Liveliness Testing | REPORT

Partner Chains

August 2025

Version: 1.0

Supplied by: IOJP & IOHK

Author: Skylar Simoncelli & Nikolaos Dymitriadis

Document Information

Version Control			
Version Num.	Date	Summary of Changes	Updated By

Table of Contents

Overview

This report summarizes a series of liveliness tests conducted on a simulated Partner Chain network environment

The environment was spawn with our Docker local-environment-dynamic tool:
https://github.com/input-output-hk/partner-chains/tree/master/dev/local-environment-dynamic

This tool automated the full end-to-end setup of a Partner Chain with customisable number of db-syncs, permissioned validator nodes and registered validator nodes

The environment was run on a single AWS instance with 192 CPU + 382 GB memory, with all validators communicating over localhost through Docker networking

Test 1

In this first round of testing we wanted to find a maximum number of nodes that we could run with this local-environment-dynamic tool for the resources available on this AWS instance.

When deploying more than 300 permissioned validators, we hit a **maxTxSize** limit in the Cardano mainchain shelley genesis configuration when funding all addresses in a single mainchain transaction. We increased this to a much larger size so it would not be hit again

When deploying more than 500 permissioned validators when then hit systemd limitations caused by Docker trying to start 500+ containers at the same time, which we worked around by batching the containers into groups of 25 and staggering the starts with Docker's **depends_on** function

When deploying more than 800 permissioned validators we hit a **maxTxSize** equivalent in the partner-chains-node registration commands processing this many keys at once

The highest number of concurrent peers we ever saw on any single node was 165. We did not have any limits set on the maximum number of peers for each node, and so this limitation was possibly caused by Docker networking unable to handle more subsequent P2P connections between nodes

Test 2

In this second round of testing we spawned 100 permissioned validators and 10 db-sync + postgres stacks (all reading from the same cardano node.socket), and D parameter set to 100:0. The 100 permissioned validators were round-robin distributed between the 10 postgres nodes:

node-1 > db-sync-1
node-2 > db-sync-2
node-3 > db-sync-3
…
node-11 > db-sync-1
node-12 > db-sync-2
node-13 > db-sync-3
etc…

We let the chain produce 1000 blocks with all 10 db-syncs running, and performed the following test process:

block	action
0	chain init with 10 db-syncs
1000	shutdown db-sync-1
1500	shutdown db-sync-2
2000	shutdown db-sync-3
2500	shutdown db-sync-4
3000	started db-sync-4
3500	started db-sync-3
5882	started db-sync-1 and db-sync-2

After shutting down the first db-sync (**docker stop db-sync-1**), block production slowed as some finalisations were failing from the committee seats out of sync, but blocks were still finalising quickly

From block 1500 the chain was still finalising blocks just a bit slower again

From block 2000 we noticed block production was very slow as finalisations were failing, but blocks were still finalising as less than 33% of the committee are down

From block 2500 we stopped seeing blocks finalising completely, as more than 33% of the committee is down

At block 3000 we resumed db-sync-4 (**docker start db-sync-4**) with no additional steps taken, and blocks began finalising again

At block 3500 we resumed db-sync-3 and finalisation sped up. At 5882 we resumed db-sync-1 and db-sync-2 and the chain was fully back to original block production rate

Test 3

In this test we deployed 50 permissioned nodes and 50 registered nodes with a single shared db-sync + postgres, and conducted tests of D parameter switching from fully permissioned to fully registered

We completed these changes at the below blocks:

Block	Action
0	**D Parameter** 50:0
414	**D Parameter** changed from 50:0 to 35:15
790	**D Parameter** changed from 35:15 to 25:25
1267	**D Parameter** changed from 25:25 to 15:35
1738	**D Parameter** changed from 15:35 to 0:50

Each change to **D Parameter** takes 2 epochs to take effect (40 blocks), so our theoretical expected results were:

Block Range	D Parameter	Expectation
0-454	50:0	100% blocks produced by permissioned validators
454-820	35:15	57% permissioned, 43% blocks registered
820-1307	25:25	50% permissioned, 50% blocks registered
1307-1778	15:35	43% permissioned, 57% blocks registered
1778+	0:50	100% blocks produced by registered validators

Our test results matched this closely within acceptable limits. Had we left more time between D Parameter changes, we expect these percentages would trend further towards our expected percentages:

Block Range	Permissioned	Registered
0-454	100.00%	0.00%
454-820	67.76%	32.24%
820-1307	51.75%	48.25%
1307-1778	31.21%	68.79%
1778+	0.00%	100.00%