Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

ORACLE Diamond Partner

The advantage of using an Oracle Exalogic-based Big Data strategy for the acquisition of streaming data

Oracle-based Big Data Strategy

Introduction

By definition, Big Data is massive. But with constantly increasing numbers of smart connected devices, more and more sensors in machines, and the continuing spread of information technologies worldwide in relation to the internet of things, even that definition is an understatement.

The size of Big Data is overwhelming as is its storage. But today, Big Data is no longer simply about size and it is also no longer about storage. Putting it to productive use has gone beyond simply storing it or doing basic analysis on it. To make the most of Big Data, the question to ask is no longer, “Where do we put it?” but rather, “How do we use it?”

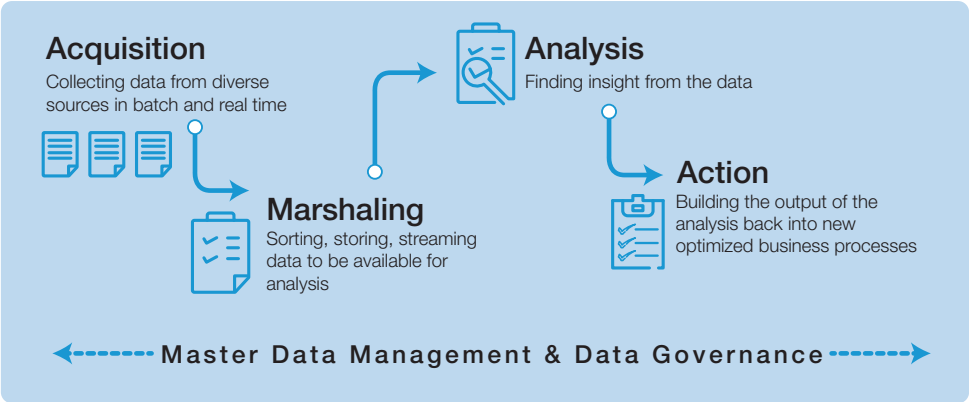
Another important question – and one that is often overlooked – is: “How do we acquire the data?” Implementing a technological strategy for acquisition of streaming, real-time data and batch-oriented streaming data is the key to a successful Big Data acquisition strategy. Streaming data refers to all “data creating data sources” and not to “data containing data sources” as will be described in this document.

Within this document, Capgemini provides a high-level blueprint on how acquisition of streaming data can be achieved with an Oracle-based Big Data strategy. Compared to home grown solutions, Oracle technology – namely, the portfolio of Oracle engineered systems – offer much greater deployment and management ease as well as lower total cost of ownership. Capgemini provides a large portfolio of services and solutions in the field of Big Data with multiple vendors among which Oracle is one of the key technology partners.

1. Big Data Lifecycle Flow

In general, the Big Data lifecycle flow consists of four major steps. Capgemini identifies the below mentioned steps in the lifecycle of Big Data; acquisition, marshalling, analysis and action (Figure 1).

Figure 1: Master Data Management and Data Governance



1.1 Big Data Acquisition

Collecting data from diverse sources in batch and real time. While your Big Data collection can be created from databases within your company, it can also include the unstructured streaming data coming from social media networks, websites, or sensor networks within an industrial factory. From an internet of things and connected devices perspective, IT must deal with a wide range of issues. At the network edge they need to deal with disparate device types, OS/platforms and data complexities

1.2 Big Data Marshalling

Sorting, storing, streaming data to be available for analysis. When data is acquired during the acquisition phase it is commonly unstructured data and in most cases not ready for immediate analysis. For this the process of 'marshalling' is used. In this process, acquired data is cross referenced and mapped against data in the Master Data Management repository. This process ensures a higher quality of data to be analyzed and an easier process of conducting meaningful analysis and taking automated actions upon the analysis.

1.3 Big Data Analysis

Finding insight from the data. The analysis of combined data sources, both streaming real-time data as well as data that is at rest and within your company for years, is the main driving force within a Big Data strategy. Often this is inaccurately seen as the primary technical part of a Big Data solution while forgetting the other components that are needed to create a full end- to-end Big Data solution. Within Big Data Analysis techniques like Map Reduce and technological solutions like Apache Hadoop®, Hadoop Distributed File System (HDFS), and in-memory analytics come into play to turn the acquired and marshaled data into meaningful insights for the business.

1.4 Big Data Action

Building the output of the analysis back into new optimized business processes. The business can optimize the Big Data inputs by acting upon the analysis directly in real time or acting upon the newly acquired insights to change the middle and long-term strategies of a company.

1.5 Master Data Management and Data Governance

Mastering data management and conducting proper data governance are important to the success of a Big Data strategy. While not directly a part of the Big Data strategy, the success of a Big Data strategy and the implementation of it are heavily dependent on Master Data Management and data governance. Master Data Management and data governance are key to understanding the meaning of acquired data and interpreting results that wll impact business processes and entities in the widest sense possible.

2. Big Data Acquisition

Big Data Acquisition concerns how you can acquire your data into your Big Data collection; sources can be databases within your company as well as the unstructured streaming data coming from social media networks, websites, or sensor networks within an industrial factory.

2.1 Understanding Data Sources

A data source can be considered everything that contains data or is creating data.

Sources containing data, for example, would include a database or data-store where the data is at rest up to a certain level. The data contained in the database or data-store is not necessarily static but can also be dynamic. An ERP database, which by nature is an OLTP (Online Transactional Processing) database, is considered a data-containing data source.

A data-creating data source creates the data and transmits it directly or stores it for a limited amount of time. Consider a sensor, for example — as soon as the sensor is triggered or has executed a timed measurement it will broadcast the information. As a deviation on this, there are sensors that will store the collected data in a queue until the moment it is collected by a collector. The sole purpose of the device, however, is not storing the data but rather creating the data.

Understanding the concept and the differences between "data-creating data sources" and "data- storing data sources" is vital when planning a data acquisition strategy. All data-creating data sources are considered streaming data.

When developing a data acquisition strategy as part of an enterprise-wide Big Data strategy, a number of data sources will commonly be identified by data consumers who have an idea as to which data they would like to collect for future analysis. At the same time it is commonly seen that this is only a fraction of all data that is created by systems and sensors within the enterprise. Big Data is about collecting all of this data. Even though some of the data might not directly be seen as valuable, it is very likely that it will become valuable at a later stage. When developing a data-acquisition strategy as part of a Big Data strategy, the input from data consumers can be included, however it should never be seen as the full set of data that needs to be collected, but rather, it should be treated as a starting point.

2.2 Streaming Data Acquisition

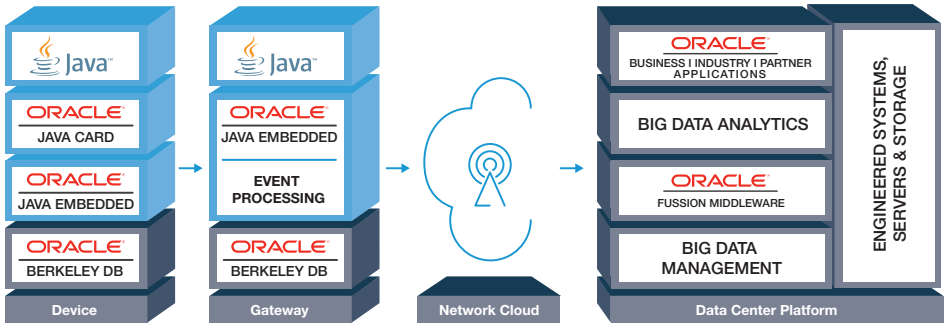
Streaming data is considered all data that is originating from a data-creating data source, rather than a data-containing data source. Streaming data is by its very nature volatile data and needs to be captured as part of the data creation process or very shortly after the creation of the data.

A streaming-data source that produces data will not hold the data after it is collected or broadcast. Therefore, it is essential to capture this data correctly as there is no option to re-capture a second time.

Even though there are a large number of technologies and standards available for sending data from a source to a destination, it is considered a Capgemini best practice to make use of the TCP/IP and HTTP(S) protocol for transporting data between the two.

This best practice is in line with Oracle strategies such as Oracle's internet of things platform (Figure 2). In this strategy, the device/sensor side is included as part of a full Oracle stack. In this document we do not incorporate this in detail and we will focus on the strategy after the device/sensor has created data which need to be captured and analyzed.

Figure 2: Oracle's Internet of Things Platform (Source: Oracle)



2.3 Streaming Data Push Acquisition

A streaming-data push acquisition strategy is based upon sources that are capable of pushing the data to the receiver. Commonly this is done by the data source calling a webservice at the receiver side. After confirmation of receiving the data, the data is removed from the queue at the data-creation side.

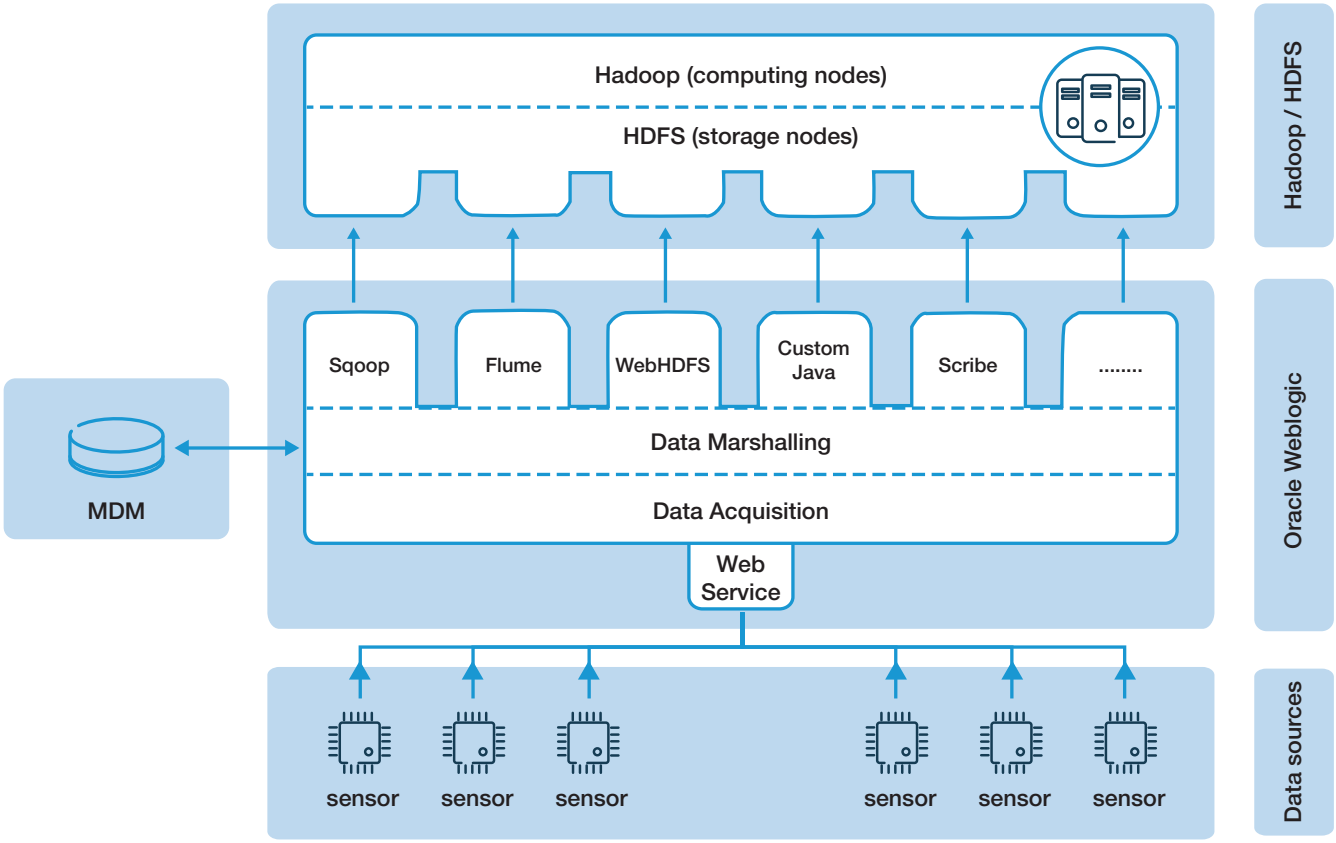
A streaming-data push strategy is often introduced when a device is unable to hold a large set of data or in cases where the data is not created regularly, however, needs to be acted upon the moment it is created. This is similar to triggered sensors like an open/close valve sensor or vibration sensor in an industrial installation or even a scanning device in a shop. Both examples do not produce data on a predefined schedule like a temperature sensor would do. However, when they do produce data, the data needs to be acted upon in most cases directly and the sensor and/or device is not capable of holding a large set of data.

A number of technologies are available and proposed for pushing data from a data creator to a data receiver. The initial webservice based thinking for capturing streaming data is based upon a paper from NASA's Jet Propulsion Laboratory in which the initial idea of massive sensor based data capturing with webservices was promoted.

Figure 3 shows a high-level representation of a setup for capturing streaming data sent out by sensors. Sensors are an example in this case. Any data creating source that is broadcasting its data to an endpoint can be added to this example; for example streaming data from social networks.

In Figure 3 all data producers send the collected data directly to the webservice endpoint the moment the data is created at the source. The endpoint can be an SOAP-based webservice, a RESTFul-based webservice, or any other technology desired, however a commonly used technology based upon HTTP(S) standards is preferred.

Figure 3: Direct Webservice Endpoint



After collection, data received by the webservice can be tagged and enriched during the data-marshalling process. This ensures that incoming data is written in a more enriched format to HDFS (Hadoop Distributed File System). This is commonly a single action per data point. By directly adding the meta-data to the data, there will be no need to sacrifice computing power on every cycle within your Hadoop cluster for adding this data "on the fly" during Hadoop job executions.

Data received and enriched during the marshalling process can then be written to HDFS where it will become available for Hadoop for future processing and analysis. There are a large number of options available to write data to HDFS. Depending on a large number of factors, the preferred writing mechanism can differ, for example, the speed and amount of data that needs to be written, the fact that you can, and may, want to buffer data in your webservice server, and the already existing and used technologies in your Hadoop/HDFS cluster.

2.4 Streaming Data Pull Acquisition

A streaming-data, pull-acquisition strategy is based upon sources that are incapable of pushing the data to the receiver, however, they do have the capability to hold a relatively large amount of data within the source and this data can be accessed remotely for data retrieval. Commonly, this is done by a scheduled process that executes a request to the data sources and receives the returning data from the requested sources.

As an example, the source can be accessed by calling the known address of a sensor and retrieving the data in an XML or JSON format. This strategy calls for the need to store all the addresses of the data sources and to schedule an acquisition request on a regular basis.

Figure 4: Streaming Data Pull-Acquisition Strategy

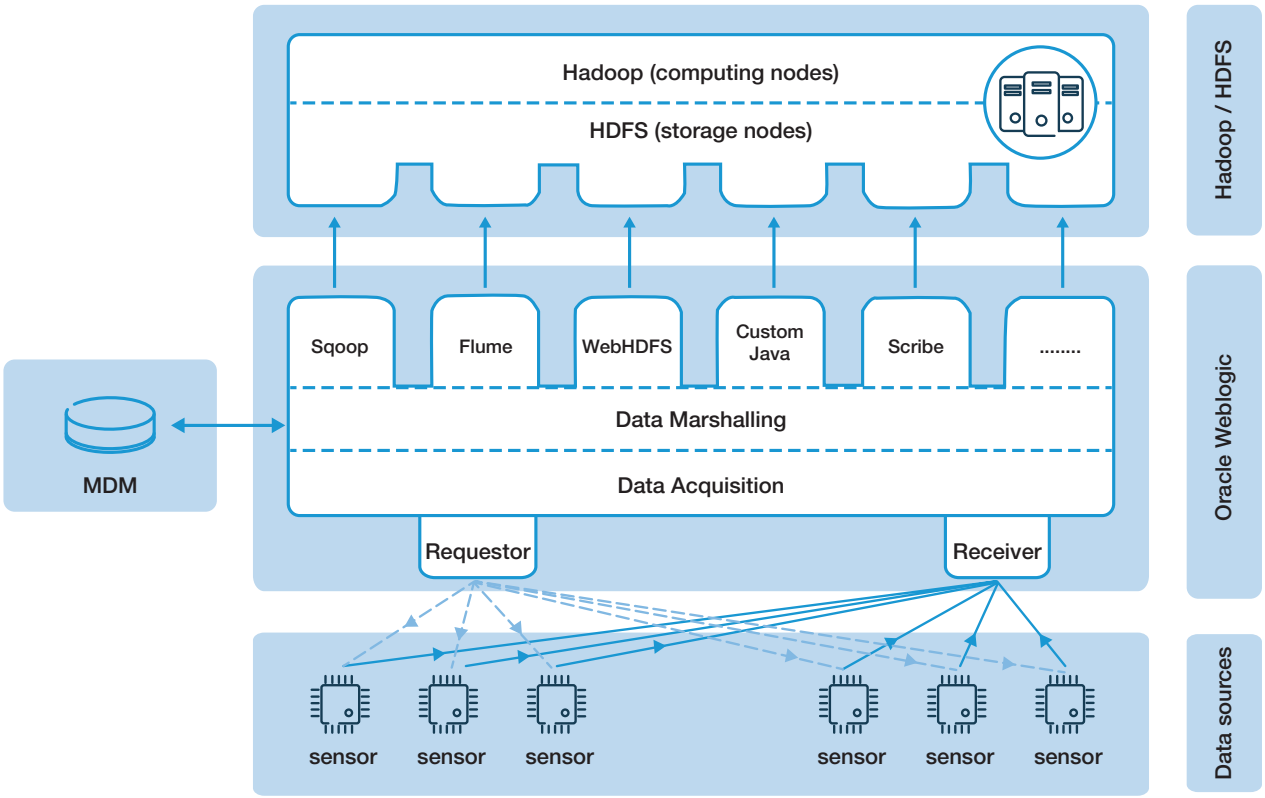


Figure 4 represents a streaming-data, pull-acquisition strategy. Here you can see the request and response mechanism. The requestor is triggered by the scheduler to request a specific source for the data. When the source can be reached, a sensor in this example, responds by returning the data to the receiver. The receiver will forward the data to the marshalling process where the data can be tagged and enriched before it is written to HDFS.

The form in which the communication with the data source is conducted depends largely on the capabilities of the source. Preferred solutions are making use of a HTTP(S)-based webservice technology. This can be either an SOAP or RESTFul based webservice. However, due to the diversity of "data-storing data sources" a large number of communication technologies are available. For example, SNMP (Simple Network Management Protocol) is a commonly seen technology for devices and sensors.

The benefit of a streaming-data pull acquisition is that the source can hold the data for a longer time and can be downloaded in a large(r) batch. This means that this strategy is also applicable in situations where the data source is not connected to a network at all times, for example, sensor data coming from devices used during transportation or in equipment which is regularly out of reach of a network. Streaming-data acquisition architectural considerations should take into consideration if data sources will always be connected or can have periods of time during which they are not reachable for acquisition.

2.5 Big Data Acquisition and Data Marshaling

Data acquired via a pull or a push mechanism will come into your data acquiring solution at one point in time and is most likely not yet related to already existing data points. It can be extremely beneficial for downstream-analysis processes to tag and enrich this data before it is written to your HDFS storage.

For example, you might get consumer information and purchase information from a loyalty card which is only tied to the ID of the loyalty card. While the collected data is reflective of the customer, physically it is not yet tied to this customer, so it only provides an ID which can be used to make the link. If, for example, you have a process that is analyzing purchase differences based upon time and gender of the person who made the purchase you will have to verify the gender associated with this specific loyalty card ID.

Another way of doing this is to enrich this record with the gender information in combination with other often used information, and add this to one data structure before writing it to HDFS. In this case, you use a little more storage, however, you save on CPU cycles needed for checking the gender every time an analytical process is executed.

The gender information in combination with a loyalty card ID is only a simplified example. Numerous additional information sets can be added to an incoming data set. Information commonly is queried from a Master Data Management store or in some cases from other data sources containing data that is beneficial from a data enrichment point of view.

2.6 Big Data Acquisition Architectural Decisions

The acquisition of Big Data, whether social data, sensor data, log data, transactional data, or any other data source, is an important part of the overall Big Data solution architecture. Your acquisition architecture and strategy have an important impact on the architecture of both your downstream Big Data strategy, as well as your upstream data- creation architecture.

There are some architectural questions that should be taken into consideration within the realm of data acquisition. For example:

- Are my data sources primarily data-creating data sources or data-containing data sources?
- What are the capabilities for pull or push acquisition?
- What is the speed required for updates within my Big Data collection?
- What is the amount of data I will acquire in a certain time frame?

Understanding your data sources, the capabilities of the sources, and the demand of the downstream processes will influence your acquisition architecture and at the same time drastically influence architectural decisions in both downstream and upstream architecture. Due to this influence, it is important that acquisition is taken into account within the overall architecture and is not considered an afterthought.

3. Acquisition and Oracle Exalogic

3.1 Introduction to Oracle Exalogic Elastic Cloud

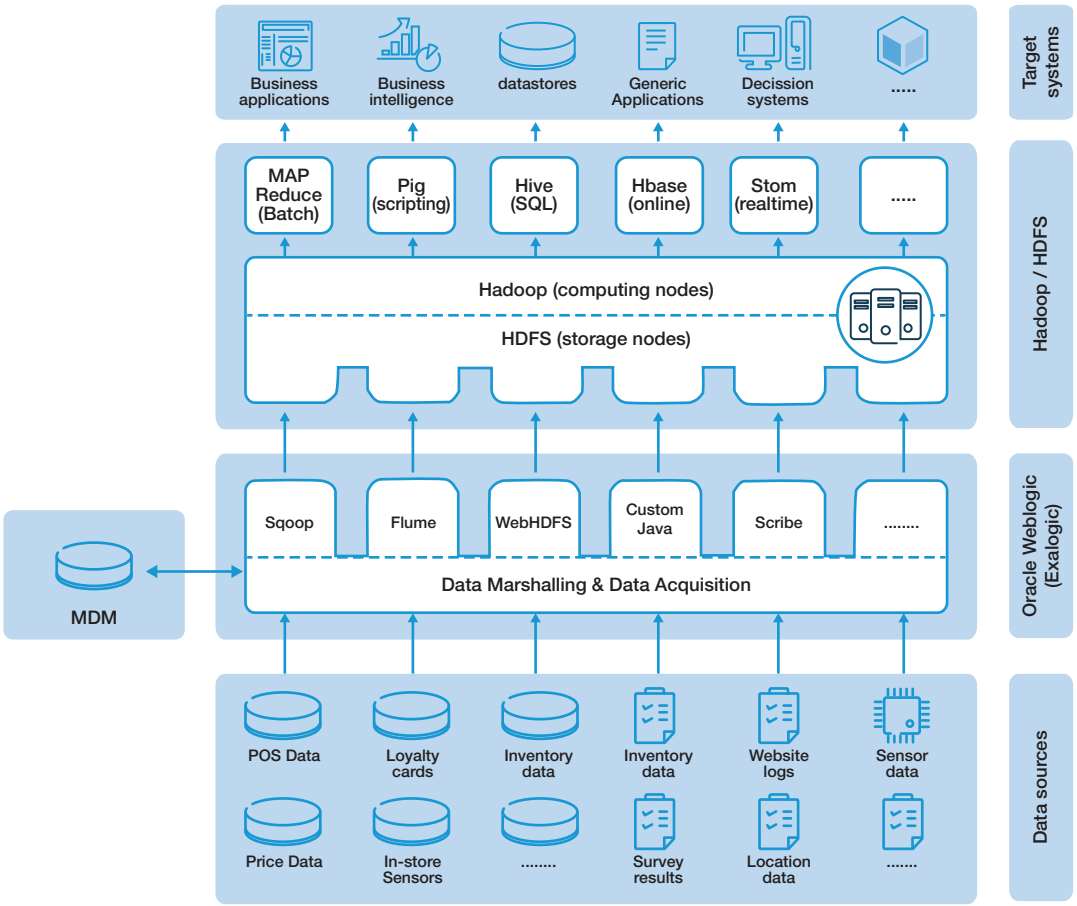
The Oracle Exalogic Elastic Cloud is an Engineered System. It consists of software, firmware, and hardware, on which enterprises may deploy Oracle business applications, Oracle Fusion Middleware, or software products provided by Oracle partners, as well as custom-built applications. Oracle Exalogic is designed to meet the highest standards of reliability, availability, scalability, and performance for a wide variety of time-sensitive and mission-critical workloads.

Oracle Exalogic dramatically improves performance of standard Linux, Solaris and Java applications without requiring code changes and reduces costs across the application lifecycle, from initial setup to ongoing maintenance, as compared to conventional enterprise-application platforms and private clouds assembled from disparate components sourced from multiple vendors.

Oracle Exalogic is an open system assembled by Oracle from its portfolio of standards-based, best-of-breed component products and technologies. The Oracle Exalogic system reflects best-practices acquired and refined from thousands of customer deployments and extensive R&D effort.

In the overall deployment, as shown in Figure 5, the Oracle Exalogic engineered system is positioned as a data receieving front before your Hadoop/HDFS implementation. Oracle provides the Oracle Big Data Appliance which can take the role for operating Hadoop/HDFS implementation; however a customer-build solution can also be implemented here.

Figure 5: Overall Deployment Strategy



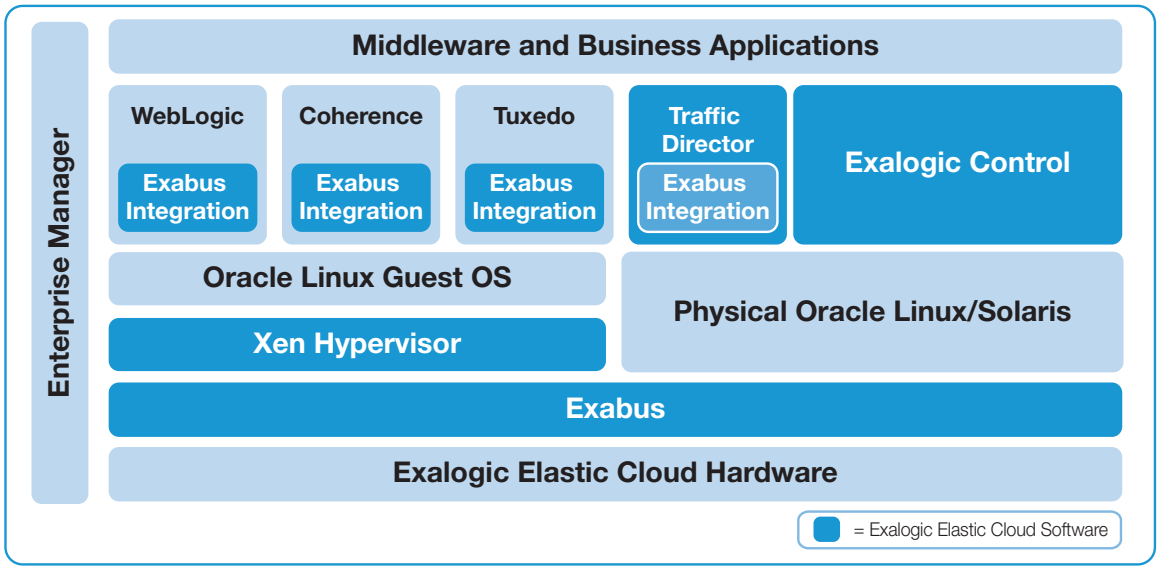
3.2 Aquisition with Oracle Exalogic

The Oracle Exalogic engineered system is primarily developed as part of the Oracle Engineered Systems portfolio to enable customers to run applications on an engineered system. Oracle Exalogic is primarily designed to run Oracle WebLogic Server. Within the described Big Data pull acquisition, Big Data pull-acquisition webservices and Java-based applications are positioned to be a preferred choice.

Regardless of whether you are implementing a pull acquisition or a push acquisition, or a mix of both, an Oracle Exalogic platform can provide a big benefit for deploying your solution on Oracle Engineered Systems.. The Oracle Exalogic engineered system has been designed to be the basis for an elastic-cloud platform specifically designed for running Oracle WebLogic Servers.

Figure 6 provides insight into the high level architecture of Oracle Exalogic. When deploying a Big Data acquisition strategy as shown in Figure 5, the primary components used will be the Oracle WebLogic Server component and the Oracle Traffic Director—preferably also using the Oracle Linux and Java Virtual Machine (JVM) combination to ensure you can create multiple virtual environments per physical node in your Oracle Exalogic engineered system.

Figure 6: Oracle Exalogic Architecture



Oracle Exalogic provides the option to make a decision if you want to run a physical installation, meaning one operating system installation per physical node, or running your software on top of a hypervisor. Most customers select the option of using the virtualized option opposed to the single operating system per physical node. The reason for this is that it is providing a more flexible, cloud-based infrastructure.

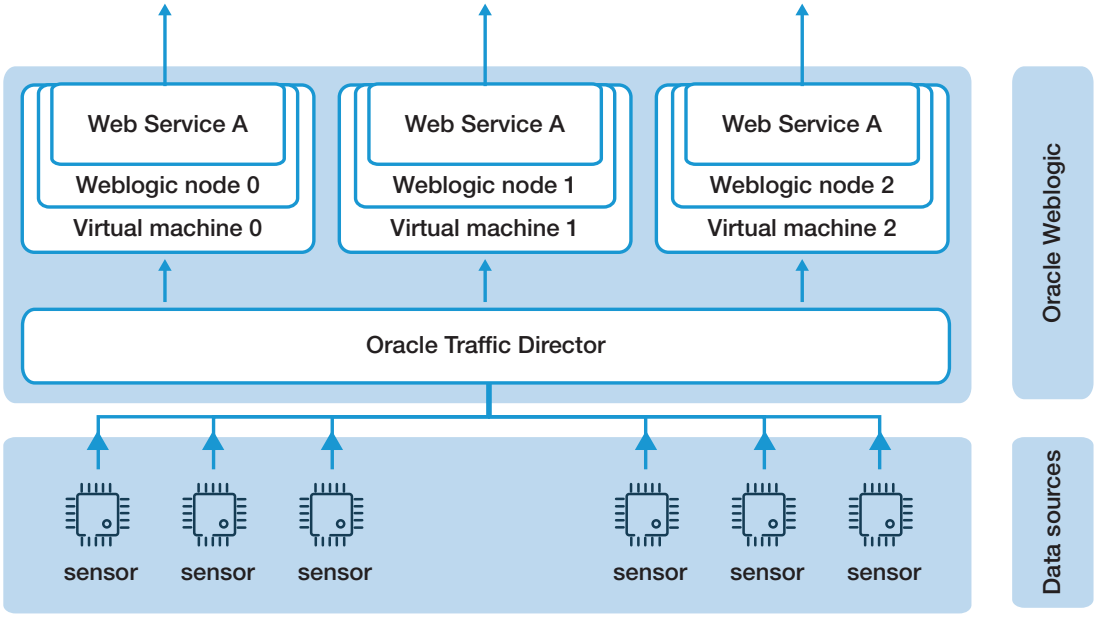
3.3 Example Blueprint Deployment

Figure 7 example blueprint is based on a theoretical customer situation in which the customer has the need to capture, consolidate, and analyze the data from thousands of sensors that are deployed throughout the customer's manufacturing machinery.

Within this scenario all sensors have the capability to push the information to a central location. All sensors are designed to send out information every 20 seconds, which comes down to 4,320 information packages per sensor that need to be captured. As this specific customer has thousands of sensors per factory and is operating multiple factories, the total amount of information packages that need to be received on a daily basis per second is staggering.

Oracle Exalogic can be deployed to enable the customer to create a platform that is capable of handling the amount of data sent and that needs to be processed in a Big Data strategy. Oracle Exalogic For this specific scenario, Oracle Exalogic has been configured to work together with Oracle JVM which enables the customer to run multiple, virtualized instances of Oracle Linux per physical compute node.

Figure 7: Network Load Balancing for Big Data



In the above example (Figure 7), all sensors are pointing towards a single point of entry, the Oracle Traffic Director. The Oracle Traffic Director receives all input via standard ethernet connections and uses the optimized InfiniBand fabric to distribute and load-balance the incoming data to a number of virtual machines on the compute nodes that run the webservices for capturing within the Oracle WebLogic Servers.

To ensure high availability, the Oracle Traffic Director consists out of a dual-node setup where both nodes are running on a different physical compute node within the Oracle Exalogic engineered system. This means that if one of the Oracle Traffic Director nodes fails, the surviving Oracle Traffic Director will take over even if one of the physical servers has malfunctioned.

The virtual servers running an Oracle WebLogic instance are distributed over multiple physical nodes to ensure the same manner of high availability as the Oracle Traffic Director. In this setup, one or more instances can fail while the capturing process of incoming data remains unharmed.

When data is captured by an Oracle WebLogic Server, the resulting data is written to HDFS where it can be processed by Hadoop. In this example blueprint, the HDFS implementation is running on an Oracle engineered system. The Oracle Big Data Appliance engineered system is capable of connecting to an Oracle Exalogic machine via InfiniBand. Communication between capturing and the storing and processing of Big Data is done based on InfiniBand which greatly speeds the process. As networking speed is one of the main bottlenecks in high performance computing and data-intensive computing; the benefits of the InfiniBand high-speed throughput are directly adding to the overall performance of the solution.

By using an Oracle Exalogic machine for capturing the data, an Oracle Big Data Appliance for storing and processing the data, and potentially other engineered systems for database capabilities and business intelligence, you can create a complete infrastructure based upon Oracle Engineered Systems for the entire Big Data lifecycle. By utilizing Oracle Engineered Systems, you benefit from the fact that Oracle has engineered its systems to work together and can connect to each other with high-speed connections based upon InfiniBand.

About the author

Johan Louwers

Johan Louwers has worked as a developer, administrator, manager, project manager, managing consultant and senior architect within several IT companies and IT departments. He specializes in Oracle technology, infrastructure technology, and IT strategy and has been advising and actively working with a large range of customers and companies to help enterprises excel in their day-to-day operations and provide them with cutting-edge technology and solutions.

He is currently a managing consultant and senior architect at Capgemini. Specialized in Oracle technology, infrastructure solutions, and cloud computing, Johan has been selected by Capgemini to be part of the global Capgemini Expert Connect program and is considered a thought leader, providing active advice and support to enterprises around the globe.

He is one of Capgemini's leading global resources on Oracle Engineered Systems and converged infrastructure in combination with Big Data.

He also spearheads Capgemini cloud initiatives around Oracle Run, a cloud-based hosting platform specifically designed within the Capgemini datacenters to provide an Oracle optimized cloud platform for customers using Oracle VM, Oracle Linux and Oracle Enterprise Manager.

Johan actively maintains a blog, johanlouwers.blogspot.com, and he is an active developer and researcher in the fields of Big Data, map-reduce, Hadoop, HDFS, Linux technology and datacenter optimalization and has interest in security, database technology, and open source technology.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

ORACLE Diamond Partner

For more information about Capgemini's offerings for Oracle Engineered Systems, visit:

www.capgemini.com/oracle-engineered-systems

About Capgemini

With almost 145,000 people in over 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2014 global revenues of EUR 10.573 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Learn more about us at
www.capgemini.com

The information contained in this document is proprietary. ©2015 Capgemini. All rights reserved.
Rightshore® is a trademark belonging to Capgemini.

MCOS_GI_AH_20150220