

ORACLE®

オラクル・コンサルが語る！ データベース・サーバー 集約/統合の極意

日本オラクル株式会社
テクノロジーコンサルティング統括本部
テクニカルアーキテクト本部
基盤ソリューション部
シニアコンサルタント
岩切 友樹



 #odddtky

日本オラクル、今年最大の技術トレーニングイベント

**Oracle DBA &
Developer Day 2013**

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。

Program Agenda

- Introduction
 - データベース・サーバー集約/統合
 - 本セッションについて
- Exadata Consolidation Tips
 - CPUに関するTips
 - Memoryに関するTips
 - Diskに関するTips
 - OS／Databaseに関するTips
 - その他のTips

Introduction

データベース・サーバー 集約/統合



- Mixed Workloads
- Data Warehousing
- OLTP

データベース・サーバー集約/統合

Why データベース・サーバー集約/統合？

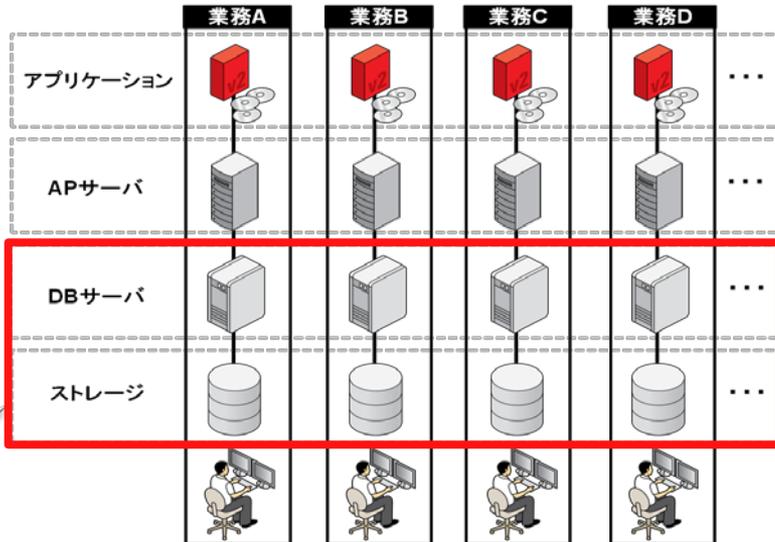
今日統合(Consolidation)が非常に注目度の高い理由として、
以下のような問題解決が目的であると考えられます。

ITコストの増大

個々のシステムごとに
ばらばらにサーバーを調達

システムの信頼性

個々のシステムごとに
HA構成を検討



セキュリティ

個々のシステムごとに
セキュリティ対策を検討

今回お話する内容

データ品質

システム間のデータの
やり取りに伴う手間

データベース・サーバー集約/統合

データベース・サーバー集約/統合の方法論

- スキーマ統合
 - 1つのデータベースを複数のアプリケーションが共有する構成
- サーバー統合
 - 1つのサーバーを複数のデータベースが共有する構成
- マルチテナント化 ★New in 12c
 - 1つのコンテナ・データベースを複数のプラグブル・データベースが共有する構成
- 仮想化 ★On Exadata Roadmap-Future
 - 1つの物理マシンを複数の仮想マシンが共有する構成

今回お話しする内容

データベース・サーバー集約/統合

サーバー統合とは？

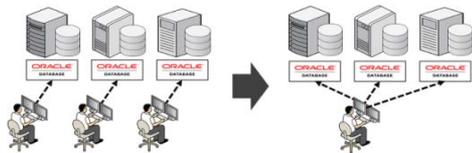
- ハードウェアの集約による統合を意味し、大きくは3つのアプローチがあります。

※サーバー統合の理想形は「サーバー物理統合」です。

今回お話する内容

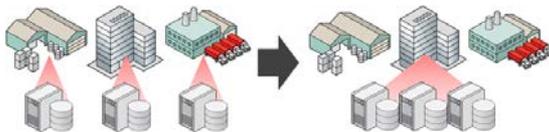
サーバー論理統合

- 複数のサーバーを論理的に1台の大きなサーバーと見なす
 - 1台もしくは複数台のサーバーにデータベースを論理的に1つに見せる (例: データベースリンク)
 - OSやミドルウェアのバージョンの統一化、標準化等
 - 集中管理により、運用管理負荷を軽減



サーバー位置統合

- 物理的に分散したサーバーを1ヶ所または数箇所に集約配置
 - 集中管理により、運用管理負荷を軽減
 - 専用のサーバー・ルームでの設置によりH/Wの故障率を低減
 - ネットワーク・インフラの再構築が必要なケース有り



サーバー物理統合

- 物理的に分散したサーバーを1台または数台にサーバー集約
 - サーバー台数の削減による運用管理コストを削減
 - サーバー・リソースの使用率向上
 - どのレベルで集約するのかがポイント
 - サーバー環境をまるごと？
 - データベースも集約？



ORACLE

データベース・サーバー集約/統合

サーバー統合におけるExadataの優位性



- 大量データ処理基盤向けの特長
 - Smart Scanによってサーバー⇄ストレージ間のI/O量を削減
 - Storage IndexによってストレージI/O量を削減
 - InfiniBandによる超高速ネットワーク
 - 必要なストレージ容量とI/O量を削減可能な列圧縮
- オンライン処理基盤向けの特長
 - 高いスケーラビリティを持つGrid Infrastructure (RAC, ASM)
 - Smart flash cacheによる極めて高いRandom I/O性能
 - 過去履歴データの保持コストを最大1/50に圧縮
 - 高いDBセキュリティとMAA (Maximum Availability Architecture) による高い業務継続性 (RAC, Data Guard等)
- システム統合基盤向けの特長
 - あらゆるワークロードに対して最適なパフォーマンスと拡張性を提供
 - マルチDB, マルチAP、及びマルチユーザ環境でも、安定したレスポンスを提供

本セッションについて



- Mixed Workloads
- Data Warehousing
- OLTP

本セッションについて

本セッションの内容

- オラクルコンサルタントが関わっているExadataのプロジェクトにおいて、複数のプロジェクトで話題となった内容をTipsとしてご紹介します。
- ExadataのバージョンはX3, OSはOracle Linux 5, Oracle Grid Infrastructure/Databaseのバージョンは11g Release2を想定しています。

Exadata Consolidation Tips (合計: 10個)

- ✓ CPUに関するTips ... 1個
- ✓ Memoryに関するTips ... 5個
- ✓ Diskに関するTips ... 1個
- ✓ OS/Databaseに関するTips ... 2個
- ✓ その他のTips ... 1個

本セッションについて

Tipsの構成

- それぞれのTipsはおおむね以下の内容で構成されています。

1.実際にプロジェクトであった話(会話形式)

2.Tipsに関連する技術情報のご紹介

3.まとめ

4.実プロジェクトの事例(参考例)

5.参考情報

Exadata Consolidation Tips

CPUに関するTips

✓ Tips1. インスタンス・ケーシングの利用



- ✓ Mixed Workloads
- ✓ Data Warehousing
- ✓ OLTP

Tips1.インスタンス・ケーシングの利用

事例で分かるインスタンス・ケーシングの重要性



プロジェクト
ご担当者様
(インフラ)

Exadata上で複数のデータベースを稼働していますが、あるデータベースでCPU使用率が高騰し、他のデータベースの処理がCPUネックで遅延しはじめているようです。どうしたらいいのでしょうか？



プロジェクト
ご担当者様
(インフラ)

CPU使用率が高騰しているデータベースについて、インスタンス・ケーシングの設定はしていますか？インスタンス・ケーシングはデータベースを停止することなく動的に設定可能です。インスタンス・ケーシングの設定を試みてはいかがでしょうか？



オラクル
コンサル

了解しました。取り急ぎ、CPU使用率が高騰しているデータベースにインスタンス・ケーシングの設定を試みます。

✓ 統合環境では特定のデータベースの処理が他のデータベースの処理を圧迫しないよう、リソース使用率の上限値を設定することが有効です。

Tips1.インスタンス・ケーシングの利用

インスタンス・ケーシング (Partitioningアプローチ)



- 使用可能なCPUスレッド数の上限を超えないよう、各インスタンスにcpu_countを設定
 - 合計値(cpu_count) <= CPUのスレッド数の合計値
- 各インスタンス間の独立性を確保
 - 各インスタンス間でCPU競合が発生しない。
 - 特定のインスタンスがidle状態の場合、そのインスタンスに割り当てられたCPUは未使用となる。
- 本番環境など、同一筐体上の他インスタンスとの独立性が要求される場合やパフォーマンス要件がクリティカルな環境での利用を推奨

cpu_countの設定値
(CPUのスレッド数の合計値:32)

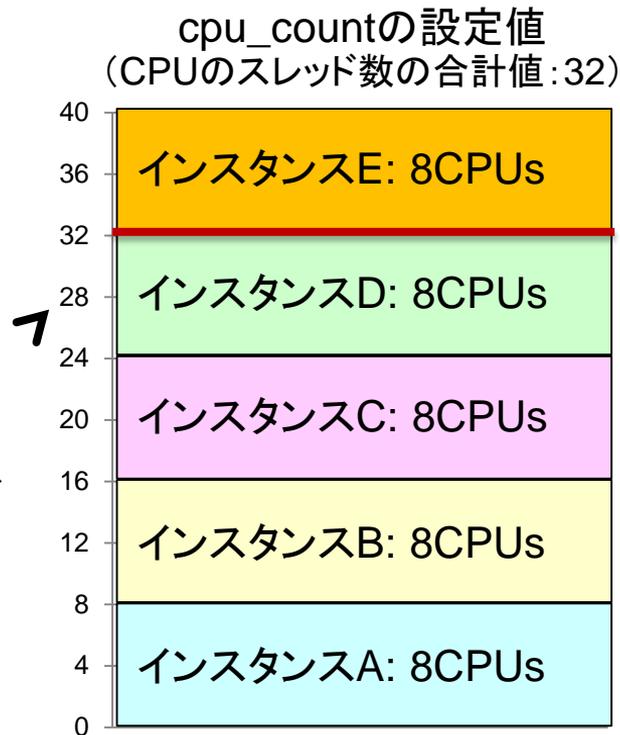


Tips1.インスタンス・ケーシングの利用

インスタンス・ケーシング (Over-Provisioningアプローチ)



- 使用可能なCPUスレッド数の上限を超え、各インスタンスにcpu_countを設定 (比率で制御)
 - 合計値(cpu_count) > CPUのスレッド数の合計値
- 各インスタンス間の独立性はないが、CPUリソースの有効活用が可能
 - 他のインスタンスのCPU使用状況に依存するため、CPU競合が発生する可能性がある。
 - 特定のインスタンスがidle状態の場合、そのインスタンスに割り当てられたCPUを別のインスタンスが使用可能。
- 開発環境やテスト環境など、厳密なパフォーマンス要件がない環境での利用を推奨



Tips1.インスタンス・ケーシングの利用



まとめ

■ インスタンス・ケーシングの利用

- 同一サーバー上に複数のインスタンスを構成する場合、インスタンス・ケーシングを使用して各インスタンスごとにCPU使用量を制限する設計が有効です。
- 他インスタンスとの独立性が要求される場合やパフォーマンス要件がクリティカルな場合、Partitioningアプローチによる設計／実装を推奨します。
- インスタンス・ケーシングはデータベースを停止することなく動的に変更可能です。
- 全体の使用率の確認(OS Watcher)や各DBごとの使用率の確認(AWRLレポートの結果)をインプットに適宜cpu_countの割当値を見直すことを推奨します。

Tips1.インスタンス・ケーシングの利用

実プロジェクトでのサイジング結果例(CPU使用量)



- オラクルコンサルタントがExadataの導入を行う場合、初期設計時点ではSPECintや現行システムの使用率をインプットにcpu_countの値を算出することがあります。

※以下はExadata X3-2(1CPU/8Core/16スレッド) × 2個 のサイジング結果例です。

◆現行システム

環境名	業務システム名	筐体名	CPU	SPECint rate2006	CPU使用率 (最大)	値
本番環境	業務システムA	Sun Fire V490	UltraSPARC IV+ 2.1GHz (1CPU/2Core/4スレッド) × 4個	78	85%	66
	業務システムB	Sun Fire V890	UltraSPARC IV+ 2.1GHz (1CPU/2Core/4スレッド) × 8個	154	90%	139
	業務システムC	Sun Fire V890	UltraSPARC IV+ 2.1GHz (1CPU/2Core/4スレッド) × 8個	154	80%	123

◆新システム

環境名	業務システム名	筐体名	CPU	SPECint rate2006	cpu_count	値
本番環境	業務システムA	Sun Server X3-2	Intel Xeon E5-2690 2.9GHz (1CPU/8Core/16スレッド) × 2個	702	4	88
	業務システムB				8	176
	業務システムC				6	132

18 / 32 (合計値)

(補足) 計算結果は1ノードあたりのサイジング結果例です。縮退時も考慮したサイジングを検討します。

Tips1.インスタンス・ケーシングの利用

【ご参考】SPECとは？ SPECintとは？



■ SPEC(Standard Performance Evaluation Corporation)とは？

- コンピューターの性能を評価するために使われるベンチマークテストの標準化を目的とした団体です。
- CPU性能、数値演算性能、Web性能など、さまざまなベンチマークプログラムの管理・配布を行っており、コンピューターを実務で使用した場合に近い性能が確認可能です。SPECのベンチマークプログラムで得られた評価値をSPEC値と呼びます。<http://www.spec.org/>

■ SPECint(Standard Performance Evaluation Corporation Integer benchmark)とは？

- SPECが策定した、システムの性能評価を行うベンチマークのひとつです。整数演算を実行するプログラムにより、性能を評価しています。<http://www.spec.org/cpu2006/CINT2006/>

Standard Performance Evaluation Corporation

Home | Benchmarks | Tools | Results | Contact | Site Map | Search | Help

Home | Benchmarks | Tools | Results | Contact | Site Map | Search | Help

- Home
- Benchmarks
 - GPU
 - Computing/Workstations
 - WebServer
 - Java Client/Server
 - Math/Science
 - Network File System
 - Power
 - DB
 - SQL
 - Virtualization
 - Web Services
- Tools
- Results/Search
- Contact Us
- Help

Standard Performance Evaluation Corporation

Home | Benchmarks | Tools | Results | Contact | Site Map | Search | Help

Home | Benchmarks | Tools | Results | Contact | Site Map | Search | Help

CINT2006 (Integer Component of SPEC CPU2006):

Benchmark	Language	Application Area	Brief Description
403.benchmark	C	Programming Language	Derived from Fort 95.7. The workload includes SparseMatrix, Mvaddc (for small integers), and speed (SPEC's test that checks benchmarks' outputs).
401.int	C	Compression	Julian Gemell's bz2 test version 1.0.2, modified to do most work in memory, rather than doing I/O.
402.gcc	C	C Compiler	Based on gcc Version 3.2, generates code for Octopus.
404.mcf	C	Operational Optimization	Various scheduling tests, a network simulator algorithm (which is also used in commercial products) to schedule public transport.
400.gemspec	C	Artificial Intelligence	From the game of Go, a simply described but deeply complex game.
405.tpcsv	C	Search Game Sequence	Pattern sequence analysis using profile hidden Markov models (profile HMMs).
405.wmg	C	Artificial Intelligence Chess	A high-ranked chess program that also gives several chess variants.

Additional Information:

- 401.int
- 402.gcc
- 404.mcf
- 400.gemspec
- 405.tpcsv
- 405.wmg
- 403.benchmark

Memoryに関するTips

- ✓ Tips2.メモリ使用量のサイジング(全体)
- ✓ Tips3.メモリ使用量のサイジング(DB)
- ✓ Tips4.データベースのメモリ管理方式
- ✓ Tips5.SGAの最小サイズの設定
- ✓ Tips6.HugePagesの設定



- ✓ Mixed Workloads
- ✓ Data Warehousing
- ✓ OLTP

Tips2.メモリ使用量のサイジング(全体)

Tips3.メモリ使用量のサイジング(DB)



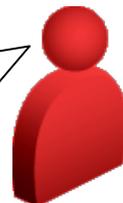
事例で分かる統合環境におけるメモリ使用量のサイジング



プロジェクト
ご担当者様
(インフラ)

これからExadata上に複数のデータベースを統合しようと思います。物理メモリはどうサイジングしたらいいのでしょうか？

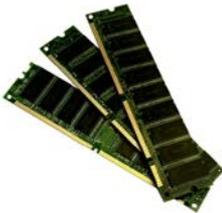
メモリのサイジングは、大まかに2つのステップで考えます。
まずメモリ全体の使用量(OS/GI/DB/余裕分の割合)を考えます。★Tips2
次にDBのメモリ使用量をサイジングします。複数のデータベースを構築する場合、各DBごとに個別にサイジングする必要があります。★Tips3
サイジングは現行システムの値やExadataのBest Practicesを参考にするといいです。



オラクル
コンサル

[Doc ID 1274318.1] Oracle Sun Database Machine Setup/Configuration Best Practices

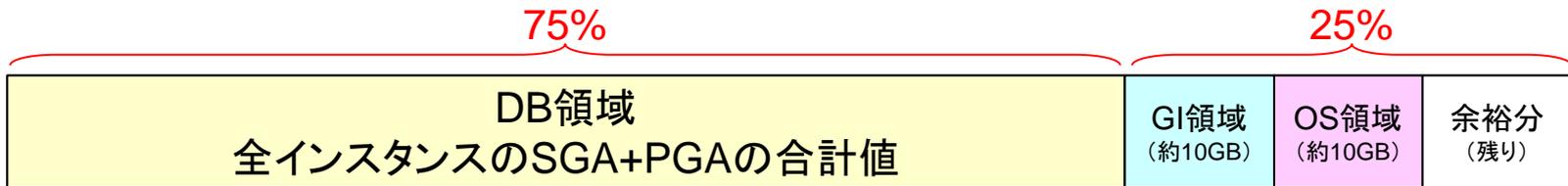
- Verify OLTP Instance Database Initialization Parameters
- Verify DW/BI Instance Database Initialization Parameters



Tips2.メモリ使用量のサイジング(全体)

全体メモリ使用量のサイジング(DB:75%/DB以外:25%)

- データベースへのメモリ割り当ては搭載メモリの75%を上限とする。
 - 同一サーバー上に複数のインスタンスを構成する場合、各インスタンスに割り当てるメモリ設定パラメータ(SGA + PGA)の合計が、搭載メモリの75%を超えないようにサイジングします。
- 残りの25%はOS、Grid Infrastructure、余裕分とする。
 - 搭載メモリの約10GBはOSが使用する領域として計算します。
 - 搭載メモリの約5~10GBはGrid Infrastructureが使用する領域として計算します。
 - 残りは余裕分とします。搭載されているメモリ容量の約5%程度は確保します。





Tips3.メモリ使用量のサイジング(DB)

各データベースのメモリ使用量のサイジング(OLTP系、DWH/BI系)

■ OLTP系のデータベースの場合

- メモリ使用量は「 $\text{sga_target} + \text{pga_aggregate_target} + 4\text{MB} * \text{processes}$ 」
- 複数のクライアントアクセスが想定されるオンライン処理が含まれるデータベースの場合、プロセスが使用するメモリ領域($4\text{MB} * \text{processes}$)を考慮します。

■ DWH/BI系のデータベースの場合

- メモリ使用量は「 $\text{sga_target} + (\text{pga_aggregate_target} * 3)$ 」
- 大量データの集計やソート、結合を行うバッチ処理が含まれるデータベースの場合、 $\text{pga_aggregate_target}$ の設定値の3倍程度までPGAが拡大することを考慮する必要があります。

DB領域 全インスタンスのSGA+PGAの合計値	GI領域 (約10GB)	OS領域 (約10GB)	余裕分 (残り)
-----------------------------	-----------------	-----------------	-------------



Tips3.メモリ使用量のサイジング(DB)

【ご参考】SGA/PGAの推奨サイズ

- 初期化パラメータの設定 (sga_target / pga_aggregate_target)
 - オラクルコンサルタントがExadataの導入を行う場合、初期設計時点ではBest Practicesの推奨サイズを設定し、テストフェーズでAWRレポートや各種アドバイザ機能を使用してチューニングするアプローチをとることがあります。

区分	OLTP系		DWH/BI系	
	SGA	PGA	SGA	PGA
Exadata X2-2 , X3-2	24GB	16GB	16GB	16GB
X2-8 , X3-8	128GB	64GB	128GB	256GB

(補足)サイズは1インスタンスあたりの推奨サイズです。

[Doc ID 1274318.1] Oracle Sun Database Machine Setup/Configuration Best Practices

- Verify OLTP Instance Database Initialization Parameters
- Verify DW/BI Instance Database Initialization Parameters



Tips2.メモリ使用量のサイジング(全体)

Tips3.メモリ使用量のサイジング(DB)

実プロジェクトでのサイジング結果例(メモリ使用量)

環境名	用途		計算式	計算結果		
				現行	新規	
本番環境	余裕率		搭載メモリの約5%程度	-	12.8	
	OS		約10GB	-	10.0	
	Grid Infrastructure		約10GB	-	10.0	
	Oracle Database	業務システムA (DWH)	SGA	SGA_TARGET	4.0	16.0
			PGA	PGA_AGGREGRATE_TARGET × 3	12.0	48.0
		業務システムB (BI)	SGA	SGA_TARGET	4.0	16.0
			PGA	PGA_AGGREGRATE_TARGET × 3	12.0	48.0
		業務システムC (OLTP)	SGA	SGA_TARGET	1.5	24.0
			PGA	PGA_AGGREGRATE_TARGET	4.5	16.0
		Process使用量		Max Process数 × 4MB	2.0	4.0

使用メモリ **204.8 GB**
 余剰メモリ **51.2 GB**

(補足) 計算結果は1ノードあたりのサイジング結果例です。縮退時も考慮したサイジングを検討します。

Tips4.データベースのメモリ管理方式



自動共有メモリ管理(SGA)+自動PGAメモリ管理(PGA)の採用

- 自動共有メモリ管理(SGA)／自動PGAメモリ管理(PGA)を採用
 - SGA／PGAを構成する各コンポーネントの個別サイズ指定を必要とせず、サイジングが比較的容易に可能。各コンポーネントのサイズは必要に応じて動的に調整される。
 - データベースの動作を安定させるため、SGAの各コンポーネントの最小サイズを指定することを推奨します。★「[Tips5.SGAの最小サイズの設定](#)」を参照
- 自動メモリ管理(SGA／PGA)は採用しない
 - Oracle Linuxの場合、自動メモリ管理とHugePagesを併用できません。
 - 自動メモリ管理を設定している場合、SGAのメモリ割り当てにはtmpfs(デフォルトサイズ: 4KB)が使用される。HugePages用に予約されたメモリ領域がデータベース用(SGA)に使用されず、メモリを無駄に消費することになり、OSやアプリケーションの安定稼働に影響を及ぼす可能性があります。

Tips5.SGAの最小サイズの設定

事例で分かる最小サイズを設定することの重要性



プロジェクト
ご担当者様
(アプリ)

本番稼働してだいぶ経ちます。しばらく安定稼働していたのですが、昨日、突然アプリケーションの性能が劣化したため、エンドユーザーから調査して欲しいと依頼されました。皆目見当がつかず、一体どうしたらいいのでしょうか？

了解しました。ちょっと分析してみますね。(しばらく調査・・・)

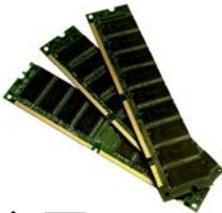
原因が分かりました。まず、AWRレポートより、一昨日と比較して昨日は処理量が増加していることが分かりました。次に、SGAの各コンポーネントの使用率を確認した所、一昨日と比較してバッファキャッシュのサイズが大幅に減っており、他のコンポーネントのサイズが拡張されていました。これまではバッファキャッシュ上で処理されていたデータがバッファキャッシュ上で処理されなくなったため、アプリケーションの性能が劣化した可能性が高いです。



オラクル
コンサル

- ✓ 処理量の増加やアプリケーションの改修による突発的な性能劣化を防ぐため、SGAの主要コンポーネント(バッファキャッシュ、共有プール)の最小サイズを指定することを推奨します。

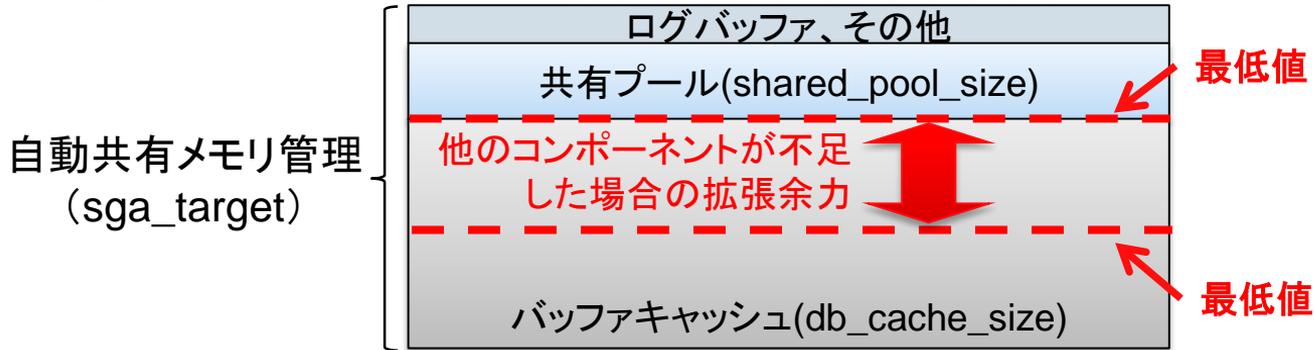
ORACLE



Tips5.SGAの最小サイズの設定

まとめ

- 突発的な性能劣化を防止し、データベースの動作を安定させるため、SGAの主要コンポーネントであるバッファキャッシュと共有プールに最小サイズを設定することを推奨します。
 - 初期設計時点で最小値の算出が困難な場合、テスト／運用フェーズで AWR レポートの結果や各種アドバイザ機能を使用して算出することを検討して下さい。
 - 自動共有メモリ管理 (SGA) の動作として、共有プール等のコンポーネントの領域が不足した場合、バッファキャッシュの領域から割り当てられます。





Tips5.SGAの最小サイズの設定

【ご参考】バッファキャッシュ、共有プールの最小サイズの算出方法

■ AWRLレポートの結果から最小サイズを算出

– Cache Sizes セクション

AWRLレポートのCache Sizes セクションで実際に使用しているサイズを確認可能です。
この結果をインプットに各コンポーネントの最小サイズの値を決定します。

– Buffer Pool Advisory セクション、Shared Pool Advisory セクション

AWRLレポートには各種アドバイザの診断結果が出力されており、提示された値に設定することによる効果も確認可能です。この診断結果も最小サイズを設定するための有効な情報となります。

Cache Sizes			
	Begin	End	
Buffer Cache:	11,712M	11,712M	Std Block Size: 8K
Shared Pool Size:	7,104M	7,104M	Log Buffer: 171,004K

Buffer Pool Advisory							
<ul style="list-style-type: none"> Only rows with estimated physical reads >0 are displayed ordered by Block Size, Buffers For Estimate 							
P	Size for Est (M)	Size Factor	Buffers (thousands)	Est Phys Read Factor	Estimated Phys Reads (thousands)	Est Phys Read Time	Est %DBTime for Rds
D	1,152	0.10	136	1.82	607,911	1	200584.00
D	2,304	0.20	272	1.41	472,197	1	129676.00
D	3,456	0.30	408	1.26	420,755	1	102799.00
D	4,608	0.39	544	1.19	396,385	1	90067.00
D	5,760	0.49	680	1.13	379,063	1	81016.00
D	6,912	0.59	816	1.09	364,995	1	73666.00
D	8,064	0.69	952	1.06	354,500	1	68183.00
D	9,216	0.79	1,088	1.04	347,684	1	64622.00
D	10,368	0.89	1,224	1.02	339,898	1	60554.00
D	11,520	0.96	1,360	1.00	335,121	1	58058.00

Tips6.HugePagesの設定

事例で分かるHugePagesを設定することの重要性



プロジェクト
ご担当者様
(インフラ)

本番稼働してだいぶ経ちます。しばらく安定稼働していたのですが、今朝、突然アプリケーションの性能が徐々に遅くなり、DBサーバがノードダウンしました。冗長化構成のため、業務停止までには至りませんでした。ユーザーから至急調査して欲しいと要望いただいています。皆目見当がついておらず、一体どうしたらいいのでしょうか？

了解しました。ちょっと調査してみますね。(しばらく調査・・・)

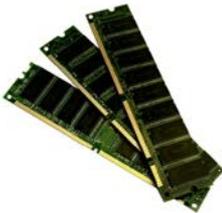
原因が分かりました。まず、OS Watcherの実行結果 (vmstat) より、DBサーバで物理メモリの不足からスワップイン／アウトが多発していたことが分かりました。スワップイン／アウトが多発していた原因ですが、OS Watcherの実行結果 (meminfo) より、ページテーブルが肥大化していることに起因しています。



オラクル
コンサル

✓ HugePagesを設定することにより、ページテーブルのエントリ数 (PTE数) が少なくなり、ページテーブルの肥大化を防止することが可能です。

[Doc ID 361323.1] HugePages on Linux: What It Is... and What It Is Not...



Tips6.HugePagesの設定

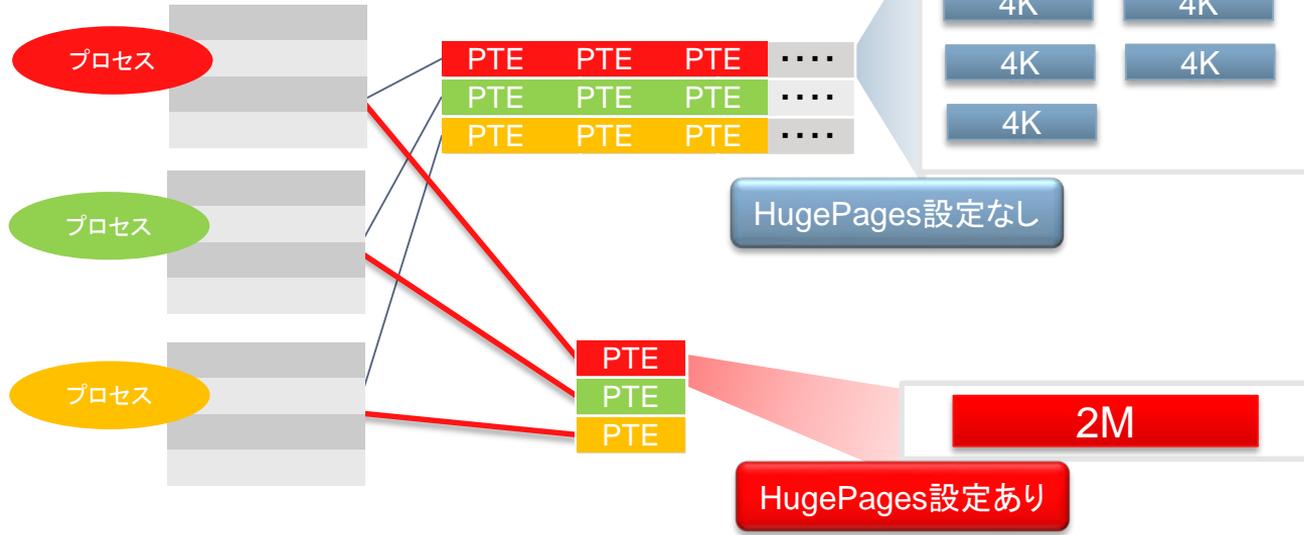
HugePagesを利用した場合、利用しない場合

仮想メモリ

Page Table

物理メモリ

※Page Table:仮想アドレスから、物理アドレスへのマッピング情報を保持



32GBのメモリをマップする場合

ページ数: $32\text{GB}/4\text{K}=8388608$
総PTEのサイズ: $1\text{PTEサイズ} \times 8388608$

↓ 150セッション存在した場合

総PTEのサイズ: **最大66GB**

ページ数: $32\text{GB}/2\text{M}=16384$
総PTEのサイズ: $1\text{PTEサイズ} \times 16384$

↓ 150セッション存在した場合

総PTEのサイズ: **最大131MB**

Tips6.HugePagesの設定



まとめ

- Exadata は搭載されている物理メモリの容量が大きいため、ASM、Database の SGA は HugePages 上に構成することが推奨されています。
 - HugePages を利用しない場合、OS の PTE で利用するメモリサイズが非常に大きくなり、物理メモリの不足からスワップが多発し、大幅なスローダウンやノード・ダウンが発生する可能性があります。
 - HugePages を利用することにより、ページテーブルのエントリ数 (PTE数) が少なくなり、ページテーブルの肥大化を防止することが可能です。
 - HugePages の推奨値は以下の Note に記載されているシェルスクリプトを実行することにより算出可能です。
 - インスタンス数が増減したタイミング、SGA のサイズを変更したタイミングで適宜見直す必要があります。

[Doc ID 401749.1] Shell Script to Calculate Values Recommended Linux HugePages / HugeTLB Configuration

Diskに関するTips

✓ Tips7.IORMの利用



- ✓ Mixed Workloads
- ✓ Data Warehousing
- ✓ OLTP



Tips7.IORMの利用

事例で分かるIORMを設定することの重要性



プロジェクト
ご担当者様
(インフラ)

Exadata上で複数のデータベースを稼働していますが、あるデータベースでバッチ処理によるI/O処理量が非常に多く、OLTP系の処理や他のデータベースの処理がI/Oネックで遅延しはじめているようです。複数のデータベースを安定稼働させる上で適切な方法はありますか？

データベースに対してI/O使用量の制限を設けてはいかがでしょうか？

IORMを使用することで、特定のデータベースによるI/O処理の専有を防ぐことができます。また、I/O競合が発生した場合に備え、最低限使用できるI/O使用量の設定も可能です。



プロジェクト
ご担当者様
(インフラ)

了解しました。現在のデータベースのI/O使用量を考慮した上で、IORMを設定してみます。



オラクル
コンサル

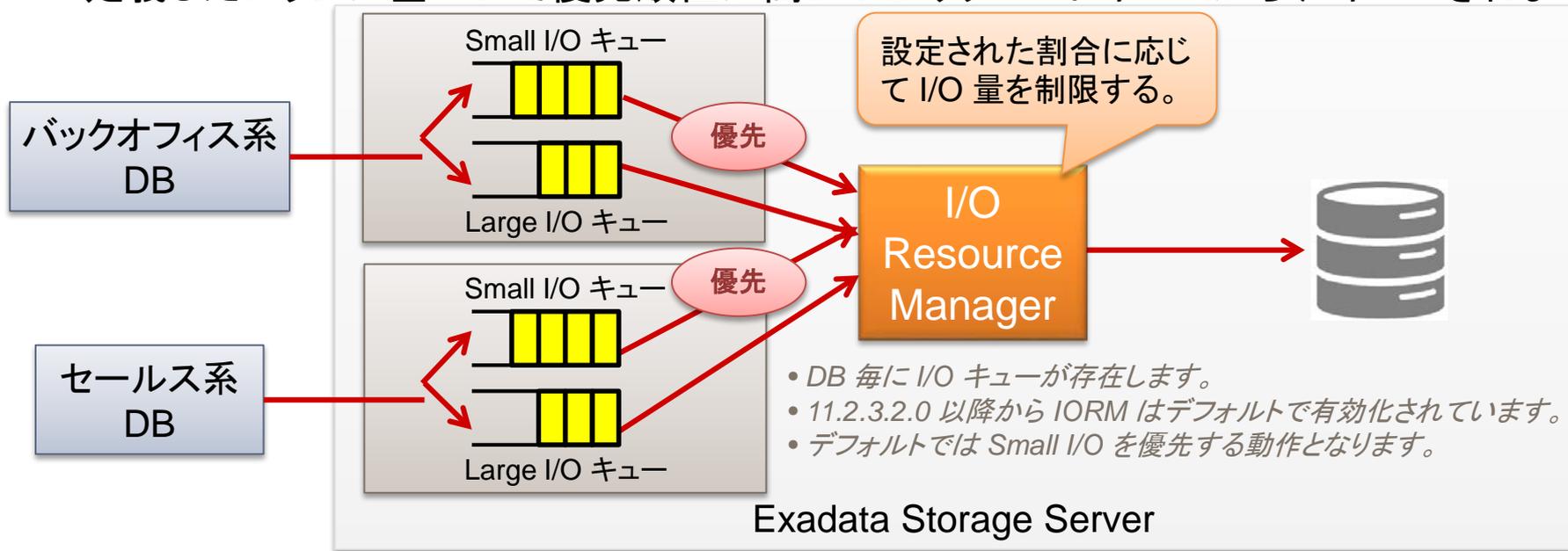
✓ インスタンス・ケーシングと同様、統合環境では特定のデータベースの処理が他のデータベースの処理を圧迫しないよう、I/O使用率の上限値を設定することが有効です。

Tips7.IORMの利用

IORM(I/O Resource Manager)のアーキテクチャ



- DB サーバーから送られてくる I/O リクエストをキューを用いて管理します。
- 定義したプランに基づいて優先順位が高い I/O リクエストキューからデキューされます。





Tips7.IORMの利用

IORMの主な特徴、設定項目

- Disk I/Oの管理単位
 - カテゴリ単位、データベース単位(データベース間プラン)、データベース内のコンシューマ・グループ単位(データベース内プラン)の3つの単位で指定することができます。
- Disk I/Oの優先度の設定(allocation/share)
 - パーセンテージを使用した I/O 優先度の設定(allocation)、または割合を使用した I/O 優先度の設定(share)が可能です。
- Disk I/Oの上限値の設定(limit)
 - Disk I/O 使用率の上限値を設定することができます。
 - Disk I/O 優先度の設定(allocation/share)と併用することができます。
- Disk I/O処理の最適化の目的を設定(objectiveパラメータ) ★11.2.2.1~
 - レイテンシー重視(low_latency)、スループット重視(high_throughput)、双方のバランスを取る(balanced)のいずれかを設定することができます。

Oracle Exadata Storage Server Software User's Guide 11g Release 2 (11.2)

- 6 Managing I/O Resources ~ 8 Using the CellCLI Utility

Tips7.IORMの利用

まとめ



- IORM (I/O Resource Manager) の利用
 - 同一サーバー上に複数のデータベースを構成する場合、IORMを使用して各データベースごとに I/O 使用量を制限する設計が有効です。
 - I/O Resource Manager はデータベースを停止することなく動的に変更可能です。
 - Disk I/O 優先度の設定 (allocation / share) と Disk I/O 上限値の設定 (limit) を併用することを推奨します。★[次ページ参照](#)
 - 設定後の監視について、Storage サーバーの Disk I/O の使用状況は Enterprise Manager の「Exadata IORM DBメトリック」上で確認できます。



Tips7.IORMの利用

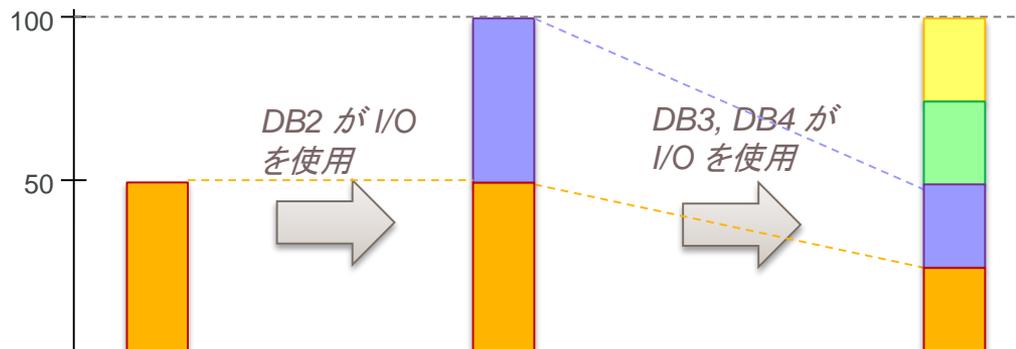
実プロジェクトでのIORMの設定事例

- Exadata Half Rack 上に4つのデータベースを構成
- Disk I/O 優先度の設定 (allocation) と Disk I/O 上限値の設定 (limit) を実装

設定内容

DB	limit	alloc
1	50%	25%
2	50%	25%
3	50%	25%
4	50%	25%

I/O 使用率の動作



DB1 が上限値の
50% 使用

DB2 が上限値の 50% 使
用することで全体として
100%

DB3, DB4 が I/O を使用すると、下
限保証値が有効となるため、25%
ずつ使用される

OS／Databaseに関するTips

- ✓ Tips8. ORACLE_HOMEの配置方針
- ✓ Tips9. サービスの分割方針



- ✓ Mixed Workloads
- ✓ Data Warehousing
- ✓ OLTP

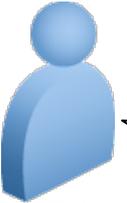
Tips8.ORACLE_HOMEの配置方針

事例で分かるORACLE_HOMEの配置方針、ログメンテナンスの重要性



A社ご担当者様
(インフラ)

Exadata上で4つのデータベースを稼働しています。各データベースごとの独立性を持たせるため、ORACLE_HOMEは各データベースごとに分割しています。4つのデータベースは密接に連携しているため、パッチ適用の際は全データベース同時にパッチ適用する運用をしています。メンテナンス時間も長くなりがちで運用が大変です。



B社ご担当者様
(インフラ)

- ✓ DB サーバーの内蔵ディスク領域 (/u01領域) が使用率100%になり、DBサーバーにログインできない障害が発生しました。
- ✓ 先日の計画メンテナンスの際、OSバックアップの取得をしたのですが、OSバックアップの取得に非常に時間がかかりました。色々調査した結果、監査ファイルが大量に生成されており、時間がかかっていたようでした。

- ✓ OneCommand で作成されるデフォルトのORACLE_HOMEを複数のデータベースで共有する構成を検討します。
- ✓ ログファイルの肥大化による内蔵ディスク領域の逼迫を防ぐため、ログローテーションを実装し、定期的にログファイルをメンテナンスする必要があります。

Tips8.ORACLE_HOMEの配置方針

ORACLE_HOMEの分割方針

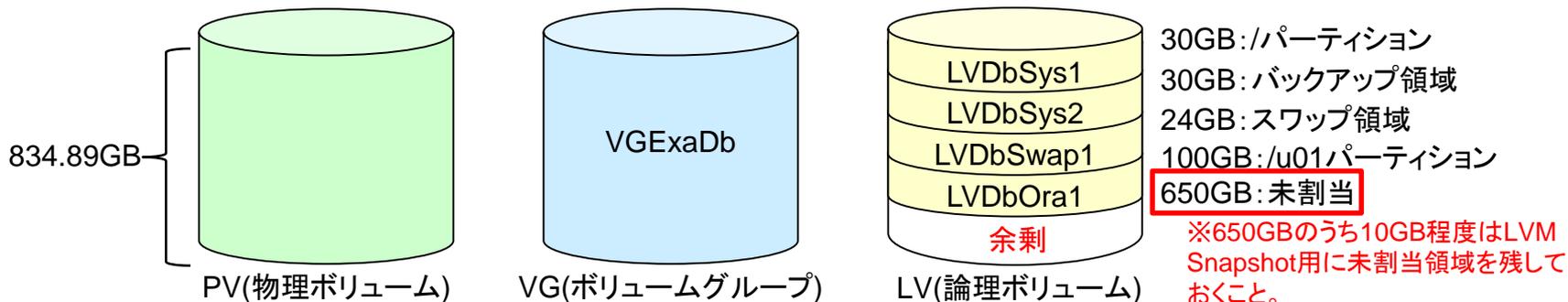
- 1つのORACLE_HOMEを複数のデータベースで共有する構成を検討します。
 - ORACLE_HOMEをシステムごとに分割して独立性を持たせるメリットよりも、ORACLE_HOMEの数が多くなることによりパッチ適用等のメンテナンス負荷が高くなるデメリットの方が大きい傾向にあります。
- パッケージ・ソフトウェアの導入要件によりORACLE_HOMEの分割が必要な場合、極力最小限の数に分割することを推奨します。サービスレベルが同じデータベース間でORACLE_HOMEを共有することを検討して下さい。
- 複数のORACLE_HOMEを持つ場合、DBサーバの内蔵ディスク領域の拡張を検討します。
 - 拡張できるサイズはExadataのバージョンによって異なります。
- ログファイルの肥大化による内蔵ディスク領域の逼迫を防ぐため、ログローテーションを実装し、定期的にログファイルをメンテナンスする必要があります。

[Doc ID Doc ID 1298957.1] Manage Audit File Directory Growth with cron

Tips8.ORACLE_HOMEの配置方針

DBサーバの内蔵ディスク領域のLVM拡張(/u01領域)

- 複数のORACLE_HOMEを持つ場合、DBサーバの内蔵ディスク領域(/u01領域)の容量が不足する可能性があるため、未使用領域の拡張を検討する必要があります。
 - Exadata X3-2 (Oracle Linux) の場合、約650GBの未使用領域が存在します。
 - システムバックアップをLVM Snapshotを使用して取得する場合、LVM Snapshot使用時の空き領域を確保しておくことが必要です。



[Doc ID 1357457.1] How to Expand Exadata Compute Node File Systems

Oracle Exadata Database Machine Owner's Guide 11g Release 2 (11.2)
-7 Maintaining Oracle Exadata Database Machine and Oracle Exadata Storage Expansion Rack

Tips9.サービスの分割方針

事例で分かるサービス分割の重要性



プロジェクト
ご担当者様
(インフラ)

Exadata上で複数のデータベースを稼働しています。あるデータベースの特定の処理がリソースを占有してしまい、他のデータベースの処理が遅延しはじめています。どうしたらいいのでしょうか？

リソースを占有している特定の処理について、サービス分割を行ってみたいかがでしょうか？分割したサービス単位でリソース・マネージャーと紐付けてリソース制限の実装が可能です。

サービスはデフォルト・サービスしか使用していません。本当はサービス分割したいのですが、データベースへの接続記述子が複数ファイルに記載されているため、本番稼働している今の時点でサービス分割をするのは厳しいです。



プロジェクト
ご担当者様
(アプリ)



オラクル
コンサル

- ✓ 設計の段階からアプリケーションごとにサービスを分割しておくことが重要です。
- ✓ データベースへの接続記述子は極力一元管理するようアプリケーションを設計して下さい。

Tips9.サービスの分割方針

サービス設計の基本方針(設計フェーズ)

- オンライン処理、バッチ処理のサービス分割
 - オンライン処理などの比較的軽微な処理は、全ノード上で稼働するサービスを作成することを検討します。
 - 特定のテーブルに大量の更新を行うようなバッチ処理は、キャッシュフュージョンが大量に発生しパフォーマンスが劣化する可能性があるため、単一ノード上で稼働するサービスを作成することを検討します。また、障害発生時を考慮し、優先ノード以外に使用可能ノードを指定することを検討します。
- 管理用サービスの作成を検討
 - 管理用のサービス(バックアップ/メンテナンス用)を作成することを検討します。
- サービスを小分けに分割しすぎない。
 - サービスの数があまりにも多いとメンテナンス作業の際に管理しにくくなる傾向があります。
 - OCRに登録される情報量が増加し、フェイルオーバー時に同期する必要のあるデータが増え、フェイルオーバーに時間がかかってしまう可能性があります。

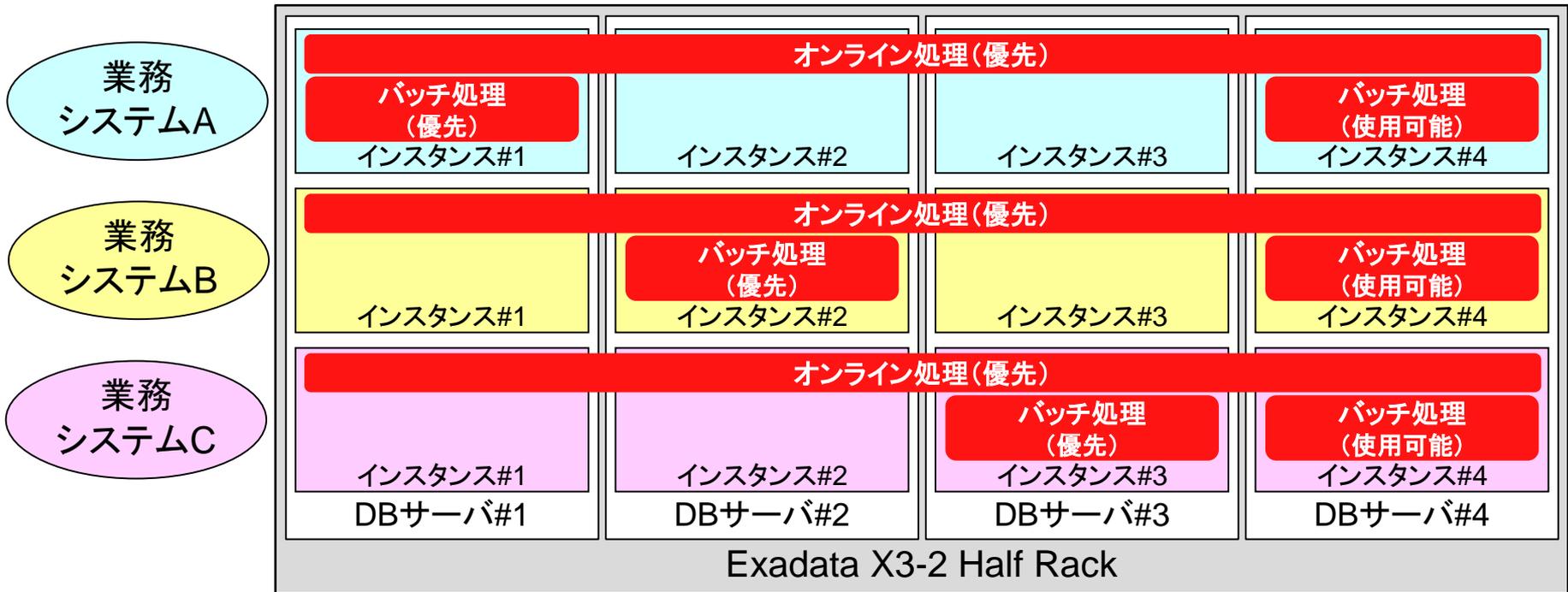
Tips9.サービスの分割方針

サービス設計の基本方針(テスト／運用フェーズ)

- 特定の処理がリソースを占有し、他の処理に影響を及ぼす場合の対処
 - 特定の処理専用のサービスを作成し、そのサービスに対してリソース・マネージャを使用してリソース制御を行うことを検討します。「サービス分割 + リソース・マネージャ」の合わせ技が効果的です。

Tips9.サービスの分割方針

実プロジェクトでのサービス設計例



その他のTips

✓ Tips10.Exachkの有効活用



- ✓ Mixed Workloads
- ✓ Data Warehousing
- ✓ OLTP

Tips10.Exachkの有効活用

Exachkとは？

- Oracle社が推奨するBest Practicesとお客様環境の設定値の乖離を比較するExadataのコンフィグレーション監査ツールです。
- Best Practicesと異なる設定値について、推奨値と乖離していることによる影響、推奨値に変更するためのアクション等を提示します。
- 1回の実行でExadata上に構成されている全てのデータベースのコンフィグレーションをチェックすることが可能です。
- Exachkの詳細は以下のNote、及び、Exachk User Guideをご参照下さい。

Oracle Exadata Assessment Report

System Health Score is 88 out of 100 [\(detail\)](#)

Cluster Summary

Cluster Name	dmC1 cluster
OS/Kernel Version	Linux309-04 OELRHEL 2.5.32-400.21.el6xk
CRS Home - Version	A079app112.0.3grid: 11.2.0.3.0
DB Home - Version - Names	A079appOracleproduct112.0.3dbhome_1: 11.2.0.3.0-2
Exadata Version	11.2.0.3.1
Number of racks	7
Database Servers	2
Storage Servers	3
IB Switches	2
exachk Version	2.2.2_20130617
Collector	exachk_exad011m_wm_072413_0942220
Collection Date	24-Jul-2013 18:08:52

Check for parameter processes

Critical	
Benefit / Impact:	Experience and testing has shown that certain database initialization parameters should be set at specific values. These are the best practice values set at deployment time. By setting these database initialization parameters as recommended, known problems may be avoided and performance maximized. The parameters are common to all database instances. The impact of setting these parameters is minimal. The performance related settings provide guidance to maintain highest stability without sacrificing performance. Changing the default performance settings can be done after careful performance evaluation and clear understanding of the performance impact.
Recommendation	Risk: If the database initialization parameters are not set as recommended, a variety of issues may be encountered, depending upon which initialization parameter is not set as recommended, and the actual set value. Action / Repair: PROCESSES=1024 is the recommendation for this hardware type. Customers should review this per application requirements.
Needs attention on	dm1
Passed on	dc01.wrlf1, dc02.wrlf2

[Doc ID 1070954.1] Oracle Exadata Database Machine exachk or HealthCheck

Tips10.Exachkの有効活用

Exachkの活用方法

■ 構築フェーズでの活用方法

- オラクルコンサルタントがExadata導入に関わった場合、初期構築が完了した後、Exachkを実行するようにしています。
- Exachkの実行結果より、NGだった項目に対して設計の妥当性を確認します。

■ 運用フェーズでの活用方法

- オラクルコンサルタントが運用アセスメントを行う際、アセスメントのインプットの一つとしてExachkの実行結果を確認しています。
- Exachkの実行結果を分類し、優先順位付けした上で適宜実装に反映していただくことを推奨しています。

<例>

- ノードやインスタンスのダウンに繋がる可能性のあるもの
- 性能低下リスク、あるいは性能向上の余地があるもの
- アプリケーションエラーやアプリケーションハングに繋がる可能性のあるもの
- ブロック破損等によるデータ・ロストに繋がる可能性のあるもの
- その他(上記以外)

Hardware and Software

ORACLE®

Engineered to Work Together

ORACLE®