



Getting Real - an introduction to real time data warehousing

Peter Scott: Principal Consultant, Rittman Mead Consulting

T : +44 (0) 8446 697 995 or (888) 631 1410 (USA) E : enquiries@rittmanmead.com W: www.rittmanmead.com

About Me

- Principal consultant at Rittman Mead Consulting
- Specialising in Data Warehouse design and performance optimisation
- Many years experience of terabyte scale data warehouses
- Contributes to the Rittman Mead Blog on Data Warehousing, Data Quality and BI
- Writes for Conspectus and other on-line journals
- Speaks on BI and Data Warehousing at User Group meetings in Great Britain, Europe and North America

Outline of presentation

- Introduction
- Three reasons to consider Realtime DW
- Techniques for Realtime DW
- Practical example of Change Data Capture
- Possible pitfalls
- Key points revisited

Things used to be so much slower



Transaction

Things used to be so much slower



Transaction



Propagation

Things used to be so much slower



Transaction



Propagation



Data load

... but now

- High speed networks linking remote sites
- faster processor speeds
 - ▶ GHz not MHz
- more, more, more
 - ▶ memory
 - ▶ storage
- faster and faster
 - ▶ interconnects
 - ▶ storage
 - ▶ memory

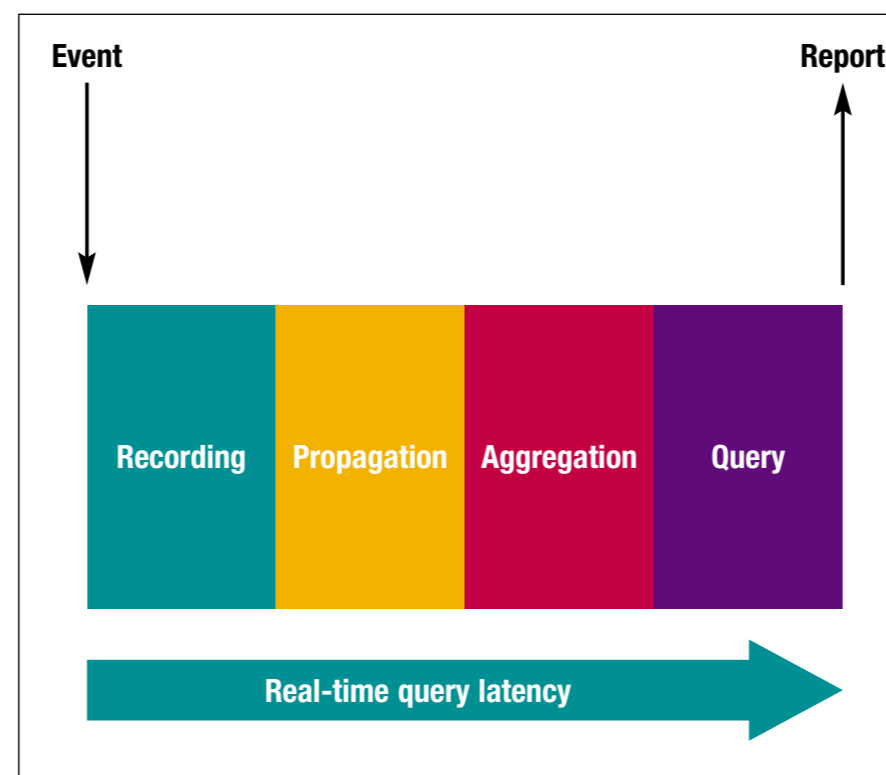
more can be done...

... but less time to do it in

- 24 x 7 business days
- Global trading and reporting
- need to push key figures out to executive's Blackberrys and iPhones
- BI feeding timely data to line of business applications

Latency

- Although things are a lot faster paced now there are still limitations on the interval between an event occurring and it being a “reportable fact” in the data warehouse.



Realtime BI \neq Realtime DW

- Realtime BI is independent of Realtime DW
 - ▶ Having one does not imply the other
 - We can have realtime reporting over production systems and not even have a data warehouse
 - We can keep our data warehouse up to date but choose to report on events up to a fixed horizon - say yesterday. This prevents reports “changing” throughout the day
- I will be mainly speaking about Realtime DW

Three Reasons to use Realtime DW

- We need to report on events that have only recently occurred
- We have not got time to load the data warehouse in a traditional batch mode
- We can't write traditional batch ETL against the source

Techniques for Realtime DW - 1

- Don't do it!
 - ▶ Report directly on the source system
 - Beware performance impact
 - ▶ Federate queries: 'yesterday' and earlier in the DW, 'today' from the source system
 - potentially less impact on source system
 - still need a daily load to move today's data into DW - but that can happen outside of working hours
 - Federated queries have the data stitched together on the BI server

Techniques for Realtime DW - 2

- Use a Realtime Replica of the source
 - ▶ Could use standby database
 - ▶ or specific reporting instance
 - need not be “full” replica
 - just items of interest
 - ▶ can use various replication techniques
 - Oracle GoldenGate can be used to create realtime reporting replica
 - ▶ Reporting structures mimic those on source system
 - May not be optimal performance

Techniques for Realtime DW - 3

- Micro-batch

- ▶ Replace daily load with frequent, small, batches
- ▶ Problems
 - detecting change on source
 - ensuring no data missed or duplicated
 - time taken to process micro-batch
 - impact on source and target systems

Techniques for Realtime DW - 4

- Intercept messages

- ▶ Many distributed systems use messaging as method of communication between system nodes
- ▶ We can listen out for these messages and use them to provide a feed to the data warehouse
- ▶ Problems
 - Messages may not provide all of the data needed
 - Messages are typically about 'fact' events
- ▶ OWB 11gR2 has the ability to consume and process AQ streams
- ▶ Write you own message consumer in J-Developer

Change Data Capture

- Key Steps
 - ▶ Identify change
 - ▶ Propagate change
 - ▶ Apply change on target

Identify changes

- In traditional extract processing we identify changes by looking for database rows with either a “timestamp” column later than the last extract time or some form of “not processed” flag
- In change data capture we use some form of database event to indicate a change has occurred.
 - ▶ Triggers
 - use Materialized View logs ???
 - ▶ Redo logs

Trigger Based Change Capture

- Write our own
- or (since Oracle 9i) use Oracle Synchronous Change Data Capture
- Both cases we populate a change table with the updated row
- NOTE !! - does not support direct path inserts as triggers will not fire !!
- NOTE !! can be high impact on source database

Redo based Change Capture

- Every time change is committed to the database redo is generated.
 - ▶ BEWARE NoLogging activities are not captured
- Change instructions held in the redo log files can be applied to a remote system to replicate the changes made on the source system.
- Oracle have various modes of using Asynchronous Change Data Capture depending on where the target database is located

Oracle Asynchronous CDC

L
a
t
e
n
c
y
+
+

- Hot Log
 - ▶ publish data as subscriber views on the source database
- Hot Log Distributed
 - ▶ mines current redo log and applies change to remote system
- AutoLog online
 - ▶ Applies change on remote system from shipped redo log
- AutoLog Archive
 - ▶ Applies change on remote system from shipped archive logs

Oracle GoldenGate

- Oracle acquired company
- Reads Oracle (and other vendor's) redo log files
- Generates own extract for propagation to target database
 - ▶ possibility to filter out tables and columns that are not of interest or to move all tables
 - ▶ can now filter on user
 - very useful for bi-directional data replication
 - ▶ can propagate data types not handled by traditional CDC
 - e.g. Spatial

Propagation

- Oracle Synchronous CDC uses subscriber views to access change set
- Oracle Asynchronous CDC propagates change using Oracle streams. User access through subscriber view
- Golden Gate propagates its generated trail file to remote systems by a variety of techniques.
 - ▶ Can be encrypted, compressed
 - ▶ trail file can be ftp'd, pumped, rendered as flat-file or XML message

Propagation - GoldenGate

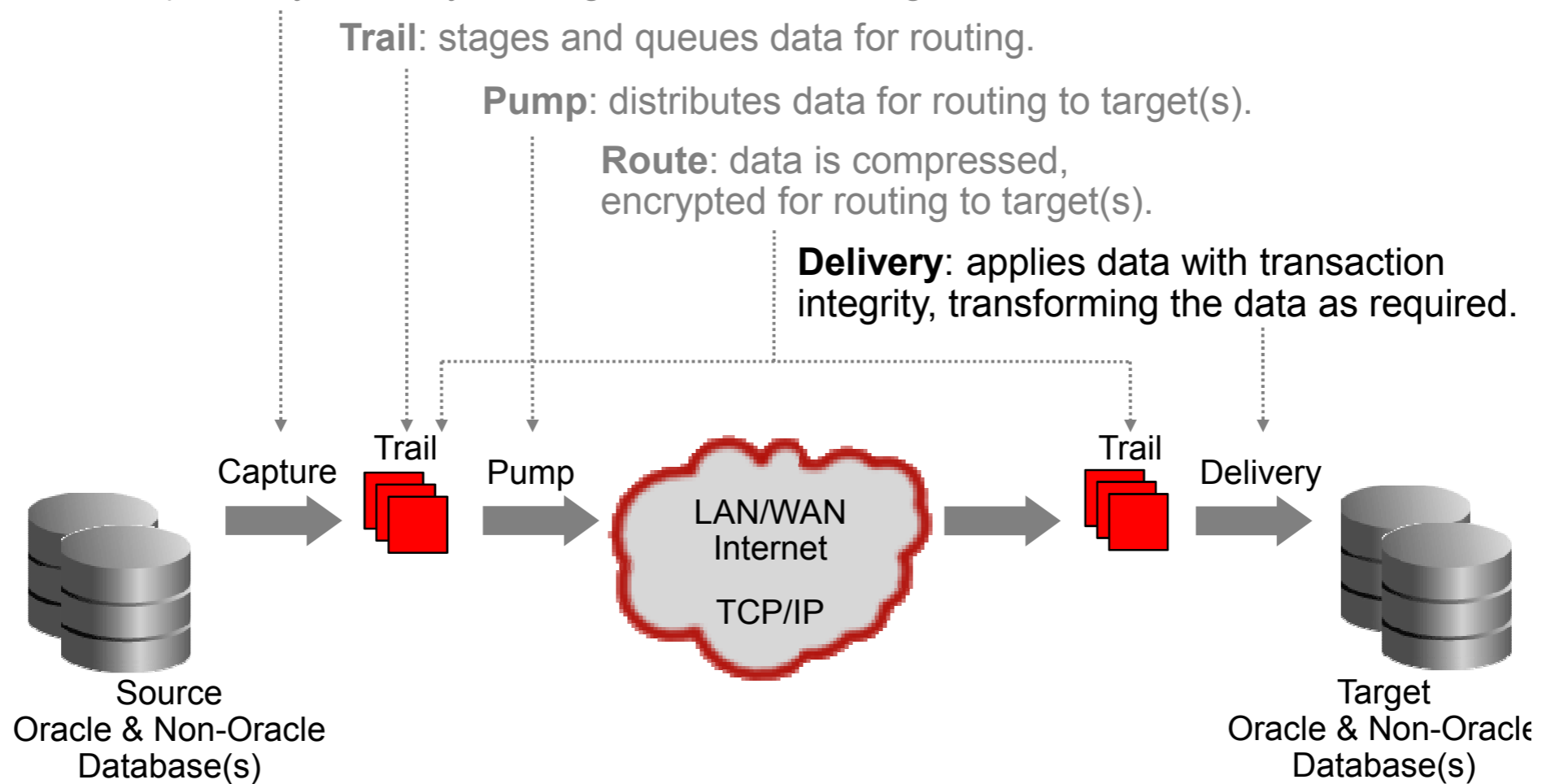
Capture: committed transactions are captured (and can be filtered) as they occur by reading the transaction logs.

Trail: stages and queues data for routing.

Pump: distributes data for routing to target(s).

Route: data is compressed, encrypted for routing to target(s).

Delivery: applies data with transaction integrity, transforming the data as required.



Applying Change - CDC

- For Oracle CDC we access change through a subscriber view.
 - ▶ these indicate:
 - the type of change
 - when the change occurred
 - ▶ typically we:
 - extend the subscriber window
 - process the data to load (select from subscriber view)
 - purge the subscriber window
 - ▶ for asynchronous CDC we should have supplementary logging enabled

Applying Change - CDC 2

- The Oracle Data Warehousing Guide describes the additional control columns present in the CDC subscriber view
- Key columns for our ETL are
 - ▶ OPERATION\$: what type of change
 - ▶ CSCN\$: The source SCN for the change
 - ▶ RSID\$: the row sequence ID. Use with CSCN\$ to determine the order in which multiple changes occur.
 - ▶ COMMIT_TIMESTAMP\$: when the change happened -
NOTE this is a DATE

Applying Change - CDC 3

- For SCD 1 we are interested in the final version of a dimensional column
- SCD 2 we need to process the history of change in order
 - ▶ We may need to add a timestamp to our staged CDC data so that we process correctly
- Fact captured by CDC also need to be timestamped especially if the dimensionality changes over time.

ETL considerations

- Oracle CDC change is published through subscriber views
- We can use these views as data sources within OWB or ODI
 - ▶ ODI (and now OWB) has knowledge modules to simplify the set-up of CDC, including GoldenGate
- If supplemental logging is not set correctly on the source database we will need to do work to synthesise the complete row to apply.
- We need to manage subscription windows; this can be done by a workflow in the ETL

Applying Change - GoldenGate

- For GoldenGate we need to process the transmitted changes.
- In its simplest form the transmitted trail file is used to populate the target tables
 - ▶ we can also generate flat files or pass XML messages for processing
- We must design the ETL process to consume the GoldenGate generated changes as they arrive. Or we must store versioned changes for processing in our micro-batch

Examples of CDC

- European e-retailer
- Order and fulfilment systems working 24/7
- No easy batch-based method to identify records to apply to data warehouse
- Some data highly volatile - price could change several times in a day for example
- Data warehouse loaded through a combination of traditional batch and ETL
- OWB 11gR1 and hand-coded views over the change views

Pitfalls of CDC

- Changes must be applied in the correct order
 - ▶ A record can change many times in a single CDC window
- Not all change is real change
 - ▶ some source systems open records for update and issue a commit even if no changes are made
 - ▶ some source systems issue several commits for what should be a single update
- If we capture “fact” by CDC we need to ensure that our dimensions are also up to date

Pitfalls of Realtime

- True realtime is impossible to achieve - there is always going to be latency
- Processing change through aggregation layers may not be worthwhile - what if the next change arrives before the aggregation is complete?
- Users are not always appreciative of reports that change through the day.... where has that “one truth” gone?

Key Points

- Think before going realtime - do you need it?
- Remember that most real-time is exposed to the data warehouse as a series of micro-batches, do you have time to complete the batch before the next one needs to be processed?
- Log based Change Data Capture is not very invasive on the source system and technologies such as GoldenGate can exploit that to give very low latencies

Thank You

- Please complete your speaker evaluation form, it really will help me give a better presentation next time!

- peter.scott@rittmanmead.com