



This repository ▾

Search or type a command



Explore Gist Blog Help



aphyr + ▾



JC



elasticsearch / elasticsearch

Watch ▾

707

★ Star

6,621

Fork

1,761

split brain condition after second network disconnect - even with minimum_master_nodes set #2117

New issue

Open OwenButler opened this issue 2 years ago · 16 comments



OwenButler commented 2 years ago

Summary:

Split brain can occur on the second network disconnect of a node, when the `minimum_master_nodes` is configured correctly($n/2+1$). The split brain occurs if the `nodeId(UUID)` of the disconnected node is such that the disconnected node picks itself as the next logical master while pinging the other nodes(`NodeFaultDetection`). The split brain only occurs on the second time that the node is disconnected/isolated.

Detail:

Using `ZenDiscovery`, Node Id's are randomly generated(A UUID): `ZenDiscovery:169`.

When the node is disconnected/isolated it the `ElectMasterService` uses an ordered list of the Nodes (Ordered by `nodeId`) to determine a new potential master. It picks the first of the ordered list:
`ElectMasterService:95`

Because the `nodeId`'s are random, it's possible for the disconnected/isolated node to be first in the ordered list, electing itself as a possible master.

The first time network is disconnected, the `minimum_master_nodes` property is honored and the disconnected/isolated node goes into a "ping" mode, where it simply tries to ping for other nodes. Once the network is re-connected, the node re-joins the cluster successfully.

The Second time the network is disconnected, the `minimum_master_nodes` intent is not honored. The disconnected/isolated node fails to realise that it's not connected to the remaining node in the 3 node cluster and elects itself as master, still thinking it's connected.

It feels like there is a failure in the transition between `MasterFaultDetection` and `NodeFaultDetection`, because it works the first time!

The fault only occurs if the `nodeId` is ordered such that the disconnected node picks itself as the master while isolated. If the `nodeId`'s are ordered such that it picks one of the other 2 nodes to be potential master then the isolated node honors the `minimum_master_nodes` intent every time.

Because the `nodeId`'s are randomly(UUID) generated, the probability of this occurring drops as the number of nodes in the cluster goes up. For our 3 node cluster it's ~50% (with one node detected as gone, it's up to the ordering of the remaining two `nodeId`'s)

Note, While we were trying track this down we found that the `cluster.service TRACE` level logging (which outputs the cluster state) does not list the nodes in election order. IE, the first node in that printed list is not necessarily going to be elected as master by the isolated node.

Detail Steps to reproduce:

Labels

bug

Milestone

No milestone

Assignee



kimchy

Notifications

Subscribe

You'll be notified when you participate or are @mentioned in this thread.

10 participants



Because the ordering of the nodeid's is random(UUID) we were having trouble getting a consistently reproducible test case. To fix the ordering, we made a patch to ZenDiscovery to allow us to optionally configure a nodeid. This allowed us to set the nodeid of the disconnected/isolated node to guarantee it's ordering, allowing us to consistently reproduce.

We've tested this scenario on the 0.19.4, 0.19.7, 0.19.8 distributions and see the error when the nodeid's were ordered just right.

We also tested this scenario on the current git master with the supplied patch.

In this scenario, node3 will be the node we disconnect/isolate. So we start the nodes up in numerical order to ensure node3 doesn't *start* as master.

1. Configure nodes with attached configs (one is provided for each node)
2. Start up nodes 1 and 2. After they are attached and one is master, start node 3
3. Create a blank index with default shard/replica(5/1) settings
4. Pull network cable from node 3
5. Node 3 detects master has gone (MasterFaultDetection)
6. Node 3 elects itself as master (Because the nodeid's are ordered just right)
7. Node 3 detects the remaining node has gone, enters ZenDiscovery minimum_master_nodes mode, prints a message indicating not enough nodes
8. Node 3 goes into a ping state looking for nodes
9. At this point, node 1 and node 2 report a valid cluster, they know about each other but not about node 3.
10. Reconnect network to node 3
11. Node 3 rejoins the cluster correctly, seeing that there is already a master in the cluster.

At this point, everything is working as expected.

1. Pull network cable from node 3 again
2. Node 3 detects master has gone (MasterFaultDetection)
3. Node 3 elects as itself as master (Because the nodeid's are ordered just right)
4. Node 3 now fails to detect that the remaining node in the cluster is not accessible. It starts throwing a number of Netty NoRouteToHostException about the remaining node.
5. According to node 3, cluster health is yellow and cluster state shows 2 data nodes
6. Reconnect network to node 3
7. Node 3 appears to connect to the node that it thinks it's still connected to. (can see that via the cluster state api). The other nodes log nothing and do not show the disconnected node as connected in any way.
8. Node 3 at this point accepts indexing and search requests, a classic split brain.

Here's a gist with the patch to ZenDiscovery and the 3 node configs.

<https://gist.github.com/3174651>



kimchy commented 2 years ago

Owner

Thanks for the detailed explanation, do you have the logs for the 3 nodes around, would love to have a look at them.



OwenButler commented 2 years ago

Hi Shay,

Logs for the three nodes here:

<https://gist.github.com/3223822>

The disconnected/isolated node is the top file, log named "splitbrain-isolatednode.log".

Timestamps of note (the clocks of the 3 nodes are within a second of each other):

14:42:49 -> first network disconnect

14:44:30 -> Reconnect

14:45:13 -> Second network disconnect

14:46:11 -> Split brain begins (the isolated node still thinks it's connected to one of the others at this point)

14:47:41 -> After second reconnect the isolated node now just sees one node.

There's a few index status request errors logged because we used elasticsearch-head to check status on the isolated node.



praveenbm5 commented 2 years ago

Looks like we are facing a similar issue...

```
[2012-08-28 06:54:20,729][INFO ][discovery.zen ] [TES3] master_left [[TES1][-G0NH7iwRQevY3La-zxaA]
[inet[/178.238.237.241:9300]]], reason [failed to ping, tried [3] times, each with maximum [30s] timeout]
[2012-08-28 06:54:20,741][INFO ][cluster.service ] [TES3] master {new [TES3]
[DlhQQmanQOOgo1qVFz8tSA][inet[/178.238.237.240:9300]], previous [TES1][-G0NH7iwRQevY3La-
zxaA][inet[/178.238.237.241:9300]]}, removed {[TES1][-G0NH7iwRQevY3La-zxaA]
[inet[/178.238.237.241:9300]]}, reason: zen-disco-master_failed ([TES1][-G0NH7iwRQevY3La-zxaA]
[inet[/178.238.237.241:9300]])
[2012-08-28 06:54:50,707][INFO ][cluster.service ] [TES3] added {[TES1][-G0NH7iwRQevY3La-zxaA]
[inet[/178.238.237.241:9300]]}, reason: zen-disco-receive(join from node[[TES1][-G0NH7iwRQevY3La-
zxaA][inet[/178.238.237.241:9300]])
[2012-08-28 07:06:03,238][INFO ][cluster.service ] [TES3] removed {[TES2][kt7r8dFxSt6bjVKUAvdcg]
[inet[/178.238.237.239:9300]]}, reason: zen-disco-node_failed([TES2][kt7r8dFxSt6bjVKUAvdcg]
[inet[/178.238.237.239:9300]]), reason failed to ping, tried [3] times, each with maximum [30s] timeout
[2012-08-28 07:06:03,278][WARN ][discovery.zen ] [TES3] not enough master nodes, current nodes:
{[TES3][DlhQQmanQOOgo1qVFz8tSA][inet[/178.238.237.240:9300]],}
[2012-08-28 07:06:03,279][INFO ][cluster.service ] [TES3] removed {[TES1][-G0NH7iwRQevY3La-zxaA]
[inet[/178.238.237.241:9300]]}, reason: zen-disco-node_failed([TES1][-G0NH7iwRQevY3La-zxaA]
[inet[/178.238.237.241:9300]]), reason failed to ping, tried [3] times, each with maximum [30s] timeout
[2012-08-28 07:06:33,290][INFO ][cluster.service ] [TES3] detected_master [TES1][-G0NH7iwRQevY3La-
zxaA][inet[/178.238.237.241:9300]], added {[TES1][-G0NH7iwRQevY3La-zxaA]
[inet[/178.238.237.241:9300]], [TES2][kt7r8dFxSt6bjVKUAvdcg][inet[/178.238.237.239:9300]]}, reason:
zen-disco-receive(from master [[TES1][-G0NH7iwRQevY3La-zxaA][inet[/178.238.237.241:9300]])
[2012-08-28 08:36:45,100][INFO ][discovery.zen ] [TES3] master_left [[TES1][-G0NH7iwRQevY3La-zxaA]
[inet[/178.238.237.241:9300]]], reason [failed to ping, tried [3] times, each with maximum [30s] timeout]
[2012-08-28 08:36:45,112][INFO ][cluster.service ] [TES3] master {new [TES3]
[DlhQQmanQOOgo1qVFz8tSA][inet[/178.238.237.240:9300]], previous [TES1][-G0NH7iwRQevY3La-
zxaA][inet[/178.238.237.241:9300]]}, removed {[TES1][-G0NH7iwRQevY3La-zxaA]
[inet[/178.238.237.241:9300]]}, reason: zen-disco-master_failed ([TES1][-G0NH7iwRQevY3La-zxaA]
[inet[/178.238.237.241:9300]])
```

tallpsmith referenced this issue a year ago

minimum_master_nodes does not prevent split-brain if splits are intersecting
#2488

Open



tallpsmith commented a year ago

@kimchy et al have you had a chance to try to reproduce this with the steps outlined above? I see a fair number of Split Brain style issues appearing on the mailing lists, and likely a smaller cluster size for them is exacerbating this issue?



tallpsmith commented a year ago

@kimchy @s1monw anyone able to comment on whether they've even tried the steps above ? I still think people are vulnerable to this condition.

It is a pain to setup the test I know, but trust me, once you see it happen, it's bad.



brusic commented a year ago

I have been watching this thread for a while since I have encountered the issue as well (running 0.20RC1). I tend to avoid adding a +1 comment on GitHub, but I wanted to lend my support to this issue.

Another related problem this week: <https://groups.google.com/forum/?fromgroups=#!topic/elasticsearch/erpa7mMT5DM>

My hopes is that the new field cache changes will alleviate memory pressure, causing fewer garbage collection (which might have been our issue as well).



s1monw commented a year ago

Owner

@tallpsmith thanks for pinging me on this one again. I started looking into this today but it might take until next week to get some results / comments on this. I haven't had this issue on my radar so it's great that you pinged again!



synhershko commented a year ago

radar blip



s1monw commented a year ago

Owner

hey folks,
sorry to come back to this so late. I have worked out a setup where I can reproduce this issue. Unfortunately, this situation can in-fact occur with zen discovery at this point. We are working on a fix for this issue which might take a bit until we have something that can bring a solid solution for this.



jprante commented 11 months ago

Maybe it should be possible to add an alternative (selectable) Leader Election algorithm? For example, Chang-Roberts or Hirschfeld-Sinclair?



brusic commented 11 months ago

Our live cluster experienced the split-brain scenario once again after a node was unresponsive during garbage collection. Plenty of logs if wanted.



synhershko commented 11 months ago

Ivan, how do you connect it to the cluster again - simple restart to that node?



brusic commented 11 months ago

Yes. I didn't have time to identify the problematic node, so I restarted the node I thought was having issues (I choose correctly, I guess I should have played PowerBall after all). One shard (on a different node) was stuck in a RECOVERING state, which I fixed by reducing the number of replicas from 2 to 1 and then increasing the number again.

Simon mentioned that issue is with zen discovery, so I am assuming switching from Multicast to Unicast will not alleviate the problem.



brusic commented 9 months ago

Have I mentioned how serious this problem is? Our production cluster has pretty much gone away. It is like a game of whackomole trying to kill instances that think they are the master.



fasher commented 9 months ago

I agree, this issue is critical please give this priority.
We gone to a single master node in our production to avoid split brains we had in the past that corrupted our index.



 **anandnalya** referenced this issue 7 months ago

NodeDoesNotExistOnMasterException Handling #3663

 Open



bitsofinfo commented 3 months ago

Any update on a timeline for a fix for this?



Write

Preview

Comments are parsed with [GitHub Flavored Markdown](#)

Leave a comment

Attach images by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Comment

