

Big Data

Agenda

- Big Data
- Problemas com as Tecnologias Atuais
- NoSQL
- Teorema CAP
- Propriedades ACID x BASE
- Modelos de dados NoSQL
- Persistência Poliglota
- Map-Reduce
- Exemplos de Aplicações com NoSQL

O que é Big Data?

- É o termo aplicado a dados que não podem ser processados ou analisados usando processos e ferramentas tradicionais (Zikopoulos et al., 2012);
- Big Data é o termo aplicado a dados muito grandes, muito rápidos ou muito “duros” para serem processados pelas ferramentas existentes (Madden, 2012).

3

Desafios do Big Data

- Como Big Data pode gerar “Big Insight”?
- As empresas têm acesso a informações valiosas, mas não sabem como extrair valor delas;
- Isto acontece porque estas informações estão em formato bruto;
 - De forma semiestruturada ou não estruturada.
- Desta forma, as empresas nem sabem se é vantajoso mantê-las;
 - Ou mesmo se elas estão aptas a mantê-las para esse propósito.

4

Desafios do Big Data



- Uma pesquisa da IBM identificou que metade dos gestores percebem que não têm acesso ao conhecimento (*insight*) que eles precisam para fazer seu trabalho (Zikopoulos et al., 2012).

5

Como surgiu o Big Data?

- A era do Big Data está em plena força porque o mundo está mudando através da:
 - Instrumentalização:
 - GPSs, Acelerômetros e sensores em geral.
 - Conectividade:
 - LANs, WANs, Wi-Fi, 3G, 4G, etc.
 - Inteligência:
 - Smartphones, Tablets, notebooks, etc.
- O Big Data está associado ao volume, variedade, e velocidade dos dados (os 3 Vs).

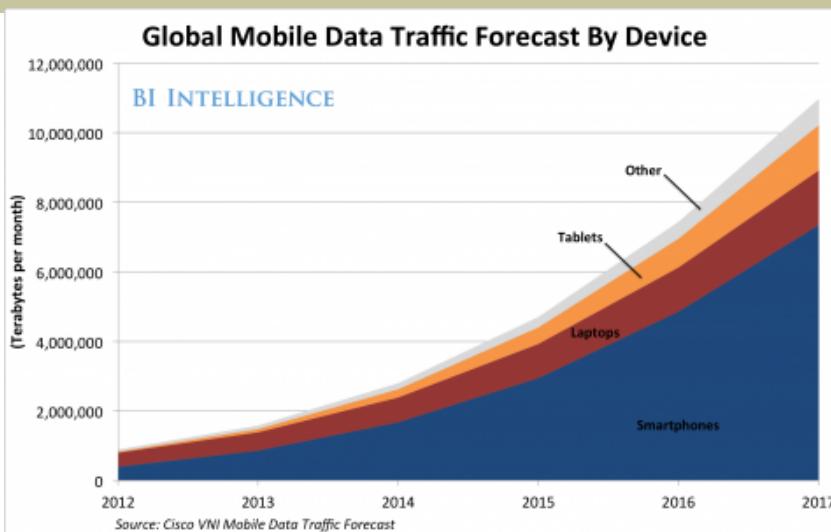
6

Os 3 Vs – Volume

- Crescimento do volume de dados armazenados por ano:
 - Em 2000 => 800.000 petabytes (PB)
 - Em 2010 => mais 1 zettabyte (ZB)
 - Em 2014 => estima-se 7ZB
 - Em 2020 => estima-se que alcance 35 (ZB)
- Exemplos:
 - Twitter: Mais de 7TB por dia
 - Facebook : Mais de 10TB por dia

7

Crescimento do Volume de Dados Gerados por Tipo de Dispositivo



Fonte: Business Insider (Jul. 2013) 8

Os 3 Vs – Volume

Name (Symbol)	Bedeutung [G 1]
Kilobyte (kB) [G 2]	10^3 Byte = 1.000 Byte
Megabyte (MB)	10^6 Byte = 1.000.000 Byte
Gigabyte (GB)	10^9 Byte = 1.000.000.000 Byte
Terabyte (TB)	10^{12} Byte = 1.000.000.000.000 Byte
Petabyte (PB)	10^{15} Byte = 1.000.000.000.000.000 Byte
Exabyte (EB)	10^{18} Byte = 1.000.000.000.000.000.000 Byte
Zettabyte (ZB)	10^{21} Byte = 1.000.000.000.000.000.000.000 Byte
Yottabyte (YB)	10^{24} Byte = 1.000.000.000.000.000.000.000.000 Byte

- 1 zettabyte = 1 sextilião de bytes

9

Os 3 Vs – Variedade

- Apenas 20% dos dados são relacionais e podem ser nitidamente formatados dentro de um esquema fixo (Zikopoulos et al., 2012);
- Entretanto, 80% dos dados gerados no mundo são semiestruturados ou não estruturados;
- Exemplos:
 - Páginas web, fóruns de mídia social, e-mail, documentos, sensores, etc.

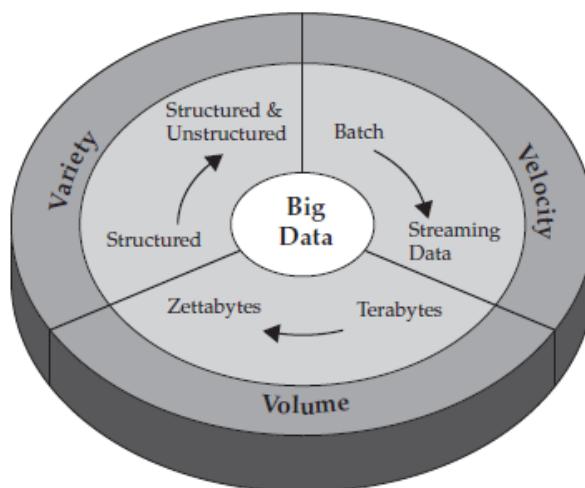
10

Os 3 Vs – Velocidade

- Tipicamente, a velocidade está relacionada a quão rápido o dado chega e é armazenado;
- Entretanto, a velocidade está relacionada também a quão rápido o dado é analisado;
- Ao invés de velocidade de crescimento, temos que pensar na velocidade de fluxo dos dados;
- As empresas precisam estar aptas a analisar seus dados em tempo real.
 - Por exemplo, isso possibilita a identificação de oportunidades antes dos seus concorrentes.

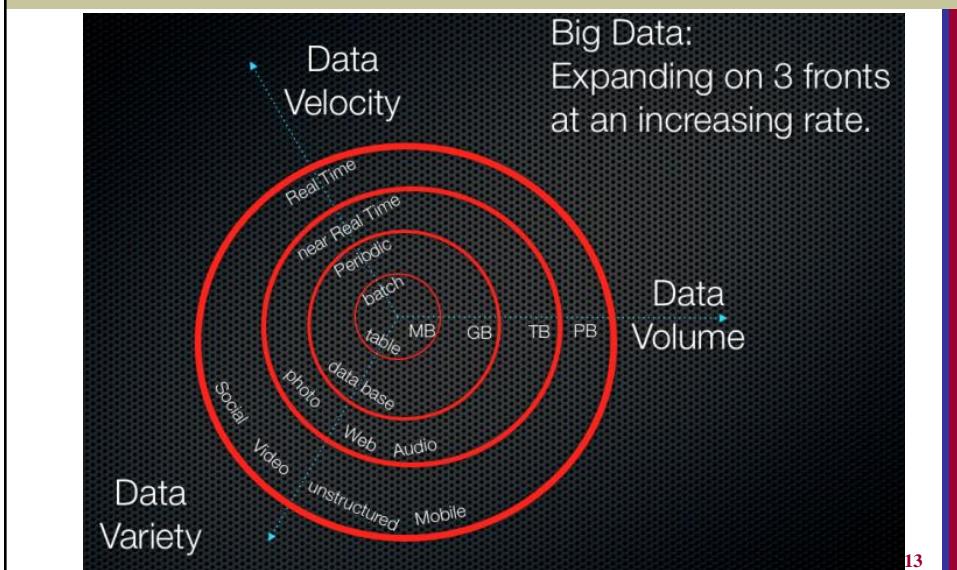
11

Os 3 Vs



Zikopoulos et al. (2012)

Os 3 Vs



Problemas com as Tecnologias Atuais

- Sistemas de banco de dados relacionais estão atingindo seu limite (Marz e Warren, 2012);
- As técnicas de gerenciamento de dados tradicionais não conseguiram suportar o Big Data;
- Dilema:
 - Enquanto o volume de dados das empresas está aumentando, o percentual de processamento e análise deles está diminuindo (Zikopoulos et al., 2012);

14

Problemas com as Tecnologias Atuais

- Segundo Sadalage e Fowler (2013):
 - Falta de flexibilidade para tratar dados semiestruturados e não-estruturados;
 - Diferença entre as estruturas de dados na memória e o modelo relacional;
 - Chamada incompatibilidade de impedância;
 - Não foram projetados para trabalhar em cluster.
 - O cluster possibilita a escalabilidade horizontal.

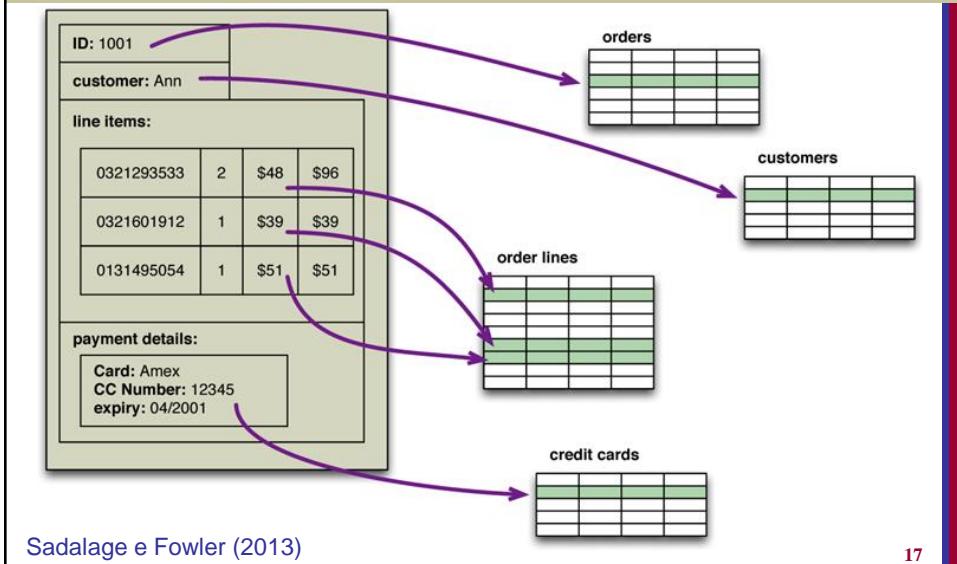
15

Problemas com as Tecnologias Atuais

- A Incompatibilidade de Impedância:
 - Diferença de representação entre os dados em memória e os armazenados em disco (Sadalege e Fowler, 2013);
 - Isso implica na necessidade de mapeamentos;
 - O modelo relacional não permite agregação de dados:
 - Uma relação é composta por atributos simples e monovalorados;
 - Não é possível armazenar e recuperar estruturas aninhadas.

16

Incompatibilidade de Impedância



Sadalage e Fowler (2013)

17

NoSQL

- Para enfrentar os desafios do Big Data, uma nova geração de tecnologias surgiu;
- Estas novas tecnologias foram agrupadas sob o termo "NoSQL".
- Este termo foi cunhado para um encontro em 2009 em San Francisco organizado por Johan Oskarson;
 - Boa hashtag no Twitter.
- NoSQL é um movimento, não uma tecnologia;
- Não há órgão regulamentador.

18

NoSQL

- Não há consenso com relação ao significado:

1. No SQL



2. Not Only SQL



3. NoSQL



19

NoSQL

- Os pioneiros no NoSQL foram:

- Google (2006):

- Modelo família de colunas (BigTable);
 - Framework Map-Reduce.

- Amazon (2007):

- Modelo chave-valor (Dynamo).

- A comunidade open source:

- Hadoop, MongoDB, CouchDB, HBase, Cassandra, Riak, Redis, Neo4J e inúmeros outros projetos.

20

NoSQL

- Algumas destas novas tecnologias são mais complexos do que as tradicionais, enquanto outras são mais simples;
- Com elas, bancos de dados podem ser escalados para conjuntos volumosos de dados;
 - Suportam clusterização/distribuição.
- O uso delas de forma eficaz requer profundo conhecimento;
- Elas não são soluções *one-size-fits-all*.

21

Mudança de Perspectiva no armazenamento de dados

- De:
 - Banco de dados de Integração:
 - Múltiplos aplicativos armazenam seus dados em um banco de dados comum.
- Para:
 - Banco de dados de aplicativo:
 - É acessado por um único aplicativo;
 - Torna mais fácil a manutenção do esquema;
 - A integração fica a cargo da interface do aplicativo.
 - Comumente feita através de serviços web.

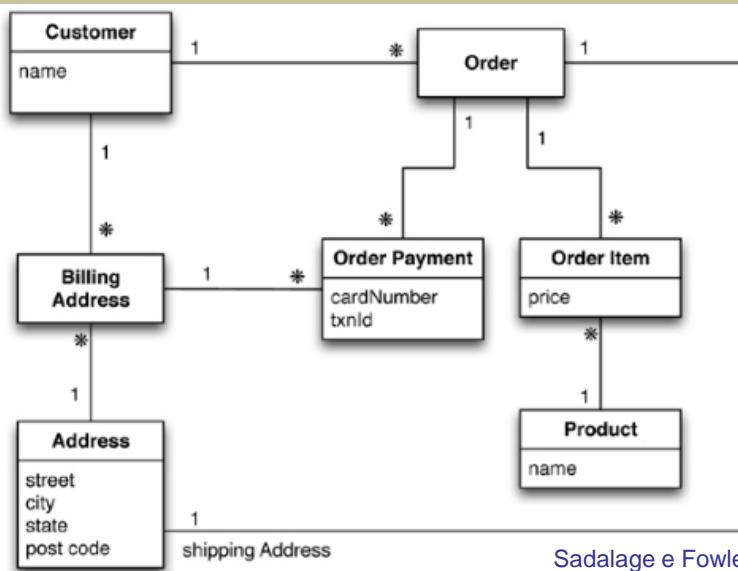
22

Características dos Modelos NoSQL

- **Modelo de Dados Agregado:**
 - Permite representar dados em estruturas mais complexas do que um conjunto de tuplas;
 - Agregado é um conjunto de objetos relacionados que podem ser tratados como uma unidade;
 - Agregados realizam operações atômicas;
 - Agregado = transação.
 - O agregado é a unidade natural para replicação e fragmentação, que facilita a execução em cluster;
 - Resolve o problema de incompatibilidade de impedância.

23

Agregação: Modelo Relacional

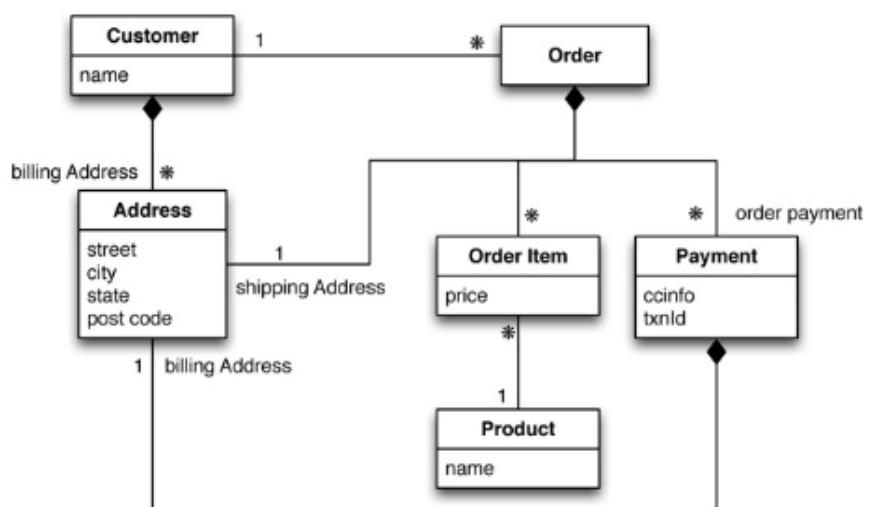


Agregação: Modelo Relacional

Customer		Orders	
Product		BillingAddress	
OrderItem		Address	
Id	Name	Id	CustomerId
1	Martin	99	1
Id	Name	Id	CustomerId
27	NOSQL Distilled	55	77
Id	OrderId	Id	AddressId
100	99	1	77
Id	OrderId	Id	City
100	99	77	Chicago
Id	OrderId	CardNumber	BillingAddressId
33	99	1000-1000	55
			abelif879rft

Sadalage e Fowler (2013)

Agregação: Modelo Agregado



Sadalage e Fowler (2013)

Agregação: Modelo Agregado

```
// in customers
{
  "id":1,
  "name":"Martin",
  "billingAddress":[{"city":"Chicago"}]
}

// in orders
{
  "id":99,
  "customerId":1,
  "orderItems":[
    {
      "productId":27,
      "price": 32.45,
      "productName": "NoSQL Distilled"
    }
  ],
  "shippingAddress":[{"city":"Chicago"}]
  "orderPayment": [
    {
      "ccinfo":"1000-1000-1000-1000",
      "txnId":"abelif879rft",
      "billingAddress": {"city": "Chicago"}
    }
  ],
}
}
```

Sadalage e Fowler (2013)

Características dos Modelos NoSQL

- Sem esquema:
 - Não exige a existência de um esquema definido antes do armazenamento de algum dado;
 - Facilita o armazenamento de dados não uniformes;
 - Permite a evolução dos dados a medida que se conhece mais sobre eles;
 - Inclusão e/ou exclusão de dados é automática
 - Entretanto, existe um esquema implícito.
 - O esquema está no código do aplicativo que acessa os dados.

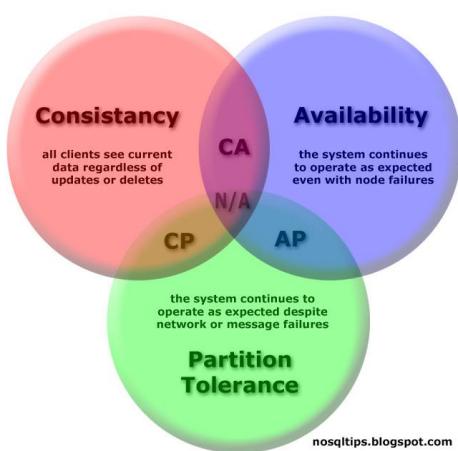
Características dos Modelos NoSQL

- Feitos para executar em cluster;
 - Viabiliza o processamento de grandes volumes de dados, através da escalabilidade horizontal.
- Grande parte de código aberto;
 - Existem várias iniciativas de sucesso feitas pela comunidade de software livre.
- Abrem uma gama de opções para armazenamento;
 - Possuem vários modelos.
- Não possuem uma linguagem padrão;
 - Não possuem uma linguagem tipo SQL.

29

Teorema CAP

- Proposto por Brewer em 2000;
- Diz que pode-se ter **no máximo duas** destas propriedades para um sistema de armazenamento de dados.
 - Consistência (C)
 - Disponibilidade (A)
 - Tolerância a partição da rede (P).



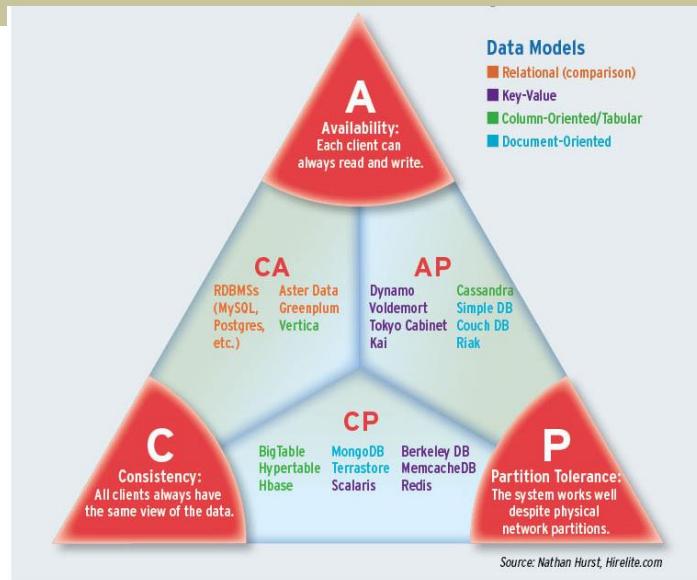
30

Teorema CAP

- Bancos de dados centralizados são CA;
- Pode-se ter um cluster CA, porém se houver uma partição na rede todos os nós saem do ar;
- Clusters devem ser tolerantes a partição;
 - Então, deve-se balancear a consistência e a disponibilidade.
- Se for necessário manter a consistência, então deve-se limitar a disponibilidade;
- Há casos em que se pode lidar com respostas inconsistentes;
 - Exemplo: Sites de compras e reservas de hotéis.

31

Teorema CAP



32

Propriedades ACID x BASE*

ACID

- A consistência é prioridade;
- O isolamento é forte;
- Apenas valores confirmados são aceitos;
- Abordagem pessimista;
- Mais complexo e demorado;
 - 2PC (two-phase commit)

BASE

- A disponibilidade é prioridade;
- A isolamento é fraca;
- Valores desatualizados são permitidos;
- Abordagem otimista;
- Mais simples e rápido;
 - Versionamento - multi-version concurrency control (MVCC)

Brewer (2000)

*Basically Available, Soft-State, Eventual Consistency³³

Modelos de Dados NoSQL – Documento

- São orientados a agregados chamados de documentos;
 - Os documentos podem ser do tipo XML, JSON, BSON, etc.
 - Não confundir com doc. de texto, planilhas, etc.
- Cada documento tem um ID para identificá-lo;
- A estrutura do documento é visível:
 - Pode-se definir as estruturas e tipos permitidos;
 - Pode-se fazer consultas baseadas nos campos do documento e recuperar parte do agregado.

34

Modelos de Dados NoSQL – Documento

- Comparação da terminologia Relacional x Documento

Oracle	MongoDB
database instance	MongoDB instance
schema	database
table	collection
row	document
rowid	_id
join	DBRef

Sadalage e Fowler (2013)

Modelo de Dados NoSQL – Documento

JSON

```
{
  "firstname": "Pramod",
  "citiesvisited": [
    "Chicago",
    "London",
    "Pune",
    "Bangalore"
  ],
  "addresses": [
    {
      "state": "AK",
      "city": "DILLINGHAM",
      "type": "R"
    },
    {
      "state": "MH",
      "city": "PUNE",
      "type": "R"
    }
  ],
  "lastcity": "Chicago"
}
```

XML

```
<person>
  <firstname>Pramod</firstname>
  <citiesvisited>
    <cityvisited>Chicago</cityvisited>
    <cityvisited>London</cityvisited>
    <cityvisited>Pune</cityvisited>
    <cityvisited>Bangalore</cityvisited>
  </citiesvisited>
  <addresses>
    <address>
      <state>AK</state>
      <city>DILLINGHAM</city>
      <type>R</type>
    </address>
    <address>
      <state>MH</state>
      <city>PUNE</city>
      <type>R</type>
    </address>
  </addresses>
  <lastcity>Chicago</lastcity>
</person>
```

Modelo de Dados NoSQL – Chave-Valor

- Também são orientados a agregados;
- A estrutura do agregado não é visível;
 - Armazena um bloco de bytes.
- Base simples baseada em Tabela Hash.
 - É uma tabela de duas colunas: Key, Value
 - Key: String, Value: Blob
- O agregado só pode ser recuperado procurando por sua chave;
- Pode-se adicionar metadados aos agregados para indexação e busca.

37

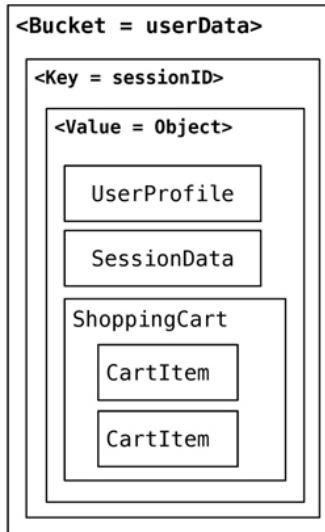
Modelo de Dados NoSQL – Chave-Valor

- Comparação da terminologia Relacional x Chave-valor

Oracle	Riak
database instance	Riak cluster
table	bucket
row	key-value
rowid	key

Sadalage e Fowler (2013)
38

Modelo de Dados NoSQL – Chave-Valor



Bucket: coleção de chave-valor no Riak

Sadalage e Fowler (2013)³⁹

Modelo de Dados NoSQL – Família de Colunas

- Possui uma estrutura agregada em dois níveis:
 - 1º nível: Chave identificadora da linha (agregado)
 - 2º nível: Chave identificadora da coluna.
- As colunas são organizadas em famílias de colunas;
 - Cada coluna faz parte de uma única família;
 - Assume-se que os dados de uma família de coluna serão acessados juntos.
- Uma linha é formada por um conjunto de famílias de colunas.

40

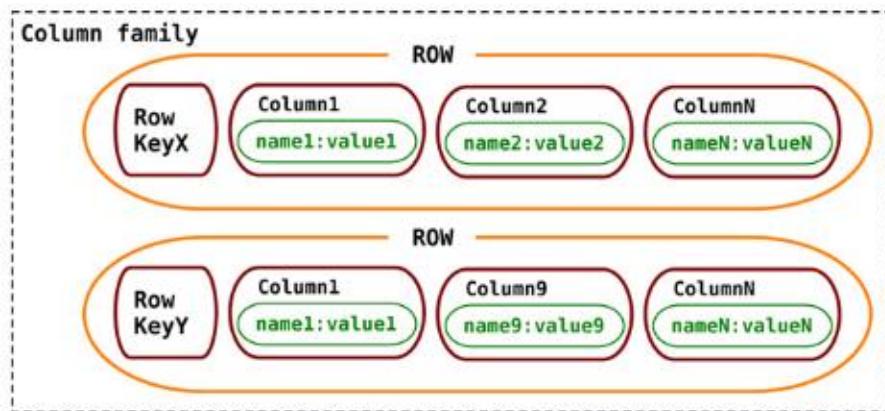
Modelo de Dados NoSQL – Família de Colunas

- Comparação da terminologia Relacional x Família de Coluna

RDBMS	Cassandra
database instance	cluster
database	keyspace
table	column family
row	row
column (same for all rows)	column (can be different per row)

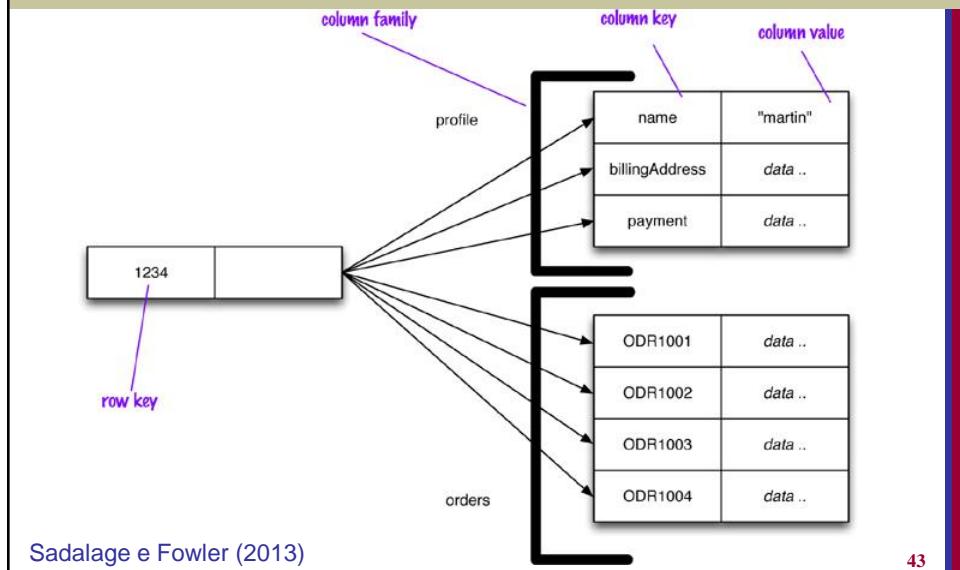
Sadalage e Fowler (2013) 41

Modelo de Dados NoSQL – Família de Colunas



Sadalage e Fowler (2013) 42

Modelo de Dados NoSQL – Família de Colunas

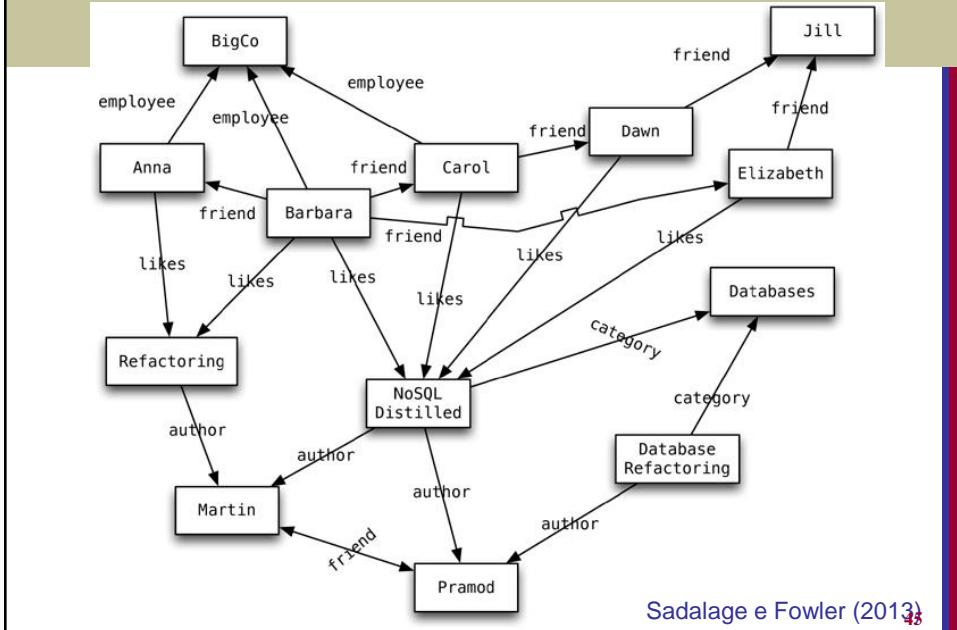


Modelo de Dados NoSQL – Grafos

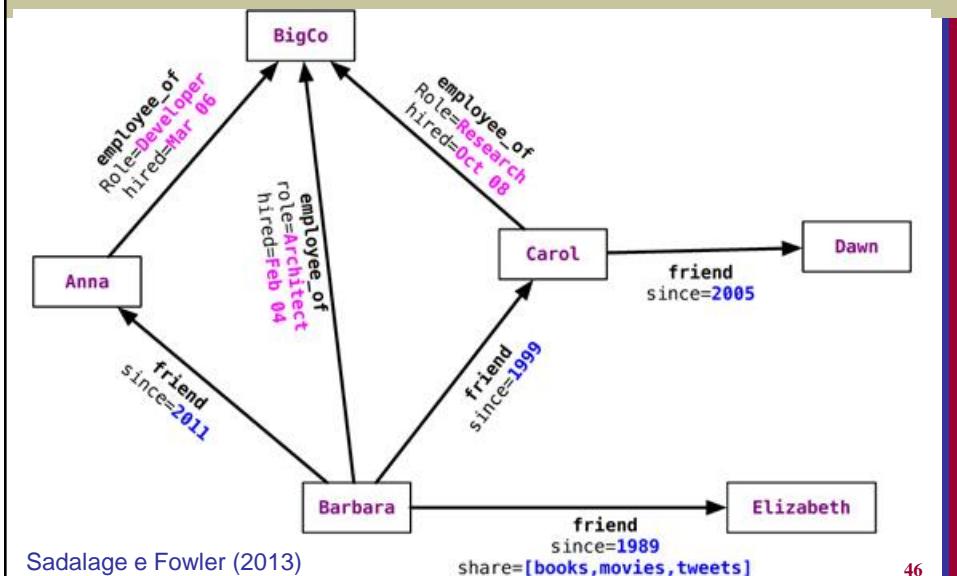
- Modelo simples, composto por nós conectados por arestas;
 - Nós e arestas podem ter propriedades.
- É utilizado para registros pequenos com interconexões complexas;
 - Diferentemente dos agregados que são usados para registros grandes com conexões simples;
- Modelo onde o foco são os relacionamentos.
 - Podem ser implementados em BDs relacionais, mas o desempenho piora rapidamente a medida que os *joins* aumentam.

44

Modelo de Dados NoSQL – Grafos



Modelo de Dados NoSQL – Grafos



Modelo de Dados NoSQL – Grafos

Ex.: Encontrar amigos de amigos

Table 1 Execution times for multiple join queries using MySQL database engine on data set of 1,000 users

Depth	Execution time (seconds) 1,000 users	Records returned
2	0.028	~900
3	0.213	~999
4	10.273	~999
5	92,613.150	~999

Table 2 Execution times for graph traversal using Neo4j on data set of 1,000 users

Depth	Execution time (seconds) 1,000 users	Records returned
2	0.04	~900
3	0.06	~999
4	0.07	~999
5	0.07	~999

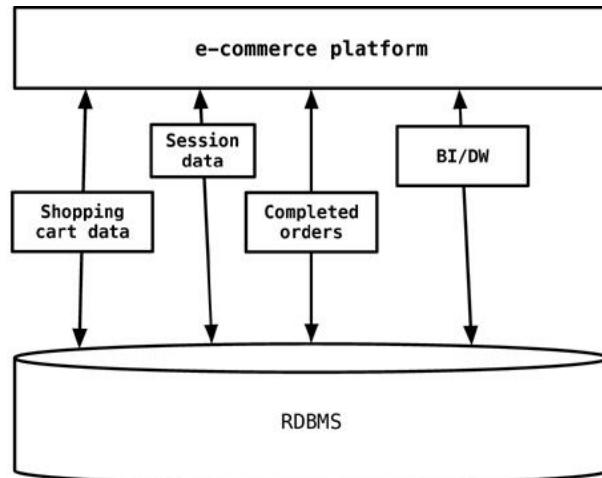
Partner et al. (2013)
47

Persistência Poliglota

- Diferentes bancos de dados são projetados para resolver diferentes problemas;
- Utilizar um único banco de dados para todas as necessidades resulta em baixo desempenho;
 - Ex.: Dados transacionais, informações de sessões, relacionamentos em grafo.
- Cada mecanismo de banco de dados é projetado para executar muito bem certas operações em certas estruturas de dados.

Persistência Poliglota

- Persistência tradicional



49

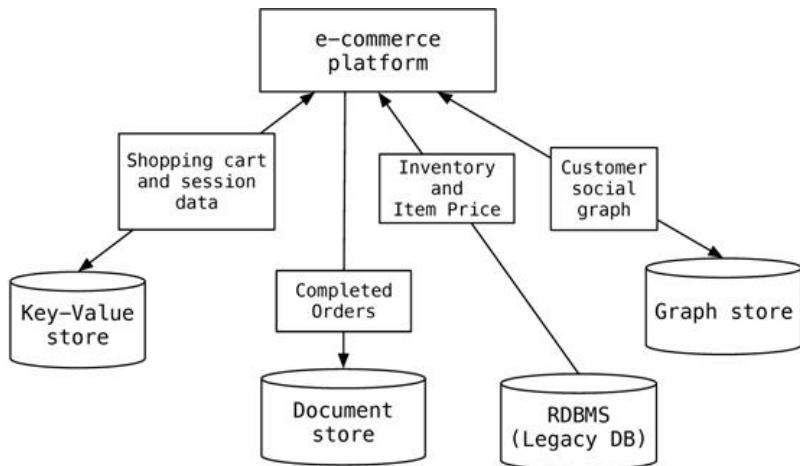
Persistência Poliglota

- Derivado do termo programação poliglota:
 - Os aplicativos devem ser escritos com a mistura de linguagens necessárias para resolvê-lo.
- O termo persistência poliglota é utilizado para definir a utilização híbrida de diferentes modelos de dados como solução de persistência;
- Quebra o paradigma de armazenamento *one-size-fits-all*.
 - Os DBAs precisam adquirir novas habilidades.

50

Persistência Poliglota

- Persistência poliglota



51

Map-Reduce

- Surgiu como forma de organizar/distribuir o processamento entre as máquinas de um cluster;
 - Objetivo: tentar processar os dados localmente, diminuindo a quantidade transferida pela rede.
- É composta por duas funções (Dean e Ghemawat, 2004):
 - *Map*: processa um par chave/valor para gerar um conjunto de pares chave/valor intermediários;
 - *Reduce*: combina todos os valores intermediários associados com a mesma chave para gerar o resultado;

52

Map-Reduce

- Assinatura das funções e seus retornos:

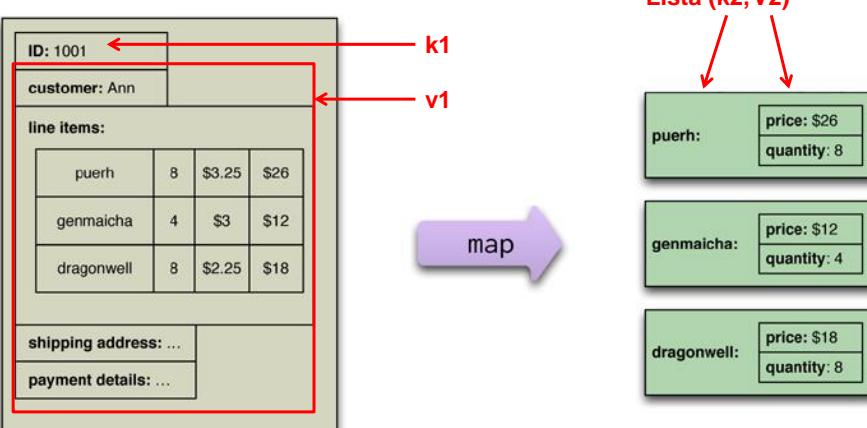
Assinatura da função	Retorno
map(k_1 , v_1)	lista(k_2 , v_2)
reduce(k_2 , lista(v_2))	lista(v_2)

- O framework map-reduce organiza as tarefas de mapeamento executadas em cada nó e repassa os resultados para o reduce.

53

Map-Reduce

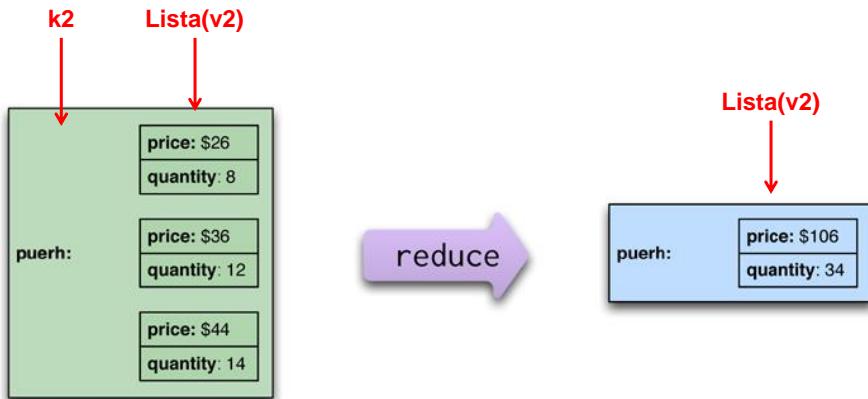
- Função Map:



54

Map-Reduce

- Função Reduce:



55

Exemplos de Aplicações com NoSQL: Construção de Mapas Rodoviários

- Dados coletados por GPS:



Google earth

56

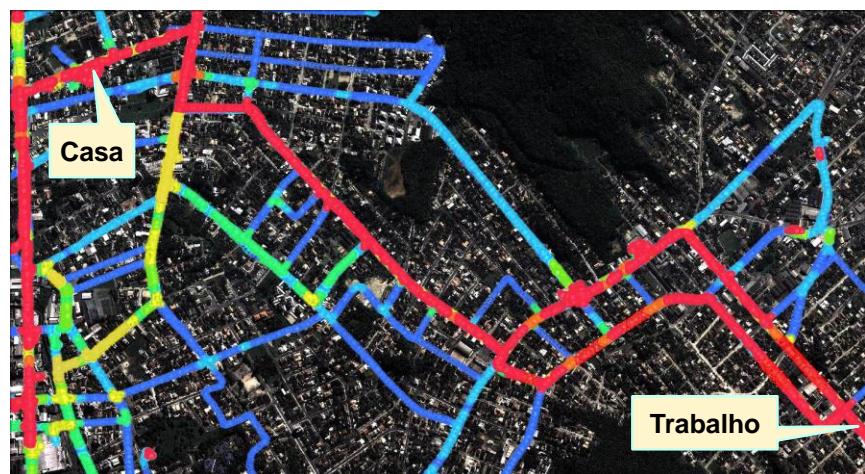
Exemplos de Aplicações com NoSQL: Construção de Mapas Rodoviários

- Mapa rodoviário:



Exemplos de Aplicações com NoSQL: Mineração de perfis de usuários

- Em que lugar a pessoa trabalha e/ou estuda?



Mestrado em Computação Aplicada

- Mestrado acadêmico;
- Gratuito e com possibilidade de bolsa de estudos;
- Oferecido na UDESC-Joinville;
- Inscrição semestral;
 - É possível fazer disciplinas como aluno especial.
- Linhas de Pesquisa:
 - Engenharia de Software;
 - Computação Gráfica;
 - Sistemas Computacionais.
- www.joinville.udesc.br/ppgca

59

Referências

- Brewer, E. Towards Robust Distributed Systems, 2000.
- Dean, J., e Ghemawat, S. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1), 2008.
- Madden, S. From Database to Big Data. *IEEE Internet Computing*, 2012.
- Marz, N., Warren, J. Big Data – Principles and Best Practices of Scalable Realtime Data Systems, 2012.
- Partner, J., et al. Neo4j in Action. Manning Publications Company, 2013.
- Sadalage, P., Fowler, M. NoSQL Essencial. Novatec, 2013.
- Zikopoulos, P. et al. Understanding Big Data – Analytics for Enterprise Class Hadoop and Streaming Data, 2012.

60

Big Data

From Big Data to Big Insights

Prof. Fabiano Baldo