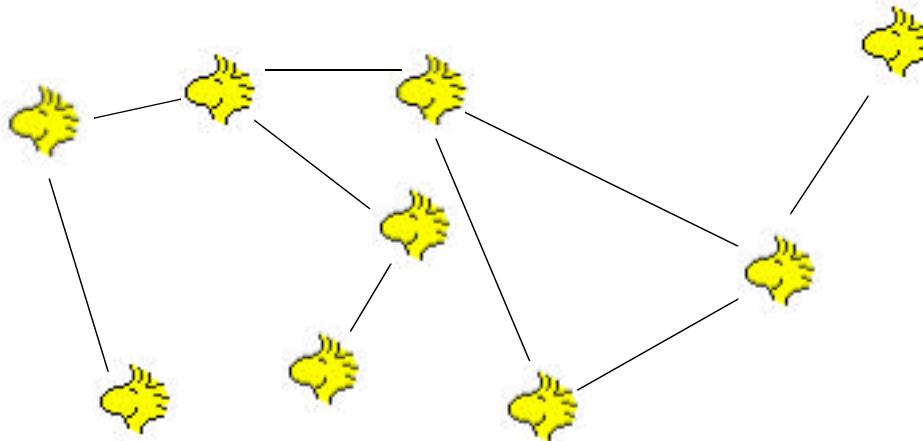


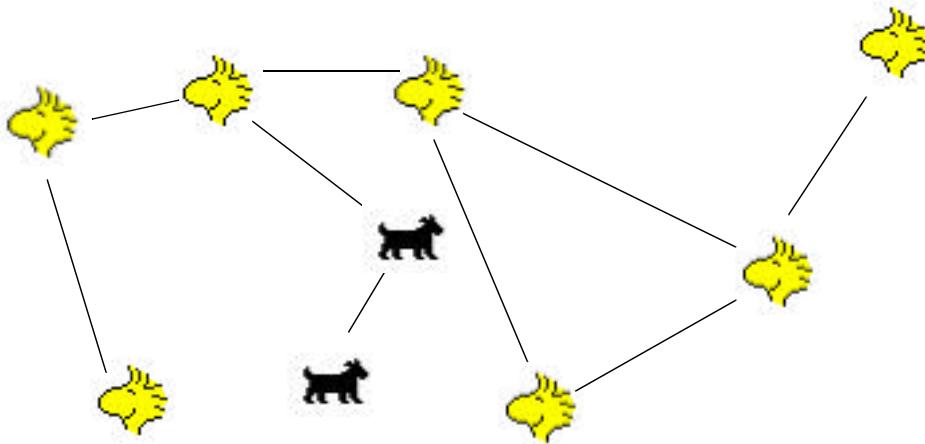
## Replicação de Dados



### Vantagens:

- Menor atraso no acesso: pode ser feito em cópia próxima
- Melhor desempenho, maior concorrência
- Tolerância a falhas: se uma cópia não puder ser acessada pode-se acessar alguma outra

# Replicação de Dados



## Desvantagens:

- Necessita de um controle para que não surjam inconsistências entre as cópias
- Ocupa mais espaço

# Replicação de Dados

Vamos considerar duas operações sobre os dados replicados:

- Leitura
- Escrita

Critério para verificação da consistência:

## *Serialização em cópia única*

“Execução de um conjunto de operações distribuídas deixa o dado replicado em um estado igual a se estas operações fossem aplicadas a uma única cópia do dado, em alguma ordem estritamente sequencial”.

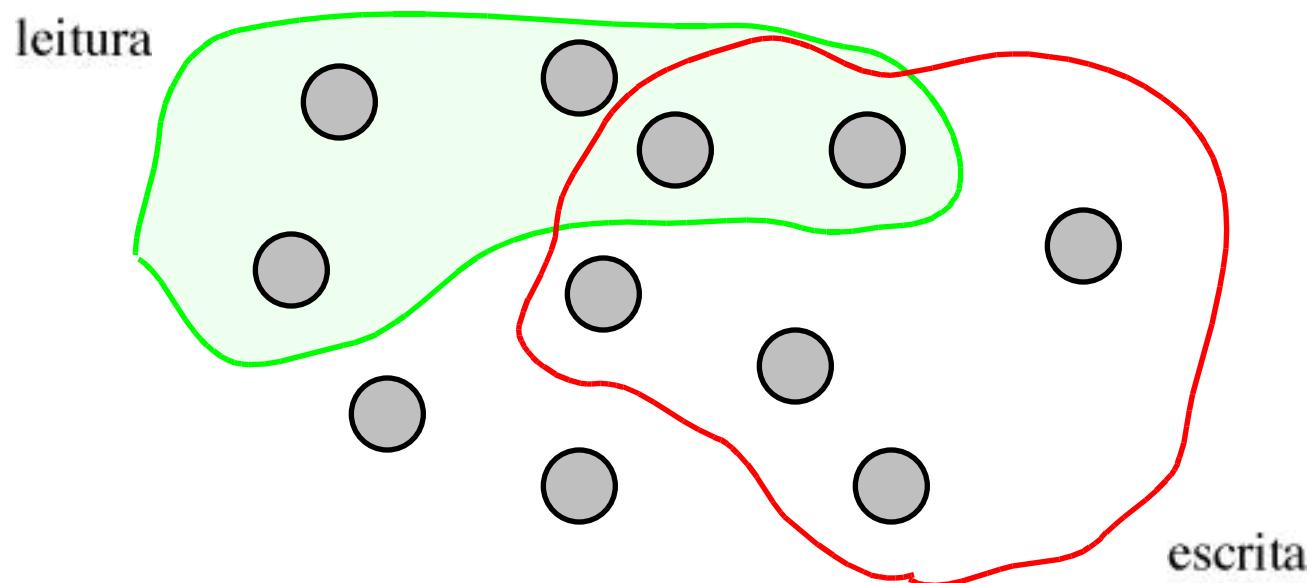
Vamos assumir que

- cópia é carimbada com número de versão, incrementado a cada escrita
- números de versão podem ser comparados para determinar qual a cópia mais recente

## Serialização em Cópia Única

Uma maneira de garantir que o critério de serialização em cópia única seja satisfeito é somente permitir operações de maneira que

- Qualquer leitura tenha intersecção não nula com qualquer escrita
- Quaisquer duas escritas tenham intersecção não nula



## **Replicação de Dados**

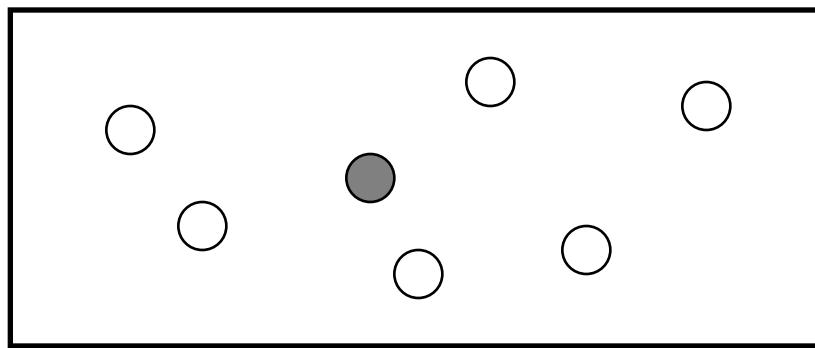
Protocolos são comparados quanto a

- grau de distribuição
- custo para cada operação (número de mensagens, número de cópias envolvidas)
- distribuição de carga entre as cópias (em termos de número de operações)
- disponibilidade do dado para uma dada operação (leitura, escrita)
  - depende da probabilidade de o nó estar operando corretamente (“disponibilidade do nó”)
  - dá uma medida da tolerância a falhas do protocolo

# Protocolos Tradicionais

## Cópia Primária

Stonebraker, 79



### Vantagem:

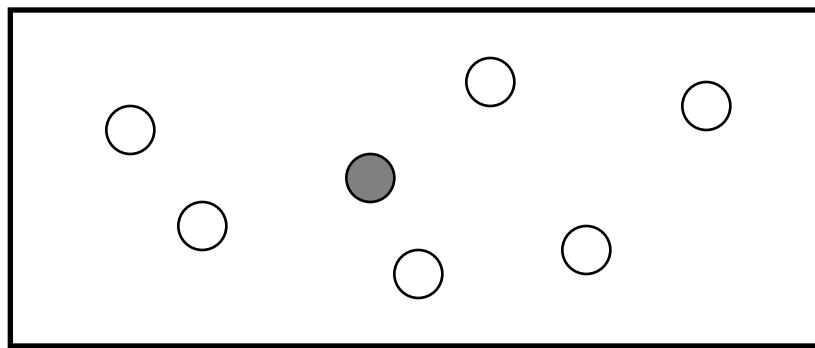
- Escreve e lê sempre de uma única cópia

### Problemas:

- Congestionamento
- Eleição de uma nova cópia primária em caso de falha

## Protocolos Tradicionais

Lê um, Escreve todos (**ROWA**)



Vantagem:

- Lê sempre de uma única cópia

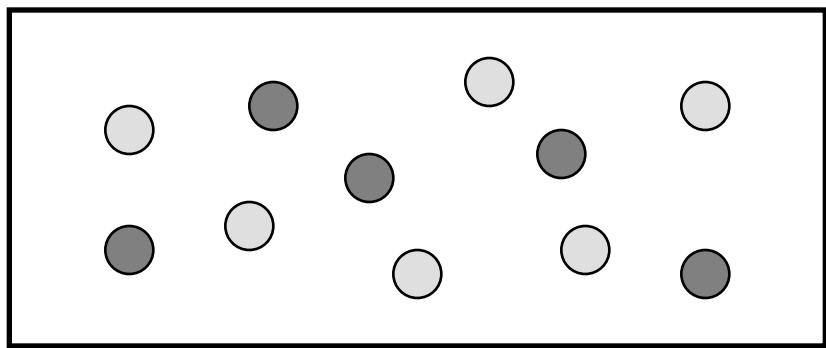
Problemas:

- Escrita muito cara
- Baixa tolerância a falhas para escrita

# Protocolos Tradicionais

## Votação Simples

Thomas, 79



- N cópias
- Quorum para efetuar operação:  
 $\lfloor N / 2 \rfloor + 1$

### Vantagem:

- Totalmente distribuído

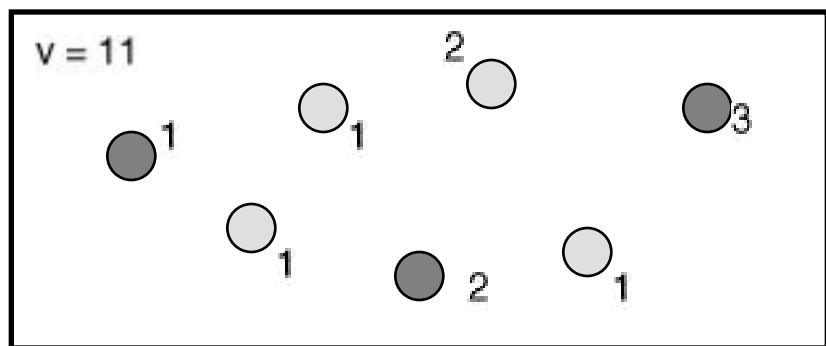
### Problemas:

- Degradação da disponibilidade à medida que nós falham
- Custo da leitura é relativamente alto

# Protocolos Tradicionais

## Votação Ponderada

Gifford, 79



Vantagem:

- Totalmente distribuído
- pode-se privilegiar nós mais robustos

Note: atribuindo-se um voto a cada cópia

- $r = 1, w = N \rightarrow$  ROWA
- $r = w = \lfloor N / 2 \rfloor + 1 \rightarrow$  votação simples

- $N$  cópias, votos diferenciados (total de votos =  $v$ )
- Quorums de leitura ( $r$ ) e escrita ( $w$ ):
  - $r + w > v$
  - $2w > v$

Problemas:

- Degradação da disponibilidade à medida que nós falham
- Quorums grandes

## Melhorando a Disponibilidade: Votação Dinâmica

Jajodia, S., Mutchler, M., "Dynamic voting algorithms for maintaining the consistency of a replicated database", ACM Trans. Database Systems, jun90

Idéia: formar quorums apenas a partir das cópias *correntes* dos dados

A cada cópia são associados dois valores

- cardinalidade
  - indica quantos nós participaram da última alteração
- nó privilegiado
  - usado em caso de desempate caso cardinalidade seja par

# Votação Dinâmica

## Exemplo

Suponha cinco cópias A, B, C, D e E ( $A < B < C < D < E$ )

	A	B	C	D	E
Versão	9	9	9	9	9
Card.	5	5	5	5	5
NóPriv.	-	-	-	-	-

C inicia escrita, detecta partição CDE:

	A	B	C	D	E
Versão	9	9	10	10	10
Card.	5	5	3	3	3
NóPriv.	-	-	-	-	-

## Votação Dinâmica

Exemplo (cont):

	A	B	C	D	E
Versão	9	9	10	10	10
Card.	5	5	3	3	3
NóPriv.	-	-	-	-	-

C inicia escrita, detecta partição CD:

	A	B	C	D	E
Versão	9	9	11	11	10
Card.	5	5	2	2	3
NóPriv.	-	-	C	C	-

C e D conseguem  
atualizar o dado  
(impossível em  
votação tradicional)

## Votação Dinâmica

Exemplo (cont.):

	A	B	C	D	E
Versão	9	9	12	11	10
Card.	5	5	1	1	3
NóPriv.	-	-	C	C	-

Problema: protocolo pode bloquear mesmo com maioria funcionando normalmente

Se C falhar, mesmo que todos os outros nós voltem a se comunicar, alterações ficam interditadas até a recuperação de C

## **Protocolos que utilizam estruturas lógicas**

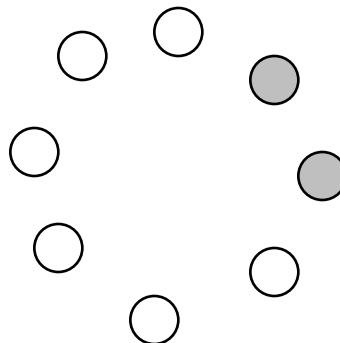
Para um grande número de cópias protocolos que utilizam votação tradicional são muito caros (quorums crescem linearmente com o número de cópias)

Idéia:

- organizar as cópias em uma estrutura lógica
- utilizar esta estrutura lógica para melhorar o custo do protocolo

## Protocolo em Anel

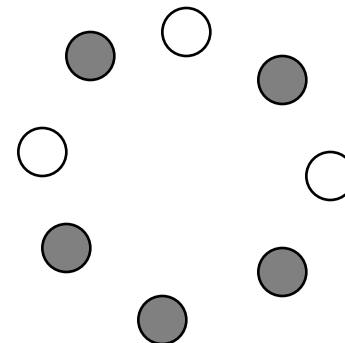
Cópias são organizadas logicamente em anel



Quorum de Leitura:

- quaisquer duas cópias adjacentes

Tamanho: 2 (fixo!)



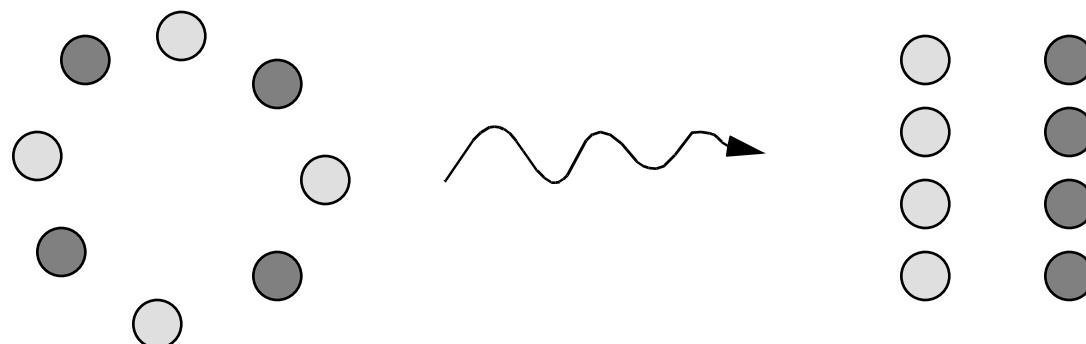
Quorum de Escrita:

- volta completa alternada (cópia sim, cópia não), mais um se N par

Tamanho:  $\lceil N / 2 \rceil + 1$

## Protocolo em Anel

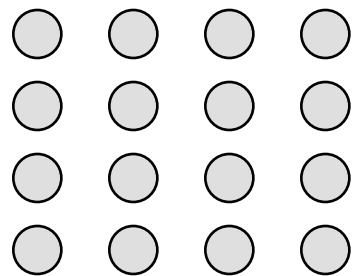
Uma outra maneira de olhar um anel



- **Quorum de leitura:** um de cada coluna
- **Quorum de escrita:** uma coluna inteira mais um de cada coluna restante

## Protocolo em Grade

Cheung, Ammar Ahmad, “A high performance scheme for maintaining replicated data”, *Proc.IEEE 6th Conf. Data Engineering*, 1990.



Tamanho do quorum:

- **Quorum de leitura:** pelo uma cópia de cada coluna
- **Quorum de escrita:** todas as cópias de pelo menos uma coluna mais um quorum de leitura

