Chainstack

Chainstack Dedicated Nodes performance benchmark

A cross-protocol and cloud environment analysis

Table of contents

Executive summary

This executive summary outlines the findings from a comprehensive performance assessment of Chainstack dedicated blockchain nodes across multiple protocols, including Ethereum, Polygon, BNB Smart Chain, and Solana. The study employed a methodical approach using specialized load test profiles designed to mimic real-world user interactions and measure the robustness, efficiency, and maximum Requests per Second (max RPS) of nodes in various configurations and cloud environments.

Key findings

Ethereum

- **Chainstack Cloud Latitude environment:**
  - Positive points
    - Superior transaction handling capacity with a maximum RPS of 1670.
    - Efficient scaling capabilities with peak resource consumption at 20 GB RAM and 6.92 CPU cores.
    - Robust response times: 320 ms for the 95th percentile, 800 ms for the 99th percentile.
- **Virtuozzo (VZO) environment:**
  - Positive points
    - Robust performance with a maximum RPS of 740.
    - Reasonable resource utilization: 9.86 GB RAM and 6.61 CPU cores.
  - Improvement points
    - Lower maximum RPS compared to Chainstack Cloud Latitude.
    - Higher response time for the 99th percentile: 820 ms.
- **Amazon Web Services (AWS) environment:**
  - Positive points
    - Best average response time among environments: 180 ms for the 95th percentile, 590 ms for the 99th percentile.
    - Moderate resource usage: 15.9 GB RAM and 5.4 CPU cores.
  - Improvement points
    - Lower maximum RPS compared to Chainstack Cloud Latitude and VZO.

Polygon

- **Chainstack Cloud Latitude environment:**
  - Positive points
    - Highest maximum RPS of 800.
    - Optimal response times: 160 ms for the 95th percentile, 960 ms for the 99th percentile.
    - Moderate resource utilization: 11.7 CPU cores and 37.6 GB memory.
- **VZO environment:**
  - Positive points
    - Consistent performance with response times of 270 ms for the 95th percentile, 1400 ms for the 99th percentile.
  - Improvement points
    - Higher resource usage: 12.7 CPU cores and 89.9 GB memory.
    - Slower response times for the 99th percentile: 1400 ms.
- **AWS environment:**
  - Positive points
    - Comparable upper limit response times with VZO.
  - Improvement points
    - Higher resource consumption: 14.1 CPU cores and 39.1 GB memory.
    - Slower response times: 310 ms for the 95th percentile, 1400 ms for the 99th percentile.

BNB Smart Chain

- **Chainstack Cloud Latitude environment:**
  - Positive points
    - Highest maximum RPS of 910 among environments.
    - Efficient peak load handling: 230 ms for the 99th percentile.
    - Comparatively low resource consumption: 21.4 GB memory and 13.5 CPU cores.
- **VZO environment:**
  - Positive points
    - Reliable performance with consistent response times.
  - Improvement points
    - Lower maximum RPS compared to Chainstack Cloud Latitude.
    - Higher memory consumption: 82.3 GB.
- **AWS environment:**
  - Improvement points
    - Higher resource usage: 88.4 GB memory and 12.4 CPU cores.
    - Slower response times: 320 ms for the 95th percentile, 860 ms for the 99th percentile.

Solana

- **Chainstack Cloud NYC environment without GPA:**
  - Positive points
    - Outstanding transaction processing with an average RPS of 5906.84.
    - Enhanced performance and efficiency: median response times of 66 ms, 99th percentile response times of 2000 ms.
    - Efficient resource management: average CPU at 40.4, memory at 860 GB.
- **Chainstack Cloud AMS environment with GPA:**
  - Positive points
    - Better controlled response times: median at 69 ms, 99th percentile at 3300 ms.
    - Efficient CPU utilization: average at 31, peaking at 42.
- **Chainstack Cloud NYC environment with GPA:**
  - Improvement points
    - High variance in response times under heavy load: 99th percentile peaking at 5000 ms.
    - Significant resource usage: average CPU at 41.6, peaking at 42, and memory usage up to 880 GB.
- **Chainstack Cloud AMS environment without GPA:**
  - Improvement points
    - Lower average RPS: 1389.
    - Higher memory usage: averaging at 1040 GB, maxing at 1070 GB.
- **General observations:**
  - Positive points
    - Load profiles without GPA achieve higher RPS.
    - NYC environment consistently outperforms AMS in RPS and 99th percentile response times.
    - AMS environment shows higher memory usage with lower CPU utilization, suggesting different resource management strategies.
  - Improvement points
    - GPA load profiles show lower RPS compared to non-GPA profiles.
    - AMS environment shows less efficient transaction processing compared to NYC, with higher memory and lower CPU usage.

Methodology

The performance assessments employed a methodical approach to evaluate the efficiency and robustness of dedicated nodes across various protocols. These evaluations involved simulating user activities from diverse locations and different cloud providers, aiming to rigorously test the nodes' capacity to handle substantial traffic and workloads.

Key metrics such as response times, failure rates, and maximum Requests Per Second (max RPS) were meticulously analyzed to derive insights into the node's ability to manage and process high volumes of requests efficiently.

Ethereum

For evaluations in this category, the `EthereumFullTop10` specialized load test profile was used. This profile was crafted using genuine user interaction data to ensure an accurate representation of real-world usage. It includes a variety of JSON-RPC method calls, each weighted according to its real-life usage frequency.

The methodology for the Ethereum test involved starting with a relatively low number of virtual users, such as 200, and incrementally increasing this number until the node began to regularly return errors. Results were taken from the highest number of virtual users the node could handle without regularly returning errors.

For example, in the VZO environment, the load consisted of 800 users, whereas Chainstack Cloud Latitude handled 1600 users, and AWS managed 650 users. This variable loading explains differences in response times and max RPS across environments, with Chainstack Cloud Latitude sustaining higher loads while maintaining competitive response times and max RPS.

During the testing period, the nodes were scrutinized in their full operational mode, configured with the following resource limits:

Virtuozzo and Chainstack Cloud Latitude

- **CPU:** 8 cores
- **RAM:** 28GB

AWS

- **CPU:** 7 cores
- **RAM:** 22GB

Polygon

In these evaluations, we also utilized the `EthereumFullTop10` profile. The nodes underwent testing under full operational conditions, set with the following resource allocation:

AWS, Chainstack Cloud Latitude, and Virtuozzo

- **CPU:** 15 cores
- **RAM:** 117Gb

BNB Smart Chain

For these assessments, we employed the `EthereumFullTop10` profile, as well. The nodes were tested in a fully operational state, with specific resource allocations set as follows:

AWS, Chainstack Cloud Latitude, and Virtuozzo

- **CPU:** 16 cores
- **RAM:** 121Gb

Solana

The **`SolanaGeneral`** specialized load test profile was employed for Solana evaluations, utilizing the **`getProgramAccounts`** (GPA) method tailored to specific Solana programs. The GPA method, essential for retrieving lists of accounts managed by a particular program, was used with filters designed to interact with the **`Stake11111111111111111111111111111111111111`** program with the following code:

```
[
    "Stake11111111111111111111111111111111111111",
        {
        "commitment": "finalized",
        "encoding": "jsonParsed",
        "filters":
            [
                {
                "memcmp":
                    {
                    "bytes": "2K9XJAj3VtojUhyKdXVfGnueSvnyFNfSACkn1CwgBees",
                    "offset": 12
                    }
                }
            ]
        }
]
```

This setup employed a `memcmp` filter to search for accounts with specific data bytes at a given offset, facilitating targeted analysis of node performance under complex query loads. The `memcmp` filter, short for "memory comparison" filter, enables data comparison in any field stored within an account. Specifically, it allows accounts filtering to find those that match a given set of bytes at a specified position.

Testing configurations varied:

- Some tests were conducted **without GPA** to focus on general performance metrics of the Solana network, establishing baseline data for node performance.
- Other tests incorporated the **Staking GPA**, using the `getProgramAccounts` method to analyze how Solana nodes manage memory and processing power during more complex and resource-intensive operations.

power during more complex and resource-intensive operations.

For the Solana tests, the same number of users was used for all test scenarios, but the "waiting time" between requests for each user was removed. This adjustment meant that each virtual user sent as many requests as possible consecutively and as quickly as possible, waiting only for the response of the previous request before sending another. Consequently, response times directly influenced max RPS, with lower response times resulting in higher max RPS.

The resource limits set for these tests were as follows:

Chainstack Cloud AMS1 environment

- **CPU:** 42 cores
- **RAM:** 1100GB

Chainstack Cloud NYC1 environment

- **CPU:** 42 cores
- **RAM:** 1500GB

Profiles

Ethereum Full Top10

The JSON-RPC method calls utilized in the `EthereumFullTop10` profile, along with their respective weights, are as follows:

1. `eth_call`: Conducts a read-only smart contract method invocation. Parameters for this call are predefined. It is the most frequent call with a weight of 51.5%.
2. `eth_getTransactionReceipt`: Retrieves a transaction's receipt using its hash. Parameters are dynamically generated. It has a weight of 17.8%.
3. `eth_blockNumber`: Fetches the current block number, carrying a weight of 7.4%.
4. `eth_getTransactionByHash`: Obtains a transaction using its hash. This method is weighted at 5.5%.
5. `eth_getBlockByNumber`: Retrieves a block using its number, with parameters dynamically generated, holding a weight of 4.9%.
6. `eth_getBalance`: Returns the account balance for a specified address, weighted at 4.9%.
7. `eth_chainId`: Provides the current chain ID, with a weight of 3.7%.
8. `eth_getLogs`: Fetches the logs based on a query, weighted at 3.1%.
9. `web3_clientVersion` and `eth_getBlockByHash`: Each has a lower frequency, with a weight of 0.6%.

Solana General profile

The JSON-RPC method calls in the `SolanaGeneral` profile, along with their corresponding weights for the overall ratio, where the Staking GPA has been used, are specified as follows:

1. `getAccountInfo`: This method is used for retrieving detailed information about a specific account. Given its extensive usage in account management and transaction verification, it has the highest weight of 58.5% in the total ratio.
2. `getTransaction`: Retrieves detailed information of a transaction by its identifier. Essential for tracking and auditing transactions, it is weighted at 7.6%.

3. **`getTokenAccountsByOwner`**: Finds all accounts owned by a particular wallet address and filtered by token type. This method is critical for portfolio management and transaction analysis, with a weight of 8.8%.
4. **`getMultipleAccounts`**: Obtains information for multiple accounts simultaneously. This call facilitates efficient data retrieval, especially in client applications, and is weighted at 8.8%.
5. **`getBalance`**: Returns the balance of an account at a given address. This fundamental call for account management carries a weight of 4.4%.
6. **`getLatestBlockhash`**: Provides the most recent block hash on the blockchain. Necessary for recent transaction validations and other time-sensitive operations, it has a weight of 4.4%.
7. **`getSignaturesForAddress`**: Retrieves the list of transaction signatures involving a specific account. This method, key for auditing and historical data, is weighted at 4.4%.
8. **`getBlockHeight`**: Fetches the current block height of the blockchain. This is crucial for understanding the current state and progression of the chain, weighted at 0.9%.
9. **`getProgramAccounts`**: Returns a list of accounts managed by a specific program. This is useful for developers and service providers, carrying a weight of 0.3%.
10. **`getRecentBlockhash`**: Provides the most recent block hash used for making recent transaction submissions. It has a minor weight of 0.2%.
11. **`getSignatureStatuses`**: Checks the processing status of signatures for transactions. This is important for confirming transaction statuses, weighted at 0.2%.
12. **`getSlot`**: Retrieves the current slot number of the blockchain. This method, used in various synchronization and diagnostic tasks, is weighted at 1.2%.

Other methods like **`getBlocks`**, **`getConfirmedSignaturesForAddress2`**, and **`getEpochInfo`** are also part of the profile, each contributing a minor weight of 0.1% due to their specific use cases in blockchain analysis and diagnostics.

In tests where no GPA has been selected, the profile excludes the **`getProgramAccounts`** method.

Results

The performance evaluation of our nodes was meticulously carried out across various environments to ascertain their resilience, scalability, and efficiency under load. The results explore performance metrics like max requests per second (RPS), average response times (for 95th and 99th percentile of requests), and the peak resource utilization (CPU and Memory).

Ethereum

The assessments were conducted across three distinct environments: Virtuozzo (VZO), Chainstack Cloud Latitude, and Amazon Web Services (AWS), each employing the Ethereum Full Top10 profile.

VZO environment

The Ethereum node hosted in the VZO environment demonstrated a maximum RPS of 740, indicating its capability to handle a considerable number of transactions per second. The average response time for the 95th percentile of the requests was recorded at 240 ms, showcasing the node's responsiveness under load. Furthermore, the average response time for the 99th percentile of the requests extended to 820 ms, reflecting the higher end of latency under peak conditions.

Figure 1: Ethereum dedicated node VZO environment performance

Resource utilization peaked at 6.61 CPU cores and 9.86 GB of memory, illustrating the node's efficiency in managing computational and memory resources.

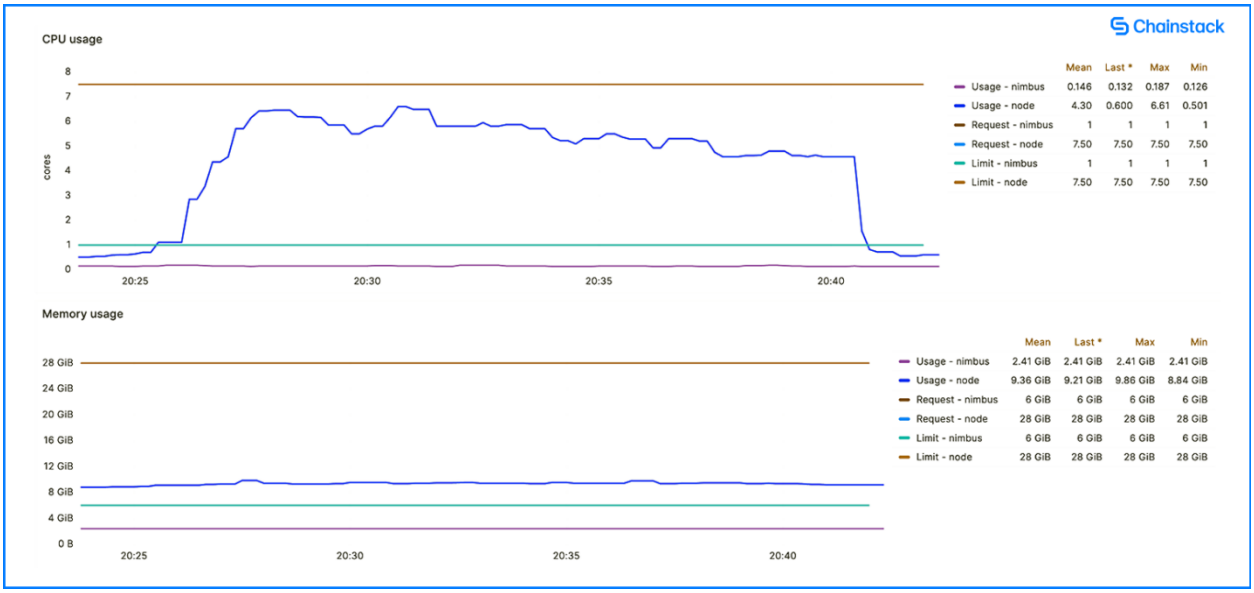

Figure 2: Ethereum dedicated node VZO environment resource usage

Chainstack Cloud Latitude environment

In the Chainstack Cloud Latitude environment, the Ethereum node exhibited a significantly higher maximum RPS of 1670, underscoring its superior capacity to process transactions rapidly. The average response time for the 95th percentile of the requests was observed at 320 ms, with the 99th percentile average stretching to 800 ms. These figures suggest a balanced performance, even as the transaction rate increased.

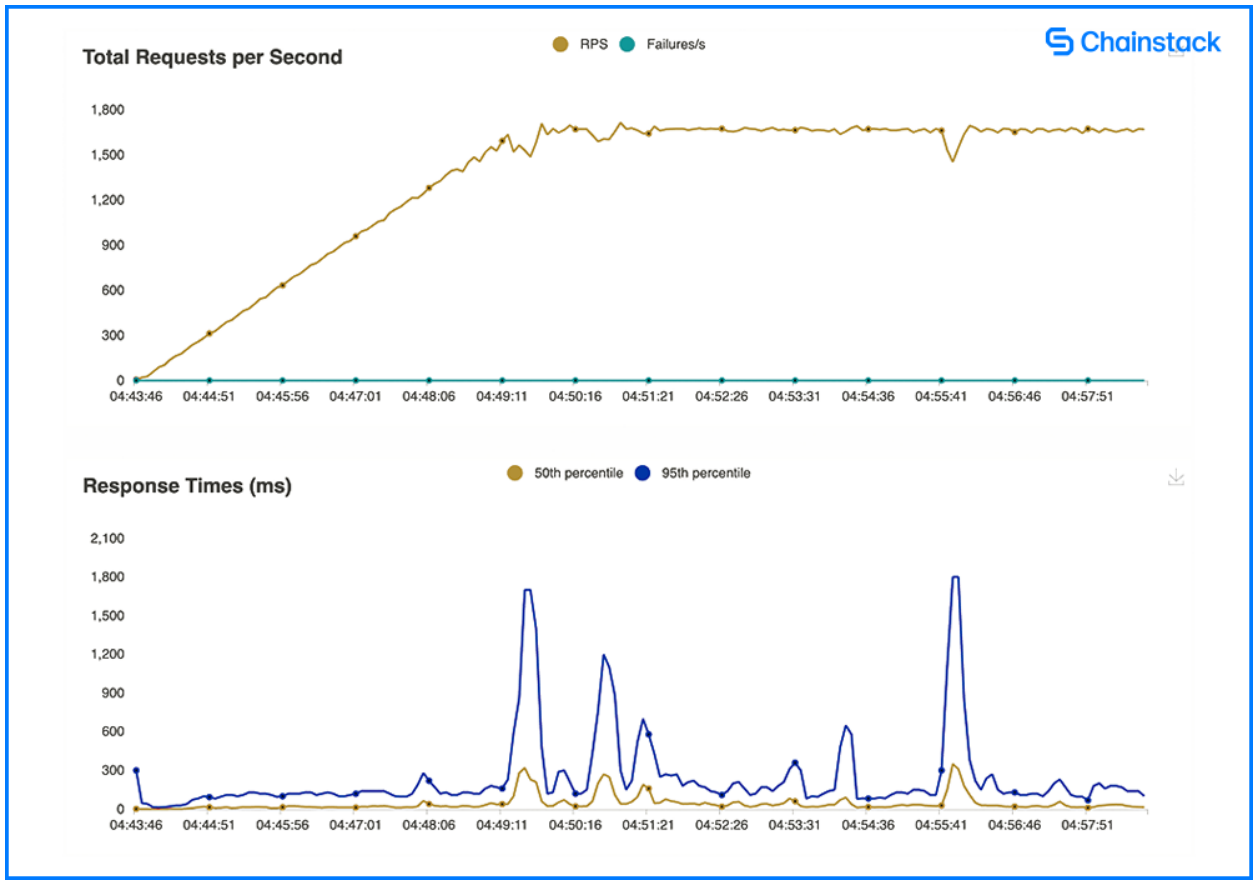

Figure 3: Ethereum dedicated node Chainstack Cloud Latitude environment performance

The node's resource consumption reached a maximum of 6.92 CPU cores and 20 GB of memory, indicating a proportional increase in resource utilization corresponding to the enhanced transaction handling capability.

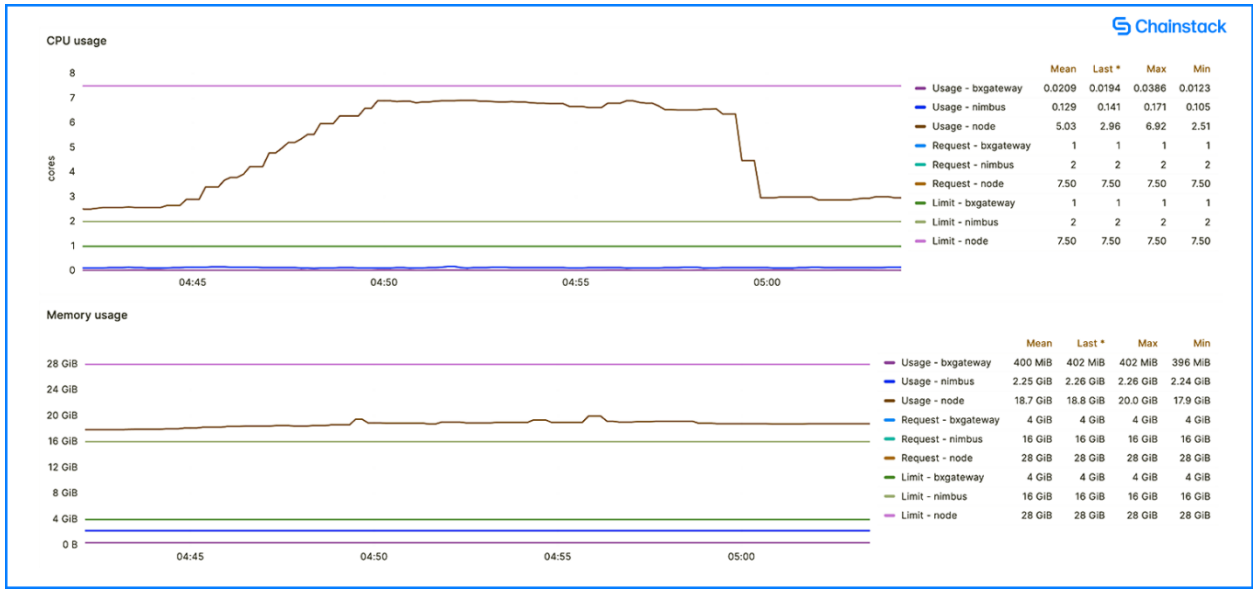

Figure 4: Ethereum dedicated node VZO environment resource usage

AWS environment

The AWS-hosted Ethereum node showed a maximum RPS of 677.6, reflecting its robust transaction processing ability in a cloud environment. The node achieved an average response time for the 95th percentile of 180 ms, the lowest among the tested environments, which underscores its efficiency. The 99th percentile average response time was 590 ms, revealing a commendable performance consistency.

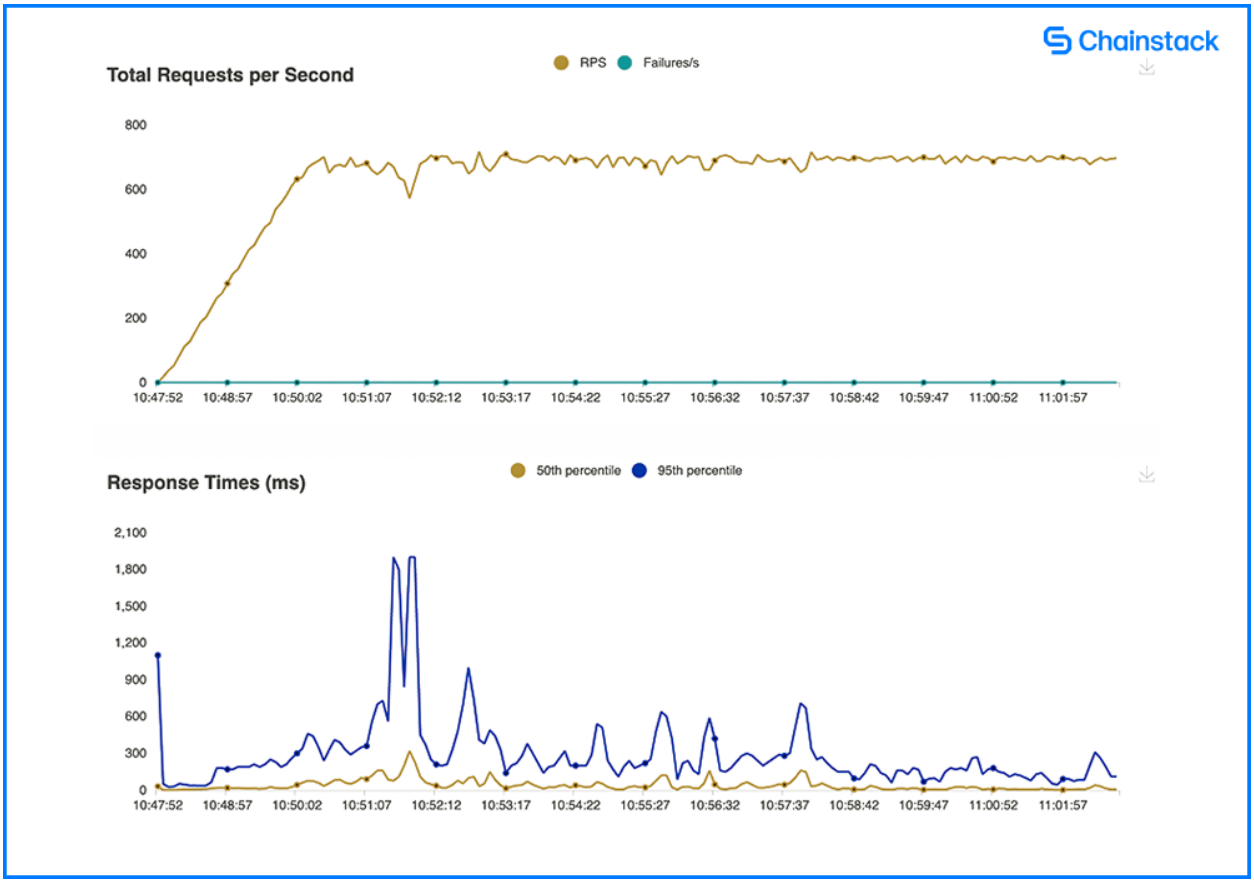

Figure 5: Ethereum dedicated node AWS environment performance

Peak resource usage was noted at 5.4 CPU cores and 15.9 GB of memory, suggesting an optimized balance between max RPS and resource consumption.

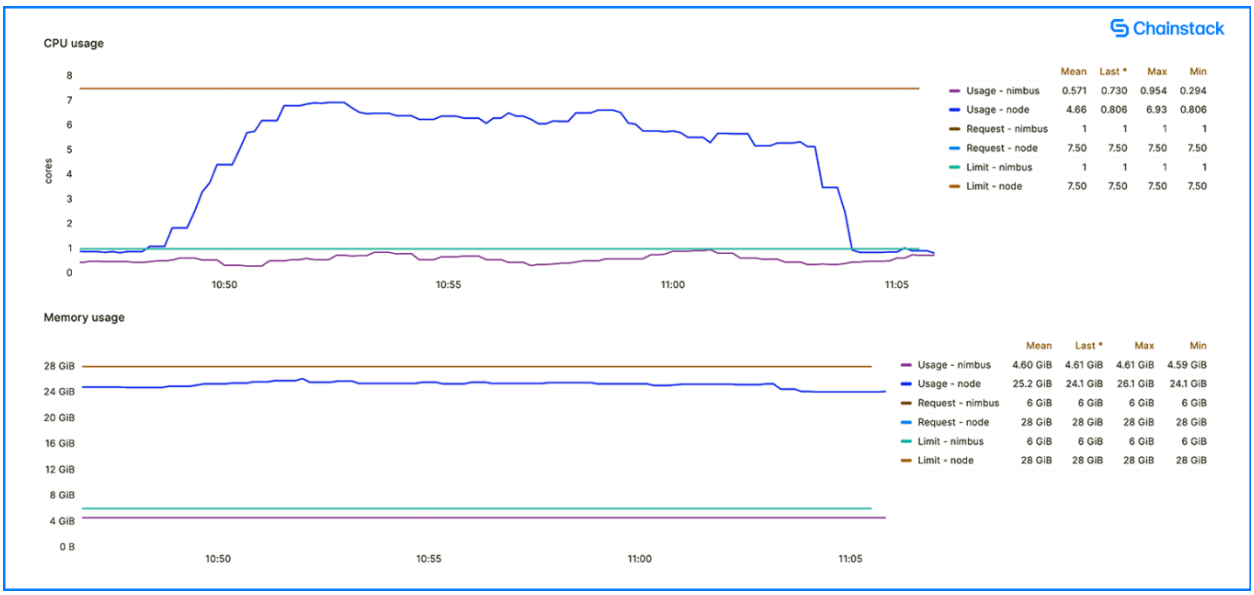

Figure 6: Ethereum dedicated node AWS environment resource usage

Comparison of Ethereum node performance across environments

The comparative analysis of Ethereum node performance across VZO, Chainstack Cloud Latitude, and AWS environments provides valuable insights into how different cloud infrastructures and configurations impact node efficiency, responsiveness, and resource utilization.

This subsection outlines the key differences and highlights observed across the tested environments, facilitating a deeper understanding of each setting's advantages and limitations.

Max RPS

- Chainstack Cloud Latitude demonstrated the highest max RPS of 1670, indicative of its superior processing power and network optimization.
- VZO showed a moderate capacity with a max RPS of 740, reflecting a balanced performance for general use cases.
- AWS presented a slightly lower max RPS at 677.6, which, while competitive, suggests room for optimizations or adjustments based on specific workload requirements.

Response time efficiency

- The AWS environment led in responsiveness, achieving the lowest average response times at the 95th and 99th percentiles at 180 ms and 590 ms respectively, showcasing its efficiency in handling requests with minimal latency.
- VZO followed closely, with average response times at the 95th and 99th percentiles of 240 ms and 820 ms, highlighting its capability to maintain service quality under load.
- Chainstack Cloud Latitude, despite its high max RPS, recorded slightly higher response times at the 95th and 99th percentiles of 320 ms and 800 ms, suggesting a trade-off between processing speed and latency.

Resource utilization

- Chainstack Cloud Latitude observed the highest resource utilization, peaking at 6.92 CPU cores and 20 GB of memory, correlating with its high transaction processing capability.
- AWS demonstrated efficient resource management with 5.4 CPU cores and 15.9 GB of memory usage, balancing performance and consumption effectively.
- VZO maintained modest resource demands at 6.61 CPU cores and 9.86 GB of memory, suitable for scenarios prioritizing cost-efficiency over maximum RPS.

Ethereum implications

The comparative analysis reveals distinct performance characteristics and operational efficiencies of Ethereum nodes across different environments, with each presenting unique advantages:

- Chainstack Cloud Latitude excels in handling high transaction volumes, making it suitable for applications requiring maximum RPS.
- AWS stands out in terms of response time efficiency, ideal for latency-sensitive applications.
- VZO offers a balanced approach, with moderate max RPS and resource utilization, positioned well for cost-effective deployments or applications with average performance requirements.

Polygon

The performance evaluation for Polygon nodes was thoroughly executed across three different environments: VZO, Chainstack Cloud Latitude, and AWS. Each setup utilized the Ethereum Full Top_10 profile with the default configuration settings.

VZO environment

The Polygon node in the VZO setting achieved a Max RPS of 550, showcasing its capacity to handle a significant volume of transactions. The average response time for the 95th percentile was recorded at 270 ms, indicating responsive performance under load. For the 99th percentile of requests, the response time increased to 1400 ms, highlighting the upper latency limits during peak usage.

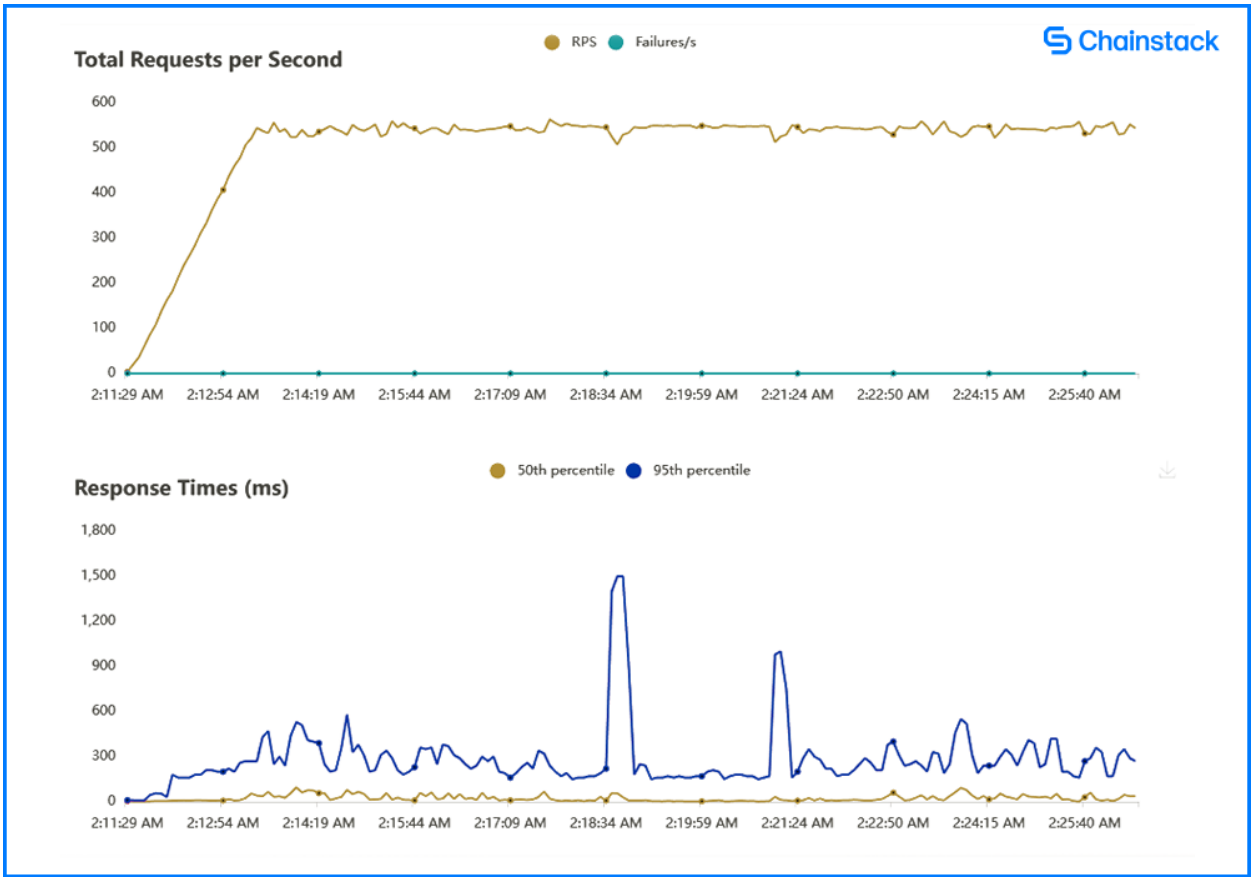

Figure 7: Polygon dedicated node VZO environment performance

The resource usage peaked at 12.7 CPU cores and 89.9 GB of memory, reflecting the node's resource management strategy.

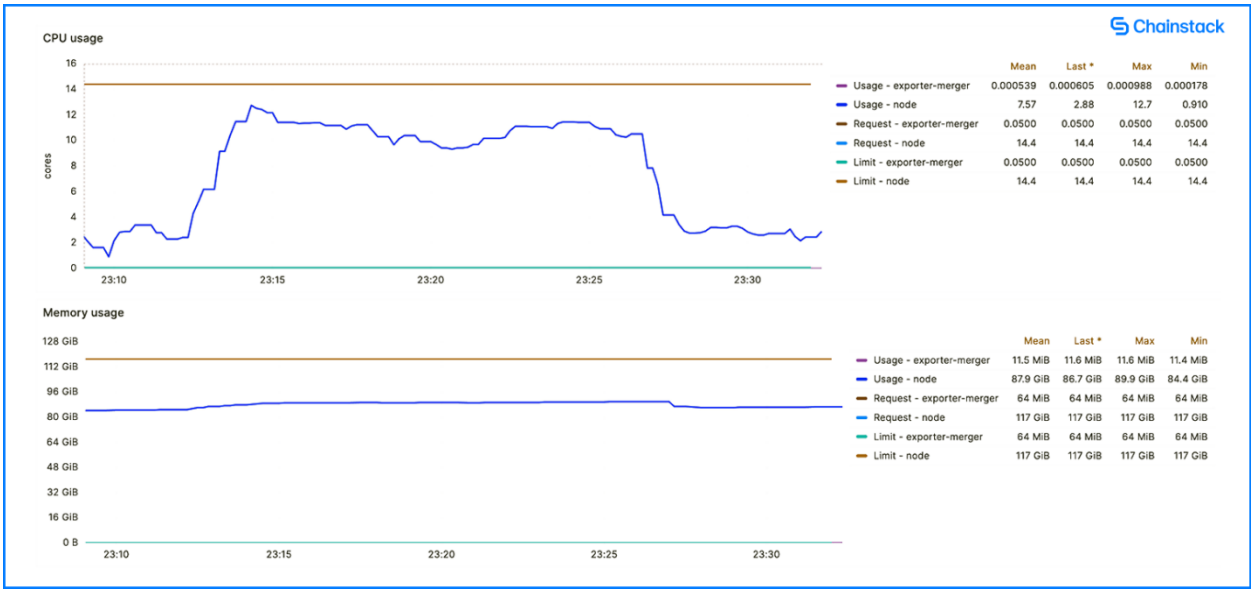

Figure 8: Polygon dedicated node VZO environment resource usage

Chainstack Cloud Latitude environment

In the Chainstack Cloud Latitude environment, the Polygon node displayed a higher Max RPS of 800, emphasizing its enhanced ability to process transactions quickly. The node maintained an average response time for the 95th percentile of 160 ms, and the response time for the 99th percentile was reduced to 960 ms compared to the VZO environment.

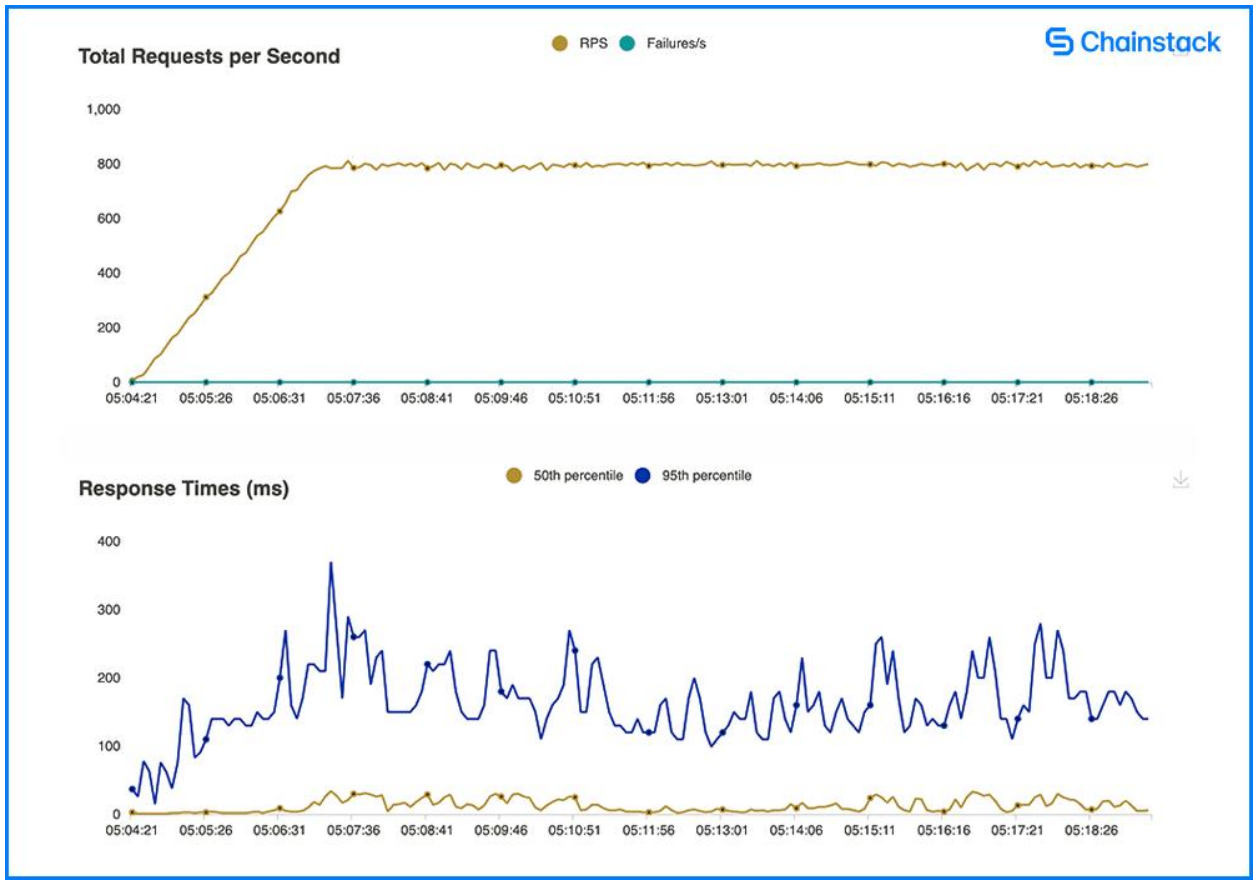

Figure 9: Polygon dedicated node Chainstack Cloud Latitude environment performance

The maximum resource utilization recorded was 11.7 CPU cores for processing and 37.6 GB of memory for data handling.

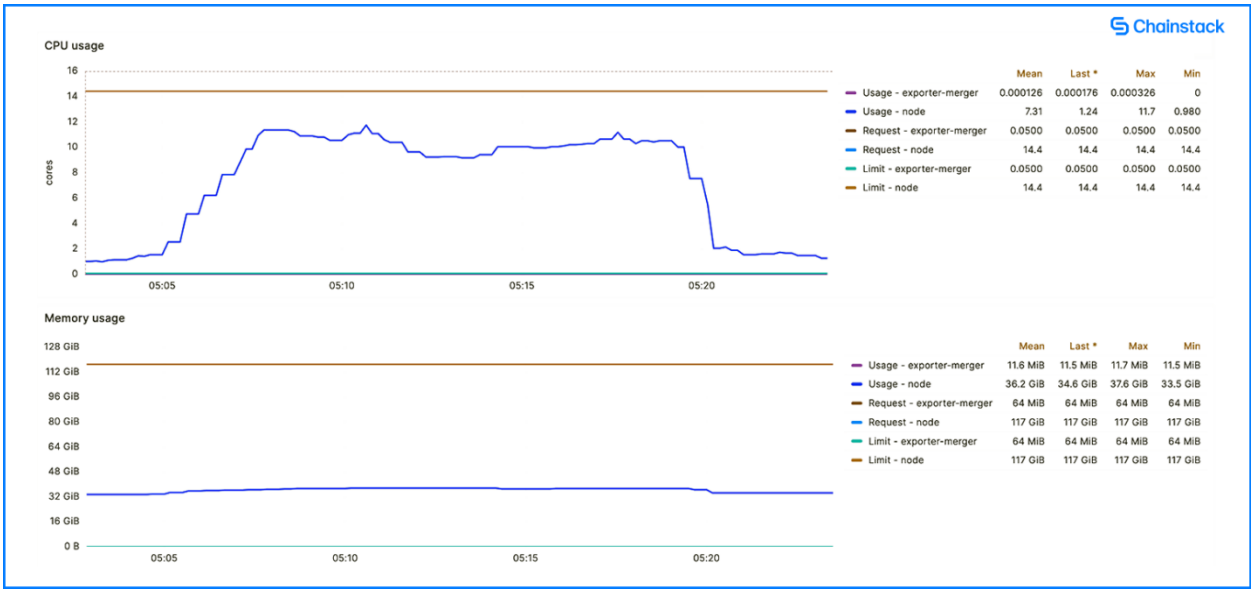

Figure 10: Polygon dedicated node Chainstack Cloud Latitude environment resource usage

AWS environment

The AWS-hosted Polygon node reported a Max RPS of 460, indicating its robust processing capability in a cloud-based infrastructure. This setup achieved an average response time for the 95th percentile of 310 ms, with the response time for the 99th percentile matching the VZO environment at 1400 ms.

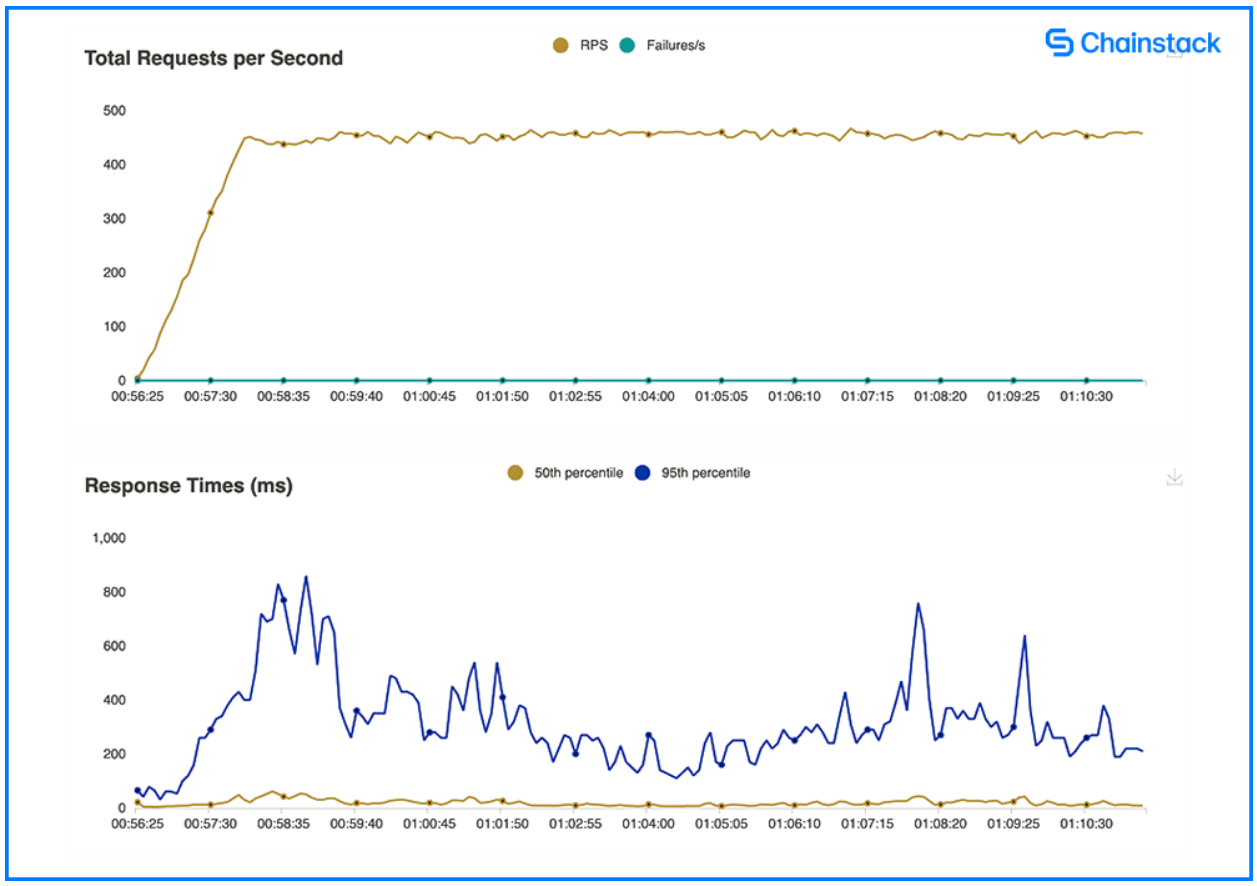

Figure 11: Polygon dedicated node AWS environment performance

The node consumed up to 14.1 CPU cores and 39.1 GB of memory at peak load, illustrating its resource allocation efficiency.

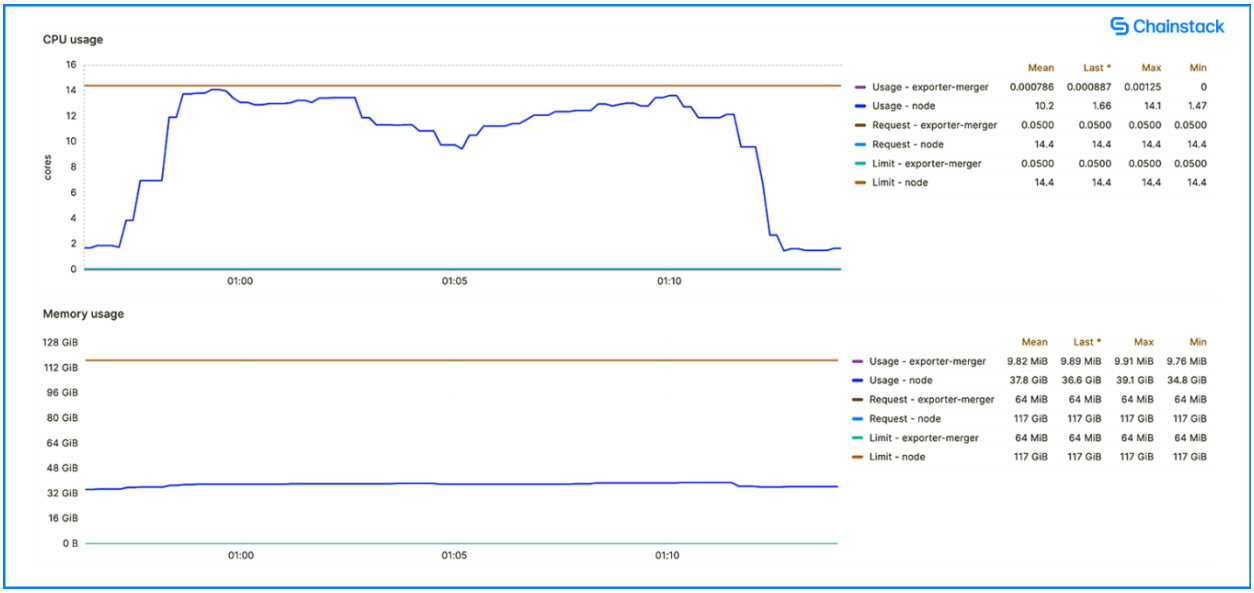

Figure 12: Polygon dedicated node AWS environment resource usage

Comparison of Polygon node performance across environments

This comparative analysis across VZO, Chainstack Cloud Latitude, and AWS environments sheds light on the varying performance dynamics of Polygon nodes, influenced by distinct cloud infrastructures and configurations. Key observations include:

Max RPS

- Chainstack Cloud Latitude showcased the highest max RPS at 800, indicating its superior capability in handling transactions.
- VZO presented a moderate throughput with 550 max RPS, offering balanced performance suitable for various applications.
- AWS demonstrated a throughput of 460 max RPS, suggesting potential areas for optimization to enhance transaction processing capacity.

Response time efficiency

- Chainstack Cloud Latitude led in response efficiency, achieving the lowest average response times at the 95th and 99th percentiles at 160 ms and 960 ms, respectively, showcasing its prowess in minimizing latency.

- AWS and VZO environments exhibited higher response times, with AWS showing slightly slower responses at the 95th percentile but matching VZO at the 99th percentile, indicating a focus on consistency over speed.

Resource utilization

- VZO recorded the highest memory usage at 89.9 GB, along with significant CPU utilization at 12.7 cores, pointing to its resource-intensive operation mode.
- Chainstack Cloud Latitude and AWS demonstrated more efficient resource management, with Chainstack Cloud Latitude utilizing less memory but maintaining high CPU usage, and AWS showing a balanced approach with 14.1 CPU cores and 39.1 GB of memory.

Polygon implications

The comparative analysis reveals distinct performance characteristics and operational efficiencies of Polygon nodes across different environments, each offering unique advantages:

- **Chainstack Cloud Latitude** excels in transaction speed and efficiency, making it suitable for applications that demand high max RPS and quick response times.
- **VZO** is noted for its capacity to handle high-volume transactions with substantial resource usage, ideal for scenarios that require robust processing power.
- **AWS** provides a balanced mix of robust processing and moderate resource consumption, well-suited for a variety of applications that need reliable performance without excessive resource expenditure.

BNB Smart Chain

The performance evaluation for BNB Smart Chain nodes was meticulously executed across three distinct environments: VZO, Chainstack Cloud Latitude, and AWS. Each utilized the `BNB Smart Chain Full Top_10` profile with default configuration settings.

VZO environment

The BNB Smart Chain node in VZO achieved a Max RPS of 460, displaying its capability to process a substantial number of transactions. The average response time for the 95th percentile was noted at 300 ms, indicating effective responsiveness under operational load. The response time for the 99th percentile reached 640 ms, revealing latency peaks during intensive use.

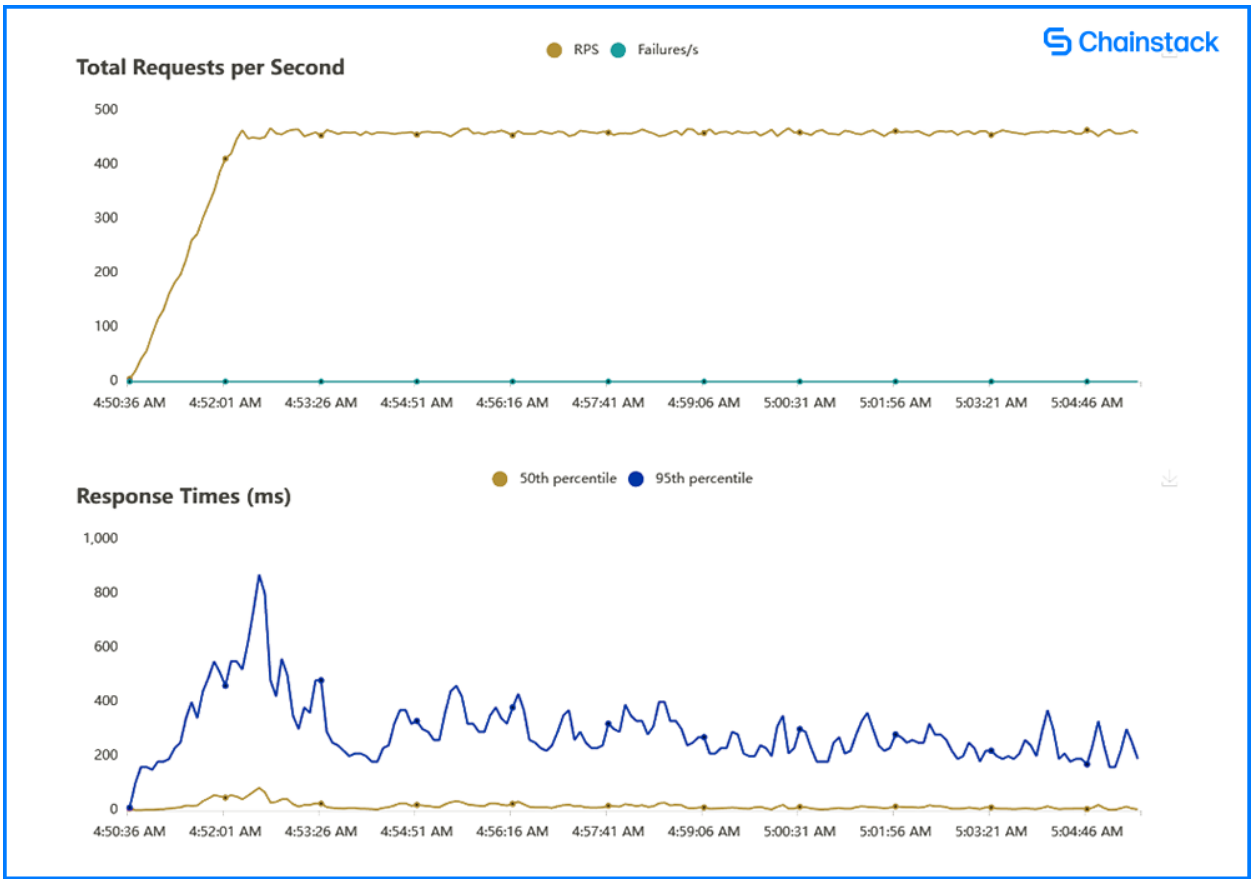

Figure 13: BNB Smart Chain dedicated node VZO environment performance

Peak resource consumption was measured at 13.9 CPU cores and 82.3 GB of memory, demonstrating the node's resource allocation strategy.

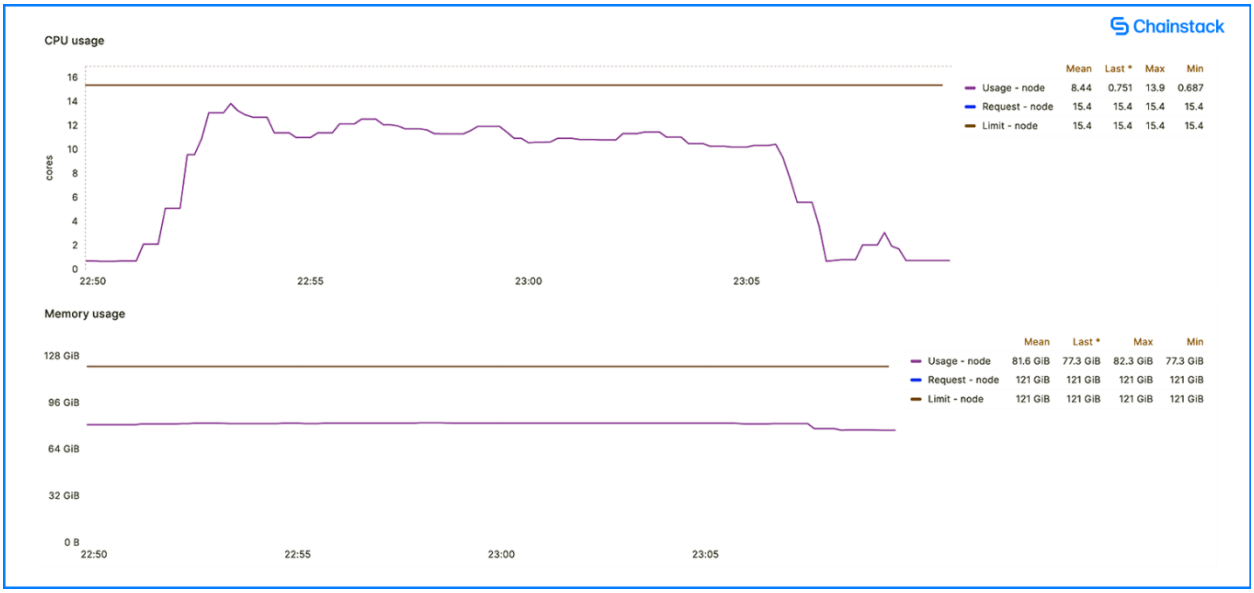

Figure 14: BNB Smart Chain dedicated node VZO environment resource usage

Chainstack Cloud Latitude environment

In Chainstack Cloud Latitude, the BNB Smart Chain node showcased a higher Max RPS of 910, indicating enhanced transaction processing capabilities. The node achieved an average response time for the 95th percentile of 510 ms, with the response time for the 99th percentile significantly reduced to 230 ms, highlighting efficient handling of transactions even under peak loads.

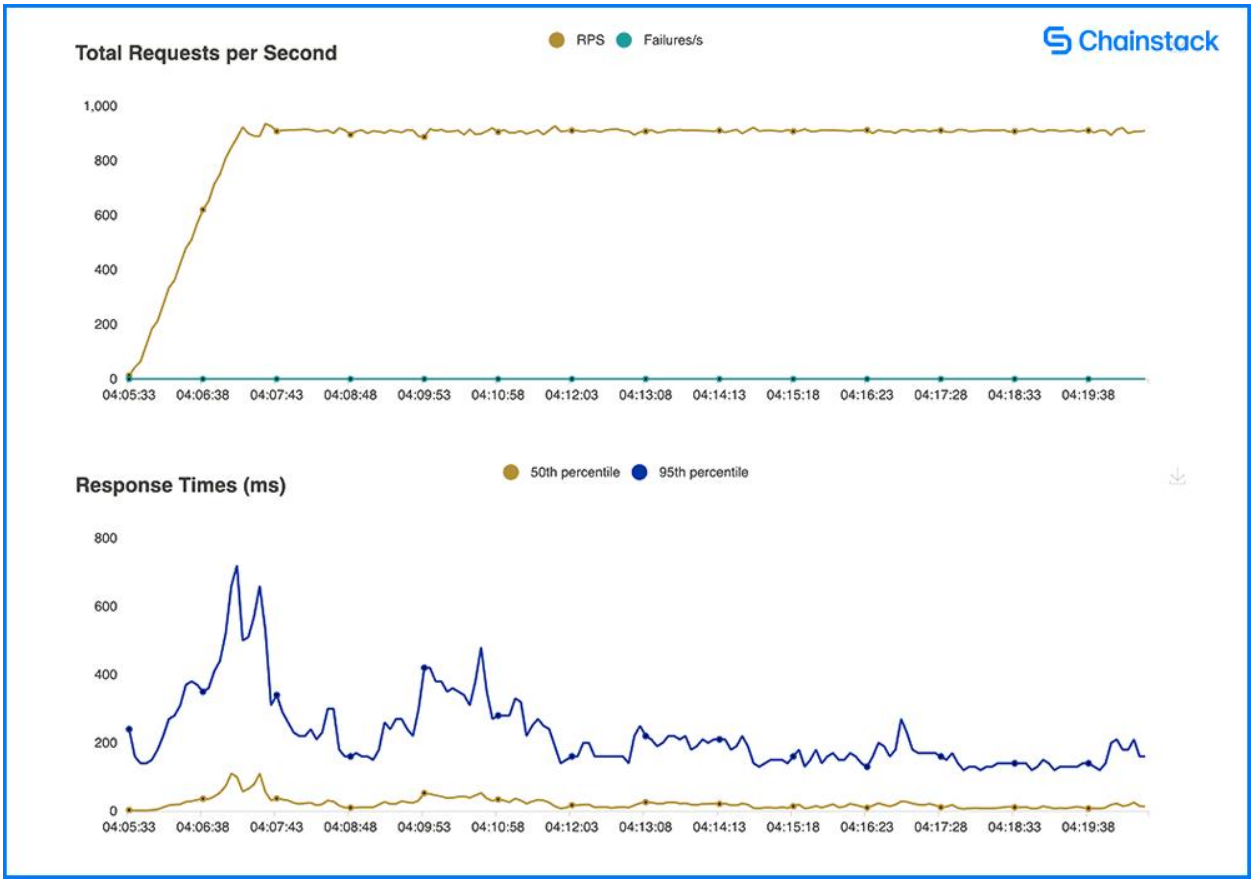

Figure 15: BNB Smart Chain dedicated node Chainstack Cloud Latitude environment performance

Maximum resource usage was observed at 13.5 CPU cores and 21.4 GB of memory, illustrating a balanced approach to computational and memory resources.

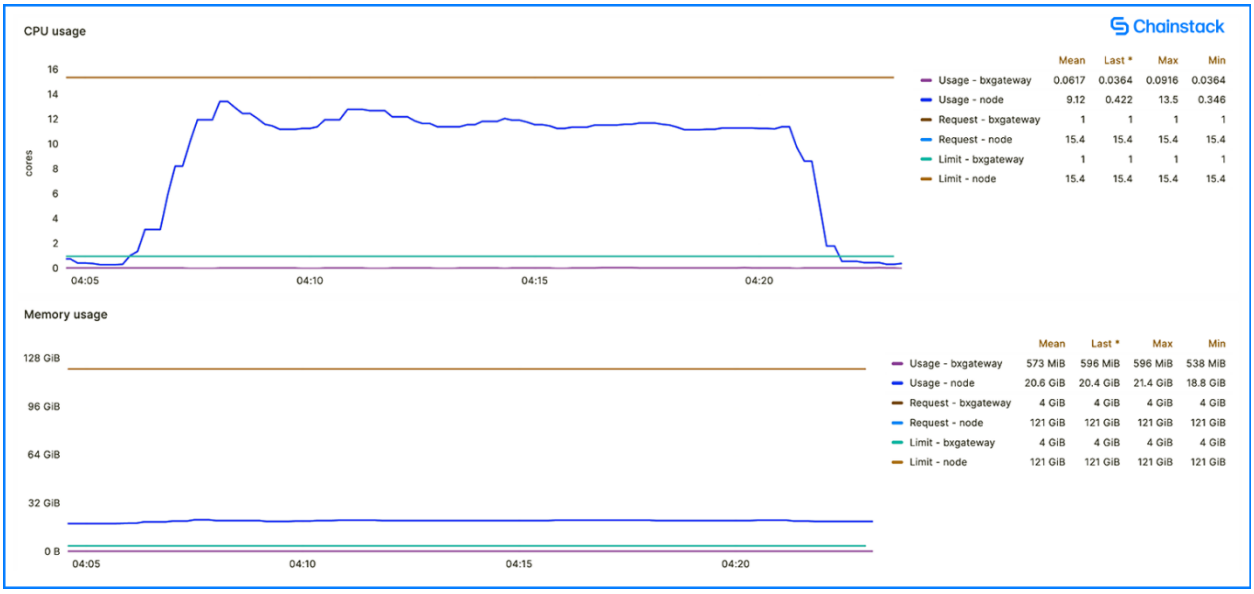

Figure 16: BNB Smart Chain dedicated node Chainstack Cloud Latitude environment resource usage

AWS environment

The AWS environment hosted a BNB Smart Chain node that reported a Max RPS of 710, reflecting its strong transaction processing power in a cloud-based infrastructure. This setup recorded an average response time for the 95th percentile of 320 ms and a response time for the 99th percentile of 860 ms, showcasing consistent performance across a spectrum of operational demands.

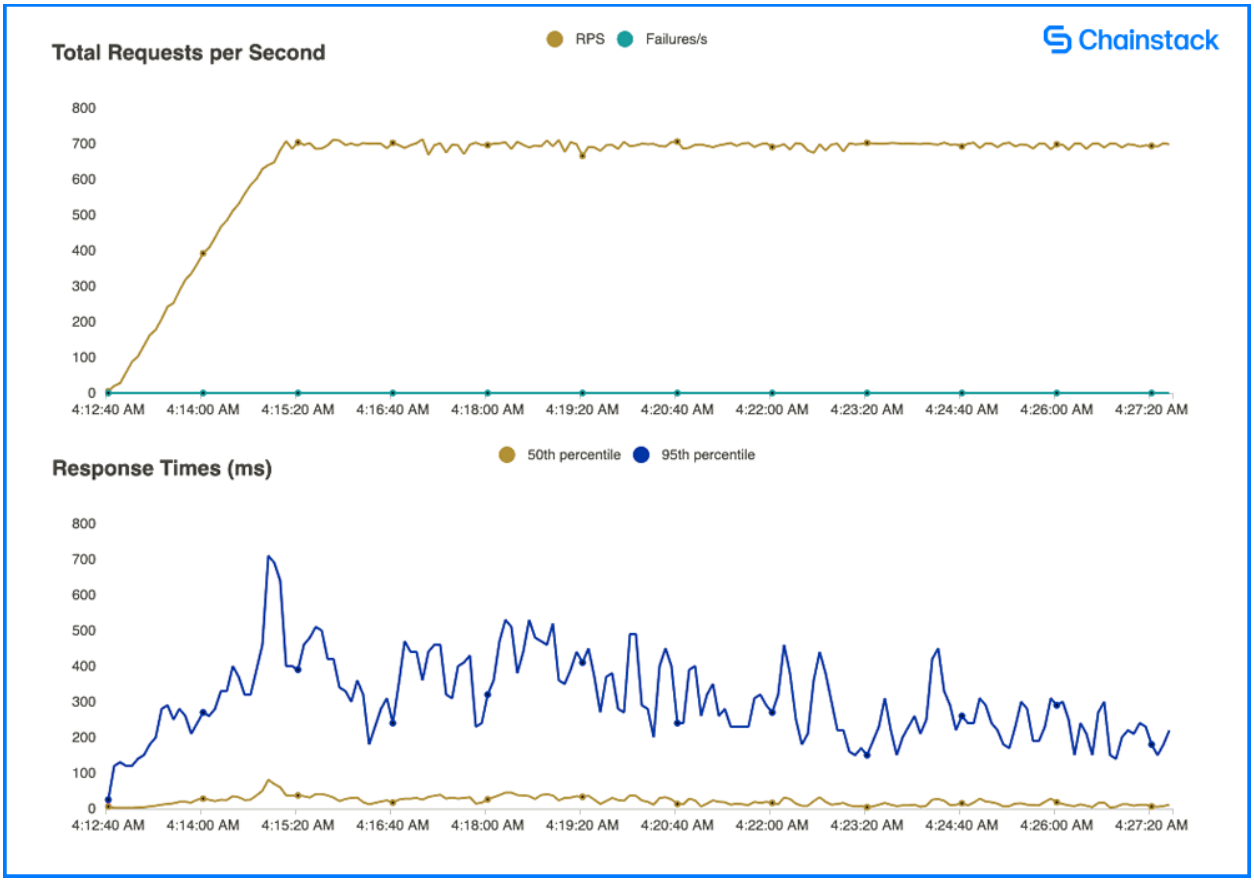

Figure 17: BNB Smart Chain dedicated node AWS environment performance

The peak resource utilization was noted at 12.4 CPU cores and 88.4 GB of memory, indicating an efficient management of resources to support operational requirements.

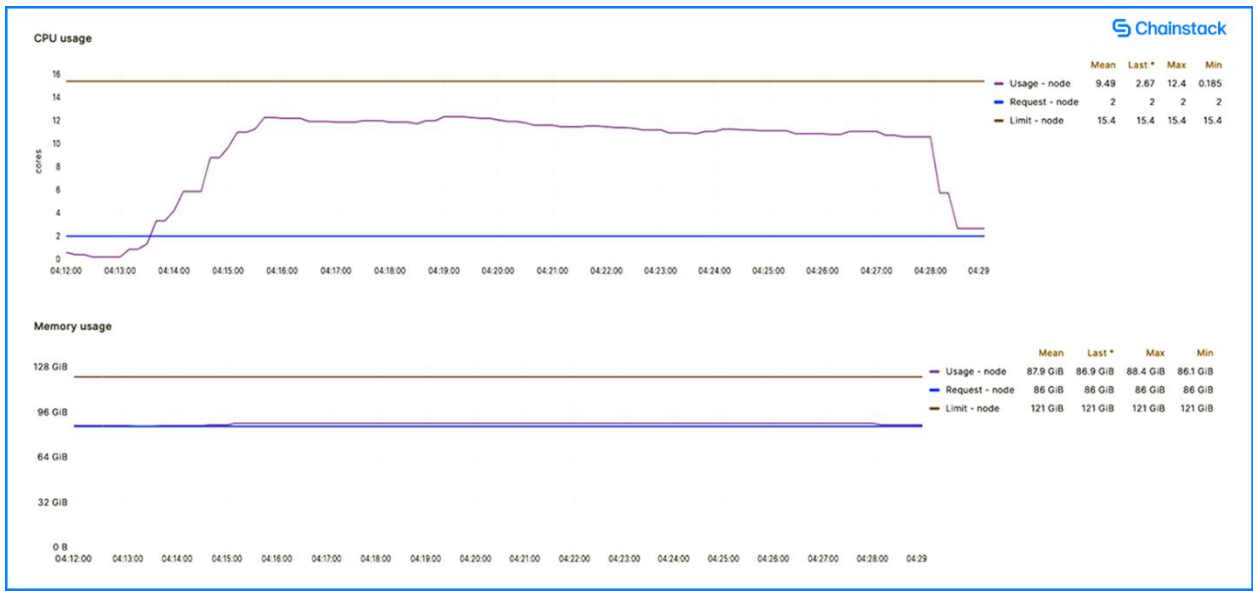

Figure 18: BNB Smart Chain dedicated node AWS environment resource usage

Comparison of BNB Smart Chain node performance across environments

The comparative analysis provides insights into the operational dynamics of BNB Smart Chain nodes across VZO, Chainstack Cloud Latitude, and AWS environments, each reflecting unique performance attributes influenced by different infrastructure setups. Key observations include:

Max RPS

- Chainstack Cloud Latitude stood out with the highest max RPS at 910, showcasing its superior handling capabilities.
- AWS followed with a robust throughput of 710 max RPS, demonstrating strong performance.
- VZO presented a max RPS of 460, indicating its reliable but slightly more constrained transaction processing capacity.

Response time efficiency

- Chainstack Cloud Latitude demonstrated exceptional efficiency with the lowest 99th percentile response time at 230 ms, emphasizing its capability to minimize delays.
- AWS and VZO environments showed higher response times, with AWS providing a slightly better performance at the 95th percentile, indicating their focus on maintaining quality of service.

Resource utilization

- VZO and AWS showed higher memory usage, with 82.3 GB and 88.4 GB respectively, pointing to their strategies for accommodating heavy workloads.
- Chainstack Cloud Latitude demonstrated a more conservative resource usage approach, with a significant reduction in memory consumption to 21.4 GB, suggesting an optimized resource management framework.

BNB Smart Chain implications

The comparative analysis reveals distinct performance characteristics and operational efficiencies of BNB Smart Chain nodes across different environments, each offering unique advantages:

- **Chainstack Cloud Latitude** stands out for its exceptional max RPS and response efficiency, making it an ideal choice for scenarios that demand high-performance capabilities.
- **AWS** offers a balanced mix of robust processing and resource efficiency, suitable for a broad range of applications that require both reliability and moderate resource consumption.
- **VZO** is noted for its reliable performance and substantial resource allocation, well-suited for deployments where consistent operation is prioritized over extreme performance peaks.

Solana

Performance evaluations for Solana nodes were thoroughly conducted in two significant Chainstack Cloud environments: NYC1 and AMS1. Each environment hosted tests under

two distinct load profiles: one excluding the `getProgramAccounts` (GPA) method and another including it, focusing on how the inclusion of GPA impacts transaction processing capabilities.

Chainstack Cloud NYC1 environment with Staking GPA

In the Chainstack Cloud NYC1 setting with GPA enabled, the Solana node achieved an impressive average RPS of 2203.62. The median response time for transactions was notably swift at 55ms, though the response time at the 99th percentile peaked at 5000 ms, indicating a variance under heavy load conditions.

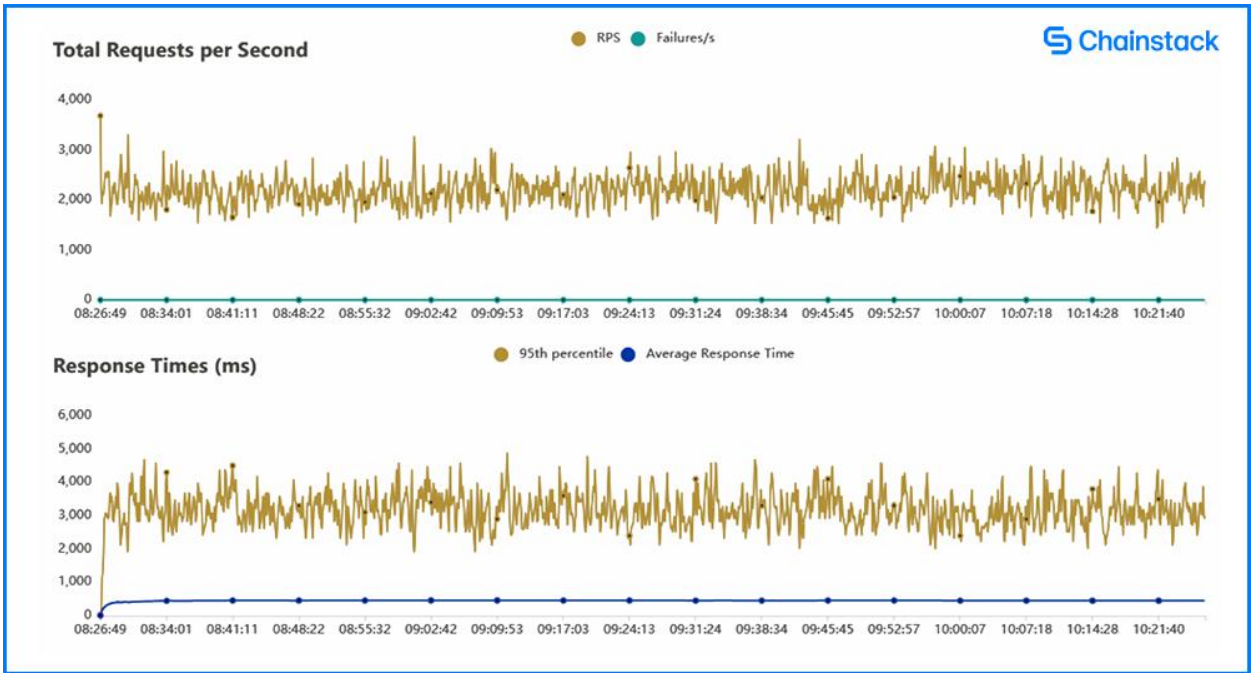

Figure 19: Solana Staking GPA dedicated node NYC1 environment performance

Resource usage was significant, with an average CPU utilization of 41.6 and peaking at 42, alongside an average memory usage of 845 GB, maxing out at 880 GB.

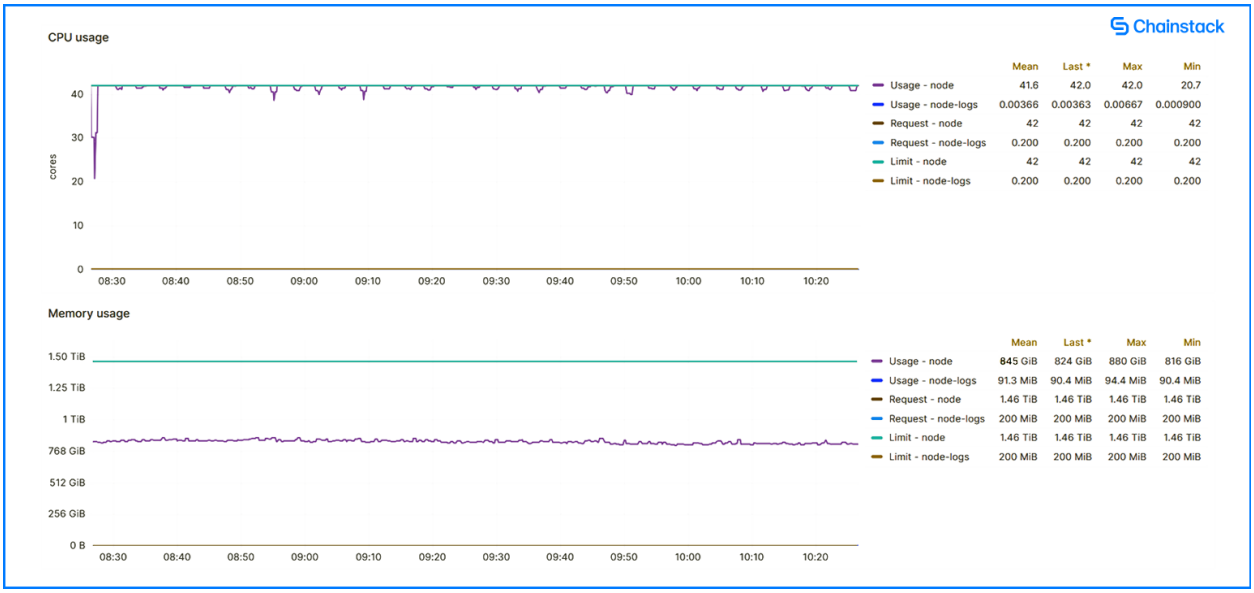

Figure 20: Solana Staking GPA dedicated node NYC1 environment resource usage

Chainstack Cloud AMS1 environment with Staking GPA

The Chainstack Cloud AMS1 environment with GPA saw an average RPS of 1811, with a slightly longer median response time of 69ms. The response time at the 99th percentile was better controlled at 3300 ms.

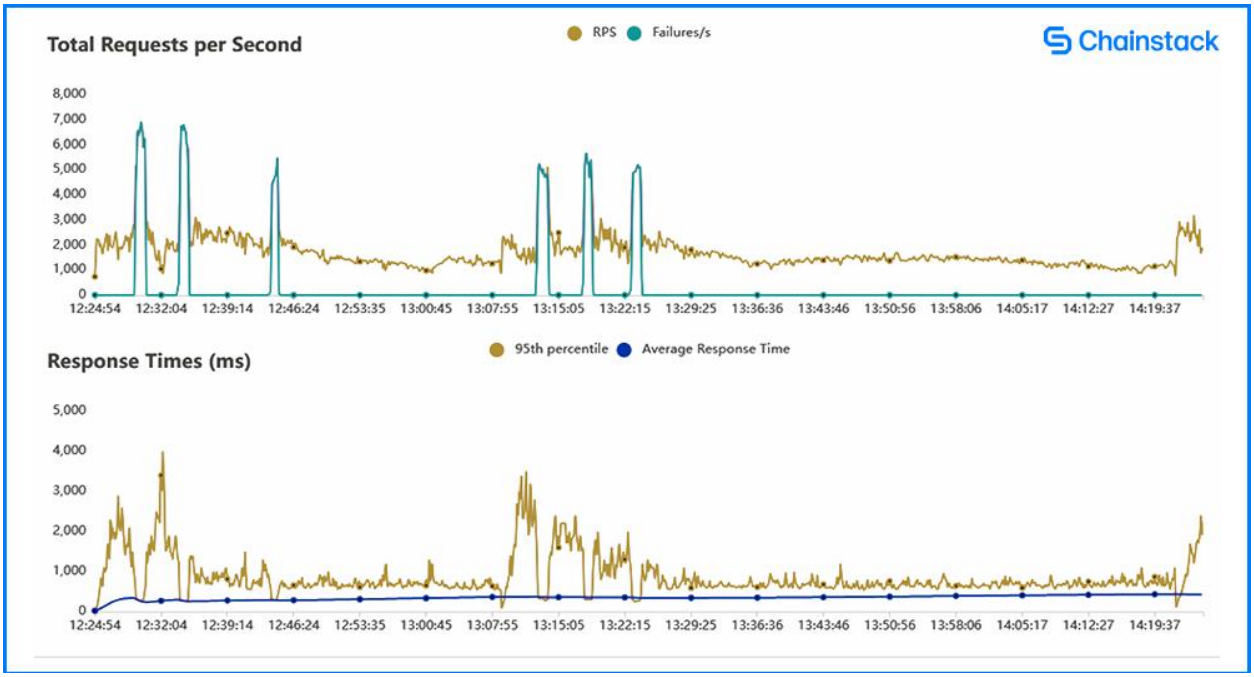

Figure 21: Solana Staking GPA dedicated node AMS1 environment performance

CPU usage averaged at 31, spiking to 42, while memory usage was considerably higher, averaging at 1005 GB and reaching up to 1060 GB.

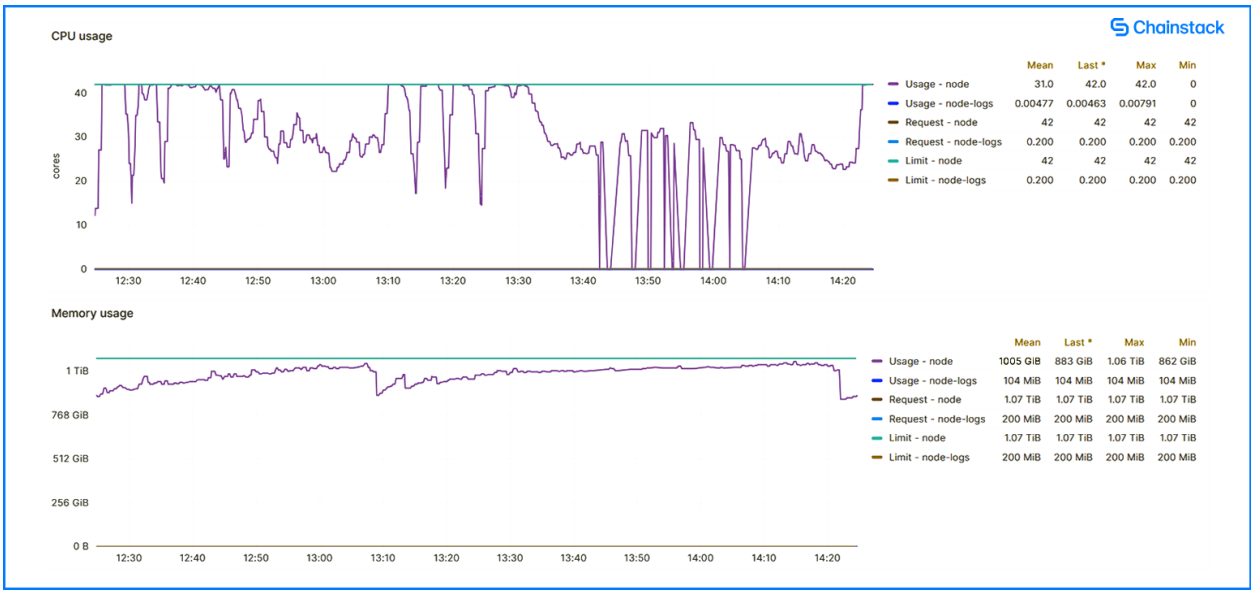

Figure 22: Solana Staking GPA dedicated node AMS1 environment resource usage

Chainstack Cloud NYC1 environment no GPA

Without GPA, the Chainstack Cloud NYC1 node demonstrated an astonishing average RPS of 5906.84, with a median response time of 66ms and a response time at the 99th percentile of 2000 ms, showing enhanced performance and efficiency.

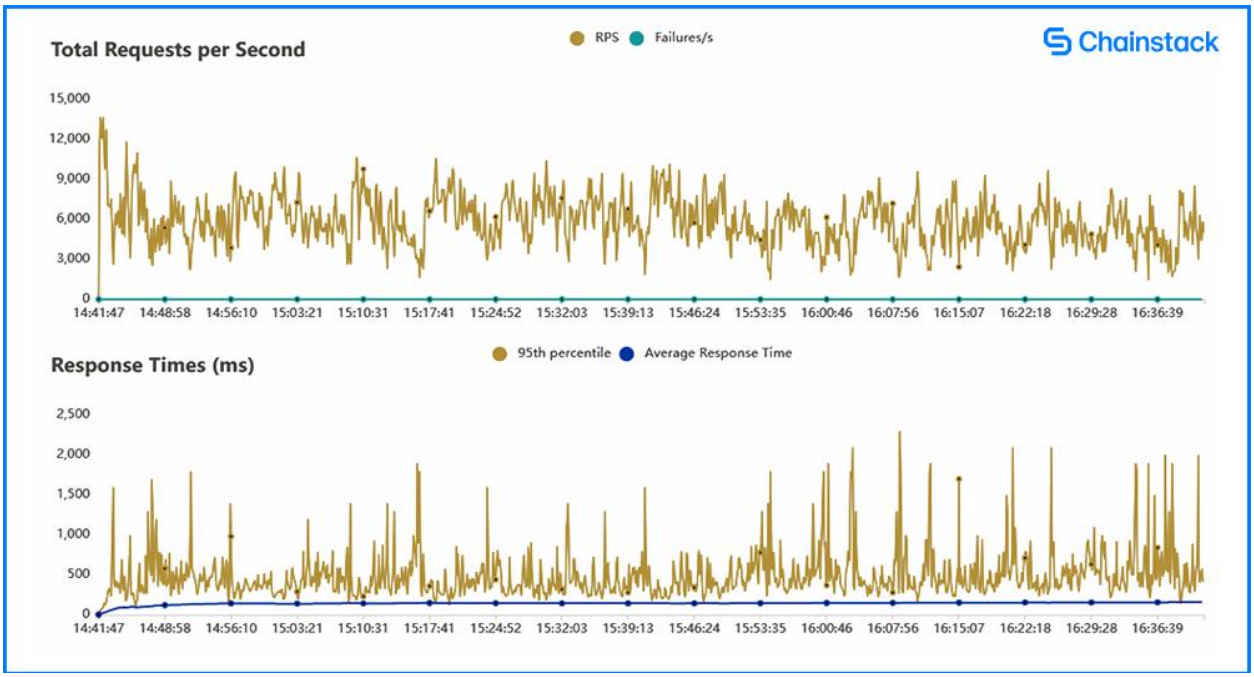

Figure 23: Solana No GPA dedicated node NYC1 environment performance

CPU and memory resources were closely managed, with average CPU at 40.4 and memory at 860 GB.

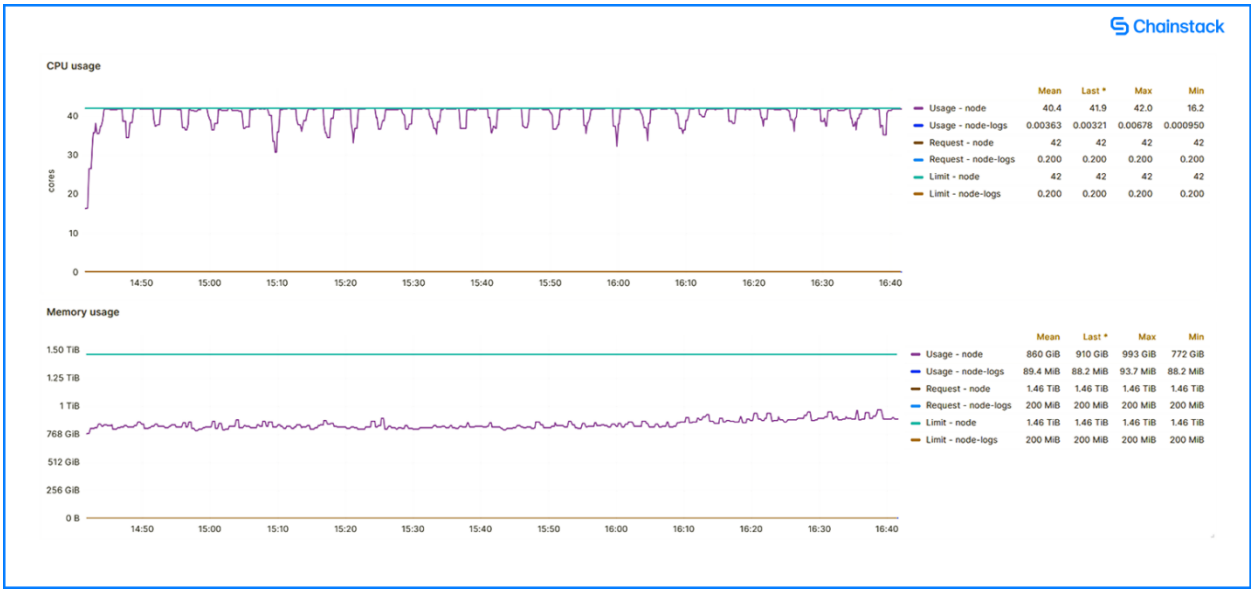

Figure 24: Solana No GPA dedicated node NYC1 environment resource usage

Chainstack Cloud AMS1 environment no GPA

In Chainstack Cloud AMS1 without GPA, the node displayed an average RPS of 1389, with median and 99th percentile response times at 67ms and 3400 ms, respectively.

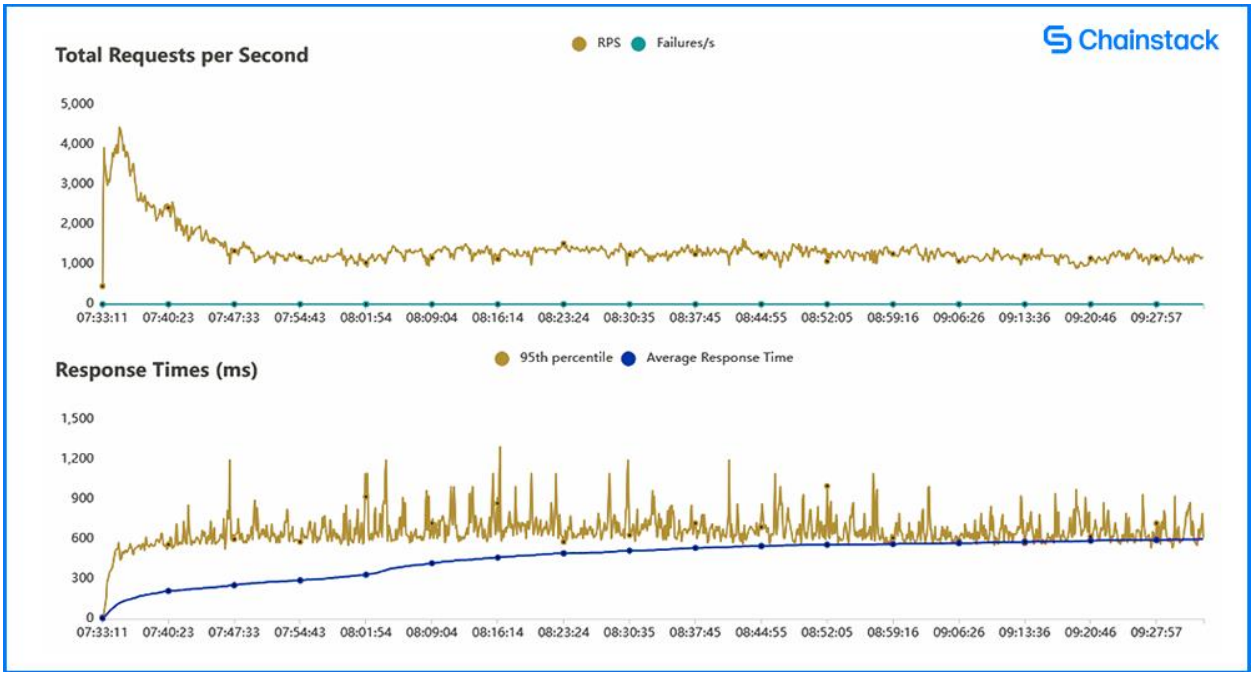

Figure 25: Solana No GPA dedicated node AMS1 environment performance

This configuration saw a reduced average CPU usage at 18.0, peaking at 35.4, while memory usage averaged at 1040 GB, maxing at 1070 GB.

Figure 26: Solana No GPA dedicated node AMS1 environment resource usage

Comparison of Solana node performance across environments

GPA vs. No GPA

- Nodes without GPA generally achieved higher RPS, especially evident in the NYC1 environment. This indicates a potential trade-off between transaction speed and the stability provided by staking incentives.
- The NYC1 environment with GPA outperformed AMS1 in RPS and had a better response time at the 99th percentile, suggesting more efficient handling of peak loads.
- Without GPA, NYC1 achieved significantly higher RPS and faster response times at the 99th percentile than AMS1, showcasing its superior capacity to process transactions rapidly.

NYC1 vs. AMS1

- NYC1 consistently outperformed AMS1 in terms of RPS, both with and without GPA, highlighting location-specific advantages in handling transactions.
- AMS1 environments showed more substantial memory usage but less CPU utilization, suggesting different resource management strategies between the two locations.

Solana implications

The comparative analysis of Solana nodes across different configurations and environments highlights distinct performance characteristics and operational efficiencies, each presenting unique advantages:

- **With GPA load profile:** When the load profile includes the `getProgramAccounts` method, the nodes demonstrate enhanced stability, making this configuration suitable for applications that prioritize consistent performance over peak transaction speeds.
- **Without GPA load profile:** When the load profile excludes the `getProgramAccounts` method, the nodes typically achieve max RPS. This setup is ideal for scenarios that demand maximum processing speed and are less sensitive to fluctuations in stability.

- **Environment considerations**: The choice between environments like Chainstack Cloud AMS1 and NYC1 shows significant impacts on both resource utilization and transaction processing capabilities, indicating the importance of selecting the right environment based on specific workload requirements.

Overall implications

Based on the detailed performance evaluations across various blockchain protocols and environments, several strategic implications emerge that can guide future optimizations and configurations:

Tailored configuration for optimal performance

- **Ethereum:** The superior performance of Ethereum in the Chainstack Cloud Latitude environment, with the highest RPS and efficient resource utilization, highlights the importance of tailoring configurations to balance max RPS, response times, and resource use.
- **BNB Smart Chain:** The low resource use combined with high transaction rates in the Chainstack Cloud Latitude environment suggests that optimizations focused on computational efficiency can significantly enhance performance.
- **Polygon and Solana:** Both protocols demonstrated that different load profiles can vastly affect performance outcomes. For Polygon, Chainstack Cloud Latitude excelled in transaction speed, while for Solana, tests excluding the `getProgramAccounts` method from the load achieved higher max RPS, emphasizing the importance of specific load strategies tailored to the unique demands of each blockchain.

Cloud environment selection

The choice of cloud environment has proven crucial, as demonstrated by the varying performance metrics across AWS, Virtuozzo, and Chainstack Cloud:

- **Ethereum and BNB Smart Chain:** Performed best in environments optimized for their specific workload characteristics. For example, AWS provided the lowest response times for Ethereum, while Chainstack Cloud Latitude maximized max RPS for BNB Smart Chain.
- **Polygon and Solana:** These protocols showed that different testing environments can dramatically impact transaction speeds and system resilience. Solana, in particular, highlighted the necessity for environments capable of handling extremely high workloads, especially when the load profile includes demanding methods such as `getProgramAccounts`.

Impact of GPA on node efficiency

Incorporating the `getProgramAccounts` method in the load significantly impacts node efficiency. Tests involving GPA demonstrated lower peak performance, emphasizing the need for its careful use, particularly in environments with high transaction volumes where consistency and speed is key.

Performance consistency across environments

Ensuring consistent performance across different environments is essential, as demonstrated by all protocols analyzed. A standard set of performance benchmarks should be developed for each protocol to maintain service quality irrespective of the deployment environment. This consistency is particularly crucial for applications requiring high reliability and low latency.

Conclusion

The performance assessments detailed in this study provide crucial insights into the operational capabilities of blockchain nodes across multiple environments and protocols including Ethereum, Polygon, BNB Smart Chain, and Solana. These insights are instrumental for stakeholders in making well-informed decisions regarding infrastructure planning and node deployment strategies.

- **Ethereum** nodes demonstrated varying capacities with the Chainstack Cloud Latitude environment outperforming others by handling up to 1670 RPS, showcasing the importance of environment-specific optimizations for achieving high max RPS and efficient resource utilization.
- **Polygon** nodes revealed the importance of balancing load capacity and response times, with the Chainstack Cloud Latitude environment achieving the best max RPS and response balance, further emphasizing the role of strategic environment alignment to operational demands.
- **BNB Smart Chain** displayed robust transaction handling capabilities, particularly in the Chainstack Cloud Latitude environment which achieved the highest RPS and notably efficient response times under peak load, underscoring the potential for resource efficiency in high-demand scenarios.
- **Solana** RPC methods such as `getProgramAccounts` can be very demanding on the nodes, so the stability and reliability of the underlying infrastructure is of utmost importance. Between the two environments, NYC1 exhibited this well, while AMS1 struggled under a GPA load and started returning errors in some instances during the tests.

The study's findings underline the necessity for tailored node configurations that optimize performance metrics such as max RPS, response times, and resource utilization. Moreover, the choice of cloud environment is shown to significantly impact the performance, necessitating careful selection to match the blockchain's operational requirements and workload characteristics.

Ultimately, these benchmarks serve as a guide for deploying blockchain nodes in a manner that aligns with specific application needs and performance expectations, ensuring that the infrastructure is not only effective at meeting current demands but also scalable for future growth and technological advancements.

Chainstack is the limitless Web3 development stack to build applications for every scale, powering applications in DeFi, NFT, gaming, analytics, and everything in between. From startups to large enterprises, Chainstack enables thousands of companies to cut down the time to market, costs and risks associated with creating and scaling decentralized applications. By offering fast, reliable, and easy-to-use infrastructure solutions distributed globally, we make sure innovators can focus on what's important.

Chainstack provides unified access to multi-chain node and data APIs, distributed compute and storage, and the ever-expanding list of services and tools to build amazing applications across all prominent Web3 protocols. Our enterprise-grade platform has intuitive developer experience, battle-tested reliability, predictable pricing, and outstanding customer support.

For any questions or feedback, contact us at contact@chainstack.com