

Bancos de Dados em “Clouds”



Bancos de Dados em “Clouds”

Erik Williams Zirke Osta
Rafael Brundo Uriarte

Agenda

- Introdução;
- Fundamentos;
- Estudo comparativo das Ferramentas;
- Conclusões e Trabalhos Futuros.

Agenda

- **Introdução;**
- Fundamentos;
- Estudo comparativo das Ferramentas;
- Conclusões e Trabalhos Futuros.

Motivação

- Cloud Computing:
 - Mudança no modelo de fornecimento de serviços de TI;
 - Alvo de muitos estudos e pesquisas;
 - Oportunidade única de progredir e ajudar a quebrar alguns dos atuais paradigmas da computação;
 - Impacto no modo de desenvolver software e gerir recursos computacionais.

Breve Definição

- *Cloud Computing é o fornecimento de recursos computacionais para terceiros por fornecedores independentes e distribuídos geograficamente. Um cliente contrata recursos computacionais ao invés de usar recursos locais e não precisa saber onde os recursos estão instalados.*
- Podem ser fornecidos deste modo:
 - Infra-estrutura;
 - Software;
 - Plataformas de desenvolvimento;
 - Bancos de dados.

Historia de Cloud Computing

- Origens do nome;



- Utility Computing:

- *“As redes de computadores ainda estão na sua infância, mas, assim que estiverem crescendo e ficando mais sofisticadas, provavelmente veremos a expansão do utility computing, que como energia elétrica e telefone, serão serviços fornecidos nas casas e escritórios pelo país.”*



Leonard Kleinrock

Definição de Cloud Computing

- *“Cloud é um tipo de sistema paralelo e distribuídos que consiste de uma coleção de computadores inter-conectados e virtualizados que são dinamicamente “instanciados” e são apresentados como um ou mais recursos computacionais baseados em acordos de nível de serviços pré-estabelecidos o provedor do serviço e o clientes.”*

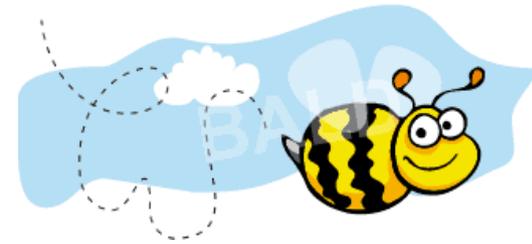
Rajkumar Buyya

Agenda

- Introdução;
- **Fundamentos;**
- Estudo comparativo das Ferramentas;
- Conclusões e Trabalhos Futuros.

Tendências

- *Buzzword;*



- Momento do Cloud Computing:
 - Barateamento do hardware;
 - Recursos Ociosos;
 - Amadurecimento das tecnologias base.

Vantagens

- Flexibilidade/Escalabilidade;
- Tecnologia de Ponta;
- Informação em qualquer lugar;
- ***Custos!!!***



Outras Definições

- SaaS(Software as a Service);
- IaaS(Infrastructure as a Service);
- PaaS(Platform as a Service);
- DbaaS(Database as a Service).

Fundamentos do DbaaS

- Database As a Service:
 - Informação de qualquer lugar;
 - Escalabilidade;
 - Performance;
 - Uso sob demanda.

Bases Relacionais

- Funcionam bem em poucos nós;
- Dificuldade de cuidar dos princípios básicos do ACID em bases distribuídas;
- JOINS e outras operações são caras e não escaláveis;
- É necessário uma solução para que a base de dados possua bom desempenho e seja altamente escalável.

Valor e chave(Key/Value Databases)

- Sacrifica muitos recursos pela escalabilidade;
- Não existe relações explícitas definidas entre domínios;
- Domínios sem definir esquemas;
- Orientada a atributos.

Carro	
Key	Atributos
1	Fabricante: Ford Modelo: Ecosport Cor: Prata Ano: 2008
2	Fabricante: Ford Modelo: Ecosport Cor: Amarelo Ano: 2009 Transmissão: Auto

Key/Value Databases

- Vantagens:
 - Foco na Escalabilidade;
 - Mais próxima do problema dos programadores ;
- Desvantagens:
 - Integridade tem que ser garantida pela aplicação;
 - Domínios são atrelados as aplicações;
 - Compatibilidade;



Agenda

- Introdução;
- Fundamentos;
- **Estudo comparativo das Ferramentas;**
- Conclusões e Trabalhos Futuros.

Ferramentas Seleccionadas

- AmazonSimpleDB;
- Drizzle;
- FathomDB;
- Google BigTable;
- Vertica.



AmazonSimpleDB

- SGBD Orientado a chave (Key/Value):
 - A query só pode ser executada em até 5 segundos;
 - Os atributos são todos Strings:
 - Strings de até 1024 bytes;
 - 256 atributos por Domínio;
 - Não garante a consistência dos dados;
 - É fornecido através de serviços.



Drizzle

- SGBD Semi-Relacional
 - Para melhorar a escalabilidade não possui:
 - stored procedures
 - query cache
 - prepared statements
 - Views
 - Triggers
 - Grants
 - O fonte e instaláveis estão disponíveis.



FathomDB

- SGBD Relacional:
 - Relacional;
 - Utiliza padrões do MySQL;
 - Grande preocupação com monitoramento e backup dos dados;
 - Usa o serviço de Cloud da Amazon(E2C);
 - É fornecido como serviço.

Google BigTable

- SGBD Orientado a chave (Key/Value):
 - É utilizado no Youtube, Google Maps e outras apps;
 - 500mb gratuitos para os primeiros 500 registros;
 - Custos até 10 vezes menores que da Amazon SimpleDB;
 - É fornecido como serviço.



Vertica

- SGBD Analítico:
 - Grande preocupação com performance;
 - Pode ser usada na Cloud da Amazon(E2C);
 - É fornecido como serviço.



Ferramenta	Forma de Distribuição	Modelo de Dados	Esquema Explícito	APIs	Administração
FathomDB	Serviço	Relacional	SIM	Java	Replicação, Backup, Monitoramento
Vertica	Serviço	Analítico	SIM	ODBC	Compressão
AmazonSimpleDB	Serviço	Estruturada (Attribute-oriented key/value)	Esquema dinâmico e flexível, mas não obrigatório	Java, SOAP, REST, C#, Python	-
Google Bigtable	Serviço	Estruturado (Attribute-oriented key/value)	Esquema dinâmico e flexível, mas não obrigatório	SOAP, REST, Java, C#, Perl, PHP, VB.NET	Compressão, Caching, Replicação
Drizzle	Instalável	Semi-Relacional	SIM	JDBC, PHP, Python,	Facilidade de instalação e gerenciamento

Análise das Ferramentas

- Ferramentas analisadas ainda não estão maduras;
- Novas ferramentas surgindo no mercado;
- Banco de Dados Relacionais parece não ser tão adequado para Bases em uma Cloud.

Agenda

- Introdução;
- Fundamentos;
- Estudo comparativo das Ferramentas;
- **Conclusões e Trabalhos Futuros.**

Áreas de Pesquisa

- Pesquisas privadas avançadas;
- Pesquisas acadêmicas começam a dar sinais de melhora;
- Laboratório LRG/UFSC.

Ferramentas para Cloud Computing

- Eucalyptus;



- CloudSim;



- VCL.



Conclusão e Trabalhos Futuros

- DBaaS ainda está em seus primeiros passos;
- É um mercado que tende a crescer rapidamente;
- Ferramentas ainda não estão maduras;
- Falta compatibilidade entre as Clouds;
- Se o projeto precisa de grande escalabilidade, DbaaS é ideal.

Bibliografia

- Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities. Rajkumar Buyya;
- An Internet vision: the invisible global infrastructure. Leonard KleinRock;
- Bigtable: a distributed storage system for structured data;
- Websites das respectivas ferramentas;
- Outros, ver artigos;

Dúvidas



Bancos de Dados em Clouds

Erik Williams Zirke Osta
Rafael Brundo Uriarte

Mais Sobre Key/Value DB

Database Definition	
Relational Database	Key/Value Database
<ul style="list-style-type: none">• Database contains tables, tables contain columns and rows, and rows are made up of column values. Rows within a table all have the same schema.• The data model is well defined in advance. A schema is strongly typed and it has constraints and relationships that enforce data integrity.• The data model is based on a “natural” representation of the data it contains, not on an application’s functionality.• The data model is normalized to remove data duplication. Normalization establishes table relationships. Relationships associate data between tables.	<ul style="list-style-type: none">• Domains can initially be thought of like a table, but unlike a table you don’t define any schema for a domain. A domain is basically a bucket that you put items into. Items within a single domain can have differing schemas.• Items are identified by keys, and a given item can have a dynamic set of attributes attached to it.• In some implementations, attributes are all of a string type. In other implementations, attributes have simple types that reflect code types, such as ints, string arrays, and lists.• No relationships are explicitly defined between domains or within a given domain.

Mais Sobre Key/Value DB

Data Access

Relational Database

- Data is created, updated, deleted, and retrieved using SQL.
- SQL queries can access data from a single table or multiple tables through table joins.
- SQL queries include functions for aggregation and complex filtering.
- Usually contain means of embedding logic close to data storage, such as triggers, stored procedures, and functions.

Key/Value Database

- Data is created, updated, deleted, and retrieved using API method calls.
- Some implementations provide basic SQL-like syntax for defining filter criteria.
- Basic filter predicates (such as =, !=, <, >, <=, and >=) can often only be applied.
- All application and data integrity logic is contained in the application code.

Mais Sobre Key/Value DB

Application Interface

Relational Database

- Tend to have their own specific API, or make use of a generic API such as OLE-DB or ODBC.
- Data is stored in a format that represents its natural structure, so must be mapped between application code structure and relational structure.

Key/Value Database

- Tend to provide SOAP and/or REST APIs over which data access calls can be made.
- Data can be more effectively stored in application code that is compatible with its structure, requiring only relational “plumbing” code for the object.

Mais Sobre Key/Value DB

- Quando usar uma key/value Db:
 - Quando os dados são orientados a documento;
 - Quando o ambiente de desenvolvimento é orientado a objeto este tipo de base pode reduzir o tempo de implementação e a necessidade de adaptar o código;
 - A aplicação é simples e pode ser usada com este tipo de base;
 - Quando as principais preocupações são grande escalabilidade, uso sob-demanda ou custos.