Low Latency and High Volume Architecture
Contents

- Business Problem
- NFR Challenges
- Solution
- Technology overview
- Performance Monitoring
- Approach
- Outcome
- Lessons Learned
Business problem
Overview

Current Situation

- Interest Rate Swaps
- Manual Processes
- Multiple sources
- Trading Mechanism risk to dealers
Overview

Current Market Environment

- Introducing change in a volatile environment
- Change vs execution confidence

Credit Crunch hits high street

‘Market can find solutions to sub prime debt markets’

Volatile markets......
NFR - Key Challenges

Performance
- Trade negotiation is time-sensitive, < 100 ms latency required
- Year one volume 20,000 msg/s scaling to 100,000 msg/s by year five.

Reliability
- No loss of trade messages / transactions

Ease of Integration
- Dealer & Distributors

Availability
- 99.99%

Time to Market
- Go Live in 9 Months
Solution
Solution - Functional Overview
1. Dealer streaming in prices
2. Payload transformed and sent to pricing services
3. Generates composite prices, twice every seconds
Technology - Products

- **Gateway Tier**
  - Cameron US
    - Quote Gateway
  - Trade Gateway
  - Cameron MDS
    - Market Data Gateway

- **Middleware Tier**
  - Fiorano JMS
    - Middleware Server
  - Fiorano JMS
    - Middleware Server

- **Application Tier**
  - Spring
    - Trade Services
  - Spring
    - Admin Services
  - Spring
    - Pricing Services
  - Spring
    - Reporting Services

- **Data Tier**
  - Oracle RAC
Technology - Product Performance

- **Cameron Universal & Market data Server**
  - **Fix Servers**
    - FIX is a standard communications protocol within the Investment Banking domain but is flexible enough to be customised
    - Cameron US/MDS can handle > 25,000 msg/s

- **Fiorano MQ**
  - **Pure JMS Server**
  - Can handle > 80,000 non-persistent msg/s
Technology - Product Performance

- WebLogic Real Time provided a huge performance improvement compared to standard Sun JVM
  - With - 15ms latency
  - Without - significantly > 100ms latency
- Underpinned key components in each tier
  - Fix Gateway Servers
  - JMS Servers
  - Spring
- Made Java solution a feasible and good choice for this system
- System handling ~ 25,000 msg/s with 95th percentile latency ~15ms
Performance Monitoring

- JMX Monitoring Framework

1. MBean Server for each component/service
2. Admin Services retrieves all statistics periodically
3. Monitoring Web App displays statistics to users

Gateway Tier
- Quote Gateway
- Trade Gateway
- Market Data Gateway

Middleware Tier
- Middleware Server

Application Tier
- Trade Services
- Admin Services
- Pricing Services
- Reporting Services

Data Tier
- Database
Performance Monitoring

- Administration application provides detailed statistics:
  - Latency
    - Per message type
    - Per Component if required
  - Throughput
  - Service/Server Status

- Provides an easy to understand dashboard to the LiquidityHub Staff.
Approach

- Reference Architecture
- Design completed in horizontal and vertical
- 4 architects, 4 ??
  - Responsibilities
  - Hands on ?
  - Expertise in the correct area
Outcome

- Designed, built and delivered the system in the 9 months
  - System meets year 1 peak performance targets
    - Current latency figures 4 - 8ms
  - System can scale to much higher message volumes
Lessons Learned

• Create more testing tool
  • Distributor GUI test harness
  • Log search and view

• Early prototyping helped de-risk the architecture

• Consider virtualisation

• Managing 17 organisations is not easy
  • Weekly tech forums worked well
  • Dedicated relationship manager
Break
Website
http://www.codingthearchitecture.com

London User Group
Monthly mix of presentation and discussion

Google Group
http://groups.google.com/codingthearchitecture