Financial industry solutions
White paper
October 2009

WebSphere® software

Develop high-volume, low-latency finance solutions with IBM WebSphere MQ Low Latency Messaging

Contents

Overview

In today's financial world, IT organizations face tremendous pressures to automate, integrate and optimize the transaction lifecycle. The underlying messaging systems must be able to support extremely low latency and very high message throughput with reliability, high availability, flexible message delivery, fast message filtering, and effective congestion control.

This white paper discusses how IBM WebSphere® MQ Low Latency Messaging addresses the current challenges of the financial markets industry with solutions designed for reliable multicast and unicast messaging; high-performance, efficient, fine-grained message filtering; message acknowledgement to increase desired reliability; message persistence; automated failover for high availability; and monitoring and congestion control.

Understand today's IT challenges for finance messaging

The ability to handle ever-increasing message rates while reducing latency at every point in the financial transaction lifecycle is a key factor for success in today's financial world. Faced with rising competition in a global marketplace, financial organizations are aggressively investing to drive profit growth, and a "first mover" advantage is crucial.

These developments have led to the widespread use of model-driven trading and algorithmic execution, real-time portfolio and risk management, and the adoption of new technologies such as hybrid computing and stream processing. At the same time, organizations are striving to maintain, improve and dynamically monitor performance levels, even as the complexity and volume of data analysis continue to soar.

Highlights

Financial organizations are striving to maintain, improve and dynamically monitor performance levels, even as the complexity and volume of data analysis continue to soar.

In a business world like this, a difference in message response times of microseconds can mean millions gained or lost. Accordingly, IT professionals must develop innovative solutions for finance messaging that provide the following capabilities:

- **High performance** *with extremely low, sub-millisecond latency and extremely high message volumes at rates of millions of messages per second.*
- **Message delivery flexibility** *with both one-to-many and many-to-many multicast messaging and point-to-point unicast messaging.*
- **Reliable message delivery** *with fine-grained control of message delivery assurance.*
- **High availability** *to maintain system service levels and to protect the integrity of the data stream when components fail.*
- **Message persistence** *at wire speeds for message recovery and auditing.*
- **Monitoring and congestion control** *to automatically detect bottlenecks and streamline data flow.*
- **High-speed message filtering** *that supports fine-grained data multiplexing and efficient data segmentation.*

WebSphere MQ Low Latency Messaging is a transport fabric product engineered for the rigorous latency and throughput requirements typical of today's financial trading environments.

Explore the capabilities of low-latency messaging technology

To address today's IT challenges for financial markets messaging, IBM Research and IBM Software Group developed WebSphere MQ Low Latency Messaging, a transport fabric product engineered for the rigorous latency and throughput requirements typical of today's financial trading environments. The product is fully daemonless and provides peer-to-peer transport for one-to-one, one-to-many and many-to-many data exchange. It also exploits the IP multicast infrastructure to ensure scalable resource conservation and timely information distribution.

WebSphere MQ Low Latency Messaging is the newest addition to the WebSphere MQ family of messaging products. WebSphere MQ Low Latency Messaging complements existing IBM WebSphere MQ technology. Together, the products provide a single messaging infrastructure for each step in the financial transaction lifecycle—from the ingestion of market data through order execution and culminating in post-trade confirmation and settlement.

Highlights

When further combined with the IBM WebSphere DataPower® appliance, this suite of IBM technologies enables high-performance enterprise service bus (ESB) architectures for greater flexibility and integration within the enterprise.

Designed to dramatically improve throughput and reduce latency while ensuring system reliability, WebSphere MQ Low Latency Messaging can help financial services organizations enhance the responsiveness of their existing trade infrastructure while developing new solutions for emerging business opportunities.

With these benefits in mind, we will examine some of the current features and capabilities provided by WebSphere MQ Low Latency Messaging.

High performance

Throughput on WebSphere MQ Low Latency Messaging can saturate a 1 Gigabit Ethernet network, with average latency as low as 27 seconds for small message sizes typical of market data.

Several factors contribute to the high performance enabled by WebSphere MQ Low Latency Messaging. For example, a unique method of message packetization enables delay-free, high-speed data delivery. Proprietary batching technology dynamically optimizes packetization for reliable delivery and lowest latency based on throughput, message sizes, receiver, and system feedback. In addition, very compact packet headers leave more network bandwidth for application data. Throughput on WebSphere MQ Low Latency Messaging can saturate a 1 Gigabit Ethernet network, with average latency as low as 27 seconds for small message sizes typical of market data.*

WebSphere MQ Low Latency Messaging also delivers support for next-generation interconnects to enable higher throughput with lower latency, reduced latency variability, and low central processing unit (CPU) consumption. 10 Gigabit Ethernet is an emerging standard for enterprise-wide high-performance data distribution. InfiniBand is a proven, next-generation interconnect standard that offers the lowest latency and highest transmission rates. The Open Fabrics Enterprise Distribution (OFED) provides an industry set of libraries to enable remote direct memory access (RDMA) communication on these fabrics. IBM has worked closely with Mellanox and Voltaire to achieve superior results with this standard.

Highlights

Performance testing on InfiniBand has shown a maximal throughput reaching 91 million messages per second. Average latency is demonstrated at below 5 microseconds for small message sizes typical of market data at rates of up to 1 million messages per second.

With OFED 1.4 libraries and Voltaire's multiservice director-class InfiniBand switches, WebSphere MQ Low Latency Messaging has been shown to substantially reduce end-to-end application latency and maximize system throughput. The Voltaire ISR 9024D-M InfiniBand Switch Module for BladeCenter® delivers high-performance, low-latency server switching that enables IBM BladeCenter server systems to form high-performance clusters and grids for demanding computing applications and database clusters. Performance testing on InfiniBand has shown a maximal throughput reaching 91 million messages per second. Average latency is demonstrated at below 5 microseconds for small message sizes typical of market data at rates of up to 1 million messages per second.*

For the lowest latency communications between processes located in the same physical system, WebSphere MQ Low Latency Messaging can use shared memory as a transport. This functionality takes advantage of new multi-core chip technologies and co-location strategies that are becoming prevalent, achieving latency of below 1 microsecond.*

To help ensure that performance levels are maintained, WebSphere MQ Low Latency Messaging provides a comprehensive monitoring facility to verify end-to-end system performance and to quickly recognize and diagnose problems as they occur. Monitoring data is available to the application and to external systems via application programming interfaces (APIs) to statistics structures for both transmitter and receiver applications. Because the amount of monitoring may have some effect on system performance, the monitoring level is an adjustable run-time configuration option.

Highlights

WebSphere MQ Low Latency Messaging provides reliable, high-speed unicast and multicast message delivery.

Flexible message delivery

WebSphere MQ Low Latency Messaging provides a multicast transport for high-speed, one-to-many communications through User Datagram Protocol (UDP) with receiver feedback. Although typical multicast implementations offer only best-effort, unreliable message delivery, the addition of delivery options for receiver feedback enables reliable delivery with minimal loss of speed.

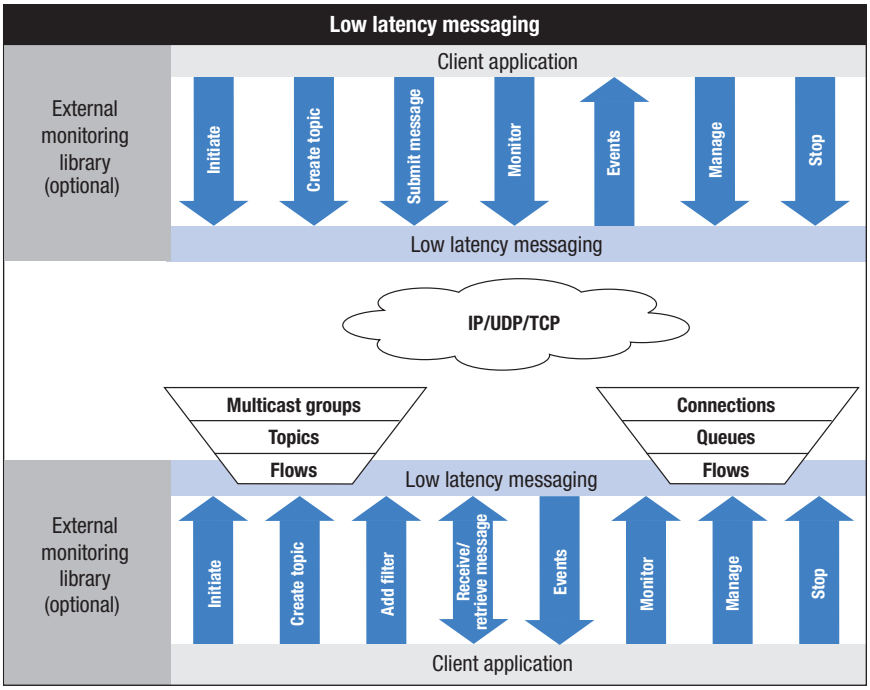

Figure 1: WebSphere MQ Low Latency Messaging interfaces with enterprise applications in either multicast or unicast mode.

WebSphere MQ Low Latency Messaging offers two transports in addition to reliable multicast. The first alternative is a lightweight, point-to-point UDP transport with positive- or negative-feedback reliability and traffic control features similar to the multicast offering. With positive acknowledgment, all packets are acknowledged, whereas negative acknowledgment provides feedback only if a packet is lost. To facilitate the design of the messaging infrastructure, one or more unicast and multicast destinations can be specified for a transmitter topic. The second alternative offers reliable, point-to-point, unicast messaging over Transmission Control Protocol over Internet Protocol (TCP/IP), in which reliability and traffic control are primarily handled by the TCP protocol. These alternatives provide the ability to deliver a stream of data reliably across a wide area network (WAN) or through a firewall, and at very high speeds.

The mapping of streams to multicast or TCP/IP connections is very flexible. For example, a multicast group or connection can be allocated per stream, or a number of streams can be sent to one multicast group or connection. It is not possible to distribute one stream's data among a number of multicast groups or connections. If several streams share one group or connection, the receivers are still able to demultiplex the data because every packet carries a stream ID in its packet transport layer header and can be effectively classified at early processing stages. This efficient packet filtering can help solve the address space problem; multicast group address space is a limited resource, and sometimes only a few group addresses are available for an application due to either administrative or technical restrictions.

Similarly, TCP connections are heavyweight objects, so the fewer used the better. Stream multiplexing allows a virtually unlimited number of separate data channels to share a few available multicast groups or connections.

Topic resolution

A topic defines a uniquely named logical message stream. New to the 2.3 release of WebSphere MQ Low Latency Messaging are facilities to perform topic resolution. This allows applications to submit minimal information about the message stream, such as the topic name, and the topic resolution services will provide the additional information (i.e., multicast groups and unicast addresses) required to successfully initialize the topic. The mapping of the topic name to the required values is provided by a set of user-defined rules, and can be used to simplify application setup.

Reliability

WebSphere MQ Low Latency Messaging has been architected to support reliability at a number of levels. A packet transport layer resides above the datagram layer, incorporating IP, UDP and TCP. For multicast communication, the packet transport layer is modified to conform to the Pragmatic General Multicast (PGM) protocol standard.

The packet transport layer helps ensure reliability through a fully developed acknowledgment mechanism. Negative acknowledgments are supported for all transports, although in unicast messaging over TCP/IP, negative acknowledgements (NAKs) are used only for stream failover due to TCP's inherent reliability. When NAKs are used, WebSphere MQ Low Latency Messaging incorporates several techniques like a sliding repair window and duplicate NAK suppression to maximize reliability with minimal protocol overhead. This level of reliability enables each client either to receive all the packets or to detect unrecoverable packet loss. Positive acknowledgement can also be used with non-TCP/IP communications to provide higher levels of reliability.

Highlights

WebSphere MQ Low Latency Messaging supports multiple options for message acknowledgement, supporting varying degrees of reliable message delivery.

Message acknowledgment control

WebSphere MQ Low Latency Messaging supports multiple options for specifying how the acknowledgement of message delivery is performed, to support varying degrees of reliable message delivery. The default form of feedback is through the use of negative acknowledgements (NAKs). This is the optimal method for throughput and simplicity, but message loss is possible if a receiver does not note a packet loss and transmit the NAK before the packet is removed from the transmitter's buffer.

A higher degree of reliability can be achieved through the use of positive acknowledgements (ACKs), which guarantees that the packet has been received and processed by the receiver. For one-to-many communications, additional levels of reliability can be specified using the unique "Wait-N ACK" feature. This allows a transmitter to specify a configurable number of ACKs that must be received before packets can be removed from the history buffer. The receiver applications can also control their sending of ACKs, including when the ACK is sent, allowing for mixed modes of reliability on a single topic and assurance that message processing is completed before acknowledgement. Transmitters can further request notification (synchronously or asynchronously) when ACKs are received for message delivery assurance. These capabilities provide the application-specified flexibility in message delivery assurance that is required for high-performance architectures that integrate many different applications.

WebSphere MQ Low Latency Messaging can handle both out-of-order packets and lost packets in the network.

Packet and message management

WebSphere MQ Low Latency Messaging can handle both out-of-order packets and lost packets in the network. To control the packet order and allow receivers to detect missing packets and request their retransmission, the transmitter sequentially numbers the packets it sends and treats the data flow as a packet stream. The streams are a fundamental concept of the packet transport layer. Each stream has a number that uniquely identifies the physical packet sequence originating at one source.

The application has an "unreliable streaming" transmission mode for real-time data and other information feeds that do not require delivery guarantees. This mode uses a "fire and forget" approach. The stream packets are sent out by a transmitter. Receivers join the multicast group that corresponds to the stream and receive the packets or listen on a specified port in the case of unicast transmission. If a number of streams use the same group, the stream ID included in each packet header is used to filter out irrelevant packets.

The reliable streaming mode uses either an ACK or NAK mechanism to recover the losses. Using ACKs, the receiver acknowledges each packet with the stream ID and the range of received packets. With the NAK mechanism, once a receiver detects a gap in the packet sequence, it can send a datagram with the stream ID and number (or range) of missing packets to the transmitter, requesting retransmission. The stream objects in the transmitter keep the sent packets in a buffer. The repair facility, which is a separate thread in the transmitter process, listens for NAKs and uses their contents to identify and resend packets.

The transmitter cannot keep the packets forever. The streaming data is virtually unlimited in size, so old data (packets) must be discarded at some point. With an ACK mechanism, packets are not discarded until they are acknowledged by the receiver. With a NAK mechanism they are discarded when the buffer is full and new packets are sent.

A message transport layer is built on top of the packet transport layer. This service is responsible for reliable message delivery, and it implements a publish/subscribe messaging model by mapping the message topics onto the packet transport streams. The service allows for symmetric data exchange, with any host being able to both transmit and receive messages in a daemonless fashion. The layer functionality incorporates a batching (burst suppression) mechanism for bandwidth-optimal delivery of small and medium messages, along with a message fragmentation/assembly mechanism for delivery of large messages.

Highlights

WebSphere MQ Low Latency Messaging provides increased reliability through wire-speed message and event persistence with retrieval capabilities, supplementing the in-memory history buffer.

Message persistence

A fundamental limit to the reliability within WebSphere MQ Low Latency Messaging is the size of the history buffer used to resend packets missed by a receiver. The Low Latency Message Store provides increased reliability through wire-speed message and event persistence with retrieval capabilities, supplementing the in-memory history buffer.

Low-latency messaging applications can utilize the Message Store to work around otherwise unrecoverable packet loss, retrieving messages from the disk store that are no longer available from the transmitter's history buffer. The Message Store can also be used to initialize a late-joining or restarted application into a given (current) state using a replay of messages from the store, minimizing impact on the actual transmitter originating the messages. The Message Store is highly configurable, allowing for filtering of the messages that are stored.

Additional levels of assured delivery can be provided by configuring the Message Store with positive acknowledgement mechanisms. Using "Wait-N ACK" and configuring the Message Store as a mandatory receiver can assure that messages are received and persisted. In addition, the Message Store can be configured to acknowledge or forward messages only after they have been written to the permanent store. Configuration options specifying the maximum amount of time that messages can be buffered before being written to the store ensure that data is persisted for low-volume message streams.

The persistence of messages can be critical to satisfy the auditing needs of an organization. The Message Store supports a number of highly available configurations to protect against component and application failure to satisfy availability requirements.

WebSphere MQ Low Latency Messaging congestion facilities help ensure that market data infrastructures continue to perform even when connected applications are overburdened.

Monitoring and congestion control

Many existing applications experience difficulty with the volume of events they must consume from today's volatile markets. WebSphere MQ Low Latency Messaging congestion facilities help ensure that the infrastructure continues to perform even when connected applications are overburdened.

Both multicast and unicast transports include methods to monitor traffic (including transmission rate, losses and retransmissions, and latency) to notify the application of network congestion problems, and to manage these detected problems by handling slow receivers or regulating the transmission rate.

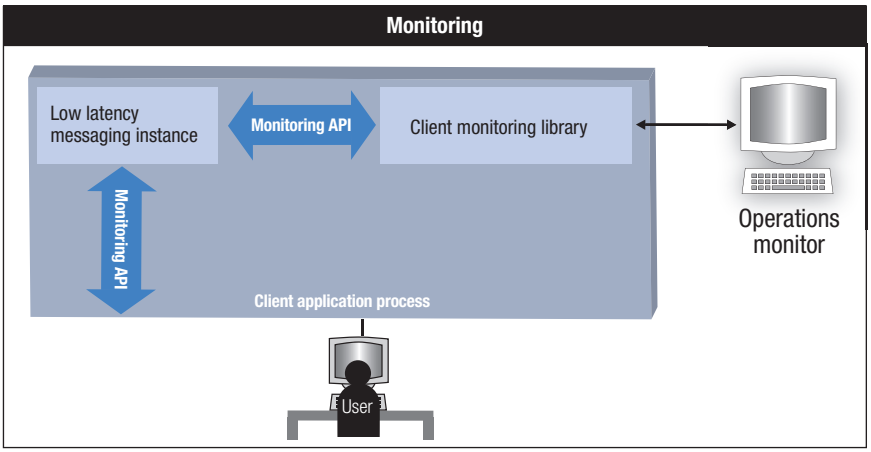

Figure 2: WebSphere MQ Low Latency Messaging includes a robust monitoring API.

WebSphere MQ Low Latency Messaging includes a robust monitoring API that provides access to aggregate and per-topic statistics involving message rates; packets and messages that are received, filtered, or lost; current receivers; NAKs processed; transmitter and receiver topic latency information; and other key data. The level of detail is configurable and ranges from basic buffer utilization information to detailed histograms of internal and external latency timings. An extensible module interface can also be used to integrate monitoring data into any external monitoring tool.

The inability to precisely synchronize timing mechanisms across machines makes the determination of end-to-end latency measurements difficult. WebSphere MQ Low Latency Messaging 2.3 provides message-based clock synchronization technology from the IBM T.J. Watson Research Center. This technology preview can be used within the product's monitoring functions for precise end-to-end latency measurements.

Highlights

Additionally, several options are available for network congestion management. By default, WebSphere MQ Low Latency Messaging does not regulate data transmission, so submitted messages are sent as fast as possible. A simple transmission *static rate limit policy*, based on the token bucket algorithm, can be activated to set the maximal rate at which a transmitter is allowed to send data. A *dynamic rate policy* is intended for situations where no receiver should be excluded, even temporarily, from the session. When the receiver set experiences difficulties and reports losses exceeding a certain level, the transmission rate is reduced until losses are below the threshold.

Per-instance limits can be implemented for the amount of memory that low-latency messaging may consume. When this amount nears exhaustion, configurable event notifications are triggered. Buffer limits can include per-topic limits on the size of transmit and receiver buffers, as well as configurable time- or space-based cleaning parameters.

ACK/NAK limits can also be implemented, with event notification thresholds set for when limits are exceeded. Slow-consumer policies can include the automatic or manual suspension or expulsion of receivers that have exceeded NAK-generation thresholds.

WebSphere MQ Low Latency Messaging enables both coarse-grained, topic-based filtering and fine-grained message filtering, making the most efficient use of network bandwidth and processing resources.

Advanced message filtering

Label-switched dynamic accumulation technology in WebSphere MQ Low Latency Messaging enables a high degree of message multiplexing and filtering—well beyond the granularity of basic multicast streams and topics. Both coarse-grained, topic-based filtering and fine-grained message filtering are available. This flexibility allows WebSphere MQ Low Latency Messaging to control the amount of data that is delivered to each application, making the most efficient use of network bandwidth and processing resources.

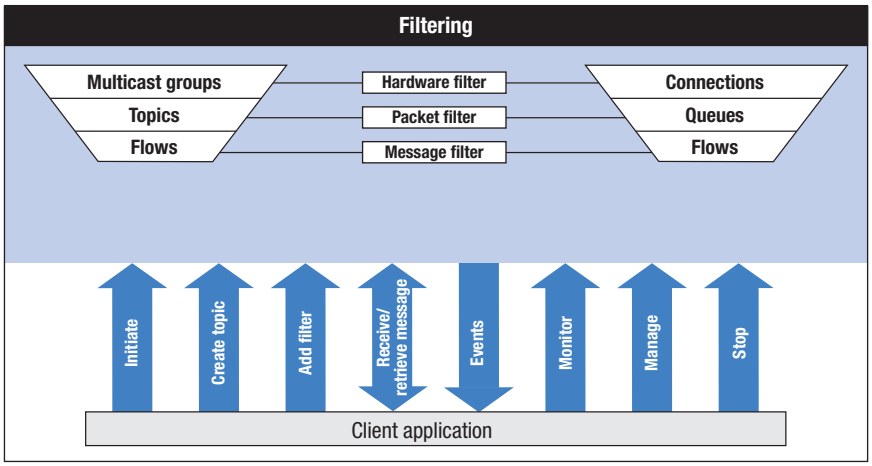

Figure 3: WebSphere MQ Low Latency Messaging offers coarse- and fine-grained filtering options.

The message stream is processed at multiple layers before delivery to the application. The data is analyzed and forwarded according to multiple parameters in different processing stack layers, such as topic and message properties and/or content in the messaging layer. Performing the analysis for each forwarded or consumed packet and message is costly, and it easily becomes the bottleneck of the system throughput. The TurboFlow technology maps the relevant data parameters to an integer or bit-map label. These labels are assigned by the transmitting application to the data chunk when it enters the processing system and are used by the receiving layers to make the routing or filtering decision based on the label(s), instead of the full parameter analysis.

WebSphere MQ Low Latency Messaging also supports a more robust message filtering mechanism similar to Java™ Message Service (JMS) message selection. This supports complex message filtering that may be required by the application. Messages can be assigned strongly typed properties (integer, double, byte, or string) by the transmitter. The receiver can define a SQL-92 expression or implement a filtering callback to select the messages it wishes to receive.

Highlights

WebSphere MQ Low Latency Messaging supports three failover options to meet the specific technology and business requirements of each organization.

Stream failover and high availability

Stream failover provides system reliability without a loss of performance—a business-critical benefit for today's organizations that need both high availability and high performance for their 24x7 applications. WebSphere MQ Low Latency Messaging supports three failover options: Reconnect, Replay or Duplex. Organizations can decide on different options based on their specific technology and business requirements.

Reconnect involves two or more nodes at the transmitter tier, potentially connected to different networks. One node is elected as primary, and other nodes are set to backup mode. WebSphere MQ Low Latency Messaging suppresses messages in backup nodes. Upon primary node failure, a backup node is activated, and data is sent from that node.

On the receiver tier, WebSphere MQ Low Latency Messaging connects to all networks, joins all relevant multicast groups, and completes all relevant unicast connections. It detects activation of a new sender, and it creates a failover event with information on failover time, messages lost, the new source identity, and the active network. WebSphere MQ Low Latency Messaging then passes the event to the application event listener. Messages are delivered from the new source to the same application message listener, making the message input failover transparent to the application component.

Decision criteria: Fast failover, but messages could be lost.

Replay is similar to the Reconnect option: Two or more nodes at the transmitter tier, potentially connected to different networks, with one node elected as primary and other nodes set to backup mode.

For Replay, WebSphere MQ Low Latency Messaging backup node accepts messages submitted by the sending application (or component) and builds history buffers as usual, but it does not send packets out. Upon primary node failure, a backup node is activated, and WebSphere MQ Low Latency Messaging starts sending packets from that node in late-join enabled mode. It receives retransmission requests for missing data and resends packets that contain the required messages.

At the receiver tier for Replay, WebSphere MQ Low Latency Messaging performs the same setup as in Reconnect (network connections, multicast groups joins) and also generates a failure event when the activation of a new sender is detected. It then passes the event to the application event listener, and it starts reception of the new data stream while detecting if messages were lost during the failover.

In addition, WebSphere MQ Low Latency Messaging sends a retransmission request to the activated backup sender of any missed packets. It also delivers messages from the new source to the same application message listener, making the failover transparent to the application (component).

Decision criteria: Failover takes longer, but no messages are lost.

Duplex involves two or more nodes at the transmitter tier, potentially connected to different networks. At least two active nodes are elected while others may function in standby backup mode. WebSphere MQ Low Latency Messaging sends messages in all active nodes and suppresses messages in the standby nodes as when in Replay mode. Upon an active node failure, the remaining active nodes are uninterrupted and continue to send messages. In addition, a standby node is promoted to become a new active node.

On the receiver tier, WebSphere MQ Low Latency Messaging connects to all networks, joins all relevant multicast groups, and completes all relevant unicast connections. It receives messages from both active sources, filters out duplicate messages, and delivers a unified message stream to the application message listener, making the duplex message input transparent to the application (component). Activation of a new sender is detected and generates a failover event with information on the new source identity and active network that is passed to the application event listener.

Highlights

Decision criteria: Immediate failover, though with extra load on network and reception resources.

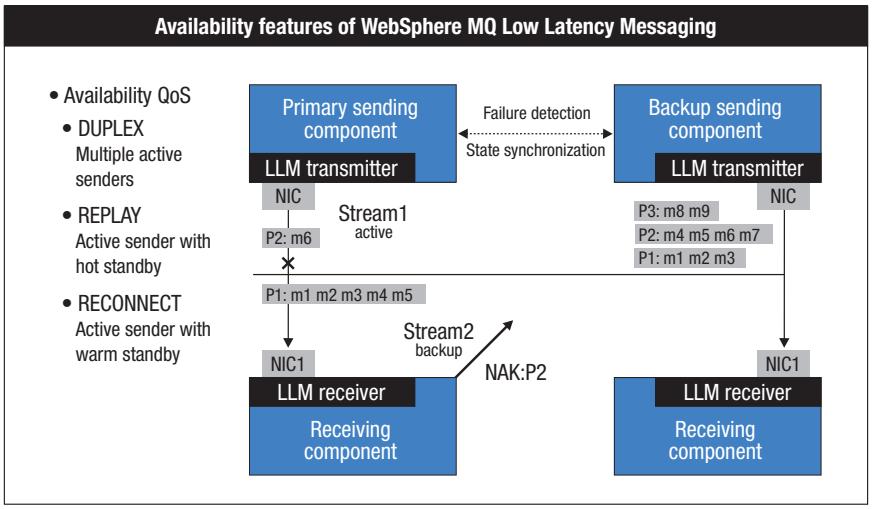

Figure 4: WebSphere MQ Low Latency Messaging handles lost of out-of-order packets and provides stream failover for system reliability.

RCMS provides a layer of high availability and consistent ordered delivery using the high-performance transport fabric offered by WebSphere MQ Low Latency Messaging.

Advanced availability capabilities

WebSphere MQ Low Latency Messaging facilitates the development of highly available transmitters and receivers using the Reliable and Consistent Message Streaming (RCMS) component. RCMS provides a layer of high availability and consistent ordered delivery using the high-performance transport fabric offered by WebSphere MQ Low Latency Messaging. RCMS utilizes Reliable Multicast Messaging (RMM), which provides high delivery performance, reliability, late joiner support, congestion and traffic control.

RCMS defines the concept of a "tier," which consists of a group of components (tier members) that are replicas of each other. Each replica executes the application's logic as if it were the only component. RCMS connects the tier members and ensures availability in the case of a failure. The application can define the level of redundancy it wants to use; with X tier members running, up to X-1 members can fail and the application will continue to function. RCMS detects component failure and migrates the data stream to a backup member without message loss.

RCMS provides facilities to perform state synchronization of tier members. The failover options available with RCMS are similar to those described above (Reconnect, Replay, and Duplex), but RCMS handles failure detection and automatic role changes with split-brain prevention policies. RCMS automatically synchronizes the state of the tier member's incoming and outgoing traffic, and helps synchronize the state of the application itself, allowing a new tier member to start full functioning in parallel with existing peers. These capabilities permit the application developer to focus on the application functionality while RCMS handles most of the complexities associated with high availability.

The total ordering feature of RCMS enforces a consistent order of message delivery from a number of independent data transmitters to multiple receivers. Total ordering assures that all receivers see exactly the same order of incoming messages. This can be critical for some applications functioning as a tier; if the processing of the messages affects the application state, total ordering can be used to guarantee that the applications maintain identical state.

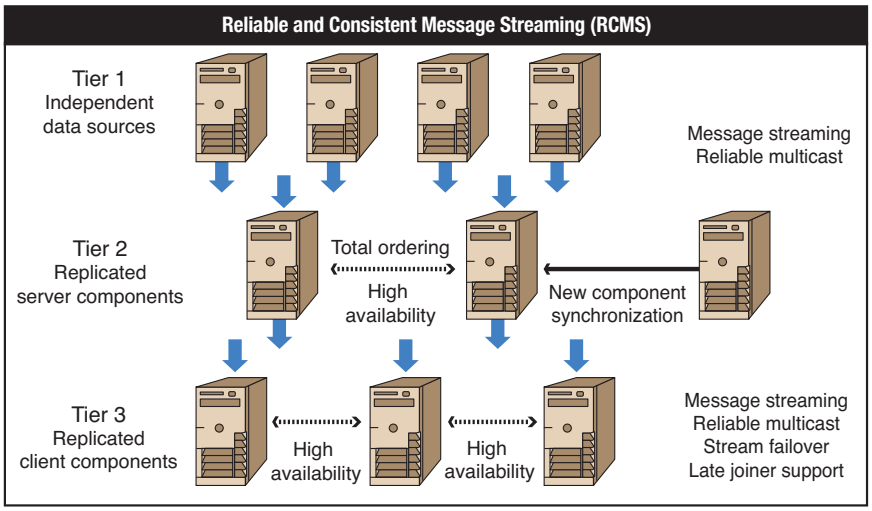

Figure 5: The RCMS component of WebSphere MQ Low Latency Messaging provides advanced high-availability features such as failure detection, component synchronization, and stream failover.

Highlights

Review the benefits of WebSphere MQ Low Latency Messaging

WebSphere MQ Low Latency Messaging can be used almost anywhere within the market data and trading lifecycle, delivering a variety of critical information that includes:

- *Market data from exchanges to market data consumers.*
- *Market and reference data within the enterprise to analytic or trading applications.*
- *Trade data such as positions or orders to direct market access and other trading applications.*
- *Event notifications for systems monitoring, risk analytics, and compliance applications.*

WebSphere MQ Low Latency Messaging provides reliable multicast and unicast messaging, high-performance and fine-grained message filtering, message persistence, and stream failover for high availability. The product supports millions of logical message flows with APIs to monitor statistics and performance. It also provides deep visibility into the status of the network, senders, and receivers. Congestion and traffic rates are controlled by the automatic detection and management of slow consumers.

WebSphere MQ Low Latency Messaging solutions are highly configurable and can be run on a large number of platforms.

In addition, WebSphere MQ Low Latency Messaging is highly configurable and can be adapted for a variety of application messaging and threading requirements. The solution runs on a large number of platforms, including Microsoft® Windows®, Linux®, zLinux, HP-UX Itanium® and various Sun Solaris platforms including SPARC and x86.

IBM has a proven heritage of providing solutions that address a wide range of needs within financial institutions, particularly in middle- and back-office environments. Now, with the addition of WebSphere MQ Low Latency Messaging, solutions can be designed specifically for the very high-performance, low-latency requirements of the finance industry. This means that financial services organizations can develop effective messaging solutions with the speed, capacity, reliability and flexibility required for success in today's financial markets.

For more information

To learn more about IBM solutions with WebSphere MQ Low Latency Messaging, contact your IBM representative or IBM Business Partner, or visit **ibm.com**/software/integration/wmq/llm

* The performance numbers listed for WebSphere MQ Low Latency Messaging are based on measurements using standard IBM benchmarks in a controlled environment. The actual throughput that any application will experience may vary depending upon considerations such as message size, transmission rate, hardware platform and network configuration. Therefore, no assurance can be given that an individual application will achieve the throughput or latency stated here. Customers should conduct their own testing. For more detailed performance information, consult your IBM sales representative.

IBM Corporation Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
October 2009

WSW14066-USEN-00