

# 10.7.1 against 10.6.2

Plutus countdown loop workload

Michael Karg, Cardano Performance & Tracing

2026-04-17

# Contents

Manifest .....	2
Analysis .....	4
Resource Usage .....	4
Anomaly control .....	4
Forging .....	5
Individual peer propagation .....	5
End-to-end propagation .....	5
Appendix A: charts .....	6
Cluster performance charts .....	6
Appendix B: data dictionary .....	21
Block propagation metrics .....	21
Cluster performance metrics .....	22

## **Manifest**

We compare 10.6.2 (Conway) and 10.7.1 (Conway) relative to 10.6.2 (Conway), under Plutus countdown loop workload.

	10.6.2	10.7.1
Analysis date	2026-02-18	2026-04-17
Cluster system start date	2026-02-17	2026-04-16
Cluster system start time	16:17:10	15:11:39
Identifier	10.6.2	10.7.1
Run batch	1062pargc	10.7.1
GHC version	9.6.7	9.6.7
cardano-node version	10.6.2	10.7.1
ouroboros-consensus version	0.30.0.1	3.0.1.0
ouroboros-network version	0.22.6.0	1.1.0.0
cardano-ledger-core version	1.18.0.0	1.20.0.0
plutus-core version	1.57.0.0	1.61.0.0
cardano-crypto version	1.3.0	1.3.0
cardano-prelude version	0.2.1.0	0.2.2.0
cardano-node git	ac2f15d	045bc18
ouroboros-consensus git	96a9e1b	c87aa76
ouroboros-network git	b49dfd9	a98c885
cardano-ledger-core git	faa7a9d	94e9618
plutus-core git	8ab7a76	6d66070
cardano-crypto git	unknown	a741501
cardano-prelude git	68e015f	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	886109.15384	888579.46153
Log objects analysed per host	611278.32692	610335.44230
Host run time, s	71862.4	71877.9
Host log line rate, Hz	12.330	12.362
Total log objects analysed	31786473	31737443
Run time, s	71867	71881
Analysed run duration, s	56017	56008
Run time efficiency	0.77	0.77
Node start spread, s	5.4046962	5.2099573
Node stop spread, s	2.7185423	2.0063052
Slots analysed	56015	56006
Blocks analysed	2695	2709
Blocks rejected	850	804

## Analysis

### Resource Usage

	10.6.2	10.7.1	$\Delta$	$\Delta\%$
Forge loop starts, units	0.99993	0.99996	0.000	0
Process CPU usage, %	3.8979	1.1368	-2.761	-71
RTS GC CPU usage, %	0.2748	0.2303	-0.045	-16
RTS Mutator CPU usage, %	3.6196	0.9058	-2.714	-75
Major GCs, events	0.0011	0.0009	-0.000	-18
Minor GCs, events	0.3486	0.3351	-0.014	-4
Kernel RSS, MiB	6460.	6597.9	137.819	2
RTS heap size, MiB	6397.	6528.9	131.828	2
RTS live GC dataset, MiB	2524.6	2689.8	165.229	7
RTS alloc rate, MiB/s	10.321	9.9105	-0.411	-4
Filesystem reads, KiB/s	0.0	0.0	0.000	NaN
Filesystem writes, KiB/s	222.78	207.7	-15.083	-7
CPU 85% spans, slots	8.7969	7.4508	-1.346	-15

### Anomaly control

	10.6.2	10.7.1	$\Delta$	$\Delta\%$
Blocks per host, blocks	69.942	69.288	-0.654	-0.9
Filtered to chained blocks, :	0.7604	0.7691	0.009	1
Chained to forged blocks, :	0.9751	0.9748	-0.000	-0.0
Height & slot battles, blocks	0.0	0.0	0.000	NaN
Block size, Bytes	2996.	2996.1	0.019	0.0

## Forging

	10.6.2	10.7.1	$\Delta$	$\Delta\%$
Started forge loop iteration, s	0.00172	0.00229	0.001	33
Acquired block context, s	7.0e-5	6.4e-5	-0.000	0
Acquired ledger state, s	0.00013	0.00011	-0.000	0
Acquired ledger view, s	3.0e-5	2.9e-5	-0.000	0
Leadership check duration, s	0.00041	0.00042	0.000	0
Ledger ticking, s	0.02066	0.01685	-0.004	-18
Mempool snapshotting, s	0.0017	0.00165	-0.000	0
Leadership to forged, s	0.00018	0.00021	0.000	0
Forged to announced, s	0.0006	0.00056	-0.000	0
Forged to sending, s	0.00591	0.00492	-0.001	-17
Forged to self-adopted, s	0.04219	0.04035	-0.002	-4
Slot start to announced, s	0.02551	0.02218	-0.003	-13

## Individual peer propagation

	10.6.2	10.7.1	$\Delta$	$\Delta\%$
First peer notice, s	0.0273	0.02372	-0.004	-13
First peer fetch, s	0.03421	0.0296	-0.005	-13
Notice to fetch request, s	0.00126	0.00129	0.000	0
Fetch duration, s	0.12277	0.12306	0.000	0.2
Fetches to announced, s	0.00071	0.00067	-0.000	0
Fetches to sending, s	0.0433	0.04248	-0.001	-2
Fetches to adopted, s	0.04431	0.04086	-0.003	-8

## End-to-end propagation

	10.6.2	10.7.1	$\Delta$	$\Delta\%$
0.50 adoption, s	0.28493	0.27713	-0.008	-3
0.80 adoption, s	0.44396	0.43613	-0.008	-2
0.90 adoption, s	0.45524	0.44745	-0.008	-2
0.92 adoption, s	0.4582	0.45019	-0.008	-2
0.94 adoption, s	0.46186	0.45297	-0.009	-2
0.96 adoption, s	0.46691	0.45778	-0.009	-2
0.98 adoption, s	0.47473	0.46495	-0.010	-2
1.00 adoption, s	0.50579	0.48973	-0.016	-3

# Appendix A: charts

## Cluster performance charts

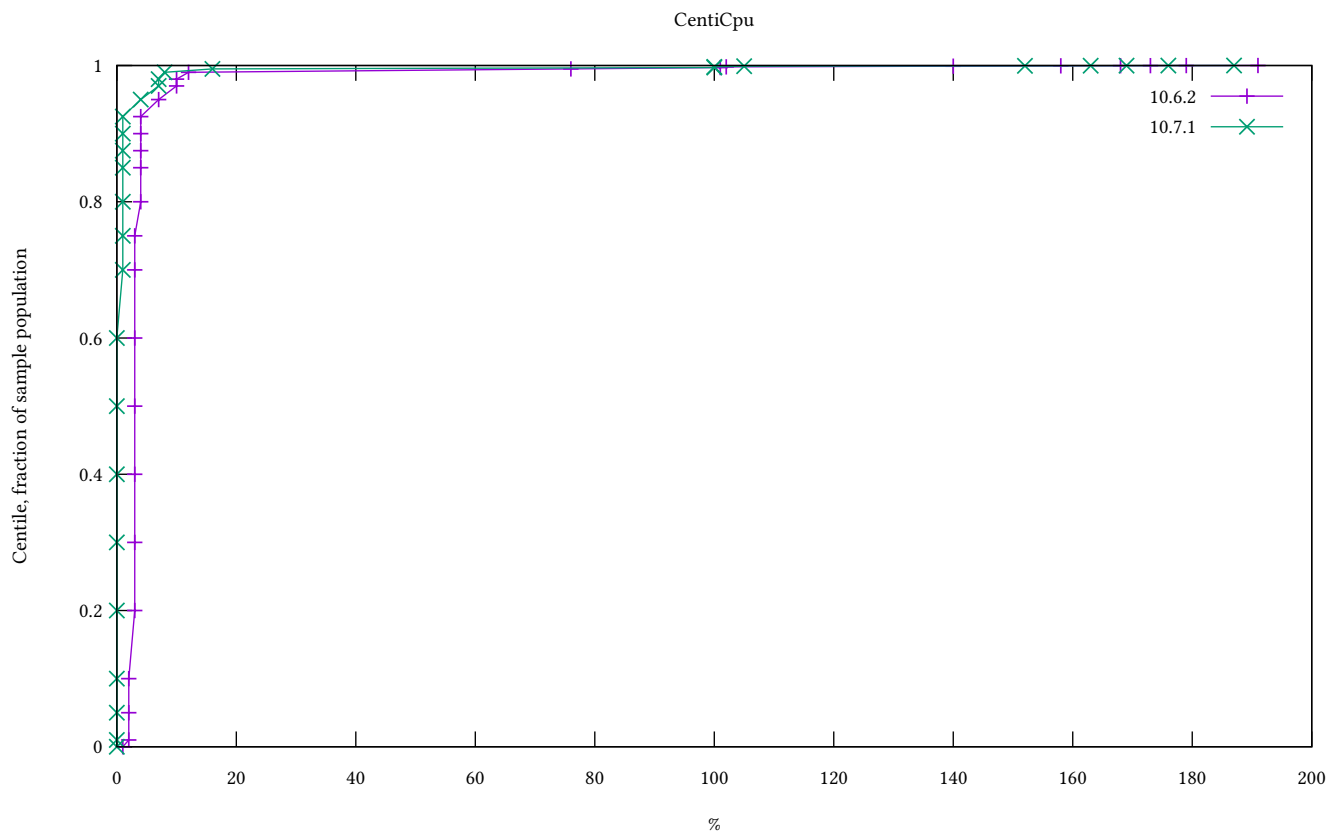


Figure 1: Process CPU usage

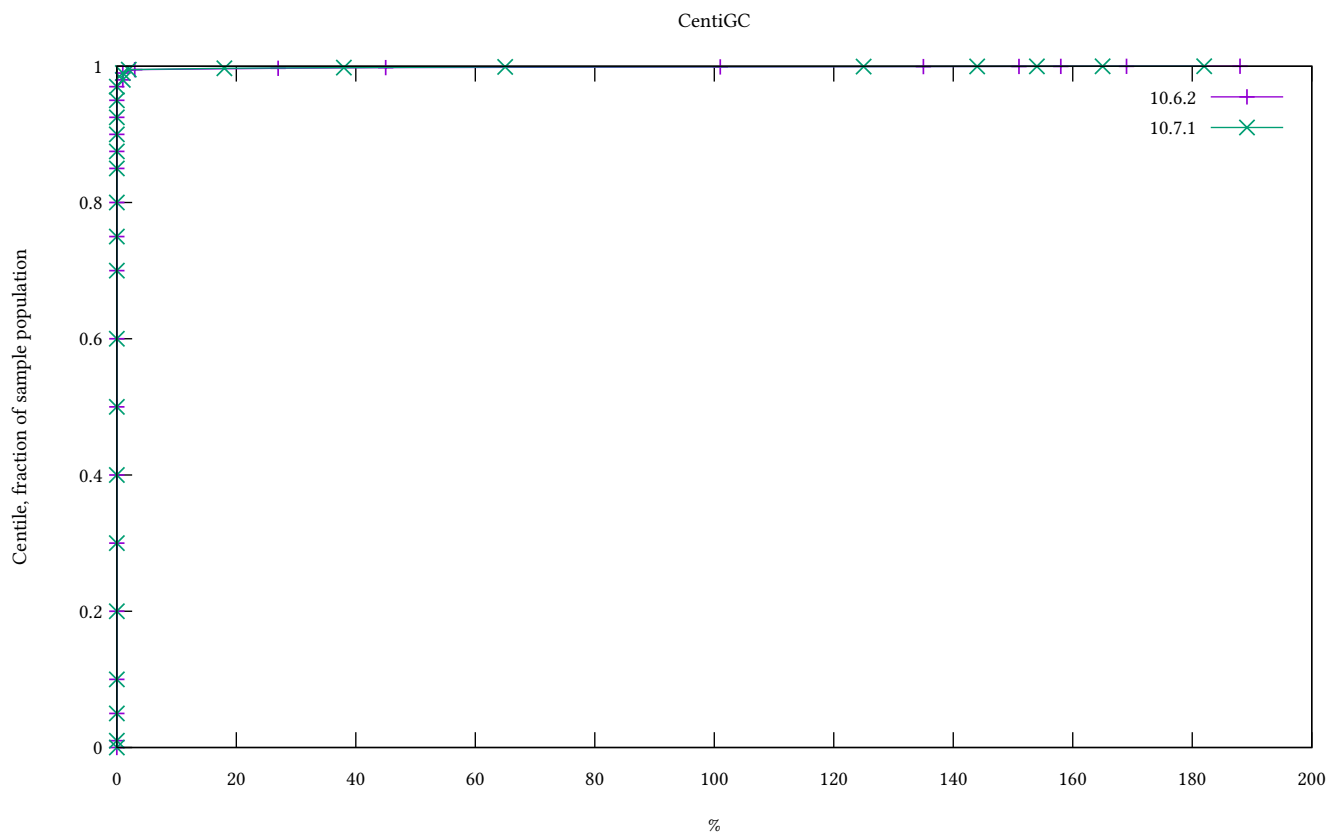


Figure 2: RTS GC CPU usage

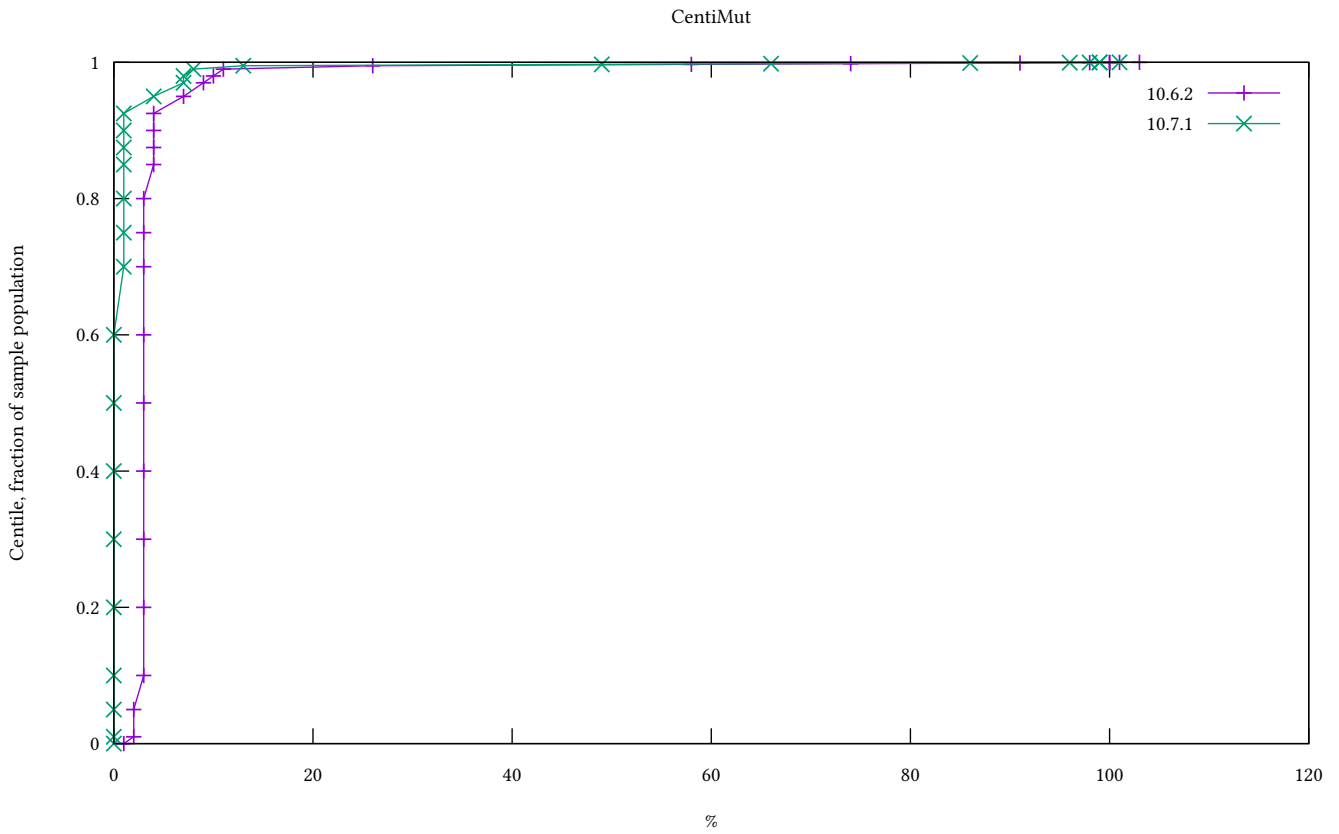


Figure 3: RTS Mutator CPU usage

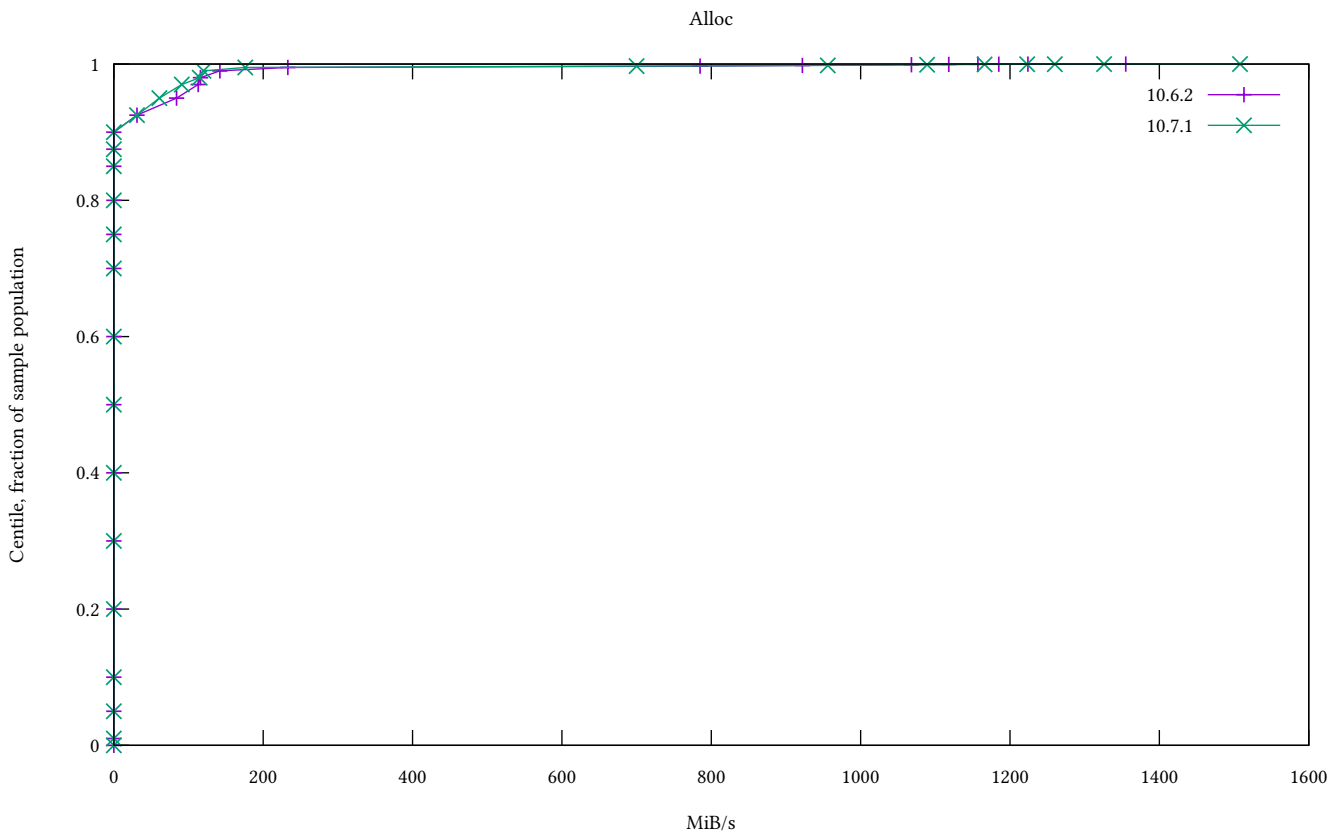
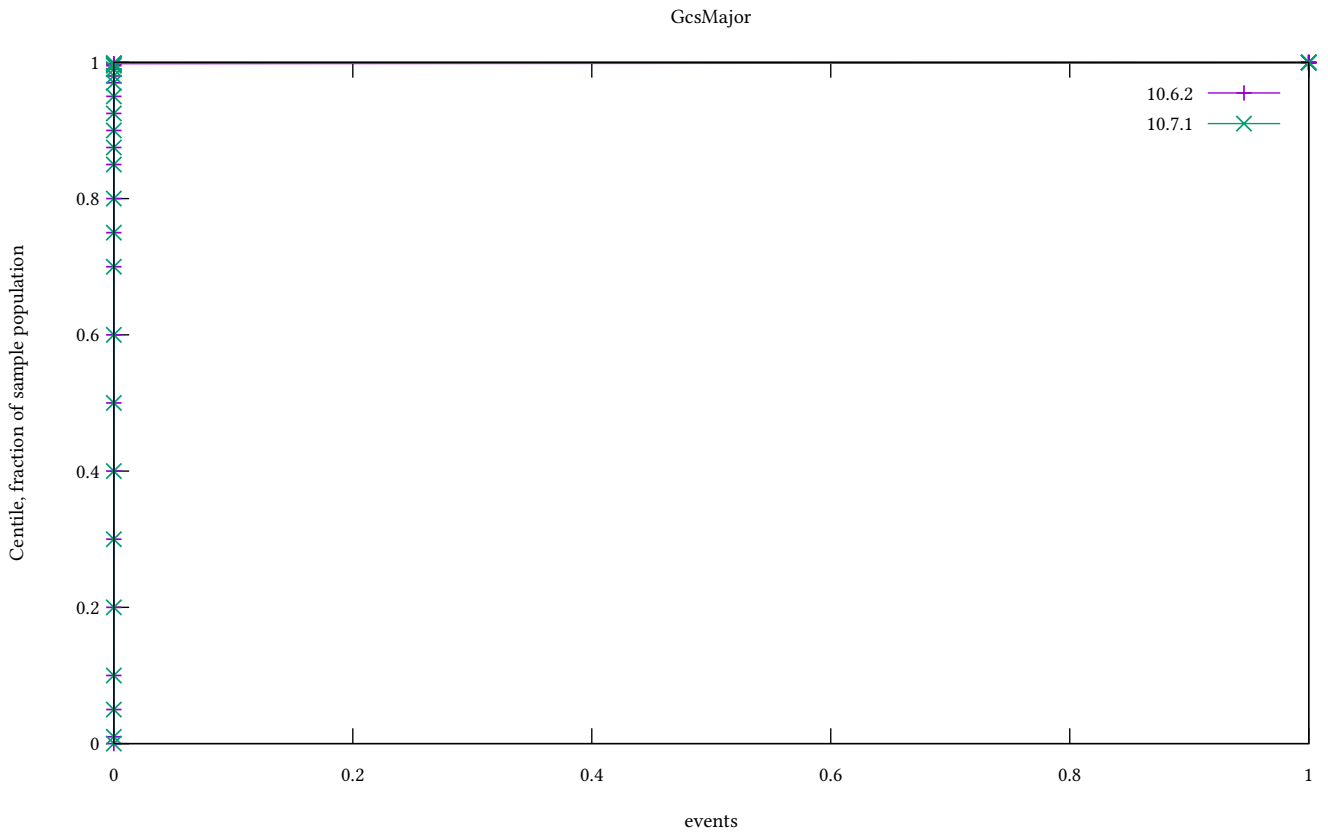
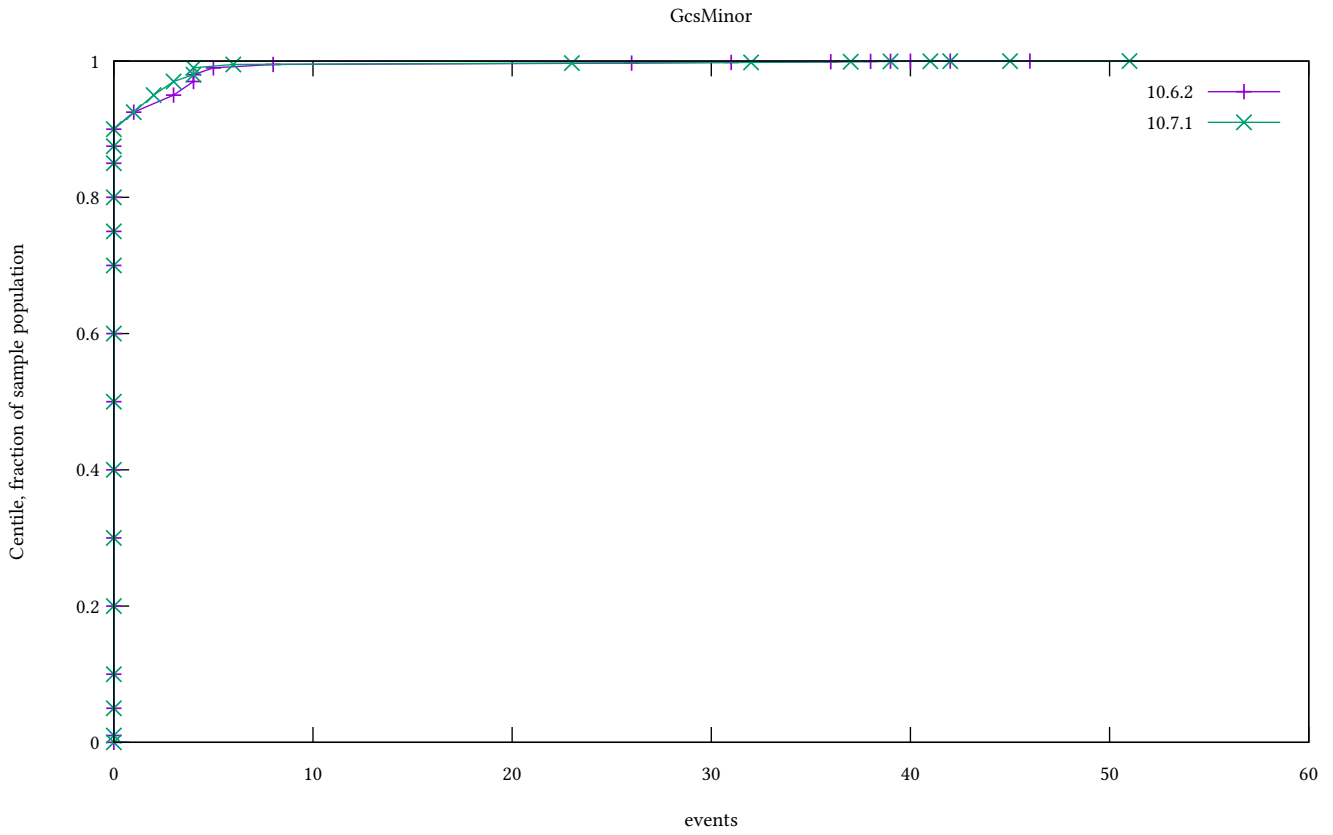


Figure 4: RTS alloc rate



*Figure 5: Major GCs*



*Figure 6: Minor GCs*

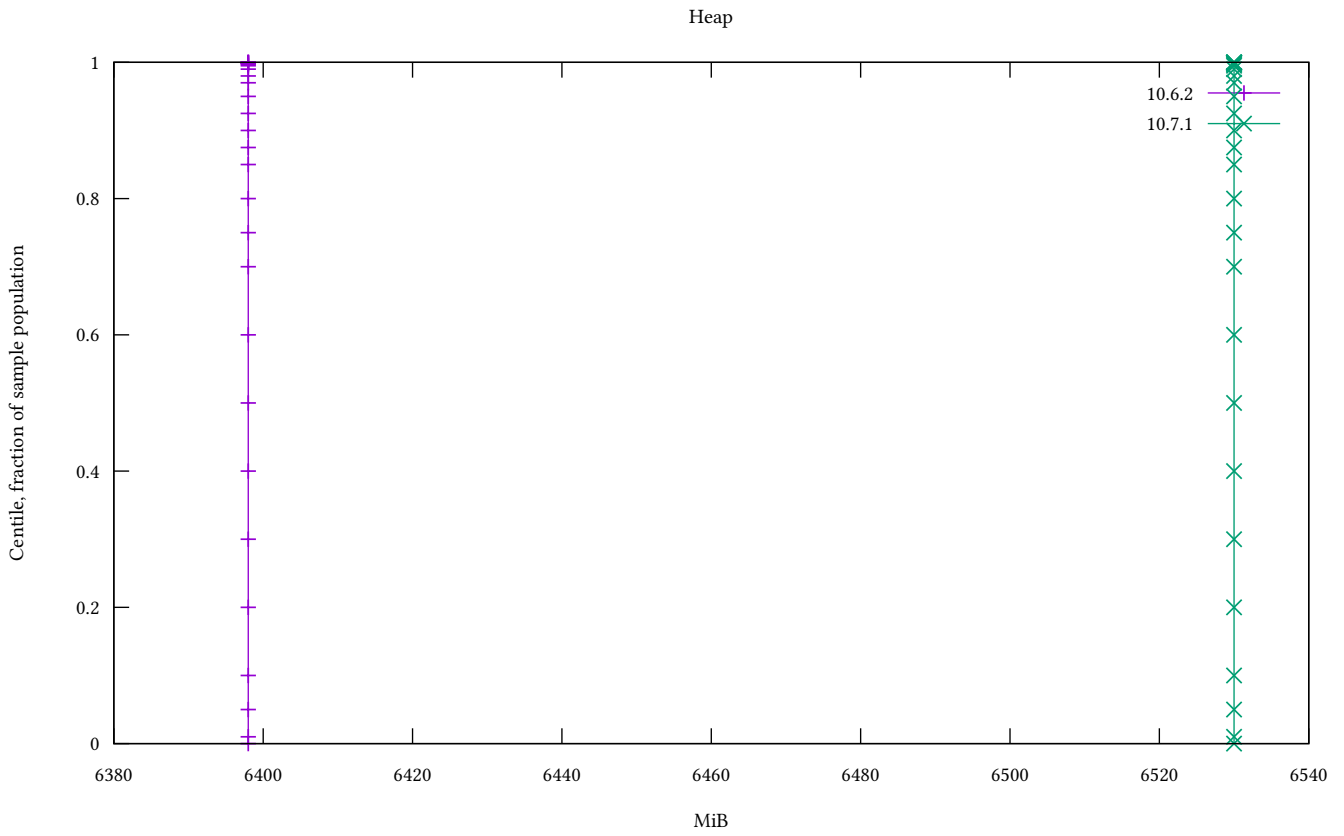


Figure 7: RTS heap size

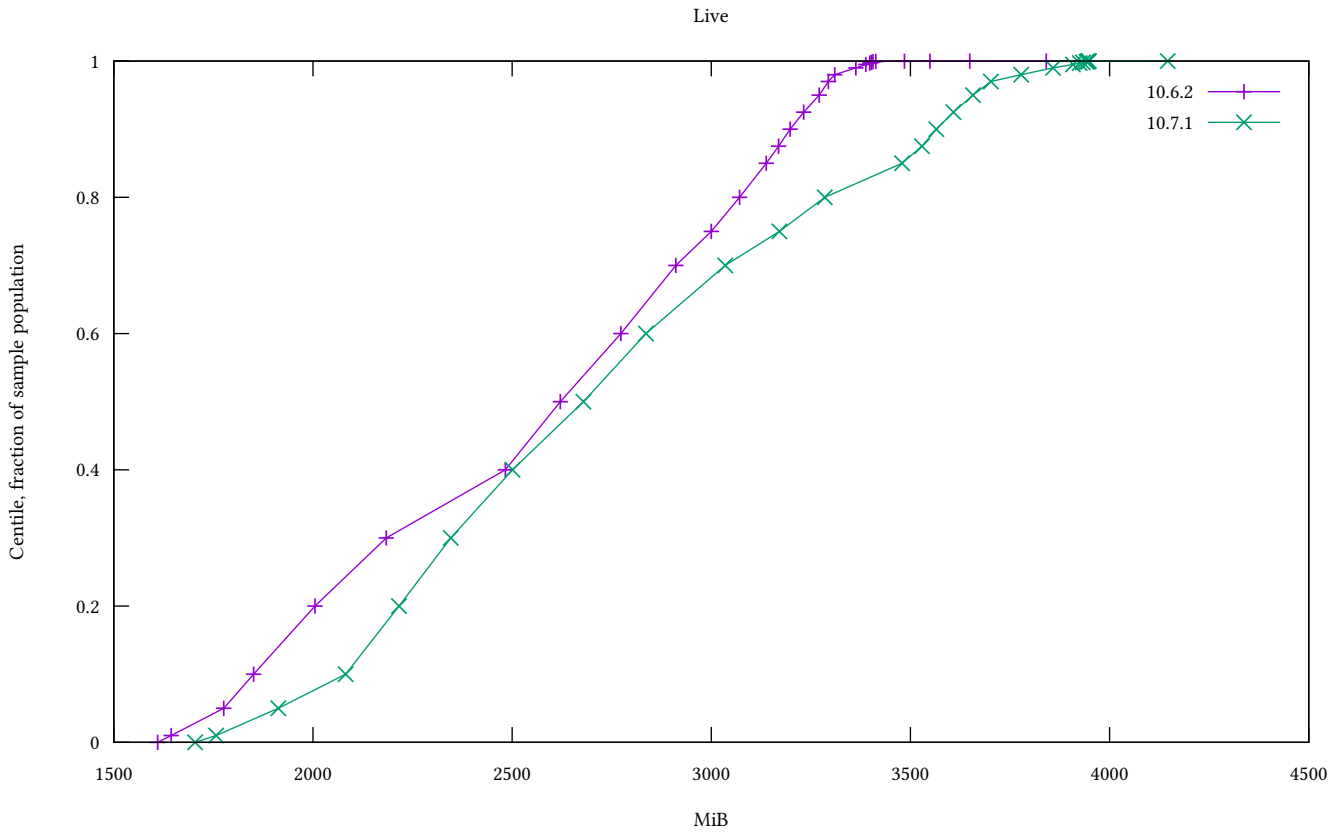


Figure 8: RTS live GC dataset

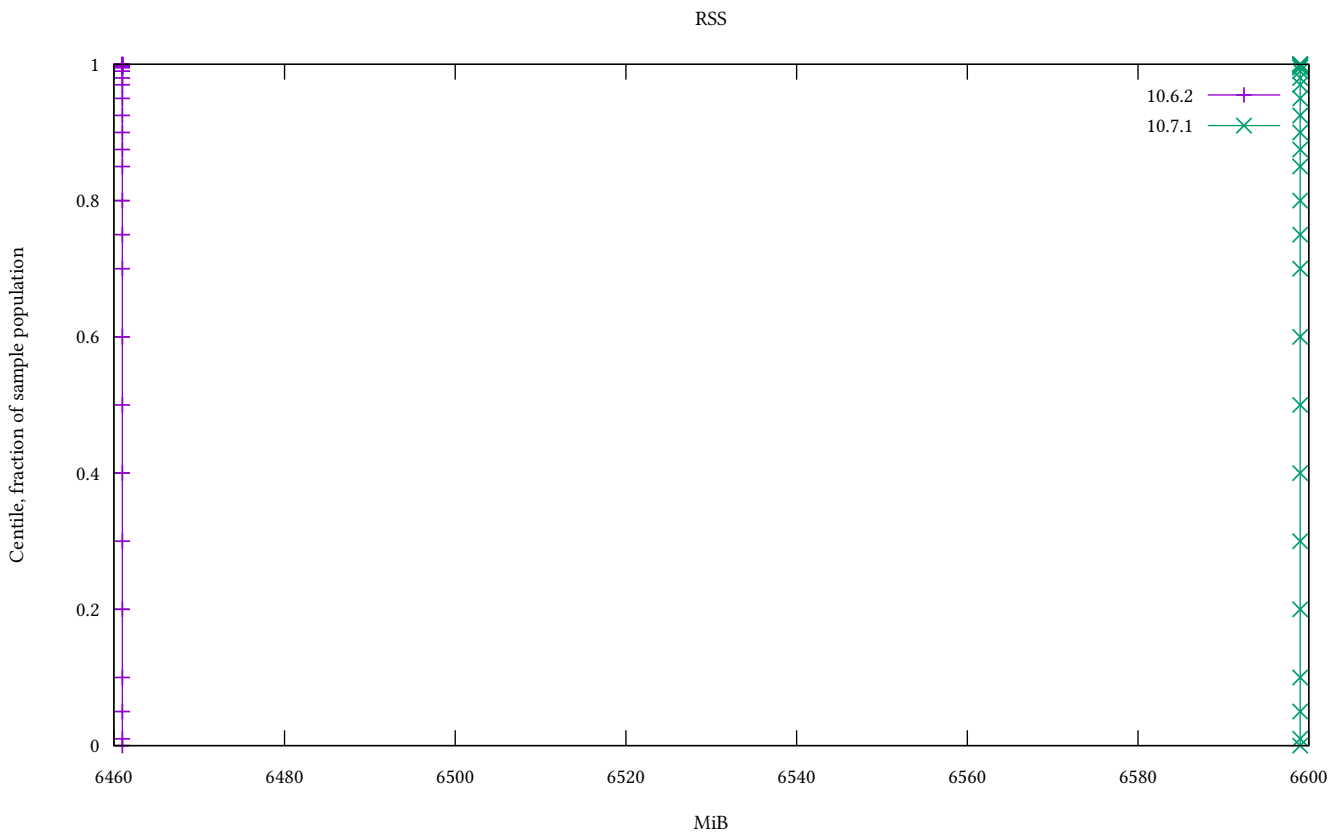


Figure 9: Kernel RSS

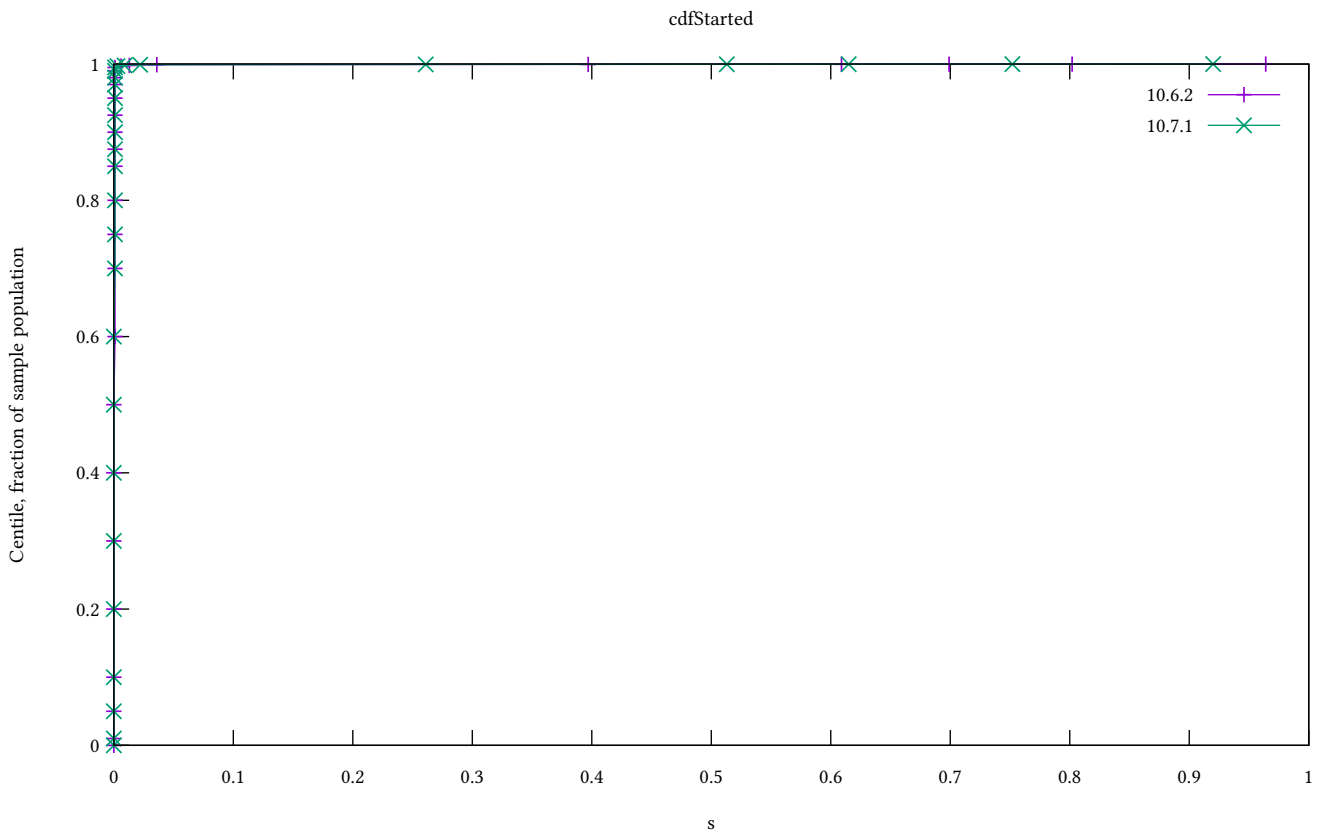


Figure 10: Forge loop tardiness

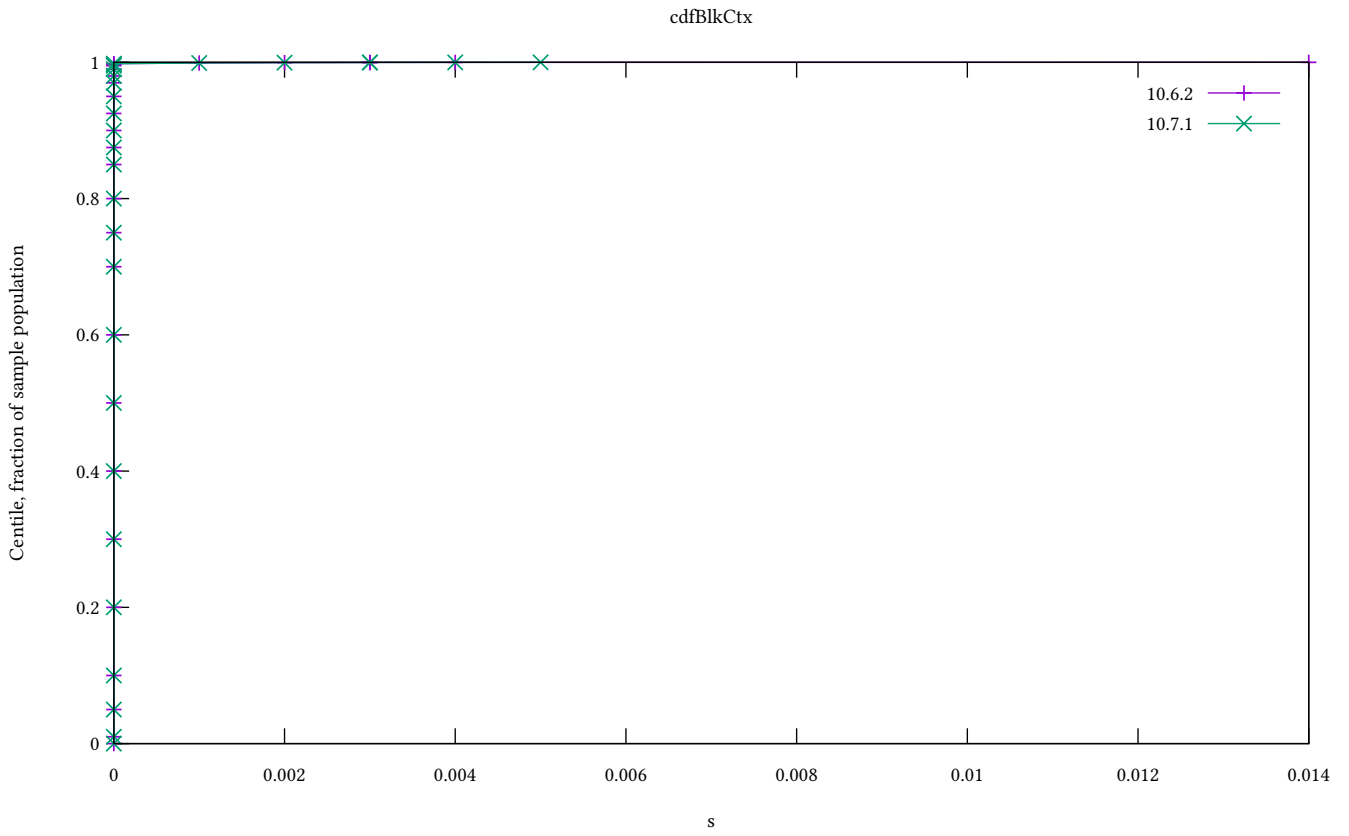


Figure 11: Block context acquisition delay

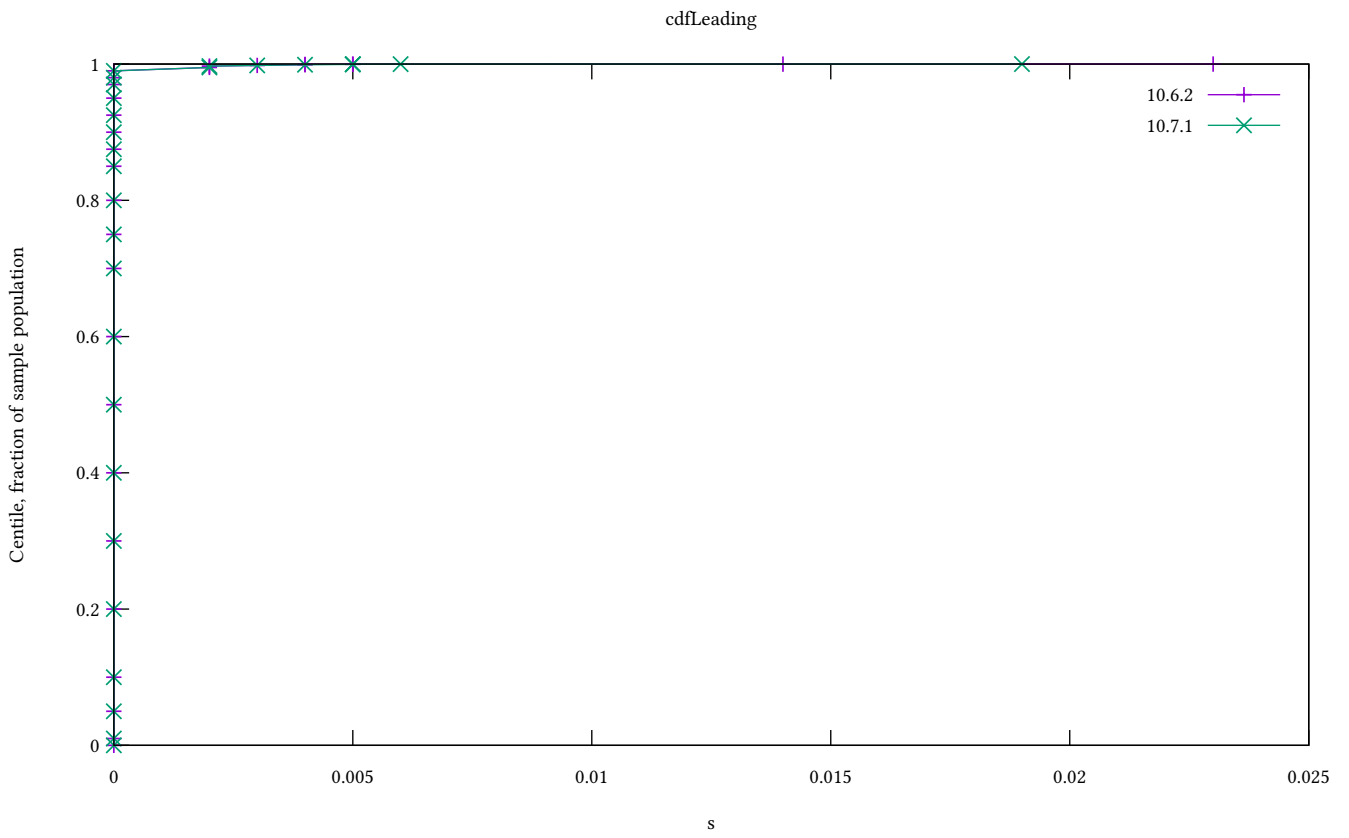


Figure 12: Leadership check duration

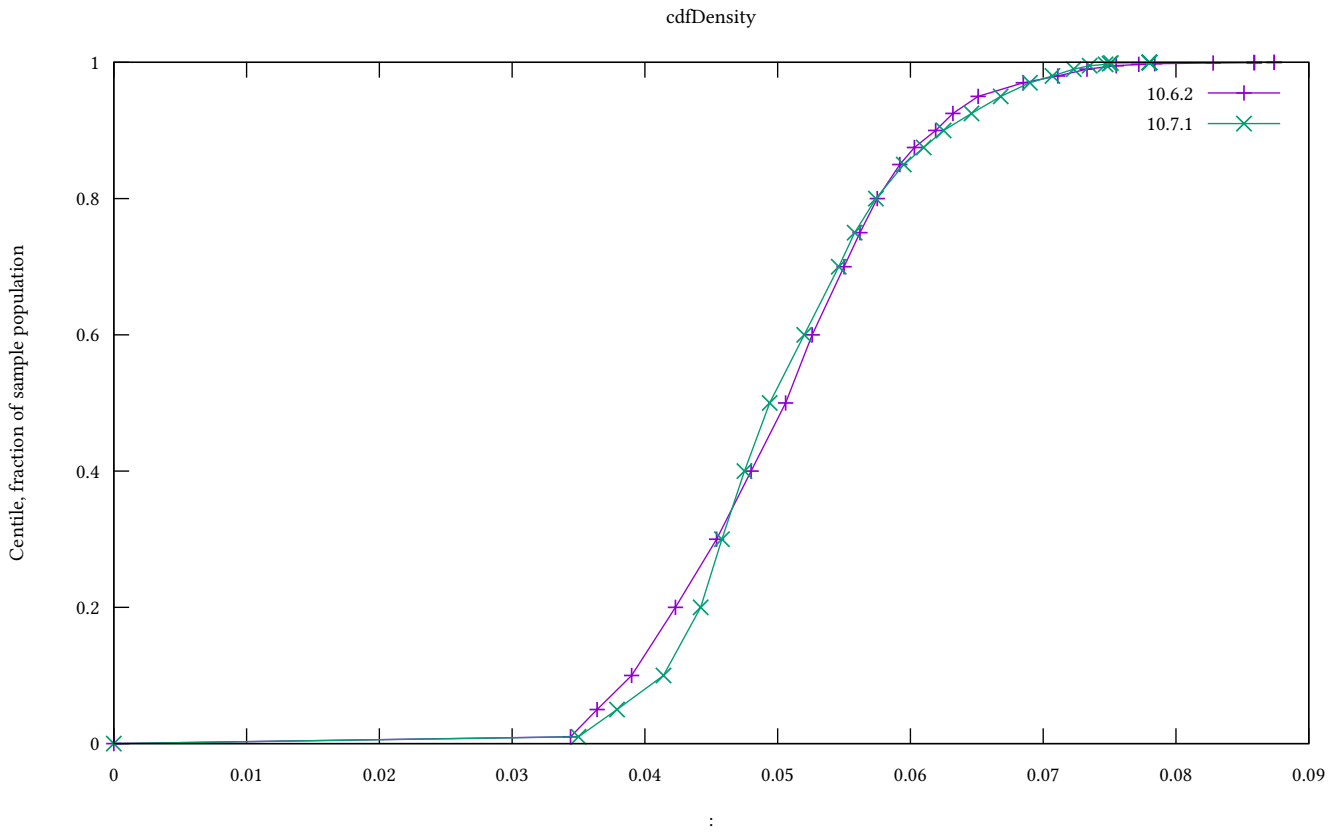


Figure 13: Chain density

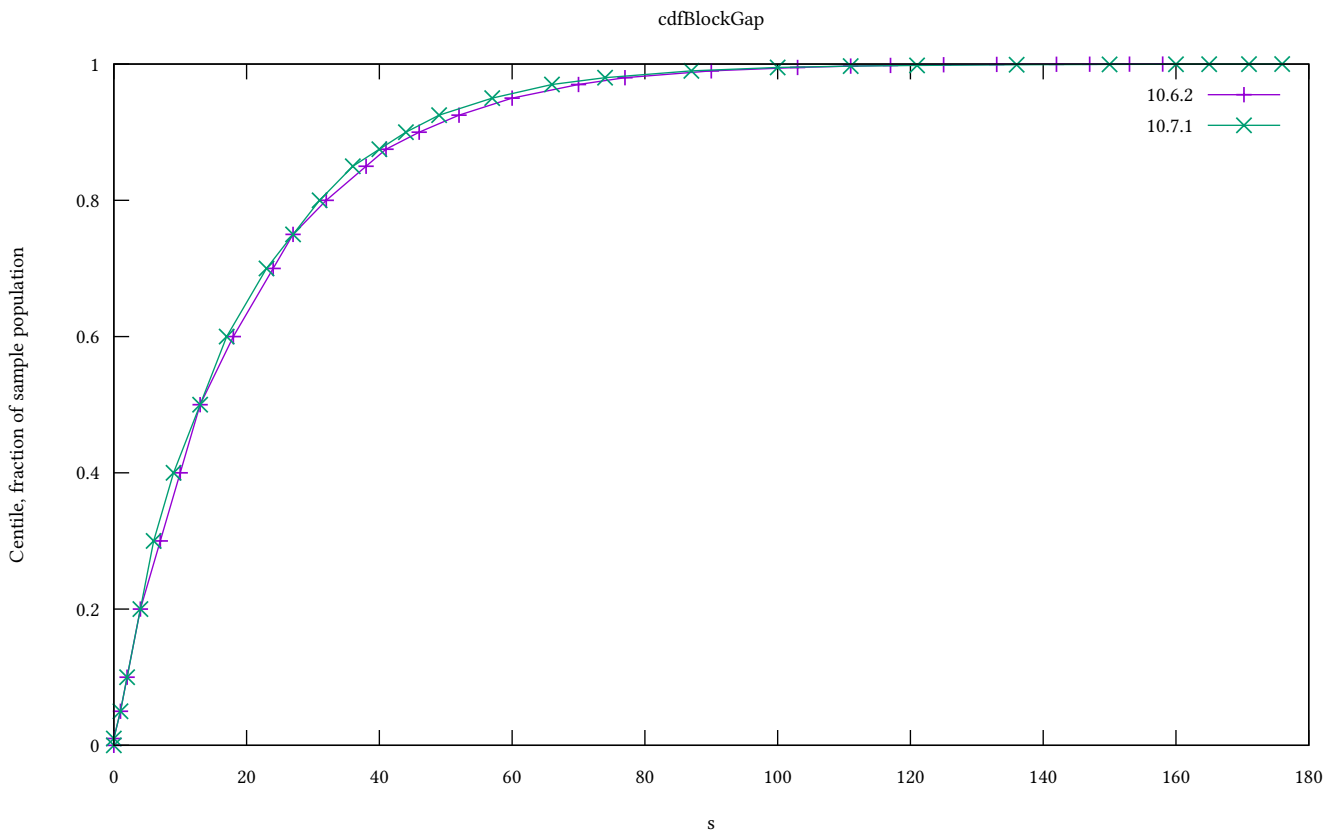


Figure 14: Interblock gap

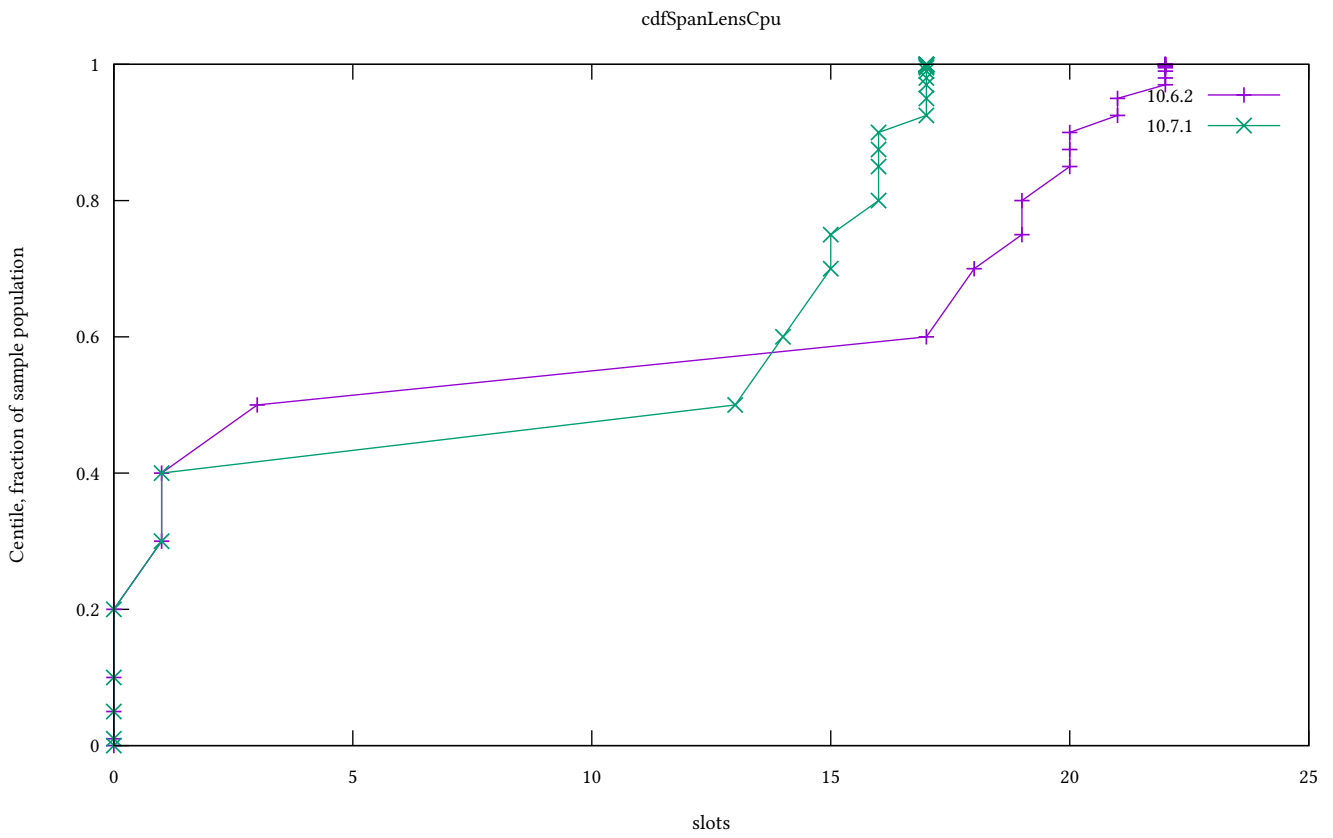


Figure 15: CPU 85% spans

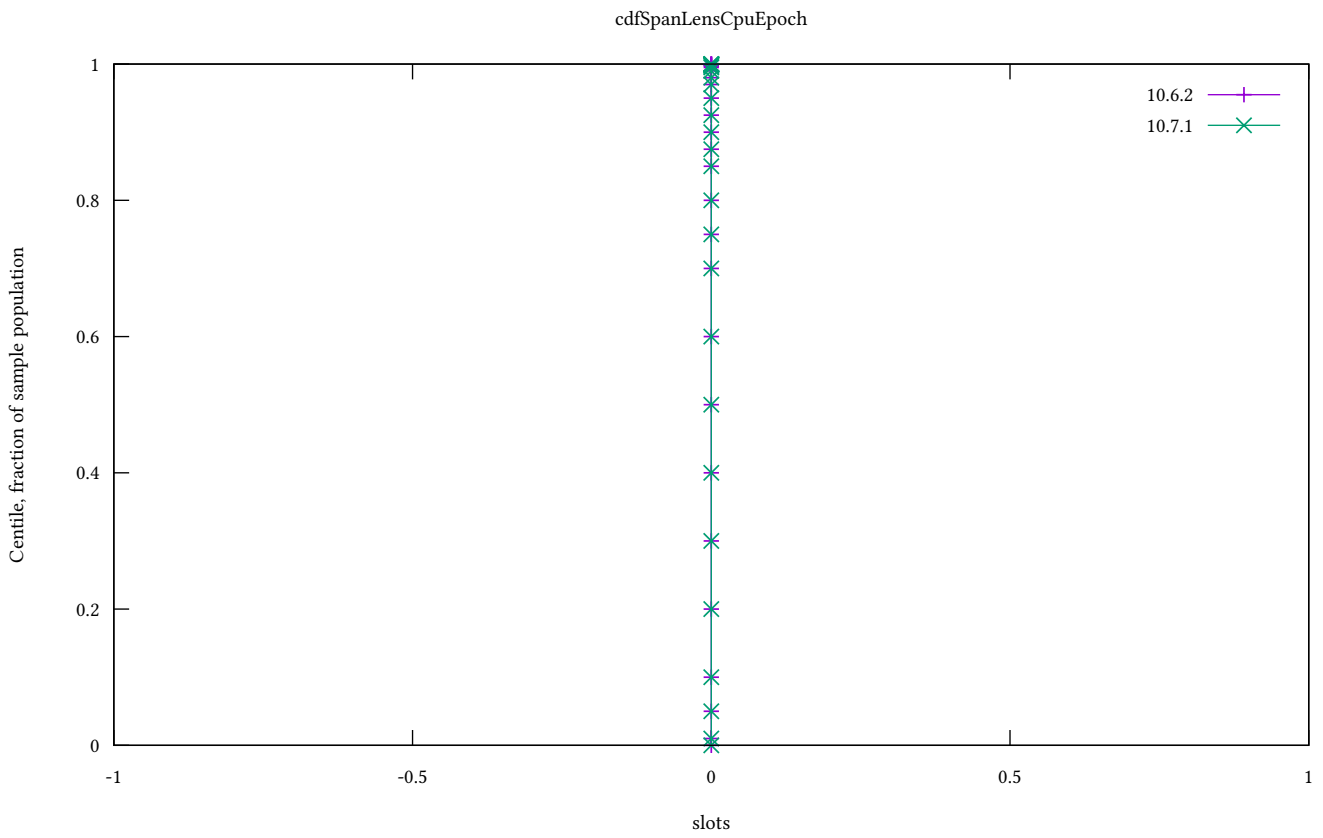


Figure 16: CPU spans at Ep boundary

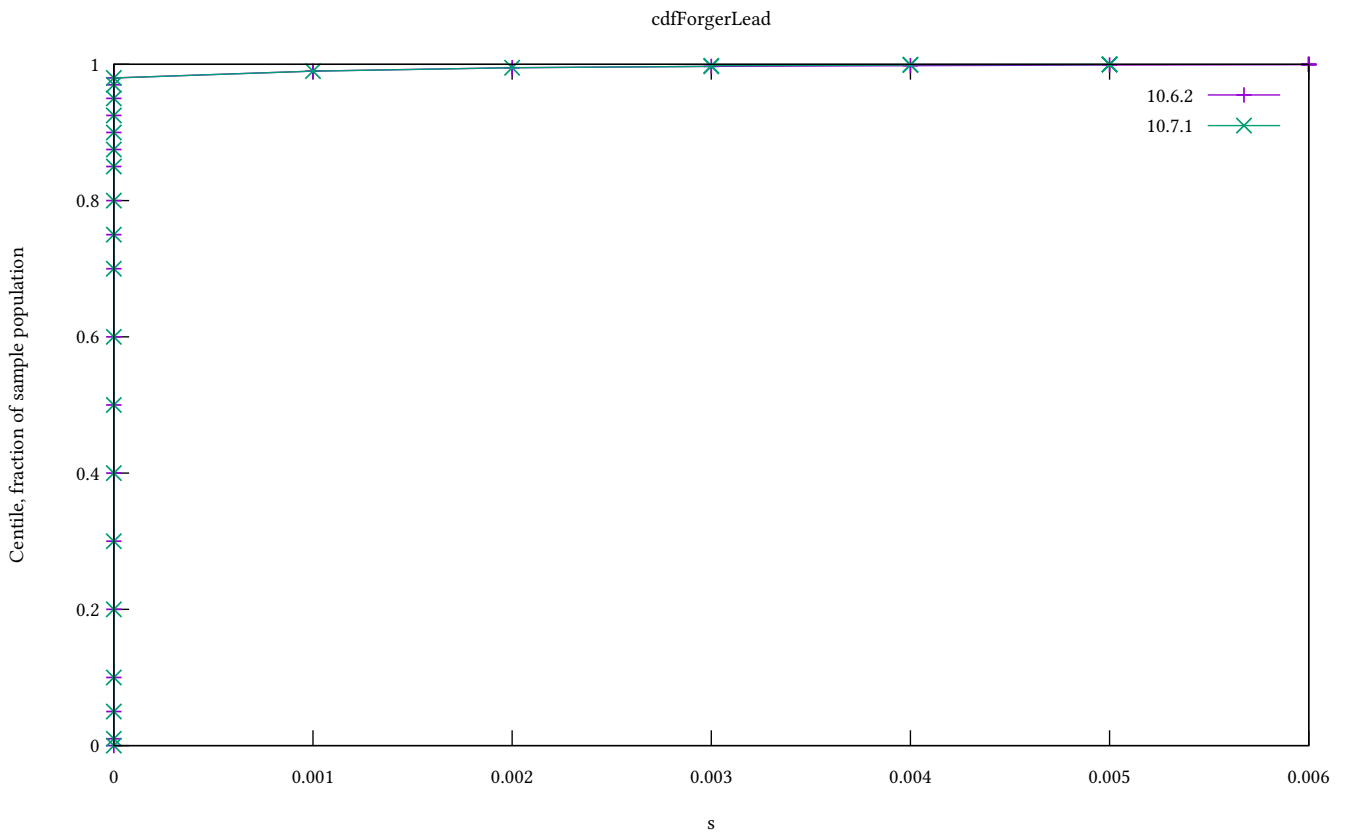


Figure 17: Leadership check duration

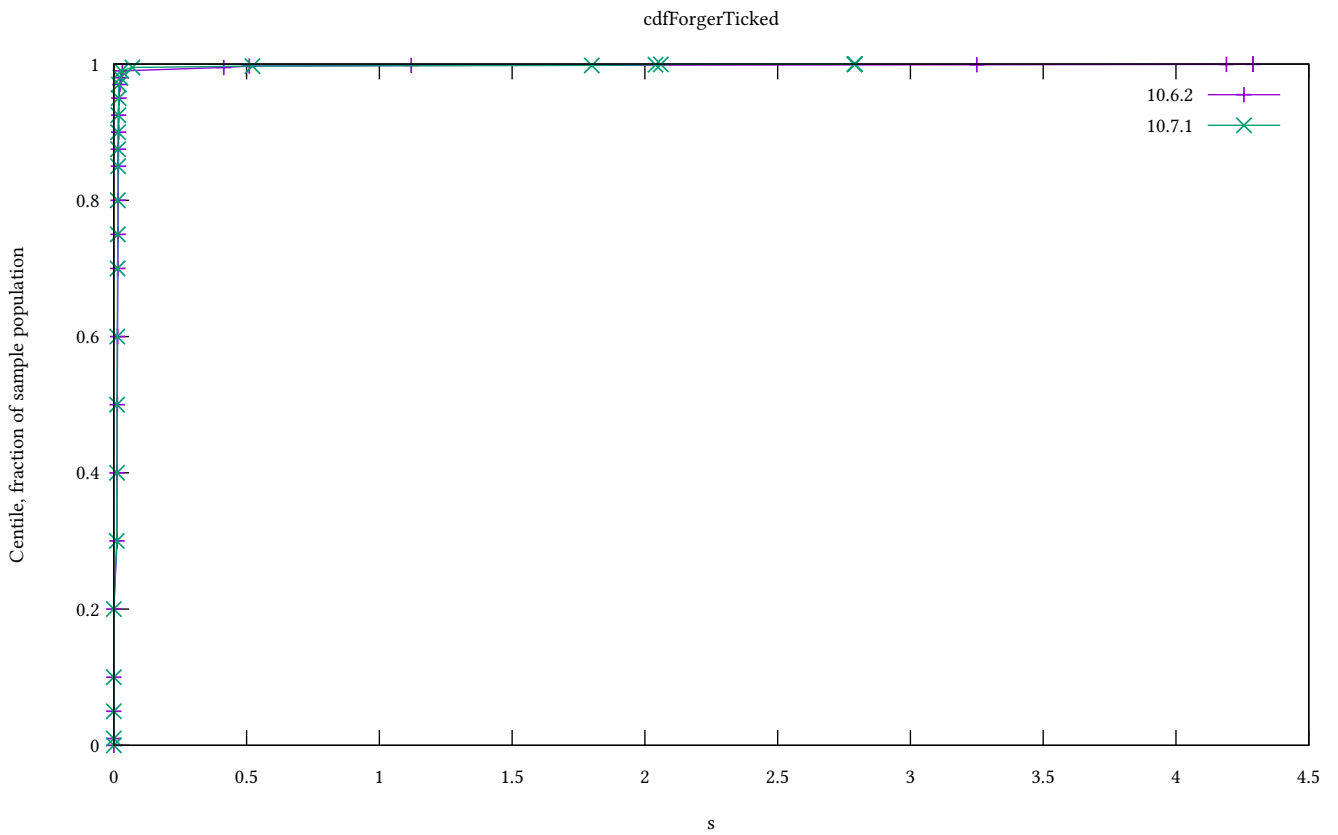


Figure 18: Ledger ticking

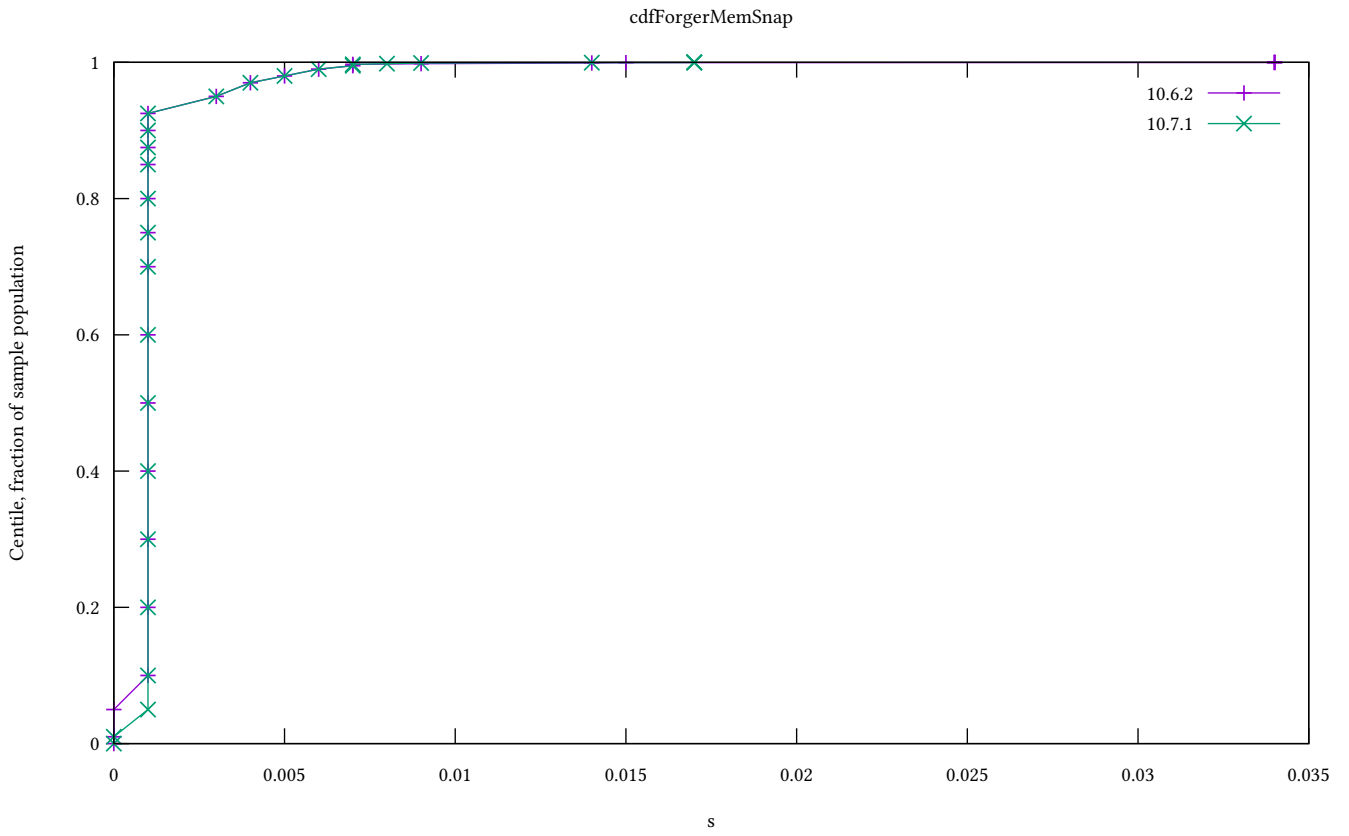


Figure 19: Mempool snapshotting

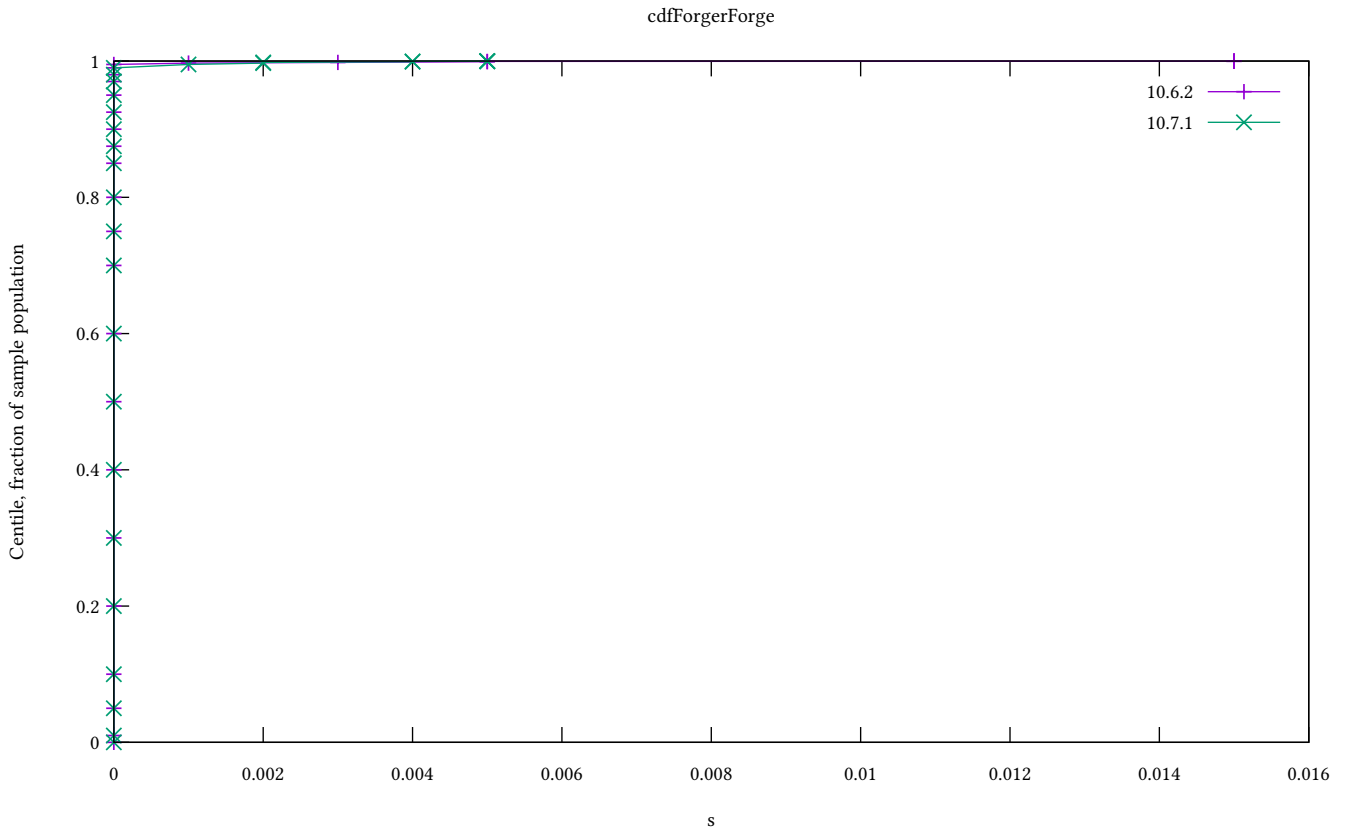


Figure 20: Leadership to forged

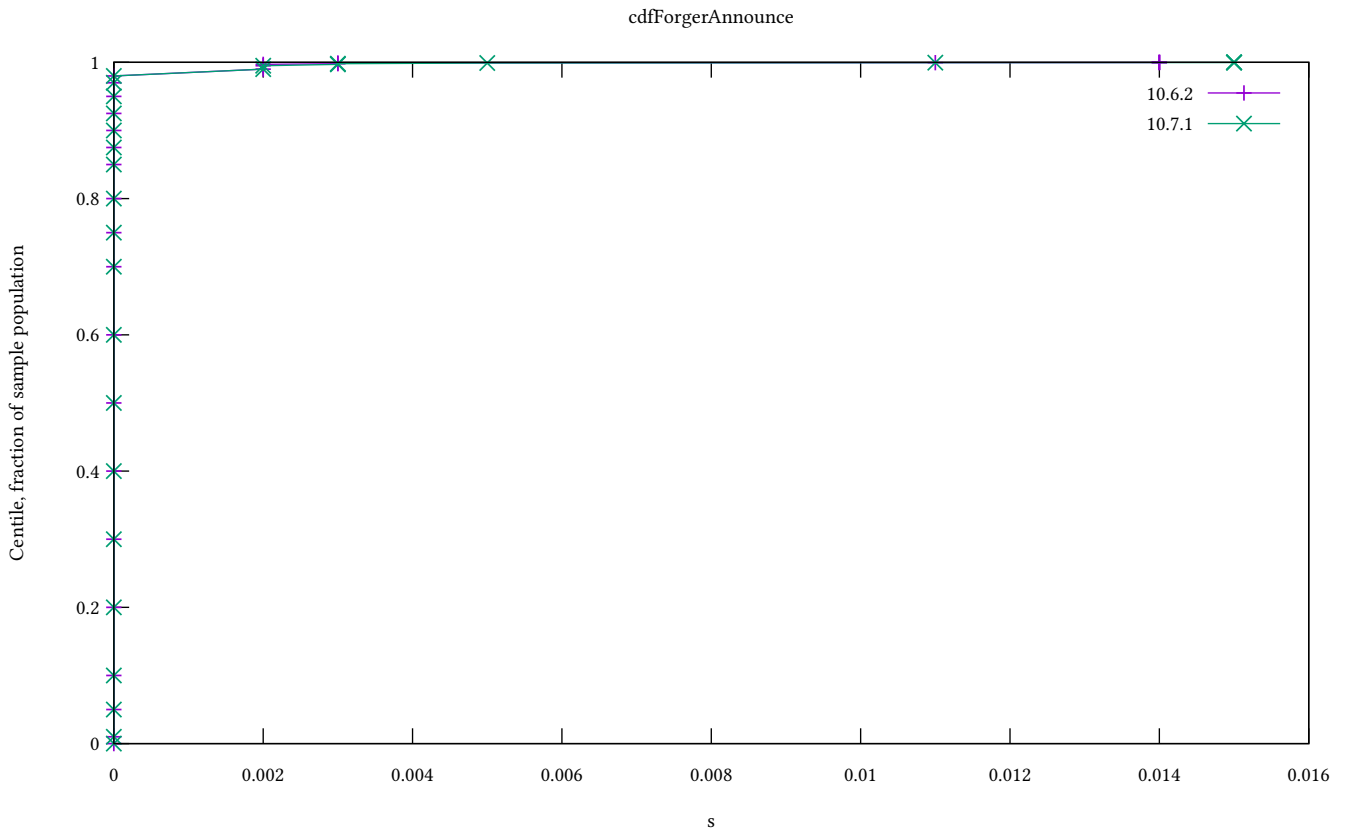


Figure 21: Forged to announced

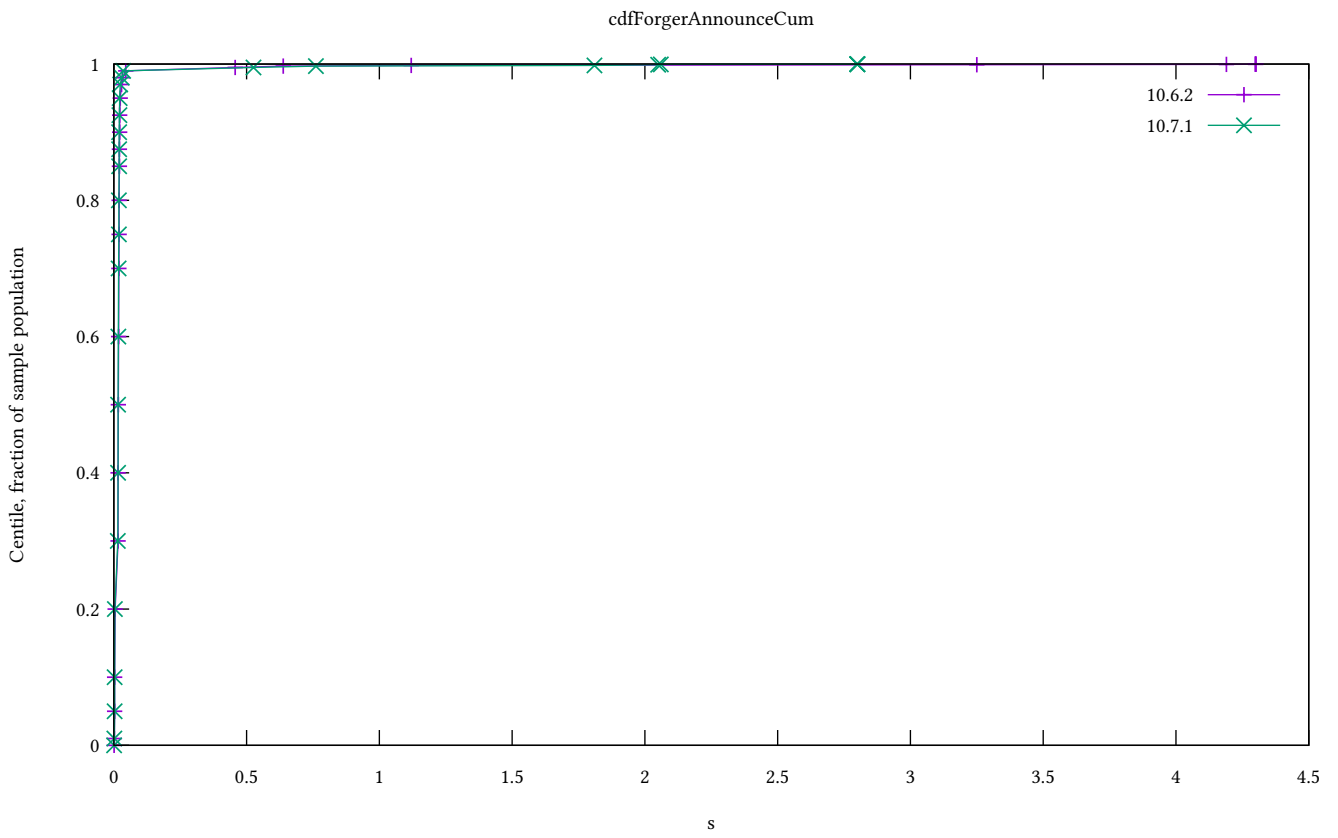


Figure 22: Slot start to announced

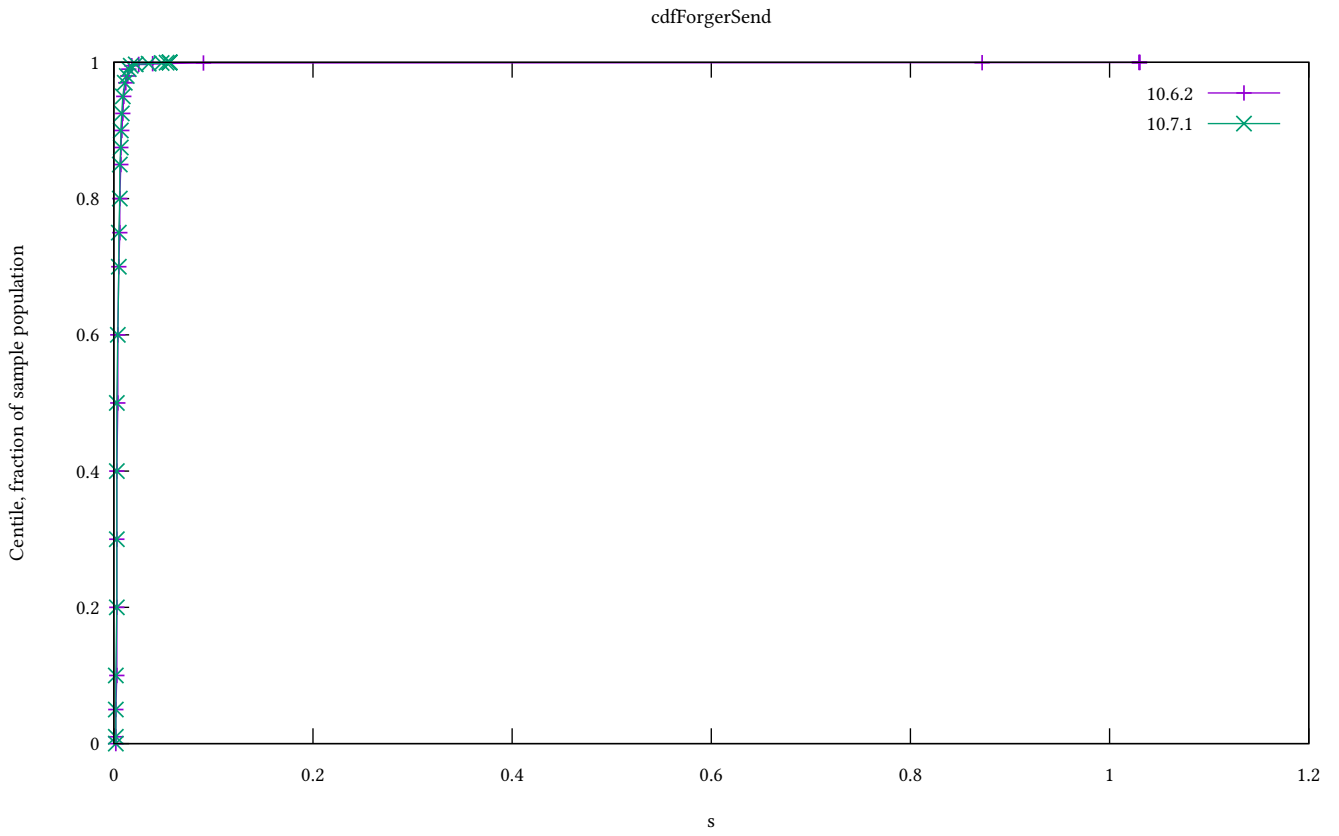


Figure 23: Forged to sending

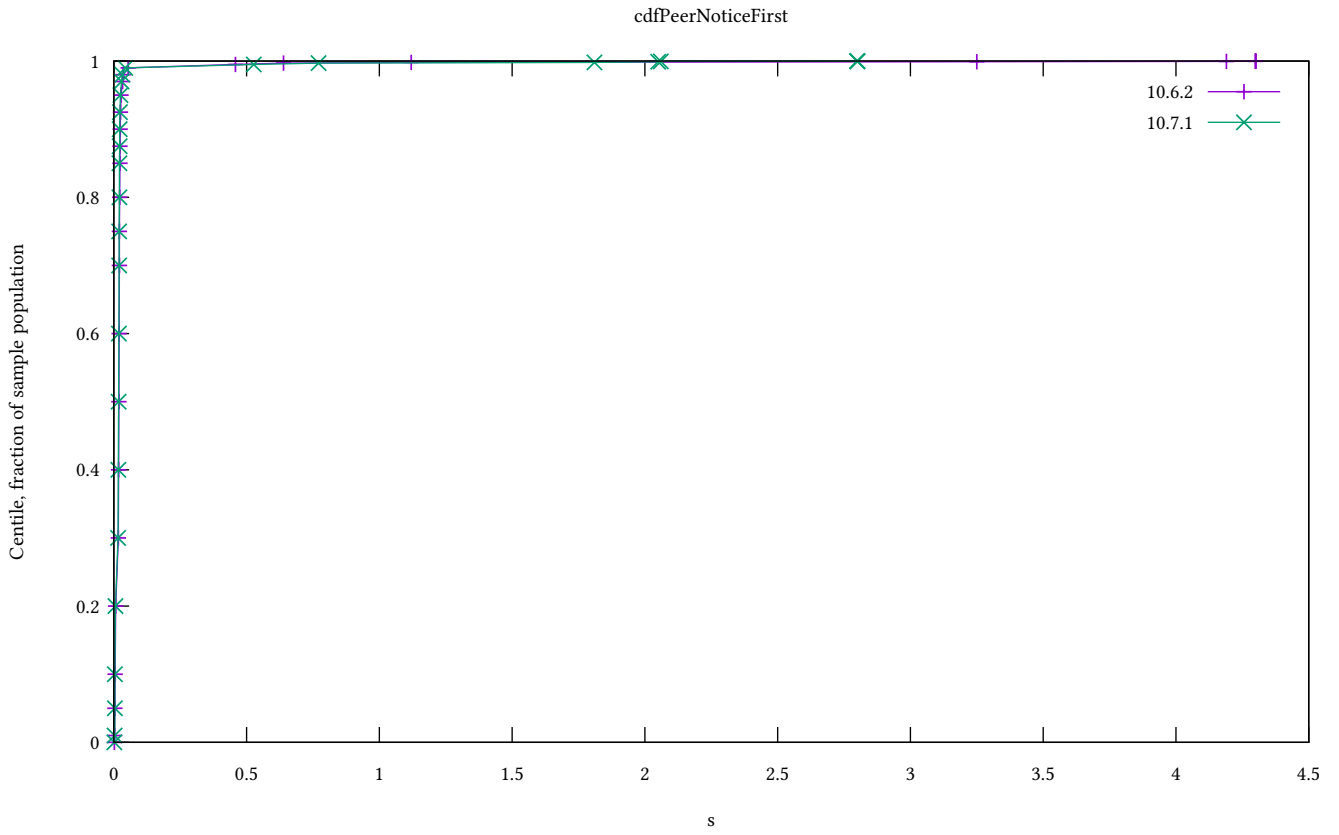


Figure 24: First peer notice

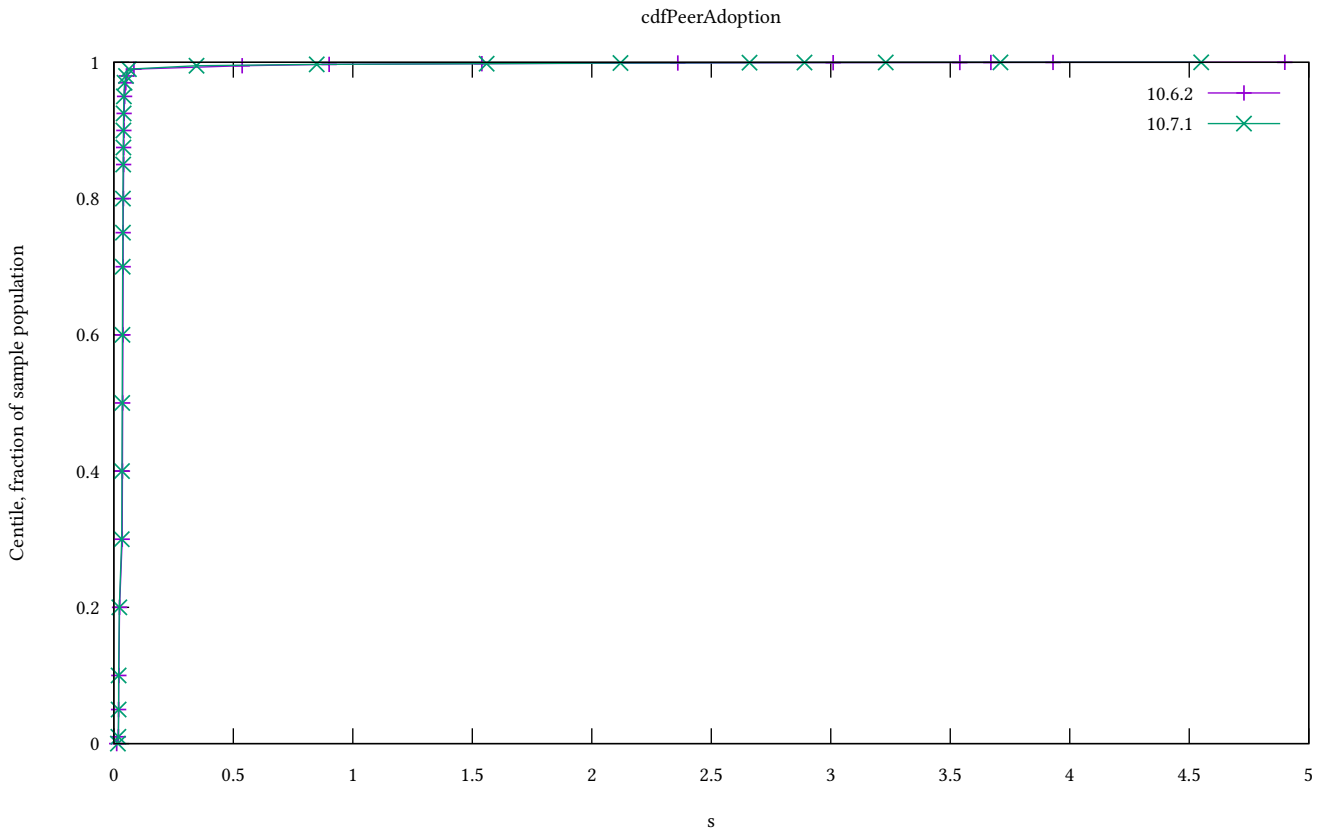


Figure 25: Fetched to adopted

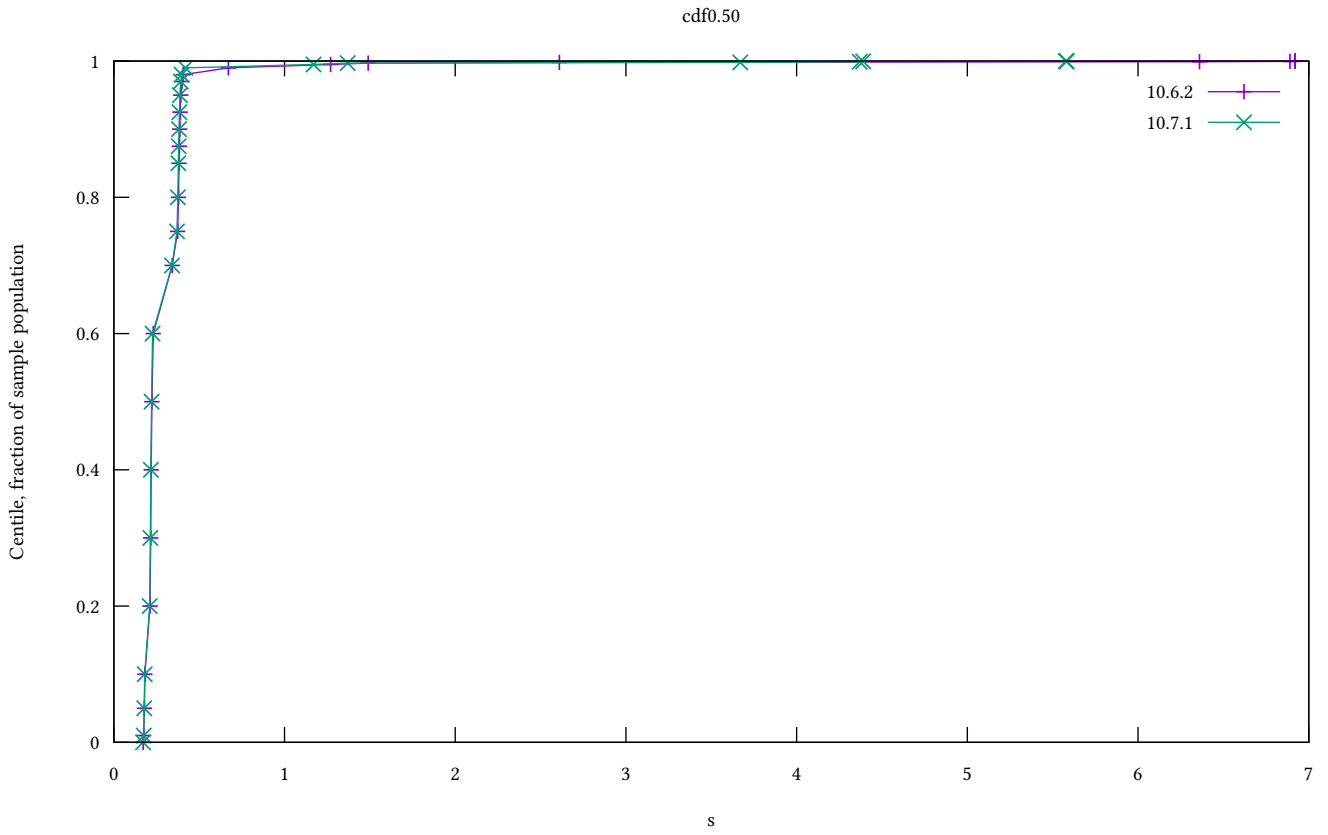


Figure 26: 0.50 adoption

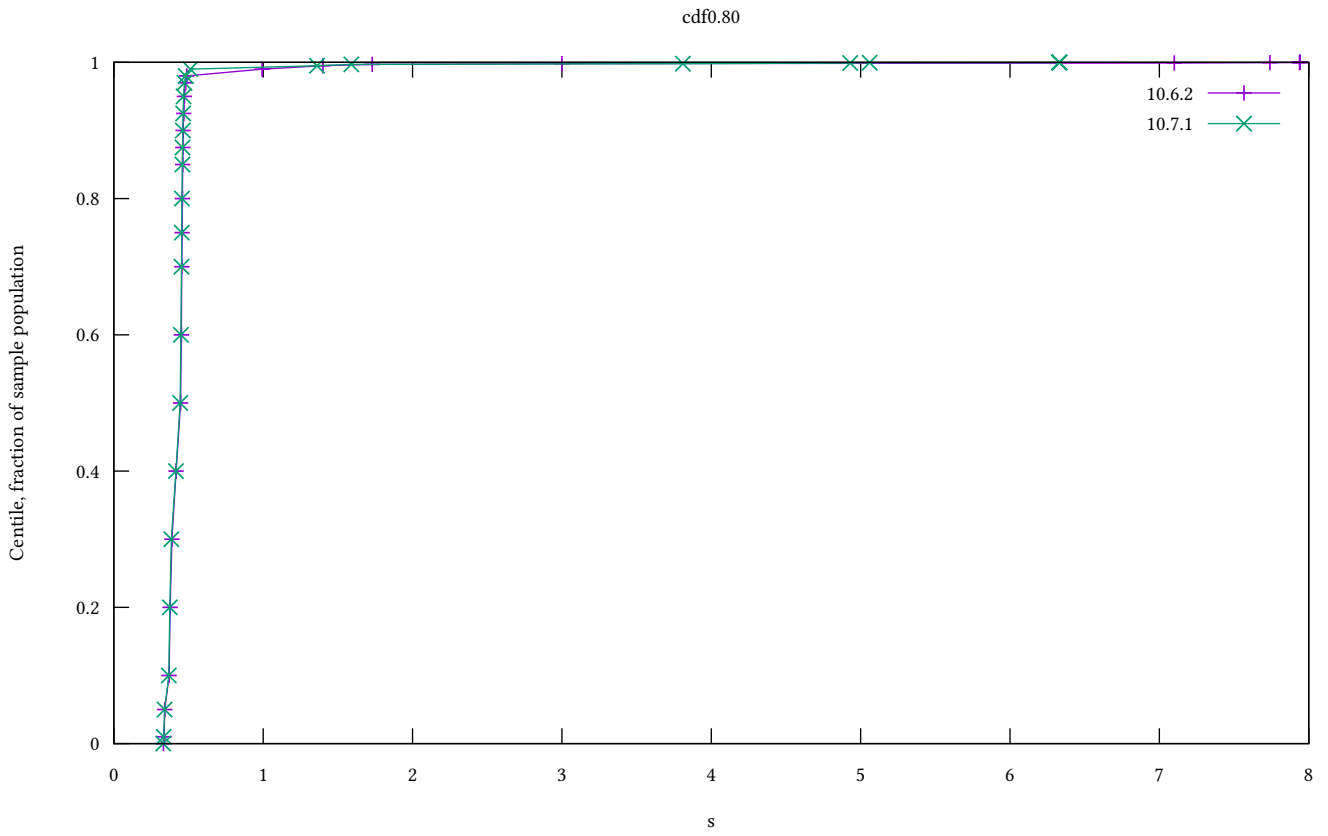


Figure 27: 0.80 adoption

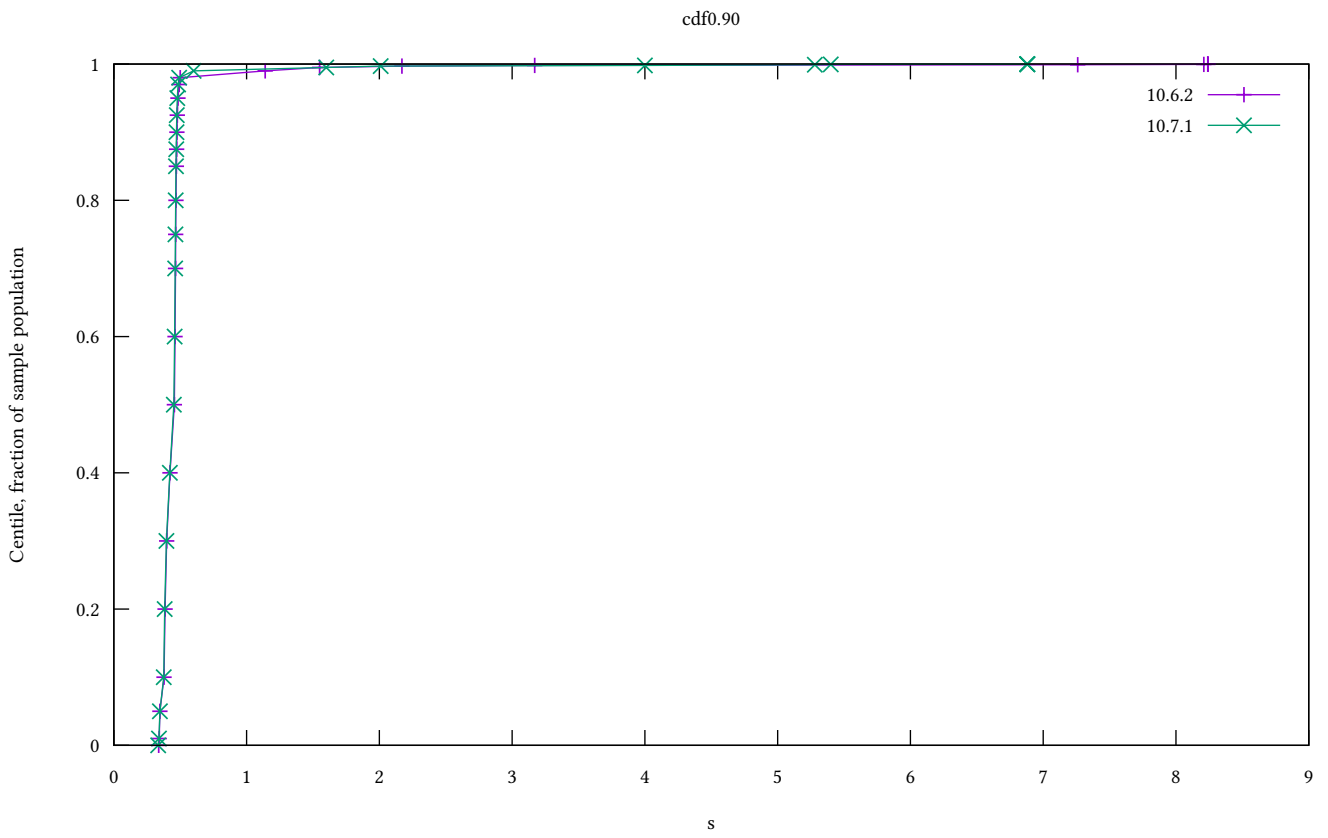


Figure 28: 0.90 adoption

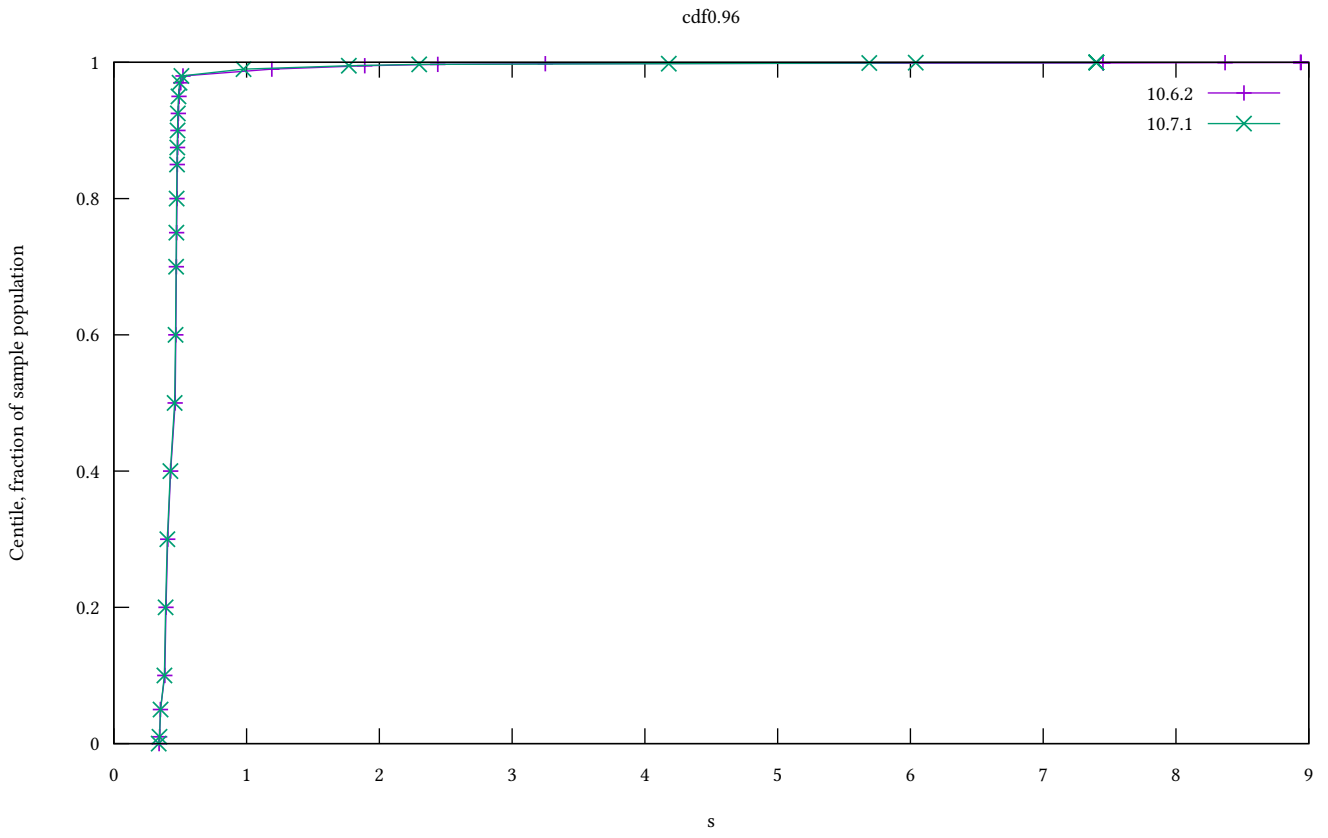


Figure 29: 0.96 adoption

## 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