Large Data Infrastructure and Performance Guidance

Table of Contents

Overall Architecture

Deephaven architecture achieves scalability by separating the data ingestion functions from the query engine functions. This functional divide allows the data ingestion processes and the query engine processes to scale at different rates.

Data Access

Deephaven relies on a shared file system architecture for access to its data. Currently, the larger installations use ZFS filers to manage the storage mounts that are exported to the query and data servers via NFS over 10g network interfaces.

Intraday Data

Intraday data is any data that is considered near real-time. Depending on the data size and rate, the underlying storage should typically be high-throughput (SSD) or contain sufficient caching layers to allow fast commits.

Intraday data is written to the database via the Data Import Server (see below) or other various import services.

Dedicated Data Import Servers are used to import Intraday data to remove potential impacts on the Data Import Servers.

Historical Data

Historical data is populated by performing a merge of the Intraday data to the historical file systems. As storage needs grow, further storage can be easily added in the form of writable partitions without the need to reorganize existing directory or file structures, as Deephaven will automatically use additional historical partitions as they are added.

Intraday data is merged to Historical data on the Data Import Servers during off hours.

Example System

The following criteria demonstrate an example Deephaven installation. However, additional capabilities are possible with additional hardware resources.

Data Size

- Near-real-time data ingestion: 2TB per day
- Other data imports: 3TB per day
- Online historical data: 336TB (approximately 2.3PB uncompressed)

Query Usage

- Active queries across multiple query servers: 1,000
- Queries update in real-time as data becomes available
- Sizes range from small (<1GB of dedicated RAM) to large (>100GB of dedicated RAM)

- These power dashboard-style GUIs for about 100 users, automated trading strategies, batch-style research workloads, surveillance, and interactive queries.

Data Import Server

The Data Import Server is mainly a write-centric server with access to fast storage. Typically, this will be an SSD or volumes of SSDs, depending on the data speed and storage size required.

Hardware and Operating Systems

- 2 servers (for redundancy)
- 40 cores (hyperthreaded) @ 3.0 Ghz Intel
- 256GB RAM
- Linux - RedHat 7 or derivative
- Network: 10g mtu 8192
- Storage:
    - Root - sufficient local disk space
    - Database mount - 250GB SSD mirror local
    - Intraday data - 9GB SSD either local or iScsi
    - Historical data - 16 NFS mounts @ 35TB each

Performance Profile

- CPU
    - Idle: 90%
    - IOWait: 3%
    - User: 5%
- Memory
    - File cache: 75%
    - Process in Resident: 10%
- Intraday disks
    - 90 MB/s writes sustained

Query Servers

Query servers provide compute, memory and query access to storage. They host the memory-intensive applications.

Hardware and Operating Systems

- 20 servers
- 40 cores (hyperthreaded) @ 3.0 Ghz Intel
- 512GB RAM
- Linux - RedHat 7 or derivative
- Network: 10g mtu 8192
- Storage:
    - Root - sufficient local disk space
    - Database mount - 250GB SSD mirror local
    - Historical data - 16 NFS mounts @ 35TB each

Performance Profile

- CPU
    - Idle: 72%
    - User: 25%
- Memory
    - Processes in Resident: 34%
    - File cache: 65%

Import Servers

Import servers are similar to Data Import Servers (DIS) servers, but are less write-intensive. They are used to perform large-batch imports of data, typically overnight.

Hardware and Operating Systems

- 5 servers
- 24 cores(hyperthreaded) @ 3.0 Ghz Intel
- 192GB RAM
- Linux - RedHat 7 or derivative
- Network: 10g mtu 8192
- Storage:
    - Root - sufficient local disk space
    - Database mount - 250GB SSD mirror local
    - Intraday data - Sufficient space for the import via local, iScsi, nfs or other
    - Historical data - 16 NFS mounts @ 35TB each

Performance Profile

- CPU
    - Idle: 65%
    - User: 27%
- Memory
    - Processes in Resident: 10%
    - File cache: 40%

Scaling

Deephaven has been designed to scale horizontally:

- Greater capacity can be achieved by adding more Data Import Processes/Servers.
- Support for a larger number of queries and/or larger queries is expanded by adding more Query Servers. These can also be segregated by creating query server pools where queries may be run by users.
- The ability to scale the volume of data being ingested and/or the query volume is predicated on the underlying storage system. Therefore, having a solid scaling strategy to address IO bottlenecks should be an important part of any Iris installation.