11.2.1 against 11.1.1

Plutus countdown loop workload

Michael Karg, Cardano Performance & Tracing
2026-11-12

Contents

Manifest 2
Analysis 3
- Resource Usage 3
- Anomaly control 3
- Forging 4
- Individual peer propagation 4
- End-to-end propagation 4
- Ledger Snapshots 5
- Transaction Processing 5

Addendum: Plutus Workload Calibration 6
Appendix A: charts 7
- Cluster performance charts 7

Appendix B: data dictionary 22
- Block propagation metrics 22
- Cluster performance metrics 23

Manifest

We compare `11.1.1` (Conway) and `11.2.1` (Conway) relative to `11.1.1` (Conway), under Plutus countdown loop workload.
NB: LO stands for log object, i.e. a trace message emitted by a node.

	11.1.1	11.2.1
Analysis date	2026-11-10	2026-11-12
Cluster system start date	2026-09-07	2026-11-11
Cluster system start time	12:34:55	12:52:03
Identifier	11.1.1	11.2.1
Run batch	11.1.1	11.2.1
GHC version	9.6.7	9.6.7
cardano-node	11.1.1 @ 07d5a1c	11.2.1 @ eee0e14
ouroboros-consensus	4.2.1.0 @ 82ecba3	5.4.0.0 @ 808e400
ouroboros-network	1.2.0.0 @ c45735a	2.0.0.0 @ 4646039
cardano-ledger-core	1.21.0.0 @ f649f97	1.23.0.0 @ 95b3690
plutus-core	1.65.0.0 @ b2db512	1.71.0.0 @ 261766a
cardano-crypto	1.3.0 @ a741501	1.4.0 @ ac2e12a
cardano-prelude	0.2.2.0 @ 2b6092c	0.2.2.0 @ 2b6092c
Era	conway	conway
Delegation map size	1000000	1000000
Stuffed UTxO size	4000000	4000000
Final UTxO size	5061276	5061276
DRep count	10000	10000
Extra tx payload	100	100
Tx inputs	1	1
Tx Outputs	1	1
TPS	0.850	0.850
Transaction count	61200	61200
Plutus script	Loop	Loop
Machines	52	52
Filters applied, count	4	4
LO emitted / host	899127.788	898825.981
LO analysed / host	588135.923	588289.115
Host run time, s	71945.885	71977.488
Host log line rate, Hz	12.497	12.488
LO analysed, total	30583068	30591034
Run time, s	71959	71986
Analysed run duration, s	56039	56062
Run time efficiency	0.779	0.779
Node start spread, s	22.190	19.012
Node stop spread, s	2.903	1.736
Slots analysed	56037	56060
Blocks analysed	2749	2704
Blocks rejected	873	894

Analysis

Resource Usage

	11.1.1	11.2.1	Δ	Δ%
Forge loop starts, units	0.99998	0.99998	0.0	0
Process CPU usage, %	1.1594	1.1545	−0.005	−0.4
RTS GC CPU usage, %	0.2227	0.2336	0.0109	5
RTS Mutator CPU usage, %	0.9375	0.9218	−0.0157	−2
Major GCs, events	0.0008	0.0009	0.0	0
Minor GCs, events	0.3367	0.3311	−0.0056	−2
Kernel RSS, MiB	7217.4	6611.2	−606.19	−8
RTS heap size, MiB	7147.2	6535.2	−611.99	−9
RTS live GC dataset, MiB	3575.4	2754.3	−821.13	−23
RTS alloc rate, MiB/s	9.959	9.7619	−0.197	−2
Filesystem reads, KiB/s	0.0	4.7e-5	0.0	0
Filesystem writes, KiB/s	208.91	208.95	0.0395	0.0
CPU 85% peaks, ep. trans., slots	1.6385	1.7009	0.0624	4
CPU 85% peaks, other, slots	1.3118	1.326	0.0142	1

Anomaly control

	11.1.1	11.2.1	Δ	Δ%
Blocks per host, blocks	71.154	71.058	−0.0962	−0.1
Filtered to chained blocks, :	0.7585	0.7514	−0.0071	−0.9
Chained to forged blocks, :	0.9791	0.9737	−0.0054	−0.6
Slot battles, blocks	0.02242	0.02699	0.00457	20
Height battles, blocks	0.00071	0.00108	0.00037	52
Block size, Bytes	2996.0	2996.0	−0.003	−0.0

Forging

	11.1.1	11.2.1	Δ	Δ%
Started forge loop iteration, s	0.00093	0.00091	−0.0	0
Acquired block context, s	6.3e-5	5.8e-5	−0.0	0
Acquired ledger state, s	0.00011	0.00011	−0.0	0
Acquired ledger view, s	3.0e-5	3.1e-5	0.0	0
Leadership check duration, s	0.00042	0.00042	−0.0	0
Ledger ticking, s	0.01794	0.01839	0.00045	3
Mempool snapshotting, s	0.00147	0.00121	−0.00026	−18
Leadership to forged, s	0.00021	0.0002	−0.0	0
Forged to announced, s	0.00055	0.00052	−0.0	0
Forged to sending, s	0.00486	0.00475	−0.00011	−2
Forged to self-adopted, s	0.03993	0.03902	−0.00092	−2
Slot start to announced, s	0.02173	0.02186	0.00013	0.6

Individual peer propagation

	11.1.1	11.2.1	Δ	Δ%
First peer notice, s	0.02327	0.02333	0.0	0
First peer fetch, s	0.02884	0.02872	−0.00012	−0.4
Notice to fetch request, s	0.00128	0.00126	−0.0	0
Fetch duration, s	0.13242	0.12712	−0.00531	−4
Fetched to announced, s	0.00067	0.00064	−0.0	0
Fetched to sending, s	0.04187	0.04334	0.00147	4
Fetched to adopted, s	0.04116	0.04216	0.001	2

End-to-end propagation

	11.1.1	11.2.1	Δ	Δ%
0.50 adoption, s	0.27694	0.28729	0.01035	4
0.80 adoption, s	0.47793	0.44526	−0.03267	−7
0.90 adoption, s	0.4891	0.45499	−0.03412	−7
0.92 adoption, s	0.49247	0.45761	−0.03486	−7
0.94 adoption, s	0.49548	0.46039	−0.03509	−7
0.96 adoption, s	0.4993	0.46408	−0.03523	−7
0.98 adoption, s	0.50533	0.47037	−0.03496	−7
1.00 adoption, s	0.53179	0.483	−0.04879	−9

Ledger Snapshots

	11.1.1	11.2.1	Δ	Δ%
Taken, count	13.0	13.0	0.0	0
Duration, s	16.491	16.814	0.3229	2
Live after majGC, during, MiB	2276.4	1900.5	−375.87	−17
CPU 85% peaks, snapshot, slots	15.917	16.212	0.2944	2
Live after majGC, outside, MiB	1852.3	1903.8	51.525	3

Transaction Processing

	11.1.1	11.2.1	Δ	Δ%
Txns processed, count	1642.0	1647.1	5.0962	0.3
Txns accepted, count	797.35	793.35	−4.0	−0.5
Txns rejected, count	844.63	853.73	9.0962	1
Reject ratio, :	0.5141	0.518	0.0038	0.7
Mempool fill level, count	8.0	8.0	0.0	0
Proc CPU ÷ txns, ms	397.05	394.79	−2.2604	−0.6
GC CPU ÷ txns, ms	76.328	79.829	3.501	5
Mut CPU ÷ txns, ms	320.99	315.27	−5.718	−2

Addendum: Plutus Workload Calibration

	11.1.1	11.2.1
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

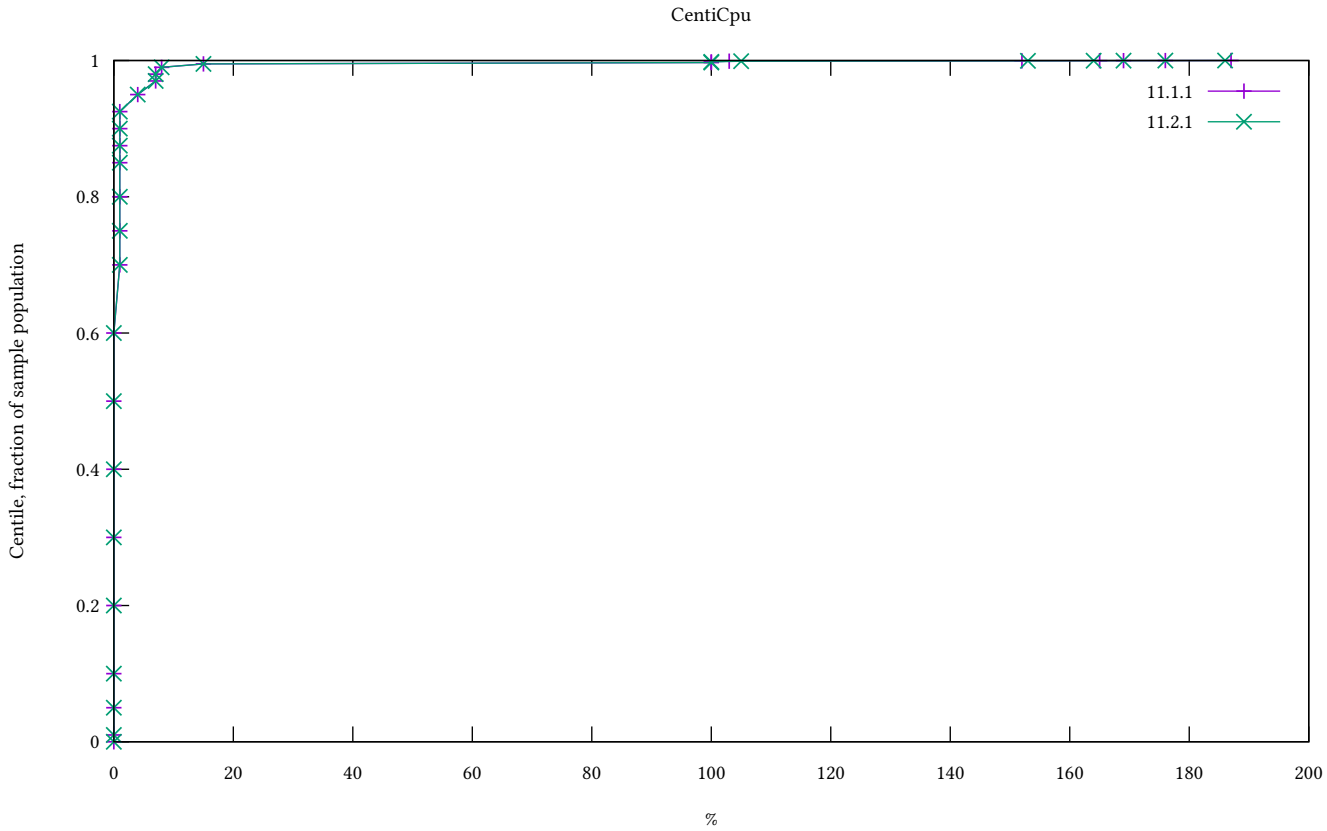


Figure 1: Process CPU usage

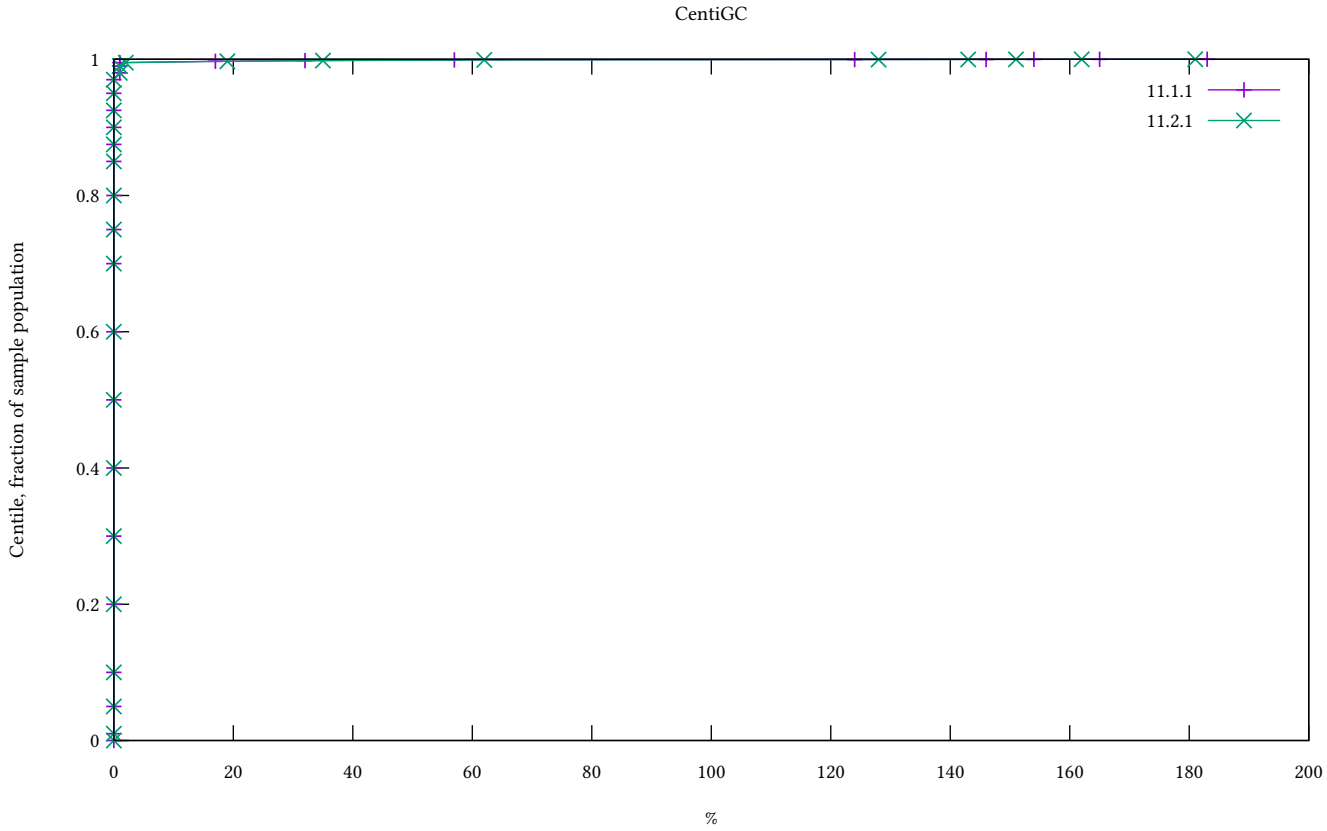


Figure 2: RTS GC CPU usage

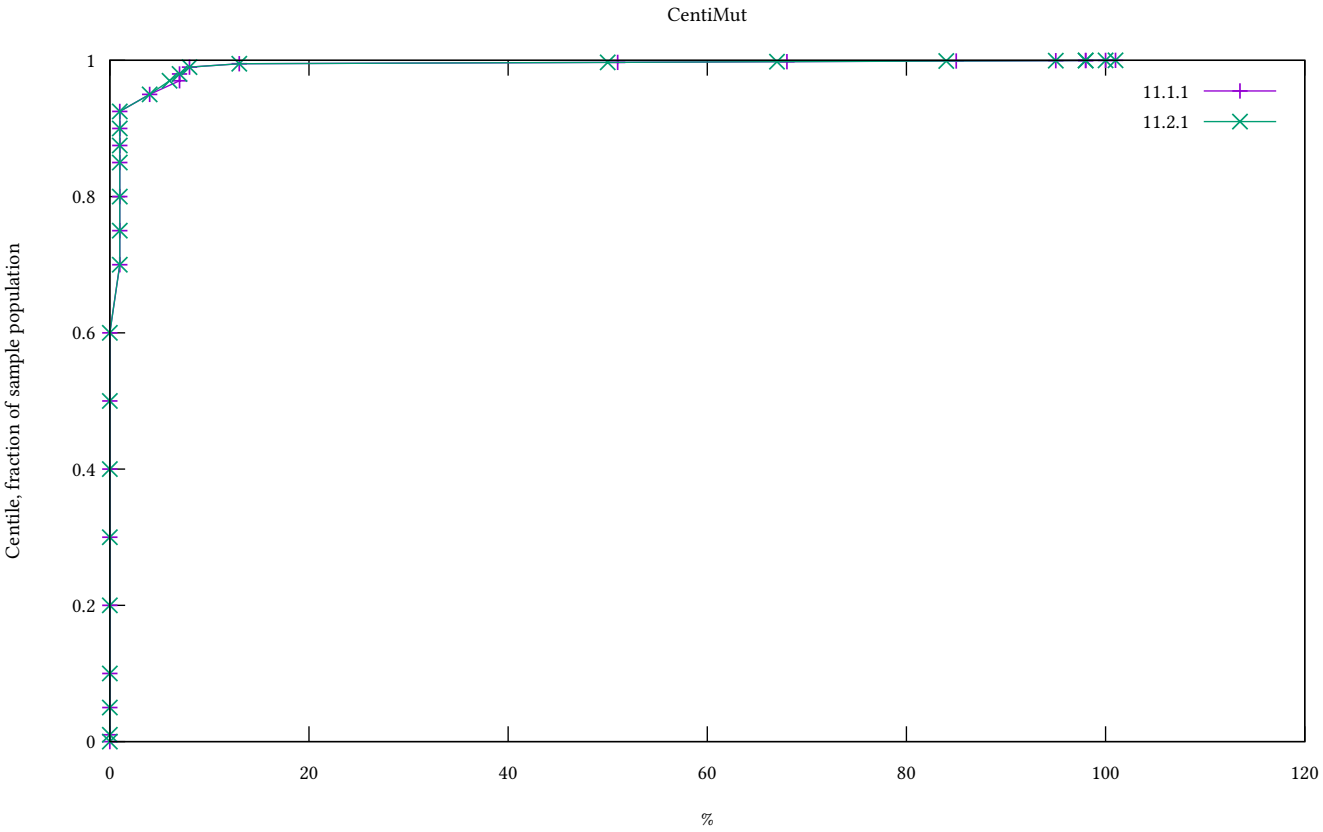


Figure 3: RTS Mutator CPU usage

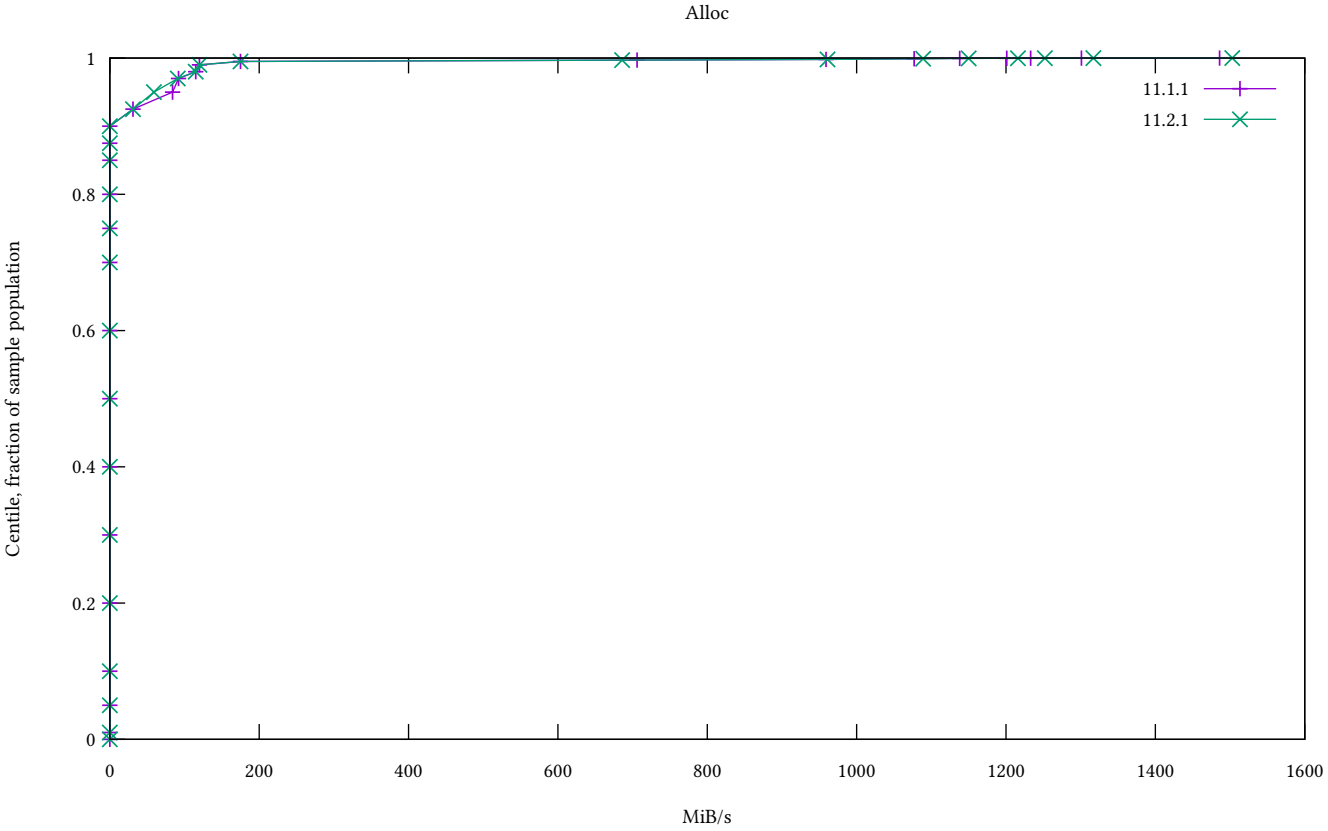


Figure 4: RTS alloc rate

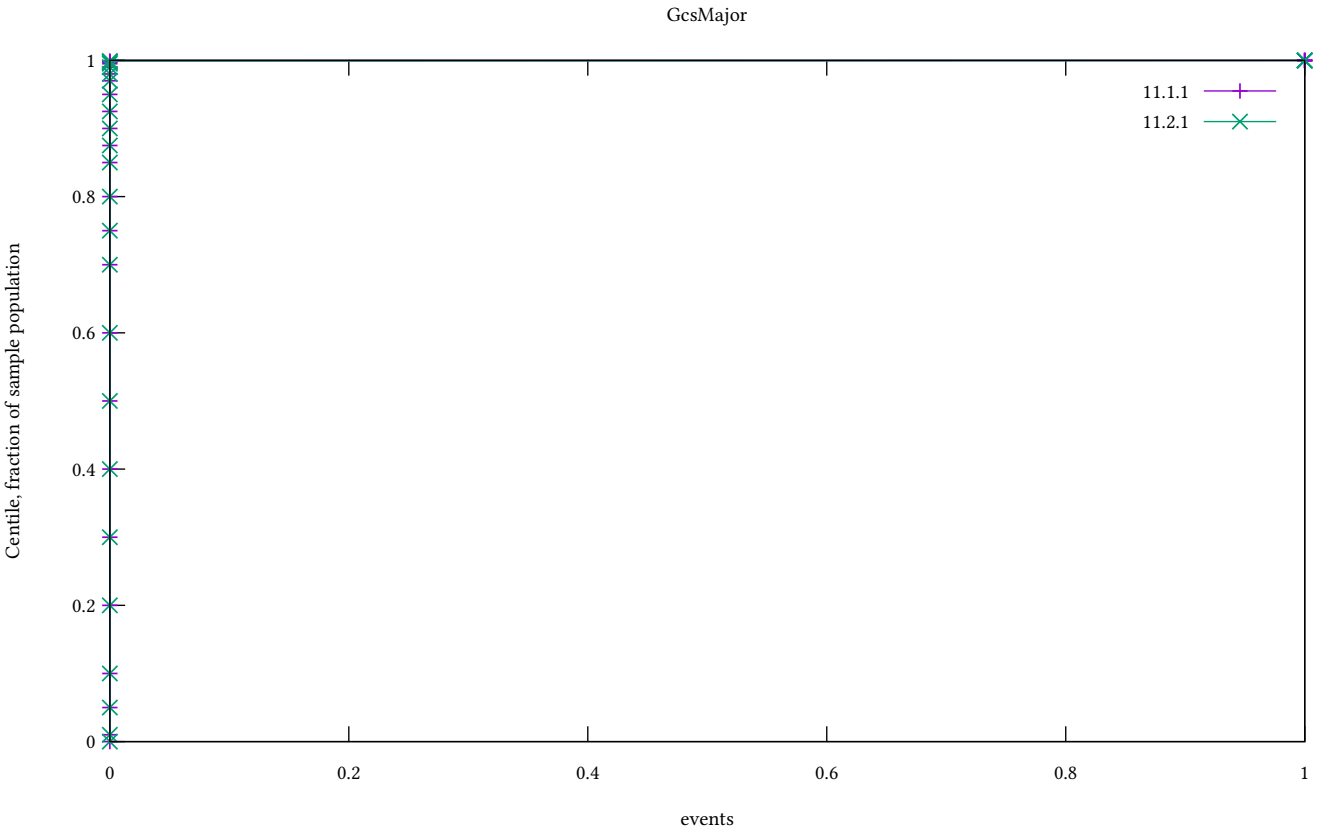


Figure 5: Major GCs

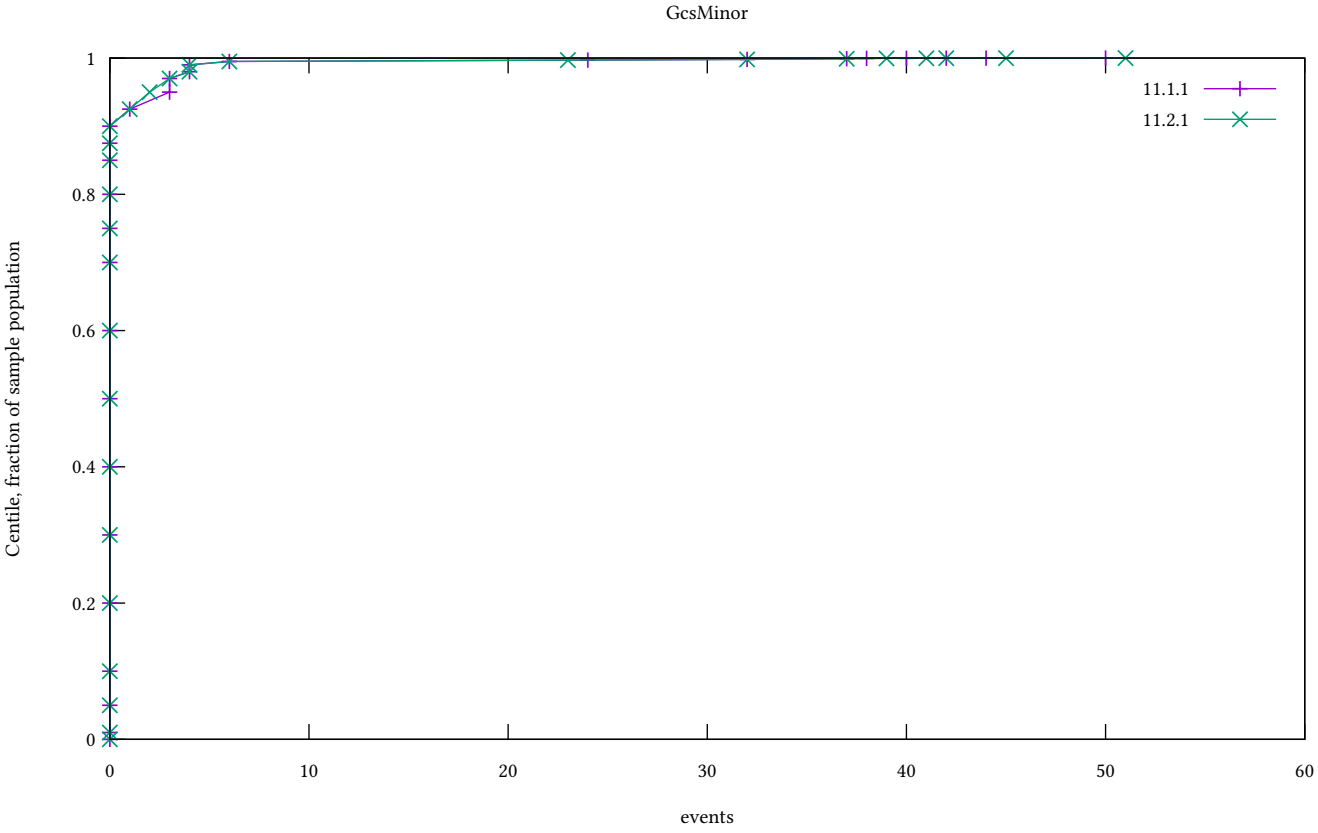


Figure 6: Minor GCs

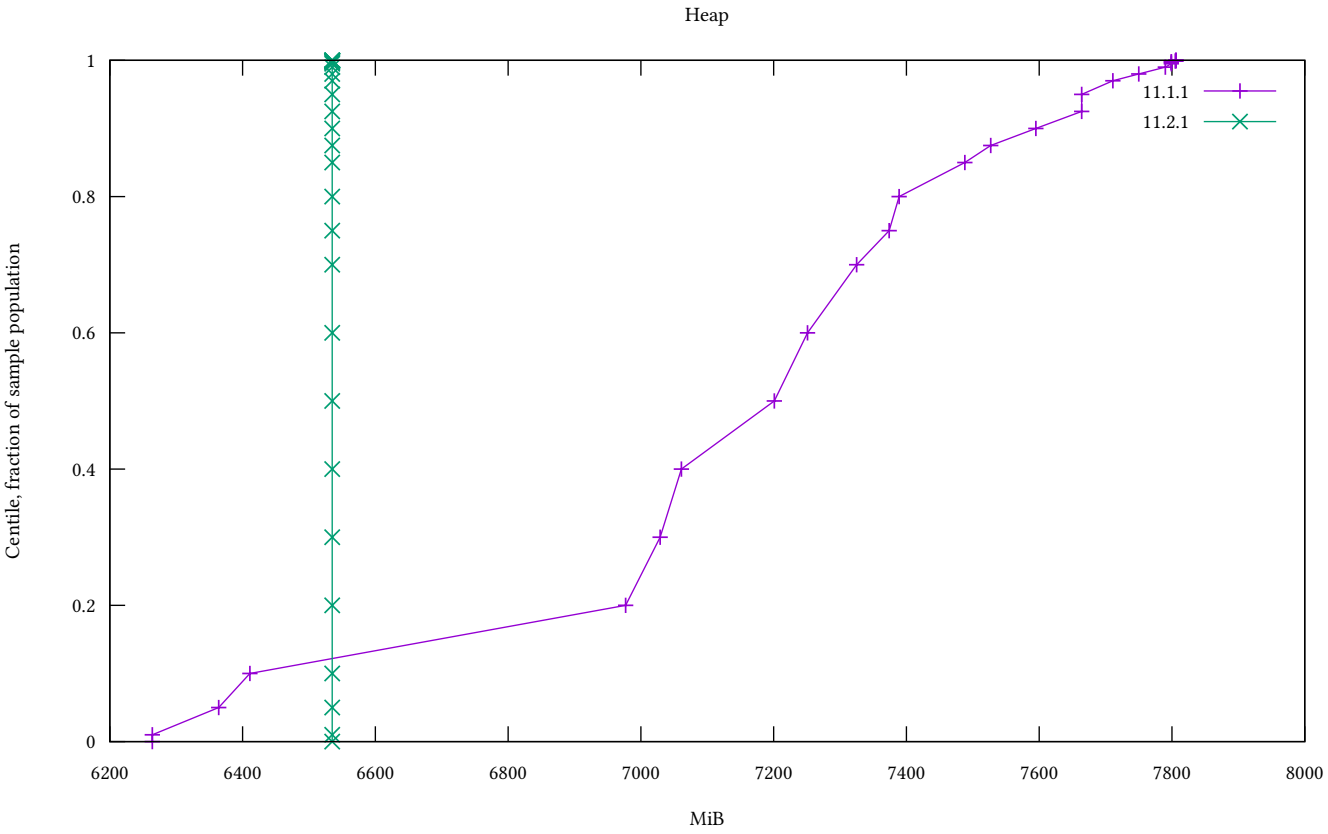


Figure 7: RTS heap size

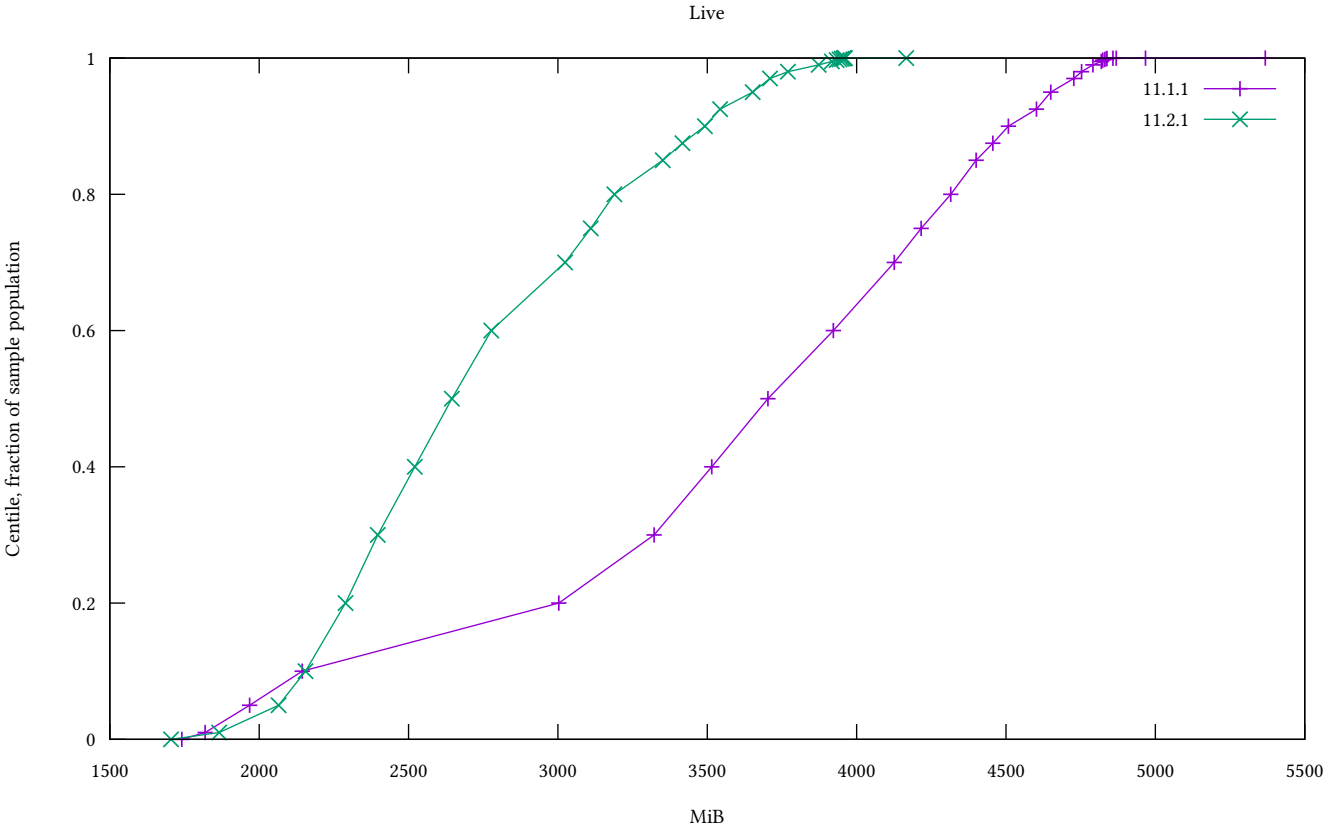


Figure 8: RTS live GC dataset

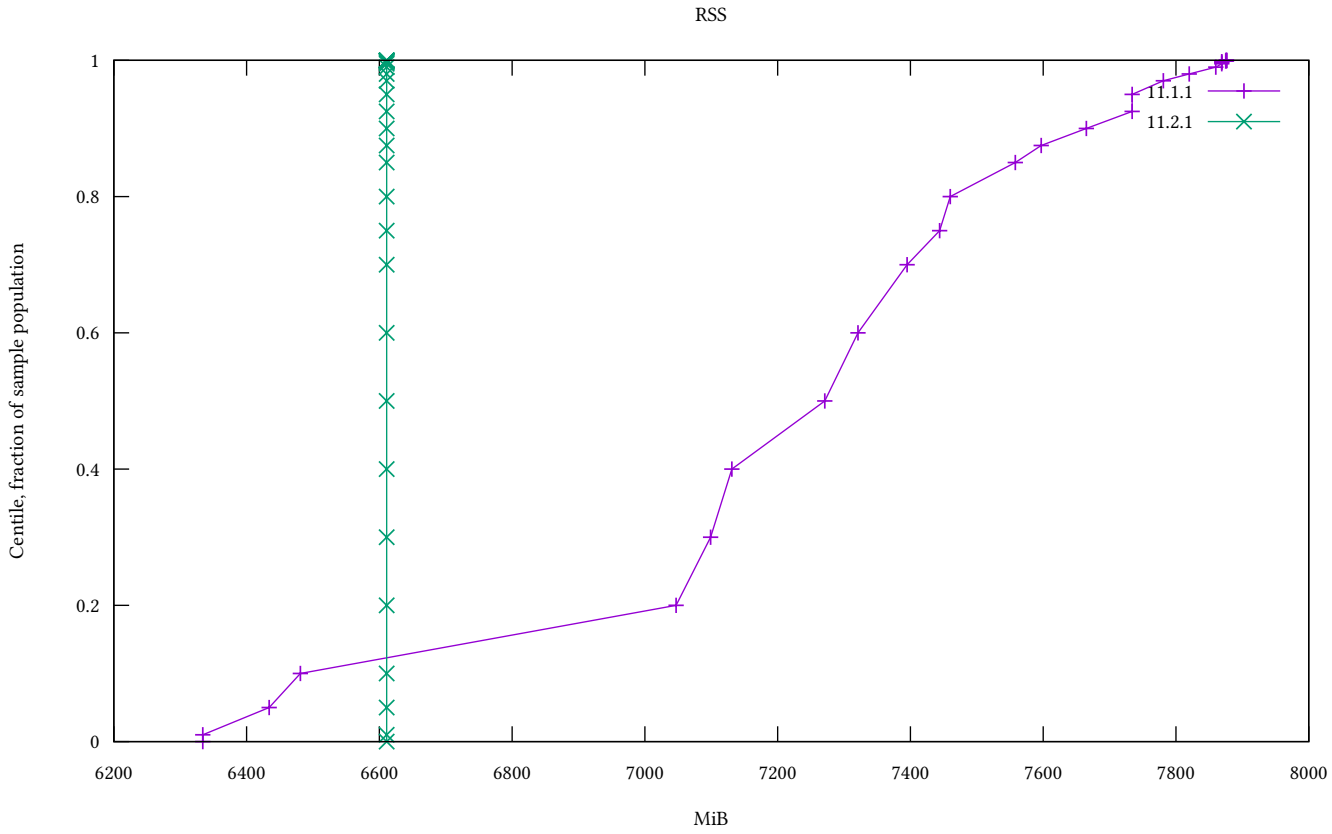


Figure 9: Kernel RSS

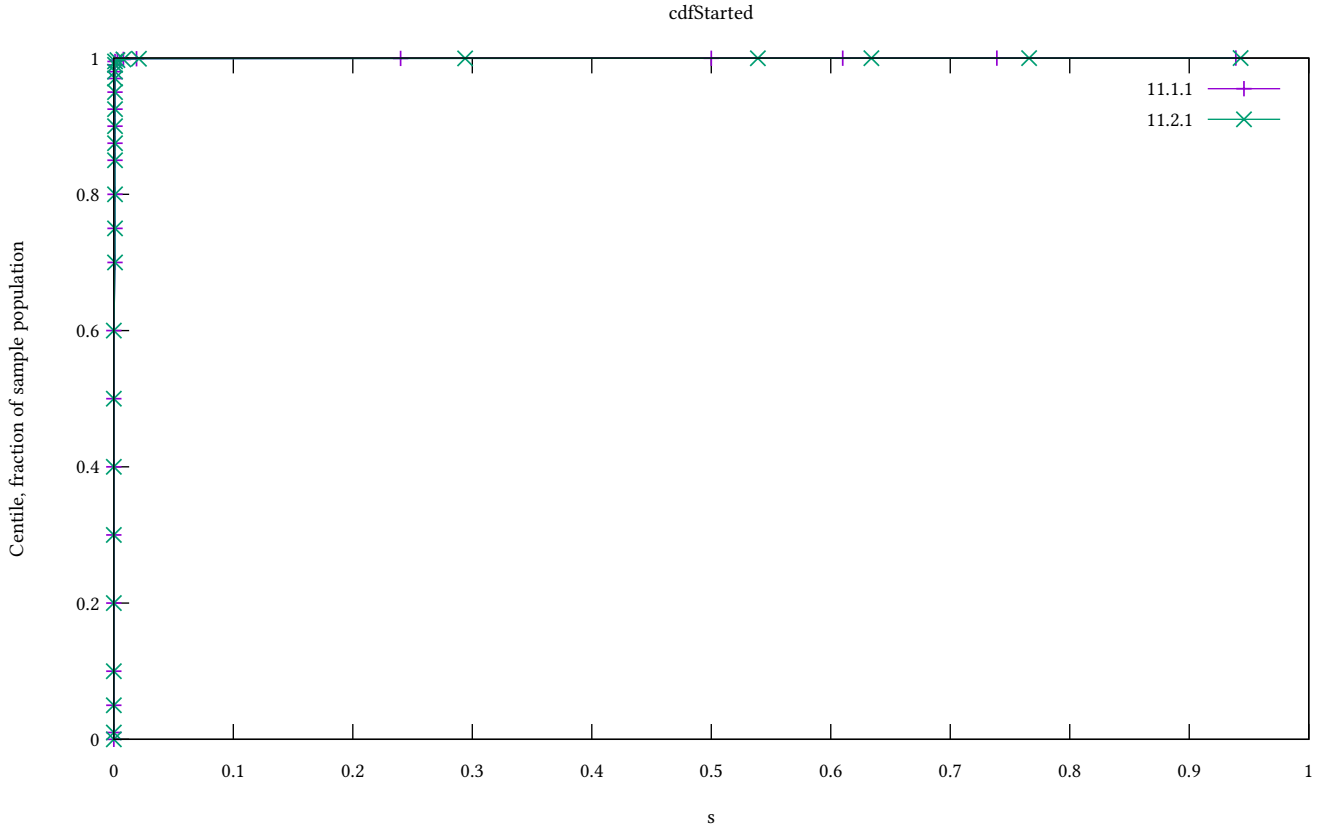


Figure 10: Forge loop tardiness

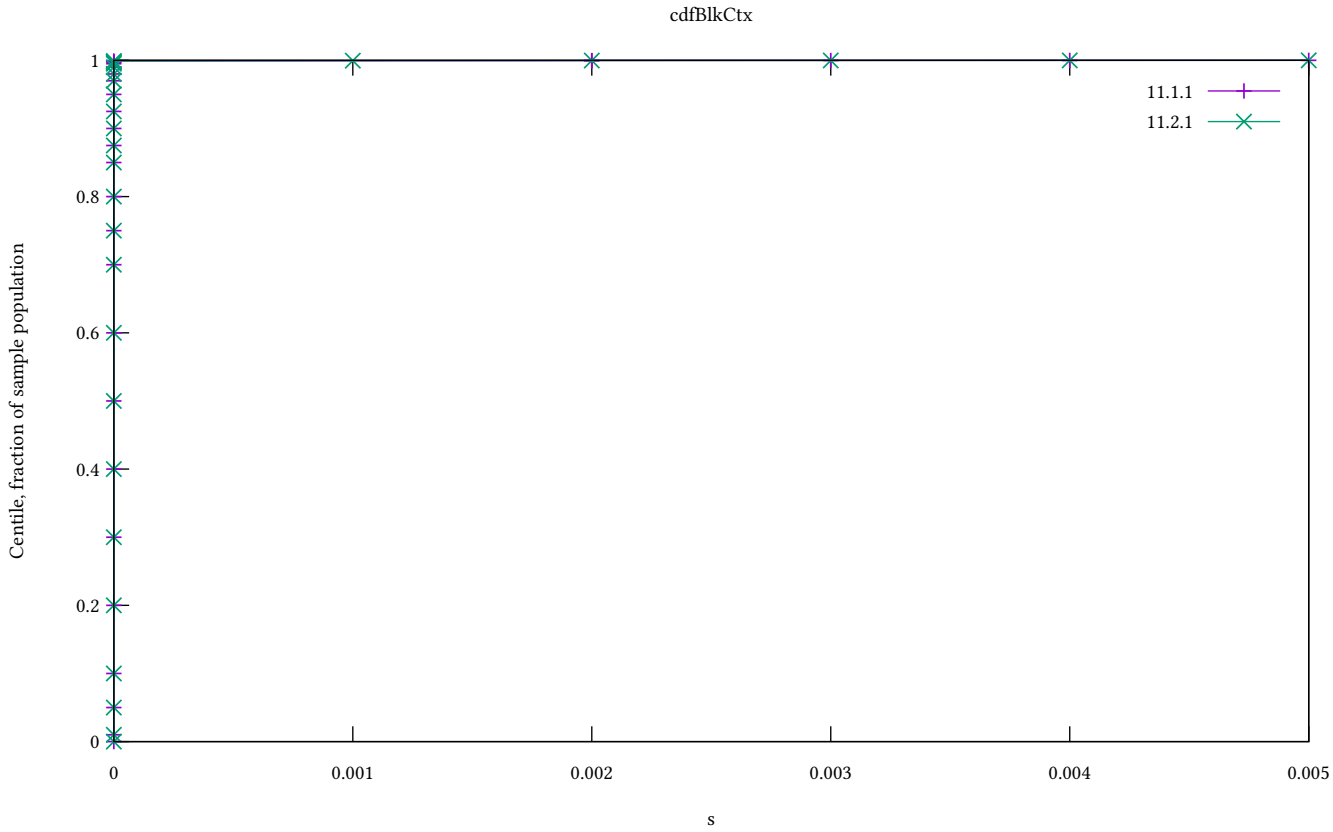


Figure 11: Block context acquisition delay

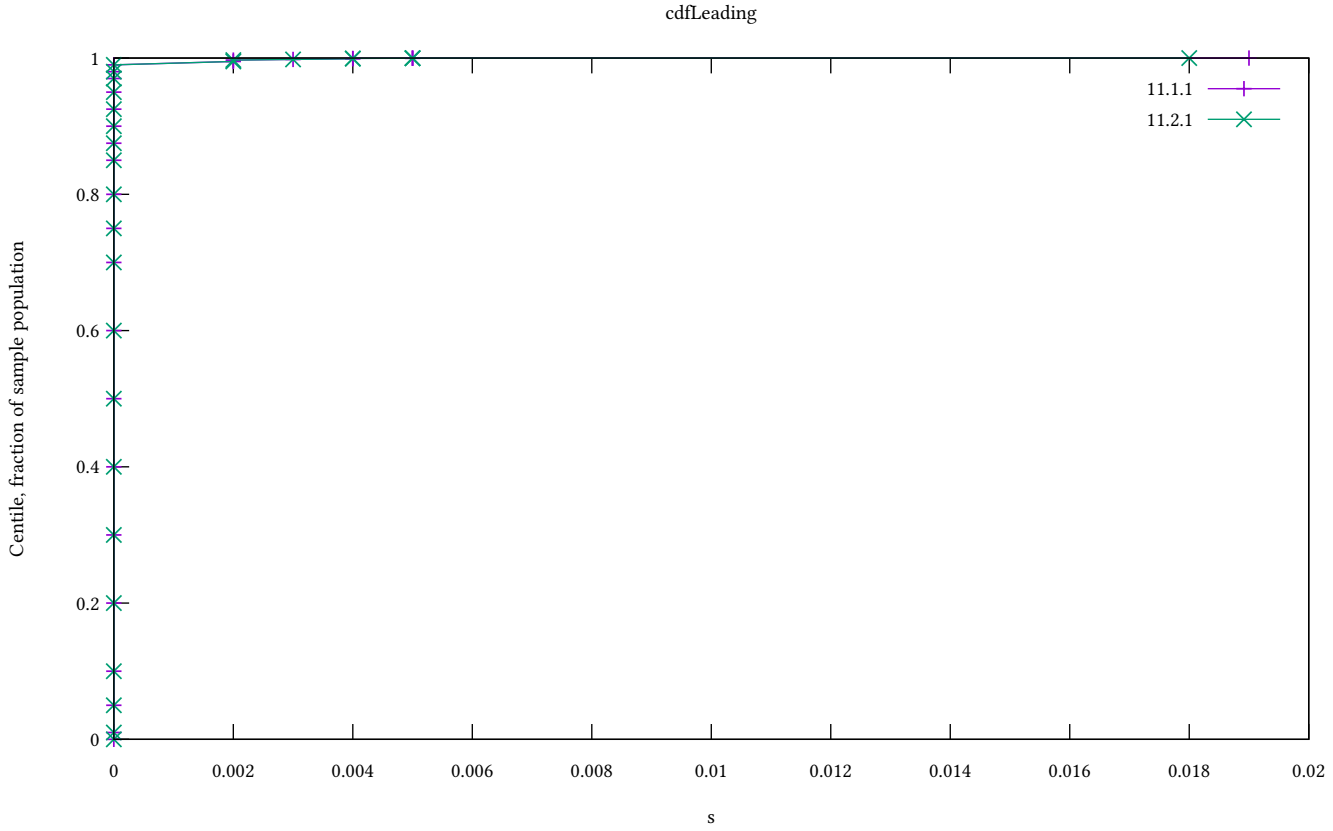


Figure 12: Leadership check duration

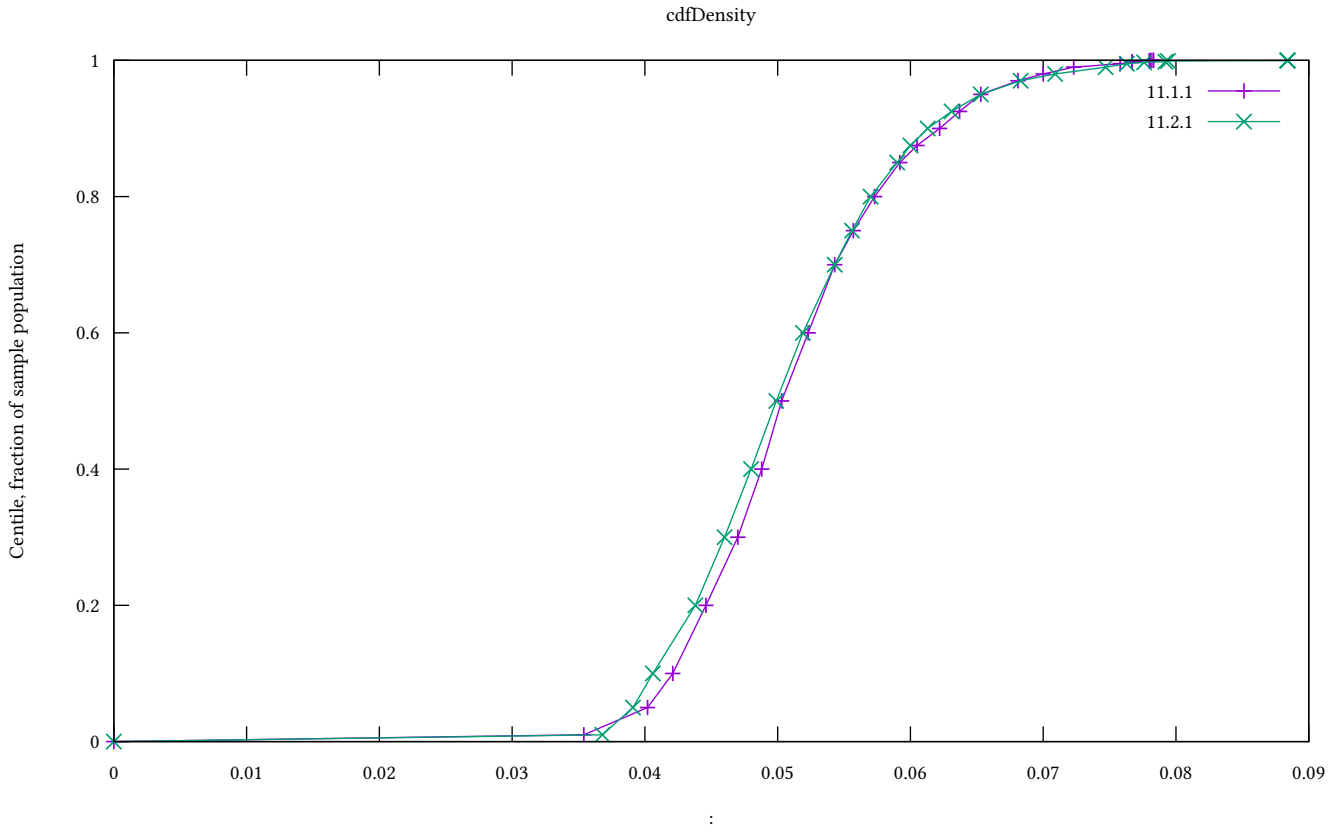


Figure 13: Chain density

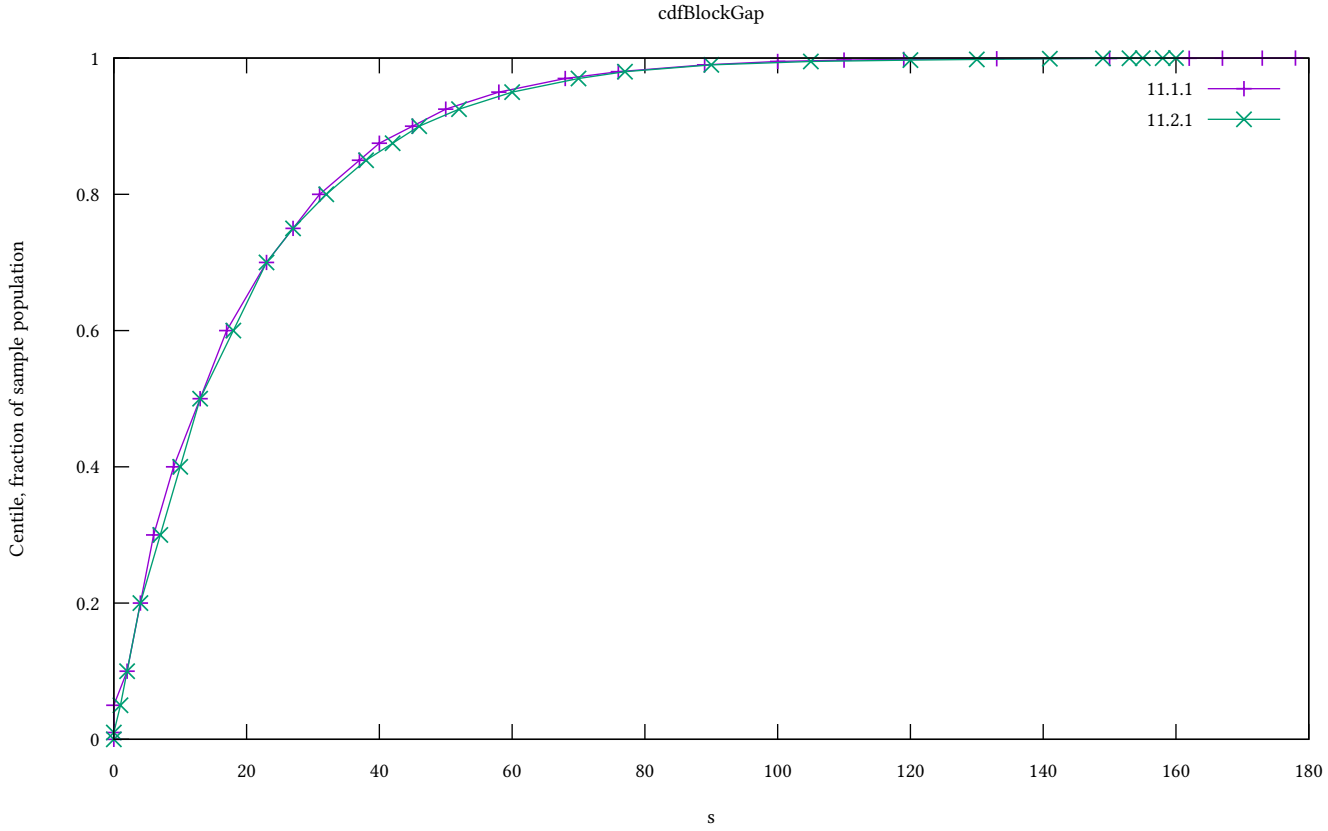


Figure 14: Interblock gap

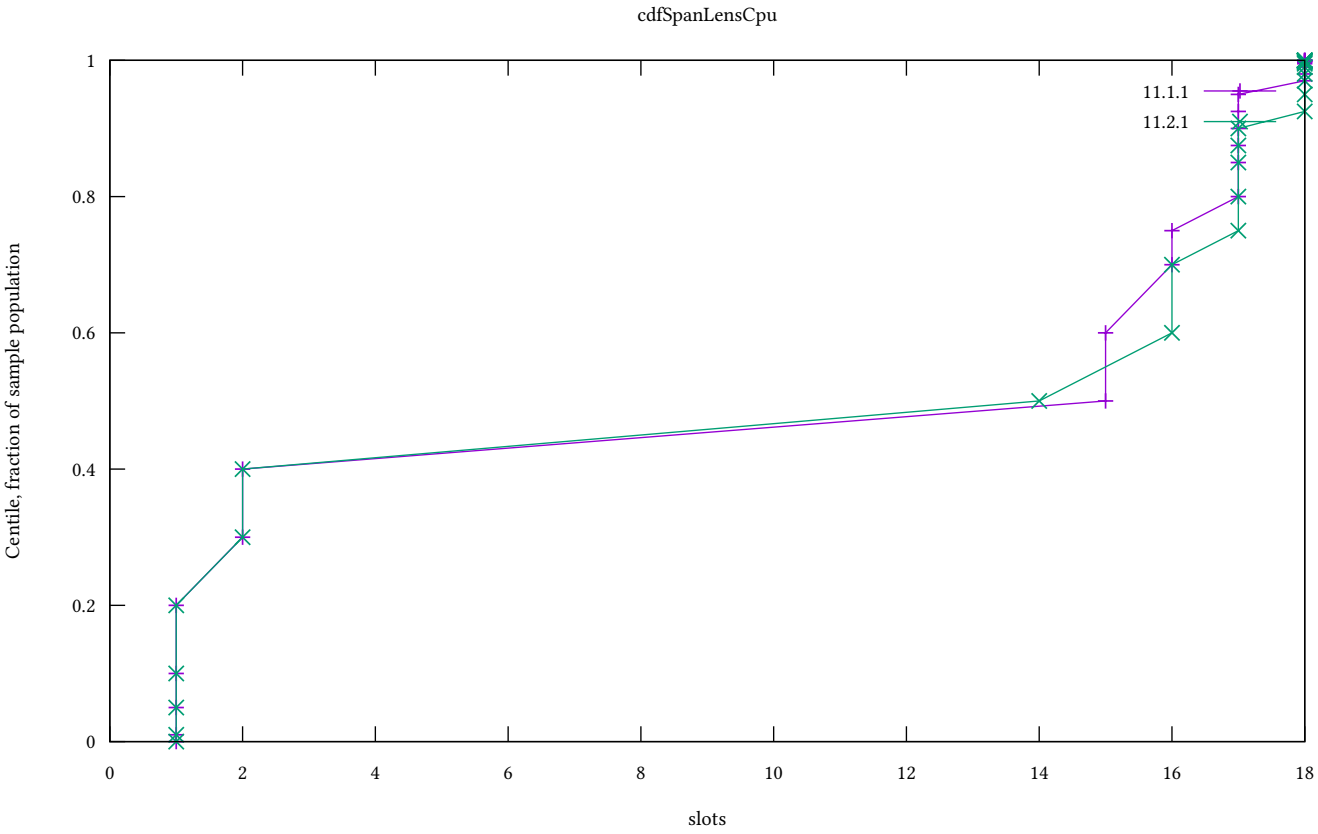


Figure 15: CPU 85% peaks, all

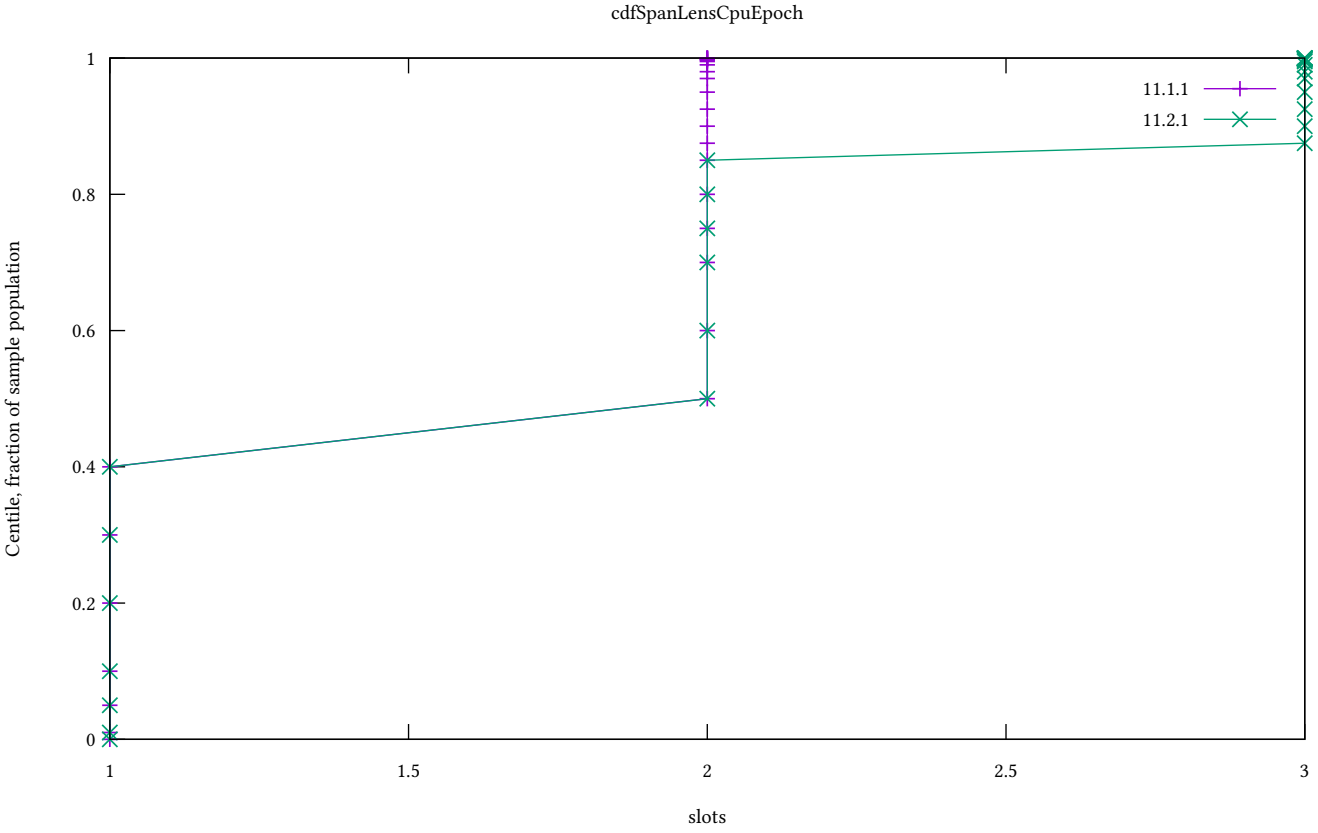


Figure 16: CPU 85% peaks, ep. trans.

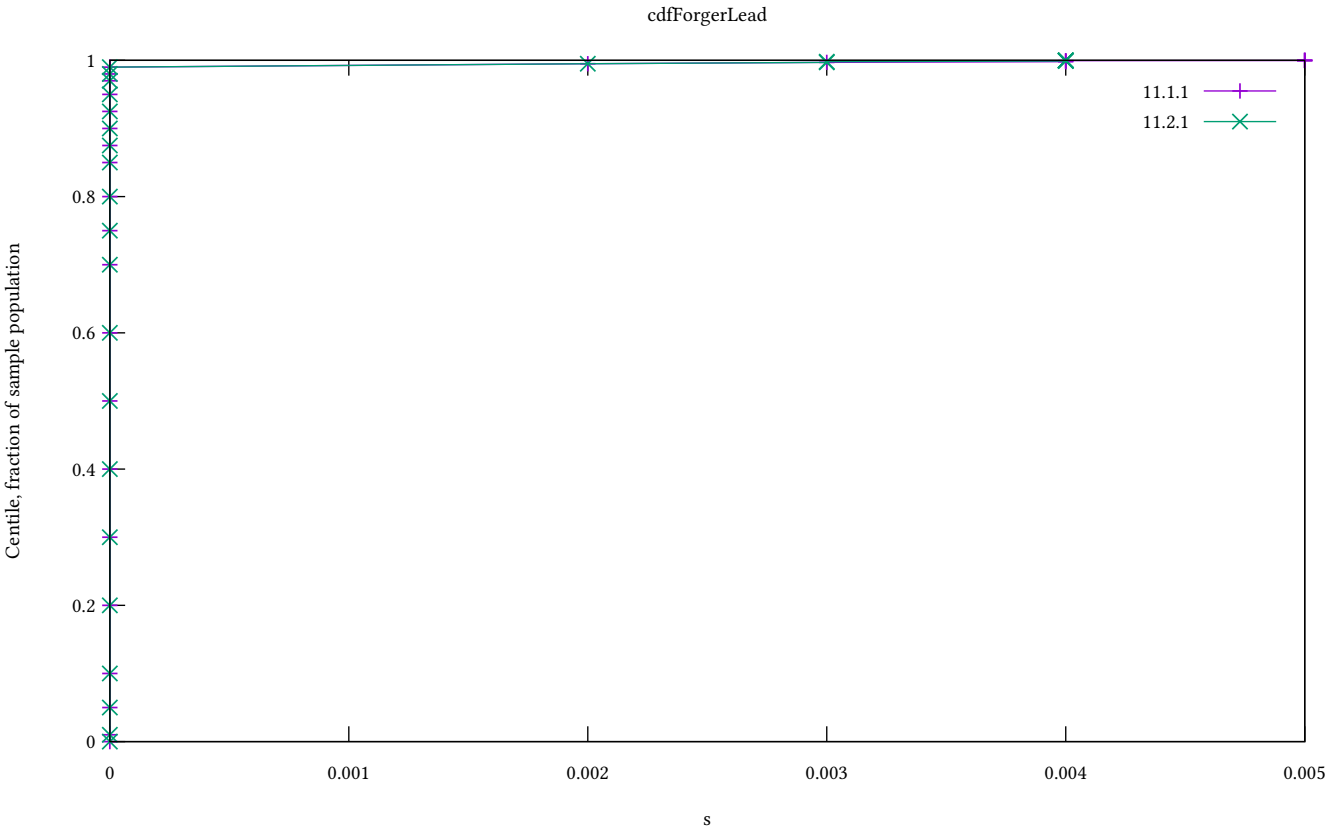


Figure 17: Leadership check duration

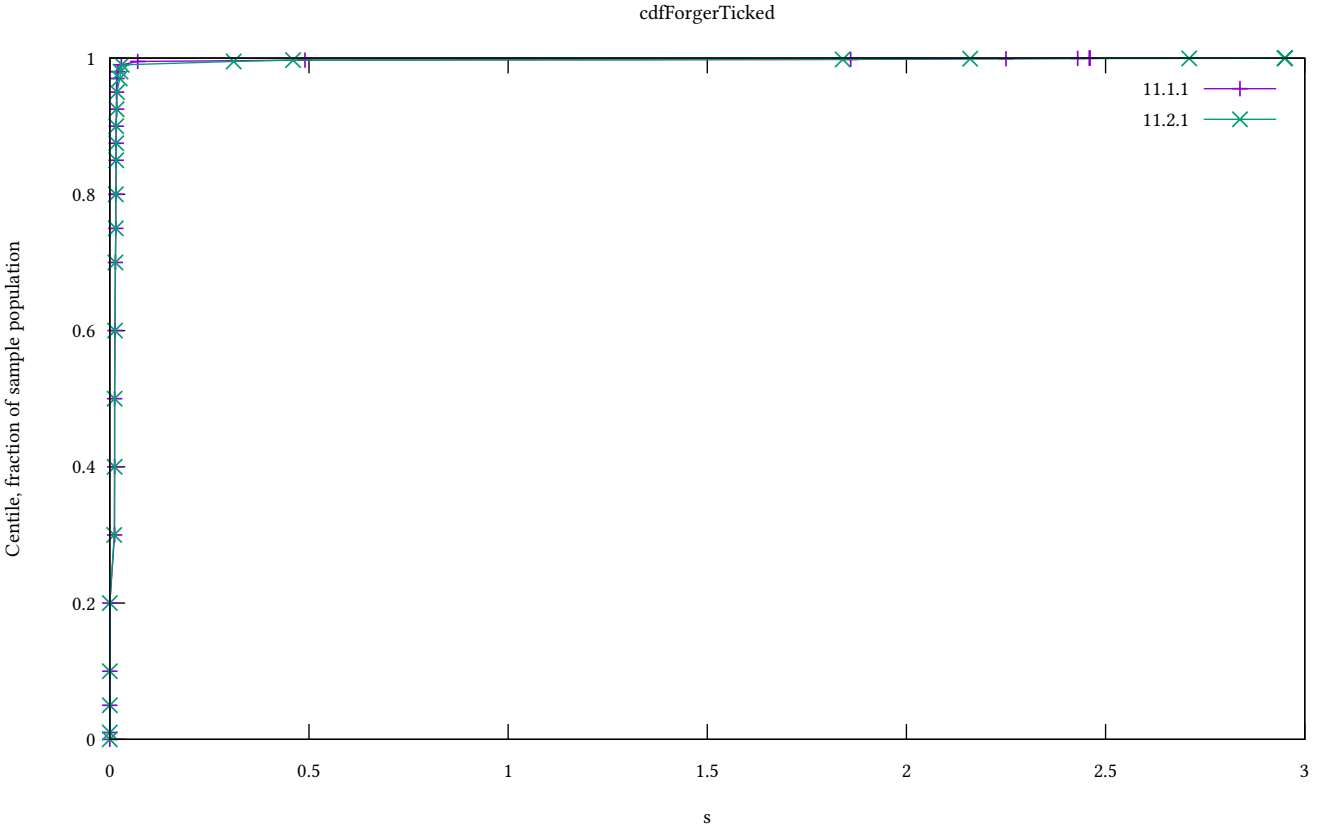


Figure 18: Ledger ticking

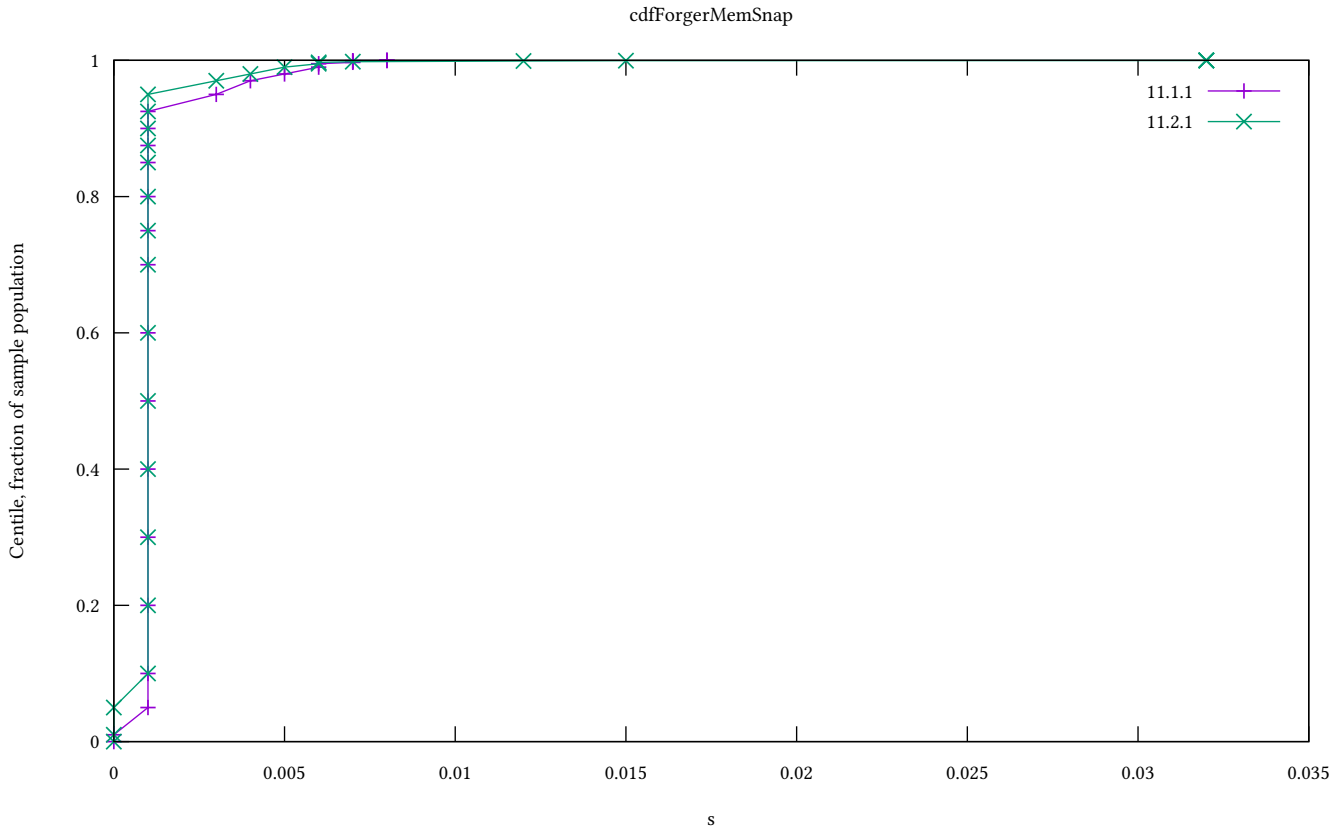


Figure 19: Mempool snapshotting

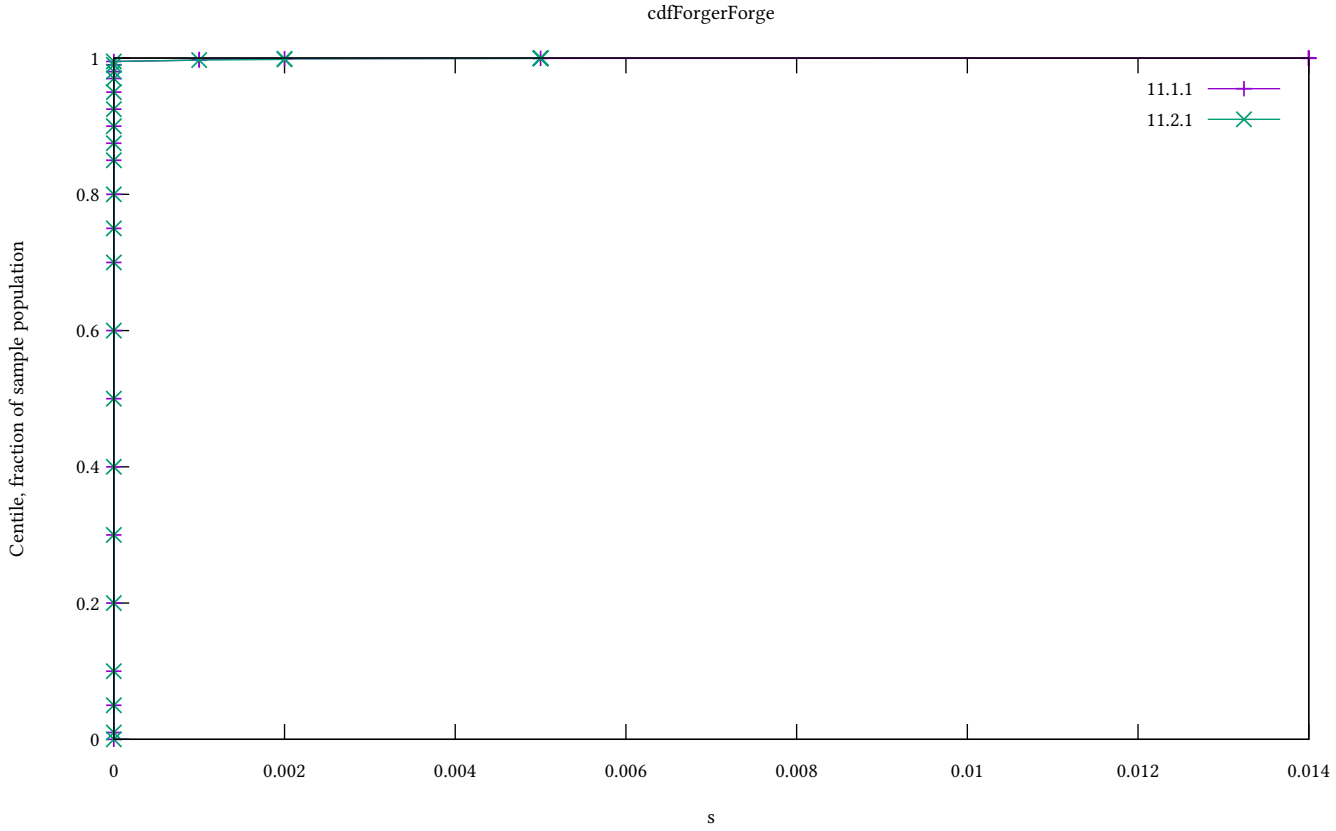


Figure 20: Leadership to forged

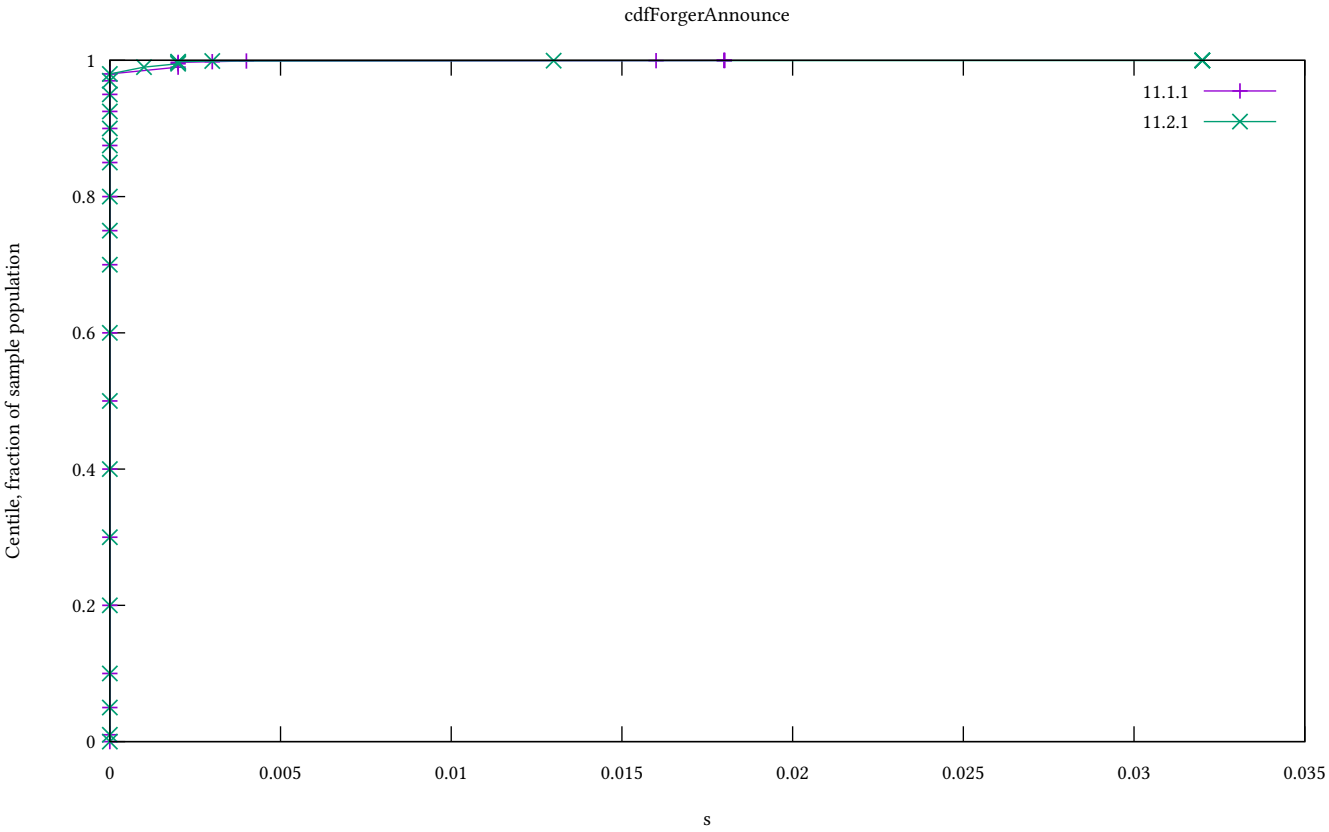


Figure 21: Forged to announced

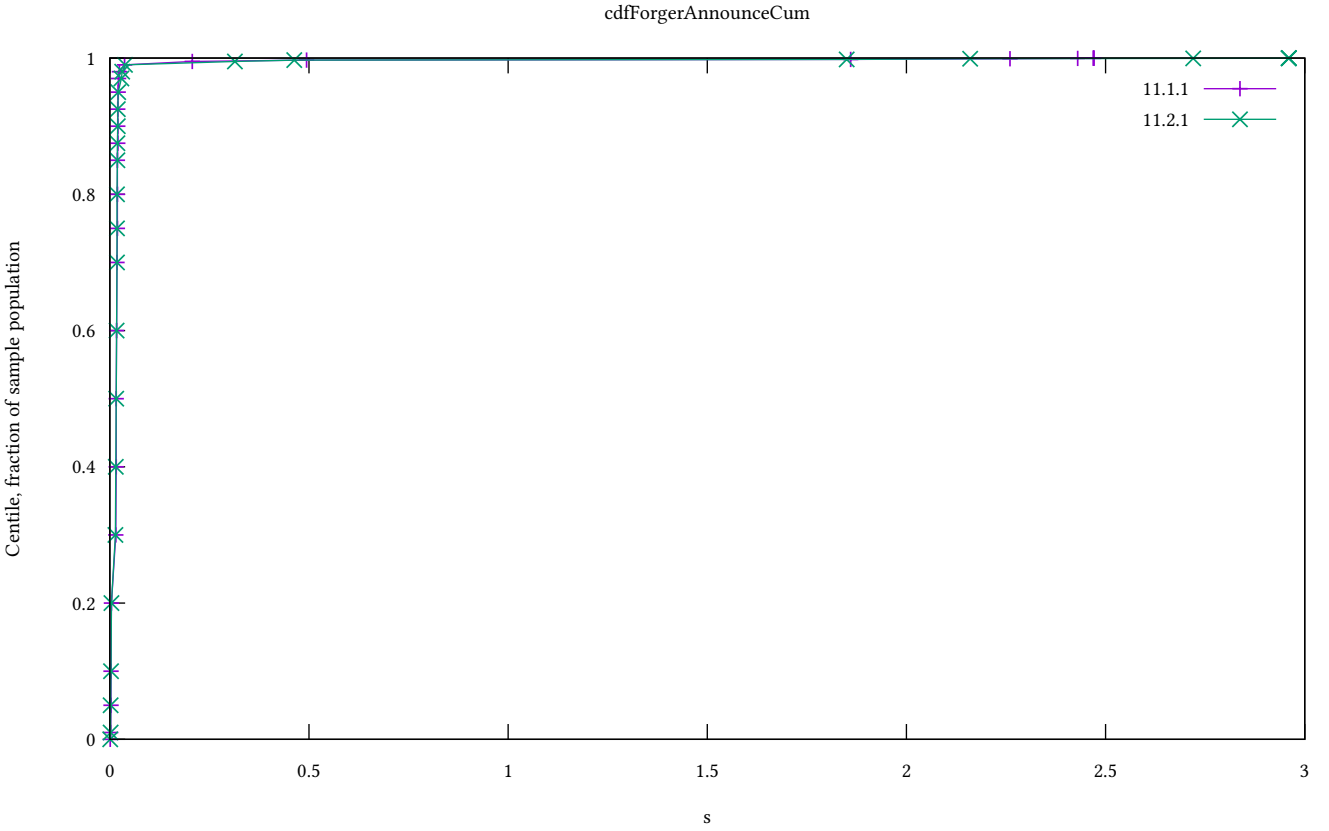


Figure 22: Slot start to announced

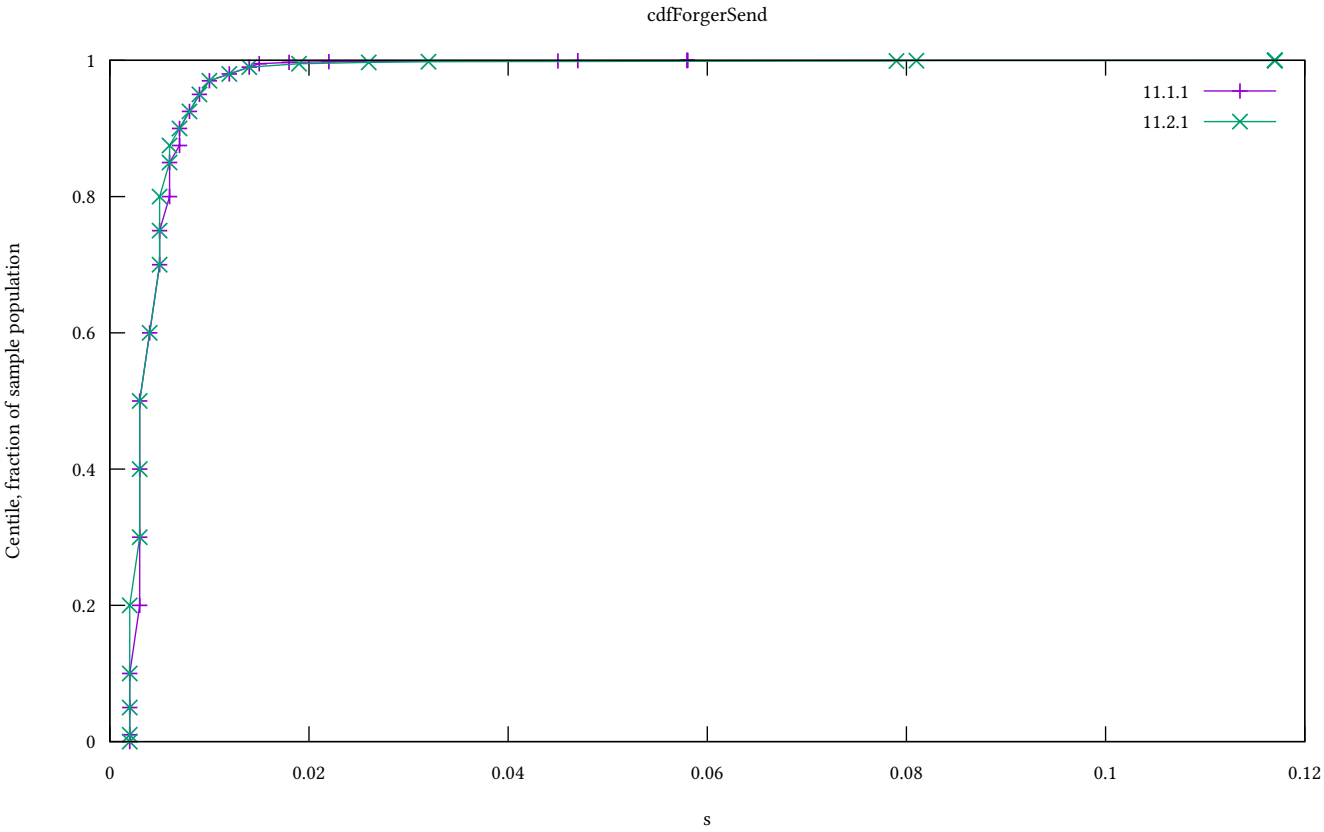


Figure 23: Forged to sending

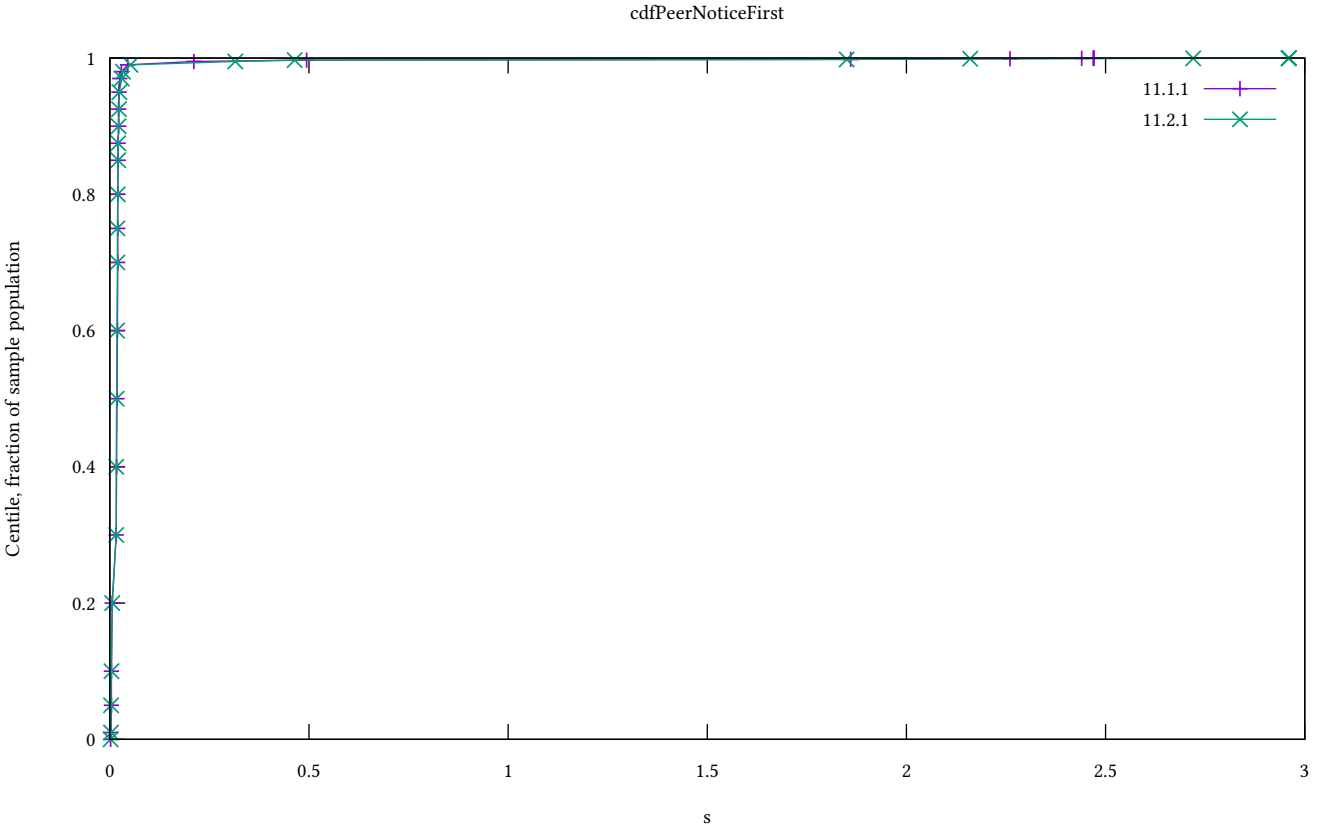


Figure 24: First peer notice

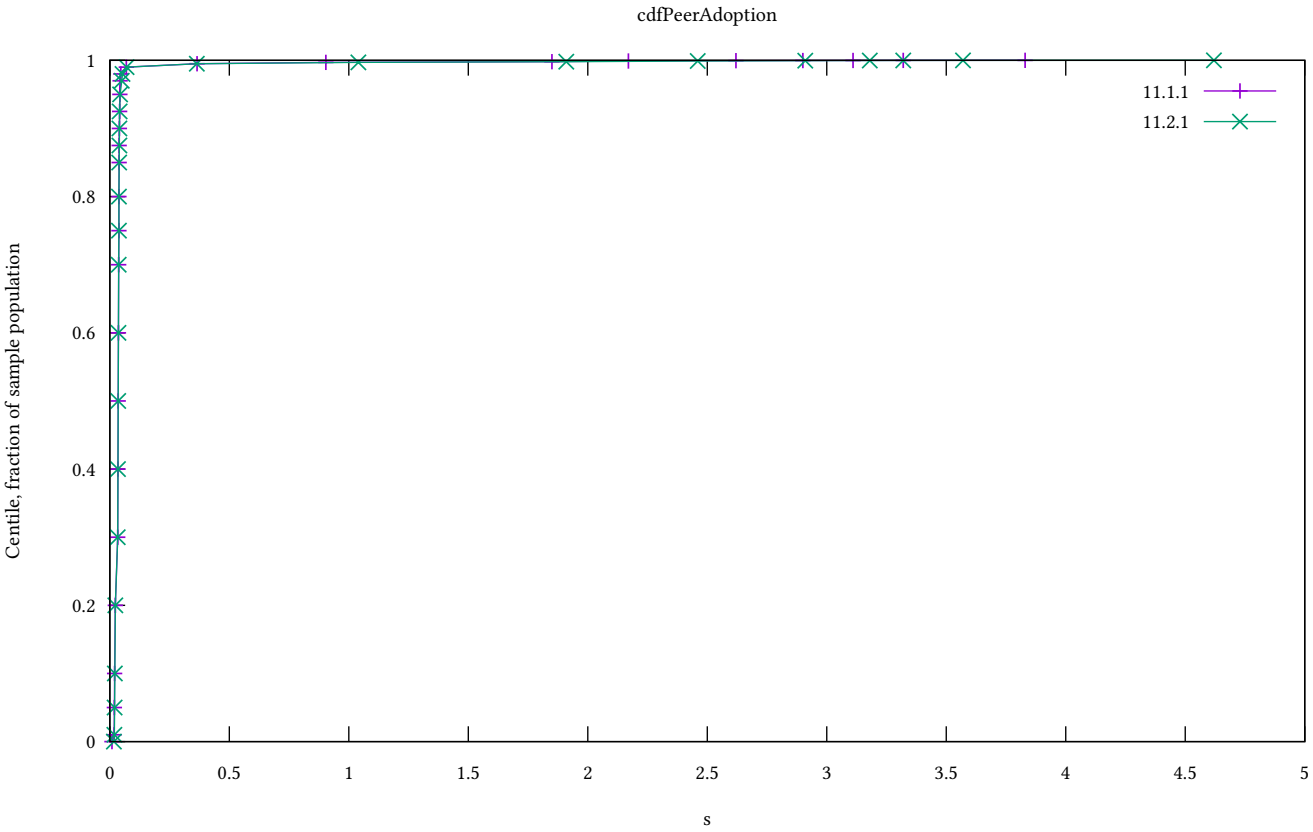


Figure 25: Fetched to adopted

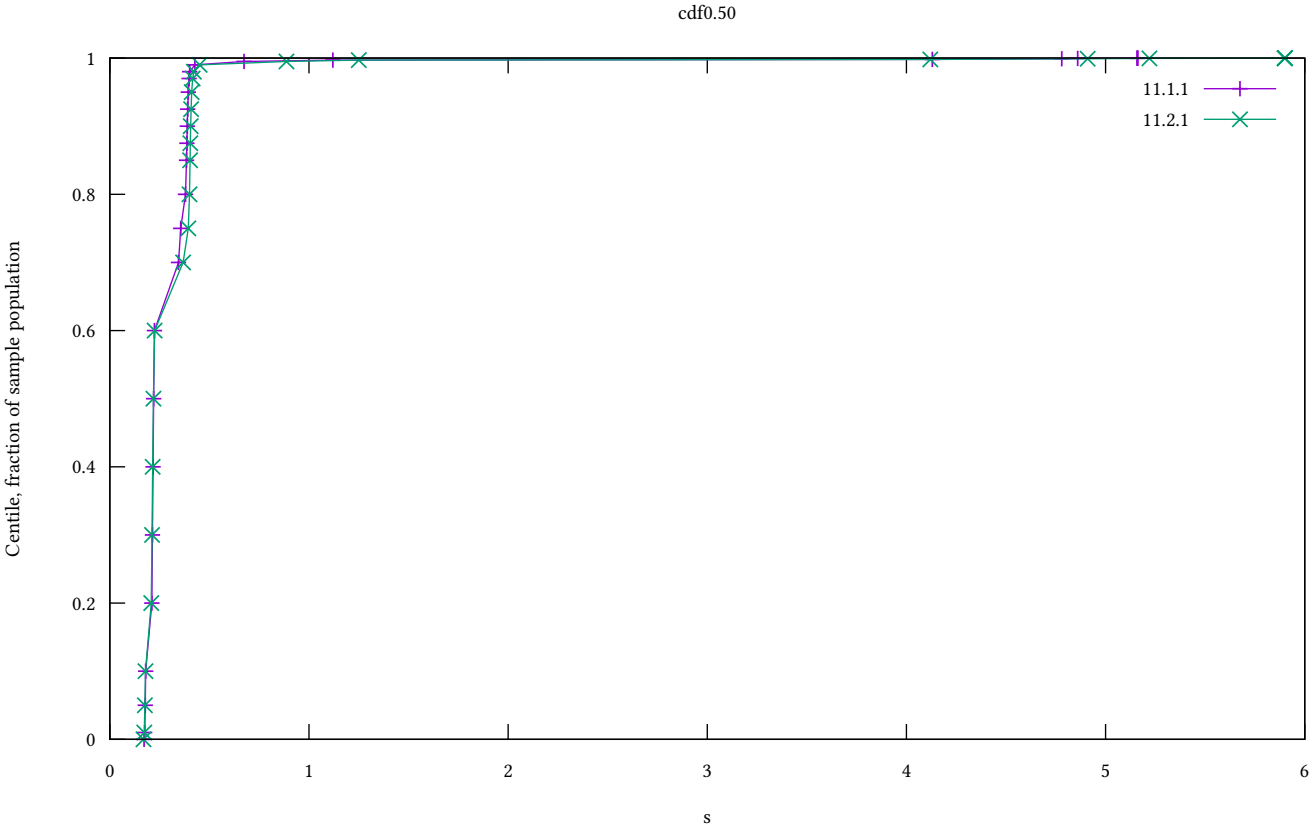


Figure 26: 0.50 adoption

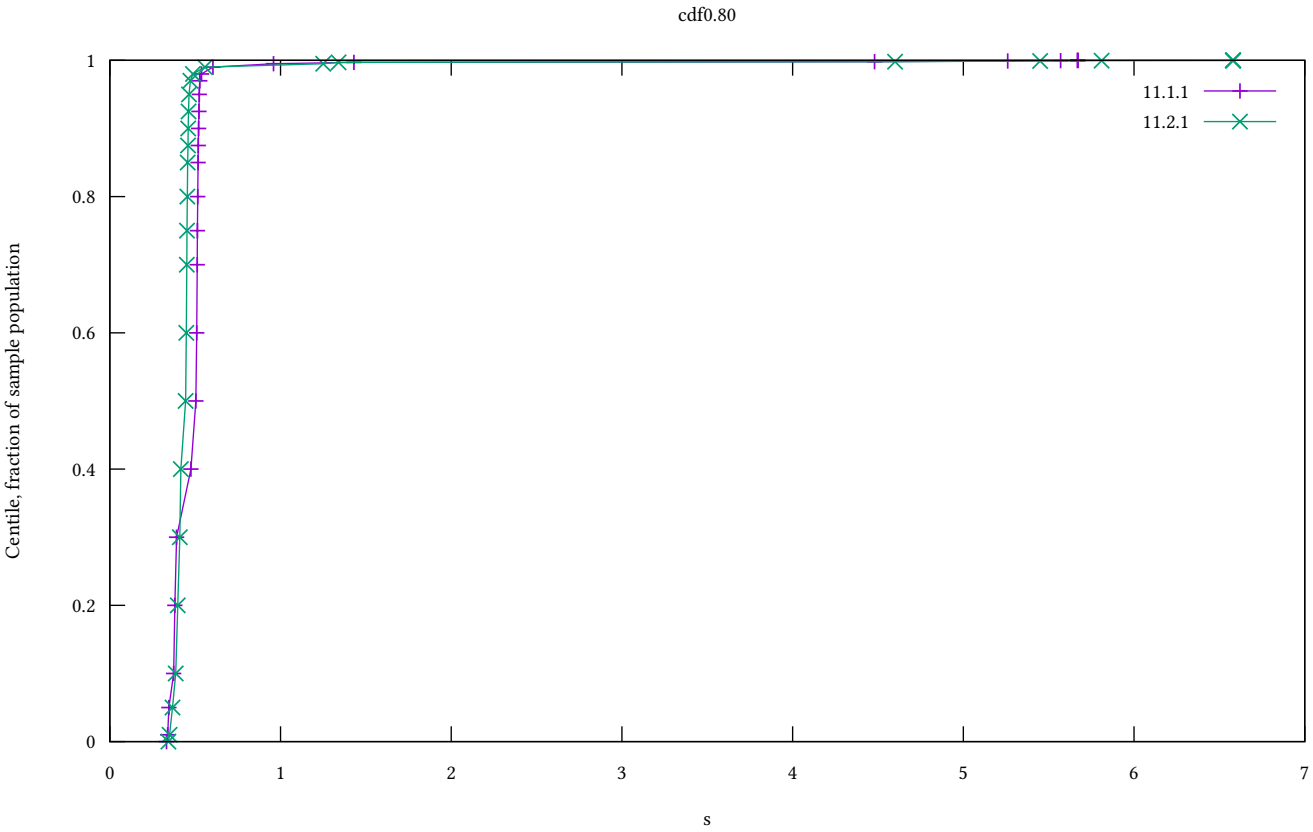


Figure 27: 0.80 adoption

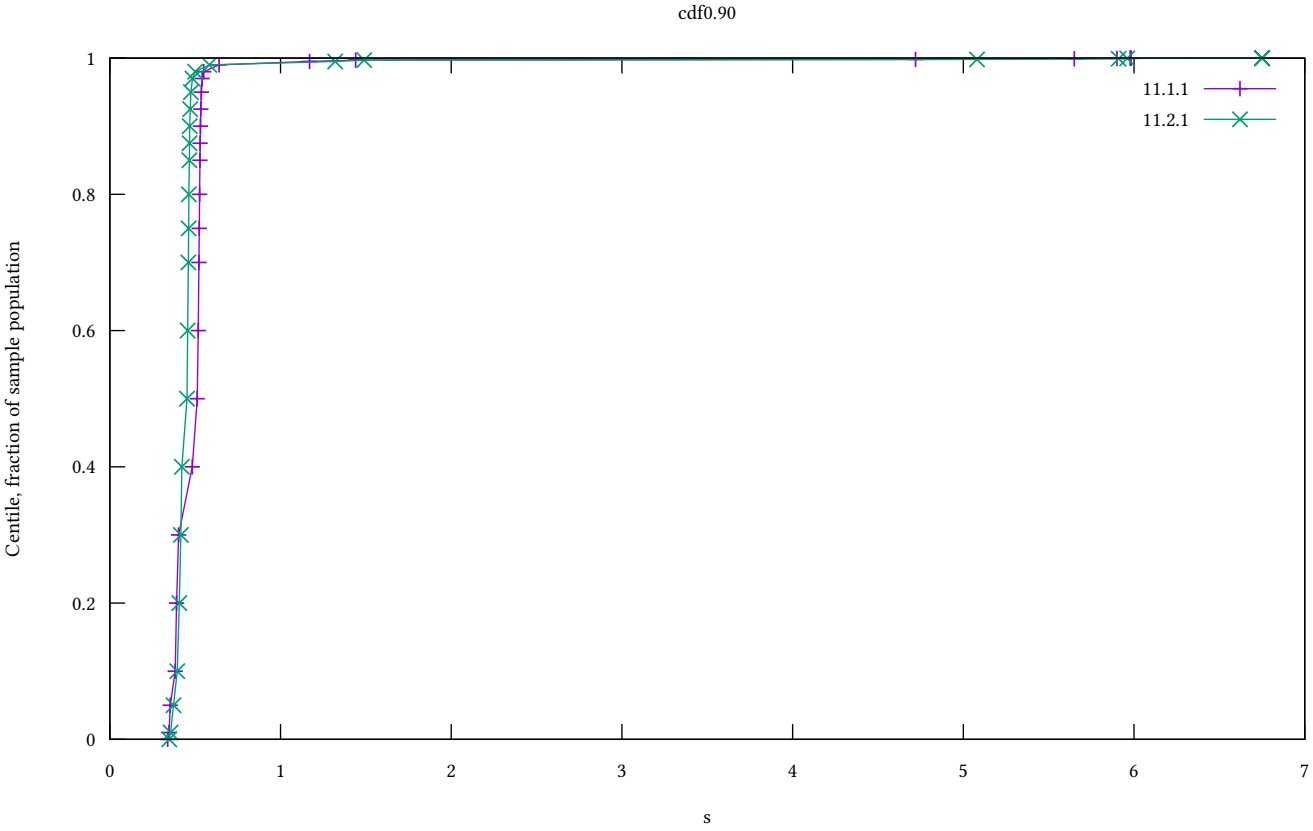


Figure 28: 0.90 adoption

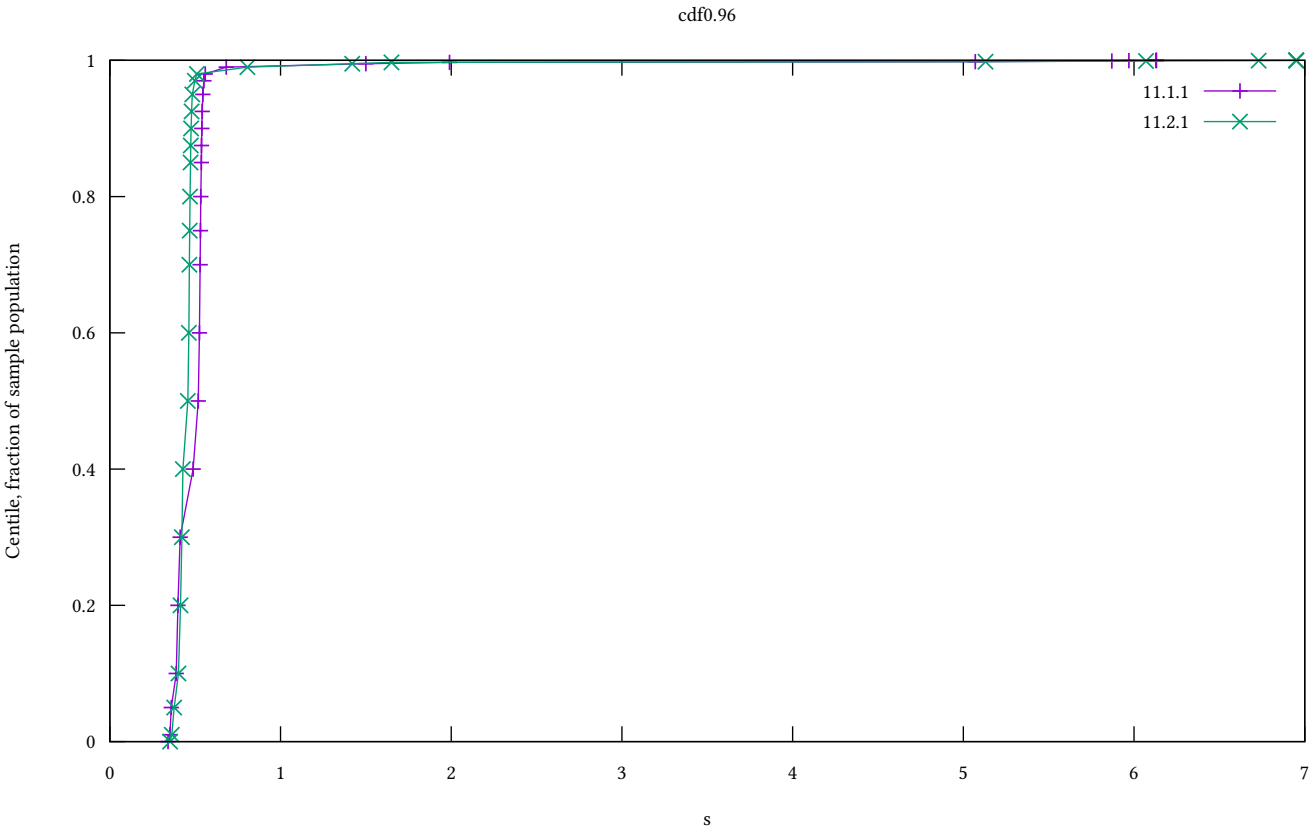


Figure 29: 0.96 adoption

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.

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

Height battles (*cdfHeightBattle*) – For a given block on chain, number of other blocks at the same height, forged for another slot

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

Slot battles (*cdfSlotBattle*) – For a given block on chain, number of other blocks forged for the same slot, at any height

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

Proc CPU ÷ txns (*cdfCpuPerTx*) – Kernel-reported process CPU time, divided by the transactions processed. Not a per-tx cost: includes block validation, forging, snapshots and GC

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

GC CPU ÷ txns (*cdfGcPerTx*) – RTS-reported GC CPU time, divided by the transactions processed (as above). A heap regression shows here first

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

Live after majGC, during (*cdfLiveMajGcDuring*) – RTS-reported live data after a major GC that occurred during a ledger snapshot write (from its start until 5s after its completion)

Live after majGC, outside (*cdfLiveMajGcOutside*) – RTS-reported live data after a major GC that occurred outside of ledger snapshot writes. The decisive measure of retained data; usually few samples

Mempool fill level (*cdfMempoolTxs*) – Transactions in the mempool, per analysed slot (time-weighted). Its fill level, not its configured capacity

Mut CPU ÷ txns (*cdfMutPerTx*) – RTS-reported mutator CPU time, divided by the transactions processed (as above)

Duration (*cdfSnapDuration*) – Time it took to write a ledger snapshot to disk (ChainDB.LedgerEvent.Snapshot.TookSnapshot)

Taken (*cdfSnapTaken*) – Ledger snapshots taken (completed) per host, during the analysed slots. 0 most likely indicates a trace parsing failure

CPU 85% peaks, all (*cdfSpanLensCpu*) – Length of over-85% CPU usage peaks (consecutive slots), slots: all of them, whatever their cause

CPU 85% peaks, ep. trans. (*cdfSpanLensCpuEpoch*) – Length of the over-85% CPU usage peak ending at most 3 slots before the epoch's first block is adopted (the ledger's epoch transition), counted from the epoch's first slot, slots

CPU 85% peaks, other (*cdfSpanLensCpuOther*) – Length of over-85% CPU usage peaks neither at an epoch transition nor during a ledger snapshot write, slots. Unexplained saturation shows here

CPU 85% peaks, snapshot (*cdfSpanLensCpuSnap*) – Length of over-85% CPU usage peaks overlapping a ledger snapshot write (from its start until 5s after its completion), slots. Compared to Duration: CPU-bound vs. I/O-bound snapshots. Needs snapshot durations, as stored by locli 3.0

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

Txns accepted (*cdfTxsAccepted*) – Processed transactions that were added to the mempool

Txns processed (*cdfTxsProcessed*) – Transactions received from peers and submitted to the mempool, per host, over the analysed slots

Reject ratio (*cdfTxsRejectRatio*) – Rejected / processed transactions. Explains differences between raw and per-transaction CPU usage

Txns rejected (*cdfTxsRejected*) – Processed transactions not added to the mempool: invalid, or already known. Run total of processed minus accepted, as the node's per-batch count isn't reliable