Develop High-Volume, Low-Latency Solutions with Confinity Low Latency Messaging (CLLM)

Table of Contents

Executive Summary

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 reliability, high availability, flexible message delivery, fast message filtering, and effective congestion control.

Overview

This white paper discusses how Confinity Low Latency Messaging (CLLM) 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 for recovery and auditing; automated failover for high availability; and monitoring and congestion control.

Understand today's IT challenges for messaging in financial markets

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 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 this environment, a difference in message response time of microseconds can mean millions gained or lost. Accordingly, IT professionals must develop innovative solutions for financial messaging that provide the following capabilities:

- **High performance** with extremely low, sub-millisecond latency and extremely high message volumes at a rate of millions of messages per second. Efficient communications between co-located applications supported with shared-memory transport. Native support for Remote Direct Memory Access (RDMA) over InfiniBand and 10 Gigabit Ethernet (GbE).
- **Message delivery flexibility** with one-to-many and many-to-many multicast messaging, point-to-point unicast messaging, load-balanced delivery to multiple receivers, as well as WAN delivery.
- **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, durable subscriptions, and auditing.
- **Monitoring and congestion control** to report latency, enable applications to detect bottlenecks, and streamline data flow.
- **High-speed message filtering** that supports fine-grained data multiplexing and efficient data segmentation on receivers and enabling efficient network usage by transmitters.

Explore the capabilities of low-latency messaging technology

To address today's IT challenges for financial markets messaging, IBM Research laboratories developed Low Latency Messaging (LLM), a transport fabric product engineered for the rigorous latency and

throughput requirements typical of today's financial trading environments. The product is daemon-less with peer-to-peer transport for one-to-one, one-to-many, and many-to-many data exchange. It exploits the IP multicast infrastructure to ensure scalable resource conservation and timely information distribution.

Low Latency Messaging (LLM) was part of the messaging products family from IBM Corporation before it was acquired by Confinity Solutions. Confinity Low Latency Messaging, or CLLM as it is called now, provides 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. It can be combined with traditional message queueing technologies (e.g. MQ series or AMQP) or cloud-based offerings such as Kafka for greater flexibility and integration within the enterprise.

Designed to dramatically improve throughput and reduce latency while ensuring system reliability, CLLM 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 CLLM.

High performance

Several factors contribute to the high performance provided by CLLM. 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 CLLM can saturate a 10 Gigabit Ethernet network, with an average latency as low as 4.5 microseconds for small message sizes typical of market data*

CLLM 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 support is mandatory requirement for enterprise-wide high-performance data distribution. InfiniBand is a proven, state-of-the-art 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. CLLM has been developed in collaboration with Mellanox and Voltaire to achieve superior results using this standard. The product also supports high-performance communications between co-located applications using shared memory as a transport, scaling with the increasing processors and cores available on the latest hardware platforms.

With OFED 1.6 libraries and Voltaire's multiservice director-class InfiniBand switches, CLLM has substantially reduced end-to-end application latency and maximized system throughput. Performance testing on InfiniBand has shown a maximal throughput reaching 110 million messages per second. The average latency is demonstrated at below 3 microseconds for small message sizes typical of market data at rates of up to 1 million messages per second*

With OFED 1.6 libraries and Mellanox adapters with RoE (RDMA over Ethernet) support, CLLM has delivered best-in-class results for latency and throughput over 10 Gigabit Ethernet.

For the lowest latency communications between processes located in the same physical system, CLLM can use shared memory as a transport. This functionality takes advantage of multicore chip technologies and co-location strategies that are becoming prevalent, achieving latency of around 1 microsecond*.

To help ensure that performance levels are maintained, CLLM 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 for both transmitter and receiver applications. Because the amount of monitoring may have some effect on system performance, the monitoring level is an adjustable runtime configuration option.

Flexible message delivery

CLLM 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.

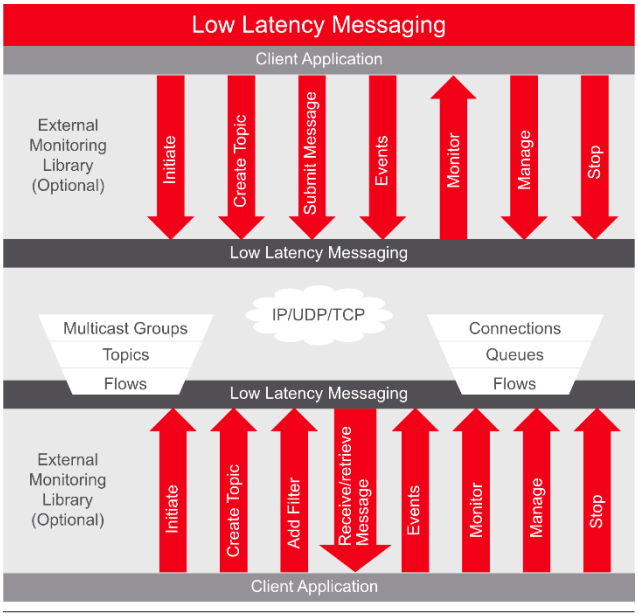

Figure 1: CLLM interfaces with enterprise applications in either multicast or unicast mode.

CLLM 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.

The second alternative offers reliable, point-to-point, unicast messaging using 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 speed.

To facilitate the design of the messaging infrastructure, CLLM supports the unique Multicast-by-Unicast (MBU) transport, which allows the creation of a single topic with multiple physical destinations.

Using an MBU topic, the application can send a message once and the message will be routed in the most efficient manner to shared memory, UDP unicast or multicast destinations.

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 several multicast groups or connections. If several streams share one group or connection, the receivers are still able to de-multiplex 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.

CLLM supports load-balanced delivery of messages over a message stream. Multiple receivers can subscribe to a message stream and each message is

delivered to only one of the receivers. Load- balanced message delivery can be used to provide lower latency or high availability by fan-out of a message to multiple processing applications.

Topic resolution

A topic defines a uniquely named logical message stream. CLLM has centralized topic configuration allowing applications to submit minimal information about the message stream (i.e., the topic name) and the topic configuration services will provide the required configuration (i.e., multicast groups, unicast addresses or shared memory segments) 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. Most parameters needed to create or connect to the message stream are specified using the topic resolution services, simplifying the configuration of application endpoints and providing a centralized point for management of the messaging environment.

Reliability

CLLM has been architected to support reliability at several levels. A packet transport layer resides above the datagram layer, incorporating IP, UDP and TCP. For multicast communication, the packet transport layer conforms 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, CLLM 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.

Message acknowledgment control

CLLM 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 packet loss is possible if a receiver does not recognize 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 assure 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-NACK" 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.

Packet and message management

CLLM 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 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 product has an “unreliable streaming” transmission mode for real-time data and other information feeds that do not require delivery assurances. This mode uses a “fire-and-forget” approach where there is no retransmission of packets. Other applications will require reliable, in-order delivery of all stream 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 sequence (or range) of missing packets to the transmitter, requesting retransmission. The transmitter maintains a history queue per stream. 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, possibly throttling the transmitter. With a NAK mechanism they are discarded when the transmitter’s history buffer is full and new packets are sent.

The 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 peer-to-peer data exchange, with any host being able to both transmit and receive messages in a daemon-less 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.

Message persistence

A fundamental limit to the reliability within CLLM 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. The Message Store is highly configurable and allows filtering of the stored messages.

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.

CLLM supports the request of previously transmitted messages of a message stream from the Message Store and transparent transition to the live transmitter message stream, using the Message Store Late-join feature. With durable subscriptions, an application can stop and, when restarted, it will receive from the Message Store the messages that were sent while it was stopped. Then, it switches back to the live message stream.

The version 3.0 of CLLM now includes this functionality to support durable subscriptions of message streams within the license.

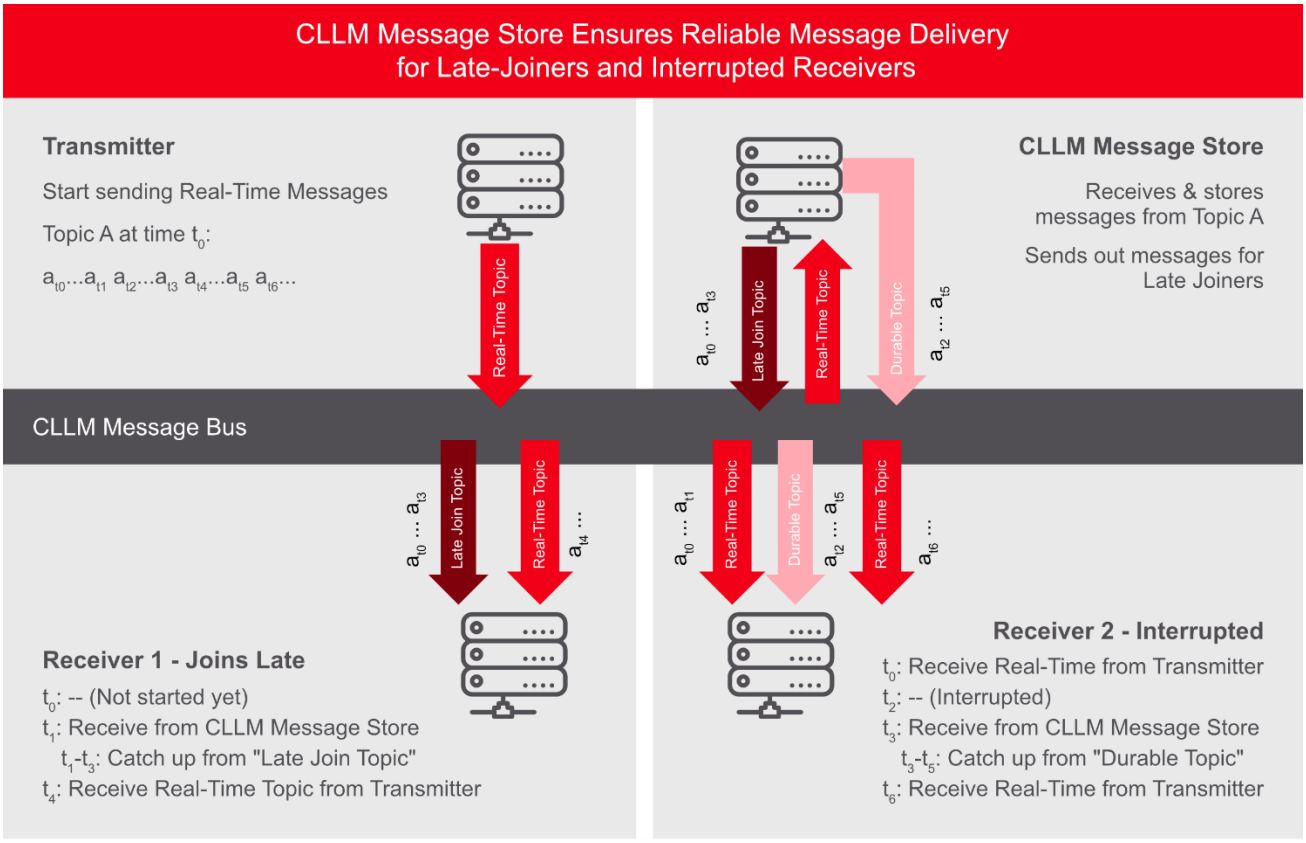

Figure 2: CLLM provides seamless message recovery from the Message Store component

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.

Monitoring and congestion control

Many existing applications experience difficulties with the volume of events they must consume from today’s volatile markets. CLLM congestion facilities help ensure that the infrastructure continues to per- form 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 net- work congestion problems, and to manage these detected problems by handling slow receivers or regulating the transmission rate.

CLLM includes a comprehensive 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. CLLM provides message-based clock synchronization technology.

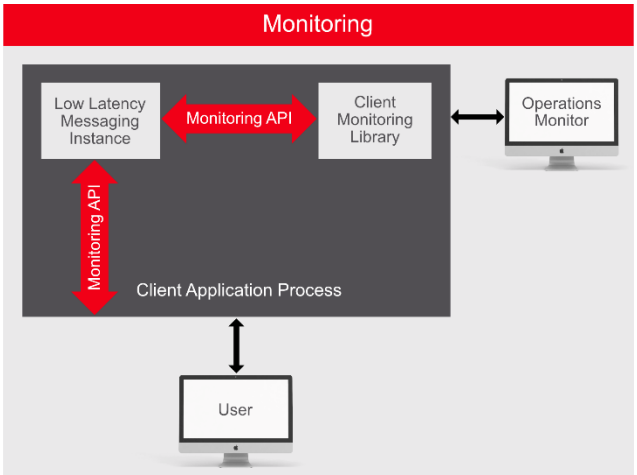

Figure 3: CLLM includes a robust monitoring API.

This technology preview can be used within the product's monitoring functions for precise end-to-end latency measurements.

Additionally, several options are available for network congestion management. By default, CLLM 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.

Advanced message filtering

Label-switched dynamic accumulation technology in CLLM 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 control of the amount of data that is delivered to each application, making the most efficient use of network bandwidth and processing resources.

The message stream is processed at multiple layers before delivery to the application. The data is analyzed and for- warded 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.

TurboFlow technology maps the relevant data parameters, such as a symbol name, to a string, integer or bitmap label - depending on the ease-of-use and performance needs of the application. Labels are assigned by the transmitting application to the message when transmitted and used by the receiving layers to make the routing or filtering decision, instead of the full parameter analysis. Millions of logical flows can be efficiently handled using TurboFlow technology.

CLLM 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.

High availability capabilities

CLLM 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 CLLM.

RCMS utilizes Reliable Multicast Messaging (RMM), which provides high performance, reliability, late joiner support, congestion, and traffic control.

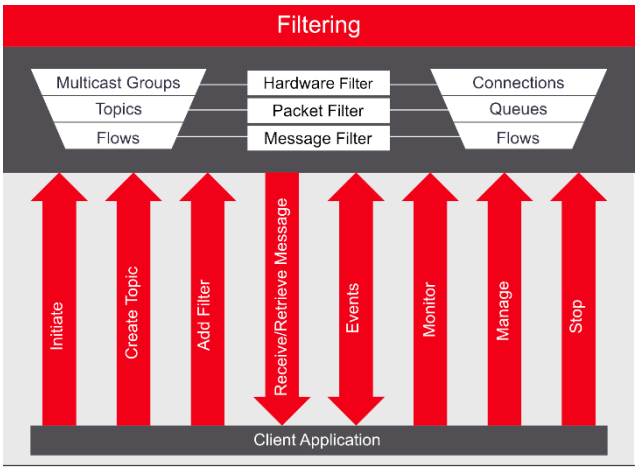

Figure 4: CLLM offers coarse- and fine-grained filtering options.

RCMS via transparent stream failover provides high availability for the message stream. It delivers system reliability with negligible performance impact - a business-critical benefit for today's organizations that need both high availability and high performance for their 24x7 applications. Organizations can decide on different options based on their specific technology and business requirements.

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, as well as handling 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 product supports two levels of failover for a transmitter tier which can be classified as tier duplex ("hot-hot") or tier ("hot-warm"). An RCMS transmitter tier with members in a "hot-warm" configuration satisfies most of the high availability needs of an application. However, for environments needing extremely fast failover, a duplex tier can be used.

In a typical, hot-warm RCMS transmitter tier there are two or more nodes, potentially connected to different networks, with one node designated as primary mode and others as backup mode. In a backup node, CLLM accepts messages submitted by the sending application (or component) and builds history buffers as usual, but it does not send packets out. If a primary node should fail, a backup node is activated, and CLLM starts sending packets from that node. It receives retransmission requests for missing data and resends packets that contain the required messages.

Using this RCMS tier configuration, the CLLM application achieves high availability with rapid and lossless failover. However, retransmission of messages can occur because the backup node in a transmitter tier may begin transmitting messages at some point in the message stream later than the last packet the receiver set has seen. This packet sequence gap results in retransmissions and increased latency of the messages that were missed.

For the quickest failover without message loss or retransmission, the RCMS transmitter tier can be configured in duplex tier (“hot-hot”) mode. Two active nodes in the tier are designated as primary nodes and all others are in backup mode.

CLLM sends messages in the active nodes and suppresses messages in the backup nodes. If an active node should fail, a backup node becomes active and starts sending messages in this node.

In the RCMS receiver tier, CLLM performs the required setup in each node. It connects to all networks, joins all relevant multicast groups, and completes all relevant unicast connections. If it receives messages from an RCMS transmitter tier in “hot-warm” configuration, RCMS will receive a failure event when the activation of a new sender in the transmitter tier is detected in the event of failover. The event is delivered to the application event listener and RCMS starts reception of the new data stream while detecting if messages were lost during any failover. CLLM sends a retransmission request to the activated backup sender for any missed packets. It also delivers messages from the new source to the same application message listener, making the failover transparent to the application message processing.

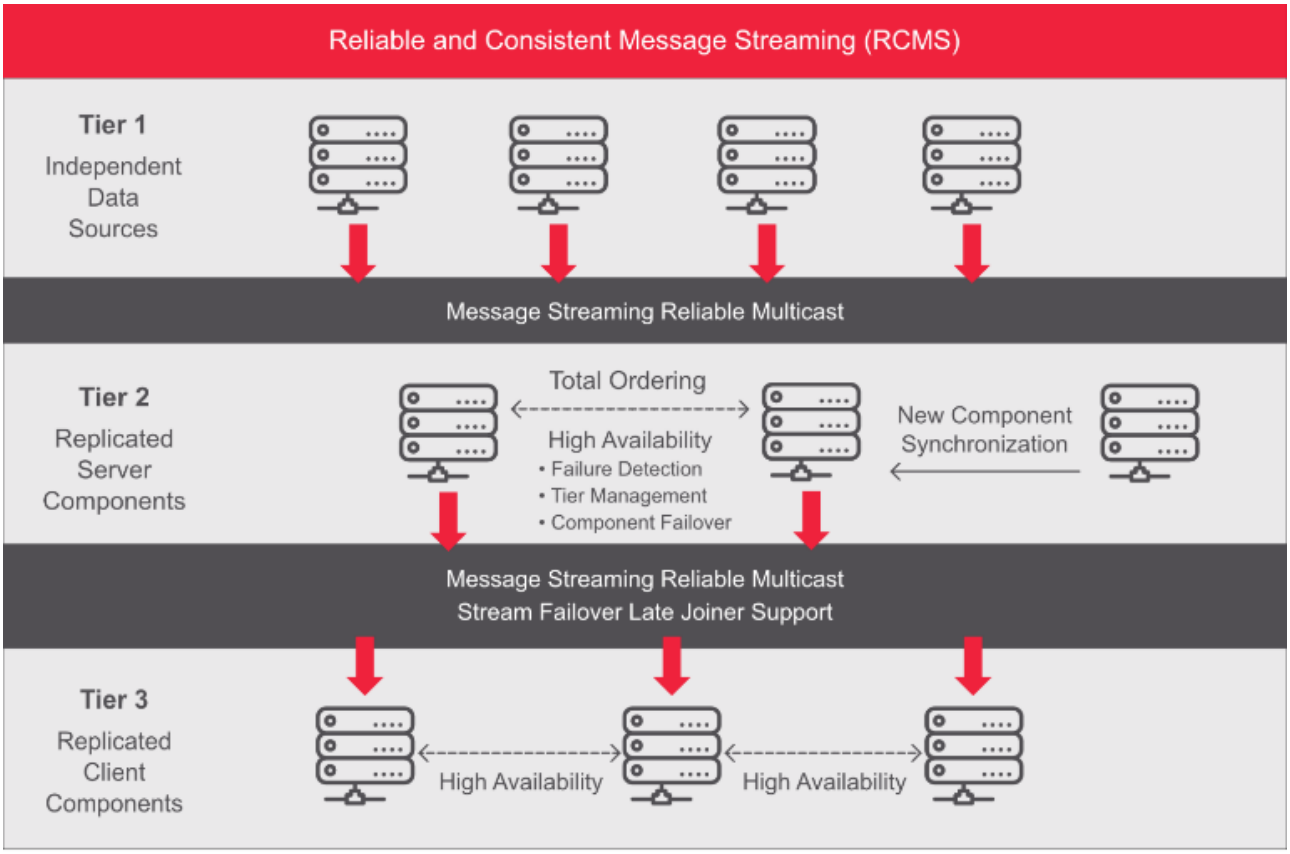

Figure 5: The RCMS component of CLLM provides advanced high-availability features such as failure detection, component synchronization, and stream failover

If the RCMS receiver tier is receiving messages from a RCMS transmitter tier in “hot-hot” configuration, the receiver nodes will receive duplicate messages from the duplex active transmitter nodes. Duplicate messages are identified and discarded at the RCMS layer, so that the application is only delivered one copy of the message. In the event of failure of one of the duplex active transmitters, there will be no message loss as the receivers will continue to receive the other active message stream.

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.

Review the benefits of Confinity Low Latency Messaging

CLLM 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.

CLLM provides reliable multicast and unicast messaging, high-performance and fine- grained message filtering, message persistence, and advanced high availability capabilities. The product efficiently supports millions of logical message flows. It also provides APIs to monitor statistics and performance, allowing 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, CLLM is highly configurable and can be adapted for a variety of application messaging and threading requirements. The solution runs on Microsoft® Windows® platforms and Linux® platforms.

With CLLM solutions can be designed specifically for the very high-performance, low-latency requirements of the financial services 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 Confinity Low Latency Messaging, please visit our website at www.confinity-solutions.com or contact us at info@confinity-solutions.com

Confinity Solutions GmbH
Mergenthalerallee 45-47
D-65760 Eschborn
Germany

* The performance numbers listed for Confinity CLLM are based on measurements using standard 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.

Please recycle

CS-LLM-ENG-01