

Oracle Developer Day 2026

最新OCI基盤の進化と 新IAM移行事例

株式会社野村総合研究所

朝日英彦
阿部祐希
落合俊貴

2026年5月21日

NRI

Envision the value,
Empower the change



NRIの会社紹介

会社概要

社名	株式会社野村総合研究所
英文社名	Nomura Research Institute, Ltd.
創業日	1965年4月1日
資本金	25,655,413,800円
従業員数	7,645人 (NRIグループ16,679人)
本社	〒100-0004 東京都千代田区大手町1-9-2 大手町フィナンシャルシティ グランキューブ
上場証券取引所	プライム市場／証券コード：4307

2025年3月31日現在

4つの事業領域

NRIグループはコンサルティングやさまざまなITソリューションの提供を通じて、社会や産業を確かな明日へ導くとともに、世界中のお客様と新しい価値を共創しています。

01 | コンサルティング CONSULTING

創業以来、シンクタンクとしての深い知見と先見性を活かし、官民の様々な分野で戦略策定・政策立案を支援してきました。また、政策・産業・事業・技術への深い理解とお客様との対話を通じ、課題解決に向けた施策を提案し、伴走しています。

AIをはじめとする技術革新が加速し、社会や市場の変化が予測困難となる今、ビジネスを次のステージへと導くには、先見性に裏付けられたマネジメントコンサルティング、先進技術で事業・業務革新を加速するシステムコンサルティング、それらを統合する実行力が不可欠です。未来を見据え、今を変えることで、お客様の最良のパートナーであり続けることを目指します。



02 | 金融ITソリューション FINANCIAL IT SOLUTIONS

NRIグループは、金融ビジネスを取り巻く環境変化を高い洞察力で捉える研究員やコンサルタント、ITソリューションサービスを提供するビジネスアナリストやデジタル人材の連携によって次世代ソリューションを提供し続け、金融機関の事業継続を多方面から支えています。近年は、金融機関や政策当局、異業種プレーヤーなどとの価値共創により、金融機能の変革に取り組んでいます。金融は、社会の重要インフラです。進化し続けるデジタル技術との相性を常に考えながら、安定かつ先進的な社会インフラを実現し、社会課題に果敢にチャレンジしていきます。



03 | 産業ITソリューション INDUSTRIAL IT SOLUTIONS

産業分野の業界トップ企業のビジネスパートナーとして、コンサルティングからシステム開発や運用まで、一貫したサービスを提供しています。

コンサルタントとエンジニアが共同でお客様のビジネス環境やデータを分析しながら、最適なITソリューションを提供しています。また、コンサルティング部門から運用部門まで、NRIグループのリソースをインテグレーションし、お客様のデジタル戦略を柔軟かつスピーディーに実現します。

長年にわたってミッションクリティカルなシステムを構築・運用してきた経験と実績で、これからもお客様の事業インフラとしてのシステム基盤を支えていきます。



04 | IT基盤サービス IT PLATFORM SERVICES

IT技術の革新が加速する中、巨大化・複雑化が進むITシステム基盤の重要性が増しています。NRIグループは先進的な技術を見通し、戦略的にサービスやソリューションに取り入れ、お客様の成長や変革の実現をサポートします。先進技術の調査・研究やAI技術の活用も積極的に実施しています。また、マルチクラウドを含むシステム全体を運営するマネージドサービスやお客様の働く環境を創り出すデジタルワークプレイス事業などを展開しています。さらに、高度化するサイバー脅威に対応するデジタルトラスト事業やお客様が直面するセキュリティ課題の解決、総合的なセキュリティレベルの向上を支援しています。



プレゼンター紹介



野村総合研究所
Oracle ACE Pro
朝日 英彦



野村総合研究所 クラウドサービス推進部
アソシエイトテクニカルエンジニア
阿部 祐希



野村総合研究所 証券基盤サービス一部
アソシエイトテクニカルエンジニア
落合 俊貴

01

NRIのOCI活用への取り組み

02

OCIの新IAM「Identity Domains」への移行事例

03

Exadata&Compute性能対策実践ノウハウ

1. NRIのOCI活用への取り組み

1. NRIのOCI活用への取り組み








自己紹介

■朝日 英彦（Oracle ACE Pro）

- 金融業界向けのシステム基盤設計・構築等、特にマルチクラウド環境におけるミッションクリティカルシステムのプロジェクト推進に従事。現在は保険業界向けのシステムモダナイズやクラウドマイグレーションなどを推進
- 「マルチクラウドデータベースの教科書」共著者



■ Oracle関連資格等

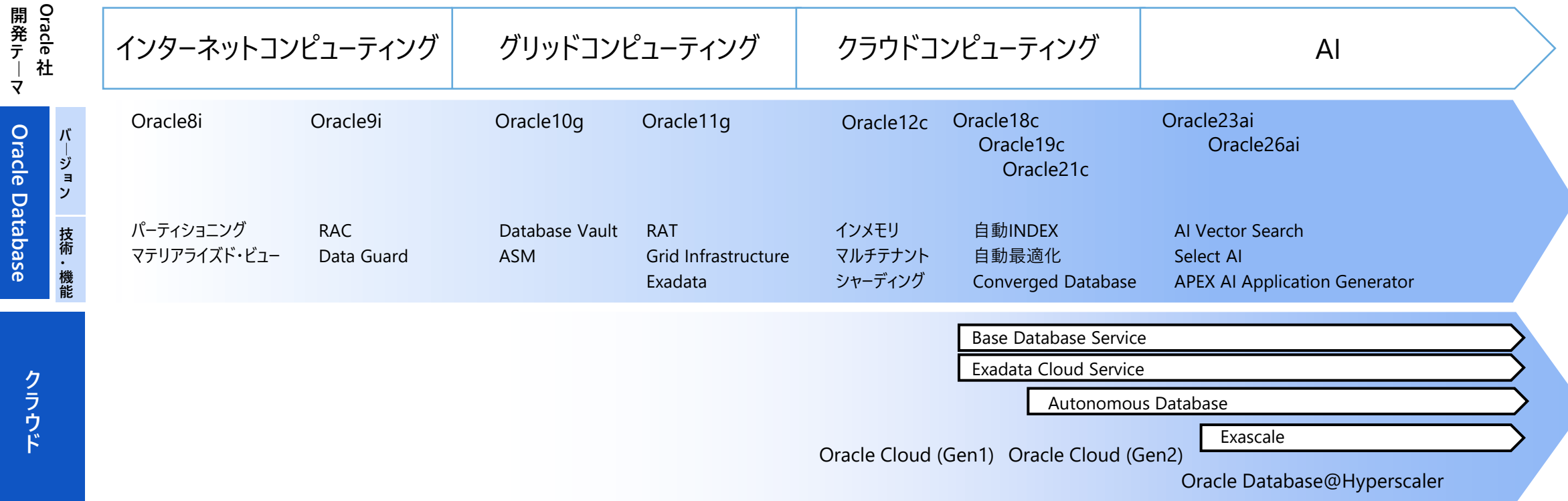
- Oracle ACE Pro (Database) 
- 2025 OCI Top Partner Engineer
- Oracle Cloud Infrastructure 2024 Architect Professional 
- Oracle Cloud Infrastructure 2024 Generative AI Certified Professional 
- Oracle AI Vector Search Certified Professional 
- Oracle Database Cloud Administrator 2023 Certified Professional 
- Oracle Autonomous Database Cloud 2023 Certified Professional 
- Oracle Base Database Services 2023 Certified Professional 
- My Oracle Support Community Most Valued Contributor 2022~ 

■ クラウド・データベース資格等

- Japan AWS Top Engineer 2022~ 
- AWS Community Builder 2023~ 
- AWS Certified Database – Specialty 
- AWS Certified Solutions Architect Professional 
- Microsoft Certified: Azure Database Administrator Associate 
- Microsoft Certified: Azure Solutions Architect Expert 
- Google Cloud Certified Professional Cloud Database Engineer 
- Google Cloud Certified Professional Cloud Architect 
- 情報処理技術者（データベース）

1. NRIのOCI活用への取り組み

NRIのOCIとOracle DBに対する取り組み



NRIの取り組み

オンプレ時代からミッションクリティカルを支えるデータベースとしてOracle Databaseを採用。RACやData Guardといった技術を活用し、システムを運用。

自社の金融系システムにて、高度化する業務要件に応え、さらなる信頼性とパフォーマンスを求めてExadataを採用。以降、複数システムで活用を続けノウハウを蓄積。

プライベートクラウドで稼働していた自社システムで、Exadataを含むOracle DBの移行先としてOCIを採用。OCIサービスを活用しクラウドに最適化された運用への移行を推進。

Oracle Alloy 基盤のソブリン金融AIプラットフォーム「YUIAI」を構築。
AIによるデータ活用を安全安心に行うため、Oracle DBとOCIのさらなる活用推進に加え、セキュリティへの取り組みを強化。

1. NRIのOCI活用への取り組み

NRIのOCIとOracle DBに対する取り組み

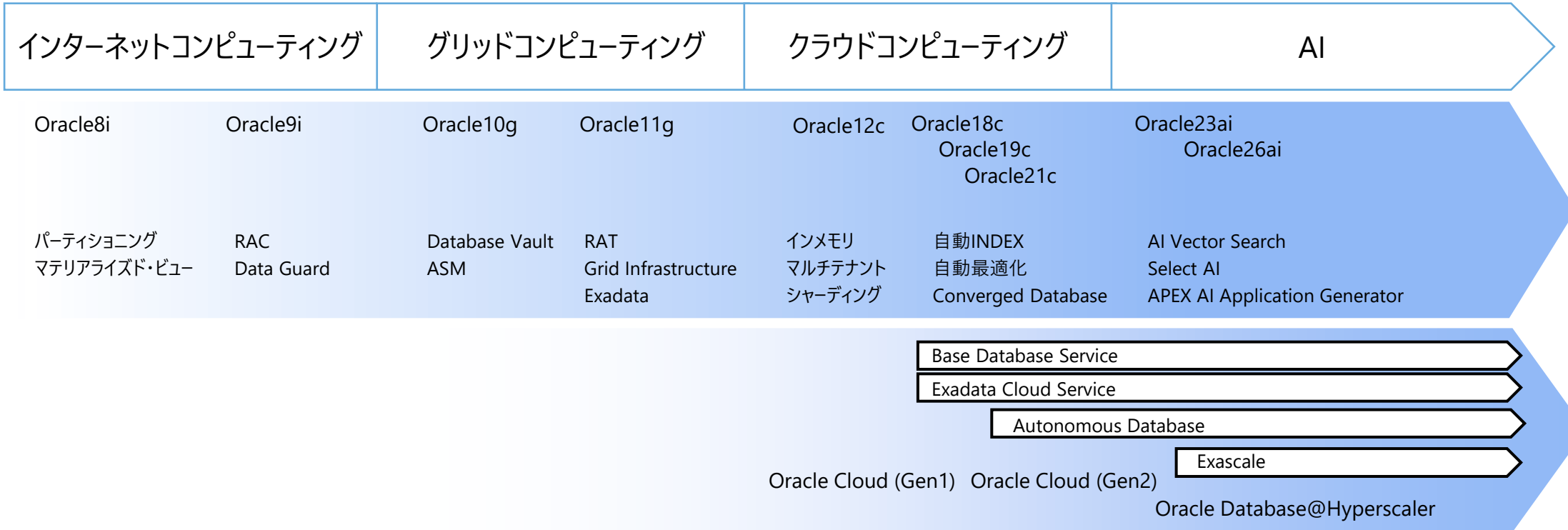
Oracle社
開発テーマ

Oracle Database

バージョン

技術・機能

クラウド



本セッションで紹介

オンプレ時代からミッションクリティカルを支えるデータベースとしてOracle Databaseを採用。RACやData Guardといった技術を活用し、システムを運用。

自社の金融系システムにて、高度化する業務要件に応え、さらなる信頼性とパフォーマンスを求めてExadataを採用。以降、複数システムで活用を続けノウハウを蓄積。

プライベートクラウドで稼働していた自社システムで、Exadataを含むOracle DBの移行先としてOCIを採用。
OCIサービスを活用しクラウドに最適化された運用への移行を推進。

Oracle Alloy 基盤のソブリン金融AIプラットフォーム「YUIAI」を構築。
AIによるデータ活用を安全安心に行うため、Oracle DBとOCIのさらなる活用推進に加え、セキュリティへの取り組みを強化。

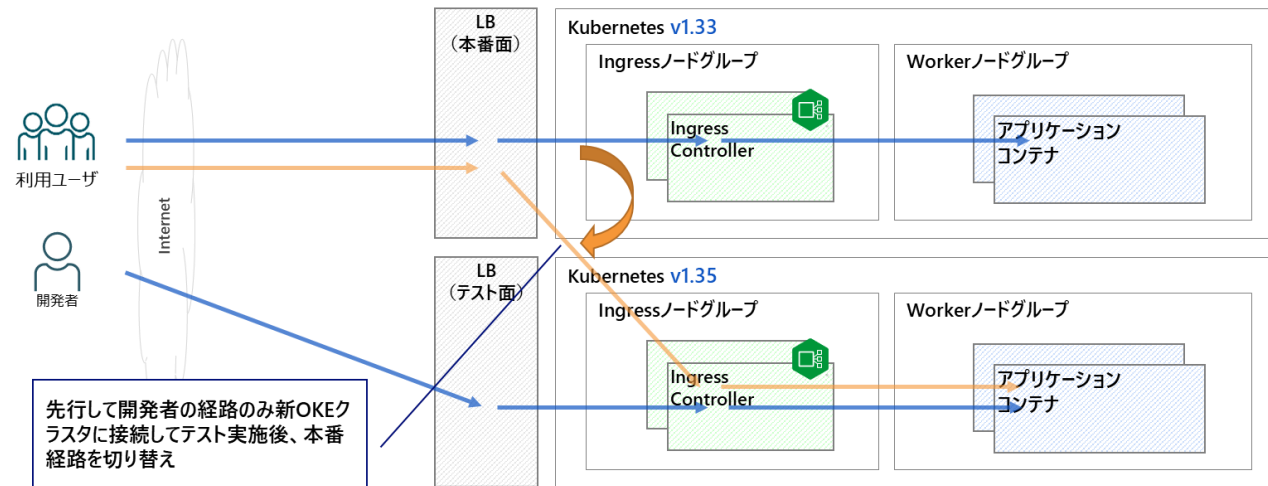
NRIの取り組み

1. NRIのOCI活用への取り組み

OKEバージョンアップ標準化

■標準化チームで事前に新しいバージョンの検証を実施

- 各プロジェクトでも最終確認を実施



新旧OKEクラスタを並行稼働させ、業務影響を与えずバージョンアップを実施

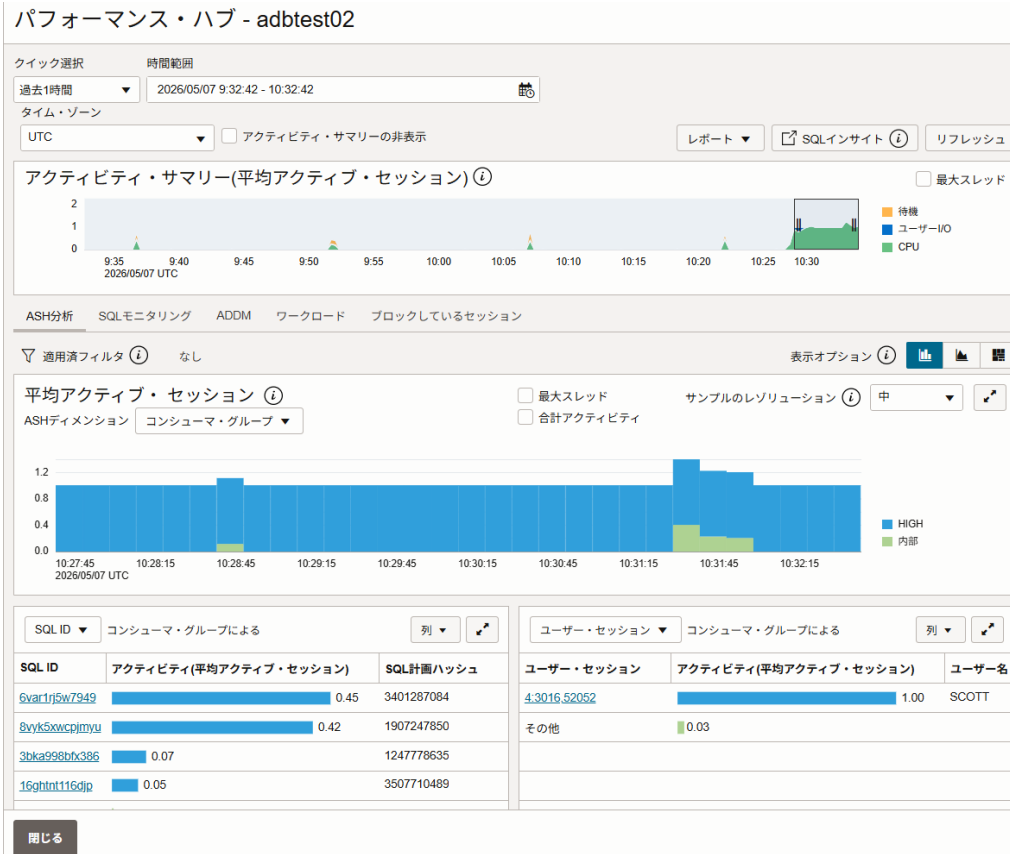
アプリケーションレイヤーにおいては、OCI上に展開されたKubernetes環境（OKE：Oracle Container Engine for Kubernetes）を活用し、再構築と検証サイクルの高速化が図られました。これにより、従来は手間のかかった修正・反映作業も即時対応が可能となり、開発現場からは「スピード感がまったく違う」との声が上がっています。

事例出典：https://atlab.nri.co.jp/cases/20251216_oci/

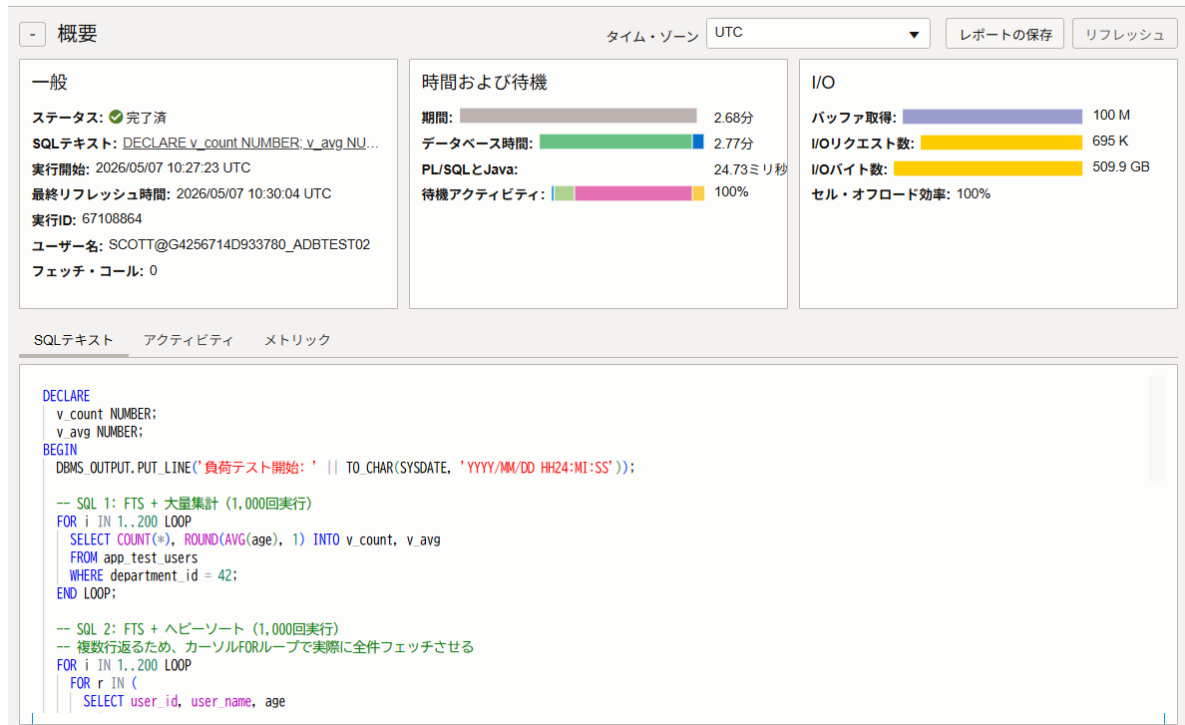
おまけ：DBAも楽したい！

今までの課題：DBのチューニングとかの相談相手がいない

- AWR/ADDM、OCI上だとパフォーマンスハブなど、いろいろと状況を確認できるツールはそろってきていました。一方で『全体のサマリを作る』とか『この推奨されているチューニング案を採用した場合の影響ってどうなるだろう?』というような相談ができる相手が限られていました

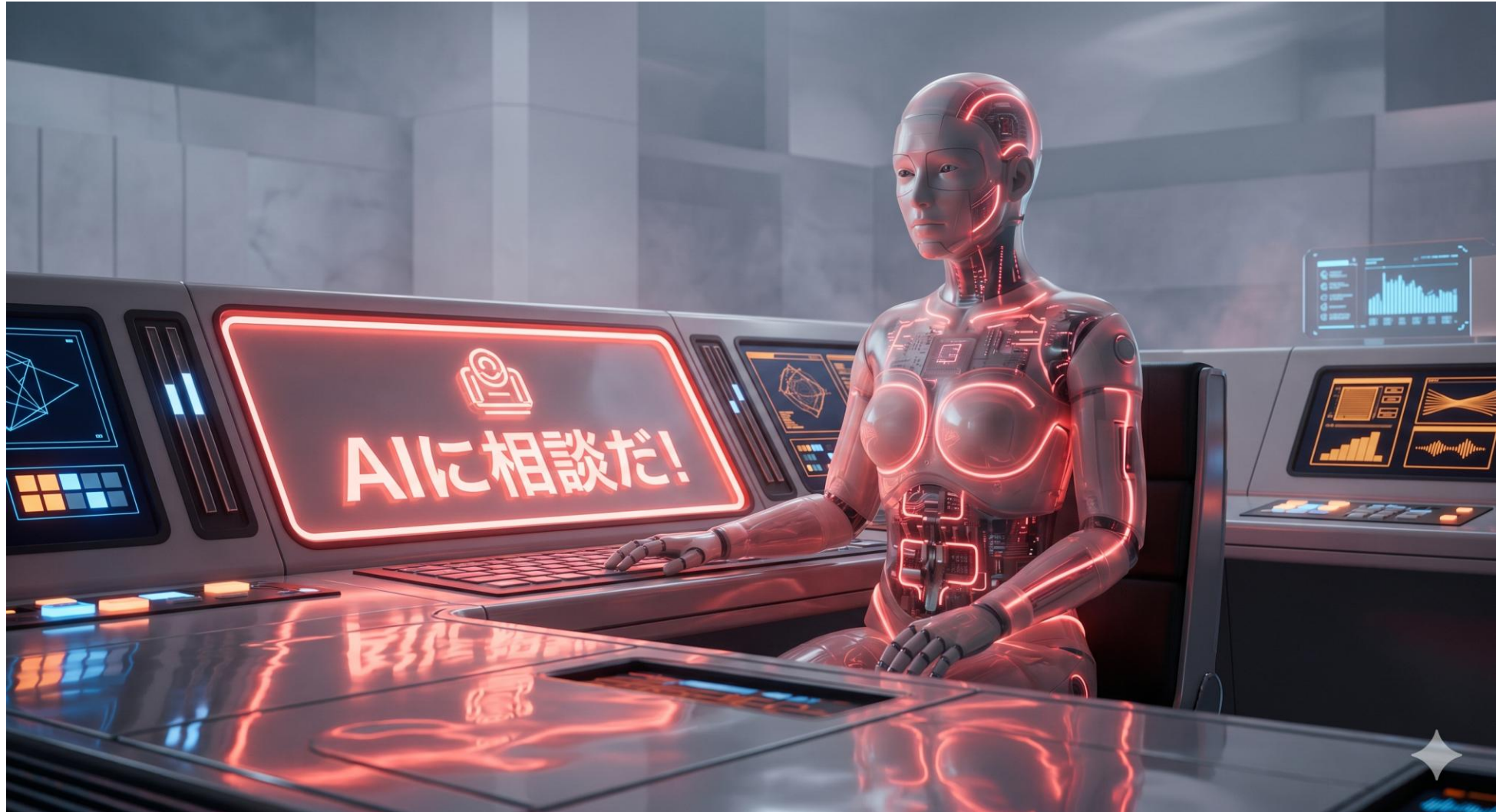


←戻る パフォーマンス・ハブ: SQL 0rngmdjpt3dctのリアルタイムSQLモニタリング - adbtest02



おまけ

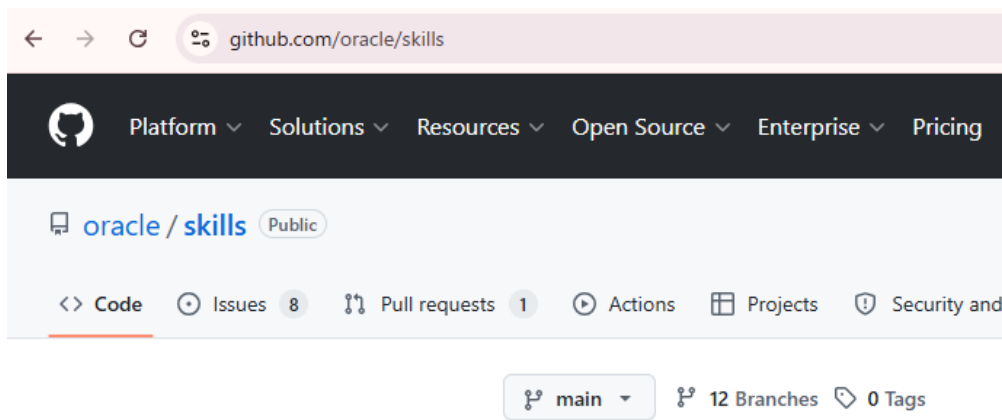
解決案：AIに相談だ！



oracle / skills

■ oracle / skills が公開されています

- Database の ADMIN業務のほか、チューニングなども



- ADMIN
- Agent
- Monitoring
- Performance
- Security
- ...

重要なセクションを読む

1. レポートヘッダー

データベースのバージョン、インスタンス名、ホスト、CPU数、スナップショット期間を確認します。結論を出す前に、必ずこれらの情報が対象環境と一致していることを確認してください。

2. ロードプロファイル

主要指標の1秒あたりおよびトランザクションあたりのレートを表示します。

メトリック	それはどういう意味か
DB時間	平均アクティブセッション数（経過秒数で割った値）
DB CPU	1秒あたりの実際のCPU消費量
サイズをやり直す	書き込み負荷の高いワークロード指標
論理読み取り	バッファキャッシュI/O
ブロックの変更	DML活動
物理的な読み取り	実際のディスクI/O
ハードパース	カーソル再利用の問題
解析する	解析呼び出しの総数（ハード+ソフト）
ログイン	接続の変動

ハードパース速度が1秒あたり100回を超える場合は、ほぼ間違いなくバインド変数の欠落または接続プールの問題を示しています。

※Chromeプラグインによる日本語訳

おまけ

AWRの解析を依頼

■ Claude Code + oracle / skill

- Claude Codeは LiteLLM 経由で利用



AWRの解析
を行って！

AWR 解析サマリー

基本情報

項目	値
DB Name	A TEST
Instance	A TEST
Release	Oracle 19.24.0.0.0
CDB	YES (PDB: PDTEST)
RAC	NO
Host	testdb01 (Linux x86 64-bit)
CPU / Cores	4 CPUs / 2 Cores
Memory	31.06 GB
解析対象期間	2026-04-22 23:00 ~ 2026-04-23 03:00
Snap ID 範囲	10138 ~ 10142

総合評価

期間	DB Time (mins)	AAS	AAS/CPU	評価
23:00 ~ 00:00 (ベースライン)	0.73	0.01	0.003	● 正常
00:00 ~ 01:00	1,034.72	17.2	4.3	● 要対応
01:00 ~ 02:00	1,707.40	28.4	7.1	● 要対応
02:00 ~ 03:00	2,468.50	41.2	10.3	● 要対応

00:00 以降に急激なパフォーマンス劣化が発生。深夜バッチ起因の CPU 飽和と判断。

00:00移行に急激なパフォーマンス劣化が発生。深夜バッチ起因の CPU 飽和と判断。

おまけ

チューニング案の提示



チューニング案
出して！

ちょっと
怖いな・・・

チューニング考察: SQL <SQL_ID> 全5案 メリット・デメリット

対象情報

項目	値
DB Name	<DB名>
解析対象期間	2026-04-22 23:00 ~ 2026-04-23 03:00
対象 SQL	<SQL_ID> (<DAOクラス>.<メソッド名>)
参照 AWR サマリー	20260423_<DB名>_snap10138-10142_summary.md

対応案 一覧

案A: NOT EXISTS → MINUS に書き換え

	内容
メリット	Optimizer が HASH 系結合を選びやすくなり NESTED LOOPS ANTI を回避できる可能性が高い
デメリット	NOT EXISTS と MINUS は NULL の扱いが異なる。業務データで等価性の確認が必要
デメリット	SELECT リストの列・型を厳密に揃える必要があり書き換えコストが中程度
業務要件への影響	要確認
担当	アプリ開発 + DBA

案B: HASH_AJ ヒントで HASH ANTI JOIN を強制

```
WHERE NOT EXISTS (  
  SELECT /*+ HASH_AJ */ 'X' FROM TESTABLE TEMP2 ...  
)
```

	内容
メリット	SQL 構造・返却データを変えずに1行追加のみで NESTED LOOPS ANTI を回避できる
デメリット	アプリコード (Java) の改修・テスト・リリース工数が必要
デメリット	CTE の二重評価問題は別途 MATERIALIZE ヒントで対処しないと残る
業務要件への影響	なし
担当	アプリ開発 + DBA

デメリット：SELECT リストの列・型を厳密にそろえる必要があり置き換えコストが中程度



業務要件に
影響ない案を
出して！

チューニング考察: SQL <SQL_ID> 問題2 対応案

対象情報

項目	値
DB Name	<DB名>
解析対象期間	2026-04-22 23:00 ~ 2026-04-23 03:00
対象 SQL	<SQL_ID> (<DAOクラス>.<メソッド名>)
参照 AWR サマリー	20260423_<DB名>_snap10138-10142_summary.md

問題の概要

症状	詳細
CPU/Logical I/O 爆発	00:00~01:00 の Buffer Gets 合計 38.7億ブロック (1実行あたり 1.85M gets)
実行回数の急増	23:00~00:00:00 の 8回/時 → 00:00~01:00 で 2,092回/時
avg Elapsed の劣化	0.85s (正常時) → 最大 163.4s (02:00~03:00)
実行計画	Plan Hash Value: (NESTED LOOPS ANTI)

対応案 メリット・デメリット

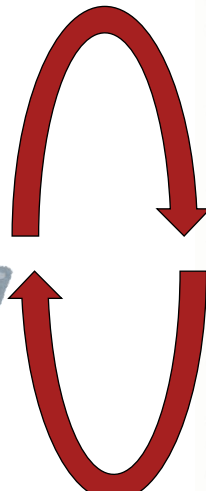
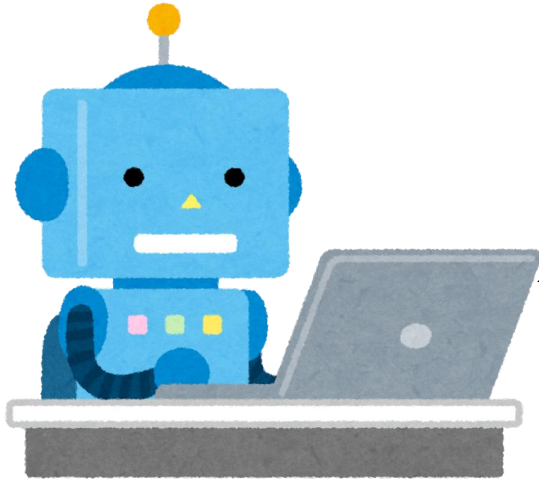
案D: (<カラムA>, <カラムC>) 複合インデックス作成

```
CREATE INDEX <インデックスB>
ON <テーブルA> (<カラムA>, <カラムC>);
```

観点	内容
メリット	SQL・アプリに一切手を入れない。DBA 作業のみで完結する
メリット	LIKE CONCAT(:6, '%') は前方一致のため Index Range Scan が有効に機能する
メリット	亜種 SQL <SQL_ID_2> にも同時に効果が出る
デメリット	根本問題 (CTE 二重評価・NESTED LOOPS ANTI) は解消されない
デメリット	INSERT/UPDATE 時のインデックスメンテナンスコストが増加する
デメリット	<テーブルA> が大テーブルの場合、作成中のロックと領域使用量に注意が必要

SQL・アプリに一切手を入れない。DBA作業のみで完結する

AIによる自動化！？（+ MCP ツール利用）



Phase 2 開始です。「adbtest02_dba_copilot に接続して、直近で遅かった SQL を診断してください」

sqlcl: connect を使用しました >

Step 1: SCOTT スキーマで直近で遅かった SQL を V\$SQL から特定

sqlcl: run-sql を使用しました >

Step 2: 実行計画を確認 (TABLE ACCESS FULL の確認)

sqlcl: run-sql を使用しました >

Step 3: ASH で待機イベントとリソース消費を分析

sqlcl: run-sql を使用しました >

Step 4: テーブル統計・インデックス状況を確認

sqlcl: run-sql を使用しました >

Step 5: 並列化の余地を確認 (PARALLEL 設定・DOP)

sqlcl: run-sql を使用しました >

I have all the data I need. Now let me compile the diagnosis report and save it. Let me also think about what oracle/skills tells us about this situation.

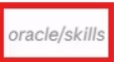
Key findings:

1. **SQL**: Full table scan on DUMMY_DATA_LOAD (10億行)
2. **Execution plan**: TABLE ACCESS STORAGE FULL → Smart Scan (ADB/Exadata storage offloading)
3. **ASH**: 85.7% cell smart table scan (User I/O), 14.3% on CPU
4. **Wait**: avg_wait_ms = 10ms per sample (storage I/O)
5. **Table stats**: last_analyzed = 2026-01-09 (about 4 months ago), 1.08 billion rows, no indexes
6. **Parallel**: degree=1, no parallelism configured
7. **Filter**: code='TARGET_CODE_99999' AND description LIKE '%SLOW%' - both non-indexed, description has LIKE '%SLOW%' (leading wildcard = can't use index even if one existed)

The root cause is clear: we're doing a full table scan on 1.08 billion rows with no indexes and a leading wildcard in the filter predicate, though ADB's Smart Scan is mitigating the impact somewhat by offloading the work to storage.

Even at 6.4 seconds, this query becomes problematic at scale with frequent execution. I can improve this by adding an index on the CODE column, though the LIKE '%SLOW%' pattern won't benefit from a standard index and would require Oracle Text for substring matching. Increasing the parallel degree would also help reduce scan time across the massive dataset.

診断データが揃いました。oracle/skills の知識を踏まえてレポートを生成し、使用クエリも保存します。



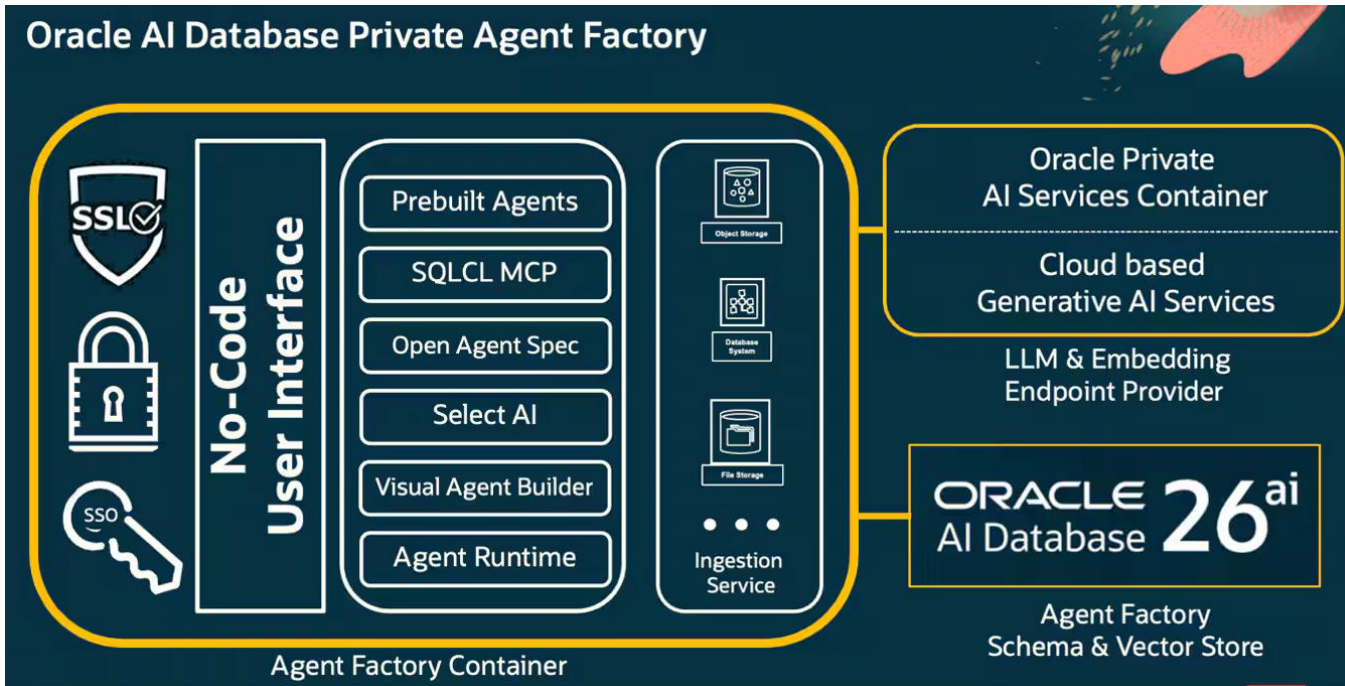
一度指示を出すと自律的に考えて考察を進めてくれる

※前頁までとは別の検証結果

おまけ

Private Agent Factory

■ AI エージェントをマネージするためのノーコードのエージェント型プラットフォーム



	Claude Code + oracle/skills + SQLcl MCP (or ADB MCP)	Oracle Private Agent Factory
実行環境	ローカル	OCIマネージド
ガバナンス	ローカル側の設定	OCI IAM + ポリシー
Oracle/OCI 統合	MCP経由	ネイティブ統合
使い分け?	軽い・速い・安い?	エンタープライズ運用・組織統制

01

NRIのOCI活用への取り組み

02

OCIの新IAM「Identity Domains」への移行事例

03

Exadata&Compute性能対策実践ノウハウ

2. OCIの新IAM「Identity Domains」への移行事例

はじめに（自己紹介）

■阿部 祐希（あべ ゆうき）

2016年 野村総合研究所に入社

金融機関向けのプライベートクラウド運営を経験したのち、野村総合研究所（NRI）のデータセンターで稼働する専用パブリッククラウドサービス「NRIクラウド OCI 区画」の維持管理・運営に従事。社内でのOCIの情報発信活動や人材育成・コミュニティ活動にも従事。

■資格

Oracle Cloud Infrastructure 2025 Architect Professional

Oracle Cloud Infrastructure 2025 Security Professional

Oracle Cloud Infrastructure 2025 Generative AI Professional

Oracle Cloud Infrastructure 2025 Networking Professional

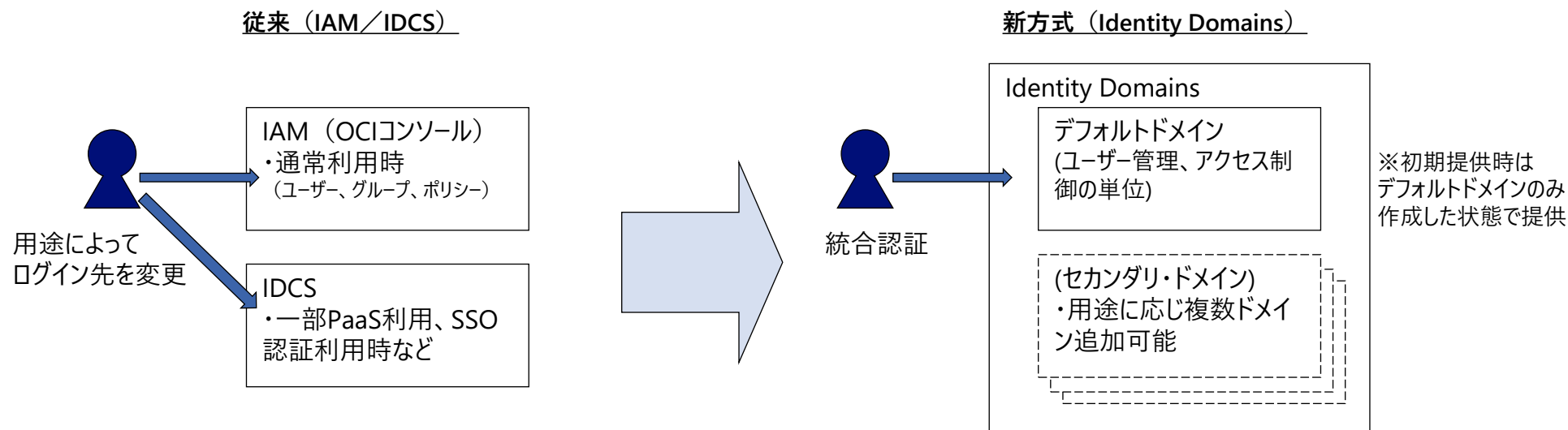


「Oracle Alloy」を活用した「NRIクラウド OCI 区画」の提供・運営を担当

はじめに

■ Identity Domainsとは

- 従来OCIの認証基盤（及びコンソール）として、IAM(※1)とIDCS(※2)の2つが提供されていたが、Identity Domains として統合された環境が提供された
- ※1) IAM：OCIコンソール上のユーザー、グループ、ポリシー等の管理基盤
- ※2) IDCS：一部PaaSを利用する際やSSO認証時に提供される管理基盤。OCIコンソールとは別で提供。



2022年後半頃からIdentity Domainsへの移行が始まる。

→ 本日は移行の体験談として移行時に対応したことや良かった点、困った点をお話したいと思います。

移行の流れ

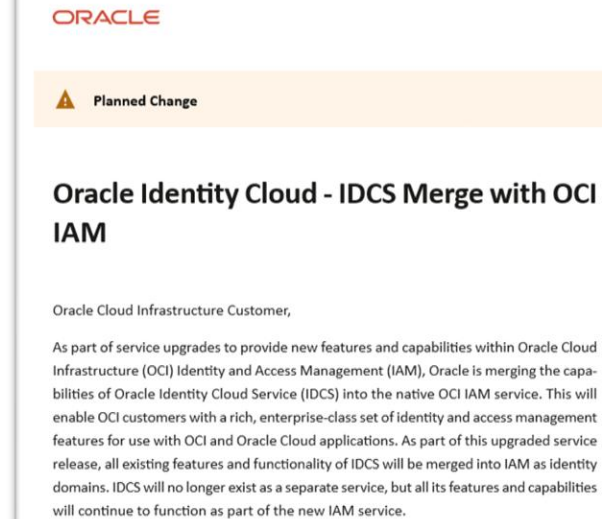
■ 移行日の 2 - 4週間前

- Planned Change 通知が届く
 - Audit, Event ログにグループ・ユーザー作成通知大量

■ 移行当日

- 通知された時間にすぐに移行されるわけではなく、1-2時間たってから移行完了
 - ログイン画面が変更されたら移行完了
 - ドメイン作成などもその時点から作成可能
(移行時の影響については次ページ)

移行予定通知



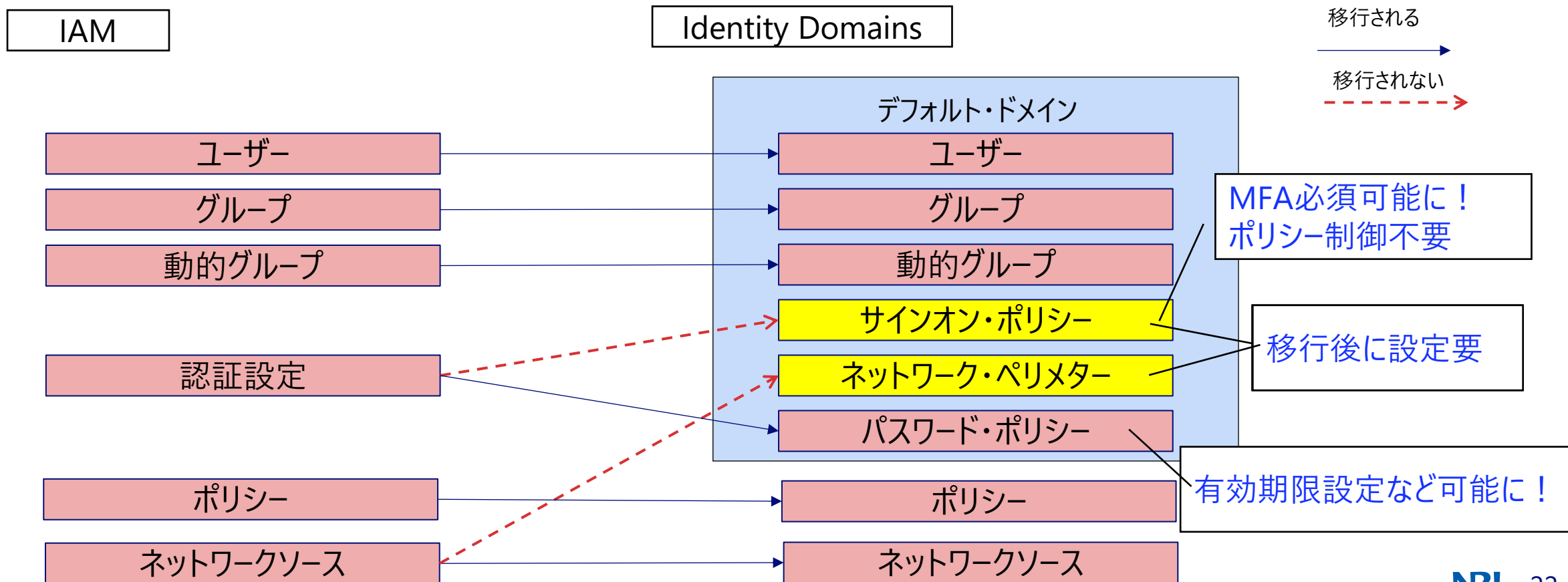
Identity Domains ログイン画面



Identity Domainsへの移行中の影響

■ 移行中の認証は問題なし

- 移行時間中 Cloud shell, Computeから OCI CLIを連続実行 → **エラー無し**
- ユーザーのMFAやパスワード、APIキーも引き続き利用可能

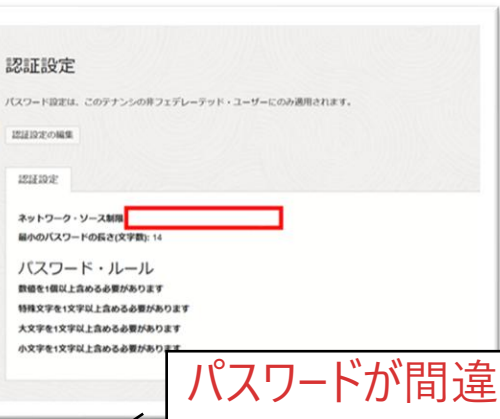


Identity Domainsで変わったこと①

■ OCIコンソールログインのグローバルIPアドレス制御

- 導入前は「認証設定」+「ネットワーク・ソース」で設定
- 導入後は、「サインオン・ルール」+「ネットワーク・ペリメター」で設定

IAM



パスワードが間違っている
というエラーメッセージ

拒否画面

Your login attempt failed. The credentials you entered do not match our records. If you have unsuccessfully attempted to authenticate with this username multiple times your account may be locked out and must be reset via the administrator. If this is a new user account, please contact your administrator for your temporary password.

Oracle Cloud Infrastructure 資格証明でサインイン

ユーザー名

パスワード

Identity Domains



拒否画面

ORACLE Cloud

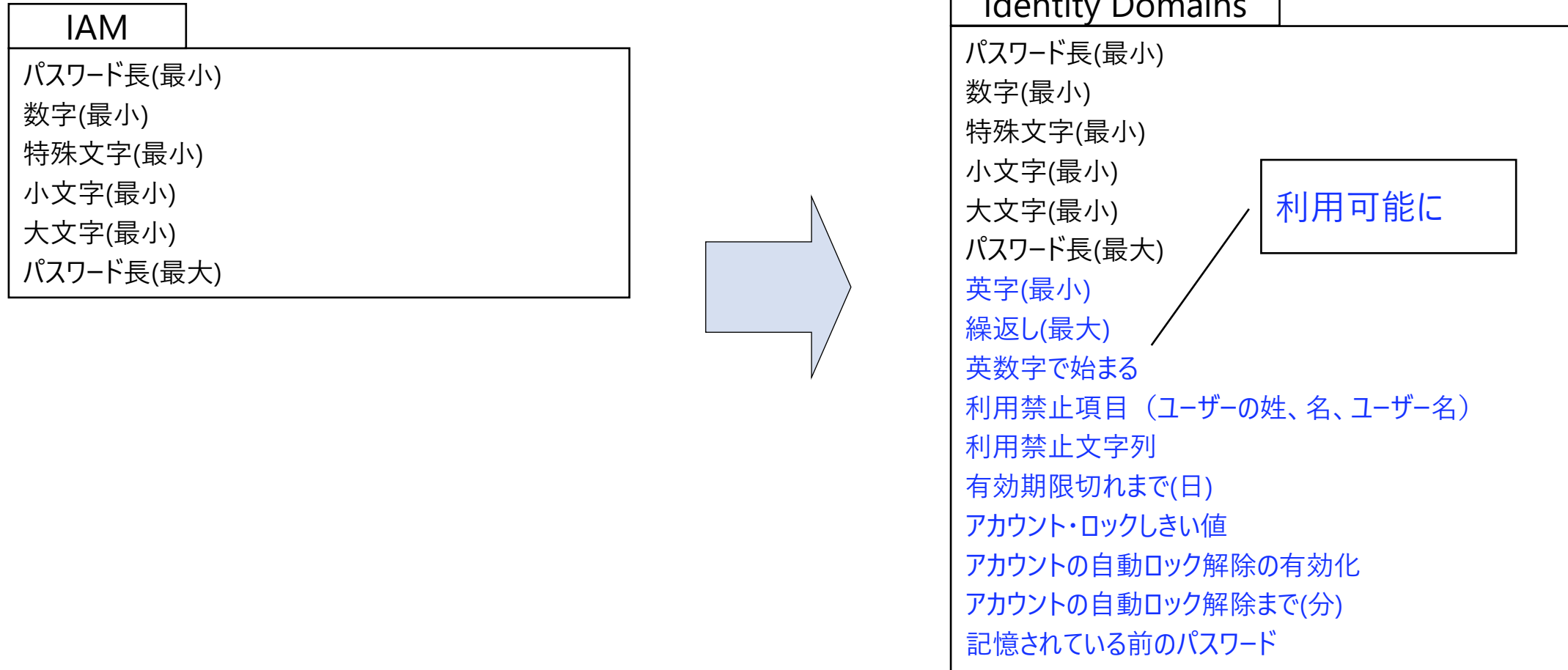
アイデンティティ・ドメイン

ポリシーによる拒否が明記
→問い合わせ激減

サインオン・ポリシーはアクセスを拒否します。

Identity Domainsで変わったこと②

- IAMと比較してIDCSと同等のより複雑なパスワードポリシーの設定が可能に！



監査要件が厳しいお客様や、MFAを利用できない場合の複雑なパスワードにも対応可能に！

Identity Domains移行でのトラブル①

■メンバーシップOCIDが変わっていた

- Resource managerでユーザー管理していたテナンシーで
移行後メンバーシップOCIDの差分が出てしまい、そのままでは差分管理が管理できなくなった。
 - **メンバーシップOCID**：グループとユーザーの紐づけ設定※OCIコンソール上からは確認できない
- Stateファイルを手動で修正して対応。

Stateファイル

```
{
  "schema_version": 0,
  "attributes": {
    "compartment_id": "ocidl.tenancy. [REDACTED]",
    "group_id": "ocidl.group.o [REDACTED]",
    "id": "ocidl.groupmembership. [REDACTED]",
    "inactive_state": null,
    "state": "ACTIVE",
    "time_created": "2022-04-06 04:54:58.682 +0000 UTC",
    "timeouts": null,
    "user_id": "ocidl.user. [REDACTED]"
  }
}
```

Identity Domains移行でのトラブル②

■プランによる単位時間当たりのAPI実行数

- Resource manager でユーザー作成のジョブが並列度が高い場合に失敗
 - API 429エラーが発生。レート制限による仕様
 - Identity Domainsには ライセンスタイプごとにAPIレート制限が異なる仕様
→ プランを適切に見直しをするか並列度を下げる必要 ※デフォルトはFree

すべてのアイデンティティ・ドメイン・タイプに対するレート限度

APIグループ	あたり	Free	Oracle Apps	Oracle Apps Premium	Premium	External User
AuthN	秒	10	50	80	95	90
AuthN	分	150	1000	2100	4500	3100

参考：<https://docs.oracle.com/ja-jp/iaas/Content/Identity/sku/api-rate-limiting.htm>

まとめ

- Identity Domainsへの移行事例のご紹介をさせていただきました。

MFAがサインオンポリシーで制御できるようになった

ログイン拒否のエラーメッセージが明確化された

- 移行されたことで、これまでIAMで設定していた部分の簡略化が可能になっています。
- すでに移行済の場合は意図しない設定になっている場合は、見直しをこの機会にお願いします。
- 今回ご紹介していない新しい概念もありますので、IAMから利用されていた方はこの機会に設計の見直しなどしていただけるとよいと思います。
 - 例：Administrator Roles、ドメイン権限、ブランディング設定

- 01 NRIのOCI活用への取り組み
- 02 OCIの新IAM「Identity Domains」への移行事例
- 03 Exadata&Compute性能対策実践ノウハウ

3. Exadata&Compute性能対策実践ノウハウ

はじめに（自己紹介）

■ 落合 俊貴（おちあい としき）

- ✓ 2023年 野村総合研究所に入社
- ✓ OCI上で稼働するミッションクリティカルな**金融系システム**の開発・運用保守に従事
- ✓ “**Oracle Exadata Database Service**を用いた**堅牢なDB基盤**”と“**OKE**による**モダンなコンテナ環境**”を融合させた大規模システムの構築プロジェクトを推進中

■ 資格

✓ Oracle関連資格

- Oracle Cloud Infrastructure Certified Architect Professional
- OCI Generative AI Certified Professional
- Oracle AI Vector Search Professional
- Oracle Master Gold DBA 2019

✓ その他の資格等

- Japan All AWS Certifications Engineers（AWS“全冠”）
- Red Hat 認定エンジニア (RHCE)
- 高度情報処理技術者（ネットワーク）
- Cisco Certified Network Associate (CCNA)
- ディープラーニングG検定



OCIを活用した大規模システムの開発・運用保守を担当

3. Exadata&Compute性能対策実践ノウハウ

はじめに（需要に応じたインフラ構築 ～オンプレ VS OCI～）

- OCIをはじめとするクラウド環境では、拡張を前提としたインフラ設計が容易に可能



- ユーザ数が増加し、将来的に性能不足が発生しそう...
- とはいえ、現時点から余分なコストはかけたくない...

オンプレミス

- ✓ 物理機器の調達が**必要**
 - リードタイムを加味した発注が必要
 - 巨額の先行投資が必要
 - 増設後も維持コストが固定で発生

完全なシステム管理が可能だが
拡張性の観点では不利

OCI（クラウド）

- ✓ 物理機器の調達が**不要**
 - 即時にスケール可能
 - 初期費用が基本的に不要
 - コストは従量課金ベース

需要に連動した
拡張性の高いインフラ構築が可能

OCIのクラウドメリットを享受した、拡張性のあるインフラ構築のポイントについてご説明

3. Exadata&Compute性能対策実践ノウハウ

はじめに（本セクションのアジェンダ）

■ Exadata&Compute性能対策実践ノウハウ

- “拡張性のあるインフラ構築”について、OCIでの基盤構築プロジェクトの経験をもとにインフラ観点からご説明

3.1 性能対策実践ノウハウ ～Compute編～

- OCI Computeの特徴
- Compute関連のサイジング要素
- Computeの拡張性
- ネットワークIO / ディスクIOの考え方
- ディスクIO（ブロックボリューム）の拡張性
- ケーススタディ ～ 性能のボトルネック ～
- Computeの性能対策実践ポイント

3.2 性能対策実践ノウハウ ～Exadata編～

- はじめに（OCIのデータベースサービス）
- Exadataの特徴と構成
- Exadataの歴史と進化
- X8MとX11Mの性能比較
- Exadata(ExaDB-D)の拡張性
- Exadataの性能対策実践ポイント

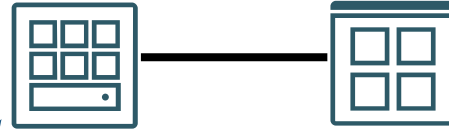
- まとめ

3.1. 性能対策実践ノウハウ ～Compute編～

OCI Computeの特徴

- “フレキシブル・シェイプ”ではCPU/メモリを柔軟に選択でき、ブロックボリュームはオンラインで性能変更が可能

仮想マシン(VM) ブロックボリューム



※VPU：ボリューム・パフォーマンス・ユニット

OCIのComputeでは柔軟なサイジングが可能

➤ 一般的なクラウドサービス

「2 vCPU/8 GB」「4 vCPU/16 GB」など、
予め決められた固定スペックの組から選択

➤ OCIのCompute (フレキシブル・シェイプ)

CPUコア数(OCPU)とメモリ(GB)を完全に独立して、
自由に組み合わせてプロビジョニング可能

ブロックボリュームは「VPU」単位で性能を調整可能

✓ 4タイプから選択。「超高性能(UHP)」は10段階でスケール可能

- 低コスト (0 VPU)：1GBあたり 2 IOPS, 240 KB/s
- バランス(10 VPU)：1GBあたり 60 IOPS, 480 KB/s
- 高パフォーマンス(20 VPU)：1GBあたり 75 IOPS, 600 KB/s
- 超高性能(30-120 VPU)：1GBあたり最大225 IOPS, 1,800 KB/s

オンライン状態のまま即座にパフォーマンスレベルが変更可能

詳細はこちらを参照：<https://docs.oracle.com/en-us/iaas/Content/Block/Concepts/blockvolumeperformance.htm>

OCIではVMやストレージの性能を柔軟に変更するための仕組みが提供されている

3.1. 性能対策実践ノウハウ ～Compute編～

Compute関連のサイジング要素

■ Compute関連のサイジング要素として、主にCPU/メモリ/ネットワークIO/ディスクIOの4つが挙げられる

- CPU/メモリ

- シェイプごとにスケールアップ上限値が異なる
- 構築完了後からでも、必要に応じて**容易に拡張可能**（OS再起動は必要）

- ネットワークIO

- OCPU数に応じて最大のネットワーク帯域幅がスケール
- 基本的に、上限までは1 OCPUにつき1 Gbpsずつスケールするため、こちらも**容易に拡張可能**

- ディスクIO（ブロックボリューム）

- IOPS と スループット(MB/s) それぞれに上限値が存在。
 - 非UHPディスクの上限は 50,000 IOPS、680 MB/s
 - UHPディスクの上限は 300,000 IOPS、2,680 MB/s

注意

後から非UHP→UHPディスクに変更できないケースあり(後述)

- **オンライン状態のまま**即座にパフォーマンスレベルが変更可能（OS再起動やデタッチは不要※）

※ボリュームが複数アタッチされている場合を除く

構築時に、Computeシェイプ・UHPディスク要否を正しく判断すれば拡張は容易

3.1. 性能対策実践ノウハウ ～Compute編～

Computeの拡張性

■ 最新のシェイプを活用することで十分な拡張余力のあるインフラ構築が可能

➤ 上限値：代表的な汎用シェイプごとの上限値は下表の通りで、最新モデルでは天井が大きく引き上げられた

シェイプ	プロセッサ	最大OCPU	最大メモリ (GB)	最大ネットワーク帯域(Gbps)	
VM.Standard.E6.Flex	第5世代 AMD EPYC	126	1,454	99	2025年
VM.Standard.E5.Flex	第4世代 AMD EPYC	94	1,049	40	2023年
VM.Standard3.Flex	第3世代 Intel Xeon	32	512	32	2021年

※1OCPU = 2 vCPU

➤ 需要(使用量)の計測：

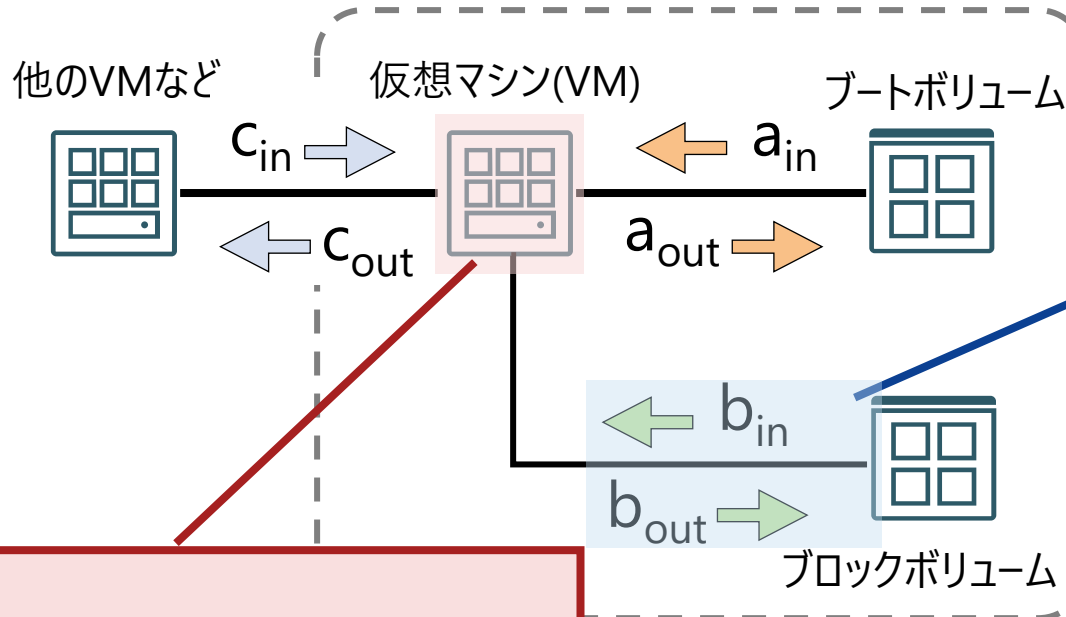
- CPU/メモリ：Computeのメトリック “CPUUtilization” や “MemoryUtilization” から取得可能
- ネットワークIO：VNICのメトリック “VnicFromNetworkBytes” と “VnicToNetworkBytes” から取得可能

👉 過去90日ぶんの使用量を保持でき、将来の需要予測の材料となる

「小さく生んで大きく育てる」方針で、
需要に合わせて CPU&ネットワークIO / メモリ を拡張し性能対策できる

ネットワークIO / ディスクIOの考え方

■ ネットワークIO / ディスクIOについては、上限値の見極めがCPU/メモリに比べ複雑



ネットワークIO

- IN / OUT方向それぞれに対して上限が存在
 - ブート/ブロックボリュームに対するIO帯域も含む
- ☞ 例えばVMが 4 OCPU (= 上限4 Gbps) の場合、
- $a_{in} + b_{in} + c_{in} < 4 \text{ Gbps}$
 - $a_{out} + b_{out} + c_{out} < 4 \text{ Gbps}$

となる

ディスクIO

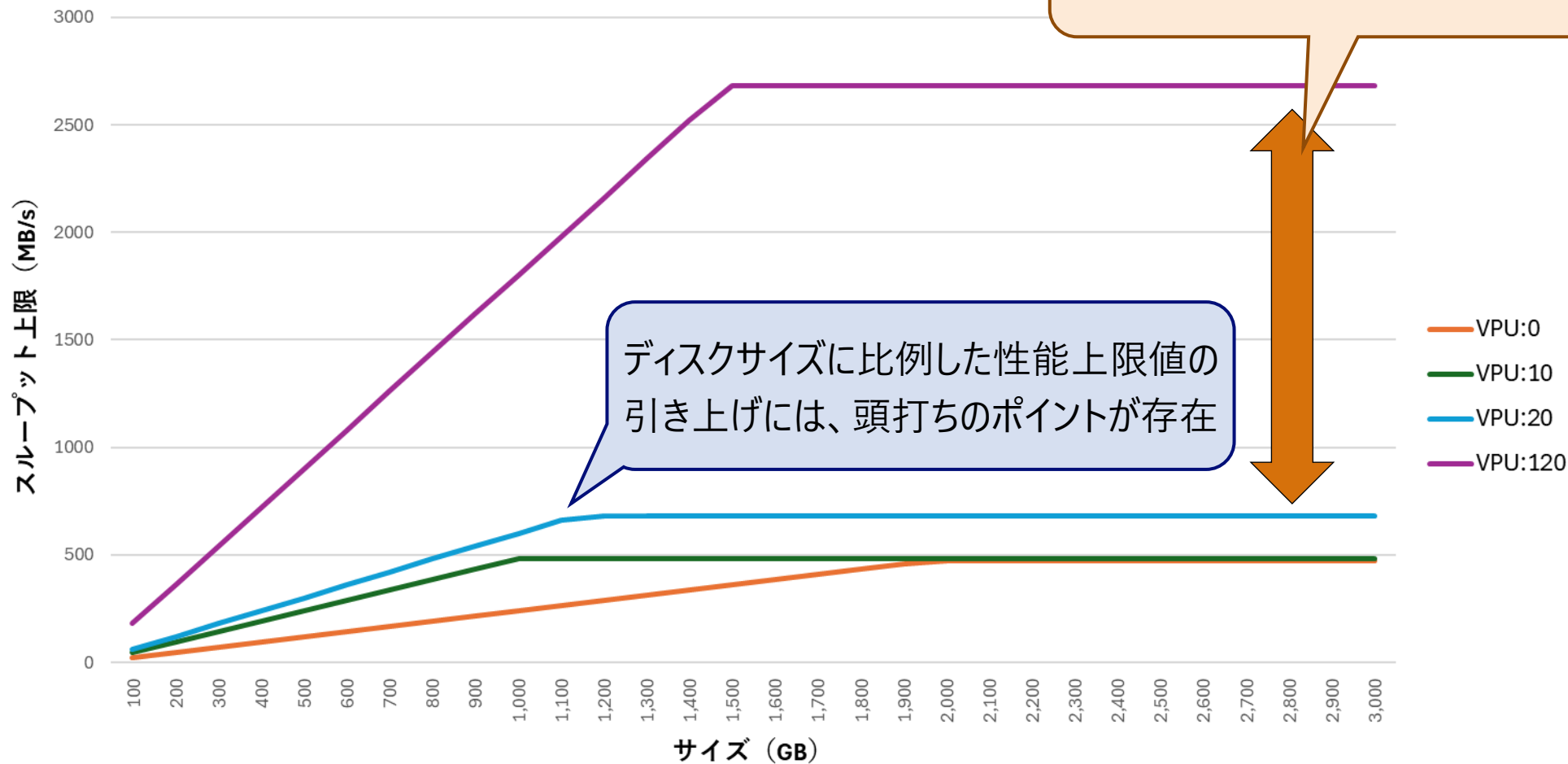
- 各ボリュームごとに、IOPS・スループット(MB/s) ぞれぞれの上限が存在
 - Read/Writeの和に対して上限が設定される
- ☞ 非UHPディスクの場合、
- $b_{in} + b_{out} < 50,000 \text{ IOPS}$
 - $b_{in} + b_{out} < 680 \text{ MB/s}$

となる

- 上限値はボリュームサイズとVPUによって決まる
- ☞ ボリュームサイズ(GB)を上げると、初めはそれに比例して上限値が引き上げられるが、一定サイズ到達で上限値が頭打ちになる
- それ以上に性能を高めたい場合はVPUを上げる必要あり (UHP化の検討など)

【補足】ディスクIOの考え方

■ ディスクIOの性能上限はディスクサイズとVPUに依存



3.1. 性能対策実践ノウハウ ～Compute編～

ディスクIO（ブロックボリューム）の拡張性

■ UHPディスクは高い拡張性をもつため、ディスクIO性能対策として有効

- 非UHPディスクのIO上限は高くなく、構成上のネックになりやすい

☞ UHPディスクを採用すれば、

- 非UHPディスクに比べIOPSが6倍、スループット(MB/s)が約4倍までスケール可能
- 30～120 VPUの間を10段階(10 VPU単位)で制御できるため、必要な性能のぶんだけを設定可能

- ただし、UHPディスクを利用するためには、Computeが「マルチパス」に対応していることが必要

【躓きポイント】：OCPU数の壁

- 「16 OCPUの壁」：マルチパス構成が利用可能となるのは16 OCPU以上のVMのみ
 - これはUHPディスク化したい時に、後からスケールアップすれば問題なし
- 「8 OCPUの壁」：インスタンスの**一番初めの構築時に、8 OCPU以上で作成することが必要**
 - これを守らないと、後から16 OCPU以上にしてもマルチパスが有効にならないことがある

これらのポイントを押さえておけば、後から非UHPディスク→UHPディスクに変更することも可能

マルチパス対応の構成にしておけば、ディスクIOについても高い拡張性を確保可能

3.1. 性能対策実践ノウハウ ～Compute編～

ケーススタディ ～ 性能のボトルネック ～

- ケーススタディ：なぜディスクIO（スループット）が出ないのか？

VM（VM.Standard.E5.Flex：2 OCPU / 32GB メモリ）のディスク書き込み処理が遅いため、アタッチされているブロックボリュームをUHPディスクにして、1,000 MB/s (8 Gbps相当) のスループットが出るように設定した。しかし、OCIメトリックを確認すると、240 MB/s (約2 Gbps) で頭打ちになっている。

性能改善するために怪しい箇所はどこか？

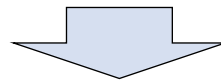
3.1. 性能対策実践ノウハウ ～Compute編～

ケーススタディ ～ 性能のボトルネック ～

■ ケーススタディ：なぜディスクIO（スループット）が出ないのか？

VM（VM.Standard.E5.Flex：2 OCPU / 32GB メモリ）のディスク書き込み処理が遅いため、アタッチされているブロックボリュームをUHPディスクにして、1,000 MB/s (8 Gbps相当) のスループットが出るように設定した。しかし、OCIメトリックを確認すると、240 MB/s (約2 Gbps) で頭打ちになっている。

性能改善するために怪しい箇所はどこか？



【原因】 VMを「2 OCPU」でプロビジョニングしていたため

1 OCPUあたりのVMのネットワーク帯域上限は 1 Gbps であるため、このVMの最大帯域は2 Gbps

👉 ブロックボリューム側のVPUをどれだけ高くしても、VM側のVNICで詰まってしまい性能が発揮できない

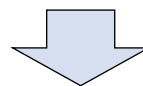
【解決策】 ストレージ性能 (VPU) を上げる前に、VMのOCPU数をスケールアップ

Computeの性能対策実践ポイント

■ Computeの性能対策においては、OCPU数のサイジングが特に重要となる

性能対策における実践ポイントは...

- ◆ 利用シェイプのCPU/メモリ/ネットワークIO上限値を把握し、将来の需要を満たす拡張余力があることを確認しておく
- ◆ 非UHPのブロックボリュームは、ディスクIOがボトルネックになりやすい
 - 16 OCPU以上のVMに対しては、（VMがマルチパス対応の状態ならば、）UHPディスク化による拡張性向上が可能
 - 16 OCPU未満のVMについては、新規ディスクを追加しIOを分散させる（ディスク分割）等の対応策を要検討
- ◆ ディスクIOを高める際、VM (VNIC) のネットワークIOを連動して上げる必要がないか点検が必要
 - ネットワークIOの上限値はOCPU数に応じて決定される



- ✓ ディスクIO/ネットワークIOの性能観点からも、適切なOCPU数を決定することが重要
 - 事前にOCPUサイジングを計画しておくことは、予期せぬインフラ費用増加を防ぐことにもつながる
 - CPUコア数に応じて課金が発生するミドルウェアを利用する場合は特に注意

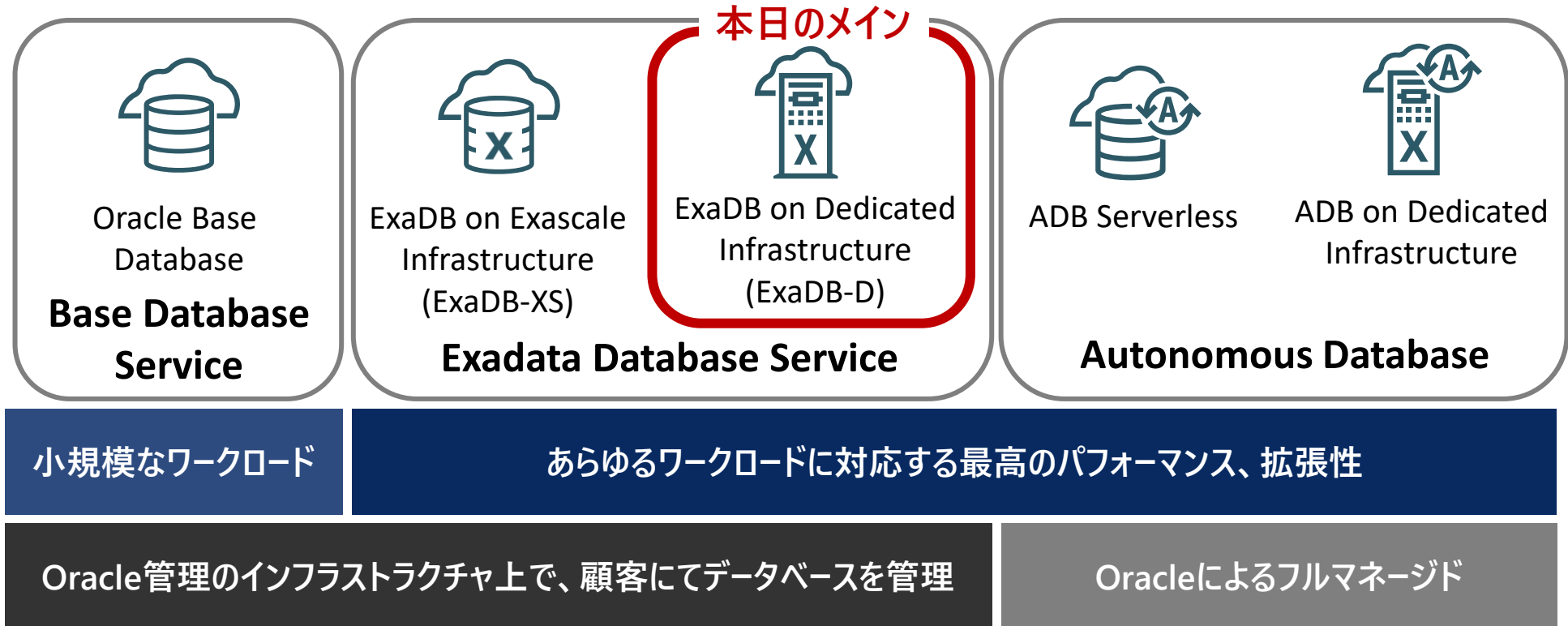
将来を見越したOCPUサイジングが、性能・コスト観点の両方にとって重要

3.2. 性能対策実践ノウハウ ～Exadata編～

3.2. 性能対策実践ノウハウ ～Exadata編～

はじめに（OCIのデータベースサービス）

- OCIでは要件に応じて多様なデータベースサービスが選択可能



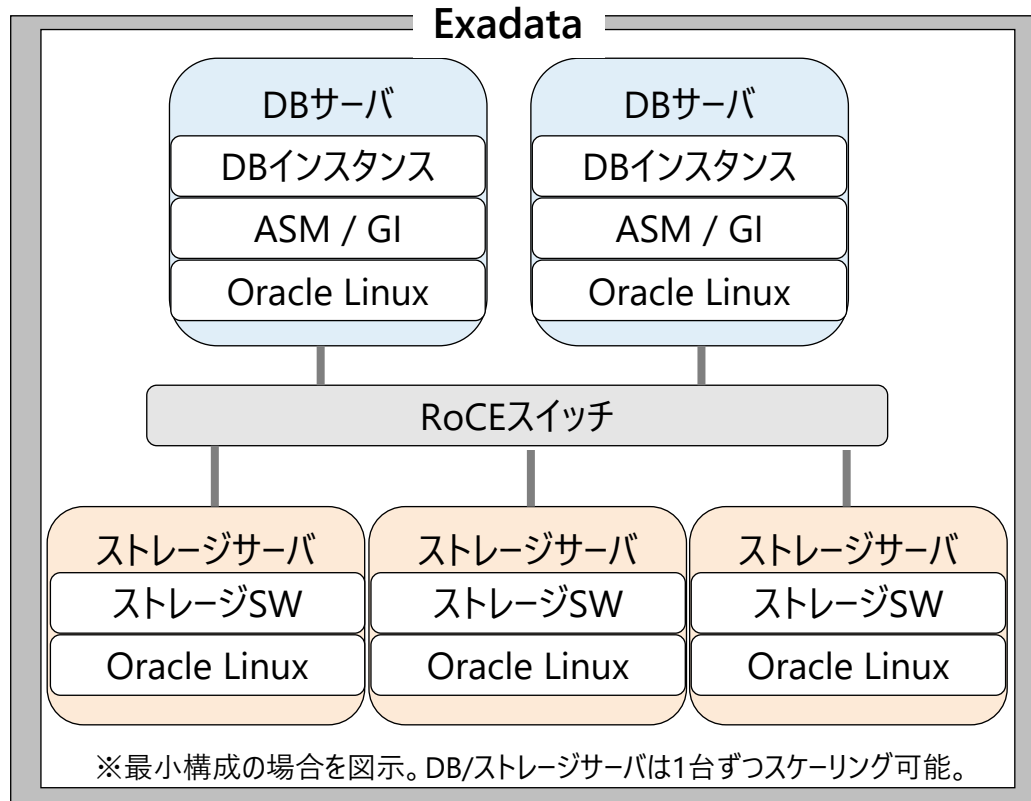
<https://speakerdeck.com/oracle4engineer/exadata-database-cloud-technical-detail>

本日は、超高パフォーマンスを実現するExadata(ExaDB-D)を中心にご説明

3.2. 性能対策実践ノウハウ ～Exadata編～

Exadataの特徴と構成

■ Exadataは、Oracle DBのパフォーマンス・可用性を最大化するために設計されたDB基盤



<https://www.oracle.com/database/technologies/exadata/architecture/>

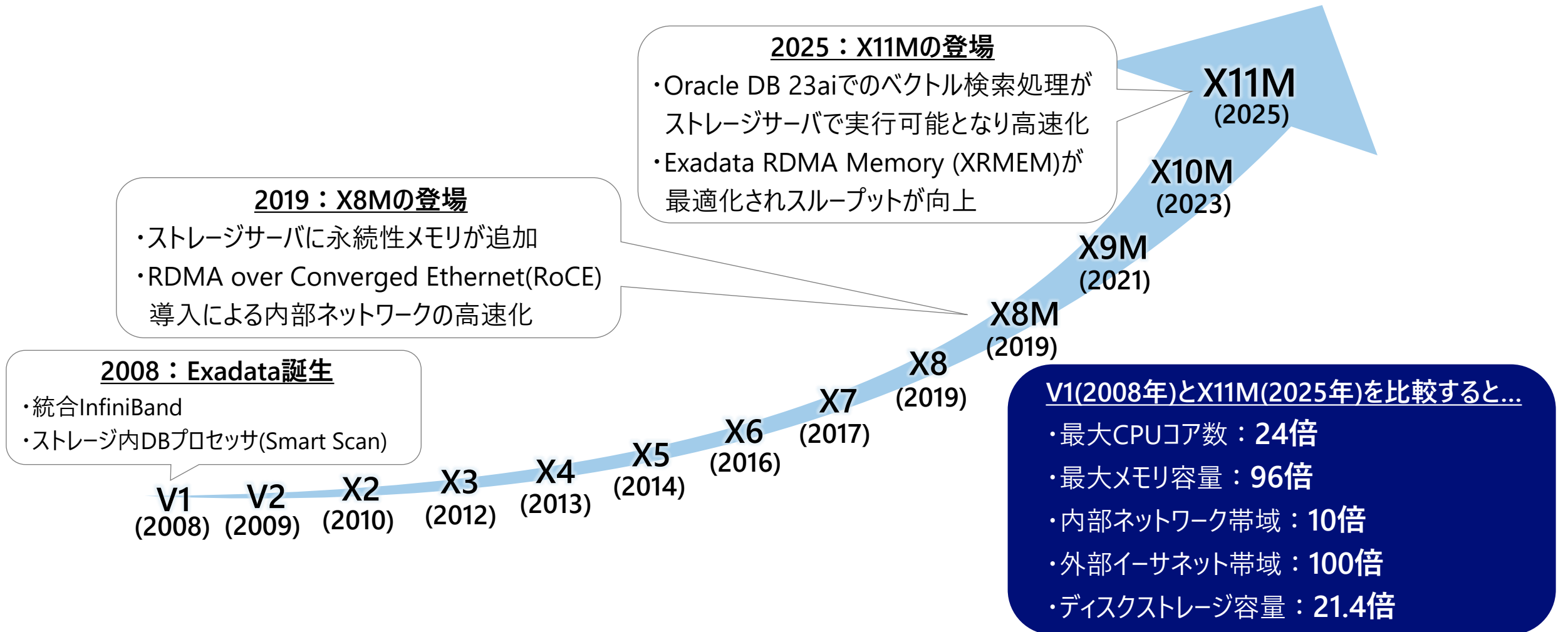
- Exadataは大きく三つの要素で構成される
 - データベースサーバ
 - 内部ネットワーク(RoCE) ※ RoCE : RDMA over Converged Ethernet
 - ストレージサーバ
- ✓ それぞれの要素が冗長化されており高可用な基盤
- ✓ DB層とストレージ層で分散処理することで高性能を実現
 - CPU/メモリが搭載された演算処理可能なストレージサーバにより、データベース処理の一部をストレージサーバ側で実行 (=Smart Scan)

Oracle Databaseとして最適化されたハード・ソフトウェア基盤

3.2. 性能対策実践ノウハウ ～Exadata編～

Exadataの歴史と進化

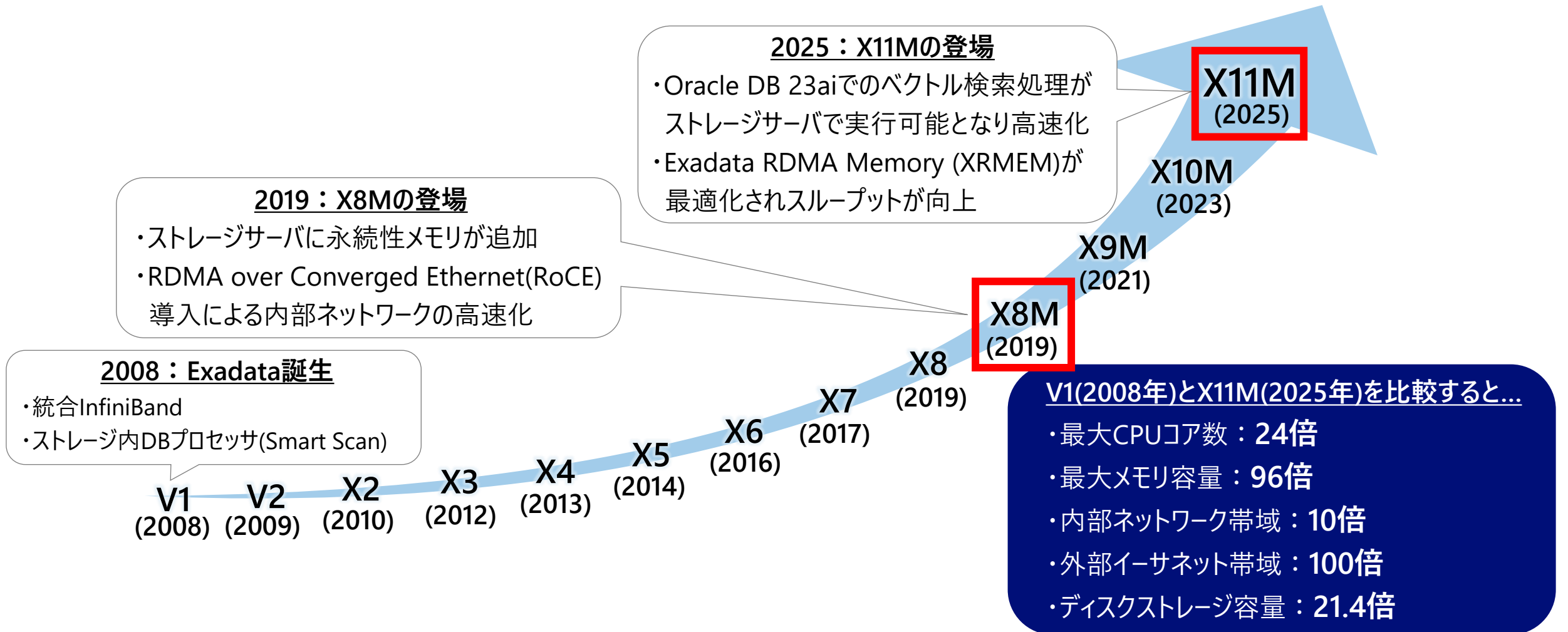
- Exadataは定期的にアップデートされており、進化を続けている



3.2. 性能対策実践ノウハウ ～Exadata編～

Exadataの歴史と進化

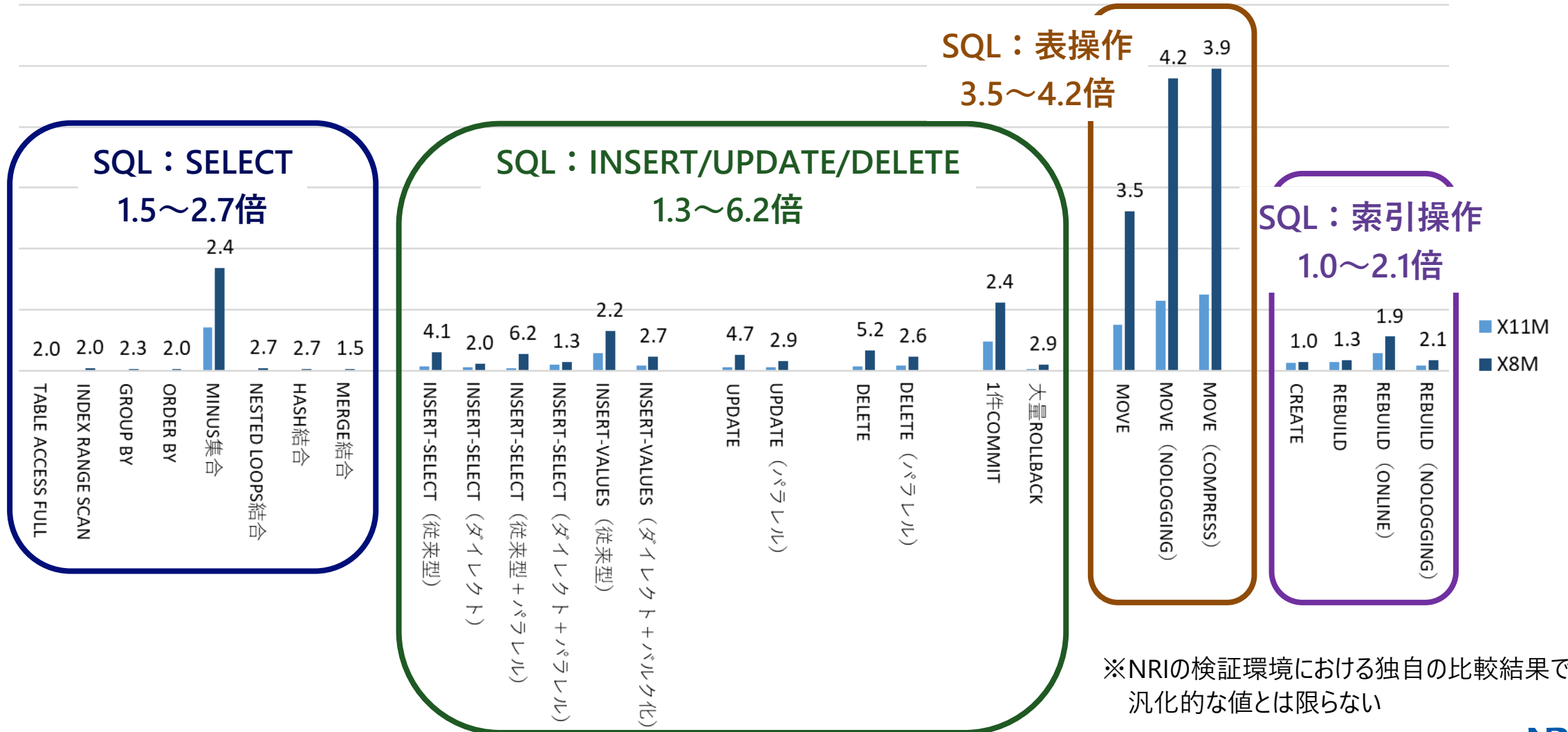
- Exadataは定期的にアップデートされており、進化を続けている



3.2. 性能対策実践ノウハウ ～Exadata編～

X8MとX11Mの性能比較

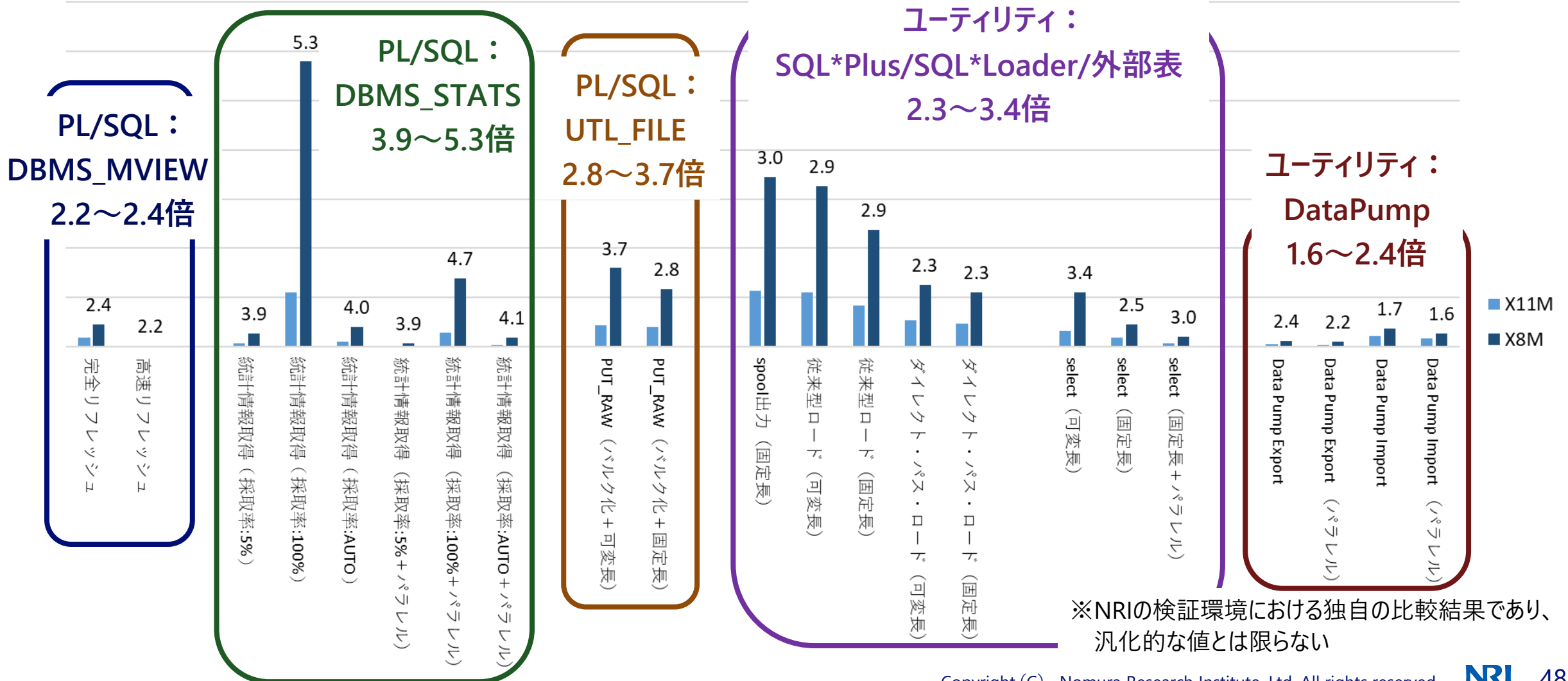
■ X8MとX11Mにて様々な処理時間を比較し、処理ごとに何倍程度の高速化が実現されたのかを確認 (※)



3.2. 性能対策実践ノウハウ ～Exadata編～

X8MとX11Mの性能比較

■ X8MとX11Mにて様々な処理時間を比較し、処理ごとに何倍程度の高速化が実現されたのかを確認 (※)



3.2. 性能対策実践ノウハウ ～Exadata編～

X8MとX11Mの性能比較（考察）

■ X11Mでは、X8Mに比べ約2～5倍の高速化が測定された。

この結果をアーキテクチャの違いから考察

- ① DBサーバの演算能力とメモリ帯域の飛躍的向上
- ② ストレージ・アーキテクチャの刷新
- ③ ネットワーク帯域の倍増

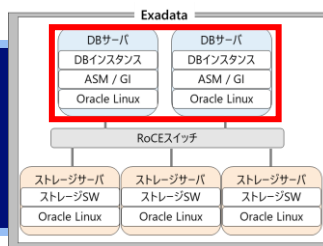
分類	操作	高速化
SQL : INSERT	INSERT-SELECT (従来型+パラレル)	6.2 倍
PL/SQLパッケージ : DBMS_STATS	統計情報取得 (採取率:100%)	5.3 倍
SQL : DELETE	DELETE	5.2 倍
SQL : UPDATE	UPDATE	4.7 倍
PL/SQLパッケージ : DBMS_STATS	統計情報取得 (採取率:100%+パラレル)	4.7 倍
SQL : 表操作	MOVE (NOLOGGING)	4.2 倍
SQL : INSERT	INSERT-SELECT (従来型)	4.1 倍
PL/SQLパッケージ : DBMS_STATS	統計情報取得 (採取率:AUTO+パラレル)	4.1 倍
PL/SQLパッケージ : DBMS_STATS	統計情報取得 (採取率:AUTO)	4.0 倍
SQL : 表操作	MOVE (COMPRESS)	3.9 倍
PL/SQLパッケージ : DBMS_STATS	統計情報取得 (採取率:5%+パラレル)	3.9 倍
PL/SQLパッケージ : DBMS_STATS	統計情報取得 (採取率:5%)	3.9 倍
PL/SQLパッケージ : UTL_FILE	PUT_RAW (バルク化+可変長)	3.7 倍
SQL : 表操作	MOVE	3.5 倍
ユーティリティ : 外部表	select (可変長)	3.4 倍
ユーティリティ : 外部表	select (固定長+パラレル)	3.0 倍
ユーティリティ : SQL*Plus	spool出力 (固定長)	3.0 倍
ユーティリティ : SQL*Loader	従来型ロード (可変長)	2.9 倍
SQL : UPDATE	UPDATE (パラレル)	2.9 倍
SQL : ROLLBACK	大量ROLLBACK	2.9 倍
ユーティリティ : SQL*Loader	従来型ロード (固定長)	2.9 倍
PL/SQLパッケージ : UTL_FILE	PUT_RAW (バルク化+固定長)	2.8 倍
SQL : INSERT	INSERT-VALUES (ダイレクト+バルク化)	2.7 倍
SQL : SELECT	HASH結合	2.7 倍
SQL : SELECT	NESTED LOOPS結合	2.7 倍

分類	操作	高速化
SQL : DELETE	DELETE (パラレル)	2.6 倍
ユーティリティ : 外部表	select (固定長)	2.5 倍
SQL : SELECT	MINUS集合	2.4 倍
SQL : COMMIT	1件COMMIT	2.4 倍
PL/SQLパッケージ : DBMS_MVIEW	完全リフレッシュ	2.4 倍
ユーティリティ : Data Pump	Data Pump Export	2.4 倍
ユーティリティ : SQL*Loader	ダイレクト・パス・ロード (固定長)	2.3 倍
SQL : SELECT	GROUP BY	2.3 倍
ユーティリティ : SQL*Loader	ダイレクト・パス・ロード (可変長)	2.3 倍
SQL : INSERT	INSERT-VALUES (従来型)	2.2 倍
ユーティリティ : Data Pump	Data Pump Export (パラレル)	2.2 倍
PL/SQLパッケージ : DBMS_MVIEW	高速リフレッシュ	2.2 倍
SQL : 索引操作	REBUILD (NOLOGGING)	2.1 倍
SQL : SELECT	TABLE ACCESS FULL	2.0 倍
SQL : SELECT	INDEX RANGE SCAN	2.0 倍
SQL : SELECT	ORDER BY	2.0 倍
SQL : INSERT	INSERT-SELECT (ダイレクト)	2.0 倍
SQL : 索引操作	REBUILD (ONLINE)	1.9 倍
ユーティリティ : Data Pump	Data Pump Import	1.7 倍
ユーティリティ : Data Pump	Data Pump Import (パラレル)	1.6 倍
SQL : SELECT	MERGE結合	1.5 倍
SQL : INSERT	INSERT-SELECT (ダイレクト+パラレル)	1.3 倍
SQL : 索引操作	REBUILD	1.3 倍
SQL : 索引操作	CREATE	1.0 倍

3.2. 性能対策実践ノウハウ ～Exadata編～

X8MとX11Mの性能比較（考察）

- ① DBサーバの演算能力とメモリ帯域の飛躍的向上
- ② ストレージ・アーキテクチャの刷新
- ③ ネットワーク帯域の倍増



■ 約4～6倍高速化された処理については、“DBサーバの演算能力とメモリ帯域の飛躍的向上”が寄与したと考察

分類	操作	高速化
SQL : INSERT	INSERT-SELECT (従来型+パラレル)	6.2 倍
PL/SQLパッケージ : DBMS_STATS	統計情報取得 (採取率:100%)	5.3 倍
SQL : DELETE	DELETE	5.2 倍
SQL : UPDATE	UPDATE	4.7 倍
PL/SQLパッケージ : DBMS_STATS	統計情報取得 (採取率:100%+パラレル)	4.7 倍
SQL : 表操作	MOVE (NOLOGGING)	4.2 倍
SQL : INSERT	INSERT-SELECT (従来型)	4.1 倍
PL/SQLパッケージ : DBMS_STATS	統計情報取得 (採取率:AUTO+パラレル)	4.1 倍
PL/SQLパッケージ : DBMS_STATS	統計情報取得 (採取率:AUTO)	4.0 倍
SQL : 表操作	MOVE (COMPRESS)	3.9 倍
PL/SQLパッケージ : DBMS_STATS	統計情報取得 (採取率:5%+パラレル)	3.9 倍
PL/SQLパッケージ : DBMS_STATS	統計情報取得 (採取率:5%)	3.9 倍
SQL : 表操作	MOVE	3.5 倍

DBサーバの演算能力とメモリ帯域の飛躍的向上

- ✓ CPU : Intel Xeon(2.4-3.9 GHz, 48コア)から第5世代 AMD EPYC(2.6-4.5 GHz, 192コア)に変更
 - ☞ クロック周波数が増加し、コア数も4倍に増強
- ✓ メモリ : DDR4(2933 MT/s)からDDR5(6400 MT/s)に
 - ☞ データ転送レートが2倍に向上

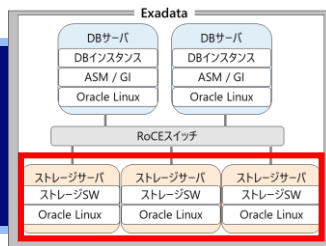


更新系DML (INSERT/DELETE/UPDATE) や 統計情報取得(DBMS_STATS) や 表操作(MOVE) :
 バッファキャッシュ上の空きブロック探索、UNDO/REDOの大量生成、PGA上でのソート・ハッシュ化、データ圧縮など、DBサーバ側でCPU・メモリ帯域を消費する処理が高速化

3.2. 性能対策実践ノウハウ ～Exadata編～

X8MとX11Mの性能比較（考察）

- ① DBサーバの演算能力とメモリ帯域の飛躍的向上
- ② **ストレージ・アーキテクチャの刷新**
- ③ ネットワーク帯域の倍増



■ 約3～4倍高速化された処理については、“ストレージ・アーキテクチャの刷新”などが寄与したと考察

分類	操作	高速化
PL/SQLパッケージ： UTL_FILE	PUT_RAW（バルク化＋可変長）	3.7 倍
ユーティリティ： 外部表	select（可変長）	3.4 倍
ユーティリティ： 外部表	select（固定長＋パラレル）	3.0 倍
ユーティリティ： SQL*Plus	spool出力（固定長）	3.0 倍
ユーティリティ： SQL*Loader	従来型ロード（可変長）	2.9 倍
SQL：UPDATE	UPDATE（パラレル）	2.9 倍
ユーティリティ： SQL*Loader	従来型ロード（固定長）	2.9 倍
PL/SQLパッケージ： UTL_FILE	PUT_RAW（バルク化＋固定長）	2.8 倍
SQL：DELETE	DELETE（パラレル）	2.6 倍
ユーティリティ： 外部表	select（固定長）	2.5 倍

ストレージ・アーキテクチャの刷新

- ✓ ストレージサーバで最新のNVMeフラッシュ(PCIe Gen5)を採用
- ☞ X8MのGen3フラッシュと比較して約4倍のI/O性能



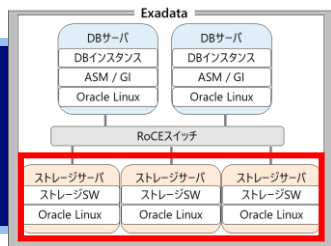
OS上のファイル操作 (**UTL FILE**/**外部表の参照**/**SQL*Plus spool出力**/**SQL*Loader**)：

本検証のファイル入出力先である ACFS（ストレージサーバ上）へのI/Oや、文字列フォーマット変換が主体となる処理であるため、
ストレージのI/O性能向上と、DBサーバでのデータパース処理の高速化(前述)が相乗効果を生んだ

3.2. 性能対策実践ノウハウ ～Exadata編～

X8MとX11Mの性能比較（考察）

- ① DBサーバの演算能力とメモリ帯域の飛躍的向上
- ② **ストレージ・アーキテクチャの刷新**
- ③ ネットワーク帯域の倍増



■ 約2.5～3倍高速化された処理の一部については、“ストレージ・アーキテクチャの刷新”などが寄与したと考察

分類	操作	高速化
SQL : ROLLBACK	大量ROLLBACK	2.9 倍
SQL : SELECT	HASH結合	2.7 倍
SQL : SELECT	NESTED LOOPS結合	2.7 倍
SQL : SELECT	MINUS集合	2.4 倍
SQL : COMMIT	COMMIT	2.4 倍
PL/SQLパッケージ : DBMS_MVIEW	完全リフレッシュ	2.4 倍
SQL : SELECT	GROUP BY	2.3 倍

ストレージ・アーキテクチャの刷新

✓ PMEM(Intel Optane)から、DRAMベースのXRMEMに変更され、応答速度が短縮 (19 μsec → 14μsec)



大量のランダムアクセスが発生する処理(大量ROLLBACK /NESTED LOOPS結合) :

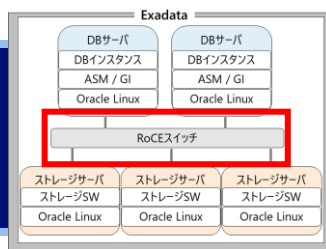
XRMEMでの応答速度短縮によりデータの取得が高速化し、さらにDBサーバの処理能力向上(前述)が相まって高速化

※ XRMEM : Exadata RDMA Memory

3.2. 性能対策実践ノウハウ ～Exadata編～

X8MとX11Mの性能比較（考察）

- ① DBサーバの演算能力とメモリ帯域の飛躍的向上
- ② ストレージ・アーキテクチャの刷新
- ③ ネットワーク帯域の倍増



■ 約2～2.5倍高速化された処理の一部については、“ネットワーク帯域の倍増”などが寄与したと考察

分類	操作	高速化
ユーティリティ： Data Pump	Data Pump Export	2.4 倍
ユーティリティ： SQL*Loader	ダイレクト・パス・ロード（固定長）	2.3 倍
ユーティリティ： SQL*Loader	ダイレクト・パス・ロード（可変長）	2.3 倍
SQL：INSERT	INSERT-VALUES（従来型）	2.2 倍
ユーティリティ： Data Pump	Data Pump Export（パラレル）	2.2 倍
PL/SQLパッケージ： DBMS_MVIEW	高速リフレッシュ	2.2 倍
SQL：索引操作	REBUILD（NOLOGGING）	2.1 倍
SQL：SELECT	TABLE ACCESS FULL	2.0 倍
SQL：SELECT	INDEX RANGE SCAN	2.0 倍
SQL：SELECT	ORDER BY	2.0 倍
SQL：INSERT	INSERT-SELECT（ダイレクト）	2.0 倍

ネットワーク帯域の倍増

✓ 内部ネットワークがPCIe Gen5対応で実効200 GbpsのRoCE（アクティブ/アクティブ構成）となり、帯域が倍増



Smart Scanをフル活用する処理(TABLE ACCESS FULL)やダイレクト・パス・ロード/インサートやDataPump Export：

- DBサーバのメモリ(SGA)をバイパスする処理について、RoCEの実効性能の倍増と、ストレージ・アーキテクチャの進化(前述)が反映され高速化
- さらに、DataPump ExportやSQL*Loaderのような外部ファイル連携を伴うユーティリティ処理においては、DBサーバにおけるパース処理の性能向上(前述)も寄与

3.2. 性能対策実践ノウハウ ～Exadata編～

X8MとX11Mの性能比較（まとめ）

- X11Mでは、最新の技術を取り入れ、各構成要素を強制的に強化することで処理性能を向上させている
 - X11M(2025年)では、X8M (2019年)に比べ約2～5倍の高速化(※)が測定された。 ※NRI検証環境での数値
この結果は、下記のアーキテクチャの進化に起因すると考察した。
 - ✓ DBサーバの演算能力とメモリ帯域の飛躍的向上：
CPUは最新の第5世代AMD EPYCを採用し、コア数も4倍に。メモリはDDR5を採用しデータ転送レートが倍増
 - ✓ ストレージ・アーキテクチャの刷新：
フラッシュ・ドライブはPCIe Gen5 NVMeを採用し、さらにPMEMからDRAMベースの超高速XRMEMへ移行
 - ✓ ネットワーク帯域の倍増：
100Gbps RoCEを、PCIe Gen5バス対応のデュアルポート・アクティブ/アクティブ構成として実効200Gbpsへ拡張
 - AIベクトル検索での「AI Smart Scan」におけるTop-Kフィルタリング処理なども強化され、従来比で最大30倍高速化

Exadataのハードウェアバージョンアップが、性能対策の一手となり得る

3.2. 性能対策実践ノウハウ ～Exadata編～

Exadata(ExaDB-D)の拡張性

■ ハードウェアバージョンアップの他にも、Exadataに備わるスケール機能によって性能対策が可能

✓ オンラインCPUスケールアップ

- DBを停止することなく、マネジメントコンソールやOCI CLIからスケールアップ/ダウンが可能
- 必要なタイミングだけ性能を增強することで、ランニングコストを最適化

✓ リソースのエラスティックな拡張

- 必要に応じてDBサーバ/ストレージサーバのスケールアウトが可能（DBサーバは2～32台 / ストレージサーバは3～64台）
 - SGA/PGAの不足や、大量の同時接続・演算処理がボトルネックとなる場合：DBサーバを追加
 - ディスクI/O帯域幅や並列処理能力を引き上げたい場合：ストレージサーバを追加

アプリケーション改修を伴うことなく、インフラレイヤで性能対策が可能

3.2. 性能対策実践ノウハウ ～Exadata編～

Exadataの性能対策実践ポイント

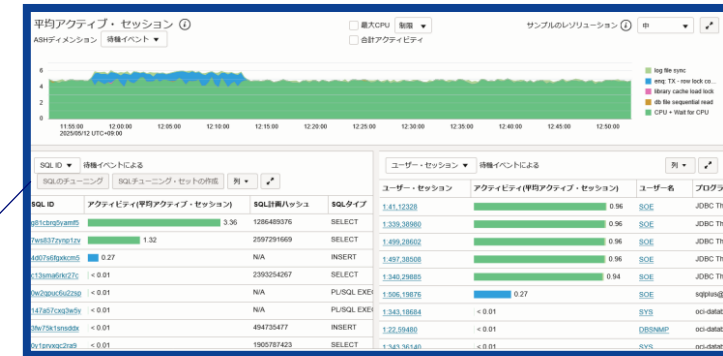
■ 局所的なボトルネックが存在する場合はDatabase Managementサービスで視覚的かつ迅速に特定

性能対策における実践ポイントは...

- ◆ 最新Exadataではハードウェアが格段に強化されているため、まずは早期にアプリケーションを乗せて稼働させてみる
- ◆ 将来の需要を満たすCPU/ディスクIO性能が確保できることを確認

- Exadata関連の統計はAWRの“Exadata Resource Statistics”セクションから確認可能
- ディスクIOが不足しそうな場合はストレージサーバの台数を追加する などの拡張を実施

これだけでも十分な性能が得られる可能性が高いが...



<https://speakerdeck.com/oracle4engineer/oci-database-management>

◆ もし局所的にSQL性能が気になる場合は、Database Managementサービスを活用

- DB全体のうち高負荷のSQLの特定（トップダウンアプローチ）と、特定の遅い処理の詳細調査（ボトムアップアプローチ）の両方が画面からシームレスに実施可能
- ASHのサンプリングが1秒おきであるため、AWRでは特定が難しい 短時間で完了するSQLの検知・分析も可能

圧倒的なハードウェアで全体性能を底上げし、
局所的なボトルネック対策にはDatabase Managementサービスを活用

3. Exadata&Compute性能対策実践ノウハウ

まとめ

■ OCIを活用した“拡張性のあるインフラ”を構築することで、アプリケーション改修を伴わない性能対策が可能

- ✓ Compute：各種性能上限の特性を理解することで、柔軟なスケールアップを実施してボトルネックを解消できる
- ✓ Exadata：AI・最新ワークロードにも対応する桁違いの物理性能(X11Mの進化)を基盤として活用できる
- ✓ OCIは進化を継続しており、最新のハードウェア・技術を取り入れることで性能改善を見込むことができる

OCIならではの柔軟なスケラビリティと、Oracle DBに最適化されたExadataの圧倒的な処理能力を組み合わせることで、「コストを最適化しつつ、将来のあらゆる性能要件の増大にも耐える基盤」の構築が可能



**Envision the value,
Empower the change**