INTEGRATING SAP WITH BIG DATA

An introductory guide to integrating SAP with Big Data and how to succeed at it.

GOETZ LESMANN

- Chief Service Officer at Datavard -

A Publication of

DATAVARD

TABLE OF CONTENTS

- Introduction
- SAP and data lakes: Use cases and scenarios
- Opening up SAP for data scientists
- Bringing data from SAP to Hadoop
- Bridging the gap of software logistics
- Consuming Hadoop data from within SAP
- Pitfalls when using SAP data on Hadoop
- Appendix

INTRODUCTION

Everybody is talking about Big Data. Some say it's overly hyped. But the usefulness of Big Data technologies and scenarios is unquestionably enormous, especially for large companies with many customers that typically use SAP to manage finances and logistics. Those are the types of companies that are looking at options for integrating their ERP system and ABAP-based business applications with Hadoop and Big Data scenarios. In this series of chapters I will explain how Datavard views SAP and Big Data. We will show you use cases, technologies, and options that are available to you for making the most of this disruptive technology.

ABOUT THE AUTHOR

Goetz Lessmann is a co-founder of Datavard, an international provider of services and software solutions for data management and system landscape optimization (SLO). He manages the professional services organization of Datavard, which includes Datavard Custom Development, a R&D unit with heavy focus on SAP & Big Data integration. He has more than 20 years of experience in SAP, software development and consulting services.

1

CHAPTER ONE

SAP AND DATA LAKES: USE CASES AND SCENARIOS

SAP AND DATA LAKES: USE CASES AND SCENARIOS

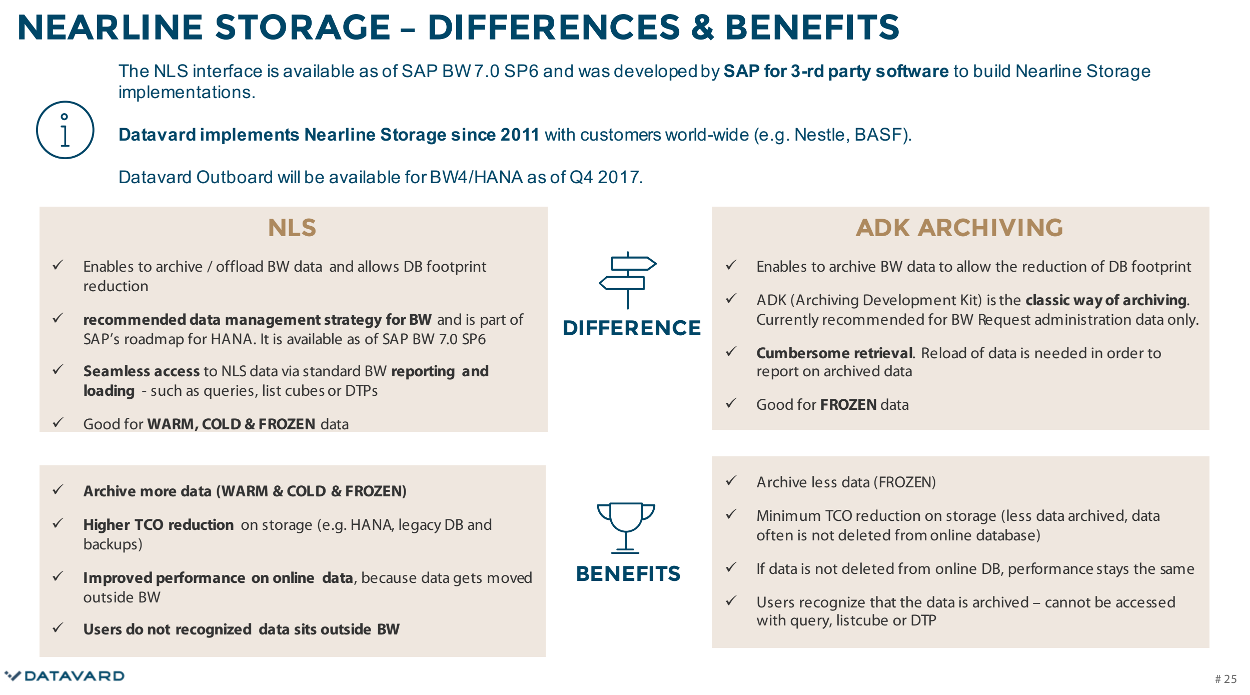
In this chapter, I will illustrate some scenarios and use cases to provide insights and background information on SAP and Big Data. The following chapter further explains the challenges of opening up access to SAP for data scientists, and vice versa opening up Big Data for SAP.

Offloading and footprint reduction

There are two major reasons for the need of such integration which we at Datavard typically hear from our customers:

- The first is offloading. This means keeping the costs of running SAP systems in check by reducing data volume.
- The second reason is true integration between the SAP world and the Big Data world. This involves consuming data: SAP data in Hadoop applications and data lakes and Big Data from SAP.

Offloading (in SAP, "archiving" for ERP or "Near-Line Storage" for BW) is a low hanging fruit, the use of which can help leverage an existing Hadoop cluster or bring a new cluster to produce value rapidly. NLS is often referred to as "archiving for SAP BW," while offering additional features and flexibility. The following figure illustrates the key differences and benefits of NLS when compared to classical ADK archiving.



NEARLINE STORAGE – DIFFERENCES & BENEFITS

The NLS interface is available as of SAP BW 7.0 SP6 and was developed by **SAP for 3-rd party software** to build Nearline Storage implementations.

Datavard implements Nearline Storage since 2011 with customers world-wide (e.g. Nestle, BASF).

Datavard Outboard will be available for BW4/HANA as of Q4 2017.

DIFFERENCE

NLS

- ✓ Enables to archive / offload BW data and allows DB footprint reduction
- ✓ **recommended data management strategy for BW** and is part of SAP's roadmap for HANA. It is available as of SAP BW 7.0 SP6
- ✓ **Seamless access** to NLS data via standard BW **reporting and loading** - such as queries, list cubes or DTPs
- ✓ Good for **WARM, COLD & FROZEN** data

ADK ARCHIVING

- ✓ Enables to archive BW data to allow the reduction of DB footprint
- ✓ ADK (Archiving Development Kit) is the **classic way of archiving**. Currently recommended for BW Request administration data only.
- ✓ **Cumbersome retrieval**. Reload of data is needed in order to report on archived data
- ✓ Good for **FROZEN** data

BENEFITS

NLS:

- ✓ **Archive more data (WARM & COLD & FROZEN)**
- ✓ **Higher TCO reduction** on storage (e.g. HANA, legacy DB and backups)
- ✓ **Improved performance on online data**, because data gets moved outside BW
- ✓ **Users do not recognized data sits outside BW**

ADK archiving:

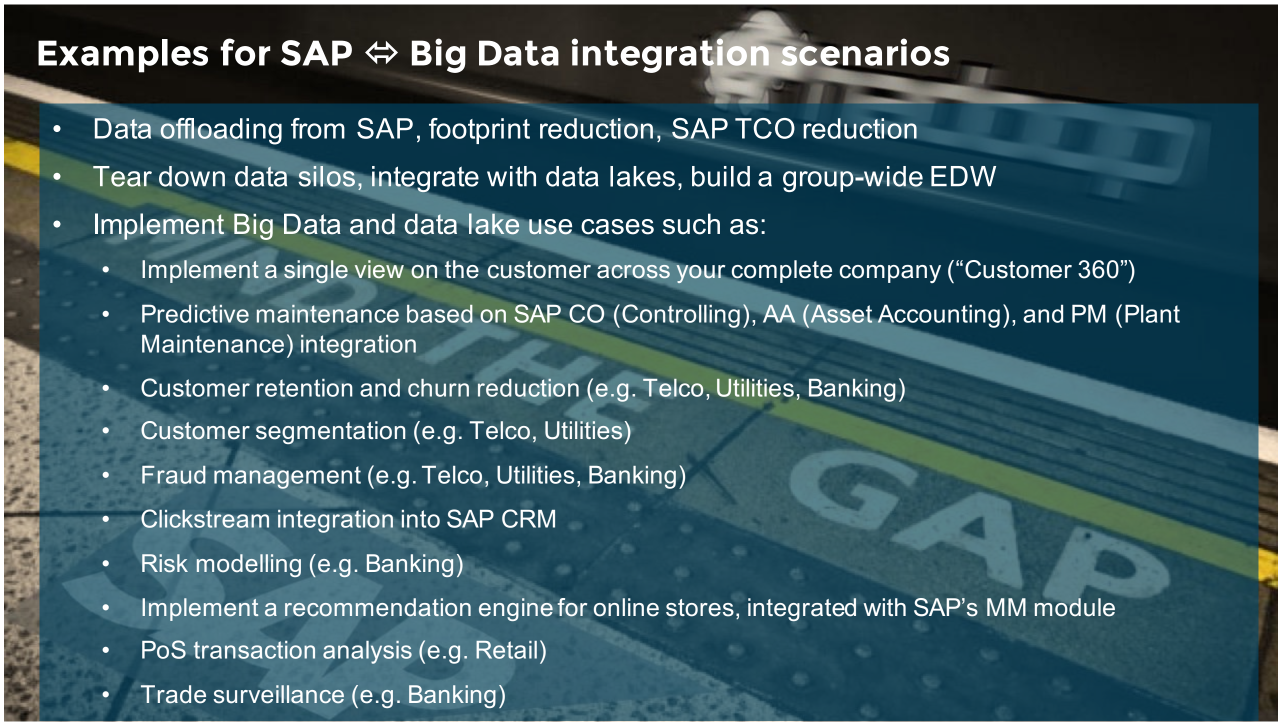
- ✓ Archive less data (FROZEN)
- ✓ Minimum TCO reduction on storage (less data archived, data often is not deleted from online database)
- ✓ If data is not deleted from online DB, performance stays the same
- ✓ Users recognize that the data is archived – cannot be accessed with query, listcube or DTP

DATAVARD

You can offload data from SAP using traditional archiving (e.g. using the SARA transaction in SAP ERP) or using NLS to offload data from SAP BW. Using Datavard OutBoard you can offload NLS data from SAP BW to many storage types such as Hadoop with HIVE, HDFS or Impala, but also classical RDBMS such as Oracle, MSS, plain files or SAP IQ. Datavard Storage Management implements data tiering and aging, and even better: OutBoard is a write-enabled NLS solution.

Integration

After all, Big Data is about unstructured data, streaming, social media, videos of cats reacting to cucumbers, etc. etc. But when you look deeper into the matter you will realize you need to tear down data silos to achieve a true integration.
True integration between SAP and a data lake can help you with various scenarios.



Examples for SAP ⇔ Big Data integration scenarios

- Data offloading from SAP, footprint reduction, SAP TCO reduction
- Tear down data silos, integrate with data lakes, build a group-wide EDW
- Implement Big Data and data lake use cases such as:
  - Implement a single view on the customer across your complete company ("Customer 360")
  - Predictive maintenance based on SAP CO (Controlling), AA (Asset Accounting), and PM (Plant Maintenance) integration
  - Customer retention and churn reduction (e.g. Telco, Utilities, Banking)
  - Customer segmentation (e.g. Telco, Utilities)
  - Fraud management (e.g. Telco, Utilities, Banking)
  - Clickstream integration into SAP CRM
  - Risk modelling (e.g. Banking)
  - Implement a recommendation engine for online stores, integrated with SAP's MM module
  - PoS transaction analysis (e.g. Retail)
  - Trade surveillance (e.g. Banking)

I'll just mention some of them here in detail to explain what are typical scenarios and reasons for integrating SAP with Big Data.

Security and data protection

You can start with analyzing SAP log files and user actions to detect intrusion, hacks, misuse of data and security, and data protection problems. This is a very complex use case, and depends on your type of business, industry, data, etc. But its all about fraud detection, and looking at the log files can help you understand many kinds of security and data issues. You can implement many scenarios, from researching data protection breaches up to proactive monitoring and surveillance.

Customer 360

You can get a 360-degree view of your customer through one dashboard that your sales representatives or pre-sales engineers can access. Such a dashboard can answer all kinds of questions about your customer: Who are you? What did you buy? Where are you? What area is this? What are you interested in? What did other, similar customers buy from us? What can we offer you? and many more. Such questions can be answered by tapping into all data sources in your company, structured and unstructured. Of course, sales data, sales history, conditions, etc. from your SAP ERP system and the contacts from SAP CRM would be very interesting to include in such a dashboard.

Churn reduction

Related to the Customer 360, but more specialized, is Churn Reduction. This will be of interest if you are in telco, utilities, retail or financial services (i.e. industries with many customers, providing B2C services). You can detect which of your customers are prone to switching to the competition based on their data, history, location and overall situation. You can proactively address this with marketing and customer service, be it with special actions, improved customer service, new offers, etc.

Predictive vs. reactive maintenance

You can analyze your assets such as equipment or vehicles to set up a process for predictive maintenance. With predictive maintanence, you switch from reacting to problems to performing maintenance based on the actual use of the equipment. Production orders, old maintenance orders from SAP, as well as sensor data can help build such a use case.

There are many more use cases, but what all of these have in common is that the quality of implemention will be much better by integrating SAP and non-SAP data into a data lake that allows you access to whatever data you need. But there are several challenges when you want to integrate SAP with a data lake.

In the next chapter I will explain the challenge of SAP's complexity combined with its somewhat German-style data model and what this means for data scientists. I will follow up with options for moving data between SAP and Hadoop, and how you can consume and virtualize data (both with and without Datavard's products). The third chapter in the series will be about managing software logistics in such an integrated scenario. The fourth chapter will show you options for consuming Hadoop data from SAP.

2

CHAPTER TWO

OPENING SAP FOR DATA SCIENTISTS

OPENING SAP FOR DATA SCIENTISTS

This is the second chapter about SAP and its integration with Hadoop. The first chapter described scenarios and use cases. Here, I will discuss the challenges for data scientists of accessing SAP and opening up Big Data for SAP. The third chapter will show methods for data integration. The fourth chapter will talk about software logistics. Finally, the fifth chapter will explain how to consume Hadoop data within SAP.

In the previous chapter I was describing the general background of why companies look at an integration between SAP and Hadoop.

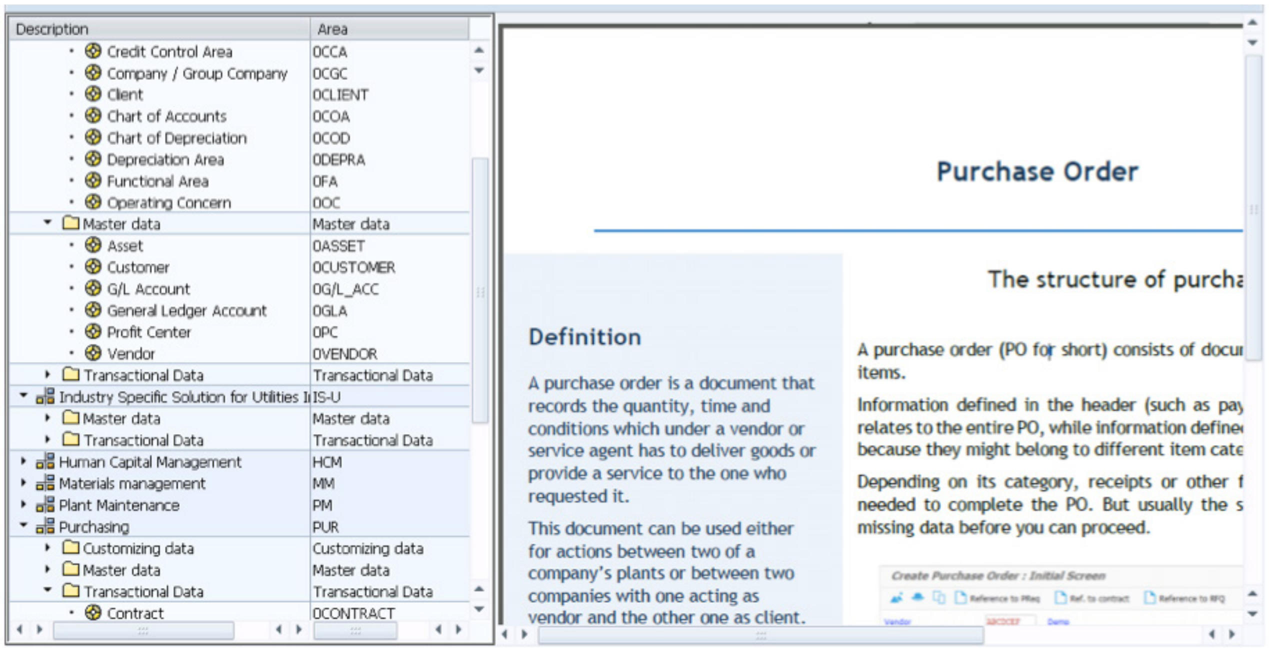
When it comes to business software, there is no way around SAP: 76% of all transactions on the planet go through SAP systems and software in some way, form or shape. That's a truly impressive achievement of the Walldorf, Germany-based software company. However, what is a dream come true for SAP, it is a nightmare for data scientists. SAP software is inherently complex. Well, that's no surprise looking at the incredibly broad coverage of industries and business processes that SAP supports. However, what works well for SAP engineers and experts creates problems for data scientists from the Big Data world who want to access and consume SAP data, e.g. for data lakes.

Big Data scenarios such as customer 360, fraud analysis or security monitoring require flexible, and efficient access to data. SAP offers numerous interfaces for accessing SAP data, including SAP BW with delta extraction, SAP VORA to connect with Hadoop, Data Services as ETL tool, BAPIs, IDOCs, among others.

But data scientists need more than just access to data. Access and interfaces provide the basis for working with the data, but an important ingredient is still missing: a semantic layer which helps making sense of the data. There is a lot of confusion in the market about what SAP VORA actually is: Middleware? A processing engine? A shell around SDA with a powerful engine on Hadoop? A way to integrate SAP HANA with Hadoop? Truth to tell, it is all of that and more. A strong point for HANA, VORA is of course also the strong commitment of SAP's development and technology vision. But what our team at Datavard has learned is that data scientists need even more. This is one of the gaps we are aiming to bridge.

An SAP system viewed from the outside looks like a black box. When you crack it open with the help of some interfaces, views or extraction tools, you end up with about 10,000 tables worth of data, all in the shape and form of SAP data. The first challenge is to identify the source of data. After you have done that, it still is difficult to read and make use of with semi-German field names, cryptic values, etc.
Here is a real life example…

A data scientist is tasked with enriching a data lake with purchase order data. If our data scientist knows SAP, she will (of course!) know that the relevant data is in table EKKO and EKPO, which are German abbreviations for "Einkaufsbelegkopf" and "Einkaufsbelegposition"[1] . When our data scientist looks into these tables and their fields and data, she will of course quickly realize that she needs to consider only EBELNs ("Einkaufsbelegnummer" or purchase order in English), where field BSTYP ("Einkaufsbelegtyp") has the values "F," "K," and "L" (because type "A" would be "Anfrage," which is German for request for quotation). Easy peasy, right?



Description	Area
Credit Control Area	0CCA
Company / Group Company	0CGC
Client	0CLIENT
Chart of Accounts	0COA
Chart of Depreciation	0COD
Depreciation Area	0DEPRA
Functional Area	0FA
Operating Concern	0OC
Master data	Master data
Asset	0ASSET
Customer	0CUSTOMER
G/L Account	0G/L_ACC
General Ledger Account	0GLA
Profit Center	0PC
Vendor	0VENDOR
Transactional Data	Transactional Data
Industry Specific Solution for Utilities I	IS-U
Master data	Master data
Transactional Data	Transactional Data
Human Capital Management	HCM
Materials management	MM
Plant Maintenance	PM
Purchasing	PUR
Customizing data	Customizing data
Master data	Master data
Transactional Data	Transactional Data
Contract	0CONTRACT

Purchase Order

The structure of purcha

Definition

A purchase order is a document that records the quantity, time and conditions which under a vendor or service agent has to deliver goods or provide a service to the one who requested it.

This document can be used either for actions between two of a company's plants or between two companies with one acting as vendor and the other one as client.

A purchase order (PO for short) consists of docu items.

Information defined in the header (such as pay relates to the entire PO, while information define because they might belong to different item cate

Depending on its category, receipts or other f needed to complete the PO. But usually the s missing data before you can proceed.

[1]) Yes, we Germans lack a sense of humor, but compensate with overly long words which sound like the language of the Orks from Lord of the Rings.

To overcome these challenges, we added the Business Process Library (BPL), which includes several hundred business objects, to our Datavard Glue. The BPL serves several purposes:

- Data scientists can identify business objects in SAP that are relevant for their use cases. The BPL includes business objects along with documentation and description, quick help guides and a technical model of the relevant SAP tables.
- The BPL leverages the ETL features of Datavard Glue to extract or even stream SAP data in real time to Hadoop.
- The BPL translates SAP table definitions and structures into English language fields, both in the data model on Hadoop and in field descriptions.
- The BPL translates field values from SAP into meaningful, English language-based values.

With the help of the Datavard Glue Business Process Library, data scientists can easily identify the relevant tables and set up a data flow from SAP to Hadoop in the blink of an eye. What's more, the tables on Hadoop will have English-language field names: PURCHASING_DOC for the purchase order number, DOCUMENT_TYPE for the type of the purchase order, and "PURCHASEORDER" or "CONTRACT" instead of "F" or "A."

3

CHAPTER THREE

BRINGING DATA FROM SAP TO HADOOP

BRINGING DATA FROM SAP TO HADOOP

After previously describing use cases and ways to open access to SAP for data scientists, this chapter will show methods for data integration. In the next chapter we will cover software logistics, before going into ways of how to consume Hadoop data within SAP.

There are many reasons why you may want to bring data from your SAP business applications to Hadoop (e.g. to enrich your data lake with ERP data).

Whatever your reasons, your method for integration will typically be one of the following:

- You can use a classical ETL tool or a combination of various such tools
- You can start coding and build your custom CSV download program which you can then load with Hadoop standard functionality into the Hive metastore
- You will certainly want to have a good look at SAP VORA
- You can use a middleware, such as Datavard's Storage Management which we added to our product Datavard Glue (which works very well with SAP VORA).

Let's look at all the options you have; but keep in mind there will be more, such as the way back. When you consider ETL tools or any integration method, you should not only consider a one-way road from SAP to Hadoop, but also the way back from Hadoop to SAP. This path back can be data replication and ETL (which we do not recommend) or data virtualization (which we do recommend). I will cover this topic later in this ebook.

Using ETL tools

The first idea is straightforward. You can basically use any classic ETL tool to access the SAP database, and gather data from there. Some of these tools may even offer delta mechanisms, and if you use a combination of ETL (let's say data services) and the SLT Replication Server you can (with some effort) implement near real-time streaming of SAP to Hadoop.

Pros:

- You may have an ETL software already in use, and can simply leverage that.
- Solutions for ETL typically offer powerful features for data cleansing and data transformation (the "T" in ETL).
- Some of the ETL features will help you make sense of data by introducing semantics. You may have a central repository or data dictionary.

Cons:

- Often, such implementations are lengthy, tedious and complex from a hardware and software point of view. Typically, ETL solutions require a complex setup, and sit like a spider in the middle of a web.
- It's likely you will need additional hardware on top of your SAP system and Hadoop cluster.
- On a technical level, you need to open up access to the SAP system's database for the ETL tool (e.g. via ODBC or JDBC). At the SAP level, you have no control over which data you allow to be accessed.
- Some SAP tables are stored as "blobs," meaning intransparent tables (at least up to certain SAP release levels). Maybe check table RFBLG or CDCLS if you're curious.
- The overall architecture will be complex and maybe fragile. Testing will be difficult and require frequent re-testing, such as when upgrading one of the components. For example, you may be running the SLT replication server to feed a database with SAP deltas in near real-time, and from there use data services to generate CSV files, which in turn you load into Hive using Hive SQL (admittedly, this is a bit of an over-engineered scenario, but we actually encountered this once).
- You cannot easily use SAP authorizations to control access to SAP data.
- Typically, such a data extraction is a one-way road. For example, though extracting SAP data and providing it for Hadoop may be rather easy, you may find it difficult to implement lookups against Hadoop from SAP ABAP.

Make or buy? Let's make.

Of course you can happily hack away and build your own, custom, solution for extracting data. Using ABAP, you can call up standard SAP interfaces (e.g. BAPIs) and use them to gather data. You can use Open SQL and access SAP tables directly.

Pros:

- Very flexible solution.
- Can run directly on your SAP application server.
- Tight integration into SAP's software logistics (Transport Management System).
- You can use SAP authorizations to restrict and control access to SAP data.
- Let's not forget: coding's fun, right? That's assuming somebody else will maintain your code in the years to come.

Cons:

- You're re-inventing the wheel (and it probably will not be perfectly round and wheel shaped).

- You will need to maintain this solution going forward. Sounds easy, but do you know what changes will come with ABAP, S/4HANA and future SAP developments? It may be more complicated than you think.
- Scalability may be an issue. For small tables like customer master data, this may be a feasible approach, but for volume data this will be tricky.
- Capturing deltas will not be straightforward.

What about SAP VORA?

When it comes to the integration between HANA and the Big Data world, SAP is making a strong push for SAP VORA. And indeed, VORA is gaining traction and definitely is a rising star. It is based on the powerful Spark technology, which SAP enhanced and extended with in-memory features (among others).

However, Vora provides technology access only and does not help you with accessing data in a semantic way. Well, it actually doesn't need to, because it was never intended for that purpose. VORA is a powerful processing engine, and offers access to SAP data from and to the Netweaver stack through JDBC and SDA. You can make use of this data with a heavily improved and extended Spark engine with powerful SQL features.

As far as data processing goes, these features of SAP VORA will help you a lot. Regarding accessing and consuming data from SAP business applications and the ABAP world in general, you may want to use it in tandem with a solution such as Datavard Glue. With Glue you can leverage the ABAP dictionary, the Glue BPL and look at data access and processing from ABAP level. This is because SAP VORA will access the ABAP world only through ODBC and JDBC, not through the SAP business application itself.

Pros:

- It's an SAP in-house solution, so installation and integration can be expected to be fairly easy. You can expect fewer problems with integration, regression testing or change management when upgrading a component.
- SAP obviously makes a huge commitment to VORA and keeps improving and extending the solution, assuring future stability and safety.
- Access goes both ways. You can access SAP data from VORA (using SDA) and process VORA data in HANA.

Cons:

- You open up the complete SAP system to the Big Data world without full control or visibility on which data is being accessed since access is at table level. The ABAP applications don't know which data is being accessed and have no way to prevent this (such as using SAP roles and authorizations).

- Accessing data using SDA can potentially be heavy on resources such as network, CPU and memory. That's why we recommend replication of data rather than data federations to access SAP business data from the Hadoop and Big Data world.
- There are some release dependencies between SAP VORA and Hadoop components. These are becoming less and less with SAP continuously improving the product. If you already have a Hadoop cluster, you should nevertheless check the compatibility of your cluster with SAP VORA, unless you want to ramp up a second VORA-only cluster.
- VORA does not help you with identifying relevant SAP business data for your use case, because it is not a "semantic" middleware, but rather a powerful processing engine.

These drawbacks don't mean you shouldn't use SAP VORA. But you may want to use it as a processing engine for data science and analytics rather than as a way to set up, model and control ETL data flows and data access.

So, how would you do all of the above quickly and easily?
One solution for connecting SAP systems with various storage types and technologies (such as Cloudera's Impala, or SAP VORA) is our Datavard Glue. Using Glue, you can easily create a data model on your Big Data platform (e.g. HIVE), and populate it with SAP data by creating extractors and data flows. You have full control over which data is accessed because Glue is built entirely in ABAP and leverages SAP authorizations. You have logs, protocols and full visibility as to which data you replicate to Hadoop. Glue extractors offer various powerful ways of delta extraction, up to a near real-time feed from SAP to Hadoop. You can access data from Hadoop from SAP and, depending on your SQL engine on Hadoop, even embed such SQL code in user exits in your business processes to perform lookups against Hadoop. Glue makes development of any Hadoop or Big Data integration easy by leveraging the SAP Transport Management System for software logistics.

Pros:
- Datavard Glue is a very lean solution, implemented in ABAP. You don't need additional hardware and the installation only requires importing a transport request (after which you have to configure the connections to your cluster of course, which will be more work than the installation itself).
- Glue leverages SAP's proven technology: Transport Management system, authorizations, etc.
- Glue comes with the Business Process Library to open the SAP data model and translate SAP fields and values into "speaking" values. In the chapter "Open up access to SAP for data scientists" I describe this in more detail.

- Glue offers ETL features such as full extraction, delta extraction and near real-time data feeds. It is also a two-way path where you can access SAP data from Hadoop, and Hadoop data from SAP.

Cons:

- Glue currently supports only your SAP system and CSV files which you may have from other sources. Support for other systems and databases-native HANA databases, SAP IQ, SAP ASE—is in the works for release toward the end of 2017.

Other than that, I'm having a hard time finding disadvantages of this solution, except maybe that we're overly proud of it.

How to Glue

To set up data replication from SAP ERP to Hadoop, for example, here's what you do:

1. Log in to your development system and access the Datavard Glue Cockpit and Glue Workbench. Then create a table on Hadoop - which is as simple as creating a table in the ABAP workbench.
2. Create an extractor, where you can specify the delta type for data collection, you define data filters, create field mappings, define transformations, etc.
3. Set up a variant for your extractor where you basically customize your extractor, for example with the values of your data filter.
4. Unit test the extraction by scheduling a job to execute the extractor with your variant.
5. Release your transport request with the objects you created and transport to your QA system for full testing
6. Finally, transport to SAP production and schedule your extraction job.

4

CHAPTER FOUR

BRIDGING THE GAP OF SOFTWARE LOGISTICS

BRIDGING THE GAP OF SOFTWARE LOGISTICS

This is the fourth chapter about SAP and its integration with Hadoop. The first chapter described scenarios and use cases. The second chapter explaind the challenges of opening up access to SAP for data scientists and opening up Big Data for SAP, followed by a chapter on technical possibilities for data integration. This chapter is about software logistics. The fifth and final chapter explains how to consume Hadoop data within SAP.

When we set out to build our solution Datavard Glue some years ago, our motivation was both simple and lazy. The truth is, I was simply not in the mood to figure out how to build a data model on Hadoop.

Let's face it: 20 years of working with SAP have brutally spoiled me. SAP comes with great features. The Data Dictionary. The ABAP workbench. Transports and the TMS. Security and authorizations. An easy way of scheduling jobs, not to mention process chains! Working with SAP is easy, once you've mastered the initial hurdles and swallowed the bitter pill of working with SAP GUI.

This laziness from back then turns out to be really helpful nowadays for our customers. Embedding the integration between SAP and Hadoop in a really tight way has many advantages. For example, the installation is extremely easy and you don't even need additional hardware. Also, as an SAP professional, you can immediately start to work with creating and populating a data model to consume SAP data in your data lake, even if you're not skilled with Hadoop and Big Data tools.

The tight integration means that you can use standard SAP features and tools to build and test your SAP-to-Hadoop integration in the same way you develop your SAP solutions. You build on a development system, test on a test system, and then put it into operation on your production system. For those used to working with SAP, that sounds perfectly natural. But Big Data experts and data scientists aren't necessarily keen to develop applications or do data research against a test cluster with no data.

In the Big Data world you basically develop on or deploy directly to your production cluster. If you're the data protection officer of a company, you may get worry wrinkles when you consider this from a compliance perspective (Sarbanes-Oxley or Europe's General Data Protection Regulation).

Datavard Glue offers the following features of ETL tools, combined with APIs, for our powerful middleware storage management:

- Central data dictionary to define your SAP data lake integration
- Extraction from SAP to Hadoop, including delta extraction and near real-time data feeds
- ETL features to help you implement data transformations during data replication from SAP to Hadoop
- Data federation features, such as Virtual InfoProviders on SAP BW
- Business Process Library to enable access to SAP for data scientists by introducing a semantic layer
- Leverage the SAP authorization model to control data flows, and even utilize SAP authorization data on Hadoop
- Consume data from Hadoop in your SAP applications using data flows, or even in real-time data access through our ABAP API
- Rapid development of Hadoop-based data models
- Use the SAP Transport Management system
- Orchestrate your data flows from SAP and have full control over who can access what SAP data

Datavard Glue comes with many ETL features. We recommend replicating SAP data to Hadoop to consume it in your data lake or Big Data applications. Replication may sound like you are creating redundancies, storage problems, extraction troubles and long runtimes. But one of the demos that I like to show is how to replicate 300,000 sales documents from ERP to Hadoop in less than five minutes in a full load to Hadoop (with deltas this would obviously be even much faster).

For the way back - consuming Big Data from your SAP applications - we recommend data federations and data virtualization. This way, you are in control over which data you expose to Hadoop and avoid performance impacts on your production SAP systems. From a Big Data perspective, SAP data is low volume, so the replication won’t be a problem from a volume or footprint perspective. For the way back - consuming SAP data from Hadoop - we recommend to not persist the data on SAP so as to not increase the footprint of your SAP landscape.

By the way, a lot of thought went into the naming of our solution. We could have called it “Datavard Bridge” because we want to bridge between the two worlds. But the integration is so tight, that we chose something that would indicate that. We indeed glue the two worlds together.

5

CHAPTER FIVE

CONSUMING HADOOP DATA FROM WITHIN SAP

CONSUMING HADOOP DATA FROM WITHIN SAP

This is the fifth chapter about SAP and its integration with Hadoop. The first chapter described scenarios and use cases. The second chapter explained the challenges of opening up access to SAP for data scientists, and opening up access to Big Data for SAP. This was followed by an chapter on methods for data integration and a chapter about software logistics. Finally, this chapter explains the many options for consuming Hadoop data from within SAP ABAP applications.

With classical ETL tools such as VORA, SDA and JDBC, you can connect SAP with Hadoop. With Datavard Glue you can achieve a very tight integration between SAP and your data lake. In the previous chapters in this ebook I was highlighting many features such as delta extraction, near real-time data feeds and the Business Process Library. But you still will need to access data from your Hadoop cluster or other Big Data platform from SAP.

In this chapter, I will explain the following options to do this:

- ETL tool to load and persist data in SAP
- Glue extractors to replicate data and persist it in SAP
- SDA to access Hadoop data through HANA views
- BW Virtual InfoProviders
- Glue Data Browser
- Glue ABAP API

ETL tool to load and persist data in SAP

You can use any ETL tool which can connect to both Hadoop (HIVE, Impala, Vora, etc.) and SAP. An example for such a tool would be SAP's own BODS. However, you need to be aware that the extraction (while maybe offering powerful transformation features) would persist data on the SAP side. For any true Big Data integration, persisting data on SAP seems plain wrong, because it would increase the footprint of the SAP system unnecessarily. So this would make sense only for small data sets. Also, any kind of real-time access would not be possible.

Glue extractors to replicate data and persist it in SAP

Datavard Glue is a solution to closely integrate the SAP world with Big Data platforms, such as Hive, Impala or SAP VORA. Glue, with its ETL and data modeling features, introduces a semantic layer on top of SAP to allow non-SAP skilled people like Java developers or data scientists to access and make sense of the meaning of SAP data. access and make sense of the meaning of SAP data.

Full disclosure: I do not recommend this option. Technically, this is feasible, and you can set up data extractors and data flows just a you do when replicating data from SAP to Hadoop. However, we recommend to not persist Big Data on SAP. The reason is simple: to not increase the footprint of your SAP system unnecessarily. There may be exceptions to this, but simply put: storing data on the SAP side (especially on HANA) may be a costly exercise. We recommend caution if you decide to go with this option.

SDA to access Hadoop data through HANA views

SDA is a technology by SAP which allows connections to HANA with other databases and builds data federations. Using SDA, you can create virtual tables on HANA which pull data from other sources via JDBC/ODBC. SDI allows for real-time replication of data.

You can create SDA views on your SAP system on HANA that will read data from VORA. This is entirely feasibly; but you need to watch out for an impact on performance impact. Especially if you don't know how your Big Data integration will evolve over time. SDA can put quite some load on your system, especially in terms of memory, when database tables are merged.

Also, you need to be aware that you will need to work both in the HANA studio and in the ABAP workbench if you want to integrate with ABAP-based applications. In a nutshell, this method of integration will work, but it may be a bit more difficult to set up and maintain than you would expect.

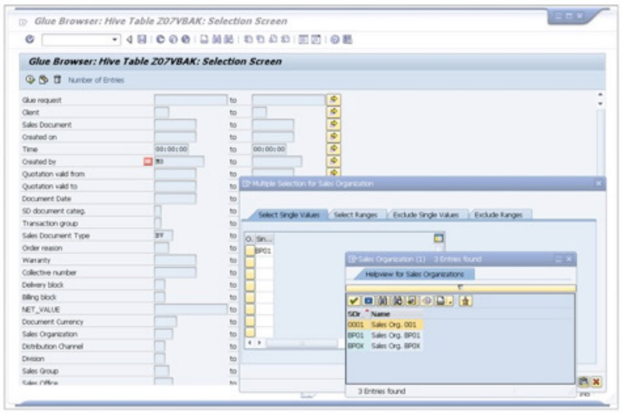
When considering processing engines on Hadoop, bear in mind SDA comes with some restrictions. While SAP VORA is of course fully supported, there are limitations with other engines such as Impala. If you want to go with SDA, I recommend checking closely before any implementation or POC as to whether your Hadoop version and the components you're running on Hadoop are supported. With SDI, you will also need to install additional components on the Hadoop cluster to enable the use of SDI.

BW Virtual InfoProviders with Glue

This is a very powerful way of creating a tight integration, but requires you to have the SAP Business Warehouse component in active use in your system. If you do have this running, you can generate Virtual InfoProvides using Datavard Glue which will read data from Hadoop dynamically. You can embed these virtual providers in multi-providers, in your data flows, or use them directly in BW queries. When such a query is executed, the system will look into your Big Data platform through the virtual provider and Glue's access to Datavard Storage Management.

Glue Data Browser

The Glue Data Browser is a powerful tool which helps you when developing your data model and when you populate your data lake. The figure below is a sample from Hive of the SAP data model and SD document headers from table VBAK. The screen shot on the left hand illustrates the power of SAP select-options; the screen shot on the right hand shows the result of the selection.



Glue Browser: Hive Table Z07VBAK: Selection Screen

Number of Entries

Glue request		to	
Client		to	
Sales Document		to	
Created on		to	
Time	00:00:00	to	00:00:00
Created by	[illegible]	to	
Quotation valid from		to	
Quotation valid to		to	
Document Date		to	
SD document categ.		to	
Transaction group		to	
Sales Document Type	BV	to	
Order reason		to	
Warranty		to	
Collective number		to	
Delivery block		to	
Billing block		to	
NET_VALUE		to	
Document Currency		to	
Sales Organization		to	
Distribution Channel		to	
Division		to	
Sales Group		to	
Sales Office		to	

Multiple Selection for Sales Organization

Select Single Values | Select Ranges | Exclude Single Values | Exclude Ranges

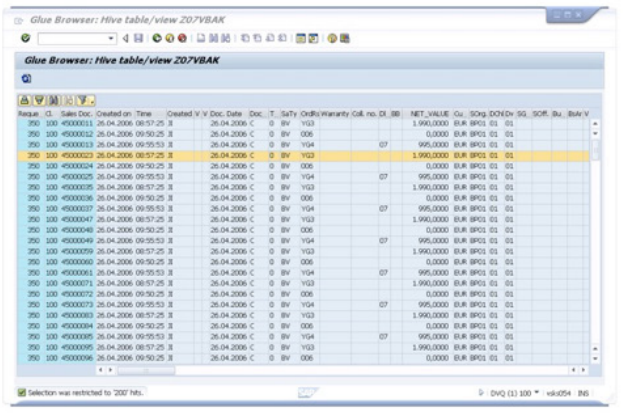
O.	Sin...
	BP01

Sales Organization (1) 3 Entries found

Helpview for Sales Organizations

SOrr	Name
0001	Sales Org. 001
BP01	Sales Org. BP01
BPOX	Sales Org. BPOX

3 Entries found



Glue Browser: Hive table/view Z07VBAK

Reque	Cl.	Sales Doc.	Created on	Time	Created	V	V	Doc. Date	Doc.	T.	SaTy	OrdRs	Warranty	Coll. no.	DI	BB	NET_VALUE	Cu.	SOrg.	DChl	Dv	SG	SOff.	Bu.	BsAr	V
350	100	45000011	26.04.2006	08:57:25	JI			26.04.2006	C	0	BV	YG3					1.990,0000	EUR	BP01	01	01					
350	100	45000012	26.04.2006	09:50:25	JI			26.04.2006	C	0	BV	006					0,0000	EUR	BP01	01	01					
350	100	45000013	26.04.2006	09:55:53	JI			26.04.2006	C	0	BV	YG4			07		995,0000	EUR	BP01	01	01					
350	100	45000023	26.04.2006	08:57:25	JI			26.04.2006	C	0	BV	YG3					1.990,0000	EUR	BP01	01	01					
350	100	45000024	26.04.2006	09:50:25	JI			26.04.2006	C	0	BV	006					0,0000	EUR	BP01	01	01					
350	100	45000025	26.04.2006	09:55:53	JI			26.04.2006	C	0	BV	YG4			07		995,0000	EUR	BP01	01	01					
350	100	45000035	26.04.2006	08:57:25	JI			26.04.2006	C	0	BV	YG3					1.990,0000	EUR	BP01	01	01					
350	100	45000036	26.04.2006	09:50:25	JI			26.04.2006	C	0	BV	006					0,0000	EUR	BP01	01	01					
350	100	45000037	26.04.2006	09:55:53	JI			26.04.2006	C	0	BV	YG4			07		995,0000	EUR	BP01	01	01					
350	100	45000047	26.04.2006	08:57:25	JI			26.04.2006	C	0	BV	YG3					1.990,0000	EUR	BP01	01	01					
350	100	45000048	26.04.2006	09:50:25	JI			26.04.2006	C	0	BV	006					0,0000	EUR	BP01	01	01					
350	100	45000049	26.04.2006	09:55:53	JI			26.04.2006	C	0	BV	YG4			07		995,0000	EUR	BP01	01	01					
350	100	45000059	26.04.2006	08:57:25	JI			26.04.2006	C	0	BV	YG3					1.990,0000	EUR	BP01	01	01					
350	100	45000060	26.04.2006	09:50:25	JI			26.04.2006	C	0	BV	006					0,0000	EUR	BP01	01	01					
350	100	45000061	26.04.2006	09:55:53	JI			26.04.2006	C	0	BV	YG4			07		995,0000	EUR	BP01	01	01					
350	100	45000071	26.04.2006	08:57:25	JI			26.04.2006	C	0	BV	YG3					1.990,0000	EUR	BP01	01	01					
350	100	45000072	26.04.2006	09:50:25	JI			26.04.2006	C	0	BV	006					0,0000	EUR	BP01	01	01					
350	100	45000073	26.04.2006	09:55:53	JI			26.04.2006	C	0	BV	YG4			07		995,0000	EUR	BP01	01	01					
350	100	45000083	26.04.2006	08:57:25	JI			26.04.2006	C	0	BV	YG3					1.990,0000	EUR	BP01	01	01					
350	100	45000084	26.04.2006	09:50:25	JI			26.04.2006	C	0	BV	006					0,0000	EUR	BP01	01	01					
350	100	45000085	26.04.2006	09:55:53	JI			26.04.2006	C	0	BV	YG4			07		995,0000	EUR	BP01	01	01					
350	100	45000095	26.04.2006	08:57:25	JI			26.04.2006	C	0	BV	YG3					1.990,0000	EUR	BP01	01	01					
350	100	45000096	26.04.2006	09:50:25	JI			26.04.2006	C	0	BV	006					0,0000	EUR	BP01	01	01					

Selection was restricted to '200' hits.

DVQ (1) 100 | vsks054 | INS

Sample from Hive of the SAP data model and SD document headers from table VBAK.

The Data Browser may be more of a development tool, but you can use this in a very flexible way and even leverage the same technology in your business processes. More on that coming up. The Glue Data Browser is a SAP SE16-style program that lets you look into tables stored on VORA, Impala or good old HIVE. With some of these engines, such as Cloudera's Impala, you will usually not encounter a significant difference with your online database.

Glue ABAP API

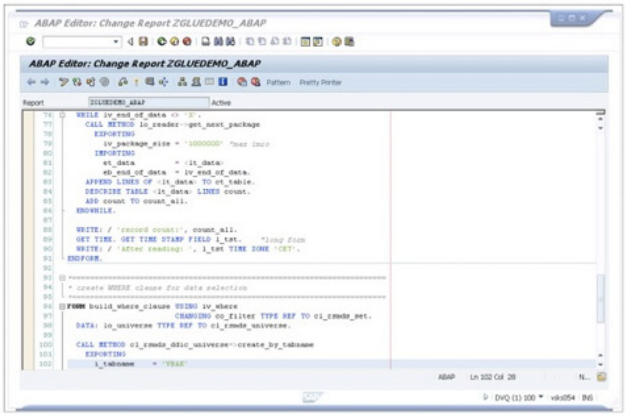
The ABAP API in Glue is a very powerful integration feature. You can use a set of ABAP objects classes and methods in your own ABAP code to access data in your Big Data platform. Depending on the execution engine, you can embed lookups as SQL "select" statements in user exits to pull data from Hadoop that is embedded in user exits into your SAP ERP system. You can easily perform such queries against your data lake using SAP VORA or Impala and retrieve results in dialog.

A simple sample ABAP program which I use in Glue demos shows how to perform such lookups in 120 lines of code—which actually include a lot of helpful comments. The comments in this little ABAP demo help make the code understandable and readable.

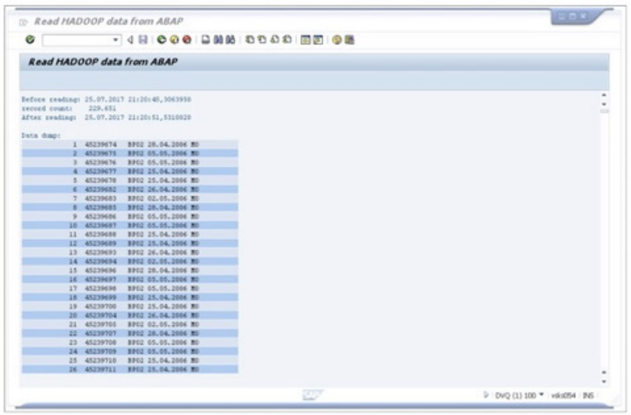
```
* ABAP demo for Glue data access:  Access data in Hadoop and work
* with this data from ABAP
REPORT zgluedemo_abap NO STANDARD PAGE HEADING.

DATA:  gt_data      TYPE TABLE OF vbak, "// Global data
       gs_workarea LIKE vbak,
       gv_zebra.

PERFORM select_data_vbak USING 'ZGLUEVBAK' "// Read Data
                               'ERNAM EQ ''MO'''
                         CHANGING gt_data.

SKIP. WRITE: / 'Data dump:'. "// Display data
LOOP AT gt_data INTO gs_workarea.
  IF gv_zebra EQ space. " simple zebra toggle for standard list
    FORMAT COLOR 2. gv_zebra = 'X'.
  ELSE.
    FORMAT COLOR 4. gv_zebra = space.
  ENDIF.
  WRITE: /5 sy-tabix, gs_workarea-vbeln, gs_workarea-vkorg,
            gs_workarea-erdat, gs_workarea-ernam.
ENDLOOP.
FORMAT COLOR OFF. WRITE: / 'Done'.

*======================================================================
* Read data from HIVE table ZGLVBAK
*======================================================================
FORM select_data_vbak
     USING iv_tab iv_where
     CHANGING ct_table TYPE STANDARD TABLE.
  DATA:   ls_field_sel   TYPE /dvd/sm_s_field_sel,
          lt_field_sel   TYPE /dvd/sm_t_field_sel,
          lo_reader      TYPE REF TO /dvd/sm_if_tab_reader,
          lo_filter      TYPE REF TO cl_rsmds_set,
          lo_data        TYPE REF TO data,
          lv_end_of_data TYPE xfeld,
          l_tst          TYPE timestampl.
  FIELD-SYMBOLS: <lt_data> TYPE ANY TABLE.

*--- dynamic creation of internal table for select
  CREATE DATA lo_data TYPE TABLE OF vbak.
  ASSIGN lo_data->* TO <lt_data>.

*--- select fields and where-condition for selection
  ls_field_sel-tabname    = iv_tab.
  DEFINE afield.
    ls_field_sel-fieldname = '&1'.
    APPEND ls_field_sel TO lt_field_sel.
  END-OF-DEFINITION.

*--- specify which fields we want to receive back from Hadoop
  afield vbeln. afield erdat. afield ernam. afield vbtyp. afield vkorg.
  PERFORM build_where_clause USING iv_where CHANGING lo_filter.

*--- prepare reading
  CALL METHOD /dvd/sm_cl_tab_storman=>get_reader
    EXPORTING
      iv storid    = 'H_CDC_IMPQ'
      iv_tabname   = ls_field_sel-tabname
      it_field     = lt_field_sel
    RECEIVING
      rref_reader = lo_reader.

*--- open cursor
  CALL METHOD lo_reader->open_cursor
    EXPORTING
      iv_tabname   = iv_tab
      it_field     = lt_field_sel
      iref_filter = lo_filter.

  GET TIME. GET TIME STAMP FIELD l_tst.    "long form
  WRITE: / 'Before reading:', l_tst TIME ZONE 'CET'.

*--- loop data packages until out of data
  DATA:  count_all TYPE i, count TYPE i.
  WHILE lv_end_of_data <> 'X'.
    CALL METHOD lo_reader->get_next_package
      EXPORTING
        iv_package_size = '1000000' "max 1mio
      IMPORTING
        et_data          = <lt_data>
        eb_end_of_data  = lv_end_of_data.
    APPEND LINES OF <lt_data> TO ct_table.
    DESCRIBE TABLE <lt_data> LINES count.
    ADD count TO count_all.
  ENDWHILE.

  WRITE: / 'record count:', count_all.
  GET TIME. GET TIME STAMP FIELD l_tst.    "long form
  WRITE: / 'After reading: ', l_tst TIME ZONE 'CET'.
ENDFORM.

*======================================================================
* create WHERE clause for data selection
*======================================================================
FORM build_where_clause USING iv_where
                        CHANGING co_filter TYPE REF TO cl_rsmds_set.
  DATA: lo_universe TYPE REF TO cl_rsmds_universe.

  CALL METHOD cl_rsmds_ddic_universe=>create_by_tabname
    EXPORTING
      i_tabname     = 'VBAK'
    RECEIVING
      r_r_universe = lo_universe
    EXCEPTIONS
      not_found     = 1
      OTHERS        = 2.

  CALL METHOD lo_universe->create_set_from_string
    EXPORTING
      i_string - iv_where " e.g. `KUNNR > '0000000300'`
    RECEIVING
      r_r_set  = co_filter.
ENDFORM.
```

Sample ABAP demo program.

The following Figure illustrates the use of this code. The screen shot on the left shows the program in the ABAP editor; the screenshot on the right is the result of the program on the ABAP standard list, i.e. the "stdout" corresponding feature of ABAP.


ABAP Editor: Change Report ZGLUEDEMO_ABAP
ABAP Editor: Change Report ZGLUEDEMO_ABAP
Pattern
Pretty Printer
Report
ZGLUEDEMO_ABAP
Active
ABAP
Ln 102 Col 28
DVQ (1) 100
INS

```
  WHILE lv_end_of_data <> 'X'.
    CALL METHOD lo_reader->get_next_package
      EXPORTING
        iv_package_size = '1000000' "max 1mio
      IMPORTING
        et_data          = <lt_data>
        eb_end_of_data  = lv_end_of_data.
    APPEND LINES OF <lt_data> TO ct_table.
    DESCRIBE TABLE <lt_data> LINES count.
    ADD count TO count_all.
  ENDWHILE.

  WRITE: / 'record count:', count_all.
  GET TIME. GET TIME STAMP FIELD l_tst.    "long form
  WRITE: / 'After reading: ', l_tst TIME ZONE 'CET'.
ENDFORM.

*======================================================================
* create WHERE clause for data selection
*======================================================================
FORM build_where_clause USING iv_where
                        CHANGING co_filter TYPE REF TO cl_rsmds_set.
  DATA: lo_universe TYPE REF TO cl_rsmds_universe.

  CALL METHOD cl_rsmds_ddic_universe=>create_by_tabname
    EXPORTING
      i_tabname     = 'VBAK'
```



```
Before reading: 25.07.2017 21:20:45,3063950
record count:     229.651
After reading:  25.07.2017 21:20:51,5310020

Data dump:
          1  45239674  BP02 28.04.2006 MO
          2  45239675  BP02 05.05.2006 MO
          3  45239676  BP02 05.05.2006 MO
          4  45239677  BP02 25.04.2006 MO
          5  45239678  BP02 25.04.2006 MO
          6  45239682  BP02 26.04.2006 MO
          7  45239683  BP02 02.05.2006 MO
          8  45239685  BP02 28.04.2006 MO
          9  45239686  BP02 05.05.2006 MO
         10  45239687  BP02 05.05.2006 MO
         11  45239688  BP02 25.04.2006 MO
         12  45239689  BP02 25.04.2006 MO
         13  45239693  BP02 26.04.2006 MO
         14  45239694  BP02 02.05.2006 MO
         15  45239696  BP02 28.04.2006 MO
         16  45239697  BP02 05.05.2006 MO
         17  45239698  BP02 05.05.2006 MO
         18  45239699  BP02 25.04.2006 MO
         19  45239700  BP02 25.04.2006 MO
         20  45239704  BP02 26.04.2006 MO
         21  45239705  BP02 02.05.2006 MO
         22  45239707  BP02 28.04.2006 MO
         23  45239708  BP02 05.05.2006 MO
         24  45239709  BP02 05.05.2006 MO
         25  45239710  BP02 25.04.2006 MO
         26  45239711  BP02 25.04.2006 MO
```

This illustrates the use of ABAP code.

I also like using this ABAP program in demos to show the speed of access which Hadoop currently offers. While a query against Hive may take quite a while, it will finish in mere seconds or milliseconds on Impala.

6

CHAPTER SIX

PITFALLS WHEN USING SAP DATA ON HADOOP

PITFALLS WHEN USING SAP DATA ON HADOOP

This chapter focuses on different challenges in using SAP data on Hadoop. I will discuss some tricky aspects of data from SAP's ERP solution and give some pointers as to potential solutions and workarounds. I will also discuss how our Datavard solution, Datavard Glue, deals with these challenges.

One may think that using SAP data on Hadoop is fairly easy. However, things are not as they seem because there are challenges. You should carefully look at each before making any architecture decisions.
The most important challenges are the following, which I've sorted sequentially based on the points at which you will encounter them during your implementation project:

1. Extracting data from SAP
2. Enriching data from SAP
3. Making sense of cryptic SAP data
4. Delta mechanisms and extracting deltas
5. Streaming data from SAP to Hadoop
6. Frequent updates to SAP data
7. Consuming the results of Hadoop processing from within SAP

Extracting data from SAP

Data extraction (ETL) from SAP can be solved through many tools and approaches. For some tables, like master data, it will be fairly easy to extract the data 1:1 from SAP to Hadoop. Some other SAP tables may be more difficult to extract. Just to give you some examples, this may be due to data volumes (SD document line items) or how SAP stores the data, such as FI document line items from table BSEG, which do not exist in the database but reside in cluster table RFBLG.
Chapter 3 of this e-book covers different options for getting SAP data to Hadoop.

My recommendation is to go for a lean approach, which allows for flexibility while at the same time minimizing manual work. With our Datavard Glue we implemented a 100% ABAP-based ETL solution that does not require additional hardware or a complex setup, instead running directly embedded in a Netweaver ABAP stack.

Enriching data from SAP

Enriching data can mean filling blank fields, adding data, joining data of different sources or normalizing data, among others. As this is basically a standard step within data flows or data extraction, it can be covered by ETL tools. However, I want to mention this challenge as a separate call out here. The reason is that quite often you will find that it makes a lot of sense to integrate this step into ABAP programs. This is because at that point you will have all data ready at your fingertips. It is quite easy to determine lookups and enrich data.

Here are some simple examples:

- Looking up values of fields, texts, or descriptions is rather easy. For example, you might want to rename a document type with a more speaking value.
- It is quite simple to add recent revenue with a particular customer to data extracted from SAP to Hadoop by performing lookups into the LIS in SAP's ERP during extraction (table S001).
- You may want to translate SAP user names into domain names or real names during the extraction. You could replace the SAP user name of the person who performed a change of data or who posted a document with their real names.

Making sense of cryptic SAP data

SAP data has a tendency to be cryptic, making little sense to data scientists (and that is after identifying which data is relevant for a certain Big Data scenario). In typical scenarios, you will spend quite a bit of time and effort—not to mention frustration and money—performing such operations. You may grumble, but that is a necessary part of the data cleansing.

At Datavard, we tackle this challenge through our Business Process Library, which

- allows data scientists to identify relevant data in SAP by introducing a semantic layer of business objects into the data extraction and even providing "cheat sheets" of important facts and details.
- includes a key for SAP fields to use names on Hadoop.
- includes a key for making field values more indicative, for example, by renaming document type values such as "AA" or "02" to reflect their use such as "asset accounting" or "purchase order."

You can find more information on this topic in the chapter "SAP & Big Data Challenge Opening SAP for data scientists."

Delta mechanisms and extracting deltas

When extracting data from SAP to Big Data platforms, you need to perform an initial data replication. After the completion of this initial load, you will want to consider new and updated data only. However, capturing deltas from SAP is more difficult than you may think. SAP's very own extraction from ERP to BW, for example, solves this through numerous SAP-coded extractors which handle delta queues.

I get questions from our customers rather frequently whether it would be possible to simply call the standard SAP BW extractors, which are part of SAP's ERP code base, and thus generate a delta to feed any kind of Hadoop database. This may sound elegant, but it's risky. SAP did not release the function modules and classes which are the basis of the BW extraction for custom use by customers. As a consequence, such a use of the SAP code, while technically possible, may cause unwanted side effects and problems in support. For that reason, we cannot advise such a solution. Even if it may be feasible today, it could be on shaky ground in the future. SAP may change some of the logic, the interfaces or the code itself, and you would need to perform constant regression testing and make plenty of adjustments.

With our Datavard Glue, we solved this so as to avoid any such problems. We implemented delta mechanisms based on the data itself in the following ways:

1. We can use date, time and timestamp fields to collect new data.
2. We can capture new data based on number ranges.
3. We implement database triggers to catch any kind of update, insertion or deletion of data.

Depending on the type of data, the SAP application and the nature of the data, you can cherry pick from these different extraction methods to implement deltas in a flexible, but safe way.

Streaming data from SAP to Hadoop

SAP is not very well suited for data streaming. After all, SAP's business applications are transactional systems, running on a relational database. Therefore, it is not as easy as appending sensor data to an existing table on Hive using a solution such as Kafka to stream data from SAP to Hadoop. Some ETL tools support using database mechnisms such as transaction logs to generate deltas, but this is very much application and database specific.

Our solution generates database triggers which capture all inserts, deletes and updates to tables, and can subsequently replicate the data to Hadoop. By scheduling extraction jobs based on a delta collected in this way, you can set up a near real-time streaming of data to Hadoop.

Frequent updates to SAP data

SAP data tends to be updated frequently. This is not only new data being added, but true updates. This happens when, for example:

- a user changes master data, such as the name of a customer
- a user cancels a document. SAP will update some fields in the document (e.g. the status), or create a follow-up document with a reference to the original document.
- aggregate tables are updated, such as the LIS tables in SAP ERP
- periodic mass processing is executed, such as payroll, dunning, MRP

The consequence is that such updates need to be identified on the SAP side before exposing the data to Hadoop. But also on the Hadoop side, individual records need to be updated. That sounds easy, but actually it is rather tricky. Most processing engines on Hadoop do not readily support updates—SAP VORA being one of the exceptions.

For the Hive store, we implemented such solutions using a request-driven delta-merge into our ETL and integration solution Datavard Glue. Our focus was to enable updates using Hive partitioning features while minimizing the manual work required for implementing relevant scripts and keeping them up to date.

Summary

When using SAP data on Hadoop, we recommend assessing the big picture of how you're going to consume the SAP data. Just focusing on some aspects of ETL simply isn't enough. You need to consider all aspects, including operational costs, regression testing, future stability, scalability and the overall complexity of the solution. For example, custom-made ABAP programs for individual bits and pieces of data seem to be a valid idea, but will prove challenging when it comes to extensions, adjustments and long-term TCO. You need to keep them up to date and perform regression testing, after all.

A

APPENDIX

Links and more information

- Part 1: SAP & HADOOP: Use Cases For Big Data Integration
  https://www.linkedin.com/feed/update/urn:li:activity:6290918362520391680

- Part 2: SAP & Big Data Challenge: Opening SAP for Data Scientists
  https://www.linkedin.com/feed/update/urn:li:activity:6291381379951394816

- Part 3: Data Integration between SAP and Hadoop
  https://www.linkedin.com/feed/update/urn:li:activity:6293064100884942848

- Part 4: Bridging the gap of software logistics
  https://www.linkedin.com/feed/update/urn:li:activity:6295128246312935424

- Part 5: Consuming Hadoop data from within SAP
  https://www.linkedin.com/feed/update/urn:li:activity:6296415476671221760

- Glue demo & recording
  https://www.linkedin.com/feed/update/urn:li:activity:6285454289150312448

- SAP TechEd Snapshot: Everyone Is Looking into SAP HANA and Hadoop
  http://www.datavard.com/en/sap-teched-hana-hadoop/

- “Using Big Data to Drive Customer 360”
  https://www.slideshare.net/cloudera/using-big-data-to-drive-customer-360

- “4 Big Data definitions – which one is yours?”
  http://www.datavard.com/en/blog-big-data-definitions/

- “HADOOP & SAP HANA - The gap between big data platforms is closing.”
  https://www.linkedin.com/pulse/offloading-sap-data-hadoop-nls-jan-mesza-
  ros

Thank you for reading this e-book. Don´t hesitate to contact me if you want to learn more.

Goetz Lessmann
TalkToGoetz@datavard.com