Application: gvSIG desktop - gvSIG feature requests #4798
El driver de Spatialite debería usar el índice espacial

01/22/2018 05:03 PM - Cesar Martinez Izquierdo

Status:	Closed	**% Done:**	0%
Priority:	Normal	**Spent time:**	0.00 hour
Assignee:	Joaquín del Cerro Murciano		
Category:			
Target version:	2.4.0-2847 (rev. org.gvsig.desktop-2.0.216)		
gvSIG version:	2.4.0	**Add-on resolve version:**	
Keywords:		**Add-on resolve build:**	
Has patch:		**Proyecto:**	
Add-on name:	Unknown	**Hito:**	
Add-on version:			

Description

Actualmente no se están usando los índices espaciales de Spatialite, porque la query de una tabla que tiene índice espacial es diferente de la que no lo tiene. Como no distinguimos si la tabla tiene o no índice espacial, de momento se hacen todas las queries ignorando el índice espacial. Sería muy interesante que sí se usase el índice, mejoraría mucho el rendimiento para capas pesadas.

Associated revisions

Revision 43739 - 01/25/2018 08:39 AM - Joaquín del Cerro Murciano

refs #4798. Modificaciones necesarias en el core para poder usar indices espaciales con SpatiaLite

Revision 138 - 01/25/2018 08:41 AM - Joaquín del Cerro Murciano

refs #4798. Modificaciones para implementar el uso de indices espaciales con SpatiaLite en las consultas

Revision 453 - 01/25/2018 08:42 AM - Joaquín del Cerro Murciano

refs #4798. Modificaciones deribadas de la implementacion del uso de indices espaciales con SpatiaLite en las consultas

Revision 101 - 01/26/2018 12:35 PM - Joaquín del Cerro Murciano

refs #4798. Modificaciones deribadas de la implementacion del uso de indices espaciales con SpatiaLite en las consultas

Revision 154 - 01/26/2018 12:35 PM - Joaquín del Cerro Murciano

refs #4798. Modificaciones deribadas de la implementacion del uso de indices espaciales con SpatiaLite en las consultas

Revision 127 - 01/27/2018 05:51 PM - Joaquín del Cerro Murciano

refs #4798. Modificaciones deribadas de la implementacion del uso de indices espaciales con SpatiaLite en las consultas

History

#1 - 01/23/2018 12:12 PM - Joaquín del Cerro Murciano

- Target version set to 2.4.0-2850-final (rev. org.gvsig.desktop-2.0.220)

- Assignee set to Joaquín del Cerro Murciano

- Status changed from New to In progress

#2 - 01/25/2018 08:36 AM - Joaquín del Cerro Murciano

He estado viendo como habria que implementar esto ya que parecia que no solo seria cosa de tocar el proveedor de SpatiaLite pudiendo afectar tambien al core.

Al final he tenido que tocar bastantes cosas del core.

La idea ha partido de que el proveedor de SpatiaLite sobre-escribiese el metodo ST_Intersects del SQLBuilder para crear un fragmento de SQL que usase el indice espacial si este existe para alguno de los parametros de la funcion.

Hasta ahi sencillo. Pero a la funcion le llegaba solamente el nombre de la columna geometria. No le llegaba informacion de si existe un indice creado para esa columna, ni siquiera del tipo de datos de la columna.

Para solucionar esto he pensado que cuando se llama al metodo "column" del SQLBuilder, que ahora mismo recibe unicamente el nombre de la columna, se pudiese pasar tanto el nombre como el FeatureAttributeDescriptor (fad) de la columna. Asi si se recibe el fad, tendriamos mas informacion sobre la columna, incluyendo si existe o no un indice espacial sobre ella.

Para hacer esto he tenido que subir al ExpressionBuilder el interface y la clase ColumnDescriptorBuilder/ColumnDescriptorBuilderBase que habia el el SQLBuilder y cuando se crea un objeto Variable/VariableBase con el metodo column+fad se le inserta un ColumnDescriptor con todos los datos que se conocen de la columna. Si se llama a column con un nombre de columna el ColumnDescriptor asociado estara a null.

Con estos cambios se puede tener acceso al ColumnDescriptor desde dentro de la funcion ST_Intersects de SpatiaLite, y ya podemos consultar si hay indices espaciales sobre esa columna para adaptar la sql.

El problema que me he encontrado entondes es que tenia acceso a la informacion que describe la columna; pero no podia saber a que tabla pertenecia esa columna. Para hacerlo he modificado el FeatureAttributeDescriptor y el FeatureType, de forma que al fad se le pueda consultar a que FeatureType pertenece y al FeatureType se le puede preguntar a que store pertenece.

Con estos cambios, cuando se llama al metodo column del SQLBuilder se guarda, un ColumnDescriptor que tiene la informacion que describe a la columna y ademas los parametros asociados al store a la que pertenece (en el caso de jdbc, datos de conexion y tabla a la que pertenece, por ejemplo).

Los cambios necesarios para implementar esto los puedo agrupar en:

- Cambios en el core en el SQLBuilder/SQLExpression, sobre todo deribados de mover ColumnDescriptorBuilder/ColumnDescriptorBuilderBase. Ademas he cambiado el nombre de ColumnDescriptorBuilder a ColumnDescriptor. Esto genera perdida de compatibilidad con el codigo hasta ahora existente en los proveedores de BBDD, pero como estas clases se introdujeros post gvSIG 2.3, y la 2.4 aun no ha salido a la calle, no deberia haber nadie usandolas aun.

- Cambios en el FeatureAttributeDescriptor y FeatureType. Los cambios en el API son menores. Aparecen uno o dos metodos, getStore y getFeatureType. Para implementar estos metodos ha habido que hacer bastantes modificaciones en el core; pero no parece que vayan a introducir problemas.

- Cambios en las clases que implementan los evaluadores de MapContext usados en SpatialEvaluatorsFactory. El cambio es muy leve, Unicamante que cuando se llama a la funcion "column" del ExpressionBuilder se le pasa el fad en lugar del nombre del campo. Lo he hecho para todos los evaluadores aunque en principio el proveedor de SpatiaLite solo usaria los de intersect. Esto permitiria hacer otras optimizaciones con otras funciones como podria ser con crosses o contains.

- Cambios en el SQLBuilder del proveedor de SpatiaLite. Al final de la jugada aqui es donde menos cambios hay, simplemente se sobreescribe el metodo ST_Intersects y en funcion de los parameytros de este se decide crear una fragmento de sql u otro.

- Cambios en los proveedores de base de datos existentes (H2, PostgreSQL, MSSQLServer, Oracle) para adaptar el cambio de nombre del ColumnDescriptorBuilder.

EL proveedor ahora mismo esta generando algo como:

```
(ST_Intersects(ST_GeomFromWKB(...),"...column-name...") AND ROWID IN (
  SELECT ROWID FROM SpatialIndex WHERE
    f_table_name = '...table-name...' AND
    f_geometry_column = '...column-name...' AND
    search_frame = "...table-name..."."...column-name..."
))
```

Acabando teniendose sentencias como:

```
SELECT ST_AsBinary("geometry"), "pk" FROM "esp_provincias_4326"
WHERE
 ( ST_Intersects(ST_GeomFromWKB((x'000...034'), (4326)),"geometry") AND
   ROWID IN (
    SELECT ROWID FROM SpatialIndex
      WHERE
        f_table_name = 'esp_provincias_4326' AND
        f_geometry_column = 'geometry' AND
        search_frame = "esp_provincias_4326"."geometry"
   )
 )
```

Aun hay algun problema. En ocasiones la capa me ha parecido que deja de usar el indice espacial. Pero lo voy a dejar aqui y cuando haya mas tiempo ya lo proberemos mas y de ser asi ya se levanta un bug con ello.

Ademas de esto, no tengo claro la mejora de rendimiento. En general puede ser alta, pero tal como se realizan las consultas por indices espaciales en SpatiaLite intuyo que pueden haber casos que acabe siendo mas lento buscar por indice espacial que sin el (cuando se haga un intersect con una superficie que devuelba todos los registros y la capa sea muy grande). Pero bueno, es cuestion de ir viendolo segun lo usemos.

#3 - 01/25/2018 08:43 AM - Joaquín del Cerro Murciano

- Target version changed from 2.4.0-2850-final (rev. org.gvsig.desktop-2.0.220) to 2.4.0-2847 (rev. org.gvsig.desktop-2.0.216)

- Status changed from In progress to Fixed

#4 - 01/30/2018 09:16 AM - Álvaro Anguix

- Status changed from Fixed to Closed