

QGIS Application - Bug report #9444

WFS client not requesting new features when not-cached

2014-01-28 12:19 AM - Jonathan Moules

Status:	Closed		
Priority:	Normal		
Assignee:			
Category:	Web Services clients/WFS		
Affected QGIS version:	master	Regression?:	No
Operating System:		Easy fix?:	No
Pull Request or Patch supplied:		Resolution:	
Crashes QGIS or corrupts data:		Copied to github as #:	18038
Description			
<p>Might be a regression.</p> <p>1) Add http://maps.warwickshire.gov.uk/gs/ows as a WFS service</p> <p>2) Add "Street Lights" - uncheck "cache features".</p> <p>500 random features should be returned - this is feature limit of the service. The dataset contains 51639 features.</p> <p>3) Zoom in and pan around all you want - QGIS won't make any requests to the WFS server.</p> <p>4) Zoom out. Once you have zoomed out further than the the original level, QGIS will make a new request and will do so every time you zoom out further.</p> <p>Assuming it's confirmed and not just me, I suggest this might be a blocker - it makes WFS unusable on feature limited services.</p> <p>Master - 51928ed</p>			
Related issues:			
Related to QGIS Application - Feature request # 9450: New WFS connection opti...		Open	2014-01-29

History

#1 - 2014-01-28 01:07 AM - Neil Benny

We have been experiencing the same issue. Datasets of different sizes and with different feature limits all have this same issue.

When we load a limited WFS into QGIS as a non-cached layer it will load the area that can be seen on the map window, if more features than are viewable given WFS feature limits are located on the current view, it will draw up to the feature limit (in my case 5,000) and then stop. No message is given to say that the feature limit has been reached.

If you then zoom or pan the view QGIS won't make a new request on the WFS server, leaving the only option to remove the layer and add it in again (refresh button has no effect).

This has been the case on the QGIS 2.0.1 release and the 105ecfa master.

#2 - 2014-01-28 01:18 AM - Jukka Rahkonen

Everybody can not repeat the issue with these steps but one must zoom to correct area before 2) gives any features. They are not really random features but 500 first features from within the BBOX of the map window. If BBOX of the map is outside data extents the query does not find anything from WFS. Area around coordinates 401523,260523 in EPSG:27700 should work, of then one can make first request with "cache features" checked and zoom to

layer for getting to the right area.

Otherwise I can confirm Jonathan's observations.

#3 - 2014-01-28 01:42 AM - Richard Duivenvoorde

sorry for the longread...

I wrote a small blog about this 'problem' <http://www.ggis.nl/2013/04/29/qgis-en-wfs-caching/?lang=en> and thought about it. But I think you cannot blame QGIS for this problem, or at least not make it the only one to solve this.

Crux is that when you have 'cached' checked, QGIS sends a WFS request WITHOUT boundingbox, while if you do a cached layer QGIS does NOT send a bbox (and is hoping that it receives all features in one go....).

Apparently you want to see more Street Lights the the server gives back in one time. So: let the WFS return 10000 features then. Then uncheck cached, and QGIS will ask for all features in your bbox every pan/zoom action. As long as you will not zoom out at a level in which there are more then 10000 lights, you are ok.

If you come in a zoom level in which there are MORE features then the SERVER returns in one request, I think you have a problem, because most server implementations will always return the same features if you request them without a bbox.

What you want is for different reasons hard to implement I think.

First you want QGIS to do requests for every pan/zoom action.

A) You can have that with the non-cached option.

B) Even if QGIS sends a request every time you pan one pixel, but the server will just send you the first x features it has. You will not be helped because those will be the same features all the time.

Only 'feature' I can think of is that QGIS sends bbox requests, AND cache all features it receives.

BUT:

1) this will not fix point B above

2) it will force QGIS to whenever it receives features, to see which of those features are already available in the layer (retrieved from the server) and ignore those when adding to the layer.

So while we maybe could implement 2) I think it is a data/server problem.

If you want from this server see/retrieve a lot of features at one time: raise the maxnum features to return at the server (preferably all features in one go, otherwise sending a bbox wfs request to it will make it harder for your feature). Or just make it a download service :-)

Other option: make the layer both a WMS and a WFS layer (non cached), so you will have best of both worlds.

I'm not sure if the feature request 2) is hard to implement, but I think the intrinsic nature of wfs services is such that it is best served as a mix with WMS if dealing with bigger datasets.

#4 - 2014-01-28 04:20 AM - Jonathan Moules

Hi Richard - I think you're more referring to what I reported in this ticket - #8871 - which is a different thing. I know from that how the cached option works. Problem is, the non-cached isn't working. Absolutely no requests are sent except for zoom-out-beyond-previous-max-zoom events. When they are sent, they include a BBOX.

Increasing the service limit is undesirable - it's low as a means of reducing the risk of an accidental Denial Of Service (DOS) from over-enthusiastic users. :-)

#5 - 2014-01-28 06:20 AM - Richard Duivenvoorde

if I run QGIS in debug mode here on Linux, and load your wfs and pan the view, I see this message:

src/providers/wfs/qgswfsprovider.cpp: 270: (getFeatures) Layer Street Lights GetRenderedOnly: no fetch required

coming from

<https://github.com/qgis/QGIS/blob/master/src/providers/wfs/qgswfsprovider.cpp#L270>

so QGIS is looking if (see comments): "has rendered extent expanded beyond last-retrieved WFS extent" so it assumes that IF it has retrieved the features in a certain extent, that it has retrieved ALL features that are in that extent?

So in your case, if you start with a very big bbox, and receive only 500 points (of the 5000 available in THAT extent), then QGIS assumes it has retrieved all features for that bbox and will not send a request anymore until you move out of that big bbox.

While if you start with a small bbox, QGIS will keep requesting features.

Seems indeed there is a bug somewhere there in the logic...

#6 - 2014-01-28 06:33 AM - Richard Duivenvoorde

On second thought, it is not so much a bug in the logic I think, but more that what you want is a different 'refresh features strategy' (thinking about the different OpenLayers Strategy classes here): an 'always fire WFS request'-strategy.

While now we have a 'you have received all features in this extent, I'll only fire a new request if your extent is outside the extent from the last request'-strategy.

QGIS cannot know that you only received part of the features in the last WFS request...

If you would raise the maxfeaturenumber of your WFS to 10000, I think you would be happy with current 'strategy' :-)

But because of the low maxfeaturenumber now, the best way for you would be to have QGIS request features ALL THE TIME?

So, not so much a 'bug' but more a discrepancy between client and server limitations/expectations

#7 - 2014-01-28 07:23 AM - Jukka Rahkonen

Hi,

Not all QGIS users are also managing the WFS server they are using so they could raise the server side maxFeatures limit. I think that "Fire a new request every time you pan or zoom" is a good default strategy for big layers. QGIS might have a more visible message than the feature count that there is now in the bottom left corner telling "You received 500 features from WFS". That could ring bells for the user that perhaps all features did not come and it would be better to zoom in for getting a smaller BBOX.

There could be an advanced setting "I know the server side maxFeatures limit, it is xxxx, please honour it" and then QGIS could decide not to request new data from WFS if map window after pan/zoom is within the original extents and feature count is less than maxFeatures. This would be a good strategy with my server which has maxFeatures at 100000 features. There could also be a client side maxFeatures limit because reading 100000 addresses or land parcels from WFS is not necessarily what the user really wants to do.

I actually miss a third "cache with BBOX" strategy: QGIS takes the extents of the map view or user draws a rectangle on the map and QGIS is using that in BBOX filter of WFS request. QGIS 1.8 (?) used to have a check box for view extents and I used that option a lot.

#8 - 2014-01-29 04:13 AM - Jonathan Moules

| If you would raise the maxfeaturenumber of your WFS to 10000, I think you would be happy with current 'strategy' :-)

No because there are > 50,000 features in the layer. And if I increase it to 60,000 what about if/when I put a layer in with >1 million features? There's no way I'm ever setting it that high (although it'd be interesting to know if QGIS or GeoServer would flake out first :-))! :-o

| But because of the low maxfeaturenumber now, the best way for you would be to have QGIS request features ALL THE TIME?

That's what I expect it to do. I turned off the "cache features" precisely because that's how I want it to behave!

Jukkas suggestions are good, though we're at risk of turning this into a duplicate of #8871

I think to resolve this ticket:

- QGIS should always request new data on any pan/zoom when "caching" is disabled.

I've also created - #9450 for the good suggest from Jukka about the user manually entering a number

#9 - 2014-05-26 07:40 AM - Vincent Mora

Actually, the bug with no-cache seems to have worsened with time. With today's master, the `QgsWFSProvider::getFeatures(const QgsFeatureRequest& request)` is never called. This function is where the special treatment of BBOX (i.e. not cached) is made.

For testing I used the wfs sources above. With cache I get 500 features, without only one feature (probably as a side effect of `typeDetectionUri.addQueryItem("MAXFEATURES", "1")` when the feature type is unknown since with other source, no features are fetched)

To prove that, with the non-cached wfs layer selected, in the python console:

```
>>>len([f for f in iface.activeLayer().getFeatures(QgsFeatureRequest(iface.mapCanvas().extent()))])
1
>>>len([f for f in iface.activeLayer().dataProvider().getFeatures(QgsFeatureRequest(iface.mapCanvas().extent()))])
500
```

I think a lot of the code in `QgsWFSProvider` should be moved to `QgsWFSFeatureSource`.

#10 - 2014-12-19 05:42 AM - Maarten Vermeyen

Hi,

Just dropping in on this discussion, my apologies if this remark has already been made elsewhere.

| QGIS cannot know that you only received part of the features in the last WFS request...

It would be able to know this if the WFS servers set a 206 status code and a corresponding range header when returning partial content. In this case, a client like qgis would know if an incomplete result was returned, regardless of the specified feature limit on the server or bounding box.

An improved caching mechanism would be able to leverage this. If a request with a BBOX returned a 200 status code, no new request would be sent to the server when zooming in. If it gets 206 back with a range header, indicating an incomplete result, the client could do two things:

1. send additional request for partial content to get the subsequent ranges, based on the range header returned by the server. This would mean that the full set is always returned, but multiple http requests are sent by the client for a single zoom action.
2. display the incomplete result set, but send out new requests when zooming further in.

Off course this depends to a large extent on the server implementation. I haven't checked whether there are any WFS servers implementing this.

Cheers

#11 - 2014-12-25 06:59 AM - Jukka Rahkonen

The issue is not that server would send incomplete response. This discussion handles the case when the requested BBOX contains 2000 features but WFS server is set to send at maximum 1000 features. The response from the server is technically correct and complete.

With WFS servers which support paging the completeness could be tested by making a new paged request. "Got 1000 features, try what happens with a new request with &COUNT=1000&START_INDEX=999?". If no new features are returned, the first set was complete. If new features are returned but number of those is less than 1000 the set is complete now. Otherwise continue with START_INDEX=1999.

Practical problem is that GetFeature from WFS server without paging parameters and for example with parameters COUNT=1000 and START_INDEX=0 do not necessarily return the same 1000 features.

#12 - 2015-11-06 11:51 AM - Raymond Nijssen

@Maarten I like this idea (using status 206) since it is really hard to find out if the server limit has been reached. Unfortunately GeoServer (2.7.1.1) returns a subset with status 200 ok. I haven't checked other wfs servers.

What about QGIS just showing a quick warning message when the number of received features is a multiple of 500? Easy to implement and 99.8 % accurate. Quite similar to Jukka's proposal 2 years ago.

#13 - 2015-11-29 09:05 AM - Tom Vrijbrief

Vincent Mora wrote:

Actually, the bug with no-cache seems to have worsened with time. With today's master, the QgsWFSProvider::getFeatures(const QgsFeatureRequest& request) is never called. This function is where the special treatment of BBOX (i.e. not cached) is made.

For testing I used the wfs sources above. With cache I get 500 features, without only one feature (probably as a side effect of typeDetectionUri.addQueryItem("MAXFEATURES", "1") when the feature type is unknown since with other source, no features are fetched)

To prove that, with the non-cached wfs layer selected, in the python console:

```
| | | len([f for f in iface.activeLayer().getFeatures(QgsFeatureRequest(iface.mapCanvas().extent()))])
```

1

```
| | | len([f for f in iface.activeLayer().dataProvider().getFeatures(QgsFeatureRequest(iface.mapCanvas().extent()))])
```

500

I think a lot of the code in QgsWFSProvider should be moved to QgsWFSFeatureSource.

I fixed this behaviour so data is fetched for a non caching WFS layer when you pan the view.

<https://github.com/tomtor/QGIS/commit/188625cd21b339b080c13df55283c5218107de7b>

#14 - 2016-01-16 12:20 PM - Jeremy Palmer

Thanks Tom for the fix. The general non-caching issue (e.g panning and new fetches are made) has now been resolved, however the fix for the server-side max features could be improved a little by raising the issue (e.g Exactly 500 features fetched which suggests hitting a download limit) as a notification so users can directly made aware of the limitation.

#15 - 2016-01-16 12:31 PM - Jeremy Palmer

In regards to the server-side max feature limitation, I believe this ticket's approach would be more robust: #9450 (as reference above), because the current code assumes the server administrator has set 500 as the limit.

#16 - 2016-01-16 04:57 PM - Jeremy Palmer

I've also noticed another issue when using the non-cached option which is similar to server-side feature limit issue. How to replicate:

1. Add a large WFS layer, uncheck the cache option
2. Zoom to the layer's extent, abort the WFS fetch GML Data dialog before it finishes.

The layer is now at it's full extent but has an incomplete cache of features. It doesn't matter if you zoom out further or zoom in, the layer cache can not be refreshed as the provider thinks it has all of the features.

#17 - 2016-01-19 09:28 AM - Even Rouault

If the request would be done in WFS 2.0.0, the GetCapabilities answer would give us the number of maximum features that can be returned :

<http://maps.warwickshire.gov.uk/gs/ows?SERVICE=WFS&REQUEST=GetCapabilities>

--> <ows:Constraint name="CountDefault"><ows:NoValues/><ows:DefaultValue>500</ows:DefaultValue></ows:Constraint></ows:Operation>

And GetFeature too :

http://maps.warwickshire.gov.uk/gs/ows?SERVICE=WFS&REQUEST=GetFeature&VERSION=2.0.0&TYPENAME=Public_Data_DB:STREET_LIGHTS_WSHIRE

--> <wfs:FeatureCollection numberMatched="52408" numberReturned="500"...>

If numberMatched > numberReturned, the response got truncated.

#18 - 2016-04-11 11:05 AM - Even Rouault

- Target version set to Version 2.16
- Status changed from Open to Closed
- % Done changed from 0 to 100

Fixed in 2.16 per commit:9040ec1