

8.9.0 and 8.8.0 against 8.7.2

Plutus countdown loop 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 Plutus countdown loop workload.

	8.7.2	8.8.0	8.9.0
Analysis date	2023-12-21	2024-02-21	2024-03-08
Cluster system start date	2023-12-20	2024-02-20	2024-03-07
Cluster system start time	10:54:40	09:22:26	17:22:11
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	1	1	1
Tx Outputs	1	1	1
TPS	0.85	0.85	0.85
Transaction count	61200	61200	61200
Plutus script	Loop	Loop	Loop
Machines	52	52	52
Number of filters applied	4	4	4
Log text lines emitted per host	663138.17307	735146.69230	742063.40384
Log objects emitted per host	663108.17307	735116.69230	742033.40384
Log objects analysed per host	544638.0	544516.28846	548548.86538
Host run time, s	71922.4	71911.7	71933.8
Host log line rate, Hz	9.2202	10.222	10.315
Total log objects analysed	28321176	28314847	28524541
Run time, s	71927	71917	71940
Analysed run duration, s	56018	56010	56026
Run time efficiency	0.77	0.77	0.77
Node start spread, s	10.126933	11.014852	10.477582
Node stop spread, s	1.4301309	3.3131731	3.1893302
Perf analysis start spread, s	0	0	0
Perf analysis stop spread, s	1	3	4
Slots analysed	56017	56008	56023
Blocks analysed	2688	2704	2806
Blocks rejected	840	838	876

# Chapter 2

## Analysis

### 2.1 Resource Usage

	8.7.2	8.8.0	$\Delta$	$\Delta\%$	8.9.0	$\Delta$	$\Delta\%$
Forge loop starts, #	0.99908	0.9991	0.000	0	0.99906	-0.000	0
Process CPU usage, %	5.6345	5.6132	-0.021	0	5.947	0.313	6
RTS GC CPU usage, %	0.58154	0.58168	0.000	0	0.62191	0.040	7
RTS Mutator CPU usage, %	5.0468	5.0145	-0.032	-1	5.3037	0.257	5
Major GCs, #	0.00092	0.00093	0.000	0	0.00094	0.000	0
Minor GCs, #	1.5744	1.578	0.004	0	1.5958	0.021	1
Kernel RSS, MB	8325.8	8278.9	-46.900	-1	8279.9	-45.900	-1
RTS heap size, MB	8276.3	8227.9	-48.400	-1	8228.7	-47.600	-1
RTS live GC dataset, MB	3158.0	3237.0	79.000	3	3223.0	65.000	2
RTS alloc rate, MB/s	48.643	48.786	0.143	0	49.295	0.652	1
Filesystem reads, KB/s	0.0	0.0	0.000	nan	0.0	0.000	nan
Filesystem writes, KB/s	202.46	202.38	-0.080	0	203.55	1.090	1
CPU 85% spans, slots	0.16865	0.17932	0.011	7	0.18846	0.020	12
Sample count	(291>)	(291>)			(291>)		

### 2.2 Anomaly control

	8.7.2	8.8.0	$\Delta$	$\Delta\%$	8.9.0	$\Delta$	$\Delta\%$
Blocks per host, blocks	69.634	69.942	0.308	0	72.288	2.654	4
Filtered to chained block ratio, /	0.7614	0.76369	0.002	0	0.76297	0.002	0
Chained to forged block ratio, /	0.97424	0.97367	-0.001	0	0.97947	0.005	1
Height & slot battles, blocks	0.00111	0.0	-0.001	-90	0.0	-0.001	-90
Block size, B	2948.0	2948.0	0.000	0	2948.0	0.000	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	0.00116	0.0018	0.001	86	0.00118	0.000	0
Acquired block context, s	0.02084	0.02099	0.000	0	0.02222	0.001	5
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	2e-05	0.000	0
Leadership check duration, s	0.00037	0.00039	0.000	0	0.0004	0.000	0
Ledger ticking, s	0.02088	0.01817	-0.003	-14	0.01976	-0.001	-5
Mempool snapshotting, s	0.06599	0.06525	-0.001	-2	0.06747	0.001	2
Leadership to forged, s	0.00039	0.0004	0.000	0	0.00041	0.000	0
Forged to announced, s	0.00052	0.00052	0.000	0	0.00053	0.000	0
Forged to sending, s	0.00497	0.00448	-0.000	0	0.0045	-0.000	0
Forged to self-adopted, s	0.05192	0.04901	-0.003	-6	0.04841	-0.004	-8
Slot start to announced, s	0.11027	0.10764	-0.003	-3	0.11208	0.002	2
Sample count	(2688)	(2704)			(2806)		

## 2.4 Individual peer propagation

	8.7.2	8.8.0	$\Delta$	$\Delta\%$	8.9.0	$\Delta$	$\Delta\%$
First peer notice, s	0.11182	0.10927	-0.003	-3	0.11371	0.002	2
First peer fetch, s	0.117	0.11367	-0.003	-3	0.11821	0.001	1
Notice to fetch request, s	0.00101	0.00102	0.000	0	0.00106	0.000	0
Fetch duration, s	0.12232	0.12198	-0.000	0	0.12257	0.000	0
Fetches to announced, s	5e-05	5e-05	0.000	0	5e-05	0.000	0
Fetches to sending, s	0.04125	0.04102	-0.000	0	0.04168	0.000	0
Fetches to adopted, s	0.05062	0.0502	-0.000	0	0.05058	-0.000	0
Sample count	(2688)	(2704)			(2806)		

## 2.5 End-to-end propagation

	8.7.2	8.8.0	$\Delta$	$\Delta\%$	8.9.0	$\Delta$	$\Delta\%$
0.50 adoption, s	0.36969	0.36504	-0.005	-1	0.37264	0.003	1
0.80 adoption, s	0.52861	0.5257	-0.003	-1	0.53074	0.002	0
0.90 adoption, s	0.53676	0.53559	-0.001	0	0.54222	0.005	1
0.92 adoption, s	0.53869	0.53793	-0.001	0	0.54499	0.006	1
0.94 adoption, s	0.54076	0.5402	-0.001	0	0.54775	0.007	1
0.96 adoption, s	0.54566	0.54376	-0.002	0	0.55134	0.006	1
0.98 adoption, s	0.55172	0.54761	-0.004	-1	0.55674	0.005	1
1.00 adoption, s	0.58157	0.56502	-0.017	-3	0.57749	-0.004	-1
Sample count	(2688)	(2704)			(2806)		

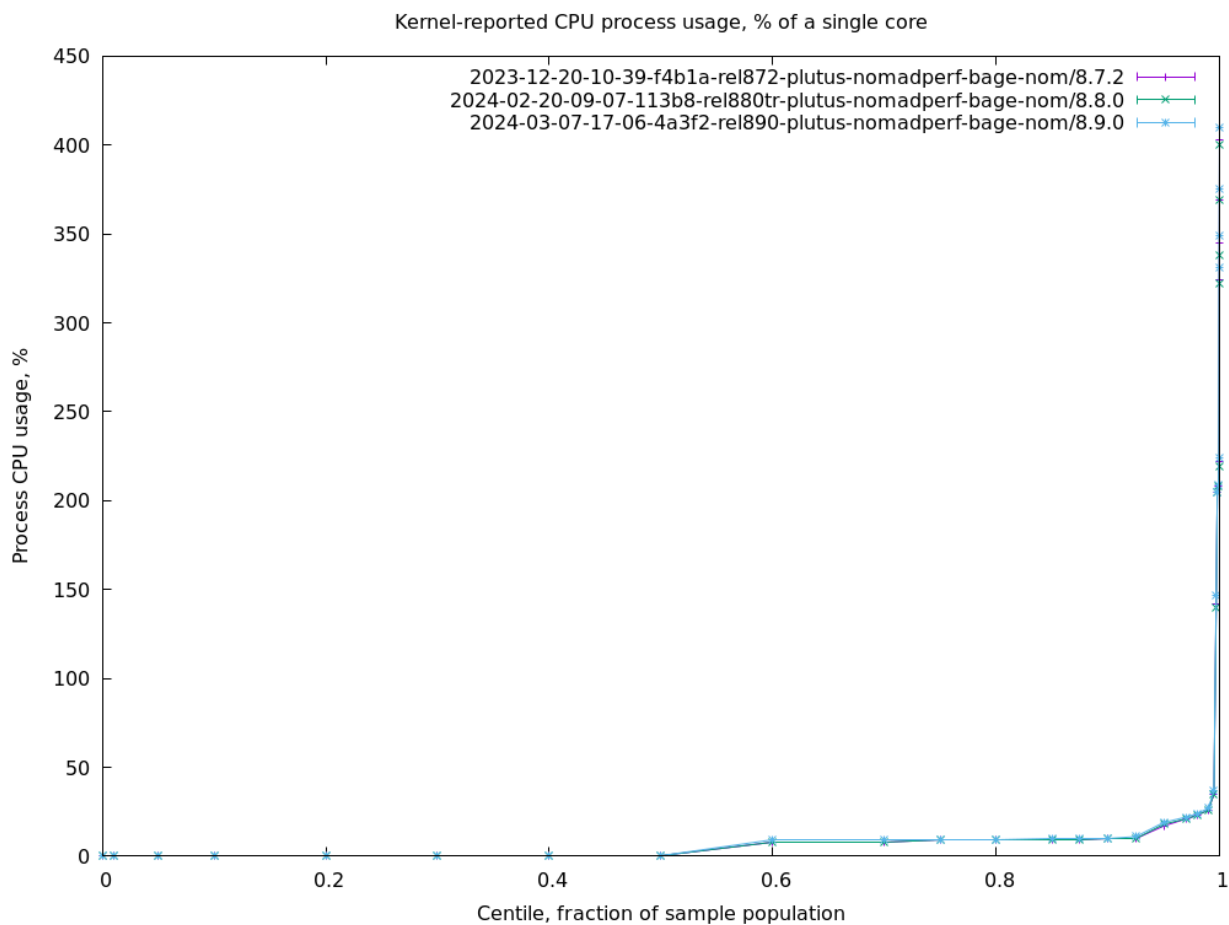
## Part I

# Appendix A: charts

# Chapter 3

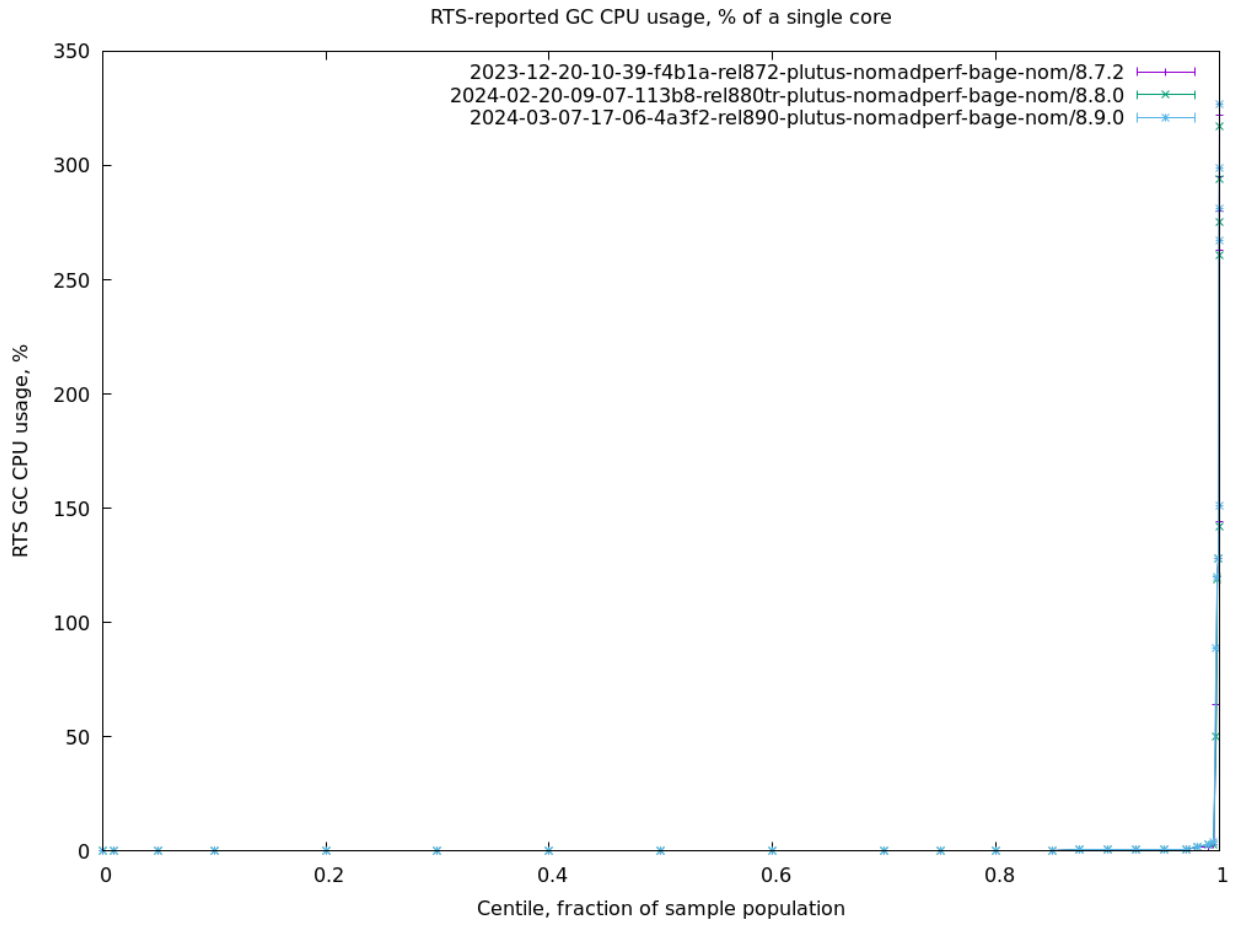
## Cluster performance charts

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

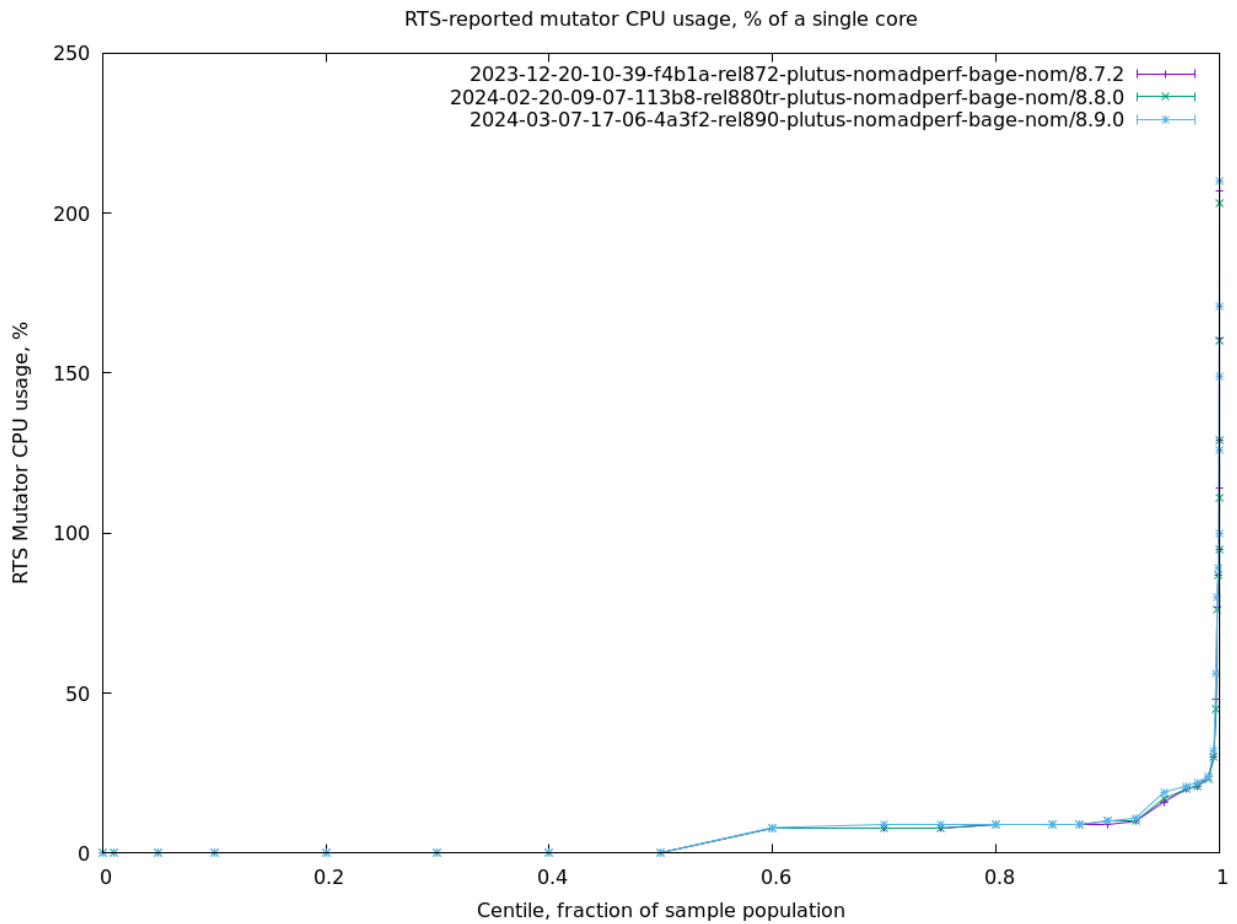


**RTS GC CPU usage (CentiGC)** RTS-reported GC CPU usage, % of a single core

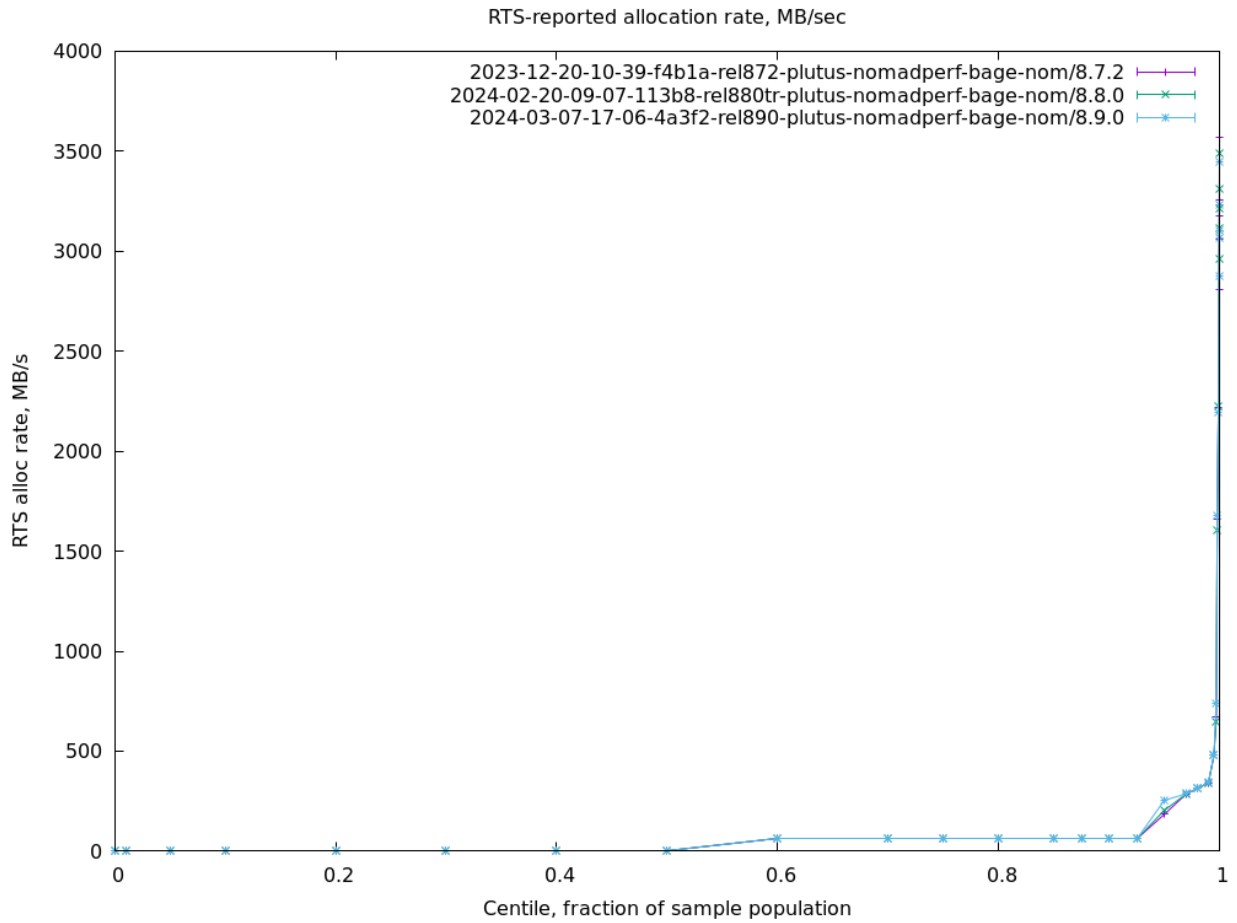




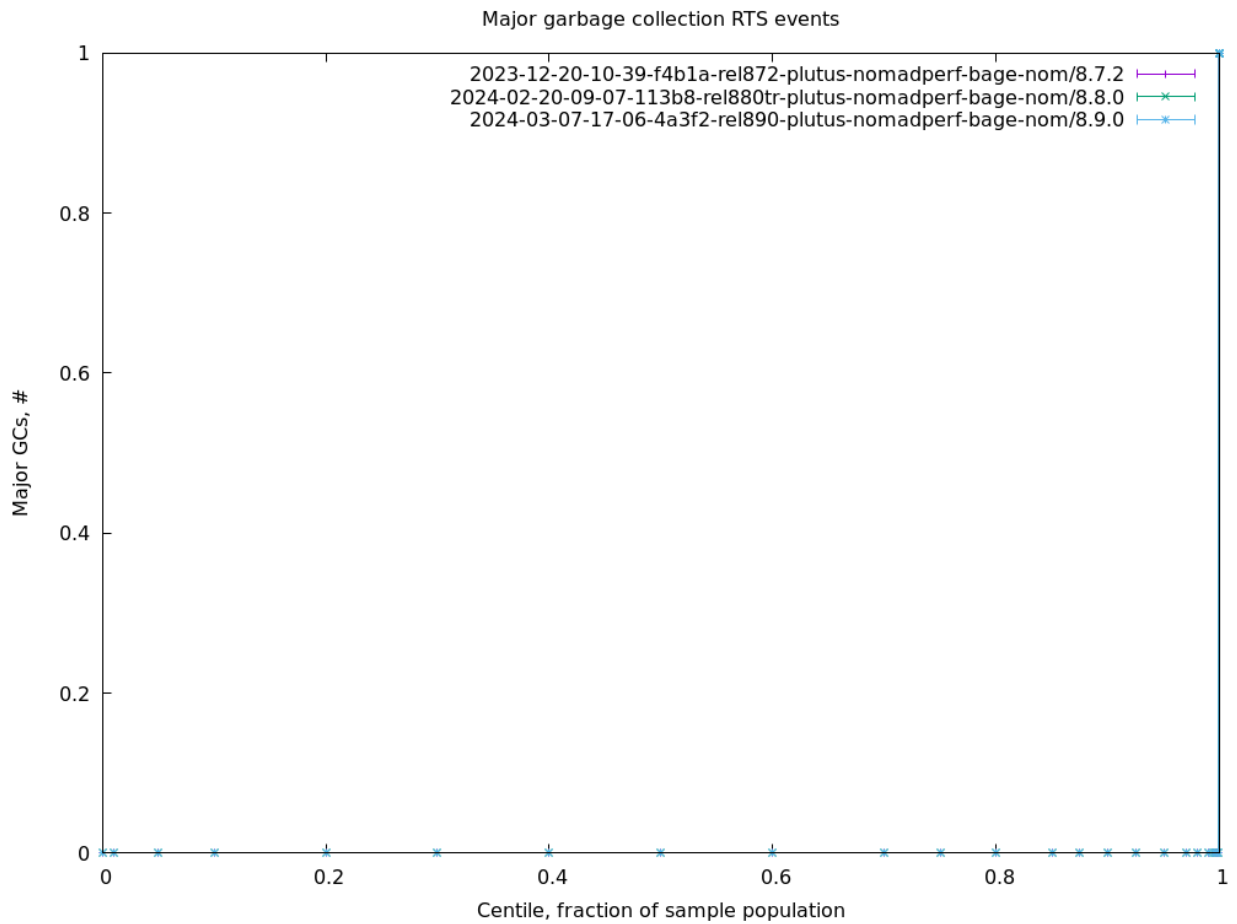
**RTS Mutator CPU usage (CentiMut)** RTS-reported mutator CPU usage, % of a single core



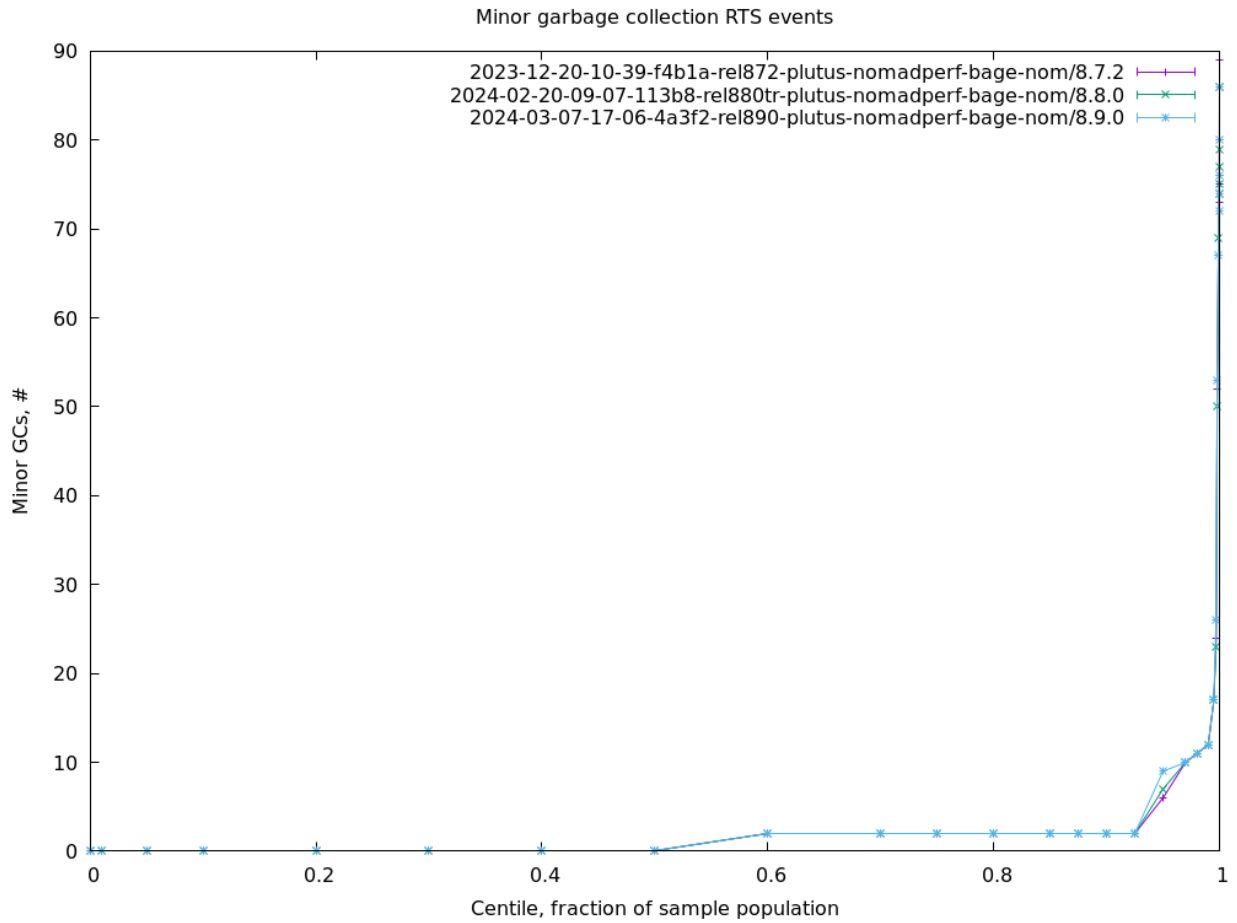
**RTS alloc rate (Alloc)** RTS-reported allocation rate, MB/sec



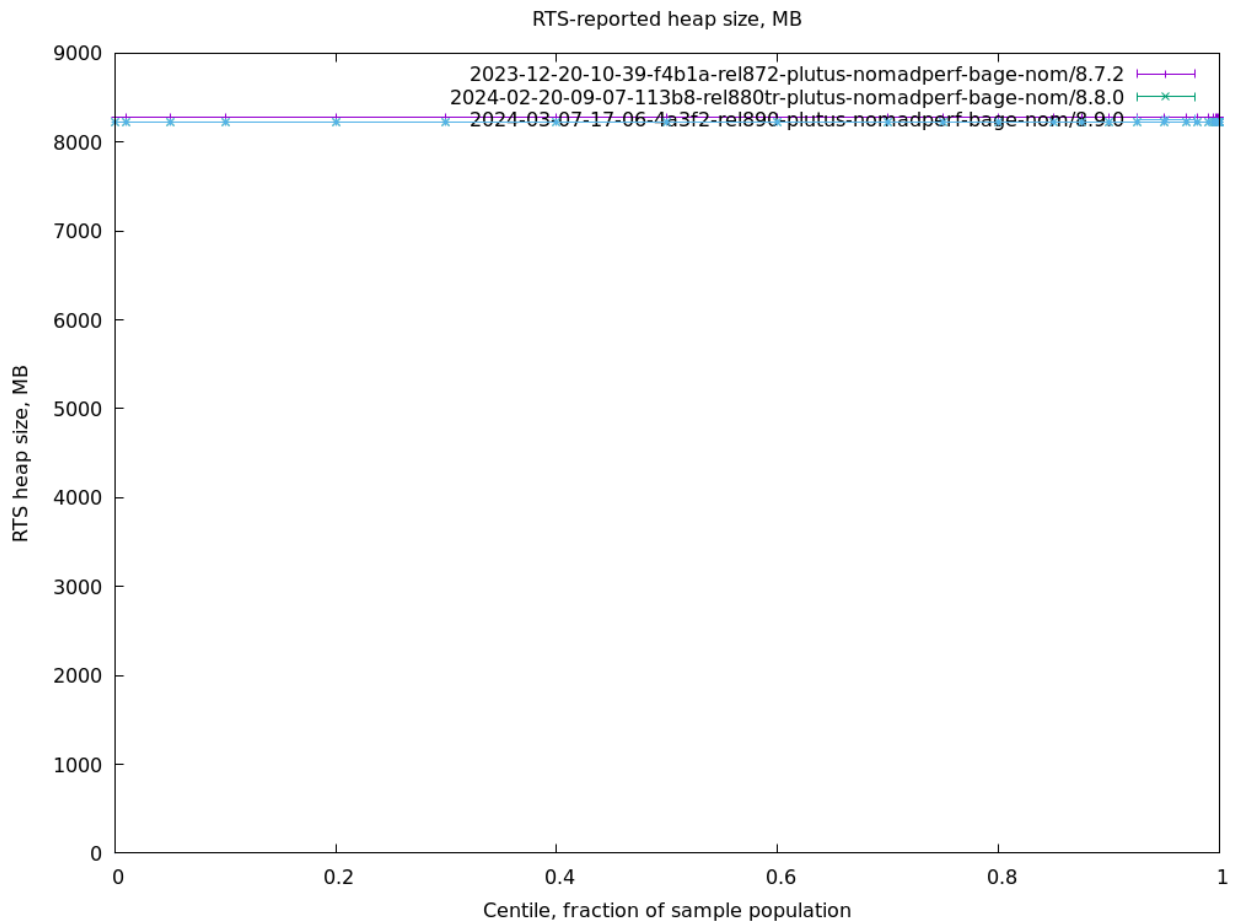
Major GCs (GcsMajor) Major garbage collection RTS events



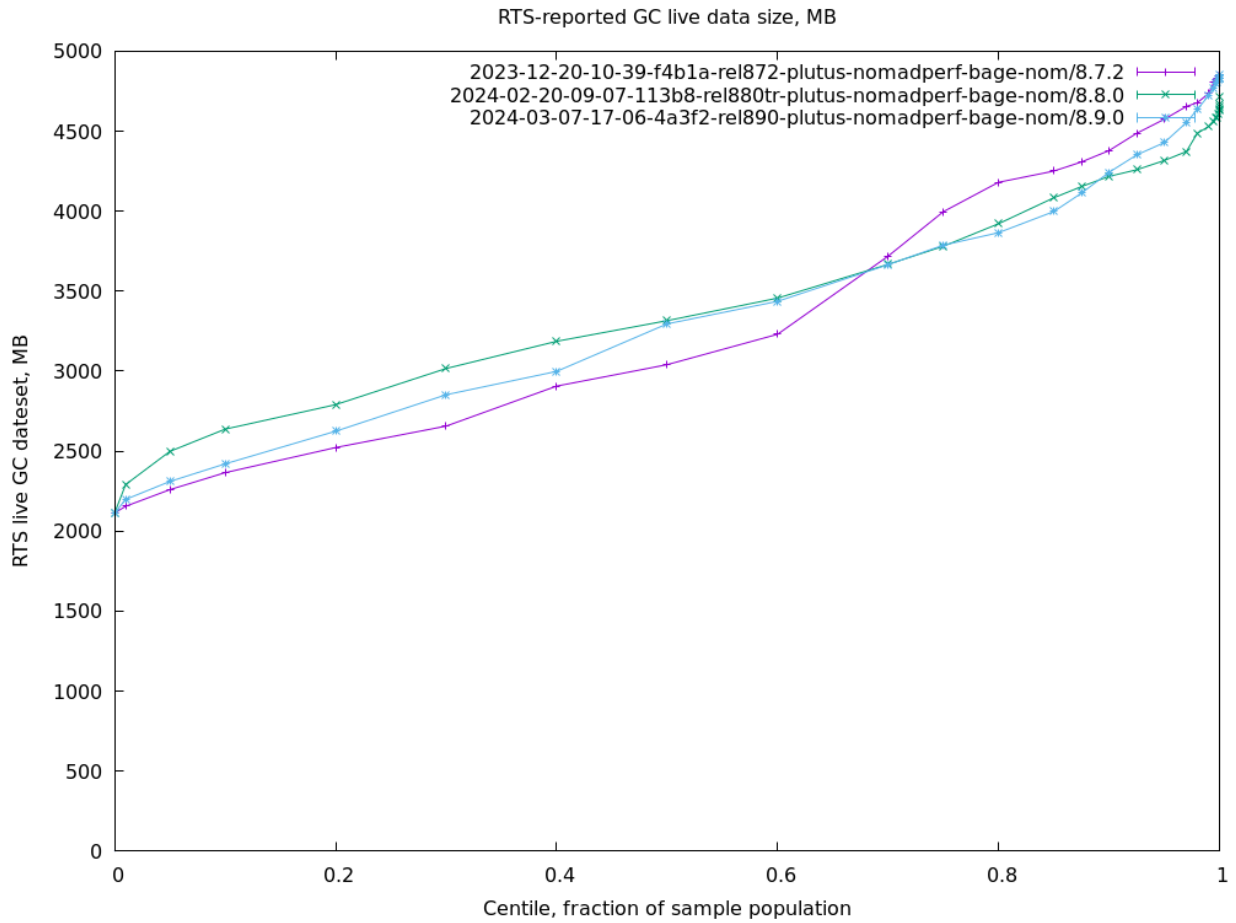
Minor GCs (GcsMinor) Minor garbage collection RTS events



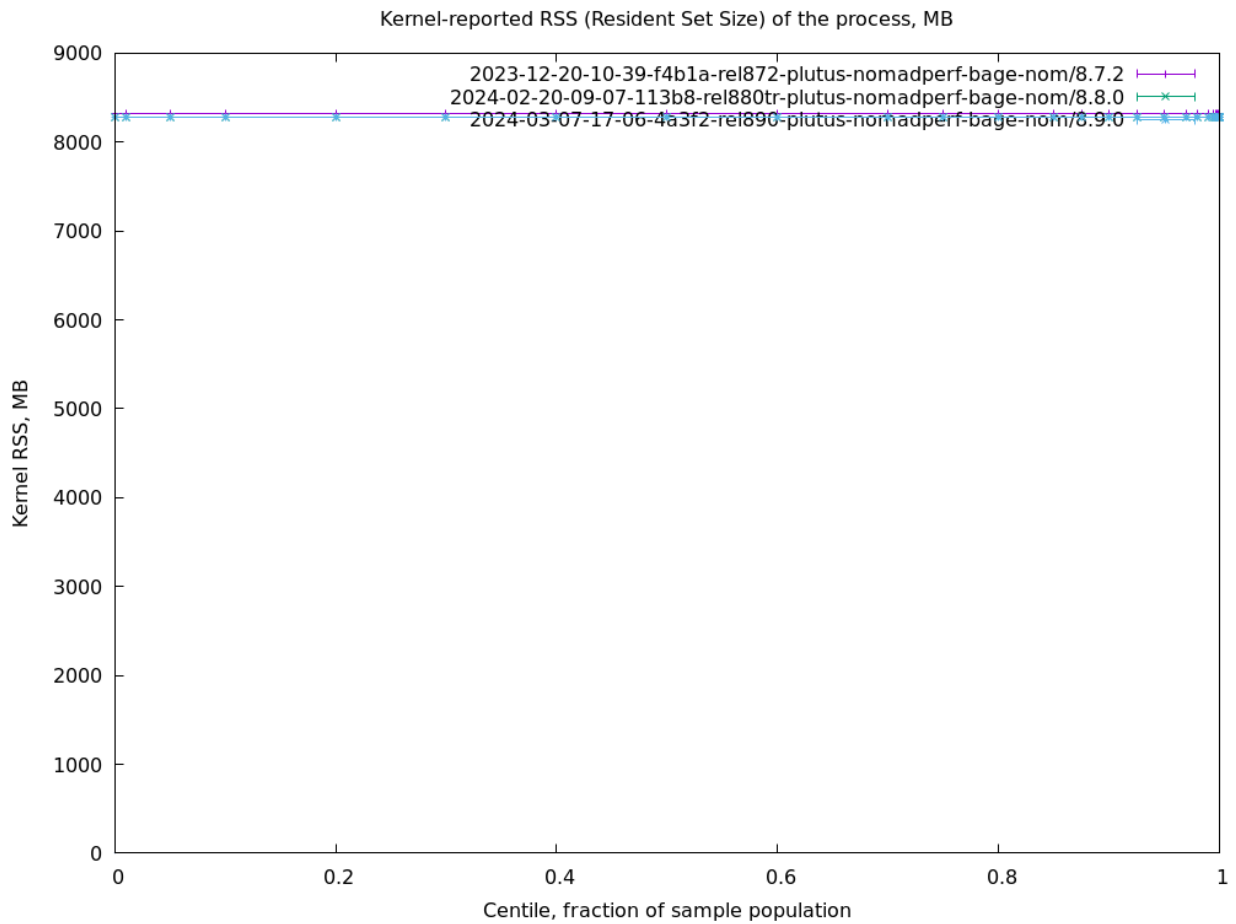
RTS heap size (Heap) RTS-reported heap size, MB



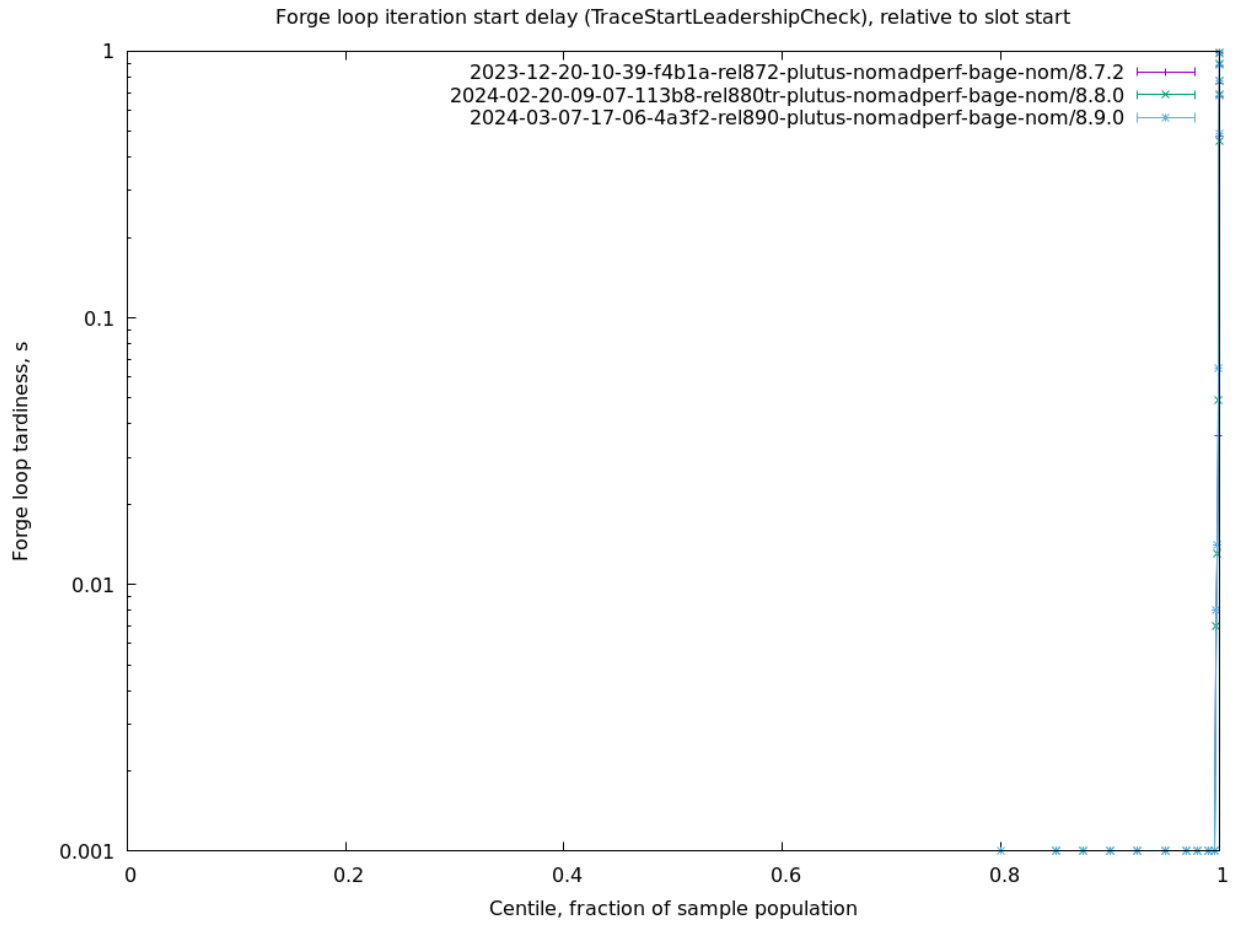
RTS live GC dataset (Live) RTS-reported GC live data size, MB



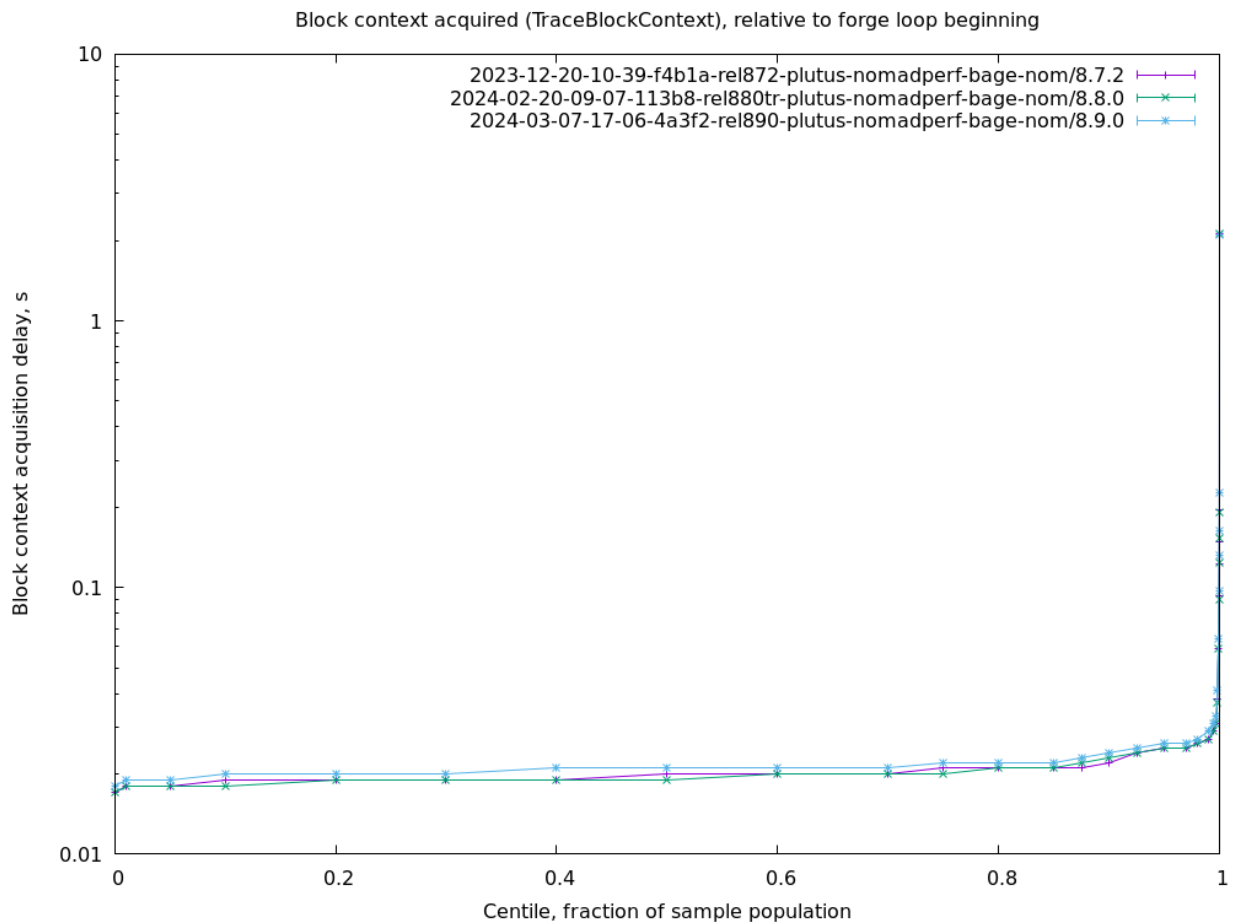
**Kernel RSS (RSS)** Kernel-reported RSS (Resident Set Size) of the process, MB



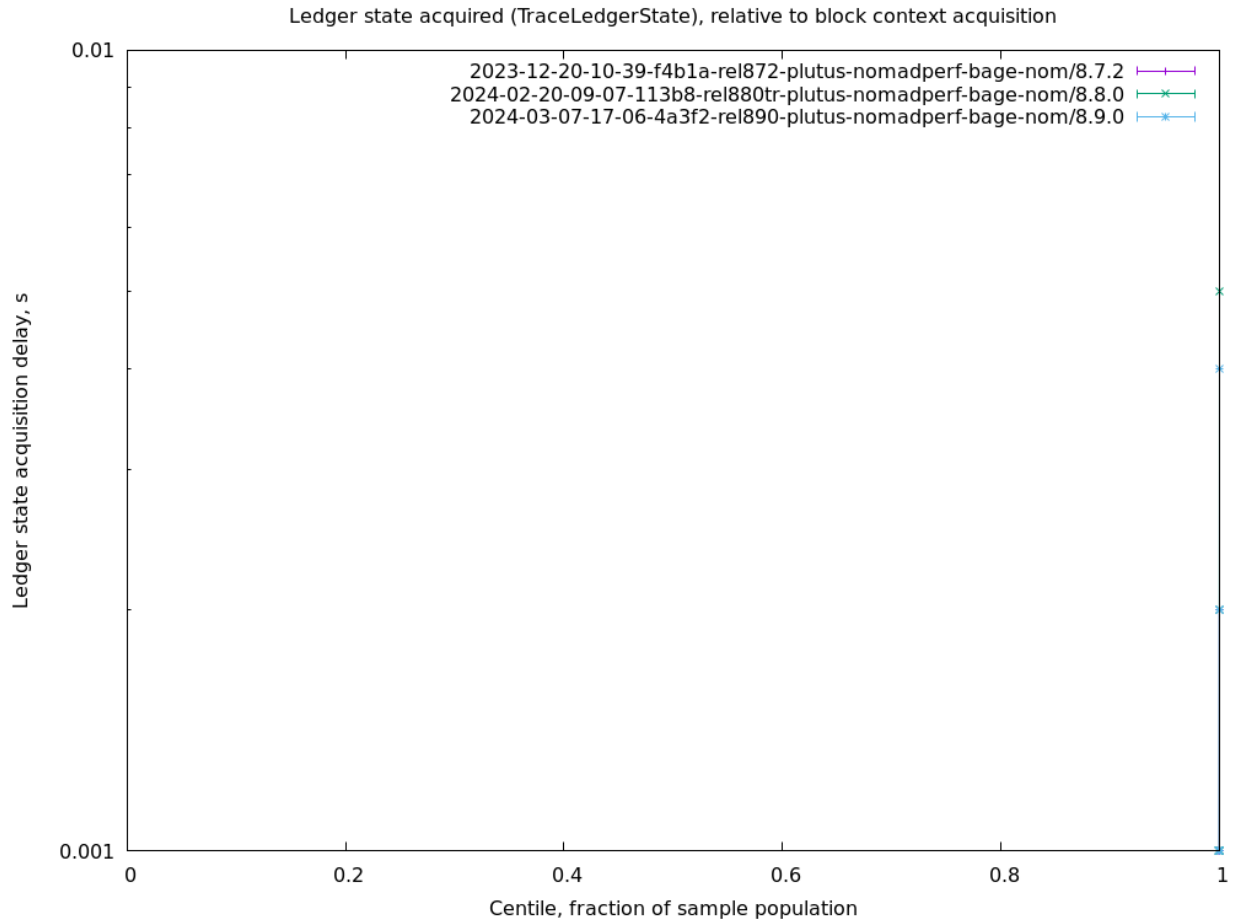
**Forge loop tardiness (cdfStarted)** Forge loop iteration start delay (TraceStartLeadershipCheck), relative to slot start



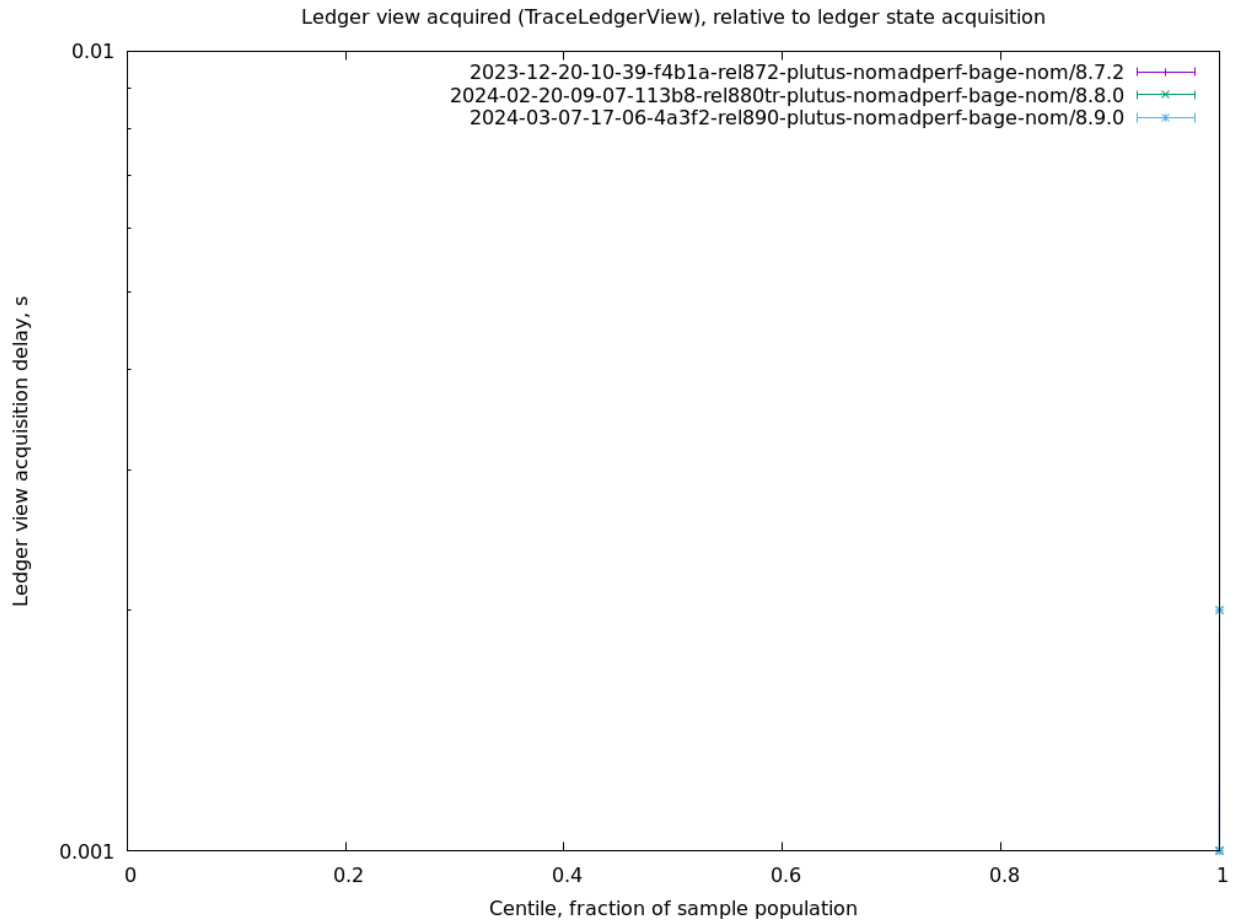
**Block context acquisition delay (cdfBlkCtx)** Block context acquired (TraceBlockContext), relative to forge loop beginning



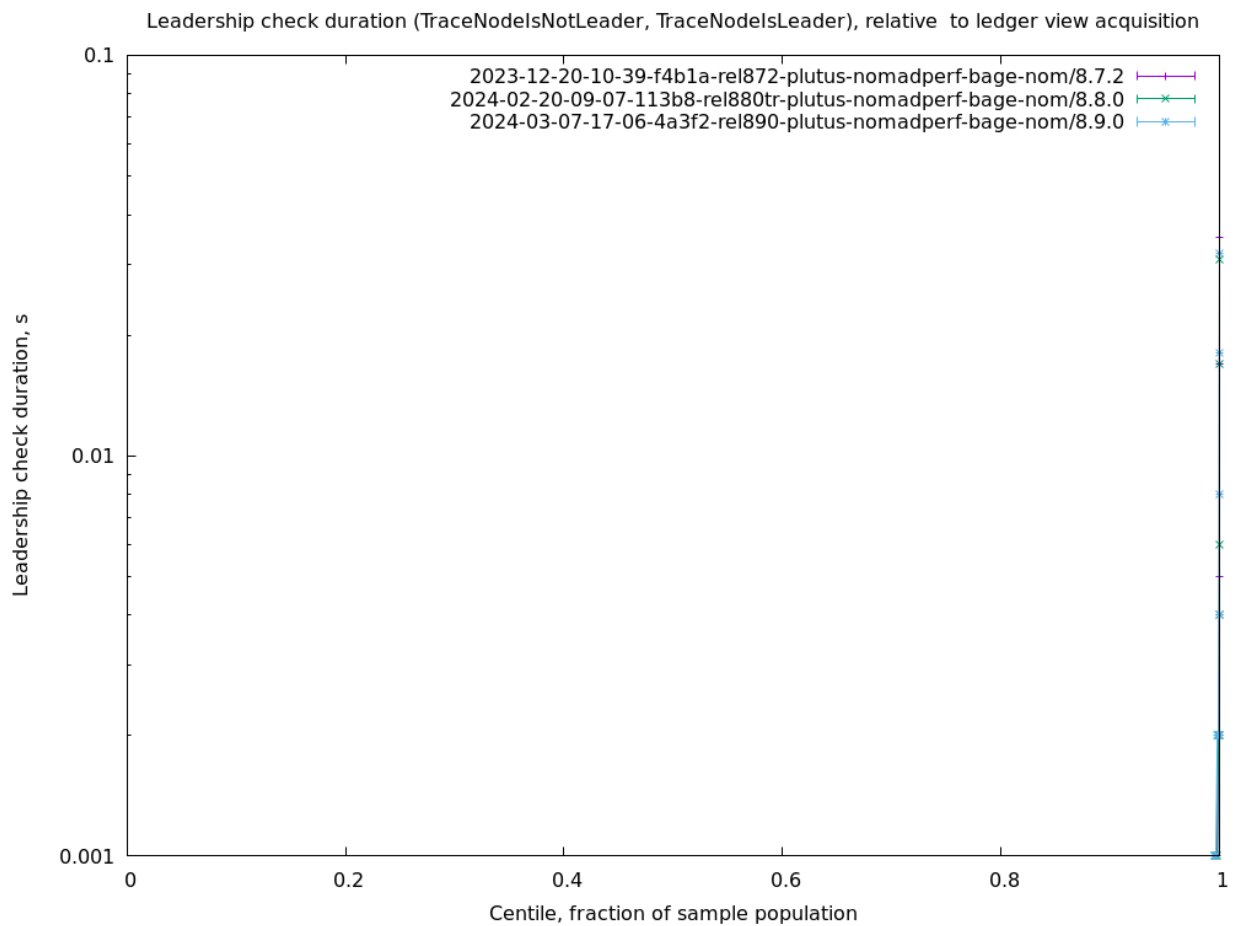
**Ledger state acquisition delay (cdfLgrState)** Ledger state acquired (TraceLedgerState), relative to block context acquisition



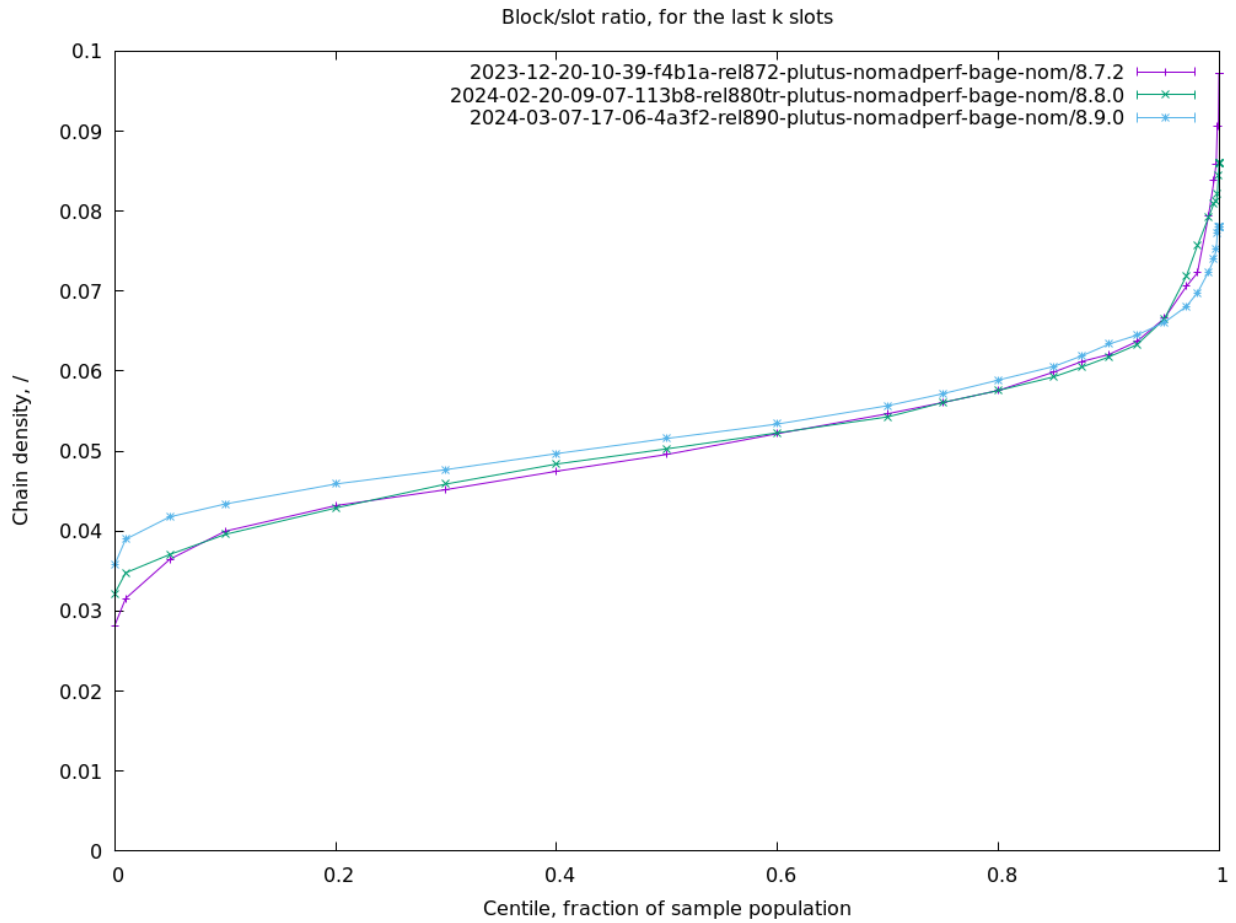
**Ledger view acquisition delay (cdfLgrView)** Ledger view acquired (TraceLedgerView), relative to ledger state acquisition



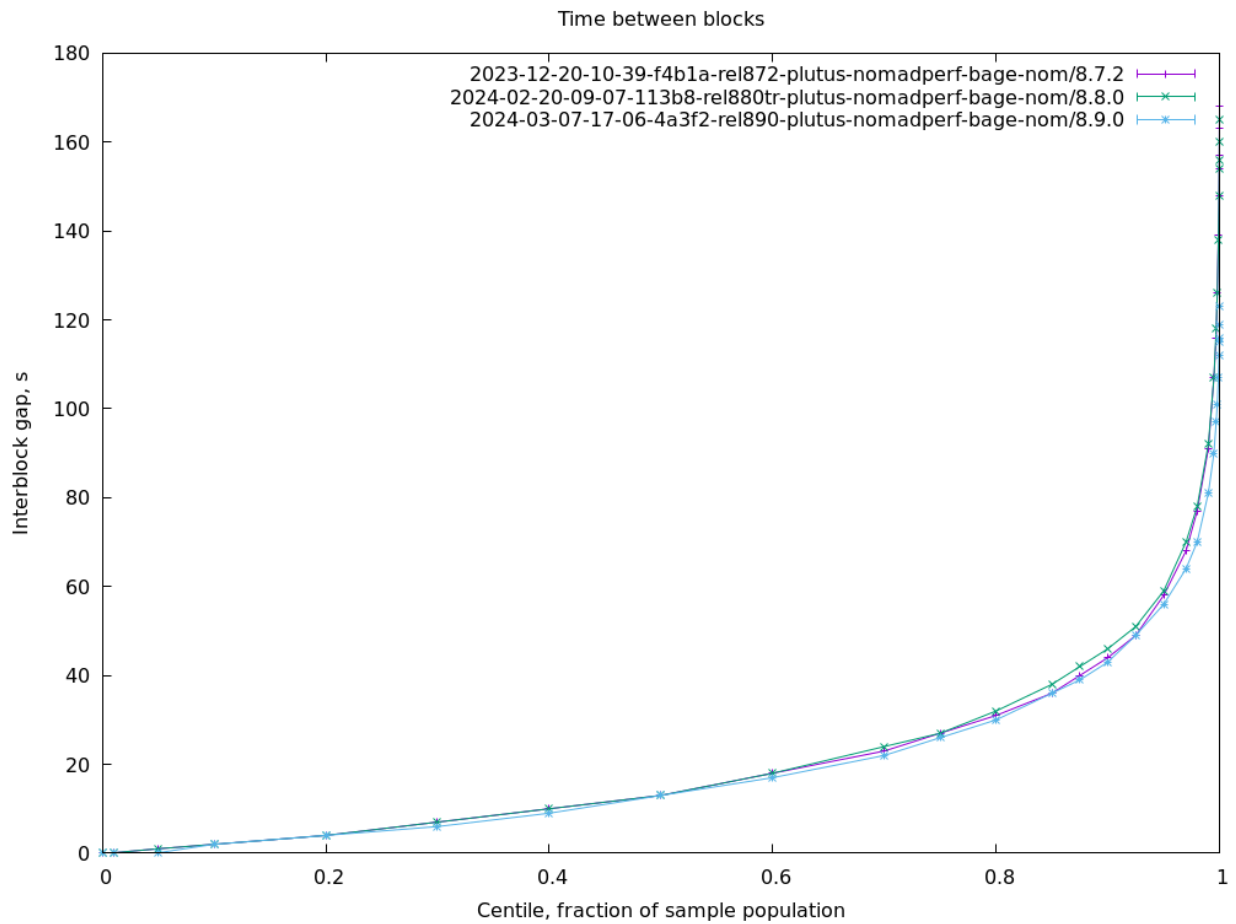
**Leadership check duration (cdfLeading)** Leadership check duration (TraceNodeIsNotLeader, TraceNodeIsLeader), relative to ledger view acquisition



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

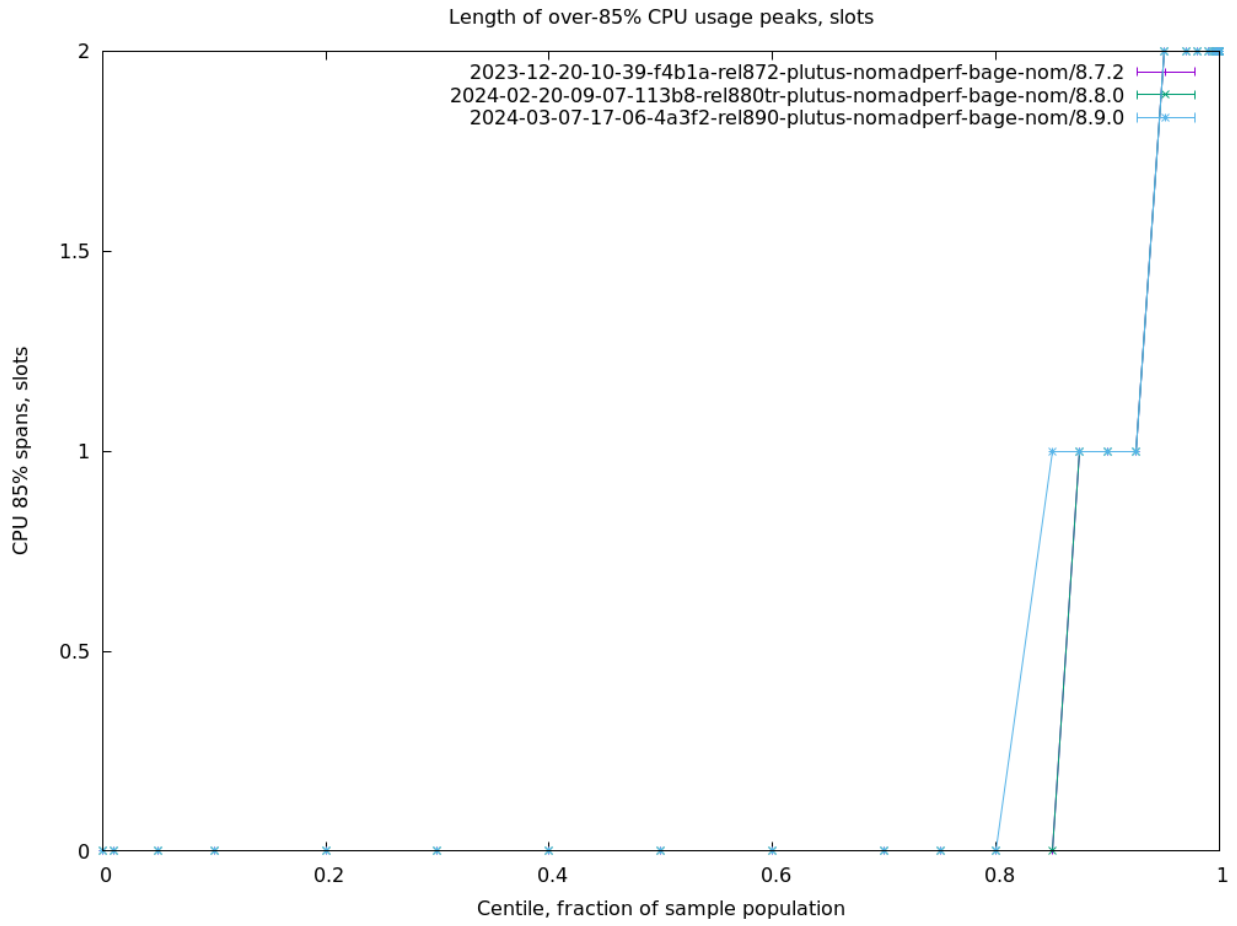


Interblock gap (cdfBlockGap) Time between blocks

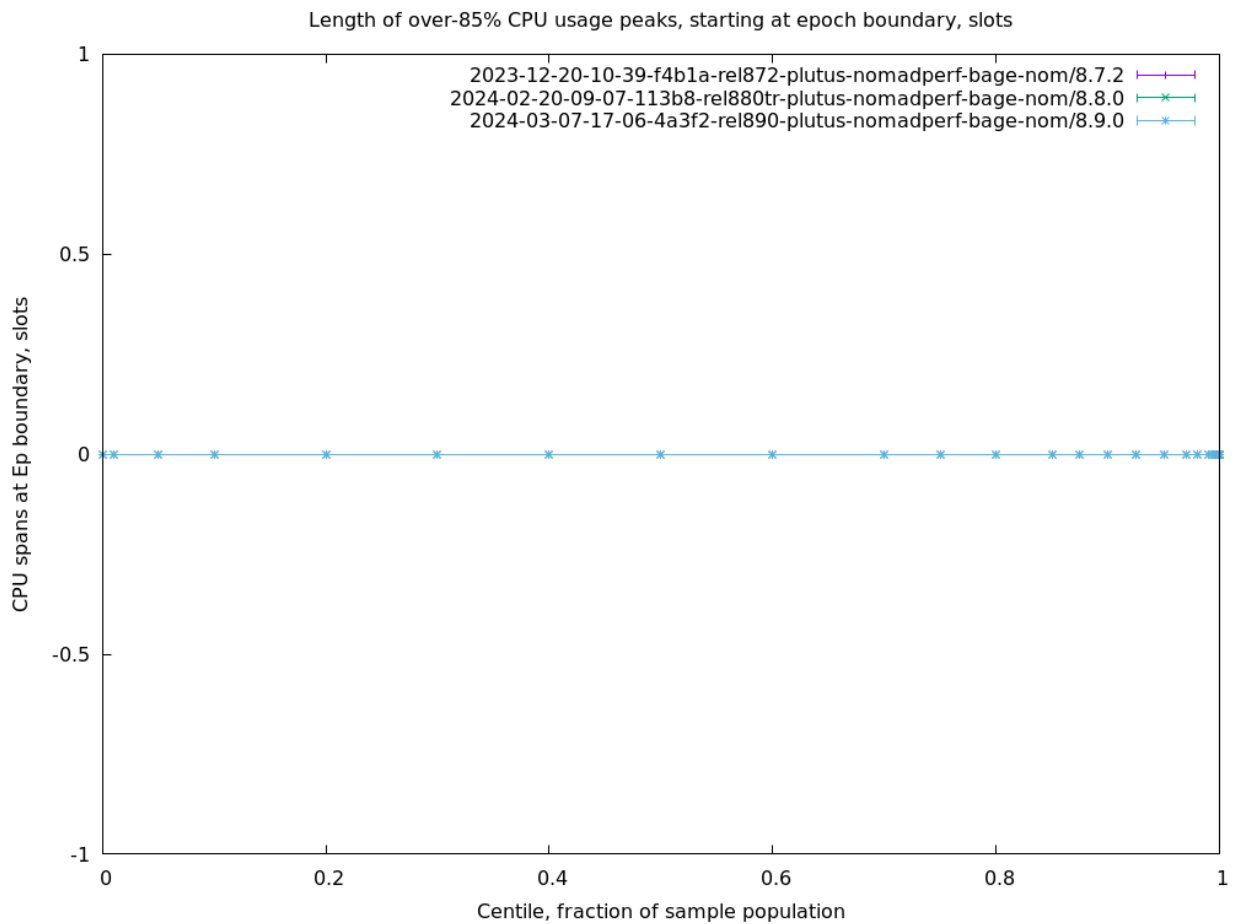


CPU 85% spans (cdfSpanLensCpu) Length of over-85% CPU usage peaks, slots

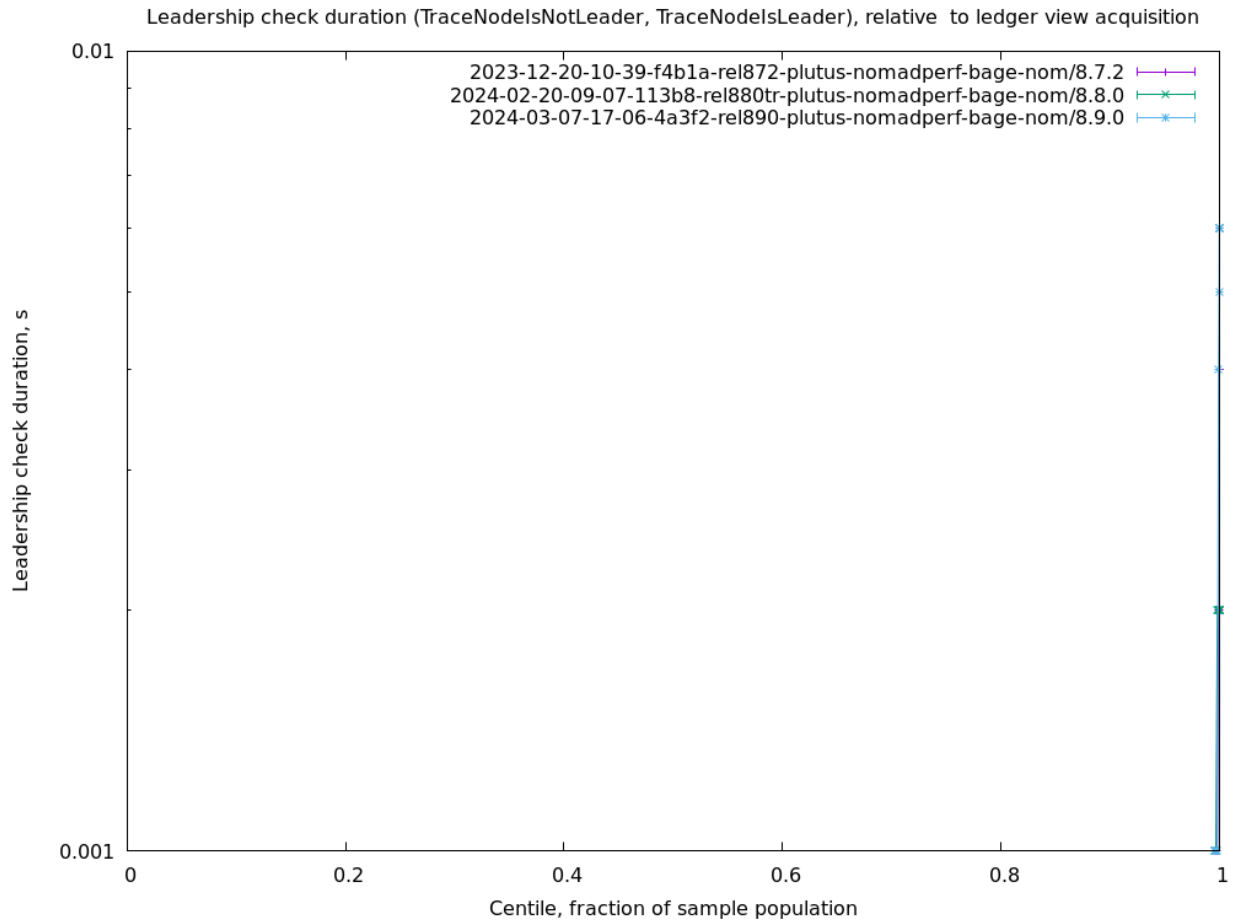




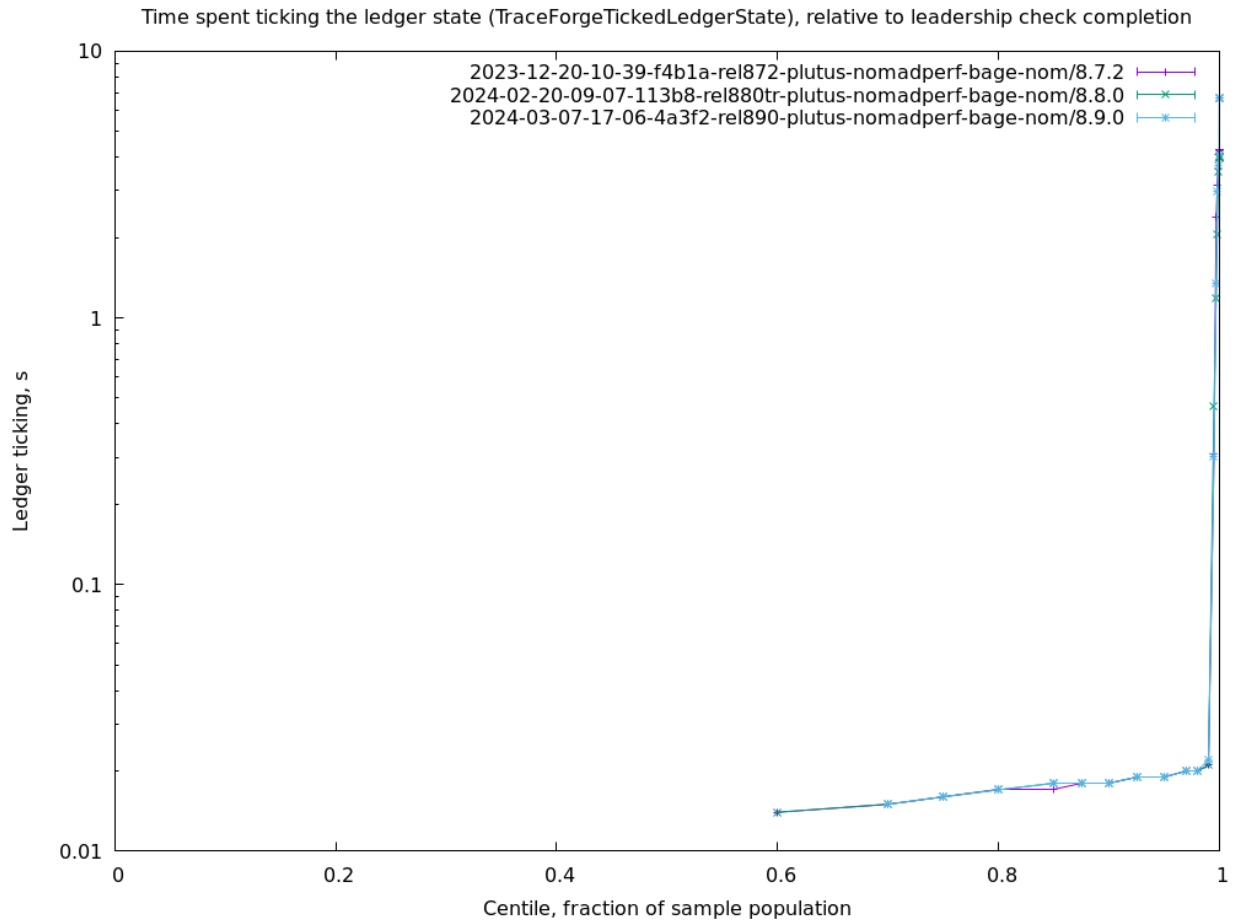
CPU spans at Ep boundary (cdfSpanLensCpuEpoch) Length of over-85% CPU usage peaks, starting at epoch boundary, slots



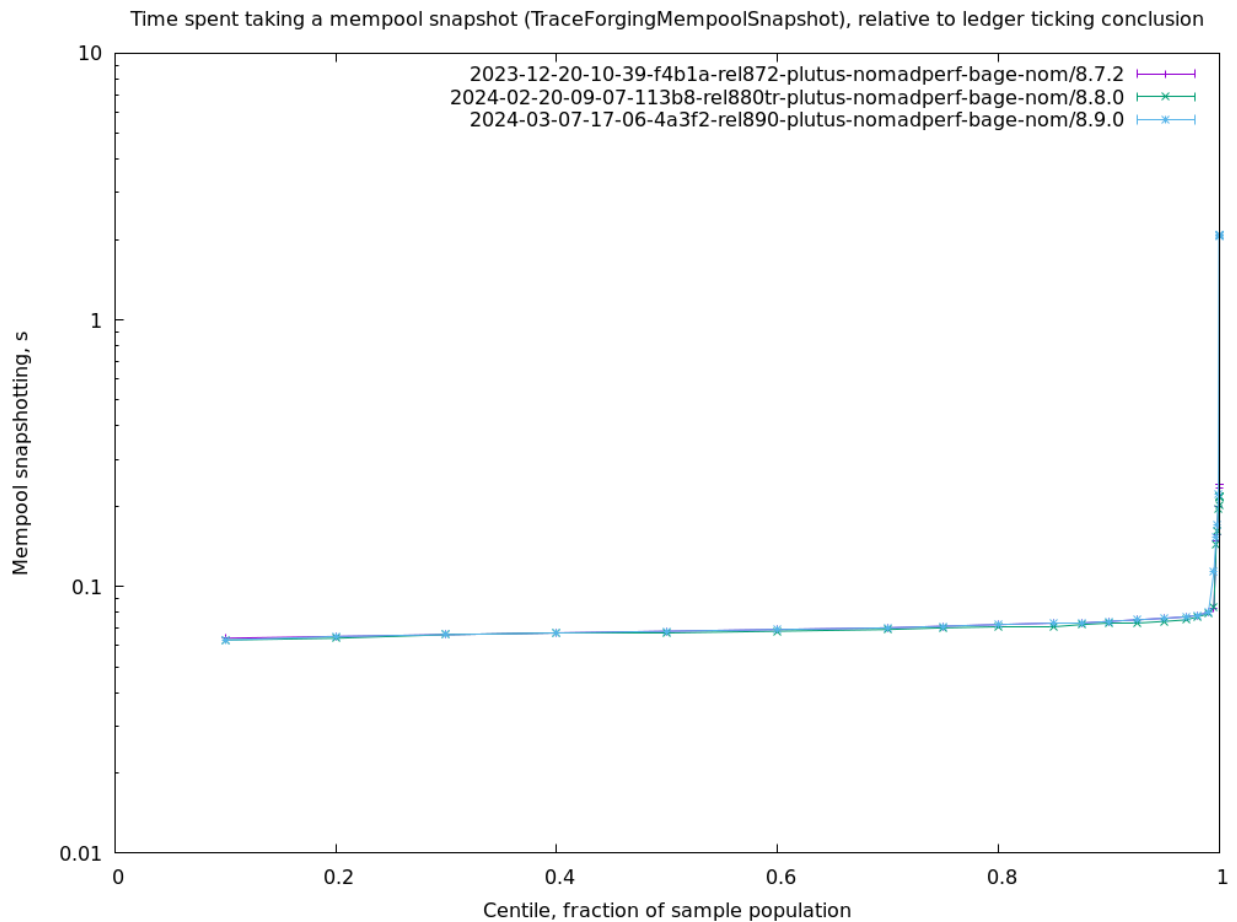
**Leadership check duration (cdfForgerLead)** Leadership check duration (TraceNodeIsNotLeader, TraceNodeIsLeader), relative to ledger view acquisition



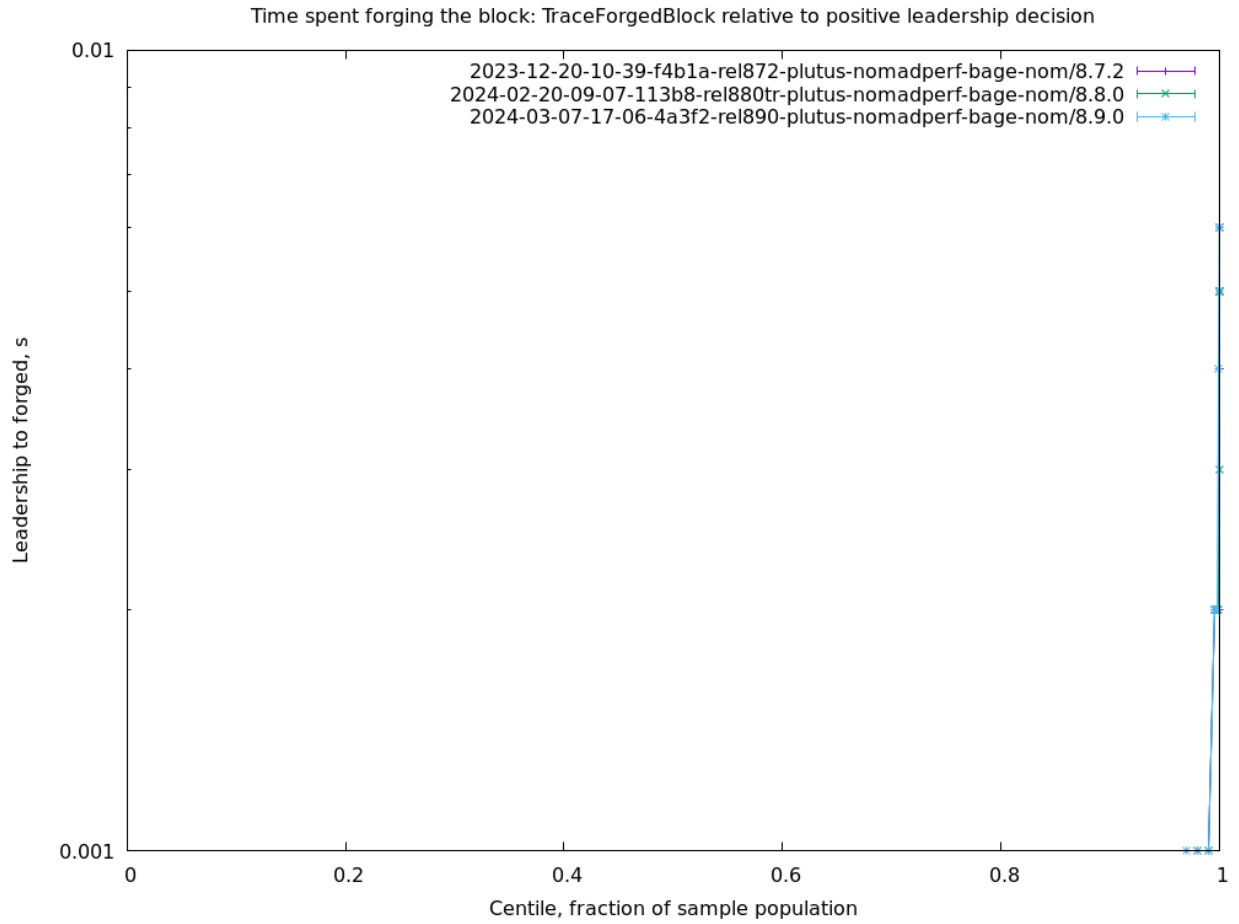
**Ledger ticking (cdfForgerTicked)** Time spent ticking the ledger state (TraceForgeTickedLedgerState), relative to leadership check completion



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

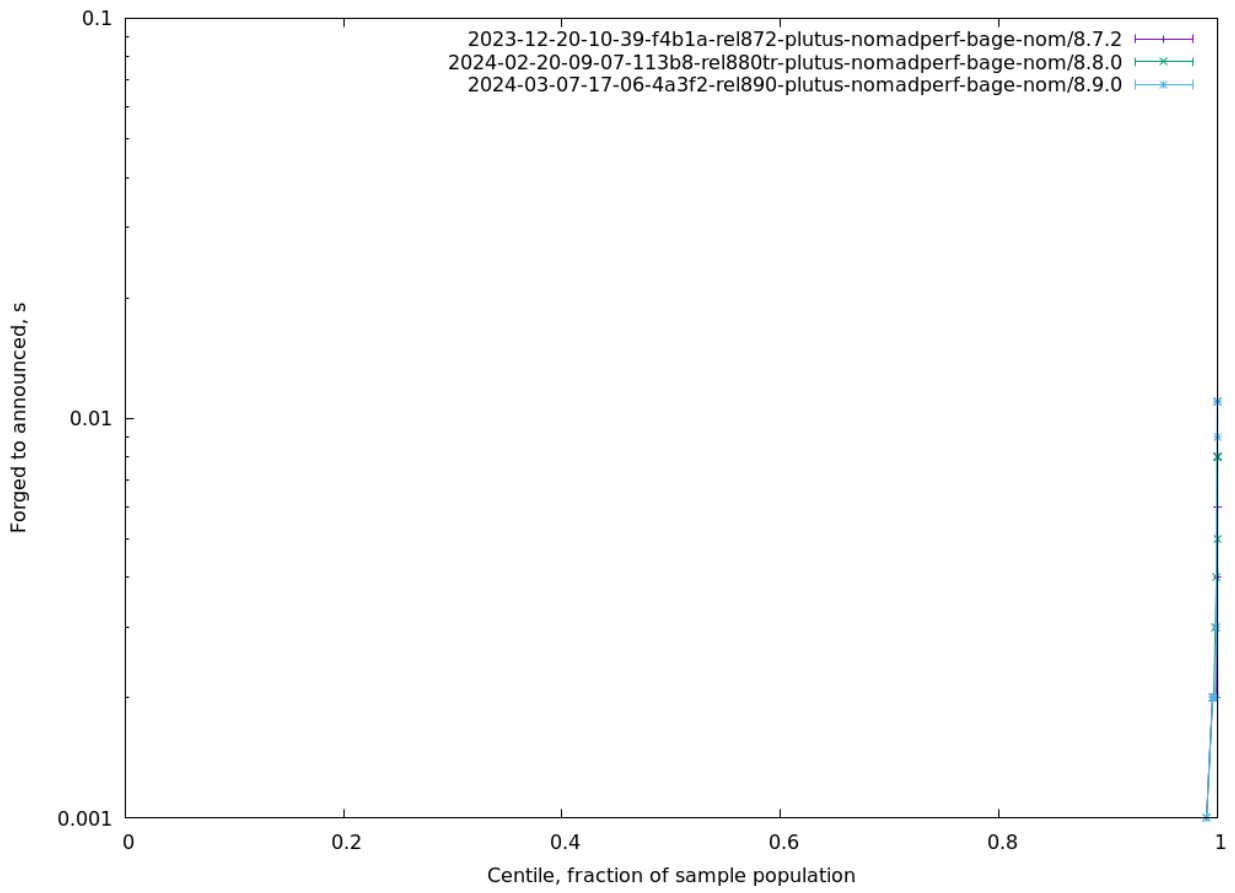


**Leadership to forged (cdfForgerForge)** Time spent forging the block: TraceForgedBlock relative to positive leadership decision

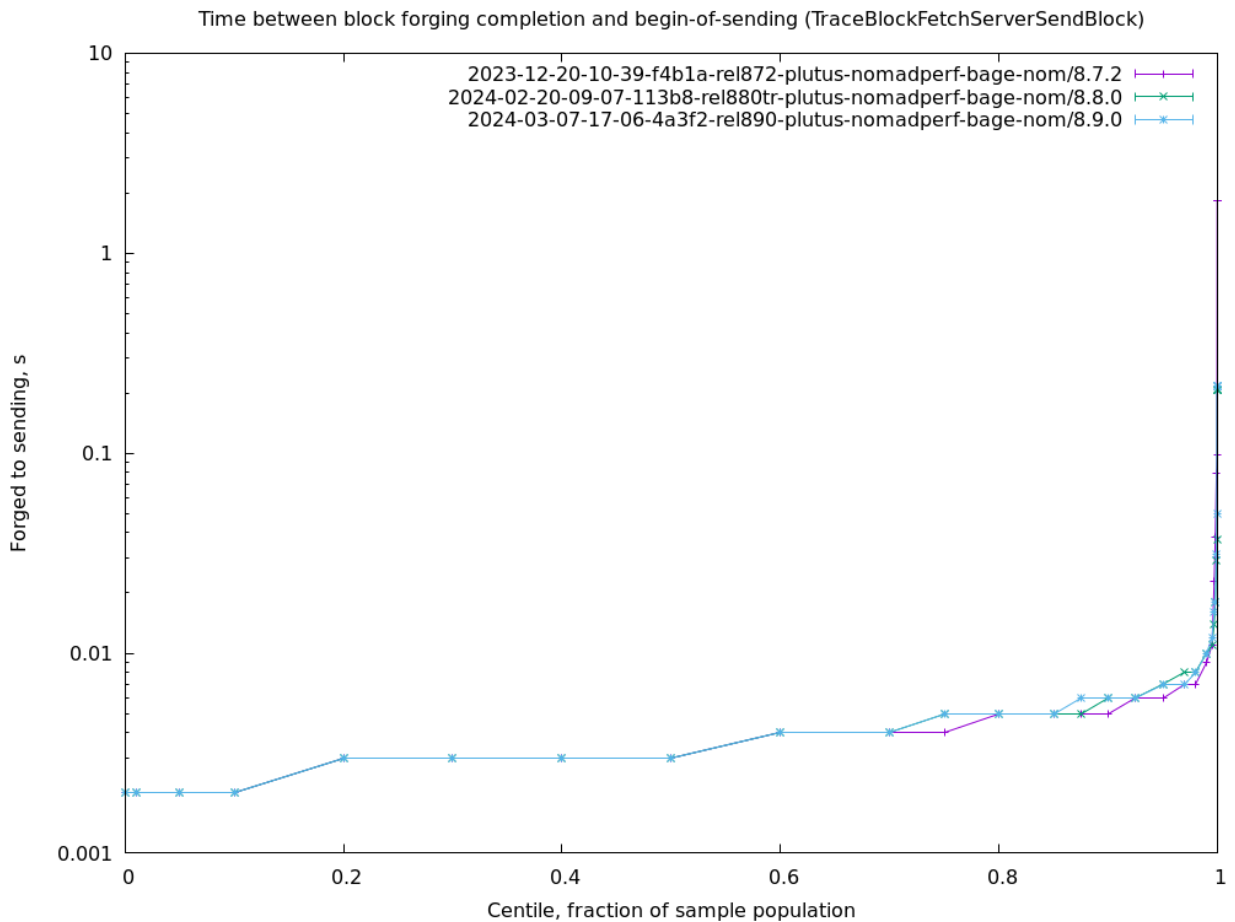


**Forged to announced (cdfForgerAnnounce)** Time between block forging completion and header announcement (ChainSyncServerEvent.TraceChainSyncServerRead.AddBlock)

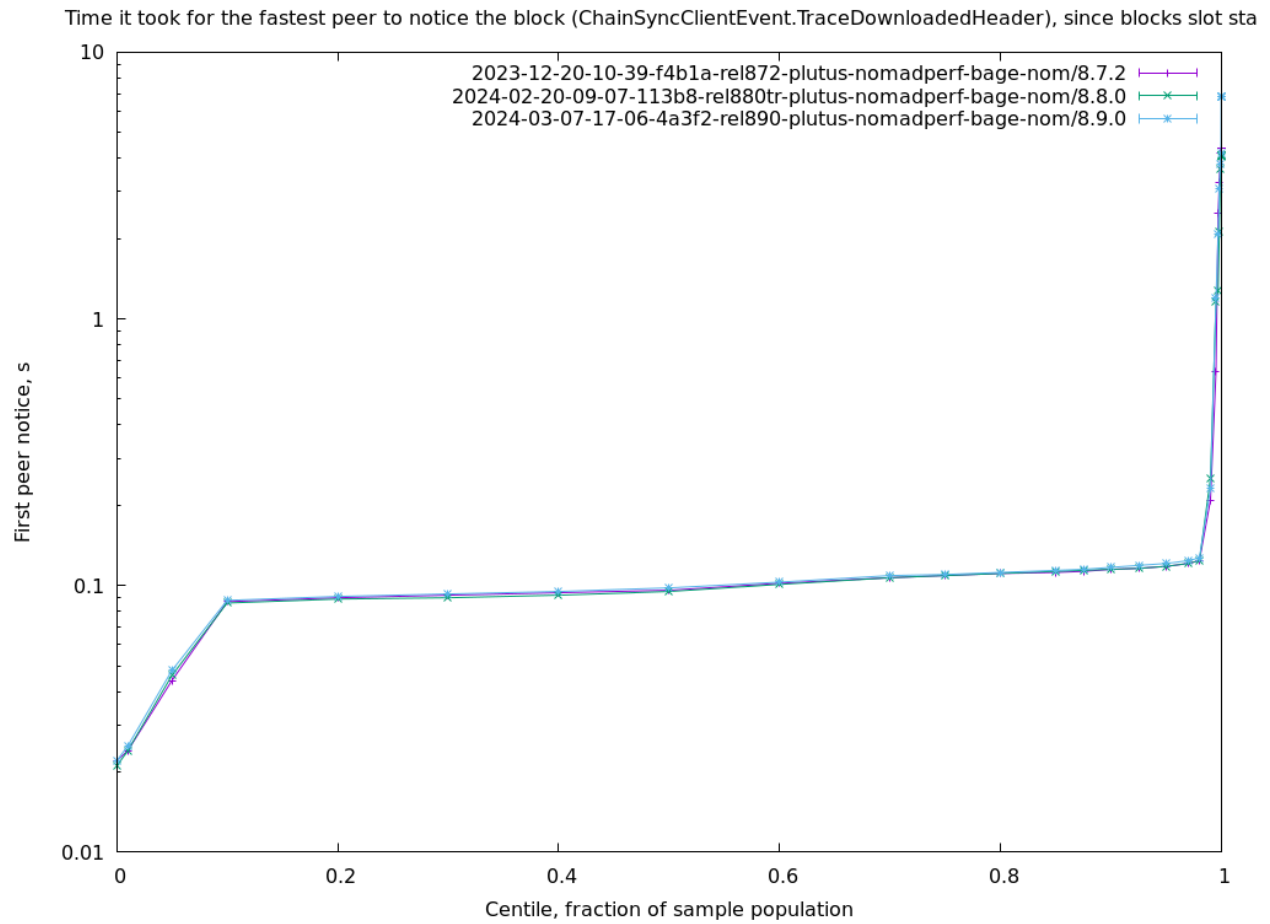
Time between block forging completion and header announcement (ChainSyncServerEvent.TraceChainSyncServerRead.AddF



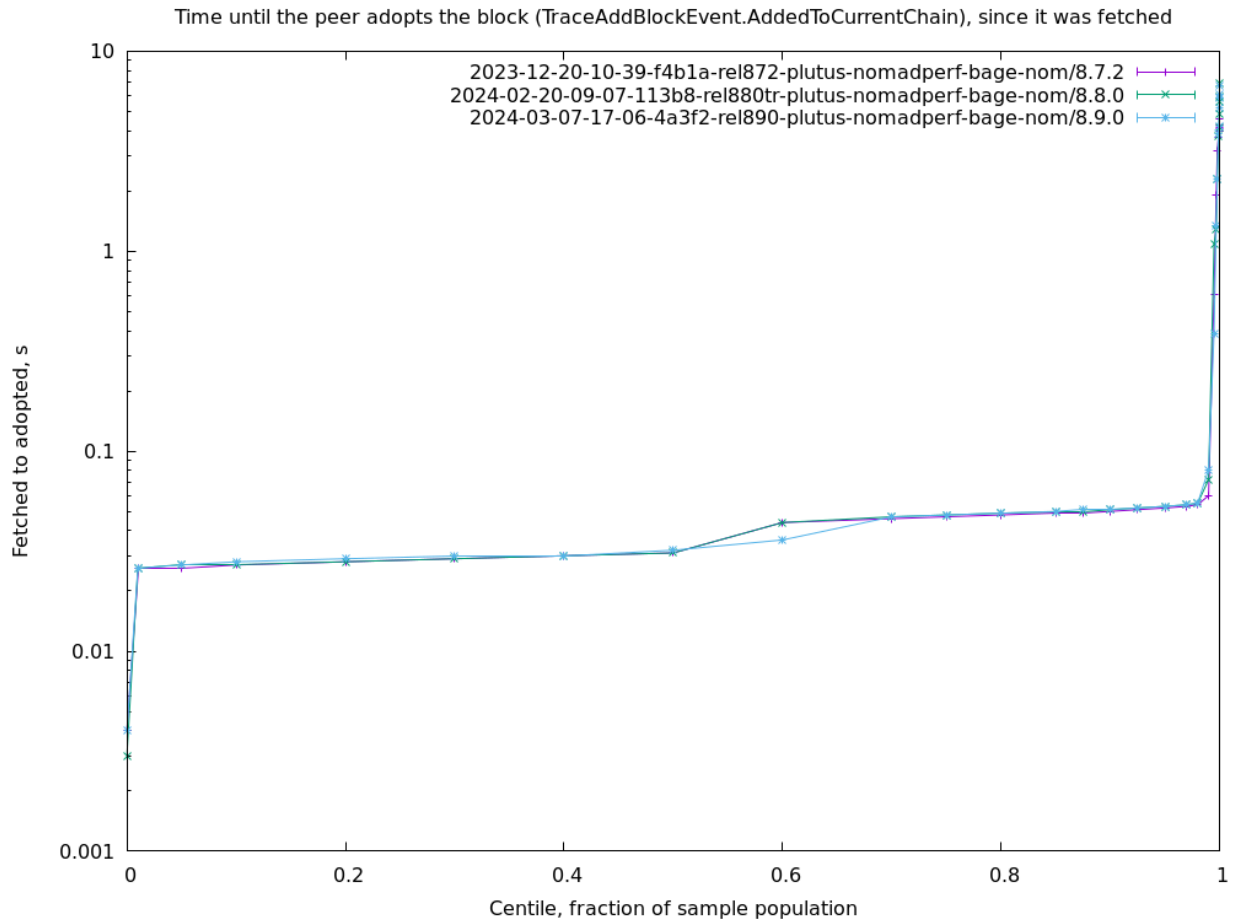
**Forged to sending (cdfForgerSend)** Time between block forging completion and begin-of-sending (TraceBlockFetchServerSendBlock)



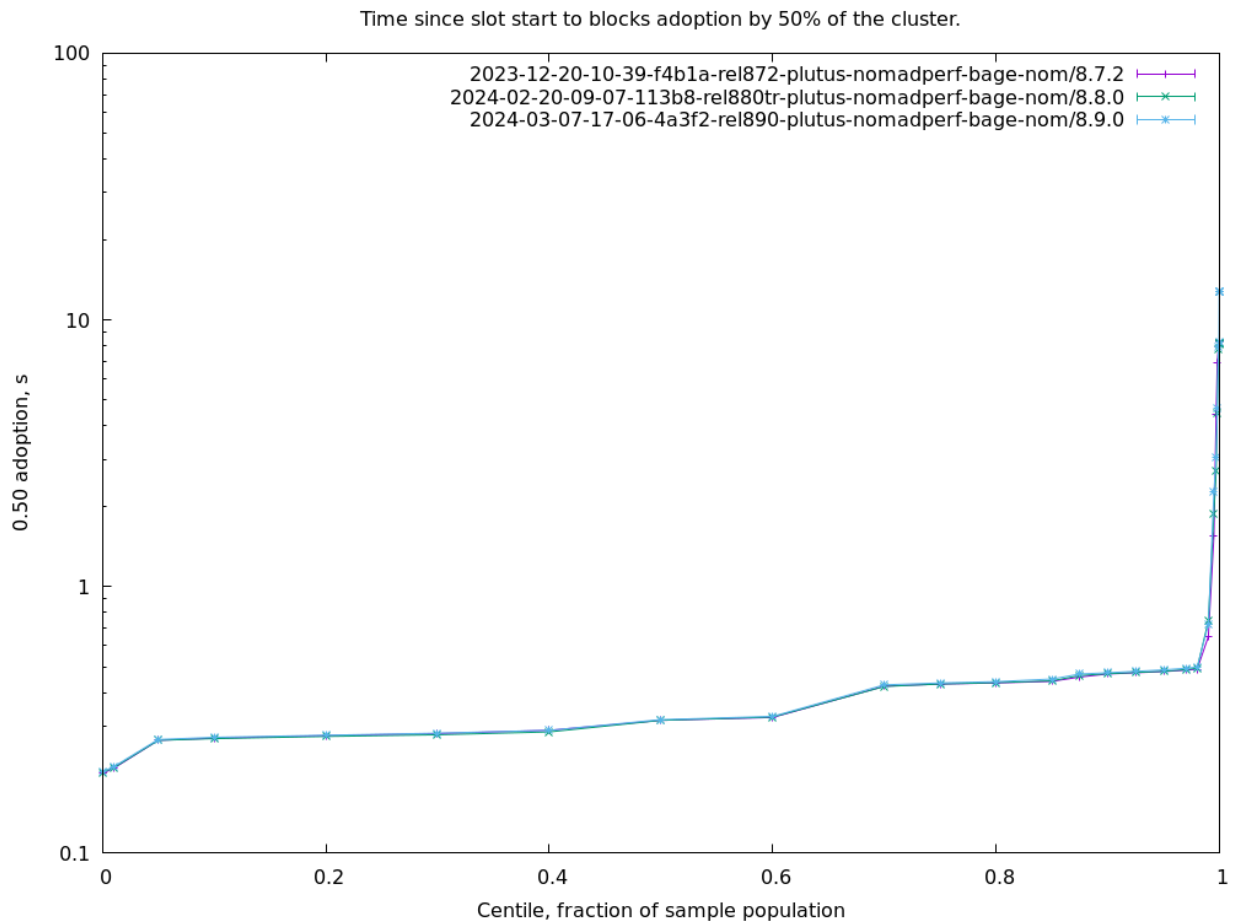
**First peer notice (cdfPeerNoticeFirst)** Time it took for the fastest peer to notice the block (ChainSyncClientEvent.TraceDownloadedHeader), since block's slot start



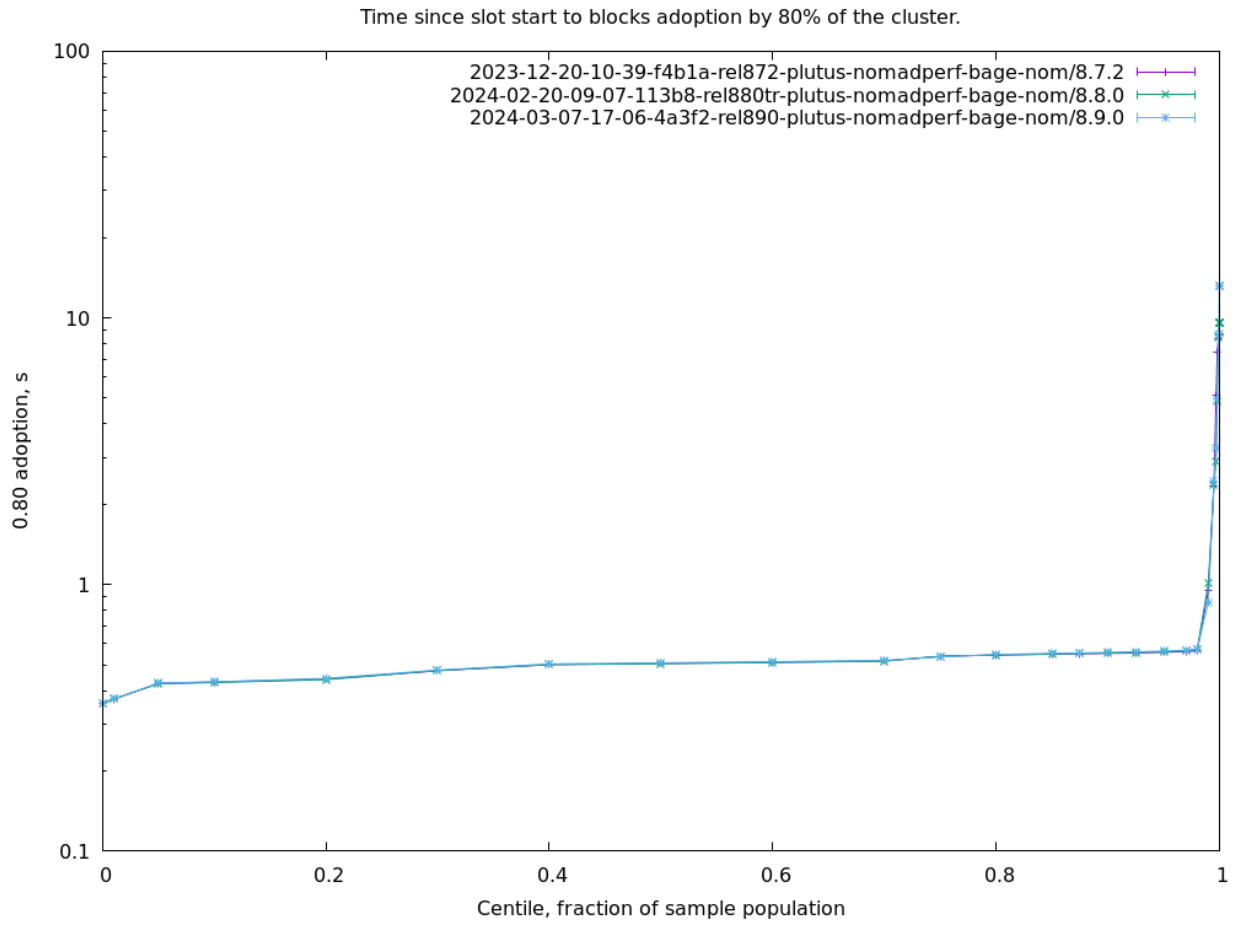
**Fetches to adopted (cdfPeerAdoption)** Time until the peer adopts the block (TraceAddBlockEvent.AddedToCurrentChain), since it was fetched



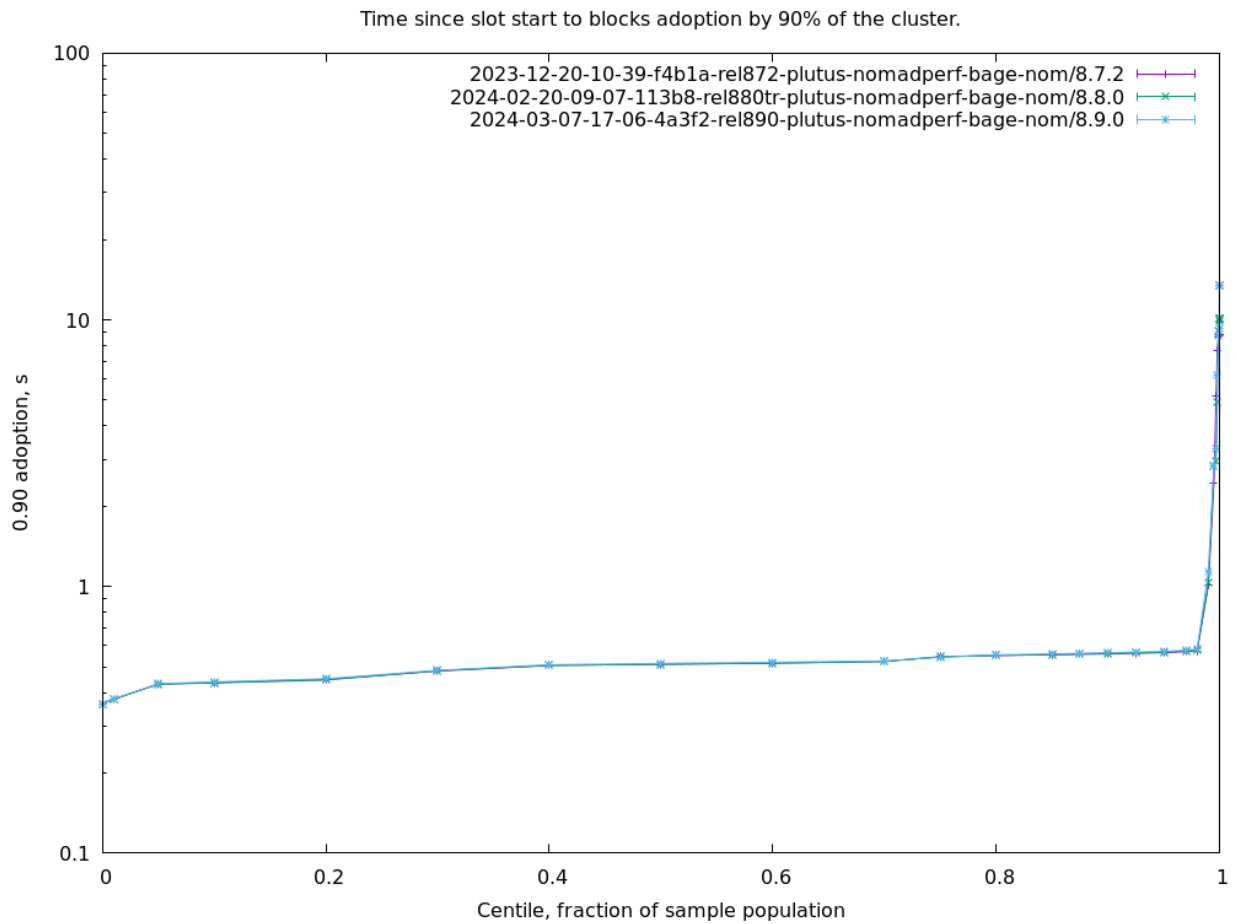
**0.50 adoption (cdf0.50)** Time since slot start to block's adoption by 50% of the cluster.



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



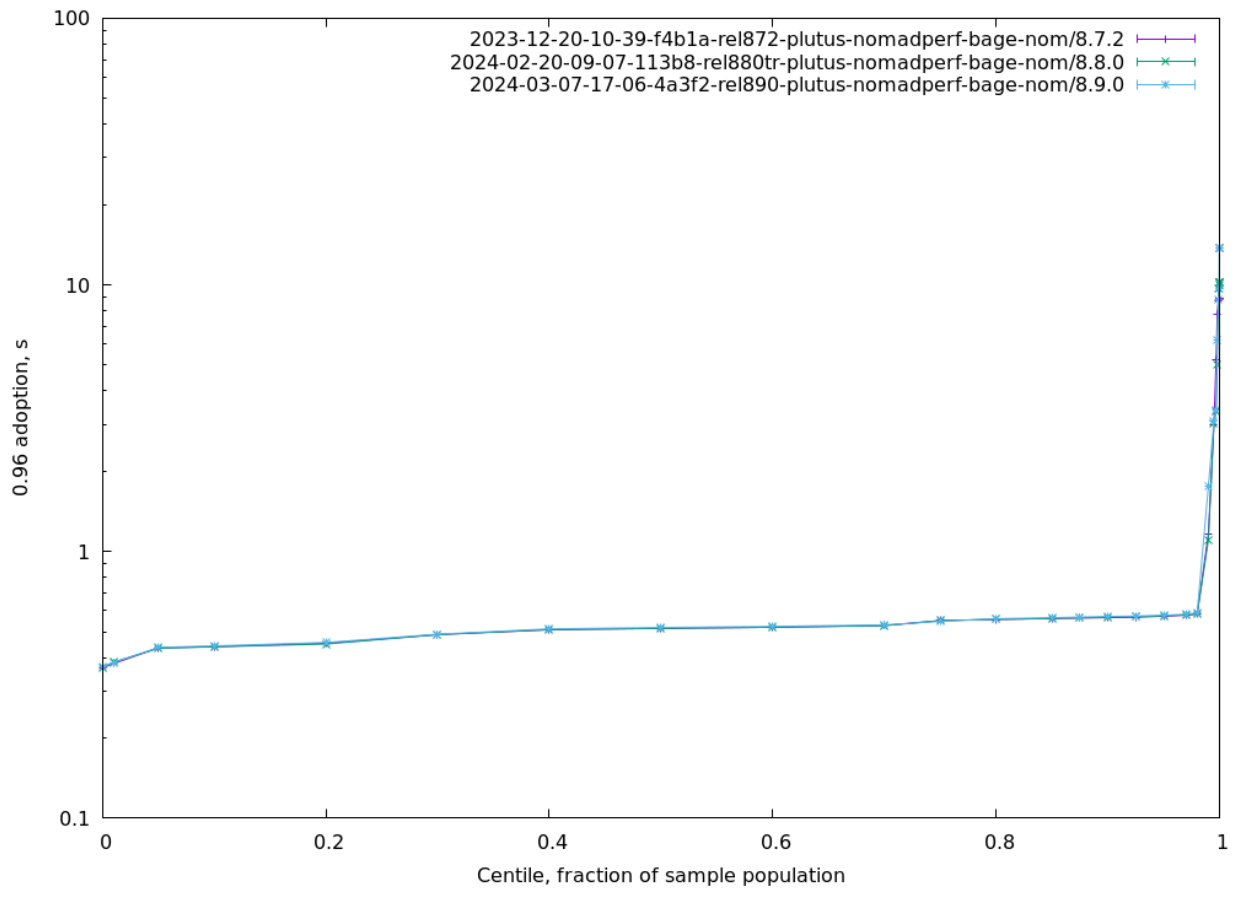
**0.90 adoption (cdf0.90)** Time since slot start to block's adoption by 90% of the cluster.



**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