Lenovo Big Data Validated Design for Real-time Streaming Analytics with Cloudera Enterprise on ThinkSystem Servers

Last update: **23 September 2018**
Version 1.1

Describes the reference architecture for Cloudera Enterprise, powered by Apache Hadoop and Apache Spark

Solution based on the powerful, versatile Lenovo ThinkSystem SR650 and SR630 servers

Deployment considerations for high-performance, cost-effective and scalable solutions

Contains detailed information about choosing the appropriate Intel® processors and storage technologies

Lenovo

Intel

Cloudera

LENOVO PRESS

Table of Contents

1 Introduction

This document describes a real-time streaming reference architecture for Cloudera Enterprise on Lenovo ThinkSystem servers with locally attached storage. It provides a predefined and optimized hardware infrastructure for the Cloudera Enterprise, a distribution of Apache Hadoop and Apache Spark with enterprise-ready capabilities from Cloudera. The reference architecture provides the planning, design considerations, and best practices for implementing a real-time streaming Cloudera Enterprise solution with Lenovo products. For complete reference architecture details, refer to the Lenovo Big Data Validated Design for Cloudera Enterprise and VMware.

Lenovo, Intel, and Cloudera worked together on this document, and the reference architecture that is described herein was validated by Lenovo and Intel.

Hadoop is an open source software framework that is used to reliably manage large volumes of structured and unstructured data. Cloudera expands and enhances this technology to provide real-time streaming analytics capabilities that also meet enterprise demands such as management, security, governance, and analytics. The Intel® Optane™ Solid State Drive (SSD) helps eliminate storage bottlenecks and allows bigger, more affordable data sets. It can accelerate applications, reduce transaction costs for latency-sensitive workloads, and improve overall data center TCO. Intel® Optane™ technology is fast, inexpensive, and non-volatile, and can maximize processor utilization. Cloudera Enterprise, deployed on Lenovo ThinkSystem servers with Lenovo networking components, provides superior performance, reliability, and scalability. The reference architecture supports entry through high-end configurations and the ability to easily scale as the use of big data and real-time analytics grows.

The intended audiences for this reference architecture include IT professionals, technical architects, sales engineers, and consultants to assist in planning, designing, and implementing the big data solution with Lenovo hardware. It is assumed that you are familiar with Hadoop components and capabilities. For more information about Hadoop, see “Resources” on page 31.

2 Business problem and business value

This section describes the business challenges associated with real-time streaming environments and the value offered by the Cloudera solution that uses Lenovo hardware powered by Intel® processors.

2.1 Business problem

The world is predicted to generate more than 40 million TB of data by 2020. This data comes from everywhere in a constant stream, from sensors that are used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone global positioning system (GPS) signals. It is big data, and it never stops flowing. Traditionally, decisions were made based on reports of monthly data roll-ups, and this approach still serves well for historical and archived data where time is not a crucial factor. But for time-sensitive data, answers must be delivered within seconds to be of great value for organizations.

To stay competitive, enterprises must deploy real-time streaming machine-learning applications in their data centers. When analyzed in the right context, data streams from financial markets, mobile devices, the Internet of Things (IoT), clickstreams, business transactions, and other sources can provide hidden value and insights.

It can be challenging, however, to set up an easy-to-deploy data storage and processing infrastructure that can deliver the promised value in a very short amount of time. Spending months hiring dozens of skilled engineers to piece together a data management environment is very costly and often leads to frustration from unrealized goals. Furthermore, the data processing infrastructure needs to be easily scalable in addition to achieving desired performance and reliability objectives.

2.2 Business value

Ninety-one percent of CIOs say streaming data analytics can positively impact their company's bottom line. Using the real-time streaming analytics reference architecture, an overview of which is shown in Figure 1, enterprises can expect the following benefits:

- Fast insights and decision making
- The ability to quickly and effectively respond to constantly changing business conditions
- Enhanced regulatory compliance

This reference architecture can be used across several enterprise use cases:

- Serve real-time data at scale by injecting real-time data and analysis into decision points across the organization:
  - High concurrency and low latency ensure all users can access data
  - Cloudera search enables data access across the organization
  - Data-based applications distribute custom, easy-to-digest information
- Perform predictive modelling to combine real-time data with vast historical data to better anticipate future events:
  - Increase conversion rate of cross-sell and upsell opportunities
  - Better predict creditworthiness and lifetime customer value
  - IoT: Perform predictive maintenance to reduce downtime costs

- Execute monitoring and detection to protect the enterprise network and data by identifying patterns in large network data sets:
    - Decrease network downtime via predictive maintenance enabled by active collection and monitoring of network data
    - Identify advanced persistent threats (APTs) via data collection and anomaly analysis
    - Work with advanced cybersecurity technologies such as Apache Spot

For additional references, analytics capabilities, and use case content, refer to Cloudera's web site.

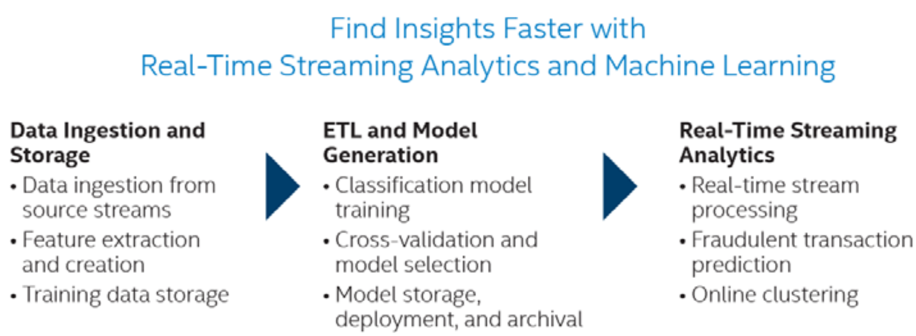

Figure 1. Intel, Lenovo, and Cloudera have collaborated to create a real-time streaming architecture that uses machine learning to turn raw data into deep business insights.

Real-time streaming analysis benefits a broad set of industries:

- The healthcare industry is rapidly turning to big data, machine learning, and real-time analytics to monitor patient safety, personalize patient results, and assess clinical risk as well as reduce patient readmission—all of which improve organizational efficiency and patient experience.
- The transportation industry faces rapid changes. Autonomous vehicles, advanced driver assistance systems, and the IoT are exponentially increasing transportation data set volumes. Plus, more of the population is moving to urban areas, increasing traffic density. Real-time data streaming coming from edge devices can help predict speed, travel times, and best routes, and manage traffic density.
- Retailers with stores located worldwide as well as an online presence struggle with real-time inventory tracking. Real-time streaming analytics based on a central scalable repository can improve operational efficiency, drive higher sale volumes, identify trend insights, and enhance customer satisfaction.
- Call center logs can act as a central input to predict customer churn. These logs can include data from sensors, online interactions, interactive voice response systems, and IT support systems. By streaming this data into a data lake, enterprises can create models for churn prediction as well as derive insights into customer metrics and incidents.
- Cybersecurity is among the most critical threats facing enterprises, with increasingly sophisticated hackers continuously seeking vulnerabilities they can use to steal data. Tools such as Apache Spot can help expedite threat detection with machine-learning models that perform streaming analysis of network flows, DNS data, proxies, and so on.
- Financial transactions represent a continuous stream of data from several sources. Real-time streaming analytics can aggregate this data to improve business health and reveal insights. In addition, previous transactional data can be used to train machine-learning models to predict fraudulent transactions.

3 Requirements

The functional and non-functional requirements for this reference architecture are desribed in this section.

3.1 Functional requirements

A big data solution supports the following key functional requirements:

- Ability to handle various workloads, including batch and real-time analytics
- Industry-standard interfaces that allow applications to work with Cloudera
- Ability to handle large volumes of data of various data types
- Various client interfaces

3.2 Non-functional requirements

Customers require their big data solution to be easy, dependable, and fast. The following non-functional requirements are key:

- Easy:
    - Ease of development
    - Easy management at scale
    - Advanced job management
    - Multi-tenancy
    - Easy-to-access data by various user types
- Dependable:
    - Data protection with snapshot and mirroring
    - Automated self-healing
    - Insight into software/hardware health and issues
    - High availability (HA) and business continuity
- Fast:
    - Superior performance
    - Scalability
- Secure and governed:
    - Strong authentication and authorization
    - Kerberos support
    - Data confidentiality and integrity

4 Architectural overview

Lenovo, Cloudera, and Intel are taking advantage of innovations in silicon and software by creating a real-time streaming analytics reference architecture (see Figure 2). Besides showing the general categories of activity, such as data ingestion, analytics, and data visualization, the figure also indicates several node personalities, which may be spread across several types of activity. Refer to Table 2 on page 20 for how these node personalities are configured.

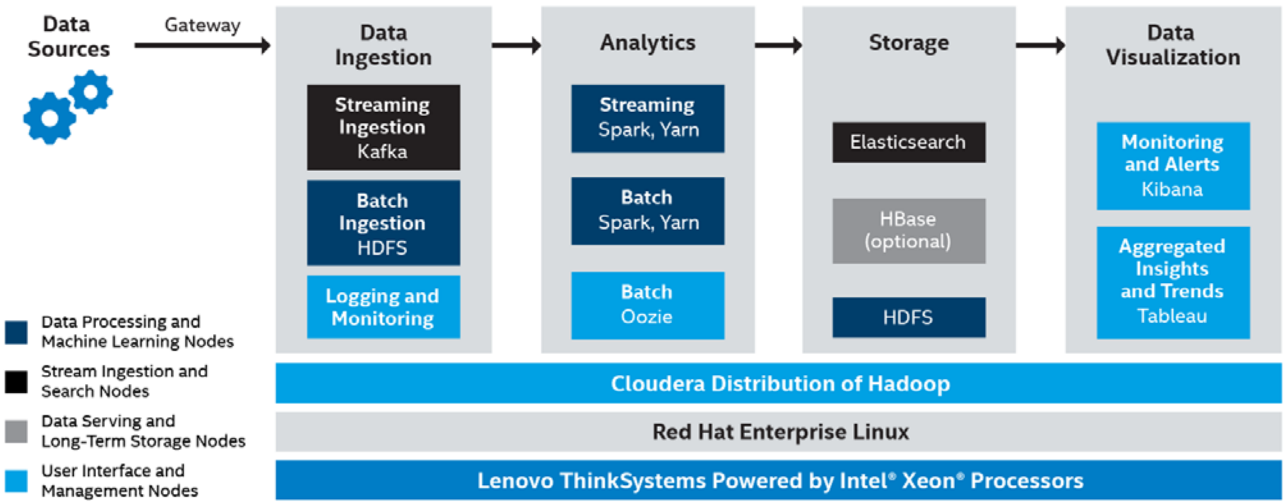

Figure 2. Real-time streaming analytics stack

The reference architecture uses several components to accomplish the workflow:

- Data ingestion. Historical transactions are stored in the Hadoop Distributed File System (HDFS) for batch analytics while real-time data from the web, smartphones, and card readers are streamed to Kafka. Records are represented in CSV strings and may use Apache Avro for serialization and Snappy compression. For secure implementations, encryption can be added at every stage of the pipeline along with Kerberos for authentication. Logging and monitoring are accomplished with tools such as Elasticsearch, Kibana, and Logstash.
- Batch analytics. The reference architecture uses Spark and YARN along with Oozie and HDFS to perform feature engineering. This task includes dropping duplicates or filtering unformatted records, and feature imputation and binning, as well as attribute generation using aggregations. Seaborn or matplotlib can be used for visualizing feature correlations. Once all features have been created, a features vector is assembled to train the machine-learning algorithms. The whole machine-learning pipeline is packaged and saved for auditing purposes and for reuse during the streaming pipeline.
- Streaming analytics. New transactions are pulled directly from Kafka brokers at various frequencies. Sizing and mini-batch intervals depend on throughput and latency service level agreements (SLAs). For example, 20 seconds might be appropriate for certain workloads, while others might need to process the pipeline every second, or several times per second. The reference architecture uses Spark Streaming for stream processing. Spark Streaming, apart from its maturity, provides seamless integration with Kafka and YARN and with available libraries that support CSV parsing, Spark ML, and Elasticsearch-Hadoop (ES-Hadoop) connectors. Spark Streaming also enjoys widespread community involvement. The streaming workflow involves processing data through online clustering

(K-Means) and fraud detection (Logistic Regression and Random Forest Classification) algorithm pipelines. Transactions are initially processed using feature-engineering pipelines and then fitted into the models to obtain predictions. Fraudulent records are pushed out to Elasticsearch for rapid indexing and search as well as saved to HBase for long-term storage (Cassandra can act as an alternative).

- Data visualization. The reference architecture uses Kibana as the analytics and visualization platform. Kibana is open source and is designed to work with Elasticsearch as part of the Elastic Stack. Additionally, Elastic offers X-Pack extensions for the enterprise that bundle security, alerting, monitoring, reporting, and graph capabilities.

5 Component model

5.1 Cloudera Enterprise overview

Cloudera Enterprise (see Figure 3) supports mission-critical and real-time big data analytics across different industries, such as financial services, retail, media, healthcare, manufacturing, telecommunications, government organizations, and leading Fortune 100 and Web 2.0 companies.

Cloudera Enterprise contains the following components:

- Analytic SQL: Apache Impala
- Search Engine: Cloudera Search
- NoSQL: HBase
- Stream Processing: Spark
- Machine Learning: Spark MLlib
- Cloudera Manager and associated services
- Cloudera Navigator
- Cloudera Kafka

For more information, visit cloudera.com/content/cloudera/en/products-and-services/product-comparison.html and refer to the Lenovo Big Data Validated Design for Cloudera Enterprise and VMware.

The Cloudera solution is OS-independent. Cloudera supports many Linux operating systems, including Red Hat Linux and SUSE Linux. For more information about the versions of supported operating systems, visit http://www.cloudera.com/documentation/enterprise/latest/topics/cm_ig_cm_requirements.html.

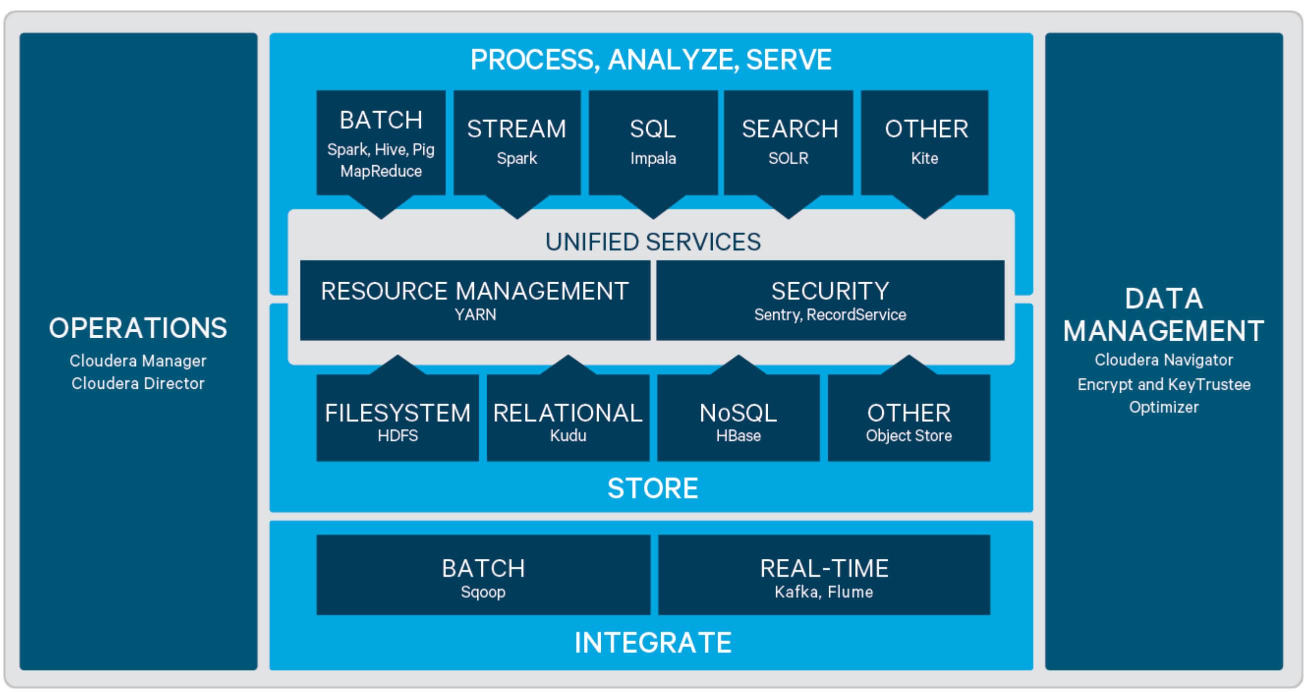

Figure 3. Cloudera Enterprise key capabilities

5.2 Software considerations

5.2.1 OS

The reference architecture is based on Red Hat Enterprise Linux (RHEL) 7.3 deployments. Users need to ensure that the latest official patches are installed on all nodes of the deployment and any necessary upgrades have been executed before starting the installation.

Collaboration between Intel, Red Hat and Cloudera yields a platform built for:

- Layered security due to hardware offload of encryption/decryption, OS-level security, and identity management in RHEL and Security Enhanced Linux (SELinux)
- Optimized performance with a combination of drivers, libraries, and extensions for cluster environments and Intel® components
- Simplified ease of use and seamless operation management with RHEL OpenStack Platform

5.2.2 Libraries

- Snappy compression. Enabling intermediate compression can make jobs run faster without requiring application changes. Data sent by publishers to Kafka brokers as well as temporary intermediate files created by Hadoop for the shuffle phase can benefit from compression; the final output may or may not be compressed. Snappy is ideal in this case because it compresses and decompresses very quickly compared to other compression algorithms, such as Gzip.
- Machine-learning acceleration. Intel® Math Kernel Library (Intel® MKL) is a library of optimized math routines that are hand-optimized specifically for Intel processors. For example, it includes highly optimized routines for linear algebra, Fast Fourier Transforms (FFT), vector math, and statistics functions. These mathematical operations are building blocks for machine learning and related analytic algorithms, and thus integration with Intel MKL delivers a massive performance boost for machine-learning workloads. Spark is already instrumented to take advantage of optimized implementations of these routines using netlib-java, but still requires the addition of an implementation like Intel MKL to activate these optimizations. The benefits of these performance gains are clear: Improved performance means you can train with larger data sets, explore a larger range of the model hyper-parameter space, and train more models. In many cases, it alleviates the need to buy any specialized hardware for your machine-learning workloads. Performance gains vary differently for individual algorithms and depend on the business and data science use case.

5.2.3 Cloudera Enterprise

Cloudera Distribution of Hadoop (CDH) is a popular Hadoop distribution that is 100 percent open source. It includes all the leading Hadoop ecosystem components to store, process, discover, model, and serve unlimited data, and it is engineered to meet the highest enterprise standards for stability and reliability.

To access the Cloudera solution from the corporate network, users can log into the Cloudera client from outside the firewall by using Secure Shell (SSH) on port 22. CDH provides a seamless setup to install, manage, and maintain various Hadoop components, and also provides several interfaces that allow administrators and users to perform administration and data functions, depending on their roles and access level. Hadoop application programming interfaces (APIs) can be used to access data. Cloudera APIs can be used for cluster management and monitoring. Cloudera data services, management services, and other services run on the nodes in the cluster. In this reference architecture, various services from CDH were used. Spark v2 and Kafka v0.10 gateways were installed on all nodes of the cluster while the actual service hosts were installed only on a handful of nodes. The spark-streaming-kafka_0.8 module was used to integrate Spark and Kafka.

Storage is a component of each data node in the cluster. Data can be incorporated into Cloudera Enterprise storage through the Hadoop APIs or network file system (NFS), depending on the customer's needs.

A database is required to store the data for Cloudera manager, Hive metastore, and other services. Cloudera provides an embedded database for test or proof-of-concept (POC) environments and an external database is required for a supportable production environment.

5.2.4 Elastic products

Elastic (ELK) products, including X-Pack, provide a scalable, distributed search framework on top of Apache Lucene to store and index large amounts of data and make that data accessible in near real time (see Figure 4). Elasticsearch comes with various connectors to integrate user workflows, including components from CDH. Through the Elasticsearch Hadoop integration, Elasticsearch enables users to enhance their workflow with a search and analytics engine. Elasticsearch provides a rich language to ask better questions and obtain fast, clear answers.

Apart from Elasticsearch, Kibana provides a unified dashboard interface for creating visualizations while the X-Pack plugin enables monitoring, security, and enhanced add-ons for analyzing the data further.

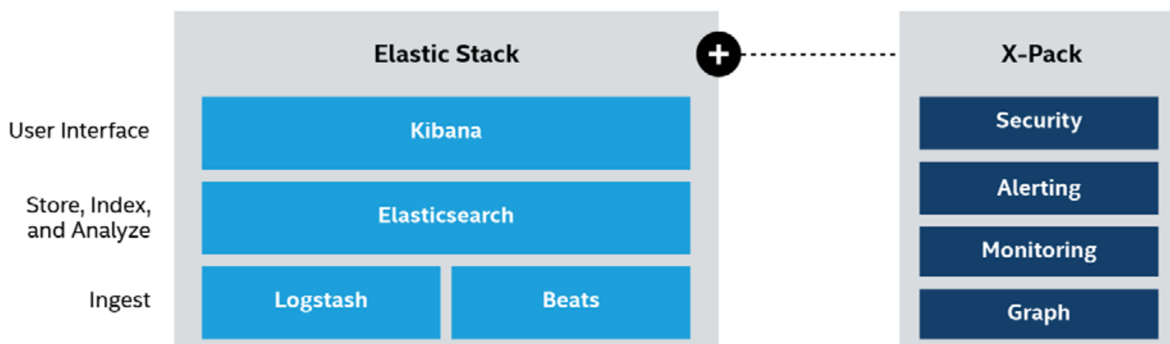

Figure 4. The Elastic software stack

5.3 Apache Spark in CDH 5.10

[Spark](#) has recently become very popular and is being adopted as a preferred framework for a variety of big data use cases ranging from batch applications that use MapReduce or Spark with data sources such as click streams, to real-time applications that use sensor data. The Spark stack is shown in Figure 5. As depicted, the foundational component is the Spark Core. Spark is written in the Scala programming language and offers simple APIs in Python, Java, Scala, and SQL.

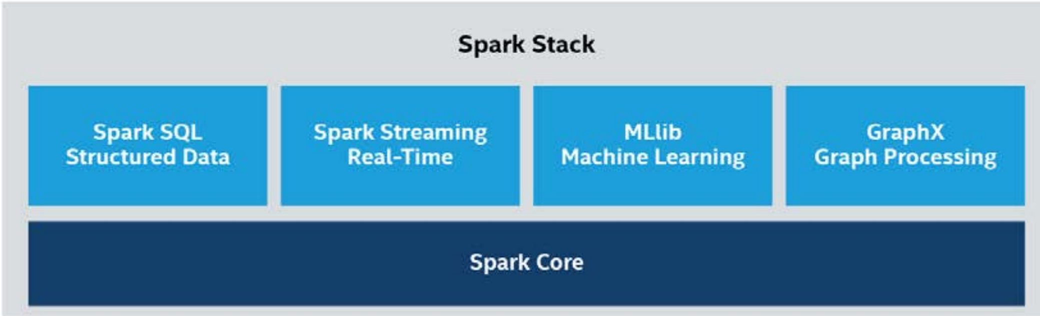

Figure 5. The Spark stack

In additional to the Spark Core, the Spark stack allows extensions in the form of libraries. The most common extensions are Spark MLlib for machine learning, Spark SQL for queries on structured data, Spark Streaming for real-time stream-processing, and Spark GraphX for handling graph databases. Other extensions are also available. Cloudera doesn't currently support GraphX or SparkR. There are also caveats for Spark SQL support—please refer to the Cloudera Spark Guide.

Note that while the majority of big data infrastructures are built using Intel processors, the implementations in MLlib are not necessarily optimized for Intel® architecture. It is therefore in many developers' best interest to make Spark MLlib run faster on Intel processor-based clusters. Some popular machine-learning acceleration libraries from Intel include the Intel MKL and the Intel® Data Analytics Acceleration Library (Intel® DAAL).

The Spark stack shown in Figure 5 supports multiple project types. Typical big data usage scenarios to date have deployed the Hadoop stack for batch processing separately from another framework for stream processing, and yet another one for advanced analytics such as machine learning. Spark combines these frameworks in a common architecture, thereby allowing easier management of the big data code stack and also enabling reuse of a common data repository.

Spark offers multiple advantages over Hadoop MapReduce. These advantages include:

- Fault-tolerant distributed data structures (resilient distributed datasets, known as RDDs)
- More operations available for data processing
- Ease-of-use (increased developer productivity)
- Support for many types of clusters
- Easy connection to many types of data sources.

The Spark stack shown in Figure 5 can run in a variety of environments. It can run alongside the Hadoop stack, leveraging Hadoop YARN for cluster management. Spark applications can run in a distributed mode on a cluster using a master/slave architecture that uses a central coordinator called "driver" and potentially large number of "worker" processes that execute individual tasks in a Spark job. The Spark executor processes also provide reliable in-memory storage of data distributed across the various nodes in a cluster. The components of a distributed Spark application are shown in Figure 6.

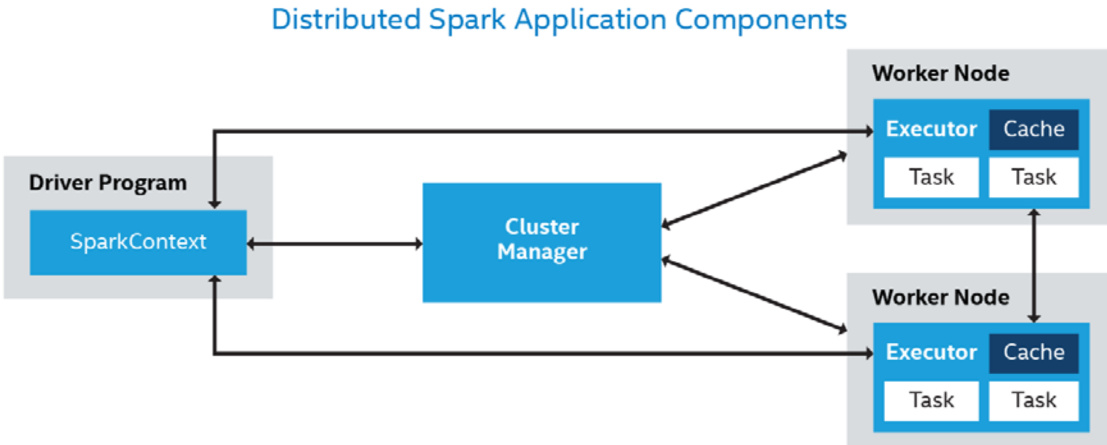

Figure 6. Distributed Spark application component model

A key distinguishing feature of Spark is the data model, based on RDDs. This model enables a compact and reusable organization of data sets that can reside in main memory and can be accessed by multiple tasks. Iterative processing algorithms can benefit from this feature by avoiding storing and retrieving data sets from disks between computation iterations—delivering significant performance gains compared to MapReduce.

RDDs support two types of operations: Transformations and Actions. Transformations are operations that return a new RDD, while Actions return a result to the driver program. Spark groups operations together to reduce the number of passes taken over the data. This evaluation technique requires minimal work and enables faster data processing. Spark also allows caching data in memory for persistence to enable multiple uses of the same data. This is another technique contributing to faster data processing.

5.4 Other open source components

This sub-section provides details of Elasticsearch, Kibana, and other packages.

5.4.1 Apache Kafka

Kafka is used for building real-time data pipelines and streaming apps. It is horizontally scalable, fault-tolerant, extremely fast, and runs in production in thousands of companies. Its key purposes are three-fold:

- Publish and subscribe – Read and write streams of data like a messaging system
- Process – Write scalable stream processing applications that react to events in real time
- Store – Store streams of data safely in a distributed, replicated, fault-tolerant cluster

5.4.2 Elasticsearch and Kibana

Hadoop shines as a batch processing system, but serving real-time results can be challenging. For truly interactive data discovery, ES-Hadoop allows indexing of Hadoop data into the Elastic Stack to take full advantage of the speedy Elasticsearch engine and Kibana visualizations (see Figure 7). With ES-Hadoop, it is easy to build dynamic, embedded search applications to serve Hadoop data or perform deep, low-latency analytics using full-text, geospatial queries and aggregations. From product recommendations to genomic sequencing, ES-Hadoop opens up a new world of broad applications.

Integration for Spark Streaming and Kafka are available via the ES-Hadoop library connectors, making it easy for stream processing application to read/write to Elasticsearch.

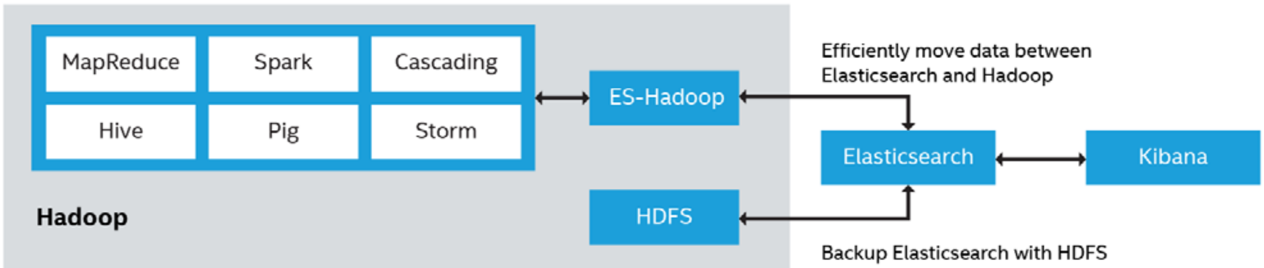

Figure 7. Elasticsearch-Hadoop

6 Operational model

This section describes the operational model for the Cloudera reference architecture. To show the operational model for different sized customer environments, three different models are provided for supporting different amounts of data. Throughout the document, these models are referred to as half rack, full rack, and multi-rack configuration sizes. The multi-rack is three times larger than the full rack.

The discussion in the following sections provides details germane to a real-time streaming implementation. For a general discussion of server hardware, cluster nodes, systems management, and networking, refer to the Lenovo Big Data Validated Design for Cloudera Enterprise and VMware.

6.1 Overview

A Cloudera deployment consists of cluster nodes, networking equipment, power distribution units, and racks.

- Management. The reference architecture handles both systems management (using Cloudera Manager) and hardware management (using Lenovo XClarity™ Administrator). In addition, xCAT provides a scalable distributed computing management and provisioning tool that provides a unified interface for hardware control, discovery, and operating system deployment. It can be used to facilitate or automate the management of cluster nodes. For more information about xCAT, see “Resources” on page 31. For more information on installing and managing Cloudera Enterprise, see their latest documentation.
- Networking. The reference architecture specifies two networks: a data network and an administrative management network.
    - The data network creates a private cluster among multiple nodes and is used for high-speed data transfer across nodes, and for importing data into the Cloudera cluster. The Cloudera cluster typically connects to the customer’s corporate data network. For cross-rack networking, additional switches per cluster are required.
    - The hardware management network is a 1 GbE network for out-of-band hardware management. Through the XClarity™ Controller management module (XCC) within the ThinkSystem SR650 server, the out-of-band network enables the hardware-level management of cluster nodes, such as node deployment, BIOS configuration, hardware failure status, and server power states.
- Predefined configurations. The predefined configurations can be implemented as-is or modified based on specific customer requirements, such as lower cost, improved performance, and increased reliability. Key workload requirements, such as the data growth rate, sizes of datasets, and data ingest patterns help in determining the proper configuration for a specific deployment. A best practice when a Cloudera cluster infrastructure is designed is to conduct the proof of concept testing by using representative data and workloads to ensure that the proposed design works.

6.2 Hardware considerations

The following sub-sections discuss considerations associated with servers, CPUs, memory, storage, and networking.

6.2.1 Server

High-throughput stream processing jobs are well-suited to scale horizontally; they also perform better with more cores. In addition, they usually benefit from hyper-threading and are minimally impacted by NUMA designs. This reference architecture uses Lenovo servers SR630 (1U) and SR650 (2U) servers and Lenovo RackSwitch G8052 and G8272 top of rack (TOR) switches.

6.2.2 CPU

Intel® Xeon® processor Scalable family is a new microarchitecture with many additional features compared to the previous-generation Intel® Xeon® processor E5-2600 v4 series. These features include increased processor cores, increased memory bandwidth, non-inclusive cache, Intel® Advanced Vector Extensions 512 (Intel® AVX-512), Intel® Memory Protection Extensions (Intel® MPX), Intel® Ultra Path Interconnect (Intel® UPI), and sub-NUMA clusters.

6.2.3 Memory

Memory requirements depend on the capacity planning for a particular business use case. Typically, real-time streaming applications are both compute- and memory-intensive. Also, additional memory on nodes can help with caching data so processes do not have to flush to or read from disks when data volumes increase. Queuing systems like Apache Kafka and indexing/search components such as Cloudera Solr and Elasticsearch rely heavily on the OS file-system cache, which can benefit from high memory capacity. Typical memory recommendations range from 256 GB to 768 GB on data/worker nodes and 192 GB to 256 GB on master/client nodes.

High-memory systems also enable support of a containerized infrastructure on a single node such as Docker Swarm or a Kubernetes-based architecture. JVM garbage collection is one of the most important tuning parameters involved in large-scale data analytics. Performance bottlenecks are usually alleviated by allocating large memory buffers to workers and executors – even though adding memory may seem to be a temporary workaround, it is crucial for quick resolution of performance issues in several production scenarios.

6.2.4 Storage

The performance of an analytics solution relies heavily on the efficiency and reliability of associated storage. Because real-time stream processing frameworks are sensitive to read/write latencies, it is imperative to use a highly performant storage infrastructure that can adhere to several kinds of workloads. Furthermore, components have differing requirements for storage, depending on the throughput or latency SLAs they are trying to provide.

Intel's scalable storage systems turn data growth into opportunity so innovative businesses can gain a competitive advantage. While SSDs have long been used to accelerate workloads, new technologies such as NVMe and Intel Optane technology help achieve new thresholds. The Intel® SSD DC P4500 series is well-suited to handle high-volume sequential writes catering to YARN shuffles and incoming Kafka gateway data.

Considering low-latency needs for Elasticsearch queries, Intel Optane technology helps scale the workload for random reads and writes as traffic changes and evolves for real-time streaming.

SATA hard disk drive (HDD) drives are recommended to store archival data. These drives are typically deployed in a “just a bunch of disks” (JBOD) array configuration. JBOD is preferred over RAID as RAID’s overhead might not provide value, considering Apache Hadoop components have built-in implementation for replication, fault tolerance, and high availability. A RAID configuration is limited by the speed of the slowest disk in the RAID array, whereas disk operations in a JBOD array are independent, providing a higher average read/write throughput.

While ext4 and xfs are among the recommended filesystems for drives among large-scale workloads, we have tested all drives formatted with ext4 in this reference architecture.

6.2.5 Networking

Big data and streaming solutions require a high-bandwidth networking interface. Transmission and receiver connections may suffer from contention in a poorly configured or saturated network, which can result in packet re-transmission and network latencies. At least a 10 Gigabit Ethernet (GbE) interface is recommended for the entire architecture, and preferably a 25 GbE interface and Ethernet network adapters could be used for maximum performance. Additional tuning, such as enabling jumbo frames, should be performed to raise the bandwidth efficiency by reducing the overhead. However, one must ensure that all devices in the network are configured for jumbo frames to avoid any potential performance issues.

6.3 Cluster nodes

The Cloudera reference architecture is implemented on a set of nodes that make up a cluster. A Cloudera cluster consists of two types of nodes: worker nodes and master nodes. Worker nodes use ThinkSystem SR650 servers with locally attached storage, and master nodes use ThinkSystem SR630 servers.

Worker nodes run data (worker) services for storing and processing data.

Master nodes run the following types of services:

- Management control services for coordinating and managing the cluster
- Miscellaneous and optional services for file and web serving

Table 1 provides information about which software is installed on which nodes. Note that the roles shown are based on full-rack deployment. Services might be co-located for smaller configurations.

Table 1. Service layout matrix

Hardware Component	Associated Software
Master node1	Cloudera Distribution of Hadoop (CDH) Management Services
Master node2	
Master node3	
Edge node	Kibana/Gateway
Stream ingestion + search1	Kafka and Elasticsearch
Stream ingestion + search2	
Stream ingestion + search3	
Stream ingestion + search4	
Store & data serving1	Long-term storage, HDFS, and HBase/Cassandra
Store & data serving2	
Store & data serving3	
Store & data serving4	
Store & data serving5	
Store & data serving6	
Streaming Analytics1 ZooKeeper	Spark (batch) Spark Streaming YARN Oozie ZooKeeper (can be moved to ingestion nodes)
Streaming Analytics2 ZooKeeper	
Streaming Analytics3 ZooKeeper	
Streaming Analytics4	
Streaming Analytics5	
Streaming Analytics6	

6.3.1 Worker nodes

Server nodes run on the ThinkSystem SR650 servers which include an 128 GB SSD for the OS and 256 GB of base memory, as well as an HDD controller and hardware storage protection for both the OS and HDFS (Cloudera maintains a total of three copies of data stored within the cluster. The copies are distributed across data servers and racks for fault recovery.)

Table 2 provides configuration recommendations for each node personality. For example, the data processing nodes are configured differently than the data serving nodes. Figure 2 on page 8 shows how these node personalities are distributed across the various parts of the analytics stack.

The Intel® Xeon® Scalable processors recommended will provide a balance in performance vs. cost for Cloudera worker nodes. Higher core count and frequency processors are available for compute intensive workloads

Table 2. Different worker node configurations

Node Personality (refer to Figure 2)	Components	Nodes	Quantity per Node	Notes
Data Processing & Machine Learning Training/Inference (Spark, YARN, Spark Streaming)	Intel® Xeon® Platinum 8168 Processor 24 cores 2 sockets (96 threads with hyper-threading enabled)	6		
	384 GB DRAM			
	2 TB SATA HDD 3.5”/2.5”		8x	
	25 GbE/10 GbE NIC		2 ports	
	Intel® SSD DC P4500 Series 4 TB		1x	Shuffle acceleration
Data Serving & Long-Term Storage (HBase/Cassandra/ HDFS)	Intel® Xeon® Gold 6138 Processor 20 cores	6		
	384 GB DRAM			
	4 TB drives: 14x 4 TB NL SAS 3.5 inch (56 TB total) Alternate HDD capacities available: 6 TB drives; 14x 6 TB NL SAS 3.5 inch (84 TB total) 8 TB drives: 14x 8 TB NL SAS 3.5 inch (112 TB total) 10 TB drives: 14x 10 TB NL SAS 3.5 inch (140 TB total)		14x	
	25 GbE/10 GbE NIC		2 ports	
	Intel® SSD DC P4500 Series 4 TB		Up to 8x	Alternative to HDD
Stream Ingestion & Search	Intel® Xeon® Gold 6138 Processor 20 cores	4		
	384 GB DRAM			
	Intel® Optane™ SSD DC P4800X Series 375 GB		1x	Elasticsearch
	Intel SSD DC P4500 Series 4 TB		1x	Kafka
	25 GbE/10 GbE NIC		2 ports	
	Intel® SSD DC P4500 Series 4 TB		1x	Alternative Elasticsearch SSD
User Interface & Management	Intel® Xeon® Gold 5118 Processor 12 cores	3		Cloudera Distribution of Hadoop Management Services and Kibana
	192 GB DRAM			
	Intel® SSD DC D3600 Series 1 TB		2x	
	25 GbE/10 GbE NIC		2 ports	

6.3.2 Master nodes

The master node is the nucleus of the HDFS and supports several other key functions that are needed on a Cloudera cluster by running a number of services such as YARN and ZooKeeper, among others. Table 3 lists the recommended components for a master node; these recommendations can be customized according to client needs. The Intel Xeon Scalable processors and minimum memory specified in Table 3 are recommended to provide sufficient performance as a Cloudera master node.

Table 3. Master node configuration

Component	Master node configuration
Server	ThinkSystem SR630
Processor	2x Intel® Xeon® Silver 4114 Processor 12-core 2.1 GHz
Memory - base	128 GB – 8x 16 GB 2666 MHz RDIMM
Disk (OS / local storage)	OS: Dual M.2 128 GB SSD with RAID 1 Data: 8x 2 TB 2.5” SAS HDD
HDD controller	ThinkSystem RAID 930-16i 4 GB Flash 12 Gb controller
Hardware storage protection	OS: RAID1 NameNode/Metastore: RAID 1 Database: RAID 10 ZooKeeper/QJN: No hardware protection; JBOD HDDs; multiple service instances across master nodes provide redundancy
Hardware management network	Integrated 1 GbE BaseT XCLARITY™ CONTROLLER (XCC) Interface
Data network adapter	Broadcom NetXtreme Dual Port 10 GbE SFP+ Adapter

7 Deployment considerations

This section describes real-time streaming considerations for deploying the Cloudera solution. For generic information on sizing the predefined cluster configurations, refer to the [Lenovo Big Data Validated Design for Cloudera Enterprise and VMware](), which discusses the following additional topics:

- Designing for lower cost
- Designing for high ingest rates
- Designing for in-memory processing with Spark
- Designing for Hadoop in a virtualized environment
- Estimating disk space
- Scaling considerations
- High availability considerations
- Migration considerations
- Selection of components like HDD controllers and data network adapters

7.1 Hardware optimizations

The following sub-sections provide recommendations for optimizing hardware performance.

7.1.1 Set the CPU governor mode

To check available CPU governors, run cpupower:

```
[root@master ~]# cpupower frequency-info
analyzing CPU 0:
  driver: intel_pstate
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency:  Cannot determine or is not supported.
  hardware limits: 1.20 GHz - 3.70 GHz
  available cpufreq governors: performance powersave
  current policy: frequency should be within 1.20 GHz and 3.70 GHz.
                  The governor "performance" may decide which speed to use
                  within this range.
  current CPU frequency: 1.21 GHz (asserted by call to hardware)
  boost state support:
    Supported: yes
    Active: yes
```

To check the current status of CPU governor, run the following:

```
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
```

It is recommended to set the governor to performance mode as well ensure that it is set so for all cores across all the nodes in the cluster.

7.1.2 Enable Intel® Turbo Boost Technology 2.0

It is highly advisable to enable Intel Turbo Boost Technology so that cores can operate at a higher frequency when required by bursty workloads. In general, we have seen better performance with turbo mode enabled. Ensure the scaling driver is set to intel_pstate across all cores:

```
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_driver
```

To get a snapshot of the processor's current state and debug idling cores, run:

```
[root@master ~]# turbostat --debug -P

Package    Core     CPU Avg_MHz   %Busy Bzy_MHz TSC_MHz     SMI  CPU%c1  CPU%c3
CPU%c6  CPU%c7 CoreTmp  PkgTmp Pkg%pc2 Pkg%pc3 Pkg%pc6 PkgWatt RAMWatt   PKG_%
RAM_%
```

7.1.3 Enable Intel® Hyper-Threading Technology

Because big data real-time streaming involves high-throughput data ingestion and processing, the availability of cores generally helps linear scalability. As long as the data pipelines do not rely heavily on caches, there will be no impact due to NUMA access, and hyper-threading will benefit the overall processing to a great extent. To check if hyper-threading is enabled, run `lscpu` or "`dmidecode –t processor`" and search for threads.

Note: For extremely latency-sensitive real-time streaming use cases, it may be advisable to avoid NUMA access and turn off hyper-threading, but this will impact the overall throughput and scalability.

7.1.4 Disable Transparent HugePages

"Transparent HugePages" (THP) is a Linux kernel feature intended to improve performance by making more efficient use of a processor's memory-mapping hardware. It is enabled (”enabled=always”) by default in most Linux distributions.

THP gives some applications a small performance improvement (at best about 10% but more typical improvements range from 0% to 3%). However, THP can also cause significant performance problems, or even apparent memory leaks. To avoid these problems, on the servers set THP to "enabled=madvise":

```
echo madvise | sudo tee /sys/kernel/mm/transparent_hugepage/enabled
```

Also set `transparent_hugepage=madvise` on the kernel command line (such as in /etc/default/grub).

This change will allow applications that are optimized for THP to obtain the performance benefits and prevent possible problems for other applications.

7.2 Software optimizations

7.2.1 Operating System Limits (ulimit)

- Open files: Because Spark executors rely heavily on opening file handles, failures might occur if this limit is low. Increase this value in /etc/security/limits.conf to 1048576 or higher.
- Max user processes: Similarly, increase this value to 1048576 in /etc/security/limits.conf for the Hadoop user (or the user running the Spark jobs).

7.2.2 Network Maximum Transmission Units (MTU)

Set MTU values to 9000, which signifies jumbo frames. Increasing the default MTU size can provide significant performance gains. This can be updated using /etc/sysconfig/network-scripts/ifcfg-<interface> and restarting the network service.

7.2.3 YARN Node Manager

Disable VMEM Check: Typically, long-running jobs might exceed the virtual memory limits enforced on containers and hence it is advisable to disable this check on YARN.

7.2.4 Spark

- Serializer: Spark can use the KryoSerializer library (version 2) to serialize objects more quickly. Kryo is significantly faster and more compact than Java serialization (often as much as 10x).
- Memory: There are several recommended memory optimizations:
  - YARN memory overhead: Using YARN as the resource manager for container orchestration requires memory to be provided to each container. Typically, we have set YARN memory overhead (spark.yarn.executor.memoryOverhead) between 786 MB to 1 GB so there are no issues during container management due to garbage collection or VMEM/PMEM bounds.
  - Dynamic allocation: As the workload throughput varies over time, it is advisable to use dynamic allocation. Dynamic allocation helps to scale up and down easily. The requirement here is to enable the Shuffle Service using the spark.shuffle.service.enabled flag.
  - Initial executors: If it is known that there will be a lot of throughput as soon as the task starts, set the spark.dynamicAllocation.initialExecutors flag to identify the initial number of executors. This value should be in between spark.dynamicAllocation.minExecutors and spark.dynamicAllocation.maxExecutors, which can be adjusted as necessary.

7.2.5 ZooKeeper

ZooKeeper and Kafka brokers must run on separate nodes to avoid resource contention.

7.3 Cluster scaling configurations

The following sections provide recommendations for sizing by rack capacity and by throughput. Which configurations are best for a particular deployment depend on specific workloads and performance requirements.

7.3.1 Sizing by rack capacity

Our experiments focused on financial services industry (FSI) fraud-detection use cases, where incoming and outgoing records were 512 to 1,024 bytes and 1 KB, respectively. Message sizes may vary for other verticals and use cases, depending on data complexity and machine-learning pipelines. Therefore, capacity planning should be performed to identify the correct rack configuration.

Table 4 shows recommended configurations and capacity. Usable storage assumes 3x replication. Kafka data retention can be configured as part of the overall CDH setup. The days of retention reflect the maximum duration data can be held until expiration. Holding data longer than this duration causes issues with storing new records, which may cause an offset exception to occur, due to unavailable offsets during commits. Typical data retention duration ranges from 1 hour to 12 hours for high-volume/high-throughput systems.

When choosing storage solutions, enterprises should consider end-to-end processing as well as data queries for insight and search. The choice between Intel Optane SSDs or Intel® SSD DC P4500 Series should balance budget and SLA latency requirements. As documented in "Maximize Data Value with a Real-Time Streaming Analytics Solution," an Intel Optane SSD can complete some tasks much faster than an Intel SSD DC P4500 Series; however Intel Optane SSDs are more expensive than 3D NAND SSDs. For more information, refer to the description of the Intel® SSD Data Center Family.

Table 4. Configurations for varying rack capacities

	Half Rack	**Full Rack**	**Multi-Rack (3x)**
Storage Capacity: 4 TB drives for stream processing @ 100 K transactions/second with 1% probability of fraud			
Raw Storage for Streaming	192 TB	336 TB	960 TB
Usable with 25% Reserve	48 TB	84 TB	240 TB
Maximum Kafka Data Retention	4.8 Days	8.4 Days	24 Days
Storage Capacity: 10 TB drives for stream processing @ 500 K transactions/second with 1% probability of fraud			
Raw Storage for Streaming	480 TB	840 TB	2400 TB
Usable with 25% Reserve	120 TB	210 TB	600 TB
Maximum Kafka Data Retention	2.4 Days	4.2 Days	12 Days
Node and Storage Configuration			
Streaming/Search Nodes (Kafka/Elasticsearch)	4	7	20
HDD per Streaming Node	12	12	12
Intel® Optane SSD or Intel® SSD DC P4500 Series (Search)	2	2	2
Compute Nodes (Spark Streaming)	5	10	35
Master Nodes	3	3	3
Total Processing Nodes in Rack	12	20	58

7.3.2 Sizing by throughput

In real-time big data architectures, streaming traffic is generally represented by throughput capacity. The throughput SLAs are typically configured using average load and burst/spike load. Hence, it is necessary to configure systems so that they can handle these requirements appropriately.

Throughput and latency optimizations often fall at opposite ends of the performance spectrum, because increasing throughput often results in higher end-to-end latency. Customers should assess their throughput and latency SLA needs and proceed with capacity planning accordingly. As mentioned in Table 5, Spark Streaming batch size is a configurable parameter in typical Spark Streaming applications. This duration affects the end-to-end latency, because Spark will wait for the time period to complete before it starts processing all records in that mini-batch. Note also that query latencies typically depend on query complexities as well as on the underlying storage framework.

It is important to consider end-to-end processing needs, because it can affect many use cases besides fraud detection, including measuring campaign effectiveness, offering targeted advertising and automated coupons, and identifying cross-selling opportunities. Overall, the reference architecture described here can accommodate many customer scenarios simply by configuring the batch size.

Table 5. Configurations for varying average and spike/burst throughput

Average Throughput (Messages/Second)	**100 K**	**500 K**	**1 Million**	**5 Million**
Spike Throughput (Messages/Second)	**300 K**	**800 K**	**2 Million**	**8 Million**
Raw Data Ingestion per day	10 TB	50 TB	100 TB	500 TB
Suggested Data Expiration	12 Hours	12 Hours	12 Hours	12 Hours
Streaming/Search Nodes	4	4	4	7
Storage Capacity for Streaming Nodes	4 TB Drive * 2 per node	4 TB Drives * 6 per node	4 TB Drives * 12 per node	10 TB Drives * 12 per node
Additional Storage for Search/Indexing	375 GB Intel® Optane SSD or 1 TB P4500 Series Intel SSD per node	2 * 375 GB Intel® Optane SSD or 1 TB P4500 Series Intel SSD per node	2 * 375 GB Intel® Optane SSD or 2 * 1 TB P4500 Series Intel SSD per node	2 * 375 GB Intel® Optane SSD or 2 * 1 TB P4500 Series Intel SSD per node
Processing Nodes	**5**	5	5	10
Rack Suggestion	Half Rack	Half Rack	Half Rack	Full Rack
End-to-End Processing Latency Measures (Configurable as Spark Streaming Batch Size)	5-20 seconds	20-30 seconds		
Search Latency Measures	Elasticsearch Exact Query Latency is about 8 ms on Intel® Optane™ SSDs (See section **Error! Reference source not found.** for more information)			

7.4 Security

As we deal with data both at rest and in-flight, security plays a pivotal role in any enterprise deployment. All open source components are well-integrated with Apache Kerberos and Sentry to provide authentication and authorization. Further group accesses and access control lists (ACLs) can be created; also, encryption can be used to enforce data usage policies. Integration with custom frameworks designed specifically for each enterprise will vary. However, integrations with OAuth based frameworks, LDAP Servers etc. are commonly done during deployment.

For complete documentation on security implementation on CDH, you may refer to the following document: https://www.cloudera.com/documentation/enterprise/latest/PDF/cloudera-security.pdf.

7.5 Additional notes on setup

Spark 2 and Kafka 0.10 need to be installed separately after complete CDH installation. It is necessary to create Spark and Kafka gateway roles on all hosts. Typically, JBOD is used to set up the disk volumes, and these mount points are used by HDFS to store data with 3x replication. Versioning of other services are based on the default packaging of Cloudera CDH 5.10 parcels. For version details, refer here.

7.5.1 Kafka Manager

We found Kafka Manager to be a useful tool to monitor Kafka topics and broker health and metrics. To obtain Kafka Manager, use one of the following methods:

- Clone this Repo: https://github.com/yahoo/kafka-manager and checkout tag 1.3.3.6
- Download these bits: https://github.com/yahoo/kafka-manager/releases/tag/1.3.3.6

To build Kafka Manager, run:

```
./sbt -Dhttp.proxyHost=[proxy_uri] -Dhttp.proxyPort=[proxy_port] -
Dhttps.proxyHost=[proxy_uri] -Dhttps.proxyPort=[proxy_port] clean dist
```

Note: sbt is bundled with the download. Downloading Scala bits takes time. Wait for 1-2 minutes based on your network bandwidth for some downloads to show up. You may have to delete some earlier Ivy Cache (~/.ivy2/cache) if there are issues with resolving certain libraries due to version conflicts.

Once the build is successful, follow these steps:

1. unzip target/universal/kafka-manager*.zip and cd the extracted directory.
2. export ZK_HOSTS=[ZK_QUORUM]
3. Run bin/kafka-manager -Dhttp.port=9090
4. Open the WebUI: http://localhost:9090

Now add the KafkaCluster:

1. Put in Cluster ZooKeeper hosts
2. Choose Kafka version 0.10.1.0
3. Enable JMX Polling (leave the other settings as default)

We found it very helpful to add new instances to Kafka and repartition the topics using the Kafka Manager user interface. When new instances are added, make sure the topics are balanced across all brokers, which may entail reassigning the partitions.

7.5.2 Cloudera parcel versions

- CDH5: 5.10.1-1.cdh5.10.1.p0.10
- KAFKA: 2.1.0-1.2.1.0.p0.115
- SPARK2: 2.1.0.cloudera1-1.cdh5.7.0.p0.120904

Run Solr/ELK and other indexing engines on separate nodes, because they consume considerable CPU and memory resources.

8 Bill of Materials

The entire Bill of Materials (BOMs) for different configurations of hardware for this reference architecture can be found in the Lenovo Big Data Validated Design for Cloudera Enterprise and VMware. The BOM includes the part numbers, component descriptions, and quantities. In general, the BOMs are provided for the following categories of infrastructure:

- Master nodes
- Worker nodes
- System management nodes
- Management network switches
- Data network switches
- Racks
- Cables

Note that BOM lists are not meant to be exhaustive and must always be verified with the configuration tools. Any discussion of pricing, support, and maintenance options is outside the scope of this document.

BOM information is for the United States; part numbers and descriptions can vary in other countries. Other sample configurations are available from your Lenovo sales team. Components are subject to change without notice.

9 For more information

This reference architecture document has benefited very much from the detailed and careful review comments provided by colleagues at Lenovo, Intel, and Cloudera. Use the following information to gain more technical information about the various components of this reference architecture:

Lenovo

- Ajay Dholakia – Sr. Engineering Staff Member, Chief Architect for Big Data and AI Solutions
- Prasad Venkatachar – Sr. Solutions Product Manager

Cloudera

- Sean Gilbert – Director of Partner Sales

Intel

Visit https://www.intel.com/content/www/us/en/big-data/big-data-analytics-turning-big-data-into-intelligence.html

Resources

The Lenovo Big Data Validated Design for Cloudera Enterprise and VMware contains an extensive list of resources for Lenovo, Cloudera, and open source software topics. In addition, you can visit the following sites for information.

Intel:

- Intel Xeon Scalable Processors: https://newsroom.intel.com/press-kits/next-generation-xeon-processor-family
- Intel Data Center Information: http://www.intel.com/performance/datacenter
- Intel® SSD Data Center Family: https://www.intel.com/content/www/us/en/products/memory-storage/solid-state-drives/data-center-ssds.html
- Intel Financial Services Solutions: http://intel.com/FSI

Cloudera:

- References, analytics capabilities, and use case content: https://www.cloudera.com/products/operational-db.html

Other open source software resources:

- Apache Spot: http://blog.cloudera.com/blog/2016/09/spot-fighting-cyber-threats-via-an-open-data-model/
- Elasticsearch: https://www.elastic.co/guide/en/elasticsearch/reference/5.5/query-dsl.html
- Hadoop Virtualization Extensions (HVE): https://issues.apache.org/jira/browse/HADOOP-8468
- Kafka: http://kafka.apache.org
- Solr: https://cwiki.apache.org/confluence/display/solr/Taking+Solr+to+Production

Document history

Version 1.0	05 January 2018	• First version
Version 1.1	23 September 2018	• Fixed typographical errors

Trademarks and special notices

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used Lenovo products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-Lenovo products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by Lenovo. Sources for non-Lenovo list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. Lenovo has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-Lenovo products. Questions on the capability of non-Lenovo products should be addressed to the supplier of those products.

All statements regarding Lenovo future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local Lenovo office or Lenovo authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in Lenovo product announcements. The information is presented here to communicate Lenovo’s current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard Lenovo benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

Any references in this information to non-Lenovo websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this Lenovo product and use of those websites is at your own risk.