Big Data: Practical and Real Use Cases

Jean-Pierre Dijcks
Oracle
Redwood City, CA, USA

Keywords
Big Data, Data Warehouse, ETL, Hadoop, NoSQL, Analytics, Big Data Appliance

Introduction

The goal of this paper is to explain in a few succinct patterns how organizations can start to work with big data and identify credible and doable big data projects. This goal is achieved by describing a set of general patterns that can be seen in the market today.

Big Data Usage Patterns

The following usage patterns are derived from actual customer projects across a large number of industries and cross boundaries between commercial enterprises and public sector. These patterns are also geographically applicable and technically feasible with today's technologies.

This paper will address the following four usage patterns:

- Data Factory – a pattern that enable an organization to integrate and transform – in a batch method – large diverse data sets before moving this data into an upstream system like an RDBMS or a NoSQL system. Data in the data factory is possibly transient and the focus is on data processing.
- Data Warehouse Expansion with a Data Reservoir – a pattern that expands the data warehouse with a large scale Hadoop system to capture data at lower grain and higher diversity, which is then fed into upstream systems. Data in the data reservoir is persistent and the focus is on data processing as well as data storage as well as the reuse of data.
- Information Discovery with a Data Reservoir – a pattern that creates a data reservoir for discovery data marts or discovery systems like Oracle Endeca to tap into a wide range of data elements. The goal is to simplify data acquisition into discovery tools and to initiate discovery on raw data.
- Closed Loop Recommendation and Analytics system – a pattern that is often considered the holy grail of data systems. This pattern combines both analytics on historical data, event processing or real time actions on current events and closes the loop between the two to continuously improve real time actions based on current and historical event correlation.

Pattern 1: Data Factory

The core business reason to build a Data Factory as it is presented here is to implement a cost savings strategy by placing long-running batch jobs on a cheaper system. The project is often funded by not spending money on the more expensive system – for example by switching Mainframe MIPS off - and instead leveraging that cost savings to fund the Data Factory. Figure 1 shows a simplified implementation of the Data Factory.

As Figure 1 shows, the data factory must be scalable, flexible and (more) cost effective for processing the data. The typical system used to build a data factory is Apache Hadoop or in the case of Oracle's Big Data Appliance – Cloudera's Distribution including Apache Hadoop (CDH).

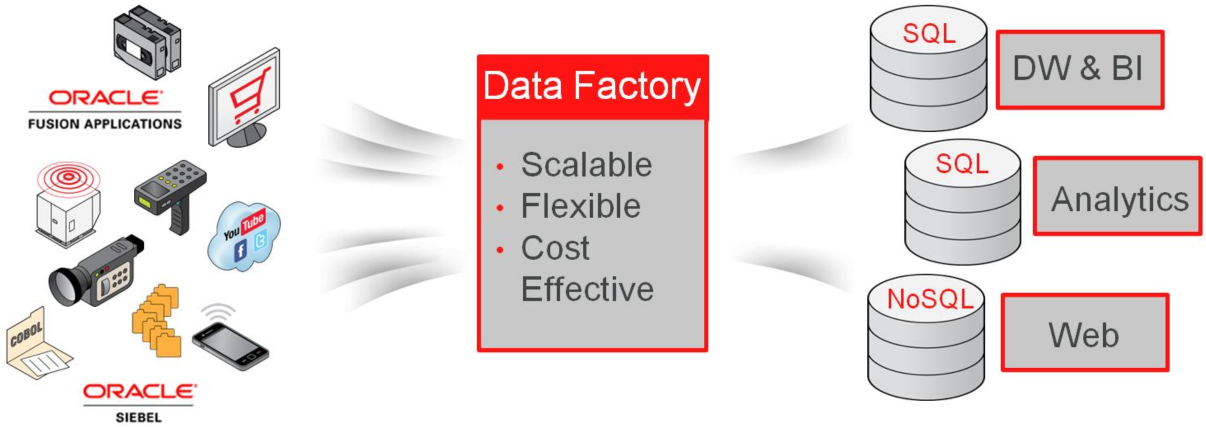

Figure 1 Data Factory

Hadoop (and therefore Big Data Appliance and CDH) offers an extremely scalable environment to process large data volumes (or a large number of small data sets) and jobs. Most typical is the offload of large batch updates, matching and de-duplication jobs etc. Hadoop also offers a very flexible model, where data is interpreted on read, rather than on write. This idea enables a data factory to quickly accommodate all types of data, which can then be processed in programs written in Hive, Pig or MapReduce.

As shown in Figure 1 the data factory is an integration platform, much like an ETL tool. Data sets land in the data factory, batch jobs process data and this processed data moves into the upstream systems. These upstream systems include RDBMS's which are then used for various information needs. In the case of a Data Warehouse, this is very close to pattern 2 described below, with the difference that in the data factory data is often transient and removed after the processing is done.

This transient nature of data is not a required feature, but it is often implemented to keep the Hadoop cluster relatively small. The aim is generally to just transform data in a more cost effective manner.

In the case of an upstream system in NoSQL systems, data is often prepared in a specific key-value format to be served up to end applications like a website. NoSQL databases work really well for that purpose, but the batch processing is better left to Hadoop cluster.

It is very common for data to flow in the reverse order or for data from RDBMS or NoSQL databases to flow into the data factory. In most cases this is reference data, like customer master data. In order to process new customer data, this master data is required in the Data Factory.

Because of its low risk profile – the logic of these batch processes is well known and understood – and funding from savings in other systems, the Data Factory is typically an IT department's first attempt at a big data project. The down side of a Data Factory project is that business users see very little benefits in that they do not get new insights out of big data.

Pattern 2: Data Warehouse Expansion

The common way to drive new insights out of big data is pattern two. Expanding the data warehouse with a data reservoir enables an organization to expand the raw data captured in a system that is able to add agility to the organization. The graphical pattern is shown in Figure 2.

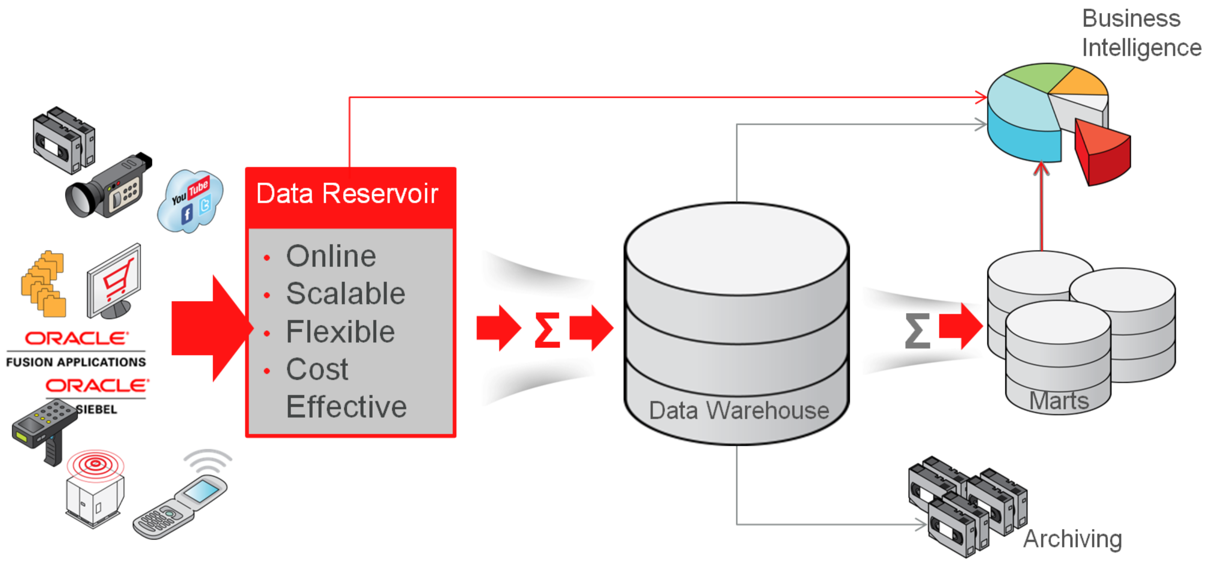

Figure 2 Data Warehouse Expansion with a Data Reservoir

A Data Reservoir – like the Data Factory from Pattern 1 – is based on Hadoop and Oracle Big Data Appliance, but rather then have transient data and just process data and then hand the data off, a Data Reservoir aims to store data at a lower than previously stored grain for a period much longer than previous periods.

The Data Reservoir is initially used to capture data, aggregate new metrics and augment (not replace) the data warehouse with new and expansive KPIs or context information. A very typical addition is the sentiment of a customer towards a product or brand which is added to a customer table in the data warehouse.

The addition of new KPIs or new context information is a continuous process. That is, new analytics on raw and correlated data should find their way into the upstream Data Warehouse on a very, very regular basis.

As the Data Reservoir grows and starts to become known to exist because of the new KPIs or context, users should start to look at the Data Reservoir as an environment to “experiment” and “play” with data. With some rudimentary programming skills power users can start to combine various data elements in the Data Reservoir, using for example Hive. This enables the users to verify a hypotheses without the need to build a new data mart. Hadoop and the Data Reservoir now becomes an economically viable sandbox for power users driving innovation, agility and possibly revenue from hitherto unused data.

Pattern 3: Information Discovery

Agility for power users and expert programmers is one thing, but eventually the goal is to enable business users to discover new and exciting things in the data. Pattern 3 combines the data reservoir with a special information discovery system to provide a Graphical User Interface specifically for data discovery. This GUI emulates in many ways how an end user today searches for information on the internet.

To empower a set of business users to truly discover information, they first and foremost require a Discovery tool. A project should therefore always start with that asset.

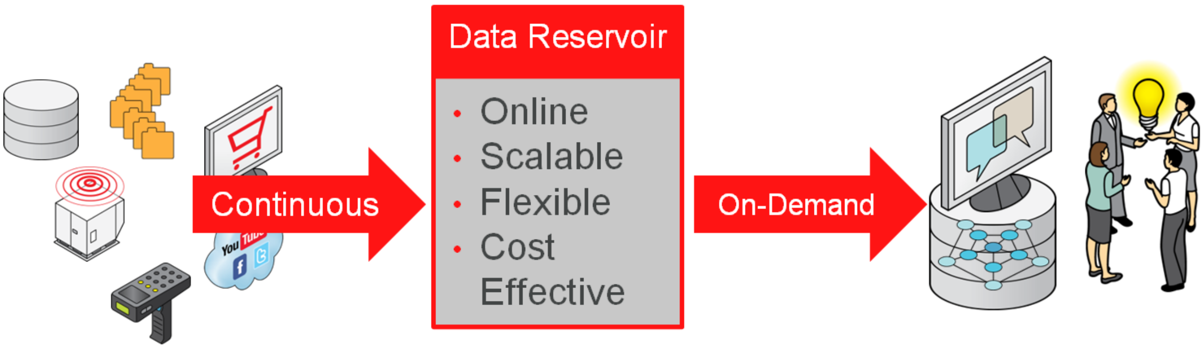

Figure 3 Information Discovery with a Data Reservoir

Once the Discovery tool (like Oracle Endeca) is in place, it pays to start to leverage the Data Reservoir to feed the Discovery tool. As is shown in Figure 3, the Data Reservoir is continuously fed with new data. The Discovery tool is a business user's tool to create ad-hoc data marts in the discovery tool. Having the Data Reservoir simplifies the acquisition by end users because they only need to look in one place for data.

In essence, the Data Reservoir now is used to drive two different systems; the Data Warehouse and the Information Discovery environment and in practice users will very quickly gravitate to the appropriate system. But no matter which system they use, they now have the ability to drive value from data into the organization.

Pattern 4: Closed Loop Recommendation and Analytics System

So far, most of what was discussed was analytics and batch based. But a lot of organizations want to come to some real time interaction model with their end customers (or in the world of the Internet of Things – with other machines and sensors).

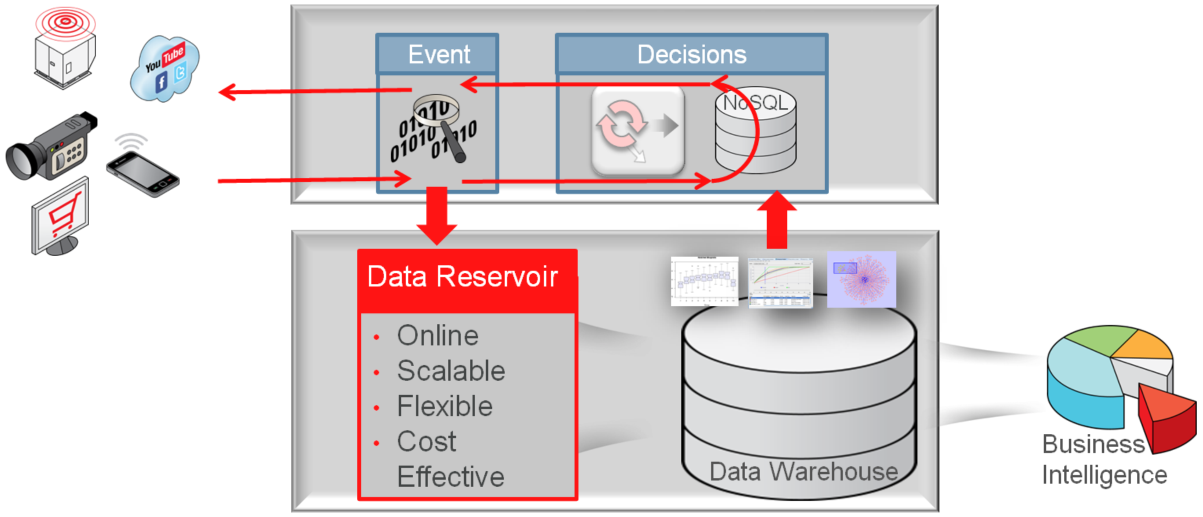

Figure 4 Closed Loop Recomendation and Analytics

Hadoop is very good at providing the Data Factory and the Data Reservoir, at providing a sandbox, at providing massive storage and processing capabilities, but it is less good at doing things in real time.

Therefore, to build a closed loop recommendation system – which should react in real time – Hadoop is only one of the components[1].

Typically the bottom half of Figure 4 is akin to pattern 2 and is used to catch all data, analyze the correlations between recorded events (detected fraud for example) and generate a set of predictive models describing something like "if a, b and c during a transaction – mark as suspect and hand off to an agent". This model would for example block a credit card transaction.

To make such a system work it is important to use the right technology at both levels. Real time technologies like Oracle NoSQL Database, Oracle Real Time Decisions and Oracle Event Processing work on the data stream in flight. Oracle Big Data Appliance, Oracle Exadata/Database and Oracle Advanced Analytics provide the infrastructure to create, refine and expose the models.

Summary

Today's big data technologies offer a wide variety of capabilities. Leveraging these capabilities with the existing environment and skills already in place according to the four patterns described does enable an organization to benefit from big data today. It is a matter of identifying the applicable pattern for your organization and then to start on the implementation. The technology is ready. Are you?

Address:
Jean-Pierre Dijcks
Oracle
500 Oracle Parkway
M/S 4op7
Redwood City, CA, 94065
USA

Phone:	+1 650 607 5394
E-Mail	jean-pierre.dijcks@oracle.com
Internet:	blogs.oracle.com/datawarehousing

[1] In the Hadoop world, a real time system can be built on HBase which does provide real time abilities. However for the pattern described here, the recommendation engine etc. will still need to be built.