

11.1.1 against 11.0.1-rerun

value-only 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 value-only workload.

	11.0.1-rerun	11.1.1
Analysis date	2026-09-10	2026-09-06
Cluster system start date	2026-09-09	2026-09-05
Cluster system start time	20:38:17	21:53:06
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	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	4356352.9423	4111880.8846
Log objects analysed per host	1904855.5	1828821.8846
Host run time, s	63860.7	63943.7
Host log line rate, Hz	68.216	64.305
Total log objects analysed	99052486	95098738
Run time, s	63865	63958
Analysed run duration, s	48010	48033
Run time efficiency	0.75	0.75
Node start spread, s	5.5639538	21.882692
Node stop spread, s	2.0936877	4.2428693
Slots analysed	48009	48030
Blocks analysed	2296	2206
Blocks rejected	887	898

Analysis

Resource Usage

	11.0.1-rerun	11.1.1	Δ	$\Delta\%$
Forge loop starts, units	0.9999	0.9999	0.000	0
Process CPU usage, %	2.5706	2.4216	-0.149	-6
RTS GC CPU usage, %	0.3136	0.2904	-0.023	-7
RTS Mutator CPU usage, %	2.2562	2.1309	-0.125	-6
Major GCs, events	0.0009	0.0008	-0.000	-13
Minor GCs, events	0.707	0.636	-0.071	-10
Kernel RSS, MiB	6903.7	8182.2	1278.537	19
RTS heap size, MiB	6834.7	8112.1	1277.416	19
RTS live GC dataset, MiB	3091.8	4128.8	1036.977	34
RTS alloc rate, MiB/s	21.695	19.53	-2.165	-10
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	237.74	238.86	1.125	0.5
CPU 85% spans, slots	7.357	8.2369	0.880	12

Anomaly control

	11.0.1-rerun	11.1.1	Δ	$\Delta\%$
Blocks per host, blocks	63.288	61.923	-1.365	-2
Filtered to chained blocks, :	0.7218	0.7096	-0.012	-2
Chained to forged blocks, :	0.967	0.9636	-0.003	-0.4
Height & slot battles, blocks	0.0	0.00091	0.001	NaN
Block size, Bytes	88963	88956	-6.092	-0.0

Forging

	11.0.1-rerun	11.1.1	Δ	$\Delta\%$
Started forge loop iteration, s	0.00133	0.00091	-0.000	-32
Acquired block context, s	6.6e-5	6.2e-5	-0.000	0
Acquired ledger state, s	0.0001	0.00011	0.000	0
Acquired ledger view, s	2.9e-5	3.5e-5	0.000	0
Leadership check duration, s	0.00041	0.00042	0.000	0
Ledger ticking, s	0.01864	0.01826	-0.000	-2
Mempool snapshotting, s	0.04796	0.04206	-0.006	-12
Leadership to forged, s	0.0008	0.00081	0.000	0
Forged to announced, s	0.00066	0.00066	-0.000	0
Forged to sending, s	0.00552	0.00571	0.000	3
Forged to self-adopted, s	0.0658	0.06647	0.001	1
Slot start to announced, s	0.07001	0.06333	-0.007	-10

Individual peer propagation

	11.0.1-rerun	11.1.1	Δ	$\Delta\%$
First peer notice, s	0.07171	0.0651	-0.007	-9
First peer fetch, s	0.08251	0.07587	-0.007	-8
Notice to fetch request, s	0.00137	0.00144	0.000	0
Fetch duration, s	0.38299	0.38649	0.004	0.9
Fetches to announced, s	0.00088	0.00087	-0.000	0
Fetches to sending, s	0.04234	0.04168	-0.001	-2
Fetches to adopted, s	0.06794	0.06857	0.001	0.9

End-to-end propagation

	11.0.1-rerun	11.1.1	Δ	$\Delta\%$
0.50 adoption, s	0.60998	0.60771	-0.002	-0.4
0.80 adoption, s	1.05147	1.05466	0.003	0.3
0.90 adoption, s	1.06913	1.07285	0.004	0.3
0.92 adoption, s	1.07307	1.07718	0.004	0.4
0.94 adoption, s	1.07754	1.08171	0.004	0.4
0.96 adoption, s	1.08213	1.08706	0.005	0.5
0.98 adoption, s	1.08964	1.09376	0.004	0.4
1.00 adoption, s	1.10887	1.10904	0.000	0.0

Observations

Resources

1. Process CPU usage decreases by 6% (Mutator -6%, GC CPU -7%); trace emission drops by the same 6% (log line rate 68.2 - > 64.3 Hz).
2. Kernel RSS and RTS heap size increase by 1.25 GiB or 19%; the RTS live GC dataset increases by 1.01 GiB or 34%, accounting for most of the RSS increase.
3. Allocation rate decreases by 10%, Minor GCs by 10% and Major GCs by 13%.
4. CPU 85% spans lengthen by 0.9 slots or 12%; node start spread is 21.9 s against 5.6 s.

Forging

1. Mempool snapshotting improves by 5.9 ms or 12%; net, a new header is announced 6.7 ms or 10% earlier into the slot. Every other step of the forge loop moves by under 1 ms.

Peer propagation

1. First peer notice and first peer fetch improve by 6.6 ms, or 9% and 8%, carrying over the earlier header announcement.
2. Block Fetch duration is practically unchanged (3.5 ms or 0.9% slower), and so are the remaining peer-to-peer steps.

End-to-end propagation

1. Cluster adoption is unchanged: 3 ms to 5 ms (0.3% - 0.5%) slower from the 80th centile upward, 2 ms or 0.4% faster at the median.

Appendix A: charts

Cluster performance charts

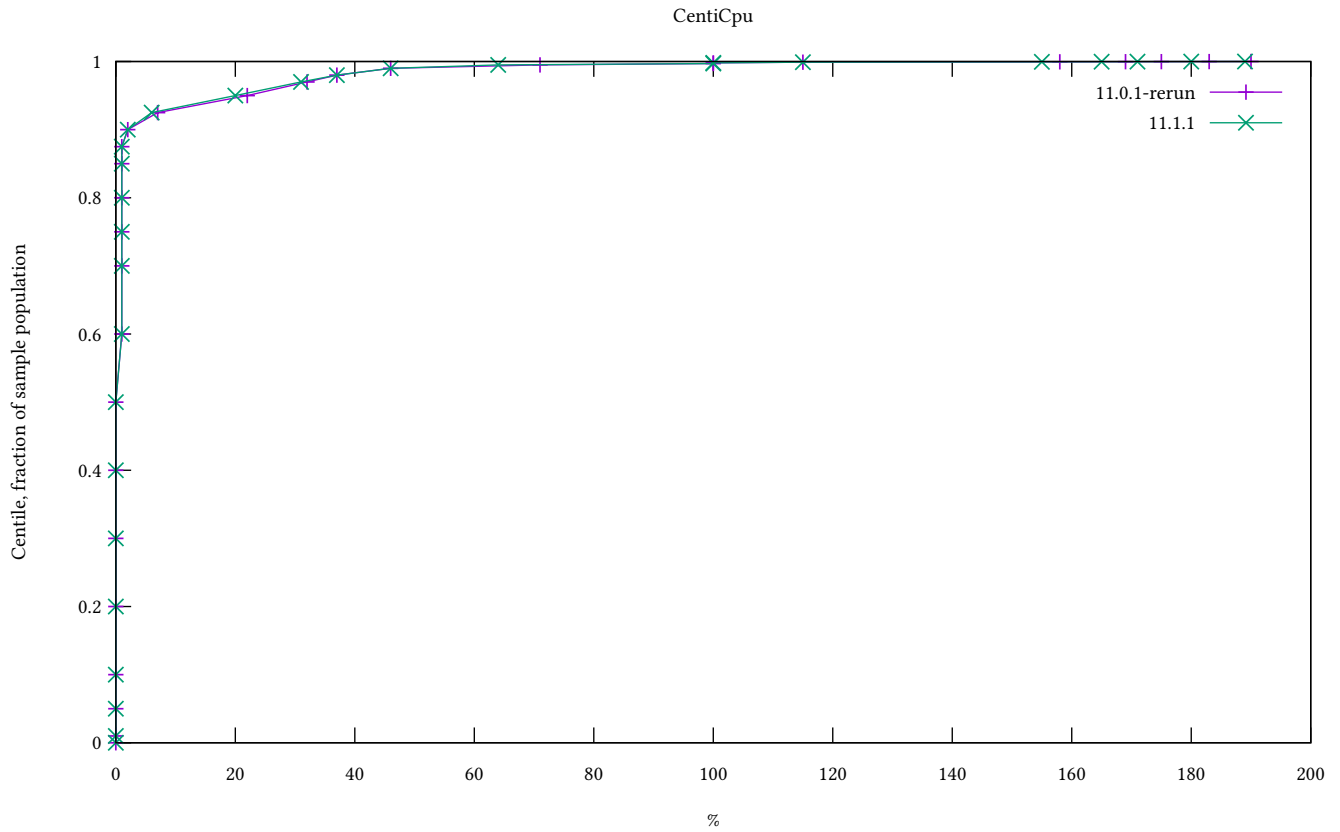


Figure 1: Process CPU usage

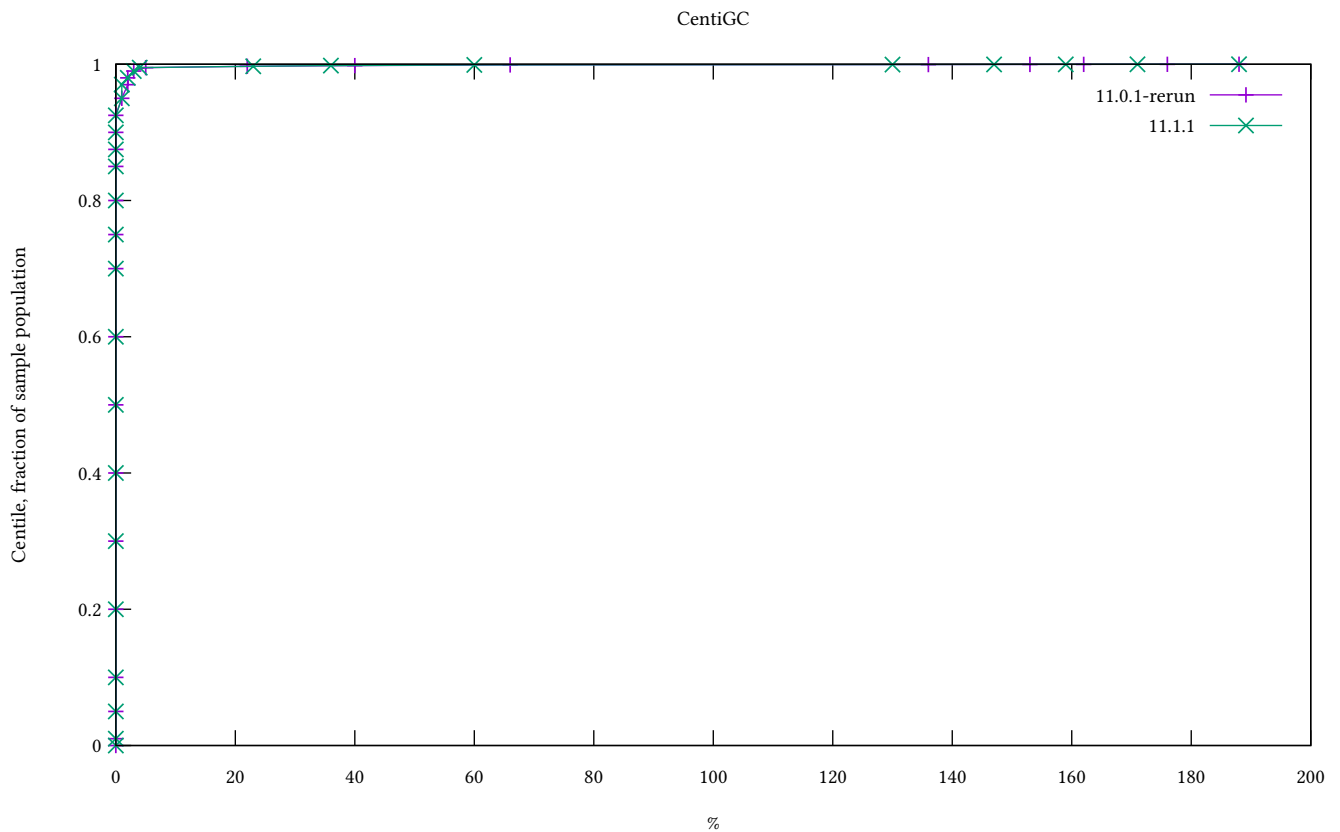


Figure 2: RTS GC CPU usage

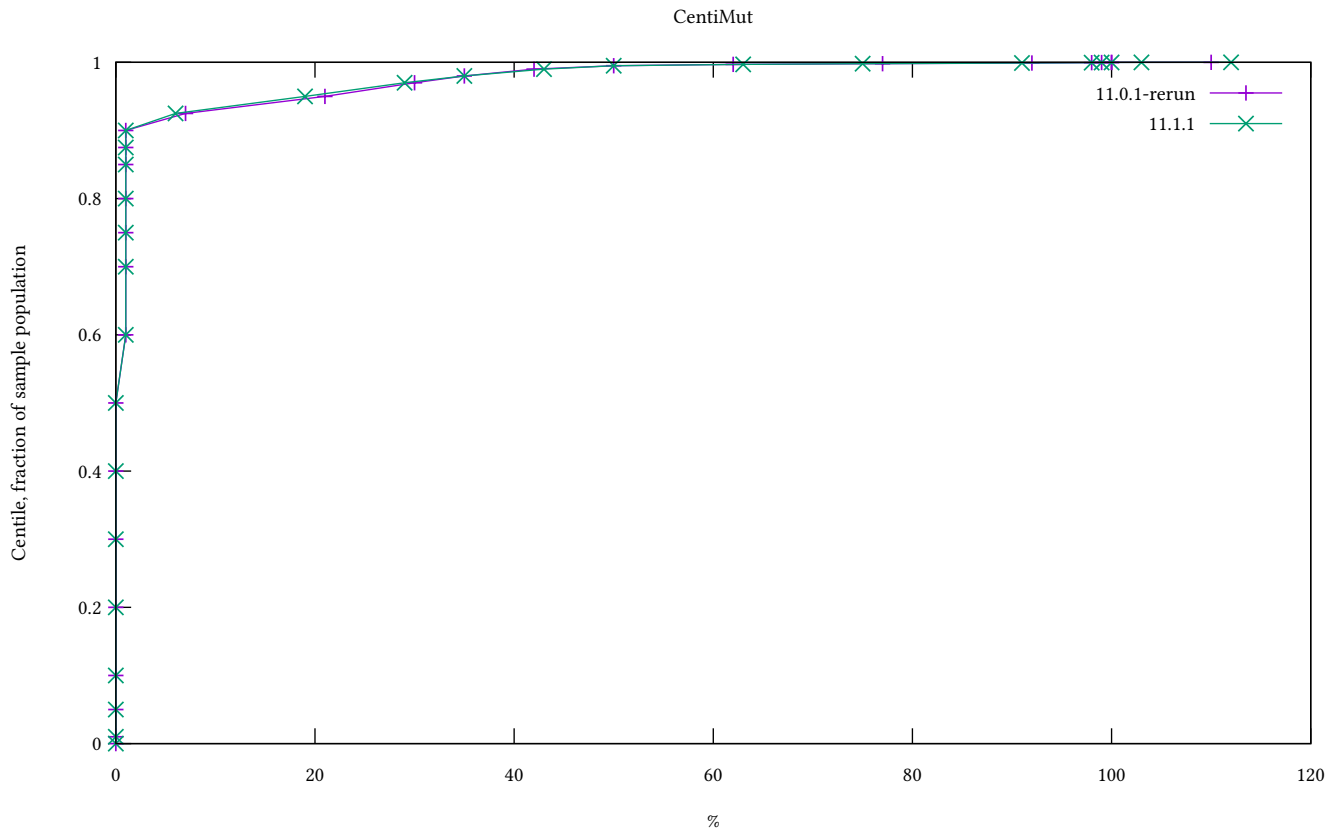


Figure 3: RTS Mutator CPU usage

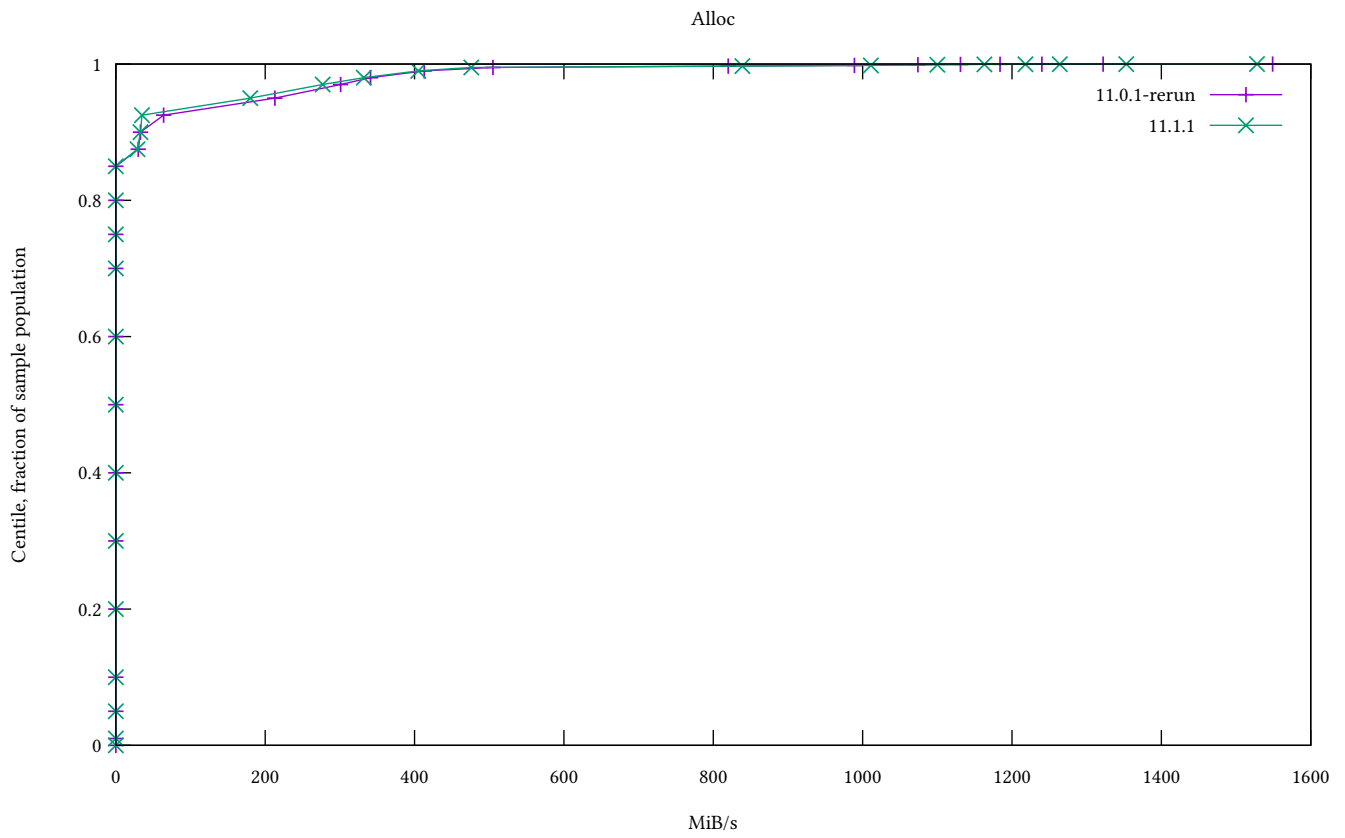


Figure 4: RTS alloc rate

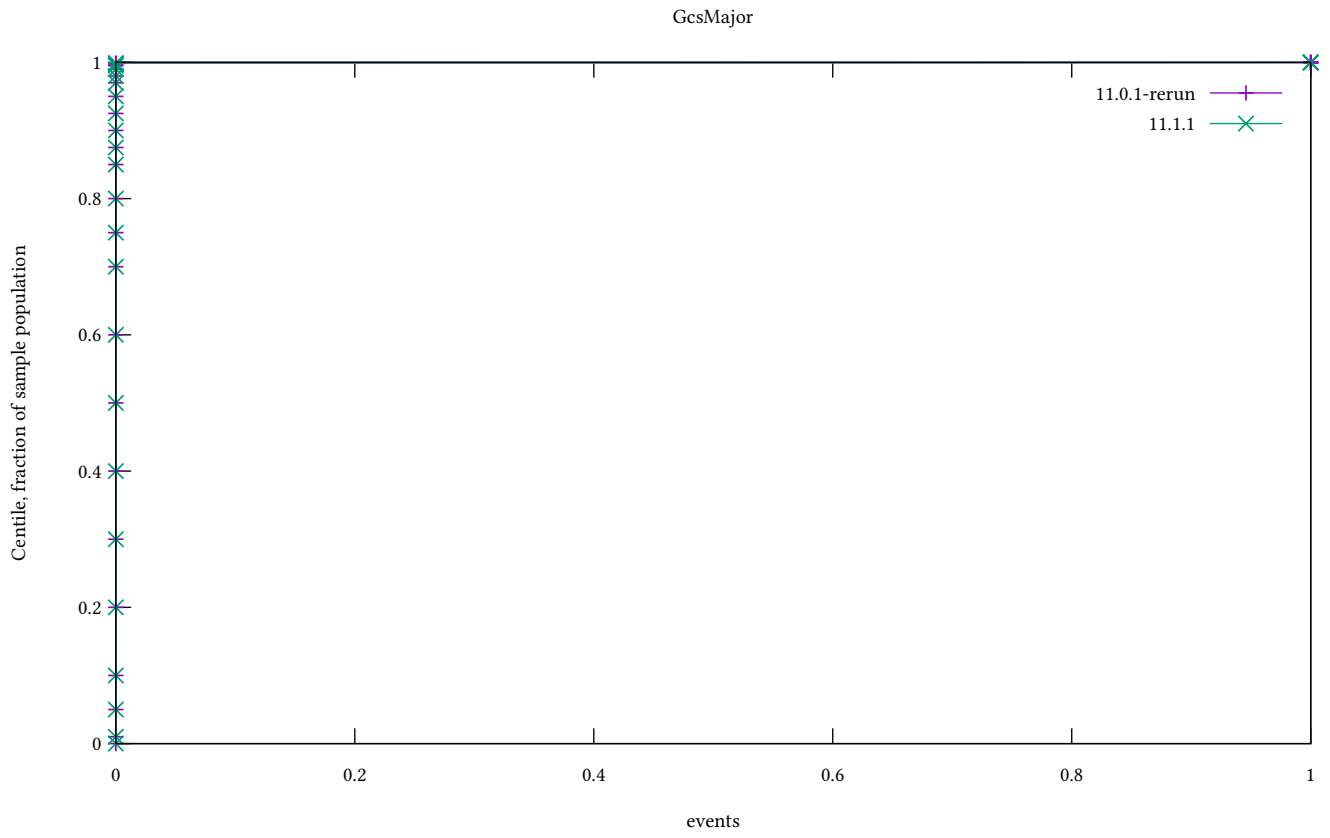


Figure 5: Major GCs

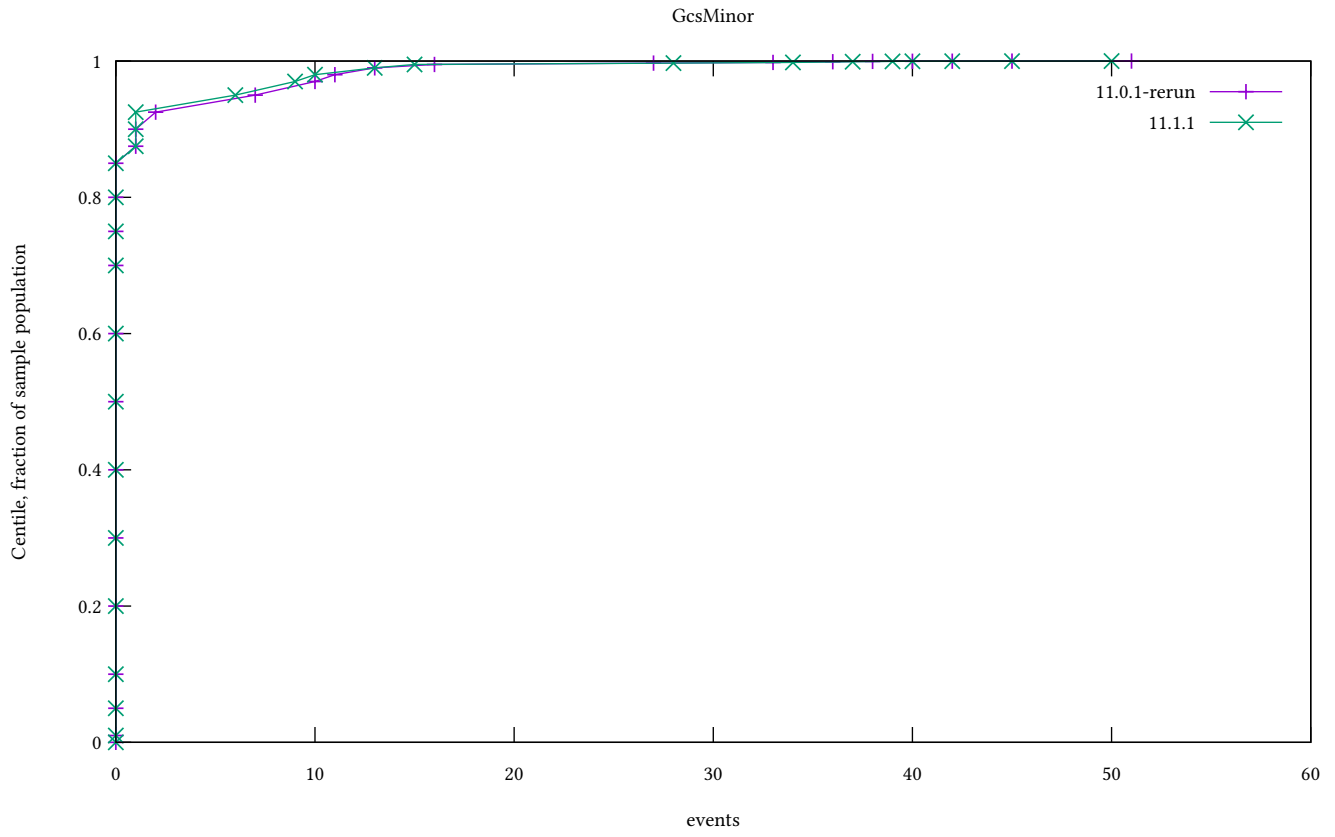


Figure 6: Minor GCs

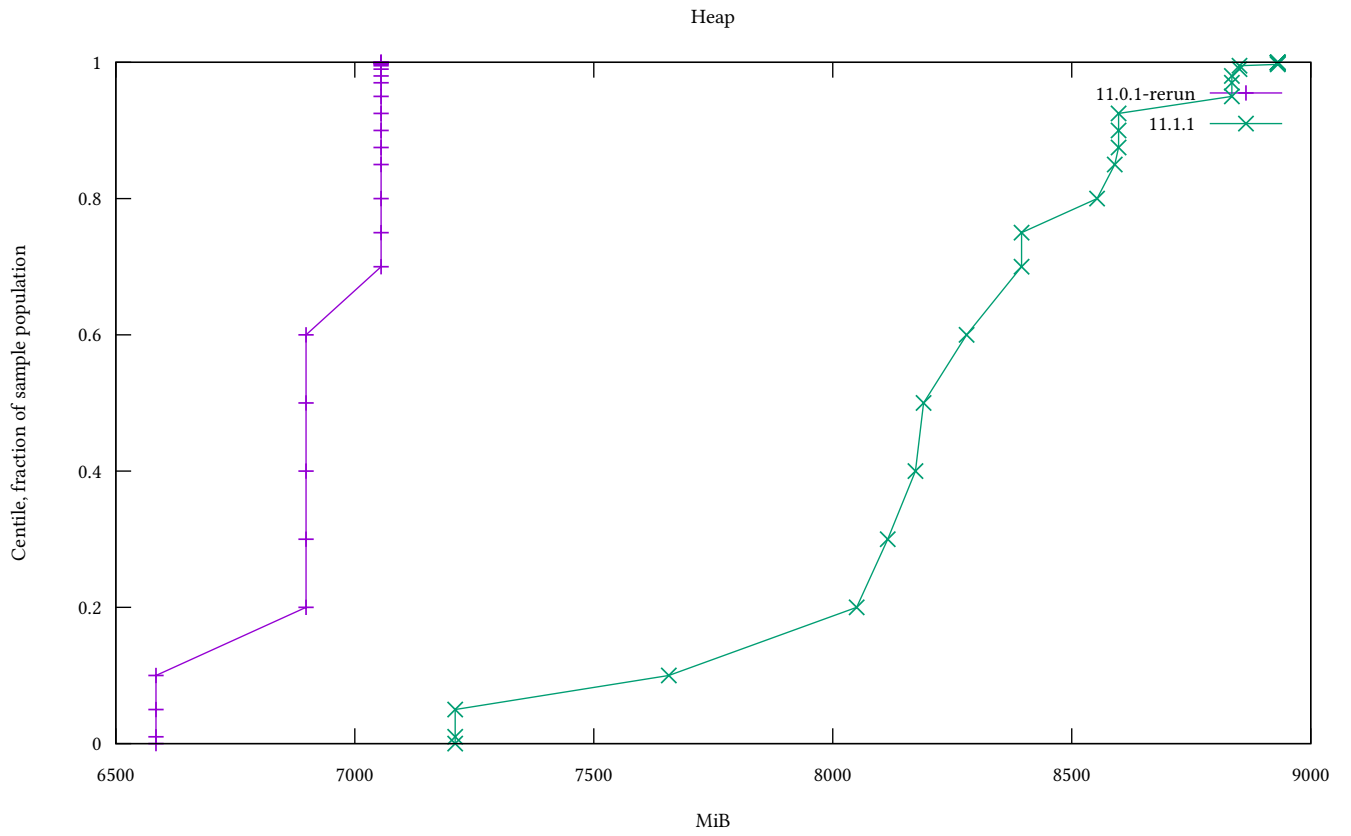


Figure 7: RTS heap size

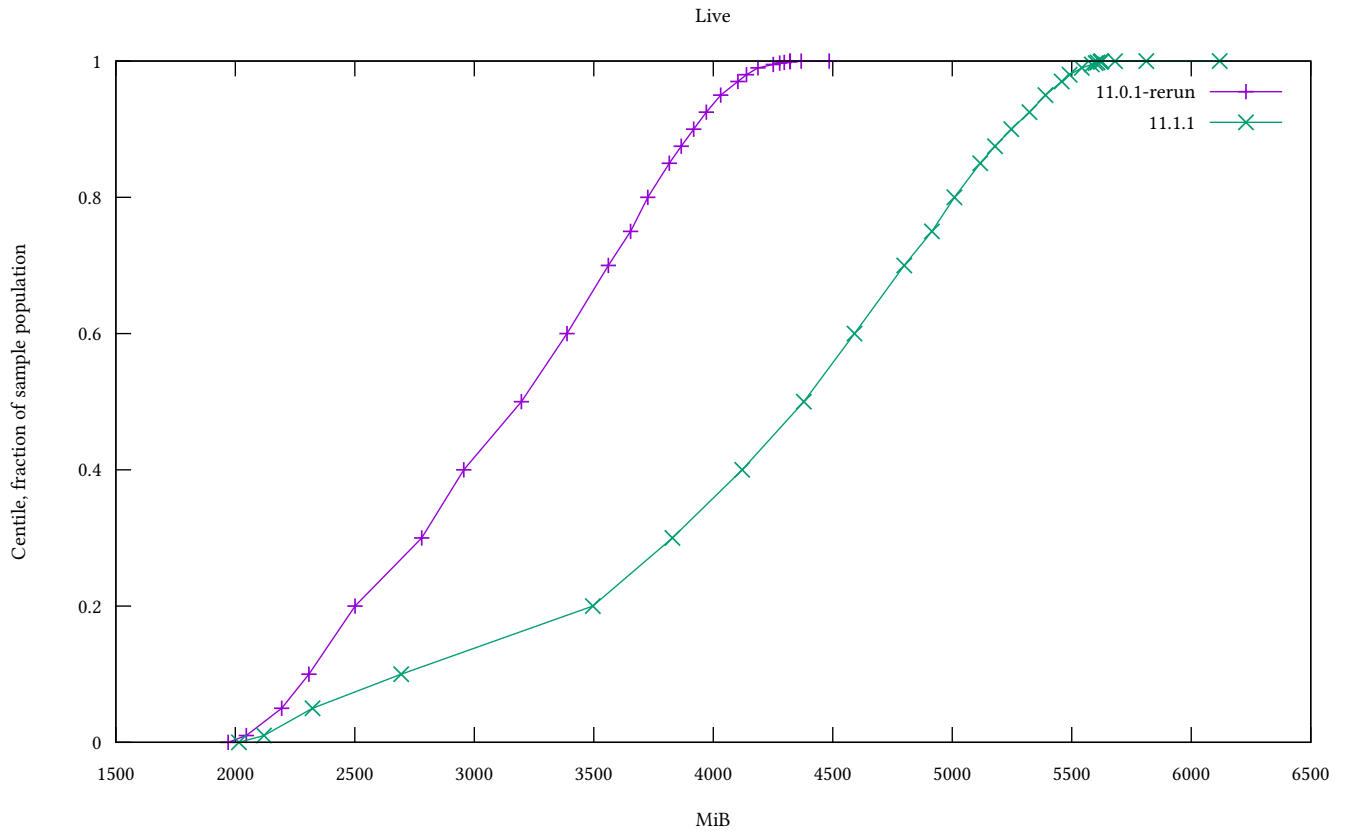


Figure 8: RTS live GC dataset

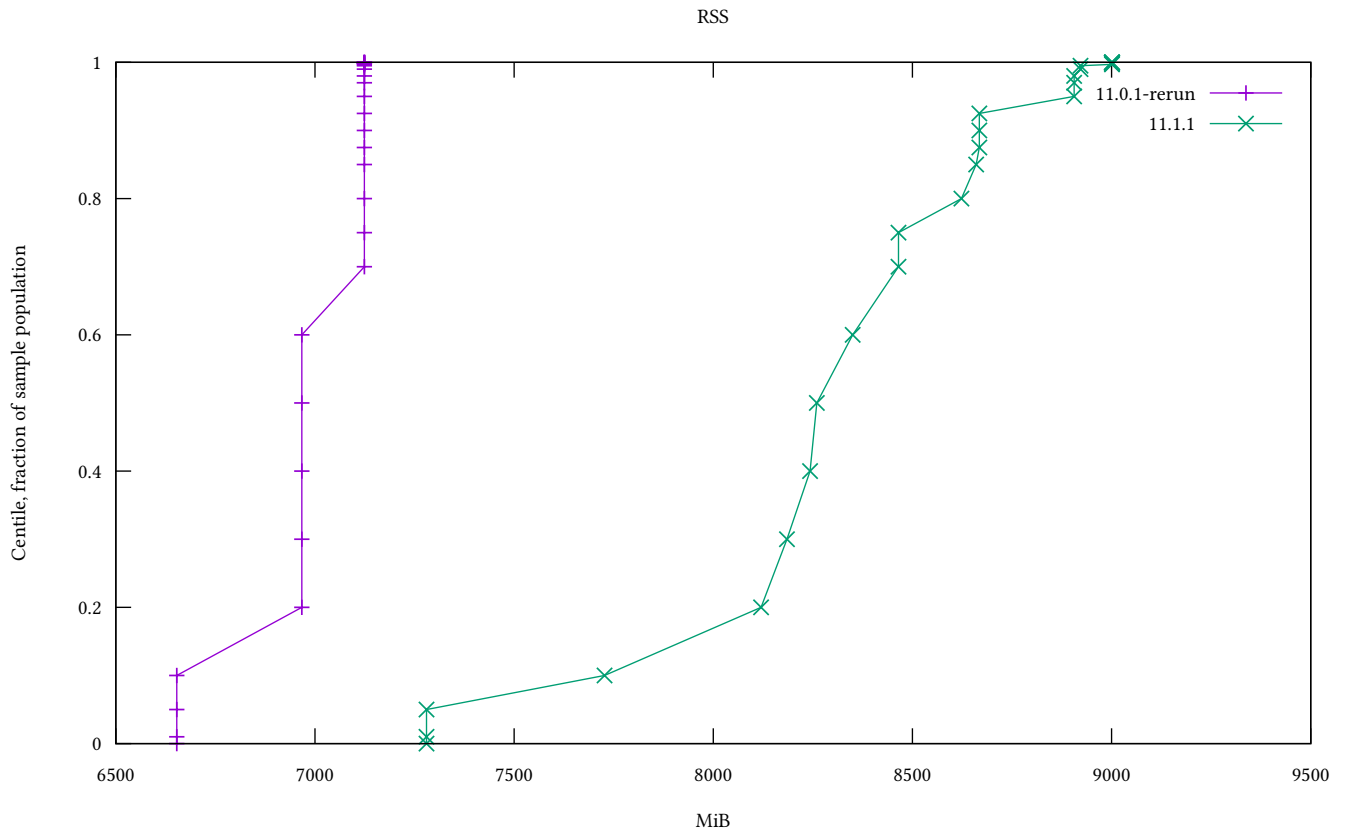


Figure 9: Kernel RSS

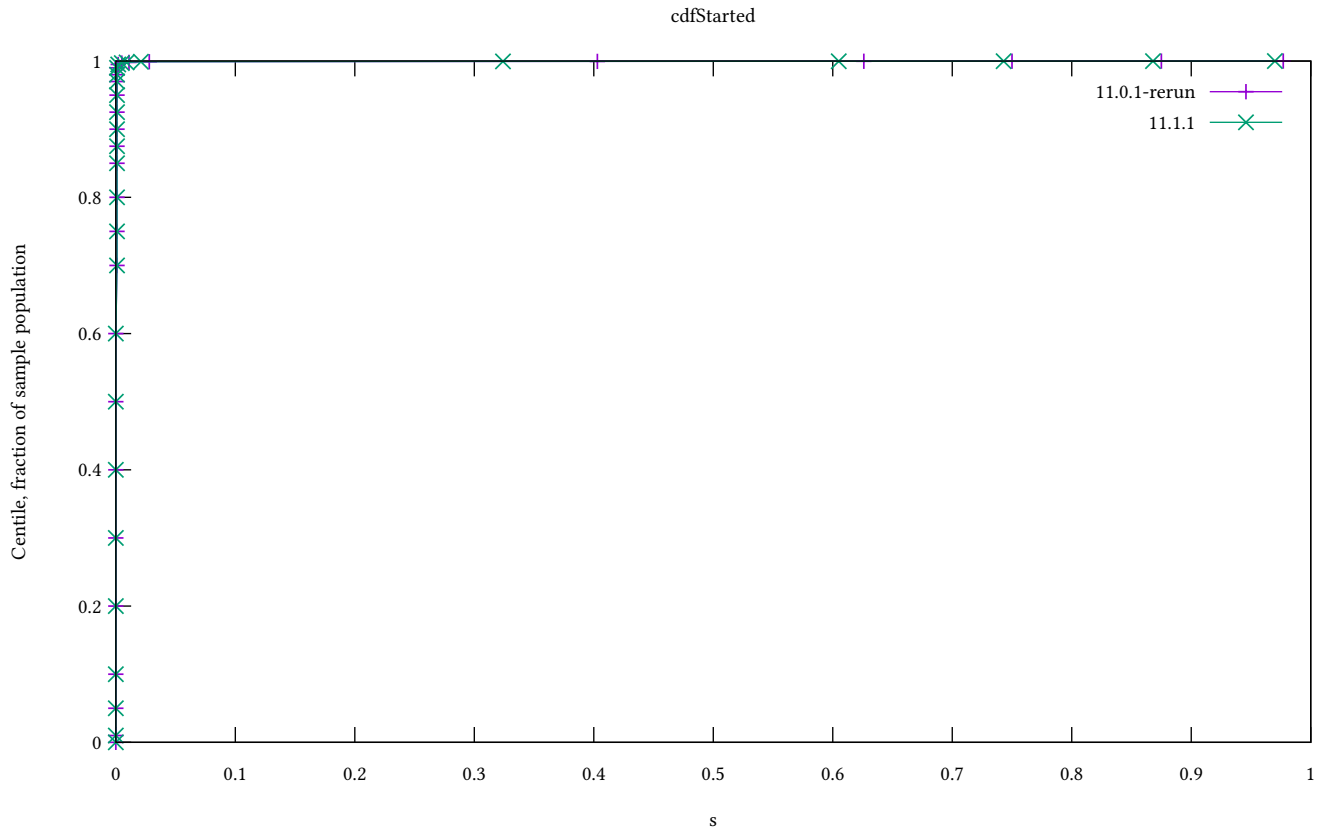


Figure 10: Forge loop tardiness

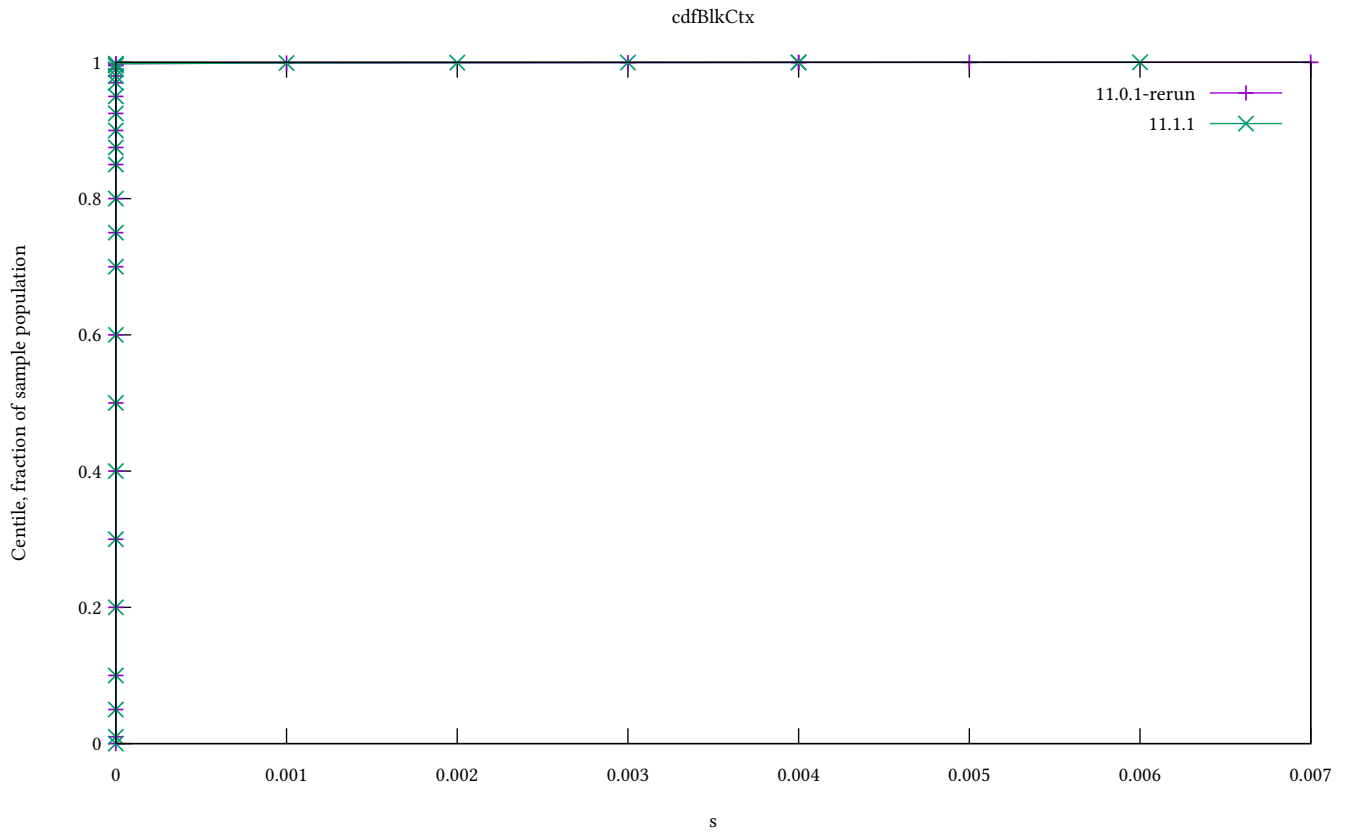


Figure 11: Block context acquisition delay

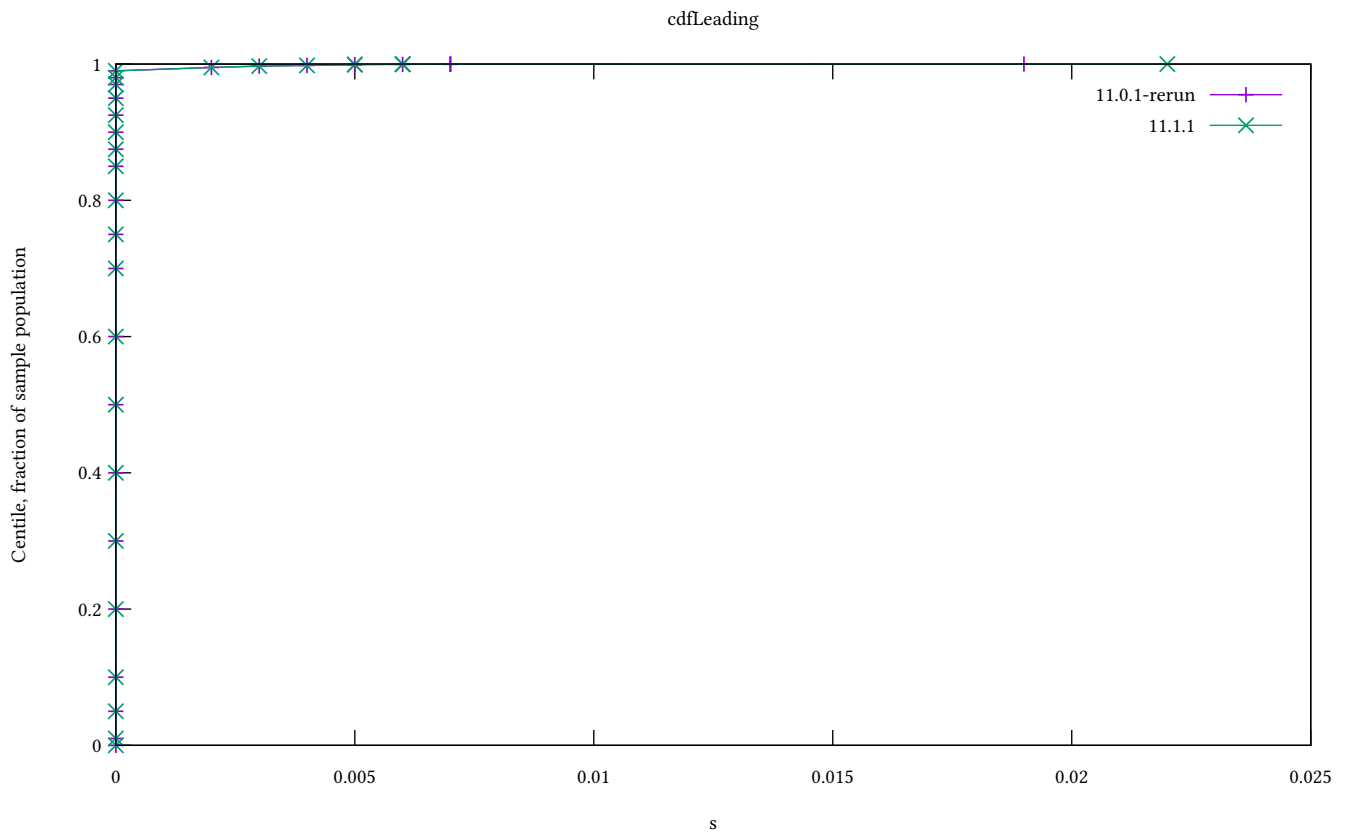


Figure 12: Leadership check duration

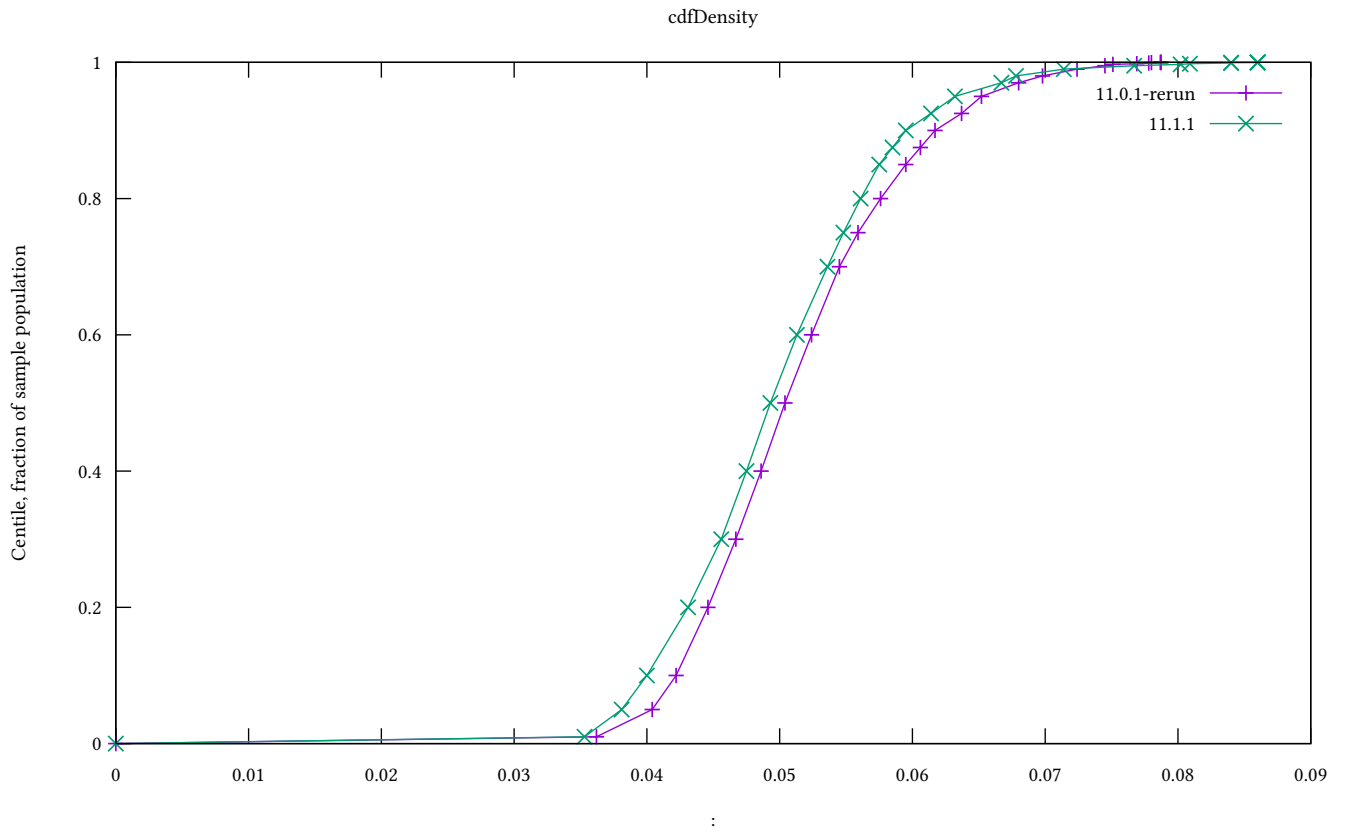


Figure 13: Chain density

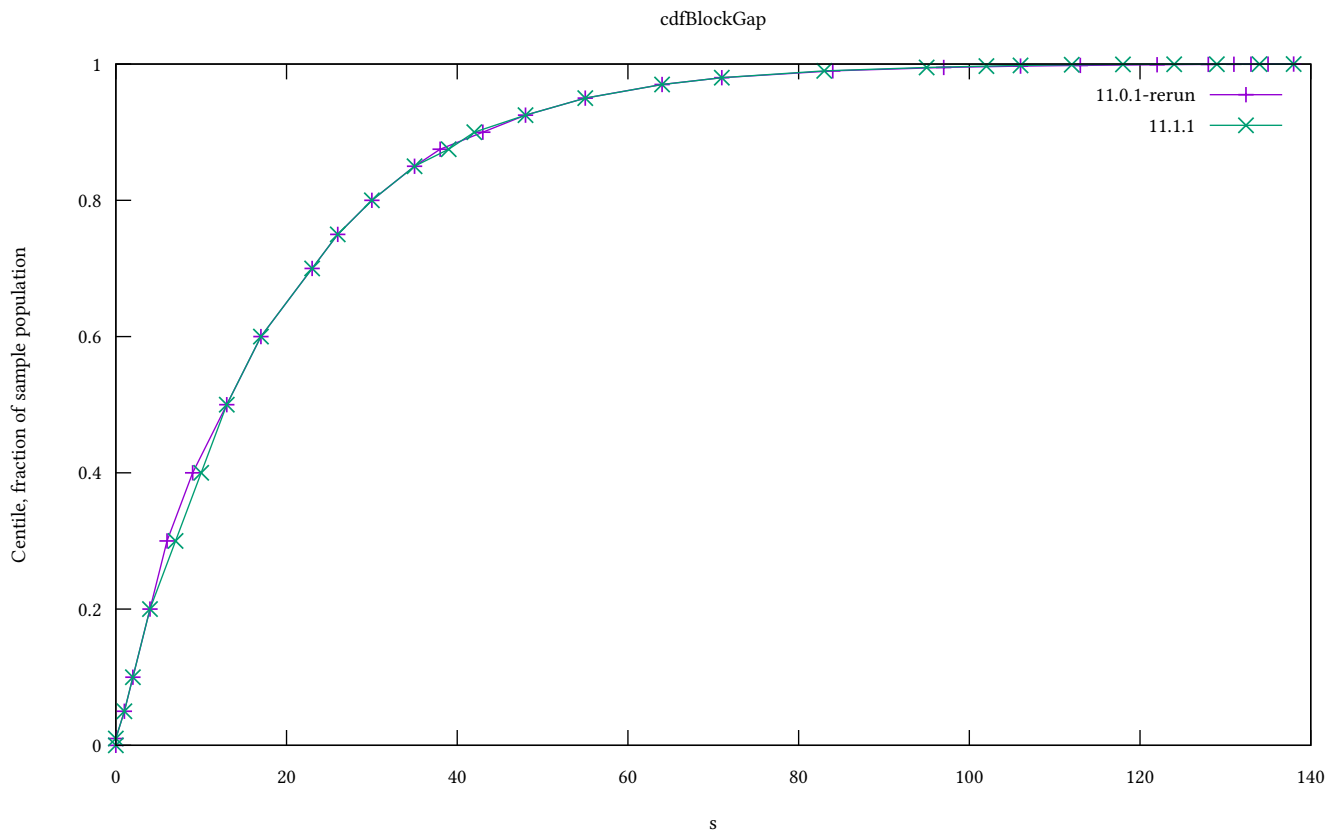


Figure 14: Interblock gap

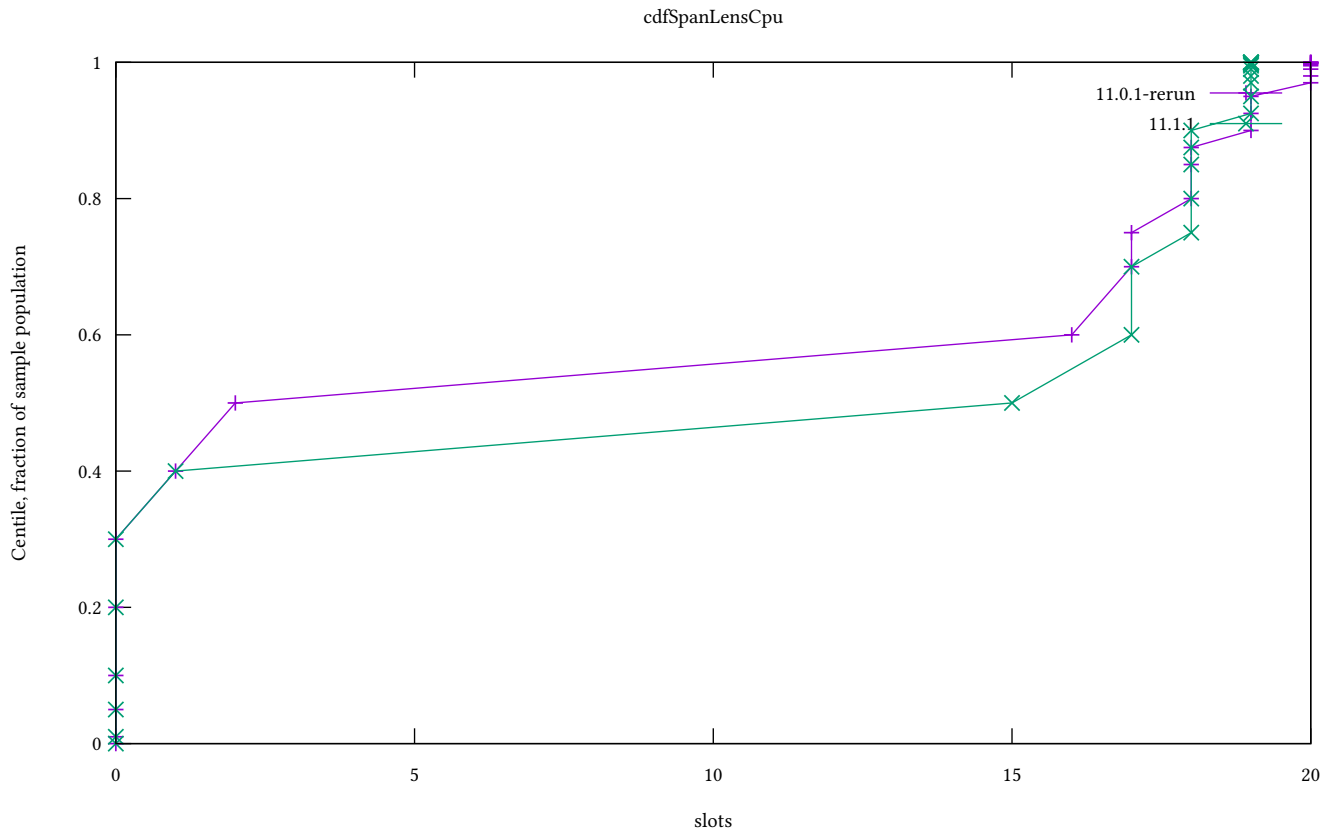


Figure 15: CPU 85% spans

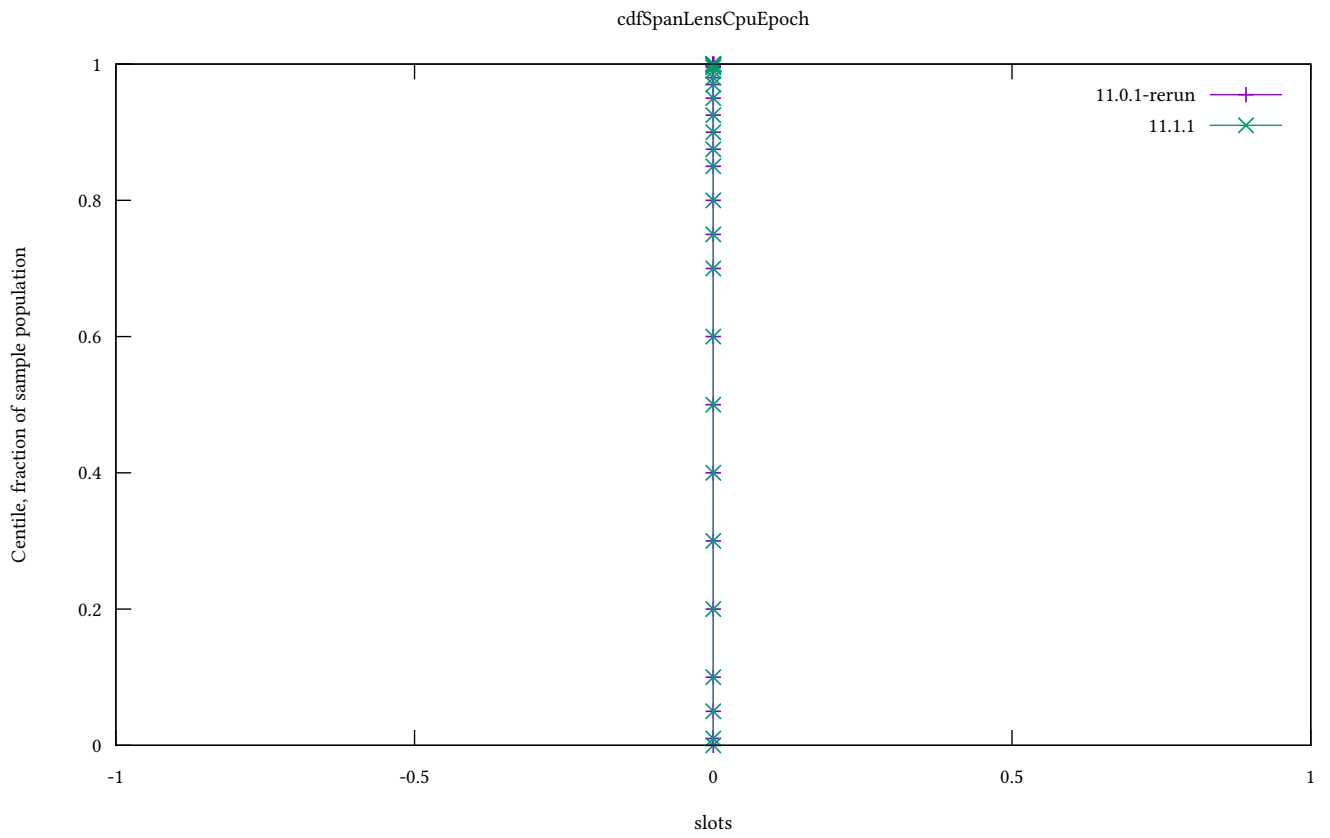


Figure 16: CPU spans at Ep boundary

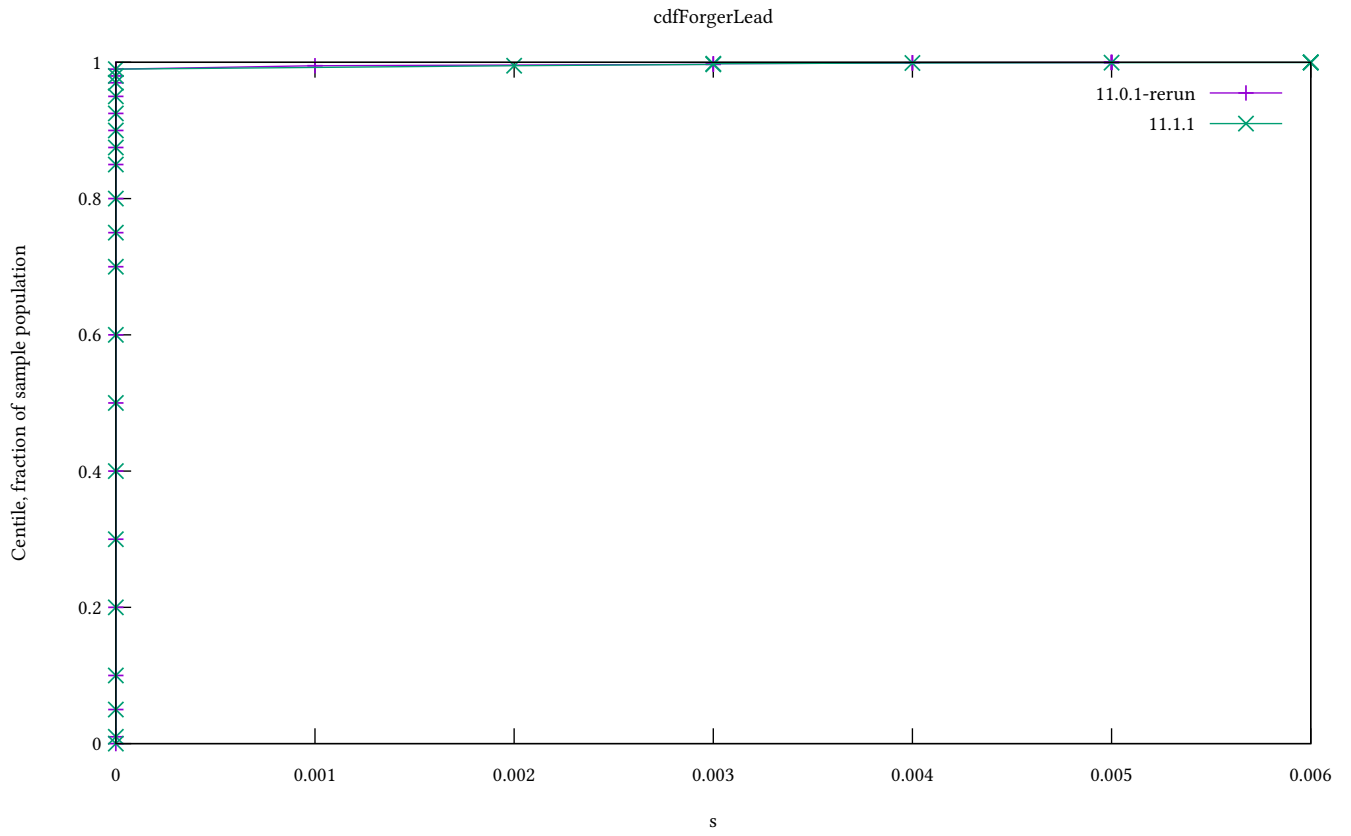


Figure 17: Leadership check duration

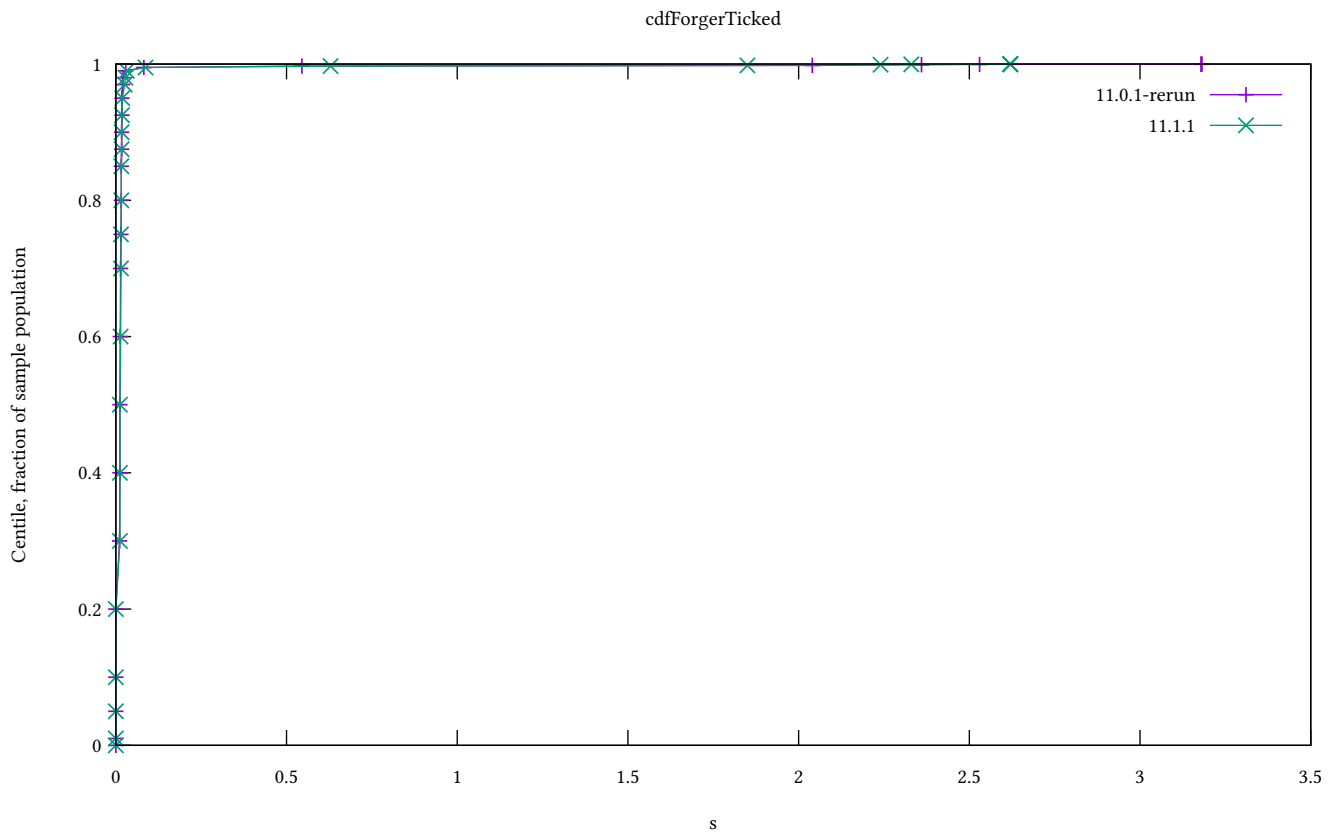


Figure 18: Ledger ticking

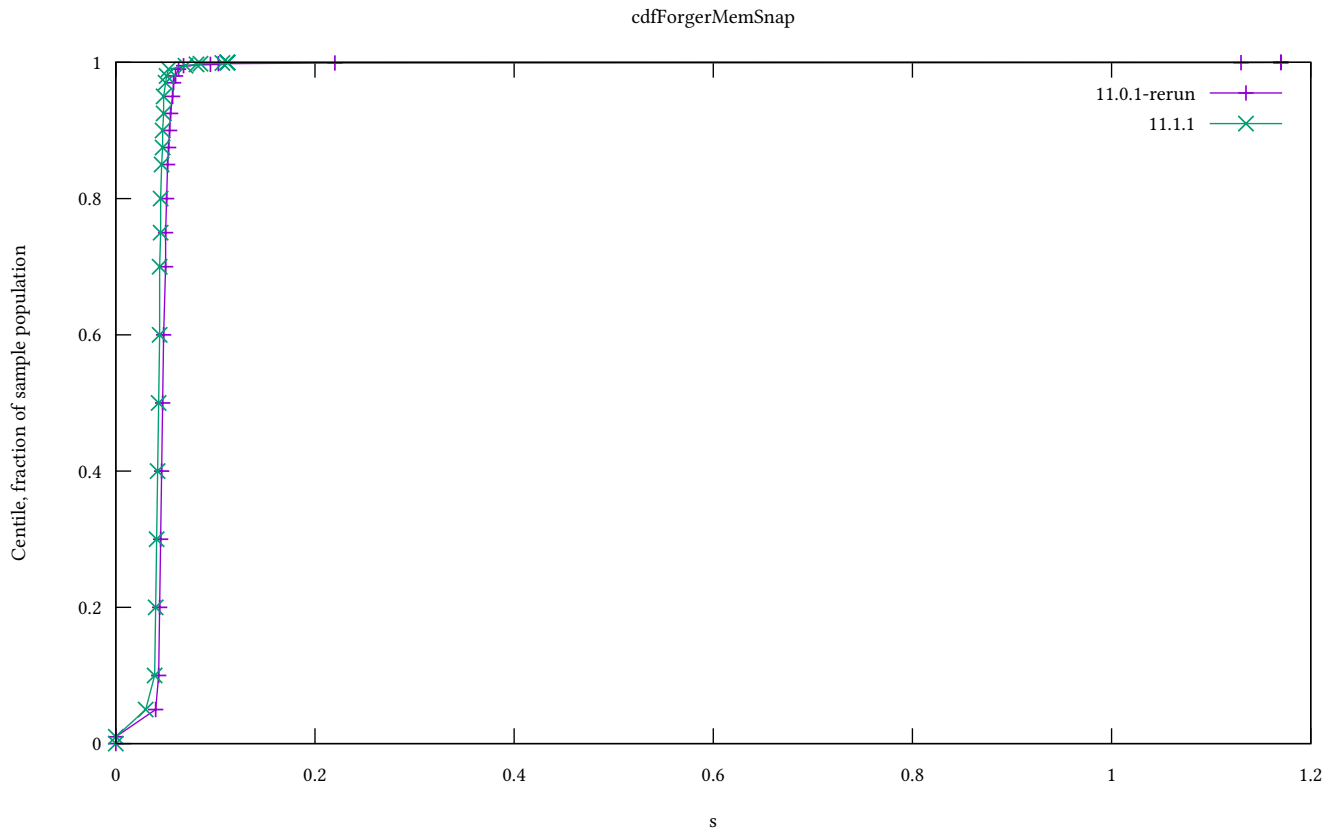


Figure 19: Mempool snapshotting

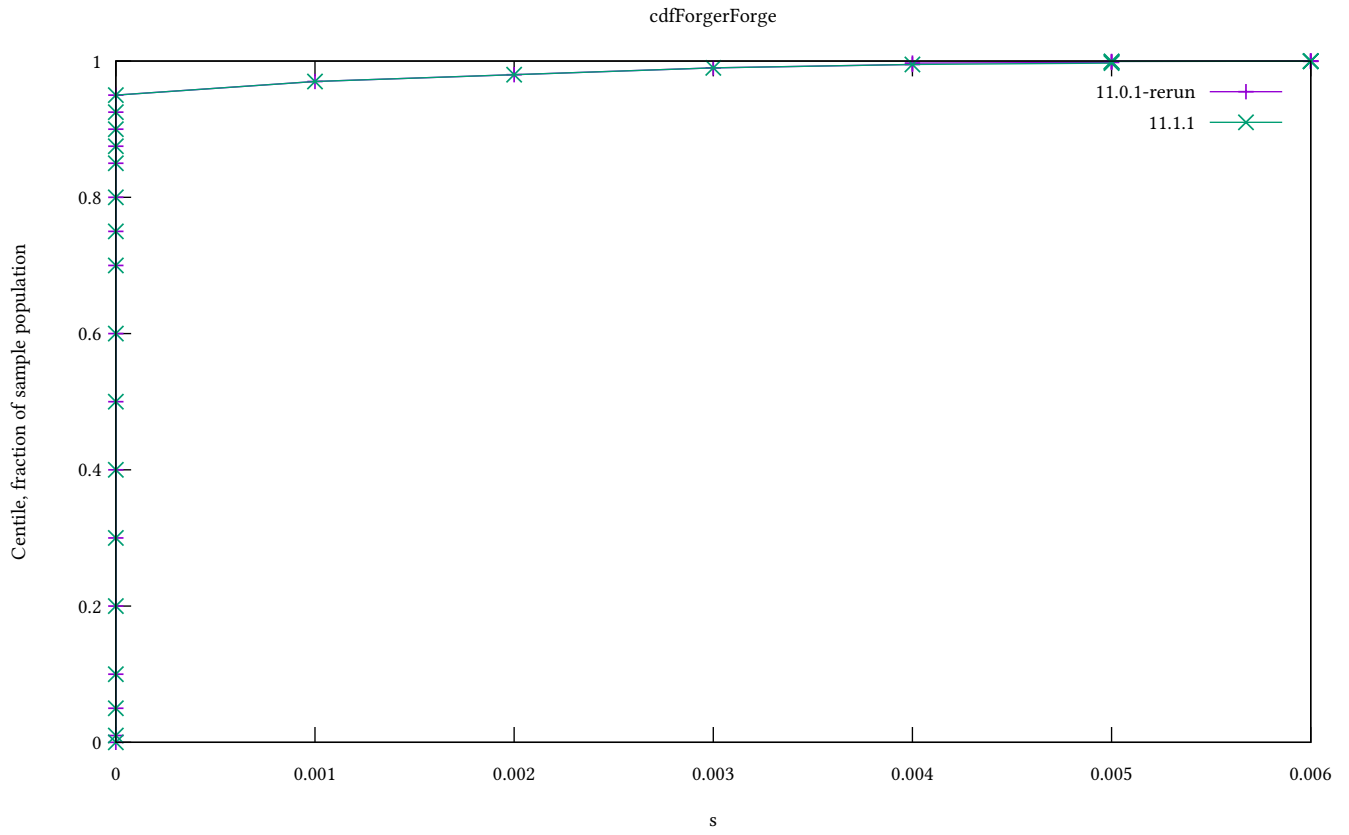


Figure 20: Leadership to forged

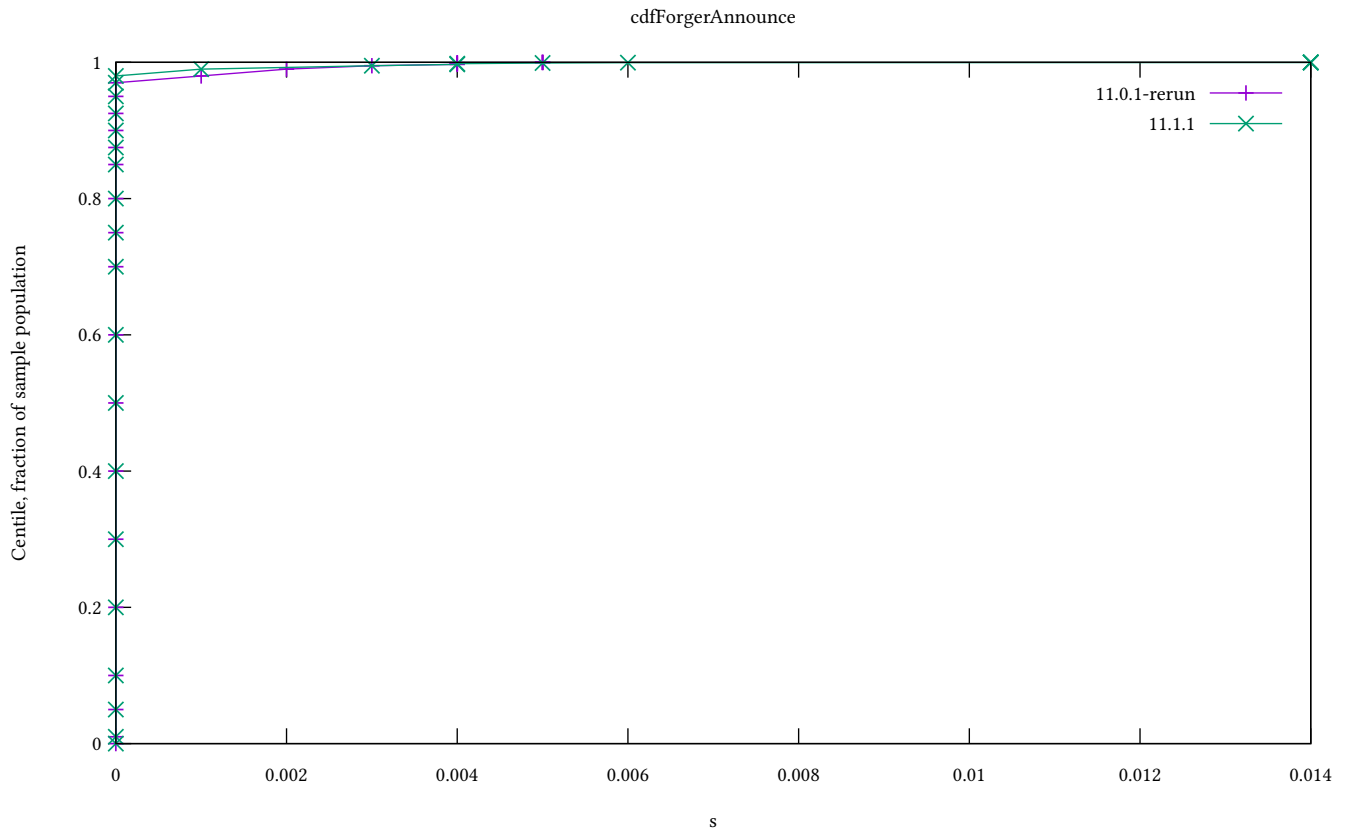


Figure 21: Forged to announced

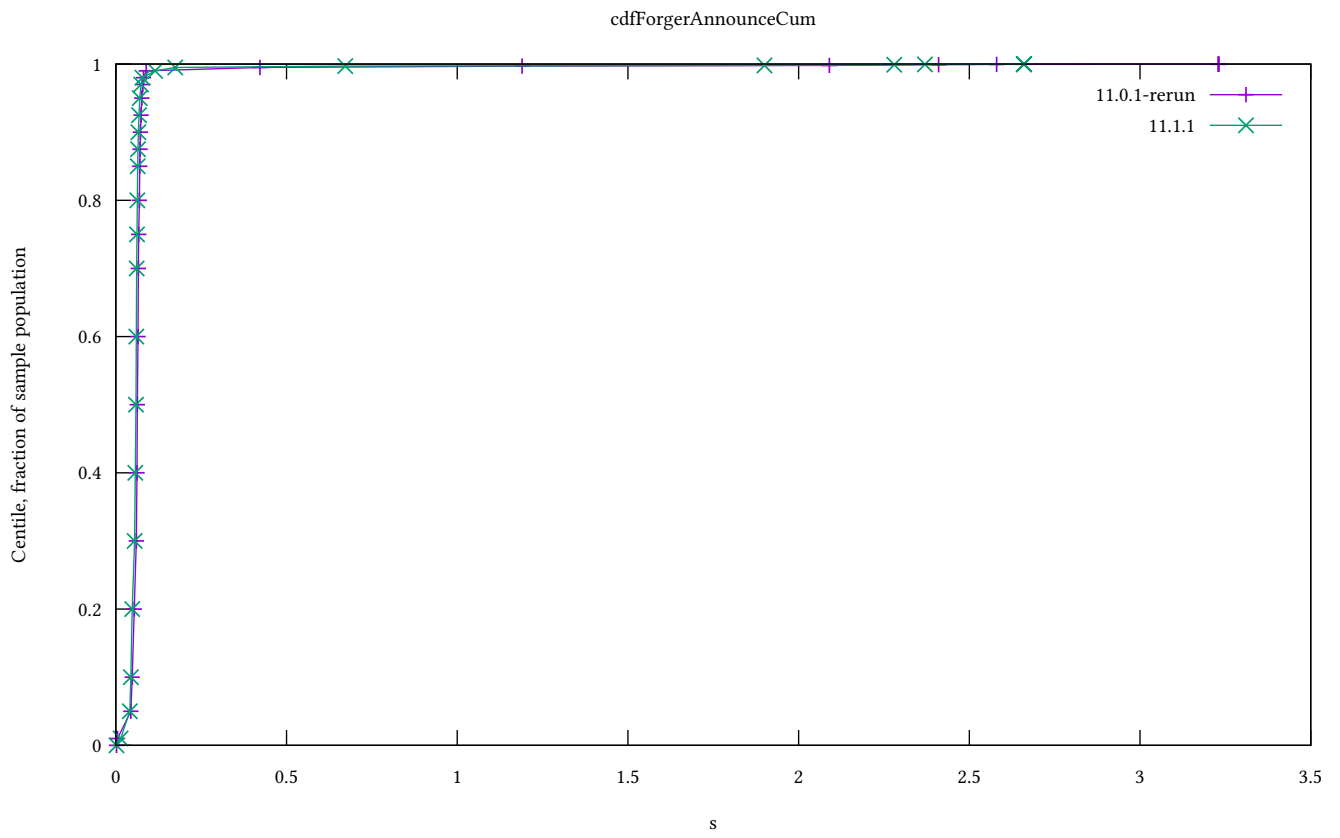


Figure 22: Slot start to announced

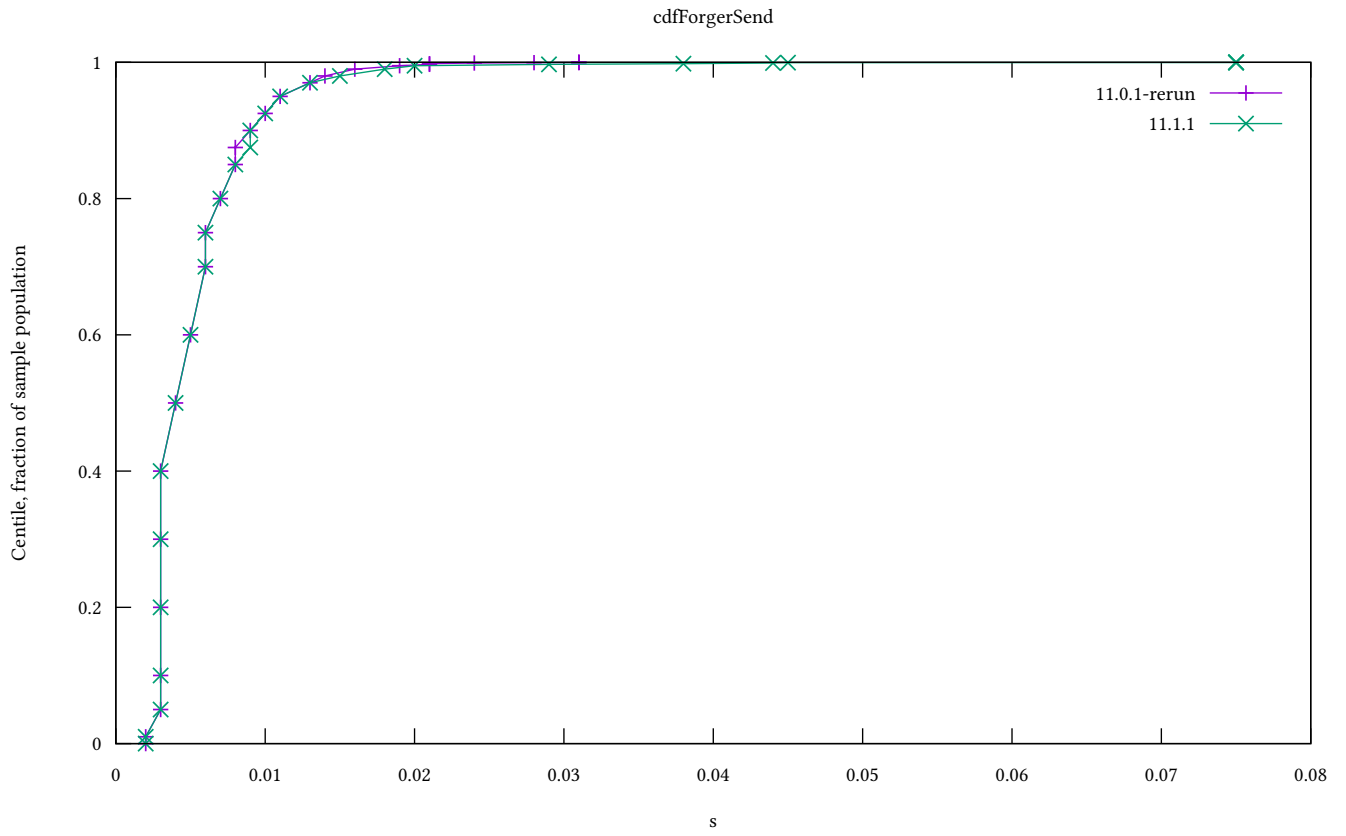


Figure 23: Forged to sending

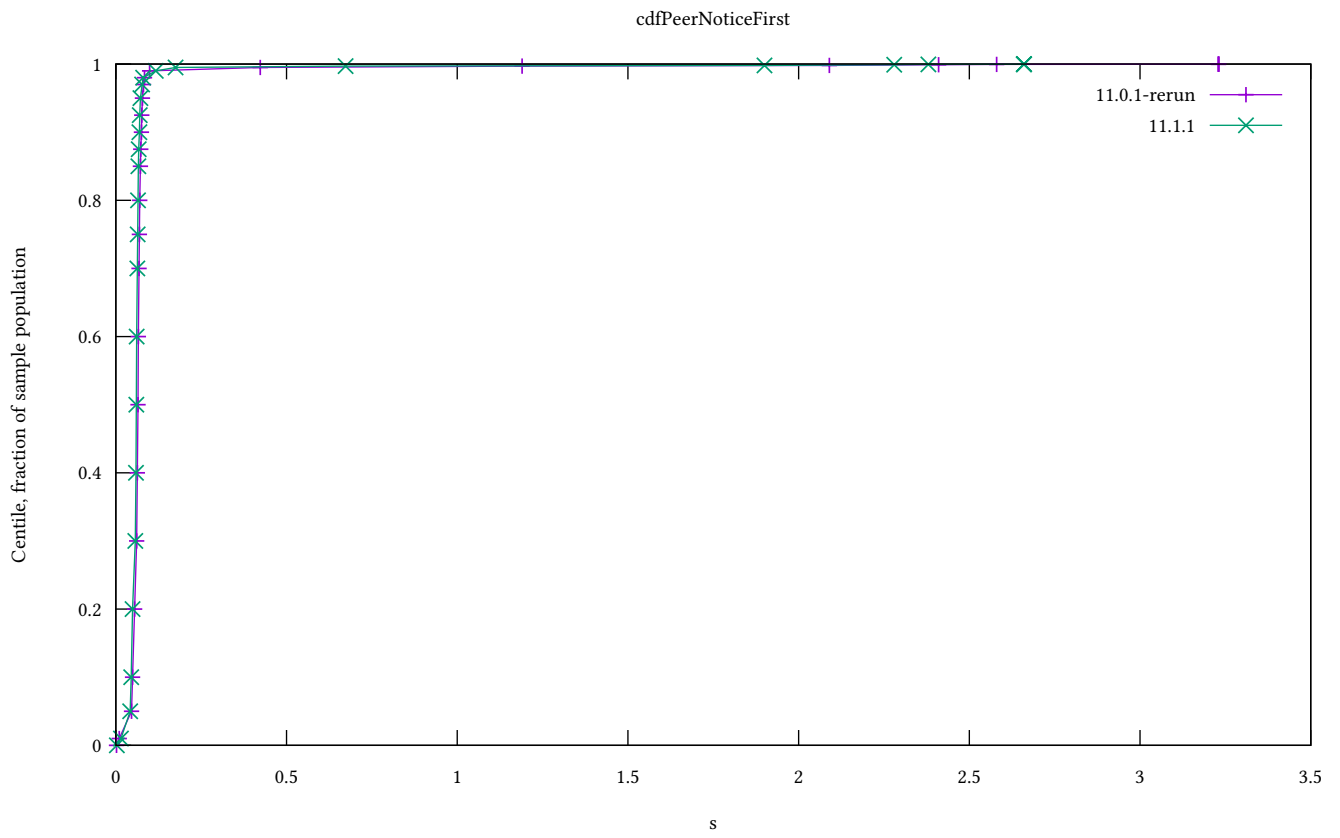


Figure 24: First peer notice

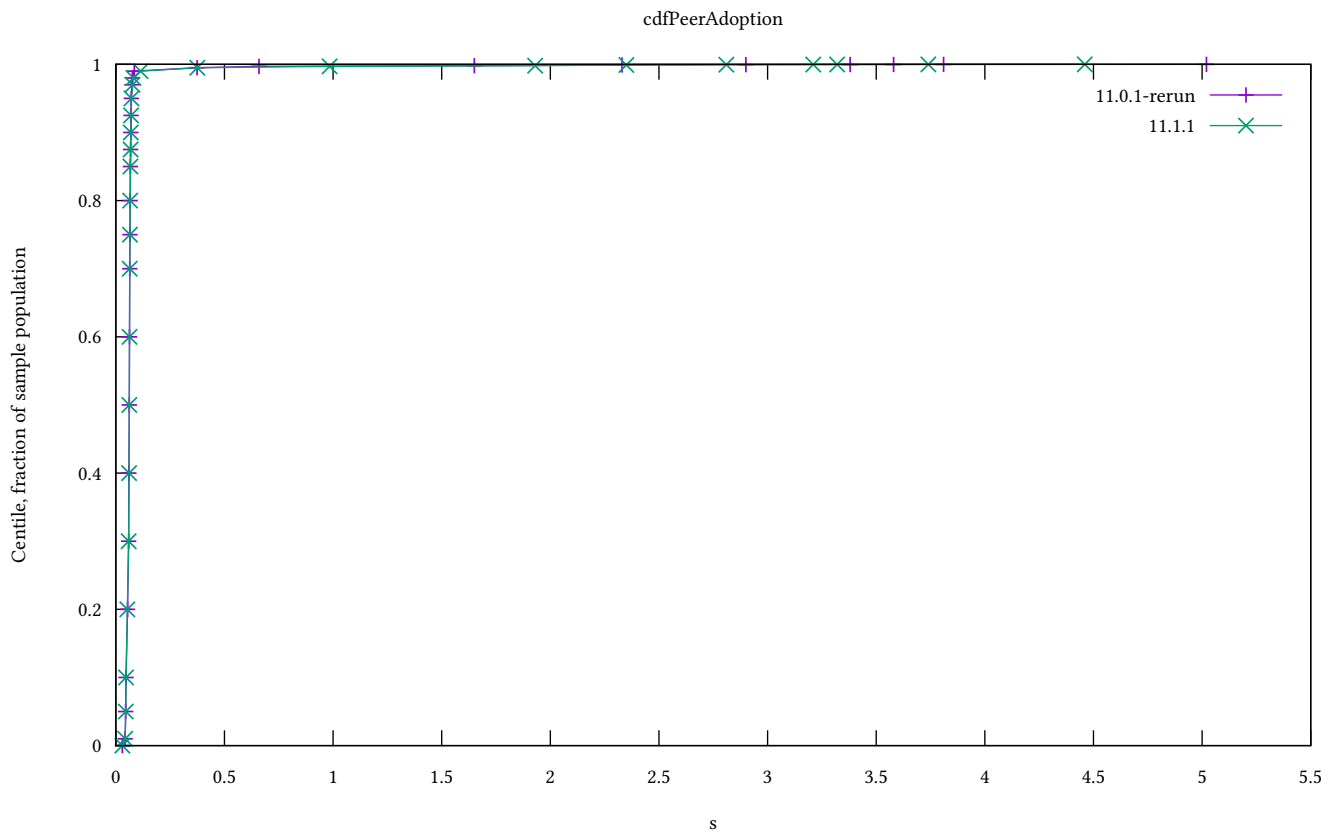


Figure 25: Fetched to adopted

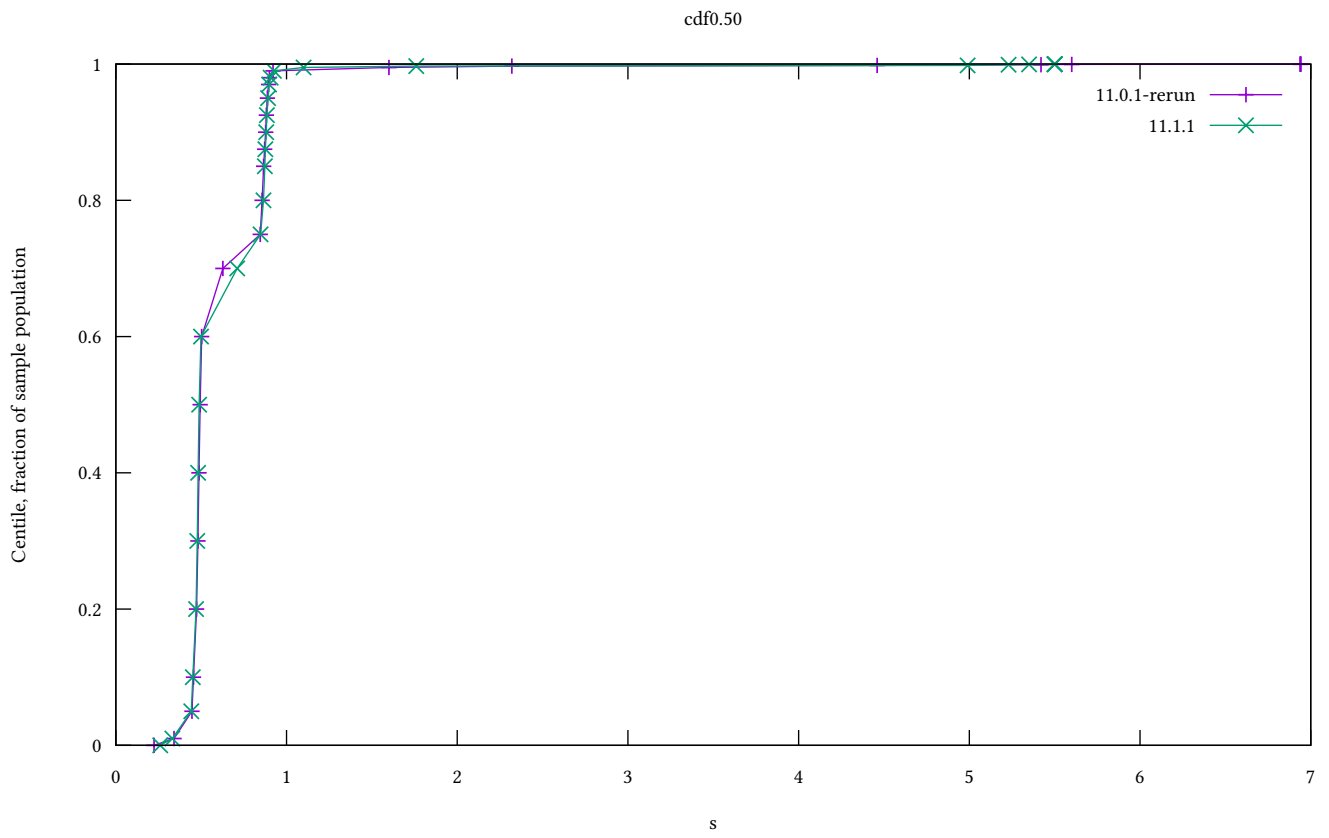


Figure 26: 0.50 adoption

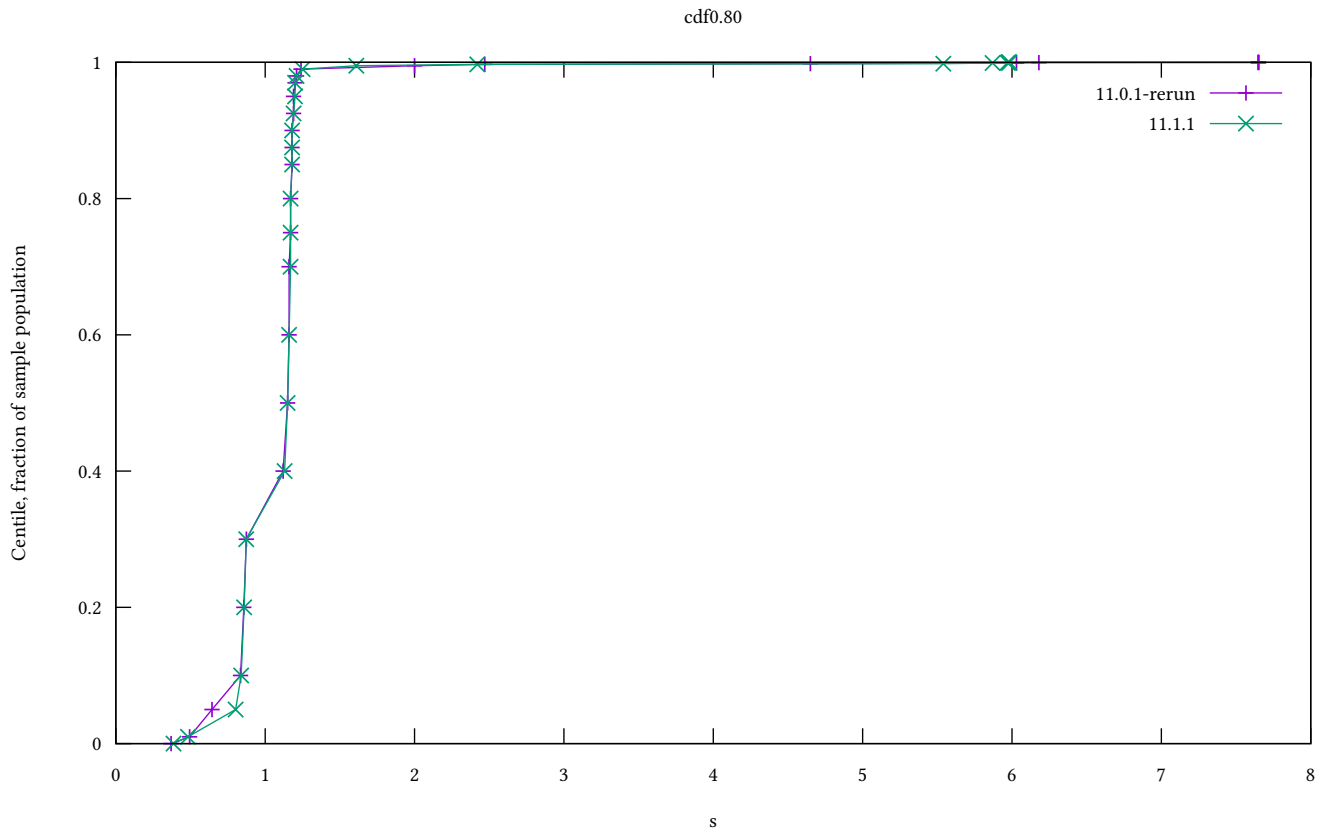


Figure 27: 0.80 adoption

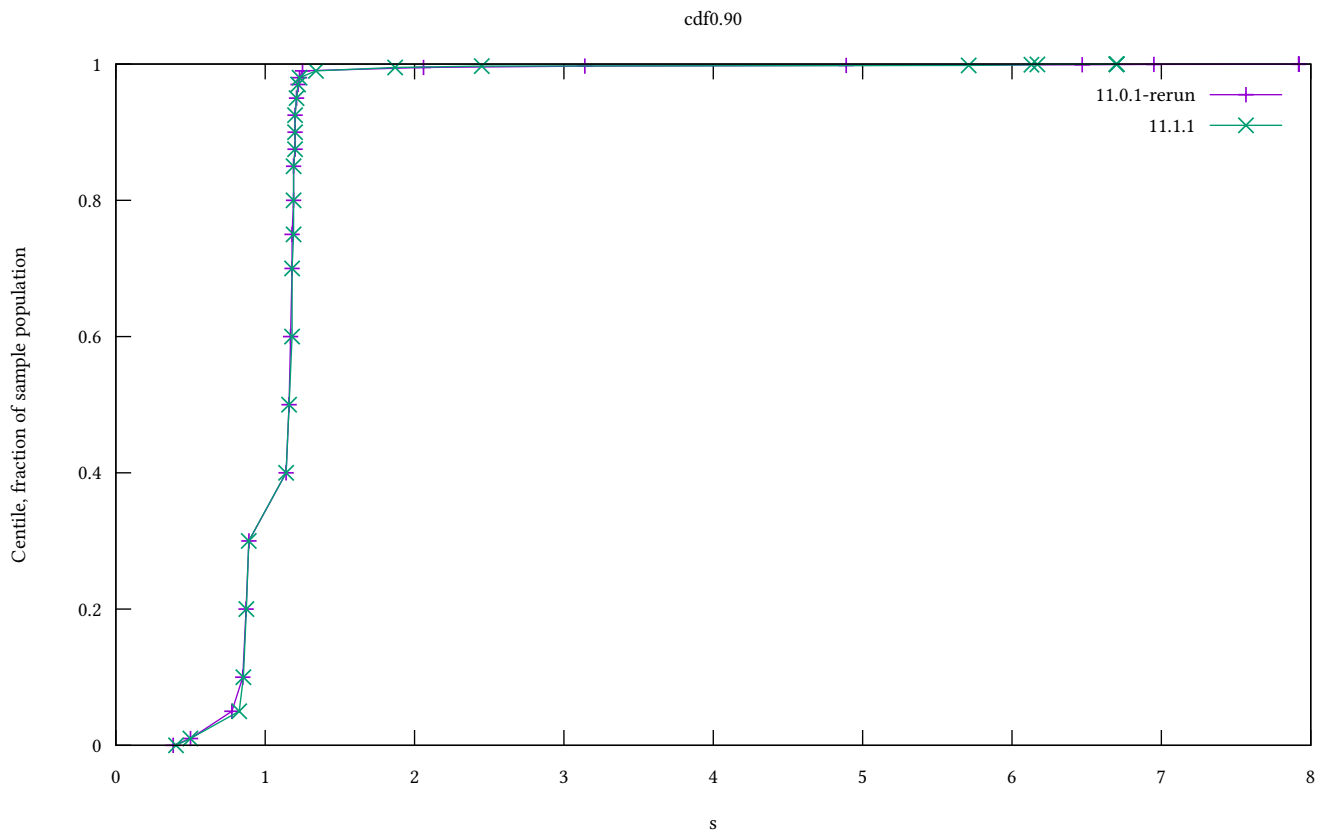


Figure 28: 0.90 adoption

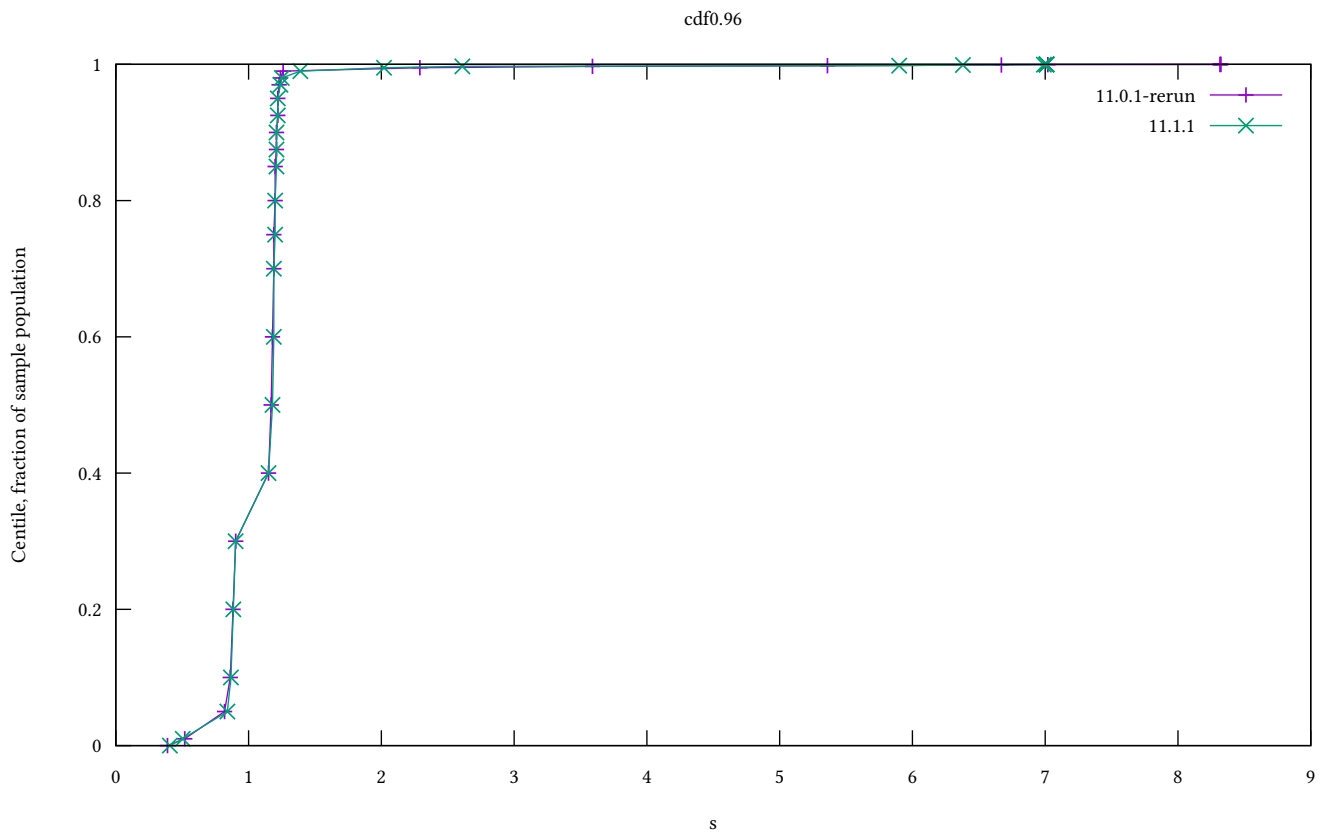


Figure 29: 0.96 adoption

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