

D4Science Infrastructure - Task #12031

gCubeApps - IS-Collector: too many open connections

Jun 26, 2018 10:10 AM - Roberto Cirillo

Status:	Closed	Start date:	Jun 26, 2018
Priority:	Urgent	Due date:	
Assignee:	Luca Frosini	% Done:	100%
Category:	Application	Estimated time:	0.00 hour
Target version:	node66.p: too many open connection		
Infrastructure:	Production		
Description			
There are too many open connections on IS-Collector of gCubeApps VO hosted on: <code>node66.p.d4science.research-infrastructures.eu:8080</code> At this moment there are 445 opened connections: making a comparison with D4Research VO, at this moment the connections opened are only 4. In addition on node66.p there are some clients with an high number of connections opened. At this moment, from 90.147.166.112 we have 67 opened connections. Where 90.147.166.112 has the following hostname <code>ip-90-147-166-112.ct1.garrservices.it.</code> Furthermore I'm not able to access to this host. All this opened connections making the IS-collector service too slow. We should find all the hosts with more than one opened connection and understand why there are more than one opened connection in order to decrease the number of opened connections on node66.p.d4science.			
Related issues:			
Related to D4Science Infrastructure - Support #12032: ip-90-147-166-112.ct1.g...		Closed	Jun 26, 2018
Related to D4Science Infrastructure - Support #12033: rstudio2.d4science.org:...		Closed	Jun 26, 2018
Related to D4Science Infrastructure - Support #12034: ckan-bb1.d4science.org:...		Closed	Jun 26, 2018
Related to D4Science Infrastructure - Task #12058: IS-Collector: Restart all ...		Closed	Jun 28, 2018
Related to D4Science Infrastructure - Support #12254: geoserver-french-tunaat...		Closed	Jul 27, 2018

History

#1 - Jun 26, 2018 10:12 AM - Roberto Cirillo

- Status changed from New to In Progress

#2 - Jun 26, 2018 10:17 AM - Roberto Cirillo

- Related to Support #12032: ip-90-147-166-112.ct1.garrservices.it: too many opened connection to node66.p.d4science added

#3 - Jun 26, 2018 10:37 AM - Roberto Cirillo

- Related to Support #12033: rstudio2.d4science.org: too many opened connetions to node66.p.d4science added

#4 - Jun 26, 2018 11:21 AM - Roberto Cirillo

- Related to Support #12034: ckan-bb1.d4science.org: too many opened connections to node66.p.d4science added

#5 - Jun 26, 2018 04:01 PM - Roberto Cirillo

- File rstudio2-ghn.2018-06-25.0.log added

- File ckan-bb1-ghn.2018-06-25 (cut).log added

- Status changed from In Progress to Feedback

- Assignee changed from Roberto Cirillo to Luca Frosini

- % Done changed from 0 to 50

I think, finally, I've localized the problem.

Yesterday, at 6 p.m. I've restarted the IS-Collector and Registry services of gCubeApps VO in order to see if the connections number decreased on node66.p but nothing changed.

Today I've identified two hosts with an high number of opened connections to node66: rstudio2.d4science.org, ckan-bb1.d4science.org but unfortunately, from container's logs I've not found any exception.

Now, I've checked the log on both the containers when the IS-Collector was restarted (yesterday at 6 p.m) and I found the same behavior on both the containers:

first exception:

```
18:02:12.357 [pool-2-thread-2] ERROR ProfileManager: cannot publish container (see details)
com.sun.xml.internal.ws.client.ClientTransportException: HTTP transport error: java.net.ConnectException: Connection refused (Connection refused)
```

I think this is right, because the container try to register the profile to the Registry every minute.

After that exception, on both the containers there are hundred of accounting exceptions.

This means, in my opinion, that each exception correspond to one connection to IS-collector. So the high number of connection is caused by accounting framework.

```
18:04:58.061 [AccountingScheduledThread-46] ERROR MethodRetriever: exception invoking method execute in service com.sun.proxy.$Proxy47 using proxy
18:04:58.061 [AccountingScheduledThread-46] ERROR PersistenceBackendConfiguration: InvocationTargetException - AccountingPersistenceConfiguration not initialized correctly. It will not be used. Trying the next one if any.
java.lang.RuntimeException: com.sun.xml.internal.ws.fault.ServerSOAPFaultException: Client received SOAP Fault from server: null Please see the server log to find more detail regarding exact cause of the failure.
    at org.gcube.resources.discovery.icclient.ICClient.submit(ICClient.java:47)
```

Please @luca.frosini@isti.cnr.it could you take a look and give me your opinion?

Please notice that on rstudio2 is running the smartgears distro released on gcube 4.10 but on ckan-bb1 there is the smartgears distro released on gCube 4.11

In attachment the ghn.log of both the containers.

Any opinion is appreciated.

#6 - Jun 26, 2018 04:22 PM - Luca Frosini

- Status changed from Feedback to In Progress

#7 - Jun 26, 2018 06:09 PM - Luca Frosini

- Status changed from In Progress to Feedback

- Assignee changed from Luca Frosini to Roberto Cirillo

Looking the code, I didn't find any reason why accounting should open so many connections but I was looking the latest version.

I suggest to update the container and check what happen. If, the situation is not solved please please open me a bug and I'll further investigate the issue.

#8 - Jun 27, 2018 09:48 AM - Roberto Cirillo

- Assignee changed from Roberto Cirillo to Luca Frosini

Are you agree with my analysis? if not why there are a lot of accounting exception on the log in attachment? Have you checked the exceptions? I think we should start from this point.

I would also like to say that this is a very critical problem that affects the whole infrastructure, please take care of this.

I cannot do anything else: I've already said that on ckan-bb1 there is the latest smartgears distribution and the issue is reproduced here [#12034](#) as you already know.

Feel free to close this incident and open a bug but you should provide a better analysis asap.

@lucio.lelli@isti.cnr.it @pasquale.pagano@isti.cnr.it any opinion?

#9 - Jun 27, 2018 12:48 PM - Pasquale Pagano

What should I say? There is an important issue and the logs have highlighted a potential source of this problem. This has to be investigated and we need to understand why now this is becoming an issue while it was not a month ago.

#10 - Jun 27, 2018 02:08 PM - Luca Frosini

I investigated the issue and Accounting seems not querying the IS continuously as @roberto.cirillo@isti.cnr.it supposed.

Anyway I agree with Roberto to change the log level for such a class, so that we will have a confirm.

Talking with @lucio.lelii@isti.cnr.it we suppose that what is shown in the log is another issue happening when the IS is stopped and I solved it.

#11 - Jun 27, 2018 02:56 PM - Roberto Cirillo

After the restart of the containers identified yesterday, the situation seems to be returned to more normal levels. Of course the problem is not solved and other services could have the same problem.

#12 - Jun 27, 2018 03:02 PM - Roberto Cirillo

@luca.frosini@isti.cnr.it , @lucio.lelii@isti.cnr.it the log seems to be clear to me:

There were a lot of accounting threads that were connected to the IS when the IS-Collector was restarted. Are you agree with this or not?

#13 - Jun 27, 2018 04:55 PM - Luca Frosini

Roberto Cirillo wrote:

There were a lot of accounting threads that were connected to the IS when the IS-Collector was restarted. Are you agree with this or not?

No I don't. I'm still not convinced the thread are related to accounting. There were a lot of thread but you don't know which library or service is the responsible.

#14 - Jun 27, 2018 06:18 PM - Roberto Cirillo

OK you are still not convinced but I say that because all the threads in the logs are named "AccountingScheduledThread" so I guess the threads are related to accounting.

In particular, in the logs there are two kind of exceptions:

```
2018-06-25 18:05:04,357 [AccountingScheduledThread-33] ERROR PersistenceBackendConfiguration: InvocationTargetException -AccountingPersistenceConfiguration not initialized correctly. It will not be used. Trying the next one if any.
java.lang.RuntimeException: com.sun.xml.internal.ws.fault.ServerSOAPFaultException: Client received SOAP Fault from server: null Please see the server log to find more detail regarding exact cause of the failure.
    at org.gcube.resources.discovery.icclient.ICClient.submit(ICClient.java:47)
    at org.gcube.resources.discovery.client.impl.DelegateClient.submit(DelegateClient.java:50)
    at org.gcube.accounting.persistence.AccountingPersistenceConfiguration.getServiceEndpoint(AccountingPersistenceConfiguration.java:58)
    at org.gcube.accounting.persistence.AccountingPersistenceConfiguration.<init>(AccountingPersistenceConfiguration.java:41)
    at sun.reflect.GeneratedConstructorAccessor138.newInstance(Unknown Source)
    at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
    at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
    at org.gcube.documentstore.persistence.PersistenceBackendConfiguration.getInstance(PersistenceBackendConfiguration.java:35)
    at org.gcube.documentstore.records.aggregation.AggregationScheduler.getConfiguration(AggregationScheduler.java:268)
    at org.gcube.documentstore.records.aggregation.AggregationScheduler$ReloaderThread.run(AggregationScheduler.java:398)
    at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
    at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:308)
    at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$301(ScheduledThreadPoolExecutor.java:180)
    at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:294)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
Caused by: com.sun.xml.internal.ws.fault.ServerSOAPFaultException: Client received SOAP Fault from server: null Please see the server log to find more detail regarding exact cause of the failure.
    at com.sun.xml.internal.ws.fault.SOAP11Fault.getProtocolException(SOAP11Fault.java:178)
    at com.sun.xml.internal.ws.fault.SOAPFaultBuilder.createException(SOAPFaultBuilder.java:124)
    at com.sun.xml.internal.ws.client.sei.StubHandler.readResponse(StubHandler.java:238)
    at com.sun.xml.internal.ws.db.DatabindingImpl.deserializeResponse(DatabindingImpl.java:189)
    at com.sun.xml.internal.ws.db.DatabindingImpl.deserializeResponse(DatabindingImpl.java:276)
    at com.sun.xml.internal.ws.client.sei.SyncMethodHandler.invoke(SyncMethodHandler.java:104)
    at com.sun.xml.internal.ws.client.sei.SyncMethodHandler.invoke(SyncMethodHandler.java:77)
    at com.sun.xml.internal.ws.client.sei.SEIStub.invoke(SEIStub.java:147)
    at com.sun.proxy.$Proxy46.execute(Unknown Source)
    at sun.reflect.GeneratedMethodAccessor77.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
```

```

at java.lang.reflect.Method.invoke(Method.java:498)
at org.gcube.common.clients.stubs.jaxws.proxies.MethodRetriever.invoke(MethodRetriever.java:25)
at com.sun.proxy.$Proxy47.execute(Unknown Source)
at org.gcube.resources.discovery.icclient.ICClient.callService(ICClient.java:63)
at org.gcube.resources.discovery.icclient.ICClient.submit(ICClient.java:40)
... 16 common frames omitted

```

2018-06-25 18:07:16,862 [AccountingScheduledThread-26] ERROR PersistenceBackendConfiguration: InvocationTargetException -AccountingPersistenceConfiguration not initialized correctly. It will not be used. Trying the next one if any.

```

javax.xml.ws.WebServiceException: java.net.ConnectException: Connection refused (Connection refused)
at com.sun.xml.internal.ws.transport.http.client.HttpClientTransport.readResponseCodeAndMessage(HttpClientTransport.java:195)
at com.sun.xml.internal.ws.transport.http.client.HttpTransportPipe.createResponsePacket(HttpTransportPipe.java:226)
at com.sun.xml.internal.ws.transport.http.client.HttpTransportPipe.process(HttpTransportPipe.java:217)
at com.sun.xml.internal.ws.transport.http.client.HttpTransportPipe.processRequest(HttpTransportPipe.java:130)

```

```

at com.sun.xml.internal.ws.transport.DeferredTransportPipe.processRequest(DeferredTransportPipe.java:124)
at com.sun.xml.internal.ws.api.pipe.Fiber.__doRun(Fiber.java:1121)
at com.sun.xml.internal.ws.api.pipe.Fiber._doRun(Fiber.java:1035)
at com.sun.xml.internal.ws.api.pipe.Fiber.doRun(Fiber.java:1004)
at com.sun.xml.internal.ws.api.pipe.Fiber.runSync(Fiber.java:862)
at com.sun.xml.internal.ws.client.Stub.process(Stub.java:448)
at com.sun.xml.internal.ws.client.sei.SEIStub.doProcess(SEIStub.java:178)
at com.sun.xml.internal.ws.client.sei.SyncMethodHandler.invoke(SyncMethodHandler.java:93)
at com.sun.xml.internal.ws.client.sei.SyncMethodHandler.invoke(SyncMethodHandler.java:77)
at com.sun.xml.internal.ws.client.sei.SEIStub.invoke(SEIStub.java:147)
at com.sun.proxy.$Proxy46.execute(Unknown Source)
at sun.reflect.GeneratedMethodAccessor77.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at org.gcube.common.clients.stubs.jaxws.proxies.MethodRetriever.invoke(MethodRetriever.java:25)
at com.sun.proxy.$Proxy47.execute(Unknown Source)
at org.gcube.resources.discovery.icclient.ICClient.callService(ICClient.java:63)
at org.gcube.resources.discovery.icclient.ICClient.submit(ICClient.java:40)
at org.gcube.resources.discovery.client.impl.DelegateClient.submit(DelegateClient.java:50)
at org.gcube.accounting.persistence.AccountingPersistenceConfiguration.getServiceEndpoint(AccountingPersistenceConfiguration.java:58)
at org.gcube.accounting.persistence.AccountingPersistenceConfiguration.<init>(AccountingPersistenceConfiguration.java:41)
at sun.reflect.GeneratedConstructorAccessor138.newInstance(Unknown Source)
at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
at org.gcube.documentstore.persistence.PersistenceBackendConfiguration.getInstance(PersistenceBackendConfiguration.java:35)
at org.gcube.documentstore.records.aggregation.AggregationScheduler.getConfiguration(AggregationScheduler.java:268)
at org.gcube.documentstore.records.aggregation.AggregationScheduler$ReloaderThread.run(AggregationScheduler.java:398)
at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:308)
at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$301(ScheduledThreadPoolExecutor.java:180)
at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:294)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)

```

Caused by: java.net.ConnectException: Connection refused (Connection refused)

```

at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
at java.net.Socket.connect(Socket.java:589)
at java.net.Socket.connect(Socket.java:538)
at sun.net.NetworkClient.doConnect(NetworkClient.java:180)
at sun.net.www.http.HttpClient.openServer(HttpClient.java:463)
at sun.net.www.http.HttpClient.openServer(HttpClient.java:558)
at sun.net.www.http.HttpClient.parseHTTPHeader(HttpClient.java:840)
at sun.net.www.http.HttpClient.parseHTTP(HttpClient.java:678)
at sun.net.www.protocol.http.HttpURLConnection.getInputStream0(HttpURLConnection.java:1587)
at sun.net.www.protocol.http.HttpURLConnection.getInputStream(HttpURLConnection.java:1492)
at java.net.HttpURLConnection.getResponseCode(HttpURLConnection.java:480)

```

```
at com.sun.xml.internal.ws.transport.http.client.HttpClientTransport.readResponseCodeAndMessage(HttpClientTransport.java:191)
... 37 common frames omitted
```

IMHO the first exception is related to an active connection that has been broken when the IS-Collector service has been restarted:

```
Caused by: com.sun.xml.internal.ws.fault.ServerSOAPFaultException: Client received SOAP Fault from server: null
Please see the server log to find more detail regarding exact cause of the failure
```

the second one is related to an attempt to connect to the IS-Collector service but the IS-Collector service was down (Connection Refused):

```
Caused by: java.net.ConnectException: Connection refused (Connection refused)
```

Of course, this is only my opinion based only on the log files.
If you are agree, I'm going to close this ticket and restart each service (with accounting log level set to TRACE) that has an high number of connections opened to the IS-Collector, in order to highlight better the problem.

#15 - Jun 28, 2018 09:57 AM - Roberto Cirillo
- Related to Task #12058: IS-Collector: Restart all the smartgears services with too many open connections to IS-Collector added

#16 - Jun 28, 2018 09:57 AM - Roberto Cirillo
- Tracker changed from Incident to Task

#18 - Jul 27, 2018 11:40 AM - Luca Frosini
- Related to Support #12254: geoserver-french-tunaatlas: strange accounting logs and high load added

#19 - Nov 02, 2018 11:01 AM - Luca Frosini
- Status changed from Feedback to Closed
- % Done changed from 50 to 100

Files			
rstudio2-ghn.2018-06-25.0.log	740 KB	Jun 26, 2018	Roberto Cirillo
ckan-bb1-ghn.2018-06-25 (cut).log	4.01 MB	Jun 26, 2018	Roberto Cirillo