Instantly Transmit & Store Massive Amounts of Pub/Sub Market Data with Big Memory
The Publish-Subscribe (Pub/Sub) Messaging Pattern

The publish-subscribe messaging pattern, or pub/sub messaging, provides asynchronous service-to-service communication. With pub/sub middleware, senders categorize published messages into classes without knowledge of which subscribers. Similarly, subscribers only receive messages that are of interest. Pub/sub messaging is used to power event-driven analytics, or to decouple pub/sub functions from core applications for higher performance, reliability and scalability.

Demand for Real-Time Market Data is Outpacing Infrastructure

Applications ranging from News, to Risk Analysis, to Trading Engines rely on streaming real-time market data from a network of exchanges. Pub-Sub middleware is used to quickly and efficiently deliver this data to subscribers.

Pub/sub scales well for a small number of publisher and subscriber nodes and for low message volume. However, as the number of nodes and messages grows, instability emerges, limiting the scalability of a pub/sub networks. Instability at large scale is commonly caused by load surges. Another source of instability is slowdowns where many applications using the system result in the message volume flow to an individual subscriber to slow.

As user and application demand for real-time market data accelerates, the IT industry is developing technology that will allow Pub/Sub to scale. That means addressing the cost and performance of Pub/Sub infrastructure for data ingest, compute, storage and transport.

Big Memory consisting of Intel Optane DC Persistent Memory and MemVerge Memory Machine software play a key role in allowing Pub/Sub networks to keep pace with growth by scaling massively.
SOLUTION

Big Memory

Defining Big Memory
Big Memory is a class of computing where the new normal is mission-critical applications and data living in byte-addressable, and much lower cost, persistent memory.

It has all the ingredients needed to handle the growth of IMDB blast zones by accelerating crash recovery. Big Memory can scale-out massively in a cluster and is protected by a new class of memory data services that provide snapshots, replication and lightning fast recovery.

The Foundation is Intel Optane DC Persistent Memory
The Big Memory market is only possible if lower cost persistent memory is pervasive. To that end, IDC forecasts revenue for persistent memory to grow at an explosive compound annual growth rate of 248% from 2019 to 2023.

MemVerge Software is the Virtualization Layer
Wide deployment in business-critical tier-1 applications is only possible if a virtualization layer emerges to deliver HPC-class low latency and enterprise-class data protection. To that end, MemVerge pioneered Memory Machine™ software.
HOW IT WORKS

Real-Time Queues

Shared & Persistent Memory Meets Low Latency RDMA

MemVerge Memory Machine software virtualizes DRAM and persistent memory on single server and/or multiple servers in a cluster.

The MemVerge Publish/Subscribe messaging API provides MemVerge customers with the tools to create a Pub/Sub system to transmit and store massive amounts of time sequence data such as stock exchange trading ticks.

Built for Remote Direct Memory Access (RDMA) and Intel Optane DC Persistent Memory Modules, the MemVerge API creates a persistent and cost-effective memory space for massive message queues.

A Pub/Sub system often consists of many physical servers and is structured in multiple levels. In this multi-tiered system, often one server works as a publisher. In it a process accepts messages from upstream and publishes to RDMA subscribers located on the same server, which in turn push messages to subscriber servers.

In the subscriber servers there are publisher processes (aka producers) which publish messages to its local subscribers (aka consumers or business apps).

In a MemVerge API-driven Pub/Sub system, publishers and subscribers use a shared memory space to publish and receive messages. The shared memory space is a combination of both local and remote PMEM and DRAM across RDMA infrastructure within the Pub/Sub system.

Pub/Sub Workflow with Big Memory

1. Publish locally to DRAM ring buffer
2. Persist message to local PMEM
3. Local subscriber receives the message
4. RDMA subscriber receives the message
5. RDMA subscriber writes the message to remote DRAM buffer
6. RDMA subscriber writes metadata to RDMA Publisher
7. RDMA publisher writes metadata to the DRAM buffer
8. Persist the message to PMEM on subscriber server
9. Remote subscriber (consumer) receives the message
RESULTS

Single Server

Single Publisher server publishes market trading ticks to 80+ subscriber via RDMA.

Local publisher publishes to ring buffer hosted in DRAM.

Each consumer thread will busy poll the Ring buffer to get published ticks.

A scanner process polls the ring buffer to persist the messages onto PMEM for fast recovery.

![Graph showing single server performance](image-url)
RESULTS

Multi-Server

Two servers are connected via RDMA over 100Gb/s Ethernet. Each server is set up the same way as the single server setup on the previous page.

To measure the RDMA latency:
- Publisher publishes a message to remote subscriber server (trip-1)
- Subscriber/Producer then publishes the message to the consumer in the publisher server (trip-2)
- The message carries the time stamp when it is originated
- Round trip latency is used to calculate latency over RDMA
- One RDMA subscriber/Producer on each subscriber server

Eight Nodes

`mv_pubsub 1-8 Node 100K/s Performance`  
RoCE, 1GB DRAM Buffer, 600GB PMEM Buffer, 180s Run

<table>
<thead>
<tr>
<th>Subscribers</th>
<th>50B Msg Avg</th>
<th>128B Msg Avg</th>
<th>256B Msg Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Latency (us)
Greater Memory Capacity & Plug-Compatible Access to Persistent Memory for ML Applications

Watch “The Art of Big and Fast Data”

Help us shape the future of Big Memory

- Join the Big Memory Community to receive news, tech tips and early access to new software releases.
- Attend Big Memory University and earn a certificate.
- Deploy a Memory Machine PoC to see the power of Big Memory for yourself.

Send me info