

# NoSQL: Vantagens, Desvantagens e Compromissos

Mauricio De Diana (mestrando)  
Marco Aurélio Gerosa (orientador)

# Agenda

Definição de NoSQL

Atributos de qualidade e trocas

Modelo de dados

Escalabilidade

Transações

Consistência e disponibilidade

Desempenho

# Dados em larga escala na web - Publicações

Bigtable (Google)

Dynamo (Amazon)

PNUTS (Yahoo!)

Além de:

GFS, MapReduce, Chubby...

Chang, F. et al (2006). Bigtable: A Distributed Storage System for Structured Data.

DeCandia, G. et al (2007). Dynamo: Amazon's Highly Available Key-value Store.

Cooper, B. et al (2008). PNUTS: Yahoo!'s Hosted Data Serving Platform.

# Dados em larga escala na web - FLOSS

Cassandra

MongoDB

Neo4J

Redis

Riak

...

# SGBDs NoSQL

Não-relacionais

Não-ACID

Distribuídos

# SGBDs NoSQL

Não-relacionais

Não-ACID

Distribuídos

Para modelos de dados:

24/09, 14:00, CEC (lab 06): Aula de MAC5855

# Atributos de qualidade

Tempo de projeto

Modificabilidade

Testabilidade

...

Tempo de execução

Desempenho

Disponibilidade

Escalabilidade

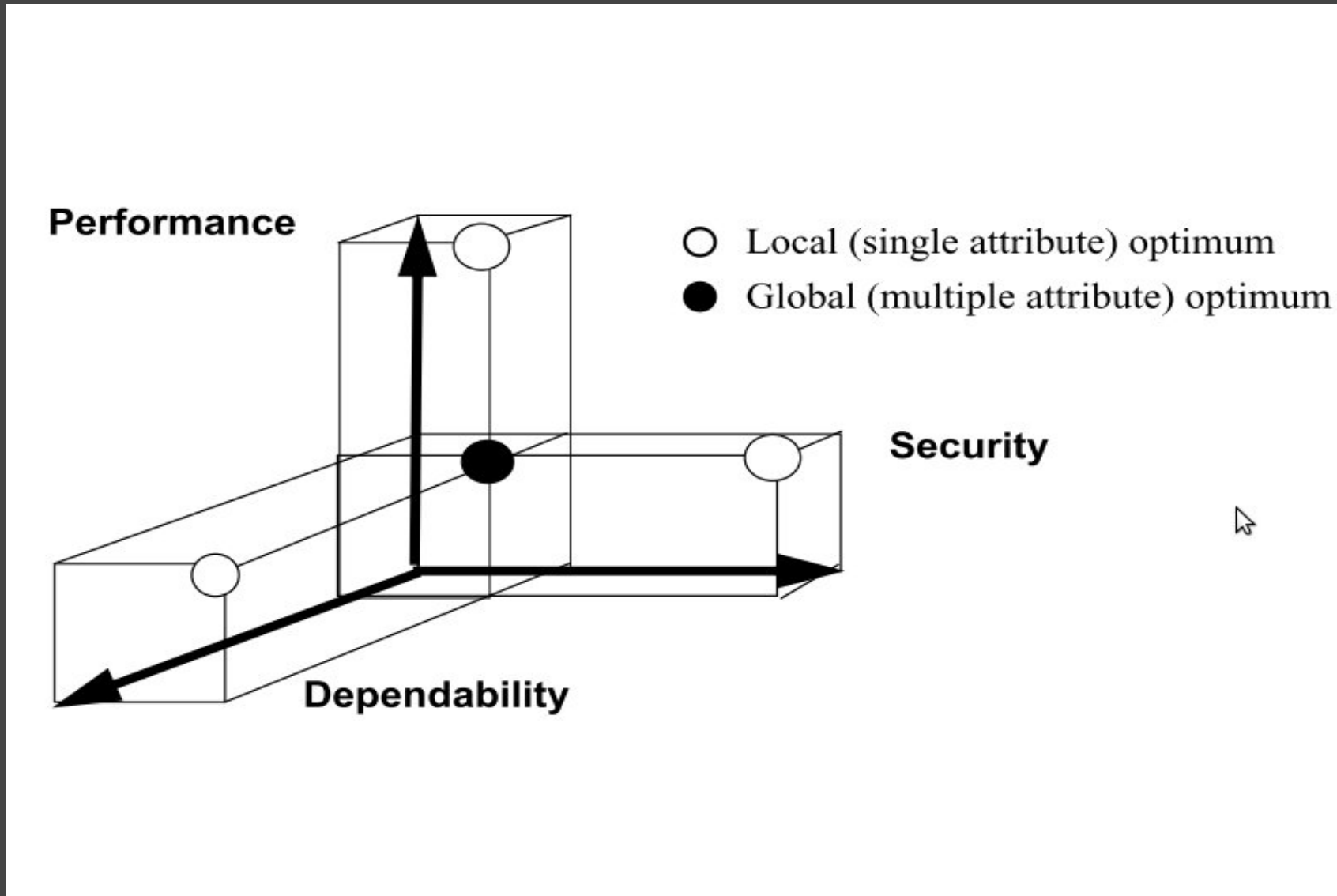
Segurança

...

Barbacci, M. (1995). Quality Attributes.

Bass, L. et al. (2003). Software Architecture in Practice. 2a ed.

# Atributos de qualidade - Trocas



# Modelo de dados

# Modelos não-relacionais

Grafos: interconectividade dos dados é tão ou mais importante quanto os dados em si

Chave-valor: modelo simples

Orientado a documentos: dados semiestruturados

Angles, R. e Gutierrez, C. (2008) Survey of Graph Database Models.  
Buneman, P. (1997). Semistructured Data.

# SGBDs relacionais

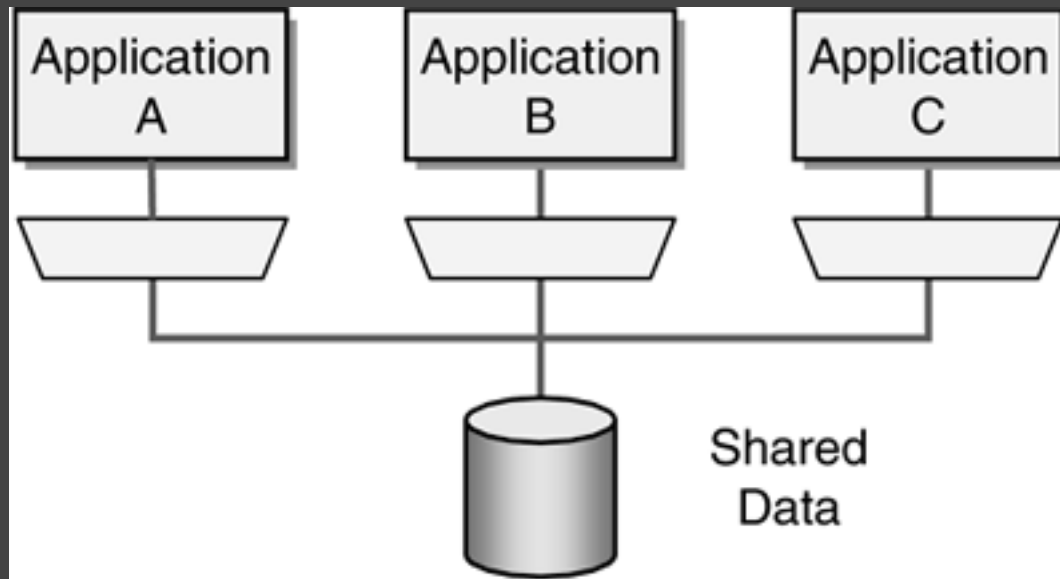
Esquema

Restrições de integridade  
(entidade, referencial, domínio)

Normalização

# Esquema

## Banco de dados compartilhado



# Diferença de impedância

OO

Herança

Encapsulamento

Polimorfismo

...

Relacional

Relacionamentos

Junções

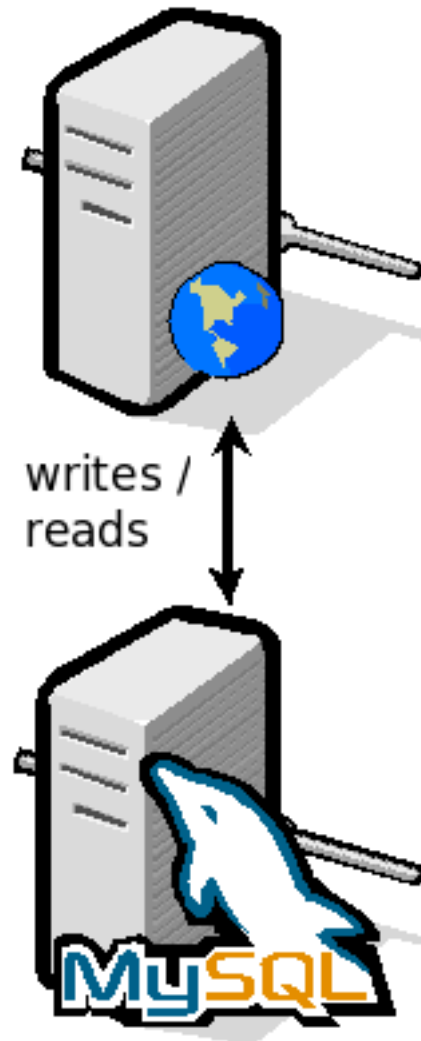
Normalização

...

Ambler, S. The Object-Relational Impedance Mismatch. <http://www.agiledata.org/essays/impedanceMismatch.html>

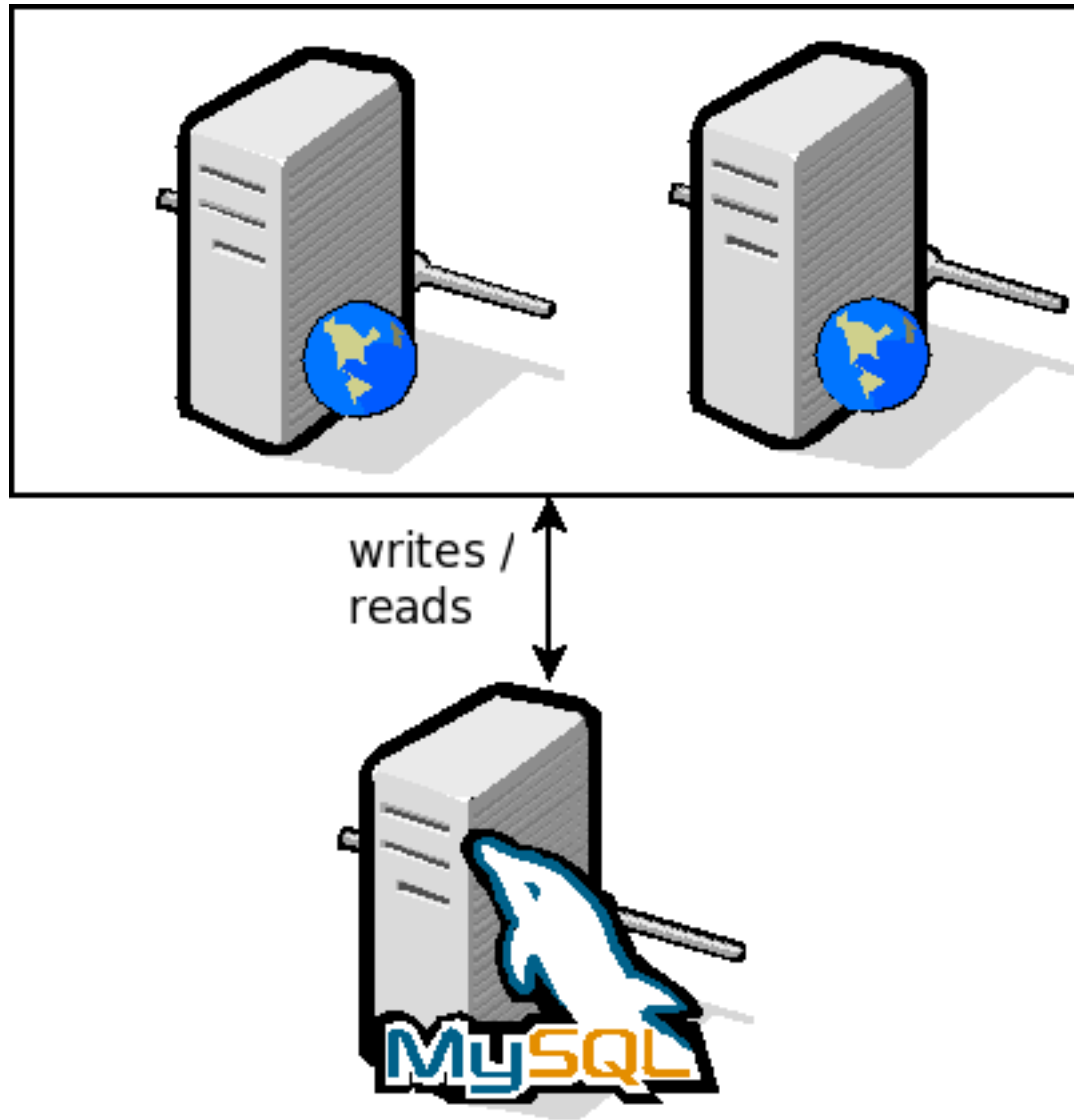
# Escalabilidade

# Escalando relacional

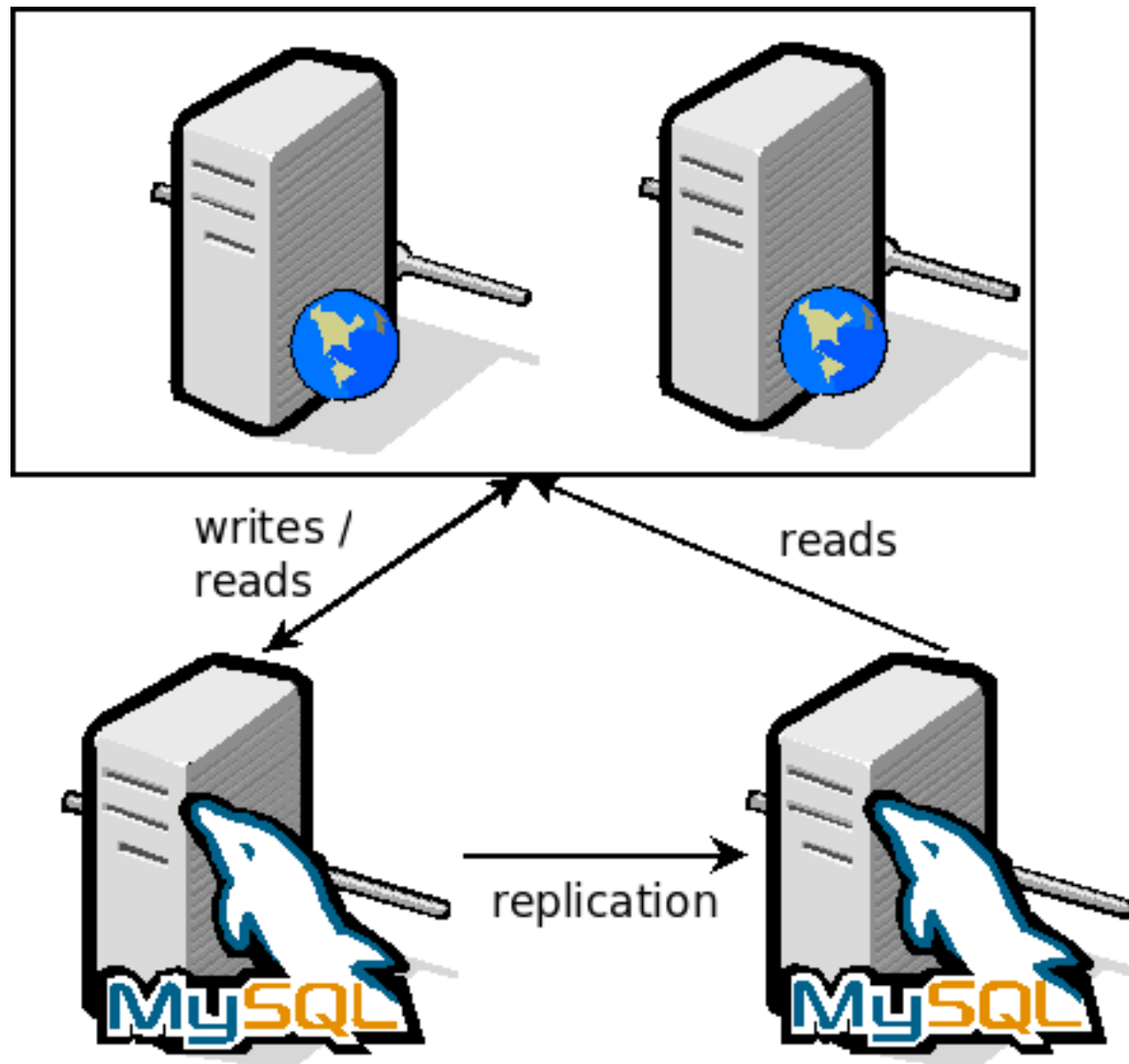


# N servidores web

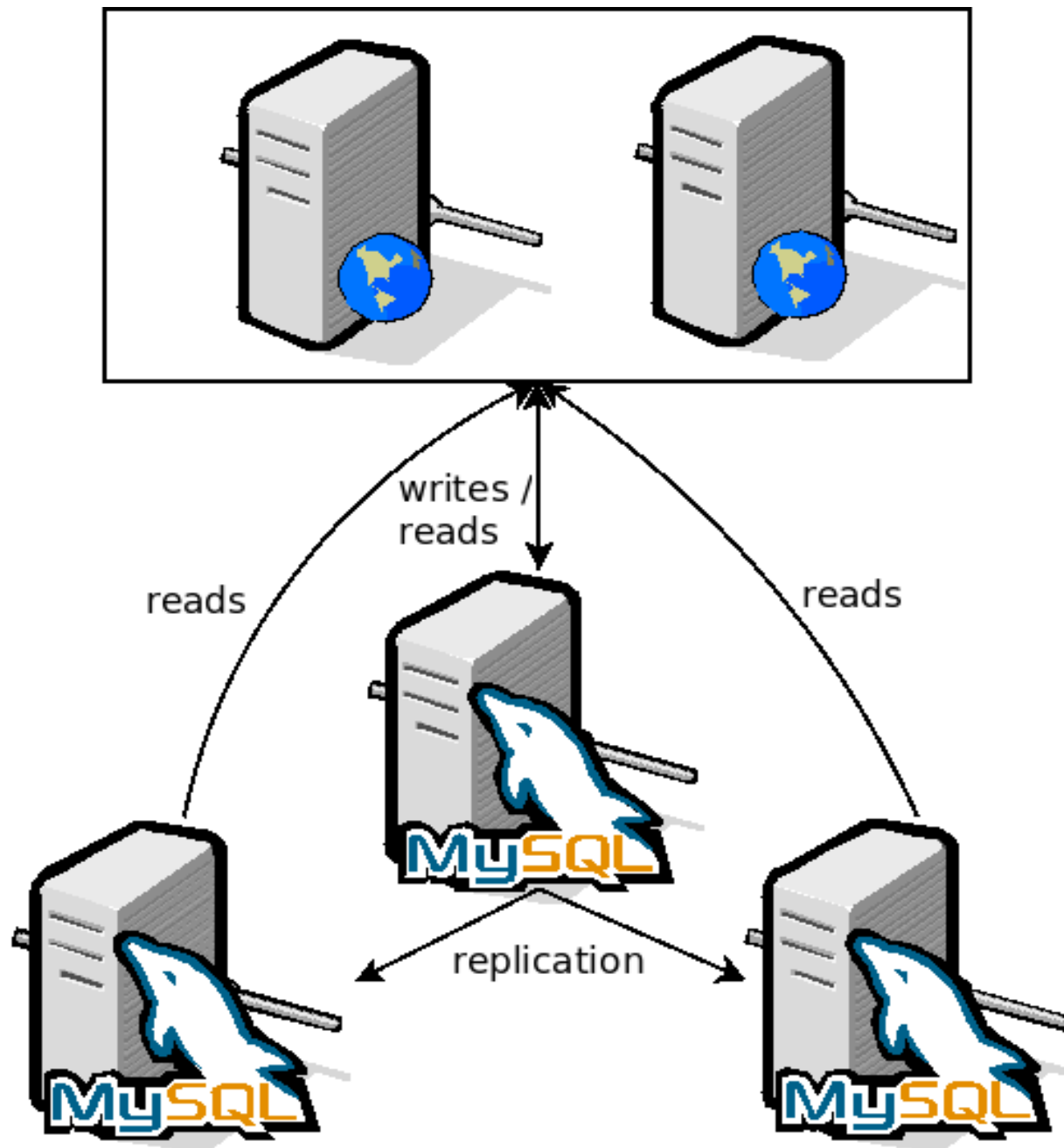
## 1 db



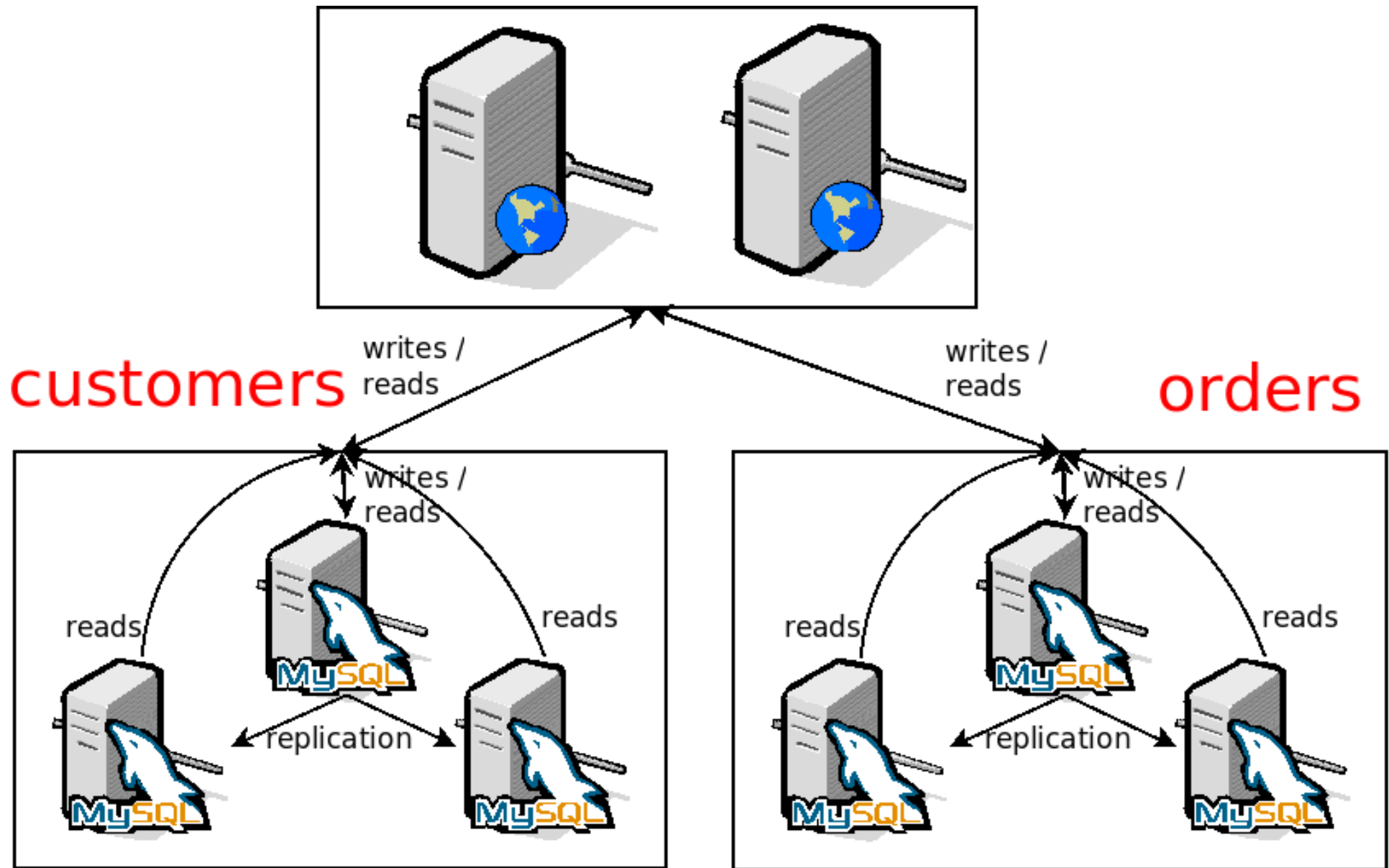
# Mestre / Escravo



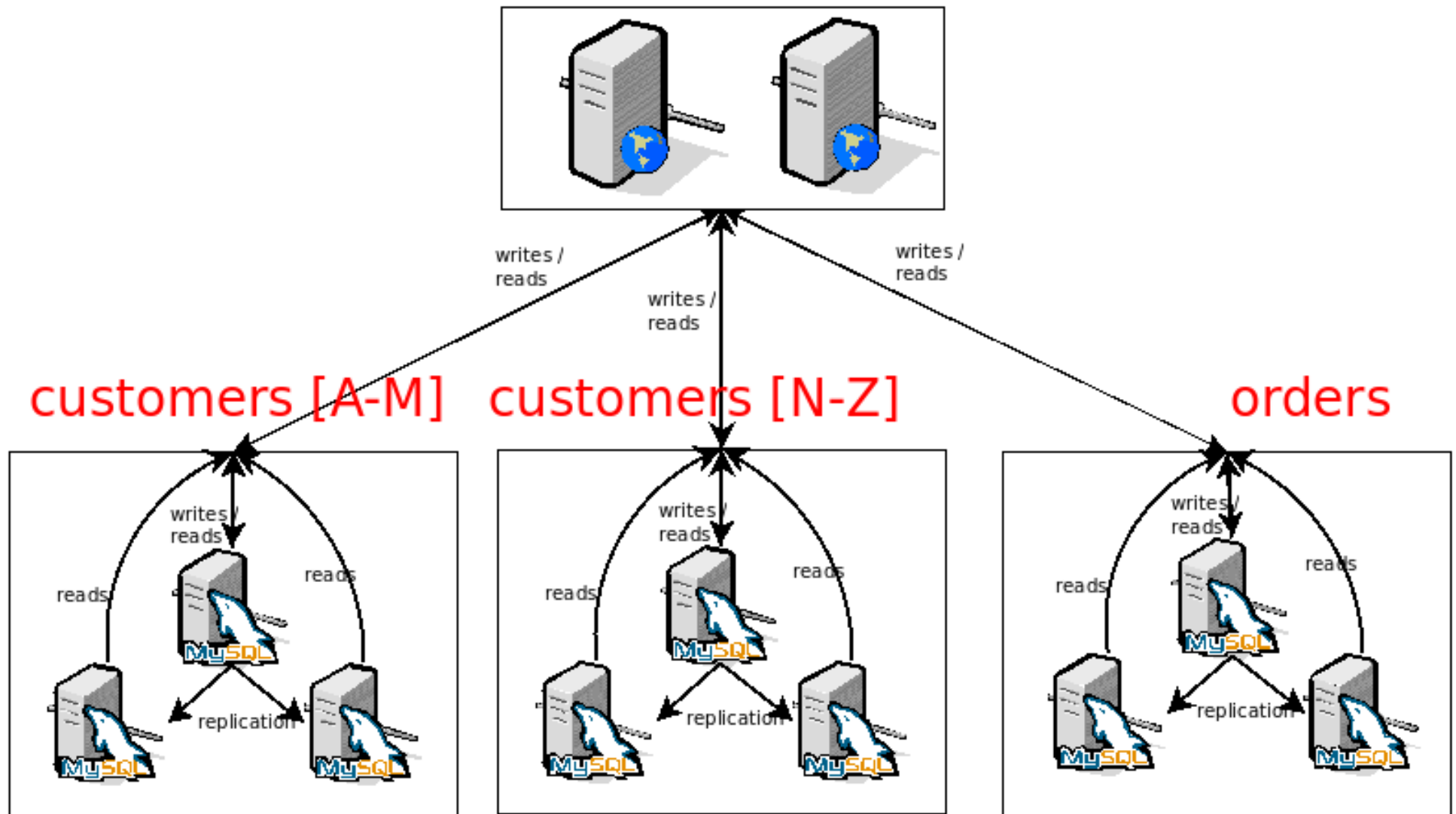
# Mestre / Escravo



# Particionamento funcional



# Particionamento horizontal



# Perde-se

Junções

Normalização

Integridade de entidade e referencial

Transparência de localização

# Transações

# ACID

Atomicidade

Consistência

Isolamento

Durabilidade

# ACID e escalabilidade

Atomicidade: protocolo distribuído (2PC)

Consistência: problemas com réplicas

Isolamento: *locks* distribuídos

Abadi, D. (2010). The problems with ACID, and how to fix them without going NoSQL.  
<http://dbmsmusings.blogspot.com/2010/08/problems-with-acid-and-how-to-fix-them.html>

# Exemplos de trocas com ACID em NoSQL

Atomicidade e isolamento: em único banco de dados (Bigtable)

Consistência: em momento indeterminado (Dynamo)

Durabilidade: memória + *snapshotting* (Redis)

Pritchett, D. (2008). BASE: An Acid Alternative.

Vogels, W. (2009). Eventually Consistent.

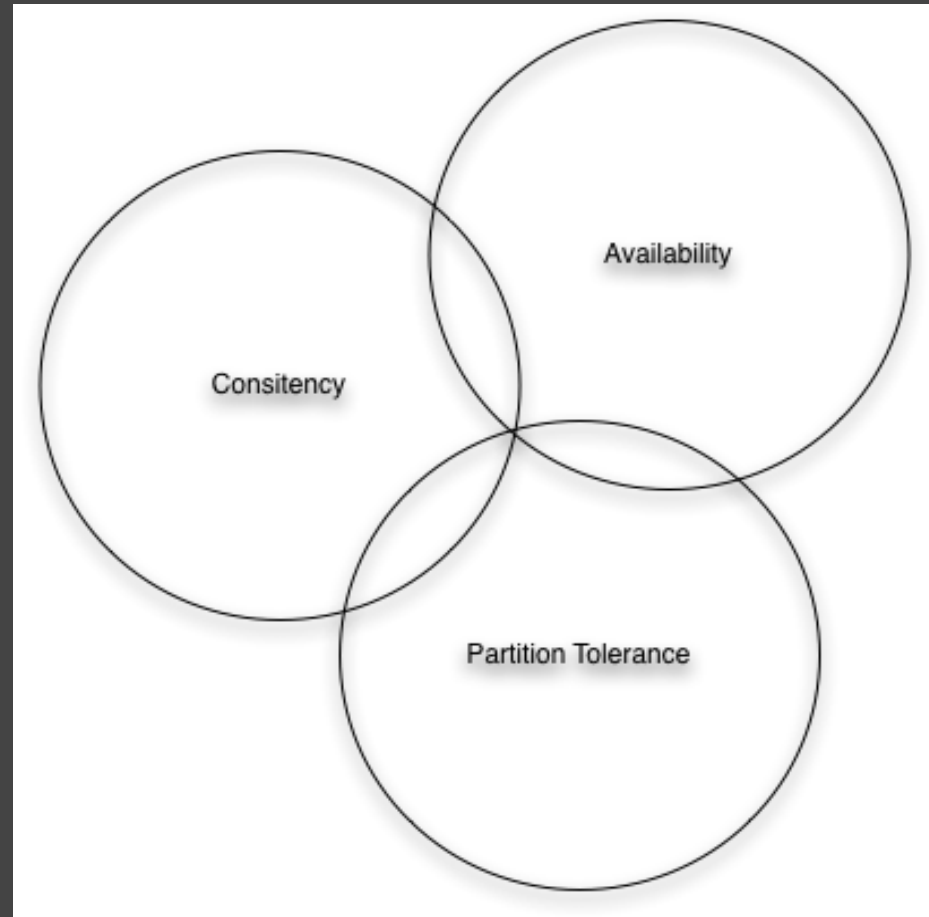
# Consistência e Disponibilidade

# Teorema CAP

Consistência

Disponibilidade

Tolerância à partição



Fonte: <http://blog.mattwoodward.com/>

Brewer, E. (2000). Towards Robust Distributed Systems.

Gilbert, S. e Lynch, N. (2002). Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-tolerant Web Services

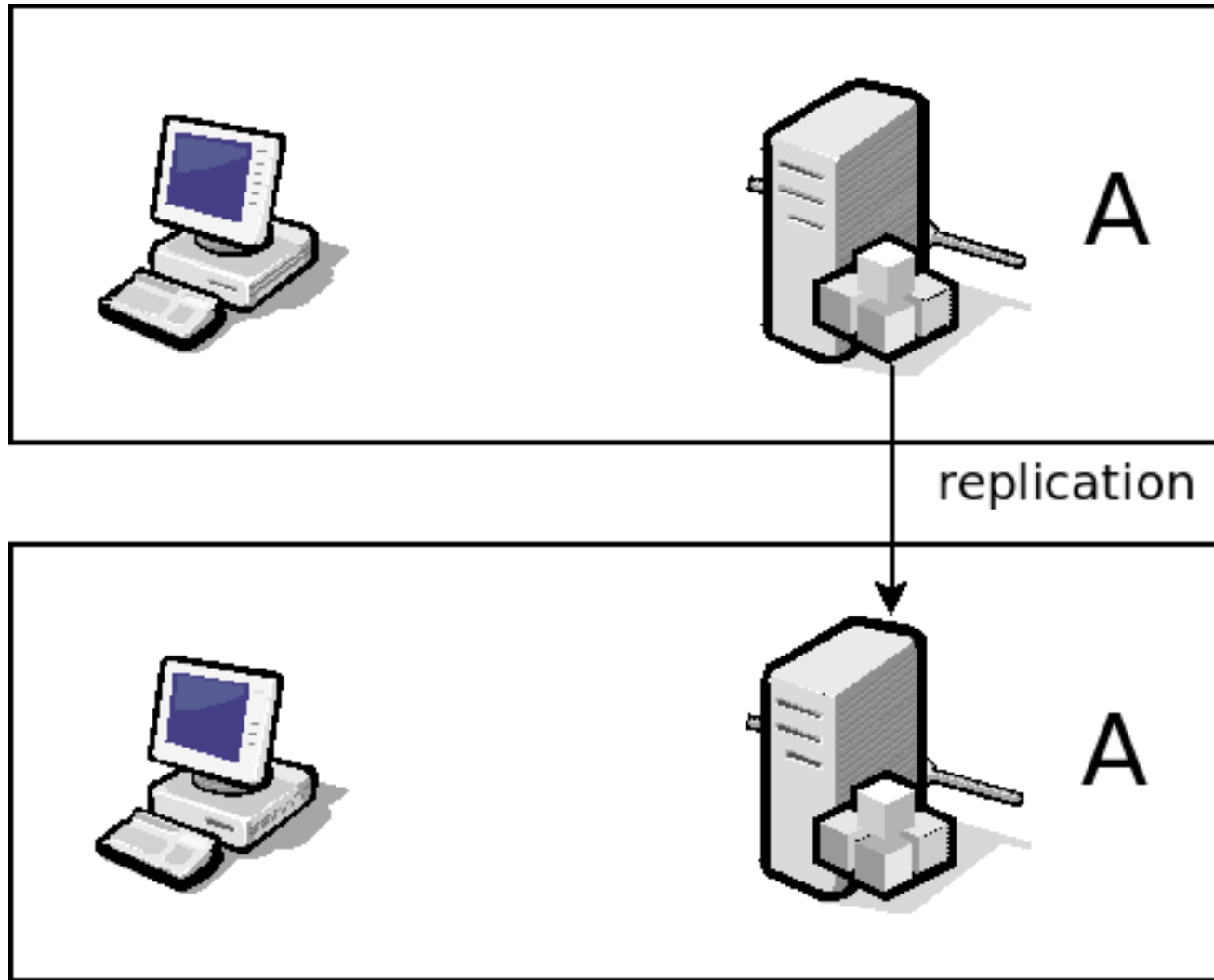
# ACID x BASE

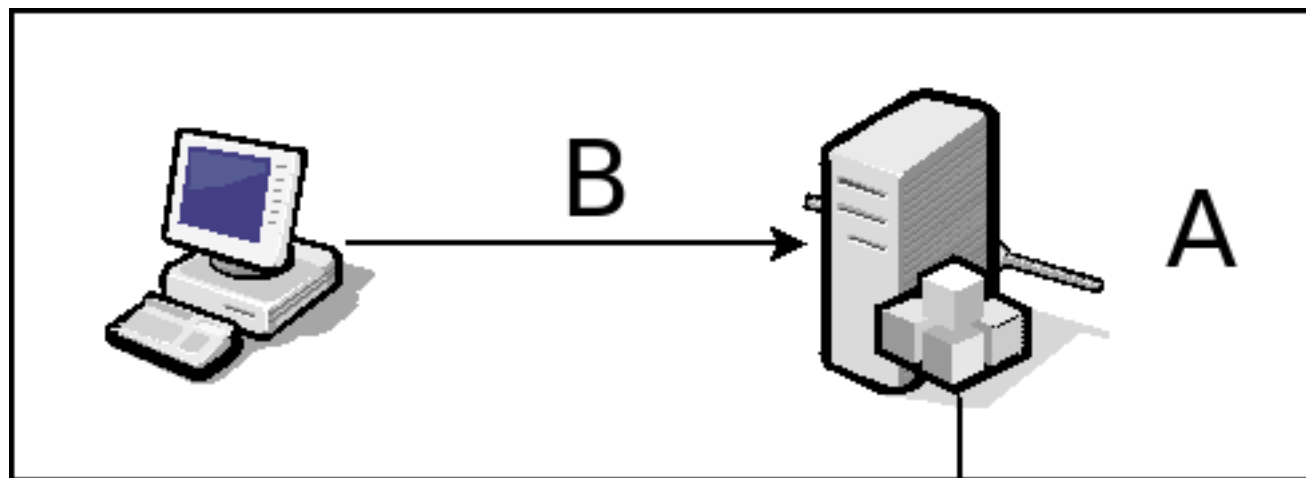
Basicamente disponível (*Basically available*)

*Soft state*

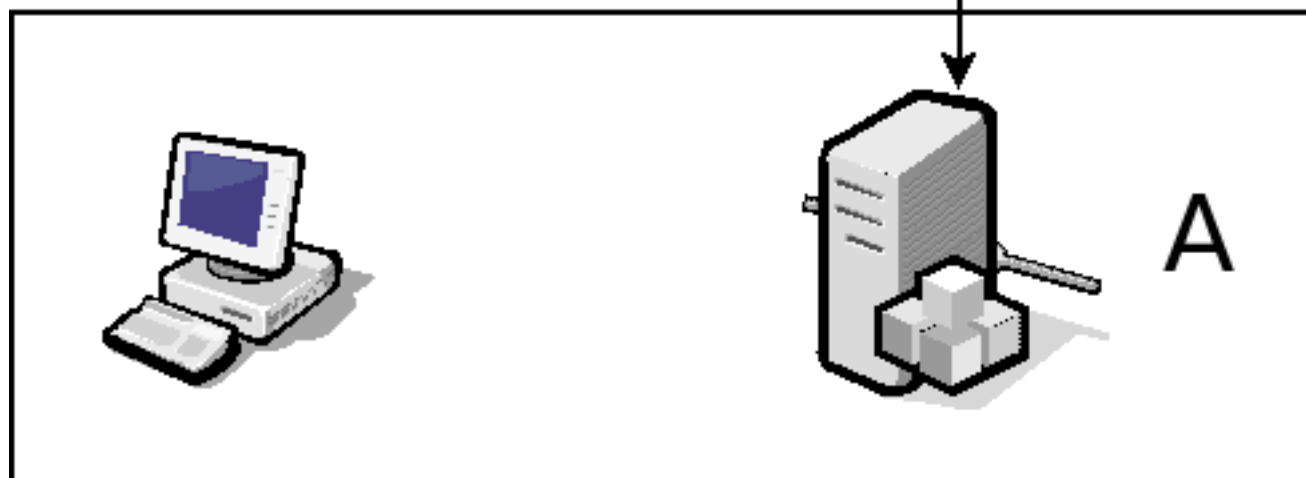
Consistente em momento indeterminado (*Eventually consistent*)

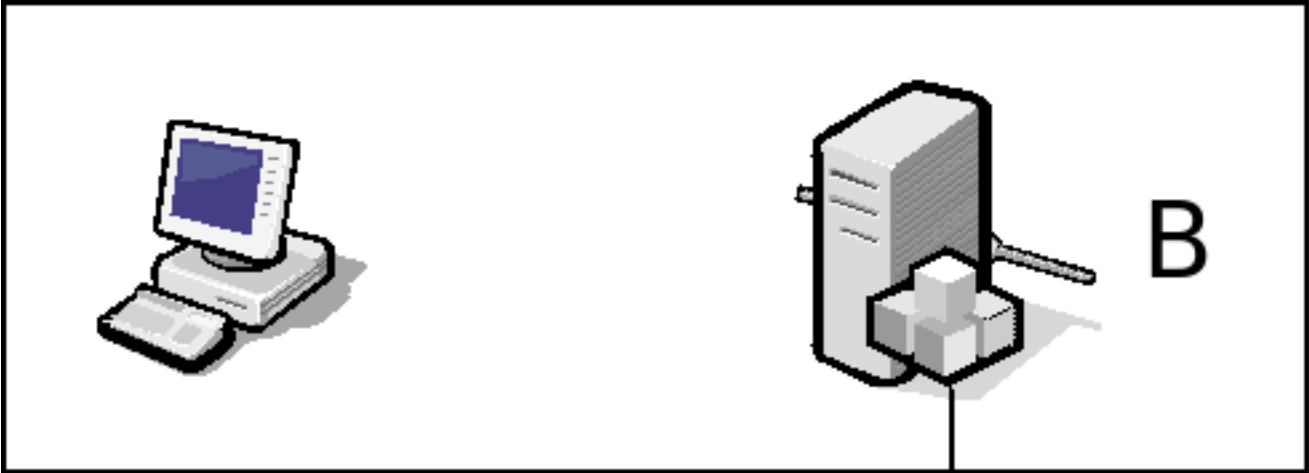
# Consistência em momento indeterminado



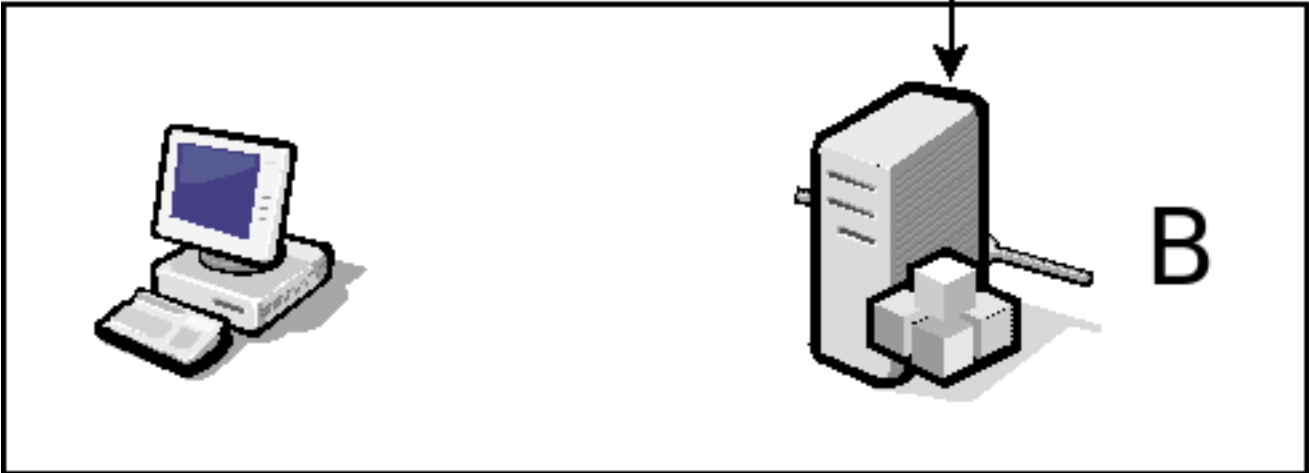


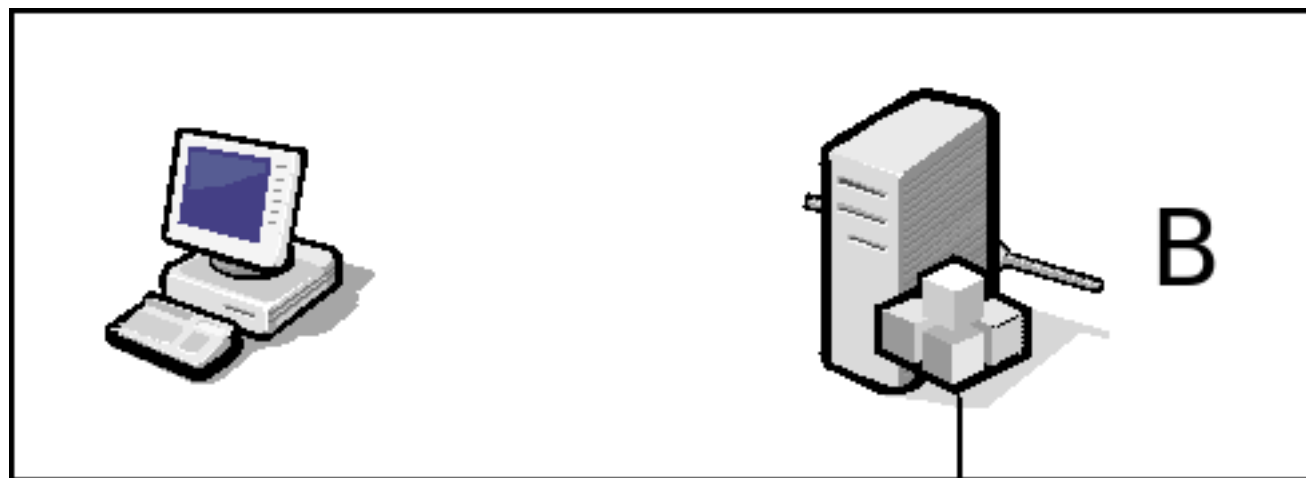
replication



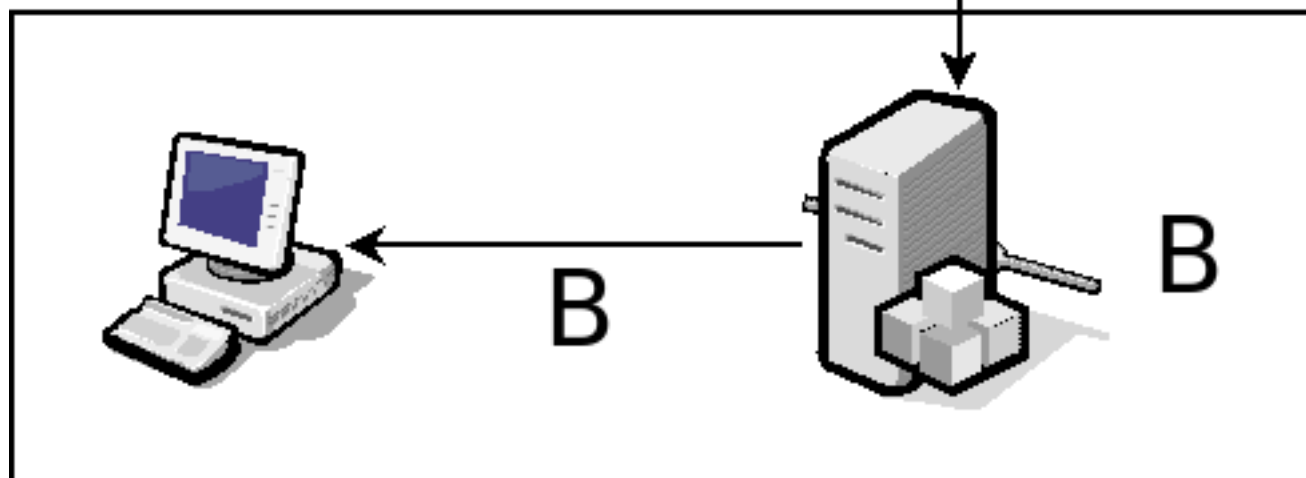


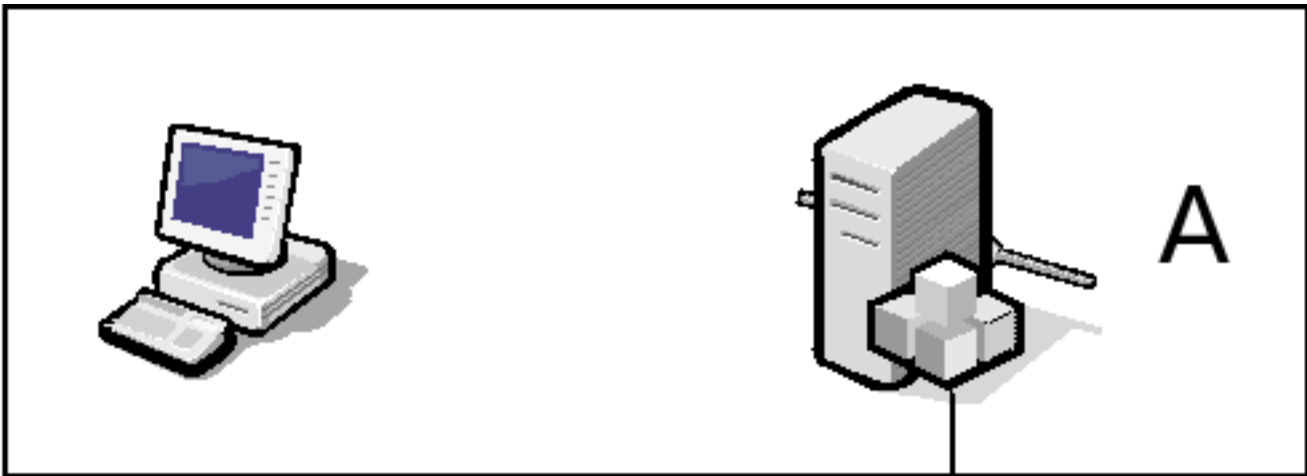
replication



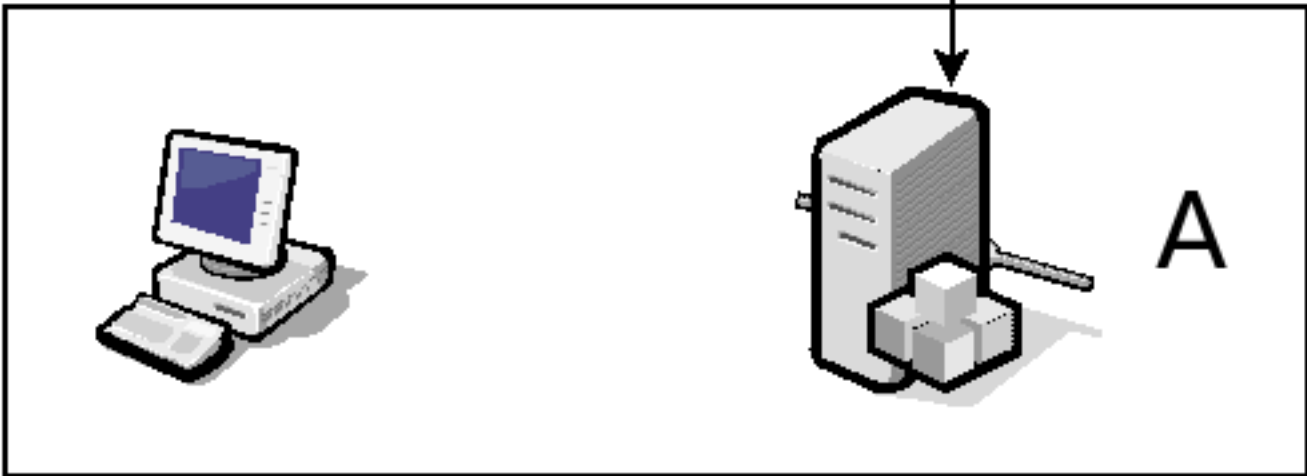


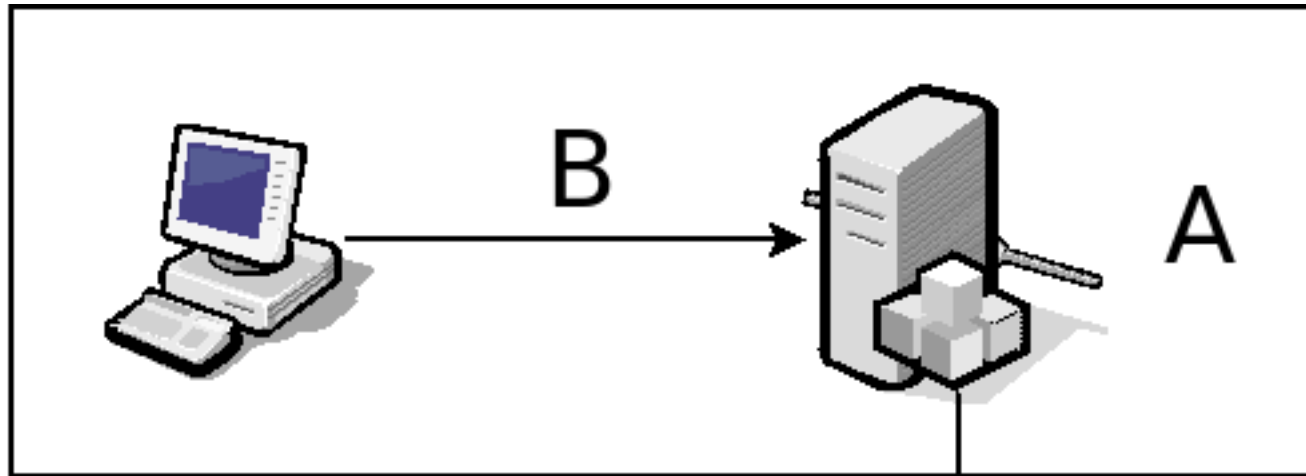
replication





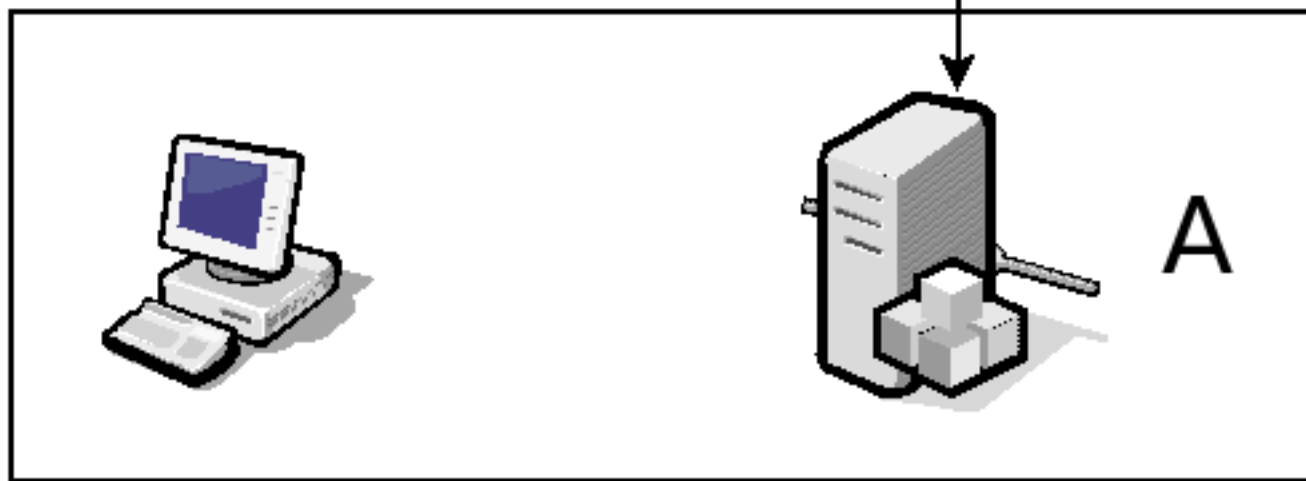
replication

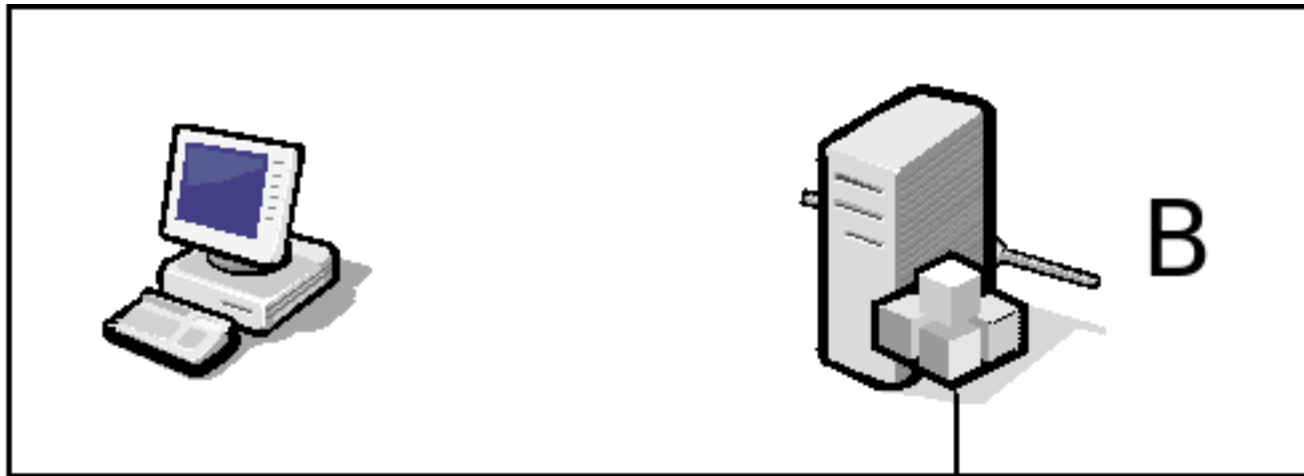




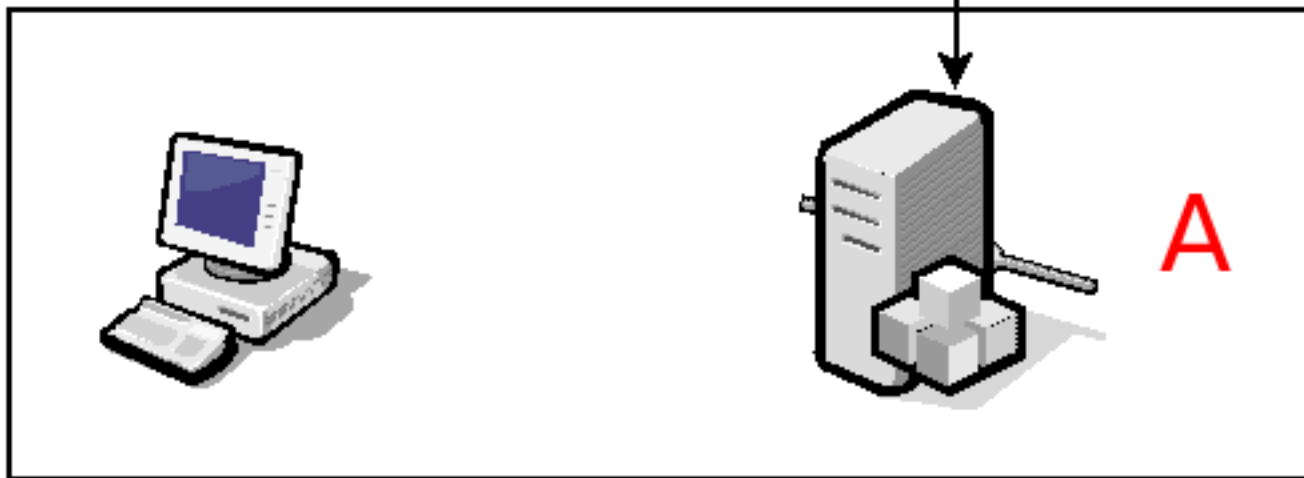
replication

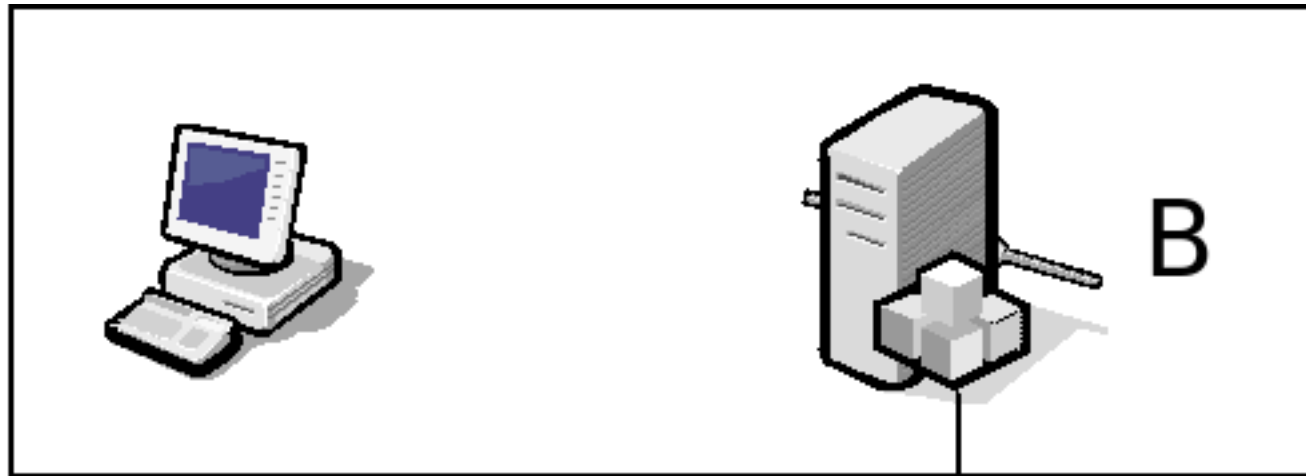
---



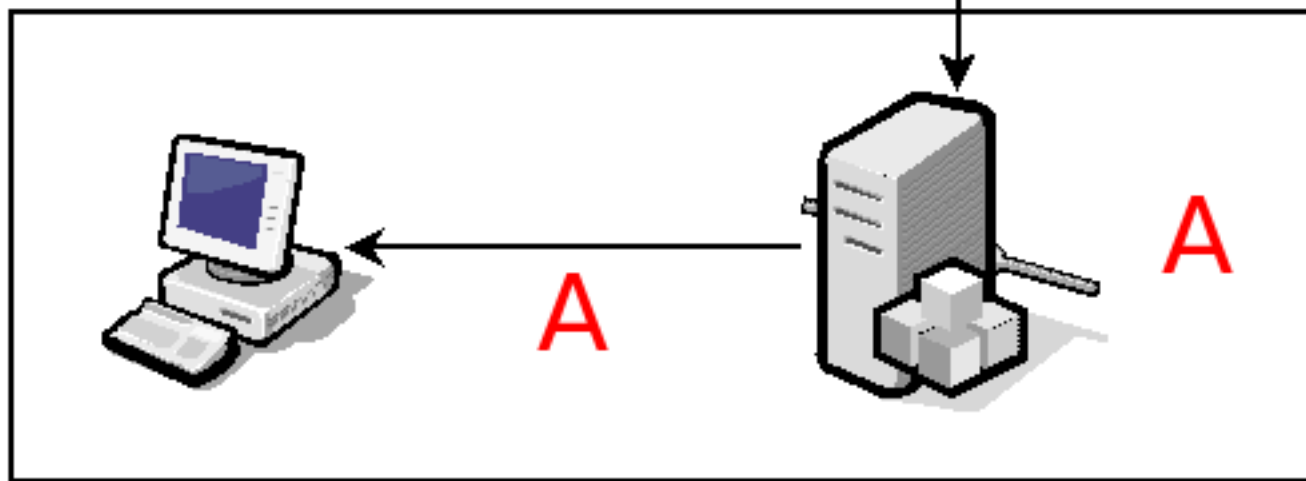


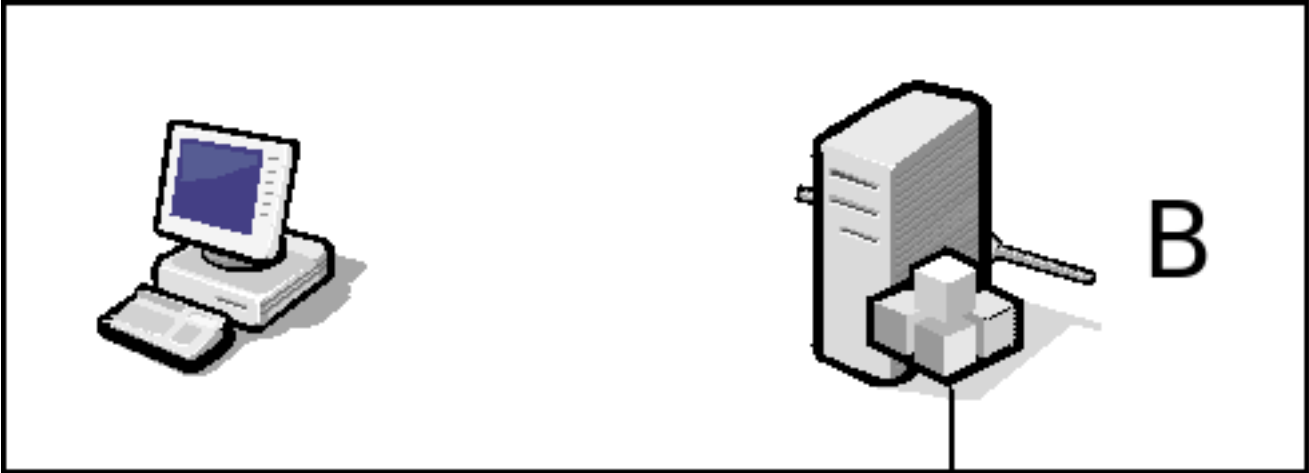
replication



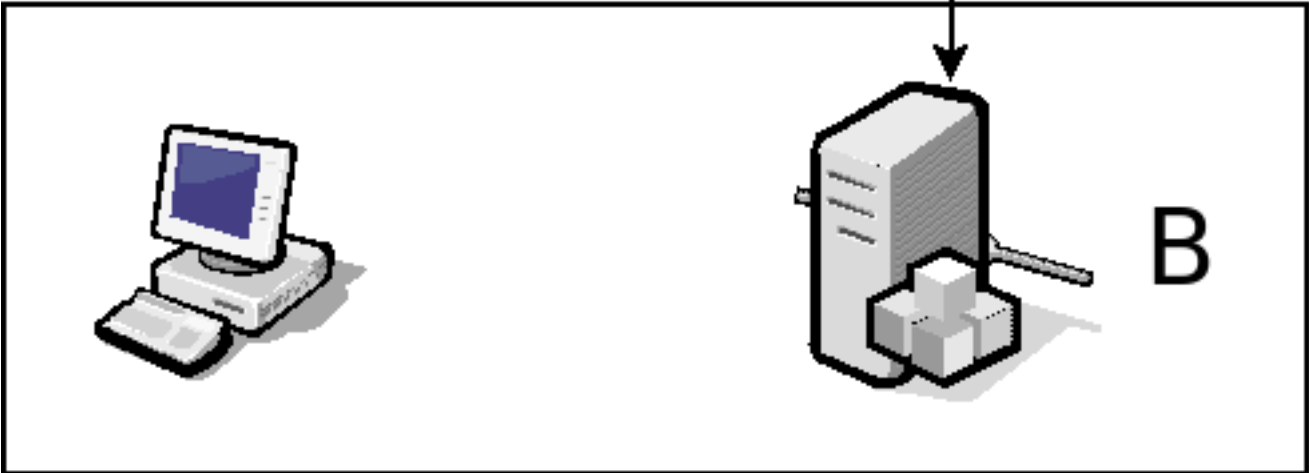


A horizontal red line spans the width of the diagram. The word "replication" is written in black text below the line, centered under the arrow that connects the two environments.





replication



# Consertando inconsistências

Fazer nada

Tentar novamente

Ação de compensação

Hohpe, G. (2004). Starbucks Does Not Use Two-Phase Commit. [http://www.eaipatterns.com/ramblings/18\\_starbucks.html](http://www.eaipatterns.com/ramblings/18_starbucks.html)

Desempenho

# Cenários

Cargas de trabalho diferentes precisam de tratamento diferente

## OLTP

Operação

CRUD pequeno e rápido

Consultas simples

## OLAP

Informação (BI)

Batches demorados

Consultas complexas

Stonebraker, M. e Cetintemel, U. (2005). "One Size Fits All": An Idea Whose Time Has Come and Gone.

# Experimentos

## OLTP

H-Store (VoltDB)

82x mais rápido que um SGBD comercial

## OLAP

C-Store (Vertica)

124x mais rápido que um SGBD comercial orientado a linha

21x mais rápido que um SGBD comercial orientado a coluna

Stonebraker, M. et al. (2007). The End of an Architectural Era (It's Time for a Complete Rewrite).

Stonebraker, M. et al. (2005). C-store: a Column-oriented DBMS.

# Tempos em um SGBDR

*Logging, locking, latching* e gerenciamento de buffer

1/60 das instruções em uma transação são trabalho útil

20x mais rápido sem esses sub-sistemas

Harizopoulos, S. et al. (2008). OLTP Through the Looking Glass, and What We Found There.

# Especializações

Problema não está no modelo relacional, nem na SQL

Ordens de grandeza de diferença

Persistência poliglota

# Conclusões

Contexto importa

Balancear vantagens e desvantagens é fundamental

# Próximos passos

Foco em chave-valor (DHT)

Desenvolvimento de taxonomia

Arquitetura de referência

Experimentos de escalabilidade e desempenho

riak\_core

Benchmark para cloud do Yahoo!

Obrigado.

Comentários, dúvidas, sugestões?

mdediana@ime.usp.br  
mdediana@gmail.com  
@mdediana