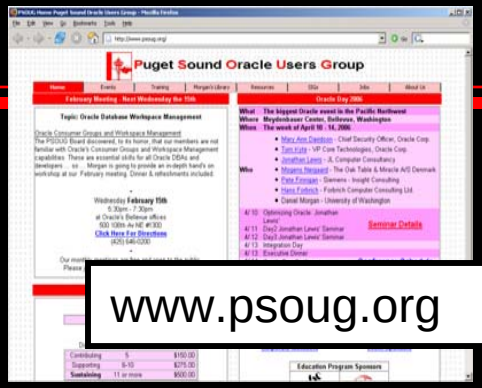
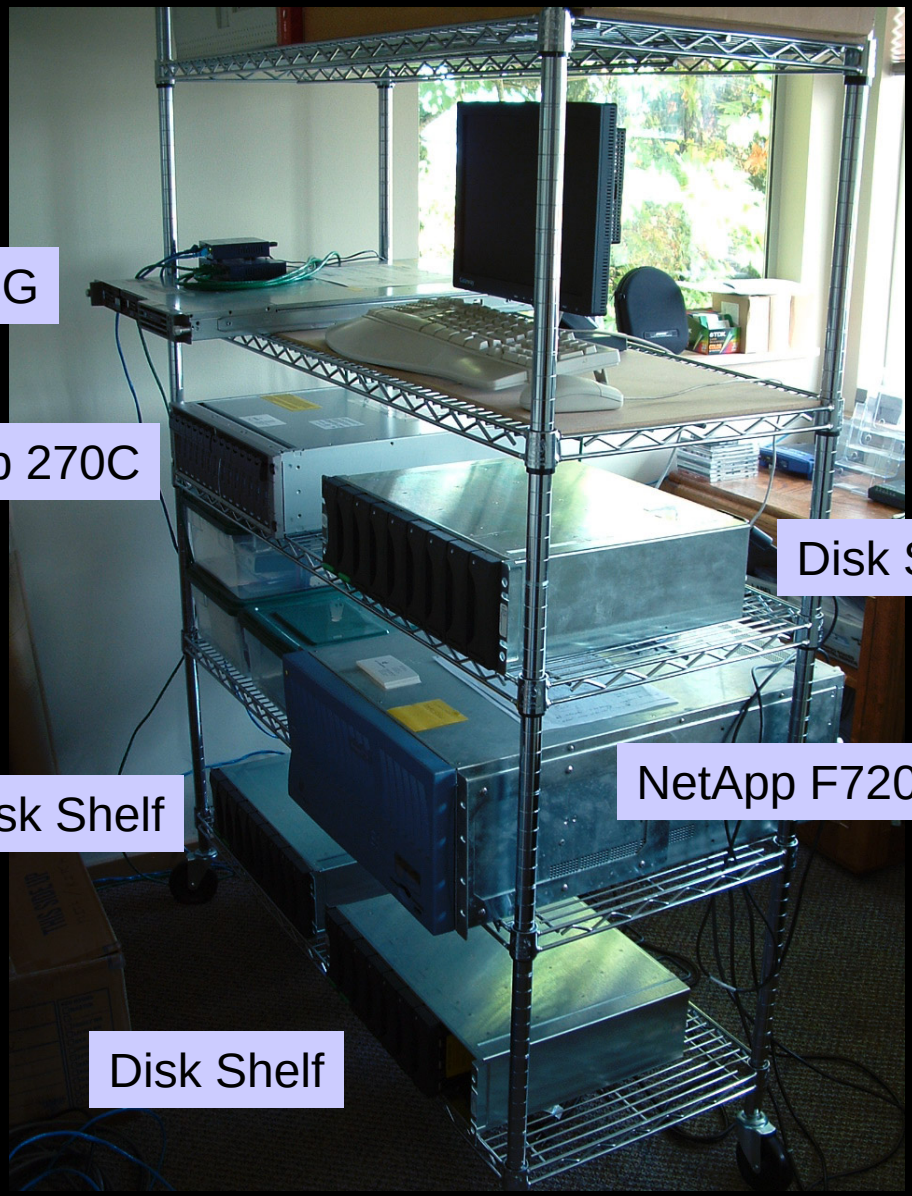
Oracle Streams and Change Data Capture (CDC) 10gR2

Introduction: Daniel Morgan



- IS/IT since 1969
  - IBM Mainframes (360/370)
  - Fortran IV
  - COBOL 10+ years ... and no I don't want to talk about it
- Oracle RDBMS since version 6
- University of Washington
  - Author UW Oracle curricula & primary instructor
- Education Chair: PSOUG
  - The "Morgan" of Morgan's Library
- Member: UKOUG
- Member: British-American Chamber of Commerce
- 20+ of hand's-on consulting (and other bad habits)

PSOUG Server Room



Objectives

- Replication Technologies
  - Advanced Replication
  - Data Guard
  - Streams
  - Change Data Capture (CDC)
- CDC Concepts and Architecture
  - Synchronous & Asynchronous / HotLog & AutoLog
  - Publish and Subscribe
  - Built on Advanced Queueing and LogMiner
- Implementation
- Performance
- Capture
- Publish & Subscribe
- Labs & Demo

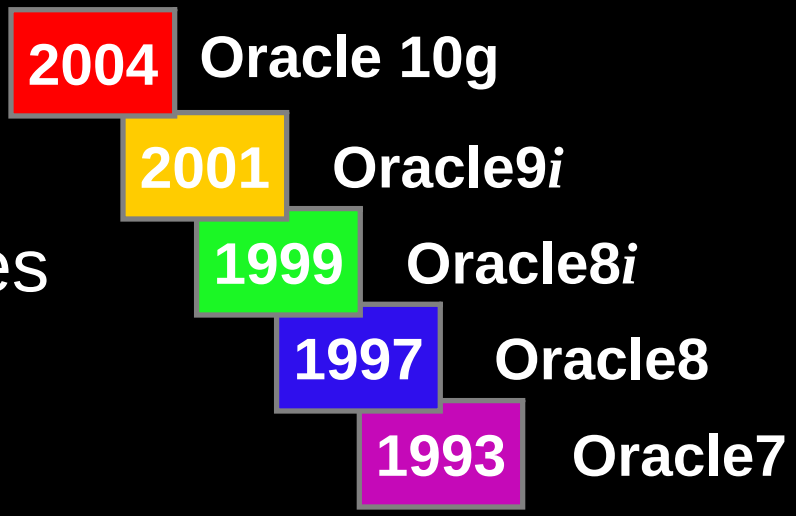
Legacy Capture

Method	Major Issues
Application Logging / Triggers	Maintenance, transaction impact
Timestamp, Change Key Column	Application design & performance impact, no before image
Table Differencing	Impractical for large tables, high transport cost, not timely
Log Sniffing	Not supported, does not track DB releases, security issues, rocket science

Replication Technologies

Database Replication Methods

- Advanced Replication
- Change Data Capture
- Data Guard **(use for HA)**
- Export – Import
- RMAN
- Streams (since 9.2.0.6)
- Transportable Tablespaces
- Hardware Based
  - snap mirror / snap restore



Concepts & Architecture

What is Streams Technology ...

- Oracle's marketing message
  - A replacement for Advanced Replication
    - Captures and distributes database updates, events, and application messages
    - Can automatically apply updates to destination databases, or pass events and messages to custom procedures and applications
  - Combining these capabilities provides an extremely flexible solution for replication, message queuing, and event notification solutions

and what is Change Data Capture (CDC) ...

- Captures change data from operational system(s) as it occurs
- Part of an Extract / Transform / Load (ETL) process DSS / Data warehouse, & applications
- Optimizes extraction and transformation
- Data change management framework
- Leverages mature technologies
  - Advanced Queuing
  - Log Miner
  - Triggers

... and why should I care?

- Less vulnerable to network, hardware, and application failures
- No requirement for real-time logical or dedicated connections
- Connections can be reestablished later w/o loss
- Different parts of an application can run at different speeds
- Cradle to the grave tracking
- Built-in auditing
- Built-in exception handling
- Less finger pointing (than with third-party solutions)

Change Data Capture

- Built on the Oracle Streams infrastructure
  - Evolves with the database
- Efficiently identifies and captures all data that has been changed during a transaction and who made them
- Transparently makes change data available where it is required including ETL tools and applications
- Quickly identifies and processes only the changed data: Not entire tables
- Offers tradeoffs between timely change delivery and source system overhead

Change Data Capture

- Transactional consistency for changes across multiple source tables is guaranteed
- Assumes complete change management responsibility
- You don’t need to be rocket scientist

Streams vs. CDC

- From asktom.oracle.com

Hi Tom,

can you please provide a classification of streams and change data capture. I guess the main difference is that streams covers event capture, transport (transformation) and consumption. CDC only the capture. But if you consider only event capture, are there technical differences between streams and change data capture? What was the main reason to made CDC as a separate product?

Followup:

think of streams like a brick. think of CDC like a building made of brick. streams can be used to build CDC. CDC is built on top of streams (async CDC is anyway, sync CDC is trigger based). they are complimentary, not really competing.

Streams Objects and Owners

- SYS
- SYSTEM
- SYSMAN
- WMSYS
- PUBLIC
- IX

The same as AQ!

- Database Links
- Evaluation Contexts
- Function
- Indexes
- Packages & Package Bodies
- Queues
- Rule Sets
- Sequences
- Synonyms
- Tables
- Types & Type Bodies
- Views

CDC Concepts

- Logical Change Record (LCR)
- Implementation Choices
  - Synchronous
  - Asynchronous Capture
    - Hotlog
    - Autlog
- Change Sets
- Local & Downstream Capture
- Publish & Subscribe
- Propagation & Consumption
- Does not include:
  - Positive & Negative Rule Sets
  - Horizontal & Vertical Subsetting
  - Transformation

Logical Change Record (LCR)

- Logical change records, or LCRs, describe changes made to a single row of a table modified with a single DML statement
- A DML statement that operates on multiple rows within a table will generate multiple LCRs
  - For example an UPDATE or DELETE
- A transaction can consist of multiple DMLs
- Each LCR includes
  - name of the changed table
  - the old and new values for changed columns
  - values for the key columns
- Changes can be applied to the correct rows at the destinations and conflicts detected

Logical Change Records (LCRs)

- LCRs are created by the capture process
- The capture process reformats changes that are captured from the redo log into two types of LCRs:
  - Row LCRs
- The capture process implicitly enqueues LCRs
- CDC does not process DDL LCRs

Row LCRs

- A row LCR describes a DML change to the data in a single row or a LOB column in a row:
  - Single data change may yield multiple LCRs.
  - Each row LCR is an object of type **LCR$_ROW_RECORD**

```
STATIC FUNCTION CONSTRUCT(
source_database_name IN VARCHAR2,
command_type         IN VARCHAR2,
object_owner         IN VARCHAR2,
object_name          IN VARCHAR2,
tag                  IN RAW DEFAULT NULL,
transaction_id       IN VARCHAR2 DEFAULT NULL,
scn                  IN NUMBER DEFAULT NULL,
old_values           IN SYS.LCR$_ROW_LIST DEFAULT NULL,
new_values           IN SYS.LCR$_ROW_LIST DEFAULT NULL)
RETURN SYS.LCR$_ROW_RECORD;
```

Implementation

Multiple Implementation Choices

- Synchronous
- Asynchronous
  - HotLog
  - Distributed HotLog
  - AutoLog

	Sync CDC	Async HotLog	Async AutoLog
Available	Oracle 9*i* EE Oracle 10g SE	Oracle 10g EE	Oracle 10g EE
Source system cost	Transaction delay, system resources	System resources	Minimal (~2%)
Part of Txn	Yes	No	No
Latency	Real time	Near real time	Varies with topology, checkpoint & log switch interval
Systems	1	1	2

Synchronous Capture

- Synchronous
  - Uses triggers on the source database to capture change data. It has no latency because the change data is captured continuously and in real time on the source database. The change tables are populated when DML operations on the source table are committed
  - While the synchronous mode of Change Data Capture adds overhead to the source database at capture time, this mode can reduce costs (as compared to attempting to extract change data using table differencing or change-value section) by simplifying the extraction of change data

Asynchronous Capture

- Captures change data after the changes have been committed to the source database by using the database redo log files
  - HotLog
    Change data is captured from the online redo log file on the source database. There is a brief latency between the act of committing source table transactions and the arrival of change data
  - AutoLog
    Change data is captured from a set of redo log files managed by redo transport services. Redo transport services control the automated transfer of redo log files from the source database to the staging database

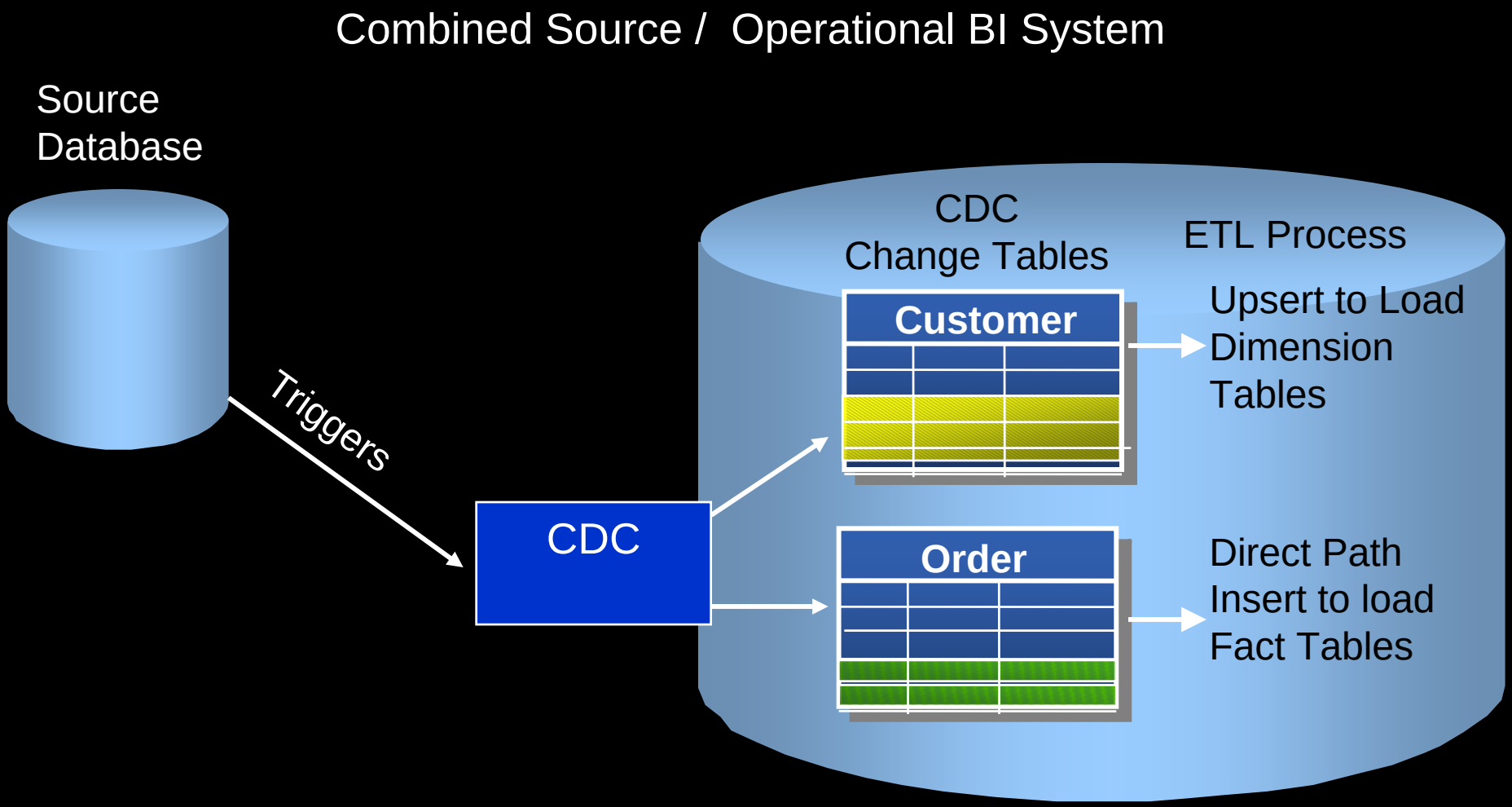
Choosing A Capture Configuration

Mode	Staging Location	Latency	Performance Impact
Synchronous	Must be the same as the source database	None	Adds overhead to source database transactions
Asynchronous HotLog	Must be the same as the source database	Populated as new committed transactions arrive	Minimal impact on source database for supplemental logging and capture
Asynchronous Distributed HotLog	Splits between Source and Staging Databases	Populated as new committed transactions arrive	Minimal impact on source database for supplemental
Asynchronous AutoLog	Typically remote from the source database	Depends on redo log switch frequency	When remote has the least impact on performance

How Sync CDC Works

- Uses internal triggers to capture before and/or after images of new and updated rows
- Has the same performance implications as capture via user triggers
- Delivers change data in real-time

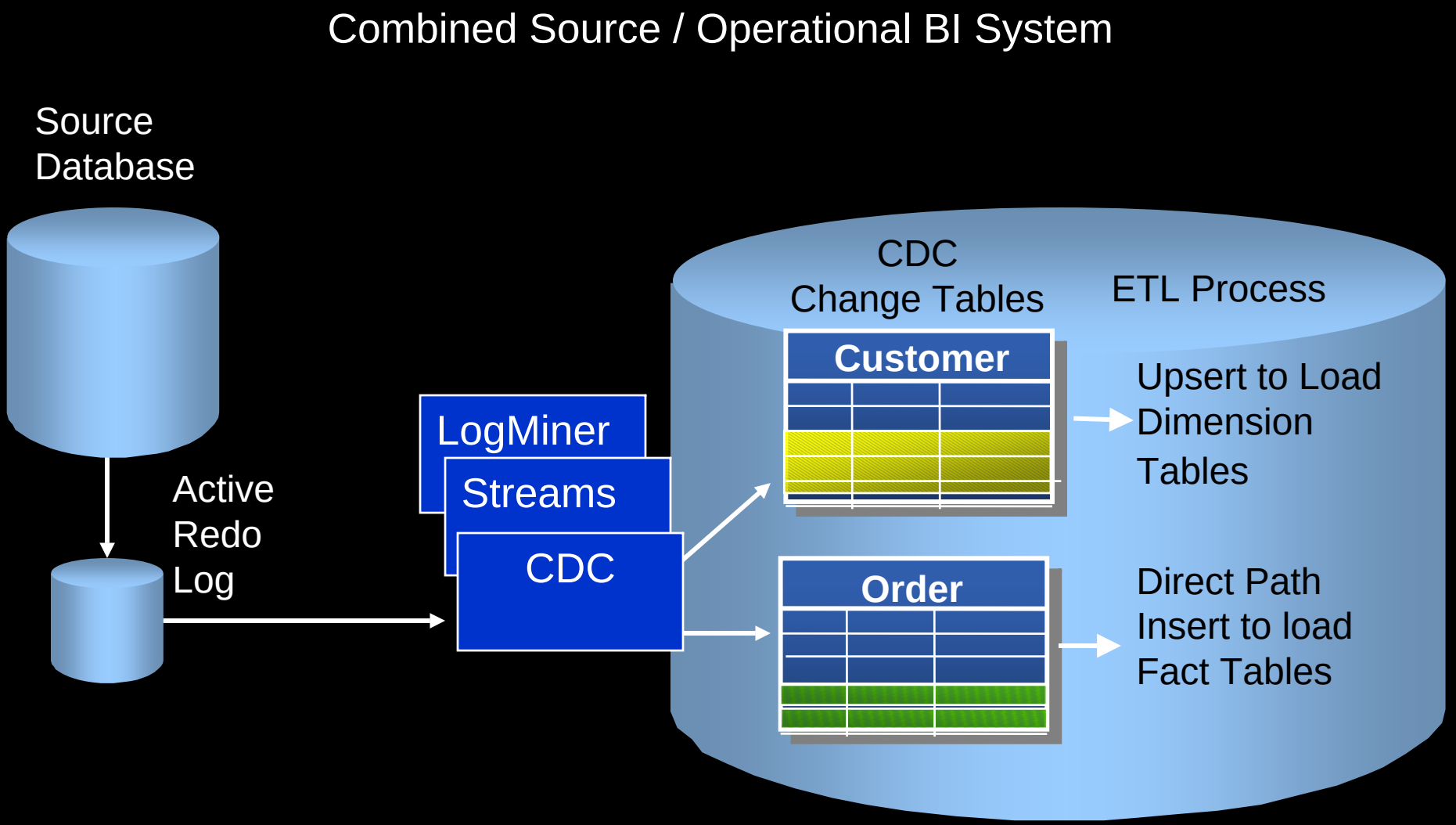
Synchronous CDC HotLog



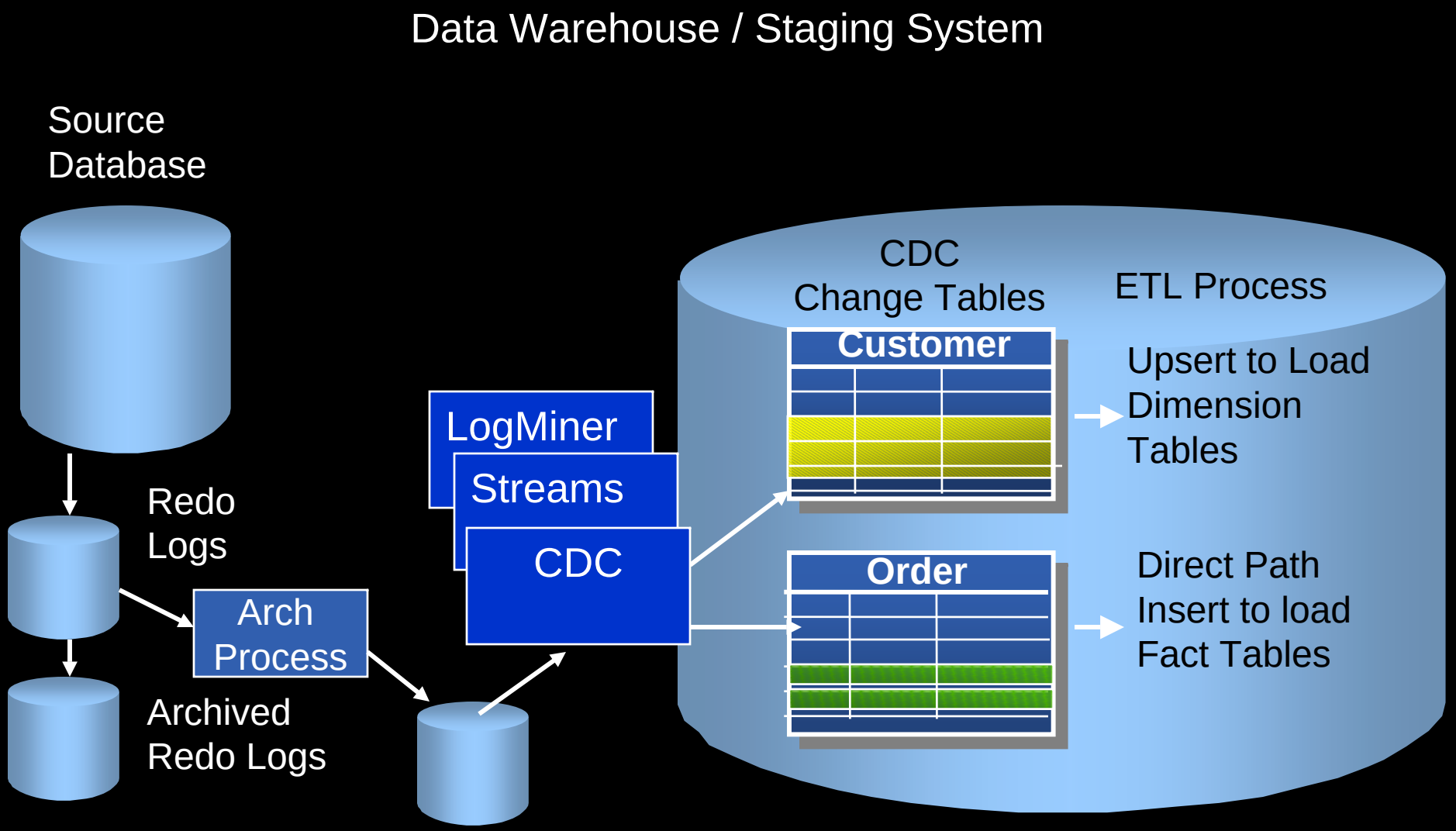
How Async CDC Works

- Relational interface to Streams
  - Prepackaged Streams application
  - Asynchronously captures change data from redo/archive logs
  - Presents relational interface to change data stream
- Can operate on source system (hot log) or staging system (auto log)

Async CDC HotLog



Async CDC AutoLog



Change Sets

- A logical grouping of change data that is guaranteed to be transactionally consistent and that can be managed as a unit
  - Synchronous Mode
    New change data arrives automatically as DML operations on the source tables are committed. Publishers can define new change sets in the predefined SYNC_SOURCE change source or use the predefined change set, SYNC_SET. The SYNC_SET change set cannot be altered or dropped.
  - Asynchronous & Distribution HotLog
    New change data arrives automatically, on a transaction-by-transaction basis from the current online redo log file. Publishers define change sets in the predefined HOTLOG_SOURCE change source.
  - Asynchronous AutoLog
    New change data arrives automatically, on a log-by-log basis, as log transport services makes redo log files available. Publishers define change sets in publisher-defined AutoLog change sources.

AutoLog Only

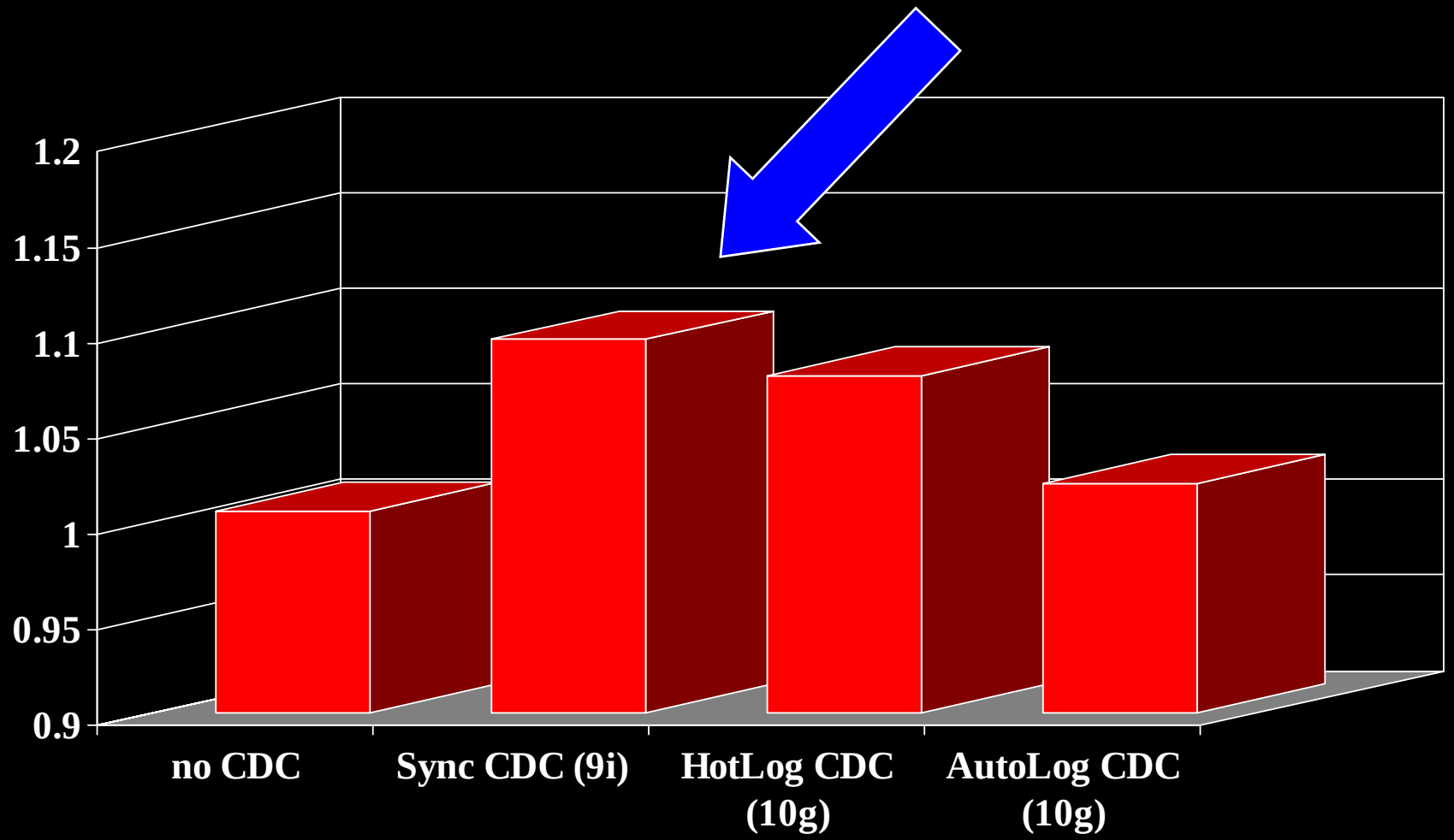
- REMOTE_LOGIN_PASSWORDFILE database initialization parameter must be set to SHARED on both the source and staging databases
- A password file must exist on both the machines hosting the source and staging databases

```
$ orapwd FILE=orapw PASSWORD=oracle1 ENTRIES=10
```

Performance

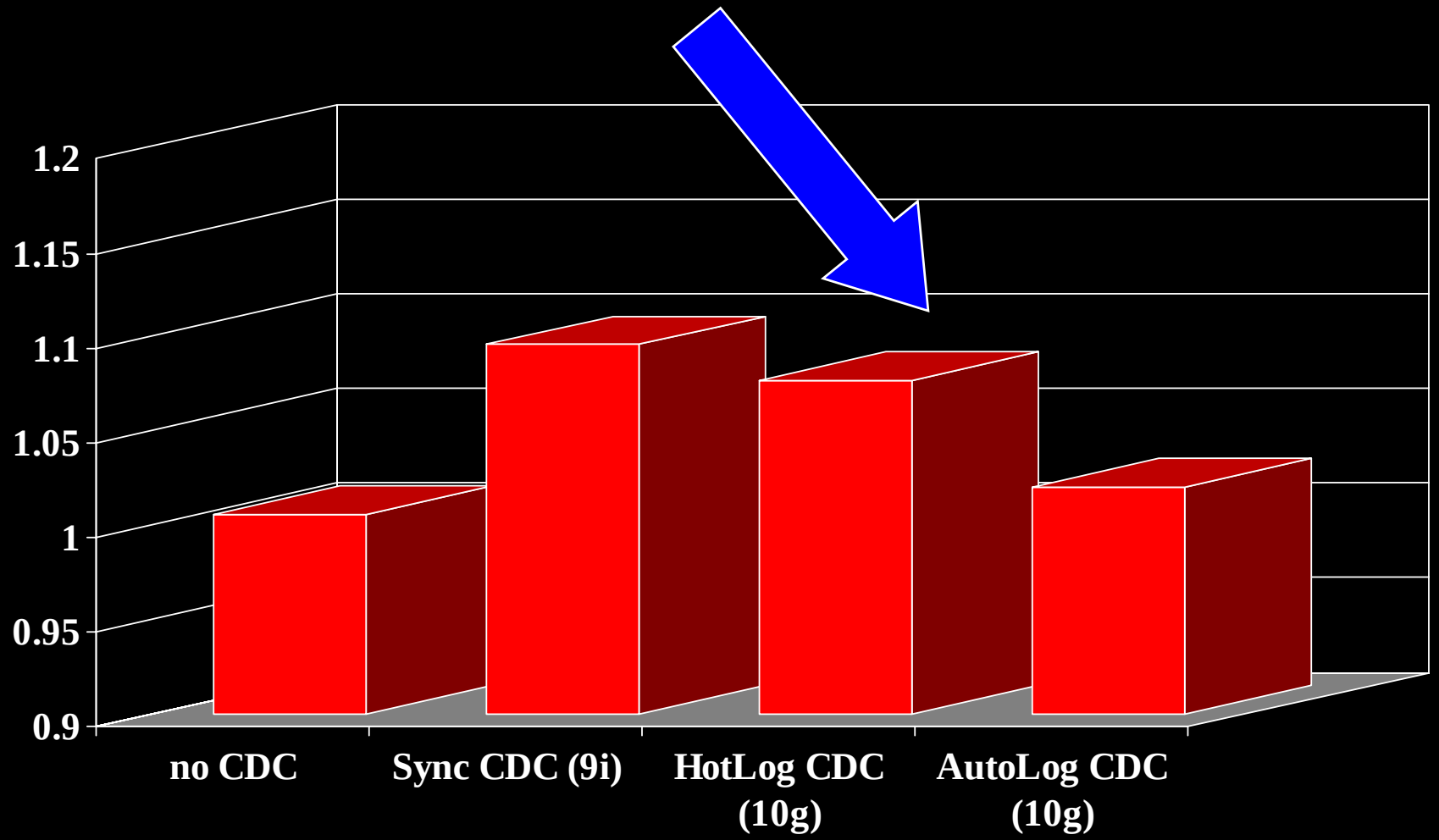
CDC Performance

Transaction elongated by 10%
Relative impact varies depending on other overhead



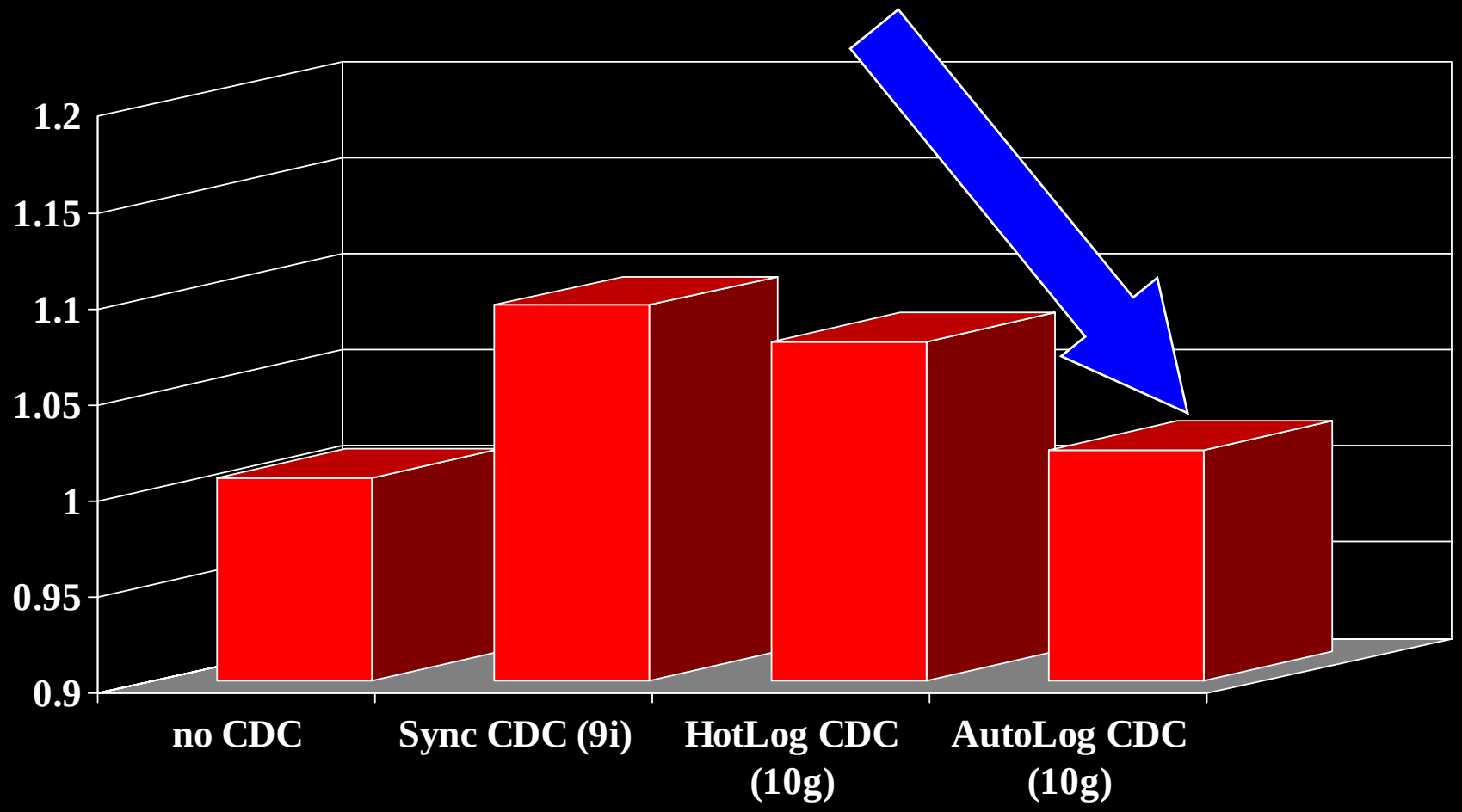
CDC Performance

Transaction elongated by 8%
Can reduce elongation by adding RAC nodes / CPUs

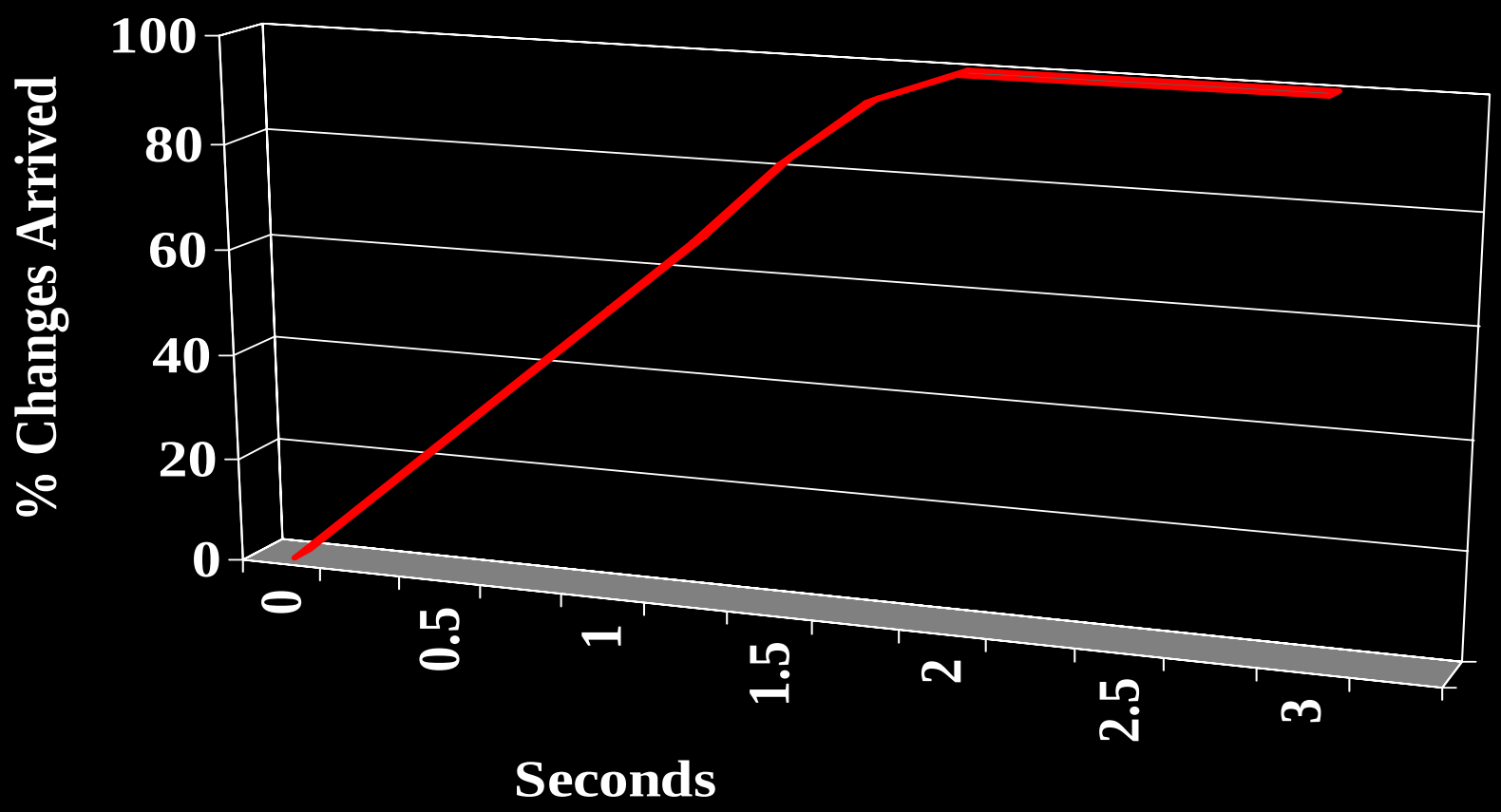


CDC Performance

Transaction elongation virtually eliminated
Change capture processing moved off system



HotLog Latency Performance



About ½ the change data arrived in 1 second
Virtually all the change data arrived in 2 seconds

Capture

Local Capture

- Oracle CDC supports capture of events
  - on the source database
  - on the staging database
- Implicit capture is used by Oracle Streams replication to automatically capture changes to a table. Typically, with implicit capture, the server captures DML and DDL events for a source database by mining the source redo logs and archive logs locally.
  - CDC sees DDL events but does not apply them
- Alternatively, the server can mine the redo logs or archived logs of the source database at an alternate database, assuming the alternative database is on a similar platform type and operating system.

Downstream Capture

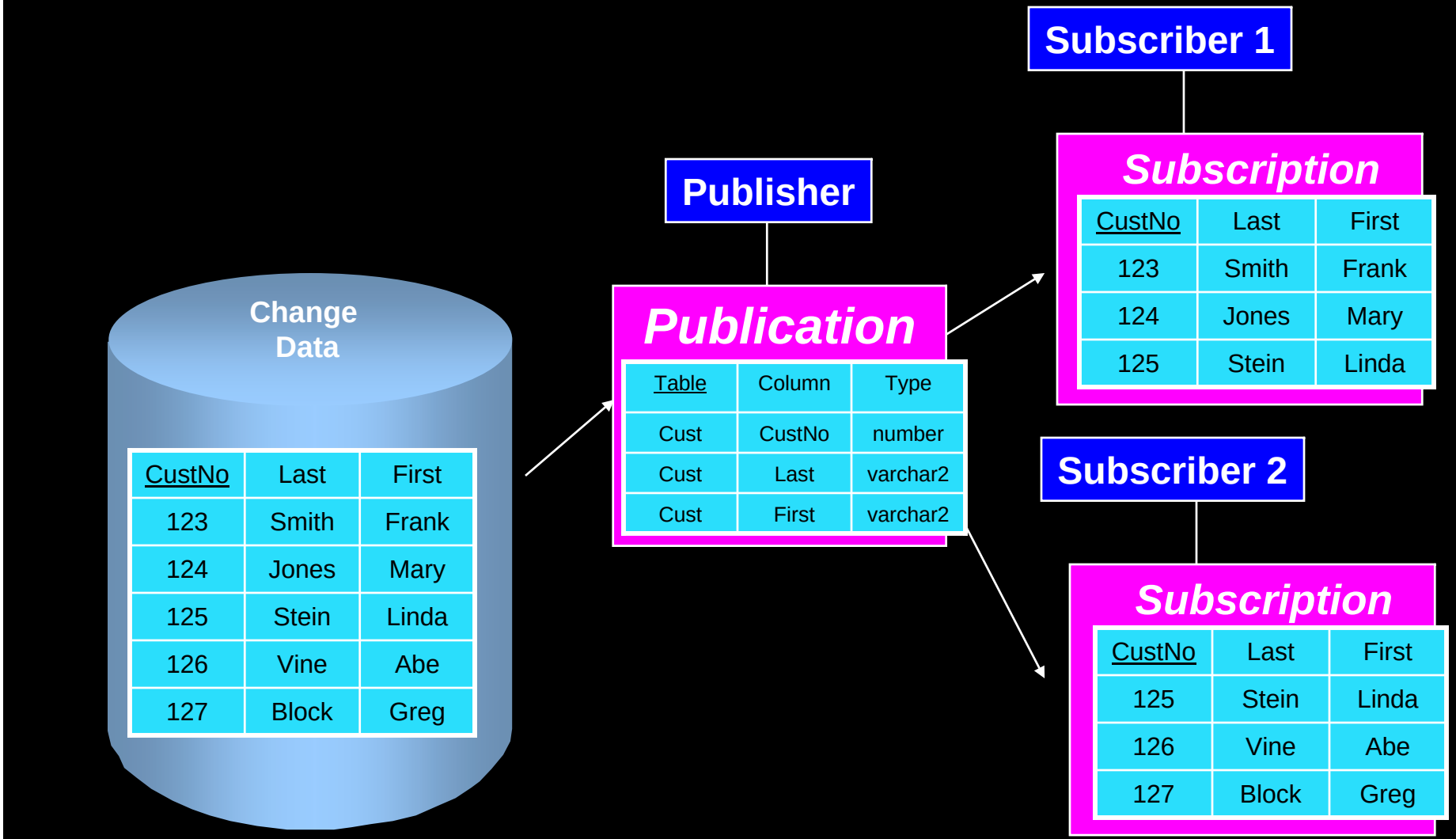
- Oracle also supports mining of archived redo logs at a downstream database. Although Oracle supports only one real-time downstream capture process at a downstream database, this real-time process can co-exist with multiple downstream archived-log processes.

Publish & Subscribe

CDC Publish & Subscribe

- Publisher supplies, subscribers consume change data
- Model allows sharing of change data across users and applications
- Coordinates retention / purge of change data
- Prevents application from accidentally processing change data more than once
- Guarantees transactional consistency of change data across source tables via change sets

Using CDC Publish & Subscribe



Publisher Concepts

- Change source
  - Defines the source system to CDC
- Change set
  - Collection of source tables for which transactionally consistent change data is needed
- Change table
  - Container to receive change data
  - Is published to subscribers

Publisher Package

- DBMS_CDC_PUBLISH
  - CREATE / ALTER / DROP_AUTOLOG_CHANGE_SOURCE
  - CREATE / ALTER / DROP_CHANGE_SET
  - CREATE / ALTER / DROP_CHANGE_TABLE
  - PURGE
  - PURGE_CHANGE_SET
  - PURGE_CHANGE_TABLE
  - DROP_SUBSCRIPTION

Publish

- Rule based publication based on AQ
- Publisher Views
  - change_sources
  - change_sets
  - change_tables
  - xxx_source_tables
  - xxx_published_columns
  - xxx_subscriptions
  - xxx_subscribed_tables
  - xxx_subscribed_columns

Note: xxx = dba, all, or user

Publisher Privileges & Roles

- For All Publishing
  - CREATE SESSION
  - CREATE TABLE
  - EXECUTE_CATALOG_ROLE
  - SELECT_CATALOG_ROLE
  - RESOURCE (which components?)
- For Asynchronous Publishing
  - CREATE SEQUENCE
  - DBA (which components?)
  - Grantee specified in:
    DBMS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE

Subscriber Concepts

- The subscriber creates a subscription from an available publication
- The subscription provides a moving window (view) to the change data
- Subscriptions go against a single change set and are transactionally consistent
- When all subscribers have advanced past old change data, it is automatically purged

Subscriber Package

- DBMS_CDC_SUBSCRIBE
  - CREATE_SUBSCRIPTION
  - SUBSCRIBE
  - ACTIVATE_SUBSCRIPTION
  - EXTEND_WINDOW (alter change set)
  - PURGE_WINDOW (alter change set)
  - DROP_SUBSCRIPTION

Subscribe

- Rule based subscription based on AQ
- Subscriber Views
  - xxx_source_tables
  - xxx_published_columns
  - xxx_subscriptions
  - xxx_subscribed_tables
  - xxx_subscribed_columns

Note: xxx = dba, all, or user

CDC Security

- Sync publisher must have SELECT access to the source table
- Async publisher must have EXECUTE_CATALOG_ROLE privilege
- Publisher uses GRANT and REVOKE on change tables to control subscriber access

Propagation

- Events in the staging area can be propagated to staging areas in other databases.
- To simplify network routing and reduce WAN traffic, events need not be transmitted to all databases and applications.
- Asynchronous Change Data Capture generates components of Oracle Streams to capture change data and to populate change sets. These components must not be reused or shared for any other purposes. For example, the capture queue that Change Data Capture generates for a Distributed HotLog change source should not be used as the source queue for new user-created Streams propagations.

Consumption

- Messages in a staging area are consumed by the apply engine, where the changes they represent are applied to a database, or they are consumed by an application.
- The Oracle Streams apply engine applies:
  - DML changes
  - DDL changes
- When the destination database is an Oracle database, the apply engine runs locally on the system hosting the Oracle database.

Labs

Streams Specific Parameters

- Change Data Capture (CDC)
  - Required for Asynchronous Hot-Log mode
    - Archivelog Mode
    - Forced Logging Mode
    - Supplemental Logging

```
SHOW PARAMETER aq_tm_processes (min 3)
SHOW PARAMETER compatible (10.1.0 or above)
SHOW PARAMETER global_name (TRUE)
SHOW PARAMETER job_queue_processes (min 2, 4-6 recommended)
SHOW PARAMETER java_pool_size (50000000)
SHOW PARAMETER open_links (note less than the default 4)
SHOW PARAMETER shared_pool_size (0 or at least 200MB)
SHOW PARAMETER streams_pool_size (min 150MB, 10MB/capt. 1MB/apply)
SHOW PARAMETER undo_retention (min. 3600 = 1 hr.)
                              * * *
ALTER SYSTEM SET aq_tm_processes=3 scope=BOTH;
ALTER SYSTEM SET compatible='10.2.0.1.0' scope=SPFILE;
```

Supplemental Logging

- Places additional column data into a redo log whenever an UPDATE operation is performed
- Can be specified at the database level or table level for primary key columns, unique index columns, foreign key columns, or all columns
- Can be enabled for specific non-key columns of a particular table using supplemental log groups
- Must be configured at the source site

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (
PRIMARY KEY, UNIQUE) COLUMNS;
```

Determining Enabled Supplemental Logging

- Database level:

```
SELECT SUPPLEMENTAL_LOG_DATA_MIN MIN,
       SUPPLEMENTAL_LOG_DATA_PK PK,
       SUPPLEMENTAL_LOG_DATA_UI UI,
       SUPPLEMENTAL_LOG_DATA_FK FK,
       SUPPLEMENTAL_LOG_DATA_ALL "ALL"
FROM GV$DATABASE;
```

- Table level:

```
SELECT owner, table_name, log_group_type
FROM DBA_LOG_GROUPS;
```

Additional HotLog & AutoLog Specific Params

```
SHOW PARAMETER PARALLEL_MAX_SERVERS
(current value) + (5 * (the number of change sets planned))

SHOW PARAMETER PROCESSES
(current value) + (7 * (the number of change sets planned))

SHOW PARAMETER SESSIONS
(current value) + (2 * (the number of change sets planned))
```

Create The CDC Administrator

```
CREATE USER cdcadmin
IDENTIFIED BY cdcadmin
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp
QUOTA 0 ON SYSTEM
QUOTA 0 ON SYSAUX
QUOTA 20M ON users;

GRANT create session TO cdcadmin;
GRANT create table TO cdcadmin;
GRANT create sequence TO cdcadmin;
GRANT create procedure TO cdcadmin;
GRANT select_catalog_role TO cdcadmin;
GRANT execute_catalog_role TO cdcadmin;
GRANT dba TO cdcadmin;

GRANT execute ON dbms_cdc_publish TO cdcadmin;
GRANT execute ON dbms_cdc_subscribe TO cdcadmin;

exec dbms_streams_auth.grant_admin_privilege(CDCADMIN');
```

Live CDC Demo

The demo can be found at:
www.psoug.org/reference/streams_demo2.html

Additional Oracle demos can be found at:
www.psoug.org/reference/

Oracle Support of CDC is Excellent

```
SQL> BEGIN
2

dbms_cdc_publish.create_change_table('CDCADMIN','CDC_DEMO_EMP_CT',
'CDC_DEMO_SET', 'HR', 'CDC_DEMO_EMPLOYEES', 'EMPLOYEE_ID NUMBER(6),
FIRST_NAME VARCHAR2(20), LAST_NAME VARCHAR2(25), EMAIL VARCHAR2(25),
PHONE_NUMBER VARCHAR2(20),HIRE_DATE DATE,JOB_ID VARCHAR2(10),
SALARY NUMBER(8,2), COMMISSION_PCT NUMBER(2,2), MANAGER_ID
NUMBER(6), DEPARTMENT_ID NUMBER(4)', 'both', 'Y', 'N', 'N', 'N', 'N', 'N', 'Y', NULL);
3 END;
4 /
BEGIN
*
ERROR at line 1:
ORA-00600: internal error code, arguments: [], [], [], [], [], [], [], []
ORA-06512: at "SYS.DBMS_CDC_PUBLISH", line 611
ORA-06512: at line 2
```

A clear case of user error: Mine!

CDC Summary

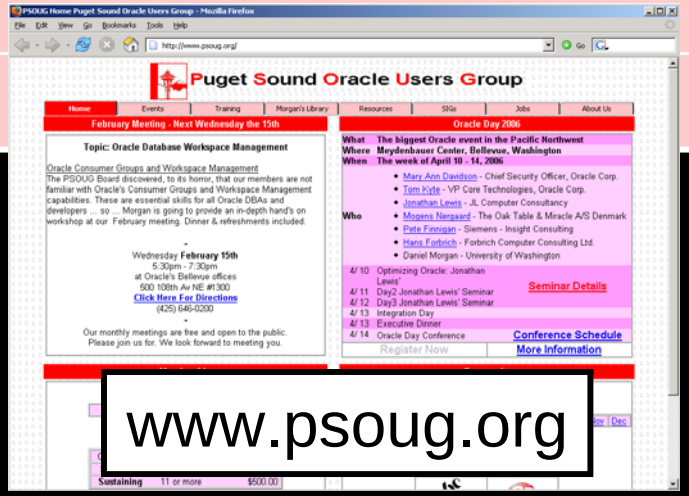
- CDC assumes the burden of change capture
- Change data is guaranteed consistent and complete
- Change data can be shared across users and applications
- CDC delivers change data where you need it, when you need it, and with minimal overhead

Recommended Resources

- Oracle Technology Network
  - http://otn.oracle.com
- Tahiti
  - http://tahiti.oracle.com
- Metalink
  - http://metalink.oracle.com
- Ask Tom
  - http://asktom.oracle.com
- Morgan's Library
  - www.psoug.org (AQ, CDC, Streams)

Shameless PSOUG Plug

AQ Topics	Streams Topics
DBMS_AQ	DBMS_CAPTURE_ADM
DBMS_AQADM	DBMS_CDC_PUBLISH
DBMS_AQELM	DBMS_CDC_SUBSCRIBE
DBMS_TRANSFORM	DBMS_STREAMS
	DBMS_STREAMS_ADM
	DBMS_STREAMS_AUTH



Thank you for your interest in CDC

For more information, or to provide feedback, please contact me at:

damorgan@u.washington.edu

All demo code is available at:

www.psoug.org

Questions