**PostGIS概述**

openGauss提供PostGIS Extension（版本为PostGIS-2.4.2）。PostGIS Extension是PostgreSQL的空间数据库扩展，提供如下空间信息服务功能：空间对象、空间索引、空间操作函数和空间操作符。PostGIS Extension完全遵循OpenGIS规范。

GaussDB A中PostGIS Extension依赖第三方开源软件如下。依赖的第三方软件需要用户进行单独安装，用户如需使用PostGIS功能，请提交工单或联系技术支持人员提交申请。依赖的第三方软件需要用户进行单独安装，具体请参考PostGIS安装。

* Geos 3.6.2
* Proj 4.9.2
* Json 0.12.1
* Libxml2 2.7.1
* Gdal 1.11.0

# PostGIS安装

GaussDB A数据库的PostGIS Extension源码包可通过网站<https://opengauss.obs.cn-south-1.myhuaweicloud.com/dependency/postgis-xc-master-2020-09-17.tar.gz>获取。该Extension需使用GCC-7.3（GNU编译器套件）进行编译安装。

* 在整个安装过程中，可使用make -sj和make install -sj命令并行加速编译，-sj命令极低概率性出现安装错误，如果安装失败则请使用make和make install进行串行安装。
* 对于ARM物理机，在configure时增加如下编译参数：--build=aarch64-unknown-linux-gnu。

#### ****操作步骤****

1. GCC-7.3（GNU编译器套件）编译器安装。

PostGIS安装依赖GCC-7.3编译器。GCC-7.3编译器推荐使用源码由低版本的gcc和g++编译器进行升级安装。若集群中没有低版本gcc和g++编译器，可以通过挂载操作系统镜像等方法进行安装，这里不做赘述。GCC-7.3编译器相关依赖软件包，包括gcc-7.3.0、gmp-4.3.2、mpfr-2.4.2、mpc-1.0.3，安装包下载地址分别为：

<https://ftp.gnu.org/gnu/gcc/gcc-7.3.0/gcc-7.3.0.tar.gz>

<https://ftp.gnu.org/gnu/gmp/gmp-4.3.2.tar.gz>

<https://ftp.gnu.org/gnu/mpfr/mpfr-2.4.2.tar.gz>

<https://ftp.gnu.org/gnu/mpc/mpc-1.0.3.tar.gz>

* 1. 以操作系统用户omm登录安装有MPPDB服务的任一主机。执行source ${BIGDATA\_HOME}/mppdb/.mppdbgs\_profile命令启动环境变量。
  2. 创建GCC安装主目录$GAUSSHOME/gcc和代码下载目录$GAUSSHOME/gcc/packages，并下载软件包gcc-7.3.0.tar.gz、gmp-4.3.2.tar.gz、mpc-1.0.3.tar.gz、mpfr-2.4.2.tar.gz至$GAUSSHOME/gcc/packages目录。
  3. mkdir $GAUSSHOME/gcc

mkdir $GAUSSHOME/gcc/packages

* 1. 解压下载软件。
  2. cd $GAUSSHOME/gcc/packages
  3. tar -xzf gcc-7.3.0.tar.gz
  4. tar -xzf gmp-4.3.2.tar.gz
  5. tar -xzf mpc-1.0.3.tar.gz

tar -xzf mpfr-2.4.2.tar.gz

* 1. 创建GCC安装目录。
  2. mkdir $GAUSSHOME/gcc/gcc-7.3.0
  3. mkdir $GAUSSHOME/gcc/gcc-7.3.0/depend
  4. mkdir $GAUSSHOME/gcc/gcc-7.3.0/depend/gmp-4.3.2
  5. mkdir $GAUSSHOME/gcc/gcc-7.3.0/depend/mpfr-2.4.2
  6. mkdir $GAUSSHOME/gcc/gcc-7.3.0/depend/mpc-1.0.3

mkdir $GAUSSHOME/gcc/gcc-7.3.0/depend/gcc

* 1. 安装gmp-4.3.2。

进入$GAUSSHOME/gcc/packages/gmp-4.3.2目录，执行下列命令完成gmp安装操作:

cd $GAUSSHOME/gcc/packages/gmp-4.3.2

./configure --prefix $GAUSSHOME/gcc/gcc-7.3.0/depend/gmp-4.3.2

make -sj

make install -sj

* 1. 安装mpfr-2.4.2。

进入$GAUSSHOME/gcc/packages/mpfr-2.4.2目录，执行以下命令完成mpfr安装操作:

cd $GAUSSHOME/gcc/packages/mpfr-2.4.2

./configure --prefix $GAUSSHOME/gcc/gcc-7.3.0/depend/mpfr-2.4.2 --with-gmp=$GAUSSHOME/gcc/gcc-7.3.0/depend/gmp-4.3.2

make -sj

make install -sj

* 1. 安装mpc-1.0.3。

进入$GAUSSHOME/gcc/packages/mpc-1.0.3目录，执行下列命令完成mpc安装操作:

cd $GAUSSHOME/gcc/packages/mpc-1.0.3

./configure --prefix=$GAUSSHOME/gcc/gcc-7.3.0/depend/mpc-1.0.3 --with-gmp=$GAUSSHOME/gcc/gcc-7.3.0/depend/gmp-4.3.2 --with-mpfr=$GAUSSHOME/gcc/gcc-7.3.0/depend/mpfr-2.4.2

make -sj

make install -sj

* 1. 安装gcc-7.3.0。

a) 执行如下命令添加环境变量。

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$GAUSSHOME/gcc/gcc-7.3.0/depend/gmp-4.3.2/lib:$GAUSSHOME/gcc/gcc-7.3.0/depend/mpfr-2.4.2/lib:$GAUSSHOME/gcc/gcc-7.3.0/depend/mpc-1.0.3/lib

b)进入$GAUSSHOME/gcc/packages/gcc-7.3.0目录，执行下列命令完成gcc安装操作。

cd $GAUSSHOME/gcc/packages/gcc-7.3.0

./configure --prefix=$GAUSSHOME/gcc/gcc-7.3.0/depend/gcc -disable-multilib --with-gmp=$GAUSSHOME/gcc/gcc-7.3.0/depend/gmp-4.3.2 -enable-languages=c,c++ --with-mpfr=$GAUSSHOME/gcc/gcc-7.3.0/depend/mpfr-2.4.2 --with-mpc=$GAUSSHOME/gcc/gcc-7.3.0/depend/mpc-1.0.3

make -sj

make install -sj

c) 执行如下命令添加环境变量。

export CC=$GAUSSHOME/gcc/gcc-7.3.0/depend/gcc/bin/gcc

export CXX=$GAUSSHOME/gcc/gcc-7.3.0/depend/gcc/bin/g++

export LD\_LIBRARY\_PATH=$GAUSSHOME/gcc/gcc-7.3.0/depend/gcc/lib64:$LD\_LIBRARY\_PATH

export PATH=$GAUSSHOME/gcc/gcc-7.3.0/depend/gcc/bin:$PATH

1. zlib安装。

Libxml2软件包编译安装依赖zlib无损数据压缩库。omm用户可执行find /usr/ -name libz.a查看zlib是否安装(如果libz.a存在则zlib已安装)。

若zlib未安装，可从网站<https://sourceforge.net/projects/libpng/files/zlib/1.2.8/zlib-1.2.8.tar.gz/download>下载，并以root用户进入代码目录并执行如下命令进行安装，并再次切换回omm用户

./configure

make -sj

make install -sj

su - omm

source ${BIGDATA\_HOME}/mppdb/.mppdbgs\_profile

安装成功后，可以在/usr/local/lib下找到libz.a。

1. autoconf和automake安装。JSON-C软件包编译安装依赖autoconf和automake工具。若集群中没有autoconf和automake工具，可以通过挂载操作系统镜像等方法进行安装，这里不做赘述。
2. PostGIS依赖库安装。
   1. 从网站<https://opengauss.obs.cn-south-1.myhuaweicloud.com/dependency/postgis-xc-master-2020-09-17.tar.gz>获取PostGIS源码至$GAUSSHOME目录，下载压缩包，解压后需将文件夹重命名为postgis-xc。
   2. cd $GAUSSHOME
   3. 以操作系统用户omm登录安装有MPPDB服务的任一主机。执行source ${BIGDATA\_HOME}/mppdb/.mppdbgs\_profile命令启动环境变量。
   4. 分别编译Geos、Proj、JSON-C、Libxml2、Gdal、PostGIS并生成相关动态链接库。编译命令为：

Geos

* + - cd $GAUSSHOME/postgis-xc/geos-3.6.2
    - chmod +x ./configure
    - ./configure --prefix=$GAUSSHOME/install/geos
    - make -sj

make install -sj

Proj

* + - cd $GAUSSHOME/postgis-xc/proj-4.9.2
    - chmod +x ./configure
    - ./configure --prefix=$GAUSSHOME/install/proj
    - make –sj
    - cd $GAUSSHOME/install
    - mkdir proj
    - cd ./proj
    - mkdi bin

make install -sj

JSON-C

* + - cd $GAUSSHOME/postgis-xc/json-c-json-c-0.12.1-20160607
    - chmod +x ./configure
    - ./configure --prefix=$GAUSSHOME/install/json
    - make -sj

make install -sj

Libxml2

* + - cd $GAUSSHOME/postgis-xc
    - tar xzf libxml2-2.7.1.tar.gz
    - cd $GAUSSHOME/postgis-xc/libxml2-2.7.1
    - chmod +x ./configure
    - ./configure --prefix=$GAUSSHOME/install/libxml2
    - make -sj

make install -sj

Gdal

* + - cd $GAUSSHOME/postgis-xc/gdal-1.11.0
    - chmod +x ./configure
    - chmod +x ./install-sh
    - ./configure --prefix=$GAUSSHOME/install/gdal --with-xml2=$GAUSSHOME/install/libxml2/bin/xml2-config --with-geos=$GAUSSHOME/install/geos/bin/geos-config --with-static\_proj4=$GAUSSHOME/install/proj CFLAGS='-O2 -fpermissive -pthread'
    - make -sj

make install –sj

如果编译出现类似/home/carrot/data/openGauss-server/third\_party/buildtools/gcc/res/lib64/libstdc++.la 找不到，可以自建目录，将libstdc++.la拷贝进去，然后再make -sj

（如果libstdc++.so出现类似问题，按同样方法处理）

PostGIS

* + - cd $GAUSSHOME/postgis-xc/postgis-2.4.2
    - sed -i -e s'/-Werror//g' $GAUSSHOME/lib/postgresql/pgxs/src/Makefile.global
    - patch -p5 < postgis\_2.4.2-1.patch
    - chmod +x ./configure
    - ./configure --prefix=$GAUSSHOME/install/pggis2.4.2 --with-pgconfig=$GAUSSHOME/bin/pg\_config --with-projdir=$GAUSSHOME/install/proj --with-geosconfig=$GAUSSHOME/install/geos/bin/geos-config --with-jsondir=$GAUSSHOME/install/json --with-xml2config=$GAUSSHOME/install/libxml2/bin/xml2-config --with-raster --with-gdalconfig=$GAUSSHOME/install/gdal/bin/gdal-config --with-topology --without-address-standardizer CFLAGS='-O2 -fpermissive -DPGXC -pthread -D\_THREAD\_SAFE -D\_\_STDC\_FORMAT\_MACROS -DMEMORY\_CONTEXT\_CHECKING -w' CC=g++
    - make -sj

make install -sj

* 1. omm用户执行$GAUSSHOME/share/postgis目录下的脚本文件PostGIS\_install.sh，完成PostGIS相关动态链接库在集群节点中的分发。

sh $GAUSSHOME/share/postgis/PostGIS\_install.sh

此脚本实现拷贝相关动态库到特定目录，内容如下，可以手动拷贝:



动态链接库分发脚本执行完毕后，可执行下列命令删除$GAUSSHOME/postgis安装目录。

rm -rf $GAUSSHOME/postgis-xc

若用户不想保留GCC7.3编译器，可删除GCC7.3安装目录并在~/.bashrc文件中删除安装GCC7.3时添加的环境配置信息。

rm -rf $GAUSSHOME/gcc

* 1. 重启集群。

gs\_om -t stop && gs\_om -t start

# PostGIS使用

#### 创建Extension

创建PostGIS Extension可直接使用CREATE EXTENSION命令进行创建：

|  |  |
| --- | --- |
| 1 | CREATE EXTENSION postgis; |
| 使能PostGIS中的栅格功能，可直接使用CREATE EXTENSION命令进行创建：   |  |  | | --- | --- | | 1 | CREATE EXTENSION postgis\_raster; | | | |  |  |
| 使能PostGIS中的拓扑功能，使用CREATE EXTENSION命令进行创建之前先执行：set behavior\_compat\_options='bind\_procedure\_searchpath';   |  |  | | --- | --- | | 1 | CREATE EXTENSION postgis\_topology; | |  |  | | | |  |  |

#### 使用Extension

PostGIS Extension函数调用格式为：

|  |  |
| --- | --- |
| 1 | SELECT GisFunction (Param1, Param2,......); |

其中GisFunction为函数名，Param1、Param2等为函数参数名。下列SQL语句展示PostGIS的简单使用，对于各函数的具体使用，请参考[《PostGIS-2.4.2用户手册》](https://download.osgeo.org/postgis/docs/postgis-2.4.2.pdf)。

示例1：几何表的创建。

|  |  |
| --- | --- |
| 1  2 | CREATE TABLE cities ( id integer, city\_name varchar(50) );  SELECT AddGeometryColumn('cities', 'position', 4326, 'POINT', 2); |

示例2：几何数据的插入。

|  |  |
| --- | --- |
| 1  2  3 | INSERT INTO cities (id, position, city\_name) VALUES (1,ST\_GeomFromText('POINT(-9.5 23)',4326),'CityA');  INSERT INTO cities (id, position, city\_name) VALUES (2,ST\_GeomFromText('POINT(-10.6 40.3)',4326),'CityB');  INSERT INTO cities (id, position, city\_name) VALUES (3,ST\_GeomFromText('POINT(20.8 30.3)',4326), 'CityC'); |

示例3：计算三个城市间任意两个城市距离。

|  |  |
| --- | --- |
| 1 | SELECT p1.city\_name,p2.city\_name,ST\_Distance(p1.position,p2.position) FROM cities AS p1, cities AS p2 WHERE p1.id > p2.id; |

#### 删除Extension

在GaussDB A中删除PostGIS Extension的方法如下所示：

|  |  |
| --- | --- |
| 1 | DROP EXTENSION postgis [CASCADE]; |

如果Extension被其它对象依赖（如创建的几何表），需要加入CASCADE（级联）关键字，删除所有依赖对象。

若要完全删除PostGIS Extension，则需由omm用户使用gs\_om工具移除PostGIS及其依赖的动态链接库，格式如下：

|  |  |
| --- | --- |
| 1 | gs\_om -t postgis -m rmlib |
|  |  |

# PostGIS支持和限制

#### 支持数据类型

GaussDB A的PostGIS Extension支持如下数据类型:

* box2d
* box3d
* geometry\_dump
* geometry
* geography
* raster
* 栅格相关的三个GUC参数postgis.gdal\_datapath、postgis.gdal\_enabled\_drivers及postgis.enable\_outdb\_rasters内部已经完全使能，使用时不用再手动设置。
* 创建Postgis和使用Postgis不是同一个用户时，请设置如下guc参数：SET behavior\_compat\_options = 'bind\_procedure\_searchpath';

#### 支持的操作符和函数列表

| **表1 PostGIS Extension支持的操作符和函数列表** | |
| --- | --- |
| **函数分类** | **包含函数** |
| Management Functions | AddGeometryColumn、DropGeometryColumn、DropGeometryTable、PostGIS\_Full\_Version、PostGIS\_GEOS\_Version、PostGIS\_Liblwgeom\_Version、PostGIS\_Lib\_Build\_Date、PostGIS\_Lib\_Version、PostGIS\_PROJ\_Version、PostGIS\_Scripts\_Build\_Date、PostGIS\_Scripts\_Installed、PostGIS\_Version、PostGIS\_LibXML\_Version、PostGIS\_Scripts\_Released、Populate\_Geometry\_Columns 、UpdateGeometrySRID |
| Geometry Constructors | ST\_BdPolyFromText 、ST\_BdMPolyFromText 、ST\_Box2dFromGeoHash、ST\_GeogFromText、ST\_GeographyFromText、ST\_GeogFromWKB、ST\_GeomCollFromText、ST\_GeomFromEWKB、ST\_GeomFromEWKT、ST\_GeometryFromText、ST\_GeomFromGeoHash、ST\_GeomFromGML、ST\_GeomFromGeoJSON、ST\_GeomFromKML、ST\_GMLToSQL、ST\_GeomFromText 、ST\_GeomFromWKB、ST\_LineFromMultiPoint、ST\_LineFromText、ST\_LineFromWKB、ST\_LinestringFromWKB、ST\_MakeBox2D、ST\_3DMakeBox、ST\_MakeEnvelope、ST\_MakePolygon、ST\_MakePoint、ST\_MakePointM、ST\_MLineFromText、ST\_MPointFromText、ST\_MPolyFromText、ST\_Point、ST\_PointFromGeoHash、ST\_PointFromText、ST\_PointFromWKB、ST\_Polygon、ST\_PolygonFromText、ST\_WKBToSQL、ST\_WKTToSQL |
| Geometry Accessors | GeometryType、ST\_Boundary、ST\_CoordDim、ST\_Dimension、ST\_EndPoint、ST\_Envelope、ST\_ExteriorRing、ST\_GeometryN、ST\_GeometryType、ST\_InteriorRingN、ST\_IsClosed、ST\_IsCollection、ST\_IsEmpty、ST\_IsRing、ST\_IsSimple、ST\_IsValid、ST\_IsValidReason、ST\_IsValidDetail、ST\_M、ST\_NDims、ST\_NPoints、ST\_NRings、ST\_NumGeometries、ST\_NumInteriorRings、ST\_NumInteriorRing、ST\_NumPatches、ST\_NumPoints、ST\_PatchN、ST\_PointN、ST\_SRID、ST\_StartPoint、ST\_Summary、ST\_X、ST\_XMax、ST\_XMin、ST\_Y、ST\_YMax、ST\_YMin、ST\_Z、ST\_ZMax、ST\_Zmflag、ST\_ZMin |
| Geometry Editors | ST\_AddPoint、ST\_Affine、ST\_Force2D、ST\_Force3D、ST\_Force3DZ、ST\_Force3DM、ST\_Force4D、ST\_ForceCollection、ST\_ForceSFS、ST\_ForceRHR、ST\_LineMerge、ST\_CollectionExtract、ST\_CollectionHomogenize、ST\_Multi、ST\_RemovePoint、ST\_Reverse、ST\_Rotate、ST\_RotateX、ST\_RotateY、ST\_RotateZ、ST\_Scale、ST\_Segmentize、ST\_SetPoint、ST\_SetSRID、ST\_SnapToGrid、ST\_Snap、ST\_Transform、ST\_Translate、ST\_TransScale |
| Geometry Outputs | ST\_AsBinary、ST\_AsEWKB、ST\_AsEWKT、ST\_AsGeoJSON、ST\_AsGML、ST\_AsHEXEWKB、ST\_AsKML、ST\_AsLatLonText 、ST\_AsSVG、ST\_AsText、ST\_AsX3D、ST\_GeoHash |
| Operators | &&、&&&、&<、&<|、&>、<<、<<|、=、>>、@ 、|&> 、|>>、~、~=、<->、<#> |
| Spatial Relationships and Measurements | ST\_3DClosestPoint、ST\_3DDistance、ST\_3DDWithin、ST\_3DDFullyWithin、ST\_3DIntersects、ST\_3DLongestLine、ST\_3DMaxDistance、ST\_3DShortestLine、ST\_Area、ST\_Azimuth、ST\_Centroid、ST\_ClosestPoint、ST\_Contains、ST\_ContainsProperly、ST\_Covers、ST\_CoveredBy、ST\_Crosses、ST\_LineCrossingDirection、ST\_Disjoint、ST\_Distance、ST\_HausdorffDistance、ST\_MaxDistance、ST\_DistanceSphere、ST\_DistanceSpheroid、ST\_DFullyWithin、ST\_DWithin、ST\_Equals、ST\_HasArc、ST\_Intersects、ST\_Length、ST\_Length2D、ST\_3DLength、ST\_Length\_Spheroid、ST\_Length2D\_Spheroid、ST\_3DLength\_Spheroid、ST\_LongestLine、ST\_OrderingEquals、ST\_Overlaps、ST\_Perimeter、ST\_Perimeter2D、ST\_3DPerimeter、ST\_PointOnSurface、ST\_Project、ST\_Relate、ST\_RelateMatch、ST\_ShortestLine、ST\_Touches、ST\_Within |
| Geometry Processing | ST\_Buffer、ST\_BuildArea、ST\_Collect、ST\_ConcaveHull、ST\_ConvexHull、ST\_CurveToLine、ST\_DelaunayTriangles、ST\_Difference、ST\_Dump、ST\_DumpPoints、ST\_DumpRings、ST\_FlipCoordinates、ST\_Intersection、ST\_LineToCurve、ST\_MakeValid、ST\_MemUnion、ST\_MinimumBoundingCircle、ST\_Polygonize、ST\_Node、ST\_OffsetCurve、ST\_RemoveRepeatedPoints、ST\_SharedPaths、ST\_Shift\_Longitude、ST\_Simplify、ST\_SimplifyPreserveTopology、ST\_Split、ST\_SymDifference、ST\_Union、ST\_UnaryUnion |
| Linear Referencing | ST\_LineInterpolatePoint、ST\_LineLocatePoint、ST\_LineSubstring、ST\_LocateAlong、ST\_LocateBetween、ST\_LocateBetweenElevations、ST\_InterpolatePoint、ST\_AddMeasure |
| Miscellaneous Functions | ST\_Accum、Box2D、Box3D、ST\_Expand、ST\_Extent、ST\_3Dextent、Find\_SRID、ST\_MemSize |
| Exceptional Functions | PostGIS\_AddBBox、PostGIS\_DropBBox、PostGIS\_HasBBox |
| Raster Management Functions | AddRasterConstraints、DropRasterConstraints、AddOverviewConstraints、DropOverviewConstraints、PostGIS\_GDAL\_Version、PostGIS\_Raster\_Lib\_Build\_Date、PostGIS\_Raster\_Lib\_Version、ST\_GDALDrivers、UpdateRasterSRID |
| Raster Constructors | ST\_AddBand、ST\_AsRaster、ST\_Band、ST\_MakeEmptyRaster、ST\_Tile、ST\_FromGDALRaster |
| Raster Accessors | ST\_GeoReference、ST\_Height、ST\_IsEmpty、ST\_MetaData、ST\_NumBands、ST\_PixelHeight、ST\_PixelWidth、ST\_ScaleX、ST\_ScaleY、ST\_RasterToWorldCoord、ST\_RasterToWorldCoordX、ST\_RasterToWorldCoordY、ST\_Rotation、ST\_SkewX、ST\_SkewY、ST\_SRID、ST\_Summary、ST\_UpperLeftX、ST\_UpperLeftY、ST\_Width、ST\_WorldToRasterCoord、ST\_WorldToRasterCoordX、ST\_WorldToRasterCoordY |
| Raster Band Accessors | ST\_BandMetaData、ST\_BandNoDataValue、ST\_BandIsNoData、ST\_BandPath、ST\_BandPixelType、ST\_HasNoBand |
| Raster Pixel Accessors and Setters | ST\_PixelAsPolygon、ST\_PixelAsPolygons、ST\_PixelAsPoint、ST\_PixelAsPoints、ST\_PixelAsCentroid、ST\_PixelAsCentroids、ST\_Value、ST\_NearestValue、ST\_Neighborhood、ST\_SetValue、ST\_SetValues、ST\_DumpValues、ST\_PixelOfValue |
| Raster Editors | ST\_SetGeoReference、ST\_SetRotation、ST\_SetScale、ST\_SetSkew、ST\_SetSRID、ST\_SetUpperLeft、ST\_Resample、ST\_Rescale、ST\_Reskew、ST\_SnapToGrid、ST\_Resize、ST\_Transform |
| Raster Band Editors | ST\_SetBandNoDataValue、ST\_SetBandIsNoData |
| Raster Band Statistics and Analytics | ST\_Count、ST\_CountAgg、ST\_Histogram、ST\_Quantile、ST\_SummaryStats、ST\_SummaryStatsAgg、ST\_ValueCount |
| Raster Outputs | ST\_AsBinary、ST\_AsGDALRaster、ST\_AsJPEG、ST\_AsPNG、ST\_AsTIFF |
| Raster Processing | ST\_Clip、ST\_ColorMap、ST\_Intersection、ST\_MapAlgebra、ST\_Reclass、ST\_Union、ST\_Distinct4ma、ST\_InvDistWeight4ma、ST\_Max4ma、ST\_Mean4ma、ST\_Min4ma、ST\_MinDist4ma、ST\_Range4ma、ST\_StdDev4ma、ST\_Sum4ma、ST\_Aspect、ST\_HillShade、ST\_Roughness、ST\_Slope、ST\_TPI、ST\_TRI、Box3D、ST\_ConvexHull、ST\_DumpAsPolygons、ST\_Envelope、ST\_MinConvexHull、ST\_Polygon、ST\_Contains、ST\_ContainsProperly、ST\_Covers、ST\_CoveredBy、ST\_Disjoint、ST\_Intersects、ST\_Overlaps、ST\_Touches、ST\_SameAlignment、ST\_NotSameAlignmentReason、ST\_Within、ST\_DWithin、ST\_DFullyWithin |
| Raster Operators | &&、&<、&>、=、@、~=、~ |

#### 空间索引

GaussDB A数据库的PostGIS Extension支持GIST (Generalized Search Tree) 空间索引（分区表除外）。相比于B-tree索引，GIST索引适应于任意类型的非常规数据结构，可有效提高几何和地理数据信息的检索效率。

使用如下命令创建GIST索引：

|  |  |
| --- | --- |
| 1 | CREATE INDEX indexname ON tablename USING GIST ( geometryfield ); |

#### 扩展限制

* 只支持行存表；
* 不支持BRIN索引；
* spatial\_ref\_sys表在扩容期间只支持查询操作。

备注：【建议使用CentOS Linux release 7.6.1810 (Core) x86\_64，已测试通过】

PostGIS编译注意事项：

1. 先将$GAUSSHOME/include/postgresql/server目录下的头文件都删除，然后将源码目录下的 src/include下面所有文件拷贝到 $GAUSSHOME/include/postgresql/server

将binarylibs中的extension\_dependency.patch文件拷贝到源码目录下，执行git am extension\_dependency.patch,将源码目录下的extension\_dependency.h拷贝到$GAUSSHOME/include/postgresql/server

1. 将binarylibs中的dependency下，openssl、cjson、kerberos、eSDKOB.hS头文件拷贝到$GAUSSHOME/include/postgresql/server中。

如：

cp -r openGauss-server/binarylibs/dependency/centos7.6\_x86\_64/openssl/comm/include/openssl $GAUSSHOME/include/postgresql/server

cp –r openGauss-server/binarylibs/dependency/centos7.6\_x86\_64/cjson/comm/include/cjson $GAUSSHOME/include/postgresql/server

cp –r openGauss-server/binarylibs/dependency/centos7.6\_x86\_64/kerberos/comm/include/gssapi\* $GAUSSHOME/include/postgresql/server

cp –r openGauss-server/binarylibs/dependency/centos7.6\_x86\_64/ libobs/comm/include/eSDKOBS.h $GAUSSHOME/include/postgresql/server/access/obs

1. 将pg\_config\_os.h 软链接删除，重新链接到 $GAUSSHOME/include/postgresql/server/port/linux.h

ln –s $GAUSSHOME/include/postgresql/server/port/linux.h $GAUSSHOME/include/postgresql/server/pg\_config\_os.h

1. 将源码中的get\_PlatForm\_str.sh 拷贝到 $GAUSSHOME/lib/postgresql/pgxs/src

该脚本在 openGauss-server/src

1. 修改postgis-2.4.2\configure文件第13518行POSTGIS\_PGSQL\_VERSION="$PGSQL\_MAJOR\_VERSION$PGSQL\_MINOR\_VERSION" 改为POSTGIS\_PGSQL\_VERSION=92

将binarylibs中的postgis.patch拷贝到postgis-2.4.2目录下面，执行git am postgis.patch;

1. PostGIS编译完成后，将相关动态库和配置文件拷贝到相应目录.

如：

cp $GAUSSHOME/json/lib/libjson-c.so.2 $GAUSSHOME/lib/libjson-c.so.2

cp $GAUSSHOME/geos/lib/libgeos\_c.so.1 $GAUSSHOME/lib/libgeos\_c.so.1

cp $GAUSSHOME/proj/lib/libproj.so.9 $GAUSSHOME/lib/libproj.so.9

cp $GAUSSHOME/geos/lib/libgeos-3.6.2.so $GAUSSHOME/lib/libgeos-3.6.2.so

cp $GAUSSHOME/pggis2.4.2/lib/liblwgeom-2.4.so.0 $GAUSSHOME/lib/liblwgeom-2.4.so.0

1. 从postgis-2.4.2源码包中，将postgis.control拷贝到$GAUSSHOME/share/postgresql/extension/

修改数据库安装目录下面的postgis\_topology--2.4.2.sql文件：修改第365行，将type integer改成type\_geo，将939行的ret.type改成ret.type\_geo,将WHEN type改成WHEN type\_geo，一共八处，将tg.type :改成tg.type\_geo :，一共四处。