openGauss SQL兼容性需求check-in评审报告

评审纪要

时间：2022年9月6日
特性名称：openGauss兼容MySQL日期处理函数
特性责任人：
评审人：

设计checklist

模块	类别	检查项	检查项详细说明	是否涉及	是否完成检查
GUC参数	功能	新增GUC参数适配项	①guc.cpp里需要新增guc参数的相应代码，包括参数的校验/赋值等功能函数； ②如果要加入postgres.conf，需要修改postgresql_single.conf.sample 路径：src/common/backend/utils/misc/ 一般以注释方式加入，如果非常确认使用某个默认值，也可以以非注释方式加入 ③如果要允许gs_guc工具设置，请修改cluster_guc.conf 路径：src/bin/gs_guc/cluster_guc.conf	是	是
升级	兼容性	版本号变更	src/common/backend/utils/init/globals.cpp：GRAND_VERSION_NUM	否	
升级	兼容性	前向兼容性	1）涉及系统表修改的特性，必须提供系统表升级脚本和回滚脚本，修改版本号文件 2）对于涉及修改持久化数据（如日志）的特性，必须考虑新老版本共存时的兼容性场景，必要情况下，需要增加版本号以进行识别 3）对于涉及修改执行态数据格式（如syscache结构）的特性，必须考虑新老版本共存时的兼容性场景，必要情况下，需要增加版本号以进行识别	否	
升级	兼容性	验证升级正常	涉及到升级时需要验证如下场景： 1. 集群环境安装：该步骤会验证集群环境安装是否正常 2. 集群环境升级：从老的版本（1.1.0版本）升级到最新版本是否正常	否	
公共数据结构	功能	hashtable	hashtable使用注意事项 ①在使用hash表时，entry必须封装成一个结构体，第一个成员必须是hash key 否则会造成查找失败！！ ②封装的结构体必须只有两个成员：key+value，即value需要封装起来 否则hash表在search的时候会core！！ ③hash表在创建的时候，entry 的size一定要正确 否则在enter的时候会造成写越界，不会报错但是其它变量内存会被踩，不易发现！！	否	
公共数据结构	功能	禁止使用commandTag判断Query的类型	需要对SELECT\INSERT\UPDATE\DELETE\MERGE INTO等query类型进行识别判断时，应该使用CmdType进行判断或新增标识。 禁止使用commandTag作为判断标识，因为commandTag在带returning*语句执行中变化。	否	
系统函数	兼容性	系统表升级是否正确	通过对比initdb和升级后的db中pg_proc每一列的值，避免出现升级和非升级不一致的情况	否	

模块	类别	检查项	检查项详细说明	是否涉及	是否完成检查
系统函数	兼容性	新增数据类型适配FENCED函数	新增数据类型如果是FENCED函数使用，需要修改UDFArgumentHandler函数，把数据类型添加到对应收发分支。	否	
系统函数	功能	NULL值处理	对于非strict函数允许接收NULL入参，在函数逻辑中应对NULL值进行细致的处理	是	是
前向兼容性	兼容性	Node相关数据结构删减	禁止对Node相关数据结构（如Query、RangeTblEntry）数据成员删减	否	
兼容性	兼容性	兼容性设计原则	目前支持兼容性PG,A,B,C;对于每一个兼容点要求：1、如果该兼容点为非冲突兼容点，应该是各个兼容模式下均支持，不需要开关控制 2、如果该兼容性点为冲突兼容点，应该由兼容开关控制。	是	是
查询解析	兼容性	添加关键字场景	为了尽可能的防止移进规约冲突，最大化的用关键字作为对象名，关键字分有4类：非保留关键字、列名关键字、类型函数关键字、保留关键字。他们的约束逐渐加强，比如保留关键字不可以用做表名。添加关键字需要考虑如下： 1. 添加新的关键字最好先作为非保留关键字添加 2. 如果要加其他关键字类型那么需要考虑是否有前向兼容问题。比如排查客户场景是否有用该关键字作为对象名。 3. 关键字还需要加入kwlist.h文件中。	是	是
查询解析	兼容性	Gram.y中处理语法冲突	1、合理使用a_expr/b_expr/c_expr使用场景 为了尽可能防止冲突，划分了a_expr/b_expr/c_expr，比如BETWEEN a_expr AND b_expr中 AND属于a_expr范畴，b_expr中不含AND。否则就无法区分BETWEEN之后是Bool判断还是范围。c_expr是a_expr和b_expr的交集。所以 尽量用a_expr这样支持的范围最广。如果用b_expr/c_expr 那需要考虑潜在的用户问题场景（参考BETWEEN...AND...例子）并在手册中写清楚约束。	是	是
其他	功能	新增已有的数据结构成员适配序列化与反序列化	注意适配： src/common/backend/nodes/copyfuncs.cpp src/common/backend/nodes/equalfuncs.cpp src/common/backend/nodes/outfuncs.cpp src/common/backend/nodes/readfuncs.cpp	否	

模块	类别	检查项	检查项详细说明	是否涉及	是否完成检查
公共子系统	内存定位定界	局部变量内存释放，非程序结尾位置，释放进行置空；全局内存变量释放，无论什么位置均需要置空	防止内存使用出现use-after-free模式，内存越界	否	
公共子系统	内存定位定界	对于使用数组、下标访问内存操作场景，是否提前判断下标符合内存空间要求	防止出现heap overflow场景	否	
公共子系统	内存定位定界	内存分配报错日志中，是否需要新增信息，便于分析报错原因	内存报错场景下，快速定位内存报错原因	否	
资源负载管理	兼容性	只能新增系统表参数，不可以删除已有的参数	前向兼容	否	
资源负载管理	功能	新增系统表参数的默认初始化数值要合理，不影响已有的功能	前向兼容	否	
HA用例	功能	ha check	1.HA相关设计实现需要严格系统的测试。 2.需要保证最简单的功能验证，包括搭建主备环境，确认主备可以正常同步，以及跑ha check用例，确认所有用例可以跑过。 3.需要完成各种故障场景的验证，设计详细的故障库，确保覆盖各个场景。	否	
安全	未公开接口	是否新增或修改函数、视图、系统表	新增或修改函数、视图、系统表需刷新产品文档，考虑权限控制	是	是

模块	类别	检查项	检查项详细说明	是否涉及	是否完成检查
安全	未公开接口	是否新增SQL语法	新增SQL语法需刷新产品文档，支持记录审计日志	否	

特性LLT测试方案及验证结论

基本测试

```
my_db=# select dayname('2000-1-1');
 dayname
----------
 Saturday
(1 row)

my_db=# select monthname('2000-1-1');
 monthname
-----------
 January
(1 row)

my_db=# select time_to_sec('12:12:12');
 time_to_sec
-------------
       43932
(1 row)

my_db=# select month('2000-1-1');
 month
-------
     1
(1 row)

my_db=# select day('2000-1-1');
 day
-----
   1
(1 row)

my_db=# select date('2000-1-1 12:12:12');
    date
------------
 2000-01-01
(1 row)

my_db=# set b_compatibility_mode = true;
SET
my_db=# select last_day('2000-1-1');
  last_day
------------
 2000-01-31
(1 row)

my_db=# select week('2000-1-1');
 week
------
    0
(1 row)

my_db=# select yearweek('2000-1-1');
 yearweek
----------
   199952
(1 row)
```

回归测试验证

测试集	结论
fastcheck	通过
memcheck	通过
hacheck	

Fastcheck

```
============== shutting down postmaster
stop postmaster now!

======================
 All 195 tests passed.
 Total Time: 506.760446s
======================
```

Memcheck

本地代码

```
============== shutting down postmaster
stop postmaster now!

======================
 All 195 tests passed.
 Total Time: 1715.628318s
======================
```

```
 229 Sep 17 12:06 runlog.1148043
8.5K Sep 17 11:40 runlog.1154036
7.9K Sep 17 12:00 runlog.1167558
```

官方代码

```
 229 Sep 17 04:34 runlog.935227
8.5K Sep 17 04:08 runlog.941424
7.9K Sep 17 04:28 runlog.950276
```

内容与官方一致

```
    #0 0xa1bc33 in __interceptor_memcpy ../../.././libsanitizer/
    #1 0x520e1ef in memcpy_s (/home/atwww/sda/openGauss-server/
    #2 0x3ebd3ff in heap_fill_tuple(tupleDesc*, unsigned long*,
rver/src/gausskernel/storage/access/common/heaptuple.cpp:240
    #3 0x3ec5f93 in heap_form_tuple(tupleDesc*, unsigned long*,
    #4 0x2deaf33 in CreateTrigger(CreateTrigStmt*, char const*,
rc/gausskernel/optimizer/commands/trigger.cpp:800
```

hacheck

代码检视结论

编码规范检查

编程规范

请对照社区安全编码规范进行排查：

https://gitee.com/opengauss/security/blob/master/guide/SecureCoding.md

https://gitee.com/opengauss/security/blob/master/guide/SecureCompile(C&C++).md

内存使用排查

覆盖率89%

LCOV - code coverage report

			Hit	Total	Coverage
Current view:	top level	**Lines:**	470	528	89.0 %
Test:	increment.info	**Functions:**	189	659	28.7 %
Date:	2022-09-17 13:52:38	**Branches:**	0	0	-

Directory	Line Coverage		Functions		Branches	
/home/atwww/sda/openGauss-server/contrib/dolphin	81.8 %	18 / 22	84.2 %	16 / 19	-	0 / 0
/home/atwww/sda/openGauss-server/contrib/dolphin/plugin_parser	96.2 %	25 / 26	30.8 %	28 / 91	-	0 / 0
/home/atwww/sda/openGauss-server/contrib/dolphin/plugin_utils/adt	89.0 %	427 / 480	26.4 %	145 / 549	-	0 / 0

Generated by: LCOV version 1.14

鲲鹏平台乱序排查

编号	排查依据	排查结果
1	**是否为多线程并发场景?**	否
2	**是否涉及线程间共享数据?**	否
3	**是否未对共享数据加锁保护?**	否
4	线程间共享数据是否存在无锁同步模式?	否
5	是否 **未** 在正确的位置插入合适的内存屏障	否

代码检视意见

提出人	意见	答复详情
pengjiong	contrib/dolphin/include/plugin_utils/datetime.h 宏用全大写	已修改
pengjiong	contrib/dolphin/plugin_utils/adt/date.cpp 将9999使用B_FORMAT_MAX_YEAR_OF_DATE替换	已修改
pengjiong	contrib/dolphin/plugin_utils/adt/date.cpp strtol可以处理前置空格，无需跳过	已删除此函数，将字符串用openGauss原有解析过程解析
pengjiong	contrib/dolphin/plugin_utils/adt/date.cpp " 0 " 后面也有空格的场景会返回false，符合预期吗?	已修改，用原有解析过程解析
pengjiong	contrib/dolphin/plugin_utils/adt/date.cpp timestamp.cpp 里面有很多相关的宏，弄到头文件里面一起用吧，去掉魔鬼数字，其他的都一起整改下 DAYS_PER_COMMON_YEAR/DAYS_PER_LEAP_YEAR/DAYS_PER_WEEK	已修改
penjiong	contrib/dolphin/plugin_utils/adt/date.cpp NumericVar2lldiv就这一个地方用，返回bool也不判断，如果返回值没用的话改成void	已修改返回值为void
pengjiong	contrib/dolphin/plugin_utils/adt/date.cpp 注释换成英文的	已修改，已删除此处无用注释
pengjiong	contrib/dolphin/plugin_utils/adt/datetime.cpp 这里if里面的 hasD 赋值没有意义，后面不会再用到 hasD 了	已删除此函数，复用原有解析函数。
pengjiong	contrib/dolphin/include/plugin_utils/datetime.h 这里好几个 b_db_func_xx 的函数，和原始函数差异小于10%，请尝试提取公共函数或者增加入参的方式复用原有逻辑	已通过增加入参的方式复用原有逻辑。
pengjiong	contrib/dolphin/plugin_utils/adt/datetime.cpp 这里的魔鬼数字都统一处理下	已将有意义的魔鬼数字进行了处理。
chenxiaobin	contrib/dolphin/plugin_utils/adt/date.cpp 这里可以直接将宏b_db_enable_zero_day传进来，没有搞一个局部变量的必要，其他地方类似	已修改。
chenxiaobin	contrib/dolphin/plugin_utils/adt/date.cpp 已修改	已修改。
chenxiaobin	contrib/dolphin/plugin_utils/adt/date.cpp 函数名改一下吧，类似b_db_is_date_zero	已修改
王修强	contrib/dolphin/include/plugin_utils/datetime.h 这里的函数命名为何是下划线和驼峰混合命名的?	这里的函数与原有函数差距小，已删除，通过增加入参的方式复用原有逻辑。
王修强	contrib/dolphin/plugin_postgres.cpp 缺少static关键字	已增加

遗留问题

测试建议