VAST Database Connector for Trino

High-Performance Analysis Interface

Jason Russler
Technical Director, VAST

First: Who is VAST?

Who?

- Software company founded in 2016
- Out of stealth in 2019
- Large-scale, high-performance, cost-effective, all-flash storage infrastructure (pick all three!)
- Core technologies that make VAST possible:
  - VAST software
  - High-speed (100/400Gb) commodity networks w/ RDMA (or Infiniband)
  - Storage-class memory (SCM)
  - Read-intensive media (QLC flash)
- Together, these technologies have been very successful for VAST

About VAST Data

The Fastest Growing Enterprise SW Company In Infrastructure History

$100M+
Annual Recurring Revenue

Elite “Centaur” Status
2.5x Y/Y Growth

$9.1B
Series D | April 2021

The Most Valuable Private Enterprise Software Infrastructure Company In History

100%
Recommended

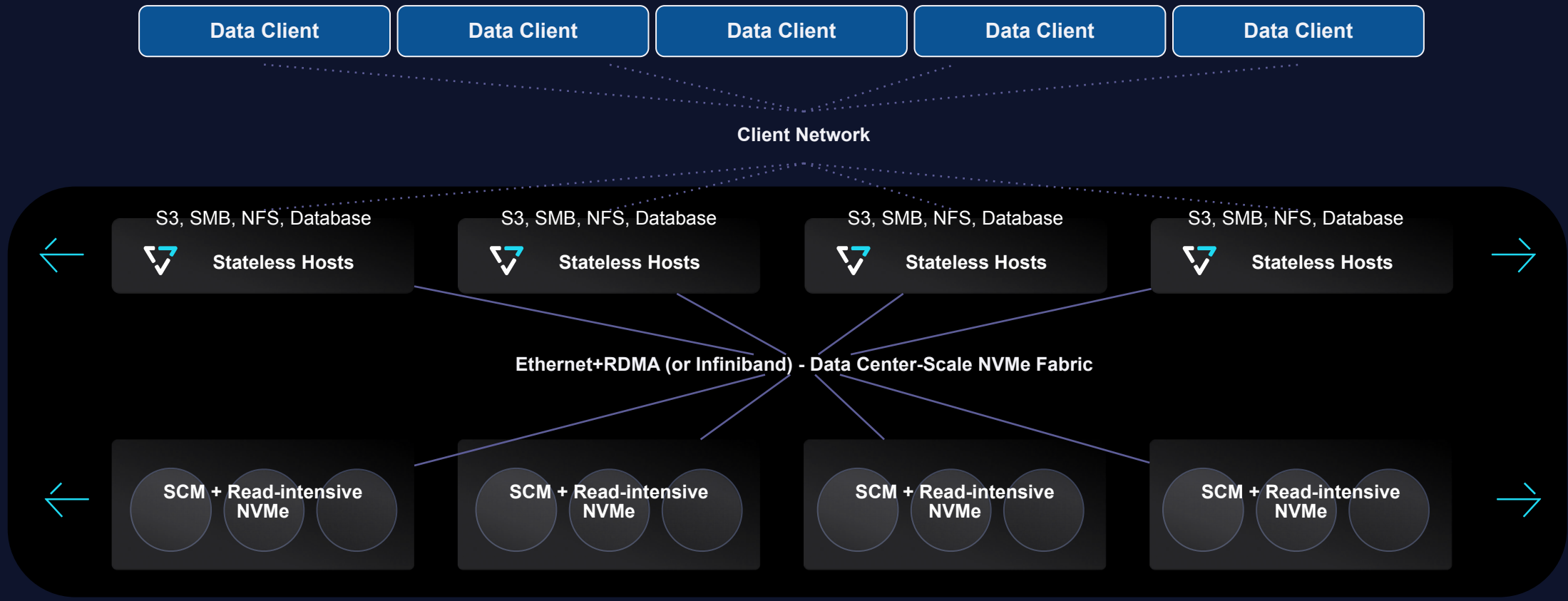
Gartner peerinsights
Source: VAST Customer Surveys

Cash Flow Positive Last Three Years

Gartner	Gartner	Forbes	Deloitte.	FAST COMPANY	CRN
#1 For Analytics	**Challenger**	**#16 Out Of 500**	**#5 Fastest Growing**	**#10 Most Innovative**	**Emerging Vendor**
Critical Capabilities Report	DFS&OS Magic Quadrant™	America’s Best Start-ups	Technology Companies	For Data Science	CRN Channel Awards

Hardware Infrastructure Architecture

In-datacenter hardware/software data solution



Second: What is the VAST Database?

Tabular format for distributed storage

VAST Database is:

- Tabular/structured data presentation
- On all-flash infrastructure
- Hybrid row/columnar (row group) storage format
- For ***transacting on*** and ***analyzing*** structured data (ACID compliant)
- REST API for access (NoSQL-type requests)
- Implements filters, column functions, indices, projections, etc.

Benefits:

- Efficient use of compute/storage resources when storing data as well as when scanning/filtering
- Scalable (and very large-scale)
- Automated maintenance and optimization

What is the point?

Performance

- Fast as hell: warehouse performance and *function* at data lake scale
- Profoundly reduced infrastructure footprint

Simplicity

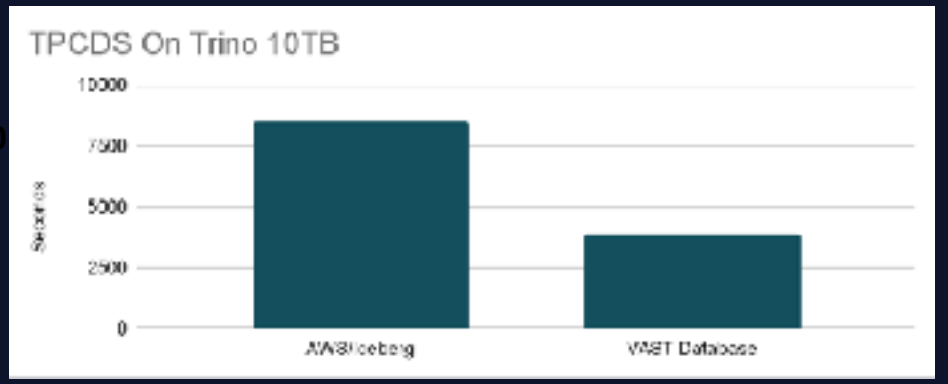
- Easy to configure and tune
- Transactional (OLTP)
- No maintenance operations

8501

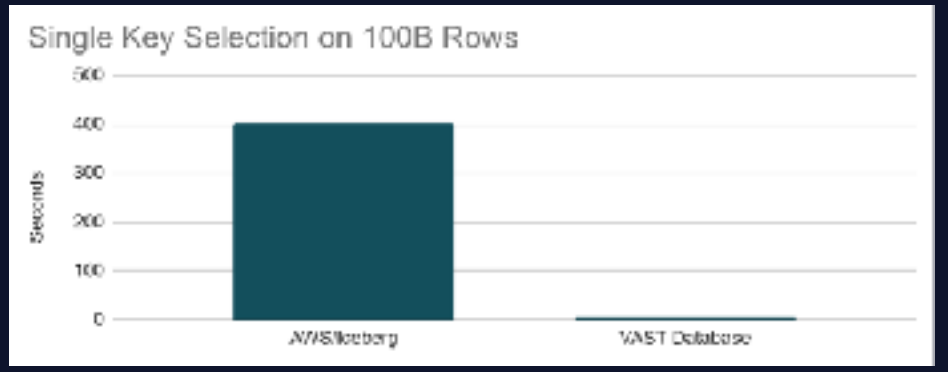
8501

TPCDS + Trino + VAST vs. Iceberg on AWS

850



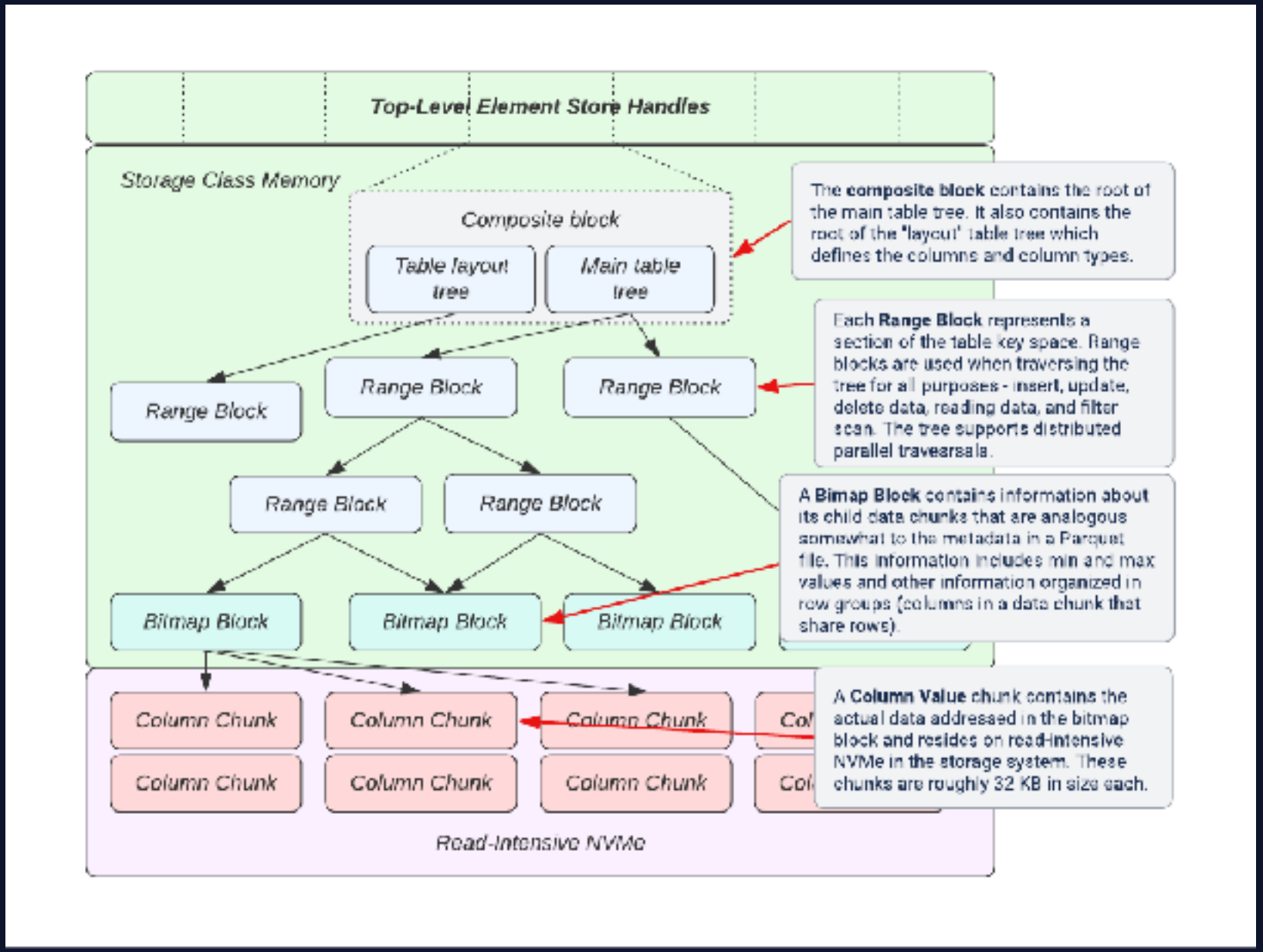
Selection/scan 75 times faster than Iceberg on unordered data with equivalent hardware



Closer Look

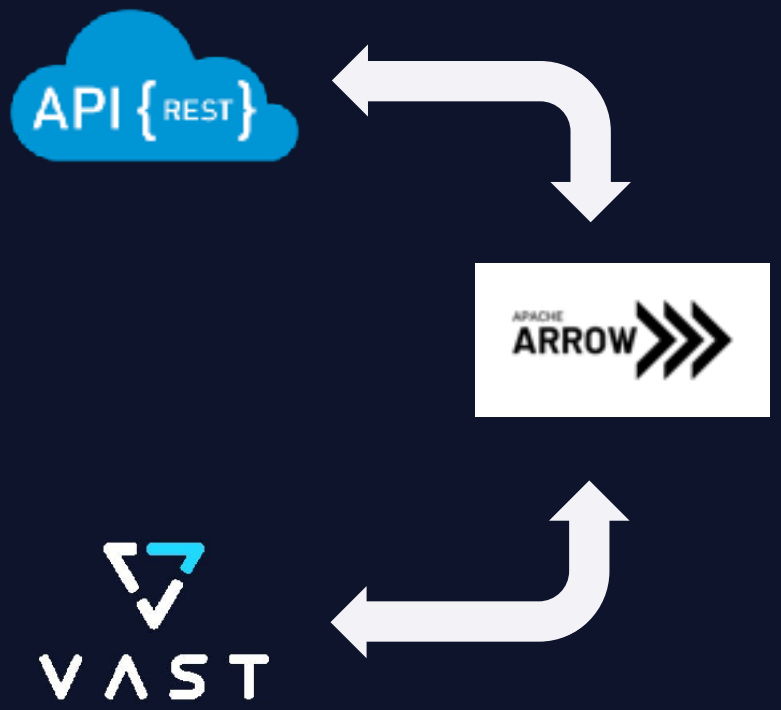
Logical layout in Storage

- Keyspace data structures in SCM
- Table and column metadata in SCM
- Data blocks stored in NVMe



How do you use it?

Tabular format on distributed storage



The VAST database is interfaced via a REST API:

- Built on the AWS v4 signature system
- Apache Arrow interchange format
- Intended for parallel use
- No separate metadata housing - all schema, table definition, statistics, etc. are stored natively in the database.

Execution engines

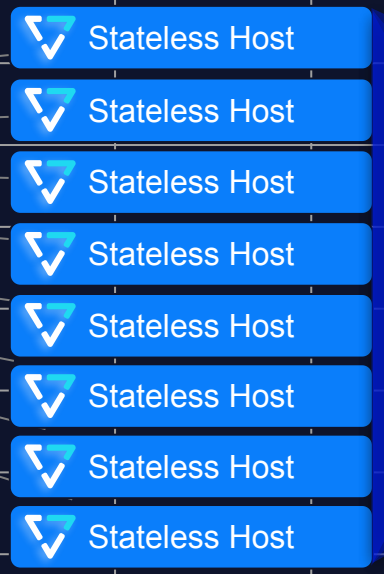
- Trino (first engine!)
- Others (but who cares?)

Accessing the VAST Database

Architecture

REST API for interacting with tabular data:

API Requests Push-downs:
projection, predicate, dereference & column expression



VAST Catalog Anatomy

Catalog configuration

- **endpoint** is used for metadata access as well as data access if **data_endpoints** is not defined.
- **region**, **access_key_id** and **secret_access_key** are self-explanatory
- **num_of_splits** effectively determines the per-node parallelism within Trino and is consistent with Trino nomenclature
- **num_of_subsplits** determines further division of API tasks on the VAST workers.
- **data_endpoints** allows the VAST driver to load-balance API requests across the VAST VIP pool and can be used in lieu of HTTP load-balancing elsewhere in the network path.

```
connector.name=vast
endpoint=http://vast.rest.endpoint:80
region=us-east-1
access_key_id={{AWS_V4_ACCESS_KEY}}
secret_access_key={{AWS_V4_SECRET_ACCESS_KEY}}

# Preliminary tuning parameters
num_of_splits=64
num_of_subsplits=10

vast.http-client.request-timeout=60m
vast.http-client.idle-timeout=60m

# Hard-code the IPs if “endpoint” is not load-balanced
data_endpoints=http://172.29.70.1,http://172.29.70.2...
```

https://github.com/vast-data/vast-db-connectors

Thank You!