QGIS Application - Feature request #10947
Use Filter() instead of STIntersects() for better performance in MSSQL

2014-07-24 03:43 PM - Markus Woehling

Status:	Closed		
Priority:	High		
Assignee:	Nathan Woodrow		
Category:	Data Provider/MSSQL		
Pull Request or Patch supplied:	No	**Resolution:**	
Easy fix?:	No	**Copied to github as #:**	19300

Description

I've noticed that "STIntersects()" is used to query the currently visible elements - I think you should consider using "Filter()" to get (much) better performance.

Associated revisions

Revision 62af54ec - 2017-06-02 04:33 AM - Nyall Dawson

[mssql] Use Filter instead of STIntersects to improve query performance

...and refine validity test from 57dc3c7

Using Filter is more performant since it does a bounding box only
check when an appropriate spatial index is available. This matches
the behavior with other providers, where the provider filter only
does a bounding box check and callers must perform the actual
intersection check if required.

While Filter also avoids SQL server closing the iterator upon
encountering invalid geometries, we can't rely on this because SQL
server will transparently fall back to STIntersects if it
decides there's no suitable spatial indexes available, and then
throw an exception on invalid geometries. So we still require
the STISValid check when using Filter.

The extent calculation from 57dc3c7 has been refined to
avoid the very expensive STMakeValid call. Instead we use a (ugly!)
workaround to skip invalid geometries using STIsValid only
inside the min/max x/y aggregates. Note that we can't just
dump a WHERE STIsValid clause in to the statement because SQL
server still throws the exception. Gotta love it!

Fix #15752, #10947

History

#1 - 2014-07-24 04:08 PM - Nathan Woodrow

- Assignee set to Nathan Woodrow

#2 - 2014-07-24 04:18 PM - Nathan Woodrow

I have thought that this was the case too however I can't seem to get better performance on SQL Server 2008. Could be my data and indexes, although I

wouldn't suspect so.

Are you able to run a few on your database to see if you see any difference.

#3 - 2014-07-25 04:15 PM - Markus Woehling

I've done some performance tests with interesting results ;-)

I used "SELECT <id> FROM <table> WHERE ..." with four different conditions:
1) WHERE <geo>.STIntersects(geometry::Parse(<wkt>)) = 1
2) WHERE <geo>.STIntersects(@geometryParameter) = 1
3) WHERE <geo>.Filter(geometry::Parse(<wkt>)) = 1
4) WHERE <geo>.Filter(@geometryParameter) = 1

I used four different tables with 100,000 to 1,000,000 rows (three tables with points, one with linestrings).
The tables have quite different "fragmentation", because some of them have rows with "wrong" coordinates (e.g. points at with x=0, y=0).

The results on SQL 2008 are;
Case 4) always gives us the best performance, case 1) always the worst (factor 3 to 10!).
Case 2) and 3) are in the middle, sometimes 2) is better, sometimes 3).

The results on SQL 2012 quite the same, but in some cases 1) is even 100 times slower than 4), because it's 10 times slower than on SQL 2008. Maybe this depends on my hardware, SQL 2008 and SQL 2012 are running on different machines.

So I think you should always use Filter() and you should try to use a parameter for the geometry.

I also tried to pass the wkt-string as parameter [WHERE <geo>.STIntersects(geometry::Parse(@wkt)) = 1]: This makes no difference.

I used .NET for the tests, so this is the SQL statement for case 4):
declare @p3 sys.geometry
set @p3=convert(sys.geometry,0x0...binary.stuff...0)
exec sp_executesql N'SELECT <id> FROM <table> WHERE <geo>.Filter(@box) = 1',N'@box [geometry]',@box=@p3

Hope that helps.

#4 - 2014-07-25 06:42 PM - Nathan Woodrow

Thanks for the tests. I will implement 4) if it's the fastest.

#5 - 2016-06-24 10:20 PM - Nathan Woodrow

- Priority changed from Normal to High

#6 - 2016-10-21 10:27 AM - Markus Woehling

I think 4) is more complex to implement than 3).
Because 3) would also improve performance a lot, I want to propose that implementation of 3) would be enough for now.
There are even some corner cases where the SQL query optimizer does NOT use a spatial index if STIntersects() is used instead of Filter().
In this cases 3) would help a lot (not using a spatial index degrades performance massively, of course).

#7 - 2017-05-01 12:47 AM - Giovanni Manghi

- Easy fix? set to No

#8 - 2017-06-03 08:03 AM - Nyall Dawson

- % Done changed from 0 to 100

- Status changed from Open to Closed

Applied in changeset commit:qgis|62af54ecb2666d0bde0f7af102d663ded5cda97d.