AMPS I/O Screams with Memory Channel Storage™ Architecture

Performance is Our Business

At 60East, our business is high-performance messaging for the most demanding enterprise applications. These applications process millions of transactional messages a second with content filtering, analytics, and queries over message state. We build AMPS (Advanced Message Processing System), our high performance messaging engine, to take advantage of every improvement in performance and reliability offered by hardware vendors. For AMPS customers, messages are money – but only if the messages are delivered reliably, within the performance guarantees for the application. To get this level of performance, AMPS pushes hardware systems to the limit.

Storage solutions are one of the traditional bottlenecks of a high performance system. Innovation in flash-memory based storage is one of the most exciting current trends in hardware. The more performance we can get out of the hardware, the more messages our customers can process each second.

60East was excited by the potential of Memory Channel Storage™ architecture (MCS), and we wanted to know how the technology would work for AMPS. 60East partnered with Diablo Technologies to evaluate AMPS performance using MCS storage as compared to other solutions, including popular PCIe-based technology.

Putting MCS to the Test

60East and Diablo Technologies evaluated MCS performance against a leading PCIe-based flash storage solution. 60East ran our standard I/O profiling tool and workload simulator for the tests. This tool uses the same high performance I/O engine in AMPS to replicate the I/O patterns that AMPS uses under heavy load. We can also adjust the mix of writes (transactions into the system) and bookmark replays (queries over the system). We ran both tests on the same server, changing the storage device for each test.

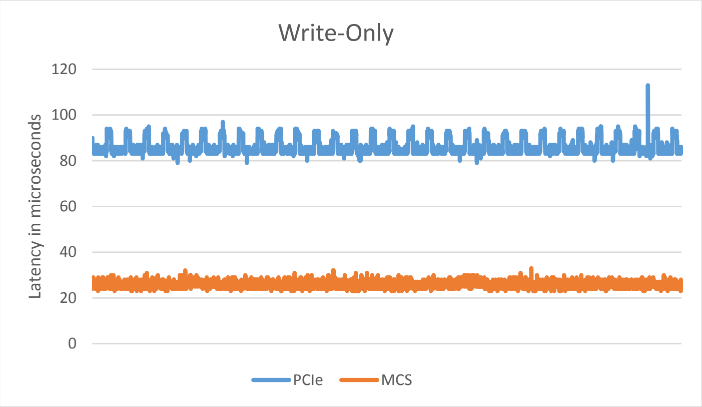
Hardware Configuration	
IBM System x3650 M4	
Processor	Intel Xeon E5-2690
Sockets	2
Cores	16 (8 per socket)
Memory	128GB
MCS	8x200GB beta units (RAID-0)

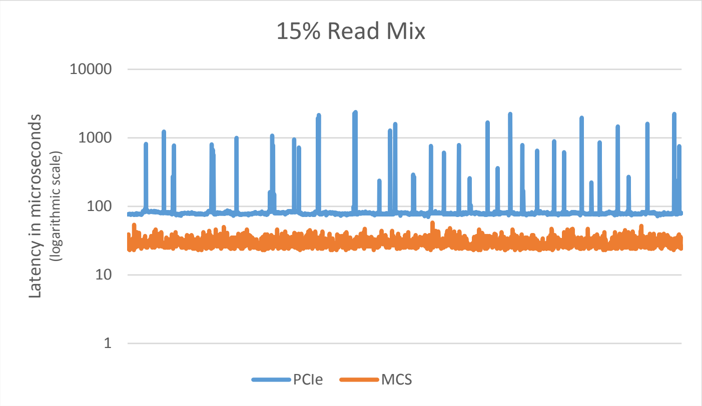
We ran tests using only writes to the system, simulating an application doing only transactional pub/sub. We then ran tests more characteristic of an application workload, which includes reads over the written data to simulate client message replay requests from the AMPS transaction log. Both tests used 512 byte messages to simulate an order processing workload.

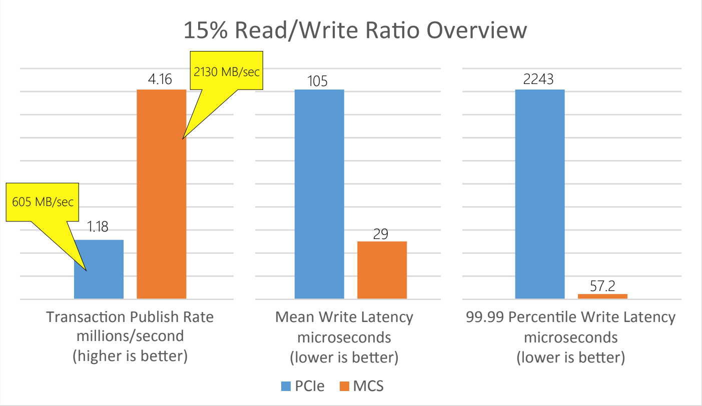
The Bottom Line

In our testing, MCS outperforms the PCIe solution with lower latency, higher bandwidth, and more consistent performance overall. The results of our testing are shown in the graphs to the right.

60East is excited by MCS. We know our demanding customers can use this performance to dramatically reduce latency in their messaging workflows. We also know that, in one page, we can only scratch the surface of our testing and our product. For more information, contact sales@crankuptheamps.com.





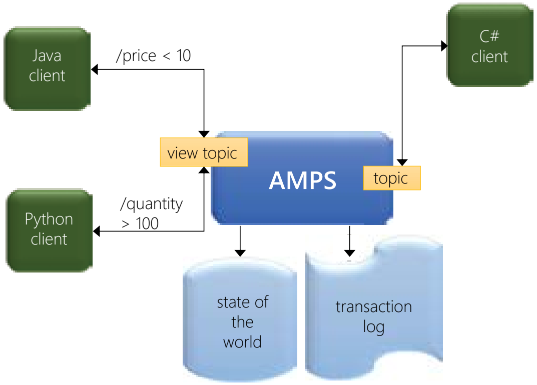


AMPS In a Nutshell

What is AMPS?

AMPS, the Advanced Message Processing System, is a reliable high-performance publish and subscribe engine designed for applications that demand the highest performance and lowest latency possible.

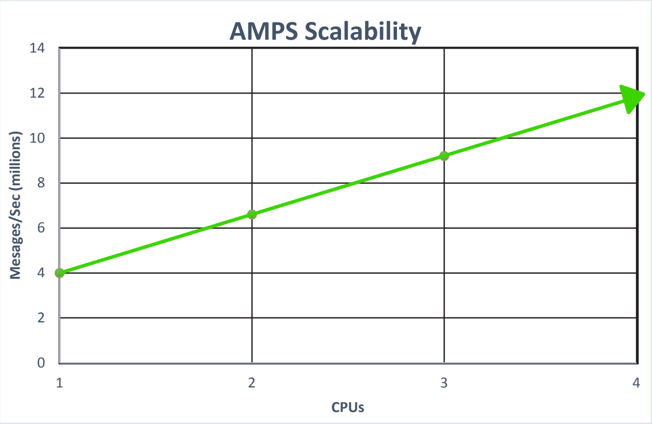
Think of AMPS as an integrated message broker, database, and analytics engine – with every part of the system optimized for both performance and ease of use.



Engineered for Performance

AMPS is optimized to unlock the full potential of cutting-edge hardware for maximum throughput. 60East takes advantage of the latest trends in hardware and cultivates close partnerships with leaders, such as Intel, as well as companies working on emerging trends in hardware, such as Diablo Technologies. AMPS includes features for high-availability, transactional consistency, replication, analytics, views over topics and content-based filtering. By including these features in the server, we make it easier for the application as a whole to perform well. AMPS doesn't just move messages or service queries; it provides all of the functionality your applications need for delivering reliable high performance services in the real world.

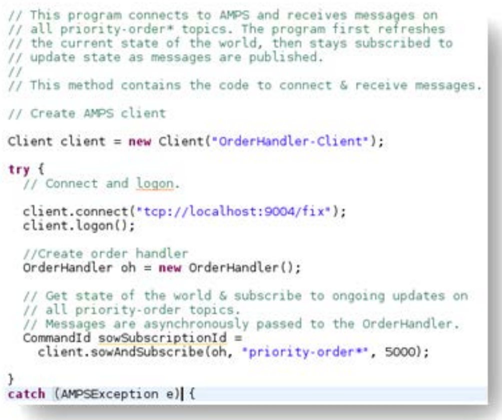
AMPS applications in production regularly process volumes on the order of millions of messages per second, with predictable latency and throughput.



Designed for Ease of Use

AMPS client libraries are designed with clarity, simplicity, and performance in mind. 60East provides libraries in Java, C#, C, C++, and Python to meet the needs of a wide variety of developers and environments. We believe in clear, declarative interfaces rather than elaborate class hierarchies. At the same time, we take advantage of modern language features and make it easy to use advanced features like high-availability, content filtering, and analytics.

60East conducts ongoing usability testing on our client libraries and documentation to simplify and clarify development with AMPS.



```
// This program connects to AMPS and receives messages on
// all priority-order* topics. The program first refreshes
// the current state of the world, then stays subscribed to
// update state as messages are published.
//
// This method contains the code to connect & receive messages.

// Create AMPS client

Client client = new Client("OrderHandler-Client");

try {
  // Connect and logon.

  client.connect("tcp://localhost:9004/fix");
  client.logon();

  //Create order handler
  OrderHandler oh = new OrderHandler();

  // Get state of the world & subscribe to ongoing updates on
  // all priority-order topics.
  // Messages are asynchronously passed to the OrderHandler.
  CommandId sowSubscriptionId =
    client.sowAndSubscribe(oh, "priority-order*", 5000);

}
catch (AMPSException e) {
```

World-Class Support

Staffed by veterans of companies such as Morgan Stanley, Bank of America/Merrill Lynch, IBM, Microsoft, and Rogue Wave Software, the 60East team provides unparalleled support with direct access to the development team. For more information, contact sales@crankuptheamps.com.