

D4Science Infrastructure - Task #1504

Task # 1502 (Closed): Automate the dataminer installation as a loadbalanced service

Install a haproxy instance in front of the dev dataminer services

Nov 23, 2015 04:15 PM - Andrea Dell'Amico

Status:	Closed	Start date:	Nov 23, 2015
Priority:	Normal	Due date:	
Assignee:	_InfraScience Systems Engineer	% Done:	100%
Category:	System Application	Estimated time:	0.00 hour
Target version:	Computational Infrastructure upgrade to smartgears		
Infrastructure:	Development		
Description			
The haproxy instance can use the round robin balancer. It seems that nor cookies or sessions need to be provided.			

History

#1 - Nov 23, 2015 04:16 PM - Andrea Dell'Amico

- Target version changed from 197 to Computational Infrastructure upgrade to smartgears

#2 - Nov 26, 2015 11:47 AM - Andrea Dell'Amico

- Status changed from New to In Progress

Hostname and IP will be: dataminer-d-d4s.d4science.org 146.48.123.63

#3 - Dec 02, 2015 12:15 PM - Andrea Dell'Amico

- Subject changed from Install a haproxy instance in front of the dataminer services to Install a haproxy instance in front of the dev dataminer services

#4 - Dec 02, 2015 01:03 PM - Andrea Dell'Amico

- % Done changed from 0 to 40

Created the VM, starting the configuration.

#5 - Dec 02, 2015 02:49 PM - Andrea Dell'Amico

- Status changed from In Progress to Feedback

- % Done changed from 40 to 100

The haproxy balancer is ready.

The nginx logging configuration on the target host (dataminer2-d-d4s.d4science.org only) has been changed to log both the haproxy IP and the original one. The access_log line is now, for example:

```
146.48.123.63 forwarded for 146.48.123.149 - - [02/Dec/2015:14:46:54 +0100] "GET /wps/ HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/46.0.2490.86 Safari/537.36"
```

#6 - Dec 02, 2015 02:50 PM - Andrea Dell'Amico

The dataminer[1:2]-d-d4s.d4science.org nodes are still directly reachable on port 80, as requested.

#7 - Dec 10, 2015 05:02 PM - Gianpaolo Coro

Calls to dataminer-d-d4s.d4science.org are always sent to dataminer1. @andrea.dellamico@isti.cnr.it could you please give it a look?

#8 - Dec 10, 2015 05:39 PM - Andrea Dell'Amico

Can you try from a different IP? Even if it's a round robin configuration haproxy tries to deliver the requests to the same backend when the source does not change.

And a request: what URL should the load balance check to be sure that the service is working? I'm now testing /wps, but it's a 302.

#9 - Dec 10, 2015 06:08 PM - Gianpaolo Coro

- File `TestDevTokenLoadBalance.txt` added

I attach all the http links to test the balancer and the algorithms on Dataminer.

#10 - Dec 10, 2015 06:12 PM - Andrea Dell'Amico

Well, you need to chose one for the load balancer :). A light one possibly, they're executed once per second. We can use some of the others to produce a detailed nagios check, maybe.

#11 - Dec 10, 2015 07:11 PM - Andrea Dell'Amico

- Status changed from *Feedback* to *In Progress*

- % Done changed from 100 to 80

I've investigated the timeout occurrences a bit, and it seems a matter of tuning the client and server timeouts in the haproxy configuration. I've raised them both, and now the query

http://dataminer-d-d4s.d4science.org/wps/WebProcessingService?request=Execute&service=WPS&Version=1.0.0&gcube-token=d7a4076c-e8c1-42fe-81e0-bdecbe1e8074a&lang=en-US&Identifier=org.gcube.dataanalysis.wps.statisticalmanager.synchserver.mappedclasses.generators.BIONYM&DataInputs=Matcher_1=LEVENSHTEIN;Matcher_4=NONE;Matcher_5=NONE;Matcher_2=NONE;Matcher_3=NONE;Threshold_1=0.6;RawTaxaNamesTable=http://goo.gl/N9e3pC;Threshold_2=0.6;Accuracy_vs_Speed=MAX_ACCURACY;MaxResults_2=10;MaxResults_1=10;Threshold_3=0.4;RawNamesColumn=species;Taxa_Authority_File=FISHBASE;Parser_Name=SIMPLE;OutputTableLabel=bionymwps;MaxResults_4=0;Threshold_4=0;MaxResults_3=0;MaxResults_5=0;Threshold_5=0;Use_Stemmed_Genus_and_Species=false;Activate_Preparing_Processing=true

seems to always succeed. We should test with longer ones, if possible: the client sends keepalive signals to the server, we need to set the timeout higher than that value.

#12 - Dec 11, 2015 08:37 AM - Gianpaolo Coro

According to the WPS specifications, timeout should be decided by the user's client. Processing can last also days and theoretically could be executed also in "synchronous" mode by the client. Is it possible to either set the timeout to infinite or to 10/20 days?

#13 - Dec 11, 2015 08:53 AM - Andrea Dell'Amico

There is no 'infinite timeout' in tcp or http.

We only need to ensure that the proxy keep alive timeout is longer than the client (or the standard tcp) one.

#14 - Dec 11, 2015 09:41 AM - Gianpaolo Coro

<>
Not according to Apache: <https://hc.apache.org/httpcomponents-client-4.2.x/tutorial/html/connmgmt.html>

Anyway, if infinite is not contemplated by haproxy, is it possible to set it to 20 days at least?

#15 - Dec 11, 2015 10:59 AM - Andrea Dell'Amico

Gianpaolo Coro wrote:

<>
Not according to Apache: <https://hc.apache.org/httpcomponents-client-4.2.x/tutorial/html/connmgmt.html>

Anyway, if infinite is not contemplated by haproxy, is it possible to set it to 20 days at least?

That only says that the *client* will wait indefinitely. Servers have timeouts too. At http level, and tcp level. What we are interested in for this case is called http persistent connections and the way for them to work reliably is to have keepalive doing the job correctly.

Yesterday I raised the haproxy keepalive timeouts so that they should be much longer of the keepalive interval, and your tests that weren't reliable now are always completing.

We now need tests that last longer, and only if they fail because of timeouts we can try and raise them to an unreasonable high level.

#16 - Dec 11, 2015 11:07 AM - Gianpaolo Coro

Since we are talking about a computational service, the longest run I have lasts 7 days in the development environment. I don't know the computational time of all the possible algorithms we will integrate in the future.

Is it possible to quantify the current timeout or shall we wait 7 days to understand if another tuning is needed?

#17 - Dec 11, 2015 11:20 AM - Andrea Dell'Amico

Gianpaolo Coro wrote:

Since we are talking about a computational service, the longest run I have lasts 7 days in the development environment. I don't know the computational time of all the possible algorithms we will integrate in the future.

Is it possible to quantify the current timeout or shall we wait 7 days to understand if another tuning is needed?

I only have to check the client behaviour. A test lasting some then minutes is sufficient. If it fails I have a last haproxy configuration options to try, before falling back to increase the timeouts.

#18 - Dec 11, 2015 03:38 PM - Andrea Dell'Amico

- *Status changed from In Progress to Feedback*

- *% Done changed from 80 to 100*

It turned out that the http library used by the R jobs does not behave correctly with keep alive. So we are going to use a huge keepalive timeout (60 days) for the time being.

#19 - Dec 11, 2015 07:09 PM - Gianpaolo Coro

- *Status changed from Feedback to Closed*

Files

TestDevTokenLoadBalance.txt	29.3 KB	Dec 10, 2015	Gianpaolo Coro
-----------------------------	---------	--------------	----------------