10.5.4 against 10.5-baseline

Plutus countdown loop workload

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

Contents

Manifest

We compare `10.5-baseline` (Conway) and `10.5.4` (Conway) relative to `10.5-baseline` (Conway), under Plutus countdown loop workload.

	10.5-baseline	10.5.4
Analysis date	2025-06-29	2026-02-07
Cluster system start date	2025-06-28	2026-02-06
Cluster system start time	19:41:44	17:25:54
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	1	1
Tx Outputs	1	1
TPS	0.85	0.85
Transaction count	61200	61200
Plutus script	Loop	Loop
Machines	52	52
Number of filters applied	4	4
Log objects emitted per host	909734.61538	915544.32692
Log objects analysed per host	611825.59615	613953.84615
Host run time, s	71830.2	71860.1
Host log line rate, Hz	12.665	12.740
Total log objects analysed	31814931	31925600
Run time, s	71835	71863
Analysed run duration, s	56010	56009
Run time efficiency	0.77	0.77
Node start spread, s	4.9408652	5.6183688
Node stop spread, s	4.7299718	2.1555507
Slots analysed	56007	56007
Blocks analysed	2710	2771
Blocks rejected	881	888

Analysis

Resource Usage

	10.5-baseline	10.5.4	Δ	Δ%
Forge loop starts, units	0.99889	0.99894	0.000	0
Process CPU usage, %	3.8865	4.0087	0.122	3
RTS GC CPU usage, %	0.235	0.2437	0.009	4
RTS Mutator CPU usage, %	3.6464	3.7587	0.112	3
Major GCs, events	0.0008	0.0008	−0.000	0
Minor GCs, events	0.9828	0.9875	0.005	0.5
Kernel RSS, MiB	8198.	7962.4	−235.542	−3
RTS heap size, MiB	8136.7	7901.1	−235.613	−3
RTS live GC dateset, MiB	4074.5	3381.6	−692.972	−17
RTS alloc rate, MiB/s	28.726	28.847	0.120	0.4
Filesystem reads, KiB/s	0.0001	0.0	−0.000	−100
Filesystem writes, KiB/s	227.39	225.1	−2.289	−1
CPU 85% spans, slots	10.397	10.644	0.246	2

Anomaly control

	10.5-baseline	10.5.4	Δ	Δ%
Blocks per host, blocks	70.981	72.288	1.308	2
Filtered to chained blocks, :	0.7559	0.7574	0.001	0.2
Chained to forged blocks, :	0.9727	0.973	0.000	0.0
Height & slot battles, blocks	0.00074	0.00072	−0.000	0
Block size, Bytes	2996.1	2996.1	−0.007	−0.0

Forging

	10.5-baseline	10.5.4	Δ	Δ%
Started forge loop iteration, s	0.00139	0.00106	−0.000	−23
Acquired block context, s	6.6e-5	6.9e-5	0.000	0
Acquired ledger state, s	9.9e-5	9.5e-5	−0.000	0
Acquired ledger view, s	3.0e-5	3.0e-5	0.000	0
Leadership check duration, s	0.00042	0.00043	0.000	0
Ledger ticking, s	0.02455	0.02343	−0.001	−5
Mempool snapshotting, s	0.00155	0.0015	−0.000	0
Leadership to forged, s	0.00017	0.00017	0.000	0
Forged to announced, s	0.00066	0.00067	0.000	0
Forged to sending, s	0.00539	0.00643	0.001	19
Forged to self-adopted, s	0.04785	0.04724	−0.001	−1
Slot start to announced, s	0.02893	0.02745	−0.001	−5

Individual peer propagation

	10.5-baseline	10.5.4	Δ	Δ%
First peer notice, s	0.03085	0.02932	−0.002	−5
First peer fetch, s	0.03608	0.03547	−0.001	−2
Notice to fetch request, s	0.00121	0.0012	−0.000	0
Fetch duration, s	0.12217	0.12669	0.005	4
Fetched to announced, s	0.00083	0.0008	−0.000	0
Fetched to sending, s	0.04335	0.04298	−0.000	−0.9
Fetched to adopted, s	0.04738	0.04481	−0.003	−5

End-to-end propagation

	10.5-baseline	10.5.4	Δ	Δ%
0.50 adoption, s	0.28996	0.28804	−0.002	−0.7
0.80 adoption, s	0.45116	0.46417	0.013	3
0.90 adoption, s	0.46431	0.47493	0.011	2
0.92 adoption, s	0.46727	0.47799	0.011	2
0.94 adoption, s	0.46997	0.48168	0.012	2
0.96 adoption, s	0.4748	0.48518	0.010	2
0.98 adoption, s	0.48303	0.49098	0.008	2
1.00 adoption, s	0.50602	0.50126	−0.005	−0.9

Appendix A: charts

Cluster performance charts

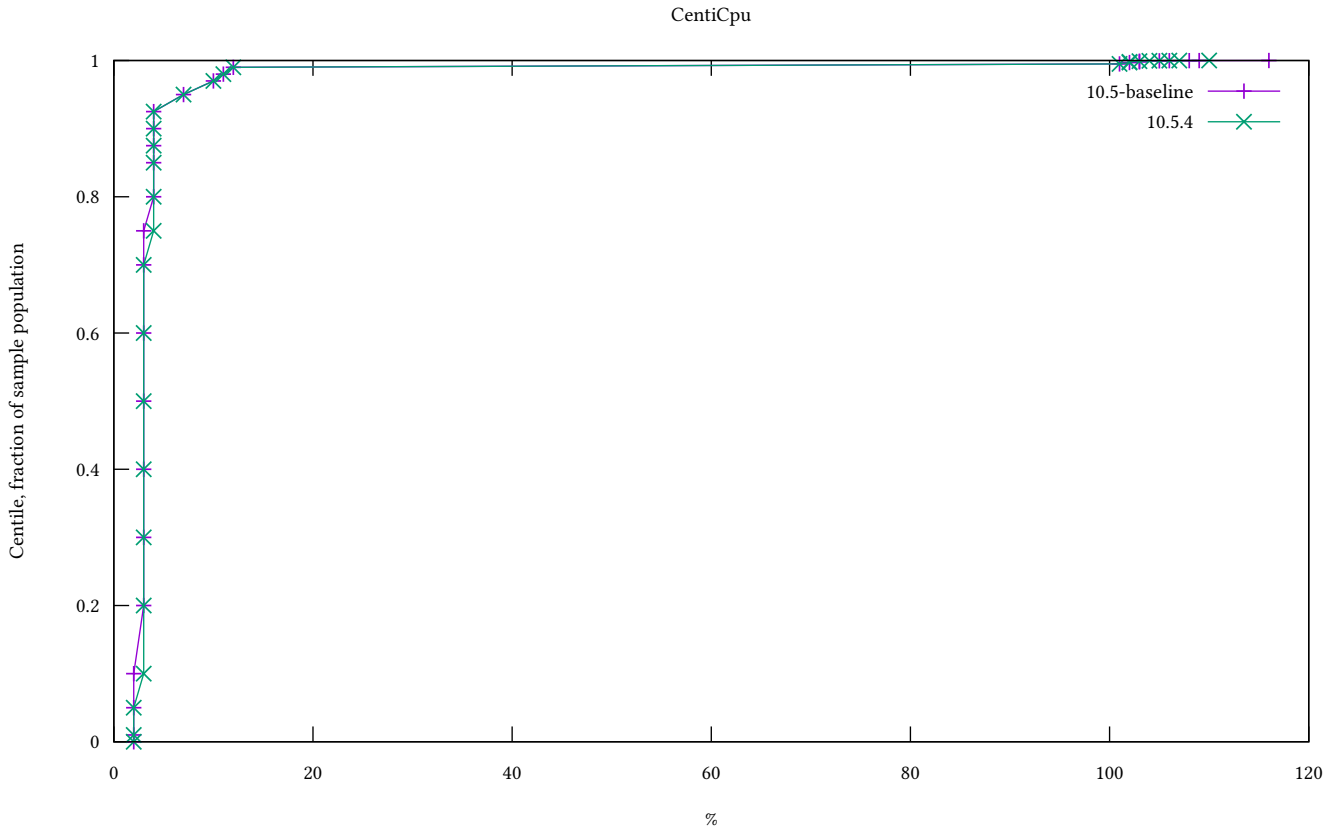

Figure 1: Process CPU usage

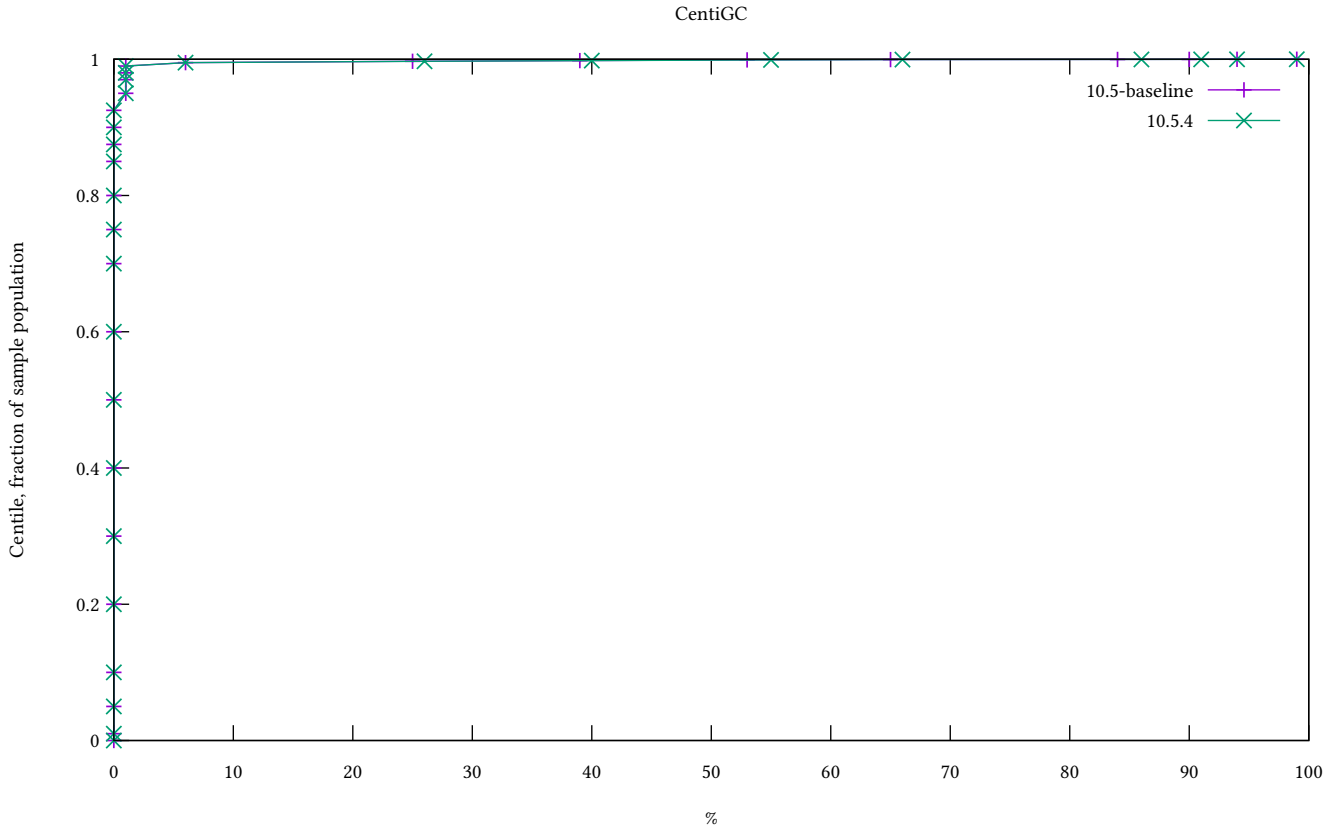

Figure 2: RTS GC CPU usage

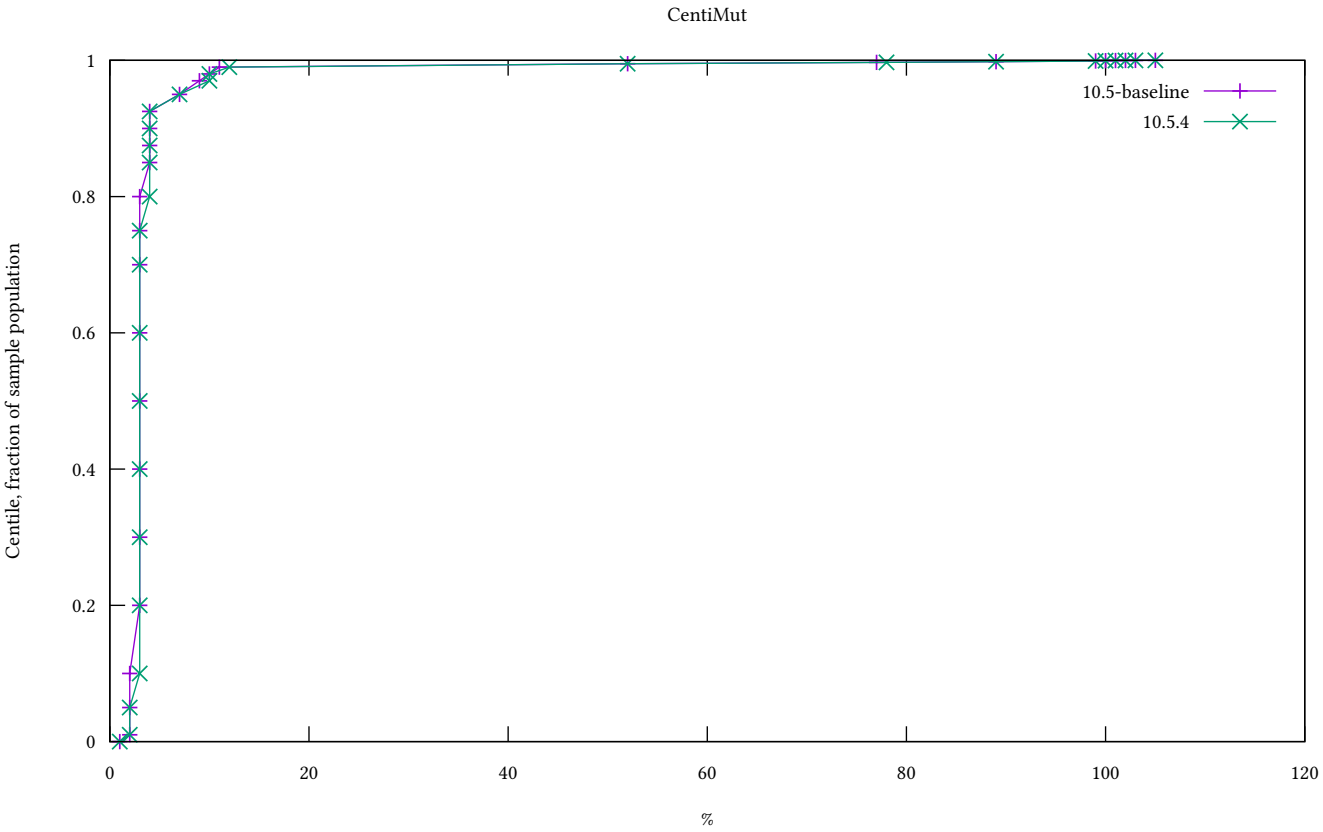

Figure 3: RTS Mutator CPU usage

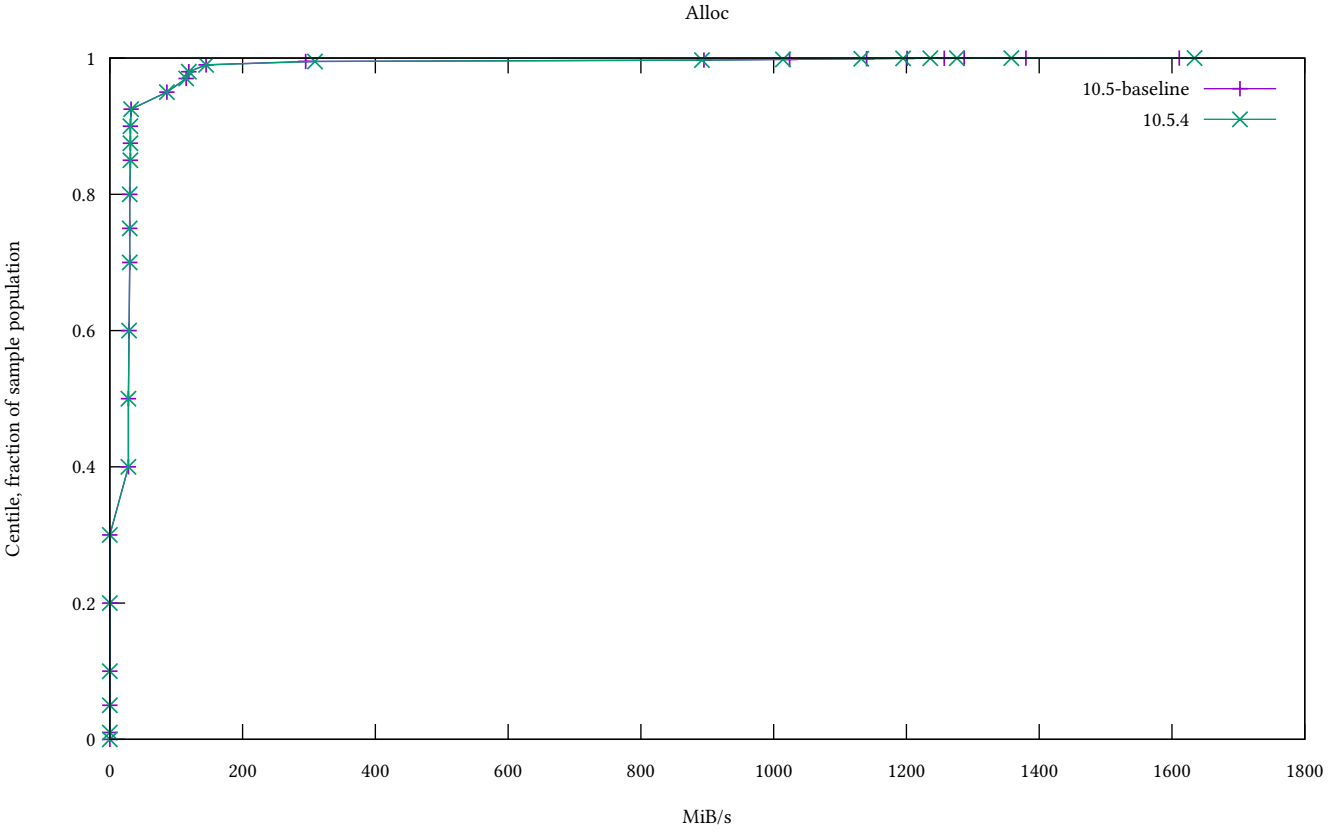

Figure 4: RTS alloc rate

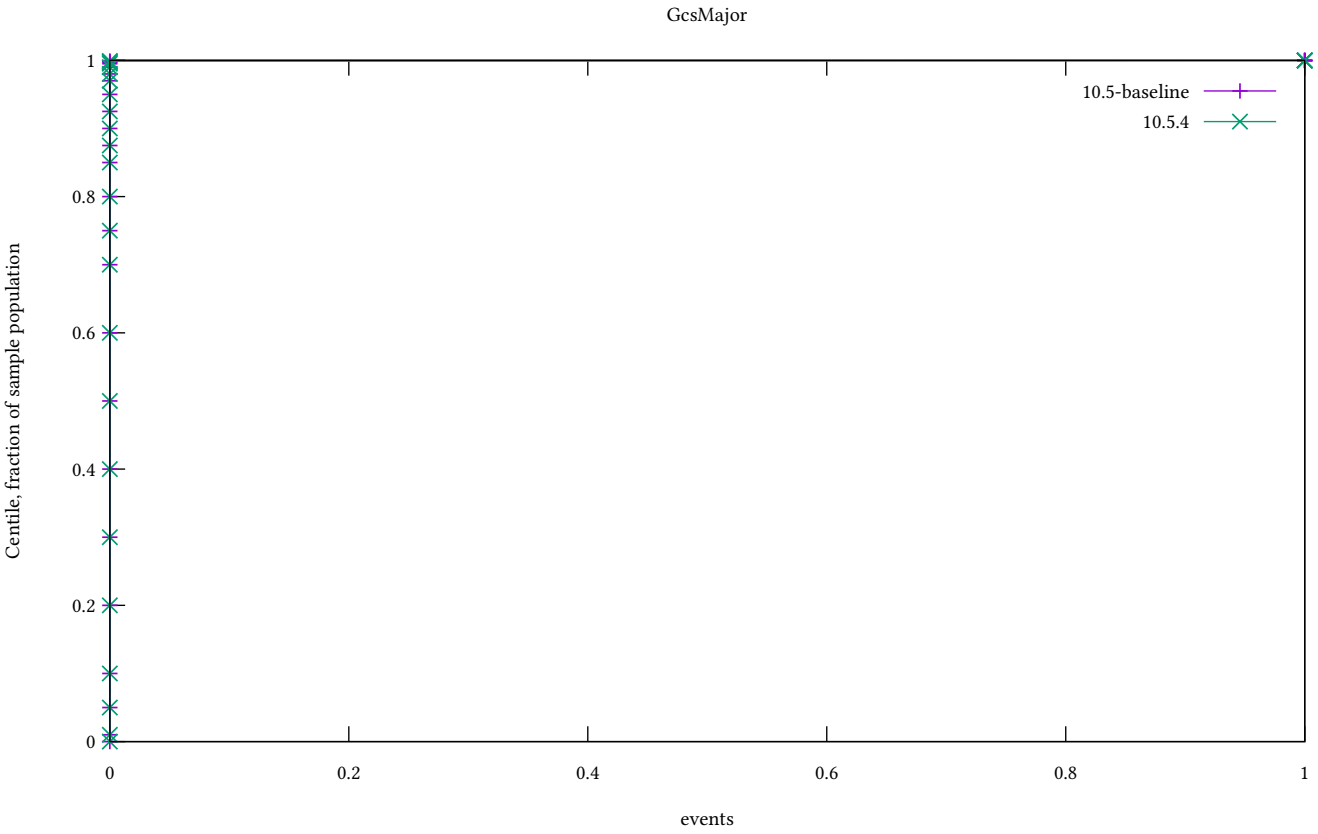

Figure 5: Major GCs

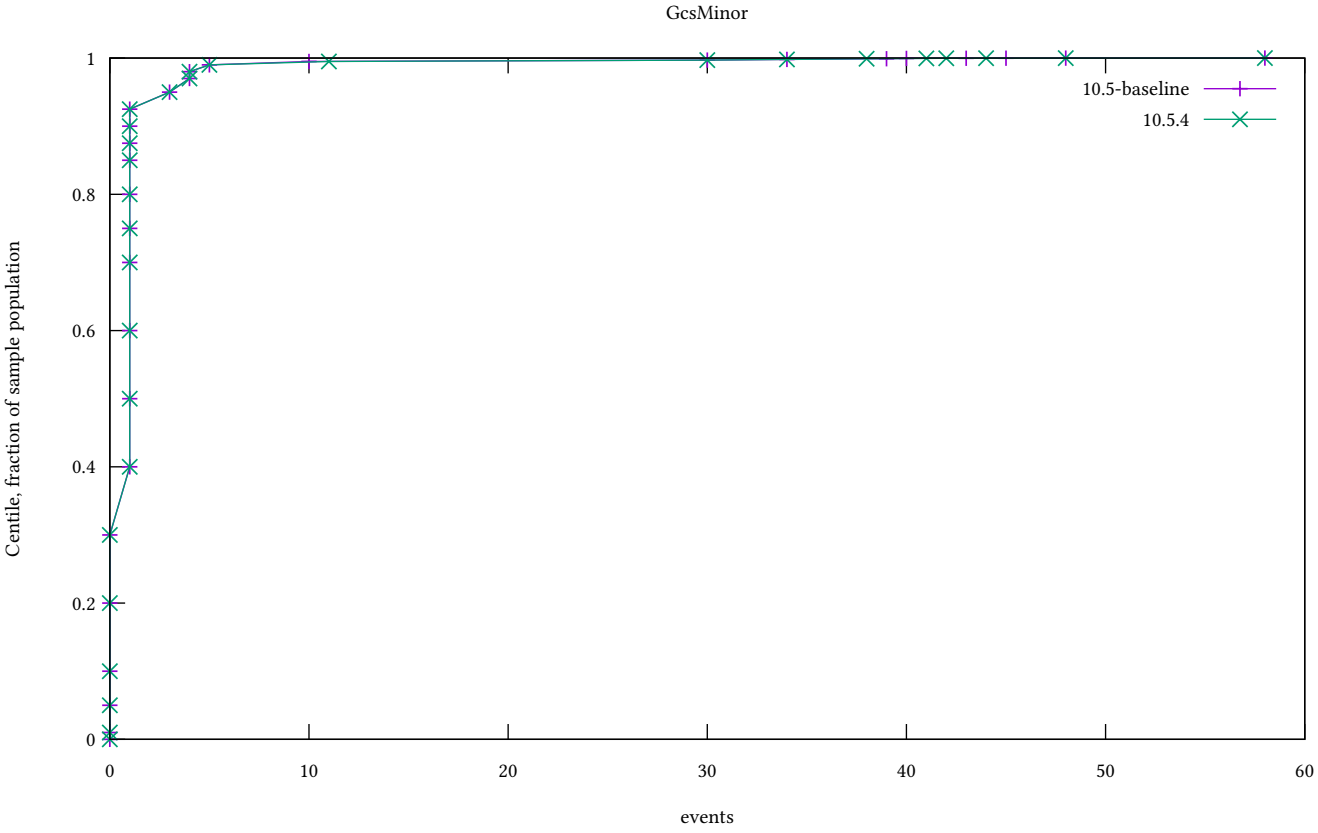

Figure 6: Minor GCs

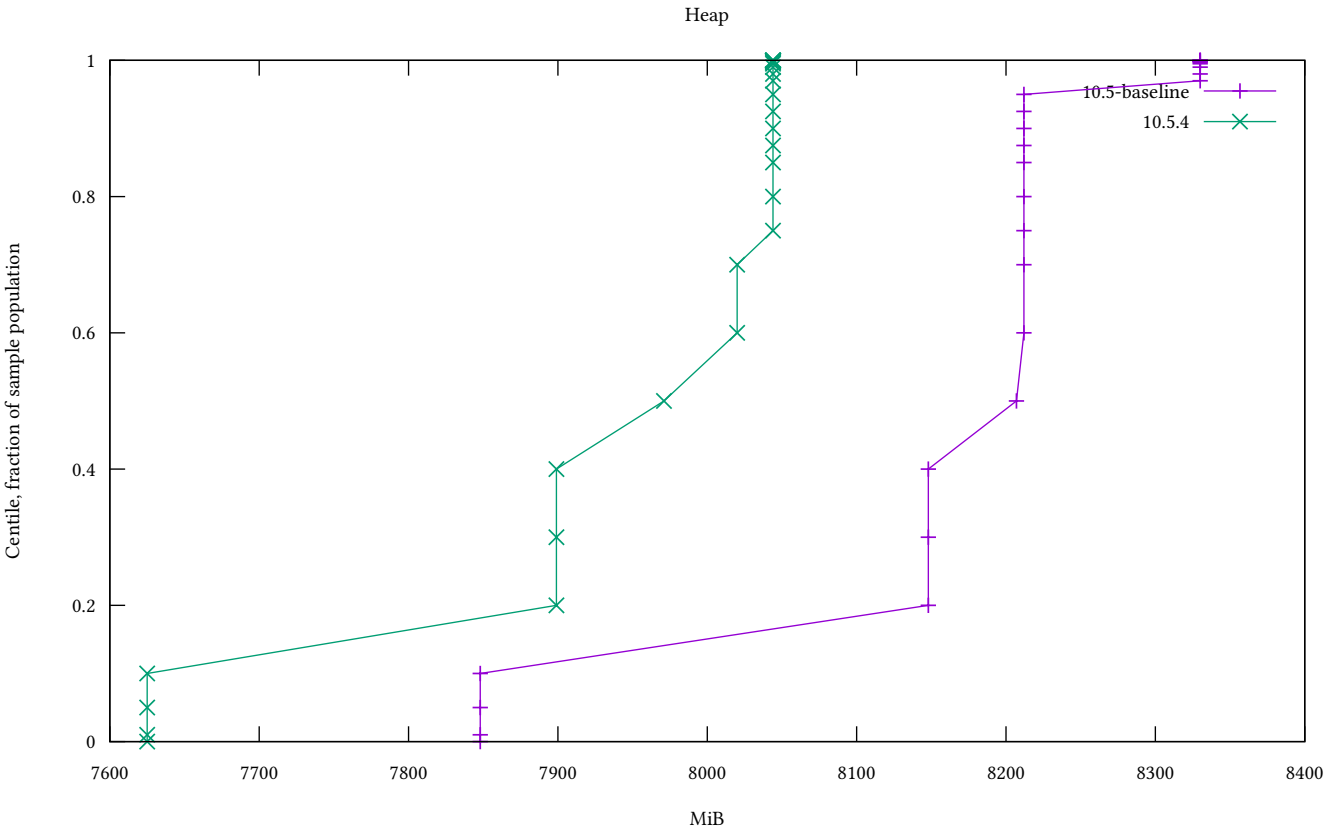

Figure 7: RTS heap size

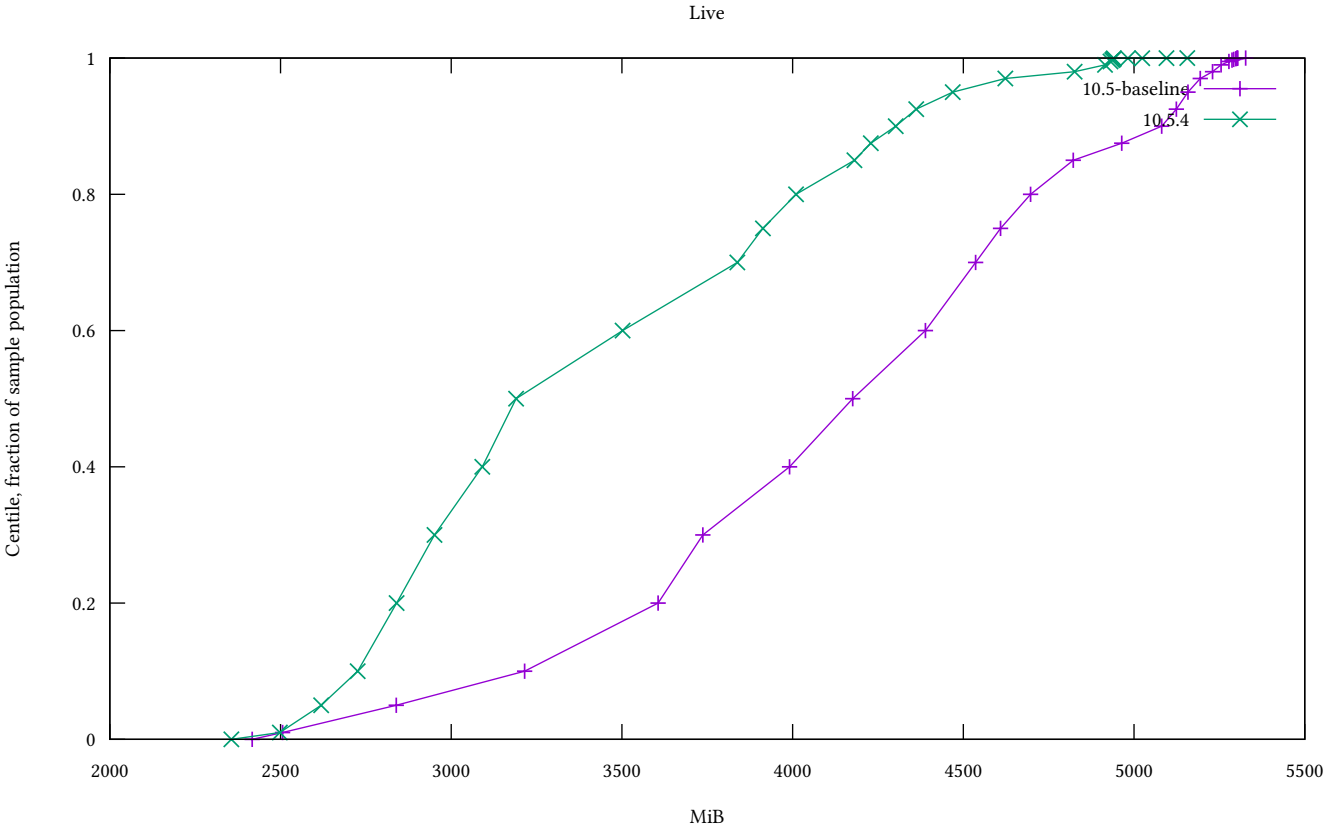

Figure 8: RTS live GC dateset

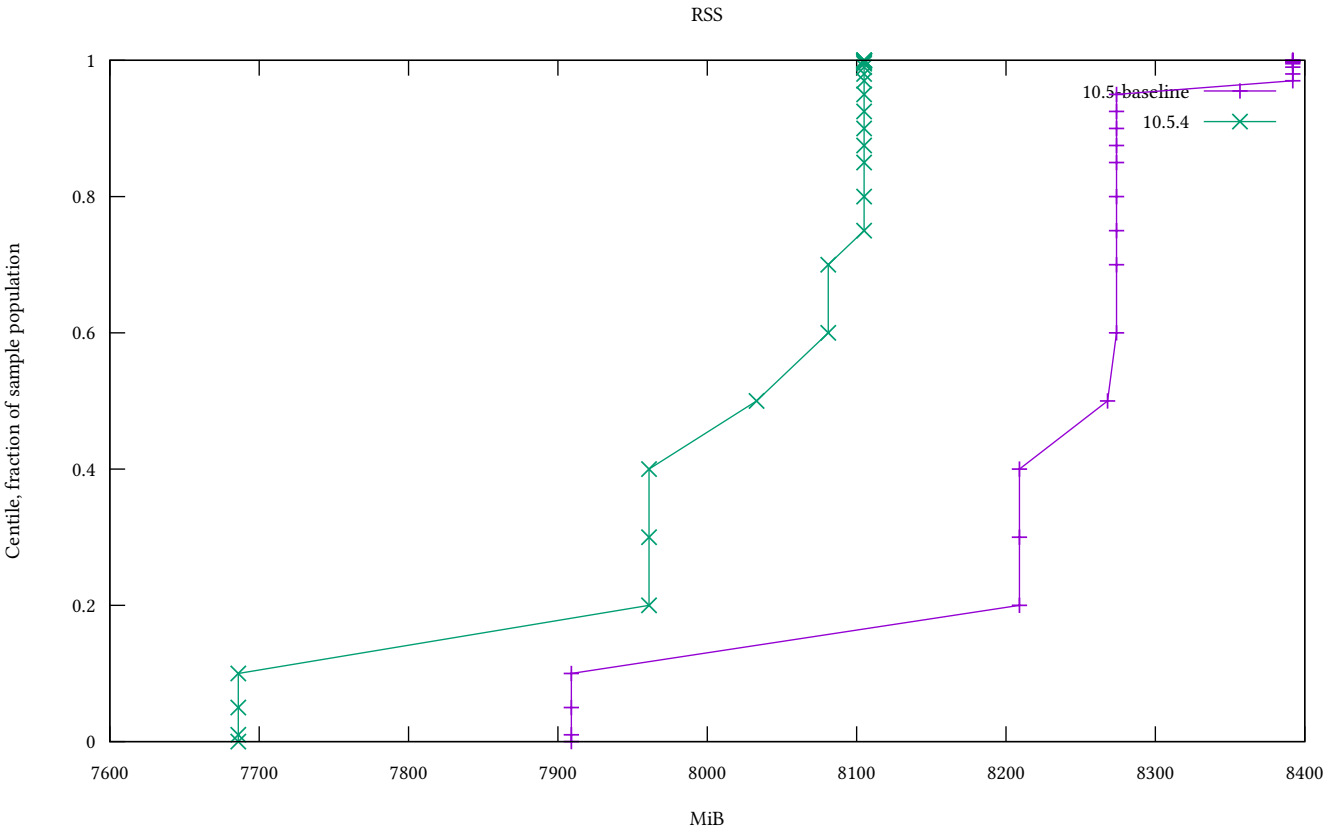

Figure 9: Kernel RSS

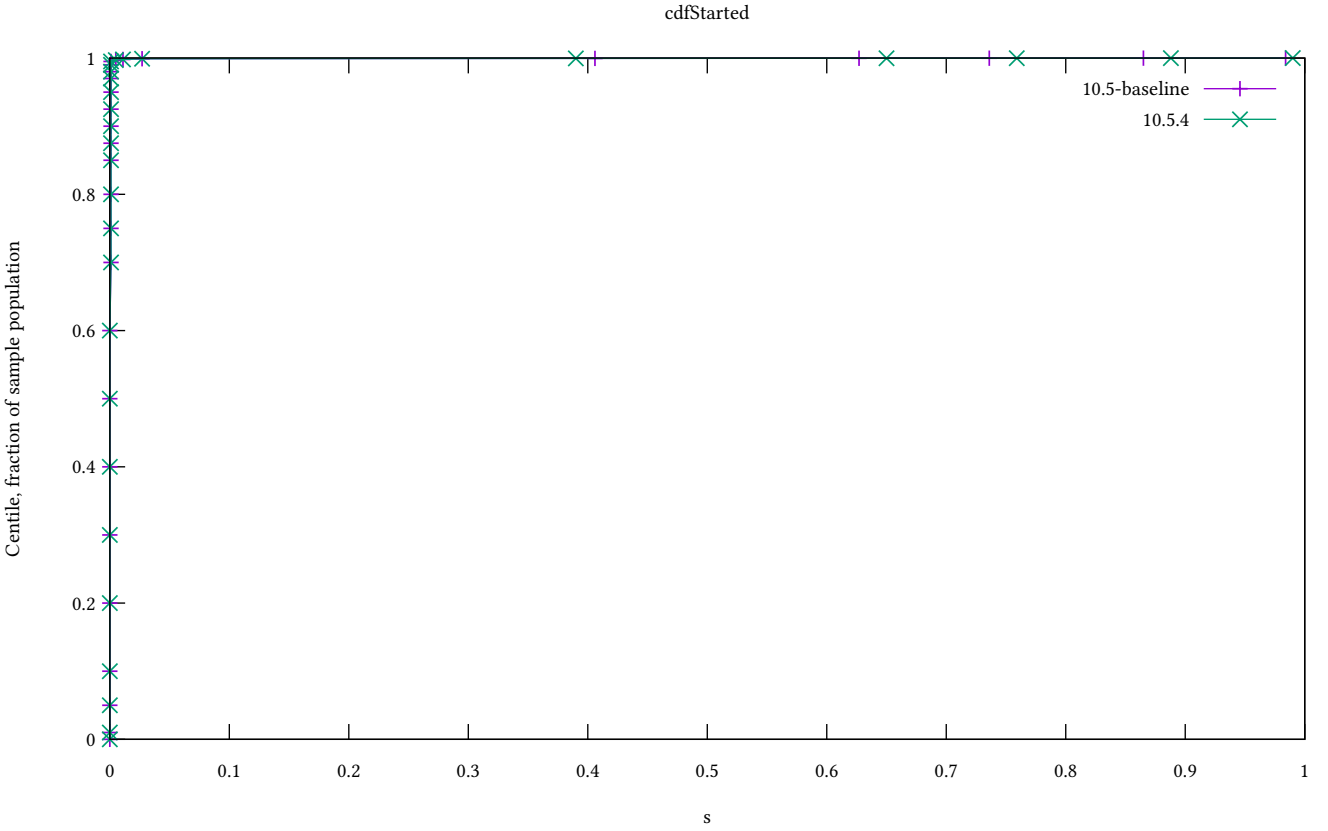

Figure 10: Forge loop tardiness

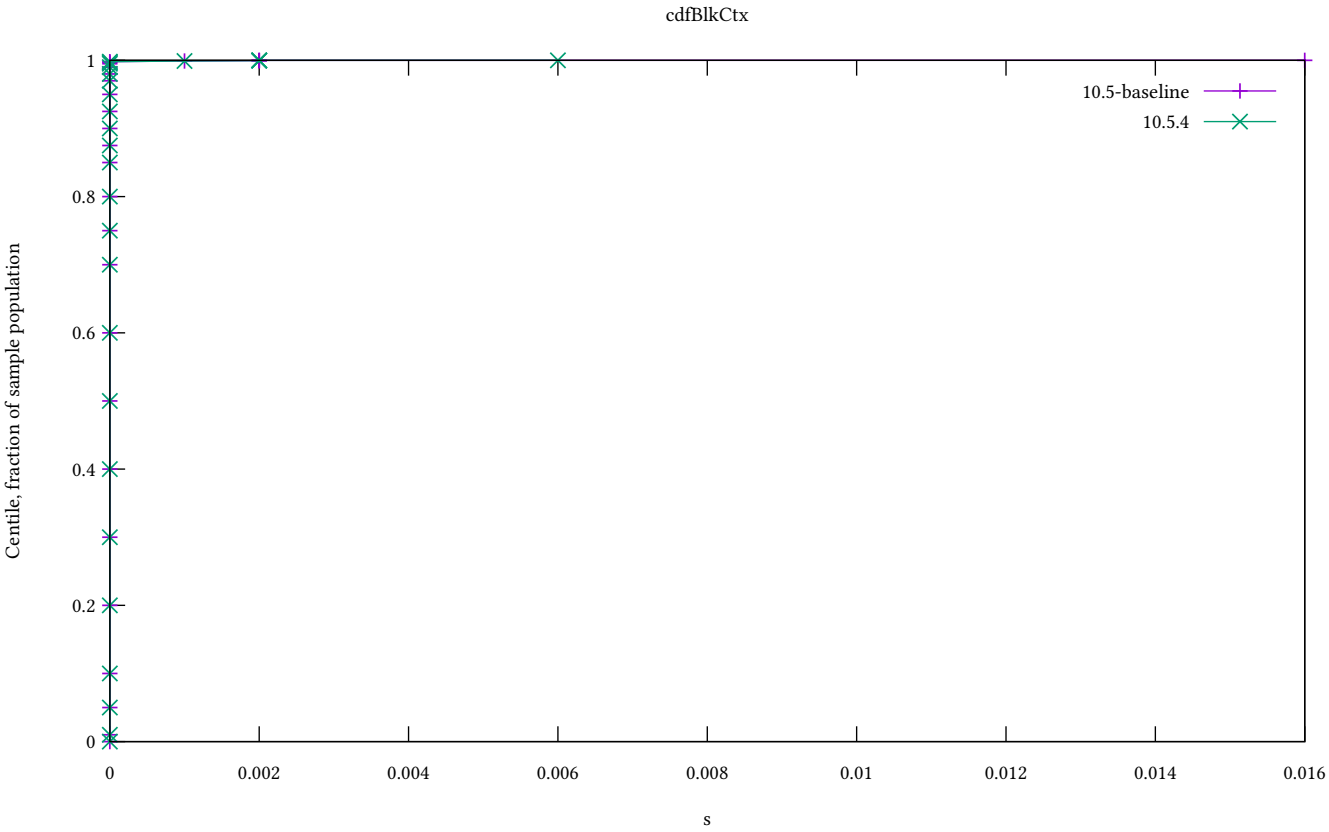

Figure 11: Block context acquisition delay

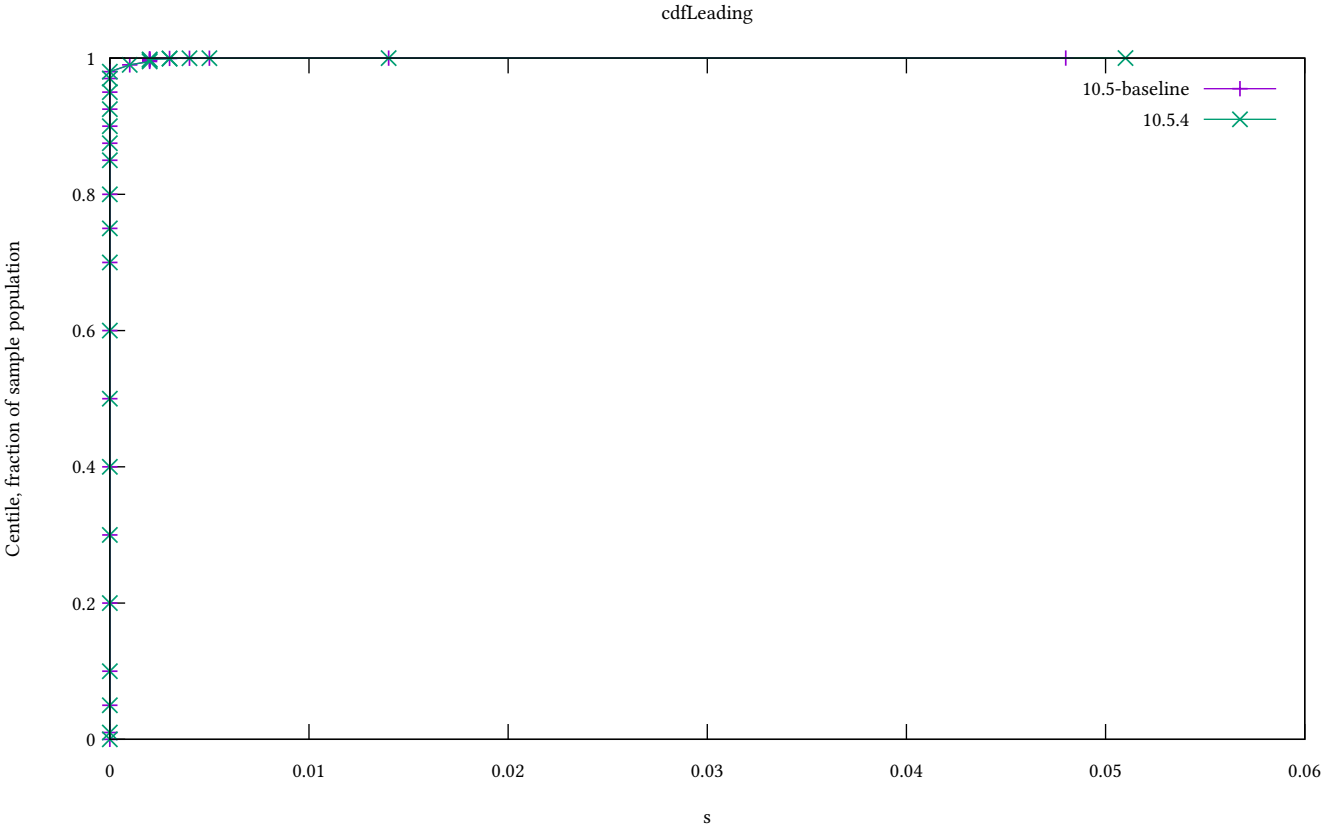

Figure 12: Leadership check duration

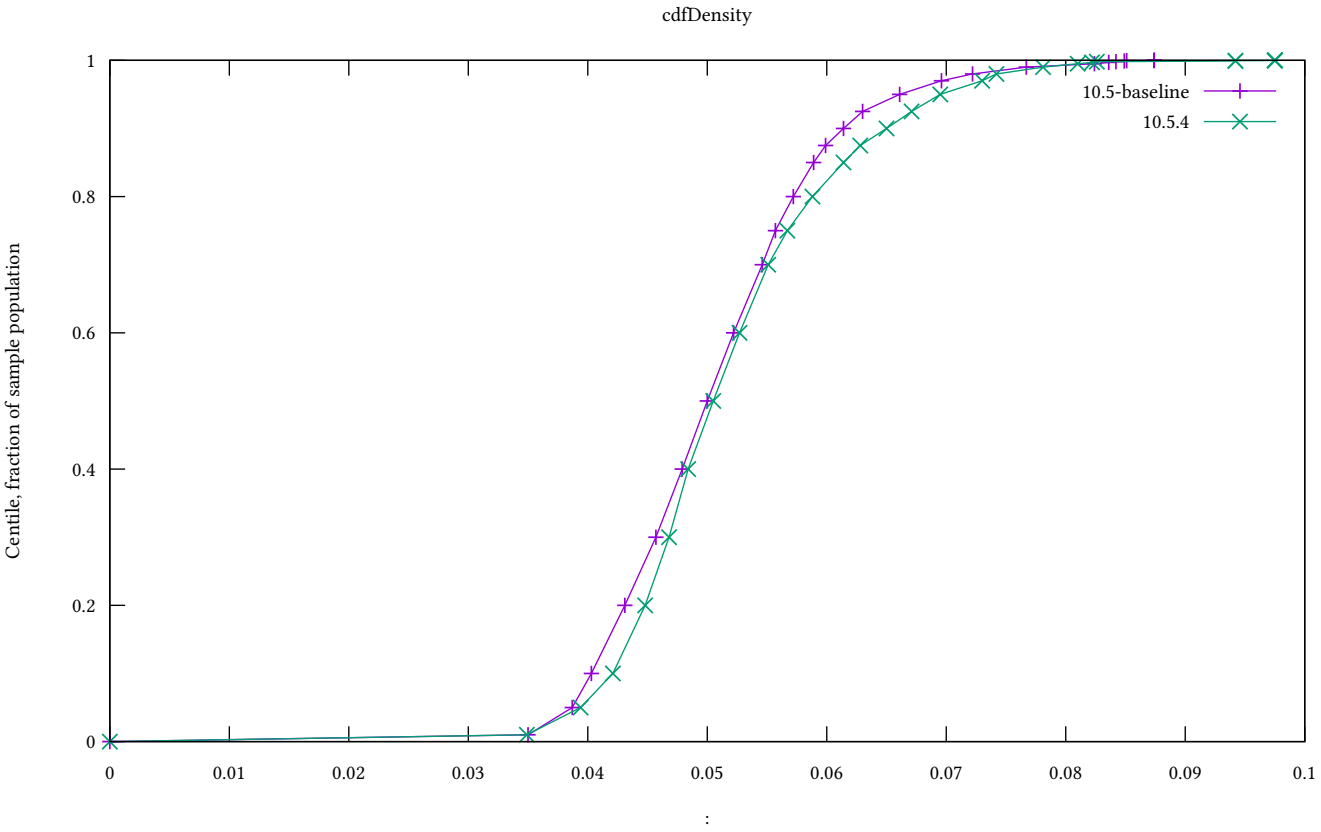

Figure 13: Chain density

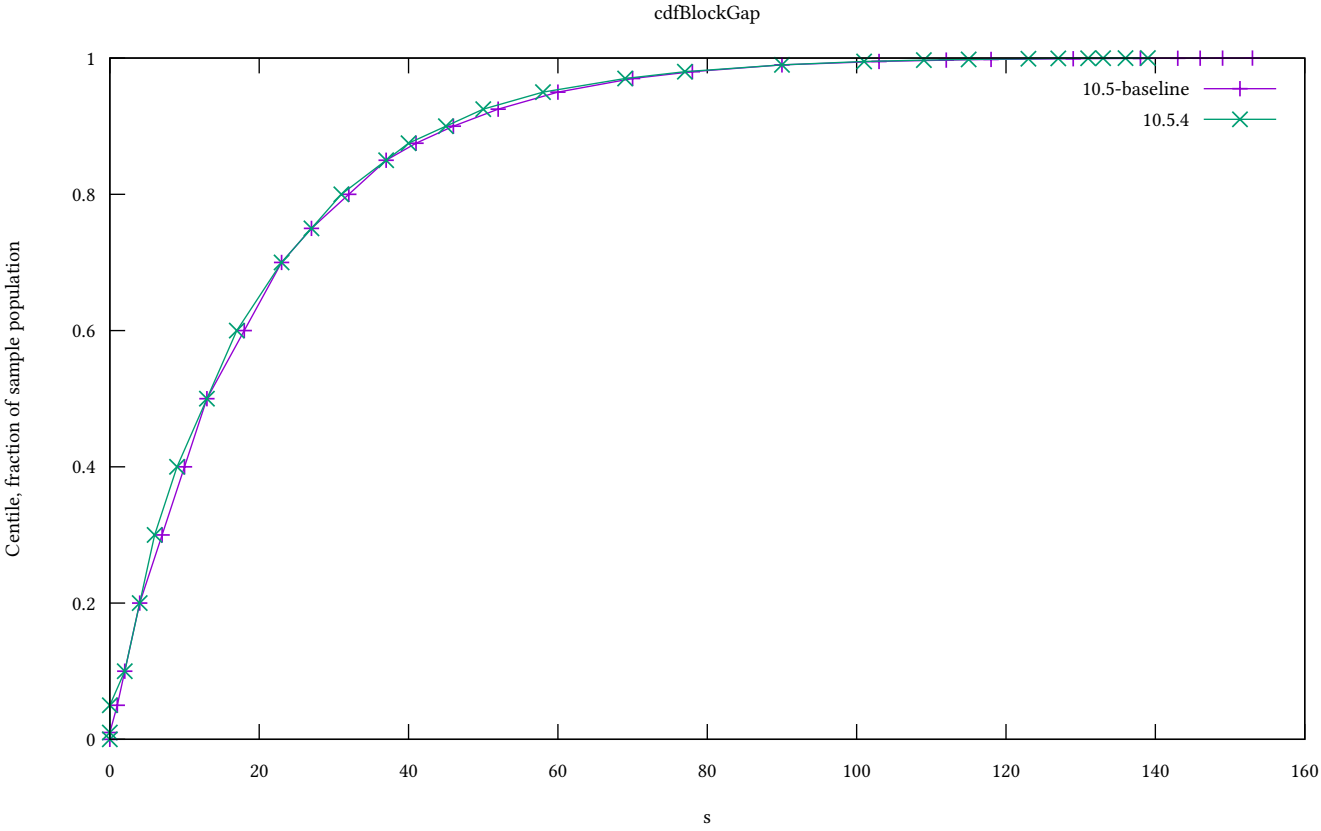

Figure 14: Interblock gap

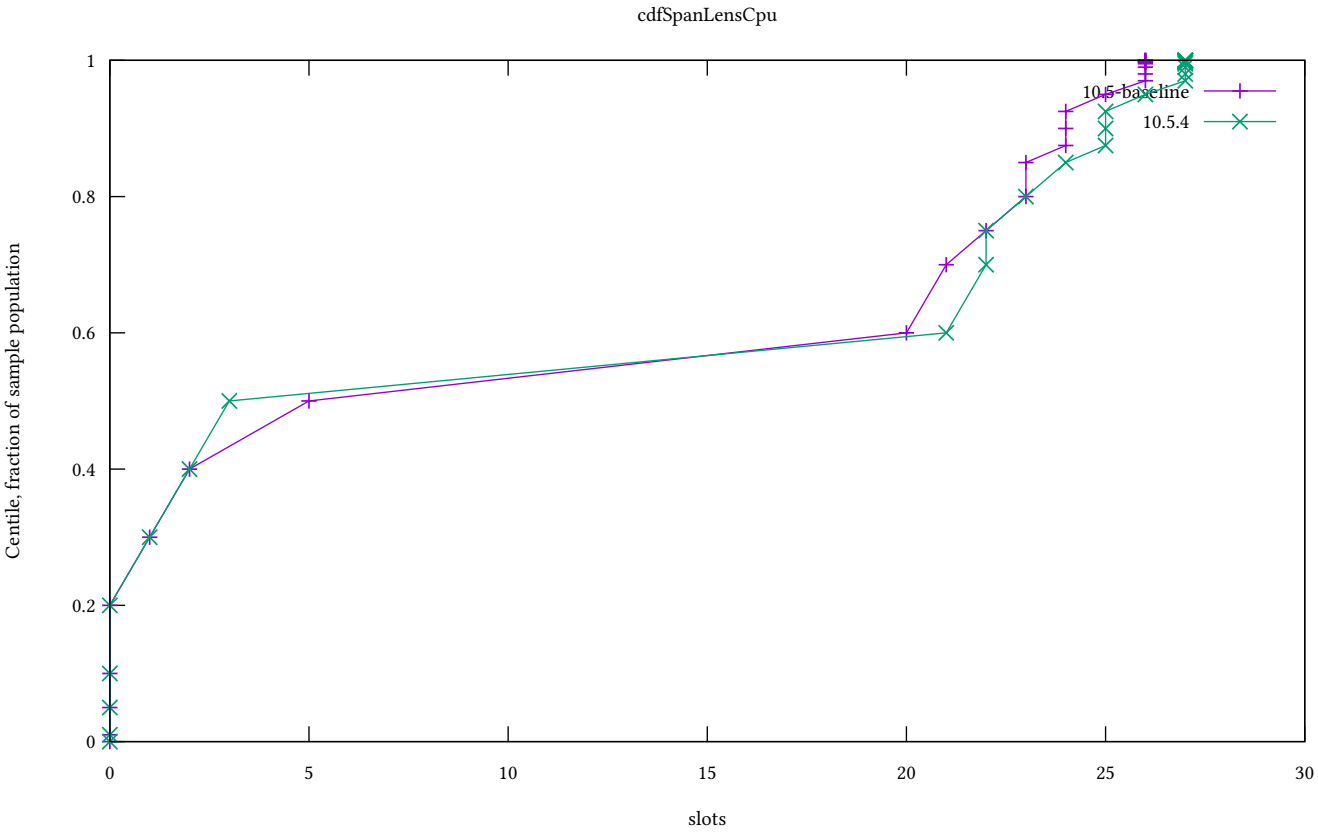

Figure 15: CPU 85% spans

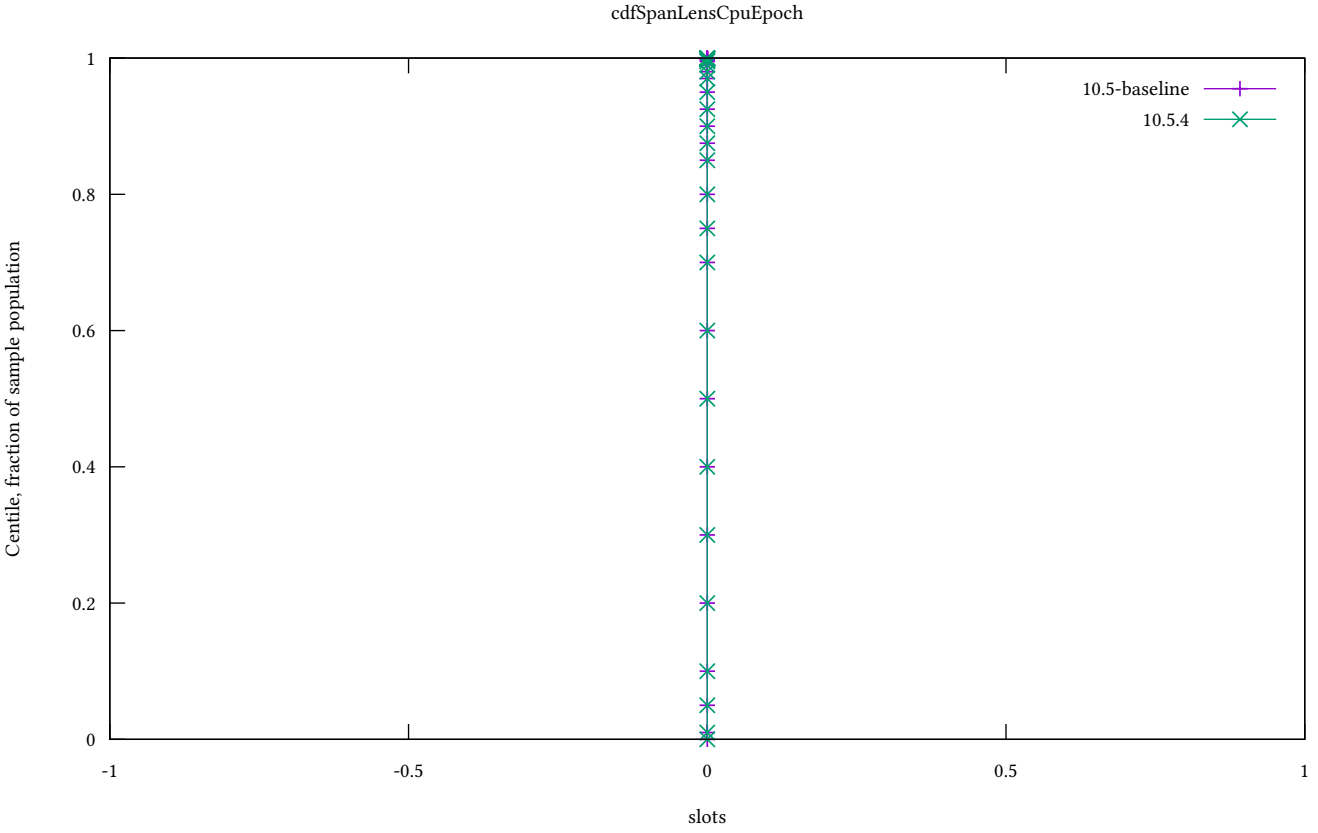

Figure 16: CPU spans at Ep boundary

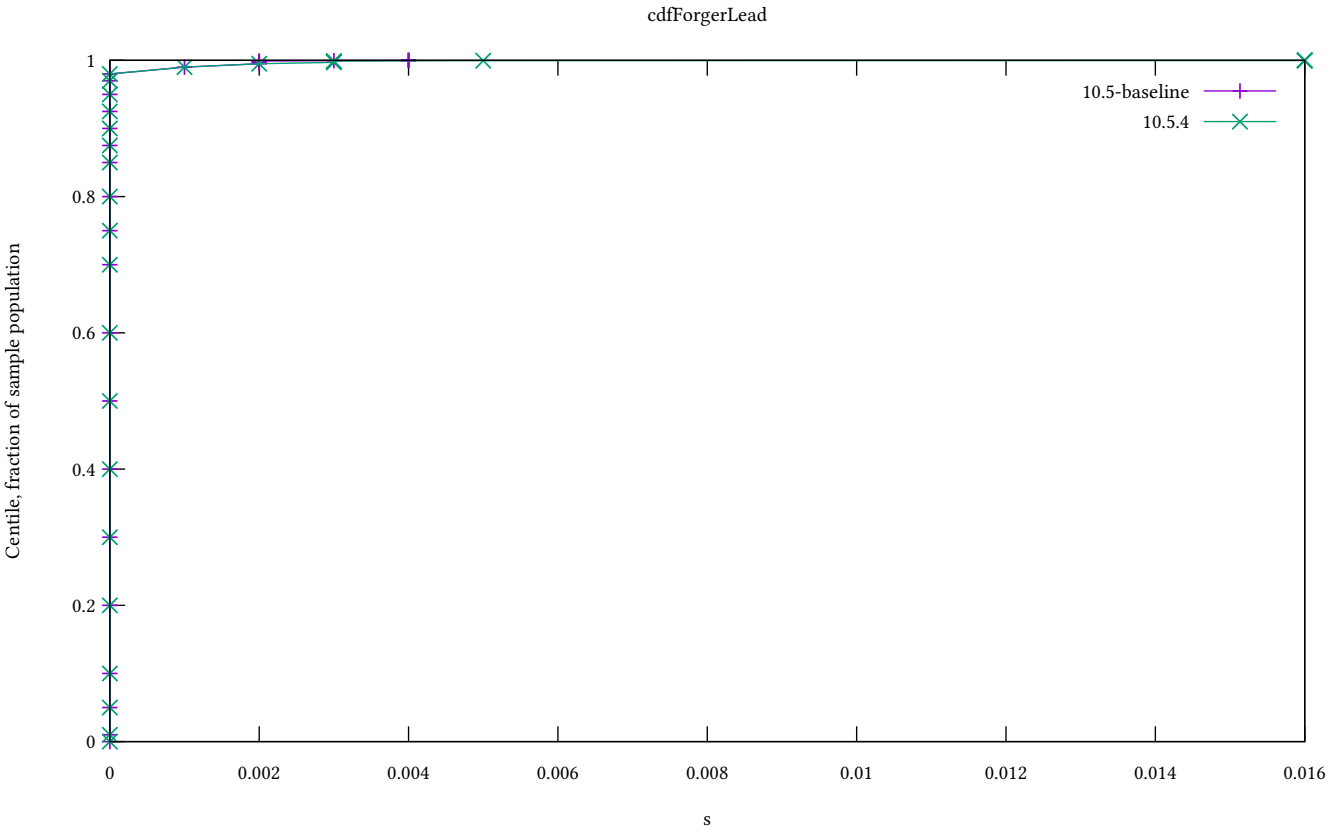

Figure 17: Leadership check duration

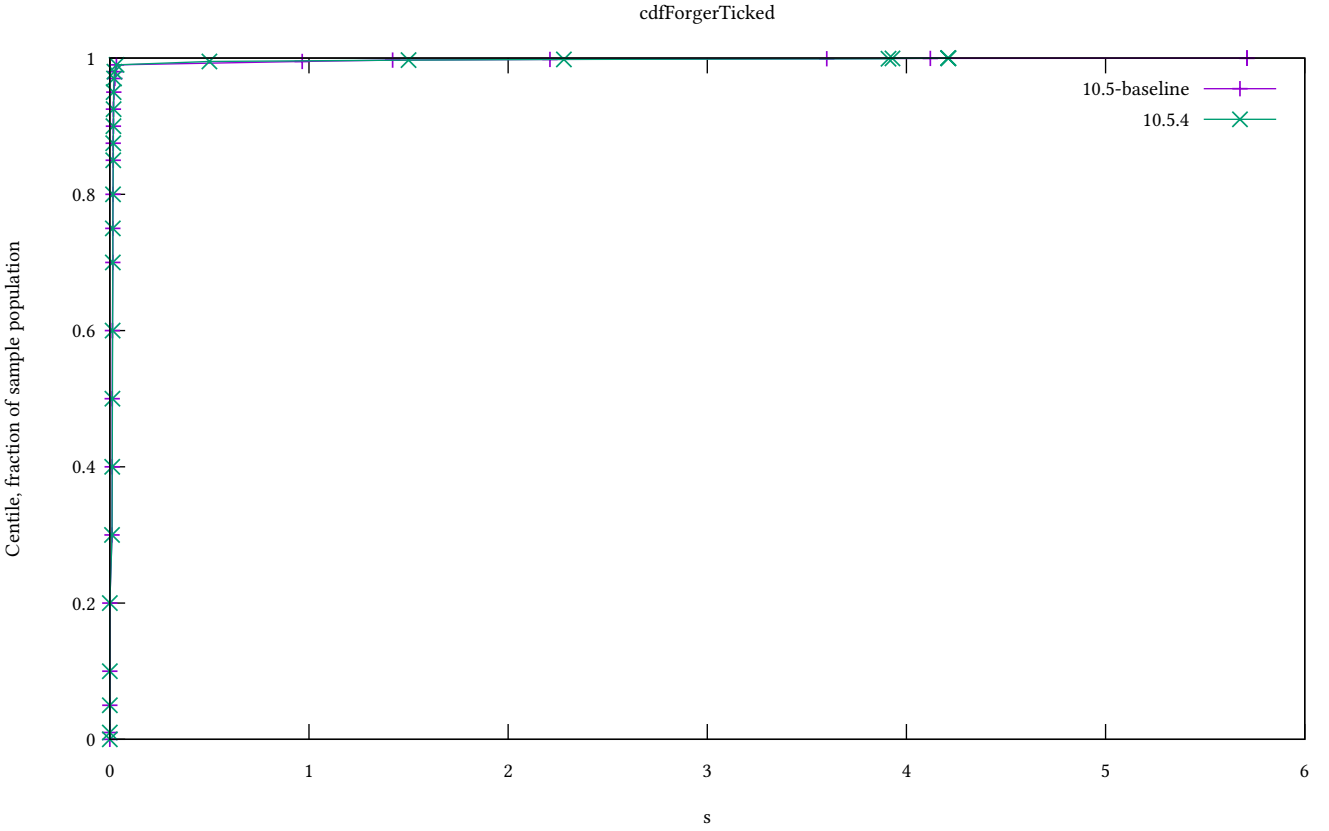

Figure 18: Ledger ticking

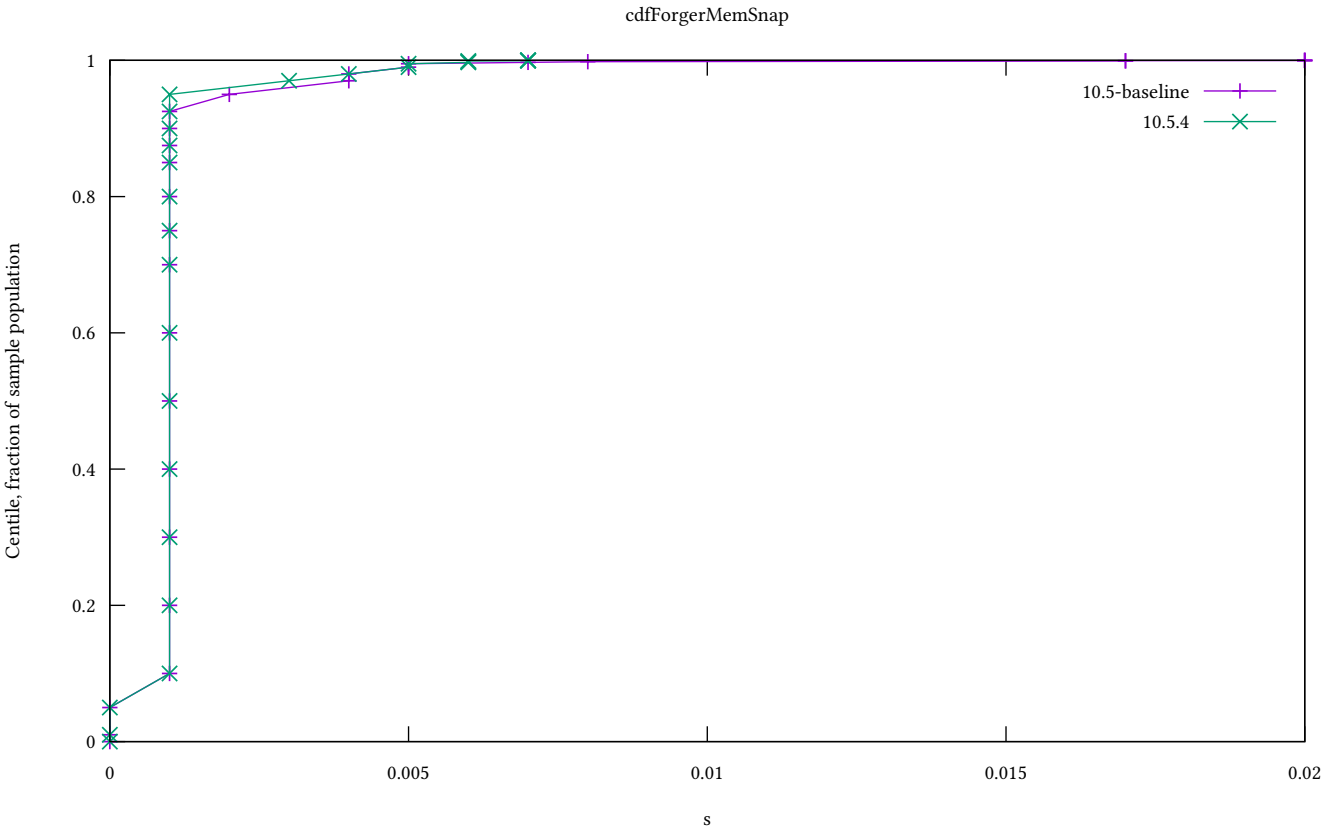

Figure 19: Mempool snapshotting

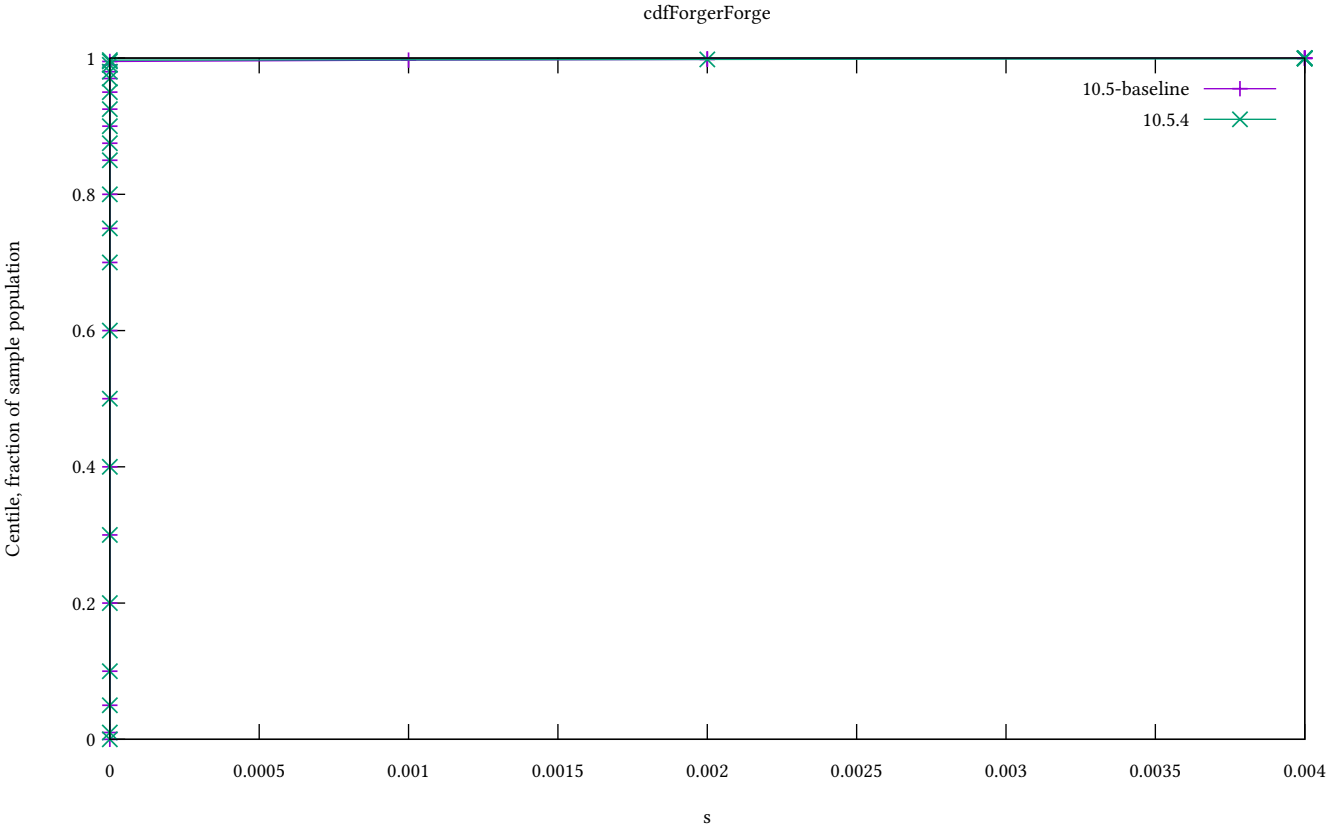

Figure 20: Leadership to forged

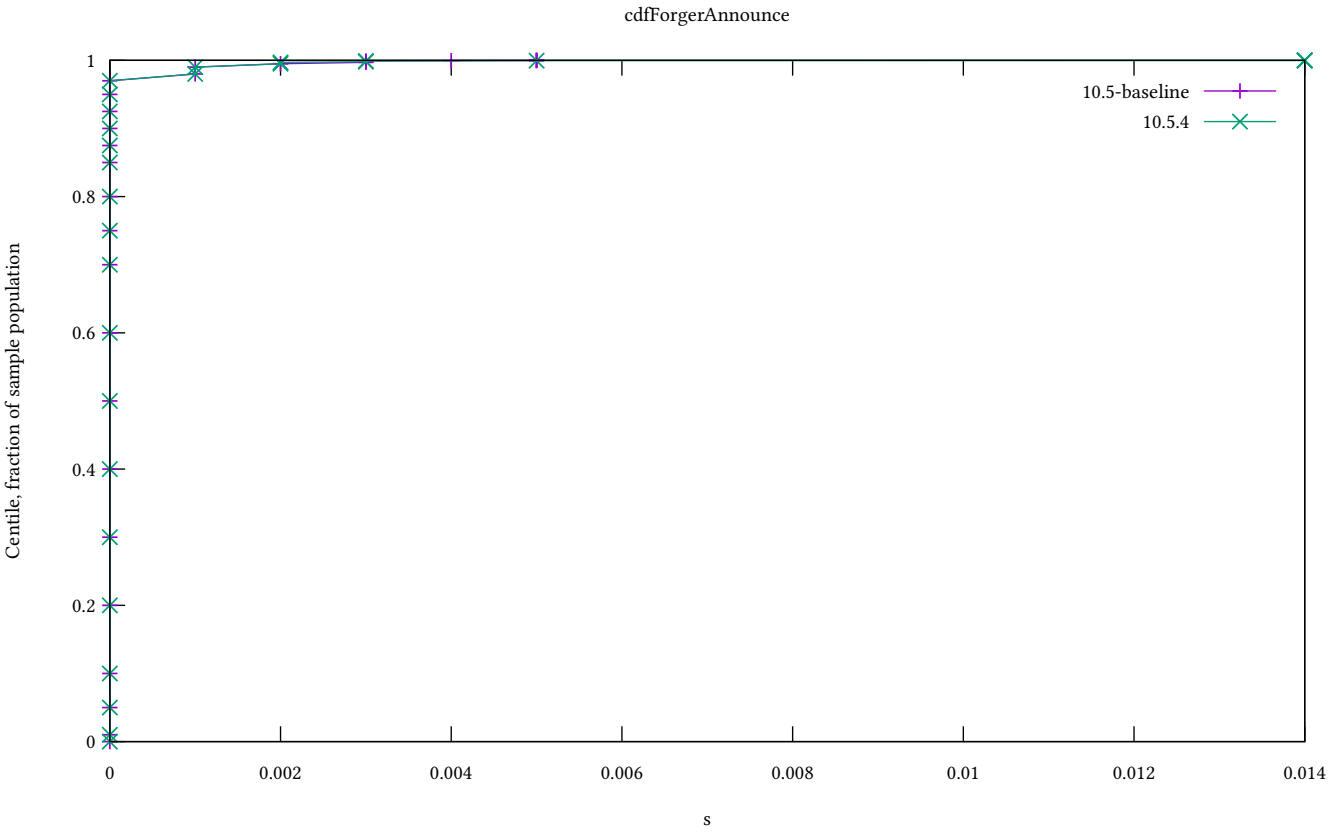

Figure 21: Forged to announced

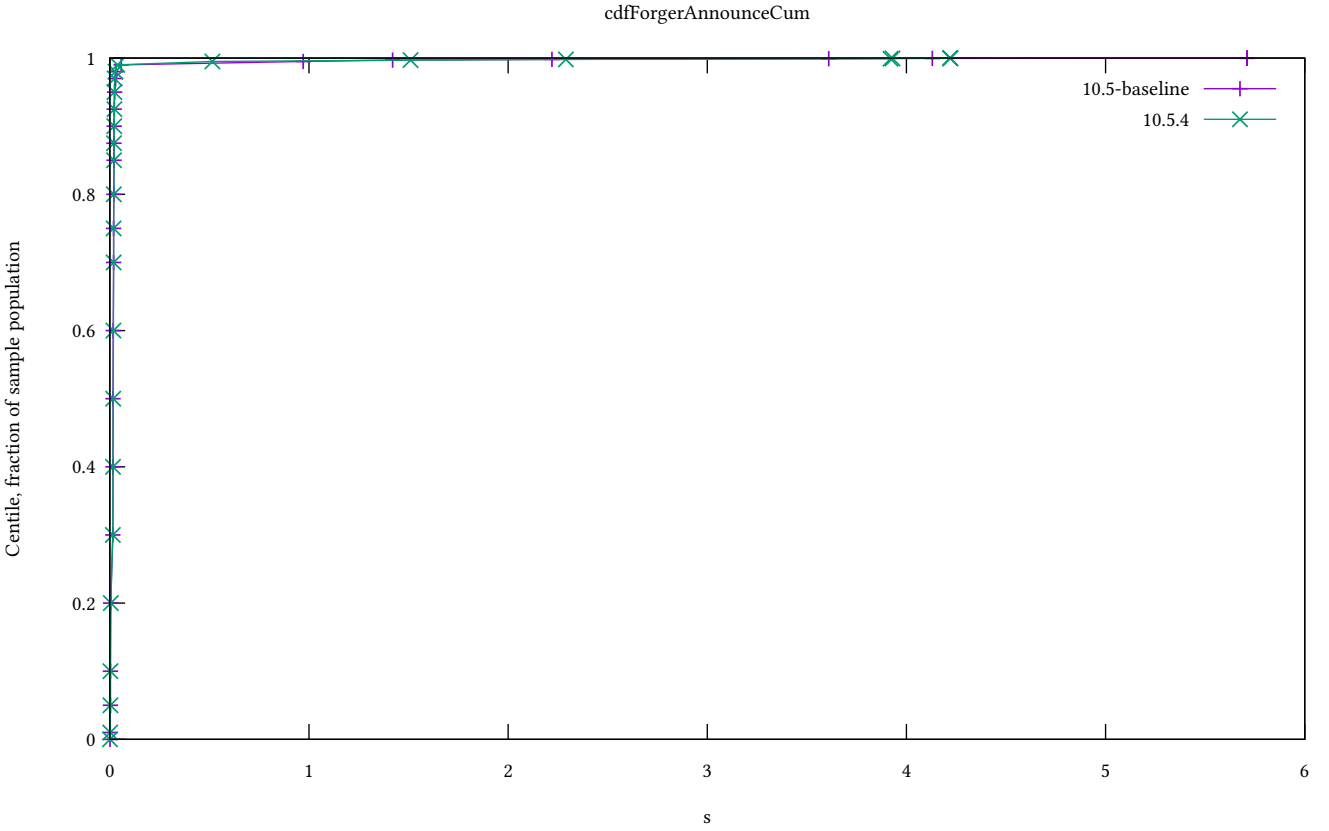

Figure 22: Slot start to announced

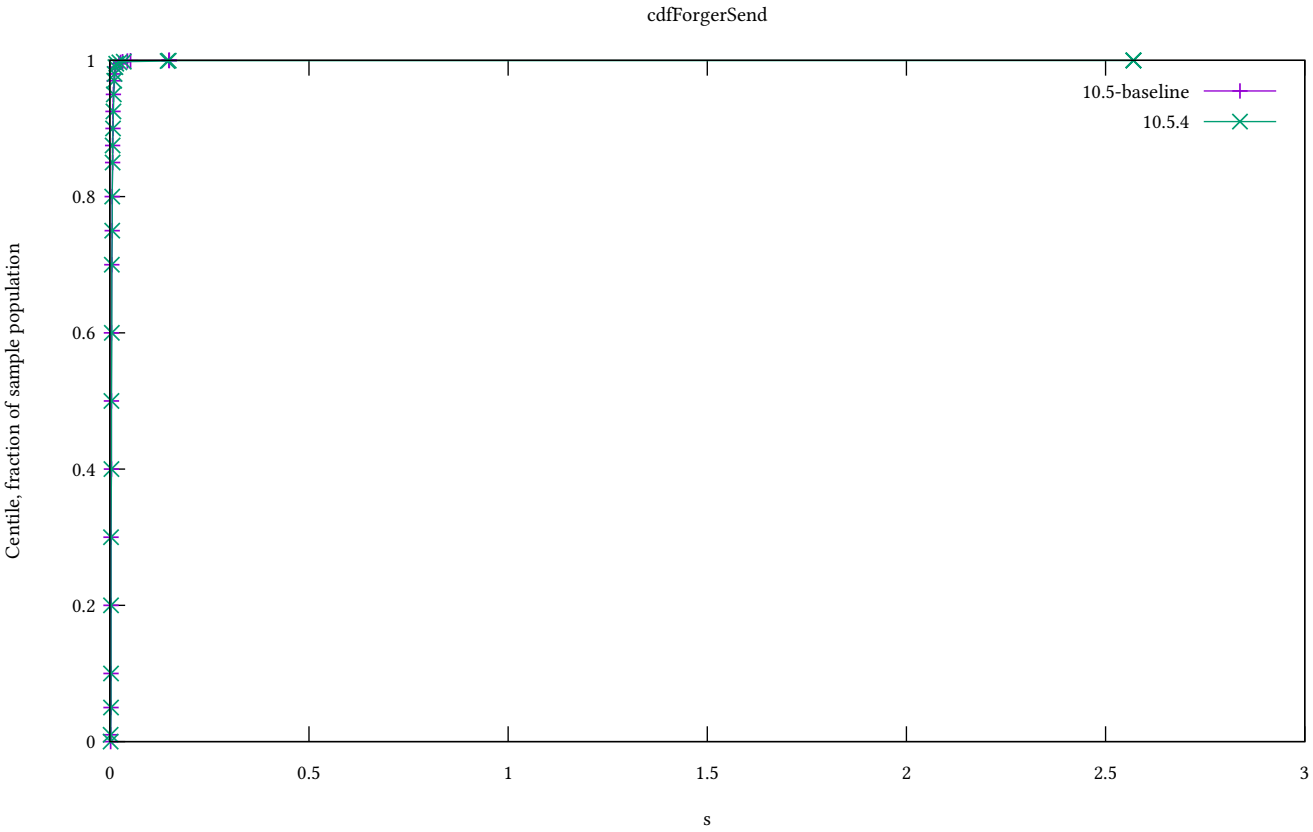

Figure 23: Forged to sending

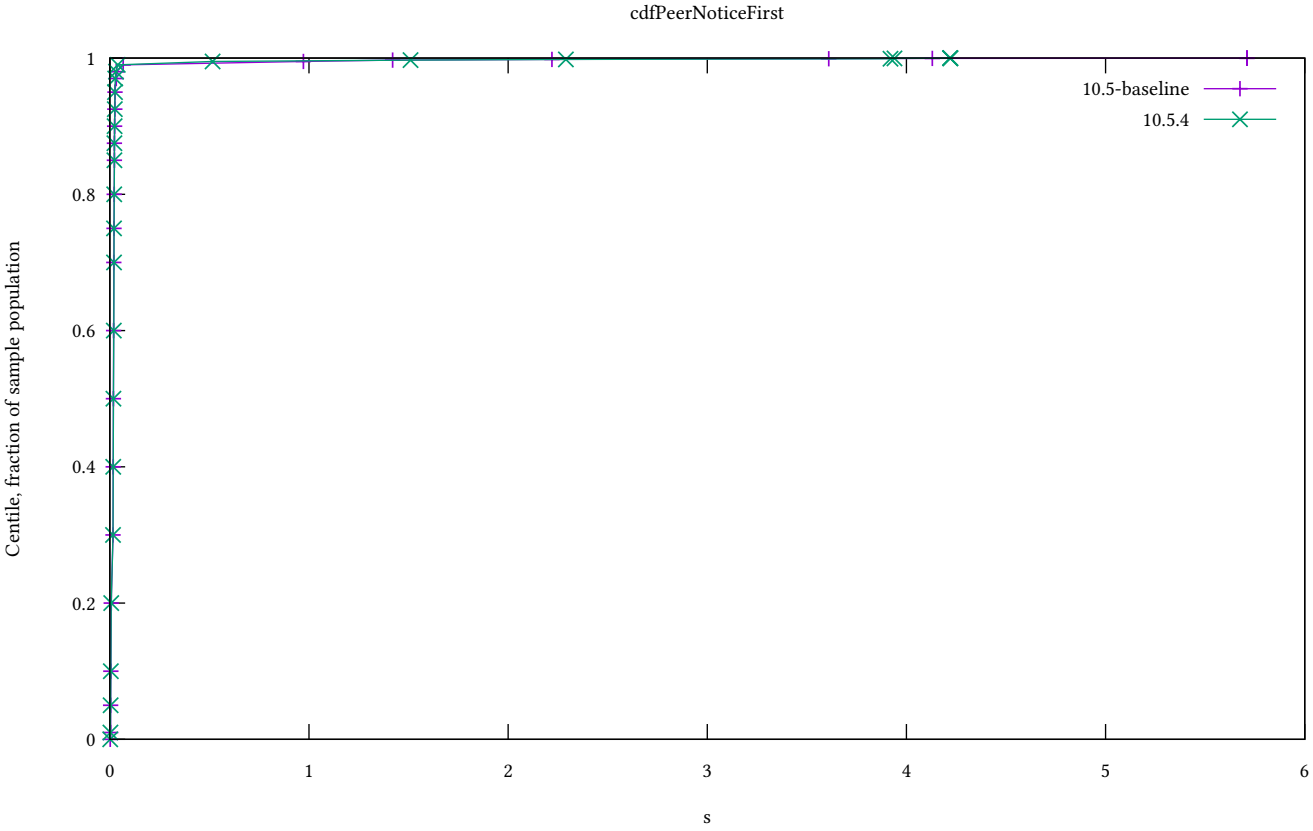

Figure 24: First peer notice

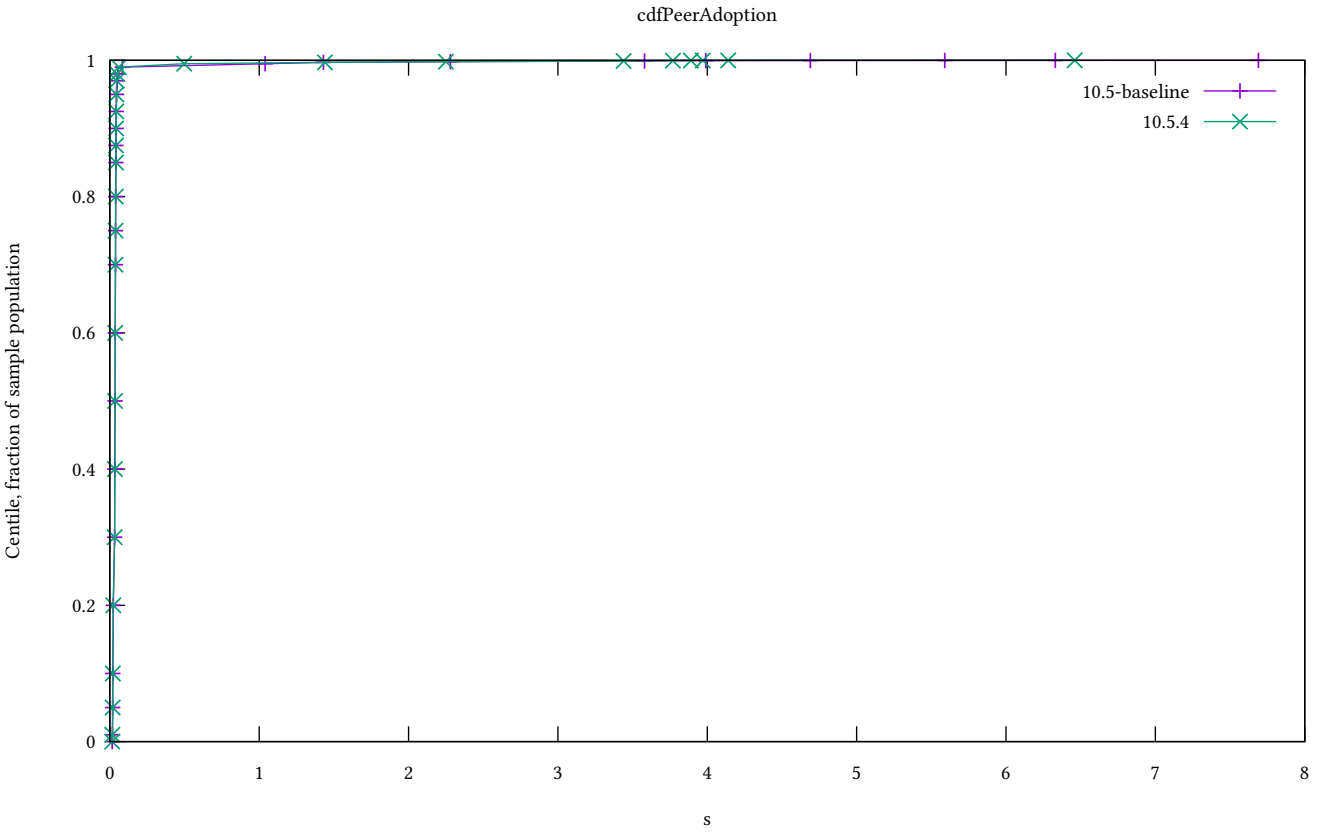

Figure 25: Fetched to adopted

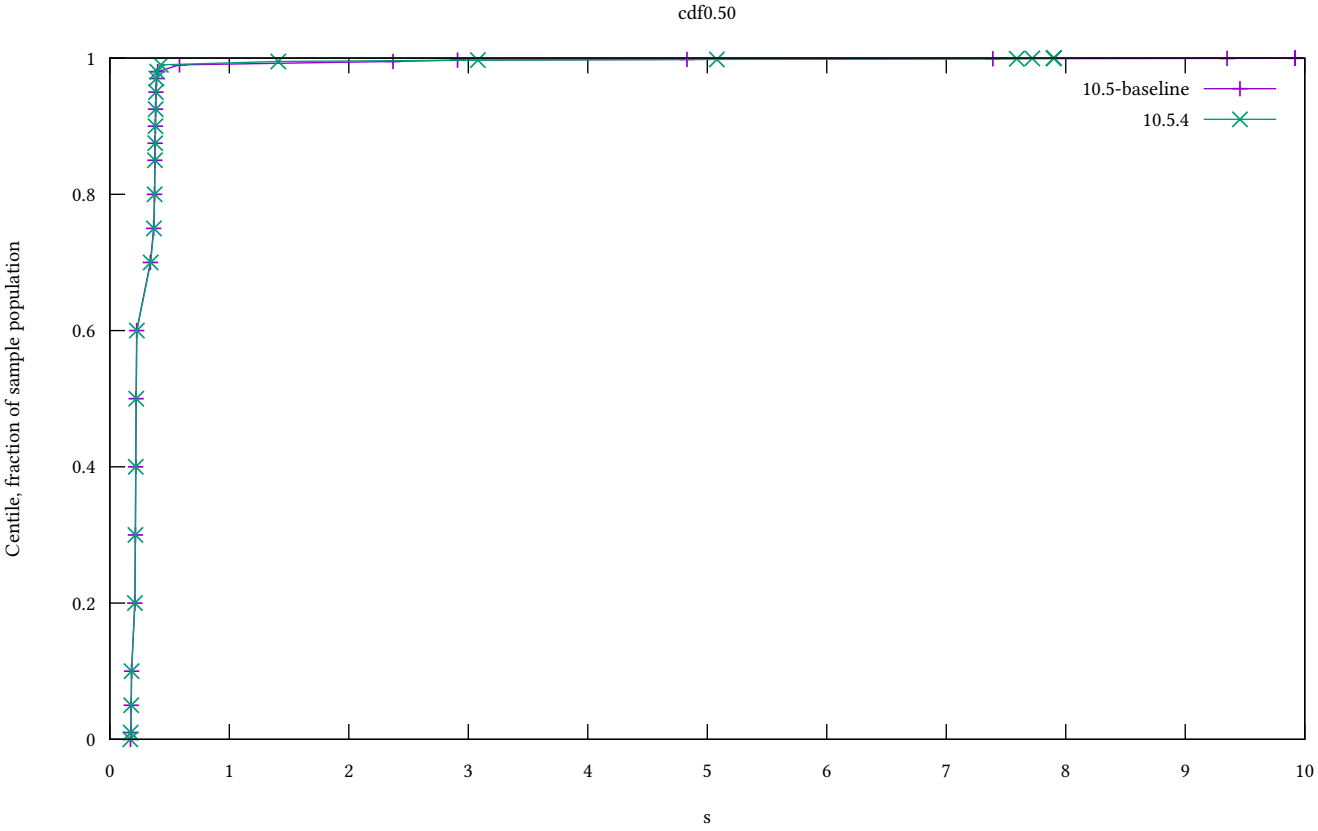

Figure 26: 0.50 adoption

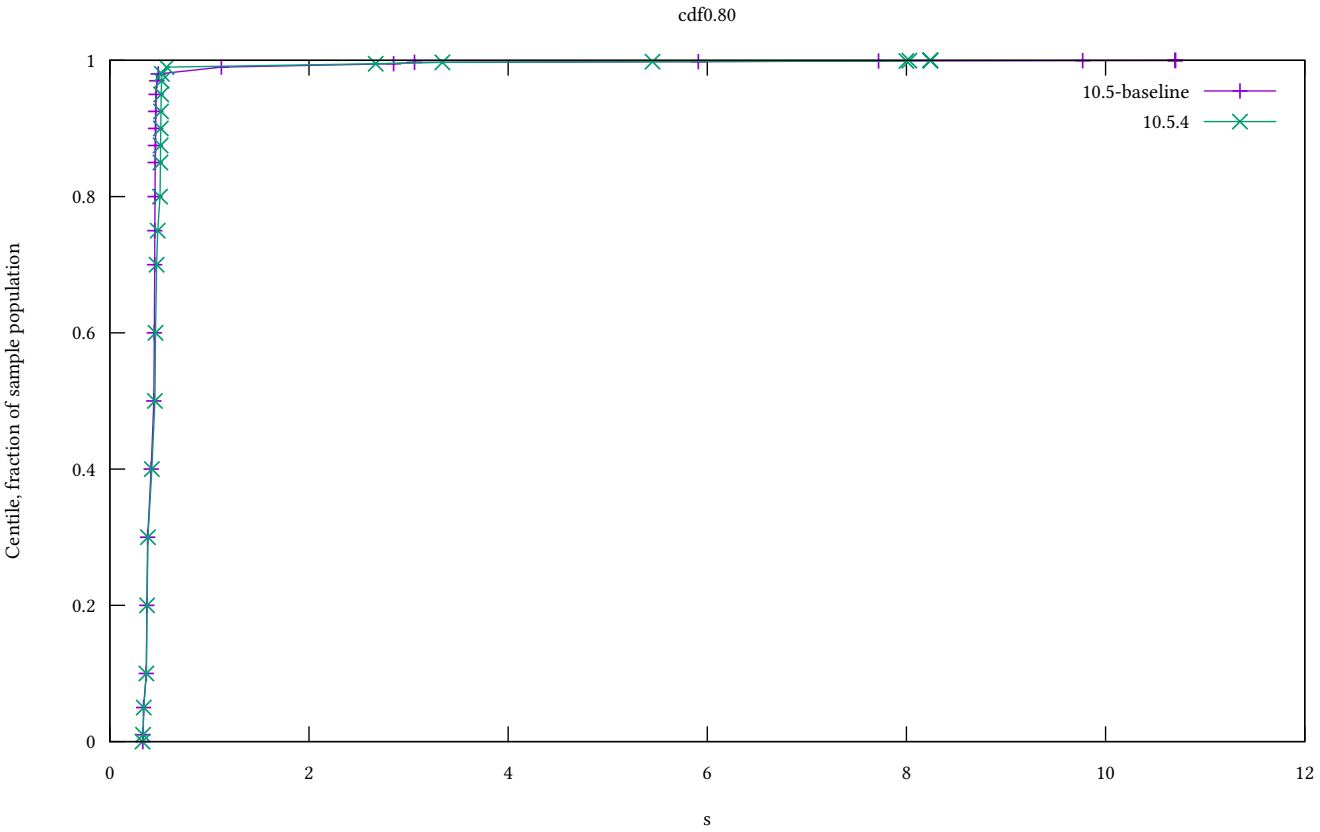

Figure 27: 0.80 adoption

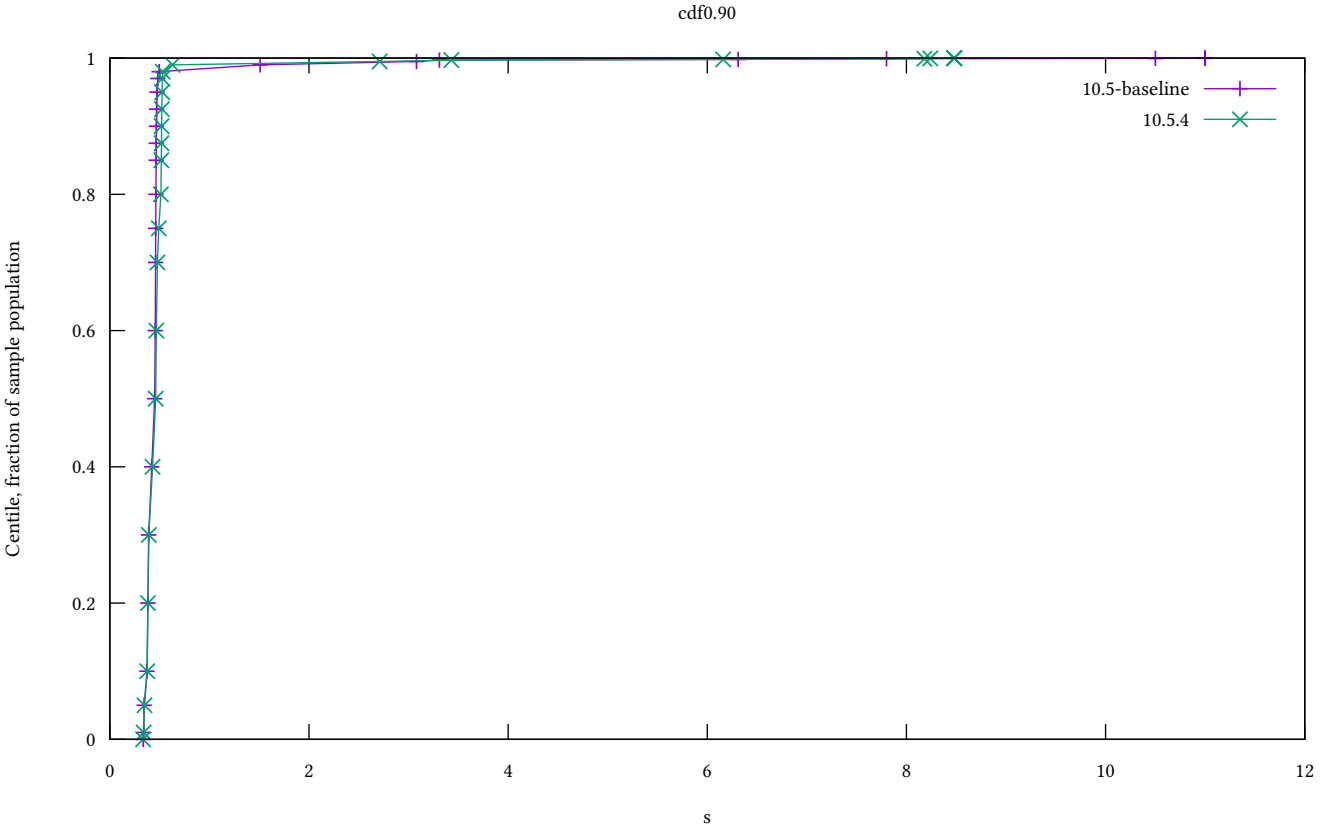

Figure 28: 0.90 adoption

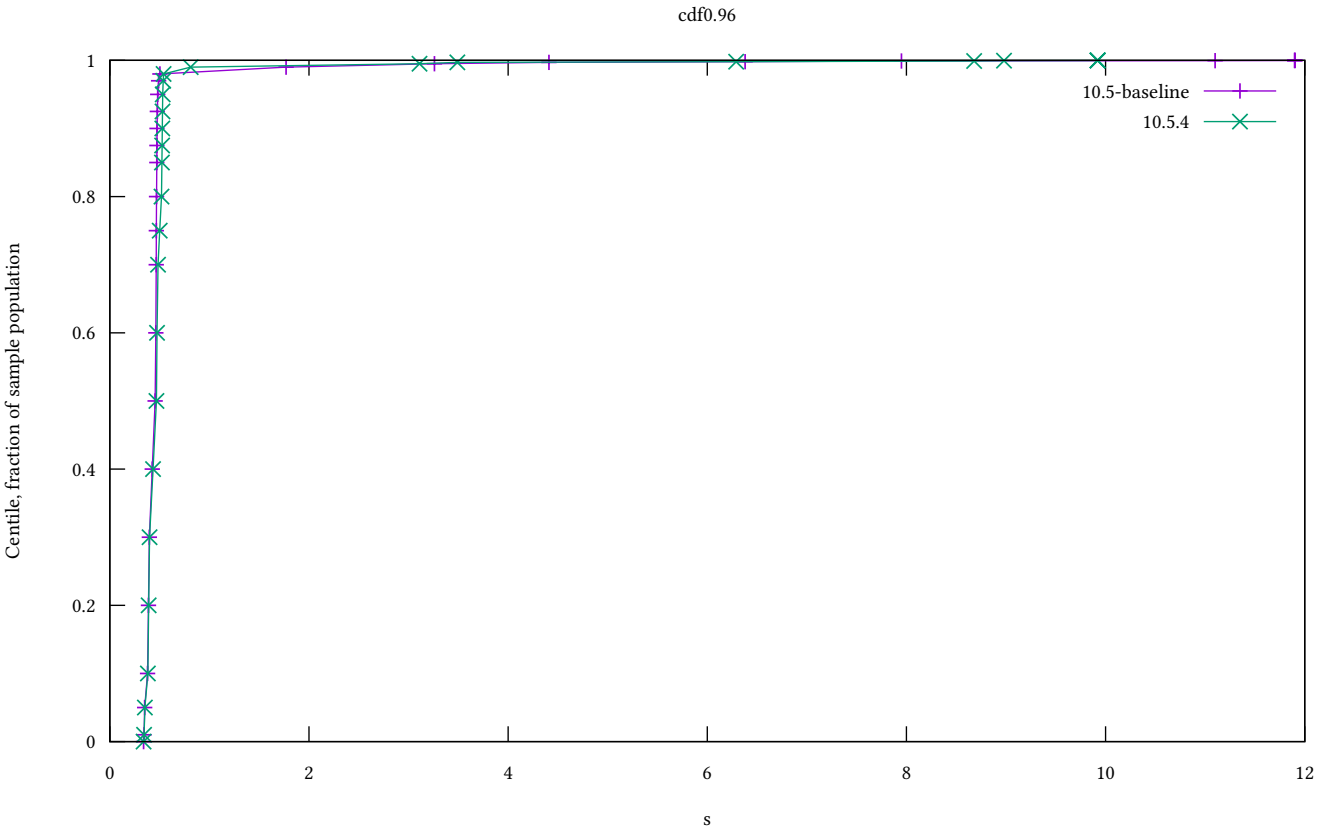

Figure 29: 0.96 adoption

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

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