

Messaging, Storage, or both

Two sides of the same coin

Sanjeev Kulkarni



Messaging vs Storage

- Messaging
 - Propagating future data: waiting for new messages to arrive
 - Low Latency Message dispatch and consumption
- Storage
 - Storing past (historic) data
 - High throughput query processing
 - Make sure data is durably and efficiently stored and zero data loss

Messaging / Storage

They talk about the same data
(Slightly Different Context)

What if you've got

- A fast durable, distributed pub/sub messaging
 - Flexible Traditional Messaging: Queuing and Pub/Sub
 - Focus on low latency message dispatch and consumption
- Backed by a scalable log store
 - durable message store with replication, replica recovery
 - Taking care of efficient encodings for high throughput
 - Is independently and linearly scalable.

Apache Pulsar

- A fast durable, distributed pub/sub messaging
 - Flexible Traditional Messaging: Queuing and Pub/Sub
 - Focus on message dispatch and consumption
 - Remove data as soon as possible if they are not needed.

Apache BookKeeper

- A replicated log storage
 - Low-latency durable writes
 - Simple repeatable read consistency
 - Highly available
 - Store many logs per node
 - Linearly scalable



Unified Solution

- Single system having
 - The ability to process the past
 - The ability to process the future
 - The ability to keep intermediate states for future queries
- Resulting in
 - Ability to specify computation in one unified way
 - System taking care of the context(real-time vs historical)