

10.5.0 against 10.4.1

Plutus countdown loop workload

Michael Karg, Cardano Performance team

2025-07-01

# 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 10.5.0 (Conway) relative to 10.4.1 (Conway), under Plutus countdown loop workload.

	10.4.1	10.5.0
Analysis date	2025-06-13	2025-06-29
Cluster system start date	2025-06-12	2025-06-28
Cluster system start time	09:20:10	19:41:44
Identifier	10.4.1	10.5.0
Run batch	10.4.1	1050fix
GHC version	9.6.5	9.6.5
cardano-node version	10.4.1	10.5.0
ouroboros-consensus version	0.26.0.1	0.27.0.0
ouroboros-network version	0.20.1.0	0.21.2.0
cardano-ledger-core version	1.17.0.0	1.17.0.0
plutus-core version	1.45.0.0	1.45.0.0
cardano-crypto version	1.3.0	1.3.0
cardano-prelude version	0.2.1.0	0.2.1.0
cardano-node git	9993fc5	28152ff
ouroboros-consensus git	9a2493b	8e3afe1
ouroboros-network git	d5d2042	879683d
cardano-ledger-core git	a9e78ae	a9e78ae
plutus-core git	ba16ec6	ba16ec6
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	832060.69230	909734.61538
Log objects analysed per host	538215.80769	611825.59615
Host run time, s	71853.2	71830.2
Host log line rate, Hz	11.580	12.665
Total log objects analysed	27987222	31814931
Run time, s	71857	71835
Analysed run duration, s	56011	56010
Run time efficiency	0.77	0.77
Node start spread, s	5.5197763	4.9408652
Node stop spread, s	2.6588120	4.7299718
Slots analysed	56009	56007
Blocks analysed	2688	2710
Blocks rejected	838	881

# Chapter 2

## Analysis

### 2.1 Resource Usage

	10.4.1	10.5.0	$\Delta$	$\Delta\%$
Forge loop starts, #	0.99885	0.99888	0.000	0
Process CPU usage, %	5.6055	3.8864	-1.719	-31
RTS GC CPU usage, %	0.25585	0.23499	-0.021	-8
RTS Mutator CPU usage, %	5.3409	3.6464	-1.695	-32
Major GCs, #	0.00078	0.00078	0.000	0
Minor GCs, #	1.3154	0.98284	-0.333	-25
Kernel RSS, MB	8289.3	8197.9	-91.400	-1
RTS heap size, MB	8228.3	8136.7	-91.600	-1
RTS live GC dataset, MB	3775.6	4074.5	298.900	8
RTS alloc rate, MB/s	40.376	28.726	-11.650	-29
Filesystem reads, KB/s	0.0	0.00013	0.000	nan
Filesystem writes, KB/s	228.87	227.39	-1.480	-1
CPU 85% spans, slots	10.138	10.397	0.259	3
Sample count	(291>)	(291>)		

### 2.2 Anomaly control

	10.4.1	10.5.0	$\Delta$	$\Delta\%$
Blocks per host, blocks	69.5	70.98	1.480	2
Filtered to chained block ratio, /	0.76302	0.75593	-0.007	-1
Chained to forged block ratio, /	0.97567	0.97269	-0.003	0
Height & slot battles, blocks	0.00186	0.00073	-0.001	-54
Block size, B	2996.0	2996.0	0.000	0
Sample count	(52)	(52)		

## 2.3 Forging

	10.4.1	10.5.0	$\Delta$	$\Delta\%$
Started forge loop iteration, s	0.00128	0.00138	0.000	0
Acquired block context, s	0.02415	6e-05	-0.024	-99
Acquired ledger state, s	9e-05	9e-05	0.000	0
Acquired ledger view, s	2e-05	3e-05	0.000	0
Leadership check duration, s	0.00038	0.00041	0.000	0
Ledger ticking, s	0.02308	0.02454	0.001	4
Mempool snapshotting, s	0.00155	0.00155	0.000	0
Leadership to forged, s	0.00016	0.00016	0.000	0
Forged to announced, s	0.00065	0.00066	0.000	0
Forged to sending, s	0.00514	0.00539	0.000	0
Forged to self-adopted, s	0.04461	0.04784	0.003	7
Slot start to announced, s	0.0514	0.02892	-0.022	-43
Sample count	(2688)	(2710)		

## 2.4 Individual peer propagation

	10.4.1	10.5.0	$\Delta$	$\Delta\%$
First peer notice, s	0.05316	0.03085	-0.022	-41
First peer fetch, s	0.05812	0.03607	-0.022	-38
Notice to fetch request, s	0.00115	0.0012	0.000	0
Fetch duration, s	0.12286	0.12217	-0.001	-1
Fetches to announced, s	0.00075	0.00082	0.000	0
Fetches to sending, s	0.04308	0.04334	0.000	0
Fetches to adopted, s	0.04583	0.04738	0.002	4
Sample count	(2688)	(2710)		

## 2.5 End-to-end propagation

	10.4.1	10.5.0	$\Delta$	$\Delta\%$
0.50 adoption, s	0.31138	0.28996	-0.021	-7
0.80 adoption, s	0.47315	0.45115	-0.022	-5
0.90 adoption, s	0.48454	0.4643	-0.020	-4
0.92 adoption, s	0.48812	0.46726	-0.021	-4
0.94 adoption, s	0.49142	0.46996	-0.021	-4
0.96 adoption, s	0.49475	0.47479	-0.020	-4
0.98 adoption, s	0.4998	0.48303	-0.017	-3
1.00 adoption, s	0.51956	0.50601	-0.014	-3
Sample count	(2688)	(2710)		

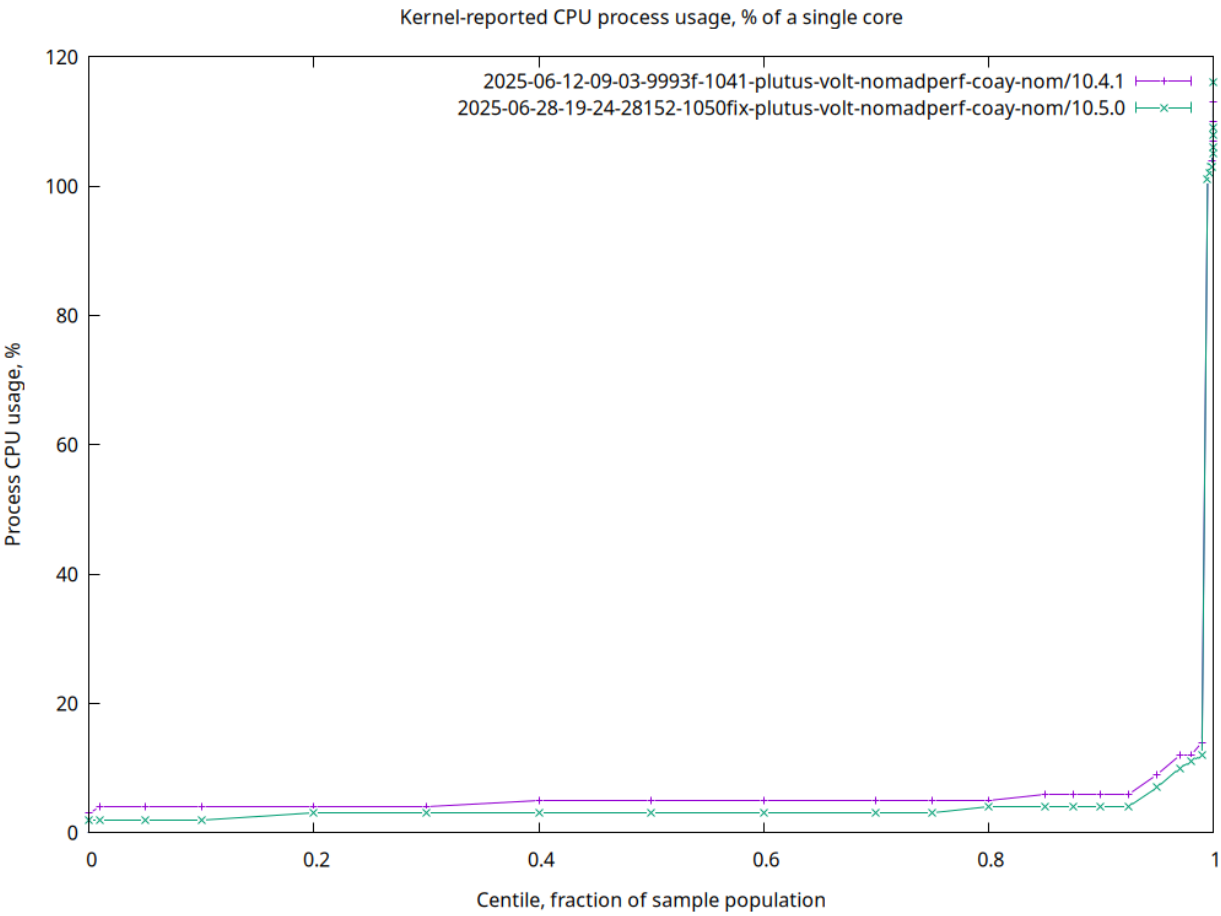
## Part I

### Appendix A: charts

# Chapter 3

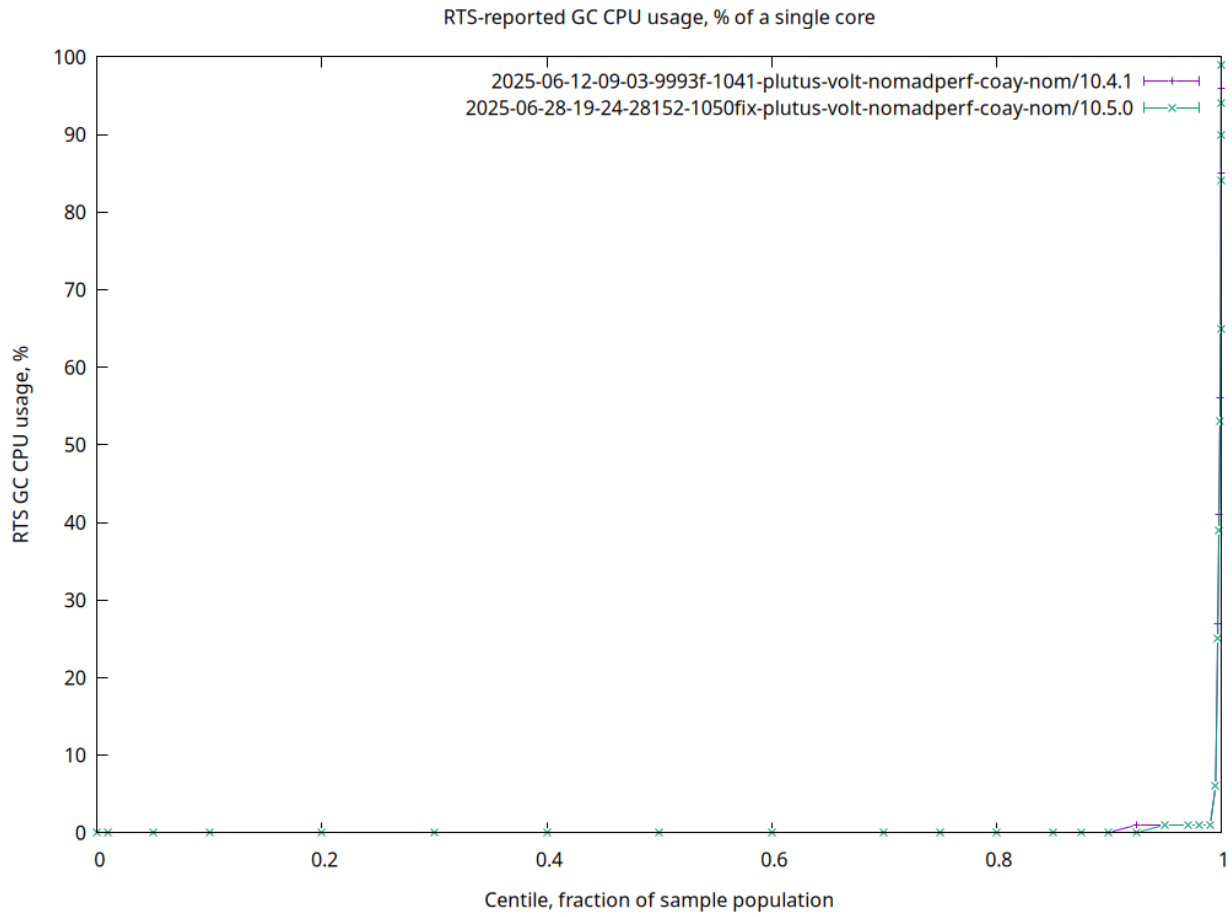
## Cluster performance charts

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

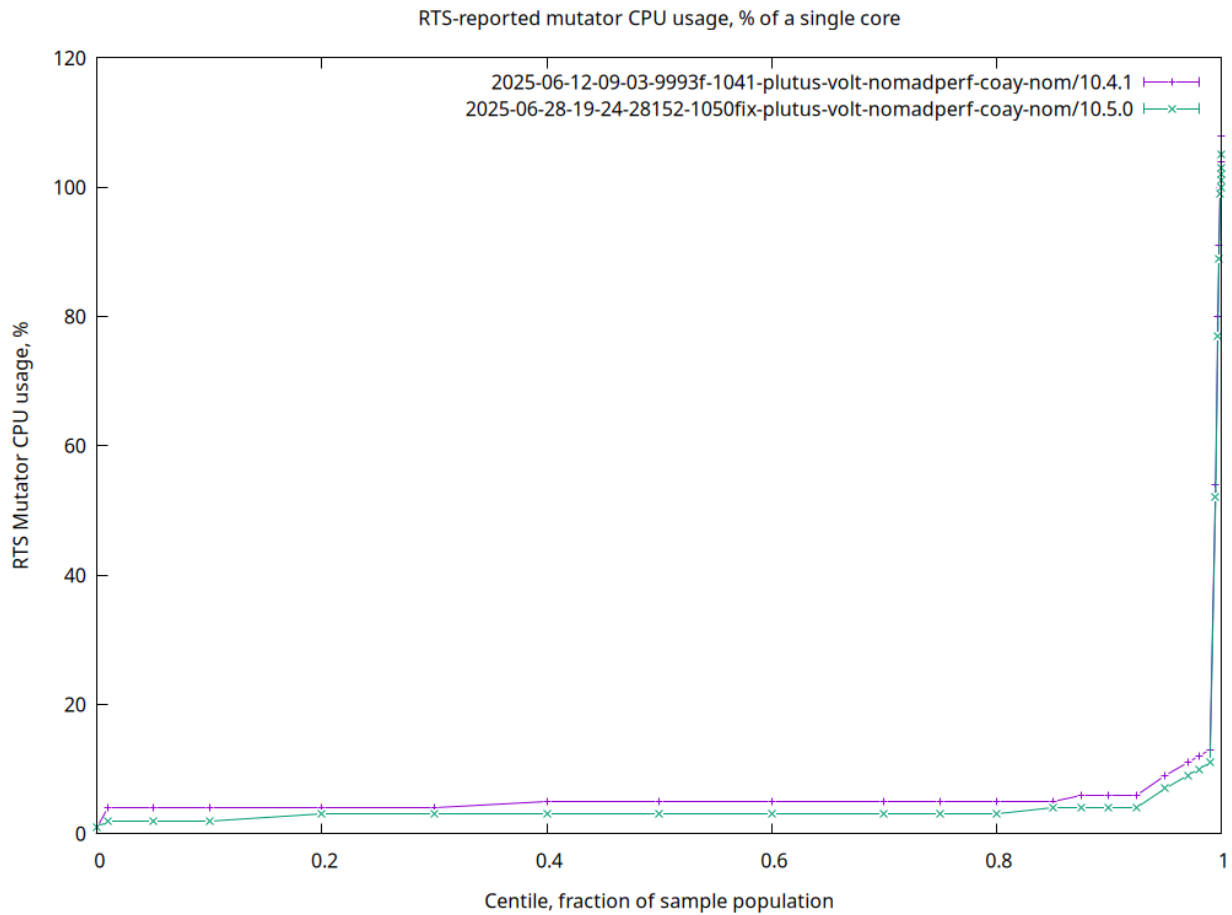


**RTS GC CPU usage (CentiGC)** RTS-reported GC CPU usage, % of a single core

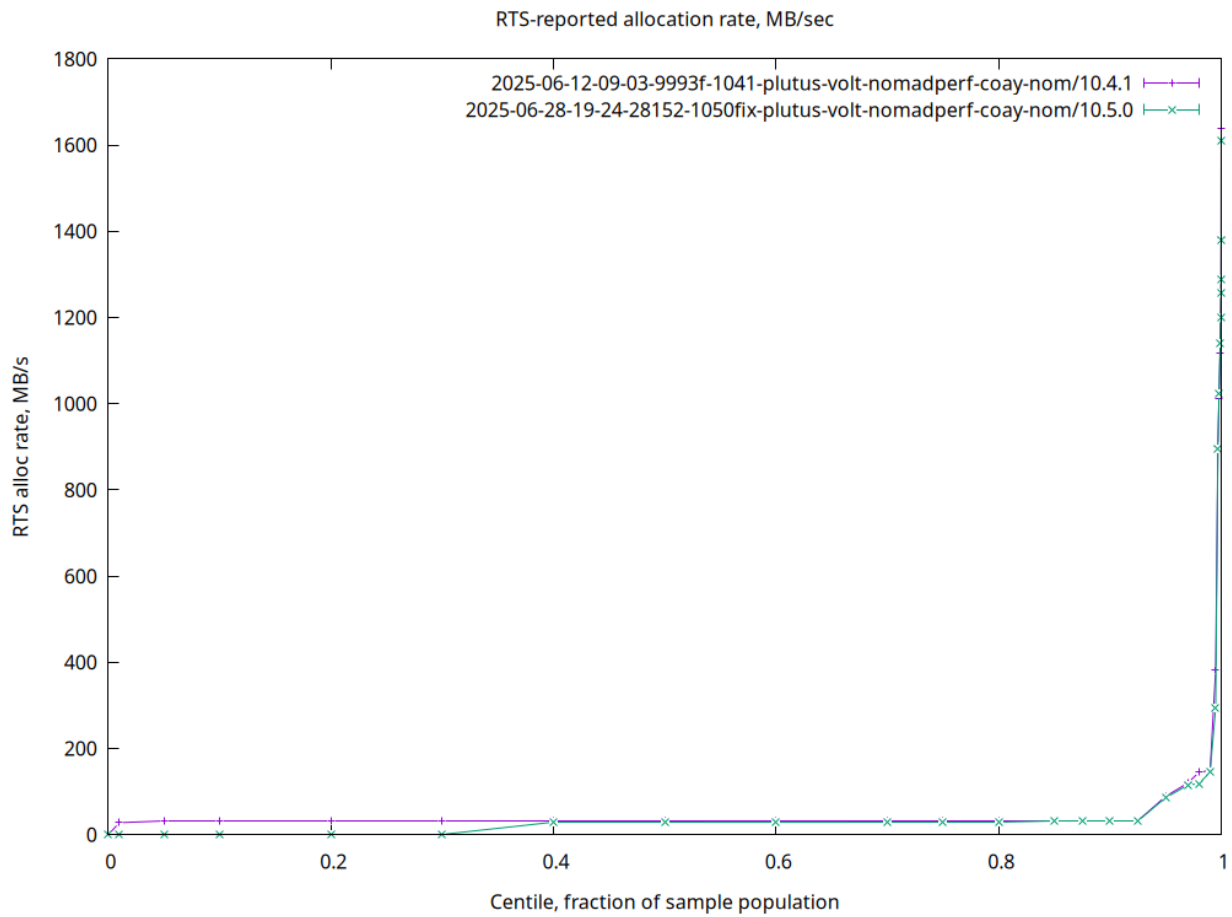




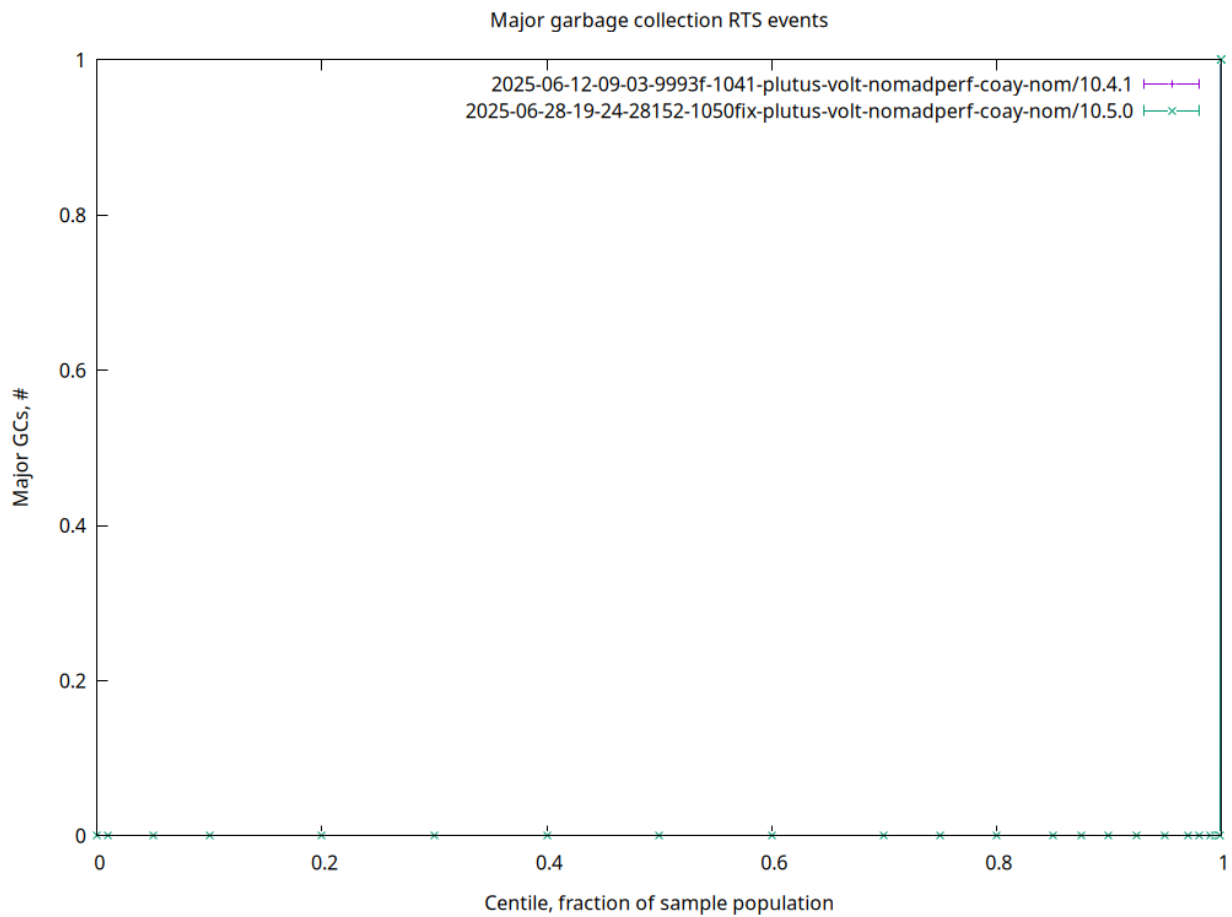
**RTS Mutator CPU usage (CentiMut)** RTS-reported mutator CPU usage, % of a single core



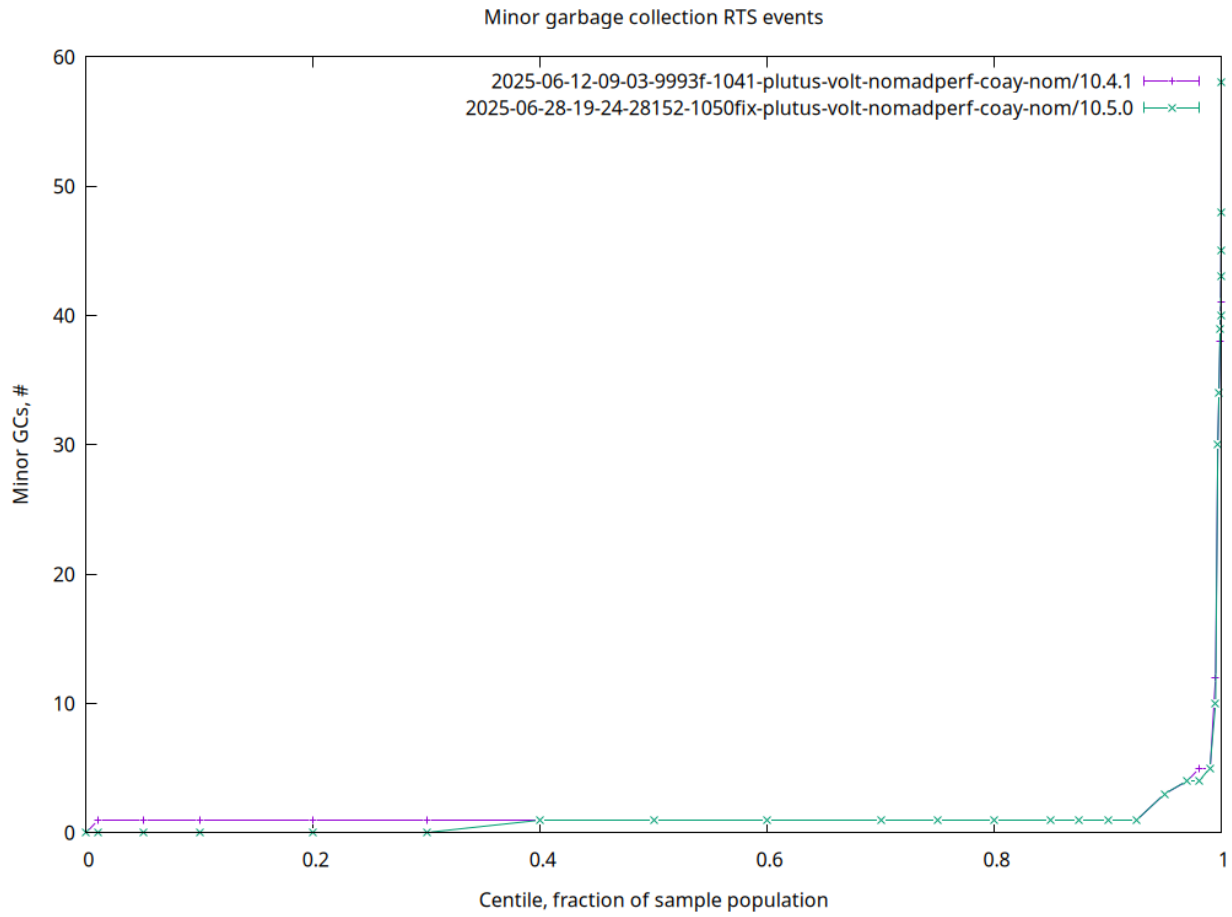
**RTS alloc rate (Alloc)** RTS-reported allocation rate, MB/sec



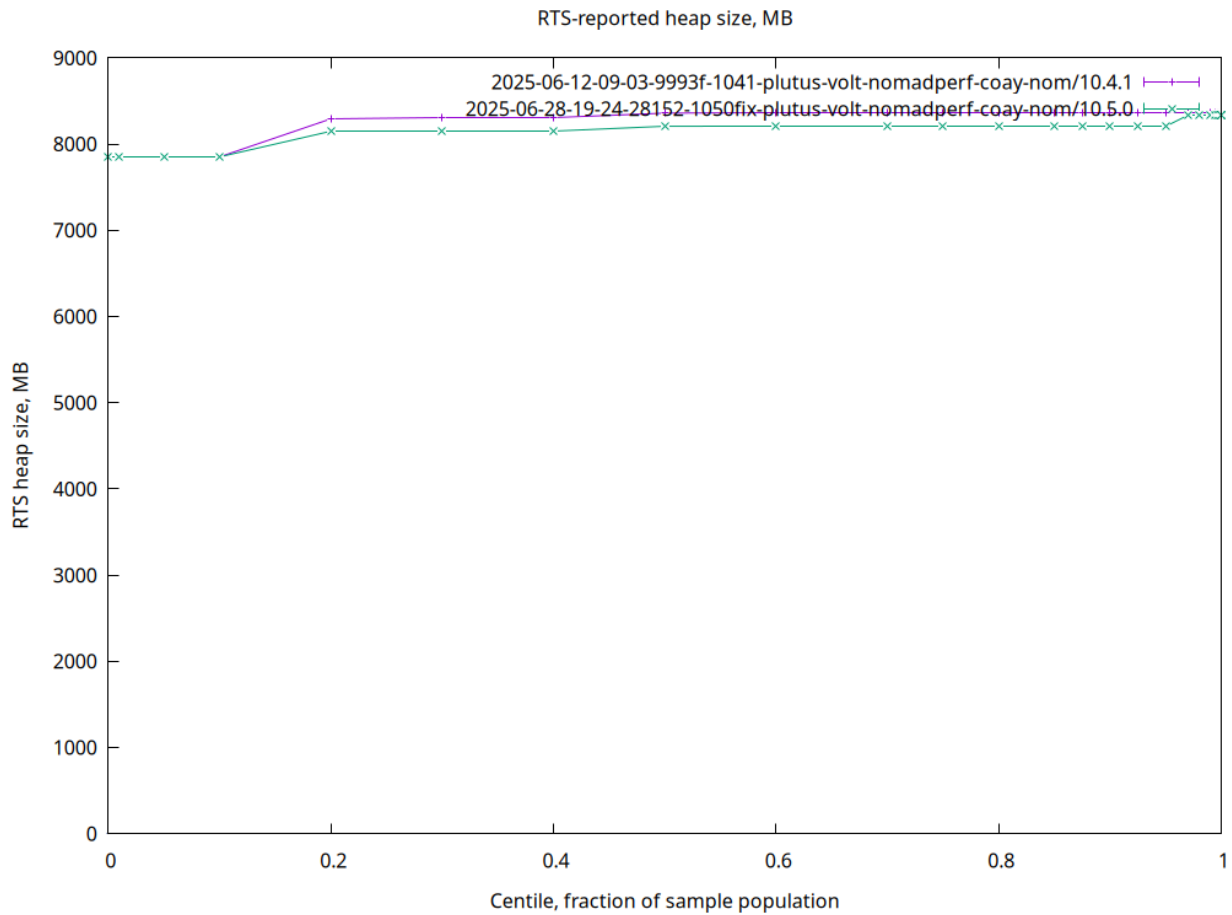
**Major GCs (GcsMajor)** Major garbage collection RTS events



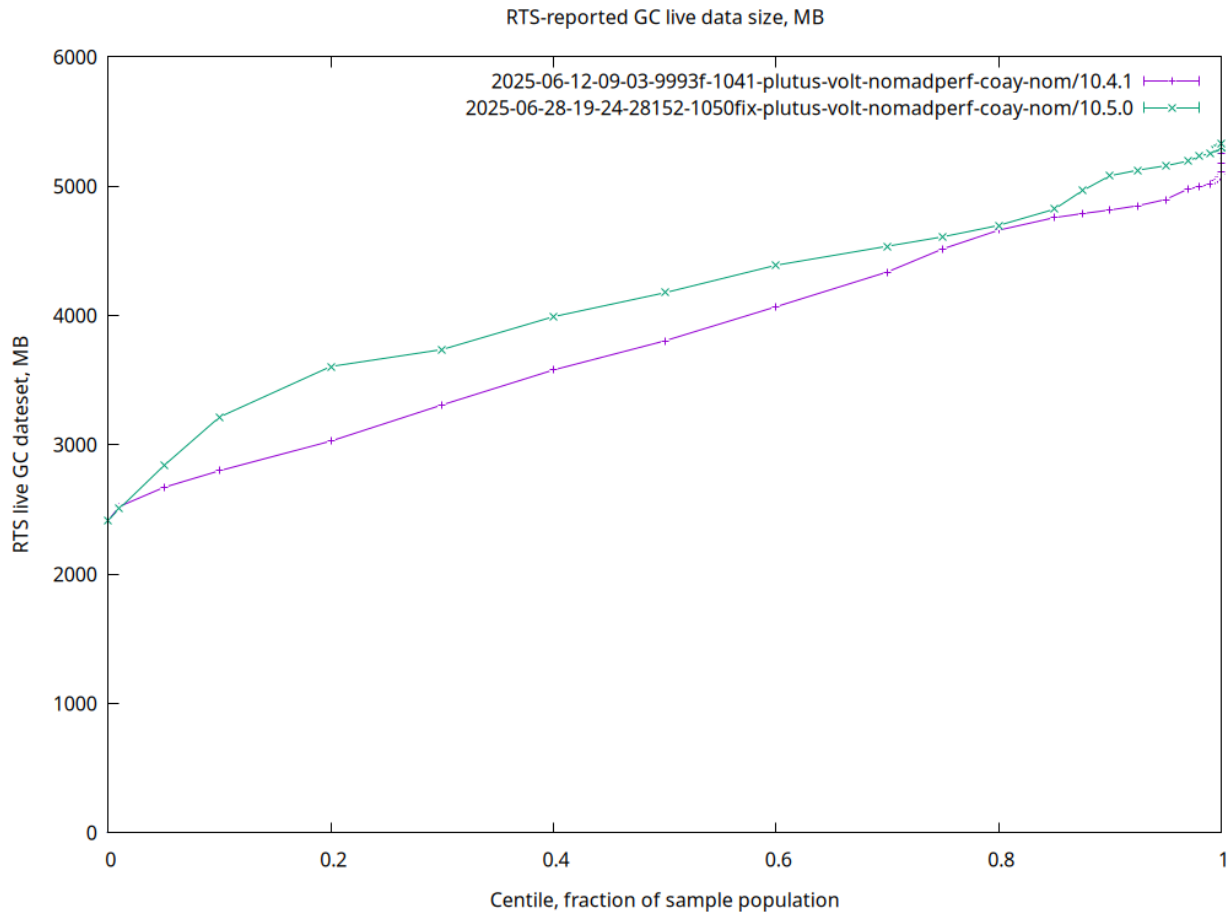
**Minor GCs (GcsMinor)** Minor garbage collection RTS events



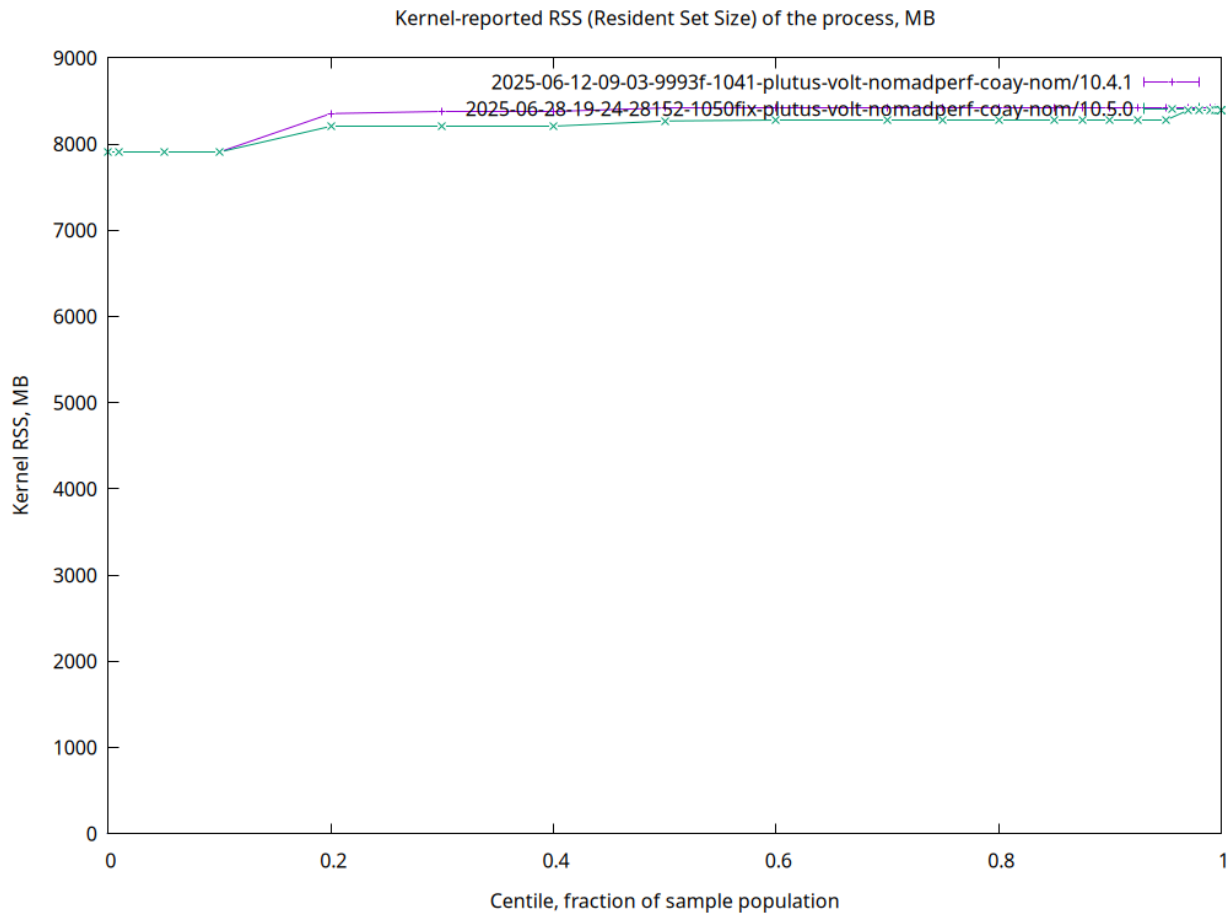
**RTS heap size (Heap)** RTS-reported heap size, MB



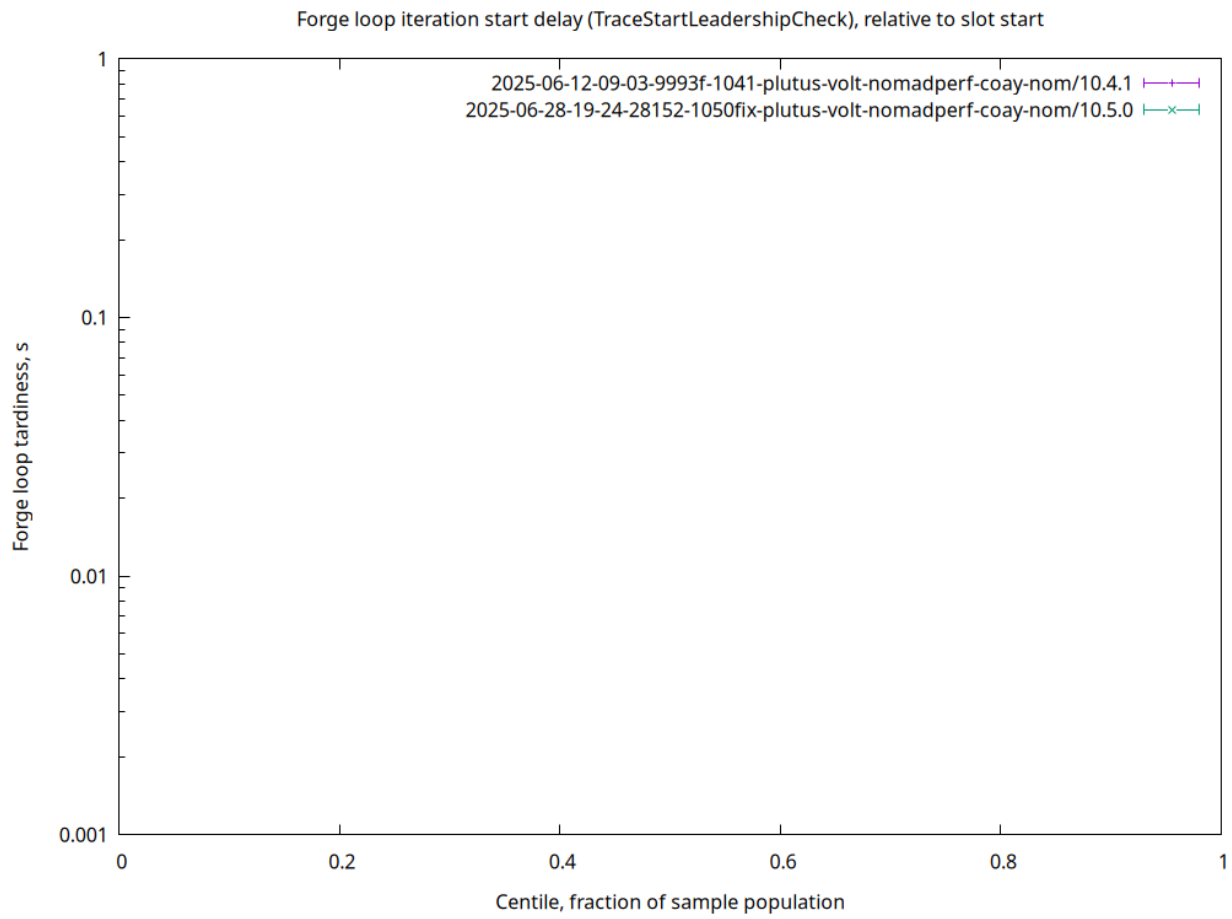
**RTS live GC dataset (Live)** RTS-reported GC live data size, MB



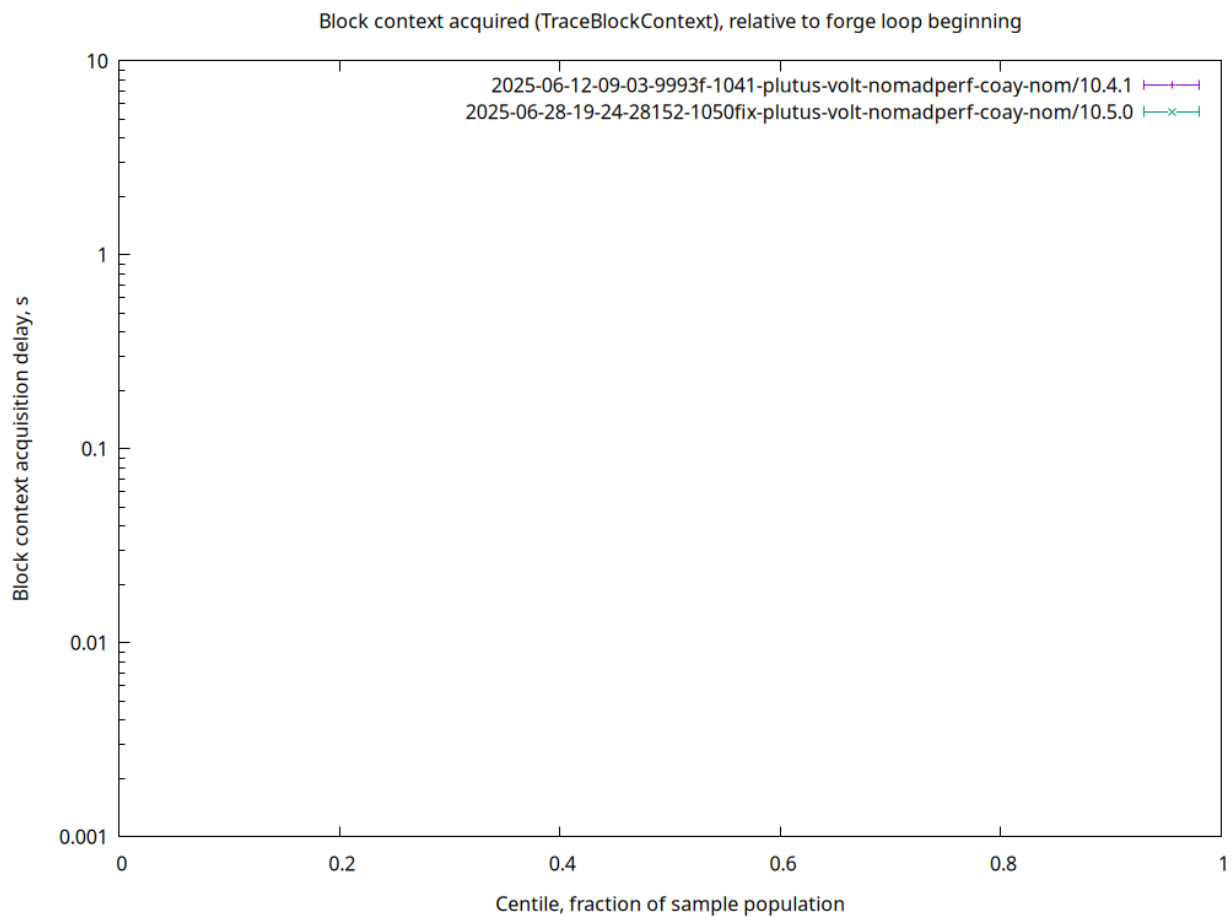
**Kernel RSS (RSS)** Kernel-reported RSS (Resident Set Size) of the process, MB



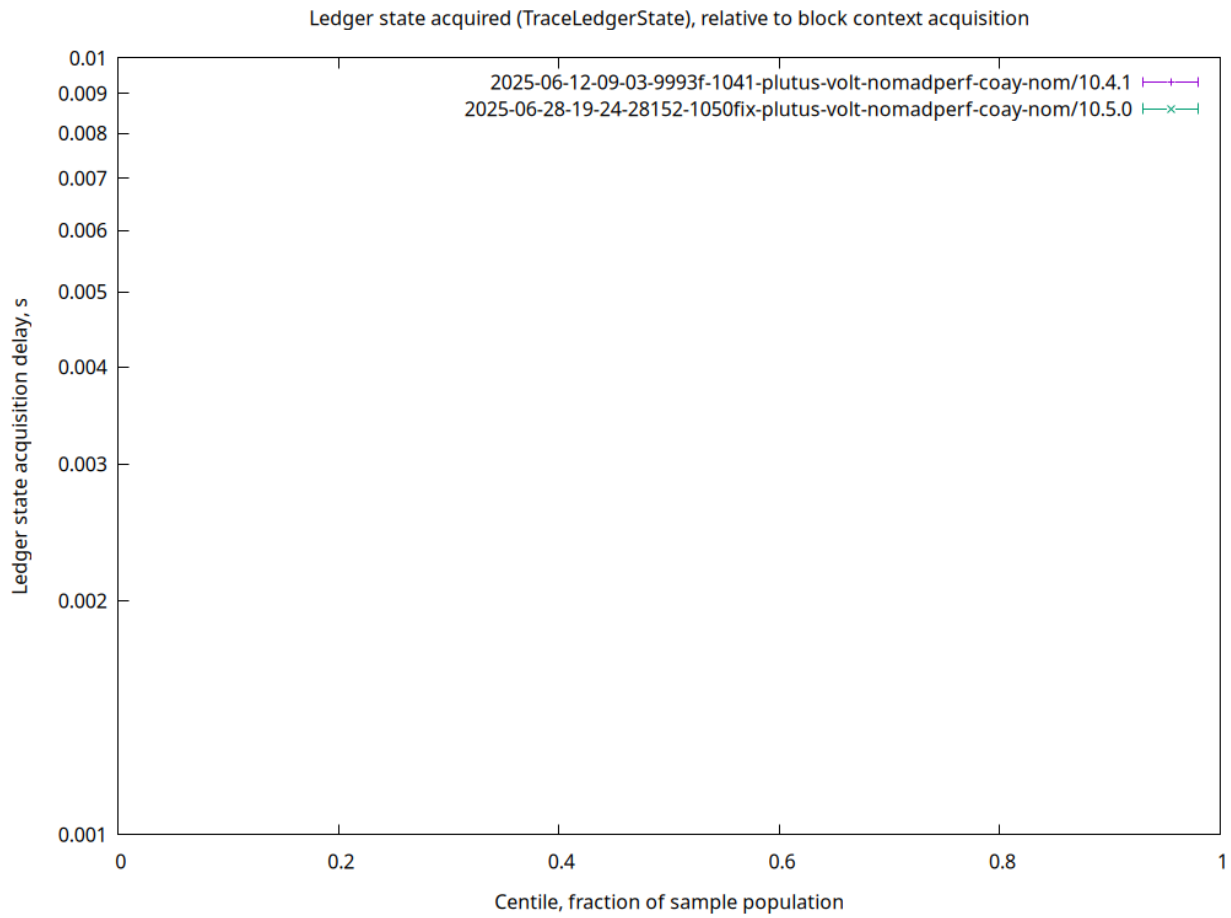
**Forge loop tardiness (cdfStarted)** Forge loop iteration start delay (TraceStartLeadershipCheck), relative to slot start



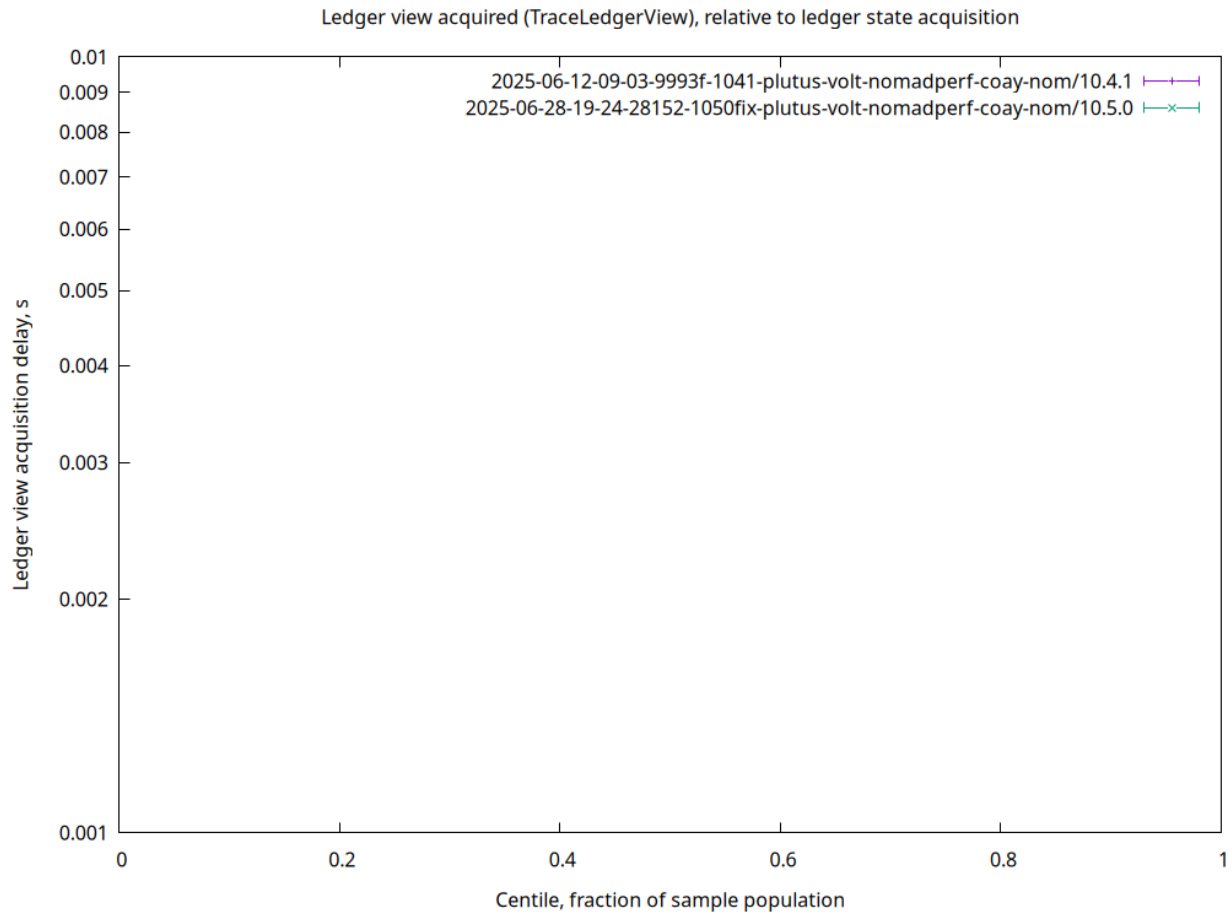
**Block context acquisition delay (cdfBlkCtx)** Block context acquired (TraceBlockContext), relative to forge loop beginning



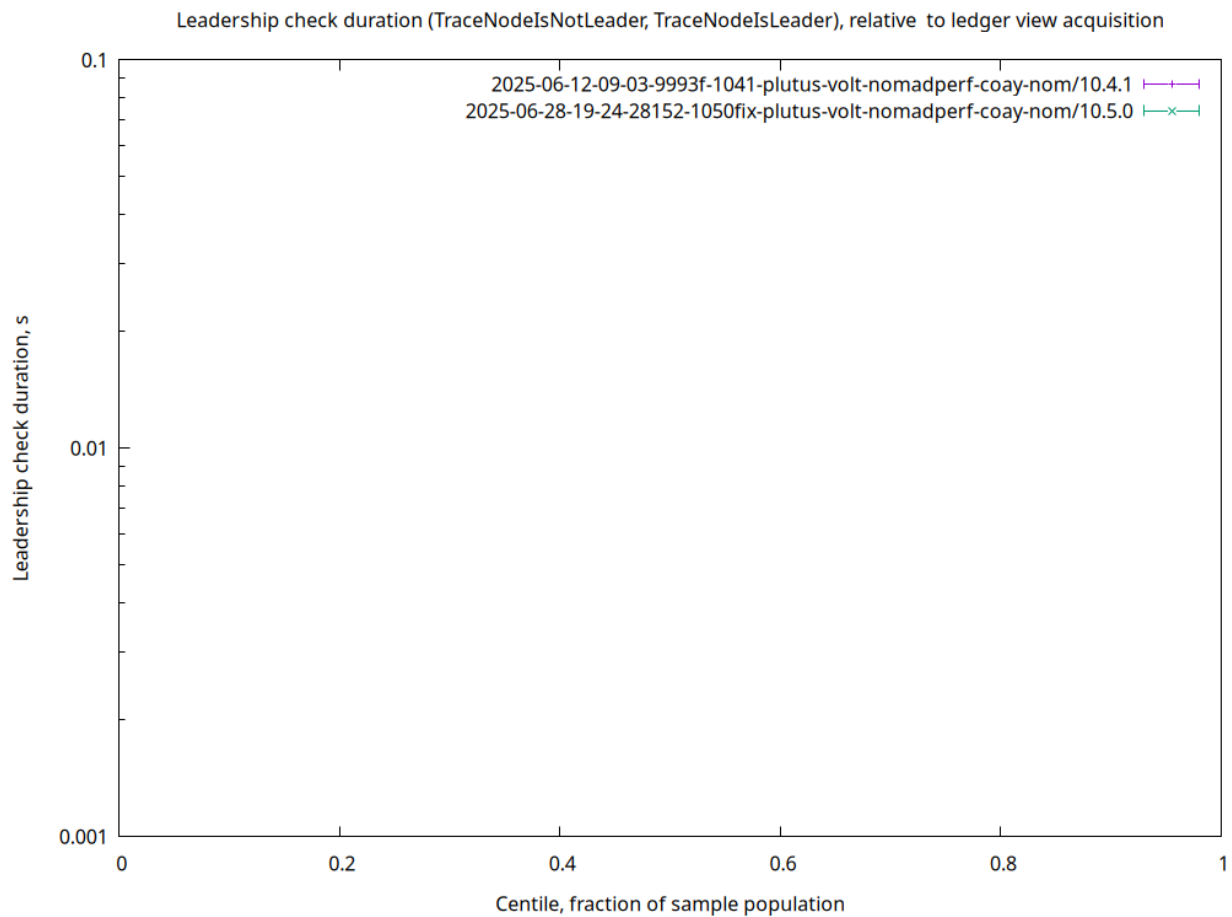
**Ledger state acquisition delay (cdfLgrState)** Ledger state acquired (TraceLedgerState), relative to block context acquisition



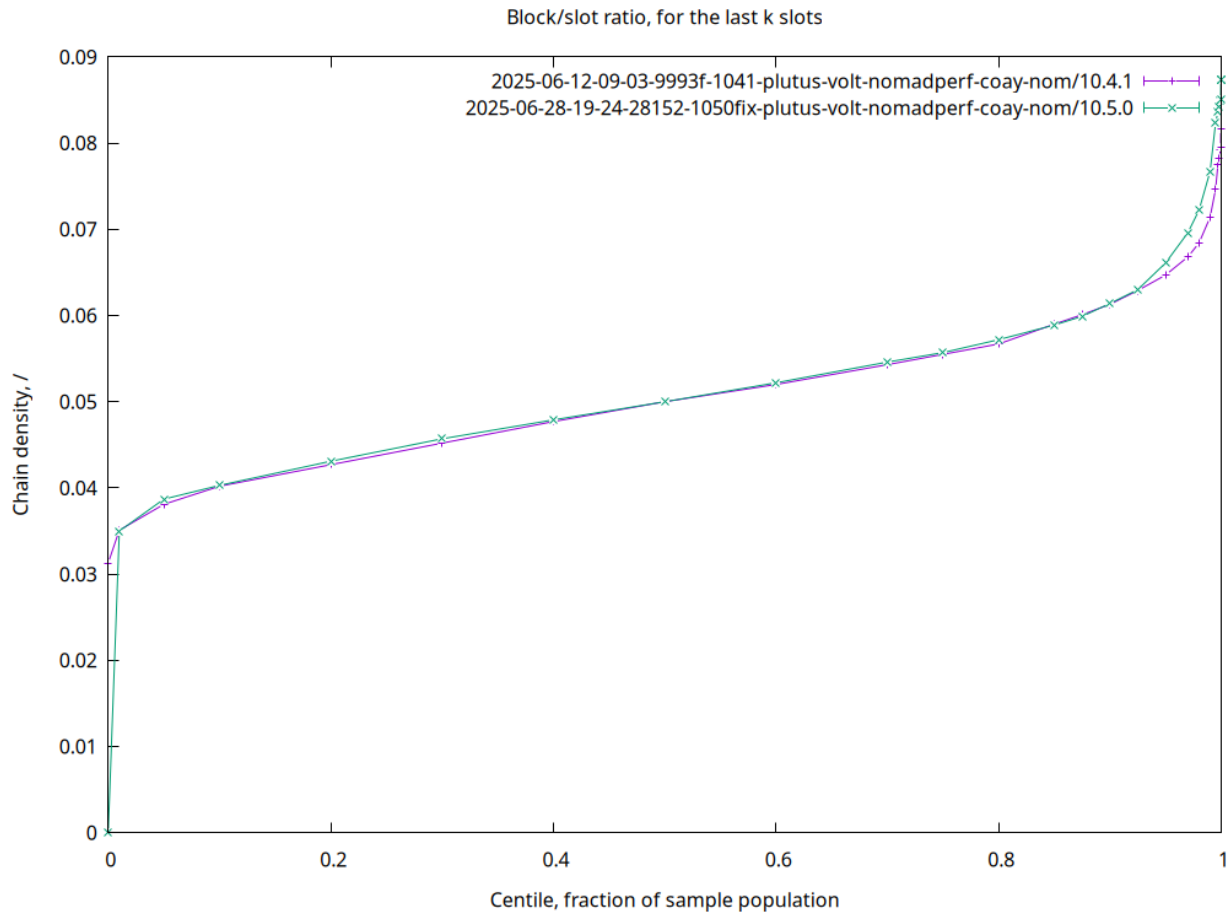
**Ledger view acquisition delay (cdfLgrView)** Ledger view acquired (TraceLedgerView), relative to ledger state acquisition



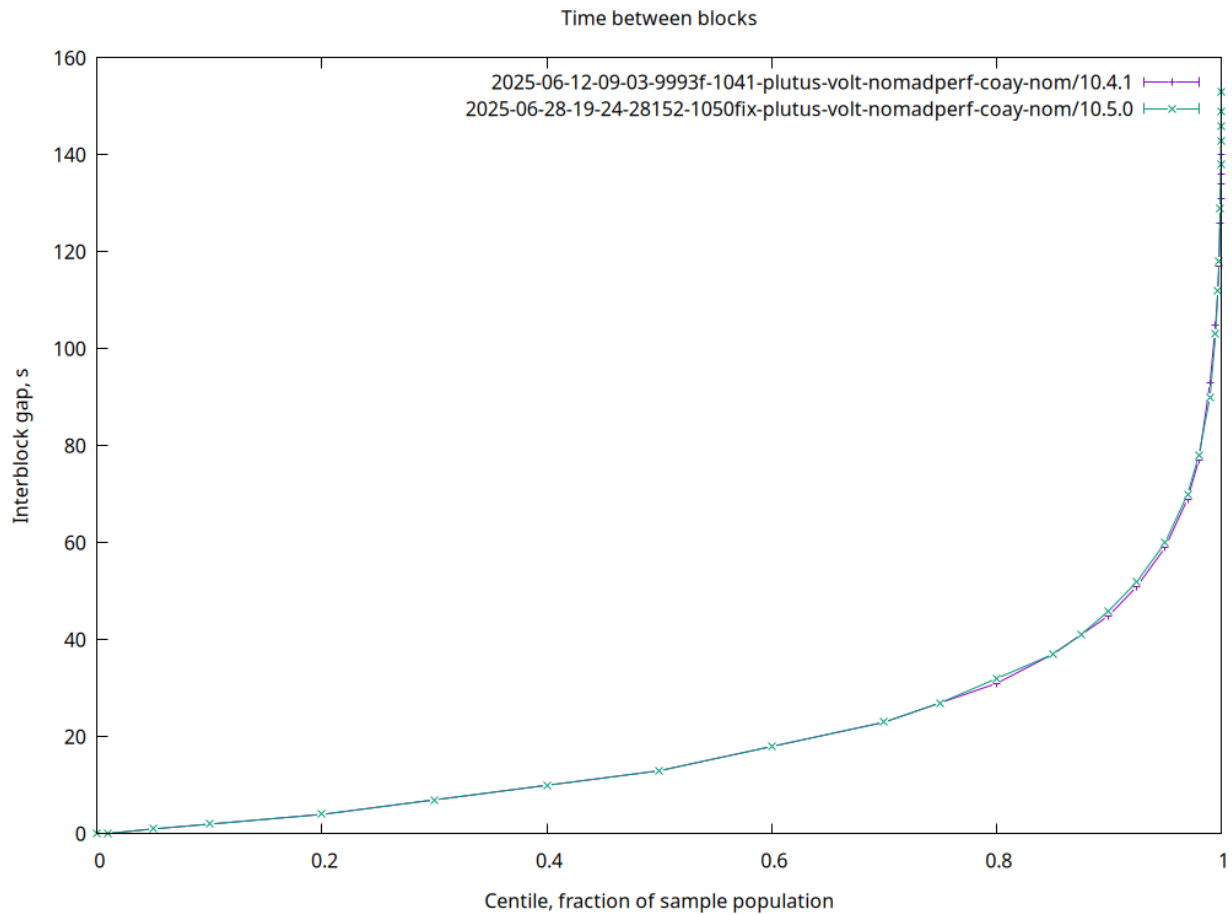
**Leadership check duration (cdfLeading)** Leadership check duration (TraceNodeIsNotLeader, TraceNodeIsLeader), relative to ledger view acquisition



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

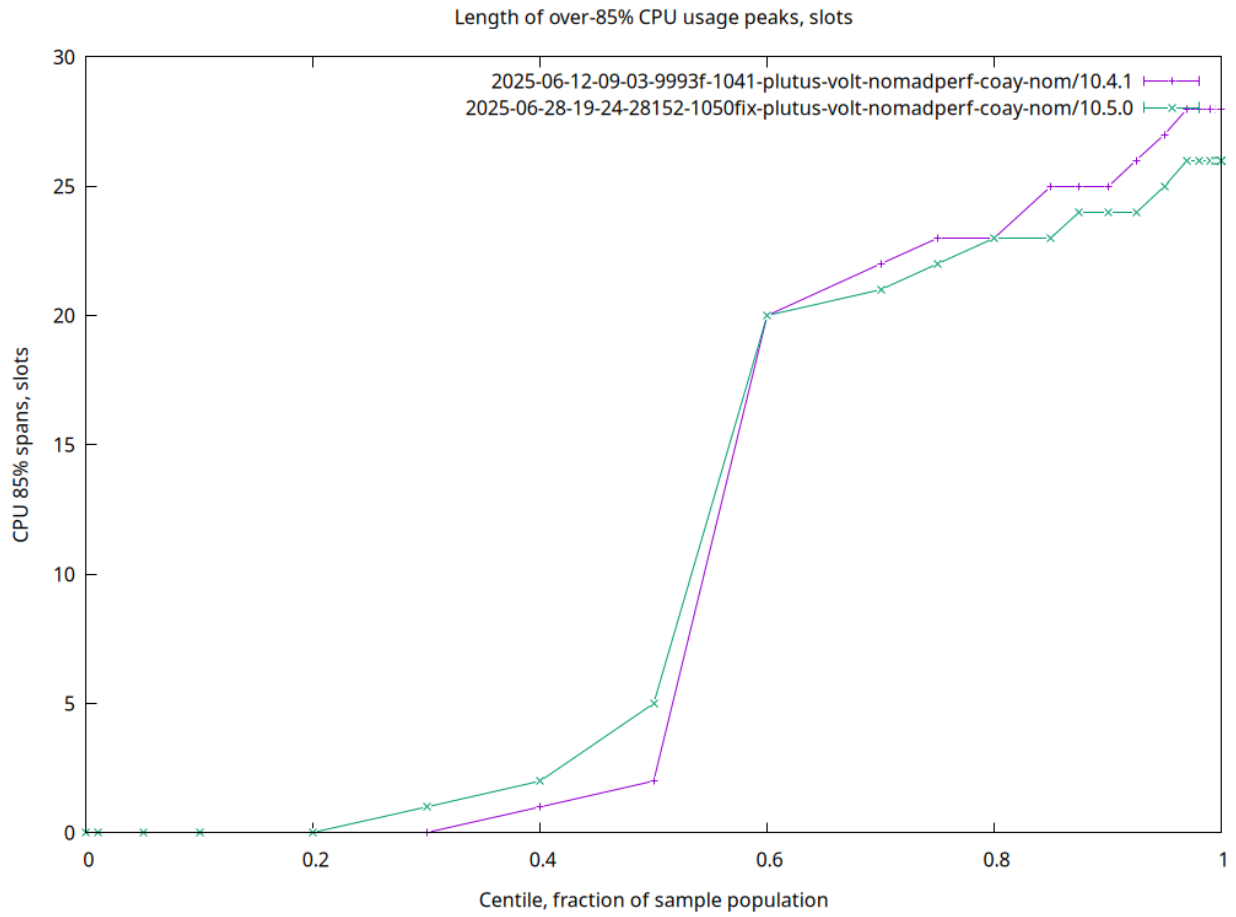


Interblock gap (cdfBlockGap) Time between blocks

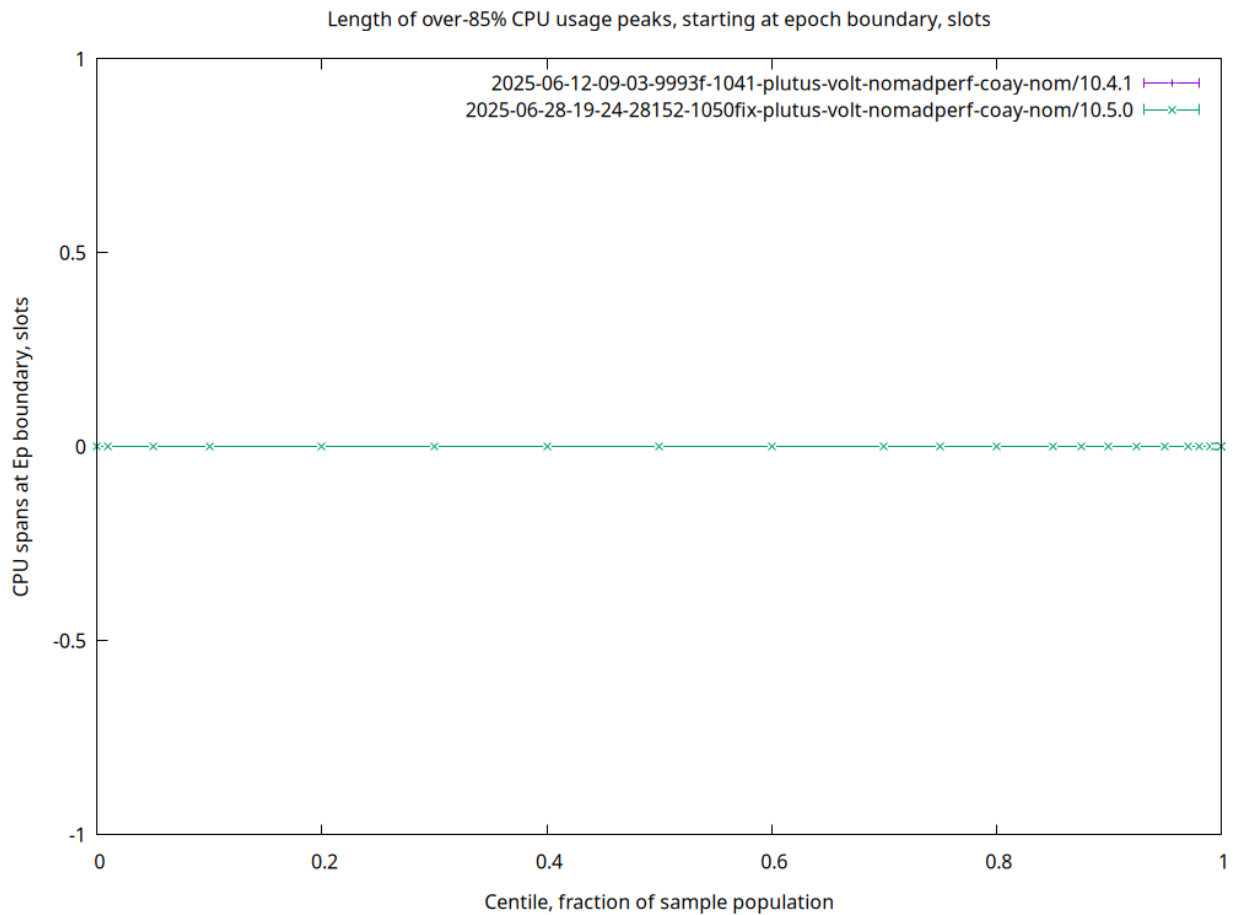


CPU 85% spans (cdfSpanLensCpu) Length of over-85% CPU usage peaks, slots

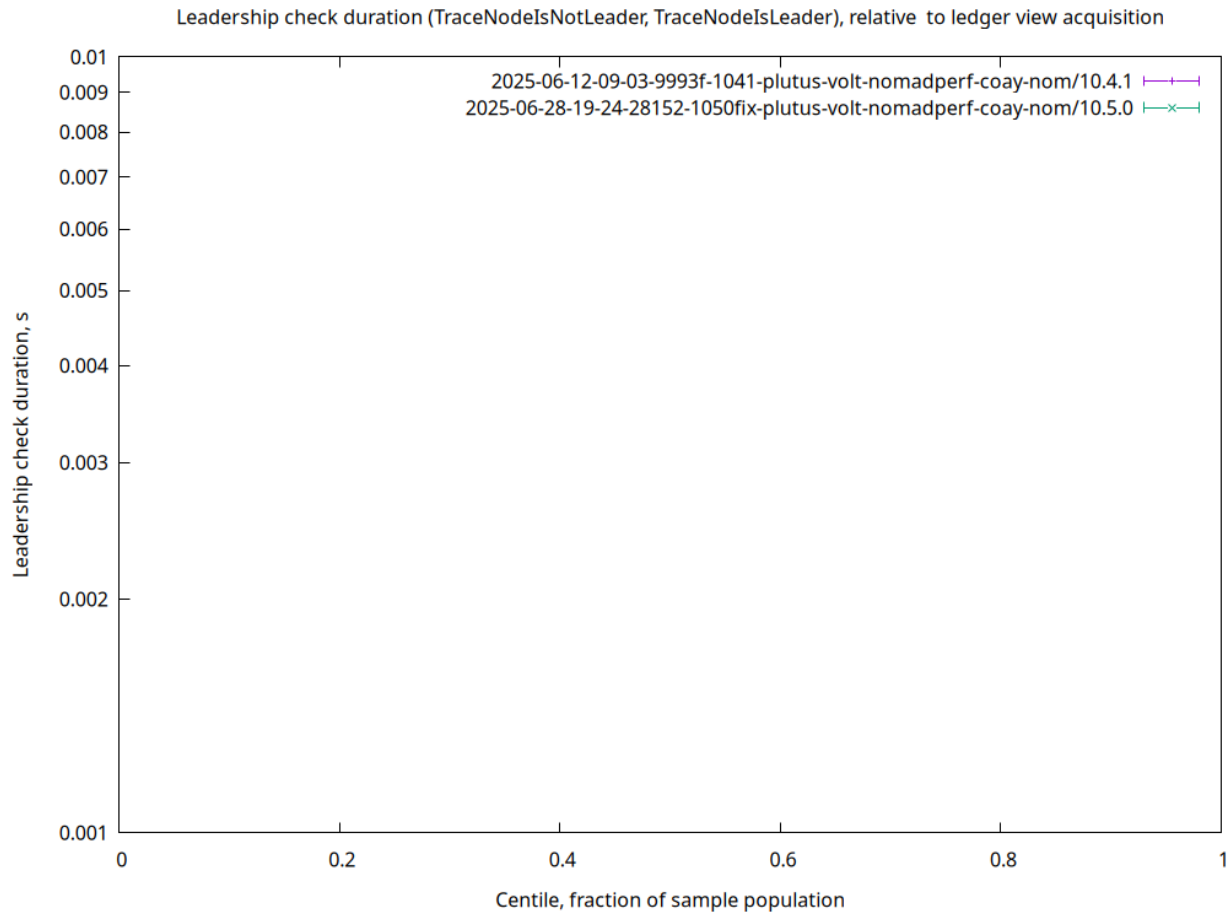




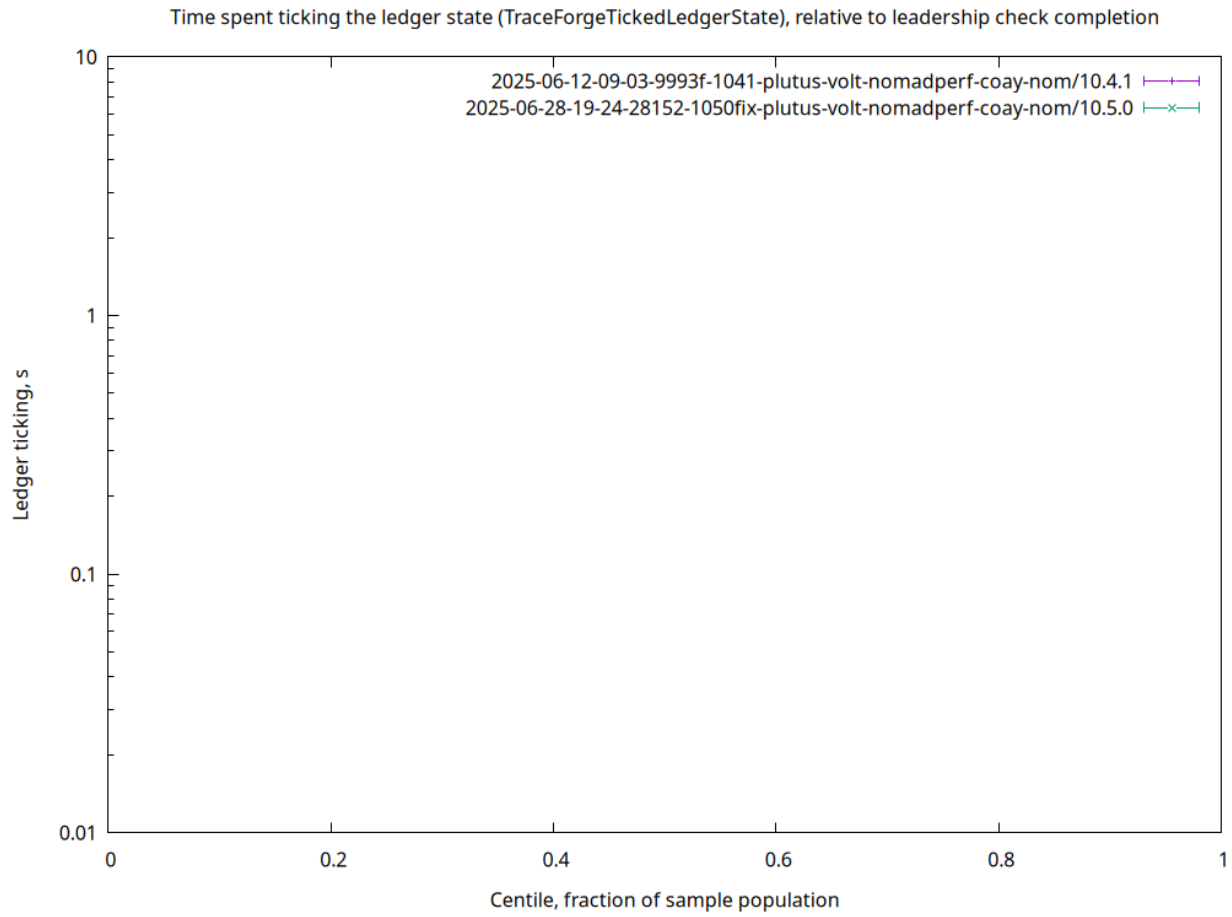
**CPU spans at Ep boundary (cdfSpanLensCpuEpoch)** Length of over-85% CPU usage peaks, starting at epoch boundary, slots



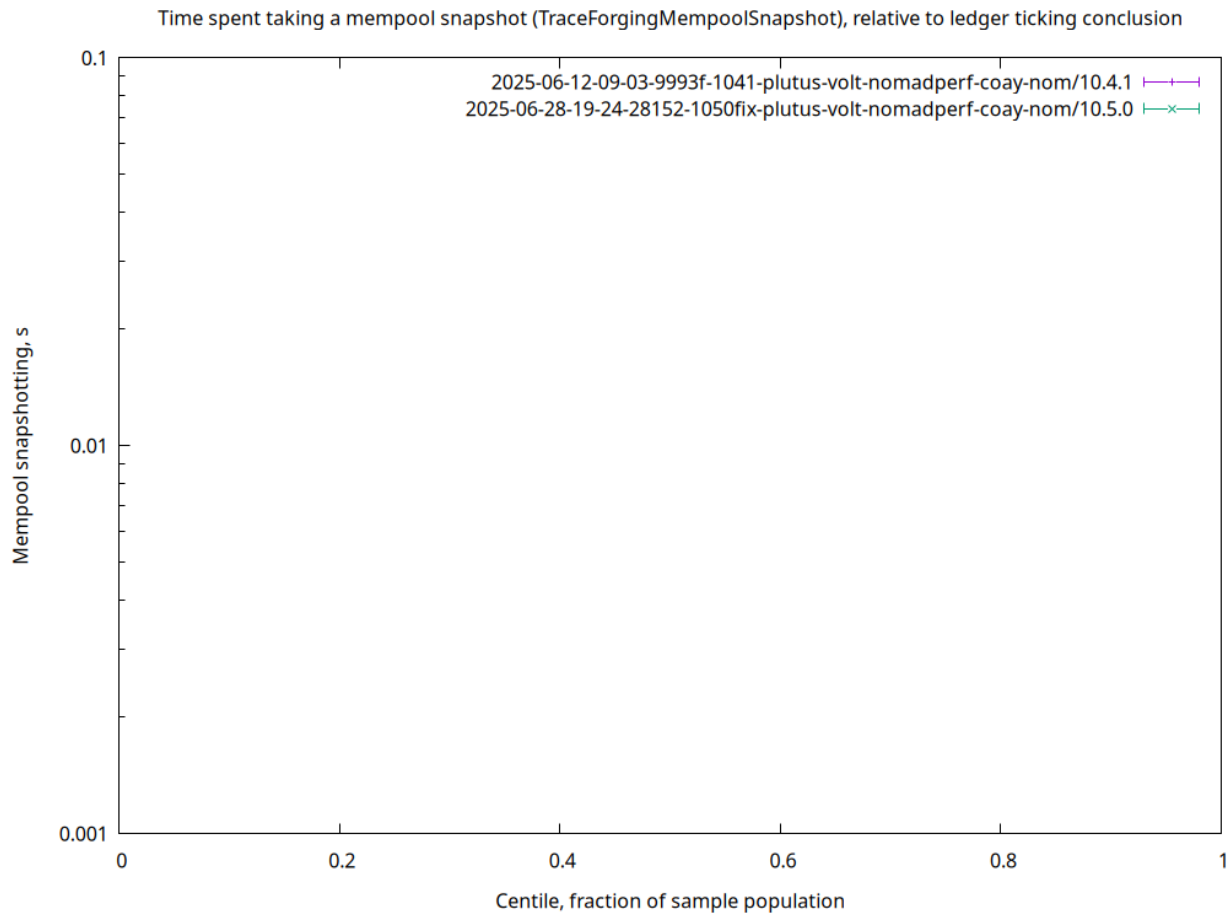
**Leadership check duration (cdfForgerLead)** Leadership check duration (TraceNodeIsNotLeader, TraceNodeIsLeader), relative to ledger view acquisition



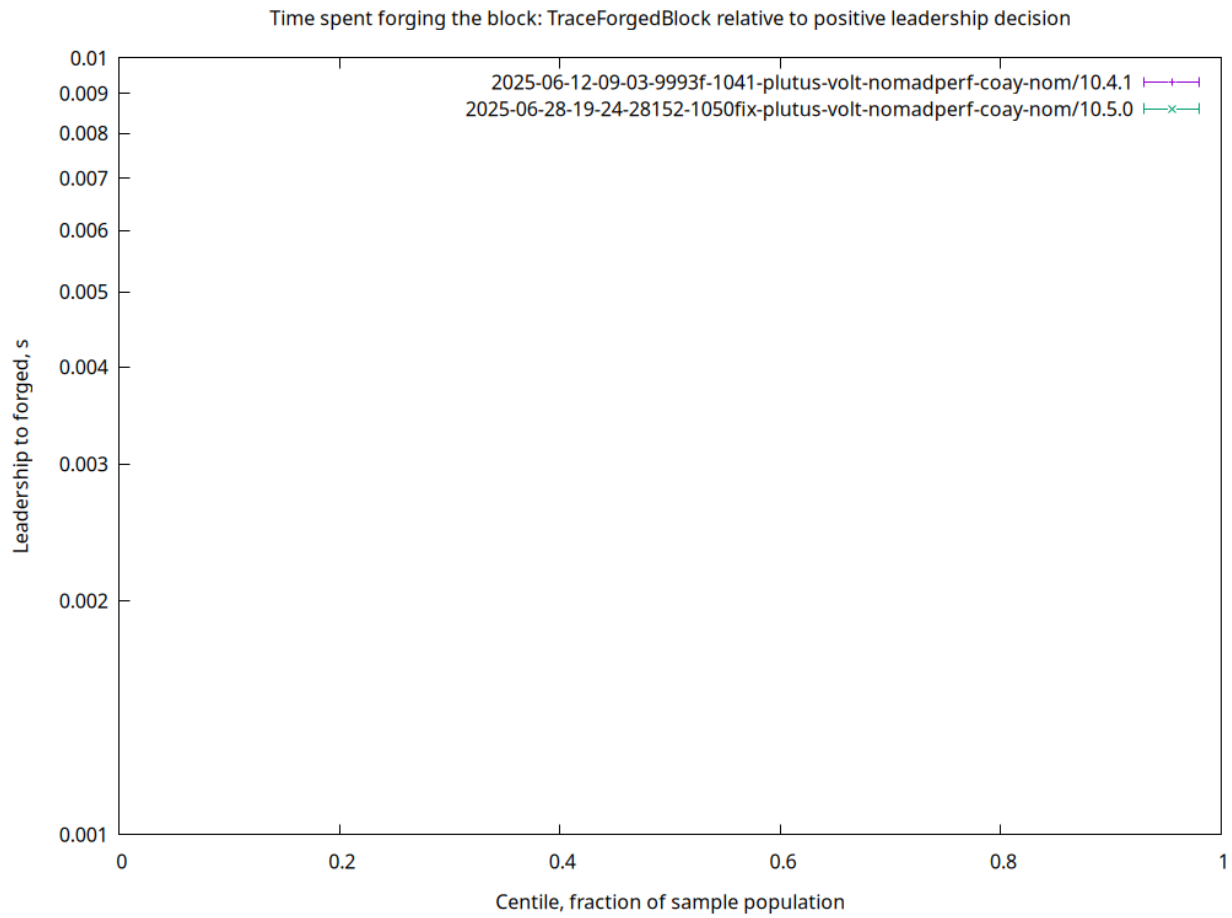
**Ledger ticking (cdfForgerTicked)** Time spent ticking the ledger state (TraceForgeTickedLedgerState), relative to leadership check completion



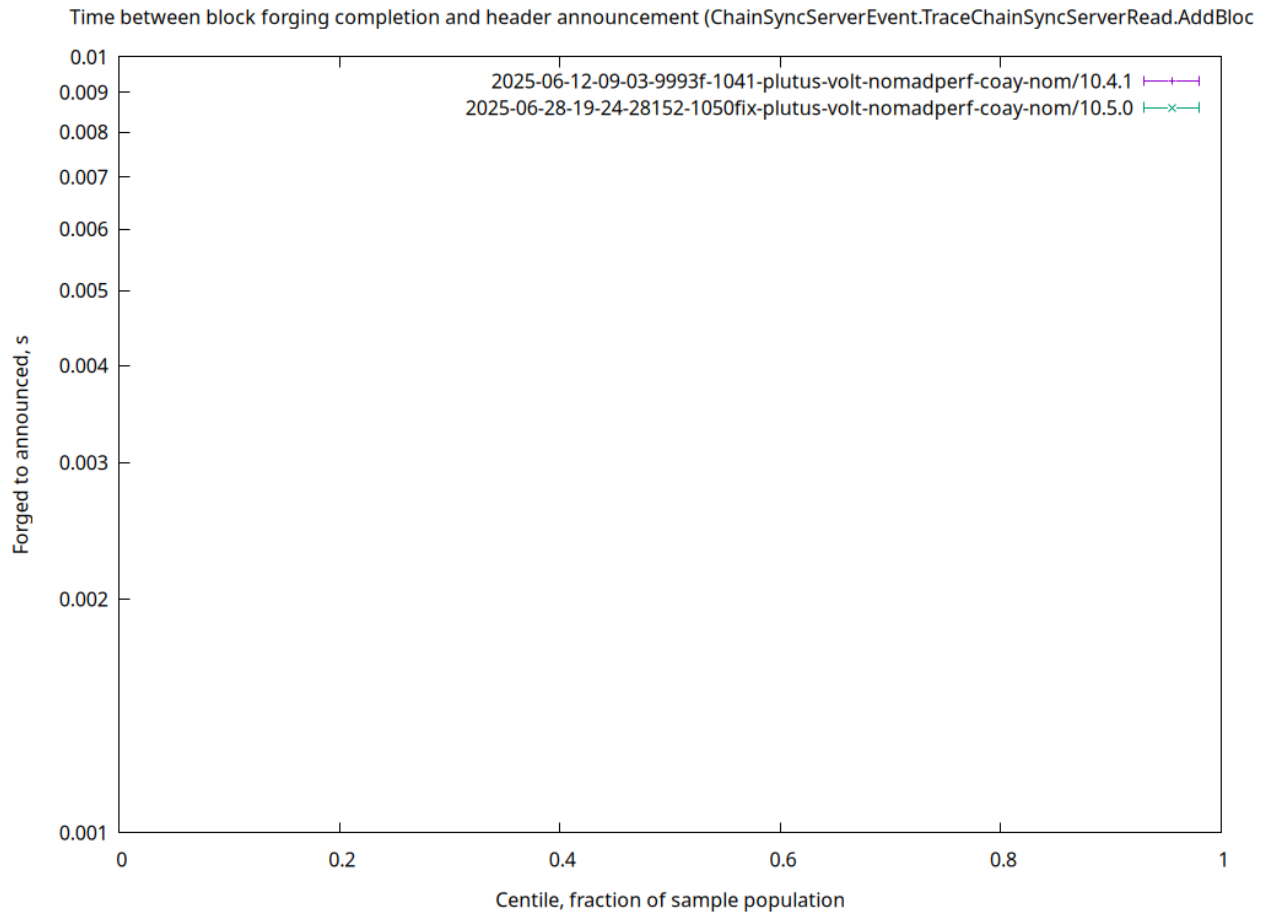
**Mempool snapshotting (cdfForgerMemSnap)** Time spent taking a mempool snapshot (TraceForgingMempool-Snapshot), relative to ledger ticking conclusion



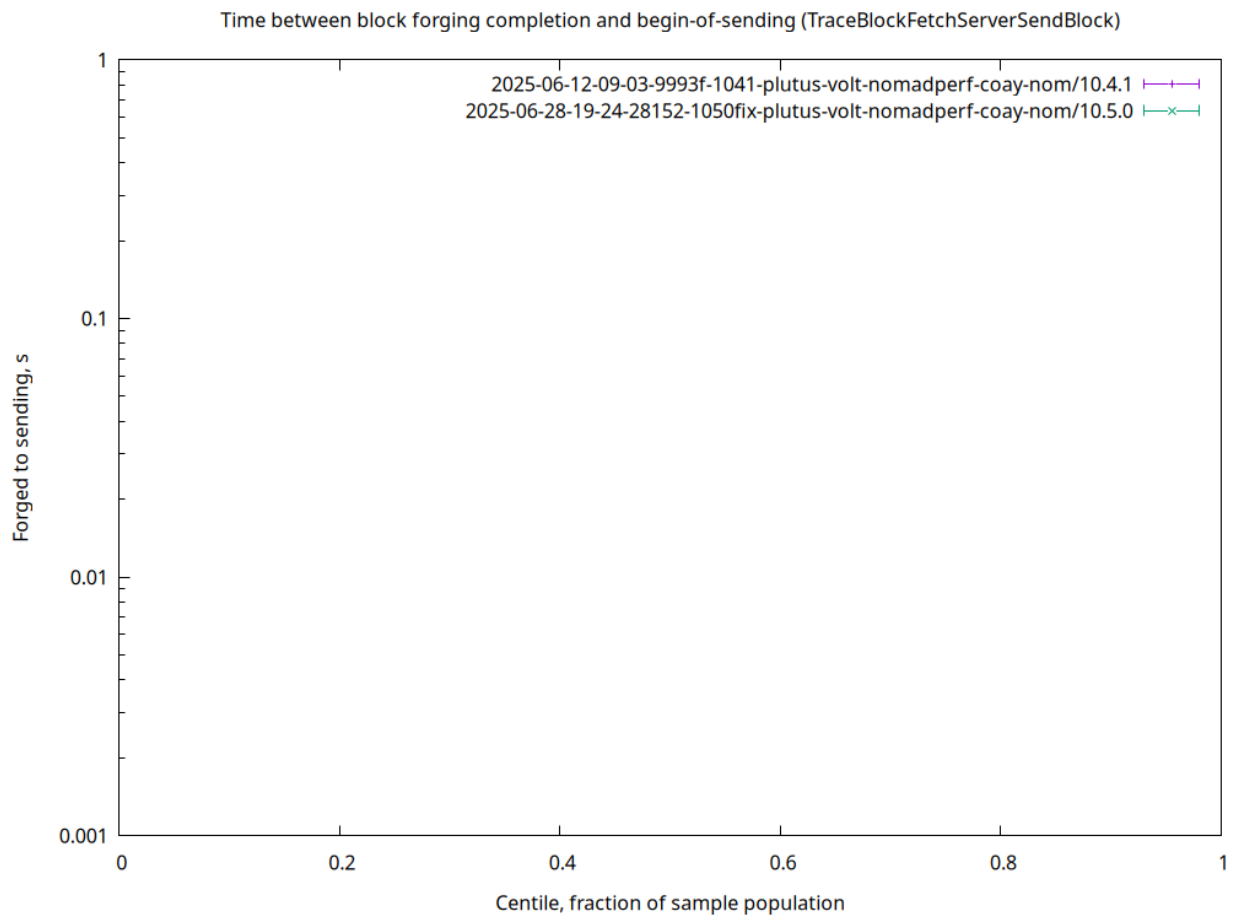
**Leadership to forged (cdfForgerForge)** Time spent forging the block: TraceForgedBlock relative to positive leadership decision



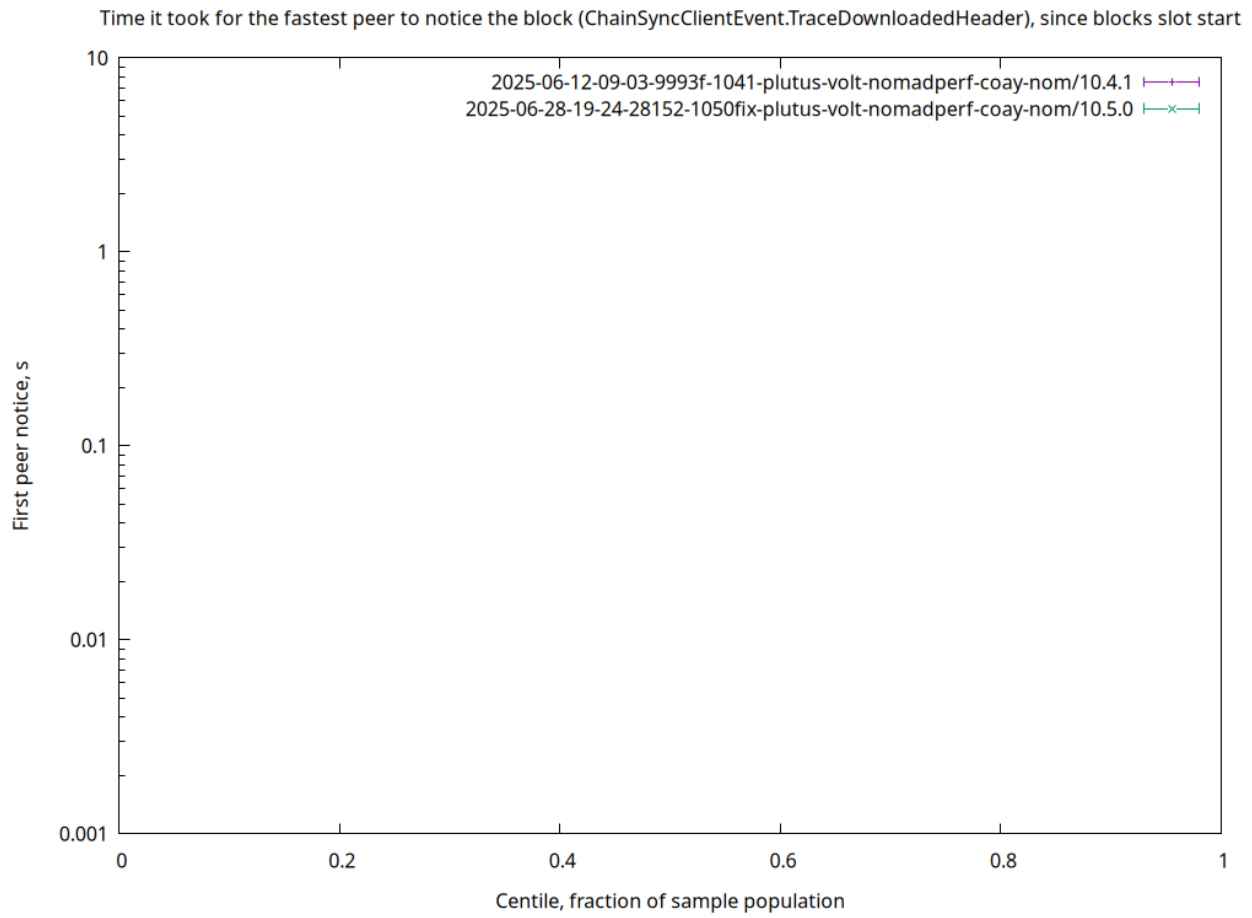
**Forged to announced (cdfForgerAnnounce)** Time between block forging completion and header announcement (ChainSyncServerEvent.TraceChainSyncServerRead.AddBlock)



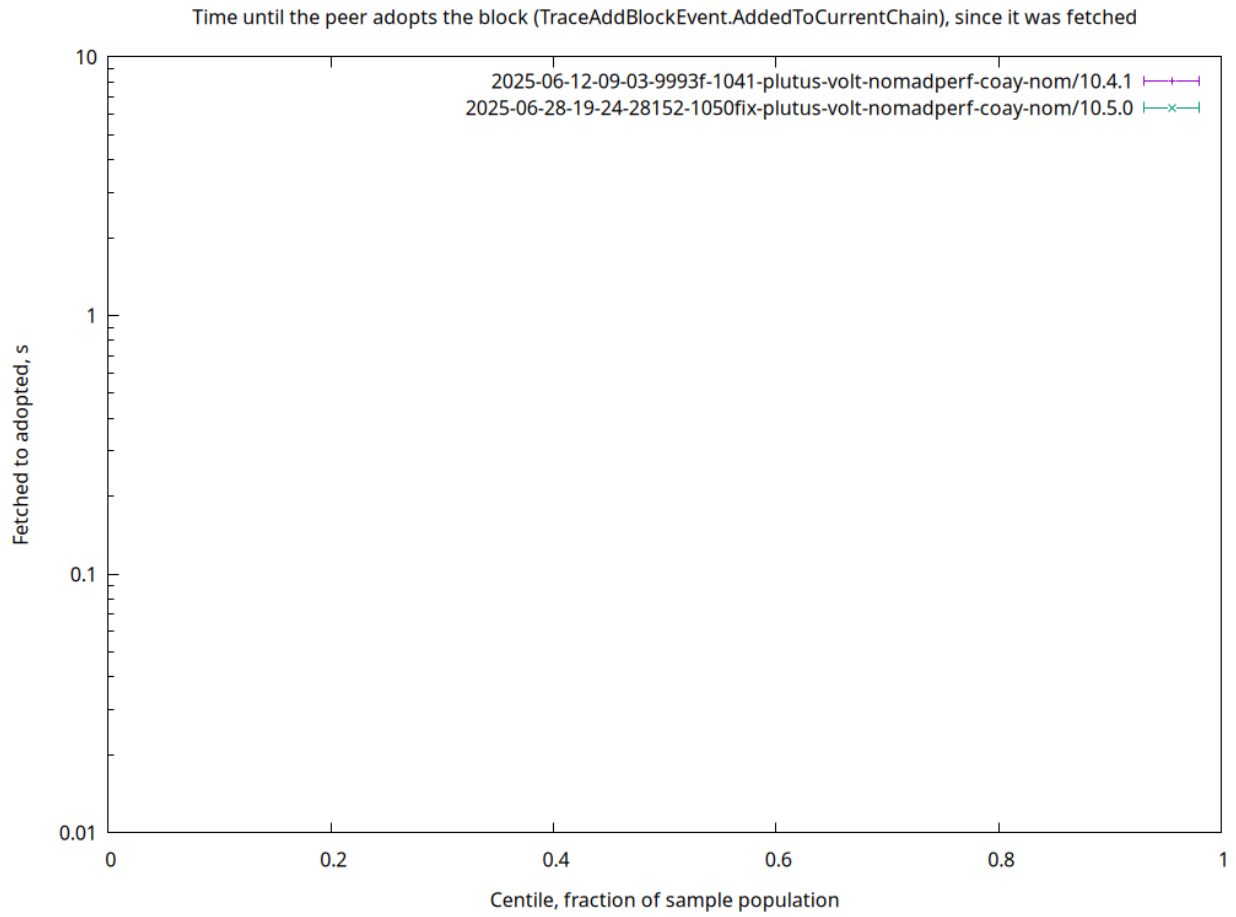
**Forged to sending (cdfForgerSend)** Time between block forging completion and begin-of-sending (TraceBlockFetchServerSendBlock)



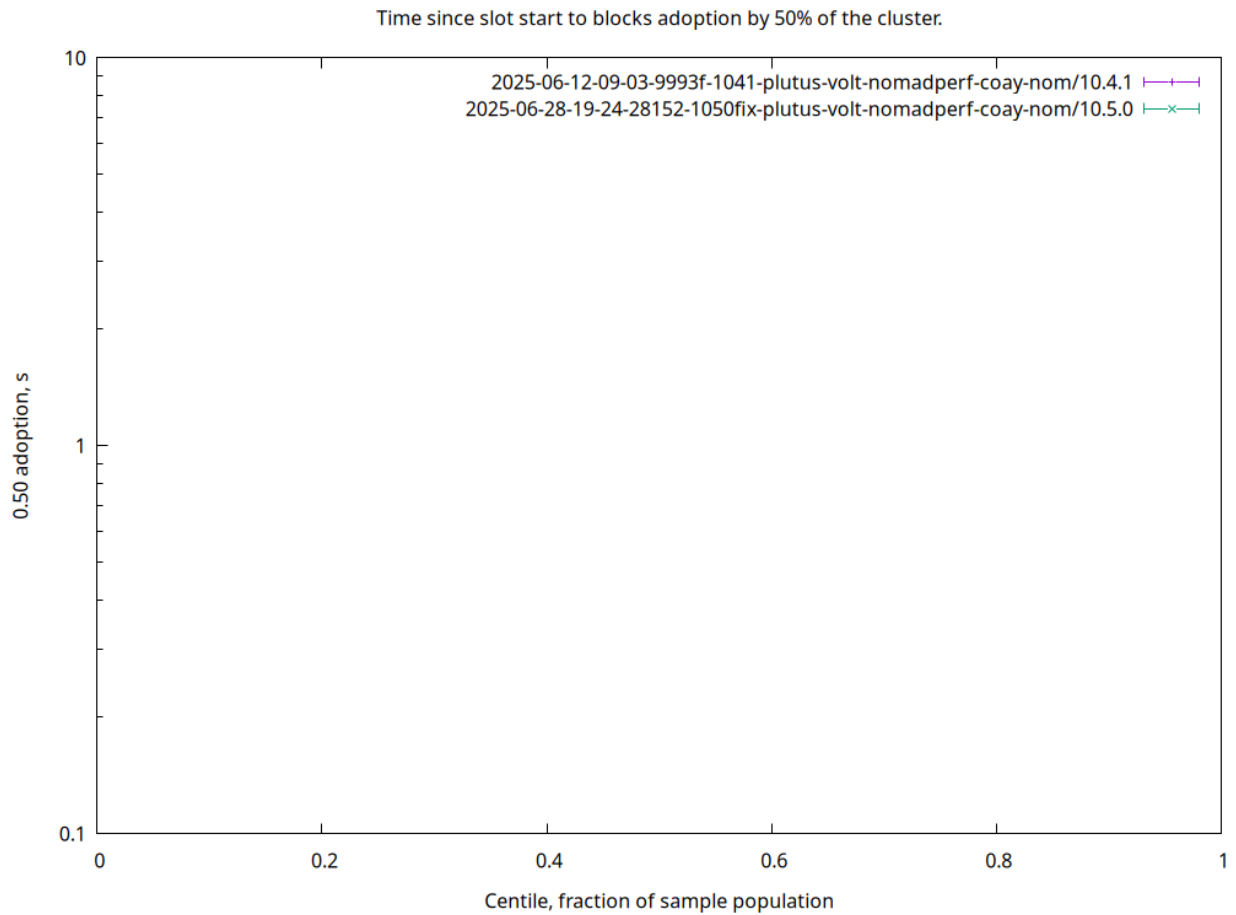
**First peer notice (cdfPeerNoticeFirst)** Time it took for the fastest peer to notice the block (ChainSyncClientEvent.TraceDownloadedHeader), since block's slot start



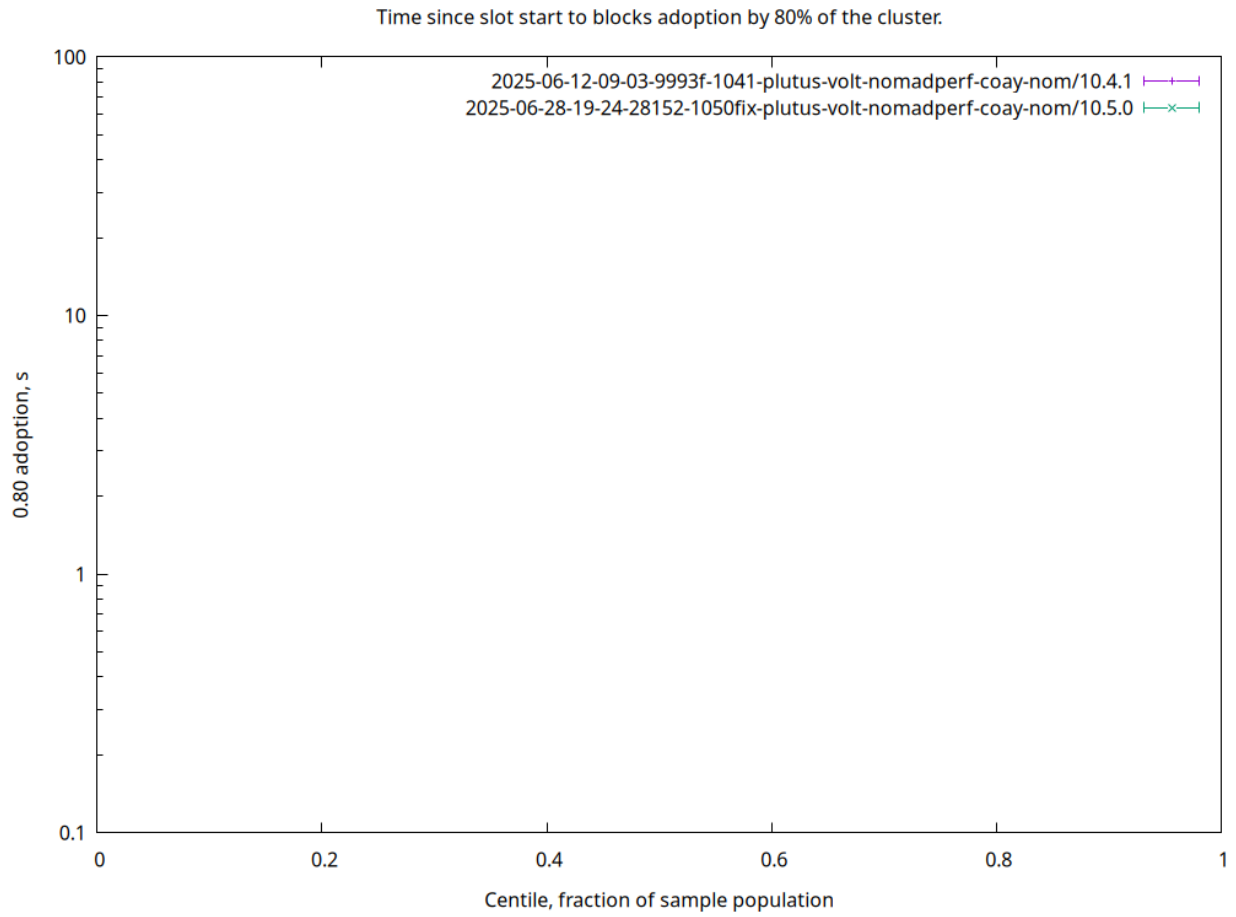
**Fetches to adopted (cdfPeerAdoption)** Time until the peer adopts the block (TraceAddBlockEvent.AddedToCurrentChain), since it was fetched



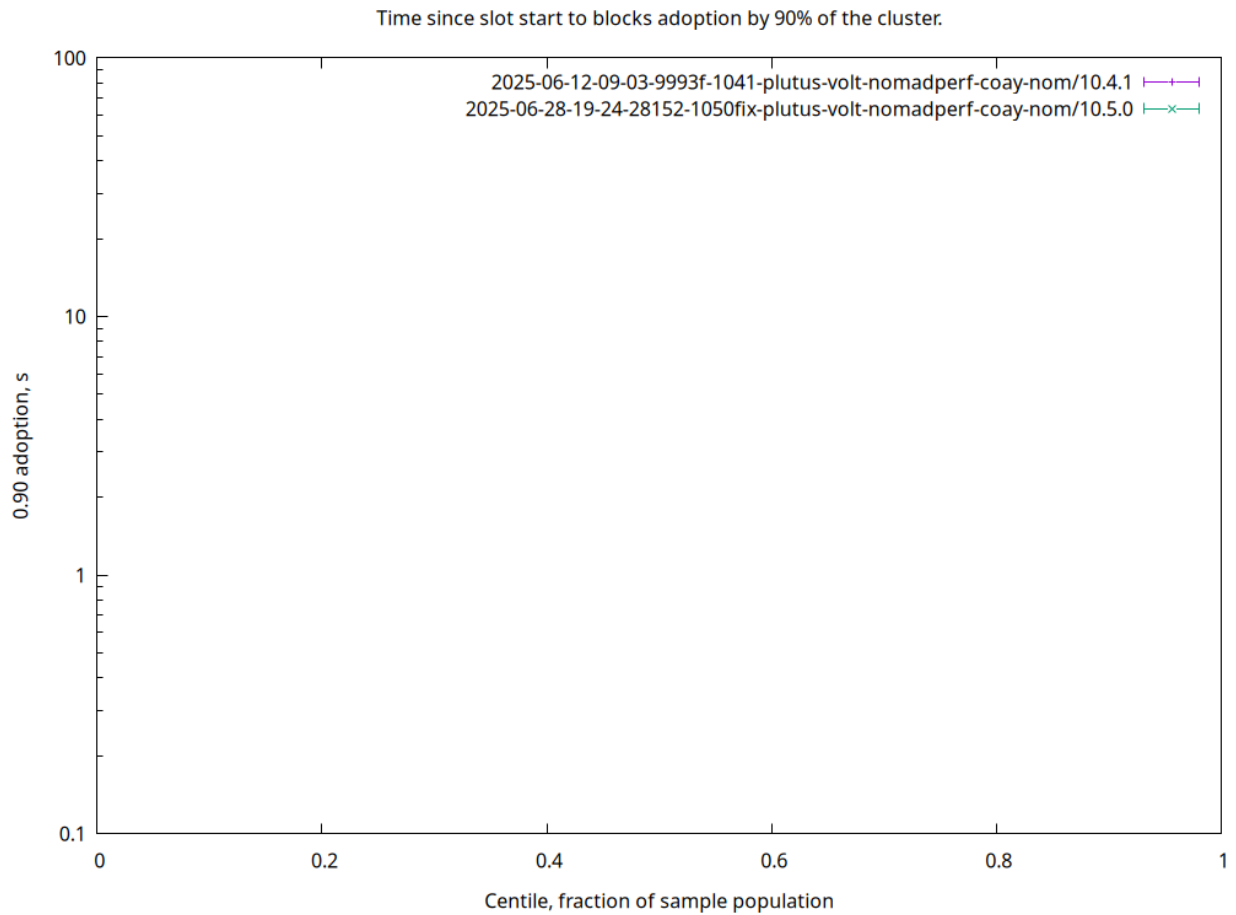
**0.50 adoption (cdf0.50)** Time since slot start to block's adoption by 50% of the cluster.



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

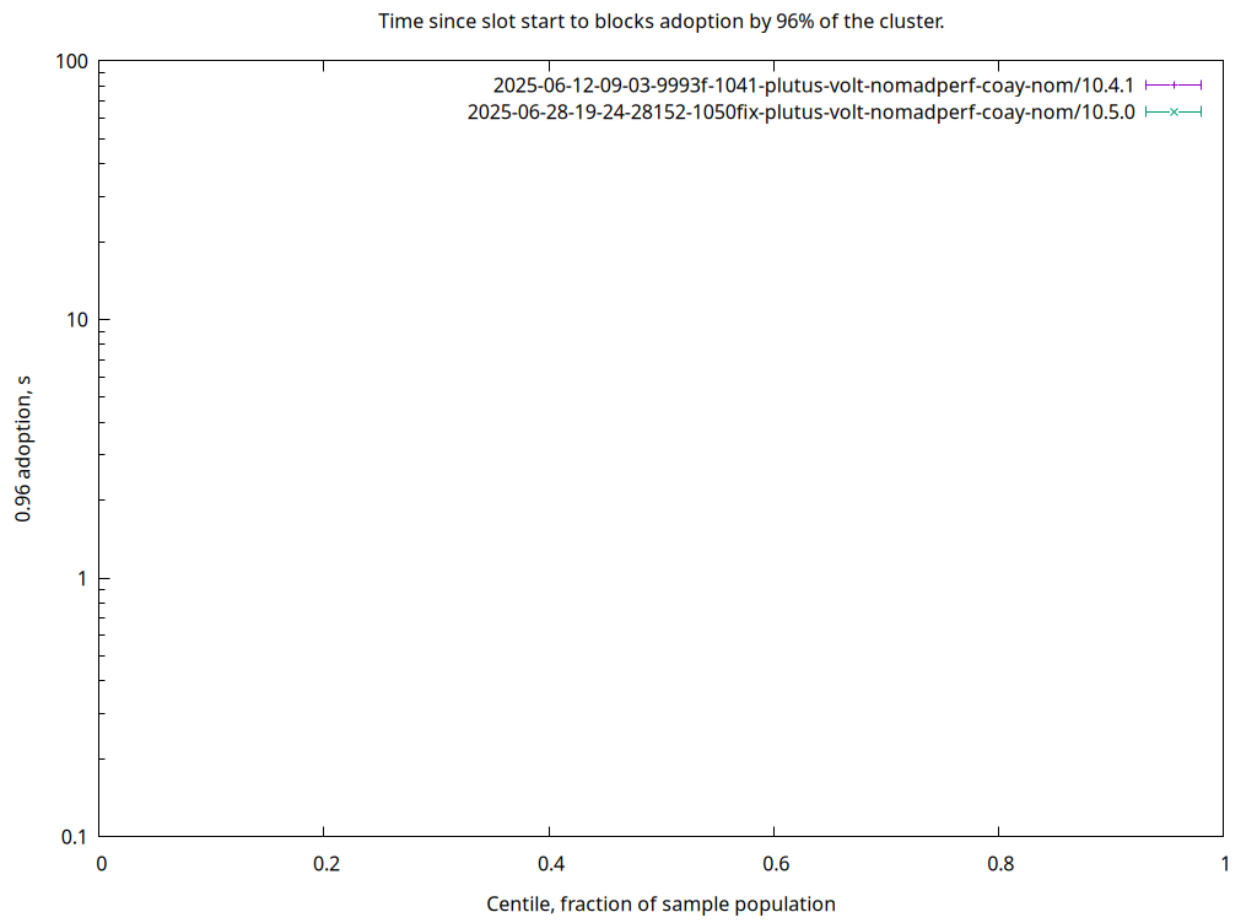


**0.90 adoption (cdf0.90)** Time since slot start to block's adoption by 90% of the cluster.



**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 (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

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

**Fetchd 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

**Fetchd 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