

Décodage adaptatif pour les systèmes multimodulaires redondants

Clément PERNET

en collaboration avec

M. Khonji, J-L. Roch, T. Roche et T. Stalinski

INRIA-MOAIS, LIG, Grenoble Université

Journée nationales du Calcul Formel, CIRM, Luminy
lundi 3 mai 2010

Plan

Introduction

- Calculs exacts haute performance
- Sécurité des calculs distribués
- Tolérance aux fautes
- Algèbre linéaire exacte
- Restes Chinois

Les codes de résidus redondants

- Dans Z : point de vue Mandelbaum
- Dans $K[X]$: point de vue Reed Solomon
- Point de vue généralisé

Approche adaptative

- Première approche
- Détection d'un saut
- Expérimentations
- Terminaison anticipée

Plan

Introduction

Calculs exacts haute performance

Sécurité des calculs distribués

Tolérance aux fautes

Algèbre linéaire exacte

Restes Chinois

Les codes de résidus redondants

Dans Z : point de vue Mandelbaum

Dans $K[X]$: point de vue Reed Solomon

Point de vue généralisé

Approche adaptative

Première approche

Détection d'un saut

Expérimentations

Terminaison anticipée

Calculs exacts haute performance

Domaines de calcul de base

- ▶ \mathbb{Z}, \mathbb{Q} \Rightarrow taille variable
- ▶ $\mathbb{Z}_p, \text{GF}(p^k)$ \Rightarrow arithmétique propre
- ▶ $K[X]$ pour $K = \mathbb{Z}, \mathbb{Z}_p, \dots$

Calculs exacts haute performance

Domaines de calcul de base

- ▶ \mathbb{Z}, \mathbb{Q} \Rightarrow taille variable
- ▶ $\mathbb{Z}_p, \text{GF}(p^k)$ \Rightarrow arithmétique propre
- ▶ $K[X]$ pour $K = \mathbb{Z}, \mathbb{Z}_p, \dots$

Nombreux domaines d'application:

Théorie des nombres:

- ▶ calcul de tables de courbes elliptiques, de formes modulaires,
- ▶ test de conjectures

Crypto:

- ▶ Attaques algébriques (bases de Groebner, cribles quadratiques, calcul d'index,...)
- ▶ Recherches de grands nombres premiers

Théorie des graphes: tests de conjectures (isomorphisme,...)

Plan

Introduction

Calculs exacts haute performance

Sécurité des calculs distribués

Tolérance aux fautes

Algèbre linéaire exacte

Restes Chinois

Les codes de résidus redondants

Dans \mathbb{Z} : point de vue Mandelbaum

Dans $K[X]$: point de vue Reed Solomon

Point de vue généralisé

Approche adaptative

Première approche

Détection d'un saut

Expérimentations

Terminaison anticipée

Calcul parallèle

Vers un calcul massivement parallèle

PC individuels \Rightarrow multi/many cores

- ▶ Fin de la course à la fréquence CPU
- ▶ Multi-core: 2, 4, 6, ...
- ▶ Many-cores: futur proche, déjà dans les GPU's
- ▶ multi-GPU

Calcul parallèle

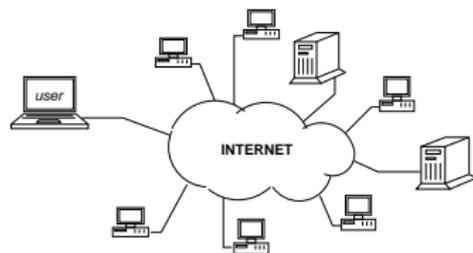
Vers un calcul massivement parallèle

PC individuels ⇒ multi/many cores

- ▶ Fin de la course à la fréquence CPU
- ▶ Multi-core: 2, 4, 6, ...
- ▶ Many-cores: futur proche, déjà dans les GPU's
- ▶ multi-GPU

HPC ⇒ Calcul distribué, global computing

- ▶ serveurs
- ▶ grappes, grilles,
- ▶ calcul volontaire, pair à pair
- ▶ calcul ambient (cloud computing)



Sécurité du calcul

Calcul pair à pair

- ▶ Projets populaires: Mersenne Prime search, SETI@Home, Folding@Home, BOINC
- ▶ Petaflops (670 000 PS3s) en 2007

Sécurité du calcul

Calcul pair à pair

- ▶ Projets populaires: Mersenne Prime search, SETI@Home, Folding@Home, BOINC
- ▶ Petaflops (670 000 PS3s) en 2007
- ▶ **Quel niveau de confiance?**
 - ▶ mauvaise configuration de la machine, overclocking
 - ▶ programme malicieux
 - ▶ corruption à grande échelle (client patché redistribué)

Sécurité du calcul

Calcul pair à pair

- ▶ Projets populaires: Mersenne Prime search, SETI@Home, Folding@Home, BOINC
- ▶ Petaflops (670 000 PS3s) en 2007
- ▶ **Quel niveau de confiance?**
 - ▶ mauvaise configuration de la machine, overclocking
 - ▶ programme malicieux
 - ▶ corruption à grande échelle (client patché redistribué)

Calcul distribué et ambient

- ▶ Connexion / Déconnexion de ressources
- ▶ Corruption (tâche forgée)

Plan

Introduction

Calculs exacts haute performance

Sécurité des calculs distribués

Tolérance aux fautes

Algèbre linéaire exacte

Restes Chinois

Les codes de résidus redondants

Dans \mathbb{Z} : point de vue Mandelbaum

Dans $K[X]$: point de vue Reed Solomon

Point de vue généralisé

Approche adaptative

Première approche

Détection d'un saut

Expérimentations

Terminaison anticipée

Tolérance aux fautes

Différents types de fautes

- ▶ Pannes franches (crash, déconnexion...)
- ▶ Congestion de réseau
- ▶ Attaques malicieuses

Tolérance aux fautes

Différents types de fautes

- ▶ Pannes franches (crash, déconnexion...)
 - ▶ Congestion de réseau
 - ▶ Attaques malicieuses
- ⇒ Modèle de la faute **Byzantine** (pas toujours défaillant)
- ▶ calcul le plus souvent correct ⇒ blacklisting impossible

Tolérance aux fautes

Différents types de fautes

- ▶ Pannes franches (crash, déconnexion...)
 - ▶ Congestion de réseau
 - ▶ Attaques malicieuses
- ⇒ Modèle de la faute **Byzantine** (pas toujours défaillant)
- ▶ calcul le plus souvent correct ⇒ blacklisting impossible

Différentes approches:

- ▶ **Réplication**: vote, spot-checking, blacklisting, exécution partielle sur ressources sûres
- ▶ **Vérification de post-condition** sur les résultats
- ▶ **ABFT**: Algorithm-based fault tolerance

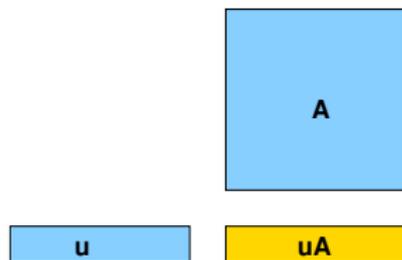
ABFT: Algorithmic Based Fault Tolerance

- Idée: incorporer la redondance dans l'algorithme
 - ⇒ utiliser des propriétés spécifiques au problème

ABFT: Algorithmic Based Fault Tolerance

Idée: incorporer la redondance dans l'algorithme
⇒ utiliser des propriétés spécifiques au problème

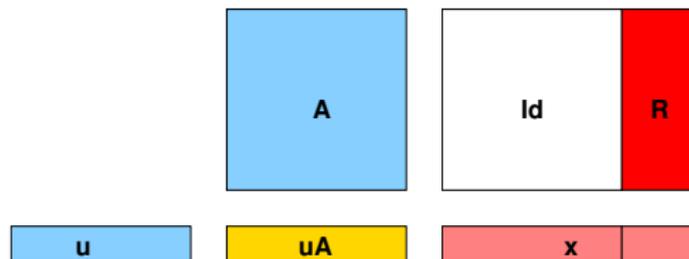
Exemple: Produit matrice vecteur [Dongarra & Al. 2006]



ABFT: Algorithmic Based Fault Tolerance

Idée: incorporer la redondance dans l'algorithme
⇒ utiliser des propriétés spécifiques au problème

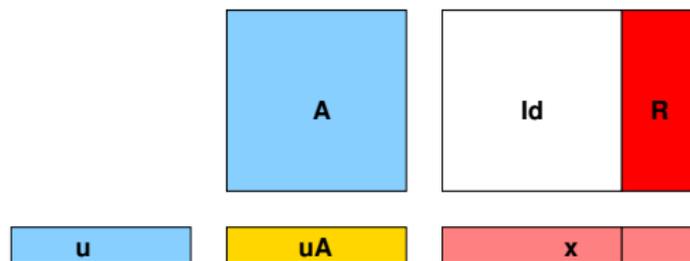
Exemple: Produit matrice vecteur [Dongarra & Al. 2006]



ABFT: Algorithmic Based Fault Tolerance

Idée: incorporer la redondance dans l'algorithme
⇒ utiliser des propriétés spécifiques au problème

Exemple: Produit matrice vecteur [Dongarra & Al. 2006]



- ▶ précalcule le produit $B = A \times [I \mid R]$
- ▶ calcule $x = uB$ en parallèle
- ▶ décode/corrige x

Plan

Introduction

Calculs exacts haute performance

Sécurité des calculs distribués

Tolérance aux fautes

Algèbre linéaire exacte

Restes Chinois

Les codes de résidus redondants

Dans \mathbb{Z} : point de vue Mandelbaum

Dans $K[X]$: point de vue Reed Solomon

Point de vue généralisé

Approche adaptative

Première approche

Détection d'un saut

Expérimentations

Terminaison anticipée

Algèbre linéaire exacte

Mathematics is the art of reducing any problem to linear algebra

W. Stein

⇒ La brique de base à optimiser pour de nombreuses applications

Algèbre linéaire exacte

Mathematics is the art of reducing any problem to linear algebra

W. Stein

⇒ La brique de base à optimiser pour de nombreuses applications

- ▶ Produit de matrices dans $GF(p)$
- ▶ Éliminations: Gauss, Gram-Schmidt (LLL), ...
- ▶ Itération de Krylov
- ▶ Algorithme des restes chinois

Algèbre linéaire exacte

Mathematics is the art of reducing any problem to linear algebra

W. Stein

⇒ La brique de base à optimiser pour de nombreuses applications

- ▶ Produit de matrices dans $GF(p)$
- ▶ Éliminations: Gauss, Gram-Schmidt (LLL), ...
- ▶ Itération de Krylov
- ▶ **Algorithme des restes chinois**

Plan

Introduction

Calculs exacts haute performance

Sécurité des calculs distribués

Tolérance aux fautes

Algèbre linéaire exacte

Restes Chinois

Les codes de résidus redondants

Dans \mathbb{Z} : point de vue Mandelbaum

Dans $K[X]$: point de vue Reed Solomon

Point de vue généralisé

Approche adaptative

Première approche

Détection d'un saut

Expérimentations

Terminaison anticipée

Restes Chinois et évaluation/interpolation

Évaluer P en a

\leftrightarrow

Reduire P modulo $X - a$

Restes Chinois et évaluation/interpolation

Évaluer P en a

\leftrightarrow

Reduire P modulo $X - a$

Polynômes

Evaluation:

$P \bmod X - a$

Évaluer P en a

Interpolation:

$$P = \sum_{i=1}^k \frac{\prod_{j \neq i} (X - a_j)}{\prod_{j \neq i} (a_i - a_j)}$$

Restes Chinois et évaluation/interpolation

Évaluer P en a

\leftrightarrow

Reduire P modulo $X - a$

Polynômes	Entiers
Evaluation: $P \bmod X - a$ Évaluer P en a	$N \bmod m$ “Évaluer” N en m
Interpolation: $P = \sum_{i=1}^k \frac{\prod_{j \neq i} (X - a_j)}{\prod_{j \neq i} (a_i - a_j)}$	$N = \sum_{i=1}^k a_i \prod_{j \neq i} m_j (\prod_{j \neq i} m_j)^{-1} [m_i]$

Terminaison anticipée

Restes chinois classiques

- ▶ borne β sur le résultat
- ▶ Choix des n_i : tels que $n_1 \dots n_k > \beta$

⇒ algorithme déterministe

Terminaison anticipée

Restes chinois classiques

- ▶ borne β sur le résultat
- ▶ Choix des n_i : tels que $n_1 \dots n_k > \beta$

⇒ algorithme déterministe

Terminaison anticipée

- ▶ Pour chaque nouveau modulo n_i :
 - ▶ reconstruire $y_i = f(x) \bmod n_1 \dots n_i$
 - ▶ Si $y_i = y_{i-1}$ ⇒ terminé

⇒ probabiliste Monte Carlo

Terminaison anticipée

Restes chinois classiques

- ▶ borne β sur le résultat
- ▶ Choix des n_i : tels que $n_1 \dots n_k > \beta$

⇒ algorithme déterministe

Terminaison anticipée

- ▶ Pour chaque nouveau modulo n_i :
 - ▶ reconstruire $y_i = f(x) \bmod n_1 \dots n_i$
 - ▶ Si $y_i = y_{i-1}$ ⇒ terminé

⇒ probabiliste Monte Carlo

Avantage:

- ▶ nombre de moduli adaptatif selon la valeur de sortie
- ▶ Intéressant quand
 - ▶ borne pessimiste: matrices creuses, structurées, ...
 - ▶ pas de borne disponible

Plan

Introduction

- Calculs exacts haute performance
- Sécurité des calculs distribués
- Tolérance aux fautes
- Algèbre linéaire exacte
- Restes Chinois

Les codes de résidus redondants

- Dans Z : point de vue Mandelbaum
- Dans $K[X]$: point de vue Reed Solomon
- Point de vue généralisé

Approche adaptative

- Première approche
- Détection d'un saut
- Expérimentations
- Terminaison anticipée

Les codes de résidus redondants

Principe:

- ▶ Utilisation des restes chinois pour la parallélisation
- ▶ Fautes byzantines affectant certains calculs modulaires
- ▶ Reconstruction tolérante aux fautes
 - ⇒ Tolérance aux pannes algorithmique (ABFT)

Plan

Introduction

- Calculs exacts haute performance
- Sécurité des calculs distribués
- Tolérance aux fautes
- Algèbre linéaire exacte
- Restes Chinois

Les codes de résidus redondants

- Dans \mathbb{Z} : point de vue Mandelbaum**
- Dans $K[X]$: point de vue Reed Solomon
- Point de vue généralisé

Approche adaptative

- Première approche
- Détection d'un saut
- Expérimentations
- Terminaison anticipée

Algorithme de Mandelbaum, dans \mathbb{Z}

Théorème des restes chinois

$$x \in \mathbb{Z} \quad \longleftrightarrow \quad \begin{array}{|c|c|c|c|} \hline x_1 & x_2 & \dots & x_k \\ \hline \end{array}$$

où $p_1 \times \dots \times p_k > x$ et $x_i = x \pmod{p_i} \forall i$

Algorithme de Mandelbaum, dans \mathbb{Z}

Théorème des restes chinois

$$x \in \mathbb{Z} \quad \longleftrightarrow \quad \begin{array}{|c|c|c|c|c|c|} \hline x_1 & x_2 & \dots & x_k & x_{k+1} & \dots & x_n \\ \hline \end{array}$$

où $p_1 \times \dots \times p_n > x$ et $x_i = x \pmod{p_i} \forall i$

Algorithme de Mandelbaum, dans \mathbb{Z}

Théorème des restes chinois

$$x \in \mathbb{Z} \quad \longleftrightarrow \quad \begin{array}{|c|c|c|c|c|c|} \hline x_1 & x_2 & \dots & x_k & x_{k+1} & \dots & x_n \\ \hline \end{array}$$

où $p_1 \times \dots \times p_n > x$ et $x_i = x \pmod{p_i} \forall i$

Définition

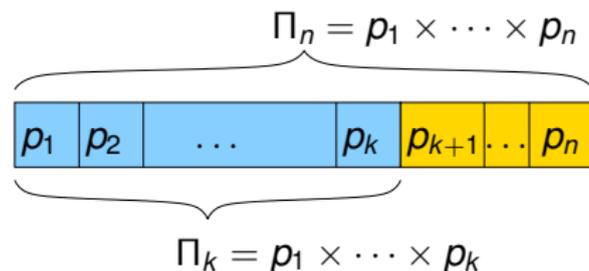
(n, k) -code: $C =$

$$\left\{ (x_1, \dots, x_n) \in \mathbb{Z}_{p_1} \times \dots \times \mathbb{Z}_{p_n} \text{ t.q. } \exists x, \begin{cases} x < p_1 \dots p_k \\ x_i = x \pmod{p_i} \forall i \end{cases} \right\}$$

Principe

Propriété

$$X \in C \text{ ssi } X < \Pi_k.$$



$$\text{Redondance : } r = n - k$$

Principe

Canal de transmission



Calcul

Principe

Canal de transmission **bruité**

≡

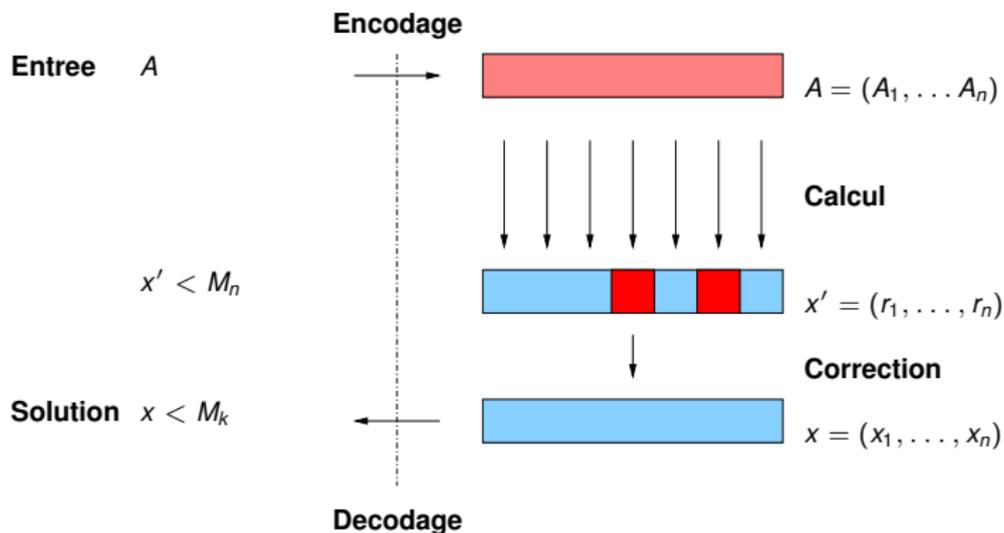
Calcul **non sûr**

Principe

Canal de transmission **bruité**

≡

Calcul **non sûr**



Propriétés du code

Modèle d'erreur:

- ▶ Erreur: $E = X' - X$
- ▶ Support de l'erreur: $I = \{i \in 1 \dots n, E \neq 0 \pmod{p_i}\}$
- ▶ Impact de l'erreur: $\Pi_F = \prod_{i \in I} p_i$

Propriétés du code

Modèle d'erreur:

- ▶ Erreur: $E = X' - X$
- ▶ Support de l'erreur: $I = \{i \in 1 \dots n, E \neq 0 \pmod{p_i}\}$
- ▶ Impact de l'erreur: $\Pi_F = \prod_{i \in I} p_i$

Détecte jusqu'à r erreurs:

Si $X' = X + E$ avec $X \in C, \#I \leq r$,

$$X' > \Pi_k.$$

- ▶ Redondance $r = n - k$, distance: $r + 1$
- ▶ \Rightarrow peut corriger jusqu'à $\lfloor \frac{r}{2} \rfloor$ erreurs en théorie
- ▶ Plus délicat en pratique...

Correction

- ▶ $\forall i \notin I : E \bmod p_i = 0$
- ▶ E est un multiple de Π_V : $E = Z\Pi_V = Z\prod_{i \notin I}$
- ▶ $\text{pgcd}(E, \Pi) = \Pi_V$

Mandelbaum 78: reconstruction rationnelle

$$\begin{aligned} X &= X' - E = X' - Z\Pi_V \\ \frac{X}{\Pi} &= \frac{X'}{\Pi} - \frac{Z}{\Pi_F} \end{aligned}$$

$$\Rightarrow \left| \frac{X'}{\Pi} - \frac{Z}{\Pi_F} \right| \leq \frac{1}{2\Pi_F^2}$$

$\Rightarrow \frac{Z}{\Pi_F} = \frac{E}{\Pi}$ est une réduite de $\frac{X'}{\Pi}$

\Rightarrow reconstruction rationnelle de $X' \bmod \Pi$

Capacité de correction

Mandelbaum 78:

- ▶ 1 symbole = 1 résidu
- ▶ Algo polynomial si $e \leq (n - k) \frac{\log p_{\min} - \log 2}{\log p_{\max} + \log p_{\min}}$
- ▶ exponentiel en pire cas (perturbation aléatoire)

Goldreich Ron Sudan 99 pondération des résidus
⇒ équivalent

Guruswami Sahai Sudan 00 polynomial invariablement

Capacité de correction

Mandelbaum 78:

- ▶ 1 symbole = 1 résidu
- ▶ Algo polynomial si $e \leq (n - k) \frac{\log p_{\min} - \log 2}{\log p_{\max} + \log p_{\min}}$
- ▶ exponentiel en pire cas (perturbation aléatoire)

Goldreich Ron Sudan 99 pondération des résidus
⇒ équivalent

Guruswami Sahai Sudan 00 polynomial invariablement

Interprétation:

- ▶ Les erreurs ont des poids variables selon leur impact

$$\Pi_F = \prod_{i \in I} p_i$$

- ▶ Exemple: $X = 20, p_1 = 2, p_2 = 3, p_3 = 101$
 - ▶ 1 erreur sur $X \bmod 2$, ou $X \bmod 3$, peut être corrigée
 - ▶ mais pas sur $X \bmod 101$

Plan

Introduction

- Calculs exacts haute performance
- Sécurité des calculs distribués
- Tolérance aux fautes
- Algèbre linéaire exacte
- Restes Chinois

Les codes de résidus redondants

- Dans Z : point de vue Mandelbaum
- Dans $K[X]$: point de vue Reed Solomon**
- Point de vue généralisé

Approche adaptative

- Première approche
- Détection d'un saut
- Expérimentations
- Terminaison anticipée

Analogie avec les code de Reed Solomon

Gao02 Décodage des codes RS par Euclide étendu:

- ▶ Restes Chinois dans $K[X]$
- ▶ $p_i = X - a_i$
- ▶ Encodage = évaluation en a_i
- ▶ Decodage = interpolation
- ▶ Correction = Euclide étendu

Analogie avec les code de Reed Solomon

Gao02 Décodage des codes RS par Euclide étendu:

- ▶ Restes Chinois dans $K[X]$
 - ▶ $p_i = X - a_i$
 - ▶ Encodage = évaluation en a_i
 - ▶ Decodage = interpolation
 - ▶ Correction = Euclide étendu
- ⇒ Généralisable pour des p_i de degrés quelconques
- ⇒ notion d'impact variable selon le degré de p_i
- ⇒ unification nécessaire [Sudan 01,...]

Plan

Introduction

- Calculs exacts haute performance
- Sécurité des calculs distribués
- Tolérance aux fautes
- Algèbre linéaire exacte
- Restes Chinois

Les codes de résidus redondants

- Dans Z : point de vue Mandelbaum
- Dans $K[X]$: point de vue Reed Solomon
- Point de vue généralisé**

Approche adaptative

- Première approche
- Détection d'un saut
- Expérimentations
- Terminaison anticipée

Point de vue généralisé: code d'amplitude

- ▶ Dans un anneau euclidien \mathcal{A} muni d'une fonction euclidienne ν
- ▶ Distance

$$\begin{aligned} \Delta : \mathcal{A} \times \mathcal{A} &\rightarrow \mathbb{R}_+ \\ (x, y) &\mapsto \sum_{i|x \neq y} \log_2 \nu(P_i) \end{aligned}$$

Définition

(n, k) -code d'amplitude $C = \{x \in \mathcal{A} : \nu(x) < \kappa\}$,
 $n = \log_2 \Pi$, $k = \log_2 \kappa$.

Point de vue généralisé: code d'amplitude

- ▶ Dans un anneau euclidien \mathcal{A} muni d'une fonction euclidienne ν
- ▶ Distance

$$\begin{aligned} \Delta : \mathcal{A} \times \mathcal{A} &\rightarrow \mathbb{R}_+ \\ (x, y) &\mapsto \sum_{i|x \neq y[P_i]} \log_2 \nu(P_i) \end{aligned}$$

Définition

(n, k) -code d'amplitude $C = \{x \in \mathcal{A} : \nu(x) < \kappa\}$,
 $n = \log_2 \Pi$, $k = \log_2 \kappa$.

Propriété (Quasi MDS)

$d > n - k$ en général, et $d \geq n - k + 1$ dans $K[X]$.

\Rightarrow taux de correction = amplitude maximale d'une erreur pouvant être corrigée

Intérêt

- ▶ Généralisation sur tout anneau euclidien
- ▶ Représentation naturelle des quantités d'information
- ▶ Plus besoin de trier les moduli
- ▶ Capacités de correction plus fines

Intérêt

- ▶ Généralisation sur tout anneau euclidien
- ▶ Représentation naturelle des quantités d'information
- ▶ Plus besoin de trier les moduli
- ▶ Capacités de correction plus fines
- ▶ **Décodage adaptatif**: tirant parti de toute la redondance disponible
- ▶ **Terminaison anticipée**: sans connaissance à priori de borne sur le résultat

Interprétation de Mandelbaum

Remarque

Reconstruction rationnelle \Rightarrow Algorithme d'Euclide Étendu partiel

Propriété

L'algorithme d'Euclide étendu, appliqué à (E, Π) et à $(X' = X + E, \Pi)$, effectue les mêmes premières itérations jusqu'à $r_i < \Pi_V$.

$$\begin{array}{l|l} u_{i-1}E + v_{i-1}\Pi = \Pi_V & u_{i-1}X' + v_{i-1}\Pi = r_{i-1} \\ u_iE + v_i\Pi = 0 & u_iX' + v_i\Pi = r_i \end{array} \Rightarrow u_iX = r_i$$

Décodage par amplitude, à taux borné

Amplitude based decoder over R

Donnée: Π, X'

Donnée: $\tau \in \mathbb{R}_+ \mid \tau < \frac{\nu(\Pi)}{2}$: borne sur l'amplitude max de correction

Résultat: $X \in R$: message corrigé t.q. $\nu(X)4\tau^2 \leq \nu(\Pi)$

begin

$\alpha_0 = 1, \beta_0 = 0, r_0 = \Pi;$

$\alpha_1 = 0, \beta_1 = 1, r_1 = X';$

$i = 1;$

while $(\nu(r_i) > \nu(\Pi)/2\tau)$ **do**

 Soit $r_{i-1} = q_i r_i + r_{i+1}$ la div. euclidienne de r_{i-1} par r_i ;

$\alpha_{i+1} = \alpha_{i-1} - q_i \alpha_i$;

$\beta_{i+1} = \beta_{i-1} - q_i \beta_i$;

$i = i + 1$;

return $X = -\frac{r_i}{\beta_i}$

end

- ▶ atteint le taux de correction maximal

Décodage par amplitude, à taux borné

Amplitude based decoder over R

Donnée: Π, X'

Donnée: $\tau \in \mathbb{R}_+ \mid \tau < \frac{\nu(\Pi)}{2}$: borne sur l'amplitude max de correction

Résultat: $X \in R$: message corrigé t.q. $\nu(X)4\tau^2 \leq \nu(\Pi)$

begin

$\alpha_0 = 1, \beta_0 = 0, r_0 = \Pi;$

$\alpha_1 = 0, \beta_1 = 1, r_1 = X';$

$i = 1;$

while $(\nu(r_i) > \nu(\Pi)/2\tau)$ **do**

 Soit $r_{i-1} = q_i r_i + r_{i+1}$ la div. euclidienne de r_{i-1} par r_i ;

$\alpha_{i+1} = \alpha_{i-1} - q_i \alpha_i$;

$\beta_{i+1} = \beta_{i-1} - q_i \beta_i$;

$i = i + 1$;

return $X = -\frac{r_i}{\beta_i}$

end

- ▶ atteint le taux de correction maximal
- ▶ nécessite une connaissance a priori de τ
⇒ Comment rendre le taux de correction adaptatif?

Plan

Introduction

- Calculs exacts haute performance
- Sécurité des calculs distribués
- Tolérance aux fautes
- Algèbre linéaire exacte
- Restes Chinois

Les codes de résidus redondants

- Dans Z : point de vue Mandelbaum
- Dans $K[X]$: point de vue Reed Solomon
- Point de vue généralisé

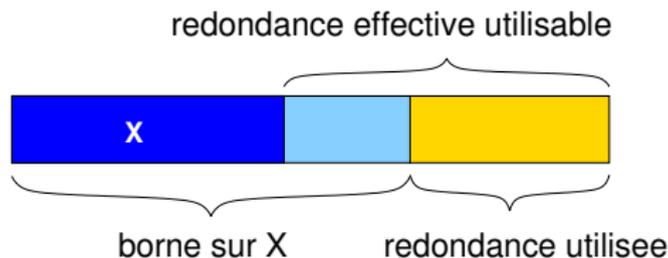
Approche adaptative

- Première approche
- Détection d'un saut
- Expérimentations
- Terminaison anticipée

Approche adaptative

Buts multiples:

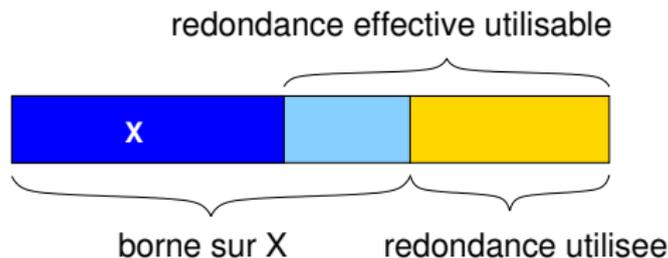
- ▶ A n fixé, taux de correction dépendant de la borne sur X
 - ⇒ taux de correction pessimiste
 - ⇒ comment tirer parti de toute la redondance disponible?



Approche adaptative

Buts multiples:

- ▶ A n fixé, taux de correction dépendant de la borne sur X
 - ⇒ taux de correction pessimiste
 - ⇒ comment tirer parti de toute la redondance disponible?



- ▶ Permettre la terminaison anticipée: n variable et borne inconnue

Plan

Introduction

- Calculs exacts haute performance
- Sécurité des calculs distribués
- Tolérance aux fautes
- Algèbre linéaire exacte
- Restes Chinois

Les codes de résidus redondants

- Dans Z : point de vue Mandelbaum
- Dans $K[X]$: point de vue Reed Solomon
- Point de vue généralisé

Approche adaptative

- Première approche**
- Détection d'un saut
- Expérimentations
- Terminaison anticipée

Première approche adaptative

Critère de terminaison dans Euclide Étendu:

- ▶ $\alpha_{i+1}\Pi - \beta_{i+1}E = 0$
 - ⇒ $E = \alpha_{i+1}\Pi / \beta_{i+1}$
 - ⇒ tester si β_j divise Π
- ▶ tester la cohérence des bornes: $\nu(X) \leq \frac{\nu(\Pi)}{4\nu(\beta_j)^2}$
- ▶ Mais plusieurs candidats possibles
 - ⇒ discrimination par une post-condition sur le résultat

Première approche adaptative

Critère de terminaison dans Euclide Étendu:

- ▶ $\alpha_{i+1}\Pi - \beta_{i+1}E = 0$
 - ⇒ $E = \alpha_{i+1}\Pi / \beta_{i+1}$
 - ⇒ tester si β_j divise Π
- ▶ tester la cohérence des bornes: $\nu(X) \leq \frac{\nu(\Pi)}{4\nu(\beta_j)^2}$
- ▶ Mais plusieurs candidats possibles
 - ⇒ discrimination par une post-condition sur le résultat

Exemple

p_i	3	5	7
x_i	2	3	2

- ▶ $x = 23$ avec 0 erreur
- ▶ $x = 2$ avec 1 erreur

Plan

Introduction

- Calculs exacts haute performance
- Sécurité des calculs distribués
- Tolérance aux fautes
- Algèbre linéaire exacte
- Restes Chinois

Les codes de résidus redondants

- Dans Z : point de vue Mandelbaum
- Dans $K[X]$: point de vue Reed Solomon
- Point de vue généralisé

Approche adaptative

- Première approche
- Détection d'un saut**
- Expérimentations
- Terminaison anticipée

Détection d'un saut

$$\alpha_j \Pi - \beta_j (X + E) = r_j \quad \Rightarrow \quad \alpha_j \Pi - \beta_j E = r_j + \beta_j X$$



Détection d'un saut

$$\alpha_j \Pi - \beta_j (X + E) = r_j \quad \Rightarrow \quad \alpha_j \Pi - \beta_j E = r_j + \beta_j X$$



Détection d'un saut

$$\alpha_i \Pi - \beta_i (X + E) = r_i \quad \Rightarrow \quad \alpha_i \Pi - \beta_i E = r_i + \beta_i X$$



Détection d'un saut

$$\alpha_i \Pi - \beta_i (X + E) = r_i \quad \Rightarrow \quad \alpha_i \Pi - \beta_i E = r_i + \beta_i X$$



$$X = -r_i / \beta_i$$

- ▶ A l'itération finale: $\nu(r_i) \approx \nu(\beta_i X)$
- ▶ Si besoin, saut entre r_{i-1} et r_i .

Détection d'un saut

$$\alpha_i \Pi - \beta_i (X + E) = r_i \quad \Rightarrow \quad \alpha_i \Pi - \beta_i E = r_i + \beta_i X$$



$$X = -r_i / \beta_i$$

- ▶ A l'itération finale: $\nu(r_i) \approx \nu(\beta_i X)$
- ▶ Si besoin, saut entre r_{i-1} et r_i .
- ▶ \Rightarrow Ajout d'un *blanc* 2^g pour détecter le saut de 2^g

Détection d'un saut

$$\alpha_i \Pi - \beta_i (X + E) = r_i \quad \Rightarrow \quad \alpha_i \Pi - \beta_i E = r_i + \beta_i X$$



$$X = -r_i / \beta_i$$

- ▶ A l'itération finale: $\nu(r_i) \approx \nu(\beta_i X)$
- ▶ Si besoin, saut entre r_{i-1} et r_i .
- ▶ \Rightarrow Ajout d'un *blanc* 2^g pour détecter le saut de 2^g

Détection d'un saut

$$\alpha_i \Pi - \beta_i (X + E) = r_i \quad \Rightarrow \quad \alpha_i \Pi - \beta_i E = r_i + \beta_i X$$

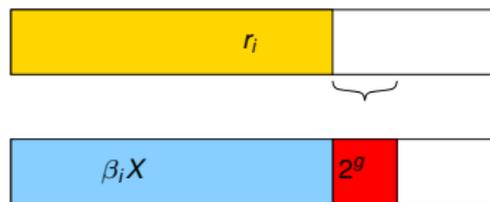


$$X = -r_i / \beta_i$$

- ▶ A l'itération finale: $\nu(r_i) \approx \nu(\beta_i X)$
- ▶ Si besoin, saut entre r_{i-1} et r_i .
- ▶ \Rightarrow Ajout d'un *blanc* 2^g pour détecter le saut de 2^g

Détection d'un saut

$$\alpha_i \Pi - \beta_i (X + E) = r_i \quad \Rightarrow \quad \alpha_i \Pi - \beta_i E = r_i + \beta_i X$$

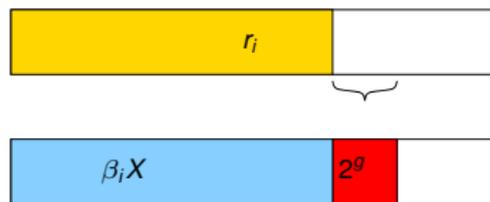


$$X = -r_i / \beta_i$$

- ▶ A l'itération finale: $\nu(r_i) \approx \nu(\beta_i X)$
- ▶ Si besoin, saut entre r_{i-1} et r_i .
- ▶ \Rightarrow Ajout d'un *blanc* 2^g pour détecter le saut de 2^g

Détection d'un saut

$$\alpha_i \Pi - \beta_i (X + E) = r_i \quad \Rightarrow \quad \alpha_i \Pi - \beta_i E = r_i + \beta_i X$$



$$X = -r_i / \beta_i$$

- ▶ A l'itération finale: $\nu(r_i) \approx \nu(\beta_i X)$
- ▶ Si besoin, saut entre r_{i-1} et r_i .
- ▶ \Rightarrow Ajout d'un *blanc* 2^g pour détecter le saut de 2^g

Propriété

- ▶ *Perte de capacité de correction: très faible en pratique*
- ▶ *Test de la divisibilité pour les candidats restants*
- ▶ *Réduit fortement le nombre de tests de divisibilité*

Plan

Introduction

- Calculs exacts haute performance
- Sécurité des calculs distribués
- Tolérance aux fautes
- Algèbre linéaire exacte
- Restes Chinois

Les codes de résidus redondants

- Dans Z : point de vue Mandelbaum
- Dans $K[X]$: point de vue Reed Solomon
- Point de vue généralisé

Approche adaptative

- Première approche
- Détection d'un saut

Expérimentations

- Terminaison anticipée

Expérimentations

Taille de l'erreur	10	50	100	200	500	1000
$g = 2$	1/446	1/765	1/1118	2/1183	2/4165	1/7907
$g = 3$	1/244	1/414	1/576	2/1002	2/2164	1/4117
$g = 5$	1/53	1/97	1/153	2/262	1/575	1/1106
$g = 10$	1/1	1/3	1/9	1/14	1/26	1/35
$g = 20$	1/1	1/1	1/1	1/1	1/1	1/1

Table: Nombre de candidats dans l'algorithme du saut: c/d signifie que d candidats sont apparus avec un saut $> 2^g$, et c parmi eux ont passé le test de divisibilité. $n \approx 6001$ (3000 moduli), $\kappa \approx 201$ (100 moduli).

Expérimentations

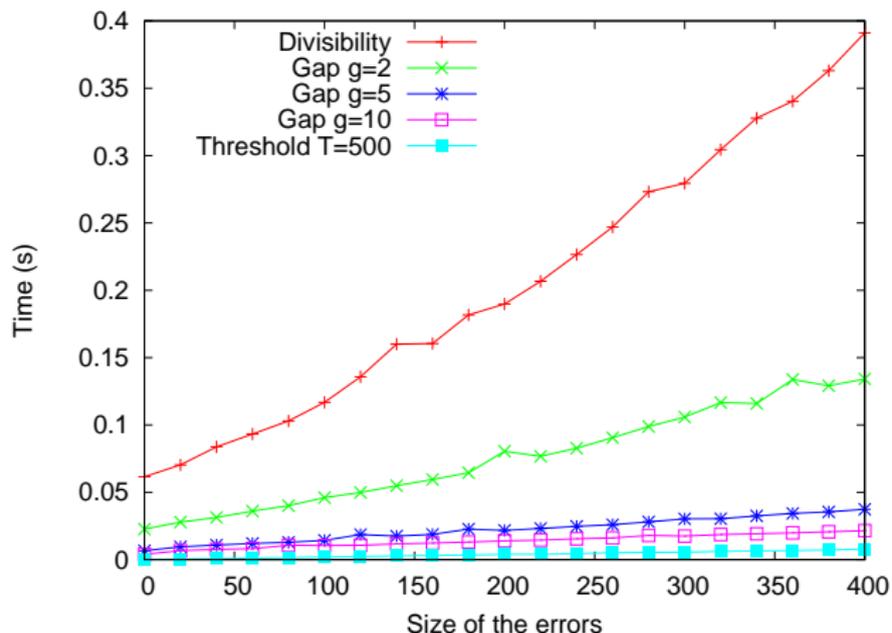


Figure: Comparaison pour $n \approx 26\,016$ ($m = 1300$ moduli de 20 bits), $\kappa \approx 6001$ (300 moduli) et $\tau \approx 10007$ (environ 500 moduli).

Expérimentations

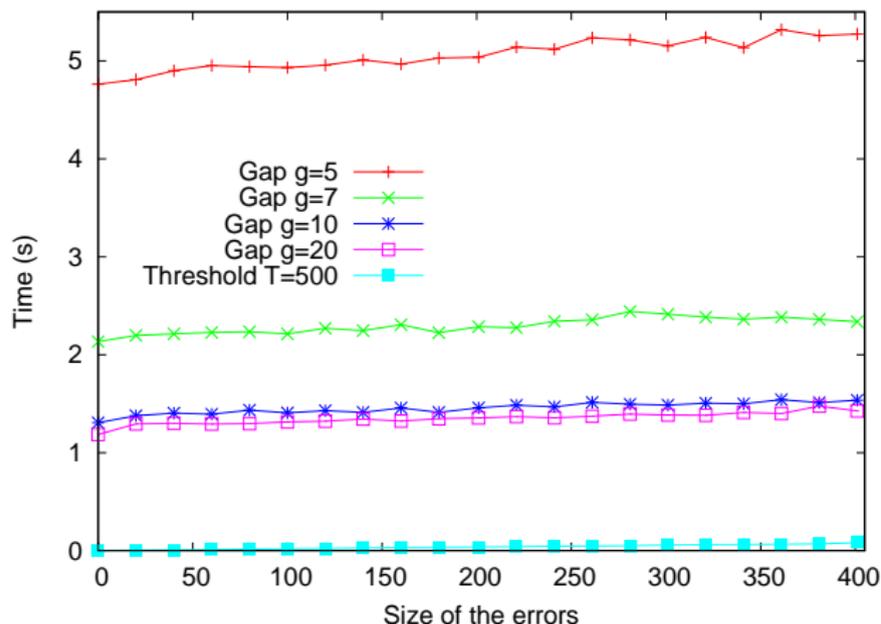


Figure: Comparaison pour $n \approx 200\,917$ ($m = 10000$ moduli de 20 bits), $\kappa \approx 17\,0667$ (8500 moduli) et $\tau \approx 10498$ (500 moduli).

Gap: algorithme d'Euclide jusqu'au bout \Rightarrow surcoût

Plan

Introduction

- Calculs exacts haute performance
- Sécurité des calculs distribués
- Tolérance aux fautes
- Algèbre linéaire exacte
- Restes Chinois

Les codes de résidus redondants

- Dans Z : point de vue Mandelbaum
- Dans $K[X]$: point de vue Reed Solomon
- Point de vue généralisé

Approche adaptative

- Première approche
- Détection d'un saut
- Expérimentations
- Terminaison anticipée**

Terminaison anticipée

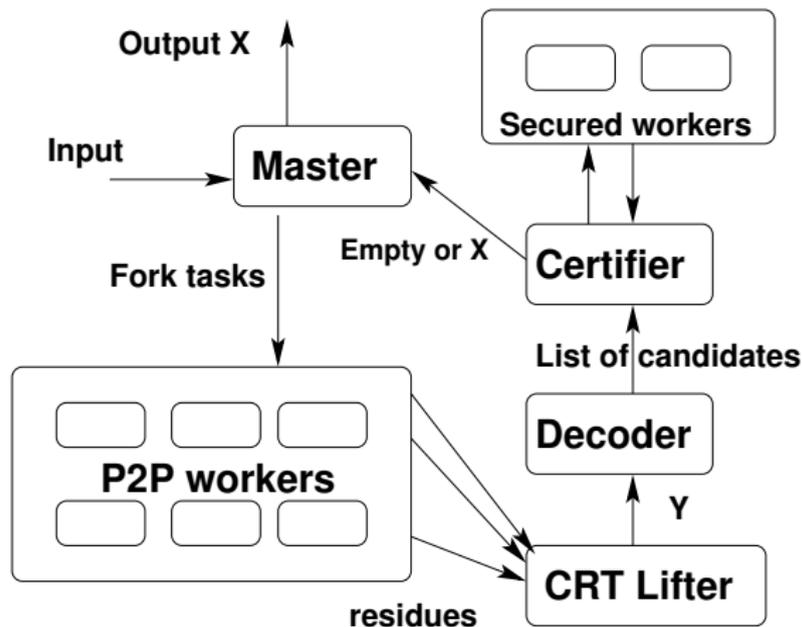


Figure: Principe d'un calcul distribué tolérant aux fautes avec terminaison anticipée

Conclusion

Nouvelle métrique pour les codes de résidus redondants:

- ▶ Unification
- ▶ Bornes plus fines sur les capacités de correction
- ▶ permet le décodage adaptatif au maximum des capacités

Décodage adaptatif et terminaison anticipée

- ▶ Plusieurs approches
- ▶ Méthode du saut: surcoût limité, performances meilleures
- ▶ Insertion dans un schéma global de terminaison anticipée

Perspective

- ▶ Explication théorique de l'efficacité du saut (distribution moyenne des q_i dans Euclide Étendu).
- ▶ Meilleurs taux de corrections pour \mathbb{Z} et $K[X]$ seulement.
- ▶ Appliquer au décodage par liste [Sudan, Guruswami]