

特性LLT测试方案及验证结论

基本测试

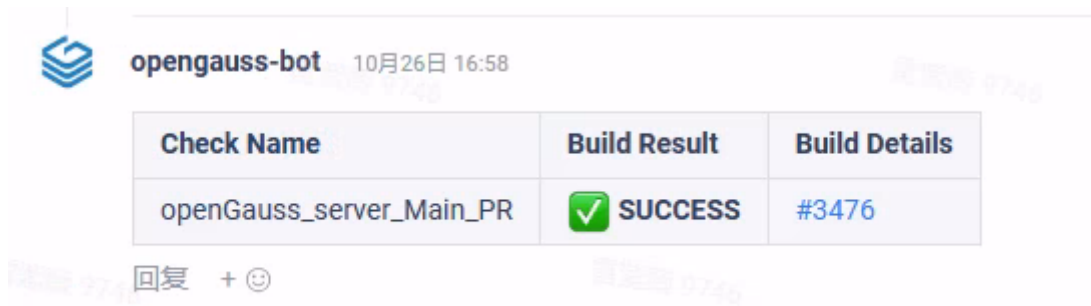
基本功能测试

回归测试验证

| 测试集 | 结论 |
|--------------------|------|
| fastcheck | 全部通过 |
| memcheck | 无影响 |
| hacheck | 无影响 |
| subscription/check | 全部通过 |

Fastcheck

有时候会碰到record_slow_sql_in_proc.sql报错，不过看到社区也有其他评审有这个问题，不知道是不是master本身的错。



| Check Name | Build Result | Build Details |
|--------------------------|--------------|---------------|
| openGauss_server_Main_PR | ✅ SUCCESS | #3476 |

Memcheck

目前发现master分支在测试sortgroup_agg.sql时挂库，所以注释掉了sortgroup_agg。

以下是master和加入继承表的分支的测试结构。虽然有错，但和master分支表现一致，判断为无内存泄漏。

```
Direct leak of 1608 byte(s) in 1 object(s) allocated from:
  #0 0xb02150 in __interceptor_malloc
  .././.././libsanitizer/asan/asan_malloc_linux.cc:62
  #1 0x4d6a3b3 in makeEmptyPGconn /opt/jenkins_slave/inherit/openGauss-main/src/common/interfaces/libpq/fe-connect.cpp:3339
  #2 0x4d5d071 in PQconnectStart /opt/jenkins_slave/inherit/openGauss-main/src/common/interfaces/libpq/fe-connect.cpp:847
  #3 0x4d5ae0e in PQconnectdb /opt/jenkins_slave/inherit/openGauss-main/src/common/interfaces/libpq/fe-connect.cpp:452
```

```
#4 0x31e4185 in AutonomousSession::AttachSession()
/opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/process/tpcop/autonomoustransaction.cpp:124
#5 0x31e4b79 in CreateAutonomousSession()
/opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/process/tpcop/autonomoustransaction.cpp:239
#6 0x4e33149 in plpgsql_exec_autonm_function(PLpgsql_function*,
FunctionCallInfoData*, char*) /opt/jenkins_slave/inherit/openGauss-
main/src/common/pl/plpgsql/src/pl_exec.cpp:1165
#7 0x4ddea0b in plpgsql_inline_handler /opt/jenkins_slave/inherit/openGauss-
main/src/common/pl/plpgsql/src/pl_handler.cpp:1172
#8 0x1c5f902 in OidFunctionCall1Coll(unsigned int, unsigned int, unsigned
long) /opt/jenkins_slave/inherit/openGauss-
main/src/common/backend/utills/fmgr/fmgr.cpp:1836
#9 0x2aace3d in ExecuteDoStmt(DoStmt*, bool)
/opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/optimizer/commands/functioncmds.cpp:3338
#10 0x4dd5a2b in InsertGsSource(unsigned int, unsigned int, char const*, char
const*, bool) /opt/jenkins_slave/inherit/openGauss-
main/src/common/pl/plpgsql/src/pl_handler.cpp:231
#11 0x4dd6034 in PkgInsertGsSource /opt/jenkins_slave/inherit/openGauss-
main/src/common/pl/plpgsql/src/pl_handler.cpp:263
#12 0x4de255a in plpgsql_package_validator
/opt/jenkins_slave/inherit/openGauss-
main/src/common/pl/plpgsql/src/pl_handler.cpp:1518
#13 0xc9c5ab in PackageSpecCreate(unsigned int, char const*, unsigned int,
char const*, bool, bool) /opt/jenkins_slave/inherit/openGauss-
main/src/common/backend/catalog/gs_package.cpp:485
#14 0x2b335db in CreatePackageCommand(CreatePackageStmt*, char const*)
/opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/optimizer/commands/packagecmds.cpp:147
#15 0x31af211 in standard_ProcessUtility(processutility_context*,
_DestReceiver*, bool, char*, ProcessUtilityContext, bool)
/opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/process/tpcop/utility.cpp:3394
#16 0x7f25f0cc8e90 in gsaudit_ProcessUtility_hook
/opt/jenkins_slave/inherit/openGauss-
main/contrib/security_plugin/gs_policy_plugin.cpp:819
#17 0x31da2cd in pgsaudit_ProcessUtility /opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/process/tpcop/auditfuncs.cpp:1419
#18 0x32c64f4 in hypo_utility_hook /opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/dbmind/kernel/hypopg_index.cpp:167
#19 0x31a71c2 in ProcessUtility(processutility_context*, _DestReceiver*,
bool, char*, ProcessUtilityContext, bool) /opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/process/tpcop/utility.cpp:1564
#20 0x319aa1b in PortalRunUtility /opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/process/tpcop/pquery.cpp:1736
#21 0x319b70f in PortalRunMulti /opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/process/tpcop/pquery.cpp:1915
#22 0x3196d7c in PortalRun(PortalData*, long, bool, _DestReceiver*,
_DestReceiver*, char*) /opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/process/tpcop/pquery.cpp:1191
#23 0x315ca01 in exec_simple_query /opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/process/tpcop/postgres.cpp:3000
```

```

#24 0x317b2f7 in PostgresMain(int, char**, char const*, char const*)
/opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/process/tpop/postgres.cpp:9096
#25 0x2fb0f2b in BackendRun /opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/process/postmaster/postmaster.cpp:9404
#26 0x2fd2ffe in int GaussDbThreadMain<(kn1_thread_role)2>(kn1_thread_arg*)
/opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/process/postmaster/postmaster.cpp:13919
#27 0x2fcad94 in InternalThreadFunc /opt/jenkins_slave/inherit/openGauss-
main/src/gausskernel/process/postmaster/postmaster.cpp:14550
#28 0x4dc1d15 in ThreadStarterFunc /opt/jenkins_slave/inherit/openGauss-
main/src/common/port/g_s_thread.cpp:380
#29 0x7f26017eafec (/lib64/libpthread.so.0+0x8fec)

```

hacheck

sh run_ha_multi_single.sh

```

[omm@arm64-openeuler2003-docker openGauss-main]$ cd src/test/ha
[omm@arm64-openeuler2003-docker ha]$ pwd
/opt/jenkins_slave/inherit/openGauss-main/src/test/ha
[omm@arm64-openeuler2003-docker ha]$ vim standby_env.sh
[omm@arm64-openeuler2003-docker ha]$ sh run_ha_multi_single.sh && sh
init and start the database
run the ha_schedule 1 time
testcase                result    time(s)
multi_standby_single/inc_build_failover    .... ok   167
multi_standby_single/inc_build_failover_twice .... ok   170
multi_standby_single/inc_build_reconnect   .... ok   167
multi_standby_single/switchover           .... ok   237
multi_standby_single/switchover_timeout   .... ok   392
multi_standby_single/failover              .... ok   225
multi_standby_single/params                .... ok   356
multi_standby_single/xlog_redo_apply_delay .... ok    10
multi_standby_single/failover_with_data    .... ok   263
multi_standby_single/hash_index            .... ok   436
multi_standby_single/consistency.sh        .... ok    0
stop the database
all 11 tests passed.
total time: 2767s

```

sh run_ha_single.sh (master分支就无法通过, dummystandby编译时可能需要别的配置, 失败原因和继承表无关)

```

[omm@arm64-openeuler2003-docker ha]$ sh run_ha_single.sh
run the ha_schedule 1 time
testcase                result    time(s)
data_replication_single/datareplica_cstore .... FAILED67
[omm@arm64-openeuler2003-docker ha]$ vim results/data_replication_single/datareplica_cstore.log

```

subscription/check

```
[omm@arm64-openeuler2003-docker subscription]$ cd ../../../../
[omm@arm64-openeuler2003-docker openGauss-main]$ source profile_tmp
[omm@arm64-openeuler2003-docker openGauss-main]$ cd src/test/subscription
[omm@arm64-openeuler2003-docker subscription]$ make check p=26170
sh /opt/jenkins_slave/inherit/openGauss-main/src/test/subscription/
removing /opt/jenkins_slave/inherit/openGauss-main/src/test/subscription/
removing /opt/jenkins_slave/inherit/openGauss-main/src/test/subscription/
init and start the database
testcase                                result    time(s)
rep_changes                             .... ok   27
pub_switchover                           .... ok   15
types                                    .... ok   13
constraints                              .... ok   10
binary                                    .... ok   15
diff_schema                              .... ok   12
generated                                 .... ok   10
rewrite                                   .... ok   10
sync                                       .... ok   17
encoding                                  .... ok    5
ddl                                        .... ok    5
matviews                                  .... ok    4
change_wal_level                          .... ok   21
skiplsn                                    .... ok   20
disable                                    .... ok    5
pub_subconflict                           .... ok   50
bugs                                       .... ok   30
stop the database
all 17 tests passed.
total time: 316s
[omm@arm64-openeuler2003-docker subscription]$
```

升级验证

| 验证集 | 结论 |
|------------------------------|----|
| openGauss LTS版本升级到最新master | 通过 |
| 社区最新master回滚到旧版本 | 通过 |
| openGauss LTS版本再升级到最新master | 通过 |
| openGauss LTS版本升级提交到最新master | 通过 |

openGauss LTS版本升级到最新master

由5.0.0升级到5.1.1:

```

[cm6@KylinMaster script]$ gs_upgradectl -t auto-upgrade -X /home/cluster2.xml
Static configuration matched with old static configuration files.
Performing inplace rollback.
Rollback succeeded.
Checking upgrade environment.
Successfully checked upgrade environment.
Successfully started cluster.
Start to do health check.
Successfully checked cluster status.
Backing up current application and configurations.
Successfully backed up current application and configurations.
Backing up cluster configuration.
Successfully backup hotpatch config file.
Successfully backed up cluster configuration.
Installing new binary.
Restoring cluster configuration.
Successfully restored cluster configuration.
Successfully started cluster.
Start check CMS parameter.
Modifying the socket path.
Successfully modified socket path.
Successfully started cluster.
copy certs from /opt/cm6/install/app_a07d57c3 to /opt/cm6/install/app_90995c0a.
Successfully copy certs from /opt/cm6/install/app_a07d57c3 to /opt/cm6/install/app_90995c0a.
Switch symbolic link to new binary directory.
Successfully switch symbolic link to new binary directory.
Successfully started cluster.
Successfully started cluster.
Waiting for the cluster status to become normal.
.
The cluster status is normal.
Start to do health check.
Successfully checked cluster status.
Upgrade main process has been finished, user can do some check now.
Once the check done, please execute following command to commit upgrade:

gs_upgradectl -t commit-upgrade -X /home/cluster2.xml

```

openGauss最新master回滚到旧版本

```

[cm6@KylinMaster script]$ gs_upgradectl -t auto-rollback -X /home/cluster2.xml
Static configuration matched with old static configuration files.
Performing inplace rollback.
Checking static configuration files.
Successfully checked static configuration files.
Successfully started cluster.
Restoring cluster configuration.
Successfully rollback hotpatch config file.
Successfully restored cluster configuration.
Start roll back CM instance.
Switch symbolic link to old binary directory.
Successfully switch symbolic link to old binary directory.
Successfully started cluster.
Restoring application and configurations.
Successfully restored application and configuration.
Restoring cluster configuration.
Successfully rollback hotpatch config file.
Successfully restored cluster configuration.
Clean up backup catalog files.
Successfully started cluster.
Successfully cleaned new install path.
Rollback succeeded.

```

openGauss LTS版本再升级到最新master

预安装:

```
Creating SSH trust.
Creating the local key file.
Successfully created the local key files.
Appending local ID to authorized_keys.
Successfully appended local ID to authorized_keys.
Updating the known_hosts file.
Successfully updated the known_hosts file.
Appending authorized_key on the remote node.
Successfully appended authorized key on all remote node.
Checking common authentication file content.
Successfully checked common authentication content.
Distributing SSH trust file to all node.
Distributing trust keys file to all node successfully.
Successfully distributed SSH trust file to all node.
Verifying SSH trust on all hosts.
Successfully verified SSH trust on all hosts.
Successfully created SSH trust.
Successfully created SSH trust for [cm6] user.
Checking OS software.
Successfully check os software.
Checking OS version.
Successfully checked OS version.
Creating cluster's path.
Successfully created cluster's path.
Set and check OS parameter.
Setting OS parameters.
Successfully set OS parameters.
Warning: Installation environment contains some warning messages.
Please get more details by "/home/gauss_update/script/gs_checkos -i A -h KylinMaster
Set and check OS parameter completed.
Preparing CRON service.
Successfully prepared CRON service.
Setting user environmental variables.
Successfully set user environmental variables.
Setting the dynamic link library.
Successfully set the dynamic link library.
Setting Core file
Successfully set core path.
Setting pssh path
Successfully set pssh path.
Setting Cgroup.
Successfully set Cgroup.
Set ARM Optimization.
Successfully set ARM Optimization.
Fixing server package owner.
Setting finish flag.
Successfully set finish flag.
Preinstallation succeeded.
```

再升级:


```
[cm6@KylinMaster script]$ gs_upgradectl -t auto-upgrade -X /home/cluster2.xml
Static configuration matched with old static configuration files.
Performing inplace rollback.
Rollback succeeded.
Checking upgrade environment.
Successfully checked upgrade environment.
Successfully started cluster.
Start to do health check.
Successfully checked cluster status.
Backing up current application and configurations.
Successfully backed up current application and configurations.
Backing up cluster configuration.
Successfully backup hotpatch config file.
Successfully backed up cluster configuration.
Installing new binary.
Restoring cluster configuration.
Successfully restored cluster configuration.
Successfully started cluster.
Start check CMS parameter.
Modifying the socket path.
Successfully modified socket path.
Successfully started cluster.
copy certs from /opt/cm6/install/app_a07d57c3 to /opt/cm6/install/app_90995c0a.
Successfully copy certs from /opt/cm6/install/app_a07d57c3 to /opt/cm6/install/app_90995
Switch symbolic link to new binary directory.
Successfully switch symbolic link to new binary directory.
Successfully started cluster.
Successfully started cluster.
Waiting for the cluster status to become normal.
.
The cluster status is normal.
Start to do health check.
Successfully checked cluster status.
Upgrade main process has been finished, user can do some check now.
Once the check done, please execute following command to commit upgrade:

    gs_upgradectl -t commit-upgrade -X /home/cluster2.xml
```

openGauss LTS版本升级提交到最新master

提交升级:

```
[cm6@KylinMaster script]$ gs_upgradectl -t commit-upgrade -X /home/cluster2.xml
NOTICE: Start to commit binary upgrade.
Start to check whether can be committed.
Can be committed.
Start to set commit flag.
Set commit flag succeeded.
Start to do operations that cannot be rollback.
Cancel the upgrade status succeeded.
Start to clean temp files for upgrade.
Clean up backup catalog files.
Successfully cleaned old install path.
Successfully started cluster.
Clean temp files for upgrade succeeded.
NOTICE: Commit binary upgrade succeeded.
```

代码检视结论

编码规范检查

编程规范

无编码不合规处

内存使用排查

处于同一上下文

```
Query* query_tree_mutator(Query* query, Node* (*mutator)(Node*, void*), void* context)
{
    Assert(query != NULL && IsA(query, Query));

    if (!(flags & QTW_DONT_COPY_QUERY)) {
        Query* newquery = NULL;

        FLATCOPY(newquery, query, Query, true);
        if(newquery->resultRelations)
            newquery->resultRelations = (List*)copyObject(query->resultRelations);
        query = newquery;
    }
}
```

覆盖率

LCOV - code coverage report

| Directory | Line Coverage | Functions |
|---|------------------|-----------------|
| /opt/jenkins_slave/workspace/review_open6auss_coverage/src/common/backend/nodes | 100.0 % (2 / 2) | 100.0 % (1 / 1) |
| optimizer/commands | 100.0 % (2 / 2) | 100.0 % (1 / 1) |
| optimizer/paths | 100.0 % (2 / 2) | 100.0 % (1 / 1) |
| optimizer/plan | 100.0 % (1 / 1) | 100.0 % (1 / 1) |
| optimizer/psps | 100.0 % (1 / 1) | 100.0 % (1 / 1) |
| runtime/executor | 88.2 % (45 / 51) | 83.3 % (5 / 6) |

Summary: Lines: 53 / 59 (89.8%), Functions: 10 / 11 (90.9%)

Generated by LCOV version 1.14