Node 10.3 / GHC 9.6: tx budget memory limit x1 / x1.5 / x2 $$\rm Plutus\ countdown\ loop\ workload}$

Michael Karg, Cardano Performance team

2025-03-28

Contents

1	Manifest	2
2	Analysis	4
	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
I 3	Appendix A: charts Cluster performance charts	6 7
II	Appendix B: data dictionary	25
4	Block propagation metrics	26
5	Cluster performance metrics	28

Manifest

We compare loop-memx1.5 (Conway) and loop-memx2 (Conway) relative to 10.3-ghc965 (Conway), under Plutus countdown loop workload.

	10.3-ghc 965	loop-memx1.5	loop-memx2
Analysis date	2025-03-18	2025-03-19	2025-03-20
Cluster system start date	2025-03-17	2025-03-18	2025-03-19
Cluster system start time	11:25:04	17:18:06	loop-memx2 2025-03-20 2025-03-19 16:15:13 10.2.1 103ghc965 9.6.5 (10.3) 0.22.0.0 0.20.00 1.17.0.0 1.40.00 1.2.0 0.2.1.0 a37b7b4 minimal minimal minimal 9152d0e 68e015f conway 1000000 4000000 1000 100 1 0.85 61200 Loop 52 4 837146.80769 540300.25 71853.5 11.650 28095613 71856 56011 0.77 4 3153367
Identifier	10.2.1	10.2.1	10.2.1
Run batch	$103 { m ghc} 965$	$103 { m ghc} 965$	$103 \mathrm{ghc} 965$
GHC version	9.6.5	9.6.5	9.6.5
cardano-node version	(10.3)	(10.3)	(10.3)
ouroboros-consensus version	0.22.0.0	0.22.0.0	0.22.0.0
ouroboros-network version	0.20.0.0	0.20.0.0	0.20.0.0
cardano-ledger-core version	1.17.0.0	1.17.0.0	1.17.0.0
plutus-core version	1.40.0.0	1.40.0.0	1.40.0.0
cardano-crypto version	1.2.0	1.2.0	1.2.0
cardano-prelude version	0.2.1.0	0.2.1.0	0.2.1.0
cardano-node git	e969710	a37b7b4	a37b7b4
ouroboros-consensus git	minimal	minimal	minima
ouroboros-network git	minimal	minimal	minimal
cardano-ledger-core git	minimal	minimal	minimal
plutus-core git	minimal	minimal	minimal
cardano-crypto git	9152 d0 e	9152 d0 e	$9152 d0 \epsilon$
cardano-prelude git	68e015f	68e015f	68e015
Era	conway	conway	conway
Delegation map size	1000000	1000000	1000000
Stuffed UTxO size	4000000	4000000	4000000
DRep count	10000	10000	10000
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 objects emitted per host	848140.98076	839460.67307	837146.80769
Log objects analysed per host	544002.59615	541594.96153	540300.25
Host run time, s	71853.5	71850.0	71853.5
Host log line rate, Hz	11.803	11.683	11.650
Total log objects analysed	28288135	28162938	28095613
Run time, s	71858	71855	71856
Analysed run duration, s	56022	56017	56011
Run time efficiency	0.77	0.77	0.77
Node start spread, s	5.5873925	5.1672224	4.3153367
Node stop spread, s	2.7119011	3.0697526	1.6161739
Slots analysed	56020	56014	56010
Blocks analysed	2825	2696	2695
		0.40	0.55

Analysis

2.1 Resource Usage

	10.3-ghc 965	loop-memx1.5	Δ	$\Delta\%$	loop-memx2	Δ	$\Delta\%$
Forge loop starts, $\#$	0.99891	0.99892	0.000	0	0.99893	0.000	0
Process CPU usage, $\%$	5.5109	5.5295	0.019	0	5.5666	0.056	1
RTS GC CPU usage, $\%$	0.2659	0.26647	0.001	0	0.2669	0.001	0
RTS Mutator CPU usage, $\%$	5.2374	5.2592	0.022	0	5.2941	0.057	1
Major GCs, $\#$	0.00085	0.00087	0.000	0	0.00085	0.000	0
Minor GCs, $\#$	1.33	1.396	0.066	5	1.4467	0.117	9
Kernel RSS, MB	8120.4	8005.8	-114.600	-1	8025.5	-94.900	-1
RTS heap size, MB	8064.0	7949.6	-114.400	-1	7969.2	-94.800	-1
RTS live GC dateset, MB	3439.0	3422.9	-16.100	0	3474.2	35.200	1
RTS alloc rate, MB/s	40.74	42.601	1.861	5	44.004	3.264	8
Filesystem reads, KB/s	3e-05	0.0	-0.000	0	0.0	-0.000	0
Filesystem writes, KB/s	228.21	228.37	0.160	0	227.12	-1.090	0
CPU 85% spans, slots	11.466	10.849	-0.617	-5	11.307	-0.159	-1
Sample count	(291>)	(291>)			(291>)		

2.2 Anomaly control

	10.3-ghc 965	loop-memx1.5	Δ	$\Delta\%$	loop-memx2	Δ	$\Delta\%$
Blocks per host, blocks	72.884	69.75	-3.134	-4	70.346	-2.538	-3
Filtered to chained block ratio, $/$	0.76297	0.76165	-0.001	0	0.75553	-0.007	-1
Chained to forged block ratio, $/$	0.97722	0.97546	-0.002	0	0.97658	-0.001	0
Height & slot battles, blocks	0.0007	0.00037	-0.000	0	0.00074	0.000	0
Block size, B	2996.0	3011.2	15.200	1	3012.0	16.000	1
Sample count	(52)	(52)			(52)		

2.3 Forging

	10.3-ghc 965	loop-memx1.5	Δ	$\Delta\%$	loop-memx2	Δ	$\Delta\%$
Started forge loop iteration, s	0.00105	0.00143	0.000	0	0.00128	0.000	0
Acquired block context, s	0.02334	0.02331	-0.000	0	0.02227	-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	2e-05	0.000	0
Leadership check duration, s	0.0004	0.00038	-0.000	0	0.00038	-0.000	0
Ledger ticking, s	0.0235	0.02306	-0.000	0	0.02461	0.001	4
Mempool snapshotting, s	0.00178	0.00172	-0.000	0	0.00172	-0.000	0
Leadership to forged, s	0.00016	0.00015	-0.000	0	0.00015	-0.000	0
Forged to announced, s	0.00063	0.00063	0.000	0	0.00064	0.000	0
Forged to sending, s	0.0052	0.00535	0.000	0	0.00529	0.000	0
Forged to self-adopted, s	0.04189	0.05232	0.010	24	0.06255	0.021	50
Slot start to announced, s	0.05098	0.0508	-0.000	0	0.05117	0.000	0
Sample count	(2825)	(2696)			(2695)		

2.4 Individual peer propagation

	10.3-ghc 965	loop-memx1.5	Δ	$\Delta\%$	loop-memx2	Δ	$\Delta\%$
First peer notice, s	0.0528	0.05266	-0.000	0	0.05301	0.000	0
First peer fetch, s	0.05773	0.05776	0.000	0	0.05795	0.000	0
Notice to fetch request, s	0.00115	0.00126	0.000	0	0.00121	0.000	0
Fetch duration, s	0.12335	0.12236	-0.001	-1	0.12235	-0.001	-1
Fetched to announced, s	0.00073	0.00096	0.000	0	0.00072	-0.000	0
Fetched to sending, s	0.04266	0.04434	0.002	5	0.04293	0.000	0
Fetched to adopted, s	0.04504	0.05557	0.011	24	0.06441	0.019	42
Sample count	(2825)	(2696)			(2695)		

2.5 End-to-end propagation

	10.3-ghc 965	loop-memx1.5	Δ	$\Delta\%$	loop-memx2	Δ	$\Delta\%$
0.50 adoption, s	0.31168	0.32046	0.009	3	0.32826	0.017	5
0.80 adoption, s	0.47047	0.48323	0.013	3	0.48845	0.018	4
0.90 adoption, s	0.48044	0.49599	0.016	3	0.50018	0.020	4
0.92 adoption, s	0.48305	0.49885	0.016	3	0.50369	0.021	4
0.94 adoption, s	0.4865	0.50131	0.015	3	0.50642	0.020	4
0.96 adoption, s	0.4901	0.50449	0.014	3	0.51082	0.021	4
0.98 adoption, s	0.4947	0.51346	0.019	4	0.52225	0.028	6
1.00 adoption, s	0.5192	0.55024	0.031	6	0.55404	0.035	7
Sample count	(2825)	(2696)			(2695)		

Part I

Appendix A: charts

Cluster performance charts

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



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





Centile, fraction of sample population

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



RTS-reported allocation rate, MB/sec

Major GCs (GcsMajor) Major garbage collection RTS events

Major garbage collection RTS events



Centile, fraction of sample population

Minor GCs (GcsMinor) Minor garbage collection RTS events

Minor garbage collection RTS events



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

RTS-reported heap size, MB



centile, fraction of sample populat

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

RTS-reported GC live data size, MB



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





Centile, fraction of sample population

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

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



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

Block context acquisition delay, s

Block context acquired (TraceBlockContext), relative to forge loop beginning

Centile, fraction of sample population

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

14

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

Block/slot ratio, for the last k slots

Interblock gap (cdfBlockGap) Time between blocks

Time between blocks

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

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

Centile, fraction of sample population

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

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

Centile, fraction of sample population

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.AddBloc

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

20

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

Time it took for the fastest peer to notice the block (ChainSyncClientEvent.TraceDownloadedHeader), since blocks slot start

Fetched 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.

Time since slot start to blocks adoption by 80% of the cluster.

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

Centile, fraction of sample population

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

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 (ChainSync-ServerEvent.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
- Fetched to adopted (cdfPeerAdoption) Time until the peer adopts the block (TraceAddBlockEvent.AddedToCurrentChain). since it was fetched
- Fetched to announced (cdfPeerAnnounce) Time it took a peer to announce the block (ChainSyncServerEvent.TraceChainSy 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.CompletedBlockFets) 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
- Fetched to sending (cdfPeerSend) Time until the peer started sending the block (BlockFetchServer.SendBlock), since it was fetched

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 dateset (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