

Managing state growth & corresponding issues

As the Orbit ecosystem grows, more and more teams are choosing to build on the Arbitrum tech stack. Orbit offers a feature-rich and scalable rollup stack that allows teams to focus on their ecosystem growth and build great products. As a result, many Orbit chains are seeing usage and throughput increase rapidly, often with sustained transaction load at the chain's throughput limit for months on end.

This page aims to educate Orbit chain operators and owners on safely operating a high throughput chain. Amongst all the factors, the primary consideration is state growth rate and state size. We'll discuss how increased state size affects the performance of different components in the Orbit stack and how certain metrics can be used to indicate a need to upgrade various infrastructure components.

Understanding state size and state growth rate

When we say 'state size', we mean the total amount of data recorded on the blockchain; state size is a critical metric for node performance as a larger state creates higher infrastructure requirements on nodes for storage and searching through existing states.

The state growth rate is simply the rate at which state size increases. A high state growth rate creates higher requirements for nodes to process state transitions and perform operations needed to keep up with the tip of the chain.

The critical Nitro stack parameter affecting state growth and state growth rate is the gas speed limit. Offchain Labs has set the gas speed limit to a default (7,000,000 gas/s), ensuring a safe, sustained operating limit for Orbit chains. You can read more about the nuances of the gas speed limit [here](#).

The default gas speed limit is designed to ensure Orbit chains operate performantly and sustainably.

Behaviour at ultra-high throughput

At high state growth rates, especially in cases where a chain is pushing past prescribed limits, an Orbit chain may display certain behaviors that either indicate or result from the chain load being higher than its infrastructure can support. The following is a list of such behaviors.

1. Read Operation Bottleneck at High Disk Latency

As the number of read requests on a chain grows, the impact of disk latency on performance becomes more pronounced. The performance impact can be considered the total amount of read requests made as a multiple of the disk latency. High read request volumes may necessitate switching to using low-latency local NVMe drives.

2. Increased single-core CPU and RAM Utilization

Observing high utilization on single-core CPU and RAM indicates that you may require more performant hardware. As this trend continues, hardware investments become prohibitively expensive for the ecosystem or require increasingly custom solutions, which decreases accessibility for node runners.

3. Increased Total State Database Size

The accelerated state database growth rate, on the order of multiple terabytes of data per month, indicates that your chain may require increasing drive sizes. Played out over time, this may force node runners on the chain to adopt prohibitively expensive or hard-to-procure drives (e.g., those notes available on major cloud providers).

4. Increased Disk Write Operations per Second

The number of write operations per second directly correlates to state size growth. As the state growth rate increases, ecosystem nodes that aren't properly resourced may fall out of sync with the chain.

5. Sync from Genesis Time

As state size increases, the time a new node needs to catch up to the chain also increases. A large state size and state growth rate can result in new nodes catching up to the chain in the worst case.

Summary of symptoms, mitigations, risks

The general trend with any issue in the table below is as follows:

- The simple resolutions involve moving to more expensive infrastructure.
- When simple resolutions are exhausted, infrastructure becomes both expensive and bespoke (options that are available)

cloud providers do not support)

- The long-term risk (and point of no return) is when infrastructure requirements are too expensive or too inaccessible for node runners.

Behaviour Risk Mitigations & Considerations Performance degradation due to storage reads at high disk latency An increase in read operations causes nodes to spend more accessing disk state. As the number of read operations increases, these delays can degrade chain performance. Upgrade drives to local NVMe (PCIe Gen4/Gen5, not configured with RAID) with higher speeds. In the short term, NVMe usage will greatly increase the cost of node runners. In the extreme, you may run out of usable drive specifications on available infrastructure vendors. Growing or constant sequencer backlog (using `arb_sequencer_backlog`) over a sustained period. Seeing a growing or persistent backlog implies that nodes cannot keep up with the transaction load accepted by the sequencer. Large state database size and high growth rate of the state database A large state database size will require that nodes run more expensive disks. This reduces the economic feasibility for node runners. In extreme cases, the required disk size may be unsupported by accessible cloud service providers. The primary resolution is to upgrade the disk size requirement for nodes on your chain. High utilization of single-core CPU and RAM As with the cases above, this symptom implies a need to upgrade hardware. The main risk is the economic feasibility and long-term accessibility of new hardware options. The only resolution is to upgrade your node's CPU and RAM. [Edit this page](#) Last updated on Nov 4, 2024 [Previous Quickstart](#) [Next Manage gas speed limit](#)