# 11.1.0 against 11.0.1

value-only 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 value-only workload.

| | 11.0.1 | 11.1.0 |
|---|---|---|
| Analysis date | 2026-05-07 | 2026-08-24 |
| Cluster system start date | 2026-05-06 | 2026-08-24 |
| Cluster system start time | 08:35:25 | 01:20:54 |
| 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 | 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 | 5032616.9615 | 3988285.0769 |
| Log objects analysed per host | 2127801.4615 | 1799107.8269 |
| Host run time, s | 63894.2 | 63938.8 |
| Host log line rate, Hz | 78.765 | 62.376 |
| Total log objects analysed | 110645676 | 93553607 |
| Run time, s | 63899 | 63950 |
| Analysed run duration, s | 48023 | 48016 |
| Run time efficiency | 0.75 | 0.75 |
| Node start spread, s | 5.4642255 | 21.092847 |
| Node stop spread, s | 3.0434470 | 2.6560945 |
| Slots analysed | 48020 | 48014 |
| Blocks analysed | 2244 | 2263 |
| Blocks rejected | 961 | 892 |

# Analysis

## Resource Usage

| | 11.0.1 | 11.1.0 | Δ | Δ% |
|---|---|---|---|---|
| Forge loop starts, units | 0.9999 | 0.99964 | −0.000 | −0.0 |
| Process CPU usage, % | 2.9488 | 2.322 | −0.627 | −21 |
| RTS GC CPU usage, % | 0.3362 | 0.2846 | −0.052 | −15 |
| RTS Mutator CPU usage, % | 2.6129 | 2.037 | −0.576 | −22 |
| Major GCs, events | 0.0009 | 0.0007 | −0.000 | −25 |
| Minor GCs, events | 0.7875 | 0.6065 | −0.181 | −23 |
| Kernel RSS, MiB | 6974.2 | 9116.7 | 2142.564 | 31 |
| RTS heap size, MiB | 6905.2 | 9046.7 | 2141.536 | 31 |
| RTS live GC dateset, MiB | 3101.2 | 3918.7 | 817.482 | 26 |
| RTS alloc rate, MiB/s | 24.387 | 18.535 | −5.852 | −24 |
| 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 | 7.2e-5 | 0.000 | NaN |
| Filesystem writes, KiB/s | 236.95 | 238.94 | 1.981 | 0.8 |
| CPU 85% spans, slots | 6.9956 | 9.1334 | 2.138 | 31 |

## Anomaly control

| | 11.0.1 | 11.1.0 | Δ | Δ% |
|---|---|---|---|---|
| Blocks per host, blocks | 63.808 | 62.615 | −1.192 | −2 |
| Filtered to chained blocks, : | 0.6995 | 0.7175 | 0.018 | 3 |
| Chained to forged blocks, : | 0.9659 | 0.9697 | 0.004 | 0.4 |
| Height & slot battles, blocks | 0.00713 | 0.0053 | −0.002 | −26 |
| Block size, Bytes | 88963 | 88970 | 7.356 | 0.0 |

## Forging

| | 11.0.1 | 11.1.0 | Δ | Δ% |
|---|---|---|---|---|
| Started forge loop iteration, s | 0.00176 | 0.00093 | −0.001 | −47 |
| Acquired block context, s | 6.2e-5 | 6.4e-5 | 0.000 | 0 |
| Acquired ledger state, s | 0.00011 | 0.00012 | 0.000 | 0 |
| Acquired ledger view, s | 3.1e-5 | 3.4e-5 | 0.000 | 0 |
| Leadership check duration, s | 0.00043 | 0.00043 | −0.000 | 0 |
| Ledger ticking, s | 0.01659 | 0.01921 | 0.003 | 16 |
| Mempool snapshotting, s | 0.04764 | 0.0428 | −0.005 | −10 |
| Leadership to forged, s | 0.0008 | 0.00082 | 0.000 | 0 |
| Forged to announced, s | 0.00066 | 0.00066 | −0.000 | 0 |
| Forged to sending, s | 0.00574 | 0.0058 | 0.000 | 0 |
| Forged to self-adopted, s | 0.06514 | 0.06704 | 0.002 | 3 |
| Slot start to announced, s | 0.06809 | 0.06505 | −0.003 | −4 |

## Individual peer propagation

| | 11.0.1 | 11.1.0 | Δ | Δ% |
|---|---|---|---|---|
| First peer notice, s | 0.06985 | 0.06686 | −0.003 | −4 |
| First peer fetch, s | 0.08073 | 0.07789 | −0.003 | −4 |
| Notice to fetch request, s | 0.00136 | 0.00135 | −0.000 | 0 |
| Fetch duration, s | 0.34774 | 0.37185 | 0.024 | 7 |
| Fetched to announced, s | 0.00092 | 0.00089 | −0.000 | 0 |
| Fetched to sending, s | 0.04242 | 0.04287 | 0.000 | 1 |
| Fetched to adopted, s | 0.06774 | 0.06946 | 0.002 | 3 |

## End-to-end propagation

| | 11.0.1 | 11.1.0 | Δ | Δ% |
|---|---|---|---|---|
| 0.50 adoption, s | 0.5947 | 0.62934 | 0.035 | 6 |
| 0.80 adoption, s | 0.94617 | 0.99001 | 0.044 | 5 |
| 0.90 adoption, s | 0.96464 | 1.00837 | 0.044 | 5 |
| 0.92 adoption, s | 0.96881 | 1.01262 | 0.044 | 5 |
| 0.94 adoption, s | 0.97305 | 1.01829 | 0.045 | 5 |
| 0.96 adoption, s | 0.97896 | 1.02393 | 0.045 | 5 |
| 0.98 adoption, s | 0.98682 | 1.03211 | 0.045 | 5 |
| 1.00 adoption, s | 1.00738 | 1.04928 | 0.042 | 4 |

# Observations

## Resources

1. Process CPU usage decreases by 21% under saturation workload (Mutator −22%, GC CPU −15%).
2. Allocation rate decreases by 24%; Minor GCs by 23% and Major GCs by 25%.
3. Kernel RSS and RTS heap size increase by  2.1 GiB or 31%; the RTS live GC dataset by  817 MiB or 26%.
4. Observed CPU 85% span durations lengthen by 31% ( 2.1 slots).
5. Log objects emitted per host decrease by 21% (host log line rate 78.765 -> 62.376 Hz).

## Forging

1. Mempool snapshotting improves by 5ms or 10%; Ledger ticking regresses by 2.6ms or 16%.
2. As a net effect, a block producer announces a new header 3ms or 4% earlier into a slot.
3. The remaining block production timings show no significant changes (sub-millisecond deltas).

## Peer propagation

1. Peers notice a new block 3ms or 4% earlier, and start fetching 4% earlier.
2. However, Fetch duration exhibits a clear regression by 24ms (or 7%)

## End-to-end propagation

1. We observe an increase in end-to-end propagation times of 4% - 6% across all centiles (median +35ms; 100th centile +42ms).
2. This increase can be attributed to the additional time spent in Fetch duration.

# Appendix A: charts

## Cluster performance charts

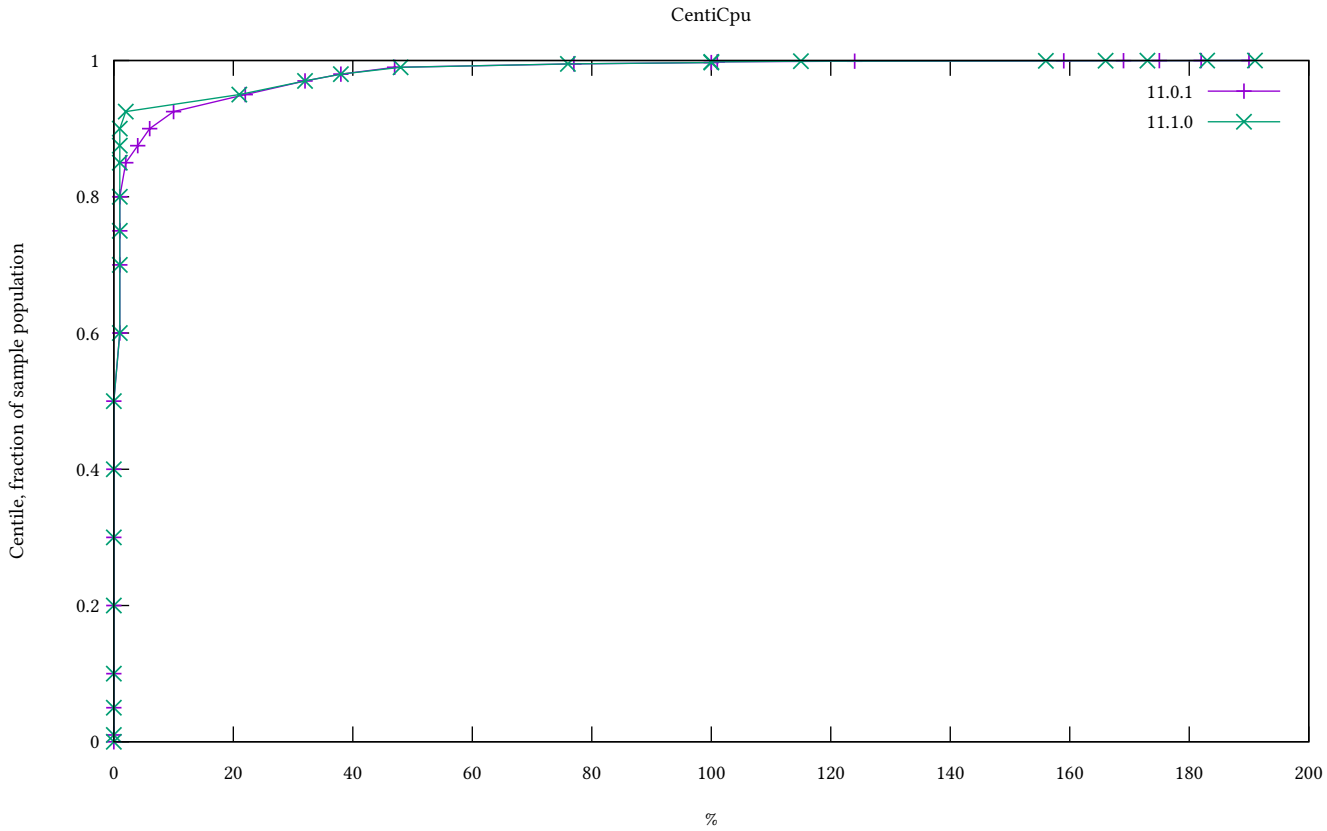


*Figure 1: Process CPU usage*

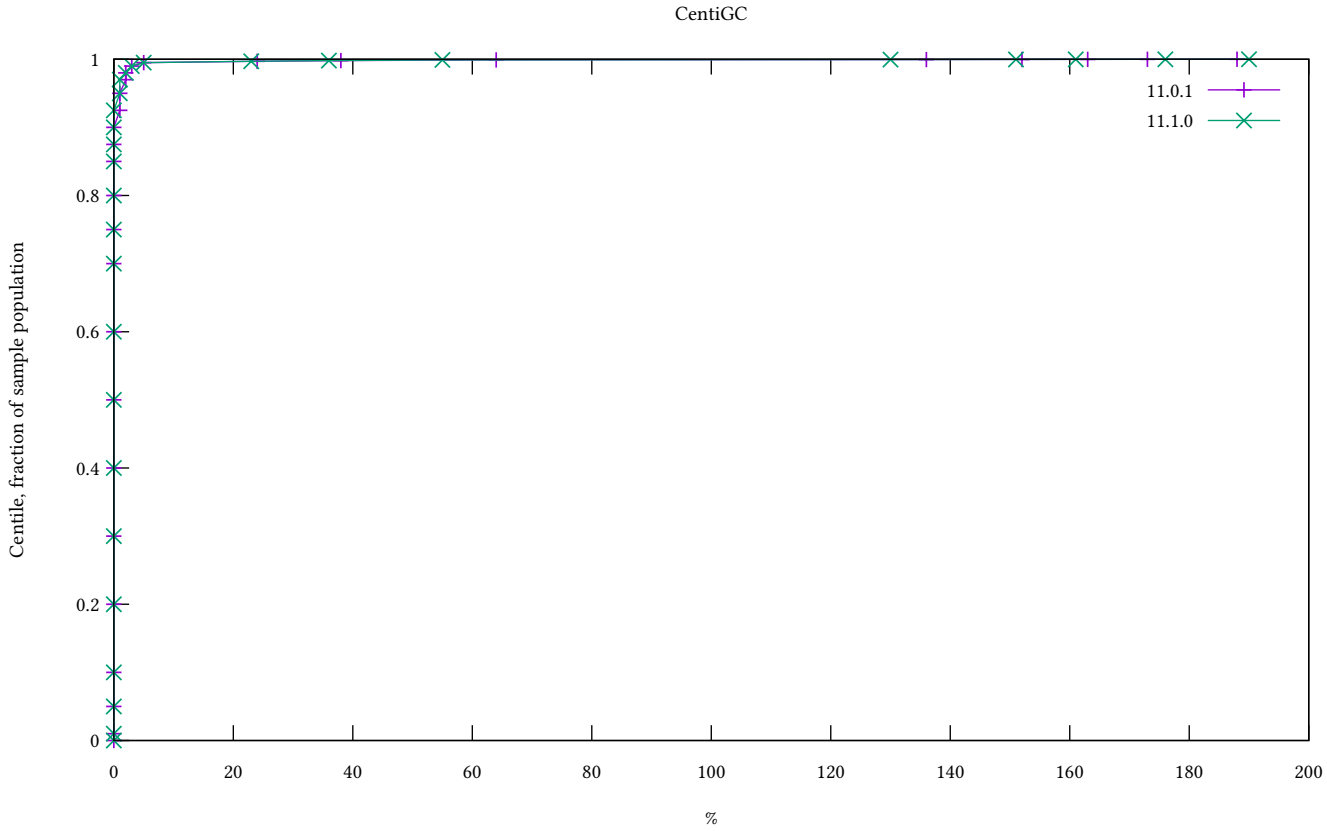


*Figure 2: RTS GC CPU usage*

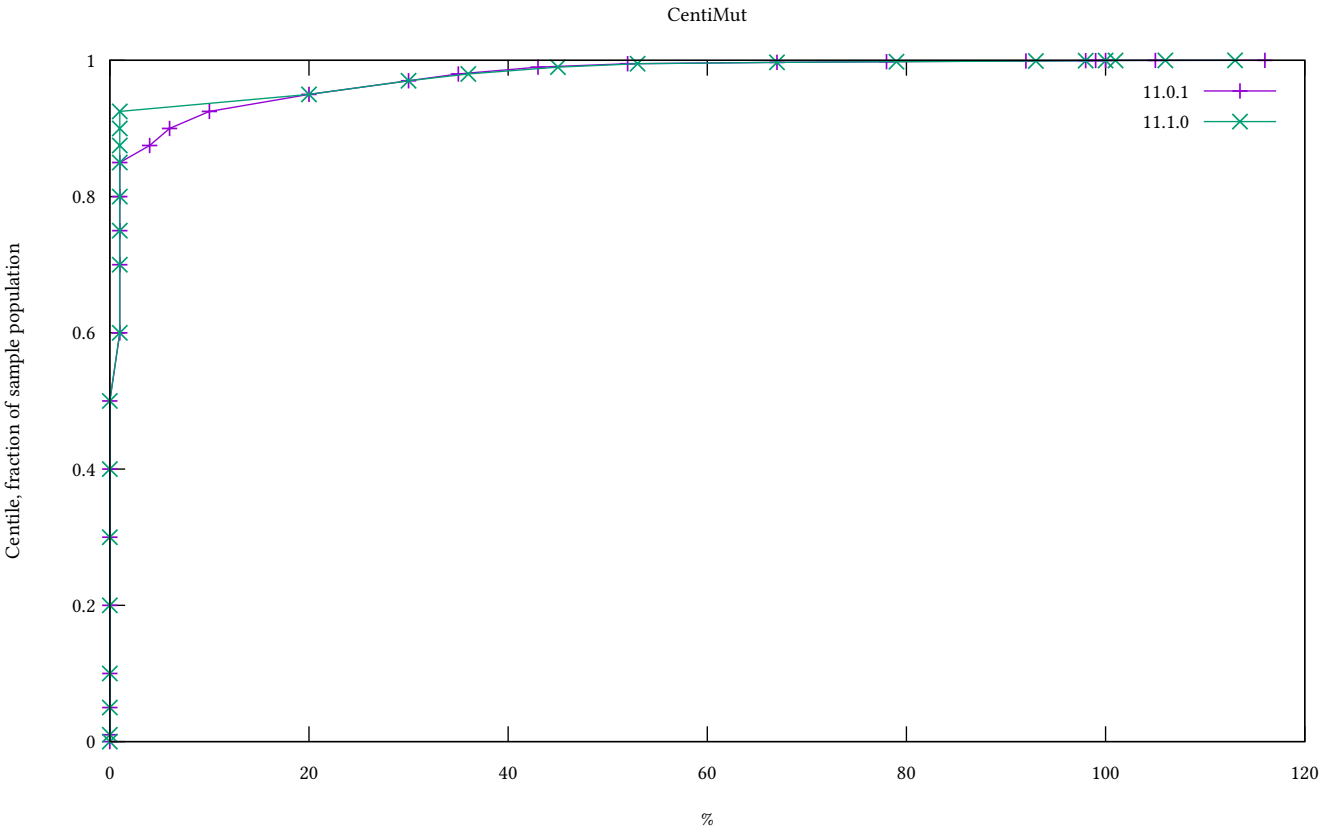


*Figure 3: RTS Mutator CPU usage*

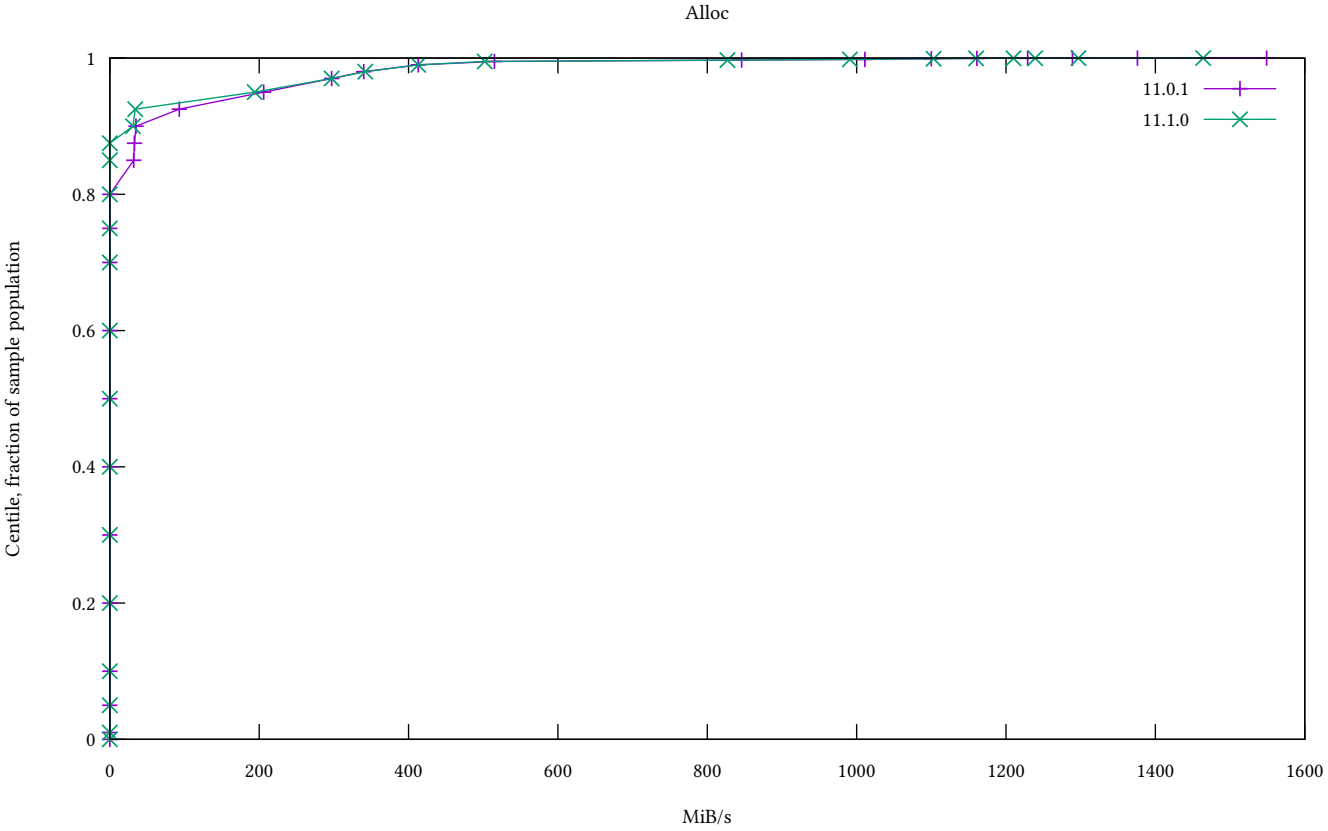


*Figure 4: RTS alloc rate*

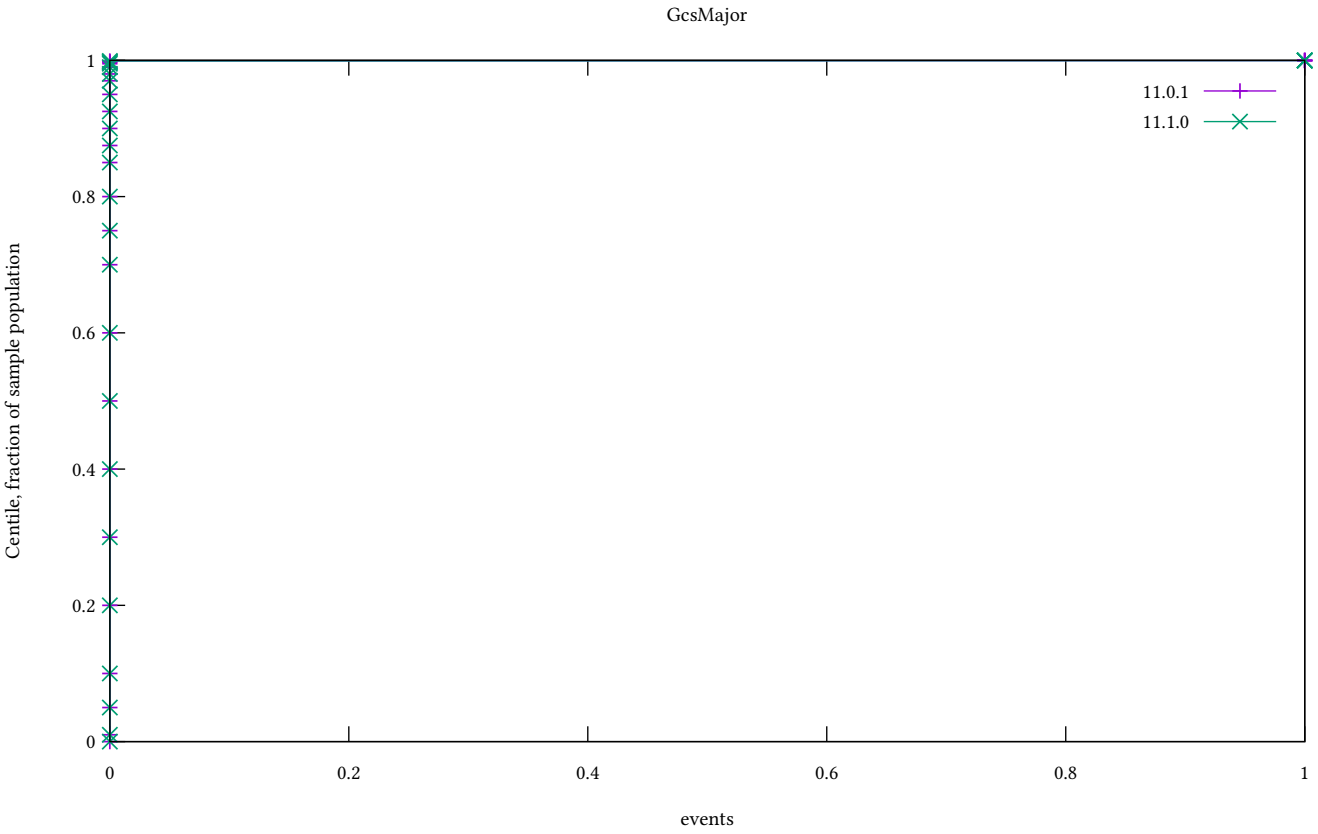


*Figure 5: Major GCs*

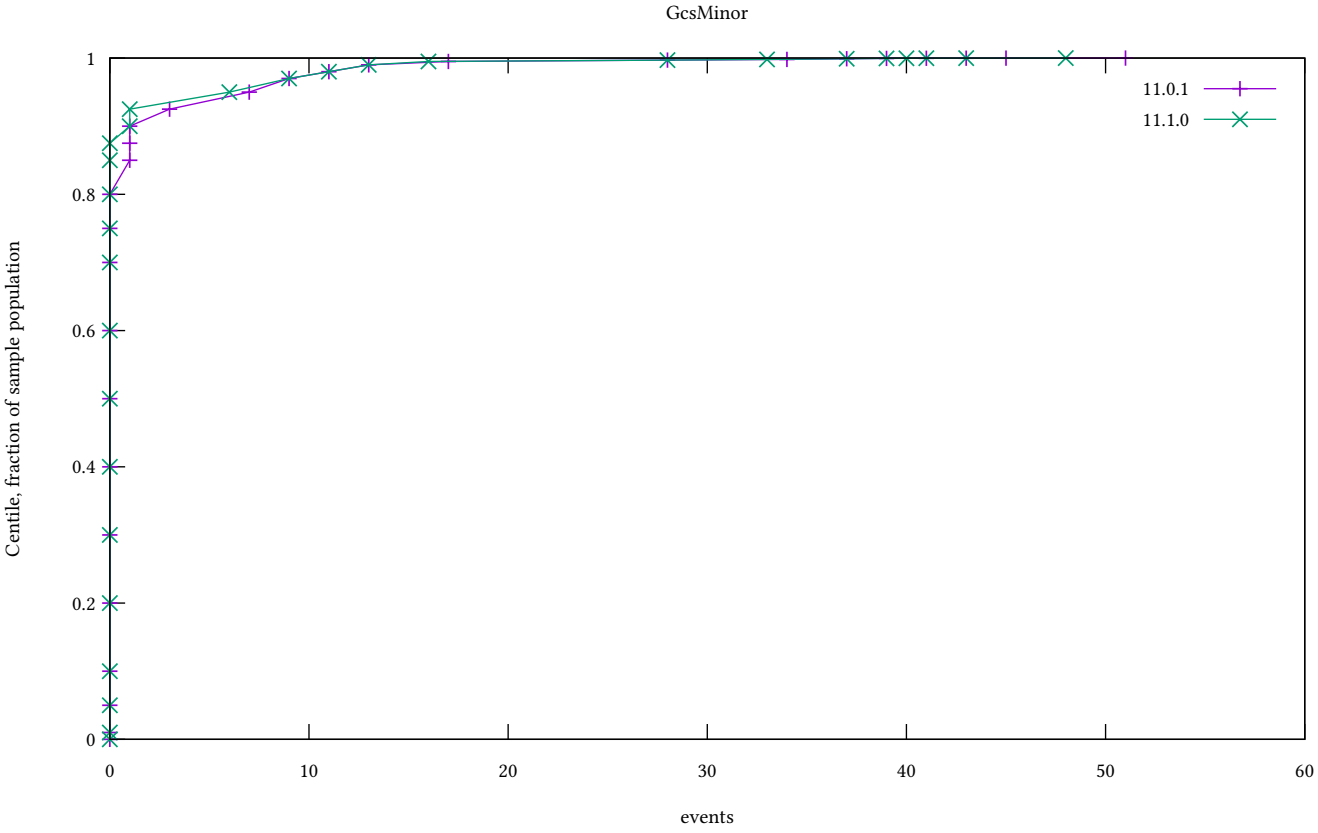


*Figure 6: Minor GCs*

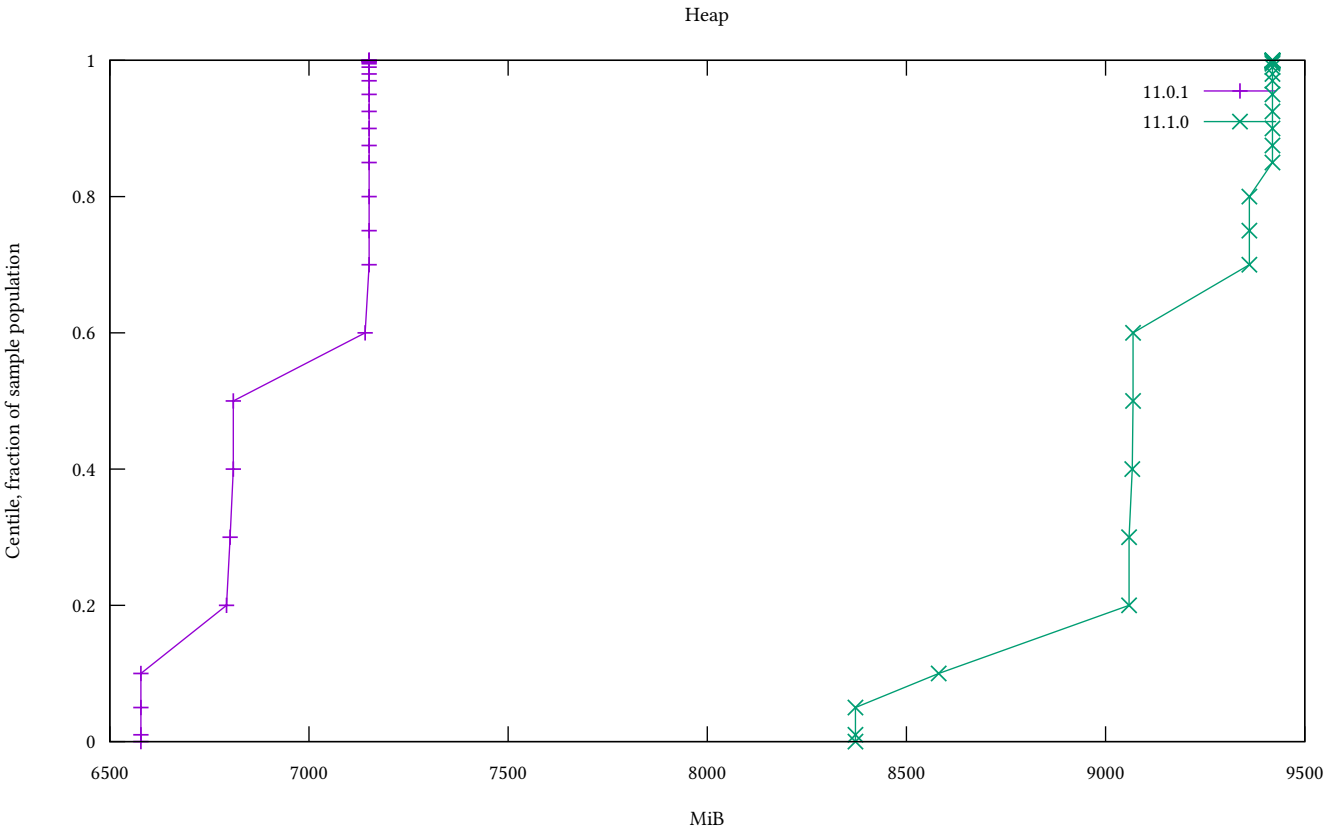


*Figure 7: RTS heap size*

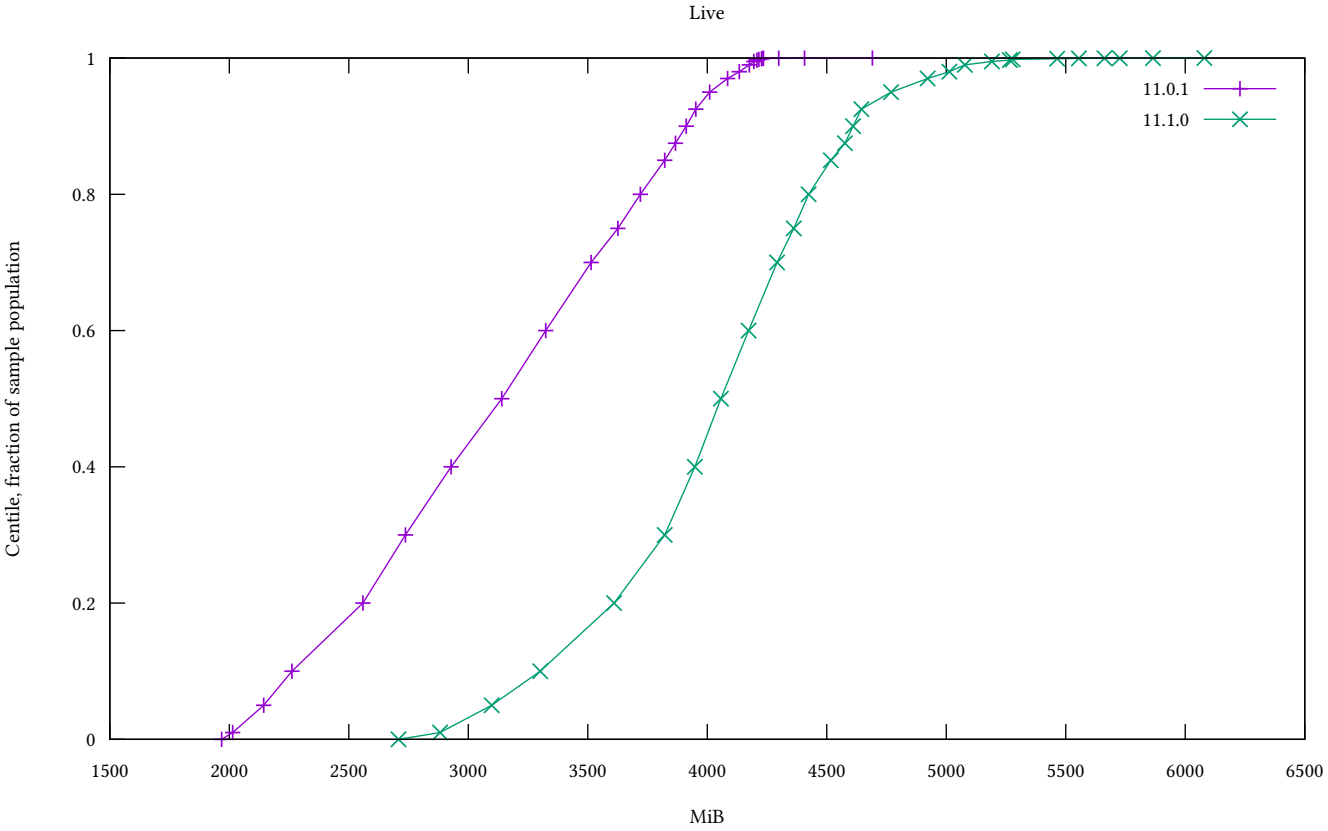


*Figure 8: RTS live GC dateset*

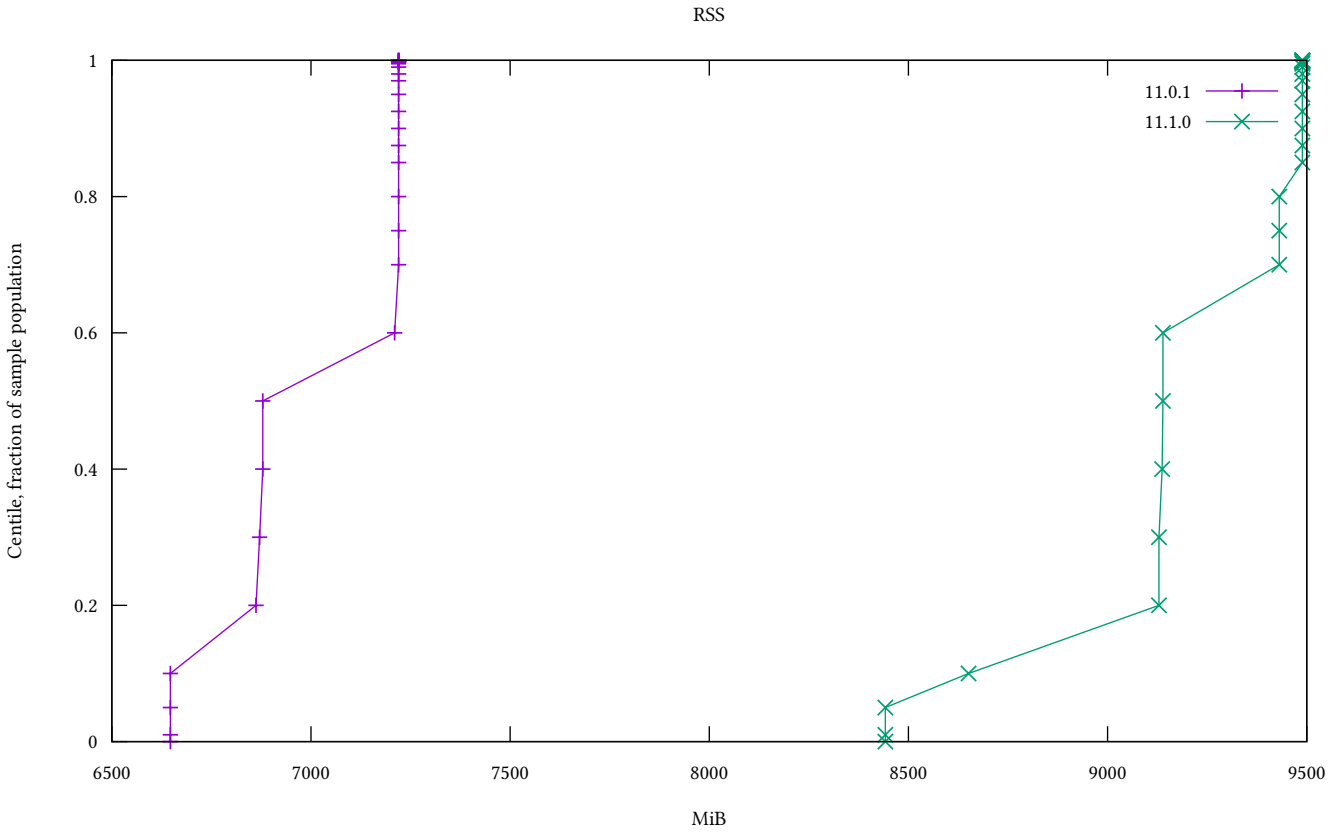


*Figure 9: Kernel RSS*

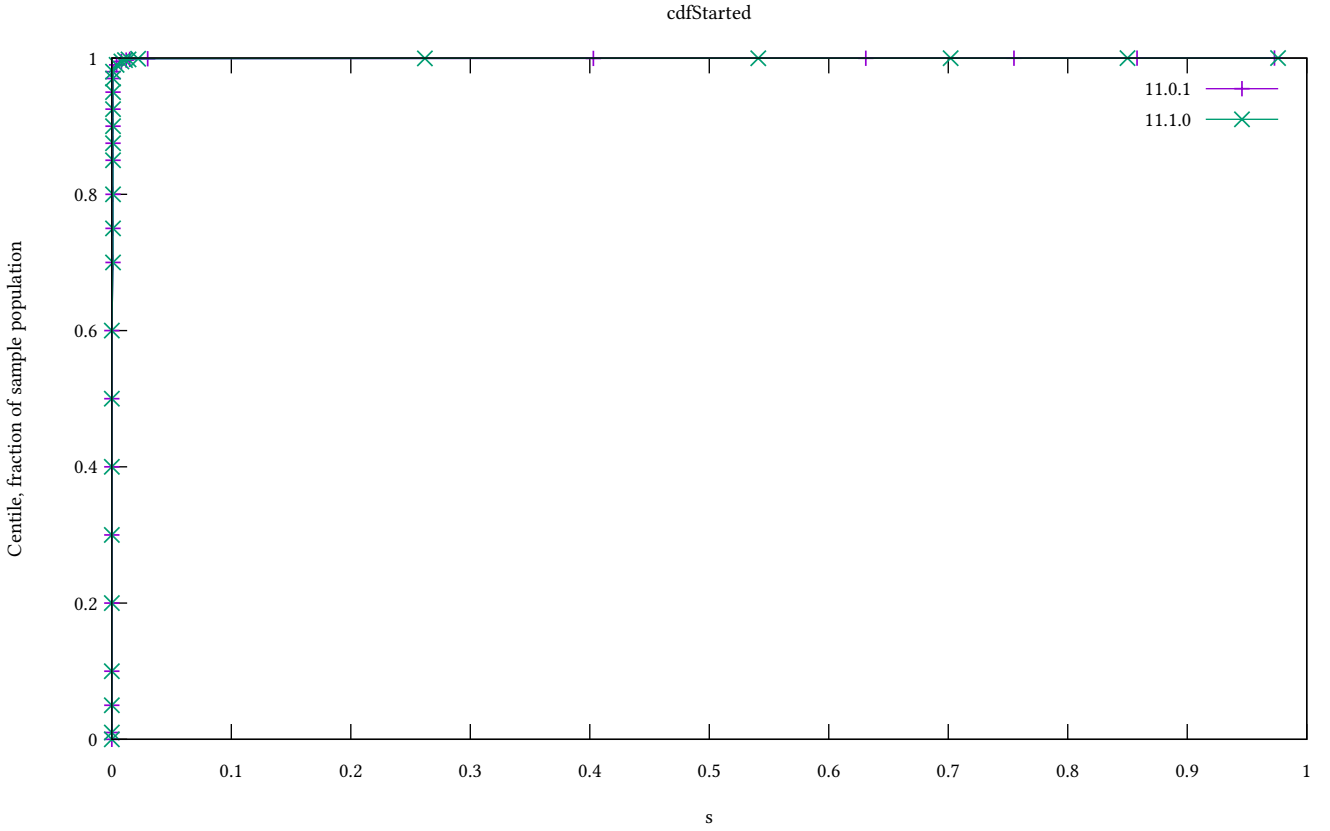


*Figure 10: Forge loop tardiness*

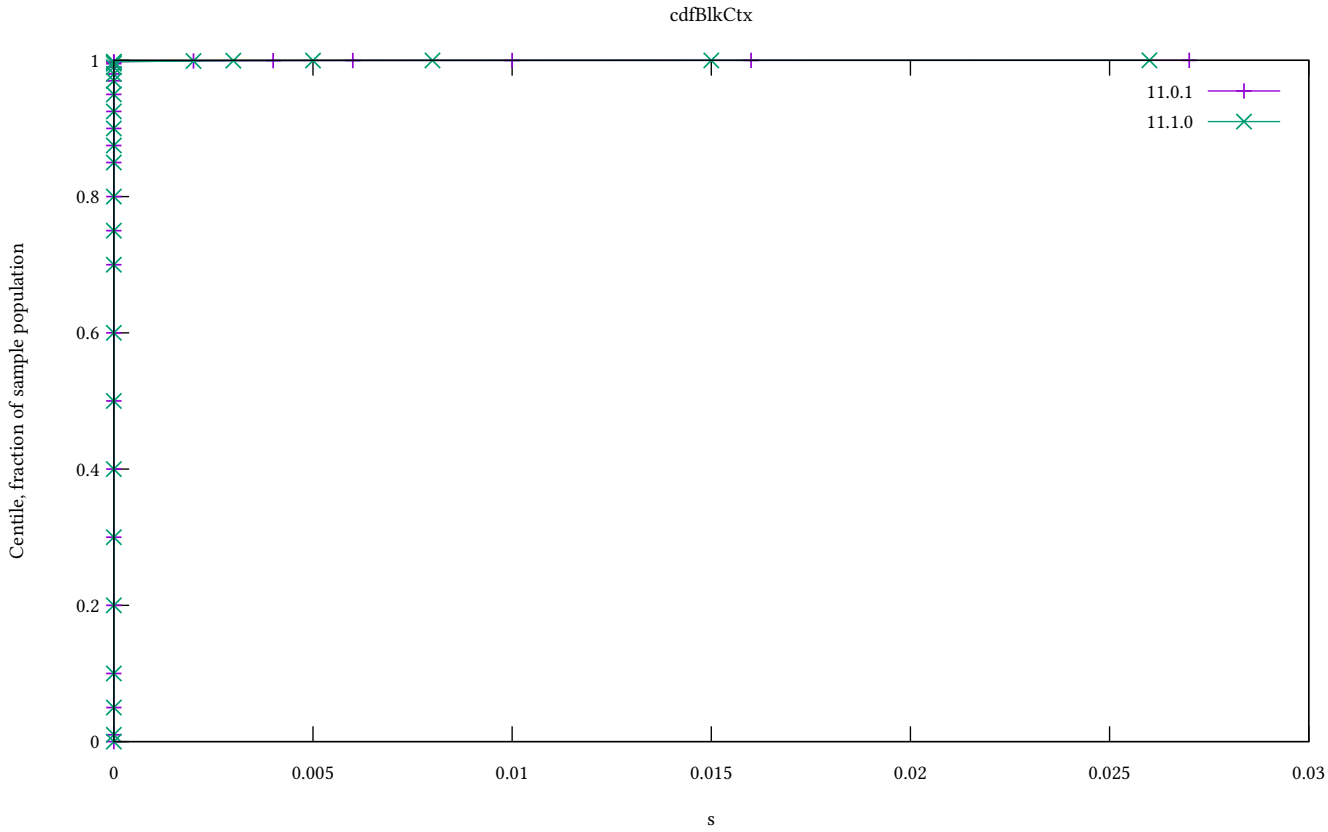


*Figure 11: Block context acquisition delay*

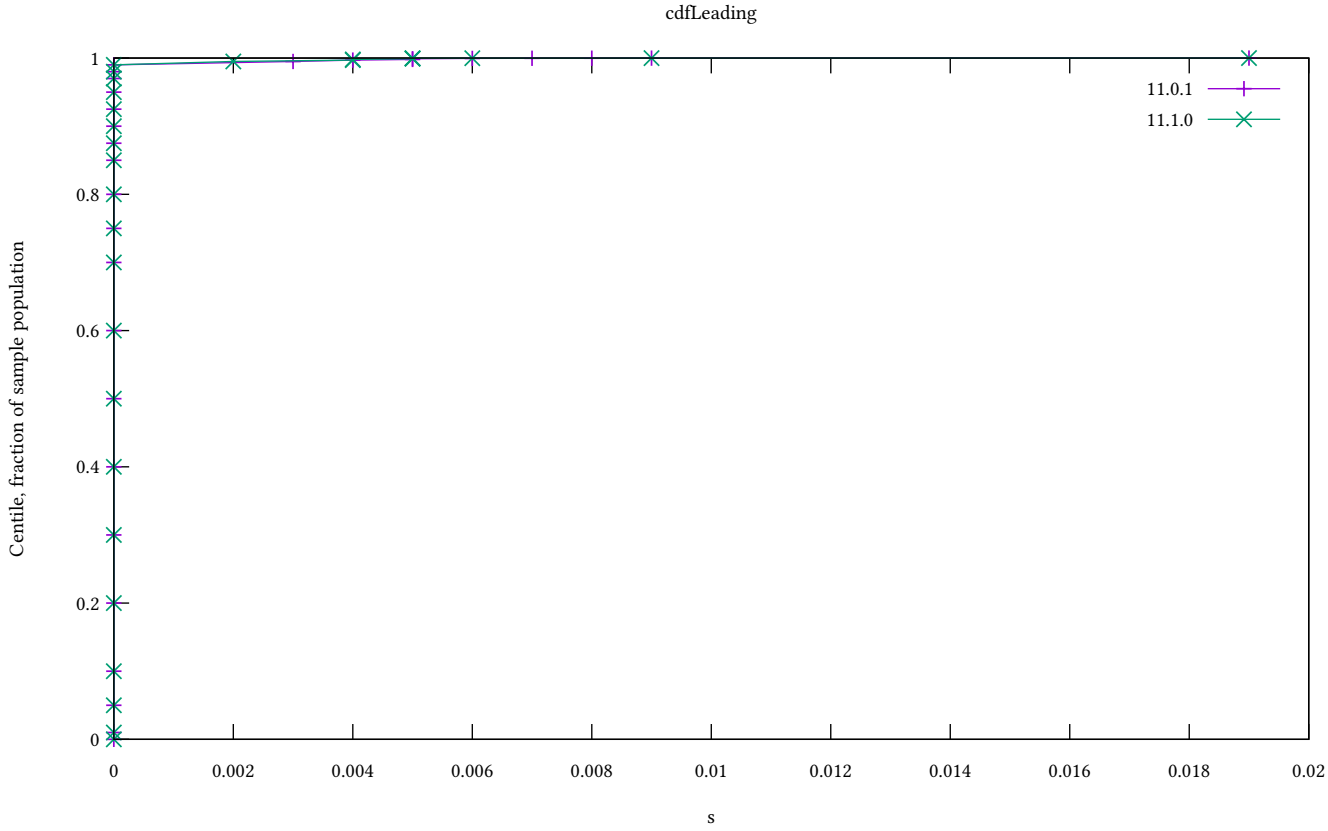


*Figure 12: Leadership check duration*

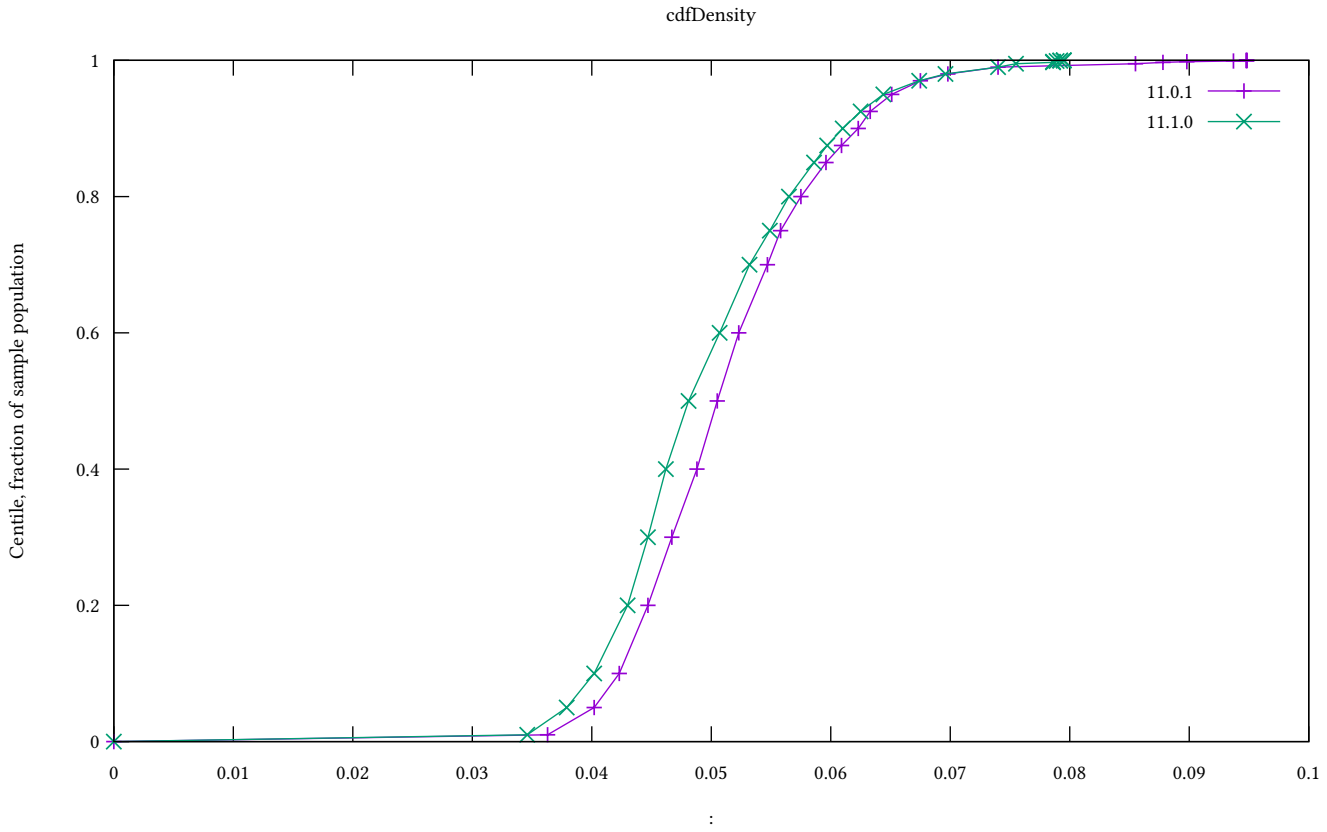


*Figure 13: Chain density*

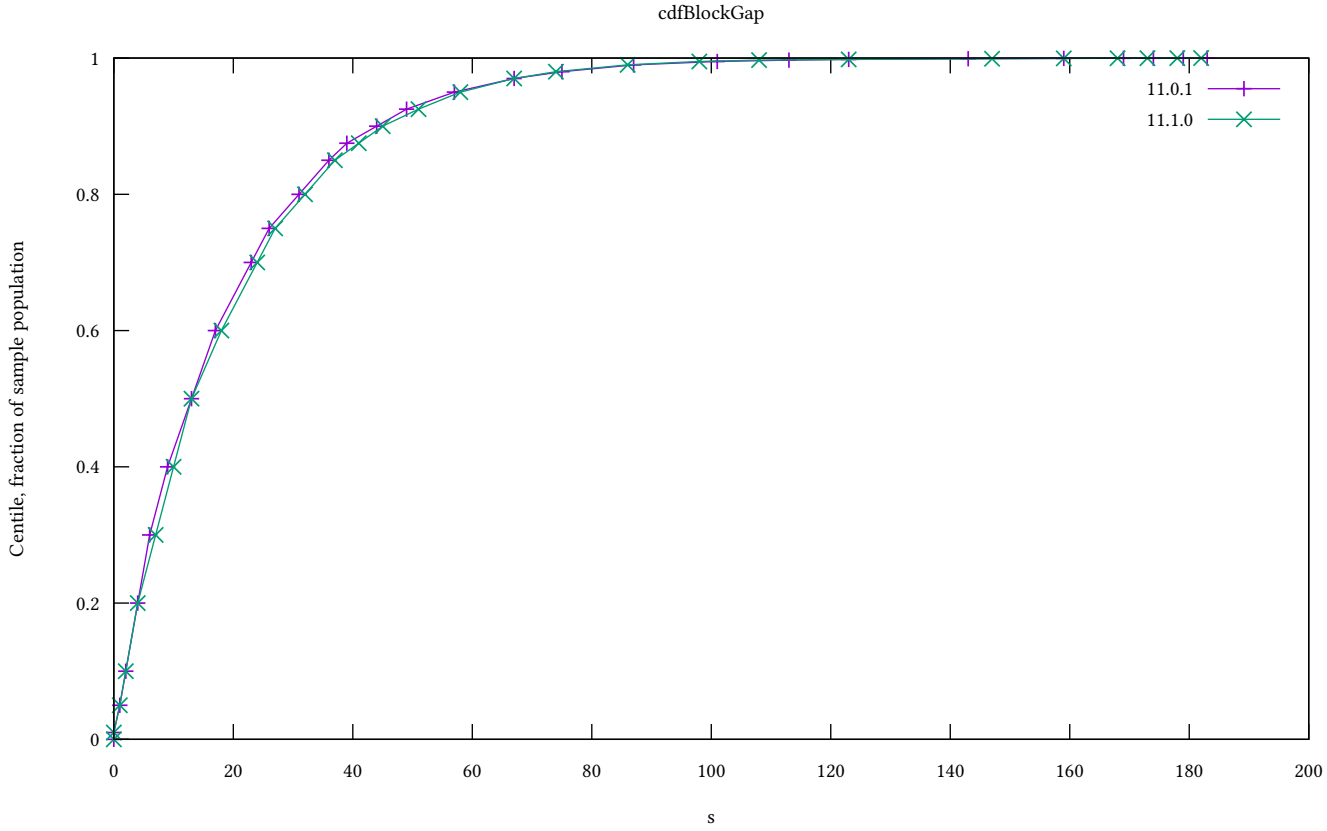


*Figure 14: Interblock gap*

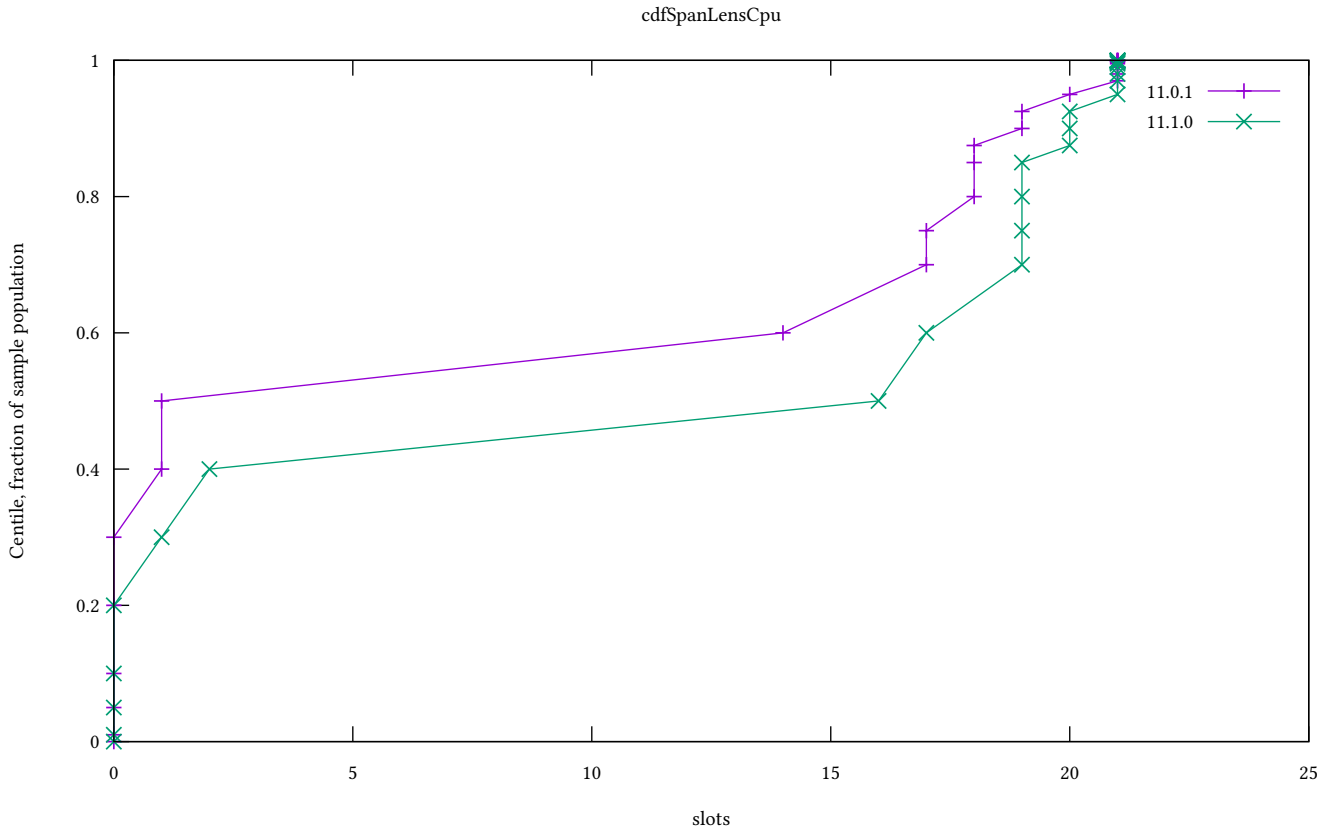


*Figure 15: CPU 85% spans*

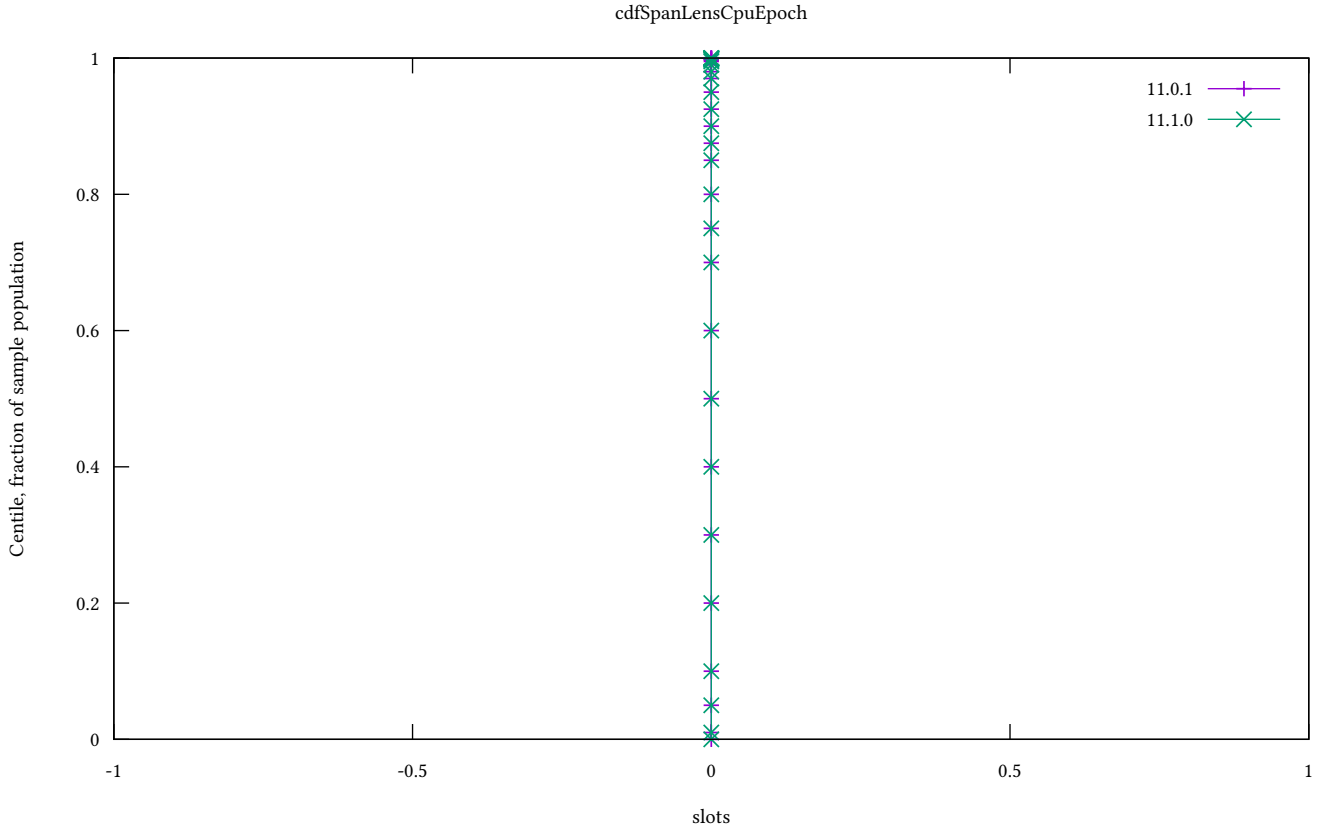


*Figure 16: CPU spans at Ep boundary*

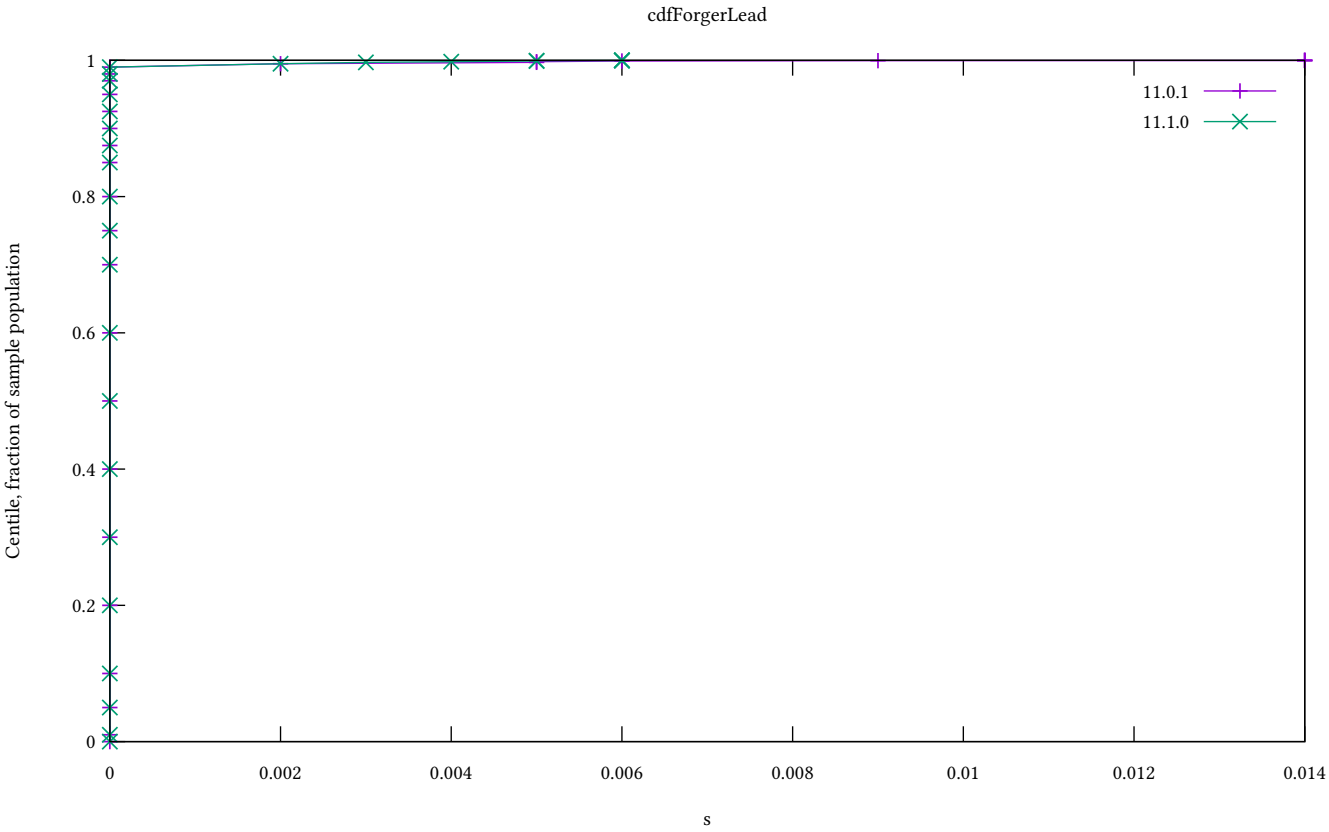


*Figure 17: Leadership check duration*

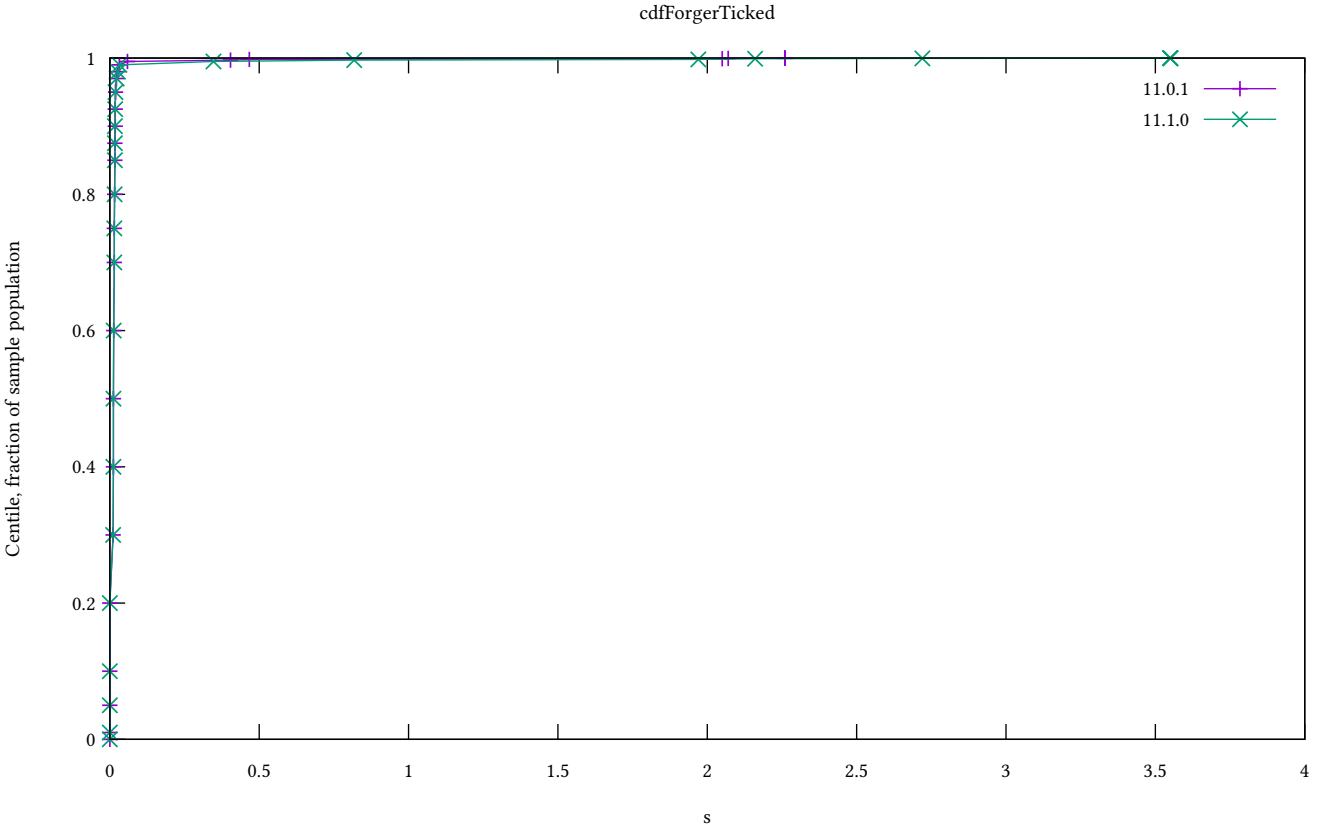


*Figure 18: Ledger ticking*

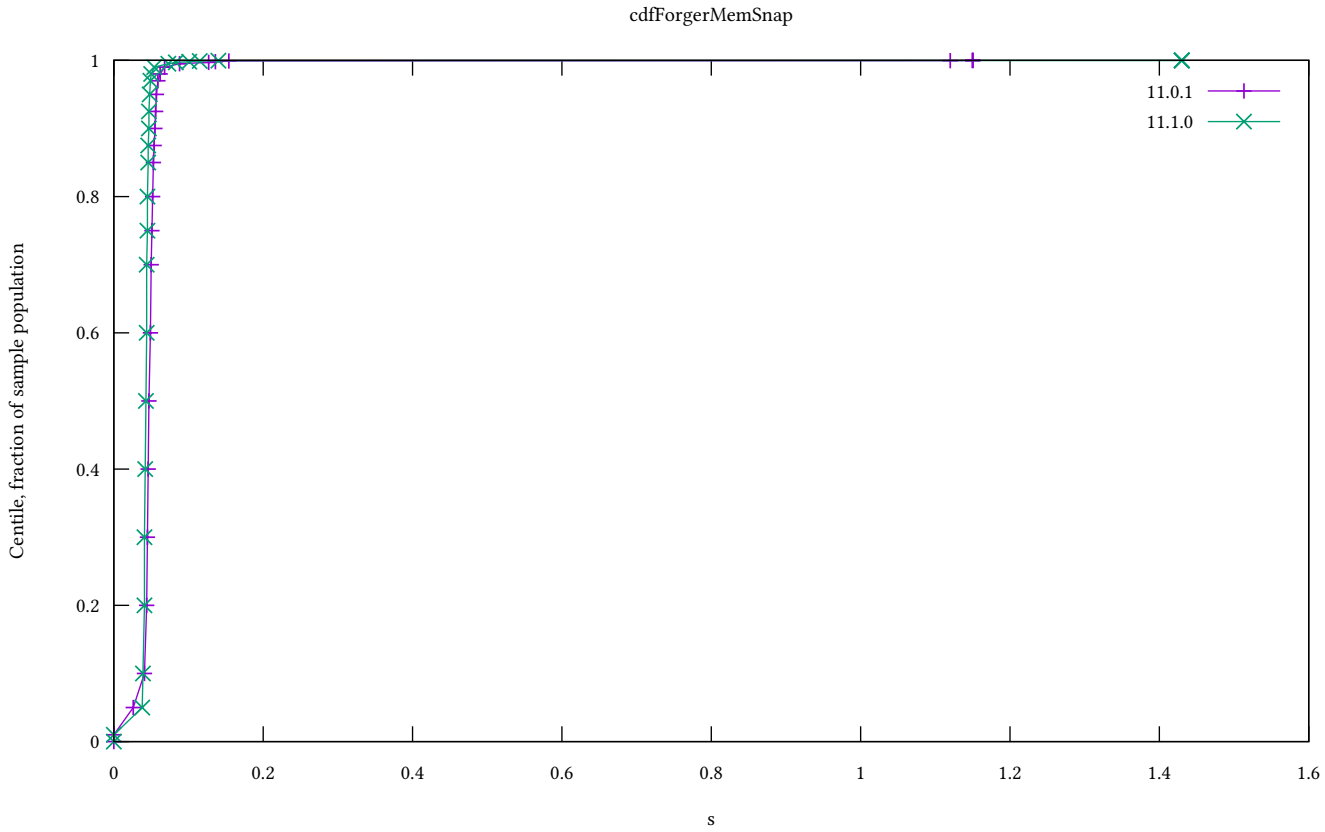


*Figure 19: Mempool snapshotting*

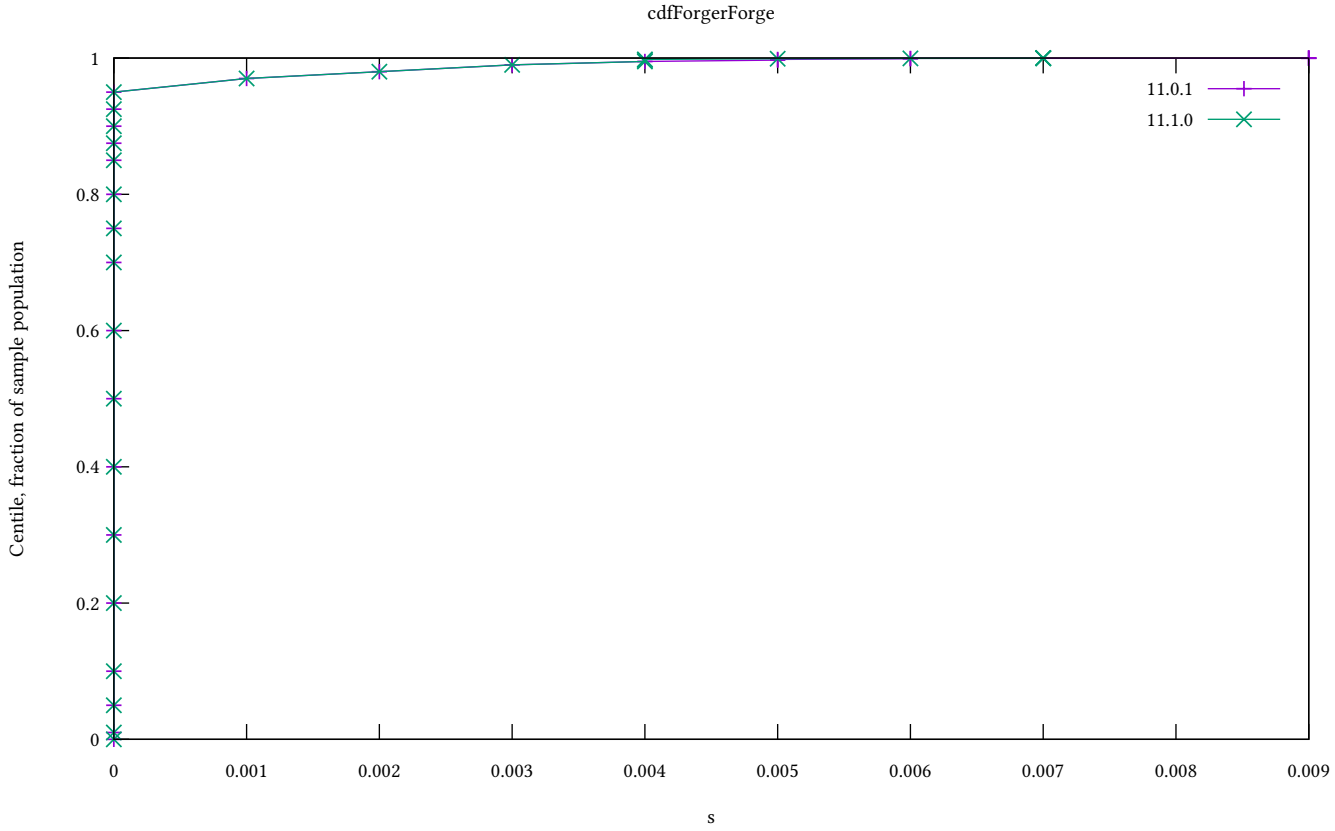


*Figure 20: Leadership to forged*

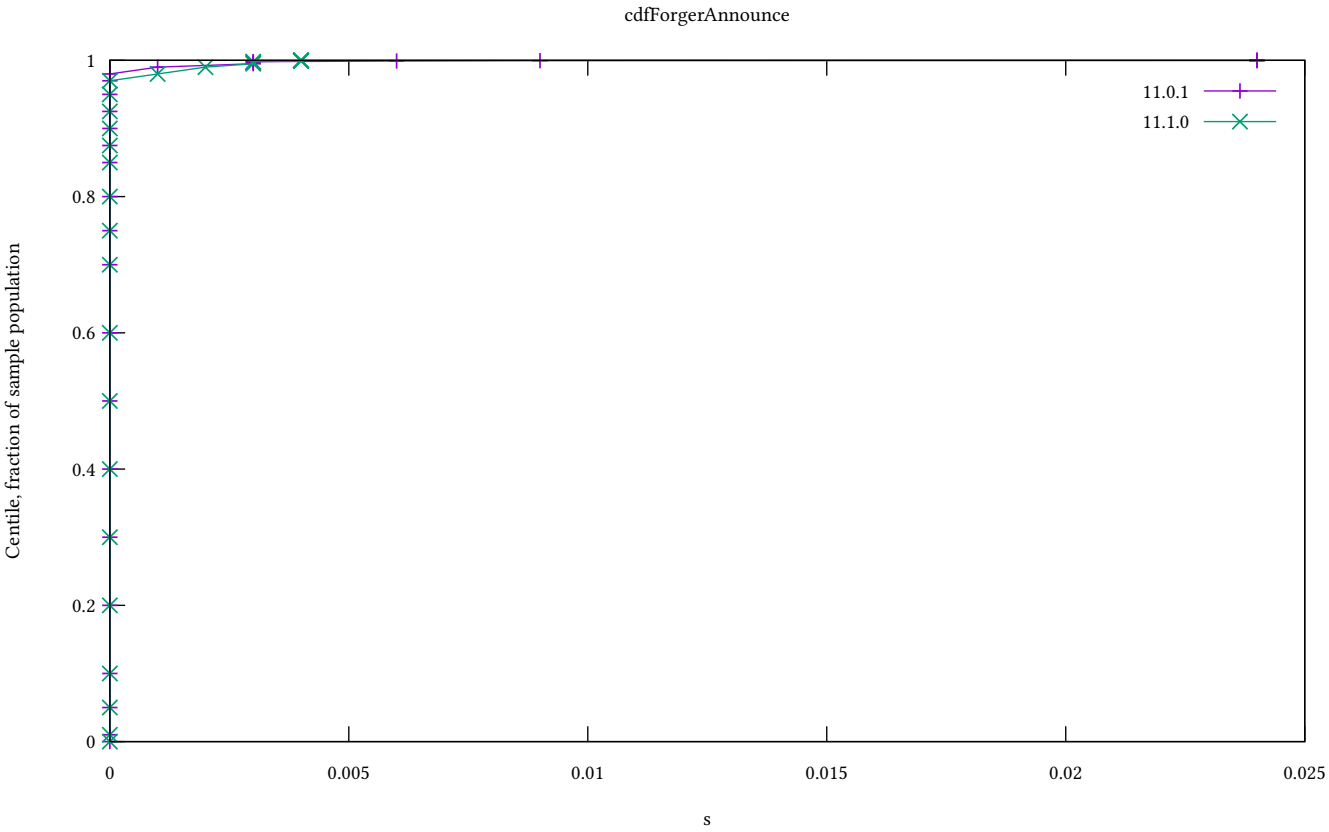


*Figure 21: Forged to announced*

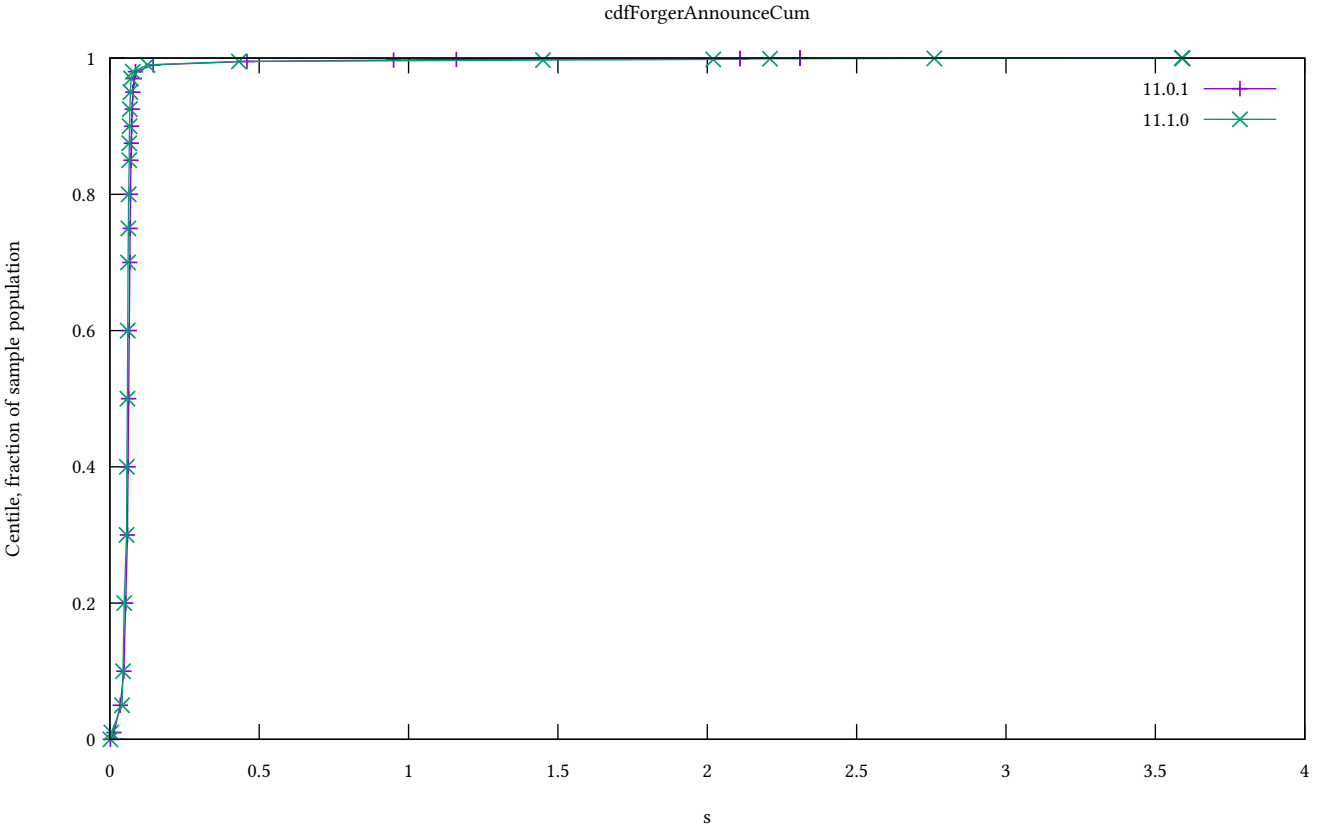


*Figure 22: Slot start to announced*

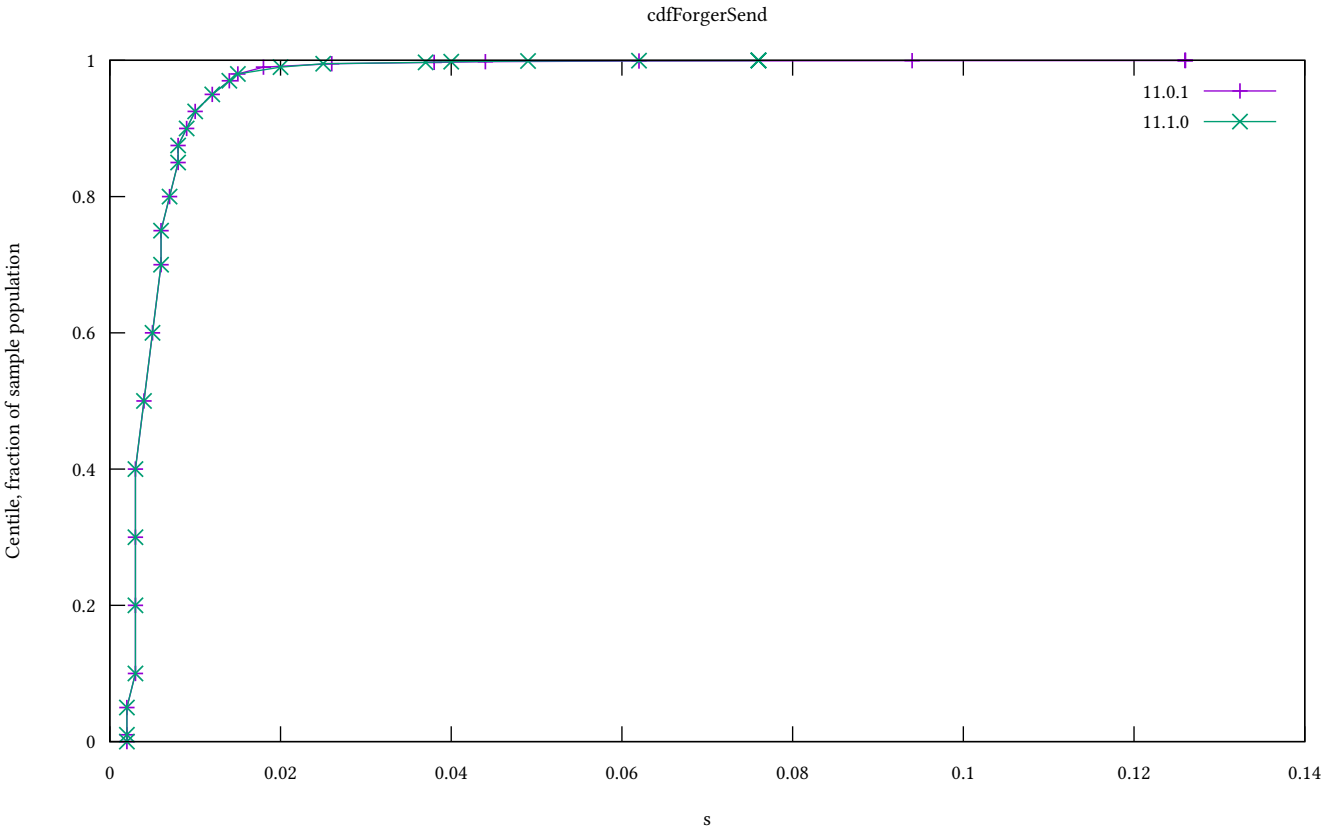


*Figure 23: Forged to sending*

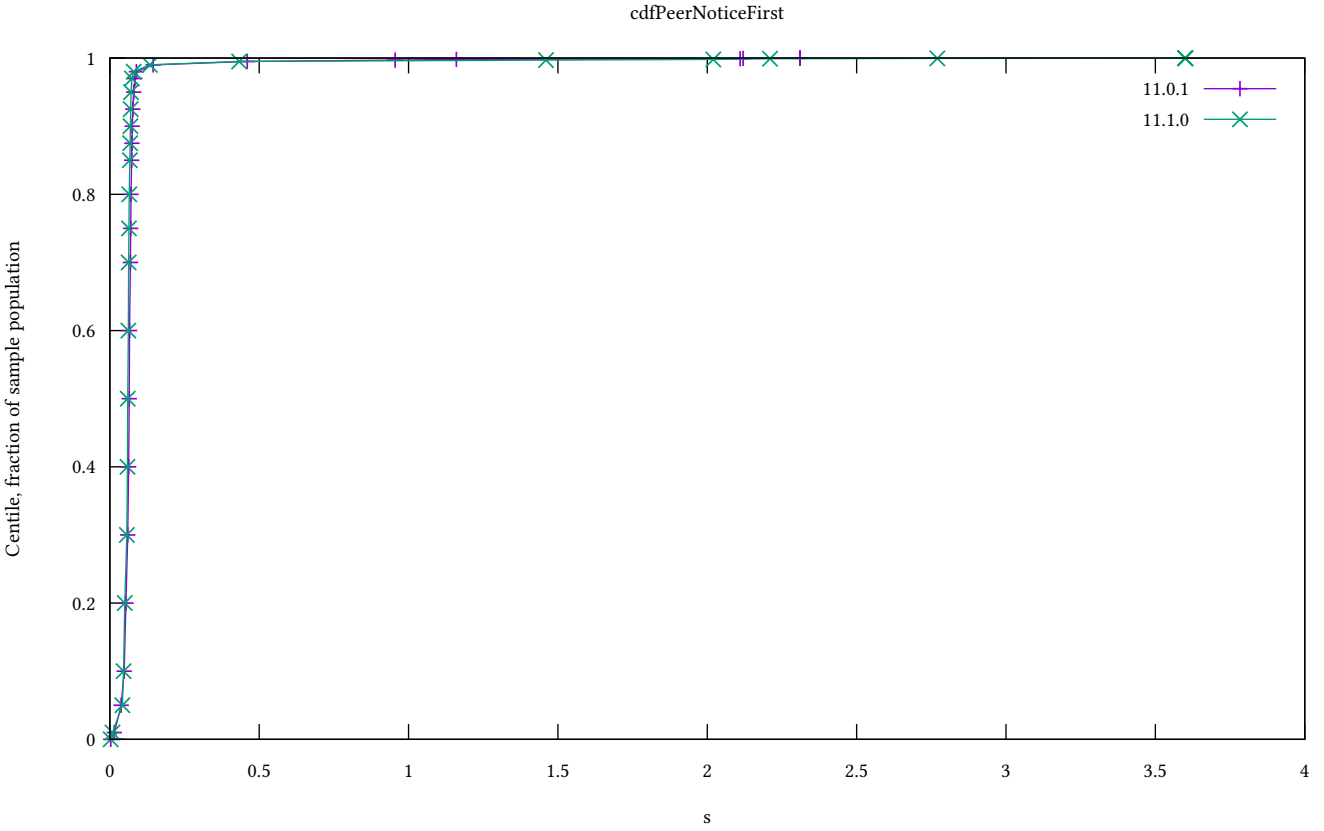


*Figure 24: First peer notice*

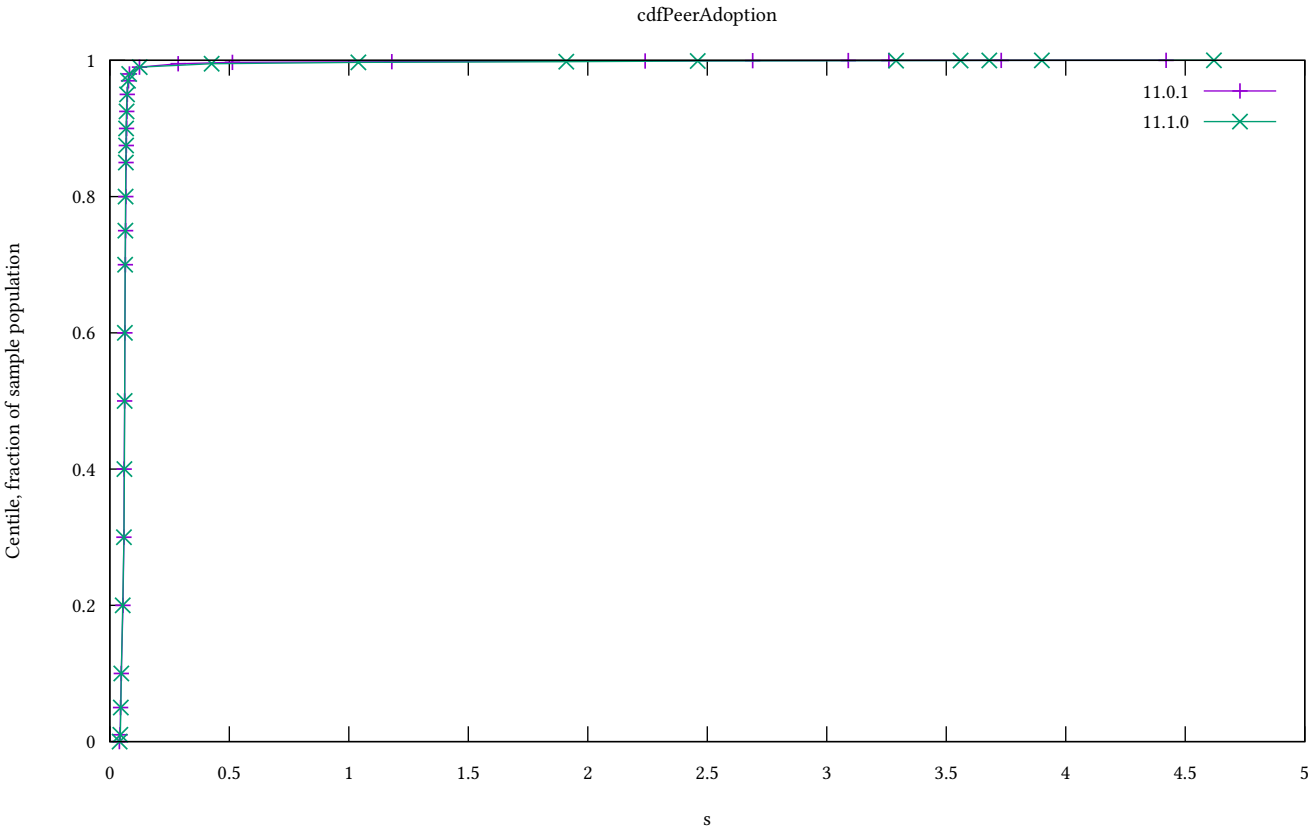


*Figure 25: Fetched to adopted*

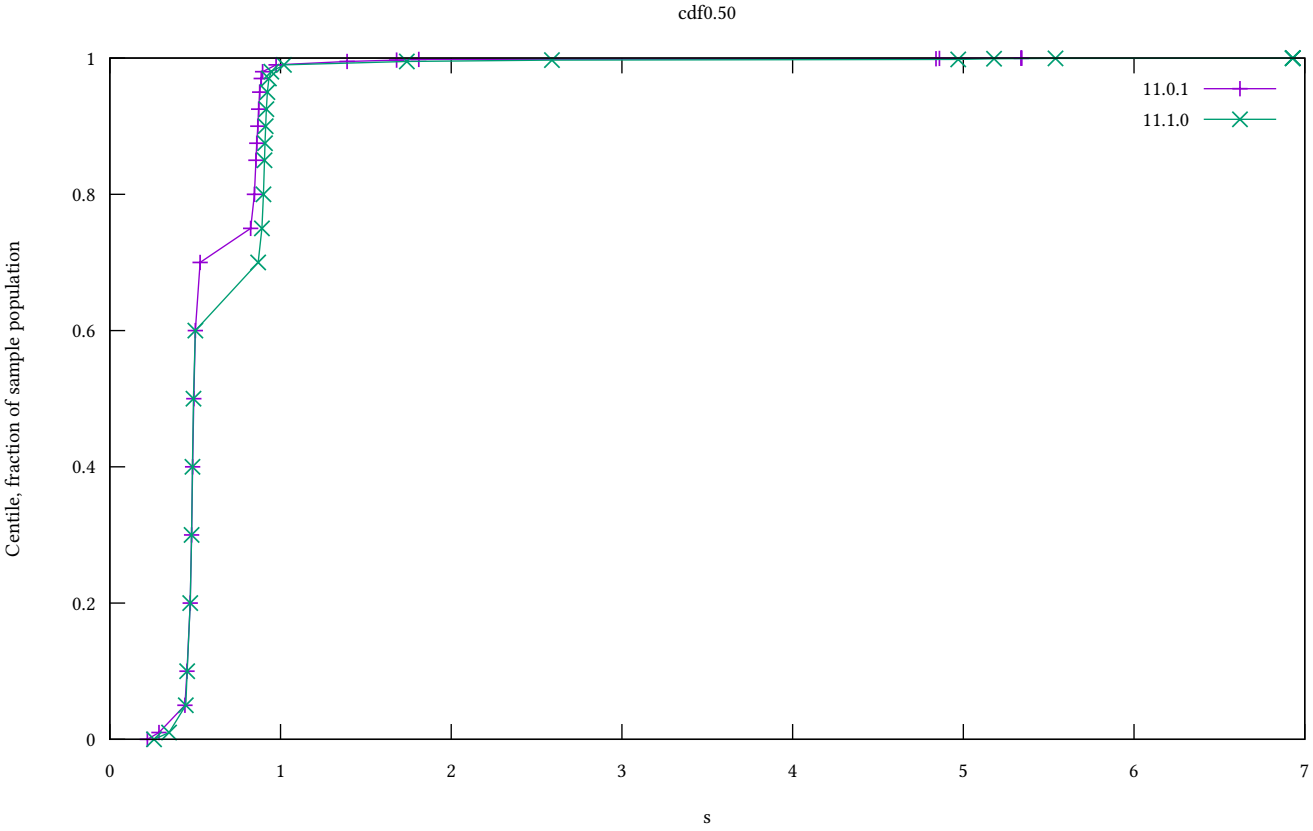


*Figure 26: 0.50 adoption*

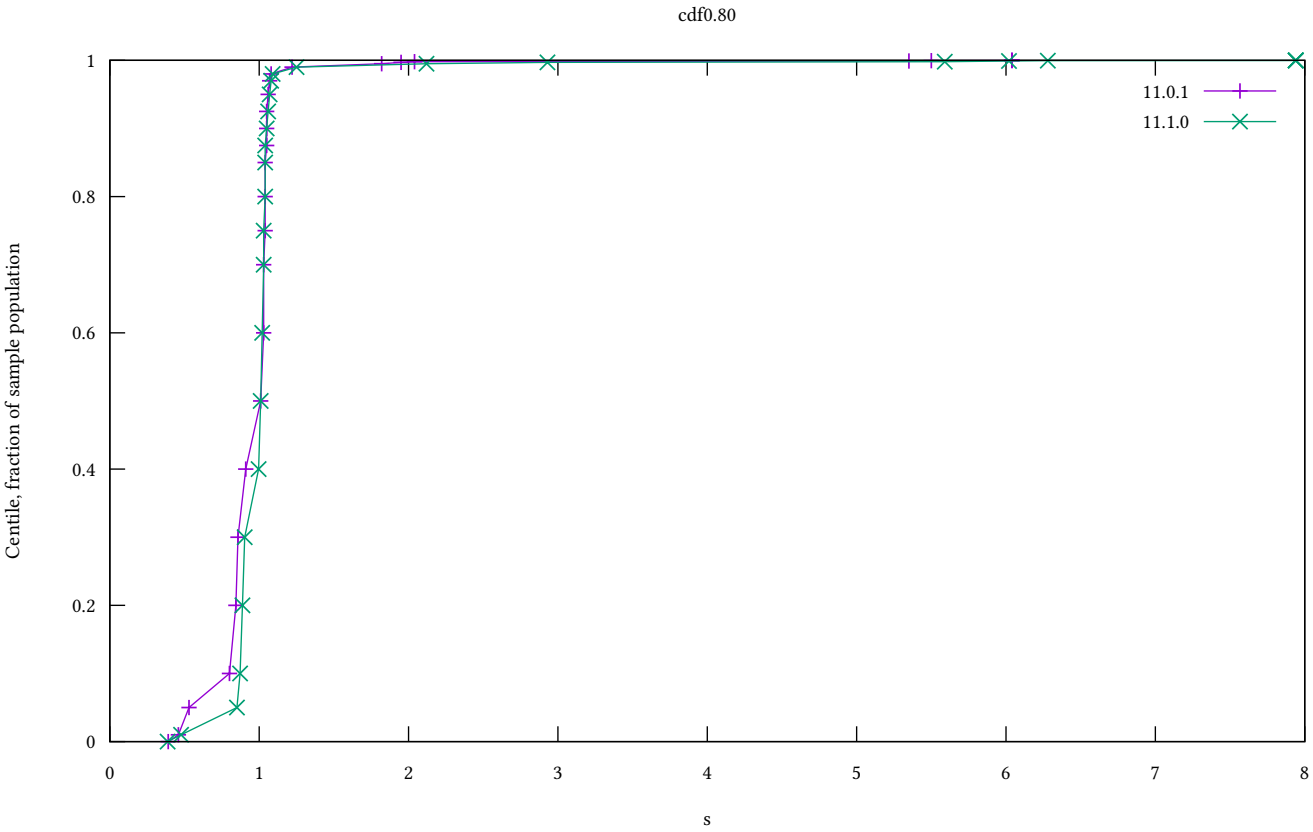


*Figure 27: 0.80 adoption*

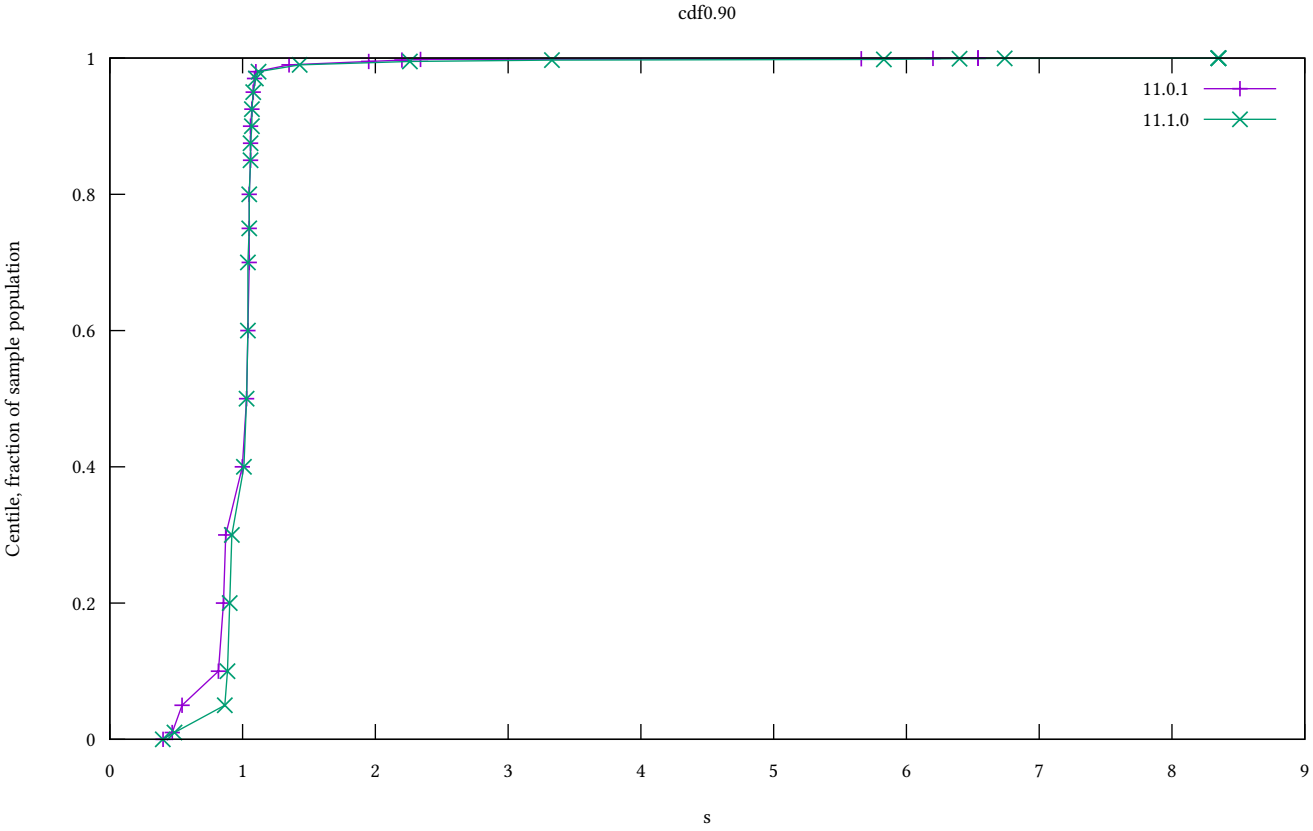


*Figure 28: 0.90 adoption*

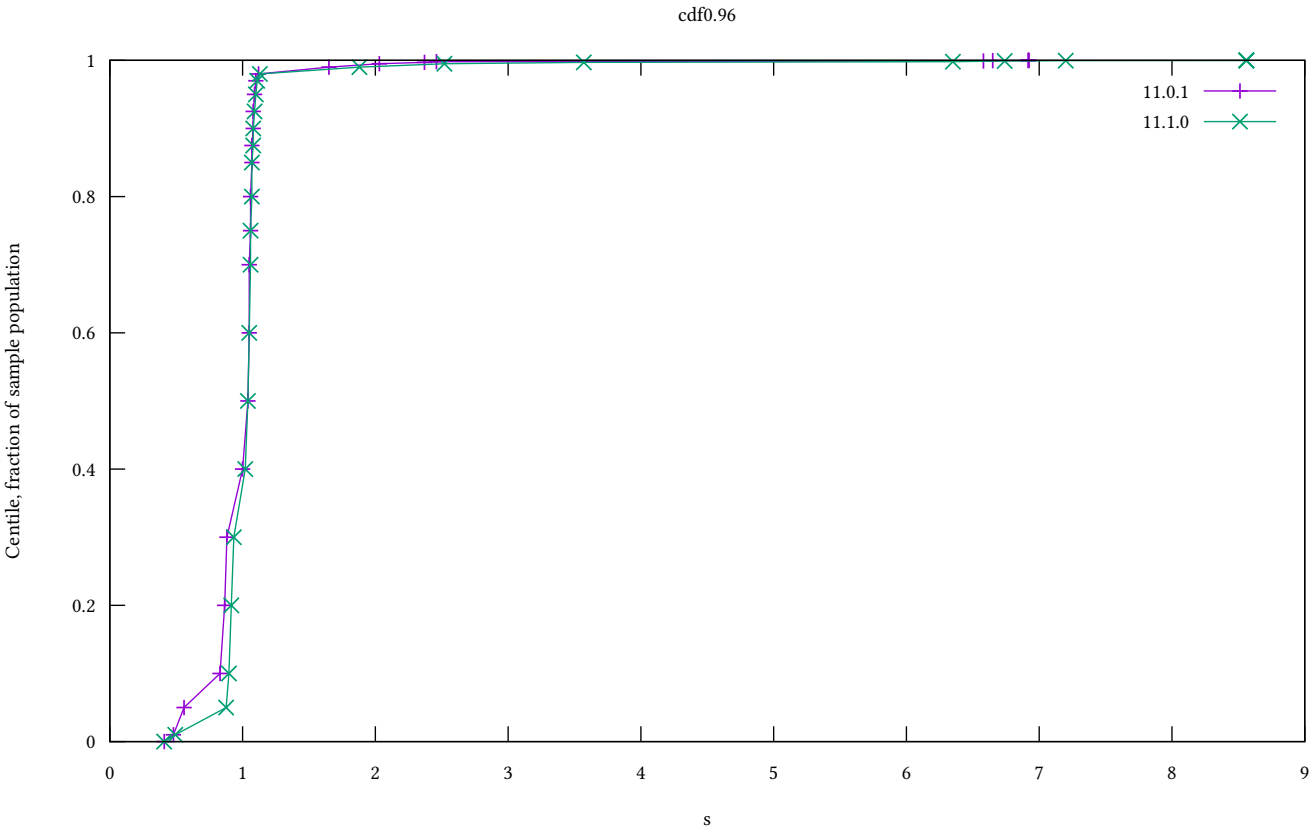


*Figure 29: 0.96 adoption*

# 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