
HiRDB技術解説

ユーザ事例に学ぶ HiRDB設計のコツと トラブルシューティング

2016/5

株式会社 日立製作所 ICT事業統括本部
サービスプラットフォーム事業本部 DB部



Contents

1. はじめに
2. システム設計時のコツ
3. システム構築時のコツ
4. アプリケーション開発時のコツ
5. おわりに



1. はじめに

本資料では、稼動環境も含めたHiRDBシステムの設計および構築、アプリケーション開発などのシーン別に、トラブル事例に基づいた設計のコツをご紹介します。

この設計のコツを活かして設計／構築／開発していただくことで、構築期間の短縮や本番環境でのトラブル抑止に繋がります。

また、トラブルが起きてしまった場合の対策についても解説します。

なお、HiRDB Version 9 09-50(以降バージョン09-50と表記)では、省略値のままでお勧めの設定が適用されるように省略値の変更を行いました。各事例でバージョン09-50の省略値変更についても説明しますので、バージョン09-50の適用につきましてもご検討ください。

1-2 作業分類毎の設計コツ記載場所

現象 HiRDBのシステム設計および構築、アプリケーション開発などの作業分類とそれに関連する設計のコツを記載した本テキストでの掲載場所を示します。

作業分類		設計のコツ記載場所
システム設計	HiRDBパラメタ関連	2. 1 節
	スキーマ設計関連	2. 2 節
	ネットワーク関連	2. 3 節
	クラスタ関連	2. 4 節
	運用設計関連	2. 5 節
	OS関連	2. 6 節
	ディスク関連	2. 7 節
	メモリ関連	2. 8 節
システム構築		3章
アプリケーション開発		4章

2. システム設計時のコツ

2.1 HiRDBパラメタ関連

2.2 スキーマ設計関連

2.3 ネットワーク関連

2.4 クラスタ関連

2.5 運用設計関連

2.6 OS関連

2.7 ディスク関連

2.8 メモリ関連

- コツ1 : 無応答の現象を回避するために、監視時間のパラメタは設定しましょう！
 - ▶ 2-1-1 ~ 2-1-2
 - ▶ 2-1-7 ~ 2-1-8
- コツ2 : 監視時間パラメタの関係性に注意しよう！
 - ▶ 2-1-2 ~ 3
- コツ3 : プロセス生死確認を実施しよう！
 - ▶ 2-1-4 ~ 2-1-5
- コツ4 : HiRDBクライアント–サーバ間の通信にFWを設置する場合、受信ポートは固定させる！
 - ▶ 2-1-9~2-1-10

現象 SQLが無応答状態になった。

原因 PDCWAITTIMEが0の場合、通信障害やディスク障害が発生した時に、無応答状態になることがある。この場合、PDCWAITTIMEを省略していたため0が仮定され、SQL実行中に通信障害が発生したことにより、無応答状態になった。

コツ  PDCWAITTIME(クライアントの最大待ち時間)に監視時間を指定してください。

 バージョン09-50で、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。また、PDCWAITTIMEが0の場合でも、実行時間が30分以上経過したSQLに対して警告を出力するようになりました。

現象

HiRDBシステム定義pd_lck_wait_timeout(排他待ち最大時間)を180秒から600秒に拡張したにもかかわらず、SQLが210秒ほどで、KFPA11732-E「タイムアウト」でエラーとなった。

原因

クライアント環境変数PDCWAITTIME(クライアントの最大待ち時間)を210秒から拡張しなかったため、この監視時間を超えたことによるタイムアウトとなってしまった。

コツ

🕒 SQLの実行時間は、PDCWAITTIMEで監視します。よって、PDCWAITTIMEとpd_lck_wait_timeoutに指定する時間は、下記の関係を保つように設定してください。
pd_lck_wait_timeout指定時間を超過した場合に、排他待ちのタイムアウト情報を出力するようになります。

$PDCWAITTIME > pd_lck_wait_timeout$

また、アプリケーション側の監視時間は、PDCWAITTIMEより大きい値にしてください。

トランザクションキューイング機能使用時

トランザクションのキューイング時間は最大で、pd_ha_trn_queuing_wait_timeオペランド^(*1)の値(デフォルト180秒)とpd_ha_trn_restart_retry_timeオペランド^(*2)の値(デフォルト60秒)の合計時間になります。このため、PDCWAITTIMEとpd_lck_wait_timeoutの値に注意してください。

PDCWAITTIME: 系切り替え中もPDCWAITTIMEが計測されているため、系の切り替えに掛かる時間を加算してオペランドの値を決めてください。

pd_lck_wait_timeout: キューイングされているトランザクションが障害の発生していないユニットで排他を確保することがあるため、系の切り替えに掛かる時間を加算してオペランドの値を決めてください。

(*1): トランザクションのキューイング待ち時間 (*2): トランザクション開始要求エラー時のリトライ時間の上限

2-1-3 HiRDB監視時間設定パラメタの説明

<監視時間の説明>

■PDCONNECTWAITTIME:

- ・HiRDBクライアント側での監視。
- ・HiRDBサーバとの接続時、サーバから応答が帰ってくるまでの時間を監視します。監視時間超過時には、UAPIにエラーを返します。

■PDCWAITTIME:

- ・HiRDBクライアント側での監視。
- ・AP(Client)がSQL処理要求をHiRDBサーバ(DB処理プロセス)に送信してから、SQL処理結果をHiRDBサーバから受け取るまでの時間を監視します。監視時間超過時には、DB処理プロセスを終了します。

■PDSWAITTIME:

- ・HiRDBサーバ側での監視。
- ・SQL処理結果をAP(Client)に返してから、次のSQL処理要求をAP(Client)から受けるまでの時間を監視します。監視時間超過時には、DB処理プロセスを終了します。

■PDSWATCHTIME:

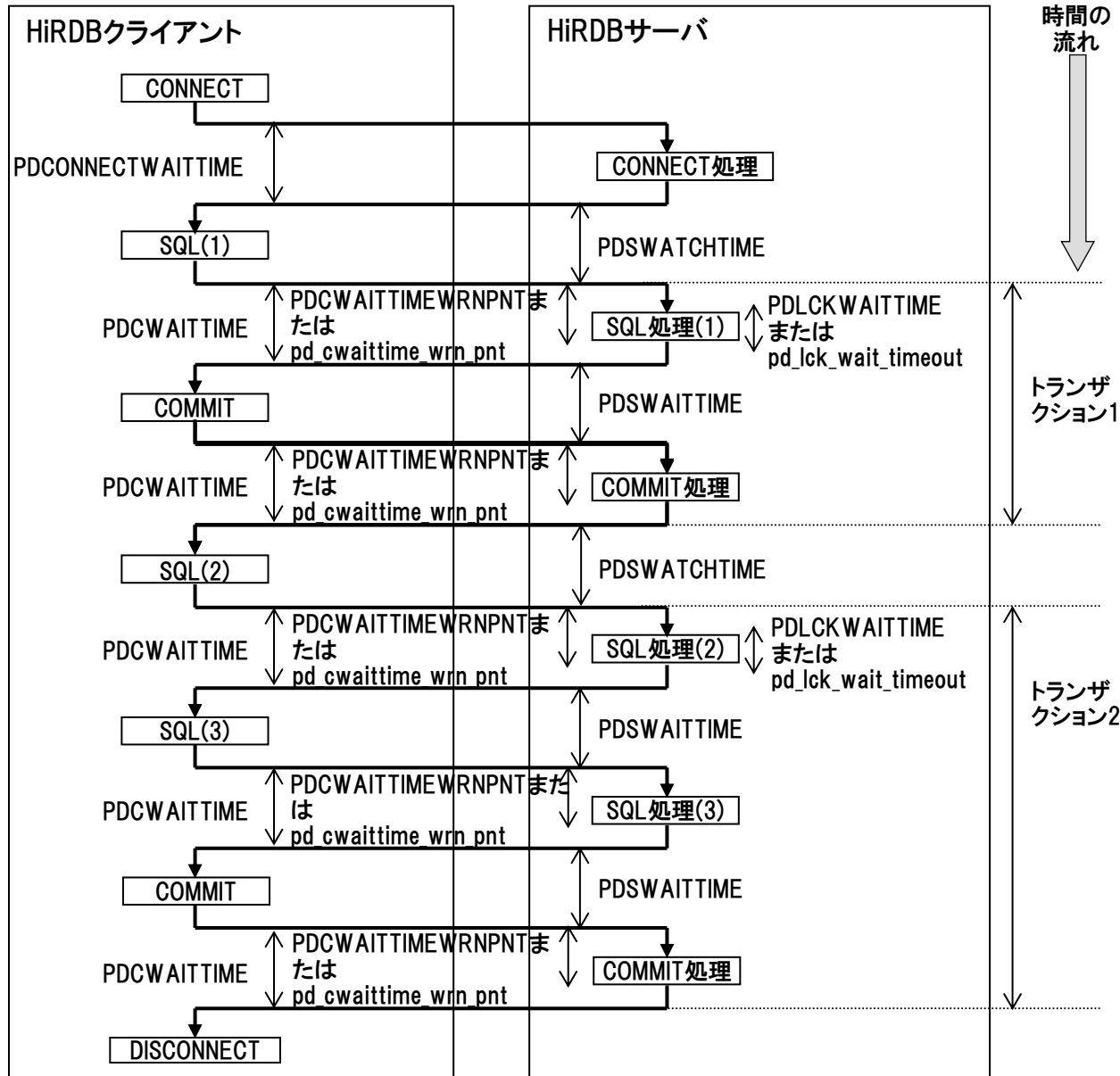
- ・HiRDBサーバ側での監視。
- ・トランザクションの完了をAP(Client)に返してから、次の新規トランザクションとしてのSQL処理要求をAP(Client)から受けるまでの時間を監視します。監視時間超過時には、DB処理プロセスを終了します。

■PDLCKWAITTIME

- pd_lck_wait_timeout:
- ・HiRDBサーバ側での監視。
- ・排他待ち時間を監視します。監視時間超過時にはUAPIにエラーを返します。

■PDCWAITTIMEWRNPNT

- pd_cwaittime_wrn_pnt:
- ・HiRDBサーバ側での監視。
- ・SQL実行時間警告出力機能を使用する場合にこのオペランドを指定します。
- ・SQLの実行時間を監視し、設定した時間の超過時やPDCWAITTIMEに設定した時間に対する比率を超過した場合に、そのSQLに対して警告情報を出力します。



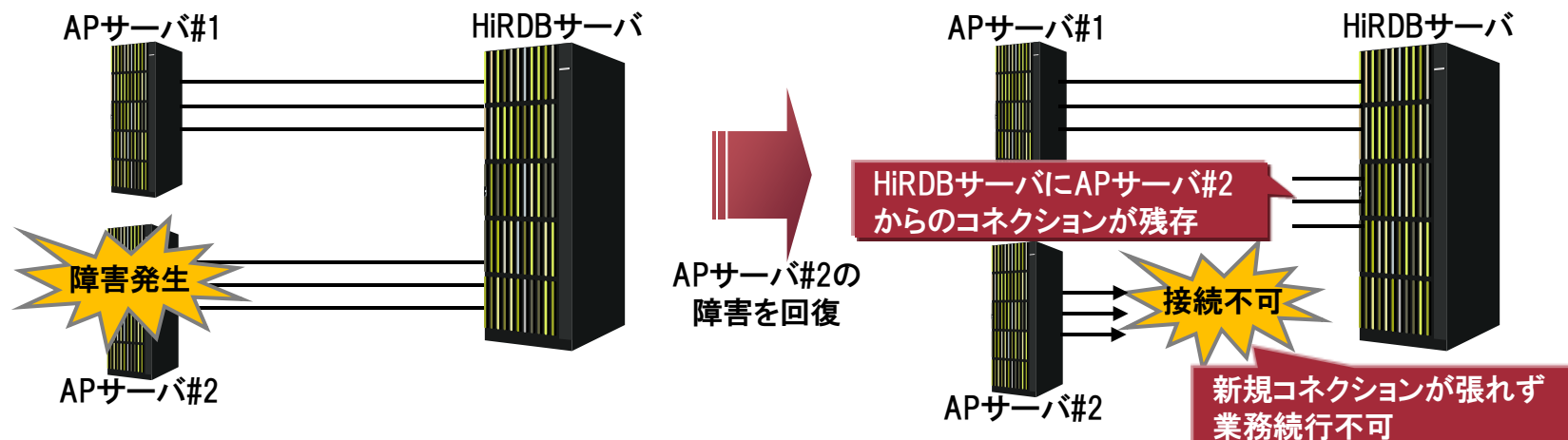
2-1-4 HiRDBサーバ側での監視時間設定(1)

現象

UAPとHiRDBサーバ(SDS、FES)が異なるマシンに配置された環境で、UAPが稼動するマシンやネットワークがダウンし、APサーバの障害回復後、新たなHiRDBへの接続がエラーとなった。

原因

UAPが稼動するマシンやネットワークがダウンすると、HiRDBサーバ側にコネクションが残存する(通信路にリセットが上がらないため、PDSWAITTIMEやPDSWATCHTIMEが無限に設定された環境では残存し続ける)ため、KFPA11932-E(pd_max_usersオーバー)でエラーとなる場合があります。



2-1-5 HiRDBサーバ側での監視時間設定(2)

コツ  PDSWAITTIMEおよびPDSWATCHTIMEの監視時間を無限にしないでください。さらに、パケット送信機能(PDKALVL=2)を指定してください。

 バージョン09-50で、HiRDBクライアントが推奨モードの場合、PDKALVLの省略値が2になりました。ただし、シングルスレッドのHiRDBクライアントライブラリでは、PDKALVLの指定は無効です。

PDSWAITTIME、PDSWATCHTIMEの設定

PDSWAITTIMEには、システムのタイマ設計を厳密に行いたい場合、異常と判断する待ち時間を設定してください。厳密に行う必要がない場合、デフォルト値(600秒)があるため、設定は省略しても構いません。

PDSWATCHTIMEには、OpenTP1の常駐SPP(クライアントが常時接続しているUAP)の場合は、0を指定してください。一方、接続プールを適用したCosminexusのUAPなど(クライアントが常時接続しているUAP)の場合は、0以外を指定しパケット送信機能(PDKALVL=2)を使用して定期的に通信をさせてください。詳細は、**2-1-6** を参照してください。

それ以外で、プロセス残存を回避する目的で指定する場合、業務上トランザクション外で異常と判断する時間を設定してください。

対策 残存コネクションを解放し、コネクションを再接続してください。

残存コネクションを解放し、コネクションを再接続する方法

pdls -d prcで接続時刻とクライアントのIPアドレスを確認し、障害発生前に接続していたプロセスを-d指定のpdcancelコマンドで、強制停止してください。

2-1-6 コネクションプーリング使用時の時間監視

コネクションプーリングとは

コネクションプーリングは、データベースへの接続(コネクション)をプールしておき、それを再利用することで処理性能の向上を図る方法です。

データベースにアクセスするたびに接続を取得する場合に比べて、性能劣化を防ぐ効果があります。

Webアプリケーションサーバなどで、よく用いられている方法です。

コネクションプーリング使用時の設定

コネクションプーリングを使用している環境において、HiRDBサーバ側で時間監視をしていると、プーリングしている接続が切れてしまいます。よって、HiRDBサーバ側での監視時間(クライアント環境変数PDSWATCHTIME)に、0(無限待ち)を設定するか、パケット送信機能を使用して定期的に通信をさせてください。

■パケット送信機能

HiRDBクライアントからHiRDBサーバに対して、定期的にパケットを送信する機能です。

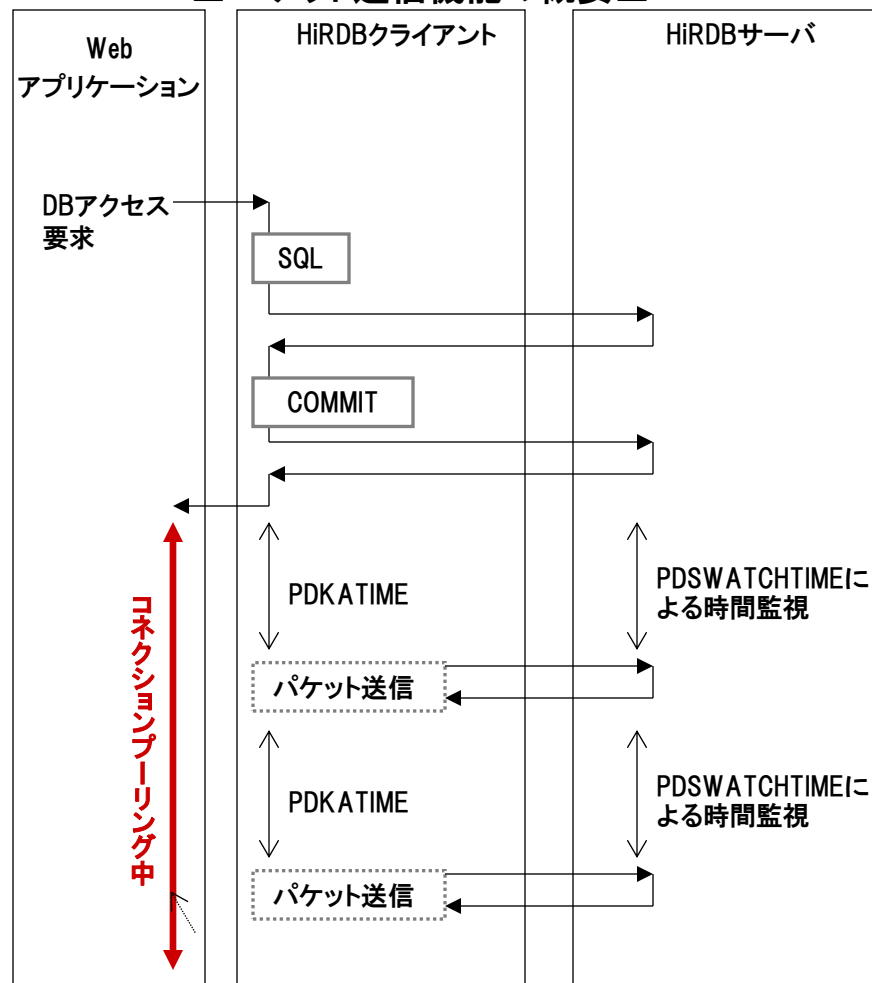
【コネクションプーリング使用時の設定例】

PDKALVL=2:定期的にパケットを送信し、
PDSWATCHTIMEの監視時間をリセット

PDKATIME=3000:パケット送信間隔(秒)

PDSWATCHTIME=3600:HiRDBサーバ側の監視時間(秒)

■パケット送信機能の概要■



バージョン09-50で、HiRDBクライアントが推奨モードの場合、PDKATIMEの省略値が600になりました。

2-1-7 ユティリティの実行監視時間(1)

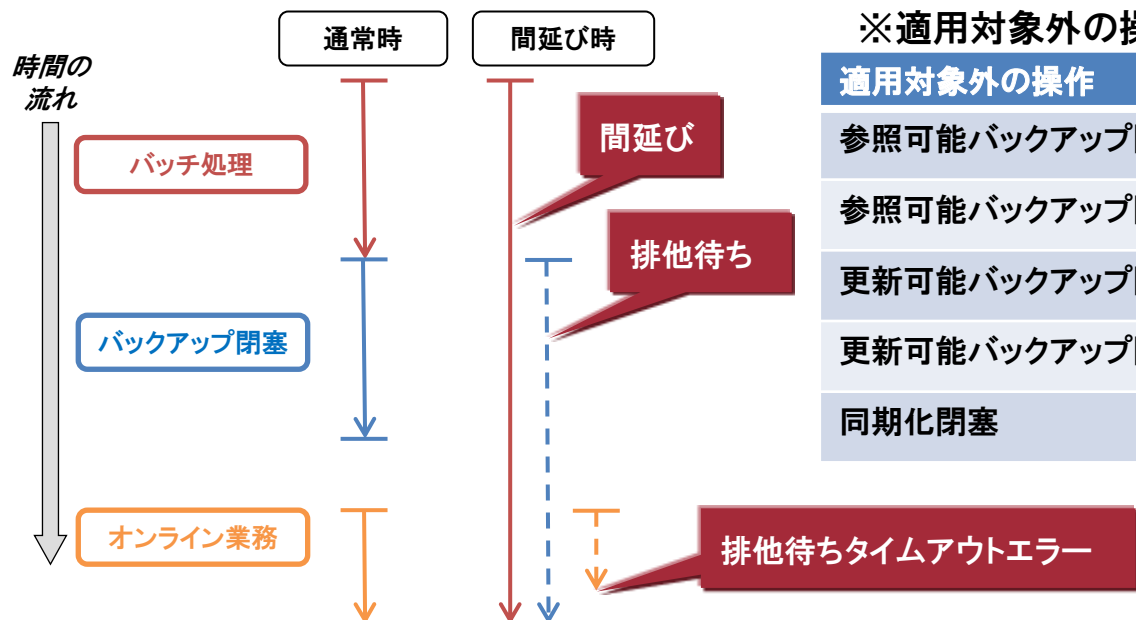
New

現象

直前のバッチ処理が間延びした影響で、バックアップ閉塞(pdhold -b)が長時間排他待ちして終了しなかった。この結果、同一RDエリアに対する後続のトランザクションが排他待ちタイムアウトエラーとなった。

原因

UAPやコマンド・ユティリティは、基本的にはpd_lck_wait_timeoutで排他待ち時間を監視する。しかし、バックアップ閉塞はpd_lck_wait_timeoutの適用対象外(※)である。このため、直前のバッチ処理が終了するまでは、バックアップ閉塞は排他待ちし続ける。この結果、後続のトランザクションはpd_lck_wait_timeoutの適用対象であるため、指定時間経過後に排他待ちタイムアウトエラーとなった。



※適用対象外の操作は下記

適用対象外の操作	対応するコマンド
参照可能バックアップ閉塞	pdhold -b指定
参照可能バックアップ閉塞(更新WAITモード)	pdhold -b, -w指定
更新可能バックアップ閉塞	pdhold -b, -u指定
更新可能バックアップ閉塞(WAITモード)	pdhold -b, -w, -u指定
同期化閉塞	pdhold -s指定



コツ

■ v0950より前のHiRDBを使用している場合

長時間排他待ちしているバックアップ閉塞をすぐに終了させたい場合は、pdcancelコマンドでシングルサーバプロセス, 又はバックエンドサーバプロセスを終了してください。なお、pdcancelコマンドの-uオプションのUAP識別子には文字列pdholdを指定してください。

■ v0950以降のHiRDBを使用している場合

バックアップ閉塞または同期化閉塞の実行時間(排他待ちを含む)をユティリティごとに変更する場合は、各ユティリティで-Wオプションを指定してください。ユティリティ全般に対して同一時間で監視する場合は、システム共通定義pd_cmd_exec_timeオペランドを指定してください。

2-1-9 ファイアウォールを設置する場合

現象

HiRDB/シングルサーバ側にファイアウォールが設定されている場合、クライアント環境変数PDHOSTとPDNAMEPORTに設定したIPアドレスとポートを通過するように設定したが、HiRDBに接続できなかった。

原因

クライアント接続用のポートは、デフォルトではOSの割り当てる空ポート番号を使う。たまたま割り当てられたポートが通過させるようになっていなかったため、HiRDBに接続できなかった。

コツ

🕒 HiRDBサーバ側にファイアウォールが設定されている場合は、クライアント環境変数PDHOSTとPDNAMEPORTだけではなく、HiRDBシステム定義pd service portも設定してクライアント接続用のポートを固定し、そのポートも通過させるようにしてください。

■クライアント環境変数

PDHOST=hirdb01(172.16.0.10)

PDNAMEPORT=20000

PDCLTRCVPORT=30000(*1)

■クライアント側 ファイアウォールの設定■(*1)

方向:受信

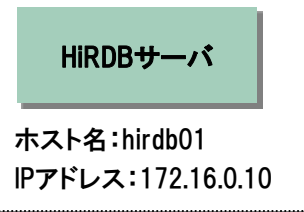
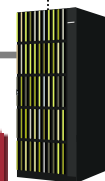
通過させるIPアドレス:192.168.0.10

通過させるポート:30000

■HiRDBシステム定義

set pd_name_port=20000

set pd_service_port=20001



■サーバ側 ファイアウォールの設定■

方向:受信

通過させるIPアドレス:172.16.0.10

通過させるポート:20000、20001

(*1):クライアント側にファイアウォールがある場合に指定してください。

HiRDB/パラレルサーバの場合

HiRDB/パラレルサーバを構成するサーバ間にファイアウォールを設置することはできません。

開発環境などで、意図せずファイアウォールを設置してしまい、SQLが突如エラーとなった事例があります。この場合、通信に関連するエラーとなるため、原因判明に長時間を要してしまう場合もあります。ファイアウォールの設定には、十分ご注意ください。

Windowsの場合

Windows Firewallでは、24時間通信がないと通信が切断されてしまいます。

24時間以上未使用だったコネクションを使ってSQLを投げるとWindows Firewallで遮断された事例があります。この場合、PDKALVLに1または2を指定し、HiRDBクライアントからHiRDBサーバに対して、定期的にパケットを送信するようにしてください。PDSWATCHTIMEに監視時間を設定している場合は、PDKALVLに2を指定してください。

 バージョン09-50で、PDKALVLの省略値が2(HiRDBクライアントが推奨モードの場合)になりました。ただし、シングルスレッドのHiRDBクライアントライブラリでは、PDKALVLの指定は無効です。

NAT^(*)によるアドレス変換

NATによるアドレス変換は、1対1の変換だけサポートしています。NAPT(IPマスカレード)のような、グローバルIPアドレスとローカルIPアドレスを1対複数で変換する機能には対応していません。

また、NATによるアドレス変換を設置している場合は、HiRDBシステム定義pd_service_portの設定だけでなく、HiRDBの高速接続機能を使用する設定にしてください。

→クライアント環境変数PDSERVICEGRP、PDSERVICEPORT、PDSRVTYPEなどの設定が必要となります。詳細は、マニュアル「システム導入・設計ガイド」の「ファイアウォールやNATが設置されている場合の設定」を参照してください。

(*)：NAT=「Network Address Translation」の略語。組織内でのみ通用するIPアドレス(ローカルアドレス)と、インターネット上のアドレス(グローバルアドレス)を透過的に相互変換する。

現象 HiRDB/パラレルサーバのプロセスの異常終了回数監視機能を使用した環境で、BESの異常終了多発により該当ユニットが終了したが、正常なFESユニットまで終了してしまった。

プロセスの異常終了回数監視機能

サーバプロセスの異常終了が多発すると、新たなサービスが受け付けられない状態になることがあります。しかし、サーバプロセスの異常終了だけでは、HiRDBは異常終了しないため、実質オンライン停止状態になります。

この状態を回避するため、HiRDBには「プロセスの異常終了回数監視機能」があります。サーバプロセスの異常終了回数が、一定時間内にHiRDBシステム定義pd_down_watch_procの値を超えた場合、HiRDBユニットは異常終了し、系切り替えを行います。

pd_down_watch_procは、システム共通定義(pdsys)とユニット制御情報定義(pdutsys)に指定できます。

原因 プロセスの異常終了回数の監視機能をFESが存在するユニットにも設定していたため、BESの異常終了に伴う内部的なFESの終了をも監視対象としてしまい、正常なユニットが終了してしまった。

コツ  pd_down_watch_procオペランドは、FESが存在するユニットには、指定しないでください。

現象

Windowsにおいて、数百ギガバイト程度のファイルを入出力するデータベース複写ユティリティ(pdcopy)やデータベース再編成ユティリティ(pdrorg)を実行すると、非常に長い時間を要した。CPU利用率は80%以上と高いが、ユティリティ実行時間の経過とともに、時間あたりのディスクI/O回数が減少する状況に陥っている。

原因

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(privileged time)が上がり、ディスクI/O処理が遅延したため。

コツ

💡 数百ギガバイトのファイルにデータを入出力する場合、ユティリティがファイルにアクセスするときに入出力に使用するバッファ長(HiRDBシステム共通定義pd utl file buff size)に、1メガバイト程度を指定してください。大きい値を指定すると、ファイルへの入出力回数を削減できます。



バージョン09-50で、pd utl file buff sizeの省略値が1メガバイトになりました。

対策

pd_utl_file_buff_sizeをデフォルト値(32キロバイト)から1メガバイトに拡大し、OSの負荷を下げてください。

New

現象

HiRDB起動後に初めてクライアント環境定義PDCWAITTIME設定値を超えて
HiRDBクライアントがタイムアウトした場合、全ユニットでI/Oが遅延し性能が落ちてしまう。

原因

HiRDB/パラレルサーバでpd_clt_waittime_over_dump_level=all(省略値)にすると、クライアントタイムアウト事象発生時に全ユニットで共用メモリダンプを出力(初回のみ)するため、出力先(運用ディレクトリ配下)へのI/O要求が遅れ、HiRDBサーバの処理が遅延する。

コツ



pd clt waittime over dump levelオペランドにshm fesonlyを指定してください。
shm_fesonlyを指定しても必要最低限のトラブルシュート情報は取得されます。



バージョン09-60で、pd clt waittime over dump levelの省略値が
shm fesonlyになりました。

2. システム設計時のコツ

2.1 HiRDBパラメタ関連

2.2 スキーマ設計関連

2.3 ネットワーク関連

2.4 クラスタ関連

2.5 運用設計関連

2.6 OS関連

2.7 ディスク関連

2.8 メモリ関連

- コツ1 : データを削除しない表に対して、セグメント数を指定した
SEGMENT REUSEは使用しない！



2-2-1 ~ 2-2-2

- コツ2 : 256バイト以上のVARCHAR列には、NO SPLITを指定しよう！



2-2-3

- コツ3 : 性能を確保するためには、RDエリアの設計も大事！

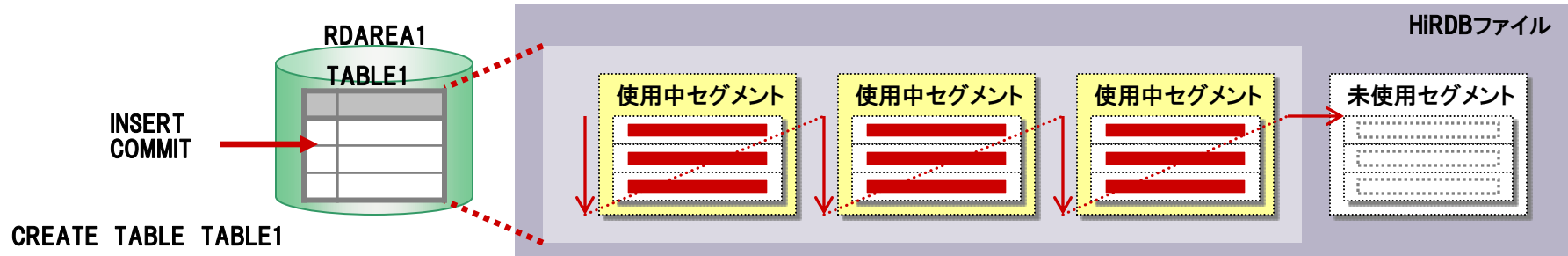


2-2-4 ~ 2-2-5

2-2-1 空き領域の再利用機能(SEGMENT REUSE)(1) **HITACHI** Inspire the Next

現象 データの追加を継続していると、ある時点からデータ格納処理性能が遅延し始めた。

原因 データを削除しない(空き領域が存在しない)表であるにもかかわらず、セグメント数を指定したSEGMENT REUSEを使用していたため、使用するセグメント数が指定した値を超えた段階で新たな未使用セグメントを確保する前に、空き領域を探す処理が動作していた。



```
CREATE TABLE TABLE1
(C1 INTEGER, C2 INTEGER)
IN RDAREA1
SEGMENT REUSE 3
OPTION 3
```

→ : 空き領域を探す処理

— : データ格納領域

- - - : 未使用領域

 バージョン09-50で、「SEGMENT REUSE」の指定有無に関係なく、空き領域の再利用機能(セグメント数を指定なし)を適用するようになりました。

解説

セグメント数を指定したSEGMENT REUSEでは、使用セグメント数が指定値に達した時点で、新たに未使用セグメント確保するのではなく、先頭のセグメントから空き領域を探します。ただし、使用中セグメント内に空き領域が存在しなかった場合は、新たに未使用セグメントを確保し、データを格納します。これ以降、新たな未使用セグメントを確保する前には、必ず先頭からすべての使用中セグメントに対して、空き領域を探す処理が動作します。

2-2-2 空き領域の再利用機能(SEGMENT REUSE)(2) HITACHI Inspire the Next

 データを削除しない表に対して、セグメント数を指定したSEGMENT REUSEは使用しないでください。

コツ データを格納した順番で削除するなど、連続した空き領域が発生する表の場合は、SEGMENT REUSEを使用することで、空き領域を再利用でき格納効率が上がります。また、データ配置の乱れも防げます。

SEGMENT REUSE指定の注意事項①

SEGMENT REUSEのセグメント数に、有効なデータ量を格納するためのセグメント数よりも少ない値を指定した場合も、同様の現象となります。
SEGMENT REUSEのセグメント数は、有効なデータ量を見積もった上で設定してください。

SEGMENT REUSE指定の注意事項②

SEGMENT REUSEを指定した表に対して、データを大量削除した後に、別のデータを挿入するという処理を行う場合、大量削除後に一旦commitしてください。
同一ランザクション内の削除により、空き状態となったページは、再利用の対象になりませんので、注意してください。

SEGMENT REUSE指定の注意事項③

1行のデータがページを超えて分岐した場合、再利用されませんので注意してください。
なお、NULL値をデータありに更新、可変長列の実長が長くなるように更新、または、実長がページ長を超えるBINARY型データを挿入・更新した場合などに、分岐する場合があります。

この場合でも、空き領域の再利用機能を使用し、格納効率を向上させるオプション機能をバージョン09-04でサポートしています。
詳細は、マニュアル「SQLリファレンス」のCREATE TABLEのSEGMENT REUSEオプションの再利用オプションを参照してください。

 なお、バージョン09-50のCREATE TABLE文では、再利用オプション値として推奨値である3を常に仮定し、再利用オプションは非公開としました。

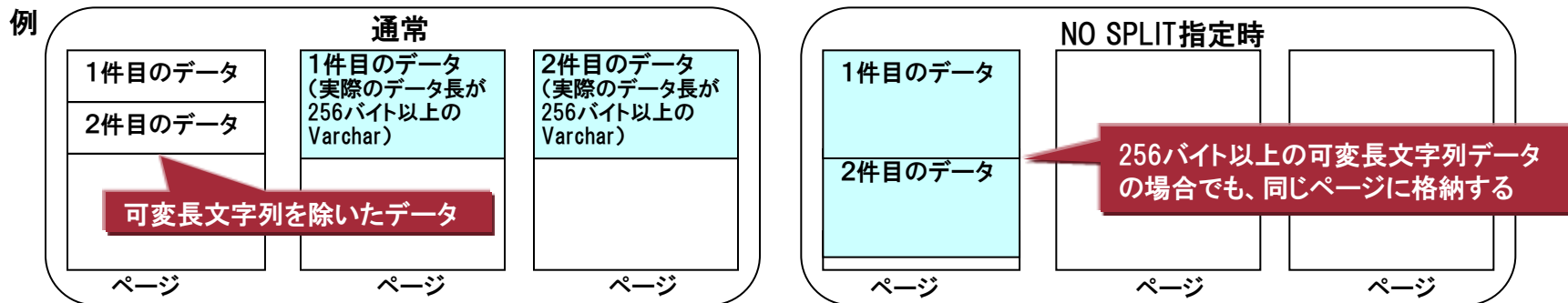
2-2-3 256バイト以上の可変長文字列に対するノースプリットオプション

現象 ユーザのデータ量と比較して実際のRDエリアの使用量が大幅に多い。

原因 可変長文字列型の列に格納するデータ長が256バイト以上あったため、同じ行の他の列とは異なるページに格納された。1行のデータが複数ページに格納されることで、元のデータ量よりもRDエリアの使用量が大幅に増加した。

コツ 💡 データ長が256バイト以上になる可変長文字列型の列データを他の列と同じページに格納するには、ノースプリットオプションを適用(CREATE TABLE時、該当列にNO SPLITを指定)します。1行のデータの格納ページを減らすことで、RDエリアの使用量を削減できます。

💡 HiRDB Version 9 09-50では表定義時、定義長が256バイト以上の可変長文字列に対して、省略値でNO SPLIT指定を適用するようにしました。ただし、旧バージョンから09-50にバージョンアップした表は、表定義内容は変わらないため、NO SPLIT指定を旧バージョンで未指定だった列に適用するには「対策」に記載の対応が必要です。



対策 NO SPLITを適用していない表にNO SPLITを適用するには、まず表のデータをアンロードして退避します。次にNO SPLITを指定し表の再定義をするか、表にデータが無い状態で256バイト以上の可変長文字列に対してALTER TABLE文でNO SPLITへ列属性を変更します。その後、表のデータをリロードして回復してください。

コツ ①

 ユーザ用RDエリアのセグメントサイズに小さい値を指定しないでください。

事例 ①

データ挿入処理で性能低下を招き、システム全体の性能に悪影響を与えることがあります。

コツ ②

 初期設定ユーティリティ(pdinit)および構成変更ユーティリティ(pdmod)で作成するRDエリアのページサイズは、表の行長を考慮して決定してください。

事例 ②

表にデータを追加する場合、実際の行の長さがRDエリアのページサイズを超えると、KFPA11809-Iエラー(行が長すぎて格納できない)になります。

セグメントサイズ、ページサイズおよびRDエリア設計の考え方については、**2-2-5** に示します。

2-2-5 RDエリアの設計の考え方

■セグメントサイズの考え方

- (1)通常、RDエリア容量の総ページ数の1/10～1/100程度のサイズがお勧めです。
また、複数のRDエリアで似たサイズは揃えておくとう管理しやすいです。
- (2)大量のデータ挿入をする表では、できるだけセグメントサイズを大きく(最大は16000)し、セグメント確保回数を減らすと効率が良いです。

■ページサイズの考え方

- (1)表のページは、表の行長とページアクセスの競合を考慮し、決定してください。
 - ・検索を優先する場合、ページサイズを大きくしてください。
 - ・同一ページへの更新が多い場合、ページサイズは8キロバイト程度と1ページの格納件数を少なくすることでアクセス時の競合を減らしてください。
- (2)インデクスのページは、入出力の効率の良い4096～8192バイト程度としてください。

■RDエリア設計の考え方

RDエリアを設計する際、1RDエリアに1つの表またはインデクス、1HiRDBファイルシステム領域に1RDエリアを配置する構成とすると、監視やチューニングをきめ細かく行えるため、お勧めです。
しかし、RDエリア数、ファイル数が多いとファイル数等の制限あるいは、運用面の都合により、実現が難しいこともあります。そのような場合は、性質の似たものでまとめると良いです(競合の考慮も忘れないでください)。

※RDエリアの設計は業務の内容にも関連しますので、HiRDB技術資料「業務別 HiRDB設計のコツ」で詳しく解説しています。

HiRDB技術資料のURL

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

2-2-6 分割表におけるユニークインデクスの定義

現象

既存表を横分割したところ、既存のユニークインデクス(プライマリキー)が定義できなくなった。

原因

横分割表にユニークインデクスを定義するには、分割キーを含める必要があるため。

コツ

 横分割表にユニークインデクスが必要な場合は、分割キーをインデクス構成列に含めてください。



バージョン09-50で、インデクス構成列数の上限を16個から64個に拡大しました。

対策

以下を検討してください。

- 分割キーをインデクス構成列に追加することを検討してください。
- サーバ内横分割のみの場合は、非分割のインデクスを定義することを検討してください。



注意事項

- 分割キーを構成列に追加する場合は、アプリケーションで期待するユニークチェックとなっているか確認してください。
- 非分割のインデクスを定義すると、メンテナンス時間を短縮するための表の横分割の効果がなくなります。

現象 更新可能バックアップ閉塞を用いたバックアップファイルからのDBリストアができなかった。

原因 LOB列に対して、列回復制約(RECOVERY ALL:ログ取得モード)を定義し忘れた。
このため、更新可能バックアップ閉塞中にデータベースを更新した際の追加のシステムログが出力されず、このバックアップでは回復できなかった。

 デフォルトではLOBデータがシステムログに出力されないため、LOB列を回復する場合は、列回復制約を指定してください。

コツ 更新可能バックアップ閉塞を適用する場合は、LOB列の列回復制約をRECOVERY ALL (ログ取得モード)としてください。または、LOB列を含む表に対して、参照可能バックアップ閉塞を用いたバックアップファイルからのDBリストアを行うことを検討してください。
また、データ連動する場合も、抽出対象のLOB列の列回復制約をRECOVERY ALLにしてください。

注意事項

列回復制約をALLにすると更新ログ量が増加するため、システムログファイル容量も増強する必要があります。

2. システム設計時のコツ

2.1 HiRDBパラメタ関連

2.2 スキーマ設計関連

2.3 ネットワーク関連

2.4 クラスタ関連

2.5 運用設計関連

2.6 OS関連

2.7 ディスク関連

2.8 メモリ関連

現象

大規模で高トラフィックのパラレル環境でMGRユニットからのアライブ通信がエラーになり、FESまたはBESがKFPS05289-E(STOP(A))になった。
Linuxのメッセージ:nf_conntrack: table full, dropping packet.

原因

nf_conntrackモジュール(*1)が管理するセッション管理テーブルがいっぱいになり、新規セッションパケットが破棄されたため。(*1):セッション管理やiptablesによるパケットフィルタリングを行う。

コツ

🕒 通常は考慮不要ですが、大規模で高トラフィックなシステムでは、以下の設定を検討してください。

■セッション管理テーブルの最大エントリ数(net.netfilter.nf_conntrack_max)に、HIRDBやOSで必要なポート数を設定する。

また、TIME_WAIT状態のTCPコネクションが多くなるので、TIME WAIT時間(デフォルト120秒: net.netfilter.nf_conntrack_tcp_timeout_time_wait)を短くする。

対策

この現象が発生した場合は、上記のパラメタを設定するか、以下の対策を行ってください。

■nf_conntrackモジュールによるセッション管理やiptablesによるパケットフィルタリングをしていない場合は、iptablesサービスを停止する。

<iptablesサービス停止方法>

```
# service iptables stop
# chkconfig iptables off
# chkconfig ip6tables off
```

2. システム設計時のコツ

2.1 HiRDBパラメタ関連

2.2 スキーマ設計関連

2.3 ネットワーク関連

2.4 クラスタ関連

2.5 運用設計関連

2.6 OS関連

2.7 ディスク関連

2.8 メモリ関連

コツ

 クラスタ(MSFC)停止時には、HiRDB/ClusterServiceも停止するよう、HA Toolkit Extensionの停止用バッチファイルに、HiRDB/ClusterServiceの停止処理を入れてください。

事例

クラスタ起動時にHiRDB/ClusterServiceを起動しますが、稼動中だと起動済エラーとなります。起動済エラーとなると%PDDIR%\CONF\pdshastr.batからpdstartコマンドが実行されます。このpdstartコマンドは、HA Toolkit Extensionのアカウント(ローカルシステムアカウント)で実行されますが、HiRDBプロセスサーバ(pdprcd)の実行アカウントと異なるため、pdprcdが異常終了するという事例があります。

HA Toolkit Extensionの停止用バッチファイル(hatestop.bat)例

【相互系切り替え環境の場合】

```
start C:\HATkEx\HAMon\bin\hatesbystp UNT1
start C:\HATkEx\HAMon\bin\hatesbystp UNT2
@net stop HiRDBClusterServiceUNT1
@net stop HiRDBClusterServiceUNT2
exit
```



注意 事項

強制停止後やユニットダウン後のHiRDB再開始時には、最後に取得したシンクポイントダンプ以降のすべてのシステムログファイルを入力します。このためHiRDBの再開始時の起動時間が正常開始時よりも長くなり、以下の監視タイマ値を超過する可能性があります。

- ①HiRDBのpd_system_complete_wait_time(pdstartコマンド完了待ち時間)
- ②HA Toolkit Extensionのactpatrol(サーバ起動監視時間つまりHiRDBの起動時間)
- ③MSFCの待ちのタイムアウト値(リソースつまりHiRDB/ClusterServiceがオンライン・オフライン終了するまでの最大時間)

コツ

💡 現象を把握し易くするために、以下の関係となるように設定してください。

② > ① or ③ > ①

ただし、HA Toolkit Extensionのactpatrolは、指定可能な最大値が600秒であるため、actpatrolには0(監視しない)を指定(*1,*2)し、MSFCの待ちのタイムアウトで現実的な値を設定してください。

(*1):pd_system_complete_wait_timeの指定範囲が610～3600秒のため。

(*2):actpatrolは省略せずに0を指定してください。

MSFCの待ちのタイムアウト値を増やしても、actpatrol省略した場合、60秒が仮定されます。

2. システム設計時のコツ

2.1 HiRDBパラメタ関連

2.2 スキーマ設計関連

2.3 ネットワーク関連

2.4 クラスタ関連

2.5 運用設計関連

2.6 OS関連

2.7 ディスク関連

2.8 メモリ関連

- コツ1 : 回復するRDエリアの組み合わせを決めておこう！
▶ 2-5-1
- コツ2 : システム用RDエリアのバックアップについても確認しておこう！
▶ 2-5-2
- コツ3 : ユティリティをオンライン業務と同時に実行する場合は、
オンライン業務性能への影響を最小限に抑えよう！
▶ 2-5-5 ~ 2-5-6
- コツ4 : 運用ディレクトリがあるディスクの空き容量を確保しておこう！
▶ 2-5-7
- コツ5 : システムログは自動ログアンロード機能を適用しよう！
▶ 2-5-10

2-5-1 同時にバックアップを取得する必要があるRDエリア

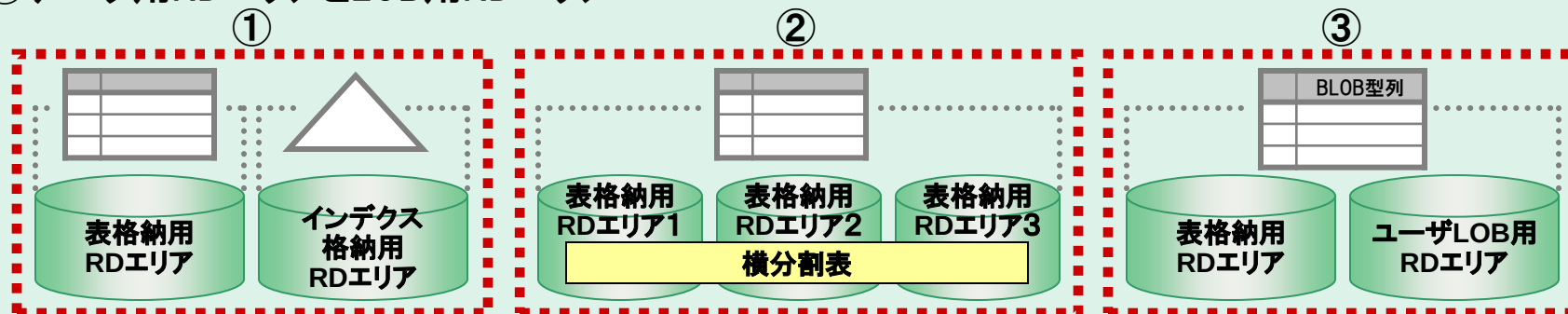
現象 表のデータが格納されているRDエリアを、バックアップファイルを使って回復した後、表にアクセスしたところ、RDエリアが閉塞した。(KFPH00308-EおよびKFPH00306-Eが出力)

原因 別のRDエリアに格納されているインデックスの情報と、不整合が発生したため。

コツ 🕒 バックアップファイルを使って、RDエリア単位の回復を行う場合は、そのRDエリアに格納されている情報と、関連のあるRDエリアを同時に回復するようにしてください。

同時にバックアップを取得する必要があるRDエリア例

- ① データ用RDエリアとインデックス用RDエリア
- ② 横分割表の場合、すべてのデータ用格納RDエリア
- ③ データ用RDエリアとLOB用RDエリア



関連するRDエリアは例に示したものだけではないので、詳細はマニュアル「HiRDBシステム運用ガイド」の「同時にバックアップを取得する必要があるRDエリア」を参照してください。

また、pdrdreflsコマンドで関連するRDエリアを確認することができます。

⚠️ ただし、ディクショナリ表の検索が多くなるので、負荷が高い時には実行しないでください。

2-5-2 システム用RDエリアのバックアップ

現象

定義系SQLを実行(インデクス追加)後、インデクス用RDエリアをバックアップファイルを使って回復した後、表にアクセスしたところ、RDエリアが閉塞した。

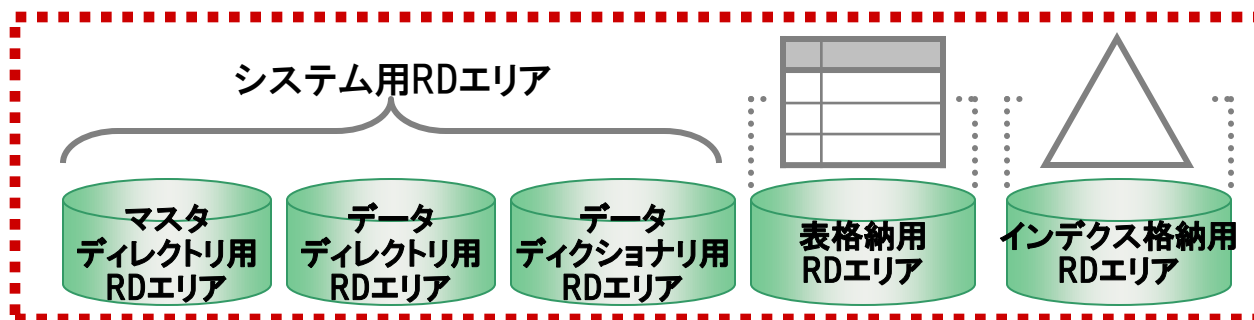
原因

定義系SQL実行後にユーザ用RDエリアを回復する場合は、システム用RDエリアも同時に回復して整合性を取る必要があるが、システム用RDエリアのバックアップを取得せず、システム用RDエリアを回復していなかったため、インデクスの内部キーとシステム用RDエリアが保持する内部キーが不一致となり、RDエリアが閉塞した。

コツ ①

💡 **定義系SQL実行前後で、処理対象のユーザ用RDエリアだけでなく、その処理によって更新されるシステム用RDエリアのバックアップも忘れずに取得してください。**

■ 定義系SQLを実行した場合に同時にバックアップを取得する必要があるRDエリアの例



コツ ②

💡 システム用RDエリアを含めた**回復の手順を、運用設計で確認**しておいてください。

⚠ 注意事項

pdrdreflsコマンドでは、ユーザ用RDエリアの関連は確認できますが、システム用RDエリアは対象外です。

現象

マシンのルートディレクトリが格納されたディスクが壊れたため、AIXのシステムバックアップ機能を使って取得したバックアップから回復したが、その後のHiRDB起動がエラーとなった。

原因

HiRDBの起動には、HiRDBの通信情報の格納先である/dev/HiRDB/pthディレクトリが必要であるが、このディレクトリはバックアップ対象ではなかったため、回復されなかった。

コツ

🕒 UNIXの場合、HiRDB運用ディレクトリ下以外にも、HiRDBの起動に必要なファイルやディレクトリが存在しますので、それらもバックアップ対象としてください。

HiRDBの起動に必要なファイルについては、マニュアル「システム導入・設計ガイド」の「HiRDBのディレクトリ及びファイル構成」を参照してください。

対策

HiRDBを再度セットアップしてください。
セットアップ処理の延長で、/dev/HiRDB/pthディレクトリが作成されます。

現象 Windows Server 2008でHiRDBの起動が失敗した。

原因 Windows Server 2008で標準的に起動されるサービスApplication Experience Lookup Service(AELookupSvc)^(*)を停止したため、OSがファイルをロックし、HiRDBの起動が失敗した。

(*)： Windows Server 2012でも標準的に起動されます。

Application Experience Lookup Service(AELookupSvc)

アプリケーションの起動時に、そのアプリケーションのアプリケーション互換性検索要求を処理します。

 アプリケーション互換性のソフトウェアが更新されるには、このサービスがアクティブである必要があります。

コツ  Application Experience Lookup Service(AELookupSvc)は有効にしておいてください。

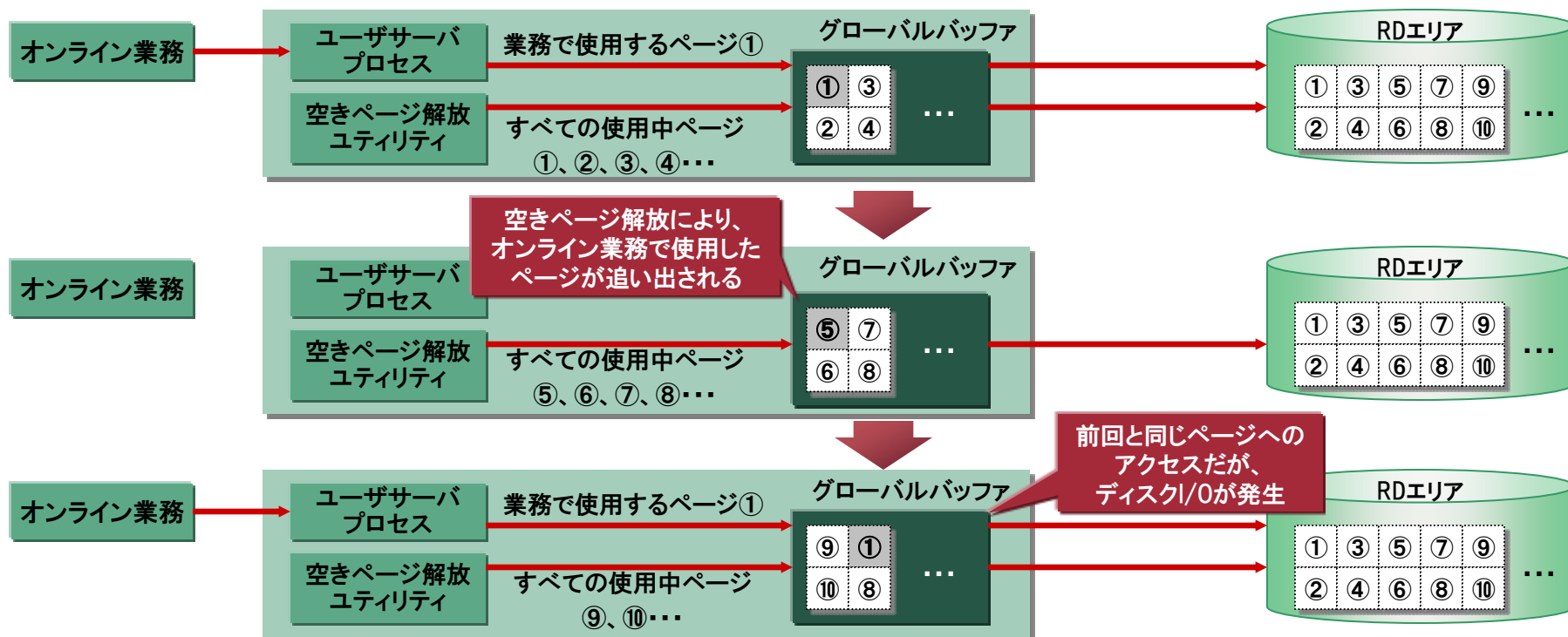
2-5-5 空きページ解放ユティリティの実行(1)

現象

オンライン業務と並行して空きページ解放ユティリティ(pdreclaim)を実行したところ、オンライン業務性能が劣化した。

原因

空きページ解放ユティリティでは、すべての使用中ページを参照するため、オンライン業務で使用するページをグローバルバッファ上から追い出してしまい、オンライン業務の延長で発生するディスクI/O回数が増加したことで、業務性能に影響を及ぼした。



コツ

 空きページ解放と業務アプリケーションを並行実行する場合に、オンライン性能を劣化させないために、空きページ解放ユティリティ(pdreclaim)のoption文のglobalbuffer lruにnoを指定してください。

 バージョン09-50で、globalbuffer lruの省略値がnoになりました。

また、オンライン停止中に実行した空きページ解放ユティリティ終了後の業務開始直後の性能劣化も抑えることができます。

noを指定した場合、空きページ解放ユティリティ(pdreclaim)が読み込んだページは常に最古のページとして扱い、グローバルバッファ不足時には最初に追い出されるページとなります。同様に、データベース状態解析ユティリティ(pddbst)実行と業務アプリケーションを並行実行する場合に、オンライン性能を劣化させないためには、-bオプションを指定(pddbst -b)します。

 バージョン09-50で、省略値で-bオプションを仮定(-sオプションを指定していない場合)しました。

注意事項

空きページ解放の実行間隔について、検討してください。

空きページ解放の効果が低いにもかかわらず、頻繁に実行することは避けてください。

空きページ解放ユティリティの効果(解放したページ数など)は、ユティリティの実行結果で確認できます。

現象

HiRDBが連続してダウンするなどの現象が発生した後、原因を対処し再起動を試みたが、起動に失敗した。

原因

HiRDBでは、通常24時間ごとに、出力日時の古いトラブルシュート情報を削除している。しかし、HiRDBシステムやユーザサーバが連続してダウンするなどの現象が発生した場合、トラブルシュート情報によってHiRDB運用ディレクトリの容量を圧迫してしまう場合があり、HiRDB運用ディレクトリの容量不足により、HiRDBの起動に失敗した。

コツ

 pdinfogetコマンドでトラブルシュート情報を取得して退避した後、pdccspoolコマンドで不要なトラブルシュート情報を削除するようにしてください。

コマンド実行例

```
pdccspool -d 0 -k all
```

pdinfogetコマンドの実行方法と取得可能な情報については、**付録B. 5** を参照してください。

AIXでの注意事項

AIXでは、アタッチしている共用メモリの情報までCOREファイルに出力するため、COREファイルのサイズが巨大(数百メガバイトの事例もあり)になりますので、特に注意してください。

なお、環境変数CORE_NOSHMを指定することにより、共用メモリの情報を出力抑止することができます^(*1)。

HiRDBでは、共用メモリは別途ダンプ出力しますので、共用メモリの情報が各プロセスのCOREファイルに含まれている必要はありません。

よって、HiRDBシステム定義のシステム共通定義(pdsys)に、以下の環境変数を指定してください。

【pdsys指定例】

```
putenv CORE_NOSHM ""
```

 バージョン09-50で、環境変数CORE_NOSHM^(*2) は自動設定するようにし、システム共通定義への指定を不要にしました。

また、上記で設定した環境変数はHiRDBコマンド実行環境にも設定する必要があります。

(*1):前提条件として、システム属性(sys0)のfullcoreパラメタがtrueの必要があります。

(*2):その他、PSALLOC、NODISCLAIMIについても自動設定するようにしました。

ユーザサーバプロセスの異常終了の場合

ユーザサーバプロセスがダウンした場合、syslog(Windowsの場合はイベントログ)やクライアントのエラーログなどから原因がすぐに判明しない時は、ダウンしたプロセスのCOREファイルを退避してください。(pdinfogetコマンドで取得できます。)

COREファイルの出力先は、
「HiRDB運用ディレクトリ/spool/save/サーバ名n」(n:1～3)です。

COREファイルは、各サーバ毎に3つまでしか退避されないため、すぐに上書きされてしまう可能性があります。ダウンの要因がすぐに特定できない場合などは、できるだけ早くCOREファイルを退避してください。

系切り替え構成の場合

系切り替え構成の場合、現用系と予備系の両方のトラブルシュート情報を取得してください。
また、HAモニタなどクラスタ製品の情報も併せて取得をお願いします。

HiRDB/パラレルサーバの場合

HiRDB/パラレルサーバでは、syslog(Windowsの場合はイベントログ)のメッセージなど、マネージャユニットにしか存在しないトラブルシュート情報があります。したがって、非マネージャユニットがダウンした場合は、ダウンしたユニットのトラブルシュート情報だけでなく、マネージャユニットのトラブルシュート情報も取得してください。

自動ログアンロード機能の推奨

HiRDBでは、システムログファイルを循環利用していますが、一度使用したシステムログファイルを再び使用可能にするためには、ログ情報をアンロードする必要があります。システムログファイルのアンロード処理を実行しなかったために、使用可能なシステムログファイルがなくなり、システムダウンに陥るケースもあります。

これを防ぐために、システムログファイルが満杯になってスワップした時点で、HiRDBが自動的にアンロードする「自動ログアンロード機能」の適応を推奨します。

【設定方法】

HiRDBシステム定義pd_log_auto_unload_pathに、アンロードログファイルの出力先ディレクトリ名を指定します。

ただし、アンロードログファイル出力先ディレクトリの容量不足が発生した場合は、自動ログアンロード機能は停止します。アンロード先のディスク容量にも、注意してください。

システムログファイルの空き容量監視

メッセージによる監視を行う場合は、システムログファイルの空き容量が警告値未満になった際に、警告メッセージをsyslog(Windowsの場合はイベントログ)へ出力する「システムログファイル空き容量監視機能」を適用してください。

長大トランザクション実行時の注意事項

自動ログアンロード機能を適応している場合でも、長大トランザクションを実行すると、シンクポイントが有効化されないため、満杯になったシステムログファイルがアンロードできる状態にはなりません。

1つの長大トランザクションですべてのシステムログファイルを使い切ってしまう、システムダウンに陥るケースもあります。これを防ぐために、長大トランザクションを実行する場合は、ログレスモードでの実行も検討してください。ただし、ログレスモードで実行した場合は、システムログを使ったDB回復ができませんので、必ず実行前後でバックアップを取得するようにしてください。

システムログファイルの容量不足が発生したら

システムログファイルの容量不足により、システムがダウンしてしまった場合は、再起動するために使用可能なシステムログファイルを用意する必要があります。

対処方法の詳細については、マニュアル「システム運用ガイド」の「障害が発生したときの対処方法」の「システムログファイルの容量不足によってHiRDB(ユニット)が異常終了したときの対処方法」を参照してください。

2. システム設計時のコツ

2.1 HiRDBパラメタ関連

2.2 スキーマ設計関連

2.3 ネットワーク関連

2.4 クラスタ関連

2.5 運用設計関連

2.6 OS関連

2.7 ディスク関連

2.8 メモリ関連

現象

Windowsのクラスタ環境において、グローバルバッファサイズを2ギガバイトから20ギガバイトに拡張したところ、HA Toolkit ExtensionがHiRDBのスローダウンを検知して系切り替えが発生した。

原因

NTFSキャッシュのフラッシュが発生すると、フラッシュが完了するまでボリューム制御ブロックが排他される。フラッシュされるデータ量が多い(グローバルバッファサイズを2ギガバイトから20ギガバイトに拡張した影響)と、プログラムが長時間にわたってボリューム制御ブロックの排他待ちとなり、処理の沈み込みが発生することがある。

キャッシュフラッシュを引き起こす操作

- ディスクの管理の操作
- オンライン稼働中のWindowsログオン/ログオフ操作
- システム情報のSCSIを表示
- システム情報ファイルの出力(SCSIを含む場合)
- デフラグの実施
- chkdsk.exeの実行
- diskpart.exe、nfi.exe、fsutil.exe(MSからダウンロードできる非標準のツール)

コツ

 **共用メモリの割り当て先は、ページングファイルにすることを推奨**します。

ページングファイルは、通常のファイルに比べて、ファイルキャッシュのフラッシュの契機が少ないため、NTFSのファイルキャッシュのフラッシュの完了待ちによるスローダウンを軽減でき、トランザクションの遅延や系の切替えの発生を低減できます。



ページングファイルに割り当てる場合の注意事項

HiRDBで使用する共用メモリをページングファイル(仮想メモリ)に割り当てる場合、仮想メモリの使用量を確認して、不足する場合は拡張してください。

対策 HiRDBが使用するメモリマップドファイルの割り当て先を、運用ディレクトリ下の通常ファイルから、ページングファイルに変更してください。

ページングファイルへの変更手順

1. HiRDBを正常終了し、HiRDBのサービスを停止します。
2. `pdntenv -shmfile page`
3. HiRDBおよびHiRDBのサービスを再開始します。

現象

I/Oスケジューリング方式の設定がデフォルト(cfq)のまま運用をしていて、I/O多重度(数百オーダー)が増加した際に、ディスクの応答遅延を繰り返し、長時間ダウンとなった。

原因

デフォルトの設定はcfqスケジューラになっています。
cfqスケジューラは、I/O多重度が高い場合、Read要求のレイテンシ性能(1つのI/O要求の処理にかかる時間)が著しく劣化する傾向があります。

コツ

💡 Linuxで新規システムを構築する場合、I/Oスケジューリング方式にdeadlineを設定してください。

本件の詳細については、LinuxマニュアルのI/Oスケジューリング方式で確認してください。

2. システム設計時のコツ

2.1 HiRDBパラメタ関連

2.2 スキーマ設計関連

2.3 ネットワーク関連

2.4 クラスタ関連

2.5 運用設計関連

2.6 OS関連

2.7 ディスク関連

2.8 メモリ関連

現象

Windows Server 2008 R2で、HiRDBのDB閉塞処理が沈み込む(数分から数十分掛かるようになる)という事例があります。

原因

日立ファイバチャネルアダプタの性能向上オプションPerformance Option (-p perf) enable (Windows用)の設定により、I/O性能が低下することがあります。

コツ

 **Performance Option (-p perf) は、I/O多重起動数が増えた場合に、IOPS性能劣化を軽減するオプションです。しかし、環境やI/O特性によりdisableに設定したほうが性能が向上する場合があります。**

対策

I/O性能が低下した場合、Windows OSの「電源プラン」の設定を「高パフォーマンス」に変更することで、I/O性能を改善できる可能性があります。
「高パフォーマンス」設定を実施できない場合、または実施してもI/O性能が改善しない場合は、Performance Option (-p perf) にdisableの設定を試してください。

OSの省電力モードを無効化する方法

[コントロールパネル]- [電源オプション]を開き、[お気に入りのプラン]をデフォルトの[バランス]から[高パフォーマンス]に変更します。[バランス]の場合は、省電力機能が有効になります。
[高パフォーマンス]に変更すると、省電力機能が使用されなくなるため、サーバの消費電力が多少増加します。

2. システム設計時のコツ

2.1 HiRDBパラメタ関連

2.2 スキーマ設計関連

2.3 ネットワーク関連

2.4 クラスタ関連

2.5 運用設計関連

2.6 OS関連

2.7 ディスク関連

2.8 メモリ関連

現象 共用メモリの確保に失敗し、HiRDBが起動できない。(UNIXの場合)

原因 HiRDBの起動時に確保するユニットコントローラ用共用メモリのサイズが、OSのオペレーティングシステムパラメタshmmax^(*)の値を超えていたため。

(*) : Solarisの場合はshminfo_shmmax

コツ  ユニットコントローラ用共用メモリは、1つのセグメントに確保します。OSのオペレーティングシステムパラメタshmmaxの値は、この計算値以上にしてください。

 グローバルバッファ用共用メモリは、HiRDBシステム定義SHMMAXの値でセグメントを分割します。OSのオペレーティングシステムパラメタshmmaxの値は、HiRDBシステム定義SHMMAXの値以上にしてください。

共用メモリサイズの見積もりについては、マニュアル「システム導入・設計ガイド」の「ユニットコントローラが使用する共用メモリの計算式」または「グローバルバッファが使用する共用メモリの計算式」を参照してください。

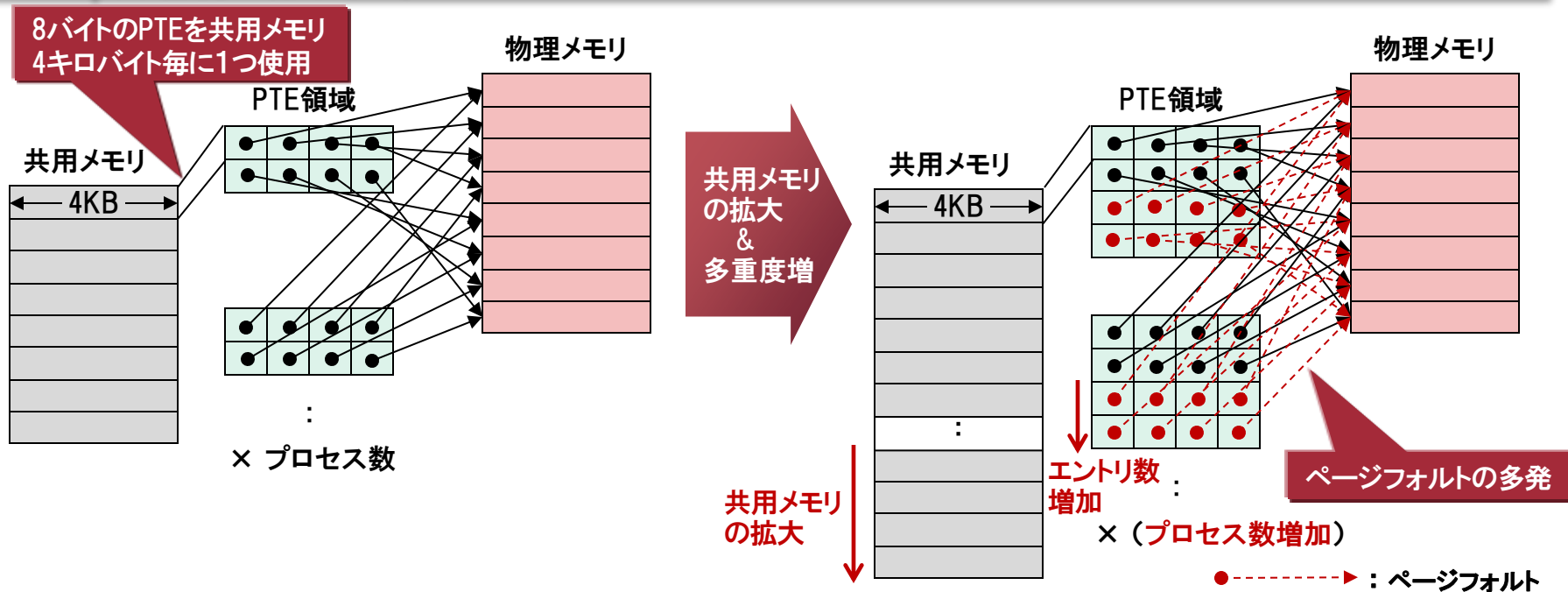
2-8-2 共用メモリのページサイズ(Linux)(1)

現象

排他制御用プールサイズ(pd_lck_pool_size)を32,000キロバイトから128,000キロバイトに拡大したところ、UAPの10多重同時実行で性能が1割程度低下した。

原因

Linuxでは、仮想メモリアドレスと物理メモリアドレスを対応付けるため、PTE(Page Table Entry)というOSの領域を用いて管理している。このPTEはプロセスごとに生成され、各プロセスの必要とするメモリサイズに応じた数分のエントリが生成される。8バイトのPTEを共用メモリ4キロバイト(通常の1ページ)毎に1つ使用するため、共用メモリの拡大でPTEのエントリ数が増加する。また、業務で使用するUAPの多重度が増加するとPTEの数も増加する。PTEの数、PTEのエントリ数が増加すると物理メモリ不足によりページフォルトが多発し、不要なページング処理が多くなり、性能が低下した。

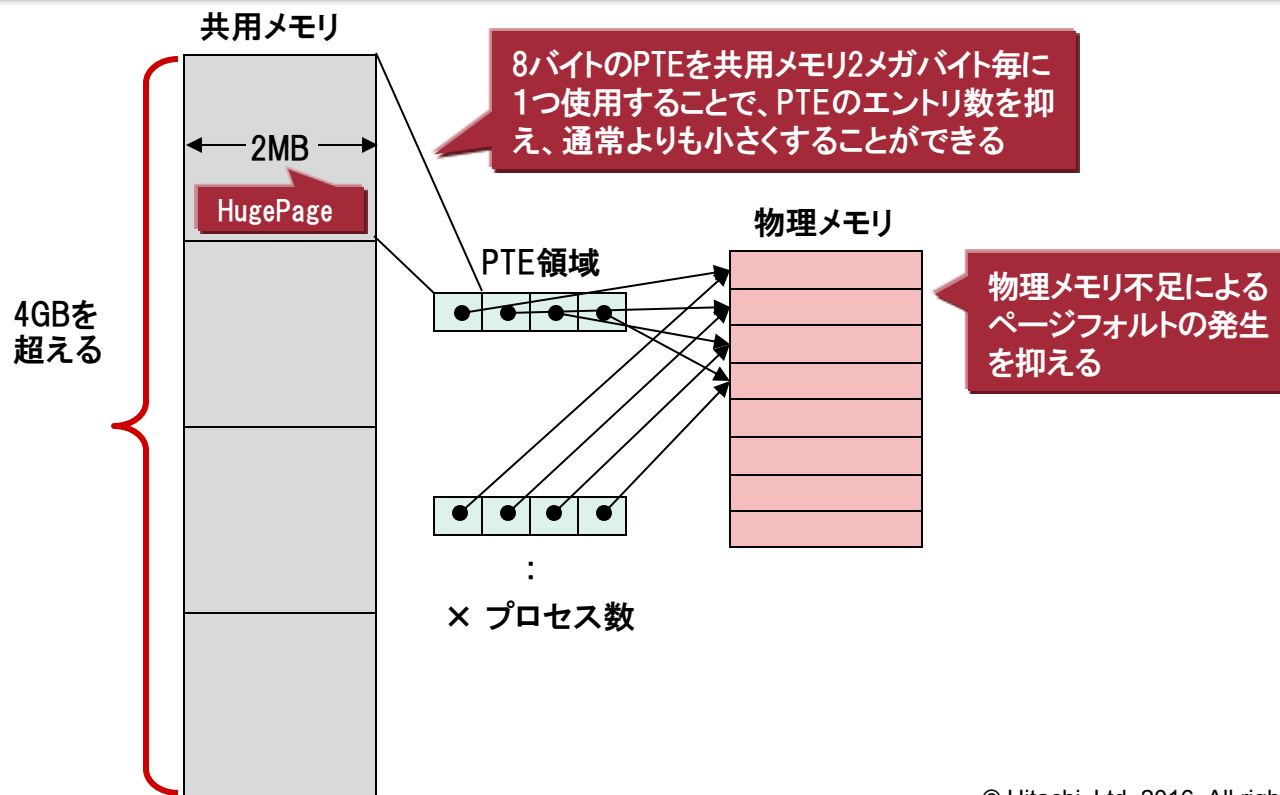


2-8-3 共用メモリのページサイズ(Linux)(2)

コツ

💡 共用メモリの所要量が4ギガバイトを超過する場合は、**共用メモリのページサイズ拡大機能** (HiRDBが使用する共用メモリのページサイズを、Linux OSの通常のページサイズ(4キロバイト)から、HugePageのページサイズ(2メガバイト)に変更する機能:バージョン09-50サポート) **を適用**してください。

これにより、8バイトのPTEを共用メモリ2メガバイト(HugePage1ページ)毎に1つ使用するため、PTE領域を通常よりも小さくすることができ、物理メモリ不足によるページフォルトの発生を抑えることができます。



2-8-4 共用メモリ使用数(Windows)(1)

現象 Windows版HiRDBでHiRDBの起動に失敗した。

原因 システム環境変数PDUXPLSHMMAXに必要なリソース数(共用メモリ使用数)が正しく設定されていない。過大な値だとメモリ不足になり、過小だと管理エントリが不足して共用メモリが確保できない。

 以下の場合、PDUXPLSHMMAXに適正な値が設定されません。
確認して、PDUXPLSHMMAXに適性な値を設定してください。

コツ

1. PDUXPLSHMMAXを設定していない。
2. 古いバージョンのマニュアルの見積り式を参照している。
見積もり式は、バージョンごとに変わることがあります。使用しているバージョンのマニュアルを参照してください。見積もり式は、マニュアル「システム導入・設計ガイド」の「リソース数に関連する環境変数の見積もり」を参照してください。
3. PDUXPLSHMMAXをシステム環境変数以外のHiRDBシステム定義などで設定している。
4. マシンにユニットを追加する際に、追加するユニットの情報でシステム環境変数に指定した値を見直ししなかった。

 バージョン09-50で、省略値ではHiRDBが自動計算するようになりました。このため、通常は指定する必要はありませんが、自動計算する際に想定している構成よりもリソース数を多く必要とする構成の場合は指定が必要になります。設定方法の選択基準については、マニュアル「HiRDB Version 9 システム導入・設計ガイド」の「リソース数に関連する環境変数の見積もり」を参照してください。

対策

複数ユニットがマシンにある場合には、各ユニットに適した値を設定できるので、pdntenvの-srオプションでユニットごとにリソース数およびリソース容量を設定してください。システム環境変数の設定があっても、ユニット単位の設定値が優先されます。

その他のシステム環境変数

以下のシステム環境変数も同様に、最新のマニュアルで見積って指定してください。

- ・PDUXPLMSGMNI: メッセージキュー識別子数
- ・PDUXPLMSGTQL: メッセージキューテーブル数
- ・PDUXPLSEMMAX: セマフォ識別子数

pdntenvの-srオプションで設定可能です。

 PDUXPLSHMMAXと同様に、上記のシステム環境変数についても、バージョン09-50では省略するとHiRDBが自動計算するようになりました。このため、通常は指定する必要はありませんが、自動計算する際に想定している構成よりもリソース数を多く必要とする構成の場合は指定が必要になります。設定方法の選択基準については、マニュアル「HiRDB Version 9 システム導入・設計ガイド」の「リソース数に関連する環境変数の見積もり」を参照してください。

PDUXPL系パラメタを変更した場合の注意事項

PDUXPL系パラメタの設定値を有効にするには、OS再起動(サービス再起動)が必要です。変更したら、pdntenvコマンドで設定値を確認してください。

3. システム構築時のコツ

- コツ1 : HiRDB起動前にはpdconfchkでパラメタのチェックをしよう！



3-3 ~ 3-4

- コツ2 : 系切り替え構成では、指定するホスト名に注意しよう！ **New**



3-5 ~ 3-6

3-1 運用ディレクトリのセットアップ(Linux)

New

現象

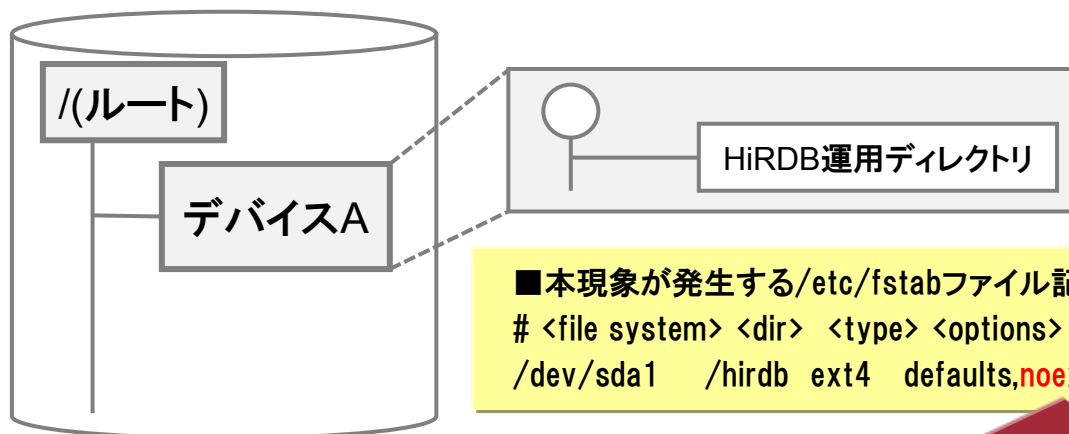
LinuxでHiRDBセットアップ時にpdsetupコマンドをrootユーザで実行したが、権限がない旨のエラーが出力されコマンドの実行に失敗した。

原因

HiRDBの運用ディレクトリを配置するディレクトリを/etc/fstabファイルに記載しマウントしていたが、オプションに“**noexec**”を指定していた。
noexecオプションを指定したマウントポイント上では、rootユーザであってもファイルの実行が出来ない。

コツ

💡/etc/fstab内の“HiRDBの運用ディレクトリを配置するマウントディレクトリ”のオプションから“**noexec**”を削除してください。



HiRDB運用ディレクトリのあるデバイスをmountする場合はnoexec指定をしない

現象 新しいマシンにHiRDBをセットアップしたが、HiRDBが起動できなかった。

原因 リモートシェルの設定をしていなかった。

コツ  HiRDB/パラレルサーバや高速系切り替え機能を使用する場合は、各サーバマシン間および現用系と予備系マシン間でリモートシェルを実行します。したがって、これらのマシン間で相互にログインできるよう、HiRDB管理者の\$HOME/.rhostsに設定してください。
また、HiRDBではリモートシェルをrootユーザで実行するものがあるため、rootユーザのリモートシェルの設定もしてください。

現象 HiRDBシステム定義不正で、HiRDBが起動できなかった。

原因 HiRDBシステム定義の指定方法に誤りがあった。

コツ  HiRDBを起動する前に、HiRDBシステム定義の内容を確認するコマンド pdconfchkを使って、内容に誤りがないことを確認してください。

 バージョン09-50では、運用を継続していくと、将来的にユニットダウン・DB破壊等が発生する可能性がある、危険度の高いHiRDBシステム定義について、pdconfchkコマンドでチェック(警告メッセージを出力し、リターンコード4で終了)できるようになりました。

 HiRDB Configuration Assistant 01-03では、稼働中のHiRDBをバージョン09-50にバージョンアップするとき、現在指定している各種パラメタの設定値が推奨値になっているのか容易にチェックすることができるようになりました。

確認方法

内容確認は、2回に分けて実行します。

(1)システムファイル^(*)作成前

pdconfchkコマンドに-nオプションを付けて実行し、HiRDBシステム定義の内容のみチェックします。
HiRDBシステム定義の内容に誤りがあると、システムファイルの作成でエラーになる可能性がありますので、必ず確認してください。

【コマンド実行例】 pdconfchk -n

(2)システムファイル^(*)作成後

pdconfchkコマンド(オプションなし)を実行し、HiRDBシステム定義の内容と、システムファイルの存在およびアクセス権限についてチェックします。

【コマンド実行例】 pdconfchk

注意事項

HiRDBシステム定義のシステム共通定義(pdsys)のpdunit文で指定するホスト名および運用ディレクトリは、間違わないようにしてください。指定されたホスト名および運用ディレクトリの環境をチェックしますので、ここに誤りがあると、正しいチェックができません。ホスト名および運用ディレクトリは、コマンド実行前に確認してください。

(*) システムログファイル、シンクポイントダンプファイル、ステータスファイル

3-5 定義ミスによるHiRDB起動失敗(1)

New

現象

スタンバイ型系切り替え構成を使用するHiRDBについて、ネットワークに関する以下のエラー事象が発生する。

事例1:HiRDBを起動したところ、以下のメッセージを出力し起動に失敗する

KFPS00349-E UNT1 Insufficient network port

KFPS00352-I UNT1 Port number maintenance information

KFPS01861-E UNT1 Error occurred during command execution. reason code=COMMUNICATION

事例2:HiRDBを起動したところ、以下のメッセージを出力し、起動が完了しない。

KFPS05078-I Unable to recognize HiRDB system initialize completion

事例3:HiRDBは起動しているが、HiRDBクライアントが以下のメッセージを出力し、HiRDBサーバに接続できない

KFPA11723-E Communication error occurred, reason=HiRDB NOT UP

原因

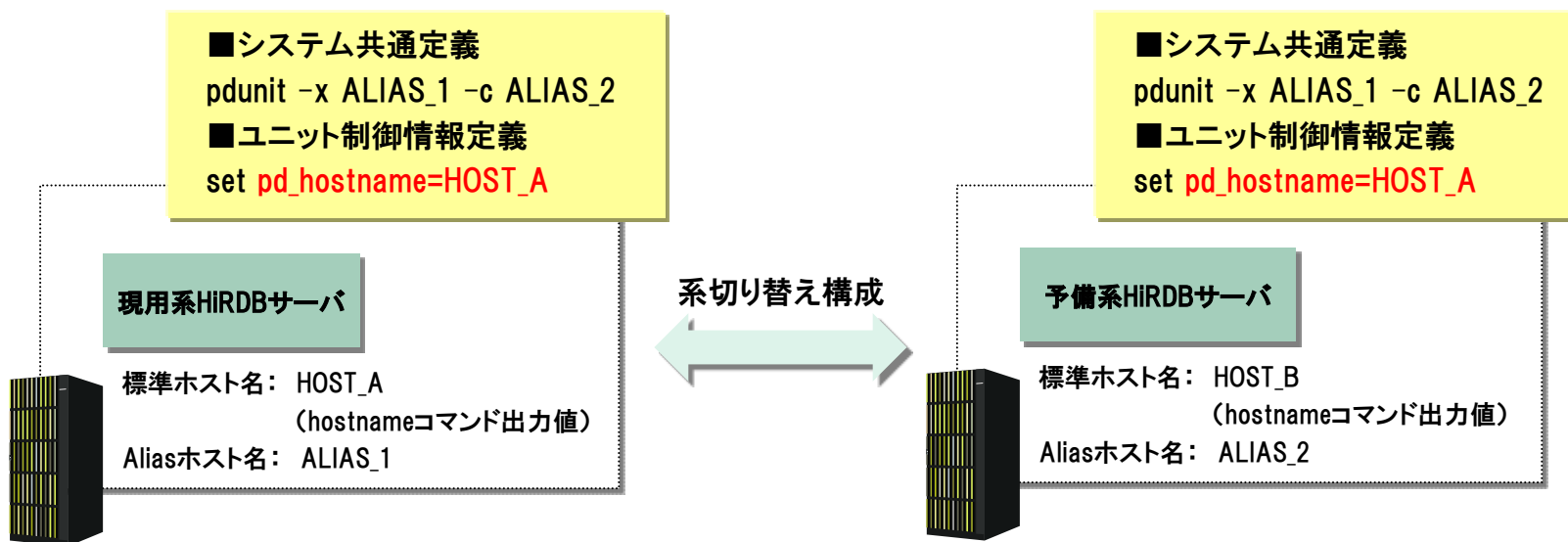
ユニット制御情報定義pdutsysのpd_hostnameオペランドに、標準ホスト名(OSのhostnameコマンドの出力値)ではなく、別名(エイリアスIPアドレス用のホスト名等)を指定していた。

3-6 定義ミスによるHiRDB起動失敗(2)

New

コツ

💡 pd hostnameオペランドには、OSのhostnameコマンドの出力値を記載してください。
HiRDBに標準ホスト(OSのhostnameコマンドの出力値)を使用せず別ホスト名(エイリアスIPアドレス用)のみを使用している場合でも、系切り替え構成ではHiRDBが実行系・待機系かを判断するためにpd_hostnameオペランド記載の標準ホスト名を使用します。
また、pd_hostnameオペランドには予備系であっても現用系の標準ホスト名を指定する必要があります。



現象 ファイルアクセスエラーが発生し、HiRDBが起動できなかった。

原因 HiRDBファイルシステム領域を配置するボリュームにアクセス権限がなかった。

コツ  ボリュームにHiRDB管理者へのアクセス権限を設定してください。
アクセス権限だけでなく、ディスクの活性化(例えば、AIXの場合はvaryonvgコマンド)も実行してください。特に、系切り替え構成においては、IPアドレスのアップおよびディスクの活性化がなされておらず、HiRDB起動不可になった事例もあります。

コツ

 Windows版HiRDBでデータベース初期設定ユーティリティ(pdinit)を実行する場合は、-uオプションで認可識別子を、-pオプションでパスワードを個別に指定して実行してください。

解説

pdinit実行時にDBA権限が決まります。-uおよび-pオプションの指定がない場合は、HiRDBが固定名称でDBA権限を作成するため、認可識別子、パスワードが推測可能となり、マシン環境によっては、DBA権限が他者に使用される恐れがあります。

コツ

🔔 簡易セットアップツールの標準セットアップやカスタムセットアップで作成した環境を本番環境として使用する場合、システムログの設定は必ず見直しをしてください。

解説

簡易セットアップツールでは、開発環境を想定しているため、システムログのアンロードを行わない設定になっています。したがって、そのまま本番環境に適応すると、ディスク障害時に、DBを最新状態に回復できない状況に陥るので、注意してください。

システムログのアンロードを行わない設定

HiRDBシステム定義 `pd_log_unload_check = N` を指定すると、システムログファイルのアンロード状態を、HiRDBがチェックしません。よって、システムログのアンロード処理が不要となります。ただし、この設定の場合、バックアップ取得時点にしかDBを回復できませんので、注意してください。

その他見直し項目

各種キャッシュサイズ、常駐プロセスの設定も見直してください。

簡易セットアップツールのウィザードセットアップ

HiRDB Version 8 08-04以降では、簡易セットアップツールにおいて、ウィザード形式で環境作成ができる「ウィザードセットアップ」をサポートしました。

ウィザードセットアップでは、任意のDBサイズで環境作成することができます。また、システムログのアンロード運用についても、ウィザード画面で選択することができます。

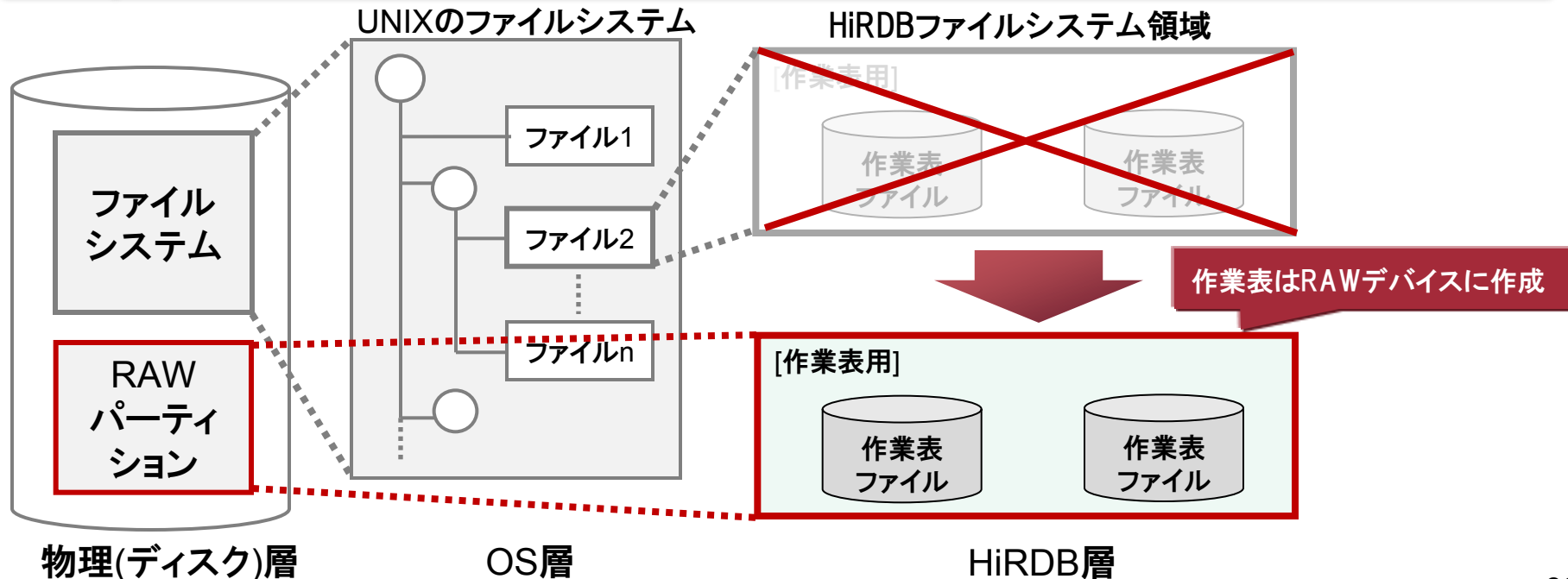
3-10 作業表の作成場所(UNIX)

現象 UAP実行時に、PDCWAITTIMEオーバによるタイムアウトが多発する。

原因 カーソルクローズ時、作業表のfsync^(*)を発行するが、同時に大量の作業表を扱うSQLが発行されたことにより、fsyncに時間が掛かり、PDCWAITTIMEオーバでタイムアウトとなった。

(*)：メモリ上にあるファイルの内容をストレージデバイス上のものと同期させる。

コツ 💡 OS制御(ファイルキャッシュ)の動作に影響を受けない(fsyncを発行しない)ように、作業表へのアクセスが多いシステムでは、作業表をRAWデバイスに作成することを推奨します。

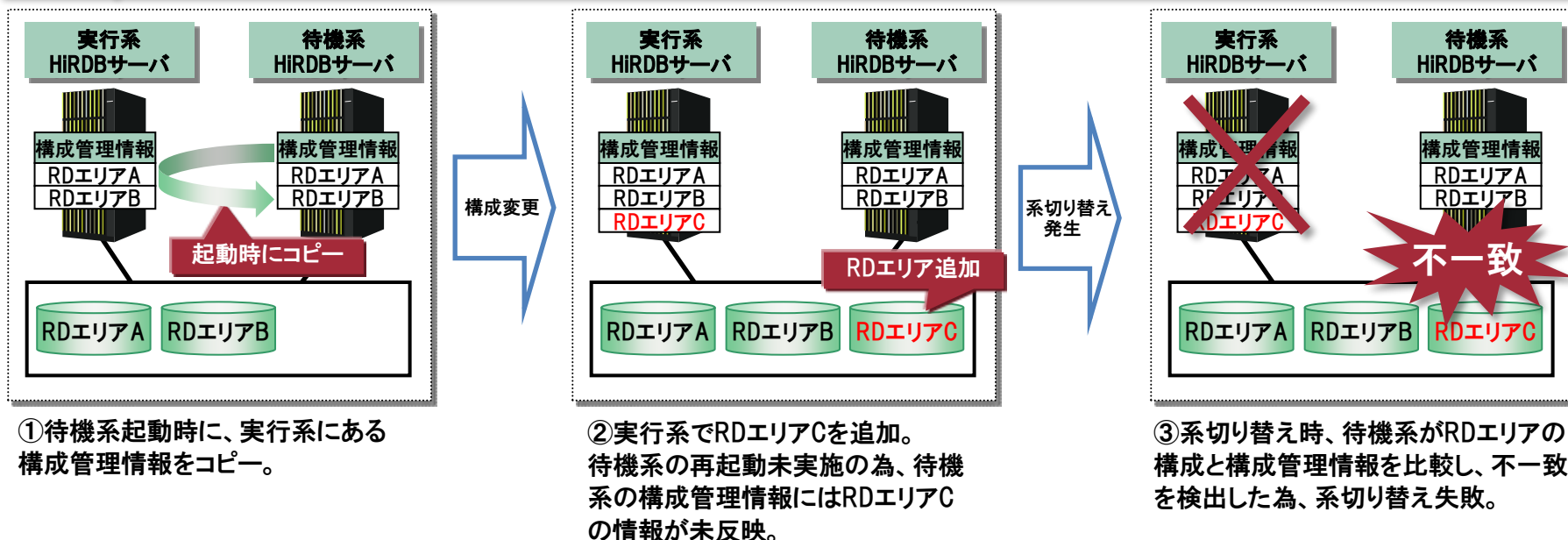


3-11 構成変更後に系切り替え失敗(1)

New

現象 高速系切り替え、又は1:1スタンバイレス型系切り替え構成を使用するHiRDBについて、RDエリアを追加した後に、系切り替えすると、系切り替えに失敗した。

原因 RDエリア追加後に、待機系HiRDBサーバを再起動していなかったためRDエリアの構成とHiRDBがメモリ中に保持する構成管理情報が不一致となり、系切り替えに失敗した。
高速系切り替え、又は1:1スタンバイレス型系切り替え構成では、HiRDBのシステムサーバを起動したまま待機するため、構成変更を認識するには再起動が必要となる。



コツ 構成変更を実施した場合は、待機系のHiRDBを再起動してください。

■主な構成変更の例は以下の3つです。
(1)RDエリアの追加 (2)RDエリアの削除 (3)RDエリアの拡張

待機系HiRDBの再起動を必要とする構成変更内容

New

待機系HiRDBの再起動を必要とする構成変更内容の詳細は、適用している系切り替え構成に応じて以下のマニュアルを参照してください。

●高速系切り替え構成

- [システム運用ガイド]－[系切り替え機能の運用]－[スタンバイ型系切り替え(サーバモード)の運用]
 - －[系の切り替え時間の短縮(ユーザサーバホットスタンバイ, 高速系切り替え機能)]

●1:1スタンバイレス型系切り替え構成

- [システム運用ガイド]－[系切り替え機能の運用]－[1:1スタンバイレス型系切り替えの運用]
 - －[1:1スタンバイレス型系切り替え機能使用時の注意事項]

4. アプリケーション開発時のコツ

- コツ1：データの大量更新の前にはLOCK TABLE文を実行しよう！



4-1

- コツ2：クライアント環境変数の優先順位に注意！



4-2

- コツ3：コネクションプールを使ったシステムでは、このクライアント環境変数を設定しておこう！



4-4

- コツ4：ブロック転送機能は、通信バッファサイズ指定での使用がお勧め！



4-5 ~ 4-6

- コツ5：アクセスパスは必ず確認しよう！



4-9 ~ 4-10

現象

開発環境で、データの大量削除を実行したら、KFPA11912-E「排他制御のための作業領域不足」でエラーとなった。

原因

開発環境では、本番環境では想定していない操作を行うことがあります。この場合は、本番環境では想定していないデータの大量削除を行ったことにより、本番環境に基づいて見積もりの排他制御のための作業領域が不足した。

コツ

💡 大量更新する場合は、表の操作が単独の実行であっても、LOCK文の指定を検討してください。ただし、表中の全行を削除する場合は、PURGE TABLE文を使用してください。

LOCK文、PURGE TABLE文

LOCK文は行単位の排他より、排他オーバーヘッドや排他資源の節約ができますので、資源の消費を抑えるために、LOCK文を使って、排他制御用プールサイズの不足を回避できます。

PURGE TABLE文は、表データの管理情報をクリアします。このため、DELETE文を実行して一つ一つの行を削除するより、高速に表を空にします。また、管理情報をクリアするため、削除した領域は再利用可能となり、使用中空きページが発生しません。

対策

データ削除実行前に、削除対象の表に対して、排他モードのLOCK文(LOCK TABLE ~ IN EXCLUSIVE MODE)を実行して、表単位の排他を確保するようにできないか、検討してください。また、KFPA11912-EでSQLがエラーになった場合、syslog(Windowsの場合はイベントログ)にKFPS00443-I(排他制御用プール不足)やKFPS00444-I(排他資源要求がPDLOCKLIMITの上限値に達した)が出力され、全体的に不足している場合は、排他制御用プールサイズや排他資源要求数を大きくしてください。

現象

Windowsで、テスト環境のHiRDBから本番環境のHiRDBへ接続先を変更するため、Windowsディレクトリ下のHiRDB.iniファイルのPDHOSTのホスト名を変更したが、接続先が変わらない。

原因

OSの環境変数に「PDHOST＝テスト環境のホスト名」が設定されていたため、変更が有効になっていなかった。

クライアント環境変数の検索順序

①環境変数グループ^(*)

②UAP内部で設定した環境変数

③OSに設定した環境変数

④Windowsディレクトリ下のHiRDB.ini(Windowsのみ)

上記の検索順を考慮して、何の設定が有効になっているかを確認してください。現在有効になっているクライアント環境変数は、SQLトレースでも確認できます。

また、クライアント環境変数は、OpenTP1のユーザサービス定義やユーザサービスデフォルト定義など、上位製品の環境変数として指定することもありますので、注意してください。

(*)：複数接続機能およびOLTP下のUAPの場合などに使用します。

コツ



OSの環境変数にクライアント環境変数を設定すると、複数のシステムで有効になり、トラブルの要因になることが多いため、できるだけ**システム個別に設定**するようにしてください。



TYPE4 JDBCの注意事項

Type4 JDBCドライバの場合は、接続情報に指定された内容に従い、クライアント環境変数を参照します。

OSの環境変数や、Windowsディレクトリ下のHiRDB.iniは参照しませんので、注意してください。

Type4 JDBCドライバにおけるクライアント環境変数の優先順位については、マニュアル「UAP開発ガイド」の「Type4 JDBCドライバ」の「接続情報の優先順位」を参照してください。

4-3 クライアント環境変数グループの登録先(Windows)

現象

WindowsのInternet Information Services (IIS)からHiRDBに接続しようとした所、KFPA11724-E (reason=NOT ENVIRONMENT GROUP:環境変数グループが見つからない)でエラーとなった。

原因

環境変数グループの登録先をユーザグループとしていた。WindowsのサービスからHiRDBに接続する場合、環境変数グループの登録先は、システムグループとする必要があります。

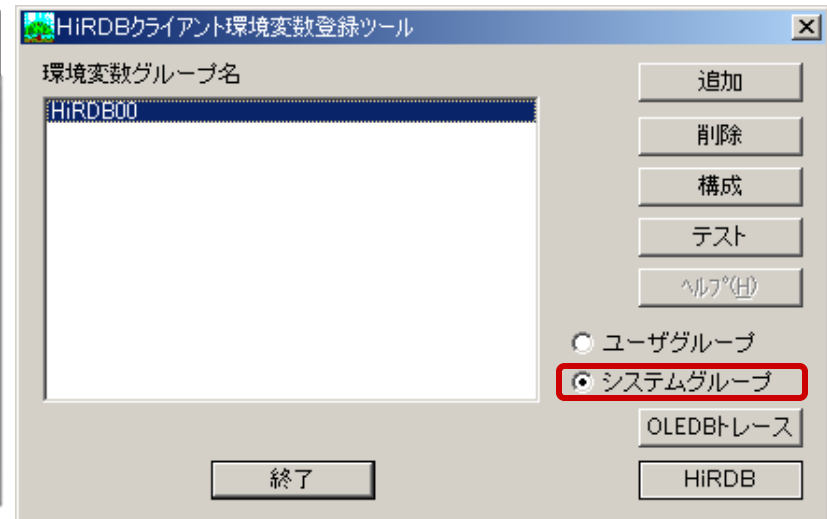
コツ

🕒 IISのようにWindowsのサービスからHiRDBに接続する場合、**環境変数グループの登録先は、システムグループを指定**してください。OLTP下のX/Openに従ったAPIを使用したUAPから接続する場合も同様です。

環境変数グループの登録

Windowsでクライアント環境変数グループをレジストリに登録する場合は、HiRDBクライアント環境変数登録ツールを使用します。
登録先として、「ユーザグループ」と「システムグループ」が選択できます。

WindowsのサービスからHiRDBに接続する場合は、「システムグループ」の環境変数グループしか参照できません。UAPの構成に応じて、登録先を検討してください。



現象

ADO.NET対応アプリケーションからHiRDBにアクセスする環境で、HiRDBの系切り替えが発生した場合、系切り替えが完了したにもかかわらずHiRDBへの接続エラーが発生した。

原因

ADO.NETのプーリング機能を使用すると、接続を再利用しますが、HiRDBが系切り替えしたにもかかわらず、接続を再利用してしまうため、この現象となります。

コツ

 ADO.NETのプーリング機能を使用する場合は、**自動再接続機能を使用**して、HiRDBクライアント側で自動的に再接続するようにください。

自動再接続機能とは

サーバプロセスダウン、系切り替え、ネットワーク障害などの要因で、HiRDBサーバとの接続が切断された場合に、HiRDBクライアント側で、自動的に再接続します。

本機能を使用する場合は、クライアント環境変数PDAUTORECONNECT=YESを指定します。

ADO.NETのプーリング機能

ADO.NETのプーリング機能を使用する場合は、コンストラクタHiRDBConnectionのプロパティPoolingでtrueを指定します。

 なお、規定値はtrueですので、明示的に指定していない場合も、上記現象に該当しますので、注意してください。

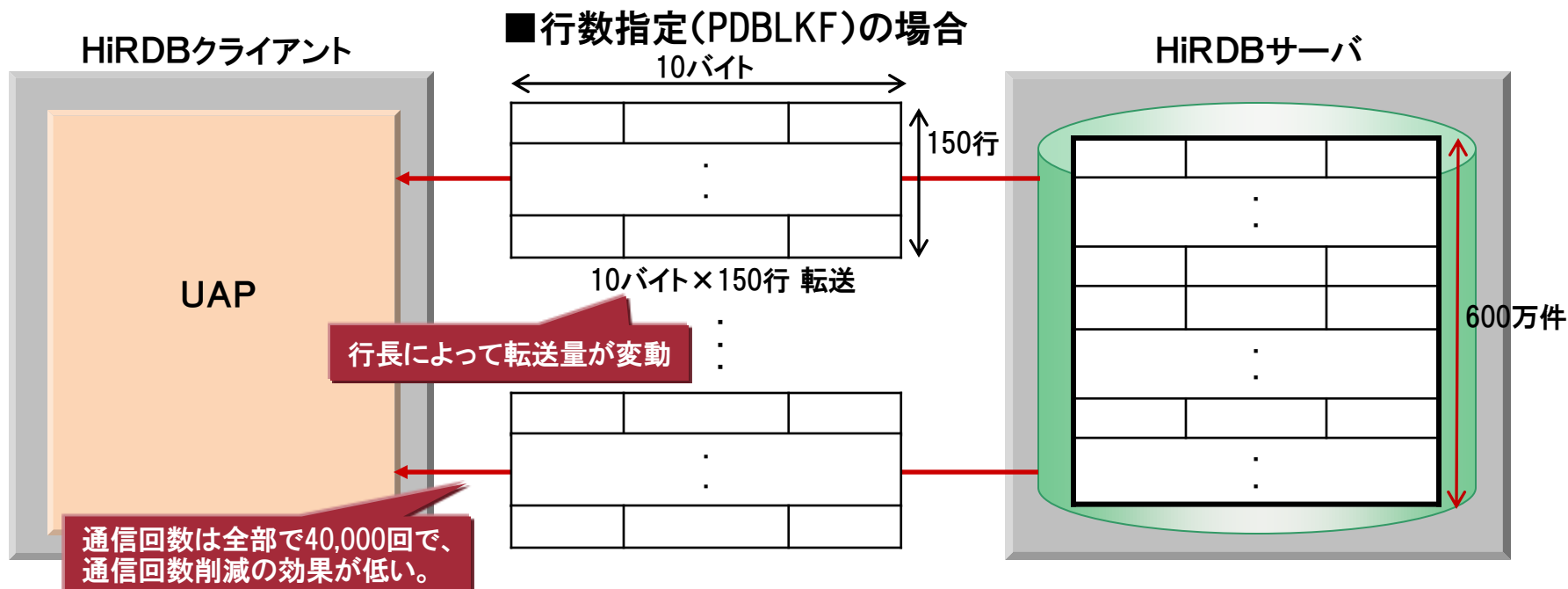
4-5 ブロック転送機能(1)

現象 データの大量検索(600万件)に、数時間がかかる。

原因 行数指定のブロック転送数機能(PDBLKF=150)を使用した場合、行長が短い(10バイト)と大量検索では通信回数削減の効果が低いため、数時間掛かった。

ブロック転送機能とは

ブロック転送機能とは、検索結果をメモリ上に構築し、複数行をまとめてクライアント側へ転送する機能のことです。配列を使用できないプログラミング方式の場合、ブロック転送機能を使用することで、クライアントとサーバ間の通信回数を削減することができます。



4-6 ブロック転送機能(2)

コツ

💡 ブロック転送数機能は、通信バッファサイズ指定(PDBLKBUFSIZE)での使用がお勧めです。
バッファサイズは、OSの通信バッファサイズを目安に指定してください。
これにより、指定した通信バッファサイズのメモリの範囲内で、最適な行数のブロック転送を行うことができます。

💡 バージョン09-50で、ブロック転送機能はデフォルトで有効となります。ユーザがブロック転送機能の指定を何もしない場合、PDBLKBUFSIZEの省略値が10キロバイトとなります。PDBLKの省略値は1行で従来から変更ありません。

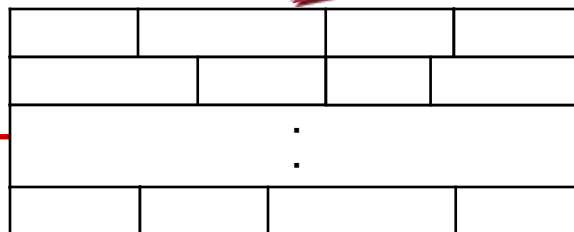
■通信バッファサイズ指定(PDBLKBUFSIZE)の場合

大きさは、OSの通信バッファサイズ
を目安に指定する。

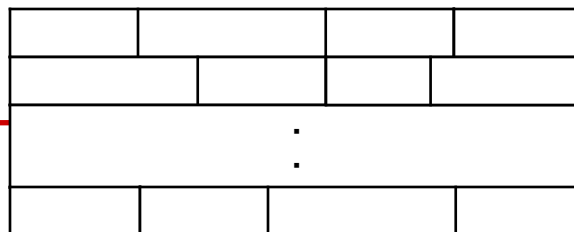
HiRDBクライアント

UAP

このケースではPDBLKBUFSIZEに
50(キロバイト)を指定し、通信回数を
1,172回まで削減(97%減)した。



バッファサイズに収まる行数分転送



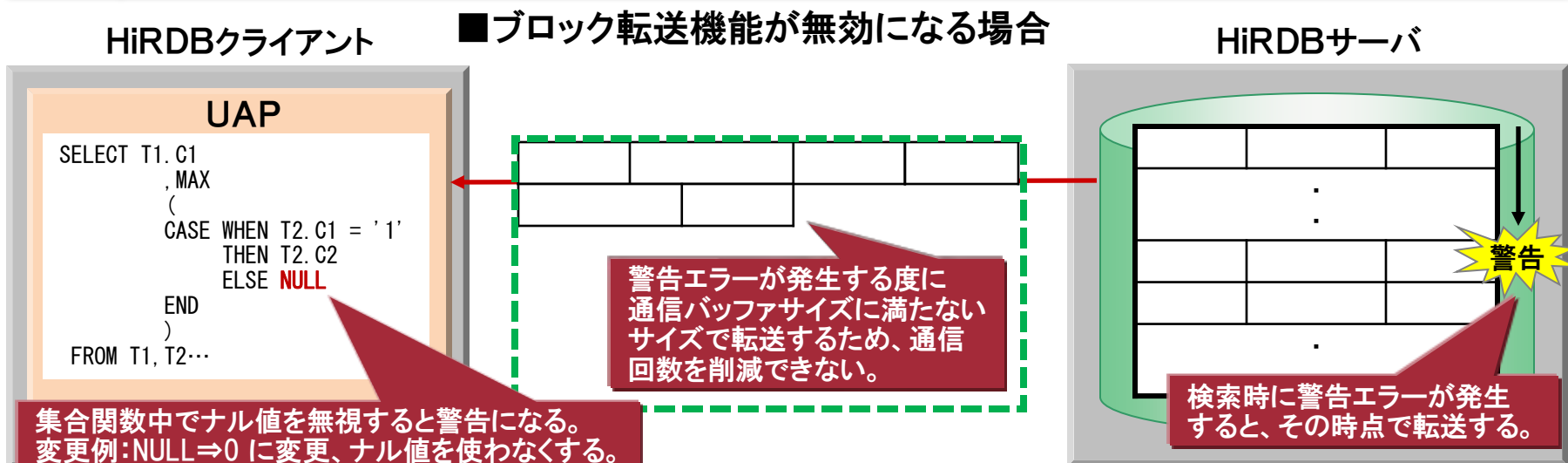
HiRDBサーバ

600万件

4-7 ブロック転送機能が無効になるケース

現象 大量検索(600万件)を行う類似SQL中1つだけが、他の3倍の時間が掛かる。

原因 検索時に警告エラーになると、警告エラー発生時点で転送処理を行います。通信バッファサイズ(PDBLKBUFSIZE)に満たないサイズで転送するため、ブロック転送機能の効果が得られなかった。



コツ ブロック転送機能が無効になっていないか、テスト時に警告の有無を確認してください。

警告エラーの場合は、SQLトレースの「SQLWARN」の先頭が「W」またはSQL連絡領域(SQLCA)の「SQLWARN0」の先頭が「W」となります。

SQLトレースの詳細は、マニュアル「UAP開発ガイド」の「UAPの障害対策」の「SQLトレース機能」を
SQL連絡領域(SQLCA)の詳細は、マニュアル「UAP開発ガイド」の「付録A.1 SQL連絡領域の構成と内容」参照してください。

対策 警告エラーにならないように、SQLを修正してください。

現象 データの大量検索(600万件)の性能評価のためにSQLトレースを取得したが、SQLトレースの出力に時間が掛かった。

原因 SQLトレース取得では、実行するオペレーション毎に情報を出力するためのI/Oが発生する。大量検索の場合は、I/Oが多くなり時間が掛かる。

 バージョン09-50で、SQLトレースに出力する情報を拡充し、より詳細な情報を出力するようにしました。SQLトレースの出力処理に掛かった時間もわかるので、SQLトレース出力時にSQL実行時間とSQLトレース出力時間のどちらで時間が掛かっているか切り分けができるようになりました。

コツ  SQLトレース取得にはオーバーヘッドが掛かることを考慮して、以下を検討してください。

- RAMディスク等の高速にアクセスできる場所に出力する。
- クライアント環境変数PDSQLTRCOPENMODEにSQLトレースの出力時間を短縮できる指定値CNCTを指定する。

現象 本番環境で、UAPの実行に時間を要した。

原因 テスト時に、アクセスパスの確認を行っていなかったため、効率の悪いアクセスパスのSQLがあり、UAPの実行に時間を要した。

コツ  効率の良いアクセスパスを意識したSQL設計(検索方法、結合方式、排他オプションなど)を行ってください。
APのテスト時には必ず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てください。なっていない場合は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てください。
アクセスパス情報の取得については、HiRDB技術資料「HiRDBチューニング解説」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

4-10 アクセスパスの確認:その2

現象

pdsq|コマンドからの検索では1秒で返るが、同じ検索条件のSQLをUAPから実行すると13秒かかる。

原因

pdsq|コマンドでは、「LIKE 'ABC%」(前方一致)を指定してアクセスパスを確認したが、UAPでは、「ABC%」部分を?パラメタに変更(?パラメタは'ABC%」)した。「LIKE 'ABC%」と「LIKE ?」では、インデクスの優先順位が異なるため、優先して使用されるインデクスが変わり、検索時間が遅くなった。

インデクスの優先順位については、HiRDB技術資料「アプリケーション開発ガイド」を参照してください。

インデクス:(C1,C2)

WHERE C1 LIKE 'ABC% AND C2>100

C1 LIKE 'ABC%'の方がC2>100よりも優先順位が高いため、C1列上のインデクスが優先して使用される。

WHERE C1 LIKE ? AND C2>100

C2>100の方がC1 LIKE ?よりも優先順位が高いため、C2列上のインデクスが優先して使用される。

コツ

💡 LIKE述語の前方一致比較を定数で行うのと?パラメタで行うのでは、アクセスパスが異なるので、アクセスパスの確認は、実際に使用するSQLに合わせて行うようにしてください。

HiRDB技術資料のURL

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

New

現象

不当にデータ件数が少ない状態で、最適化情報収集ユーティリティ(pdgetcst)を用いて最適化情報を収集した後に、最適化情報を再収集せずに運用を継続したところ、デッドロックが多発した。

原因

データの分布情報が変わる更新を行ったにも関わらず、最適化情報を収集しなかった。このため、古い最適化情報を基にHiRDBがTABLE SCANのアクセスパスを選択した。この結果、全データに対して一時的にロックを取得して検索し、別の更新処理との間でデッドロックが発生した。

4-12 最適化情報収集の影響でデッドロック増加(2)

New

最適化情報を収集するタイミングによって、以下の通りアクセスパスが変わります。

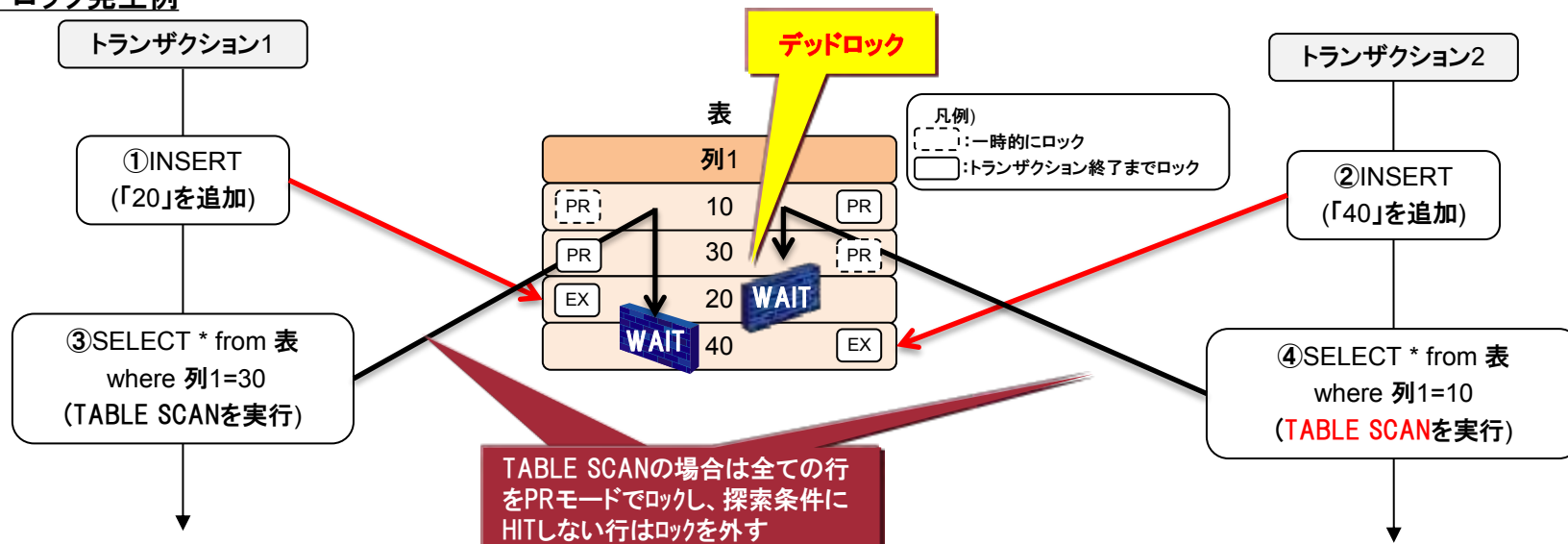
①データ件数が少ない時に収集した場合

検索対象外の行にも一時的にロックを取得して全データを検索するTABLE SCANのアクセスパスを選択します。

②データ件数が多い時に収集した場合

検索対象外の行にはロックを取得しない、インデクスを使用するINDEX SCANのアクセスパスを選択します。

デッドロック発生例



コツ

💡 HiRDBでは、最適化情報の収集登録を実施しなくても効率の良いアクセスパスを選択できます。データの分布情報が変わる更新を行う業務の場合は、最適化情報を収集しないでください。マニュアル「HiRDBパフォーマンスガイド」を基に、効率の良いアクセスパスになるようにSQL単位でチューニングを実施してください。

4-13 ステートメントプーリング使用時の定義系SQLの実行 (1) HITACHI Inspire the Next

Update

現象 運用中に表名変更をしようとしたところ、SQLがタイムアウトでエラーとなった。

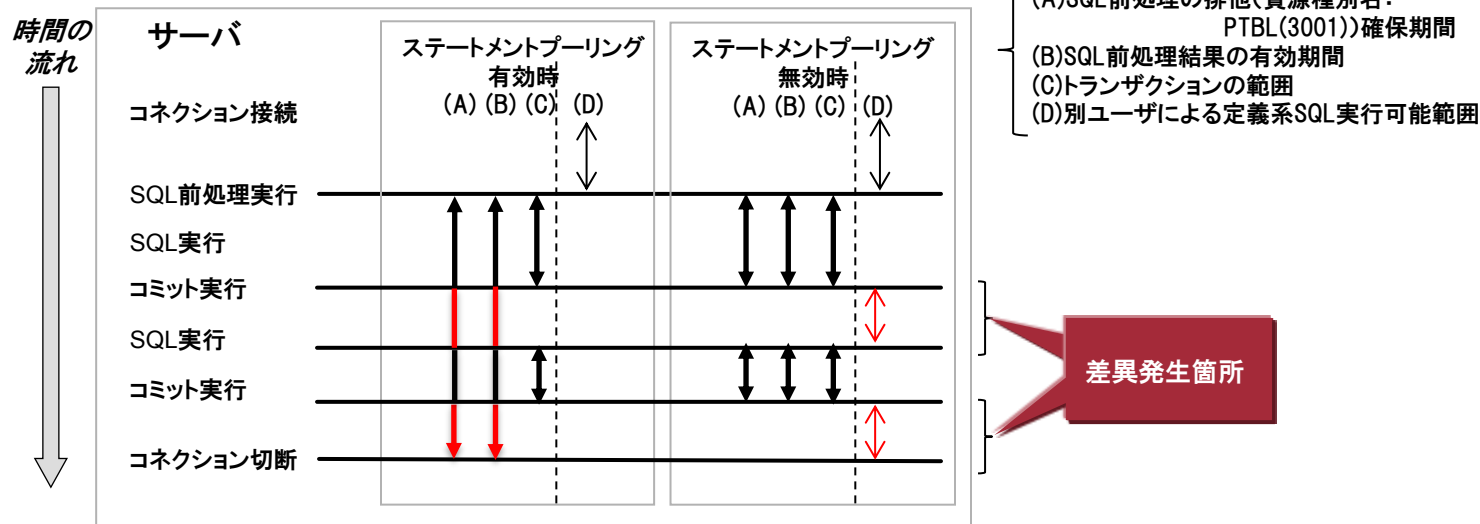
原因 Cosminexus側でステートメントプーリングが有効になっていたため(*1)、Type4 JDBCドライバを使った操作系SQLで誰かがスキーマ資源にアクセスしている時に、定義系SQLを実行すると、下記の場合は定義系SQLが排他待ちの状態になるため、タイムアウトした。

- ・スキーマ資源にアクセスしているコネクションを切断するまでの間
- ・STATEMENT_CLOSE_BEHAVIORにTRUEを設定している場合は、スキーマ資源にアクセスしているステートメントオブジェクトをクローズしコミットするまでの間

ホールダブルカーソルを使用して誰かがスキーマ資源にアクセスしている時に、定義系SQLを実行した場合も、スキーマ資源にアクセスしているコネクションを切断するまでの間、定義系SQLは排他待ちの状態になります。

(*1): HiRDBではCosminexusに合わせてステートメントプーリングはデフォルトで有効となっています。

SQL前処理排他に対する定義系SQL実行可能範囲



New

コツ



ステートメントプーリングが有効な状態、またはホールダブルカーソルを使用している場合は、定義系SQLは実行できません。本番運用中に定義系SQLを実行する運用を検討している場合は、以下の何れかの対処を検討してください。

- ・業務を一旦止めてDISCONNECTした後で定義系SQLを実行する。
- ・APサーバでSTATEMENT_COMMIT_BEHAVIORにFALSEを指定してステートメントプーリングを無効にする、かつ、ホールダブルカーソルを使用しない。

確認方法

定義SQL実行時に排他待ちとなるかを確認する場合は、「pdls -d lck -a」を実行してください。実行結果から、アクセス対象の表に対する排他情報中に、資源種別「3001」の情報がなければ確認してください。該当する情報がある場合は、定義系SQLを実行すると排他待ちになります。

4-15 暗黙的ロールバックの無視

現象 JDBCドライバからのINSERTバッチで、ユニークエラー後、KFPA11901-E(カーソルオープン前にFETCH文を実行)でエラーになった。

原因 INSERTバッチでユニークエラーが発生すると、暗黙的ロールバックを行う。UAPの処理として、例外(ユニークエラー)を意図的に無視するロジックだったが、暗黙的ロールバック後、暗黙的ロールバックによってクローズされたカーソルでFETCH文を実行し、KFPA11901-Eとなった。

```
prepareStatement pstmt1 (SELECT)
pstmt1.executeQuery (カーソルオープン)
pstmt1.rs.next(FETCH)
```

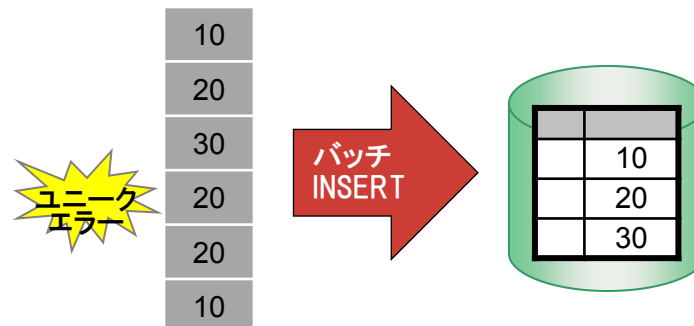
```
prepareStatement pstmt2(INSERT )
```

```
pstmt2.addBatch pstmt2 ×n
```

```
pstmt2. executeBatch → KFPA11803-E(ユニークエラー)暗黙的ロールバック
```

例外無視

```
pstmt1.rs.next(カーソル②でFETCH) → KFPA11901-E(OPEN前のFETCHエラー)
```



コツ

 更新SQLでは、暗黙的ロールバックの考慮を入れてください。または、例外発生時は無条件にROLLBACK文を発行するようにしてください。

コツ

 クライアント環境定義PDCLTAPNAMEに、アプリケーションの種別や実行者が識別できる名称を設定してください。

解説

何か問題が発生した際、どのAP(業務AP(オンライン、バッチ)、BIツール、pdsqなど)からのSQLか直ぐに特定することができます。

PDCLTAPNAMEの設定ルール

- 1.AP毎にアプリケーションの種別や実行者が識別できる名称を設定してください。
- 2.pdsq実行時は、使用者のIDを設定してください。
- 3.UAPの識別名称に英数字以外の文字を使用した場合は、pdcancelコマンドを実行できないことがありますので、英数字だけで構成される名称にすることをお勧めします。
- 4.次の文字列は、UAPの識別名称に使用しないでください。
 - ・「pd」で始まる文字列
 - ・「hds」で始まる文字列
 - ・「0」で始まる文字列

UAP名称が表示される情報

指定した名称は、次の各情報のUAP名称として表示されます。

- ・ pdsコマンドの表示結果
- ・ SQLトレースファイル
- ・ 接続ユーザ情報ファイル(%PDDIR%¥spool¥cncctusrinf)
- ・ UAPに関する統計情報

4-17 HiRDB SQL Executerでのデータ誤更新

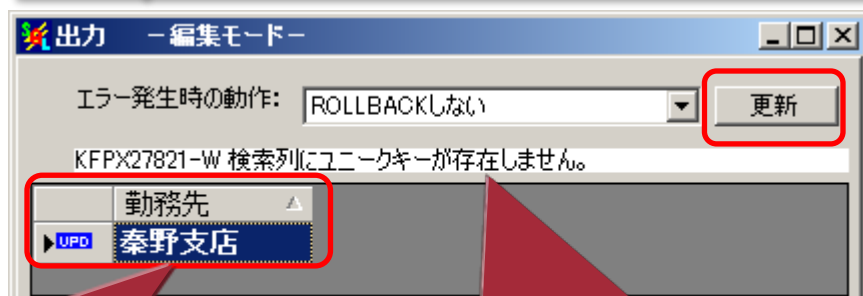
現象

SQL Executerは、ユーザが記述した検索SQLの検索結果データを表形式で表示し、表示したデータを表計算ソフトのように編集することで、DBを更新する機能(*1)を提供している。この機能を使用したデータ更新において、検索結果が1行であるにも関わらず複数行更新してしまった。

(*1):HiRDB Version 9でサポート。

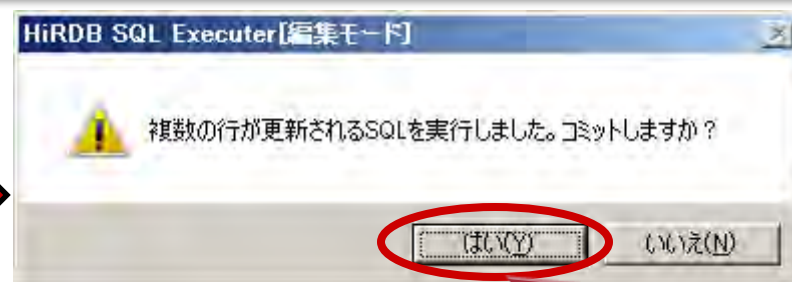
原因

この更新機能では、検索SQLの検索結果データに行を特定する為の一意(ユニーク)なキー情報が含まれていない場合に、更新対象以外のデータ(見えないデータ)も更新するため、更新が複数行となった。



勤務先データを変更し「更新」ボタンを押す。

検索結果にユニークなキー情報を含んでいないため、警告メッセージを出力。



「はい」を押すと意図しない更新対象以外のデータ(見えないデータ)も更新してしまう。

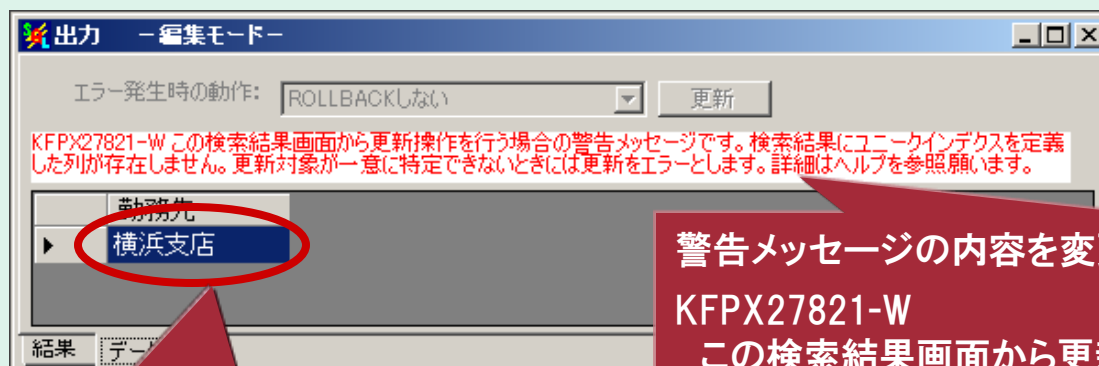
コツ

💡 更新時に更新行が複数であった場合、コミット前にダイアログにより続行是非の確認をしますが、「はい」を押すと意図しない更新対象以外のデータ(見えないデータ)も更新してしまうため、十分注意してください。

対策

見えないデータの更新をエラーとする対策版(09-00-06,09-01-05,09-02-02)を適用してください。対策の詳細については、**4-26** を参照してください。

4-18 HiRDB SQL Executerの改善内容

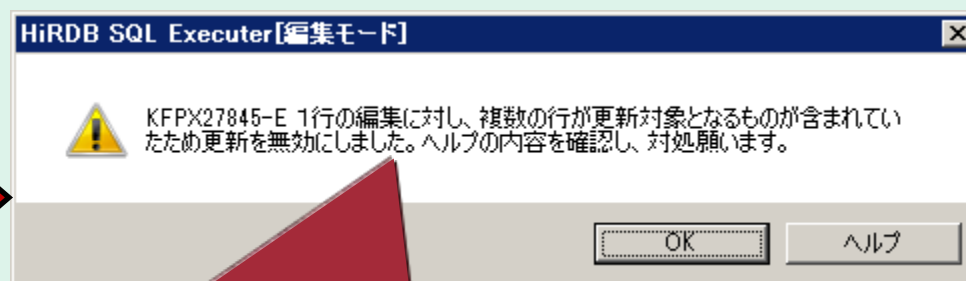


警告メッセージの内容を変更。

KFPX27821-W

この検索結果画面から更新操作を行う場合の警告メッセージです。検索結果にユニークインデックスを定義した列が存在しません。更新対象が一意に特定できないときには更新をエラーとします。詳細はヘルプを参照願います。

検索結果を編集し、更新ボタンを押す。



更新対象が一意に特定できないときには更新をエラーとし、見えない行を更新しようとした場合に処理を中止します。

KFPX27845-E

1行の編集に対し、複数の行が更新対象となるものが含まれていたため更新を無効にしました。ヘルプの内容を確認し、対処願います。

5. おわりに

HiRDB

これからもHiRDBに
ご期待ください！



付録A. その他事例

現象

HiRDB Text Search Plug-inのn-gramインデクスが格納されているRDエリアが壊れたため、インデクスの再作成(CREATE INDEX)を実施したところ、非常に多くの時間を要してしまい、その間、表にアクセスできず、結果的に業務停止時間を長くしてしまった。

原因

n-gramインデクスの場合は、インデクス作成方法(CREATE INDEXのPLUGINオプションCREATEMODEに指定)を省略すると即時登録となりますが、即時登録はCREATE INDEXの実行時間が長くなります。CREATE INDEX実行時は、表へ占有モードの排他がかかりますので、全文検索以外の列に対する操作も、その間は再開できません。

コツ

 **n-gramインデクスの場合は、インデクス作成方法に注意**してください。初期分割登録および分割遅延登録を選択すると、CREATE INDEX自体の実行時間は短くすみめますので、全文検索型以外の列に対する操作は、すぐに再開できます。

n-gramインデクスのインデクス作成方法の指定

HiRDB Text Search Plug-inのn-gramインデクスの作成方法指定には、以下の種類があります。

■ 即時登録

CREATE INDEX実行時にインデクスへ文書を登録します。文書追加時は、追加処理の延長でインデクスに登録します。

■ 初期分割登録

CREATE INDEX実行時には、文書をインデクスへ登録しません。インデクス作成開始ユティリティ(phnstartidx)を実行して、インデクスへの文書登録を実行します。すべての文書の登録が完了した以降は、文書追加処理の延長でインデクスに登録します。

■ 分割遅延登録

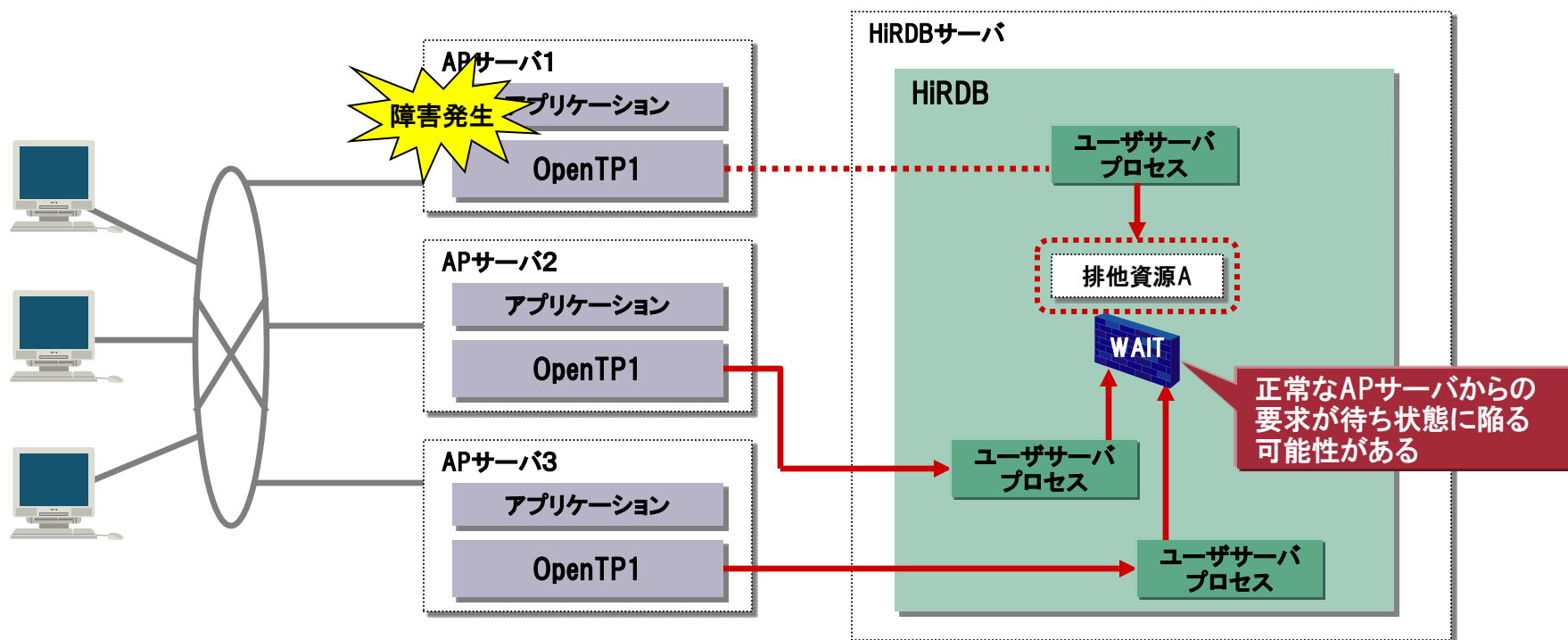
CREATE INDEX実行時には文書をインデクスへ登録しません。インデクス作成開始ユティリティ(phnstartidx)を実行して、インデクスへの文書登録を実行します。

事例

トランザクションマネージャに障害が発生した場合、長時間再起動せずに放置しておくと、HiRDBの排他資源などが解放されない状態に陥り、他のトランザクションマネージャからの要求が待ち状態になることがあります。

コツ

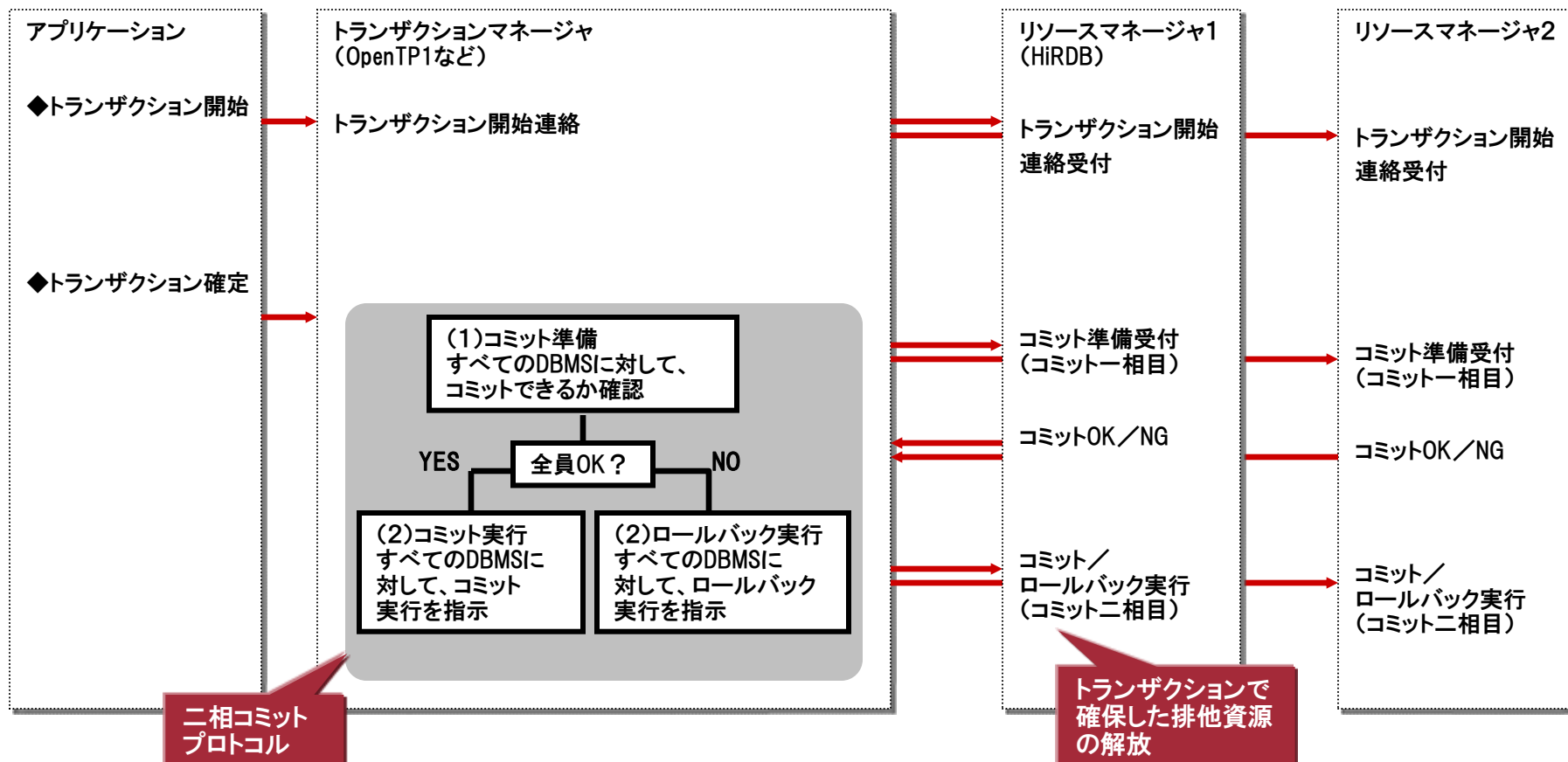
💡 OpenTP1などトランザクションマネージャを使用して分散トランザクションを管理しているシステムでは、トランザクションマネージャを必ず系切り替え構成にしてください。



解説

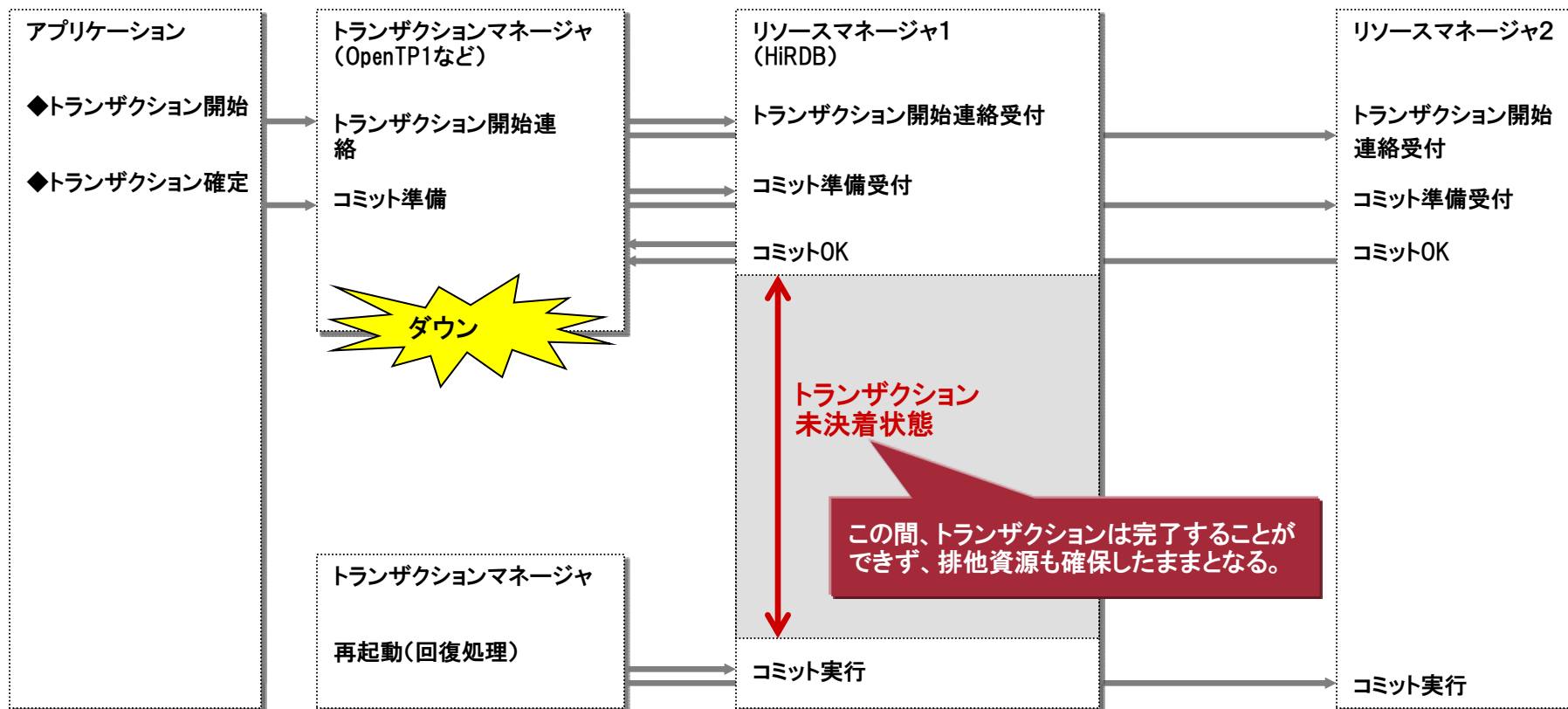
分散トランザクションシステムにおいて、トランザクションマネージャは、二相コミットプロトコルを使用して、トランザクションの終了処理を行います。

■トランザクションマネージャによるコミット処理



解説

コミット準備実行後、コミット実行の要求を発行する前に、トランザクションマネージャに障害が発生した場合、HiRDBの該当トランザクションは、未決着状態のまま残ります。トランザクションマネージャを再起動し、トランザクションを完了させる必要があります。



(※):トランザクションマネージャがすぐに起動できない場合、HiRDBのpdcmntコマンドを使用して、未決着状態のトランザクションを完了させることもできます。対策方法の詳細は、マニュアル「システム運用ガイド」の「OLTPシステムとの連携中に通信障害などが発生したときの対処方法」を参照してください。

事例

OpenTP1とXA連携した環境で、トランザクションの移行機能を使用しない(PDXAMODE=0)設定だと、xa_prepare()が失敗すると、その次のトランザクションで、xa_start()がエラーとなり、UAPが異常終了する、という事例があります。

コツ

💡 OpenTP1とXA連携した環境ではトランザクションの移行機能を使用する指定になっているか確認してください。

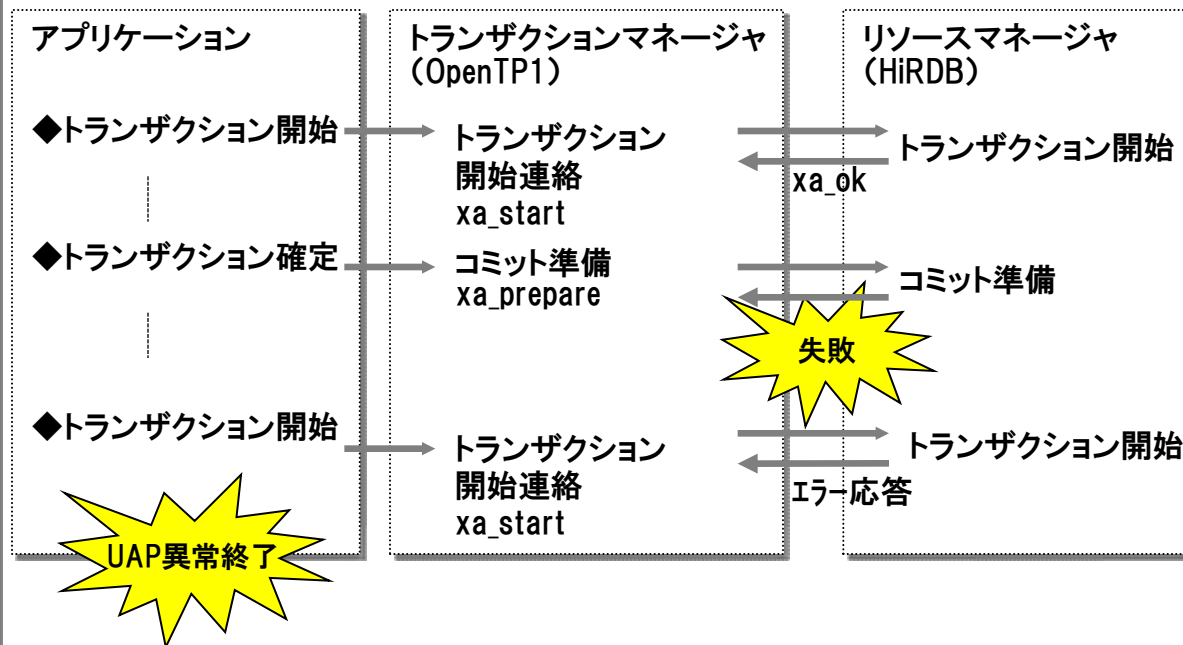
対策

この現象を回避するためには、トランザクションの移行機能を使用してください。また、トランザクションの移行を使用すると、OpenTP1のコミット最適化およびプリペア最適化にHiRDBが対応します。したがって、OpenTP1のtrnstringオペランドには-dオプションを指定しないでください。

【指定方法】

クライアント環境定義
PDXAMODE=1

TP1のトランザクションサービス定義
trnstringオペランドは-dオプションを省略



解説

HiRDB XAライブラリを使用してHiRDBに接続するUAPでは、HiRDBにアクセスしたときと異なるプロセスでトランザクションのコミット処理を実行できます。

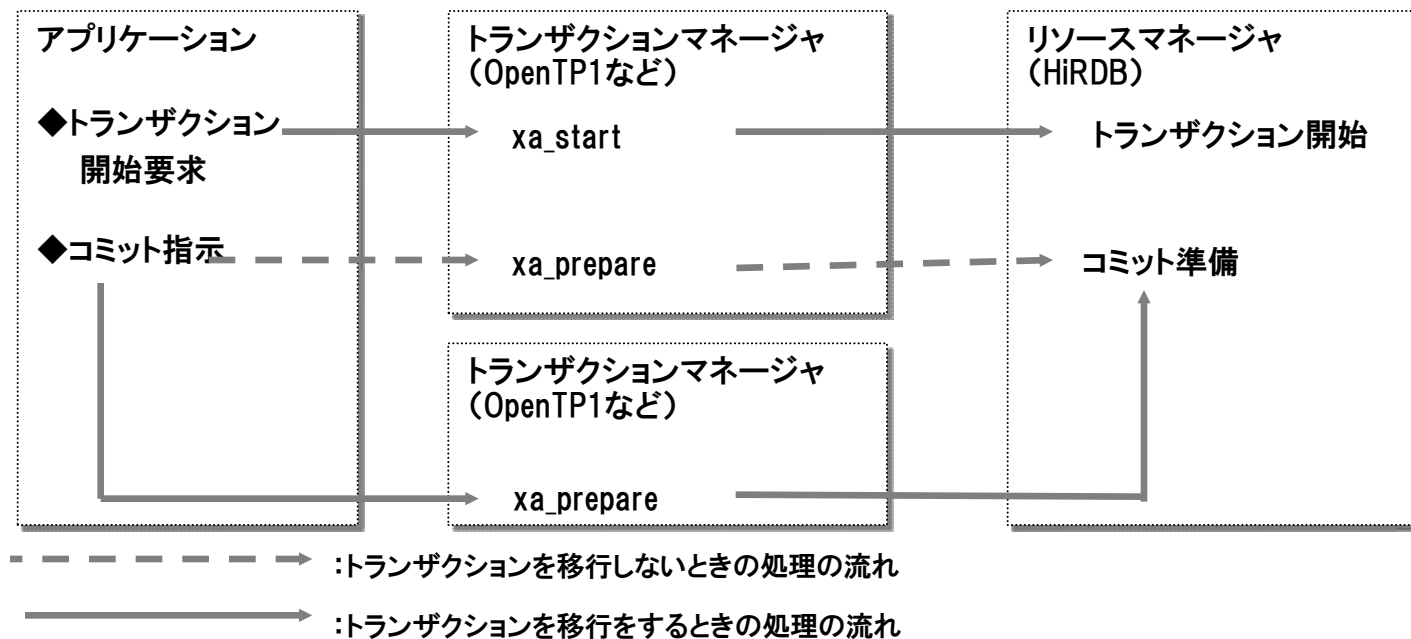
メリット

トランザクションの移行を使用すると、トランザクションマネージャのUAP処理はトランザクションの完了を待たなくても次回のサービス要求を受け付けられます。

このため、この機能を使用しないときに比べて、少ないプロセス数でUAPを実行できます。

ただし、HiRDBが使用するサーバプロセス数および排他待ち^(*)回数が増える場合があります。

(*)：トランザクションが完了するまでの間、次回のサービス要求によるHiRDBへのアクセスは、排他待ちになる場合が増えます。



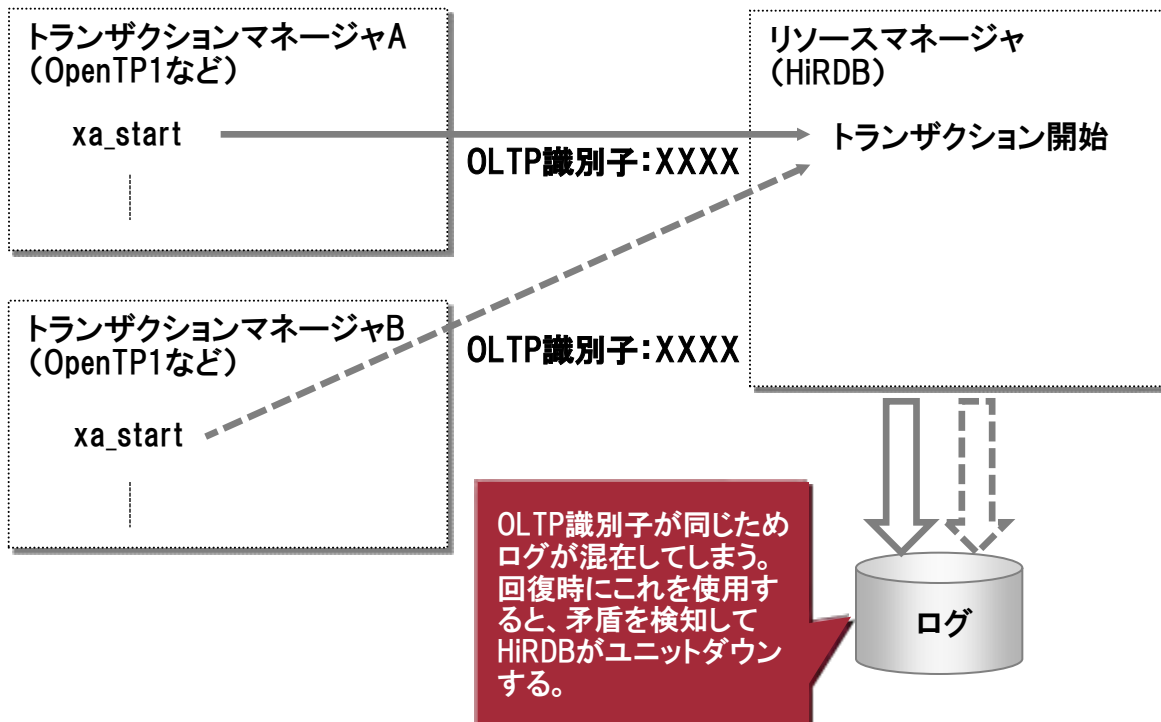
事例 複数のOpenTP1とXA連携した環境で、各OpenTP1定義のクライアント環境変数PDTMIDが、ユニーク値でなかったため、回復処理時に矛盾検知してHiRDBがユニットダウンするという事例があります。

コツ 複数のOLTPからX/Openに従ったAPIを使用して一つのHiRDBサーバをアクセスする場合、それぞれのOLTPにユニークな識別子を設定しているか確認してください。

対策

複数のOLTPからX/Openに従ったAPIを使用して一つのHiRDBサーバをアクセスする場合、それぞれのOLTPにユニークな識別子を指定してください。

【指定方法】
クライアント環境定義
PDTMID= OLTP識別子



事例 最適化情報の更新でトラブルが発生したが、更新前の最適化情報をバックアップしていなかったため、復旧に時間が掛かった、という事例があります。

コツ  最適化情報を更新する場合、トラブルに備えて更新前の最適化情報をバックアップとして保存しておいてください。

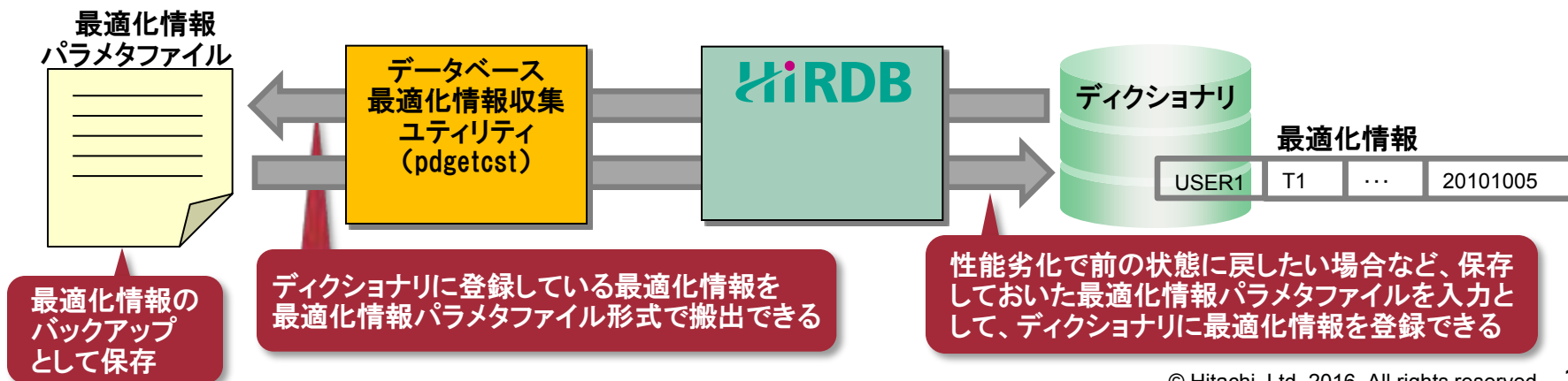
最適化情報パラメタファイルへの最適化情報の搬出

HiRDB Version 9 09-01でデータベース最適化ユティリティ(pdgetcst)に以下の機能を追加しています。
ディクショナリに登録している最適化情報を、最適化情報パラメタファイルの形式で搬出することができます。
本機能を用いて、最適化情報のバックアップとして保存したり、搬出した最適化情報パラメタファイルを用いて最適化情報の任意の項目を編集し、再登録するなどの運用が可能です。

【コマンド実行例】

```
pdgetcst -e /tmp/param_file -a USER1 -t T1
```

※最適化情報とは、表の行数、インデクス構成列のデータ分布情報など、表、列、およびインデクスに関する統計情報を集めたものである。
HiRDBは、コストベースでSQLの最適化を行い、自動的に最適なアクセスパスを決定する。最適化情報をディクショナリに登録しておくことで、最適化情報を利用して最適化の精度を向上し、より最適なアクセスパスを決定することができる。





付録B. トラブル発生時の対処方法



付録B.トラブル発生時の対処方法

付録B.1 概要

付録B.2 エラー発生

付録B.3 無応答

付録B.4 レスポンス遅延

付録B.5 トラブルシュート情報

運用時のトラブルは、主に以下の3種類の現象に分類されます。

- エラー発生
- 無応答
- レスポンス遅延

 各現象ごとに、原因の切り分け方法について解説します。

付録B.トラブル発生時の対処方法

付録B.1 概要

付録B.2 エラー発生

付録B.3 無応答

付録B.4 レスポンス遅延

付録B.5 トラブルシュート情報

【現象】

エラー発生

メッセージの
確認

【原因】

メッセージから
判断可能

タイムアウト

■クライアント側、ユティリティ・コマンド側
サーバに接続できない
サーバが起動していない

DB閉塞

HiRDB
起動不可

その他
(原因不明)

■サーバ側
メッセージなし

■サーバ側
ユーザサーバダウン

■サーバ側
システムダウン
ユニットダウン

稼働状態
確認

正常稼働中

停止中

【対策】

メッセージに従い
対策する

原因を排除し
再実行

接続設定の
見直し

保守員
連絡

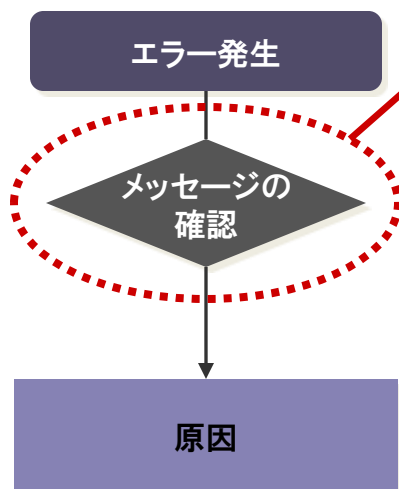
原因を排除し
再実行

原因を排除し
再起動

DB回復

原因を排除し
再実行

保守員
連絡



障害が発生した時刻付近に、エラーメッセージや警告メッセージが出ていないか、確認してください。
また、HiRDB以外のOSやディスクのエラーが出ていないか確認してください。

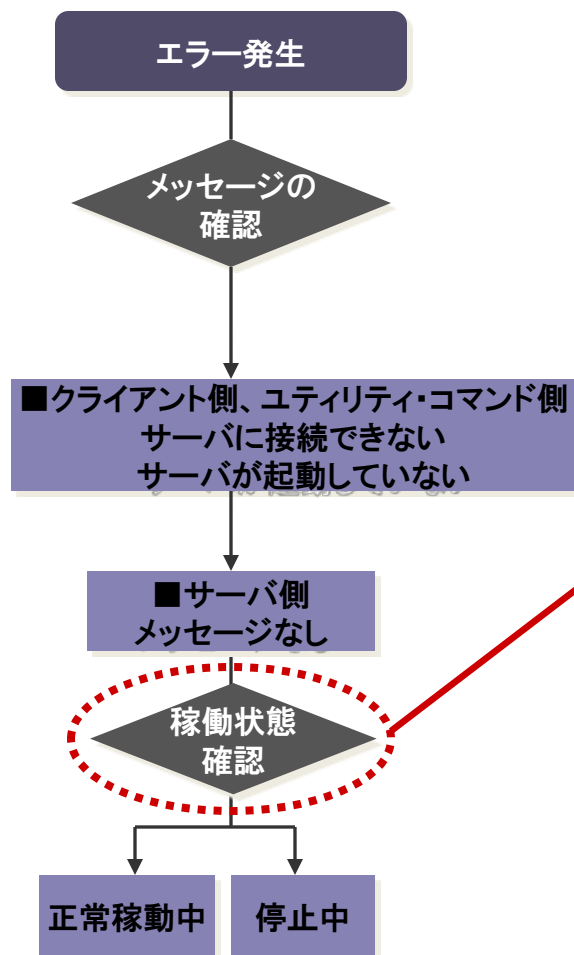
HiRDBのメッセージ種別

■KFPxxxxxx-I	: インフォメーション
■KFPxxxxxx-W	: 警告
■KFPxxxxxx-E	: エラー

HiRDBのメッセージ出力先

- HiRDBクライアントのエラーログ(アプリケーションの場合)
SQLのエラーメッセージが、クライアント環境変数PDCLTPATHに指定したディレクトリ下の、pderr1.trcおよびpderr2.trcというファイルに出力されます。
- 標準エラー出力、標準出力(ユティリティ、コマンドの場合)
実行結果を示すメッセージが、標準エラー出力や標準出力に出力されます。
- syslog^(*1)(Windowsの場合はイベントログ)
HiRDBのメッセージが出力されます。OSのビューアーで参照できます。
- メッセージログファイル
HiRDBのメッセージが出力されます。pdcatコマンドで参照できます。

(*1): OS(UNIX)のシステムログをsyslogと表記します。syslogは、/etc/syslog.confで設定したファイルに出力されます。



稼働状態確認方法

pdlsコマンドで、ユニットおよびサーバの起動状態が確認できます。

【コマンド実行例】

■HiRDB/シングルサーバの場合

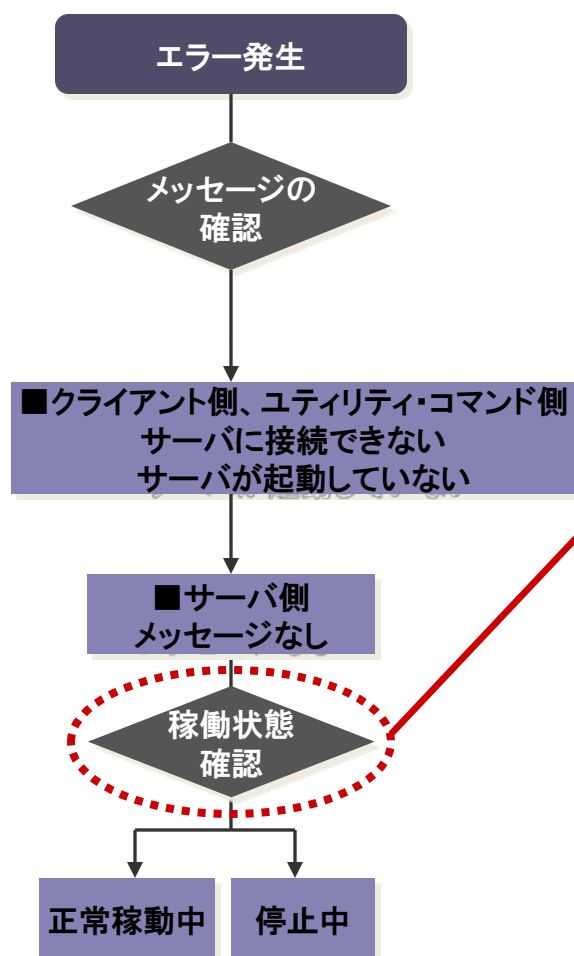
pdls	HOSTNAME(151452)	UNITID	SVID	STATUS	STARTTIME
	HOST1	unt1	*****	ACTIVE	151422
	HOST1	unt1	sds01	ACTIVE	151422

「STATUS」が「ACTIVE」の場合、起動していることを示しています。

■HiRDB/パラレルサーバの場合

pdls	HOSTNAME(150345)	UNITID	SVID	STATUS	STARTTIME
	HOST1	unt1	*****	ACTIVE	150335
	HOST1	unt1	fes1	ACTIVE	150329
	HOST1	unt1	bes2	ACTIVE	150335
	HOST1	unt1	bes1	ACTIVE	150335
	HOST1	unt1	dic1	ACTIVE	150258
pdls	HOSTNAME(150346)	UNITID	SVID	STATUS	STARTTIME
	HOST2	unt2	*****	ACTIVE	150333
	HOST2	unt2	bes4	ACTIVE	150332
	HOST2	unt2	bes3	ACTIVE	150333

HiRDB/シングルサーバおよびHiRDB/パラレルサーバのマネージャユニットが起動していない状態で、pdlsコマンドを実行した場合は、KFPS01853-Wのメッセージが出力されます。



系の状態確認方法

IPアドレスを引き継がない系切り替え機能またはスタンバイレス型系切り替え機能を使用している場合は、系の切り替えが発生しているかまたは系の状態も確認してください。

系の状態確認方法は、**pdls -d ha**コマンドで確認します。

【コマンド実行例】

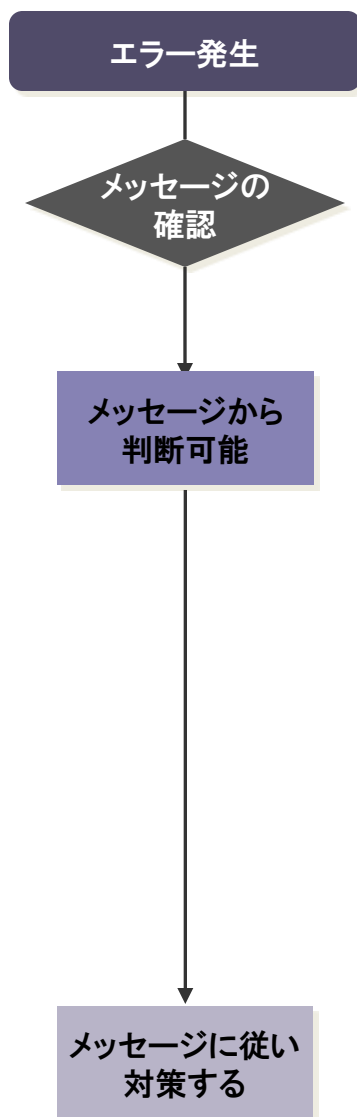
```
pdls -d ha
```

UNITID	INITIAL-HOST	[STATUS]	RESERVED-HOST	[STATUS]
unt1	HOST1	[ONL]	HOST2	[SBY]
unt2	HOST2	[ONL]	HOST1	[SBY]

現用系の状態を表示します。

予備系の状態を表示します。

コマンド実行結果の詳細は、マニュアル「コマンドリファレンス」を参照してください。



解説

エラーメッセージから原因が判明する場合は、メッセージの内容に従い、対策してください。

【例】

■SQLの使い方に誤りがある。

→SQLを修正してください。

■入力データの形式に誤りがある。

→入力データを修正してください。

■排他が確保できない。

→デットロックや排他待ちでタイムアウトになった場合の調査方法については、マニュアル「システム運用ガイド」の「システムの稼働環境を知る方法」の「デットロックが発生した場合」を参照してください。

■メモリなどのリソースが足りない。

→リソースを定義、もしくは拡張してください。

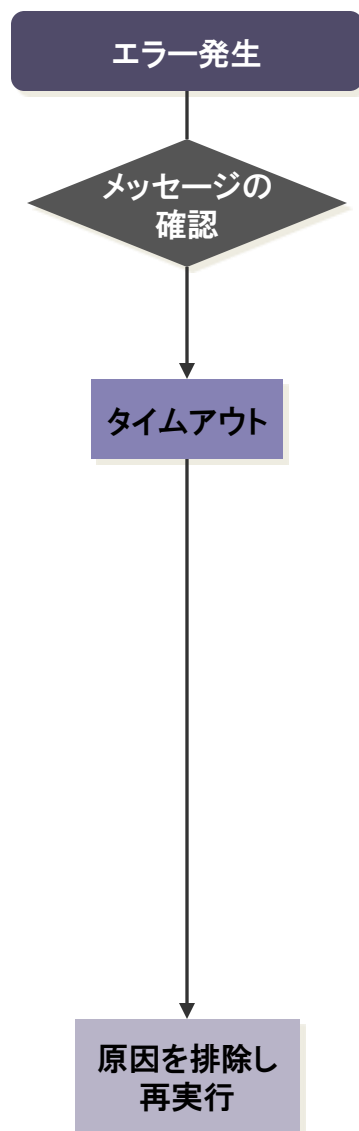
①表を結合する検索を行う場合は、作業表用ファイル(HiRDBシステム定義pdworkに指定するファイル)を使用しますので、SQLが確定したら、必ず見積もりをしてください。

②操作内容によっては、リソース拡張以外の方法で回避した方がよい場合もあります。



4-1

の事例が該当します。



解説

アプリケーションやユティリティ、コマンドにタイムアウトのメッセージが出力された場合は、タイムアウトに至った原因を調査し、原因を排除した上で、再実行してください。

【例】

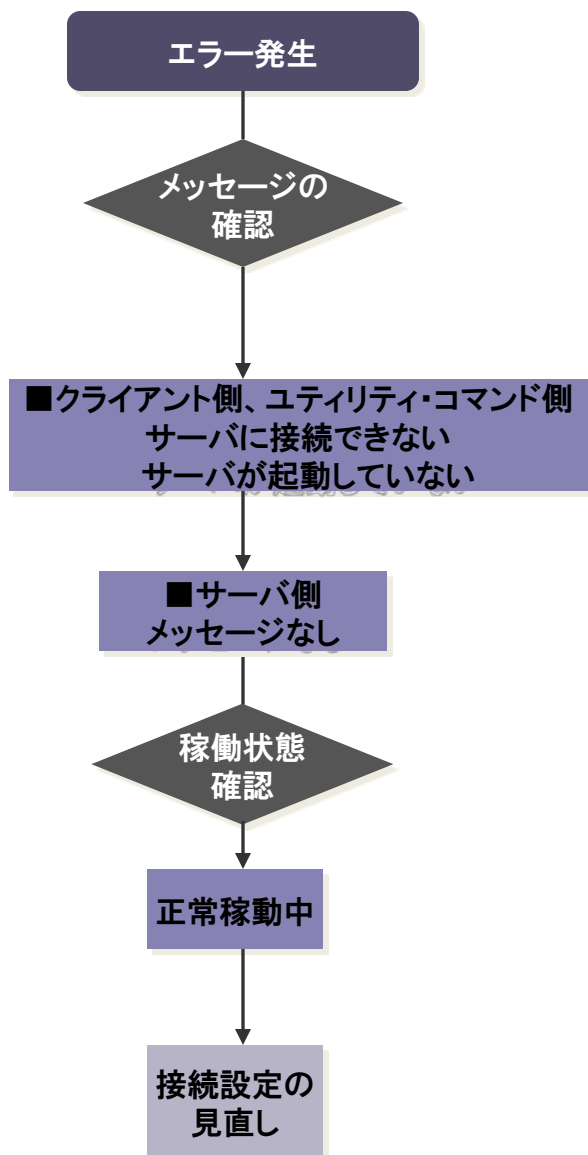
■SQLの実行に想定以上の時間がかかる。

→レスポンス遅延の調査方法 ▶ **付録B-4** を参照してください。

■ネットワークの障害により、発生する場合があります。

■監視時間の設定に問題がある場合があります。

▶ **2-1-1** に事例を紹介します。



解説

HiRDBサーバが稼働しているにもかかわらず、接続できない場合は、クライアント環境変数の設定に問題がある場合があります。

【例】

■クライアント環境変数の設定が不正。

→クライアント環境変数の設定方法には、複数の手段があります。
優先順位を考慮して、不要な設定が有効にならないよう注意してください。

▶ **4-2** に事例を紹介します。

■ファイアウォールの設定に問題がある場合もあります。

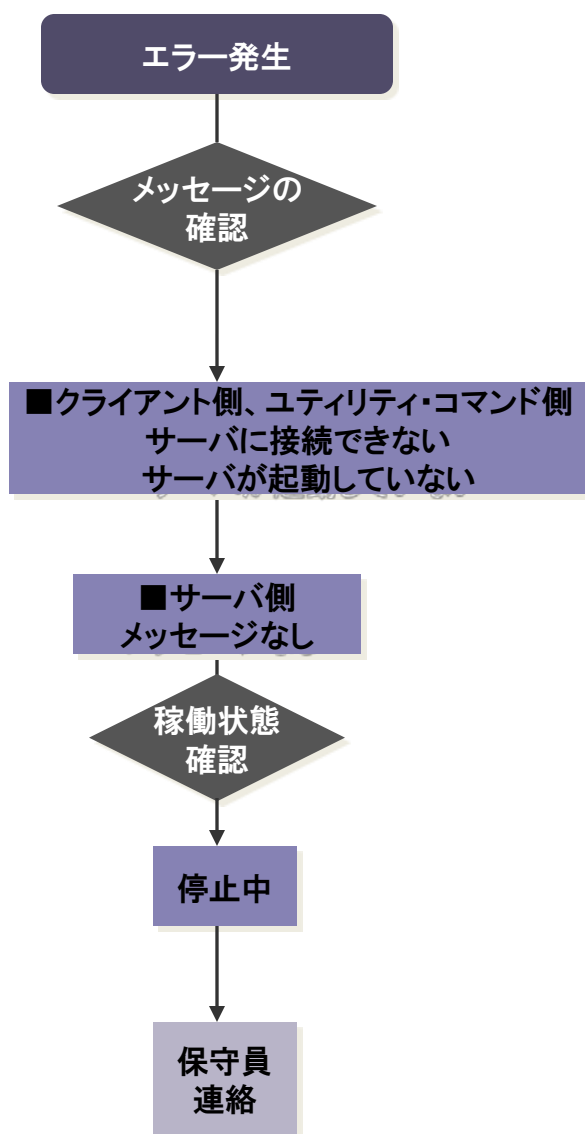
▶ **2-1-9** に事例を紹介します。

その他に以下のような事例もあります。

【例】

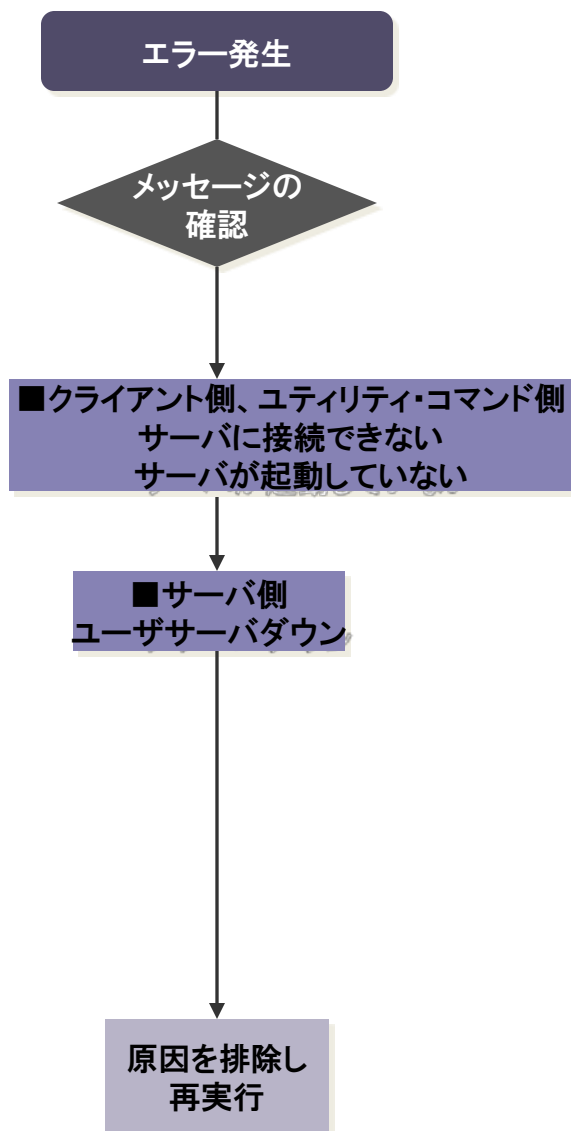
■TCPコネクションの残存により接続不可となる事例もあります。

▶ **2-1-4** に事例を紹介します。



解説

HiRDBサーバが停止した理由が不明な場合は、調査に必要な資料を収集して、サポートサービスまで連絡してください。
→調査に必要な資料については、**付録B. 5** で説明します。

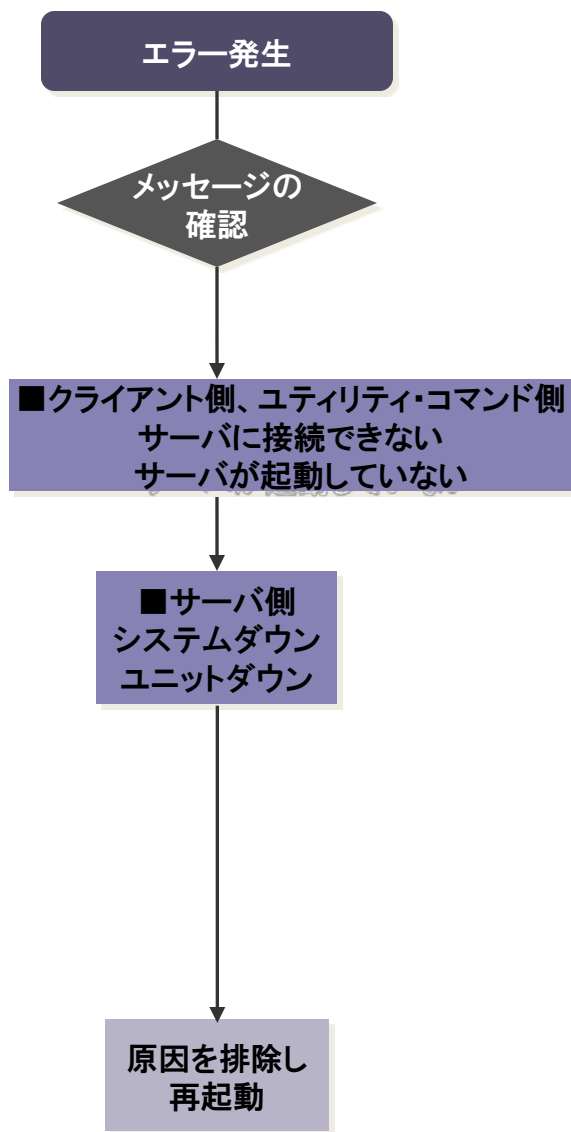


解説

メッセージの内容から原因を調査し、原因を排除した上で、再実行してください。

【例】

- サーバ待ち時間(クライアント環境変数PDSWAITTIME、PDSWATCHTIME)の指定を超えた場合、ユーザーサーバは終了します。接続をつないだままにしないよう、注意してください。
→Webアプリケーションなど、コネクションプーリング機能を使用している場合の設定方法については、タイムアウトの事例を参照してください。
- メッセージの内容から原因が判断できない場合は、調査に必要な資料を収集して、サポートサービスまで連絡してください。
同じ現象が繰り返し発生する場合は、トラブルシュート情報を早期に退避する必要があります。
→詳細は、**付録B. 5** で説明します。



解説

メッセージの内容から原因を調査し、原因を排除した上で、HiRDBサーバを再起動してから、処理を再実行してください。

【例】

■システムログファイルの容量不足によるシステムダウン

HiRDBで発生しやすい障害の1つです。

システムログファイルの容量不足に陥らないための運用について

▶ **2-5-10** で説明します。

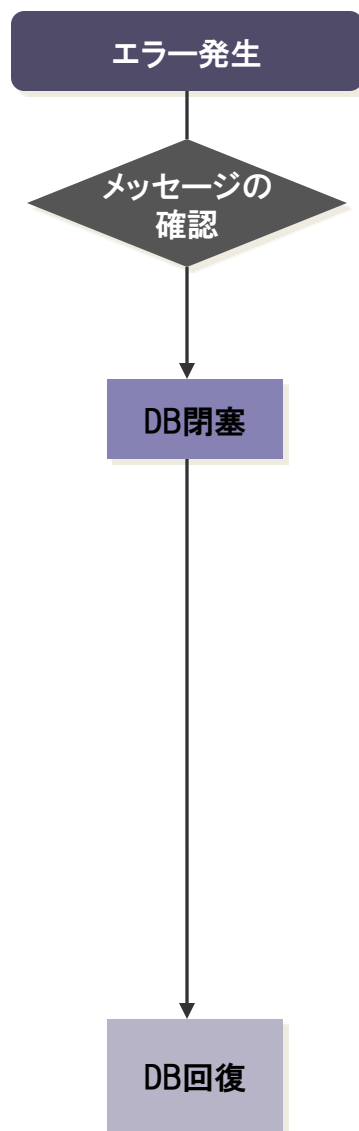
■HiRDBの設定により、ユニットダウンを誘発してしまう事例もあります。

▶ **2-1-11** に事例を紹介します。

■メッセージの内容から原因が判断できない場合は、調査に必要な資料を収集して、サポートサービスまで連絡してください。

システムダウンを繰り返す場合は、HiRDB運用ディレクトリの容量にも注意が必要です。

→詳細は、**付録B. 5** で説明します。



解説

RDエリアが閉塞した場合は、バックアップから回復した上で、再実行してください。

【例】

■ディスク障害により、データ入出力がエラーとなり、RDエリアが閉塞。
→ディスク障害対策後に、バックアップから回復してください。

回復方法の概要などについて、▶ **付録B-2-12** で説明します。

■回復手順の誤りによる障害事例があります。

▶ **2-5-1** に事例を紹介します。

解説 RDエリアのあるディスクで障害が発生した場合の回復手順について解説します。

回復手順

1. ハードディスクを初期化します。
2. パーティションを設定します。
3. ファイルシステムの初期化をします。
(キャラクタ型スペシャルファイルとして使用していた場合は不要です。)
4. HiRDBファイルシステム領域を初期設定します。
5. システムファイルを作成します。
6. RDエリアを回復します。

詳細は、マニュアル「システム運用ガイド」の「障害が発生したときの対処方法」の「ディスク障害が発生したときの対処方法」を参照してください。

注意事項

RDエリアの閉塞が発生した場合は、syslog(Windowsの場合はイベントログ)に、ハードウェア(ディスク)からのエラーメッセージが出力されていないかを確認してください。

ハードウェア障害の有無により、DB回復手順が異なりますので、HiRDBサポートサービスへは、ハード障害の状況を併せて連絡してください。

また、ハードウェア障害と思われる場合には、ハードウェアサポートサービスへの連絡もお願いします。



解説

syslog(Windowsの場合はイベントログ)に出力されたメッセージから、起動不可となった原因を調査し、原因を排除した上で、再度起動してください。

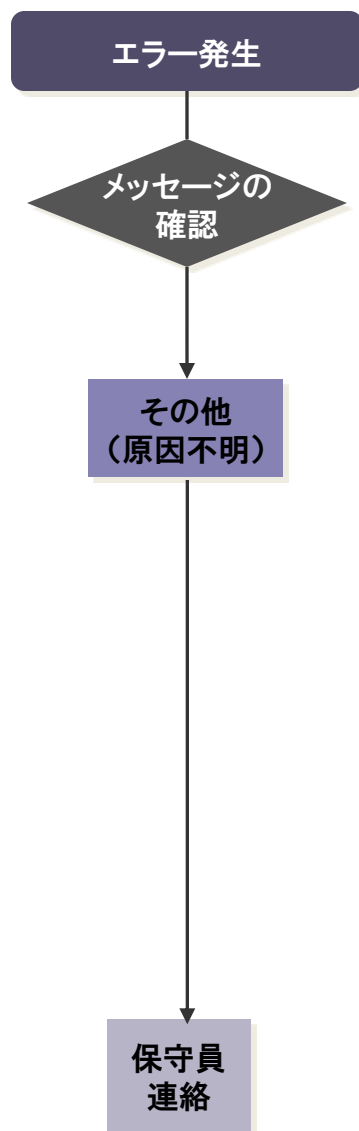
■システム構築時の場合は、構築時のトラブル事例を参照してください。

■Windowsの場合は、Application Experience Lookup Service (AELookupSvc)を停止するとOSがファイルをロックするため、HiRDB を起動することはできません。

▶ **2-5-4** に事例を紹介します。

■UNIXの場合は、HiRDBの起動に、HiRDB運用ディレクトリ以外にHiRDBが作成するディレクトリおよびファイルが必要です。

▶ **2-5-3** に事例を紹介します。



解説

エラーが発生した理由が不明な場合は、調査に必要な資料を収集して、サポートサービスまで連絡してください。

→調査に必要な資料については、**付録B. 5**で説明します。

■誤操作によって、意図しないデータを更新してしまう事例があります。

▶ **4-17** に事例を紹介します。



付録B.トラブル発生時の対処方法

付録B.1 概要

付録B.2 エラー発生

付録B.3 無応答

付録B.4 レスポンス遅延

付録B.5 トラブルシュート情報

【現象】

無応答

実行制御の
確認

【原因】

実行制御に
変化がある

実行制御に
変化がない

実行制御が
クライアント側

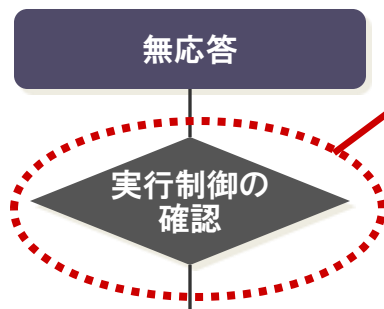
実行制御が
サーバ側

【対策】

性能
調査

アプリケーション側
を調査

保守員
連絡



無応答に陥った場合は、pdls -d rpcを実行して、実行制御がどこにあるかを確認してください。

■アプリケーションの場合

①「Client Information」の「PID」が、アプリケーションのプロセスIDに該当する項番を探します。

②同じ項番のシングルサーバもしくはフロントエンドサーバの「STATUS」を確認します。

【STATUSの表示】

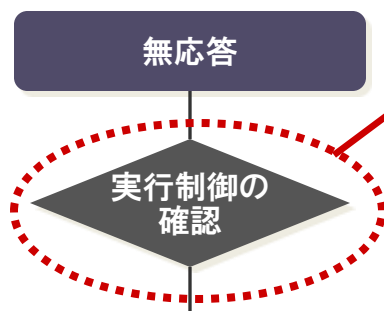
- ・SUSPEND(CLT): 実行制御はHiRDBクライアント側
- ・SUSPEND(QUE): 実行制御はHiRDBクライアント側(キュー待ち)
- ・SUSPEND(SVR): 実行制御はHiRDBサーバ側(受信待ち状態)
- ・ACTIVE: 実行制御はHiRDBサーバ側 → ③この場合は、「SYS_EVENT」の内容も確認します。

④繰り返し実行して、「STATUS」および「SYS_EVENT」の表示に変化があるか確認してください。

pdls -d rpc実行例(アプリケーションの場合)

```

pdls -d rpc
[0001]
Client Information :
  PID      ADDR      NAME      OS      GROUP      XA
  3508     10.209.76.169  UAP1      P      PC          N
Server Information :
  PID      SVID      STATUS      SYS_EVENT      USR_EVENT      CRITICAL(MASK)  SERVICE      SYS_CALL
  2612     sds01     SUSPEND(CLT) 0x00000000 00000000 00000402  0x00000000    N(0x00000000)  p_f_sqa_csvrinf  0x0300007e
[0002]
① Client Information :
  PID      ADDR      NAME      OS      GROUP      XA
  576      10.209.76.169  UAP2      P      PC          N
Server Information :
  PID      SVID      ② STATUS      SYS_EVENT      USR_EVENT      CRITICAL(MASK)  SERVICE      SYS_CALL
  2664     sds01     SUSPEND(CLT) 0x00000000 00000000 00000402  0x00000000    N(0x00000000)  p_f_sqa_ccommit  0x0300007e
pdls -d rpc
[0001]
<中略>
[0002]
Client Information :
  PID      ADDR      NAME      OS      GROUP      XA
  576      10.209.76.169  UAP2      P      PC          N
Server Information :
  PID      SVID      ③ STATUS      SYS_EVENT      USR_EVENT      CRITICAL(MASK)  SERVICE      SYS_CALL
  2664     sds01     ACTIVE      0x00400000 00000000 00010000  0x00032009    Y(0x00000000)  p_f_sqa_cfetchn  0x01000076
  
```



無応答に陥った場合は、pdls -d rpcを実行して、実行制御がどこにあるかを確認してください。

■ユーティリティの場合

①「Server Information」の「PID」が、ユーティリティのプロセスIDに該当する項番を探します。

②同じ項番のシングルサーバもしくはフロントエンドサーバの「STATUS」を確認します。

【STATUSの表示】

・SUSPEND(SVR):実行制御はHiRDBサーバ側(受信待ち状態)

・ACTIVE:実行制御はHiRDBサーバ側 → ③この場合は、「SYS_EVENT」の内容も確認します。

④繰り返し実行して、「STATUS」および「SYS_EVENT」の表示に変化があるか確認してください。

pdls -d rpc実行例(ユーティリティの場合)

pdls -d rpc
[0001]

Client Information :

PID	ADDR	NAME	OS	GROUP	XA
2540	10.209.76.169	pdzmk2	P	PC	N

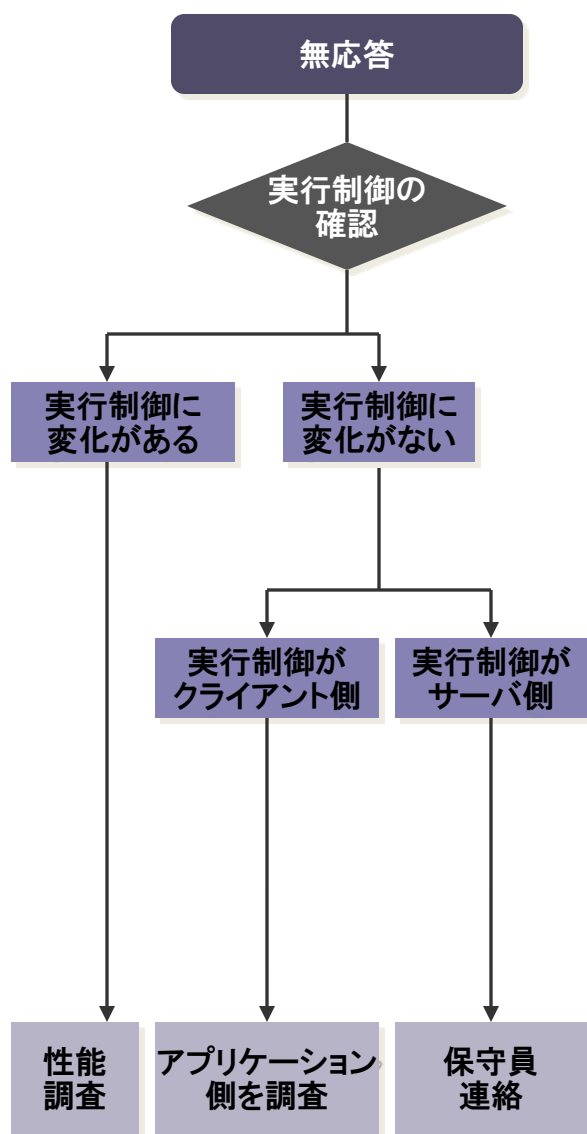
Server Information :

PID	SVID	STATUS	SYS_EVENT	USR_EVENT	CRITICAL(MASK)	SERVICE	SYS_CALL
1832	sds01	SUSPEND(CLT)	0x00000000 00000000 00000402	0x00000000	N(0x00000000)	p_f_sqa_csvrinf	0x0300007e

[0002]

Server Information :

PID	SVID	STATUS	SYS_EVENT	USR_EVENT	CRITICAL(MASK)	SERVICE	SYS_CALL
2416	sds01	ACTIVE	0x00400000 00000000 00020000	0x0003200a	Y(0x10000000)	p_f_lod_slav_serv_02	0x00000000
① 4016	_pd0load	SUSPEND(SVR)	0x00000000 00000000 00000208	0x00052120	N(0x00000000)	0x1c00007e	0x1c00007e
3228	0mload0	SUSPEND(SVR)	0x00000000 00000000 00000208	0x00052123	N(0x00000000)	p_f_lod_mstr_service	0x1c00007e



解説

- 実行制御の表示に変化がある場合は、処理を実行中であるため、想定以上に時間を要していると考えられます。
→ レスポンス遅延の調査方法 **付録B. 4** を参照してください。
- 実行制御に変化がなく、クライアント側に制御がある場合は、アプリケーション側に原因がないかを調査してください。
- 実行制御に変化がなく、サーバ側に制御がある場合は、調査に必要な資料を取得して、保守員に連絡してください。
→ 調査に必要な資料については、**付録B. 5** で説明します。

付録B.トラブル発生時の対処方法

付録B.1 概要

付録B.2 エラー発生

付録B.3 無応答

付録B.4 レスポンス遅延

付録B.5 トラブルシュート情報

【現象】

レスポンス遅延

変更点の
把握

【原因】

環境を変更

処理量の増加

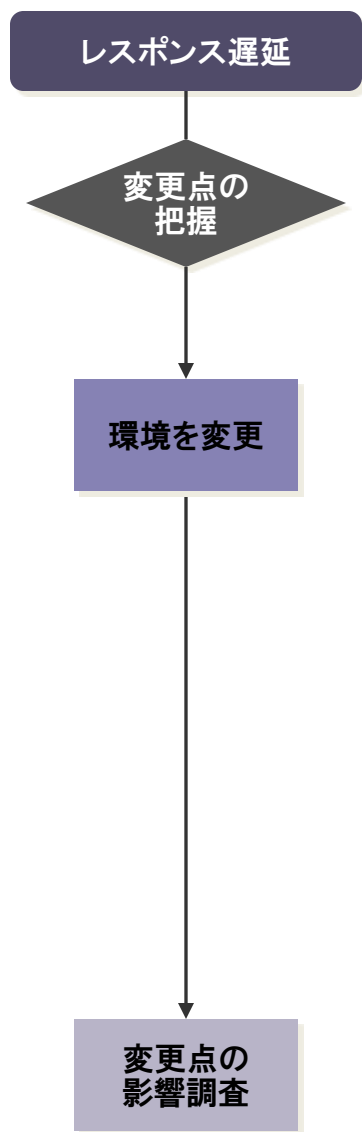
変更なし
(不明)

【対策】

変更点の
影響調査

増加部分の
影響調査

性能調査



解説

レスポンス遅延が発生した場合は、発生契機となる環境の変更がないか確認してください。

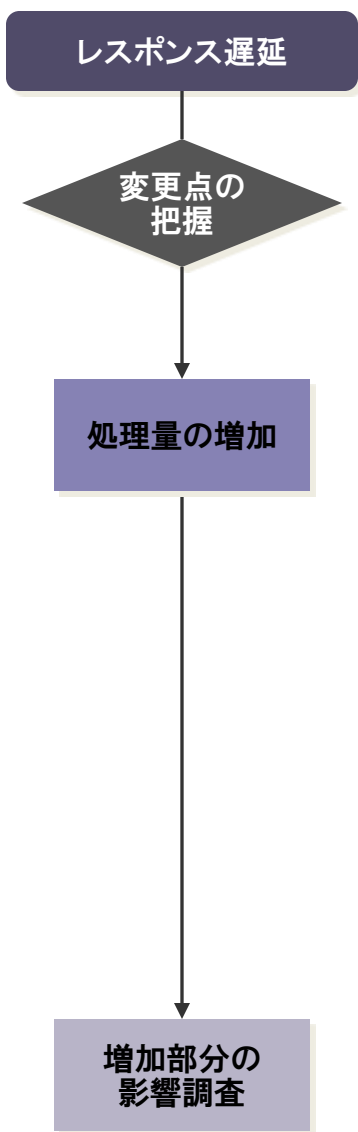
UAPを変更した、もしくはOSやHiRDBのパラメタを変更した場合は、変更点の影響に関して調査してください。

■SQLのチューニングが必要な場合

→性能調査のページ **付録B-4-5～B-4-12** にて、説明します。

■パラメタ設定値が適切でない場合、結果として性能劣化に陥る場合があります。

▶ **4-5** に事例を紹介します。



解説

データ量やトランザクション量が増加している場合は、以下の原因が想定されます。

■インデクスを定義していなかったため、データの全件検索を行っていたが、データ量が増えたことにより、性能問題が顕在化。

→表を設定する段階から、インデクス定義を検討することが必要です。

■データ量、トランザクション量が増えたことにより、グローバルバッファのヒット率が低下し、性能劣化に陥る。

→グローバルバッファのヒット率は80%以上であることが目安です。

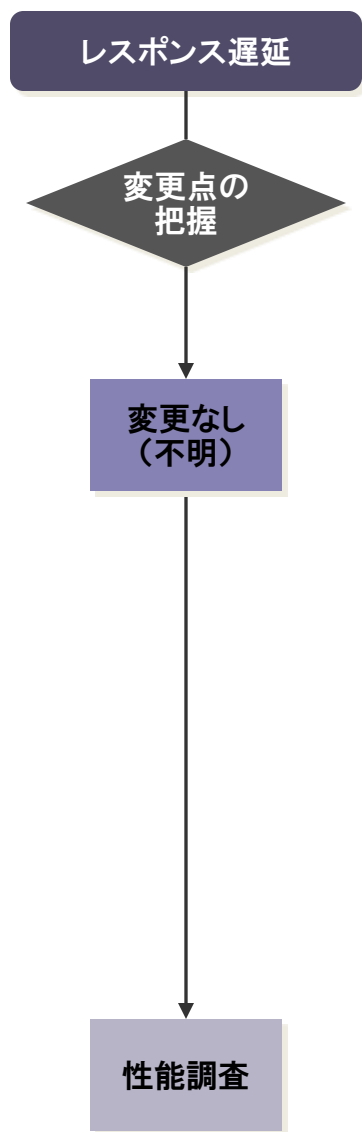
データ量、トランザクション量の推移に合わせて、拡張を検討してください。

ただし、DBの断片化により、グローバルバッファのヒット率が低下している場合があります。定期的にDBの格納状態を確認し、適時DB再編成を実施してください。

■トランザクション量が増加したことにより、排他待ちやデッドロックが発生し、性能劣化に陥っている。

→性能劣化の原因を特定するためには、正常稼働時から統計情報の取得をお願いします。

取得が必要な統計情報については、性能調査のページ **付録B-4-5～B-4-12**にて説明します。



解説

DB格納状態の変化などによって、レスポンス遅延が発生することもあります。

■DB容量の増加に伴い発生するレスポンス遅延の事例があります。

▶ **2-2-1** に事例を紹介します。

■DBメンテナンス運用が、オンライン業務に影響を与えることがあります。

▶ **2-1-12、2-5-5** に事例を紹介します。

レスポンス遅延の要因が明確でない場合は、性能調査を実施します。

■統計情報を解析し、問題点を把握します。

→性能劣化の原因を特定するためには、正常稼働時から統計情報を取得しておいてください。正常稼働時と性能劣化発生時の情報を突き合わせて、問題点を絞り込みます。

取得が必要な統計情報について、▶ **付録B-4-5** で説明します。

■正常稼働時の情報がない場合は、どこに時間がかかっているかを調査します。

SQL実行時間の確認方法について、▶ **付録B-4-11～B-4-12** で説明します。

取得する統計情報

■OSの統計情報

- ・CPU利用率
- ・ディスク利用率 など

■HiRDBの統計情報

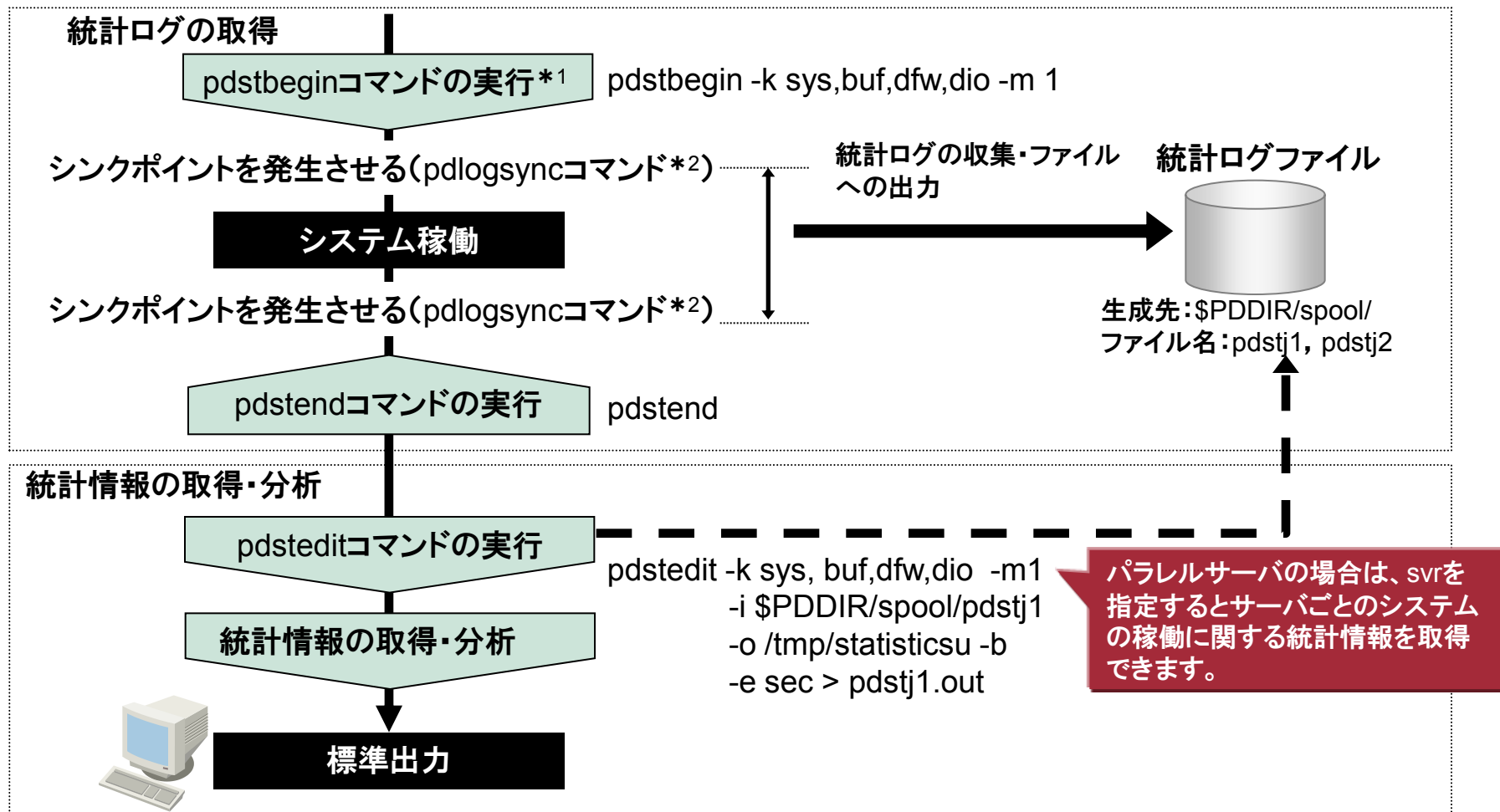
- ・システムの稼働に関する統計情報
- ・グローバルバッファプールに関する統計情報^(*)
- ・デファードライト処理に関する統計情報
- ・データベースの入出力に関する統計情報(バージョン09-50サポート)

HiRDBの統計情報取得方法の詳細については、**付録B-4-6** を参照してください。

統計情報の確認項目については、**付録B-4-7～B-4-10** を参照してください。

(*)：グローバルバッファプールに関する統計情報は、上記の取得方法以外にpdbufsコマンドで取得することもできます。

■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。

運用コマンド、ユティリティの詳細は、HiRDBマニュアル「コマンドリファレンス」-「pdstbegin(統計情報の出力開始)」または「統計解析ユティリティ(pdstedit)」を参照してください。

■問題点切り分け時、主に確認する項目【システム稼働統計情報(sys)】

フィールド名(*1)		内容
TRANSACTION	COMMIT	COMMIT回数
	ROLLBACK	ROLLBACK回数
LOCK	WAIT TIME	排他待ち時間
	QUEUE LEN	排他待ちユーザ数
	DEADLOCK	デッドロック件数
	USE LOCK TABLE	排他制御用プールパーティション使用率
SYNC POINT	GET INTERVAL TIME	シンクポイントダンプ取得間隔時間
LOG	WAIT THREAD	カレントバッファなしによる待ち回数
	OUTPUT BLOCK LEN	出力ブロック数
	WAIT BUFFER FOR IO	入出力待ちバッファ面数
DICTIONARY	TBL-DEF GET REQ	表定義情報取得要求回数
	TABLE CACHE HIT	表定義情報用バッファヒット回数
FES-BES-DIC(SDS) INFORMATION	SQLOBJ GET REQ	SQLオブジェクト取得要求回数
	SQLOBJ CACHE HIT	SQLオブジェクト用バッファヒット回数
RPC	RESPONSE ON OWN UNIT	自ユニットのサーバへのサービスレスポンス時間
	RESPONSE TO OTHER UNIT	他ユニットのサーバへのサービスレスポンス時間
	EXEC TIME ON OWN UNIT	自ユニットのサーバから1サービス当りの実行時間
	EXEC TIME FROM OTHER UNIT	他ユニットのサーバから1サービス当りの実行時間

(*1):HiRDB統計情報を、DAT形式で出力した場合のフィールド名称です。

■問題点切り分け時、主に確認する項目【グローバルバッファプールの統計情報(buf)】

フィールド名(*1)	内容
BUFFER NAME	グローバルバッファ名
LOG GET TIME	統計ログ取得時刻
UPDATE GET	更新GET回数
UPDATE HIT RATIO	更新バッファヒット率
UPDATE FLUSH	更新バッファフラッシュ回数
REF GET	参照GET回数
REF HIT RATIO	参照バッファヒット率
REF FLUSH	参照バッファフラッシュ回数
READ	実READ回数
WAIT READ	入力待ち発生回数
WRITE	実WRITE回数
WAIT WRITE	出力待ち発生回数
WAIT EXCLUSIVE	バッファ排他待ち回数
BUFHIT RATIO	グローバルバッファのヒット率
LOCK WAIT CNT	排他競合待ち回数

(*1):HiRDB統計情報を、DAT形式で出力した場合のフィールド名称です。

HiRDBの性能チューニングにおいて、グローバルバッファが最も効果が大きい項目になりますので、まずグローバルバッファのヒット率を確認してください。

■問題点切り分け時、主に確認する項目【データベースの入出力に関する統計情報(dio)】

フィールド名	内容
LOG GET TIME	統計ログを取得した時刻
READ CNT	取得時間間隔内に発生したread回数
READ MEASURE CNT	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 TIME(MICRO) SUM	時間を計測したreadの時間のマイクロ秒の合計値
WRITE CNT	取得時間間隔内に発生したwrite回数
WRITE MEASURE CNT	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 TIME(MICRO) SUM	時間を計測したwriteの時間のマイクロ秒の合計値

解説

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

$$\begin{aligned} \text{READ単価[us]} &= \text{「READ TIME(MICRO) SUM」} \div \text{「READ MEASURE CNT」} \\ \text{WRITE単価[us]} &= \text{「WRITE TIME(MICRO) SUM」} \div \text{「WRITE MEASURE CNT」} \end{aligned}$$

■ 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
2014/11/10 23:29:44		115	115	150034
2014/11/10 23:29:44		114	114	441211
2014/11/10 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
:

SQLオブジェクト用バッファ統計情報

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、実行回数、ディスクI/O回数などが確認できます。



バージョン09-50で、実行時間が
マイクロ秒単位で表示できるようになりました。

この統計情報は、パラメタなど事前の環境設定をすることなく、取得できることが特長です。

この統計情報から、性能上問題のあるSQL文を特定してください。

問題となるSQL文は、SQLトレースとUAP統計レポートを取得して再度実行し、問題の要因を調査してください。

取得方法

SQLオブジェクト用バッファの統計情報は、pdobilsコマンドで表示できます。取得方法については、HiRDBマニュアル「パフォーマンスガイド」

—「SQLオブジェクト用バッファの統計情報の確認方法」を参照してください。

また、pdobilsコマンドの詳細は、

HiRDBマニュアル「コマンドリファレンス」

—「pdobils(SQLオブジェクト用バッファの統計情報表示)」を参照してください。

確認項目

SQLオブジェクト用バッファの統計情報のうち、初期調査で確認する項目については、

HiRDBマニュアル「パフォーマンスガイド」

—「SQLオブジェクト用バッファの統計情報の確認方法」

—「ここは必ず確認しよう」

を参照してください。

SQLトレース

SQLトレースでは、SQLのオペレーションコードごとの実行時間などが確認できます。



バージョン09-50で、トレースのフォーマットが新しくなり、より詳細な情報がわかるようになりました。

UAP統計レポート

Update

アプリケーション実行時のSQLに関する情報を提供する機能です。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、アクセスパス情報が確認できます。

取得および調査方法

以下のクライアント環境変数を設定します。

- **PDCLTPATH** : トレースの出力先ディレクトリを指定します。
- **PDSQLTRACE** : トレースファイルの最大サイズを指定します。
- **PDUAPREPLVL** : UAP統計レポートの取得レベルを指定します。

- ・取得方法については、HiRDBマニュアル「パフォーマンスガイド」
—「SQLトレースとUAP統計レポートの取得方法」を参照してください。
- ・調査方法については、HiRDBマニュアル「パフォーマンスガイド」
—「SQLトレースとUAP統計レポートの確認方法」を参照してください。

解説

HiRDB Control Manager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。起動状態の確認や、syslogのメッセージの参照も、容易に行えます。

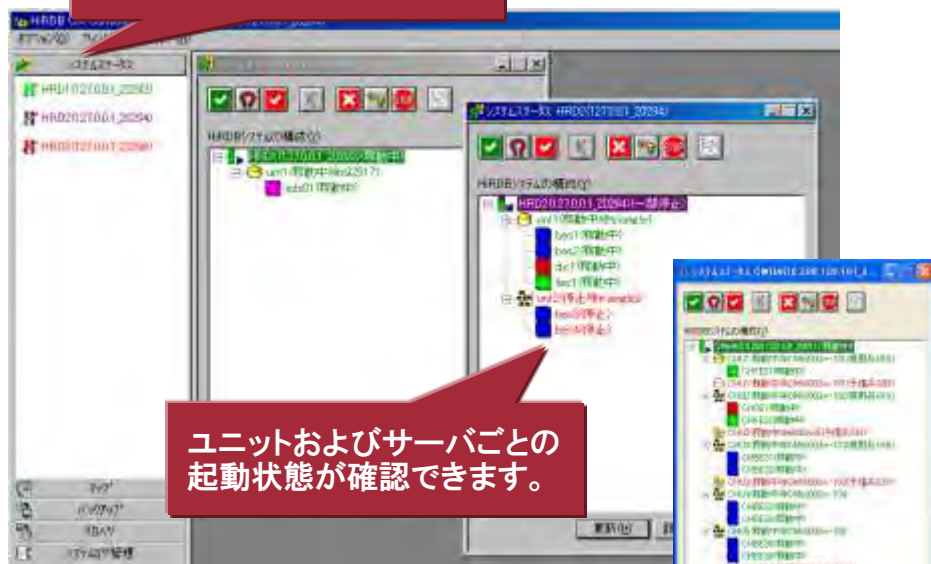
CM-Agentは、全OSのHiRDBサーバに標準付属！

CM-Server、CM-Consoleは、Windows版HiRDBサーバに標準付属！

■HiRDBの起動状態の確認

HiRDBの起動状態が、視覚的に確認できます。

起動中は緑、停止中は赤で表示されます。

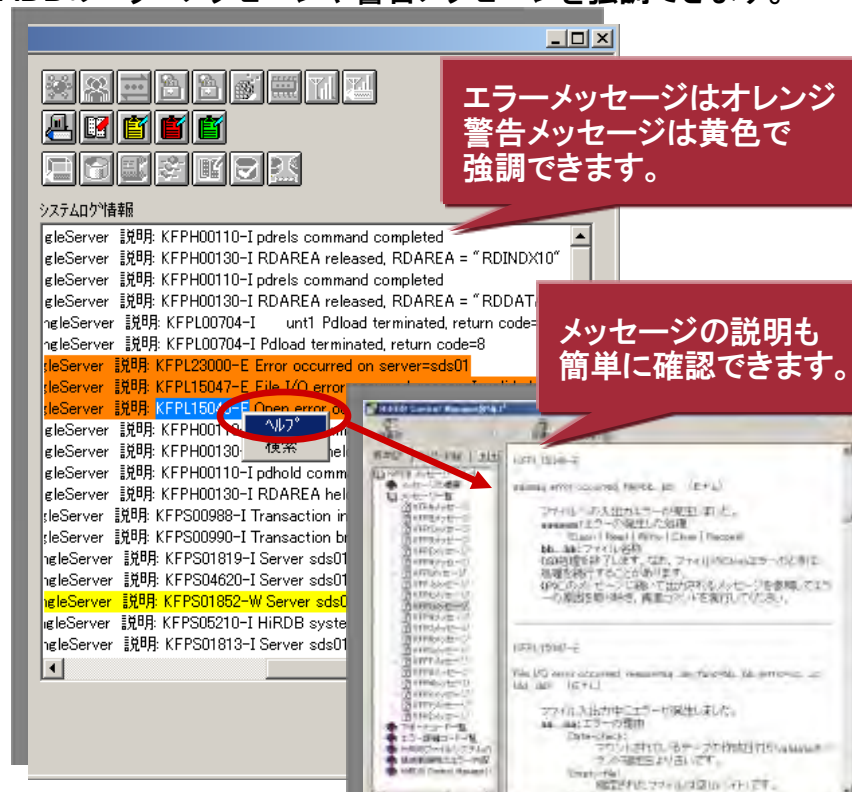


系切り替え構成の場合、現用系・予備系の構成と各系の状態が確認できます。

■syslog(Windowsの場合はイベントログ)の表示

syslogのメッセージを表示できます。

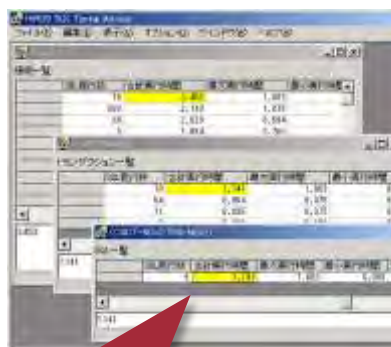
また、HiRDBのエラーメッセージや警告メッセージを強調できます。



解説 HiRDB SQL Tuning Advisor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。

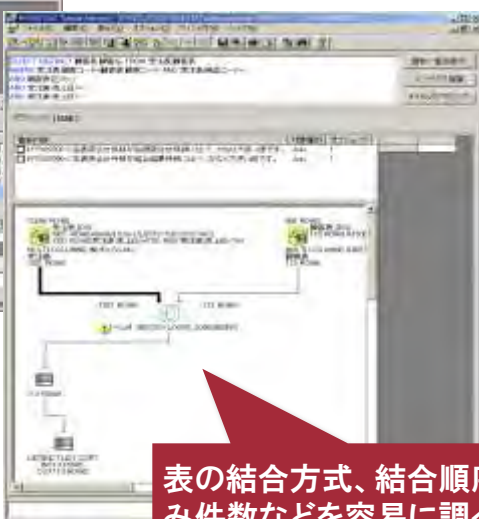
原因究明・分析

■トレース解析機能



コネクション、SQLなどの集計、着目したいUAP名でのフィルタリング、任意の列でのソートなどで様々な分析ができます。

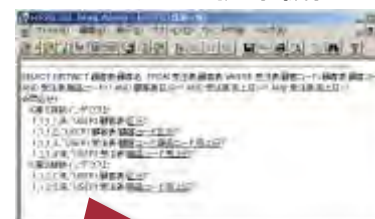
■アクセスパス解析機能



表の結合方式、結合順序、絞り込み件数などを容易に調べることができます。

対処

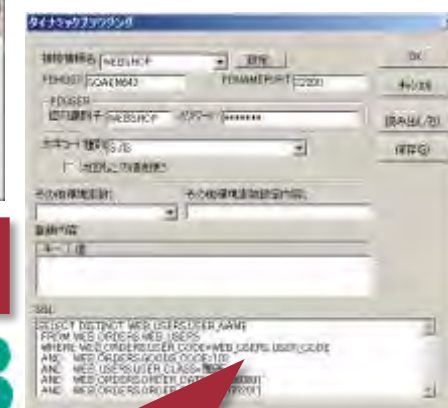
■インデックス提案機能



追加するインデックスを候補の中から検討できます。



■アクセスパスの事前確認機能



修正したSQLのアクセスパスをアプリケーションを修正する前に確認できます。

解説

uCosminexus Application Serverでは、HiRDBとの接続に使用しているコネクションを一意に識別するための接続情報(コネクションID)を性能解析トレース(PRFトレース)に出力しています。これらの情報を突き合わせて確認することで、トラブルが発生したコネクションを特定できます。

■性能解析トレース出力例

HiRDBのサーバ名、コネクト番号、サーバのプロセスIDが「INT」欄に出力されます。(*1)

	A	B	C	D	E	F	G	T	U	V
	PRF	Process	Thread(hashcode)	Trace	ProcessName	Event	Date	ReceiveSCD	INT	OPR
1	PRF									
2	Rec	1048	2580(5990238)	54	MyServer	0x8c00	2005/	****		
3	Rec	1048	2580(5990238)	55	MyServer	0x8b00	2005/	****		
4	Rec	1048	2580(5990238)	56	MyServer	0x8b80	2005/	****		
5	Rec	1048	2580(5990238)	57	MyServer	0x8b81	2005/	****		
6	Rec	1048	2580(5990238)	58	MyServer	0x8b01	2005/	****		
7	Rec	1048	2580(5990238)	59	MyServer	0x8c01	2005/	****	sds01:2:3380	
8	Rec	1048	2580(5990238)	60	MyServer	0x8c34	2005/	****		
9	Rec	1048	2580(5990238)	61	MyServer	0x8c35	2005/	****		
10	Rec	1048	2580(5990238)	62	MyServer	0x8ccc	2005/	****		
11	Rec	1048	2580(5990238)	63	MyServer	0x8ccd	2005/	****		
12	Rec	1048	2580(5990238)	64	MyServer	0x8d40	2005/	****		
13	Rec	1048	2580(5990238)	65	MyServer	0x8d41	2005/	****		
14	Rec	1048	2580(5990238)	66	MyServer	0x8c20	2005/	****	sds01:2:3380	
15	Rec	1048	2580(5990238)	67	MyServer	0x8c21	2005/	****		

<説明>INT:「(サーバ名):(コネクト番号):(サーバのプロセスID)」
(例)「sds01:2:3380」

(*1):この情報は、以下のイベントIDの時に出力されます。

0x8C01:DataSource.getConnection()メソッドの終了直前の処理で出力された情報

0x8C03:DataSource.getConnection(String username, String password)メソッドの終了直前の処理で出力された情報

0x8C20:Connection.close()メソッドの開始直後の処理で出力された情報

0x8C40:ManagedConnection.associateConnection()メソッドの呼び出し時の処理で出力された情報

■SQLトレース出力例

UAP実行ステータス情報の中に、サーバ名、コネクト番号、サーバのプロセスIDが出力されます。UAP実行ステータス情報は、HiRDB接続時に出力されます。

CONNECTION STATUS :

CURHOST(triangle)

CURPORT(4543) SVRNAME(sds01)

CNCTNO(2) SVRPID(3380) CLTPID(2292) CLTTID(0)

<説明>

SVRNAME(サーバ名):sds01

CNCTNO(コネクト通番):2

SVRPID(サーバのプロセスID):3380

性能解析トレースとSQLトレースを突き合わせて、
トラブルが発生したコネクションを特定してください。



付録B.トラブル発生時の対処方法

付録B.1 概要

付録B.2 エラー発生

付録B.3 無応答

付録B.4 レスポンス遅延

付録B.5 トラブルシュート情報

解説 障害調査のためには、以下のトラブルシューティング情報が必要となります。

トラブルシューティング情報		SQL エラー	ユティリティ、 コマンド エラー	無応答	レスポンス 遅延
HiRDB クライアント	エラーログファイル (クライアント環境変数PDCLTPATHの下の pderr*.trc)	○			
	SQLトレース情報、UAP統計レポート情報 (クライアント環境変数PDCLTPATHの下の pdsq1*.trc)				△
HiRDB サーバ	ユティリティ、コマンドのメッセージ (標準出力、標準エラー出力)		○		
	HiRDB運用ディレクトリ/SPOOLの下	◎	◎	◎	◎
	HiRDBの情報(起動状態など)	◎	◎	◎	◎
OS	syslog (Windowsの場合はイベントログ)	◎	◎	◎	◎
	OSの情報(バージョンやプロセス情報など)	◎	◎	◎	◎

凡例 ○:取得が必要 ◎:取得が必要(pdinfogetコマンドで取得可能) △:場合により、取得が必要

解説 トラブルシュート情報は、pdinfogetコマンドで一括して取得してください。

1. 容量の調査

該当環境のトラブルシュート情報の取得に必要な、ディスク容量を算出し、表示します。

【コマンド実行例】

```
pdinfoget -m
```

2. 情報の取得

手順(1)で表示された空き容量を確保し、トラブルシュート情報の取得を実行します。

【コマンド実行例】

```
pdinfoget -e /tmp1      :初期情報ファイル出力先ディレクトリ名  
                -d /tmp2 :詳細情報ファイル出力先ディレクトリ名
```

3. 初期情報の送付

まず、トラブルの切り分けに必要な初期情報ファイルを送付します。次のメッセージが出力された時点で、初期情報の取得は完了していますので、コマンドの終了を待たずに、送付することができます。

KFPN10403-I init information file output, file=初期情報ファイルパス名

【初期情報ファイル名】

初期情報ファイル:pdinfoget_ホスト名称_init_日時.tar(*1)

実行ログ:pdinfoget_comlog_日時(*1)

4. 詳細情報の送付

コマンド終了後、詳細情報ファイルを送付します。

【詳細情報ファイル名】

詳細情報ファイル名:pdinfoget_ホスト名称_detail_日時.tar(*1)

(*1): 日時は「MMDDhhmmss」(MM: 月、MM: 日、hh: 時、mm: 分、ss: 秒)です。

- HITACHI, BladeSymphony, Cosminexus, HAモニタ, HiRDB, OpenTP1, uCosminexus, Virtageは、株式会社 日立製作所の商標または登録商標です。
- 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。
- Linuxは、Linus Torvalds氏の日本およびその他の国における登録商標または商標です。
- Microsoft .NET は、お客様、情報、システムおよびデバイスを繋ぐソフトウェアです。
- Microsoft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。
- 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。
- Red Hatは、米国およびその他の国でRed Hat, Inc. の登録商標もしくは商標です。
- Windows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。
- Windows NT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。
- Windows Server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。
- Windows Vista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。
- その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

END



ユーザ事例に学ぶ HiRDB設計のコツとトラブルシューティング

2016/5

株式会社 日立製作所 ICT事業統括本部
サービスプラットフォーム事業本部 DB部

HITACHI
Inspire the Next