Protocolo Hierárquico em Anel: Um Esquema Eficiente para Leitura de Dados Replicados

Nabor das Chagas Mendonça (nabor@dcc.unicamp.br)
Ricardo de Oliveira Anido (ranido@dcc.unicamp.br)
Depto. de Ciência da Computação
IMECC - UNICAMP
CxP 6065 - CEP 13081-970
Campinas - SP

Resumo

Votação é um mecanismo tradicionalmente usado para a manutenção da consistência de dados replicados em sistemas de computação distribuídos. Um dos problemas com o mecanismo de votação é o tamanho dos quorums necessários a cada acesso ao dado. Para reduzir o tamanho dos quorums, alguns protocolos de controle de dados replicados organizam as cópias em uma estrutura lógica, e utilizam essa estrutura na construção dos quorums.

Neste trabalho são apresentados dois protocolos para a manutenção da consistência de dados replicados. Ambos organizam as cópias em uma estrutura lógica de anel e usam as propriedades de adjacência características dessa estrutura para reduzir o tamanho dos quorums. O *protocolo simples em anel* usa um único anel e obtém um quorum de leitura de duas cópias (constante), e um quorum de escrita de maioria do total de cópias. O *protocolo hierárquico em anel*, uma generalização do primeiro protocolo, utiliza uma estrutura com múltiplos níveis de anéis. O quorum de leitura no protocolo hierárquico em anel é independente do total de cópias do dado e o quorum de escrita é, em geral, menor que a maioria do total de cópias. Os dois protocolos são tolerantes a falhas e não precisam de nenhum procedimento especial de reconfiguração quando as falhas ocorrem.

1 Introdução

A replicação de dados é uma técnica comumente usada para melhorar a confiabilidade de sistemas distribuídos. Manter cópias idênticas de um mesmo dado armazenadas em diferentes computadores do sistema evita que o dado fique inacessível (ou mesmo seja destruído) devido a falhas de algum deles, tornando o sistema mais confiável. Outra vantagem é um menor tempo de acesso para leitura, uma vez que a replicação aumenta a chance de o dado poder ser lido em uma cópia armazenada no próprio local que requisitou o acesso.

Uma notória desvantagem decorrente da replicação de dados, porém, é o alto custo de se manter a consistência entre as cópias, sendo necessário, para isso, que todos os acessos ao dado replicado sejam sincronizados. Deve existir, portanto, um protocolo que controle os acessos ao dado e que garanta a consistência entre suas cópias. Este protocolo deve ainda ser tolerante a possíveis falhas nos computadores do sistema ou na rede de comunicação.

O esquema de votação [18, 12] é a mais conhecida solução para a manutenção da consistência de dados replicados. Basicamente, ele consiste em somente permitir que o dado seja alterado se for obtida permissão da maioria simples de suas cópias [18], ou de um número de cópias que

represente a maioria simples da soma total dos votos atribuídos a cada uma (votação ponderada) [12]. O tamanho dos quorums de leitura e escrita podem variar, mas deve-se observar as restrições de que a soma dos quorums de leitura e escrita seja maior que o total de votos atribuídos às cópias, e de que o quorum de escrita seja no mínimo a maioria simples desse total. Outras variações foram propostas ao esquema de votação, tais como votação dinâmica [7, 13, 17], votação com testemunha [16], votação baseada em partições virtuais [1], votação com fragmentação [3] e votação multidimensional [5].

O maior problema com os protocolos que utilizam o esquema de votação é o tamanho dos quorums necessário a cada acesso ao dado replicado. Para contorná-lo, alguns trabalhos [2, 9, 14, 4] propuseram organizar as cópias do dado em uma estrutura lógica e tirar proveito dessa estrutura para reduzir o tamanho dos quorums. Em [9] as cópias são organizadas numa matriz (*grid*) de dimensões $\sqrt{n} \times \sqrt{n}$ (n é o número total de cópias) e, dessa forma, chega-se a quorums da ordem de $O(\sqrt{n})$; a votação em níveis de hierarquia feita em [14] necessita apenas de $n^{0.63}$ cópias para os quorums; e o protocolo de quorum em árvore de [4] consegue quorums de tamanho proporcional a $O(\log n)$ cópias.

Este trabalho propõe um novo esquema para manter a consistência de dados replicados que também utiliza uma estrutura lógica para formar os quorums. Aqui as cópias são organizadas logicamente em uma estrutura de anel, e a partir das informações de adjacências características dessa estrutura é possível obter um quorum de leitura consideravelmente pequeno sem produzir um quorum de escrita demasiadamente grande. São apresentados dois protocolos. No primeiro, o *protocolo simples em anel*, as cópias são agrupadas em um único anel. O quorum de escrita é a maioria alternada das cópias no anel e o de leitura é constante e igual a duas cópipas. A disponibilidade do dado, com um elevado número de cópias, é bastante alta para uma operação de leitura mas muito baixa para uma escrita. No segundo, o *protocolo hierárquico em anel*, uma generalização do protocolo simples, as cópias são organizadas em uma estrutura com múltiplos níveis de anéis, onde os anéis de um nível são formados por outros anéis no nível imediatamente inferior, e as cópias do dado compõem os anéis no nível mais baixo. O quorum de escrita depende do número de anéis (ou cópias do dado, no nível mais baixo) por nível e, em geral, é menor que a maioria do total de cópias. O quorum de leitura é exponencial na quantidade de níveis da hierarquia e independe do número total de cópias. Comparado ao protocolo simples, o protocolo hierárquico em anel propicia uma maior disponibilidade para escrita, com uma pequena redução na disponibilidade para leitura. Ambos protocolos são tolerantes a falhas e a eventuais particionamento da rede, possuindo ainda a vantajosa propriedade de, quando da ocorrência de falhas, não necessitarem de nenhum procedimento adicional de reconfiguração.

Na próxima seção é definido o modelo adotado. A seção 3 dá a motivação deste trabalho e descreve os dois protocolos em detalhe. Uma avaliação da disponibilidade do dado é feita na seção 4. A seção 5 compara o protocolo hierárquico de quorum em anel com outros trabalhos correlatos, e a seção 6 conclui o artigo.

2 Modelo Adotado

Um sistema distribuído consiste numa coleção de computadores independentes que se comunicam apenas pela troca de mensagens enviadas através de uma rede de comunicação. Cada computador, com sua respectiva memória local, é denominado um *nó* do sistema. Um nó pode ficar inacessível

devido a sua própria falha, falha de outros nós ou falha da rede de comunicação[1]. Falhas podem levar a uma situação de particionamento da rede, onde o sistema fica subdividido em grupos de nós denominados partições [10]. Nós de uma mesma partição conseguem comunicar-se, mas nós de partições diferentes ficam inacessíveis entre si.

Um dado lógico é replicado armazenando-se mais de uma cópia física em diferentes nós. São permitidas operações de leitura e escrita sobre os dados replicados, e o nó que requisitar uma dessas operações deve antes obter permissão de um número de cópias do dado (quorum) especificado pelo protocolo de controle.

Para representar os quorums será adotada a notação de *coterie* [11]. Seja S o conjunto de nós que possuem uma cópia de um dado replicado. Uma *coterie* sobre S é definida como um par de conjuntos $W = \{w_1, \ldots, w_n\}$ e $R = \{r_1, \ldots, r_m\}$ tal que seus elementos são subconjuntos de S e possuem as seguintes propriedades:

1. Minimalidade: não existe um par de elementos (a, b) em W (R) tal que a seja subconjunto de b; e
2. Intersecção: todo par de elementos em W tem uma intersecção não nula, e qualquer elemento de R tem uma intersecção não nula com todo elemento de W.

Os conjuntos W e R representarão, respectivamente, o quorum de escrita (q_w) e o quorum de leitura (q_r).

Para a prova de correção será adotado o critério de *one-copy serializability* [8], pelo qual, para se garantir a manutenção da consistência entre as cópias de um dado replicado, duas operações de escrita ou uma de escrita e outra de leitura não podem acontecer concorrentemente. Note-se que os quorums formados seguindo a definição de *coterie* satisfazem esse critério.

3 Os Protocolos em Anel

Esta seção dá as motivações que levaram à elaboração deste trabalho e descreve em detalhes o protocolo simples e o protocolo hierárquico.

3.1 Motivação

Todos os protocolos existentes para controle de dados replicados priorizam (ou dependem de) fatores como o custo das operações de leitura e escrita, a disponibilidade do dado e o grau de tolerancia a falhas. O esquema de leitura-em-um escrita-em-todos (*read-one write-all*), por exemplo, precisa de apenas uma cópia para formar o quorum de leitura, mas tem a incoveniente necessidade de um quorum de escrita formado por todas cópias do dado (não tolerando a falha de um único nó)[2].

Considerando-se ambientes onde os acessos para leitura superam os acessos para escrita, a principal motivação deste trabalho foi desenvolver um protocolo que mantivesse o quorum de leitura razoavelmente pequeno, sem exigir com isso um quorum de escrita demasiadamente grande.

Primeiramente será descrito o protocolo simples. O protocolo hierárquico, sendo uma generalização deste, será apresentado em seguida.

[1] Considera-se que os nós e os canais físicos (*links*) da rede de comunicação são *fail-stop*, ou seja, não comportam-se maliciosamente e ficam inoperantes somente devido a danos físicos.

[2] A comparação do protocolo proposto com outros existentes é feita na seção 5.

3.2 O Protocolo Simples em Anel

No protocolo simples, todas as cópias do dado replicado são organizadas logicamente em uma única estrutura de anel. Os quorums de leitura e escrita são obtidos através de algoritmos específicos de construção de quorum, os quais fazem uso das informações de adjacência entre as cópias, embutidas na estrutura em anel para garantir a intersecção entre os quorums e reduzir-lhes o tamanho.

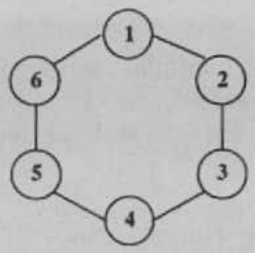

Figura 1: Seis cópias organizadas em uma estrutura de anel

Um quorum de leitura é formado obtendo-se permissão de acesso de quaisquer duas cópias adjacentes do anel. Um quorum de escrita deve obter permissão da metade das cópias alternadamente no anel e de mais uma outra cópia qualquer — perfazendo a maioria simples do total de cópias.

Como exemplo, considere-se que um dado replicado possui 6 cópias organizadas em anel (figura 1). São candidatos a quorum de leitura os seguintes conjuntos de nós: {1,2}, {2,3}, {3,4}, {4,5}, {5,6} e {6,1}. Os conjuntos candidatos a quorum de escrita são: {1,3,5,6}, {1,3,5,4}, {1,3,5,2}, {2,4,6,1}, {2,4,6,3} e {2,4,6,5}.

Note-se que os quorums gerados pelo protocolo simples satisfazem as propriedades de minimalidade e intersecção, e, portanto, constituem uma *coterie* (o fato de os quorums serem distintos e do mesmo tamanho satisfaz a propriedade de minimalidade; a satisfação da propriedade de intersecção será demonstrada mais adiante, na prova de correção do protocolo).

3.2.1 A Estrutura em Anel

Seja $S = \{s_1, \ldots, s_n\}$ o conjunto de nós que possuem uma cópia do dado replicado. Uma estrutura em anel é facilmente imposta nesse conjunto através da definição das operações de sucessor (Suc(i)) e predecessor (Pred(i)). Essas operações usam as informações de adjacência entre as cópias já existentes em S (s_1 precede s_2 e s_2 sucede s_1, por exemplo) mais o fato que, organizadas em um anel, s_1 sucederá s_n e s_n precederá s_1. Elas recebem como entrada a posição i ($1 \leq i \leq n$) de um nó $s_i \in S$, e devolvem a posição do sucessor ou predecessor de i, dependendo da operação (figura 2).

Outra operação definida sobre S é a de obtenção de permissão de acesso ao dado GetPermission(i). Essa operação retorna o valor $TRUE$ (sucesso) se o nó s_i permite que sua cópia do dado seja acessada pelo nó requisitante, ou $FALSE$ (fracasso) caso ele não permita o acesso ou o nó requisitante não consiga contactá-lo (devido à falha de s_i ou à ocorrência de um particionamento da rede).

```
Pre(pos)
{
  if (pos == 1)
    return(n)
  else
    return(pos-1)
}
```

```
Suc(pos)
{
  if (pos == n)
    return(1)
  else
    return(pos+1)
}
```

Figura 2: Operações de sucessor e predecessor no protocolo simples em anel.

```
GetReadQuorum()
{
    quorum = num_examined = 0
    copy = i (1 ≤ i ≤ n)
    while ((!quorum) && (num_examined < n)){
        if (GetPermission(copy)){
            copy = Suc(copy)
            quorum = GetPermission(copy)
            num_examined++
        }
        copy = Suc(copy)
        num_examined++
    }
    return(quorum)
}
```

Figura 3: Algoritmo para construção do quorum de leitura no protocolo simples em anel.

3.2.2 Construção dos Quorums

Para a formação de um quorum de leitura é suficiente ser obtida a permissão de acesso de qualquer par de cópias adjacentes no anel. O algoritmo de construção de quorum de leitura (figura 3) circulará pelo anel até que um quorum seja obtido ou que todas as cópias já tenham sido examinadas (nesse caso, o quorum não foi obtido e o acesso ao dado é negado).

O quorum de escrita é formado com a obtenção de permissão da maioria simples das cópias de forma tal que todo par s_i, s_j de cópias adjacentes no anel tenha pelo menos uma cópia incluída no quorum ($s_i \in q_w$ ou $s_j \in q_w$). O algoritmo é mostrado na figura 4.

3.2.3 Prova de Correção e de Não-equivalência com Associação de Votos

Conforme descrito na seção 2, para o protocolo estar correto é necessário provar que ele obedece ao critério de *one-copy serializability*, ou seja, que ele não permite duas operações conflitantes (duas escritas ou uma leitura e uma escrita) acontecerem em paralelo. Isso é feito provando-se os seguintes lemas.

Lema 1 *Em um anel de tamanho n, o protocolo simples em anel garante que há uma intersecção não nula entre qualquer quorum de leitura e qualquer quorum de escrita.*

Prova. A prova vem direto dos algoritmos de construção de quorum. O quorum de leitura é formado por duas cópias adjacentes no anel; o algoritmo de construção do quorum de escrita

```
Try(copy)
{
    quorum = fail = num_examined = 0
    while ((!fail) && (num_examined < ⌊n/2⌋)){
        if (GetPermission(copy)){
            copy = Suc(Suc(copy))
            num_examined++
        }
        else fail = 1
    }
    if (fail) return(0)
    else{
        if (!odd(n)) copy = Pred(copy)
        quorum = GetPermission(copy)
        return(quorum)
    }
}

GetWriteQuorum()
{
    quorum = num_examined = 0
    copy = i (1 ≤ i ≤ n)
    while ((!quorum) && (num_examined < n)){
        quorum = Try(copy)
        copy = Suc(copy)
        num_examined++
    }
    return(quorum)
}
```

Figura 4: Algoritmo para construção do quorum de escrita no protocolo simples em anel.

garante que para todo par de cópias adjacentes ao menos uma estará incluída em seu respectivo quorum. Portanto, sempre haverá uma intersecção não nula entre qualquer quorum de leitura e qualquer quorum de escrita. □

Lema 2 *Em um anel de tamanho n, o protocolo simples garante que há uma intersecção não nula entre quaisquer quorums de escrita.*

Prova. A prova vem direto do fato que todo quorum de escrita construído pelo protocolo simples é formado pela maioria simples ($\lfloor \frac{n}{2} \rfloor + 1$) das cópias do anel. Como $2(\lfloor \frac{n}{2} \rfloor + 1) > n$, é garantido que quaisquer quorums de escrita terão uma intersecção não nula. □

Uma propriedade interessante do protocolo simples em anel é que a *coterie* gerada por ele não pode ser gerada por nenhuma associação de votos no protocolo de votação [12].

Lema 3 *Não existe uma associação de votos equivalente ao protocolo simples.*

Prova. Por contradição. Considere-se o anel representado na figura 1. Seja $v_1, \ldots, v_6$ a associação de votos atribuída às 6 cópias do dado e V_t a soma total dos votos. Considere-se os dois conjuntos de cópias candidatos a quorum de escrita {1,3,4,5} e {1,2,4,6}, e outros dois conjuntos {1,2,3,4} e {1,4,5,6} que não formariam quorum algum (de acordo com o protocolo simples). Para a associação de votos atribuída às cópias ser equivalente ao protocolo simples, as seguintes inequações devem ser satisfeitas:

$$\left.\begin{array}{lcll} v_1 + v_3 + v_4 + v_5 & > & V_t/2 & (1) \\ v_1 + v_2 + v_4 + v_6 & > & V_t/2 & (2) \end{array}\right\} \text{formam quorum}$$

$$\left.\begin{array}{lcll} v_1 + v_2 + v_3 + v_4 & < & V_t/2 & (3) \\ v_1 + v_4 + v_5 + v_6 & < & V_t/2 & (4) \end{array}\right\} \text{não formam quorum}$$

Resolvendo-se (1) e (3) conclui-se que $v_2 < v_5$. Por (2) e (4) conclui-se que $v_5 < v_2$, criando uma contradição. Portanto, não existe nenhuma associação de votos (inteiros positivos) que satisfaça essas duas condições, o que valida o lema. □

3.3 O Protocolo Hierárquico em Anel

O protocolo hierárquico é uma generalização do protocolo simples descrito na seção anterior. Aqui a estrutura usada é uma uma hierarquia com múltiplos níveis de anéis, onde os elementos dos anéis no nível mais baixo são as cópias do dado, e os elementos dos anéis em um nível mais alto são anéis do nível imediatamente abaixo. Anéis em níveis diferentes podem ter números diferentes de elementos, mas anéis em um mesmo nível têm sempre o mesmo tamanho — o número total de cópias do dado será o produtório do número de elementos por anel em cada um dos níveis (figura 5).

A construção dos quorums é similar à do protocolo simples, mas agora acontece nível a nível, do nível mais baixo para o mais alto. Um quorum de leitura, por exemplo, é formado se for obtida permissão de dois elementos adjacentes do anel no nível mais alto. Obter permissão de um elemento, nesse caso da leitura, significa formar um quorum de leitura no anel correspondente a esse elemento. Este processo é repetido até que se chegue ao primeiro nível, onde a permissão é pedida diretamente às cópias do dado. Situação análoga ocorre na construção do quorum de escrita.

Figura 5: Uma estrutura hierárquica de anéis.

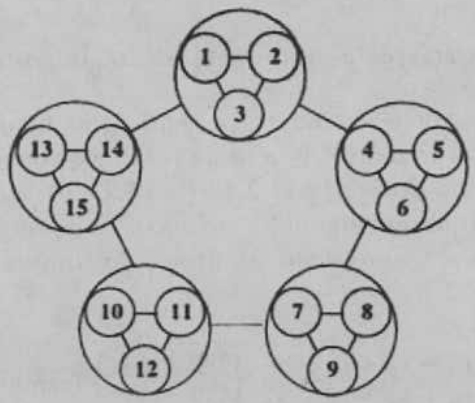

Figura 6: 15 cópias organizadas em uma hierarquia de anéis com 2 níveis.

Tome-se como exemplo a hierarquia representada na figura 6. Neste caso, o dado replicado possui 15 cópias que estão organizadas numa hierarquia de dois níveis de anéis. No primeiro nível, as cópias estão agrupadas em anéis com 3 elementos (cópias) cada. No segundo, o anel maior é formado por 5 anéis no nível inferior. Alguns conjuntos de cópias candidatos a quorum de leitura são: {1,2,13,14}, {2,3,4,5} e {7,8,11,12}, entre 45 outros possíveis. Como candidatos a quorum de escrita tem-se os conjuntos {1,2,7,8,10,11}, {4,5,11,12,14,15} e {7,9,13,15,2,3}, entre 135 outros possíveis. Note-se que os conjuntos candidatos a quorum no protocolo hierárquico também constituem uma *coterie*. Vale ressaltar, como pode ser observado nesse exemplo e considerando o quorum de leitura igual a 4, que o quorum de escrita obtido ($q_w = 6$) é menor que o mínimo permitido no esquema de votação tradicional — no esquema de votação, com 15 cópias e um quorum de leitura de tamanho 4, seria necessário um quorum de escrita de pelo menos 12 cópias.

3.3.1 A Estrutura Hierárquica em Anel

Seja $S = \{s_1, \ldots, s_n\}$ o conjunto de nós que possuem uma cópia do dado replicado. Uma hierarquia de anéis imposta em S consiste numa estrutura lógica de L níveis onde as cópias do dado estejam no nível 0, e elementos lógicos (anéis) sejam definidos nos níveis mais altos. Um elemento no nível 1 é um anel formado por m_1 cópias. Um elemento no nível 2 é um anel formado por m_2 elementos no nível 1. Generalizando-se, um elemento no nível i é um anel formado por m_i elementos no nível $i-1$, com $1 \leq i \leq L$. O número total de cópias (n) é dado por $\prod_{i=1}^{i=L} m_i$.

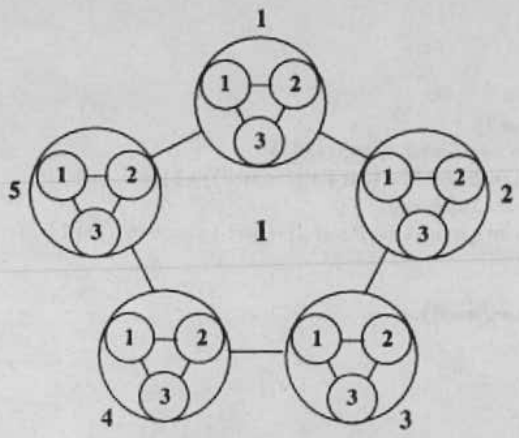

Figura 7: As posições dos elementos em relação ao próximo nível mais alto.

```
Pre(pos,size)
{
    if (pos == 1)
        return(size)
    else
        return(pos-1)
}
```

```
Suc(pos,size)
{
    if (pos == size)
        return(1)
    else
        return(pos+1)
}
```

Figura 8: Operações de sucessor e predecessor no protocolo hierárquico em anel.

Para simplificar a descrição dos algoritmos de construção dos quorums, será usada a seguinte notação: a posição de um elemento em um nível i é definida univocamente apenas em relação ao elemento no nível $i+1$ ao qual ele pertence (figura 7). Essa notação simplificará a forma de um elemento em um nível qualquer da hierarquia ser referenciado pelos algoritmos. Um elemento de um anel em um nível i, por exemplo, será referenciado pela sua posição p ($1 \leq p \leq m_i$) dentro desse anel.

Como no protocolo hierárquico pode haver anéis com diferentes números de objetos, aqui as operações de sucessor e predecessor são ligeiramente modificadas e têm o tamanho do anel como um parâmetro a mais de entrada (figura 8). A operação GetPermission(i) permanece a mesma, com $1 \leq i \leq n$.

3.3.2 Construção dos Quorums

No protocolo hierárquico em anel os quorums são construídos iterativamente, nível a nível. A construção dos quorums em cada nível da hierarquia de anéis é similar à construção dos quorums no protocolo simples. Um quorum de leitura em um nível i é formado pela construção com sucesso de dois quorums de leitura em dois elementos adjacentes no nível $i-1$. Um quorum de escrita, da mesma forma, é formado pela construção com sucesso de outros quorums de escrita na maioria alternada dos elementos no nível $i-1$.

Os algoritmos de construção de quorum de leitura e escrita são definidos recursivamente. Um

```
ReadQuorum(level,first)
{
   if (level > 0){
      quorum = num_examined = 0
      element = i (1 ≤ i ≤ m[level])
      while ((!quorum) && (num_examined < m[level])){
         if (ReadQuorum(level-1,(m[level-1]*(first+element-2))+1)){
            element = Suc(element,m[level])
            quorum = ReadQuorum(level-1,(m[level-1]*(first+element-2))+1)
            num_examined++
         }
         element = Suc(element,m[level])
         num_examined++
      }
      return(quorum)
   }
   else return(GetPermission(first))
}

GetReadQuorum()
{
   return(ReadQuorum(L,1))
}
```

Figura 9: Algoritmo para construção do quorum de leitura no protocolo hierárquico em anel.

pedido de permissão de acesso a um elemento em um nível i é implementado através de uma chamada recursiva do algoritmo para os elementos correspondentes no nível $i-1$. No nível 0, o pedido é feito diretamente às cópias do dado. Dois parâmetros são necessários: o nível corrente e a posição global (em relação ao nível L) do primeiro elemento no nível corrente. O primeiro parâmetro identifica o nível no qual o algoritmo está correntemente tentando construir o quorum. O segundo é usado para calcular a posição global de uma cópia no nível 0 e será o argumento da operação GetPermission(i). As figuras 9 e 10 mostram os algoritmos de contrução dos quorums de leitura e escrita.

3.3.3 Tamanho dos Quorums

Seja L o número de níveis da hierarquia, m_i $(1 \leq i \leq L)$ o número de elementos no nível $i-1$ que compõem um elemento no nível i, e $n = \prod_{i=1}^{i=L} m_i$ o número total de cópias do dado. O tamanho dos quorums de leitura e escrita do protocolo hierárquico é dado por:

$$q_r = 2^L \qquad e \qquad q_w = \prod_{i=1}^{i=L} \lfloor \frac{m_i}{2} \rfloor + 1$$

Note-se que o quorum de leitura cresce exponencialmente em L e independe de n e dos valores de m_i. Portanto, para se reduzir o quorum de leitura deve-se minimizar L. Quanto maior forem os valores de m_i, ainda menor será a proporção q_r/n. Por outro lado, o quorum de escrita, que é igual ou menor que a maioria simples das cópias ($\prod_{i=1}^{i=L} \lfloor \frac{m_i}{2} \rfloor + 1 \leq \lfloor (\prod_{i=1}^{i=L} m_i)/2 \rfloor + 1$), diminui quando L aumnenta.

```
Try(element,level,first)
{
    if (level > 0){
        quorum = fail = num_examined = 0
        while ((!fail) && (num_examined < ⌊m[level]/2⌋)){
            if (WriteQuorum(level-1,(m[level-1]*(first+element-2))+1)){
                element = Suc(Suc(element,m[level]))
                num_examined++
            }
            else fail = 1
        }
        if (fail) return(0)
        else{
            if (!odd(m[level])) element = Pred(element,m[level])
            quorum = WriteQuorum(level-1,(m[level-1]*(first+element-2))+1)
            return(quorum)
        }
    }
    else return(GetPermission(first))
}

WriteQuorum(level,first)
{
    quorum = num_examined = 0
    element = i (1 ≤ i ≤ m[level])
    while ((!quorum) && (num_examined < m[level])){
        quorum = Try(element,level,first)
        element = Suc(element,m[level])
        num_examined++
    }
    return(quorum)
}

GetWriteQuorum()
{
    return(WriteQuorum(L,1))
}
```

Figura 10: Algoritmo de construção do quorum de escrita no protocolo hierárquico em anel.

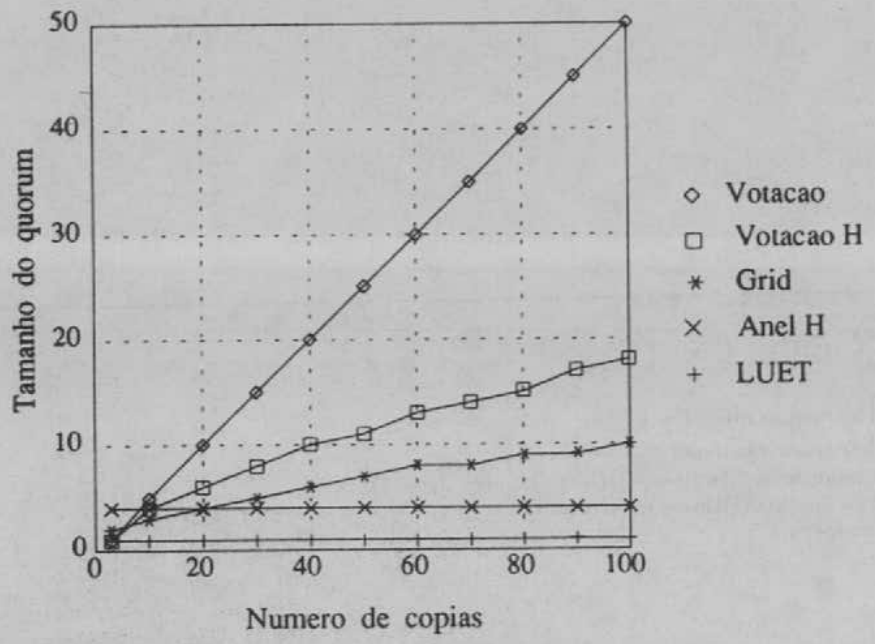

Figura 11: Tamanho esperado do quorum de leitura.

Considerando-se ambientes onde as operações de leitura acontecem em número maior que as de escrita, uma estratégia óbvia para diminuir os custos de comunicação é estabelecer um valor de L tão grande quanto for aceitável o quorum de leitura resultante. Outro fator a ser considerado no estabelecimento dos valores de m_i e L é a disponibilidade do dado. A disponibilidade no protocolo simples e no protocolo hierárquico será analisada na seção 4.

Um caso particular do protocolo hierárquico ocorre quando o número de elementos por anel é igual em todos os níveis. Seja $m_i = d$ $(1 \leq i \leq L)$. O número de níveis é dado por $L = \log_d n$ e os quorums resultantes são:

$$q_r = n^{\log_d 2} \qquad e \qquad q_w = (\lfloor \frac{d}{2} \rfloor + 1)^{\log_d n}$$

Nesse caso, fazendo-se, por exemplo, $d = \sqrt{n}$, tem-se $L = 2$, $q_r = 4$ e $q_w = \frac{n}{4} + \sqrt{n} + 1$ (para d par) ou $q_w = \frac{n+2\sqrt{n}+1}{4}$ (para d ímpar) — um quorum de leitura constante e um quorum de escrita assintoticamente bem menor que a maioria simples das cópias.

As figuras 11 e 12 mostram gráficos comparativos entre o protocolo hierárquico e o esquema de leitura-em-um escrita-em-todos (**LUET**), votação tradicional (**Votação**) [18], votação em níveis de hierarquia (**Votação H**) [14] e o protocolo *grid* (**Grid**) [9]. Considerando-se o aspecto do custo de formação de quorum de leitura em relação ao tamanho do quorum de escrita, e que o número de operações de leitura supera o de escritas, o protocolo hierárquico apresenta nítidas vantagens em comparação aos demais.

3.3.4 Prova de Correção e de Não-equivalência com Associação de Votos

Tal como no protocolo simples, deve ser provado aqui que os quorums de operações conflitantes gerados pelo protocolo hierárquico tem pelo menos uma cópia em comum. Isso é feito provando-se

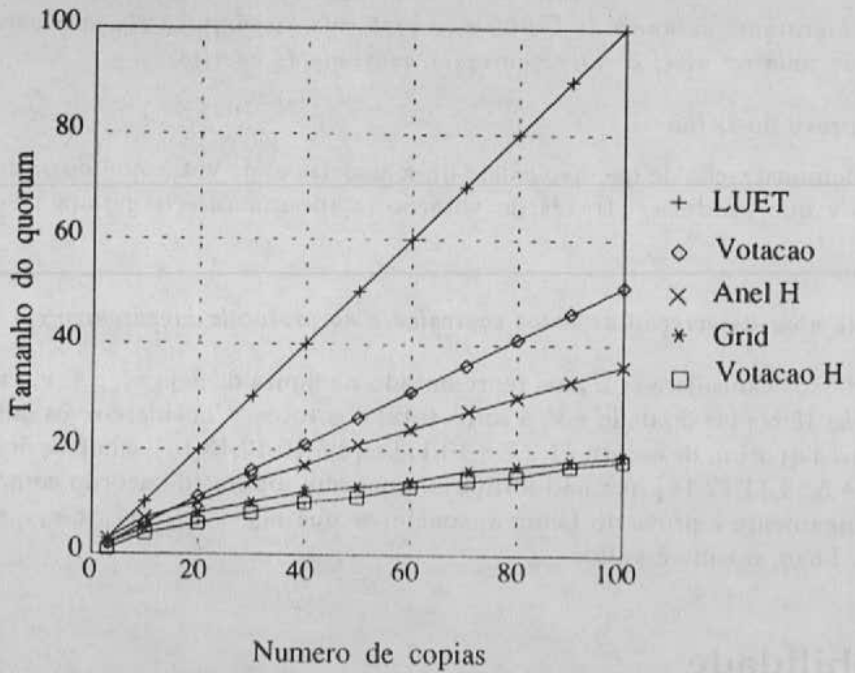

Figura 12: Tamanho esperado do quorum de escrita.

os seguintes lemas.

Lema 4 *Em um hierarquia de aneis de L níveis, o protocolo hierárquico em anel garante que há uma intersecção não nula no nível 0 entre qualquer quorum de leitura e qualquer quorum de escrita.*

Prova. Por indução nos níveis da hierarquia de anéis.

Base. Deve ser mostrado que o lema vale para o nível mais alto da hierarquia. Assim, deve ser provado o seguinte lema:

> **Lema 5** *Em um hierarquia de aneis de L níveis, o protocolo hierárquico em anel garante que há uma intersecção não nula no nível L entre qualquer quorum de leitura e qualquer quorum de escrita.*
> *Prova.* O nível L é o mais alto da hierarquia e contém um único anel com m_L elementos. Como os quorum do protocolo hierárquico são construídos similarmente aos do protocolo simples, a prova vem direto pelo Lema 1. □

Hipótese. Há uma intersecção não nula em um nível i ($1 \leq i \leq L$) da hierarquia de anéis entre qualquer quorum de leitura e qualquer quorum de escrita .

Passo. Deve ser mostrado que havendo uma intersecção entre os quorum de leitura e escrita no nível i, também haverá no nível $i-1$.

Havendo uma intersecção no nível i, deve haver pelo menos um elemento no nível $i-1$ em comum entre os quorums de leitura e escrita. Esse elemento pode ser uma cópia do dado no nível 0, o que já valida o lema, ou um anel, composto por m_i elementos no nível $i-2$, onde também foi obtido um quorum de leitura e escrita. No segundo caso, pelo Lema 1, deve haver um elemento no nível $i-2$ comum aos quorums e, portanto, haverá também uma intersecção não nula entre eles no nível $i-1$. Logo, o lema é válido. □

Lema 6 *Em uma hierarquia de aneis de L níveis, o protocolo hierárquico em anel garante que há uma intersecção não nula no nível 0 entre quaisquer quorums de escrita.*

Prova. Análoga à prova do Lema 4. □

Faz-se agora a demonstração de que não existe uma associação de votos que possa ser atribuída às cópias do dado e que produza, através de votação, a mesma *coterie* gerada pelo protocolo hierárquico.

Lema 7 *Não existe uma associação de votos equivalente ao protocolo hierárquico em anel.*

Prova. Por contradição. Considere-se o anel representado na figura 6. Seja $v_1, \ldots, v_{15}$ a associação de votos atribuída às 15 cópias do dado e V_t a soma total dos votos. Considere-se os dois conjuntos de cópias candidatos a quorum de escrita {1,2,7,8,10,11} e {4,5,10,12,13,14}, e outros dois conjuntos {1,2,7,8,10,13} e {4,5,10,11,12,14} que não formariam quorum algum (de acordo com o protocolo hierárquico). Analogamente à prova do Lema 3, conclui-se que $v_{13} < v_{11}$ e $v_{11} < v_{13}$, chegando-se numa contradição. Logo, o lema é válido. □

4 Disponibilidade

Nesta seção é analisada a probabilidade de se formar quorums, no protocolo simples e no protocolo hierárquico, considerando-se a confiabilidade dos nós.

Quando todos os nós do sistema estão em operação, qualquer quorum poderá ser formado por um nó requisitante (considerando-se que os outros nós não estejam comprometidos com alguma operação de escrita). Conforme as falhas vão surgindo, podem ocorrer situações em que o nó requerendo um acesso ao dado replicado não consegue formar o quorum correspondente. Tais situações devem ser evitadas pelo protocolo de controle das réplicas.

No protocolo simples, um quorum de leitura poderá ser formado enquanto houver pelo menos dois nós acessíveis adjacentes na estrutura em anel. O quorum de escrita, por sua vez, será formado enquanto não falharem dois nós adjacentes. Desse modo, no pior caso, o protocolo simples tolerará a falha de $\lceil \frac{n}{2} \rceil - 1$ nós (n é o número de cópias do dado) para a formação de um quorum de leitura, e de apenas 1 nó para um de escrita. No melhor caso, será tolerada a falha de $n - 2$ nós para uma operação leitura e de $\lceil \frac{n}{2} \rceil - 1$ para uma de escrita. Analogamente, o protocolo hierárquico, no pior caso, tolerará a falha de $\prod_{i=1}^{i=L} \lceil \frac{m_i}{2} \rceil - 1$ nós para uma operação de leitura e de $2^L - 1$ nós para uma de escrita. No melhor caso, será tolerada a falha de $n - 2^L$ nós para uma leitura e de $\prod_{i=1}^{i=L} \lceil \frac{m_i}{2} \rceil - 1$ para uma escrita.

A ***disponibilidade*** de um dado replicado é avaliada pelas chances de um quorum de leitura ou escrita conseguir ser formado, considerando-se a confiabilidade dos nós do sistema [6]. Assume-se que um nó esteja independentemente acessível com probabilidade ρ ($0 < \rho < 1$), e toma-se ρ como a medida da confiabilidade dos nós. Primeiramente são derivadas as expressões da disponibilidade para operações de leitura e escrita no protocolo simples. A disponibilidade para as operações no protocolo hierárquico são então expressas como recorrências baseadas na disponibilidade para as mesmas operações no protocolo simples.

Para ser expressa a disponibilidade do dado para operações de leitura e escrita no protocolo simples, é necessário antes ser calculada a probabilidade de algumas situações específicas acontecerem na estrutura em anel. Essas probabilidades de interesse, calculadas recursivamente, são:

- Probabilidade que dois nós acessíveis (bons) adjacentes existam em um anel de tamanho $n+1$, dado que um nó está inacessível (ruim):

$$P_r^{bb}(n,\rho) = \begin{cases} 0 & (n=0) \\ 0 & (n=1) \\ (1-\rho)P_r^{bb}(n-1,\rho) \; + \; \rho^2 & \\ + \; \rho(1-\rho)P_r^{bb}(n-2,\rho) & (n \geq 2) \end{cases}$$

- Probabilidade que dois nós acessíveis adjacentes existam em um anel de tamanho $n+1$, dado que um nó está acessível:

$$P_b^{bb}(n,\rho) = \begin{cases} \rho & (n=1) \\ \rho \; + \; \rho(1-\rho) & \\ + \; (1-\rho)^2 P_r^{bb}(n-2,\rho) & (n \geq 2) \end{cases}$$

- Probabilidade que dois nós inacessíveis adjacentes existam em um anel de tamanho $n+1$, dado que dois nós estão acessível e inacessível, respectivamente:

$$P_{br}^{rr}(n,\rho) = \begin{cases} 0 & (n=1) \\ 1-\rho & (n=2) \\ \rho P_{br}^{rr}(n-1,\rho) \; + \; (1-\rho)^2 & \\ + \; \rho(1-\rho)P_{br}^{rr}(n-2,\rho) & (n \geq 3) \end{cases}$$

- Probabilidade que dois nós acessíveis adjacentes existam e que dois nós inacessíveis adjacentes não existam em um anel de tamanho $n+1$, dado que um nó está inacessível:

$$P_r^{bb\wedge\neg rr}(n,\rho) = \begin{cases} 0 & (n=0) \\ 0 & (n=1) \\ \rho(1-\rho)P_r^{bb\wedge\neg rr}(n-2,\rho) & \\ + \; \rho^2(1-P_{br}^{rr}(n-1,\rho)) & (n \geq 2) \end{cases}$$

- Probabilidade que dois nós acessíveis adjacentes existam e que dois nós inacessíveis adjacentes não existam em um anel de tamanho $n+1$, dado que um nó está acessível:

$$P_b^{bb\wedge\neg rr}(n,\rho) = \begin{cases} \rho & (n=1) \\ \rho(1-P_r^{bb}(n-1,1-\rho)) & \\ + \; \rho(1-\rho)(1-P_{br}^{rr}(n-1,\rho)) & \\ + \; (1-\rho)^2 P_b^{bb\wedge\neg rr}(n-2,\rho) & (n \geq 2) \end{cases}$$

De posse dessas probabilidades, torna-se trivial o cálculo da disponibilidade do dado no protocolo simples. A disponibilidade para operações de leitura AF_r (a probabilidade que dois nós acessíveis adjacentes existam) em um anel de tamanho n é dada pela seguinte expressão:

$$AF_r(n,\rho) = \rho P_b^{bb}(n-1,\rho) \; + \; (1-\rho)P_r^{bb}(n-1,\rho).$$

Para que um quorum de escrita seja formado no protocolo simples não deve existir nenhum par de cópias inacessíveis adjacentes (para a metade alternada das cópias poder ser obtida), e deve haver também duas cópias acessíveis adjacentes (para completar a maioria das cópias). Assim, a disponibilidade do dado para operações de escrita no protocolo simples AF_w em um anel de tamanho n é dada por:

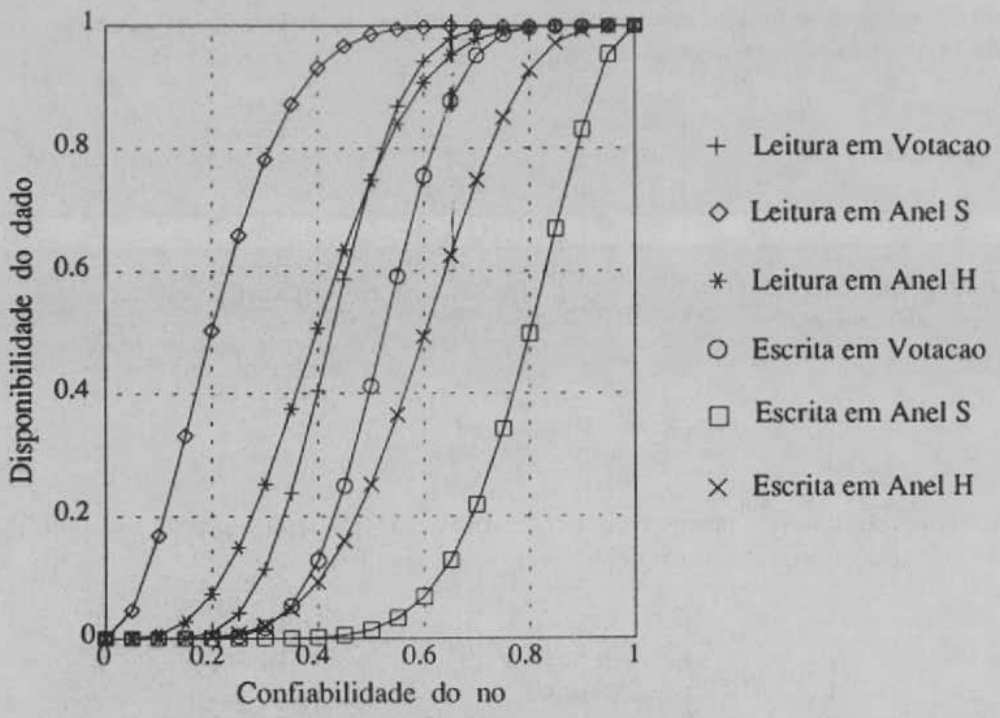

Figura 13: Disponibilidade de um dado com vinte cópias.

$$AF_w(n,\rho) = \rho P_b^{bb\wedge\neg rr}(n-1,\rho) \; + \; (1-\rho)P_r^{bb\wedge\neg rr}(n-1,\rho).$$

A disponibilidade no protocolo hierárquico é calculada nível a nível. A disponibilidade em um nível é expressa em termos da disponibilidade no nível imediatamente abaixo, e é calculada similarmente à disponibilidade no protocolo simples. Seja L o número de níveis da hierarquia, e m_i $(m_i \geq 2; 1 \leq i \leq L)$ o número de elementos do nível $i-1$ que compõem um elemento no nível i. A disponibilidade para operações de leitura no protocolo hierárquico AH_r é dada pela seguinte recorrência:

$$AH_r(i) = \begin{cases} \rho & (i=0) \\ AF_r(m_i, AH_r(i-1)) & (1 \leq i \leq L) \end{cases}$$

A disponibilidade para operações de escrita AH_w é calculada de modo análogo, e é dada por:

$$AH_w(i) = \begin{cases} \rho & (i=0) \\ AF_w(m_i, AH_w(i-1)) & (1 \leq i \leq L) \end{cases}$$

A disponibilidade do dado para uma operação de leitura no protocolo simples é bem maior que no esquema de votação, e tende rapidamente para 1 quando n é grande. A disponibilidade para escrita, entretanto, devido à restrição imposta pelo protocolo de o quorum ter de ser formado pela maioria alternada das cópias no anel, é baixa quando comparada ao esquema de votação e diminui mais ainda quando n aumenta.

A generalização para uma hierarquia de anéis, feita no protocolo hierárquico, propicia uma melhor relação entre a disponibilidade para leitura e a para escrita. Mantendo-se o número de

elementos por anel razoavelmente baixo, aumenta-se a disponibilidade para escrita e reduz-se a para leitura (e vice versa). Dependendo do tamanho aceitável dos quorums de leitura e escrita (decorrentes do número de elementos por anel e do número de níveis da hierarquia), pode-se chegar a uma disponibilidade para escrita comparável ao esquema de votação (para valores de p próximos a 1) e uma disponibilidade para leitura geralmente maior. A figura 13 mostra um gráfico comparativo da disponibilidade para operações de leitura e escrita no protocolo simples (**Anel S**), no protocolo hierárquico (**Anel H**) e no protocolo de votação tradicional (**Votação**). Foi considerado um exemplo de um dado com 20 cópias e fez-se, para o protocolo hierárquico, $L = 2$, $m_1 = 4$ e $m_2 = 5$.

5 Trabalhos Correlatos

Nesta seção o protocolo hierárquico é comparado a outros trabalhos correlatos existentes na literatura.

No esquema de votação tradicional [18], o quorum de escrita deve possuir maioria simples das cópias e o quorum de leitura deve ser o complemento do quorum de escrita mais 1. Por essas restrições, a redução do quorum de leitura implica diretamente no aumento do quorum de escrita. Fazendo-se o quorum de leitura igual a 1, o esquema de votação reduz-se ao caso extremo de leitura-em-um escrita-em-todos, onde a falha de um único nó já deixa o dado indisponível para escrita. No protocolo hierárquico em anel, o quorum de escrita é igual ou (em geral) menor que a maioria simples das cópias. O quorum de leitura é relativamente baixo comparado ao número total de cópias do dado e bem menor que o complemento do quorum de escrita.

Mantendo informações sobre a configuraçãodo sistema, o protocolo baseado em partições virtuais [1] consegue um quorum de leitura de apenas uma única cópia sem aumentar o quorum de escrita. Esse protocolo, porém, necessita de procedimentos (troca de mensagens) adicionais de reconfiguração quando ocorrem falhas para que as informações sobre a configuração do sistema em cada nó sejam atualizadas. O protocolo hierárquico em anel consegue um quorum de leitura reduzido (um mínimo de apenas duas cópias é possível) sem aumentar os custos de comunicação com procedimentos de reconfiguração — a estrutura lógica de anéis é suficiente para garantir a consistência das cópias mesmo quando da ocorrência de falhas.

Os outros protocolos que utilizam estruturas lógicas para organizar as cópias do dado replicado são [9, 14, 4]. Esses protocolos conseguem quorums menores que os do esquema de votação, e que não crescem linearmente com o número de total de cópias. No protocolo *grid* [9] as cópias são organizadas em uma matriz (*grid*) de dimensões $\sqrt{n} \times \sqrt{n}$ (n é o número total de cópias). O quorum de leitura é formado com a permissão de pelo menos uma cópia de cada uma das colunas e o de escrita deve obter um quorum de leitura e mais a permissão de todas as cópias de alguma linha. Os quorums de leitura e escrita, assim, são de tamanho $\sqrt{n}$ e $2\sqrt{n} - 1$, respectivamente. O protocolo de votação em níveis de hierarquia [14] utiliza uma árvore ternária na qual as cópias do dado são as folhas. Um nó lógico da árvore vota favoravelmente à formação de um quorum se obter o voto favorável da maioria simples de seus filhos. Repetindo-se esse processo até o nível mais alto da árvore, o quorum seria formado somente se for obtido o voto favorável da raiz. Os quorums de leitura e escrita são de mesmo tamanho e iguais a $n^{0.63}$. No protocolo de quorum em árvore [4], as cópias são organizadas em uma estrutura lógica de árvore e os quorums são formados obtendo-se permissão de nós que compõem um caminho específico nessa estrutura. No melhor caso, o quorum de escrita é proporcional a $O(\log n)$ cópias, e o quorum de leitura precisa de apenas uma única cópia correspondente à raiz da árvore.

No protocolo *grid* e no protocolo de votação em níveis de hierarquia, o quorum de leitura é reduzido mas cresce, embora não linearmente, conforme o número de cópias aumenta. Em [15] é proposto um protocolo *grid* hierárquico que, em relação ao protocolo *grid* original, melhora a disponibilidade mas mantém o mesmo tamanho dos quorums. No protocolo hierárquico em anel, o quorum de leitura é pequeno e, se o número de níveis da hierarquia não for alterado, permanece constante quando o número de cópias aumenta. Com $m_i = 3 (1 \leq i \leq L)$, o protocolo hierárquico em anel alcança os mesmos resultados da votação em níveis de hierarquia — incluindo tamanho de quorum e disponibilidade do dado.

O protocolo de quorum em árvore, num ponto de vista crítico, não pode ser considerado totalmente distribuído, uma vez que o papel desempenhado pelas cópias não é simétrico (a mesma crítica pode ser feita ao protocolo de votação ponderada [12] e suas variações e extensões). No protocolo de quorum em árvore, um quorum de leitura com uma única cópia apenas será formado se for obtida permissão de acesso da raiz. No protocolo hierárquico em anel, em contraste, as (poucas) cópias necessárias a um quorum de leitura podem ser quaisquer cópias adjacentes dentre todas as cópias do dado replicado.

A votação multidimensional [5] é uma outra abordagem para geração de *coteries* em sistemas distribuídos e, em particular, para o controle de dados replicados. Ela consiste em uma generalização da votação ponderada originalmente proposta em [12]. A cada cópia do dado é associado um vetor de dimensão k (k posições), e a cada posição do vetor é associado um número independente de votos. Os quorums são formados com a obtenção da maioria dos votos em l posições pré-determinadas do vetor. Em [5] é mostrado ainda que qualquer *coterie* possui uma associação de votos multidimensional equivalente. Assim, existe também uma associação de votos multidimensional equivalente à *coterie* gerada pelo protocolo hierárquico em anel. Com a votação multidimensional, entretanto, os nós não tomam conhecimento de suas posições dentro da estrutura lógica e, portanto, não podem tirar proveito dela para reduzir mais ainda o tamanho dos quorums.

6 Conclusão

Foi proposto um novo protocolo para o controle de dados replicados no qual o quorum de leitura é pequeno (podendo ser até constante) sem exigir um quorum de escrita demasiadamente grande. Primeiro foi apresentado o protocolo simples em anel, onde as cópias do dado replicado são organizadas em uma única estrutura de anel. O protocolo simples é então generalizado e é descrito o protocolo hierárquico em anel, onde as cópias são organizadas em uma estrutura com múltiplos níveis de anéis. Ambos protocolos usam as informações de adjacencia da estrutura em anel para manter a consitência entre as cópias e obter quorums menores. A disponibilidade propiciada pelo protocolo hierárquico em anel é comparável à do protocolo de votação tradicional para operações de escrita, e muito melhor para operações de leitura. Os protocolos são tolerantes a falhas e não precisam de nenhum procedimento adicional de reconfiguração. Considerando-se sistemas onde o número de operações de leitura notoriamente supera o de escritas, o protocolo hierárquico em anel possui destacadas vantagens em comparação a outros trabalhos na área.

Agradecimentos

Os autores agradecem Cláudio L. Lucchesi pela sua valiosa ajuda no cálculo das probabilidades envolvidas na análise da disponibilidade do dado.

Referências

[1] A. El Abbadi and S. Toueg. Maintaining Availability in Partioned Replicated Databases. *ACM Transactions on Database Systems*, pages 264–290, June 1989.

[2] D. Agrawal and A. El Abbadi. Exploiting Logical Structures of Replicated Databases. *Iformation Processing Letters*, 33(5):255–260, January 1990.

[3] D. Agrawal and A. El Abbadi. Storage Efficient Replicated Databases. Technical Report TRCS90-5, University of California, Department of Computer Science - Santa Barbara, 1990.

[4] D. Agrawal and A. El Abbadi. The Tree Quorum Protocol: An Efficient Approach for Managing Replicated Data. Technical Report TRCS90-5, University of California, Department of Computer Science - Santa Barbara, 1990.

[5] Mustaque Ahamad, Mostafa H. Ammar, and Shun Yan Cheung. Multidimensional Voting. *ACM Transactions on Computer Systems*, 9(4):399–431, November 1991.

[6] Daniel Barbara and Hector Garcia-Molina. The Reliability of Voting Mechanisms. *IEEE Transactions on Computers*, C-36(10):1197–1208, October 1987.

[7] Daniel Barbara, Hector Garcia-Molina, and Annemarie Spauster. Increasing Availability Under Mutual Exclusion Constraints with Dynamic Vote Reassignment. *ACM Transactions on Computer Systems*, pages 394–426, November 1989.

[8] Philip A. Bernstein and Nathan Goodman. The Failure and Recovery Problem for Replicated Databases. *Proceedings of the Second ACM Symposium on Principles of Distributed Computing*, pages 114–122, August 1983.

[9] S. Y. Cheung, M. Ammar, and M. Ahamad. The Grid Protocol: A High Performance Scheme for Maintaining Replicated Data. *Proceedings of the 6th IEEE International Conference on Data Engineering*, pages 438–445, 1990.

[10] Susan B. Davidson, Hector Garcia-Molina, and Dale Skeen. Consistency in Partioned Networks. *ACM Computing Surveys*, pages 341–370, September 1985.

[11] Hector Garcia-Molina and Daniel Barbara. How to Assign Votes in a Distributed System. *Journal of the ACM*, 32(4):841–860, October 1985.

[12] D. K. Gifford. Weighted Voting for Replicated Data. *Proceedings of the 7th Symposium on Operating Systems Principles*, pages 150–162, December 1979.

[13] Sushil Jajodia and David Mutchler. Dynamic Voting Algorithms for Maintaining the Consistency of a Replicated Database. *ACM Transactions on Database Systems*, pages 230–280, June 1990.

[14] Akhil Kumar. Hierarchical Quorum Consensus: A New Algorithm for Managing Replicated Data. *IEEE Transactions on Computers*, 40(9):996–1004, September 1991.

[15] Akhil Kumar and Shun Yan Cheung. A High Availability $\sqrt{N}$ Hierarchical Grid Algorithm for Replicated Data. *Iformation Processing Letters*, 40(6):311–316, December 1991.

[16] Jehan-François Pâris. Voting with Witnesses: A Consistency Scheme for Replicated Files. *Proceedings of the 6th IEEE Conference on Distributed Computing Systems*, pages 606–612, June 1986.

[17] Jin Tang and N. Natarajan. A Scheme for Maintaining Consistency of Replicated Files in Partioned Distributed Systems. *Proceedings of the 5th IEEE International Conference on Data Engineering*, pages 530–537, 1989.

[18] Robert H. Thomas. Majority Concensus Approach to Concurrency Control for Multiple Copy Databases. *ACM Transactions on Database Systems*, pages 180–209, June 1979.