Financial markets industry solutions
White paper

WebSphere® software

Develop high-volume, low-latency financial markets solutions with IBM WebSphere MQ Low Latency Messaging.

May 2008

Contents

Overview

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

This white paper provides an overview of current messaging challenges for the financial markets industry. It also discusses how IBM WebSphere® MQ Low Latency Messaging addresses these challenges with solutions designed for reliable multicast and unicast messaging; high-performance, efficient, fine-grained message filtering; stream failover for high availability; and automated monitoring and congestion control.

Understand today's IT challenges for financial messaging

The ability to handle ever-increasing message rates while reducing latency at every point in the financial transaction life cycle 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.

In a business world like this, a difference in message response times of milliseconds can mean significant financial gains or losses. Accordingly, IT

Highlights

IT professionals in financial markets must implement innovative messaging solutions that provide high performance, message delivery flexibility, high availability, monitoring and congestion control, and high-speed message filtering

professionals in financial markets must implement innovative messaging solutions that provide the following capabilities:

- **High performance** with extremely low, submillisecond latency and extremely high message volumes at rates of millions of messages per second.
- **Message delivery flexibility** with both one- and many-to-many multicast messaging and point-to-point unicast messaging.
- **High availability** to maintain service levels and protect the integrity of the data stream.
- **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.

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 engineered for the rigorous latency and throughput requirements typical of today's financial trading environments. The transport provides 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 WebSphere MQ technology. Together, the products provide a single messaging infrastructure for each step in the financial transaction life cycle – from the ingestion of market data through order execution and culminating in post-trade confirmation and settlement.

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.

WebSphere MQ Low Latency Messaging can reach a maximal throughput of 8 million 12-byte messages per second on Gigabit Ethernet and 21 million on InfiniBand (using Cisco's DAL libraries).

Average latency has been demonstrated at 93 microseconds on Gigabit Ethernet and 30 microseconds on InfiniBand (using Voltaire's VMA libraries).

With these benefits in mind, let's look at some of the current features and capabilities provided by WebSphere MQ Low Latency Messaging.

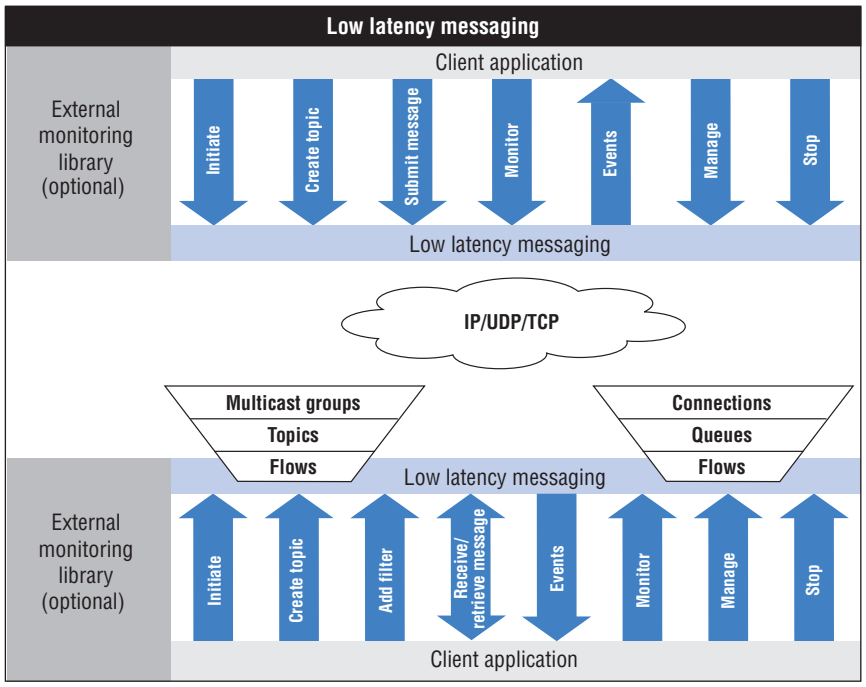
High performance

Several factors contribute to the high performance available with 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 based on throughput, message sizes, receiver and system feedback. In addition, very compact packet headers leave more network bandwidth for application data. These features allow WebSphere MQ Low Latency Messaging to reach a maximal throughput of eight million 12-byte messages per second on Gigabit Ethernet. Average latency has been demonstrated at 93 microseconds on Gigabit Ethernet for more typical market data rates of 100,000 120-byte messages per second.

WebSphere MQ Low Latency Messaging also delivers support for IP over InfiniBand to enable higher throughput with lower latency, reduced latency variability and low central processing unit (CPU) consumption. InfiniBand is a proven, next-generation interconnect standard that offers high transmission rates and scalability. IBM has worked closely with Voltaire and Cisco Systems to achieve superior results with this standard. Voltaire's VMA libraries and Cisco's DAL libraries have been shown to increase message throughput and reduce end-to-end application latency when compared to Gigabit Ethernet. For example, for 100,000 120-byte messages-per-second rates, average latency is demonstrated at 30 microseconds on InfiniBand using Voltaire's VMA libraries. Using Cisco's DAL libraries, a maximal throughput of up to 21 million messages per second for 12-byte messages has been achieved. 4X InfiniBand Switch Modules for IBM BladeCenter® from Cisco and Voltaire deliver high-performance, low-latency server switching that enables BladeCenter server systems to form high-performance clusters and grids for demanding computing applications and database clusters.

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 interface (API) 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.

Statistics are maintained for both transmitter and receiver instances and topics and represent the health of the WebSphere MQ Low Latency Messaging application. Statistics include configuration, memory and buffer utilizations; the number of bytes, packets, messages and negative acknowledgements (NAKs) processed; as well as aggregate packet, byte and NAK rates. (For more information on system monitoring, see the "Monitoring and congestion control" section on page 12.)



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

Highlights

WebSphere MQ Low Latency Messaging offers two alternative transports that provide the ability to deliver a stream of data across a WAN or through a firewall reliably, at very high speeds

Flexible message delivery

WebSphere MQ Low Latency Messaging provides a multicast transport for high-speed, one-to-many communications using the 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.

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 either 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. The second alternative offers reliable, point-to-point, unicast messaging using the TCP/IP protocol, in which reliability and traffic control are primarily handled by the TCP protocol. These alternatives provide the ability to deliver a stream of data across a wide area network (WAN) or through a firewall reliably, 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 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.

In the same way, 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.

Highlights

Reliability and high availability

WebSphere MQ Low Latency Messaging supports reliability and high availability through a variety of design strategies, features and capabilities.

ARCHITECTURAL OVERVIEW

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. NAKs are supported for all transports, although in unicast messaging over TCP/IP, 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.

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.

PACKET MANAGEMENT

WebSphere MQ Low Latency Messaging can handle both out-of-order packets or lost packets in the network. To control the packet order and allow receivers to detect missing packets and request their retransmission, the

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

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. Basically, 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 acknowledgement (ACK) mechanism or a NAK mechanism to recover the losses. With the ACK mechanism, the receiver acknowledges each packet with the stream ID and the number (or 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 the packet 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 to the arriving NAKs and uses their contents to identify and resend packets.

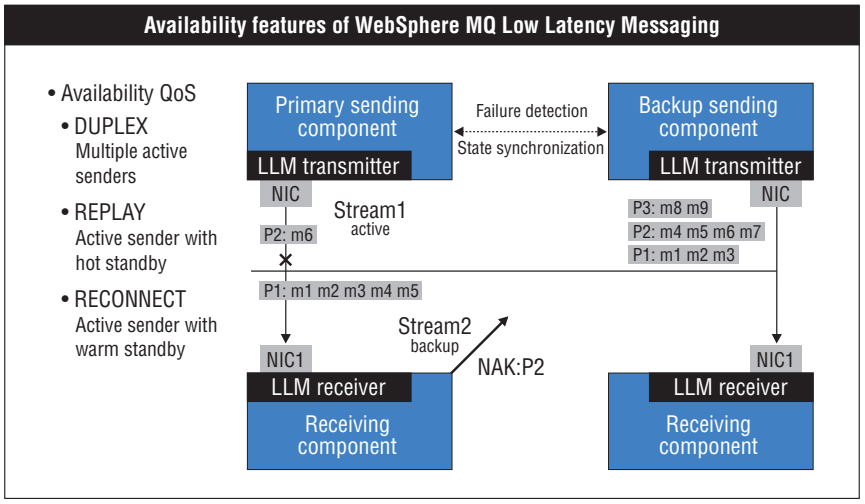
The transmitter cannot keep the packets forever. The streaming data is virtually unlimited in size. Therefore, old data (packets) must be discarded by stream objects at some point. With an ACK mechanism, packets are not discarded until they are acknowledged by the receiver. With a NAK mechanism, they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

The stream-history depth (that is, the sequential number of the last available datagram, also referred to as the trail packet number) is a basic parameter in reliable message streaming. This number changes and must immediately be

Highlights

made known to receivers, in order to detect the unrecoverable loss and suppress NAK sending.

Another important parameter is the sequential number of the last packet sent on a stream, also referred to as the front packet number. The transmitter will periodically send a small datagram (control packet) containing this number, along with other control information. This parameter is required at slow submission rates, in case the last packet/message is lost in the network. Without it, receivers cannot know they are missing the most recent data and will not request its retransmission. The packets between the trail and the front form the sliding repair window – the set of retained packets available for servicing the repair requests.



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

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

STREAM FAILOVER

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 three nodes at the transmitter tier, potentially connected to different networks. One node is elected as primary, and the remaining nodes are set to backup mode. WebSphere MQ Low Latency Messaging drops 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 concerns a scenario that is similar to the Reconnect option: two or three nodes at the transmitter tier, potentially connected to different networks, with one node elected as primary and the remaining nodes set to backup mode.

For Replay, WebSphere MQ Low Latency Messaging 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, 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 creates a failover event with information on failover time, the new source identity and the active network. 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 translates the missing message number into the packet sequence number and sends a retransmission request to the activated backup sender. It also delivers messages from the new source to the same application message listener, making the message input failover transparent to the application (component).

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

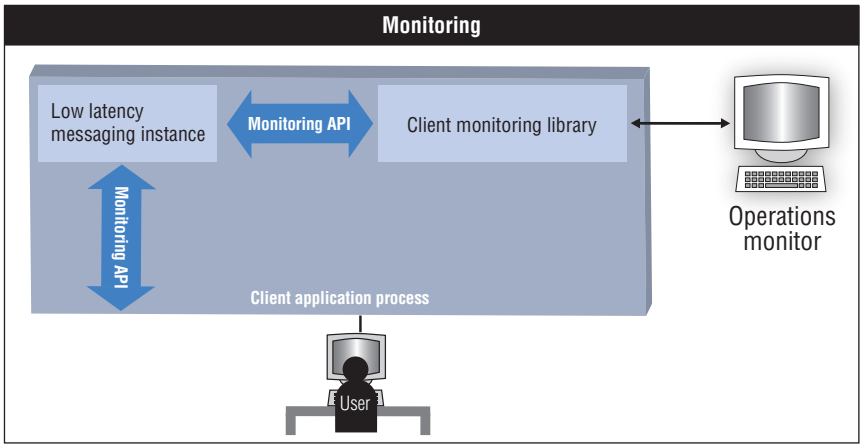
Duplex involves three nodes at the transmitter tier, potentially connected to different networks. Two active nodes are elected while the third is in standby backup mode. WebSphere MQ Low Latency Messaging sends messages in the active nodes and drops messages in the standby node. Upon an active node failure, the standby node becomes active, and WebSphere MQ Low Latency Messaging starts sending messages in this 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). At the same time, WebSphere MQ Low Latency Messaging detects activation of a new sender and creates a failover event with information on the new source identity and active network. It then passes the event to the application event listener.

Highlights

The included monitoring API provides access to aggregate and per-topic statistics involving message rates; packets and messages that are received, filtered or lost; current receivers; transmitter and receiver topic latency information; and other key data

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



WebSphere MQ Low Latency Messaging includes a robust monitoring API.

Monitoring and congestion control

Most existing applications are not necessarily prepared for the volume of events they will have to consume. 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.

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; transmitter and receiver topic latency information; and other key data. The level of detail for these statistics is configurable and ranges from basic buffer utilization information to detailed histograms of internal and external latency timings.

Additionally, several options are available for network congestion management. By default, WebSphere MQ Low Latency Messaging does not regulate data transmission. Applications submit messages to WebSphere MQ Low Latency Messaging, and the product sends them out as fast as possible.

A simple transmission *static rate limit policy*, based on the token bucket algorithm, can be activated by the application deployer/administrator 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 is near 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.

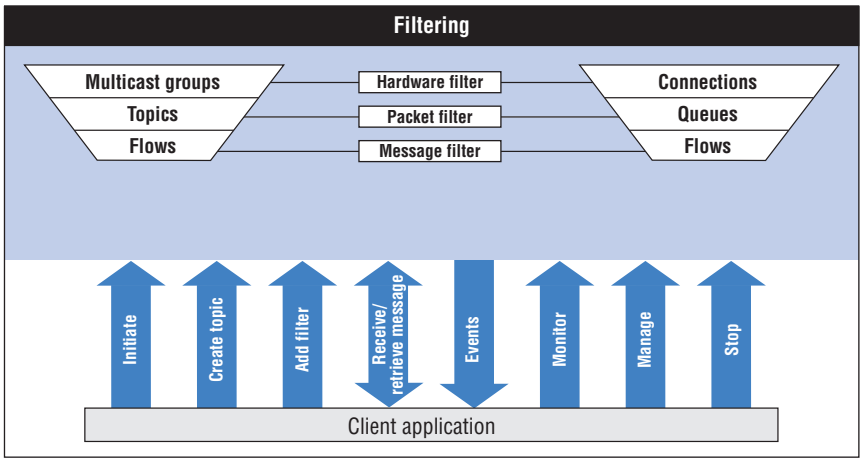
The combination of these features allows for very fine-grained monitoring and congestion management for individual topics and even specific receivers. This gives users control to implement policies specific to individual classes of data within a single application.

Advanced message filtering

Label-switched dynamic accumulation technology embedded 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, helping to make the most efficient use of network bandwidth and processing resources.

The message stream is processed by a number of nodes before it arrives at the end consumer. 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 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 offers coarse- and fine-grained filtering options.

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 life cycle, 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, and stream failover for high availability

WebSphere MQ Low Latency Messaging provides reliable multicast and unicast messaging, high-performance and fine-grained message filtering, 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.

In addition, WebSphere MQ Low Latency Messaging solutions are 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® 32 and 64, Linux® 32 and 64, and various Sun Solaris platforms including SPARC, x86, 32 and 64.

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

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

Produced in the United States of America
May 2008

TAKE BACK CONTROL WITH **WebSphere.**

SWW14001-USEN-00