

# Shiden collators analysis

block production delays

bLd Nodes 06/09/2021

#### Introduction

Since the beginning of Shiden as a parachain, we noticed some delays in the block production time, sometimes with a very significant gap to the target time of 12 seconds/block.

This document aims to analyze block production on Shiden based on chain data from August 20th to September 6th.

#### Description

Different axis of analysis have been brought for this first version of data analysis, we tried to focus mainly on the following metrics:

- Overall block production time and extreme values
- Blocks produced by collator

In this time frame, the **number of collators** has been increased **from 2 to 29**.

## **Global block production time**

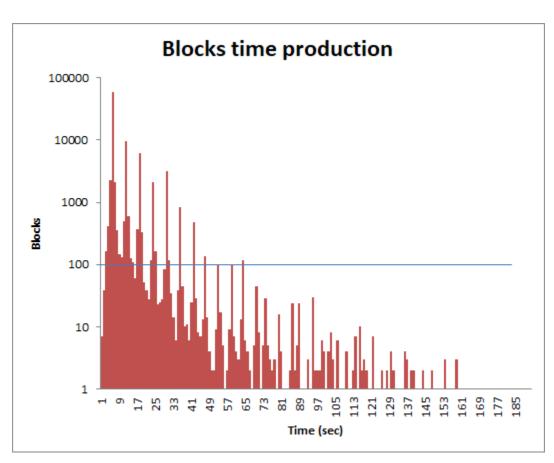
Below are 3 different timeframes of analysis showing the average block production time on Shiden. We can notice that adding new collators tends to reduce the average time of

- 16.5 seconds starting from 2 collators
- 15.2 seconds starting from 10 collators
- 14.6 seconds starting from 23 collators

Despite adding collators, the average block production time is still **2.6 seconds** higher than expected, representing **21.66**% more than expected.

## From august 20th (2 collators)

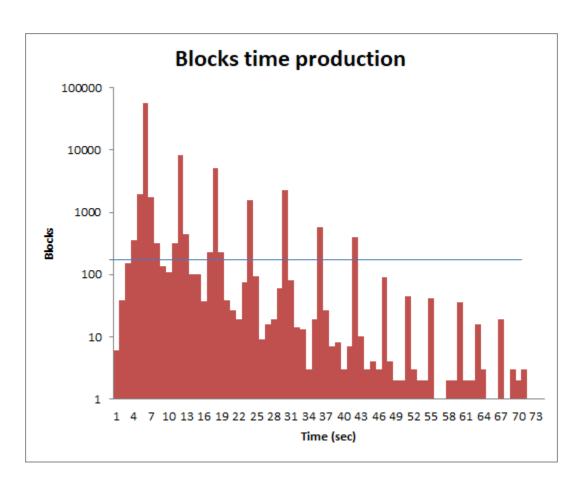
time	Nb blocks	%
< 15 seconds	65087	71,0%
15-30 seconds	20373	22,2%
31-60 seconds	5351	5,8%
> 60 seconds	398	0,4%
Total blocks		91 609
Average block	16,52	



<sup>\*</sup> total blocks have a logarithmic scale

## From august 23th (10 collators)

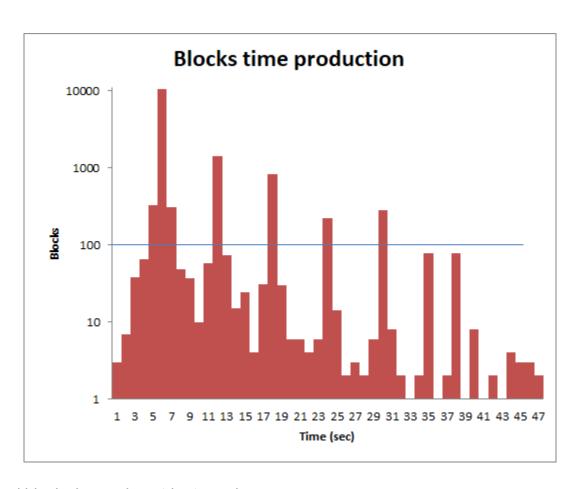
time	Nb blocks	%
< 15 seconds	61762	74,9%
15-30 seconds	16837	20,4%
31-60 seconds	3701	4,5%
> 60 seconds	159	0,2%
Total blocks		82 490
Average block	15,20	



<sup>\*</sup> total blocks have a logarithmic scale

## From sept 4th (23 collators)

time	Nb blocks	%
< 15 seconds	11551	77,8%
15-30 seconds	2790	18,8%
31-60 seconds	509	3,4%
> 60 seconds	0	0,0%
Total blocks		14 850
Average block	14,63	



<sup>\*</sup> total blocks have a logarithmic scale

## **Block production per collator**

The data below represents the number of blocks produced per day per collator.

Collator	Node location	08-20	08-21	08-22	08-23	08-24	08-25	08-26	08-27	08-28	08-29	08-30	08-31	09-01	09-02	09-03	09-04	09-05	09-06	Total
Alex PromoTeam	Germany				512	559	514	502	494	397	341	340	341	276	250	259	248	232	112	5377
akru	Finland	609	1206	1072	678	564	519	512	501	395	349	336	320	253	248	257	237	217	118	8391
bLd Nodes   ValidatorAlliance	USA					148	374	354	354	277	246	244	222	140	207	244	235	214	116	3375
Bware Labs	Germany																176	236	123	535
cp0x.com-CP287	USA									215	325	306	298	237	234	234	217	213	107	2386
HuobiPool														135	169	178	157	157	81	877
inchainworks	USA		296	847	567	462	420	432	439	335	289	275	276	199	204	167	204	167	105	5684
lemonode	Spain																75	218	110	403
luckyve	Finland				479	556	495	486	495	380	351	322	311	256	253	252	240	225	117	5218
Maarten	Netherlands		440	1090	695	579	508	504	504	399	354	345	351	274	249	260	247	231	117	7147
Masternode24♥	Germany					195	534	521	511	403	188	346	336	276	256	261	247	233	122	4429
Mile	Finland									239	356	345	334	270	242	228	205	228	118	2565
P2P.ORG																	111	239	119	469
pathrocknetwork	USA									208	185	277	281	196	191	175	189	192	96	1990
Pinknode Shiden Collator 1																		111	103	214
Polkadotters   VALIDATORALL	Germany				60	542	509	495	493	388	331	334	326	242	232	246	234	221	117	4770
Shad-Astar										245	351	339	319	263	250	250	245	228	123	2613
SHIDEN_NODE_1	Ireland													213	259	253	245	231	117	1318
space-needle	USA (west coas	it)												161	196	177	171	160	89	954
Stake Works										252	348	325	324	260	256	252	233	226	115	2591
STAKECRAFT										235	346	333	314	257	251	248	243	218	119	2564
STAKEF1X														189	245	234	215	214	113	1210
STAKENODE   VALIDATORALLIA	ANCE			302	418	429	284	428	231	400	345	338	316	269	265	262	240	228	122	4877
StakeNodes														213	263	257	252	241	116	1342
STCollator 01	Japan	576	448	482	348	312	260	227	195	162	150	134	118	128	130	134	110	114	54	4082
STCollator 02	Japan	712	526	512	387	333	288	46	0	126	169	147	145	131	109	129	115	111	57	4043
swasilenko												144	330	272	166	267	250	231	125	<b>178</b> 5
Vlady Limes	Germany				504	560	503	503	501	388	347	349	339	264	256	233	108	232	121	5208
⊜ceberg Nodes ⊜ C1	Germany				333	571	509	516	515	392	349	329	324	268	249	255	244	219	119	5192
Total Blocks		1897	2916	4305	4981	5810	5717	5526	5233	5836	5720	5908	5925	5642	5630	5712	5693	5987	3171	91609
Average block / validator		632	583	718	453	447	440	425	403	307	301	295	296	226	225	228	203	206	109	3159

The spreadsheet suggests that the node location plays a significant role in the node performance, nodes based in Europe have a much higher rate than the ones in America and Asia.

#### **Practical case**

Raw data available here: Shiden block data

The Shiden parachain is actually running under Aura PoS consensus: the collators are assigned in a sequential order for block production. This means that the collators should all produce the same amount of blocks.

In practice, we can notice that validators are missing some blocks, resulting in "skip" of the communication slot with the validator. As a result, the block is assigned to the next collator and the block production time is doubled.



In the example above, blocks 287240 and 287241 have been skipped once, produced in 24 seconds instead of 12.



Other (rarer) cases show a block can take up to 1 minute to be produced, having been skipped 5 times.

#### Unproduced block logs

On the collator side, the consensus process is well started, no error in thrown in the log.

The following file contains the full debug log from a collator who was supposed to produce the block number 215698 but failed:

https://drive.google.com/file/d/1IBGD-0kzz5X29KQBR0P4FG547esjgkvN/view?usp=sharing

On the relaychain validator side assigned to the Shiden parachain, here is a short log where we can notice several errors "Fetching collation failed due to network error" - "err=Network(Timeout)":

https://drive.google.com/file/d/10H74IZcPA2IIblXun3M-CCGlbl8wkrn2/view?usp=sharing