HITACHI
Inspire the Next

HiRDB技術解説

HiRDBチューニング解説

2019/11

株式会社 日立製作所　サービス＆プラットフォームビジネスユニット
サービスプラットフォーム事業本部　DB部

HiRDB

HITACHI
Inspire the Next

Contents

HITACHI
Inspire the Next

1. はじめに

1-1　本資料の概要

HITACHI
Inspire the Next

■ 本資料の概要

安定したシステム性能を確保するためには、テスト段階以降でのチューニングだけではなく、性能を意識した事前の設計や情報取得がカギとなります。

本資料では、各種チューニング方法の解説に加え、システム開発の各作業工程で考慮すべき項目や状況に応じて取得すべきチューニング情報についてなど、総合的にご紹介します。

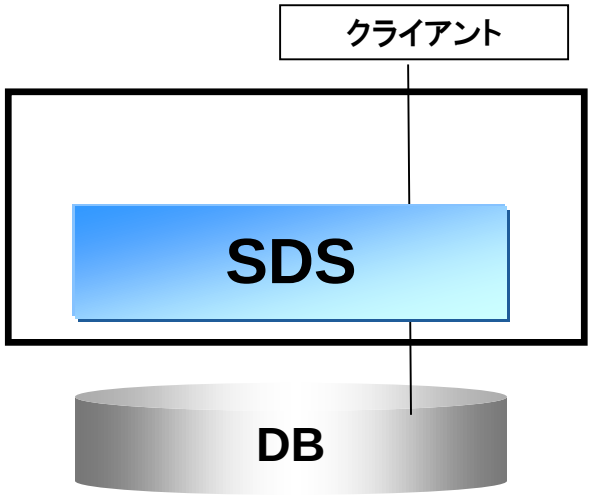
◆ 各工程で行う作業や取得するチューニング情報を解説します。
また、各種チューニング情報の内容や取得方法についても解説します。
2章: 各工程で考慮すべき項目とチューニング手順

◆ 各チューニング対象について前提知識、チューニング方法を解説します。
3章: SQLのチューニング
4章: ユティリティのチューニング
5章: 各種バッファのチューニング

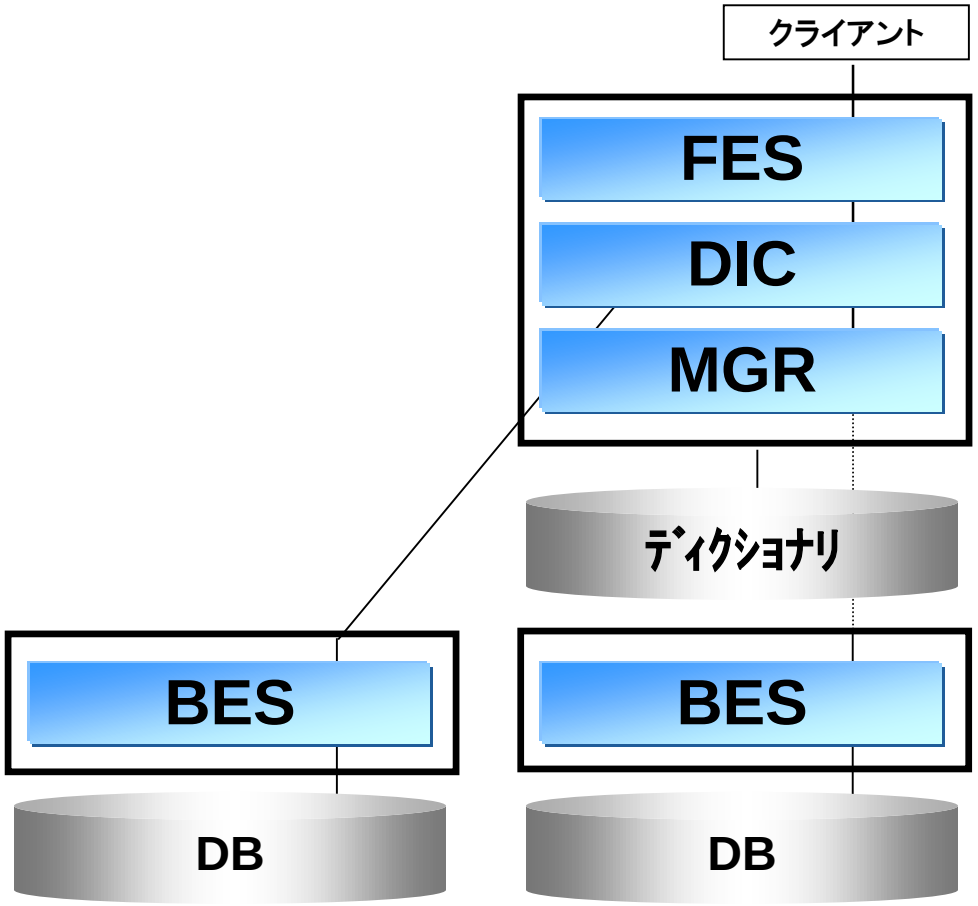
1-2 HiRDBサーバの構成要素の名称

HITACHI
Inspire the Next

■ HiRDB/Single Server



■ HiRDB/Parallel Server



用語

SDS(Single Database Server)
FES(Front End Server)
BES(Back End Server)
DIC(Dictionary Server)
MGR(System Manager)

HITACHI
Inspire the Next

2. 各工程で考慮すべき項目とチューニング手順

HITACHI
Inspire the Next

2. 各工程で考慮すべき項目とチューニング手順

2.1 各プロセスで行う作業

2.2 取得するチューニング情報

2.3 チューニング情報の取得方法

2-1-1　担当者の役割とチューニング対象

HITACHI
Inspire the Next

解説　各担当者の役割とチューニング対象を示します。

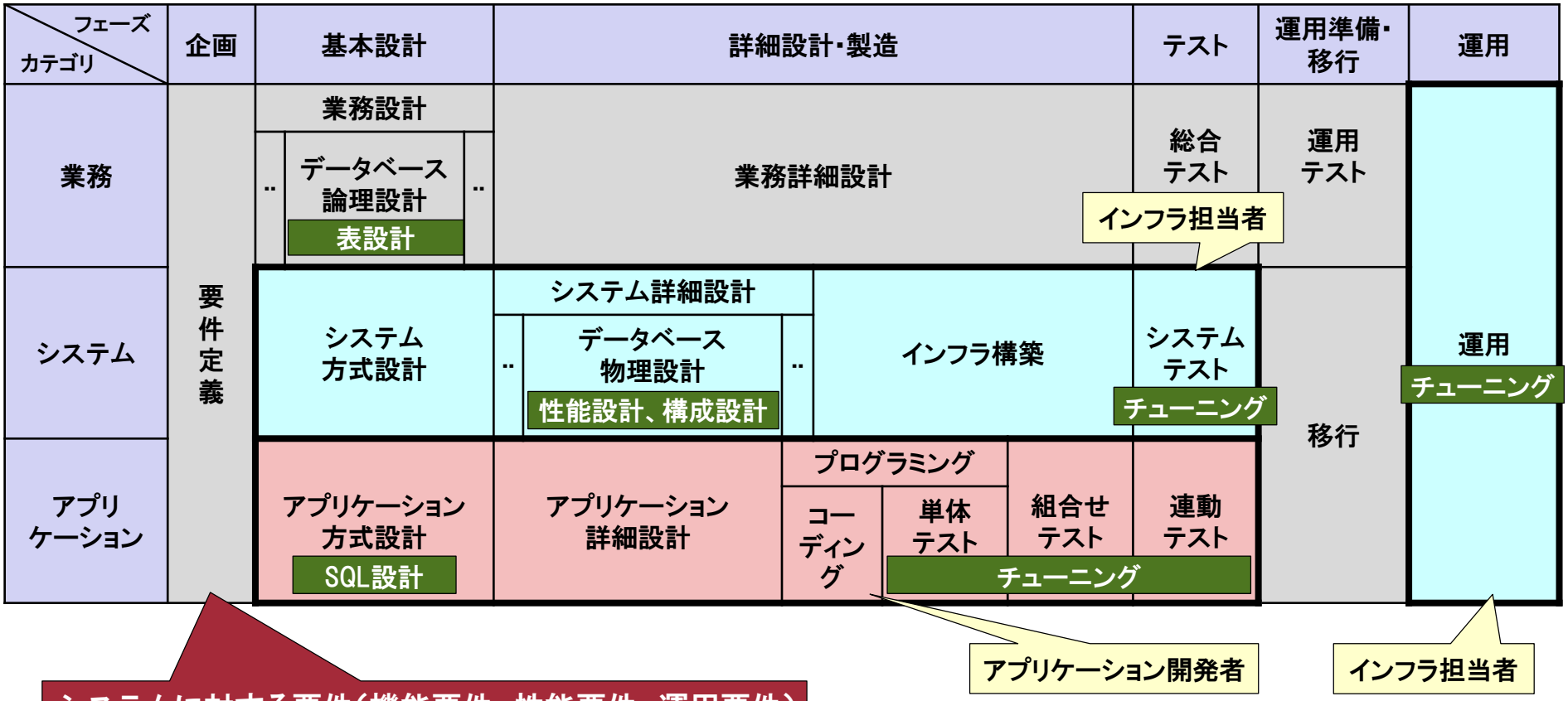
担当	役割	チューニング対象
インフラ担当者	・システムのインフラ環境の設計、構築、テスト ・本番稼働後の運用	バッファ（グローバルバッファ、ディクショナリバッファ）等
アプリケーション開発者(*1)	アプリケーションの設計、開発、テスト	SQL、DB排他制御等

(*1)：以降、アプリ開発者と表記します。

2-1-2 担当者毎のプロセス

HITACHI
Inspire the Next

解説 工程全体での各担当のプロセス(各フェーズでの作業)を以下に示します。また、各プロセスで必要な設計またはチューニングについても示します。



2-1-3 インフラ担当者の工程詳細(１)

HITACHI
Inspire the Next

解説 インフラ担当者の工程の詳細と行うべき作業について解説します。

システム方式設計	システム詳細設計（データベース物理設計）	インフラ構築	システムテスト
システム基盤の方式を検討し、必要なハードウェア・ミドルウェアの構成を見積ります。	アプリケーション方式設計からのフィードバックを取り込み、システム運用、および各システム構成要素の内部設計を実施します。	設計に従ってデータベースを構築します。	非機能要件(*1)に対する設計内容を満たしているか確認します。 (*1):定義される要件のうち、機能面以外のもの全般。性能や信頼性、拡張性、運用性、セキュリティなどに関する要件

性能設計、構成設計

データベースを構成するファイルやバッファの配置や容量を業務ごとに見積もり、ディスクやメモリの構成や必要容量を決定します。

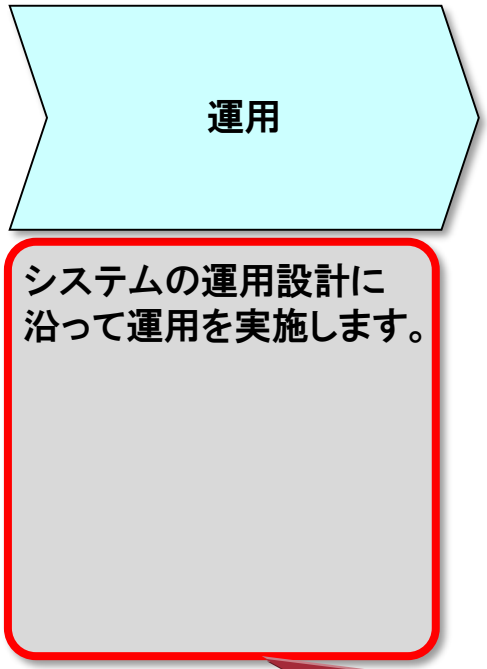
チューニング

詳細なチューニング情報(統計情報、SQLトレース、アクセスパス(*2))を取得・評価し、目標を達成するまでチューニングを繰り返し実施します。

(*2):アクセスパスとは、データベースへのアクセス手順のことです。Oracleの実行計画に相当します。

□ 本資料で解説する設計やチューニングを行うプロセス

2-1-4 インフラ担当者の工程詳細(２)



チューニング

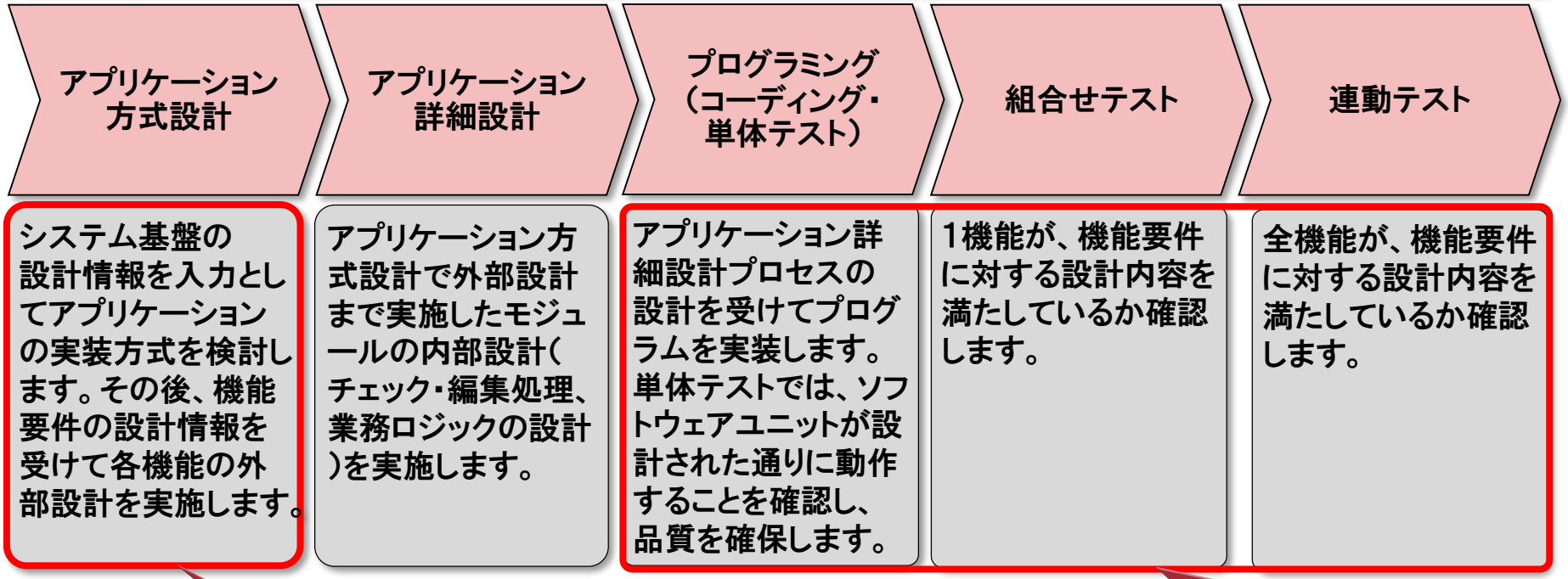
本番運用に入っても、日々の変化などを監視します。チューニングが必要そうな兆候を捉えたら詳細な情報収集を行い、必要に応じてチューニングを実施、安定した稼働を維持します。

本資料で解説する設計やチューニングを行うプロセス

2-1-5 アプリ開発者の工程詳細

HITACHI
Inspire the Next

解説 アプリ開発者の工程の詳細と行うべき作業について解説します。



SQL設計

効率の良いアクセスパスや排他制御を意識したSQL設計を行います。(検索方法、結合方式、排他オプションなど) また、性能を意識したインデクス設計も行います。

チューニング

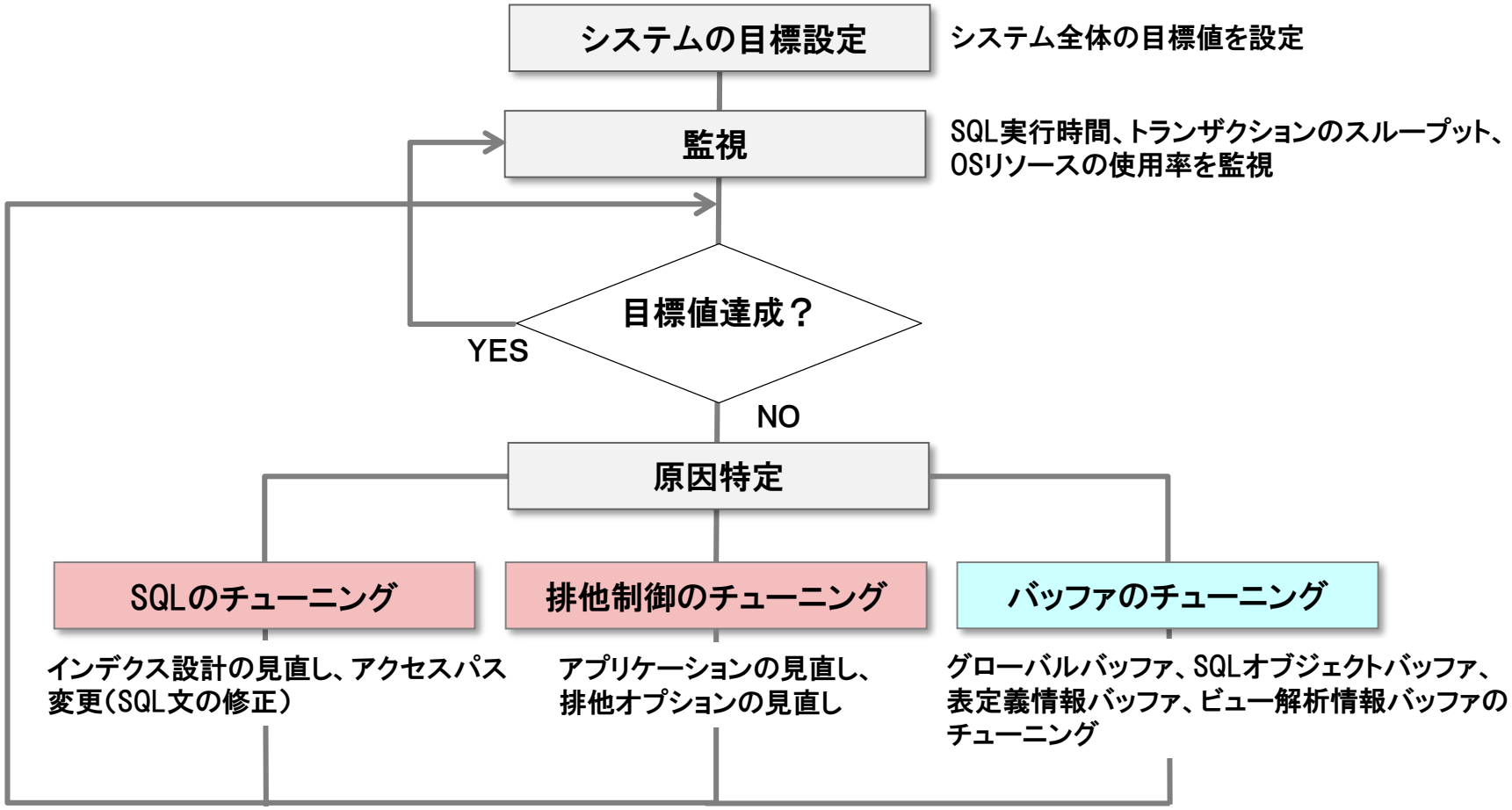
アクセスパス情報を取得し、効率の良いアクセスパスになっているか確認します。なっていなければチューニングを実施し、効率の良いアクセスパスになるまで繰り返し実施します。本番想定のデータでテストが可能なら、UAP統計レポートでSQLの処理時間やデータ処理量の妥当性についても確認します。

□ 本資料で解説する設計やチューニングを行うプロセス

2-1-6 チューニングの手順とポイント

HITACHI
Inspire the Next

解説 目標値を達成するまで、下記のステップを実施していきます。



HITACHI
Inspire the Next

2. 各工程で考慮すべき項目とチューニング手順

2.1 各プロセスで行う作業

2.2 取得するチューニング情報

2.3 チューニング情報の取得方法

2-2-1　取得するチューニング情報～担当別～

HITACHI
Inspire the Next

解説　一覧でまず「概要確認」の情報を取得し、詳細を調査する時に「詳細調査」の情報を取得します。
該当する情報に○をつけており、どちらにも使える情報については、両欄に○があります

担当	プロセス	取得するHiRDBのチューニング情報	取得方法	概要確認	詳細調査
アプリ開発者	単体テスト時	アクセスパス（UAP統計レポートまたはアクセスパス表示ユティリティ）	2-3-1～2-3-3		○
	組合せテスト、連動テスト	SQLトレース、UAP統計レポート	2-3-2		○
インフラ担当者	システムテスト	システムの稼働に関する統計情報	2-3-4～2-3-5	○	○
		グローバルバッファプールに関する統計情報		○	○
		デファードライト処理に関する統計情報		○	○
		データベースの入出力に関する統計情報		○	○
		UAPに関する統計情報			○
		SQLオブジェクト用バッファの統計情報	2-3-8	○	
		グローバルバッファの簡易統計情報	2-3-6	○	
		デッドロック、タイムアウト情報	2-3-10	○	○
		SQLトレース、UAP統計レポート	2-3-2		○
		アクセスパス（アクセスパス表示ユティリティ）	2-3-3		○
		サーバの排他制御の状態表示	2-3-9		○
	運用（監視時）	システムの稼働に関する統計情報	2-3-4～2-3-5	○	○
		グローバルバッファプールに関する統計情報		○	○
		デファードライト処理に関する統計情報		○	○
		SQL実行時間警告情報	2-3-7	○	
		グローバルバッファの簡易統計情報	2-3-6	○	
		デッドロック、タイムアウト情報	2-3-10	○	○
	運用（問題発生時）	UAPに関する統計情報	2-3-4～2-3-5		○
		SQLオブジェクト用バッファの統計情報	2-3-8	○	
		サーバの排他制御の状態表示	2-3-9		○

HiRDB Version 9 09-50サポート
詳細は、4.1節を参照

2-2-2 取得するチューニング情報～目的別～

HITACHI
Inspire the Next

＜凡例＞ ー:HiRDB以外の情報

使用目的	確認項目	取得するHiRDBのチューニング情報	概要確認	詳細調査
監視	CPU使用率、ディスクビジー率等	ー	○	○
	アプリケーションの実行時間情報	ー	○	
	トランザクションの実行回数(アプリケーションのスループット)	システムの稼働に関する統計情報	○	○
	SQL実行時間	SQL実行時間警告情報	○	
対象SQLの特定	SQL実行時間(秒単位)、実行回数、READ/WRITE回数	SQLオブジェクト用バッファの統計情報	○	
SQLのチューニング	アクセスパス	UAP統計レポート、アクセスパス表示ユティリティのアクセスパス情報		○
	SQL実行時間(ミリ秒単位)、サーバ側でのSQL実行時間(マイクロ秒単位)、アクセスパス	UAP統計レポート		○
	SQL実行時間	SQLトレース		○
排他制御のチューニング	デッドロック回数、排他待ちの発生の有無	システムの稼働に関する統計情報	○	○
	デッドロック発生時の排他情報(排他待ち時間、排他資源)	デッドロック・タイムアウト情報	○	○
	コマンド投入時の排他情報(排他資源、排他待ち時間、プロセスID)	サーバの排他制御の状態表示		○
グローバルバッファのチューニング	グローバルバッファヒット率(一定間隔で取得)	グローバルバッファの簡易統計情報	○	
	グローバルバッファ情報(ヒット率、フラッシュ回数、RDエリアへのI/O数　等)	グローバルバッファプールに関する統計情報	○	○
表定義情報用バッファのチューニング	表定義情報用バッファ情報(ヒット率)	システムの稼働に関する統計情報	○	○
ビュー解析情報用バッファのチューニング	ビュー解析情報用バッファ情報(ヒット率)	システムの稼働に関する統計情報	○	○
SQLオブジェクトバッファのチューニング	SQLオブジェクトバッファ情報(ヒット率)	システムの稼働に関する統計情報	○	○

HITACHI
Inspire the Next

2. 各工程で考慮すべき項目とチューニング手順

2.1 各プロセスで行う作業

2.2 取得するチューニング情報

2.3 チューニング情報の取得方法

2-3-1　アクセスパスの取得

HITACHI
Inspire the Next

解説

アクセスパスとは、データベースへのアクセス手順のことです。 アクセスパスを分析することで、パフォーマンス劣化の要因となるSQL文に対し、SQLチューニングの必要性があるかどうかを判断することができます。

アクセスパスを取得する方法は、2通りあります。ここではそれぞれの方法の特徴を示します。

	UAP統計レポート	アクセスパス情報ファイル
方法	SQLトレースなどと一緒にテキストで出力する（詳細は、**2-3-2**参照）	アクセスパス情報ファイルを生成し、ユティリティを使い編集する（詳細は、**2-3-3**参照）
特徴	・HiRDBクライアント(UAP)側に出力される ・テキスト形式で出力されるため、編集せずに参照できる ・アクセスパス情報ファイルに比べて、出力される情報の種類が多い ・UAPの動作と合わせて検証できる	・すべてのUAPの情報がHiRDBサーバの運用ディレクトリ下に出力されるため、出力ファイルを管理しやすい ・バイナリ形式で出力されるため、参照するための編集作業が必要 ・重複SQLの情報抑止が可能

基本的には、UAP統計レポートを利用してください。
複数のUAPの情報を一括して分析したい場合などは、アクセスパス情報ファイルの利用が有効です。

2-3-2 UAP統計レポートの取得

HITACHI
Inspire the Next

解説 UAP統計レポートの取得方法について解説します。

◆概要

UAP実行時のSQLに関する様々な情報を提供します。SQLトレース、UAP実行に関する統計情報、SQL実行時のデータ処理件数、アクセスパス情報です。

◆取得方法

UAP統計レポートは、次のクライアント環境定義に値を設定することで取得できます。出力先がカレントディレクトリでよい場合には、項番2、3のみの指定で取得できます。

#	クライアント環境定義	環境変数の内容
1	PDCLTPATH	情報出力先。省略時は、カレントディレクトリが仮定されます。
2	PDSQLTRACE	トレースのファイルサイズ(byte)を指定。0を指定した場合は、ファイルの最大のサイズとなります。省略をした場合は、情報を出力しません。
3	PDUAPREPLVL	UAP統計レポートの出力情報を指定します。出力には、アクセスパス情報、SQL単位の情報、UAP単位の情報、SQL実行時の中間結果情報があります。 アクセスパスの解析時は、aを指定し全ての情報を出力することをお勧めします。 バージョン09-50の変更点 UAP統計レポートのUAP単位の情報を、コネクション単位の出力に加え、トランザクション単位でも出力できるようになり、詳細に解析できるようになりました。トランザクション単位の出力を使うと、OLTP環境下や、コネクションプーリングなどコネクションを変更するのが困難な環境でも、容易に情報出力できます。 指定値の詳細は付録B-6を参照してください。
4	PDREPPATH	複数の結果が混ざらないように、PDCLTPATHで指定したディレクトリとは別の場所にファイルを分けて収集したい場合に指定します。

注意事項

アクセスパス情報を取得するとSQLオブジェクトバッファがヒットしなくなり、毎SQLについて解析処理を実行します。よって、多少パフォーマンス劣化します。

2-3-3 アクセスパス表示ユティリティによる取得

HITACHI
Inspire the Next

解説 アクセスパス情報ファイルを生成し、アクセスパス表示ユティリティ(pdvwopt)で表示(*1)する方法について解説します。

■手順の流れ

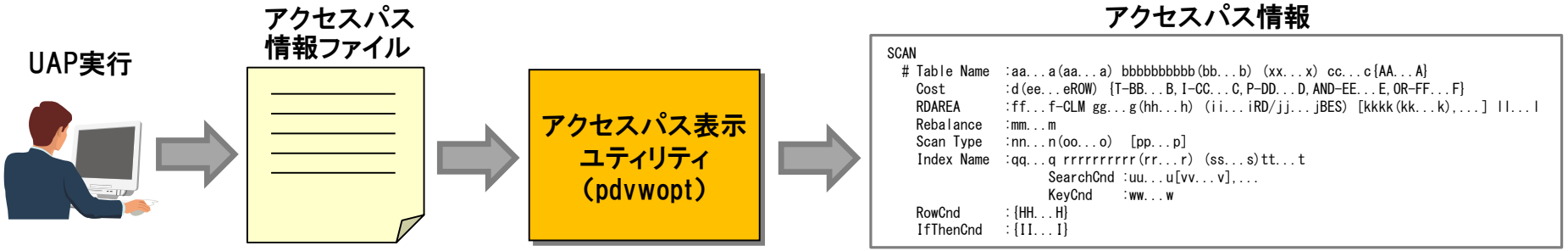
1. アクセスパス情報ファイルを取得するためのクライアント環境定義の設定
   クライアント環境定義PDVWOPTMODEに1以上を設定(*2)
2. UAP実行

→ クライアント側

3. アクセスパス表示ユティリティ(pdvwopt)でアクセスパスの情報を表示

→ サーバ側

(*2):1の場合、SQLオブジェクトがバッファ中にあるSQLについては、情報を出力しません。
2の場合、SQLオブジェクトがバッファ中にあるSQLについてもSQLオブジェクトを再作成し、情報を出力します。



(*1):HiRDBの運用支援製品「HiRDB SQL Tuning Advisor」でもアクセスパス情報を表示することができます。
詳細は、HiRDB技術資料「GUIによるHiRDBシステム開発方法の解説と演習」を参照してください。

HiRDB技術資料のURL

https://www.hitachi.co.jp/Prod/comp/soft1/hirdb/files/tech_info/index.html

2-3-4 統計情報の取得

HITACHI
Inspire the Next

解説 統計情報の取得方法について解説します。

◆概要

HiRDBでは、稼動中の使用リソースに関する様々な情報を統計情報として収集できます。トランザクション量、メモリ、通信(パラレル)、排他等の使用状況を確認しながら、チューニングを行えます。

通常は、システムの稼動に関する統計情報(sys)、グローバルバッファプールに関する統計情報(buf)、デファードライト処理に関する統計情報(dfw)、データベースの入出力に関する統計情報(dio)を取得します。
パフォーマンス低下の兆候が見られた場合に、UAPに関する統計情報(uap)を取得して、
個々のアプリケーションの動作を確認して、どの処理に原因があるのか調査します。

バージョン09-50の変更点

データベースの入出力に関する統計情報をサポートしました。トランザクション遅延などが発生した際に、データベースへの入出力で遅延が発生していないか確認するために使用します。

◆取得方法 ⇒ 詳細は、2-3-5で解説します。

◆編集・解析

テキスト形式とDAT形式(csvファイル)の2形式で出力できます。
時系列的な解析を行う場合、csvファイルで出力し、EXCEL等で編集すると解析し易いです。

◆その他

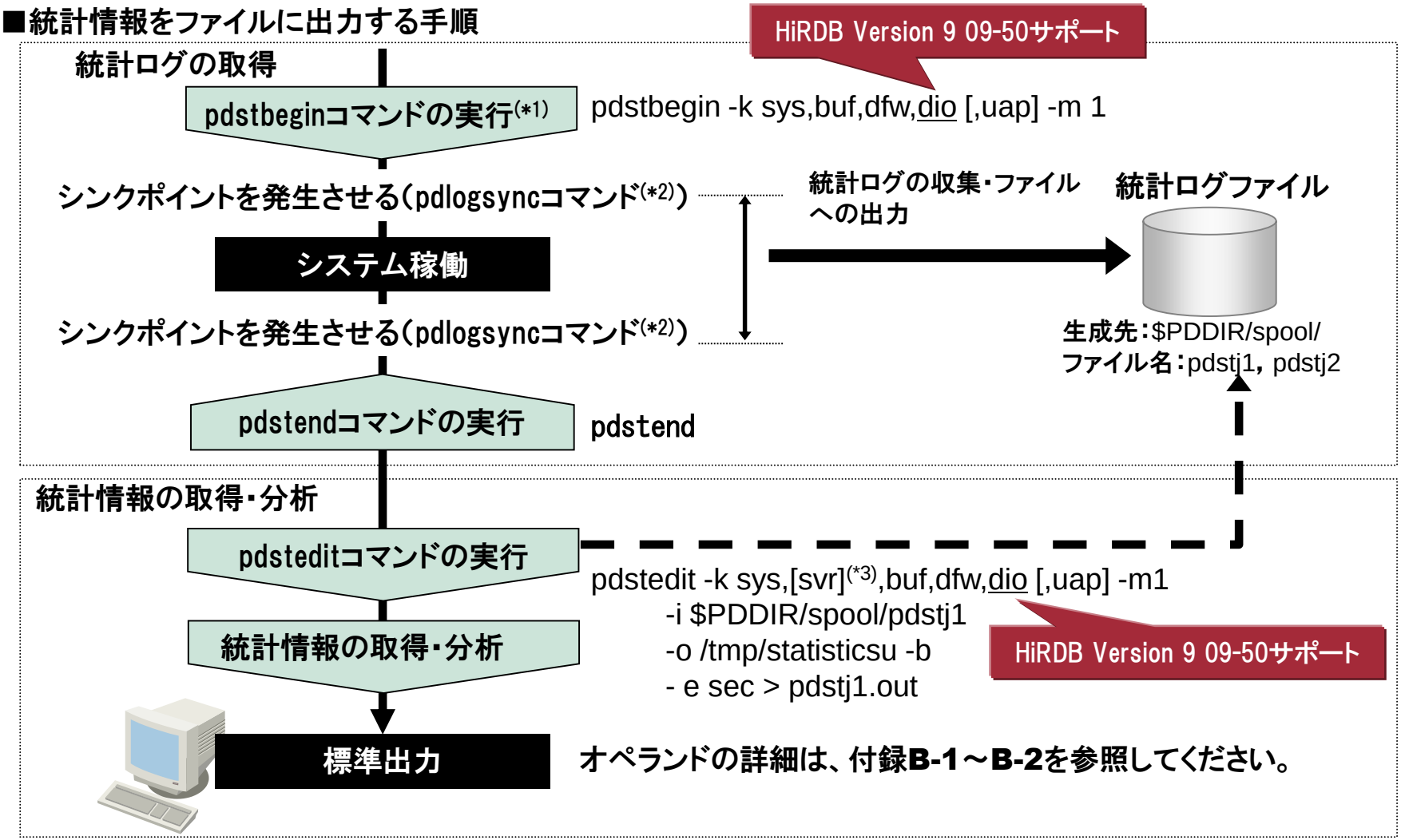
統計ログファイル(pdstj1およびpdstj2)のサイズ(pd_stj_file_size)は、デフォルトの1Mバイトでは少ない場合があるため、10Mバイト以上にすることを推奨します。正確に見積もりたい場合は、HiRDBマニュアル「システム定義」-「統計ログファイル(pd_stj_file_size)の見積もり式」を参照してください。

バージョン09-50の変更点

HiRDB Version 9 09-50以降の推奨モードでは、省略値でHiRDBの開始時から「システムの稼働に関する統計情報(出力間隔1分)」、「グローバルバッファに関する統計情報」、「デファードライト処理に関する統計情報」、「データベースの入出力に関する統計情報(出力間隔1分)」を取得するようになりました。

2-3-5 統計情報の取得方法

■統計情報をファイルに出力する手順



(*1):システム共通定義(pdsys)にpdstbeginオペランドを指定しておくと、pdstbeginコマンドの投入が不要になります。
(*2):グローバルバッファ(buf)、デファードライト(dfw)の情報を取得する場合に必要です。
(*3):パラレルサーバの場合は、svrを指定するとサーバごとのシステムの稼働に関する統計情報を取得できます。

2-3-6 グローバルバッファの簡易統計情報の取得

HITACHI
Inspire the Next

解説 グローバルバッファ情報の取得方法について解説します。

◆目的

グローバルバッファの使用状況を監視します。
取得した情報(バッファヒット率、バッファフラッシュ回数等)をみて、グローバルバッファのチューニングを行います。短期間内の統計情報を簡易的に取得できるため、まずはこれで評価をし、問題が解決しない場合は、ファイルに出力する統計情報を取得して下さい。

◆取得方法

1. シェルでpdbuflsを一定時間間隔で実行するようプログラムします。
   結果は、直前に実行したpdbufls投入時間から、今回の投入時間までの累積となります。
   DAT形式での出力が可能です。コマンドの詳細は、付録**B-3**を参照してください。
2. シェルを実行し、結果をファイルに格納します。

HiRDBの運用支援製品「HiRDB Control Manager」でもグローバルバッファ情報を表示することができます。
詳細は、HiRDB技術資料「GUIによるHiRDBシステム開発方法の解説と演習」を参照してください。

◆利点

1. 任意の一定間隔で情報を取得できる。
2. 事前準備が不要で情報を取得できる。

HiRDB技術資料のURL

https://www.hitachi.co.jp/Prod/comp/soft1/hirdb/files/tech_info/index.html

2-3-7 SQL実行時間警告情報

HITACHI
Inspire the Next

解説 SQL実行時間警告情報の取得方法について解説します。SQLの実行時間が設定した警告時間以上であった場合、そのSQLに対して警告情報を出力します。

◆目的

SQL実行時間警告出力機能は次に示す目的などに使用します。

・SQL応答待ち時間が一定時間以上のSQLに関する情報を取得してチューニングの資料にする。

・データ量の増加などでHiRDBのサーバプロセスからの応答時間が長くなるUAPについて、PDCWAITTIMEオーバが発生する可能性があることを事前に検知する。

◆取得方法

システム定義に、以下を指定します。

・pd_cwaittime_wrn_pnt = PDCWAITTIME値に占める割合あるいは時間(*1)

・pd_cwaittime_report_dir = SQL実行時間警告情報ファイルの出力先ディレクトリ

UAP実行時に、pd_cwaittime_wrn_pntで設定した割合、あるいは時間に達したSQL文が、pd_cwaittime_report_dir下のpdcwwrn1またはpdcwwrn2に出力されます。

(*1):クライアント環境定義のPDCWAITTIMEWRNPNTオペランドでも設定できます。

バージョン09-50の変更点

省略値でSQL実行時間警告情報を出力するようになりました。

2-3-8 SQLオブジェクト用バッファの統計情報の取得

HITACHI
Inspire the Next

解説 SQLオブジェクト用バッファの統計情報の取得方法について解説します。

チューニング情報を収集していない実稼動中などに、簡易的※な確認を行います。
SQLの実行時間、実行回数、ディスクI/O回数などを確認し、処理時間の掛かるSQLや入出力の多いSQLを特定します。

※簡易的→SQLオブジェクトにキャッシュされているSQLの情報のみ取得可。

V09-04よりDAT形式での出力が可能になり、解析しやすくなりました。

バージョン09-50の変更点

SQLオブジェクト用バッファの統計情報に出力する情報を拡充(実行中のSQLのSQLオブジェクト情報と、そのSQLを実行しているUAPの情報など)しました。詳細は、付録**B-4**を参照してください。

SQLオブジェクト用バッファの統計情報を出力するためには、pdobilsコマンドを用います。
pdobilsコマンドの詳細は、付録**B-4**を参照してください。

■手順の流れ

1. 統計情報のクリア
   pdobils -r

2. アプリケーションの実行

3. 統計情報を表示
   pdobils > pdobils.out

2-3-9 サーバの排他制御の状態取得

HITACHI
Inspire the Next

解説 サーバの排他制御の状態の取得方法について解説します。

◆目的

コマンド投入時の排他状況を表示します。
取得した情報(排他待ち時間、排他資源、プロセスID)をみて、排他待ちの原因を特定します。

◆取得方法

1. pdls -d lckコマンドを投入します。排他を待たせている側のプロセスを特定する場合は、-aオプションを同時に指定します。 なお、-aオプションを指定すると出力量が多くなるのでご注意ください。
2. プロセスIDより、pdls -d prcでUAP名称を参照し、UAPを特定します。
3. 資源種別、資源名称より、排他資源をディクショナリ表を検索することにより特定します。

2-3-10 デッドロック情報、排他待ち情報

HITACHI
Inspire the Next

解説 デッドロック情報、排他待ち情報の取得方法について解説します。

◆目的

デッドロック発生時、排他待ち時間限界経過時間以上の排他待ちが発生した場合、排他情報(排他待ち時間、排他資源)を取得します。

デッドロック、排他待ちの原因を特定します。

◆取得方法

1. システム定義に、以下を指定します。
   ・pd_lck_deadlock_info = Y （デッドロック、タイムアウト情報取得）
   ・pd_lck_wait_timeout = 排他待ち限界経過時間
   （排他待ち時間限界経過時間を超えて排他待ちが発生したSQLはエラーリターンします）
2. 資源種別、資源名称より、排他資源をディクショナリ表を検索することにより特定します。

◆実際によく使用されている方法

通常、pd_lck_deadlock_info = Y (デフォルト)として運用する場合が多いです。
pd_lck_wait_timeoutは業務内容に応じて指定します(デフォルトは180秒)。

◆その他

pdls -d lckと同様に、資源種別、資源名称より、排他資源をディクショナリ表を検索することにより特定できます。

HITACHI
Inspire the Next

3. SQLのチューニング

HITACHI
Inspire the Next

3. SQLのチューニング

3.1 評価ポイント

3.2 チューニング

3.3 確認するチューニング情報

3.4 チューニング情報の見方

3-1-1　はじめに

本章では、SQLチューニングの内、特に表の検索方法（インデクス）、結合方法について解説します。
「3．2　チューニング」では、前提知識の説明→関連するチューニングの順で説明します。

アクセスパス情報の出力結果（UAP統計レポートまたはアクセスパス表示ユティリティの出力結果）と照らし合せながら解説します。

アクセスパス以外のノウハウについては、HiRDB技術資料「SQLコーディングガイドライン」を参照してください。

HiRDB技術資料のURL

https://www.hitachi.co.jp/Prod/comp/soft1/hirdb/files/tech_info/index.html

3-1-2　評価ポイント

HITACHI
Inspire the Next

解説

アクセスパス情報(アクセスパス表示ユティリティ(pdvwopt)またはUAP統計レポート)の出力結果を基に以下の評価ポイントを検証します。
確認した結果、想定通りになっていない場合はチューニングを行います。

#	評価ポイント	アクセスパス表示
1	意図したスキャンタイプになっているか	Scan Type
2	意図したインデクスを使用しているか	Index Name
3	サーチ条件(アクセスパス表示は)の範囲が絞り込まれているか	SearchCnd
4	結合方式は意図した通りか	Join Type
5	表の結合順序は意図した通りか	JOIN　LおよびR順序
6	ネストループジョイン時の転送方法がBROADCAST転送 または KEY RANGE PARTIAL BROAD CAST転送になっていないか	Transfer Type

HITACHI
Inspire the Next

3. SQLのチューニング

3.1 評価ポイント

3.2 チューニング

3.3 確認するチューニング情報

3.4 チューニング情報の見方

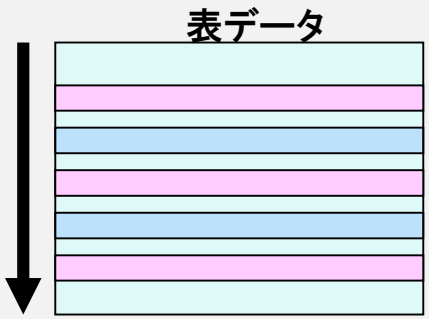
3-2-1　検索方式(１)

HITACHI
Inspire the Next

テーブルスキャン

検索条件の内容にかかわらず、検索対象表の全行をシーケンシャルにアクセスする方法です。

条件によって検索結果を絞り込める場合でも、すべてのデータページを参照するため、データ量が多いと性能は悪くなります。



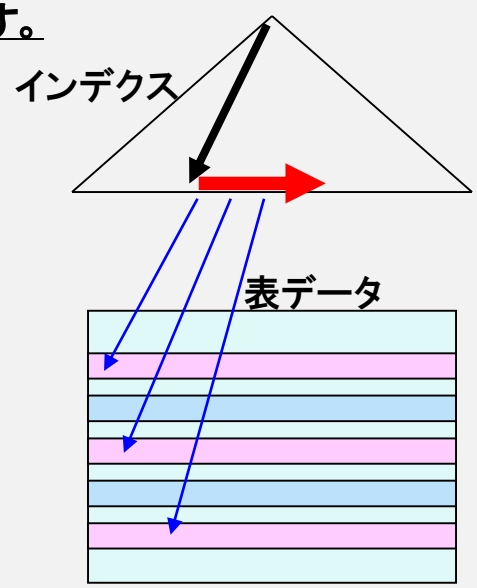
Scan Type:

TABLE SCAN

インデクススキャン

インデクスを参照して条件に該当するデータを絞り込んでから、テーブルのデータをアクセスする方法です。

インデクスであまり絞り込めない場合は、データページに対するランダムな入出力が増え、性能が悪くなります。



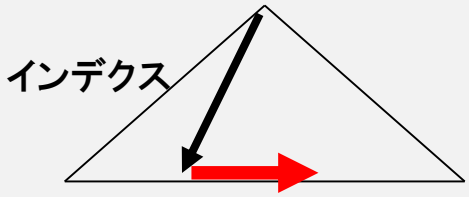
Scan Type:

INDEX SCAN
MULTI COLUMNS INDEX SCAN
PLUGIN INDEX SCAN

キースキャン

インデクスを参照してインデクス中のデータ(インデクス構成列の値または行識別子)にアクセスする方式です。

インデクスであまり絞り込めない場合でも、データページの入出力がなく、インデクスページを参照するだけなので、高速に検索できます。



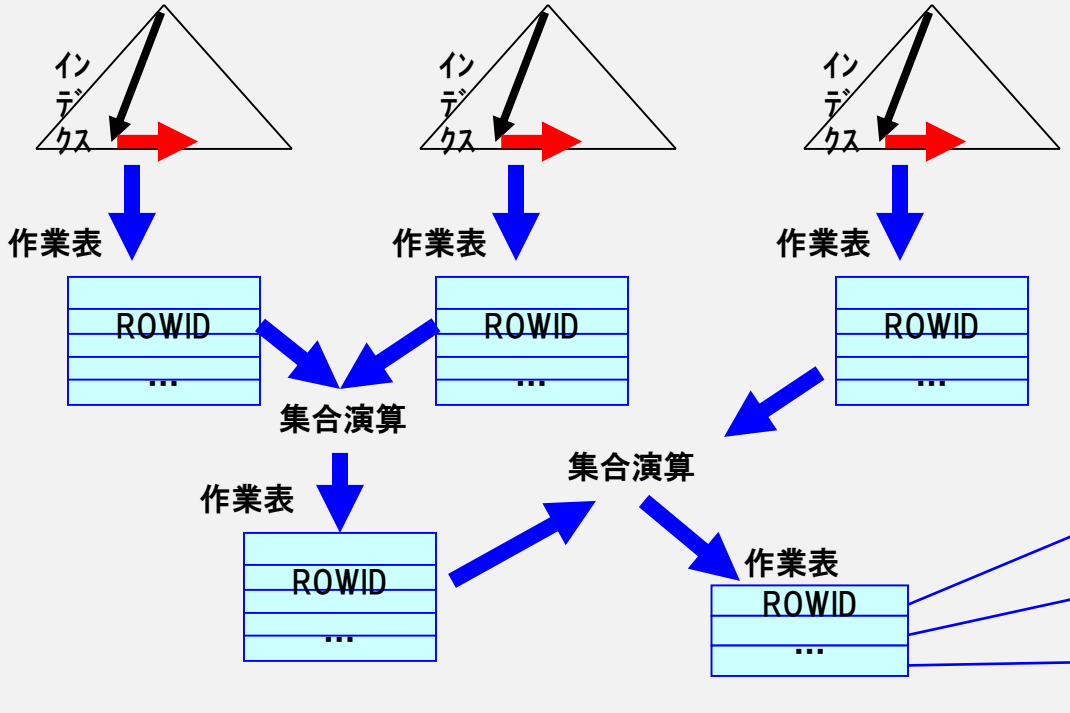
Scan Type:

KEY SCAN
MULTI COLUMNS KEY SCAN
PLUGIN KEY SCAN

HITACHI
Inspire the Next

複数のインデクスを利用した検索(その1:AND/ORどちらでも)

各条件に対して、インデクスを利用し条件を満たす行の集合を求め、行識別子(ROWID)の作業表を作成します。それらの集合間の集合演算を行い、AND/OR条件を満たす集合を求め、その集合の行識別子を使用し、表中の行を取り出します。



論理演算	集合演算
AND	積集合
OR	和集合

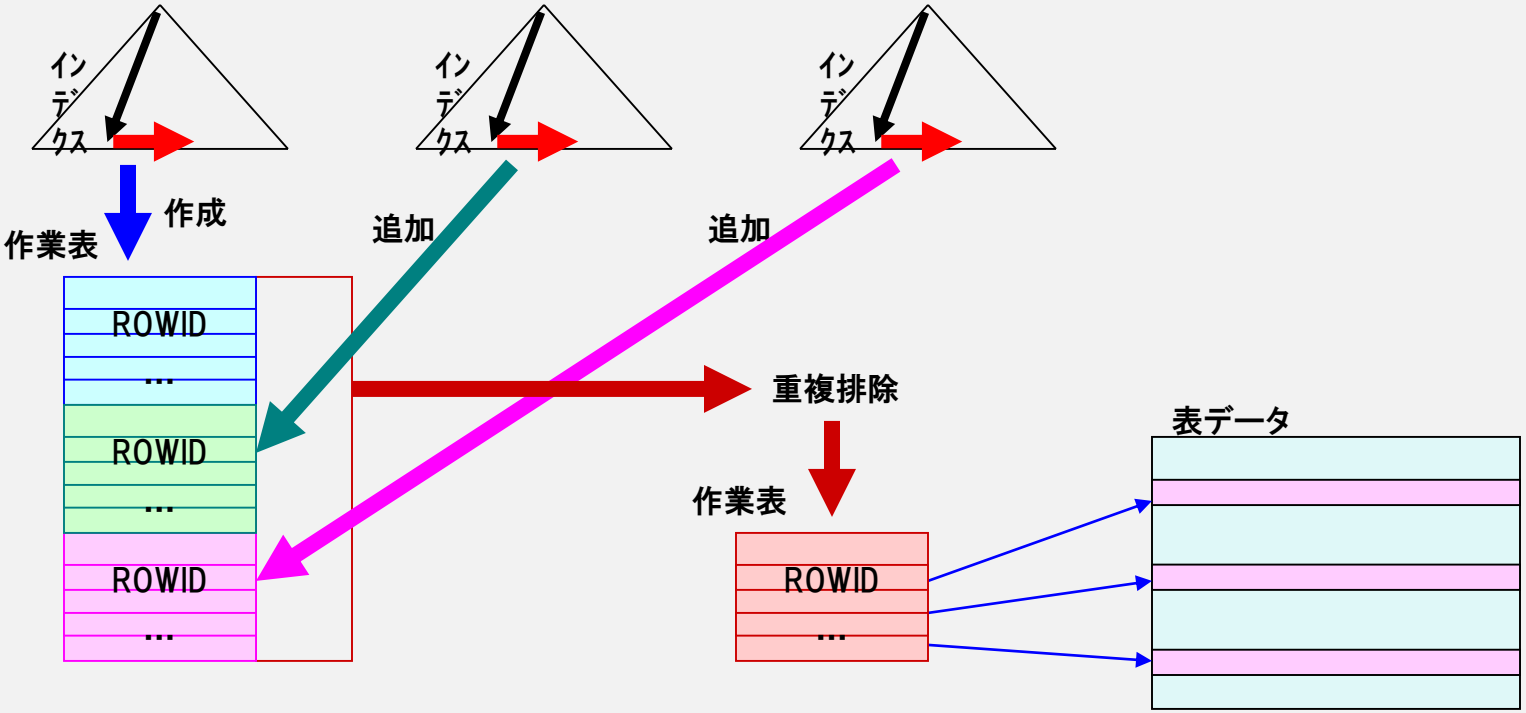
Scan Type:

AND PLURAL INDEXES SCAN

HITACHI
Inspire the Next

複数のインデクスを利用した検索(その2:OR条件のみ)

OR演算のすべてのオペランドの各条件に対して、インデクスの利用が可能な場合、一つの条件を満たす行の行識別子(ROWID)の作業表を作成し、残りの各条件を満たす行識別子を追加後に、行識別子を重複排除し、行識別子を使用し、表中の行を取り出します。



Scan Type:

OR PLURAL INDEXES SCAN

3-2-4 検索方法のチューニング(１)

HITACHI
Inspire the Next

テーブルスキャンの場合

Scan Type: TABLE SCAN

CHECK
絞り込める条件を指定している検索において、テーブルスキャンになっていないか？

ACTION
絞り込める条件の列にインデクスを定義(インデクスを定義することによって、表のデータのアクセス量を削減し、検索性能を改善できる)する。
(注)表の行数が少なく、現時点で、性能が悪くなくても、将来、行数が増加する場合や、本番環境では、行数が多いという場合にも、テーブルスキャンならば、表の行数に依存した性能になるが、インデクスを利用すると性能が安定する。

インデクススキャンの場合

Scan Type:
[MULTI COLUMNS] INDEX SCAN

CHECK
SQL中で参照する列数が少なく、さらに、高速化する必要があるか？

ACTION
インデクスの構成列の後方に、その表に関しSQL中で参照するすべての列を含めたインデクスに変更することによって、キースキャンにすることが可能。(インデクススキャンがキースキャンになると、インデクスのみの参照になり、表のデータページのアクセスを削減できる。)ただし、インデクスの構成列が増えると、インデクスの容量が増加するので、注意が必要。
また、インデクスを変更すると、他のSQLにも影響する可能性があるので、他のSQLへの配慮も必要。

3-2-5 検索方法のチューニング(２)

HITACHI
Inspire the Next

ANDまたはOR条件での複数インデクス利用の場合

Scan Type:
AND PLURAL INDEXES SCAN

CHECK
複数の=条件のAND演算ならば、複数列インデクスを定義できないか？

ACTION
絞り込める条件の列にインデクスを定義する。
複数のインデクスを利用するよりも、一つのインデクスで複数の条件による絞込みを行う方が、効率が良い。
ただし、検索パターンに制限のないような非定型業務の場合には、インデクスの数が多くならないように、実行頻度が多いSQLについて、複数列インデクスの定義を検討するのが良い。

CHECK
利用されるインデクスでの絞込みが、極端によくないものが含まれていないか？

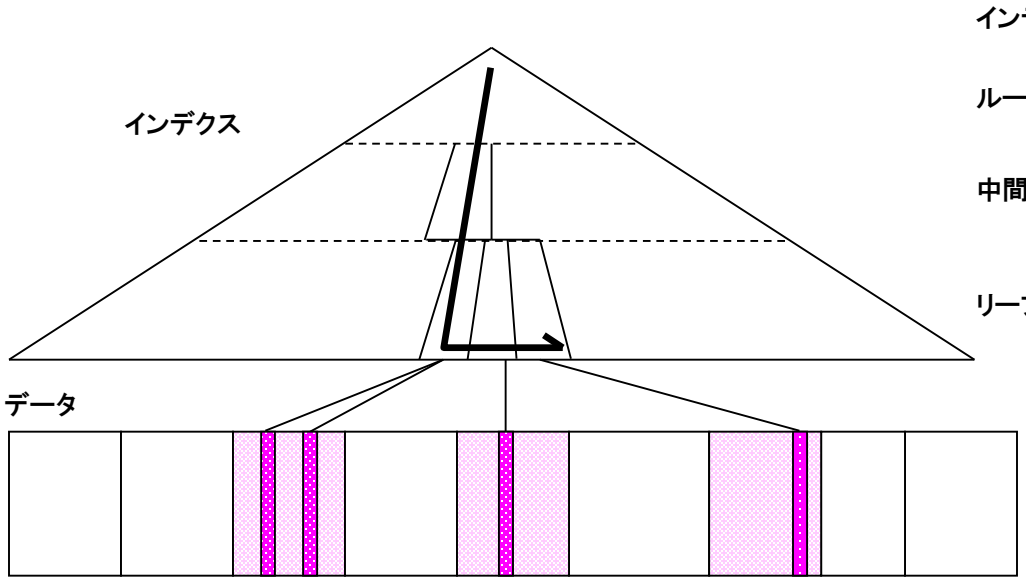
ACTION
AND条件での複数インデクス利用において、一つでも、重複が多く、あまり絞り込めないインデクスが含まれていると、そのインデクスから作成される行識別子の集合が多くなり、その集合の作成、および、その集合からの集合演算のための負荷が大きくなり、性能が出ない場合がある。
→重複の多い列には、インデクスを定義しない、若しくは、他のインデクスの構成列に含める。
（絞り込めないインデクスが利用されないようにする。）

CHECK
日付、時間など複数の列の組合せで大小比較をOR演算で行っていないか？

ACTION
可能ならば、演算を含まないように、行値構成子を使ったSQLに変更する。
（演算を含む条件に対しては、インデクスを有効に利用できない）

例：　C1>0 OR (C1=0 AND C2>0) → (C1,C2)>(0,0)

3-2-6 インデクスの効果



インデクスの構造

ルートページ
中間ページ
上位ページ

リーフページ　最下位ページ

リーフページは
シーケンシャル
アクセス

インデクス経由でのデータは
ランダムアクセス

●アクセスデータの絞込みによるI/O回数およびCPU時間の削減

絞り込める条件列に対してインデクスの定義を検討してください。

●キー順[昇順または降順]の行データ取得によるソート処理削除

ソートに用いる列に対してインデクスの定義を検討してください。

3-2-7 インデクス利用時の注意点

HITACHI
Inspire the Next

●大量データのランダム参照によるI/Oの増加

インデクスを利用して大量データをアクセスすると、ランダムにデータが参照され、その表の全データページ数を大きく超えるI/Oが発生することがあります。
ただし、データページ用のバッファが十分に大きい場合を除きます。

→ 全件または大量検索の場合は、インデクスを使用しないようにします。

●更新列のインデクスメンテナンスによる更新オーバヘッドの増加

→ インデクス定義時、更新の多い列にはインデクスを定義しないことを検討してください。

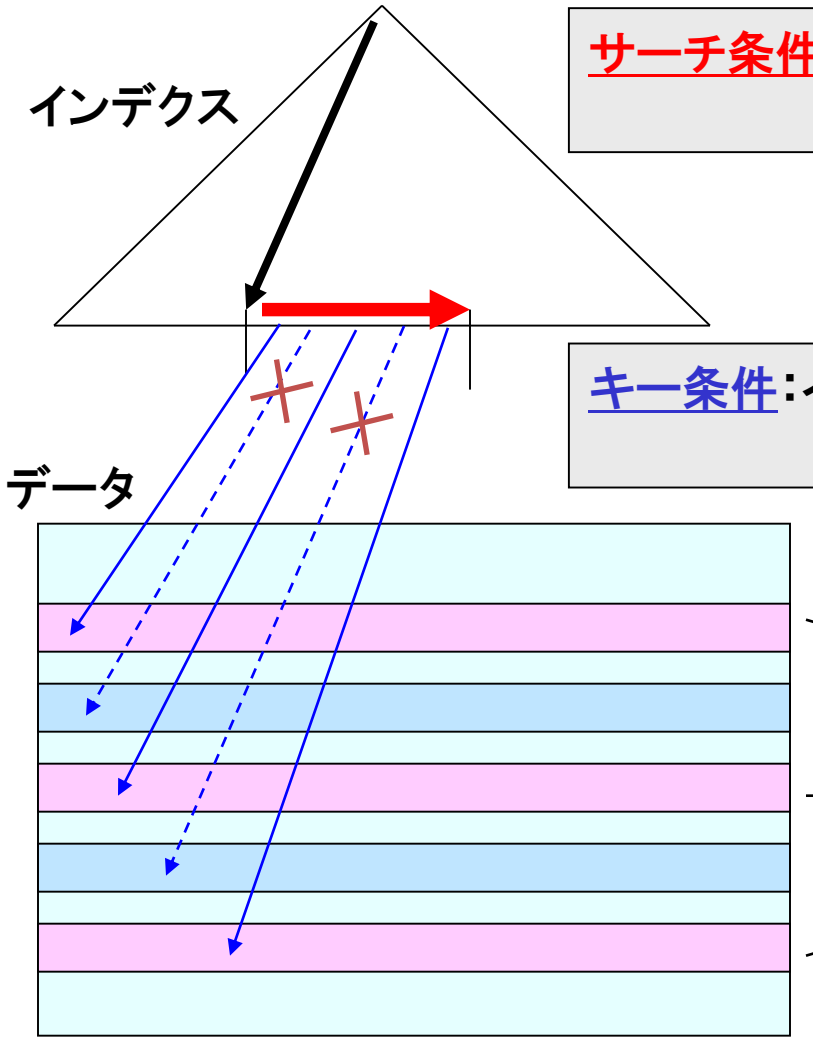
●重複の多いキー値は、インデクスメンテナンスオーバヘッド大

→ 重複の多いインデクスを定義しないことを検討してください。
ただし、重複キー値がナル値の場合は、インデクス定義でナル値の除外を指定してください。

3-2-8 表検索時の条件の分類

HITACHI
Inspire the Next

解説 SQLの探索条件は、すべてサーチ条件にて評価できるのが望ましいです。
そして、それは、SQLの記述により変わります。



サーチ条件：インデクスをサーチするための条件で、
インデクスのサーチ範囲が決定

サーチ条件なしでのインデクス利用：
インデクスリーフページのフルスキャン

キー条件：インデクス構成列のキー値で評価する条件で、
データページ中のアクセス行を削減

サーチ条件＋キー条件：
データページのアクセス行が決定

行の取出し

その他の条件：データページ中の
行データを参照して、条件を評価

3-2-9 サーチ条件の種類 単一列インデクス(KEY)

HITACHI
Inspire the Next

SearchCnd:	インデクス サーチ範囲	探索条件 (昇順スキャンの場合)		(降順スキャンの場合)	
AT[x]	x ●	KEY=x			
RANGE(CS-CE) [M‥, x]	x ——●	[MIN, x]	KEY<=x	[MAX, x]	KEY>=x
RANGE(CS-OE) [M‥, x]	——○	[MIN, x]	KEY<x	[MAX, x]	KEY>x
RANGE(CS-CE) [x, M‥]	●——	[x, MAX]	KEY>=x	[x, MIN]	KEY<=x
RANGE(OS-CE) [x, M‥]	○——	[x, MAX]	KEY>x	[x, MIN]	KEY<x
RANGE(CS-CE) [x, y]	x y ●———●	KEY BETWEEN x AND y KEY>=x AND KEY<=y		KEY BETWEEN y AND x KEY>=y AND KEY<=x	
RANGE(CS-OE) [x, y]	●———○	KEY>=x AND KEY<y		KEY>=y AND KEY<x	
RANGE(OS-CE) [x, y]	○———●	KEY>x AND KEY<=y		KEY>y AND KEY<=x	
RANGE(OS-OE) [x, y]	○———○	KEY>x AND KEY<y		KEY>y AND KEY<x	
ATS[x1],[x2],‥,[xn](*1)	x1 x2 xn ● ●‥●	KEY IN(x1, x2, ‥, xn)			
ATS[SUBQ(n)](*1)		KEY IN(副問合せ)			
IS NULL	null ●	KEY IS NULL			
IS NOT NULL	————○	KEY IS NOT NULL			
IS TRUE(*2)		contains(KEY, ' ') IS TRUE		(該当せず)	

(*1):V08-04以降のHiRDBでは、pd_apply_search_ats_num(インデクスを用いた検索で、IN述語および限定述語(=ANY(表副問合せ)、=SOME(表副問合せ))を指定する場合、サーチ条件ATSまたはRANGESを適用する絞り込み値の組み合わせ個数の上限)に30000を指定することを推奨します。

HiRDB Version 9 09-50の推奨モードでは、省略値が30000になりました。

(*2): IS TRUEのサーチ条件は、プラグインインデクスに対してのみ使用します。containsは、その一例であり、 Text Search Plug-inが提供する関数です。

3-2-10 サーチ条件の種類 複数列インデクス(KEY1,KEY2,・・・,KEYn)

HITACHI Inspire the Next

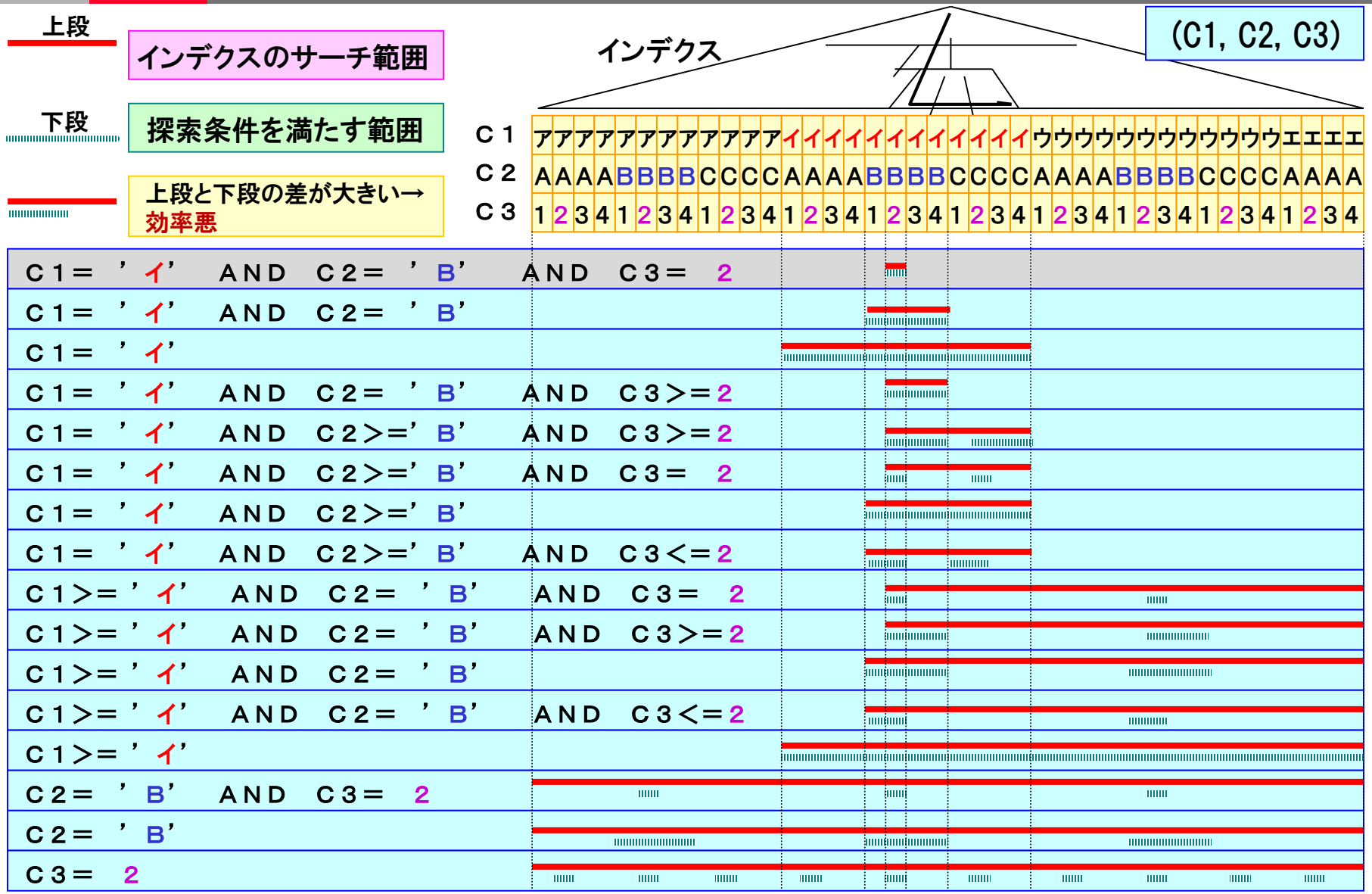
SearchCnd:	インデクス サーチ範囲	探索条件(昇順スキャンの場合)
AT[(x1, x2, ・・・, xn)]	x ●	KEY1=x1 and KEY2=x2 ・・・ KEYn=xn
AT[(x1, NULL, x3・・・, xn)]		KEY1=x1 and KEY2 IS NULL and KEY3=x3 ・・・ KEYn=xn
RANGE(CS-CE) [(v1, v2, ・・・ , x), (v1, v2, ・・・ y)]	x y ●—●	KEY1=v1 and KEY2=v2 ・・・ KEYn BETWEEN x AND y
RANGE(CS-CE) [(v1, x, MIN, ・・・), (v1, y, MAX,・・・)]	x y ●——●	KEY1=v1 and KEY2 BETWEEN x AND y
RANGE(CS-CE) [(x, MIN, ・・・), (y, MAX,・・・)]	x y ●———●	KEY1 BETWEEN x AND y
RANGE(CS-CE) [(x, MIN, ・・・), (MAX, MAX,・・・)]	x ●—————	KEY1>=x
IS NULL	null ●	KEY1 IS NULL and KEY2 IS NULL ・・・ KEYn IS NULL
IS NOT NULL	————○	(作成しない)
ATS、RANGES(*1)	● ●・・・●	KEY1 IN(x1, x2, ・・・, xn) and KEY2=y
RANGE(CS-OE), (OS-CE), (OS-OE)	———	(未サポート)
IS TRUE		(該当しない)

(*1):V08-04以降のHiRDBでは、pd_apply_search_ats_numに30000を指定することを推奨します。

HiRDB Version 9 09-50の推奨モードでは、省略値が30000になりました。

3-2-11　探索条件と複数列インデクスのサーチ範囲

HITACHI
Inspire the Next



（注）表の下3つは、リーフページのフルスキャン（**3-2-12**参照）になり、特に効率が悪いです。

3-2-12 FULL SCAN

サーチ条件種別に(FULL SCAN)が含まれている場合は、インデクス上のすべてのリーフがサーチされることを意味します。

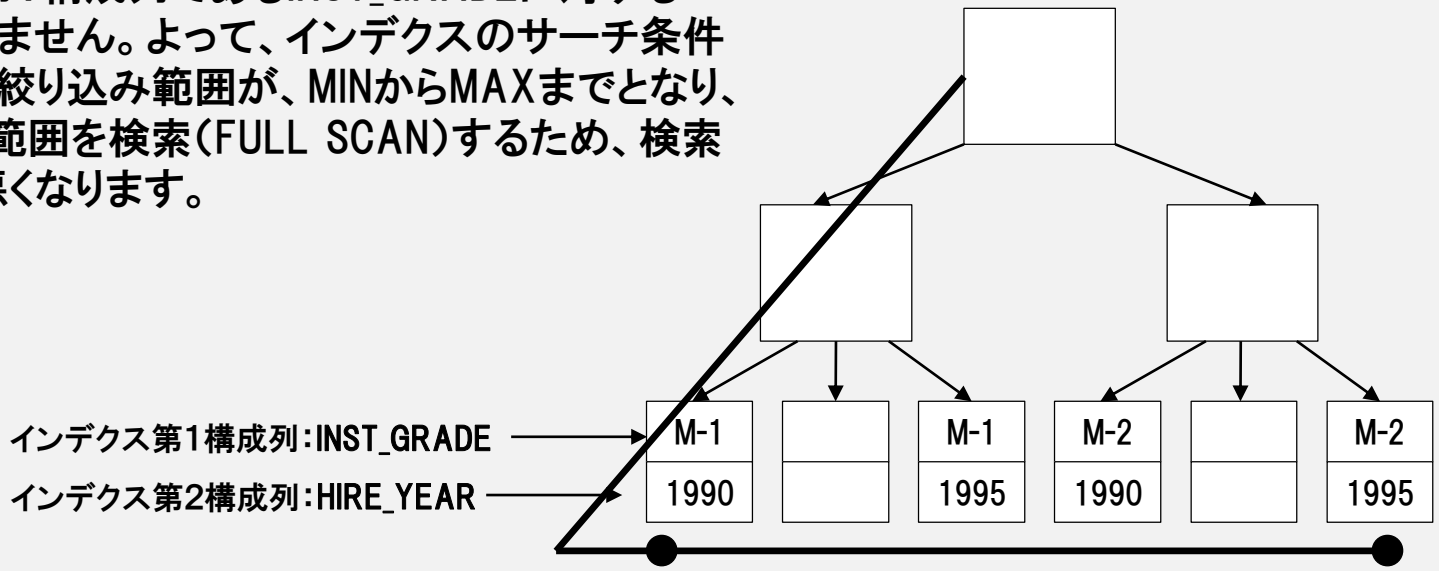
例

<INSTRUCTORS表のインデクス(複数列インデクス)>
(INST_GRADE, HIRE_YEAR)

検索条件:where HIRE_YEAR between 1990 and 1995

サーチ種別:SearchCnd:RANGE(CS-CE) [(MIN,1990),(MAX,1995)] (FULL SCAN)

このSQLでは、第1構成列であるINST_GRADEに対する条件指定がありません。よって、インデクスのサーチ条件の第1構成列の絞り込み範囲が、MINからMAXまでとなり、インデクスの全範囲を検索(FULL SCAN)するため、検索処理の効率が悪くなります。



3-2-13 検索方法のチューニング(3)

HITACHI
Inspire the Next

Scan Type:
[MULTI COLUMNS]
{INDEX | KEY} SCAN

インデクスを利用するスキャンの場合

CHECK
- 利用されるインデクスは、適切か？
- インデクスのFULL SCANになっていないか？
  (サーチ条件がない{SearchCnd:NONE(FULL SCAN)}か、
  またはサーチ条件の第1構成列の絞り込み範囲がMIN～MAX)
- インデクスのサーチ範囲は、広くないか？ または必要な範囲か？
  または指定した条件に対して効率が悪くないか？

ACTION
- 探索条件の絞り込める列(=条件列)をインデクスの構成列の前方に連続して含める。
- インデクスに含まれていない条件列をインデクスの構成列として追加する。
  ただし、更新列は、メンテナンスオーバヘッドが発生するため、含めないことを検討する。

CHECK
絞り込める条件中に演算を含んでいないか？

ACTION
可能ならば、演算を含まないように、SQLを変更する。
(演算を含む条件に対しては、インデクスを有効に利用できない)

例: C1 || C2= 'xxxxxyyyyy' (C1は、固定長) → (C1,C2)=('xxxxx','yyyyy')
SUBSTR(C1, 1, 3) = 'zzz' → C1 LIKE 'zzz%'
C1 * 12 = 60 → C1 = 5
C1 = CASE WHEN C2=1 THEN 'xxx' ELSE 'yyy' END
→ (C1 = 'xxx' AND C2=1) OR (C1 ='yyy' AND C2<>1)

3-2-14 結合方式一覧

HITACHI
Inspire the Next

解説 結合方式とその特徴について説明します。

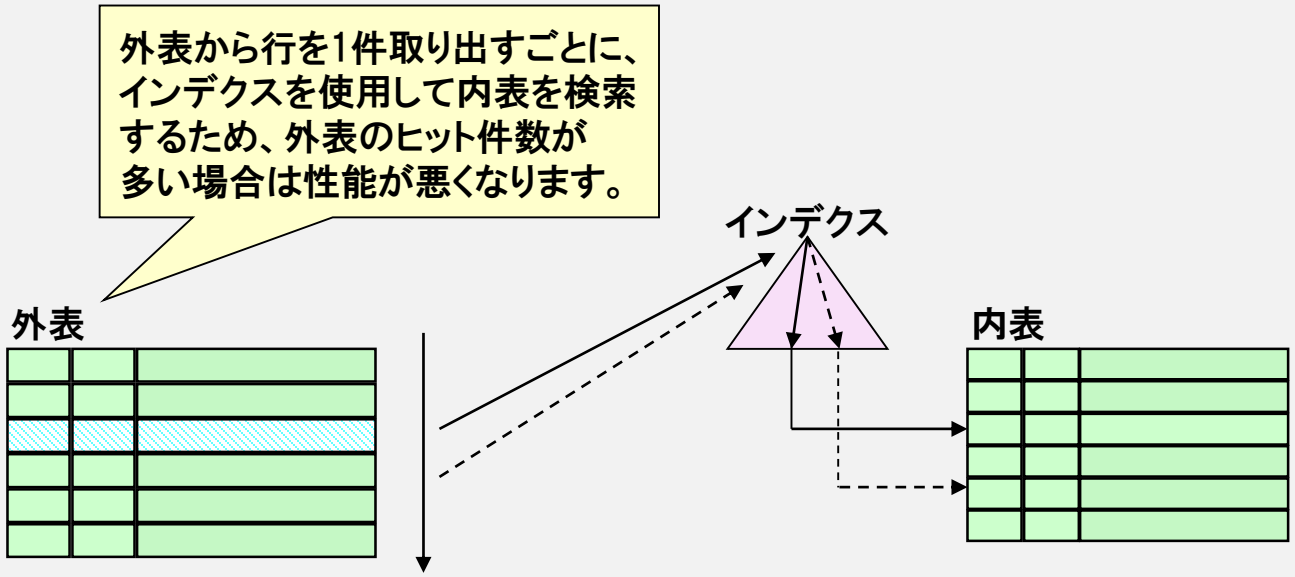
結合方式	特徴
ネストループジョイン (NESTED LOOPS JOIN)	行数を絞り込んで結合する場合に向いている。 インデクス定義が必要。
ハッシュジョイン (HASH JOIN)	大量データの結合に向いている。性能を上げるためには、大量のメモリが必要。 ハッシュジョインを使用するには定義が必要です。 詳細は付録D-9を参照してください。
マージジョイン (MERGE JOIN)	大量データの結合に向いているが、通常は、ハッシュジョインの方が性能がよい。
直積 (CROSS JOIN)	非常に性能が悪い。結合条件の指定方法により、直積か直積以外かが決まる。

3-2-15 結合方式 ネストループジョイン

ネストループジョイン(NESTED LOOPS JOIN)

ネストループジョインは、外表の結合列の値を使用して、内表の結合列に定義されているインデクスをサーチして、突き合わせを入れ子にしたものを繰り返して処理します。

ネストループジョインは、内表にインデクスが定義されていて、外表をかなり絞り込めるときに有効です。



Join Type:

NESTED LOOPS JOIN

3-2-16 結合方式 ハッシュジョイン

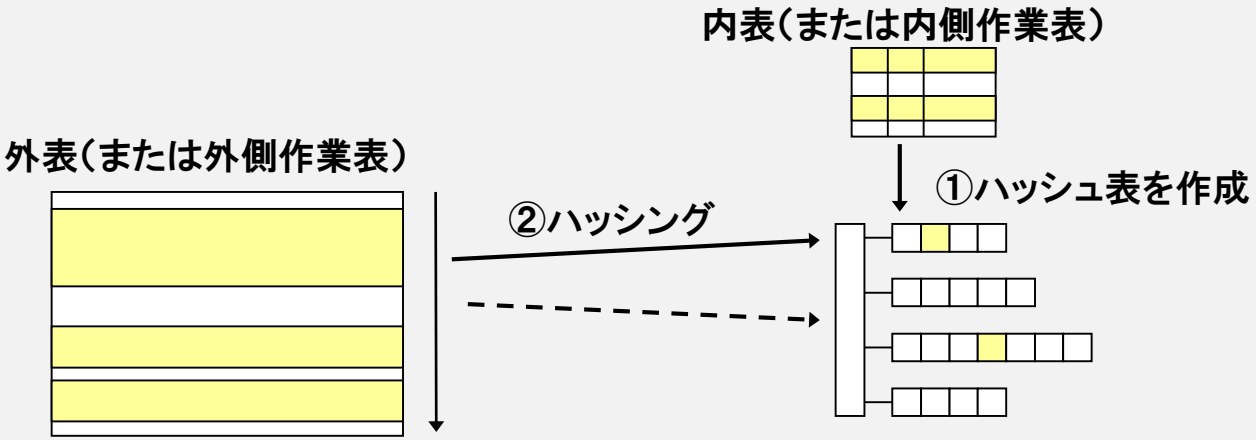
HITACHI
Inspire the Next

ハッシュジョイン(HASH JOIN)

ハッシュジョインは、あらかじめ内表の結合列の値でハッシングしてハッシュ表を作成しておいて、外表を1行取り出すごとに外表の結合列の値でハッシングして、内表から作成しておいたハッシュ表と突き合わせて結合します。

ハッシュジョインは、外表のヒット件数が多く、内表のヒット件数が少ない場合に、有効です。

内表のヒット件数が多い場合は使用するメモリが多くなります。
また、メモリを使用できなくなった分については、いったんファイルに退避するため、性能が悪くなります。



Join Type:

HASH JOIN

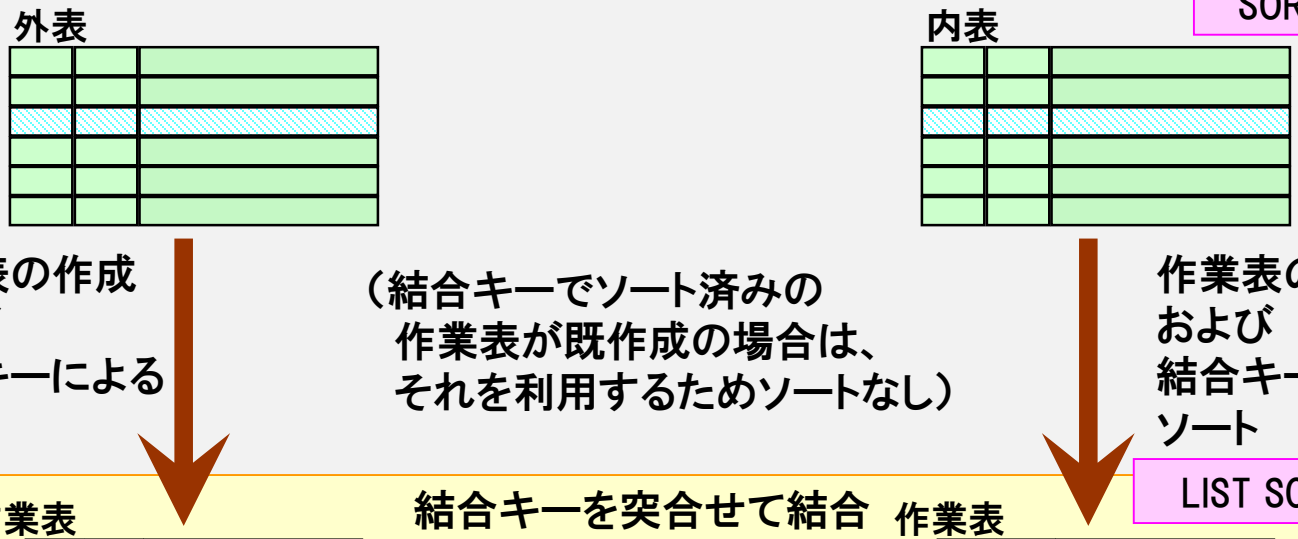
3-2-17 結合方式 マージジョイン(１)

HITACHI
Inspire the Next

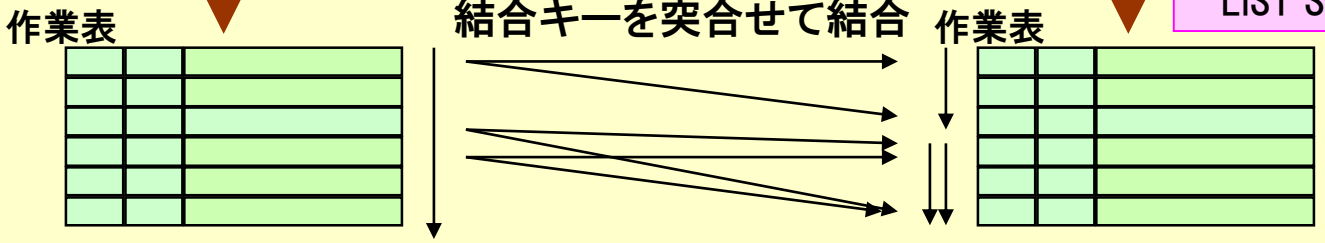
マージジョイン(MERGE JOIN)

マージジョインは、結合列でソートして、結合列の値が小さいものから順に突き合わせ処理をします。

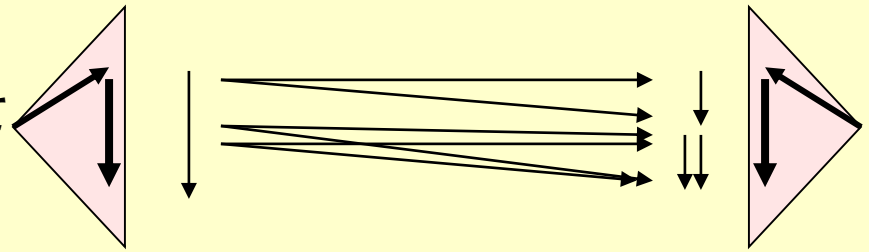
SORT MERGE JOIN



LIST SCAN MERGE JOIN



KEY SCAN MERGE JOIN
（シングルサーバのみ）



3-2-18 結合方式 マージジョイン(２)

HITACHI
Inspire the Next

Join Type:	外表	内表
SORT MERGE JOIN	SORT	SORT
KEY SCAN MERGE JOIN	KEY SCAN	KEY SCAN
LIST SCAN MERGE JOIN	LIST SCAN	LIST SCAN
L-KEY R-LIST MERGE JOIN	KEY SCAN	LIST SCAN
L-KEY R-SORT MERGE JOIN	KEY SCAN	SORT
L-LIST R-SORT MERGE JOIN	LIST SCAN	SORT
L-LIST R-KEY MERGE JOIN	LIST SCAN	KEY SCAN
L-SORT R-KEY MERGE JOIN	SORT	KEY SCAN
L-SORT R-LIST MERGE JOIN	SORT	LIST SCAN

■結合キーのインデクスがなくても結合可能です。
■データの並びによる影響がネストループジョインと比べて少ないです。
■外表および内表の両方の件数が多い場合に、ソートの負荷増加がなければ、有効です。

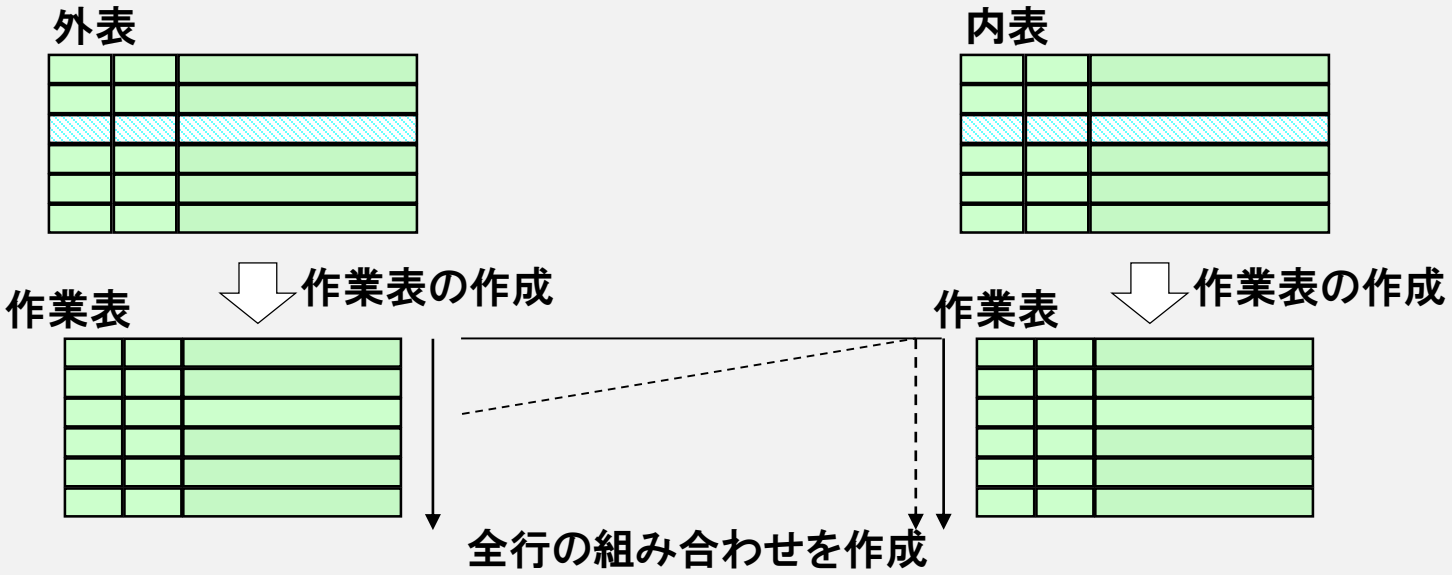
3-2-19 結合方式 直積

HITACHI
Inspire the Next

直積(CROSS JOIN)

直積は、外表のすべての行と、内表のすべての行をそれぞれ組み合わせて結合します。2表にわたった条件があれば、結合した後に判定します。

ネストループジョイン、ハッシュジョイン、マージジョインのいずれの結合方式も適用出来ない(各結合方式に有効な結合条件が存在しない)場合に使用します。



Join Type:

CROSS JOIN

3-2-20 結合検索のチューニング(１)

HITACHI
Inspire the Next

ネストループジョインの場合

Join Type:NESTED LOOPS JOIN

CHECK

内表の結合キーのインデクスは、適切か？
（結合キーと一致するインデクスか、または
結合キーのすべての列が第1～第n構成列に連続して含まれるインデクスか？
または不連続な場合、第1～第n構成列は、結合列か=条件列かIS NULL条件列）

```
SELECT * FROM T1, T2
    WHERE T1.C1=10
     AND T2.C2=T1.C2  AND T2.C3=T1.C3  AND T2.C4=T1.C4
```

内表T2の結合キー(C2, C3, C4)

INDEX ON T2(C2)	×
INDEX ON T2(C3)	×
INDEX ON T2(C2, C3)	×
INDEX ON T2(C2, C3, C4)	○
INDEX ON T2(C4, C3, C2)	○
INDEX ON T2(C2, C4, C3)	○
INDEX ON T2(C2, C3, C4, C5)	○
INDEX ON T2(C2, C3, C5, C4)	△

ACTION

外表の各行に対して、内表が繰返し検索されるネストループジョインで、内表検索時に、行データを参照しないと結合条件を評価できないようなインデクス利用効率の悪い検索を行うと、性能が著しく悪くなる。×→インデクス定義を見直してください。

3-2-21　結合検索のチューニング(２)

HITACHI
Inspire the Next

ネストループジョインの場合

Join Type:NESTED LOOPS JOIN

CHECK
内結合の場合に、外表と内表を入れ替えるには、

ACTION
次のいずれかまたはいくつかの組合せによって、外表と内表との入れ替えが可能となる可能性がある。

- 外表にしたい表には、結合列を第1構成列に含むインデクスを定義しない。
- 内表よりも、外表の方が、インデクスを利用して、サーチ条件で効率良く絞り込めるように、インデクスを定義する。
- どちらの表もインデクスによる絞込みが不可でテーブルスキャンになる場合は、最適化情報取得ユティリティで各表の行数を設定する。この場合、行数の少ない表が外表となる。
- 結合表構文(INNER JOIN)によって、外表、内表を指定する。

外表と内表を入れ替えることによって、性能を改善できる場合があります。

CHECK
ソートキーの構成列がすべて最外表の列ならば、

ACTION
インデクスを利用して、内部ソート処理を削除できる場合があります。
詳細は、付録D-3～D-6「インデクスとソート」を参照してください。
←ネストループジョインの場合、結合後も、最外側の行の順序が保存される。

3-2-22 結合検索のチューニング(3)

HITACHI
Inspire the Next

マージジョインの場合

Join Type:SORT MERGE JOIN

データのクラスタリング度やグローバルバッファの割当て方にもよりますが、特に、ソートを伴なうマージジョインになっている場合には、ネストループジョインにすることで、性能が改善できる場合があります。

CHECK
結合条件が=条件のとき、ネストループジョインにするには、
(ハッシュジョインをネストループジョインにするのも同様)

ACTION
①と②、③のどちらか一方を満たすようにしてください。

①内表に有効なインデクスを定義する。
②結合表に対してSQL最適化指定する。詳細は、付録D-10～D-14を参照してください。
③SQL最適化オプション(クライアント環境定義PDSQLOPTLVLまたはシステム定義pd_optimize_level)を指定する。詳細は、HiRDBマニュアル「UAP開発ガイド」または「システム定義」を参照してください。

◆どちらを外表にすればよいか？
一般に、絞込み率の良い条件を指定している表を外表にするのが良い。
絞り込み率が同程度ならば、絞り込み後の行数が少ない方を外表にする。ただし、行数が少ない表を内表にして、完全に、バッファに載せた方が良い場合もある。

3-2-23 結合検索のチューニング(４)

HITACHI
Inspire the Next

直積の場合

Join Type:CROSS JOIN

CHECK
SQL中に結合条件が抜けていないか？

ACTION
結合条件の指定が抜けているならば、SQLを修正する。

CHECK
2表間の条件がOR条件になっていないか？

ACTION
直積を行ってから、2表間の条件が評価されているので、
2表間の条件がOR条件を含まなくなるように、SQLを変形し、CROSS JOINにならないようにする。

```
SELECT DISTINCT * FROM T1, T2
    WHERE T1.C1=10 AND (T2.C2=T1.C2 OR T2.C3=T1.C3)
```

```
SELECT * FROM T1, T2  WHERE T1.C1=10 AND T2.C2=T1.C2
UNION
SELECT * FROM T1, T2  WHERE T1.C1=10 AND T2.C3=T1.C3
```

```
SELECT * FROM T1, T2
    WHERE T1.C1=10 AND (T2.C2=T1.C2 OR T2.C3=T1.C3)
```

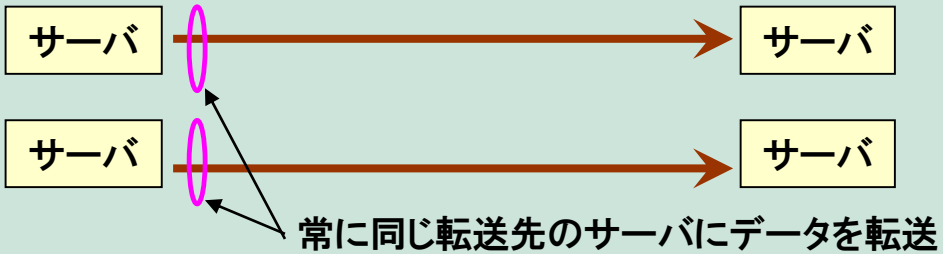
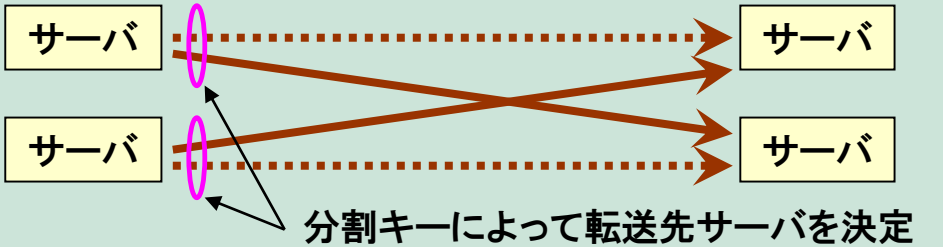
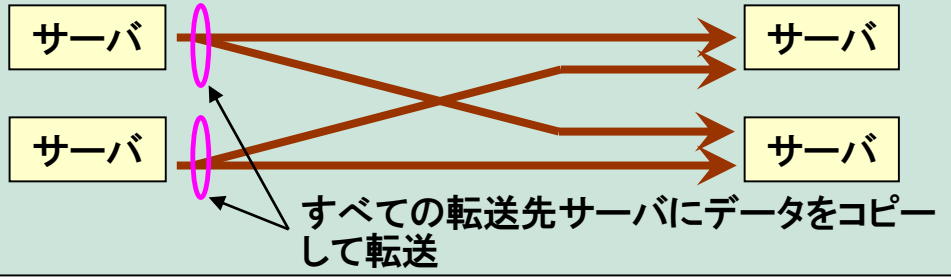
```
SELECT * FROM T1, T2  WHERE T1.C1=10 AND T2.C2=T1.C2
UNION ALL
SELECT * FROM T1, T2  WHERE T1.C1=10 AND T2.C3=T1.C3
   AND (T2.C2<>T1.C2 OR T2.C2 IS NULL OR T1.C2 IS NULL)
```

NOT NULL列ならば、IS NULL述語不要

3-2-24 データ転送方法(パラレルサーバ)

HITACHI
Inspire the Next

解説 パラレルサーバで表の結合する際、BES間のデータの転送をともないます。分割表の結合では、表の分割キーを結合キーに含むことで、効率よく処理できます。

BES間データ転送方法の種類	方式 転送元 → 転送先	転送方法の条件
1対1転送 (1 TO 1)		下記の条件をすべて満たす場合。 ・両方の表の分割キー、分割の種類、分割条件、格納先BESが完全に一致している。 ・両方の表の分割キーが結合キーに含まれている。
キーレンジ転送 (KEY RANGE) ハッシュ転送 (HASH)		下記の条件をすべて満たす場合。 ・データ転送先の表がキーレンジ分割表またはハッシュ　分割表。 ・転送先の表の分割キーが結合キーに含まれている。
ブロードキャスト転送 (BROADCAST)		分割キーが、結合キーに含まれていない。 (注)分割数が多く、外表のヒット件数が多いほど、通信および結合オーバヘッドが大きくなり、性能が悪くなる。

3-2-25　結合検索のチューニング(６)

HITACHI
Inspire the Next

ネストループジョイン(パラレルサーバ)の場合

Join Type:
NESTED LOOPS JOIN
BROADCAST

CHECK

ネストループジョイン時の転送方法がブロードキャスト転送になっていないか？

ACTION

・ネストループジョインの内表となる表の分割方法は、フレキシブルハッシュ分割にしないで、キーレンジ分割またはFIXハッシュ分割にし、内表の分割キーを結合キー(または結合列)にすることで、キーレンジ分割転送またはハッシュ分割転送にすることができる。

・内表・外表の表の分割方法を、キーレンジ分割またはFIXハッシュ分割にし、同じキー値の同じBESのRDエリアに格納されるように、分割方法・条件も合せると、1 TO 1転送にすることができる。

ただし、一つの表に結合キーになりうるキーが複数存在する場合があるので、その場合は、実行頻度や効果を考慮して、分割キーを選択する必要があります。

```
SELECT * FROM T1, T2
   WHERE T1.C1=10 AND T2.C2=T1.C2  AND T2.C3=T1.C3
```

結合キー(C2, C3)

T2の分割キー(C2, C3, C4)	×
T2の分割キー(C2, C3)	○
T2の分割キー(C2)	○

3-2-26 結合検索での表の分割列と結合条件列の関係

HITACHI Inspire the Next

パラレルサーバ限定

解説

結合検索時に表の分割列で結合できるように表を設計してください。
特にLEFT OUTER JOINの場合は、内表の分割列で結合できるように表を設計してください。

JUTYU表、ZAIKO表ともにDNOが分割列の場合

```
SELECT ZA.NAME
FROM JUTYU JU, ZAIKO ZA
WHERE JU.CNO = ZA.CNO
      AND JU.CNO = 10 ;
```

結合列に表の分割列を含まない
⇒JUTYU表データのBROADCAST転送が発生し、負荷が高くなる

×

```
SELECT ZA.NAME
FROM JUTYU JU
          LEFT OUTER JOIN ZAIKO ZA
          ON ZA.DNO = ZA.CNO
WHERE JU.CNO = 10 ;
```

内表の結合列に分割列を含まない
⇒マージジョインとなり、両表のデータ転送、ZAIKO表の全件検索、作業表作成、ソート処理が発生し、負荷が高くなる

×

```
SELECT ZA.NAME
FROM JUTYU JU, ZAIKO ZA
WHERE JU.CNO = ZA.DNO
      AND JU.CNO = 10 ;
```

内表（通常の結合の場合は、絞り込まない方の表）の結合列に分割列を含む
⇒JUTYU表のデータをZAIKO表の分割に合わせてキーレンジ転送またはハッシュ転送し効率が良い

○

⇒さらに表の分割方法が同じなら1TO1転送となり最も処理効率が良い

◎

HITACHI
Inspire the Next

3. SQLのチューニング

3.1 評価ポイント

3.2 チューニング

3.3 確認するチューニング情報

3.4 チューニング情報の見方

3-3-1　確認するチューニング情報

HITACHI
Inspire the Next

解説　チューニング情報から以下の確認項目を確認してください。

確認項目	取得するチューニング情報	確認項目の見方
SQL実行時間	SQLオブジェクト用バッファの統計情報	3-4-1参照
	UAP統計レポート	3-4-2参照
スキャンタイプ 結合方式　等		3-4-3参照
	アクセスパス情報	

HiRDB SQL Tuning Advisorでも、SQL実行時間の確認、アクセスパス解析ができます。

HiRDB SQL Executer(pdsqlコマンド)でも、SQL実行時間が確認できるようになりました。
詳細は、**3-4-4**を参照してください。

■手順

1. SQLオブジェクト用バッファの統計情報または
   UAP統計レポートでSQL実行時間を確認します。

2. 問題のあるSQLについてアクセスパス表示ユティリティ(pdvwopt)
   またはUAP統計レポートで取得したアクセスパス情報から、
   スキャンタイプ、結合方式等を確認します。

HITACHI
Inspire the Next

3. SQLのチューニング

3.1 評価ポイント

3.2 チューニング

3.3 確認するチューニング情報

3.4 チューニング情報の見方

3-4-1　SQL実行時間の見方 SQLオブジェクト用バッファの統計情報

HITACHI Inspire the Next

解説

SQLオブジェクト用バッファに格納されている統計情報から、SQLの実行時間を確認してください。ここで確認できるSQLの実行時間は、秒単位の値です。より厳密な実行時間を確認するためには、UAP統計レポートで確認してください。

pdobils

```
<< SQL OBJECT LIST >>
*SQL OBJECT NO              : 1
 STATUS                    : LRU
 TYPE                      : DYNAMIC SQL
 SIZE(B)                   : 2240
 EXECUTE COUNT             : 15
 # EXECUTE TIME AVG(s)     : 2.000000
 # EXECUTE TIME MAX(s)     : 6.000000
 # SERVICE NAME            :
 # UAP NAME                : Unknown
 # CONNECT NO              : 5
 # SQL NO                  : 10
 # RECORD DATE/TIME        : 2011/06/23 16:09:05
 DB REFERENCE GET COUNT    : 30 AVG 2 MAX 2
 DB UPDATE GET COUNT       : 0 AVG 0 MAX 0
 DB READ COUNT             : 2 AVG 0 MAX 2
 DB WRITE COUNT            : 0 AVG 0 MAX 0
--- (略) ---
 WKFILE READ COUNT         : 0 AVG 0 MAX 0
 WKFILE WRITE COUNT        : 0 AVG 0 MAX 0 ISOLATION LEVEL          : 2
 OPTIMIZE LEVEL            : 657056
 ADDITIONAL OPTIMIZE LEVEL : 1
 DEFAULT SCHEMA            : HIUSER
 SQL                       : select inst_id,inst_name,hire_year,inst_grade from instructors
                             where inst_id=9001
```

平均実行時間

最大実行時間

3-4-2 SQL実行時間の見方 UAP統計レポート

HITACHI
Inspire the Next

解説 UAP統計レポートのSQL単位の情報でSQL実行時間を確認してください。

```
CNCT  CLPID CLTID NO     OP    SEC  SQL  SQL   START-TIME    END-TIME      OP    EXEC-TIME
NO                       CODE  NO   CODE WARN                             TION
----- ----- ----- ------ ----- ---- ---- ----- ------------ ------------ ----- -----------
    2  3020  1960      1 CNCT    0    0 -0000 11:29:38.125 11:29:39.031 0000       903606
    2  3020  1960      2 SET   501    0 -0000 11:29:39.250 11:29:39.281 0D00        33653
*SQL* select inst_id,inst_name,hire_year,inst_grade from instructors where inst_id=9001
00:00:00.033653   00:00:00.016000  17653    0    0    0 2112    0    0
```

SQL実行時間

サーバ側でのSQL実行時間

```
--------------------------------------------------------------------------------
Section No      : 501
UAP Source      : dahmhird.ec
Optimize Mode   : COST_BASE_2
SQL Opt Level   : 0x000a06a0(657056) =
"PRIOR_NEST_JOIN"(32),"PRIOR_OR_INDEXES"(128),"DETER_AND_INDEXES"(512),"RAPID_GROUPING"(1024),
"DETER_WORK_TABLE_FOR_UPDATE"(131072),"APPLY_ENHANCED_KEY_COND"(524288)
Add Opt Level   : 0x00000001(1) = "COST_BASE_2"(1)
Work Table      : 0
Total Cost      : 0.077148

 ----- QUERY EXPRESSION BODY ID : 1 -----
```

3-4-3 アクセスパス情報の見方 UAP統計レポート

HITACHI
Inspire the Next

解説 UAP統計レポート中のアクセスパス情報の詳細を解説します。中間結果情報の見方については、付録Cを参照してください。アクセスパス表示ユティリティ(pdvwopt)も同様の情報が取得できます。

```
SELECT A.C1VC, B.C1I, A.C6DATE FROM STABLE01 A LEFT OUTER JOIN STABLE02 B
ON A.C1VC = B.C3VC and B.C1I <30 WHERE A.C7I = 10
```

実行したSQL

■アクセスパス情報出力例

```
Result of SQL Optimizer :
Connect No      :    2
--------------------------------------------------------------------------------
Section No      : 1
UAP Source      : pdsqlw-2264
Optimize Mode   : COST_BASE_2
SQL Opt Level   : 0x00020400(132096) = "RAPID_GROUPING"(1024),"DETER_WORK_TABLE_FOR_UPDATE"(131072)
Add Opt Level   : 0x00000001(1) =  “COST_BASE_2” (1)
Work Table      : 0
Total Cost      : 201.050172
----- QUERY ID : 1 -----
Query Type      : QUERY
JOIN
 # Join  ID     : 1
   L Table      :  STABLE01(A) 0x00020082(131202)
   R Table      :  STABLE02(B) 0x0002007f(131199)
   Join Type    : 1-CLM NESTED LOOPS JOIN(LEFT OUTER)
SCAN
 # Table Name   : STABLE01(A) 0x00020082(131202)
   Cost         : N (10000000ROW)  {T-21154.231114,I-45.004596,AND-53.342219}
   RDAREA       : NON DIVISION  (1RD) [0x06(6)] ALL
   Scan Type    : INDEX SCAN
   Index Name   : STABLE01I_1 0x00030105(196869) (1) (+C7I)
                  SearchCnd : AT [10]
... (略) ...
```

SQL最適化オプション

SQL拡張最適化オプション

静的な情報から算出したコスト情報

外表

内表

表の結合順序

結合方式の種類

データアクセス方法(スキャンタイプ)の種類

使用しているインデクス名

絞り込み条件(サーチ条件)の種類

3-4-4 SQL実行時間の見方 HiRDB SQL Executer(pdsql)

HITACHI Inspire the Next

HiRDB SQL Executer(pdsql) 09-05(2015年4月リリース)の新機能

解説 本番環境でクライアント環境定義の変更ができず、SQLトレースを取得できないような場合に、HiRDB SQL Executerで簡単にSQLの実行時間を測定できるようになります。

```
COMMAND ?      +----2----+----3----+----4----+----5----+----6----+----7----+

SET EXECTIME DETAIL;

KFPX27092-I    Processing of SET EXECTIME command completed

COMMAND ?      +----2----+----3----+----4----+----5----+----6----+----

SELECT COUNT(*) FROM USER1.CUSTOM;

KFPX27010-I              1 rows selected
SEC  OP    START-TIME                 END-TIME                   EXEC-TIME    COUNT        MAX          MIN
NO   CODE                                                                                  EXEC-TIME    EXEC-TIME
---- ---- -------------------------- -------------------------- ------------ ------------ ------------ ------------
   1 SET  2015/04/06 17:30:37.204 2015/04/06 17:30:37.204          520
   1 DESC 2015/04/06 17:30:37.204 2015/04/06 17:30:37.204          280
   1 OPEN 2015/04/06 17:30:37.204 2015/04/06 17:30:37.205          234
   1 1stF 2015/04/06 17:30:37.205 2015/04/06 17:30:37.205          471
   1 OthF 2015/04/06 17:30:37.208 2015/04/06 17:30:37.208          247            1          247          247
   1 CLOS 2015/04/06 17:30:37.209 2015/04/06 17:30:37.210          249
   0 CMIT 2015/04/06 17:30:37.210 2015/04/06 17:30:37.210          278
TOTAL:                                                            2279
ELAPSED:                                                          6442
```

HiRDB SQL ExecuterのSET EXECTIMEコマンドにDETAILを設定することで、SQLの実行時間を測定できます。

以下のコマンドの指定を組み合わせることで、測定結果をファイルにも出力できます。
- SET RESULT FILEOUT ON ファイル名
- SET ECHOBACK ON

HiRDBクライアントに処理を要求した時刻

HiRDBクライアントから応答を受け取った時刻

初回を除いたFETCHを実行した回数

HiRDBクライアントに処理を要求してから応答を受け取るまでの時間

初回を除いたFETCHを実行した実行時間のうち最大実行時間、最小実行時間

実行時間の合計

SQL ExecuterがSQLの入力を受け付けてから結果を示すメッセージを表示するまでの時間

HITACHI
Inspire the Next

4. ユティリティのチューニング

HITACHI
Inspire the Next

4. ユティリティのチューニング

4.1 データベース作成ユティリティ(pdload)のチューニング

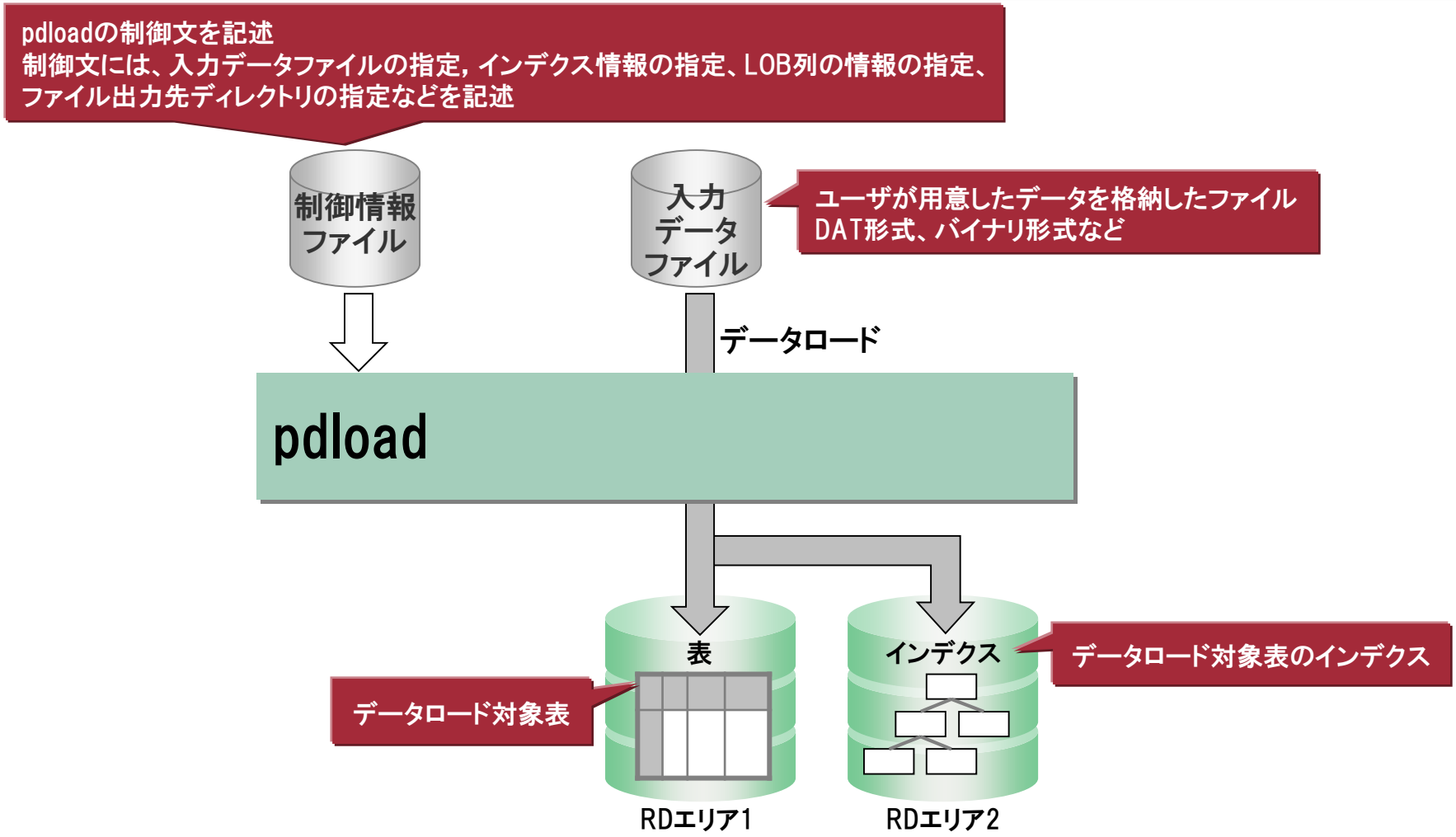
4.2 データベース再編成ユティリティ(pdrorg)のチューニング

4-1-1　データベース作成ユティリティ(pdload)の概要

HITACHI
Inspire the Next

解説

データベース作成ユティリティ(pdload)は、ユーザが用意したデータを表へデータロードするユティリティです。以下にデータロードの概要を示します。



4-1-2 pdloadコマンド実行時の検討ポイント(１)

HITACHI
Inspire the Next

解説 pdloadコマンド実行(データロード)時の検討ポイントに該当する場合は、対策を検討してください。

項目	検討ポイント	対策
初期データロード (-dあり)	インデクス作成方法	インデクス作成時間が速い一括作成モード(-i c)がお勧めです。また、一括作成の場合インデクスキー値の格納の乱れも少ないため、検索性能も期待できます。
	ログ取得方式	大量データの場合、1表／RDエリア(データロード対象表以外の表が無い)なら、ログレスモード(-l n)にすることで、データロード時間を速くできます。(注1)
追加データロード (-dなし)	インデクス作成方法	＜追加するデータ件数が母体の1割以下＞ 逐次追加(-i s)でも良いです。 ただし、日に複数回データロードする場合は、データベース再編成ユティリティ(pdrorg)で、インデクスの再作成を行ってください。 ＜上記以外の場合＞ 一括作成モード(-i c)がお勧めです。 ただし、pdloadコマンドを複数回に分けて実行する場合は、最後のpdloadコマンドの実行以外はインデクスを作成しない(-i x)で、最後のpdloadコマンドの実行でインデクスを一括作成(-i c)してください。
	ログ取得方式	ログ取得モード(-l a または -l p)を指定して、ログを取得してください。(注1)

注1:ログ取得モードに-l nまたは-l pを指定時は、バックアップ運用に応じて、pdloadコマンド実行前後でのバックアップの取得要否を検討してください。

4-1-3 pdloadコマンド実行時の検討ポイント(２)

HITACHI
Inspire the Next

項目	検討ポイント	対策
入力データファイル	入力データファイルの形式	データロード時間が一番速いバイナリ形式(-b)がお勧めです。(*1) DAT形式(csv形式)と比べて20%程度速いです。
	＜Windowsの場合＞ 入力データファイルの大きさ	入力データファイルの大きさが数百ギガバイト程度あり、入出力に使用するバッファ長(pd_utl_file_buff_size(*2))が小さい(デフォルト値は32キロバイト)場合、バッファ長を大きく(1メガバイト程度)してディスクI/O回数を削減してください。 小さいと大量のディスクI/O処理が実行されることにより、OS制御(Windowsのファイルキャッシュアクセス)によってCPU利用率が上がり、ディスクI/O処理が遅延します。
データロード対象表	表の属性	FIX属性にできる場合は、データロード時間が一番速いFIX属性がお勧めです。

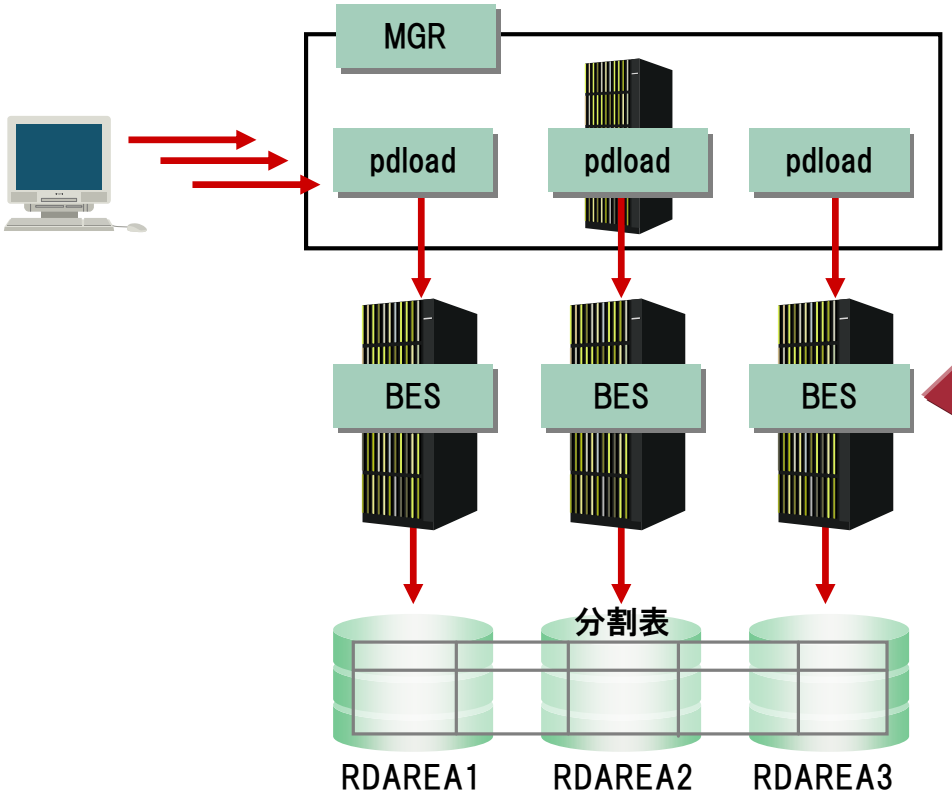
(*1):HiRDBからHiRDBへの移行では、データベース再編成ユティリティ(pdrorg)のアンロードでバイナリファイルを作成すると良いです。移行元がHiRDB以外ならHiRDB Dataextractorでバイナリファイルが生成できます。

(*2): バージョン09-50の変更点 pd_utl_file_buff_sizeの省略値が1メガバイトになりました。

4-1-4 横分割表へのデータロード

HITACHI
Inspire the Next

解説 横分割表へのデータロードは、RDエリア単位に並列実行すると速いです。



HiRDBでは、Shared Nothing方式を採用しており、
1つのRDエリアには、
1つのサーバでのみアクセスするため、
他サーバの影響を受けることなく、
独立して処理を実行できます。

4-1-5 横分割表へのデータロードの検討ポイント

HITACHI
Inspire the Next

解説 横分割表へのデータロードで検討ポイントに該当する場合は、要件に合った方法を検討してください。

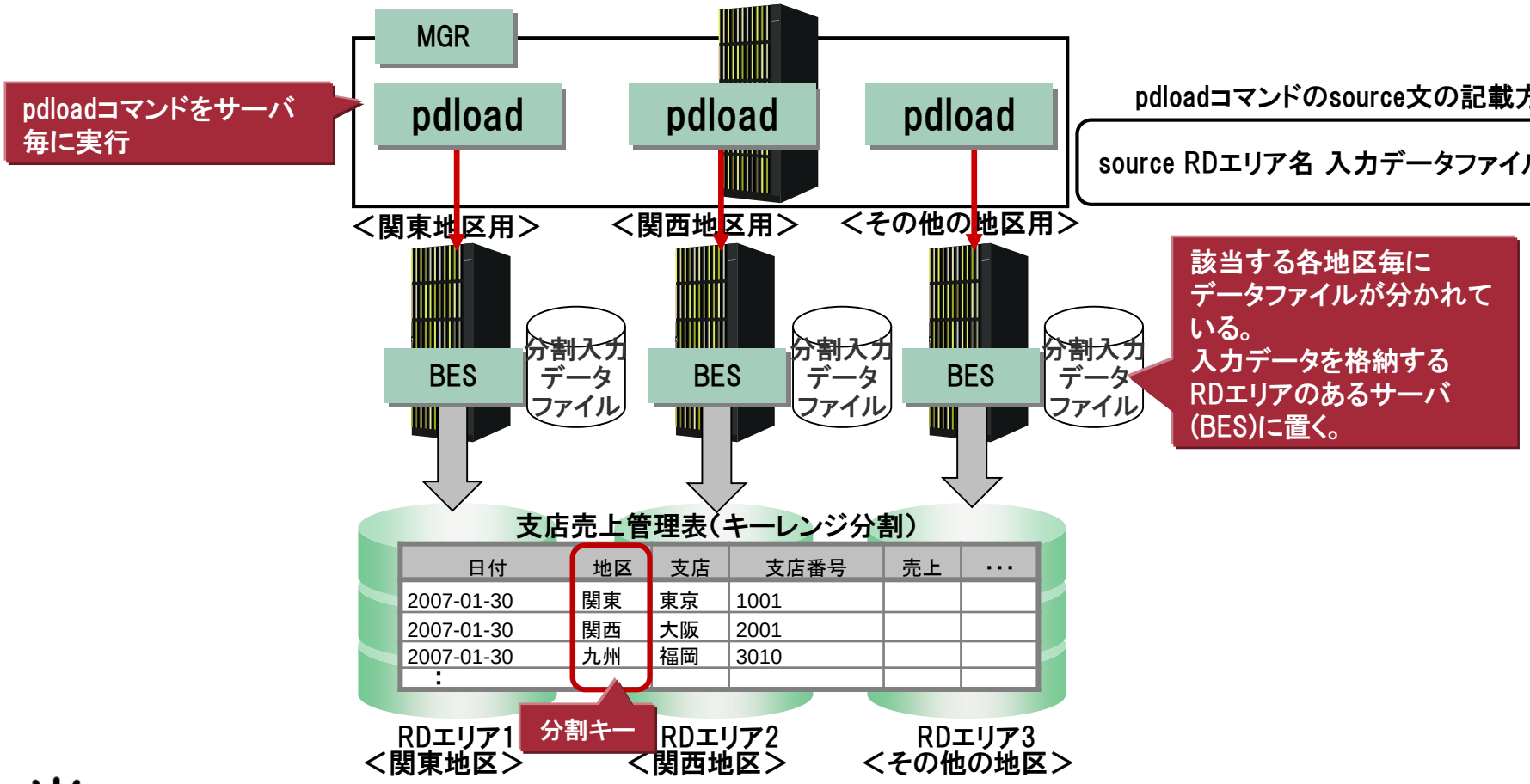
項目	検討ポイント	データロード方法
入力データファイル	入力データファイルがRDエリア毎に分かれているか、容易に分けられる	pdloadの時間を短くできるため、RDエリア単位にデータロードを並列実行する方法(**4-1-6**参照)がお勧めです。
	上記以外	データロード運用が容易なパラレルローディング機能(**4-1-7**参照)がお勧めです。

4-1-6 横分割表へのデータロード方式(1)

HITACHI
Inspire the Next

解説 横分割表にデータをロードする場合、格納するRDエリア単位に入力データファイルを分割してデータロードを並列実行すると、データロードに掛かる時間を短縮でき、表の占有時間が短縮できます。

効果:RDエリア毎にpdloadコマンドが並列で実行され、データロードに掛かる処理時間を短縮できます。



入力データが分割されていない場合は、pdloadのsrc_work文指定で表単位の入力データファイルから、RDエリア単位の入力データファイルを作成できます。

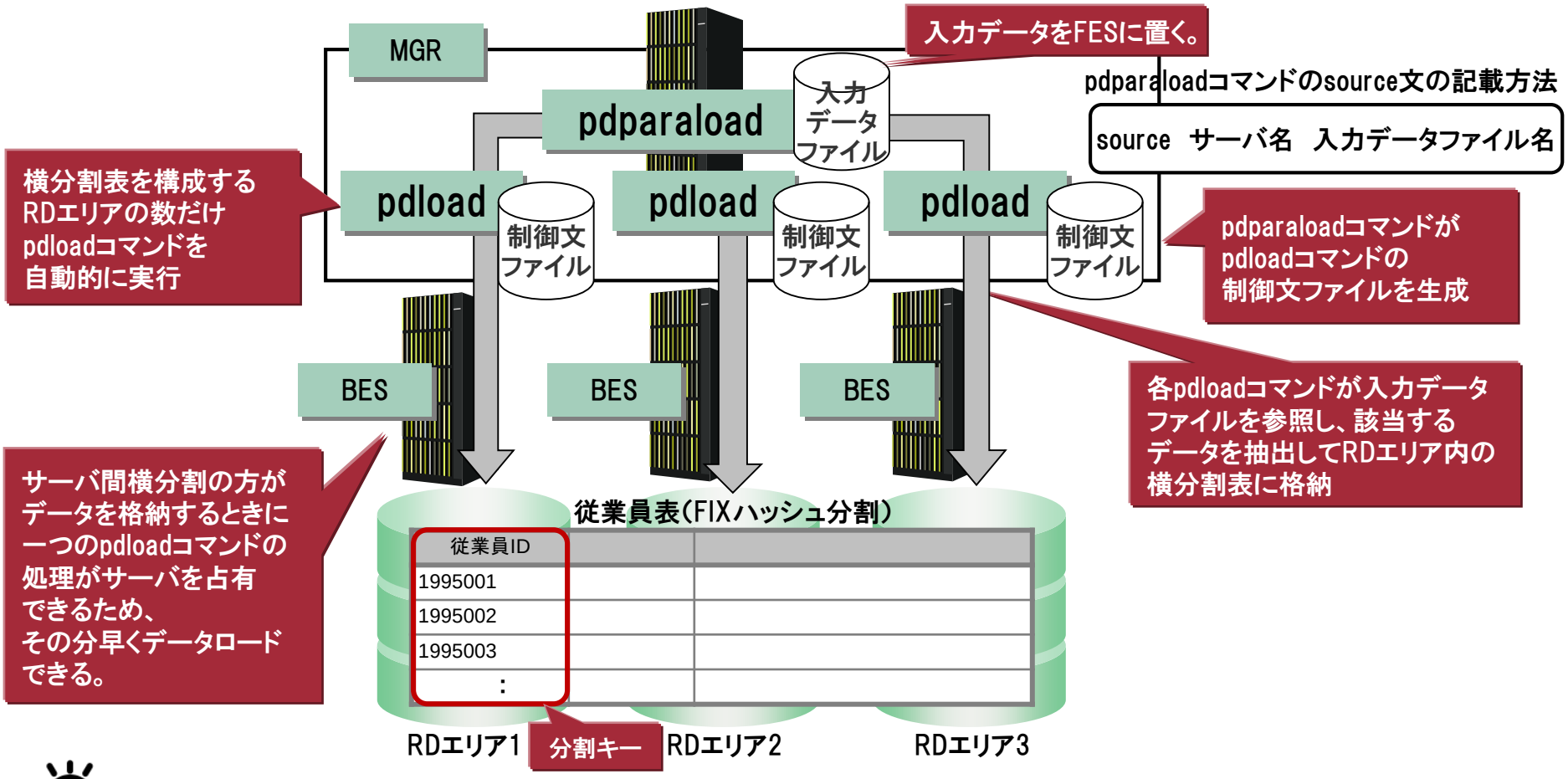
4-1-7　横分割表へのデータロード方式(２)

HITACHI
Inspire the Next

解説

横分割表にデータをロードする場合は、パラレルローディング機能を使う方法もあります。
一つの入力データファイルから横分割表を構成する複数のRDエリアに対してデータロードを並列実行する機能です。

効果：一つの入力データファイルと1回のコマンド入力で済むため、運用が容易です。

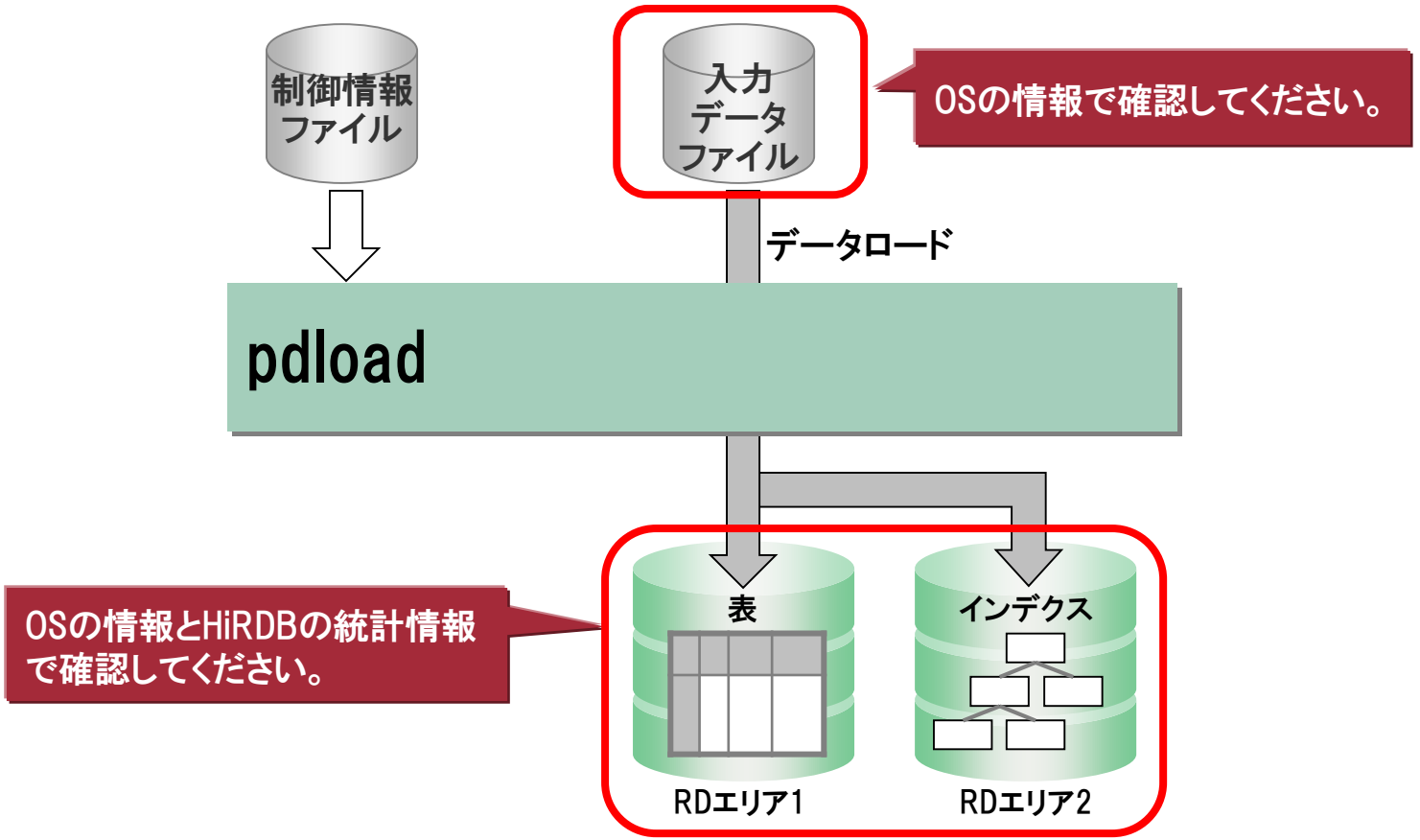


入力データをRDエリア毎に分割するのが難しい場合に、お勧めです。

4-1-8 pdloadで性能遅延が発生する場合

HITACHI
Inspire the Next

解説 pdloadで性能遅延が発生する場合、入力データファイルまたはRDエリア構成ファイルが存在するディスクのI/O性能が原因の場合が多いです。OSの情報やHiRDBの統計情報で切りわけを行います。



4-1-9 確認するチューニング情報：OSの情報

HITACHI
Inspire the Next

解説 OSの情報で、入力データファイルまたはRDエリア構成ファイルが存在するディスクのI/O性能が劣化しているか調査します。調査方法を以下に示します。

OS	取得する情報	確認内容
UNIX	sarコマンドの-dオプション指定で、リアルタイムのディスク入出力統計情報を取得	□Linuxの場合 svctmや%utilが高いデバイスで遅延 □HP-UX、AIXの場合 %busyやavservが高いデバイスで遅延
Windows	パフォーマンスモニタ	PhysicalDiskの% Disk Time(ディスクI/O処理時間の割合)が高いディスクで遅延

4-1-10　確認するチューニング情報：RDエリアのI/O時間の確認

HITACHI Inspire the Next

解説　HiRDBの統計情報で、RDエリア構成ファイルが存在するディスクのI/O性能が劣化しているか調査します。 調査方法を以下に示します。

データロード時に使用するバッファ	取得する情報	確認内容
グローバルバッファ	デファードライト処理に関する統計情報のDAT形式ファイル	実行時間に占める動作要因(CAUSE)が「トリガ:"T"」や、「RDエリアのシンクポイント:"R"」の「合計WRITE時間(DWSUM)」の割合を確認します。 割合が多ければ、I/Oで時間が掛かっていることになります。 統計情報の出力形式については、**4-1-11**を参照してください。
ローカルバッファ	データベースの入出力に関する統計情報 HiRDB Version 9 09-50サポート	統計情報より、READ単価、WRITE単価を確認します。 単価が高ければ、I/Oで時間が掛かっていることになります。 統計情報の出力形式については、**4-1-12**、**4-1-13**を、READ単価、WRITE単価の求め方については、**4-1-14**を参照してください。
	pdload実行時のチューニング情報 (pdloadコマンドのreport文指定) ※09-50から省略値で $PDDIR/spool/utlrpt下に出力	設計にお問い合わせする際に、資料を送付してください。

データロード時に使用するバッファは、グローバルバッファがお勧めです。

4-1-11　チューニング情報の見方：デファードライト処理に関する統計情報

HITACHI Inspire the Next

解説　データベースの入出力に関する統計情報のDAT形式ファイルから、**4-1-10**に示した項目を確認してください。

デファードライト処理に関する統計情報のDAT形式ファイルのレコード形式

#	フィールド名（タイトルバー）	属性	最大長	備考
1	ホスト名(HOST)	文字	32	—
2	サーバ名(SERVER)		8	—
3	ログ取得時刻(LOG GET TIME)		19	-y省略時 ″MM/DD/hh:mm″形式(-e sec省略時) ″MM/DD/hh:mm:ss″形式(-e sec指定時) -y指定時 ″YYYY/MM/DD hh:mm:ss″形式
4	動作要因(CAUSE)		1	トリガ:″T″ プレシンク:″P″ シンクポイント:″S″ データベースのシンクポイント:″D″ RDエリアのシンクポイント:″R″
15	合計WRITE時間(DWSUM)	数値	10	単位:秒(秒値未満は切り捨て)
16	項番15のマイクロ秒(DWSUMM)		6	秒値を含みません。

4-1-12 チューニング情報の見方：データベースの入出力に関する統計情報（１）

HITACHI
Inspire the Next

解説

グローバルバッファ経由のデータベースへの入出力に関する情報（DAT形式ファイル）を出力します。**4-1-10**に示した項目を確認してください。
確認する項目とREAD単価、WRITE単価の求め方については、**4-1-14**に示します。

データベースの入出力に関する統計情報のDAT形式ファイルのレコード形式

#	フィールド名（タイトルバー）		属性	最大長	備考
1	ホスト名(HOST)		文字	32	—
2	サーバ名(SERVER)			8	—
3	入出力が発生したRDエリアの名称(RDAREA NAME)			30	—
4	項番3に示すRDエリアを構成するHiRDBファイルの通番(FILE NUMBER)		数値	2	—
5	統計ログを取得した時刻(LOG GET TIME)		文字	19	″YYYY/MM/DD hh:mm:ss″形式
6	HiRDBファイルのページ長(PAGE SIZE(K))			6	単位：キロバイト
7	取得時間間隔内に発生したread回数(READ CNT)				—
8	取得時間間隔内に発生したreadのうち、時間を計測した回数(READ MEASURE CNT)		数値	10	—
9	時間を計測したreadの時間のマイクロ秒(READ TIME(MICRO))	合計値(SUM)		12	秒値を含みます。
10		最大値(MAX)			

項番3と4の値でディクショナリ表SQL_PHYSICAL_FILESを検索することで、HiRDBファイルを特定できます。

```
SELECT PHYSICAL_FILE_NAME FROM MASTER.SQL_PHYSICAL_FILES
  WHERE RDAREA_NAME = '入出力が発生したRDエリアの名称(RDAREA NAME)の値' AND
  PHYSICAL_FILE_ID = RDエリアを構成するHiRDBファイルの通番(FILE NUMBER)
  WITHOUT LOCK NOWAIT;
```

4-1-13 チューニング情報の見方：データベースの入出力に関する統計情報（2）

HITACHI
Inspire the Next

データベースの入出力に関する統計情報のDAT形式ファイルのレコード形式

#	フィールド名（タイトルバー）		属性	最大長	備考
11	取得時間間隔内に発生したwrite回数(WRITE CNT)				—
12	取得時間間隔内に発生したwriteのうち、時間を計測した回数(WRITE MEASURE CNT)		数値	10	—
13	時間を計測したwriteの時間のマイクロ秒(WRITE TIME(MICRO))	合計値(SUM)		12	秒値を含みます。
14		最大値(MAX)			
15	ランク種別(RANK KIND)		文字	2	上位10件を選定した項目 read回数　:RC read合計時間 :RS read最大時間 :RM write回数　:WC write合計時間:WS write最大時間:WM 選定項目なし(*1):*

(*1) RDエリアを構成するHiRDBファイルの入出力情報をすべて出力する場合

性能への影響を最小限に抑えるために、省略値では以下のように動作します。

■情報を出力する時間間隔:60秒間隔　　■入出力時間の計測頻度:1秒毎に100回の入出力時間を計測します。

■統計情報の出力量

出力時間間隔内にアクセスのあったRDエリア構成ファイルの入出力情報のうち、統計情報として取得する以下の項目ごとに、それぞれのユニットで上位10ファイルの情報を統計ログに出力します。

・read最大回数　・read最大時間　・read合計時間　・write最大回数　・write最大時間　・write合計時間

4-1-14 データベースの入出力に関する統計情報の評価

HITACHI
Inspire the Next

解説 データベースの入出力に関する統計情報から、以下の計算式でREAD単価[us]およびWRITE単価[us]を求めます。

READ単価[us] ＝「READ TIME(MICRO) SUM」÷「READ MEASURE CNT」
WRITE単価[us] ＝「WRITE TIME(MICRO) SUM」÷「WRITE MEASURE CNT」

■WRITEが100,000[us]遅延している例

データベースの入出力に関する統計情報

この時間帯にWRITEが100,000[us]以上遅延していることがわかる。

LOG GET TIME		WRITE CNT	WRITE MEASURE CNT	WRITE TIME(MICRO) SUM
4/11/10 23:29:44		115	115	150034
4/11/10 23:29:44		114	114	441211
4/11/10 23:29:44		114	114	133806
:		:		
2014/11/10 23:30:44	…	87	87	8980550
2014/11/10 23:30:44		88	88	8938363
2014/11/10 23:30:44		87	87	8862967
:		:		
2014/11/10 23:32:35		113	113	2423749
2014/11/10 23:32:35		112	112	444391
2014/11/10 23:32:35		109	109	944923
:		:		

計算

計算結果を可視化したもの。

WRITE単価[us]
1,304.643
3,870.272
1,173.737
:
103,224.713
101,572.307
101,873.184
:
21,449.106
3,967.777
8,669.018
:

HITACHI
Inspire the Next

4. ユティリティのチューニング

4.1 データベース作成ユティリティ(pdload)のチューニング

4.2 データベース再編成ユティリティ(pdrorg)のチューニング

4-2-1　データベース再編成ユティリティ(pdrorg)の概要

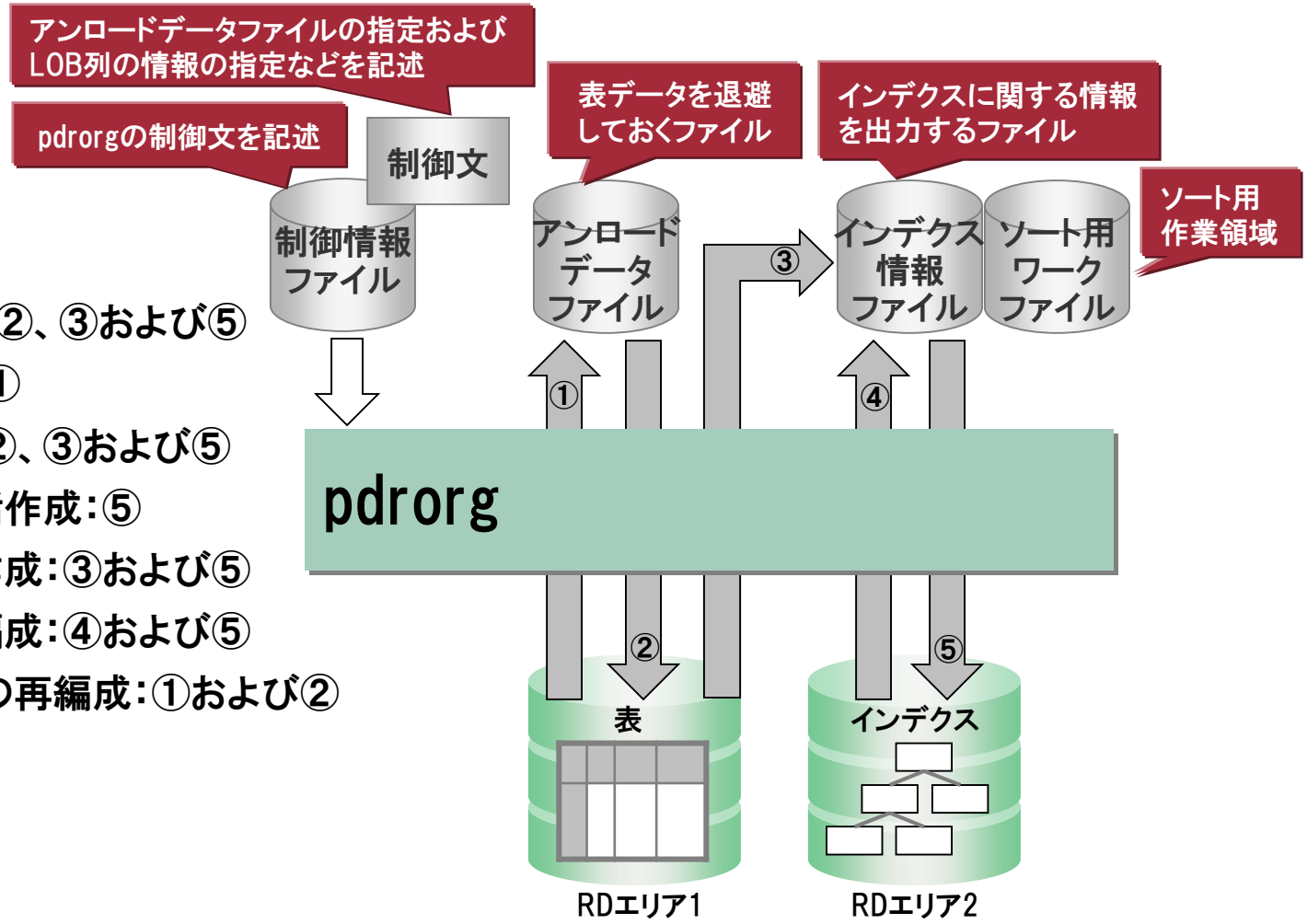
HITACHI
Inspire the Next

解説

データベース再編成ユティリティ(pdrorg)は、表およびインデクスのメンテナンスをするユティリティです。以下にpdrorgの機能と概要を示します。

pdrorgの機能

- □ 表の再編成:①、②、③および⑤
- □ 表のアンロード:①
- □ 表へのリロード:②、③および⑤
- □ インデクスの一括作成:⑤
- □ インデクスの再作成:③および⑤
- □ インデクスの再編成:④および⑤
- □ ディクショナリ表の再編成:①および②



4-2-2　pdrorgコマンド実行時の検討ポイント

HITACHI
Inspire the Next

解説　pdrorgコマンド実行時の検討ポイントに該当する場合は、対策を検討してください。

検討ポイント	対策
横分割表の 再編成の単位	1表/RDエリアなら、再編成時間の速い、RDエリア単位のパラレル再編成(-r RDエリア名)がお勧めです。 上記以外の場合は、表単位で実行するなら、再編成時間の速い、サーバ単位再編成(unload文のサーバ名指定)がお勧めです。 再編成に時間がかかる場合は、表単位など範囲を絞るように運用を変更してください。それでも時間内に収まらなければ、空きページ解放(pdreclaimコマンド)を検討してください。
＜Windowsの場合＞ データ量が多い場合	データ量が数百ギガバイト程度の場合は、データベース作成ユティリティ(pdload)と同様です。「4-1-3　pdloadコマンド実行時の検討ポイント(2)の検討ポイントが入力データファイルの大きさの欄」を参照してください。
ログ取得方式(注1)	大量データで実行時間を少しでも速くしたい場合、1表／RDエリアなら、ログレスモード(-l n)にすることで、実行時間を速くできます。 上記以外の場合は、ログ取得モード(-l a または -l p)を指定して、ログを取得してください。
再編成時の順番指定	アンロードするデータの順番指定(-b)は、表のアンロード時には有効ですが、表の再編成では意味がなく、実行時間もかかるため、表の再編成時には指定しないでください。

注1：ログ取得モードに-l nまたは-l pを指定時は、バックアップ運用に応じて、pdrorgコマンド実行前後でのバックアップの取得要否を検討してください。

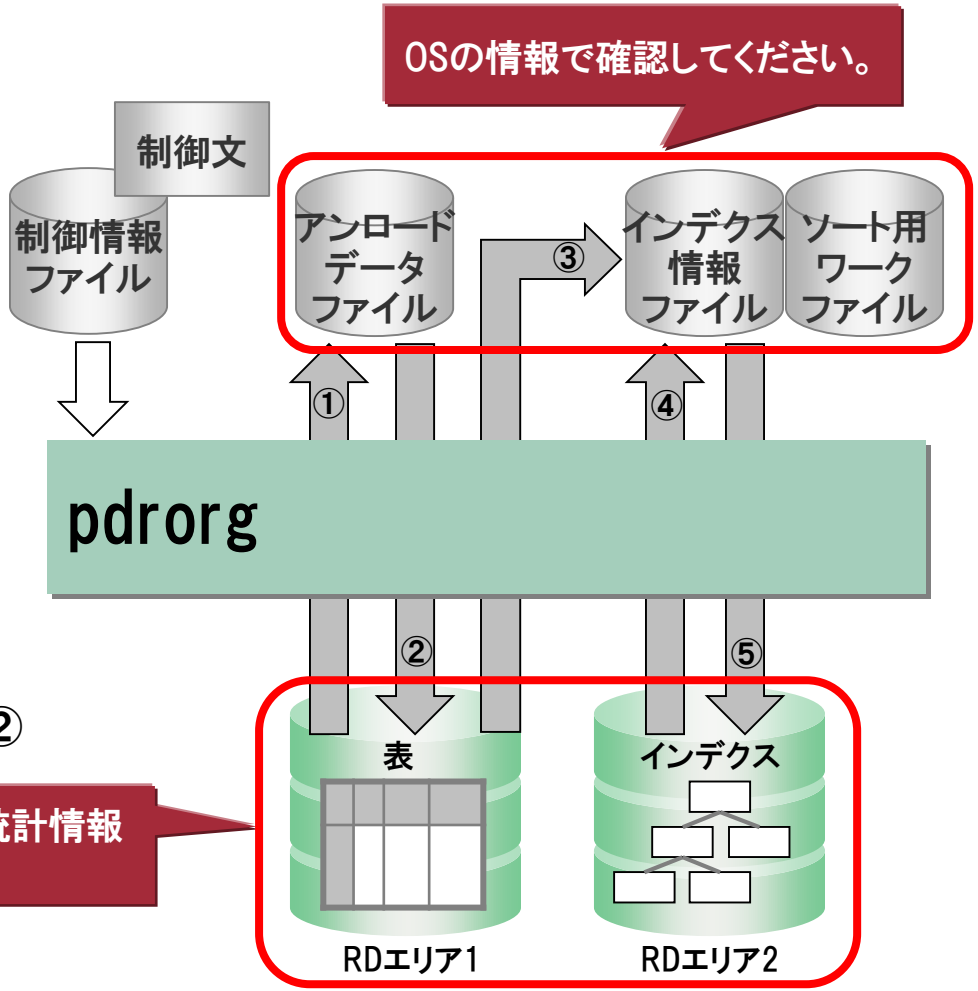
4-2-3 pdrorgで性能遅延が発生する場合

HITACHI
Inspire the Next

解説 pdrorgで性能遅延が発生する場合、入力データファイルのI/O性能かディスクI/O性能が原因の場合が多いです。OSの情報やHiRDBの統計情報で切りわけを行います。

pdrorgの機能

- □ 表の再編成:①、②、③および⑤
- □ 表のアンロード:①
- □ 表へのリロード:②、③および⑤
- □ インデクスの一括作成:⑤
- □ インデクスの再作成:③および⑤
- □ インデクスの再編成:④および⑤
- □ ディクショナリ表の再編成:①および②



4-2-4　確認するチューニング情報とチューニング情報の見方

HITACHI
Inspire the Next

解説　確認するチューニング情報とチューニング情報の見方については、データベース作成ユティリティ(pdload)と同様です。

４．１節 「データベース作成ユティリティ(pdload)のチューニング」を参照してください。

HITACHI
Inspire the Next

5. 各種バッファのチューニング

HITACHI
Inspire the Next

5. 各種バッファのチューニング

5.1 各種バッファの概要

5.2 チューニング

5.3 確認するチューニング情報

5.4 チューニング情報の見方

5-1-1　はじめに

本章では、影響の大きい以下のバッファのチューニングについて解説します。

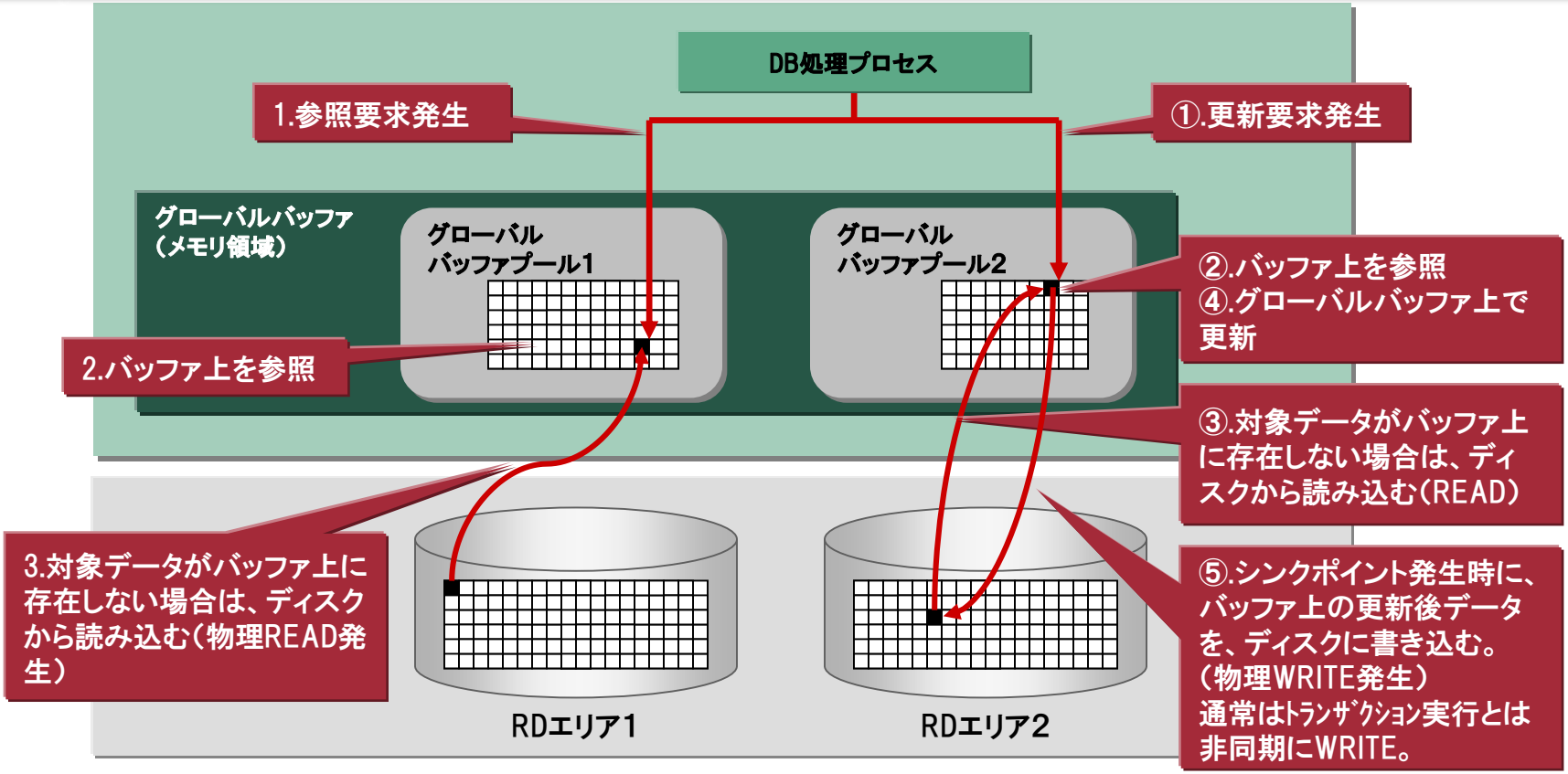
- グローバルバッファ
- SQLオブジェクト用バッファ
- 表定義情報用バッファ
- ビュー解析情報用バッファ

5-1-2　グローバルバッファとは

HITACHI
Inspire the Next

解説

グローバルバッファとは、RDエリアに格納されているデータの入出力に使用されるメモリ領域です。グローバルバッファは、バッファページ（面）という入出力の単位で構成されています。



対象データがバッファ上に存在しない割合が高い（ヒット率が低い）と、ディスクI/Oが増え性能に影響します。

5-1-3　グローバルバッファの割り当て(１)

HITACHI
Inspire the Next

解説　グローバルバッファのパフォーマンスを上げるために必要に応じて共通プール化、グループ化、専用バッファ化して下さい。

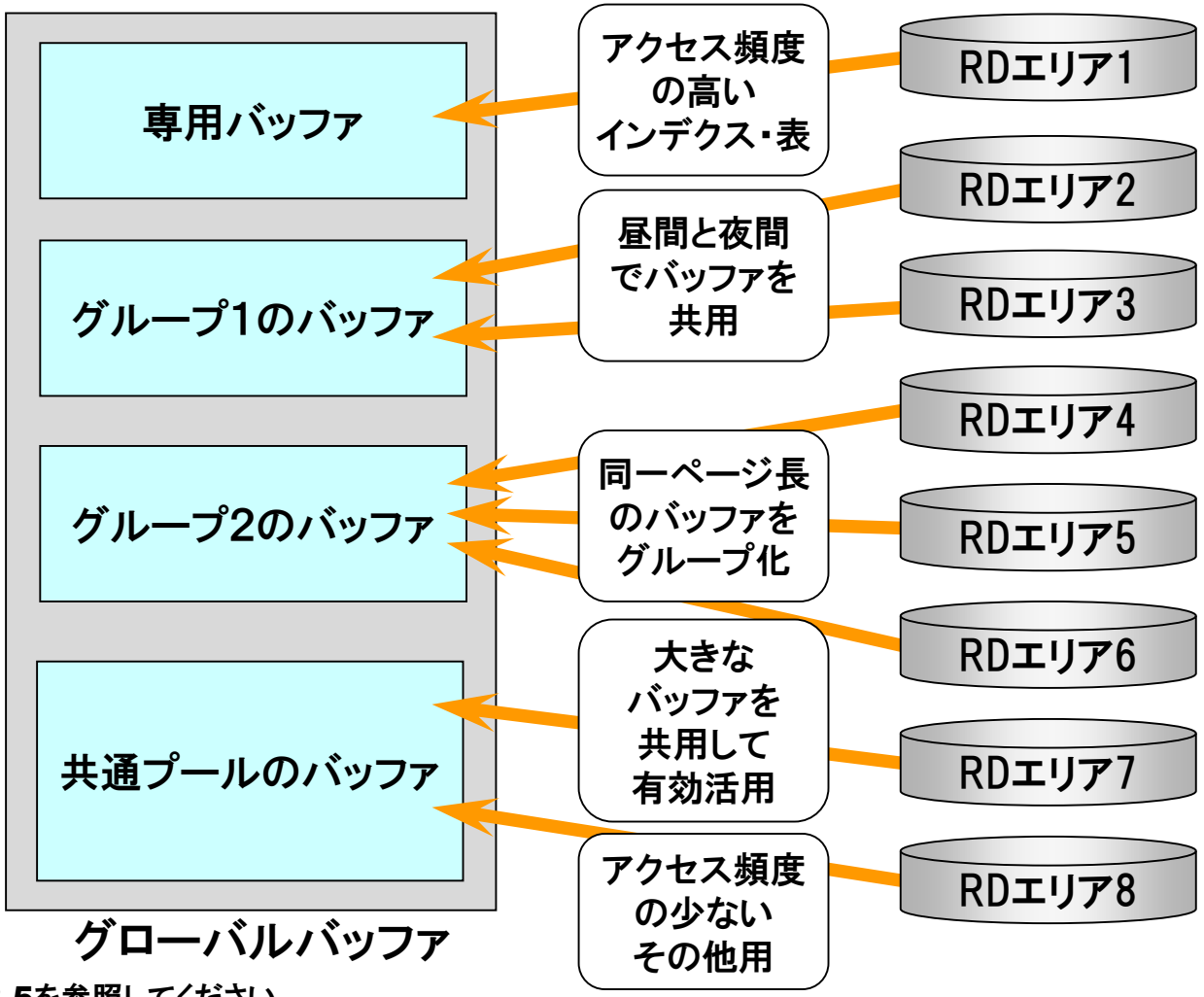
pdbuffer -r RDエリア1

pdbuffer -i インデクス名

pdbuffer
-r RDエリア2, RDエリア3

pdbuffer
-r RDエリア4, RDエリア5,
　RDエリア6

pdbuffer -o



pdbufferオペランドの詳細は、付録B-5を参照してください。

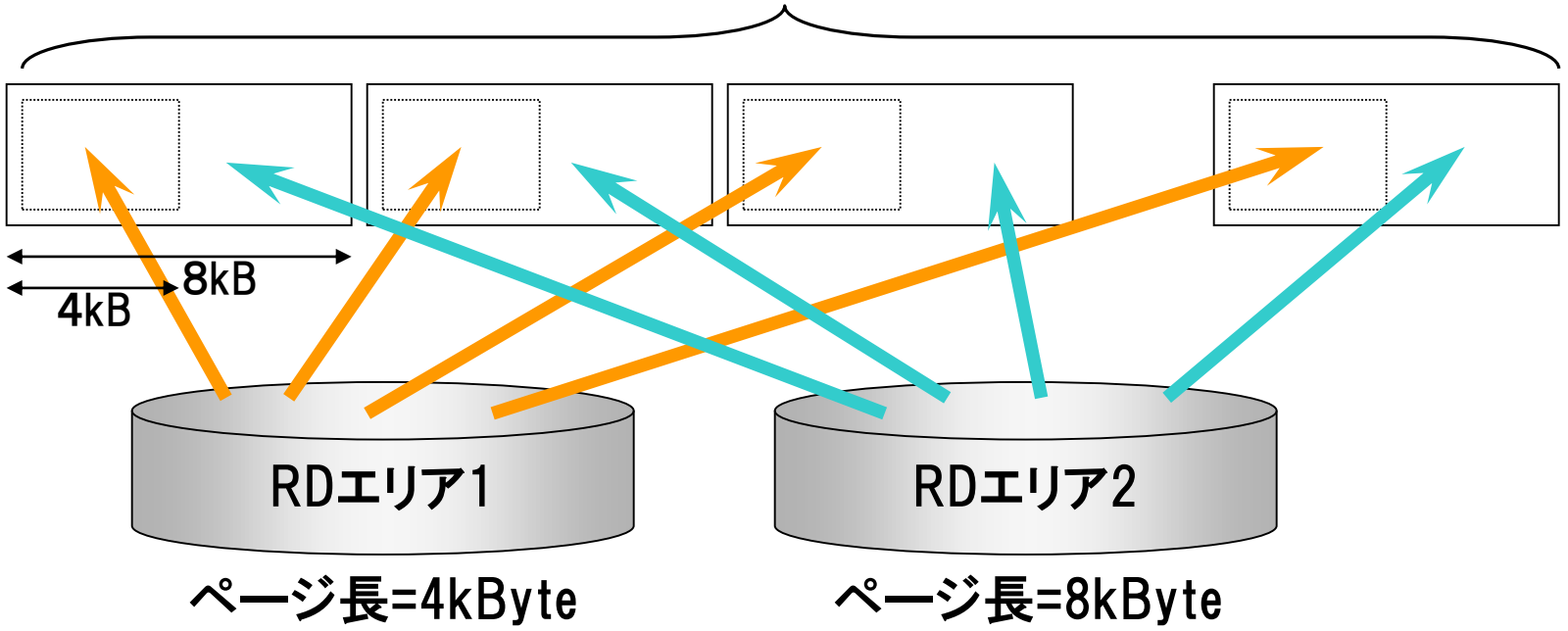
5-1-4 グローバルバッファの割り当て(２)

HITACHI
Inspire the Next

解説 RDエリア1から見ると、1面＝8kByteの前半4kByteしか使わないので、メモリの使用効率が悪い。 → ページ長が同じRDエリアをグループ化して下さい。

```
pdbuffer  -r RDエリア1,RDエリア2  -n  10
```

10面(1面の大きさは割り当てたRDエリアの最大ページ長となる)



5-1-5 グローバルバッファの面数の考え方(１)

HITACHI
Inspire the Next

解説 ユーザ用RDエリアへ割り当てるグローバルバッファの面数について、共通の考え方を説明します。

グローバルバッファの割り当て面数が、アクセスするページ数に比べ少なければ、
バッファヒット率が低下してI/O回数は増加しSQLのレスポンスは長くなります。
以下の考えで、性能要件を満たせるI/O回数となるように面数を割当ててください。

■性能要件に応じた割り当て面数の調整

グローバルバッファの割り当ては、データの特徴や業務性能、データのアクセス方法によって調整します。調整する場合の考え方を示します。

〇インデクスに割り当てるグローバルバッファの面数を十分に確保します。
インデクス検索では、インデクスページをサーチ、ヒットしたキーのあるデータページにアクセスするため、インデクスページのサーチが効率よく行えるようにするためです。

〇アクセス頻度が高い表を格納したユーザ用RDエリアのデータのヒット率が上がるように、割り当てるグローバルバッファの面数を多くします。

〇アクセス頻度が低い表、またはアクセス時間が長くても問題がない表を格納したユーザ用RDエリアに割り当てるグローバルバッファの面数を少なくします。

上記の考えを持つことで、リソースを有効利用しつつ、ヒット率を高められます。
なお、HiRDB技術資料「業務別 HiRDB設計のコツ」には、業務によるデータ操作の特徴と各種設計の考え方が示されていますので、参考としてください。

■実機による確認

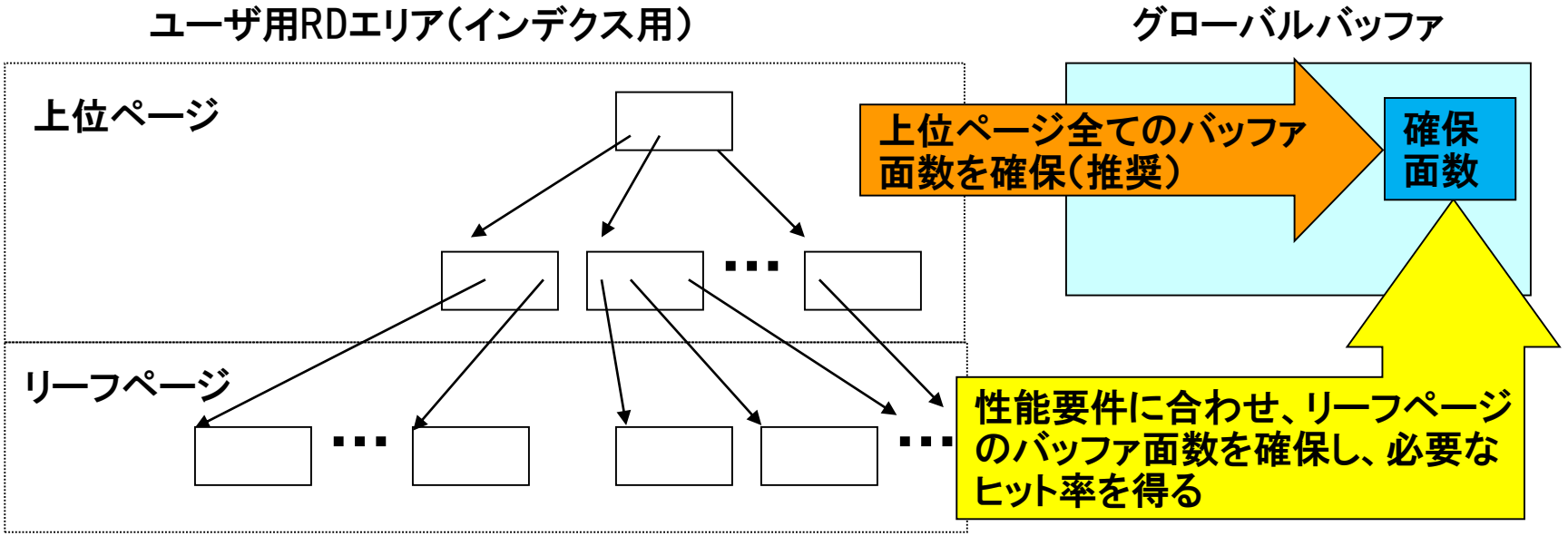
最終的にはテスト環境でバッファヒット率をモニタリングし、適切な面数を決定します。

5-1-6 グローバルバッファの面数の考え方(２)

解説 インデクスを格納しているユーザ用RDエリアへ割り当てるグローバルバッファの面数の考え方について説明します。

■インデクスを格納しているユーザ用RDエリアのバッファの面数の考え方

インデクス用バッファは、インデクスの全上位ページ分の面数を確実に確保し、さらにレスポンス要件からリーフページの面数を加えて、ヒット率が必要以上になるよう設計してください。
また、参照頻度の高いものを優先して面数を増やすことで、全体の性能が安定します。



インデクスの格納ページ数につきましては、HiRDBマニュアル「システム導入・設計ガイド」－「ユーザ用RDエリアの容量の見積もり」－「インデクスの格納ページ数の計算方法」を参照してください。漸化式のP1以外の合計が上位ページ数です。インデクスページスプリットの発生によっては、最大２倍の容量が必要になる点にご留意ください。

5-1-7　グローバルバッファの面数の考え方(３)

HITACHI
Inspire the Next

解説　要件が決まっていないなどバッファ面数が不明の場合、表やインデクスを格納するユーザ用RDエリアへ割り当てるグローバルバッファの面(ページ)数の考え方について説明します。

経験上の最低限必要なバッファ面数の目安です。お勧めの値ではありません。
バッファ面数を決める要件が決定した以降にチューニングを実施してください。

■バッファ面数が不明な表を格納しているユーザ用RDエリア
表を格納しているユーザ用RDエリアのページ数の1%を目安として
グローバルバッファの面数を割り当てます。

表を格納しているユーザ用RDエリアのグローバルバッファ面数
＝ユーザ用RDエリア(表用)のページ数×0.01

【例】 ユーザ用RDエリアの格納ページ数100000ページの場合
表を格納しているユーザ用RDエリアのバッファ面数
＝100000ページ×0.01＝1000面

■バッファ面数が不明なインデクスを格納しているユーザ用RDエリア
インデクスを格納しているユーザ用RDエリアのページ数の15%を目安として
グローバルバッファの面数を見積ります 。

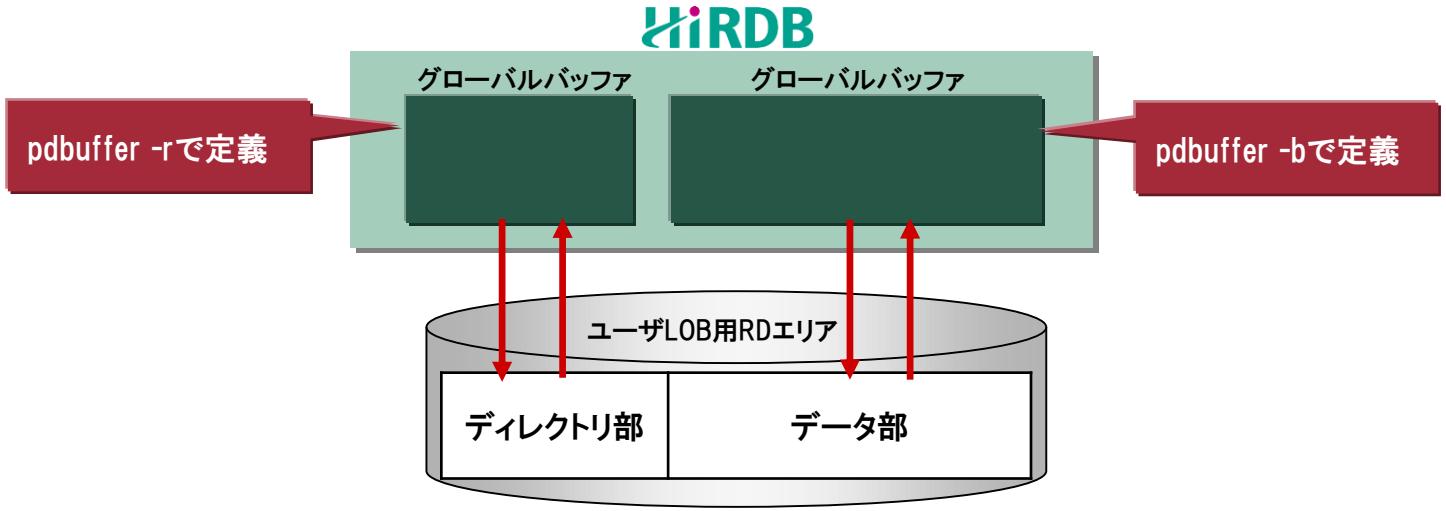
インデクスを格納しているユーザ用RDエリアのグローバルバッファ面数
＝ユーザ用RDエリア(インデクス用)のページ数(*1) ×0.15

(*1):見積れない場合は、「ユーザ用RDエリア(表用)のページ数×0.1」としてください。

5-1-8 ユーザLOB用RDエリアのグローバルバッファ

HITACHI
Inspire the Next

解説 LOB用RDエリアにはディレクトリ部とデータ部があります。ディレクトリ部とデータ部は、それぞれ異なるグローバルバッファで管理します。



ディレクトリ部とデータ部のグローバルバッファは異なる用途のグローバルバッファのため、別々に見積もる必要があります。

■ ディレクトリ部
ディレクトリ部は100%グローバルバッファにヒットすることが望ましいです。
ディレクトリ部のグローバルバッファ容量は、HiRDBマニュアル「システム導入・設計ガイド」ー「ユーザLOB用RDエリアの容量の見積もり」ー「ディレクトリページ部分の総ページ数」で見積もった値以上としてください。

■ データ部
アクセス頻度が多い場合は、グローバルバッファに乗せたほうが良いです。
データ部分のグローバルバッファ容量は、性能要件に合わせて見積もってください。

5-1-9 グローバルバッファ割り当て時の注意事項

HITACHI
Inspire the Next

解説

グローバルバッファ全体の面数が、バッファページの同時アクセス要求数に満たない場合、バッファページ不足によるSQLエラーが発生するため、注意が必要です。

以下の式を満たさない場合、バッファページ不足によりSQLエラーとなることがあります。式を満たすことをご確認ください(LOB用グローバルバッファは除く)。

バッファ面数
　≧ MIN(グローバルバッファに割り当てるRDエリアの全ページ(※1)数,
　　最大同時接続数(※2) ×4面)

※1 ディレクトリページを含みます
※2 システム共通定義pd_max_usersの値、または
　サーバ定義のpd_max_bes_processの値を目安とします

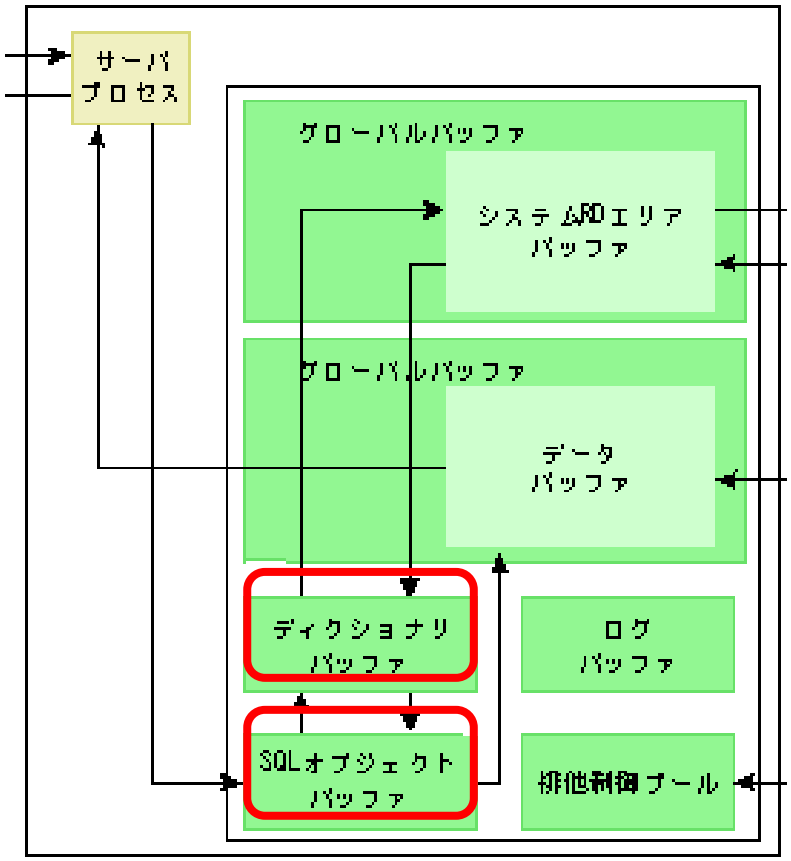
バッファページ不足発生時のエラーメッセージ

KFPA11919-E Insufficient global buffer, global buffer pool=グローバルバッファプール名称

5-1-10 SQLオブジェクト用バッファ、ディクショナリバッファ

HITACHI
Inspire the Next

解説 SQLオブジェクト用バッファ、ディクショナリバッファ(表定義情報用バッファ、ビュー解析情報用バッファ)の概要について以下に解説します。



バッファ種別	概要
SQLオブジェクト用バッファ	解析したSQLオブジェクト(*1)を格納するバッファです。
表定義情報用バッファ	一度使用した表の定義情報を格納し、ディクショナリバッファ内に確保されるバッファです。この表の定義情報は、SQL文の解析時に使用されます。
ビュー解析情報用バッファ	一度使用したビューの定義情報を格納し、ディクショナリバッファ内に確保されるバッファです。

(*1):HiRDBが解析および最適化したSQLの実行手順。

バッファ内に格納されている定義情報を使用すると、ディスクI/Oが発生しないため、性能が向上します。

HITACHI
Inspire the Next

5. 各種バッファのチューニング

5.1 各種バッファの概要

5.2 チューニング

5.3 確認するチューニング情報

5.4 チューニング情報の見方

5-2-1　グローバルバッファのチューニング

HITACHI
Inspire the Next

解説　取得したチューニング情報から以下の主な評価ポイントを検証して、該当する場合は対策してください。

評価ポイント	対策
グローバルバッファ全体のヒット率が80%未満で更新バッファフラッシュ回数、参照バッファフラッシュ回数がGET回数(*1)に対して大きい。	バッファから追い出されないように、グローバルバッファのバッファ面数(pdbufferオペランドの-nオプションの値)を大きくしてください。 フラッシュ回数が少なければヒット率だけでは単純にサイズ不足とはいえませんので、フラッシュ回数も同時に評価します。
グローバルバッファプールの排他競合待ち発生率が10%以上	グローバルバッファのプール単位の競合です。 グローバルバッファプールへのアクセスが分散されるような以下の方法を検討してください。 ●表を横分割し、グローバルバッファを分割ごとに独立して割り当てる。 ●グローバルバッファに複数のRDエリアを割り当てている場合は、RDエリアごとに独立して割り当てる。 ●インデクスについてはインデクス専用のグローバルバッファ(pdbufferの-i指定)を割り当てる。
グローバルバッファ排他待ち発生回数がGET回数(*1)に対して大きい。	グローバルバッファのページ競合なので、ページアクセスが分散されるような以下の方法を検討してください。 ●表を横分割し、グローバルバッファを分割ごとに独立して割り当てる。 ●RDエリアのページサイズを小さくする。

(*1):更新GET回数＋参照GET回数

※　ここで紹介しているポイントがすべてではありません。日立アカデミーの「HiRDBデータベースチューニング」(HiRDB講座)で詳しく解説していますので、こちらの教育もぜひ受講ください。

5-2-2 その他のバッファのチューニング

HITACHI
Inspire the Next

解説 取得したチューニング情報から以下の主な評価ポイントを検証して、該当する場合は十分なサイズを割当て、オーバヘッドを削減します。

バッファ種別	評価ポイント	対策
SQLオブジェクト用バッファ	SQLオブジェクト用バッファから出されたSQLオブジェクトの数が多い	バッファから追い出されないように、SQLオブジェクト用バッファ(システム定義のpd_sql_object_cache_sizeオペランドの値)を大きくしてください。
表定義情報用バッファ	表定義情報用バッファヒット率(*1)が80%未満の場合	表定義情報用バッファ(システム定義のpd_table_def_cache_sizeオペランドの値)を大きくしてヒット率が80%を超えるようにしてください。
ビュー解析情報用バッファ	ビュー解析情報用バッファヒット率(*2)が80%未満の場合	ビュー解析情報用バッファ(システム定義のpd_view_def_cache_sizeオペランドの値)を大きくしてヒット率が80%を超えるようにしてください。

(*1):(表定義情報用バッファヒット回数／表定義情報取得要求回数)×100

(*2):(ビュー解析情報用バッファヒット回数／ビュー解析情報取得要求回数)×100

バージョン09-50の変更点

推奨モードでは、上記記載のバッファに関するオペランドの省略値を拡大しました。
これにより、バッファ満杯による性能劣化となる事象を低減でき、性能の安定化をはかれます。

5-2-3 その他の見直し観点

HITACHI
Inspire the Next

解説 その他に以下のよう見直し観点もあります。該当する場合は、対応するオペランドの指定値を見直してください。

見直し観点	説明	設定値を見直すオペランド
サイズの大きなバイナリデータにアクセスするUAPを実行する場合がある。	バイナリデータがグローバルバッファにキャッシュされると、キャッシュされた直近の内容がメモリから追い出され、性能が一時的に低下することがあります。	pd_dbbuff_binary_data_lru
実メモリに余裕があり、性能を重視したい。	グローバルバッファ用の共用メモリを実メモリに固定することで、ページの入出力が少なくなり、性能を安定させることができます。	pd_dbbuff_attribute

※ オペランドの詳細については、HiRDBマニュアル「システム定義」を参照してください。

HITACHI
Inspire the Next

5. 各種バッファのチューニング

5.1 各種バッファの概要

5.2 チューニング

5.3 確認するチューニング情報

5.4 チューニング情報の見方

5-3-1　確認するチューニング情報

HITACHI
Inspire the Next

解説　チューニング情報から以下の確認項目を確認してください。

バッファ種別	確認項目	取得するチューニング情報	確認項目の見方
グローバルバッファ	・グローバルバッファのヒット率 ・更新バッファフラッシュ回数 ・参照バッファフラッシュ回数 ・更新GET回数 ・参照GET回数 ・排他競合待ち発生率(統計bufのみ) ・排他待ち発生回数 ・出力待ち回数(統計bufのみ)	グローバルバッファの簡易統計情報	5-4-1参照
		グローバルバッファプールに関する統計情報	5-4-2参照
SQLオブジェクトバッファ	・SQLオブジェクト用バッファから出されたSQLオブジェクトの数 ・SQLオブジェクト用バッファヒット回数 ・SQLオブジェクト取得要求回数	システムの稼働に関する統計情報	5-4-3参照
表定義情報用バッファ	・表定義情報用バッファヒット回数 ・表定義情報取得要求回数		
ビュー解析情報用バッファ	・ビュー解析情報用バッファヒット回数 ・ビュー解析情報取得要求回数		

HITACHI
Inspire the Next

5. 各種バッファのチューニング

5.1 各種バッファの概要

5.2 チューニング

5.3 確認するチューニング情報

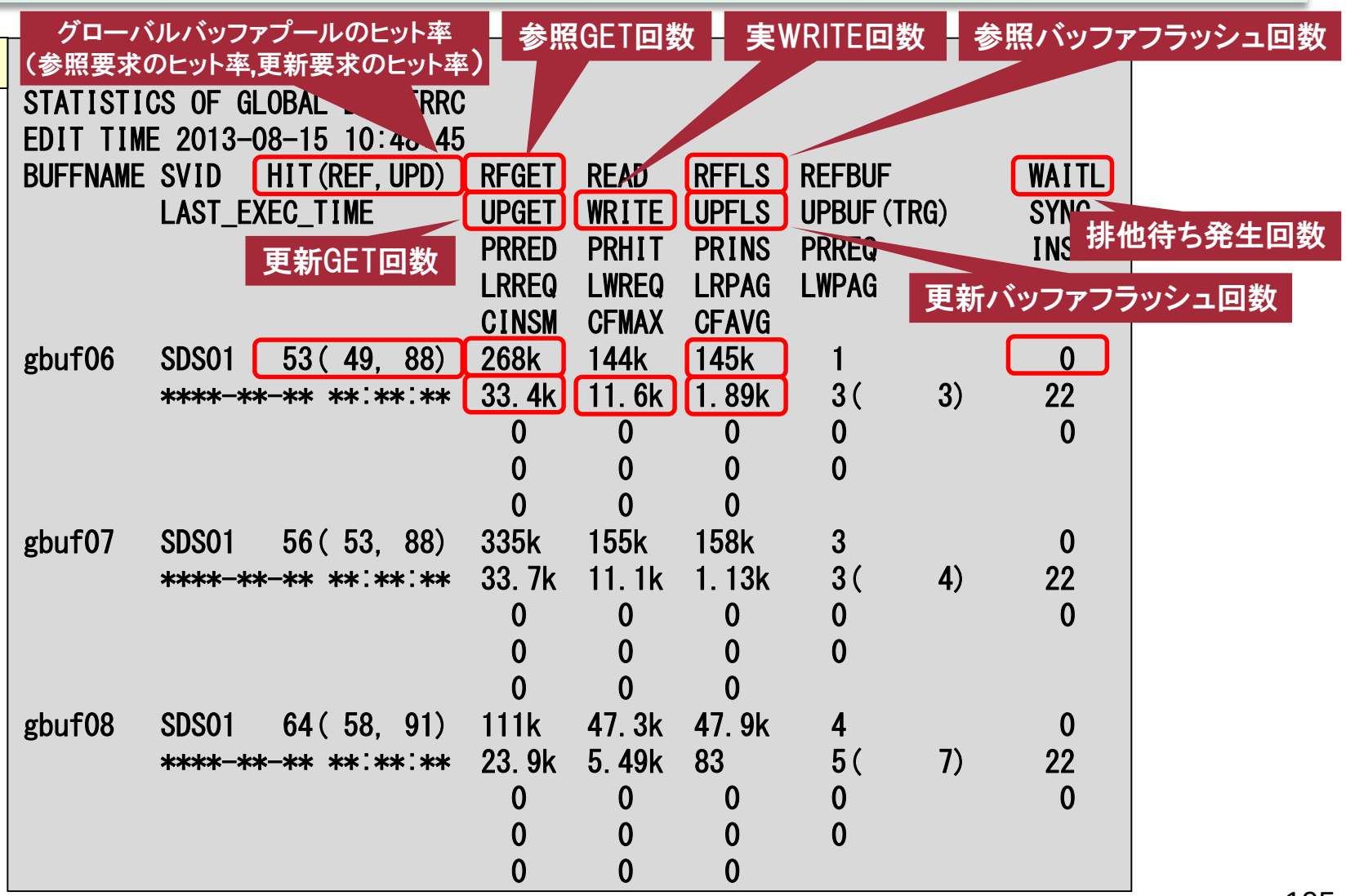
5.4 チューニング情報の見方

5-4-1　グローバルバッファプールのヒット率等の見方

HITACHI
Inspire the Next

解説　グローバルバッファの簡易統計情報から、**5-3-1**に示した確認項目を確認してください。

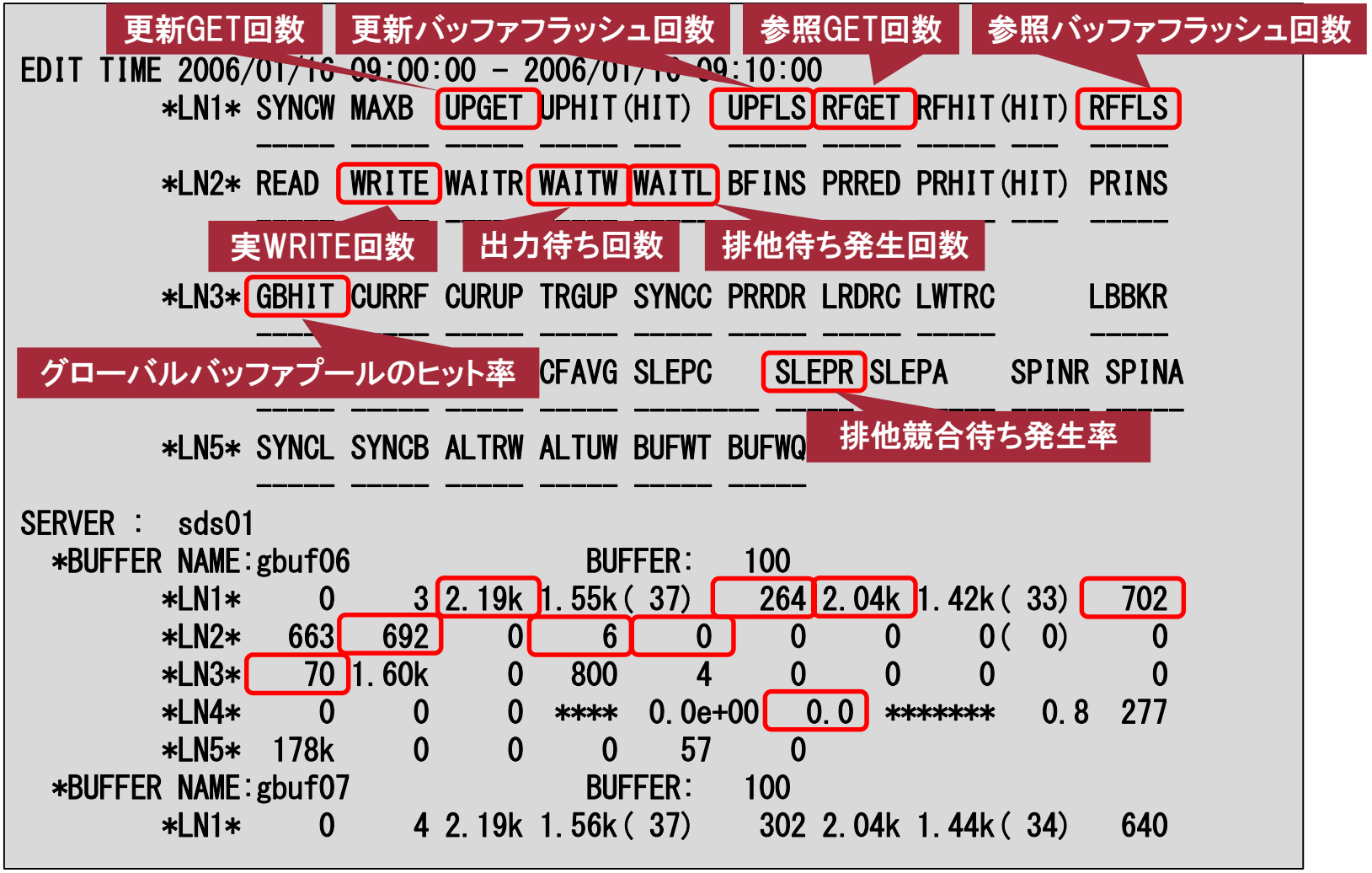
pdbufls




```
STATISTICS OF GLOBAL [illegible]RRC
EDIT TIME 2013-08-15 10:4[illegible]45
BUFFNAME SVID   HIT(REF,UPD)   RFGET   READ    RFFLS   REFBUF          WAITL
         LAST_EXEC_TIME         UPGET   WRITE   UPFLS   UPBUF(TRG)      SYNC
                                PRRED   PRHIT   PRINS   PRREQ           INS
                                LRREQ   LWREQ   LRPAG   LWPAG
                                CINSM   CFMAX   CFAVG
gbuf06   SDS01   53( 49, 88)   268k    144k    145k    1                 0
         ****-**-** **:**:**   33.4k   11.6k   1.89k   3(    3)          22
                                0       0       0       0               0
                                0       0       0       0
                                0       0       0
gbuf07   SDS01   56( 53, 88)   335k    155k    158k    3                 0
         ****-**-** **:**:**   33.7k   11.1k   1.13k   3(    4)          22
                                0       0       0       0               0
                                0       0       0       0
                                0       0       0
gbuf08   SDS01   64( 58, 91)   111k    47.3k   47.9k   4                 0
         ****-**-** **:**:**   23.9k   5.49k   83      5(    7)          22
                                0       0       0       0               0
                                0       0       0       0
                                0       0       0
```

5-4-2 グローバルバッファプールのヒット率等の見方

解説 グローバルバッファプールに関する統計情報から、**5-3-1**に示した確認項目を確認してください。




```
EDIT TIME 2006/01/16 09:00:00 - 2006/01/16 09:10:00
         *LN1* SYNCW MAXB  UPGET UPHIT(HIT)  UPFLS RFGET RFHIT(HIT) RFFLS
               ----- ----- ----- ---  ----- ----- ----- ---  -----
         *LN2* READ  WRITE WAITR WAITW WAITL BFINS PRRED PRHIT(HIT) PRINS
               ----- ----- ----- ----- ----- ----- ----- ---  -----
         *LN3* GBHIT CURRF CURUP TRGUP SYNCC PRRDR LRDRC LWTRC      LBBKR
               ----- ----- ----- ----- ----- ----- ----- -----      -----
               ----- ----- ----- ----- -------- ----- ----- ----- CFAVG SLEPC    SLEPR SLEPA    SPINR SPINA
         *LN5* SYNCL SYNCB ALTRW ALTUW BUFWT BUFWQ
               ----- ----- ----- ----- ----- -----
SERVER :  sds01
  *BUFFER NAME:gbuf06                BUFFER:   100
         *LN1*     0     3 2.19k 1.55k( 37)   264 2.04k 1.42k( 33)   702
         *LN2*   663   692     0     6     0     0     0     0(  0)     0
         *LN3*    70 1.60k     0   800     4     0     0     0          0
         *LN4*     0     0     0  **** 0.0e+00   0.0 *******   0.8   277
         *LN5*  178k     0     0     0    57     0
  *BUFFER NAME:gbuf07                BUFFER:   100
         *LN1*     0     4 2.19k 1.56k( 37)   302 2.04k 1.44k( 34)   640
```

5-4-3　バッファの取得要求回数、バッファヒット回数等の見方

HITACHI
Inspire the Next

解説　システムの稼働に関する統計情報から、**5-3-1**に示した確認項目を確認してください。

```
EDIT TIME 2012/02/03 09:00:00 - 2012/02/03 10:00:00
SERVER : ********
                                                FREQ     MAX     MIN     AVG
                                                -----    -----   -----   -----
                    :
<DICTIONARY>        # OF TBL-DEF GET REQ          295
                    # OF TBL-CACHE HIT            272
                    # OF CACHED TBL-DEF            27      12       1       6
                    USED TBL-DEF SIZE              23   26.0k   5.60k   9.95k
                    TBL-CACHE SIZE                 27    118k   26.0k   79.1k
                    :
                    # OF VIEW DEF GET REQ           0
                    # OF VIEW CACHE HIT             0
                    # OF VIEW CACHED DEF            0
                    USED VIEW SIZE                 0       0       0       0
                    VIEW CACHE SIZE                0       0       0       0
                    :
<FES-BES-DIC(SDS)   # OF SQLOBJ INFO GET          240
 INFORMATION>       # OF CACHE HIT (SQLOBJ)       41
                    # OF CACHED SQLOBJ            203      65       1      27
                    CACHED SQLOBJ TOTAL SIZE      203    277k   5.00k    137k
                    # OF SWAP OUT SQLOBJ           36
                    REQUEST SQLOBJ SIZE           163
```

- # OF TBL-DEF GET REQ：表定義情報取得要求回数
- # OF TBL-CACHE HIT：表定義情報用バッファヒット回数
- # OF VIEW DEF GET REQ：ビュー解析情報取得要求回数
- # OF VIEW CACHE HIT：ビュー解析情報用バッファヒット回数
- # OF SQLOBJ INFO GET：SQLオブジェクト取得要求回数
- # OF CACHE HIT (SQLOBJ)：SQLオブジェクト用バッファヒット回数
- # OF SWAP OUT SQLOBJ：SQLオブジェクト用バッファから出されたSQLオブジェクトの数

HITACHI
Inspire the Next

付録A. DB排他制御のチューニング

HITACHI
Inspire the Next

付録A. DB排他制御のチューニング

A.1 排他制御の概要

A.2 デッドロック回避策

A.3 確認するチューニング情報

A.4 チューニング情報の見方

付録A-1-1　はじめに

ここでは、DB排他制御のチューニングの内、特にデッドロックについて解説します。

デッドロックが発生した場合の回避策を示します。これを設計段階で考慮することで、デッドロック発生確率を減らすことができます。

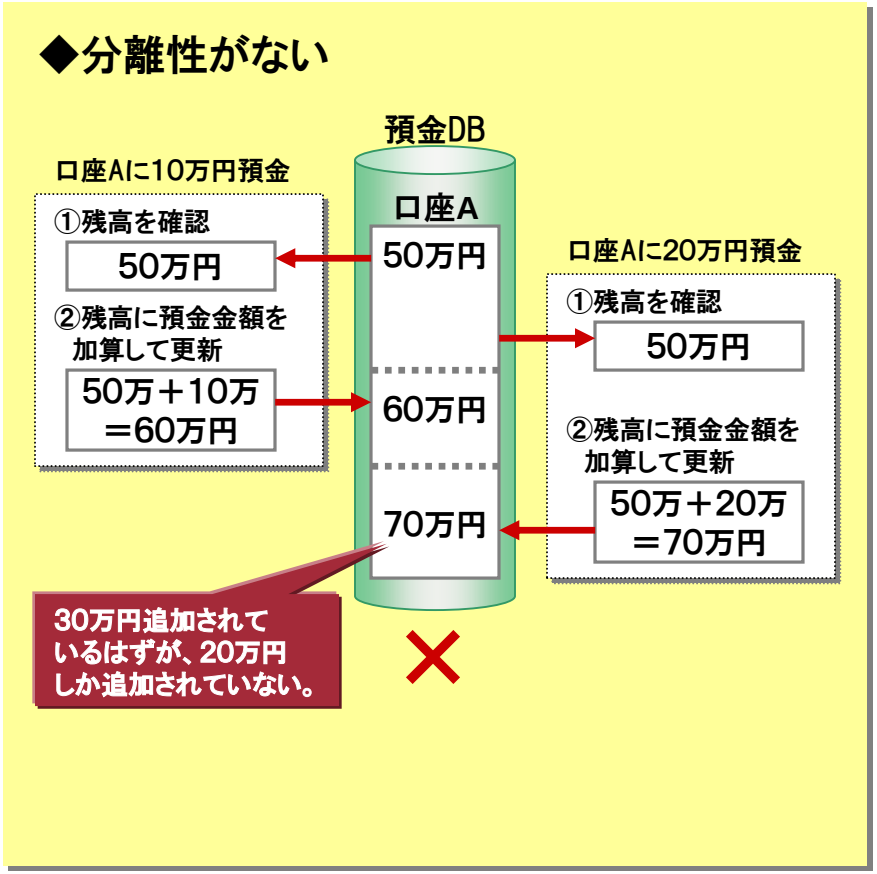
付録A-1-2　排他制御の必要性

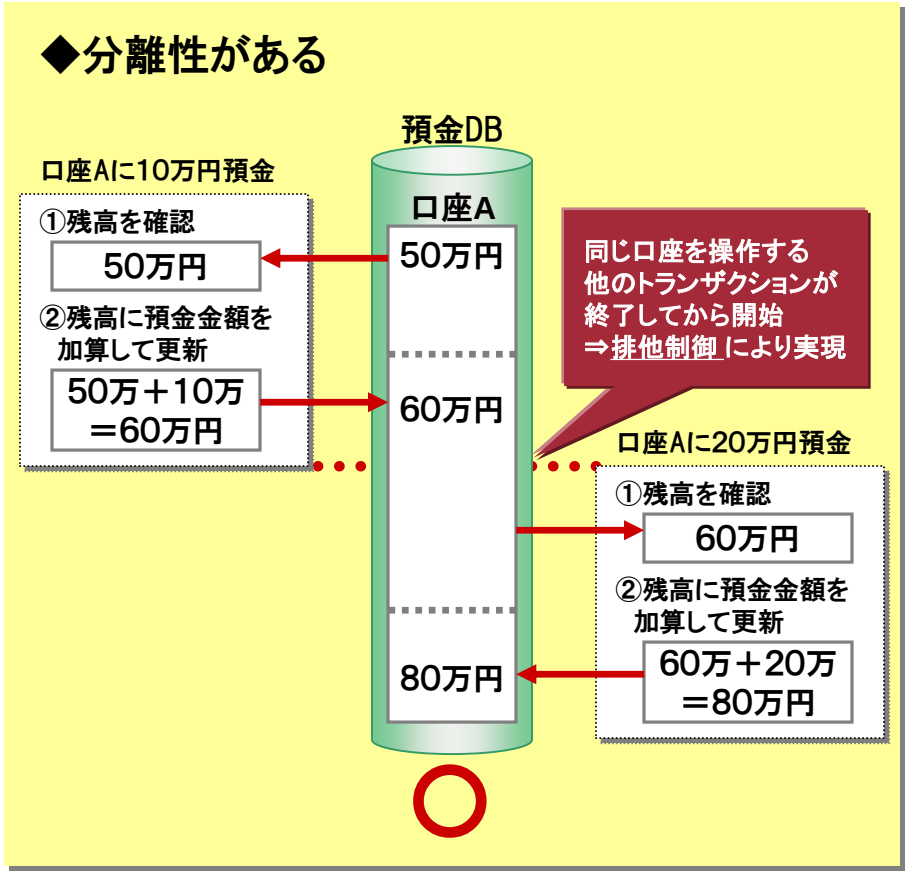
HITACHI
Inspire the Next

解説

複数のトランザクションが並行実行された場合でも、トランザクションは、他のトランザクションの前もしくは後に実行されたように見え、またがって実行されたように見えることはない、という特性のことを、トランザクションの分離性といいます。この特性は、排他制御により実現します。

■同じ口座に複数の預金操作を同時実行した場合の例





付録A-1-3　排他制御とは

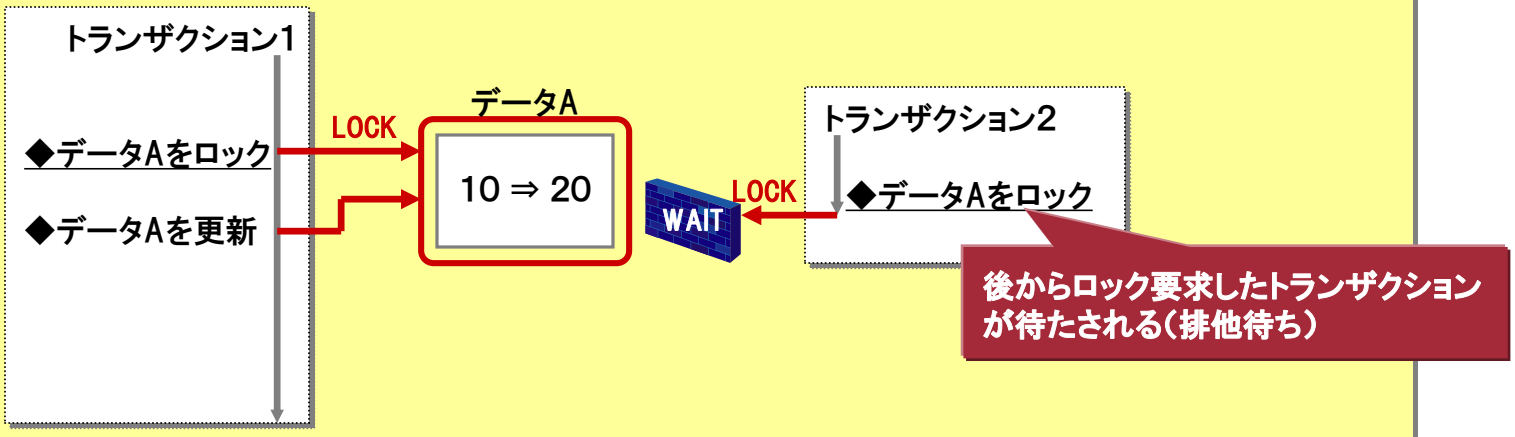
HITACHI
Inspire the Next

解説

ロック方式を用いた同時実行性制御を、排他制御といいます。
排他制御は、トランザクションの分離性を実現する最も一般的な方式です。

■ロック方式

トランザクション内でアクセスする資源にロックをかけて、他のトランザクションからのアクセスを待たせる方式。



注意事項

排他待ちが多発するとトランザクション性能が劣化します。

付録A-1-4　排他制御のモード

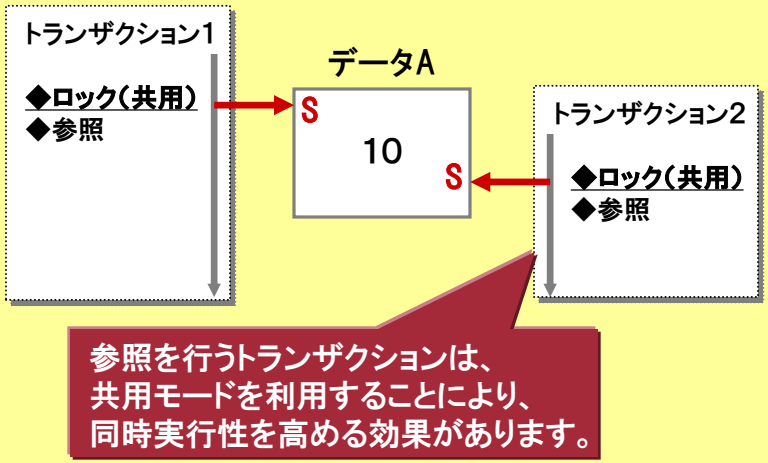
HITACHI
Inspire the Next

解説

ロックには、共用(shared)モードと排他(exclusive)モードという、2つのモードがあります。
排他モードでロックした場合、その資源に対する他のロック要求は許可されません。
これに対して、共用モードでロックした場合、その資源に対する共用モード同士のロック要求は許可されます。

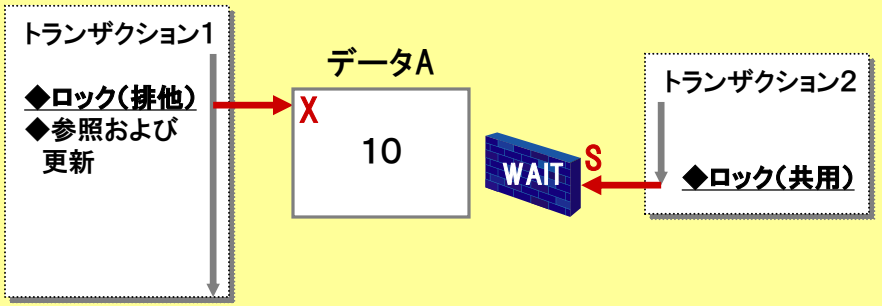
◆共用(shared)モード

そのトランザクションの読み込みを保護する目的で使用する



◆排他(exclusive)モード

そのトランザクションの読み書きを保護する目的で使用する



＜図凡例＞
S:共用モードのロック
X:排他モードのロック

＜表凡例＞
○:ロック要求が許可される
×:ロックが競合する(待たされる)

■ロックの競合関係

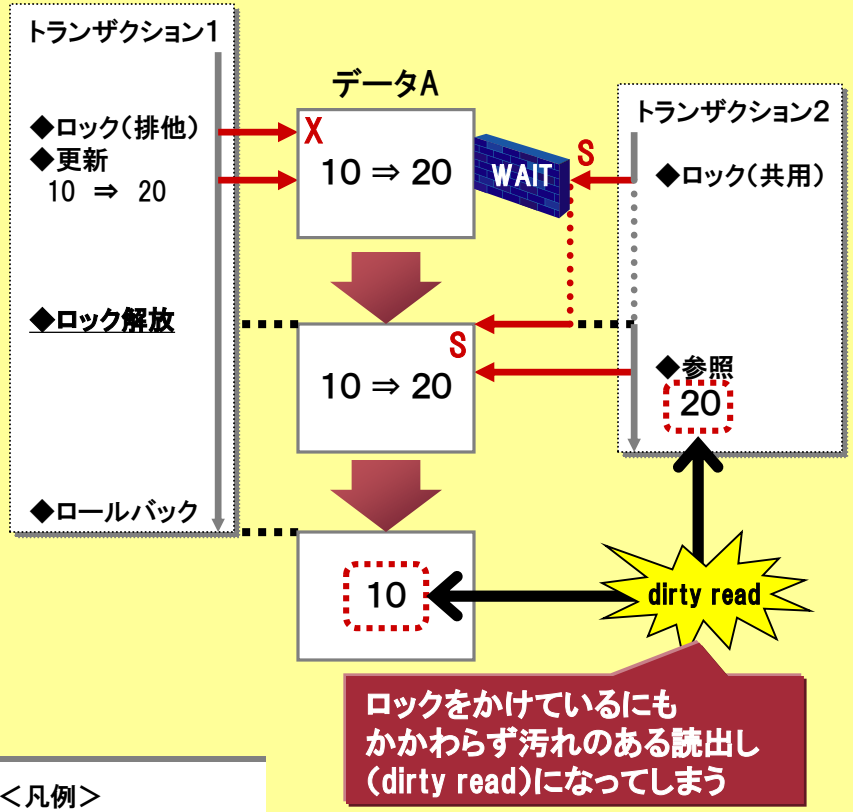
		他のトランザクションが先に保持しているモード		
		なし	共用(S)	排他(X)
要求するモード	共用(S)	○	○	×
	排他(X)	○	×	×

付録A-1-5　排他制御の期間

HITACHI
Inspire the Next

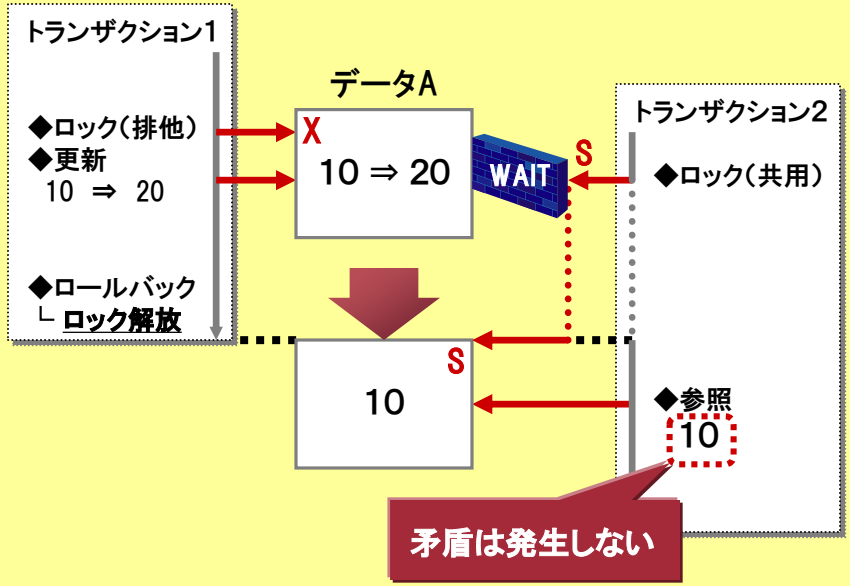
解説　トランザクションの分離性を実現するためには、トランザクション内でかけたロックは、トランザクションの終了まで保持します。

◆トランザクション終了まで保持しない場合



＜凡例＞
S:共用モードのロック
X:排他モードのロック

◆トランザクション終了まで保持する場合



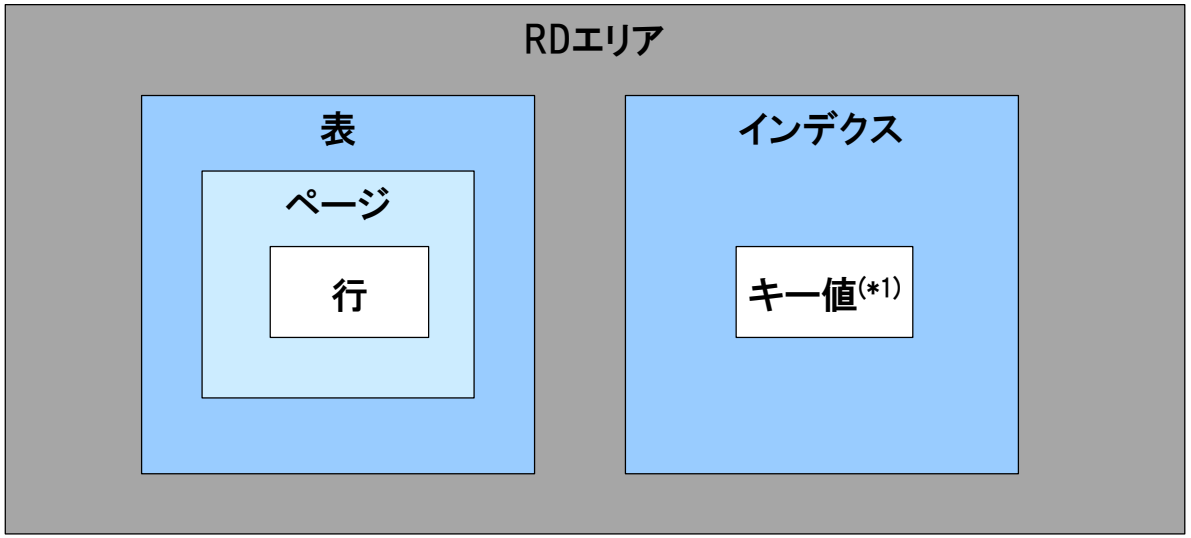
ロックの解放は、コミットもしくは
ロールバックの延長で実行します。

付録A-1-6　排他資源

HITACHI
Inspire the Next

解説　代表的な排他資源には以下のものがあります。

排他を掛ける対象を排他資源といいます。排他資源には、図のような包含関係があるため、上位の資源に排他を掛けると、それより下位の資源には排他を掛ける必要がなくなります。



(*1):デフォルトでは、インデクスキー値無排他が適用(pd_indexlock_mode =NONE)されるため、
キー値には排他を掛けません。

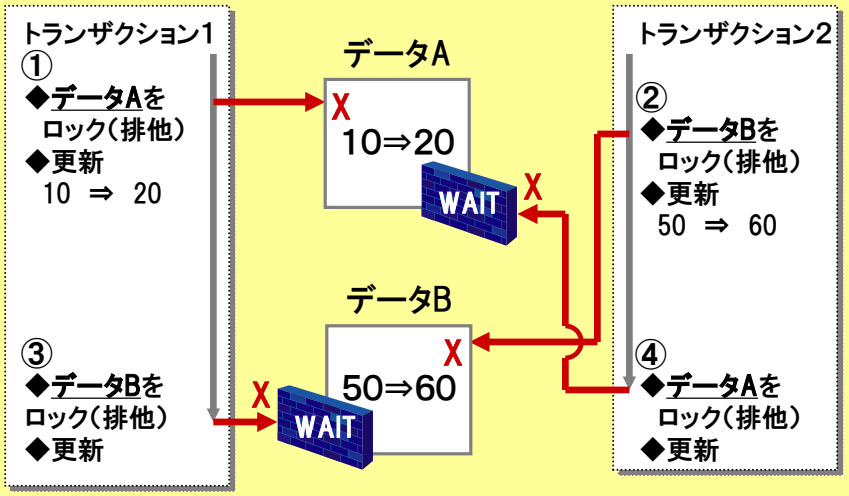
付録A-1-7　デッドロックの仕組み

HITACHI
Inspire the Next

解説

複数のトランザクションが、互いに保持しているロック資源の解放を待っており、
どちらかがロールバックしないと先に進めない状態のことをデッドロックといいます。

■デッドロックの例



＜凡例＞
S：共用モードのロック
X：排他モードのロック

トランザクションごとにアクセスするデータ順序が
異なる場合、デッドロックが発生する可能性があります。

①データAを更新するため、排他モードで排他を掛ける

②データBを更新するため、排他モードで排他を掛ける

③データBを更新するため、排他モードで排他を掛けようとしたが、
排他モードの排他が掛かっているため待ち状態

④データAを更新するため、排他モードで排他を掛けようとしたが、
排他モードの排他が掛かっているため待ち状態

HITACHI
Inspire the Next

付録A. DB排他制御のチューニング

A.1 排他制御の概要

A.2 デッドロック回避策

A.3 確認するチューニング情報

A.4 チューニング情報の見方

付録A-2-1　デッドロック回避策

HITACHI
Inspire the Next

解説　デッドロックが発生した場合、以下の回避策を検討します。
また、デッドロックを発生させないために、以下のような対策をしておくことが重要です。

#	回避策	適用効果
1	無排他検索(WITHOUT LOCK NOWAIT)を適用できないか検討する。	検索⇔更新間のデッドロックを全般的に回避できる。
2	LOCK TABLE文で表全体をロックできないか検討する。	
3	二つ以上の表をアクセスする場合、アクセス順序を統一する。	アクセス順序の逆転によるデッドロック全般の発生確率を減らせる。
4	一つの表内の行アクセス順序も格納順、キー順などにできるだけ統一する。	
5	検索した行を更新、削除する場合は、SELECT文の排他オプションにWITH EXCLUSIVE LOCKを指定する。(*2)	排他モードの行ロックを検索時に確保しておくことにより、行の更新に伴う行⇔行間のデッドロック発生確率を減らせる。
6(*1)	複数列で条件検索する場合、複数列インデクスの適用を検討する。	インデクス検索時のロック範囲を減らすことにより、行⇔行間のデッドロック発生確率を減らせる。
7(*1)	検索条件はできるだけインデクスの付いた列の=条件とする。	

(*1): インデクスキー値無排他を適用しない(pd_indexlock_mode =KEY)場合。

(*2):カーソル指定にFOR UPDATE句を指定した場合も、通常は暗黙的にEXCLUSIVEの排他がかかりますが、クライアント環境変数PDISLLVLの指定によってはEXCLUSIVEの排他がかからない場合がありますので、FOR UPDATEを排他オプションの代用にはしないでください。
クライアント環境定義にPDFORUPDATEEXLOCK=YESを指定すると、FOR UPDATE指定によって、必ずEXCLUSIVEの排他を取得するようになります。他DBMSからのUAPの移行などでFOR UPDATE指定を排他オプションとして使用する場合は、PDFORUPDATEEXLOCK=YESの指定を検討してください。

HiRDB Version 9 09-50の推奨モードでは、省略値がYESになりました。

HITACHI
Inspire the Next

付録A. DB排他制御のチューニング

A.1 排他制御の概要

A.2 デッドロック回避策

A.3 確認するチューニング情報

A.4 チューニング情報の見方

付録A-3-1　確認するチューニング情報

HITACHI
Inspire the Next

解説　チューニング情報から以下の確認項目を確認してください。

確認項目	取得するチューニング情報	確認項目の見方
デッドロック件数、排他待ち時間	システムの稼働に関する統計情報	付録A-4-1参照
資源種別、資源名称	デッドロック・タイムアウト情報	付録A-4-2参照
	サーバの排他制御の状態表示	付録A-4-3参照

◆手順

■デッドロックの場合

1. システムの稼働に関する統計情報でデッドロック件数を見てデッドロック発生の有無を確認します。
2. デッドロックが発生した場合は、エラー情報としてデッドロック・タイムアウト情報が出力されます。資源種別と資源名称を確認し、排他資源を特定(*1)します。

■排他待ちの場合

1. システムの稼働に関する統計情報で排他待ち時間を見て排他待ちの発生の有無を確認します。
2. サーバの排他制御の状態表示を取得し、資源種別と資源名称を確認します。
資源種別と資源名称より排他資源を特定(*1)します。

(*1):ディクショナリ表を検索することにより、排他資源を特定できます。
特定方法については、付録**E-1**を参照してください。

HITACHI
Inspire the Next

付録A. DB排他制御のチューニング

A.1 排他制御の概要

A.2 デッドロック回避策

A.3 確認するチューニング情報

A.4 チューニング情報の見方

付録A-4-1　デッドロック件数、排他待ち時間の見方

HITACHI
Inspire the Next

解説　システムの稼働に関する統計情報から、デッドロック件数、排他待ち時間を確認してください。

```
------------------------------------------------------------------------------
HOST = db01
------------------------------------------------------------------------------
EDIT TIME 2006/01/12 10:00:00 - 2006/01/12 10:10:00
SERVER : ********
                                          FREQ   MAX    MIN    AVG
                                          -----  -----  -----  -----
<SCHEDULE>          QUEUE LENGTH           293      2      1      1
                    MESSAGE LENGTH         293    556    548    556
<PROCESS>           # OF USER SERVER ABORT    0
                    # OF SYSTEM SERVER ABORT   0
                    # OF PROCESS                 1.03k  1.03k  1.03k
                    # OF PROCESS ON SERVICE        244    232    237
                    # OF REQ PROCESS OVER MAX    0
<TRANSACTION>       # OF COMMIT            12.4k
                    # OF ROLLBACK            19
```

排他制御情報　排他待ち時間(ミリ秒)　排他待ち回数　排他待ち時間(最大・最小・平均)

```
<LOCK>              WAIT TIME              10  8.84k     19  2.69k
                    QUEUE LENGTH           10      2      1      1
                    # OF DEADLOCK           3
                    % OF USE LOCK TABLE     1      0      0      0
```

デッドロック件数

デッドロック情報

付録A-4-2　資源種別、資源名称の見方

HITACHI
Inspire the Next

解説　デッドロック情報から資源種別、資源名称を確認し、排他資源を特定してください。

```
Deadlock information                                    Jun  2 06:12:43 2006

program:SPPY415
server:SDS    pid:5251
trnbid:q192u19200000000    actid:1-1-4  dprio:64
 occupy
  server:SDS  lock mode:PR    kind:0007
  resource info:00000600000019010002007d0000
 wait
  server:SDS  lock mode:EX    kind:0007
  resource info:00000600000019010002007d0000
wait start time 06:12:43

program:SPPE201
server:SDS    pid:5249
trnbid:q192u19200000003    actid:1-1-6  dprio:64
 occupy
  server:SDS  lock mode:PR    kind:0007
  resource info:00000600000019010002007d0000
 wait
  server:SDS  lock mode:EX    kind:0007
  resource info:00000600000019010002007d0000
 wait start time 06:12:43
```

資源種別

資源名称

付録A-4-3　資源種別、資源名称の見方

HITACHI
Inspire the Next

解説　サーバの排他制御の状態表示から資源種別、資源名称を確認し、排他資源を特定してください。

pdls -d lck

資源種別　資源名称

```
HOSTNAME : db01(061503)
SVID         PID     TID  KIND  RESOURCE
 WAIT   MODE   TRNBID                ACTID                LOCK-STATE   DPRIO
SDS          5251       1  0007  00000600000019010002007d0000
    1   PR     q192u19200000000     1-1-4                N              64
```

HITACHI
Inspire the Next

付録B. コマンド、オペランド文法

付録B-1　pdstbegin、pdstend、pdlogsync文法

HITACHI
Inspire the Next

pdstbegin 〔-k 統計情報種別〕〔-m 時間間隔〕〔-a〕〔-w〕

-k　統計情報種別：
　出力する統計情報の種別を指定します。
　　sys：システムの稼働に関する統計情報
　　buf：グローバルバッファプールに関する統計情報
　　dfw：デファードライト処理に関する統計情報
　　dio：データベースの入出力に関する統計情報　HiRDB Version 9 09-50サポート
　　uap：UAPに関する統計情報
-m　時間間隔　((1～1440))《10》：
　統計情報sysの収集間隔を、分単位で指定します。
-a：システム全体および全サーバの統計情報を出力します。
-w：統計情報にスレッド間ロック待ち時間の情報を取得する場合に指定します。(バージョン09-03より)

pdstend　〔-k 統計情報種別〕〔-a〕〔-w〕

-k 統計情報種別：
　pdstbeginコマンドの-kオプションに指定した値の中で、収集を終了させたい統計ログの種別を指定します。
-a：システム全体および全サーバの統計情報の出力を停止します。
-w：スレッド間ロック待ち時間の統計情報取得を停止します。(バージョン09-03より)

pdlogsync -d sys

シンクポイントダンプを取得します。

付録B-2　pdstedit文法

HITACHI
Inspire the Next

解説　pdsteditコマンドの文法について解説します。

pdstedit〔-k 統計情報種別〕〔-m 集計間隔〕〔-i 統計入力アンロードファイル名〕
　　〔-o DAT形式ファイル出力先ディレクトリ名〕〔-b〕〔-e sec〕

-k　統計情報種別：収集した統計ログの中で、出力させたい統計ログの種別を指定します。

-m　時間間隔　((1～1440))《60》：統計ログの集計間隔を分単位で指定します。

-i　統計入力アンロードファイル名：解析対象の統計ログファイル名を指定します。

-o　DAT形式ファイル出力先ディレクトリ名：
　統計入力アンロードファイルから統計情報を収集し、DAT形式ファイルを作成したい場合、そのDAT形式ファイルを作成するディレクトリの名称を指定します。
　※DAT形式で出力することを推奨します。

-b：
　DAT形式ファイルにタイトルバーを出力する場合に指定します。このオプションを指定する場合、同時に-oオプションも指定する必要があります。-oオプションを指定していない場合、このオプションの指定は無視されます。

-e sec：
　DAT形式ファイル出力時に、出力フォーマットを変更する場合に指定します。このオプションを指定する場合、同時に-oオプションも指定する必要があります。
　secを指定した場合、次の統計情報の統計ログ取得時刻に秒値が出力されます。
　　・グローバルバッファプールに関する統計情報
　　・データベース操作に関するHiRDBファイルの統計情報
　　・デファードライト処理に関する統計情報

付録B-3　pdbuflsコマンド

HITACHI
Inspire the Next

解説　pdbuflsコマンドの文法について解説します。

```
pdbufls　〔-k 出力種別〕〔-d〕〔-x〔-y〕〕〔-M〕〔-N〕
　　〔{-s サーバ名〔,サーバ名〕|-a グローバルバッファ名〔,グローバルバッファ名〕…}〕
　　〔-W 実行監視時間〕
```

バージョン09-50の変更点

OTHER用グローバルバッファに割り当てたRDエリア名称を表示できる-Nオプションをサポートしました。

バージョン09-50の変更点

pd_utl_exec_timeオペランドよりも多くのユティリティおよび運用コマンドを対象とした実行時間監視ができるシステム共通定義pd cmd exec timeオペランドと各ユティリティおよび運用コマンドで個別に監視時間を設定できる-Wオプションをサポートしました。

-k　出力種別　((def | sts | all))　《sts》:
　def:グローバルバッファの定義情報の表示
　sts:グローバルバッファの統計情報の表示
　all :グローバルバッファの定義情報および統計情報の表示
-d:HiRDB開始時点からのグローバルバッファの統計情報を表示する場合に指定します。
-x
　DAT形式で情報を表示する場合に指定します。-k allの場合、このオプションは指定できません。
-y:-xオプション指定時に、ヘッダを付ける場合に指定します。
-M:インメモリデータバッファの情報を表示する場合に指定します。
-N:OTHER用グローバルバッファに割り当てているRDエリアの名称を表示する場合に指定します。
-W:pdbuflsコマンドの実行時間を監視する場合に，その監視時間を分単位で指定します。

付録B-4　pdobils文法

HITACHI
Inspire the Next

解説　pdobilsコマンドの文法について解説します。

pdobils〔-s サーバ名〕〔-R｜-r〕〔-C〔区切り文字〕〔-H〕〕〔-e〕
　　〔-U〕〔-NR〕〔-N SQLオブジェクト番号〔,SQLオブジェクト番号〕…〕

下線部分は、HiRDB Version 9 09-50でサポートしたオプション

-R｜-r ：統計情報のカウンタを初期化する場合に指定します。例えば、統計情報のカウンタがオーバフローした場合など、このオプションを指定します。

-R：SQLオブジェクトバッファ統計情報を出力した後に、カウンタを初期化する場合に指定します。一定間隔で統計情報を繰り返し取得する場合は、このオプションを指定することをお勧めします。

-r：SQLオブジェクトバッファ統計情報のカウンタの初期化だけを行う場合に指定します。

-C〔区切り文字〕 ～＜文字列＞((1～10)):
統計情報をDAT形式で出力する場合に指定します。
要素を区切って出力したい場合は、区切り文字を指定します。省略した場合、タブ記号が区切り文字になります。

-H: -Cオプション指定時に、1行目にタイトル行を出力する場合に指定します。

-e:実行回数(出力形式のEXECUTE COUNTの項目)が1以上の統計情報だけを出力する場合に指定します。

-U:pdobilsコマンド実行時に、実行中のSQLのSQLオブジェクト情報と、そのSQLを実行しているUAPの情報を出力する場合に指定します。

-NR:SQL文中の改行コード(0x0A)、および復帰コード(0x0D)を、空白コード(0x20)に置き換える場合に指定します。

-N SQLオブジェクト番号〔,SQLオブジェクト番号〕… ：
SQLオブジェクトに関する保守情報を出力する場合、出力するSQLオブジェクトのSQLオブジェクト番号を指定します。

付録B-5　pdbuffer文法

HITACHI
Inspire the Next

解説　pdbufferオペランドの文法について解説します。

```
pdbuffer　-a グローバルバッファ名
{-r RDエリア名〔, RDエリア名〕… | -b RDエリア名〔, RDエリア名〕… | -o | -i 認可識別子. インデクス識別子 }
-n バッファ面数 〔-l バッファサイズ〕
〔-m 同時実行最大プリフェッチ数〕〔-p 一括入力最大ページ数〕
〔-w デファードライトトリガ時の更新ページ出力比率〕〔-y デファードライトトリガ契機の更新バッファ面数〕
```

-r　RDエリア名：グローバルバッファを割り当てるRDエリアの名称を指定します。

-b　RDエリア名：LOB用グローバルバッファを割り当てるRDエリアの名称を指定します。

-o：-rオプションで指定していないすべてのRDエリアに，グローバルバッファを割り当てる場合に指定します。

-n　バッファ面数：グローバルバッファの面数を指定します。

-l　バッファサイズ：グローバルバッファのバッファ1面のサイズをキロバイト単位で指定します。

-w　デファードライトトリガ時の更新ページ出力比率：
デファードライトトリガでの更新ページ出力比率をパーセントで指定します。

-y　デファードライトトリガ契機の更新バッファ面数：
デファードライトトリガのトリガ契機を更新バッファ面数で指定します。

付録B-6 PDUAPREPLVL文法

HITACHI
Inspire the Next

解説 PDUAPREPLVLオペランドの文法について解説します。

PDUAPREPLVL={[s] [u[o][t]] [p] [r] | [a [o][t]]}

s:SQL単位の情報が出力されます。また、SQLトレース情報も出力されます。

u:UAP単位の情報が出力されます。

p:アクセスパス情報が出力されます。

r:SQL実行時の中間結果情報が出力されます。

o:UAP単位の情報にスレッド間ロック待ち時間が出力されます。uまたはaが指定されていない場合、このオプションを指定しても無視されます。このオプションを指定すると、システム全体の性能に影響を与えるおそれがあります。通常の運用では指定しないでください。

t:UAP単位の情報をトランザクション単位に集計して出力します。u又はaが指定されていない場合、このオプションを指定しても無視されます。

HiRDB Version 9 09-50サポート

a:suprを指定した場合と同じ情報が出力されます。

s, u, pおよびrを組み合わせて指定できます(su, sr, uprなど)。uまたはaを指定した場合、o, tも指定できます。なお、sまたはaを指定しない場合、SQLトレース情報は出力されません。

HITACHI
Inspire the Next

付録C. UAP統計レポートの中間結果情報

付録C-1　SQL実行時の中間結果情報

HITACHI
Inspire the Next

解説　UAP統計レポート中のSQL実行時の中間結果情報には、「集合演算情報」、「問合せ処理情報」、「結合処理情報」、「実表検索処理情報」があります。

■SQL実行時の中間結果情報出力例

```
Result of SQL Execution :
-------------------------------------------------------------------------------
Connect No      :    2
UAP Source      : pdsqlw-2264
Section No      : 1
----- QUERY EXPRESSION BODY ID : ... -----  ← 集合演算情報
... (略) ...
----- QUERY ID : 1 -----                    ← 問合せ処理情報
Query           : 26 ROWS                   ← 問合せ結果の行数
JOIN                                        ← 結合処理情報
 # Join ID      : 1
   Row Count    : 26 ROWS                   ← 結合処理の結果の行数
   Left         : 26 ROWS                   ← 左側の結合相手から取り出した行数
   Right        : 26 ROWS                   ← 右側の結合相手から取り出した行数
   Join Type    : NESTED LOOPS JOIN(LEFT OUTER)
SCAN                                                ← 実表検索処理情報
 # Table Name   : STABLE01(A) 0x00020082(131202)
   RowCount     : 26 ROWS                           ← 実表から取り出した行数
   Index Name   : STABLE01I_1 0x00030105(196869)
                  Search     : 26 ROWS               ← サーチ条件で絞り込まれた結果の行数
 # Table Name   : STABLE02(B) 0x0002007f(131199)
   RowCount     : 26 ROWS
   Index Name   : STABLE02I_1 0x00030106(196870)
                  Search     : 18 ROWS
... (略) ...
```

付録C-2　集合演算情報＆問合せ処理情報

HITACHI
Inspire the Next

解説　集合演算情報と問合せ処理情報の詳細を解説します。

集合演算情報では、実際に行われた集合演算とその結果行数などを確認できます。
■集合演算情報出力例

```
----- QUERY EXPRESSION -----
Query          : 5 ROWS                          ← 問合せ式の結果の行数
Limit          : 5 ROWS <-- 20 ROWS              ← LIMIT処理の出力行数 <-- LIMIT処理の入力行数
Order by       : 20 ROWS                         ← ソート処理の行数
SetOpe Process : LID(1) = 20 ROWS <-- QID(1) UNION QID(2)   ← 集合演算の結果の行数
```

問合せ処理情報では、Limit処理、ソート処理、重複排除処理の結果行数などが確認できます。
■問合せ処理情報出力例

```
----- QUERY ID : 1 -----
Query          : 10 ROWS                         ← 問合せの結果の行数
Limit          : 10 ROWS <-- 75 ROWS             ← LIMIT処理の出力行数 <-- LIMIT処理の入力行数
Order by       : 75 ROWS                         ← ソート処理の行数
Distinct       : 75 ROWS <-- 120 ROWS            ← 重複排除処理の出力行数<--重複排除処理の入力行数
Having         : 120 ROWS                        ← HAVING句を評価した後の行数
Group by       : 120 ROWS <-- 150 ROWS           ← グループ分け処理出力行数<--グループ分け処理入力行数
```

付録C-3　結合処理情報

HITACHI
Inspire the Next

解説　結合処理情報の詳細を解説します。

結合処理情報は、結合処理種別と各結合表から取り出した行数を照らし合わせることで、本番データ件数が入る前のチューニング情報として役立てることができます。

■結合処理情報出力例

```
JOIN
 # Join ID      : 1
   Row Count    : 90 ROWS                    ← 結合処理結果行数
   Left         : 1 ROWS                     ← (a)左側の表から取り出した行数
   Right        : 90 ROWS                    ← (b)右側の表から取り出した行数
   Join Type    : NESTED LOOPS JOIN(INNER)   ← 結合処理種別
```

例えば、Join TypeがNESTED LOOPS JOINの場合、「(a)左側の表から取り出した行数」が本番データの特性を考慮して、十分絞り込めるかどうかを検討してください。逆に「(b)右側の表から取り出した行数」のほうが絞り込めると考えられる場合は、結合順序を変更できるかどうかを検討してください。

つまり、(a)>>(b)が成り立つ場合は結合順序の入れ替えなどの対策を検討する必要があります。

付録C-4 実表検索処理情報

HITACHI
Inspire the Next

解説 実表検索処理情報の詳細を解説します。

実表検索処理情報からは、実表から取り出した行数、サーチ条件、キー条件で絞り込まれた行数が分かりますので、インデクスの検討、インデクス構成列の並び順の検討に役立ちます。

■実表検索処理情報出力例

```
SCAN
 # Table Name   : STABLE01 0x00020082(131202)
   RowCount     : 20 ROWS                          ← (d)実表から取り出した行数
   Index Name   : (PRIMARY0000131202) 0x00030104(196868)
                      Search     : 50 ROWS    ← (c)サーチ条件で絞り込まれた結果行数
                      Key        : 40 ROWS    ← (e)キー条件で絞り込まれた結果行数
 # Table Name   : STABLE02 0x0002007f(131199)
   RowCount     : 90 ROWS                          ← (d)実表から取り出した行数
   Index Name   : (PRIMARY0000131199) 0x00030102(196866)
                      Search     : 90 ROWS    ← (c)サーチ条件で絞り込まれた結果行数
```

上記の例において、(c)はインデクスで絞り込まれたデータ件数となり、(d)はすべての条件で絞り込まれたデータ件数となります。たとえば、(c)>>(d)が成り立つ場合、インデクスでは有効に絞り込まれていないことが考えられます。この場合は、(c)≒(d)となるインデクスの検討を行うことになります。また、(e)はキー条件まで評価したヒット件数となります。(c)>>(e)の場合はインデクス構成列の並びを適切にするような検討をする必要があります。

なお、アクセスパス情報上ではインデクスのAT検索になっていても、絞込み値の重複が多い場合は、他のインデクスを利用したRANGE検索のほうが性能が良い場合もあります。このような絞込み値の重複数の調査においても、実表検索処理情報の「(c)サーチ条件で絞り込まれた結果行数」が役立ちます。

HITACHI
Inspire the Next

付録D. その他のSQL設計

付録D-1　LIKE述語のサーチ条件

HITACHI
Inspire the Next

LIKE述語の分類 （パターン文字列）	SQL 探索条件	説明 →サーチ条件化可能性
前方一致比較 （定数）	T1.C1 LIKE 'abc%'	'abc'で始まる文字列のように、 文字列の先頭部分のみ一致することを指定 →範囲条件化可能　→**サーチ条件化可能**
複合 前方一致比較 （定数）	T1.C1 LIKE 'abc%xyz' T1.C1 LIKE 'abc__'	文字列の先頭部分が指定され、 さらに、それ以外の部分のパターンも指定 →範囲条件を付加し、**サーチ条件追加可能**
前方一致以外 （定数）	T1.C1 LIKE '%abc' T1.C1 LIKE '__abc' T1.C1 LIKE '%abc%' T1.C1 LIKE '_a_b_c_'	文字列の先頭部分に任意の文字列（または文字）が 指定され、先頭部分が決まらない →範囲条件生成不可→**サーチ条件化不可**
（埋込み変数， ?パラメタ， SQL変数 または SQLパラメタ）	T1.C1 LIKE ? T1.C1 LIKE :xxx T1.C1 LIKE :xxx	パターン文字列が定数でないため、 SQL解析時には、分類できない →3ケースのアクセス方法を準備しておき、 パターン文字列値の入力時に、**切り替え**

T1.C1 LIKE 'abc%' → T1.C1 BETWEEN 'abc'||x'00'AND 'abc'||x'ff'

T1.C1 LIKE 'abc%xyz' → T1.C1 BETWEEN 'abc'||x'00' AND 'abc'||x'ff' AND
T1.C1 LIKE 'abc%xyz'

インデクスを効率よく使用するためには、
なるべく前方一致で使ってください。

ただし、T1.C1が可変長の場合、"||x'00'"無

付録D-2　LIKE述語のサーチ条件(動的切り替え)

HITACHI
Inspire the Next

分類 (パターン文字列)	SQL	アクセスパス情報のサーチ条件およびキー条件
(埋込み変数 または ?ﾊﾟﾗﾒﾀ)	T1.C1 LIKE :xxx または T1.C1 LIKE ?	SearchCnd:RANGE(CS-CE) [<?(*n*)0>, <?(*n*)f>] KeyCnd:<T1.C1 LIKE ?(*n*)>

入力値によって、前方一致か、複合前方一致か、それ以外かを判定し、条件の切り替え

→入力値が前方一致(または複合前方一致)以外のとき、
　サーチ条件がなくなり、インデクスのフルスキャンになる可能性があることに注意が必要

前方一致比較 (定数)	T1.C1 LIKE 'abc%'	SearchCnd:RANGE(CS-CE)['abc'00, 'abc'ff]
複合 前方一致比較 (定数)	T1.C1 LIKE 'abc%xyz'	SearchCnd:RANGE(CS-CE)['abc'00, 'abc'ff] KeyCnd:T1.C1 LIKE 'abc%xyz'
前方一致以外 (定数)	T1.C1 LIKE '%abc'	KeyCnd:T1.C1 LIKE '%abc'

＜アクセスパス情報の補足＞

固定長の列	'abc'00	<?(*n*)0>
可変長の列	'abc'	<?(*n*)>

n：SQL中での出現番号

埋込み変数または?ﾊﾟﾗﾒﾀ	?(*n*)
SQL変数	*SQL変数名*
SQLパラメタ	*SQLパラメタ名*

付録D-3　インデクスとソート(全件検索)(1)

HITACHI
Inspire the Next

利用インデクスとソートキーの関係

ソートキーが
　インデクスの構成列と一致する
または
　インデクスの構成列の
　　先頭部分の列の組と一致する

インデクス	C1, C2, C3
ソートキー	C1, C2, C3
ソートキー	C1, C2
ソートキー	C1

＋

キーの構成列の順序方向(昇順・降順)との関係 (GROUP BYの場合は考慮不要)

ソートキーとインデクスの対応する各列の昇順・降順指定が一致するか
または完全に逆転している

インデクス	ソートキー
C1 ASC , C2 ASC	C1 ASC , C2 ASC
C1 DESC, C2 DESC	C1 DESC, C2 DESC
C1 ASC , C2 DESC	C1 ASC , C2 DESC
C1 DESC, C2 ASC	C1 DESC, C2 ASC

内部的なソート処理削減の可能性
(インデクスソートキャンセル)

凡例
インデクスを正方向に利用
インデクスを逆方向に利用

付録D-4　インデクスとソート(全件検索)(2)

HITACHI
Inspire the Next

利用インデクスとソートキーによる内部ソート処理削減の可能性

INDEX ON T1(C1, C2, C3)　(GROUP BYの場合はASC/DESCの考慮不要)

SELECT * FROM T1 ORDER BY C1, C2, C3	○
SELECT * FROM T1 ORDER BY C1, C2, C3, C4	×
SELECT * FROM T1 ORDER BY C3, C2, C1	×
SELECT * FROM T1 ORDER BY C1, C2	○
SELECT * FROM T1 ORDER BY C2, C3	×
SELECT * FROM T1 ORDER BY C1 DESC, C2 DESC	○
SELECT * FROM T1 ORDER BY C1, C2 DESC	×

その他の必要条件

キースキャンになる
(SQL中で参照する列がすべて
インデクス構成列に含まれる)

内部ソート処理を削除するには、

参照列が少なければ、インデクス
の構成列に参照列を含める

付録D-5　インデクスとソート(条件検索)(１)

HITACHI
Inspire the Next

利用インデクスとソートキーの関係および順序方向(昇順・降順)との関係

ソートキーから＝条件列およびIS NULL条件列を除いた列の組が
　インデクスの構成列から＝条件列およびIS NULL条件列を除いた列の組と一致
または
　インデクスの構成列から＝条件列およびIS NULL条件列を除いた
　　先頭部分の列の組と一致する

＋

ソートキーから＝条件列およびIS NULL条件列を除いた列と
　インデクスの対応する各列の昇順・降順指定が
　　一致するかまたは完全に逆転している

利用インデクスとソートキーによる内部ソート処理削減の可能性

INDEX ON T1(C1, C2, C3)	(GROUP BYの場合はASC/DESCの考慮不要)
SELECT * FROM T1 WHERE C1=10 ORDER BY C1, C2, C3	○
SELECT * FROM T1 WHERE C1>10 ORDER BY C1, C2, C3	○
SELECT * FROM T1 WHERE C1=10 ORDER BY C2, C3	○
SELECT * FROM T1 WHERE C1>10 ORDER BY C2, C3	×
SELECT * FROM T1 WHERE C1=10 AND C2>20 ORDER BY C2, C3	○
SELECT * FROM T1 WHERE C1=10 ORDER BY C2, C3 DESC	×
SELECT * FROM T1 WHERE C1=10 ORDER BY C2 DESC, C3 DESC	○

付録D-6 インデクスとソート(条件検索)(2)

HITACHI
Inspire the Next

内部ソート処理を削除するには、

次の複数列インデクスを定義する
- =条件列(またはIS NULL条件列)を、第1～第n構成列として連続して含む
- ソートキーの構成列を、その順序で、第n+1構成列以降に連続して含む
  (ただし、第1～第n構成列に含めた列は除く)
- 第n+1構成列以降のソート列の昇順・降順指定をすべて同じかまたはすべて逆

(注)探索条件によりマッチした他のインデクスが利用される場合があります。

```
SELECT * FROM T1 WHERE C1=10 AND …AND Cn=20
        ORDER BY Cn+1 ASC, Cn+2 DESC, …  Cm ASC
```

```
INDEX ON T1(C1, … , Cn, Cn+1 ASC, Cn+2 DESC, …, Cm ASC)
```

その他の制限(全件検索、条件検索共通)

次の場合は、内部ソート処理は削除されない。
- DISTINCTを指定
- 表がサーバ内で複数RDエリアに分割格納され、インデクスも分割インデクス
  (分割キーのすべての構成列に=条件がある場合を除く)

(以降は、パラレルサーバのみ)
- INSERT～SELECT文中(GroupByのみの制限)
- 副問合せ中
- 集合演算を指定
- HAVING句を指定
- FOR UPDATE または FOR READ ONLYを指定

付録D-7　インデクスと集合関数 MAX／MIN(１)

HITACHI
Inspire the Next

集合関数MAX, MINでのインデクス利用

インデクスの最小値・最大値だけを参照して、または
インデクスのサーチ範囲内でキー条件を満たす最初の値までだけを参照し、
集合関数MAX, MINを求める

Group by Mode:

IMPLICIT MIN-MAX INDEX

集合関数MAX, MINでのインデクス利用条件

＜探索条件がない場合＞　--　次の条件を満たすインデクスが利用される
・MAX, MINの引数の列を、第1構成列に含む

SELECT MAX(C_1), MIN(C_1) FROM T1

INDEX ON T1(C_1 [, C_2 ･･･])

＜探索条件がある場合＞　--　次の条件を満たすインデクスが利用される
・=条件列(またはIS NULL条件列)を、第１～第n構成列として連続して含む
・MAX, MINの引数の列を、第n+1構成列に含む
・その他の条件列を第n+2構成列以降に含む

SELECT MAX(C_{n+1}), MIN(C_{n+1}) FROM T1
　　WHERE C_1=10 AND ･･･AND C_n=20 AND C_{n+2} <30 AND C_m >40

INDEX ON T1(C_1, ･･･ , C_n, C_{n+1}, C_{n+2}, ･･･, C_m [, C_{m+1} ･･･])

付録D-8 インデクスと集合関数 MAX／MIN(２)

HITACHI
Inspire the Next

集合関数MAX/MINでのインデクス利用の可否例

INDEX ON T1(C1, C2, C3, C4)

SELECT MAX(C1), MIN(C1) FROM T1	○
SELECT MAX(C2), MIN(C2) FROM T1	×
SELECT MAX(C1), MAX(C2) FROM T1	×
SELECT MAX(C3) FROM T1 WHERE C1=10 AND C2=20	○
SELECT MAX(C3) FROM T1 WHERE C1=10 AND C2<20	×
SELECT MAX(C3) FROM T1 WHERE C1=10 AND C2=20 AND C3<30	○
SELECT MAX(C3) FROM T1 WHERE C1=10 AND C2=20 AND C4<40	○
SELECT MAX(C3) FROM T1 WHERE C1=10 AND C2=20 AND C5<50	×

その他の制限

次の場合は、内部ソート処理は、削除されない。

- 結合検索を指定
- GROUP BY句を指定
- 引数の異なるMAX, MINを指定
- 探索条件に値式または256ﾊﾞｲﾄ以上の値
- 表がサーバ内で複数RDエリアに分割格納され、インデクスも分割インデクス（分割キーのすべての構成列に=条件がある場合を除く）

（以降は、パラレルサーバのみ）

- INSERT～SELECT文中
- 集合演算を指定
- HAVING句を指定
- FOR READ ONLYを指定

付録D-9　ハッシュジョインの定義

HITACHI
Inspire the Next

解説　ハッシュジョインを行うための準備について説明します。
これらの指定をしないと適用されません。

指定の種類	パラメタ	指定値
SQL拡張最適化オプションの指定	システム定義のpd_additional_optimize_level または クライアント環境変数のPDADDITIONALOPTLVL	"COST_BASE_2","APPLY_HASH_JOIN" (「コストベース最適化モード2の適用」および「ハッシュジョイン、副問合せのハッシュ実行」)
ハッシュ表サイズの指定	システム定義のpd_hash_table_size または クライアント環境変数のPDHASHTBLSIZE	ハッシュ表サイズ ここの指定値によって、作業表書き出し有無が変わり、性能を左右します。
作業表用バッファの確保方式の指定	システム定義のpd_work_buff_mode	pool(サーバプロセス単位にバッファプールとして一括して確保)
作業表用バッファのサイズの指定	システム定義のpd_work_buff_size または システム定義のpd_work_buff_expand_limit	作業表用バッファサイズ pd_work_buff_expand_limitを使用すれば、ハッシュジョイン時に動的にメモリを確保します。pdworkに指定する、作業表用ファイルの容量の見積りも行ってください。

付録D-10　SQL最適化指定

HITACHI
Inspire the Next

解説　SQLの構文中にアクセスパスを指示する機能について説明します。

指定の種類	内容
使用インデクス	表に対して検索時に使用するインデクスの指定、またはインデクス利用の抑止(テーブルスキャン)を指定
結合方式	結合表に対して結合方式を指定
副問合せ実行方式	述語中の副問合せに対して副問合せ実行方式を指定

指定されたインデクスが定義されていないなど指定が無効な場合は、
SQL最適化指定を無視する。→ SQLエラーにはならない。

↓ アクセスパス情報で、SQL最適化指定の有効・無効を確認できる

AS SPECIFIED　　:SQL最適化指定が有効
SPECIFICATION IGNORED　　:SQL最適化指定が無効
PARTIALLY IGNORED　　:SQL最適化指定は、一部有効で他は無効(*1)

(*1):使用インデクスのSQL最適化指定で複数のインデクスを指定したときに一部無視した場合

付録D-11　使用インデクスのSQL最適化指定

HITACHI
Inspire the Next

指定の方法	内容
WITH INDEX(インデクス名)	インデクス名のインデクスを使用してインデクススキャン(キースキャン)を行う
WITH INDEX (インデクス名,インデクス名, ・・・)	インデクス名のインデクスをそれぞれ使用して複数インデクス利用を行う
WITHOUT INDEX	インデクスを使用しない(テーブルスキャン)

使用例:

(1) IDX1を使用してインデクススキャンにて検索する。

```
SELECT SNAME FROM ZAIKO WITH INDEX (IDX1)
        WHERE TANKA <= 500
```

(2) IDX1とIDX2を使用して複数インデクス利用にて検索する。

```
SELECT SNAME FROM ZAIKO WITH INDEX (IDX1,IDX2)
        WHERE TANKA <= 500 OR ZSURYO > 100
```

(3) インデクスを使用しないでテーブルスキャンにて検索する。

```
SELECT SNAME FROM ZAIKO WITHOUT INDEX
        WHERE TANKA <= 500
```

付録D-12　結合方式のSQL最適化指定(１)

HITACHI
Inspire the Next

指定の方法	内容
BY NEST	ネストループジョイン(Nested-Loop Join)を行う
BY HASH	ハッシュジョイン(Hash Join)を行う
BY MERGE	マージジョイン(Merge Join)を行う

留意事項:
(1) 結合方式のSQL最適化指定は、結合表(INNER JOIN または LEFT OUTER JOIN)にのみ指定可能である。

(2) 結合の外表内表は、結合表構文の外表内表指定通りとなる。

(3)ハッシュジョインを使用する場合は、ハッシュ表サイズ、作業表用バッファのサイズに適切な値を指定する。なお、SQL拡張最適化オプションの指定は不要であるが、ハッシュ表サイズの指定、作業表用バッファの確保方法の指定、作業表用バッファのサイズの指定は必要である。

付録D-13　結合方式のSQL最適化指定(２)

HITACHI
Inspire the Next

使用例:

(1) ZAIKO表を外表、JUTYU表を内表にNested-Loop Joinにて検索する。

```
 SELECT ZAIKO.SCODE,ZAIKO.SNAME,JUTYU.TCODE
      FROM ZAIKO INNER JOIN BY NEST JUTYU
         ON ZAIKO.SCODE = JUTYU.SCODE
```

(2) ZAIKO表を外表、JUTYU表を内表にHash Joinにて左外結合の検索をする。

```
 SELECT ZAIKO.SCODE,ZAIKO.SNAME,JUTYU.TCODE
      FROM ZAIKO LEFT OUTER JOIN BY HASH JUTYU
         ON ZAIKO.SCODE = JUTYU.SCODE
```

(3) ZAIKO表を外表、JUTYU表を内表にMerge Joinにて検索する。

```
 SELECT ZAIKO.SCODE,ZAIKO.SNAME,JUTYU.TCODE
      FROM ZAIKO INNER JOIN BY MERGE JUTYU
         ON ZAIKO.SCODE = JUTYU.SCODE
```

付録D-14　副問合せ実行方式のSQL最適化指定

HITACHI
Inspire the Next

指定の方法	内容
HASH	副問合せをハッシュ実行(Hash Joinと同様の方式)で評価する
NO HASH	副問合せをハッシュ実行以外で評価する

留意事項:
　副問合せハッシュ実行を使用する場合は、ハッシュ表サイズ、作業表用バッファのサイズに適切な値を指定する。なお、SQL拡張最適化オプションの指定は不要であるが、ハッシュ表サイズの指定、作業表用バッファの確保方法の指定、作業表用バッファのサイズの指定は必要である。

使用例:
(1) 副問合せをハッシュ実行で評価する

```
SELECT SNAME FROM ZAIKO
     WHERE SCODE =ANY
        (HASH SELECT SCODE FROM JUTYU WHERE TCODE = '302S')
```

(2) 副問合せをハッシュ実行以外で評価する

```
SELECT SNAME FROM ZAIKO
     WHERE SCODE =ANY
        (NO HASH SELECT SCODE FROM JUTYU WHERE TCODE = '302S')
```

付録E. 排他資源の特定方法

付録E-1　排他資源の特定方法

HITACHI
Inspire the Next

解説　資源種別、資源名称より、排他資源をディクショナリ表を検索することにより特定できます。

＜例＞　排他資源が行の場合

（資源種別'0007'、資源名称'00000600000019010002007d0000'）
　資源名称の1～6けたがRDエリアID、17～24けたが表IDです。
　（注）資源名称の参照時は、ご使用のプラットフォームのエンディアンにご注意ください。

pdsql等で以下のSQLを実行
　RDエリア名取得(?パラメタ:x'00000006';　←INT型に合わせ4バイトに拡張して指定）

```
SELECT RDAREA_NAME FROM MASTAR.SQL_RDAREAS
WHERE RDAREA_ID = ?
WITHOUT LOCK NOWAIT;
```

　表名取得(?パラメタ:x'0002007d';)

```
SELECT TABLE_NAME FROM MASTAR.SQL_TABLES
WHERE TABLE_ID = ?
WITHOUT LOCK NOWAIT;
```

HITACHI
Inspire the Next

付録F. GUI製品の概要

付録F-1　HiRDB Control Manager概要

HITACHI
Inspire the Next

解説

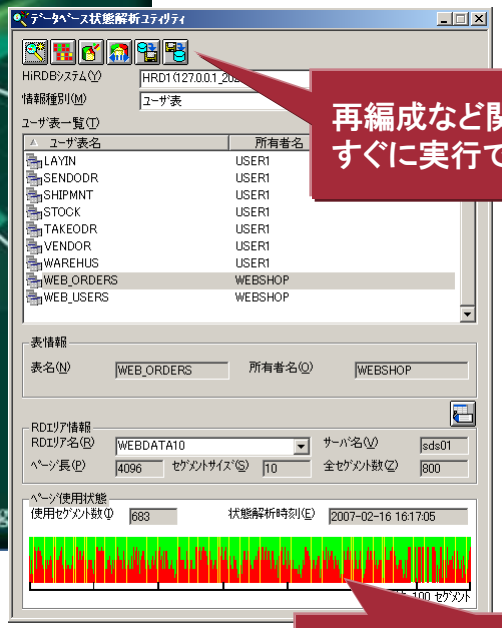
HiRDB Control Manager（以下CMと呼びます）は、GUIを使ってHiRDBの起動・停止やバックアップ・リカバリ、再編成など煩雑な運用操作を簡素化し、運用管理者の負担を軽減するための統合運用ツールです。

Agentは全OSのHiRDBサーバに標準付属！
ConsoleはWindowsのHiRDBサーバに標準付属！

■ウィザードを用いたバックアップ・リカバリ

バックアップやリカバリがウィザード形式で実行できます。

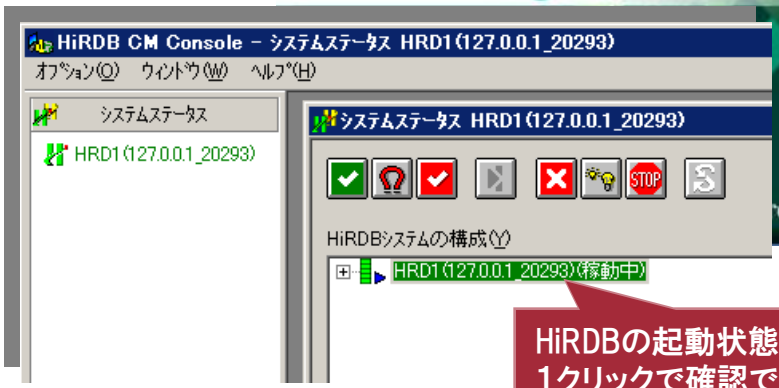
■データベース格納状態の確認



再編成など関連する操作がすぐに実行できます。

HiRDB Control Manager

■HiRDBの起動・停止



HiRDBの起動状態が1クリックで確認できます。

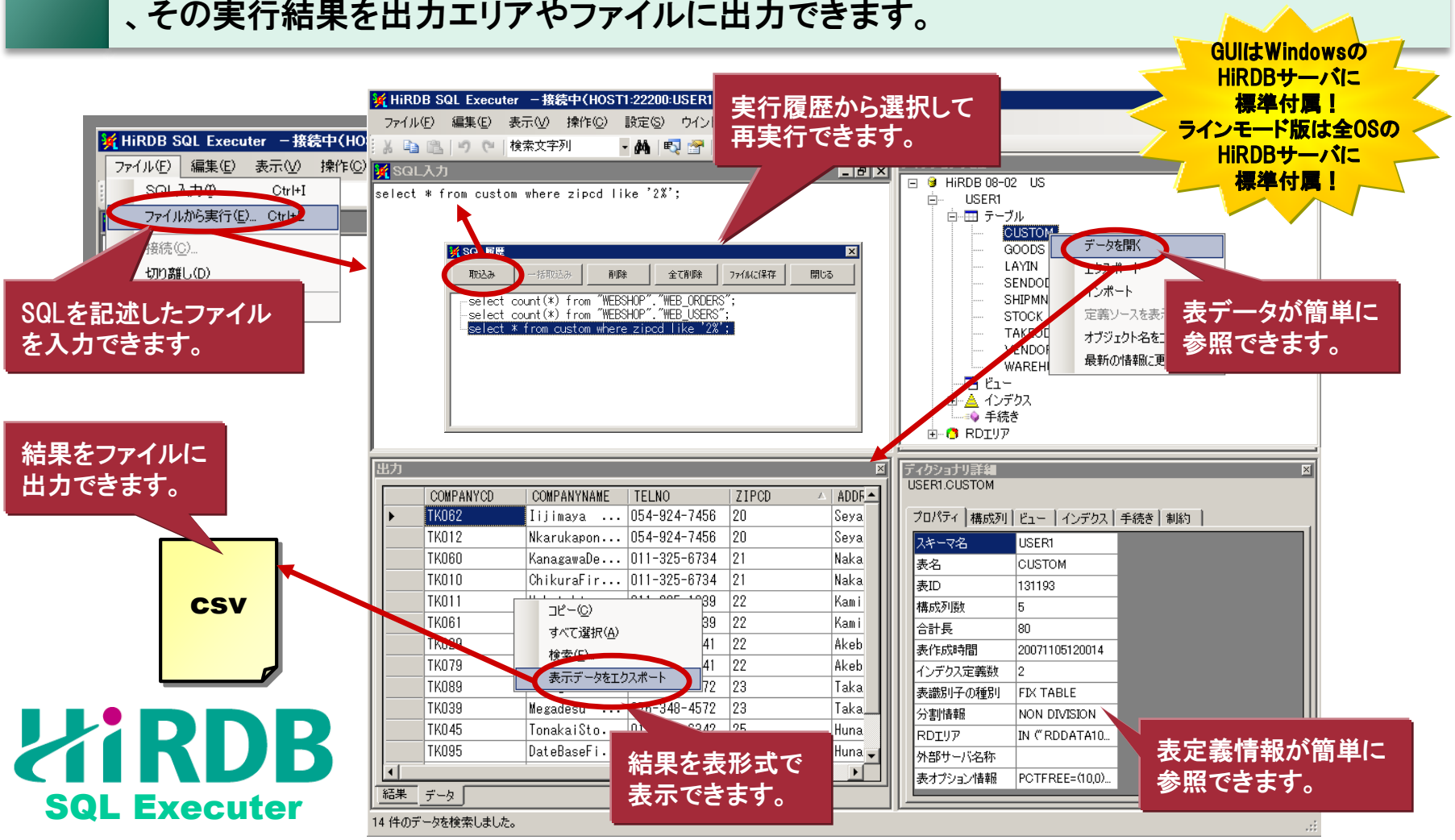
データベースの格納状態が視覚的に確認できます。

付録F-2　HiRDB SQL Executer概要

HITACHI
Inspire the Next

解説

HiRDB SQL Executer（以下Executerと呼びます）は、GUIまたはラインモードを使って会話形式でSQLを実行するツールです。入力エリアやファイルから入力したSQL文を実行し、その実行結果を出力エリアやファイルに出力できます。



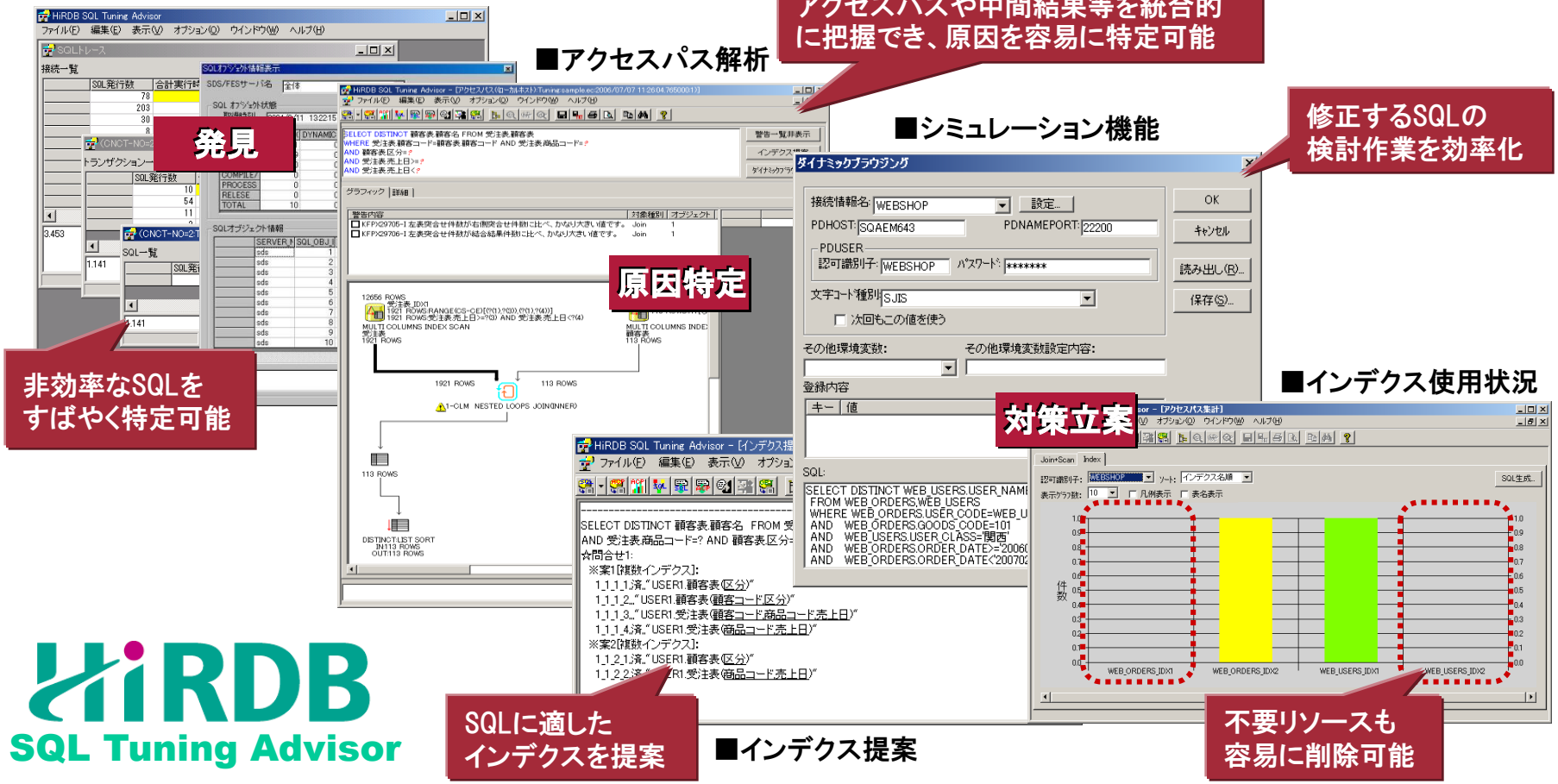
付録F-3　HiRDB SQL Tuning Advisor概要

HITACHI
Inspire the Next

解説

HiRDB SQL Tuning Advisor（以下Tuning Advisorと呼びます）は、GUIを使ってSQLのチューニング作業をわかりやすくガイダンスするツールです。性能上問題のあるアプリケーションから非効率なSQLをすばやく特定し、そのSQLの問題点を容易に特定できます。

■トレース解析

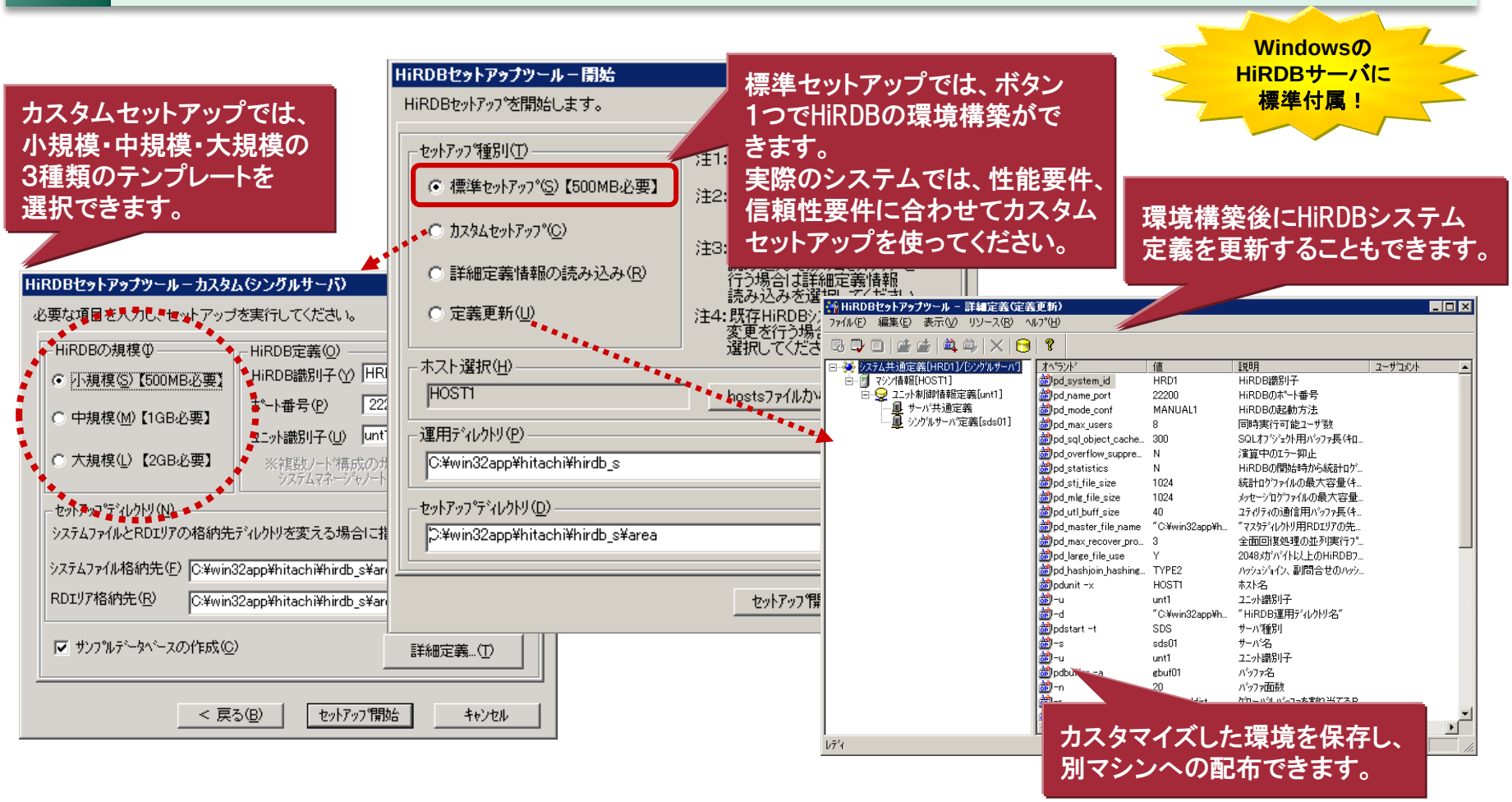


付録F-4　簡易セットアップツール概要

HITACHI
Inspire the Next

解説

簡易セットアップツールは、GUIを使ってHiRDBの環境構築を行うツールです。
HiRDB/Single Server、HiRDB/Parallel Serverの両方に対応しています。



他社商品名、商標等の引用に関する表示など

HITACHI
Inspire the Next

■本資料での表記
本資料では製品名称および名称について次のように表記しています。
ただし、それぞれのプログラムについての表記が必要な場合はそのまま表記しています。
・HP-UX，AIX，およびLinuxを総称してUNIXと表記します。
・Red Hat Enterprise LinuxをLinuxと表記します。
・Windows，Windows Serverを総称してWindowsと表記します。

■商標類
・ 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。
・ Microsoft，Windows，Windows Server，およびExcelは，米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
・ UNIXは，The Open Groupの米国ならびに他の国における登録商標です。
・ IBM，AIXは，世界の多くの国で登録されたInternational Business Machines Corporationの商標です。
・ Red Hat, and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the United States and other countries. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

・ その他記載の会社名、製品名は、それぞれの会社の商号、商標もしくは登録商標です。

■対象となる製品
記載の仕様は、HiRDB Version 9及びVersion 10です。
製品の改良により予告なく記載されている仕様が変更になることがあります。

HITACHI
Inspire the Next

END

HiRDBチューニング解説

株式会社 日立製作所　サービス＆プラットフォームビジネスユニット
サービスプラットフォーム事業本部　DB部


HITACHI
Inspire the Next