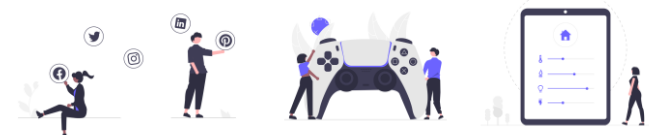


Garnet

Speaker: Badrish Chandramouli (badrishc@microsoft.com)

- Cache-stores are now more popular than ever
 - Get, Set + expiration, transactions, objects, count sketches, etc.
 - Sharded and replicated deployments are common
- Redis: **very popular API** with huge developer ecosystem
 - But, suboptimal system design, huge operational costs, weak DB features
 - Rough calculations showed that there is a huge gap compared to bare metal
- Our goal: build a **new high-performance cache-store server**
 - Cross-platform, extensible, **compatible with Redis API**
 - Provide better p99 latency, throughput, scalability, recoverability
 - Can translate to cost savings for large services
- Garnet is the resulting cache-store system
 - Up to 100X higher throughput, 4X lower latency at high percentiles
 - Better scalability and extensibility
 - Stronger DB features
- Real-World Impact
 - Azure control plane: large-scale **cache, range, graph** stores
 - Microsoft AI Platform
 - Azure CosmosDB Garnet Cache service (preview)
 - **Active OSS: <https://github.com/microsoft/garnet> (11.8K stars)**



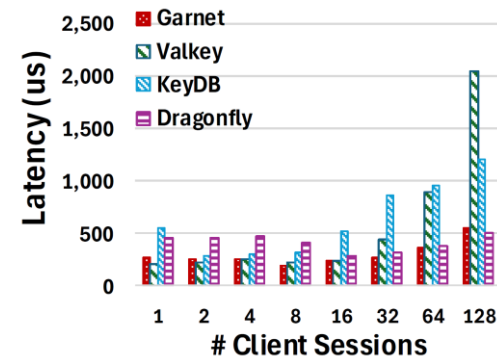
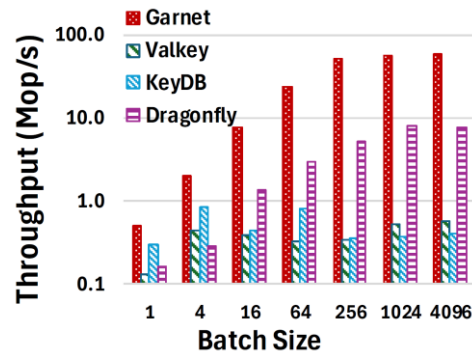
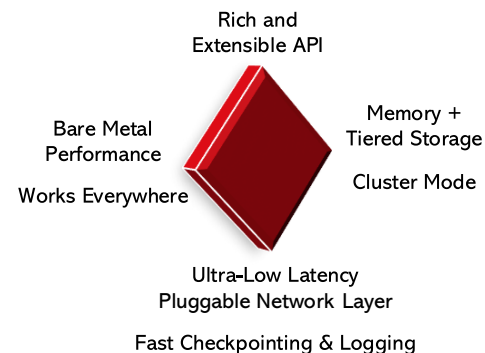
Social Media

- Session storage
- Content caches
- Job queues
- Stream processing

Gaming

- Real-time analytics
- Leaderboards
- Recommendations
- Distributed locks

Analytics



The Technology

Garnet's design is based on MSR research into { network, storage, cluster }

• Network

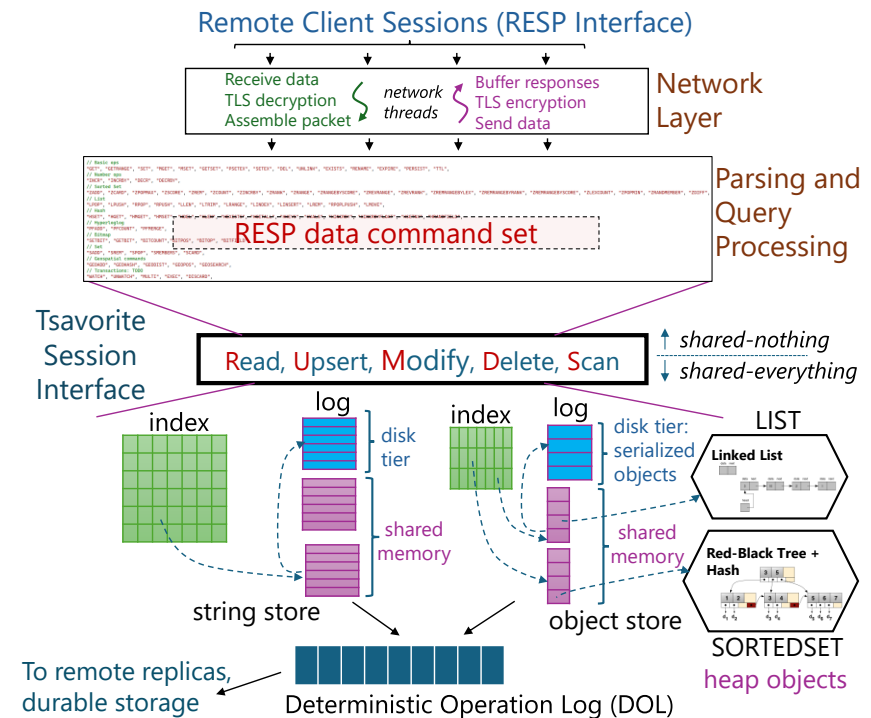
- Pluggable network layer **directly accesses** shared memory storage
- Key idea: avoid thread switching all the way, from request to response

• Storage: **Tsavorite**

- A new shared-memory KV store designed for remote data structures
 - Hash index + bucket-locked hash chains on two log-structured stores (strings, objects)
- **Revivification**: reuses space in the mutable region of log-structured stores
- Common KV schema: [key, value, expiration*, etag*] (* = optional field)
- **Narrow-waist** storage API maps **large 300+ API** Redis surface
 - Read, Upsert, Atomic Read-Modify-Write, Delete, Scan
 - Callbacks define state machine for state evolution. Example: Modify
`NeedInitialUpdate -> InitialUpdater -> InPlaceUpdater -> NeedCopyUpdate -> CopyUpdater`
- Multi-key (pessimistic) transactions within shard
- Durability: Write ahead logging, consistent background snapshots trim log

• Cluster

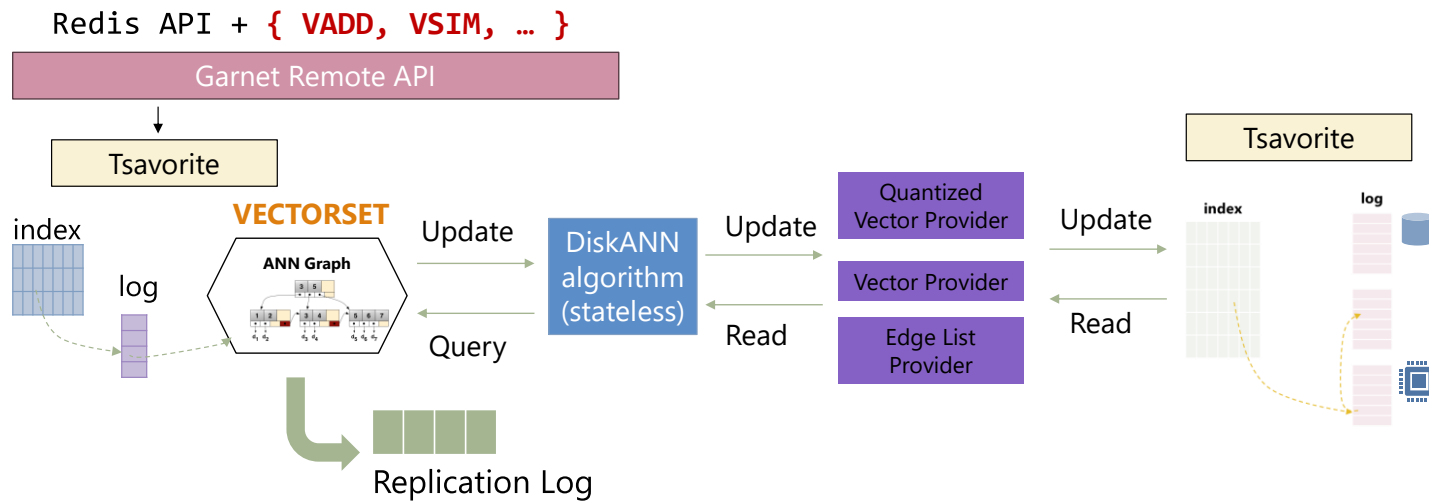
- Sharding, replication, migration based on a **passive** control plane model
- Garnet cluster can be hosted with any control plane (e.g., Kubernetes)



The Expansion

• Vector Database

- Garnet's storage engine (Tsavorite) is highly optimized for concurrent point lookups and updates
 - Capable of 100+ million ops/second per modern scale-up machine
- This is the exact access pattern for approximate nearest neighbor search (ANNS) indexing & retrieval algorithms!
- Garnet already supports “data types” as values (e.g., sorted set): added new vector set data type



• Range Index

- Adding a new data type (Range Index): { RI.GET, RI.SET, RI.SCAN, ... }
- We are integrating Bf-Tree for RI: <https://github.com/microsoft/bf-tree>
- Just another data type, but objects can span main memory and disk
- Adding non-blocking prefix consistent checkpoints to Bf-Tree

