Descrição Arquitetura Impulso.gov

Dado o desafio de montar uma arquitetura de um data lake e os requisitos propostos, decidi que a melhor opção seria manter um ambiente cloud cloud de uma BigTech do mercado para ajudar na nossa implementação. A escolhida foi a GCP (Google Cloud Plataform). Entre outras vantagens, possui uma gama de produtos serverless para dados, suporte a nível empresarial, comunidade forte, etc…

Já a arquitetura propriamente dita, optei por escolher a arquitetura denominada Data LakeHouse. Implementada inicialmente pela empresa Databricks, essa arquitetura entrega o que há de mais moderno na análise e ciência de dados. O data lakehouse inclusive já teve a “benção” de Bill Inmon, o pai do data warehouse.

Cabe aqui uma pequena introdução sobre a evolução das plataformas.

Desenvolvidos por volta de 40 anos atŕas, os DW suportaram decisões apoiadas em relatórios de BI. Conseguem manter dados históricos que foram apagados ou atualizados dos sistemas de origem. Foram construídos em cima de banco de dados relacional. Eles não possuem suporte a dados semi ou não estruturados, por isso fica muito difícil aplicar ciência de dados neles. Também é difícil a utilização de pipelines via streaming - que são os dados que chegam em tempo real permitindo uma análise do que está acontecendo no momento.

Já o Data Lake, permitiu a correção de muitos desses problemas porém uma má implementação resulta em desorganização. O silos que eram criados entre áreas impedia a democratização dos dados e as áreas continuam tendo que fazer joins, eliminar duplicidades aplicar modelos de negócio etc.. além disso, as empresas acabam pagando o dobro do custo de armazenamento para manter cópia dos dados no data lake + warehouse tradicional já que alguns dados acabam tendo que ficar no banco tradicional.

O lake house veio então para unir o melhor dos dois mundos.

Reduzir os custos operacionais e simplificar o processo de transformação e melhorar a governança. Eles usam dados em formato de arquivos abertos como Parquet e Orc que separam o schema propriamente dito, dos dados em si . É possível processar os mesmos dados sem ter que armazenar os dados em duplicidade, aplicar o ACID etc..

Então como funcionaria isso na prática?

Preferencialmente vamos utilizar o ELT.

Primeiro vamos ingerir o dado bruto para uma camada Raw ou Landing zone se necessário. O landing zone serve para dados não estruturados e dados que precisam passar por uma conversão de tipagem para serem ingeridos no bigquery.

É importante lembrar que o dado deve permanecer bruto nas camadas bronze e landing.

O bigquery é o astro. Com um formato que se assemelha a um banco de dados tabular, o bigquery é mais que isso e une o processamento e armazenamento em um único produto. No entanto, quando o dado vai ser processado, ele separa esse processamento do armazenamento, resultando assim em performance e economia. Com linguagem SQL, é possível utilizá-lo para realizar transformações básicas e até complexas como produzir um modelo de aprendizado de máquina.

O dado seria ingerido na camada de ingestão, por exemplo, pelo google dataproc que é um spark gerenciado e iria direto para armazenamento na camada bronze do bigquery.

Aí entraria novamente o dataproc em ação. Lendo o arquivo na camada bronze, fazendo as transformações necessárias para entregá-lo curado e enriquecido. As transformações do spark são feitas in memoria e geralmente só é transformado em tabela física, aquela tabela já totalmente trabalhada.

Mas caso haja necessidade, pode-se persistir a tabela intermediária numa camada Silver do Bigquery.

Por fim, temos os consumidores que irão trabalhar com os dados da camada gold. Porém, o acesso a camada bronze é dado também a cientistas e analistas que necessitarem.

Como alternativa, podemos trabalhar a mesma arquitetura mas com a plataforma do Databricks, que tem entre outras vantagens, o compartilhamento interativo em seu próprio workspace e uso de scripts em notebooks que são amplamente populares.

Por fim, temos um novo produto no mercado chamado Dataform. Ele é muito semelhante ao DBT, e entrou no escopo da google recentemente. Ele basicamente dá mais poderes ao bigquery documentando todas as transformações, orquestrando pipelines, modelando dados, controlando versões, ou seja, aplicando boas práticas de engenharia. Tudo isso com linguagem sql (uma variação de sintaxe chamada sqlx), permitindo até que analistas com baixo conhecimento em programação, possam entregar valor.

Temos a mesma arquitetura porém com o dataform tomando conta da orquestração e também auxiliando na governança.

O dado entra pela camada de ingestão direto para bronze do bigquery e, diferente do spark que usa seu próprio motor de processamento, o dataform usa o próprio motor do bigquery para processar as transformações. Isso resulta em maior performance.

As transformações intermediárias geralmente não são persistidas fisicamente no BQ, mas a tabela final sim. O dataform pode entregar também relatórios de data lineage onde algum analista, se tiver interesse, pode começar seu trabalho, a partir de uma tabela intermediária de outra transformação por exemplo. Vale a pena ler sobre ele

É uma ferramenta muito poderosa que demanda maior estudo da minha parte.

Aqui eu deixo minha conclusão. Eu iria de arquitetura lakehouse com spark na interface databricks pelo suporte, comunidade e tempo no mercado, mas sem tirar do radar o dataform que está amadurecendo e captando clientes.

Espero ter contribuído. Grato pela oportunidade.