

Node 10.2: UTxO-HD against regular node  
value-only workload

Michael Karg, Cardano Performance team

2025-02-21

# Contents

<b>1</b>	<b>Manifest</b>	<b>2</b>
<b>2</b>	<b>Analysis</b>	<b>4</b>
2.1	Resource Usage . . . . .	4
2.2	Anomaly control . . . . .	4
2.3	Forging . . . . .	5
2.4	Individual peer propagation . . . . .	5
2.5	End-to-end propagation . . . . .	5
<b>I</b>	<b>Appendix A: charts</b>	<b>6</b>
<b>3</b>	<b>Cluster performance charts</b>	<b>7</b>
<b>II</b>	<b>Appendix B: data dictionary</b>	<b>25</b>
<b>4</b>	<b>Block propagation metrics</b>	<b>26</b>
<b>5</b>	<b>Cluster performance metrics</b>	<b>28</b>

# Chapter 1

## Manifest

We compare `10.2-utxohd` (Conway) relative to `10.2-regular` (Conway), under value-only workload.

	10.2-regular	10.2-utxohd
Analysis date	2025-01-30	2025-02-11
Cluster system start date	2025-01-29	2025-02-10
Cluster system start time	09:04:31	20:33:49
Identifier	10.2	10.2
Run batch	10.2.0	1020utxohd2
GHC version	8.10.7	8.10.7
cardano-node version	10.2	10.2
ouroboros-consensus version	0.22.0.0	0.22.0.0
ouroboros-network version	0.19.0.1	0.19.0.1
cardano-ledger-core version	1.16.0.0	1.16.0.0
plutus-core version	1.37.0.0	1.37.0.0
cardano-crypto version	1.1.2	1.1.2
cardano-prelude version	0.2.1.0	0.2.1.0
cardano-node git	c4d675b	ede3302
ouroboros-consensus git	e924f61	minimal
ouroboros-network git	e91d5c4	e91d5c4
cardano-ledger-core git	b7fe1c3	minimal
plutus-core git	0effd6c	0effd6c
cardano-crypto git	6568a5e	6568a5e
cardano-prelude git	68e015f	68e015f
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	6246465.8076	6516435.8269
Log objects analysed per host	2660800.5961	2249918.2307
Host run time, s	63834.1	63858.8
Host log line rate, Hz	97.855	102.04
Total log objects analysed	138361631	116995748
Run time, s	63840	63864
Analysed run duration, s	48020	48025
Run time efficiency	0.75	0.75
Node start spread, s	8.8315893	6.2768200
Node stop spread, s	4.4862600	4.2920374
Slots analysed	48016	48021
Blocks analysed	2299	2276
Blocks rejected	996	936

# Chapter 2

## Analysis

### 2.1 Resource Usage

	10.2-regular	10.2-utxohd	$\Delta$	$\Delta\%$
Forge loop starts, #	0.99837	0.99859	0.000	0
Process CPU usage, %	9.7846	8.9401	-0.845	-9
RTS GC CPU usage, %	1.2495	1.0299	-0.220	-18
RTS Mutator CPU usage, %	8.5324	7.9066	-0.626	-7
Major GCs, #	0.00105	0.00086	-0.000	0
Minor GCs, #	2.2408	2.2523	0.012	1
Kernel RSS, MB	9273.7	9595.1	321.400	3
RTS heap size, MB	9220.1	9539.0	318.900	3
RTS live GC dataset, MB	4018.1	4126.5	108.400	3
RTS alloc rate, MB/s	69.846	68.856	-0.990	-1
Filesystem reads, KB/s	0.04703	0.00107	-0.046	-98
Filesystem writes, KB/s	252.19	256.65	4.460	2
CPU 85% spans, slots	4.5605	3.4732	-1.087	-24
Sample count	(249>)	(249>)		

### 2.2 Anomaly control

	10.2-regular	10.2-utxohd	$\Delta$	$\Delta\%$
Blocks per host, blocks	65.211	64.23	-0.981	-2
Filtered to chained block ratio, /	0.69827	0.70843	0.010	1
Chained to forged block ratio, /	0.97174	0.96218	-0.010	-1
Height & slot battles, blocks	0.00478	0.00878	0.004	84
Block size, B	88951	88966	15	0
Sample count	(52)	(52)		

## 2.3 Forging

	10.2-regular	10.2-utxohd	$\Delta$	$\Delta\%$
Started forge loop iteration, s	0.00119	0.00096	-0.000	0
Acquired block context, s	0.02778	0.02458	-0.003	-11
Acquired ledger state, s	7e-05	9e-05	0.000	0
Acquired ledger view, s	2e-05	3e-05	0.000	0
Leadership check duration, s	0.00046	0.00056	0.000	0
Ledger ticking, s	0.02914	0.032	0.003	10
Mempool snapshotting, s	0.07662	0.06043	-0.016	-21
Leadership to forged, s	0.00088	0.00087	-0.000	0
Forged to announced, s	0.0008	0.00078	-0.000	0
Forged to sending, s	0.00656	0.00611	-0.000	0
Forged to self-adopted, s	0.08271	0.07918	-0.004	-5
Slot start to announced, s	0.137	0.12032	-0.017	-12
Sample count	(2299)	(2276)		

## 2.4 Individual peer propagation

	10.2-regular	10.2-utxohd	$\Delta$	$\Delta\%$
First peer notice, s	0.13906	0.12224	-0.017	-12
First peer fetch, s	0.14925	0.13243	-0.017	-11
Notice to fetch request, s	0.00142	0.00138	-0.000	0
Fetch duration, s	0.34239	0.35496	0.013	4
Fetches to announced, s	0.0014	0.00119	-0.000	0
Fetches to sending, s	0.04767	0.04692	-0.001	-2
Fetches to adopted, s	0.08803	0.08557	-0.002	-2
Sample count	(2299)	(2276)		

## 2.5 End-to-end propagation

	10.2-regular	10.2-utxohd	$\Delta$	$\Delta\%$
0.50 adoption, s	0.68612	0.67474	-0.011	-2
0.80 adoption, s	1.0266	1.035	0.008	1
0.90 adoption, s	1.0451	1.052	0.007	1
0.92 adoption, s	1.0494	1.0554	0.006	1
0.94 adoption, s	1.0551	1.0599	0.005	0
0.96 adoption, s	1.0629	1.0661	0.003	0
0.98 adoption, s	1.0729	1.0762	0.003	0
1.00 adoption, s	1.1183	1.1143	-0.004	0
Sample count	(2299)	(2276)		

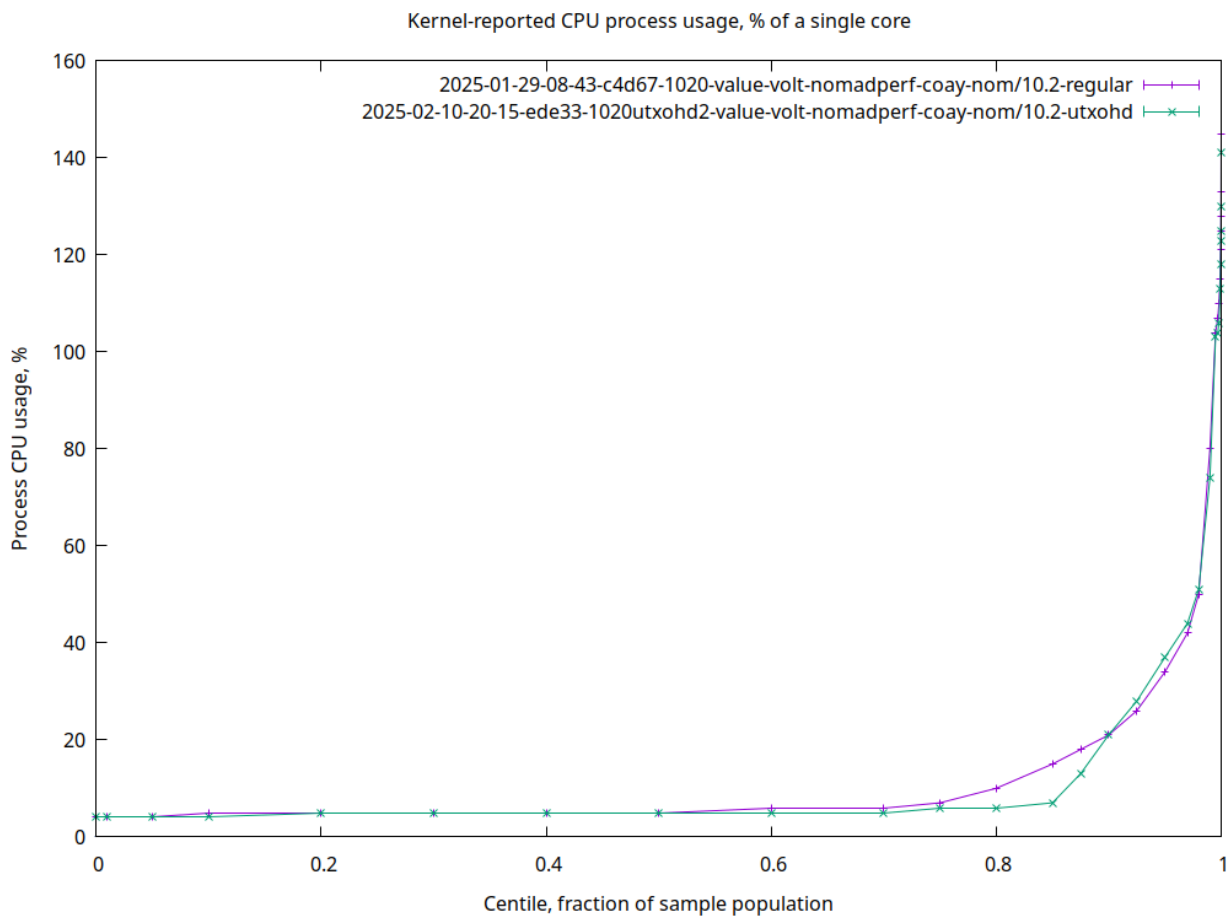
## Part I

# Appendix A: charts

# Chapter 3

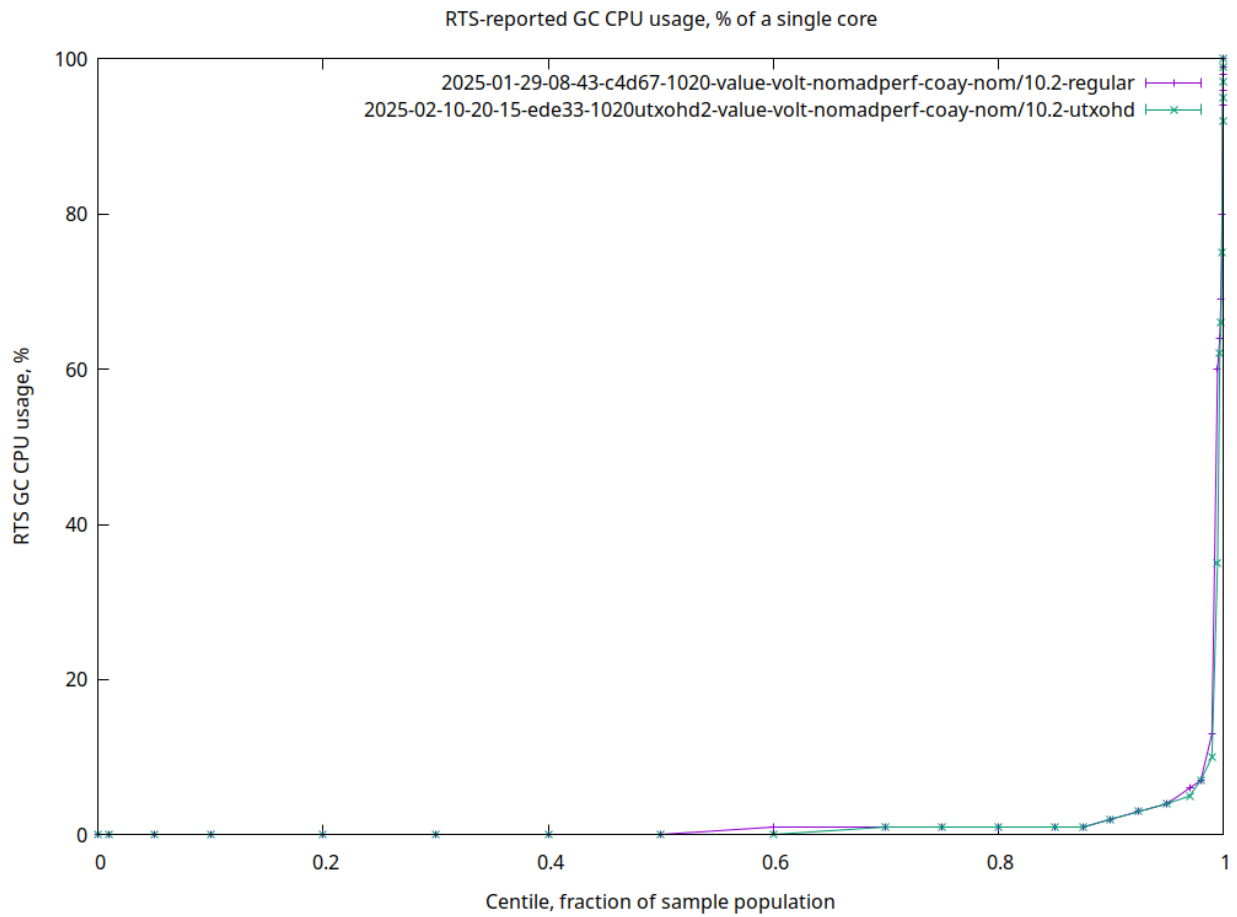
## Cluster performance charts

**Process CPU usage (CentiCpu)** Kernel-reported CPU process usage, % of a single core

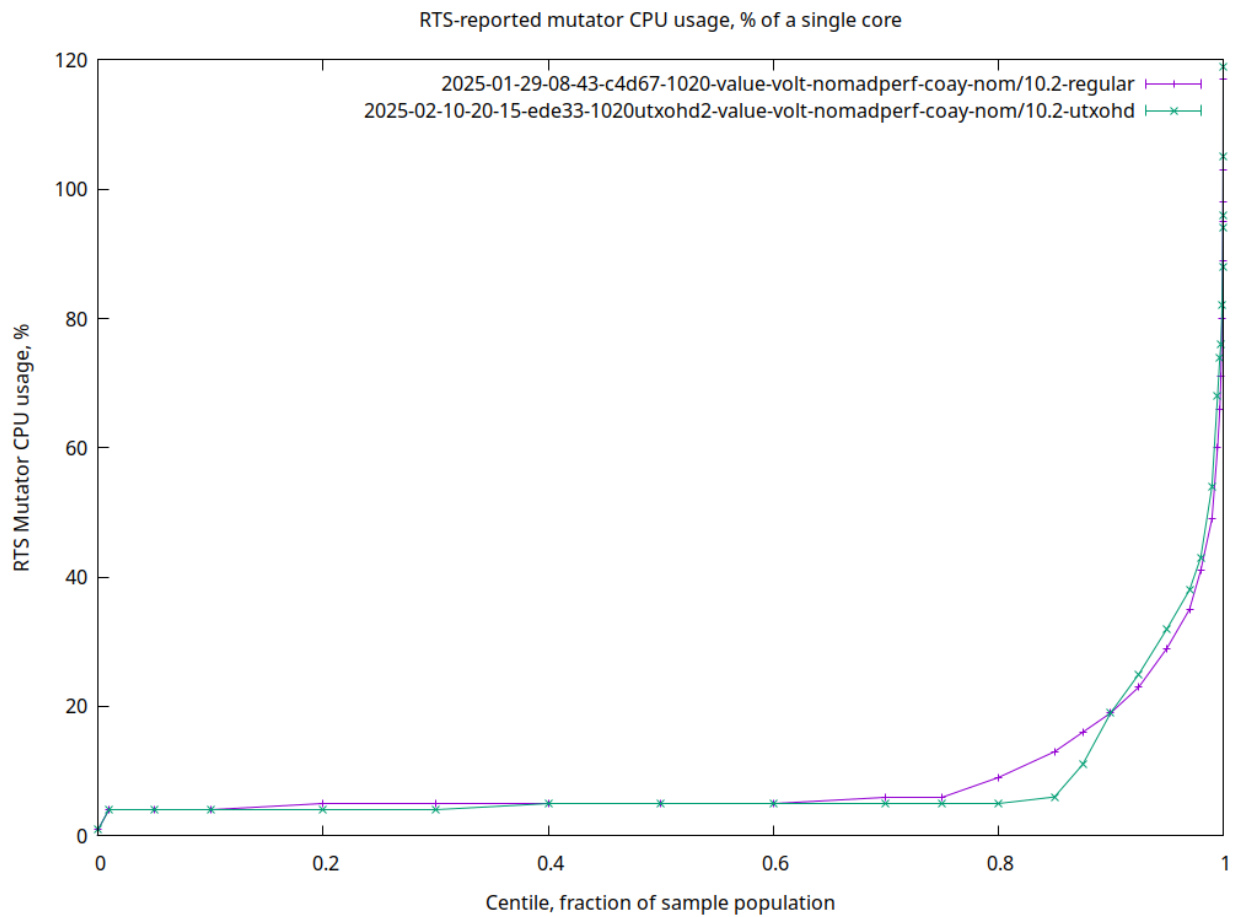


**RTS GC CPU usage (CentiGC)** RTS-reported GC CPU usage, % of a single core

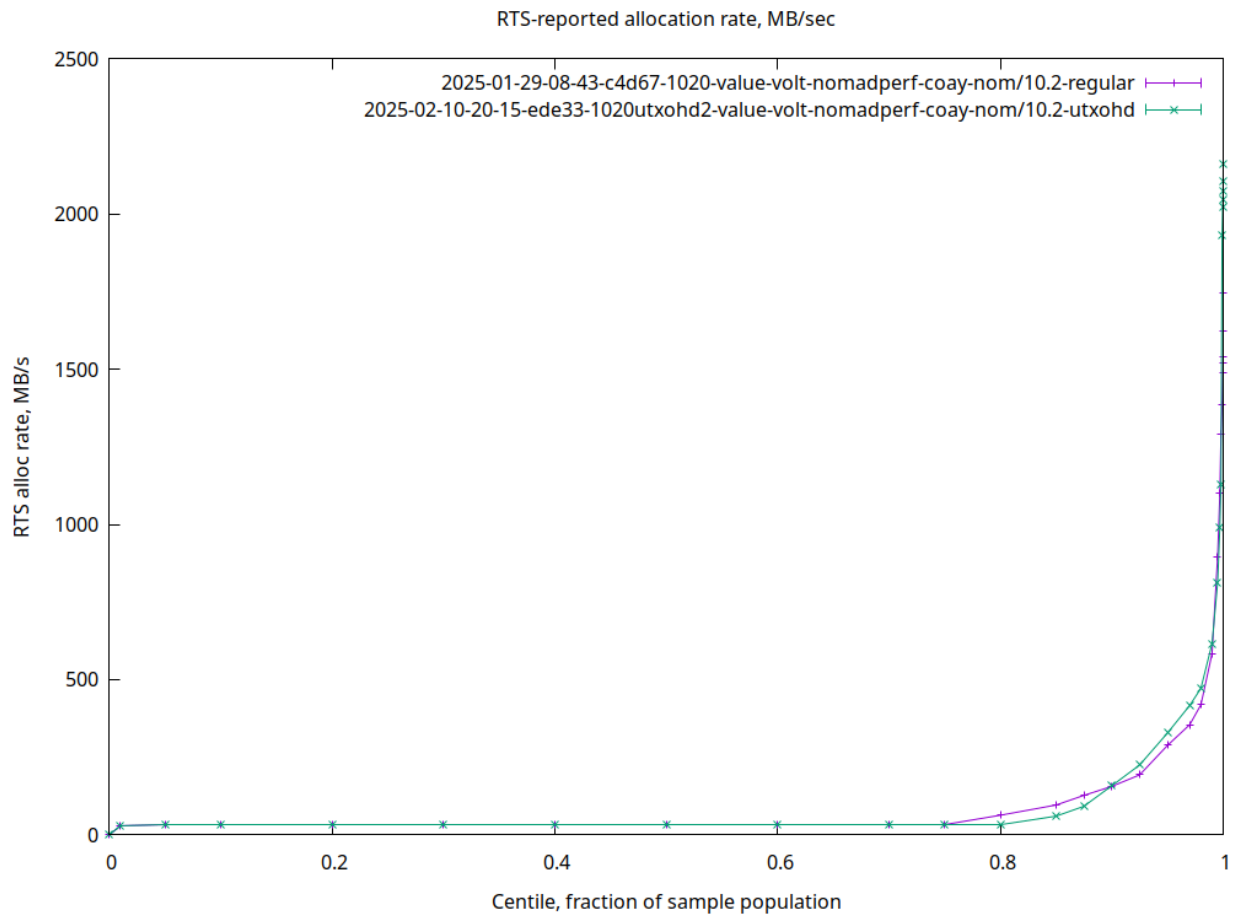




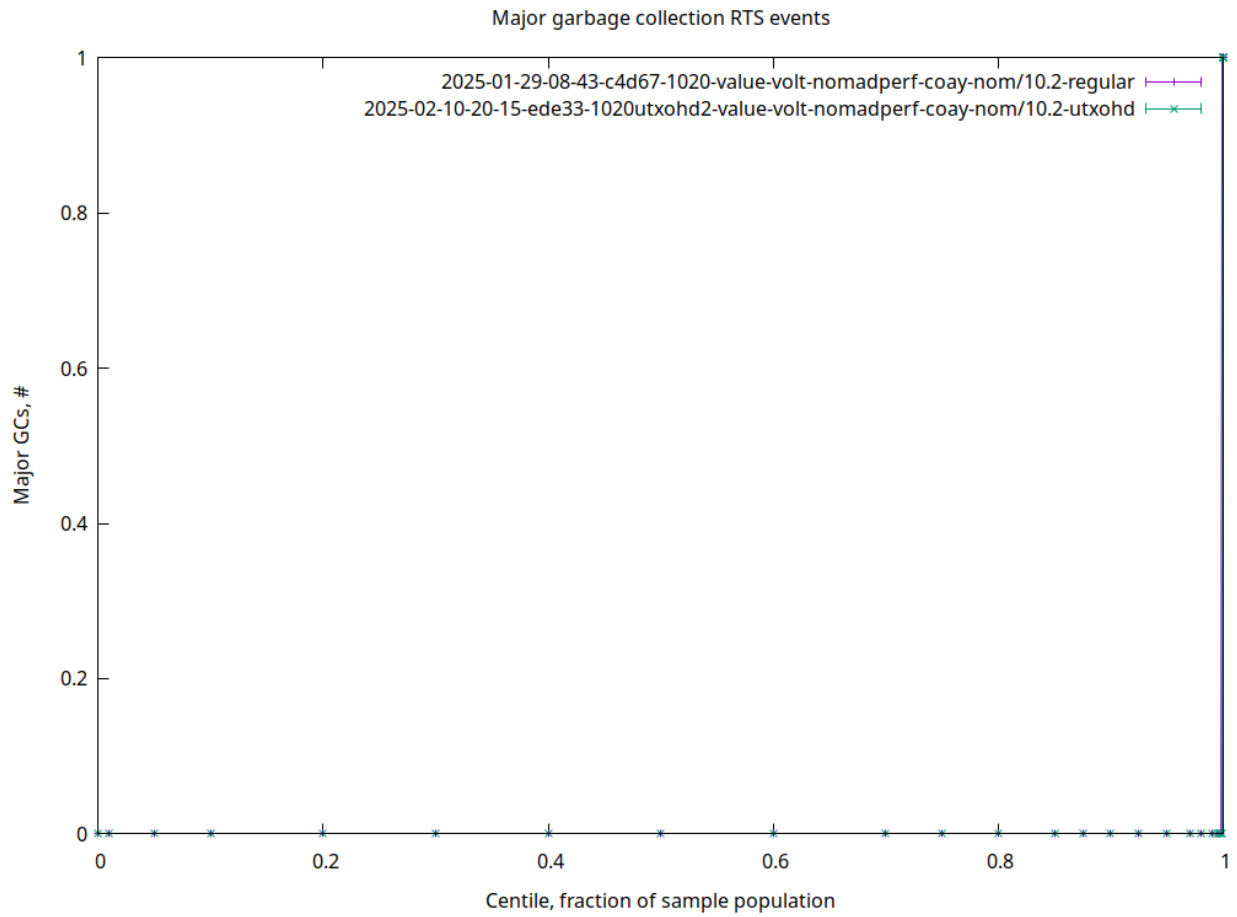
**RTS Mutator CPU usage (CentiMut)** RTS-reported mutator CPU usage, % of a single core



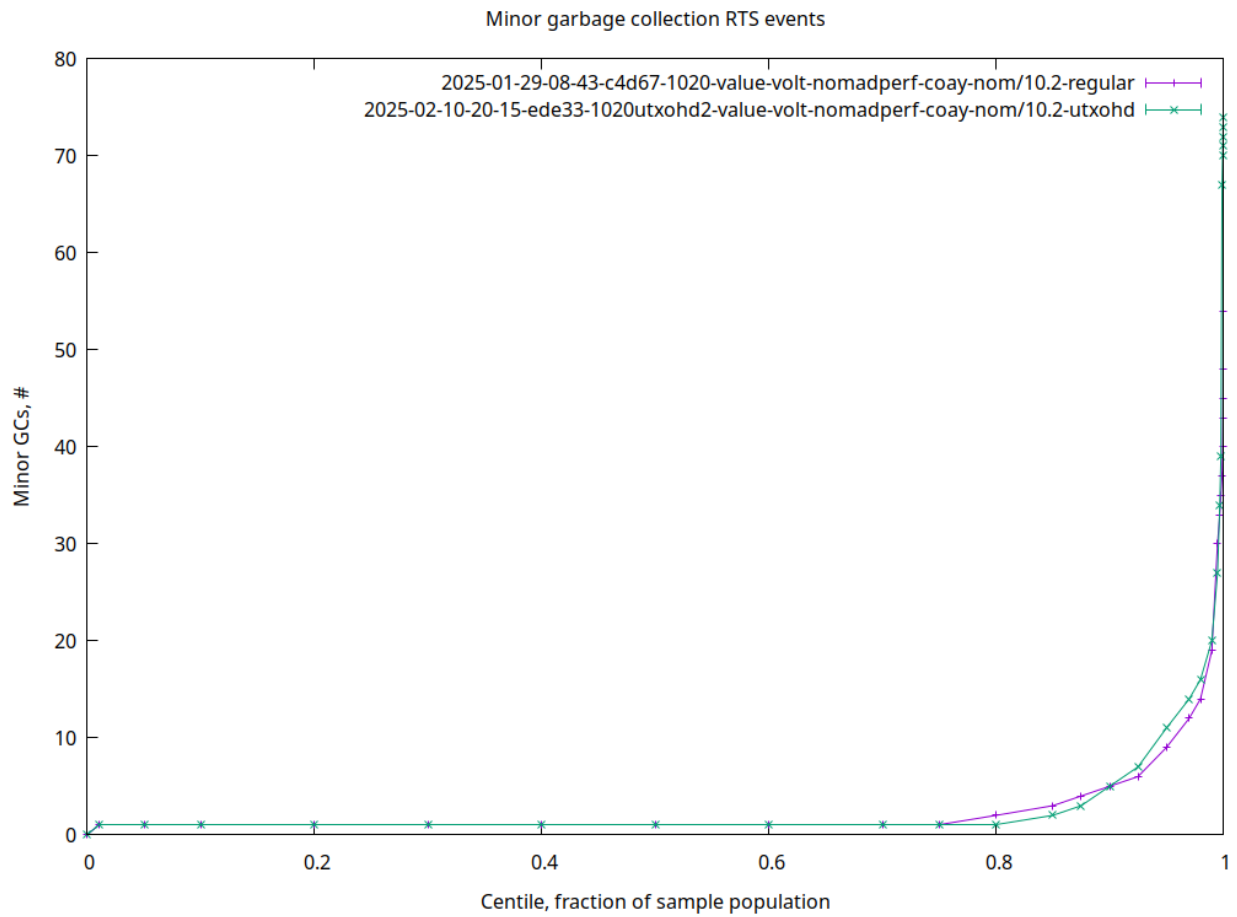
**RTS alloc rate (Alloc)** RTS-reported allocation rate, MB/sec



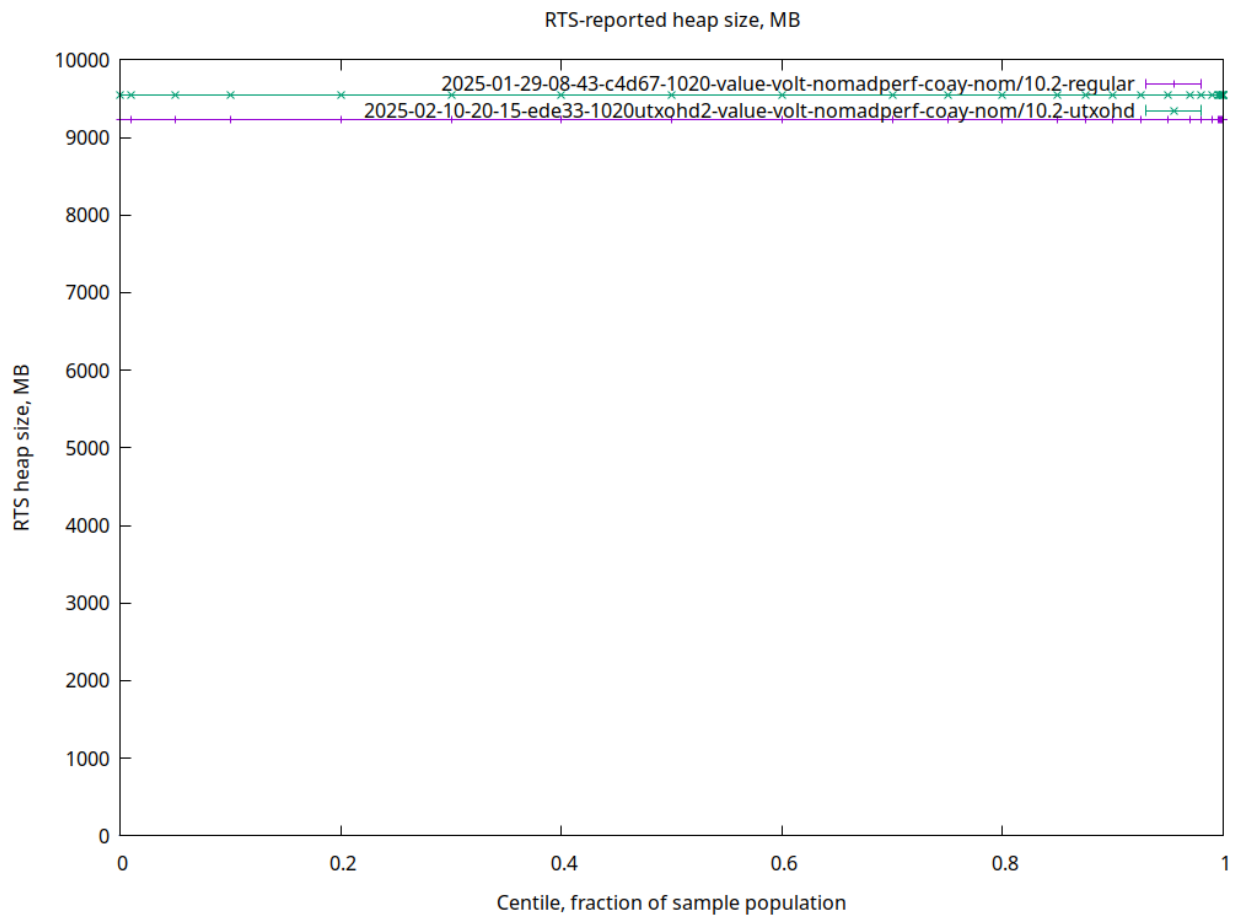
Major GCs (GcsMajor) Major garbage collection RTS events



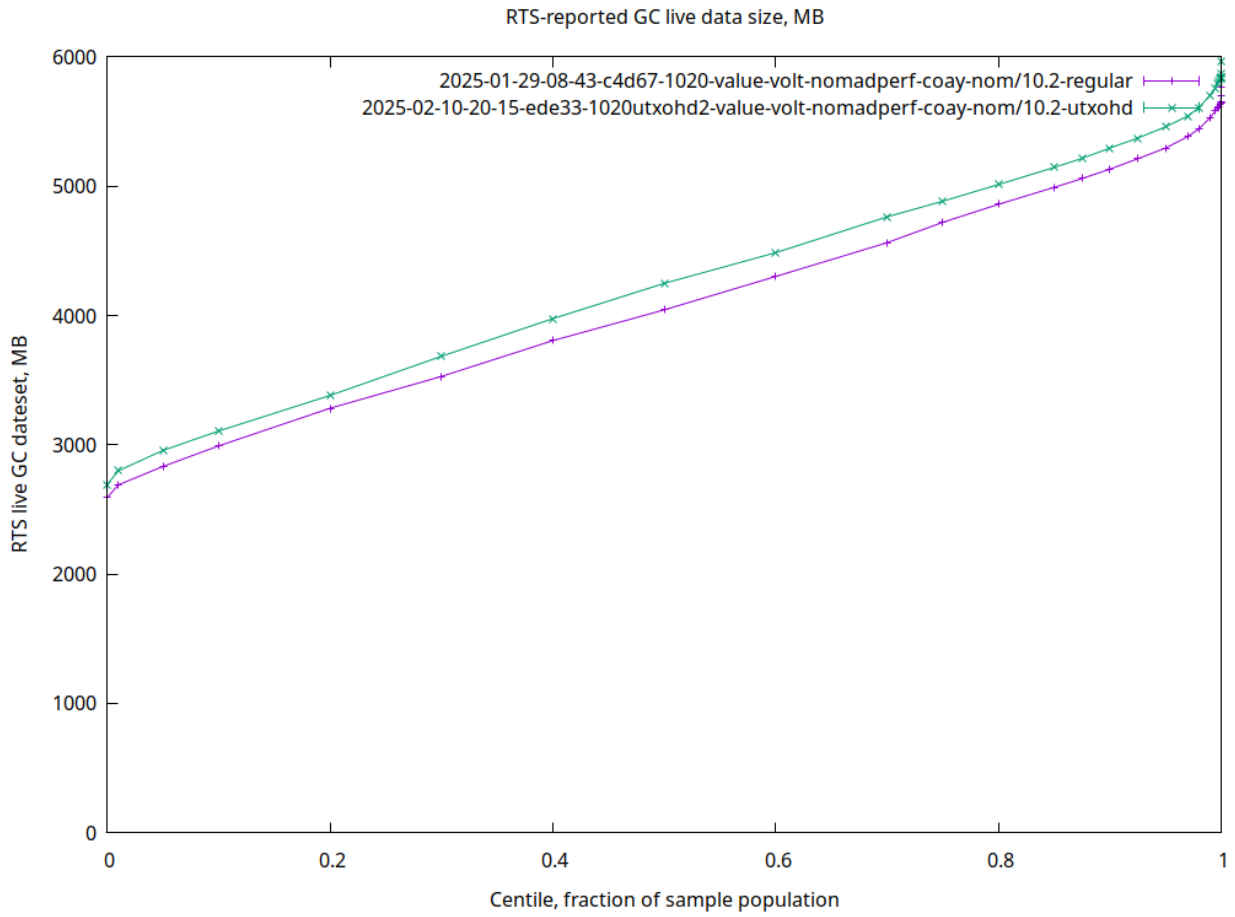
Minor GCs (GcsMinor) Minor garbage collection RTS events



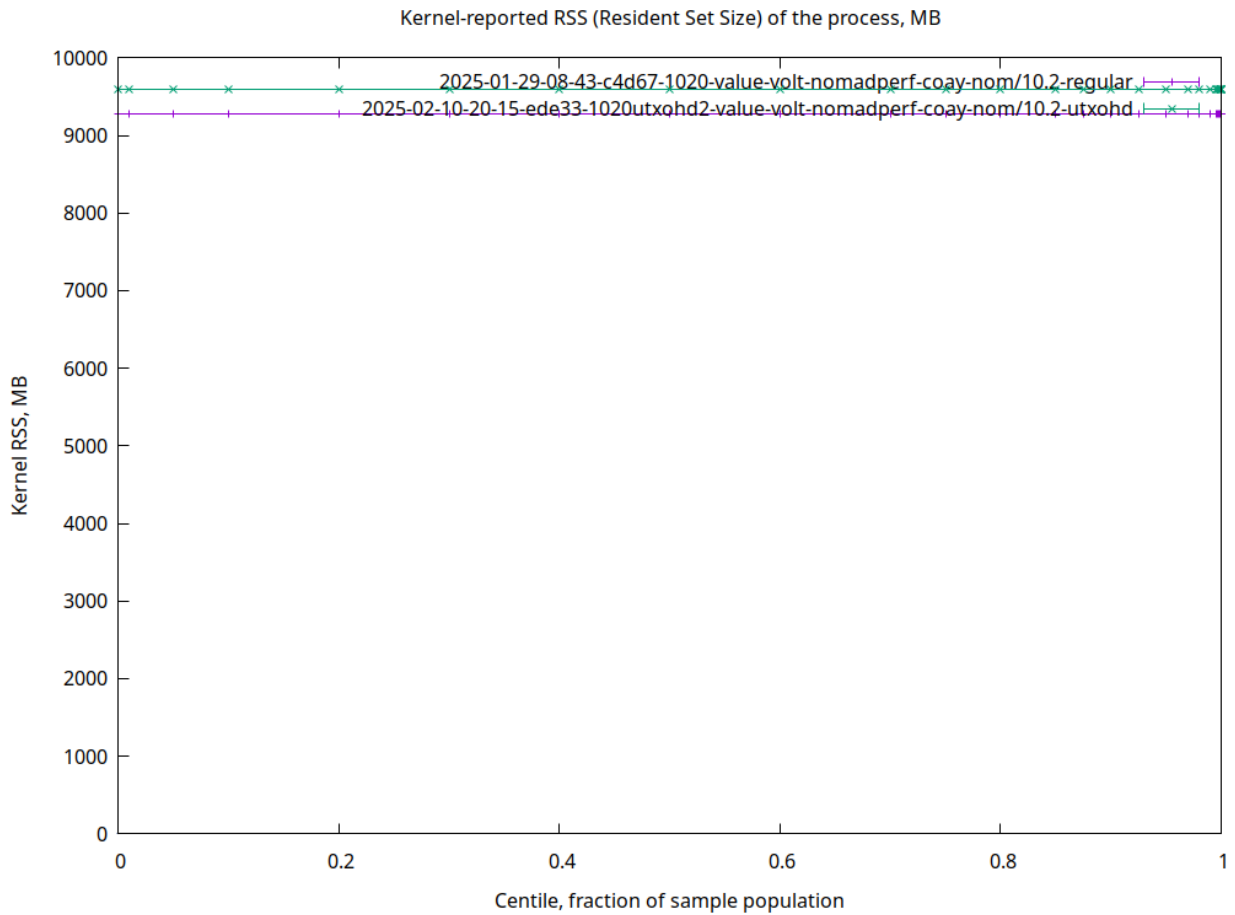
RTS heap size (Heap) RTS-reported heap size, MB



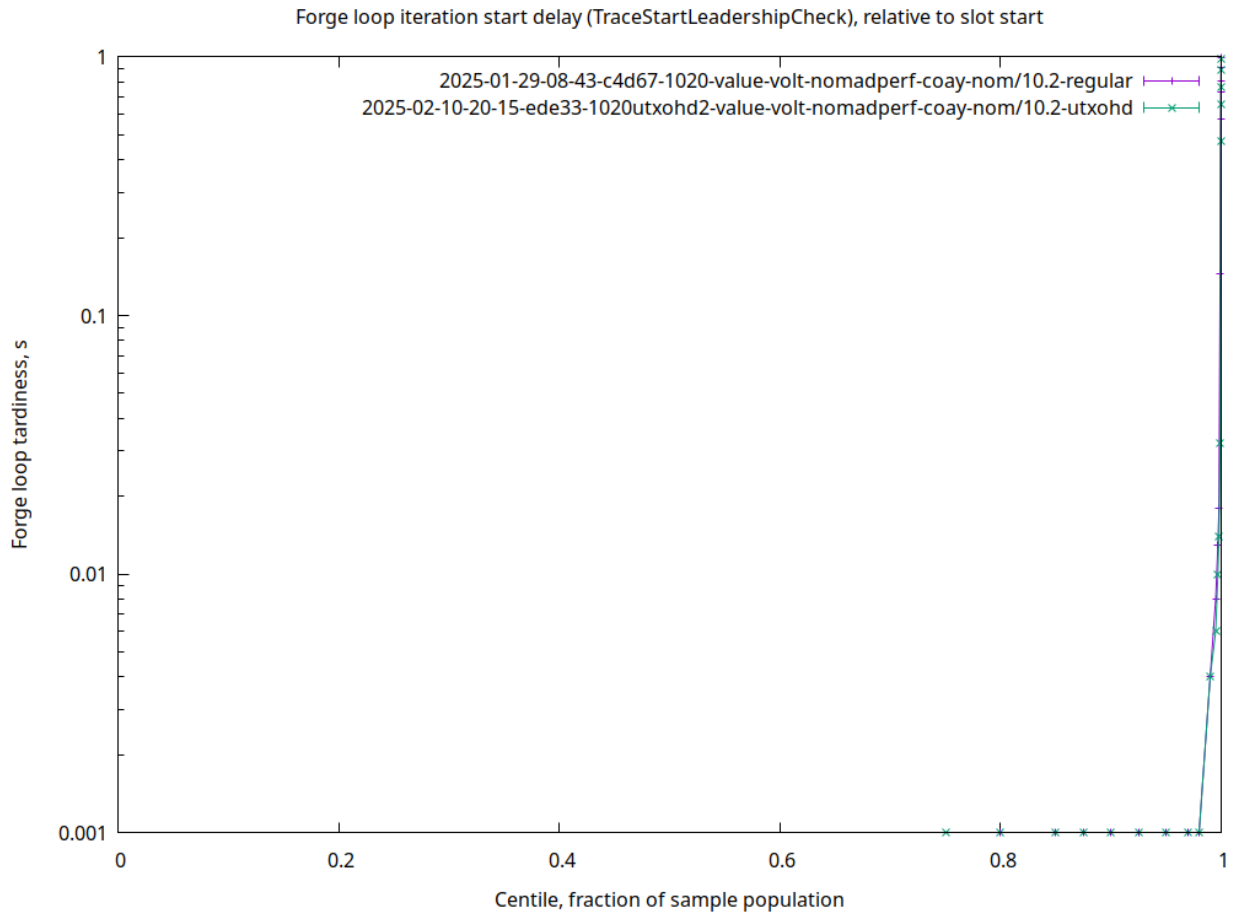
RTS live GC dataset (Live) RTS-reported GC live data size, MB



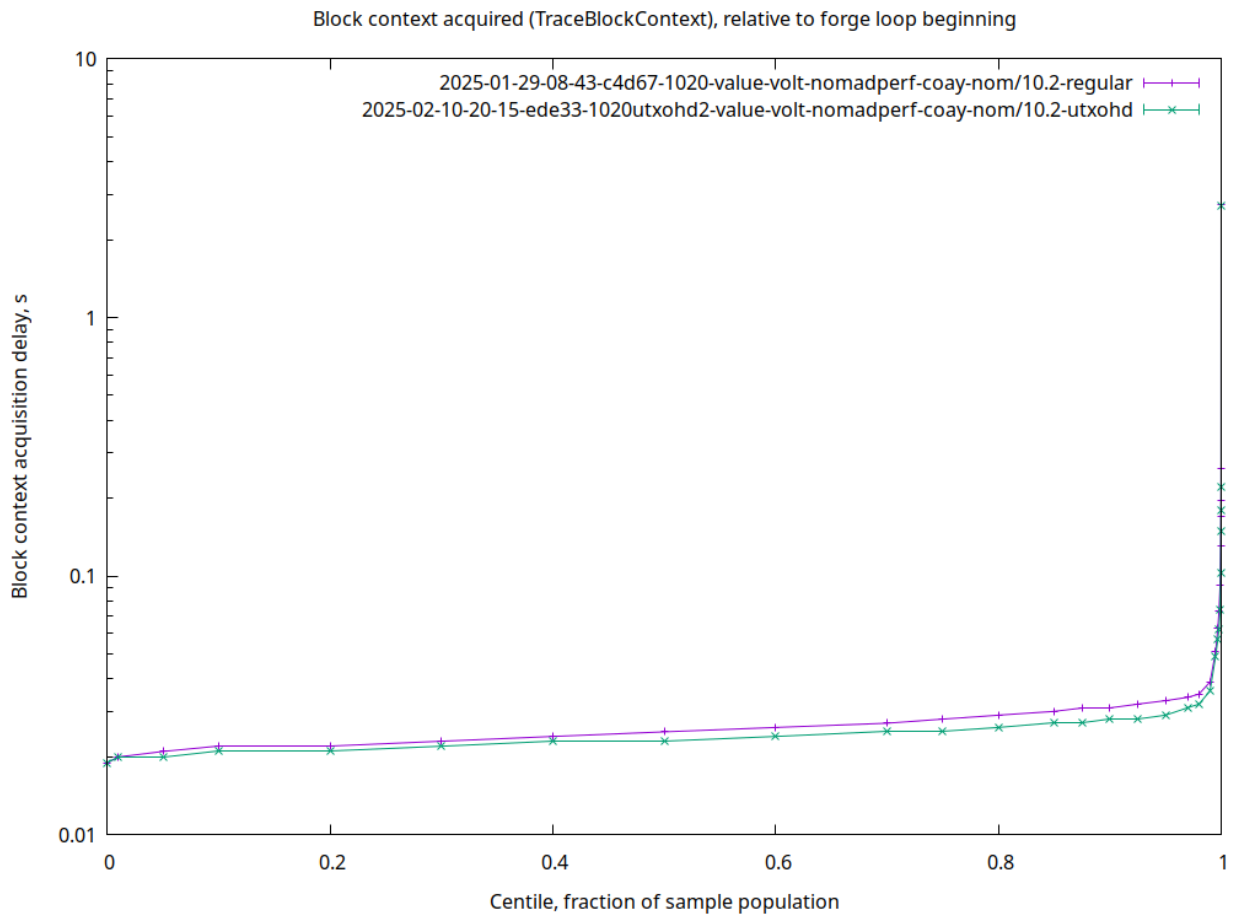
**Kernel RSS (RSS)** Kernel-reported RSS (Resident Set Size) of the process, MB



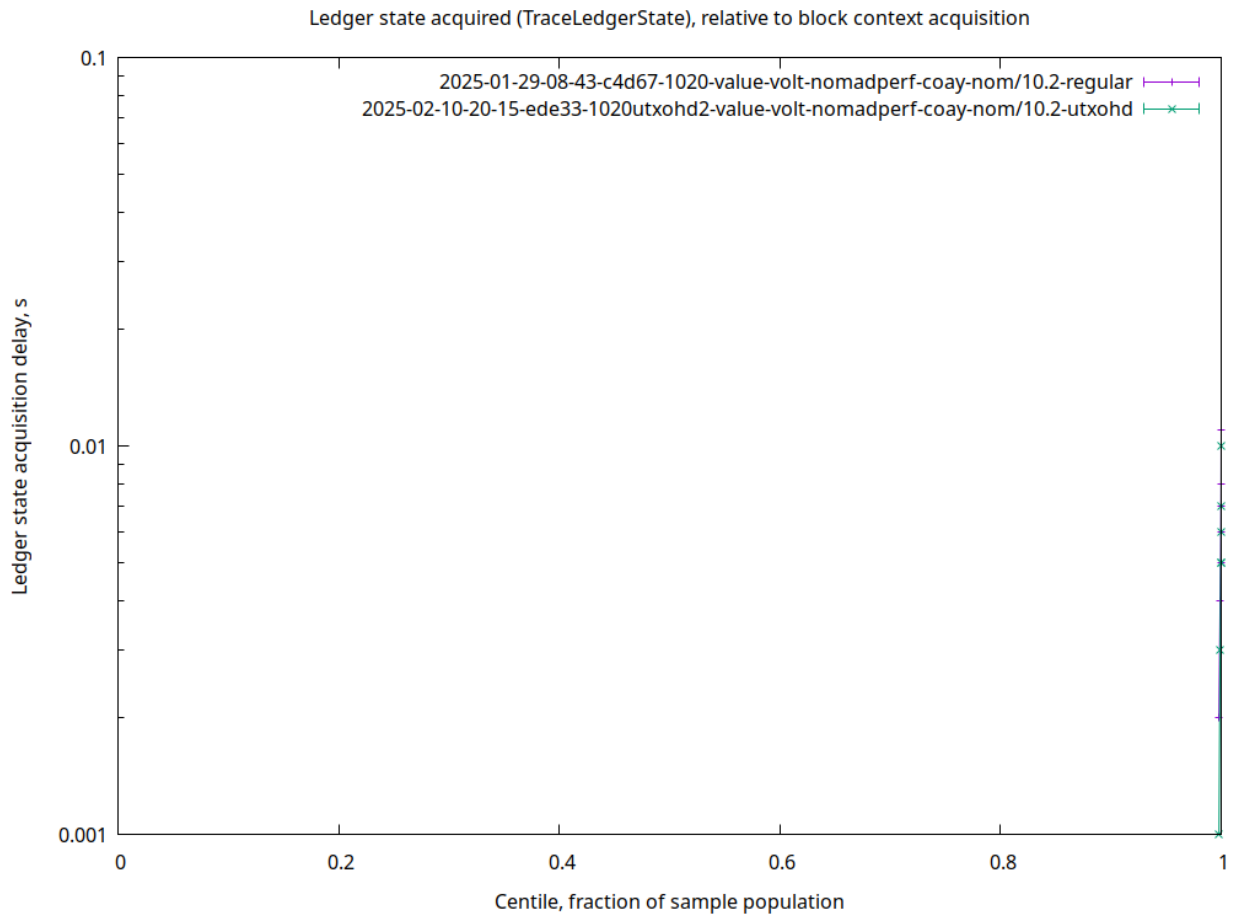
**Forge loop tardiness (cdfStarted)** Forge loop iteration start delay (TraceStartLeadershipCheck), relative to slot start



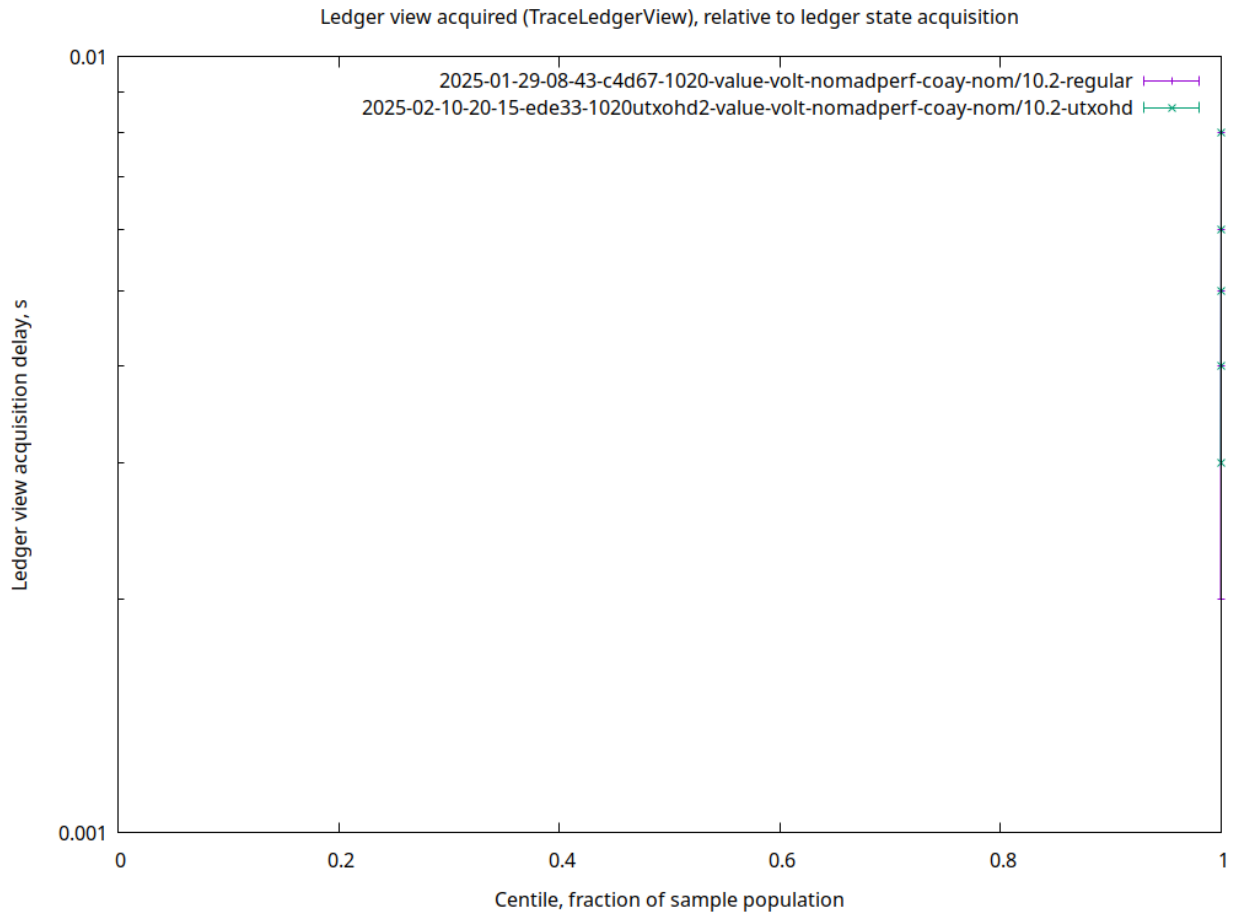
**Block context acquisition delay (cdfBlkCtx)** Block context acquired (TraceBlockContext), relative to forge loop beginning



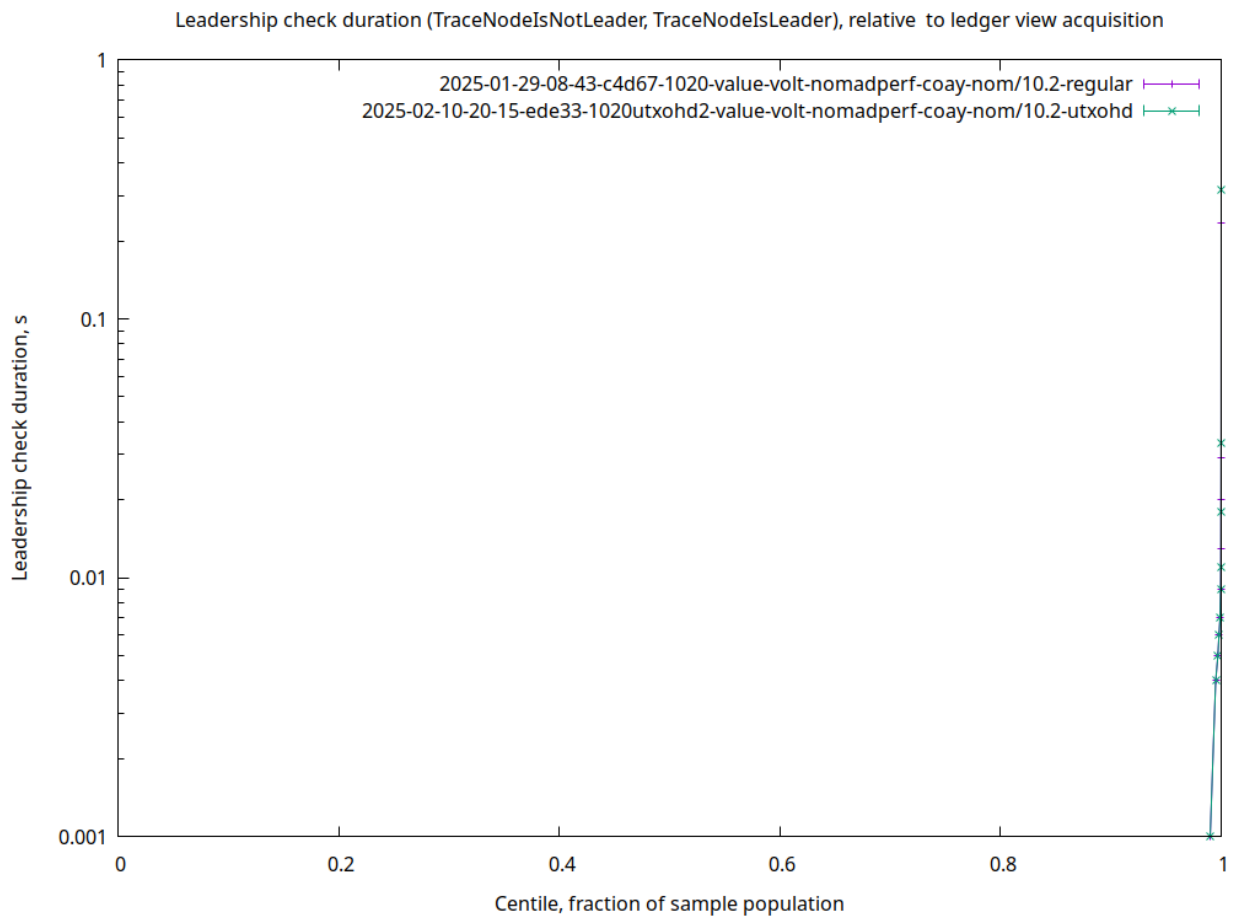
**Ledger state acquisition delay (cdfLgrState)** Ledger state acquired (TraceLedgerState), relative to block context acquisition



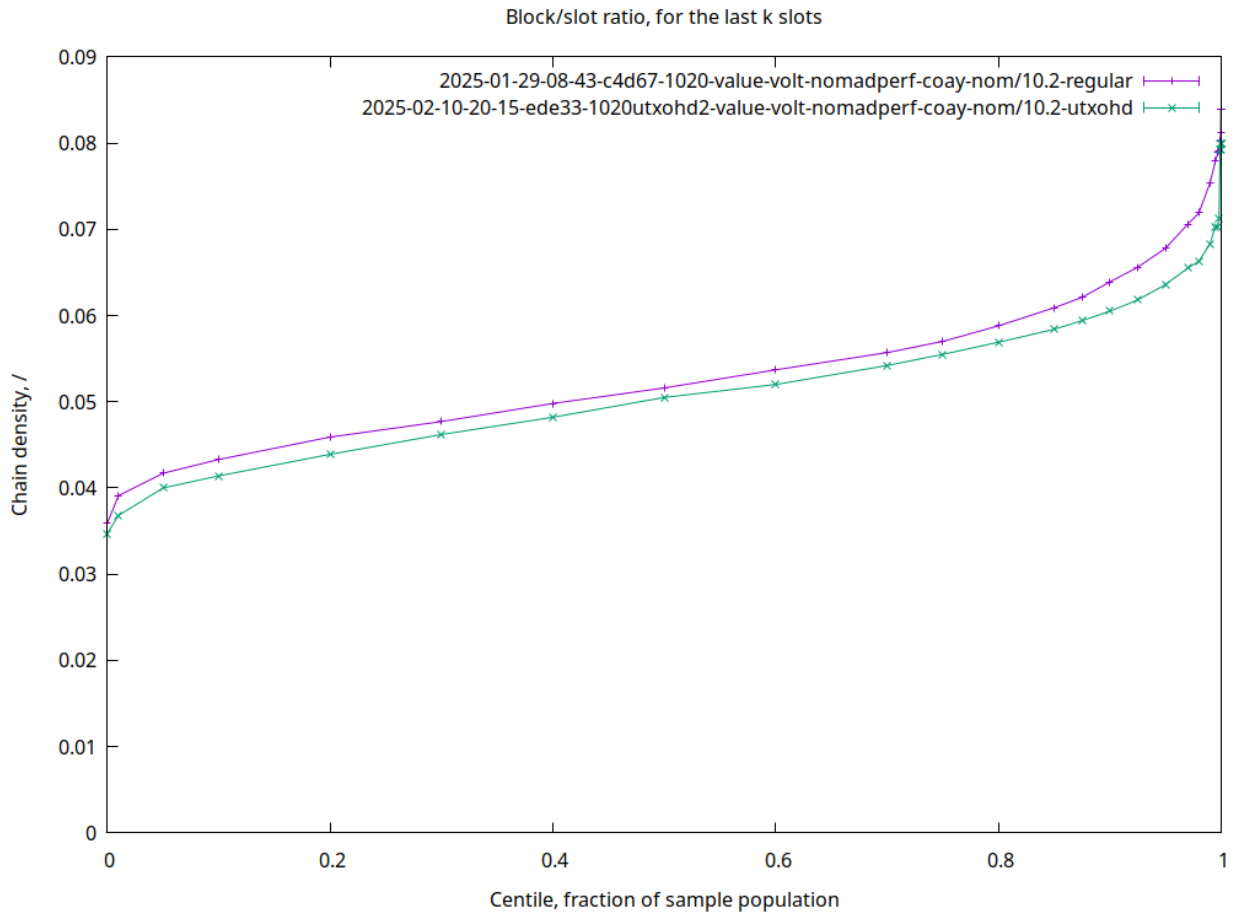
**Ledger view acquisition delay (cdfLgrView)** Ledger view acquired (TraceLedgerView), relative to ledger state acquisition



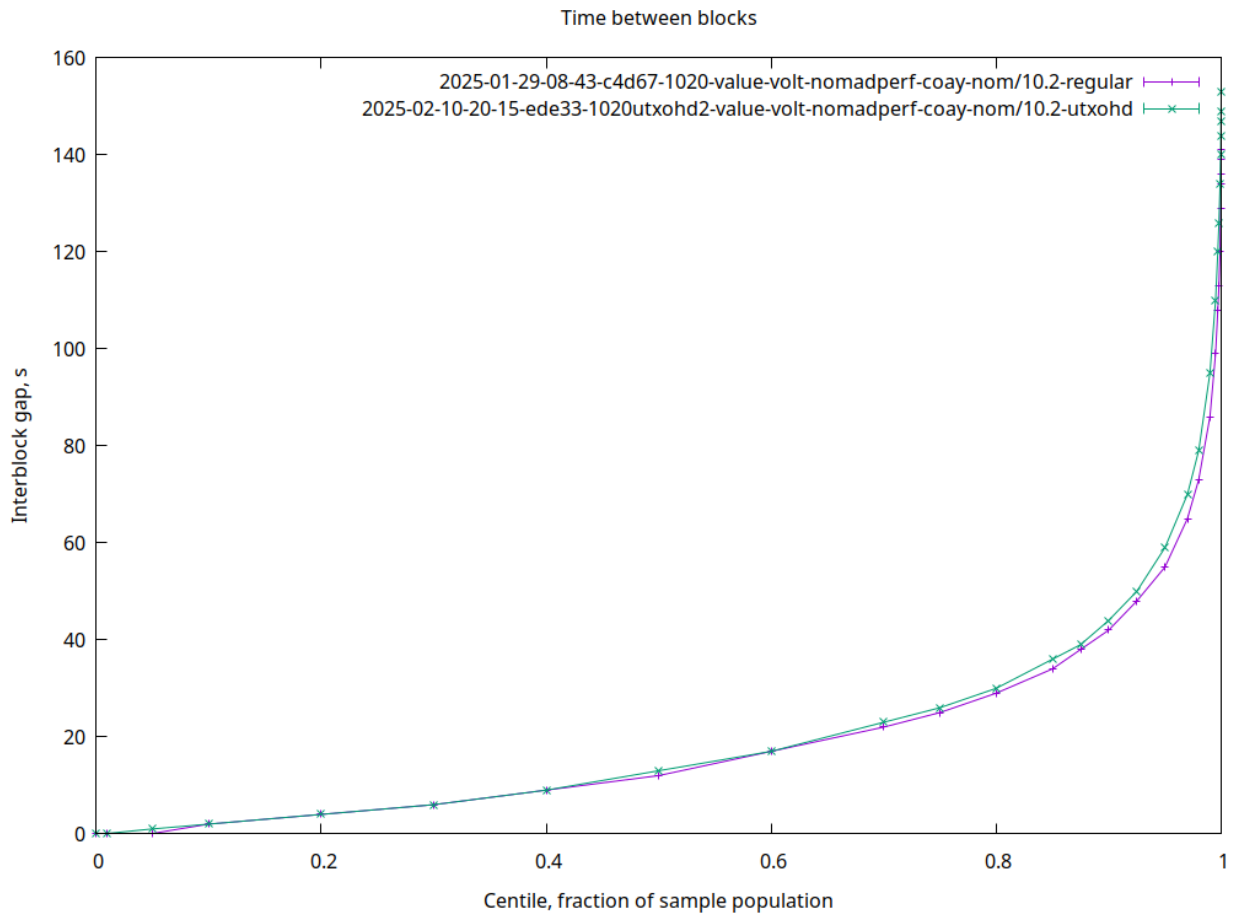
**Leadership check duration (cdfLeading)** Leadership check duration (TraceNodeIsNotLeader, TraceNodeIsLeader), relative to ledger view acquisition



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

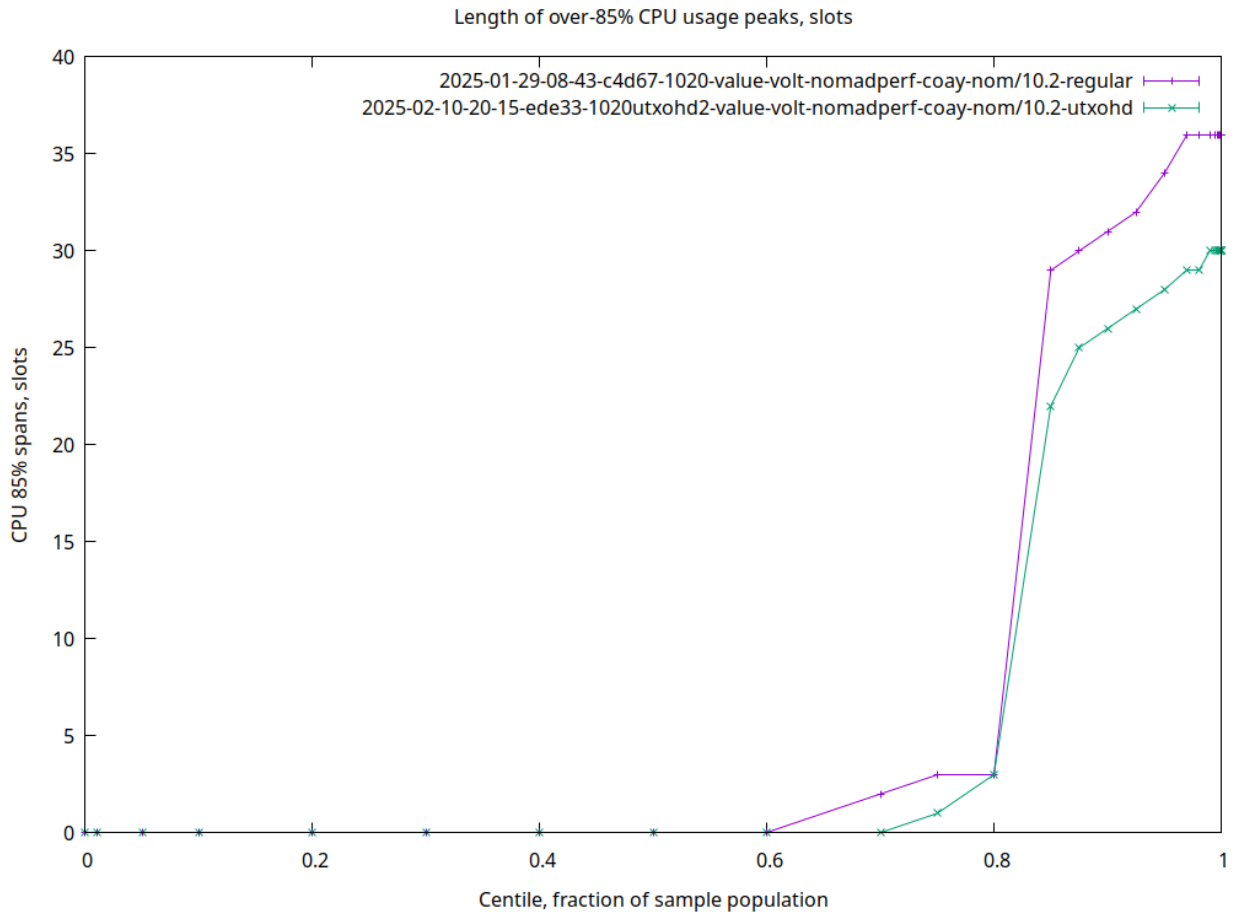


Interblock gap (cdfBlockGap) Time between blocks

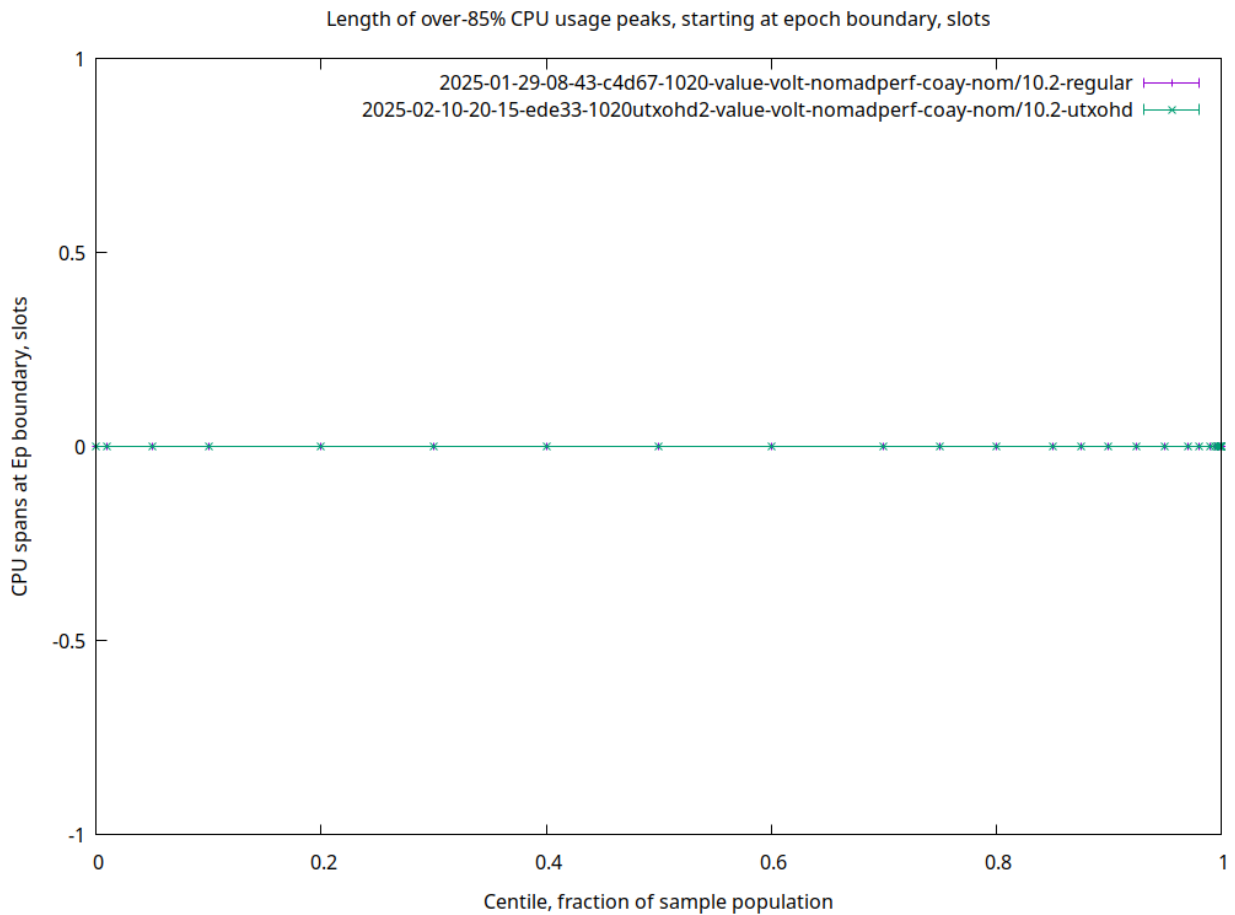


CPU 85% spans (cdfSpanLensCpu) Length of over-85% CPU usage peaks, slots

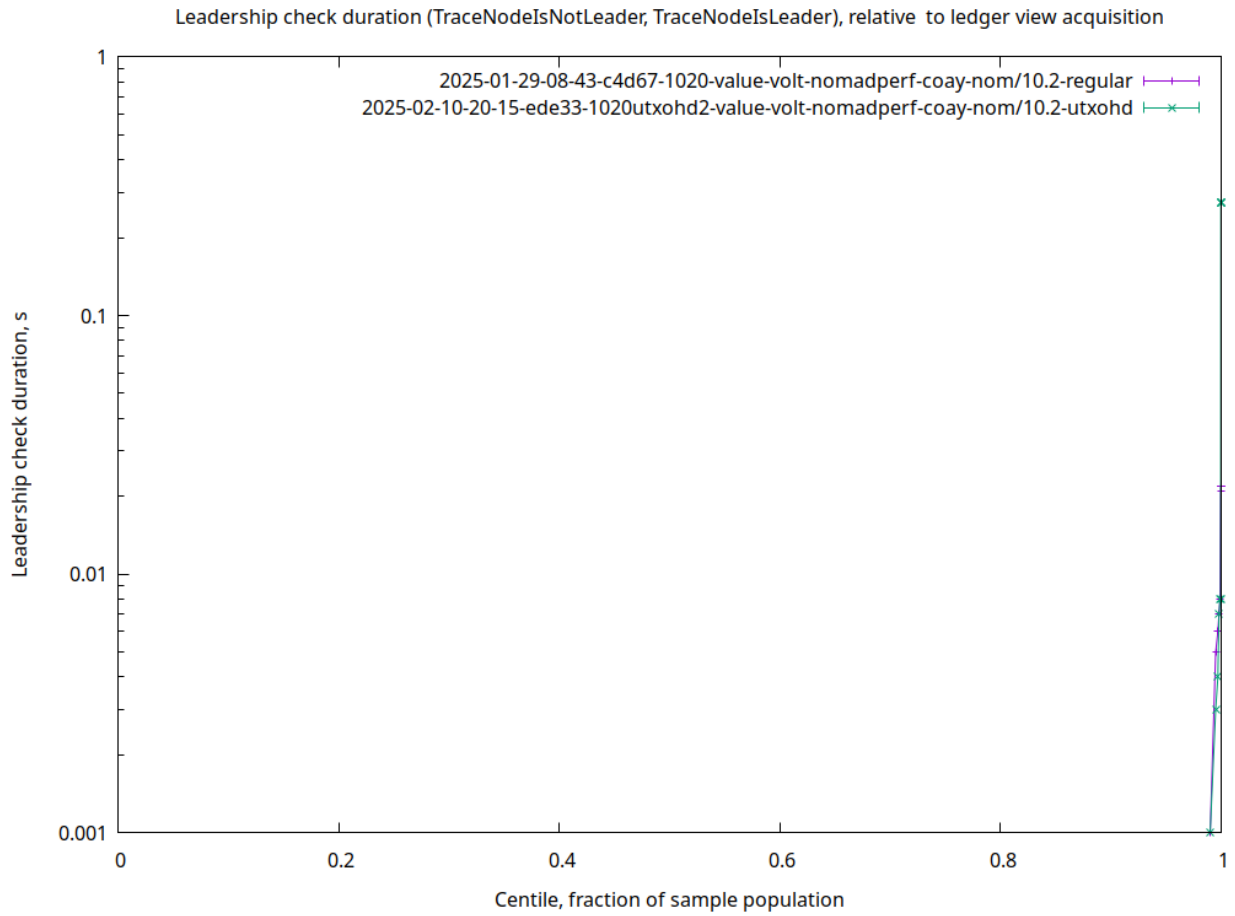




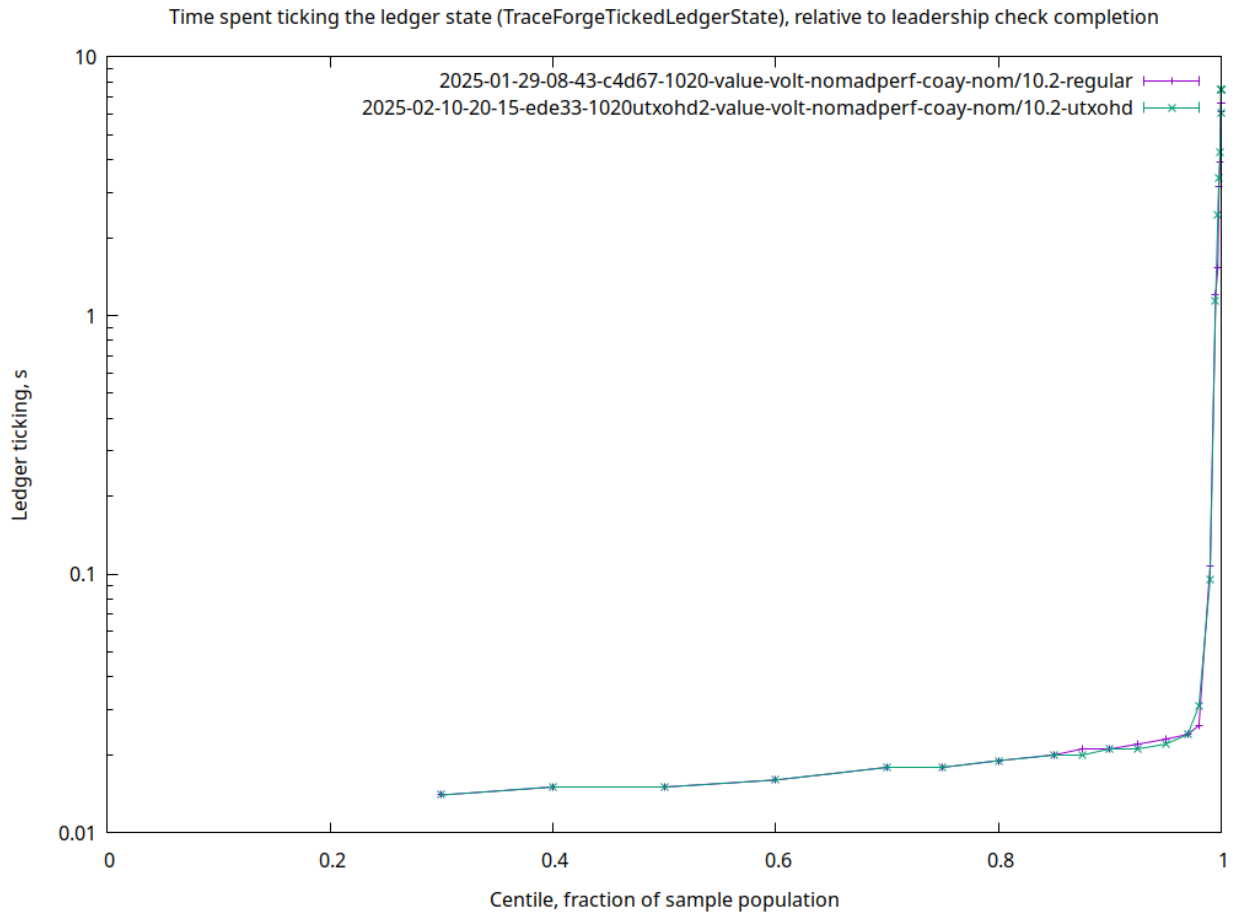
CPU spans at Ep boundary (cdfSpanLensCpuEpoch) Length of over-85% CPU usage peaks, starting at epoch boundary, slots



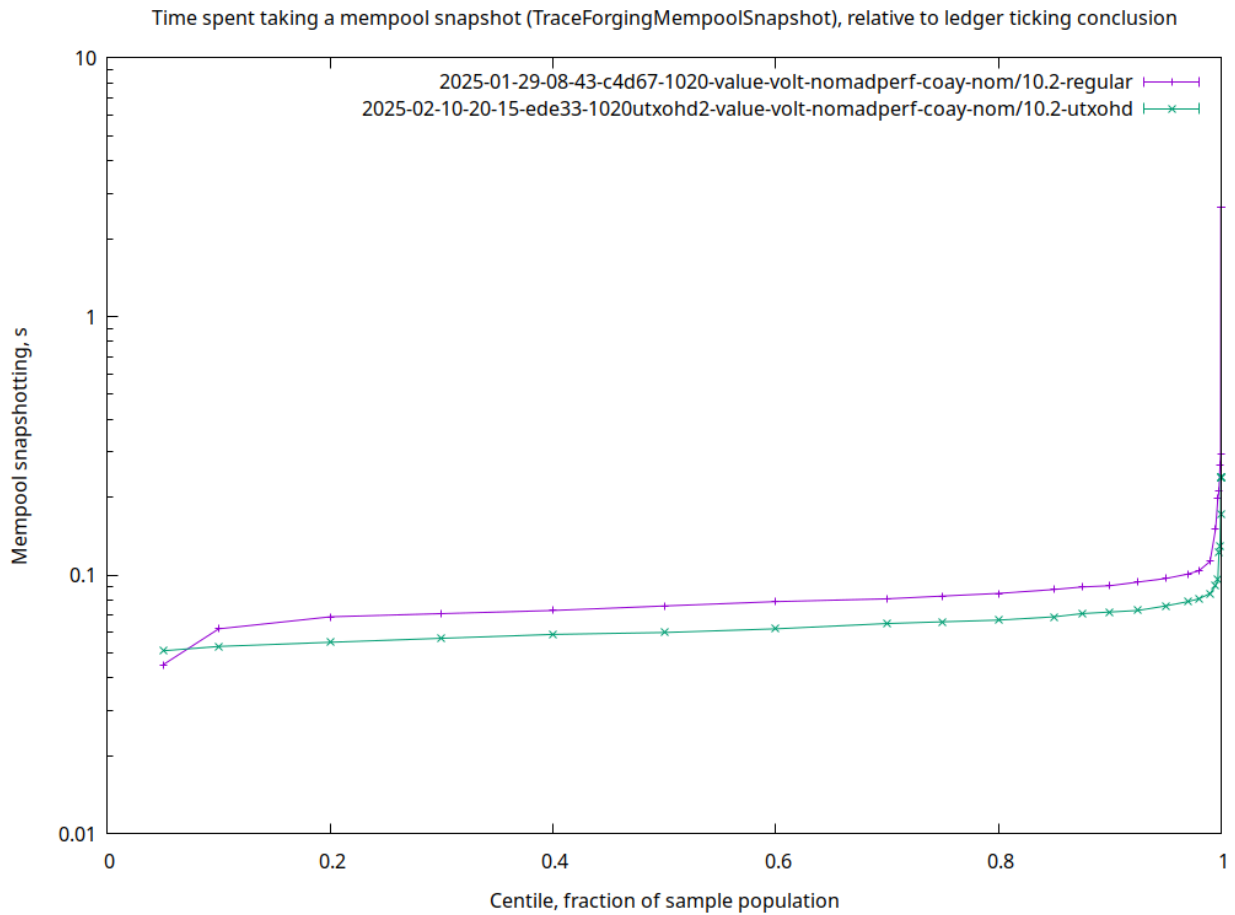
**Leadership check duration (cdfForgerLead)** Leadership check duration (TraceNodeIsNotLeader, TraceNodeIsLeader), relative to ledger view acquisition



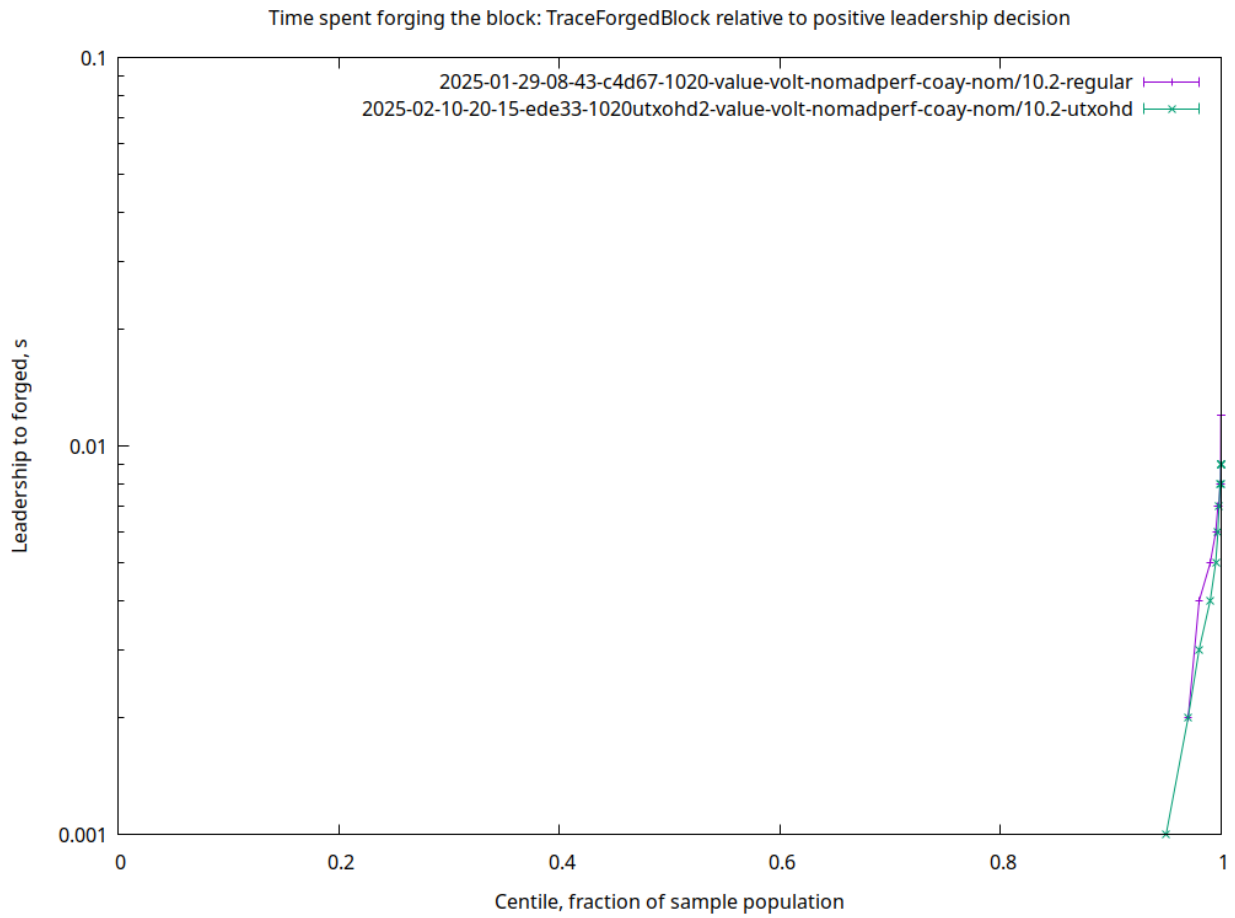
**Ledger ticking (cdfForgerTicked)** Time spent ticking the ledger state (TraceForgeTickedLedgerState), relative to leadership check completion



**Mempool snapshotting (cdfForgerMemSnap)** Time spent taking a mempool snapshot (TraceForgingMempool-Snapshot), relative to ledger ticking conclusion

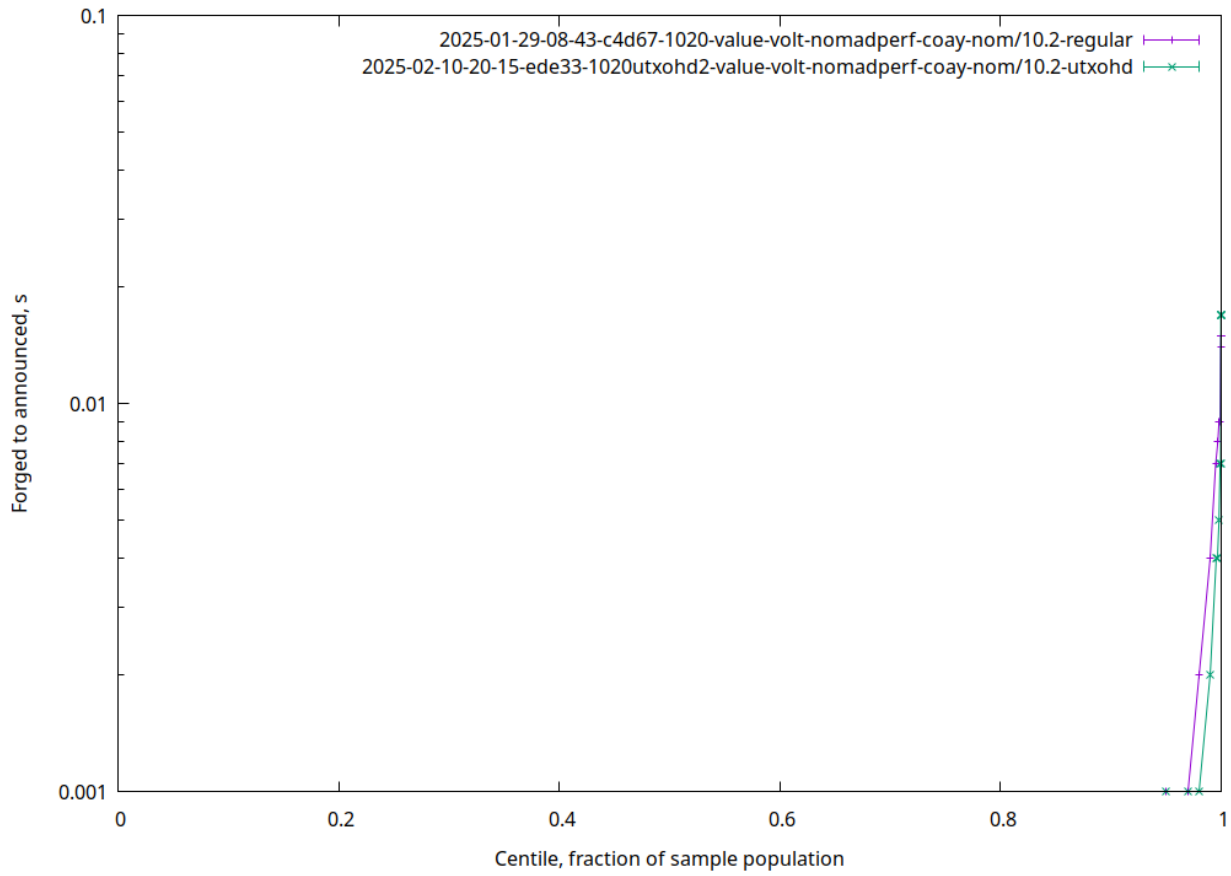


**Leadership to forged (cdfForgerForge)** Time spent forging the block: TraceForgedBlock relative to positive leadership decision



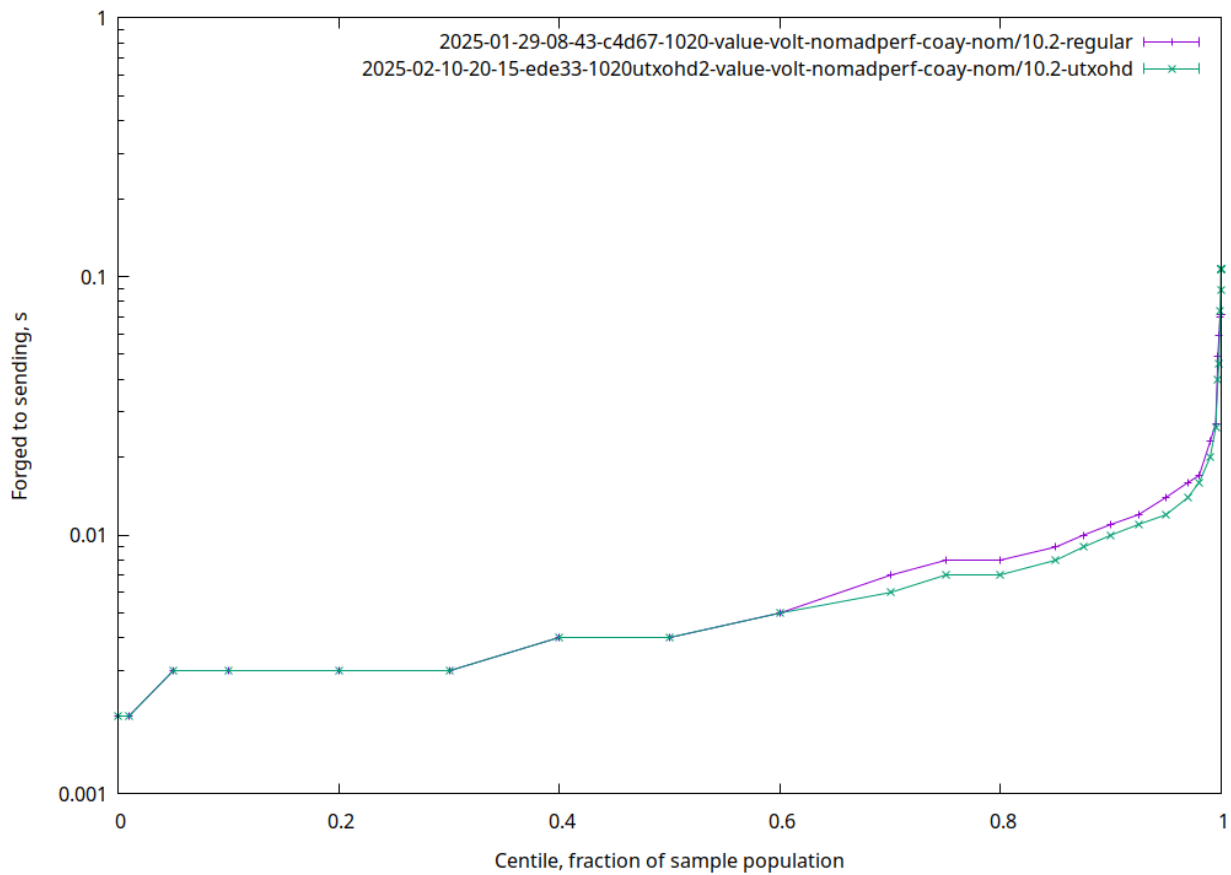
**Forged to announced (cdfForgerAnnounce)** Time between block forging completion and header announcement (ChainSyncServerEvent.TraceChainSyncServerRead.AddBlock)

Time between block forging completion and header announcement (ChainSyncServerEvent.TraceChainSyncServerRead.AddBloc

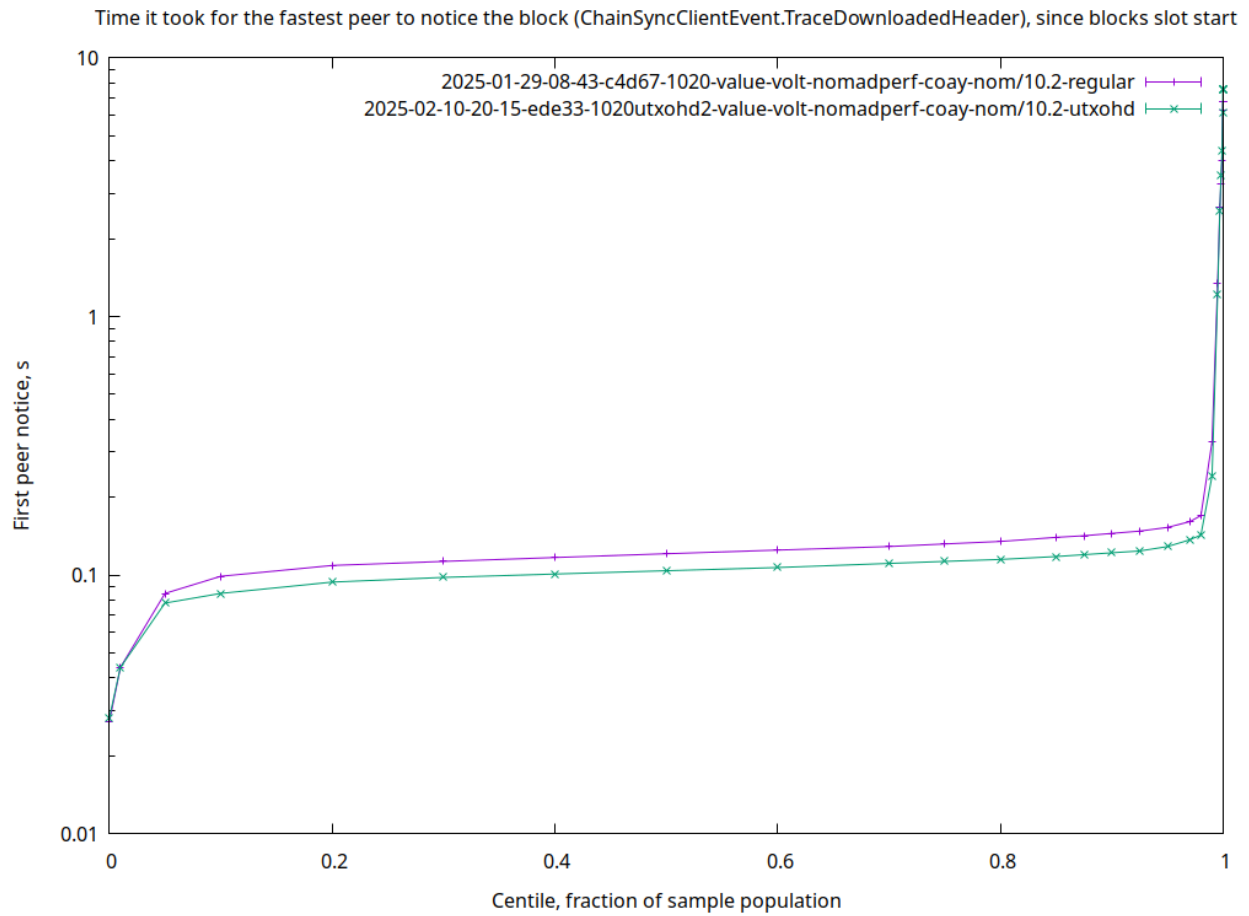


**Forged to sending (cdfForgerSend)** Time between block forging completion and begin-of-sending (TraceBlockFetchServerSendBlock)

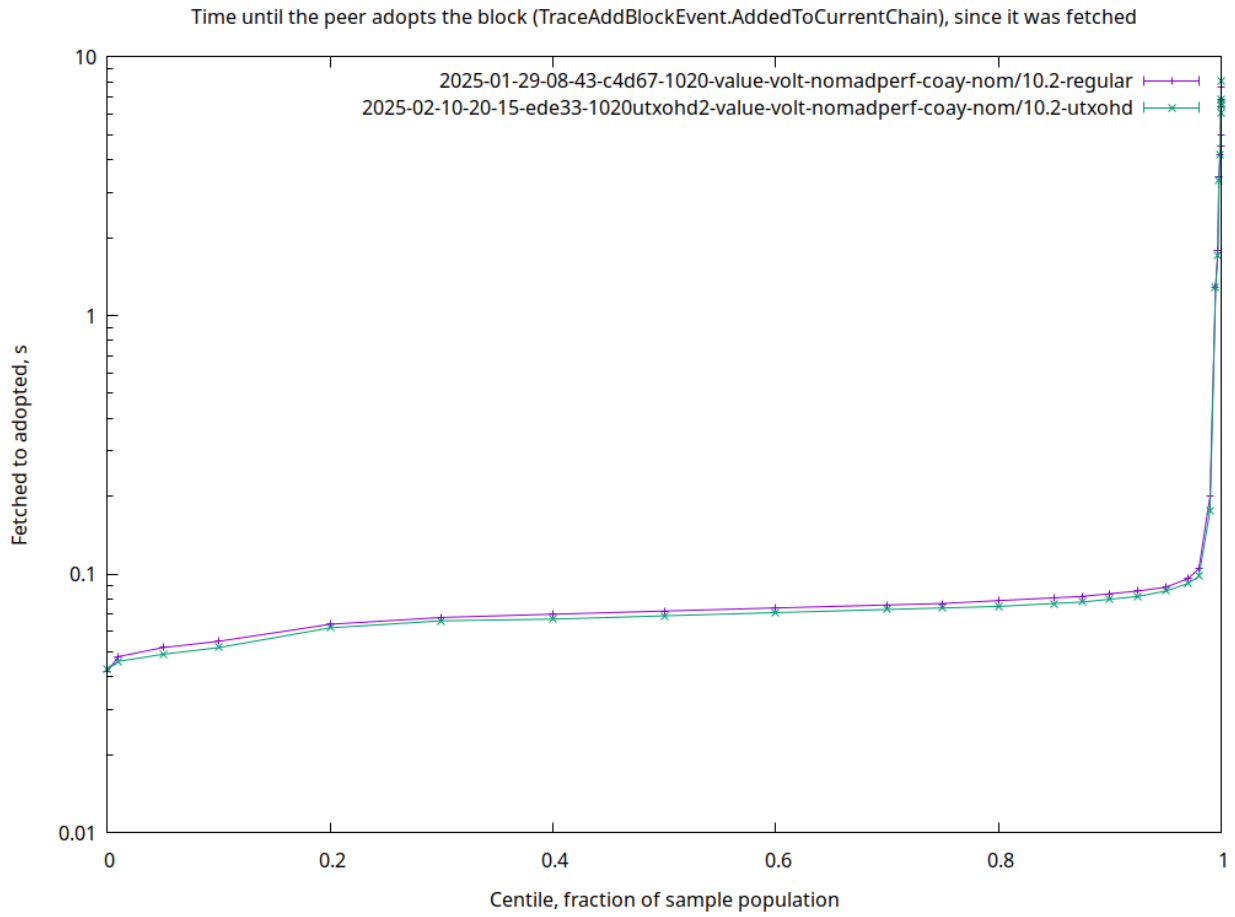
Time between block forging completion and begin-of-sending (TraceBlockFetchServerSendBlock)



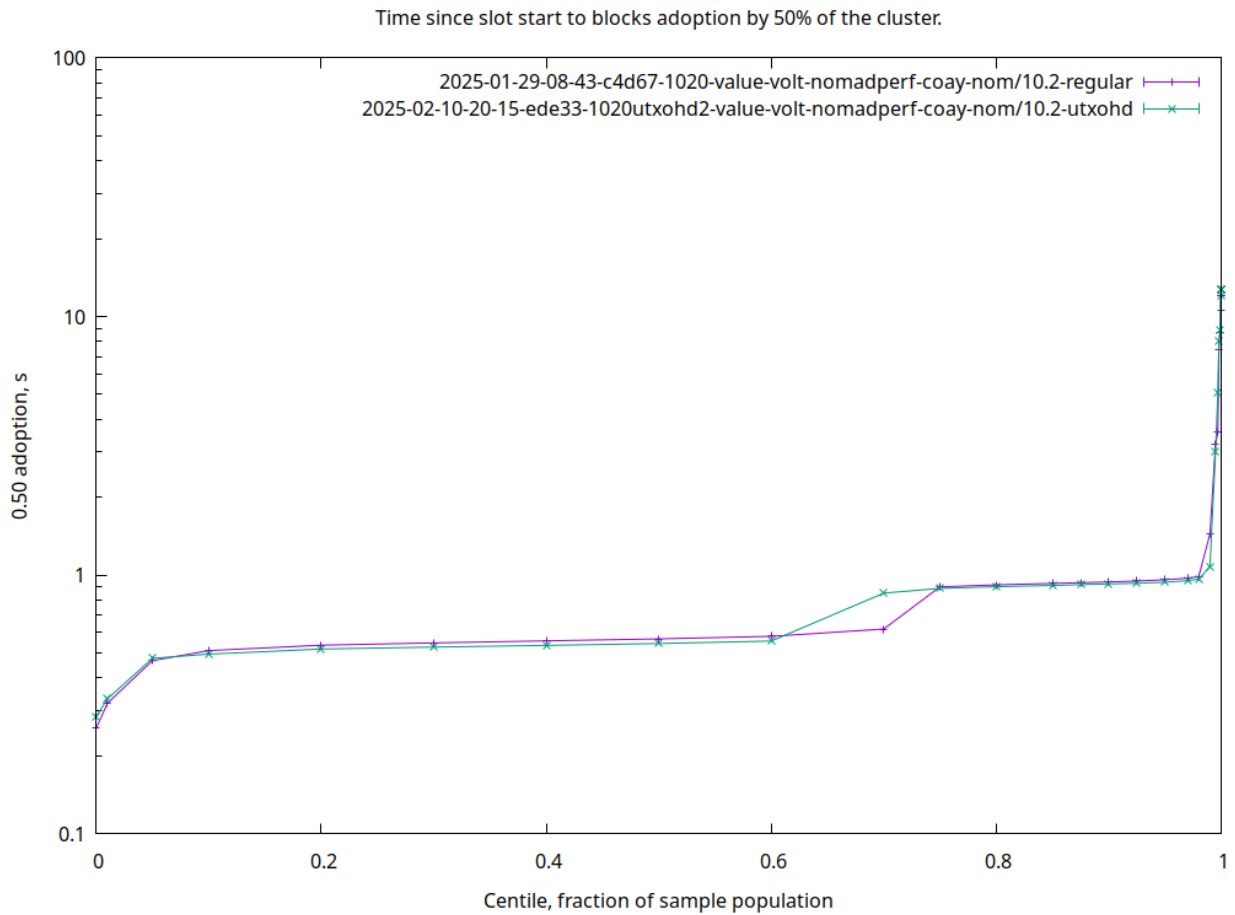
**First peer notice (cdfPeerNoticeFirst)** Time it took for the fastest peer to notice the block (ChainSyncClientEvent.TraceDownloadedHeader), since block's slot start



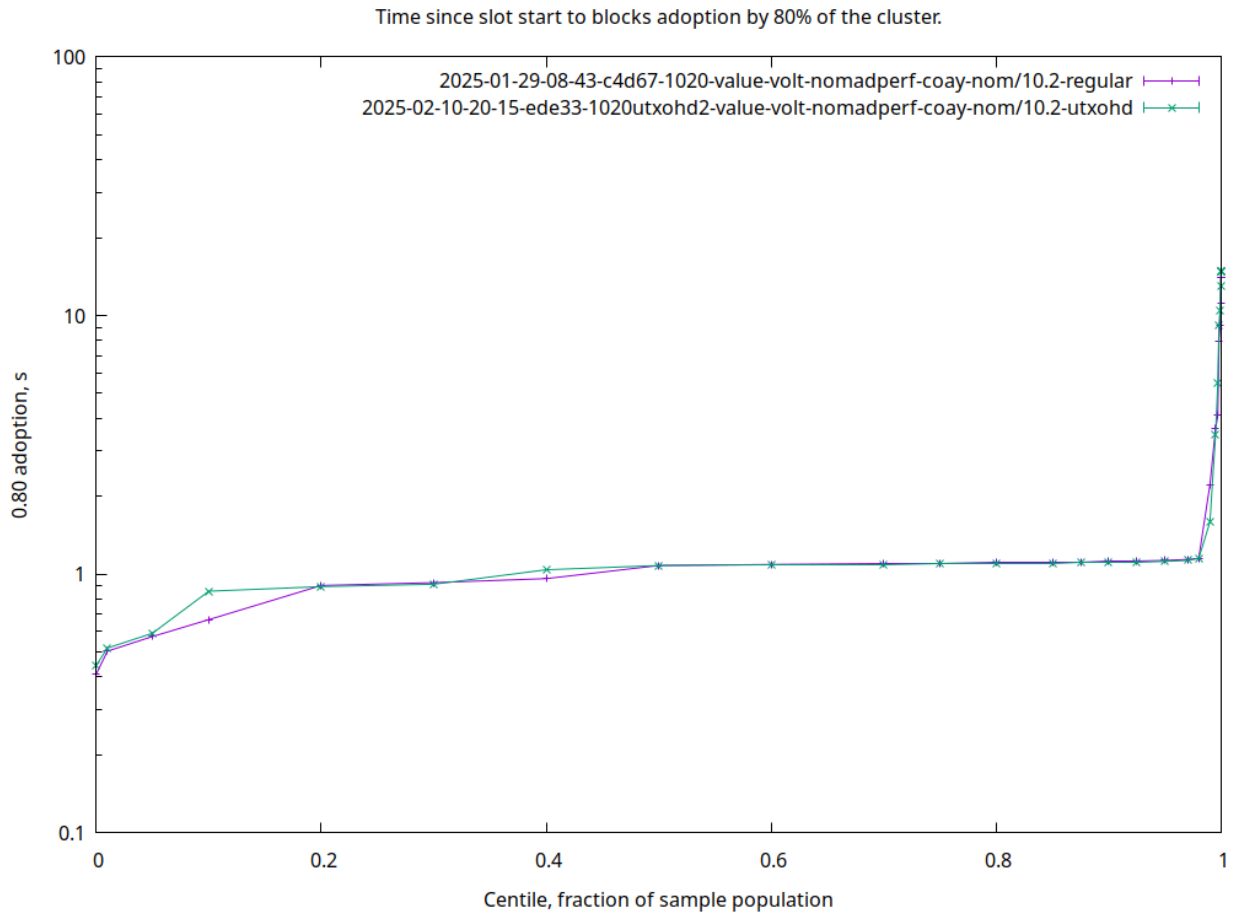
**Fetches to adopted (cdfPeerAdoption)** Time until the peer adopts the block (TraceAddBlockEvent.AddedToCurrentChain), since it was fetched



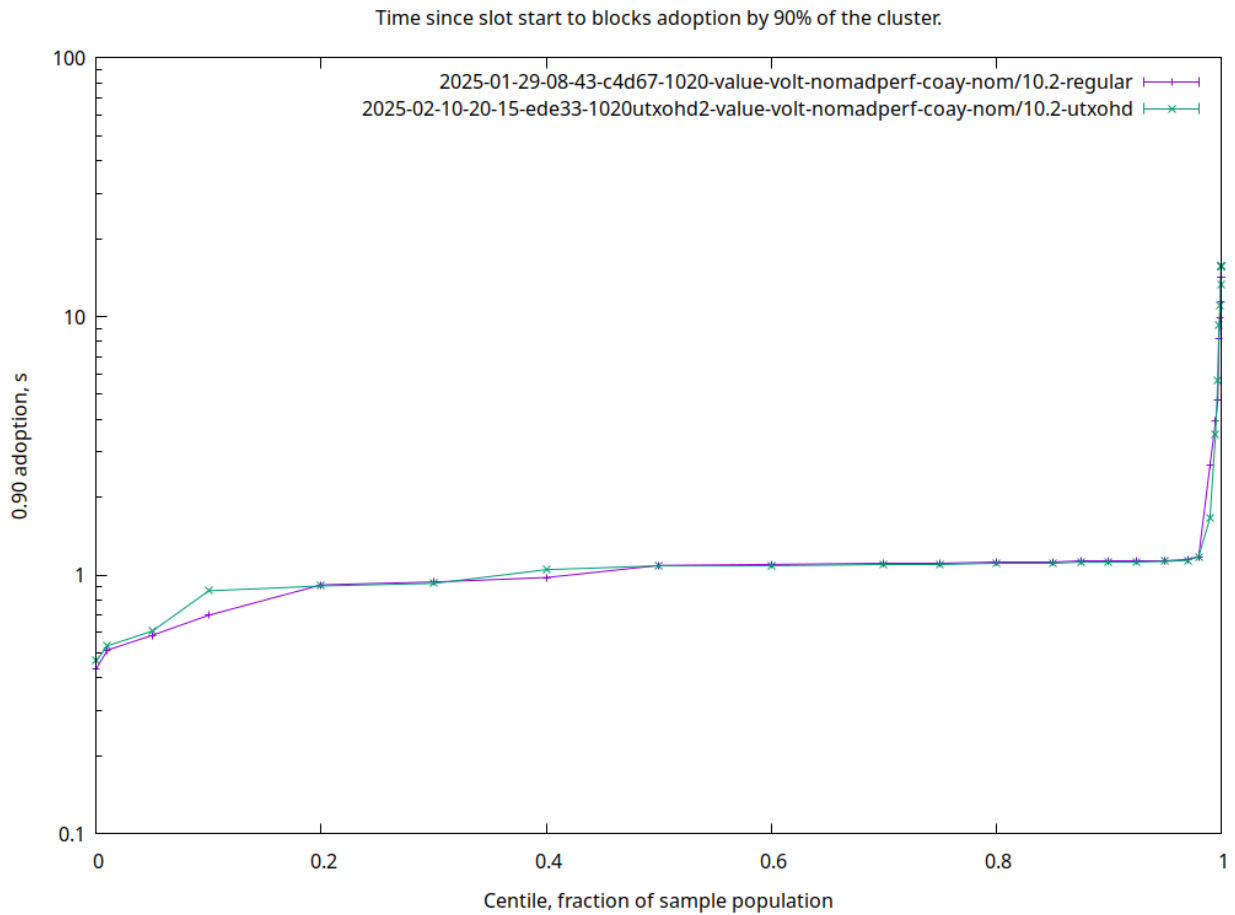
**0.50 adoption (cdf0.50)** Time since slot start to block's adoption by 50% of the cluster.



**0.80 adoption (cdf0.80)** Time since slot start to block's adoption by 80% of the cluster.



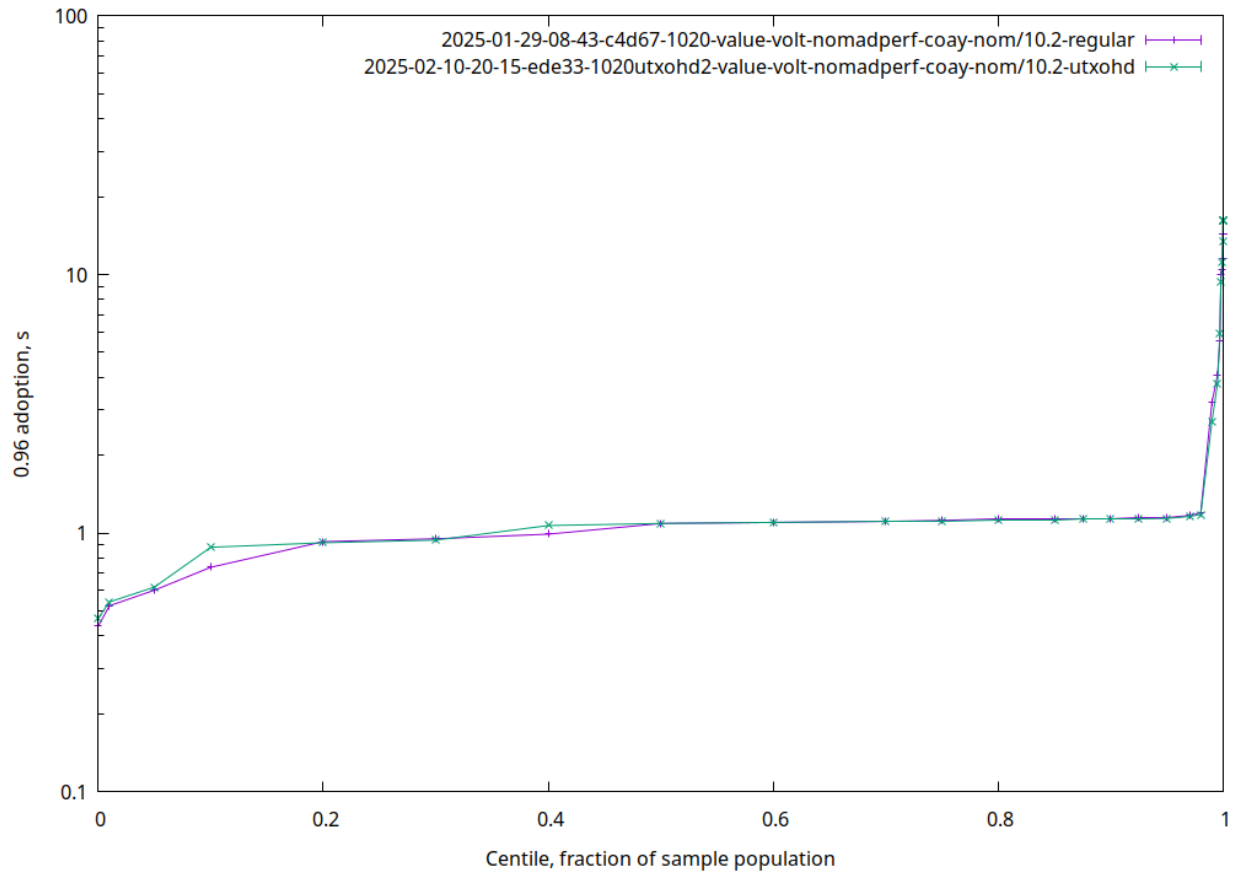
**0.90 adoption (cdf0.90)** Time since slot start to block's adoption by 90% of the cluster.



**0.96 adoption (cdf0.96)** Time since slot start to block's adoption by 96% of the cluster.



Time since slot start to blocks adoption by 96% of the cluster.



## Part II

# Appendix B: data dictionary

# Chapter 4

## 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 block ratio (cdfBlocksChainedRatio)** For each host, ratio of blocks that made into chain / all forged

**Filtered to chained block ratio (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

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

**Fetches to announced (cdfPeerAnnounce)** Time it took a peer to announce the block (ChainSyncServerEvent.TraceChainSync), 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

## Chapter 5

# 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