

10.6-pre against 10.5

Plutus countdown loop workload

Michael Karg, Cardano Performance team

2025-11-17

# 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 10.6-pre (Conway) relative to 10.5 (Conway), under Plutus countdown loop workload.

	10.5	10.6-pre
Analysis date	2025-08-03	2025-11-16
Cluster system start date	2025-08-02	2025-11-15
Cluster system start time	13:33:43	16:19:08
Identifier	10.5.0	10.6.0
Run batch	1050mmap	1060pre
GHC version	9.6.5	9.6.6
cardano-node version	10.5.0	10.6.0
ouroboros-consensus version	0.27.0.0	0.28.0.2
ouroboros-network version	0.21.2.0	0.22.4.0
cardano-ledger-core version	1.17.0.0	1.18.0.0
plutus-core version	1.45.0.0	1.53.1.0
cardano-crypto version	1.3.0	1.3.0
cardano-prelude version	0.2.1.0	0.2.1.0
cardano-node git	8db07f6	f5ac0eb
ouroboros-consensus git	8e3afe1	d3c4b5c
ouroboros-network git	879683d	ee53402
cardano-ledger-core git	a9e78ae	faa7a9d
plutus-core git	ba16ec6	babbed7
cardano-crypto git	unknown	unknown
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	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	904540.67307	881191.53846
Log objects analysed per host	610342.63461	609602.07692
Host run time, s	71849.1	71906.5
Host log line rate, Hz	12.589	12.254
Total log objects analysed	31737817	31699308
Run time, s	71853	71913
Analysed run duration, s	56010	56041
Run time efficiency	0.77	0.77
Node start spread, s	4.8371453	5.5210645
Node stop spread, s	1.9440116	5.1150143
Slots analysed	56008	56037
Blocks analysed	2658	2651
Blocks rejected	871	842

## Chapter 2

# Analysis

### 2.1 Resource Usage

	10.5	10.6-pre	$\Delta$	$\Delta\%$
Forge loop starts, #	0.99931	0.99864	-0.001	0
Process CPU usage, %	3.9406	4.1001	0.160	4
RTS GC CPU usage, %	0.23161	0.20602	-0.026	-11
RTS Mutator CPU usage, %	3.7045	3.8875	0.183	5
Major GCs, #	0.0011	0.00088	-0.000	0
Minor GCs, #	0.98239	0.39442	-0.588	-60
Kernel RSS, MB	6333.9	7280.6	946.700	15
RTS heap size, MB	6272.3	7217.9	945.600	15
RTS live GC dataset, MB	2476.6	3224.6	748.000	30
RTS alloc rate, MB/s	28.7	11.648	-17.052	-59
Filesystem reads, KB/s	1e-05	0.0	-0.000	0
Filesystem writes, KB/s	226.97	223.8	-3.170	-1
CPU 85% spans, slots	10.518	12.164	1.646	16
Sample count	(291>)	(291>)		

### 2.2 Anomaly control

	10.5	10.6-pre	$\Delta$	$\Delta\%$
Blocks per host, blocks	69.673	68.75	-0.923	-1
Filtered to chained block ratio, /	0.75208	0.75935	0.007	1
Chained to forged block ratio, /	0.97381	0.97688	0.003	0
Height & slot battles, blocks	0.00037	0.00113	0.001	270
Block size, B	2996.0	2996.0	0.000	0
Sample count	(52)	(52)		

## 2.3 Forging

	10.5	10.6-pre	$\Delta$	$\Delta\%$
Started forge loop iteration, s	0.00102	0.00128	0.000	0
Acquired block context, s	6e-05	6e-05	0.000	0
Acquired ledger state, s	9e-05	0.00012	0.000	0
Acquired ledger view, s	2e-05	3e-05	0.000	0
Leadership check duration, s	0.00042	0.00042	0.000	0
Ledger ticking, s	0.02144	0.02368	0.002	9
Mempool snapshotting, s	0.00151	0.00166	0.000	0
Leadership to forged, s	0.00016	0.00018	0.000	0
Forged to announced, s	0.00066	0.00065	-0.000	0
Forged to sending, s	0.00531	0.00541	0.000	0
Forged to self-adopted, s	0.04226	0.04661	0.004	9
Slot start to announced, s	0.02543	0.02812	0.003	12
Sample count	(2658)	(2651)		

## 2.4 Individual peer propagation

	10.5	10.6-pre	$\Delta$	$\Delta\%$
First peer notice, s	0.02721	0.0299	0.003	11
First peer fetch, s	0.03222	0.03509	0.003	9
Notice to fetch request, s	0.00119	0.00131	0.000	0
Fetch duration, s	0.12281	0.12331	0.001	1
Fetches to announced, s	0.0008	0.00089	0.000	0
Fetches to sending, s	0.04294	0.04439	0.001	2
Fetches to adopted, s	0.04457	0.04728	0.003	7
Sample count	(2658)	(2651)		

## 2.5 End-to-end propagation

	10.5	10.6-pre	$\Delta$	$\Delta\%$
0.50 adoption, s	0.28337	0.28889	0.006	2
0.80 adoption, s	0.44275	0.45267	0.010	2
0.90 adoption, s	0.45342	0.46576	0.012	3
0.92 adoption, s	0.45596	0.46845	0.012	3
0.94 adoption, s	0.45936	0.47134	0.012	3
0.96 adoption, s	0.46337	0.47826	0.015	3
0.98 adoption, s	0.46924	0.49362	0.024	5
1.00 adoption, s	0.48178	0.57099	0.089	18
Sample count	(2658)	(2651)		

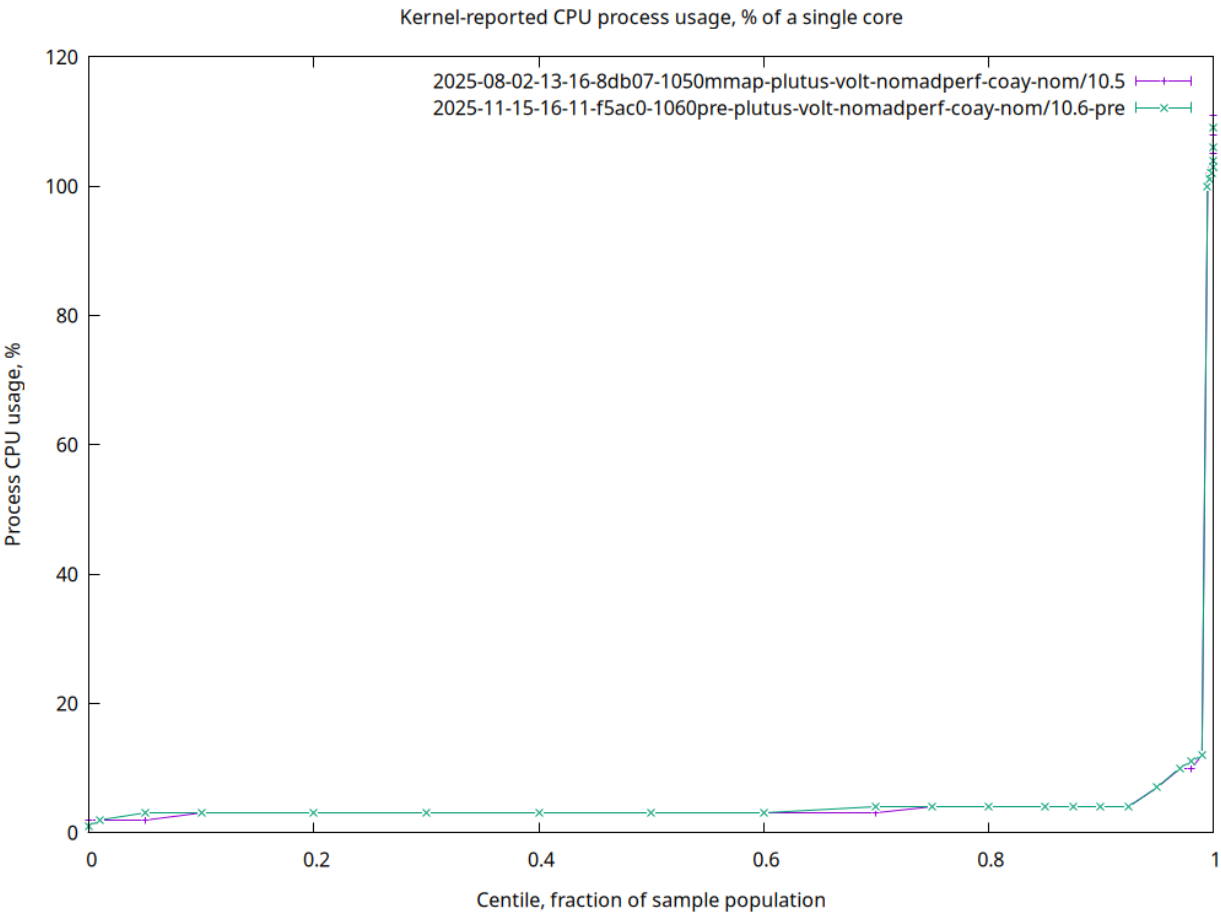
## Part I

### Appendix A: charts

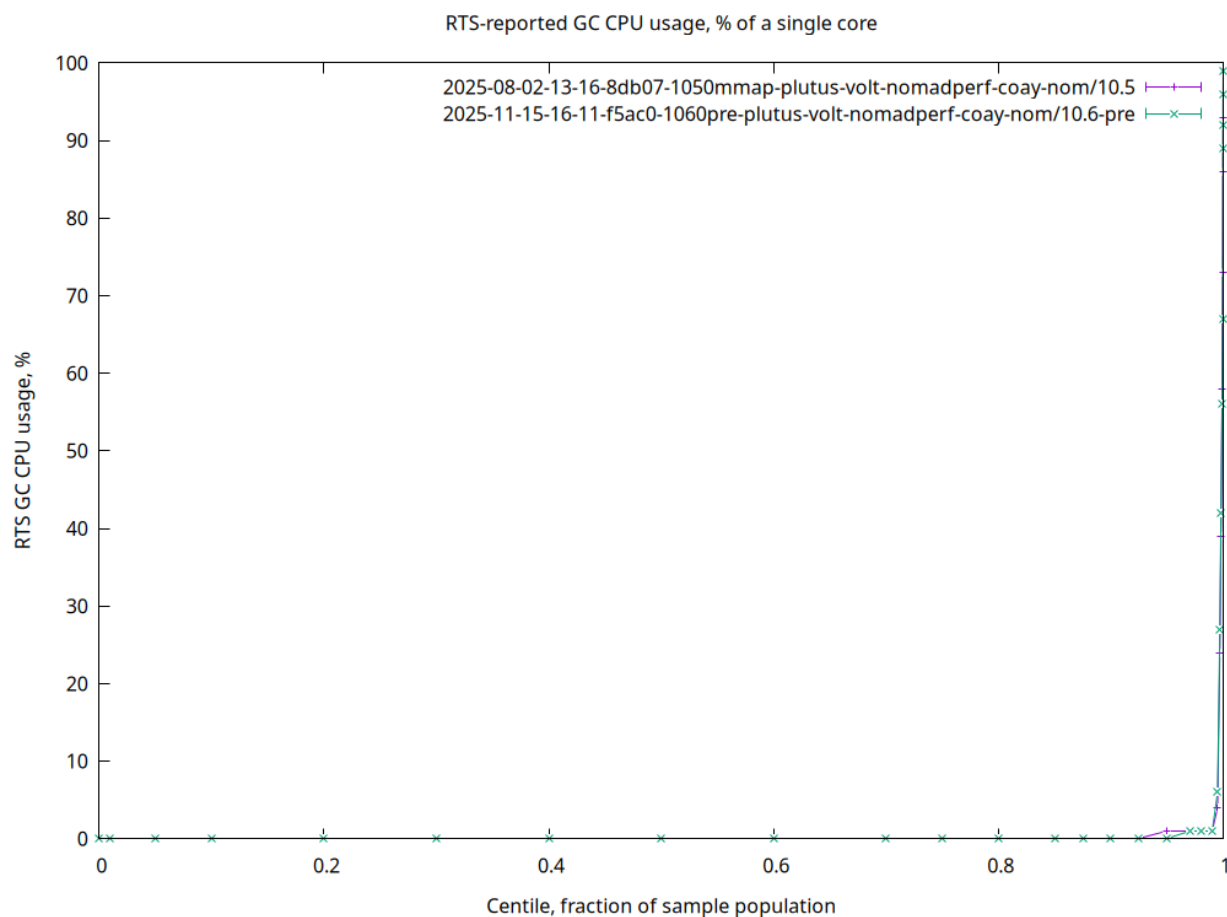
# Chapter 3

## Cluster performance charts

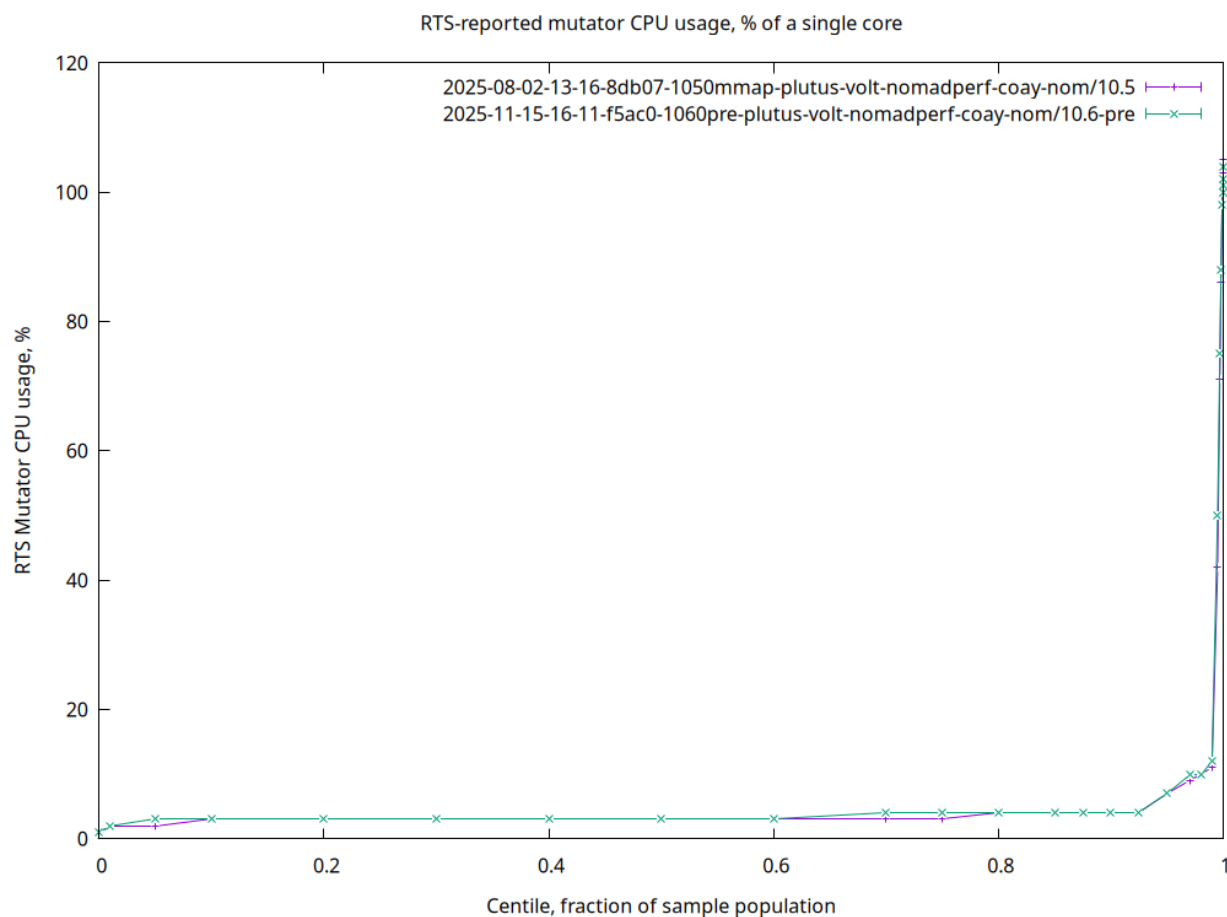
**Process CPU usage (CentiCpu)** Kernel-reported CPU process usage, % of a single core



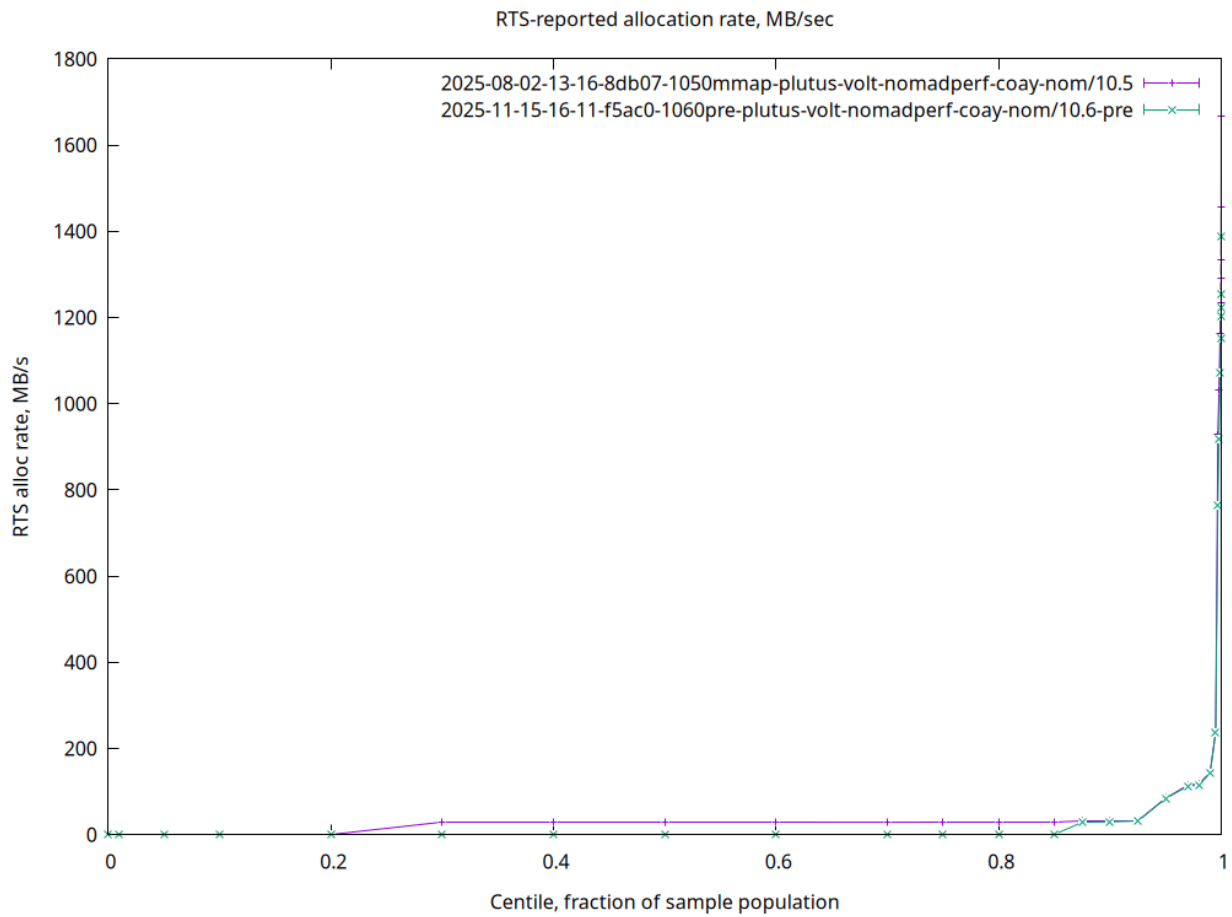
**RTS GC CPU usage (CentiGC)** RTS-reported GC CPU usage, % of a single core



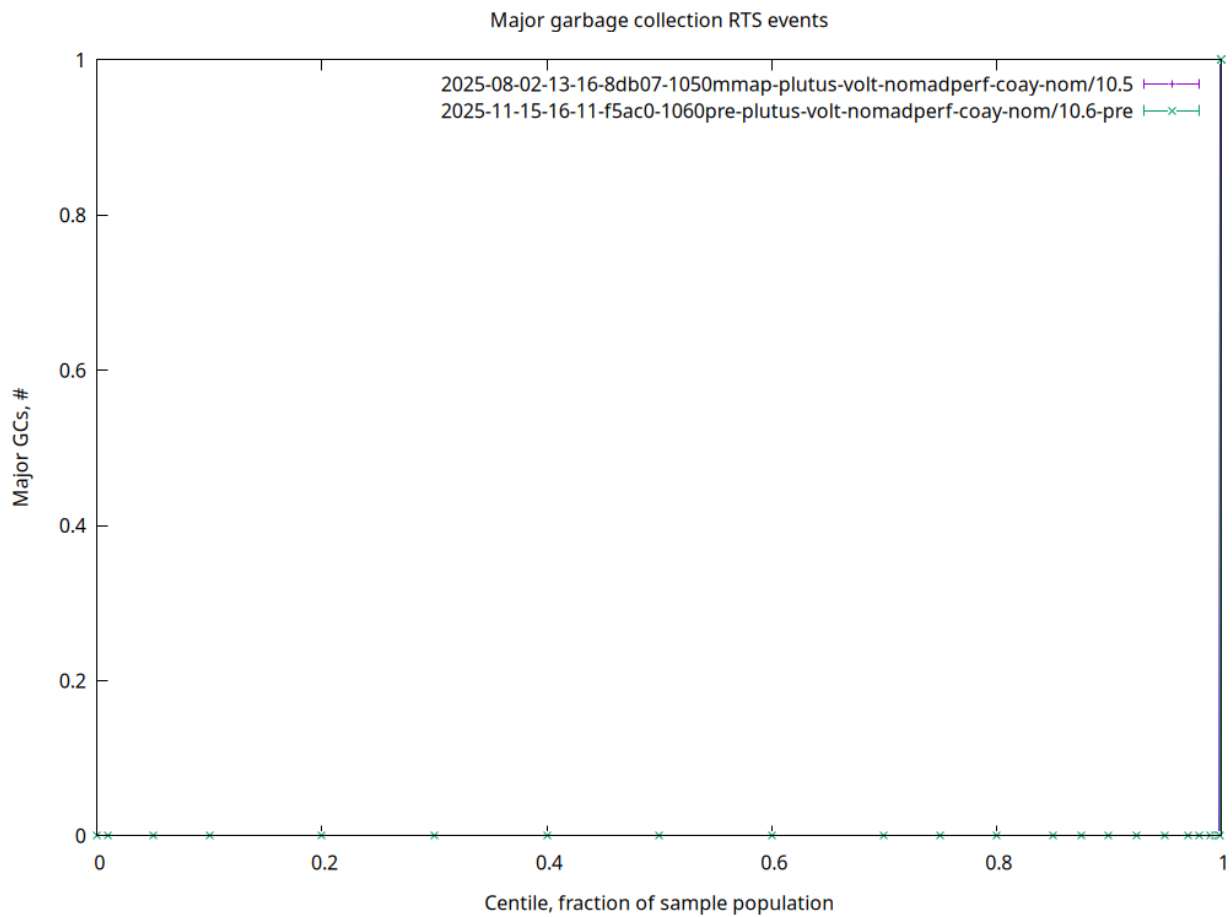
**RTS Mutator CPU usage (CentiMut)** RTS-reported mutator CPU usage, % of a single core



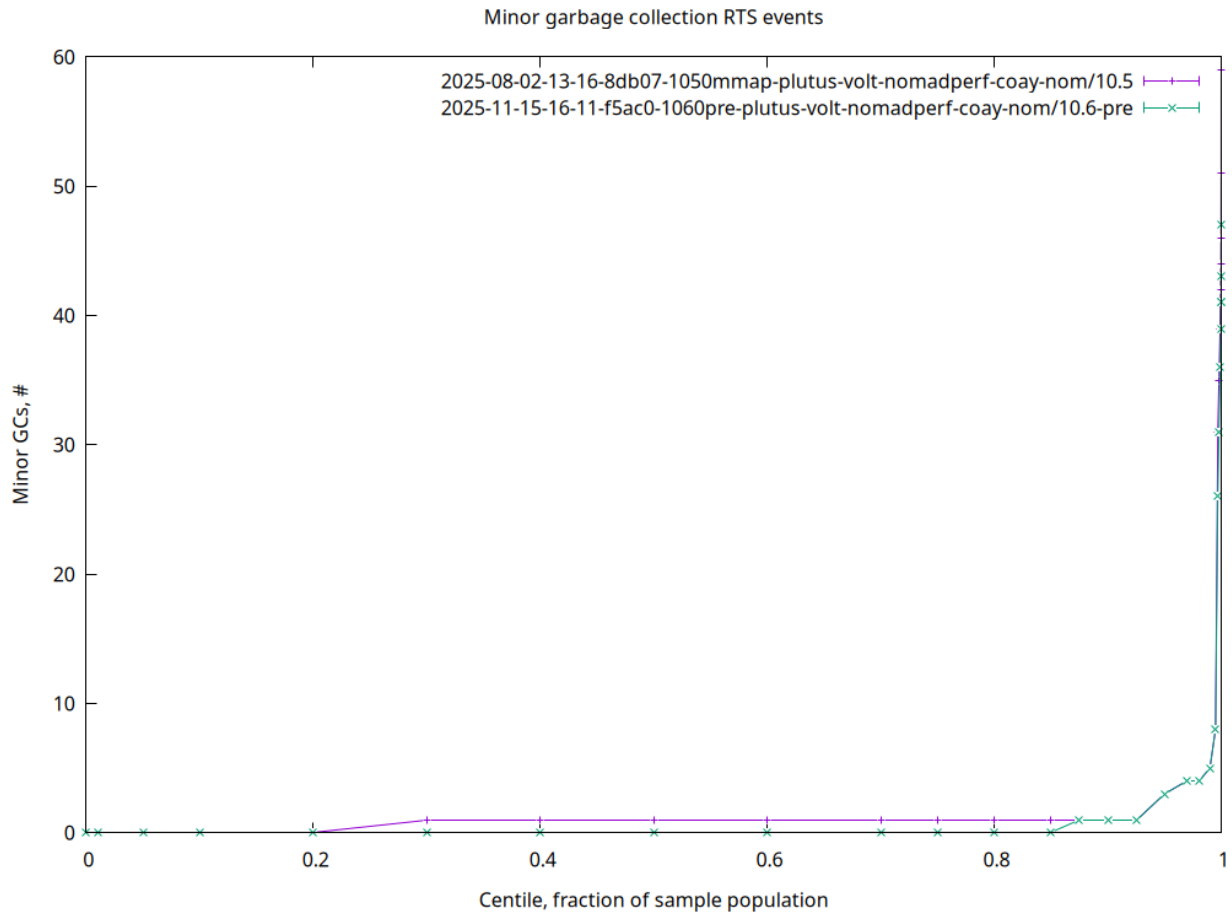
**RTS alloc rate (Alloc)** RTS-reported allocation rate, MB/sec



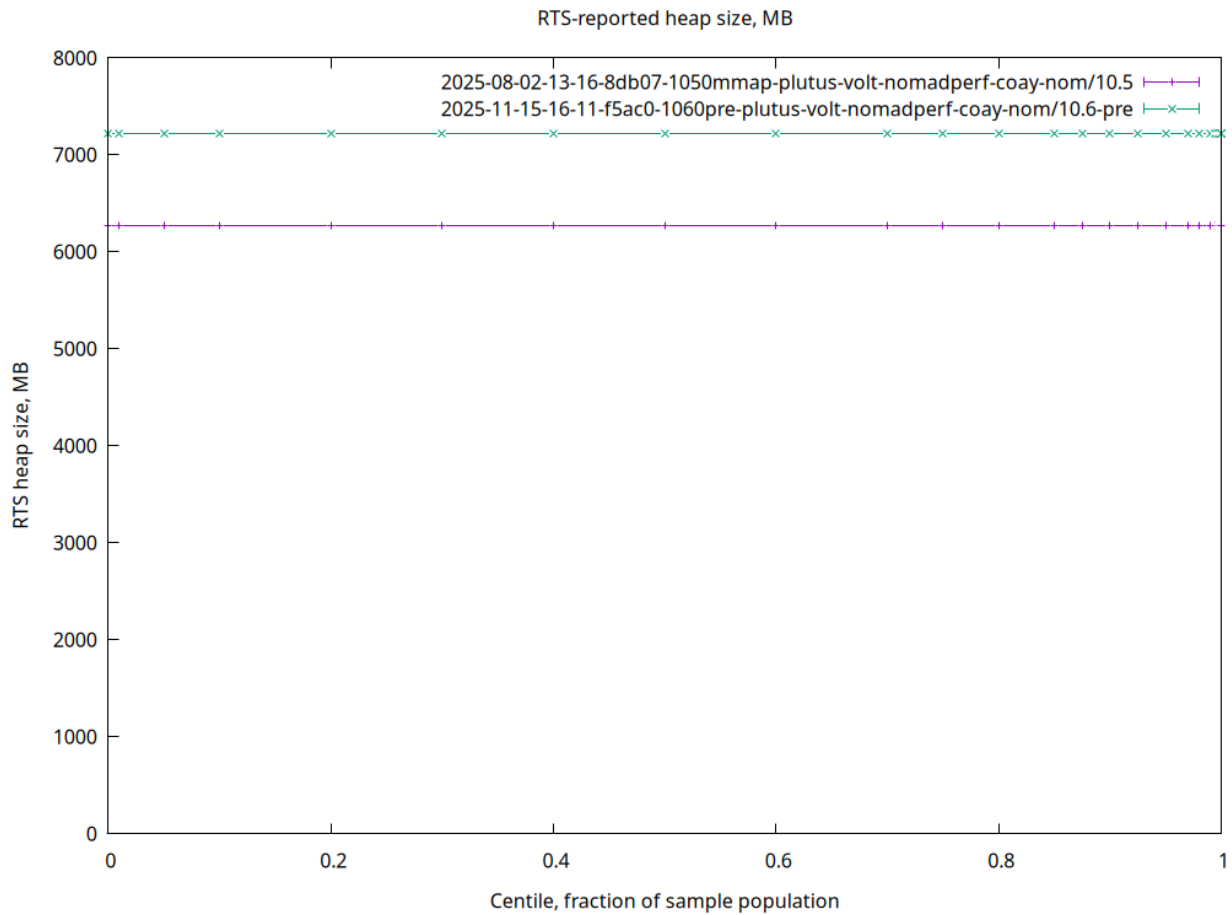
Major GCs (GcsMajor) Major garbage collection RTS events



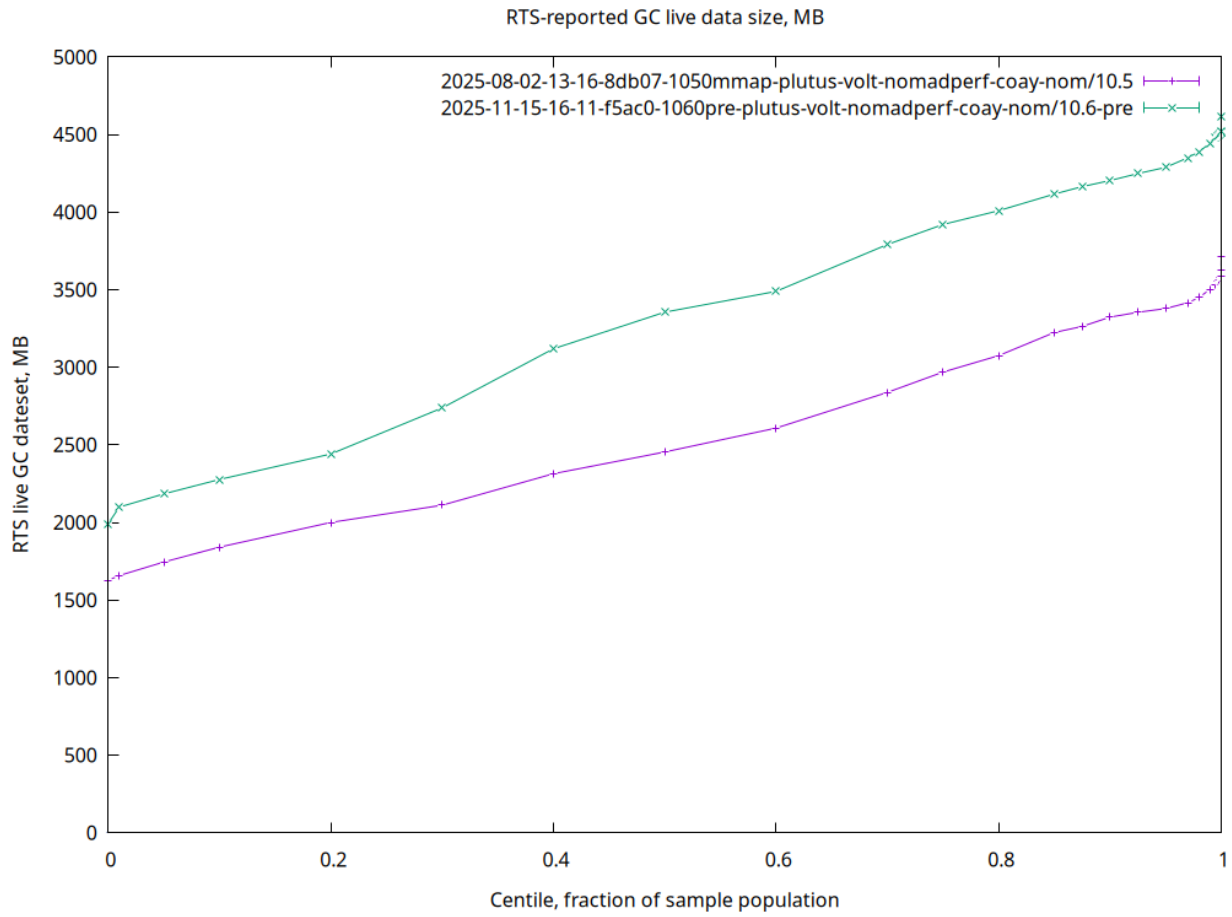
Minor GCs (GcsMinor) Minor garbage collection RTS events



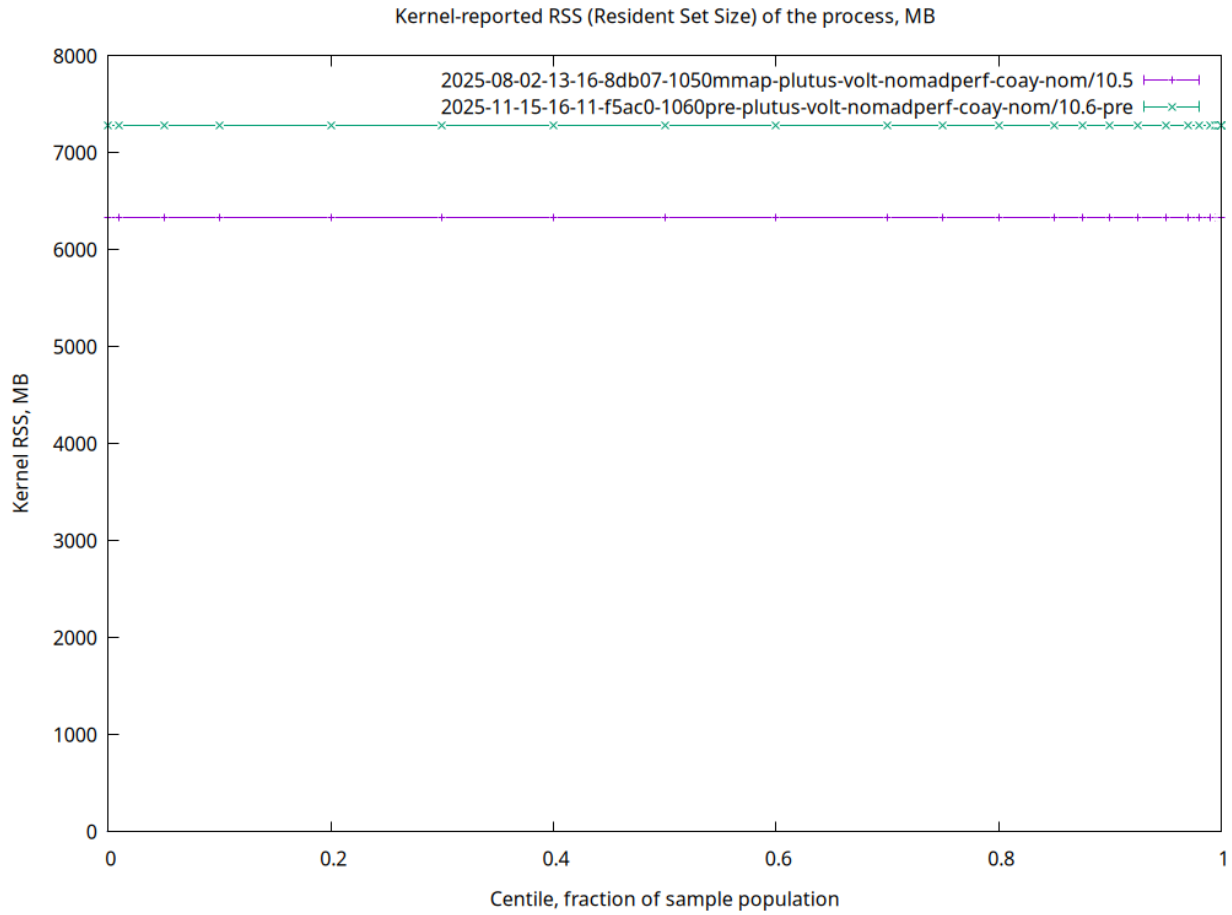
**RTS heap size (Heap)** RTS-reported heap size, MB



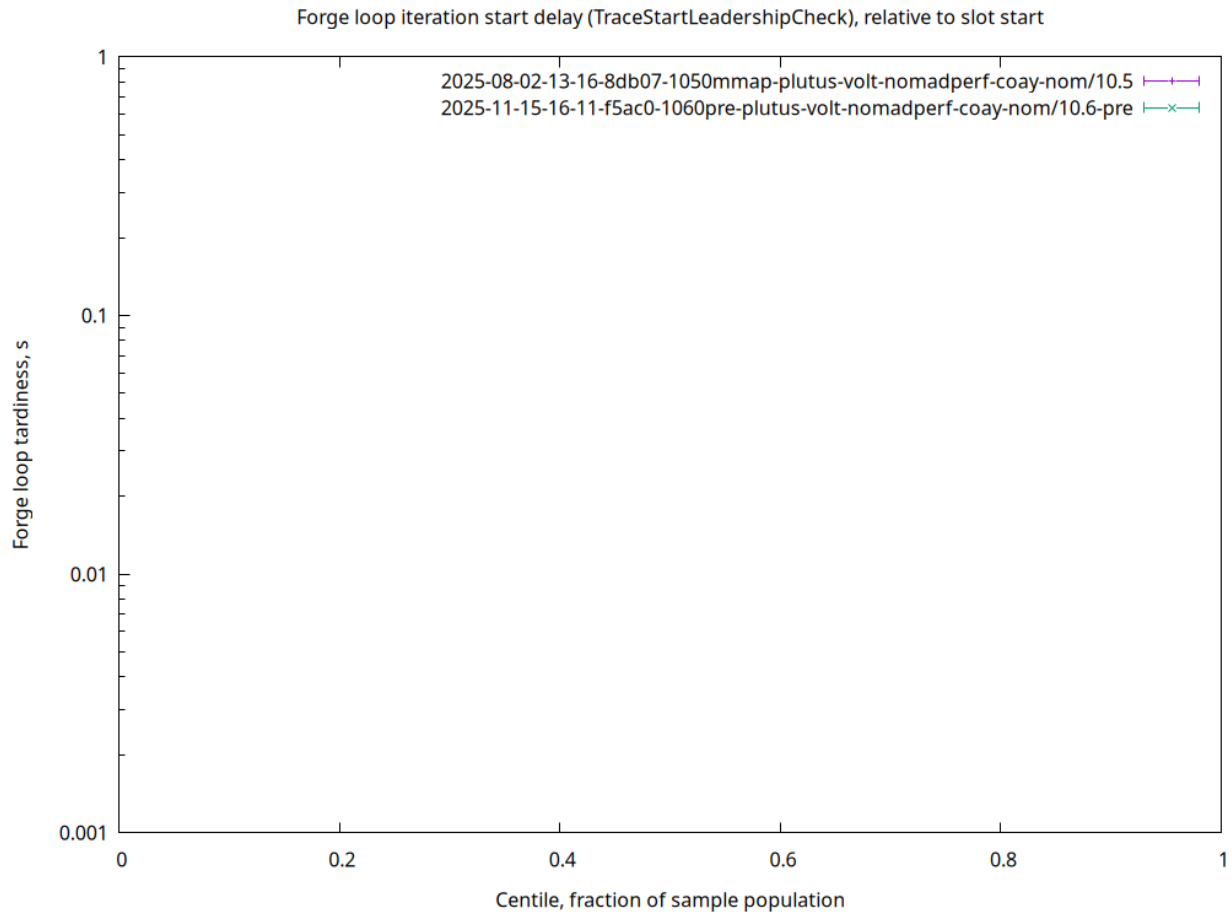
**RTS live GC dataset (Live)** RTS-reported GC live data size, MB



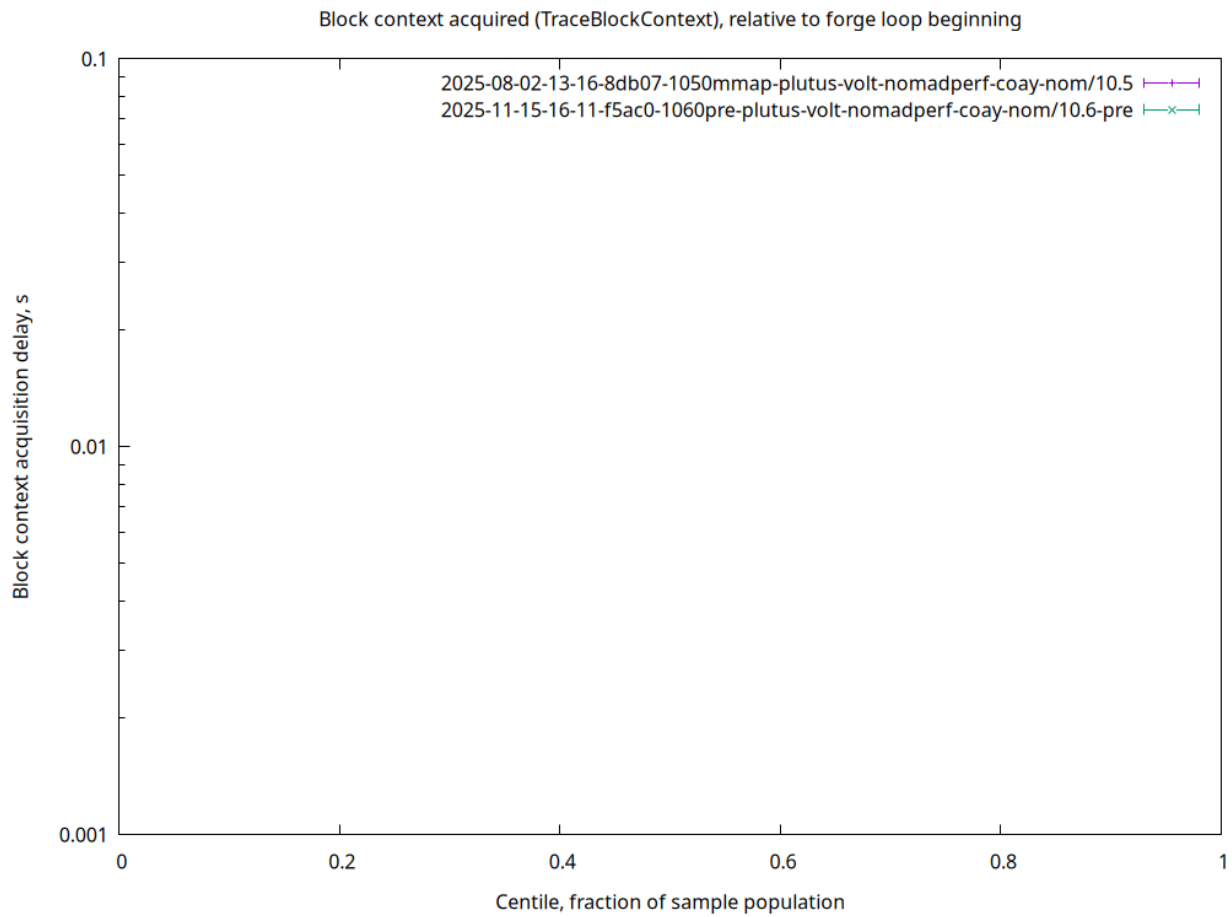
**Kernel RSS (RSS)** Kernel-reported RSS (Resident Set Size) of the process, MB



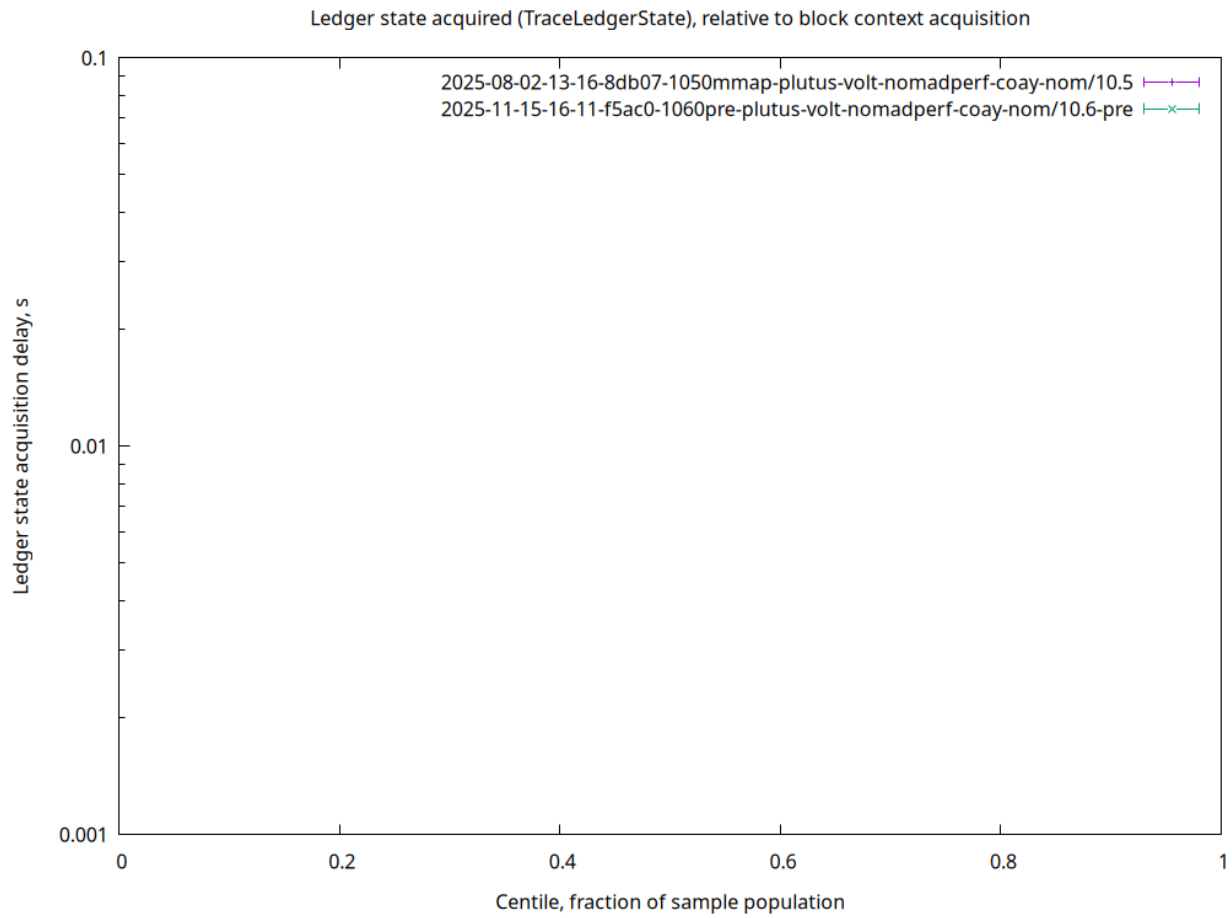
**Forge loop tardiness (cdfStarted)** Forge loop iteration start delay (TraceStartLeadershipCheck), relative to slot start



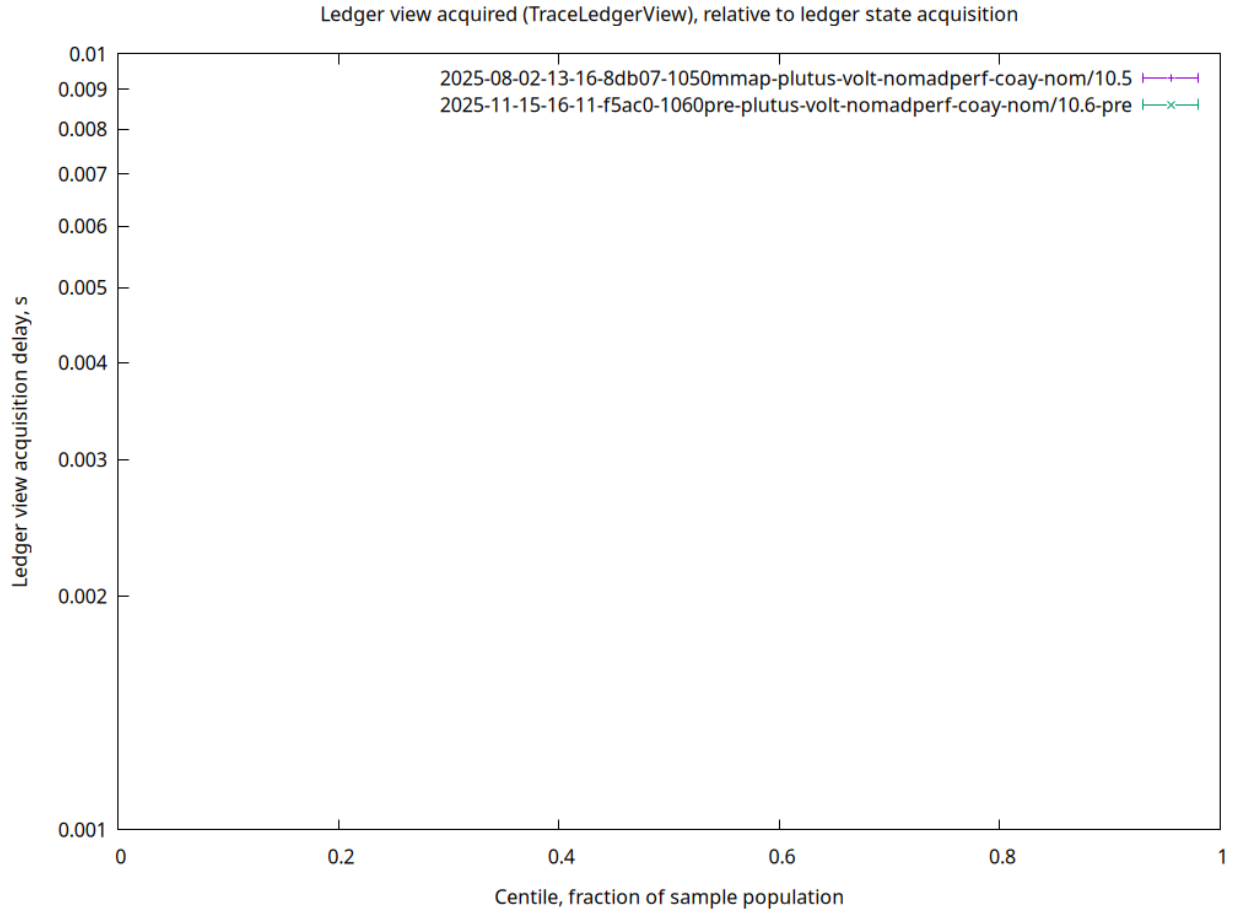
**Block context acquisition delay (cdfBlkCtx)** Block context acquired (TraceBlockContext), relative to forge loop beginning



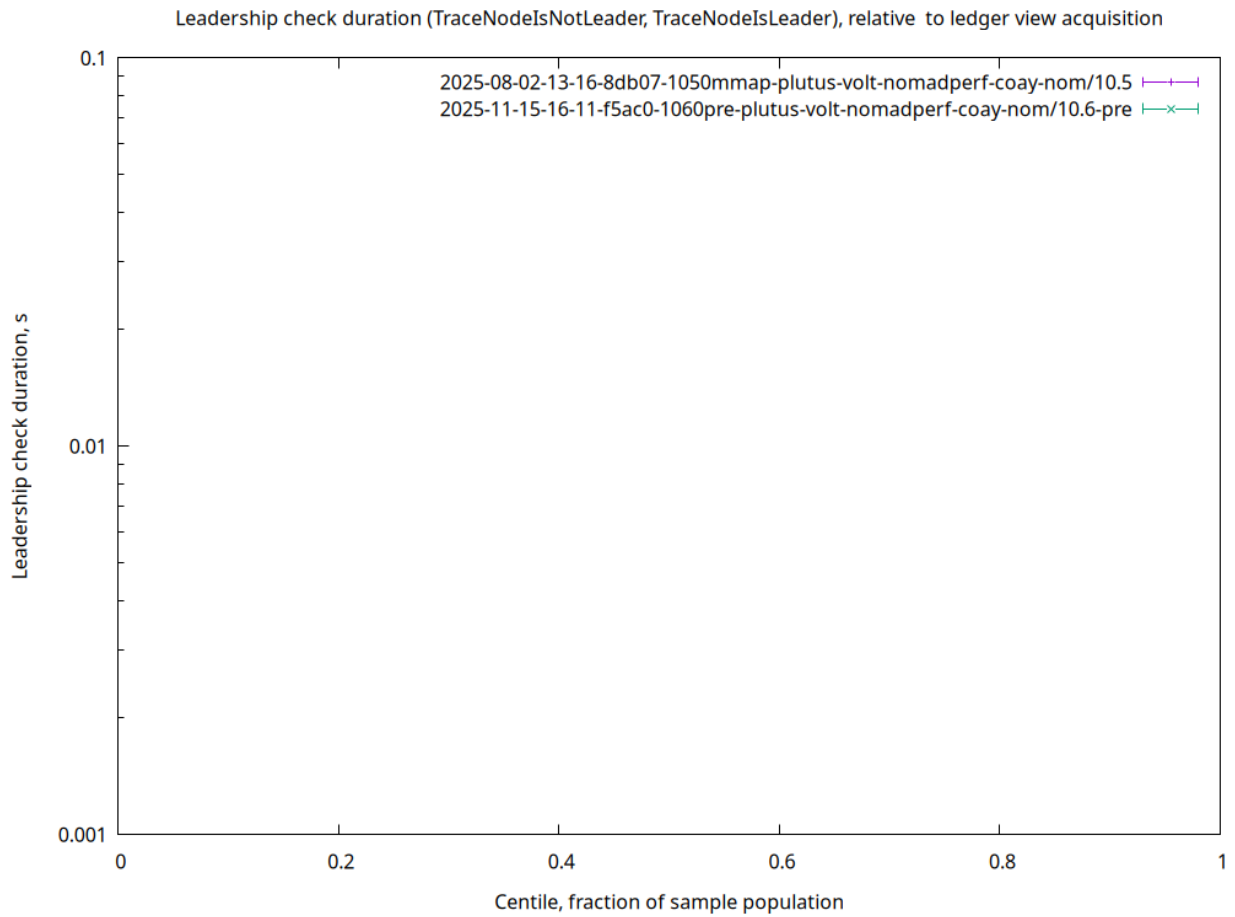
**Ledger state acquisition delay (cdfLgrState)** Ledger state acquired (TraceLedgerState), relative to block context acquisition



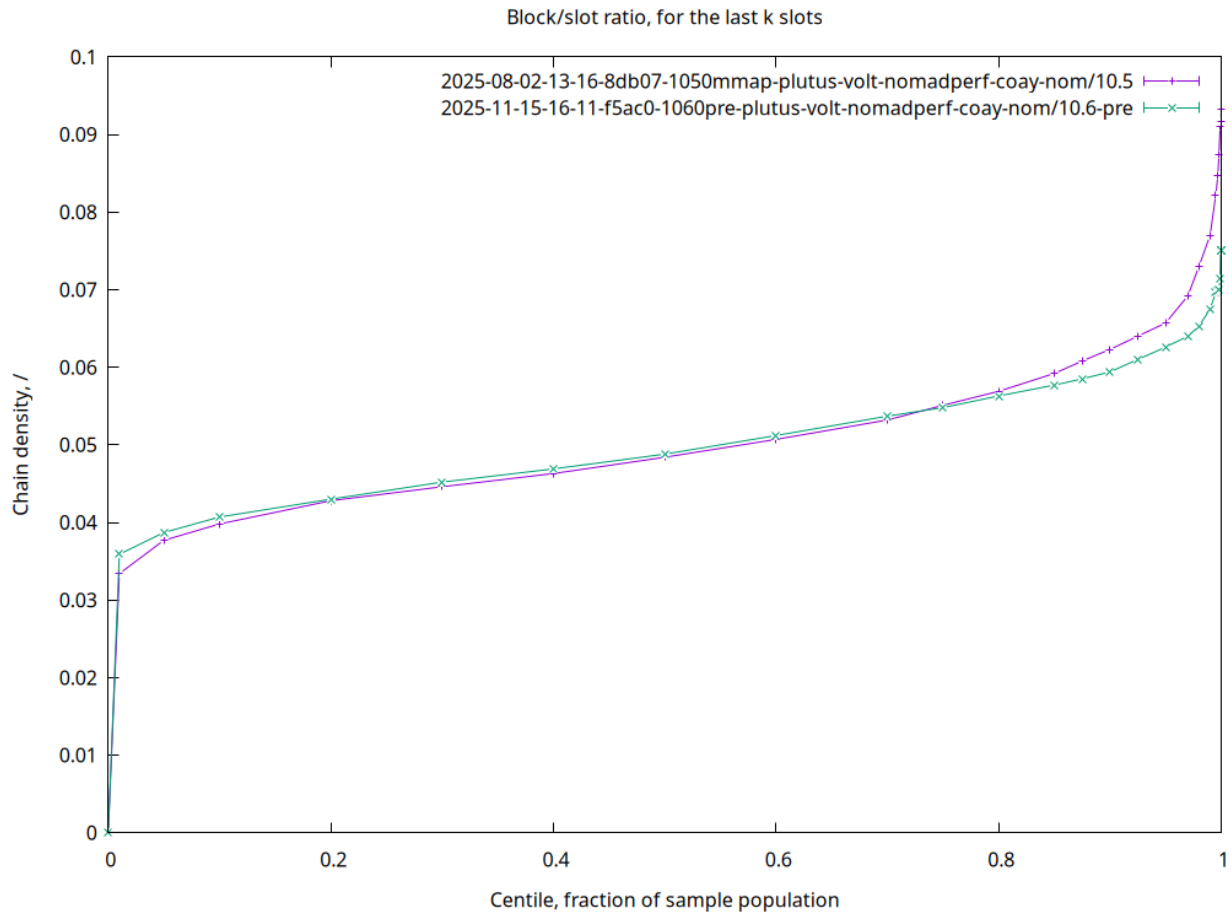
**Ledger view acquisition delay (cdfLgrView)** Ledger view acquired (TraceLedgerView), relative to ledger state acquisition



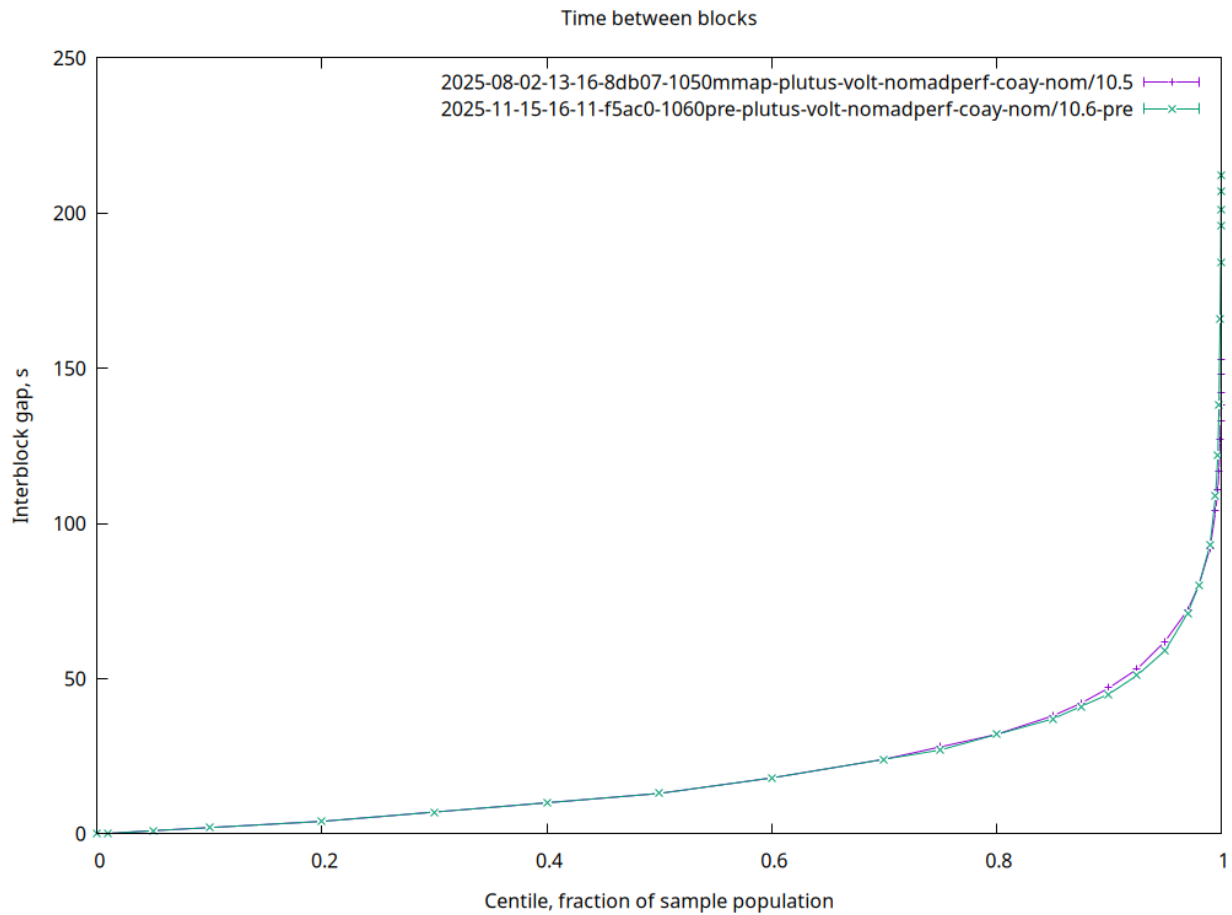
**Leadership check duration (cdfLeading)** Leadership check duration (TraceNodeIsNotLeader, TraceNodeIsLeader), relative to ledger view acquisition



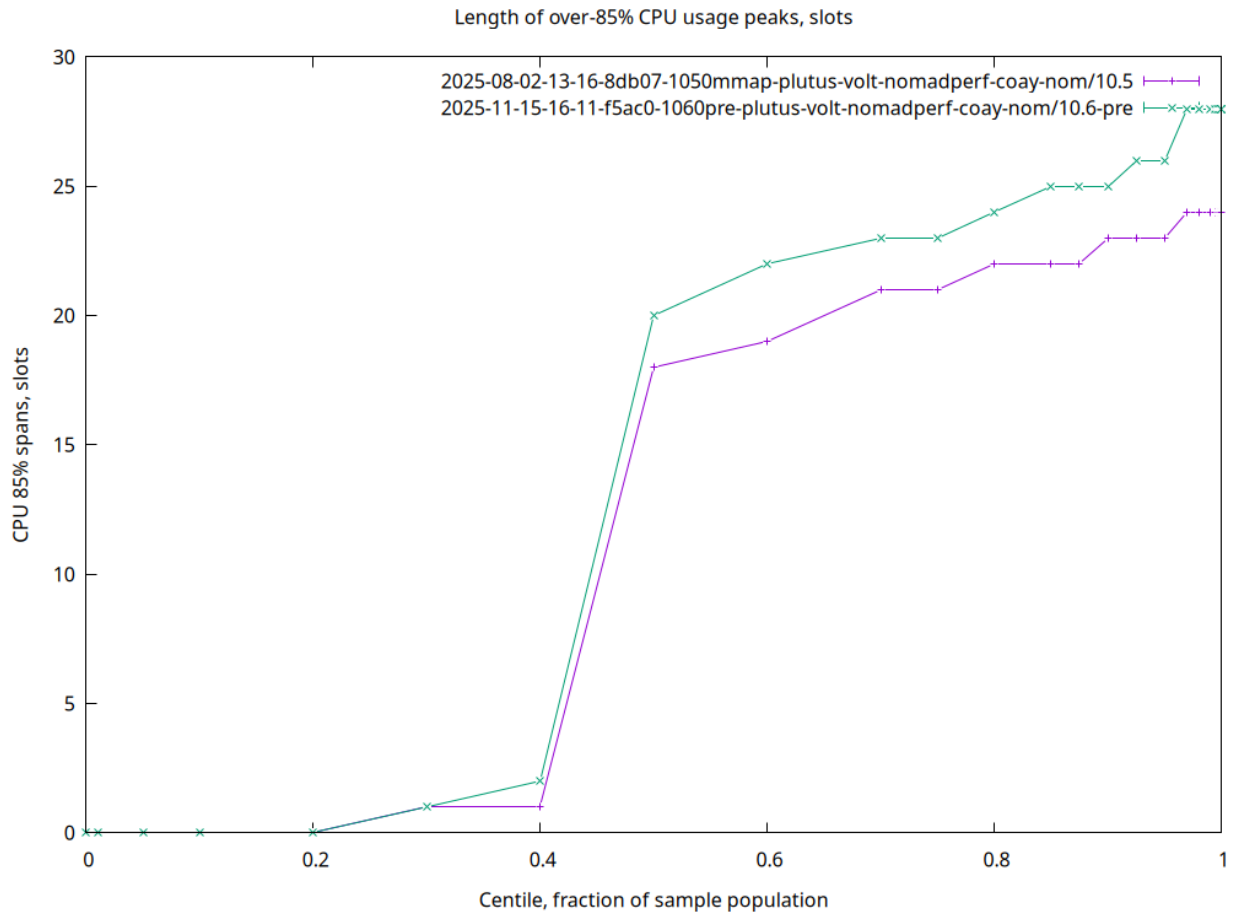
**Chain density (cdfDensity)** Block/slot ratio, for the last 'k' slots



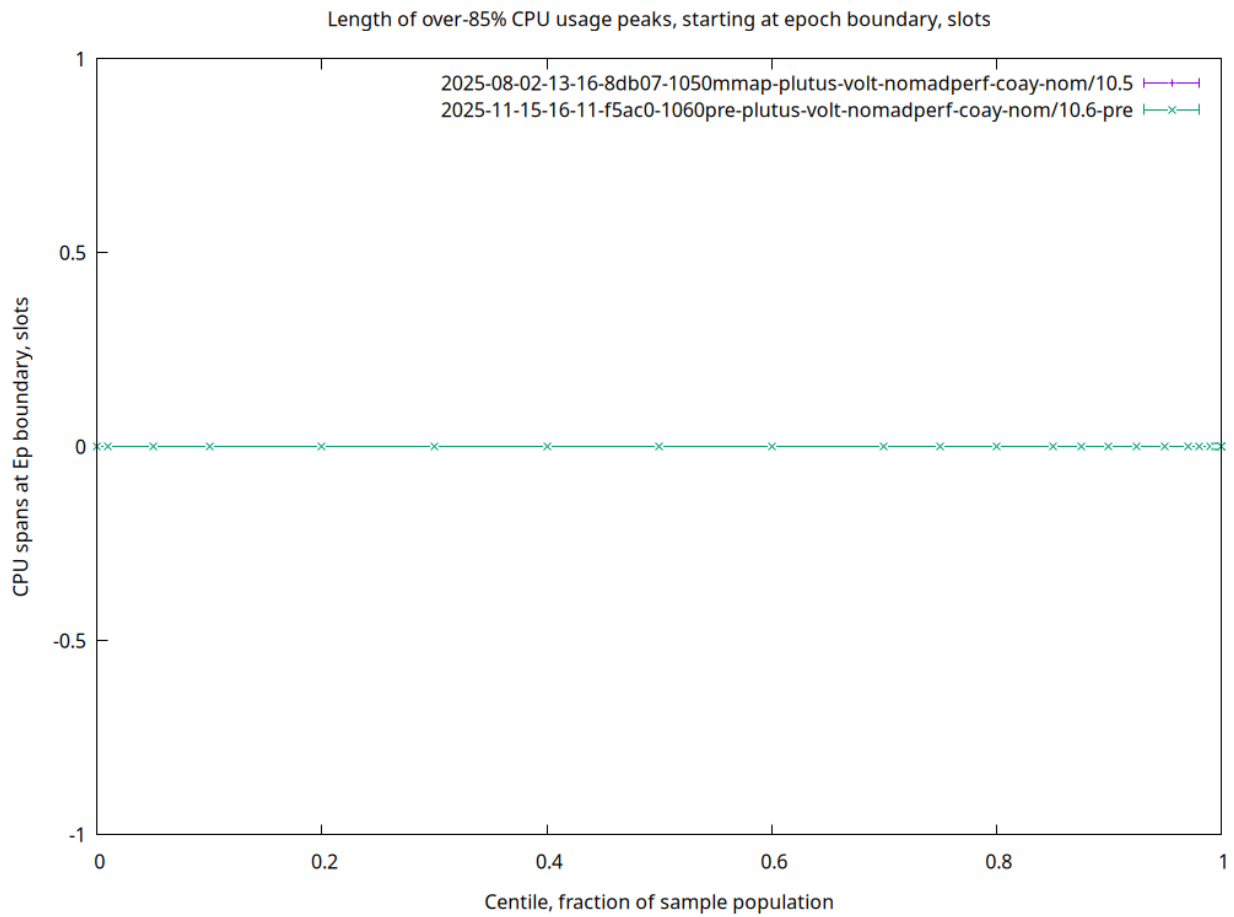
Interblock gap (cdfBlockGap) Time between blocks



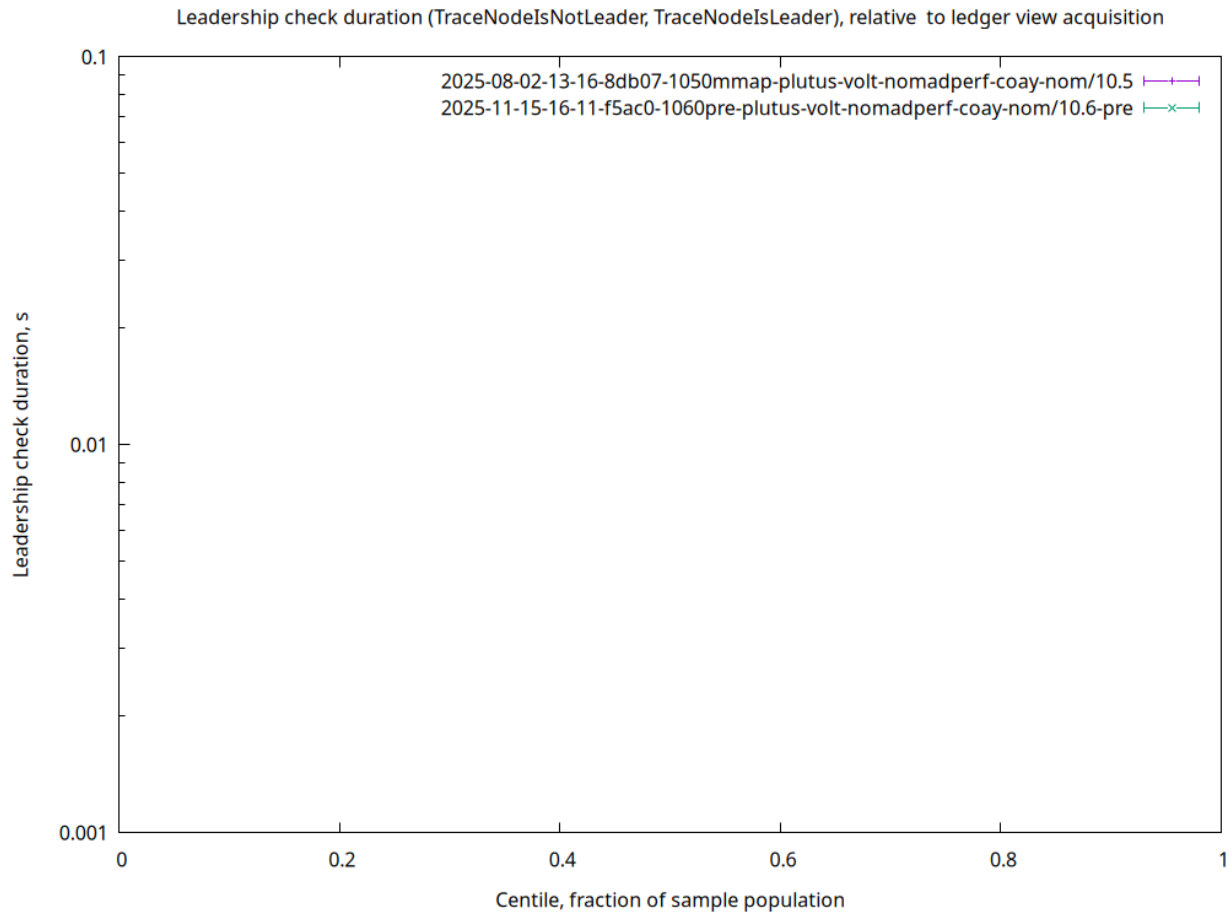
CPU 85% spans (cdfSpanLensCpu) Length of over-85% CPU usage peaks, slots



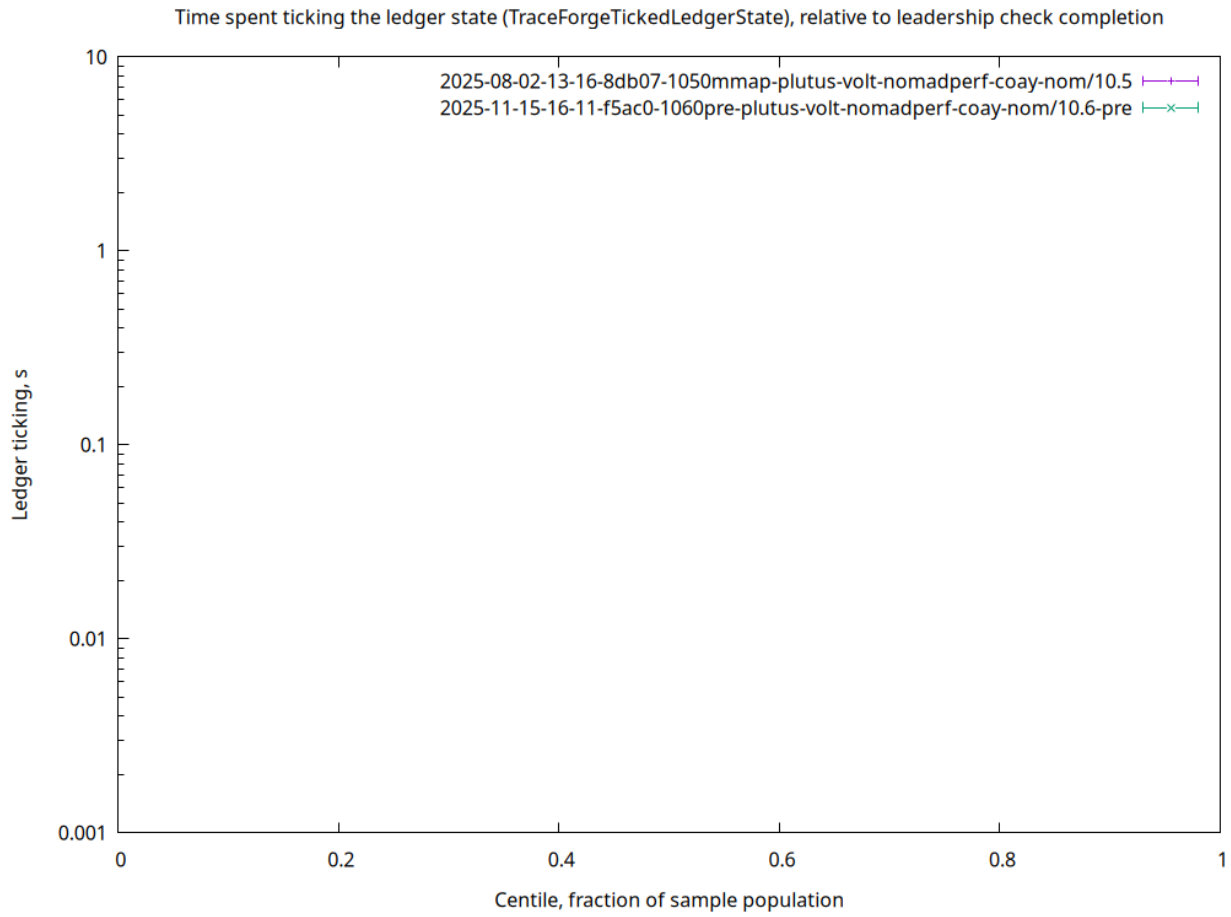
**CPU spans at Ep boundary (cdfSpanLensCpuEpoch)** Length of over-85% CPU usage peaks, starting at epoch boundary, slots



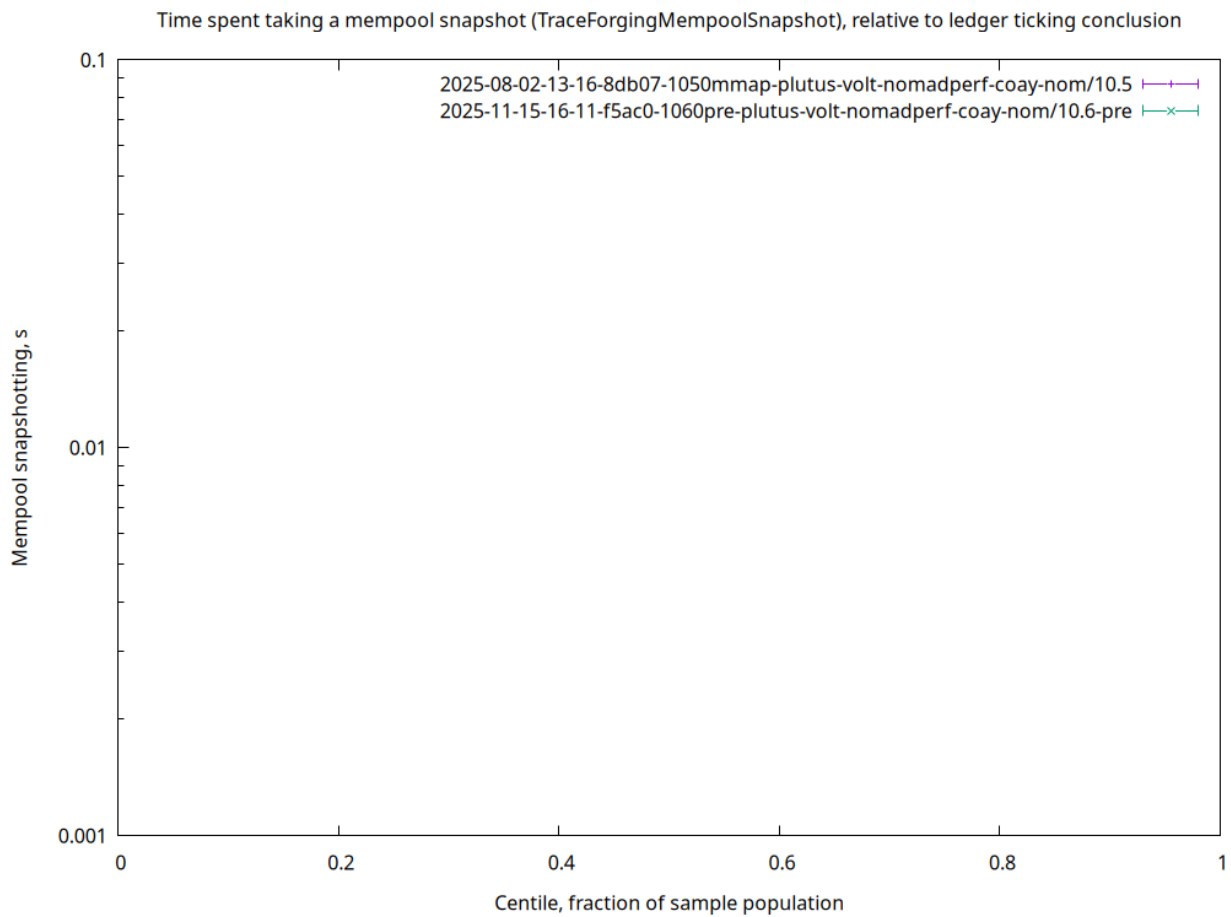
**Leadership check duration (cdfForgerLead)** Leadership check duration (TraceNodeIsNotLeader, TraceNodeIsLeader), relative to ledger view acquisition



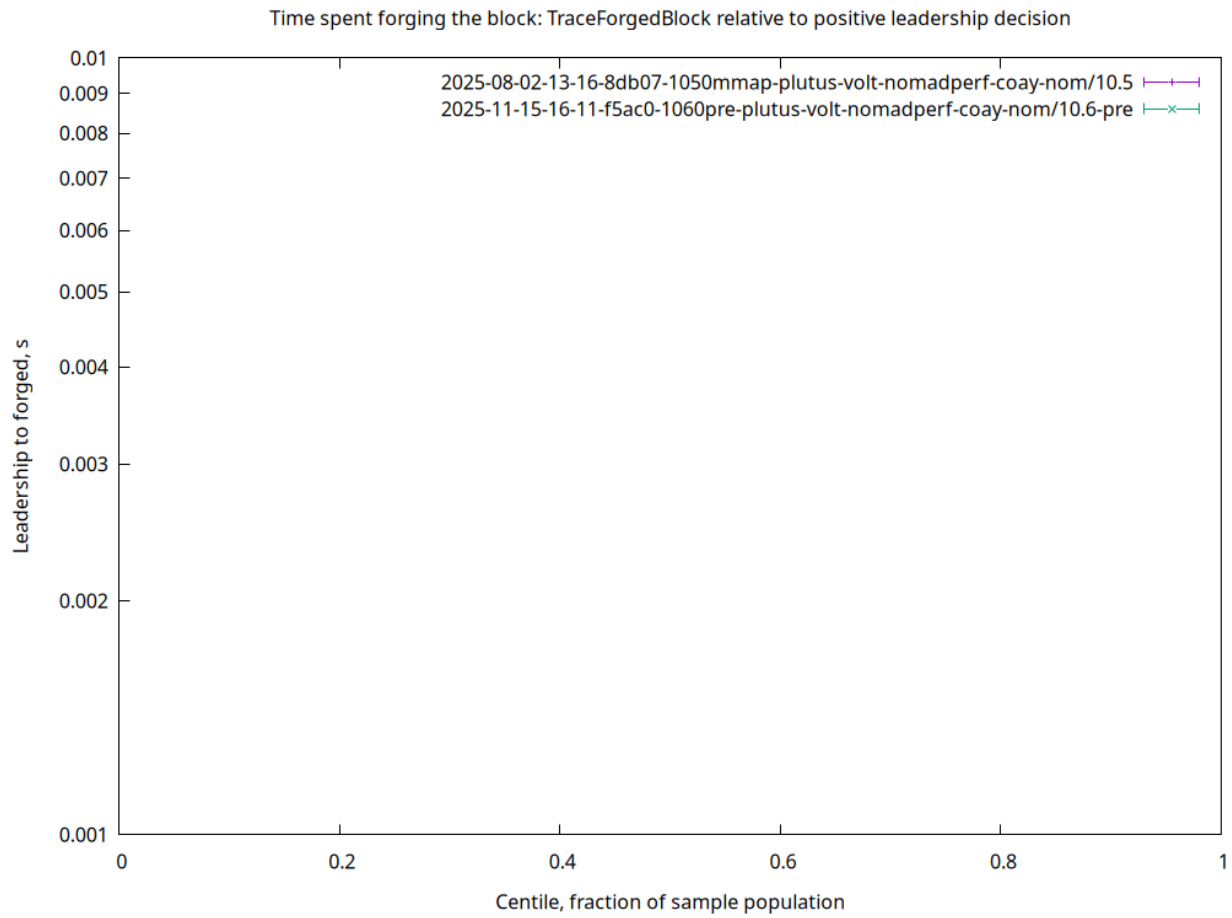
**Ledger ticking (cdfForgerTicked)** Time spent ticking the ledger state (TraceForgeTickedLedgerState), relative to leadership check completion



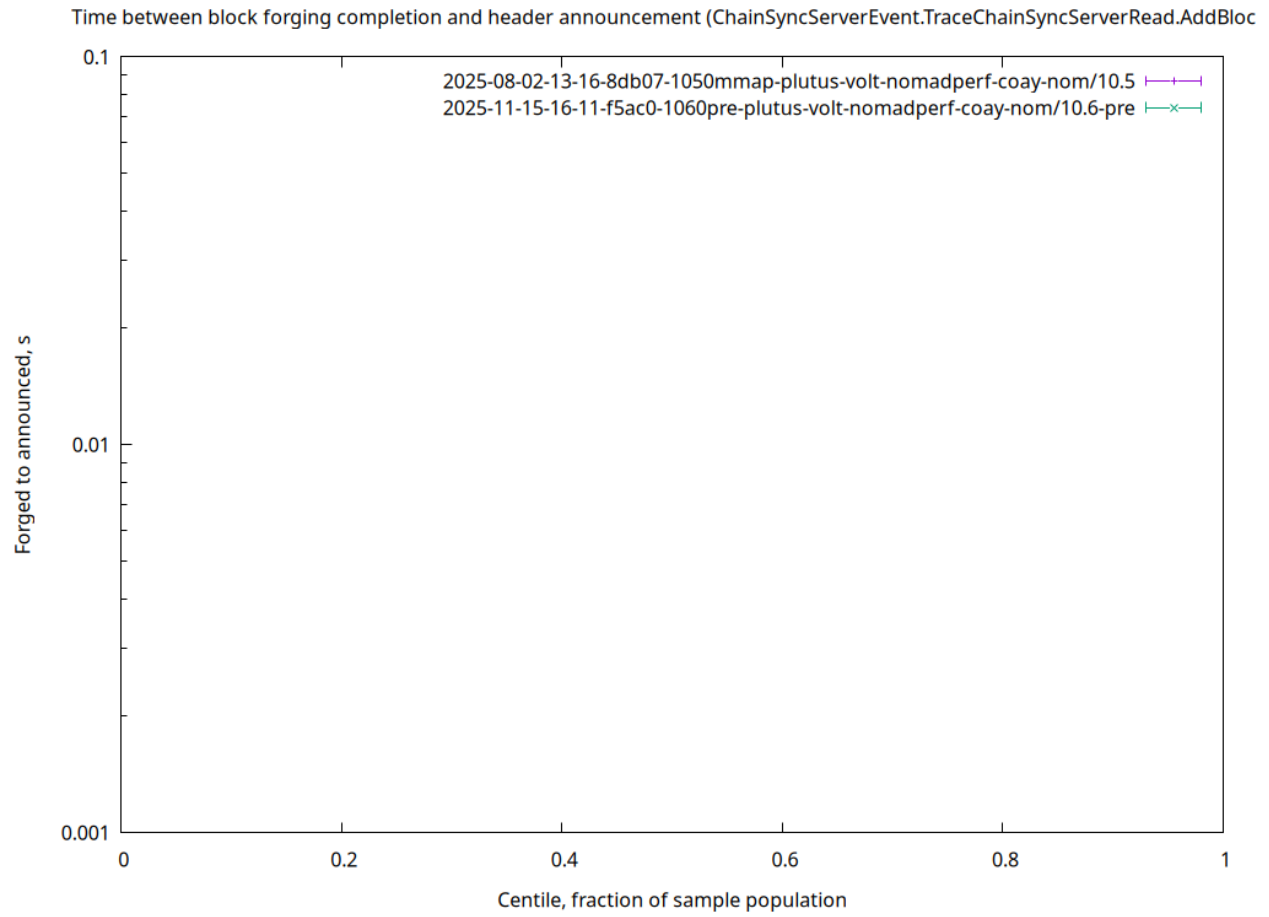
**Mempool snapshotting (cdfForgerMemSnap)** Time spent taking a mempool snapshot (TraceForgingMempool-Snapshot), relative to ledger ticking conclusion



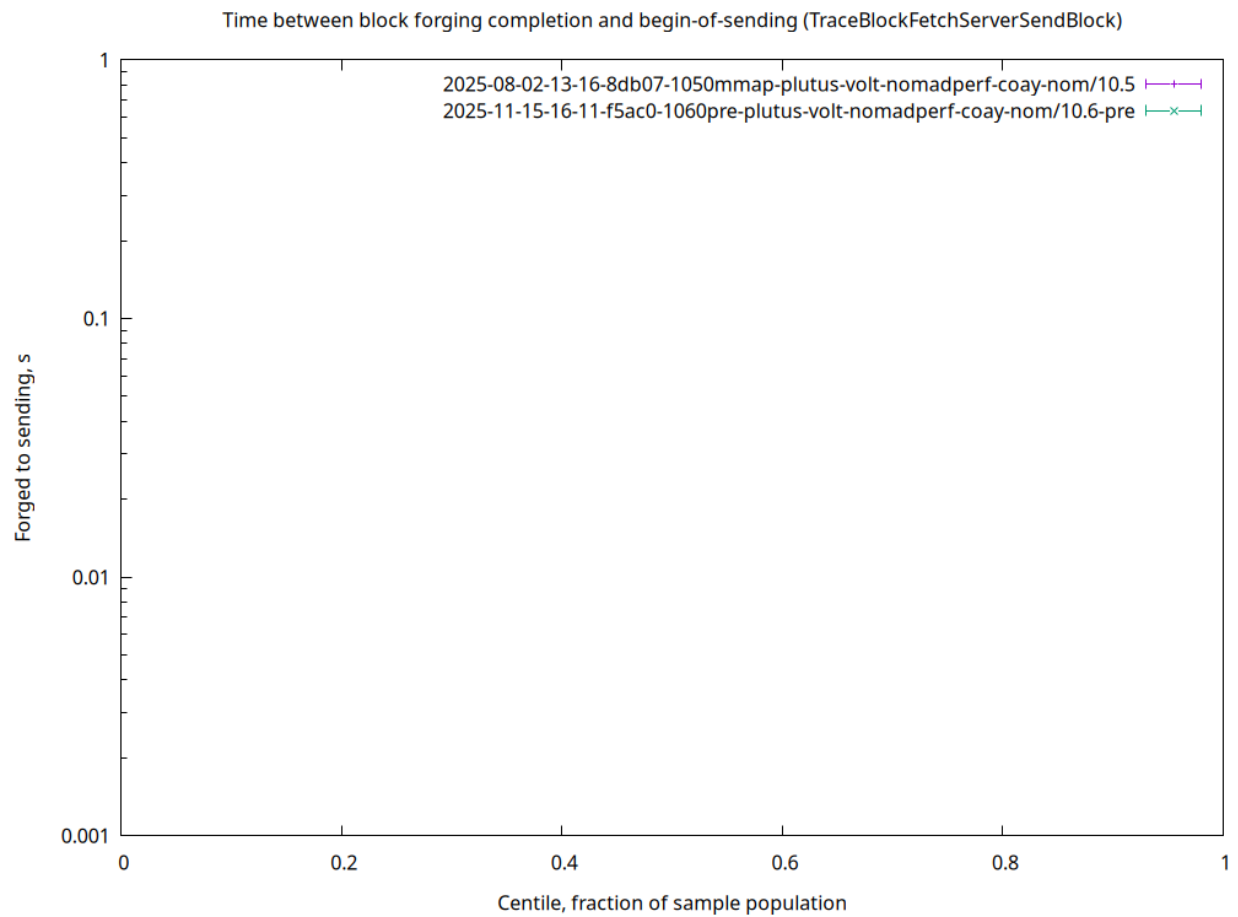
**Leadership to forged (cdfForgerForge)** Time spent forging the block: TraceForgedBlock relative to positive leadership decision



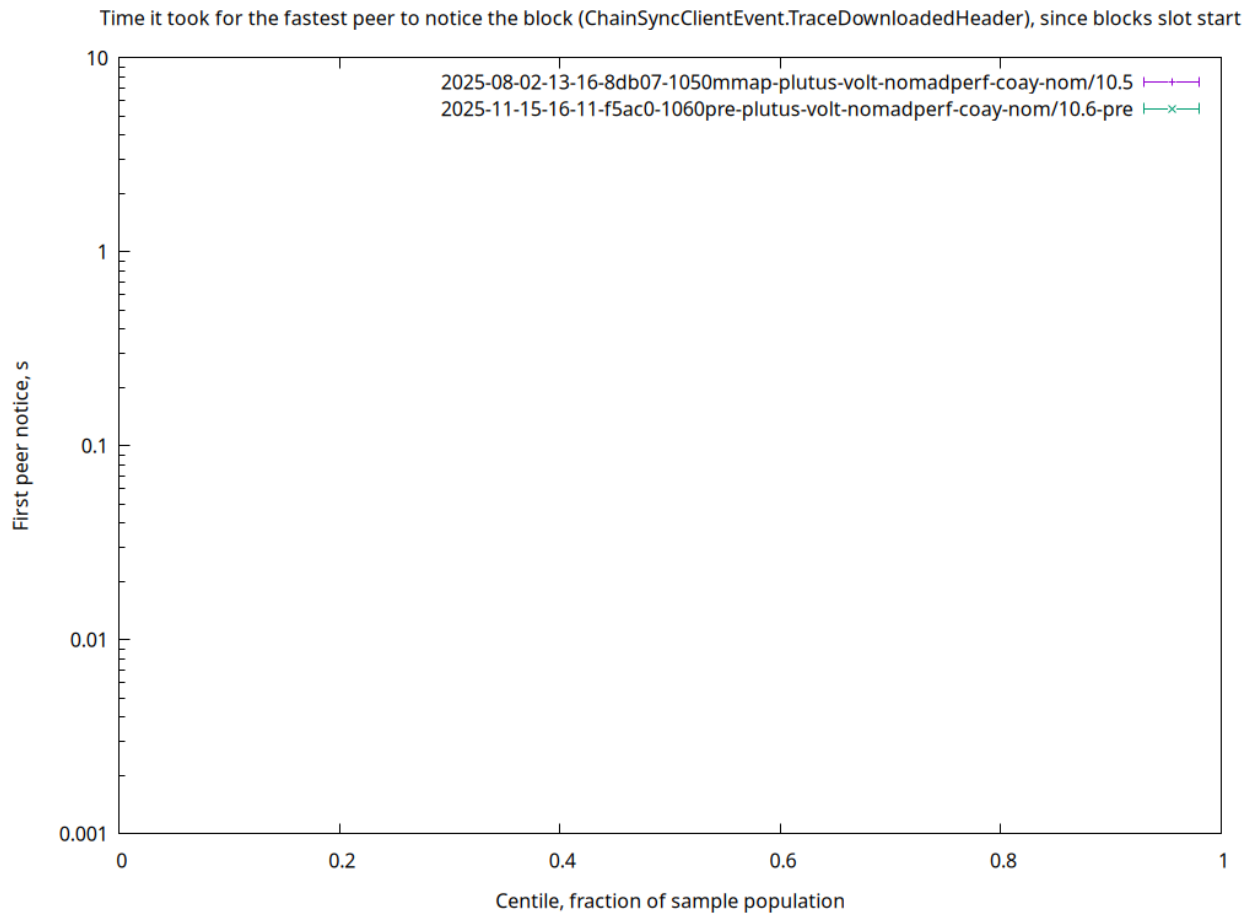
**Forged to announced (cdfForgerAnnounce)** Time between block forging completion and header announcement (ChainSyncServerEvent.TraceChainSyncServerRead.AddBlock)



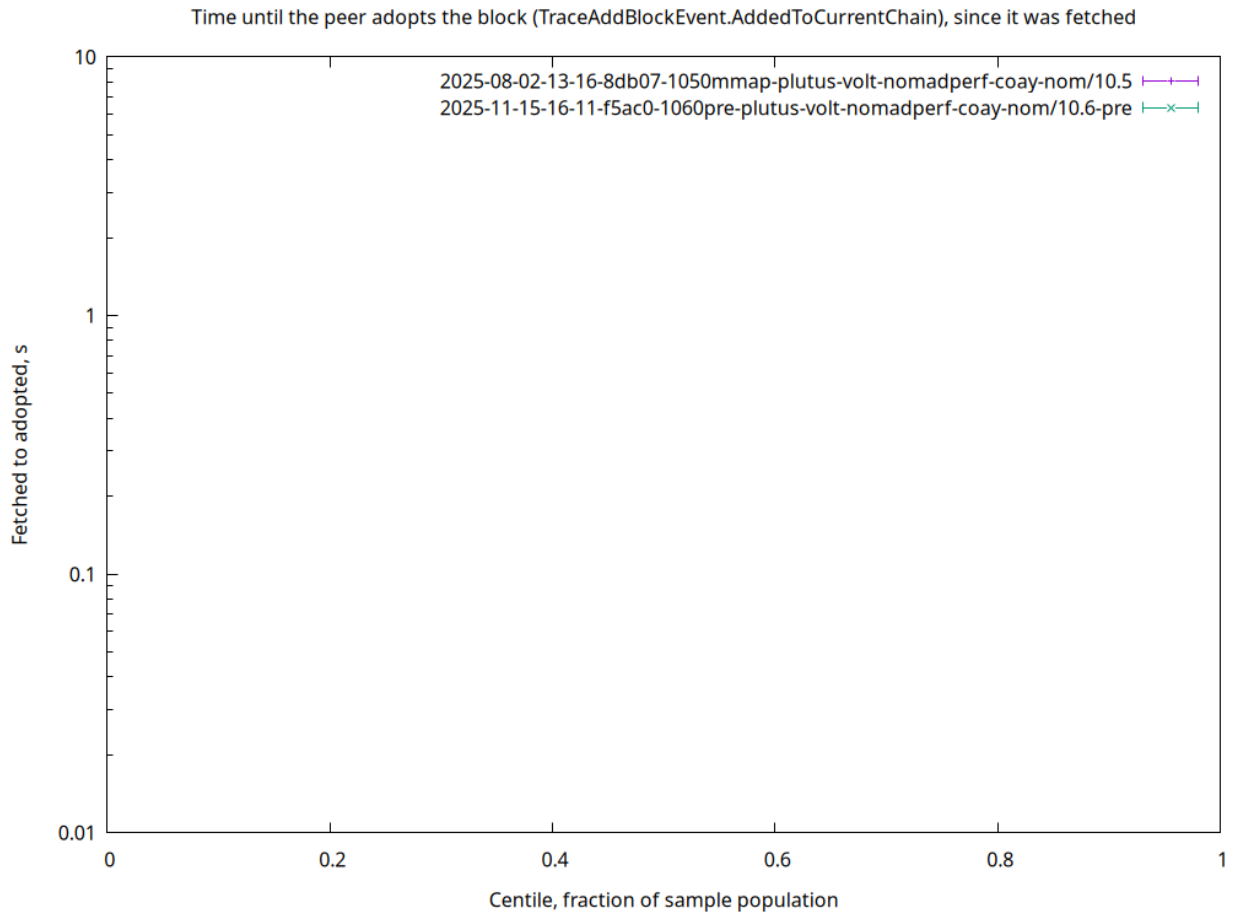
**Forged to sending (cdfForgerSend)** Time between block forging completion and begin-of-sending (TraceBlockFetchServerSendBlock)



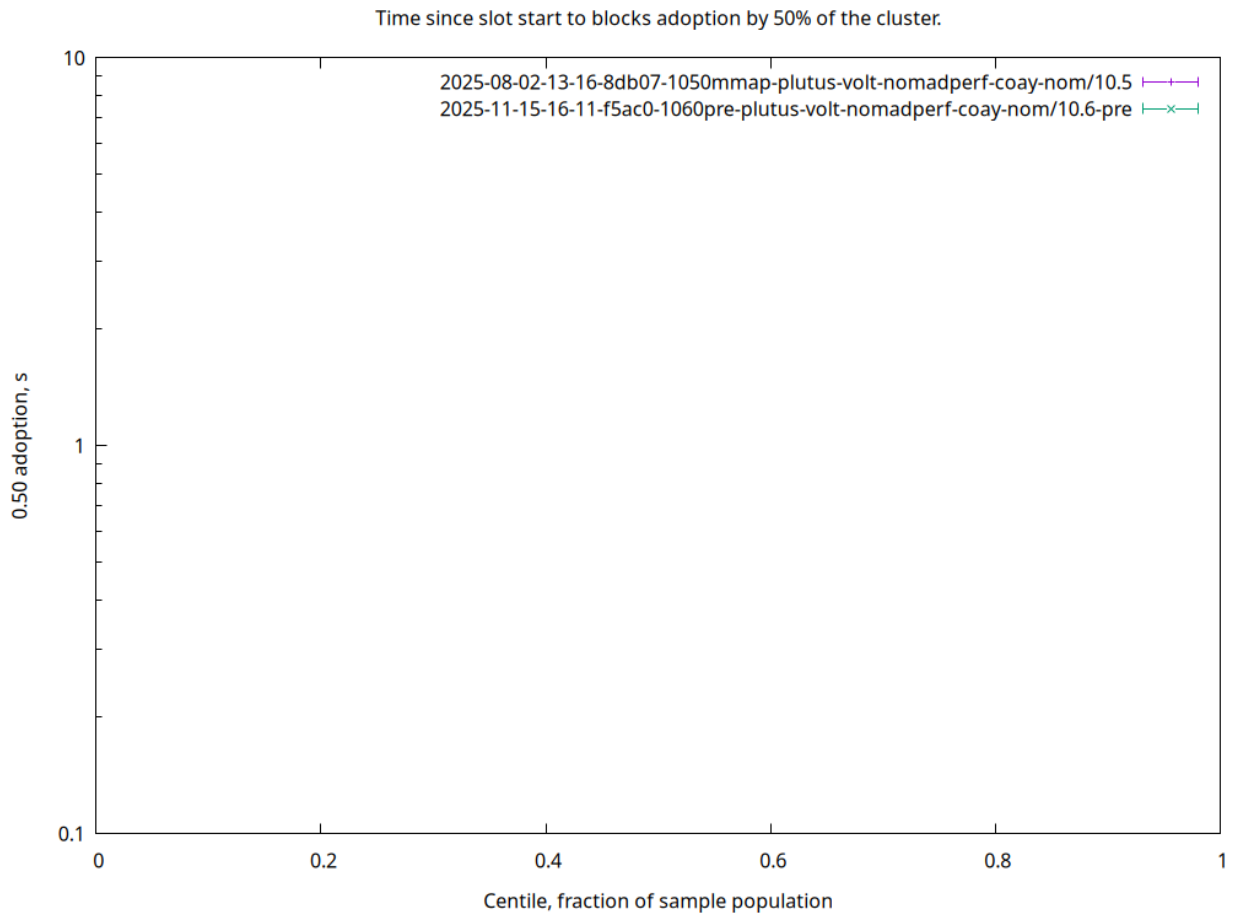
**First peer notice (cdfPeerNoticeFirst)** Time it took for the fastest peer to notice the block (ChainSyncClientEvent.TraceDownloadedHeader), since block's slot start



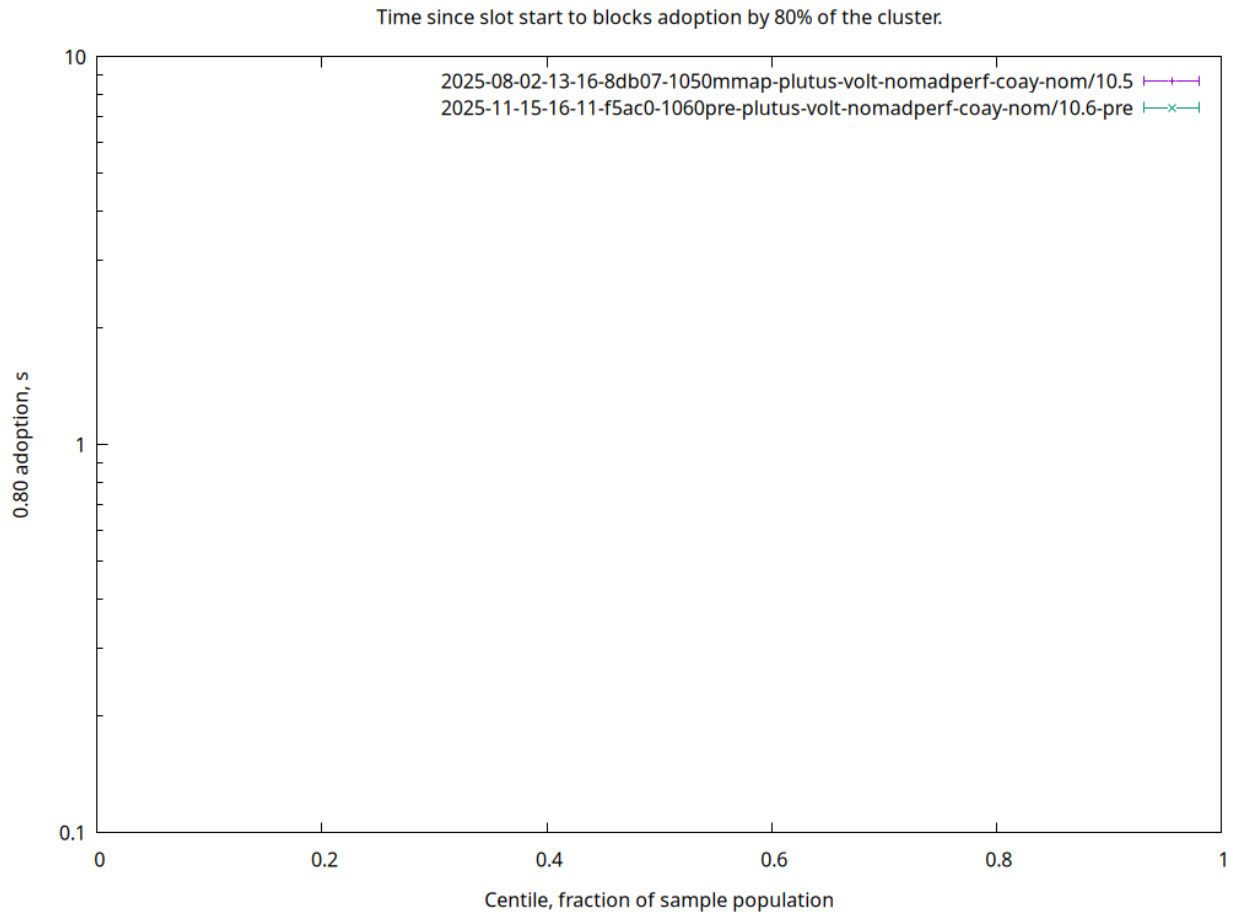
**Fetches to adopted (cdfPeerAdoption)** Time until the peer adopts the block (TraceAddBlockEvent.AddedToCurrentChain), since it was fetched



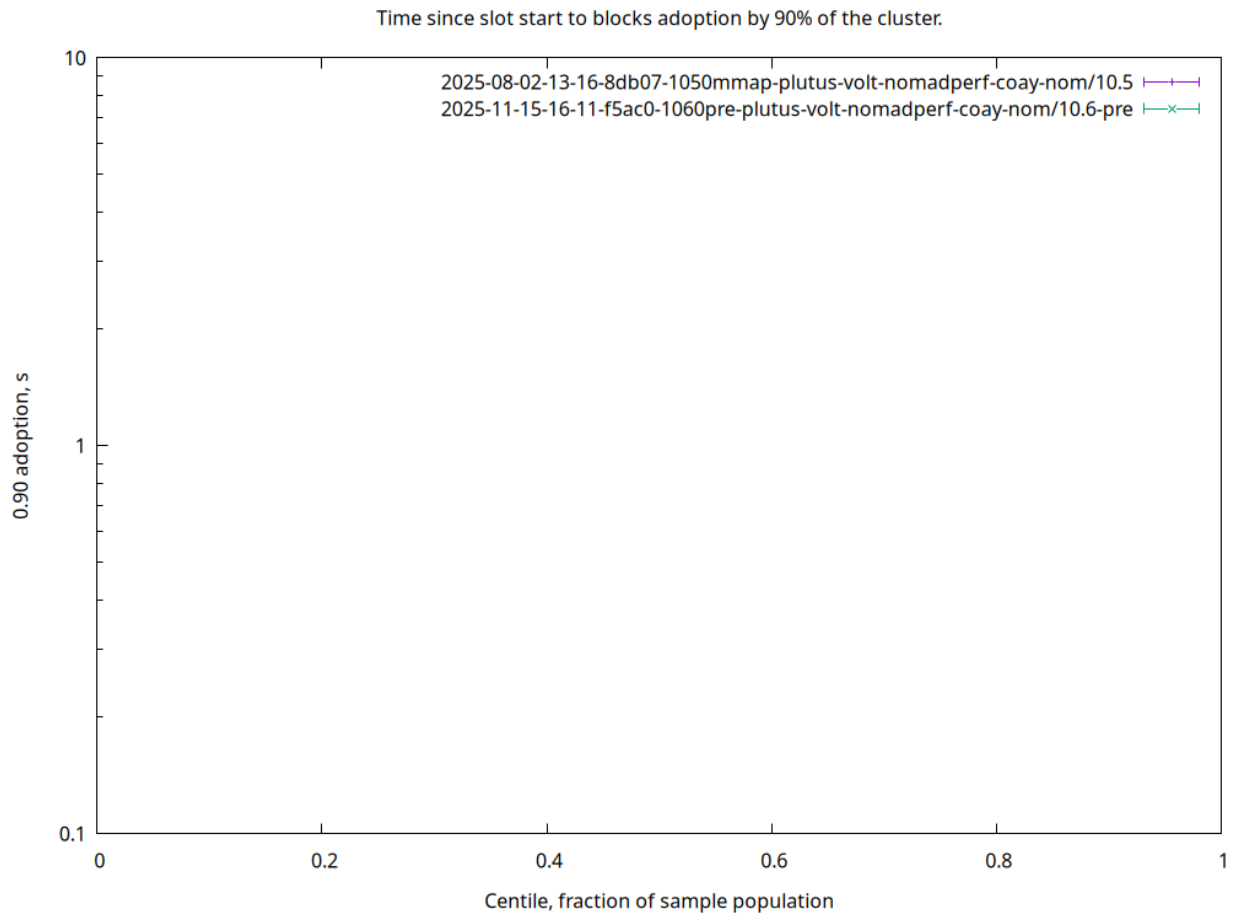
**0.50 adoption (cdf0.50)** Time since slot start to block's adoption by 50% of the cluster.



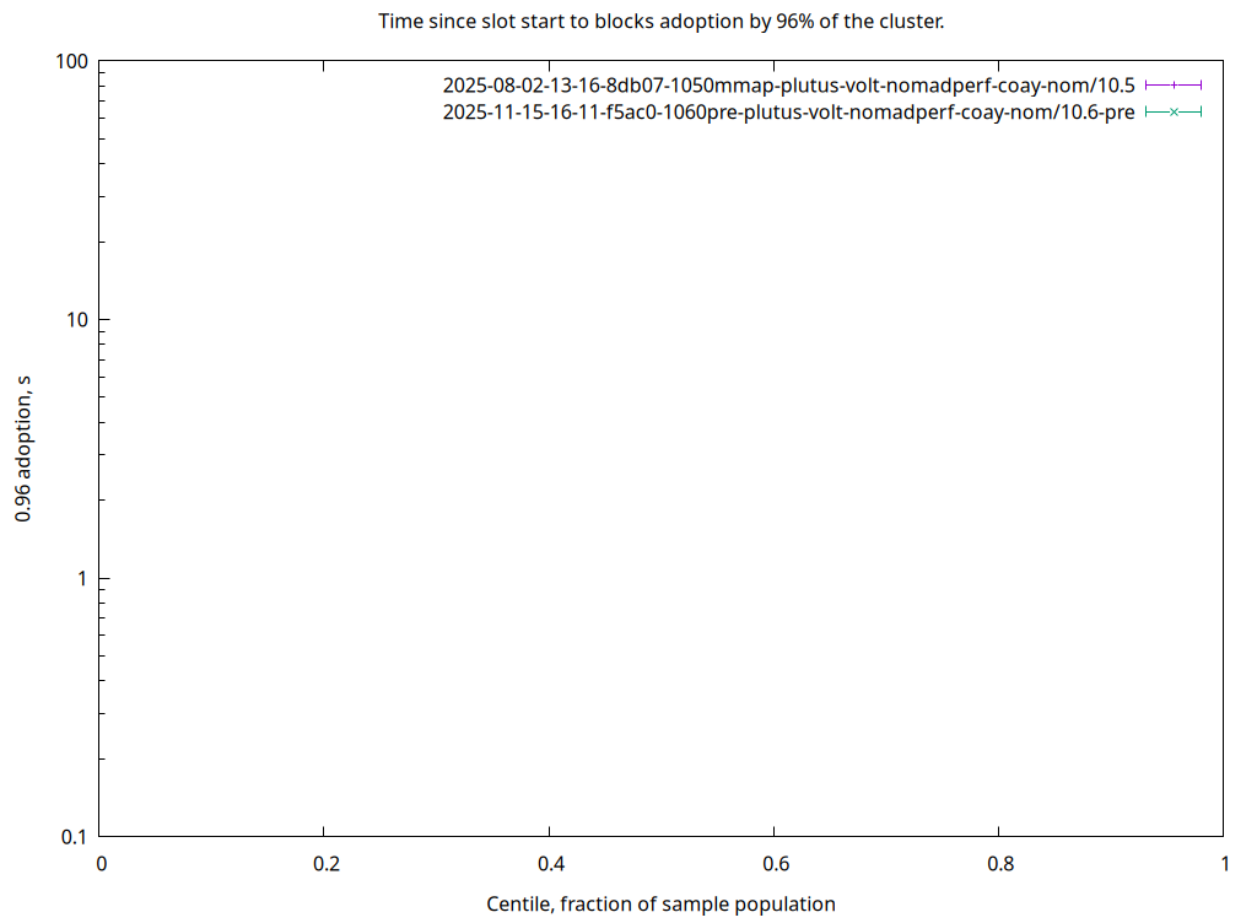
**0.80 adoption (cdf0.80)** Time since slot start to block's adoption by 80% of the cluster.



**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.



## 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 (TraceForgingMempool-Snapshot), relative to ledger ticking conclusion

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

**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