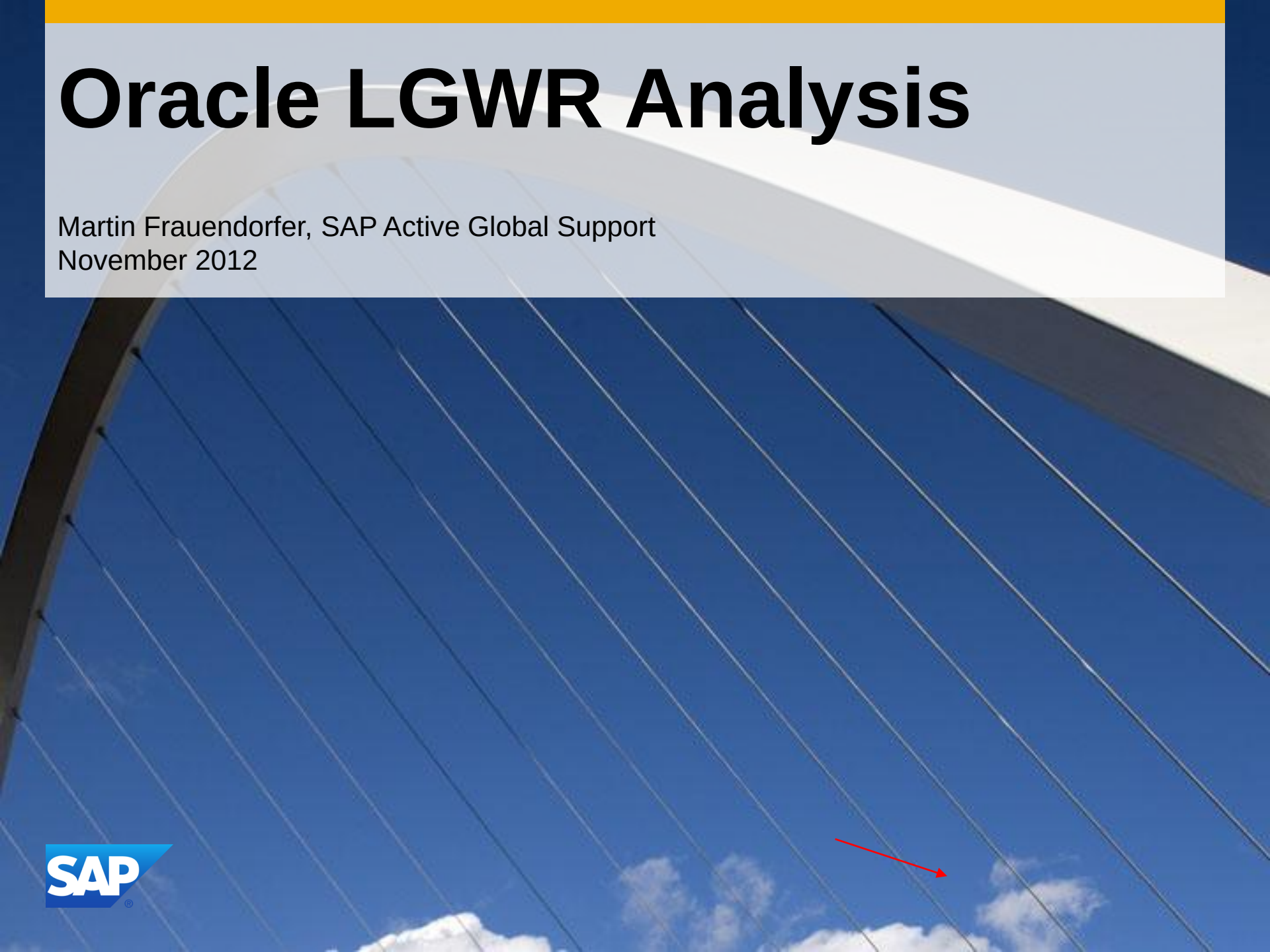
Oracle LGWR Analysis

Martin Frauendorfer, SAP Active Global Support
November 2012



Agenda

Table of Contents

- Introduction
- LGWR Overview
- Case Study – LGWR Analysis
- Other Scenarios
- Other useful Scripts

Introduction

Introduction

Remark

SQL: "<script_name>" in this presentation references SQL commands available via SAP Note 1438410.

Focus

LGWR problems like high "log file sync" wait times are sometimes difficult to analyze.

Not every LGWR problem is related to I/O bottlenecks related to the redo logs.

We discuss possibilities to analyze LGWR problems and interpret the results.

A particular focus is put on *SQL: "SystemStatistics_LGWR"*.

LGWR Overview

LGWR Overview

Typical components of the SAP database request time are:

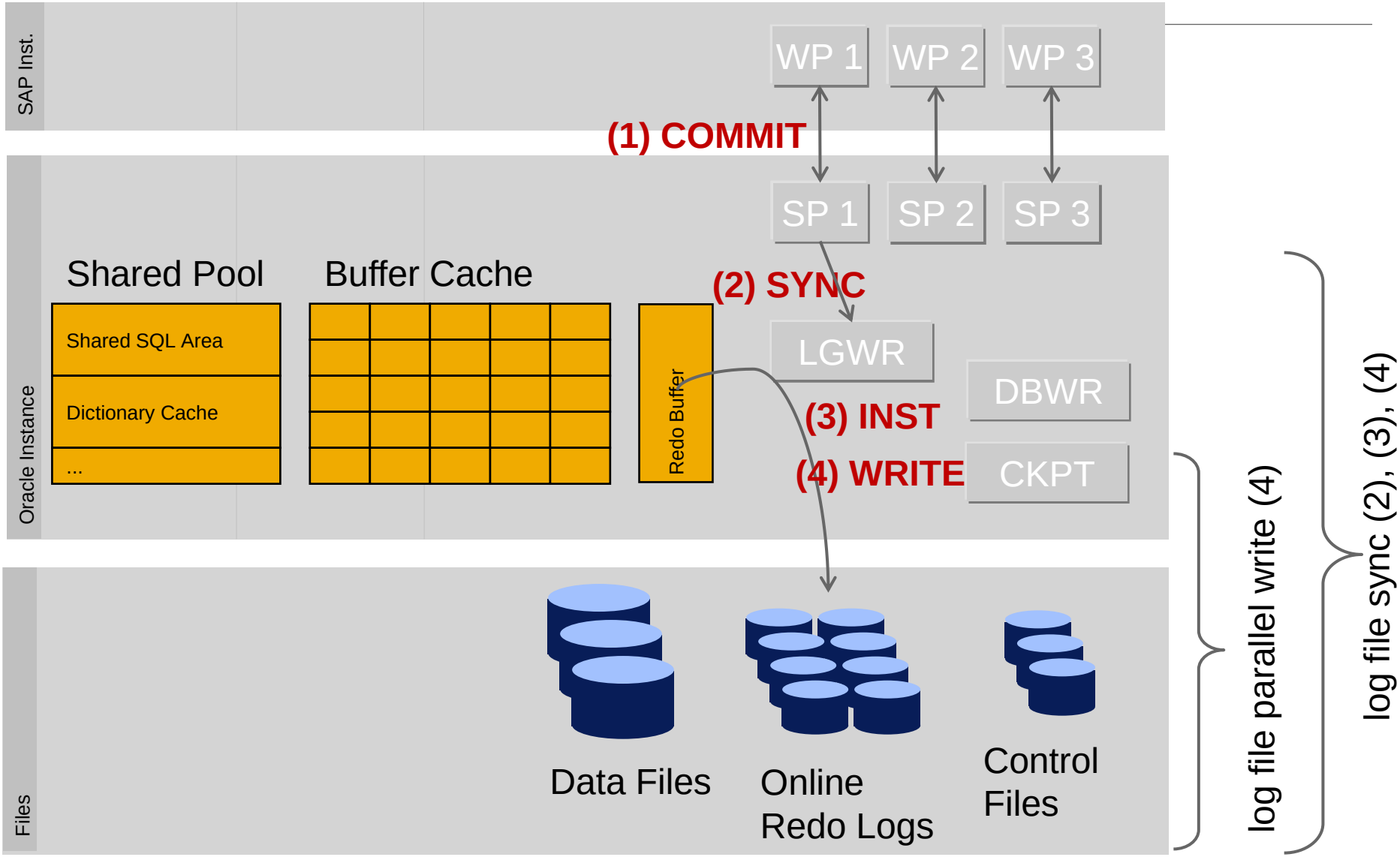
- 60 % I/O reads
- 20 % network / communication overhead between SAP and Oracle
- 10 % CPU consumption
- 5 % lock waits
- 3 % I/O writes (mainly LGWR related)
- 2 % others

Therefore LGWR is often not in the focus of performance tuning.

For various reasons LGWR can cause temporary or permanent performance problems (resulting in an increase from e.g. 3 % to more than 20 %).

This presentation provides some advice how to analyze these problems.

LGWR Overview



LGWR Overview

During a COMMIT the contents of the redo buffer need to be written to disk. Otherwise a crash could result in a loss of committed data.

The following steps are most important:

(1) Work process issues a COMMIT (including network communication)

(2) Shadow process triggers LGWR to synchronize the online redo logs with the redo buffer

(3) LGWR performs some (usually minor) instance activities

(4) LGWR issues a write request to the online redo logs

„*log file sync*“ is the total wait time from shadow process perspective (including (2), (3) and (4)).

„*log file parallel write*“ is the pure LGWR write time (4).

LGWR Overview

What data indicates a potential LGWR problem?

Average daily “log file sync” time > 10 ms

Average hourly “log file sync” time > 20 ms

“log file sync” responsible for more than 5 % of SAP database request time

What can be the reason for high “log file sync” times?

I/O bottleneck related to online redo logs (standard assumption)

Very high amount of redo log information

High amount of COMMIT operations

Bottleneck during LGWR instance activity

Other COMMIT related activities (e.g. on-commit materialized views, synchronous DataGuard replication)

LGWR Overview

The script *SQL: "SystemStatistics_LGWR"* can be used to retrieve important LGWR related information from the AWR histories. Output columns:

REDO_GB/H: Amount of redo logs generated per hour
WRITES/S: Number of "log file parallel write" requests per second
KB/WRITE: Average size of a "log file parallel write" request in KB
MS/WRT_REQ: Average "log file parallel write" time in ms
WRT_MB/S: Throughput of "log file parallel write" requests in MB per second
WRT_BUSY_%: Percentage of time the LGWR is busy with writing
SYNCS/S: Number of "log file sync" requests per second
SYNC_SESS: Average number of sessions waiting for "log file sync"
COMMITS/S: Average number of commits per second

Some columns were displayed differently in previous script versions, e.g.:

REDO_SIZE_MB instead of REDO_GB/H
REDO_WRITES instead of WRITES/S
MS/WRT_MB instead of WRT_MB/S

Case Study – LGWR Analysis

Case Study
LGWR Analysis

Initial situation

End users complain about bad performance in the recent hours

SQL: "TimedEvents_TopTimedEvents" shows the following database request time distribution for the critical hours:

EVENT_NAME	TOTAL_WAITS	TIME_WAITED	AVG_MS	PERCENT	ACT_SESS
BEGIN: 26.04.2012 08:00:07					
END: 26.04.2012 11:00:37					
INSTANCE: ALL					
log file sync	**1905695**	**106 h**	**200.17**	**36.46**	**26.43**
db file sequential read	32016018	98 h	11.01	33.69	24.43
NETWORK	374009531	42 h	0.40	14.30	10.37
CPU		25 h		8.46	6.14
Disk file operations I/O	91449	10 h	392.47	3.43	2.49
enq: TX - row lock contention	8999	5 h	2185.92	1.88	1.36
db file parallel read	230154	2 h	24.34	0.54	0.39
read by other session	527991	1 h	9.33	0.47	0.34
SQL*Net more data to client	14612312	0 h	0.12	0.17	0.12
db file scattered read	268392	0 h	6.07	0.16	0.11

- "log file sync" is top contributor to SAP database request time
- Average time of 200 ms significantly higher than thresholds of 10 to 20 ms

Case Study
LGWR Analysis

LGWR Details

Let's run the most important SQL command for understanding the LGWR behavior - *SQL: "SystemStatistics_LGWR":*

BEGIN_TIME	REDO_SIZE_MB	REDO_WRITES	KB/WRITE	MS/WRT_REQ	WRT_MB/S	WRT_BUSY_%	SYNC_REQS	MS/SYNC_REQ	SYNC_SESS	COMMITS/S
2012-04-26 12:00:50	14767.04	694215	21.27	0.81	26.23	15.63	732366	1.50	0.31	202.56
2012-04-26 11:00:37	13927.50	594603	23.42	1.93	12.12	31.82	677401	359.39	67.38	185.13
2012-04-26 10:00:30	12676.10	609875	20.78	1.48	14.05	25.01	679809	288.12	54.30	187.80
2012-04-26 09:00:16	11942.12	555668	21.49	1.54	13.96	23.66	591906	238.93	39.13	163.11
2012-04-26 08:00:07	10335.73	626520	16.50	1.01	16.41	17.46	655732	67.34	12.24	182.03
2012-04-26 07:00:06	7332.97	440234	16.66	0.76	21.98	9.27	456616	1.24	0.16	127.03
2012-04-26 06:00:55	10211.55	481516	21.21	0.97	21.93	13.12	503361	16.61	2.35	141.45
2012-04-26 05:00:54	5652.38	339563	16.65	0.63	26.51	5.92	345186	0.95	0.09	95.16
2012-04-26 04:00:52	5132.18	297113	17.27	1.33	13.01	10.95	312900	2.13	0.18	86.46
2012-04-26 03:00:51	5349.15	354748	15.08	0.60	25.33	5.86	366439	0.96	0.10	101.29

- No particularly high amount of redo generation (REDO_SIZE_MB)
- No particularly high amount of COMMITs (COMMITS/S)
- "log file parallel write" (WRT_MB/S) remains on a very good level
- "log file sync" (MS/SYNC_REQ) increases massively
- LGWR not very busy (WRT_BUSY_%)

Case Study
LGWR Analysis

First conclusion

I/O to the redo logs is not the problem (because “log file parallel write” is much quicker than “log file sync”)

High redo log volume is not the problem

High COMMIT frequency is not the problem

What remains? Perhaps other LGWR instance activities?

Let’s check what LGWR is doing over time…

This can be done in a quite reliable way using information from the Active Session History

Case Study
LGWR Analysis

LGWR activities

SQL: "ASH_AggregationPerTimeSlice":

BEGIN_TIME	OCC_TOTAL	ACT_SESS	FIGURE_1	OCC_1	PCT_1	SESS_1
Aggregated by:	EVENT					
Instance:	ALL					
Program:	%LGWR%					
Time slice:	60 s					
26.04.2012 11:37:00	2	0.33	log file parallel write	2	100	0.33
26.04.2012 11:36:00	5	0.83	log file parallel write	5	100	0.83
26.04.2012 11:35:00	5	0.83	log file parallel write	4	80	0.67
26.04.2012 11:34:00	4	0.67	log file parallel write	4	100	0.67
26.04.2012 11:33:00	6	1.00	enq: CF - contention (5 / Share/Sub-Exclusive)	3	50	0.50
26.04.2012 11:32:00	6	1.00	enq: CF - contention (5 / Share/Sub-Exclusive)	6	100	1.00
26.04.2012 11:31:00	4	0.67	enq: CF - contention (5 / Share/Sub-Exclusive)	3	75	0.50
26.04.2012 11:30:00	3	0.50	enq: CF - contention (5 / Share/Sub-Exclusive)	2	67	0.33
26.04.2012 11:28:00	1	0.17	log file parallel write	1	100	0.17
26.04.2012 11:27:00	1	0.17	log file parallel write	1	100	0.17
26.04.2012 11:25:00	2	0.33	log file parallel write	2	100	0.33
26.04.2012 11:24:00	1	0.17	LGWR wait for redo copy	1	100	0.17
26.04.2012 11:23:00	2	0.33	log file parallel write	2	100	0.33
26.04.2012 11:22:00	2	0.33	log file parallel write	1	50	0.17
26.04.2012 11:20:00	2	0.33	log file parallel write	2	100	0.33

- Most of the time LGWR works as expected ("log file parallel write")
- From time to time it waits for "enq: CF – contention" (controlfile enqueue)
- This is related to LGWR instance activity

Case Study
LGWR Analysis

"log file sync" Waits

Let's check if the CF enqueue waits correlate to increased "log file sync" waits (*SQL: "ASH_AggregationPerTimeSlice"*):

BEGIN_TIME	OCC_TOTAL	ACT_SESS	FIGURE_1	OCC_1	PCT_1	SESS_1
Aggregated by:	EVENT					
Event:	log file sync					
Instance:	ALL					
Time slice:	60 s					
26.04.2012 11:37:00	565	94.17	log file sync	565	100	94.17
26.04.2012 11:36:00	1526	254.33	log file sync	1526	100	254.33
26.04.2012 11:35:00	1283	213.83	log file sync	1283	100	213.83
26.04.2012 11:34:00	1069	178.17	log file sync	1069	100	178.17
26.04.2012 11:33:00	1467	244.50	log file sync	1467	100	244.50
26.04.2012 11:32:00	2081	346.83	log file sync	2081	100	346.83
26.04.2012 11:31:00	575	95.83	log file sync	575	100	95.83
26.04.2012 11:30:00	312	52.00	log file sync	312	100	52.00
26.04.2012 11:29:00	1	0.17	log file sync	1	100	0.17
26.04.2012 11:28:00	1	0.17	log file sync	1	100	0.17
26.04.2012 11:27:00	1	0.17	log file sync	1	100	0.17
26.04.2012 11:26:00	2	0.33	log file sync	2	100	0.33
26.04.2012 11:24:00	1	0.17	log file sync	1	100	0.17
26.04.2012 11:23:00	2	0.33	log file sync	2	100	0.33
26.04.2012 11:22:00	1	0.17	log file sync	1	100	0.17
26.04.2012 11:20:00	2	0.33	log file sync	2	100	0.33

- When the LGWR waits for the CF enqueue, more and more processes wait for "log file sync".
- So the CF enqueue waits are the reason for the "log file sync" problem.

Case Study
LGWR Analysis

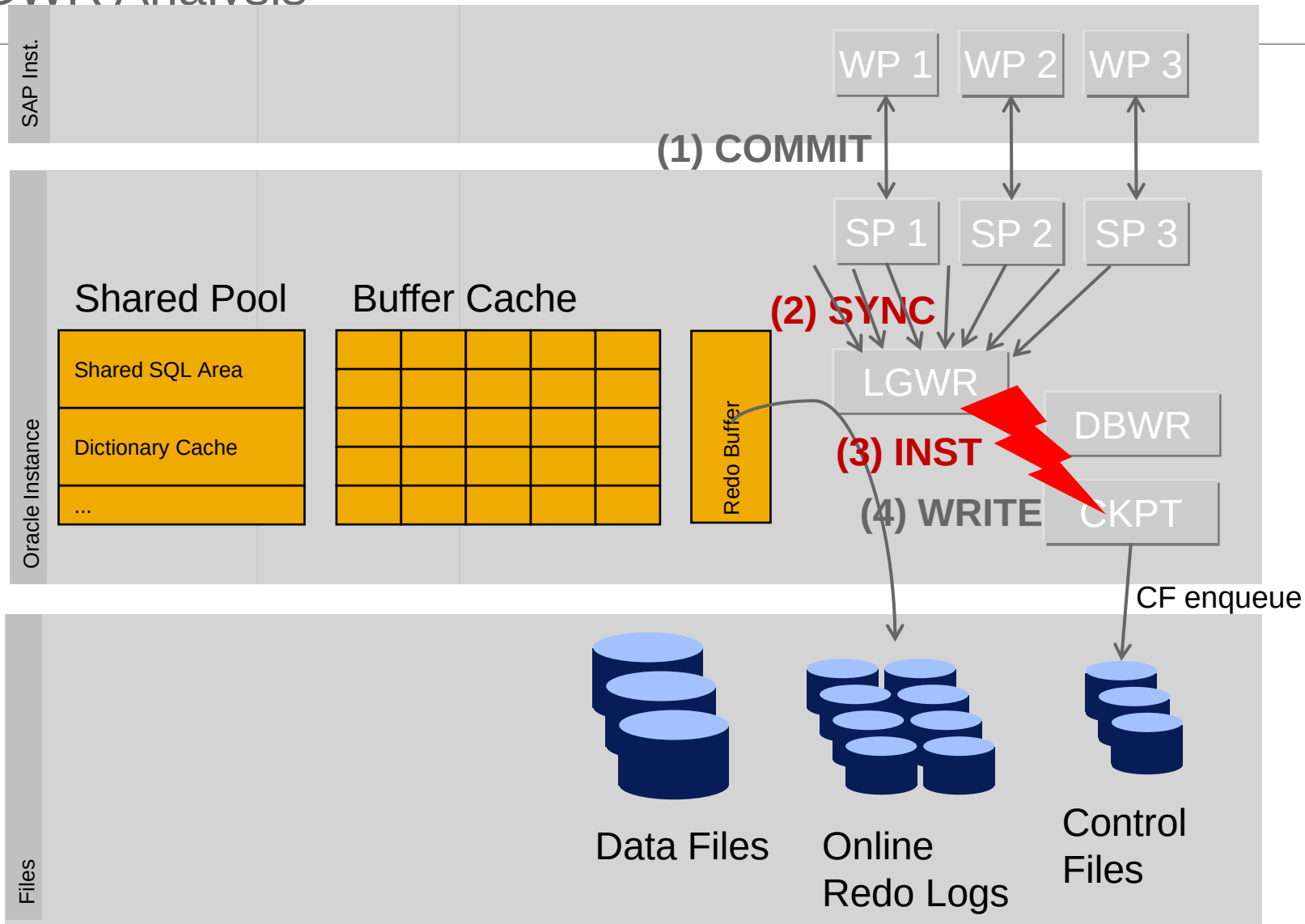
Identification of the blocking session

What is the blocking session doing that holds the CF enqueue (*SQL: "Locks_LockHolderActivities"*)?

PROGRAM	ACTION	NUM_WAITERS	WAITERS_%	AVG_WAITERS	SAMPLE_TIMES
BEGIN TIME:	15.03.2012 00:00:54				
END TIME:	26.04.2012 12:00:50				
SQL_ID:	any				
EVENT:	enq: CF - contention				
oracle@kftdershep11 (CKPT)	direct path read	227	24.97	1.14	200
oracle@kftdershep11 (CKPT)	CPU	267	29.37	1.34	199
oracle@kftdershep11 (CKPT)	control file parallel write	172	18.92	1.16	148
oracle@kftdershep11 (CKPT)	control file sequential read	119	13.09	1.17	102
oracle@kftdershep11 (CKPT)	direct path write	56	6.16	1.12	50
oracle@kftdershep11 (LGWR)	control file parallel write	15	1.65	1.00	15
oracle@kftdershep11 (LGWR)	control file sequential read	15	1.65	1.00	15
oracle@kftdershep11 (MMON)	control file parallel write	13	1.43	1.00	13
rman@kftdershep11 (TNS V1-V3)	control file sequential read	7	0.77	1.17	6
oracle@kftdershep11 (ARC0)	log file sequential read	3	0.33	1.00	3
oracle@kftdershep11 (MMON)	control file sequential read	3	0.33	1.00	3
oracle@kftdershep11 (CKPT)	db file single write	2	0.22	2.00	1
rman@kftdershep11 (TNS V1-V3)	db file sequential read	5	0.55	5.00	1
rman@kftdershep11 (TNS V1-V3)	control file parallel write	1	0.11	1.00	1
oracle@kftdershep11 (M000)	control file sequential read	1	0.11	1.00	1
oracle@kftdershep11 (CKPT)	db file sequential read	1	0.11	1.00	1
rman@kftdershep11 (TNS V1-V3)	db file single write	1	0.11	1.00	1
oracle@kftdershep11 (ARC2)	control file sequential read	1	0.11	1.00	1

- The CKPT process holds the CF enqueue.

Case Study
LGWR Analysis



Case Study
LGWR Analysis

Why does CKPT hold the CF enqueue for so long?

SQL: "ASH_Filter":

```
-------------------------------------------------------------------------------------
|SAMPLE_TIME        |ACTION                      |TIME_US   |PROGRAM                 |
-------------------------------------------------------------------------------------
|2012-04-26 11:36:37|control file parallel write |   9107411|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:36|control file parallel write |         0|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:35|control file parallel write |         0|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:34|control file parallel write |         0|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:33|control file parallel write |         0|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:32|control file parallel write |         0|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:31|control file parallel write |         0|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:30|control file parallel write |         0|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:13|control file sequential read|  29481623|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:12|control file sequential read|         0|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:11|control file sequential read|         0|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:10|control file sequential read|         0|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:09|control file sequential read|         0|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:08|control file sequential read|         0|oracle@kftdershep11 (CKPT)|
|2012-04-26 11:36:07|control file sequential read|         0|oracle@kftdershep11 (CKPT)|
-------------------------------------------------------------------------------------
```

- Different CKPT I/O activities (mainly related to the controlfiles) take ages
- 9 seconds and 29 seconds in the above extract for simple controlfile accesses!
- The bad I/O to the controlfiles results in longer CF enqueue lock times
- This massively impacts the LGWR and the „log file sync" waits

Case Study
LGWR Analysis

How to resolve the problem?

No problem with I/O to the online redo logs!

Instead **optimize the I/O to the controlfiles** by eliminating bottlenecks and misconfigurations on operating system, file system and I/O sub system level.

- Customer adapted I/O sub system configuration and problem disappeared

Also **consider Oracle bug 9095696**: "log file sync" wait time spikes with ARCHIVE_LAG_TARGET set

- Symptom: LGWR waiting for the CF enqueue in SSX-mode (REQUEST=5)
- Bug increases the risk of CF enqueues of LGWR in general
- Workaround: ARCHIVE_LAG_TARGET = 0 (was set to 1800)

Other Scenarios

Other Scenarios

High redo log volume

BEGIN_TIME	REDO_SIZE_MB	REDO_WRITES	KB/WRITE	MS/WRT_REQ	MS/WRT_MB	WRT_BUSY_%	SYNC_REQS	MS/SYNC_REQ	SYNC_SESS
2011-08-27 04:00:54	234376.22	54690	4285.54	55.99	13.06	85.53	287617	148.58	11.94
2011-08-27 03:00:06	230504.38	75387	3057.61	42.25	13.82	87.31	305624	125.29	10.50
2011-08-27 02:01:01	252411.09	164869	1530.98	18.21	11.90	84.71	340094	60.99	5.85
2011-08-27 01:00:04	252918.10	200415	1261.97	15.67	12.42	85.90	420512	47.24	5.43
2011-08-27 00:00:30	259763.78	154673	1679.44	19.74	11.75	85.43	280822	48.94	3.85
2011-08-26 23:00:31	231776.89	181921	1274.05	17.61	13.82	89.02	436321	52.59	6.38
2011-08-26 22:00:27	259981.43	118755	2189.23	27.81	12.70	91.65	319604	73.06	6.48
2011-08-26 21:00:31	267729.82	173948	1539.14	17.99	11.69	87.02	377382	48.68	5.11
2011-08-26 20:00:42	267209.80	248783	1074.07	13.28	12.37	92.08	783782	42.91	9.37
2011-08-26 19:00:48	203710.80	284561	715.88	10.09	14.10	79.93	469715	26.02	3.40
2011-08-26 18:00:44	164025.60	309638	529.73	8.63	16.29	74.14	468364	21.26	2.76
2011-08-26 17:00:46	156315.09	290785	537.56	8.90	16.56	71.93	410324	19.13	2.18
2011-08-26 16:00:51	126850.77	303133	418.47	7.79	18.61	65.68	399720	16.64	1.85
2011-08-26 15:00:56	110244.59	371630	296.65	6.16	20.75	63.64	478730	13.46	1.79
2011-08-26 14:00:07	130442.29	398481	327.35	6.47	19.77	70.69	579362	17.14	2.72
2011-08-26 13:00:13	130581.77	398841	327.40	6.64	20.28	73.67	630041	17.00	2.98
2011-08-26 12:00:21	103841.00	407900	254.57	5.63	22.10	63.90	551421	13.48	2.07
2011-08-26 11:00:26	92580.41	422322	219.22	5.05	23.03	59.32	540911	12.13	1.82
2011-08-26 10:00:25	60367.46	501994	120.26	3.53	29.33	49.16	605216	8.24	1.38
2011-08-26 09:00:15	49384.47	534569	92.38	3.07	33.21	45.43	644725	6.81	1.22
2011-08-26 08:00:25	37698.42	659156	57.19	2.36	41.24	43.31	780144	5.78	1.26
2011-08-26 07:00:24	25740.41	693959	37.09	2.02	54.54	38.99	796770	4.04	0.89

- Very high redo volume of up to 260 GB per hour (72 MB / s) (REDO_SIZE_MB)
- High write sizes up to 4 MB (KB/WRITE)
- Very good write throughput of 12 ms / MB or 300 GB / h (MS/WRT_REQ)
- LGWR up to 92 % busy with writing (WRT_BUSY_%)

Other Scenarios

High redo log volume – What to do

Use NOLOGGING operations if possible (e.g. ALTER INDEX REBUILD ... NOLOGGING)

Check for tables and indexes with high amount of „DB Block Changes“ (*SQL: „SegmentStatistics_TopSegmentsForStatisticPerAWRInterval“*).

Check from an SAP perspective if unnecessary DML operations are executed against tables returned above and eliminate them (e.g. deactivation of DBACOCKPIT histories)

Check if indexes with a high amount of „DB Block Changes“ are really required or if they can be dropped / merged

Minimize the times of tablespaces being in backup mode (this also results in an increased amount of redo logs being generated)

Use delta loads rather than full loads

Avoid OLTP Table Compression if you are already at the limit with redo generation (because it will significantly increase the redo log generation)

Other Scenarios

High COMMIT frequency

BEGIN_TIME	REDO_SIZE_MB	REDO_WRITES	KB/WRITE	MS/WRT_REQ	MS/WRT_MB	WRT_BUSY_%	SYNC_REQS	MS/SYNC_REQ	SYNC_SESS
2011-09-20 13:00:02	4734.23	238663	19.84	2.68	135.08	17.55	256755	5.46	0.39
2011-09-20 12:00:25	6654.97	207633	32.05	3.27	102.02	18.98	225154	8.63	0.54
2011-09-20 11:00:36	3968.69	194693	20.38	2.74	134.57	14.88	209864	5.04	0.29
2011-09-20 10:00:47	4815.60	179046	26.90	3.74	139.13	18.67	194299	11.44	0.62
2011-09-20 09:00:21	17792.61	252206	70.55	5.91	83.81	41.12	306591	30.81	2.60
2011-09-20 08:00:47	16860.62	276703	60.93	6.48	106.35	50.17	456786	44.72	5.72
2011-09-20 07:00:04	24446.38	381857	64.02	7.12	111.25	74.65	668113	31.37	5.75
2011-09-20 06:00:29	29683.76	381753	77.76	7.27	93.44	77.58	745656	29.67	6.19
2011-09-20 05:00:43	9059.47	212932	42.55	4.65	109.18	27.58	256010	12.76	0.91
2011-09-20 04:00:57	1536.50	95071	16.16	3.03	187.48	8.03	98090	7.82	0.21
2011-09-20 03:00:16	4393.92	353440	12.43	2.96	237.82	28.70	431193	8.15	0.97
2011-09-20 02:00:36	4364.25	212876	20.50	3.15	153.59	18.72	231280	5.69	0.37

- Increased „log file sync“ times between 06:00 and 10:00 (MS/SYNC_REQ)
- Small write sizes (KB/WRITE)
- As a consequence a small write throughput (MS/WRT_MB)
- „log file sync“ time (MS/SYNC_REQ) significantly higher than „log file parallel write“ time (MS/WRT_REQ)
- This can indicate a serialization of COMMITs
- One „log file sync“ request has to wait for several preceeding „log file parallel write“ requests
- Newer script versions also include a column COMMITS/S that will make it clearer

Other Scenarios

High COMMIT frequency – What to do

Check for applications being responsible for high amount of COMMIT

Increase there package size / reduce the COMMIT frequency on application side

The situation on the last slide was generated by a BALHDR / BALDAT deletion with a COMMIT after each record

Other Scenarios

LGWR I/O Problem (minor impact)

BEGIN_TIME	REDO_SIZE_MB	REDO_WRITES	KB/WRITE	MS/WRT_REQ	MS/WRT_MB	WRT_BUSY_%	SYNC_REQS	MS/SYNC_REQ	SYNC_SESS
10.05.2011 09:00:57	218.85	2435	89.88	**39.95**	444.54	2.70	1473	119.86	0.05
10.05.2011 08:01:00	135.47	2198	61.63	**44.30**	718.84	2.71	1328	124.54	0.05
10.05.2011 07:01:00	137.48	1959	70.18	**43.94**	626.11	2.39	1182	139.54	0.05
10.05.2011 06:00:05	165.06	1977	83.49	**44.77**	536.23	2.42	1034	114.79	0.03
10.05.2011 05:00:08	197.65	2052	96.32	**44.20**	458.85	2.52	902	96.04	0.02
10.05.2011 04:00:26	128.75	1666	77.28	**43.54**	563.35	2.02	729	117.72	0.02
10.05.2011 03:00:25	130.82	1741	75.14	**37.77**	502.68	1.83	590	49.76	0.01
10.05.2011 02:00:22	140.29	2037	68.87	**25.76**	374.01	1.46	411	14.43	0.00
10.05.2011 01:00:28	108.73	2210	49.20	**24.04**	488.53	1.48	403	33.57	0.00
10.05.2011 00:00:27	336.78	3729	90.31	**20.35**	225.28	2.11	2105	12.89	0.01
09.05.2011 23:00:33	173.02	4986	34.70	**17.73**	511.05	2.46	3791	25.75	0.03
09.05.2011 22:00:36	198.45	3395	58.45	**28.47**	487.08	2.69	1887	58.08	0.03
09.05.2011 21:00:44	198.27	3279	60.47	**26.64**	440.56	2.43	2028	68.55	0.04

- Small amount of redo logs (REDO_SIZE_MB)
- LGWR not busy (WRT_BUSY_%)
- Small write sizes (KB/WRITE)
- But very high „log file parallel write“ times (MS/WRT_REQ)
- As a consequence also very high „log file sync“ times (MS/SYNC_REQ)

Other Scenarios

LGWR I/O Problem (major impact)

BEGIN_TIME	REDO_SIZE_MB	REDO_WRITES	KB/WRITE	MS/WRT_REQ	MS/WRT_MB	WRT_BUSY_%	SYNC_REQS	MS/SYNC_REQ	SYNC_SESS	COMMITS/S
2012-08-30 19:14:10	295.69	31349	9.43	1.38	146.74	11.76	39015	2.29	0.24	94.12
2012-08-30 18:59:30	195.53	9704	20.15	1.62	80.50	1.79	8549	1.30	0.01	4.06
2012-08-30 13:30:21	556.69	62	8978.89	11317.10	1260.41	35.31	9530	21547.35	103.34	4.44
2012-08-30 13:20:50	520.84	56	9300.71	14615.18	1571.41	143.34	8880	29014.00	451.22	14.40
2012-08-30 13:10:28	359.47	45	7988.18	13996.00	1752.09	101.26	6947	27801.50	310.51	10.54
2012-08-30 13:00:18	584.46	76	7690.28	7677.76	998.37	95.66	10242	15093.64	253.42	15.93
2012-08-30 12:50:08	552.48	78	7083.04	7765.00	1096.28	99.29	11020	15520.94	280.39	17.24
2012-08-30 12:40:57	533.16	86	6199.54	6205.23	1000.92	96.85	11468	12103.74	251.92	19.90
2012-08-30 12:30:38	660.70	101	6541.62	5974.55	913.31	97.48	12699	11801.15	242.10	19.62
2012-08-30 12:20:23	662.88	90	7365.35	6676.00	906.41	97.70	11394	13289.71	246.22	17.37
2012-08-30 12:11:05	466.12	61	7641.33	7746.07	1013.71	84.68	8487	15573.83	236.87	14.51
2012-08-30 12:00:10	760.79	98	7763.16	7793.27	1003.88	116.60	13299	15548.35	315.69	19.34
2012-08-30 11:50:37	582.76	81	7194.56	7084.32	984.68	100.14	12352	14092.57	303.79	20.04
2012-08-30 11:40:18	517.70	73	7091.83	8321.51	1173.39	98.14	12152	16304.76	320.09	18.51
2012-08-30 11:31:01	570.06	86	6628.62	6082.79	917.66	93.92	12821	11836.93	272.46	21.84
2012-08-30 11:20:45	799.17	124	6444.90	5287.82	820.47	106.44	17128	10343.73	287.61	25.98
2012-08-30 11:10:29	676.97	108	6268.24	5563.80	887.62	97.55	15460	10891.24	273.34	23.38
2012-08-30 11:00:11	707.79	155	4566.36	4031.35	882.84	101.11	18570	7816.12	234.86	28.50
2012-08-30 10:50:58	891.87	505	1766.07	1105.74	626.10	100.98	25103	3483.84	158.15	42.44
2012-08-30 10:40:33	791.72	235	3369.02	2425.66	719.99	91.20	22728	4887.71	177.74	34.48
2012-08-30 10:30:09	655.58	180	3642.13	3693.06	1013.98	106.53	19914	7514.59	239.82	30.07
2012-08-30 10:20:56	1344.08	30562	43.98	7.48	170.15	41.35	38813	24.37	1.71	67.77
2012-08-30 10:10:38	2145.76	72536	29.58	4.69	158.50	55.03	84608	8.17	1.12	82.39

- Small amount of redo logs (REDO_SIZE_MB)
- LGWR very busy (WRT_BUSY_%)
- Large write sizes (KB/WRITE)
- Very high „log file parallel write“ (MS/WRT_REQ) and „log file sync“ times (MS/SYNC_REQ)

Other Scenarios

LGWR I/O Problem – What to do

Check the layers below Oracle for bottlenecks:

- Operating system
- File system
- Network to I/O sub system
- I/O sub system

Get in touch with your operating system and hardware partners for that purpose

Typical problem situations are:

- Inadequate file system block sizes
- Network overload (e.g. due to concurrent backup via same network)
- Synchronous mirroring of redo logs on I/O sub system level
- Inadequate file system mount options
- Hot spots in I/O sub system

See SAP Note 793113 for further information regarding Oracle I/O configuration.

Other Scenarios

LGWR I/O Problem (special configuration)

BEGIN_TIME	REDO_GB/H	WRITES/S	KB/WRITE	MS/WRT_REQ	WRT_MB/S	WRT_BUSY_%	SYNCS/S	MS/SYNC	SYNC_SESS	COMMITS/S
2012-11-12 15:45:14	49.43	157.27	89.40	5.67	15.77	89.17	499.40	13.32	6.65	480.25
2012-11-12 15:30:10	83.99	130.98	182.40	7.09	25.73	92.85	944.37	25.03	23.63	904.50
2012-11-12 15:15:38	70.94	139.73	144.40	6.60	21.88	92.21	912.99	22.87	20.88	874.51
2012-11-12 15:00:33	95.67	116.23	234.13	8.08	28.99	93.89	914.99	28.63	26.20	879.70
2012-11-12 14:45:21	53.87	151.71	101.00	5.90	17.12	89.50	522.06	15.44	8.06	506.84
2012-11-12 14:30:06	70.48	137.74	145.55	6.69	21.75	92.17	577.31	24.13	13.93	568.93
2012-11-12 14:15:03	47.38	155.65	86.58	5.57	15.55	86.69	516.04	13.14	6.78	508.88
2012-11-12 14:00:16	67.27	142.40	134.38	6.40	20.99	91.16	576.98	20.12	11.61	568.90
2012-11-12 13:45:15	57.07	158.17	102.62	5.75	17.83	91.02	597.95	13.26	7.93	590.56
2012-11-12 13:30:08	45.93	163.39	79.97	5.44	14.69	88.93	597.76	13.31	7.96	590.12
2012-11-12 13:15:04	46.92	154.92	86.14	5.67	15.19	87.87	562.01	16.25	9.14	553.89
2012-11-12 13:00:31	69.03	137.34	142.96	6.62	21.60	90.92	593.28	24.49	14.53	584.06
2012-11-12 12:45:22	51.43	155.37	94.16	5.63	16.74	87.40	537.24	13.51	7.26	529.42
2012-11-12 12:30:16	43.22	162.92	75.46	5.18	14.56	84.46	551.82	14.85	8.20	543.70

- LGWR quity busy (WRT_BUSY_%)
- Acceptable „log file parallel write“ times (MS/WRT_REQ)
- Rather high „log file sync“ times (MS/SYNC_REQ)
- Could be caused by high redo generation (REDO_GB/H) or high COMMITs (COMMITS/S)
- But the concerned system is a high-end system that should be able to handle this load

Other Scenarios

LGWR I/O Problem (special configuration) – What to do

An analysis of the LGWR activities using *SQL: „ASH_Aggregation“* showed this result:

EVENT	OCCURRENCES	SESSIONS	PERCENT
LGWR-LNS wait on channel	7844	0.52	56.33
log file parallel write	5925	0.39	42.55
LGWR wait for redo copy	95	0.01	0.68

So the biggest part of the LGWR write activity was not „log file parallel write“ but „LGWR-LNS wait on channel“. This wait event is related to a Dataguard standby database where LGWR applies the redo data. Normally that is not an issue, but here the system was set up with:

- MAXIMUM AVAILABILITY mode (→ synchronous replication)
- Distance of several hundred kilometers between the data centers

→ The overhead is normal for this configuration

→ Either accept this overhead, change to MAXIMUM PERFORMANCE or reduce the distance between the data centers

Other useful Scripts

SQL: „IO_IOActivityPerAWRInterval“

Displays I/O read and write load and average times (including „log file sync“, column LFS_AVG_MS) from the Oracle AWR histories.

SQL: „IO_RedoLogsPerHour“

Shows the number of redo log switches per hour

SQL: „TimedEvents_EventHistogram“
SQL: „TimedEvents_EventHistogramPerAWRInterval“

Provide information how often wait times with different durations showed up since database start / in AWR history (specify EVENT_NAME = ‚log file sync‘ or ‚log file parallel write‘)

SQL: „TimedEvents_TopTimedEvents“
SQL: „TimedEvents_TopTimedEventsPerAWRInterval“

Display the top wait events in AWR history (summarized or per AWR intervals), can be used to check if „log file sync“ is responsible for a significant fraction of the database server time

Thank You!

Contact information:

Martin Frauendorfer
Support Architect
SAP AG
Active Global Support - CoE EMEA Tech. Appl. Platf.
Raiffeisenstr. 45
martin.frauendorfer@sap.com