

10.6.2 against 10.5.4

value-only workload

Michael Karg, Cardano Performance & Tracing

2026-02-16

Contents

Manifest	2
Analysis	4
Resource Usage	4
Anomaly control	4
Forging	5
Individual peer propagation	5
End-to-end propagation	5
Appendix A: charts	6
Cluster performance charts	6
Appendix B: data dictionary	21
Block propagation metrics	21
Cluster performance metrics	22

Manifest

We compare 10.5.4 (Conway) and 10.6.2 (Conway) relative to 10.5.4 (Conway), under value-only workload.

	10.5.4	10.6.2
Analysis date	2026-02-06	2026-02-13
Cluster system start date	2026-02-05	2026-02-12
Cluster system start time	18:52:16	10:38:55
Identifier	10.5.4	10.6.2
Run batch	1054	10.6.2
GHC version	9.6.5	9.6.7
cardano-node version	10.5.4	10.6.2
ouroboros-consensus version	0.27.0.0	0.30.0.1
ouroboros-network version	0.21.6.1	0.22.6.0
cardano-ledger-core version	1.17.0.0	1.18.0.0
plutus-core version	1.45.0.0	1.57.0.0
cardano-crypto version	1.3.0	1.3.0
cardano-prelude version	0.2.1.0	0.2.1.0
cardano-node git	b0a1259	0d697f1
ouroboros-consensus git	8e3afe1	96a9e1b
ouroboros-network git	6275b8f	b49dfd9
cardano-ledger-core git	a9e78ae	faa7a9d
plutus-core git	ba16ec6	8ab7a76
cardano-crypto git	unknown	unknown
cardano-prelude git	68e015f	68e015f
Era	conway	conway
Delegation map size	1000000	1000000
Stuffed UTxO size	4000000	4000000
DRep count	10000	10000
Extra tx payload	100	100
Tx inputs	2	2
Tx Outputs	2	2
TPS	12.0	12.0
Transaction count	768000	768000
Plutus script	—	—
Machines	52	52
Number of filters applied	3	3
Log objects emitted per host	4050952.5192	4029468.6730
Log objects analysed per host	1920901.8846	1935456.6153
Host run time, s	63895.6	63971.9
Host log line rate, Hz	63.400	62.988
Total log objects analysed	99886898	100643744
Run time, s	63901	63975
Analysed run duration, s	48025	48106
Run time efficiency	0.75	0.75
Node start spread, s	4.8893897	7.0311806
Node stop spread, s	4.8149166	4.6823887
Slots analysed	48022	48105
Blocks analysed	2327	2214
Blocks rejected	875	902

Analysis

Resource Usage

	10.5.4	10.6.2	Δ	$\Delta\%$
Forge loop starts, units	0.99864	0.99882	0.000	0.0
Process CPU usage, %	5.7687	5.4359	-0.333	-6
RTS GC CPU usage, %	0.3655	0.2742	-0.091	-25
RTS Mutator CPU usage, %	5.4016	5.1573	-0.244	-5
Major GCs, events	0.0008	0.001	0.000	27
Minor GCs, events	1.4147	0.6981	-0.717	-51
Kernel RSS, MiB	8735.4	6602.5	-2132.937	-24
RTS heap size, MiB	8674.2	6539.5	-2134.695	-25
RTS live GC dataset, MiB	3886.2	2992.	-894.125	-23
RTS alloc rate, MiB/s	42.289	21.37	-20.919	-49
Filesystem reads, KiB/s	0.0013	0.0	-0.001	-100
Filesystem writes, KiB/s	256.93	252.27	-4.665	-2
CPU 85% spans, slots	6.4549	10.513	4.058	63

Anomaly control

	10.5.4	10.6.2	Δ	$\Delta\%$
Blocks per host, blocks	63.385	62.154	-1.231	-2
Filtered to chained blocks, :	0.7263	0.7121	-0.014	-2
Chained to forged blocks, :	0.9718	0.9647	-0.007	-0.7
Height & slot battles, blocks	0.00731	0.00632	-0.001	-13
Block size, Bytes	88968	88963	-4.886	-0.0

Forging

	10.5.4	10.6.2	Δ	$\Delta\%$
Started forge loop iteration, s	0.00105	0.00142	0.000	35
Acquired block context, s	7.7e-5	6.6e-5	-0.000	0
Acquired ledger state, s	0.0001	0.00012	0.000	0
Acquired ledger view, s	2.9e-5	2.9e-5	0.000	0
Leadership check duration, s	0.00044	0.00041	-0.000	0
Ledger ticking, s	0.02679	0.02142	-0.005	-20
Mempool snapshotting, s	0.04521	0.04687	0.002	4
Leadership to forged, s	0.00072	0.00077	0.000	0
Forged to announced, s	0.00078	0.00072	-0.000	0
Forged to sending, s	0.00625	0.00814	0.002	30
Forged to self-adopted, s	0.07128	0.06822	-0.003	-4
Slot start to announced, s	0.0752	0.07183	-0.003	-4

Individual peer propagation

	10.5.4	10.6.2	Δ	$\Delta\%$
First peer notice, s	0.0772	0.074	-0.003	-4
First peer fetch, s	0.0869	0.0878	0.001	1
Notice to fetch request, s	0.00135	0.00143	0.000	0
Fetch duration, s	0.36136	0.35589	-0.005	-2
Fetches to announced, s	0.00102	0.00103	0.000	0
Fetches to sending, s	0.04524	0.04504	-0.000	-0.4
Fetches to adopted, s	0.07444	0.07194	-0.003	-3

End-to-end propagation

	10.5.4	10.6.2	Δ	$\Delta\%$
0.50 adoption, s	0.62535	0.61664	-0.009	-1
0.80 adoption, s	0.98923	0.97922	-0.010	-1
0.90 adoption, s	1.00911	1.00147	-0.008	-0.8
0.92 adoption, s	1.01384	1.00622	-0.008	-0.8
0.94 adoption, s	1.01979	1.01087	-0.009	-0.9
0.96 adoption, s	1.02648	1.01726	-0.009	-0.9
0.98 adoption, s	1.03601	1.02698	-0.009	-0.9
1.00 adoption, s	1.05962	1.05967	0.000	0

Appendix A: charts

Cluster performance charts

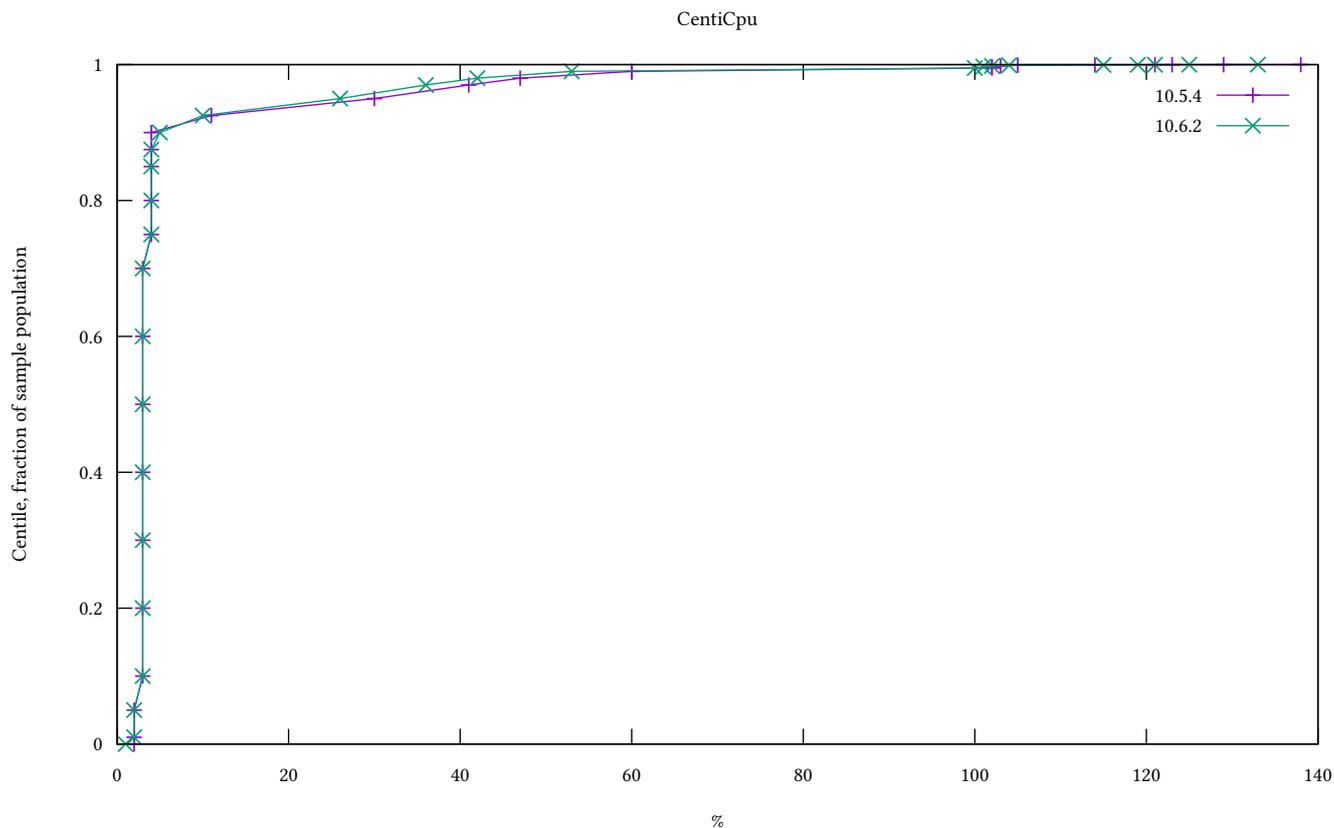


Figure 1: Process CPU usage

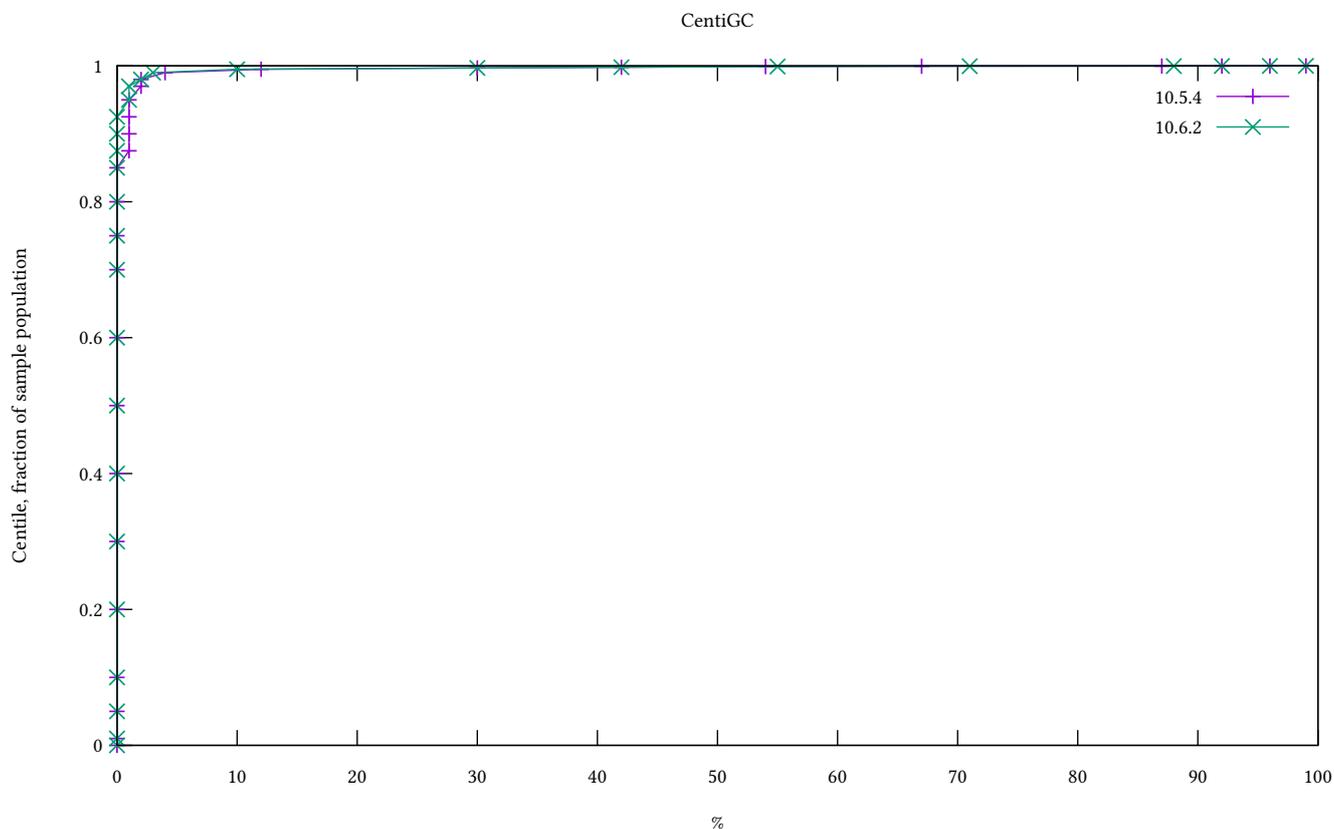


Figure 2: RTS GC CPU usage

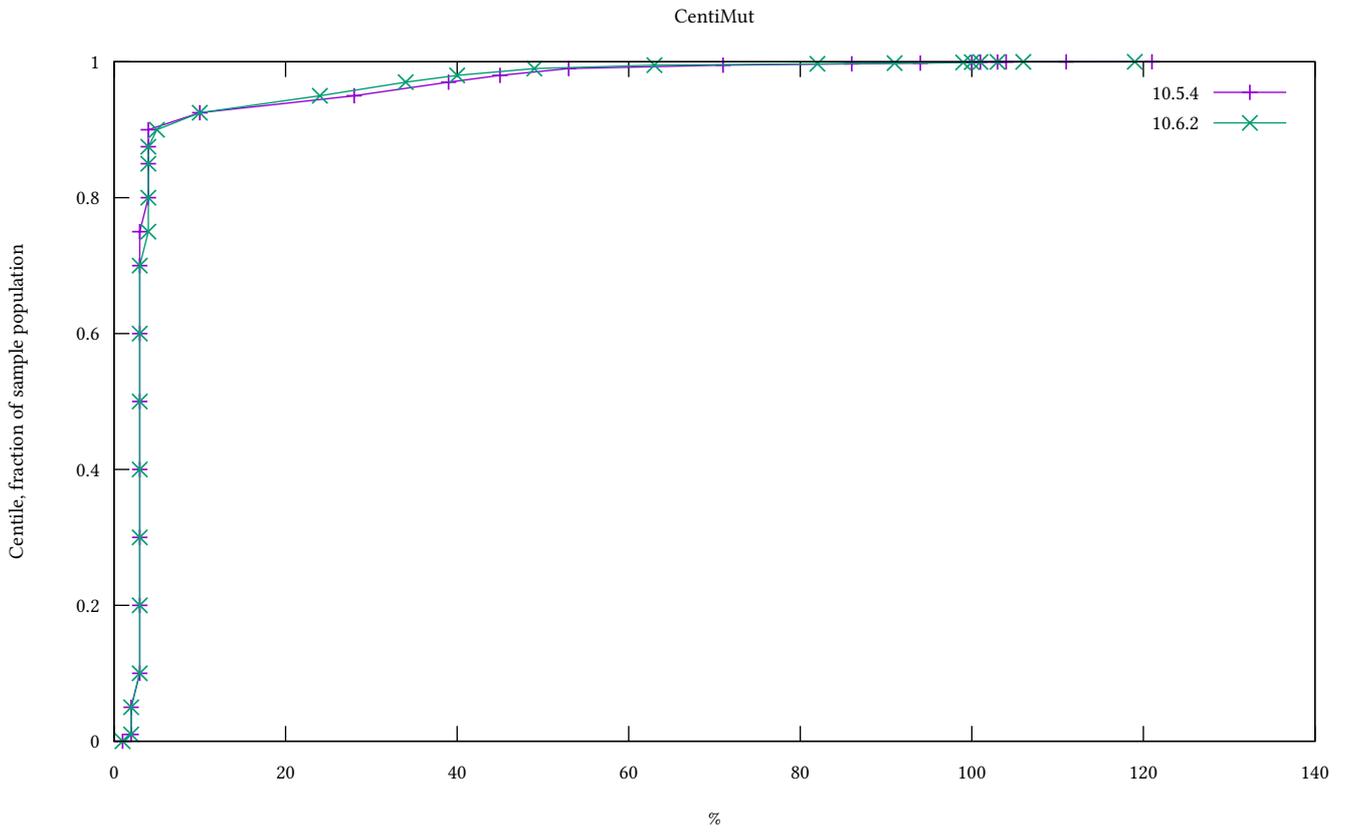


Figure 3: RTS Mutator CPU usage

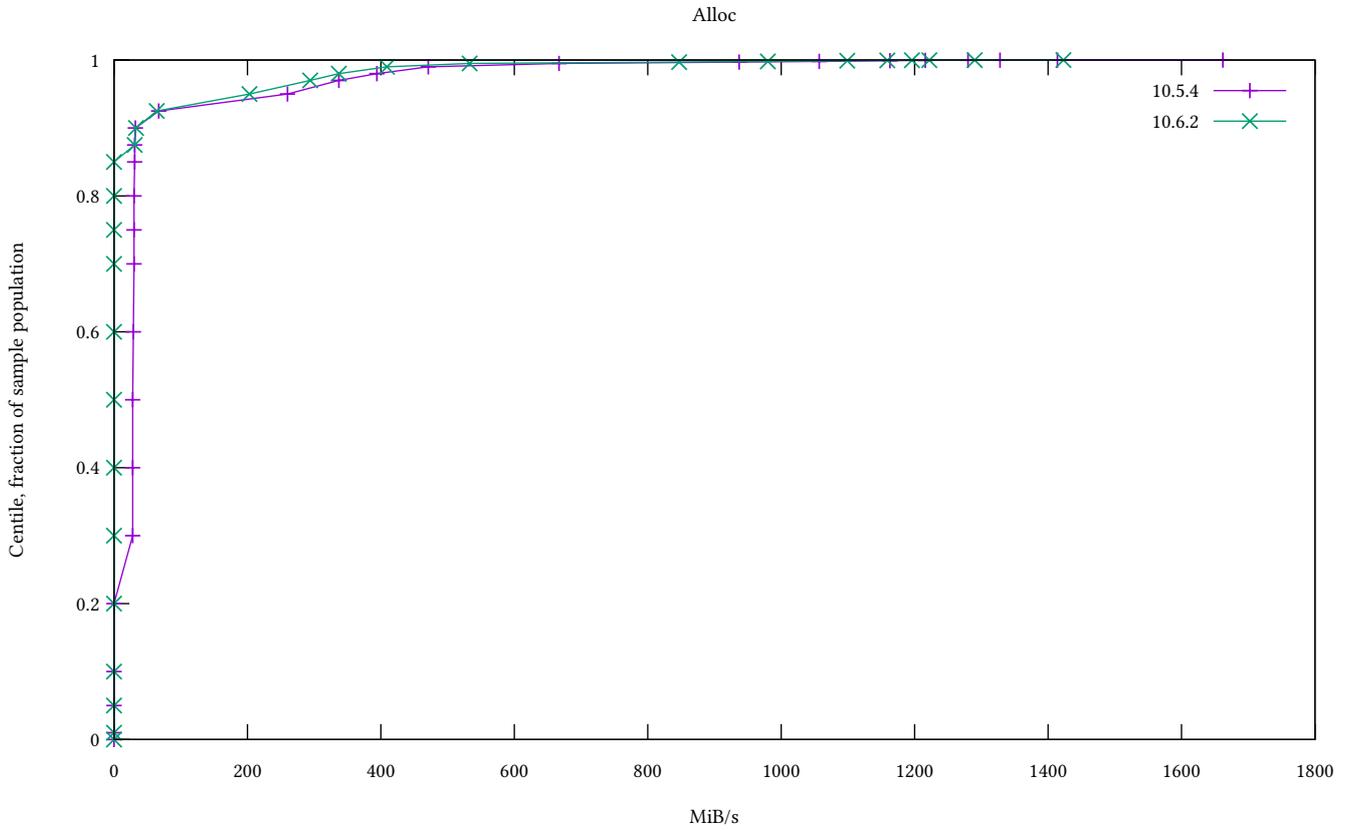


Figure 4: RTS alloc rate

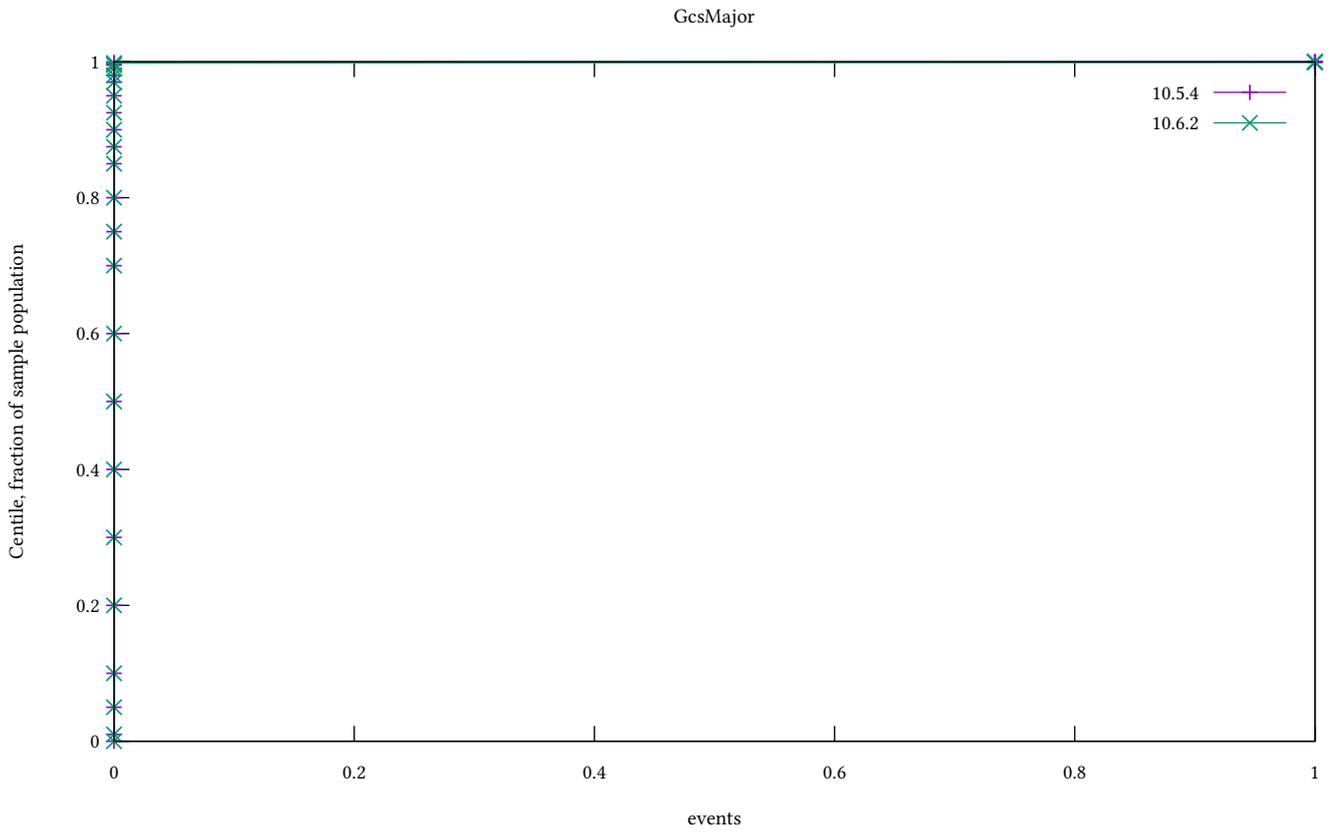


Figure 5: Major GCs

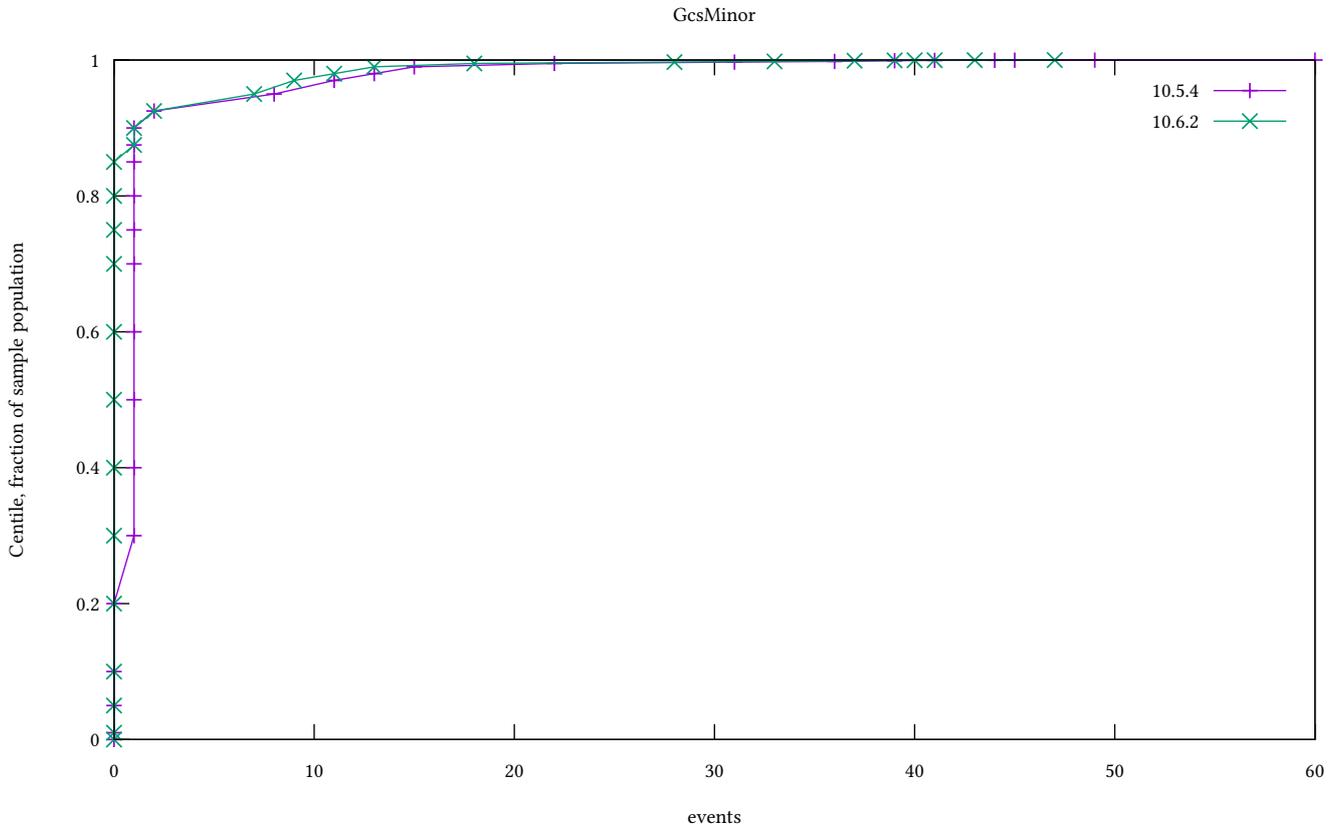


Figure 6: Minor GCs

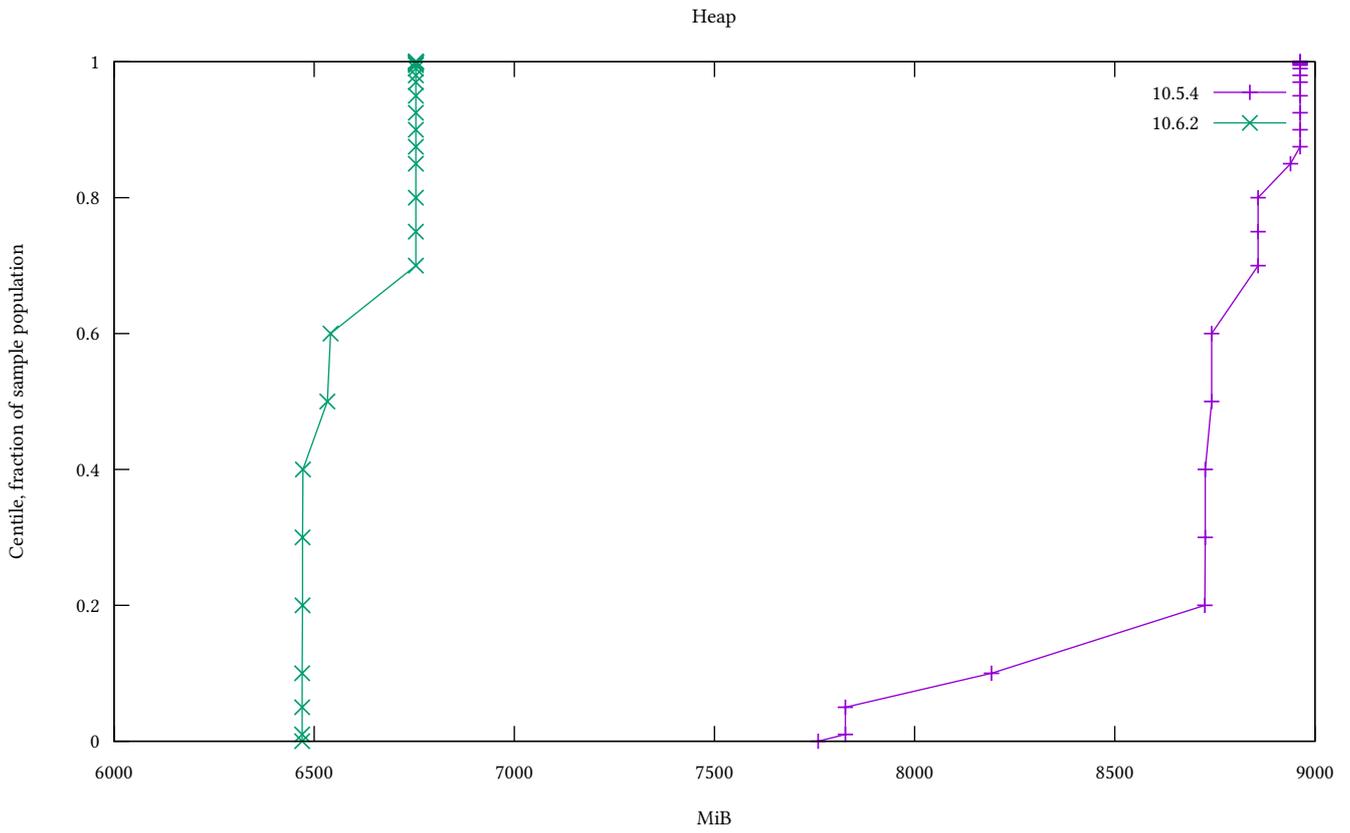


Figure 7: RTS heap size

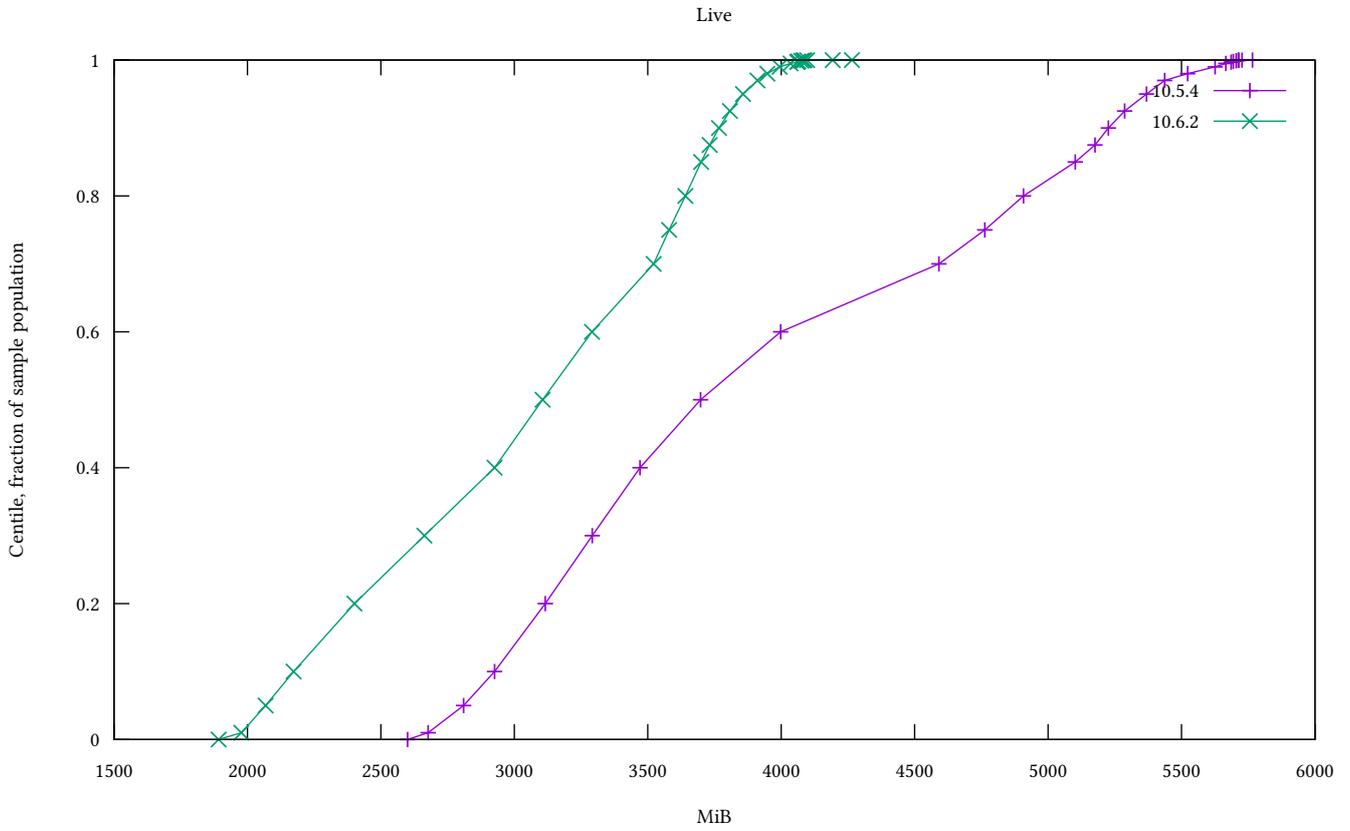


Figure 8: RTS live GC dataset

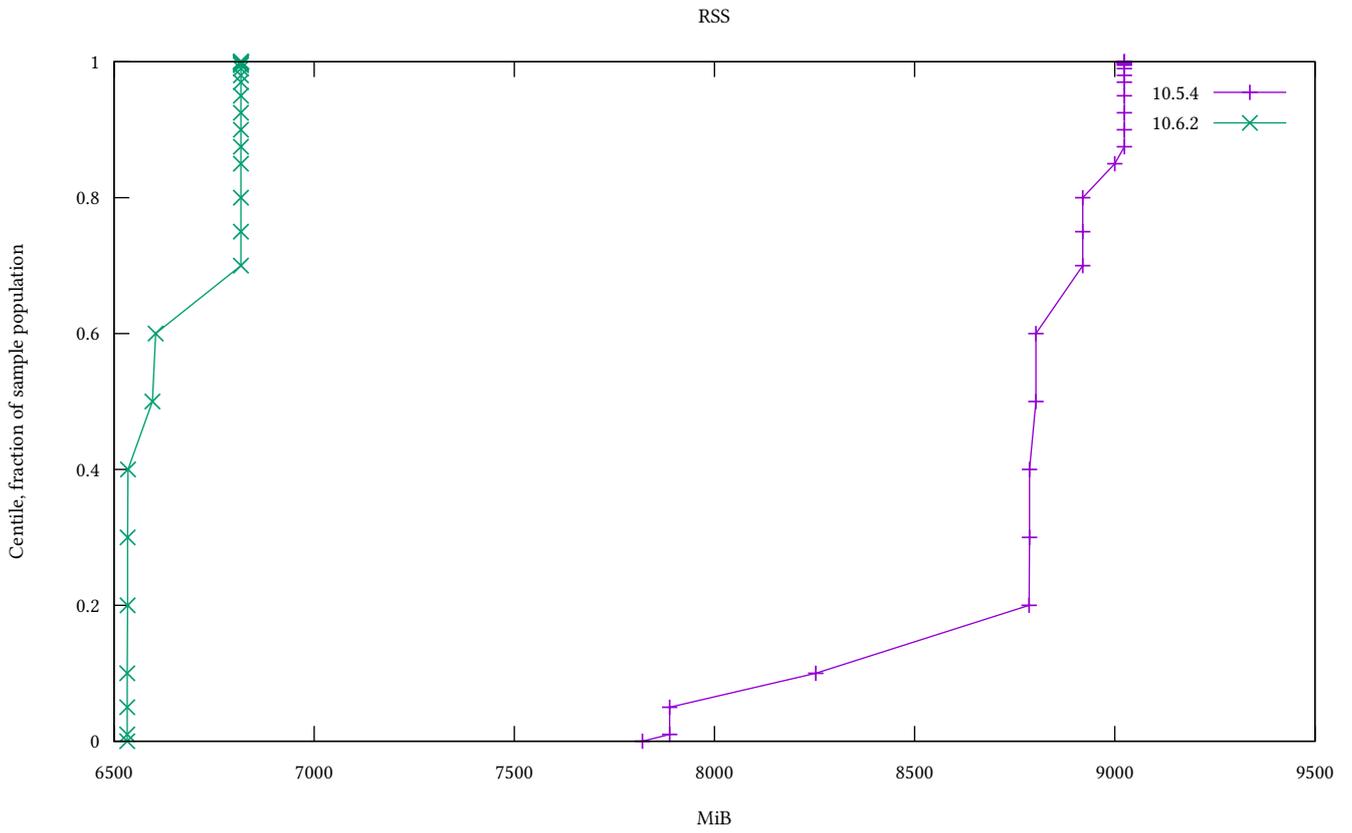


Figure 9: Kernel RSS

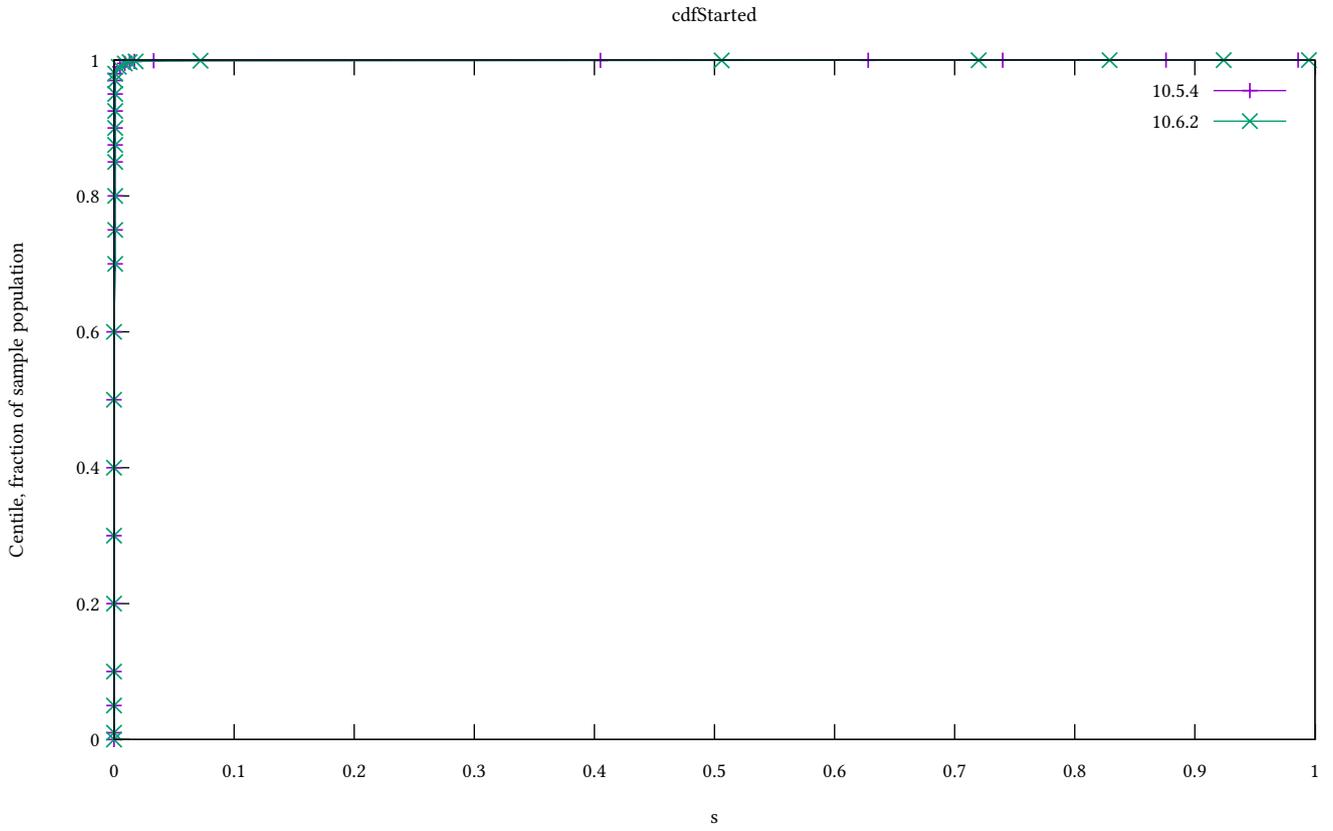


Figure 10: Forge loop tardiness

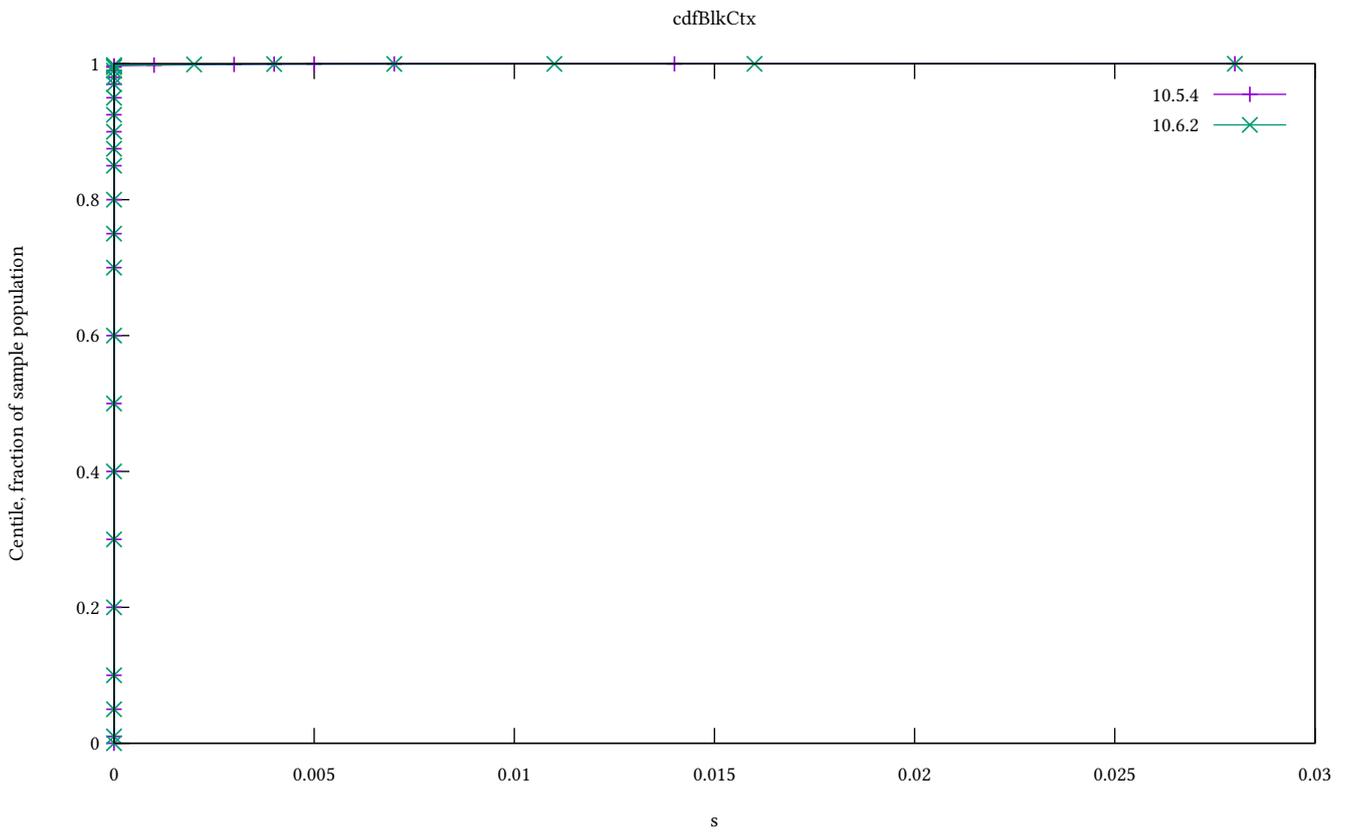


Figure 11: Block context acquisition delay

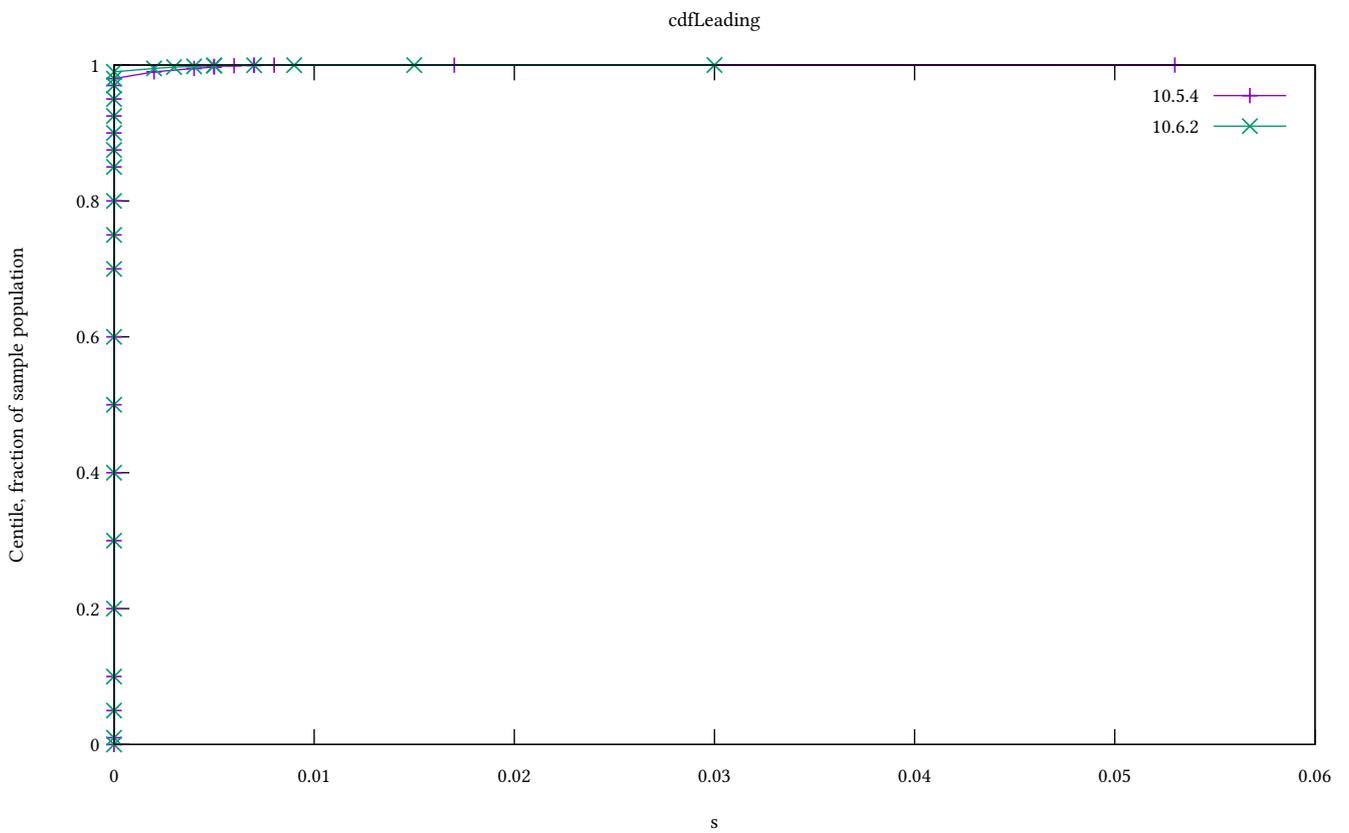


Figure 12: Leadership check duration

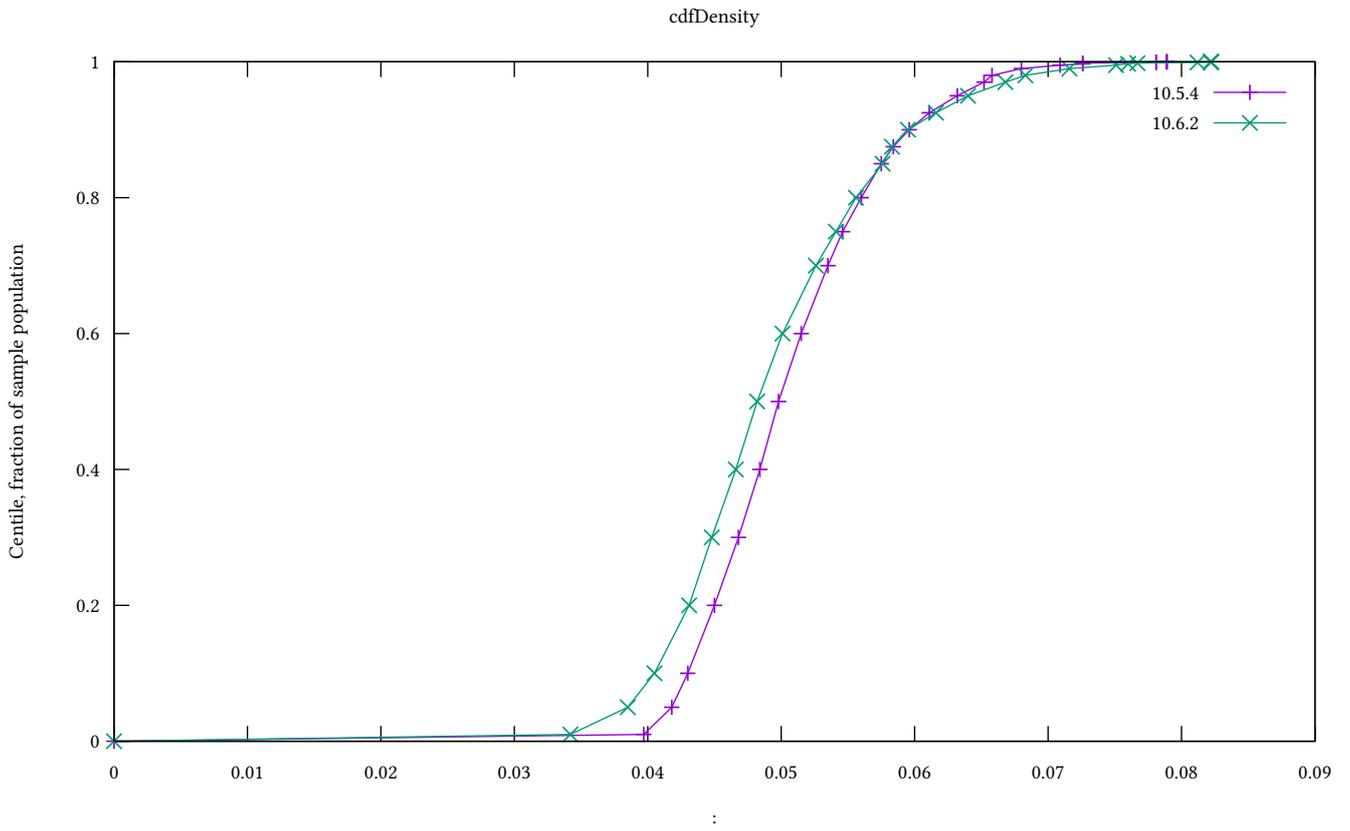


Figure 13: Chain density

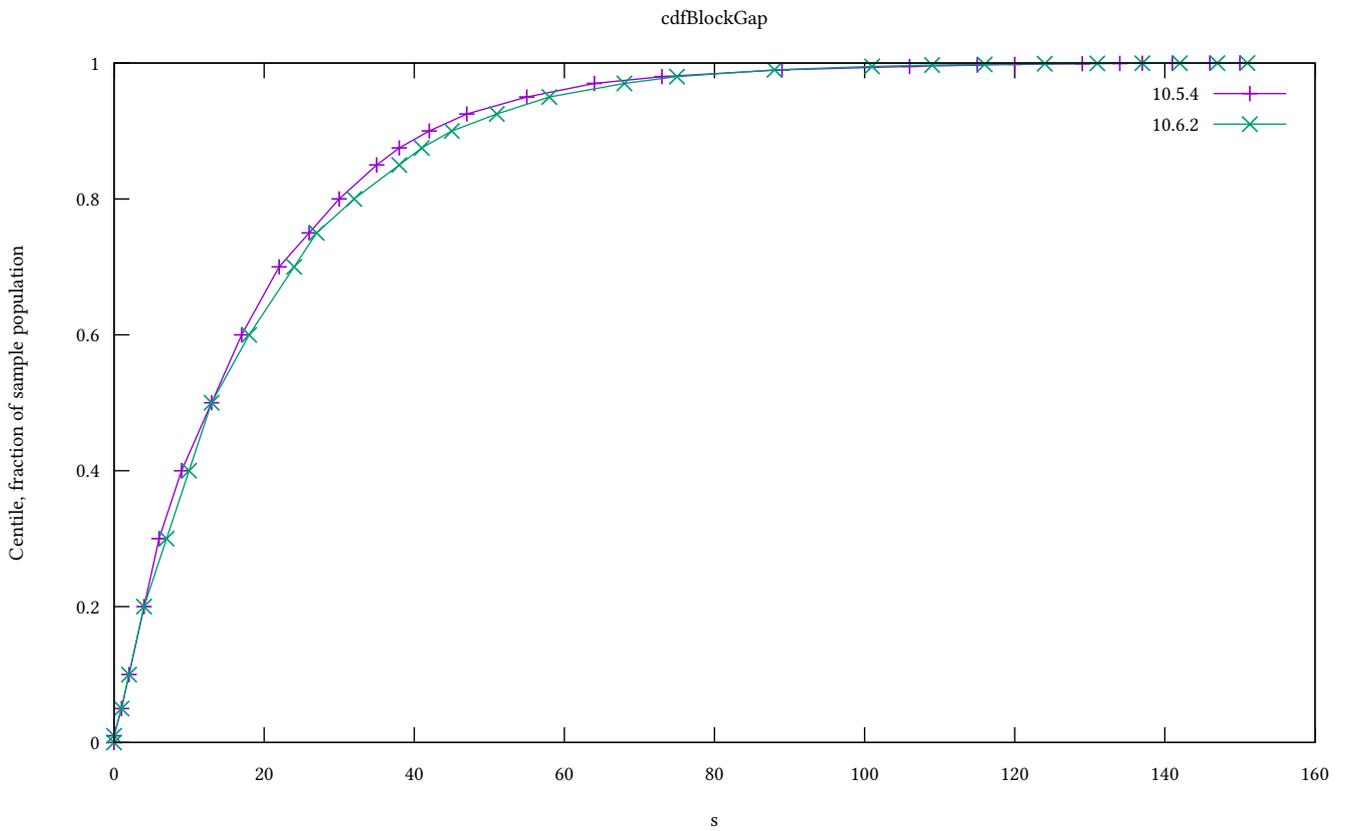


Figure 14: Interblock gap

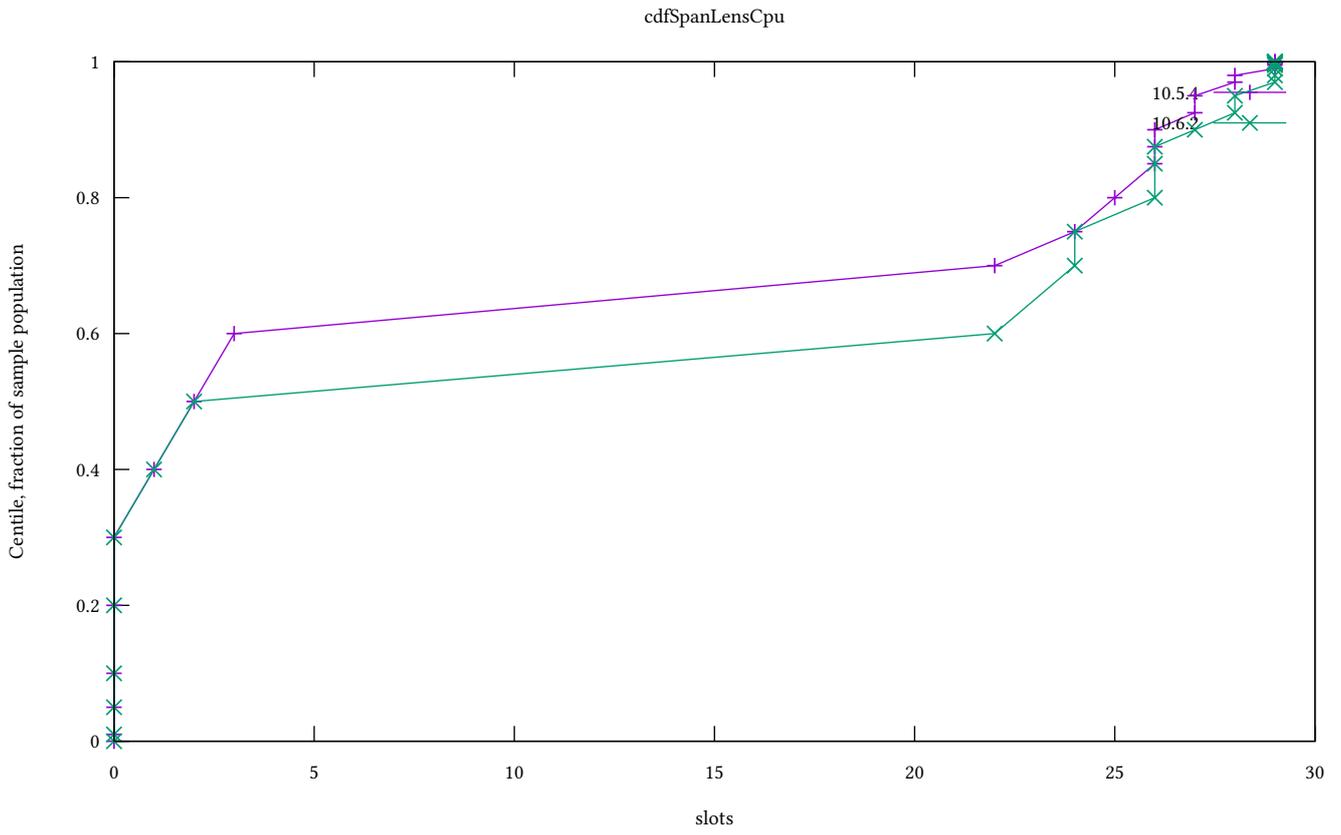


Figure 15: CPU 85% spans

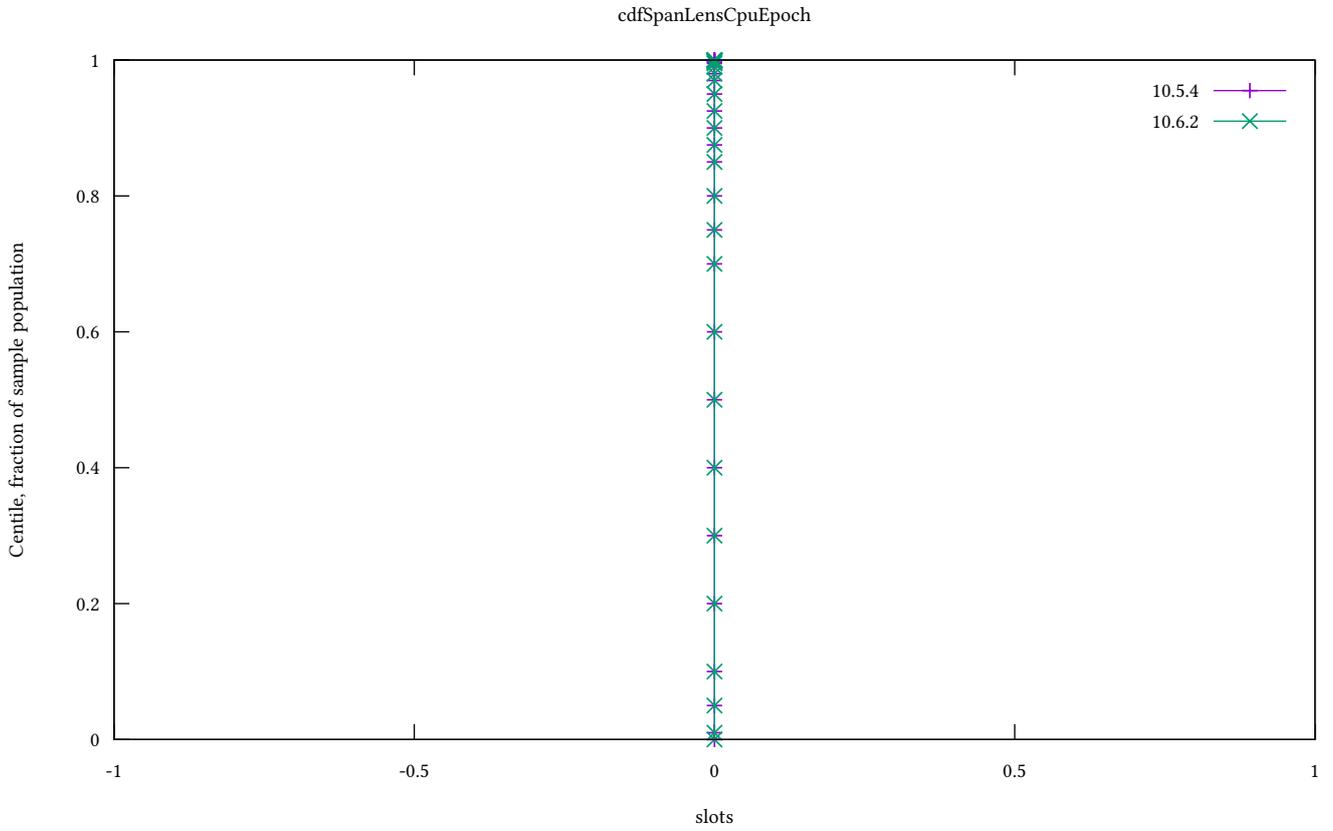


Figure 16: CPU spans at Ep boundary

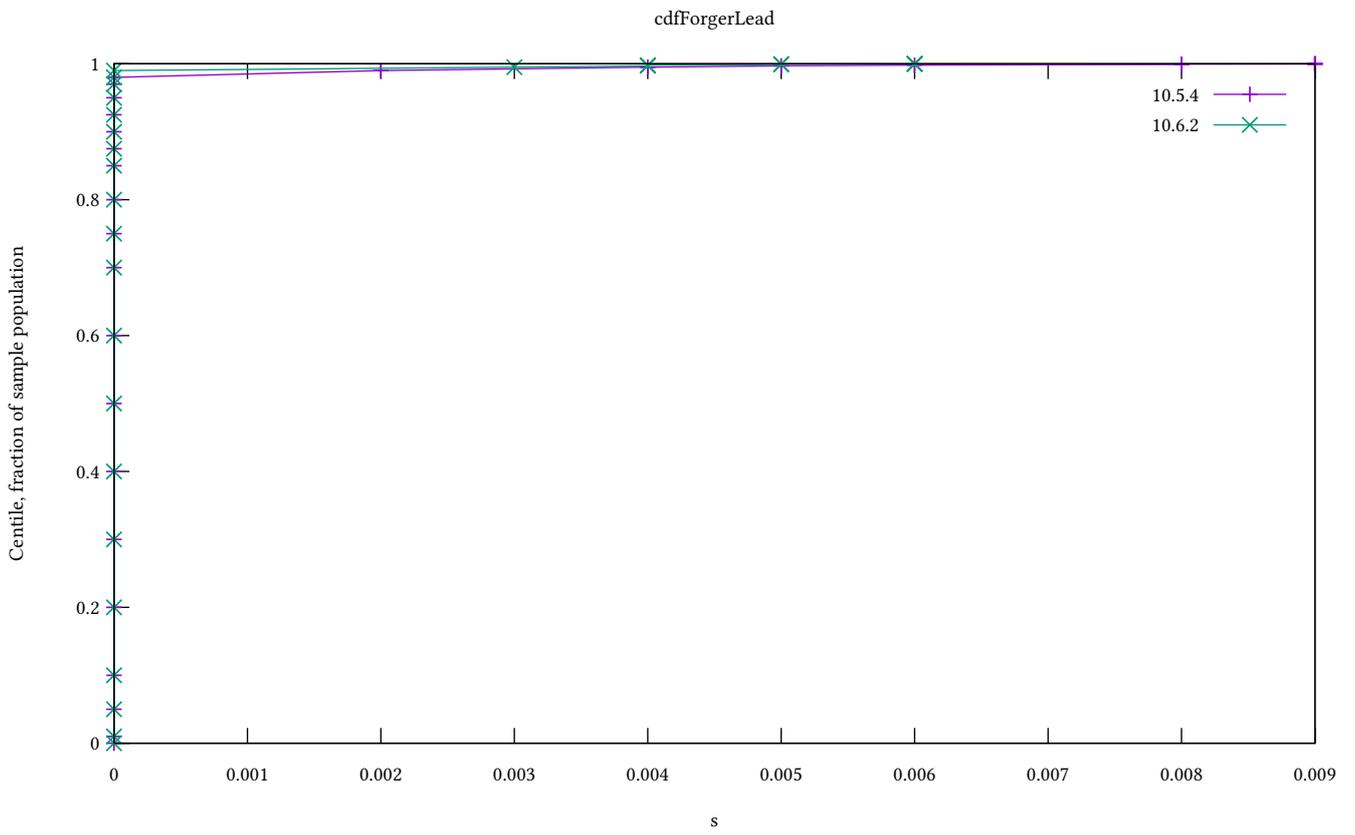


Figure 17: Leadership check duration

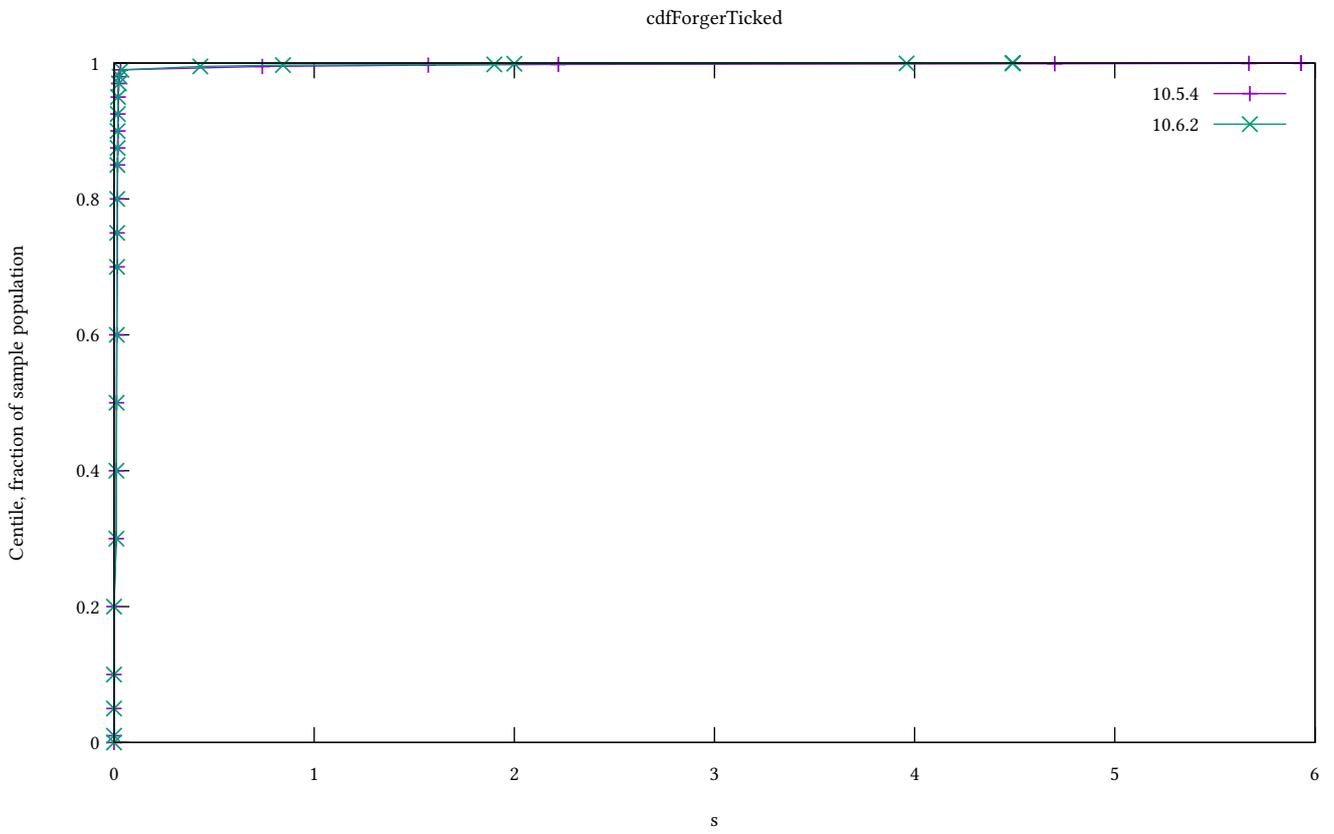


Figure 18: Ledger ticking

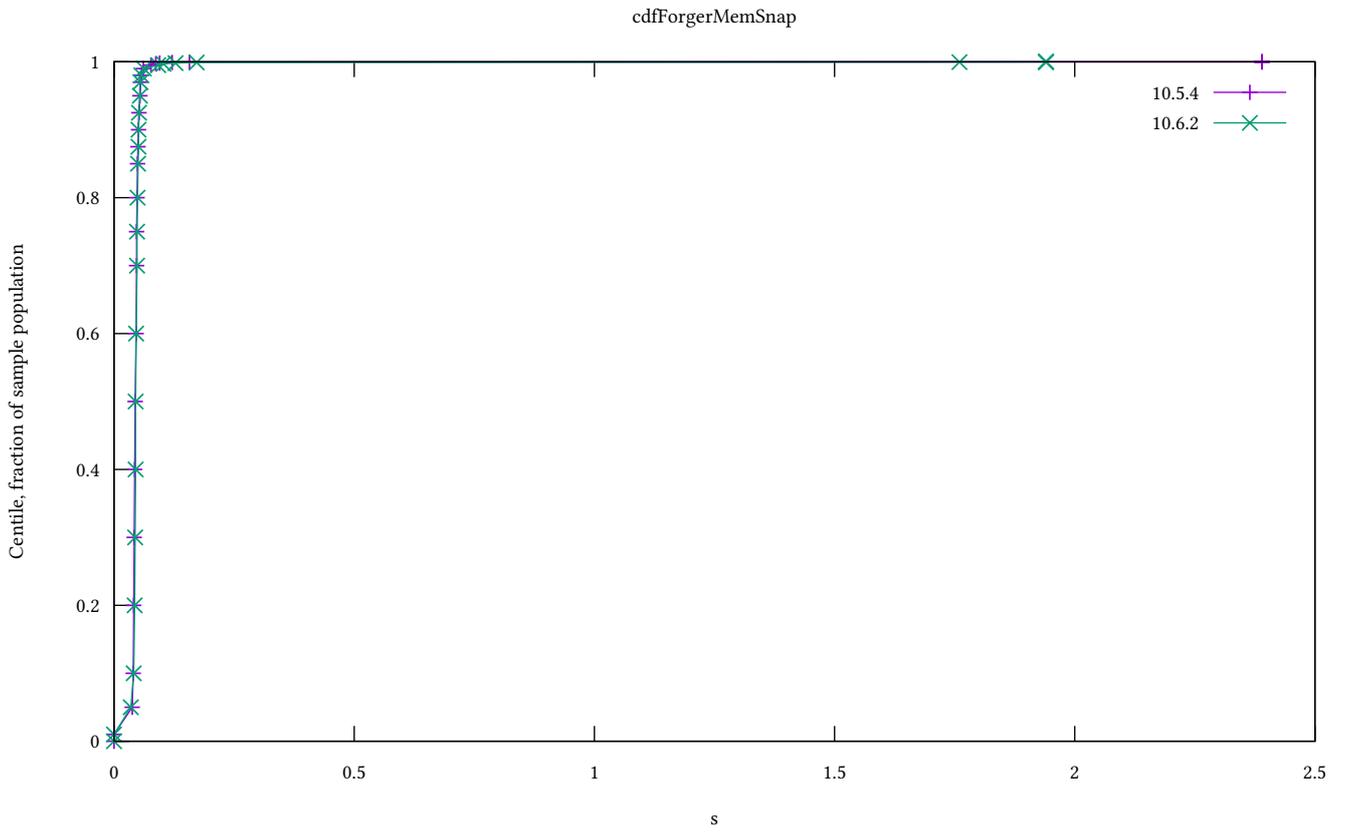


Figure 19: Mempool snapshotting

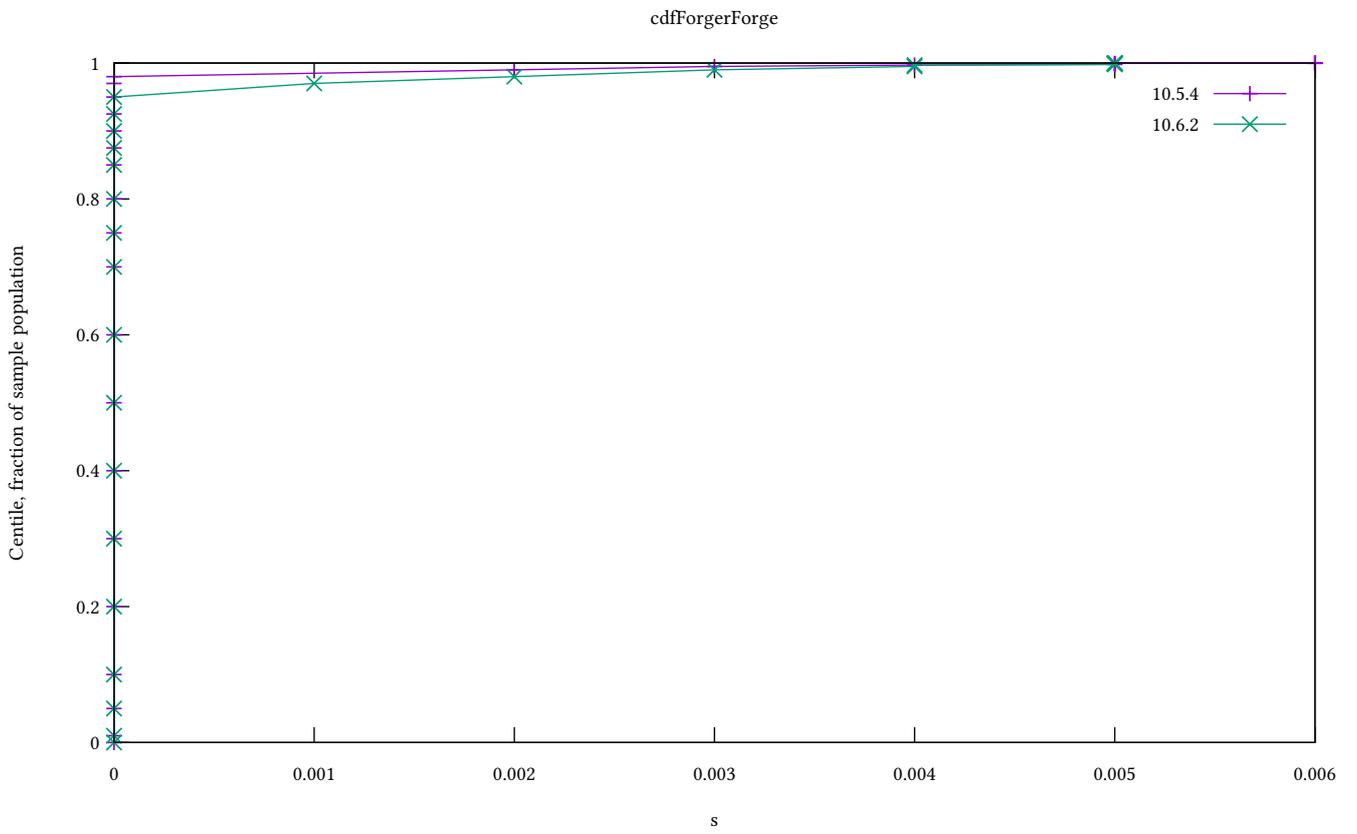


Figure 20: Leadership to forged

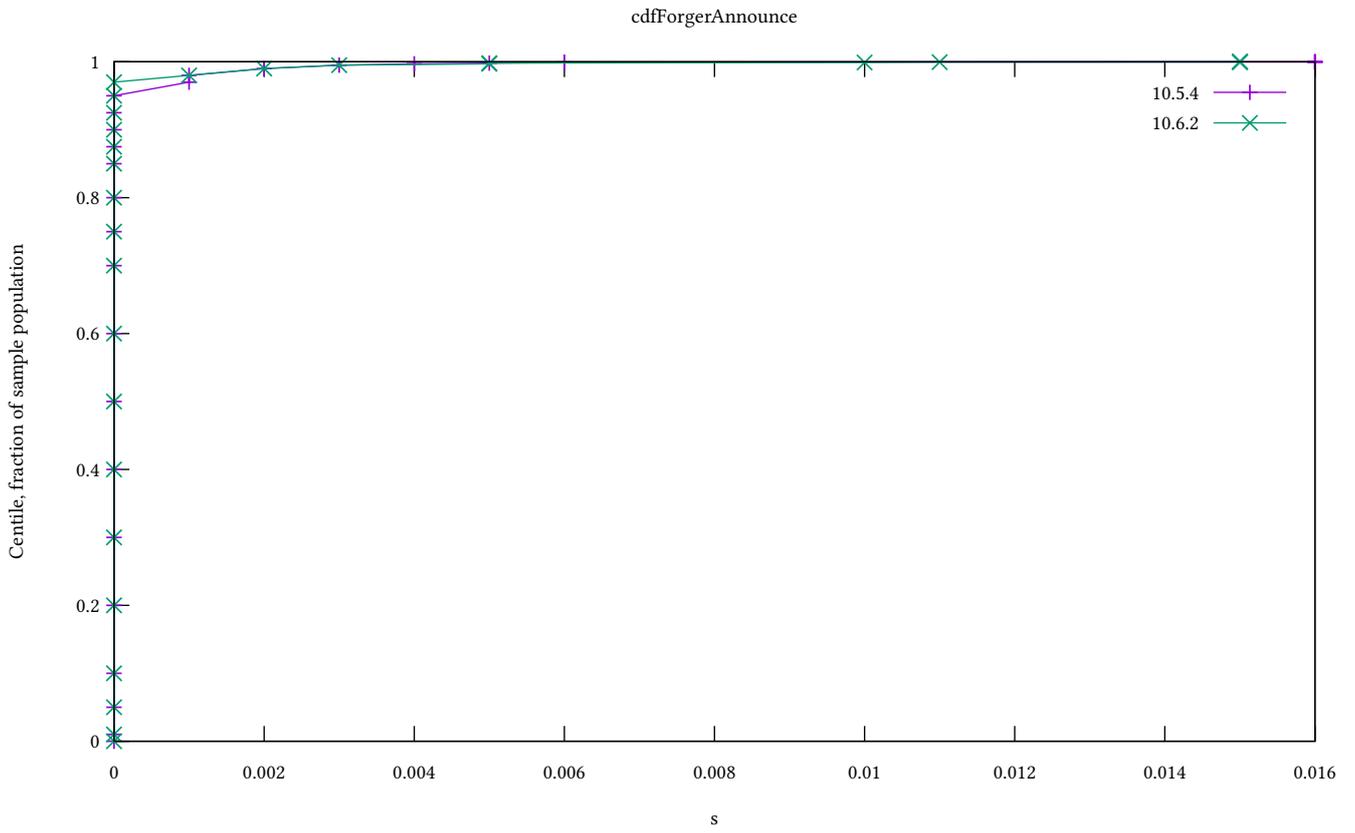


Figure 21: Forged to announced

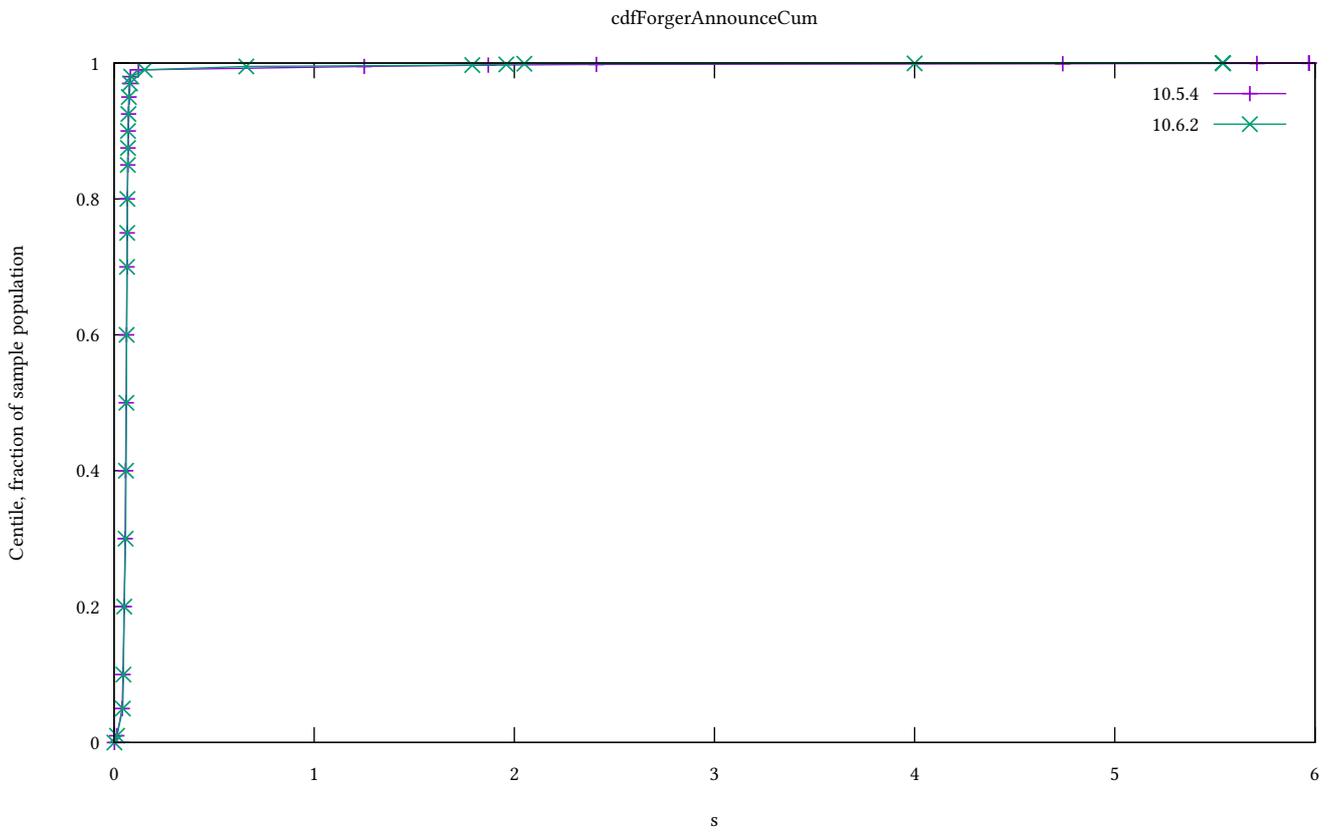


Figure 22: Slot start to announced

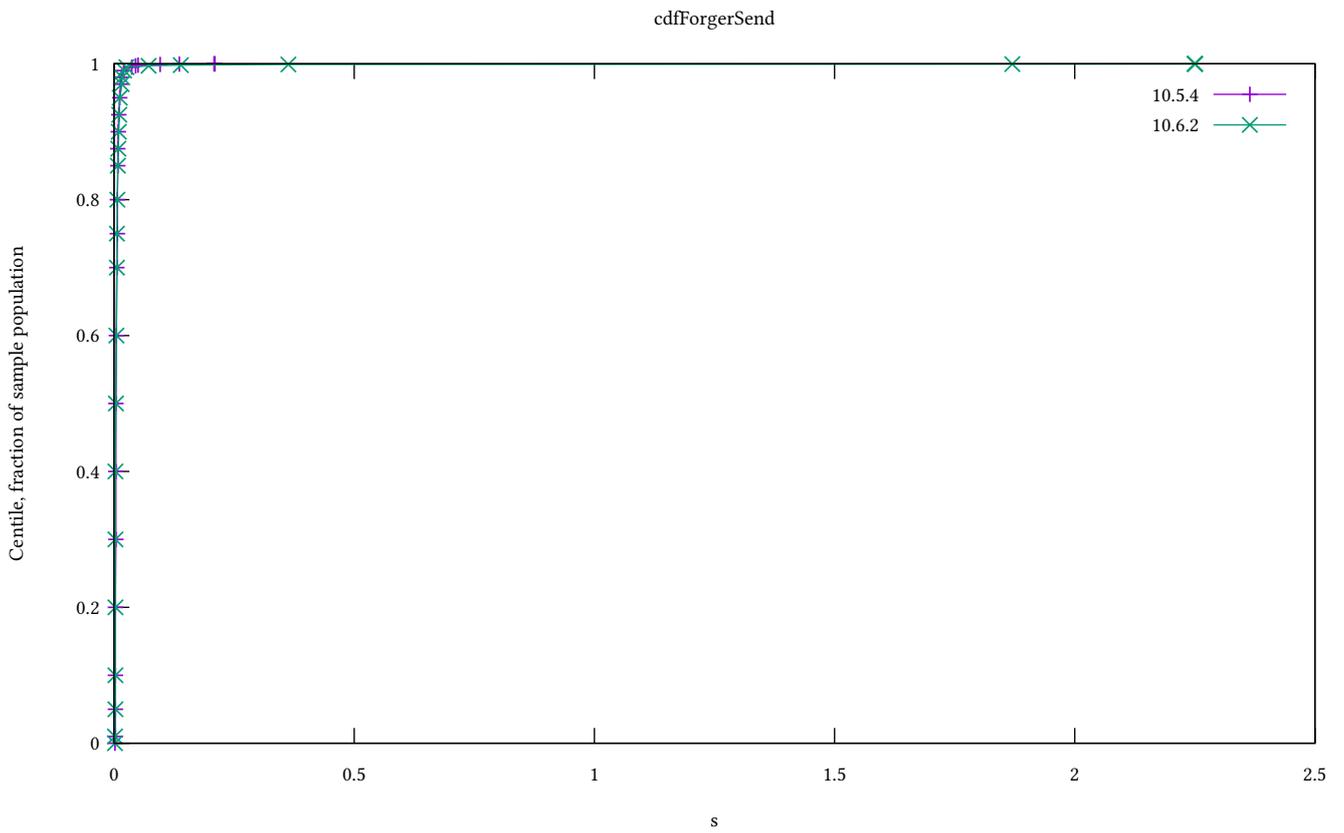


Figure 23: Forged to sending

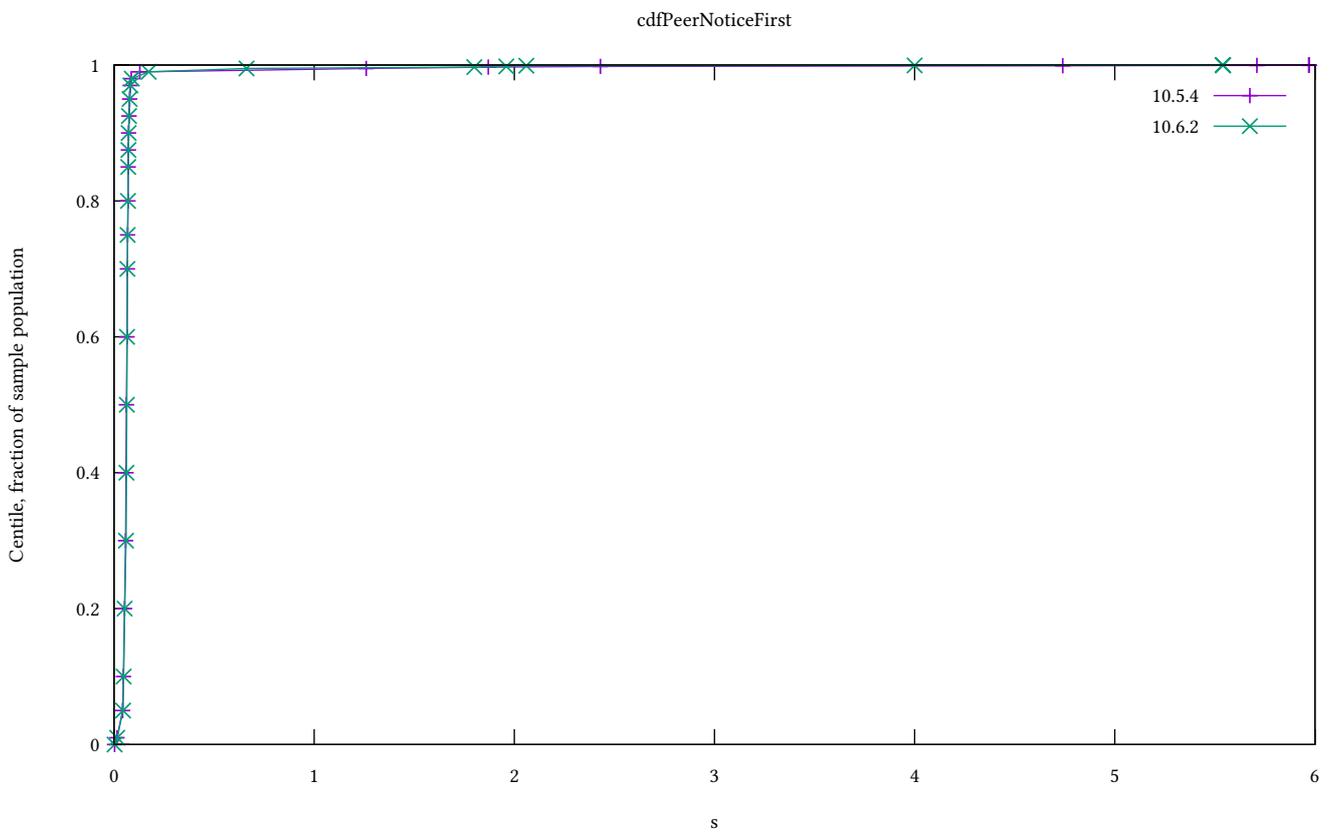


Figure 24: First peer notice

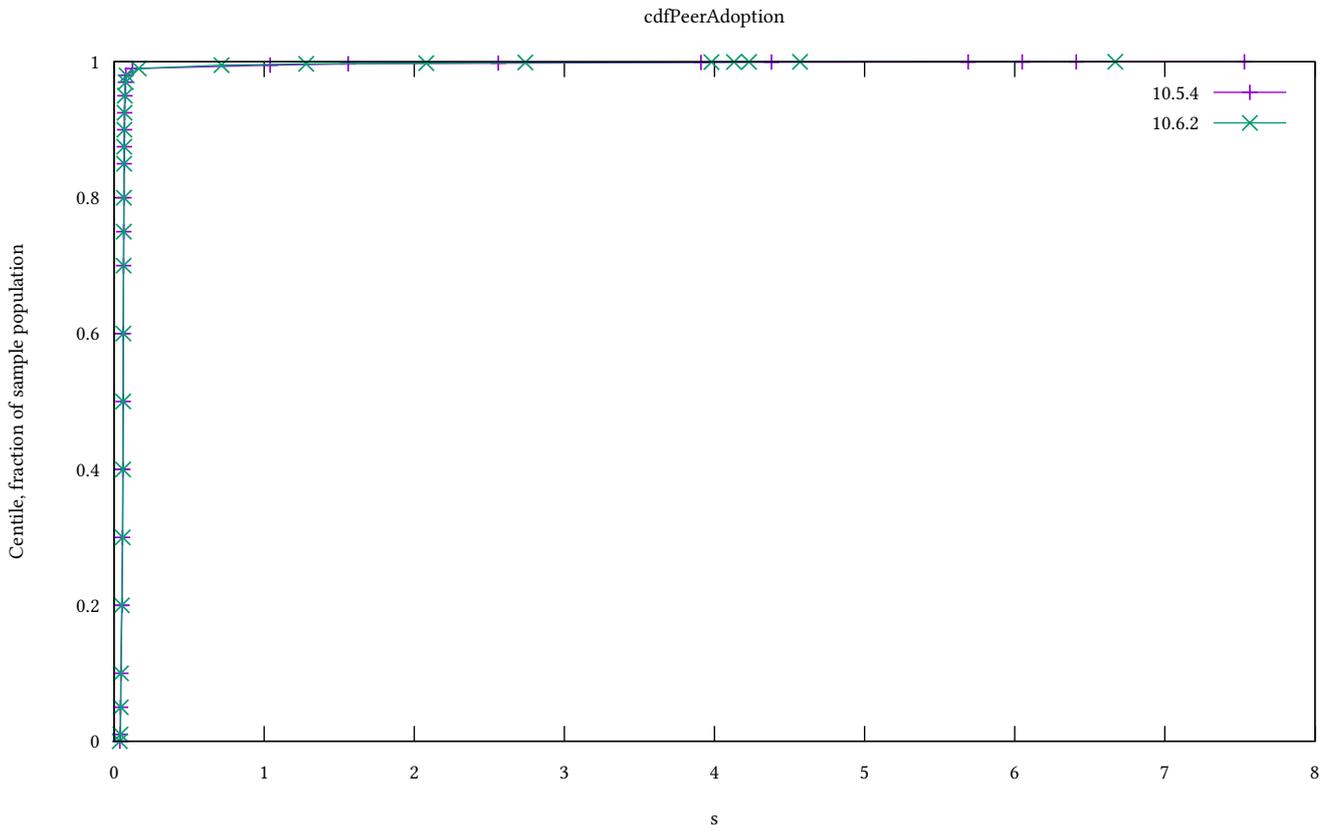


Figure 25: Fetched to adopted

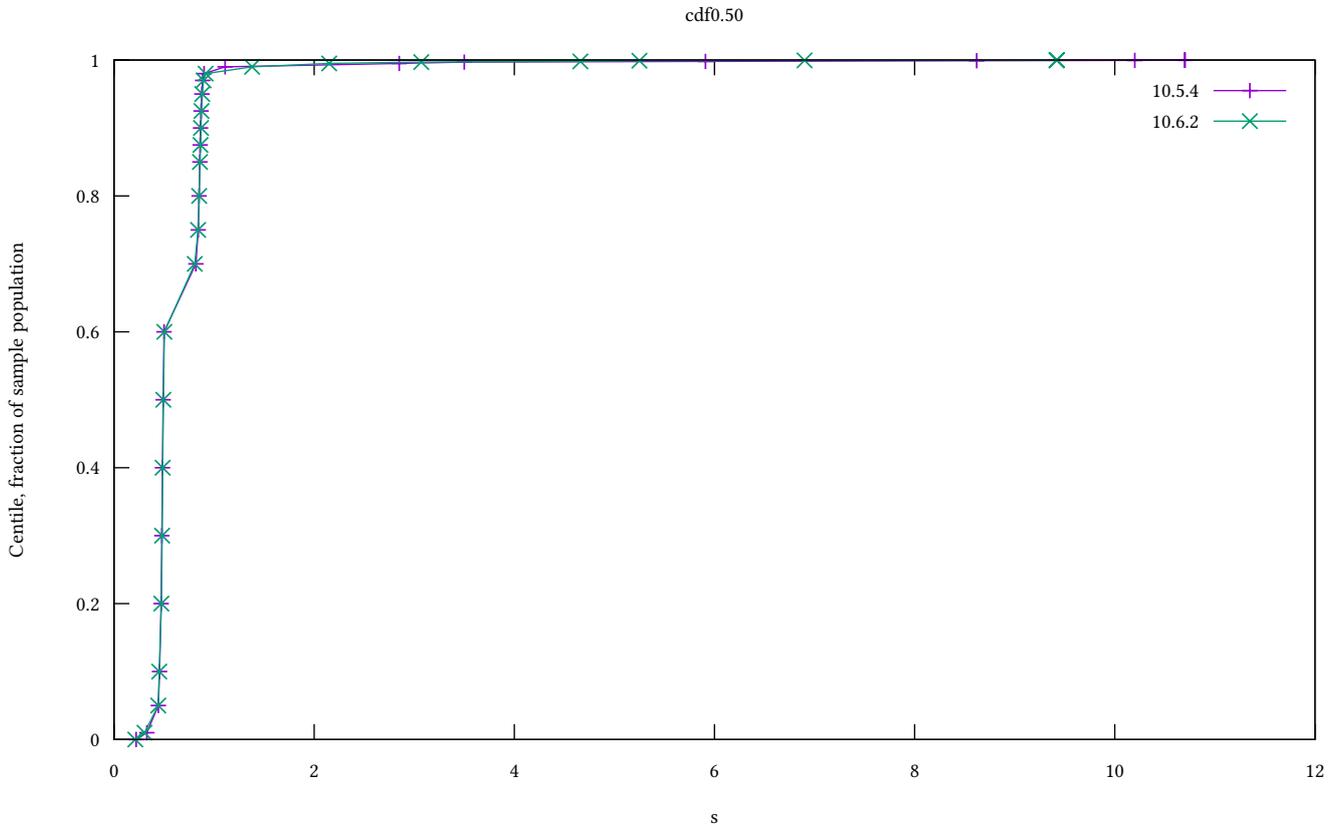


Figure 26: 0.50 adoption

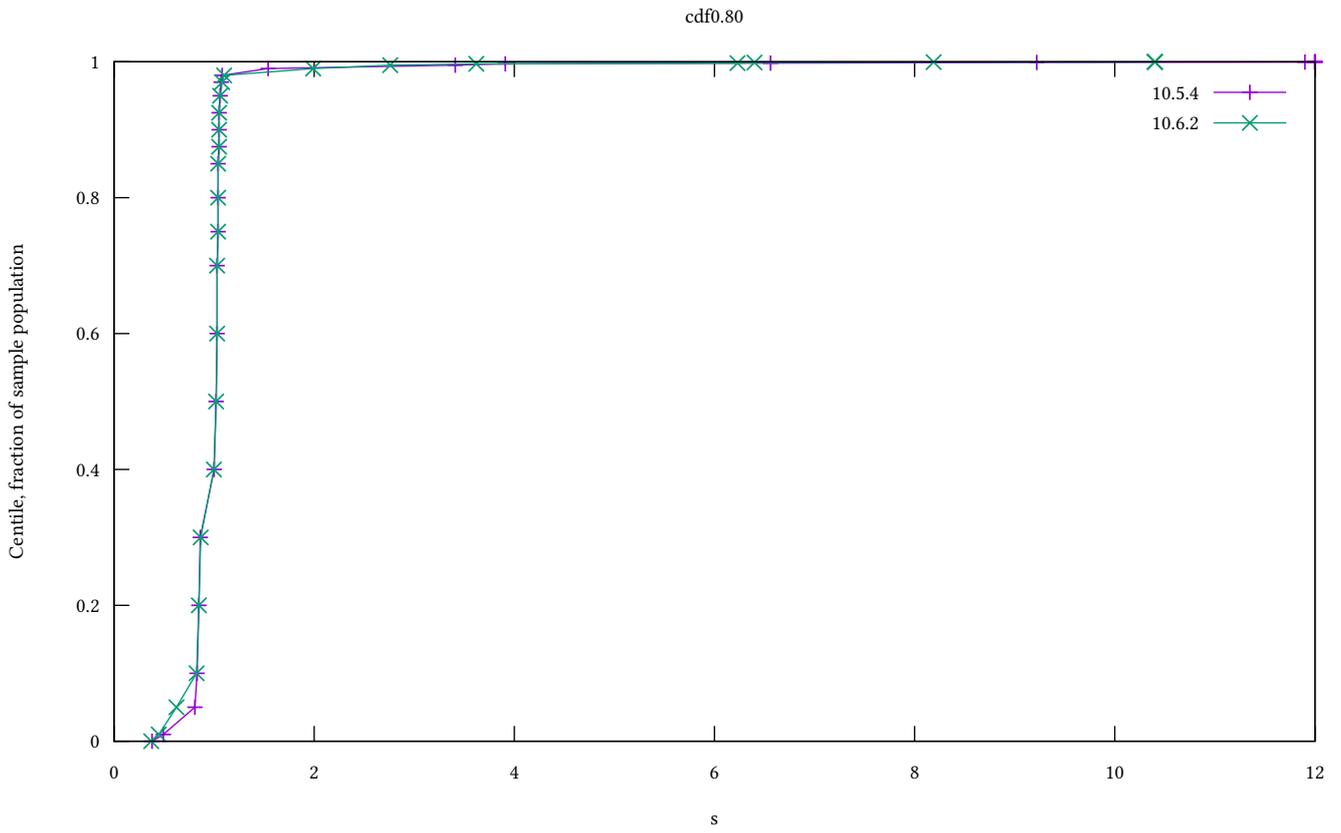


Figure 27: 0.80 adoption

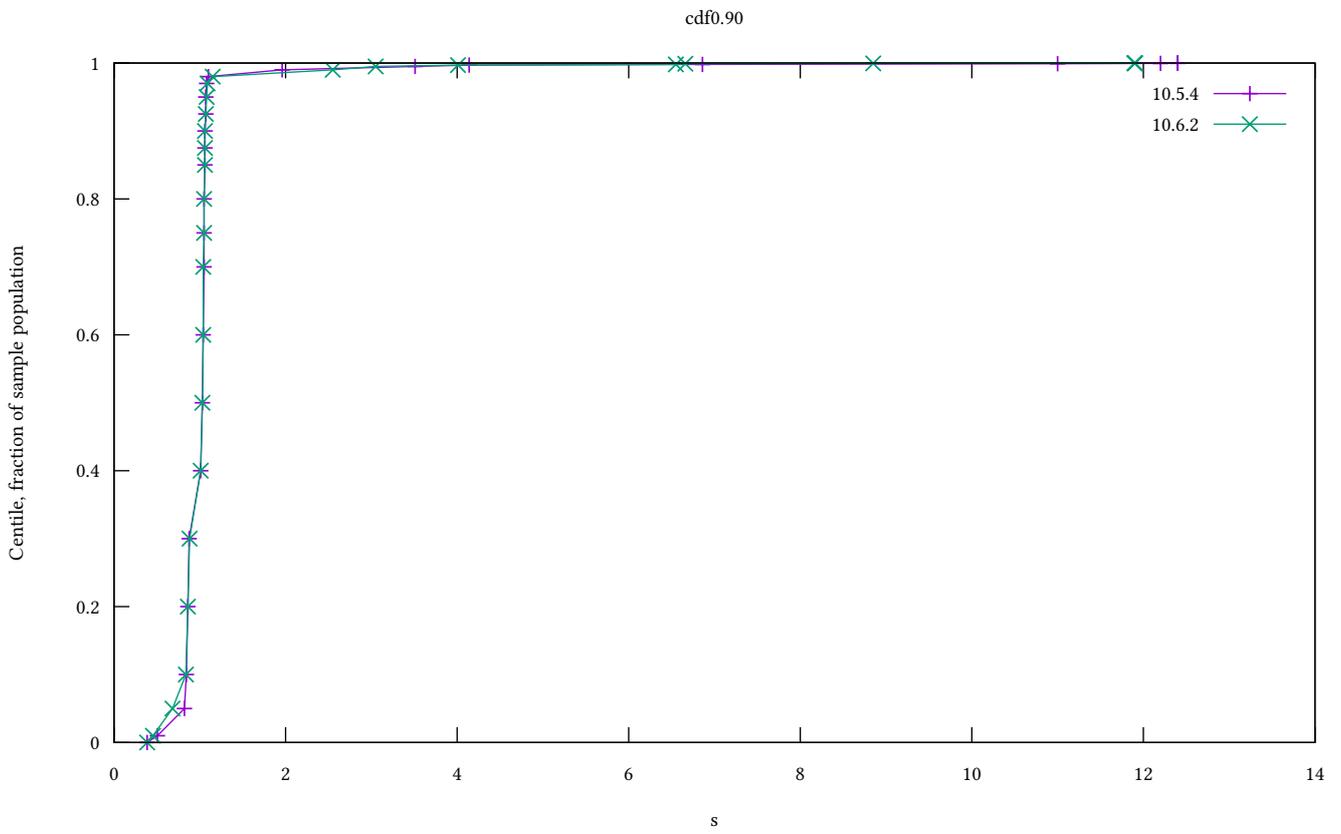


Figure 28: 0.90 adoption

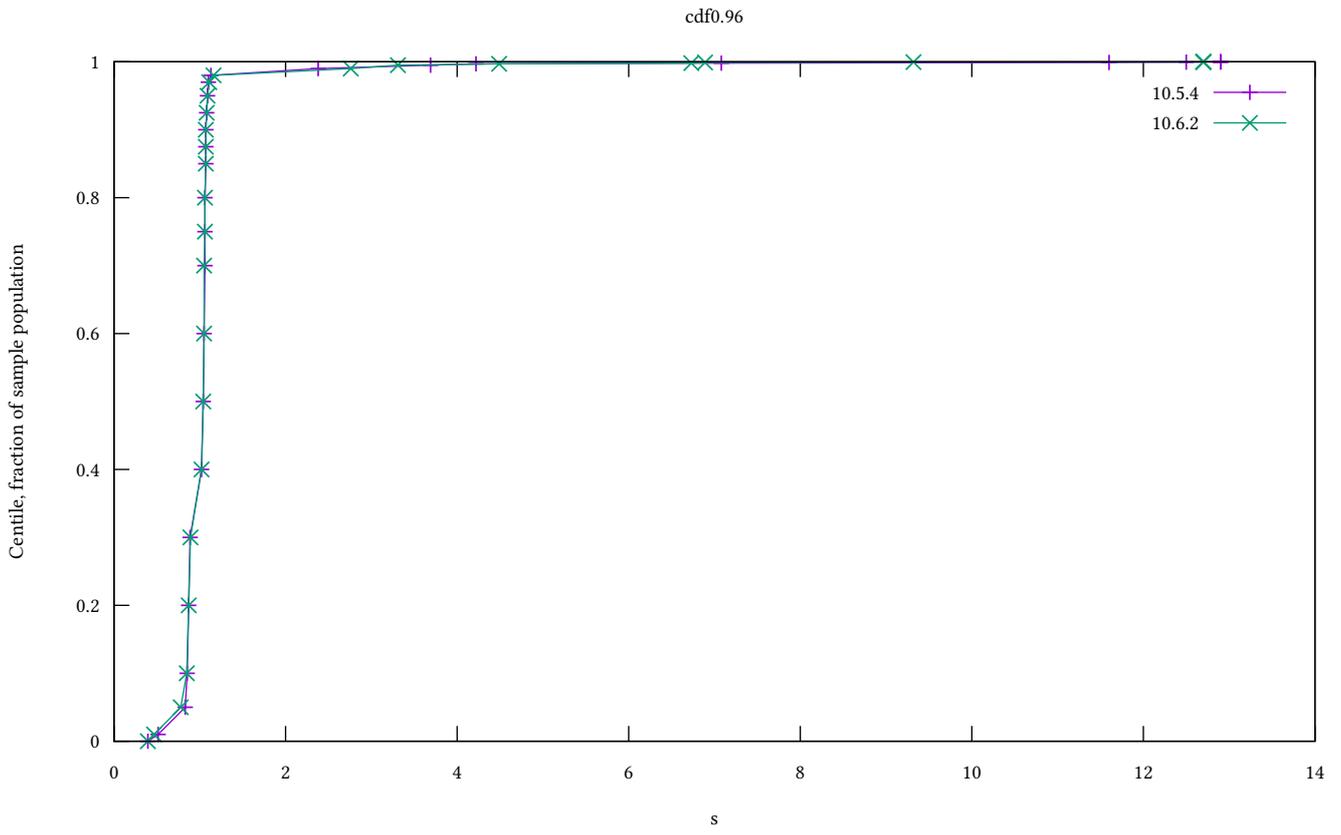


Figure 29: 0.96 adoption

Appendix B: data dictionary

Block propagation metrics

0.50 adoption (*cdf0.50*) – Time since slot start to block’s adoption by 50% of the cluster.

0.80 adoption (*cdf0.80*) – Time since slot start to block’s adoption by 80% of the cluster.

0.90 adoption (*cdf0.90*) – Time since slot start to block’s adoption by 90% of the cluster.

0.92 adoption (*cdf0.92*) – Time since slot start to block’s adoption by 92% of the cluster.

0.94 adoption (*cdf0.94*) – Time since slot start to block’s adoption by 94% of the cluster.

0.96 adoption (*cdf0.96*) – Time since slot start to block’s adoption by 96% of the cluster.

0.98 adoption (*cdf0.98*) – Time since slot start to block’s adoption by 98% of the cluster.

1.00 adoption (*cdf1.00*) – Time since slot start to block’s adoption by 100% of the cluster.

Height & slot battles (*cdfBlockBattle*) – For a given block, number of all abandoned blocks at its block height. Sum of height and slot battles

Block size (*cdfBlockSize*) – Block size, in bytes

Chained to forged blocks (*cdfBlocksChainedRatio*) – For each host, ratio of blocks that made into chain / all forged

Filtered to chained blocks (*cdfBlocksFilteredRatio*) – For each host, ratio of blocks that passed filtering / all on chain

Blocks per host (*cdfBlocksPerHost*) – For each host, number of blocks made during the entire observation period

Forged to self-adopted (*cdfForgerAdoption*) – Time between block forging completion and adoption (*TraceAdoptedBlock*)

Forged to announced (*cdfForgerAnnounce*) – Time between block forging completion and header announcement (*ChainSyncServerEvent.TraceChainSyncServerRead.AddBlock*)

Slot start to announced (*cdfForgerAnnounceCum*) – Time since slot start until header announcement (*ChainSyncServerEvent.TraceChainSyncServerRead.AddBlock*)

Acquired block context (*cdfForgerBlkCtx*) – Block context acquired (*TraceBlockContext*), relative to forge loop beginning

Leadership to forged (*cdfForgerForge*) – Time spent forging the block: *TraceForgedBlock* relative to positive leadership decision

Leadership check duration (*cdfForgerLead*) – Leadership check duration (*TraceNodeIsNotLeader*, *TraceNodeIsLeader*), relative to ledger view acquisition

Acquired ledger state (*cdfForgerLgrState*) – Ledger state acquired (*TraceLedgerState*), relative to block context acquisition

Acquired ledger view (*cdfForgerLgrView*) – Ledger view acquired (*TraceLedgerView*), relative to ledger state acquisition

Mempool snapshotting (*cdfForgerMemSnap*) – Time spent taking a mempool snapshot (*TraceForgingMempoolSnapshot*), relative to ledger ticking conclusion

Forged to sending (*cdfForgerSend*) – Time between block forging completion and begin-of-sending (*TraceBlockFetchServerSendBlock*)

Started forge loop iteration (*cdfForgerStart*) – Forge loop iteration delay (*TraceStartLeadershipCheck*), relative to slot start

Ledger ticking (*cdfForgerTicked*) – Time spent ticking the ledger state (*TraceForgeTickedLedgerState*), relative to leadership check completion

Fetch to adopted (*cdfPeerAdoption*) – Time until the peer adopts the block (*TraceAddBlockEvent.AddedToCurrentChain*), since it was fetched

Fetch to announced (*cdfPeerAnnounce*) – Time it took a peer to announce the block (*ChainSyncServerEvent.TraceChainSyncServerUpdate*), since it was fetched

Fetch duration (*cdfPeerFetch*) – Time it took the peer to complete fetching the block (*BlockFetchClient.CompletedBlockFetch*), after having requested it

First peer fetch (*cdfPeerFetchFirst*) – Time it took for the fastest peer to fetch the block (BlockFetchClient.CompletedBlockFetch), since block's slot start

First peer notice (*cdfPeerNoticeFirst*) – Time it took for the fastest peer to notice the block (ChainSyncClientEvent.TraceDownloadedHeader), since block's slot start

Notice to fetch request (*cdfPeerRequest*) – Time it took the peer to request the block body (BlockFetchClient.SendFetchRequest), after it have seen its header

Fetches to sending (*cdfPeerSend*) – Time until the peer started sending the block (BlockFetchServer.SendBlock), since it was fetched

Cluster performance metrics

RTS alloc rate (*Alloc*) – RTS-reported allocation rate, MB/sec

Process CPU usage (*CentiCpu*) – Kernel-reported CPU process usage, % of a single core

RTS GC CPU usage (*CentiGC*) – RTS-reported GC CPU usage, % of a single core

RTS Mutator CPU usage (*CentiMut*) – RTS-reported mutator CPU usage, % of a single core

Filesystem reads (*FsRd*) – Number of bytes which this process really did cause to be fetched from the storage layer, per second

Filesystem writes (*FsWr*) – Number of bytes which this process caused to be sent to the storage layer, modulo truncate(), per second

Major GCs (*GcsMajor*) – Major garbage collection RTS events

Minor GCs (*GcsMinor*) – Minor garbage collection RTS events

RTS heap size (*Heap*) – RTS-reported heap size, MB

RTS live GC dataset (*Live*) – RTS-reported GC live data size, MB

Network reads (*NetRd*) – Network reads, kB/sec

Network writes (*NetWr*) – Network writes, kB/sec

Kernel RSS (*RSS*) – Kernel-reported RSS (Resident Set Size) of the process, MB

Block context acquisition delay (*cdfBlkCtx*) – Block context acquired (TraceBlockContext), relative to forge loop beginning

Interblock gap (*cdfBlockGap*) – Time between blocks

Chain density (*cdfDensity*) – Block/slot ratio, for the last 'k' slots

Leadership check duration (*cdfLeading*) – Leadership check duration (TraceNodeIsNotLeader, TraceNodeIsLeader), relative to ledger view acquisition

Ledger state acquisition delay (*cdfLgrState*) – Ledger state acquired (TraceLedgerState), relative to block context acquisition

Ledger view acquisition delay (*cdfLgrView*) – Ledger view acquired (TraceLedgerView), relative to ledger state acquisition

CPU 85% spans (*cdfSpanLensCpu*) – Length of over-85% CPU usage peaks, slots

CPU spans at Ep boundary (*cdfSpanLensCpuEpoch*) – Length of over-85% CPU usage peaks, starting at epoch boundary, slots

Forge loop tardiness (*cdfStarted*) – Forge loop iteration start delay (TraceStartLeadershipCheck), relative to slot start

Forge loop starts (*cdfStarts*) – For any given slot, how many forging loop starts were registered