

Solace JMS Broker Delivers Highest Throughput for Persistent and Non-Persistent Delivery

Java Message Service (JMS) is a standardized messaging interface that has become a pervasive part of the IT landscape since it was introduced as part of the Java Platform, Enterprise Edition. Like all message-oriented middleware, JMS enables asynchronous, loosely-coupled communication between distributed computer systems.

JMS has become so popular, so quickly because it provides the key capabilities enterprises need to meet the needs of many applications in a standards-based platform with a well-known API that simplifies development, integration and administration.

Solace Systems, the leading supplier of hardware-based middleware and messaging solutions, provides a solution that sets new standards for JMS performance by embedding JMS routing in a purpose-built appliance.

In September of 2009 Solace conducted tests which we believe show Solace's solution to be the fastest implementation of JMS. The tests measured persistent JMS, non-persistent JMS, and even a combination of the two. This paper summarizes the results of those tests and more generally introduces the advantages and attributes of Solace's JMS solution.



Problems w/ Software Implementations of JMS

While JMS is functionally rich and meets many of the requirements of enterprise messaging, traditional software-based implementations have a few weaknesses that can restrict its usefulness and value in full-scale enterprise implementations.

Limited scale/performance

Most JMS implementations support the rate and latency requirements of a single enterprise application or service, but can't serve as a shared platform for multiple applications and services. This introduces the need to deploy multiple message buses running on many servers, along with their associated SAN connectivity, all interconnected by gateways which not only decrease performance and reliability, but increase the complexity and TCO of the system.

Persistence entails slowdown

One of the most appealing aspects of JMS is its native support for persistent messaging, whereby a copy of each message is stored to disk before being sent so in case of a system or application failure the message isn't lost. But persisting every message to disk is a time-consuming and resource-intensive process that introduces significant latency to the transaction and reduces throughput. It's also susceptible to slowdowns or failure when disconnected subscribers reconnect because the system needs to catch up by spooling out the backlog of messages (in sequence) while still satisfying current message traffic. Companies often trade reliability for performance by leaving messages in the file system cache rather than writing them to disk, accepting the risk of data loss in the event of a power failure or OS crash.

Cost and complexity of achieving high availability

Achieving high-availability of software JMS messaging systems is typically handled with additional software packages such as distributed file systems and clustering software. Each piece of software must be purchased separately and integrated by the customer, resulting in a patchwork solution that introduces problems such as slow switchovers on failure and hung file systems that require manual intervention.

Incomplete monitoring and difficult root cause analysis

The patchwork of software that surrounds and supports most software-based JMS implementations makes it difficult to isolate and address the root cause of problems that may lie in the messaging software itself, the server it runs on, the operating system, the file system or clustering software, for example.

Advantages of Solace's Solution

Scalability across LAN and WAN

With the ability to handle millions of messages a second with very low latency, Solace's solution enables JMS with performance that can meet the needs of even the most demanding applications. This lets developers use JMS in high fan-out situations that would typically require multicast. Solace's solution excels at distributing data across WANs because each router is aware of all subscriptions across the network, so it can send messages to only those routers that have relevant subscribers. In many cases these attributes can eliminate the need for IT organizations to 'mix and match' other messaging platforms into a JMS environment to meet specific requirements.

High-speed Persistence

Solace's solution can enable fully guaranteed messaging by storing messages in high-speed, redundantly-mirrored onboard RAM except for cases where offline or slow consumers can't receive messages. This means Solace's message routers can route hundreds of thousands of persistent messages per second with low consistent latency. As such, Solace eliminates the need to make the trade-off between scale/speed and persistence.

Lower TCO Through Consolidation and Simplification

When acting as a JMS broker, each Solace message router can handle a volume of messages that would require dozens of servers running a software-based broker. This dramatic reduction of servers in the data center, along with associated resources and SAN requirements, delivers significant cost savings by reducing acquisition and maintenance costs on both the software and hardware side of the equation. It also reduces the need for costly data center resources such as rack space, power and cooling, and administrative overhead.

Integrated high-availability

With Solace, high availability is an integral part of the solution at several levels. Each message router is independently fault tolerant with redundant power supplies, disks, and cooling fans. When configured as a fault-tolerant pair, two message routers are fully synchronized so that even in the event of a sudden catastrophic failure of one of the message routers, senders and receivers of messages will not be affected, and not a single message will be lost, as the other router picks up full functionality automatically within a few seconds. This fault tolerant architecture ensures that software upgrades can also be performed with virtually no service disruption by allowing one router in a redundant pair to carry the load while the other is being updated, then the reverse.

Centralized management and monitoring

Solace gives administrators a unified view of their entire messaging system. By eliminating the complexity of discrete hardware, operating system, clustering, file system and messaging software, Solace makes it easy to monitor all components and diagnose faults so they can be quickly fixed. Solace provides a GUI element manager to allow easy configuration of JMS artifacts within the JNDI which is integrated as part of the Solace message router.

Solace's hardware-based data path also means that critical monitoring statistics can be measured in real-time without impacting performance. This makes it possible to let network operators know about violations of thresholds for things like ingress or egress message rate, queue storage or disk thresholds with automatic asynchronous alerts so they can take action before it's too late.

Use Cases

Event-driven SOA

The most common implementation and understanding of service-oriented architecture (SOA) is a request/response system that routes messages using SOAP over HTTP. In many situations, however, companies want the reusability of SOA in areas that demand the real-time fanout of publish/subscribe communications. For these use cases, XML over JMS is an attractive option that is made impractical only by the performance limitations of a software-based message bus. Solace's message routers enable JMS messaging at the high volume and low latency necessary to support even the most demanding enterprise applications in an SOA environment.

Database synchronization/offloading

Most enterprises have a heterogeneous database environment thanks to best-of-breed selection, mergers and acquisitions, and the evolution that occurs over time. As a result, no one database's synchronization solution can handle sophisticated replication or distribution. These solutions typically synchronize in a batch manner using 'extract, transform and load' (ETL) methodology instead of real-time updates, so they need to use third-party change data capture software (CDC) to synchronize in real-time. Native synchronization solutions and CDC software alike are notorious for not scaling well across the WAN or in support of high fanout.

Most of these databases can export data via JMS, as can third-party CDC software, but software-based JMS simply can't keep up with the volume of inbound messages generated by source databases, nor support the fan-out necessary to synchronize target databases locally or across the WAN. Solace's hardware-based implementation of JMS can handle this traffic, giving enterprises a secure means of synchronizing distributed databases in real-time over the LAN, MAN or WAN. The existence of up-to-date sources of data in different locations lets developers give applications rapid access to the data they need and are entitled to no matter where they are physically hosted.

Grid/cloud computing

In the grid or cloud environments that enable utility computing, the coordinated assignment and execution of tasks across very large numbers of servers is critical. These computing environments simply can't scale beyond their ability to share and schedule the heavy workload that makes such systems necessary in the first place. Solace's solution supports the routing of JMS messages across grid and cloud environments at such high volume and with such low latency that the systems can handle much more volume than would otherwise be possible. Solace's solution also makes it possible for these utility computing providers to offer JMS message as a shared on-demand service.

Complex event processing

CEP applies sophisticated rules to the event stream in order to recognize patterns and trends that signify problems and opportunities, and then notifies relevant applications and people so they can take action. CEP software performs such complex operations that it's not feasible to apply CEP to high volume data streams like those common in financial services, communications services and transportation and logistics. By filtering and normalizing events for each engine, Solace enables the application of CEP to previously impossible scenarios.

Social networking and "Web 2.0"

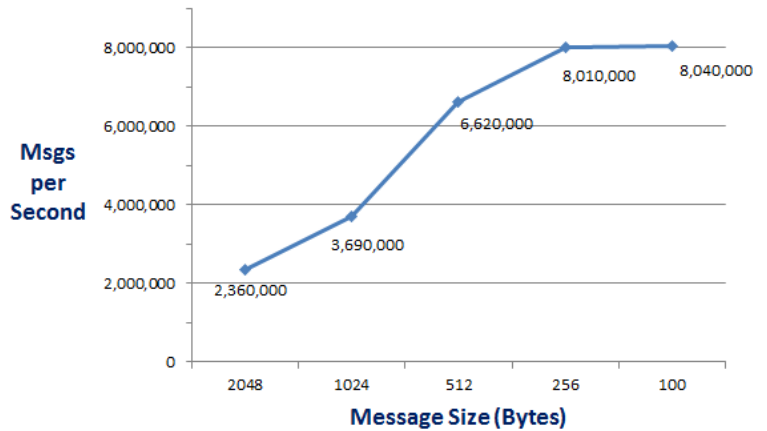
Thanks to technologies ranging from blogs and vlogs to networking sites, instant messaging, texting and Twitter, the volume of information and messages being shared between individual users and published for mass consumption is staggering. JMS is an ideal messaging platform for social networking applications because its standardized feature-rich API enables the low-overhead development that is key to the rapid creation and introduction of new services and systems. Solace's implementation of JMS supports the high, inconsistent and unpredictable volumes of traffic that new services can experience when they catch on in the market.

Performance Details

Test results show that in non-persistent mode Solace can route more than 11 million 100 byte messages per second, and when routing 1,000 byte messages, can handle more than 2 million messages per second. Tests also showed that Solace enables persistent messaging throughput of 100,000 messages per second, which is at least 5 times higher than most software solutions.

Non-Persistent JMS Performance

Each message router can process 8 million messages per second, those messages consisting of 100 byte payloads and 12 byte topics. It's important to note that for non-persistent messaging (also commonly called best effort or reliable) it's typical for a given message to be "fanned out" to many subscribers, or filtered such that it's not forwarded to any subscribers. Because of this, consuming messages and sending copies to each relevant subscribers are completely separate units of work, so we describe non-persistent message rates in terms of the total number of messages received per second plus the total number of messages sent per second. So whether Solace is processing 4 million messages in and 4 million out, or 1 million in and 7 million out, either scenario would be considered 8 million messages per second handled by the message router.



Persistent JMS Performance

For persistent messaging, most applications involve the sending of one message from one application to another for the purpose of some transaction or update.

Because of that we call each transfer of a message from the publisher, through our box, to the subscriber, 1 message per second, because it represents 1 unit of useful work. So 100,000 messages per second is how many messages pass through the content router, in and out.

When dedicated to persistent messaging, each Solace message router can process over 350,000 messages per second.

