Stateless Executor

Stateless Executor

1.Rapid Startup

The stateless executor doesn't need to load the entire state at startup. As a result, the startup process is exceptionally fast.

2.Strong Horizontal Scalability

Due to its stateless nature, nodes can easily be added or removed to meet different demands and loads, thus increasing the flexibility of horizontal scaling.

3.Lower Hardware Requirements

The stateless executor doesn't need to store the whole state, thus requiring relatively lower hardware, which reduces costs.

How Stateless Executor Works

1.Optimized Network Communication Using Websocket

Stateless executor employs websocket instead of traditional HTTP for communication. The multi-connection and multi-threaded approach allows for bulk retrieval of remote data, enhancing throughput, much like multiple roads transporting goods simultaneously.

2.Optimized State Retrieval

The stateless executor optimizes state retrieval through several key techniques:

- Pre-computing Transaction States
- : The system first calculates the required states, then executes different transactions in parallel, akin to buying a complete list at once.
- Bulk Retrieval of States
- : Retrieving states in bulk reduces network waiting times.
- State Trie Optimization Specific to Stateless
- : Aggregating potential network requests minimizes latency.
-

3.Cache Mechanism Based on State Trie Node

Through the cache mechanism based on state trie nodes, the stateless executor can reuse state values across different blocks, reducing the need for remote retrieval.

Previous Block Builder Architecture Next Private RPC Relay Last updated1 month ago On this page *Stateless Executor * How Stateless Executor Works

Was this helpful?