

openguass-WDR 诊断报告

一,简介

WDR (Workload Diagnosis Report) 基于两次不同时间点系统的性能快照数据, 生成这两个时间点之间的性能表现报表, 用于诊断数据库内核的性能故障。

- WDR报告是长期性能问题最主要的诊断手段。基于SNAPSHOT的性能基线, 从多维度做性能分析, 能帮助DBA掌握系统负载繁忙程度、各个组件的性能表现及性能瓶颈。

- SNAPSHOT也是后续性能问题自诊断和自优化建议的重要数据来源。

二,WDR主要依赖两个组件:

SNAPSHOT性能快照: 性能快照可以配置成按一定时间间隔从内核采集一定量的性能数据, 持久化在用户表空间。任何一个SNAPSHOT可以作为一个性能基线, 其他SNAPSHOT与之比较的结果, 可以分析出与基线的性能表现。

WDR Reporter: 报表生成工具基于两个SNAPSHOT, 分析系统总体性能表现, 并能计算出更多项具体的性能指标在这两个时间段之间的变化量, 生成SUMMARY 和DETAIL两个不同级别的性能数据。

1.操作步骤

```
1 vi postgresql.conf
2 enable_wdr_snapshot=on
3 enable_resource_track=on
4 disable_memory_protect=off
5 gs_guc reload
```

2.启用WDR报告的[snapshot](#)收集

```
1 $ gs_guc reload -N all -I all -c "enable_wdr_snapshot=on"
2 postgres=# select name,setting from pg_settings where name like '%wdr%';
3           name           | setting
4 -----+-----
```

5	enable_wdr_snapshot	on	-- 开启数据库监控快照功能
6	wdr_snapshot_interval	60	-- 后台Snapshot线程执行监控快照的时间间隔，默认60分钟
7	wdr_snapshot_query_timeout	100	-- 快照操作相关的sql语句的执行超时时间
8	wdr_snapshot_retention_days	8	-- 系统中数据库监控快照数据的保留天数
9			

WDR Snapshot在启动后,会在用户表空间"pg_default", 数据库"postgres"下新建schema "snapshot", 用于持久化WDR快照数据。

⚠：WDR报告生成操作均在postgres数据库下操作进行

3.启用资源实时监控功能

enable_resource_track

参数说明：是否开启资源实时监控功能。

该参数属于SIGHUP类型参数，请参考[表1](#)中对应设置方法进行设置。

取值范围：布尔型

- on表示打开资源监控。
- off表示关闭资源监控。

默认值：on

```

1 postgres=# show enable_resource_track;
2 enable_resource_track
3 -----
4 on
5 (1 row)
6

```

4.控制系统中unique sql信息实时收集功能

⚠️：配置为0表示不启用unique sql信息收集功能，会导致导出的WDR报告SQL信息缺失

`instr_unique_sql_count`

参数说明：控制系统中unique sql信息实时收集功能。配置为0表示不启用unique sql信息收集功能。

该值由大变小将会清空系统中原有的数据重新统计；从小变大不受影响。

当系统中产生的unique sql信息大于 `instr_unique_sql_count` 时，系统产生的unique sql信息不被统计。

该参数属于SIGHUP类型参数，请参考[表1](#)中对应设置方法进行设置。

取值范围：整型，0~2147483647

5.启动内存保护功能

2.1.4 启动内存保护功能

`disable_memory_protect`

参数说明：禁止内存保护功能。当系统内存不足时如果需要查询系统视图，可以先将此参数置为on，禁止内存保护功能，保证视图可以正常查询。该参数只适用于在系统内存不足时进行系统诊断和调试，正常运行时请保持该参数配置为off。

该参数属于USERSET类型参数，且只对当前会话有效。请参考[重设参数](#)中方式三的方法进行设置。

取值范围：布尔型

- on表示禁止内存保护功能。
- off表示启动内存保护功能。

```
1 postgres=# show disable_memory_protect;
2 disable_memory_protect
3 -----
4 off
5 (1 row)
6
```

三,WDR信息表

1. snapshot.snapshot

记录当前系统中存储的WDR快照信息

```
1 postgres=# \d snapshot.snapshot
2          Table "snapshot.snapshot"
3   Column      |          Type          | Modifiers
```

```

4  -----+-----+-----
5  snapshot_id | bigint          | not null    -- WDR快照序列号
6  start_ts    | timestamp with time zone |            -- WDR快照的开始时间
7  end_ts      | timestamp with time zone |            -- WDR快照的结束时间
8  -----

```

2. snapshot.tables_snap_timestamp

记录所有表的WDR快照信息

```

1  postgres=# \d snapshot.tables_snap_timestamp
2          Table "snapshot.tables_snap_timestamp"
3  Column          |          Type          | Modifiers
4  -----+-----+-----
5  snapshot_id     | bigint                 | not null    -- WDR快照序列号
6  db_name         | text                  |            -- WDR snapshot对应的data
7  tablename       | text                  |            -- WDR snapshot对应的table
8  start_ts        | timestamp with time zone |            -- WDR快照的开始时间
9  end_ts          | timestamp with time zone |            -- WDR快照的结束时间
10

```

3. WDR数据表

WDR的数据表保存在snapshot这个schema下以snap_开头的表，其数据来源于dbe_perf这个schema内的视图

```

1  postgres=# select relname from pg_class where relname like '%snap_%';
2  -----
3  snapshot.tables_snap_timestamp    -- 记录所有存储的WDR快照中
4  snapshot.snapshot                 -- 记录当前系统中存储的WDR
5  snapshot.snapshot_pkey            -- snapshot.snapshot表的primary key
6  snapshot.snap_seq                 -- 序列
7  snapshot.snap_global_os_runtime   -- 操作系统运行状态信息

```

8	<code>snapshot.snap_global_os_threads</code>	-- 线程状态信息
9	<code>snapshot.snap_global_instance_time</code> 见instance_time视图)	-- 各种时间消耗信息(时间类型
10	<code>snapshot.snap_summary_workload_sql_count</code> 上的SQL数量分布	-- 各数据库主节点的workload
11	<code>snapshot.snap_summary_workload_sql_elapse_time</code> (业务) 负载的SQL耗时信息	-- 数据库主节点上workload
12	<code>snapshot.snap_global_workload_transaction</code> 信息	-- 各节点上的workload的负载
13	<code>snapshot.snap_summary_workload_transaction</code>	-- 汇总的负载事务信息
14	<code>snapshot.snap_global_thread_wait_status</code> 塞等待情况	-- 工作线程以及辅助线程的阻
15	<code>snapshot.snap_global_memory_node_detail</code>	-- 节点的内存使用情况
16	<code>snapshot.snap_global_shared_memory_detail</code>	-- 共享内存上下文的使用情况
17	<code>snapshot.snap_global_stat_db_cu</code> 通过gs_stat_reset()进行清零	-- 数据库的CU命中情况, 可以
18	<code>snapshot.snap_global_stat_database</code>	-- 数据库的统计信息
19	<code>snapshot.snap_summary_stat_database</code>	-- 汇总的数据库统计信息
20	<code>snapshot.snap_global_stat_database_conflicts</code>	-- 数据库冲突状态的统计信息
21	<code>snapshot.snap_summary_stat_database_conflicts</code> 计信息	-- 汇总的数据库冲突状态的统
22	<code>snapshot.snap_global_stat_bad_block</code> 信息	-- 表、索引等文件的读取失败
23	<code>snapshot.snap_summary_stat_bad_block</code> 取失败信息	-- 汇总的表、索引等文件的读
24	<code>snapshot.snap_global_file_redo_iostat</code>	-- Redo(WAL)相关统计信息
25	<code>snapshot.snap_summary_file_redo_iostat</code> 信息	-- 汇总的Redo(WAL)相关统计
26	<code>snapshot.snap_global_rel_iostat</code>	-- 数据对象IO统计信息
27	<code>snapshot.snap_summary_rel_iostat</code>	-- 汇总的数据对象IO统计信息
28	<code>snapshot.snap_global_file_iostat</code>	-- 数据文件IO统计信息
29	<code>snapshot.snap_summary_file_iostat</code>	-- 汇总的数据文件IO统计信息
30	<code>snapshot.snap_global_replication_slots</code>	-- 复制节点的信息
31	<code>snapshot.snap_global_bgwriter_stat</code>	-- 后端写进程活动的统计信息
32	<code>snapshot.snap_global_replication_stat</code>	-- 日志同步状态信息
33	<code>snapshot.snap_global_transactions_running_xacts</code>	-- 各节点运行事务的信息
34	<code>snapshot.snap_summary_transactions_running_xacts</code>	-- 汇总各节点运行事务的信息
35	<code>snapshot.snap_global_transactions_prepared_xacts</code> 的事务的信息	-- 当前准备好进行两阶段提交
36	<code>snapshot.snap_summary_transactions_prepared_xacts</code> 段提交的事务的信息	-- 汇总的当前准备好进行两阶
37	<code>snapshot.snap_summary_statement</code>	-- SQL语句的全量信息
38	<code>snapshot.snap_global_statement_count</code> L/DCL语句统计信息	-- 当前时刻执行的DML/DDDL/DQ

39	<code>snapshot.snap_summary_statement_count</code>	-- 汇总的当前时刻执行的DML/DDL/DQL/DCL语句统计信息
40	<code>snapshot.snap_global_config_settings</code>	-- 数据库运行时参数信息
41	<code>snapshot.snap_global_wait_events</code>	-- event等待相关统计信息
42	<code>snapshot.snap_summary_user_login</code>	-- 用户登录和退出次数的相关信息
43	<code>snapshot.snap_global_ckpt_status</code>	-- 实例的检查点信息和各类日志刷页情况
44	<code>snapshot.snap_global_double_write_status</code>	-- 实例的双写文件的情况
45	<code>snapshot.snap_global_pagewriter_status</code>	-- 实例的刷页信息和检查点信息
46	<code>snapshot.snap_global_redo_status</code>	-- 实例的日志回放情况
47	<code>snapshot.snap_global_rto_status</code>	-- 极致RTO状态信息
48	<code>snapshot.snap_global_recovery_status</code>	-- 主机和备机的日志流控信息
49	<code>snapshot.snap_global_threadpool_status</code>	-- 节点上的线程池中工作线程及会话的状态信息
50	<code>snapshot.snap_statement_responsetime_percentile</code>	-- SQL响应时间P80、P95分布信息
51	<code>snapshot.snap_global_statio_all_indexes</code>	-- 数据库中的每个索引行、显示特定索引的I/O的统计
52	<code>snapshot.snap_summary_statio_all_indexes</code>	-- 汇总的数据库中的每个索引行、显示特定索引的I/O的统计
53	<code>snapshot.snap_global_statio_all_sequences</code>	-- 数据库中每个序列的每一行、显示特定序列关于I/O的统计
54	<code>snapshot.snap_summary_statio_all_sequences</code>	-- 汇总的数据库中每个序列的每一行、显示特定序列关于I/O的统计
55	<code>snapshot.snap_global_statio_all_tables</code>	-- 数据库中每个表（包括TOAST表）的I/O的统计
56	<code>snapshot.snap_summary_statio_all_tables</code>	-- 汇总的数据库中每个表（包括TOAST表）的I/O的统计
57	<code>snapshot.snap_global_stat_all_indexes</code>	-- 数据库中的每个索引行，显示访问特定索引的统计
58	<code>snapshot.snap_summary_stat_all_indexes</code>	-- 汇总的数据库中的每个索引行，显示访问特定索引的统计
59	<code>snapshot.snap_summary_stat_user_functions</code>	-- 汇总的数据库节点用户自定义函数的相关统计信息
60	<code>snapshot.snap_global_stat_user_functions</code>	-- 用户所创建的函数的状态的统计信息
61	<code>snapshot.snap_global_stat_all_tables</code>	-- 每个表的一行（包括TOAST表）的统计信息
62	<code>snapshot.snap_summary_stat_all_tables</code>	-- 汇总的每个表的一行（包括TOAST表）的统计信息
63	<code>snapshot.snap_class_vital_info</code>	-- 校验相同的表或者索引的Oid是否一致

```

64 snapshot.snap_global_record_reset_time      -- 重置（重启，主备倒换，数
    据库删除）openGauss统计信息时间
65 snapshot.snap_summary_statio_indexes_name    -- 表snap_summary_statio_al
    l_indexes的索引
66 snapshot.snap_summary_statio_tables_name     -- 表snap_summary_statio_al
    l_tables的索引
67 snapshot.snap_summary_stat_indexes_name      -- 表snap_summary_stat_all_
    indexes的索引
68 snapshot.snap_class_info_name               -- 表snap_class_vital_info
    的索引
69 (66 rows)
70 -----
71

```

四, WDR报告创建

1. 创建snapshot

```

1  -- 当开启enable_wdr_snapshot参数时，数据库默认每小时自动执行一次snapshot操作。
2  -- 当然特定情况下，也可以手动使用函数创建snapshot，如：select create_wdr_snapsho
    t();
3  postgres=# select * from snapshot.snapshot;
4  snapshot_id |          start_ts          |          end_ts
5  -----+-----+-----
6           1 | 2022-07-12 05:59:09.337877+08 | 2022-07-12 05:59:10.249162+08
7           2 | 2022-07-12 06:59:10.3289+08   | 2022-07-12 06:59:11.229808+08
8           3 | 2022-07-12 07:59:10.426882+08 | 2022-07-12 07:59:11.348277+08
9           4 | 2022-07-12 08:59:10.534251+08 | 2022-07-12 08:59:11.447762+08
10          5 | 2022-07-12 09:59:11.448225+08 | 2022-07-12 09:59:26.121124+08
11

```

2. 查询数据库节点信息

```

1  postgres=# select * from pg_node_env;
2  node_name | host      | process | port | installpath | datapath
3  |          |          |      |            |
4  |          |          |      |            |          log_directory

```

五,创建WDR Report

1. 生成格式化性能报告文件

```
1 postgres=# \a \t \o /gaussdb/WDR_20220712.html
```

参数说明如下:

- \a: 切换非对齐模式。
- \t: 切换输出的字段名的信息和行计数脚注。
- \o: 把所有的查询结果发送至服务器文件里。
- 服务器文件路径: 生成性能报告文件存放路径。用户需要拥有此路径的读写权限。

2. 将查询到的信息写入性能报告中

```
1 postgres=# select generate_wdr_report(1,2,'all','node','node1');
```

函数说明: generate_wdr_report()

-- 语法

```
select generate_wdr_report(begin_snap_id bigint, end_snap_id bigint, report_type
cstring, report_scope cstring, node_name cstring);
```

-- 选项:

begin_snap_id: 查询时间段开始的snapshot的id (表snapshot.snaoshot中的snapshot id)

end_snap_id: 查询时间段结束snapshot的id。默认end_snap_id大于begin_snap_id (表snapshot.snaoshot中的snapshot id)

report_type: 指定生成report的类型。例如, summary/detail/all, 其中: summary[汇总数据]/detail[明细数据]/all[包含summary和detail]

report_scope: 指定生成report的范围, 可以为cluster或者node, 其中: cluster是数据库级别的信息, node是节点级别的信息。

node_name: 当report_scope指定为node时, 需要把该参数指定为对应节点的名称。当report_scope为cluster时, 该值可以省略或者指定为空或NULL。node[节点名称]、cluster[省略/空/NULL]

⚠: 用于生成报告的两个快照应满足以下条件:

- 两次快照之间不能有节点重启。
- 两次快照之间不能有主备倒换。
- 两次快照之间不能有drop database操作。