

11.1.1 against 11.0.1-rerun

Plutus countdown loop workload

Federico Mastellone - Performance & Tracing, ICAN Group

2026-09-15

Contents

Manifest 2

Analysis 4

 Resource Usage 4

 Anomaly control 4

 Forging 5

 Individual peer propagation 5

 End-to-end propagation 5

Observations 6

 Resources 6

 Forging 6

 Peer propagation 6

 End-to-end propagation 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 11.0.1-rerun (Conway) and 11.1.1 (Conway) relative to 11.0.1-rerun (Conway), under Plutus countdown loop workload.

	11.0.1-rerun	11.1.1
Analysis date	2026-09-15	2026-09-08
Cluster system start date	2026-09-14	2026-09-07
Cluster system start time	17:16:23	12:34:55
Identifier	11.0.1-rerun	11.1.1
Run batch	1101rerun	11.1.1
GHC version	9.6.7	9.6.7
cardano-node version	11.0.1	11.1.1
ouroboros-consensus version	3.0.1.0	4.2.1.0
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	07d5a1c
ouroboros-consensus git	c87aa76	82ecba3
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	887707.23076	899127.78846
Log objects analysed per host	609773.88461	613405.61538
Host run time, s	71869.4	71945.8
Host log line rate, Hz	12.351	12.497
Total log objects analysed	31708242	31897092
Run time, s	71874	71959
Analysed run duration, s	56020	56039
Run time efficiency	0.77	0.77
Node start spread, s	4.8137800	22.189944
Node stop spread, s	3.1247031	2.9032723
Slots analysed	56018	56037
Blocks analysed	2644	2749
Blocks rejected	860	873

Analysis

Resource Usage

	11.0.1-rerun	11.1.1	Δ	$\Delta\%$
Forge loop starts, units	0.99996	0.99996	0.000	0
Process CPU usage, %	1.1441	1.1309	-0.013	-1
RTS GC CPU usage, %	0.229	0.2151	-0.014	-6
RTS Mutator CPU usage, %	0.9135	0.9138	0.000	0.0
Major GCs, events	0.0009	0.0008	-0.000	0
Minor GCs, events	0.3323	0.3362	0.004	1
Kernel RSS, MiB	6598.2	7134.6	536.442	8
RTS heap size, MiB	6529.2	7064.4	535.259	8
RTS live GC dataset, MiB	2649.3	3514.7	865.357	33
RTS alloc rate, MiB/s	9.8341	9.9432	0.109	1
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	208.16	208.91	0.743	0.4
CPU 85% spans, slots	7.482	7.9003	0.418	6

Anomaly control

	11.0.1-rerun	11.1.1	Δ	$\Delta\%$
Blocks per host, blocks	69.115	71.154	2.038	3
Filtered to chained blocks, :	0.7563	0.7585	0.002	0.3
Chained to forged blocks, :	0.974	0.9791	0.005	0.5
Height & slot battles, blocks	0.0	0.00073	0.001	NaN
Block size, Bytes	2996.	2996.	0.003	8.5e-5

Forging

	11.0.1-rerun	11.1.1	Δ	$\Delta\%$
Started forge loop iteration, s	0.0012	0.00093	-0.000	-22
Acquired block context, s	6.3e-5	6.3e-5	-0.000	0
Acquired ledger state, s	0.00011	0.00011	0.000	0
Acquired ledger view, s	3.3e-5	3.0e-5	-0.000	0
Leadership check duration, s	0.00042	0.00042	0.000	0
Ledger ticking, s	0.01886	0.01794	-0.001	-5
Mempool snapshotting, s	0.00168	0.00147	-0.000	-12
Leadership to forged, s	0.00022	0.00021	-0.000	0
Forged to announced, s	0.00057	0.00055	-0.000	0
Forged to sending, s	0.00495	0.00486	-0.000	0
Forged to self-adopted, s	0.04136	0.03993	-0.001	-3
Slot start to announced, s	0.02315	0.02173	-0.001	-6

Individual peer propagation

	11.0.1-rerun	11.1.1	Δ	$\Delta\%$
First peer notice, s	0.0247	0.02327	-0.001	-6
First peer fetch, s	0.03061	0.02884	-0.002	-6
Notice to fetch request, s	0.00129	0.00128	-0.000	0
Fetch duration, s	0.13327	0.13242	-0.001	-0.6
Fetches to announced, s	0.00068	0.00067	-0.000	0
Fetches to sending, s	0.04309	0.04187	-0.001	-3
Fetches to adopted, s	0.04253	0.04116	-0.001	-3

End-to-end propagation

	11.0.1-rerun	11.1.1	Δ	$\Delta\%$
0.50 adoption, s	0.28353	0.27694	-0.007	-2
0.80 adoption, s	0.4809	0.47793	-0.003	-0.6
0.90 adoption, s	0.49235	0.4891	-0.003	-0.7
0.92 adoption, s	0.49509	0.49247	-0.003	-0.5
0.94 adoption, s	0.49813	0.49548	-0.003	-0.5
0.96 adoption, s	0.50184	0.4993	-0.003	-0.5
0.98 adoption, s	0.50738	0.50533	-0.002	-0.4
1.00 adoption, s	0.52967	0.53179	0.002	0.4

Observations

Resources

1. Process CPU usage is practically unchanged (-1%); GC CPU usage decreases by 6% and Major GCs by 11%, while Mutator CPU usage, allocation rate and Minor GCs are unchanged.
2. Kernel RSS and RTS heap size increase by 0.52 GiB or 8%; the RTS live GC dataset increases by 0.85 GiB or 33%.
3. CPU 85% spans lengthen by 0.4 slots or 6%; node start spread is 22.2 s against 4.8 s.

Forging

1. Block production timings are practically unchanged: the largest move, slot start to announced, is a near-jitter 1.4 ms or 6% earlier into the slot, and every other step is under 1 ms.

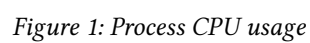
Peer propagation

1. Block Fetch duration is unchanged (0.9 ms or 0.6% faster).
2. The remaining peer-to-peer steps are unchanged too, the largest being a near-jitter 1.8 ms or 6% on first peer fetch.

End-to-end propagation

1. Cluster adoption is unchanged: the median improves by 6.6 ms or 2%, and the 80th centile and above move by under 3.5 ms, or under 1%.

Cluster performance charts



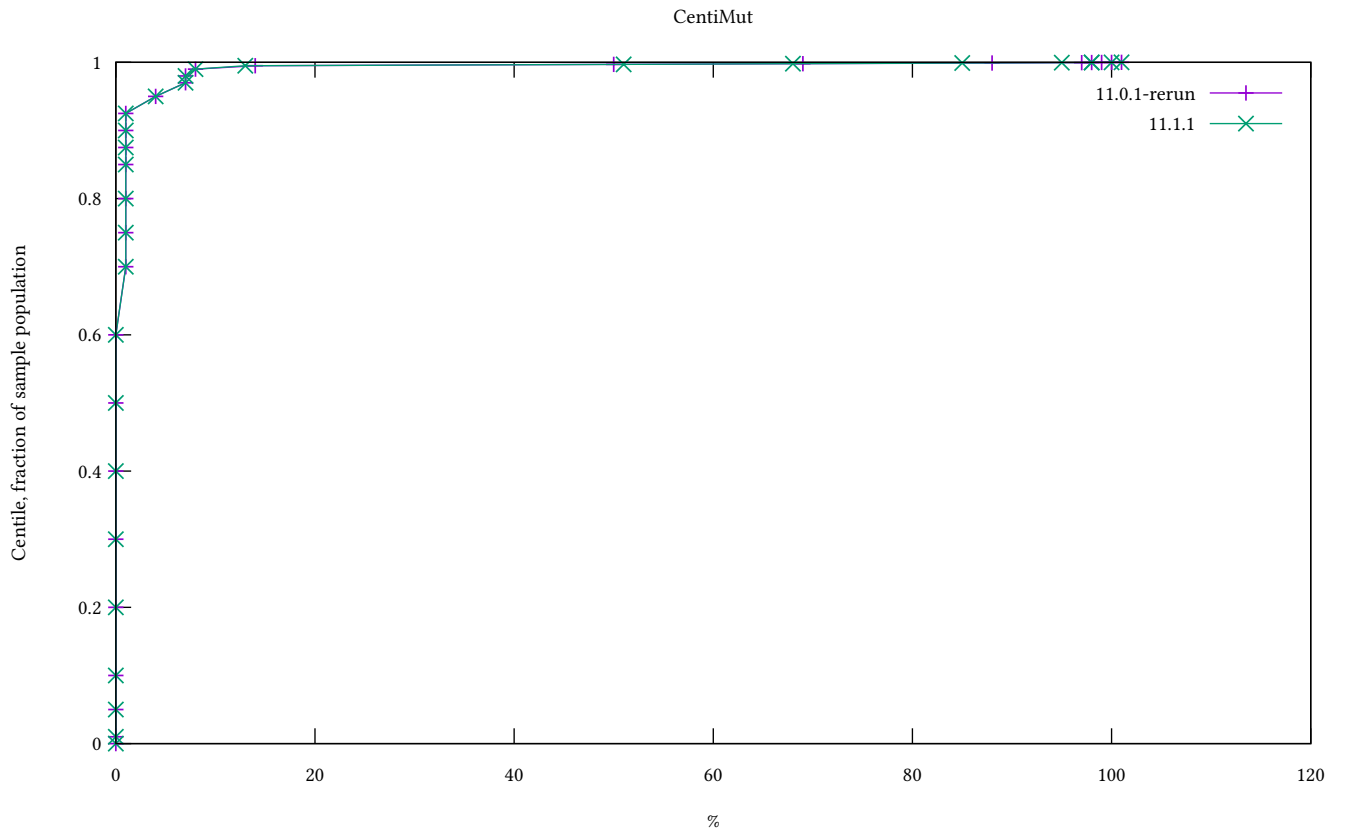


Figure 3: RTS Mutator CPU usage

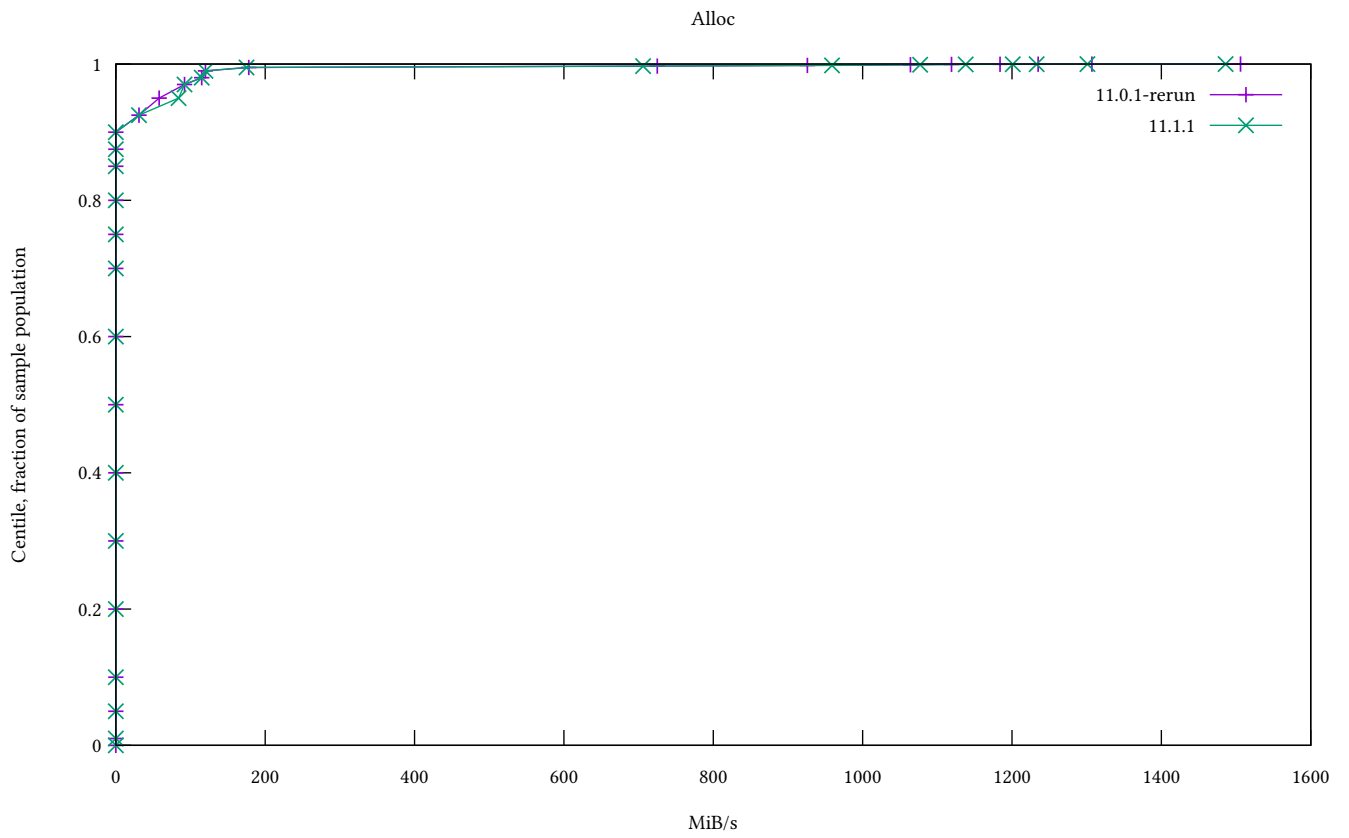


Figure 4: RTS alloc rate

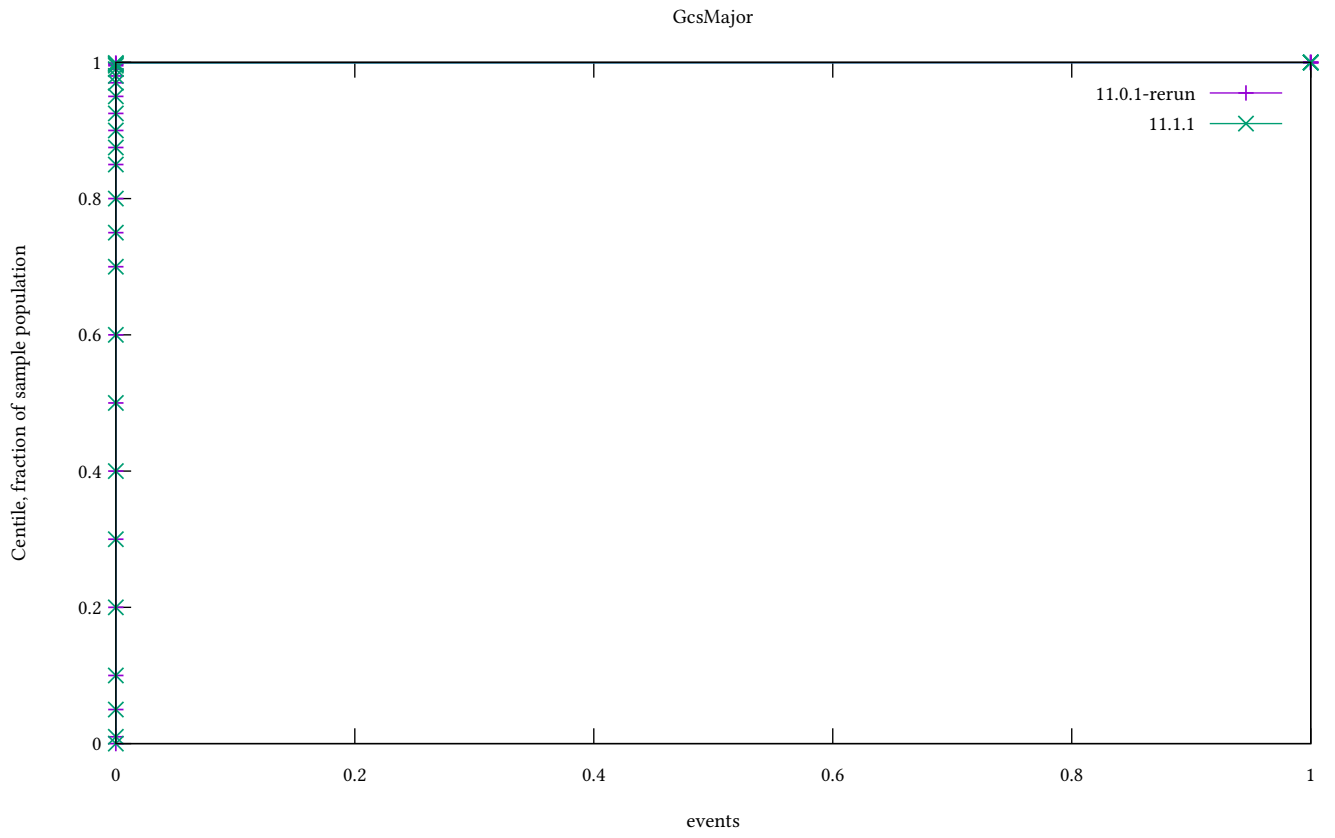


Figure 5: Major GCs

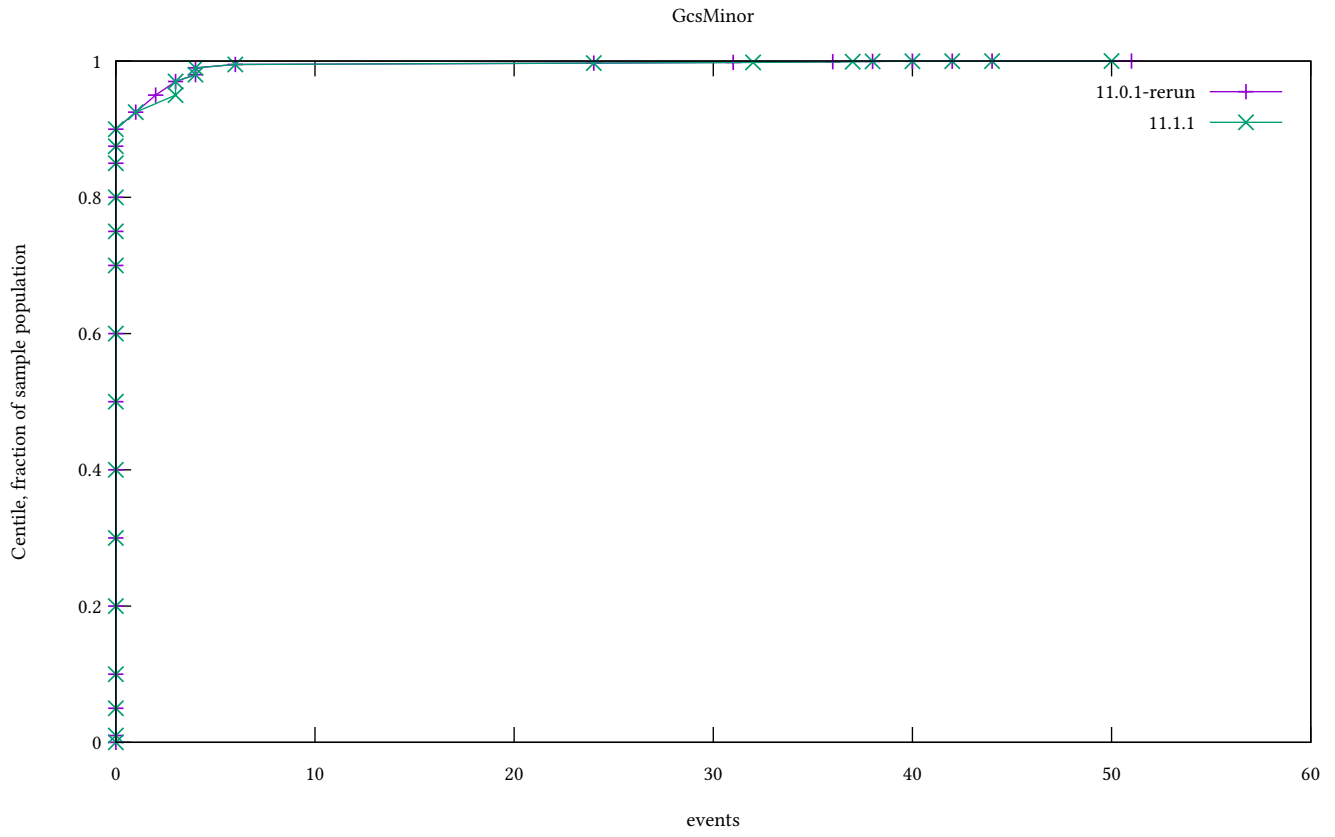


Figure 6: Minor GCs

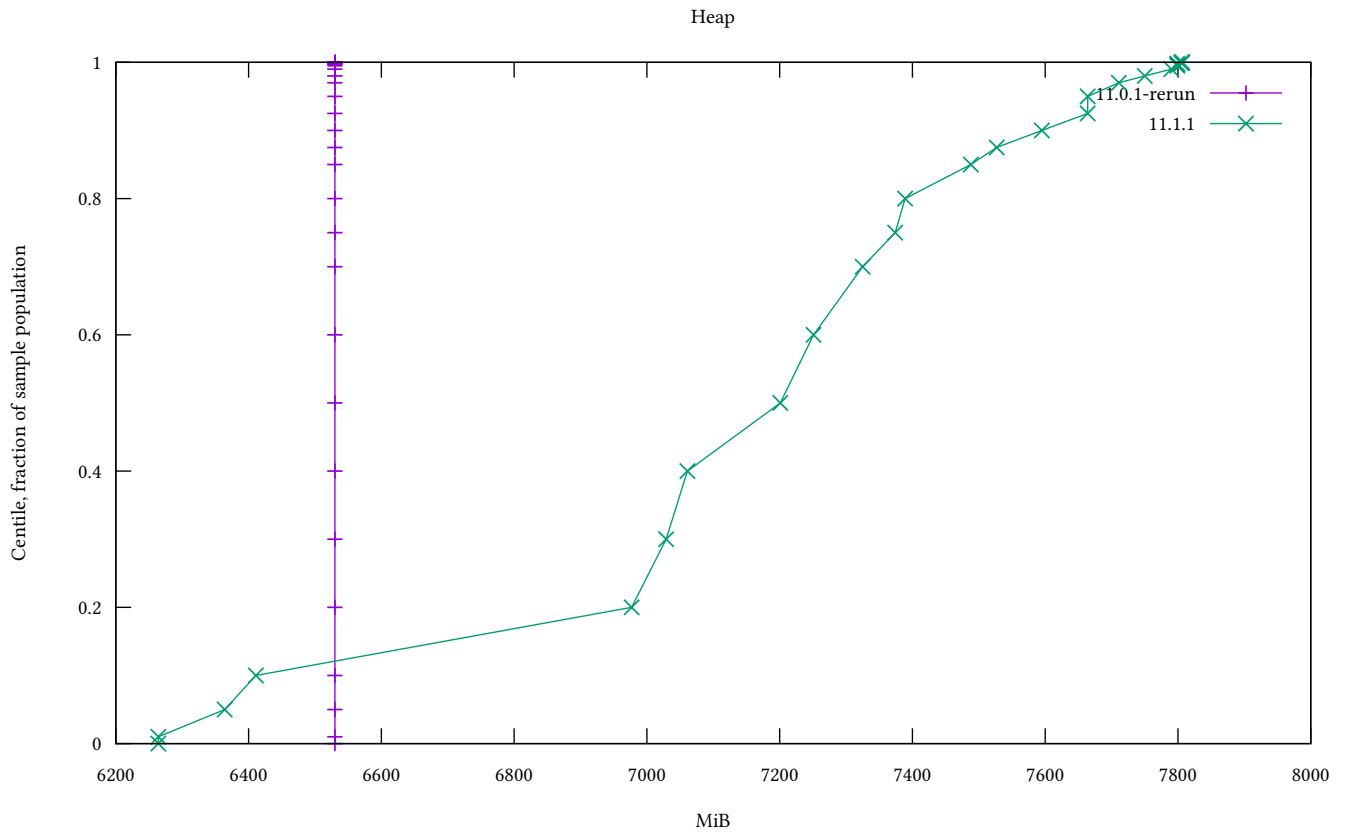


Figure 7: RTS heap size

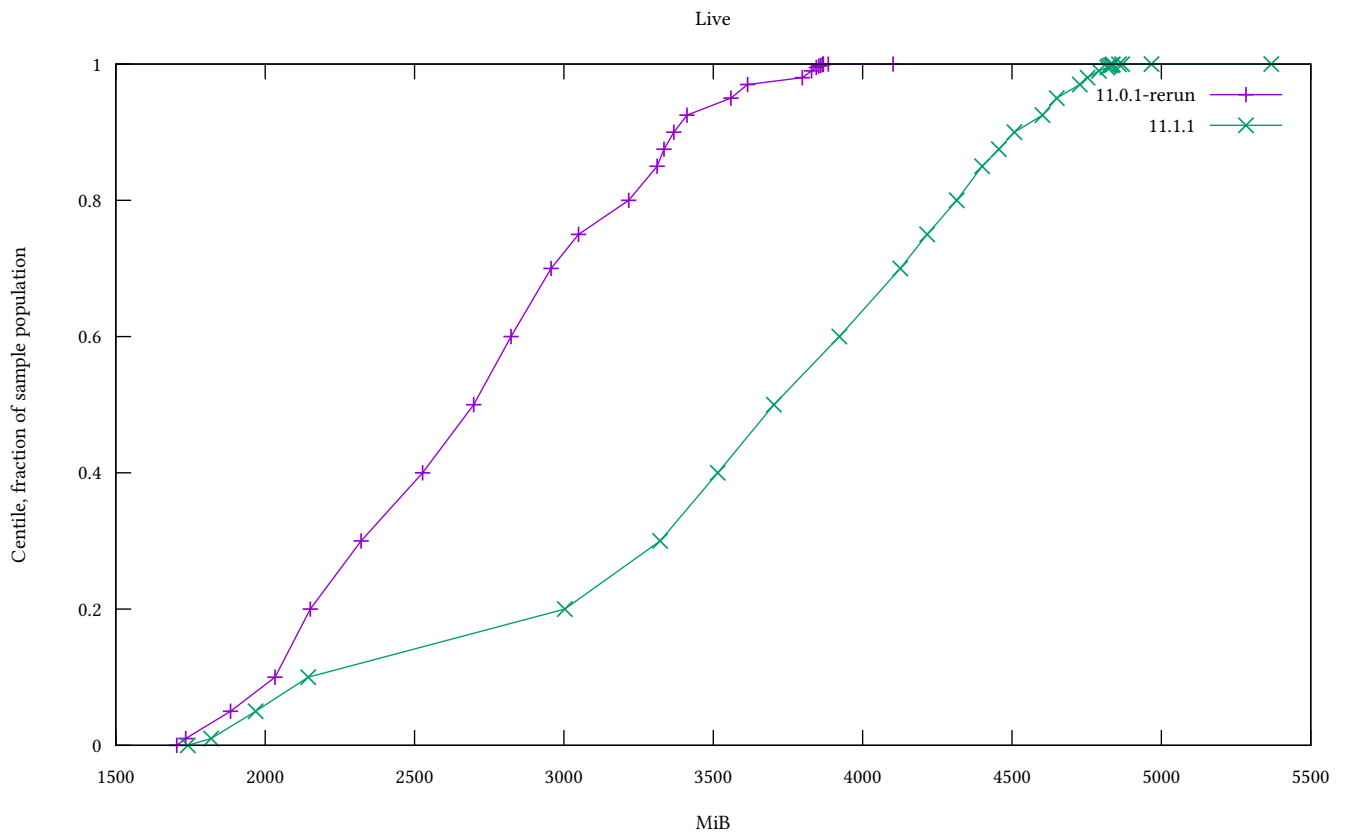


Figure 8: RTS live GC dataset

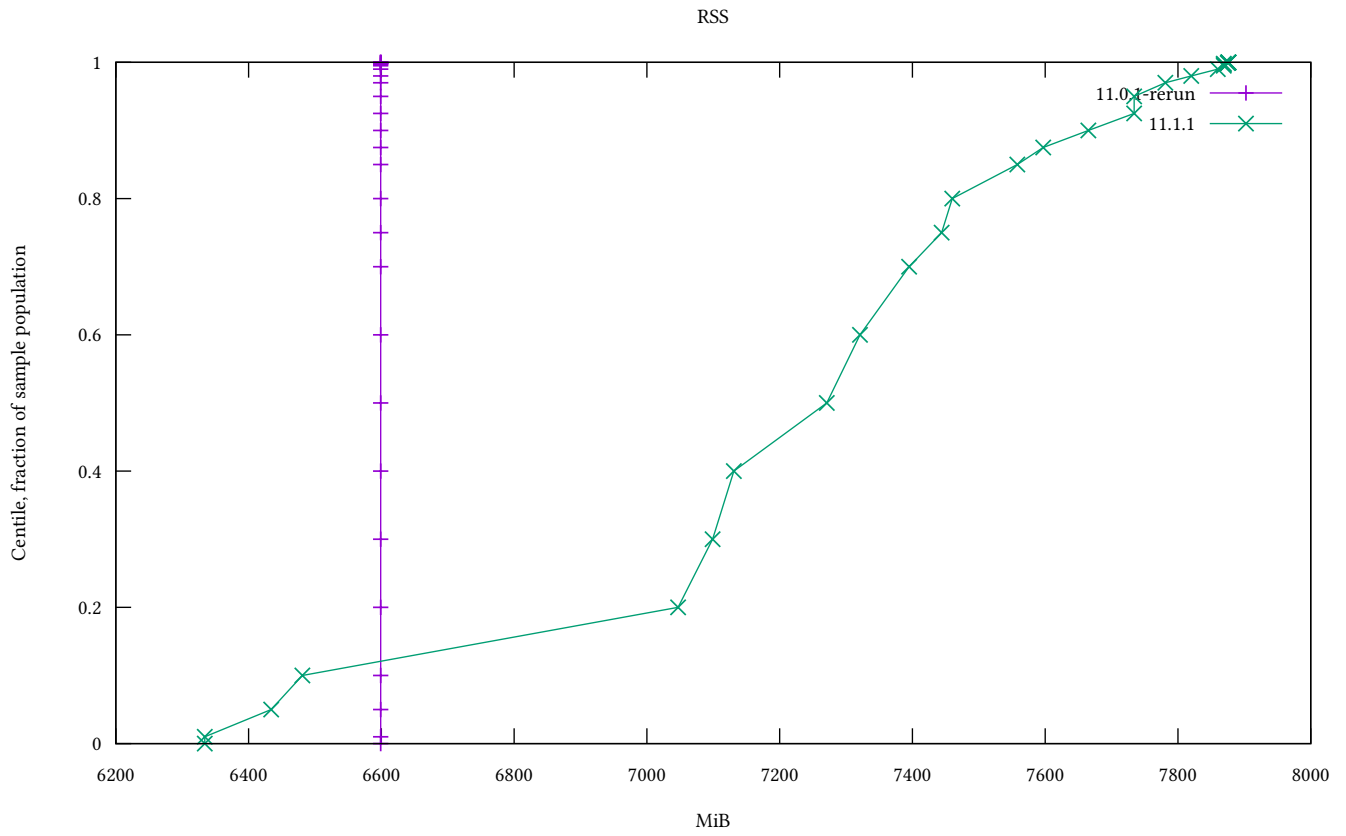


Figure 9: Kernel RSS

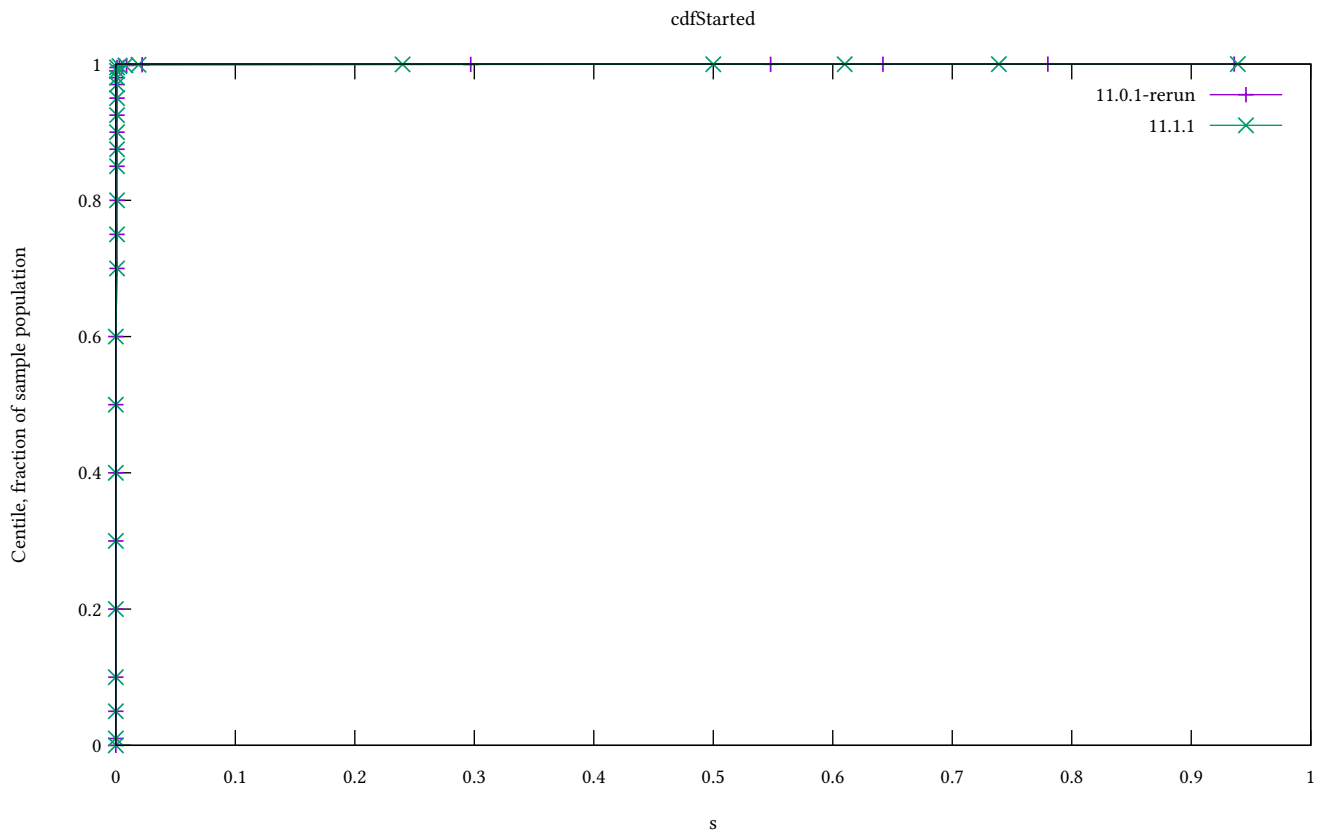


Figure 10: Forge loop tardiness

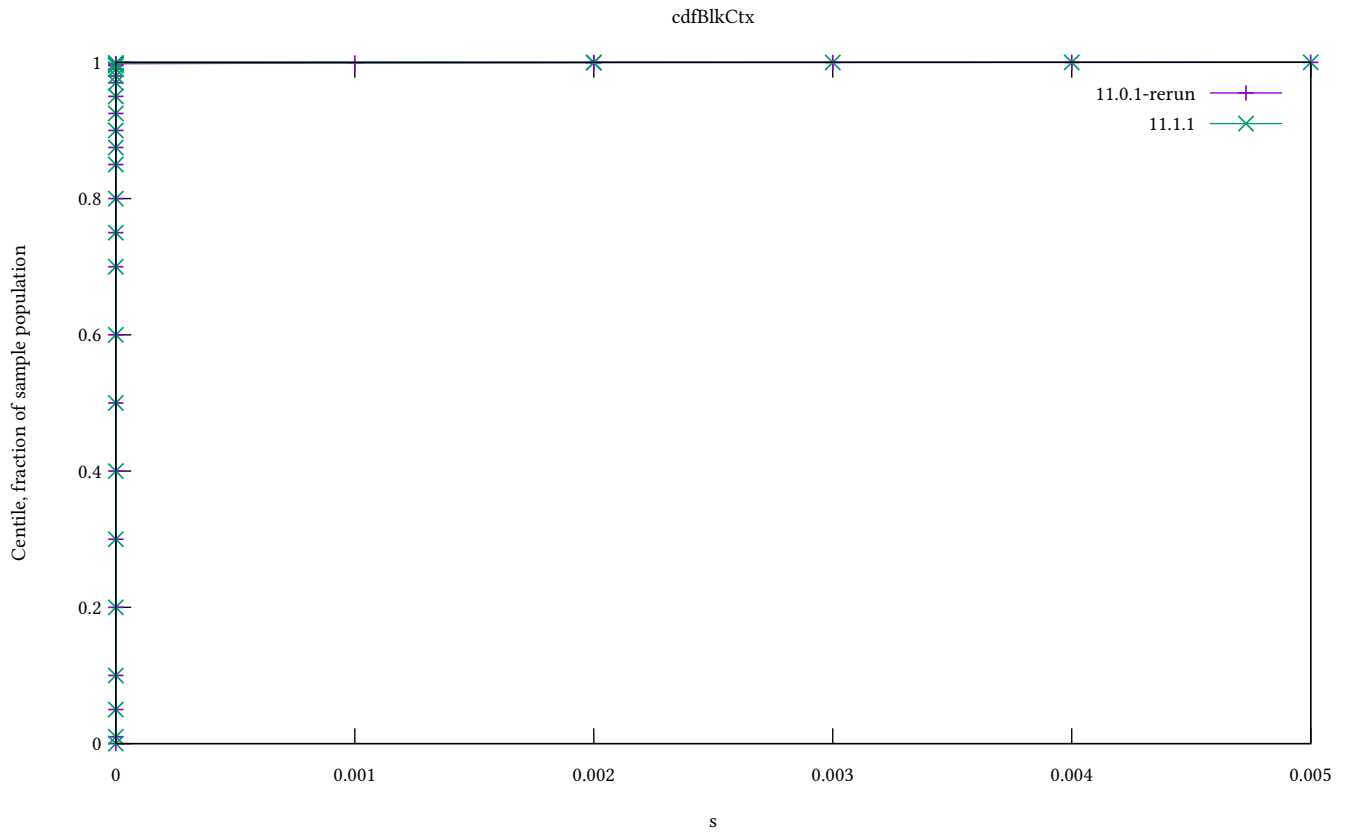


Figure 11: Block context acquisition delay

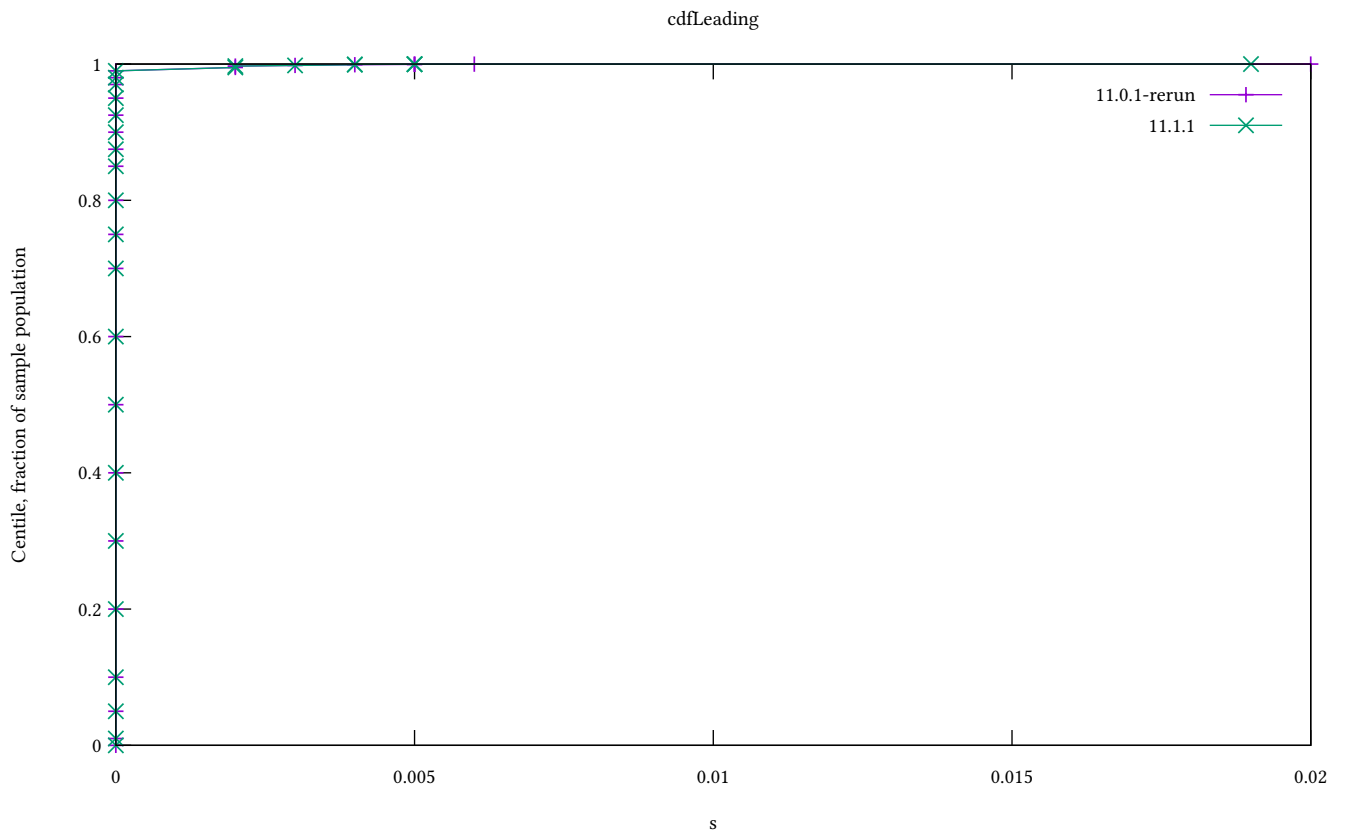


Figure 12: Leadership check duration

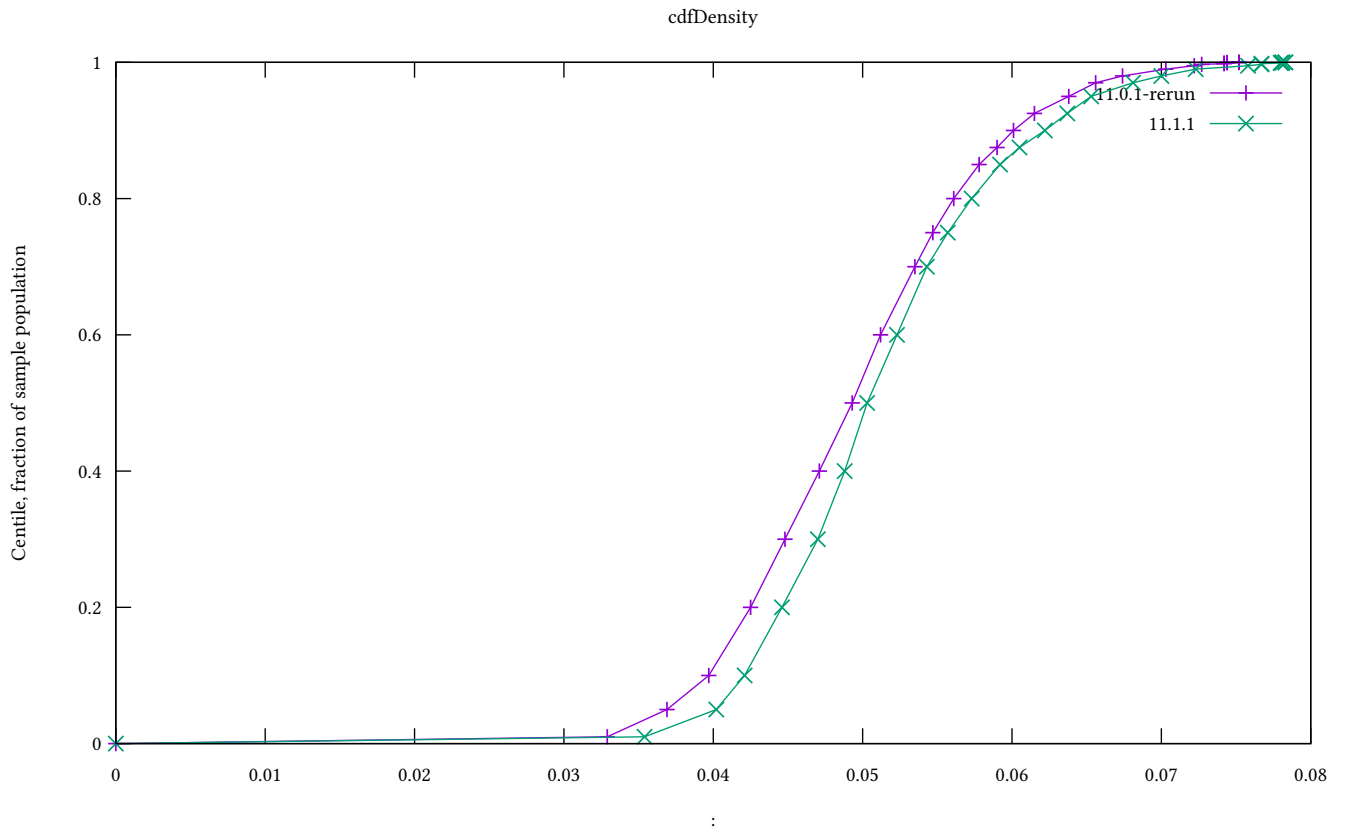


Figure 13: Chain density

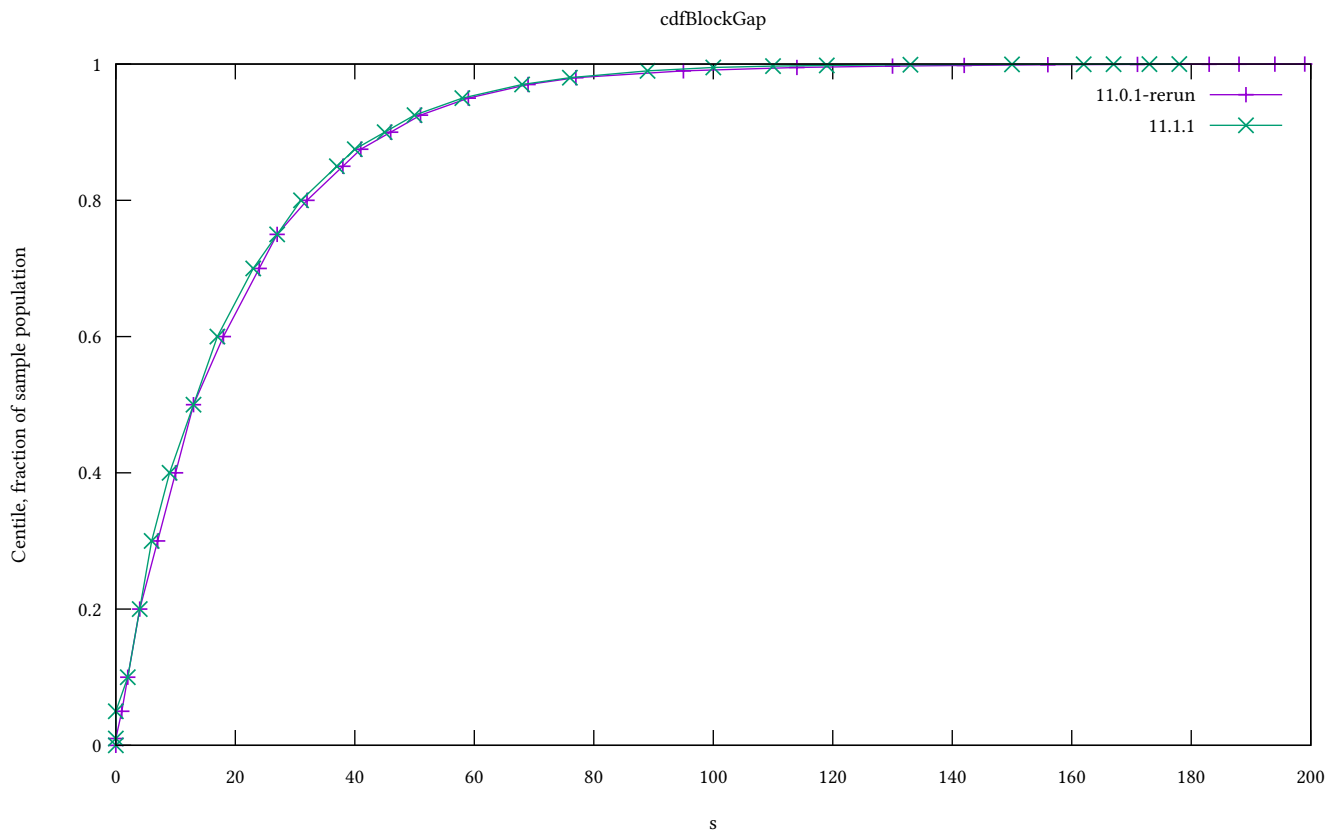


Figure 14: Interblock gap

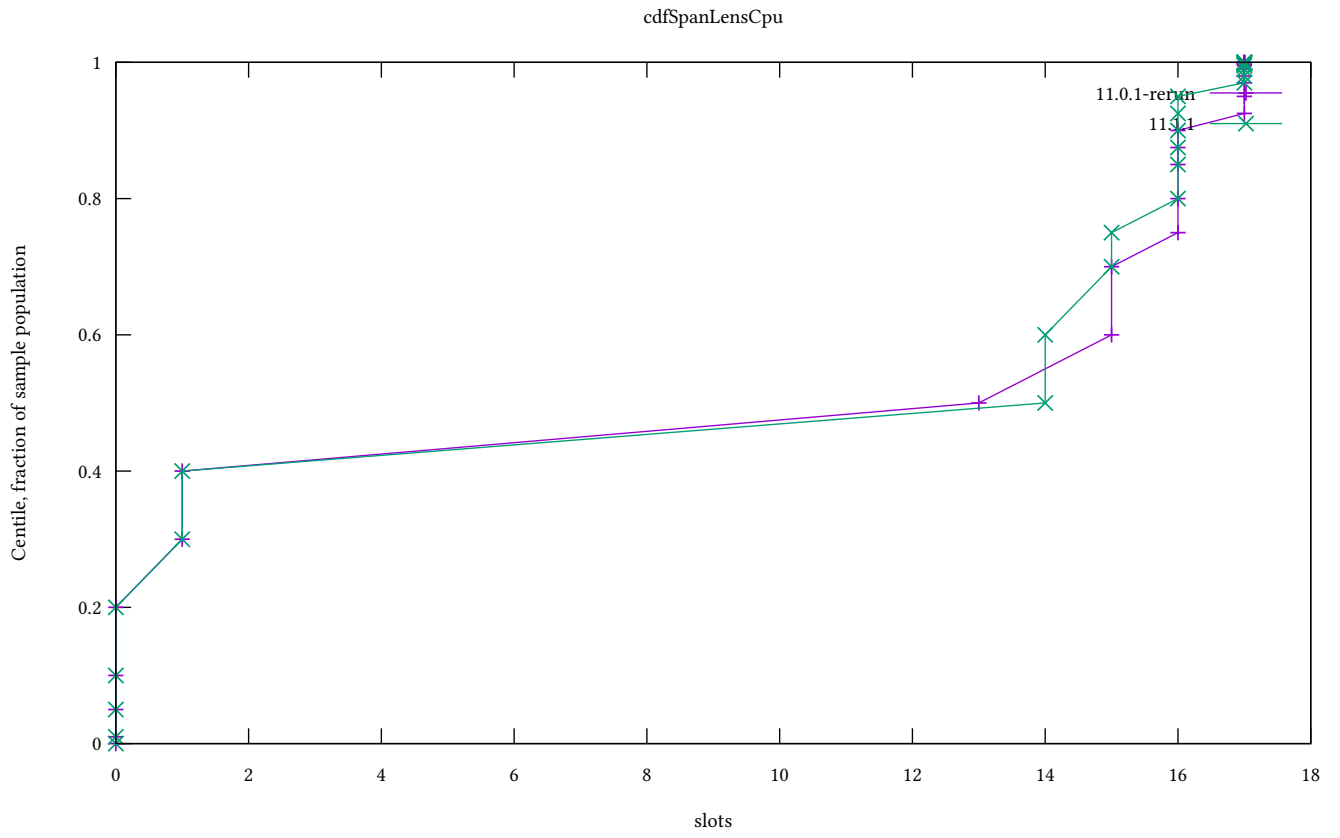


Figure 15: CPU 85% spans

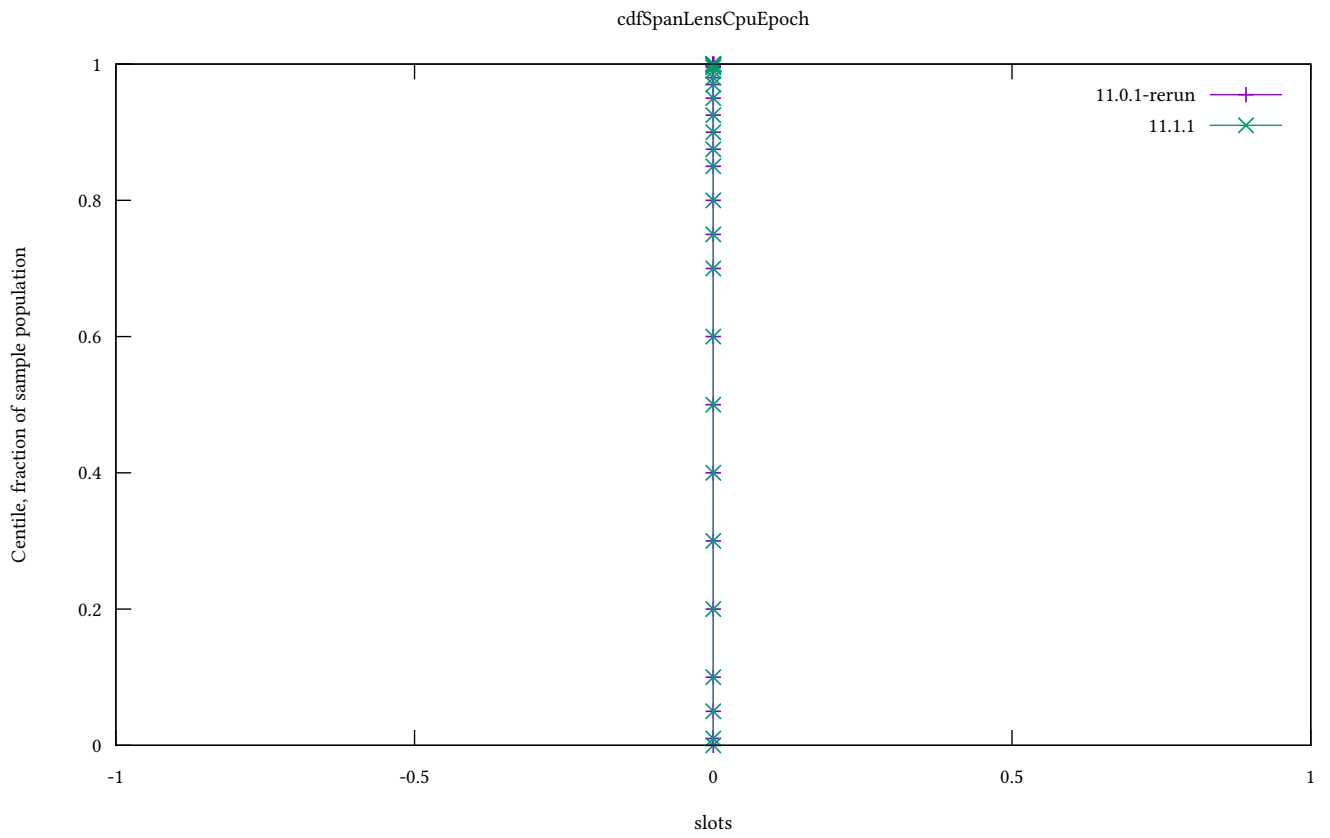


Figure 16: CPU spans at Ep boundary

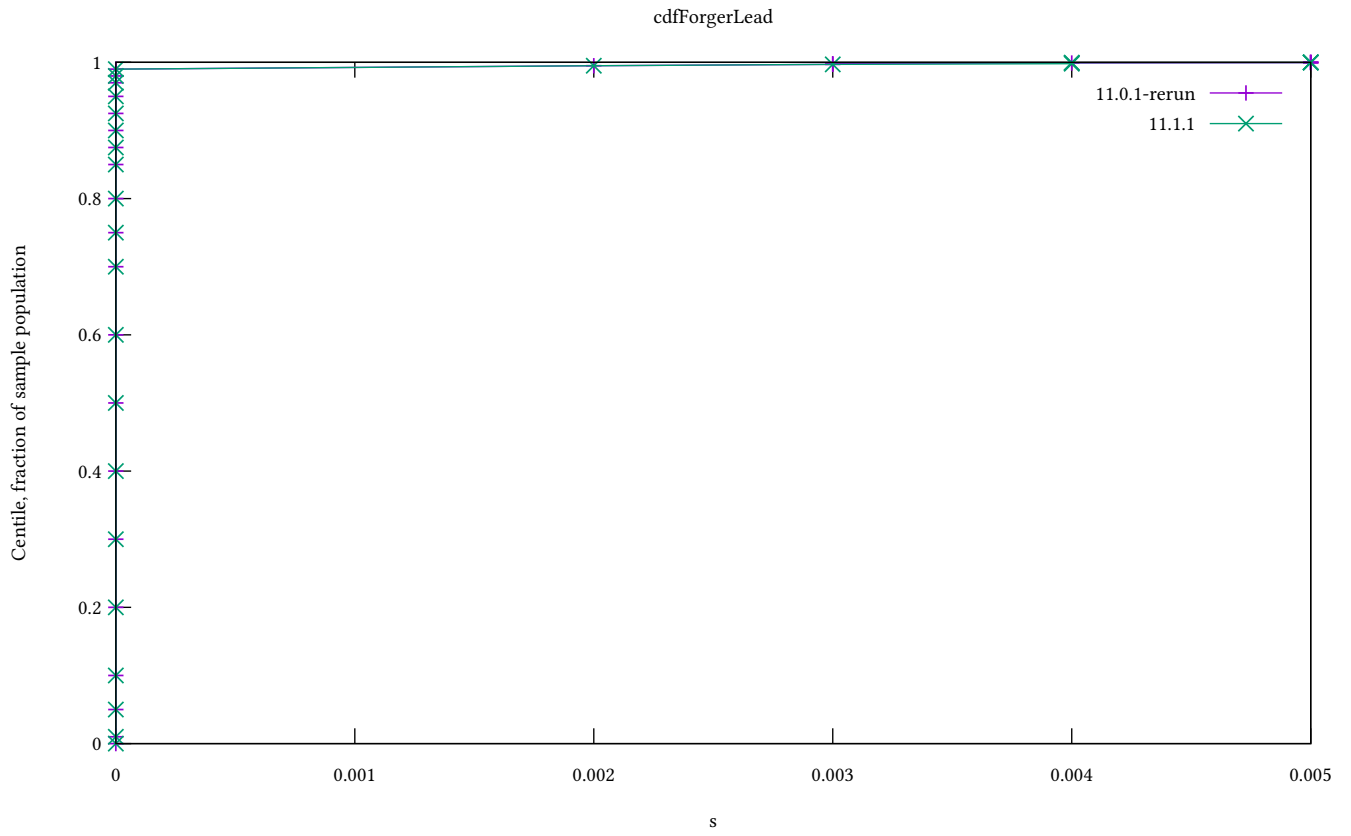


Figure 17: Leadership check duration

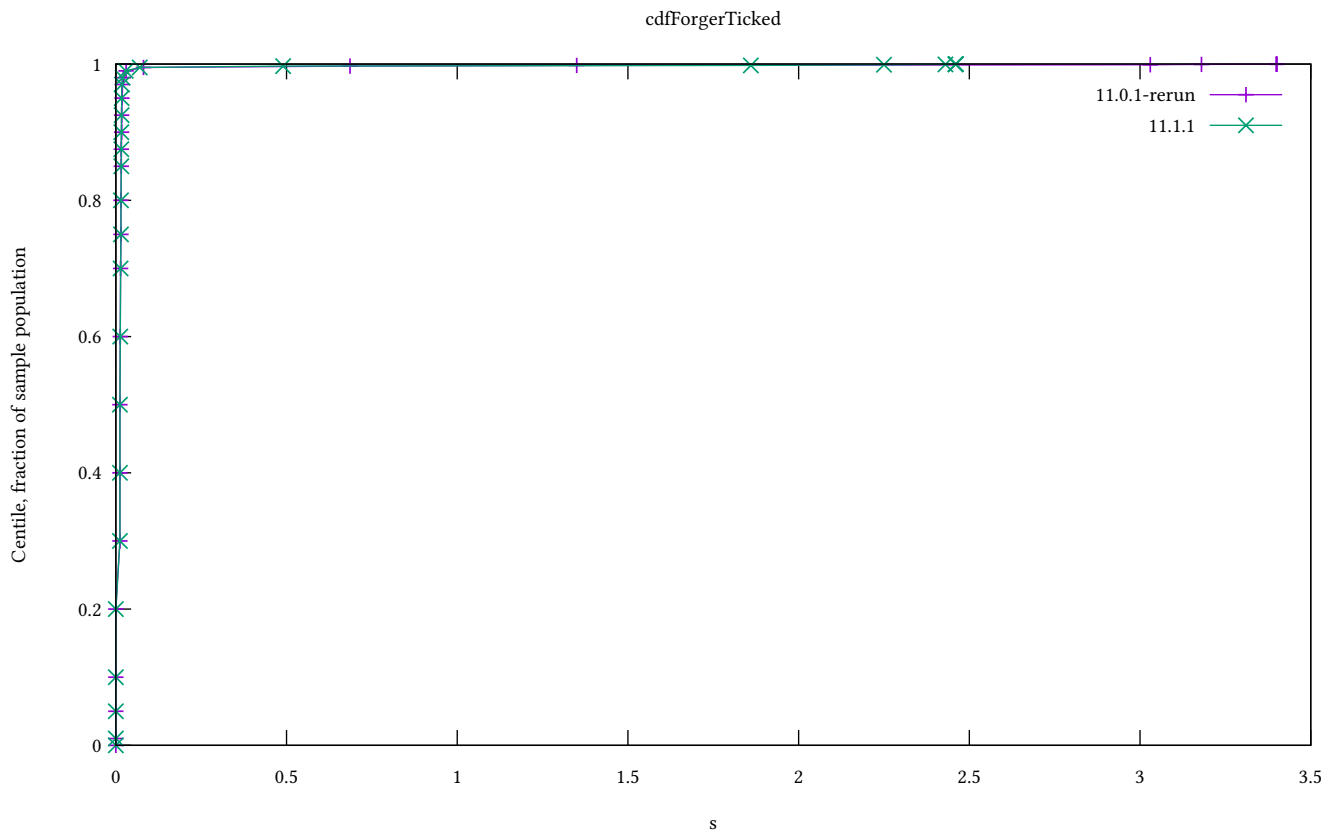


Figure 18: Ledger ticking

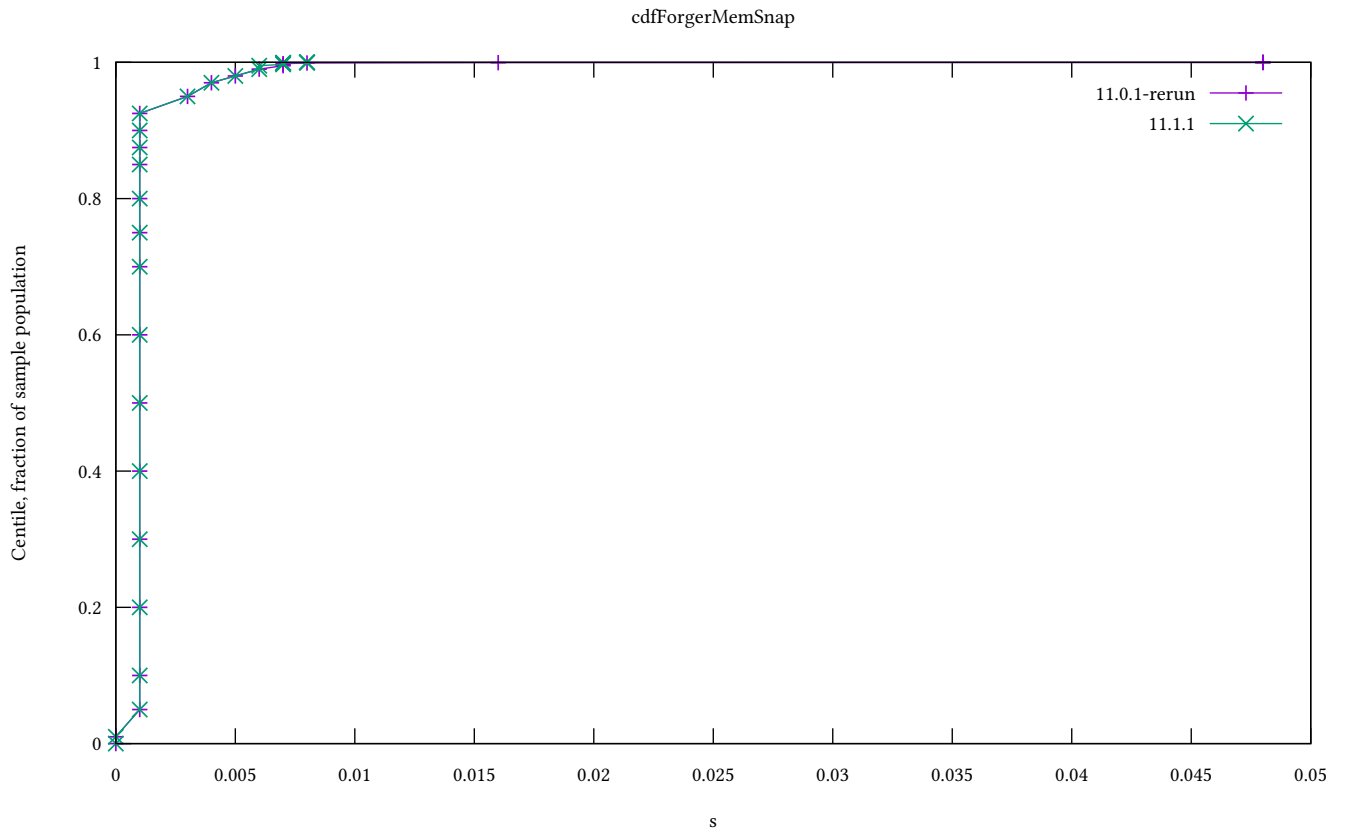


Figure 19: Mempool snapshotting

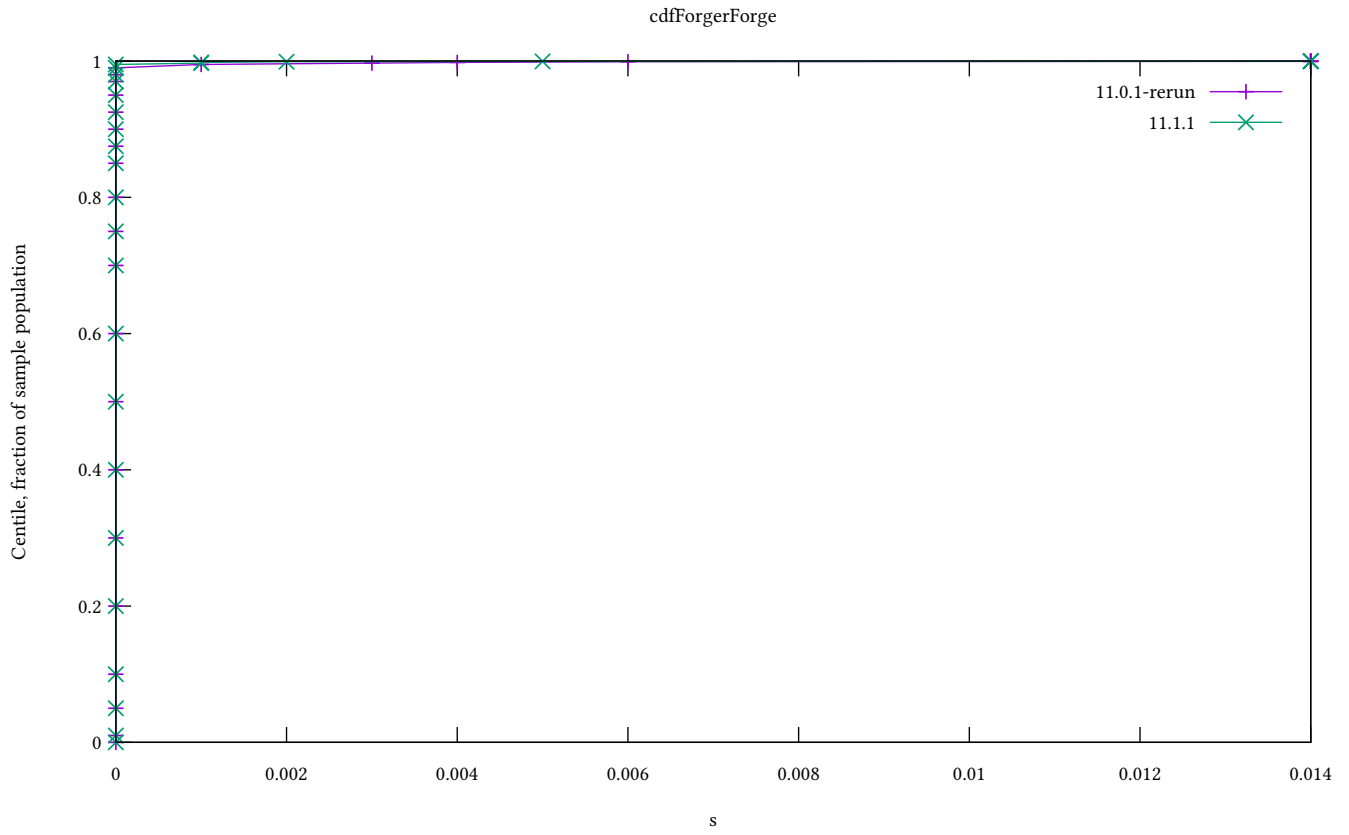


Figure 20: Leadership to forged

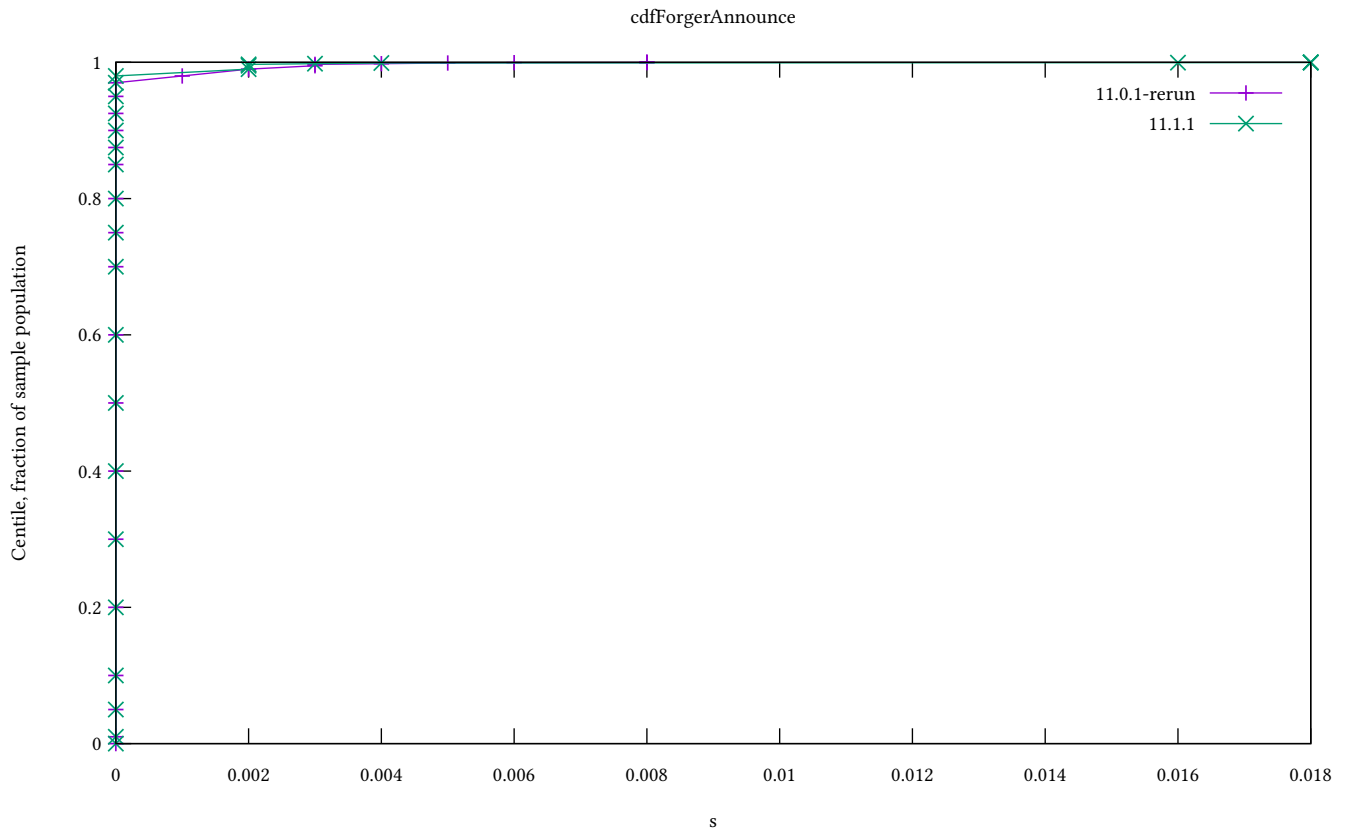


Figure 21: Forged to announced

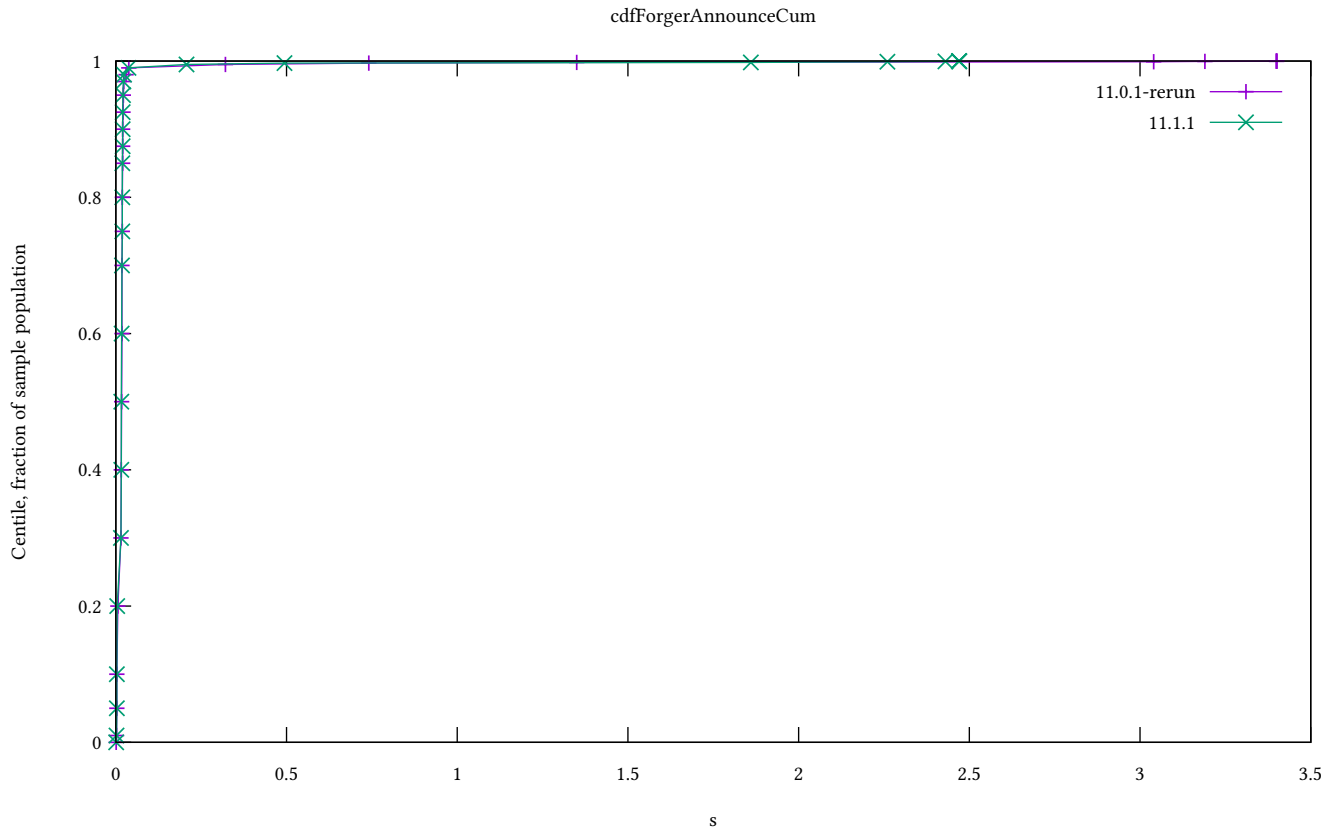


Figure 22: Slot start to announced

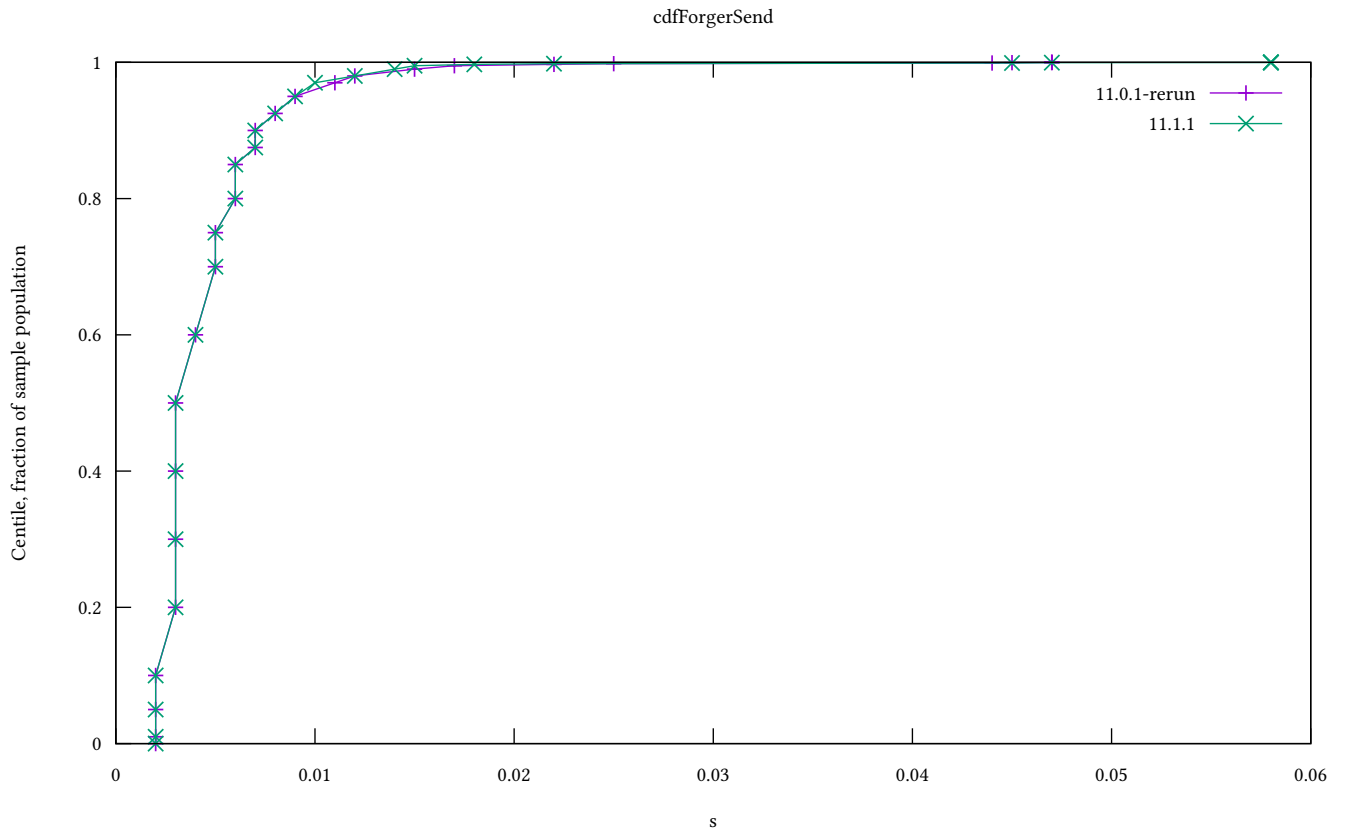


Figure 23: Forged to sending

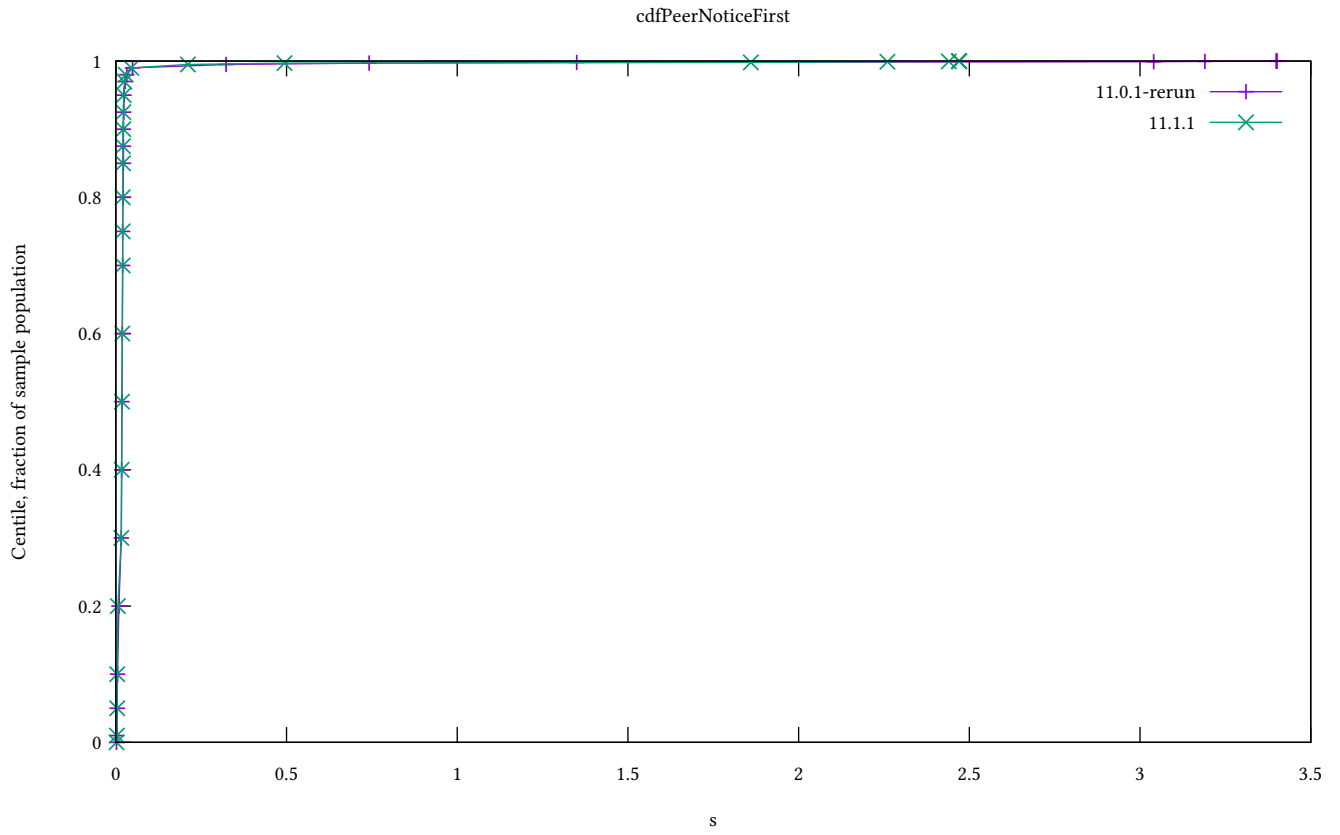


Figure 24: First peer notice

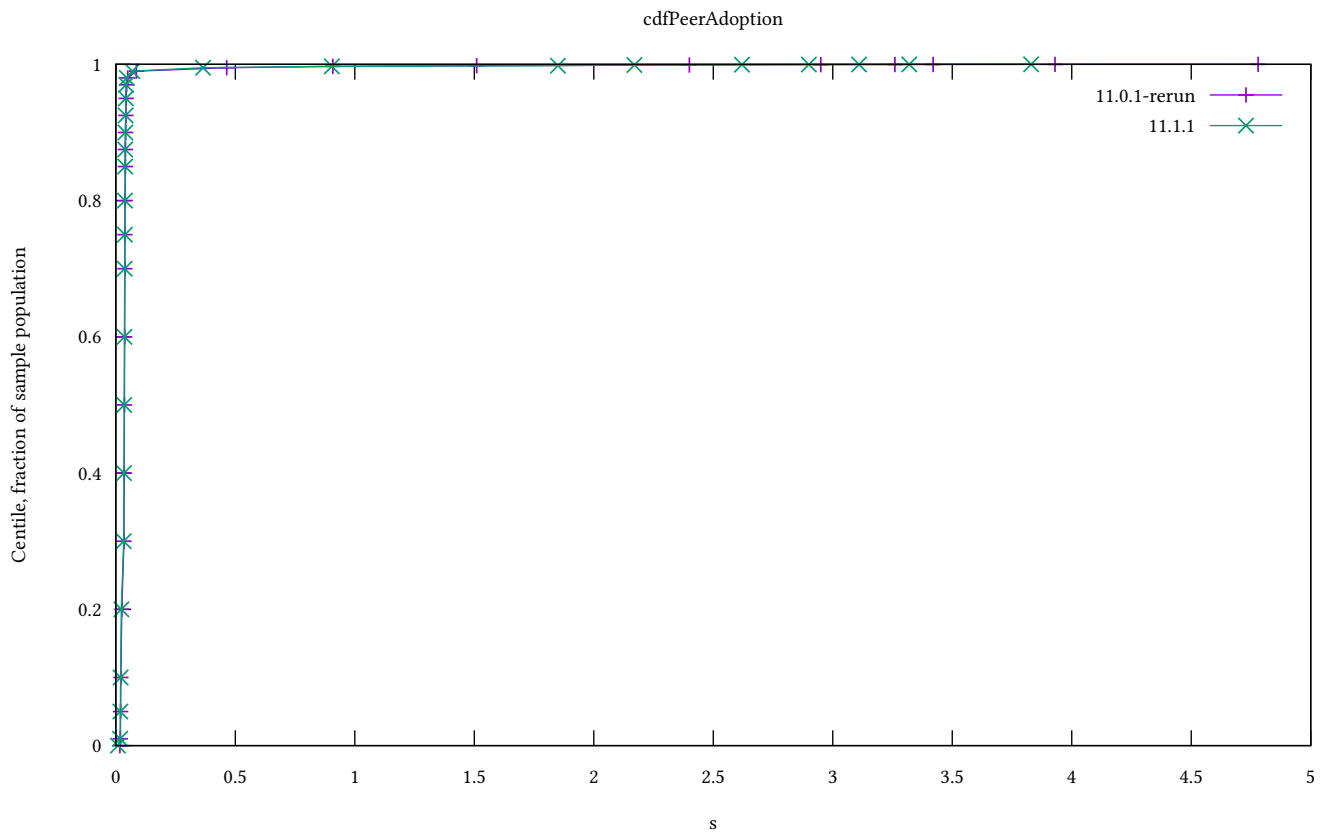


Figure 25: Fetched to adopted

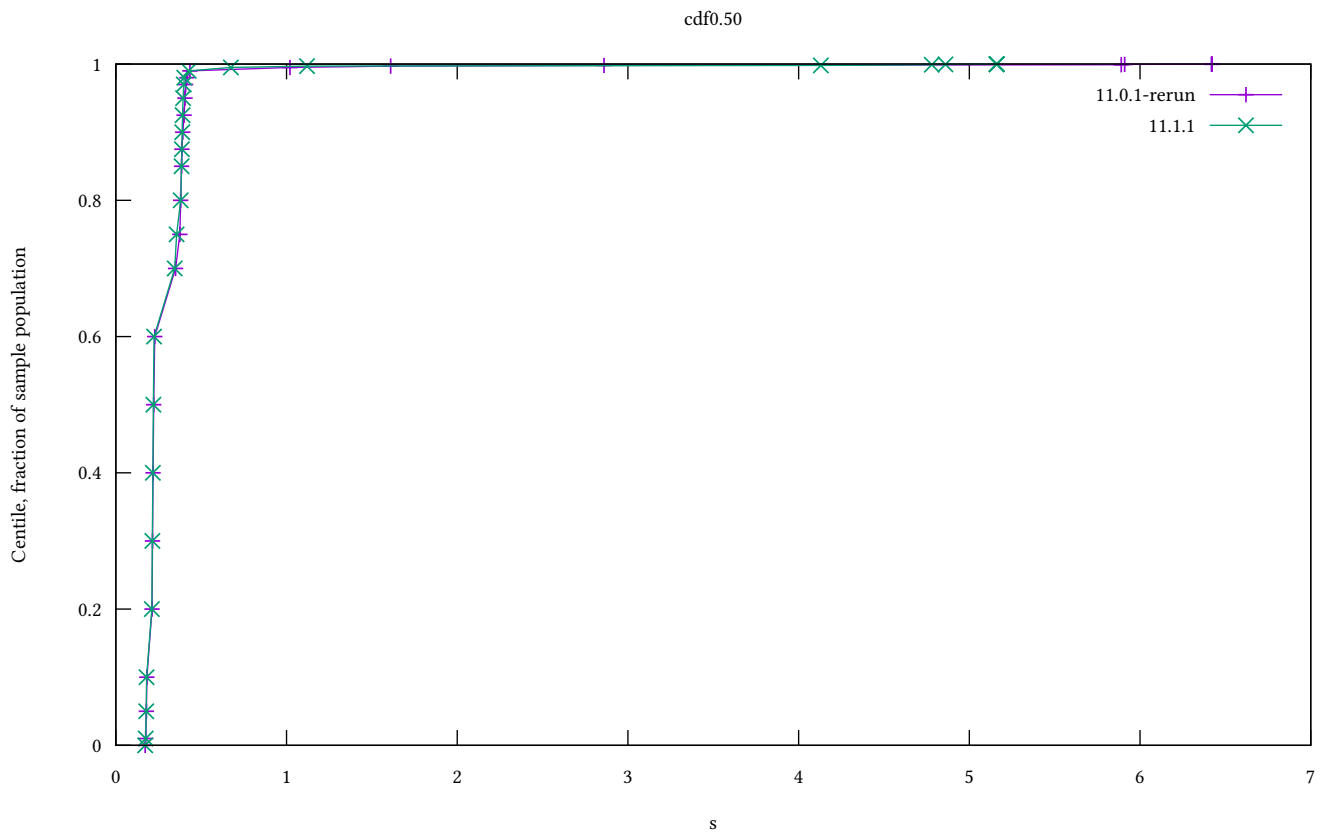


Figure 26: 0.50 adoption

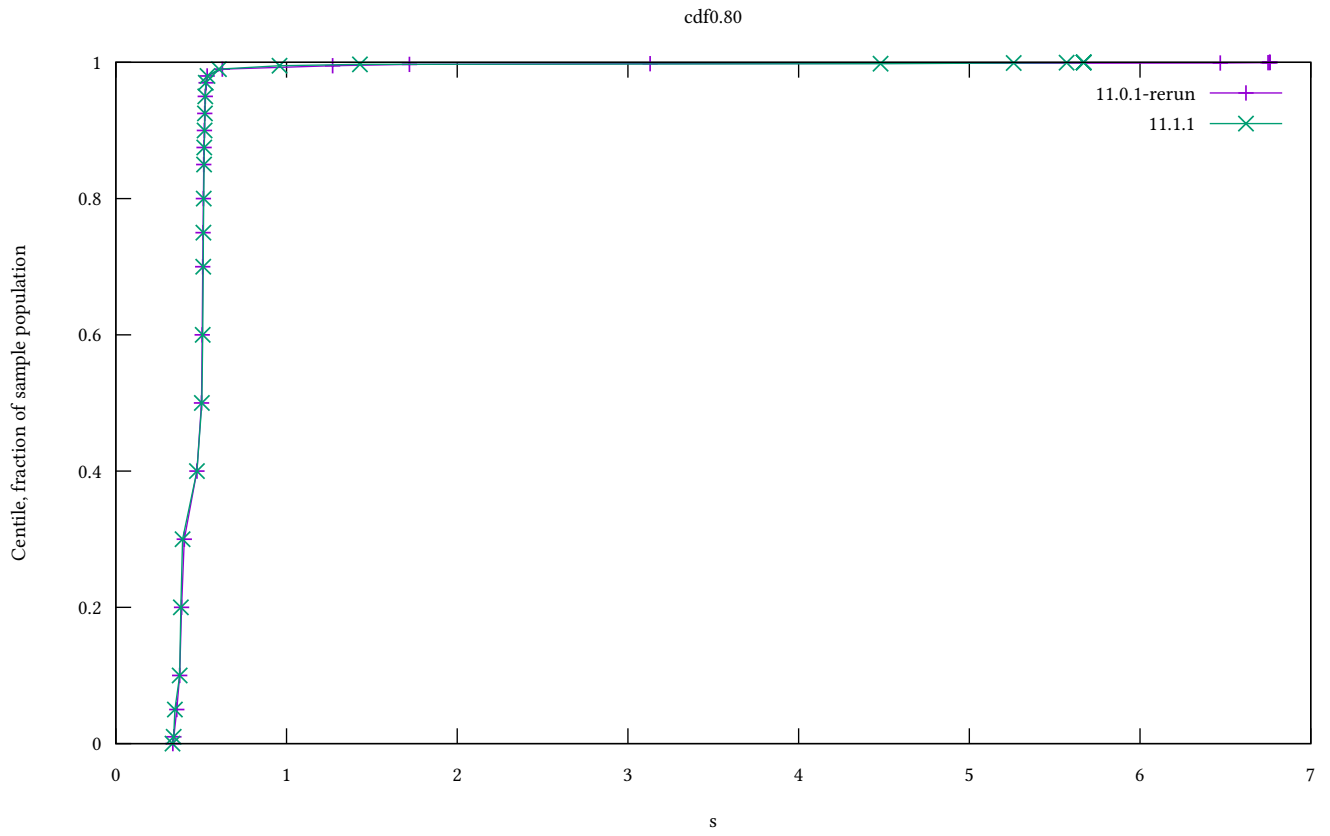


Figure 27: 0.80 adoption

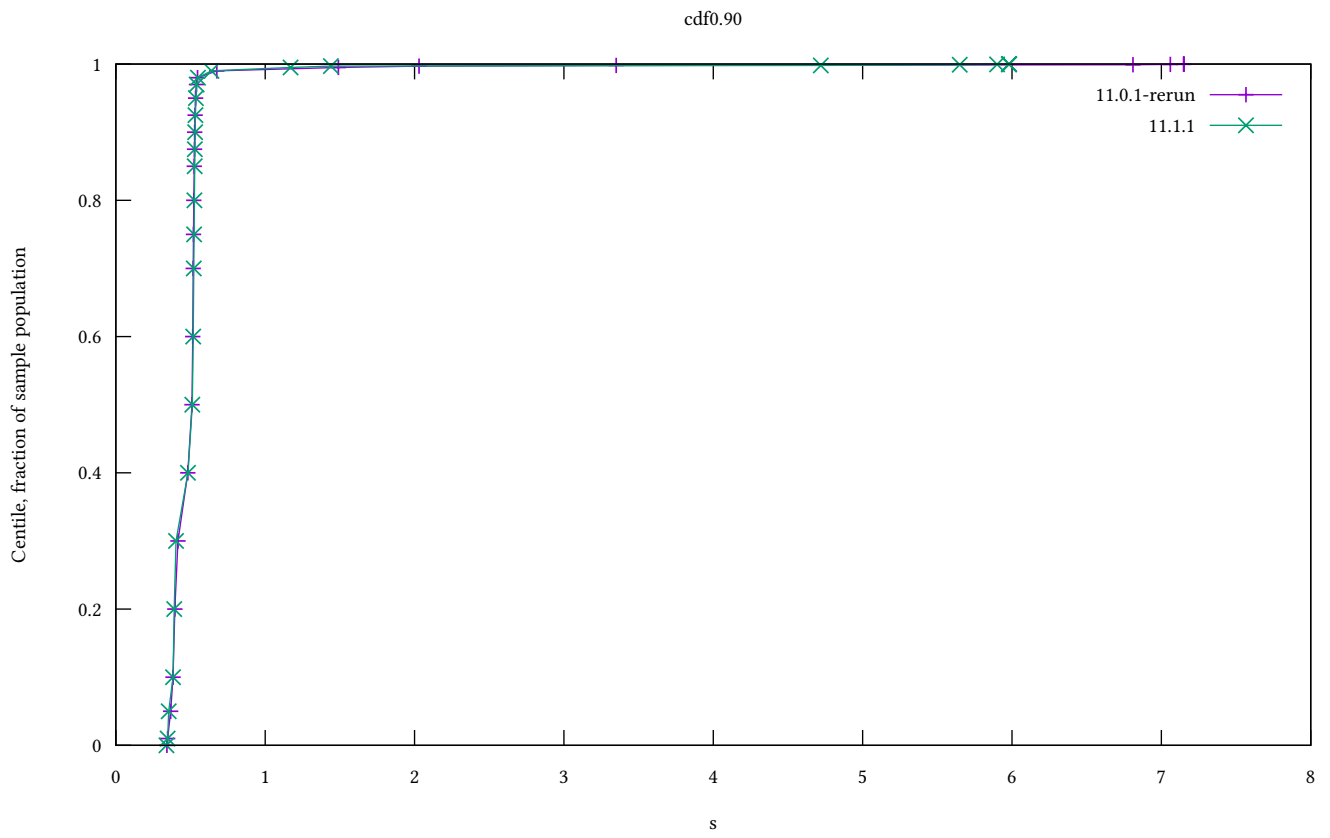


Figure 28: 0.90 adoption

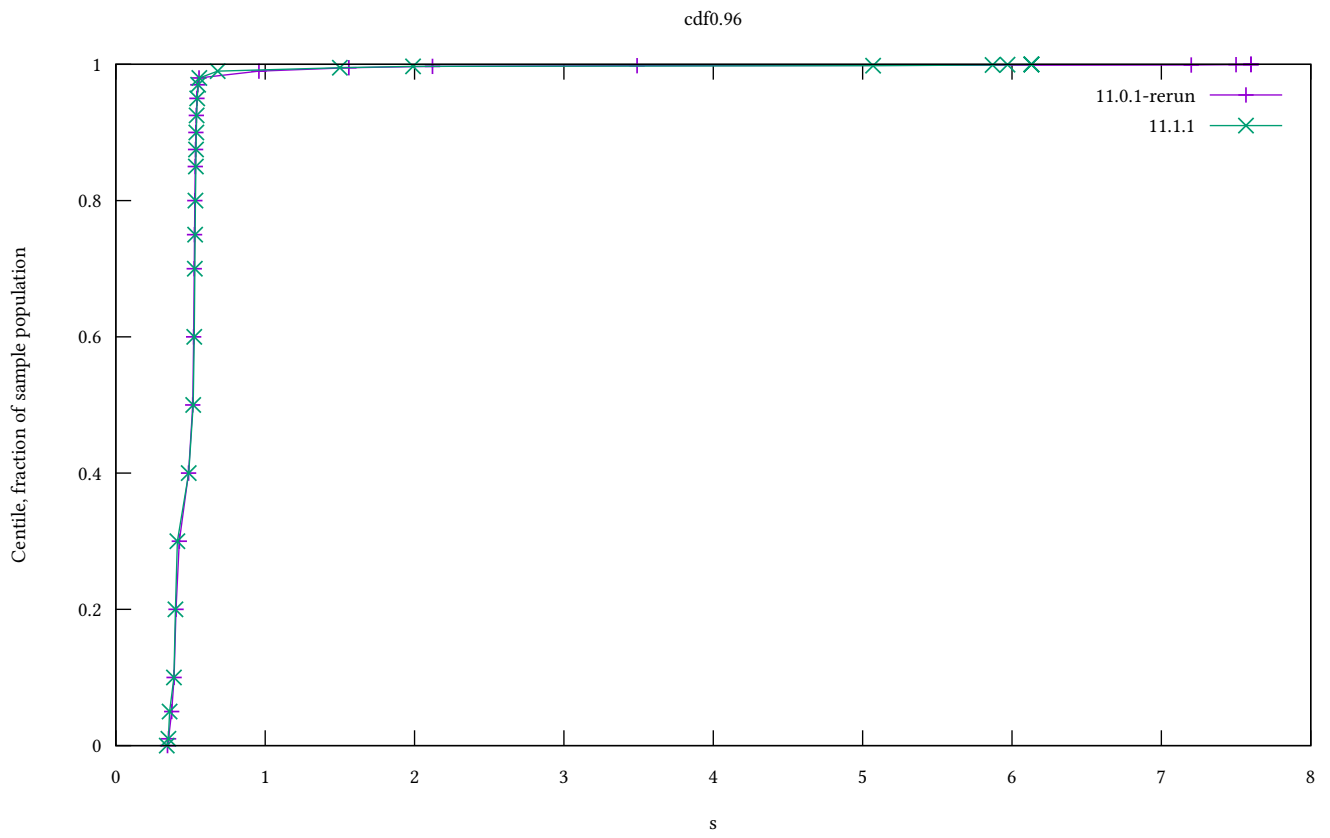


Figure 29: 0.96 adoption

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

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

Fetch 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

Fetches 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 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

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