Groupe de travail Réseau
Request for Comments n°70
Traduction Claude Brière de L'Isle

S. Crocker, UCLA
15 octobre 1970

Note sur le bourrage

Le bourrage d'un message est une chaîne de forme 10*. Pour les hôtes qui ont des longueurs de mot de 16, 32, 48, etc., bits, cette chaîne est nécessairement dans le dernier mot reçu de l'IMP. Pour les hôtes qui ont des longueurs de mot qui ne sont pas un multiple de 16 (mais qui font au moins 16 bits de long) le bit 1 va être soit dans le dernier mot, soit dans le mot après le dernier mot. Bien sûr, si le bit 1 est dans le mot qui suit le dernier mot, le dernier mot est tout à zéro.

Une tâche de codage déplaisante est de découvrir la position binaire du bit 1 au sein de son mot. Une technique évidente est de vérifier de façon répétée le bit de moindre poids, faisant glisser le mot d'une position binaire à droite si le bit de moindre poids est zéro. Les techniques suivantes sont plus agréables.

Isolation du bit de moindre poids

Soit W un mot non zéro, où la longueur du mot est n. Alors W est de la forme

```
x.........x10.........0
 \__ __/\__ __/
    V      V
  n-k-1    k
```

où $0 \leq k < n$

et les x sont des bits arbitraires.

Supposant une arithmétique de compléments à deux,

```
 W-1   =     x....x01....1
-W     =    x.......x10....0
 W     =    x.........x01....1
```

En utilisant ET, OU et OU exclusif avec diverses paires de ces quantités, de nouvelles formes utiles sont obtenues.

Par exemple,

```
 W ET W-1     xx....x00.......0
              \__ __/\__ __/
                 V      V
               n-k-1    k
```

donc en retirant le bit 1 de moindre poids ;

```
aussi W ET -W =        0....010....0
                     __ __/__ __/
                        V    V
                      n-k-1  k
```

isolant ainsi le bit de moindre poids.

Ci-dessous, on se concentre seulement sur ce dernier résultat ; cependant, dans une application particulière il peut être avantageux d'utiliser une variante.

Détermination de la position d'un bit isolé

Les deux techniques évidentes pour trouver la position binaire d'un bit isolé sont de le faire glisser de façon répétée avec des essais, comme ci-dessus, et d'utiliser un matériel de normalisation flottante. Sur le PDP-10, en particulier, l'instruction JFFO est faite pour ordonner*. Sur les machines avec une normalisation hexadécimale, par exemple. les IBM 360 et les XDS Sigma 7, le matériel de normalisation peut n'être pas très pratique. Une approche différente utilise la division et l'interrogation de tableaux.

Un mot avec un seul bit établi a une valeur d'entier non signé de 2^k pour $0 \leq k < n$. Si on choisit un p tel que mod(2^k ,p) soit distinct pour chaque $0 \leq k < n$, on peut faire un tableau de longueur p qui donne la correspondance entre mod(2^k ,p) et k. Le reste de cet article se consacre au choix d'un diviseur approprié p pour chaque longueur de mot n.

* Certaines des machines CDC ont une instruction "compte de population" qui donne le nombre de bits dans un mot. Noter que le 2^k -1 a exactement k bits à un.

Exemple

Soit n = 8 et p = 11
Alors

mod(2^0, 11) = 1
mod(2^1, 11) = 2
mod(2^2, 11) = 4
mod(2^3, 11) = 8
mod(2^4, 11) = 5
mod(2^5, 11) = 10
mod(2^6, 11) = 9
mod(2^7, 11) = 7

Cela donne un tableau de la forme

reste	position binaire
0	--
1	0
2	1
3	--
4	2
5	4
6	--
7	7
8	3
9	6
10	5

Bons diviseurs

Le diviseur p devrait être aussi petit que possible afin de minimiser la longueur du tableau. Comme le diviseur doit générer n restes distincts, le diviseur va certainement avoir besoin d'être au moins n. Un reste de zéro ne peut cependant se produire que si le diviseur est une puissance de 2. Si le diviseur est une petite puissance de 2, disons 2^j pour j < n-1, il ne va pas générer n restes distincts ; si le diviseur est une plus grande puissance de 2, le tableau de correspondance est de longueur soit 2^{n-1}, soit 2^n. On peut donc exclure zéro comme valeur de reste, de sorte que le diviseur doit être au moins de un supérieur à la longueur du mot. Cette exigence est en fait réalisée pour certaines longueurs de mot.

Soit R(p) le nombre de restes distincts que p génère lorsqu'il est divisé en puissances de 2 successivement croissantes. Les restes distincts se produisent tous pour les R(p) inférieures puissances de 2. Seuls les p impairs nous intéressent et le tableau suivant donne R(p) pour p impair entre 1 et 21.

p	R(p)	p	R(p)
1	1	13	12
3	2	15	4
5	4	17	8
7	3	19	18
9	6	21	6
11	10		

Ce tableau montre que 7, 15, 17 et 21 sont sans utilité comme diviseurs parce qu'il y a de plus petits diviseurs qui génèrent un plus grand nombre de restes distincts. Si on limite notre attention à p tel que $p > p' \geq R(p) > R(p')$, on obtient le tableau suivant des diviseurs utiles pour $p < 100$.

p	R(p)	p	R(p)
1	1	29	28
3	2	37	36
5	4	53	52
9	6	59	58
11	10	61	60
13	12	67	66
19	18	83	82
25	20		

On note que 9 et 25 sont des diviseurs utiles bien qu'ils ne génèrente respectivement que 6 et 20 restes.

Détermination de R(p)

Si p est impair, les restes
$\text{mod}(2^0, p)$
$\text{mod}(2^1, p)$
.
.
.
seront entre 1 et p-1 inclus. À une certaine puissance de 2, disons 2^t, alors k sera un reste répété, de sorte que pour un certain $k < t$, $2^k = 2^t \bmod p$.
Comme $2^{t+1} = 2^{k+1} \bmod p$
et $2^{t+2} = 2^{k+2} \bmod p$
.
.
.
etc.
tous les restes distincts se produisent pour $2^0 \ldots 2^{t-1}$. Donc, R(p) = t.

Ensuite on montre que $2^{R(p)} = 1 \bmod p$
On sait déjà que $2^{R(p)} = 2^k \bmod p$ pour certains $0 \leq k < R(p)$. Soit $j = R(p)-k$ tel que $0 < j \leq R(p)$. Alors,
$2^{k+j} = 2^k \bmod p$
ou $2^j * 2^k = 2^k \bmod p$
ou $(2^j - 1) * 2^k = 0 \bmod p$
Maintenant p ne divise pas 2^k parce que p est impair, de sorte que p doit diviser $2^j - 1$. Donc,
$2^j - 1 = 0 \bmod p$
$2^j = 1 \bmod p$

Comme j est supérieur à 0 par hypothèse et comme il n'y a pas de k autre que 0 inférieur à R(p) tel que $2^k = 2^0 \bmod p$, on doit avoir j = R(p), ou $2^{R(p)} = 1 \bmod p$.
On a donc montré que pour p impair, les restes mod $(2^k, p)$ sont uniques pour k = 0, 1,..., R(p)-1 se répètent ensuite exactement, en commençant par $2^{R(p)} = 1 \bmod p$.

On considère maintenant p pair. Soit $p = p' * 2q$,
où p' est impair. Pour $k < q$, mod $(2^k, p)$ est clairement juste 2^k parce que $2^k < p$.

Pour $k \geq q$, $\text{mod}(2^k, p) = 2^q * \text{mod}(2^{k-q}, p')$.

On peut voir d'après cela que la séquence des restes aura un segment q-1 initial de 1, 2, ...2 de longueur q, et répétant les segments de longueur R(p'). Donc, R(p) = q + R(p'). Comme on s'attend normalement à ce que R(p) ~ p, p pair ne sera généralement pas utile.

On ne connait pas de moyen direct de choisir un p pour un certain n, mais le tableau précédent a été généré à partir du programme Fortran suivant fonctionnant sur le système SEX à UCLA.

```
0
      CALL IASSGN('OC ',56)
```

```
1     FORMAT(I3,I5)
      M=1
      DO 100 K=1,100,2
      K=1
      L=0
20     L=L+1
      N=MOD(2*N,K)
      IF(N.GT.1) GO TO 20
      IF(L.LE.M) GO TO 100
      M=L
      WRITE(56,1)K,L
100    CONTINUE
      STOP
      END
```

Programme Fortran pour des diviseurs utiles en informatique

Dans le programme, K prend les valeurs d'essai de p, N prend les valeurs des restes successifs, L compte jusqu'à R(p), et M mémorise les plus grands R(p) natérieurs. L'exécution est assez rapide.

Résultats de la théorie des nombres

La quantité désignée ci-dessus par R(p) est généralement notée Ord 2 et se lit "l'ordre de 2 modulo p". La valeur maximum de Ord 2 est donnée par la fonction phi d'Euler, parfois appelée le totient. Le totient d'un entier positif p est le nombre d'entiers inférieurs à p qui sont premiers par rapport à p. Le totient est facile à calculer à partir d'une représentation de p comme produit de nombres premiers :

Soit $p = p_1^{n}\,1 * p_2^{n}\,2 \ldots p_k^{n}\,k$

où les p_i sont des nombres premiers distincts. Alors $phi(p) = (p_1 - 1) * p_1^{k-1}\,1 * (p_2 - 1) * p_2^{k-1}\,2 \ldots (p_k - 1) * p_k^{k-1}\,k$

Si p est premier, le totient de p est simplement phi(p) = p-1.

Si p n'est pas premier, le totient est plus petit.

Si a est premier par rapport à p, la généralisation d'Euler du théorème de Fermat déclare alors : $a^{phi(m)} = 1 \bmod p$.

C'est ce théorème qui met une limite supérieure à $Ord_p\,2$, parce que $Ord_p\,2$ est la plus petite valeur telle que

$$2^{Ord_p\,2} = 1 \bmod p$$

De plus, il est toujours vrai que phi(p) est divisible par $Ord_p\,2$.

Remerciements

Bob Kahn a relu une première version et fait de nombreux commentaires qui ont amélioré la présentation. Alex Hurwitz m'a assuré qu'une technique de recherche est nécessaire pour calculer R(p), et a fourni les noms des quantités et théorèmes que j'ai découverts.

[Cette RFC a été mise en forme électronique pour rentrer dans les archives en ligne de RFC archives par Guillaume Lahaye et John Hewes en juin 1997]