11.1.0 against 11.0.1

Plutus countdown loop workload

Federico Mastellone, Cardano Performance & Tracing

2026-08-25

Contents

Manifest

We compare `11.0.1` (Conway) and `11.1.0` (Conway) relative to `11.0.1` (Conway), under Plutus countdown loop workload.

	11.0.1	11.1.0
Analysis date	2026-05-06	2026-08-22
Cluster system start date	2026-05-05	2026-08-21
Cluster system start time	08:18:29	13:26:38
Identifier	11.0.1	11.1.0
Run batch	11.0.1	11.1.0
GHC version	9.6.7	9.6.7
cardano-node version	11.0.1	11.1.0
ouroboros-consensus version	3.0.1.0	4.1.0.0
ouroboros-network version	1.1.0.0	1.2.0.0
cardano-ledger-core version	1.20.0.0	1.21.0.0
plutus-core version	1.63.0.0	1.65.0.0
cardano-crypto version	1.3.0	1.3.0
cardano-prelude version	0.2.2.0	0.2.2.0
cardano-node git	6ca8737	55f0717
ouroboros-consensus git	c87aa76	c85b71d
ouroboros-network git	a98c885	c45735a
cardano-ledger-core git	94e9618	f649f97
plutus-core git	f92b7d7	b2db512
cardano-crypto git	a741501	a741501
cardano-prelude git	2b6092c	2b6092c
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	897170.55769	900073.98076
Log objects analysed per host	613433.34615	612739.44230
Host run time, s	71863.1	71935.1
Host log line rate, Hz	12.484	12.512
Total log objects analysed	31898534	31862451
Run time, s	71867	71946
Analysed run duration, s	56018	56042
Run time efficiency	0.77	0.77
Node start spread, s	5.7995522	19.712232
Node stop spread, s	2.0789854	2.1763315
Slots analysed	56016	56040
Blocks analysed	2765	2770
Blocks rejected	873	855

Analysis

Resource Usage

	11.0.1	11.1.0	Δ	Δ%
Forge loop starts, units	0.99996	0.99976	−0.000	−0.0
Process CPU usage, %	1.1413	1.15	0.009	0.8
RTS GC CPU usage, %	0.2264	0.2124	−0.014	−6
RTS Mutator CPU usage, %	0.9145	0.9371	0.023	2
Major GCs, events	0.0009	0.0007	−0.000	−24
Minor GCs, events	0.3368	0.3402	0.003	1
Kernel RSS, MiB	6596.7	8360.5	1763.874	27
RTS heap size, MiB	6527.7	8290.5	1762.865	27
RTS live GC dateset, MiB	2689.7	3585.2	895.467	33
RTS alloc rate, MiB/s	9.9606	10.055	0.095	1.0
Network reads, KiB/s	0.0	0.0	0.000	NaN
Network writes, KiB/s	0.0	0.0	0.000	NaN
Filesystem reads, KiB/s	0.0	0.0	0.000	NaN
Filesystem writes, KiB/s	205.92	208.91	2.993	1
CPU 85% spans, slots	7.6677	8.0727	0.405	5

Anomaly control

	11.0.1	11.1.0	Δ	Δ%
Blocks per host, blocks	71.577	71.885	0.308	0.4
Filtered to chained blocks, :	0.761	0.7637	0.003	0.4
Chained to forged blocks, :	0.9773	0.9699	−0.007	−0.8
Height & slot battles, blocks	0.00108	0.00036	−0.001	−67
Block size, Bytes	2996.	2996.1	0.027	0.0

Forging

	11.0.1	11.1.0	Δ	Δ%
Started forge loop iteration, s	0.00137	0.00117	−0.000	−14
Acquired block context, s	6.3e-5	6.1e-5	−0.000	0
Acquired ledger state, s	0.00011	0.00011	−0.000	0
Acquired ledger view, s	3.3e-5	3.1e-5	−0.000	0
Leadership check duration, s	0.00041	0.00042	0.000	0
Ledger ticking, s	0.01744	0.01735	−0.000	0
Mempool snapshotting, s	0.00167	0.00144	−0.000	−14
Leadership to forged, s	0.00021	0.00021	−0.000	0
Forged to announced, s	0.00056	0.00053	−0.000	0
Forged to sending, s	0.0049	0.00495	0.000	0
Forged to self-adopted, s	0.03952	0.04076	0.001	3
Slot start to announced, s	0.02187	0.02132	−0.001	−3

Individual peer propagation

	11.0.1	11.1.0	Δ	Δ%
First peer notice, s	0.0234	0.02288	−0.001	−2
First peer fetch, s	0.02929	0.02913	−0.000	−0.5
Notice to fetch request, s	0.00128	0.00129	0.000	0
Fetch duration, s	0.12351	0.1303	0.007	5
Fetched to announced, s	0.00069	0.00071	0.000	0
Fetched to sending, s	0.04291	0.04308	0.000	0.4
Fetched to adopted, s	0.04145	0.04182	0.000	0.9

End-to-end propagation

	11.0.1	11.1.0	Δ	Δ%
0.50 adoption, s	0.27863	0.28478	0.006	2
0.80 adoption, s	0.43689	0.46216	0.025	6
0.90 adoption, s	0.4472	0.47268	0.025	6
0.92 adoption, s	0.44976	0.47518	0.025	6
0.94 adoption, s	0.45218	0.47808	0.026	6
0.96 adoption, s	0.45559	0.48103	0.025	6
0.98 adoption, s	0.46157	0.49067	0.029	6
1.00 adoption, s	0.48232	0.52076	0.038	8

Observations

Resources

1. Process CPU usage is practically unchanged (+0.8%); Major GCs occur 24% less frequently.
2. Kernel RSS and RTS heap size increase by 1.7 GiB or 27%; the RTS live GC dataset by 895 MiB or 33%.
3. Observed CPU 85% spans lengthen by 5%.

Forging

1. Block production timings show no significant changes (all deltas sub-millisecond to near-jitter).

Peer propagation

1. Fetch duration regresses by 7ms or 5%; the remaining diffusion metrics show no significant changes.

End-to-end propagation

1. We observe an increase in end-to-end propagation times of 6% in the 80th centile and above, growing to 8% (+38ms) in the 100th centile (median +2%).
2. This increase can be attributed to the additional time spent in Fetch duration.

Addendum: Plutus Workload Calibration

	11.0.1	11.1.0
Protocol Parameters		
block budget		
memory	62000000	62000000
steps	20000000000	20000000000
tx budget		
memory	14000000	14000000
steps	10000000000	10000000000
Unused Budget		
block budget		
- memory	6002096	6002096
- - %	9.68%	9.68%
- steps	7580301044	7580301044
- - %	37.90%	37.90%
tx budget		
- memory	524	524
- - %	0.00%	0.00%
- steps	6895075261	6895075261
- - %	68.95%	68.95%
limiting factor	memory	memory
loops per tx	3777	3777
txns per block	4	4
loops per block	15108	15108
~ tx fee	1214868	1214868
~ tx size	533	533

Appendix A: charts

Cluster performance charts

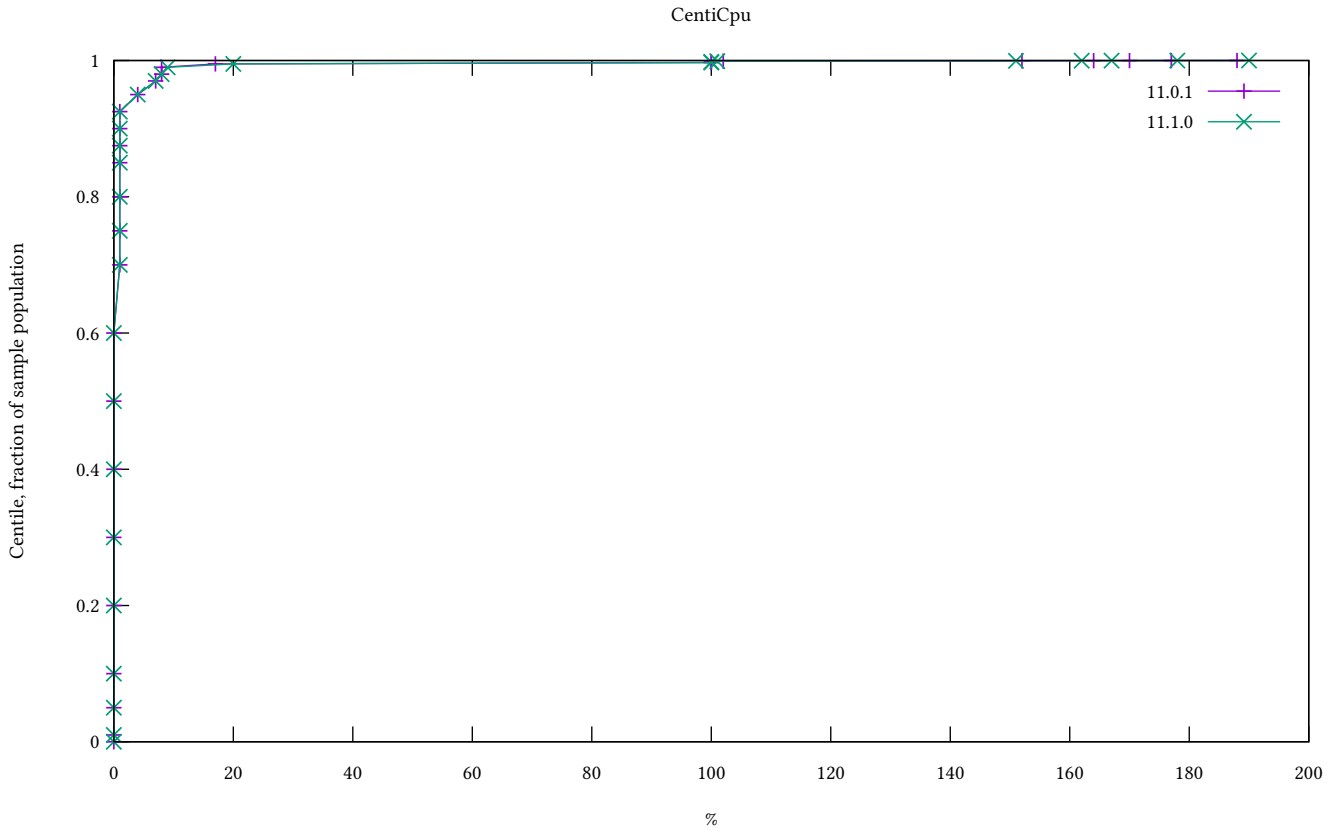

Figure 1: Process CPU usage

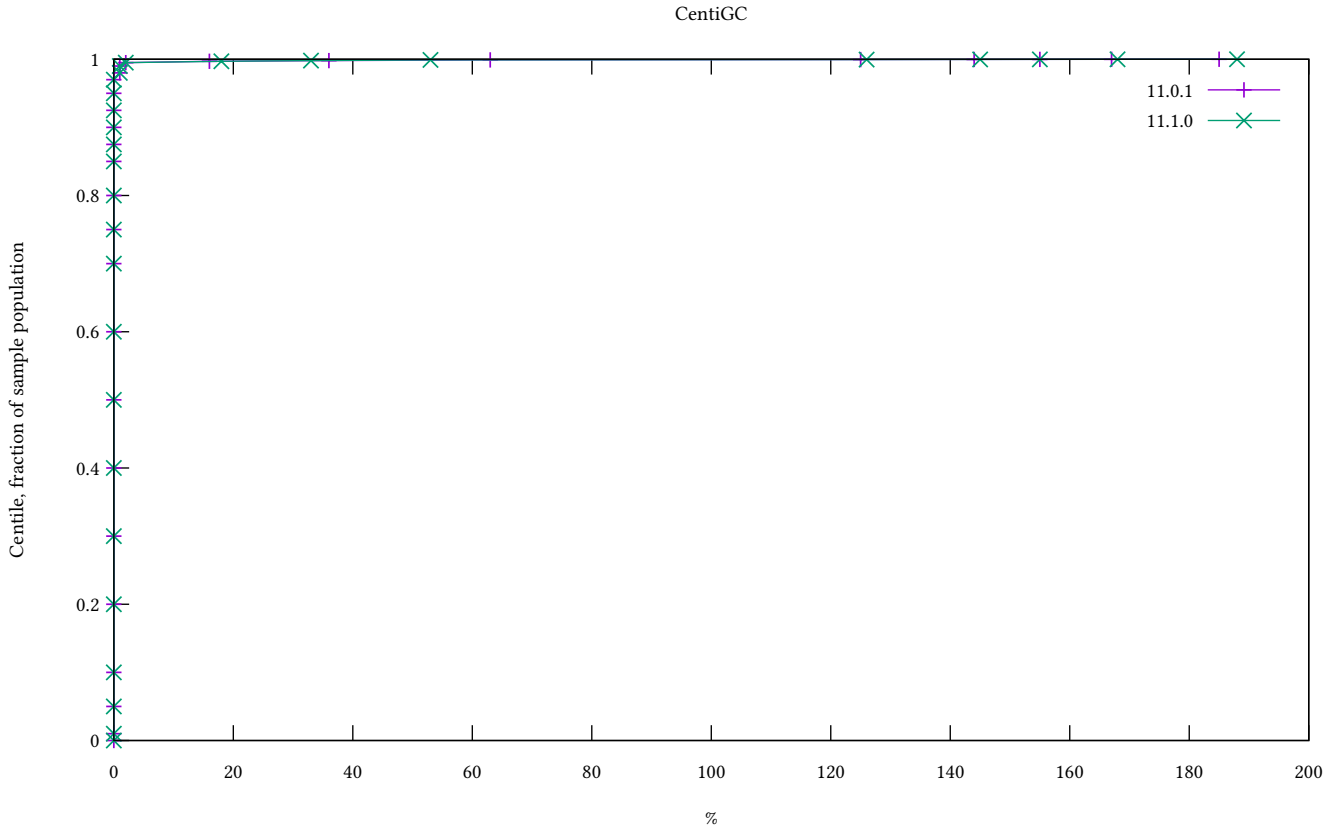

Figure 2: RTS GC CPU usage

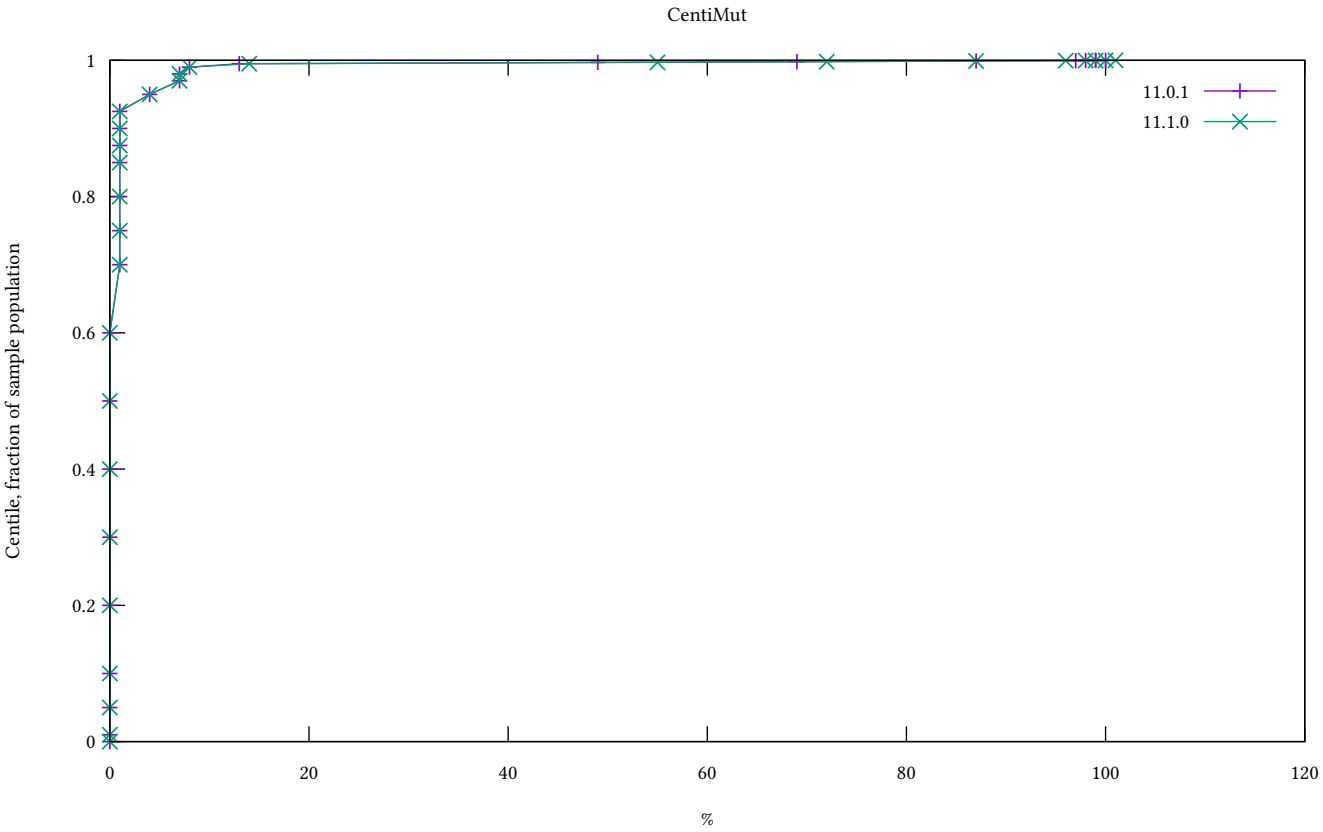

Figure 3: RTS Mutator CPU usage

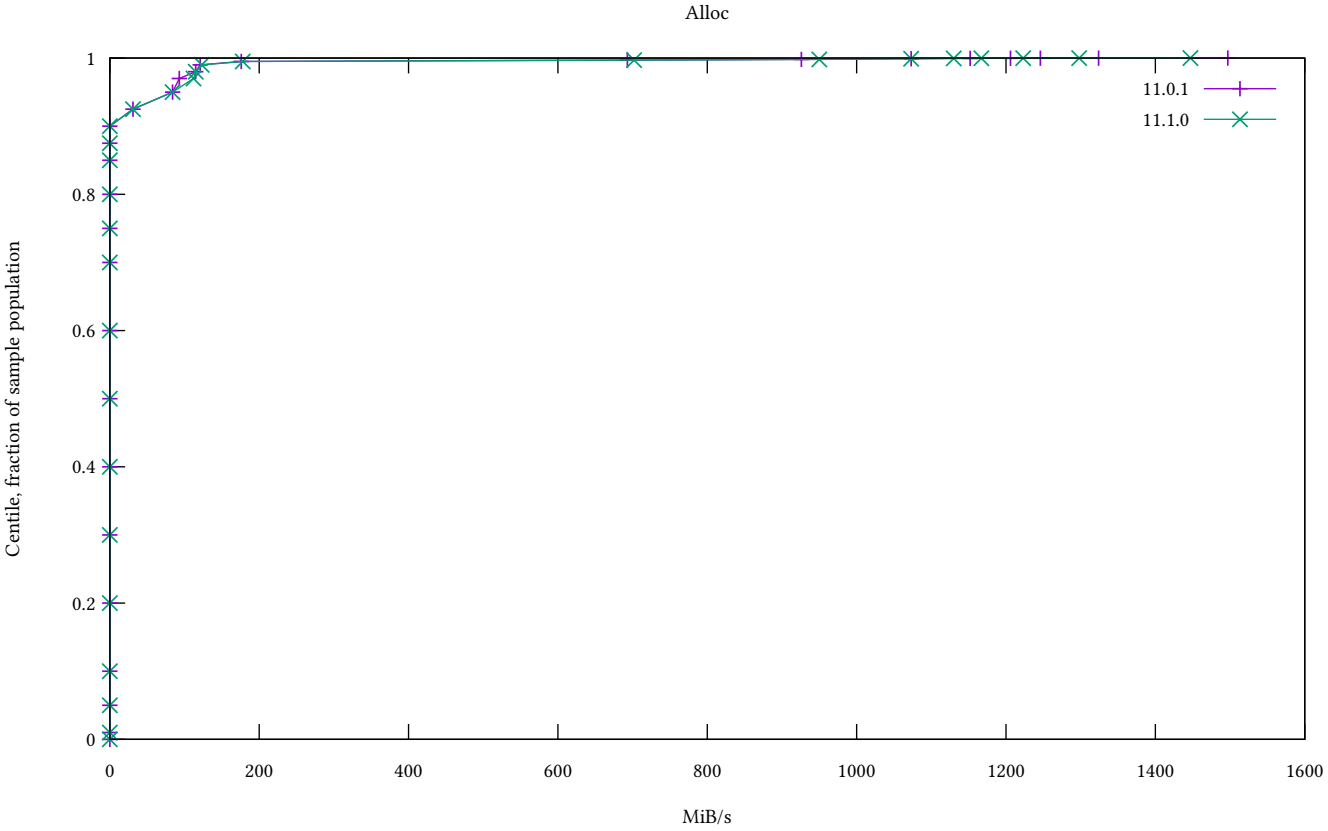

Figure 4: RTS alloc rate

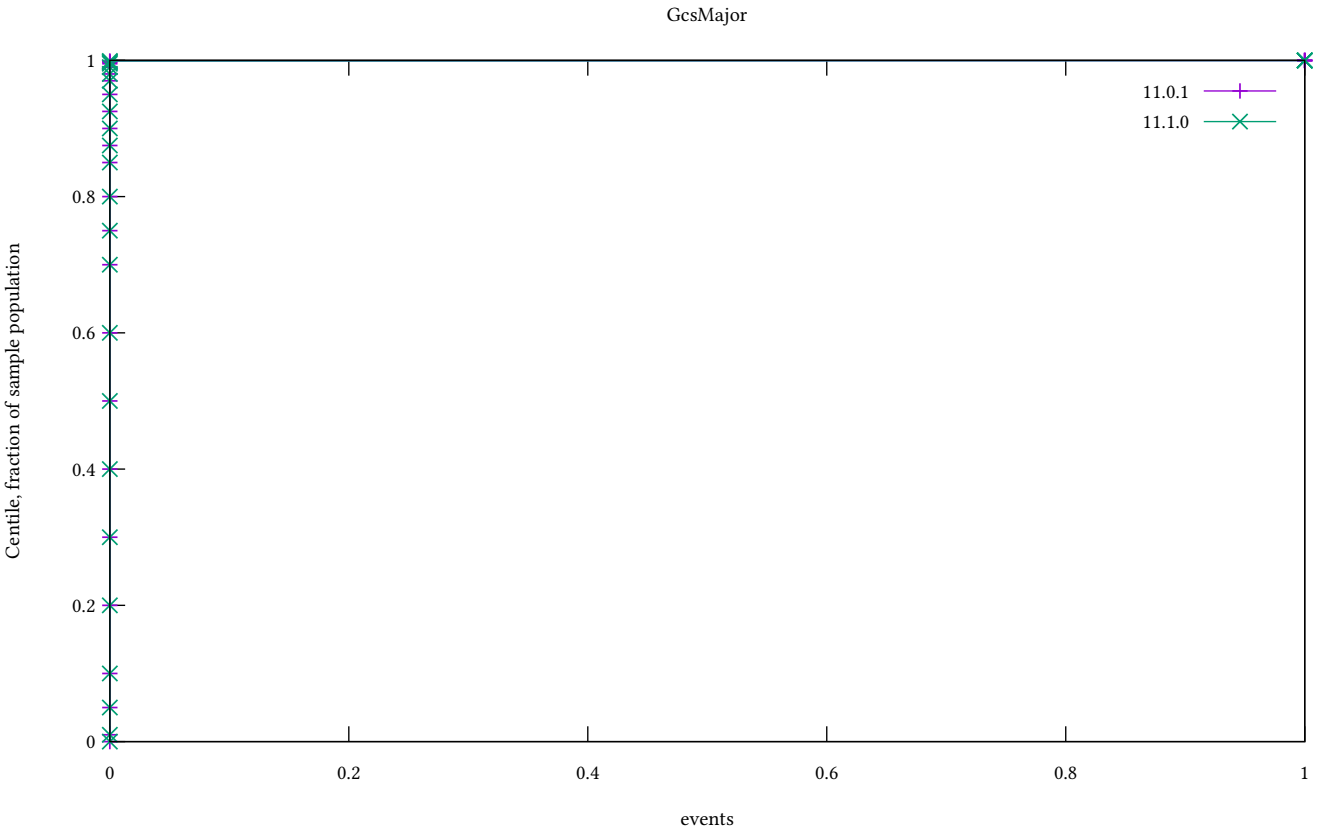

Figure 5: Major GCs

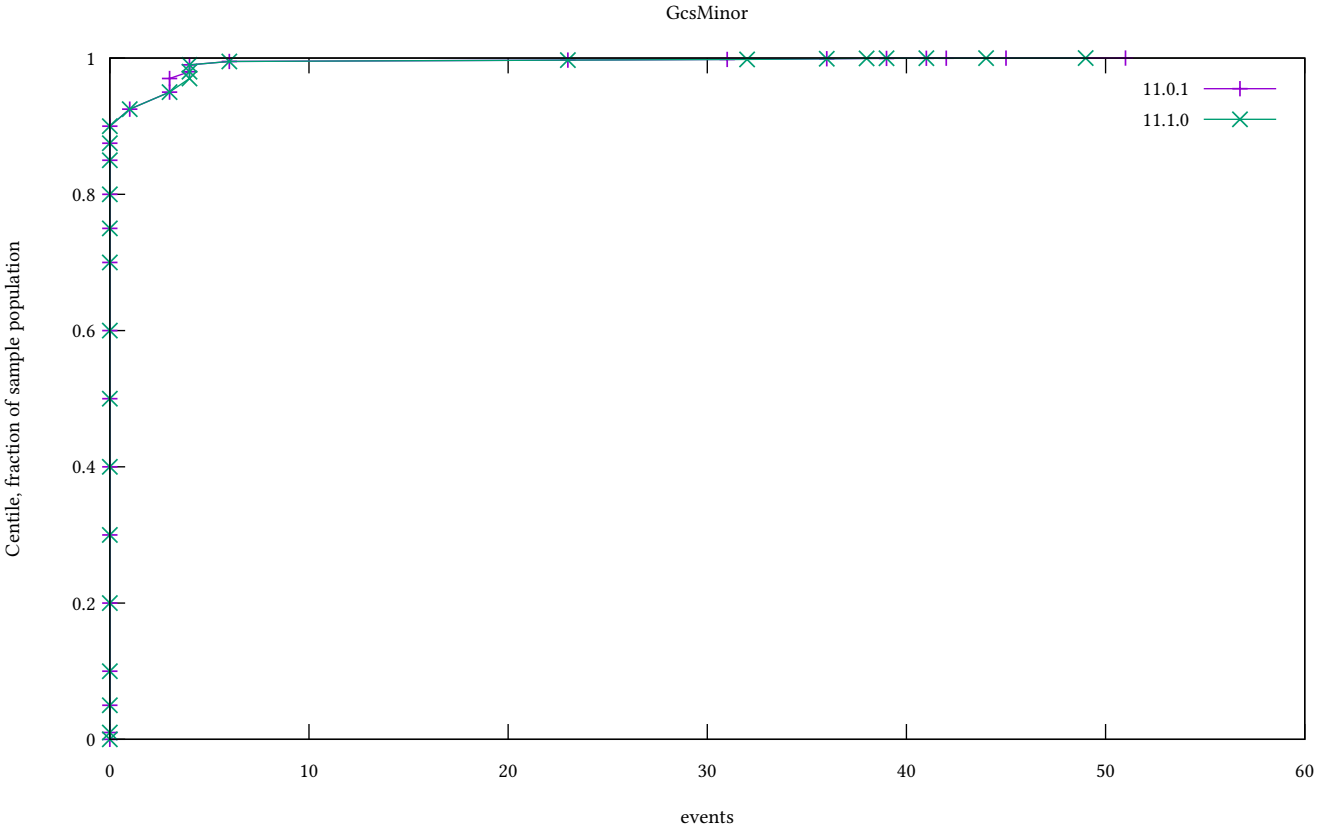

Figure 6: Minor GCs

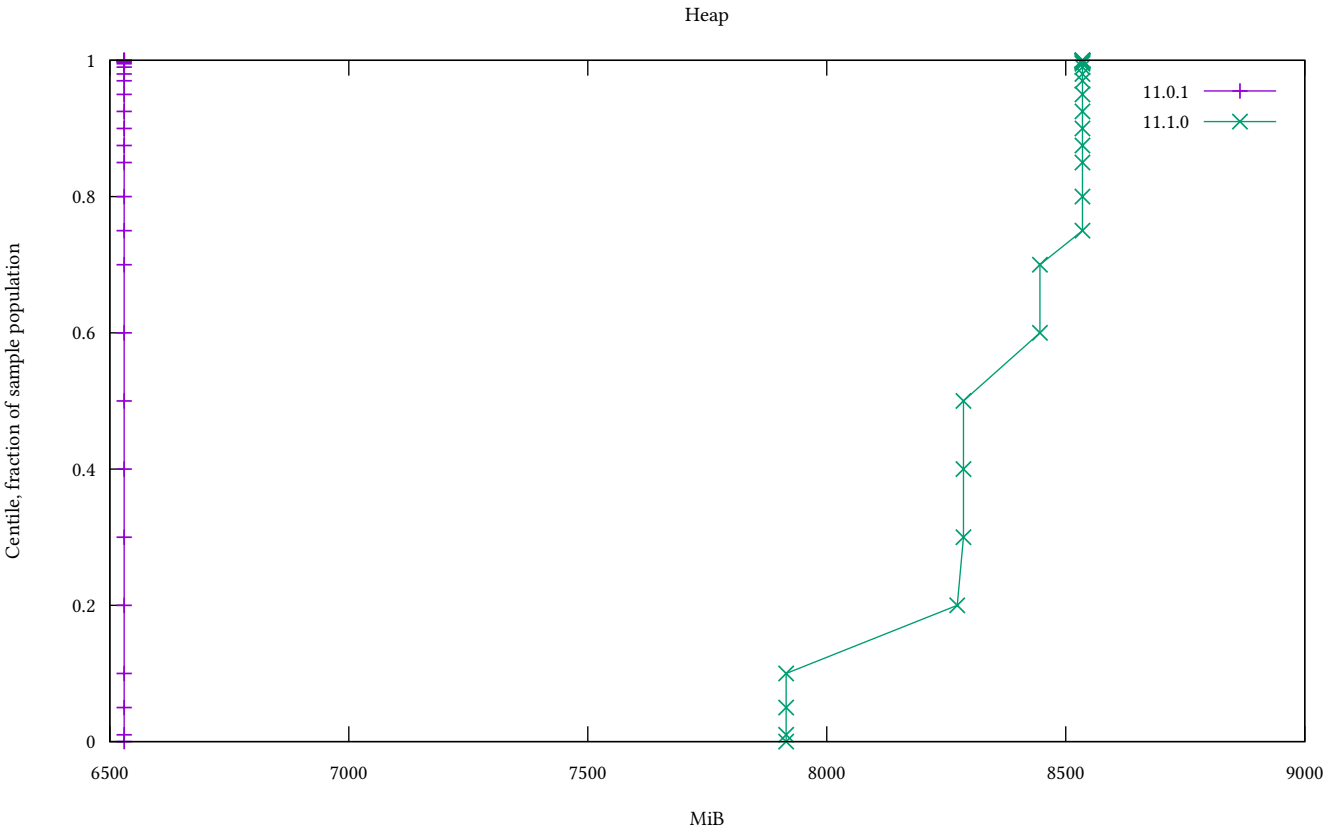

Figure 7: RTS heap size

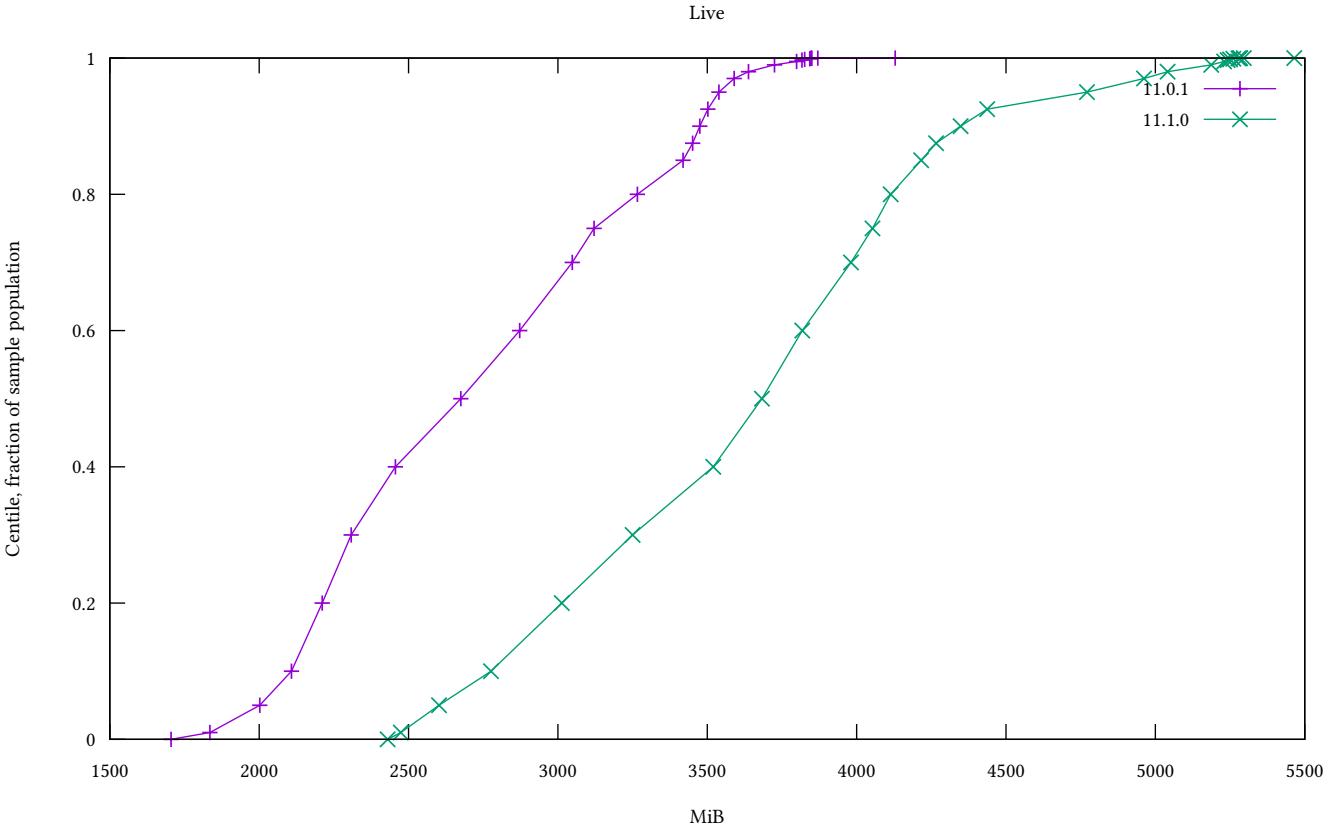

Figure 8: RTS live GC dateset

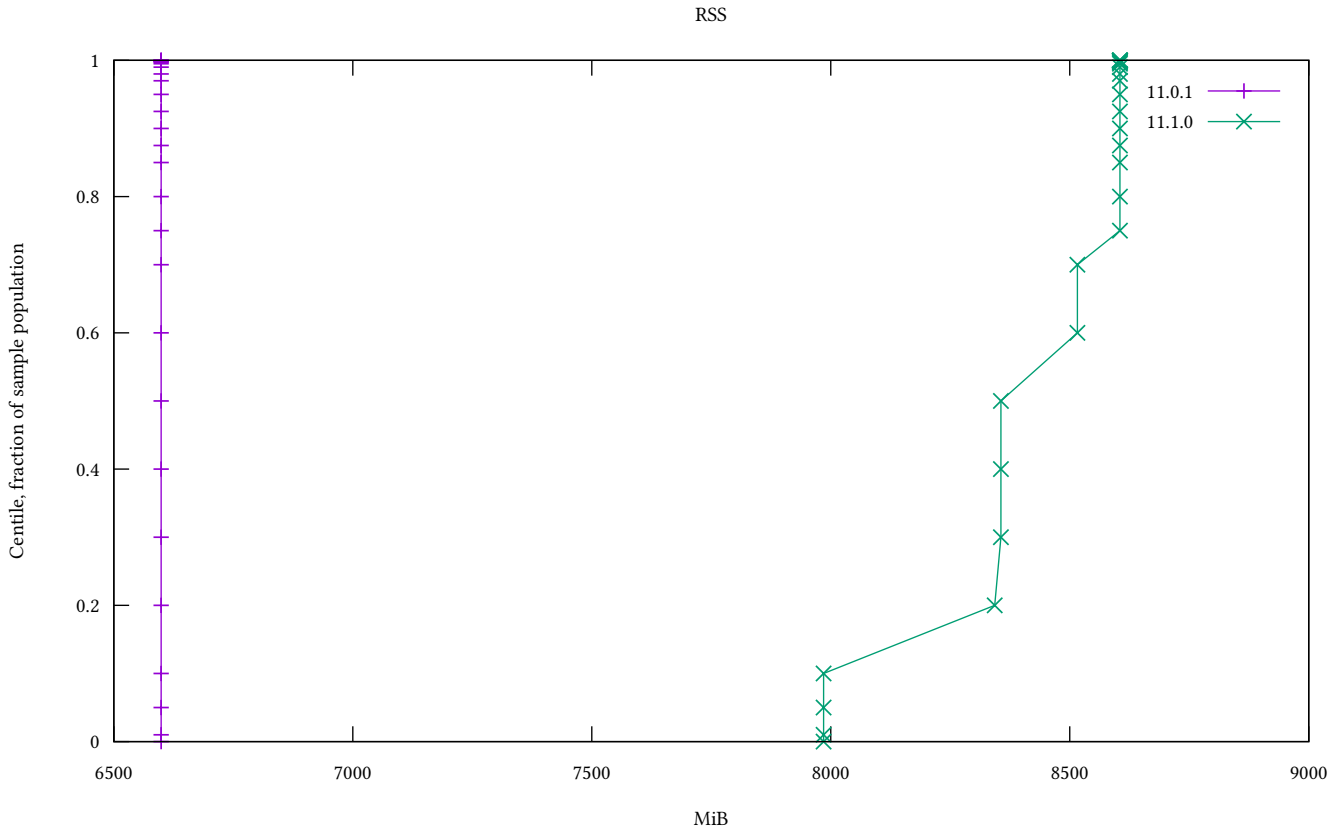

Figure 9: Kernel RSS

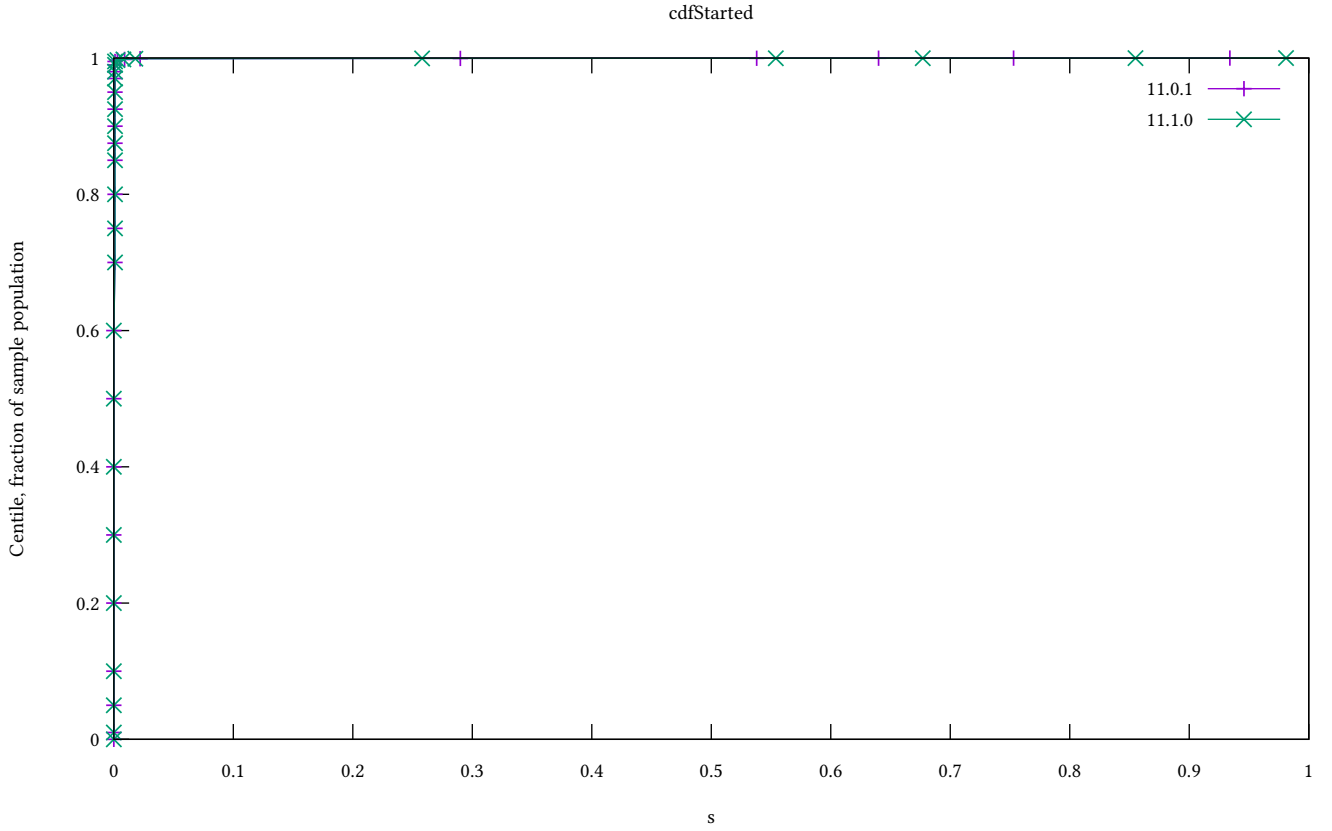

Figure 10: Forge loop tardiness

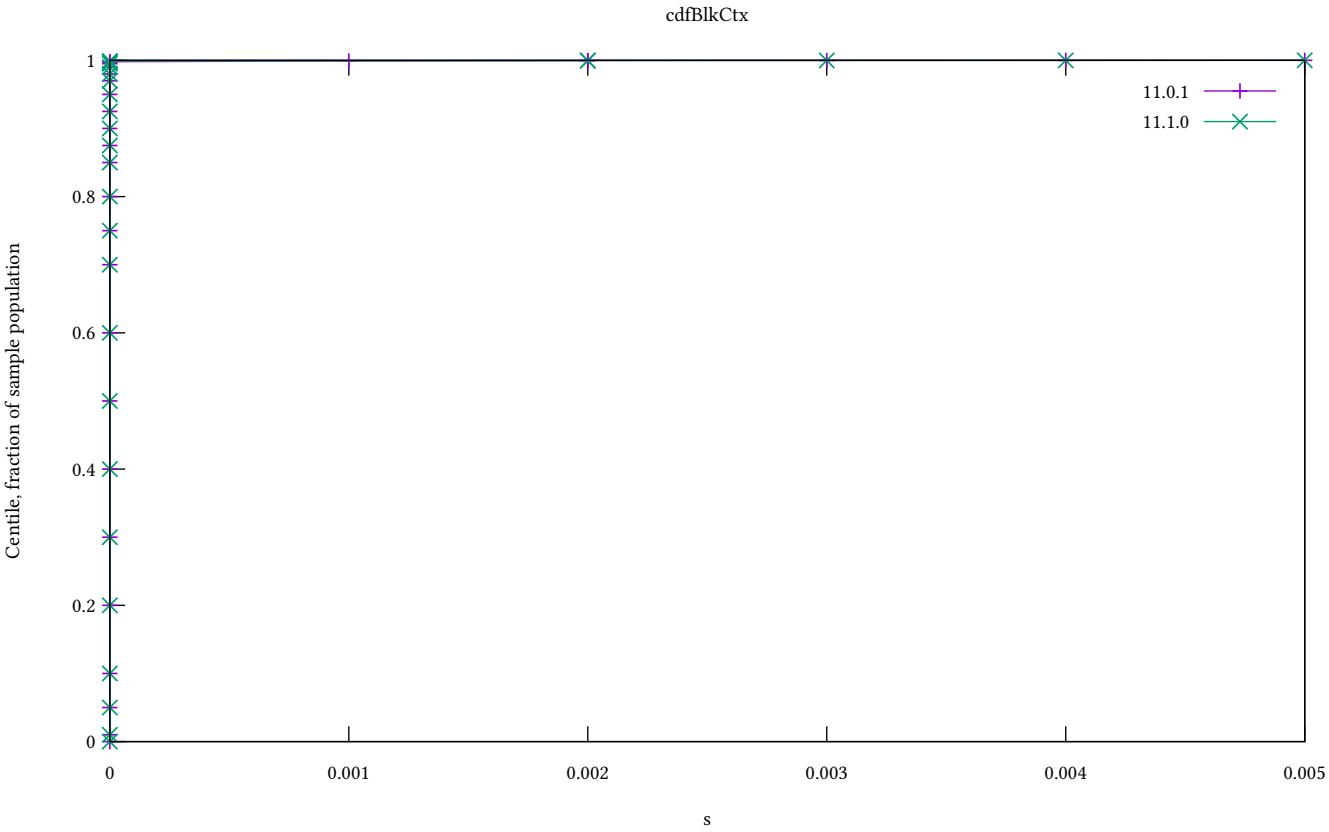

Figure 11: Block context acquisition delay

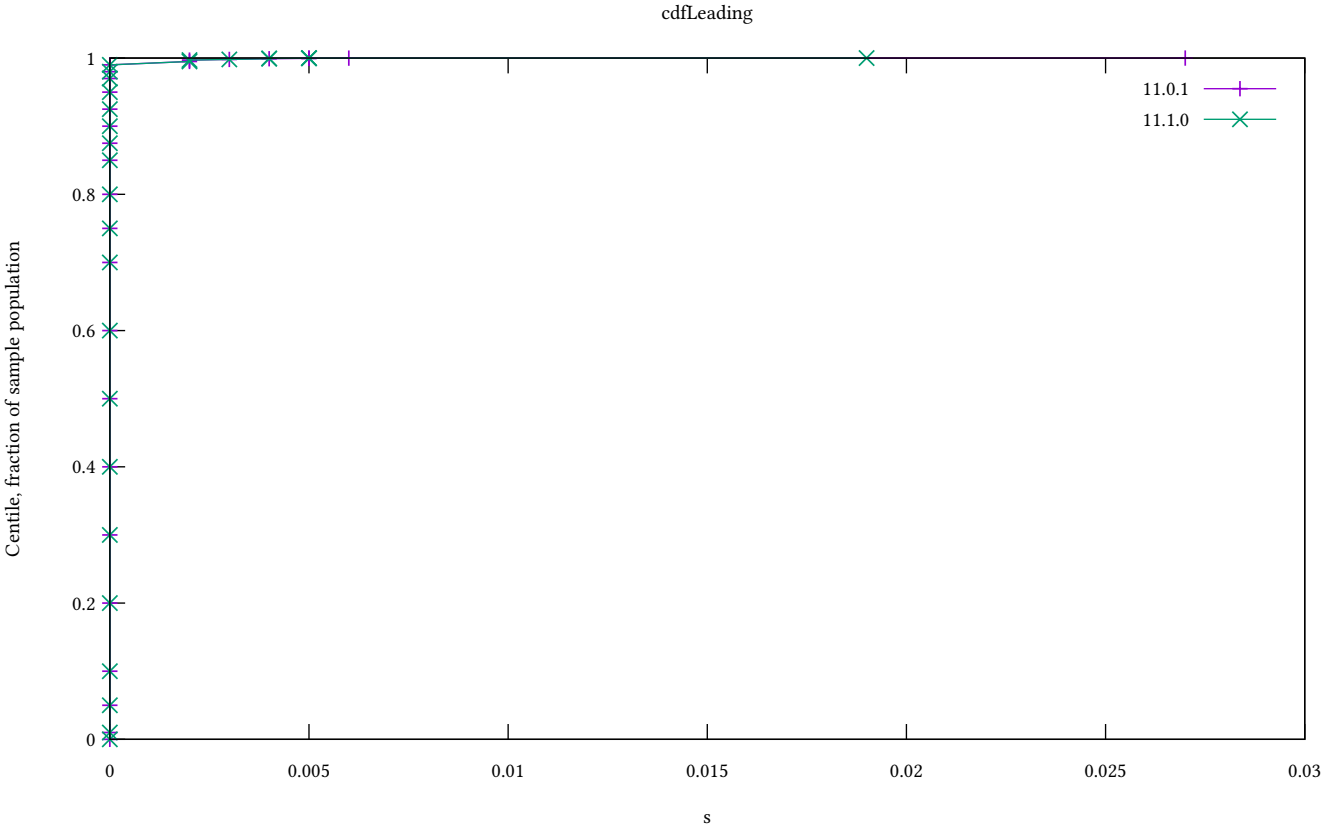

Figure 12: Leadership check duration

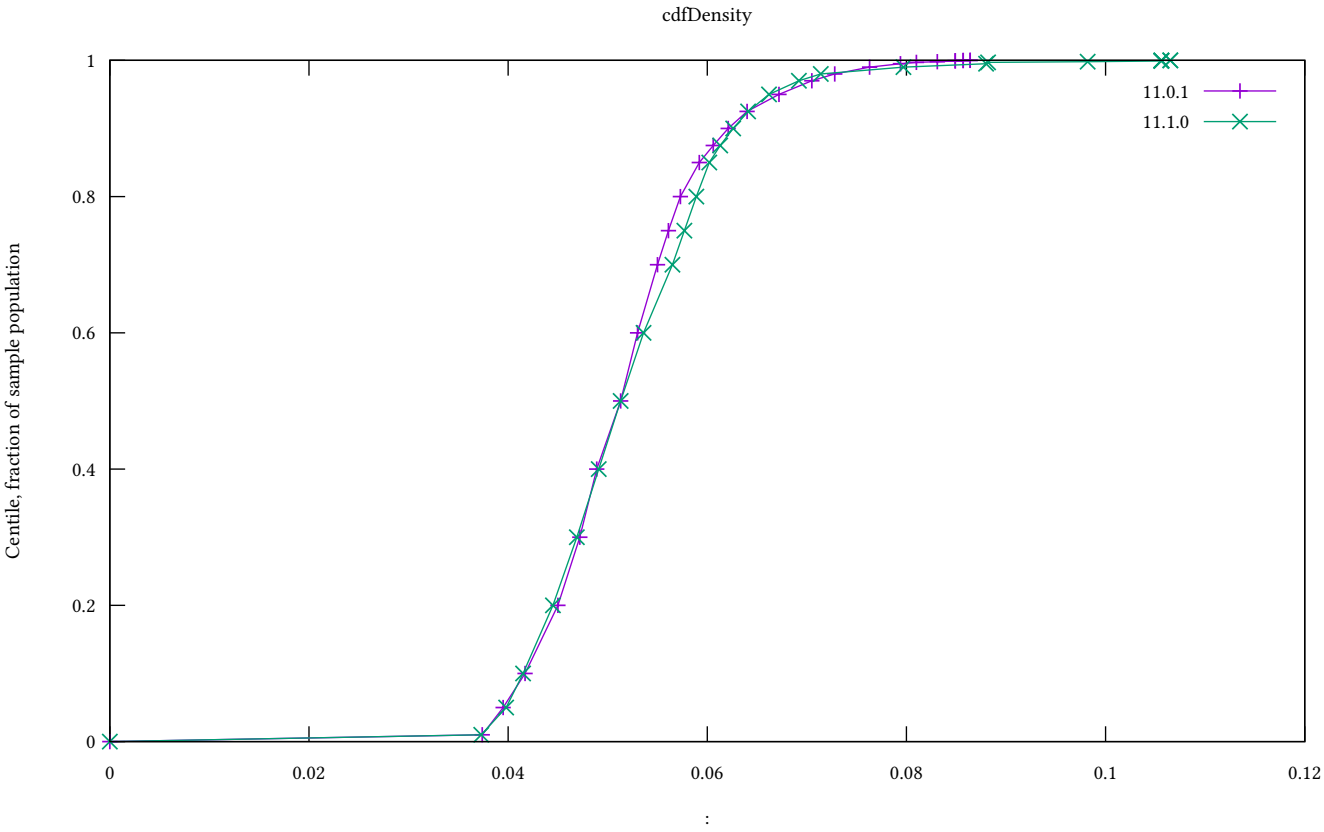

Figure 13: Chain density

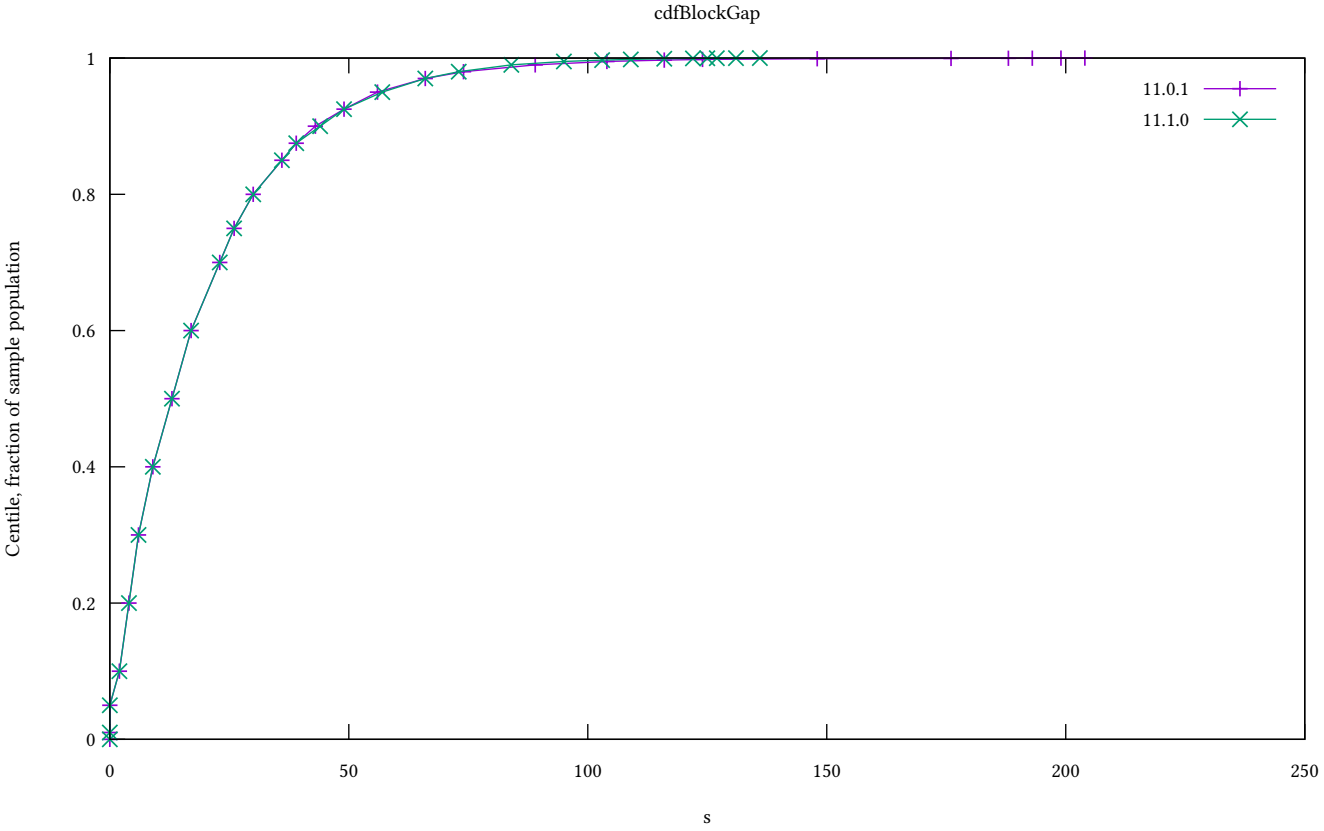

Figure 14: Interblock gap

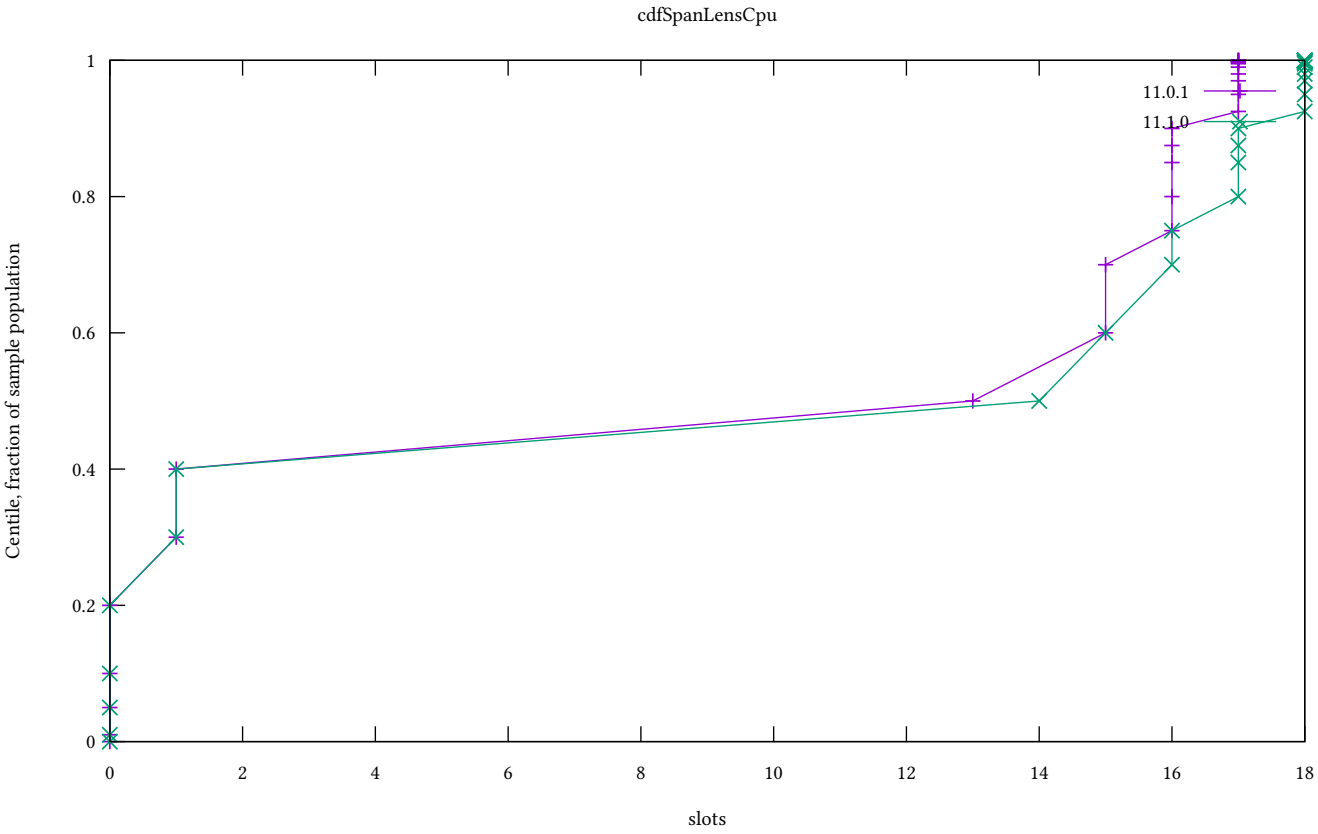

Figure 15: CPU 85% spans

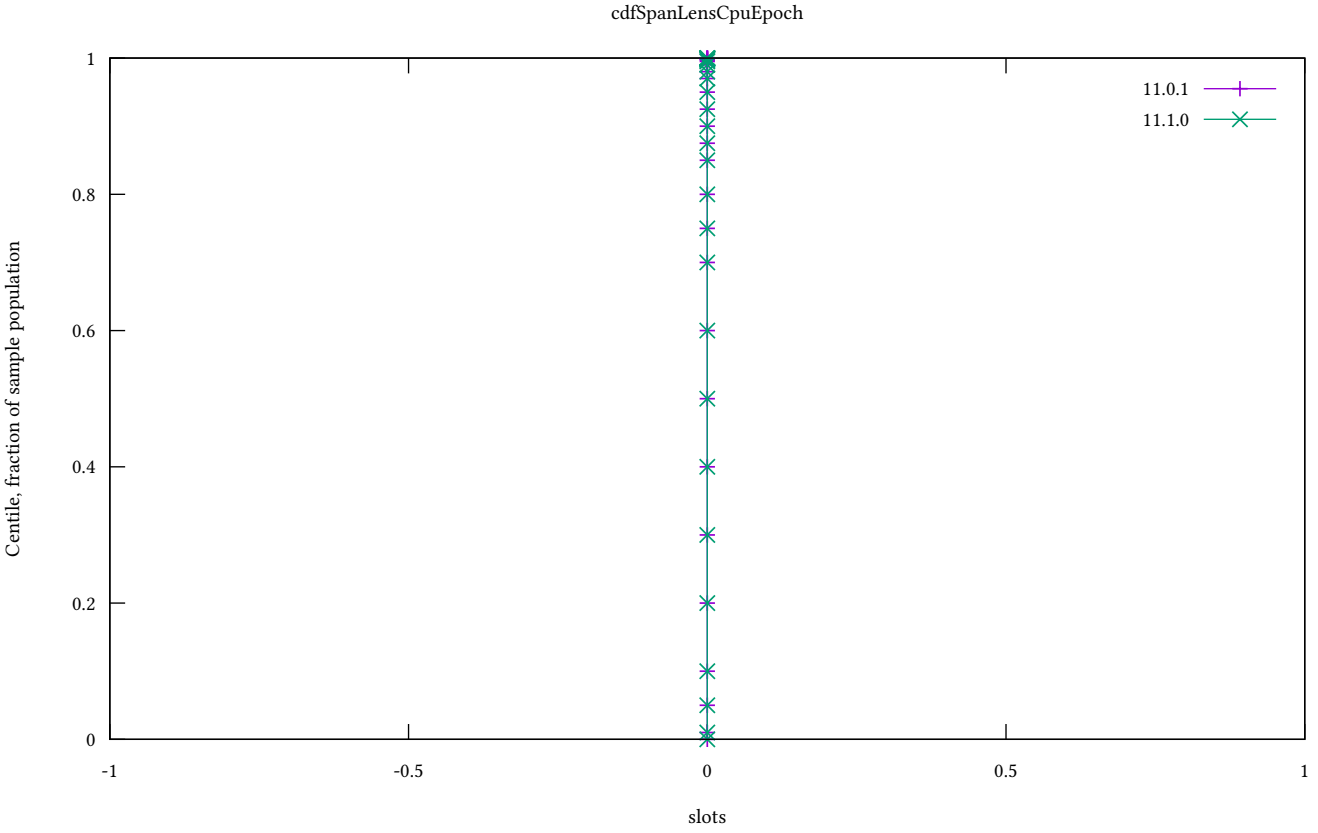

Figure 16: CPU spans at Ep boundary

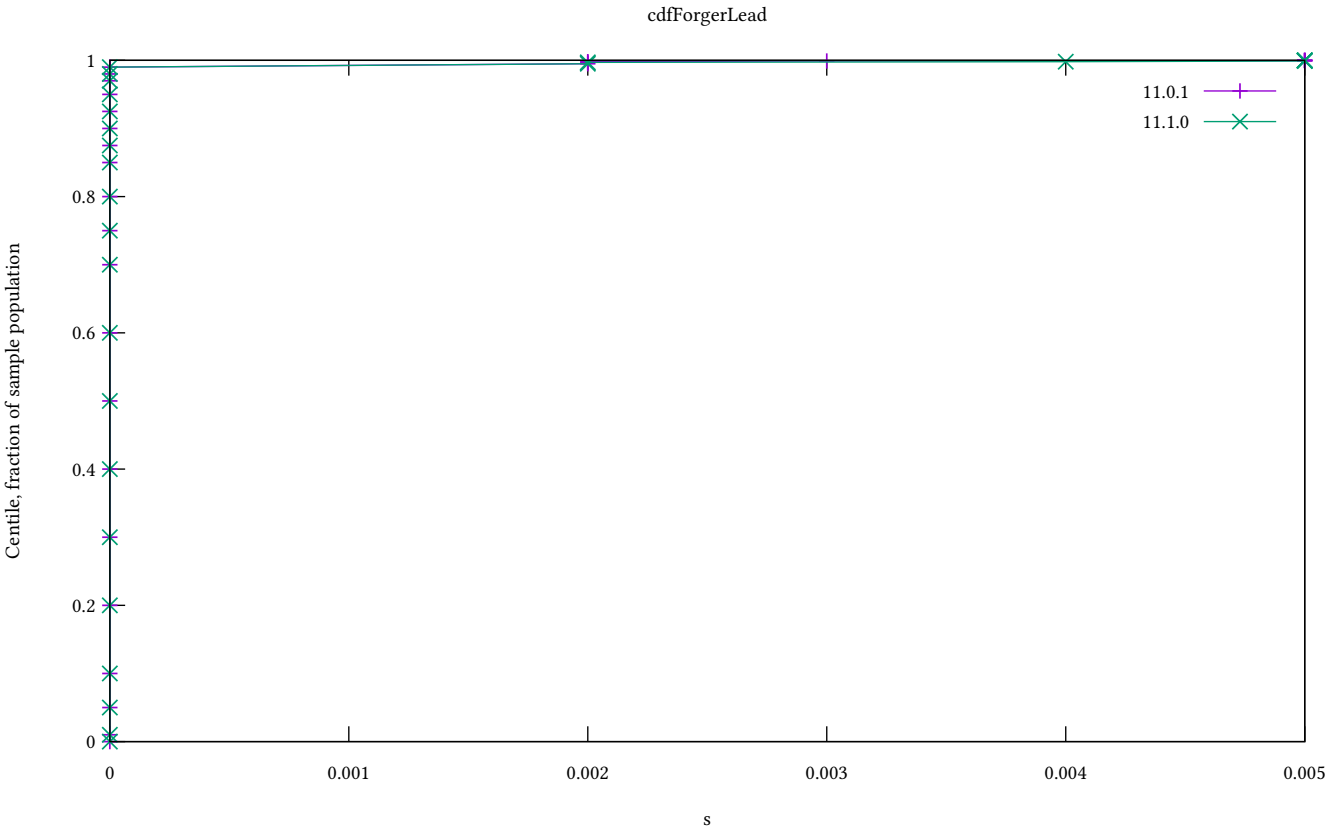

Figure 17: Leadership check duration

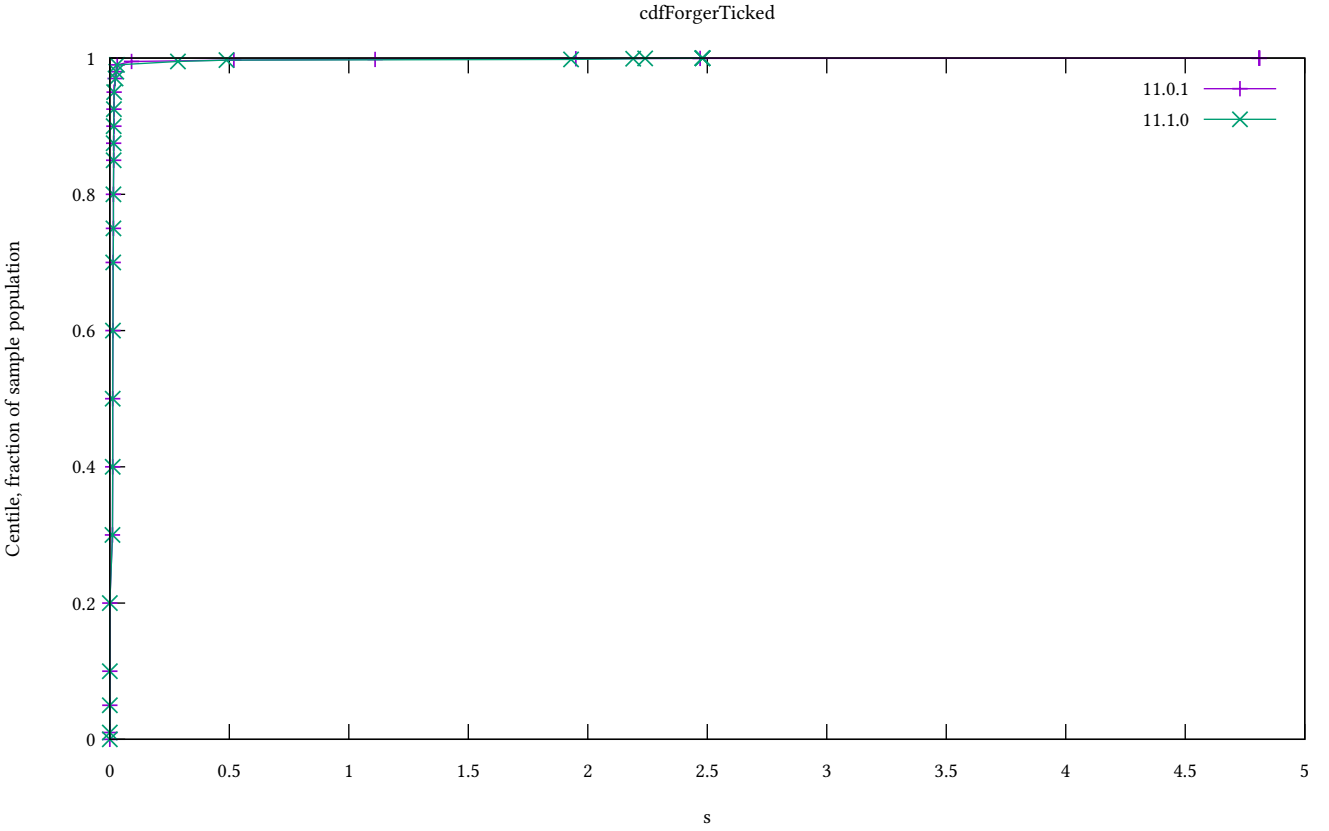

Figure 18: Ledger ticking

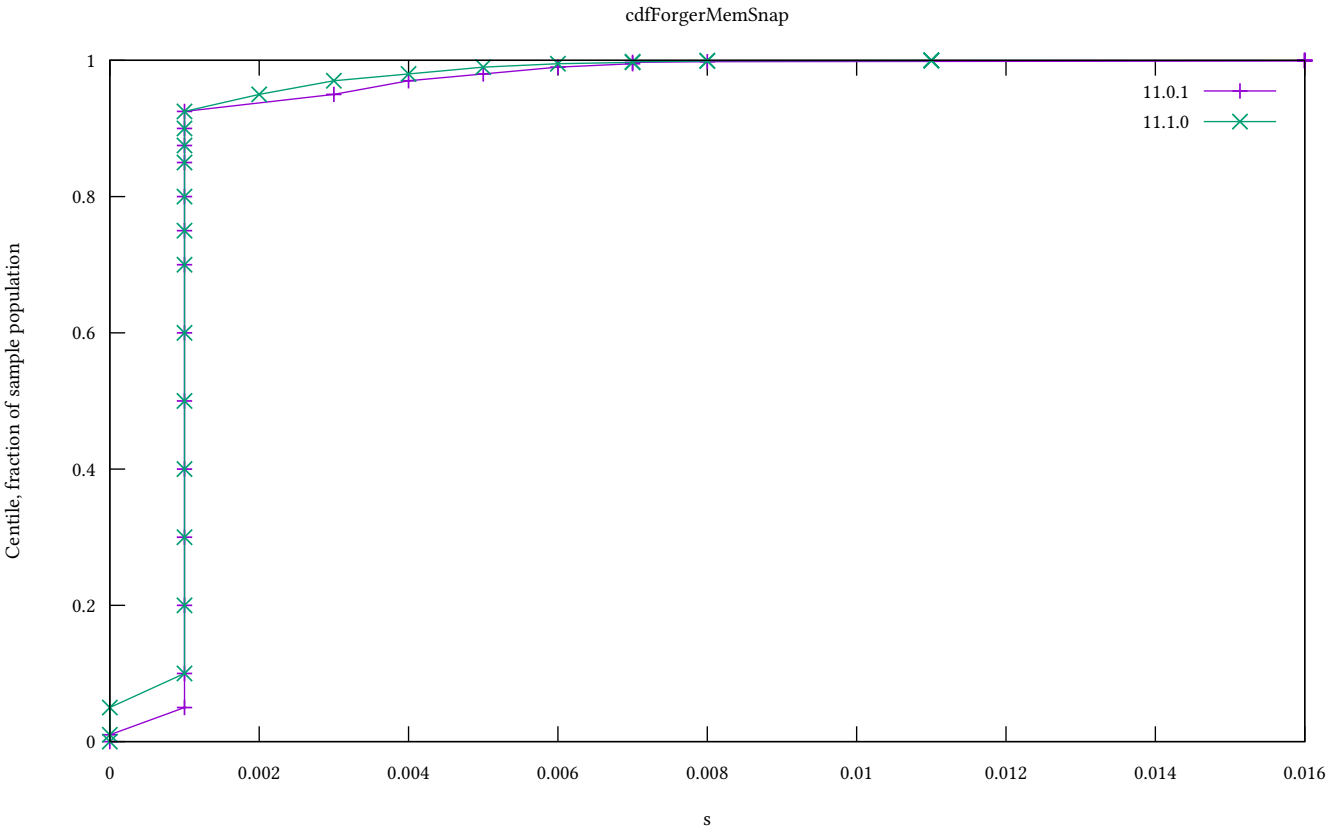

Figure 19: Mempool snapshotting

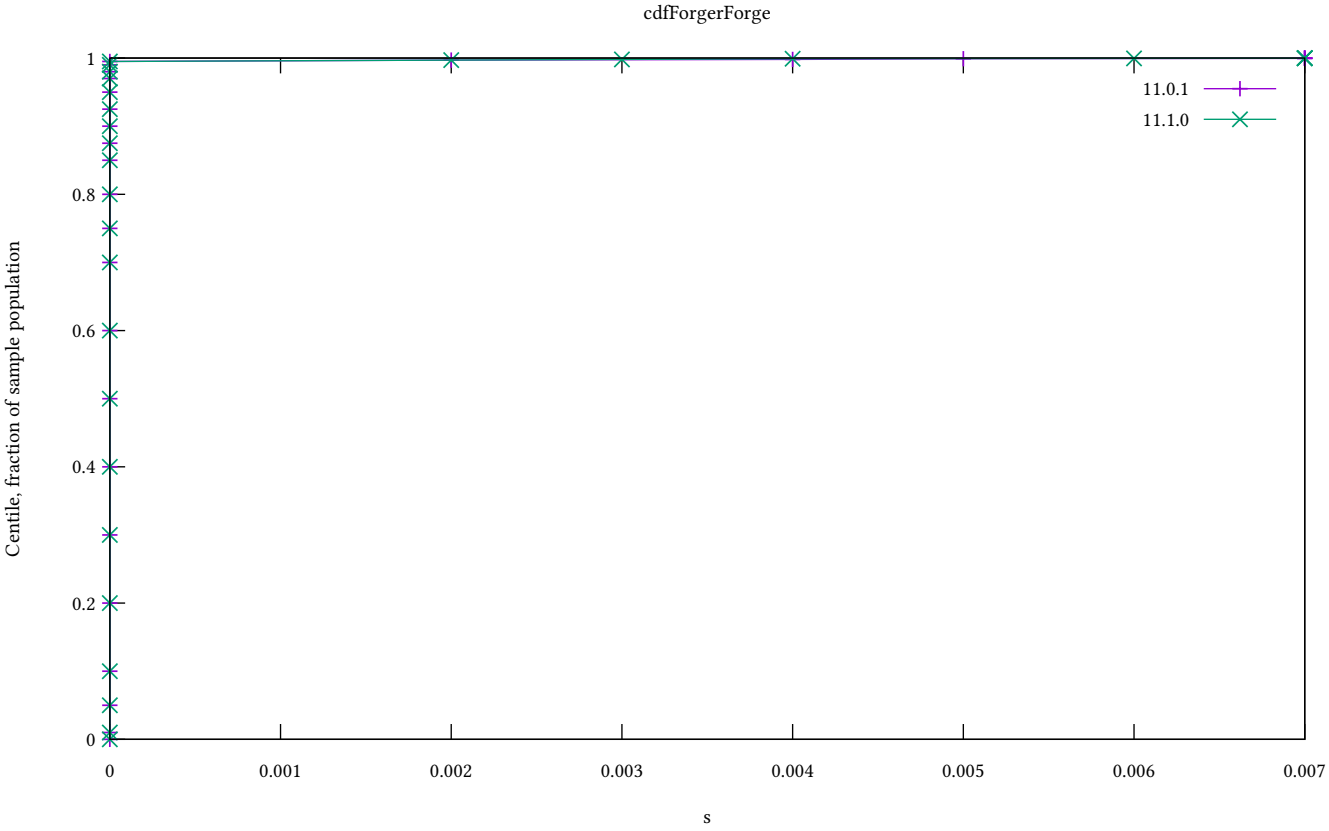

Figure 20: Leadership to forged

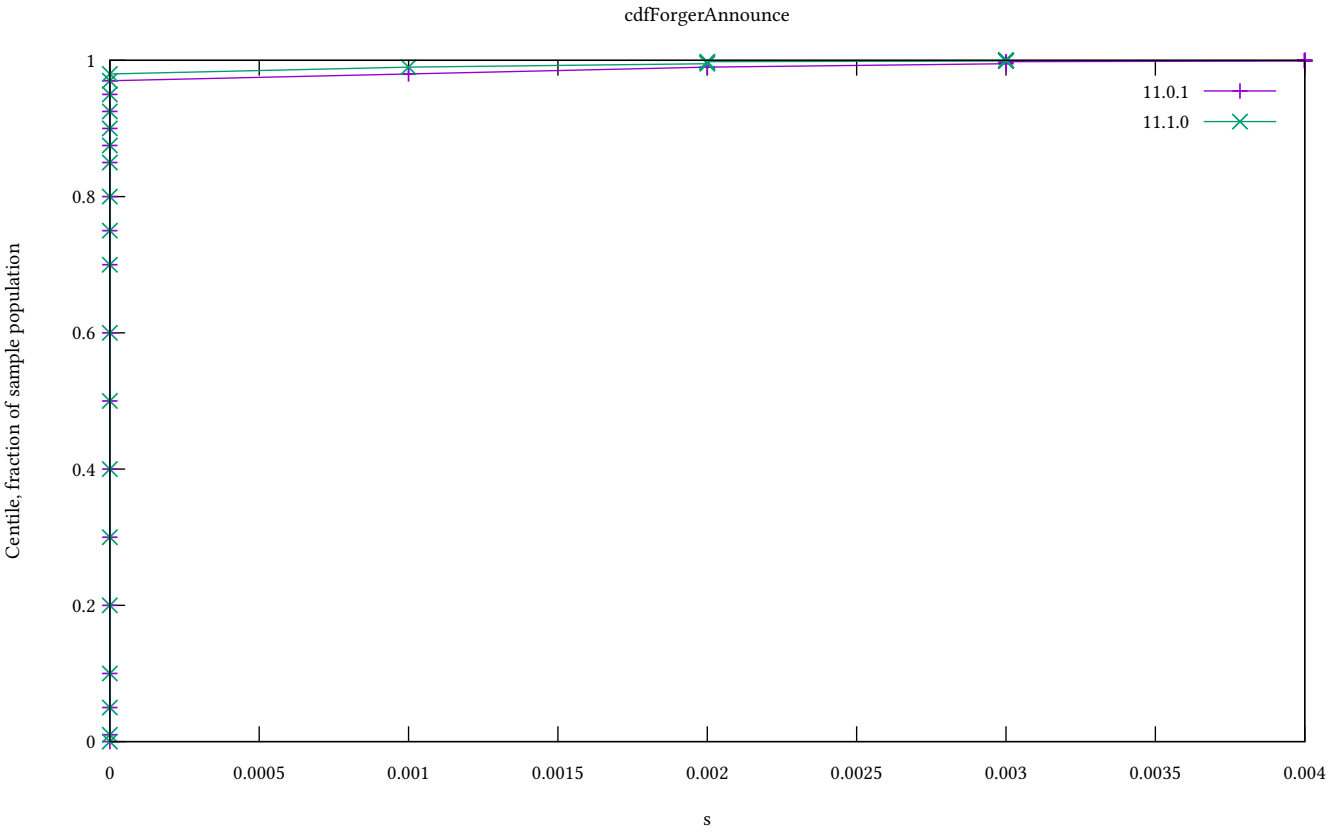

Figure 21: Forged to announced

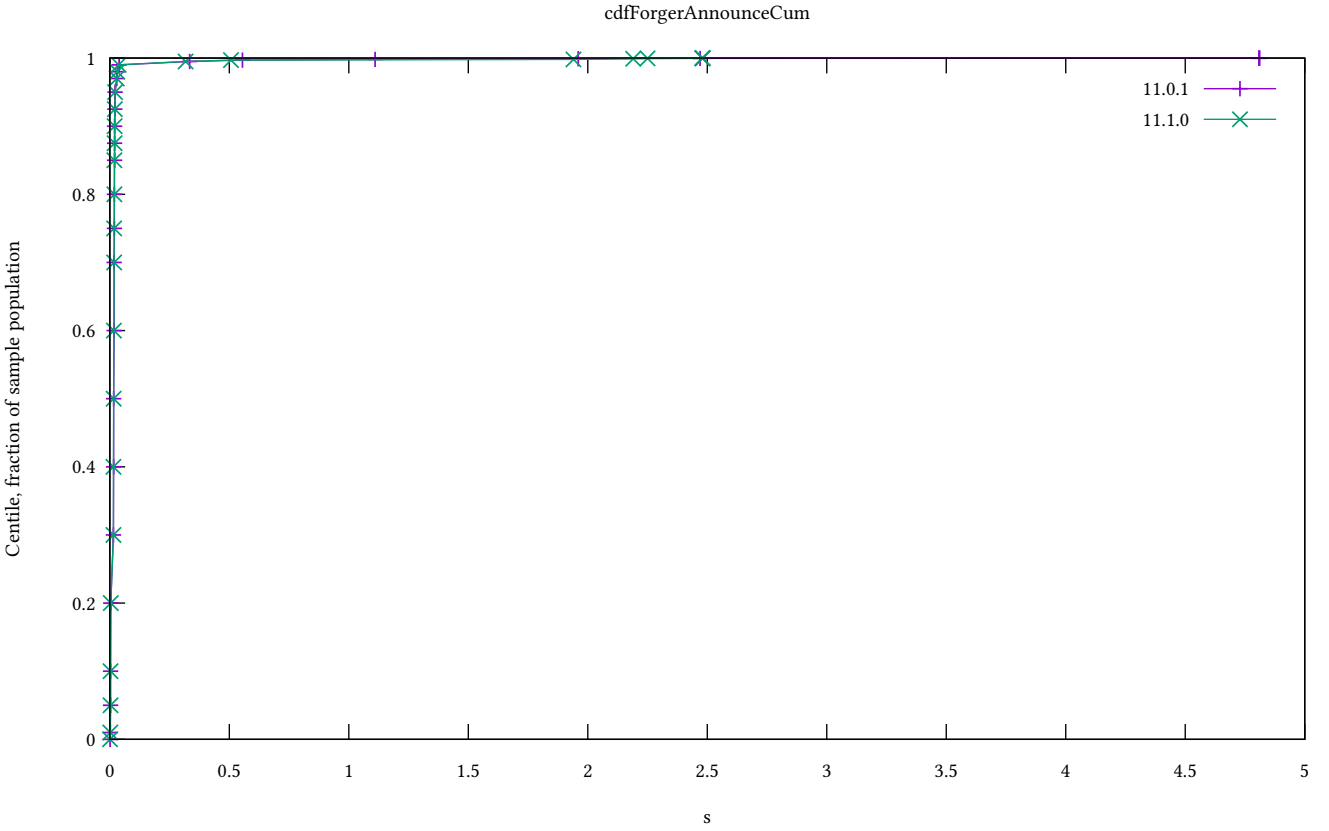

Figure 22: Slot start to announced

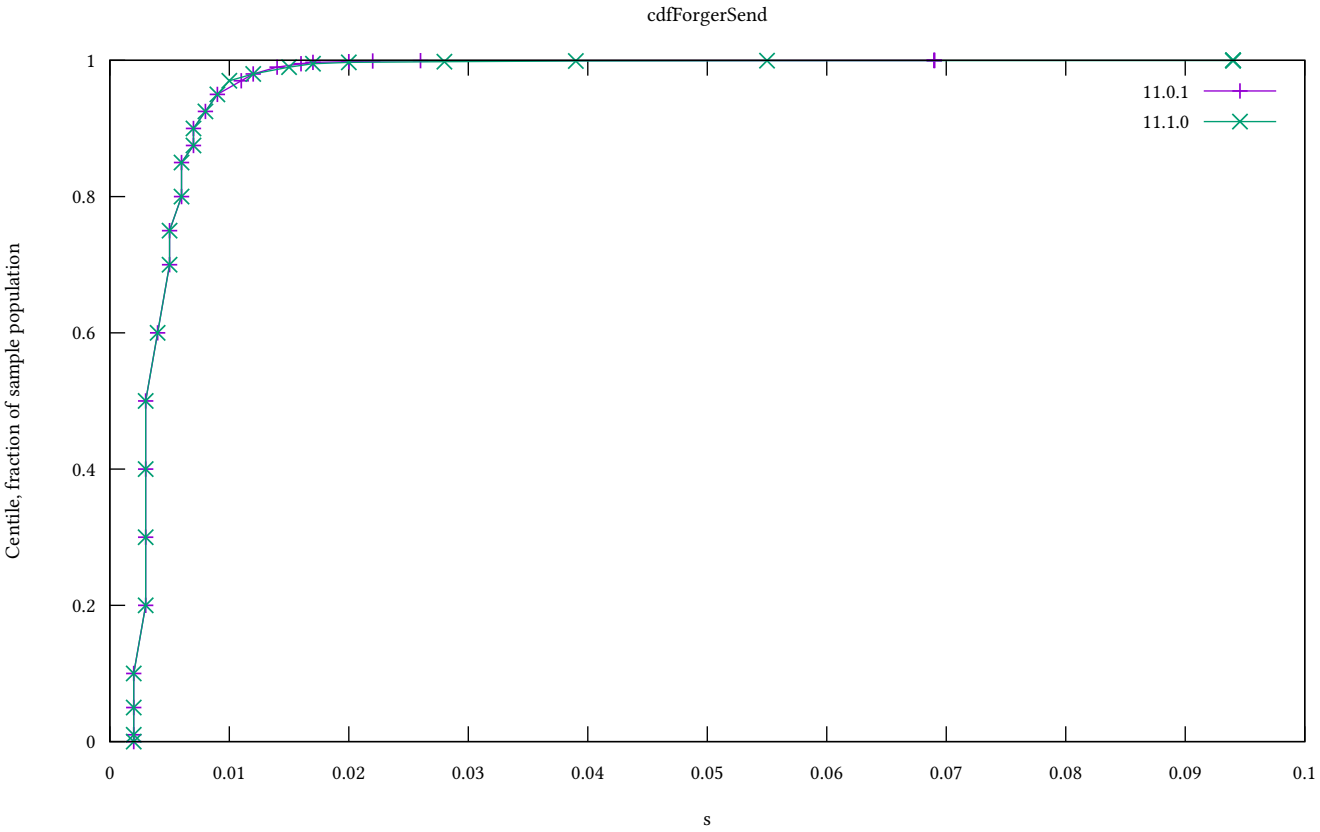

Figure 23: Forged to sending

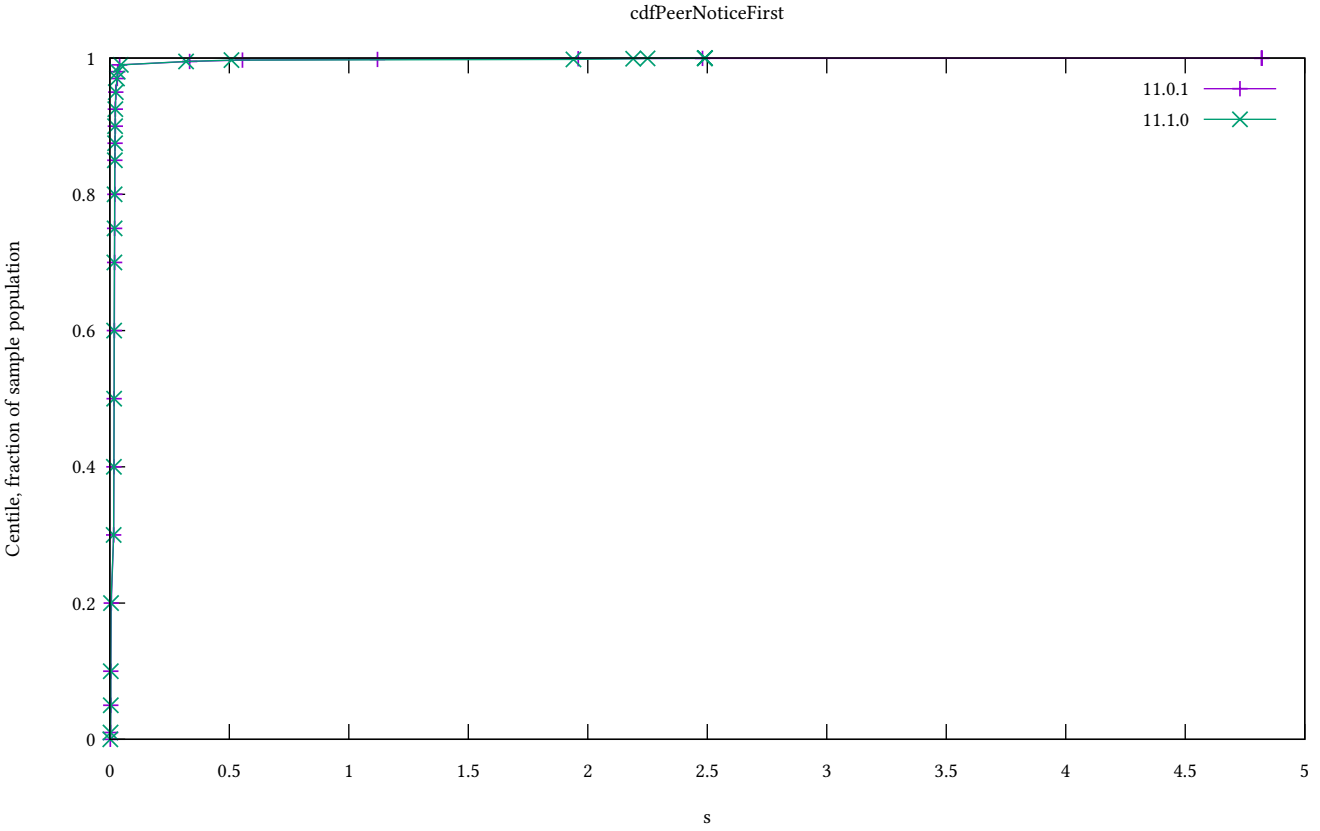

Figure 24: First peer notice

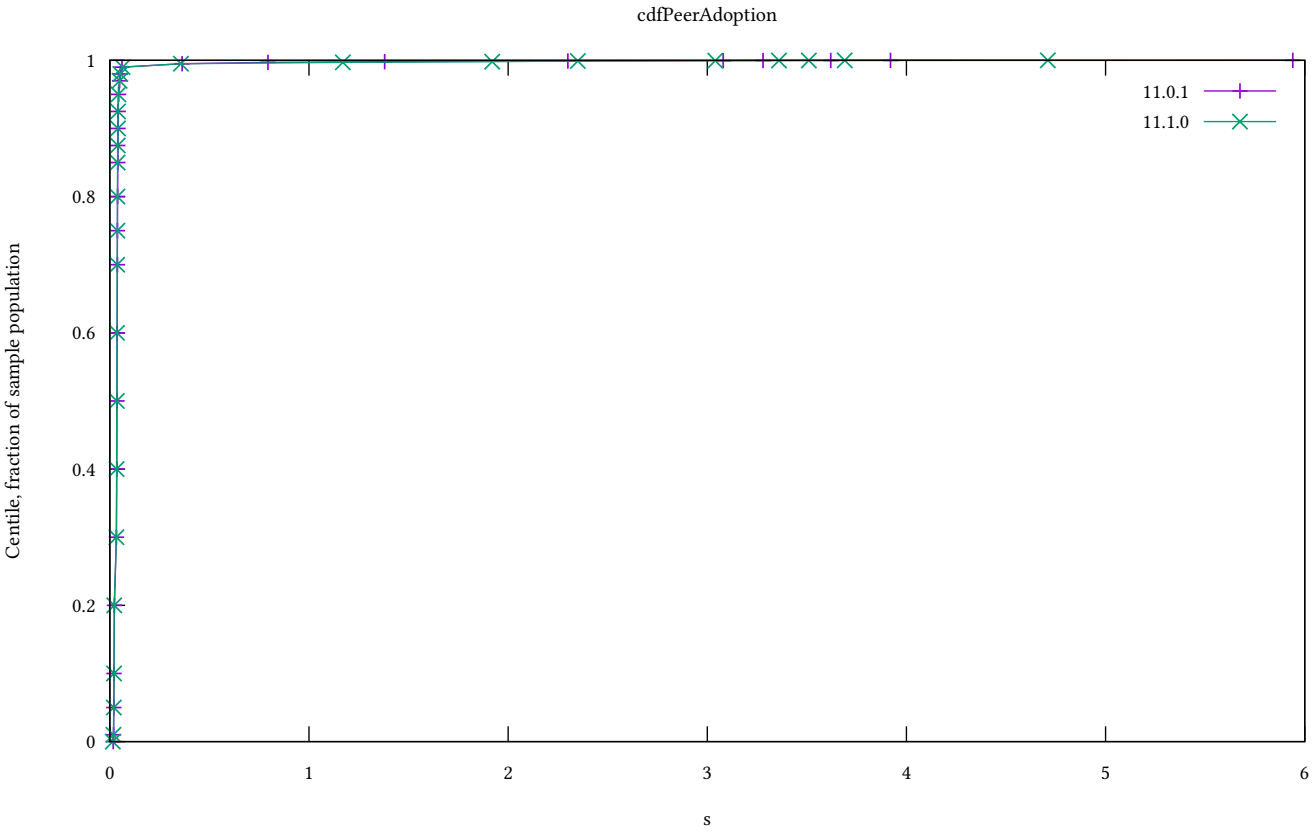

Figure 25: Fetched to adopted

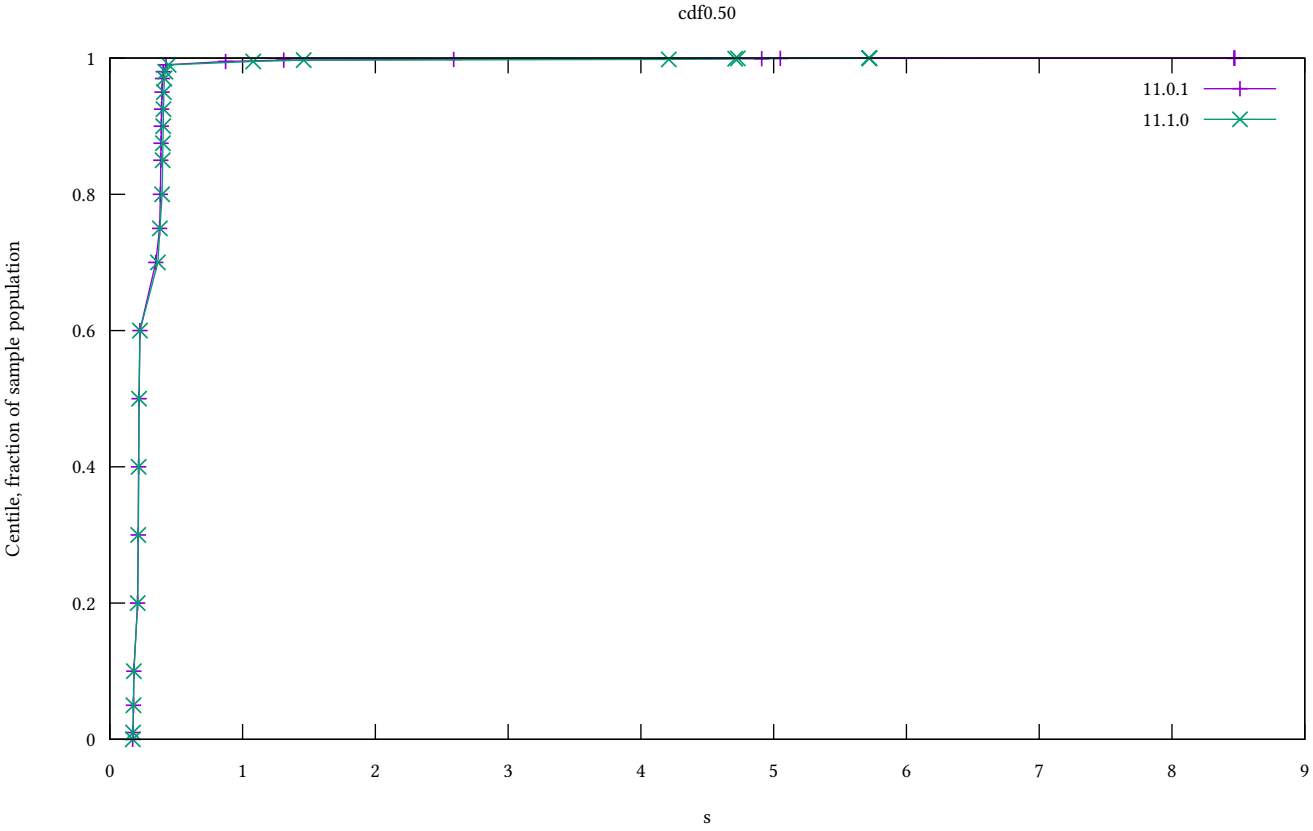

Figure 26: 0.50 adoption

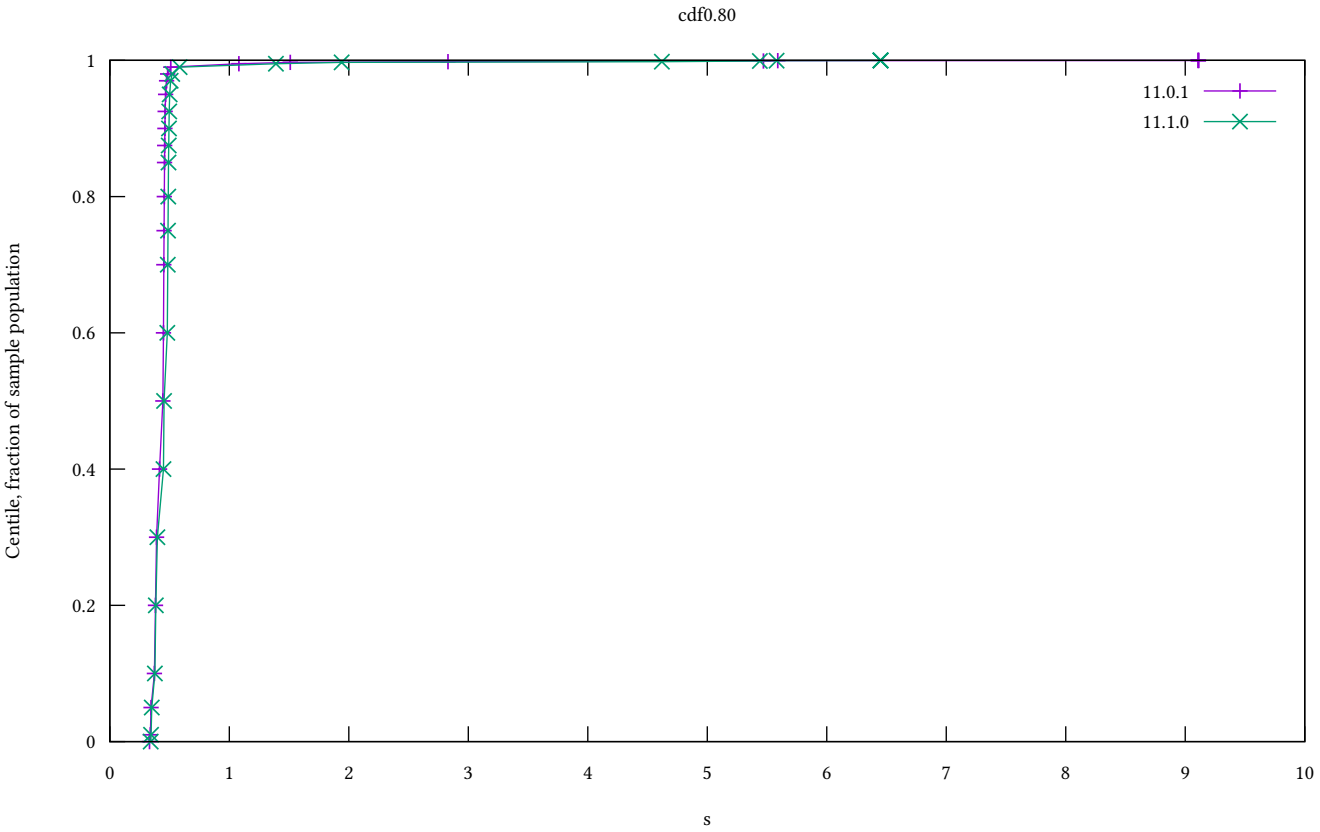

Figure 27: 0.80 adoption

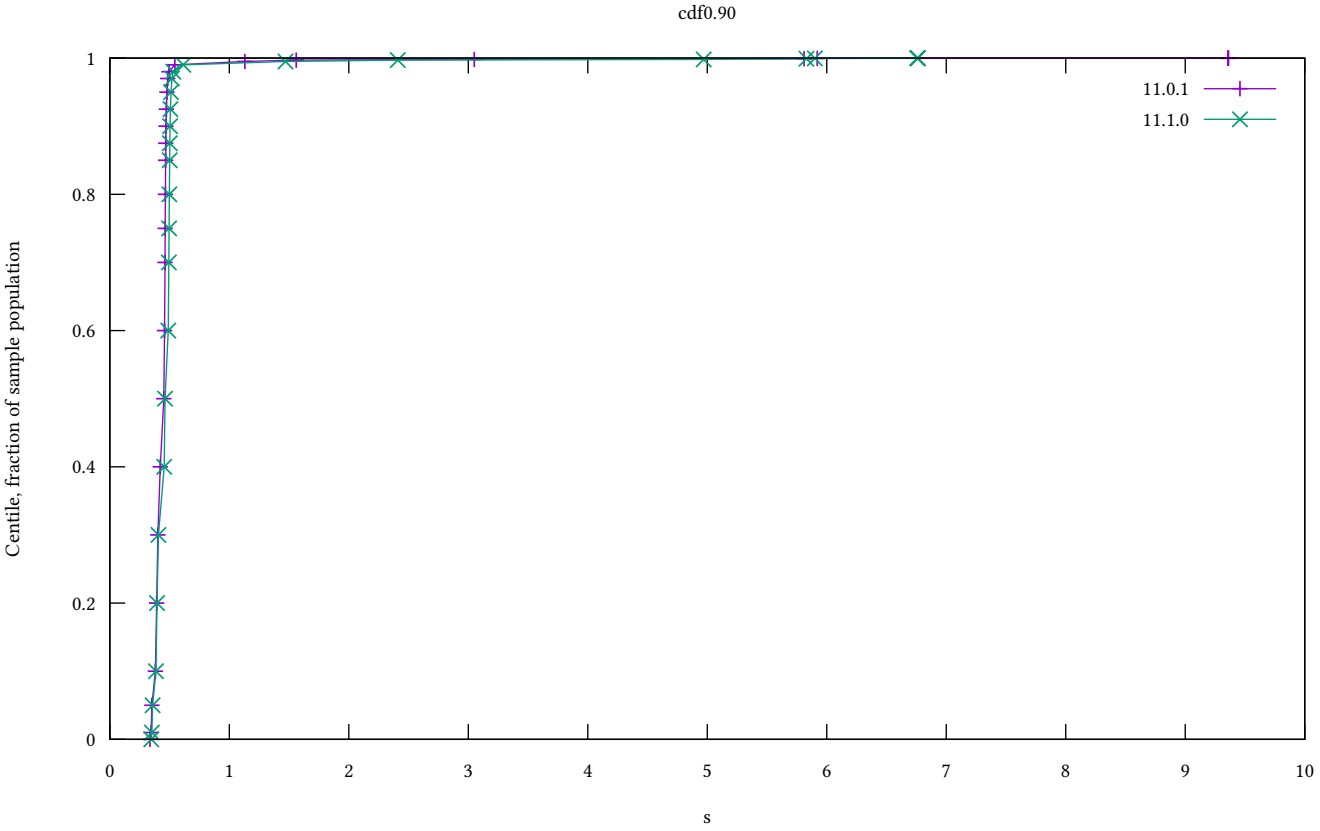

Figure 28: 0.90 adoption

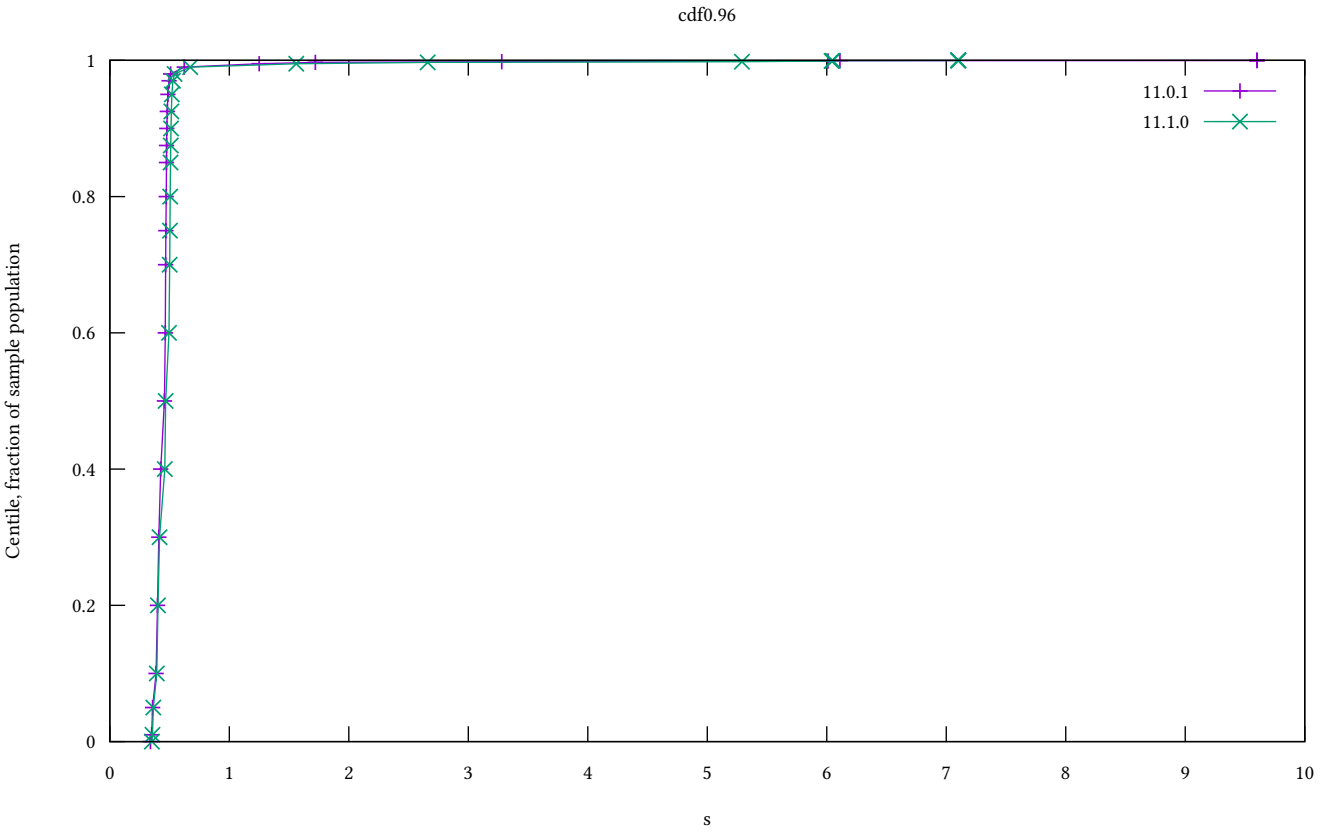

Figure 29: 0.96 adoption

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