# Contextualização

## O que é Big Data?

Big data é um conjunto de dados maior e mais complexo, especialmente de novas fontes de dados. Esses conjuntos de dados são tão volumosos que o software tradicional de processamento de dados simplesmente não consegue gerenciá-los. No entanto, esses grandes volumes de dados podem ser usados para resolver problemas de negócios que você não conseguiria resolver antes.

O volume de dados gerado atualmente é monstruoso, todos os dias bilhões de novas informações são geradas globalmente, pense em todos os Apps, Sistemas, TVs, Celulares, aparelhos com IoT (Internet of Things ou Internet das Coisas) que estão capturando, processando e armazenando novos dados. Cada clique que é dado em uma página ou aplicativo é automaticamente guardado para que possa ser Analisado.

## 5 V’s do Big Data

Atualmente, podemos dividir o Big Data em 5 V’s que formam a base para a implementação do conceito em qualquer empresa. São eles: **Volume**, **Velocidade**, **Variedade**, **Variabilidade** e **Vínculo**

### Volume

Volume é o ponto de partida para entender o BD pois a quantidade de dados importa. Com o big data, você terá que processar grandes volumes de dados não estruturados de baixa densidade. Podem ser dados de valor desconhecido, como feeds de dados do Twitter, fluxos de cliques em uma página web ou em um aplicativo para dispositivos móveis, ou ainda um equipamento habilitado para sensores. Para algumas empresas, isso pode utilizar dezenas de terabytes de dados. Para outras, podem ser centenas de petabytes.

O volume de informações influencia em dois pontos principais: **Análise** e **Armazenamento**.

Com a expansão diária de servidores, ficou muito mais fácil armazenar esses dados, principalmente com a compressão de arquivos e possibilidade de disponibilizá-lo na nuvem.

Já a análise se torna cada vez mais simples, principalmente com o uso de ferramentas próprias para o Big Data.

### Velocidade

Velocidade é a taxa mais rápida na qual os dados são recebidos e talvez administrados. Normalmente, a velocidade mais alta dos dados é transmitida diretamente para a memória, em vez de ser gravada no disco. Alguns produtos inteligentes habilitados para internet operam em tempo real ou quase em tempo real e exigem avaliação e ação em tempo real.

### Variedade

Dados aparecem com todas as formas, cores e tamanhos. Podem ser planilhas, dados estruturados, ou somente documentos de texto, vídeos e imagens. É preciso entender as variedades existentes e como cada uma deve ser analisada e armazenada.

Tipos de dados tradicionais foram estruturados e se adequam perfeitamente a um banco de dados relacional. Com o aumento de big data, os dados vêm em novos tipos de dados não estruturados. Tipos de dados não estruturados e semiestruturados, como texto, áudio e vídeo, exigem um pré-processamento adicional para obter significado e dar suporte a metadados.

### Variabilidade

A velocidade e variedade dos dados crescem constantemente, mas, adicionalmente, esses elementos mudam e não são necessariamente consistentes.

**Imagine as mídias sociais, como elas vão gerar informações de forma regular?**

Elas são totalmente dependentes das ações dos usuários e, mesmo parecendo, não ficamos 24 horas por dia conectados na web.

A ideia de variabilidade também pode ser aplicada a dados diários, sazonais e relativos a eventos específicos, o que gera um grande volume de informações e traz desafios no gerenciamento.

### Vínculo

Com um volume tão grande de dados, é muito provável que você perca o controle de tudo na hora que realmente precisar.

Um dos principais motivos disso é a dificuldade de conectar e transformar informações através de diferentes plataformas.

Portanto, é preciso conectar e correlacionar os elementos, criar hierarquias e ligações múltiplas para os dados, ou seja, criar vínculos entre eles.

## Por que Big Data foi criado?

O volume incrivelmente alto de informações gerados diariamente — e acumulado de outros anos — passou a ser visto como uma fonte de insights e não só um apanhado de dados.

Logo, foi preciso pensar em um modelo de análise que ajude as empresas a encontrar ideias valiosas no meio de tantos bytes.

O Big Data ajuda empresas a encontrarem oportunidades não só onde elas estão claras, mas em **correlações e cruzamento de dados complexos**, na decupagem de dados estruturados, não estruturados e multi-estruturados.

# Problematização

Dado o grande volume de processamento e demanda por velocidade, as arquiteturas para Big Data demandam soluções específicas que atendam as necessidades, de forma a entregar o melhor resultado aos usuários. Tendo isto em vista podemos conculir que as arquiteturas atuais não atendem a tal demanda, e as empresas precisarão renovar estas arquiteturas para atender a esta nova forma de processamento.

# Objetivos

Com base nesses dados, vamos entender quais as arquiteturas que melhor atendem aos processos do Big Data.

# Justificativa

# Embasamento Teórico e Prático

## Arquitetura para big data

Diagram

Description automatically generatedUma arquitetura de Big Data foi projetada para lidar com ingestão, processamento e análise de dados grandes ou complexos demais para sistemas de banco de dados tradicionais. O limite no qual as organizações ingressam no campo do Big Data é diferente, dependendo das capacidades dos usuários e de suas ferramentas. Para alguns, isso pode significar centenas de gigabytes de dados, enquanto para outros, centenas de terabytes. À medida que as ferramentas para o trabalho com conjuntos de Big Data evoluem, na mesma proporção evolui o significado de Big Data. Cada vez mais, esse termo se relaciona ao valor que é possível extrair dos conjuntos de dados por meio de análise avançada, em vez de estritamente o tamanho dos dados, embora nesses casos, eles tendam a ser muito grandes.

Ao longo dos anos, o cenário dos dados vem mudando. Houve uma mudança no que você pode fazer ou o que deve fazer, com os dados. O custo de armazenamento caiu drasticamente, enquanto os meios pelos quais os dados são coletados continuam aumentando. Alguns dados são recebidos a um ritmo rápido, constantemente exigindo sua coleta e observação. Outros dados são recebidos mais lentamente, mas em partes muito grandes, geralmente na forma de décadas de dados históricos. Talvez você esteja enfrentando um problema de análise avançada ou um problema que exija o aprendizado de máquina. Esses são desafios que as arquiteturas de Big Data buscam resolver.

Soluções de Big Data normalmente envolvem um ou mais dos seguintes tipos de carga de trabalho:

Processamento em lote de fontes Big Data em repouso.

Processamento em tempo real de Big Data em movimento.

Exploração interativa de Big Data.

Análise preditiva e machine learning.

A maioria das arquiteturas de Big Data inclui alguns ou todos os seguintes componentes:

Fontes de dados: todas as soluções de Big Data começam com uma ou mais fontes de dados. Os exemplos incluem:

Armazenamentos de dados de aplicativo, como bancos de dados relacionais.

Arquivos estáticos produzidos por aplicativos, como arquivos de log do servidor Web.

Fontes de dados em tempo real, como dispositivos IoT.

Armazenamento de dados: dados de operações de processamento em lote normalmente são armazenados em um repositório de arquivos distribuído que pode conter amplos volumes de arquivos grandes em vários formatos. Esse tipo de repositório geralmente é chamado data lake. As opções para implementar esse armazenamento incluem contêineres de blobs ou Azure Data Lake Store no Armazenamento do Azure.

Processamento em lote: como os conjuntos de dados são muito grandes, geralmente uma solução de Big Data deve processar arquivos de dados usando trabalhos de lote de execução longa para filtrar, agregar e preparar os dados para análise. Normalmente, esses trabalhos envolvem ler arquivos de origem, processá-los e gravar a saída para novos arquivos. Opções incluem executar trabalhos de U-SQL no Azure Data Lake Analytics, usar trabalhos Hive, Pig ou de Mapear/Reduzir personalizados em um cluster HDInsight Hadoop ou usar programas de Java, Scala ou Python em um cluster HDInsight Spark.

Ingestão de mensagens em tempo real: se a solução inclui fontes em tempo real, a arquitetura deve incluir uma maneira de capturar e armazenar mensagens em tempo real para processamento de fluxo. Isso pode ser um armazenamento de dados simples, em que as mensagens de entrada são removidas para uma pasta para processamento. No entanto, muitas soluções precisam de um repositório de ingestão de mensagens para atuar como buffer de mensagens e dar suporte a processamento de expansão, entrega confiável e outras semânticas de enfileiramento de mensagem. Opções incluem Hubs de Eventos do Azure, Hubs de IoT do Azure e Kafka.

Processamento de fluxo: depois de capturar mensagens em tempo real, a solução deve processá-las filtrando, agregando e preparando os dados para análise. Os dados de fluxo processados são gravados em um coletor de saída. O Azure Stream Analytics oferece um serviço de processamento de fluxo gerenciado baseado em consultas SQL em execução perpétua que operam em fluxos não associados. Você também pode usar tecnologias de streaming Apache de software livre, como Storm e Spark Streaming em um cluster HDInsight.

Armazenamento de dados analíticos: muitas soluções de Big Data preparam dados para análise e então veiculam os dados processados em um formato estruturado que pode ser consultado usando ferramentas analíticas. O armazenamento de dados analíticos usado para atender a essas consultas pode ser um data warehouse relacional estilo Kimball, como visto na maioria das soluções de BI (business intelligence) tradicionais. Como alternativa, os dados podem ser apresentados por meio de uma tecnologia NoSQL de baixa latência, como HBase ou um banco de dados Hive interativo que oferece uma abstração de metadados sobre arquivos de dados no armazenamento de dados distribuído. O Azure Synapse Analytics fornece um serviço gerenciado para armazenamento de dados em larga escala baseado em nuvem. O HDInsight dá suporte a Hive interativo, HBase e Spark SQL, que também pode ser usado para veicular dados para análise.

Análise e relatório: a meta da maioria das soluções de Big Data é gerar insights sobre os dados por meio de análise e relatórios. Para capacitar os usuários a analisar os dados, a arquitetura pode incluir uma camada de modelagem de dados, como um cubo OLAP multidimensional ou um modelo de dados tabular no Azure Analysis Services. Também pode dar suporte a business intelligence de autoatendimento, usando as tecnologias de modelagem e visualização do Microsoft Power BI ou do Microsoft Excel. Análise e relatórios também podem assumir a forma de exploração de dados interativos por cientistas de dados ou analistas de dados. Para esses cenários, muitos serviços do Azure dão suporte a blocos de anotações analíticos, como Jupyter, permitindo que esses usuários aproveitem suas habilidades existentes com Python ou R. Para exploração de dados em larga escala, você pode usar o Microsoft R Server, seja no modo autônomo ou com Spark.

Orquestração: a maioria das soluções de Big Data consiste em operações de processamento de dados repetidos, encapsuladas em fluxos de trabalho, que transformam dados de origem, movem dados entre várias origens e coletores, carregam os dados processados em um armazenamento de dados analíticos ou efetuam o push dos resultados diretamente para um relatório ou painel. Para automatizar esses fluxos de trabalho, você pode usar uma tecnologia de orquestração, como Azure Data Factory ou Apache Oozie e Sqoop.

## Desafios

Complexidade. Soluções de Big Data podem ser extremamente complexas, com vários componentes para lidar com a ingestão de dados de várias fontes de dados. Pode ser um desafio criar, testar e solucionar problemas de processos de Big Data. Além disso, pode haver um grande número de definições de configuração em vários sistemas que devem ser usados para otimizar o desempenho.

Conjunto de qualificações. Muitas tecnologias de Big Data são altamente especializadas e usam frameworks e idiomas que não são típicos de arquiteturas de aplicativo mais gerais. Por outro lado, as tecnologias de Big Data estão gerando novas APIs que se baseiam em linguagens mais estabelecidas. Por exemplo, a linguagem U-SQL no Azure Data Lake Analytics baseia-se em uma combinação de Transact-SQL e C#. Da mesma forma, APIs com base em SQL estão disponíveis para Hive, HBase e Spark.

Maturidade da tecnologia. Muitas das tecnologias usadas em Big Data estão em evolução. Embora tecnologias Hadoop centrais, como Hive e Pig, tenham se estabilizado, tecnologias emergentes, como Spark, apresentam grandes alterações e aprimoramentos a cada nova versão. Serviços gerenciados, como Azure Data Lake Analytics e Azure Data Factory, são relativamente jovens em comparação a outros serviços do Azure e provavelmente evoluirão ao longo do tempo.

Segurança. Soluções de Big Data normalmente se baseiam em armazenar todos os dados estáticos em um data lake centralizado. Proteger o acesso a esses dados pode ser desafiador, especialmente quando os dados devem ser ingeridos e consumidos por vários aplicativos e plataformas.

## Práticas recomendadas

Aproveitar o paralelismo. A maioria das tecnologias de processamento de Big Data distribui a carga de trabalho em várias unidades de processamento. Isso exige que os arquivos de dados estáticos sejam criados e armazenados em um formato divisível. Sistemas de arquivos distribuídos, como HDFS, podem otimizar o desempenho de leitura e gravação, e o processamento real é executado por vários nós de cluster em paralelo, o que reduz o tempo de trabalho geral.

Dados de partição. O processamento em lote geralmente ocorre em um agendamento recorrente – por exemplo, semanal ou mensal. Arquivos de dados de partição e estruturas de dados como tabelas, com base em períodos de temporais que correspondem à agenda de processamento. Isso simplifica a ingestão de dados e o agendamento de trabalho, além de tornar mais fácil solucionar problemas de falhas. Além disso, o particionamento de tabelas usadas em consultas Hive, U-SQL ou SQL pode melhorar significativamente o desempenho da consulta.

Aplicar semântica de esquema na leitura. Usar um data lake permite combinar o armazenamento de arquivos em vários formatos, sejam estruturados, semiestruturados ou não estruturados. Use semântica de esquema na leitura, que projeta um esquema nos dados quando os dados estão sendo processados, não quando estão armazenados. Isso integra flexibilidade à solução e evita gargalos durante a ingestão de dados causados pela verificação de tipo e a validação de dados.

Processar dados no local. Soluções de BI tradicionais geralmente usam um processo ETL (extração, transformação e carregamento) para mover dados para um data warehouse. Com maiores volumes de dados e uma maior variedade de formatos, soluções de Big Data geralmente usam variações de ETL, como TEL (transformação, extração e carregamento). Com essa abordagem, os dados são processados no armazenamento de dados distribuídos, transformando-os na estrutura necessária, antes de mover os dados transformados para um armazenamento de dados analíticos.

Equilibrar custos de tempo e utilização. Para trabalhos de processamento em lotes, é importante considerar dois fatores: custo unitário de nós de computação e custo por minuto de usar esses nós para concluir o trabalho. Por exemplo, um trabalho em lotes pode levar oito horas com quatro nós de cluster. No entanto, pode ser que o trabalho use todos os quatro nós somente durante as primeiras duas horas, sendo apenas dois nós necessários depois disso. Nesse caso, executar todo o trabalho em dois nós aumentaria o tempo total do trabalho, mas não o duplicaria, de modo que o custo total seria menor. Em alguns cenários de negócios, um tempo de processamento mais longo pode ser preferível ao custo mais alto do uso de recursos de cluster subutilizados.

Separar os recursos de cluster. Ao implantar clusters HDInsight, você normalmente alcança um melhor desempenho provisionando recursos de cluster separados para cada tipo de carga de trabalho. Por exemplo, embora clusters do Spark incluam Hive, se você precisar executar amplo processamento com Hive e Spark, deverá considerar implantar clusters Spark e Hadoop dedicados separados. Da mesma forma, se você estiver usando HBase e Storm para processamento de fluxo de baixa latência e Hive para processamento em lotes, considere clusters separados para Storm, HBase e Hadoop.

Orquestrar a ingestão de dados. Em alguns casos, aplicativos de negócios existentes podem gravar arquivos de dados para processamento em lote diretamente em contêineres do Azure Storage Blob, em que podem ser consumidos pelo HDInsight ou pelo Azure Data Lake Analytics. No entanto, você geralmente precisará orquestrar a ingestão de dados de fontes de dados externas ou locais para o data lake. Use um fluxo de trabalho de orquestração ou um pipeline, como aqueles compatíveis com Azure Data Factory ou Oozie, para fazer isso de maneira previsível e gerenciável centralmente.

Limpar dados confidenciais cedo. O fluxo de trabalho de ingestão de dados deve remover dados confidenciais no início do processo para evitar armazená-los no data lake.

# Análise dos resultados

A partir dos resultados obtidos com esta pesquisa identificamos que em uma arquitetura de Big Data é necessário priorizar valocidade e capacidade de processamento, aplicando as boas práticas para que seja possível entregar alto desempenho aos usuários.

# Considerações finais

Com base nos dados apresentados a cima podemos entender melhor as necessidades de uma arquitetura de Big Data, a qual demanda algumas especificidades devido a alta demanda por processamento e velocidade afim de entregar um resultado coerente ao usuário, dessa forma auxiliando em diversos setores em que é utilizado, como por exemplo em decisões gerenciais sendo possível, através do processamento de dados em massa, obter informações valiosas para as empresas, possibilitando diversos avanços.

# Referências