

Algoritmos Probabilísticos

Departamento de Ciência da Computação – IME/USP
Segundo Semestre de 2006

ALGORITMO DE VALIANT

O *hipercubo* de dimensão n é o grafo cujos vértices são as seqüências de n bits e dois vértices são adjacentes se e somente se suas seqüências diferem em exatamente um bit. Seja $N = 2^n$. Este é o número de vértices do hipercubo de dimensão n .

Um algoritmo de roteamento, dados dois vértices de um grafo, determina o caminho que um pacote enviado de um para o outro percorre. Consideramos o modelo síncrono, onde cada pacote leva uma unidade de tempo para atravessar uma aresta e apenas um pacote pode atravessar uma aresta por unidade de tempo.

Uma permutação pode ser vista como um pedido de roteamento simultâneo de um pacote por vértice, em que cada pacote tem um destino diferente. A medida de eficiência de um algoritmo de roteamento é o número máximo de unidades de tempo que ele leva para rotear todos os pacotes de acordo com uma permutação.

O seguinte algoritmo decide deterministicamente como rotear um pacote do vértice a para o vértice b no hipercubo.

```
BITFIXING( $a, b$ )
1  para  $i \leftarrow 1$  até  $n$  faça
2      se  $a_i \neq b_i$ 
3          então atravesse a aresta de  $(b_1, \dots, b_{i-1}, a_i, a_{i+1}, \dots, a_n)$ 
              para  $(b_1, \dots, b_{i-1}, b_i, a_{i+1}, \dots, a_n)$ 
```

Esse algoritmo roteia qualquer pacote em no máximo n passos. No entanto, a eficiência dele não é boa no seguinte sentido. Para algumas permutações, o número máximo de unidades de tempo que esse algoritmo leva para rotear os pacotes de acordo com essas permutações é $\Omega(\sqrt{N})$.

Exercício: Descreva uma destas permutações. Justifique a sua resposta.

Valiant [1] propôs a seguinte idéia para evitar os problemas desse algoritmo determinístico. Em vez de rotear o pacote diretamente da origem para o destino, escolha aleatoriamente um vértice do hipercubo, envie o pacote para este vértice e de lá para o seu destino.

```
VALIANT( $a, b$ )
1  escolha  $c$  aleatoriamente de  $\{1, \dots, N\}$ 
2  BITFIXING( $a, c$ )
3  BITFIXING( $c, b$ )
```

A análise da eficiência do algoritmo de Valiant foi aprimorada por Brebner, juntamente com Valiant [2]. Lembre-se que a eficiência do algoritmo é medida pelo tempo máximo, dada uma permutação, de roteamento dos pacotes de acordo com a permutação.

Seja $P = (e_1, \dots, e_m)$ um caminho que pode ser o caminho de roteamento de um pacote determinado pelo algoritmo BITFIXING. Claro que $m \leq n$. Sejam $v_0, \dots, v_m$ os vértices de P , sendo que $e_i = v_{i-1}v_i$.

Seja p uma permutação de 1 a N . Para cada vértice a do hipercubo, seja $T(a)$ o número de unidades de tempo usadas para rotear o pacote que começa no vértice a . Seja $X_1(e)$ o número de pacotes que atravessam a aresta e durante o roteamento da permutação p numa chamada de BITFIXING da linha 2 do algoritmo de Valiant e $X_2(e)$ o mesmo para a chamada da linha 3.

Se $P = (e_1, \dots, e_m)$ é o caminho que roteia o pacote do vértice a até seu vértice intermediário, então $T(a) \leq \sum_{i=1}^m X_1(e_i) + \sum_{i=1}^m X_2(e_i)$. De fato, isso é consequência do fato de que, a cada unidade de tempo, um pacote ou atravessa uma aresta ou espera um outro pacote atravessar essa aresta.

Para um caminho $P = (e_1, \dots, e_m)$ que pode ser o caminho de roteamento de um pacote determinado pelo algoritmo BITFIXING, seja $T_j(P) = \sum_{i=1}^m X_j(e_i)$, para $j = 1, 2$. Queremos delimitar superiormente $T_1(P)$ e $T_2(P)$. As variáveis $X_j(e_i)$ não são independentes, por isso teremos que fazer isso indiretamente.

Vamos nos concentrar em $T_1(P)$, pois a análise de $T_2(P)$ é semelhante. Considere então apenas as execuções da chamada de BITFIXING da linha 2 do algoritmo de Valiant.

Sejam $v_0, \dots, v_m$ os vértices de P , sendo que $e_i = v_{i-1}v_i$. Dizemos que um pacote é *ativo* em v_{i-1} se ele passa por v_{i-1} antes de ter tentado consertar o seu j -ésimo bit, onde j é o bit em que v_{i-1} e v_i diferem. O número esperado de pacotes ativos em v_{i-1} é $\frac{2^j-1}{2^j-1} = 1$, pois existem 2^j origens para tais pacotes (apenas vértices cujos bits de j até n coincidem com v_{i-1} podem ser origem de tais pacotes) e, para cada uma destas origens, o vértice intermediário escolhido tem que ter os mesmos $j-1$ bits iniciais que v_{i-1} .

Dizemos que um pacote é *ativo* em P se ele é ativo em algum dos vértices de P . Seja H o número de pacotes ativos em P . Do que foi dito acima, deduzimos que $E[H] \leq m \leq n$.

Podemos deduzir de Chernoff que $\Pr[H \geq 6n] \leq 2^{-6n}$. Se provarmos que $\Pr[T_1(P) \geq 30n | H < 6n] \leq 2^{-3n-1}$, então podemos concluir que

$$\begin{aligned} \Pr[T_1(P) \geq 30n] &\leq \Pr[H \geq 6n] + \Pr[T_1(P) \geq 30n | H < 6n] \\ &\leq 2^{-6n} + 2^{-3n-1} \\ &\leq 2^{-3n}. \end{aligned}$$

Como existem $2^n 2^n = 2^{2n}$ possíveis caminhos P , a probabilidade de que um deles seja tal que $T_1(P) \geq 30n$ é no máximo $2^{2n} 2^{-3n} = 2^{-n} = 1/N$, ou seja,

$$\Pr[T_1(P) \geq 30n] \leq 1/N.$$

De maneira análoga, podemos mostrar que

$$\Pr[T_2(P) \geq 30n] \leq 1/N,$$

e desses dois fatos, podemos concluir que, para cada vértice a do hipergrafo,

$$\Pr[T(a) \geq 60n] \leq \Pr[T_1(P) + T_2(P) \geq 60n] \leq 1/N,$$

onde P é o caminho que o pacote que sai de a percorre no hipergrafo e a última desigualdade vale pois para a soma $T_1(P) + T_2(P)$ ser pelo menos $60n$, pelo menos um entre $T_1(P)$ e $T_2(P)$ tem que ser pelo menos $30n$.

REFERÊNCIAS

- [1] L.G. Valiant, A scheme for fast parallel communication. *SIAM Journal on Computing*, 11:350–361, 1982.
- [2] L.G. Valiant and G.J. Brebner, Universal schemes for parallel communication. *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*, pages 263–277, 1981.