O'REILLY®

Streaming Change Data Capture

A Foundation for Modern Data Architectures

Kevin Petrie, Dan Potter
& Itamar Ankorion

Streaming Change Data Capture

A Foundation for Modern Data Architectures

Kevin Petrie, Dan Potter, and Itamar Ankorion

Beijing · Boston · Farnham · Sebastopol · Tokyo
O'REILLY®

Streaming Change Data Capture

by Kevin Petrie, Dan Potter, and Itamar Ankorion

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Rachel Roumeliotis
Production Editor: Justin Billing
Copyeditor: Octal Publishing, Inc.
Proofreader: Sharon Wilkey

Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Rebecca Demarest

May 2018: First Edition

Revision History for the First Edition

2018-04-25:	First Release
2018-06-08:	Second Release
2020-02-26:	Third Release

While the publisher and the authors have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the authors disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

This work is part of a collaboration between O'Reilly and Qlik. See our statement of editorial independence.

978-1-492-03249-6

[LSI]

Table of Contents

Acknowledgments

Experts more knowledgeable than we are helped to make this book happen. First, of course, are numerous enterprise customers in North America and Europe, with whom we have the privilege of collaborating, as well as Qlik's talented sales and presales organization. Ted Orme, VP of marketing and business development, proposed the idea for this book based on his conversations with many customers. Other valued contributors include Jordan Martz, Ola Mayer, Clive Bearman, and Melissa Kolodziej.

Prologue

There is no shortage of hyperbolic metaphors for the role of data in our modern economy—a tsunami, the new oil, and so on. From an IT perspective, data flows might best be viewed as the circulatory system of the modern enterprise. We believe the beating heart is *change data capture* (CDC) software, which identifies, copies, and sends live data to its various users.

Although many enterprises are modernizing their businesses by adopting CDC, there remains a dearth of information about how this critical technology works, why modern data integration needs it, and how leading enterprises are using it. This book seeks to close that gap. We hope it serves as a practical guide for enterprise architects, data managers, and CIOs as they build modern data architectures.

Generally, this book focuses on structured data, which, loosely speaking, refers to data that is highly organized; for example, using the rows and columns of relational databases for easy querying, searching, and retrieval. This includes data from the Internet of Things (IoT) and social media sources that is collected into structured repositories.

Introduction: The Rise of Modern Data Architectures

Data is creating massive waves of change and giving rise to a new data-driven economy that is only beginning. Organizations in all industries are changing their business models to monetize data, understanding that doing so is critical to competition and even survival. There is tremendous opportunity as applications, instrumented devices, and web traffic are throwing off reams of 1s and 0s, rich in analytics potential.

These analytics initiatives can reshape sales, operations, and strategy on many fronts. Real-time processing of customer data can create new revenue opportunities. Tracking devices with Internet of Things (IoT) sensors can improve operational efficiency, reduce risk, and yield new analytics insights. New artificial intelligence (AI) approaches such as machine learning can accelerate and improve the accuracy of business predictions. Such is the promise of modern analytics.

However, these opportunities change how data needs to be moved, stored, processed, and analyzed, and it's easy to underestimate the resulting organizational and technical challenges. From a technology perspective, to achieve the promise of analytics, underlying data architectures need to efficiently process high volumes of fast-moving data from many sources. They also need to accommodate evolving business needs and multiplying data sources.

To adapt, IT organizations are embracing *data lake*, *streaming*, and *cloud architectures*. These platforms are complementing and even replacing the enterprise data warehouse (EDW), the traditional structured system of record for analytics. Figure I-1 summarizes these shifts.

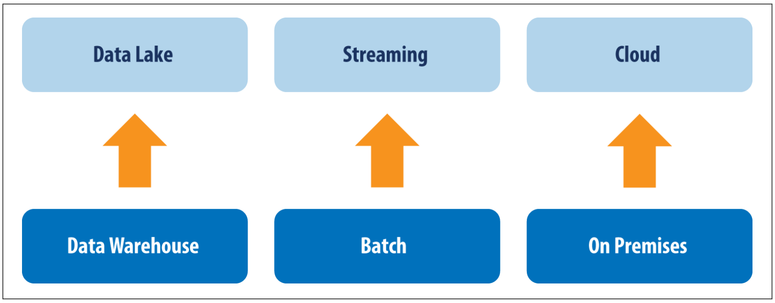


Figure I-1. Key technology shifts

Enterprise architects and other data managers know firsthand that we are in the early phases of this transition, and it is tricky stuff. A primary challenge is data integration—the second most likely barrier to Hadoop Data Lake implementations, right behind data governance, according to a recent TDWI survey (*source:* "Data Lakes: Purposes, Practices, Patterns and Platforms," TDWI, 2017). IT organizations must copy data to analytics platforms, often continuously, without disrupting production applications (a trait known as *zero-impact*). Data integration processes must be scalable, efficient, and able to absorb high data volumes from many sources without a prohibitive increase in labor or complexity. Table I-1 summarizes the key data integration requirements of modern analytics initiatives.

Table I-1. Data integration requirements of modern analytics

Analytics initiative	Requirement
AI (e.g., machine learning), IoT	**Scale:** Use data from thousands of sources with minimal development resources and impact
Streaming analytics	**Real-time transfer:** Create real-time streams from database transactions
Cloud analytics	**Efficiency:** Transfer large data volumes from multiple datacenters over limited network bandwidth
Agile deployment	**Self-service:** Enable nondevelopers to rapidly deploy solutions
Diverse analytics platforms	**Flexibility:** Easily adopt and adapt new platforms and methods

All this entails careful planning and new technologies because traditional batch-oriented data integration tools do not meet these requirements. Batch replication jobs and manual extract, transform, and load (ETL) scripting procedures are slow, inefficient, and disruptive. They disrupt production, tie up talented ETL programmers, and create network and processing bottlenecks. They cannot scale sufficiently to support strategic enterprise initiatives. Batch is unsustainable in today's enterprise.

Enter Change Data Capture

A foundational technology for modernizing your environment is change data capture (CDC) software, which enables continuous incremental replication by identifying and copying data updates as they take place. When designed and implemented effectively, CDC can meet today's scalability, efficiency, real-time, and zero-impact requirements.

Without CDC, organizations usually fail to meet modern analytics requirements. They must stop or slow production activities for batch runs, hurting efficiency and decreasing business opportunities. They cannot integrate enough data, fast enough, to meet analytics objectives. They lose business opportunities, lose customers, and break operational budgets.

CHAPTER 1

Why Use Change Data Capture?

Change data capture (CDC) continuously identifies and captures incremental changes to data and data structures (aka schemas) from a source such as a production database. CDC arose two decades ago to help replication software deliver real-time transactions to data warehouses, where the data is then transformed and delivered to analytics applications. Thus, CDC enables efficient, low-latency data transfer to operational and analytics users with low production impact.

Let's walk through the business motivations for a common use of replication: offloading analytics queries from production applications and servers. At the most basic level, organizations need to do two things with data:

- Record what's happening to the business—sales, expenditures, hiring, and so on.
- Analyze what's happening to assist decisions—which customers to target, which costs to cut, and so forth—by querying records.

The same database typically cannot support both of these requirements for transaction-intensive enterprise applications, because the underlying server has only so much CPU processing power available. It is not acceptable for an analytics query to slow down production workloads such as the processing of online sales transactions. Hence the need to analyze copies of production records on a different platform. The business case for offloading queries is to both record business data and analyze it, without one action interfering with the other.

The first method used for replicating production records (i.e., rows in a database table) to an analytics platform is *batch loading*, also known as *bulk* or *full loading*. This process creates files or tables at the target, defines their "metadata" structures based on the source, and populates them with data copied from the source as well as the necessary metadata definitions.

Batch loads and periodic reloads with the latest data take time and often consume significant processing power on the source system. This means administrators need to run replication loads during "batch windows" of time in which production is paused or will not be heavily affected. Batch windows are increasingly unacceptable in today's global, 24×7 business environment.

The Role of Metadata in CDC

Metadata is data that describes data. In the context of replication and CDC, primary categories and examples of metadata include the following:

Infrastructure
Servers, sources, targets, processes, and resources

Users
Usernames, roles, and access controls

Logical structures
Schemas, tables, versions, and data profiles

Data instances
Files and batches

Metadata plays a critical role in traditional and modern data architectures. By describing datasets, metadata enables IT organizations to discover, structure, extract, load, transform, analyze, and secure the data itself. Replication processes, be they either batch load or CDC, must be able to reliably copy metadata between repositories.

Here are real examples of enterprise struggles with batch loads (in Chapter 4, we examine how organizations are using CDC to eliminate struggles like these and realize new business value):

- A Fortune 25 telecommunications firm was unable to extract data from SAP ERP and PeopleSoft fast enough to its data lake. Laborious, multitier loading processes created day-long delays that interfered with financial reporting.
- A Fortune 100 food company ran nightly batch jobs that failed to reconcile orders and production line-items on time, slowing plant schedules and preventing accurate sales reports.
- One of the world's largest payment processors was losing margin on every transaction because it was unable to assess customer-creditworthiness in-house in a timely fashion. Instead, it had to pay an outside agency.
- A major European insurance company was losing customers due to delays in its retrieval of account information.

Each of these companies eliminated their bottlenecks by replacing batch replication with CDC. They streamlined, accelerated, and increased the scale of their data initiatives while minimizing impact on production operations.

Advantages of CDC

CDC has three fundamental advantages over batch replication:

- It enables faster and more accurate decisions based on the most current data; for example, by feeding database transactions to streaming analytics applications.
- It minimizes disruptions to production workloads.
- It reduces the cost of transferring data over the wide area network (WAN) by sending only incremental changes.

Together these advantages enable IT organizations to meet the real-time, efficiency, scalability, and low-production impact requirements of a modern data architecture. Let's explore each of these in turn.

Faster and More Accurate Decisions

The most salient advantage of CDC is its ability to support real-time analytics and thereby capitalize on data value that is perishable. It's not difficult to envision ways in which real-time data updates, sometimes referred to as *fast data*, can improve the bottom line.

For example, business events create data with perishable business value. When someone buys something in a store, there is a limited time to notify their smartphone of a great deal on a related product in that store. When a customer logs into a vendor's website, this creates a short-lived opportunity to cross-sell to them, upsell to them, or measure their satisfaction. These events often merit quick analysis and action.

In a 2017 study titled *The Half Life of Data*, Nucleus Research analyzed more than 50 analytics case studies and plotted the value of data over time for three types of decisions: tactical, operational, and strategic. Although mileage varied by example, the aggregate findings are striking:

- Data used for tactical decisions, defined as decisions that prioritize daily tasks and activities, on average lost more than half its value 30 minutes after its creation. Value here is measured by the portion of decisions enabled, meaning that data more than 30 minutes old contributed to 70% fewer operational decisions than fresher data. Marketing, sales, and operations personnel make these types of decisions using custom dashboards or embedded analytics capabilities within customer relationship management (CRM) and/or supply-chain management (SCM) applications.
- Operational data on average lost about half its value after eight hours. Examples of operational decisions, usually made over a few weeks, include improvements to customer service, inventory stocking, and overall organizational efficiency, based on data visualization applications or Microsoft Excel.
- Data used for strategic decisions has the longest-range implications, but still loses half its value roughly 56 hours after creation (a little less than two and a half days). In the strategic category, data scientists and other specialized analysts often are assessing new market opportunities and significant potential changes to the business, using a variety of advanced statistical tools and methods.

Figure 1-1 plots Nucleus Research's findings. The Y axis shows the value of data to decision making, and the X axis shows the hours after its creation.

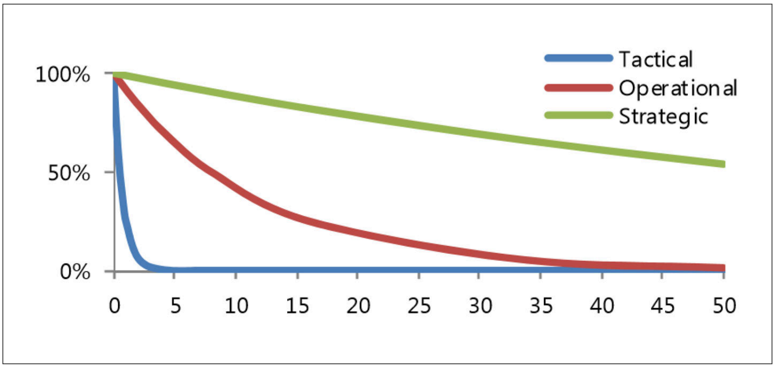


Figure 1-1. The sharply decreasing value of data over time (source: The Half Life of Data, Nucleus Research, January 2017)

Examples bring research findings like this to life. Consider the case of a leading European payments processor, which we'll call U Pay. It handles millions of mobile, online and in-store transactions daily for hundreds of thousands of merchants in more than 100 countries. Part of U Pay's value to merchants is that it credit-checks each transaction as it happens. But loading data in batch to the underlying data lake with Sqoop, an open source ingestion scripting tool for Hadoop, created damaging bottlenecks. The company could not integrate both the transactions from its production SQL Server and Oracle systems and credit agency communications fast enough to meet merchant demands.

U Pay decided to replace Sqoop with CDC, and everything changed. The company was able to transact its business much more rapidly and bring the credit checks in house. U Pay created a new automated decision engine that assesses the risk on every transaction on a near-real-time basis by analyzing its own extensive customer information. By eliminating the third-party agency, U Pay increased margins and improved service-level agreements (SLAs) for merchants.

Indeed, CDC is fueling more and more software-driven decisions. Machine learning algorithms, an example of artificial intelligence (AI), teach themselves as they process continuously changing data. Machine learning practitioners need to test and score multiple, evolving models against one another to generate the best results, which often requires frequent sampling and adjustment of the underlying datasets. This can be part of larger cognitive systems that also apply deep learning, natural-language processing (NLP), and other advanced capabilities to understand text, audio, video, and other alternative data formats.

Minimizing Disruptions to Production

By sending incremental source updates to analytics targets, CDC can keep targets continuously current without batch loads that disrupt production operations. This is critical because it makes replication more feasible for a variety of use cases. Your analytics team might be willing to wait for the next nightly batch load to run its queries (although that's increasingly less common). But even then, companies cannot stop their 24×7 production databases for a batch job. Kill the batch window with CDC and you keep production running full-time. You also can scale more easily and efficiently carry out high-volume data transfers to analytics targets.

Reducing WAN Transfer Cost

Cloud data transfers have in many cases become costly and time-consuming bottlenecks for the simple reason that data growth has outpaced the bandwidth and economics of internet transmission lines. Loading and repeatedly reloading data from on-premises systems to the cloud can be prohibitively slow and costly.

"It takes more than two days to move a terabyte of data across a relatively speedy T3 line (20 GB/hour)," according to Wayne Eckerson and Stephen Smith in their Qlik-commissioned report "Seven Considerations When Building a Data Warehouse Environment in the Cloud" (April 2017). They elaborate:

> And that assumes no service interruptions, which might require a full or partial restart…Before loading data, administrators need to compare estimated data volumes against network bandwidth to ascertain the time required to transfer data to the cloud. In most cases, it will make sense to use a replication tool with built-in change data capture (CDC) to transfer only deltas to source systems. This reduces the impact on network traffic and minimizes outages or delays.

In summary, CDC helps modernize data environments by enabling faster and more accurate decisions, minimizing disruptions to production, and reducing cloud migration costs. An increasing number of organizations are turning to CDC, both as a foundation of replication platforms such as Qlik Replicate (formerly Attunity Replicate) and as a feature of broader extract, transform, and load (ETL) offerings such as Microsoft SQL Server Integration Services (SSIS). It uses CDC to meet the modern data architectural requirements of real-time data transfer, efficiency, scalability, and zero-production impact. In Chapter 2, we explore the mechanics of how that happens.

CHAPTER 2
How Change Data Capture Works

Change data capture (CDC) identifies and captures just the most recent production data and metadata changes that the source has registered during a given time period, typically measured in seconds or minutes, and then enables replication software to copy those changes to a separate data repository. A variety of technical mechanisms enable CDC to minimize time and overhead in the manner most suited to the type of analytics or application it supports. CDC can accompany batch load replication to ensure that the target is and remains synchronized with the source upon load completion. Like batch loads, CDC helps replication software copy data from one source to one target, or one source to multiple targets. CDC also identifies and replicates changes to source schema (that is, data definition language [DDL]) changes, enabling targets to dynamically adapt to structural updates. This eliminates the risk that other data management and analytics processes become brittle and require time-consuming manual updates.

Source, Target, and Data Types

Traditional CDC sources include operational databases, applications, and mainframe systems, most of which maintain transaction logs that are easily accessed by CDC. More recently, these traditional repositories serve as landing zones for new types of data created by Internet of Things (IoT) sensors, social media message streams, and other data-emitting technologies.

Targets, meanwhile, commonly include not just traditional structured data warehouses, but also data lakes based on distributions from Hortonworks, Cloudera, or MapR. Targets also include cloud platforms such as Elastic MapReduce (EMR) and Amazon Simple Storage Service (S3) from Amazon Web Services (AWS), Microsoft Azure Data Lake Store, and Azure HDInsight. In addition, message streaming platforms (e.g., open source Apache Kafka and Kafka variants like Amazon Kinesis and Azure Event Hubs) are used both to enable streaming analytics applications and to transmit to various big data targets.

CDC has evolved to become a critical building block of modern data architectures. As explained in Chapter 1, CDC identifies and captures the data and metadata changes that were committed to a source during the latest time period, typically seconds or minutes. This enables replication software to copy and commit these incremental source database updates to a target. Figure 2-1 offers a simplified view of CDC's role in modern data analytics architectures.

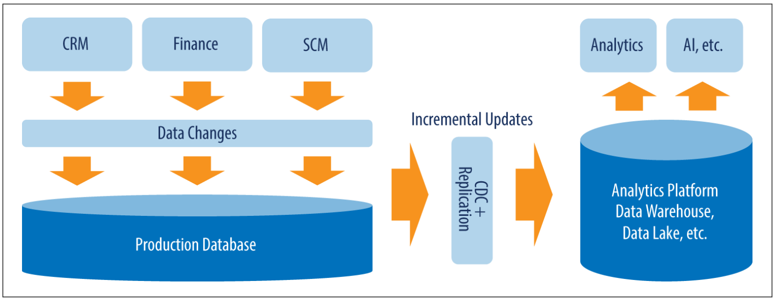


Figure 2-1. How change data capture works with analytics

NOTE

CDC is distinct from replication. However, in most cases it has become a feature of replication software. For simplicity, from here onward we will include replication when we refer to CDC.

So, what are these incremental data changes? There are four primary categories of changes to a source database: row changes such as *inserts*, *updates*, and *deletes*, as well as metadata (DDL) changes:

Inserts

These add one or more rows to a database. For example, a new row, also known as a *record*, might summarize the time, date, amount, and customer name for a recent sales transaction.

Updates

These change fields in one or more existing rows; for example, to correct an address in a customer transaction record.

Deletes

Deletes eliminate one or more rows; for example, when an incorrect sales transaction is erased.

DDL

Changes to the database's structure using its data definition language (DDL) create, modify, and remove database objects such as tables, columns, and data types, all of which fall under the category of *metadata* (see the definition in "The Role of Metadata in CDC" on page 2).

In a given update window, such as one minute, a production enterprise database might commit thousands or more individual inserts, updates, and deletes. DDL changes are less frequent but still must be accommodated rapidly on an ongoing basis. The rest of this chapter refers simply to row changes.

The key technologies behind CDC fall into two categories: identifying data changes and delivering data to the target used for analytics. The next few sections explore the options, using variations on Figure 2-2 as a reference point.

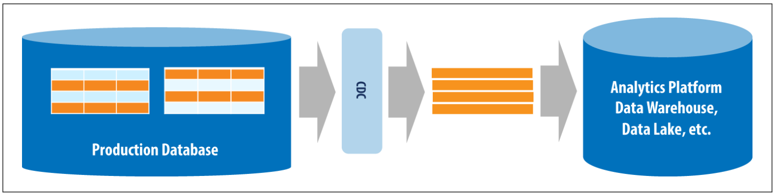


Figure 2-2. CDC example: row changes (one row = one record)

There are two primary architectural options for CDC: agent-based and agentless. As the name suggests, *agent-based* CDC software resides on the source server itself and therefore interacts directly with the production database to identify and capture changes. CDC agents are not ideal because they direct CPU, memory, and storage away from source production workloads, thereby degrading performance. Agents are also sometimes required on target end points, where they have a similar impact on management burden and performance.

The more modern, *agentless* architecture has zero footprint on source or target. Rather, the CDC software interacts with source and target from a separate intermediate server. This enables organizations to minimize source impact and improve ease of use.

Not All CDC Approaches Are Created Equal

There are several technology approaches to achieving CDC, some significantly more beneficial than others. The three approaches are *triggers*, *queries*, and *log readers*:

Triggers
: These log transaction events in an additional "shadow" table that can be "played back" to copy those events to the target on a regular basis (Figure 2-3). Even though triggers enable the necessary updates from source to target, firing the trigger and storing row changes in the shadow table increases processing overhead and can slow source production operations.

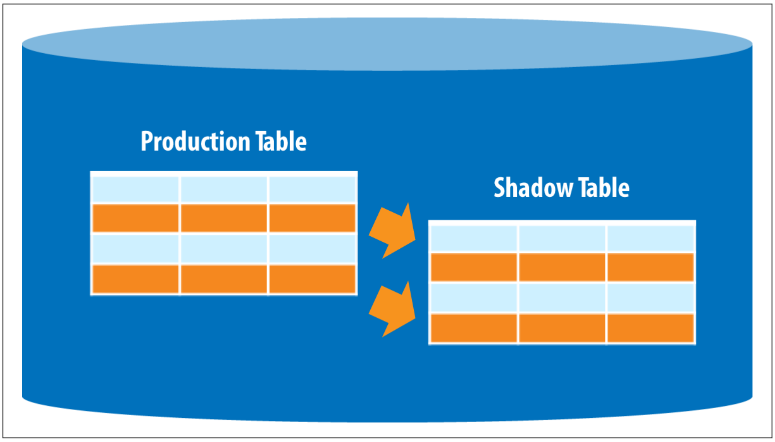


Figure 2-3. Triggers copy changes to shadow tables

Query-based CDC
: This approach regularly checks the production database for changes. This method can also slow production performance by consuming source CPU cycles. Certain source databases and

data warehouses, such as Teradata, do not have change logs (described in the next section) and therefore require alternative CDC methods such as queries. You can identify changes by using timestamps, version numbers, and/or status columns as follows:

- Timestamps in a dedicated source table column can record the time of the most recent update, thereby flagging any row containing data more recent than the last CDC replication task. To use this query method, all of the tables must be altered to include timestamps, and administrators must ensure that they accurately represent time zones.
- Version numbers increase by one increment with each change to a table. They are similar to timestamps, except that they identify the version number of each row rather than the time of the last change. This method requires a means of identifying the latest version; for example, in a supporting reference table and comparing it to the version column.
- Status indicators take a similar approach, as well, stating in a dedicated column whether a given row has been updated since the last replication. These indicators also might indicate that, although a row has been updated, it is not ready to be copied; for example, because the entry needs human validation.

Log readers

Log readers identify new transactions by scanning changes in transaction log files that already exist for backup and recovery purposes (Figure 2-4). Log readers are the fastest and least disruptive of the CDC options because they require no additional modifications to existing databases or applications and do not weigh down production systems with query loads. A leading example of this approach is Qlik Replicate. Log readers must carefully integrate with each source database's distinct processes, such as those that log and store changes, apply inserts/updates/deletes, and so on. Different databases can have different and often proprietary, undocumented formats, underscoring the need for deep understanding of the various databases and careful integration by the CDC vendor.

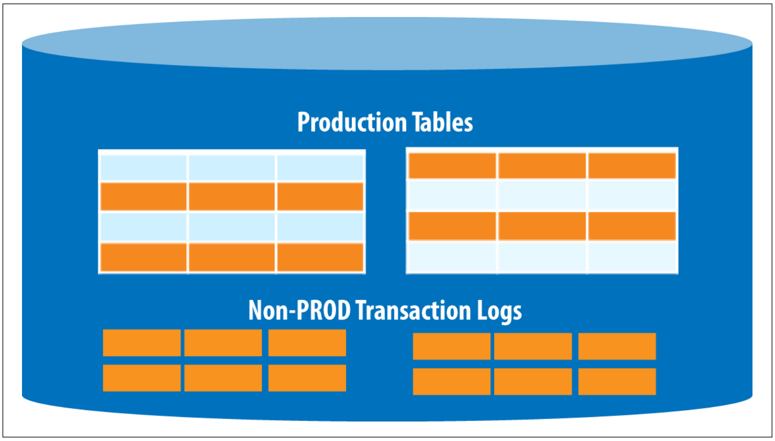


Figure 2-4. Log readers identify changes in backup and recovery logs

Table 2-1 summarizes the functionality and production impact of trigger, query, and log-based CDC.

Table 2-1. Functionality and production impact of CDC methods delivering data

CDC capture method	Description	Production impact
Log reader	Identifies changes by scanning backup/recovery transaction logs Preferred method when log access is available	**Minimal**
Query	Flags new transaction in production table column with timestamps, version numbers, and so on CDC engine periodically asks production database for updates; for example, for Teradata	**Low**
Trigger	Source transactions "trigger" copies to change-capture table Preferred method if no access to transaction logs	**Medium**

CDC can use several methods to deliver replicated data to its target:

Transactional

CDC copies data updates—also known as *transactions*—in the same sequence in which they were applied to the source. This method is appropriate when sequential integrity is more important to the analytics user than ultra-high performance. For example, daily financial reports need to reflect all completed transactions as of a specific point in time, so transactional CDC would be appropriate here.

Aggregated (also known as batch-optimized)

CDC bundles multiple source updates and sends them together to the target. This facilitates the processing of high volumes of transactions when performance is more important than sequential integrity on the target. This method can support, for example, aggregate trend analysis based on the most data points possible. Aggregated CDC also integrates with certain target data warehouses' native utilities to apply updates.

Stream-optimized

CDC replicates source updates into a message stream that is managed by streaming platforms such as Kafka, Azure Event Hubs, MapR-ES, or Amazon Kinesis. Unlike the other methods, stream-optimized CDC means that targets manage data in motion rather than data at rest. Streaming supports a variety of new use cases, including real-time location-based customer offers and analysis of continuous stock-trading data. Many organizations are beginning to apply machine learning to such streaming use cases.

The Role of CDC in Data Preparation

CDC is one part of a larger process to prepare data for analytics, which spans data sourcing and transfer, transformation and enrichment (including data quality and cleansing), and governance and stewardship. We can view these interdependent phases as a sequence of overlapping, sometimes looping steps. CDC, not surprisingly, is part of the sourcing and transfer phase, although it also can help enrich data and keep governance systems and metadata in sync across enterprise environments.

Principles of Data Wrangling by Tye Rattenbury et al. (O'Reilly, 2017) offers a useful framework (shown in Figure 2-5) to understand data workflow across three stages: *raw*, *refined*, and *production*. Let's briefly consider each of these in turn:

Raw stage
: During this stage, data is ingested into a target platform (via full load or CDC) and metadata is created to describe its characteristics (i.e., its structure, granularity, accuracy, temporality, and scope) and therefore its value to the analytics process.

Refined stage
: This stage puts data into the right structure for analytics and "cleanses" it by detecting—and correcting or removing—corrupt or inaccurate records. Most data ends up in the refined stage. Here analysts can generate ad hoc business intelligence (BI) reports to answer specific questions about the past or present, using traditional BI or visualization tools like Tableau. They also can explore and model future outcomes based on assessments of relevant factors and their associated historical data. This can involve more advanced methods such as machine learning or other artificial intelligence (AI) approaches.

Production stage
: In this stage, automated reporting processes guide decisions and resource allocation on a consistent, repeatable basis. This requires optimizing data for specific uses such as weekly supply-chain or production reports, which might in turn drive automated resource allocation.

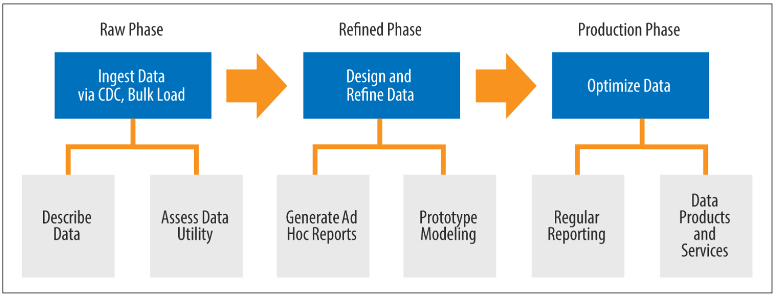


Figure 2-5. Data preparation workflow (adapted from Principles of Data Wrangling, O'Reilly, 2017)

Between each of these phases, we need to transform data into the right form. Change data capture plays an integral role by accelerating ingestion in the raw phase. This helps improve the timeliness and accuracy of data and metadata in the subsequent Design/Refine and Optimize phases.

The Role of Change Data Capture in Data Pipelines

A more modern concept that is closely related to data workflow is the *data pipeline*, which moves data from production source to analytics target through a sequence of stages, each of which refines data a little further to prepare it for analytics. Data pipelines often include a mix of data lakes, operational data stores, and data warehouses, depending on enterprise requirements.

For example, a large automotive parts dealer is designing a data pipeline with four phases, starting with an Amazon Simple Storage Service (S3) data lake, where data is landed via CDC. The company ingests 3,500 customer-specific instances of each production table in raw form via CDC in the first phase, then cleanses and merges those into single multitenant tables (still tracking change history) in the next phase. Pipeline phase 3 provides three views, including an operational data store (ODS), snapshot in time, and/or a full transaction history, and phase 4 will present the data for analytics via Tableau on top of a Redshift or Vertica data warehouse. By having different datasets in each phase, this company will be able to rewind any of its refinement changes to troubleshoot or tune the process as needed.

As with other example environments, CDC accelerates and streamlines data pipelines by efficiently feeding data into the first phase. It maintains production uptime, accelerates data refinement, and increases the scale of data that can be processed with available resources.

CHAPTER 3

How Change Data Capture Fits into Modern Architectures

Now let's examine the architectures in which data workflow and analysis take place, their role in the modern enterprise, and their integration points with change data capture (CDC). As shown in Figure 3-1, the methods and terminology for data transfer tend to vary by target. Even though the transfer of data and metadata into a database involves simple replication, a more complicated extract, transform, and load (ETL) process is required for data warehouse targets. Data lakes, meanwhile, typically can ingest data in its native format. Finally, streaming targets require source data to be published and consumed in a series of messages. Any of these four target types can reside on-premises, in the cloud, or in a hybrid combination of the two.

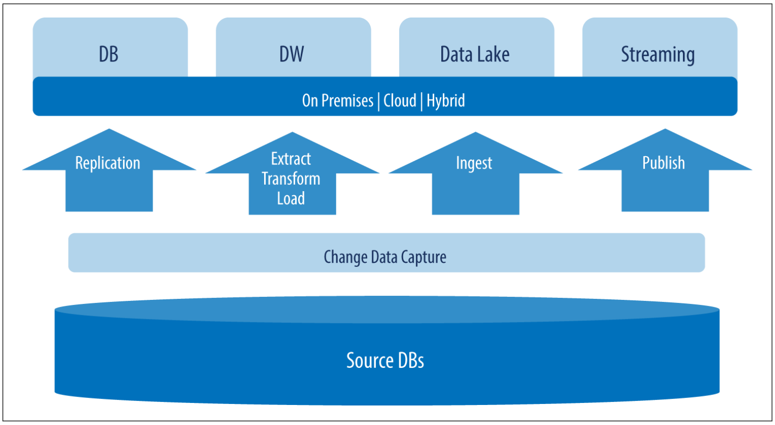


Figure 3-1. CDC and modern data integration

We will explore the role of CDC in five modern architectural scenarios: replication to databases, ETL to data warehouses, ingestion to data lakes, publication to streaming systems, and transfer across hybrid cloud infrastructures. In practice, most enterprises have a patchwork of the architectures described here, as they apply different engines to different workloads. A trial-and-error learning process, changing business requirements, and the rise of new platforms all mean that data managers will need to keep copying data from one place to another. Data mobility will be critical to the success of modern enterprise IT departments for the foreseeable future.

Replication to Databases

Organizations have long used databases such as Oracle and SQL Server for operational reporting; for example, to track sales transactions and trends, inventory levels, and supply-chain status. They can employ batch and CDC replication to copy the necessary records to reporting databases, thereby offloading the queries and analytics workload from production. CDC has become common in these scenarios as the pace of business quickens and business managers at all levels increasingly demand real-time operational dashboards.

ETL and the Data Warehouse

For several decades, data warehouses such as SQL Server, Oracle Exadata, and Teradata have served as the system of record for reporting and analytics. Although organizations also are implementing data lake and streaming platforms to address new analytics use cases with new data types, the data warehouse continues to play a central role in enterprise data strategies, addressing both traditional and new use cases.

To enable this, data must be fed to the data warehouse from operational databases via batch loads or CDC. Generally, the process begins with full loads. Data extracted from sources needs to go through the transformative raw and refinement phases, in which tools such as Talend detect, and correct or remove, corrupt and inaccurate records. The target data warehouse will stage raw data for refinement and then load and transform it in the central data warehouse database. In short, data goes through the ETL process.

CDC spreads data extraction and loading over time by continuously processing incremental updates. This in turn enables real-time transformation and real-time queries and reduces the performance impact of all three processes on data warehouse queries by smoothing CPU consumption over time. As discussed in Chapter 2, continuous CDC updates are replacing periodic batch loads as the pace of analytics has accelerated and tolerance for production downtime has declined.

CDC often integrates with native data warehouse loaders. For example, Teradata TPump is a highly parallel utility that continuously inserts, updates, and deletes data from external CDC software without locking out any users running queries on that table. Between CDC software and these native loaders, real-time data warehousing is fast becoming the norm in many environments. CDC also can be a powerful complement to data warehouse automation solutions that streamline creation, management, and updates of data warehouses and data marts. Together CDC and data warehouse automation software improve the agility of data warehouse teams and enable them to respond more quickly and easily to business events and changing business requirements.

Data Lake Ingestion

Data lakes have emerged as a critical platform for cost-effectively storing and processing a wide variety of data types. In contrast to the highly structured data warehouse, a data lake stores high volumes of structured, semistructured, and unstructured data in their native formats.

Data lakes, now widely adopted by enterprises, support myriad analytics use cases, including fraud detection, real-time customer offers, market trend and pricing analysis, social media monitoring, and more. And most of these require real-time data.

Within the Apache open source ecosystem, whose primary components are available in distribution packages from Hortonworks, Cloudera, and MapR, the historical batch processing engine MapReduce is increasingly being complemented or replaced by real-time engines such as Spark and Tez. These run on data stored on the Hadoop File System (HDFS) in the data lake.

Batch load and CDC replication software can access HDFS, potentially through the high-performance native protocol WebHDFS, which in turn uses the Kerberos open source network protocol to authenticate users. After it is connected, data architects and other users can use either batch load or CDC to land the source data in HDFS in the form of change tables. The data can be refined and queried there, using MapReduce batch processing for historical analysis.

CDC becomes critical when real-time engines like the Apache open source components Tez or Spark come into play. For example, the Spark in-memory processing framework can apply machine learning or other highly iterative workloads to data in HDFS. Alternatively, users can transform and merge data from HDFS into Hive data stores with familiar SQL structures that derive from traditional data warehouses. Here again, rapid and real-time analytics requires the very latest data, which requires CDC.

CDC software also can feed data updates in real time to data lakes based on the Amazon Simple Storage Service (S3) distributed object-based file store. S3 is being widely adopted as an alternative to HDFS. In many cases, it is more manageable, elastic, available, cost-effective, and persistent than HDFS. S3 also integrates with most other components of the Apache Hadoop stack, MapReduce, Spark, or Tez.

Publication to Streaming Platforms

A primary appeal of the Apache Kafka message-streaming platform, and more recent variants such as Amazon Kinesis and Microsoft Azure Event Hubs, is that they enable businesses to capitalize on the opportunities created by events such as a product purchase before they stream away (pun intended). These new systems are faster and more scalable and more reliable than previous messaging platforms such as Java Message Service (JMS).

In a streaming architecture, databases and other sources publish or "produce" streams of messages—for example, database transactions —that are independently subscribed to or "consumed" by various analytics engines (Storm, Spark Streaming, etc.) Messages are saved on the system broker server and can be played and replayed in sequence whenever they are needed. Consumers can group messages into topics for targeted analysis; for example, assigning each topic to a given source table. Topics themselves can be divided into partitions that are hosted on different servers to improve performance, scalability, and redundancy.

Data refinement is less relevant in the streaming model because the goal often is to respond rapidly to events rather than pursue in-depth analysis. However, source metadata/schema updates still need to be propagated into the message stream for cases in which data source structures are relevant to how the data is used. For example, an investment firm might use Kafka to distribute stock trade records to various stakeholders and might therefore need to ensure that the structure of those messages is synchronized with source transaction records.

CDC turns your data source, such as a production database, into a message producer. This is achieved by converting source updates into a stream of messages that can be divided into topics. CDC injects live updates into the message stream at near-zero latency, thereby enabling the real-time analytics that makes Kafka and its variants so appealing. With the right CDC solution, you can:

- Reduce Kafka programming requirements
- Send updates to multiple targets, not just the streaming platform

CDC also can transmit source schema/data definition language (DDL) updates into message streams and integrate with messaging schema registries to ensure that the analytics consumers understand the metadata. In addition, when CDC publishes to a standard link, Kafka Connect makes it easy to sink this data to a wide variety of target connectors such as Amazon S3, MySQL, and Elasticsearch.

Hybrid Cloud Data Transfer

We all know that cloud architectures have redefined IT. Elastic resource availability, usage-based pricing, and reduction of in-house IT management burden have proven compelling for most modern enterprises. Odds are that your organization is getting more and more serious about running operations and analytics workloads in the public cloud, most notably Amazon Web Services (AWS), Azure, or Google Cloud Platform. Most data infrastructure, including data warehouses, data lakes, and streaming systems, can run in the cloud, where they need the same data integration and processing steps outlined in this chapter.

As you adopt cloud platforms, data movement becomes more critical than ever. This is because clouds are not a monolithic, permanent destination. Odds are also that your organization has no plans to abandon on-premises datacenters wholesale anytime soon. Hybrid architectures will remain the rule. In addition, as organizations' knowledge and requirements evolve, IT will need to move workloads frequently between clouds and even back on-premises.

CDC makes the necessary data transfer more efficient and cost-effective. You can continuously synchronize on-premises and cloud data repositories without repeated, disruptive batch jobs. When implemented correctly, CDC consumes less bandwidth than batch loads, helping organizations reduce the need for expensive transmission lines. Transmitting updates over a wide area network (WAN) does raise new performance, reliability, and security requirements. Typical methods to address these requirements include multipathing, intermediate data staging, and encryption of data in flight.

A primary CDC use case for cloud architectures is zero-downtime migration or replication. Here, CDC is a powerful complement to an initial batch load. By capturing and applying ongoing updates, CDC can ensure that the target is and remains fully current upon completion of the initial batch load. You can switch from source to target without any pause in production operations.

The sports fan merchandiser Fanatics realized these benefits by using CDC to replicate 100 TB of data from its transactional, ecommerce, and back-office systems, running on SQL Server and Oracle on-premises, to a new analytics platform on Amazon S3. This cloud analytics platform, which included an Amazon EMR (Hadoop)–

based data lake, Spark, and Redshift, replaced the company's on-premises data warehouse. By applying CDC to the initial load and subsequent updates, Fanatics maintained uptime and minimized WAN bandwidth expenses. Its analysts are gaining actionable, real-time insights into customer behavior and purchasing patterns.

Microservices

A final important element of modern architectures to consider is *microservices*. The concept of the microservice builds on service-oriented architectures (SOAs) by structuring each application as a collection of fine-grained services that are more easily developed, deployed, refined, and scaled out by independent IT teams. Whereas SOA services often are implemented together, microservices are modular.

In his 2017 O'Reilly book *Reactive Microsystems*, Jonas Boner defined five key attributes of a microservice:

Isolation
: Architectural resources are separate from those of other microservices.

Autonomy
: Service logic and the teams that manage them have independent control.

Single responsibility
: A microservice has one purpose only.

Ownership of state
: Each microservice remembers all relevant preceding events.

Mobility
: A microservice can be moved to new deployment scenarios and topologies.

It's not difficult to envision the advantages of microservices for large, distributed enterprises that need to provide granular services to a wide range of customers. You want to automate and repeat processes wherever possible but still address individual errors, enhancement requests, and so on for each process in a focused fashion without interrupting other services. For instance, a global bank providing hundreds of distinct online services across multiple customer types needs to be able to rapidly update, improve, or retire individual services without waiting on other resources. Microservices help them achieve that.

CDC plays a critical role in such architectures by efficiently delivering and synchronizing data across the specialized microservice data repositories. In the next chapter, we explore how one global investment firm uses CDC to enable microservices for customers in multiple continents.

CHAPTER 4
Case Studies

Now let's explore some case studies. Each of these illustrates the role of change data capture (CDC) in enabling scalable and efficient analytics architectures that do not affect production application performance. By moving and processing incremental data and metadata updates in real time, these organizations have reduced or eliminated the need for resource-draining and disruptive batch (aka full) loads. They are siphoning data to multiple platforms for specialized analysis on each, consuming CPU and other resources in a balanced and sustainable way.

Case Study 1: Streaming to a Cloud-Based Lambda Architecture

Qlik is working with a Fortune 500 healthcare solution provider to hospitals, pharmacies, clinical laboratories, and doctors that is investing in cloud analytics to identify opportunities for improving quality of care. The analytics team for this company, which we'll call "GetWell," is using CDC software to accelerate and streamline clinical data consolidation from on-premises sources such as SQL Server and Oracle to a Kafka message queue that in turn feeds a Lambda architecture on Amazon Web Services (AWS) Simple Storage Service (S3). This architecture is illustrated in Figure 4-1. Log-based CDC has enabled them to integrate this clinical data at scale from many sources with minimal administrative burden and no impact on production operations.

GetWell data scientists conduct therapy research on this Lambda architecture, using both historical batch processing and real-time analytics. In addition to traditional SQL-structured analysis, they run graph analysis to better assess the relationships between clinical drug treatments, drug usage, and outcomes. They also perform natural-language processing (NLP) to identify key observations within physician's notes and are testing other new AI approaches such as machine learning to improve predictions of clinical treatment outcomes.

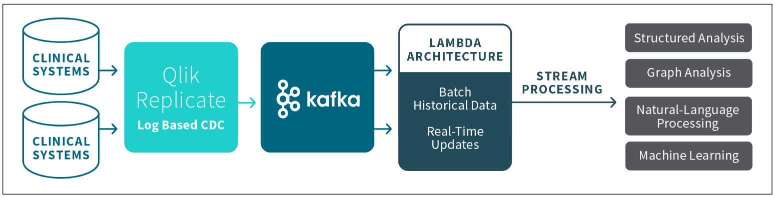


Figure 4-1. Data architecture for Kafka streaming to cloud-based Lambda architecture

CDC and Lambda Architecture

A Lambda architecture applies both batch and real-time processing to a given high-volume dataset to meet latency, throughput, and fault-tolerance requirements while combining comprehensive batch data views with online data views.

Technology components vary, but typically Apache Kafka or alternatives like Amazon Kinesis send messages to stream-processing platforms like Storm or Spark Streaming, which in turn feed repositories such as Cassandra or HBase. The batch layer is often MapReduce on HDFS, which might feed engines such as ElephantDB or Impala. Often queries are answered with merged results from both the batch and real-time engines.

CDC plays a vital role as it can both publish real-time streams to the data-in-motion infrastructure and write directly to the data-at-rest repositories.

Types of Artificial Intelligence

Artificial intelligence (AI) refers to the ability of machines to perceive and act upon environmental signals to achieve an objective, applying human-like cognitive functions such as learning and problem solving. Scientists such as Alan Turing first raised the concept of AI and laid its foundations after World War II. After a long period of inactivity, AI innovation has surged in the past 10 years, made possible by the rise of big data, distributed processing platforms such as Hadoop, and breakthroughs in special architectures such as neural networks.

Machine learning, a common attribute of AI systems, refers to predictive software that automatically teaches itself to improve its predictions without being explicitly programmed, using models based on defined data inputs. This includes *supervised learning*, which detects or matches patterns and classifies data based on predefined examples it has been fed. It also includes *unsupervised learning*, which detects new patterns without the guidance of predefined examples. Machine learning is being applied to many fields, such as medical diagnosis, fraud detection, and customer recommendation engines.

Deep learning is a type of machine learning that uses multiple layers of understanding, with the output from one layer becoming input for the next. This has been applied to complex fields such as medical research and audience targeting for mobile advertisements.

To begin to take advantage of these powerful AI technologies, organizations must move (via CDC) and structure all of the relevant and often heterogeneous enterprise data into formats that are analytics-ready. The CDC solution must also support appending new data to these structures and support slowly changing dimensions.

Case Study 2: Streaming to the Data Lake

Decision makers at an international food industry leader, which we'll call "Suppertime," needed a current view and continuous integration of production capacity data, customer orders, and purchase orders to efficiently process, distribute, and sell tens of millions of chickens each week. But Suppertime struggled to bring together these large datasets, which were distributed across several acquisition-related silos within SAP enterprise resource planning (ERP) applications. Using nightly batch replication, they were unable to match orders and production line-item data fast enough.

This slowed plant operational scheduling and prevented sales teams from filing accurate daily reports.

To streamline the process, Suppertime converted to a new Hadoop data lake based on the Hortonworks data platform and Qlik Replicate CDC. It now uses Qlik Replicate to efficiently copy all of the SAP record changes every five seconds, decoding that data from complex source SAP pool and cluster tables. Qlik Replicate injects this data stream, along with any changes to the source metadata and data definition language (DDL) changes, to a Kafka message queue that feeds HDFS and HBase consumers that subscribe to the relevant message topics (one topic per source table). Figure 4-2 illustrates this process.

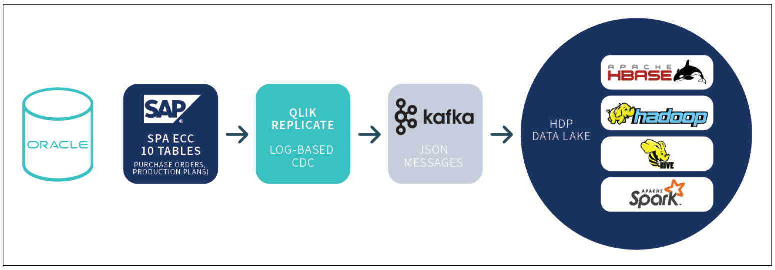


Figure 4-2. Data architecture for streaming to data lake

After the data arrives in HDFS and HBase, Spark in-memory processing helps match orders to production on a real-time basis and maintain referential integrity for purchase order tables. As a result, Suppertime has accelerated sales and product delivery with accurate real-time operational reporting. It has replaced batch loads with CDC to operate more efficiently and more profitably.

Case Study 3: Streaming, Data Lake, and Cloud Architecture

Another example is a US private equity and venture capital firm that built a data lake to consolidate and analyze operational metrics from its portfolio companies. This firm, which we'll call "StartupBackers," opted to host its data lake in the Microsoft Azure cloud rather than taking on the administrative burden of an on-premises infrastructure. Qlik Replicate CDC is remotely capturing updates and DDL changes from source databases (Oracle, SQL Server, MySQL, and DB2) at four locations in the United States. Qlik Replicate then sends that data through an encrypted File Channel connection over a wide area network (WAN) to a virtual machine–based instance of Qlik Replicate in the Azure cloud.

As shown in Figure 4-3, this Replicate instance publishes the data updates to a Kafka message broker that relays those messages in the JSON format to Spark. The Spark platform prepares the data in microbatches to be consumed by the HDInsight data lake, SQL data warehouse, and various other internal and external subscribers. These targets subscribe to topics that are categorized by source tables. With this CDC-based architecture, StartupBackers is now efficiently supporting real-time analysis without affecting production operations.

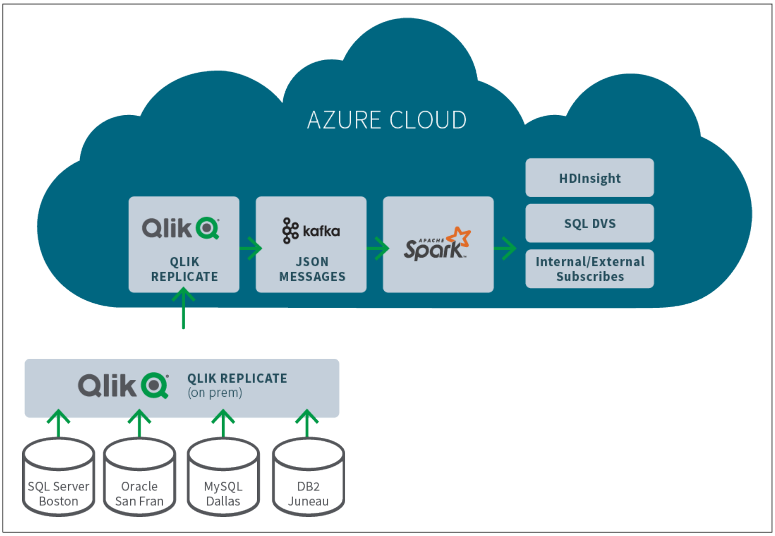


Figure 4-3. Data architecture for cloud-based streaming and data lake architecture

Case Study 4: Supporting Microservices on the AWS Cloud Architecture

The CIO of a very large investment management firm, which we'll call "Nest Egg," has initiated an ambitious rollout of cloud-based microservices as a way to modernize applications that rely on data in core mainframe transactional systems. Its on-premises DB2 z/OS production system continuously processes and reconciles transactional updates for more than a trillion dollars of assets under management. Nest Egg has deployed Qlik Replicate to capture these updates from the DB2 transaction log and send them via encrypted multipathing to the AWS cloud. There, certain transaction records are copied straight to an RDS database, using the same schemas as the source, for analytics by a single line of business.

In addition, Qlik Replicate copies transaction updates to an Amazon Kinesis message stream to which Nest Egg applies custom transformation logic. As shown in Figure 4-4, the transformed data then arrives in the AWS NoSQL platform DynamoDB, which feeds a microservices hub on RDS. Multiple regional centers, including offices based in London and Sydney, receive continuous updates from this hub via DynamoDB Streams.

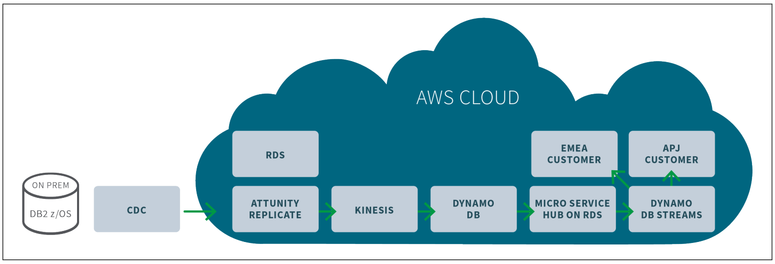


Figure 4-4. Data architecture for supporting cloud-based microservices

As a result, Nest Egg's microservices architecture delivers a wide range of modular, independently provisioned services. Clients across the globe have real-time control of their accounts and trading positions. And it all starts with efficient, scalable, and real-time data synchronization via Qlik Replicate CDC.

Case Study 5: Real-Time Operational Data Store/Data Warehouse

A military federal credit union, which we'll call "USave," involves a relatively straightforward architecture. USave needed to monitor deposits, loans, and other transactions on a real-time basis to measure the state of the business and identify potentially fraudulent activity.

To do this, it had to improve the efficiency of its data replication process. This required continuous copies of transactional data from the company's production Oracle database to an operational data store (ODS) based on SQL Server. Although the target is an ODS rather than a full-fledged data warehouse, this case study serves our purpose of illustrating the advantages of CDC for high-scale structured analysis and reporting.

As shown in Figure 4-5, USave deployed Qlik Replicate on an intermediate server between Oracle and SQL Server. The company automatically created tables on the SQL Server target, capturing the essential elements of the source schema while still using SQL-appropriate data types and table names. USave was able to rapidly execute an initial load of 30 tables while simultaneously applying incremental source changes. One table of 2.3 million rows took one minute. Updates are now copied continuously to the ODS.

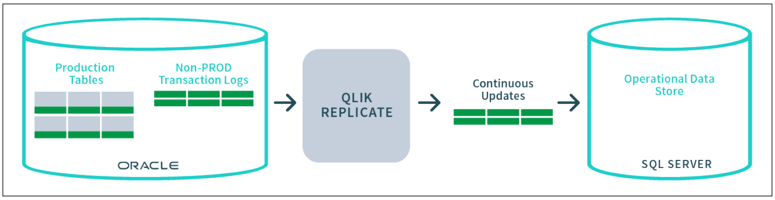


Figure 4-5. Data architecture for real-time ODS

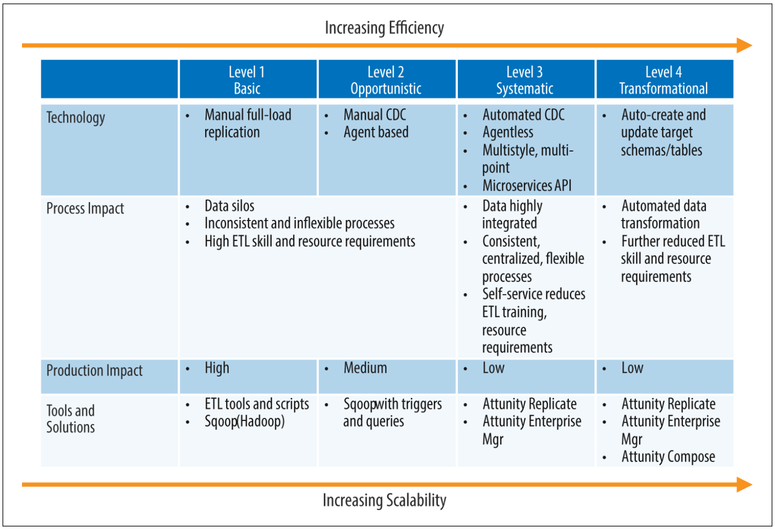
NOTE

An *operational data store* is a database that aggregates copies of production data, often on a short-term basis, to support operational reporting. The ODS often serves as an interim staging area for a long-term repository such as a data warehouse, which transforms data into consistent structures for more sophisticated querying and analytics.

CHAPTER 5

Architectural Planning and Implementation

To guide your approach to architectural planning and implementation, we have built the Replication Maturity Model, shown in Figure 5-1. This summarizes the replication and change data capture (CDC) technologies available to your IT team and their impact on data management processes. Organizations and the technologies they use generally fall into the following four maturity levels: *Basic*, *Opportunistic*, *Systematic*, and *Transformational*. Although each level has advantages, IT can deliver the greatest advantage to the business at the Transformational level. What follows is a framework, adapted from the Gartner Maturity Model for Data and Analytics (*ITScore for Data and Analytics*, October 23, 2017—see Appendix A), which we believe can help you steadily advance to higher levels.


Increasing Efficiency
Increasing Scalability

	Level 1 Basic	Level 2 Opportunistic	Level 3 Systematic	Level 4 Transformational
Technology	• Manual full-load replication	• Manual CDC • Agent based	• Automated CDC • Agentless • Multistyle, multi-point • Microservices API	• Auto-create and update target schemas/tables
Process Impact	• Data silos • Inconsistent and inflexible processes • High ETL skill and resource requirements		• Data highly integrated • Consistent, centralized, flexible processes • Self-service reduces ETL training, resource requirements	• Automated data transformation • Further reduced ETL skill and resource requirements
Production Impact	• High	• Medium	• Low	• Low
Tools and Solutions	• ETL tools and scripts • Sqoop(Hadoop)	• Sqoopwith triggers and queries	• Attunity Replicate • Attunity Enterprise Mgr	• Attunity Replicate • Attunity Enterprise Mgr • Attunity Compose

Figure 5-1. Replication Maturity Model

Level 1: Basic

At the Basic maturity level, organizations have not yet implemented CDC. A significant portion of organizations are still in this phase. During a course on data integration at a TDWI event in Anaheim in Orlando in December 2017, this author was surprised to see only half of the attendees raise their hands when asked if they used CDC.

Instead, organizations use traditional, manual extract, transform, and load (ETL) tools and scripts, or open source Sqoop software in the case of Hadoop, that replicate production data to analytics platforms via disruptive batch loads. These processes often vary by end point and require skilled ETL programmers to learn multiple processes and spend extra time configuring and reconfiguring replication tasks. Data silos persist because most of these organizations lack the resources needed to integrate all of their data manually.

Such practices often are symptoms of larger issues that leave much analytics value unrealized, because the cost and effort of data integration limit both the number and the scope of analytics projects. Siloed teams often run ad hoc analytics initiatives that lack a single source of truth and strategic guidance from executives. To move from the Basic to Opportunistic level, IT department leaders need to recognize these limitations and commit the budget, training, and resources needed to use CDC replication software.

Level 2: Opportunistic

At the Opportunistic maturity level, enterprise IT departments have begun to implement basic CDC technologies. These often are manually configured tools that require software agents to be installed on production systems and capture source updates with unnecessary, disruptive triggers or queries. Because such tools still require resource-intensive and inflexible ETL programming that varies by platform type, efficiency suffers.

From a broader perspective, Level 2 IT departments often are also beginning to formalize their data management requirements. Moving to Level 3 requires a clear executive mandate to overcome cultural and motivational barriers.

Level 3: Systematic

Systematic organizations are getting their data house in order. IT departments in this phase implement automated CDC solutions such as Qlik Replicate that require no disruptive agents on source systems. These solutions enable uniform data integration procedures across more platforms, breaking silos while minimizing skill and labor requirements with a "self-service" approach. Data architects rather than specialized ETL programmers can efficiently perform high-scale data integration, ideally through a consolidated enterprise console and with no manual scripting. In many cases, they also can integrate full-load replication and CDC processes into larger IT management frameworks using REST or other APIs. For example, administrators can invoke and execute Qlik Replicate tasks from workload automation solutions.

IT teams at this level often have clear executive guidance and sponsorship in the form of a crisp corporate data strategy. Leadership is beginning to use data analytics as a competitive differentiator. Examples from Chapter 4 include the case studies for Suppertime and USave, which have taken systematic, data-driven approaches to improving operational efficiency. StartupBackers (case study 3) is similarly systematic in its data consolidation efforts to enable new analytics insights. Another example is illustrated in case study 4, Nest Egg, whose ambitious campaign to run all transactional records through a coordinated Amazon Web Services (AWS) cloud data flow is enabling an efficient, high-scale microservices environment.

Level 4: Transformational

Organizations reaching the Transformational level are automating additional segments of data pipelines to accelerate data readiness for analytics. For example, they might use data warehouse automation software to streamline the creation, management, and updates of data warehouse and data mart environments. They also might be automating the creation, structuring, and continuous updates of data stores within data lakes. Qlik Compose (formerly Attunity Compose) for Hive provides these capabilities for Hive data stores so that datasets compliant with ACID (atomicity, consistency, isolation, durability) can be structured rapidly in what are effectively SQL-like data warehouses on top of Hadoop.

We find that leaders within Transformational organizations are often devising creative strategies to reinvent their businesses with analytics. They seek to become truly data-driven. GetWell (case study 1 in Chapter 4) is an example of a transformational organization. By applying the very latest technologies—machine learning, and so on—to large data volumes, it is reinventing its offerings to greatly improve the quality of care for millions of patients.

So why not deploy Level 3 or Level 4 solutions and call it a day? Applying a consistent, nondisruptive and fully automated CDC process to various end points certainly improves efficiency, enables real-time analytics, and yields other benefits. However, the technology will take you only so far. We find that the most effective IT teams achieve the greatest efficiency, scalability, and analytics value when they are aligned with a C-level strategy to eliminate data silos, and guide and even transform their business with data-driven decisions.

CHAPTER 6
The Qlik Platform

Qlik Replicate, a modern data integration platform built on change data capture (CDC), is designed for the Systematic (Level 3) and Transformational (Level 4) maturity levels described in Chapter 5. It provides a highly automated and consistent platform for replicating incremental data updates while minimizing production workload impact. Qlik Replicate integrates with all major database, data warehouse, data lake, streaming, cloud, and mainframe end points.

With Qlik Replicate, you can address use cases that include the following:

- Data lake ingestion
- Zero-downtime cloud migrations and real-time cloud-based analytics
- Publication of database transactions to message streams
- Mainframe data offload to data lake and streaming platforms
- Real-time data warehousing
- Enabling SAP data analytics on cloud/streaming/data lake architectures

Qlik Replicate resides on an intermediate server that sits between one or more sources and one or more targets. With the exception of SAP sources, which have special native requirements, no agent software is required on either source or target. Its CDC mechanism captures data and metadata changes through the least-disruptive method possible for each specific source, which in nearly all cases is its log reader. Changes are sent in-memory to the target, with the ability to filter out rows or columns that do not align with the target schema or user-defined parameters. Qlik Replicate also can rename target tables or columns, change data types, or automatically perform other basic transformations that are necessary for transfer between heterogeneous end points. Figure 6-1 shows a typical Qlik Replicate architecture.

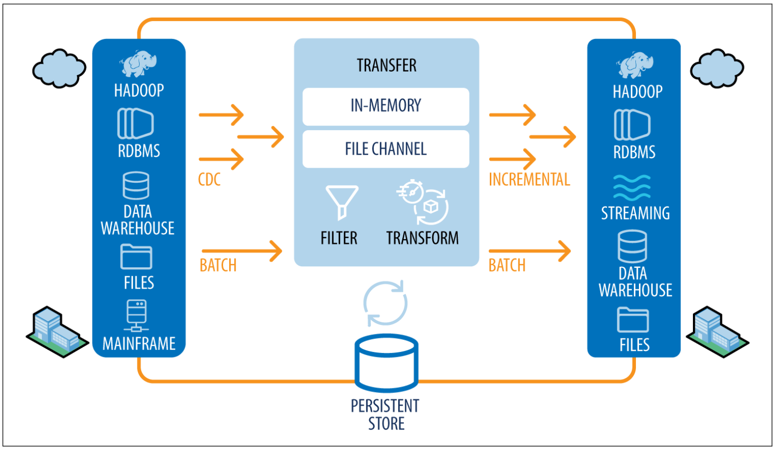


Figure 6-1. Qlik Replicate architecture

Qlik Replicate capabilities include the following:

Automation
: A web-based console enables data architects rather than extract, transform, and load (ETL) programmers to configure, control, and monitor replication tasks, thanks to a self-service approach that eliminates ETL scripting. The 100% automated process includes initial batch loads, transitions to CDC, target schema creation and schema/DDL change propagation, improving analytics agility, and eliminating the need for cumbersome recoding of brittle manual processes.

Efficient and secure cloud data transfer
: Qlik Replicate can compress, encrypt, and transfer data in parallel streams into, across, and out of cloud architectures, including Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform. Qlik technology has powered more than 55,000 data migrations to the cloud and is thus recognized as the preferred solution by its partners Microsoft and Amazon.

Rapid data lake ingestion

Qlik Replicate provides high-performance data loading to Hadoop targets such as Hadoop File System (HDFS) through native APIs and is certified with Cloudera, Hortonworks, and MapR. Time-based partitioning enables users to configure time periods in which only completed transactions are copied. This ensures that each transaction is applied holistically, providing transactional consistency and eliminating the risk of partial or competing entries.

Stream publication

Qlik Replicate enables databases to easily publish events to all major streaming services, including Apache Kafka, Confluent, Amazon Kinesis, Azure Event Hubs, and MapR-ES. Users can support multitopic and multipartition streaming as well as flexible JSON and AVRO file formats. They also can separate data and metadata by topic to integrate metadata more easily with various schema registries.

Qlik offers two additional products to improve the scalability and efficiency of data integration:

Qlik Enterprise Manager (formerly Attunity Enterprise Manager)

AEM allows you to design, execute, and monitor thousands of Qlik Replicate tasks across distributed environments. Users can group, search, filter, and drill down on key tasks; monitor alerts and KPIs in real time; and visualize metric trends to assist capacity planning and performance monitoring. AEM also provides REST and .NET APIs to integrate with workflow automation systems and microservices architectures.

Qlik Compose for Hive

This automates the creation and loading of Hadoop Hive structures as well as the transformation of enterprise data within them. Users can automatically create data store schemas and use the ACID Merge capabilities of Hive to efficiently process source data insertions, updates, and deletions in a single pass.

As shown in Figure 6-2, Qlik Replicate integrates with Qlik Compose for Hive to simplify and accelerate data pipelines. After Qlik Replicate lands data as raw deltas in the HDFS, Qlik Compose automates the creation and updates of Hadoop Hive structures as well as the transformation of data within them. Data is standardized, merged, and formatted in persistent Hive historical data stores. Qlik

Compose then can enrich and update the data and provision it to operational data stores or point-in-time snapshot views. By managing these steps as a fully automated process, Qlik Replicate and Qlik Compose accelerate data readiness for analytics.

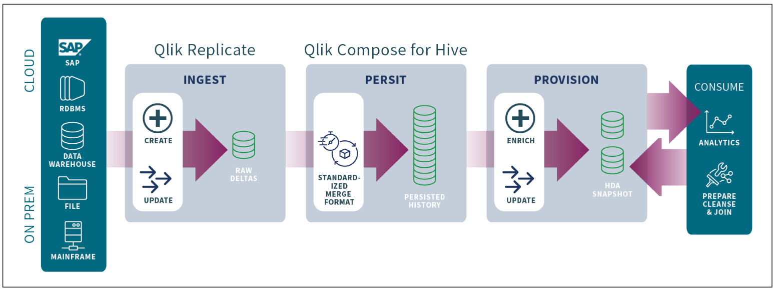


Figure 6-2. Data pipeline automation with Qlik

Qlik Replicate CDC and the larger Qlik Replicate portfolio enable efficient, scalable, and low-impact integration of data to break silos. Organizations can maintain consistent, flexible control of data flows throughout their environments and automate key aspects of data transformation for analytics. These key benefits are achieved while reducing dependence on expensive, high-skilled ETL programmers.

CHAPTER 7
Conclusion

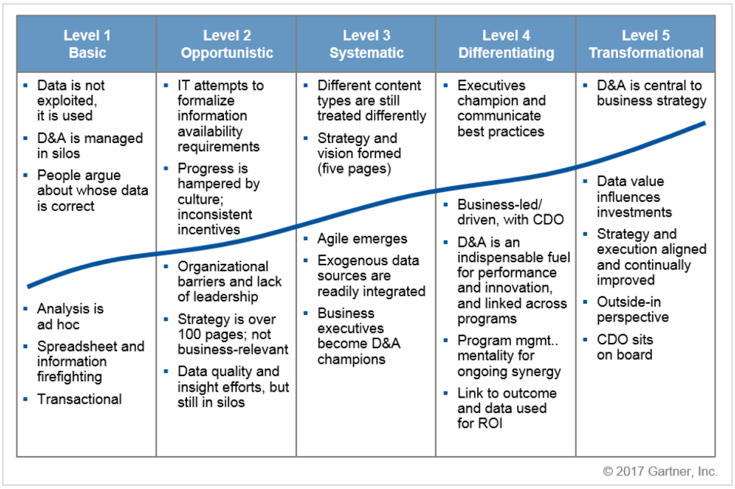
As with any new technology, the greatest barrier to successful adoption can be inertia. Perhaps your organization is managing to meet business requirements with traditional extract, transform, and load scripting and/or batch loading without change data capture. Perhaps Sqoop is enabling your new data lake to ingest sufficient data volumes with tolerable latencies and a manageable impact on production database workloads. Or perhaps your CIO grew up in the scripting world and is skeptical of graphical interfaces and automated replication processes.

But we are on a trajectory in which the business is depending more and more on analyzing growing volumes of data at a faster and faster clip. There is a tipping point at which traditional manual bulk loading tools and manual scripting begin to impede your ability to deliver the business-changing benefits of modern analytics. Successful enterprises identify the tipping point before it arrives and adopt the necessary enabling technologies. Change data capture is such a technology. It provides the necessary heartbeat for efficient, high-scale, and nondisruptive data flows in modern enterprise circulatory systems.

APPENDIX A

Gartner Maturity Model for Data and Analytics

The Replication Maturity Model shown in Figure 5-1 is adapted from the Gartner Maturity Model for Data and Analytics (*ITScore for Data and Analytics*, October 23, 2017), as shown in Figure A-1.



Level 1 Basic	Level 2 Opportunistic	Level 3 Systematic	Level 4 Differentiating	Level 5 Transformational
▪ Data is not exploited, it is used ▪ D&A is managed in silos ▪ People argue about whose data is correct ▪ Analysis is ad hoc ▪ Spreadsheet and information firefighting ▪ Transactional	▪ IT attempts to formalize information availability requirements ▪ Progress is hampered by culture; inconsistent incentives ▪ Organizational barriers and lack of leadership ▪ Strategy is over 100 pages; not business-relevant ▪ Data quality and insight efforts, but still in silos	▪ Different content types are still treated differently ▪ Strategy and vision formed (five pages) ▪ Agile emerges ▪ Exogenous data sources are readily integrated ▪ Business executives become D&A champions	▪ Executives champion and communicate best practices ▪ Business-led/ driven, with CDO ▪ D&A is an indispensable fuel for performance and innovation, and linked across programs ▪ Program mgmt.. mentality for ongoing synergy ▪ Link to outcome and data used for ROI	▪ D&A is central to business strategy ▪ Data value influences investments ▪ Strategy and execution aligned and continually improved ▪ Outside-in perspective ▪ CDO sits on board

© 2017 Gartner, Inc.

Figure A-1. Overview of the Maturity Model for Data and Analytics (D&A = data and analytics; ROI = return on investment)

About the Authors

Kevin Petrie is senior director of product marketing at Qlik. He has 20 years of experience in high tech, including marketing, big data services, strategy, and journalism. Kevin has held leadership roles at EMC and Symantec, and is a frequent speaker and blogger. He holds a Bachelor of Arts degree from Bowdoin College and MBA from the Haas School of Business at UC Berkeley. Kevin is a bookworm, outdoor fitness nut, husband, and father of three boys.

Dan Potter is VP of Product Marketing at Qlik. In this role, Dan is responsible for product marketing and go-to-market strategies related to modern data architectures, data integration, and DataOps. Prior, he held executive positions at Attunity, Datawatch, IBM, Oracle, and Progress Software where he was responsible for identifying and launching solutions across a variety of emerging markets including cloud computing, real-time data streaming, federated data, and ecommerce.

Itamar Ankorion is Senior Vice President of Technology Alliances at Qlik. In his previous role, Itamar was managing director of Enterprise Data Integration at Qlik, responsible for leading Qlik's Data Integration software business including Attunity, which was acquired by Qlik in May 2019.

Prior to the acquisition, Itamar was Chief Marketing Officer at Attunity where he was responsible for global marketing, alliances, business development and product management. Itamar has over 20 years of experience in marketing, business development and product management in the enterprise software space. He holds a BA in Computer Science and Business Administration and an MBA from the Tel Aviv University.