

10.0-pre against 9.2.0  
Plutus countdown loop workload

Michael Karg, Cardano Performance team

2024-10-21

# 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 10.0-pre (Conway) relative to 9.2.0 (Conway), under Plutus countdown loop workload.

	9.2.0	10.0-pre
Analysis date	2024-09-20	2024-10-21
Cluster system start date	2024-09-19	2024-10-20
Cluster system start time	16:14:18	08:10:42
Identifier	9.2.0	10.0
Run batch	9.2.0	10.0.0-pre
GHC version	8.10.7	8.10.7
cardano-node version	9.2.0	10.0
ouroboros-consensus version	0.20.1.0	0.21.0.0
ouroboros-network version	0.17.1.1	0.17.1.2
cardano-ledger-core version	1.14.0.0	1.15.0.0
plutus-core version	1.32.1.0	1.36.0.0
cardano-crypto version	1.1.2	1.1.2
cardano-prelude version	0.2.0.0	0.2.0.0
cardano-node git	341ea87	cdb45dd
ouroboros-consensus git	3da919f	358305b
ouroboros-network git	5dd745f	df3431f
cardano-ledger-core git	068c7a6	0ba8e73
plutus-core git	f2253e6	fb82893
cardano-crypto git	6568a5e	6568a5e
cardano-prelude git	273167c	273167c
Era	conway	conway
Delegation map size	1000000	1000000
Starting UTxO set 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 text lines emitted per host	845384.11538	852747.75
Log objects emitted per host	845354.11538	852717.76923
Log objects analysed per host	546292.75	545254.59615
Host run time, s	71849.6	71896.1
Host log line rate, Hz	11.766	11.860
Total log objects analysed	28407223	28353239
Run time, s	71854	71907
Analysed run duration, s	56017	56063
Run time efficiency	0.77	0.77
Node start spread, s	5.6496357	18.553902
Node stop spread, s	3.5745745	8.3869417
Perf analysis start spread, s	0	0
Perf analysis stop spread, s	3	8
Slots analysed	56014	56056
Blocks analysed	2728	2815
Blocks rejected	875	914

# Chapter 2

## Analysis

### 2.1 Resource Usage

	9.2.0	10.0-pre	$\Delta$	$\Delta\%$
Forge loop starts, #	0.99888	0.9987	-0.000	0
Process CPU usage, %	6.1258	5.8689	-0.257	-4
RTS GC CPU usage, %	0.6201	0.58608	-0.034	-5
RTS Mutator CPU usage, %	5.4956	5.2815	-0.214	-4
Major GCs, #	0.00091	0.00089	-0.000	0
Minor GCs, #	1.5943	1.3958	-0.199	-12
Kernel RSS, MB	9032.5	9023.9	-8.600	0
RTS heap size, MB	8978.5	8971.7	-6.800	0
RTS live GC dataset, MB	3601.5	3704.2	102.700	3
RTS alloc rate, MB/s	48.915	43.437	-5.478	-11
Filesystem reads, KB/s	2e-05	0.00011	0.000	0
Filesystem writes, KB/s	212.35	209.62	-2.730	-1
CPU 85% spans, slots	11.703	11.143	-0.560	-5
Sample count	(291>)	(291>)		

### 2.2 Anomaly control

	9.2.0	10.0-pre	$\Delta$	$\Delta\%$
Blocks per host, blocks	71.442	73.788	2.346	3
Filtered to chained block ratio, /	0.75691	0.7553	-0.002	0
Chained to forged block ratio, /	0.96986	0.9718	0.002	0
Height & slot battles, blocks	0.00036	0.00142	0.001	278
Block size, B	2996.0	2996.0	0.000	0
Sample count	(52)	(52)		

## 2.3 Forging

	9.2.0	10.0-pre	$\Delta$	$\Delta\%$
Started forge loop iteration, s	0.00083	0.00089	0.000	0
Acquired block context, s	0.02284	0.02318	0.000	0
Acquired ledger state, s	6e-05	6e-05	0.000	0
Acquired ledger view, s	2e-05	2e-05	0.000	0
Leadership check duration, s	0.00041	0.00039	-0.000	0
Ledger ticking, s	0.0259	0.02538	-0.001	-4
Mempool snapshotting, s	0.07192	0.00226	-0.070	-97
Leadership to forged, s	0.00046	0.00016	-0.000	0
Forged to announced, s	0.00058	0.00061	0.000	0
Forged to sending, s	0.00554	0.00543	-0.000	0
Forged to self-adopted, s	0.06061	0.05768	-0.003	-5
Slot start to announced, s	0.12305	0.053	-0.070	-57
Sample count	(2728)	(2815)		

## 2.4 Individual peer propagation

	9.2.0	10.0-pre	$\Delta$	$\Delta\%$
First peer notice, s	0.12527	0.05491	-0.070	-56
First peer fetch, s	0.13196	0.06058	-0.071	-54
Notice to fetch request, s	0.00121	0.00121	0.000	0
Fetch duration, s	0.12345	0.12341	-0.000	0
Fetches to announced, s	2e-05	1e-05	-0.000	0
Fetches to sending, s	0.04255	0.04274	0.000	0
Fetches to adopted, s	0.06008	0.05916	-0.001	-2
Sample count	(2728)	(2815)		

## 2.5 End-to-end propagation

	9.2.0	10.0-pre	$\Delta$	$\Delta\%$
0.50 adoption, s	0.40009	0.32613	-0.074	-18
0.80 adoption, s	0.55943	0.48736	-0.072	-13
0.90 adoption, s	0.57228	0.50189	-0.070	-12
0.92 adoption, s	0.5763	0.50653	-0.070	-12
0.94 adoption, s	0.57955	0.51582	-0.064	-11
0.96 adoption, s	0.58373	0.5235	-0.060	-10
0.98 adoption, s	0.58885	0.53392	-0.055	-9
1.00 adoption, s	0.61958	0.55951	-0.060	-10
Sample count	(2728)	(2815)		

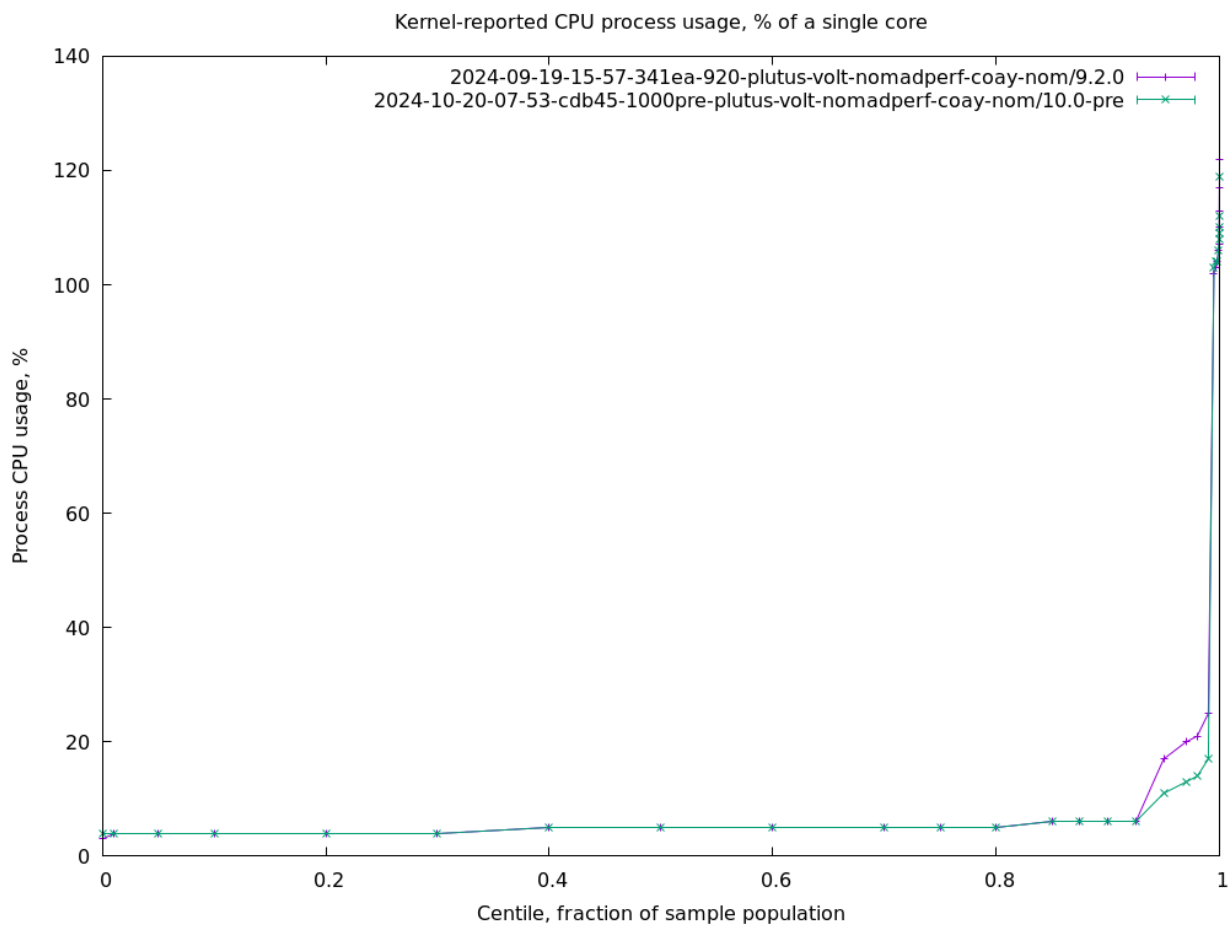
## Part I

# Appendix A: charts

# Chapter 3

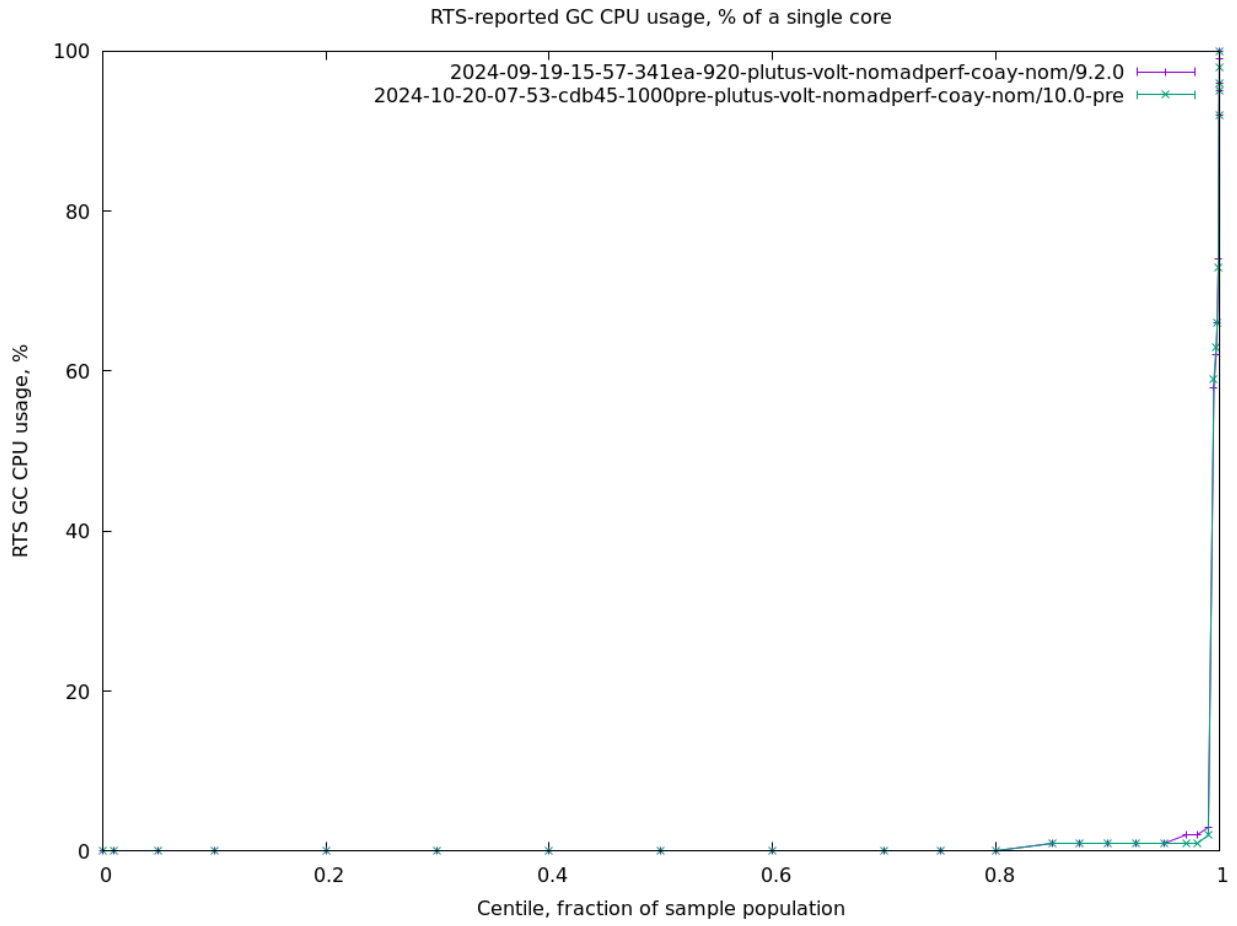
## Cluster performance charts

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

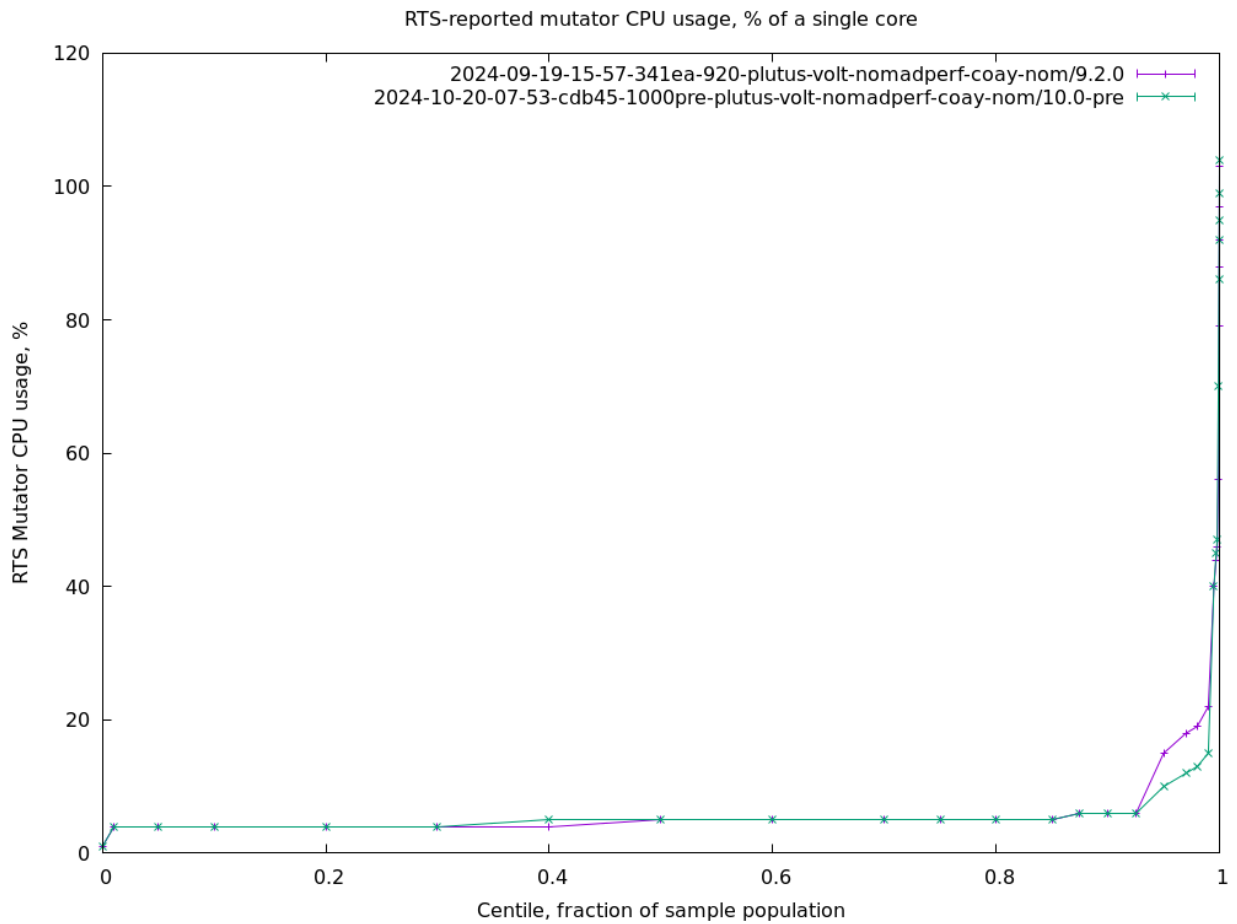


**RTS GC CPU usage (CentiGC)** RTS-reported GC CPU usage, % of a single core

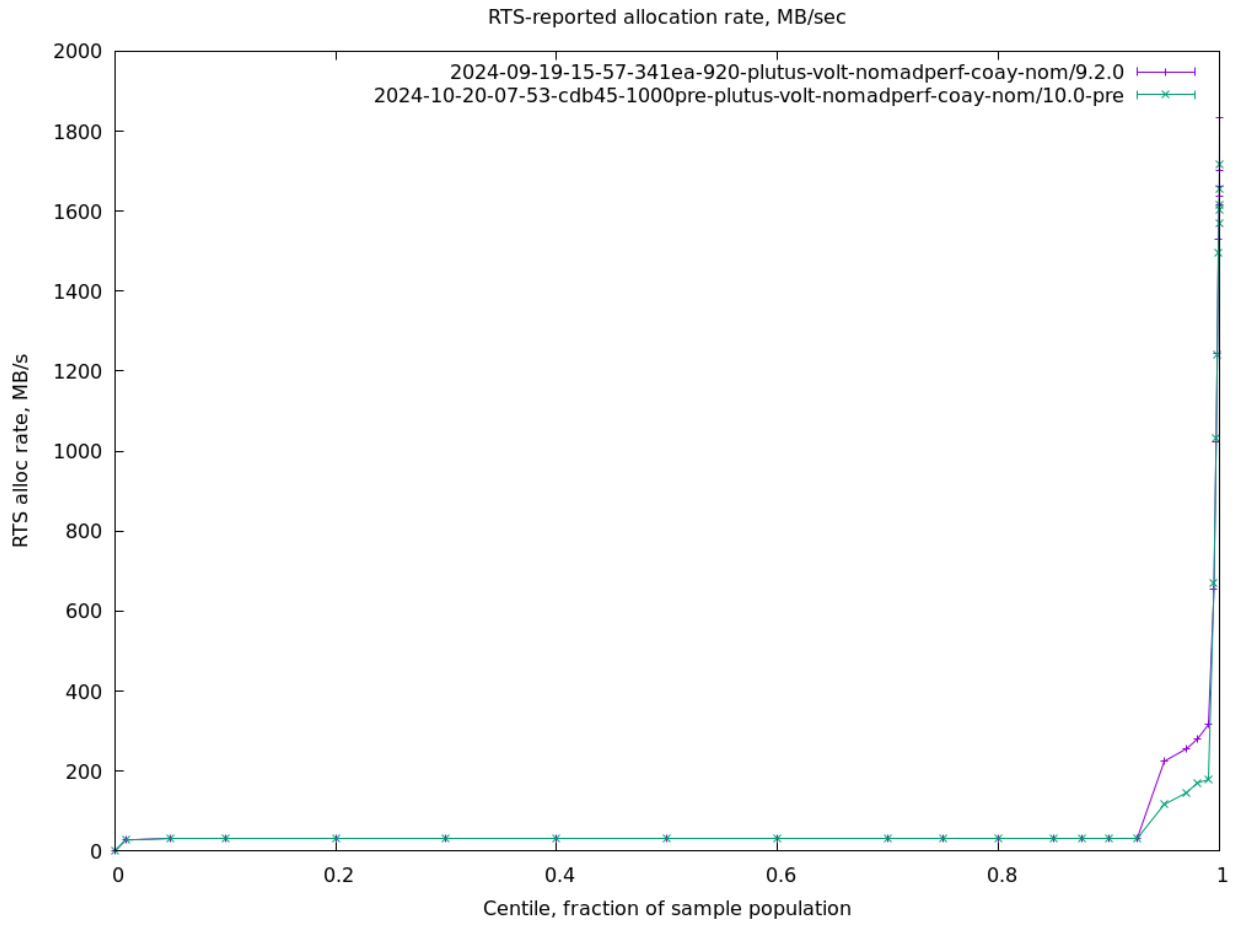




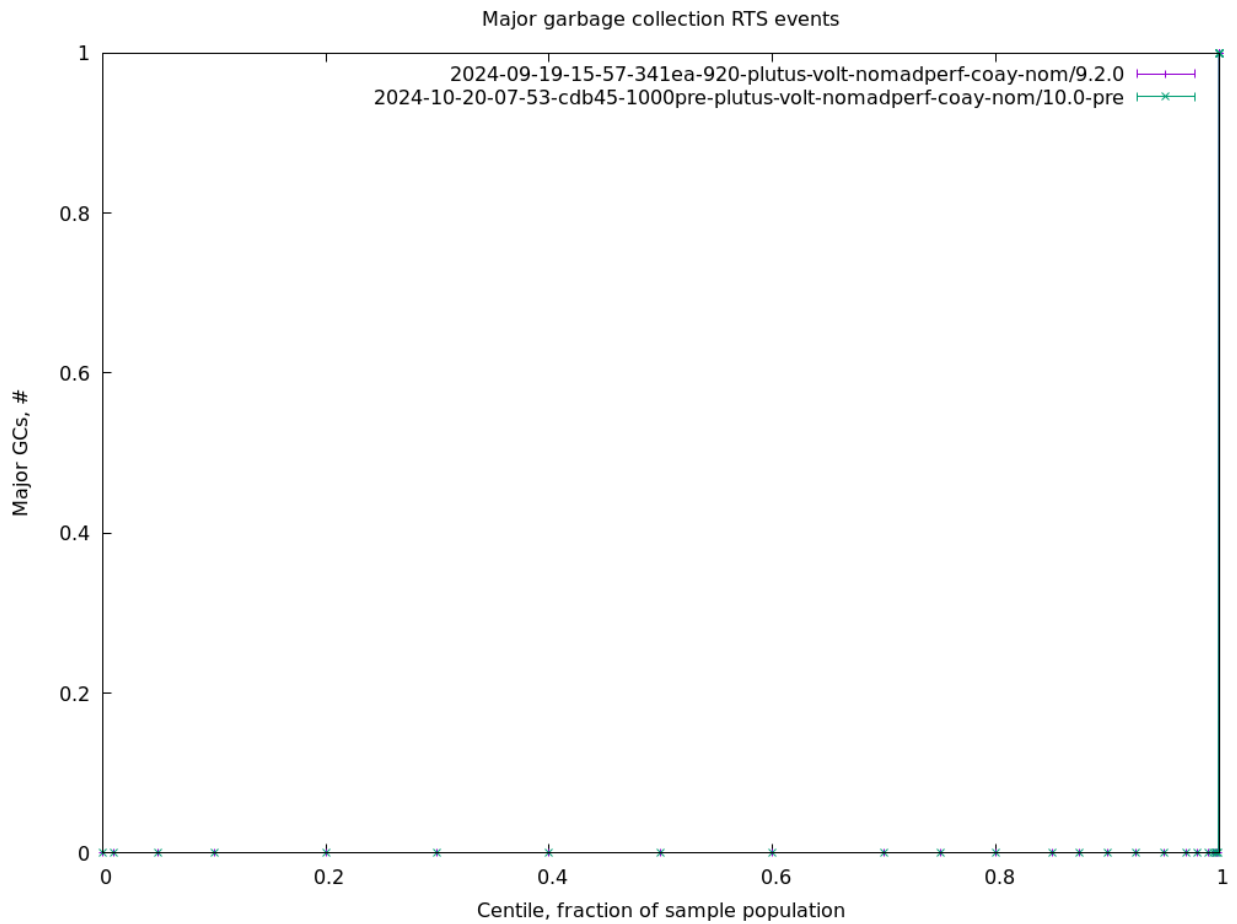
**RTS Mutator CPU usage (CentiMut)** RTS-reported mutator CPU usage, % of a single core



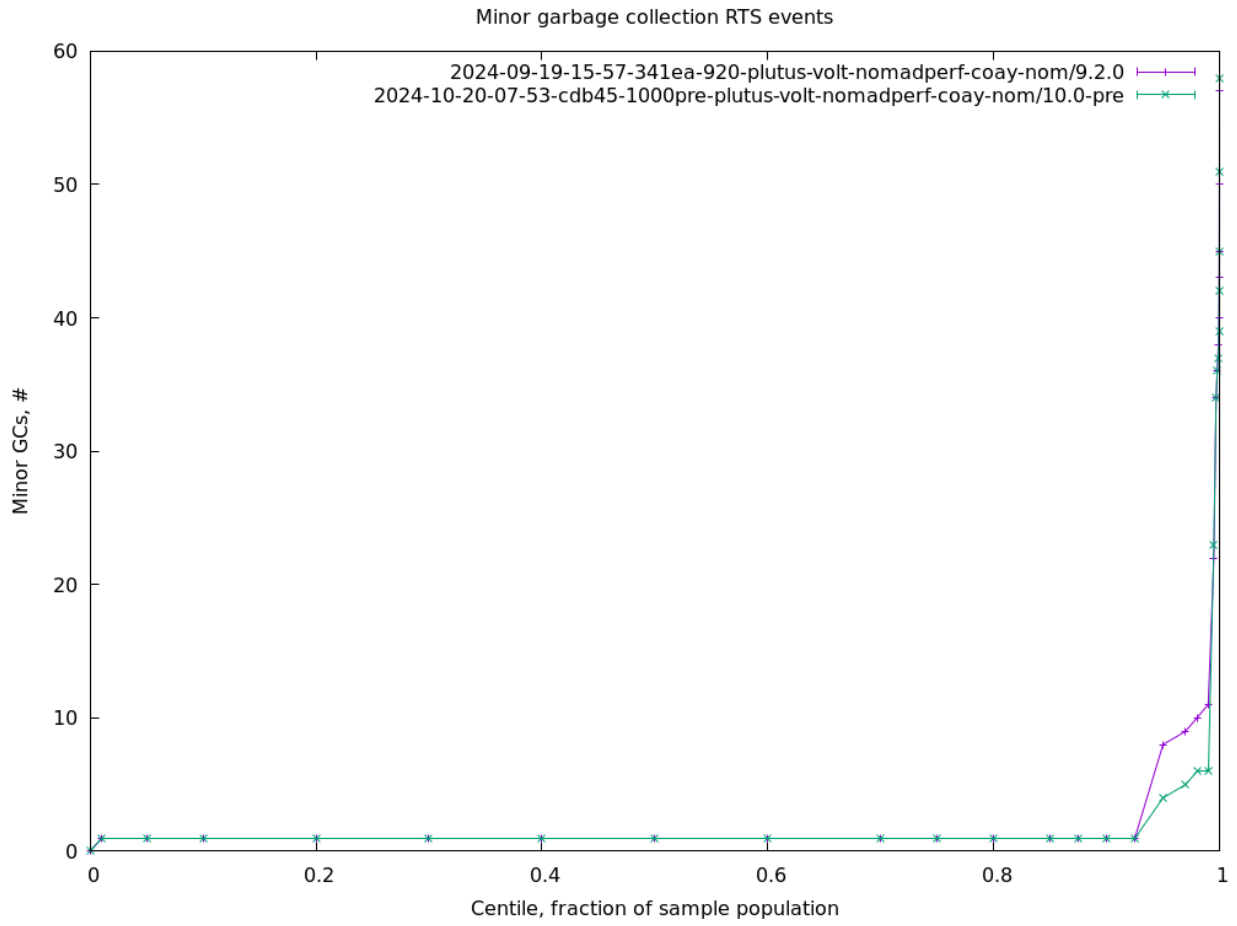
**RTS alloc rate (Alloc)** RTS-reported allocation rate, MB/sec



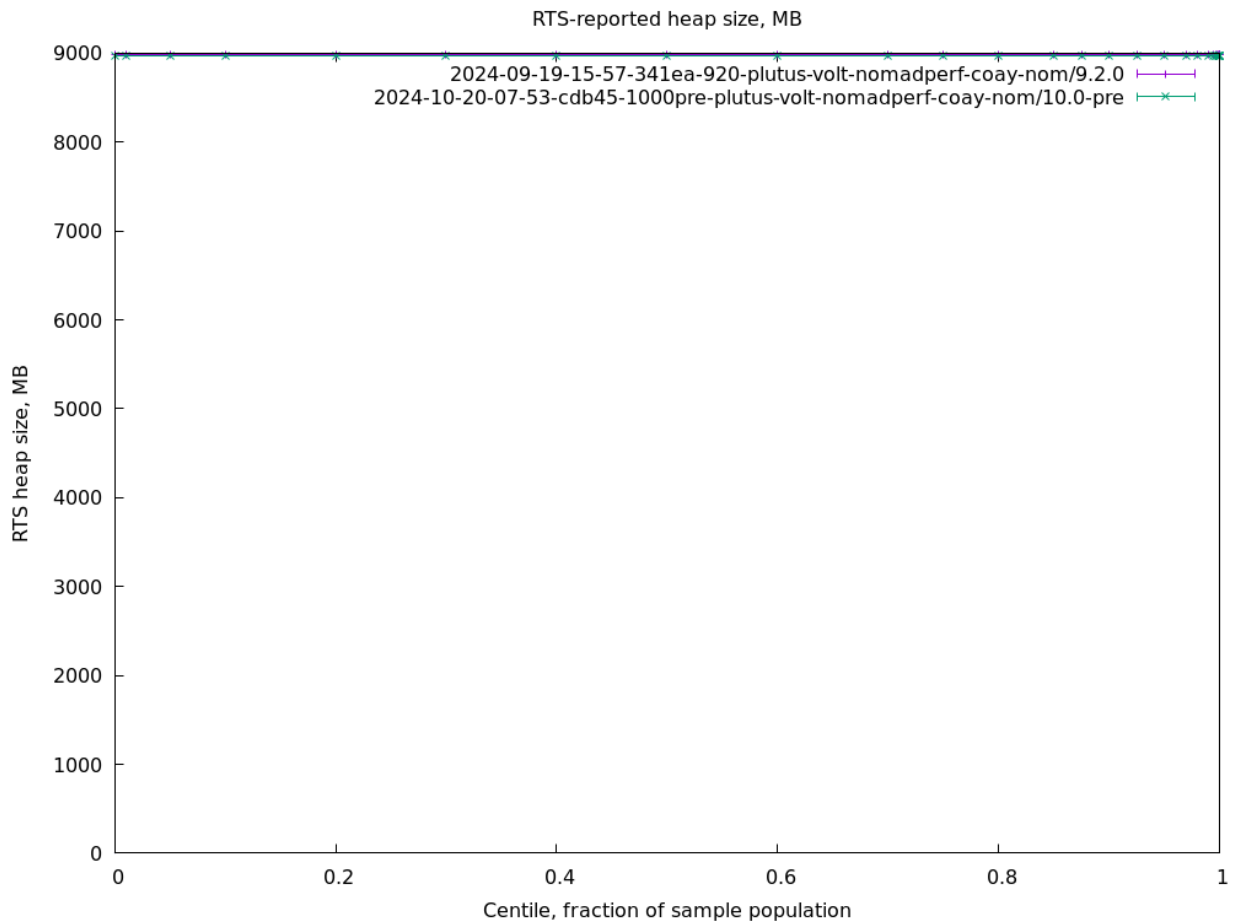
Major GCs (GcsMajor) Major garbage collection RTS events



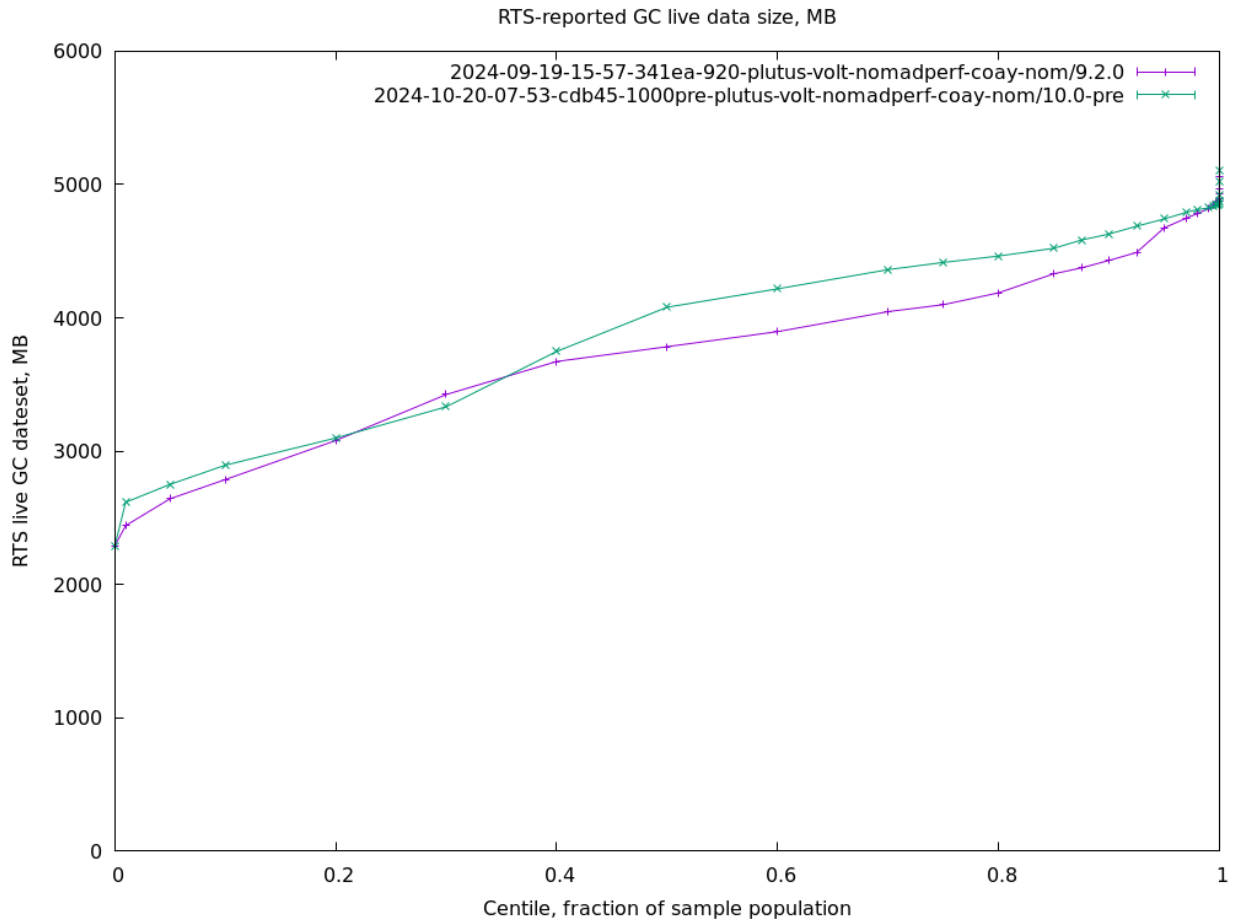
Minor GCs (GcsMinor) Minor garbage collection RTS events



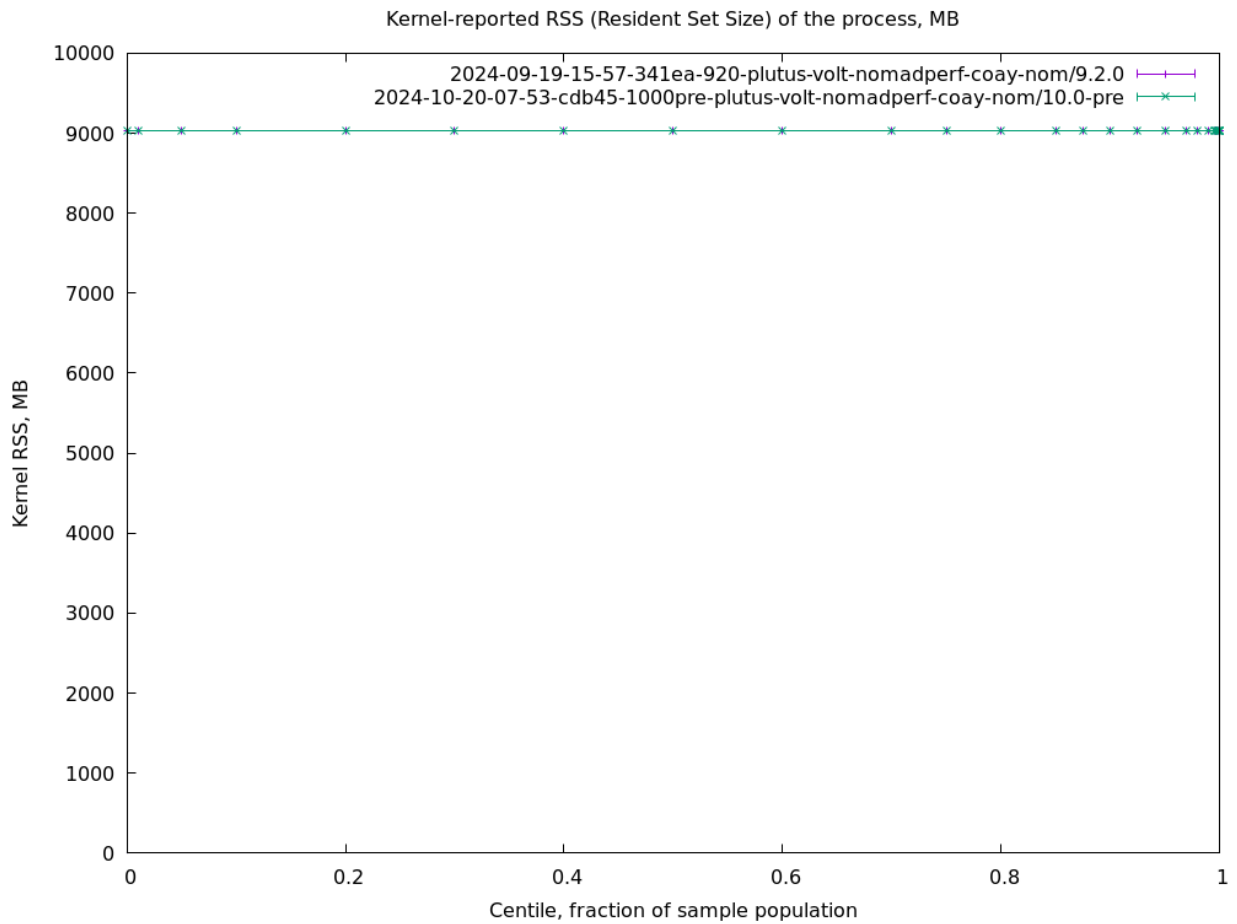
RTS heap size (Heap) RTS-reported heap size, MB



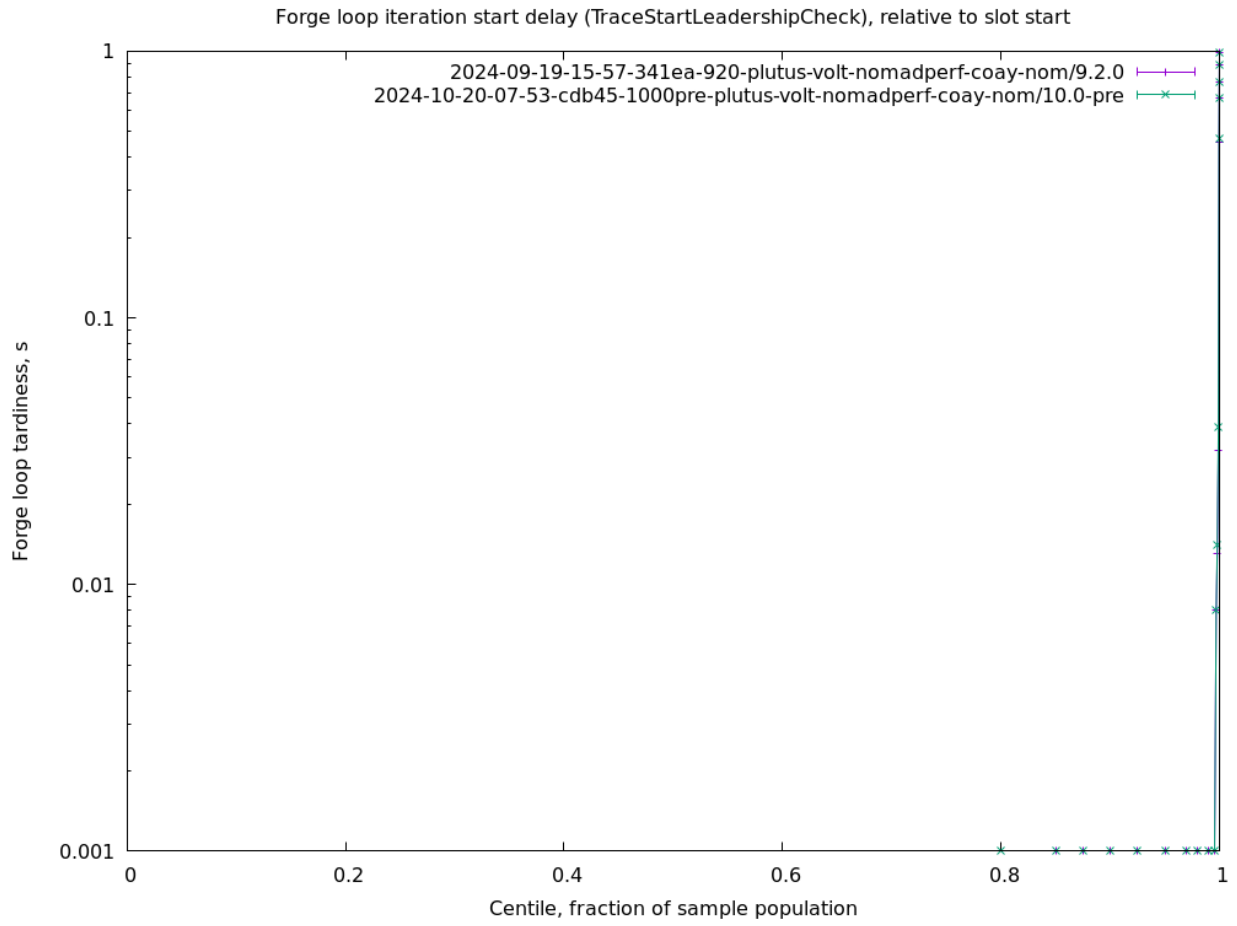
RTS live GC dataset (Live) RTS-reported GC live data size, MB



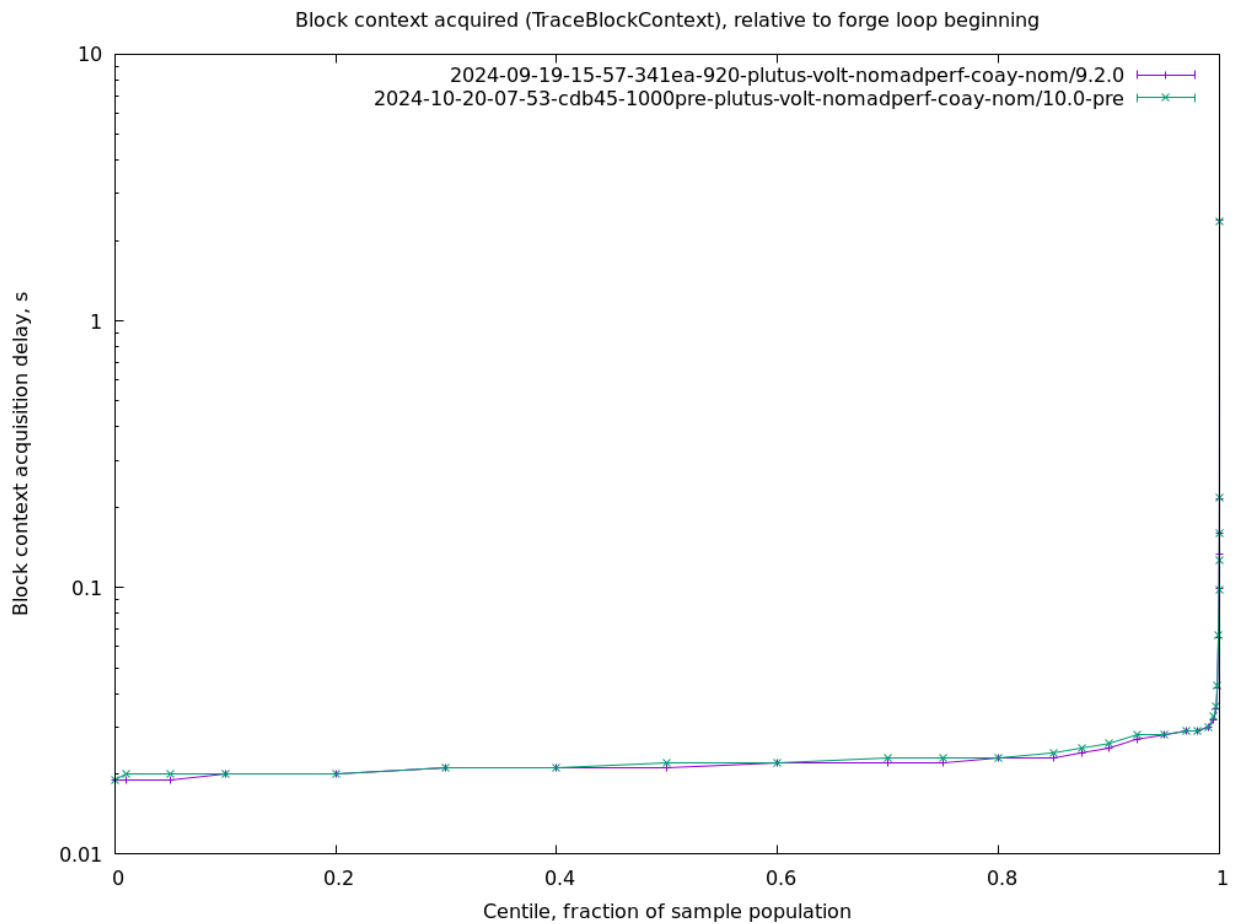
**Kernel RSS (RSS)** Kernel-reported RSS (Resident Set Size) of the process, MB



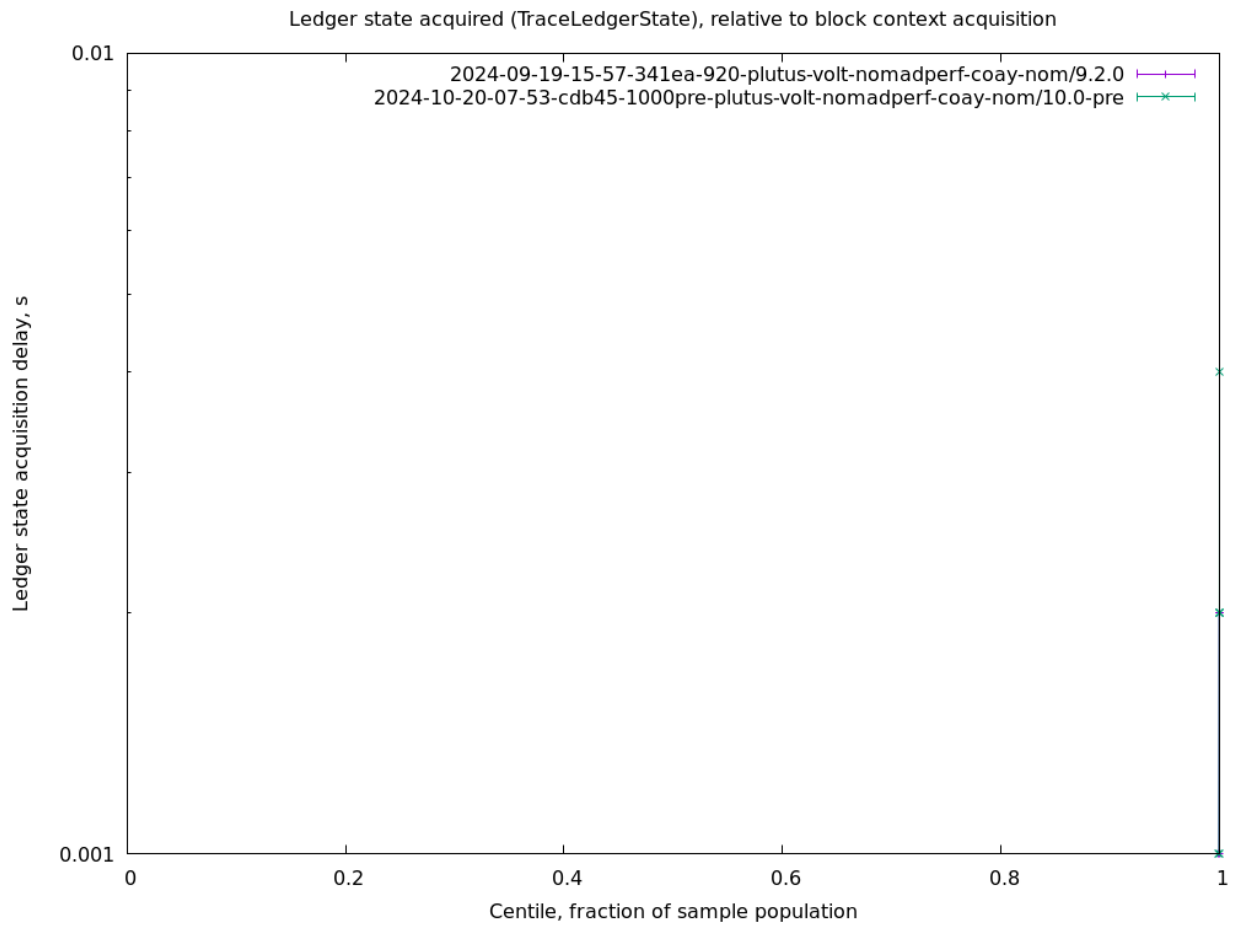
**Forge loop tardiness (cdfStarted)** Forge loop iteration start delay (TraceStartLeadershipCheck), relative to slot start



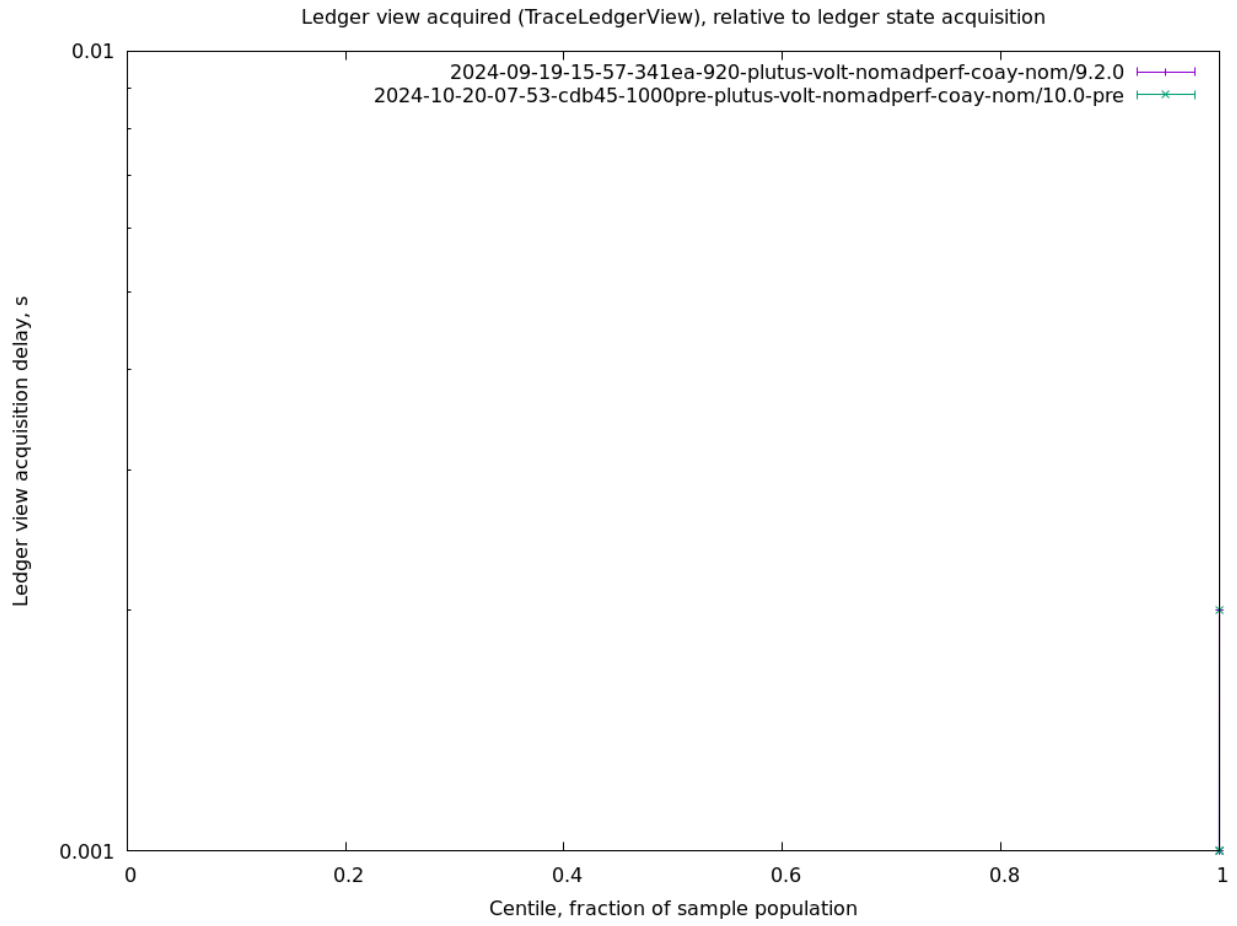
**Block context acquisition delay (cdfBlkCtx)** Block context acquired (TraceBlockContext), relative to forge loop beginning



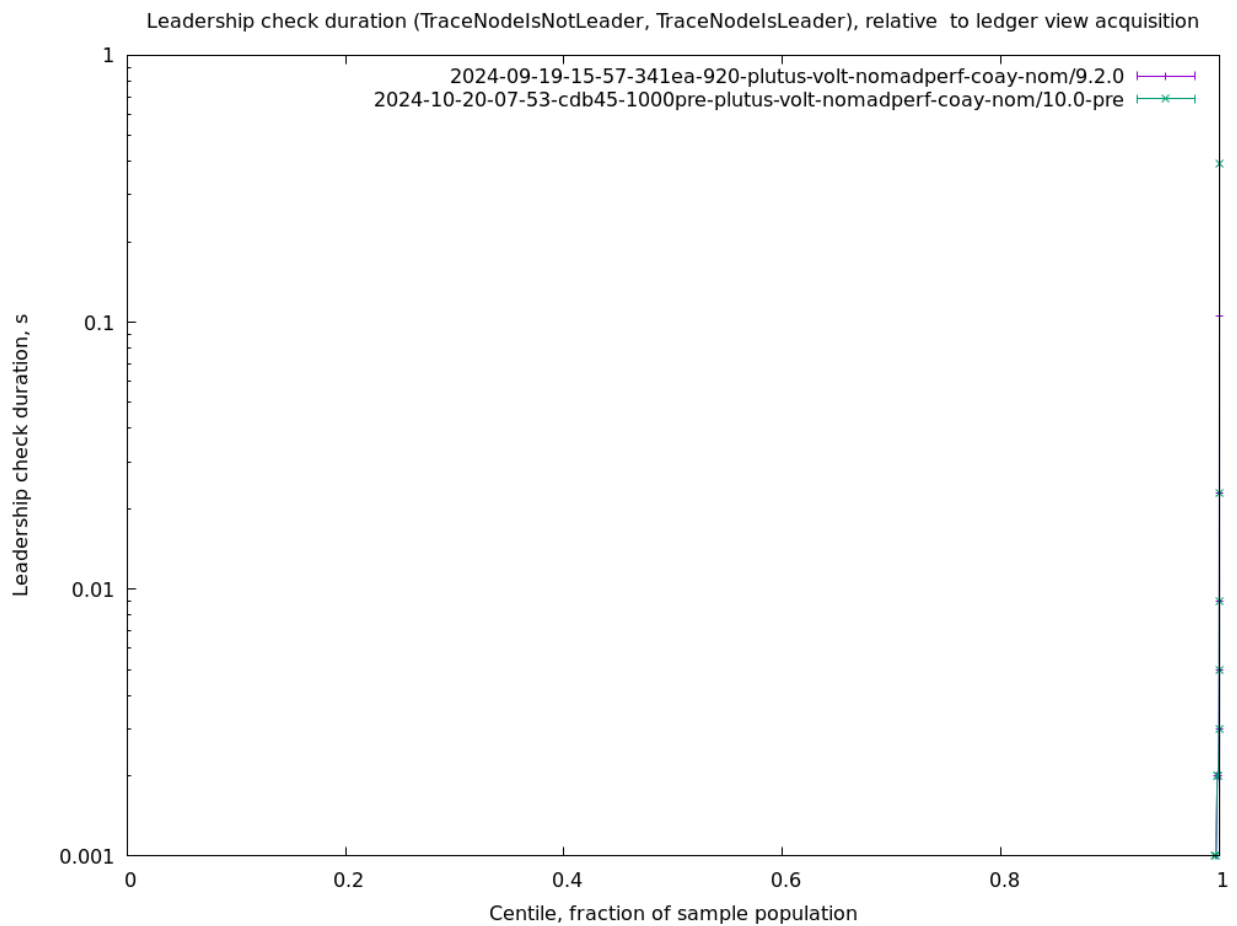
**Ledger state acquisition delay (cdfLgrState)** Ledger state acquired (TraceLedgerState), relative to block context acquisition



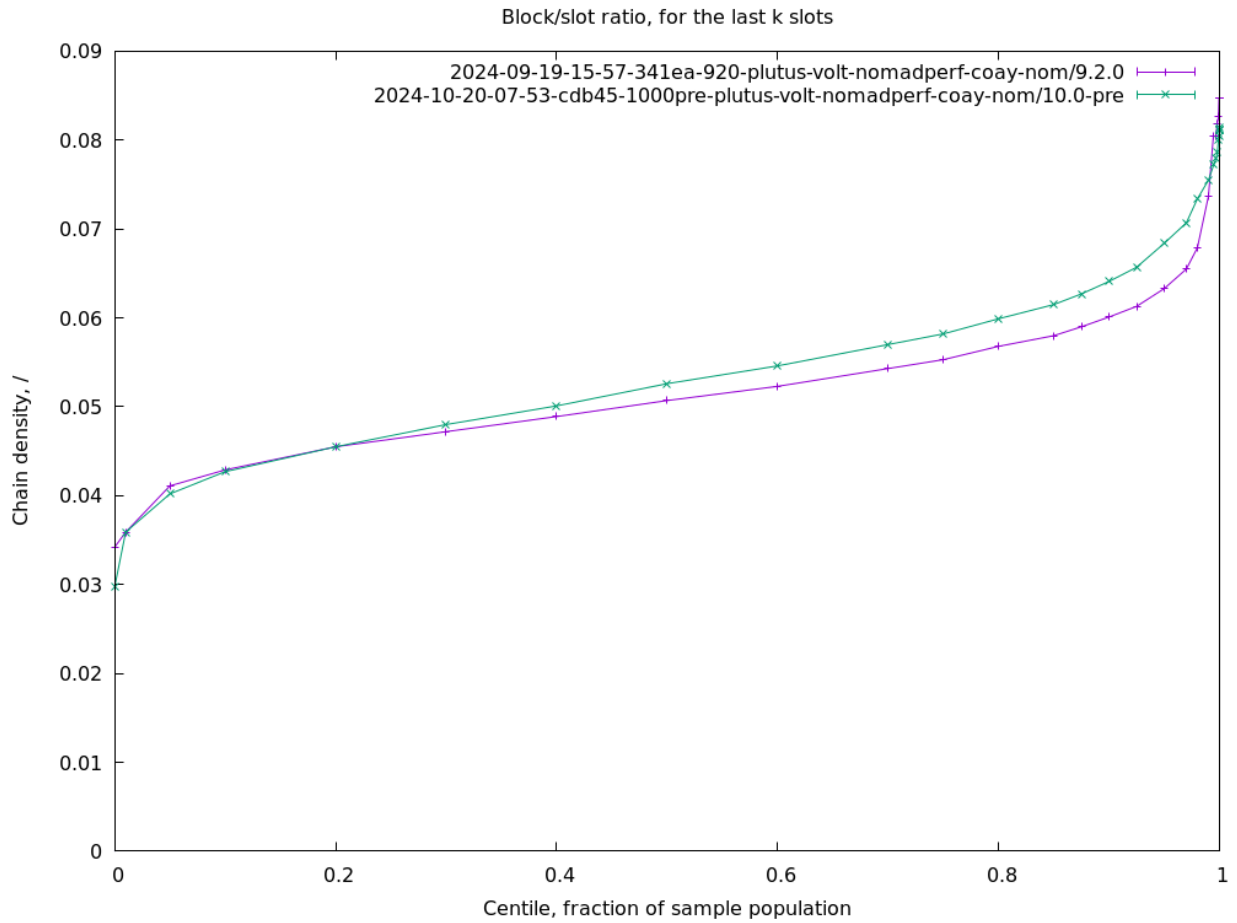
**Ledger view acquisition delay (cdfLgrView)** Ledger view acquired (TraceLedgerView), relative to ledger state acquisition



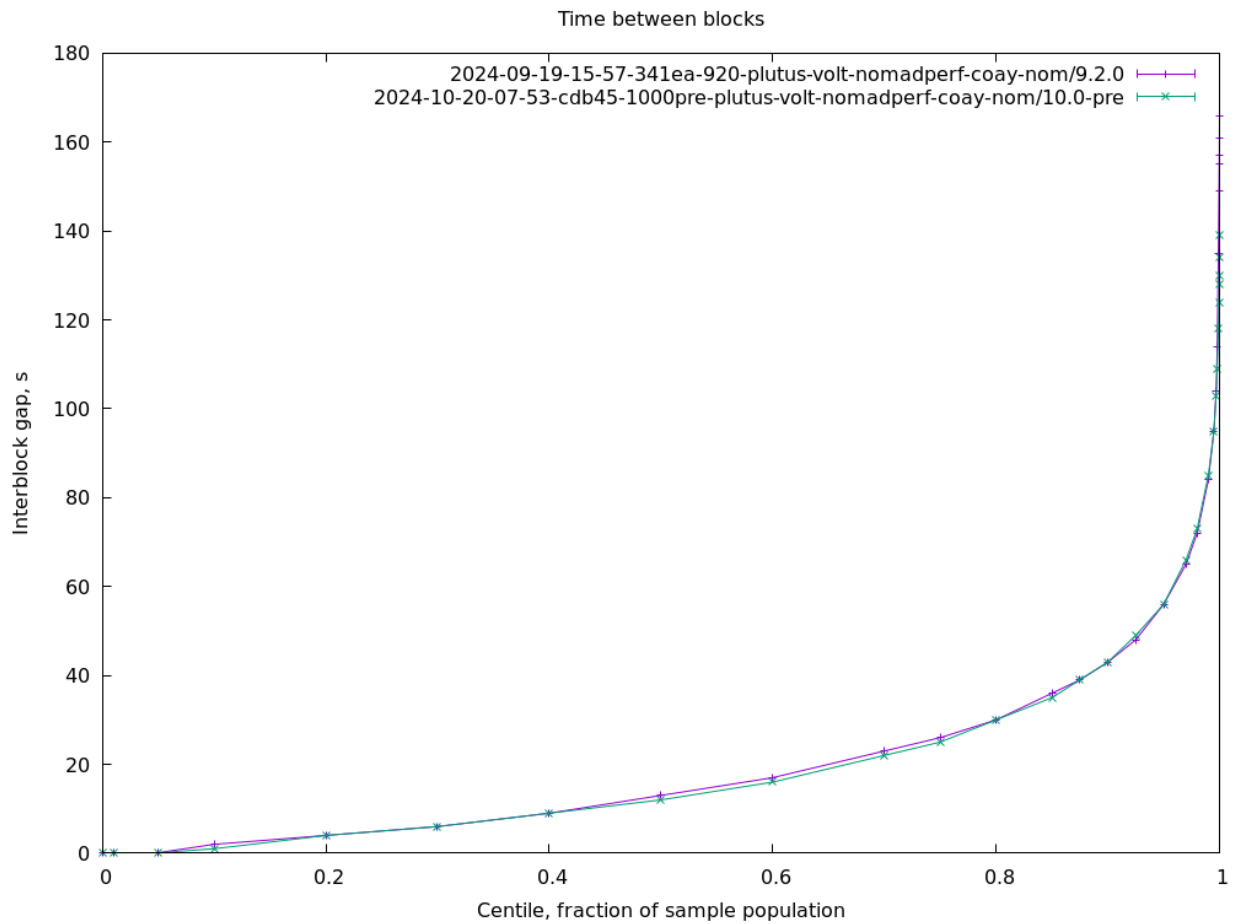
**Leadership check duration (cdfLeading)** Leadership check duration (TraceNodeIsNotLeader, TraceNodeIsLeader), relative to ledger view acquisition



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

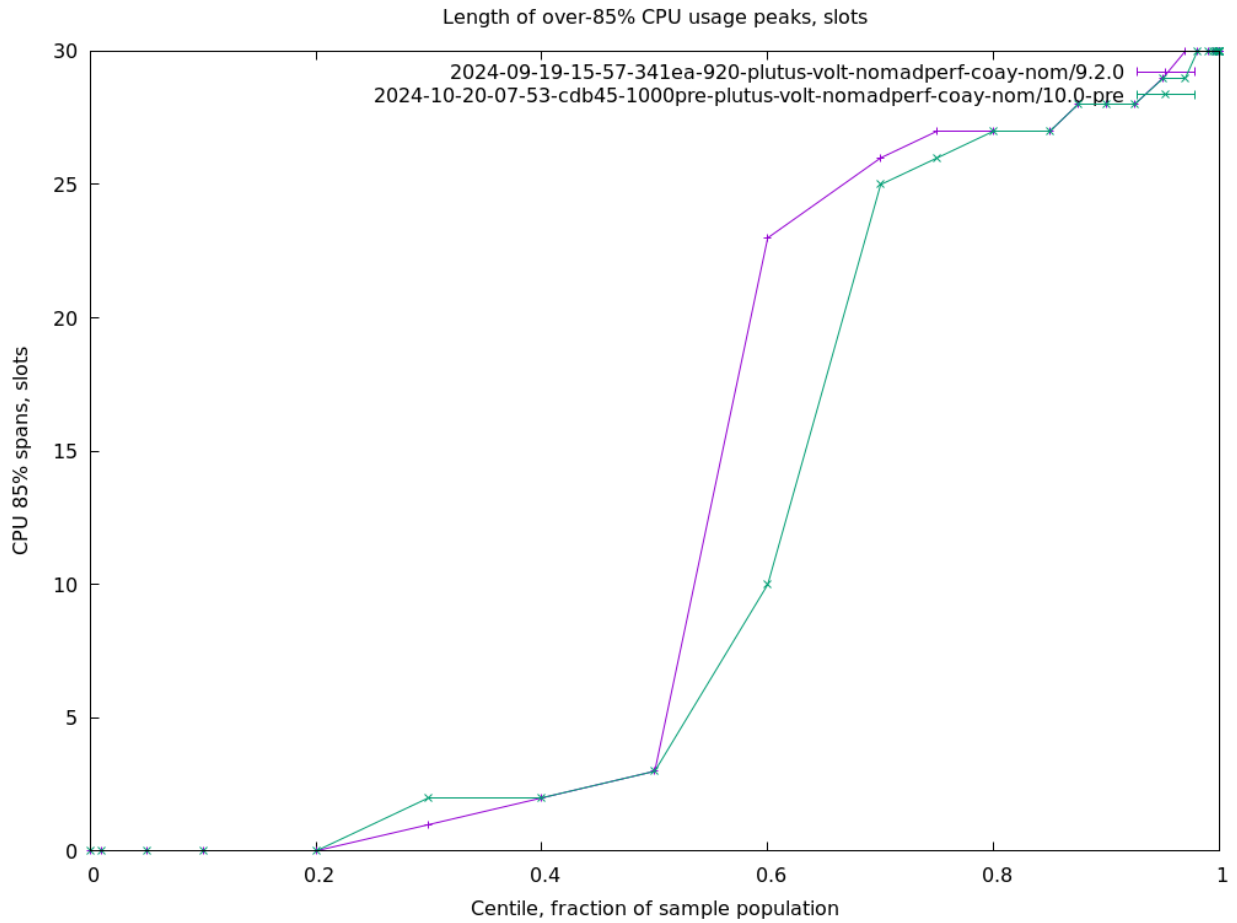


Interblock gap (cdfBlockGap) Time between blocks

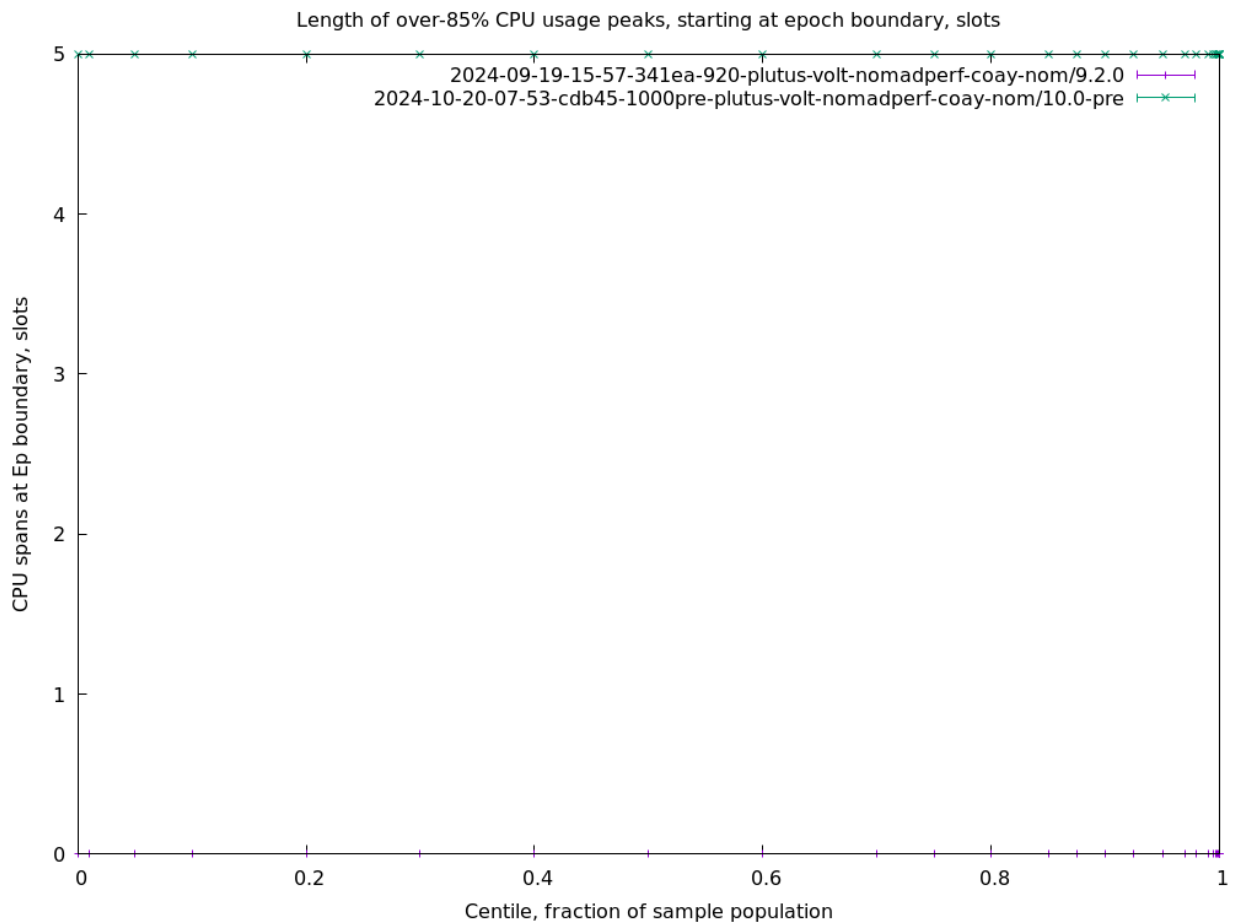


CPU 85% spans (cdfSpanLensCpu) Length of over-85% CPU usage peaks, slots

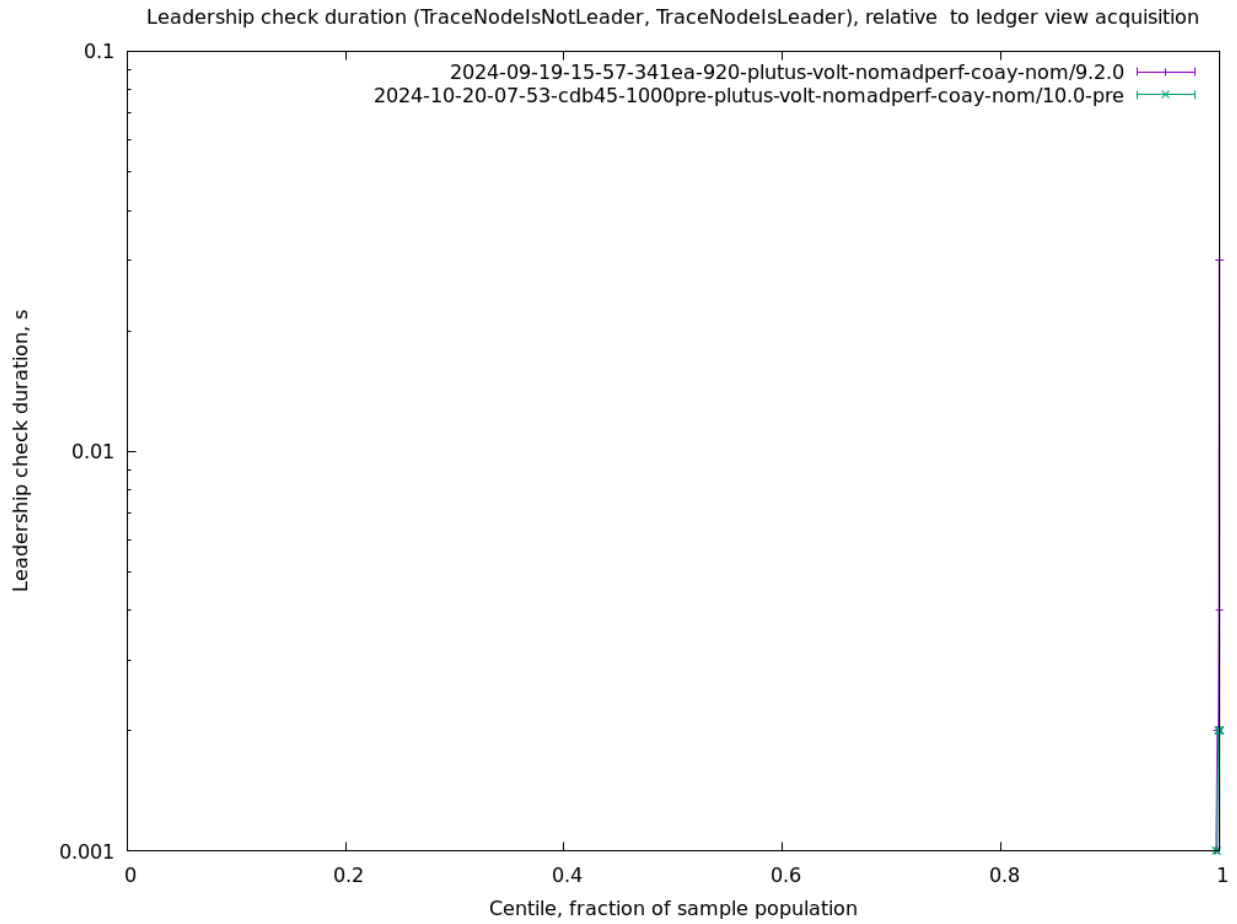




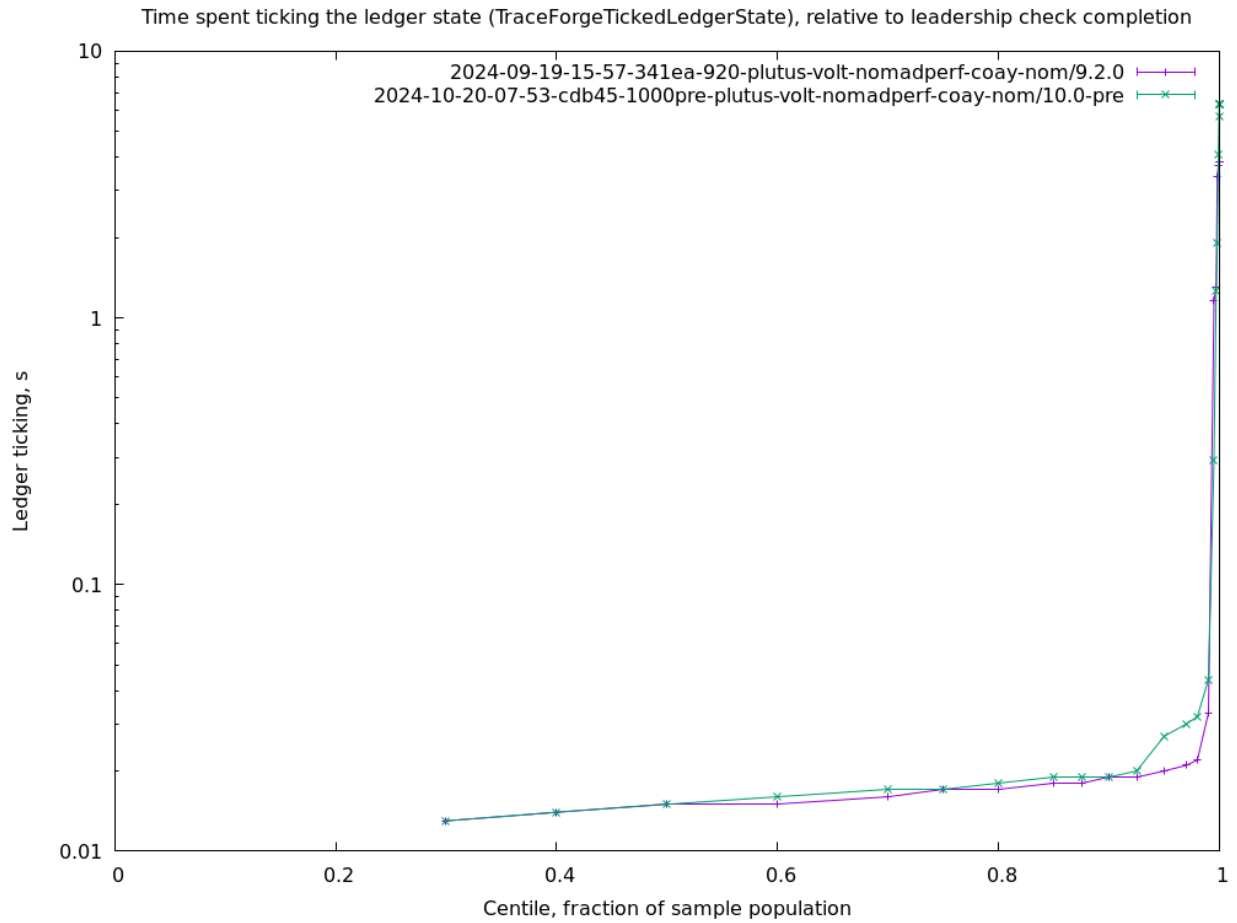
CPU spans at Ep boundary (cdfSpanLensCpuEpoch) Length of over-85% CPU usage peaks, starting at epoch boundary, slots



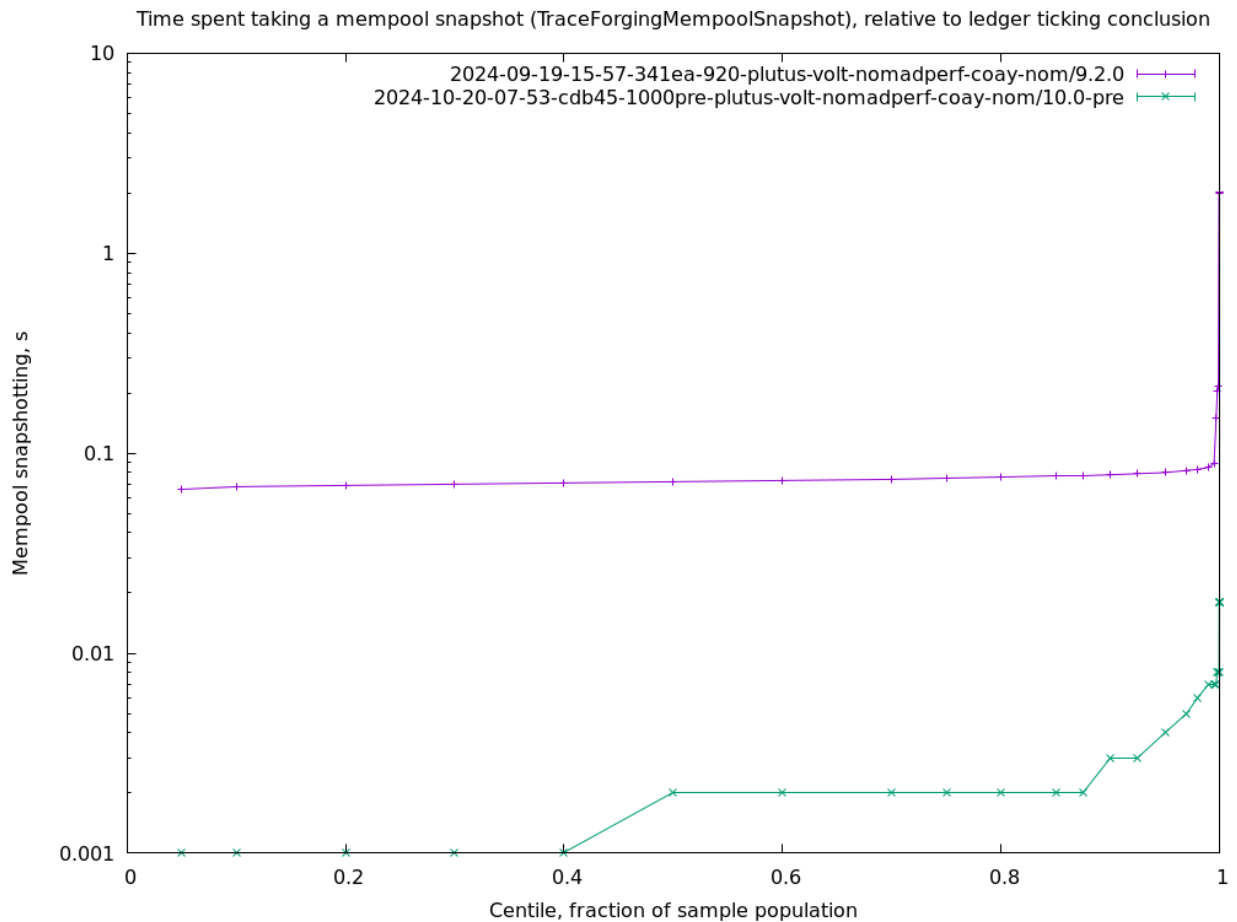
**Leadership check duration (cdfForgerLead)** Leadership check duration (TraceNodeIsNotLeader, TraceNodeIsLeader), relative to ledger view acquisition



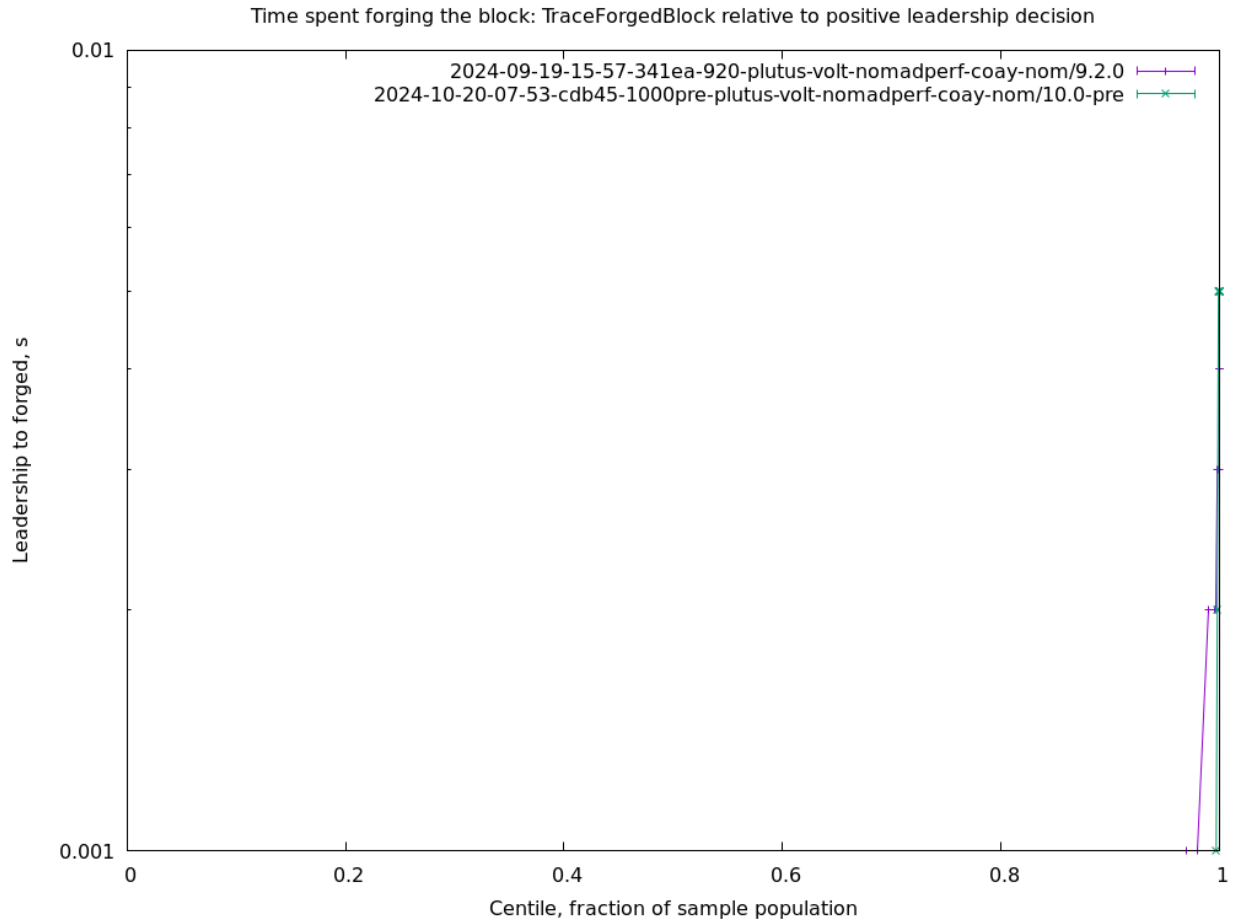
**Ledger ticking (cdfForgerTicked)** Time spent ticking the ledger state (TraceForgeTickedLedgerState), relative to leadership check completion



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

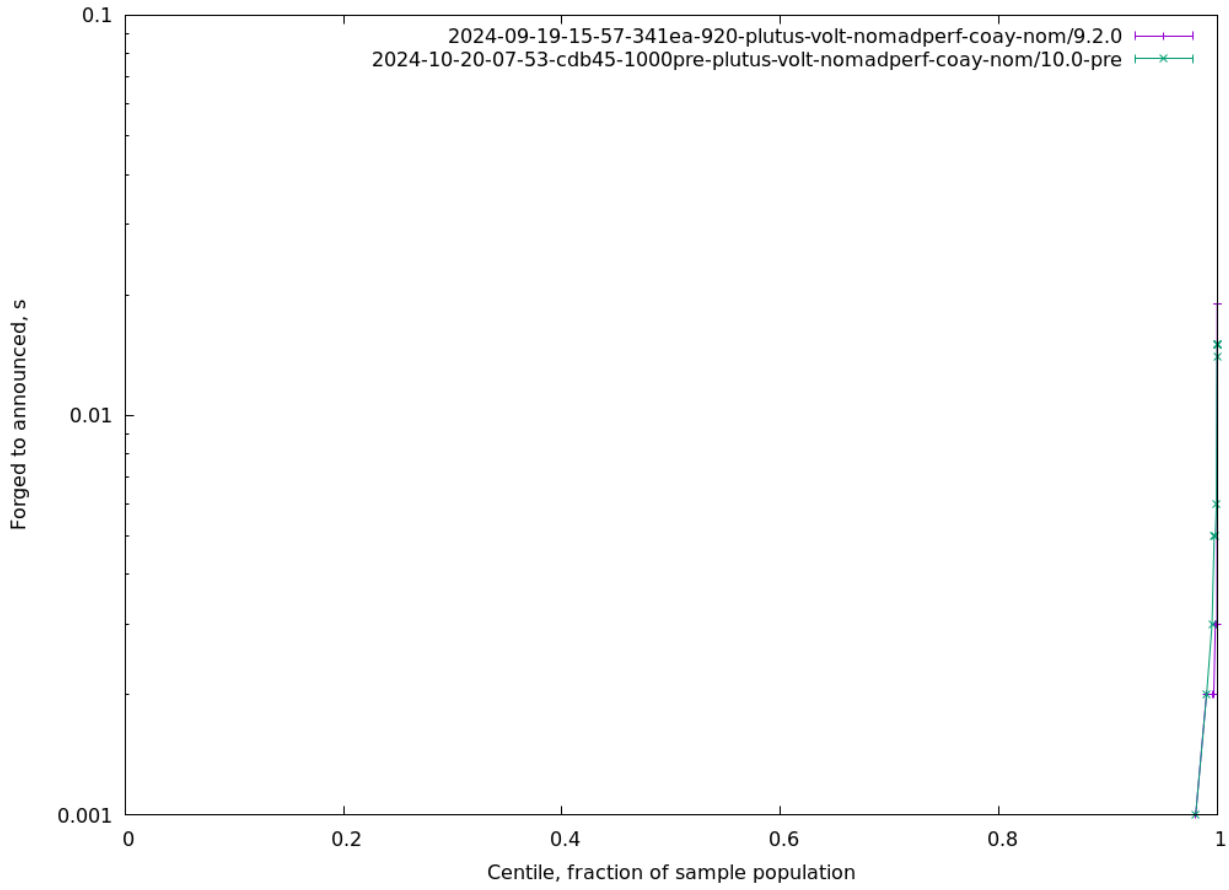


**Leadership to forged (cdfForgerForge)** Time spent forging the block: TraceForgedBlock relative to positive leadership decision

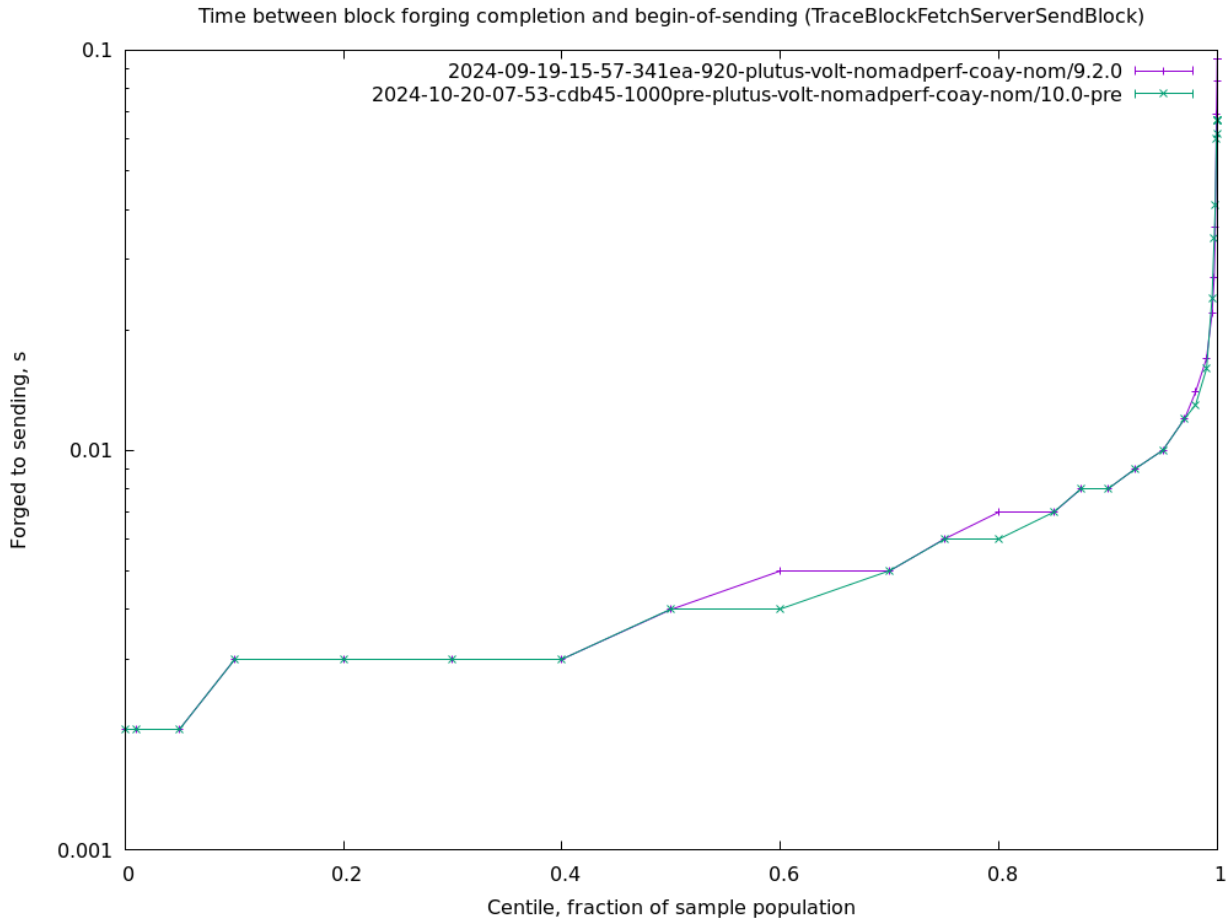


**Forged to announced (cdfForgerAnnounce)** Time between block forging completion and header announcement (ChainSyncServerEvent.TraceChainSyncServerRead.AddBlock)

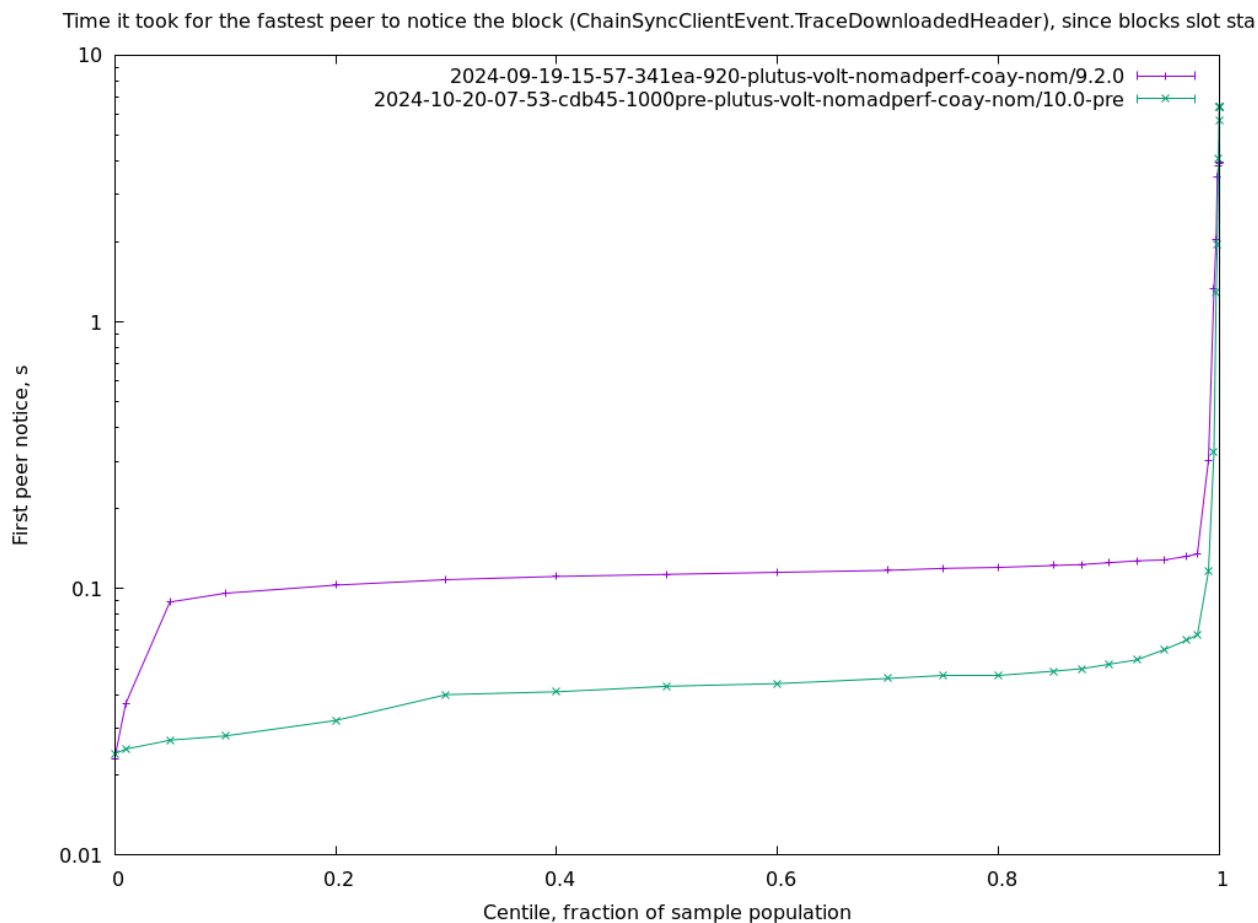
Time between block forging completion and header announcement (ChainSyncServerEvent.TraceChainSyncServerRead.Add)



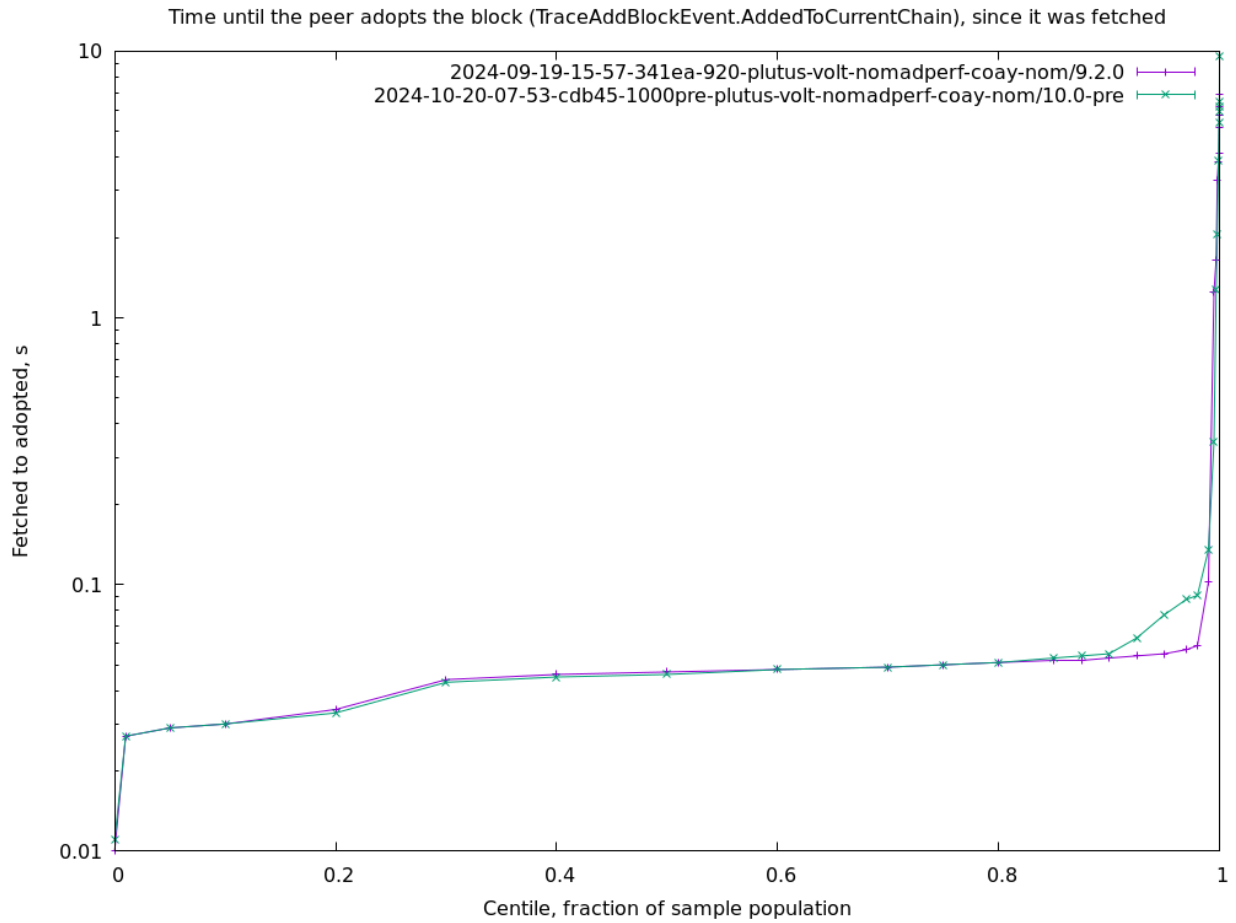
**Forged to sending (cdfForgerSend)** Time between block forging completion and begin-of-sending (TraceBlockFetchServerSendBlock)



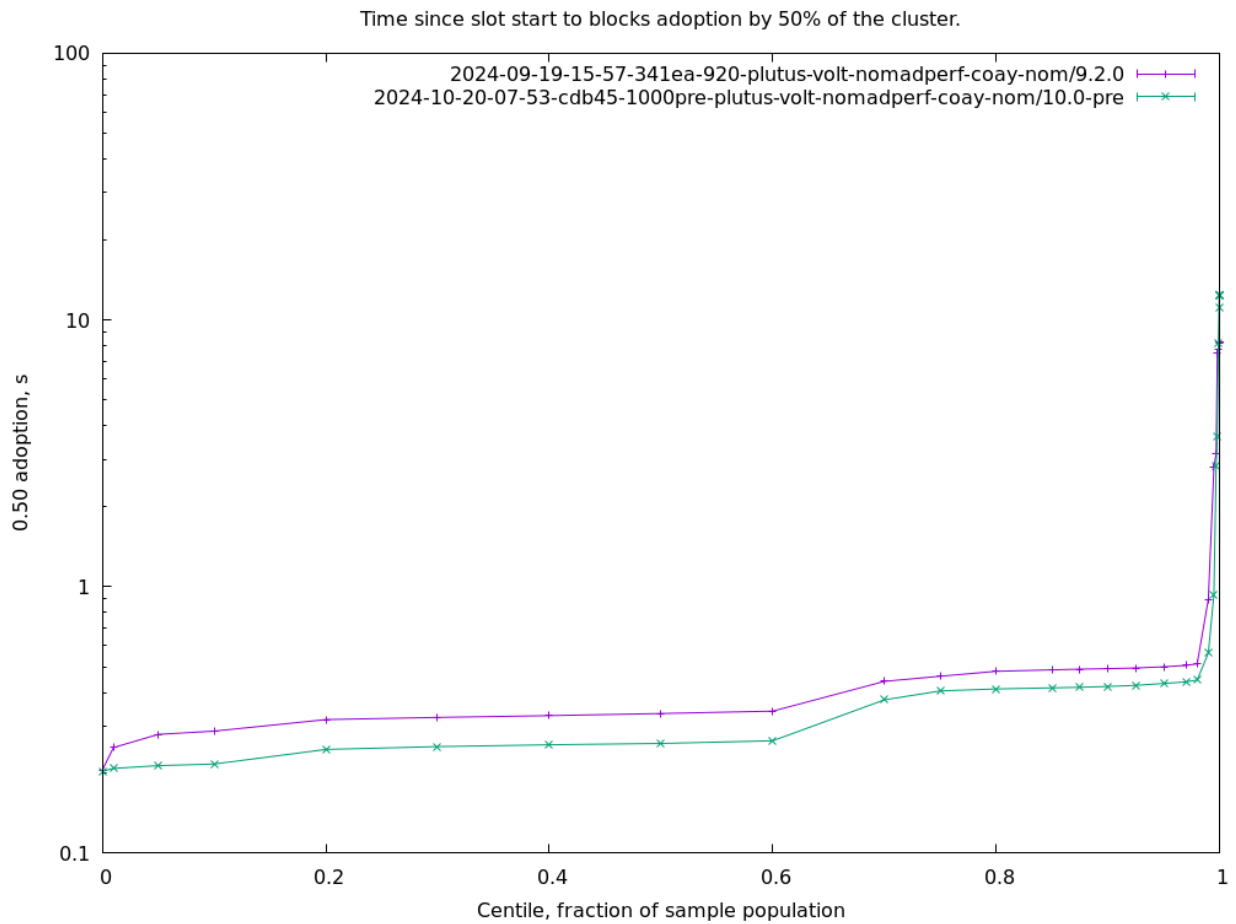
**First peer notice (cdfPeerNoticeFirst)** Time it took for the fastest peer to notice the block (ChainSyncClientEvent.TraceDownloadedHeader), since block's slot start



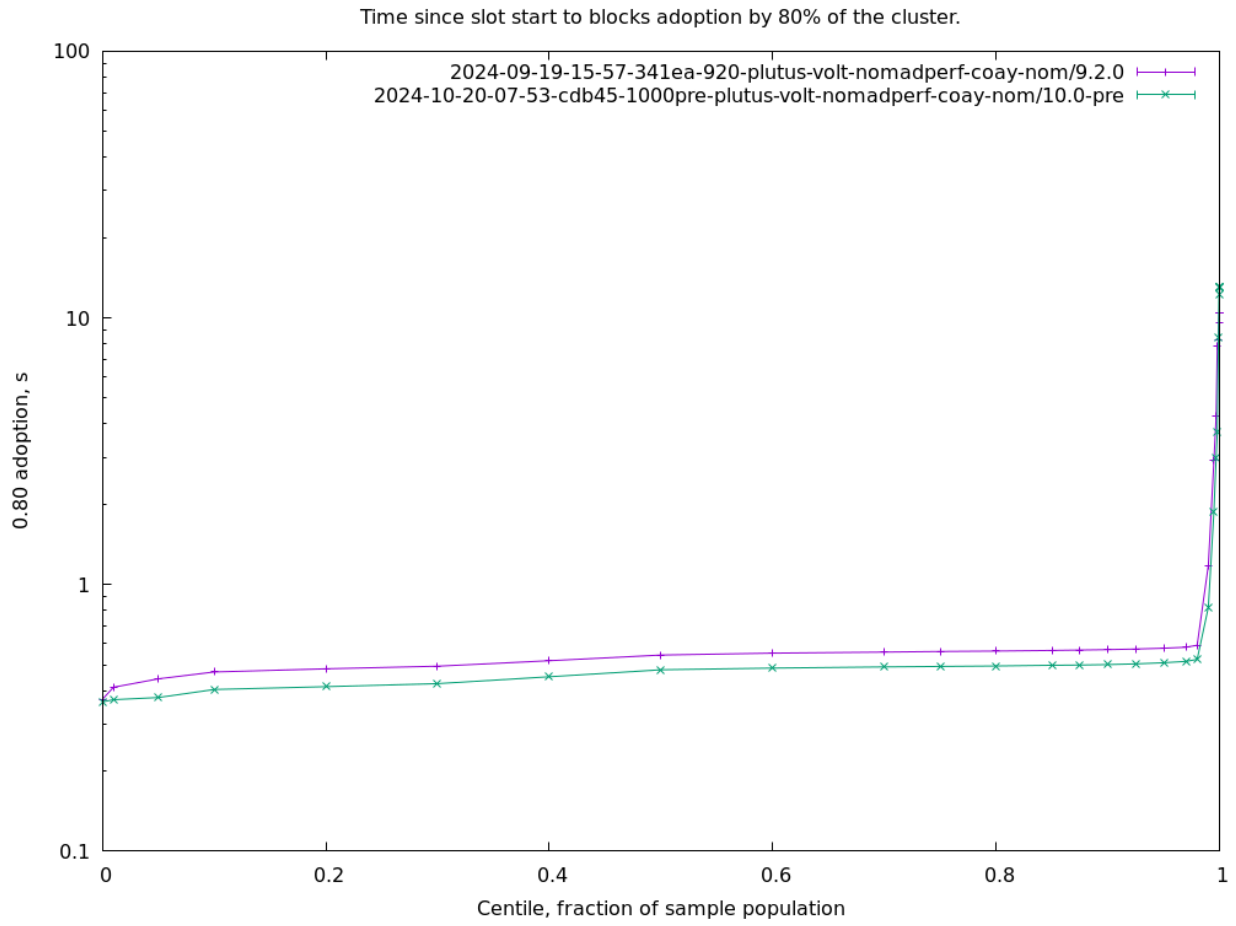
**Fetches to adopted (cdfPeerAdoption)** Time until the peer adopts the block (TraceAddBlockEvent.AddedToCurrentChain), since it was fetched



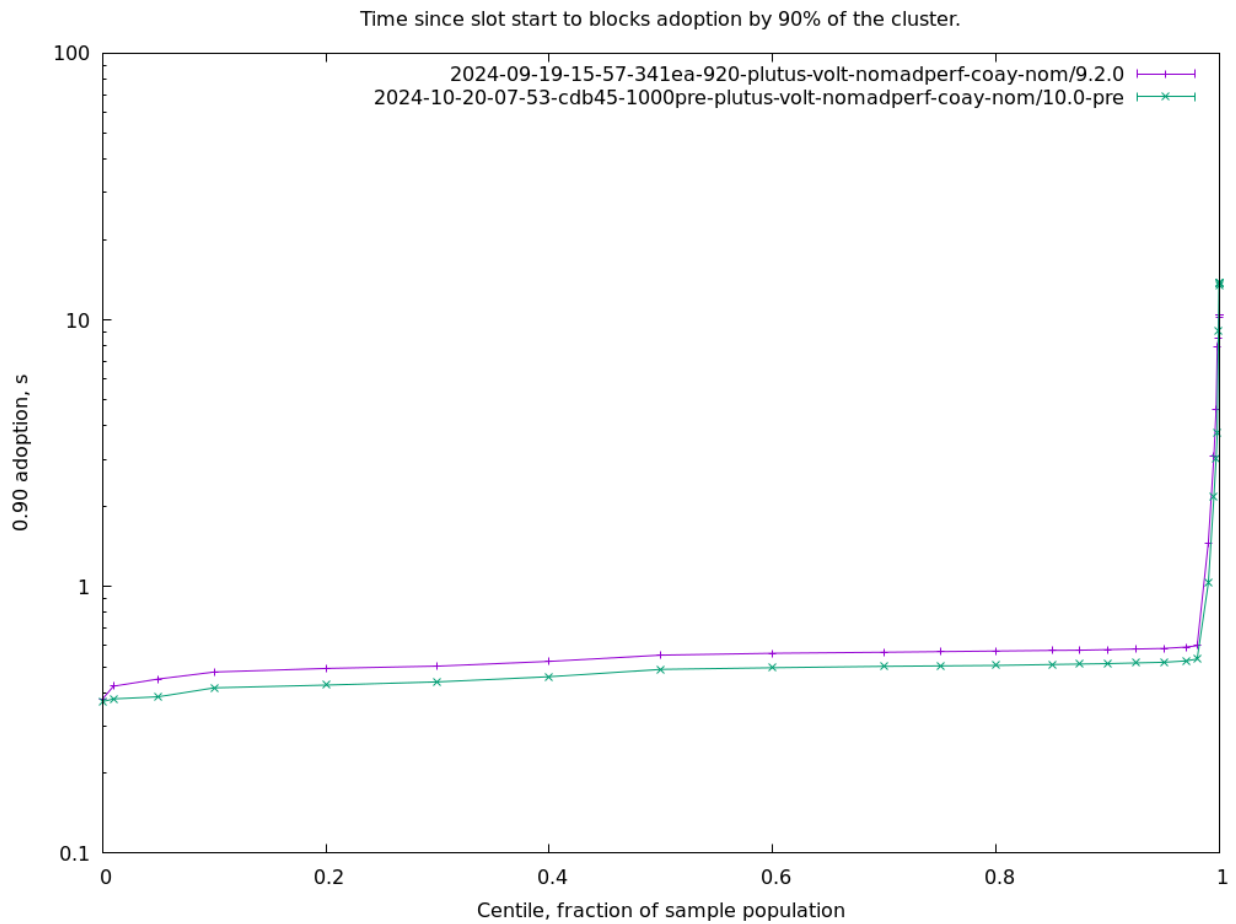
**0.50 adoption (cdf0.50)** Time since slot start to block's adoption by 50% of the cluster.



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



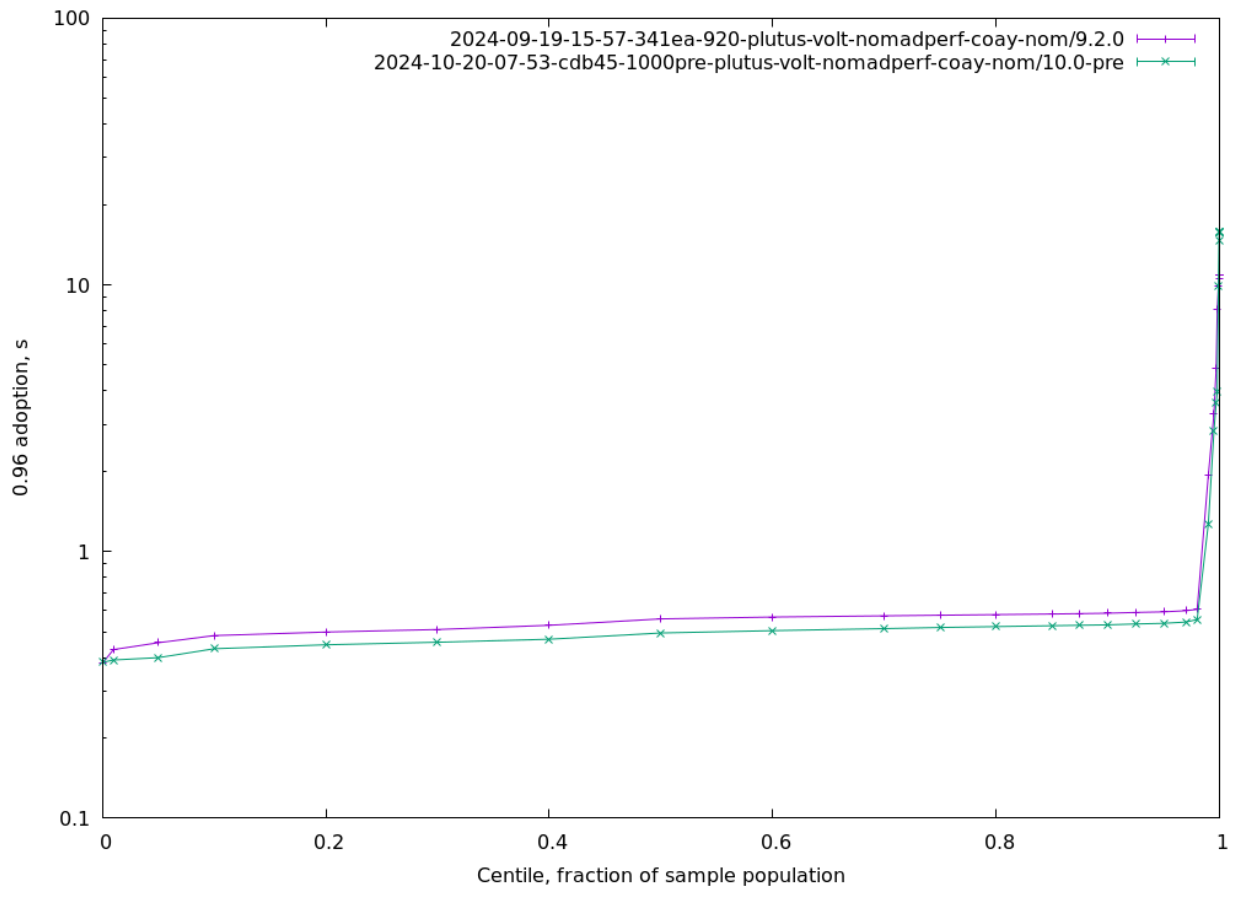
**0.90 adoption (cdf0.90)** Time since slot start to block's adoption by 90% of the cluster.



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



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