ORACLE®

ORACLE®

QCon: London 2009

Transforming the Reconciliation Process

Brian Oliver | Global Solutions Architect | brian.oliver@oracle.com

Oracle Coherence | Oracle Fusion Middleware Product Management

Agenda

- What is the Reconciliation Process?
- The types of Reconciliation
- Why do we need to “Transform” it?
- Will a Data Grid help?
- Introducing Event-Based Reconciliation
- Demonstration
- Next Steps?

The proceeding is intended to outline general product use and direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

What is Reconciliation?

What is Reconciliation?

- **Reconciliation:**
  “The process of comparing and matching figures from your records against those presented on a statement issued by a third party” - Wikipedia.

- **Purpose:**
  “… allow detection of possible discrepancies.”

- **Why:**
  Discrepancies == Risk and/or Loss

Two causes of “discrepancies”

Accidental

Deliberate

Failure to Reconcile increases Risk

- **Observation 1:**
  As the time between completed reconciliation processes ➔ ∞, potential losses tend to ➔ ∞

- **Observation 2:**
  Continuing to trade without knowing how much money you really have is really stupid (and often illegal)

Examples

- **Example 1:**
  Hedge Fund reconciled trading charges each quarter. At the end of a quarter discovered they overpaid $250,000. The broker could not repay the over payment immediately.

- **Example 2:**
  Investment manager failed to regularly reconcile positions with custodian. Trader managed to personally trade (and lose) investment funds

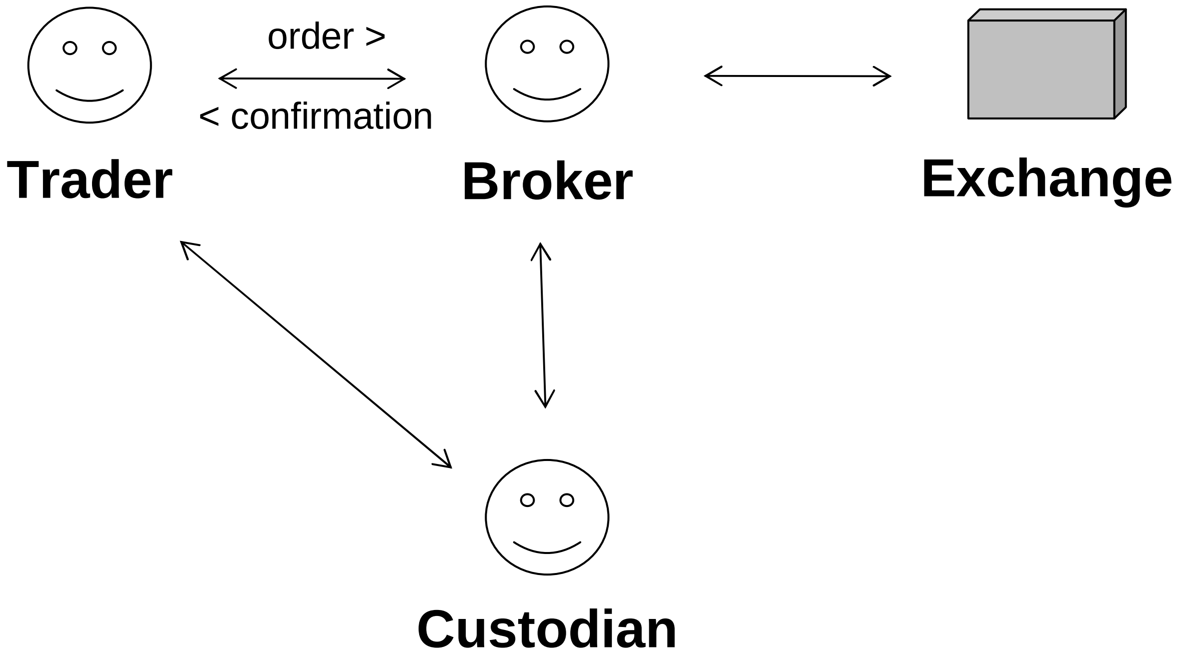
  ie: insider trading

Examples

- **Example 3:**
  Bob Cratchit (A Christmas Carol) could not go home on Christmas Eve until the bank reconciliation had been completed (for Ebenezer Scrooge)

Types of Reconciliation

Trading Relationships



Types of Reconciliation

- **Trade Reconciliation:** Do the trade confirmations (from our brokers) match our instructions (ie: orders)?
- **Position Reconciliation:** Does our position match what is held by the Custodian(s).
- **Cash Reconciliation:** Does the cash generated/lost by our trading match our bank balance?
- **Charges Reconciliation:** Do the charges/taxes we have to pay match our estimate(s)?

Reconciliation Gotcha’s

- “Perfect” matching may not be possible

- **One Possible Scenario:**
  - Trader:
    - Buy 1000 ORCL @ $13.50 (order id: 1)
  - Broker:
    - Confirm 600 ORCL @ $13.70 (order id: 1)
    - Confirm 100 ORCL @ $13.80 (order id: 1)
      <delay of several hours or a day!>
    - Confirm 200 ORCL @ $14.00 (order id: 1)
    - Confirm 100 ORCL @ $13.50 (order id: 1)

Reasons to “Transform” the Reconciliation Process?

The desire:

Real-time Risk Assessment!

Current Reconciliation Processes…

- Use client + server architectures
  - “Clients” perform the matching (move a lot of data)
- Use disk-based storage (flat files, databases)
  - Good Capacity (for handling trades)
  - Poor Performance (for matching)
- Are batch-based
  - Usually and over-night process
    (not real time)
  - May not complete in time for the next day
    (when volumes spike)

Possible Solution…

- Solution: Just do matching in memory?

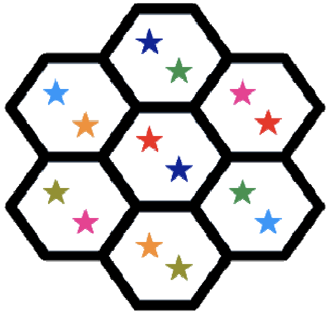
- How to address the…
  - Capacity Challenge?
  - Availability Challenge?

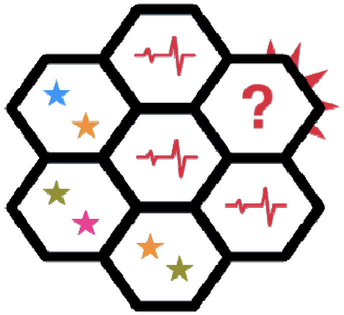
- Will a Data Grid help?
  - Solves Capacity Challenge (spreads trades across servers)
  - Solves Availability Challenge (iff resilient Data Grid)

Introduction to Data Grids

Oracle Coherence: A Unique Approach

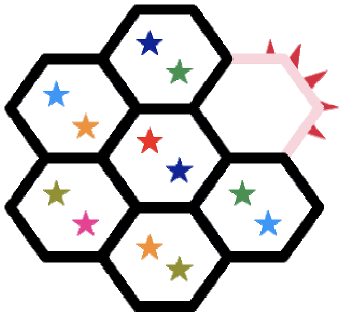
- Data is automatically partitioned and load-balanced across the Server Cluster
- Data is synchronously replicated for continuous availability



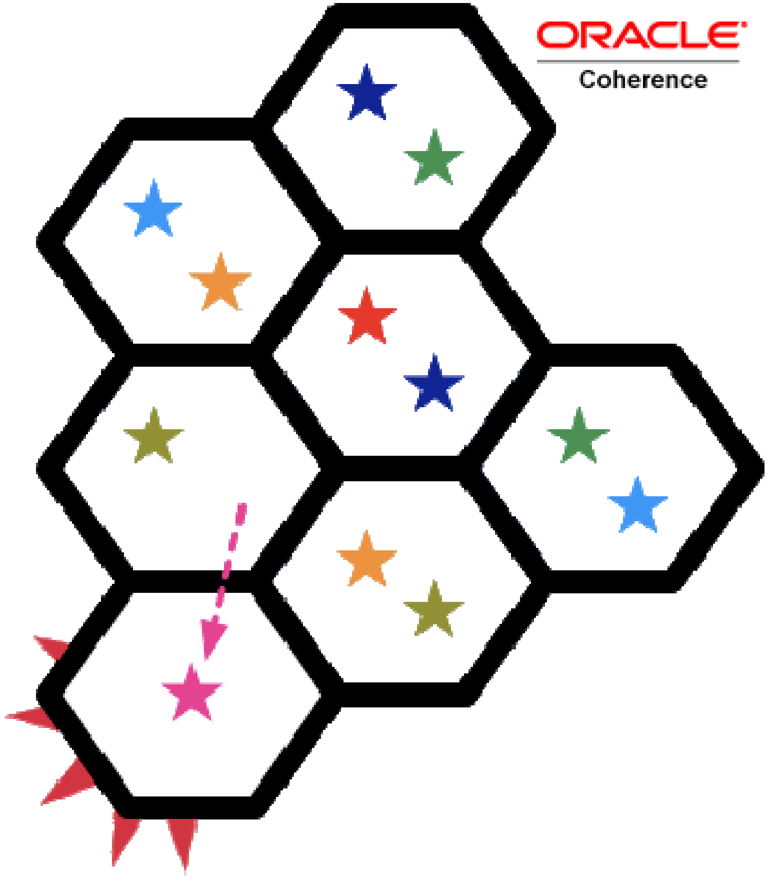


- Servers monitor the health of each other
- When in doubt, servers work together to diagnose status

- Healthy servers assume responsibility for failed server (in parallel)
- Continuous Operation: No interruption to service or data loss due to a server failure

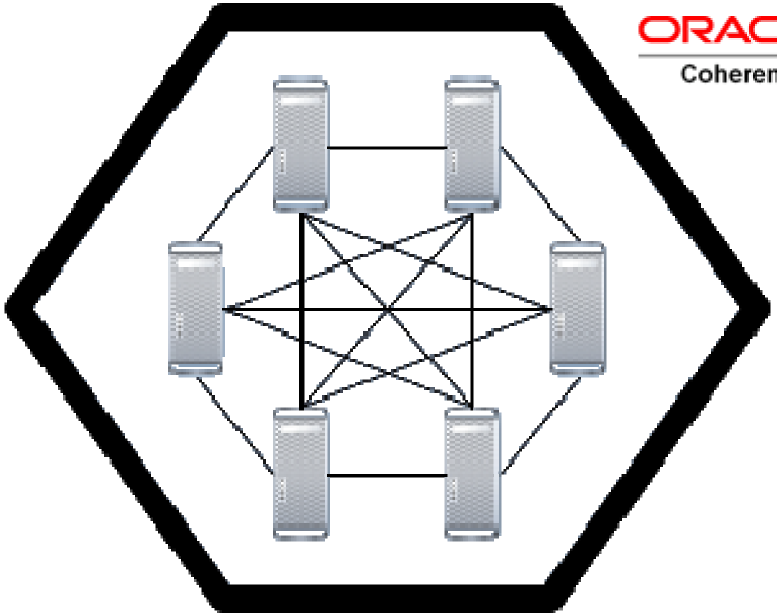


Oracle Coherence: A Unique Approach



- Dynamically scale-out during operation
- Data automatically load-balanced to new servers in the cluster
- No repartitioning required
- No reconfiguration required
- No interruption to service during scale-out
- Scale capacity and processing on-the-fly

Oracle Coherence: A Unique Approach



ORACLE Coherence

- Peer-to-Peer Clustering and Data Management Technology
- No Single Points of Failure
- No Single Points of Bottleneck
- No Masters / Slaves / Registries etc
- All members have responsibility to;
  - Manage Cluster Health & Data
  - Perform Processing and Queries
  - Work as a “team” in parallel
- Communication is point-to-point (not TCP/IP) and/or one-to-many
- Scale to limit of the back-plane
- Use with commodity infrastructure
- Linearly Scalable By Design

Will a Data Grid really help?

- **Observation 1:** Client + Data Grid architecture for matching won’t really help.
  - Problem: Network Latency becomes the bottleneck
  - Solution: Avoid moving all Data to the client. Have Data Grid perform matching **(in-place)**

Will a Data Grid really help?

- **Observation 2:** Matching Algorithms are often O(n^2).
  - Problem: Matching is essentially a “join” operation (compare all trades with all other trades)
  - Problem: In a Data Grid, a naïve “join” means every server talking with every other server.
  - Solution: Exploit parallelism of the Data Grid (ie: map reduce) and use Batching where possible.

Introducing Event-Based Reconciliation

How To: Event-based Reconciliation

- **Step 1: Loading**
  Load Trades into Data Grid as they are received (in real-time)

- **Step 2: Event processing triggers matching**
  When STREET trades are received, use map-reduce (in parallel) across the Grid to find potential matches to HOUSE trades

- **Step 3: Acknowledge Matches (in place)**
  Use in-place-updates (Entry Processors) to perform signal matches.

Event-based Reconciliation

- **Step 4: Use Events to Observe Reconciliation**
  Monitor the reconciliation process using Data Grid update events

- **Step 5: Use Data Grid Queries for Reports**
  Use Data Grid queries to produce real-time reports (reports executed in parallel)

Demonstration

Next Steps?

Next Steps?

- Opportunities for reducing network traffic
  - Increase “Batching”
    (batch updates instead of one-at-a-time)
  - Use Partition-based Map Reduce
    (reduce “entire” grid queries)
  - Use Data Affinity
- Graphical Front-End (excel integration)
- Release onto Coherence Incubator

Thanks…

ORACLE IS THE INFORMATION COMPANY