NEWISYS

Flash Memory Summit

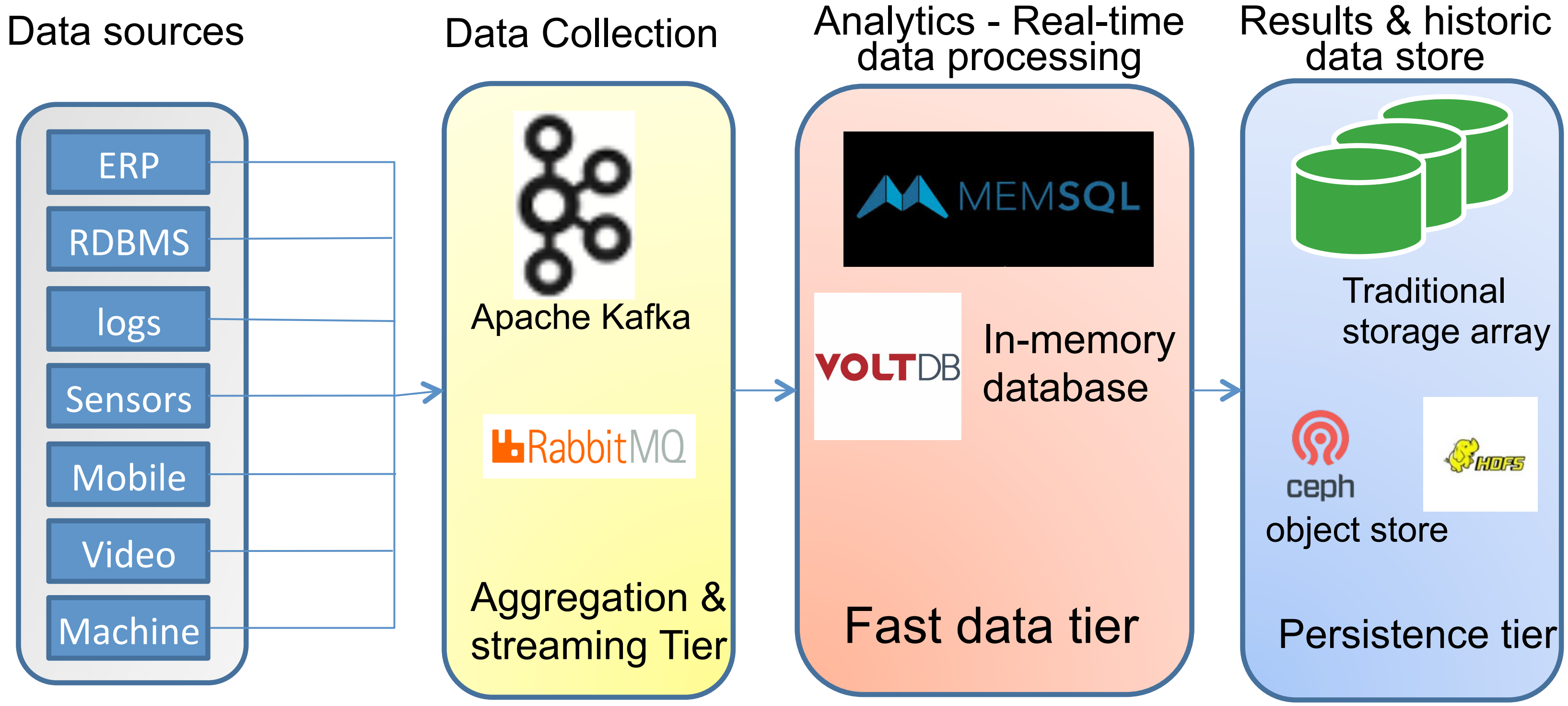
SANMINA

Scalable Big Data Pipeline over Shared NVMe

Communications • Computing & Storage • Medical Systems • Defense & Aerospace • Multimedia • Clean Technology • Industrial • Automotive

August 9th, 2018

Big Data Analytics Architecture



Data sources

- ERP
- RDBMS
- logs
- Sensors
- Mobile
- Video
- Machine

Data Collection

Apache Kafka

Aggregation & streaming Tier

Analytics - Real-time data processing

In-memory database

Fast data tier

Results & historic data store

Traditional storage array

object store

Persistence tier

Fast Data Tier

- Needs to quickly ingest and process large amounts of data
- Needs to make decisions and respond to queries based on large amounts of data from
  - Incoming streams
  - historic data and prior analysis results
- Aging data is less valuable

→ Analytics cannot be I/O bound

→ Typically uses in-memory databases

Fast Data Tier

- Needs to quickly ingest and process large amounts of data
- Needs to make decisions and respond to queries based on large amounts of data from
  - Incoming streams
  - historic data and prior analysis results
- Aging data is less valuable

➔ Analytics cannot be I/O bound

➔Typically uses in-memory databases

Scaling the In-Memory DB

- Approach 1: Buy more
  - More RAM to fit the data
  - Higher-end servers: motherboards and CPUs that can support more DIMMS & memory channels.

➔ Could be costly

➔ Still limited

Scaling the In-Memory DB [2]

- Approach 2: Scale horizontally
  - Add more servers and Distribute the DB into multiple shards
  - Each shard fits in the hosting node's memory

→ It works! Overcomes the single node's memory limit

→ Programmatic and operational complexity overhead

- → Asymmetric behavior intra vs inter-shards
- → Need to re-balance

→Cost Inefficient/Wasteful

- → CPU usage under 20%/node. Gets worst as we scale
- → less than linear scaling: hot spots end up replicated on all nodes

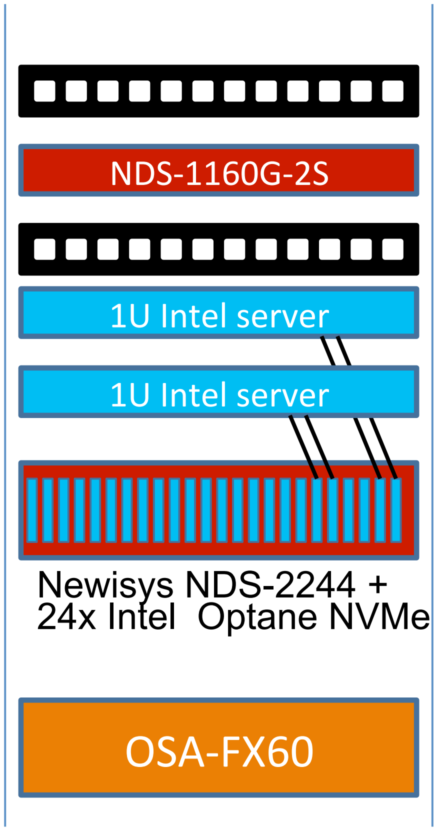
Overcoming the memory limitation

- We need to scale memory independently
  - We can already do this today with storage
- Use storage as memory
  - Need memory-grade storage ➔ Low latency NVMe
  - Need a flexibility of access and efficiency of re-use of external NVMe
  - Deterministic behavior
    - Low latency from host to non volatile memory
    - Limited jitter

Implementation Expample

Key enabling technologies:

- NVMe JBOF
- < 8 us latency NVMe disks
- IMDT/ScaleMP



Data Ingest Switch

Apache Kafka © tier
Pikes-Peak"
2x8 NVMe

Data Pipeline Switch
(40/100Gbps Mellanox)

VoltDB In-Memory DB Cluster
1TB of effective memory
2x (64MB DDR4 + 2x 375GB Optane NVMe)

Memory Extension via Intel IMDT

Newisys NDS-2244 +
24x Intel Optane NVMe

Persistence Ttier
Newisys/MPStor All-Flash Array

Results & Conclusion

- Small performance impact – around 15%
  - YCSB benchmark against the In-memory database shows
    - 10% slower on a 50-50 read/update workload
    - 19% slower on 100% read workload

- Reasonable cost. Close to 50% the total cost of all DDR solution.

What we make, makes a difference™

SANMINA