

## 遅延発生時の調査フロー

### ○EM (Oracle Enterprise Manager) の利用

#### 手順概要

(0) 発生範囲の確認

##### 発生時間帯の確認

(1) データベース・インスタンス全体の処理数の経過把握

##### Oracle 全体のリソース使用および実行数に関する統計情報

(2) インスタンス全体の(Active)セッションの待機イベントの発生状況の調査  
影響を及ぼした待機イベントの特定

(3) ハードウェア・リソース (OS 側) の利用状況の調査方法  
(使用量の見方)

(4) 待機イベントから影響を受けた SQL 文の特定

(5) 性能障害の SQL に対する対処方法の指針  
遅延原因の対象 SQL 文を見つけた後のパフォーマンス改善内容

(その他)

再現時の原因特定した SQL 文のリソース利用状況の監視とその仮説の立証  
D/B Time と I/O のページング数などの調査確認

オプティマイザが SQL 文の解析と実行計画の作成に割り当てた CPU 使用率

データベース全体の処理パフォーマンスにおける問題ポイントの見つけ方

#### 調査方法詳細

(0) 発生範囲の確認

・ 特定の人だけで発生しているのか or 全員か

発生時間帯の確認

・ 現在発生しているのか or 過去の時間 (正確な日時) に発生したのか or  
時々発生するのか

(1) データベース・インスタンス全体の処理数の経過把握  
(Oracle 全体のリソース使用および実行数に関する統計情報)

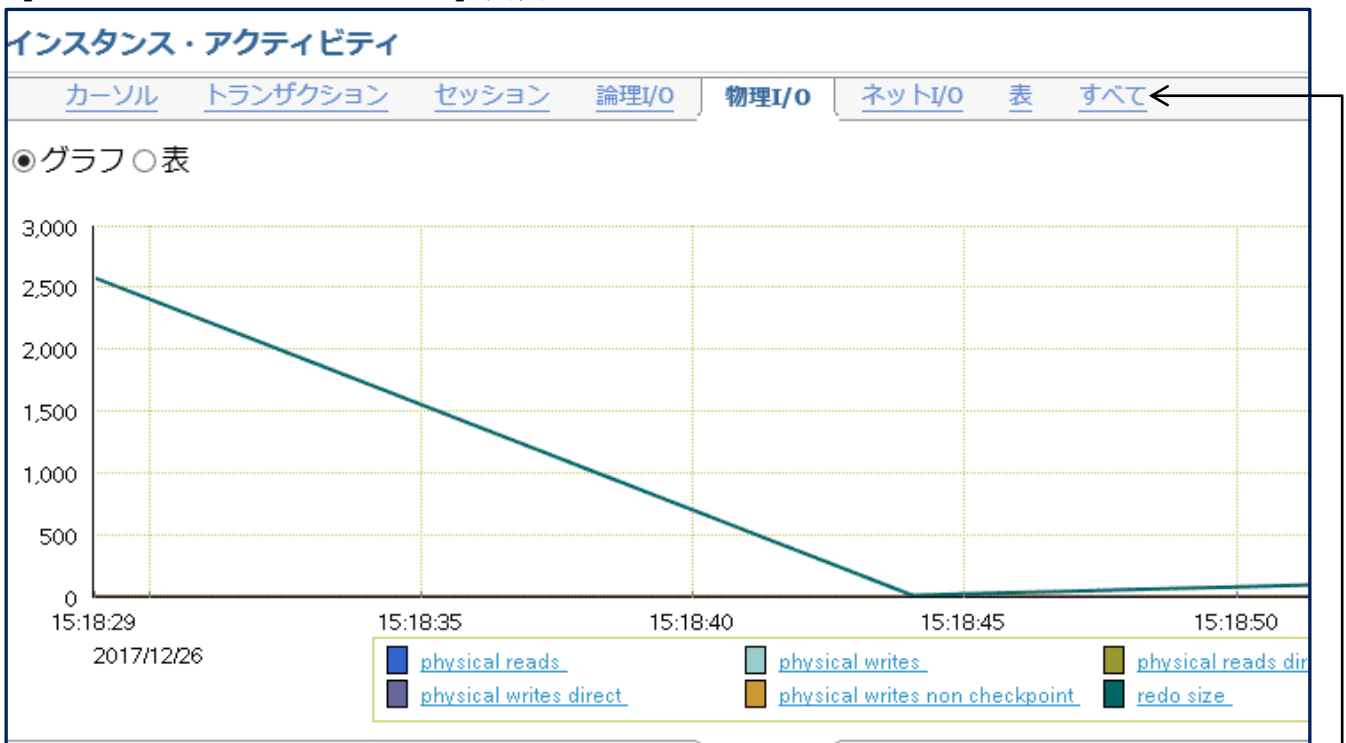
[パフォーマンス] タブ

その他の監視リンク セクション

**インスタンス・アクティビティ**

をクリック

【データベース・インスタンス】画面



カーソル処理数/秒  
トランザクション処理数/秒  
セッション発生数/秒  
論理 I/O 処理数/秒  
物理 I/O 処理数/秒  
ネット I/O 処理数/秒  
表処理数/秒

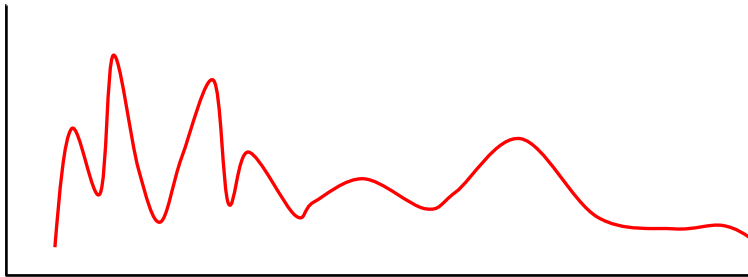
タブで、調査対象を  
切り替える

(2) インスタンス全体のセッションの待機状態の調査

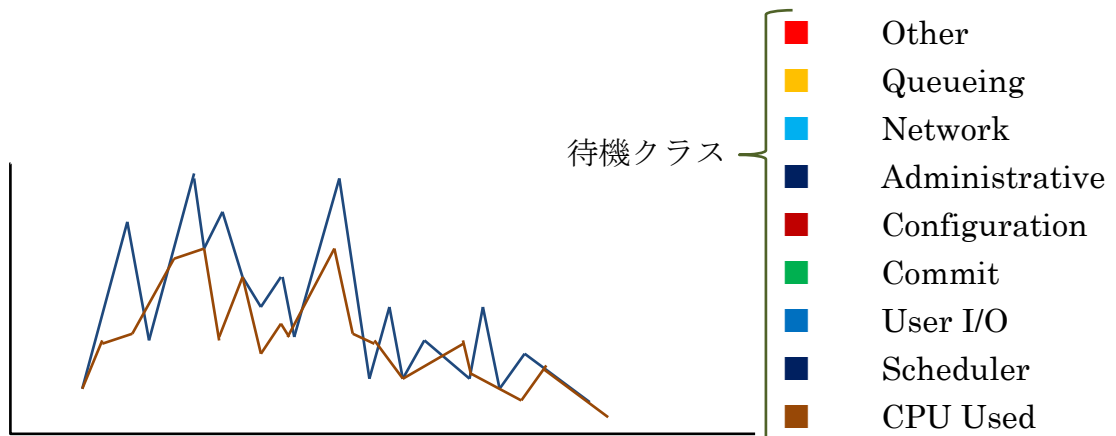
トップ・アクティビティ画面の表示操作方法  
(待機イベントの時間別発生状況グラフ)

[パフォーマンス] タブ ( ← データベース・インスタンスに関する)

ホスト (CPU 使用率)



平均アクティブ・セッション (個数)



トップ・アクティビティ

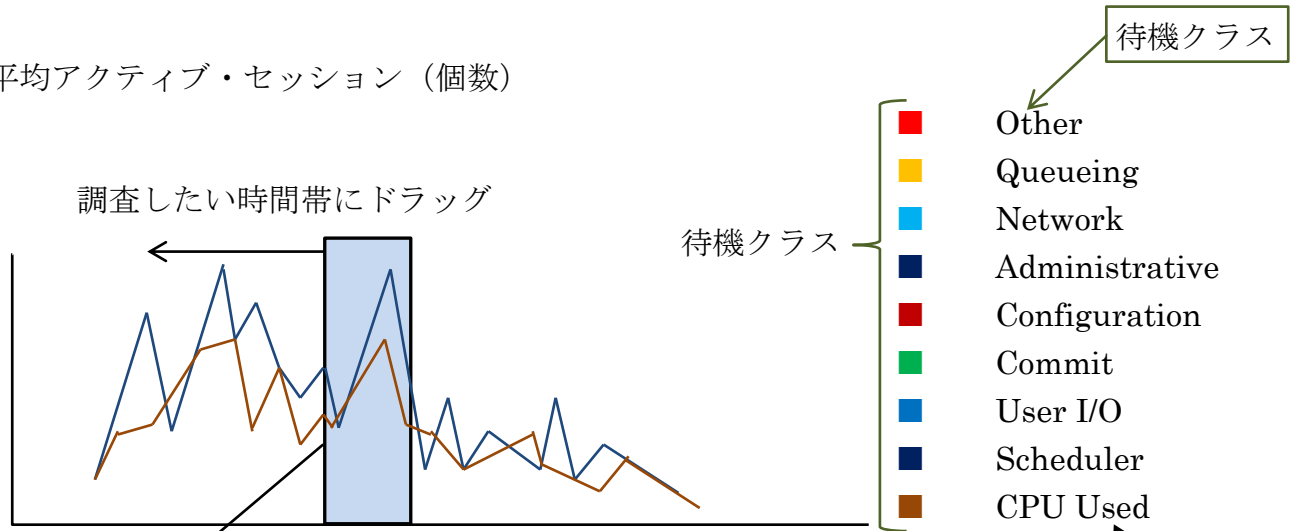
2段目の「平均アクティブ・セッション」  
のグラフの右側の凡例表示のここをクリックすると、  
待機イベントクラスの時間別発生状況のグラフが表示される

もしくは、

[パフォーマンス] タブ  
その他の監視リンク セクション

トップ・アクティビティ をクリック

平均アクティブ・セッション（個数）



選択された時間帯の中で、  
影響度が大きい SQL 文の一覧と  
その%割合を表示

凡例をクリックにより、選択した待機クラスの  
待機イベント別詳細情報が表示

System I/O : バックグラウンド・プロセスでのディスク I/O 待機  
User I/O : サーバプロセスでのディスク I/O 待機

帯グラフの「色分け」は、上の凡例の通り

**TOP SQL**

アクション

選択	トップ・アクティビティ	SQL_ID
<input type="checkbox"/>	<div style="width: 70%; background-color: #000080; background-image: linear-gradient(to right, #000080, #000000);"></div> 70%	5frhjkmlp
<input type="checkbox"/>	<div style="width: 20%; background-color: #000080; background-image: linear-gradient(to right, #000080, #000000);"></div> 20%	4lij56zas52
<input type="checkbox"/>	<div style="width: 4%; background-color: #000080; background-image: linear-gradient(to right, #000080, #000000);"></div> 4%	ukg7865gfr
<input type="checkbox"/>	<div style="width: 4%; background-color: #000080; background-image: linear-gradient(to right, #000080, #000000);"></div> 4%	kmja3fh7tf

全体に対する割合

上位#####の表示

表示

アクティビティ	ユーザー
<div style="width: 80%; background-color: #000080; background-image: linear-gradient(to right, #000080, #000000);"></div>	KOZUE
<div style="width: 20%; background-color: #000080; background-image: linear-gradient(to right, #000080, #000000);"></div>	IKURA
<div style="width: 10%; background-color: #000080; background-image: linear-gradient(to right, #000080, #000000);"></div>	MINKA
<div style="width: 5%; background-color: #000080; background-image: linear-gradient(to right, #000080, #000000);"></div>	HATUNE

これは、待機している側の SQL 文である  
(待機させた側の SQL 文ではない)

表示させられる情報

- 上位セッション
- 上位サービス
- 上位モジュール
- 上位アクション
- 上位クライアント
- 上位ファイル
- 上位オブジェクト
- 上位 PL/SQL

選択できるアクション

- SQL チューニング・アドバイザのスケジュール
- SQL チューニング・セットの作成

「性能遅延の原因追及のためのフローチャート」については、

## 影響を及ぼした待機クラス → イベントの特定

＜遅延が発生している時間帯のグラフは、通常より変化があるのか＞

### 【グラフに変化有り】の場合

- ・特定の待機イベントが多いのか
- ・指定の待機イベントだけのグラフを抽出す
- ・時間帯を指定して、待機イベント値が大きい SQL の統計情報を調査する

### 【グラフに変化無し】の場合

ー このグラフは、セッション数に対してのグラフである ー

1つの SQL 文だけが特出して負荷が高いバッチ型のジョブで、他からのセッション接続が増えなかった時間帯で発生していた場合には、グラフ表示には変化は表れない

別の方法で調査が求められる

また、レスポンスが悪い時間帯にデータベース以外の原因が影響していることも考えられる

その調査も、別途に行う

(例えば、ネットワーク障害、クライアント側障害)

(注意)

性能分析を行う場合、「性能遅延が確実に発生している時間帯」と「性能遅延が確実に発生していない時間帯」を教えてください、調査(比較)する

性能遅延を素早く検知するために、閾値監視機構を設定してシステム側に自動検出させる

(3) ハードウェア・リソース (OS 側) の利用状況の調査方法 (使用量の見方)  
 (発生していた待機イベントに関連させる)

ホスト：サーバー名

[パフォーマンス] タブ ( ← サーバー自体のパフォーマンス)

表示

パフォーマンス・サマリー

CPU 詳細

メモリの詳細

ディスクの詳細

プログラム・ホストの使用率

トップ・アクティビティが表示された場合には、  
 [ホームタブ]  
 →ホストCPUグラフの中の「その他」の凡例をクリック

CPU 詳細

- CPU 使用率
- CPU 負荷

メモリの詳細

- メモリ使用率 (%)
- スワップ使用率 (%)

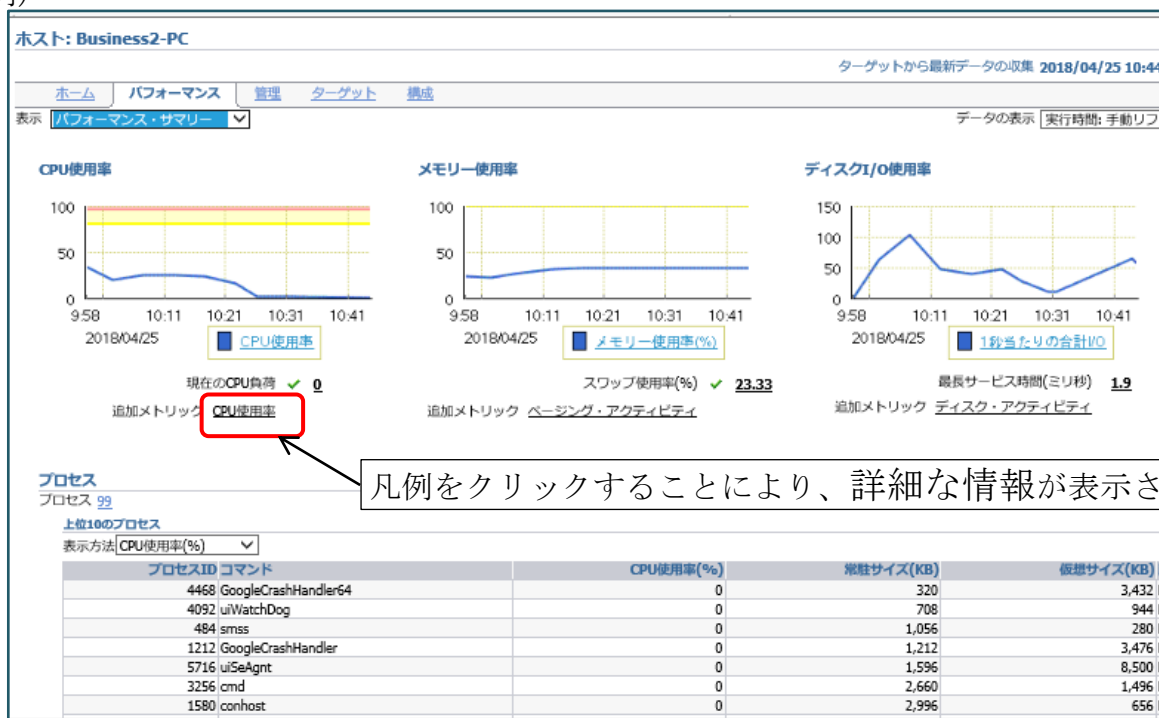
ディスクの詳細

- ディスク I/O 使用率
- 最長 I/O サービス時間(ミリ秒)

プログラム・リソースの使用率

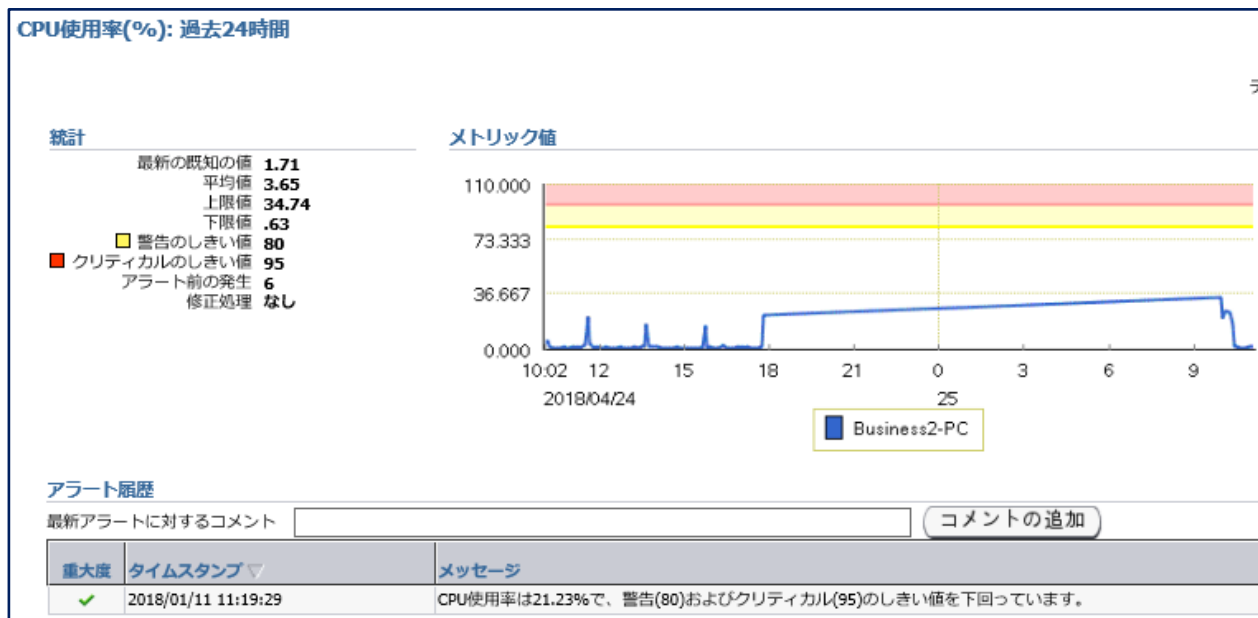
このホストでは、「プログラム・リソース使用率」ページがサポートされていません

表示例)



凡例をクリックすることにより、詳細な情報が表示される

詳細データ表示例)



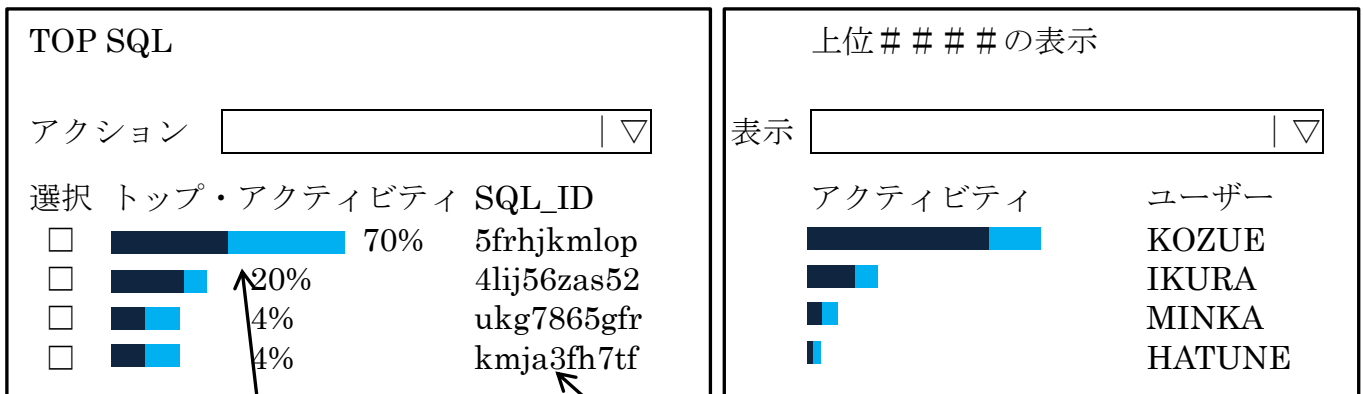
#### (4) 待機イベントから影響を受けた SQL 文の特定

【(1) のトップ・アクティビティのグラフ表示画面の下部より】

(1) で、レスポンス遅延があった時間帯を決定したら、その時間帯で動作していた SQL 文が、以下の欄の Top SQL のように表示されてくる

ここで、SQL\_ID の文字をクリックすると、各 SQL 文に対する実行統計と実行計画の詳細が表示される

※ トップ・アクティビティ画面でグラフの凡例をクリックし、特定の待機クラスだけの表示にした方が、待機クラスと SQL の関係が明確になり待機イベントの影響割合が解り易くなる

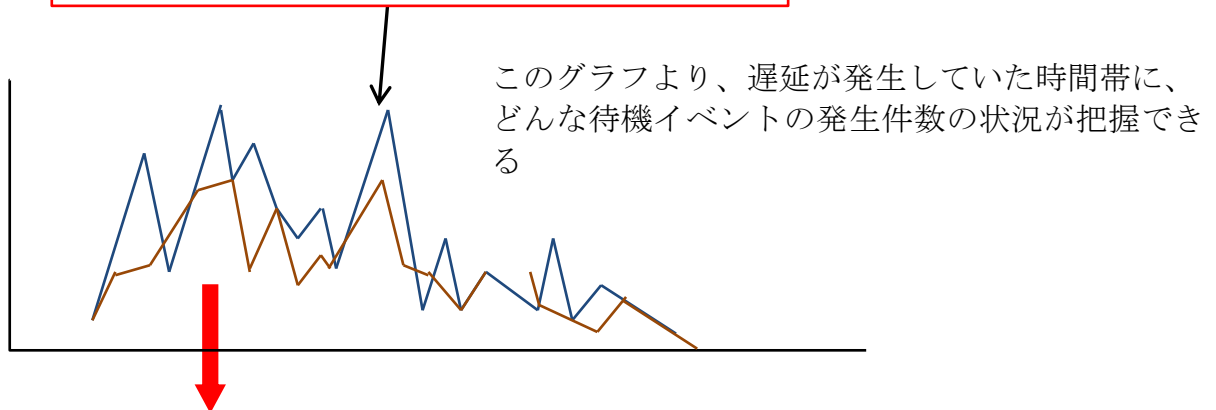


SQL ID をクリックにより、SQL 文の実行統計と実行計画が表示される

表示された実行統計の詳細を見て、レスポンス遅延の**原因**となった SQL 文と**影響を及ぼされた SQL 文**を切り分ける  
また、ボトルネックとなった根本原因（要因）を推測する

重要)

グラフ部分のクリックにより、帯グラフが示している**待機イベント**の時間別の（システム全体を対象とした）状況が表示される



待機イベントの種類から、原因特定と対処方法を行う



## (5) 性能障害の SQL に対する対処方法の指針

遅延原因の対象 SQL 文を見つけた後のパフォーマンス改善内容

(その他)

### 再現時の原因特定した SQL 文のリソース利用状況の監視とその仮説の立証

監視方法については、

[パフォーマンス] タブ → SQL 監視

#### 【SQL リアルタイム監視による実行状況の調査】

- D/B Time と I/O のページング数などが表示できます
- 長時間実行していて終わらない SQL があった場合は、現在どの処理を行っているかということも確認できます
- SQL\_ID、SQL Text で疑わしいと仮説した SQL 文の処理が開始されたら、その SQL 文の実行状況と、同時に処理される他のセッションの SQL 文の実行状況 (IO Requests) を確認して、レスポンス障害が再現しているか確認します

※ この画面表示のためには、Oracle Tuning Pack が別途必要

<http://www.oracle.com/jp/store/oemlicense-function-tool-170249-ja.pdf>

Status	Duration	SQL ID	User	Parallel	Database Time	IO Requests	Start	Ended	SQL Text
✓	46.0s	fskyn4u39marv	SYS		44.6s	76K	Sat Jan 16, 2010	Sat Jan 16, 2010	DECLARE job
✓	8.0s	4u5zq7r9y690a	SYS		9.1s	27K	Sat Jan 16, 2010	Sat Jan 16, 2010	WITH NONTI
✓	33.0s	anxyyx2tjz3g	SYS		32.4s	48K	Sat Jan 16, 2010	Sat Jan 16, 2010	WITH TIMEG
✓	20.0s	0a59tkkpx31vx	SYSTEM		19.5s	291	Fri Jan 15, 2010 2	Fri Jan 15, 2010 2	select output
✓	1.6m	gyfr7cpmbcx9r	SYS		1.6m	14K	Tue Jan 12, 2010	Tue Jan 12, 2010	DECLARE job
✓	14.0s	4ypzyny8pscjk	SYSTEM		12.6s	211	Mon Jan 11, 2010	Mon Jan 11, 2010	SELECT DBW
✓	8.0s	atkp0nywk0m99	SYSTEM		6.8s	649	Mon Jan 11, 2010	Mon Jan 11, 2010	DECLARE job
✓	8.0s	fskyn4u39marv	SYS		8.2s	15K	Sat Jan 9, 2010 4	Sat Jan 9, 2010 4	DECLARE job
✓	4.0s	anxyyx2tjz3g	SYS		4.1s	9027	Sat Jan 9, 2010 4	Sat Jan 9, 2010 4	WITH TIMEG
✓	13.0s	4ypzyny8pscjk	SYSTEM		13.3s	288	Thu Jan 7, 2010 €	Thu Jan 7, 2010 €	SELECT DBW
✓	59.0s	a0xx1k23fvg3z	SYSTEM		59.1s	742	Thu Jan 7, 2010 €	Thu Jan 7, 2010 €	DECLARE job
✓	13.0s	91pktudh76mzx	SYSTEM		13.8s	857	Thu Jan 7, 2010 €	Thu Jan 7, 2010 €	DECLARE job

SQL リアルタイム監視画面では、該当時間帯の該当 SQL 実行状況を表示して、遅くなかった時の SQL の実行状況と比較できる

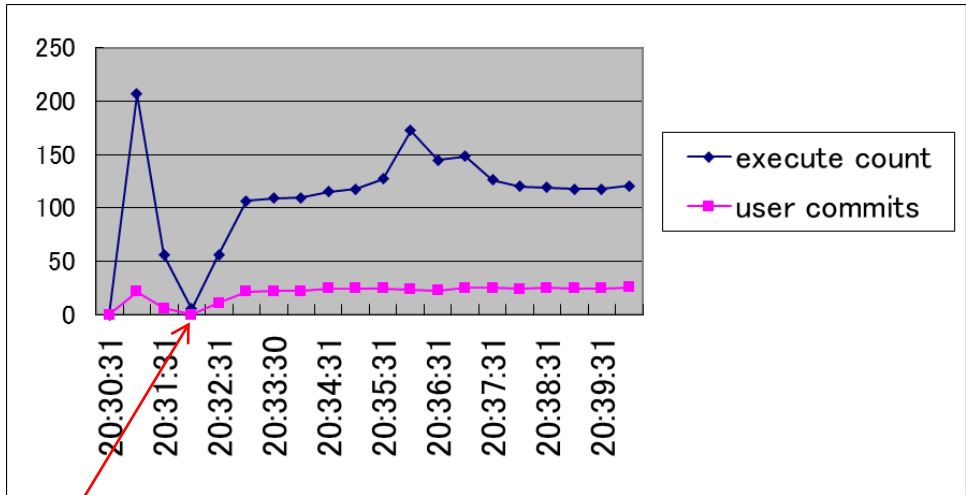
(比較項目：実行計画、処理行数、メモリ使用量)

옵ティマイザ가 SQL 文의解析と実行計画의作成に割り当てた CPU 使用率

데이터베이스全体の処理パフォーマンスにおける問題ポイントのを見つけ方

(参考) v\$session ビューでの実行中 SQL の状態調査と影響を与えた SQL の特定

v\$sysstate グラフサンプル

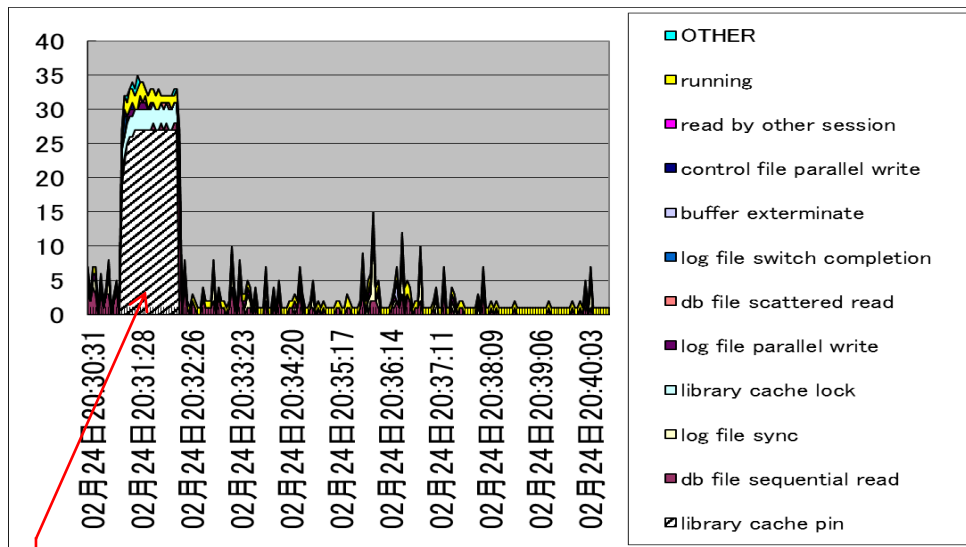


execute count . . . . . SQL 文発生実行数  
 user commit . . . . . 内部データベース処理数

user commit 数の減少は、Oracle のデータ処理量の減少を意味する  
 execute count 数が多い状態を保ち、user commit 数が減少した場合には、  
 待機イベントによる「待ち」が発生している可能性がある  
 もしくは、軽い処理の SQL ばかりの発生も考えられる

このグラフから、遅延状況が発生していた時間帯を把握する

v\$session グラフサンプル



このグラフから、遅延状況が発生していた待機イベントの種類を把握する

(1) 処理時間が長い SQL 文の調査 v\$session

- ・ v\$session ビューからは、直接求めることが出来ない  
v\$sql\_monitor ビューを使用して、求める

---

処理時間 (CPU 時間+待機時間) が 5 秒以上の SQL 文の洗い出し

---

※ CPU 時間の数値付き

```
Select SE.sid , SE.sql_id , CPU_TIME/1000000 , ELAPSED_TIME/1000000 ,
       round( ( SYSDATE - SE.SQL_EXEC_START ) * 86300 , 0 )
       ELAPSED_SED
From V$SQL_MONITOR SM , V$SESSION SE
Where SM.sql_exec_id(+) = SE.sql_exec_id
      and SM.sid(+)      = SE.sid
      and ( round( (SYSDATE - SE.SQL_EXEC_START)*86300 , 0) > 5
           or ( SM.sid is NOT NULL ) ) ;
```

- ・ round( (SYSDATE - SQL\_EXEC\_START)\*86300 , 0) ELAPSED\_SED  
(単位：秒)

待機状態が継続されている SQL 文の調査

v\$session ビュー

- ・ last\_call\_et : status 列が、「ACTIVE」 or 「INACTIVE」になってからの経過時間 (単位：秒)

※ last\_call\_et、WAIT\_TIME\_MICRO (待機時間 (マイクロ秒))、SECONDS\_IN\_WAIT (非推奨) は、待機時間継続している状態の SQL 文を見つける

SQL 文の処理が終了していなくても、一旦待機状態が解消されると次の待機に入った時には 0 からの計測となる

- ・ 処理が続いている SQL 文は、STATUS が「ACTIVE」の状態です。STATUS が「INACTIVE」の状態のセッションは、アイドル状態なので遅延には無関係です
- ・ STATE が「WAIT」の状態の処理は、待機イベントが発生していて待ち状態の SQL であり、「WAIT」以外の状態の処理は、すべてデータベースが CPU 処理を行っている状態の SQL である

```
select username , SECONDS_IN_WAIT , STATE
from v$session
where status ='ACTIVE' and username = 'KOZUE';
```

(2) その SQL が利用したリソース量（実行計画と実行統計）の把握

- CPU 時間が長時間だった SQL 文に対しては、**v\$sql\_monitor** ビューより取得することが可能である

※ ただし、CPU 時間が 5 秒以上の SQL 文しか記録されない  
保存されている期間が短いことに注意する

上記以外の SQL 文に対する実行統計情報を入手する場合は、以下のとおりである

- SQL 文の処理内容を把握します (v\$sql、v\$sqlarea より)
  - ※ 複数回実行された場合には、統計情報は実行回数の合計値となるので注意  
回数で割り、1 回分の統計値で状況を判断すること
- 実行回数、CPU 時間、実行時間、ディスク・アクセス回数、バッファ取得数、レコード処理件数などの実行統計情報を把握します (v\$sqlarea より)
- 実行計画は、explain plan for を利用して調査する

(3) 原因元と影響を受けた側の SQL の把握（待機時間が長いセッション調査）

v\$session

- STATE が「WAIT」の状態、WAIT\_TIME\_MICRO（待機時間（マイクロ秒）） or SECONDS\_IN\_WAIT が大きいレコードが、何らかの影響を受けて処理の中断により長時間の実行となってしまう SQL 処理です
- 原因元となる SQL 文と待機中断された二次側の SQL 文の分別が必要です

(4) 待ち行列（待機イベント）の確認

- v\$session に出力される enqueue などの排他制御管理のロック待ちの原因元と影響側の SQL 文の関係については、