Enabling Data Driven Projects for a Modern Enterprise

Artyom Topchyan <a.topchyan@reply.de>
Yerevan State University,
0025 Alek Manukyan 1, Yerevan, Armenia.

Abstract. With the growing volume and demand for data a major concern for an Organization trying to implement Data Driven projects, is not only how to technically collect, cleanse, integrate, access, but even more so, how and why to use it. There is a lack of unification on a logical and technical level between Data Scientists, IT departments and Business departments, as it is very unclear where the data comes from, what it looks like, what it contains and how to process it in the context of existing systems. So in this paper we present a platform for data exploration and processing, which enables Data-Driven projects, that does not require a complete organizational revamp, but provides a workflow and technical basis for such projects.

Keywords: data-driven projects, crisp, Hadoop, data vault, distributed, information retrieval, sandbox, topic modelling, streaming processing, auto-scaling, mesos, kafka

DOI: 10.15514/ISPRAS-2016-28(3)-13

For citation: Topchyan A.R. Enabling Data Driven Projects for a Modern Enterprise. Trudy ISP RAN/Proc. ISP RAS, vol. 28, issue 3, 2016, pp. 209-230. DOI: 10.15514/ISPRAS-2016-28(3)-13

1. Introduction

More and more Organizations are aiming at implementing Data-Driven projects [1][2] which aim at increasing the quality, speed and/or quantity of information gained from Data collected by the Enterprise. The main goal is increasing the quality, speed, and/or quantity of information gain for the purpose of innovation (e.g. innovating a new methodology) or the economic benefit to an organization.

This is particularly challenging for existing Enterprises with years of organizational structure and system already in place, as completely changing the way data is accumulated, handled, shared and used is not feasible. To this end in this work we present a platform for data exploration and processing, which simplifies data driven project by means of intelligent automation. The main goal of the platform is to improve any project that relies on data analysis, but at the same time can coexist with the existing landscape and not require an immediate organizational revamp.

The solution is designed to address three real world view points of issues in a Data-Driven projects flow. We accumulated these viewpoints when working with such projects at large Organizations. These viewpoints are outlined below and represent the challenged a specific parts of the organizations usually faces in the development stages of a Data-Driven project. We outline the issue and shortly outline how we solve them by means of analysis, automation and logical structure.

1.1 Issues from the point of view of the IT department

The issues an IT department has, are often of a technical nature. One major problem is the inability to give users access to raw data. Usually data can only be shared by ways of export tools, which are vendor and apply some transformations to clean up the data, this is often very useful, but in some cases, which we will outline below lead to data loss and or corruption. Another problem commonly faced is the lack of processing resources to use for any type of exploratory large scale processing. Again most system and proprietary and solve a very specific use case the department has. This also extends the first 3 issues in the sense, that there is basically no unified, system independent way to provide data in a consistent format to users. And third there is the issue of unstructured data, such as documents, log files and others. These are most often not stored in a central system,. such as a database, but are scattered around the departments and are handled in very specific ways. This data is nonetheless immensely valuable when combined with user data, and the systems which can load and interpret this data, such as monitoring and operational systems are not designed to provide facilities for data export and analysis outside their specific context. We aim to solve this problem by implementing a so called Enterprise Data Vault, which provides flexibility to ingest any type of data, while preserving its structural and logical relationship. This approach also imposes and standardization on data extraction semantics as well as format. The main goal is so that all data is extracted in a consistent way in the scope of the same framework. A central goal is to also support near-real-time ingestion for source that can benefit from this and to facilitate dynamic and evolving schemas as well as a multitude of formats.

1.2 Issues from the point of view of Data Scientists

The main issues Data Scientist encounters at most Data-Driven projects are Organizational or Information sharing related. It is very often unclear who owns the data, how much there is, what it contains and where it is stored. It is practically quite challenging to keep the data with the same structure ad without introducing a lot of structural changes as systems evolve and change with time. This is a usual issue with change management in very large Organizations and is very complex to solve directly. Most of this information is contained or can be inferred from project and data documentation as

well. This information is readily available, but scattered around dozens of systems and departments with no simple way to search or analyse it from a unified interface. Even if the data documentation is found and stakeholders are contacted there is no guarantee the data is usable as it might contain dropped fields, wrong data types and so forth, is essence no data profile is available. This is often a data quality and governance issue, but this on its own can be challenging with a very large volume of data in some systems and is often ignored or not update as projects continue and technical staff comes and goes. We aim to solve this problem by building a large scale and intelligent index of all project documentation and and information about data and people responsible for it. This model should capture changes to data, project and technical personnel without being influenced or depending on manual updates and necessity for bookkeeping. We extract the mapping of data source to project based on documentation content, the mapping between data, projects and key knowledge owners based on the data and document metadata as well as documentation content analysis. These are then connected to each other based on the data in question and augmented with comprehensive data profiles, which offer data consistency and distribution at a glance. This will allow Data Scientist to understand a lot about the data even before getting access to it. Once a decision has been made, that the data is useful a so called isolated Sandbox environment will be provisioned, which has access to the data and a cluster of computing power. The Data Scientist will have full control in his isolated environment with tools in place to fetch program dependencies, collaboration and visioning. The environments are isolated, but are collocated on the same hardware with dynamic resource allocation and monitoring, which allows a high degree of efficiency in such a highly multi-tenant environment.

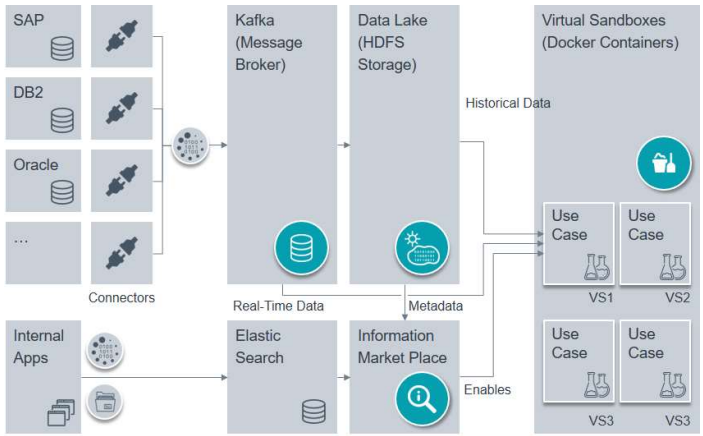


Fig. 1. Operational Data Platform

1.3 Issues from the point of view of the Business department

The business department being the most knowledgeable of business processes and what the data actually means in a business context is most often very much interested in a simple way of querying and exploring data by means of structure queries or analytical views. This is interesting for them in order to understand what effect their business decisions potentially informed by Data Scientist and Analysts, actually have. This is in most cases quite similar to what the Data Scientist and Analysts want as well. The key difference is, that the Business department is most interested in how effective all the DataDriven projects are, be it the value they bring or in case of failed projects, how much resources were wasted. We address this by the combination of the entire platform of the Enterprise Data Vault and Sandbox environments greatly increasing productivity and minimizing waiting time to access and process data, and the Information Marketplace greatly decreasing the time required to analyse and understand if a use case is viable considering the information present.

We define our proposed solution as the Operational Data Platform. The solution is based on modern concepts of resource scheduling [3][6] and immutability concepts to achieve flexibility and scalability. The entire platform is built around the concept of data streams [4]. The platform has been successfully deployed and is being used by a Large Automotive Organization in Germany to investigate Data-Driven project based on large variety of data, such as analysis time series Car Telematics Data to predict faults or patterns, analysing textual Quality Assurance data and others. A high level overview of the entire solution is presented in Fig.1 and contains all the building blocks and their connections. We will go into more detail about each individual component in the following the chapters. But first we will define in more detail, what we understand as a Data-Driven project at an Enterprise.

2. Data Driven Projects

To clarify the problem we are approaching lets define in more detail what a Data-Driven Project is and what the life cycle and goals are. A Data-Driven project aims at increasing the quality, speed and/or quantity of information gained from Data. Any type of data can be used varying in size, source and business/operational importance. Such projects usually involve Data Scientists, Business and IT working together to build up use cases by analysing, processing and presenting data in a meaningful way. The result of the project may be a report, dashboard, or a web service used by other systems. These are very involved projects and require a great degree of domain, statistical, modelling as well as large scale data processing knowledge. To highlight the problem we are solving, lets take a typical datadriven project lifecycle at a major enterprise. Most Data-Driven project

follow a variation of the Cross Industry Standard Process for Data Mining project lifecycle [8]. This varies from organization to organization depending on the maturity of the Data-Driven mindset, but CRISP is one of the most widely accepted approaches to such projects. The life-cycle usually consist of six stages of development, which can be iterated upon and repeated or completely abandoned. A slightly modified version of CRISP is:

1 Business Understanding

This initial phase focuses on understanding the project objectives and requirements from a business perspective, and then converting this knowledge into a data-driven problem definition, and a preliminary plan designed to achieve the objectives [8]. This can vary from common cases such as Fraud Analysis to large Scale Car Telemetry analysis. Outlining the case often takes on month or more.

2 Data Understanding

Once the cases is more or less clear it has to be clarified if there is any data to support it. The business department often knows what data can be used, but more substantive knowledge of the data is required, so it is a task of the business department and the Data Scientists to find out who owns and has knowledge of any data related to the case. This can take a very long time and is notoriously difficult in a large organization, because it is most often unclear who the data owner is and who is knowledgeable about this data. These are quite often different people. In our experience, this process might take upwards to two months' time and often it is discovered there is not substantial data to support the use case. This is already approximately 3 months on a case that potentially is not even possible.

3 Data acquisition and preparation

If the data is present the next challenge is to acquire even a small sample of the data, which is usually customer data and is not shared easily between departments. This is again a costly process and can take up to two months. Luckily Data Scientist can start work on at least sample data if it can be supplied. But this again does not guarantee any data will arrive in the end. The data has to be transferred and transformed into a usable state. This may also take a large amount of time and is quite often the most time consuming phase that involves technical work. In our experience this process is repeated multiple times throughout the project and each iteration may take weeks. At this stage it can be found out that the data is corrupted, with columns missing or being uninterpretable due to formatting loss or it is just very sparse.

4 Modeling

In this phase, various modeling techniques are selected and applied, and their parameters are calibrated to optimal values. This is dependant on what type of problem is being solved and is greatly influence by the type of data

available. Some techniques have specific requirements on the form of data. Therefore, stepping back to the data preparation phase is often needed [8]. Depending on the problem and budget this can take anywhere from one to two months.

5 Evaluation

At this stage some result can be shown and the models and approach evaluated, preliminary results discussed and it is decided if there any value in continuing the project.

6 Deployment

Creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that it is usable. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as establishing a fault-tolerant and monitorable process to repeat the modelling and provide periodic or real time results to the user. This is usually done in conjunction with the IT department. This process has to often be subject to the requirements and limitation of the departments hosting the solution, which greatly limits flexibility and performance. In our experience this is actually the most complicated step based on the problem and can take many months.

It should be noted, that quite often these project involved external suppliers, which means they are inherently more expensive the longer the projects take. So in essence a project can fail on at least 3 separate stages. Which might take months and can be very expensive and deliver few to no results. With our proposed solution we try to tackle this time and knowledge requirement and achieve faster success and faster failure time windows. In the solution we presenting, Data Scientist will have access to all of the project documentation and with the addition of intelligent search capabilities, they can quickly find what data is about or who to ask about it. In our experience this greatly facilitates the Business Understating. Using this information, they can easily transition to the Data Understanding phase and analyses the data source and discover the data-profile, what the data actually contains and if it is actually useful and contains the required information. If the data is there the only thing stopping the Data Scientists is the knowledge about the form of the data and authorization to access it. This is again streamlined as all the data is stored in a central repository in a very strict logical hierarchy and the data schema and access rights are presented with the data itself. This will greatly simplify some of the usual administrative and mechanical tasks a Data Scientist would have to go through in the Data preparation stage The last step is to get the authorization to use the data and request and analytical environment to process it. In contrast to how this is usually handled, we try to automate the process as much as possible. The only thing required is what

data is needed, for how much time and what kinda of processing power and tools are required. The data owner if automatically notified and they can specify which parts of the data can be used and this permission is granted based on a fine grained access control scheme. The requested environment is the automatically provisioned with all the analytical and collaboration tools built in. This aids the Data Scientist in the Modelling and evaluations steps as they have access to a fit for purpose environments, where many Data Scientists can collaborate and iterate on their findings. Once the case is ready our platform greatly simplifies the deployment process as the data, resource and tool requirements are fairly transparent at this stage. To this end lets go into more detail about the components of the solution that allow this flexibility, starting with how the data is collected, processed and stored.

3. Enterprise Data Reservoir

In order to enable all truly dynamic and Data-Driven projects, Analysts and Data Scientists need to have unimpeded access to basically all use case and customer related data. A logical first step is to aggregate all the data of the Enterprise in a central place, so that one central source of truth is viable to Data Scientists and the Business department. This is what has been traditionally done in the Data Warehousing world. Data Warehouses are central repositories of integrated data from one or more disparate sources. They store current and historical data and are used for creating analytical reports for knowledge workers throughout the enterprise. Examples of reports could range from annual and quarterly comparisons and trends to detailed daily sales analysis[16]. The problem with this approach is, that it is very structured by definition. Relationships are all predefined and data is usually transformed between the extraction and load steps to answer very specific questions. This is keys for performance and to reflect enterprise specific requirements for these views. For most Data Science use cases this is not optimal, because structured transformation tend to remove some useful data as they reflect the needs of the application, which may or may not align with the goals of the data scientist.

Let’s take a project that aims to analyze text descriptions of defects in different car models of an automotive manufacturer. These text descriptions also contain information about, which car models the issues are about. Now let’s say the business department is interested in using this data to build a report of faults by car to analyze the efficiency of the Quality Assurance department, but some older models that are not produced anymore or have their names changed since the inception of the system. So the view in the data-warehouse should contain the actual names and only the specific models the business department is interested in. This works very well and provides the report that the business department is interested in. Now a Data Science project is started to analyze these descriptions and use them to predict

possible future issues in newer models based on the problem description and historic data [10]. The data in the Data Warehouse would be extremely biased towards specific problem for specific cars, it would also essentially not contain some models or contain ambiguity between the model dictionary the data scientist has and what the data contains. would lead to the Clustering and Classifications models not generalizing to the entirety of possible issues and cars. The answer would be to go to the actual data source and use the raw data, in its original form. This is often very complicated or even not possible due to the structure of the Enterprise and the way data ownership is handled and is very costly to implement on project by project basis. The currently accepted solution for this is to load all enterprise raw data into a a single repository, a Data Lake [11]. A |Data Lake is a method of storing data within a system that facilitates the collocation of data in variable schemas and structural forms, usually object blobs or files. Data Lakes are a popular way to store data in a modern enterprise. The usual architecture is fairly similar to a Data Warehouse, with the exception of almost all transformation. The main role of a Data Lake is to serve as a single point of truth, which can be used to create use cases, which join and analyse data from multiple departments. It addresses issues of scalable and affordable storage, while keeping raw data intact by loading the data unchanged into a distributed file system, like the Hadoop File System [12] and provides and batch oriented integration layer for downstream consumers and use cases. This approach has a lot of merits, but most implementation lacks certain key aspects, which are more and more important for a modern business, such as self-describing data, tolerance to changes in the data source and support for low latency data sources. For the purpose of this platform we have adopted a variation of the Data Vault approach coupled with some concept of a Data lake implemented on top of a variation of a Lambda Architecture.

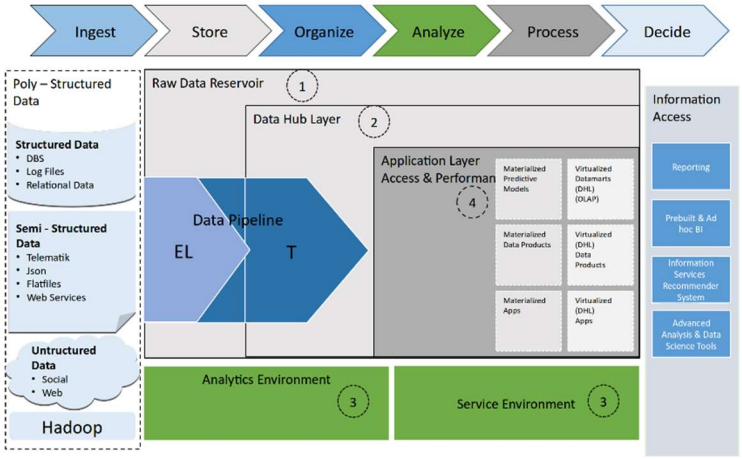
3.1. Data Reservoir

In contrast to the commonly accepted practice of just ingesting all the data as is into separate parts of the filesystem and then transforming it into a meaningful state, our approach to model the data in a more structured as we impose the format, structure and extraction semantics, but we still remain flexible as the data is still ingested in almost raw form and Data Vault modeling is only applied to sources for which it makes sense. Our approach is based on 6 layers:

1. Ingest
2. Store
3. Organize
4. Analyse

5. Process
6. Decide

The overall structure is outlined in 3. It includes all the Layer from Ingestion to Serving(Decision) layer and based on our experience key components are the Lambda Architecture underpinning this and the Organizational Hub layer, modelled as a Data Vault. The Ingest, Store, Process and Analyse are layers, which are mostly based on a variation of a Lambda Architecture and also include the Raw Data Storage layers, which is essentially a Data Lake implementation. This is outlined in Fig, 2.



Fig, 2. Data Vault Architecture

3.1.1 Lambda Architecture

Let present the Lambda Architecture from the point of view appropriate in our context of data processing. The Lambda architecture is a data-processing architecture designed to handle massive quantities of data by taking advantage of both batch- and stream-processing methods. This approach to system architecture, used in our context, attempts to balance latency, throughput, and fault tolerance by using batch processing to provide comprehensive and accurate views of batch data, while simultaneously using real-time stream processing to provide views of online data. The two view outputs can be joined before presentation [14]. In a classical Lambda Architecture [15]:

1. All data entering the system is dispatched to both a batch layer and a speed layer for processing.
2. The batch layer has two functions:
   (a) Managing the master dataset (an immutable, append-only set of raw data)
   (b) Pre-compute the batch views.
3. The serving layer indexes the batch views so that they can be queried in low-latency, ad-hoc way
4. The speed layer compensates for the high latency of updates to the serving layer and deals with recent data only.
5. Any incoming query can be answered by merging results from batch views and real-time views.

This interconnections of the layer is outlined in Fig.3.

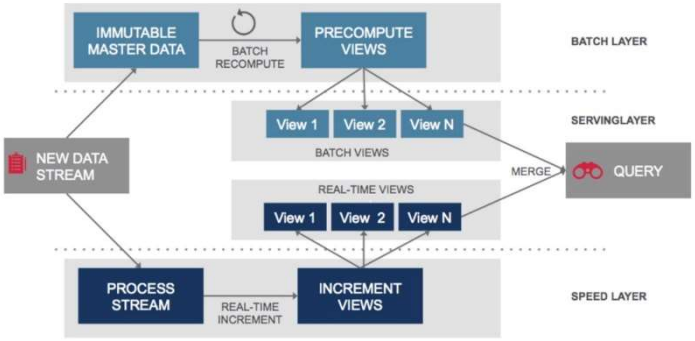


Fig. 3. Lambda Architecture

The main rational for a Lambda Architecture is to efficiently answer a query overthe entire dataset. The challenge is that running arbitrary functions of an unbounded set of data is very expensive. Thus the Lambda Architecture decouples this two processes and offloads only efficient simple queries to the real-time layer as outlined in 4. To achieve this the Lambda Architecture models the ingestion layer as a stream of datum's, which are ingested into a Distributed Message Queue, which allows to both create an Immutable Master Dataset, which is an append only historical log of all events and transactions, and a real-time layer, which answers very specific questions on the incoming stream of data. Having an Immutable set of data not only allows to easily build up analytical use cases, as well as creating reliable point in time snapshots of data, which allows to recreate the full dataset at any point in time. Most commonly in batch view in the Lambda architecture is where the data persistent to a Filesystem takes place, this is a simplification as achieving low latency exactly once persistence is quite challenging. So the persistence layer is essentially part of the processing layer. We have certain variations from the normal Lambda Architecture. Specifically, the ingestion layer is fully decoupled from the processing layer and also supports near real-time incremental persistence to HDFS as opposed to a scheduled batch job, as in the standard Lambda Architecture [14]. This allows the ingestion layer to be parallel

to the actual views that have to build on the data. We have found this to be most beneficial for large Organizations, where it is sometimes unclear what the default views and queries are before analysing the data. This also allows for the same degree of consistency offered by the Lambda Architecture, but with less latency.

3.1.2 Data Hub

The Data stored in real-time in the data reservoir has to be structured in a meaningful way for actual business application to benefit from them. Views on the stored data are created based on the business demand, existing project, some analysis or processing a Data Scientist carried or a generic view of the raw data. This might be a static periodic view or a real-time streaming view integrated into a serving layer as described above. To keep the modelling consistent, we standardize on the Data Vault approach. Data vault modeling is a database modeling method that is designed to provide a modular way to incorporate multiple operational systems [9]. It is also a method of looking at historical data that, apart from the modeling aspect, deals with issues such as auditing, tracing of data, loading speed and resilience to change, which is critical for the Data acquisition and Processing steps. Data vault contains three different types of technical entities

1. Hubs. Hubs contain a list of unique business keys with low propensity to change.
2. Links. Associations or transactions between business keys (relating for instance the hubs for customer and product with each other through the purchase transaction) are modeled using link tables.
3. Satellite. The hubs and links form the structure of the model, but have no temporal attributes and hold no descriptive attributes. These are stored in separate tables called satellites.

All the principal data sources, such as CRM data, Telematics data, Accounting Data, Product detail, Quality Assurance data are modelled in the Data Vault approach and exposed as views and APIs. The combination of real-time data ingestion, processing and Data Vault modelling leads to a simple and flexible Data Model, which solves the problem we posed in this section in a more robust way. Historic data is easily accessible in real-time, while rigid defined views are still available for more structured use cases as well as enabling a multitude or real-time use cases.

3.1.3 Implementation details

The described data model and Lambda Architecture was implemented on top of the described platform. With all the services running as immutable services in a shared cluster environment, managed by a central resource manager and service discovery solution.

The central part is the Apache Kafka broker [4], which serves as the main coupling layer for the entire solution. As described in the previous chapters we employ the

Staged Even-Driven architecture for most of the transformation, that has to implemented on the platform, but the data collection and storage is modeled fully after the Lambda Architecture.

All data source and sink connectors are running by reading and writing messages to Kafka, most of the job progress tracking, reporting and orchestration is done by using Kafka. This allows for the connectors to run in a distributed fashion with automatic or manual scaling done via Apache Mesos [3]. More connectors are launched based on demand and are fully immutable, which means if a connectors crashed, a new instance will be transparently started somewhere else and it will resume work from the last offset. This is quite critical in order to always provide a consistent and up to date view of the source systems to the Data Scientist using the platform, which is a central part of almost all the CRISP stages. Currently we support a variety of connectors specifically developed for the described platform, all of which are based on the Kafka-Connect framework [20] which provides a comprehensive framework for building data extraction and loading connectors. Compared to other similar systems the advantages are a standardized way of keeping track of job progress and resuming on failures, as well as simplified scaling due to the reliance on Kafka for balancing between processes. In our experience this greatly simplifies operation, creating new connectors as well as being agnostic to low latency or batch extraction/load as all the data is loaded into a queue and this can happen both in real-time or periodically.

In the context of the architecture these represent the framework for extracting data from almost any system a large organization might have. As each connectors deals with rather different types of systems, these connectors are also a reference implementation for most types of storage systems.

- JDBC connector
- File Stream connector
- Elastic Search connector
- HDFS connector
- Binary File connector
- Others

An important functionality required for building a useful Data Vault model is correct data partitioning and handling data schema changes. If the model stops functioning after column name changes or a full scan is always required to execute common queries, then the model has diminished value as an organization wide repository of data.

As an example let's take sensor data or documents arriving in a stream to the platform. A very common query one would need to do to analyze this data is to sort by date. If this data is stored as is in a single flat directory, this can be an extremely expensive query. For example, car sensor data may increase by terabytes each month. To address this common query pattern, we employ time based partitioning, which enables efficient filtering queries on specific data partitions. We employ a time based

partitioning with a maximum granularity on a month. The actual record assignment to time partition depends on what delivery semantics we are using. Process-time (when the event was received) and event-time (when the event actually happened). The choice if either depends on data source.

Now let's also consider what happens if a column name or data type changes. This can lead to inconsistency and even break some process that are already running if this data is ingested. To solve this all data is stored in a efficient binary file format, that supports schema evolution, Apache Avro [18] This means that event is stored in an organized manner with the current schema always stored with the data, which makes the described example much easier to handle.

Another very important aspect is exactly once delivery semantics. We are storing our data in multiple storage services, such as the Hadoop File System(HDFS) [12] and Elastic Search [13] In the case of Elastic Search, which supports updates, data can be written multiple times, without greatly impacting the system, as each write essentially overwriting parts of the record. In the case of HDFS, data can be append-only by design. For which the HDFS connector is developed in such a way that a datum is ingested only once, based on its ID and latest state of ingestion being stored in a transaction log stored in HDFS and a two phased commit process. A first commit is happening when the data is read from Kafka and then a second one after it has been successfully written in HDFS. The second commit is written in the transaction log stored in HDFS.

4. Information Marketplace

One of the biggest challenges faced by an Organization when exploring possible use cases for Data Scientists is knowledge sharing and transfer. Large Organizations have a large number of departments, which vary widely based on their size, project they take on and the way these projects are completed and documented. This leads a large variety of data sources, column names and documentation being created on the same subject by a large number of stakeholders from different departments some of whom might not be part of the Organization anymore.

This leads to challenges for Data Scientists and the IT department, which have to identify the relevant information and people or documents describing the data, especially when the project involves more than one data source.

With the use of the described model, we can provide Data Scientists with easy access to well partitioned and self-describing data, which should simplify the problem.

On the other hand, what is provided, is a technical description of the data. This approach shows where the data comes from, how and when it was stored, and what did it look like at a specific point time without breaking compatibility across the dataset. What is missing from this model is a functional description of the data,what the data actually means, how it is used and by whom. For example, documentation generated by the organizations. The problem is, such documentation is usually

scattered around different departments and is large in volume. So it is often unclear if certain topics are documented or not and if yes, how to access them.

To bridge this issue, there are large undertakings for an Organization wide change management and pushes for standardization. On a technical level this changes translate to the centralization and standardization of project related documentation as well as rigid data views in a central database. This role cannot be filled by a normal Enterprise Data Warehouse and thus requires the creation of Organization wide specialty tools and repositories for Data and Knowledge and complicated integration layers providing each department with access and management capabilities.

In reality this is a vast and complex process and can cost a large amount of money and resources from the side of the Organization and in some cases might decrease productivity. Each individual department has an approach of managing projects and in most cases such a monolithic system allows for less flexibility for individual departments. This may lead to decreased productivity. The learning period for a complete change and standardization of such processes can bring an entire department to a halt for an extended period of time.

As one of the components of our system we propose an advanced text search and schema analysis based approach, which is simpler on the organizational level as it does not require to be integrated organization-wide, and as a minimum provides a much simplified approach for Data Scientists to explore the data. The Information Marketplace(IMP) is a tool that provides an Expert with an easy to use and rich way of exploring data and connections of data.

Now we will outline the functionality in more detail and technical implementation, such as the architecture and algorithms used.

4.1 Functionality

The goal is to provide all relevant information (data, descriptions, contact links) and make the existing environment searchable and discover new connections in the organizational data. The Information Marketplace provides the user with a guided search of:

- Project Documentation
- Knowledge Owners and Information
- Metadata
- Data Sources Schema and Structure
- Data Profiles

This includes information on both data source and project of the Organization as well as all new projects and data created using the platform. This means that the Information Marketplace not only contains information about the data sources and documentation of the organisation, but there is a feedback loop that feeds all the generated data and documentation back into the IMP. Benefits are exploitative information discovery and faster implementation of analytical and data science use

cases due to direct access to relevant project documentation and overview of related data as well as more information on what other people have tried on the platform.

The functionality is exposed as a reactive text-based search of the outlined data types. The search provides the following

- main views of the data:
- Document Full-Text Search View
- Search View based on Author, Time, File-Format, Department, Datasource
- Search View based on extracted keywords, contexts, summaries
- Related document View, based on contexts and keywords
- Datasource profile view

All these views include, match highlighting, facets for author, document type, topic, date and etc. Each document can also be viewed in more detail and the classified by data sources, author, extracted topics and user comments are outlined.

The relevance score of each document is calculated based on text matches to the document content, the topics describing that document, data source and authors.

The Information marketplace also contains and displays all the automatically extracted schema data from all data sources, this is based on a central Schema Registry, which is automatically populated for all data in the Data Vault.

4.2 Implementation Details

The Information Marketplace itself a stage based stream processing application. This architecture was used for its flexibility, and for its simplicity when reasoning about scalability and fault-tolerance. The IMP holds an index of 10s of GBs of documents and this is expected to grow even larger in the near future. Apart from the documents themselves, large topic models and word vectors models are used an applied on a stream of incoming documents. The high level architecture is presented in Fig. 4.

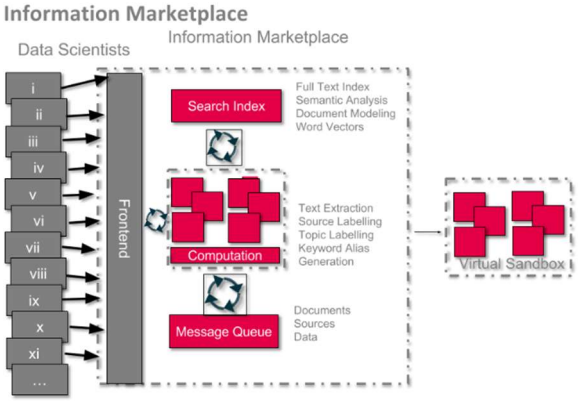


Fig. 4. IMP high level architecture

1. There are 8 principal stages of processing a document goes through before being accessible for searching:
2. Document load from source
3. Document original format to plain text conversion with metadata extraction
4. Map document to specific organizational data source
5. Extract document summary
6. Extract semantic representation of the document(contexts)
7. Find closely related/similar documents
8. Find Departments a data source might refer to
9. Index document

4.2.1 Document load and detection

This stage contains a distributed directory watcher, which watches for filesystem event and emits these events into a distributed message queue. Documents may reside on a filesystem in every Department. We need to detect all the files already there and detect the creation of new files to fulfill the low-latency requirement.

4.2.2 Document conversions

This stage implements a distributed streaming application, which consumes the events from the watcher and extracts plain-text content and metdata. This process scales by launching more instances. This is done manually or dynamically based on the stream of incoming documents. Processes can be pre-allocated based on the number of documents committed and average processing time. This functionality is based on the work presented in [17] but solves the optimization problem by applying an evolutionary algorithm similarly to [7].

The consumption is balanced across the processes using the native Kafka group balancing and offset management [19].

4.2.3 Document context extraction stage

This stage extracts the contextual topics of each documents. The documents are modeled using [23] [22]. This processes are implemented with both learning and inference functions. We are using Online Latent Dirichlet Allocation, which allows online updates to the model based on incoming documents. So each document received is added to the model and after that the topics are extracted, this allows us to continuously update the model as well as extract topics from documents. The updated model is periodically committed to disk for storage, scaling is achieved by keeping multiple copies of the process running. The predicted topics are attached to the document as a metadata field, this is necessary later on for ranking the matches.

4.2.4 Document data source classification

This stage is responsible for attaching a data source label to the document. Often it is not known if a document is referring to a certain data source in the company. This is achieved by using set distance between the schemas in the schema registry, short descriptions of the sources and sentence per sentence chunks of the incoming document, based on a modified Jacard Distance metric [21].

Scaling in achieved by launching more of these processes. The consumption is balanced across the processes.

4.2.5 Document Index

This stage is the process that actually indexes the data into a search engine, in this case Elastic Search is used to create a reverse token index of the documents. The data is keyed with the filenames and timestamps and a unique offset is generated via a hash function to ensure, that no duplicate data is written into elastic search even during node failures.

5. Analytical Sandbox Environment

With the use of the Data Hub and the IMP, a Data Scientist can find and request data required for their use cases in a simplified fashion and build up analytical use cases, such as the one described in the previous chapters.

But another issue often faced by large organizations is that even if the data is stored in an easily consumed format, and the Data Scientist knows what data they want to investigate, there are a number of points there are still unclear. Such as how to actually access and process the data, where to develop reproducible results on this data, how and where these will be deployed and used in production.

To cover this points a few things are missing, such as:

- A Consistent environment definition for use case development and deployment
- A flexible yet secure environment for the Data Scientists and Analysts to work
- On demand access to services and processing power
- Dynamic Security service
- A Collaboration Service
- Fault tolerant, flexible storage for project data and models

To this end we propose the analytical sandbox environments, which are generated, immutable environments provided to Data Scientists and Analysts and where they can build up their project on the data. It is a fully isolated environment, where the user can install or download any extra tools they require and is accessible via an analytical and console view. The environment serves as a gateway to the data and the processing back-end of the cluster. The way the sandbox is defined makes it

inherently self-describing, thus making it simpler to deploy in production as the software requirements, resource requirements and data requirements are already defined when requesting the sandbox.

5.1 Implementation

The sandbox environment is also implemented on top of the ODP architecture. Each sandbox is developed as a separate service running inside the environment. We try to provide as many built in services commonly used by Analysts and Data Scientist is Data-Driven use cases. Scientific environment, such as Python, R, Scala and processing environment such as Spark and Hive are included alongside the Hadoop environment.

To provide full isolation for security and resource sharing reasons the sandbox is implemented as a Linux container running a specified set of services, such as:

- Secure Shell access
- Ipython Console, R console, Scala Console
- Hive, Hue, Hadoop
- Pyspark, Spark, RSpark

The services are running in a single environment with shared resource. For collaboration a code repository is provided for each sandbox project and can be shared amongst collaborates. For convenience and fault tolerance the sandbox user directory is mounted unto a Network mount, running on top of a distributed File System to ensure data does not get lost on sandbox restarts as the entire environment is recreated on failures.

As we are running the sandboxes in a shared environment, network isolation and sharing is a large concern. To this end we use a network abstraction to allocate a private IP address for each sandbox [24] [25]. All the user facing services are integrated into the Organizations central user and rights management platform.

This solution has shown to scale well when only limited hardware is available. As the sandboxes are running in a shared environment with a single resource scheduler controlling all the resource, scaling this solution would be trivial as new sandboxes could be allocated to new nodes both on premise and cloud.

6. Conclusion

We described a proposed end-to-end environment for creating and running Data-Driven projects at a large scale Enterprise. The described platform can efficiently manage the resources large number of users and services. On the data modelling side we proposed a flexible way to organize and transform stored Data Sets in order to become ready to answer analytical questions and generate value. We proposed a flexible way for discovering data and interconnections of the data, based on metadata, functional descriptions and Documentation, in an automated and intelligent way, while not requiring a full departmental restructure. We also proposed a dynamic and

scalable sandbox environment to allow collaborative and shared creation of Data Science use cases based on the data and simplify the deployment of these use cases into the production. We believe the approach, proposed platform and its architecture provide a well structured environment to simplify Data-Driven projects at large organizations.

References

[1]. Rahman, Nayem, and Fahad Aldhaban. "Assessing the effectiveness of big data initiatives."2015 Portland International Conference on Management of Engineering and Technology (PICMET). IEEE, 2015.
[2]. Davenport, Thomas H, and Jill Dych´e. "Big data in big companies."International Institute for Analytics, 2013
[3]. Apache Mesos. http://mesos.apache.or, 2015.
[4]. Dunning, Ted, and Ellen Friedman. Streaming Architecture: New Designs Using Apache Kafka and Mapr Streams. O'Reilly Medi, 2016.
[5]. Welsh, Matt, D. Culler, and E. Brewer. "SEDA: an architecture for highly concurrent server applications.".Proceedings of the 18th Symposium on Operating Systems Principles (SOSP-18), Banff, Canada, 2001.
[6]. Verma, Abhishek, et al. "Large-scale cluster management at Google with Borg."//Proceedings of the Tenth European Conference on Computer Systems. ACM, 2015.
[7]. Artyom Topchyan, Tigran Topchyan. Muscle-based skeletal bipedal locomotion using neural evolution. Ninth International Conference on Computer Science and Information Technologies Revised Selected Papers1-6, 2013.
[8]. Shearer C. The CRISP-DM model: the new blueprint for data mining. J Data Warehousing ;5:1, 22, 2000
[9]. Dan Linstedt. Super Charge your Data Warehouse. Dan Linstedt. ISBN 978-0-9866757-1-3, 2010.
[10]. A. Maksai, J. Bogojeska and D. Wiesmann. "Hierarchical Incident Ticket Classification with Minimal Supervision,". IEEE International Conference on Data Mining, Shenzhen,, 2014, pp.923-928, 2014
[11]. Alex Gorelik. The Enterprise Big Data Lake: Delivering on the Promise of Hadoop and Data Science in the Enterprise. O'Reilly Medi, 2016
[12]. Tom White. Hadoop: The definitive guide. O'Reilly Medi, 2012
[13]. Dixit, Bharvi. Elasticsearch essentials, 2016
[14]. Marz, Nathan, and James Warren. Big Data: Principles and best practices of scalable real-time data systems. Manning Publications Co, 2015
[15]. Michael Hausenblas and Nathan Bijnens. Lambda Architecture. http://lambda-architecture.net/, 2015
[16]. Patil, Preeti S.; Srikantha Rao; Suryakant B.Patil. Optimization of Data Warehousing System: Simplification in Reporting and Analysis. //IJCA Proceedings on International Conference and workshop on Emerging Trends in Technology (ICWET) (Foundation of Computer Science) 9 (6):33–37, 2011
[17]. Newell, Andrew, et al. "Optimizing distributed actor systems for dynamic interactive services.".Proceedings of the Eleventh European Conference on Computer Systems. ACM, 2016

[18]. Apache Avro Project. https://avro.apache.org/docs/current, 2015.
[19]. Apache Kafka Project. http://kafka.apache.org/documentation.html, 2015.
[20]. Confluent Kafka-Connect. http://docs.confluent.io/2.0.0/connect, 2015.
[21]. Cohen, William, Pradeep Ravikumar, and Stephen Fienberg. "A comparison of string metrics for matching names and records.".Kdd workshop on data cleaning and object consolidation. Vol. 3, 2003
[22]. Hoffman, Matthew, Francis R. Bach, and David M. Blei. "Online learning for latent dirichlet allocation.".Advances in neural information processing systems, 2010
[23]. Blei, David M, Andrew Y. Ng, and Michael I.Jordan. "Latent dirichlet allocation.".Journal of machine Learning research 3.Jan: 993-1022, 2003
[24]. Project Calico. https://www.projectcalico.org, 2015.
[25]. Jain, Raj and Subharthi Paul. "Network virtualization and software defined networking for cloud computing: a survey.".IEEE Communications Magazine 51.11, 24-31, 2013

Поддержка выполнения проектов, ориентированных на данные, в современных предприятиях

Топчян А.Р. <a.topchyan@reply.de>
Ереванский государственный университет,
0025, Армения, г. Ереван, ул. А. Манукяна, дом 1

Аннотация. С ростом объема и спроса на данные основными проблемами организаций, которые пытается реализовать проекты, становится не только то, чтобы технически собрать, очистить, интегрировать данные и обеспечить к ним доступ, а в большей степени обеспечение понимания того, как и зачем их следует использовать. Отсутствует взаимопонимание на логическом и техническом уровнях между специалистами по обработке и анализу данных, ИТ-подразделениями и бизнес-подразделениями, поскольку неясно, откуда происходят данные, как они выглядит, что содержат, и как их следует обрабатывать в контексте существующих систем. В этой статье мы представляем платформу для исследования и обработки данных, что позволяет выполнять ориентированные на данные проекты без полной переделки организационной структуры предприятия при наличии поддержки требуемых процессов и технических средств.

Keywords: проекты, ориентированные на данные; crisp; Hadoop; data vault; sandbox; Mesos; Kafka

DOI: 10.15514/ISPRAS-2016-28(3)-13

Для цитирования: Топчян А.Р. Поддержка выполнения проектов, ориентированных на данные, в современных предприятиях. Труды ИСП РАН, том 28, вып, 3, 2016, стр. 209-230. DOI: 10.15514/ISPRAS-2016-28(3)-14

Список литературы

[1]. Rahman, Nayem, and Fahad Aldhaban. Assessing the effectiveness of big data initiatives. 2015 Portland International Conference on Management of Engineering and Technology (PICMET). IEEE, 2015.
[2]. Thomas H. Davenport and Jill Dyche. Big data in big companies. International Institute for Analytics, 2013
[3]. Apache Mesos. http://mesos.apache.or, 2015.
[4]. Ted Dunning and Ellen Friedman. Streaming Architecture: New Designs Using Apache Kafka and Mapr Streams. O'Reilly Medi, 2016.
[5]. Matt Welsh D. Culler, and E. Brewer. SEDA: an architecture for highly concurrent server applications. Proceedings of the 18th Symposium on Operating Systems Principles (SOSP-18), Banff, Canada, 2001.
[6]. Abhishek Verma et al. Large-scale cluster management at Google with Borg. Proceedings of the Tenth European Conference on Computer Systems. ACM, 2015.
[7]. Artyom Topchyan, Tigran Topchyan. Muscle-based skeletal bipedal locomotion using neural evolution. Ninth International Conference on Computer Science and Information Technologies Revised Selected Papers1-6, 2013.
[8]. Shearer C. The CRISP-DM model: the new blueprint for data mining. J Data Warehousing; 5:1, 22, 2000
[9]. Dan Linstedt. Super Charge your Data Warehouse. 1-3, 2010.
[10]. A. Maksai, J. Bogojeska and D. Wiesmann. Hierarchical Incident Ticket Classification with Minimal Supervision,. IEEE International Conference on Data Mining, Shenzhen,, 2014, pp.923-928, 2014
[11]. Alex Gorelik. The Enterprise Big Data Lake: Delivering on the Promise of Hadoop and Data Science in the Enterprise. O'Reilly Medi, 2016
[12]. Tom White. Hadoop: The definitive guide. O'Reilly Medi, 2012
[13]. Bharvi Dixit. Elasticsearch essentials, 2016
[14]. Nathan Marz and James Warren. Big Data: Principles and best practices of scalable real-time data systems. Manning Publications Co, 2015
[15]. Michael Hausenblas and Nathan Bijnens. Lambda Architecture. http://lambda-architecture.net/, 2015
[16]. Preeti S. Patil; Srikantha Rao; Suryakant B.Patil. Optimization of Data Warehousing System: Simplification in Reporting and Analysis. IJCA Proceedings on International Conference and workshop on Emerging Trends in Technology (ICWET) (Foundation of Computer Science) 9 (6):33–37, 2011
[17]. Newell, Andrew, et al. Optimizing distributed actor systems for dynamic interactive services. Proceedings of the Eleventh European Conference on Computer Systems. ACM, 2016
[18]. Apache Avro Project. https://avro.apache.org/docs/current, 2015.
[19]. Apache Kafka Project. http://kafka.apache.org/documentation.html, 2015.
[20]. Confluent Kafka-Connect. http://docs.confluent.io/2.0.0/connect, 2015.
[21]. William Cohen Pradeep Ravikumar, and Stephen Fienberg. A comparison of string metrics for matching names and records. Kdd workshop on data cleaning and object consolidation. Vol. 3, 2003
[22]. Matthew Hoffman, Francis R. Bach, and David M. Blei. Online learning for latent dirichlet allocation. Advances in neural information processing systems, 2010
[23]. David M. Blei, Andrew Y. Ng, and Michael I.Jordan. Latent dirichlet allocation. Journal of machine Learning research 3.Jan: 993-1022, 2003

[24]. Project Calico. https://www.projectcalico.org, 2015.
[25]. Raj Jain and Subharthi Paul. Network virtualization and software defined networking for cloud computing: a survey. IEEE Communications Magazine 51.11, 24-31, 2013