

8.9.0 and 8.8.0 against 8.7.2  
value-only workload

Michael Karg, Cardano Performance team

2024-03-10

# 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 8.8.0 (Babbage) and 8.9.0 (Babbage) relative to 8.7.2 (Babbage), under value-only workload.

	8.7.2	8.8.0	8.9.0
Analysis date	2023-12-13	2024-02-20	2024-03-07
Cluster system start date	2023-12-12	2024-02-19	2024-03-06
Cluster system start time	16:39:34	09:40:53	20:28:53
Identifier	8.7.2	8.8.0	8.9.0
Run batch	rel872	rel880tr	rel890
GHC version	8.10.7	8.10.7	8.10.7
cardano-node version	8.7.2	8.8.0	8.9.0
ouroboros-consensus version	0.14.0.0	0.15.0.0	0.16.0.0
ouroboros-network version	0.10.1.0	0.11.0.0	0.12.0.0
cardano-ledger-core version	1.9.0.0	1.10.0.0	1.10.0.0
plutus-core version	1.15.0.1	1.21.0.0	1.21.0.0
cardano-crypto version	1.1.2	1.1.2	1.1.2
cardano-prelude version	0.1.0.4	0.1.0.4	0.1.0.4
cardano-node git	f4b1a35	113b8c5	4a3f247
ouroboros-consensus git	15ae941	21558d8	a2cb6e5
ouroboros-network git	ff2331f	5618742	c86df02
cardano-ledger-core git	f85ec6f	6e2d37c	6e2d37c
plutus-core git	e2cbee0	022595e	022595e
cardano-crypto git	6568a5e	6568a5e	6568a5e
cardano-prelude git	a6f18f7	a6f18f7	a6f18f7
Era	babbage	babbage	babbage
Delegation map size	1000000	1000000	1000000
Starting UTxO set size	4000000	4000000	4000000
Extra tx payload	100	100	100
Tx inputs	2	2	2
Tx Outputs	2	2	2
TPS	12.0	12.0	12.0
Transaction count	768000	768000	768000
Plutus script	—	—	—
Machines	52	52	52
Number of filters applied	3	3	3
Log text lines emitted per host	5250111.6153	5589196.3846	5739337.6346
Log objects emitted per host	5250081.6153	5589166.3846	5739307.6346
Log objects analysed per host	2348919.4230	2458090.2884	2514109.6153
Host run time, s	63949.5	63932.9	63925.3
Host log line rate, Hz	82.098	87.423	89.782
Total log objects analysed	122143810	127820695	130733700
Run time, s	63955	63938	63932
Analysed run duration, s	48046	48023	48028
Run time efficiency	0.75	0.75	0.75
Node start spread, s	8.9680611	11.941143	9.5090943
Node stop spread, s	2.6481188	2.7378592	4.0084557
Perf analysis start spread, s	0	0	0
Perf analysis stop spread, s	3	2	4
Slots analysed	48044	48022	48024
Blocks analysed	2234	2264	2265
Blocks rejected	863	909	934

# Chapter 2

## Analysis

### 2.1 Resource Usage

	8.7.2	8.8.0	$\Delta$	$\Delta\%$	8.9.0	$\Delta$	$\Delta\%$
Forge loop starts, #	0.99872	0.99874	0.000	0	0.99871	-0.000	0
Process CPU usage, %	8.3256	8.5332	0.208	2	8.973	0.647	8
RTS GC CPU usage, %	1.1426	1.1712	0.029	3	1.2289	0.086	8
RTS Mutator CPU usage, %	7.1716	7.3485	0.177	2	7.7134	0.542	8
Major GCs, #	0.00101	0.00101	0.000	0	0.00101	0.000	0
Minor GCs, #	2.1545	2.2022	0.048	2	2.228	0.074	3
Kernel RSS, MB	8370.0	8300.6	-69.400	-1	8304.4	-65.600	-1
RTS heap size, MB	8321.0	8249.5	-71.500	-1	8253.3	-67.700	-1
RTS live GC dataset, MB	3794.2	3808.9	14.700	0	3786.3	-7.900	0
RTS alloc rate, MB/s	66.957	68.784	1.827	3	69.598	2.641	4
Filesystem reads, KB/s	0.00077	0.0004	-0.000	0	0.00014	-0.001	-130
Filesystem writes, KB/s	229.77	231.6	1.830	1	231.85	2.080	1
CPU 85% spans, slots	0.07327	0.07338	0.000	0	0.07995	0.007	10
Sample count	(249>)	(249>)			(249>)		

### 2.2 Anomaly control

	8.7.2	8.8.0	$\Delta$	$\Delta\%$	8.9.0	$\Delta$	$\Delta\%$
Blocks per host, blocks	61.384	63.019	1.635	3	63.403	2.019	3
Filtered to chained block ratio, /	0.72185	0.71265	-0.009	-1	0.70733	-0.015	-2
Chained to forged block ratio, /	0.97007	0.96907	-0.001	0	0.97009	0.000	0
Height & slot battles, blocks	0.00671	0.00485	-0.002	-30	0.00485	-0.002	-30
Block size, B	89021	89017	-4	0	89009	-12	0
Sample count	(52)	(52)			(52)		

## 2.3 Forging

	8.7.2	8.8.0	$\Delta$	$\Delta\%$	8.9.0	$\Delta$	$\Delta\%$
Started forge loop iteration, s	0.00075	0.00094	0.000	0	0.0019	0.001	133
Acquired block context, s	0.02306	0.02336	0.000	0	0.02447	0.001	4
Acquired ledger state, s	6e-05	6e-05	0.000	0	6e-05	0.000	0
Acquired ledger view, s	2e-05	2e-05	0.000	0	3e-05	0.000	0
Leadership check duration, s	0.00041	0.00043	0.000	0	0.00045	0.000	0
Ledger ticking, s	0.02186	0.02206	0.000	0	0.02136	-0.001	-5
Mempool snapshotting, s	0.06949	0.06647	-0.003	-4	0.06657	-0.003	-4
Leadership to forged, s	0.00132	0.00133	0.000	0	0.00134	0.000	0
Forged to announced, s	0.00068	0.00068	0.000	0	0.00071	0.000	0
Forged to sending, s	0.00506	0.00519	0.000	0	0.00525	0.000	0
Forged to self-adopted, s	0.07008	0.06825	-0.002	-3	0.06989	-0.000	0
Slot start to announced, s	0.11771	0.11538	-0.002	-2	0.11692	-0.001	-1
Sample count	(2234)	(2264)			(2265)		

## 2.4 Individual peer propagation

	8.7.2	8.8.0	$\Delta$	$\Delta\%$	8.9.0	$\Delta$	$\Delta\%$
First peer notice, s	0.11942	0.11718	-0.002	-2	0.11869	-0.001	-1
First peer fetch, s	0.12882	0.1261	-0.003	-2	0.12793	-0.001	-1
Notice to fetch request, s	0.0012	0.00125	0.000	0	0.00115	-0.000	0
Fetch duration, s	0.35248	0.34624	-0.006	-2	0.34279	-0.010	-3
Fetches to announced, s	-0.0	-0.0	0.000	nan	-0.0	0.000	nan
Fetches to sending, s	0.04387	0.04397	0.000	0	0.04252	-0.001	-2
Fetches to adopted, s	0.07421	0.07289	-0.001	-1	0.07333	-0.001	-1
Sample count	(2234)	(2264)			(2265)		

## 2.5 End-to-end propagation

	8.7.2	8.8.0	$\Delta$	$\Delta\%$	8.9.0	$\Delta$	$\Delta\%$
0.50 adoption, s	0.6571	0.64495	-0.012	-2	0.64098	-0.016	-2
0.80 adoption, s	1.0078	0.99288	-0.015	-1	0.98909	-0.019	-2
0.90 adoption, s	1.0245	1.0104	-0.014	-1	1.0035	-0.021	-2
0.92 adoption, s	1.0279	1.0141	-0.014	-1	1.0085	-0.019	-2
0.94 adoption, s	1.0319	1.0192	-0.013	-1	1.012	-0.020	-2
0.96 adoption, s	1.0392	1.0252	-0.014	-1	1.019	-0.020	-2
0.98 adoption, s	1.0479	1.0355	-0.012	-1	1.0282	-0.020	-2
1.00 adoption, s	1.0797	1.0669	-0.013	-1	1.0533	-0.026	-2
Sample count	(2234)	(2264)			(2265)		

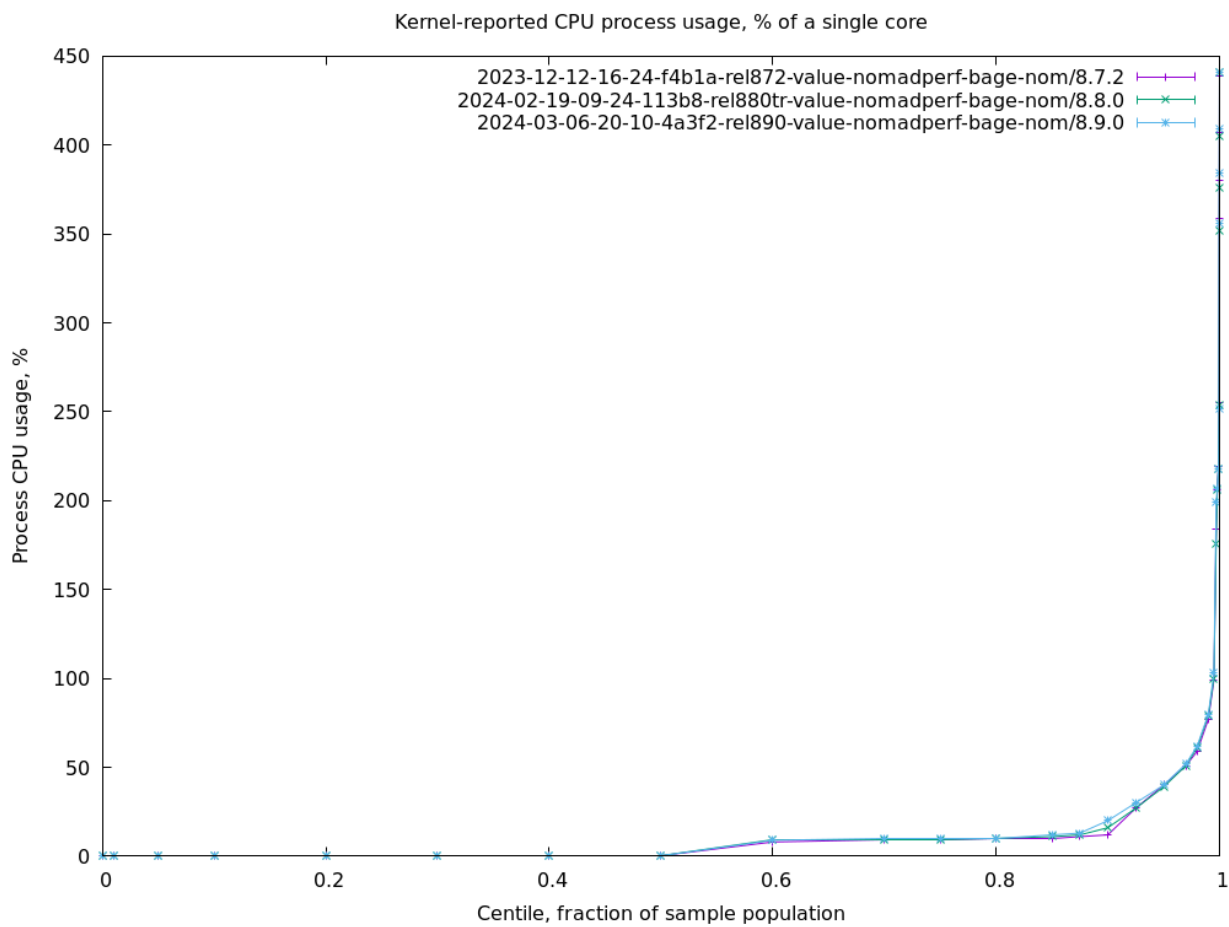
## Part I

# Appendix A: charts

# Chapter 3

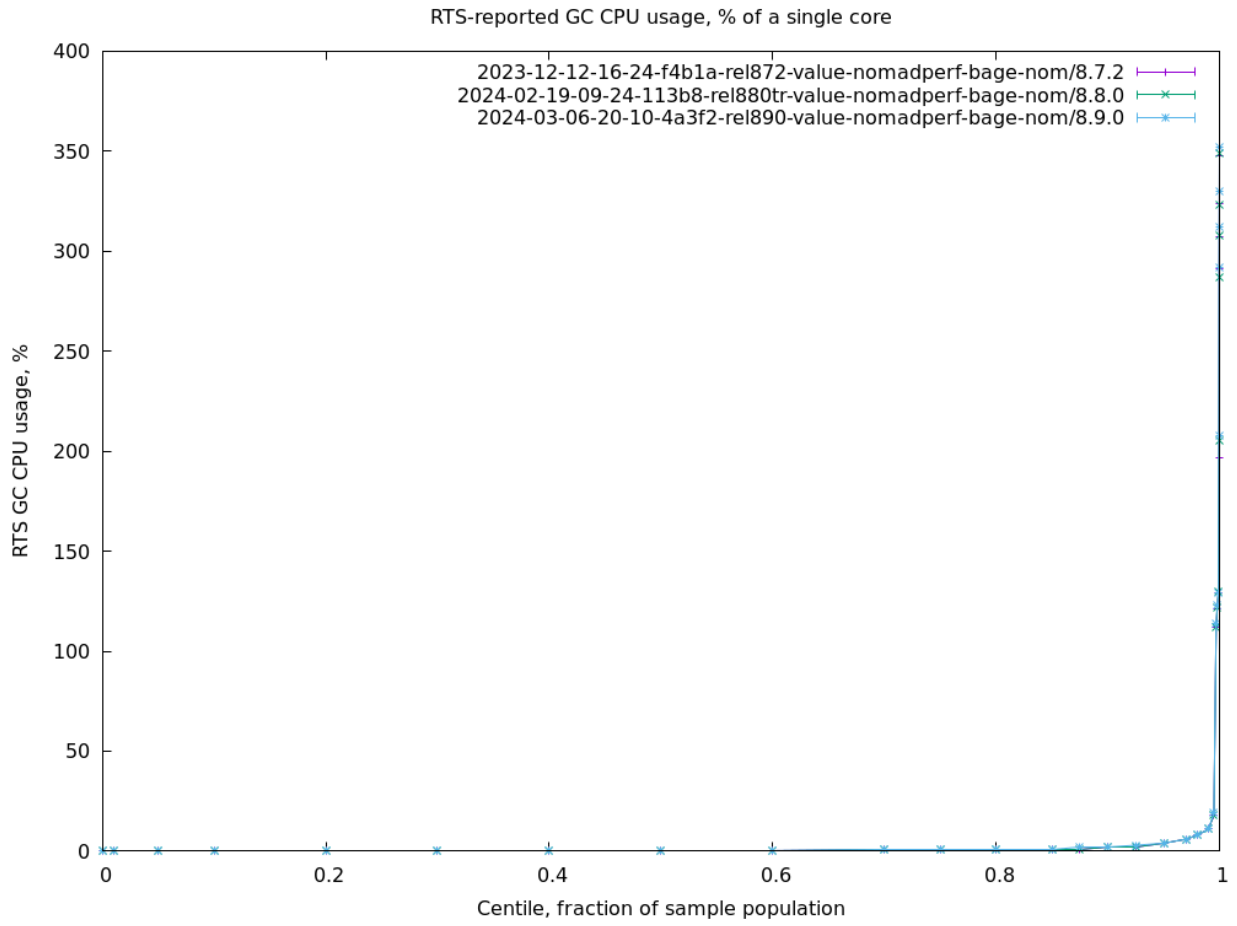
## Cluster performance charts

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

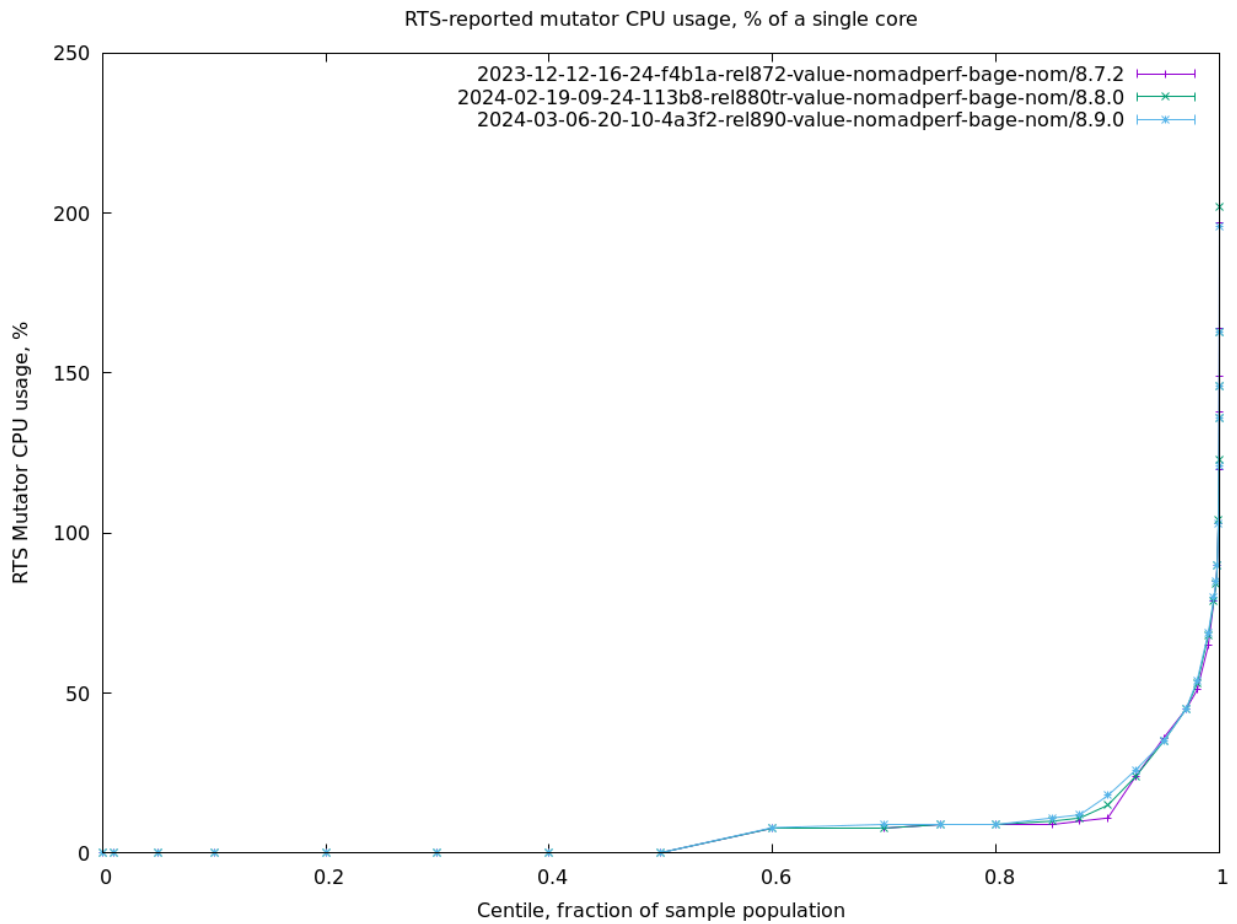


**RTS GC CPU usage (CentiGC)** RTS-reported GC CPU usage, % of a single core

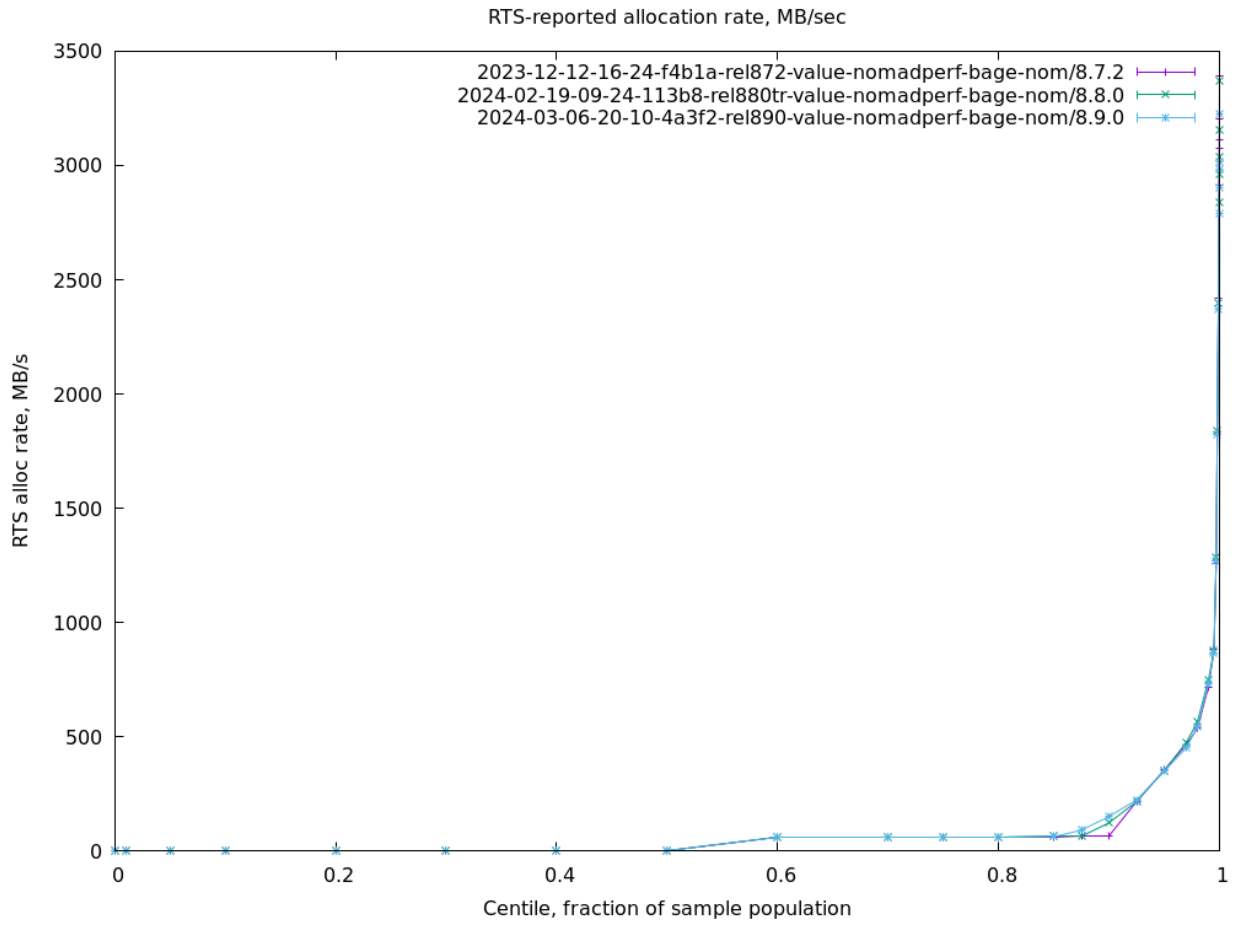




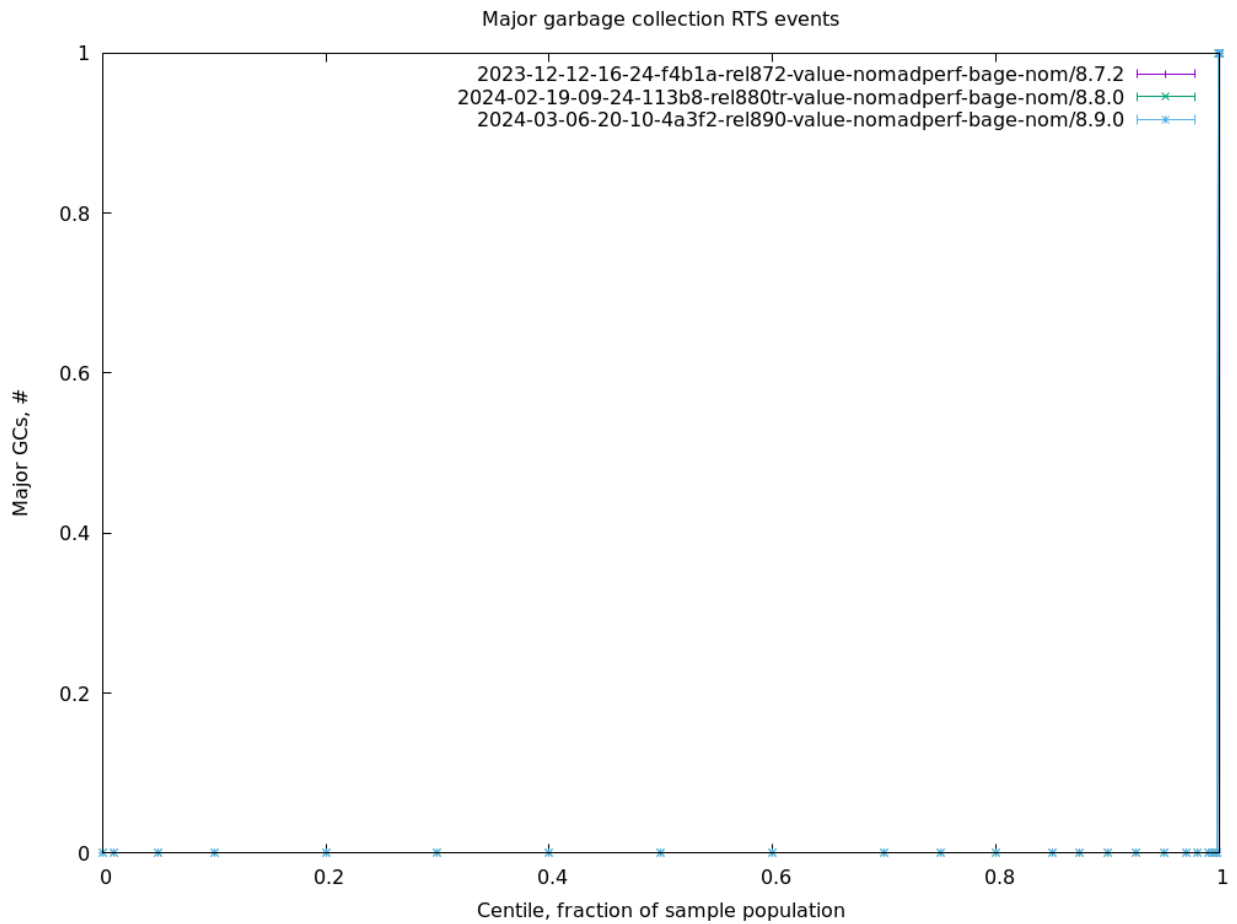
**RTS Mutator CPU usage (CentiMut)** RTS-reported mutator CPU usage, % of a single core



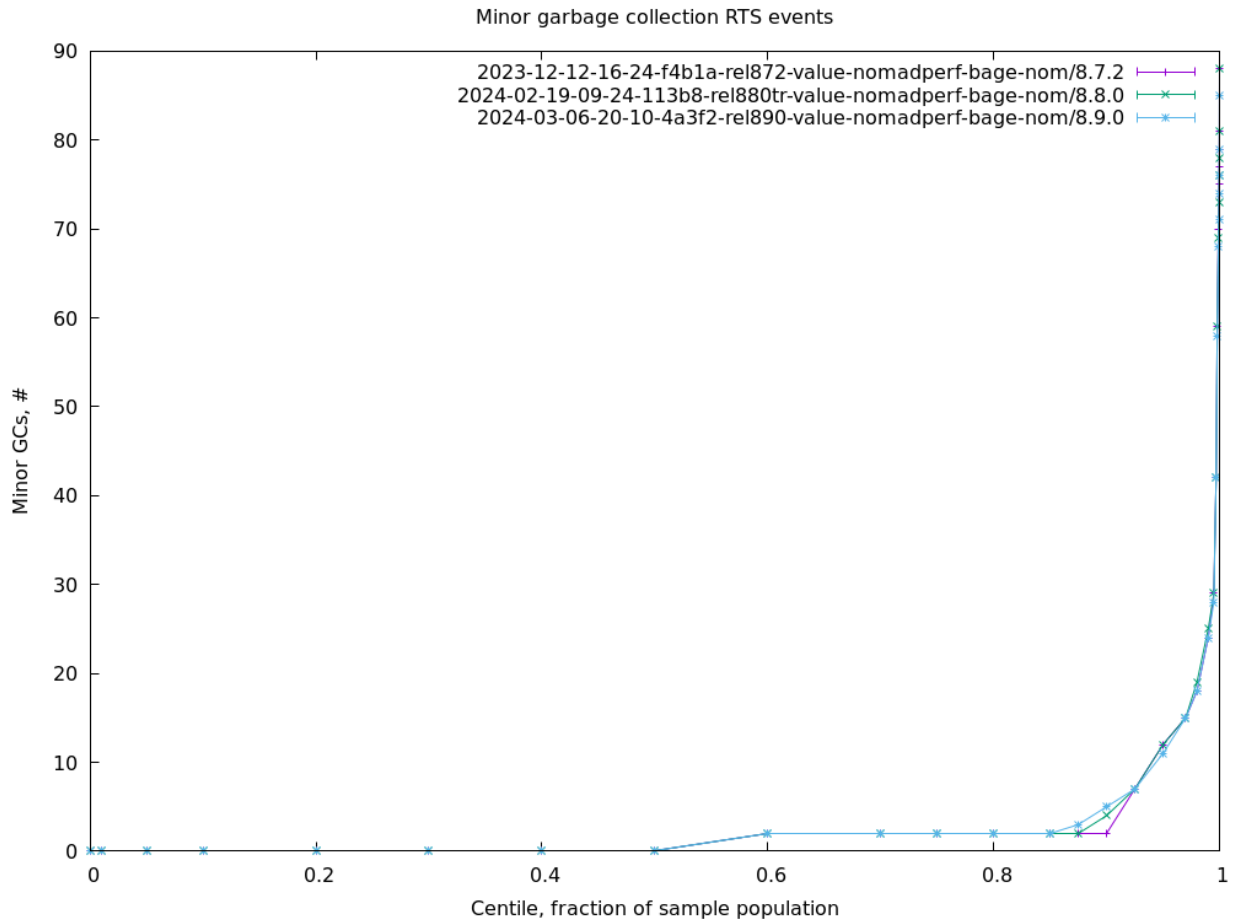
**RTS alloc rate (Alloc)** RTS-reported allocation rate, MB/sec



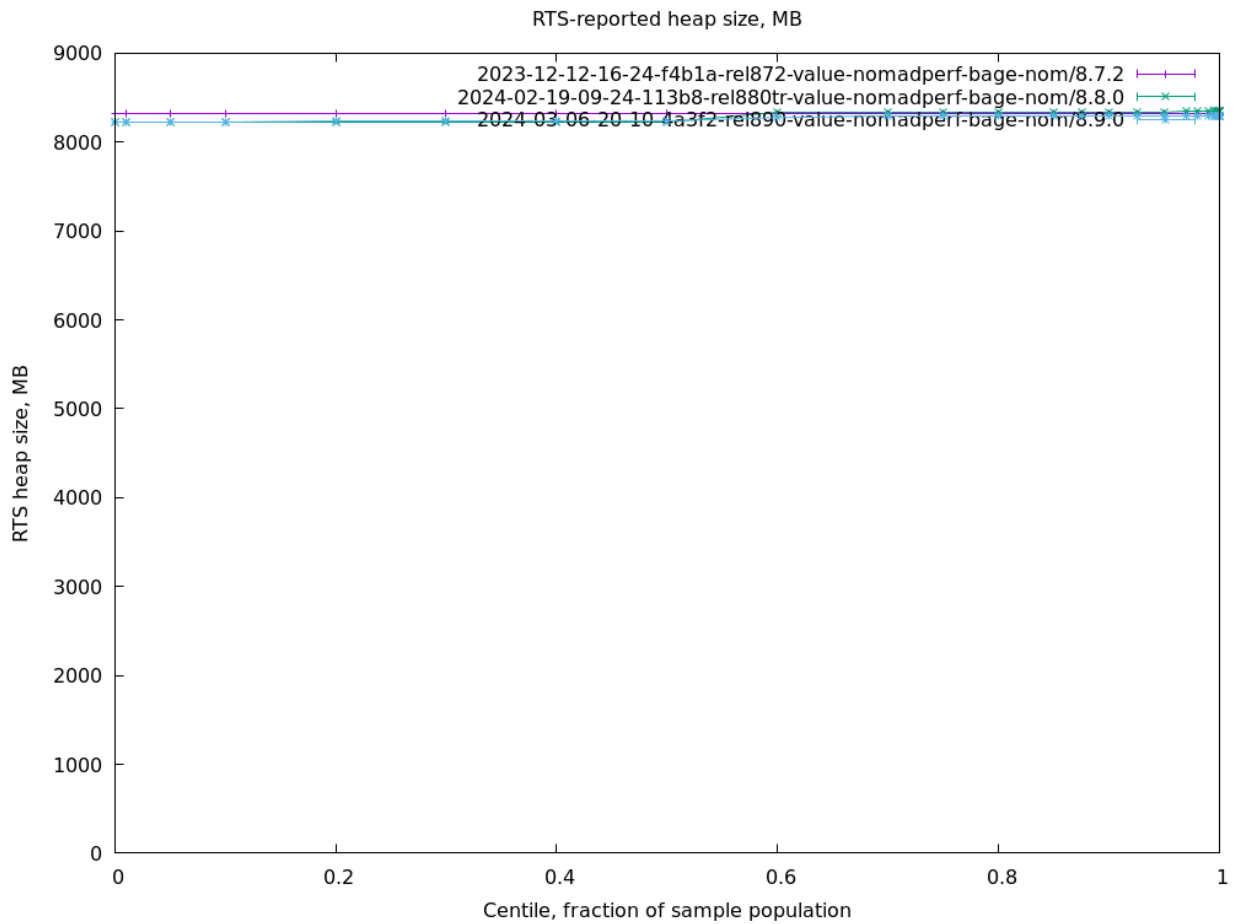
**Major GCs (GcsMajor)** Major garbage collection RTS events



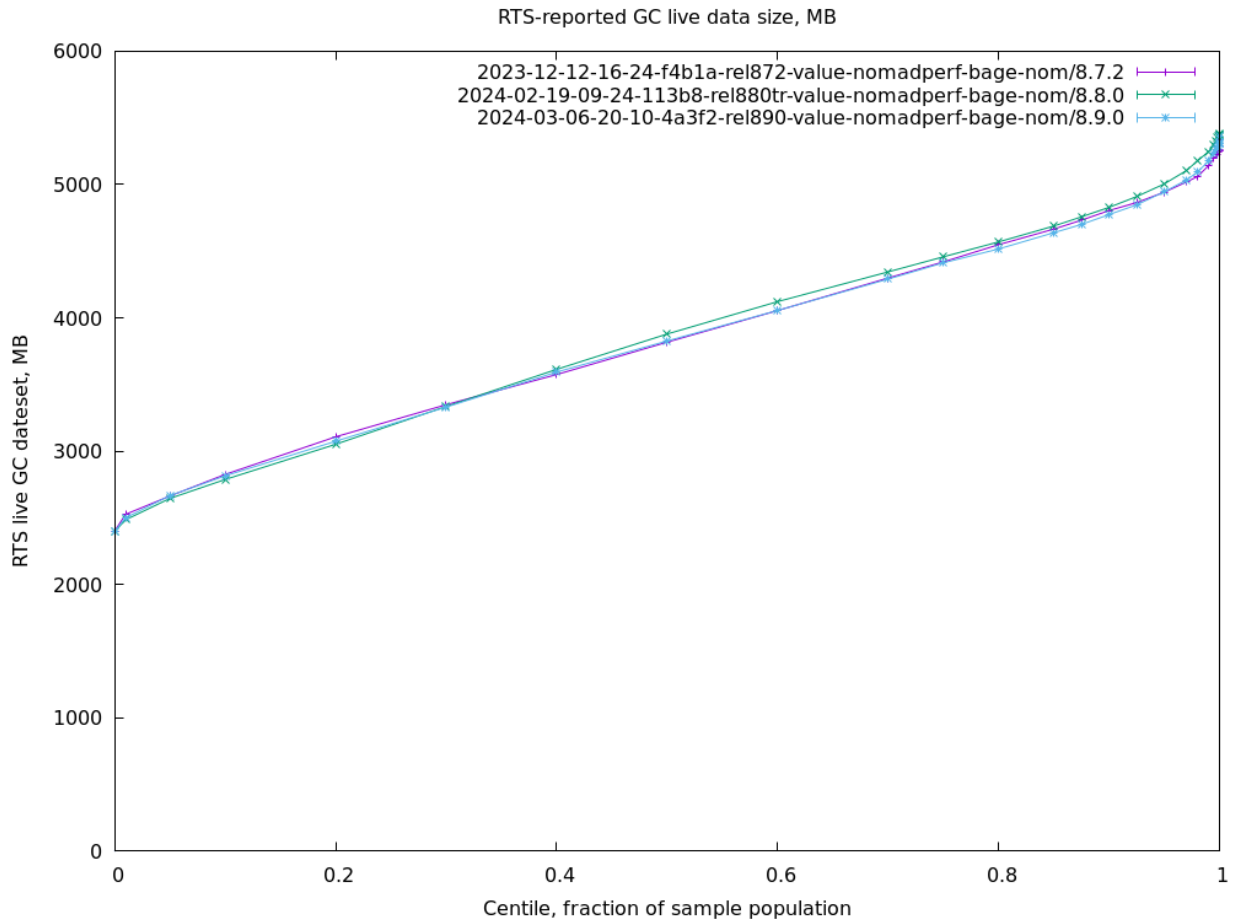
**Minor GCs (GcsMinor)** Minor garbage collection RTS events



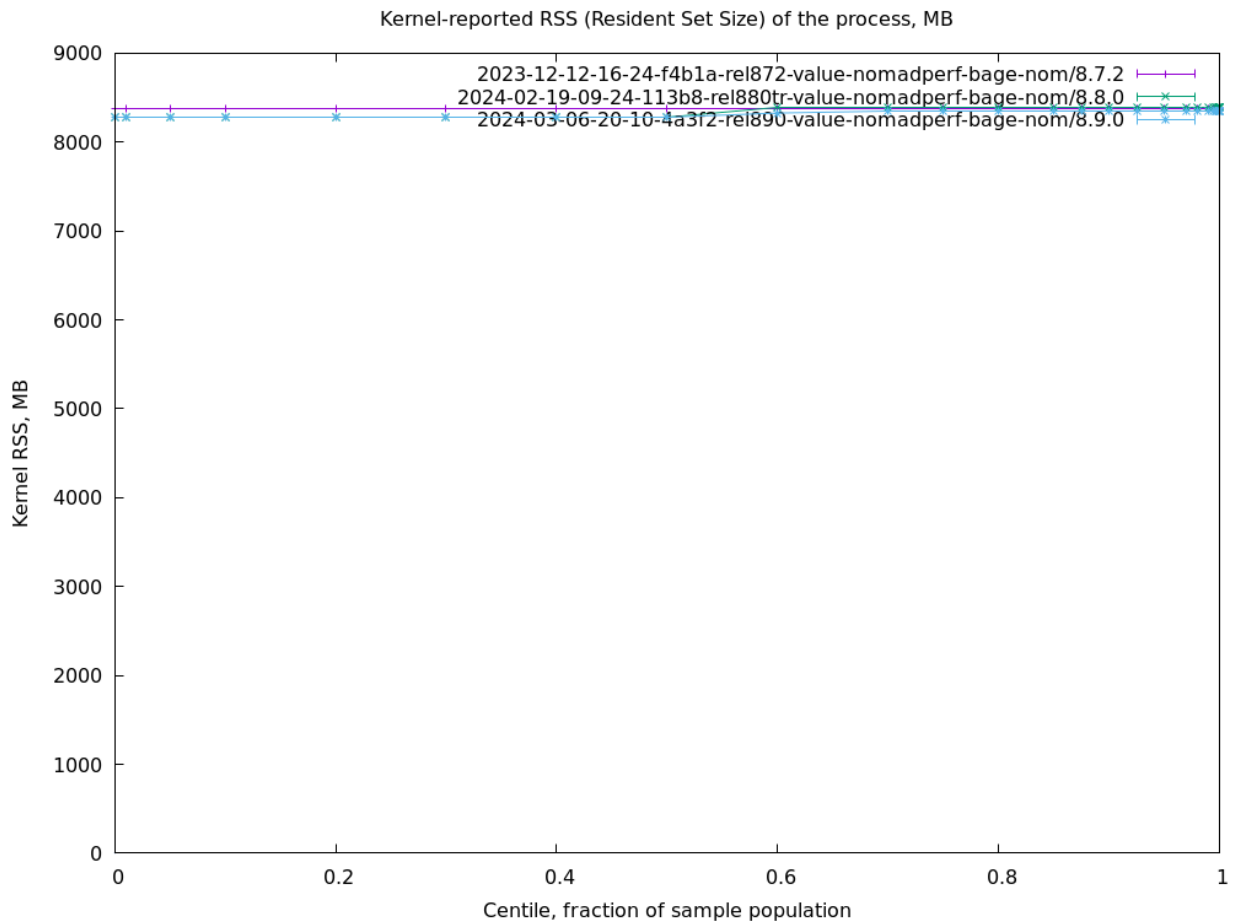
RTS heap size (Heap) RTS-reported heap size, MB



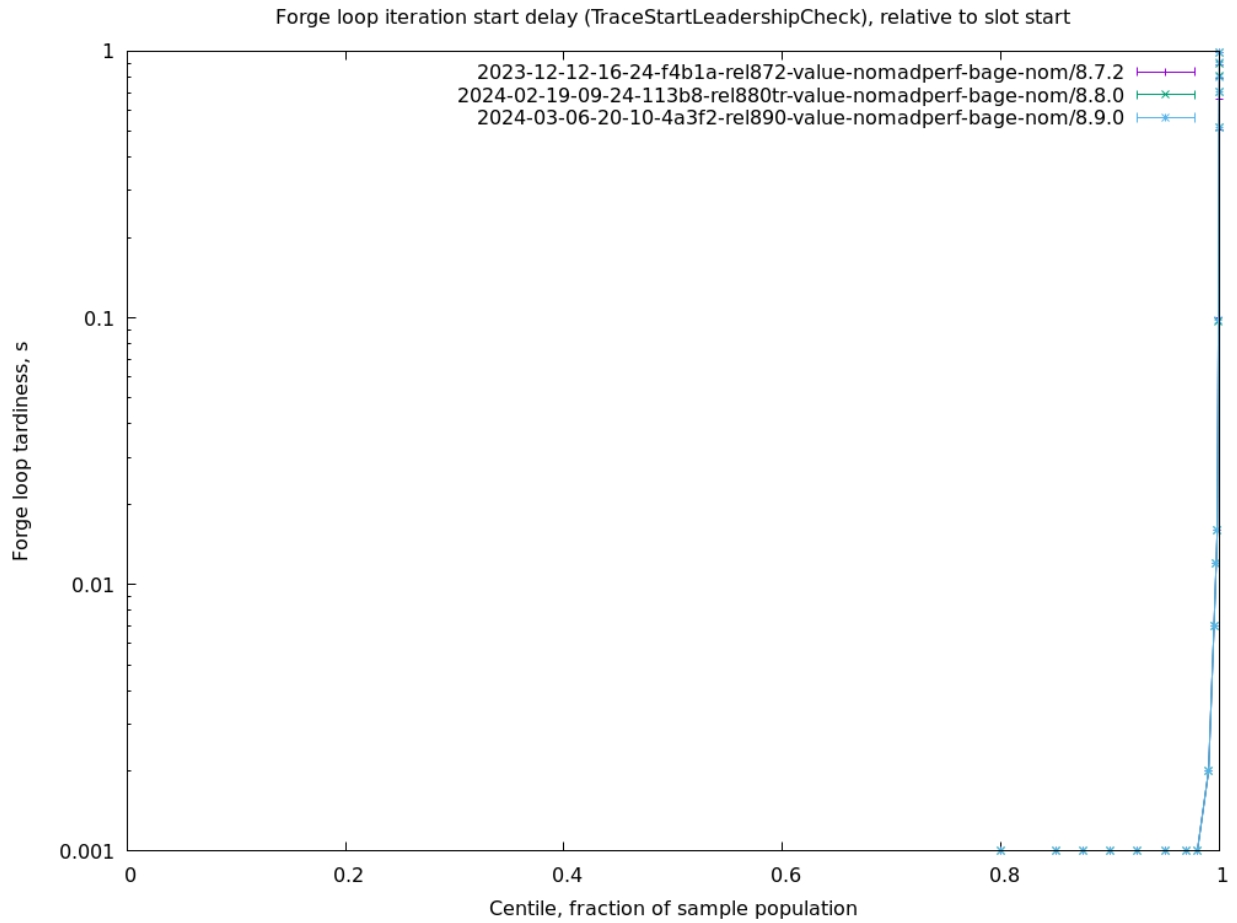
RTS live GC dataset (Live) RTS-reported GC live data size, MB



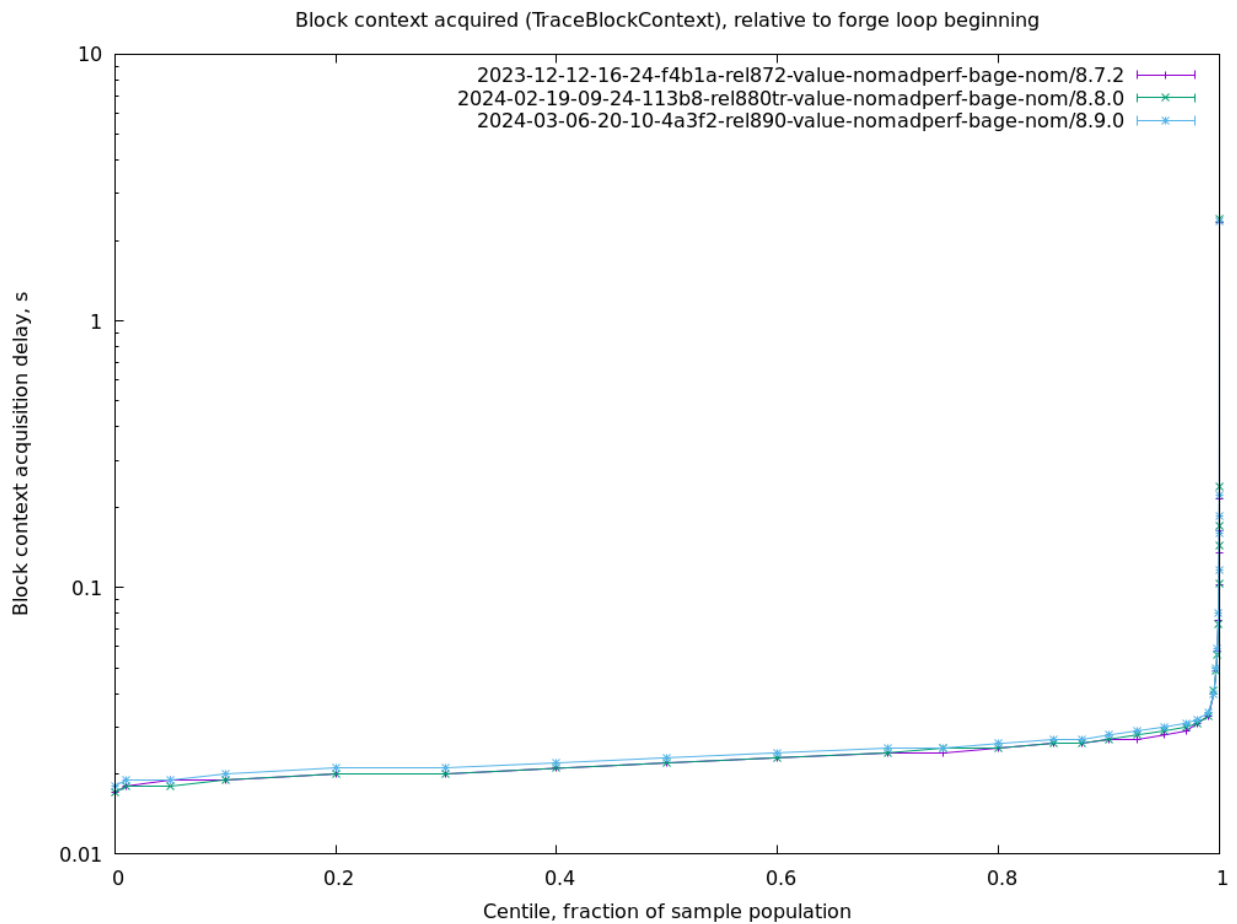
**Kernel RSS (RSS)** Kernel-reported RSS (Resident Set Size) of the process, MB



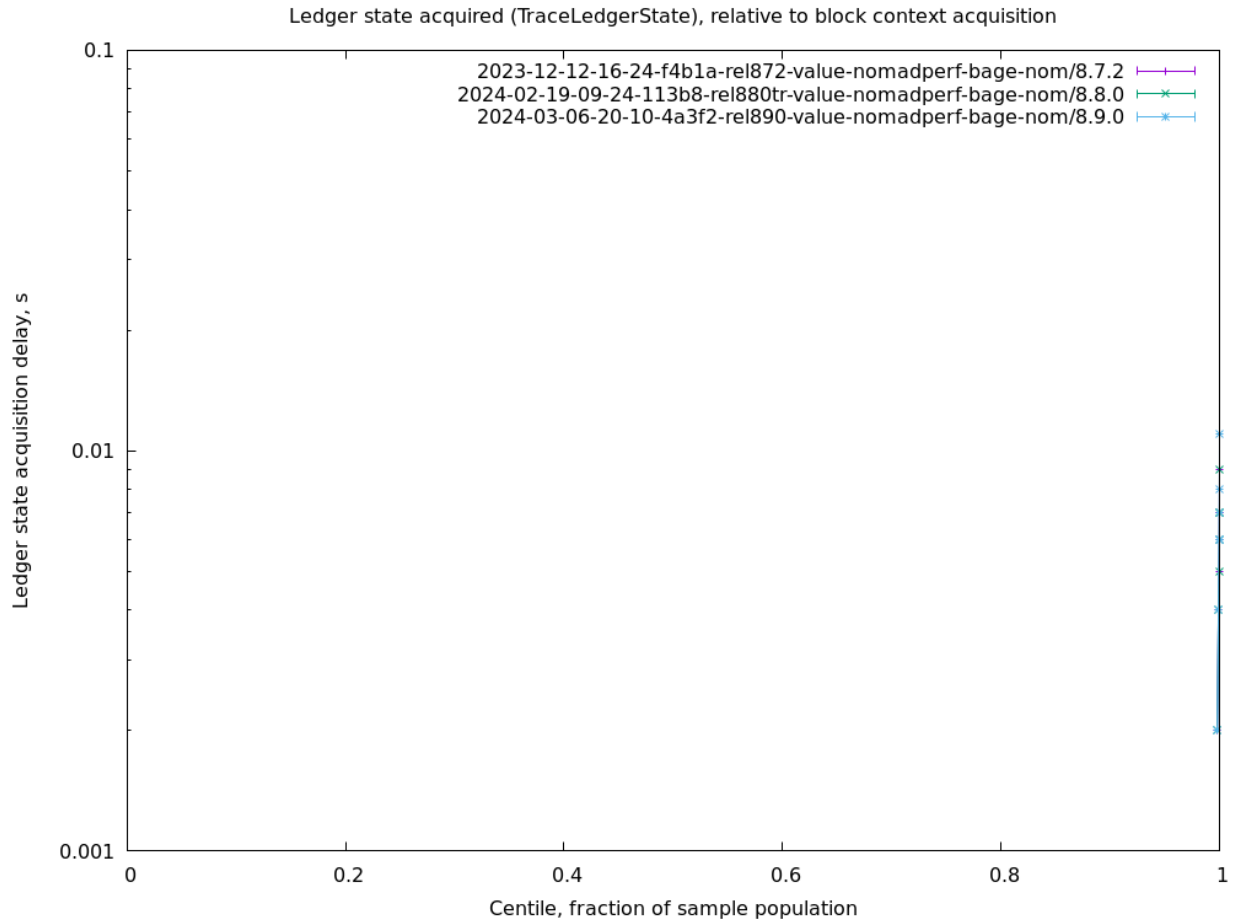
**Forge loop tardiness (cdfStarted)** Forge loop iteration start delay (TraceStartLeadershipCheck), relative to slot start



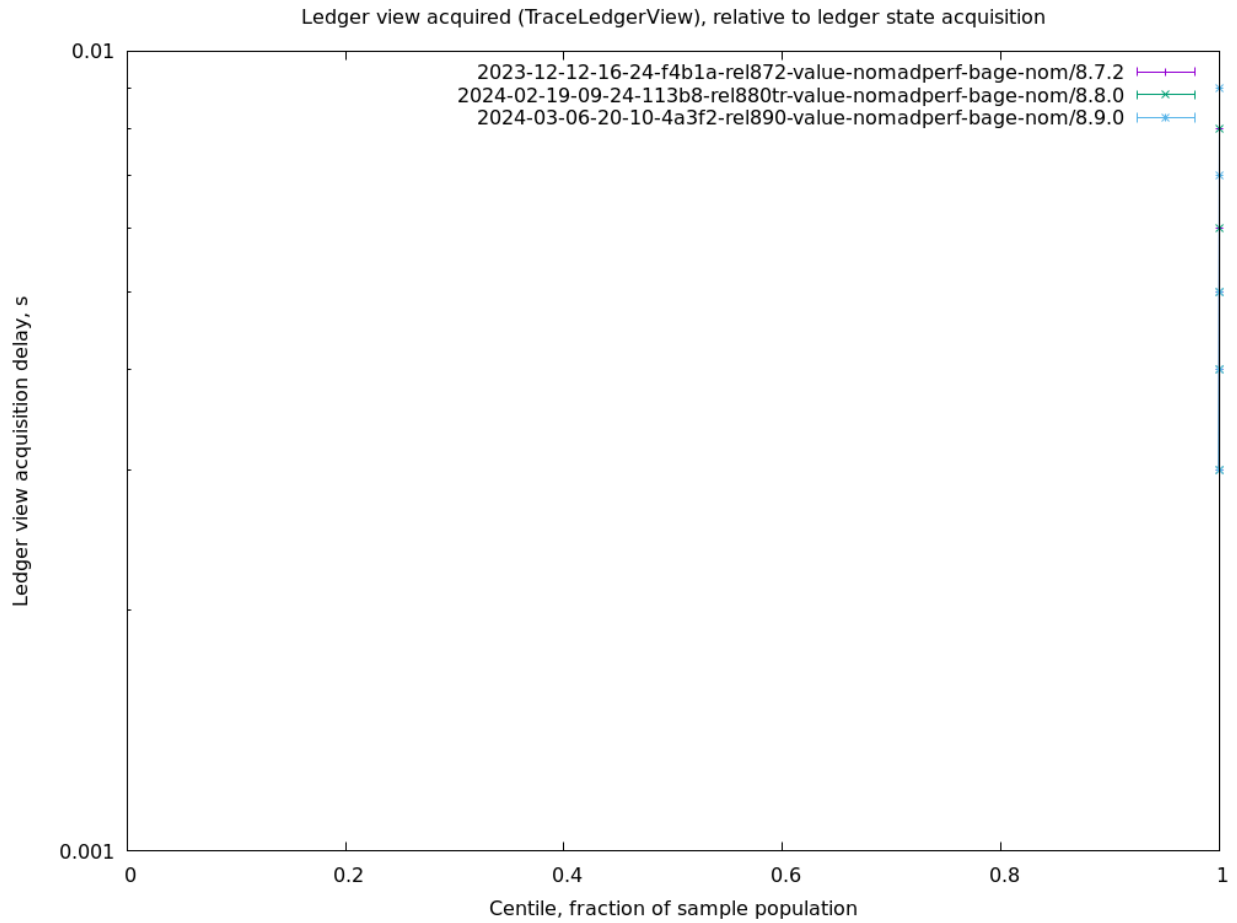
**Block context acquisition delay (cdfBlkCtx)** Block context acquired (TraceBlockContext), relative to forge loop beginning



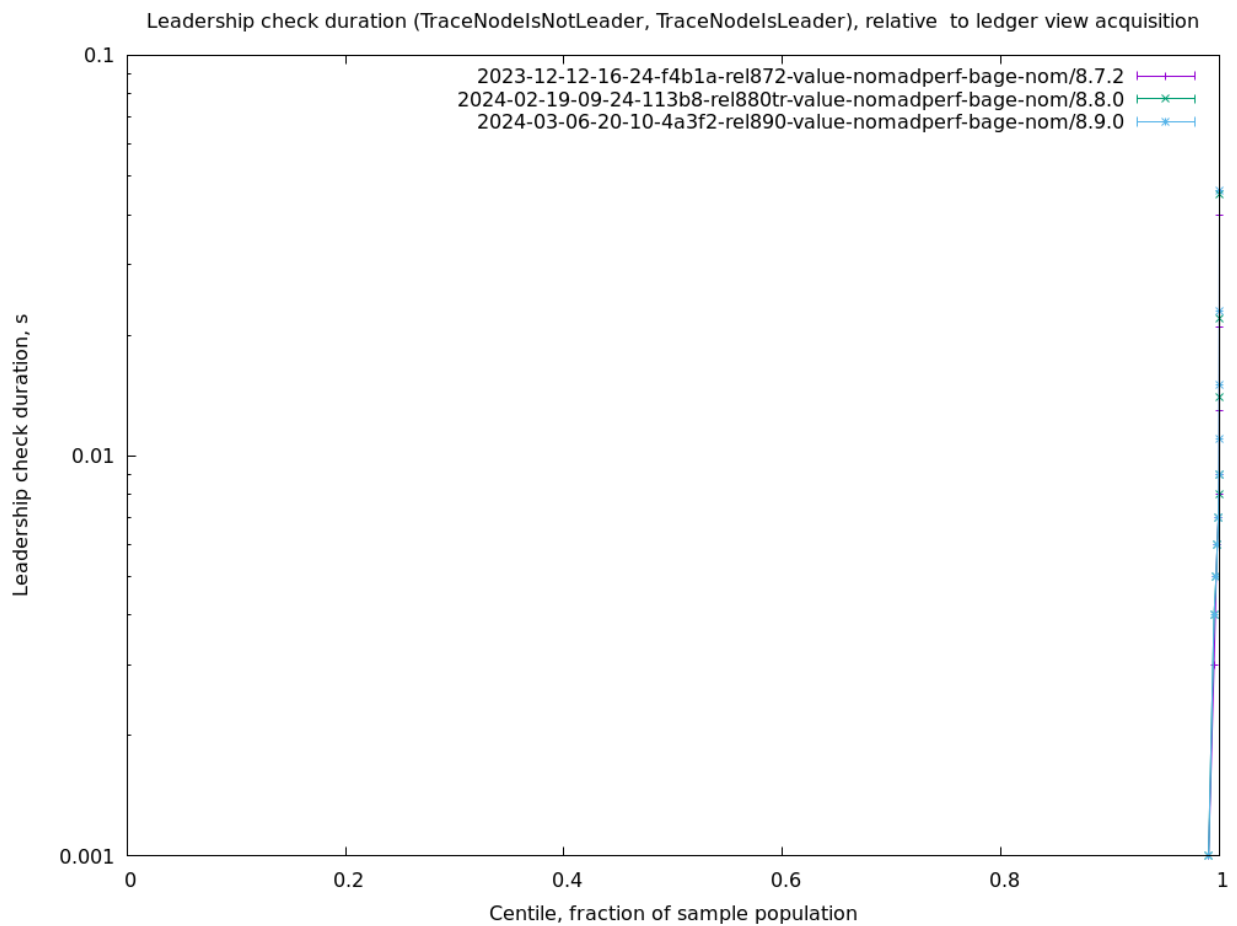
**Ledger state acquisition delay (cdfLgrState)** Ledger state acquired (TraceLedgerState), relative to block context acquisition



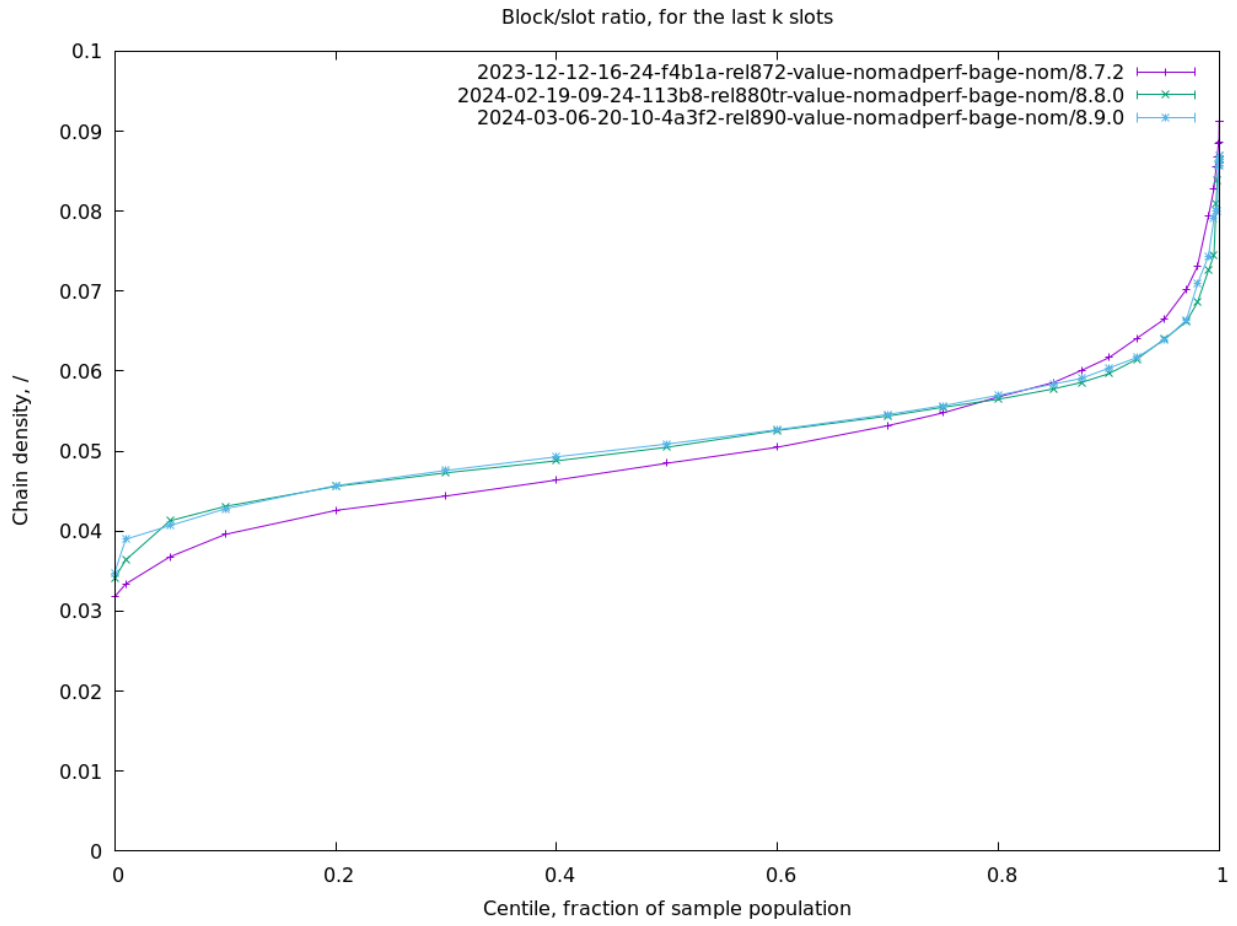
**Ledger view acquisition delay (cdfLgrView)** Ledger view acquired (TraceLedgerView), relative to ledger state acquisition



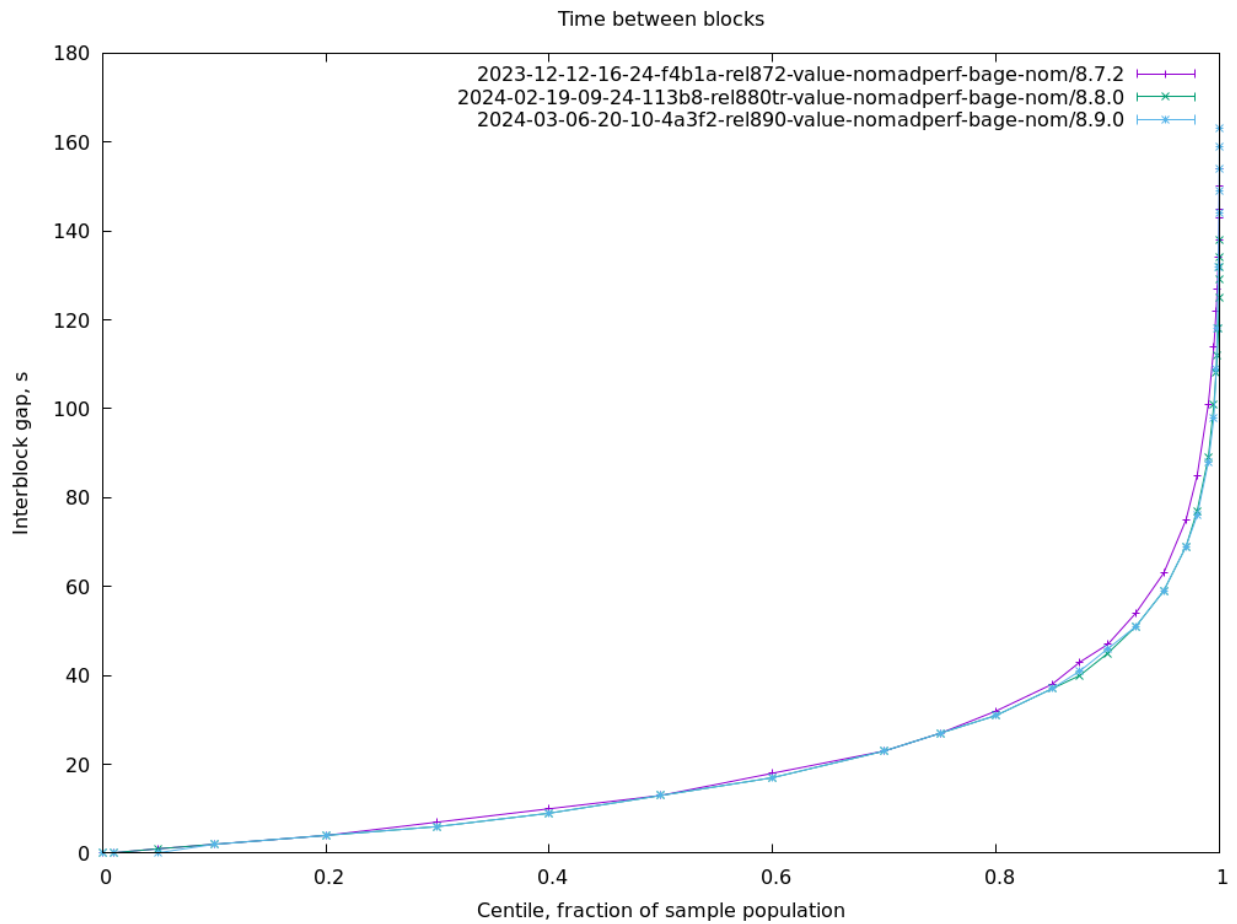
**Leadership check duration (cdfLeading)** Leadership check duration (TraceNodeIsNotLeader, TraceNodeIsLeader), relative to ledger view acquisition



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

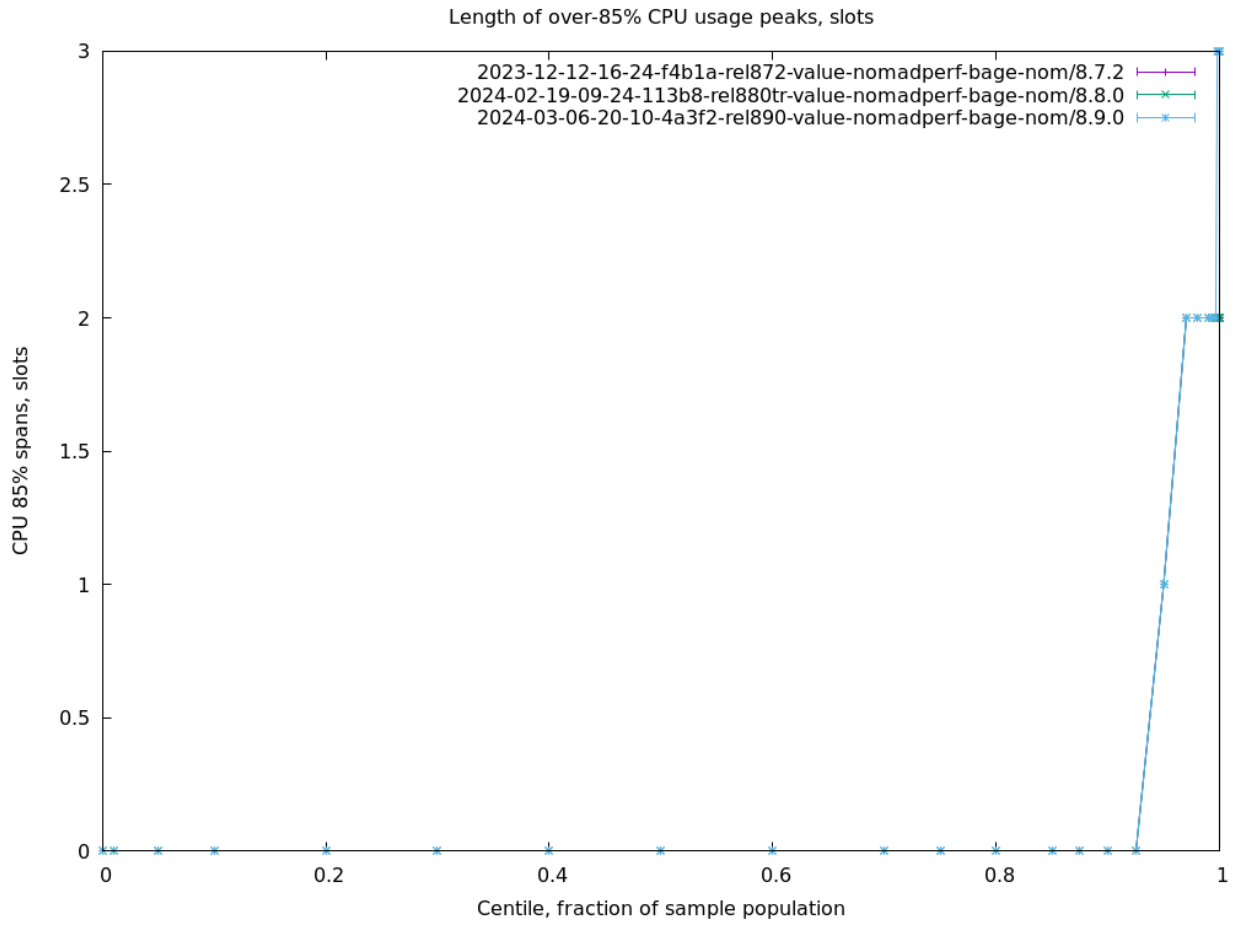


Interblock gap (cdfBlockGap) Time between blocks

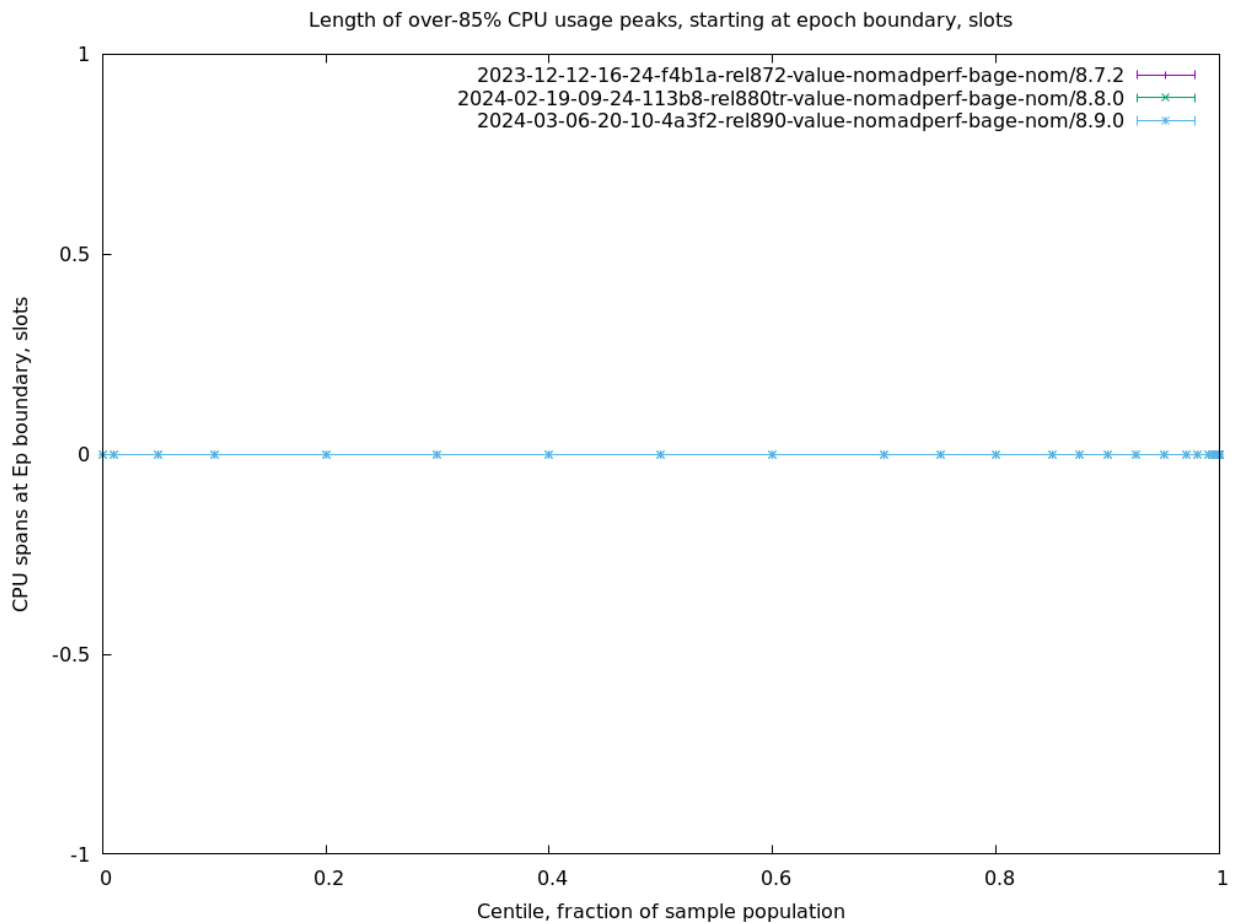


CPU 85% spans (cdfSpanLensCpu) Length of over-85% CPU usage peaks, slots

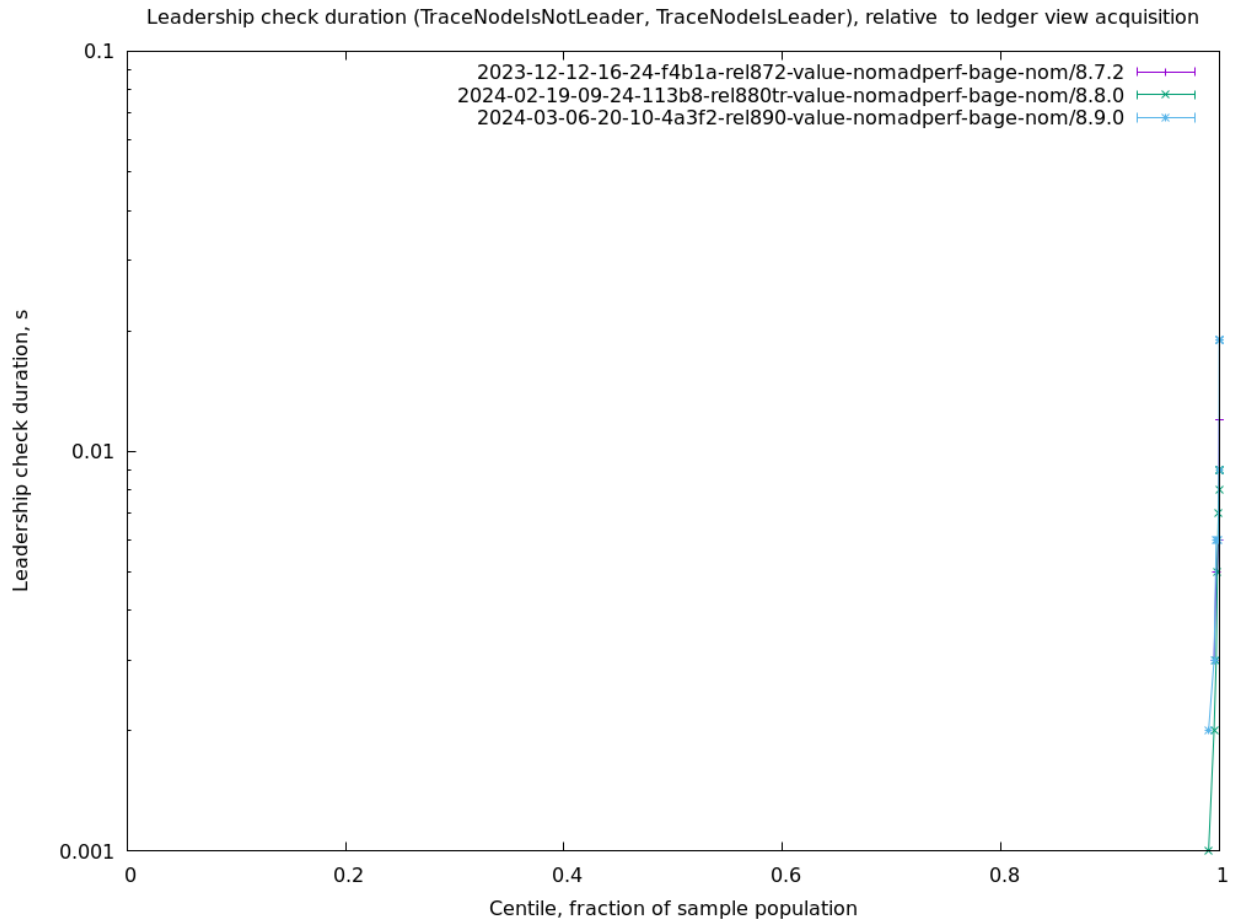




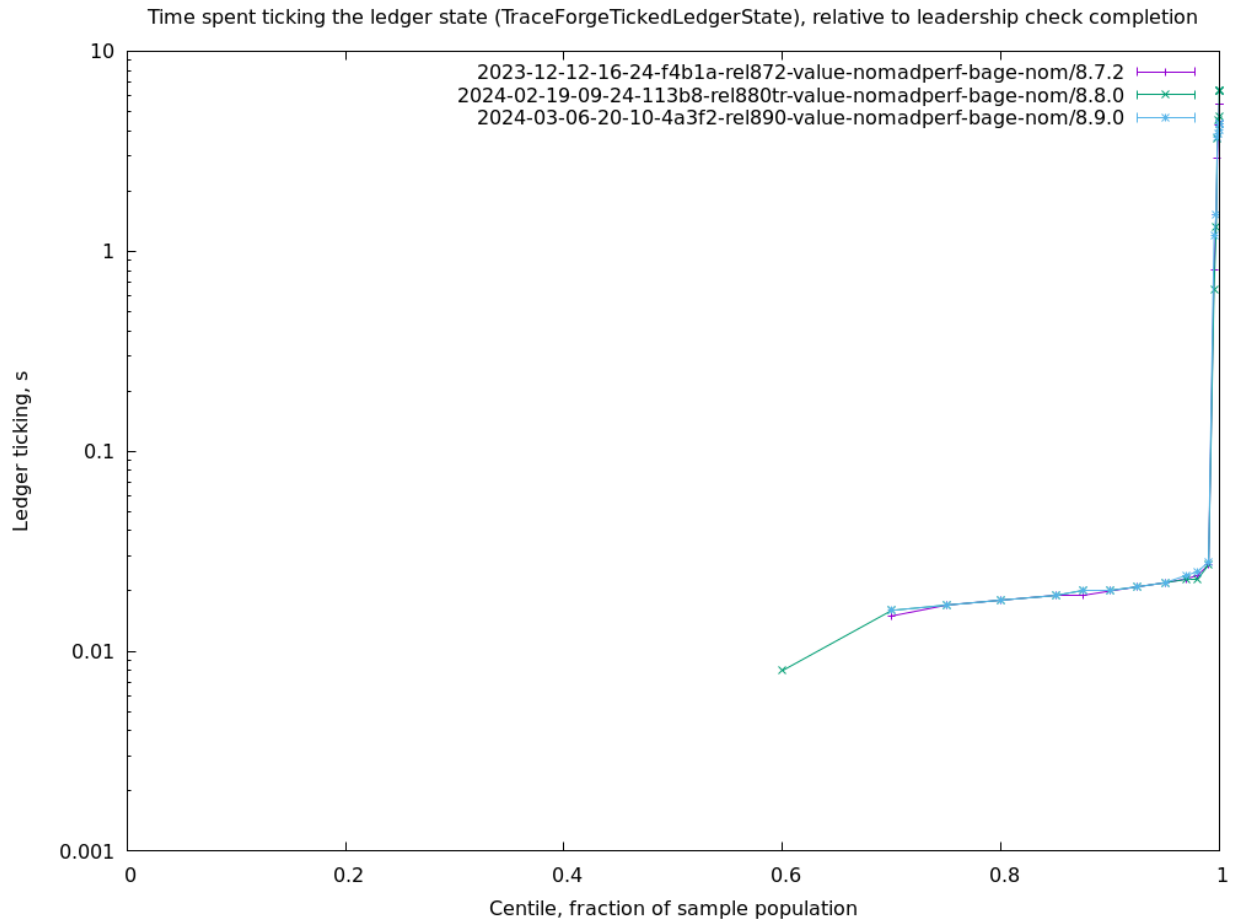
CPU spans at Ep boundary (cdfSpanLensCpuEpoch) Length of over-85% CPU usage peaks, starting at epoch boundary, slots



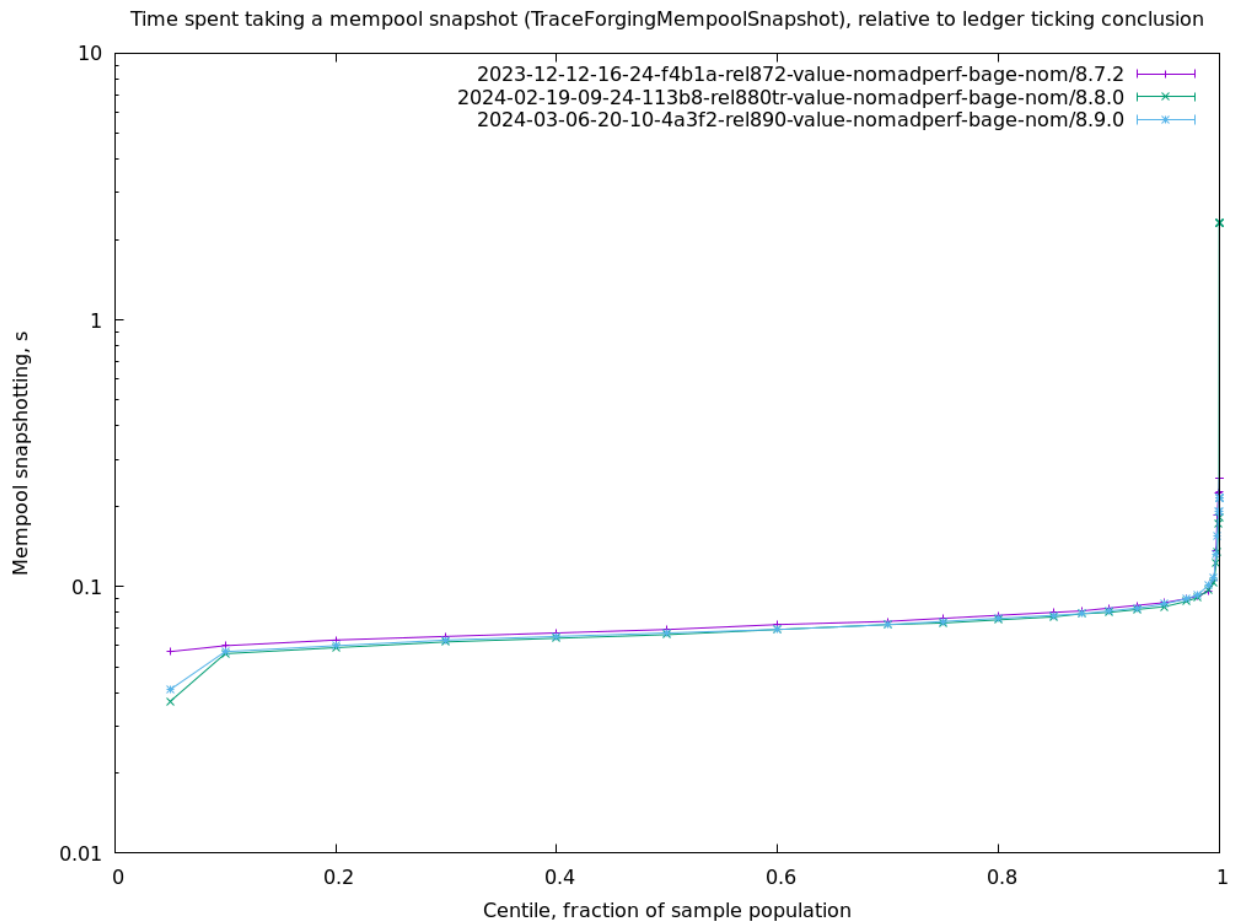
**Leadership check duration (cdfForgerLead)** Leadership check duration (TraceNodeIsNotLeader, TraceNodeIsLeader), relative to ledger view acquisition



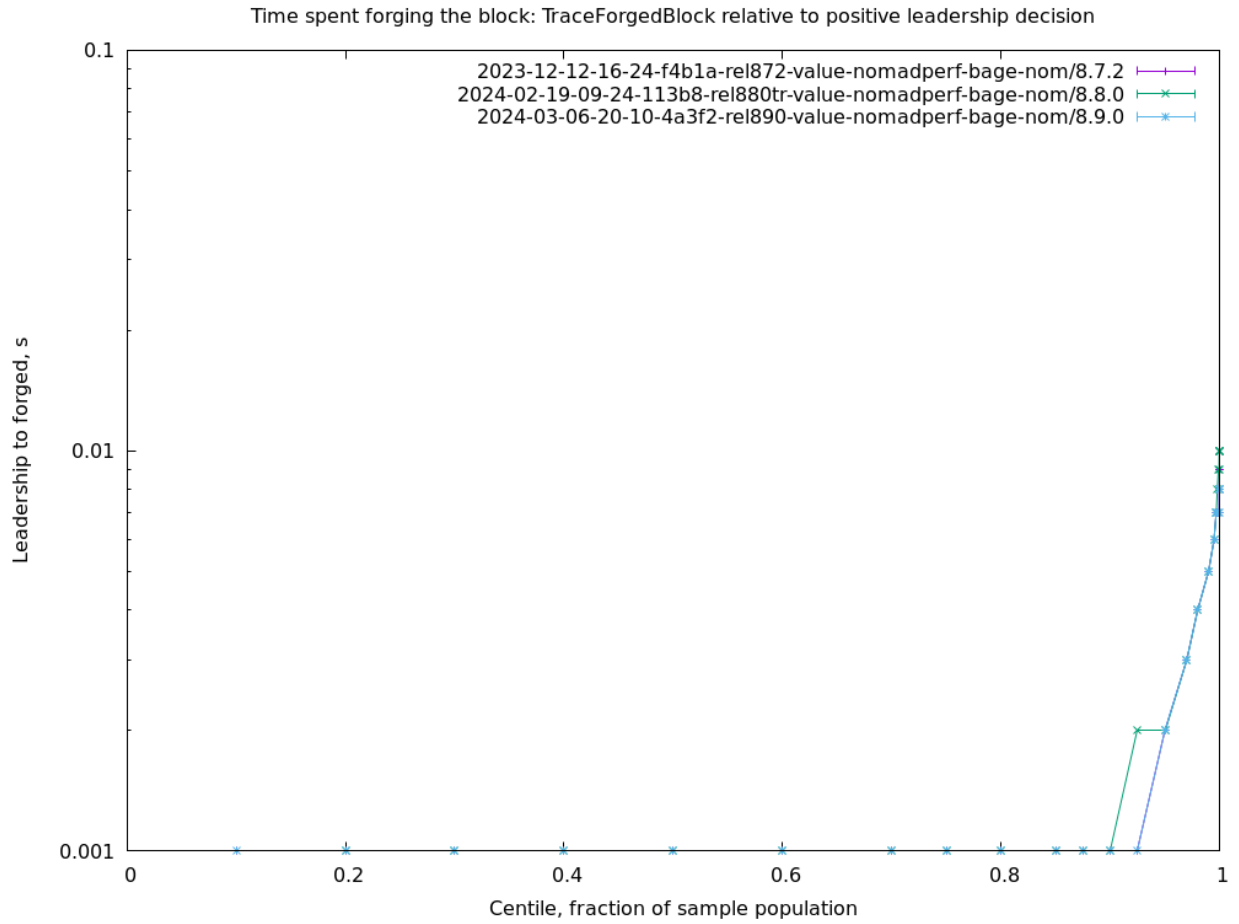
**Ledger ticking (cdfForgerTicked)** Time spent ticking the ledger state (TraceForgeTickedLedgerState), relative to leadership check completion



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

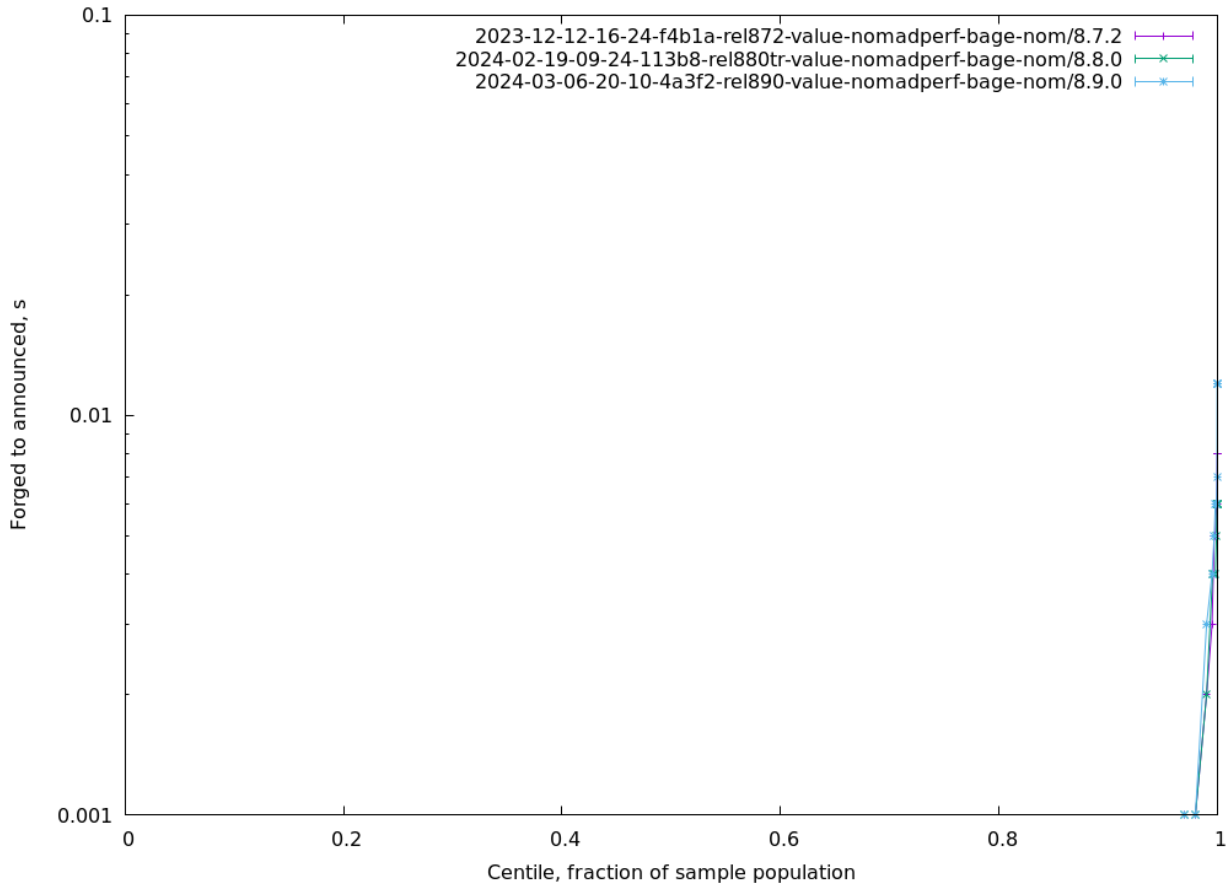


**Leadership to forged (cdfForgerForge)** Time spent forging the block: TraceForgedBlock relative to positive leadership decision



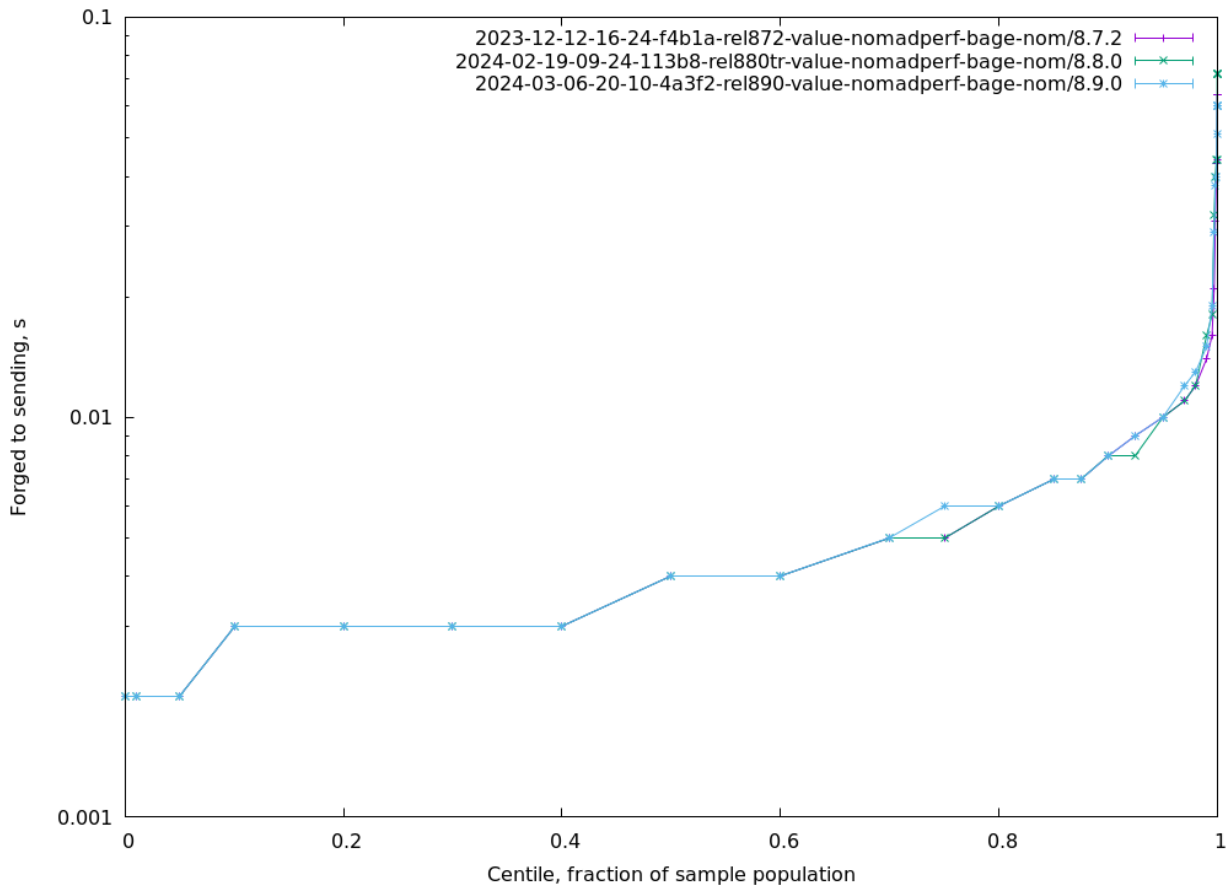
**Forged to announced (cdfForgerAnnounce)** Time between block forging completion and header announcement (ChainSyncServerEvent.TraceChainSyncServerRead.AddBlock)

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

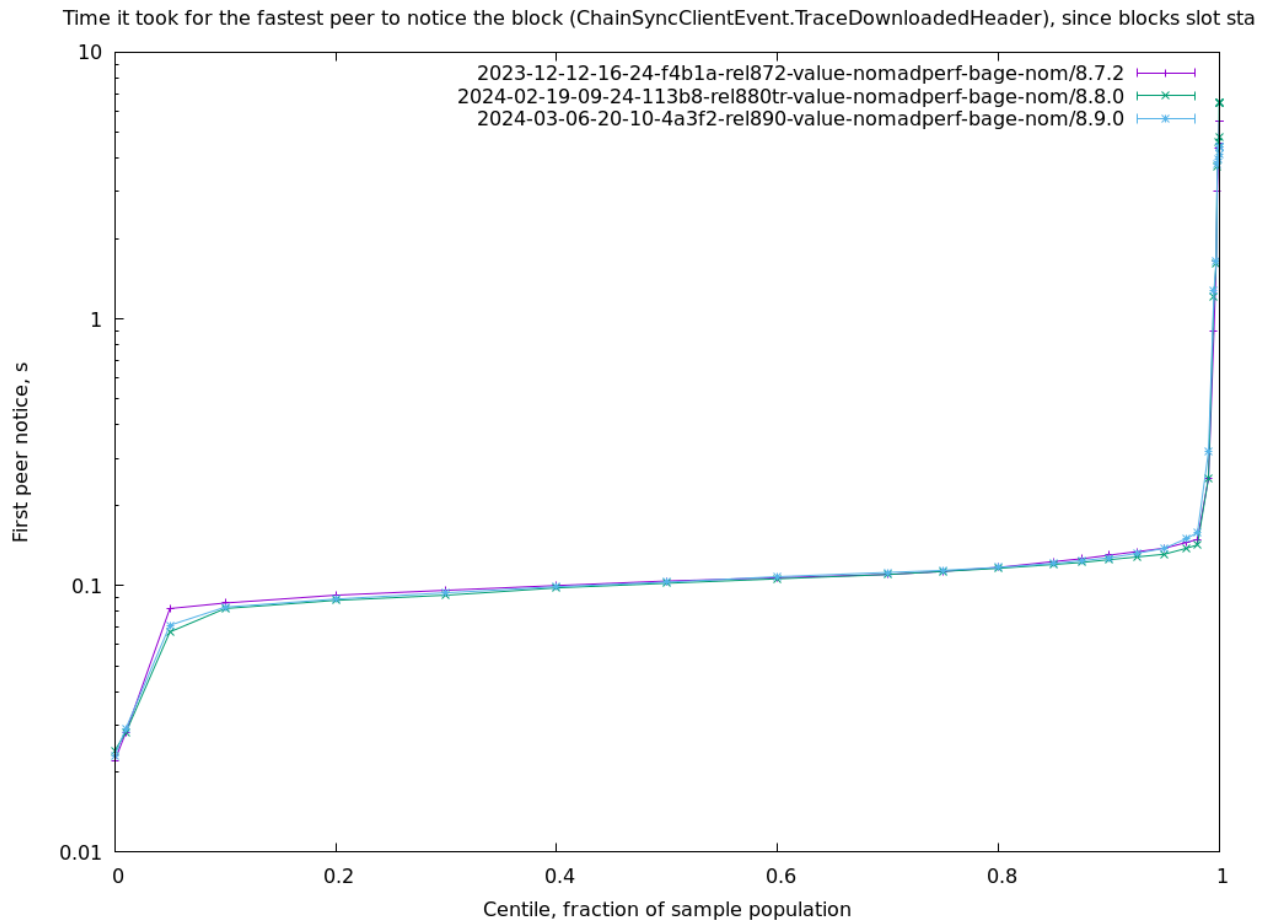


**Forged to sending (cdfForgerSend)** Time between block forging completion and begin-of-sending (TraceBlockFetchServerSendBlock)

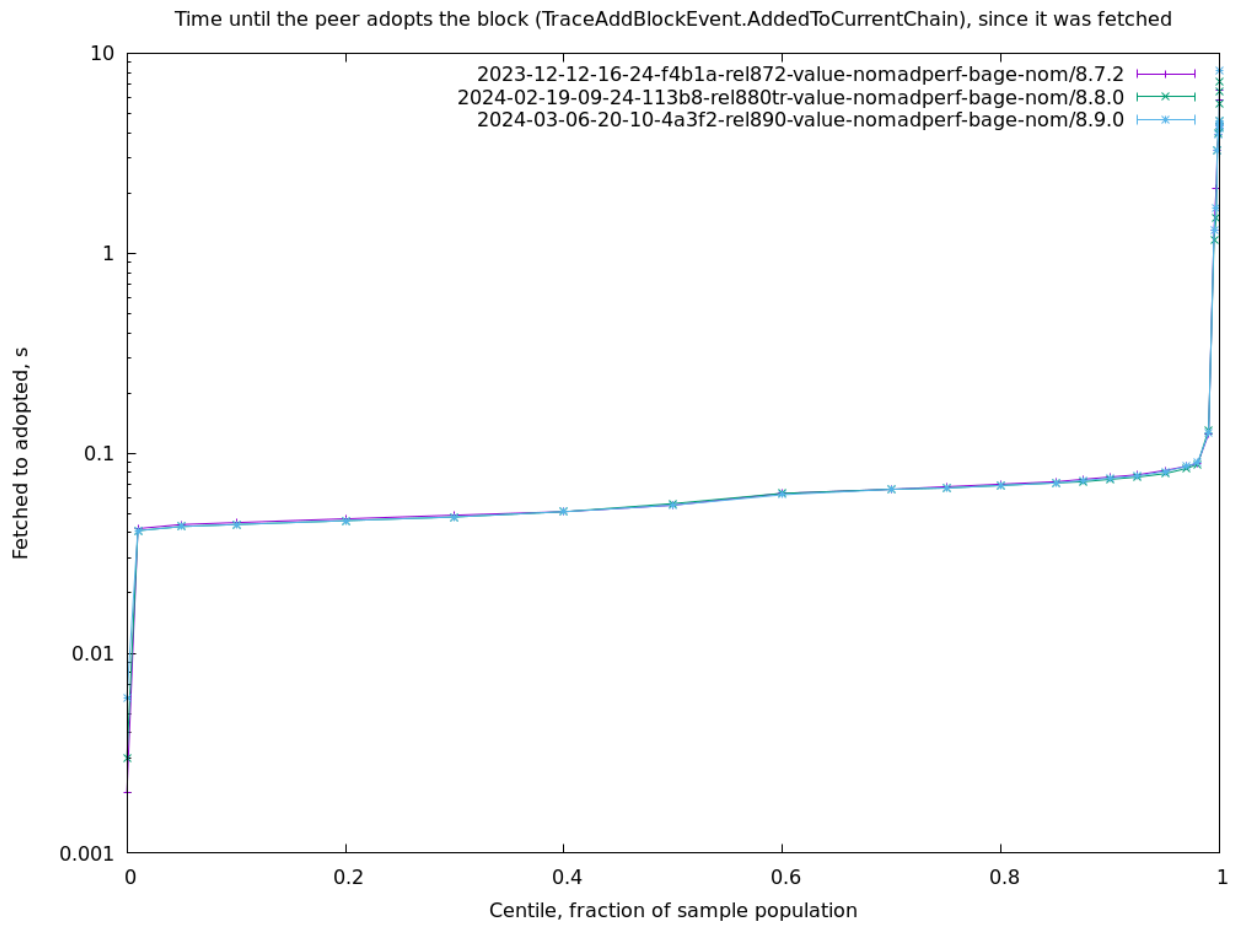
Time between block forging completion and begin-of-sending (TraceBlockFetchServerSendBlock)



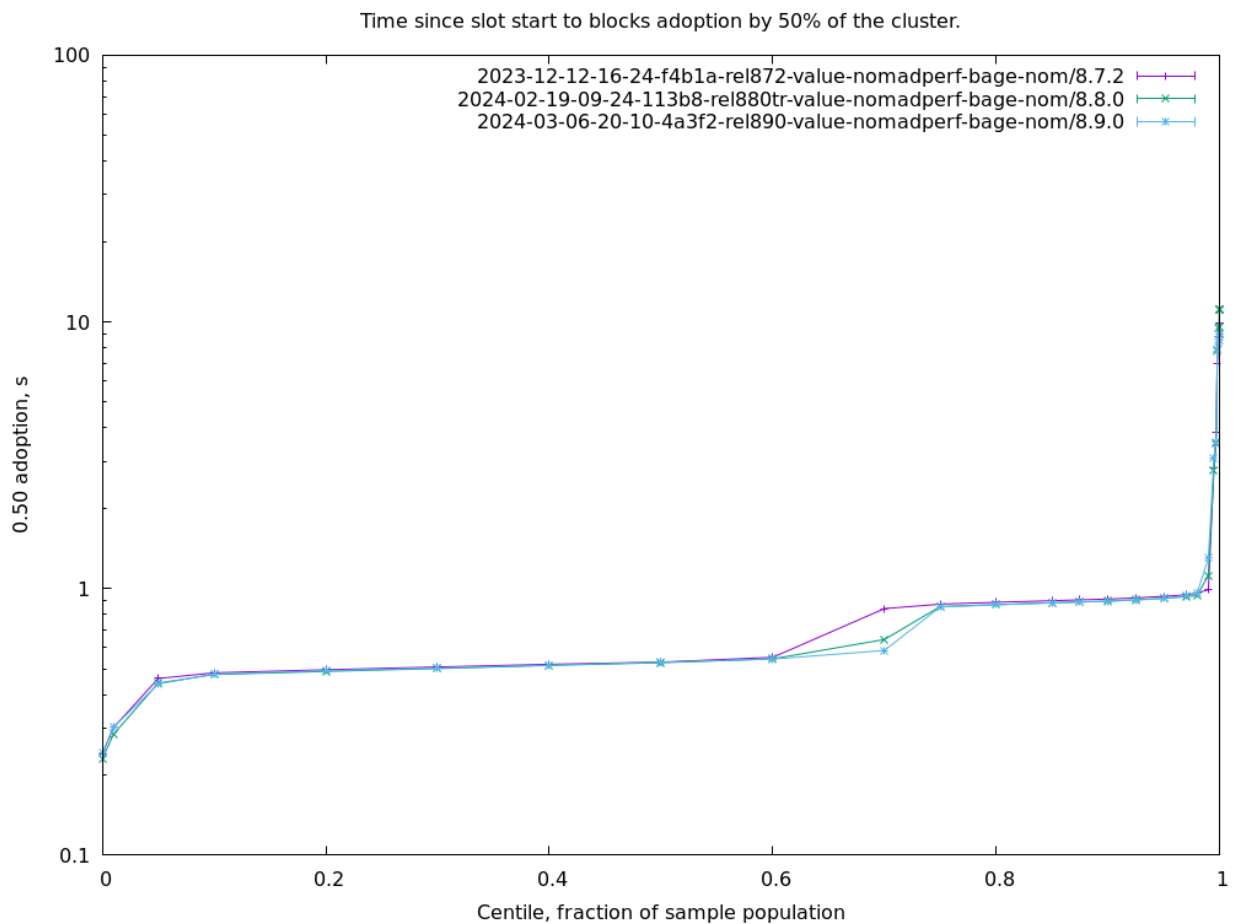
**First peer notice (cdfPeerNoticeFirst)** Time it took for the fastest peer to notice the block (ChainSyncClientEvent.TraceDownloadedHeader), since block's slot start



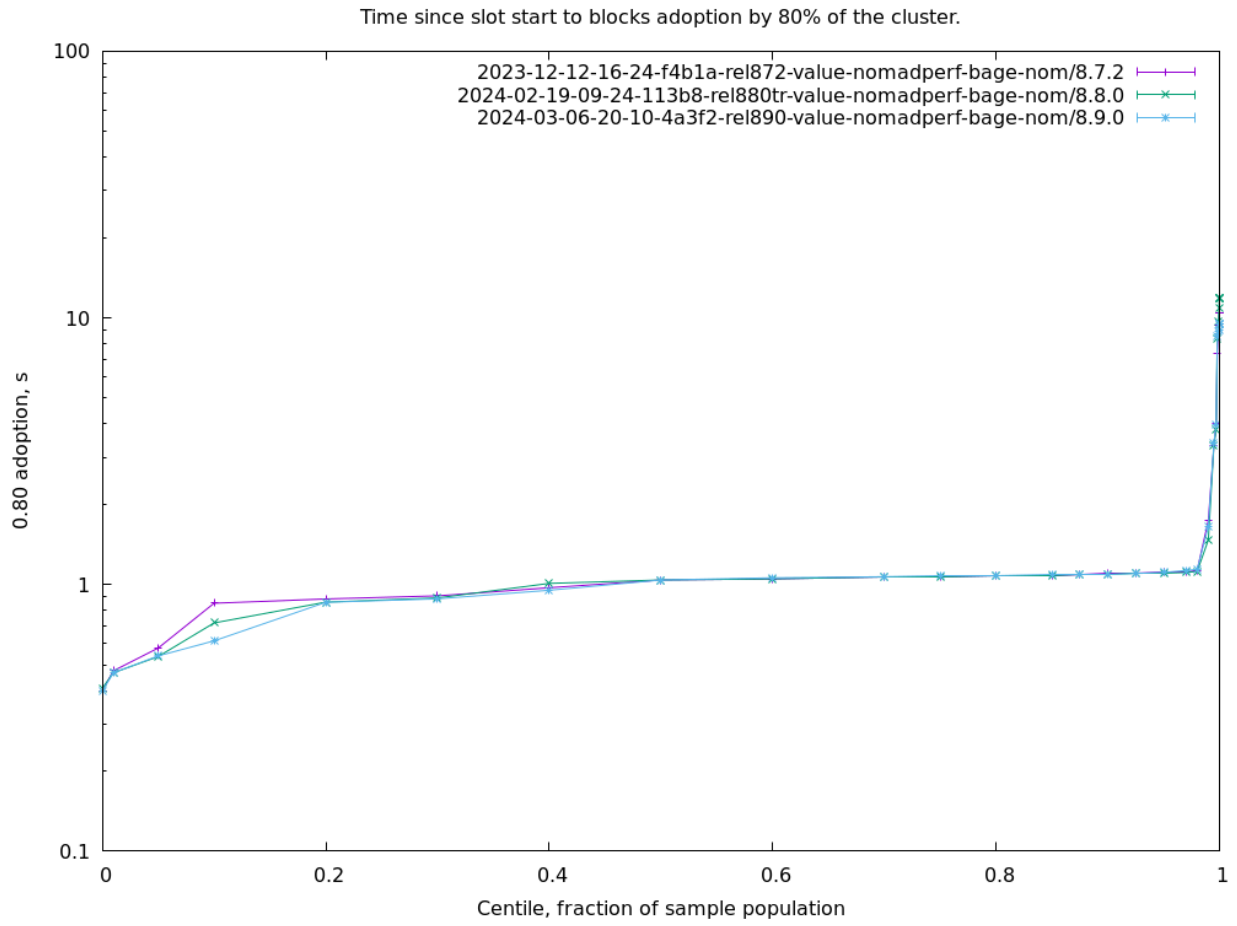
**Fetches to adopted (cdfPeerAdoption)** Time until the peer adopts the block (TraceAddBlockEvent.AddedToCurrentChain), since it was fetched



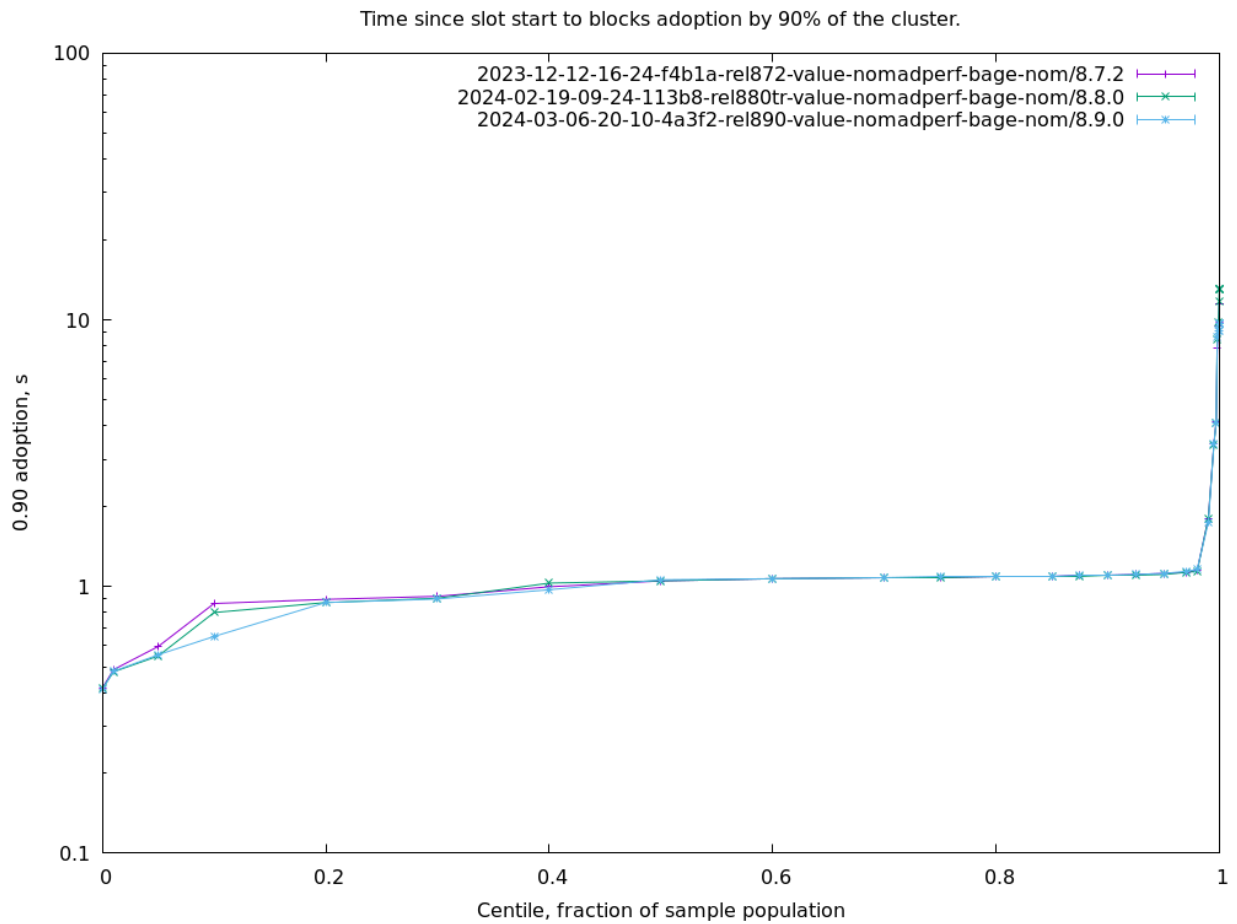
**0.50 adoption (cdf0.50)** Time since slot start to block's adoption by 50% of the cluster.



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



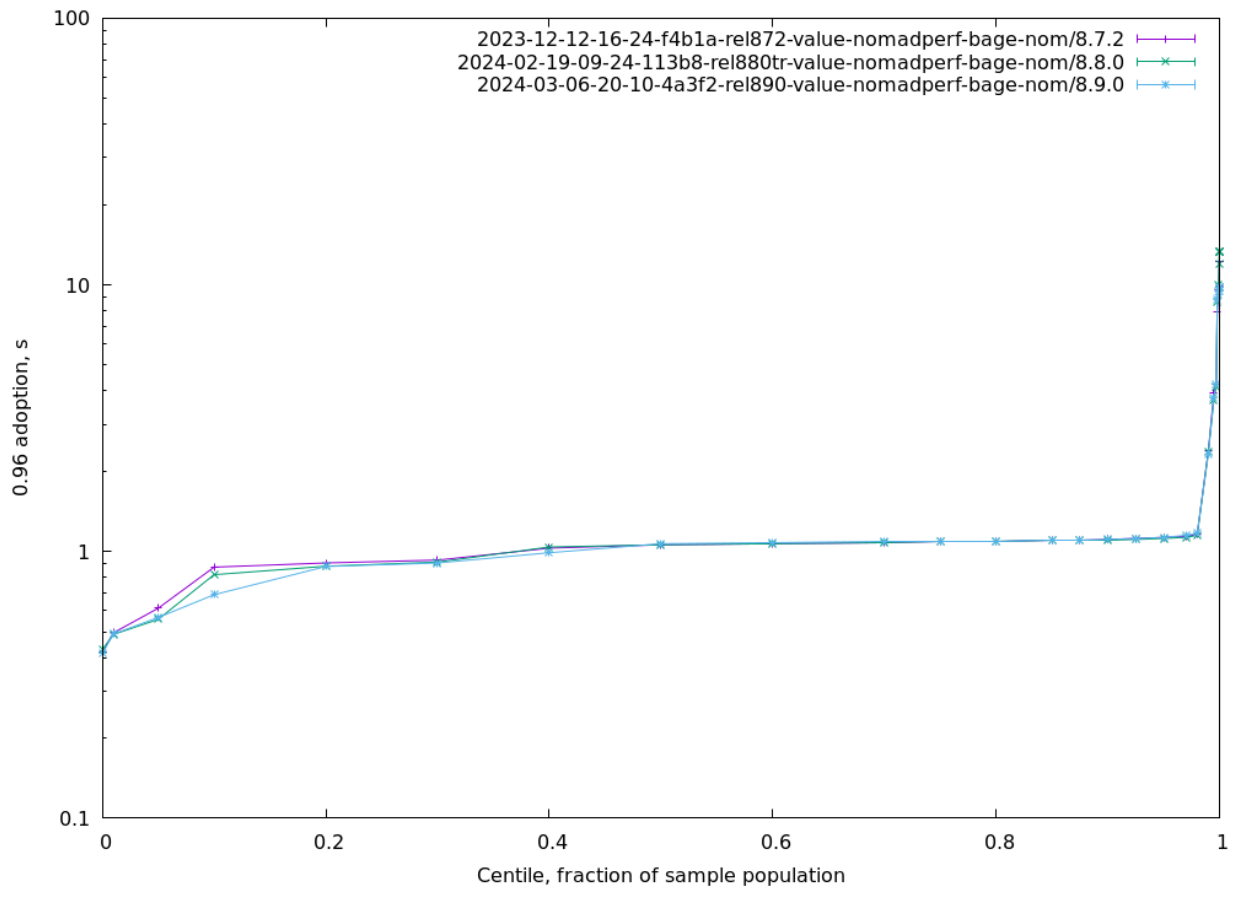
**0.90 adoption (cdf0.90)** Time since slot start to block's adoption by 90% of the cluster.



**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