Bullish

NEWS RELEASE

Bullish Custody: Reliability pillars – building resilience with the Taurus blockchain

2024-09-05

TLDR: This post explores the advanced mechanisms of the Taurus blockchain, focusing on high availability, observability, and disaster recovery to provide reliable 24/7 operations at Bullish. We delve into the architecture and strategies that underpin these capabilities.

By Danny Siu, Vincent Chan, Engineers, Eric Ma, Mohammad Nauman, Directors of Engineering

In our ongoing series detailing the innovative custody solutions[1] at Bullish, this entry focuses on the advanced mechanisms that provide high availability (HA), observability, and disaster recovery (DR) within the Taurus blockchain. For those who have not read our previous blogs, we recommend starting with our introductory blog, "**Bullish Custody: Navigating the future of digital asset security**," to gain a comprehensive overview of the series' context. Additionally, our first part in this series, "**Bullish Custody: Foundation of trust - exploring the Taurus blockchain's architectural evolution**," provides in-depth insights into the architectural foundations of the Taurus blockchain.

1. Understanding the role of the Taurus blockchain

The Taurus blockchain is integral to Bullish's infrastructure, managing key functionalities such as deposits, withdrawals, and ledger updates. Its design prioritizes high availability, disaster recovery, and observability to maintain seamless operations even under adverse conditions. This allows continuous service, even in the event of hardware breakdowns, network failures, or data center outages.

2. Taurus blockchain architecture

The Taurus system is deployed across two regions and three Availability Zones (AZs) for enhanced disaster recovery. Each region hosts a complete deployment of all functional components, facilitating regional-level

redundancy. Within each region, block producing, block syncing, streaming, query, and backup services are distributed across AZs and organized into logical clusters, maintaining data consistency through synchronized producers.

For simplicity, this diagram illustrates the topology of a single region, but the deployment model can scale to multiple regions. In the multi-region deployment, the same topology is mirrored across regions, with sync nodes interconnected for block replication.

2.1 Overview

In cloud data centers, Availability Zones (AZs) have different failure modes and usually recover quickly from a single AZ failure. However, it's rare for multiple AZs to fail simultaneously in one region, which is considered a disaster scenario handled by Taurus's architecture. Hence, the entire Taurus system is deployed across two regions, with each region hosting a complete deployment across three AZs, to facilitate service high availability and regional-level disaster recovery.

Within each region, block producing, block syncing, streaming, query, and backup services are distributed across AZs and organized into logical clusters with strict consistency requirements. Producers within each region are synchronized to maintain data consistency through the producer high availability mechanism, which we will cover in detail later. An elected active role within each cluster enhances resilience against single-zone incidents and enables seamless rolling releases on a daily basis.

2.2 System high availability for business availability

High availability is critical for avoiding single points of failure. Key strategies include:

- Redundancy: The system components are strategically deployed across multiple Availability Zones (AZs), reducing the chances of service interruption even if an entire AZ fails.
- Replication: Blockchain blocks and queue data are duplicated across AZs for added reliability.
- Failover and Load Balancing: Our system employs hot standby configurations for block producer nodes (BP) and uses load balancing and failover strategies for additional resilience.
- Monitoring: Real-time process/instance monitoring tools and a dedicated management console provide insights into the overall system health and facilitate timely troubleshooting.

With these strategies in place, every component of the system operates in concert, so that the failure of a single component or even an entire AZ does not compromise data integrity or availability.

Message queues

Taurus employs RabbitMQ for high availability in transaction and event processing. The RabbitMQ cluster is set up

across three AZs, with messages replicated to allow continuity even if a producer node fails. If a Taurus producer node fails, the system seamlessly fails over to a standby node, preserving operational continuity.

Block producers

Block producers are crucial for executing transactions and creating new blocks. The Raft consensus protocol is designed to ensure that even if one producer fails, no transactions or blocks are lost. The three block producers form a high availability group to maintain block production without data loss, ensuring continuity through the Raft consensus.

Block sync service

Block producers and sync nodes are interconnected in a mesh network, linking each sync node with each producer node within the same region. The redundant links ensure that any active producer has guaranteed dedicated links to the sync nodes. Producer blocks are broadcasted through the network.

If any producer, sync node, or AZ fails, connections from the nodes in other AZs will sync the blocks across the nodes and AZs to ensure block propagation.

The sync service nodes are behind a DNS-based load balancer, which dynamically adds or removes the sync node IPs based on the health of the nodes. The leaf nodes for query and streaming connect to the sync service through the DNS name, automatically finding available sync nodes in the cluster. The DNS-based load balancer ensures that all links from the leaf nodes are evenly distributed across available nodes, minimizing workloads and providing horizontal scalability to support large clusters when more leaf nodes are added.

Streaming and query services

High availability for streaming and query nodes is achieved through redundant deployment across AZs. Three streaming nodes and three query nodes are deployed in all AZs.

For the streaming service, all three nodes send messages to the queues, ensuring that a single streaming node failure does not result in missing messages. At Bullish, oracles are designed to handle duplicated messages for availability, using business logic internal sequence numbers to skip handling duplicate messages.

For the query service, the same DNS load balancing mechanism as the sync service node is used to achieve high availability and load balancing, ensuring horizontal scalability of the service.

Backup service

We set up redundant backup nodes in each region, and they work independently to create backups and upload them to highly available cloud storage services. These independent backups provide high availability, ensuring new

backups are created even if a single AZ fails. For nodes that fail and need to be rebuilt, the restore tools for Taurus will automatically fetch the backups from the cloud storage service and restore the nodes.

Blockchain data, especially block logs, can grow significantly over time. Since the blockchain block logs are incremental, we leverage this characteristic to generate incremental backups that only include changes, reducing storage requirements and expediting the backup process compared to creating multiple full backups.

- The node chunks the block log into segments (strides).
- Utilizes a centralized location (stride store) to store finalized historical strides.
- Each backup refers to the final strides in the stride store, eliminating the need for redundant storage.
- Then, each backup includes only the last open stride pending update, the snapshot, and other small metadata files.

This approach significantly reduces the backup size and associated storage and transmission costs. For instance, in a production environment with blocks accumulated over several years, the full block logs can easily exceed 10TB. By configuring strides to 50GB which is feasible to store the blocks for a couple of days, we reduce the single backup size from 10TB to 50GB—a reduction by a factor of 200.

2.3 Observability for maintaining system health

Observability is crucial for system health. Comprehensive monitoring systems provide real-time visibility into performance. A user-friendly management console centralizes system health information, enabling quick issue detection and resolution.

Monitoring daemons deployed with each node collect metrics and send them to Datadog Cloud Service, where alerts and dashboards facilitate system health monitoring. This setup ensures that all nodes are continuously monitored, and any anomalies are promptly addressed.

2.4 Disaster recovery for business continuity

While high availability ensures our systems can handle minor faults, a comprehensive disaster recovery strategy is essential for scenarios of significant failures, such as a region-wide outage. Here's how we handle disaster recovery:

- Regional redundancy: We maintain a redundant infrastructure in a secondary region that can take over if the primary region fails.
- Data replication: Blockchain data is continuously replicated across regions through multiple data sync links between nodes to minimize potential data loss during failover.
- Backup and restore: We implement robust backup and snapshot functionality for necessary components to facilitate rapid recovery in the event of a disaster. A highly available cloud storage service is used to store the

backups.
- Cross-cloud data replication: To handle extreme cases where the cloud storage service also fails, we replicate backups to a different cloud vendor's storage service to ensure business continuity even if one cloud provider fails.

In case of active region failure, the system seamlessly switches to the secondary region, allowing the continuity of Bullish services with potential data loss limited to blocks yet to be replicated.

3. A deep dive into producer high availability

Block producing services are critical components that must be highly available to ensure new transactions and blocks can be produced and accepted. Block producers build new blocks, and it is essential to ensure that transactions and newly produced blocks are not lost or generate conflicts within the cluster. Unlike public blockchains, Taurus, as a private blockchain setup, ensures no fork or rollback once a block is produced and committed among the producers (single block finality).

3.1 Goals

We aim to achieve two primary goals and requirements through the producer HA design:

- If any block producer node is down or block production stops, another producer node should automatically take over as the producing block producer to continue producing blocks safely. The delay should be relatively short.
- If there are conflicting blocks, one and only one will be broadcasted by the producing node and visible to the blockchain network.

3.2 Producer HA design

The system design includes adding a new plugin and modifying existing plugins for the node to work together to achieve high availability.

Producer HA plugin: The plugin is added to BPs to form a consensus group through the Raft protocol (implemented using NuRaft), to commit messages for blocks to the Raft group and reach consensus among BPs to accept the blocks. The producers work together with the producer HA plugin and its Raft cluster internally to coordinate block production.

Leader election: The Raft protocol elects a single leader, and only the leader BP can produce blocks.

- Leadership expiration: The leadership is given an expiration time to prevent overlap between two leaders within the Raft group, even in the event of network splits, so that we ensure that there is at most one leader producing blocks at any time. If the active leader is still alive before the expiration, it renews its leadership. If

the active leader fails to renew its leadership before the expiration time, it stops producing automatically.
- Leader failover: If the producing BP (leader) is down or fails to renew its leadership before its leadership expiration time (plus a graceful wait time to mitigate possible clock skews), the remaining BP nodes, if they can form a quorum (> half of the Raft group size), will elect a new leader to be the producing BP to produce blocks. If more BPs are down and the remaining BPs cannot form a quorum to elect a leader, they will retry until enough BPs join the group to form a quorum to reach consensus to elect a new leader. During this time, there is no leader and no producing BP.

Block commitment: The producing BP (leader) commits blocks produced through the Raft protocol among the BPs before adding the block to its blocklog.

- After signing a block and before including it in the blocklog, the leader BP's producer plugin first calls the producer HA plugin to broadcast and commit the new block to the Raft group to ensure the quorum of the BPs accepts the block. After the new block is confirmed by the Raft group, the new block is marked as "accepted head block."
- The net plugin and the producer plugin in the BPs in the active Raft group, upon receiving a new block, will first check a) whether the block is smaller than the current committed head block, or b) whether the new block is the "accepted head block" with the producer HA plugin. If the check fails, the net plugin and producer plugin will reject that block.
- The net plugin and producer plugin in the downstream nodeos' sync blocks as usual through the p2p network.

Independent Raft groups: We configure two independent Raft groups for the two regions for facilitating region failover.

- Each region's BPs form a Raft group.
- The Raft group has a configuration parameter is_active_raft_cluster to indicate whether it is active or not. The standby region's Raft is_active_raft_cluster is false, and no BP is allowed to produce in the standby region.
- To handle region failover, operators can activate or deactivate production in the region by changing the producer HA plugin configuration to set the is_active_raft_cluster parameter to be true.

Through the producer HA plugin, the Taurus blockchain's core block producer service is highly available across single AZ failures, and can be recovered even in disaster cases when the whole active region fails.

Summary

High availability, disaster recovery, and observability are integral to the Taurus blockchain, enabling Bullish to provide reliable, 24/7 operations. These components are continually refined to handle unprecedented situations and maintain a secure, resilient blockchain ecosystem.

For more information on the features developed and heavily used by Bullish, please check **this link**.

Feeling inspired by the cutting-edge work we're doing at Bullish? We're always on the lookout for talented individuals to join our team. **Explore our open roles** and be Bullish on your career. Want to stay up to date with the latest news from across Bullish? Follow us on **LinkedIn** and **X**.

[1]Custody solutions provided by members of the Bullish Group (1) may differ in particular jurisdictions to meet regulatory and customer requirements and expectations; and (2) will continue to evolve over time, so the information in this blogpost may become outdated. Please reach out to support@bullish.com if you would like to understand the approach currently used to protect customer assets in your jurisdiction.

ARE YOU BULLISH?

Take charge and be Bullish on your career by helping us build the best digital asset trading platform for institutions.

Work at Bullish