

表空间暴涨原因核查

类别

Oracle

2014 年 6 月 25 号客户的 users 表空间暴涨了 900G，经过查询系统监控记录，找到了相关的 sql 语句和责任人，具体过程如下：

这里需要先说明一个情况，由于之前 users 表空间使用率达到了 99%后，由于使用的是 bigfile，无法添加文件，只好把自动扩展参数打开，并且设置了每次扩展 20G，这里注意一下，如果设置过小会使很多会话发生 buffer busy waits 等待事件，但是设置这么大有个缺点就是如果 sql 语句出现笛卡儿积的话就会是表空间迅速暴涨，这里的这个例子就是这种情况下的一种。

第一步，首先查看了下 Users 表空间增长历史记录，具体截图如下，确定了 users 表空间增长的时间范围是在 6 月 25 号下午 14 点到 6 月 25 号晚上 23 点：

IN_DATE	增长大小G
2014-06-25 15:01:52	20
2014-06-25 16:21:53	60
2014-06-25 17:01:52	60
2014-06-25 18:31:53	80
2014-06-25 19:01:53	60
2014-06-25 20:41:52	180
2014-06-25 21:01:52	40
2014-06-25 22:51:52	240
2014-06-25 23:01:52	20
2014-06-26 00:01:52	120
2014-06-26 01:01:52	0

第二，从 6 月 25 号下午 14 点到晚上 23 点开始，查看了下具体段的增长情况，发现 users 表空间有一个 xxxxxx(这里屏蔽掉)用户下的临时段持续增长，涨了 859G，由临时段的名称看出都是一个段，且位于 4 号文件的 705052090 块，可以推断出是由于某一个错误 sql 导致的，而 4 号文件刚好就是 users 表空间，而临时段主要是由 2 种方式来生成：①重建索引生成②通过 CTAS 方式建表形成，重建索引不可能，因为没有哪个索引的大小达到 800G，所以只可能是哪个用户通过 CTAS 的方式建表导致的，而且在 23 点监控不到这个临时段了，可能表已经建成或者建表语句报错后临时段释放了。

大段的监控历史截图：

OWNER	SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE_NAME	sizes(G)
LIANGGRONG003	4.705052090	TEMPORARY	USERS	85
LIANGGRONG003	4.705052090	TEMPORARY	USERS	59
LIANGGRONG003	4.705052090	TEMPORARY	USERS	40
LIANGGRONG003	4.705052090	TEMPORARY	USERS	27
LIANGGRONG003	4.705052090	TEMPORARY	USERS	15

第三，仔细分析了下出现问题的时间段内 DDL 语句的监控，发现了一个错误记录，如下图，由此说明了是临时段达到了最大值 sql 语句报错了，所以空间释放了，这里我们可以看出当时的会话的 sid 是 1567，登录的 terminal 的 ip 地址为 10.31.6.61，具体同事是 xxxxxx (这里屏蔽掉)

ID	OPER_DATE	OPERATION
3789926	2014-06-26 00:01:05	SERVERERROR

OS_USER	CLIENT_IP	CLIENT_HOST	DB_SCHEMA	SID	SERIAL#	SPID	ERR
LIANGGRONG003	10.31.6.61	CNSH020001	LIANGGRONG003	1567	26035	23915	OR

第四，通过 sid 和 serial# 查看当时具体的 sql 监控，截图如下，由图看出该 sql 是从 25 号中午 11 点 35 分 30 秒开始运行，一直运行了 12 小时 17 分钟后报错，这个也和 users 表空间增长的时间范围相符

ID	SID	SERIAL#	SQL_ID	SQL_TEXT
444724	1567	26035	0wc0n5jc3wbt9	create table G_TX_DB_LABEL_base_4 NOLOGGING

SQL_EXEC_START	STATUS	COMMAND_TYPE	SQL_EXEC_ID	ELAPSED_TIME2	OS_USER
2014-06-25 11:35:30	DONE (ERROR)	CREATE TABLE	16777216	12时17分46.019秒	LIANGGRONG003

第五，把该 sql 拿出来看了下执行计划和 sql 语句,发现该执行计划的 cost 花费和预估的返回行数都超级大：

Sql 语句（这里只列出出现问题的地方）：

create table G_TX_DB_LABEL_base_4 NOLOGGING AS

SELECT

FROM G_TX_DB_LABEL_1 a

LEFT JOIN G_TX_DB_LABEL_2_comp b

ON a.单位名称 = a.单位名称

LEFT JOIN G_TX_DB_LABEL_2_comp_1 C

ON a.单位名称 = a.单位名称;

很显然，，，，，，， 连接条件写错了

Sql 执行计划，cost 和 rows 都非常的恐怖呀。。。。。。。。：

Plan hash value: 3002203763

Id	Operation	Name	Rows	Bytes	Cost (<CPU>)	Time
0	CREATE TABLE STATEMENT				85T<100>	
1	LOAD AS SELECT					
2	NESTED LOOPS OUTER		6683T	2196P	8096G	<1> 1999:59:59
3	NESTED LOOPS OUTER		251G	79T	346M	<1> 1999:59:59
4	TABLE ACCESS FULL	G_TX_DB_LABEL_1	339K	108M	3628	<1> 00:00:44
5	VIEW		740K	9406K	1020	<1> 00:00:13
* 6	FILTER					
7	TABLE ACCESS FULL	G_TX_DB_LABEL_2_COMP	740K	2894K	707	<1> 00:00:09
8	VIEW		26570	570K	32	<0> 00:00:01
* 9	FILTER					
10	TABLE ACCESS FULL	G_TX_DB_LABEL_2_COMP_1	26570	337K	18	<0> 00:00:01

```
.....
block_code,
.....
block_code_date,
memo,
TRUNC(SYSDATE) REPT_DATE
FROM G_TX_DB_LABEL_1 a
LEFT JOIN G_TX_DB_LABEL_2_comp b
ON a.单位名称 = a.单位名称
LEFT JOIN G_TX_DB_LABEL_2_comp_1 C
ON a.单位名称 = a.单位名称;
```

由此可以看出空间暴涨的原因是该 sql 最后的 3 张表的连接条件无效导致的，我把该 sql 拿出来重新执行了下发现短短 1 分钟内临时段涨了 2G 多，至此可以肯定导致 6 月 25 号空间暴涨的 sql 就是这个了

OWNER	SEGMENT_NAME	SEGMENT_TYPE	TABLESPACE_NAME	D.BYTES/1024/1024/1024
RISKNUMBER	4.672193946	TEMPORARY	USERS	0.001953125
RISKNUMBER	4.665678722	TEMPORARY	USERS	6.103515625E-5
RISKNUMBERSCJY	4.57150322	TEMPORARY	USERS	0.001953125
RISKNUMBER	4.70108818	TEMPORARY	USERS	0.0322265625
RISKDATA	7.442531498	TEMPORARY	RISKDATA	2.2529296875

最后，我给出的一些建议，建议充分利用一下我们的监控系统：

1. 加入笛卡儿积的监控，每隔 20 分钟监控一次
2. 增加对执行了 5 个小时以上的 sql 的监控
3. 增加对执行计划中预估的行数以及 cost 花费超大的 sql 的监控(例如本例中的 sql 语句)
4. 对统计信息有误的表的监控（如表实际有 200W 行，但是统计信息中的 num_rows 为 0，这种可能会出现笛卡儿积的连接）
5. 对数据库中的分区表全分区扫描的 sql 监控