

10.5.4 against 10.5-baseline

value-only workload

Michael Karg, Cardano Performance & Tracing
2026-02-09

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-baseline` (Conway) and `10.5.4` (Conway) relative to `10.5-baseline` (Conway), under value-only workload.

	10.5-baseline	10.5.4
Analysis date	2025-06-28	2026-02-06
Cluster system start date	2025-06-27	2026-02-05
Cluster system start time	19:37:29	18:52:16
Identifier	10.5-baseline	10.5.4
Run batch	1050fix	1054
GHC version	9.6.5	9.6.5
cardano-node version	10.5.0	10.5.4
ouroboros-consensus version	0.27.0.0	0.27.0.0
ouroboros-network version	0.21.2.0	0.21.6.1
cardano-ledger-core version	1.17.0.0	1.17.0.0
plutus-core version	1.45.0.0	1.45.0.0
cardano-crypto version	1.3.0	1.3.0
cardano-prelude version	0.2.1.0	0.2.1.0
cardano-node git	28152ff	b0a1259
ouroboros-consensus git	8e3afe1	8e3afe1
ouroboros-network git	879683d	6275b8f
cardano-ledger-core git	a9e78ae	a9e78ae
plutus-core git	ba16ec6	ba16ec6
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	3869633.0769	4050952.5192
Log objects analysed per host	1847240.5192	1920901.8846
Host run time, s	63879.9	63895.6
Host log line rate, Hz	60.577	63.400
Total log objects analysed	96056507	99886898
Run time, s	63885	63901
Analysed run duration, s	48017	48025
Run time efficiency	0.75	0.75
Node start spread, s	4.6618552	4.8893897
Node stop spread, s	4.2671434	4.8149166
Slots analysed	48013	48022
Blocks analysed	2245	2327
Blocks rejected	857	875

Analysis

Resource Usage

	10.5-baseline	10.5.4	Δ	Δ%
Forge loop starts, units	0.99868	0.99864	-0.000	0
Process CPU usage, %	5.5921	5.7687	0.177	3
RTS GC CPU usage, %	0.3552	0.3655	0.010	3
RTS Mutator CPU usage, %	5.2358	5.4016	0.166	3
Major GCs, events	0.0008	0.0008	0.000	0
Minor GCs, events	1.3885	1.4147	0.026	2
Kernel RSS, MiB	8994.	8735.4	-258.557	-3
RTS heap size, MiB	8932.5	8674.2	-258.281	-3
RTS live GC dateset, MiB	4041.3	3886.2	-155.084	-4
RTS alloc rate, MiB/s	41.427	42.289	0.862	2
Filesystem reads, KiB/s	0.0498	0.0013	-0.049	-97
Filesystem writes, KiB/s	256.47	256.93	0.464	0.2
CPU 85% spans, slots	6.1078	6.4549	0.347	6

Anomaly control

	10.5-baseline	10.5.4	Δ	Δ%
Blocks per host, blocks	61.923	63.385	1.462	2
Filtered to chained blocks, :	0.7248	0.7263	0.001	0.2
Chained to forged blocks, :	0.9633	0.9718	0.008	0.9
Height & slot battles, blocks	0.00757	0.00731	-0.000	-4
Block size, Bytes	88970	88968	-2.632	-0.0

Forging

	10.5-baseline	10.5.4	Δ	$\Delta\%$
Started forge loop iteration, s	0.00137	0.00105	-0.000	-23
Acquired block context, s	7.1e-5	7.7e-5	0.000	0
Acquired ledger state, s	9.7e-5	0.0001	0.000	0
Acquired ledger view, s	3.1e-5	2.9e-5	-0.000	0
Leadership check duration, s	0.00047	0.00044	-0.000	0
Ledger ticking, s	0.02613	0.02679	0.001	3
Mempool snapshotting, s	0.04459	0.04521	0.001	1
Leadership to forged, s	0.00072	0.00072	-0.000	0
Forged to announced, s	0.00078	0.00078	-0.000	0
Forged to sending, s	0.00596	0.00625	0.000	5
Forged to self-adopted, s	0.06964	0.07128	0.002	2
Slot start to announced, s	0.07425	0.0752	0.001	1

Individual peer propagation

	10.5-baseline	10.5.4	Δ	$\Delta\%$
First peer notice, s	0.07622	0.0772	0.001	1
First peer fetch, s	0.08545	0.0869	0.001	2
Notice to fetch request, s	0.00126	0.00135	0.000	0
Fetch duration, s	0.36309	0.36136	-0.002	-0.5
Fetched to announced, s	0.00101	0.00102	0.000	0
Fetched to sending, s	0.04516	0.04524	0.000	0
Fetched to adopted, s	0.07293	0.07444	0.002	2

End-to-end propagation

	10.5-baseline	10.5.4	Δ	$\Delta\%$
0.50 adoption, s	0.62605	0.62535	-0.001	-0.1
0.80 adoption, s	0.98763	0.98923	0.002	0.2
0.90 adoption, s	1.00532	1.00911	0.004	0.4
0.92 adoption, s	1.00953	1.01384	0.004	0.4
0.94 adoption, s	1.01347	1.01979	0.006	0.6
0.96 adoption, s	1.01981	1.02648	0.007	0.7
0.98 adoption, s	1.02685	1.03601	0.009	0.9
1.00 adoption, s	1.04936	1.05962	0.010	1.0

Appendix A: charts

Cluster performance charts

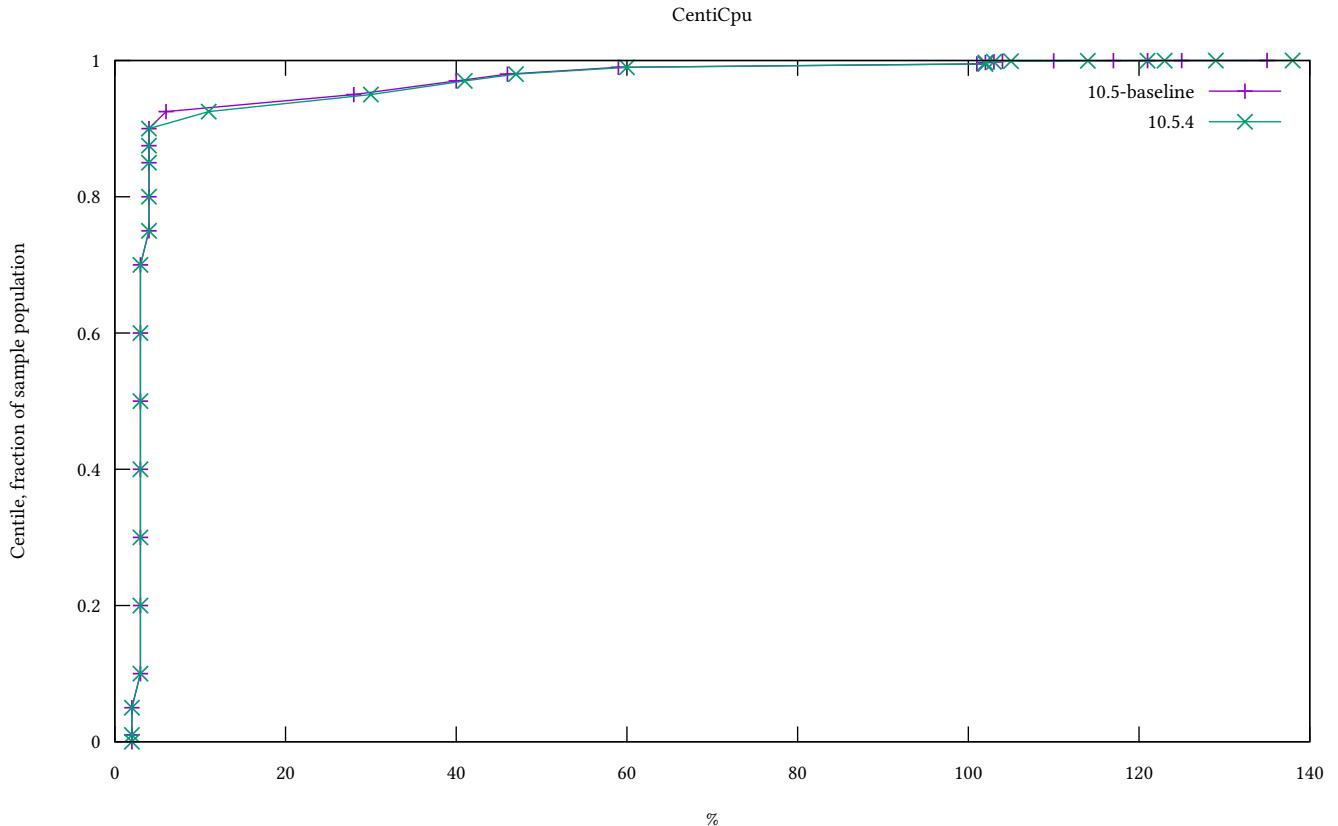


Figure 1: Process CPU usage

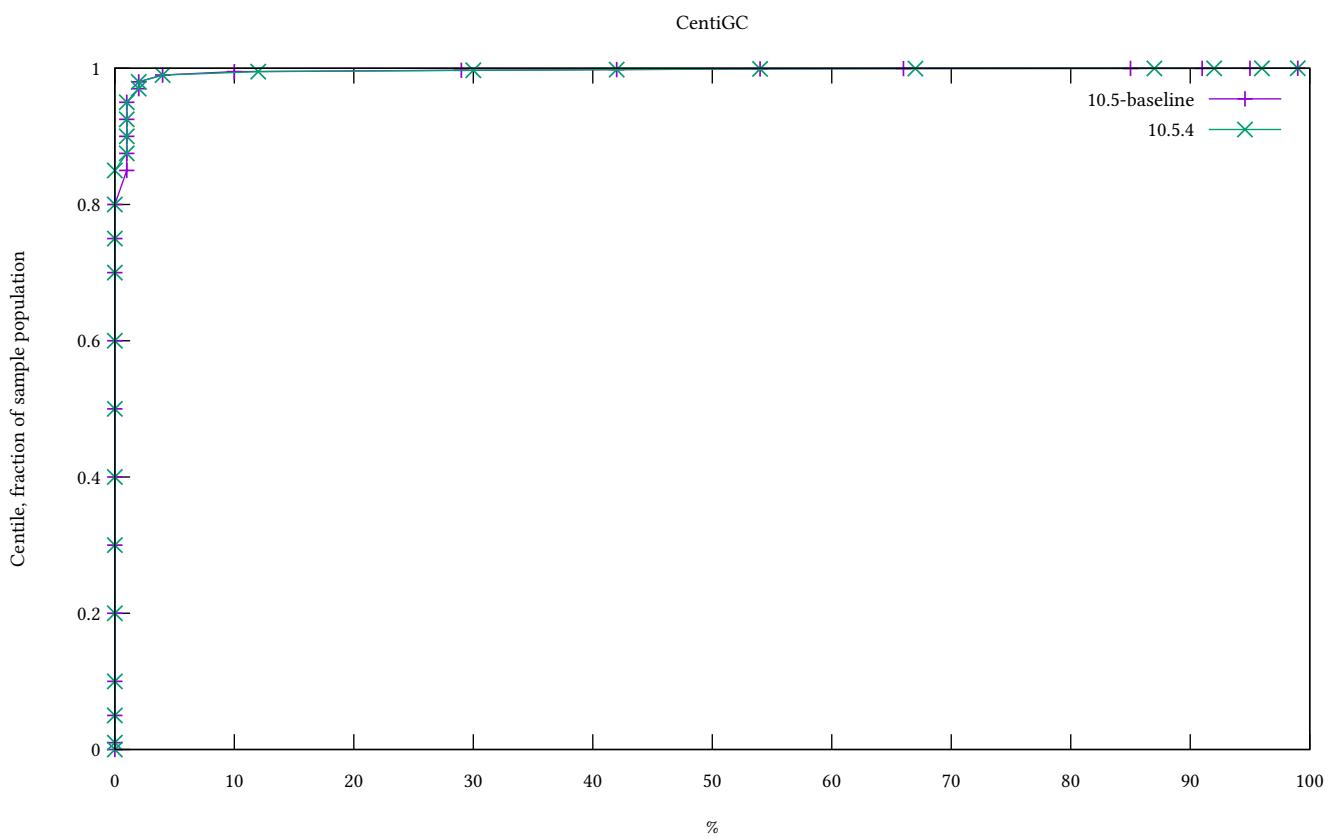


Figure 2: RTS GC CPU usage

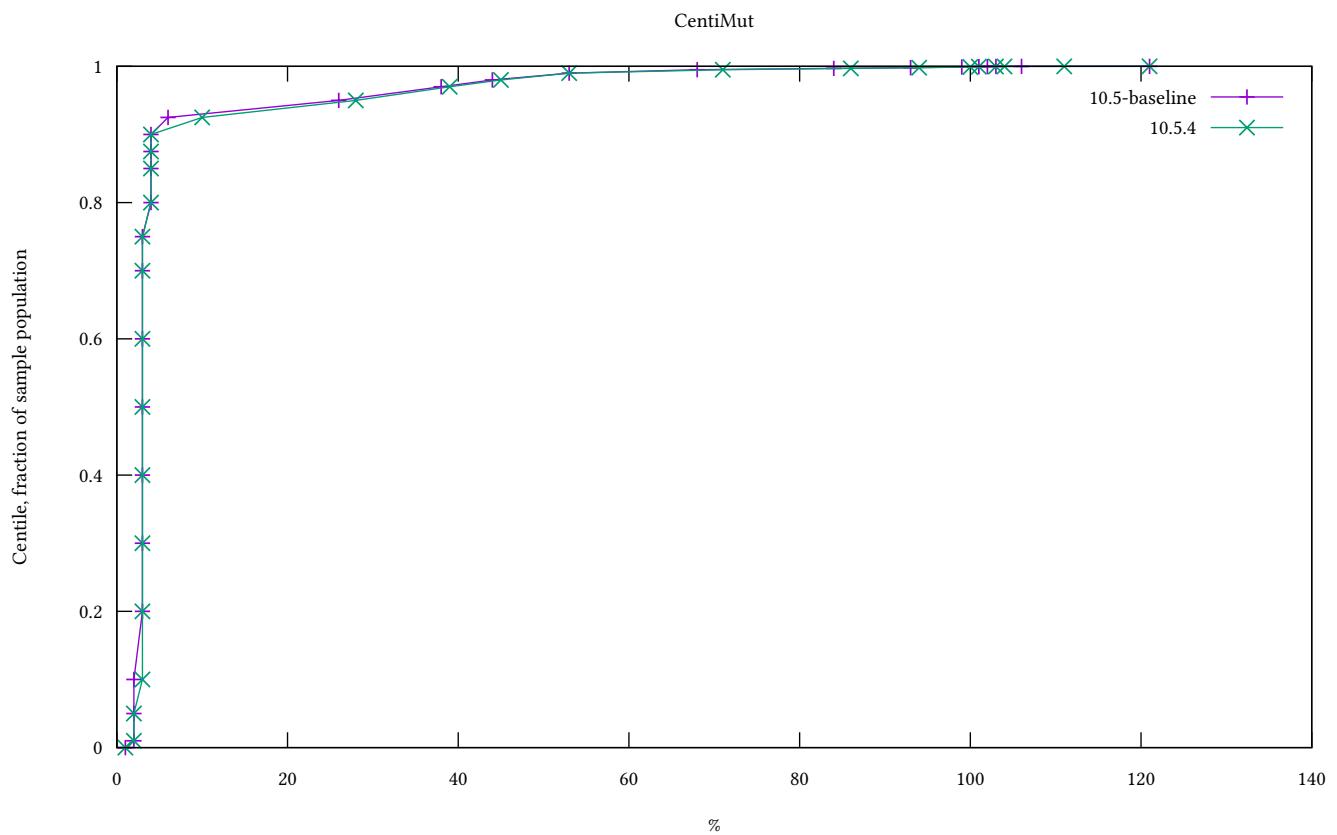


Figure 3: RTS Mutator CPU usage

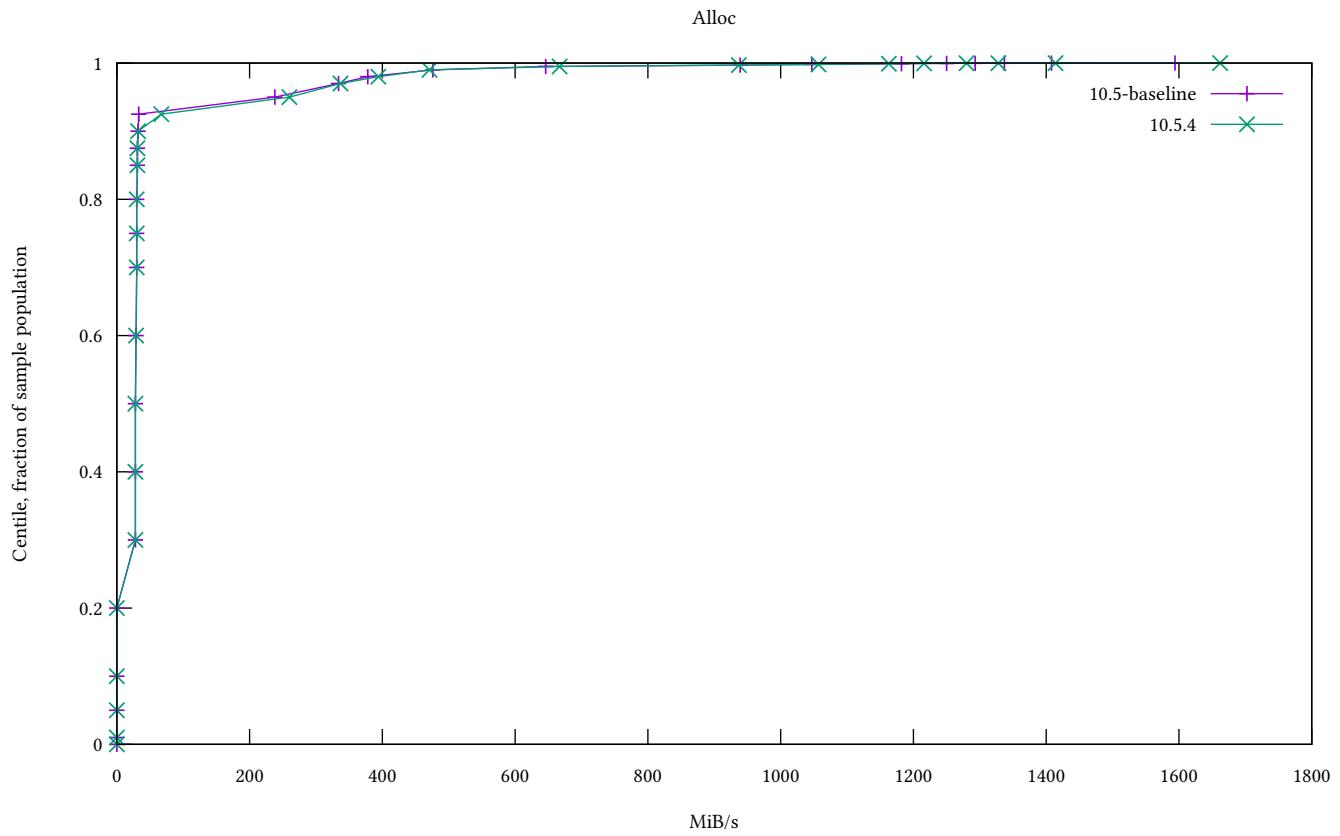


Figure 4: RTS alloc rate

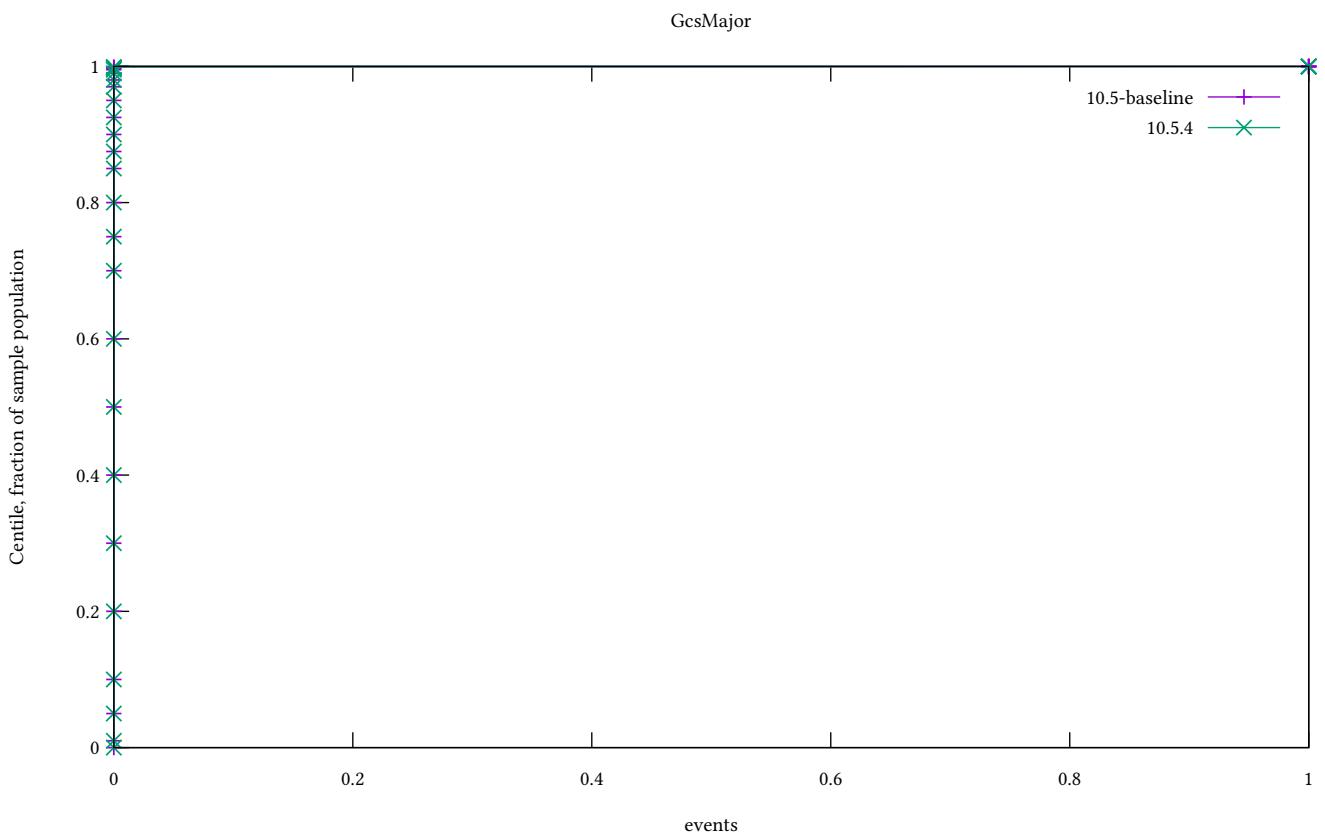


Figure 5: Major GCs

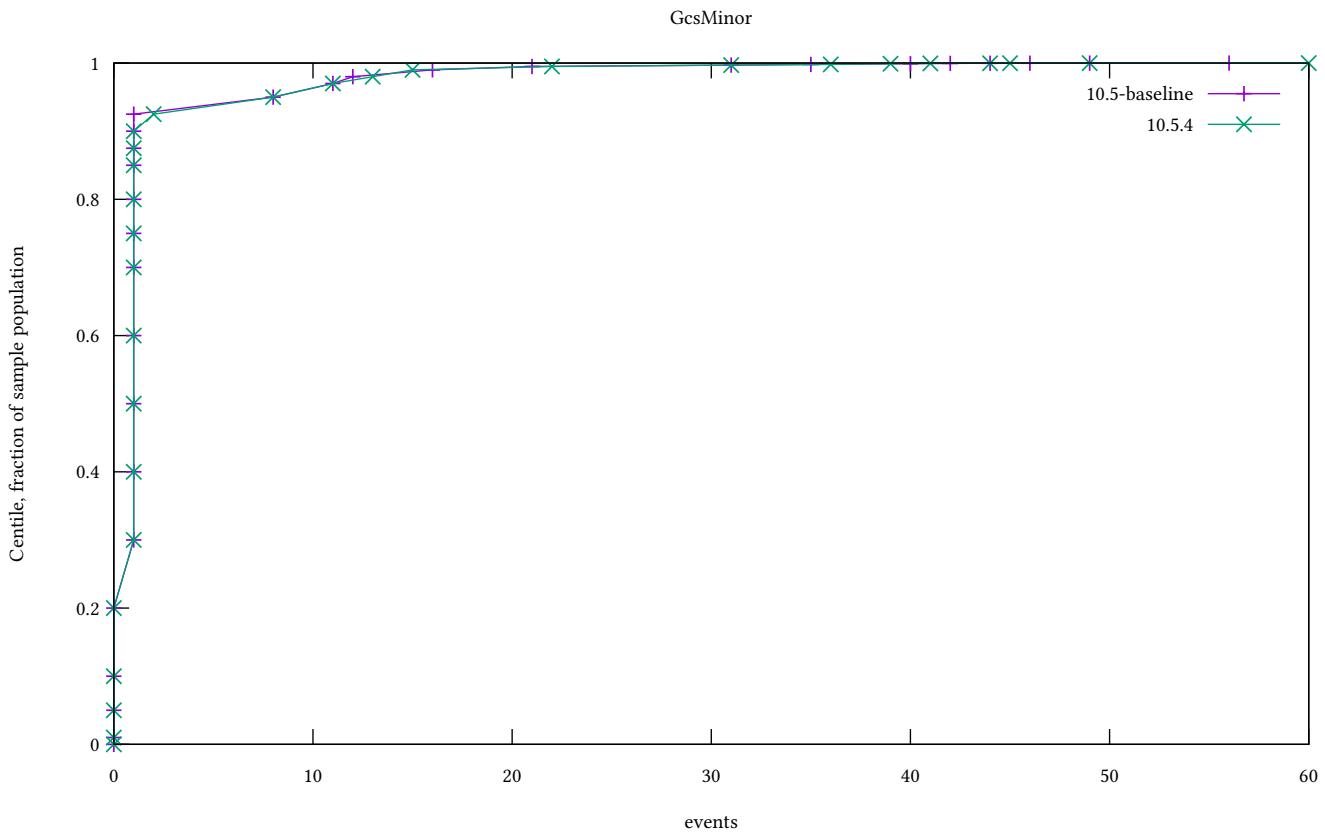


Figure 6: Minor GCs

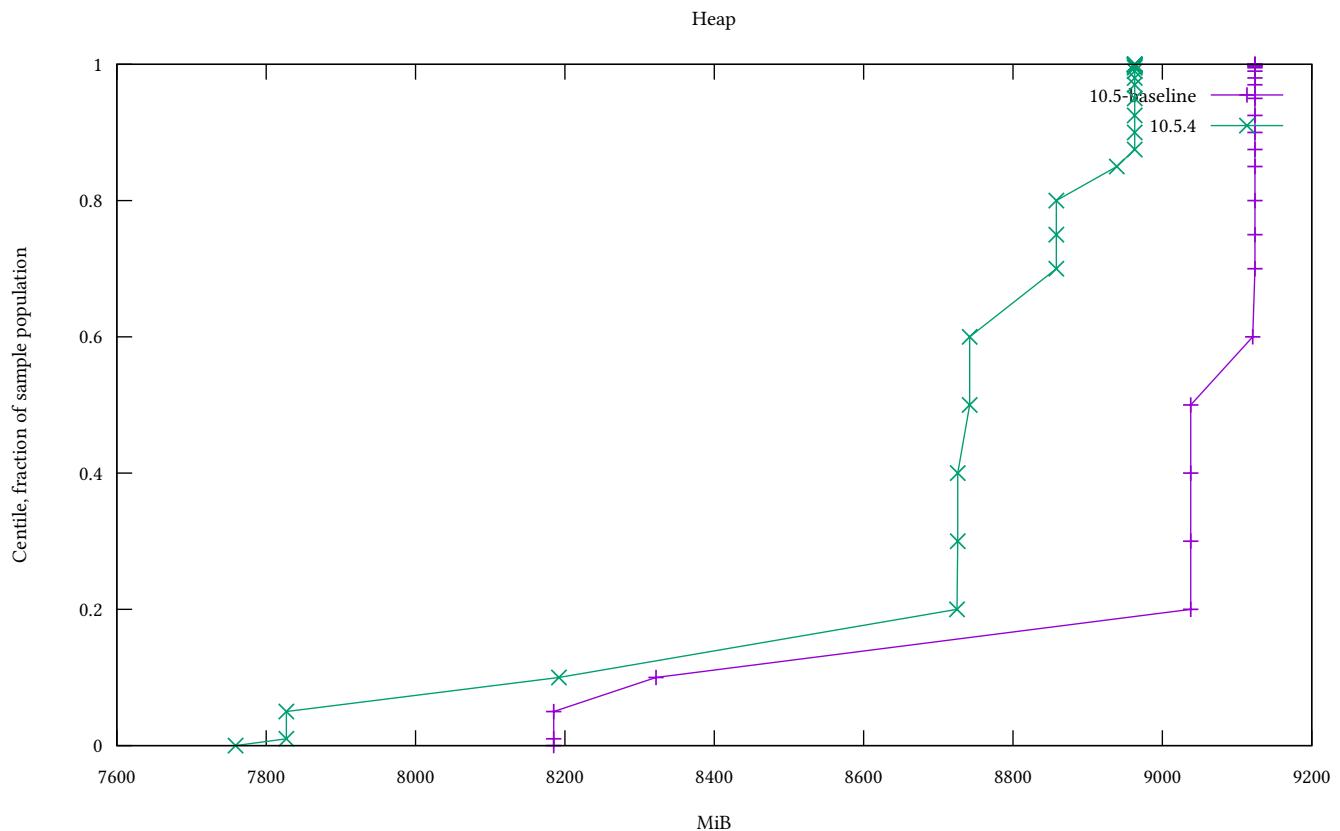


Figure 7: RTS heap size

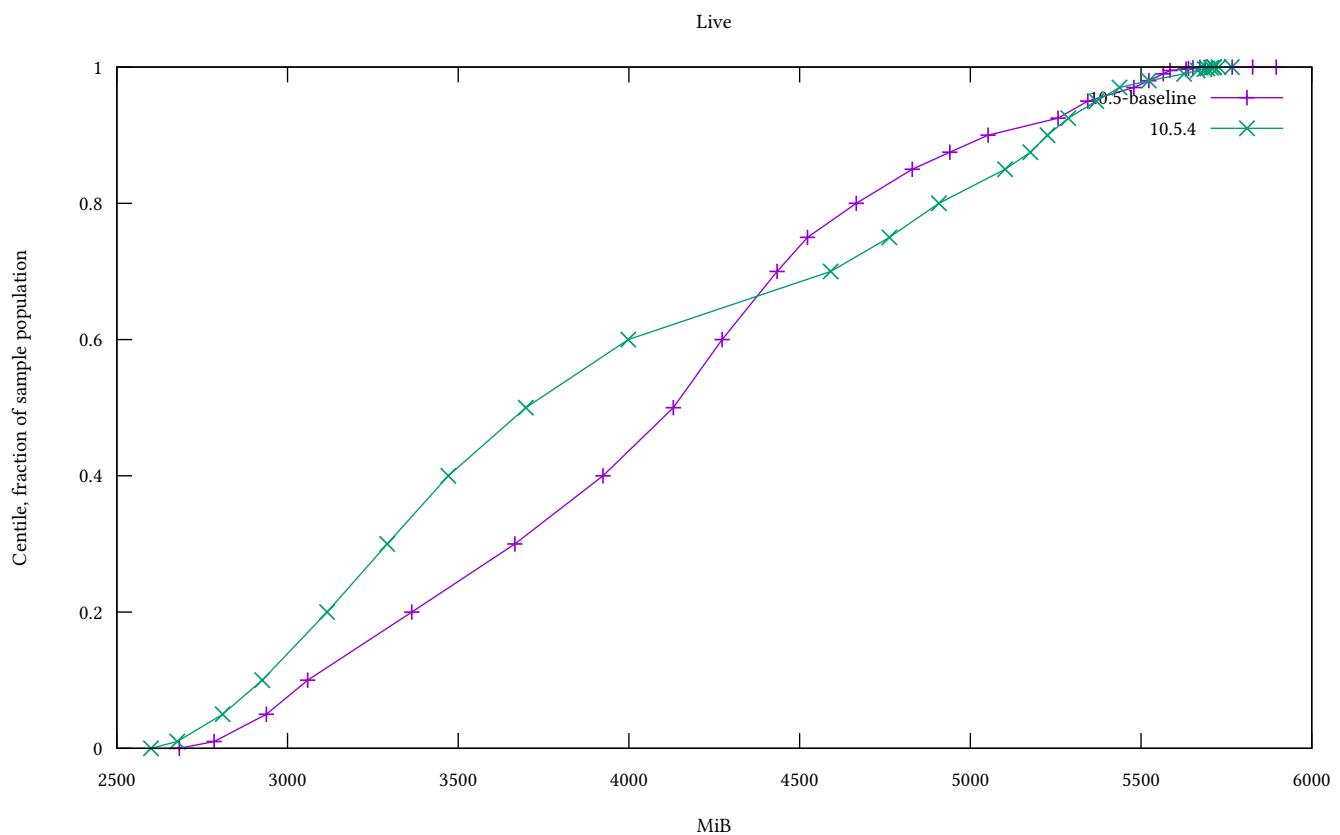


Figure 8: RTS live GC dataset

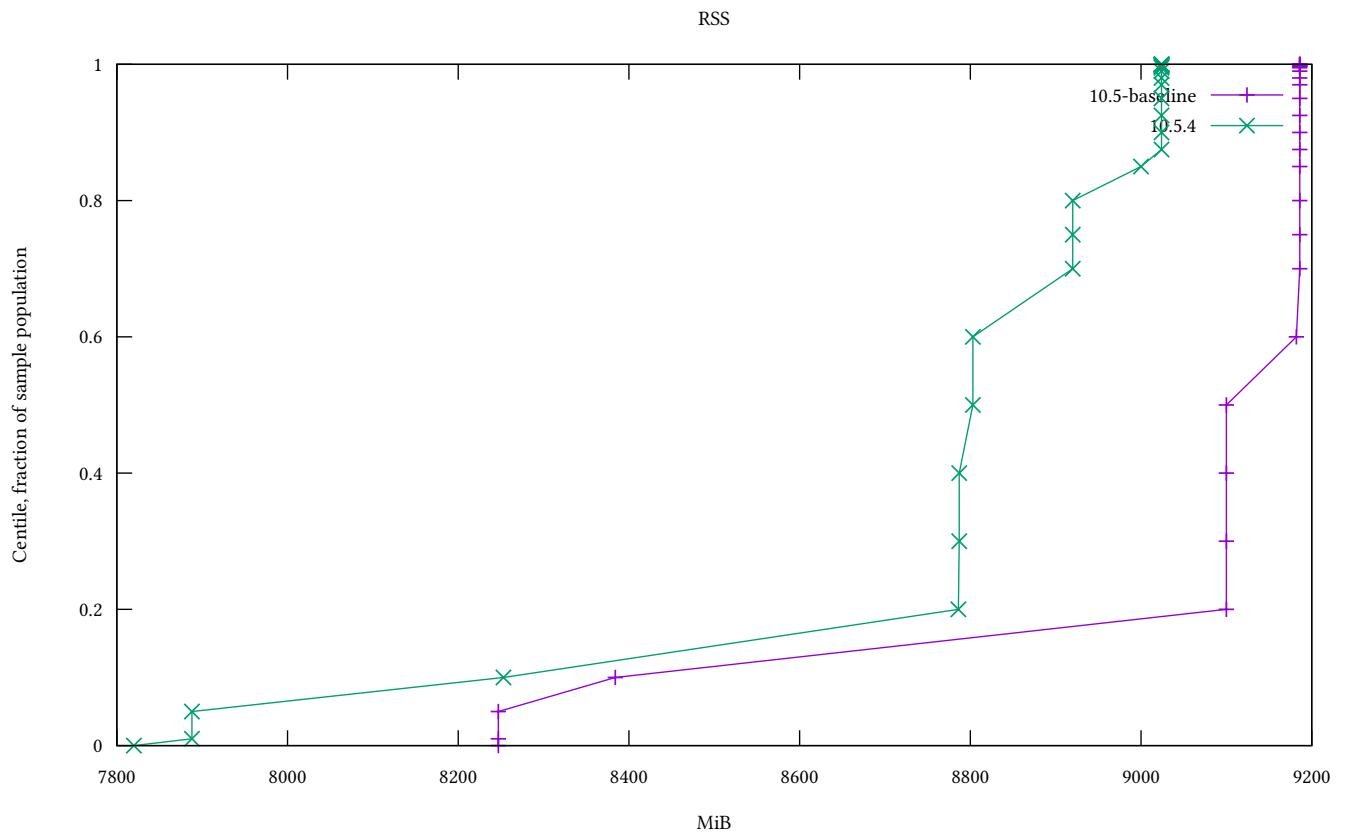


Figure 9: Kernel RSS

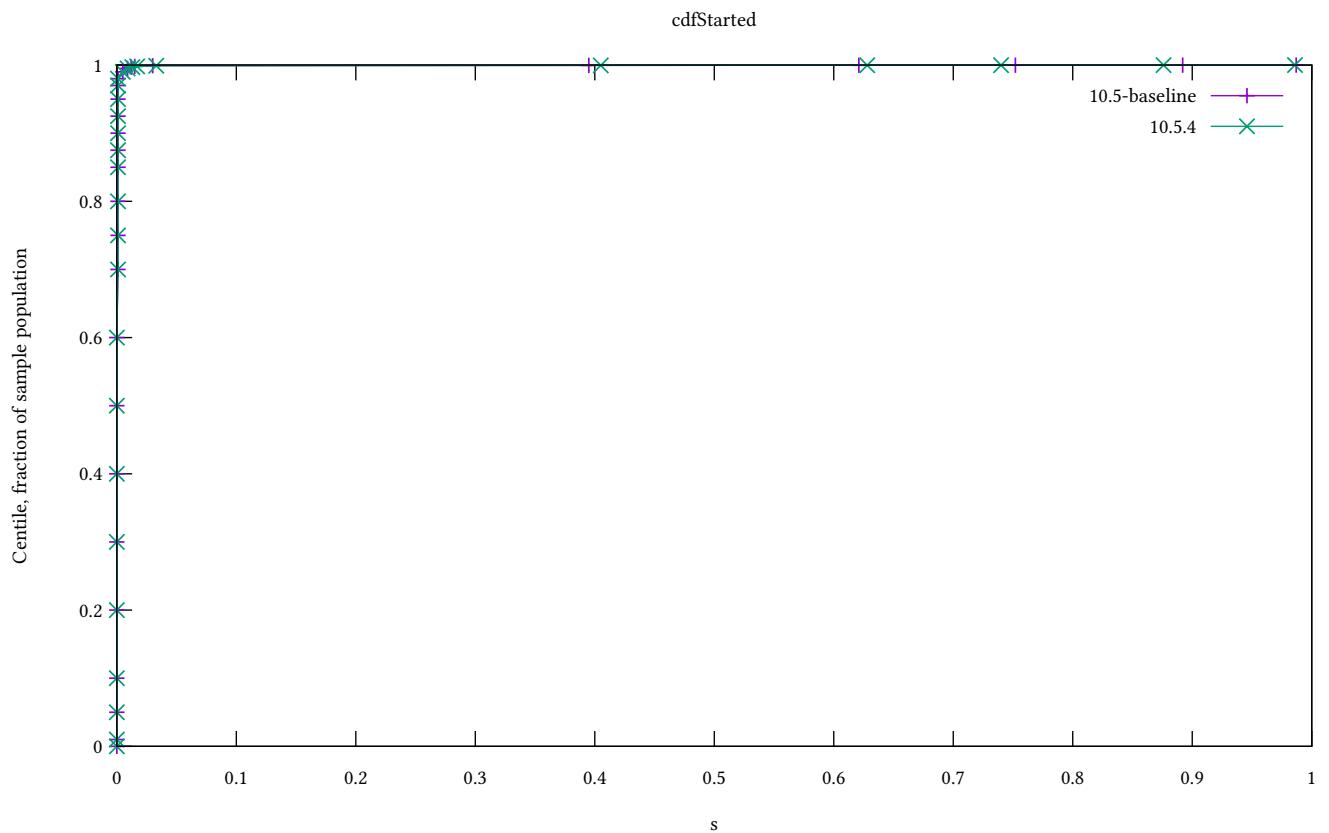


Figure 10: Forge loop tardiness

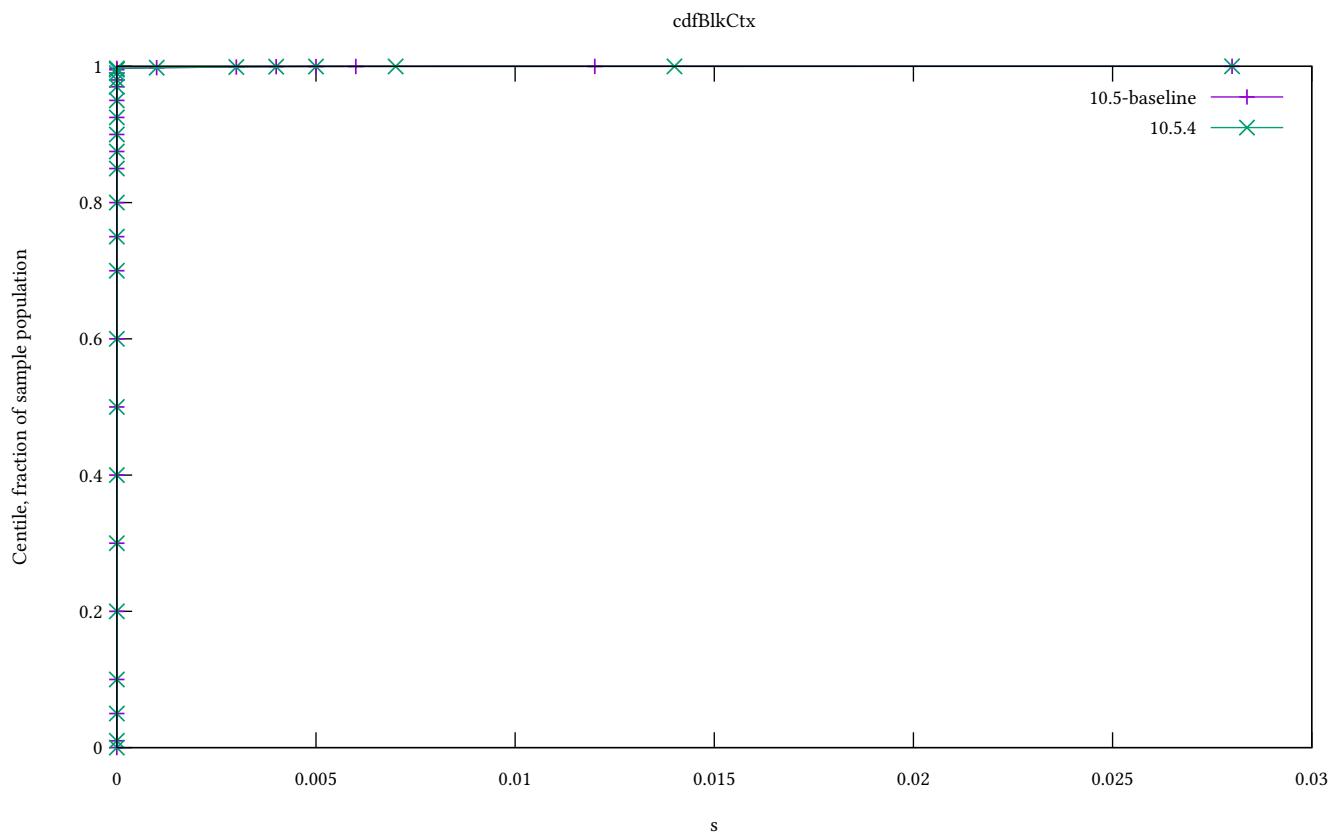


Figure 11: Block context acquisition delay

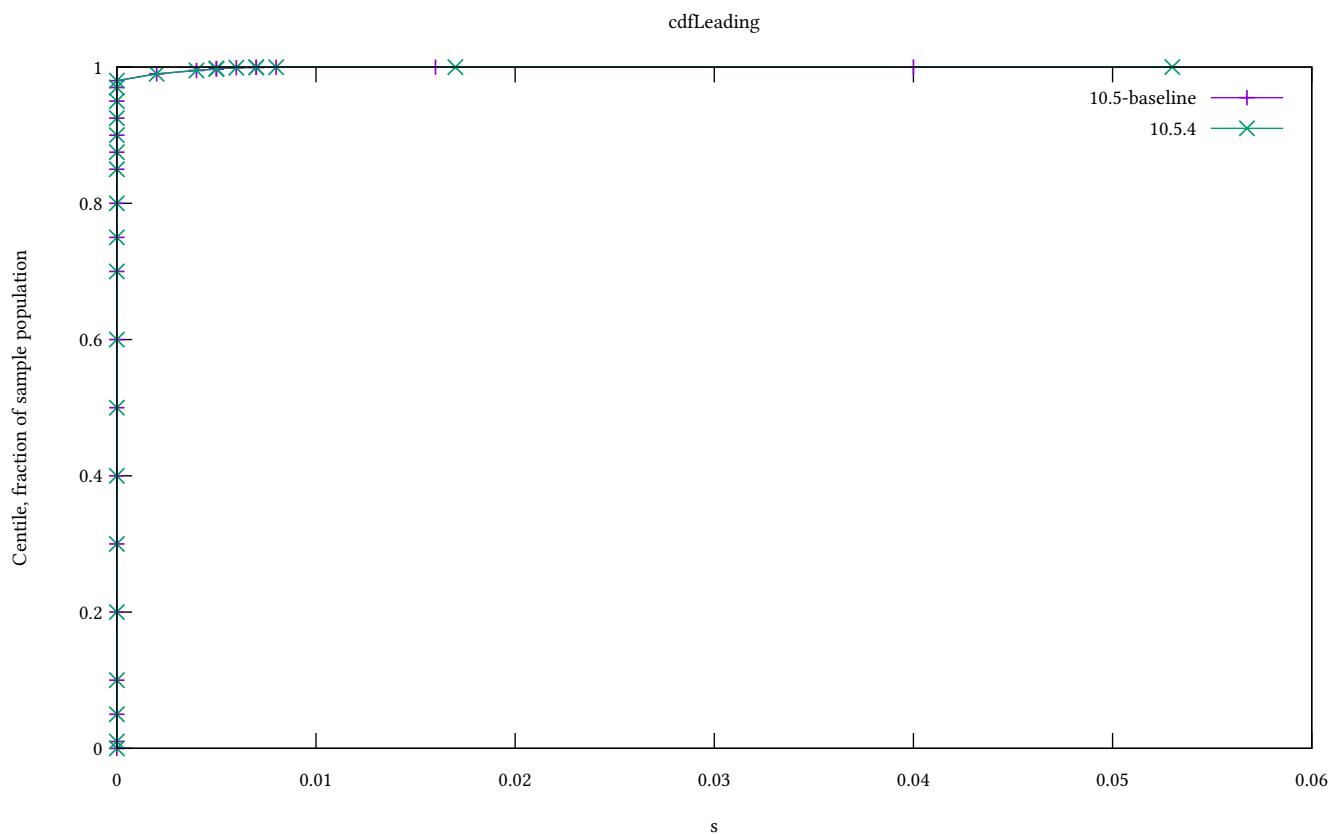


Figure 12: Leadership check duration

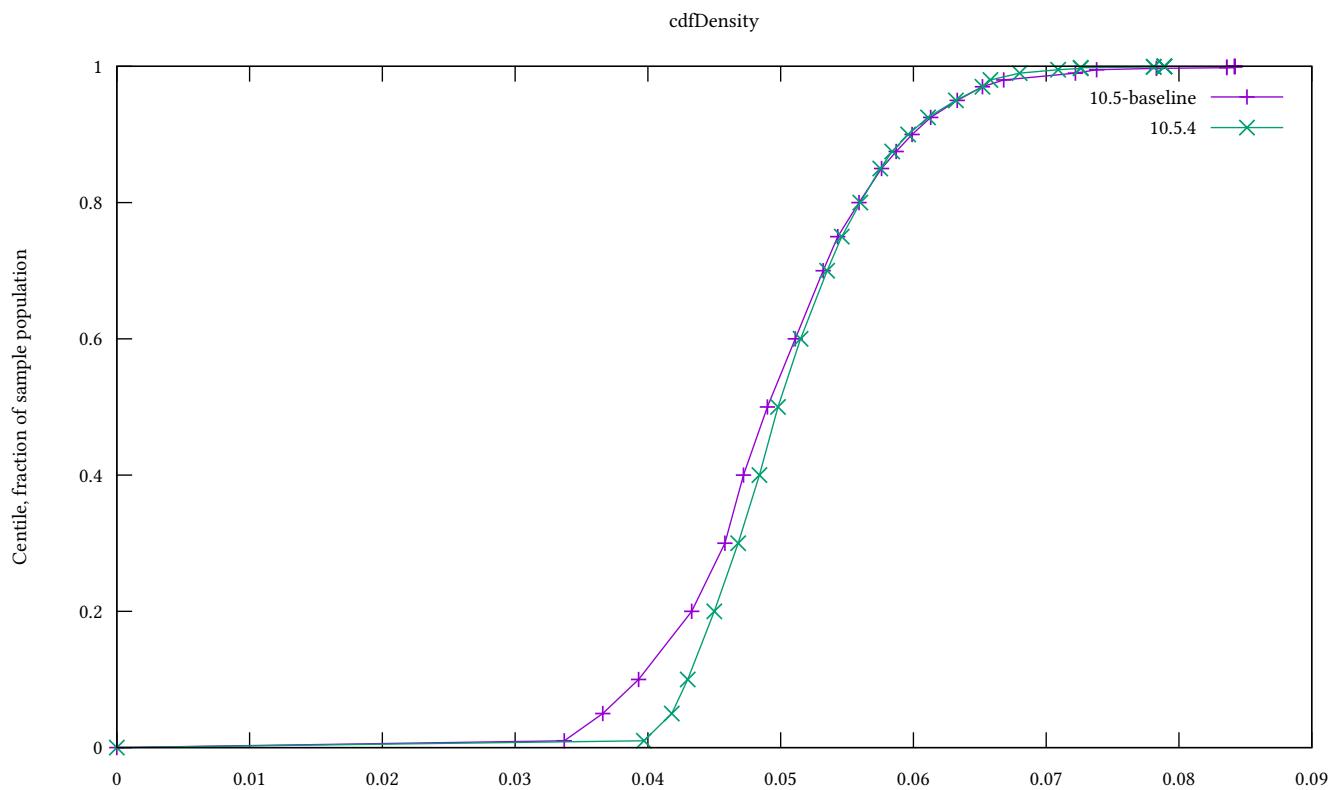


Figure 13: Chain density

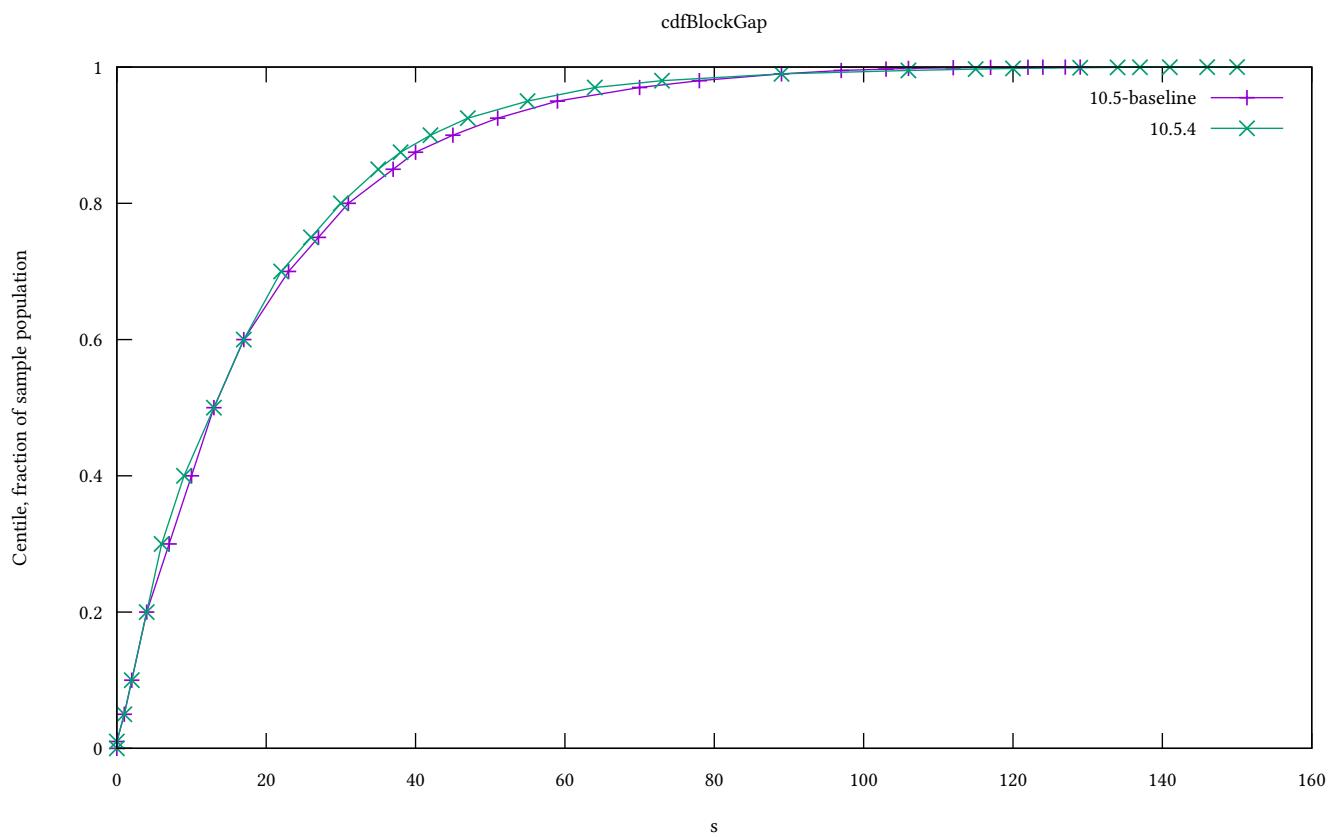


Figure 14: Interblock gap

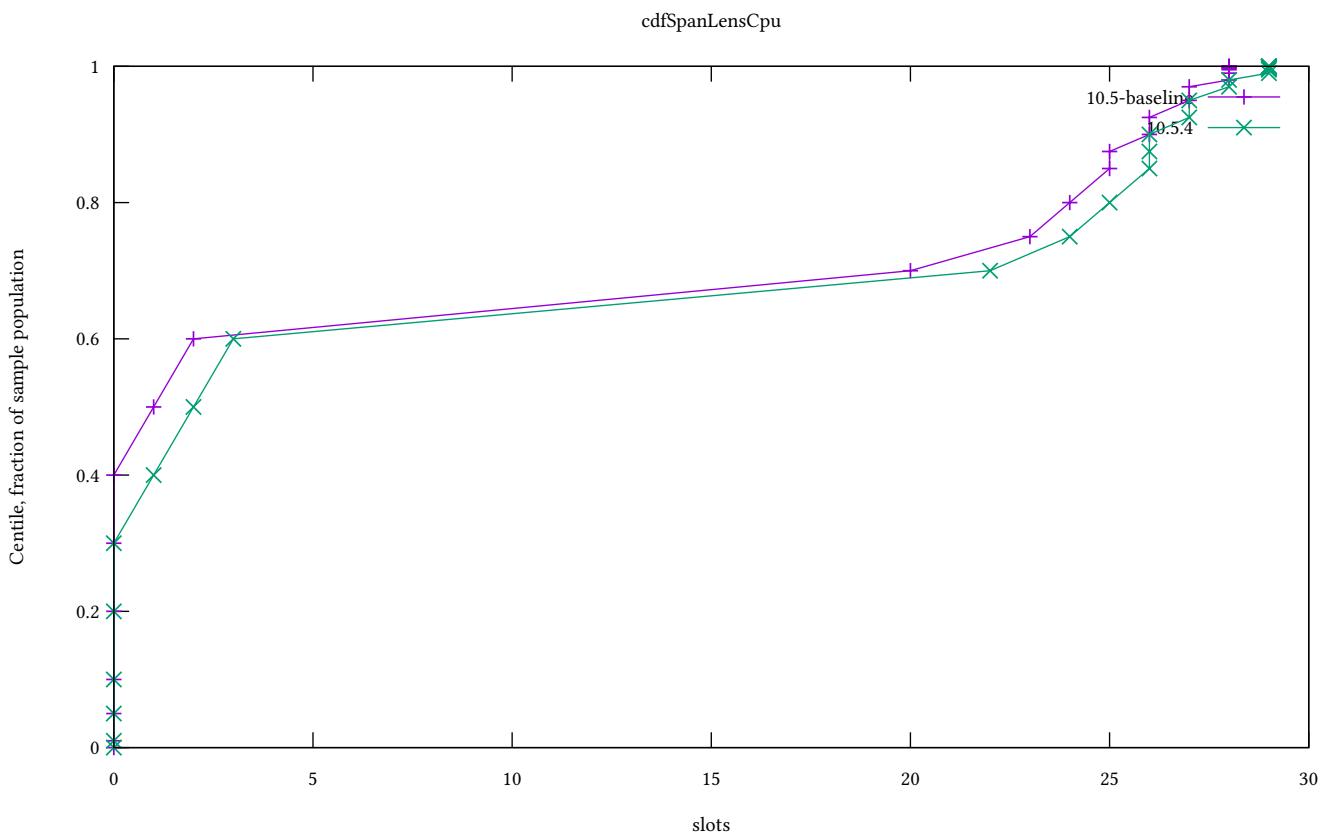


Figure 15: CPU 85% spans

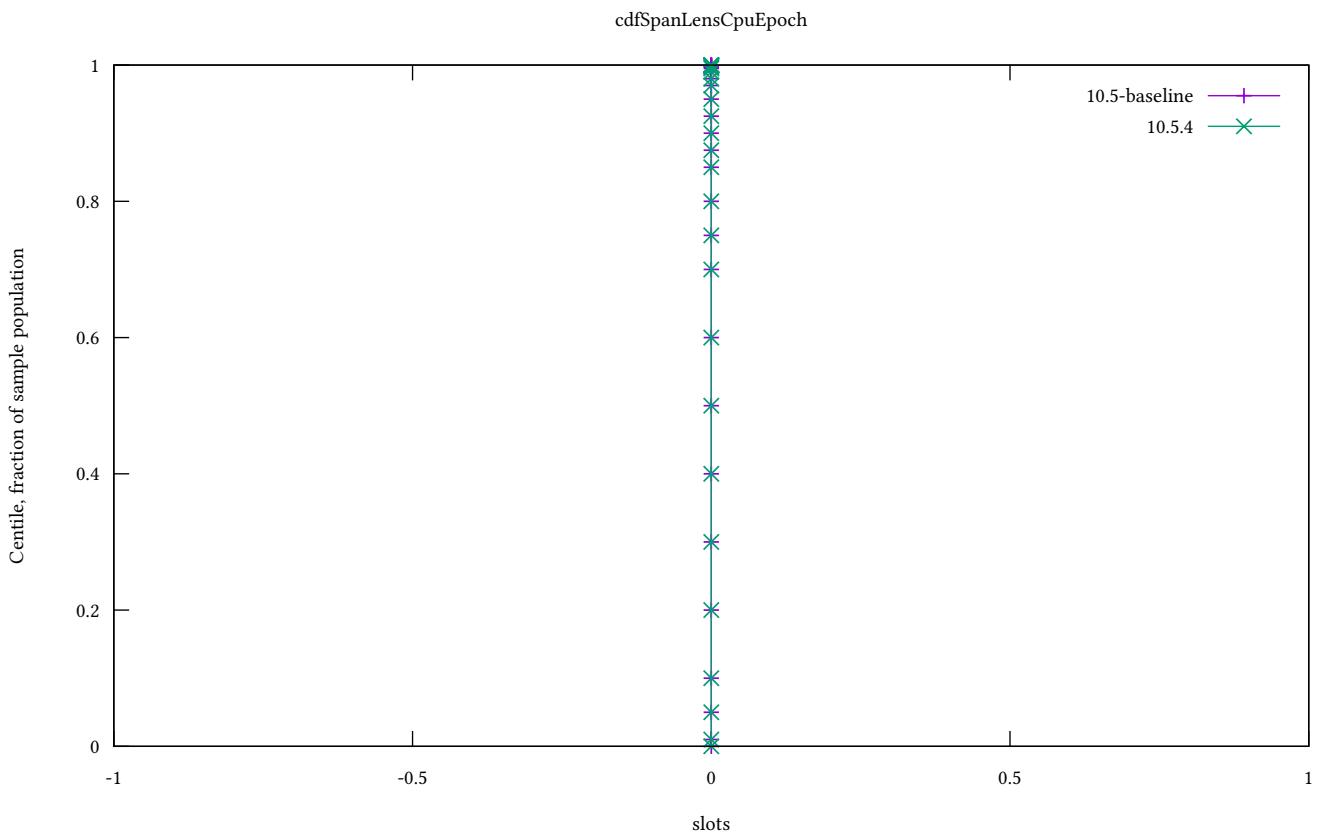


Figure 16: CPU spans at Ep boundary

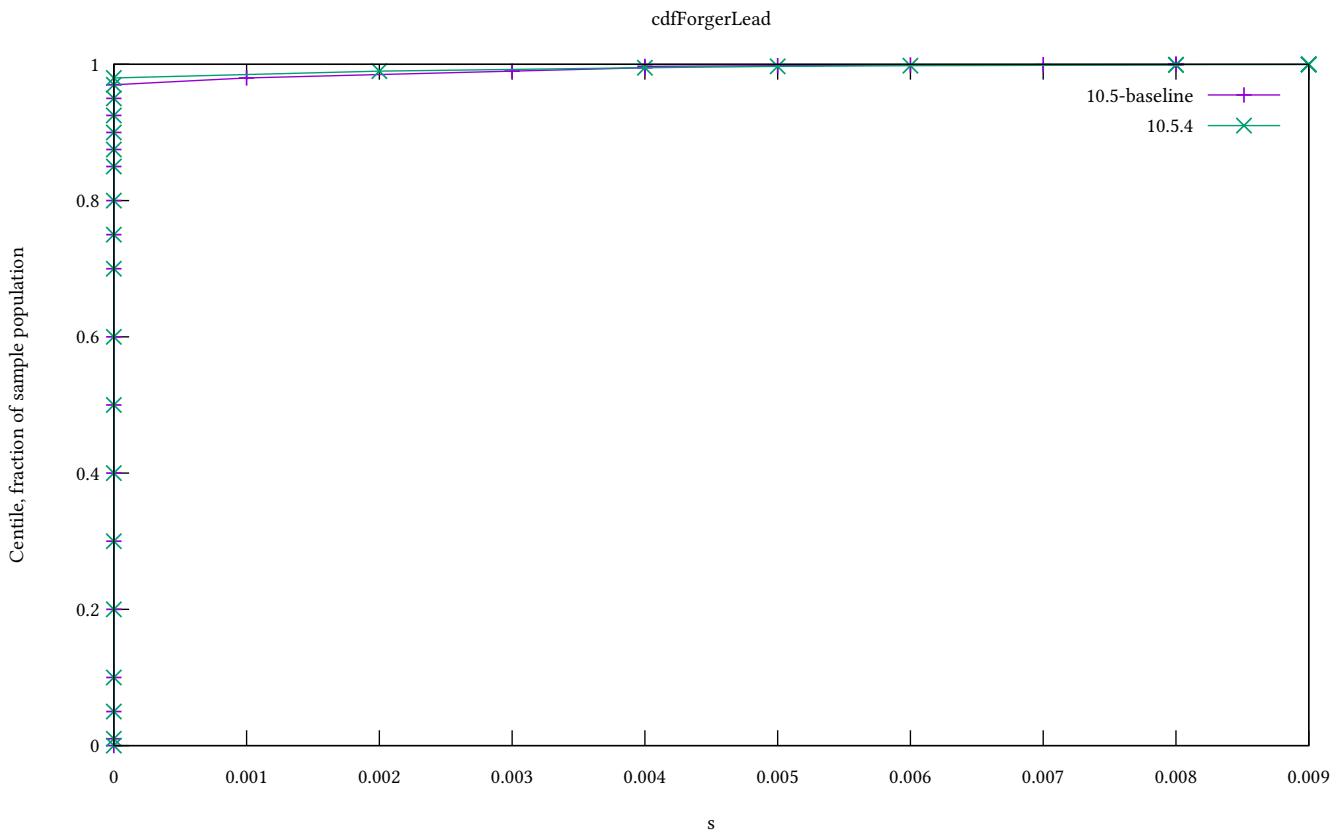


Figure 17: Leadership check duration

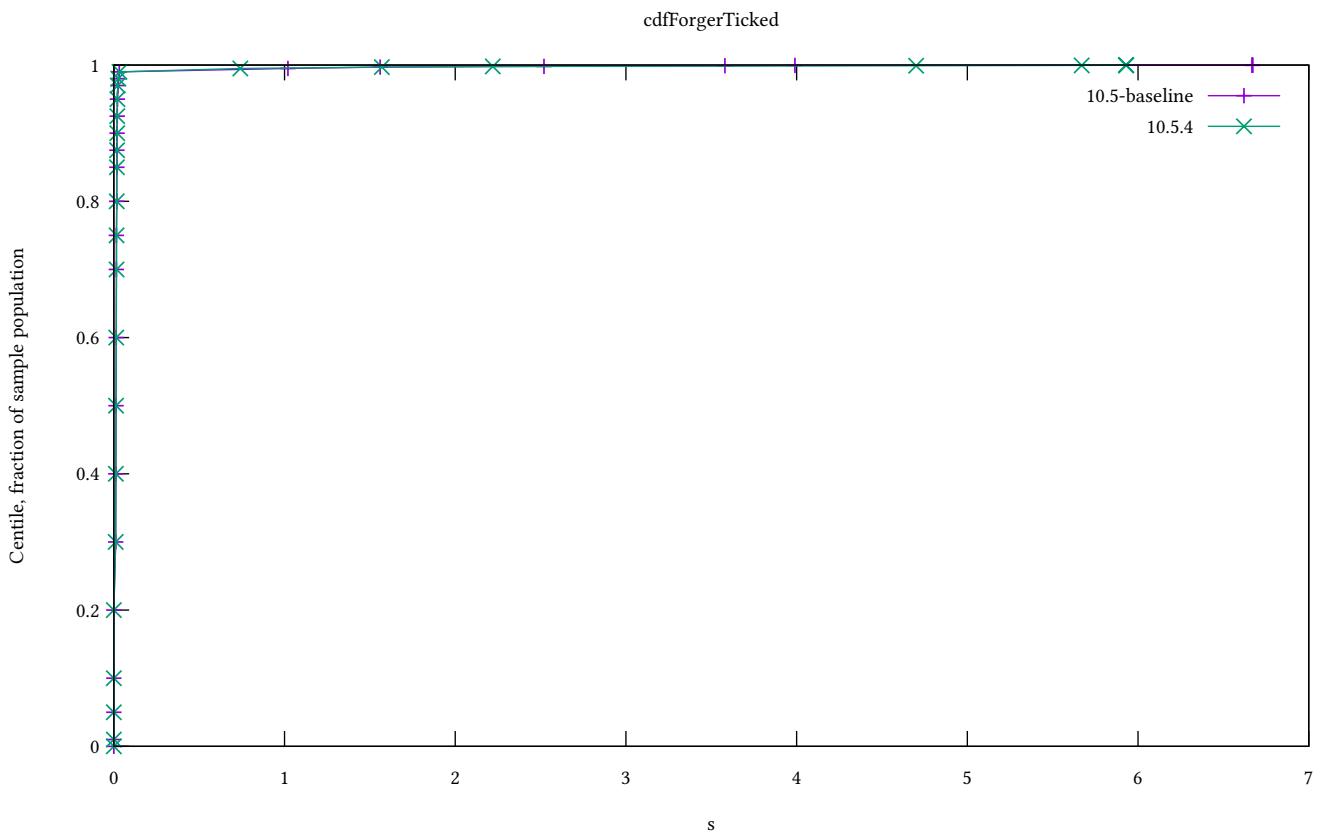


Figure 18: Ledger ticking

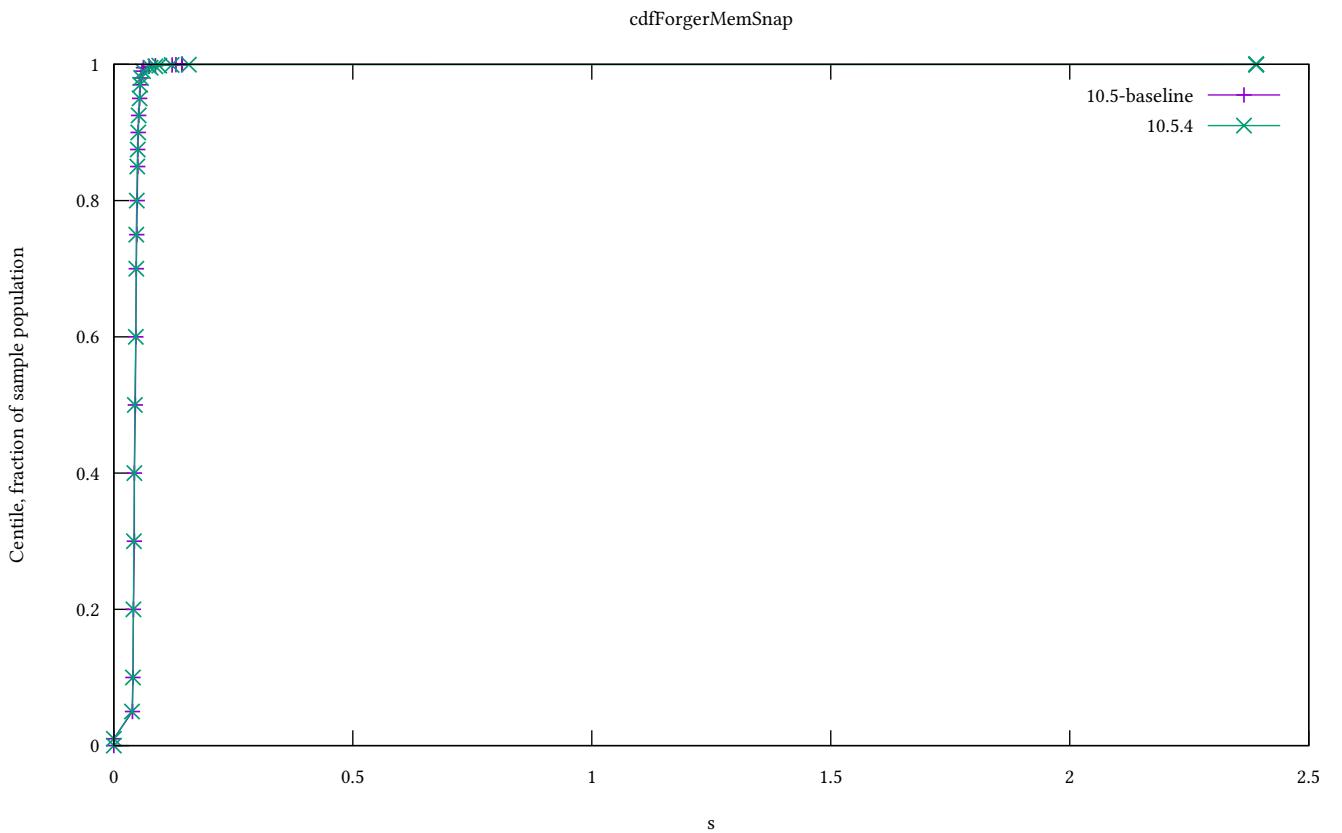


Figure 19: Mempool snapshutting

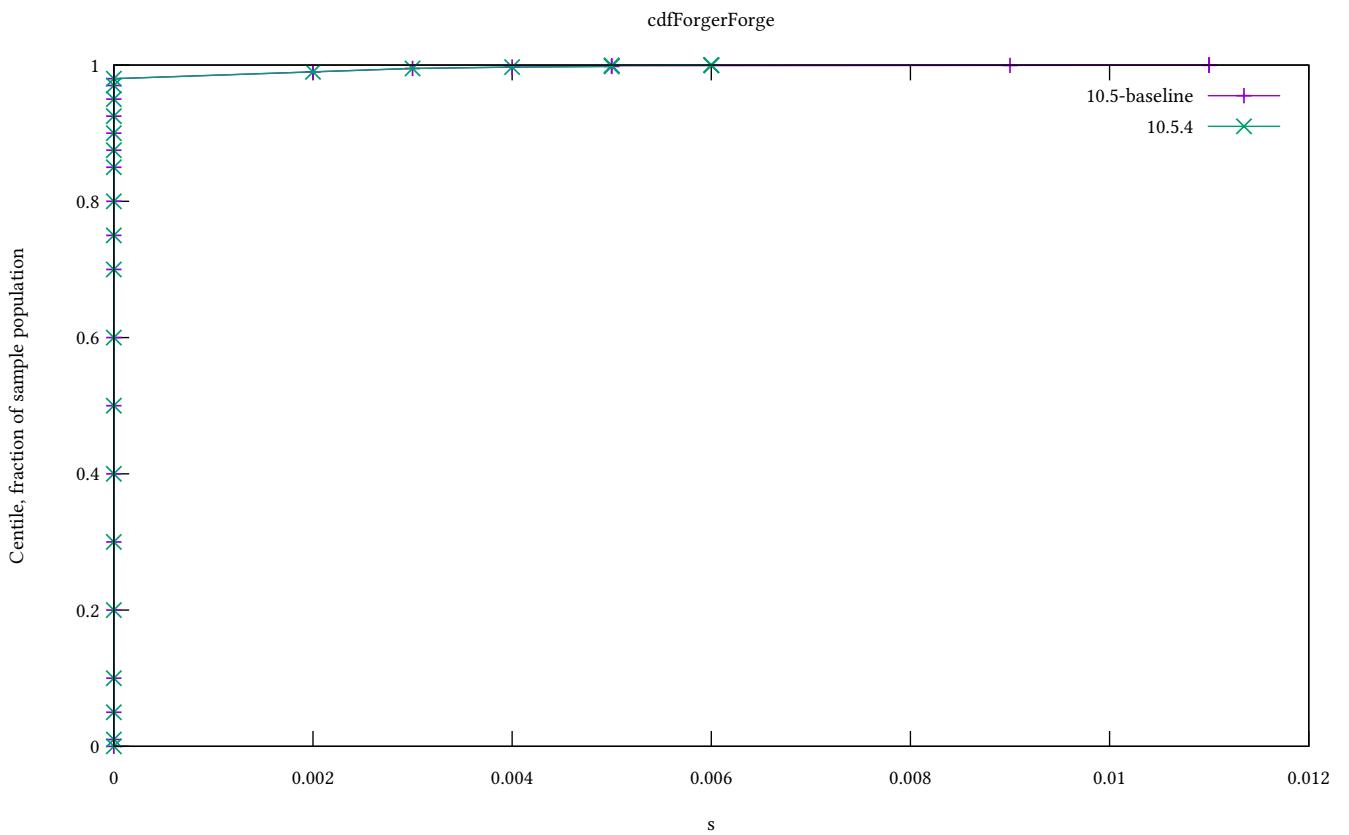


Figure 20: Leadership to forged

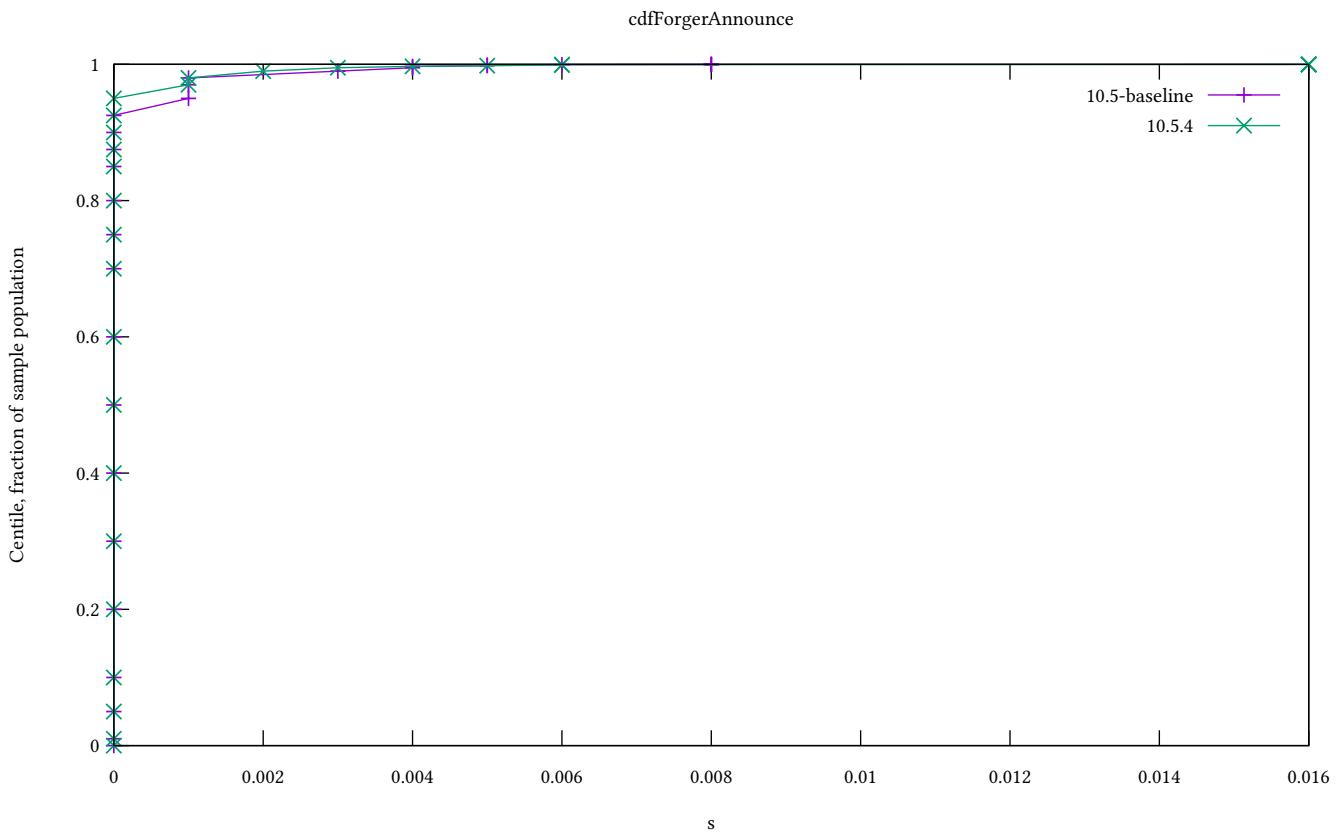


Figure 21: Forged to announced

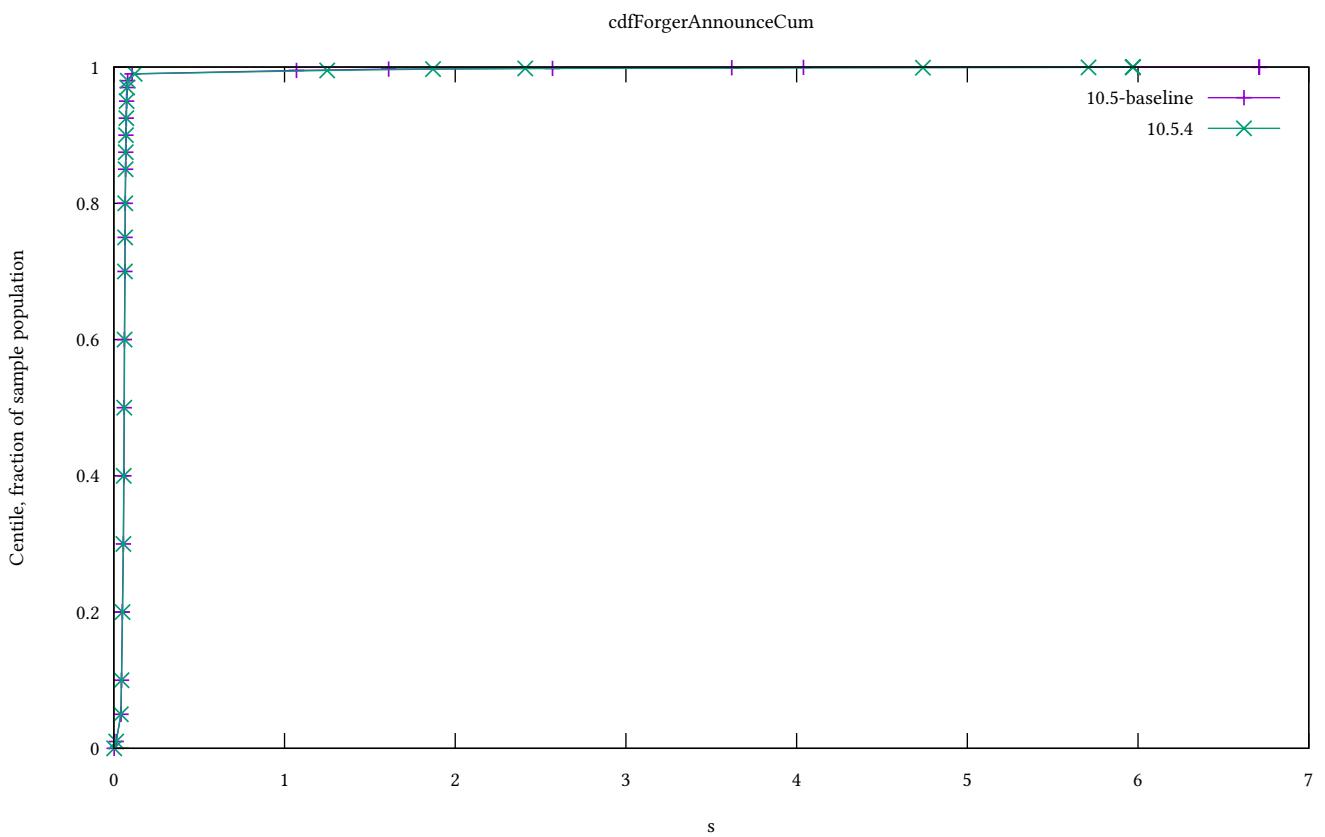


Figure 22: Slot start to announced

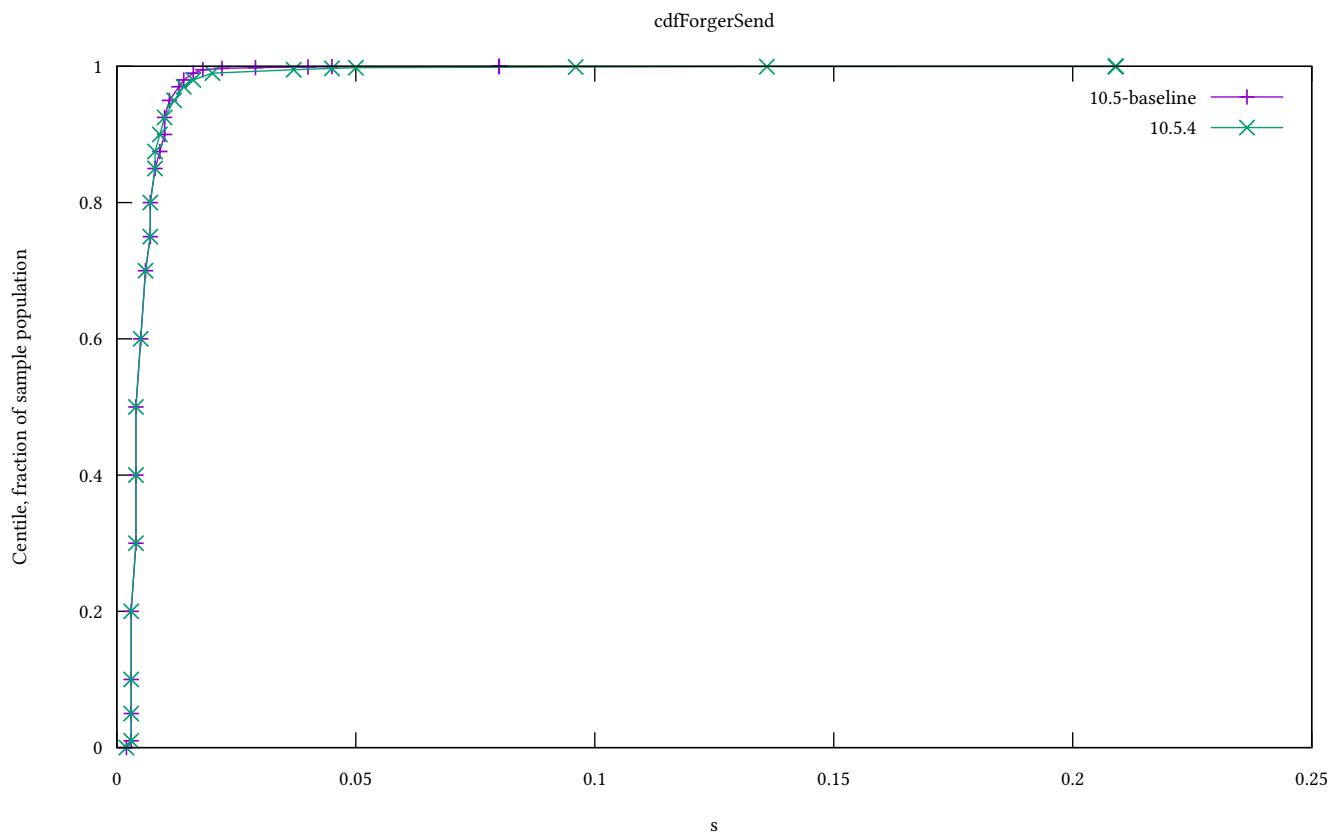


Figure 23: Forged to sending

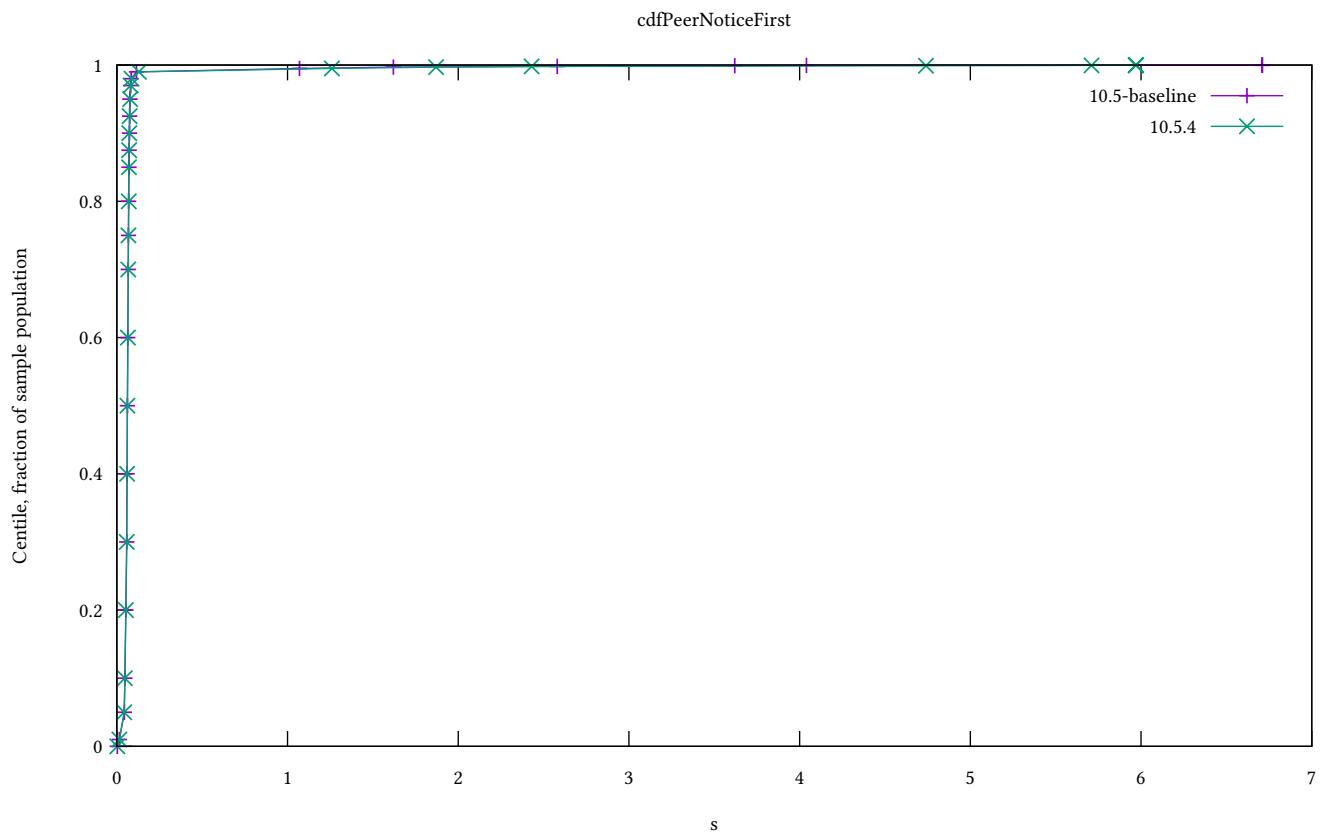


Figure 24: First peer notice

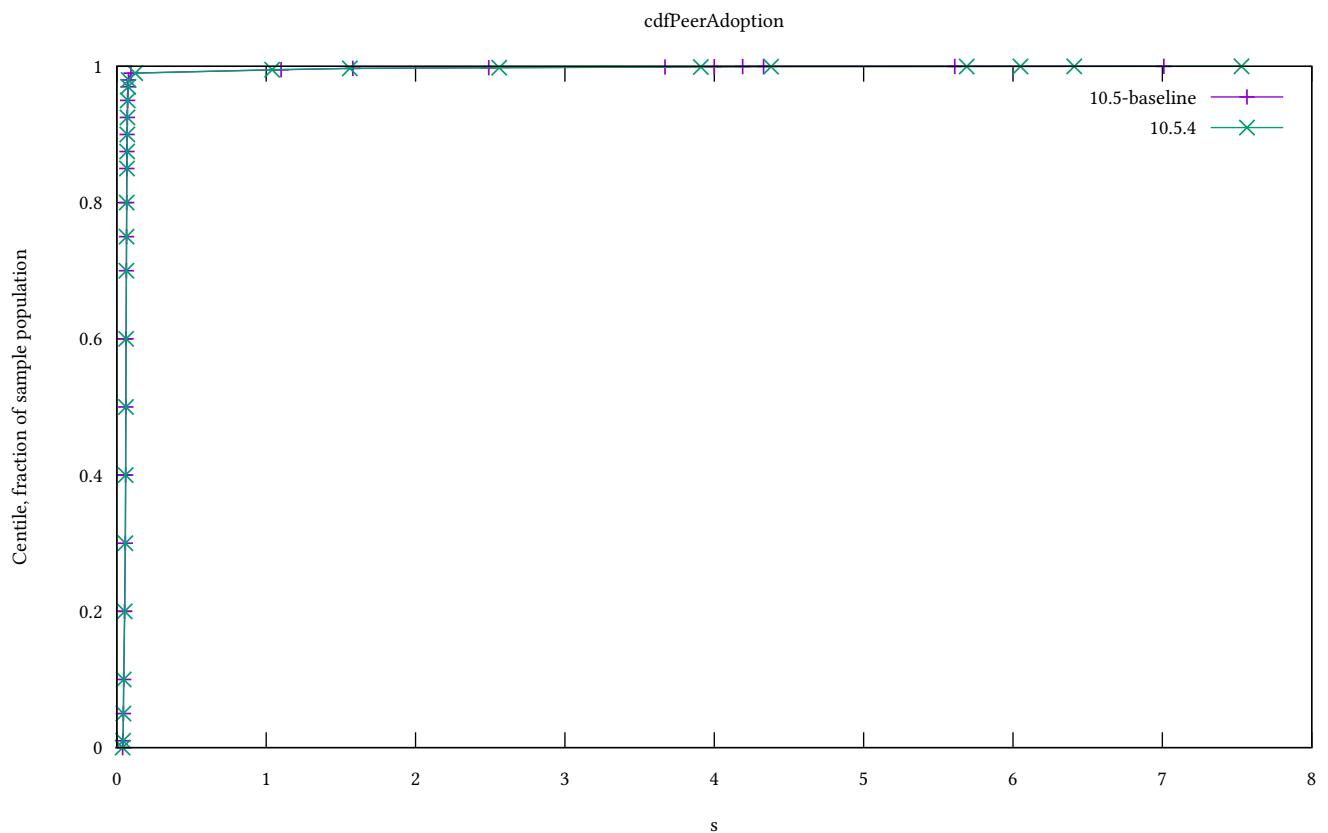


Figure 25: Fetched to adopted

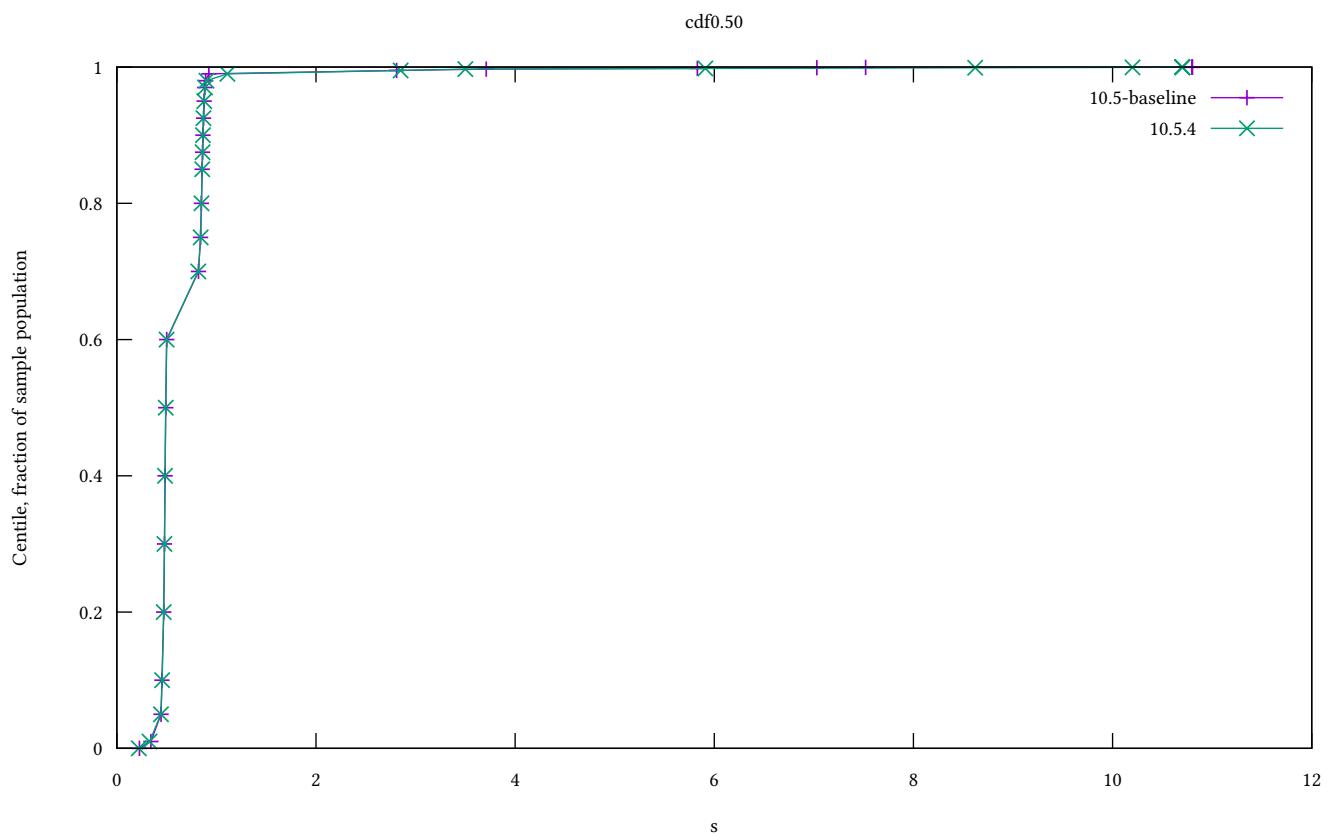


Figure 26: 0.50 adoption

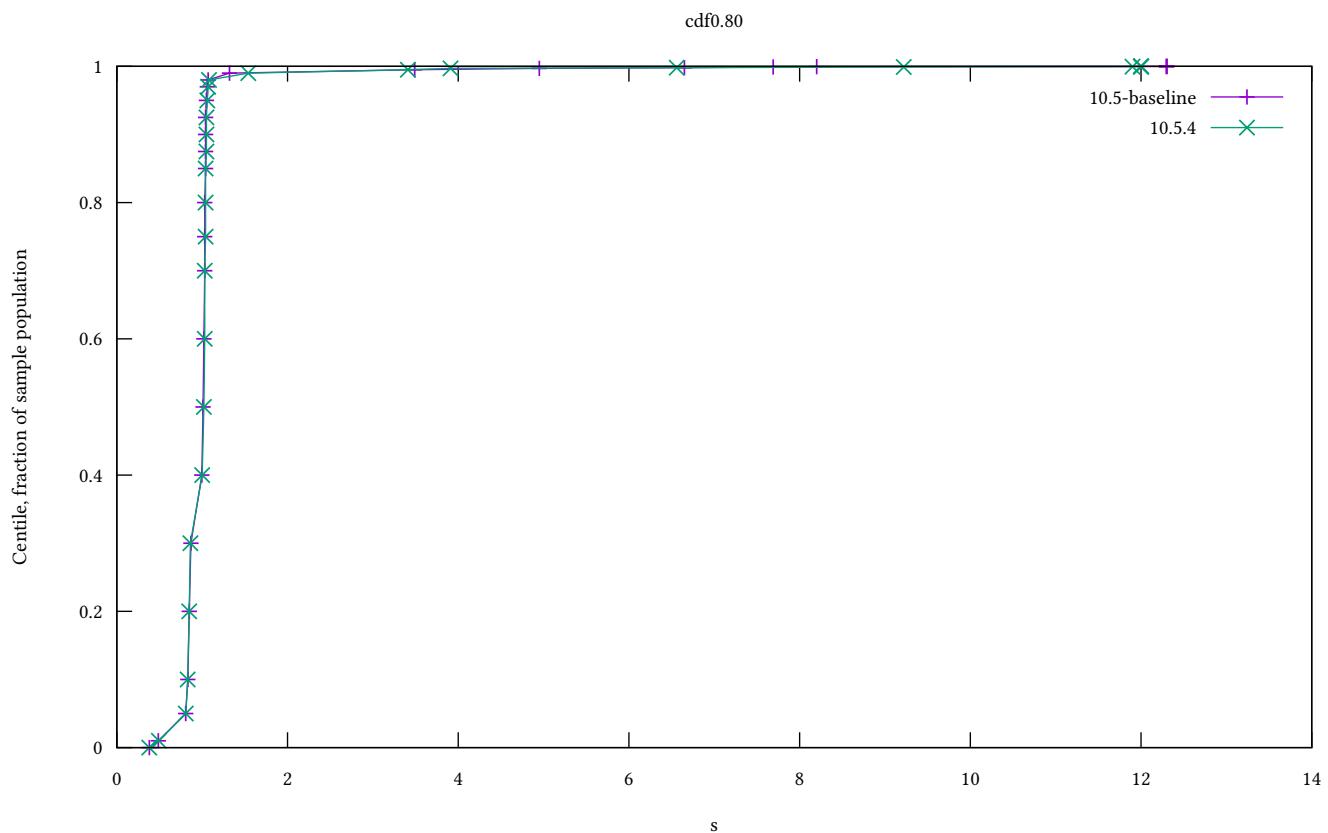


Figure 27: 0.80 adoption

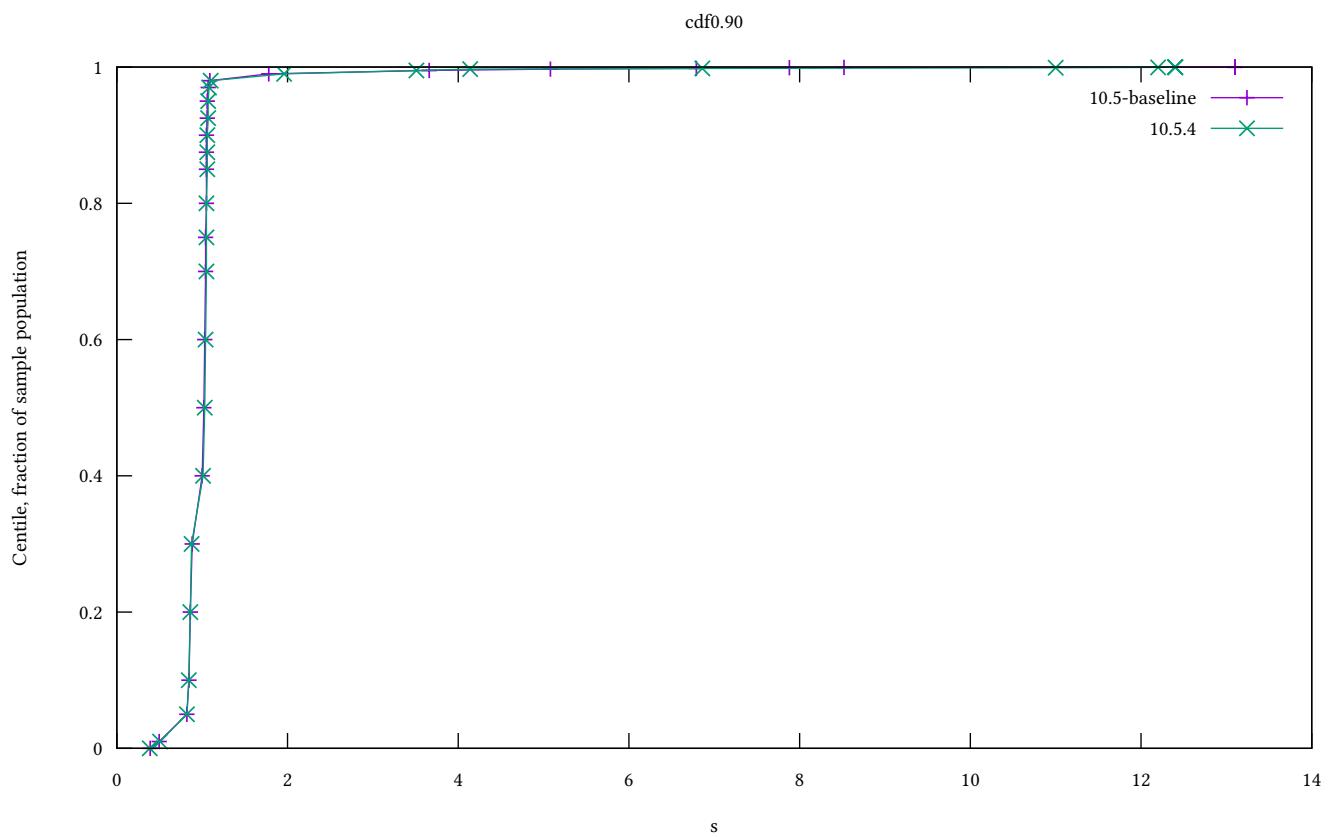


Figure 28: 0.90 adoption

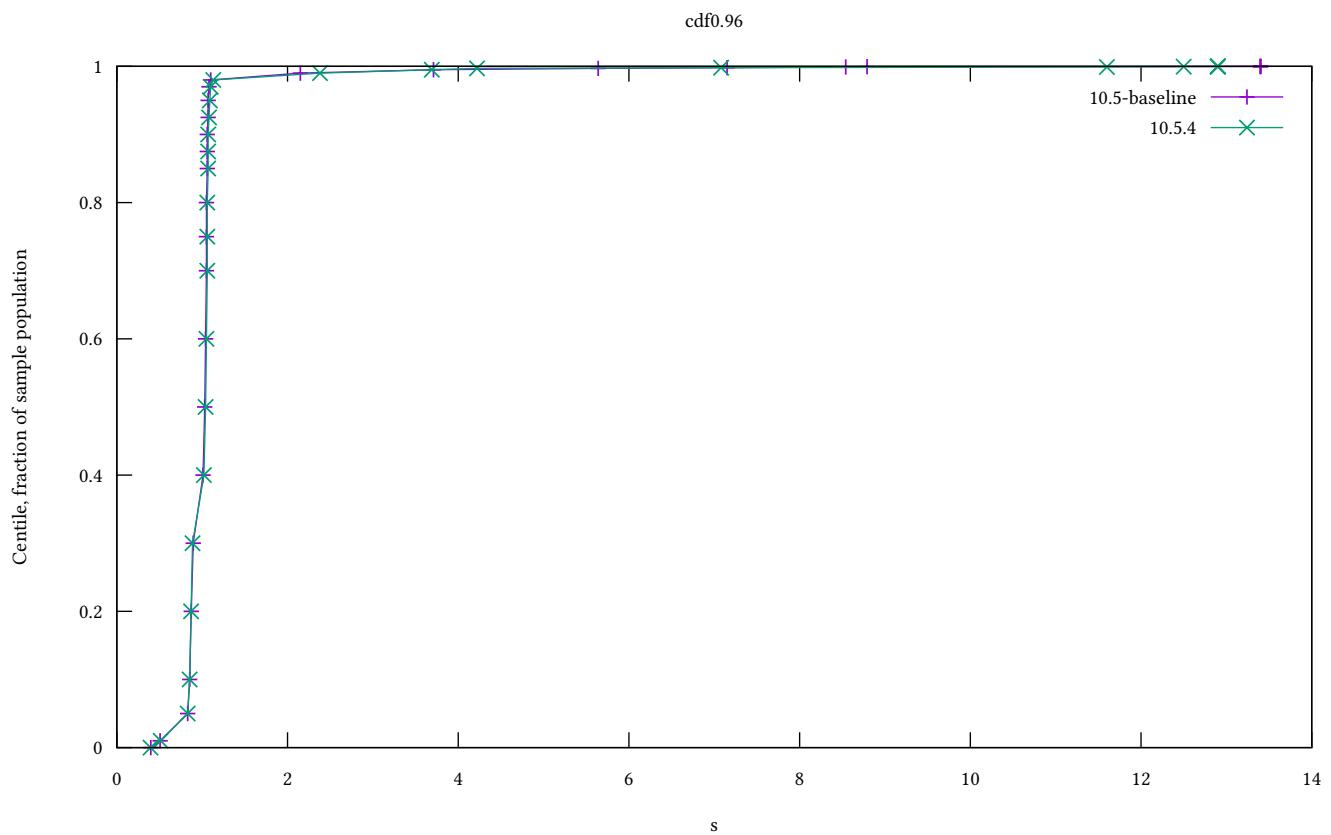


Figure 29: 0.96 adoption

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

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

Fetched 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

Fetched 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 dateset (*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