

Banco de Dados NoSQL

Prof. Henrique Batista da Silva

CAP Theorem

CAP Theorem

- Para muitas soluções em banco de dados é essencial manter **alta disponibilidade** (serviço funcionando por maior tempo possível).
- Neste cenário, pelo menos dois servidores são necessários (com computação em nuvem, vários podem ser utilizados)

CAP Theorem

- Teorema **CAP** (CAP Theorem):
- Consistência (**C** - Consistency); Alta disponibilidade (**A** - Availability); e Tolerância a particionamento dos dados na rede (**P** - Partition-tolerance)

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

CAP Theorem

- Dentre as três partições, o teorema CAP diz que é possível apenas obter **duas** delas (impossível ter todas simultaneamente).
- Vamos entender cada partição.

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

CAP Theorem

- Consistência (Consistency): quando dados são alterados por uma operação, qualquer operação posterior (ex. busca) deverá utilizar o dado atualizado.

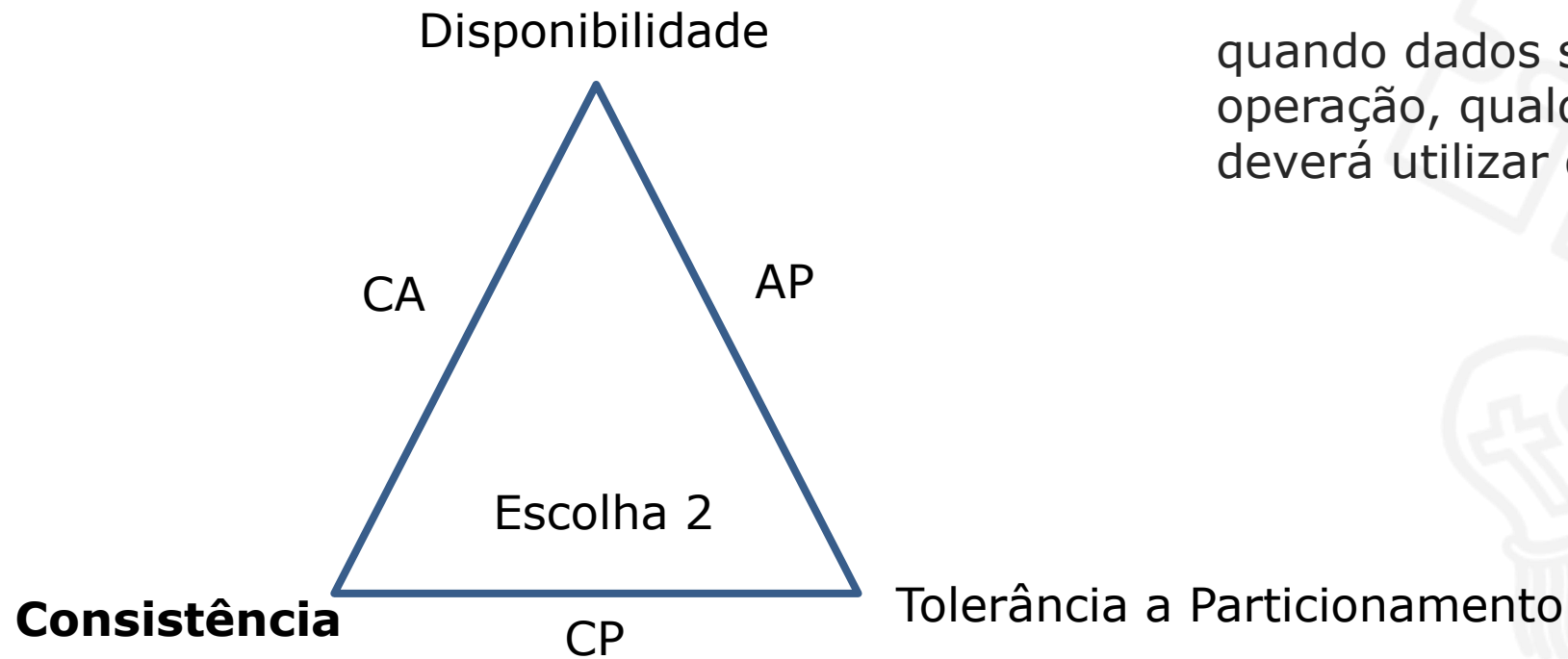
Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

CAP Theorem

- Disponibilidade (Availability): o banco deve estar disponível sempre para leitura e escrita.
- Tolerância a particionamento (Partition-tolerance): servidores devem suportar falhas na comunicação entre eles por tempo indeterminado.

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

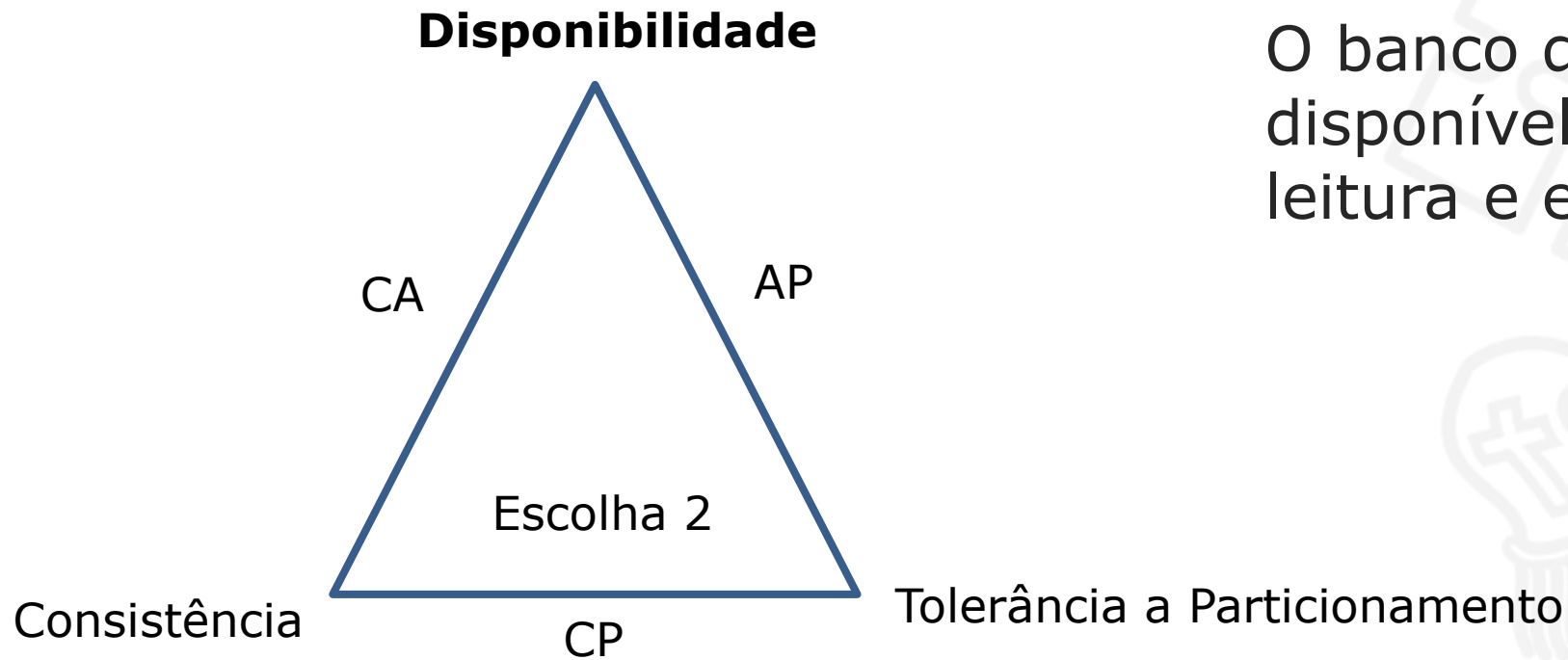
CAP Theorem



quando dados são alterados por uma operação, qualquer operação posterior deverá utilizar o dado atualizado.

Figura adaptada de: Paniz, David. NoSQL: Como armazenar os dados de uma aplicação moderna. Casa do Código, 2017.

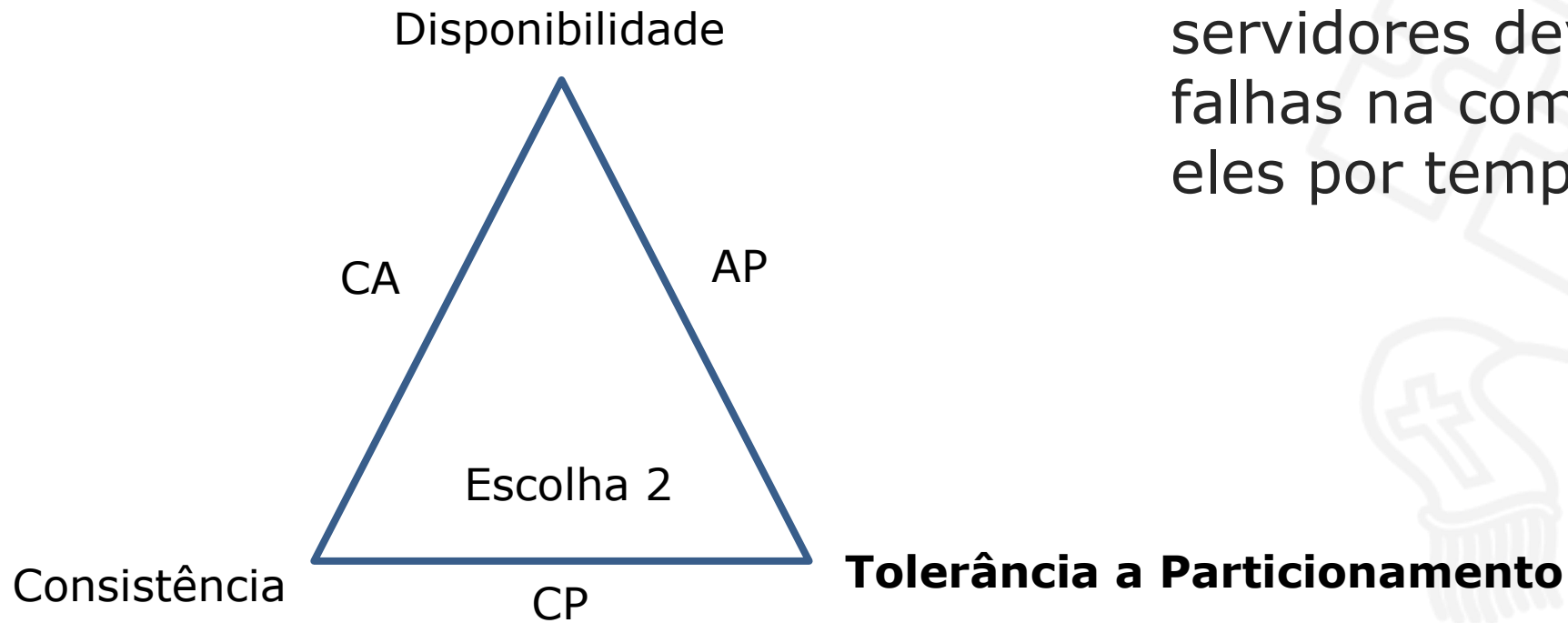
CAP Theorem



O banco deve estar disponível sempre para leitura e escrita

Figura adaptada de: Paniz, David. NoSQL: Como armazenar os dados de uma aplicação moderna. Casa do Código, 2017.

CAP Theorem

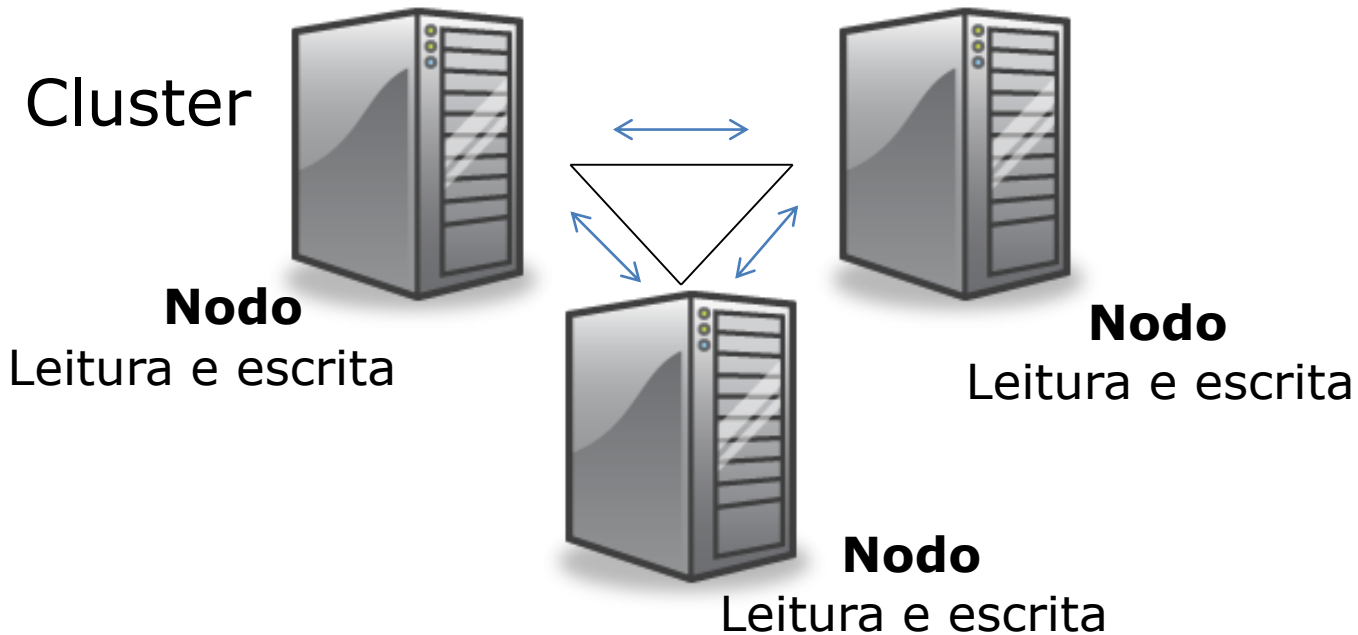


servidores devem suportar falhas na comunicação entre eles por tempo indeterminado

Figura adaptada de: Paniz, David. NoSQL: Como armazenar os dados de uma aplicação moderna. Casa do Código, 2017.

Forte consistência e tolerância a particionamento (CP)

Situação em que necessita de forte consistência (leitura dos dados sempre atualizada).



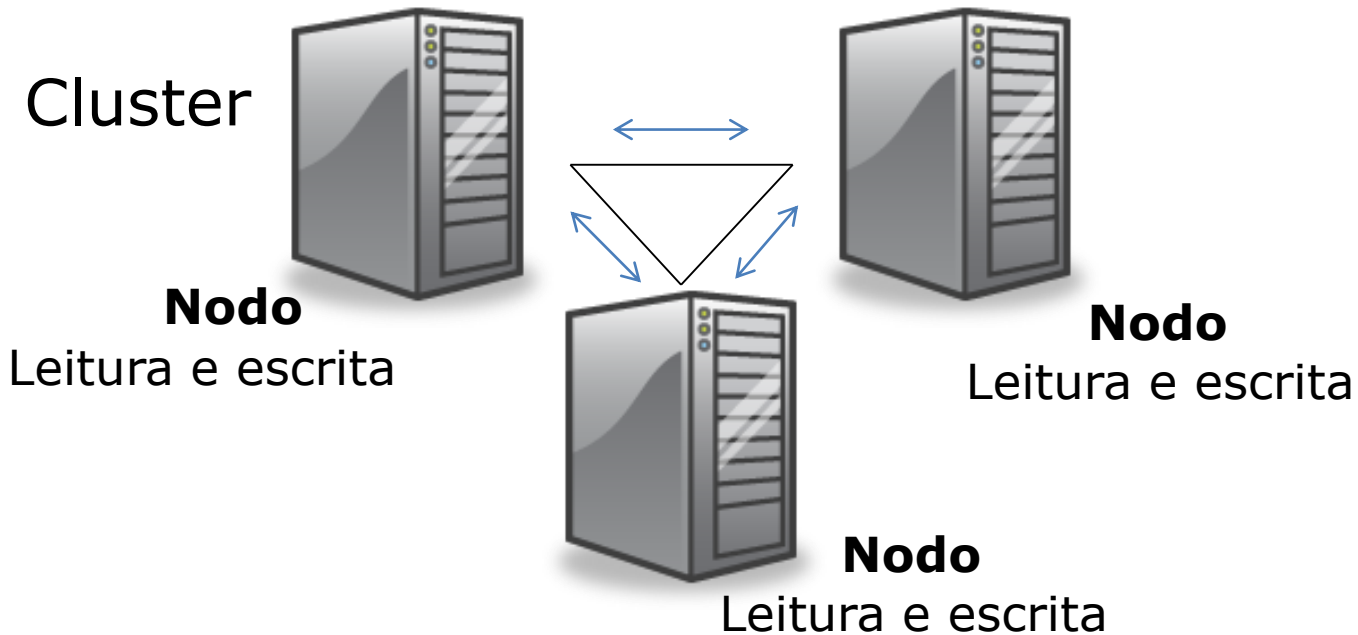
Consenso: suponha um fator de replicação N igual a 3 (figura ao lado). Considere:

- R como número de nós que devem ser contatados para confirma uma leitura;

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Forte consistência e tolerância a particionamento (CP)

Situação em que necessita de forte consistência (leitura dos dados sempre atualizada).



Consenso: suponha um fator de replicação N igual a 3 (figura ao lado).

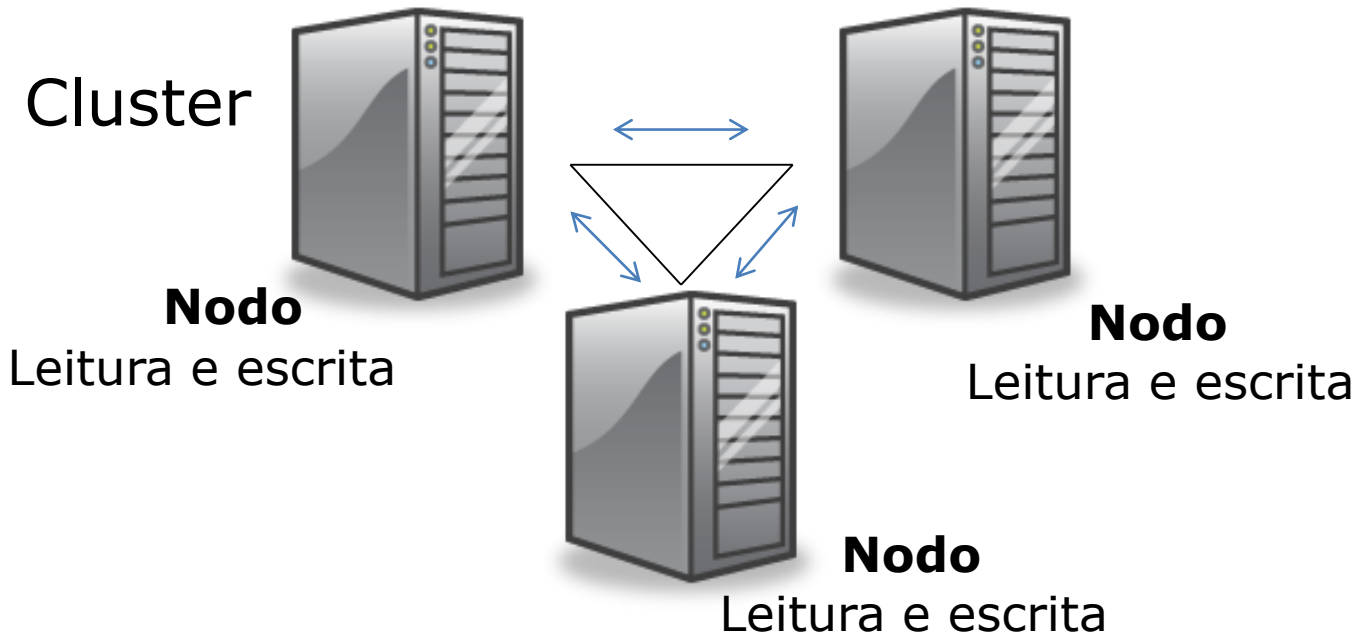
Considere:

- W números de nós a serem contatados para confirmar a escrita.

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Forte consistência e tolerância a particionamento (CP)

Situação em que necessita de forte consistência (leitura dos dados sempre atualizada).

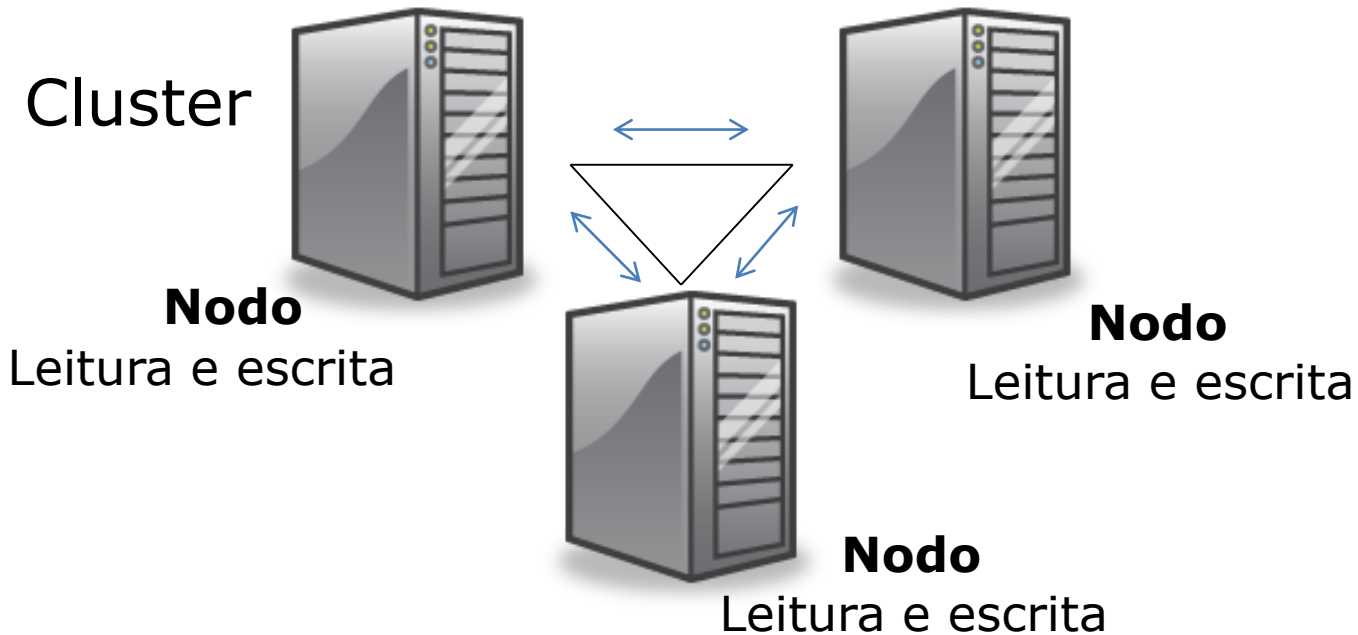


Consenso: O relacionamento entre R , W e N pode ser considerado pela desigualdade: $R + W > N$ (leitura altamente consistente).

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Forte consistência e tolerância a particionamento (CP)

Situação em que necessita de forte consistência (leitura dos dados sempre atualizada).

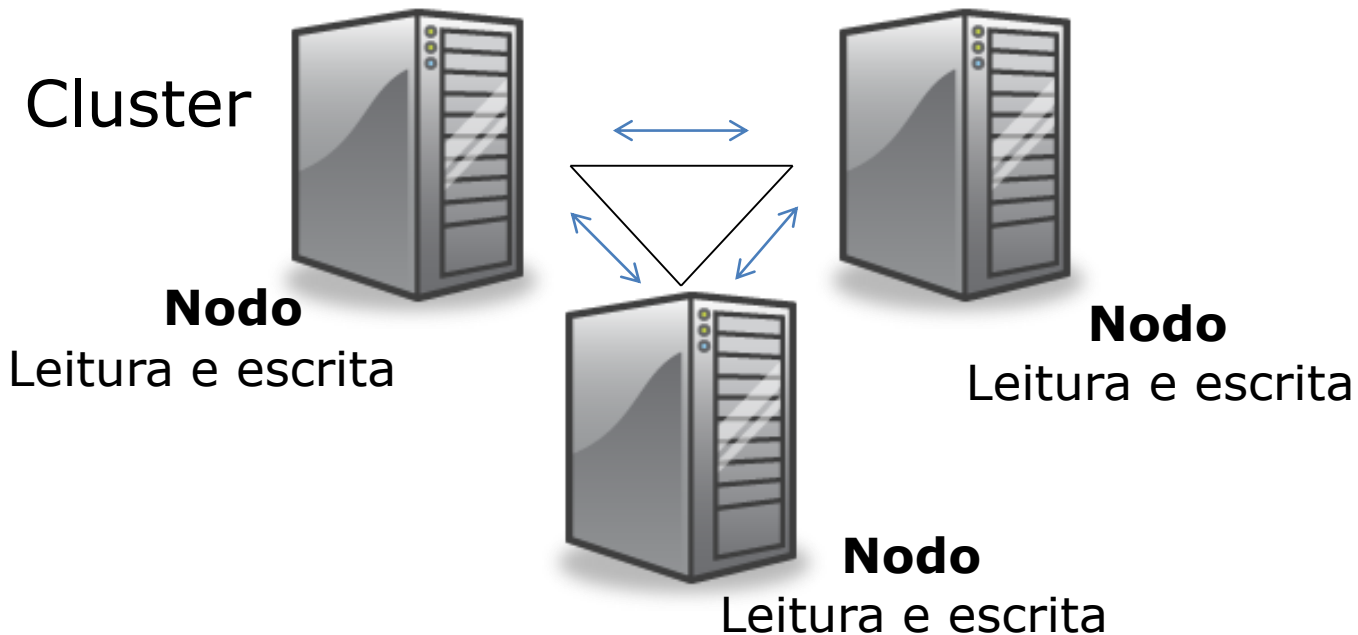


Consenso: Agora considere $R = 1$ e $W = 3$ (N continua sendo 3). Portanto, para escrever no banco é necessária a confirmação de escrita nos três nós (**quórum**).

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Forte consistência e tolerância a particionamento (CP)

Situação em que necessita de forte consistência (leitura dos dados sempre atualizada).

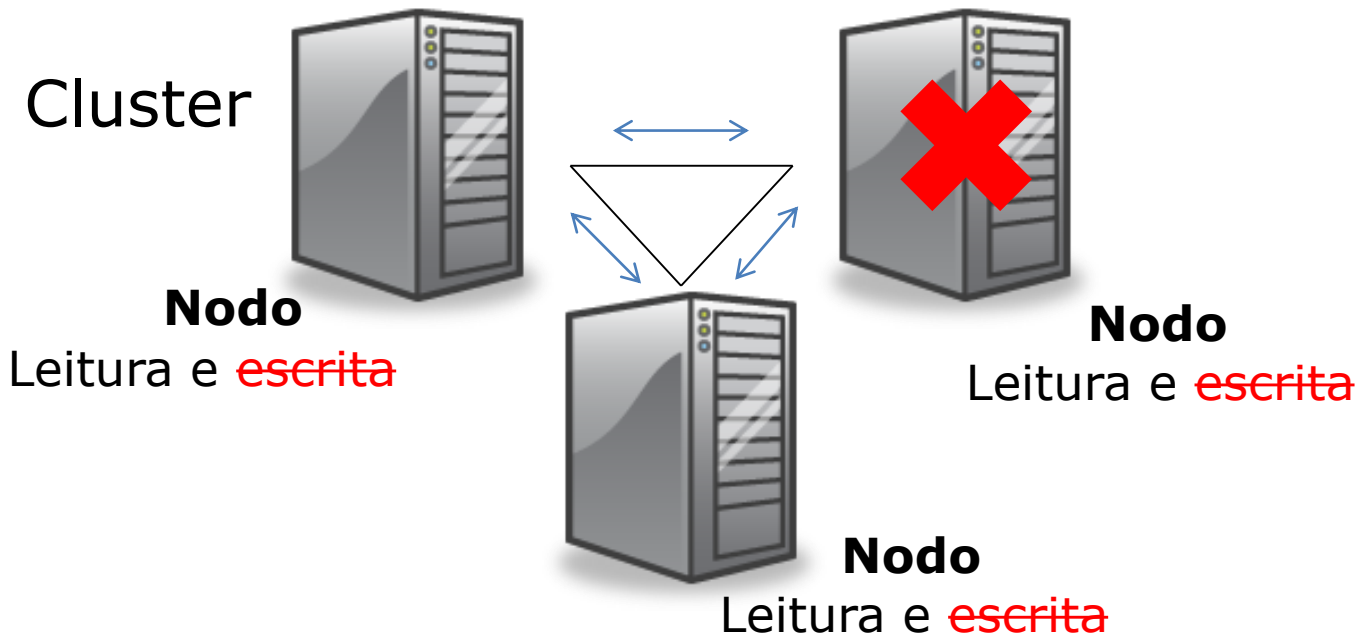


Consenso: Agora considere $R = 1$ e $W = 3$ (N continua sendo 3). Porém, para leitura, basta a confirmação de apenas um nó. Assim:
 $1(R) + 3(W) > 3(N)$.

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Forte consistência e tolerância a particionamento (CP)

Situação em que necessita de forte consistência (leitura dos dados sempre atualizada).



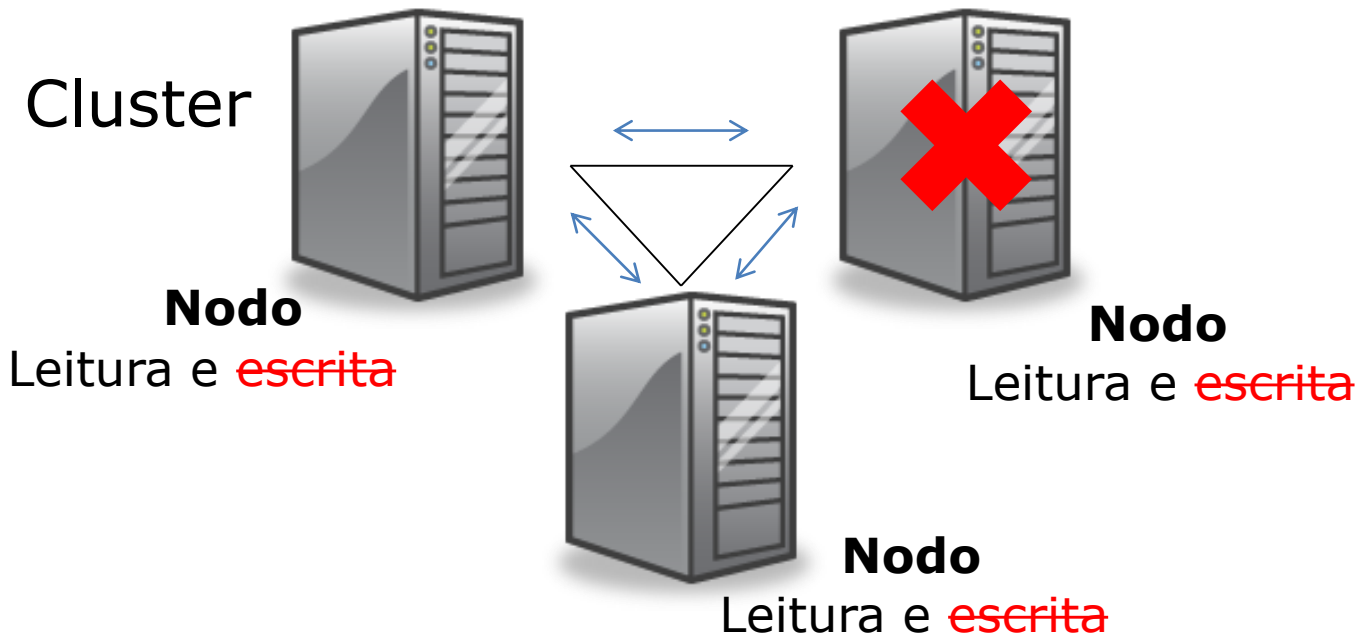
Consenso: Com a falha de um nó, a escrita é comprometida (não há quórum).

Perde-se a disponibilidade (A) para escrita (lembre-se que $W = 3$).

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Forte consistência e tolerância a particionamento (CP)

Situação em que necessita de forte consistência (leitura dos dados sempre atualizada).



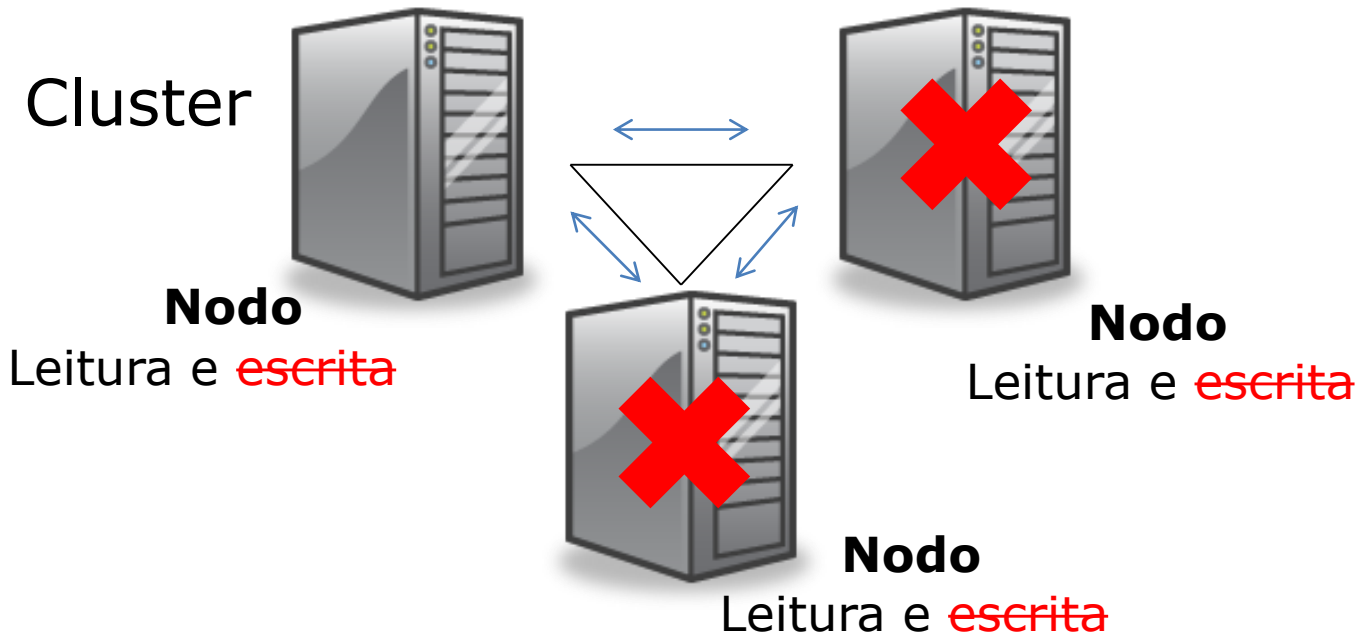
Consenso: Com a falha de um nó, a escrita é comprometida (não há quórum).

Mas a leitura ainda pode ocorrer em qualquer um dos demais nós ($R = 1$).

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Forte consistência e tolerância a particionamento (CP)

Situação em que necessita de forte consistência (leitura dos dados sempre atualizada).

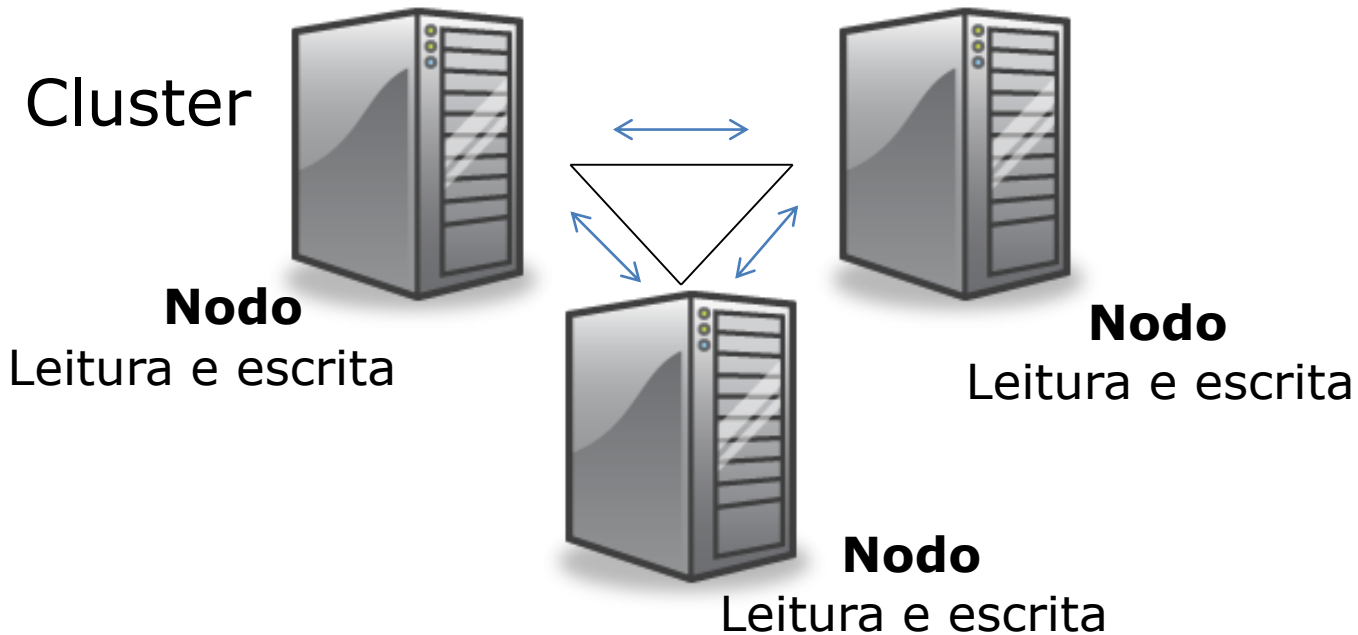


Consenso: Mesmo com falha em dois nós a leitura ainda pode ser realizada e com consistência ($R = 1$).

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Forte consistência e tolerância a particionamento (CP)

Situação em que necessita de forte consistência (leitura dos dados sempre atualizada).



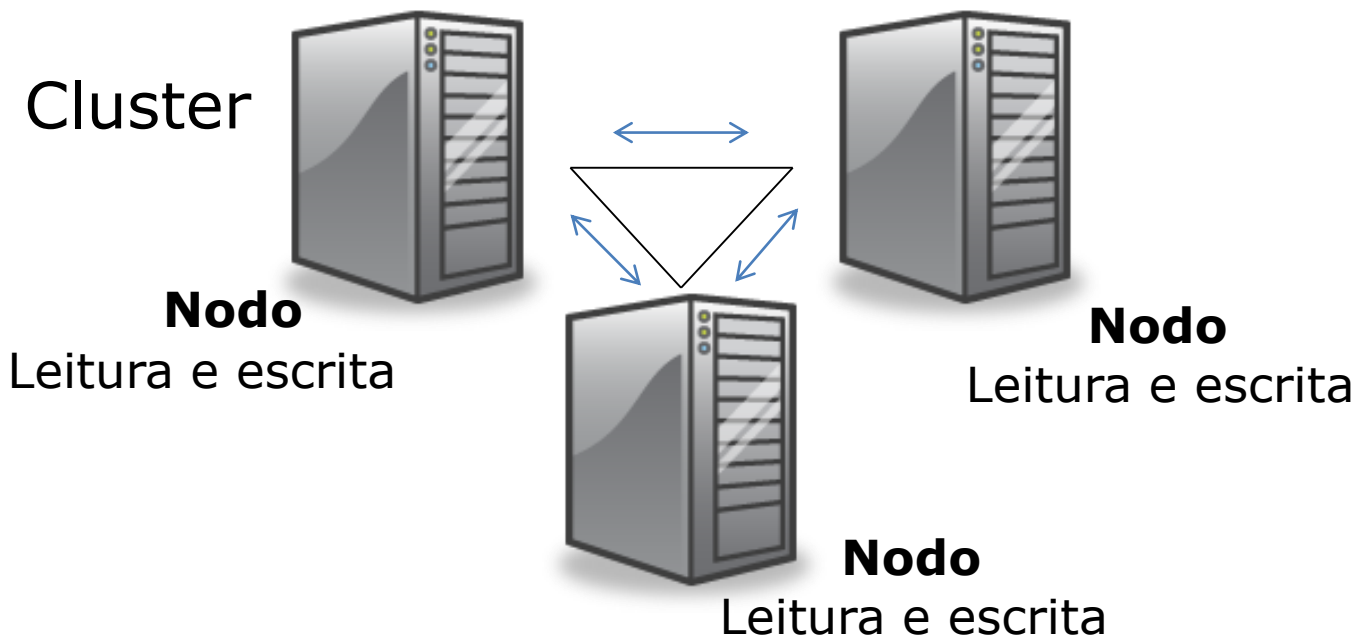
Exemplos de sistemas que implementa o conceito de consenso:

- MongoDB
- Google Big Table

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Alta disponibilidade e tolerância a particionamento (AP)

Situação em que deve-se garantir alta disponibilidade (de escrita).

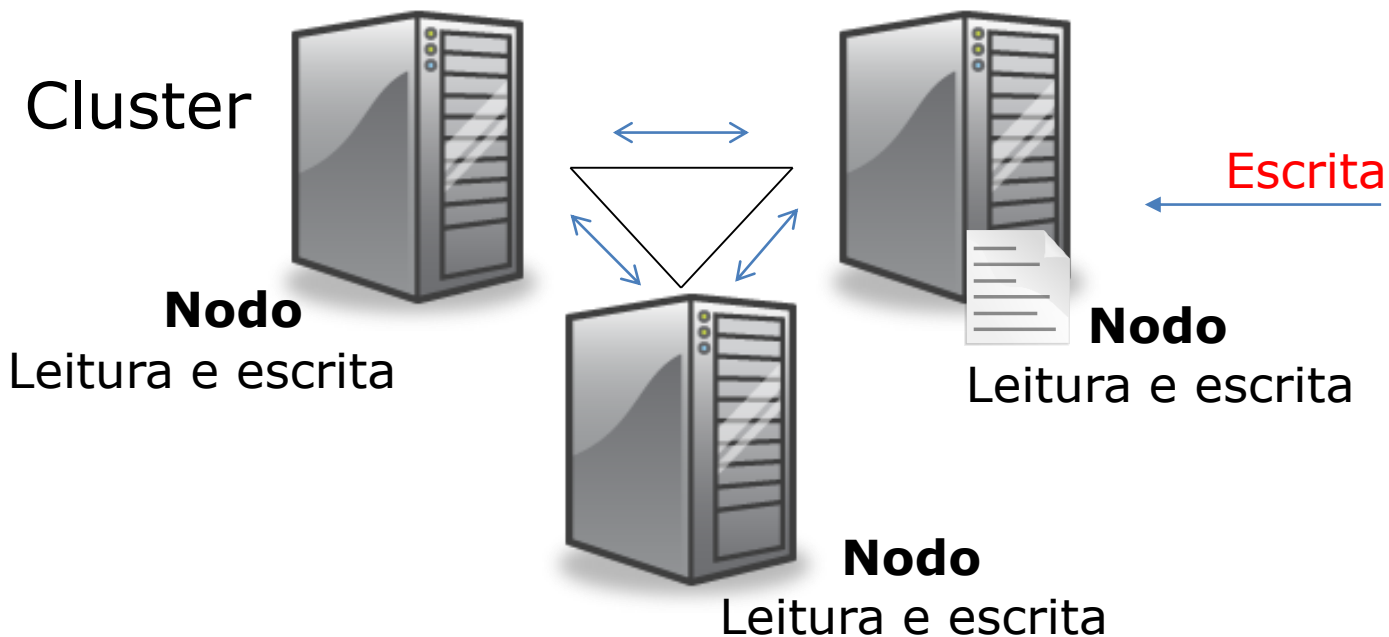


Eventual-consistency :
sempre disponível para
escrita e depois
sincroniza os dados com
os demais nós.

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Alta disponibilidade e tolerância a particionamento (AP)

Situação em que deve-se garantir alta disponibilidade (de escrita).

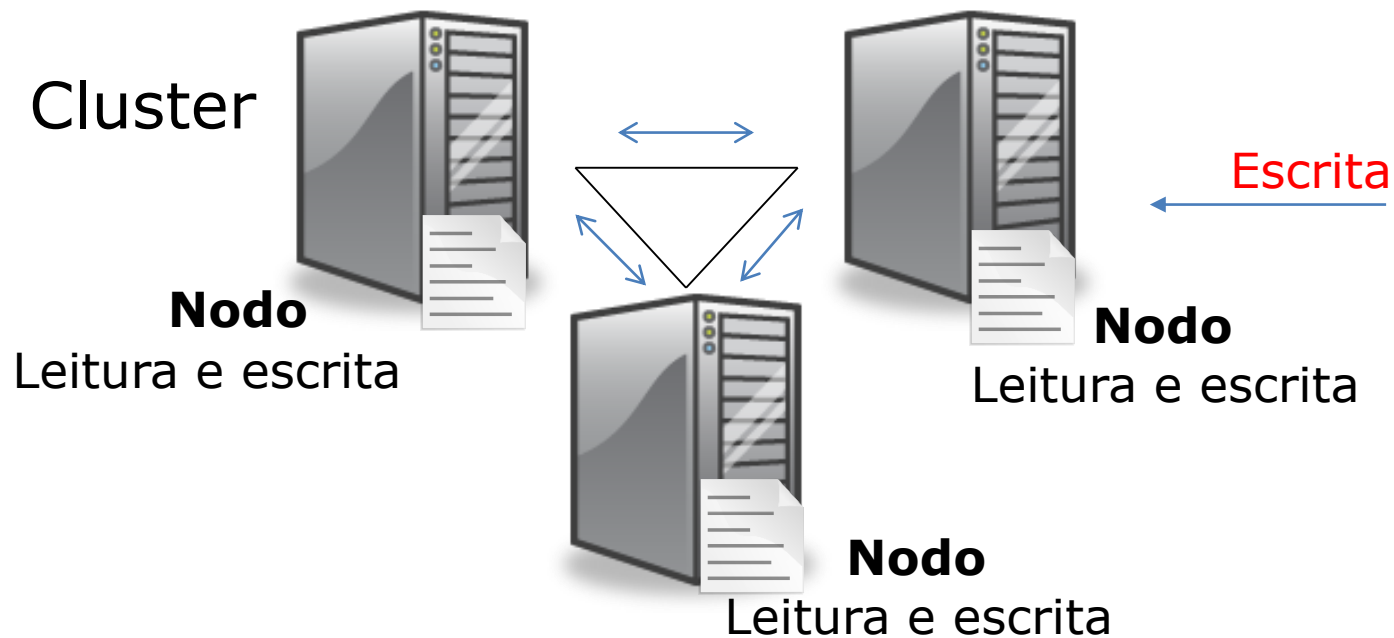


Na escrita os dados são replicados aos demais servidores no cluster

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Alta disponibilidade e tolerância a particionamento (AP)

Situação em que deve-se garantir alta disponibilidade (de escrita).

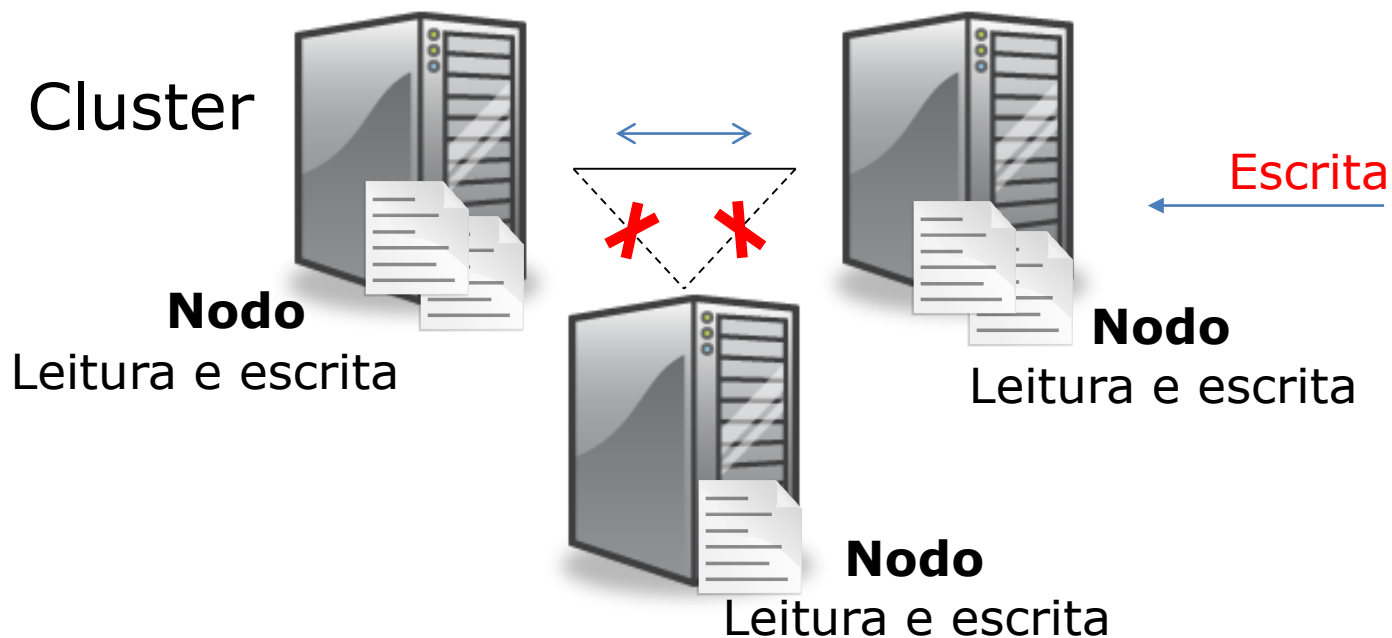


Na escrita os dados são replicados aos demais servidores no cluster

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Alta disponibilidade e tolerância a particionamento (AP)

Situação em que deve-se garantir alta disponibilidade (de escrita).

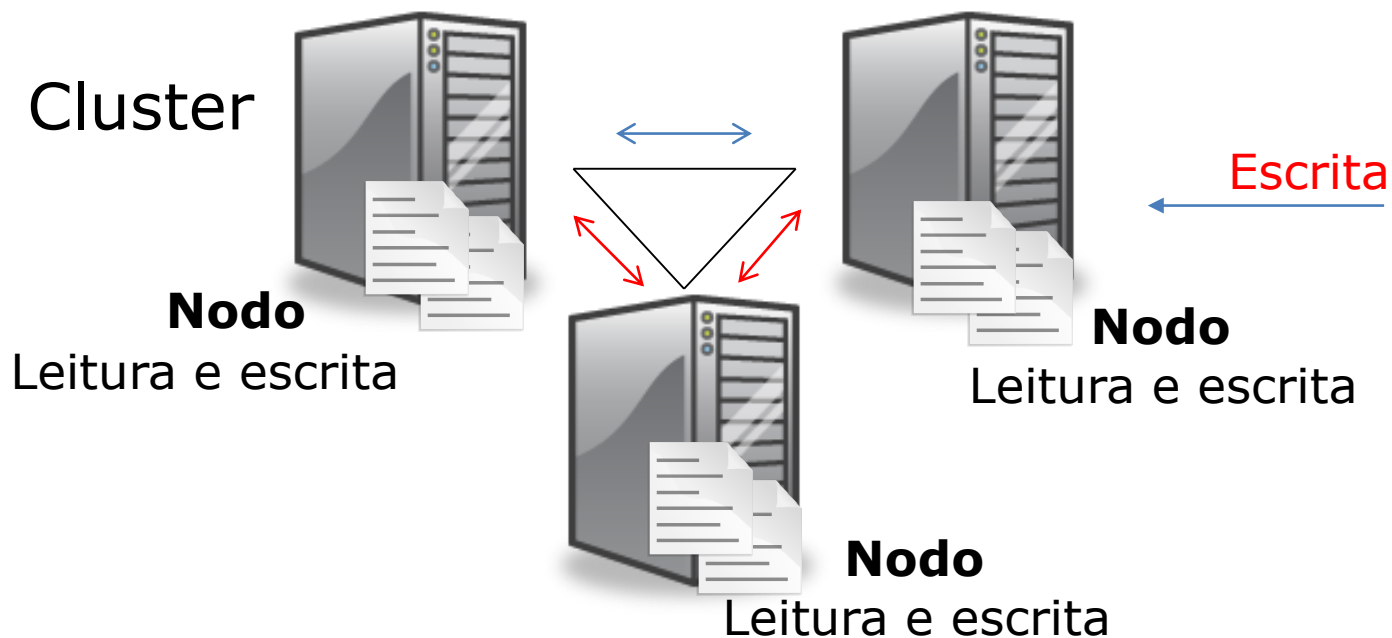


Se em uma escrita a comunicação com algum nó falhar, os dados são replicados apenas aos servidores disponíveis sem que o serviço fique indisponível (escrita ainda é possível).

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Alta disponibilidade e tolerância a particionamento (AP)

Situação em que deve-se garantir alta disponibilidade (de escrita).

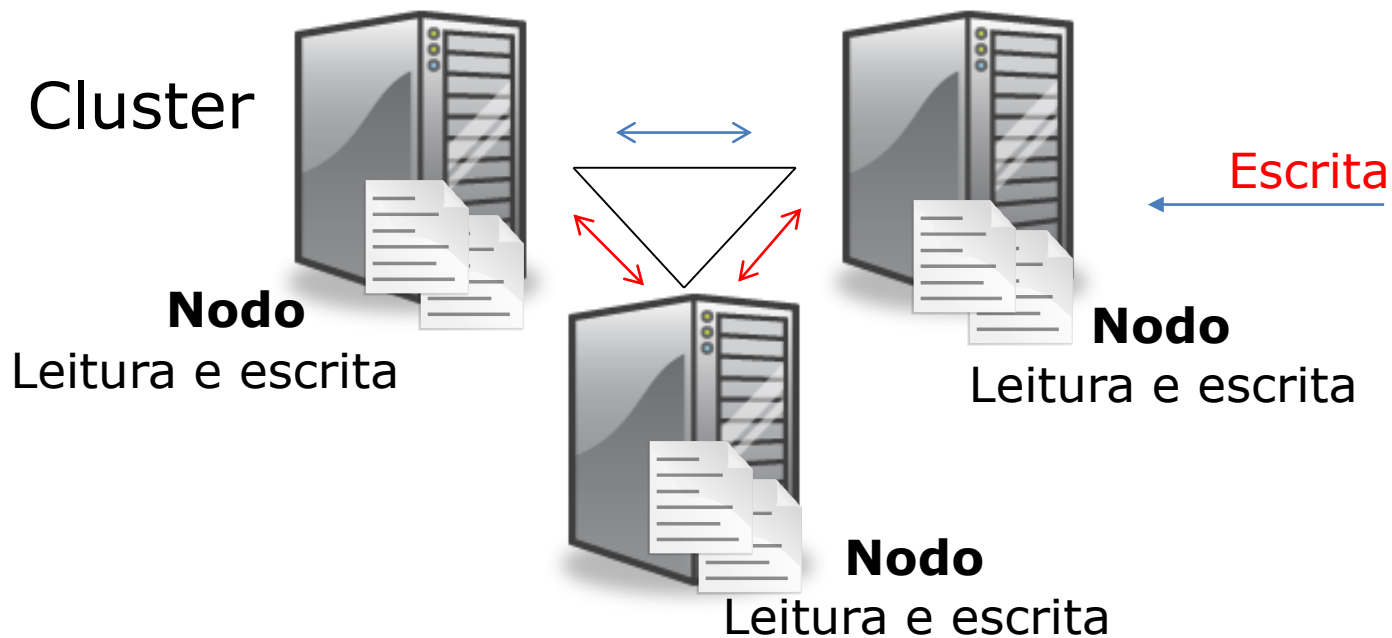


Após o reestabelecimento da comunicação, os dados são sincronizados com os demais servidores.

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Alta disponibilidade e tolerância a particionamento (AP)

Situação em que deve-se garantir alta disponibilidade (de escrita).

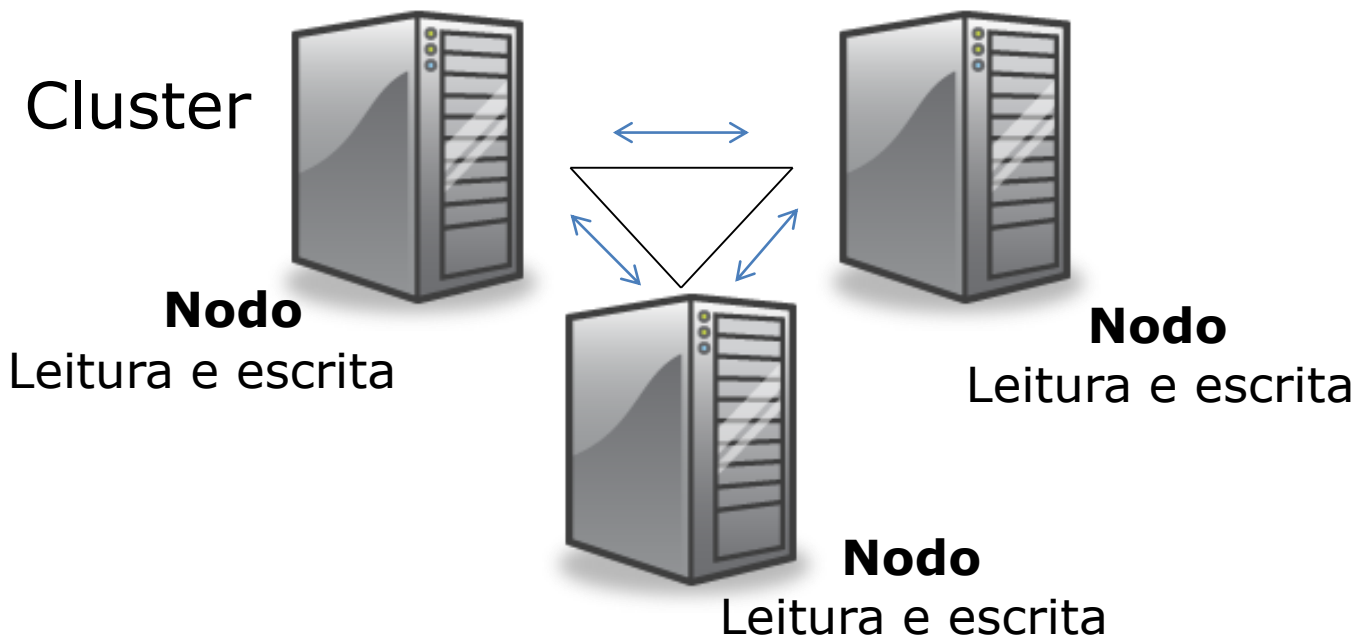


Observe que há uma janela de inconsistência durante o período anterior à sincronização dos dados (leitura desatualizada).

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Alta disponibilidade e tolerância a particionamento (AP)

Situação em que deve-se garantir alta disponibilidade (de escrita).



Exemplos de sistemas que implementa o conceito:

- Cassandra
- Riak
- DynamoDB

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

Forte consistência e alta disponibilidade (CA)

- Situação em que o sistema garante apenas alta disponibilidade (em um único nó). Uma máquina não pode ser particionada.

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

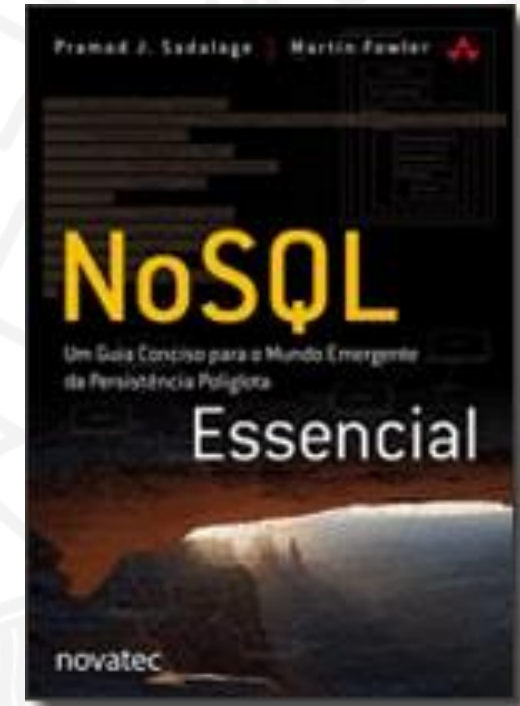
Forte consistência e alta disponibilidade (CA)

- Se houver alguma falha o sistema fica completamente indisponível até que os nós voltem a operar.
- Ex.: Banco de dados Relacionais.

Referências: Pramod J.; Sadalage, Martin Fowler. NoSQL Essencial. 2013

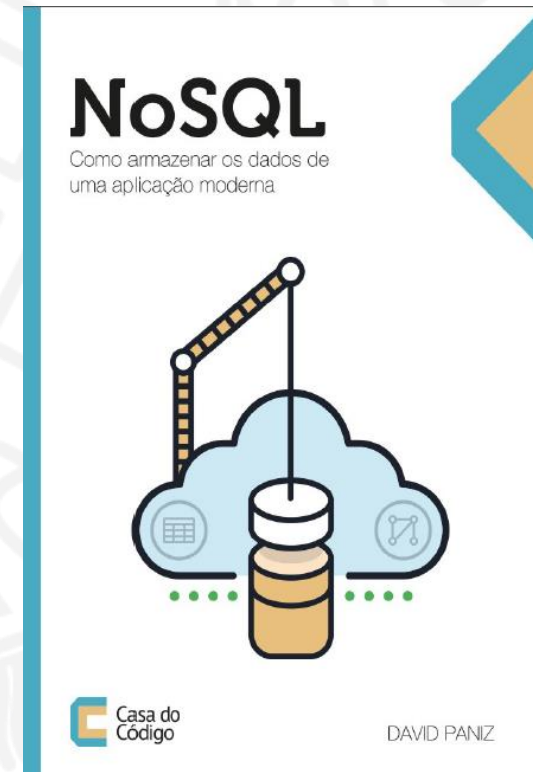
Principais Referências

Pramod J.; Sadalage, Martin Fowler.
**NoSQL Essencial: Um Guia Conciso
para o Mundo Emergente da
Persistência Poliglota.** Novatec
Editora, 2013.



Principais Referências

Paniz, David. NoSQL: **Como armazenar os dados de uma aplicação moderna**. Casa do Código, 2017.





PUC Minas
Virtual