|  |
| --- |
| **ABSTRACT** |

Organizations are experiencing tremendous growth in their data that is coming in a greater variety of formats year after year. In order to keep up with this trend, big data technologies, such as Apache Hadoop, are playing a crucial role in enabling these organizations to derive value from their data. Having skilled data analysts with knowledge of these new big data technologies is an important component for these organizations to successfully integrate into their existing data analysis workflows. The intent of this paper is to outline some of the challenges faced in large organizations and to provide a use case for evolving from a non-scalable architecture to one ready to take on the challenge of harnessing the value of big data analytics. We will explore the role of the data analyst and the challenges associated with the job. Lastly, we will look to the big data solutions currently available and cover future areas of exploration in terms of where the big data industry is headed.

|  |
| --- |
| **1 BUSINESS CHALLENGES** |

Working as a data analyst in a large Fortune 500 telecommunications company that has a worldwide presence is not an easy task. The data analyst deals with a myriad of disparate data sources and often struggles to get access to many of the data sources required to do their job effectively. The responsibilities of the data analyst generally include the following areas:

* Data Acquisition
* Data Wrangling
* Data Workflow
* Data Visualization

With each of the areas above, there are numerous challenges the data analyst faces on a daily basis.

**1.1 Challenges with Data Acquisition**

Getting access to data is one of the biggest challenges a new analyst must face when starting out in a large organization. In most cases, there is no single data mart with a copious amount of documentation necessary to locate your data ever exists. Instead, finding someone with extensive institutional knowledge, a “data expert,” is a necessity. In most cases, this data expert can point you to the write person or location where the data resides, but in the majority of cases gaining the necessary permissions is required.

With the growing number of data sources within a large enterprise, data is beginning to come in different data formats and having the necessary tools to handle these new formats is a challenge. Some of these formats include Microsoft Excel, comma separated (CSV), tab separated (TSV or TXT), JavaScript Object Notation (JSON), extensible markup language (XML), among others. Some of these data sources are stored in different databases (Teradata, Oracle, or MySQL) or are accessible through application programmable interfaces (APIs) from third party applications. Many data analysts do not possess the necessary skill sets to transform this data into usable formats to conduct a particular analysis.

**1.2 Challenges with Data Wrangling**

Once the data analyst resolves the data acquisition challenges, they are next faced with the data “munging” or “wrangling” phase. The process of data wrangling is merging, transformation, and cleansing of data coming in from the various data sources. Knowledge of unique identifiers is crucial in order to tie all of these sources together into a meaningful dataset required before analysis can begin. Many data analysts resort to conducting the majority of their data wrangling in an application such as Microsoft Excel. This approach tends to be tedious, repetitive, and extremely manual, which is a significant bottleneck in data workflows. Another major challenge in some organizations is the volume of the data needed for the analysis exceeds the computing limits of the data analyst’s computer.

**1.3 Challenges with Data Workflow**

As mentioned in the previous section, many organizations do not set the data analyst up for success due to the significant workload required to answer the questions coming in from their business stakeholders. Due to the constant change in business requirements and the lack of a centralized data repository, the data analyst must make do with the tools available to keep up with the requests for analysis from the various teams they support.

It is not uncommon for a data analyst to receive a weekly email with an attached Microsoft Excel file that is then combined with a mapping file from another data source and many other manual cleaning and transformation tasks are applied. This process is manual, time consuming, and by the time the analysis is completed and in the hands of the end users, the data is already stale. Due to the labor intensive nature of an operation like this, scaling the analysis capabilities for the organization is only achievable by hiring more data analysts.

**1.4 Challenges with Data Visualization**

Deriving value from data and providing actionable insights to end users is the ultimate goal for any data analyst. One of the major challenges with providing data that drives action is that the data analyst is often working with outdated information or is looking into the past instead of predicting where the business might end up later on down the road. For some, creating pivot tables in Microsoft Excel or charts in Microsoft PowerPoint is the extent of their data visualization strategy. The reason for this is often due to the fact that end users are familiar with these formats and are more comfortable exploring the raw data further on their own. The negative impact this has on the business is that it takes too long to produce these actionable insights and the data is often already out of date before action can even be taken by the end users.

**1.5 Business Use Case**

Given the challenges covered above, creating an efficient workflow for data analytics within an enterprise can be a very arduous and political process in order to get the necessary data and tools required to execute against a data strategy. In the sections to follow, we will cover the various big data technologies available, the process in which a data analytics framework exists today, and identify opportunities to leverage some of these big data technologies to scale within a large organization.

|  |
| --- |
| **2 BIG DATA OVERVIEW** |

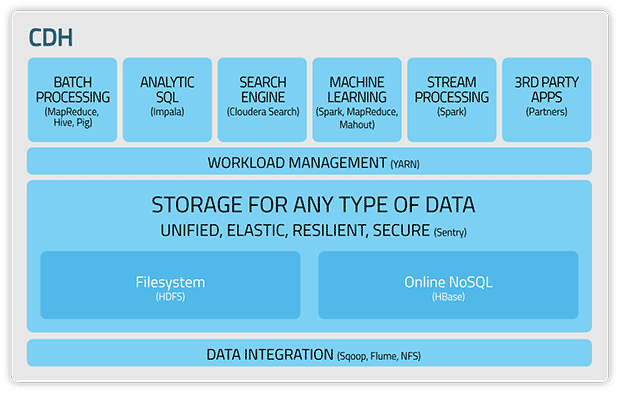
Because there are many organizations facing similar big data challenges, there have been an explosion of new technologies to address these issues including the following: Apache Hadoop, NoSQL databases, Hive, Pig, Flume, Kafka, and Storm (not an exhaustive list). Each of these technologies address a particular challenge including scalability, schema-less databases, distributed query analysis, data streaming and data workflow management. In this section we will cover a few components of the big data ecosystem. There is a large open source community contributing to these areas including the Apache Foundation, Hortonworks, Cloudera, MapR, and EMC’s Pivotal.

**2.1 Apache Hadoop**

After the release of “The Google File System” whitepaper in 2003 by Ghemewat, Gobioff, and Leung, Hadoop was created by Doug Cutting and Mike Cafarella in 2005 in what was originally developed to support the Nutch search engine project. Hadoop, as defined on the Apache Hadoop website, is a framework that allows for distributed processing across clusters of computers. The design was in a way that allowed for scaling up to thousands of nodes and was highly fault tolerant. This enables people to deploy a production Hadoop cluster on commodity hardware at a relatively low cost. This framework allows for users to work with much larger datasets due to the distributing processing power through its MapReduce system. There are many other open source projects which leverage Hadoop as the core system in which data can be stored or queried against from the Hadoop Distributed File System, or HDFS.

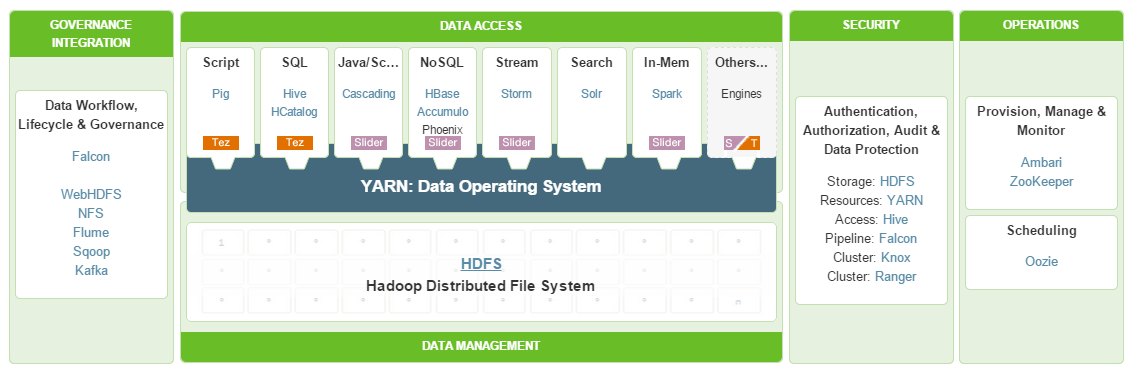
**2.2 Hadoop Distributions Overview**

There are many distributions of Hadoop, however, according to an article on Network World, the top Hadoop distributions of 2012 include **Cloudera** ($56M), **MapR** ($23M), and **Hortonworks** ($18M). Cloudera’s open source distribution is what they call the CDH, which includes a full stack of components as follows:



CDH includes a NoSQL solution with HBase and analytic SQL which they call Impala. Impala is a very fast querying language which is perfect for the impatient data analyst conducting exploratory data analysis. Apache Spark is a very exciting addition to the ecosystem as it provides in-memory analytics and real-time stream processing for Hadoop.

Another major Hadoop distribution is offered by Hortonworks, which they call the Hortonworks Data Platform or HDP. According to the Hortonworks website, HDP is the “only completely open Hadoop data platform available” (2015). HDP offers a full suite of Hadoop essentials as shown in the architecture diagram below:



As you can see, HDP contains much of the same components as CDH, including Pig, Hive, HBase, and Spark. Some of the additional capabilities included with HDP include Storm and Kafka, both of which are very popular for data stream processing.

Because organizations are able to store much more data than ever before, deriving value and actionable insight from that stored data is crucial. There are many ways to access, manipulate, and query data stored in HDFS, most notably include Pig and Hive. Pig is a high level language used for expressing data analysis programs and doing actual manipulation directly onto HDFS. The primary benefits for Pig include the language’s low learning curve, optimization capabilities and extensibility with other languages. For the data analyst more comfortable working in SQL, then there is a solution with Apache Hive. Hive is a data warehouse software that takes SQL statements and converts them to map reduce jobs to be ran on a Hadoop cluster. This is extremely valuable to the less tech savvy analysts who just want to get to their data as quick as possible without having to write their own map reduce jobs.

**2.3 Benefits of Big Data Technology**

The big data technology covered in this section provide the necessary tools required in order to scale a data analytics strategy at a fortune 500 organization. There are a variety of uses for the technologies for a variety of needs including data acquisition, data stream processing, and ad hoc analysis. The days of the data analyst being limited by the resources on their personal computer are over for those who embrace the Hadoop framework and the big data tools available. In the sections to follow, we will describe the current workflow and a proposed architecture required to achieve the scalability of an enterprise wide data analytics strategy.

|  |
| --- |
| **3 BIG DATA ARCHITECTURE** |

Within our team, we have been tasked with evolving our data architecture from one that is maintained primarily through Excel to a more automated, scalable, and self-service business intelligence solution that will empower management to make better decisions with relevant and timely data. To accomplish this, we propose leveraging a data lake that can be used for storing our organization’s raw data, as well as an intermediate staging area where the data wrangling can take place prior to pushing the data to a visualization platform. One of the primary benefits of this architecture is that the data storage and processing can be conducted in the same location – this allows us to programmatically develop the business logic and apply it in an automated fashion to new data as it enters the data lake. Based on the velocity of the data, we can then schedule when each of the sources needs to be refreshed on our visualization platform, ensuring that we always are presenting the freshest version of our data to the end users. Initially, our goal was to create this data lake using a traditional Linux server, writing Python scripts to wrangle the data and relying on shell scripting and cronjobs to schedule automated refreshes on our separate Tableau server. However, through this course we have learned that a Hadoop cluster could serve a similar purpose and have begun investigating its performance as a central component of our architecture.

After we have completed the process of automating data acquisition and wrangling for all of our disparate sources, there are many additional tools that we would like to integrate into our Big Data architecture so as to add further value on top of the data. A number of applications within the Apache Spark project look especially promising given that it is capable of processing data in a fraction of the time that it Hadoop requires to accomplish the same task. Spark SQL and Spark Streaming would allow us to query massive amounts of data without concerns for volume or velocity, while MLlib would allow us to apply machine learning algorithms at scale to the incoming data without concerns regarding performance. Additionally, the ability to access all of these libraries using the Spark Python API gives us the ability to create parallelized applications capable of conducting massive queries while simultaneously applying machine learning would enable us to create unprecedented new levels of insight from our organizational data. If we then consider adding external data to our analysis such as our customer’s stock market performance or global events, the possibilities are endless.

There are several open-source Python libraries supported by continuum analytics that also look promising for addressing Big Data issues. Blaze looks to take the data wrangling capabilities of NumPy and Pandas and extend them to distributed computing environments. It also provides a standardized interface for connecting to a variety of data sources, including Spark, to simplify the process of data acquisition and wrangling. Bokeh, a data visualization library, offers the ability to create high performance visualizations and dashboards on top of large volumes of data as well as streaming data. This would allow us to avoid moving our data from the data lake to Tableau Server, further improving application performance and the speed in which data reaches the end user. Python also has PyMongo, an API for interacting with MongoDB databases – this would provide us with even more options and flexibility regarding how we handle unstructured or semi-structured data.

|  |
| --- |
| **4 DESCRIPTION OF DATASETS** |

Within our organization, we are challenged with combining data from disparate data sources that arrive at different velocities, in different formats, and have different lifecycles for how long the data is relevant. We access much of our financial data, such as bookings and revenue, by querying Oracle databases using SQL on a weekly basis when the data is refreshed. Some of the tables can be connected directly to Tableau, whereas more complex views require first loading the data into our data lake for processing before it can be pushed to Tableau. Several of our data sources, such as headcount, utilization, and project data, can only be accessed through SAP Business Objects universes. This requires us to create reports of the raw data that we deliver to our data lake on a weekly timeframe as Excel files that are then processed and pushed up to Tableau. Our organization’s sales pipeline data is located on a third party, cloud-hosted application. We leverage Python to make API calls to the application every five minutes so that we can provide near-real-time visibility into ongoing deals. The API provides data in nested JSON, which requires parsing and additional formatting before it too can be pushed to Tableau Server. Some data, such as the organization’s plan numbers for particular timeframes and regions, are generated locally as ad-hoc text files that are uploaded to Tableau Server to overlay the data that we have already pulled from other enterprise sources. Lastly, in order to manage the variety of files that require wrangling within our data lake, we create metadata files to help track the schema and formatting of the different data sources. This allows us to maintain an agile development methodology when approaching new data requests by creating generalized functions for handling the data wrangling that reference these metadata files.

|  |
| --- |
| **5 ANALYSIS AND RESULTS** |

To conduct analyses and communicate our findings to the broader organization, we have developed a two-pronged approach within our Big Data architecture using both Impala and Tableau. Given that our data is almost entirely structured in nature, we opted to use Impala over Pig or Hive due to the speed it offers by querying the cluster directly from SQL. In our work stream, we leverage Impala primarily for exploratory data analysis so as to get a better understanding of the columns and underlying data. Once we have analyzed the data and understand what questions it can help us to answer, we leverageTableau’s Cloudera data connector to connect to our cluster and create visualizations and dashboards based on the data. Based on the varying data lifecycles, we schedule different refresh schedule for each of the sources so that the dashboards always show the most recent data.

|  |
| --- |
| **6 CONCLUSION** |

With the implementation of the proposed architecture and workflow in the sections above, the organizational, data, and political challenges from the previous environment are a distant memory. By deploying both Cloudera’s CDH and leveraging the powerful tools of Impala and data visualization with Tableau, our vision of a self-service business intelligence solution has come to fruition. The primary benefit of this new architecture is having the right data at the right time to enable decision makers within the organization. This has allowed for incredible cost savings and new revenue opportunities across the business. We are deriving new insights and patterns with data like never before. For any data analyst who feel they are stuck in their current organization, have hope, as this is a blueprint for successfully implementing a data analytics strategy in an environment slow to change.

|  |
| --- |
| **REFERENCES** |

Apache Spark - Lightning-Fast Cluster Computing. (n.d.). Retrieved December 7, 2014, from https://spark.apache.org/

Blaze. (n.d.). Retrieved December 7, 2014, from http://blaze.pydata.org/docs/latest/index.html

Bokeh. (n.d.). Retrieved December 7, 2014, from http://bokeh.pydata.org/

Cloudera. (n.d.). Retrieved December 7, 2014, from http://www.cloudera.com/content/cloudera/en/products-and-services/cdh.html

Ghemawat, S., Gobioff, H., & Leung, S. (2003, October 1). Google Research Publication: The Google File System. Retrieved October 5, 2014, from http://research.google.com/archive/gfs.html

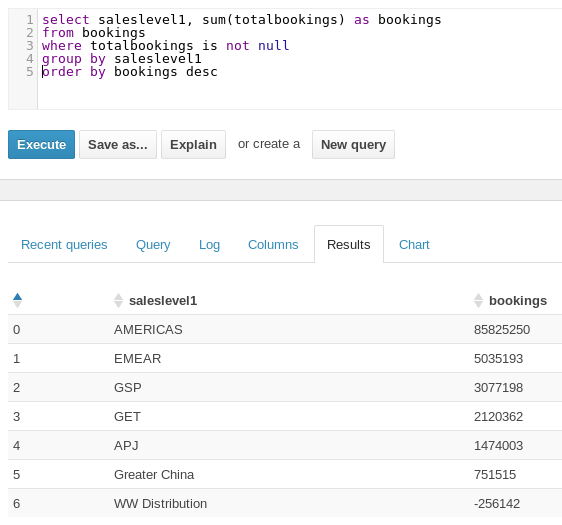
Grigorchuk, K. (n.d.). Comparing the top Hadoop distributions. Retrieved December 7, 2014, from http://www.networkworld.com/article/2369327/software/comparing-the-top-hadoop-distributions.html

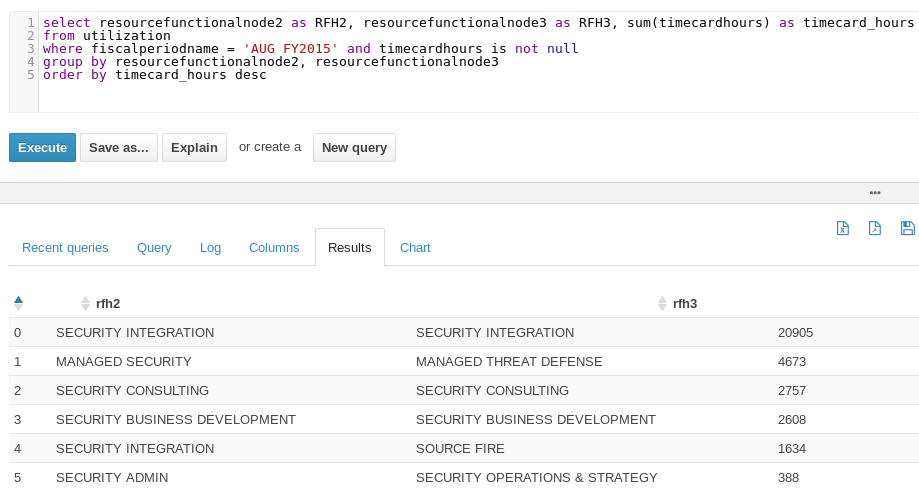
Hortonworks Data Platform delivers Enterprise Hadoop. (n.d.). Retrieved December 7, 2014, from http://hortonworks.com/hdp/

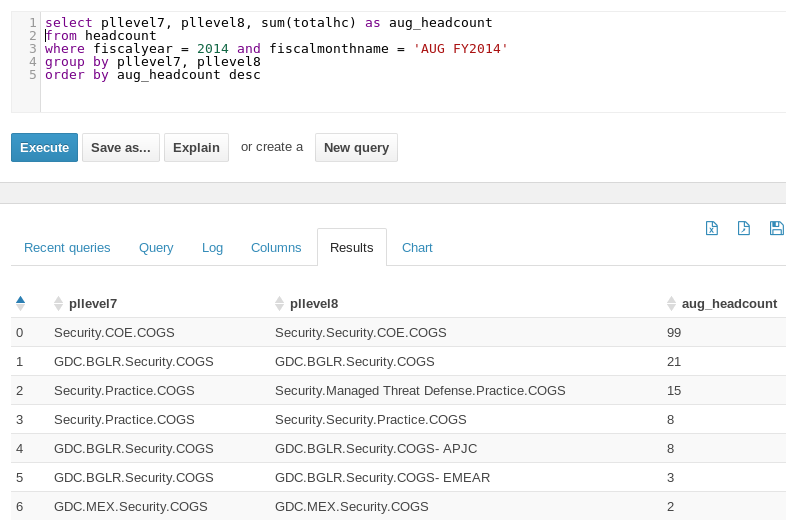
PyMongo. (n.d.). Retrieved December 7, 2014, from http://api.mongodb.org/python/current/

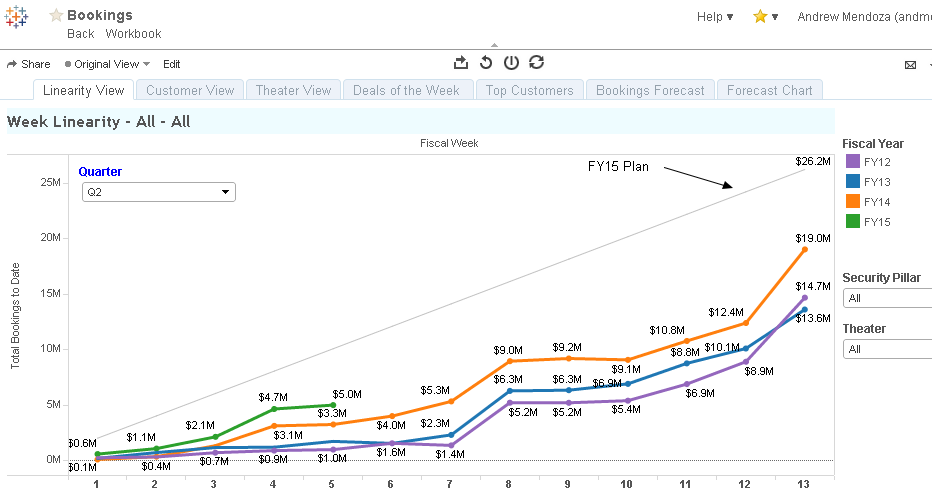
|  |
| --- |
| **APPENDIX** |

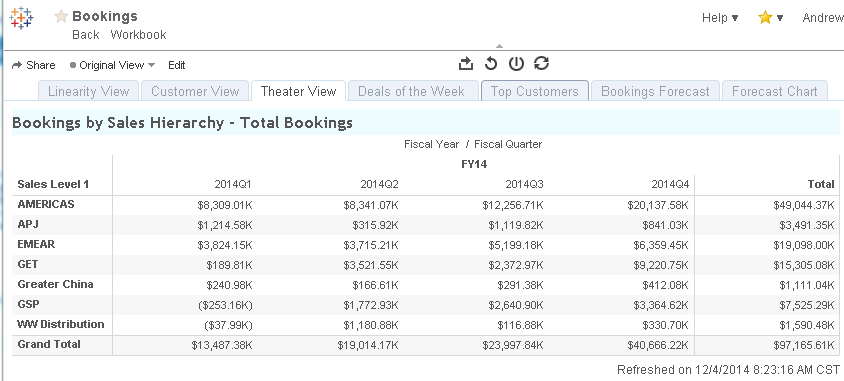
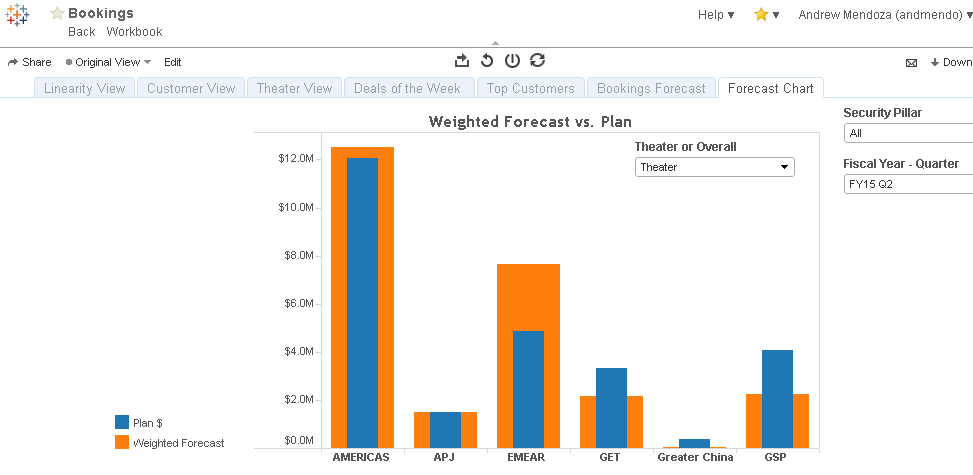
**Example Impala Queries**

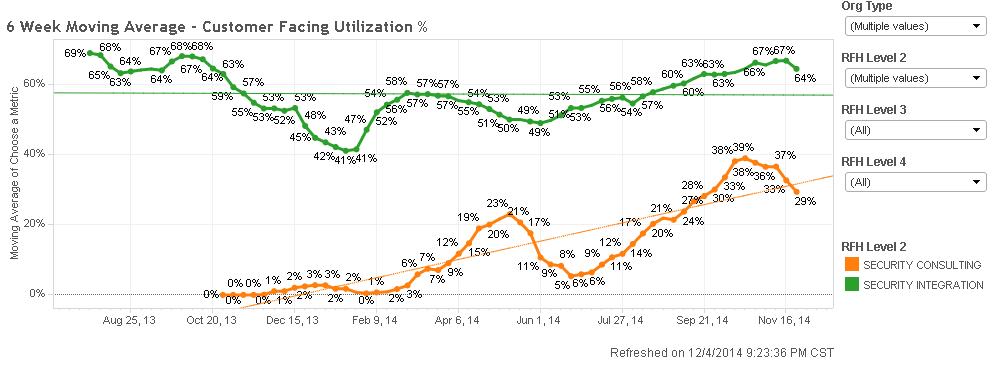


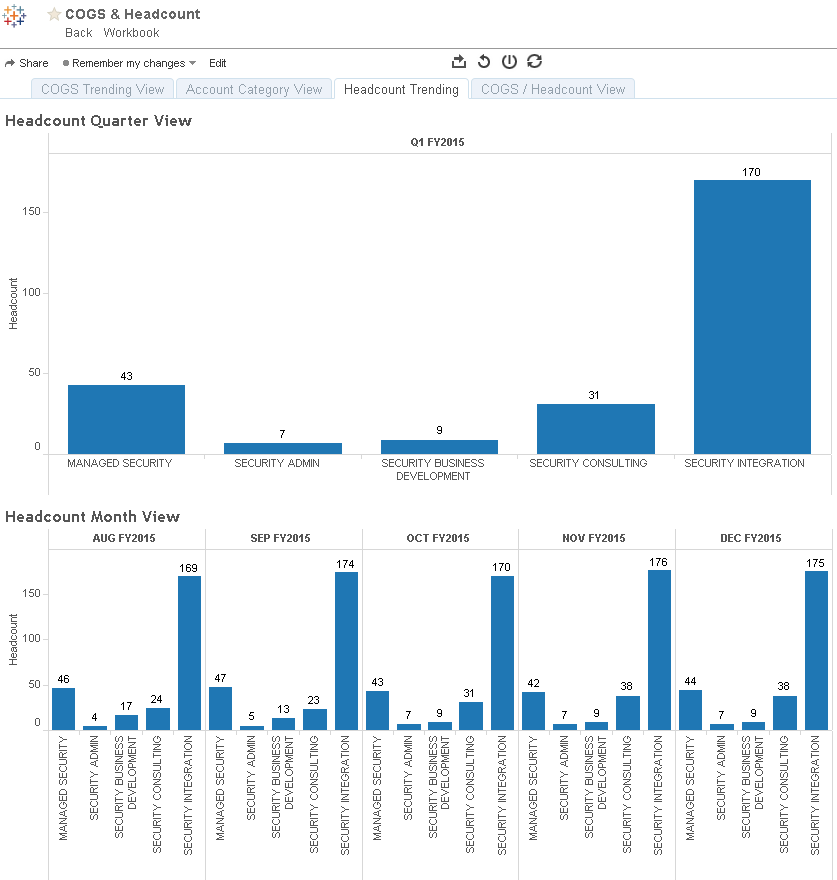
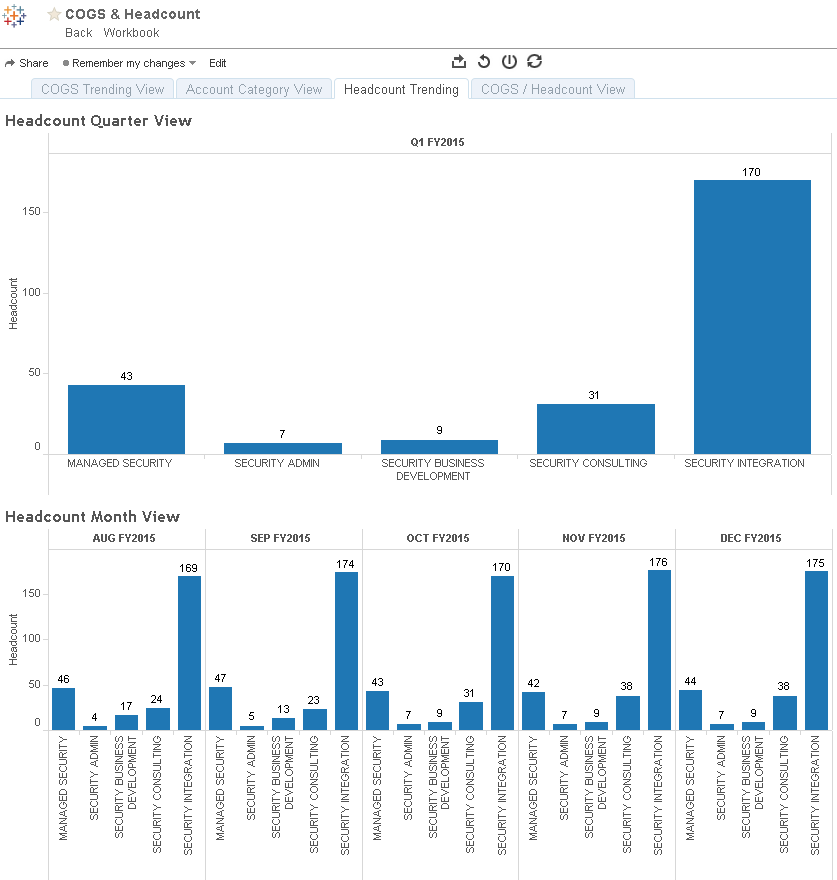




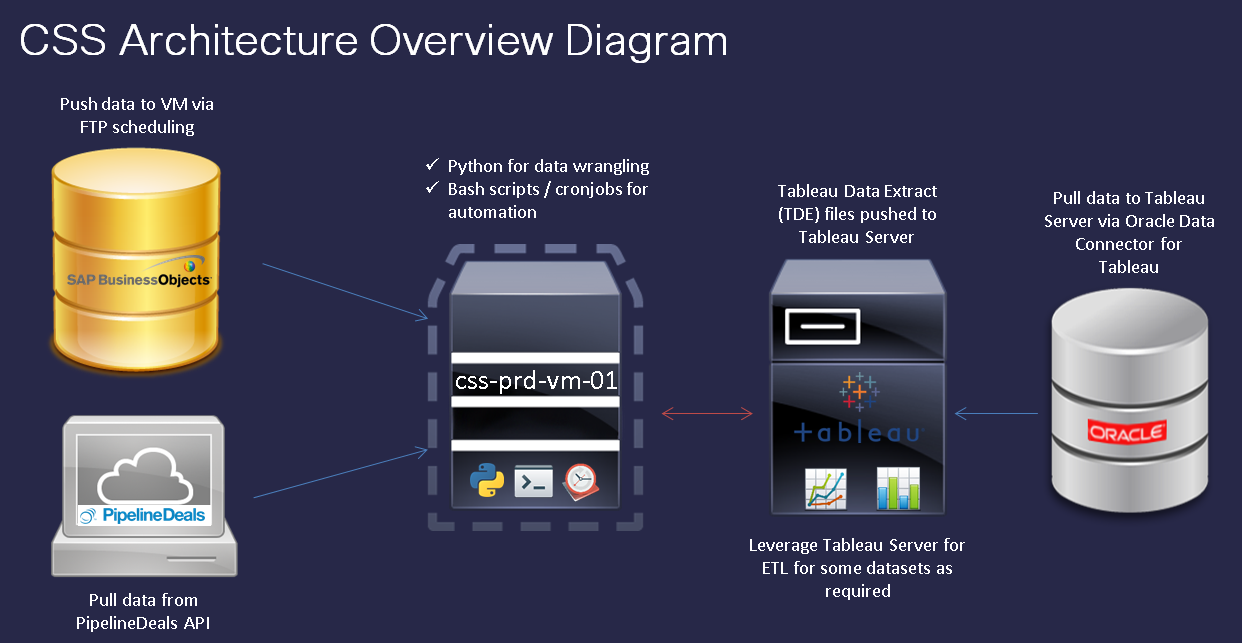
**Example Dashboards in Tableau**

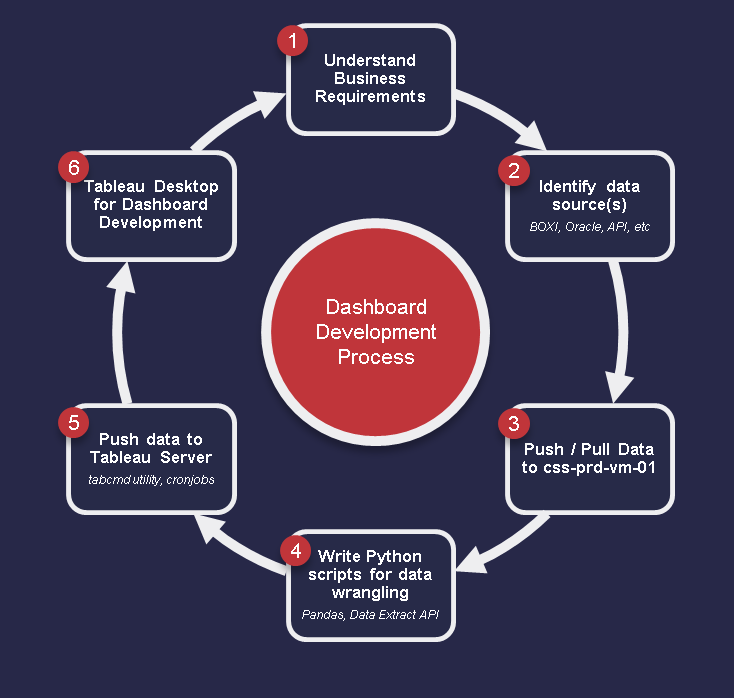






**Current Data Architecture**

**Dashboard Creation Workflow**



**Distribution of Work**

* Abstract – Nate
* Business Challenges / Introduction – Nate
* Big Data Overview – Nate
* Big Data Architecture – Andrew
* Description of Datasets – Andrew
* Analysis and Results – Andrew
* Conclusion – Nate
* Impala Scripts – Nate & Andrew
* Tableau Dashboards – Nate & Andrew
* Python Scripts – Nate & Andrew
* Architecture and Data Workflow Diagrams – Nate & Andrew







