング＆再テスト
• テスト方式の検討 • テスト範囲の検討 • ~~テスト粒度の検討~~ • テスト環境の検討 • テストデータの検討 • スケジュールの検討 • 判定基準の検討	• ~~テスト対象アプリの選定~~ • ~~テスト対象アプリの分析~~ • ~~テストシナリオの作成~~	• ~~テスト環境の構築(AP)~~ • テスト環境の構築(DB) • 環境設定 • データ移行 • ~~アプリ準備~~	• ~~アプリの実行~~ • 実行計画の確認 • 性能変動の確認 • 非互換の確認	• ~~チューニング対象絞り込み~~ • チューニングの実施 • SQLの修正 • ~~アプリの再実行~~ • 実行計画の確認 • 性能変動の確認

工数を大幅に削減可能

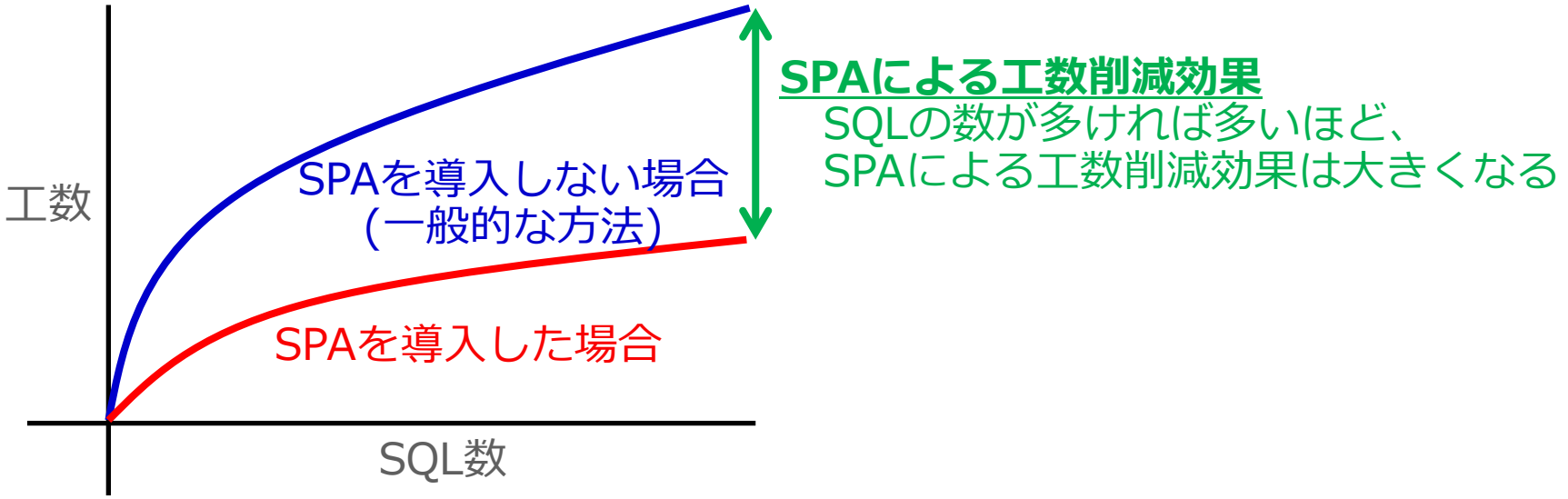
※薄グレー：SPAで削減できる作業
※オレンジ：SPAで簡素化できる作業
※グリーン：Tuning Advisorで簡素化できる作業
(別途Tuning Packが必要)

テスト実施までを短期間で行えるため、アップグレードの影響を素早く確認可能。チューニングフェーズを別契約にすることで、SIerの工数見積からリスクを削減できる。

SPAを利用することで作業を大幅に削減でき、コスト削減が見込める。

一般的なテストとSPAを使用したSQL単体性能テストの違い

SQL単体性能テストの工数比較（一般的な方法 vs SPA）



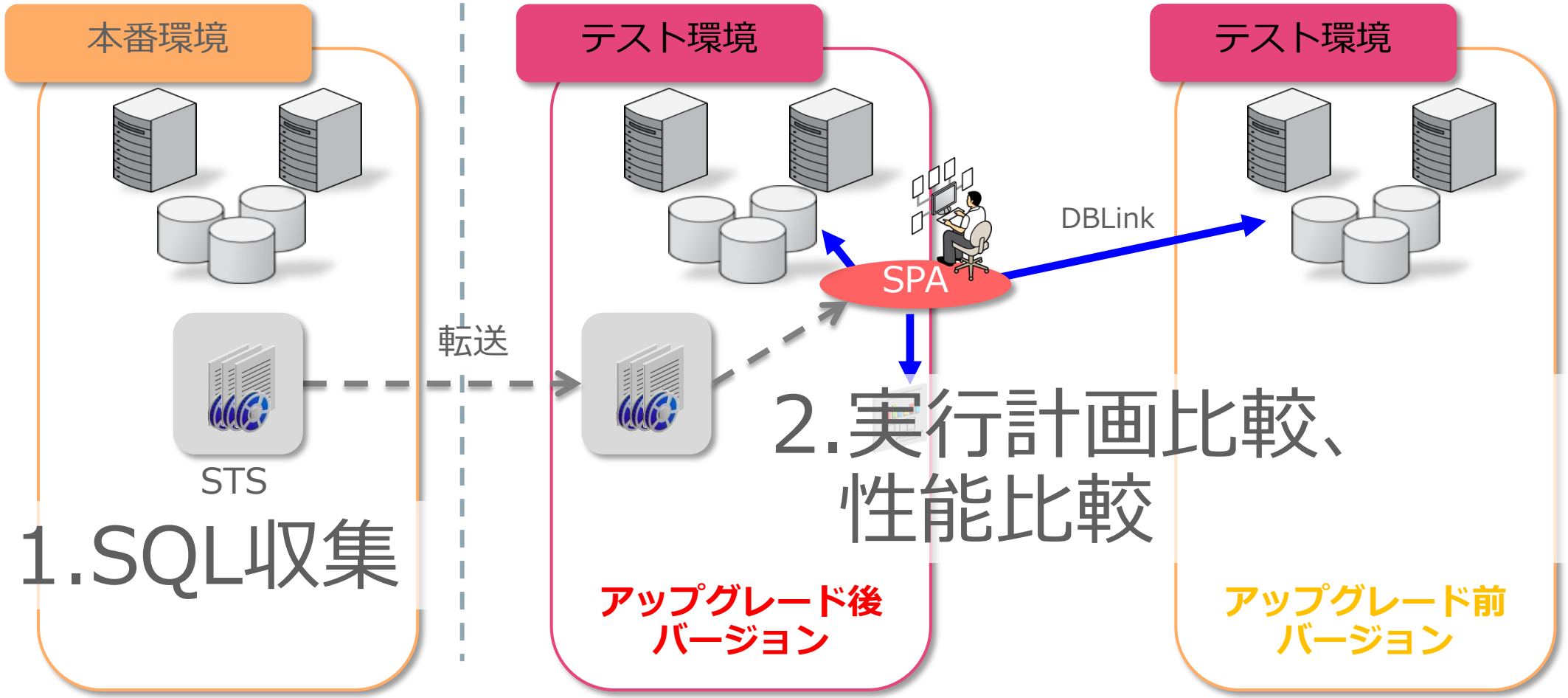
	SPAを導入しない場合（一般的な方法）	SPAを導入した場合
テスト工数	SQLに比例して増加	SQLが増えてもあまり増えない
テスト準備工数	アプリを全て実行するための準備に時間が掛かる	アプリの準備は不要で、短期間でテストを始められる
リトライ工数	再実行には多くの工数が掛かるためリトライが難しい	簡単に再実行が行えるため、トライ＆エラーを繰り返すことが可能

アジェンダ

1. DBアップグレードの課題
2. 一般的なテストとSPAを使用したSQL単体性能テストの違い
3. SPAを利用したSQL単体性能テスト手順概要
4. SPAを使用したSQL単体性能の評価事例(screen only)
5. まとめ
6. Q&A
7. Appendix
    1. SPAを利用したSQL単体性能テスト手順詳細
    2. SPAのTIPS

SPAを利用したSQL単体性能テスト手順概要

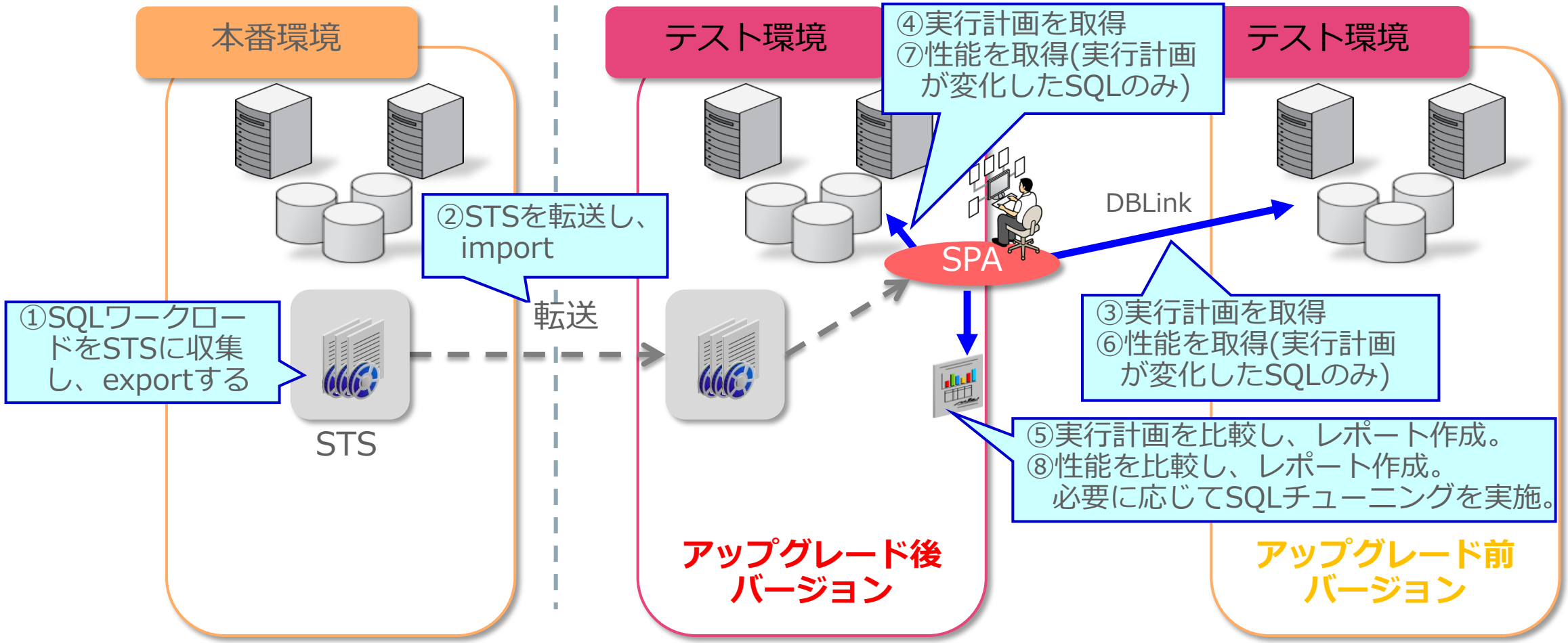
全体イメージ（概要）



※テスト環境は1つだけでもテスト可能です。

SPAを利用したSQL単体性能テスト手順概要

全体イメージ（概要フロー）



SPAを利用したSQL単体性能テスト手順概要

実行計画比較結果イメージ

- 実行計画比較レポートのサンプル

Report Summary

Projected Workload Change Impact:

Overall Impact : 0%
Improvement Impact : 0%
Regression Impact : 0%

SQL Statement Count

SQL Category	SQL Count	Plan Change Count
Overall	97	7
Unchanged	47	7
with Errors	50	0

トータルのSQL数や、実行計画が変化したSQL数、実行時エラーが発生したSQL数などが分かる

Top 47 SQL Sorted by Absolute Value of Change Impact on the Workload

object_id	sql_id	Impact on Workload	Execution Frequency	Metric Before	Metric After	Impact on SQL	Plan Change
251	90qys91313pjd	.04%	1	68	3	95.59%	y
268	c4572dkq4j9qt	.04%	1	68	3	95.59%	y
284	f6krnqfs5q0xr	.04%	1	68	3	95.59%	y
256	9k43zr1uannr6	-.04%	30	15	17	-13.33%	n
238	76cckj4yysvua	.03%	46	2	1	50%	n

具体的にどのSQLの実行計画が変化したかを特定できる

SPAを利用したSQL単体性能テスト手順概要

性能比較結果イメージ

- 性能比較レポートのサンプル

Report Details

SQL Details:

Object ID : 6
Schema Name : SQLDEMO
SQL ID : 90gys91313pjd
Execution Frequency : 1
SQL Text : SELECT * FROM tab1 WHERE c1 <= 9

Execution Statistics:

Stat Name	Impact on Workload	Value Before	Value After	Impact on SQL
elapsed_time	66.45%	.000307	.000103	66.45%
parse_time	-45.45%	.000154	.000224	-45.45%
cpu_time	0%	0	0	0%
user_io_time	0%	0	0	0%
buffer_gets	98.42%	190	3	98.42%
cost	95.59%	68	3	95.59%
reads	0%	0	0	0%
writes	0%	0	0	0%
io_interconnect_bytes	0%	0	0	0%
rows		9	9	

指定した性能項目(SQL実行時間等)で比較を行い、
・性能が変化していないSQL
・性能が改善したSQL
・性能が劣化したSQL
に分類した比較レポートを作成。

アップグレード前バージョンとアップグレード後バージョンで実行した際の性能比較結果が表示される。
許容できる変動量を超えたものをチューニング対象とする。

アジェンダ

1. DBアップグレードの課題
2. 一般的なテストとSPAを使用したSQL単体性能テストの違い
3. SPAを利用したSQL単体性能テスト手順概要
4. SPAを使用したSQL単体性能の評価事例(screen only)
5. まとめ
6. Q&A
7. Appendix
    1. SPAを利用したSQL単体性能テスト手順詳細
    2. SPAのTIPS

SPAを使用したSQL単体性能の評価事例

SCREEN ONLY

アジェンダ

1. DBアップグレードの課題
2. 一般的なテストとSPAを使用したSQL単体性能テストの違い
3. SPAを利用したSQL単体性能テスト手順概要
4. SPAを使用したSQL単体性能の評価事例(screen only)
5. まとめ
6. Q&A
7. Appendix
   1. SPAを利用したSQL単体性能テスト手順詳細
   2. SPAのTIPS

まとめ

RAT(SPA)を使うことで・・・

SQLの見える化が可能	SQLが把握できていないシステムでも、STSでSQLを収集することによりSQLの見える化が可能。
テスト工数を大幅に削減	SPAを使うことで、アップグレードによるSQLへの性能影響を簡単に確認可能。これにより、テスト工数の大幅削減が可能。
メンテナンスの影響を事前確認可能	運用中も同仕組みを活用することで、運用メンテナンスの影響（パッチ適用、索引追加、パラメータ変更など）を事前確認することが可能。

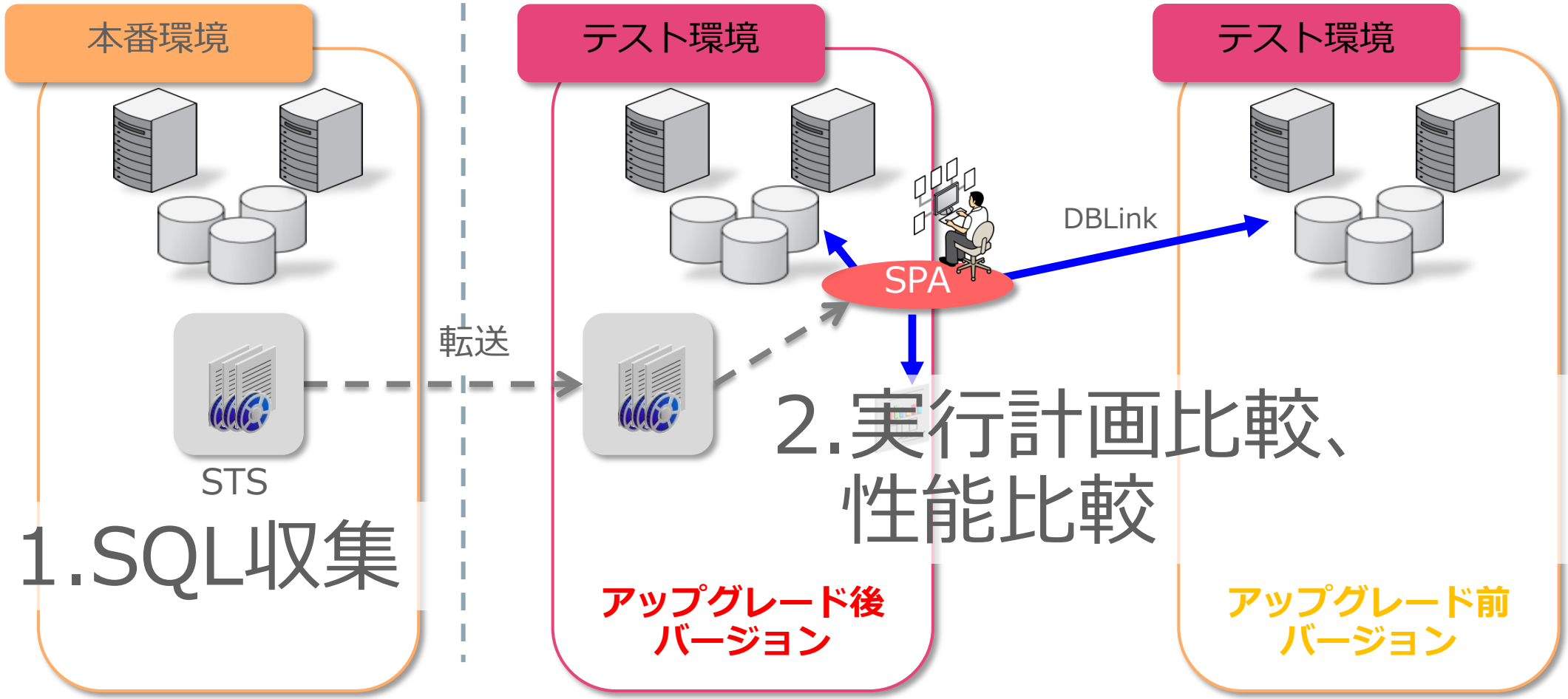
アジェンダ

1. DBアップグレードの課題
2. 一般的なテストとSPAを使用したSQL単体性能テストの違い
3. SPAを利用したSQL単体性能テスト手順概要
4. SPAを使用したSQL単体性能の評価事例(screen only)
5. まとめ
6. Q&A
7. Appendix
   1. SPAを利用したSQL単体性能テスト手順詳細
   2. SPAのTIPS

Appendix 1
SPAを利用したSQL単体性能テスト手順詳細

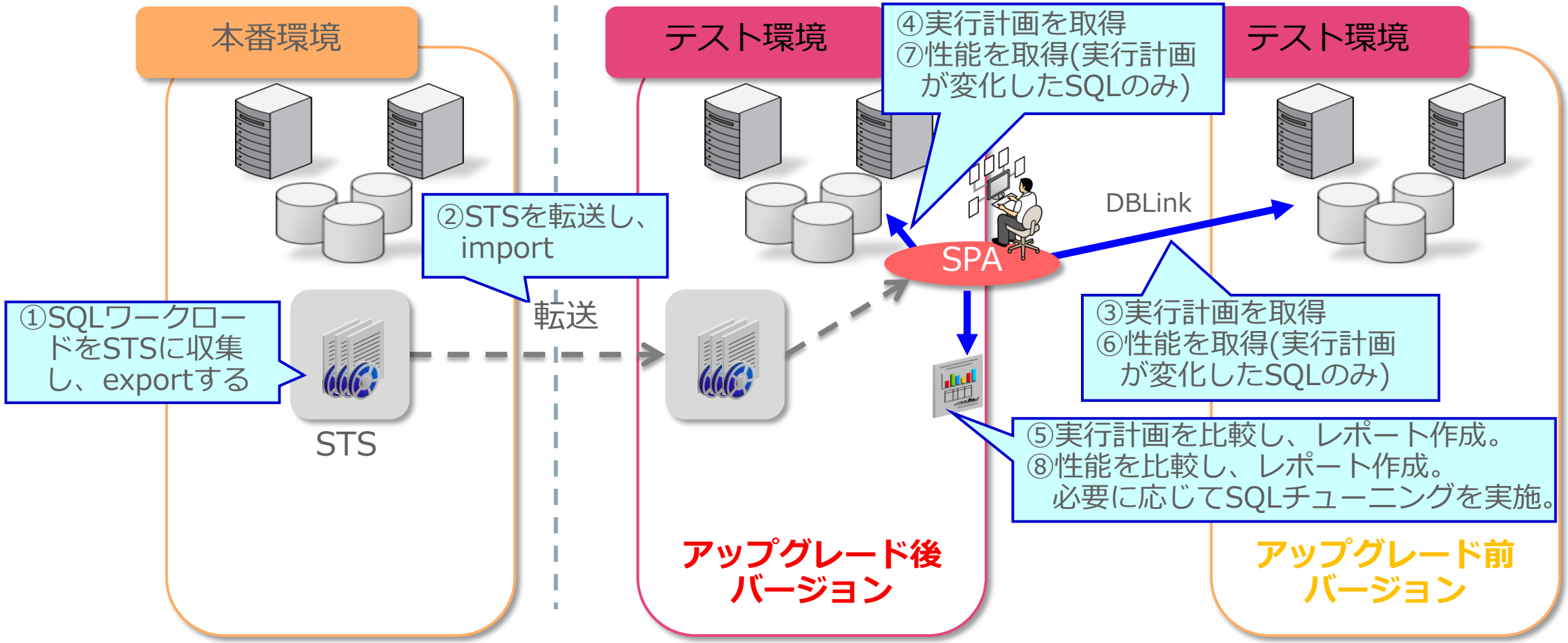
Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

全体イメージ（概要）



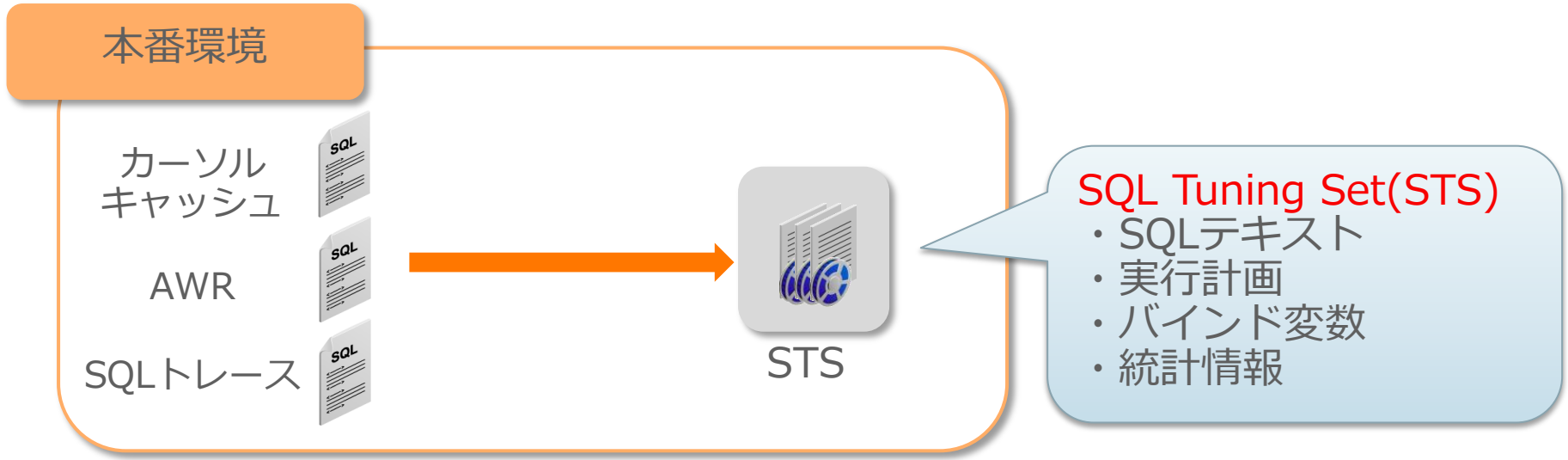
Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

全体イメージ（概要フロー）



Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

1. SQL収集／STSに格納



- テストに必要な本番環境のSQL情報を収集し、データベースオブジェクトとして保存。
- カーソルキャッシュ、AWR、SQLトレース等からSQL情報を収集可能。

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

1. SQL収集／STSに格納

- SQL収集元の選定
  SQL文の収集元を下記の3パターンから指定可能。

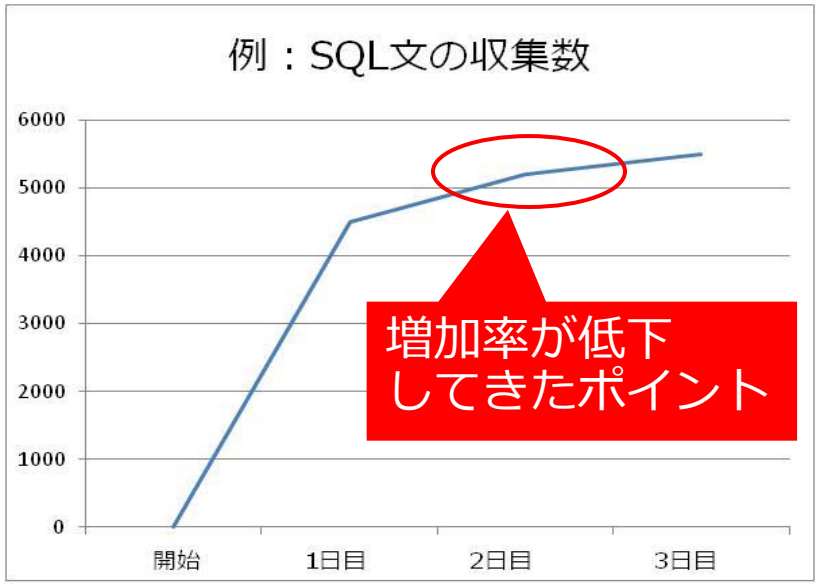
収集元	メリット	デメリット・課題
カーソル キャッシュ	・実際に本番環境上で実行されているSQL文をキャッシュから収集することが可能。 ・テストで用いるSQL文の網羅性を高めることができる。	・本番環境上のカーソルキャッシュからの収集となるため、少なからず負荷がかかる。
AWR	・本番環境に負荷を与えずにSQL文を収集することが可能。(厳密には、AWRから収集する軽微な負荷はあり)	・AWRに保存されるSQL文のみが収集対象となるため網羅性の点で課題が残る。
SQL トレース	・実際に本番環境上で実行されているSQL文を全て収集することが可能。 ・テストで用いるSQL文の網羅性を高めることができる。	・本番環境上でSQLトレースを取得する必要があり、負荷による影響が高い。

網羅性の高い「カーソルキャッシュ」の採用を推奨

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

1. SQL収集／STSに格納

- SQLの収集期間
　本番機でSQLのキャプチャーを行う際、どのくらいの期間キャプチャーすれば網羅性を高められるか分からない場合があります。
　その場合はSTSに収集したSQL数を定期的に確認し、**期間あたりの増加率が低下してきたポイントで、収集が終わったと判断**することができます。



※但し、リテラル値を使用したSQLが発行されている場合は、同じSQLであっても異なるSQLと見なされるため、SQL数が増加し続けます。

この場合は新規で収集されるSQLの傾向を確認し、収集済みのSQLと同類のSQLであれば収集完了と判断します。

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

1. SQL収集／STSに格納

- SQL文の収集負荷
  本番機でSQLのキャプチャーを行う際の負荷は下記の通りです。
  （11.1.0.6のDBで10秒ごとにカーソルキャッシュから読込んだ場合）

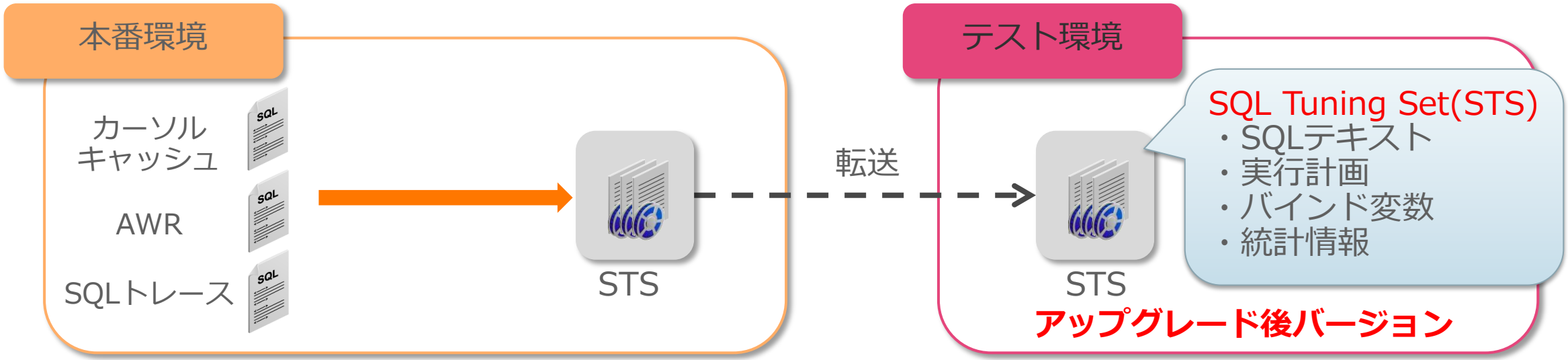
リソース	SQL収集時の影響
CPU	検証の結果、0.1%増加。(※)
メモリ	ほぼ影響なし。
ディスク	I/O影響は軽微。 SQLがSTSに格納される際、SYSAUX表領域へ微量のI/Oが発生。

（※）参照　Oracle Database 11g Oracle Real Application Testing 検証結果 ホワイトペーパー
（日本電気株式会社および日本オラクル株式会社による共同検証 ）

- SYSAUXを使用するサイズの目安としては、1SQLあたり5～10KBで見積もる。 ※あくまでも事例ベースでの目安です。システム特性によって異なります。

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

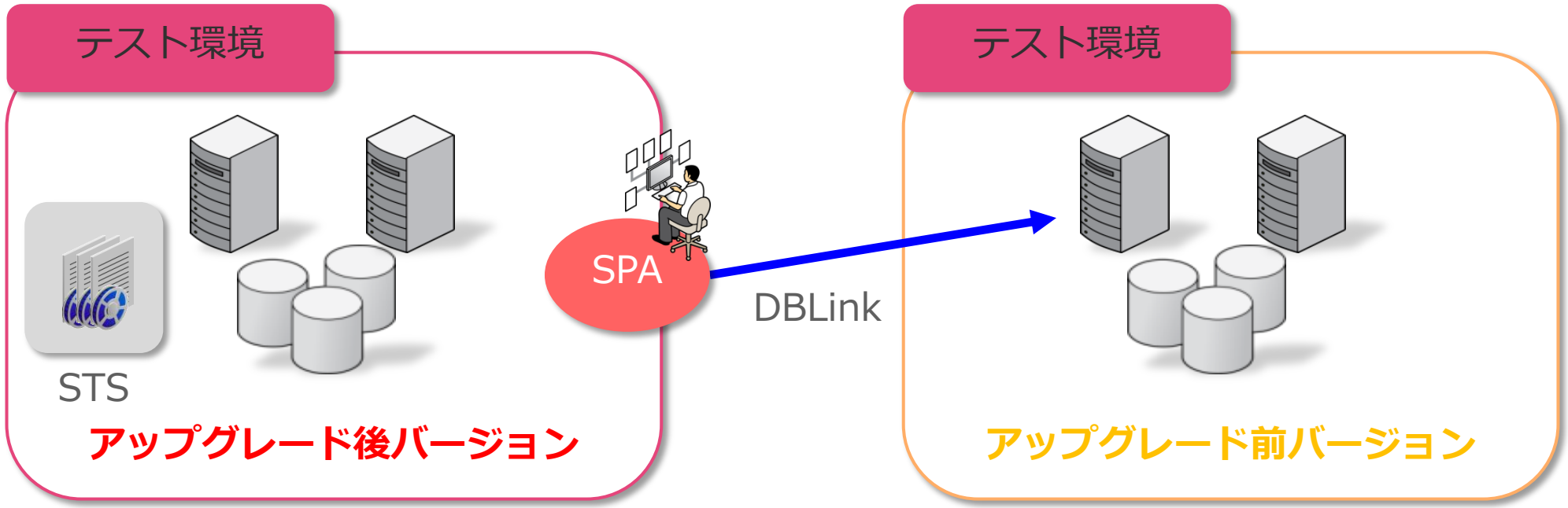
2. STSを転送



- 本番環境のSTSをDataPumpなどで転送し、テスト環境にインポート。
- STSはアップグレード後バージョンにインポートすることを推奨（手順3でアップグレード前バージョンに対しDBLink経由のSQL実行を行いますが、DBLinkは上位バージョンから下位バージョンへ実行する必要があるため）。

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

3. SPAによる実行計画取得（バージョンアップ前バージョン）



- テスト環境のアップグレード後バージョンでSPAを実行し、STSに格納されたSQLをDBLink経由でアップグレード前バージョンに対して実行する。

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

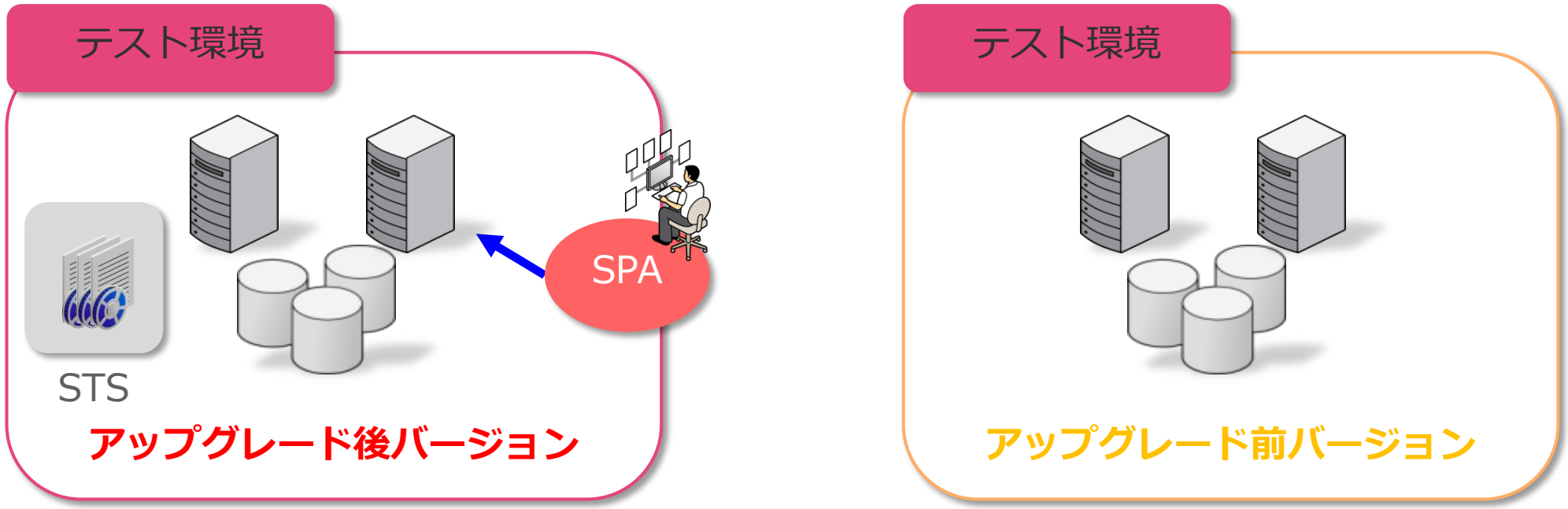
3. SPAによる実行計画取得（バージョンアップ前バージョン）

- 実行タイプ
  SQLを実行する際に実行計画だけを取得するか（EXPLAIN PLAN）、SQLを実行して性能も取得する（TEST EXECUTE）かを選択可能。
  EXPLAIN PLANに比べてTEST EXECUTEは時間が掛かるため、まずは実行計画だけを取得・比較し、実行計画が変化したものだけTEST EXECUTEで性能確認することを検討。

実行計画の生成(EXPLAIN PLAN)	SQLのテスト実行(TEST EXECUTE)
• SPAを介してSQL文の実行計画を生成する • EXPLAIN PLAN文と異なり、バインド値が 考慮され実際の実行計画が生成される	• SPAを介してSQL文をテスト実行する
まずはこちらを実行	**実行計画が変化したSQLのみ実行**

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

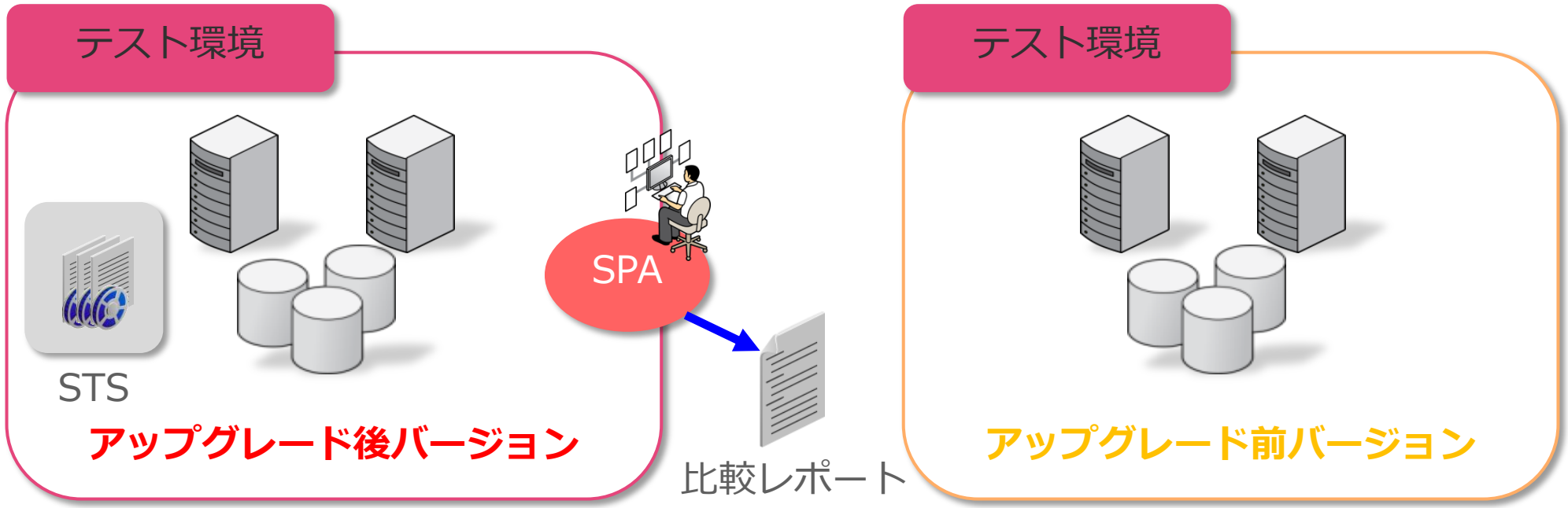
4. SPAによる実行計画取得（バージョンアップ後バージョン）



- テスト環境のアップグレード後バージョンに対し、SPAでSQLを実行する。

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

5. 実行計画比較 (分析レポート取得)



- アップグレード前バージョンとアップグレード後バージョンで取得した実行計画をもとに比較レポートを作成する。

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

5. 実行計画比較 (分析レポート取得)

- 実行計画比較レポートのサンプル

Report Summary

Projected Workload Change Impact:

Overall Impact : 0%
Improvement Impact : 0%
Regression Impact : 0%

SQL Statement Count

SQL Category	SQL Count	Plan Change Count
Overall	97	7
Unchanged	47	7
with Errors	50	0

トータルのSQL数や、実行計画が変化したSQL数、エラーが発生したSQL数などが分かる

Top 47 SQL Sorted by Absolute Value of Change Impact on the Workload

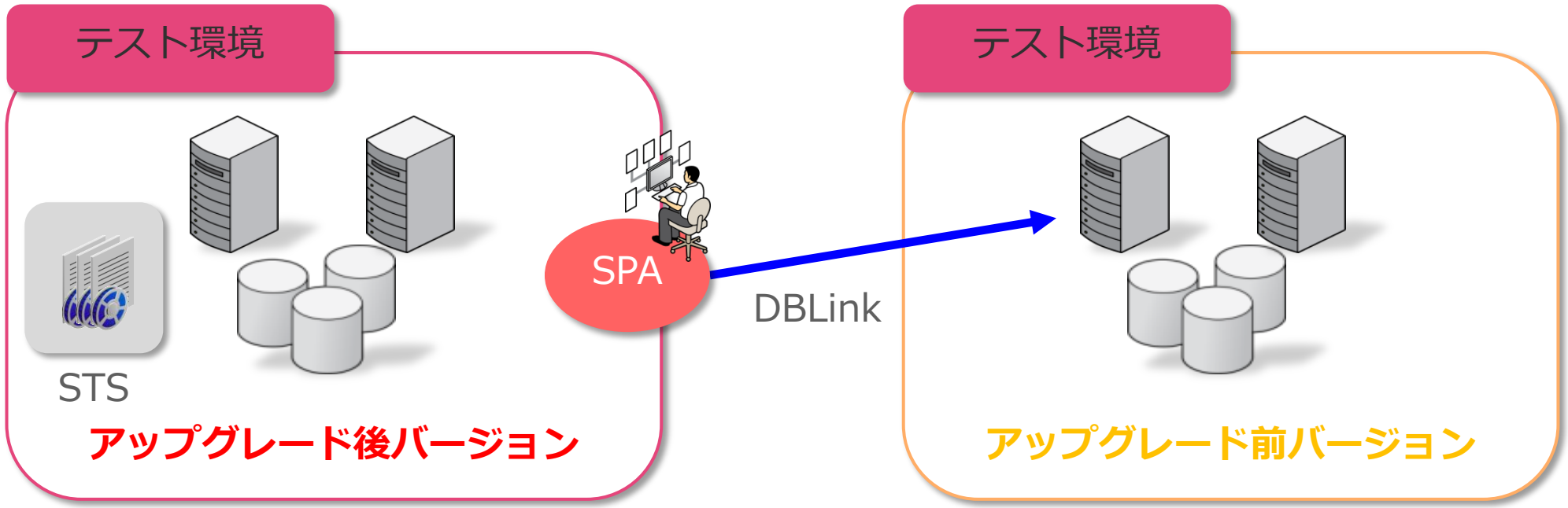
object_id	sql_id	Impact on Workload	Execution Frequency	Metric Before	Metric After	Impact on SQL	Plan Change
251	90qys91313pjd	.04%	1	68	3	95.59%	y
268	c4572dkq4j9qt	.04%	1	68	3	95.59%	y
284	f6krnqfs5q0xr	.04%	1	68	3	95.59%	y
256	9k43zr1uannr6	-.04%	30	15	17	-13.33%	n
238	76cckj4yysvua	.03%	46	2	1	50%	n

具体的にどのSQLの実行計画が変化したかを特定できる

- SQLの件数が多い場合はレポートサイズが大きくなりすぎるため、元表やビューを直接参照することも検討してください。

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

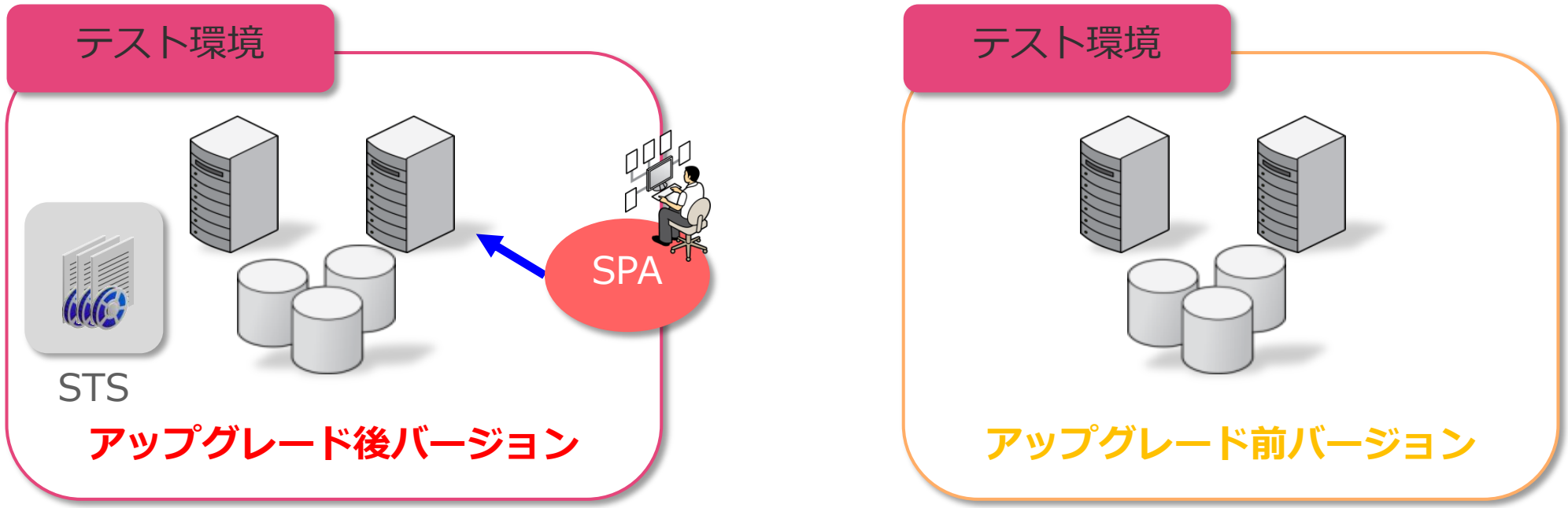
6. 実行計画が変化したSQLの性能を取得（バージョンアップ前バージョン）



- 実行計画が変化したSQLに対して性能を取得する。

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

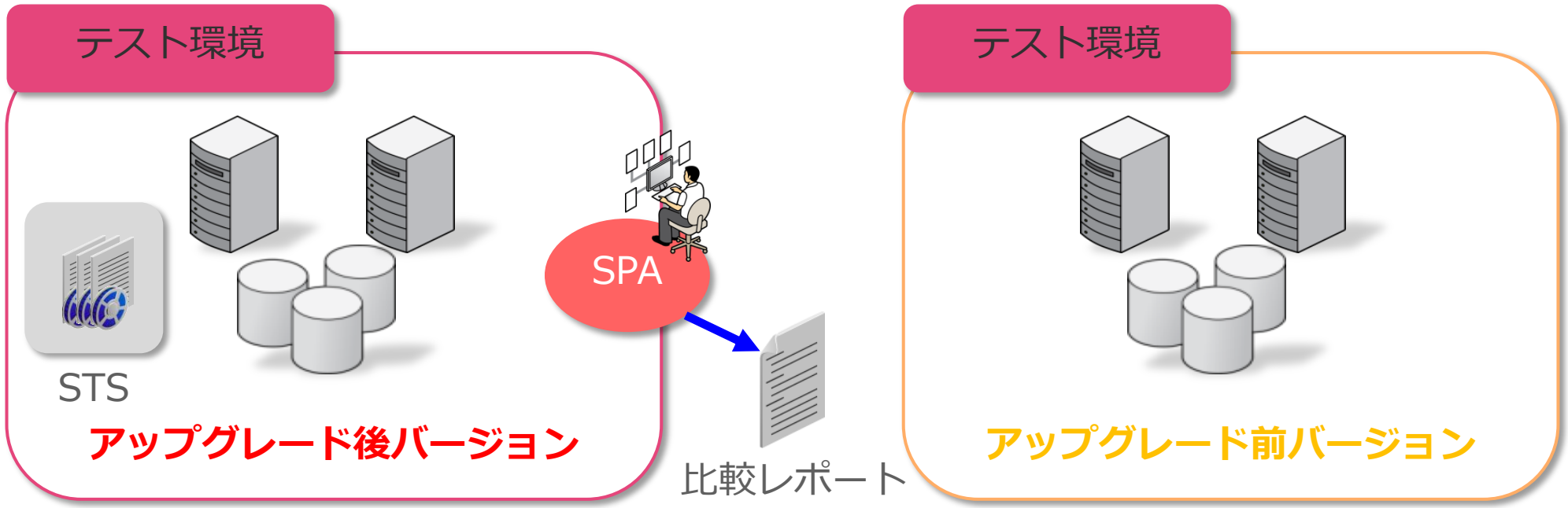
7. 実行計画が変化したSQLの性能を取得（バージョンアップ後バージョン）



- 実行計画が変化したSQLに対して性能を取得する。

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

8.チューニングが必要なSQLを特定し、対策を検討／実施



- アップグレード前バージョンとアップグレード後バージョンで取得した性能値をもとに比較レポートを作成する。

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

8.チューニングが必要なSQLを特定し、対策を検討／実施

- 比較できる性能項目

性能項目	備考
elapsed_time（デフォルト）	SQLの実行時間で比較。
cpu_time	SQLの解析、実行、フェッチに使用したCPU時間で比較。
buffer_gets	バッファ読み取りブロック数で比較。
disk_reads	ディスク読み取りブロック数で比較。
direct_writes	ダイレクト・パス書き込み数で比較。
optimizer_cost	オプティマイザ・コストで比較。
io_interconnect_bytes	ストレージへのI/Oバイト数で比較。
user_io_time	すべての行をフェッチするのに必要なI/O時間で比較。
parse_time	パース時間で比較。

比較する性能項目は外部影響を受けにくい「buffer_gets」がお勧め。
但し、性能要件を満たしていればよい場合は「elapsed_time」での比較も検討。

Appendix 1. SPAを利用したSQL単体性能テスト手順詳細

8.チューニングが必要なSQLを特定し、対策を検討／実施

- 性能比較レポートのサンプル

Report Details

SQL Details:

Object ID : 6
Schema Name : SQLDEMO
SQL ID : 90gys91313pjd
Execution Frequency : 1
SQL Text : SELECT * FROM tab1 WHERE c1 <= 9

Execution Statistics:

Stat Name	Impact on Workload	Value Before	Value After	Impact on SQL
elapsed_time	66.45%	.000307	.000103	66.45%
parse_time	-45.45%	.000154	.000224	-45.45%
cpu_time	0%	0	0	0%
user_io_time	0%	0	0	0%
buffer_gets	98.42%	190	3	98.42%
cost	95.59%	68	3	95.59%
reads	0%	0	0	0%
writes	0%	0	0	0%
io_interconnect_bytes	0%	0	0	0%
rows		9	9	

- 性能比較を行う前に、予めどの程度の性能変動までは許容するかを決めておくことを推奨。
- オンラインとバッチとでは許容する変動率を変えた方がよい。

アップグレード前バージョンとアップグレード後バージョンで実行した際の性能比較結果が表示される。
許容できる変動量を超えたものがチューニング対象となる。

Appendix 2
SPAのTIPS

Appendix 2. SPAのTIPS

①SQLキャプチャー時に業務SQLではないSQLを除外する方法

　デフォルトの状態でSQLをキャプチャーすると全てのスキーマのSQLがキャプチャーされてしまうため、Oracleの内部SQL等もキャプチャーされてしまう。

業務SQL以外は評価する必要がないためフィルタで除外し、SPAのSQL実行時間を短縮させることを推奨

Appendix 2. SPAのTIPS

①SQLキャプチャー時に業務SQLではないSQLを除外する方法

- SQLキャプチャー時にbasic_filterでPARSING_SCHEMA_NAMEを指定することにより、業務ユーザのSQLだけを抽出できる。
- 業務ユーザに限定してもOracleが内部的に実行するSQLが含まれてしまうため、これらもbasic_filterで除外して実行時間の短縮化を図ることを推奨。

Appendix 2. SPAのTIPS

①SQLキャプチャー時に業務SQLではないSQLを除外する方法

SQLキャプチャー時の除外SQLキーワード	除外理由
/* DS_SVC */	チューニングアドバイザのSQLであるため除外。
/* OPT_DYN_SAMP */	ダイナミックサンプリングのSQLであるため除外。
/* MV_REFRESH (DEL) */	MVIEWリフレッシュのSQLのため除外。
INSERT /*+ BYPASS_RECURSIVE_CHECK	MVIEWリフレッシュのSQLのため除外。
/* SQL Analyze(	オプティマイザ統計取得のSQLであるため除外。
TBLORIDX$PART$NUM	パーティションの統計情報の取得のSQLであるため除外。
SELECT s.snap_id,to_char(BEGIN_INTERVAL_TIME,'YYYYMMDDHH24MI') snap_time	AWR関連の表への検索SQLであるため除外。

Appendix 2. SPAのTIPS

①SQLキャプチャー時に業務SQLではないSQLを除外する方法

SQLキャプチャー時の除外Module	除外理由
Data Pump Master	Data pumpを実施した場合に発行される再帰的SQLを場外可能なため。
Data Pump Worker	Data pumpを実施した場合に発行される再帰的SQLを場外可能なため。

SQLキャプチャー時の除外 PLAN_HASH_VALUE	除外理由
plan_hash_value != 0	DDLや、単純なINSERT VALUESはPLAN_HASH_VALUEが「0」で発行される。それらのSQLを除外するため。

Appendix 2. SPAのTIPS

②1つのSQL_IDに複数の実行計画が存在する場合のSQL単体性能比較方法

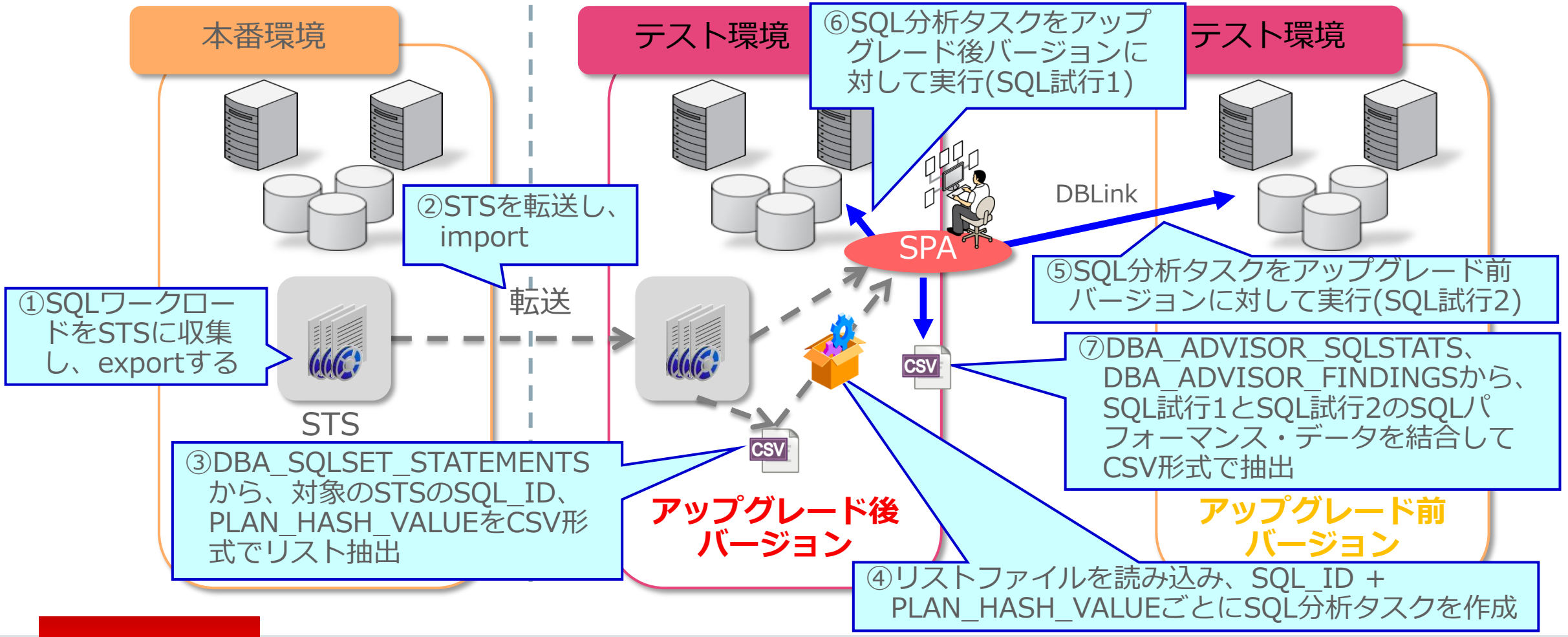
　入力値のバリエーションにより最適な実行計画が変化する場合は1つのSQLに複数の実行計画が存在する場合があるが、SPA標準機能ではどれか1つの入力値でのみテストされる。

テストの網羅性を上げるため、実行計画が変わる全ての入力値バリエーションに対し、実行計画変化の有無や性能変動の有無を確認しましょう。

Appendix 2. SPAのTIPS

②1つのSQL_IDに複数の実行計画が存在する場合のSQL単体性能比較方法

- SQL_ID＋PLAN_HASH_VALUE毎にSPA実行する手順



Integrated Cloud

Applications & Platform Services


ORACLE®