Cyber Integration, Message Fabric and Streaming Analytics

SCRE Workshop

November 17, 2015
Why a Message Fabric for Cyber Integration?

• Abstraction (Pub-Sub, Request / Response, Queuing)
  • **Separate physical systems from communication**; use any infrastructure without changing system behavior
  • **Single point of web-based management**

• Modularity
  • **Quickly add new technologies/algorithms to stay ahead**

• Efficiency
  • Instant response needed? Maybe not, but **latency matters**!

• Functionality
  • Discovery, connectivity, **reliable exchange of data**
  • Guaranteed delivery, fault tolerance, load balancing
  • **Commodity hardware = lower entry and O&M costs**
The Race to Respond

why speed is critical for Cyber Defense

Fast
- Events
- Recognize
- Respond
- You won!
- Attack started
- Attack failed

Slow
- Events
- Recognize
- Respond
- You lost!
- Attack succeeded

Starting Line
The Race to the Exchange

why speed is critical for Capital Markets

Fast

Market Data

Feed Handler

Execution

Exchange

You won!

Got
GE at 30.21

GE at 30.16

Slow

Market Data

Feed Handler

Execution

Starting Line

You lost!

Got
GE at 30.45
### Market Data Growth = Data Deluge

Aggregated One Minute Peak *Messages Per Second* Rates
Arca, CTS, CQS, OPRA, NQDS
(in *thousands*)

- > 1 Terabyte of Data per Day

<table>
<thead>
<tr>
<th>Year</th>
<th>Total</th>
<th>Options</th>
<th>Equities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dec-00</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-01</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-02</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-03</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-04</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-05</td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-06</td>
<td>160</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-07</td>
<td>310</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-08</td>
<td>559</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-09</td>
<td>696</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-10</td>
<td>1,100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-11</td>
<td>1,562</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-10</td>
<td>2,562</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-11</td>
<td>3,410</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-11</td>
<td>5,957</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-11</td>
<td>4,380</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-11</td>
<td>3,410</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dec-11</td>
<td>7,174</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Informatica
Legacy Messaging Architectures

Daemon Based Design
6 Data Hops

Broker Based Design
4 Data Hops
2004 – Need for a State Change

- Motivations / Challenges
  - Not scaling to today’s needs (yet alone tomorrow’s!)
  - Availability at risk due to single points of failure
- Brokers are a bottleneck
  - Broker is a source of contention that limits scaling
  - Broker failure disastrous to latency and stability

Remove the Broker from the Message Path!
Case Study: Direct Edge
3rd Largest US Stock Exchange in 2008 (after NYSE and NASDAQ)

- 50% reduction in hardware cost
- 75% lower latency
- Replaced brokers with peer-to-peer message fabric
- Consistent Performance & Stability with Record Transaction Volumes
- Increased resiliency
- 50% reduction in hardware cost
- Predictable performance

Source: Direct Edge 2008-2009
Near Real-Time Financial Data Analytics Framework

Counterparty Risk Assessment
Current State – Disparate Data Siloes
End-of-day extracts and long load/processing times

Risk Management
- Market Risk
  - Interest rate risk
  - Currency risk
  - Equity risk
  - Commodity risk
- Credit Risk
  - Counterparty risk
  - Sovereign risk
  - Concentration risk
  - Securitization risk
  - Credit derivative risk
- Liquidity Risk
  - Asset risk
  - Funding risk
- Operational Risk
  - Legal risk
  - IT risk
  - Reputational risk

Front & Middle Office Operations
- Trading Transactions
- Lending Transactions
- Payment Transactions
- External Transactions

App Development
- Portfolio
  - Market Data
    - Pricing
    - Products
- Client Activity
  - Credit Lines
    - Limits
    - History
- Credit Lines
  - Liquidity
  - Controls
  - Routing
- Fraud
  - Process Break
  - External Events
  - Market/Fiduciary

End-of-day Extracts

Informatica
Desired State – Correlation across all Data
An Open “Single Source of Truth” for Financial Data

Low Latency Message Fabric

Market Risk
- Interest rate risk
- Currency risk
- Equity risk
- Commodity risk

Credit Risk
- Counterparty risk
- Sovereign risk
- Concentration risk
- Securitization risk
- Credit derivative risk

Liquidity Risk
- Asset risk
- Funding risk

Operational Risk
- Legal risk
- IT risk
- Reputational risk

Trading Transactions
- Portfolio
  - Market Data
  - Pricing
  - Products

Lending Transactions
- Client Activity
  - Credit Lines
  - Limits
  - History

Payment Transactions
- Credit Lines
  - Liquidity
  - Controls
  - Routing

External Transactions
- Fraud
  - Process Break
  - External Events
  - Market/Fiduciary

High Performance Data Analytics
- Cross Silo Risk Mgmt
- Trade Lifecycle
- Market Scenario
- Trade Latency Compliance
- Trade Cost Analysis
- Portfolio Management

Near Real-time Access

Risk | Ops | Apps
Near Real-Time Data Analytics
Real-time & Historical Stock Data with Near Real-time Query

Core Components
- Massively Scalable Database
- Message Ingest
- Transform/Replay

Programmable Analytics
- SQL, MapReduce, PL/R

Low Latency Message Fabric
- Pub/Sub
- FIX
- FPML
- OPRA
- ARCA
- CME

Real-time Data Acquisition & Transformation
- Real-time Feeds
- Historical & End-of-day Data Acquisition & Transformation
- ETL and Batch Upload

Historical Database Query and Replay
- Historical & End-of-day Data

Core Components
- Near Real-time Database Query and Analytics

- Programmable Analytics

- Flat file

- • Positional Trade Strategy
- • Dashboard & Reporting
- • Risk and Compliance
- • Intraday Operations
Sample Trade Workbench Real-Time Dashboard
Not a Production View
What about real-time?

Streaming Analytics and Processing at the Edge
Processing “at the Edge” (and elsewhere)

• Considerations
  • **Aggregation** and **correlation** necessary for “big picture”
  • More **distributed** processing power than centralized
  • Raw data is necessary for some types of analysis
    • Is it more efficient to send raw + processed or process later?

• Strategies
  • Derive as much as possible as early as possible
    • **Continuous computation** – counters, distribution statistics
    • **Enrich** (tag/classify unstructured events, add provenance details – origin, identity, versioning, chain of custody)
    • **Exception monitoring** – deviations from norm, trending up/down to exceed thresholds
  • Filter, summarize, compress, transform, mask, encrypt
    • Focus on **state changes** (111100001110011100)
    • No-change is data too, but **heartbeats** may be enough
Scalable Deployment w/ Distributed Nodes
AFOC, DCGS-A, DCGS-AF, NATO, IC

**Core nodes** provide:
- Event Fusion
- Content-based Routing
- Stream interfaces

**User nodes** provide:
- User-defined rules
- Enrichment with data warehouse and MDM
- Event-driven analytics
- Integration with alerting channels and workflows

**Edge nodes** provide:
- Filtering
- Classification
- Streaming data masking
- Real-time aggregates
- In-line enrichment

All nodes managed as a single topology from a central console.
What we want in a Message Fabric
Peer-to-Peer Message Fabric

Sending CPU

Application A
Comm

1

Receiving CPU

Application B
Comm

“Nothing in the Middle” Data Hop

Network

Functions handled by modern O/S, CPU, Network and API
- routing
- forwarding
- filtering
- fan-out
- persistence

Benefits
- efficient (single data hop)
- maximizes performance
- no single points of failure
- scalable and flexible
- easier to administer

Just 2 steps to move from A to B!!!
Less is more!!
Parallel Persistence®
Parallel Persistence®

Zero System Downtime!
Zero Latency Failover!

Sending Application

Persistent Data Stores

Receiving Applications
Parallel Persistence®

Sending Application  Persistent Data Stores  Receiving Applications
Parallel Persistence®

Receiver recovers with no impact to live message stream, then rejoins the live stream!
Extended Enterprise

Considerations:

- Availability
- Authentication
- Authorization
- Bandwidth
- Encryption
- Filtering
- Firewalls
- Protocols
- Routing
Dynamic Routing - Least Cost Path

= App Domain
= WAN / firewall
= Message Router
= Sender / Receiver

Dynamic Routing - Least Cost Path
Dynamic Routing - Least Cost Path

= App Domain
= WAN / firewall
= Message Router
= Sender / Receiver
How can you combine a peer to peer message fabric with standardized interfaces and centralized management?
Streaming data collection…

WEB LOG DATA

00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down
*Mar 1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
*Mar 1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)

SERVER LOG DATA

00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down
*Mar 1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
*Mar 1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)

EVENT DATA

00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down
*Mar 1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
*Mar 1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)

SENSOR DATA

00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down
*Mar 1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
*Mar 1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)

LOCATION DATA

00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down
*Mar 1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
*Mar 1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)

DEVICE DATA

00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down
*Mar 1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
*Mar 1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
How do you manage this?

1100s
Centralized management, peer to peer data flow
Summary: Essential Characteristics

- No daemons or servers in delivery path
  - Maximize speed and scalability
  - No single points of failure
- Choice of protocols (data “payload” agnostic)
  - TCP, UDP, AMQP, unicast, multicast, shared memory, etc.
- Secure transports, handshakes and storage
  - Integrity, with or without confidentiality
- Secure message routing for extended enterprise
  - Intelligently bridge segmented networks and applications
- Centralized monitoring (with API)
  - Integrated insight from every endpoint (other layers too!)
Summary: Essential Characteristics (cont’d)

• Dynamic service and peer discovery
  • Move applications without changing configuration or code
  • Establish data flows out-of-band to minimize overhead
• Full range of qualities of service
  • From reliable (best-effort) to durable (guaranteed)
• Standards-based interfaces
  • Easily plug in third-party products and services
• Centralized management (with API)
  • Configure top-down; implement locally
• No custom hardware
  • Pure software to always run on best infrastructure
Thank You!

Gay Adams

gadams@informatica.com

cell: 301-980-9148