After-Action Report of the 2019-02-21 Factom Network Stall

On Feb 21, 2019 the factom network stopped forward progress of the blockchain for 7 hours. After 3 tries and with much community help, the blockchain started progressing again.

Root cause: Poisoned process lists from a brain swap.

A brain swap bug during an upgrade from 6.1.0 to 6.2.0 with one of the Federated servers, or a mis-configuration.

Brain swapping is how the factom network Federated server nodes, which build the blockchain, are able to upgrade software without the network stopping. It allows a single entity running two computers to seamlessly transition from one computer to the other. Typically the swap is from an older software version to a newer software version.

There are two situations where the transition is not seamless. If one node stops before the other one picks up, then the network should handle this in a safe fashion, and will promote an audit server to take over from the missing Federated one.

The more serious failure is when there is overlap between two different machines with the same identity. Both machines will attempt to build the blockchain at the same time. They will both provide messages that fill in the process list for the specific VM they are responsible for. They will send out these messages over the p2p network and different federated servers run by other ANOs will see messages at the same process list height from either the first or second brainswapped node. The other Federated servers which received process list messages from the second brainswapped server would have a poisoned process list (assuming the second one got it’s messages to a minority of the Federated server set). They would be unable to continue building the process list and the majority of other Federated servers, who saw the first brainswapped machine, would fault out the poisoned minority. If there aren’t enough Audit servers to fill in for the poisoned minority (or the faulting elections fail for another reason), then the network will stall.

This is why it so important to **set** the **ChangeAcksHeight** to a point that is **several blocks** in the **future**. The risk of hosting two identical Federated servers **risk** the **whole network not just your own node**.

This being the case, there are also known bugs that have been plaguing the testnet for a couple weeks. The aborted 6.1.1 release would activate brain swaps at different points than the 6.1.0 or 6.2.0 code.

Matt York and Clay have been shaking their head at the brainswapping code over the past few weeks. They have been debugging the issues found on the testnet.

https://github.com/FactomProject/factomd/pull/653

There is a chance that there was an intermittent timing bug in the 6.1.0 / 6.2.0 code. That is getting updated in the upcoming Kraft release (anticipated 6.2.1). Lots of unit tests have been generated to demonstrate and debug the brainswapping errors.

<https://github.com/FactomProject/factomd/tree/640010229965ead656ac33b5a2d3038da477238c/simTest>

-----

The stall began just after 2:30 pm UTC (8:30 am Texas time)

Some leaders had stopped at 180790 and some at 180789. The minority were are 180789.

Some people had upgraded from 6.1.0 to 6.2.0, the two codebases are identical except for grants which will activate at 181001. There was no correlation between the software version and the height of the blockchain node.

There was a mix of different ANOs that had servers which were behind:

1x TFA, 1x DBgrow, 1x luciapp, 1x mof, 1x layertech, 1x go immutable, 1x blockrock, 2x defacto, 2x prestigeIT, 1x BIF

Mike Brennan had been updating the restart script over the past few weeks, had only tested it on a small custom network. Some of the updates weren’t compatible given the diversity of mainnet.

----

16:00 (10 AM texas time)

After debugging restart script, it was run. It couldn't reach a number of servers. Some of them had missed being restarted. After 3 debugging attempts, the script successfully executed at 16:12 UTC. It turned out that only some of the nodes had restarted. These ANOs had at least one node that didn’t restart with the script: building IM, cryptologic, canonical ledgers, factomiatic, defacto, multicoin.

Jay Cheroske from Bedrock suggested a separated stop then start. This was opted against, since authorship and debugging would take too much time.

-----

17:40 (11:40 Texas time)

A second restart was attempted, with manual coordination attempted with the ANOs listed above. All were available but Multicoin, but their nodes could be restarted via the portainer system (just not with the script).

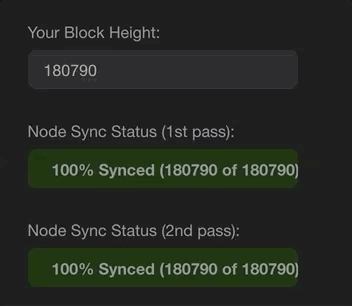
On the second restart attempt the script collected the container IDs and saved them to a generalized log file, but had an issue with logging to a separate specific log file. The script stopped before it proceeded to the start phase. This precipitated the need to implement Jay Cheroske’s idea to split the restart into two phases. It took a while, as predicted, to reformulate the script to accept a list of container IDs and hosts, as well as merging the lists and extracting the relevant data from the generalized log. There were 145 different nodes that the restart script was reaching out to.

Some ANOs took this opportunity to move Identities that were on nodes at block 180789 to follower nodes they had at 180790.

At 19:30 a coordinated start was executed, using the newly updated script and manual help from ANOs whose servers were lacking connectivity.

Examination of the control panels showed that all the nodes had started in an appropriate time.

Several of the Federated servers had started up with a flapping second pass. When a factomd node has not completed the 2nd pass, it cannot follow along with minutes, and if a leader cannot follow with minutes then it cannot act as a Federated or Audit server. This indicated that there were entries in the last block (180790) that factomd did not recognize that it had saved already.



(thanks to Alex at Factoshi for creating this image)

After a few minutes most of the Federated nodes that were showing this behavior stopped flapping. They did not fix themselves fast enough to successfully participate in the faulting elections, and the network stalled again. There were 8 missing DBsigs in the process list, but faulting was unable to fill in the missing 8 slots.

An hour after the start commenced, it was declared a failure at 20:30 UTC. A second restart attempt was planned. The stop portion of the script was run at 20:46. Other ANOs such as RewardChain and PrestigeIT used this opportunity to switch their leader identities from nodes stuck on 180789 with ones on 180790.

----

20:55

A third coordinated restart attempt was executed with the restart script and with building IM, cryptologic, canonical ledgers, factomiatic, and defacto.

After the Federated servers came out of wait mode and ignore mode, the network was missing 5 DBsigs in the process list. This meant that 5 audit servers needed to be faulted into being Federated servers.

At 21:30 (3:30 pm Texas time), the network successfully faulted out the federated servers which did not provide the 5 DBsigs. At 22:00 UTC the incident was declared over as the blockchain was progressing forward.

It was recommended that anyone who was running a server which had been faulted out (or is an audit server) should restart to correct the malformed internal authority set which might still exist in the memory.

-------

Future correction steps:

* Work with ANOs to get back connectivity for the nodes
  + This would include a network inventory to make sure ports are open and the APIs are accessible
  + Make sure that backup nodes are accessible as well
* Create a new readiness script which can be run that can check connectivity without restarting the network to identify problems ahead of time
* Confirm tags and container IDs work in the restart script along with general script improvements.

-----

Note:

This 7 hour stall has pushed back the grant activation time from approx 11:59 pm UTC on 2019-02-22 until 7:00 2019-02-23.

Addendum:

Tom from Layertech is working on fixing a similar brainswapping bug between an Audit server and a follower. This type of swap can cause a panic when the follower advances to the next block and activates as an Audit, but the existing Audit server had not yet advanced from minute 9 to the next block. They would both be sending out heartbeats at the same time and one of them would panic due to self protection mechanisms.

<https://circleci.com/gh/FactomProject/factomd/9636?utm_campaign=vcs-integration-link&utm_medium=referral&utm_source=github-build-link>