

MaxGauge 診断/分析プロセス

システム性能低下認識

システムレベル分析:トレンド、アラート等

診断/分析対象の時間帯を特定

概要分析:アクティブセッション/滞留/CPU

詳細領域分析:I/O、メモリー、ロック、上位SQL...

セッション診断/分析

SQL診断/分析

【必要時】正常時との比較分析、ログ外部情報確認



トップダウン
アプローチ

原因不明の(OS)データベース再起動問題の解決（原因トリガーの追跡と対応）

【事象】

RAC環境において、突然Oracleクラスタウェアが、データベース停止状況であると判断し、データベースの再起動をかけてしまっていた。

【解決策】

MaxGaugeのログより、再起動時点でのデータベース内処理状況より、同一ブロックへの大量な「DELETE」、「INSERT」処理がCPUを100%を使い切って、滞留が急増していることを確認。そのため、「システムが過負荷状態 → Oracleクラスタウェアの処理(ハートビート送信)が完了出来ない → データベース停止(ハング)と認識 → OS再起動」発生。対象SQLのチューニングにより、データベース負荷を軽減。データベース再起動の事象の発生を抑えた。

【効果】

MaxGaugeのログ調査以前、システム開発担当者、およびオラクルサポート担当にて原因調査を約2週間行っていたが、原因がつかめなかった。

MaxGaugeのログの参照により、約2時間で障害時の状況の把握と対象SQLのピックアップを行い、顧客へレポート。

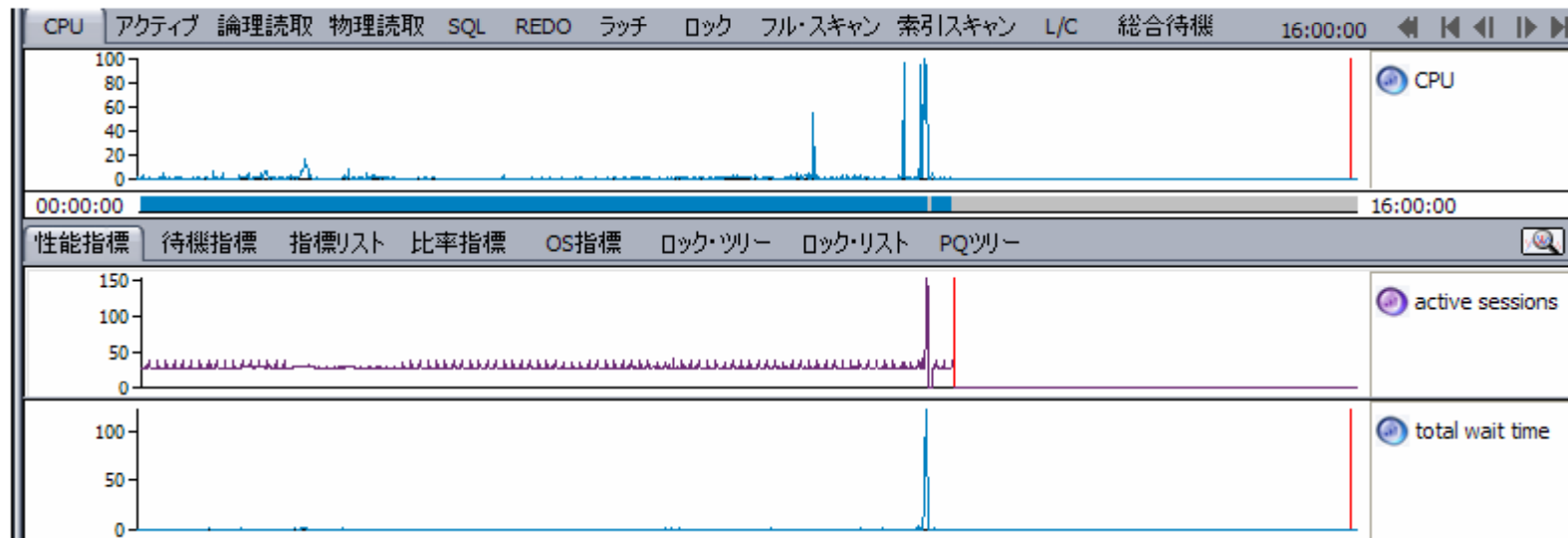
(クラスタウェアが、データベースを再起動させてしまう事象については、製品仕様の確認のため、オラクルサポート担当とのやり取りは継続)

診断/分析プロセス

● 現象

Oracle clsmom failed with fatal status 13.
Oracle CRS failure. Rebooting for cluster integrity.

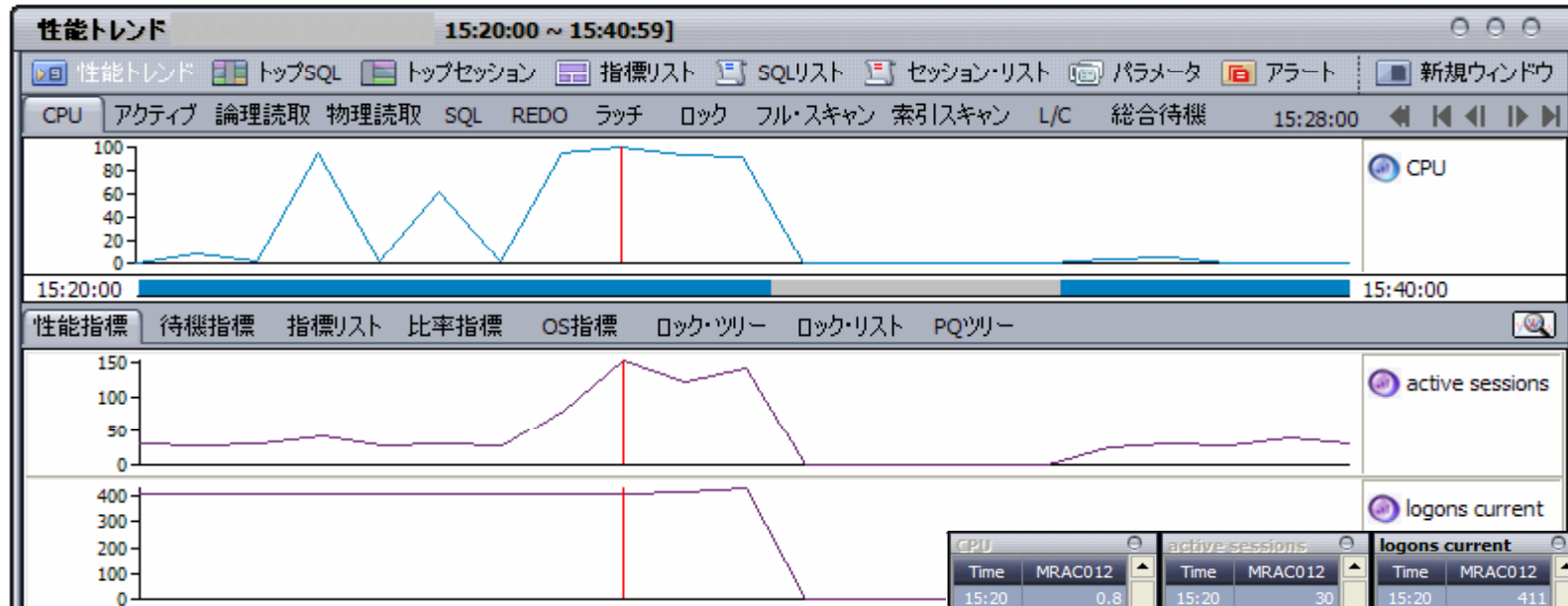
● システムレベル診断/分析



- 15:30前後で、CPU使用率が100%ほど使用されている。
- 同時間帯で、アクティブセッションが30前後から150まで急増した。
- 同時間帯で、数秒程度の滞留が100秒以上まで急増した。

診断/分析プロセス

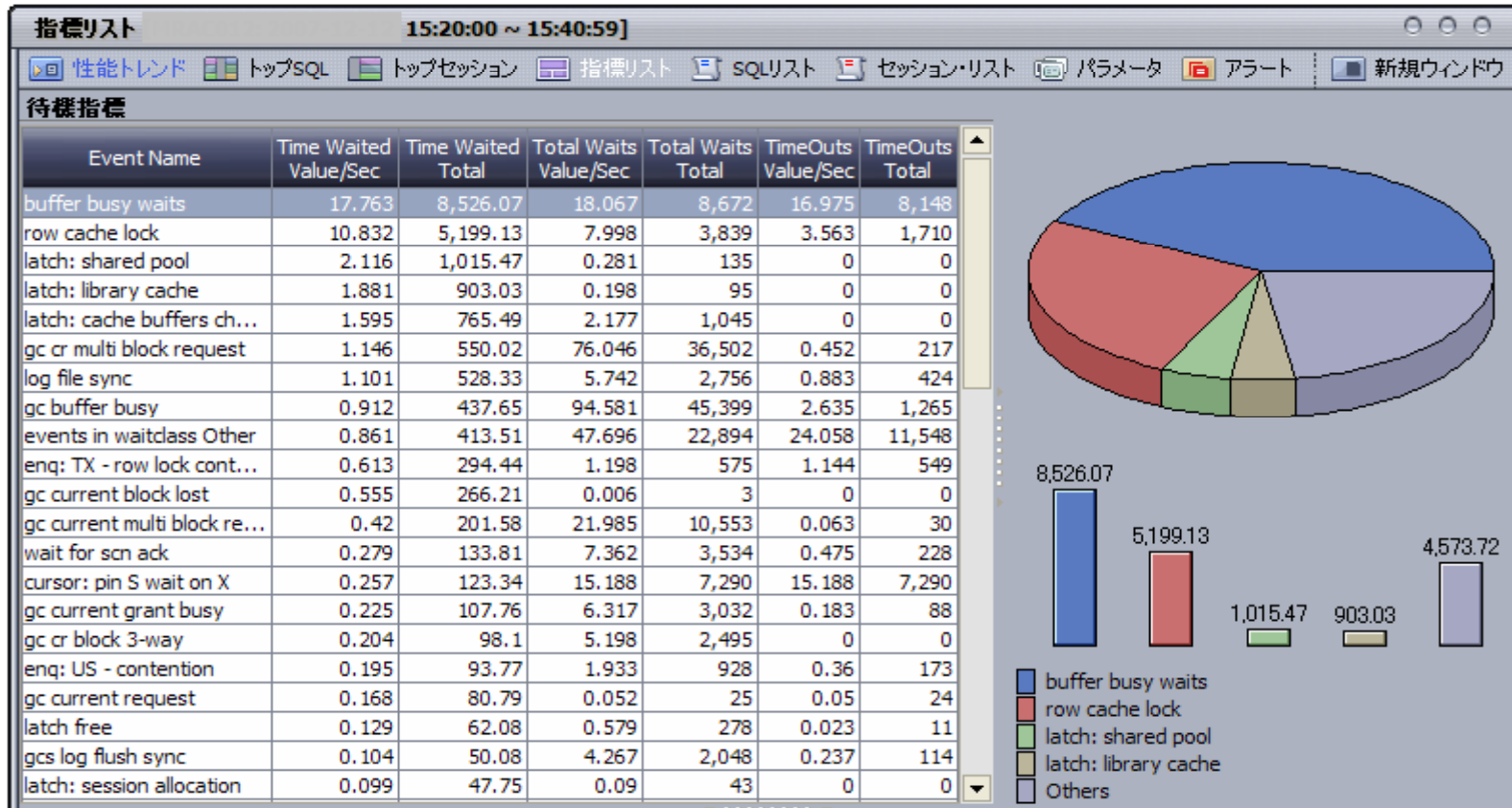
● 特定(異常現象発生)時間帯の詳細分析



- 「15:20～15:30」間で、接続数が「411→432」増加
- 同時間帯で、アクティブセッションが「30→153」増加、15:28ピーク
- CPU使用率は、「15:28」で100%になり、続いて過負荷状態

CPU		active sessions		logons current	
Time	MRAC012	Time	MRAC012	Time	MRAC012
15:20	0.8	15:20	30	15:20	411
15:21	7.8	15:21	29	15:21	410
15:22	1.5	15:22	30	15:22	410
15:23	94.8	15:23	42	15:23	411
15:24	1.5	15:24	29	15:24	410
15:25	61.5	15:25	30	15:25	411
15:26	1.7	15:26	29	15:26	410
15:27	95	15:27	82	15:27	412
15:28	100	15:28	153	15:28	412
15:29	92.8	15:29	122	15:29	420
15:30	91.1	15:30	143	15:30	432
15:31	0	15:31	0	15:31	0
15:32	0	15:32	0	15:32	0
15:33	0	15:33	0	15:33	0
15:34	0	15:34	0	15:34	0
15:35	0	15:35	0	15:35	0
15:36	3.2	15:36	25	15:36	25
15:37	4.7	15:37	30	15:37	36

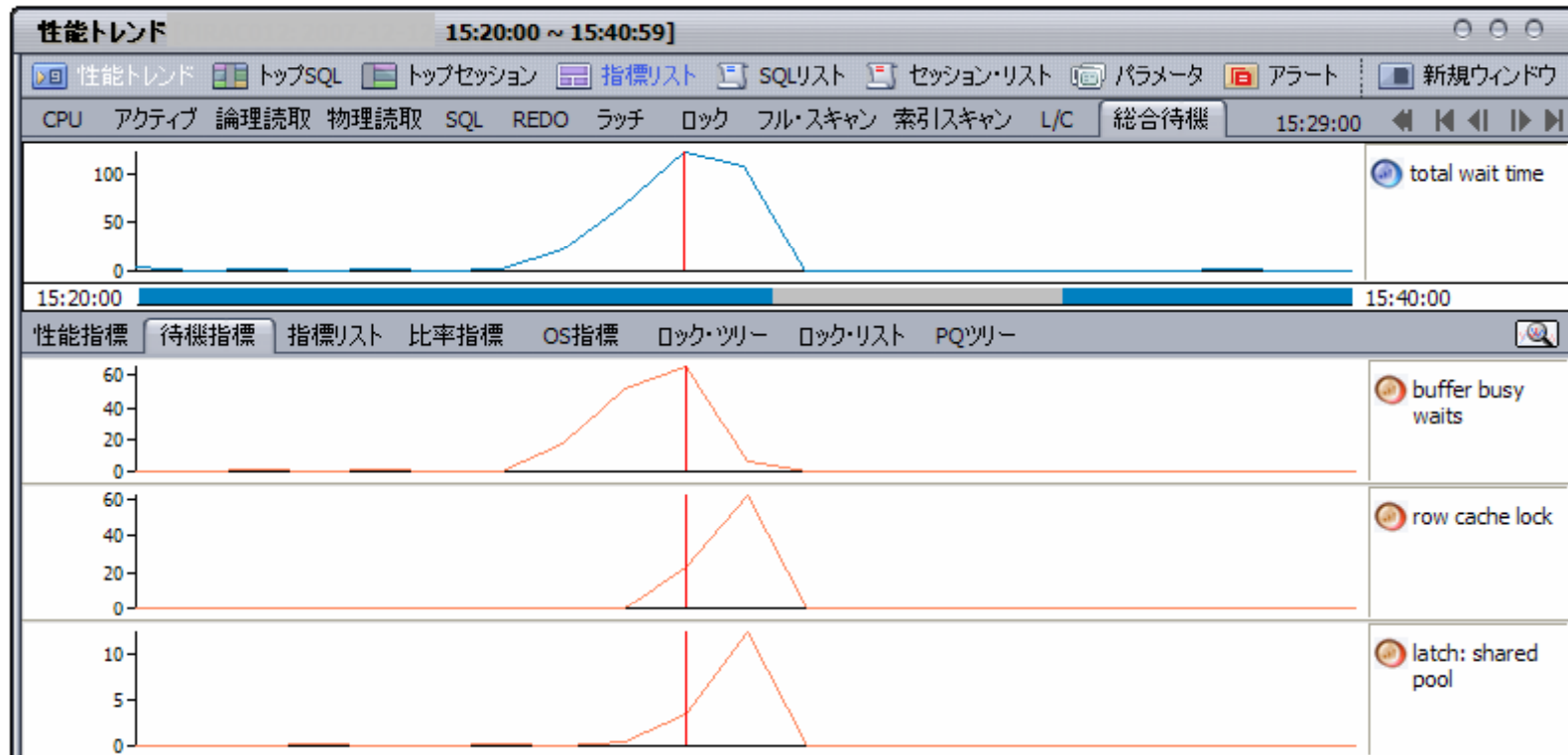
● 特定(異常現象発生)時間帯の詳細分析



- ・「15:20～15:40」間で、上位滞留は「buffer busy waits、row cache lock、latch:shard pool..」順で比較的に多数の滞留が発生
- ・「buffer busy waits」は同じブロックに対する競合現象で、全体の42%以上の滞留を占めている

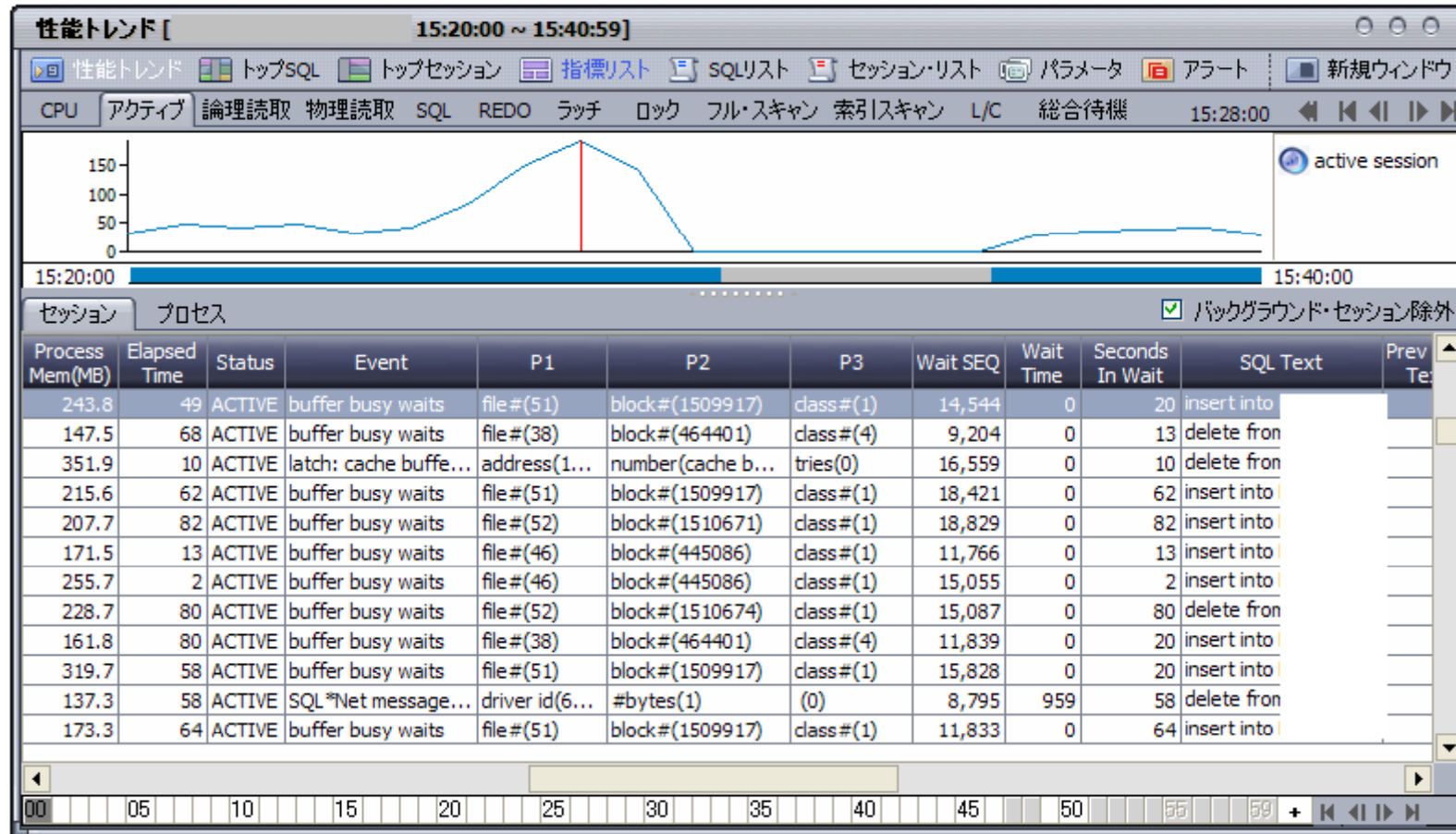
診断/分析プロセス

● 特定(異常現象発生)時間帯の詳細分析



- ・最初に「buffer busy waits」滞留が見れ、他の滞留はその後の性能低下の悪循環で現れたように見える。

● 特定(異常現象発生)時間帯のセッション分析



- ・ アクティブセッションの中「108」セッションが「HISTTBL」表に対する「DELETE、INSERT」作業を行っている。

● 特定(異常現象発生)時間帯のセッション分析

SQL統計	First Time	Last Time	SQL Text
概要	15:22:33	15:28:39	delete from
SQLリスト	15:28:40	15:29:58	insert into l

- ・「HISTTBL」表に対する「DELETE、INSERT」を実施しているセッションの履歴(詳細)を確認すると、「DELETE=6:36」後、「INSERT=1:17」を発行している。

● 特定SQL分析

SQLリスト 00:00:00 ~ 23:59:59

性能トレンド | トップSQL | トップセッション | 指標リスト | SQLリスト | セッション・リスト | パラメータ | アラート | 新規ウインドウ

基本条件

From: 00:00 To: 23:59 SQL ID: SQL Text: delete from HISTTBL

検索

From Time ▲	SQL Text	Elapsed (AVG)	LReads	PReads	Redo Size	Block Scan	Row Fetch	Sort Rows	Elapsed (SUM)	Waiting (SUM)	CPU (SUM)	Executions
13:50:00	delete from	0.05	0	0	0	0	0	0	0.25	0	0.25	5
13:50:00	delete from	0.05	0	0	0	0	0	0	0.2	0	0.2	4
14:00:00	delete from	0.05	0	0	0	0	0	0	0.4	0.05	0.35	8
14:00:00	delete from	0.05	0	0	0	0	0	0	0.35	0	0.35	7
14:10:00	delete from	0.05	0	0	0	0	0	0	0.25	0.1	0.15	5
14:10:00	delete from	0.05	0	0	0	0	0	0	0.15	0	0.15	3
14:20:00	delete from	0.05	0	0	0	0	0	0	0.15	0	0.15	3
14:20:00	delete from	0.05	0	0	0	0	0	0	0.15	0.1	0.05	3
14:30:00	delete from	0.05	0	0	0	0	0	0	0.4	0.05	0.35	8
14:30:00	delete from	0.05	0	0	0	0	0	0	0.1	0	0.1	2
14:40:00	delete from	0.058	0	0	0	0	0	0	0.75	0.3	0.45	13
14:50:00	delete from	0.317	0	0	0	0	0	0	0.95	0	0.95	3
14:50:00	delete from	0.05	0	0	0	0	0	0	0.35	0.05	0.3	7
15:00:00	delete from	0.069	0	0	0	0	0	0	0.9	0.6	0.3	13
15:00:00	delete from	0.05	0	0	0	0	0	0	0.15	0	0.15	3
15:10:00	delete from	0.32	0	0	0	0	0	0	4.8	3.2	1.6	15
15:10:00	delete from	0.05	0	0	0	0	0	0	0.2	0.05	0.15	4
15:20:00	delete from	58.45	1	0	332	0	0	0	350.7	350.65	0.05	6
15:20:00	delete from	42.142	38	0	2,020	0	0	0	3,371.4	1,984.8	1,386.6	80

- ・「DELETE FROM HISTTBL...」は終日発行されているが、「15:20:00 ~ 15:30:00」で集中して発行されている。

診断/分析プロセス

● 特定SQL分析

From Time ▲	SQL Text	Elapsed (AVG)	LReads	PReads	Redo Size	Block Scan	Row Fetch	Sort Rows	Elapsed (SUM)	Waiting (SUM)	CPU (SUM)	Executions
12:20:00	insert into	0.05	0	0	0	0	0	0	0.4	0	0.4	8
12:30:00	insert into	0.05	0	0	0	0	0	0	0.4	0	0.4	8
12:40:00	insert into	0.05	0	0	0	0	0	0	0.35	0	0.35	7
12:50:00	insert into	0.054	0	0	0	0	0	0	0.65	0.1	0.55	12
13:00:00	insert into	0.05	0	0	0	0	0	0	0.15	0	0.15	3
13:10:00	insert into	0.05	0	0	0	0	0	0	0.1	0	0.1	2
13:20:00	insert into	0.05	0	0	0	0	0	0	0.25	0	0.25	5
13:30:00	insert into	0.05	0	0	0	0	0	0	0.45	0.05	0.4	9
13:40:00	insert into	0.05	0	0	0	0	0	0	0.25	0	0.25	5
13:50:00	insert into	0.05	0	0	0	0	0	0	0.2	0	0.2	4
14:00:00	insert into	0.169	0	0	0	0	0	0	1.35	0	1.35	8
14:10:00	insert into	0.05	0	0	0	0	0	0	0.25	0	0.25	5
14:20:00	insert into	0.05	0	0	0	0	0	0	0.2	0	0.2	4
14:30:00	insert into	0.05	0	0	0	0	0	0	0.2	0	0.2	4
14:40:00	insert into	0.05	0	0	0	0	0	0	0.3	0.1	0.2	6
14:50:00	insert into	0.05	0	0	0	0	0	0	0.3	0	0.3	6
15:00:00	insert into	0.313	0	0	0	0	0	0	2.5	0.9	1.6	8
15:10:00	insert into	0.2	0	0	0	0	0	0	0.8	0.7	0.1	4
15:20:00	insert into	89.316	2,196	0	24,208	2,08...	20	0	12,861...	12,739	122.45	144

- ・「INSERT INTO HISTTBL...」も終日発行されているが、「15:20:00 ~ 15:30:00」で集中して発行されている。

● 特定SQL分析

SQLリスト 00:00:00 ~ 23:59:59

性能トレンド トップSQL トップセッション 指標リスト SQLリスト セッション・リスト パラメータ アラート 新規ウインドウ

Schema : Elapsed Time : >= 0 Executions : >= 0
Program : CPU : >= 90 Logical Reads : >= 0
Module : Waiting : >= 0 Physical Reads : >= 0

検索

From Time ▲	SQL Text	Elapsed (AVG)	LReads	PReads	Redo Size	Block Scan	Row Fetch	Sort Rows	Elapsed (SUM)	Waiting (SUM)	CPU (SUM)	Executions	CPU (AVG)
02:30:00	Insert Into	220.35	23,326,463	30	1,165,...	2,01...	348,...	0	220.35	0.1	220.25	1	220.2
03:10:00	Update	4.495	8,308,667	4,463	40,055...	-1,8...	219,...	0	170.8	29.5	141.3	38	3.7
03:10:00	Update	0.218	2,610,832	12	255,17...	671,...	2,062	0	212.5	118.3	94.2	976	0.09
03:20:00	Update	7.831	6,547,582	542	31,596...	1,21...	172,...	0	101.8	4.9	96.9	13	7.4
15:20:00	insert into	89.316	2,196	0	24,208	2,08...	20	0	12,861...	12,739	122.45	144	0.8
15:20:00	delete from	42.142	38	0	2,020	0	0	0	3,371.4	1,984.8	1,386.6	80	17.3
15:20:00	Update	1.372	1,599,468	3,120	6,696,...	2,65...	11,797	0	1,469.7	1,286.95	182.75	1,071	0.1

- CPU使用率が高いSQLの検索結果、「15:20:00 ~ 15:30:00」時間帯で集中している。

診断/分析プロセス

● 診断/分析サマリー

- ✓ 「15:26」頃からデータベースへ接続が増え、既存のセッションと合わせて、「HISTTBL」表に対する集中的なデータ削除・追加作業を実施した。
- ✓ 作業量の増加と同じデータ(ブロック)に対する大量の同時変更作業で滞留が増えCPUが限界に達した。
- ✓ このようなシステムの過負荷によって、Oracleクラスタウェアの定期的な死活監視活動のハートビート(1回/1秒)送信が、決まった時間内に正常の応答を得られなくなった。
- ✓ ノード障害(データベース停止、ハングなど)と判断し、OracleクラスタウェアがOSの再起動を実施した。

● 改善(チューニング)案

- ✓ 「HISTTBL」表に対するデータ削除・追加作業が集中しないようにロードバランスを行う。
 - 実施時間帯の分散、他ノードへの接続分散
- ✓ 同じブロックに対する競合が発生しないように「HISTTBL」表のデータを分散する。
 - パーティション化、1ブロックサイズの調整、1ブロック当りの格納データ件数の調整
- ✓ CPU使用率、アクティブセッション数、滞留、DB接続数に対する予兆監視を行う

原因不明の接続エラーの解決（原因トリガーの追跡と対応）

【事象】

1日/1ヶ月程度の頻度で接続エラーが続出する。

【解決策】

MaxGaugeのログより、当日のエラー発生時刻でDB接続数が最大設定値まで達していたことがエラーの原因になったことが確認された。続きで、接続の急増現象は激しいI/O滞留に起因することが確認され、I/O滞留を削減するため対策を適用することで接続エラーはなくなった。

【効果】

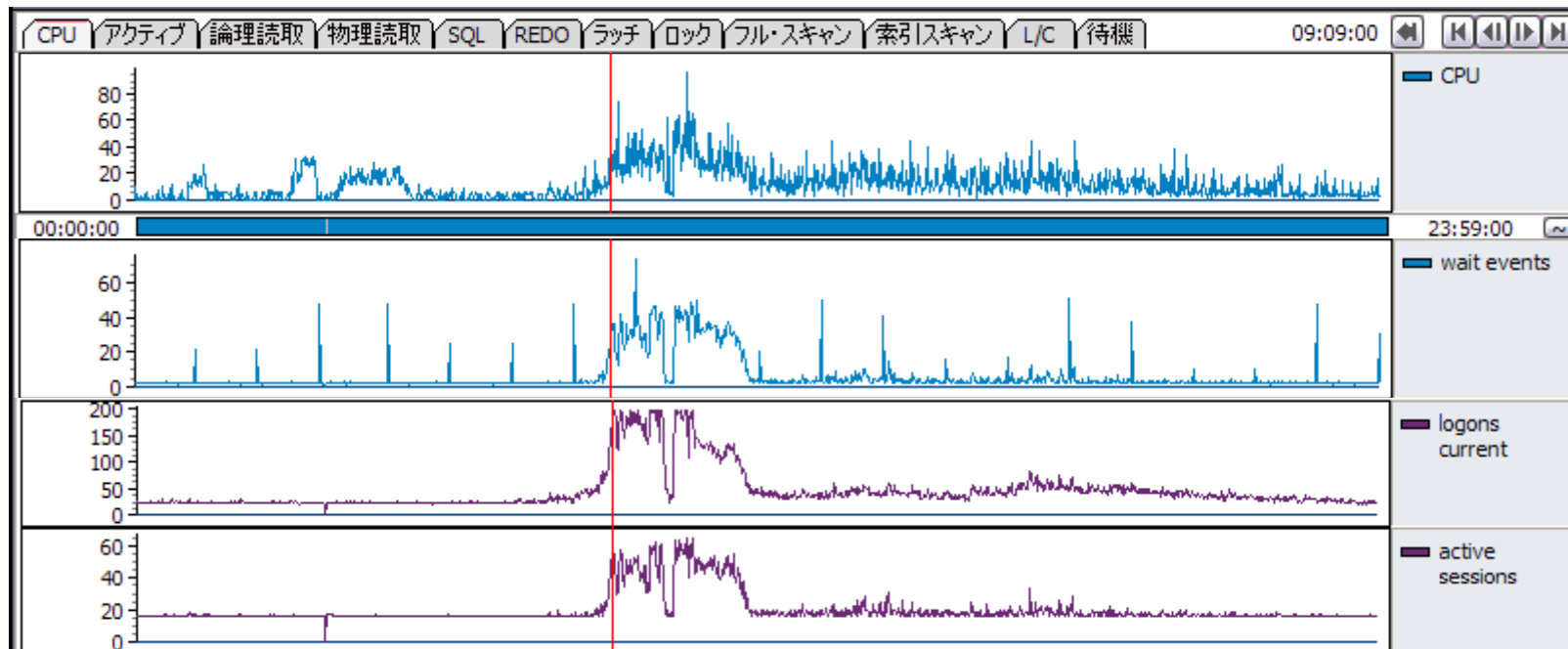
MaxGaugeのログの調査以前、システム開発担当者およびDBエキスパートにて原因調査を行っていたが、エラー発生時の稼働状況が分からなく、**約1ヶ月間**原因が掴めなかったため、一部のサービスを止めることで対応してきた。MaxGaugeのログの参照により、接続数の傾向から**約1時間**で直接的な原因が分かって、続きの分析でDB負荷のボトルネックとなっているAP(SQL)を特定して、関連対策を講じれるようになった。

診断/分析プロセス

● 現象

「1日/1ヶ月」の頻度でユーザーI/Fのブラウザで接続エラーが続出する。

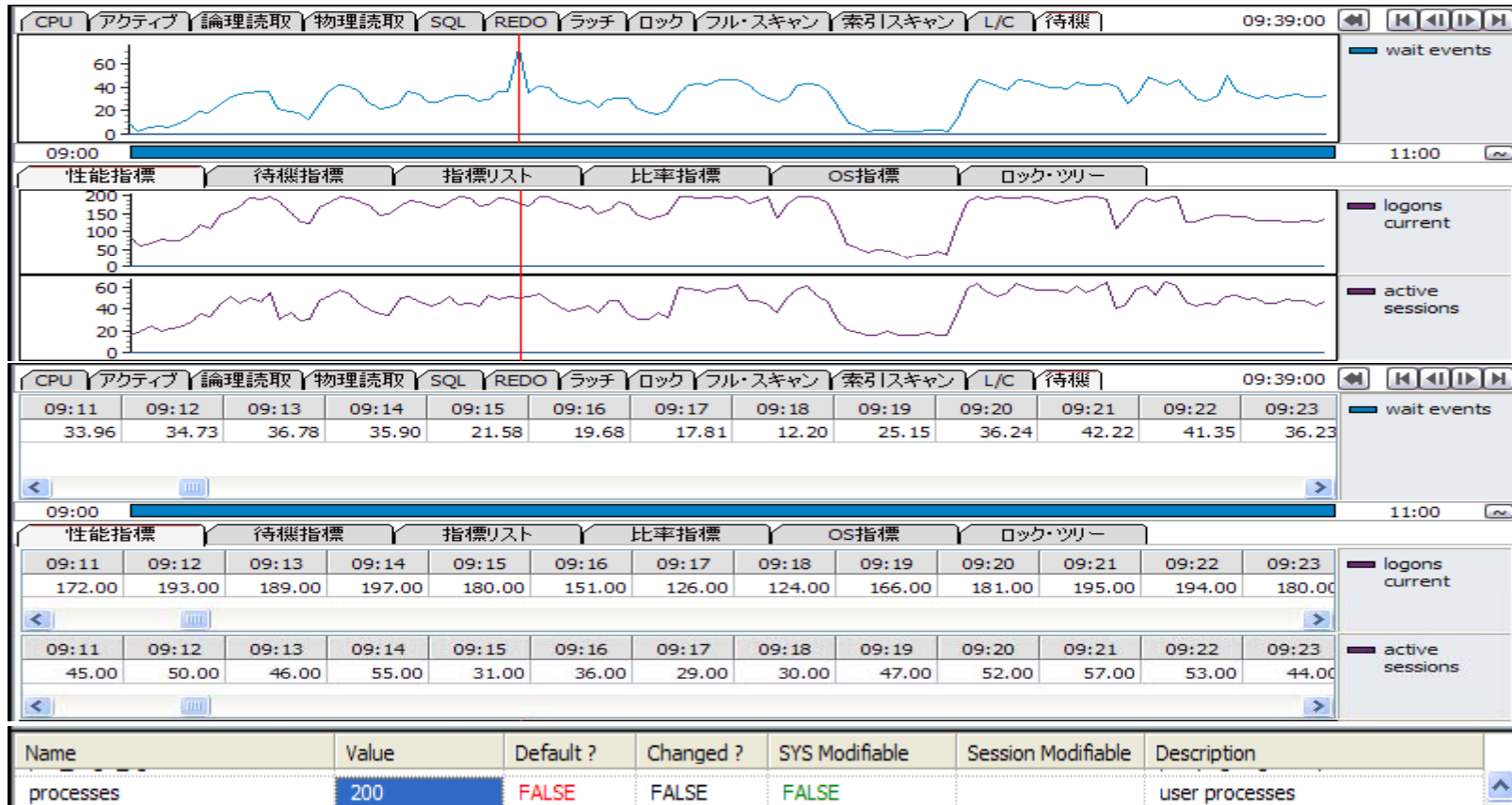
● システムレベル診断/分析:以下接続エラー発生当日のログ



- 接続エラー発生時間帯「9:00 ~ 11:00」前後で、「40% ~ 60%」のCPUが使われている。
- 同時間帯で、40秒以上まで滞留 (wait events) が急増した。
- 同時間帯のDB接続数が「200」近くまで達している。
- 同時間帯で、「40~60」のアクティブセッションが活動している。

診断/分析プロセス

● 特定(異常現象発生)時間帯の詳細分析:「9:00~11:00」時間帯



- 最大DB接続数 (processes) は「200」と設定されている
- 当時間帯で「197」まで接続 (1分後とにサンプリング) されている
- アクティブセッションの50%以上の時間が滞留で所要されている: 例) 33.96秒 (logons current) / 45個 (active sessions)

● 特定(異常現象発生)時間帯の詳細分析

指標リスト 09:00 ~ 11:00

性能指標 | 待機指標 | 比率指標

Event Name	Time Waited			Total Waits			Timeouts		
	Value/Sec	Diff	Total	Value/Sec	Diff	Total	Value/Sec	Diff	Total
db file parallel read	12.6	90,279.6	91,666.0	54.6	389,828	429,513	0.0	0	0
db file sequential read	7.9	56,252.0	57,307.0	516.4	3,686,863	4,036,490	0.0	0	0
read by other session	5.8	41,104.3	41,243.0	285.5	2,038,782	2,062,813	1.7	12,394	12,394
db file scattered read	1.1	7,874.2	8,168.0	236.4	1,688,077	1,901,979	0.0	0	0
Streams AQ: qmn slave ...	1.0	7,108.6	26,168.0	0.0	255	938	0.0	0	0
Streams AQ: qmn coord...	1.0	7,108.6	26,178.0	0.1	515	1,897	0.0	260	959
Streams AQ: waiting for...	0.5	3,692.0	11,976.0	0.0	2	6	0.0	2	6
direct path read	0.1	408.6	417.0	0.4	3,178	4,348	0.0	0	0
latch: shared pool	0.0	186.5	188.0	7.7	55,263	56,158	0.0	0	0
db file parallel write	0.0	174.0	175.0	2.8	20,198	22,818	0.0	0	0
latch: library cache	0.0	64.3	64.0	3.3	23,646	23,893	0.0	0	0
control file sequential read	0.0	53.6	66.0	0.5	3,718	13,611	0.0	0	0
log file sync	0.0	45.3	47.0	3.0	21,377	23,722	0.0	0	0
log file parallel write	0.0	42.8	82.0	6.3	45,098	138,682	0.0	0	0
class slave wait	0.0	29.9	99.0	0.0	6	20	0.0	6	20
latch free	0.0	25.2	28.0	0.8	5,576	5,621	0.0	0	0
SQL*Net break/reset to...	0.0	20.9	21.0	3.0	21,323	24,219	0.0	0	0
library cache load lock	0.0	8.0	7.0	0.0	11	11	0.0	0	0
control file parallel write	0.0	7.7	17.0	0.3	2,394	8,837	0.0	0	0
log file sequential read	0.0	7.7	7.0	0.0	207	219	0.0	0	0
enq: SQ - contention	0.0	5.8	5.0	0.0	52	53	0.0	0	0
library cache pin	0.0	5.0	5.0	0.0	350	351	0.0	0	0
buffer busy waits	0.0	3.5	3.0	0.0	251	252	0.0	0	0
kksfbc child completion	0.0	3.5	3.0	0.0	69	69	0.0	63	63
latch: cache buffers ch...	0.0	2.0	1.0	0.5	3,758	3,769	0.0	0	0
row cache lock	0.0	2.0	1.0	0.0	16	16	0.0	0	0
enq: TX - index contention	0.0	1.7	1.0	0.0	9	9	0.0	0	0

● 全体滞留の「91.1%」が物理I/O読取の滞留で発生している。

→ db file parallel read、db file sequential read、read by other session、db file scattered read

診断/分析プロセス

● 特定(異常現象発生)時間帯の詳細分析

The image displays three overlapping screenshots of the SQL Server Enterprise Manager interface, showing wait time analysis for different SQL queries. Each screenshot has a table with the following columns: SQL Text, Wait Time, Execute Count, Elapsed Time, and CPU Time.

Top Screenshot: db file parallel read

SQL Text	Wait Time	Execute Count	Elapsed Time	CPU Time
SELECT	7,044.9	98	15,118.6	165.6
select *	4,775.2	85	10,848.6	141.0
SELECT	3,834.0	204	7,923.1	77.6
SELECT	1,372.8	43	6,588.6	86.8

Middle Screenshot: db file sequential read

SQL Text	Wait Time	Execute Count	Elapsed Time	CPU Time
SELECT	2,454.1	98	15,118.6	165.6
SELECT	1,463.8	204	7,923.1	77.6
SELECT	1,257.9	43	6,588.6	86.8
select *	1,165.7	85	10,848.6	141.0

Bottom Screenshot: read by other session

SQL Text	Wait Time	Execute Count	Elapsed Time	CPU Time
SELECT	5,448.1	96	15,035.2	163.5
select *	4,761.3	85	10,848.6	141.0
SELECT	2,890.1	198	6,596.1	40.2
select *	2,637.7	197	5,806.1	37.4
SELECT	2,525.4	202	7,857.1	76.8
SELECT	1,969.8	43	6,588.6	86.8
select *	1,912.5	201	5,697.5	59.0
SELECT	1,256.3	62	5,165.1	62.8
select A	964.9	22	2,551.7	44.9
select *	876.0	42	2,705.8	36.8
SELECT	771.3	57	2,735.6	15.1
select *	755.3	54	2,679.4	13.7
SELECT	495.6	10	917.0	13.8
SELECT	461.3	37	2,381.2	25.9
SELECT	352.1	34	2,308.1	18.3
SELECT	287.6	64	1,324.0	8.9
select *	274.2	34	1,971.3	21.4
SELECT	269.8	26	1,958.3	14.0
SELECT	263.6	180	744.6	8.3
select *	261.6	6	791.3	8.3
SELECT	211.1	17	902.5	12.3
SELECT	205.2	39	1,359.3	6.6
SELECT	181.8	39	1,481.7	4.7
select *	158.8	54	834.9	4.8
SELECT	164.0	26	2,066.8	40.9
select *	162.9	8	551.1	2.9

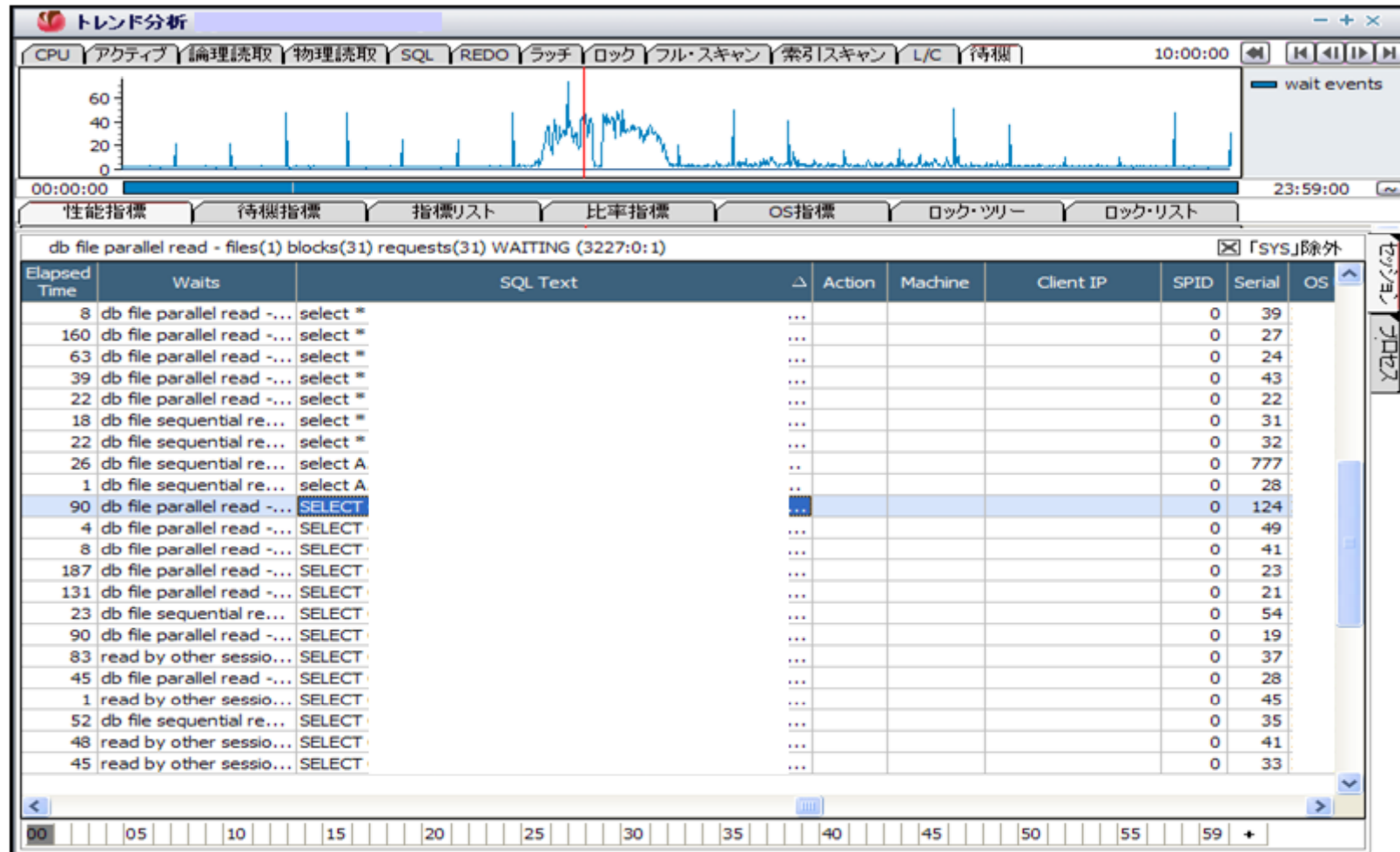
● 滞留関連SQLリストから、「特定SQL」が上位を占めていることが確認される

SELECT COUNT(*) FROM (select * from (SELECT DISTINCT ...

select * from (SELECT DISTINCT ...

診断/分析プロセス

● 特定(異常現象発生)時間帯のセッション分析



- ・「43」のユーザーアクティブセッションのうち、「30」セッションが特定SQLを実行している。

● 特定SQL分析

SQLリスト [: 08-06-30]

From 09:00 To 11:00 Schema Elapsed Time >= 0 Logical Reads >= 0
 SQL Text Program CPU >= 0 Physical Reads >= 0
 % Module Waiting >= 0
 Execut Count >= 0

Time	SQL Text	Elapsed (AVG)	Execut Count	Elapsed (SUM)	CPU (SUM)	Waiting (SUM)	CPU (AVG)	Waiting (AVG)	Logical Reads	Physical Reads
09:20	SELECT ...	135.861	14	1,902.05	23.90	1,878.15	1.707	134.154	565,850	153,983
10:00	SELECT ...	271.067	6	1,626.40	13.15	1,613.25	2.192	268.875	278,827	149,811
09:30	select * ...	114.245	19	2,170.65	34.25	2,136.40	1.803	112.442	755,458	142,109
09:50	SELECT ...	103.170	20	2,063.40	28.05	2,035.35	1.403	101.767	640,016	137,444
10:10	SELECT ...	183.406	8	1,467.25	12.45	1,454.80	1.556	181.850	263,689	132,952
09:10	SELECT ...	155.025	6	930.15	13.80	916.35	2.300	152.725	312,360	114,972
09:30	SELECT ...	121.060	20	2,421.20	34.25	2,386.95	1.712	119.347	686,946	111,792
09:20	select * ...	101.621	14	1,422.70	21.30	1,401.40	1.521	100.100	478,258	111,368
10:10	SELECT ...	5.661	9	50.95	2.30	48.65	0.256	5.406	126,834	108,497
09:20	SELECT ...	4.683	12	56.20	1.90	54.30	0.158	4.525	146,373	107,136
11:00	SELECT ...	257.275	4	1,029.10	8.50	1,020.60	2.125	255.150	186,737	106,448
10:00	select * ...	150.817	6	904.90	10.10	894.80	1.683	149.133	221,303	103,673
10:30	SELECT ...	189.133	9	1,702.20	15.15	1,687.05	1.683	187.450	360,224	98,010
10:10	SELECT ...	3.968	14	55.55	2.25	53.30	0.161	3.807	125,853	91,754
10:10	SELECT ...	10.750	4	43.00	2.45	40.55	0.613	10.137	100,744	89,800
09:10	SELECT ...	97.908	6	587.45	13.45	574.00	2.242	95.667	284,841	88,448
10:40	SELECT ...	435.433	3	1,306.30	10.00	1,296.30	3.333	432.100	154,024	88,135
10:10	select * ...	210.580	5	1,052.90	8.85	1,044.05	1.770	208.810	182,928	84,389
09:40	SELECT ...	186.858	6	1,121.15	9.35	1,111.80	1.558	185.300	244,507	82,284
		75,978		195,520.40	4,483.85	1,036.55			60,043,241	29,009,230

SELECT COUNT(*) FROM (select * from (SELECT

● 「Physical Reads」逆順のSQLリストから、「特定SQL」が上位を占めていることが確認される

SELECT COUNT(*) FROM (select * from (SELECT ...

select * from (SELECT DISTINCT...

● 特定SQL分析

Time	SQL Text	Elapsed (AVG)	Execute Count	Elapsed (SUM)	CPU (SUM)	Waiting (SUM)	CPU (AVG)	Waiting (AVG)	Logical Reads	Physical Reads
09:20	SELECT ...	135.861	14	1,902.05	23.90	1,878.15	1.707	134.154	565,850	153,983
10:00	SELECT ...	271.067	6	1,626.40	13.15	1,613.25	2.192	268.875	278,827	149,811
09:30	select *	114.245	19	2,170.65	34.25	2,136.40	1.803	112.442	755,458	142,109
09:50	SELECT ...	103.170	20	2,063.40	28.05	2,035.35	1.403	101.767	640,016	137,444
10:10	SELECT ...	183.406	8	1,467.25	12.45	1,454.80	1.556	181.850	263,689	132,952
09:10	SELECT ...	155.025	6	930.15	13.80	916.35	2.300	152.725	312,360	114,972
09:30	SELECT ...	121.060	20	2,421.20	34.25	2,386.95	1.712	119.347	686,946	111,792
09:20	select *	101.621	14	1,422.70	21.30	1,401.40	1.521	100.100	478,258	111,368
11:00	SELECT ...	257.275	4	1,029.10	8.50	1,020.60	2.125	255.150	186,737	106,448
10:00	select *	150.817	6	904.90	10.10	894.80	1.683	149.133	221,303	103,673
10:30	SELECT ...	189.133	9	1,702.20	15.15	1,687.05	1.683	187.450	360,224	98,010
09:10	SELECT ...	97.908	6	587.45	13.45	574.00	2.242	95.667	284,841	88,448
10:40	SELECT ...	435.433	3	1,306.30	10.00	1,296.30	3.333	432.100	154,024	88,135
10:10	select *	210.580	5	1,052.90	8.85	1,044.05	1.770	208.810	182,928	84,389
09:40	SELECT ...	186.858	6	1,121.15	9.35	1,111.80	1.558	185.300	244,507	82,284
09:50	select *	112.232	14	1,571.25	20.35	1,550.90	1.454	110.779	533,819	79,470
09:10	select *	140.088	4	560.35	8.80	551.55	2.200	137.887	213,038	68,331
09:10	select *	50.750	4	203.00	6.80	196.20	1.700	49.050	158,195	67,129
09:50	SELECT ...	28.504	28	798.10	9.45	788.65	0.338	28.166	240,911	64,638
			5,207	140,458.10	1,551.80	3,906.30			29,993,742	10,529,115

SELECT COUNT(*) FROM (select * from (SELECT DISTINCT

- 特定SQLは、DB全体処理時間の「71.8% = 140,458.1 / 195,520.4」、DB全体物理読取の「36% = 10,529,115 / 290,09,230」を占めている。
- 当SQLを改善できると、DB全体で最大「71.8%」の改善効果が予想される。
- 物理読取I/Oの競合による処理時間の急増現象が見られる

● 接続数に対する特定SQLの影響

接続エラー発生時間帯

時間帯	インスタンス		特定SQL			
	平均接続数	平均待機時間	[平均]処理時間	実行回数	[合計]処理時間	[合計]待機時間
6/2 09:00 - 12:00	120.47	24.76	28.54	5,090.00	145,290.90	144,069.70
6/2 12:00 - 18:00	51.47	4.35	3.22	4,951.00	15,944.70	15,002.05
6/26 09:00 - 12:00	61.46	4.84	3.83	3,440.00	13,169.10	12,515.95
6/26 12:00 - 18:00	49.25	3.48	2.10	4,177.00	8,766.10	8,126.60
6/30 09:00 - 12:00	134.24	27.91	26.89	6,879.00	184,987.40	182,983.55
6/30 12:00 - 18:00	42.03	4.34	3.66	4,788.00	17,545.35	16,591.60
7/2 09:00 - 12:00	58.65	4.40	3.78	3,253.00	12,280.70	11,650.60
7/2 12:00 - 18:00	49.51	3.50	2.30	3,893.00	8,963.05	8,307.20

- 接続エラーが発生しなかった時間帯は、平均待機時間が「3.50～4.84」秒、平均接続数が「42.03～61.46」
→ 特定SQLの平均処理時間は、「2.10～3.83」
- 接続エラーが発生した時間帯は、平均待機時間が「24.76～27.91」秒、平均接続数が「120.47～134.24」
→ 特定SQLの平均処理時間は、「26.89～28.54」で、正常時の10倍前後

診断/分析プロセス

● 改善案

改善案	作業コスト	リスク	改善効果	適用推奨順
①索引の見直し	中	中	大	A
②パーティション化	中	中	中	N/A
③特定SQLのチューニング	小	小	大	A
④バッファプールの拡張	小	小	中	B
⑤マルチバッファプールの活用	中	中	小	C
⑥「processes」パラメータの調整	小	小	大	B

「①～⑤」はI/O滞留を減らす効果が、⑥は接続エラーの閾値を高くする効果がある。その中、改善効果などを考慮し、
「③特定SQLチューニング、①索引の見直し」→「④バッファプールの拡張、⑥「processes」パラメータの調整」
→「⑤マルチバッファプールの活用」の順で、改善案の適用を推奨する。

● SQL改善例

```
SELECT *
FROM (
  SELECT DISTINCT ...
  FROM table1 b ,
       table2 a
  WHERE a.col1 = '0120'
  AND   a.col2 < SYSDATE
  AND   NVL( a.col3 , '9999/12/31' ) > SYSDATE
  AND   a.col1 = b.col1
  AND   a.col4 = b.col4
  ...
  AND   TO_CHAR( a.col5 , 'YYYYMMDD' ) >=
        TO_CHAR( SYSDATE - 7 , 'YYYYMMDD' )
  AND   a.col4 = 'ABC'
)
...
```

```
elapsed time    00:00:11.50
physical reads  28171
logical reads   29417
```



```
create index i4_table1 on table1
(col1, col4, col5, NVL( col3, '9999/12/31' ));

SELECT /*+ ORDERED INDEX(a i4_table1) */
  DISTINCT ...
FROM table1 a,
     table2 b
WHERE a.col1 = '0120'
  AND a.col2 < SYSDATE
  AND NVL( a.col3 , '9999/12/31' ) > SYSDATE
  AND a.col1 = b.col1
  AND a.col4 = b.col4
...
  AND a.col5 >= TRUNCATE( SYSDATE - 7 )
  AND a.col4 = 'ABC'
...
```

```
elapsed time    00:00:00.59
physical reads   87
logical reads   3905
```


診断/分析プロセス

● 診断/分析サマリー

- ✓ 接続エラーは、最大接続設定のOracle初期化パラメータ「processes=200」に引っかかって発生した。
- ✓ 接続数が「200」まで増加した原因は、「接続数の増加 → 普段流れるSQLの並列実行による滞留の急増 → 接続時間の延長 → 継続的な新規接続と既存接続で接続数の急増」の悪循環と推論される。

● 改善(チューニング)案

- ✓ 接続集中時滞留時間への影響が高いSQLに対する個別チューニングを行う。
- ✓ バッファークッシュを調整[↑]して、物理読取を削減する。
- ✓ 最大接続パラメータ「processes」を調整「200 → 300」する。