Opérateurs arithmétiques matériels
Residue Number System

Florent de Dinechin

13 décembre 2005

Systèmes à bases mixtes

Mixed-radix systems

On dit aussi parfois bases de Cantor.

- Utilisé dans la vie de tous les jours :
  2 semaines, 5 jours, 7 heures, 45 minutes, 13 secondes et 19 centièmes
  - 19 centièmes de secondes
  - $+13 \times 100$ centièmes de secondes
  - $+45 \times 60 \times 100$ centièmes de secondes
  - $+7 \times 60 \times 60 \times 100$ centièmes de secondes
  - $+5 \times 24 \times 60 \times 60 \times 100$ centièmes de secondes
  - $+2 \times 7 \times 24 \times 60 \times 60 \times 100$ centièmes de secondes
- On note
$$\left[\begin{array}{cccccc} 2 & 5 & 7 & 45 & 13 & 19 \\ & 7 & 24 & 60 & 60 & 100 \end{array}\right]$$
- Calculs pas très durs
- Pour l'utilité on va voir tout de suite

Représentation en RNS

A tout seigneur

Il y a un bon survey du sujet dans la thèse de Laurent-Stéphane Didier.

Cela dit l'arithmétique modulaire a près de 18 siècles.

Notations

- Soit $(m_1, ..m_n)$ un tuple d'entiers positifs
- Pour $X \in \mathbf{Z}$ on note $x_i = \langle X \rangle_{m_i}$ ou $x_i \equiv X[m_i]$ pour

$$\begin{cases} X = m_i \times q_i + x_i \\ 0 \leq x_i < m_i \end{cases}$$

- Il y a $M = \prod_{i=1}^{n} m_i$ tuples $(x_1, x_2, ..., x_n)$ différents
- qui peuvent représenter $\mathrm{PPCM}(m_1, ..m_n)$ entiers consécutifs différents
- Dans toute la suite on prendra les m_i premiers entre eux
- On peut donc représenter un intervalle de longueur M (placé où on veut par rapport à 0)
- Formellement c'est le théorème des restes chinois, mais il arrive un peu plus loin
- On appelera les m_i les *moduli* mais je ne suis pas certain de la grammaire latine...
- On appelera les x_i les *restes*.

Exemple

$$(m_1, m_2, m_3) = (2, 3, 5)$$

```
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0
0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3
|                                                   |
```

$$(m_1, m_2, m_3) = (2, 3, 4)$$

```
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0
0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1 2 3 0 1
|                       |                       |
```

Vous voyez un ordre, vous ?

Opérations faciles en RNS

Systèmes à bases mixtes

Représentation en RNS

Opérations faciles en RNS

Opérations difficiles en RNS

Opérations que je zappe tellement elles sont difficiles

Conclusion : des applications ?

L'addition

- On a la propriété

$$\langle X + Y \rangle_{m_i} \quad = \quad \left\langle \ \langle X \rangle_{m_i} \ + \ \langle Y \rangle_{m_i} \ \right\rangle_{m_i} \quad = \quad \langle x_i + y_i \rangle_{m_i}$$

- Donc l'addition se fait en parallèle sur les modules
- Vérifions sur notre exemple :

$$(m_1, m_2, m_3) = (2, 3, 5)$$

```
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0
0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3
|                                                               |
0 1 2 3 4 5 6 7 8 9 10  12  14  16  18  20  22  24  26  28  30=0
```

La soustraction

- $\forall x,\ \forall m_i$ il existe un "opposé modulo m_i", c.a.d. un chiffre mod. m_i y tel que $\langle x + y \rangle_{m_i} = 0$
- et même que c'est $y = m_i - x$
- Donc

$$\begin{aligned} \langle X - Y \rangle_{m_i} &= \left\langle \ \langle X \rangle_{m_i} \ + \ m_i - \langle Y \rangle_{m_i} \ \right\rangle_{m_i} \\ &= \langle x_i + m_i - y_i \rangle_{m_i} \end{aligned}$$

- Vérifions sur notre exemple :

$$(m_1, m_2, m_3) = (2, 3, 5)$$

```
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0
0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3
|                                                               |
0 1 2 3 4 5 6 7 8 9 10  12  14  16  18  20  22  24  26  28  30=0
```

Implémentation de l'addition modulaire

- variations sur "on fait la somme et on retranche éventuellement le modulo"
- Un de nos moduli (le plus grand) sera souvent une puissance de 2
- Choix de moduli important, on y reviendra

La multiplication

Multiplication

- On a toujours (mais c'est la dernière fois)

$$\langle X \times Y \rangle_{m_i} \quad = \quad \left\langle \langle X \rangle_{m_i} \times \langle Y \rangle_{m_i} \right\rangle_{m_i} \quad = \quad \langle x_i \times y_i \rangle_{m_i}$$

- Vérifions sur notre exemple :

$$(m_1, m_2, m_3) = (2, 3, 5)$$

```
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2
0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0 1 2 3 4 0
|                                                               |
0 1 2 3 4 5 6 7 8 9 10  12  14  16  18  20  22  24  26  28  30=0
```

Implémentation de la multiplication modulaire

- Pour des petits moduli, on peut tabuler.
- Une jolie méthode est d'utiliser

$$x_i \times y_i = \frac{(x_i + y_i)^2 - (x_i - y_i)^2}{4}$$

  qui passe très bien au modulo...
- On peut utiliser cette méthode pour calculer des multiplications normales, d'ailleurs (il faut remarquer que les parité de $x + y$ et $x - y$ sont les mêmes).
- Choix de moduli important, on y reviendra

Choix des moduli

En supposant qu'on ne fait que des additions et des multiplications...

- Contraintes :
  - qu'ils soient premiers 2 à 2
  - qu'ils permettent de couvrir un certain intervalle
- La vitesse de calcul sera la vitesse du plus lent opérateur, donc dictée par le plus gros modulo
- La vitesse de conversion dépend du nombre, donc on préfère qu'ils soient tous de la même taille.

Un exemple piqué à un collègue...

- efficacité de représentation p4
- p7

Opérations difficiles en RNS

Opérations difficiles

- Comparaisons
- Détection du signe et des dépassements de capacité
- Division
- Conversion vers/depuis les systèmes à Papa

Inverse modulo m_i

- $\forall m_i,\ \forall x \text{ tq } \langle x \rangle_{m_i} \neq 0 \text{ et } \mathrm{PGCD}(x, m_i) = 1$, il existe un "inverse modulo m_i", c.a.d. un chiffre mod. m_i y tel que $\langle x.y \rangle_{m_i} = 1$
- et même que on le note $\langle 1/x \rangle_{m_i}$ ou $\langle x^{-1} \rangle_{m_i}$
- C'est une bonne motivation pour prendre les m_i premiers

... mais cela ne nous aide pas du tout pour faire la division, parce que la division chiffre a chiffre en multipliant par les inverse a toujours un reste nul, ce n'est donc pas celle qu'on cherche.

Cela dit cela nous aide à prouver le théorème des restes chinois, qui dit que : (TSVP)

Théorème des restes chinois

Pour tout $(m_1, ..., m_n)$ avec les m_i premiers entre eux,
pour tout $(x_1, ..., x_n)$ tq $\forall i\ 0 \leq x_i < m_i$,
il existe un unique X tq $0 \leq X < M$ et $\forall i\ x_i = \langle X \rangle_{m_i}$
Preuve constructive :

- soit $M_i = M/m_i$.
- M_i est premier avec m_i et non nul donc son inverse $n_i = \langle 1/M_i \rangle_{m_i}$ existe[1].
- Clairement $\langle n_i M_i \rangle_{m_j} = 1$ si $j = i$, et 0 sinon car M_i est un multiple de m_j.
- donc $\langle x_i n_i M_i \rangle_{m_j} = x_i$ si $j = i$, et 0 sinon.
- Donc le nombre X suivant fait l'affaire

$$X = \left\langle \sum_{i=1}^{n} x_i n_i M_i \right\rangle_M$$

- (le modulo M c'est pour l'unicité)

[1] Vous avez remarqué que je note en minuscule les "petits" entiers (de la taille des moduli) et en majuscule les "grands" entiers (de la taille de M).

Autre point de vue

$$\begin{aligned} X &= \big\langle (x_1, 0, ..., 0)_{\mathrm{RNS}} + (0, x_2, 0, ..., 0)_{\mathrm{RNS}} + \ ... \\ &\quad + (0, ..., 0, x_n)_{\mathrm{RNS}} \big\rangle_M \\ &= \Big\langle \sum_{i=1}^{n} x_i \delta_i \Big\rangle_M \end{aligned}$$

où δ_i est le nombre codé par le vecteur RNS qui n'a que des 0 et un seul 1 en position i.

- $\delta_1 = (1, 0, ..., 0)$ est un multiple de m_2, m_3,... m_n donc de M_1
- C'est donc un multiple de M_1 tel que son modulo m_1 vaut 1
- Tiens, c'est la définition de mon $n_1 M_1$

Conversions de binaire (ou autre numération de position) vers RNS

Soit X l'entier à convertir.

- Diviser par m_i et garder le reste
  - au moins c'est parallèle sur les chiffres
  - mais faut n diviseurs de la taille de X
- Partir directement de l'écriture de X en binaire sur m bits :

$$X = \sum_{j=0}^{m} \xi_j \times 2^j$$

$$x_i = \langle X \rangle_{m_i} = \left\langle \sum_{j=0}^{m} \xi_j \times \langle 2^j \rangle_{m_i} \right\rangle_{m_i}$$

- On tabule les $\langle 2^j \rangle_{m_i}$ et c'est dans la poche : addition de m termes
- Si m c'est trop, écrire X en base $\beta = 2^k$ et tabuler des $\langle \xi_j \times \beta^j \rangle_{m_i}$
  (tables avec k bits de plus en entrée...)

Conversion RNS vers système de position

- On ressort le théorème du reste chinois :

$$X = \left\langle \sum_{i=1}^{n} x_i n_i M_i \right\rangle_M$$

- Tout le monde l'écrit

$$X = \left\langle \sum_{i=1}^{n} \langle x_i n_i \rangle_{m_i} M_i \right\rangle_M$$

(on va voir pourquoi)

- Problèmes d'implémentation :
  - Calculer les termes de la somme
  - Calculer la somme modulo M

Conversion RNS - système de position par le TRC

$$X = \left\langle \sum_{i=1}^{n} x_i n_i M_i \right\rangle_M \qquad \text{ou bien} \qquad X = \left\langle \sum_{i=1}^{n} \langle x_i n_i \rangle_{m_i} M_i \right\rangle_M$$

- Quelle équation utiliser ?
  - Equation de gauche :
    - Tabuler les $\delta_i = n_i M_i$ (une entrée par m_i, OK)
    - Une grosse multiplication dans chaque terme
    - Vrai problème : son résultat peut être beaucoup plus grand que M
    - Donc calcul du modulo M difficile (pas par "on enlève M si on a débordé")

Conversion RNS - système de position par le TRC

$$X = \left\langle \sum_{i=1}^{n} x_i n_i M_i \right\rangle_M \qquad \text{ou bien} \qquad X = \left\langle \sum_{i=1}^{n} \langle x_i n_i \rangle_{m_i} M_i \right\rangle_M$$

- Quelle équation utiliser ?
  - Donc plutôt équation de droite
    - Tabuler les n_i et les M_i (une entrée par m_i, OK)
    - Une petite multiplication modulaire
    - Une grosse multiplication
    - Tous les termes sont plus petits que M
  - On peut aussi tabuler tous les $\langle x_i n_i M_i \rangle_M$...
    - mais table indicée par (i, x_i) donc à M entrées...

Conversion RNS vers système de position par le TRC

$$X = \left\langle \sum_{i=1}^{n} \left\langle x_i n_i \right\rangle_{m_i} M_i \right\rangle_M$$

- Calcul de la somme modulo :
  - Le calcul de chaque terme se fait en parallèle
  - On peut utiliser un arbre d'additionneurs modulo
    - mais alors ce ne sont pas des additionneurs rapides
  - On peut utiliser un arbre d'additionneurs carry-save
    - mais alors il faut faire un modulo à la fin
    - Dans l'équation de droite le modulo final enlève au plus nM
    - Il y a des ruses pour lire le αM à enlever dans une table indicée par quelques bits de la somme bricolée.
      Par exemple, ne prendre que des moduli de la forme $2^k \pm 1$ avec de grands pour que M soit assez proche d'une puissance de 2 pour qu'on puisse déduire α à un ou deux près en regardant juste les poids forts de la somme...
      Avec un seul modulo qui est un 2^k cette idée aide encore

Conversion RNS vers mixed-radix

- Intérêt : les comparaisons (MRS est un système de position)
- Le système qui va bien c'est

$$
\begin{aligned}
(&m_{n-2}m_{n-3}\ldots m_1 m_0,\\
&m_{n-3}\ldots m_1 m_0,\\
&\ldots\\
&m_1 m_0,\\
&m_0,\\
&1)
\end{aligned}
$$

- Les chiffres vivent alors dans le même intervalle que les chiffres RNS correspondant
- mais pas avec la même valeur !

Conversion RNS vers mixed-radix : algo

$$\begin{aligned} X &= (m_{n-2}m_{n-1}...m_1m_0)x'_{n-1} + ... + (m_1m_0)x'_2 + m_0x'_1 + x'_0 \\ &= (x_{n-1},\ ...\ x_1,\ x_0)_{RNS(m_{n-1},...,m1,m_0)} \end{aligned}$$

- Clairement $x'_0 = x_0$
- Soustraire x_0 à X dans les deux systèmes
- Diviser par m_0 dans les deux systèmes
  - la division est exacte puisque son reste était m_0 qu'on vient d'enlever
  - donc c'est une multiplication modulaire par $\langle 1/m_0 \rangle_{m_0}$
- Cela donne x'_1
- Recommencer

Algorithme itératif sur les moduli, avec que des petites opérations.

Conversion RNS vers num. de position en passant par mixed-radix

- Si on a l'écriture MRS de X, on la convertit facilement en système de position standard en calculant

$$X = x'_0 + \sum_{i=0}^{n-1} (\prod_{j=0}^{i-1} m_i) x'_i$$

- avec les $\prod_{j=0}^{i-1} m_i$ calculés au vol ou stockés dans une table
- On ne coupe pas à des multiplications et des sommes sur de grands nombres.

Opérations que je zappe tellement elles sont difficiles

Des algorithmes de changement de base de moduli

- Shenoy et Kumarasan : utilise le TRC avec un modulo auxiliaire, temps $\log(n)$
- A quoi cela sert : TSVP, et aussi “extension dynamique de la précision” (hum)

La comparaison en général

Des algos en $\log(n)$ (plus rapide que convertir en binaire)

- par changement de base (Hitz et Kaltofen 95) :
  - On convertit X et Y dans les deux bases,
  - on calcule $Z = X - Y$ dans deux bases,
  - on convertit Z d'une base à l'autre.
  - Si ce n'est pas la même valeur on a eu un débordement de capacité donc $Y > X$.
- Par la parité (Chiang et Lu 92)
  - Si M impair, un débordement de capacité dans $Z = X - Y \mod M$ ajoute M, donc change la parité de Z
  - Reste plus qu'à savoir déterminer la parité d'un nombre
  - LCDE : on ne peut pas prendre un modulo qui soit pair... On repasse par le TRC et c'est compliqué
  - temps logarithmique aussi
- Par les diagonales (Dimauro, Impedovo et Pirlo 93)
  - Petit dessin de diagonales principales pour $(4, 5)$
  - Les diagonales sont ordonnées, yaka construire la fonction qui donne la diagonale d'un nombre
  - et alors yapuka comparer les diagonales de X et Y, et si c'est

La division en général

- se contente de comparaisons approchées, si vous vous souvenez
- Comparaison approchée de Hung et Parhami (94)
- Utilisation d'une représentation approchée *pseudo flottante* en parallèle à la représention modulaire (Bajard Didier Muller 96)

Cas particulier : la division X/Y dans le cas où X est un multiple de Y

- dans ce cas on multiplie chiffre à chiffre par les inverses modulaires de Y
- que l'on calcule par l'algorithme d'Euclide étendu en temps logarithmique (parait-il)

Et puis je zappe encore

- Des gadgets genre RNS redondant (une généralisation du RNS modulo $2^k - 1$)
- Des techniques assez générales de regroupement des moduli lorsqu'on préfère faire moins de calcul sur des moduli plus gros
- Des zillions de cas particuliers à base de $2^k \pm 1$...
- Montgomery et compagnie

Conclusion : des applications ?

Moralité

Si on enlève ce que j'ai zappé,

- Le RNS marche bien pour les **petits moduli**
  - on peut tabuler tout ce qui est difficile
  - on peut faire des explorations exhaustives pour choisir les moduli
- Le RNS marche bien pour les **applications** telles que
  - on connait à l'avance les domaines des valeurs (car pas de détection d'overflow)
  - il y a surtout des additions et des multiplications à faire
- Tout cela nous donne le domaine du traitement du signal
- Le RNS marche aussi bien pour les moduli à base de $2^k \pm 1$
  - mais pas de réel consensus sur l'utilité de la chose

Applications

- DSP
- DSP [2]
- DSP [3]
- Cryptographie ? Difficulté : moduli plus grands, on ne va pas pouvoir tout tabuler.

[2] Certains disent que ça consomme moins

[3] Comment on gère la virgule fixe ?