

ASPECTOS GERAIS

• Captura + gerenciamento + análise de dados que vão além dos dados estruturados típicos (Consultados por bancos de dados relacionais)

• Lida com dados **não estruturados** e em **grande volume**
(Vídeos, imagens, textos,....)

APLICAÇÕES

- Desenvolvimento de mercado
- Inovação
- Desenvolvimento de produtos e serviços
- Eficiência operacional
- Previsões de demanda
- Detecção de fraudes
- Gerenciamento de riscos

big data

CAI MUITO!

AS CARACTERÍSTICAS

- 3 v's { Volume (Grande quantidade de dados)
Velocidade (Quão rápido os dados são produzidos e tratados)
Variedade (Os dados vêm em vários tipos de formatos)
- 7 v's { Veracidade (Necessidade de lidar com dados imprecisos e incertos)
Visibilidade (É a relevância dos dados; devem estar disponíveis e visíveis)
Valor (As informações geradas devem produzir valor para a organização)
Variabilidade e complexidade
(Variação nas taxas) (Recepção de dados de diferentes fontes: devem ser integrados)



IBM:

- 3v's + veracidade + valor
+ governança (Pode ser necessário redefinir/ modificar os processos de negócios da organização)
+ pessoas (É necessário ter pessoas com aptidões específicas)

CLASSIFICAÇÕES

• Tipos de análises:

- Em tempo real
- Por batelada (Agrupados para análise posterior)

• Metodologias de processamento:

- Preditiva
- Consulta *AD HOC*
- Analítica
- Relatórios

• Frequência e tamanho dos dados

- Sob demanda
- Feed contínua, em tempo real
- Série temporal

• Tipo de dados:

- Transacionais
- Históricos
- Principais...

• Formato de conteúdo:

- Estruturados
- Não estruturados
- Barato
- de Ponta

• Fonte de dados:

- Web
- Mídia social
- Gerados por máquinas
- Gerados por pessoas

• Consumidores de dados:

- Processos de negócios
- Usuários corporativos
- Aplicativos Corporativos
- Pessoas em várias funções
- Parte dos fluxos do processo
- Outros repositórios (de dados ou aplicativos)

PADRÕES ATÔMICOS E COMPOSTOS

- Auxiliam a definir os parâmetros quando da adoção de uma solução de *big data*

- Padrões **atômicos**: ajudam a identificar a forma que os dados são:

- Consumidos
- Armazenados
- Processados
- Acessados

• São eles:

- Padrões de consumo
- Padrões de processamento
- Padrões de acesso
- Padrões de armazenamento

- Padrões **compostos**: formados por padrões atômicos (Classificados de acordo com o escopo da solução de *big data*)

NOSQL (Not only SQL)

- Alternativa aos bancos de dados relacionais tradicionais, podendo acomodar uma grande variedade de **modelos de dados** (Chave-valor, documento, colunar, de grafos...)

• Especialmente úteis para se trabalhar com grandes conjuntos de dados distribuídos

CARACTERÍSTICAS COMUNS

- Não utilizam o modelo relacional
- Boa execução em cluster
- Código aberto
- Não têm um esquema definido
- Criados para suportar propriedades da web do século XXI

big data

- **Hardware**: (Entender suas limitações ajuda na escolha da solução)
- Barato
- de Ponta

MODELOS DE DADOS (No NOSQL)

- Forma pela qual **percebemos** e **manipulamos** os dados

MODELO DE DADOS CHAVE-VALOR

- Orientação agregada
(Para trabalhar com dados com uma estrutura mais complexa que um conjunto de Tuplas)
- Cada **chave** está associada a um **único valor**
- Tipo de NOSQL **mais simples** (Fácil compreensão e implementação)
- Operações** disponíveis são mais simples (Não permite consultas mais complexas)
- Permite a visualização dos bancos de dados como uma **Tabela Hash**
- Exemplos:** *Dynamo (Amazon), Redis, Riak, Genie DB.*

MODELO DE DADOS DE DOCUMENTO

- Orientação agregada
- Permite consultas e recuperações **parciais**
- Documento:**
 - Objeto com **identificador único** e um conjunto de **campos** (*Strings, listas...*)
 - Não possui esquema**
O banco de dados não pode atuar na estrutura do documento para otimizar o armazenamento ou recuperação de partes do agregado
- Exemplos:** *Couch DB, Mongo DB*

big data

MODELO COLUNAR

- Orientação agregada
- O agregado é dividido em **famílias de colunas**
Há alguma estrutura, mas permite ao banco aproveitar a estrutura para melhorar sua acessibilidade
- Os dados são indexados por uma **trilha** (Linha, coluna) e *timestap*
- Operações de leitura/escrita são **atômicas**
- Exemplos:** *Cassandra (facebook), HBASE*

MODELO DE GRAFOS

- Registros **pequenos** com interconexões **complexas**
Ou estruturas que generalizem a noção de grafos
- Representa os dados como **grafos dirigidos** (generalizem a noção de grafos)
- Interessante quando a **topologia** dos dados é tão (ou mais) importante que os **dados** propriamente ditos.
- Grafos = Nós (vértices)
 - + Relacionamentos (arestas)
 - + Propriedades (Atributos dos nós e relacionamentos)
- Exemplos:** *NEO4J, Infinite Graph, Inforgrid, HypergraphDB*

big data



MAPREDUCE

- = Padrão que permite que **operações computacionais** sejam relacionadas em **clusters**
- Desenhado para processar **grandes volumes** de dados em **paralelo** (Dividindo o trabalho em tarefas independentes)
- Introduzido pelo **google**

ACID

✗

BASE

- = Atomicidade, Consistência, Isolamento e Durabilidade
- Princípio de design de banco de dados **relacional**
- Usa **controles de simultaneidade** pessimistas, para garantir a consistência

⟳ A operação só é finalizada se houver a certeza de que a consistência foi realizada no momento

- = *Basically Available, Soft State, Eventual Consistency*

- É otimista: aceita que a consistência está em **estado de fluxo** (Não precisa ser efetivada em tempo real)
- A disponibilidade é garantida tolerando **fallas parciais**, sem que o sistema todo falhe
- **Soft State** = O banco de dados pode estar em um estado **inconsistente** quando os dados são lidos

⟳ (Os resultados podem mudar se solicitados novamente)

Permite que os dados sirvam muitos clientes sem qualquer latência (embora servindo resultados inconsistentes)