CUYASHE : Computação sobre dados cifrados em *GPGPUs*

Autores: **Pedro Geraldo M. R. Alves,**
Diego de Freitas Aranha

Instituto de Computação – Universidade Estadual de Campinas (Unicamp)

André Luís Trigo Fernandes

Introdução

- Criptográfia durante o processamento da informação;
- *Honest-but-curious;*
- Transporte, Processamento e Armazenamento;
- Criptografia Homomórfica.

Objetivo

- Ganho de desempenho nos criptossistemas YASHE [Bos et al. 2013];
- Computação paralela em processadores gráficos de propósito geral (GPGPU's);
- Arquitetura *CUDA*;
- Teorema do Resto Chinês;
- Transformada de Fourier.

Criptografia Homomórfica

Definição 1. *Seja E uma função de cifração e D a função de decifração correspondente. Sejam m_1 e m_2 dados em claro. A dupla (E, D) forma uma cifra dita homomórfica com respeito a um operador $\diamond$ se a seguinte propriedade for satisfeita:*

$$D\left(E\left(m_1\right) \circ E\left(m_2\right)\right) = D\left(E\left(m_1 \diamond m_2\right)\right) \Rightarrow D\left(E\left(m_1\right) \circ E\left(m_2\right)\right) = m_1 \diamond m_2.$$

A operação $\circ$ é equivalente à operação $\diamond$ no espaço de criptogramas.

YASHE - *Yet Another Somewhat Homomorphic Encryption*

- é um criptossistema completamente homomórfico em nível baseado em reticulados [Bos et al. 2013]. Ele opera sobre elementos de um anel gerado por um polinômio ciclotômico.

Definição 2. *Seja* χ_{err} *a distribuição Gaussiana discreta e* χ_{key} *a distribuição estreita, que gera valores aleatoriamente dentre* $\{-1, 0, 1\}$. *O criptossistema* YASHE *é composto pelas seguintes operações:*

***Escolha de parâmetros**: Dado um parâmetro* λ *de segurança,*

1. *Escolha um inteiro* q *que define o espaço dos coeficientes dos criptogramas.*
2. *Escolha um inteiro* t *que define o espaço dos coeficientes dos textos claros. Garanta que* $1 < t < q$.
3. *Escolha um inteiro* $w > 1$, *que define o tamanho da palavra nas operações de troca de chave.*

YASHE - *Yet Another Somewhat Homomorphic Encryption*

4. *Defina o anel* $R_q = \mathbb{Z}_q[X]/\phi_n(X)$, *onde* $\phi_n(X)$ *é o* n*-ésimo polinômio ciclotômico.*

***Geração de chaves**:*

1. *Amostre dois polinômios* f' *e* g *da distribuição* χ_{key} *e defina* $f = \left[tf' + 1\right]_q$.
2. *Se* f *não for inversível em* R_q, *escolha um novo* f'.
3. *Defina* $h = [tgf^{-1}]_q$.
4. *Defina* $\gamma = [PowersOf_{w,q}(f) + e + h \cdot s]_q \in R_q^{\log_w q}$.
5. *Retorne* $(pk, sk, evk) = (h, f, \gamma)$.

YASHE - *Yet Another Somewhat Homomorphic Encryption*

***Cifração**: Dadas a chave (pk) e uma mensagem $m \in R_t$,*

1. *Amostre dois polinômios s e e da distribuição χ_{err}.*
2. *Retorne o criptograma $c = \left[\frac{q}{t} \cdot [m]_t + e + h \cdot s\right] \in R_q$.*

***Decifração**: Dadas a chave (sk) e um criptograma $c \in R_q$,*

Retorne $m = \left[\left\lfloor \frac{t}{q} \cdot [fc]_q \right\rceil\right] \in R_t$.

***Adição**: Sejam os criptogramas c_1 e c_2,*

Retorne $c_{add} = [c_1 + c_2]_q$.

YASHE - *Yet Another Somewhat Homomorphic Encryption*

***Multiplicação**: Sejam os criptogramas c_1 e c_2 e a chave evk,*

1. *Defina* $c^*_{mult} = \left[\left\lfloor \frac{t}{q} \cdot c_1 \cdot c_2 \right\rceil\right]_q$.
2. *Retorne* $c_{mult} = KeySwitch(c^*_{mult}, evk)$.

***PowersOf**$_{w,q}$:*

Retorne $\left(\left[a \cdot w^i\right]_q\right)_{i=0}^{\log_w q - 1} \in R^{\log_w q}$.

***KeySwitch**: Seja o criptograma c e a chave evk,*

Retorne $\left[\left\langle WordDecomp_{w,q}\left(c\right), evk\right\rangle\right]_q$.

***WordDecomp**$_{w,q}$: Seja o polinômio a.*

Retorne $\left(\left[a_i\right]_w\right)_{i=0}^{\log_w q - 1} \in R^{\log_w q}$.

Teorema do Resto Chinês

- Simplificar coeficientes grandes de polinômios em pequenos resíduos para facilitar a aritmética;

Definição 3. - *Transformação CRT*
Seja x *um polinômio em* R_q *e* $\{p_0, p_1, \ldots, p_{l-1}\}$ *um conjunto de* l *primos.*

Define-se: $CRT : x \rightarrow \{x \mod p_0, x \mod p_1, \ldots, x \mod p_{l-1}\}$.

Definição 4. - *Transformação Inversa da CRT , ou ICRT*
Seja x *um conjunto de* l *polinômios residuais oriundos do CRT ,* $\{p_0, p_1, \ldots, p_{l-1}\}$ *o respectivo conjunto de primos e* $M = \Pi_{i=0}^{l-1} p_i$.

Define-se: $ICRT(x) = \sum_{i=0}^{l-1} \left(\frac{M}{p_i}\right) \cdot \left(\left(\frac{M}{p_i}\right)^{-1} x_i \mod p_i\right) \mod M$.

Teorema do Resto Chinês

- YASHE é preciso que $M = \Pi_{i=0}^{l-1} p_i > n \cdot q^2$, onde *n* e *q* são os parâmetros de formação do anel.

Definição 5. - *Operações sobre o domínio do CRT*
Sejam x *e* y *polinômios em* R_q *com* X *e* Y *sendo seus respectivos conjuntos de resíduos,* $\{p_0, p_1, \ldots, p_{l-1}\}$ *um conjunto de* l *primos e uma operação* $\diamond$.

Define-se: $X \diamond Y = \{(X_0 \diamond Y_0), (X_1 \diamond Y_1), \ldots, (X_{l-1} \diamond Y_{l-1})\}$.

Transformada Rápida de Fourier

- Redução da complexidade computacional de uma operação de multiplicação polinomial;

- $\Theta\ (N^2)$ para $\Theta\ (N \log N)$ N sendo o grau dos polinômios;

Transformada Rápida de Fourier

Definição 6. - *Multiplicação de polinômios usando FFT*

Sejam dois polinômios p e q escritos como:

$$p(x) = a_0 + a_1 x + \ldots + a_{N-1} x^{N-1},$$
$$q(x) = b_0 + b_1 x + \ldots + b_{N-1} x^{N-1}.$$

Transformada Rápida de Fourier

Além disso, seja $\omega_N = e^{i\frac{2\Pi}{N}}$ *a n-ésima raíz complexa de* 1. *O produto*

$$(p \cdot q)(x) = p(x) \cdot q(x) = c_0 + c_1 x + \ldots + c_{2N-2} x^{2N-2}$$

é calculado da seguinte forma:

1. *Avalie eficientemente* $p(x)$ *e* $q(x)$ *nos pontos* $\omega_{2N}^0, \omega_{2N}^1, \ldots, \omega_{2N}^{2N-1}$.
2. *Compute os valores de* $p(x) \cdot q(x)$ *nesses pontos através da multiplicação ponto-a-ponto*

$$\begin{aligned}
(p \cdot q)(\omega_{2N}^0) &= p(\omega_{2N}^0) \cdot q(\omega_{2N}^0), \\
(p \cdot q)(\omega_{2N}^1) &= p(\omega_{2N}^1) \cdot q(\omega_{2N}^1), \\
&\vdots \\
(p \cdot q)(\omega_{2N}^{2N-1}) &= p(\omega_{2N}^{2N-1}) \cdot q(\omega_{2N}^{2N-1}).
\end{aligned}$$

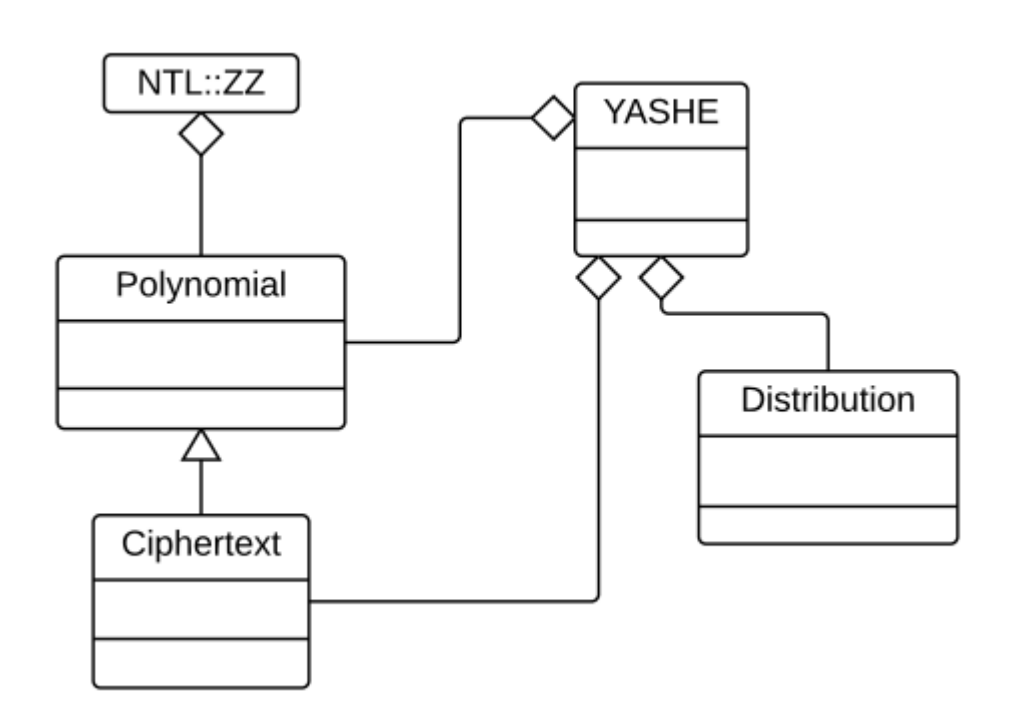
3. *Aplique a transformação inversa da Transformada Discreta de Fourier e recupere os coeficientes* c_i *tal que*

$$(p \cdot q)(x) = \sum_{i=0}^{2N-2} c_i x^i.$$

Implementação cuYASHE

- C++;
- Biblioteca *NTL* para prover a aritmética de grandes inteiros;
- Polinômios-genéricos e polinômios-criptogramas com as classes *Polynomial* e *Ciphertext*;
- Distribuição Probabilística, Gaussiana discreta e estreita, foram concentradas na classe *Distribution;*

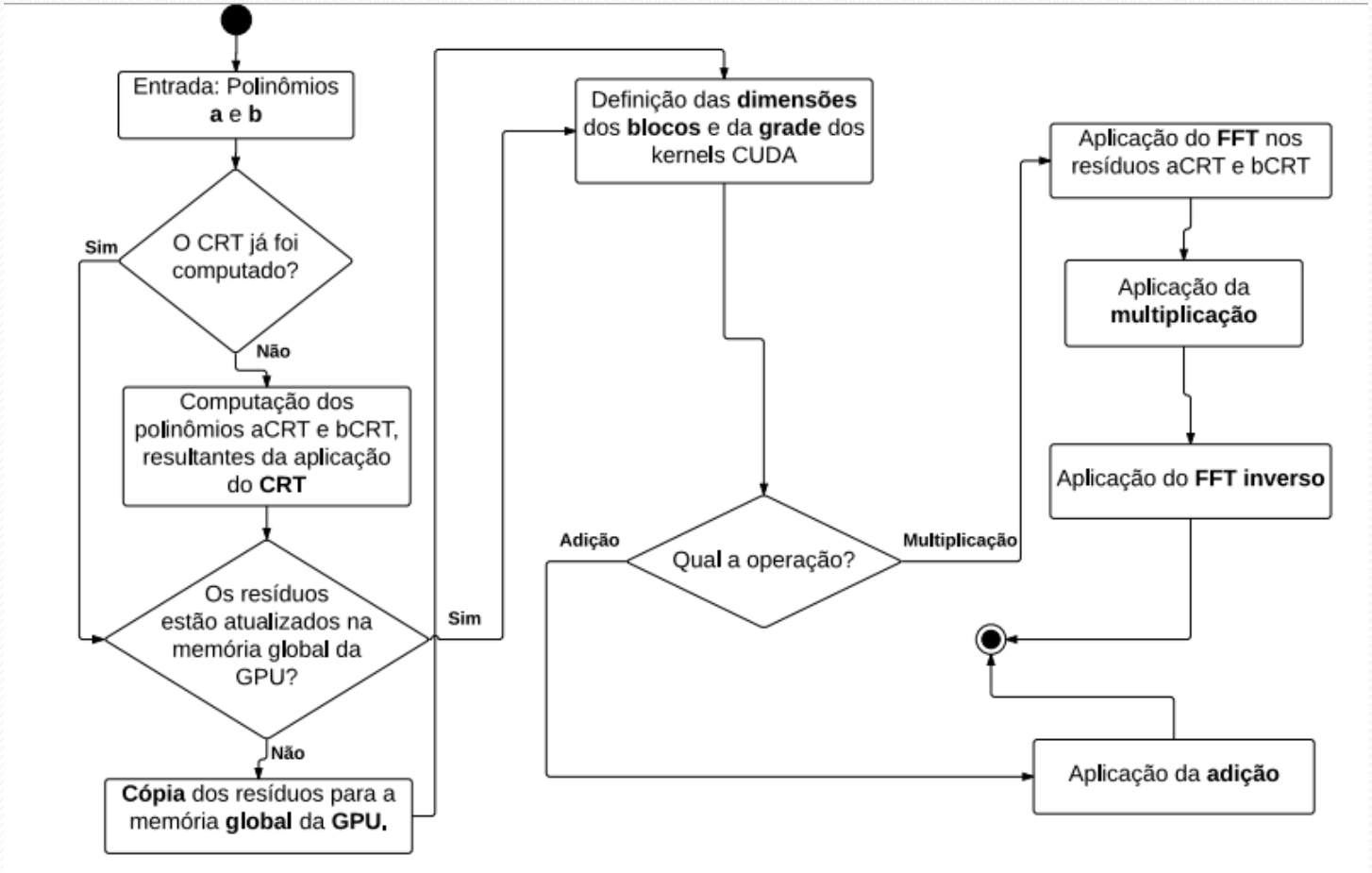
Implementação cuYASHE



Localidade na memória

- Cópia dos resíduos do *CRT* para a memória *global* da *GPU* é feita concatenando os polinômios em um único vetor de inteiros
- Acesso a memória(*coalesced*).

Localidade na memória



Aritmética Polinomial

- Paralelismo;
- *CUDA Threads;*
- Cada bloco de *CUDA Threads* é composto por *threads* de tamanho N/32 para um polinômio de grau *n-1*;
- Soma elemento a elemento dois vetores de inteiros, compostos pela concatenação dos coeficientes dos polinômios residuais do *CRT* de cada operando garantindo alinhamento dos coeficientes;
- A subtração é realizada de maneira semelhante.

Aritmética Polinomial

- Multiplicação;
- FFT variação não normalizada, *cuFFT* [NVIDIA 2015a];
- usado o modo de funcionamento *Complex-to-Complex* da *cuFFT*;
- Divisão e resto consideravelmente caro sem a exploração de alguma propriedade especial dos polinômios;
- *n*-ésimo polinômio ciclotômico é dado por $\Phi n\ (x) = x\hat{}(n/2) + 1$, para todo *n* potência de 2

Aritmética Polinomial

Lema 1. *Divisão e resto pelo n-ésimo polinômio ciclotômico. Sejam* $\Phi_n(x)$ *o n-ésimo polinômio ciclotômico com n potência de 2 e* $P(x) = \sum_{i=0}^{m-1} a_i x^i$, *com* $m > n$. *Então o quociente* $Q(x)$ *e o resto* $R(x)$ *da divisão de* $P(x)$ *por* $\Phi_n(x)$ *são dados por:*

$$Q(x) = \frac{P(x)}{x^{\frac{n}{2}}} = \sum_{i=0}^{\frac{n}{2}-1} a_{i+n/2} x^i,$$

$$R(x) = P(x) - Q(x) \cdot \Phi_n(x).$$

Resultados

Operação	cuYASHE (ms)	LN (ms)	BLLN (ms)
Cifração	32,28	16	27
Decifração	13,87	15	5
Adição Homomórfica	0,56	0,7	0,024
Multiplicação Homomórfica + KeySwitch	20,29	49	31

Operação	**Com sobrecarga[9] de memória (ms)**	**Sem sobrecarga de memória (ms)**
Adição	19,24	1,32
Multiplicação	32,28	7,01

CUYASHE : Computação sobre dados cifrados em *GPGPUs*

Autores: **Pedro Geraldo M. R. Alves,**
Diego de Freitas Aranha

Instituto de Computação – Universidade Estadual de Campinas (Unicamp)

André Luís Trigo Fernandes