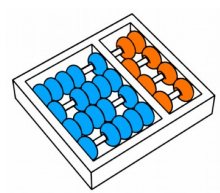
COMPUTAÇÃO SOBRE DADOS CIFRADOS EM GPGPUS

Pedro Geraldo M. R. Alves
Orientador: Diego F. Aranha

9 de junho de 2016

Instituto de Computação
Universidade Estadual de Campinas



AGENDA

INTRODUÇÃO

INTRODUÇÃO - CONTEXTO

- Popularização da oferta de serviços na nuvem.

- Popularização da oferta de serviços na nuvem.
  - Redução de custos para instalação e manutenção.

- Popularização da oferta de serviços na nuvem.
  - Redução de custos para instalação e manutenção.
  - Redução de custos de escalonamento.

- Popularização da oferta de serviços na nuvem.
  - Redução de custos para instalação e manutenção.
  - Redução de custos de escalonamento.
  - Redundância.

- Popularização da oferta de serviços na nuvem.
  - Redução de custos para instalação e manutenção.
  - Redução de custos de escalonamento.
  - Redundância.
- Transporte e armazenamento podem ser seguros.

- Popularização da oferta de serviços na nuvem.
  - Redução de custos para instalação e manutenção.
  - Redução de custos de escalonamento.
  - Redundância.
- Transporte e armazenamento podem ser seguros.
- O processamento não é seguro.

- Popularização da oferta de serviços na nuvem.
  - Redução de custos para instalação e manutenção.
  - Redução de custos de escalonamento.
  - Redundância.
- Transporte e armazenamento podem ser seguros.
- O processamento não é seguro.
- Redução de custos e de segurança.

- Popularização da oferta de serviços na nuvem.
  - Redução de custos para instalação e manutenção.
  - Redução de custos de escalonamento.
  - Redundância.
- Transporte e armazenamento podem ser seguros.
- O processamento não é seguro.
- Redução de custos e de segurança.
- Criptossistemas homomórficos podem ajudar.

INTRODUÇÃO - CONTEXTO

- Popularização da oferta de serviços na nuvem.
  - Redução de custos para instalação e manutenção.
  - Redução de custos de escalonamento.
  - Redundância.
- Transporte e armazenamento podem ser seguros.
- O processamento não é seguro.
- Redução de custos e de segurança.
- Criptossistemas homomórficos podem ajudar.
  - Suporta operações sobre dados cifrados.

- Popularização da oferta de serviços na nuvem.
  - Redução de custos para instalação e manutenção.
  - Redução de custos de escalonamento.
  - Redundância.
- Transporte e armazenamento podem ser seguros.
- O processamento não é seguro.
- Redução de custos e de segurança.
- Criptossistemas homomórficos podem ajudar.
  - Suporta operações sobre dados cifrados.
  - Baixo desempenho.

Obter **ganho de desempenho** no estado da arte
do criptossistema homomórfico YASHE utilizando GPGPUs.

FUNDAMENTAÇÃO TEÓRICA

Seja E a função de cifração de um esquema homomórfico e m_1 e m_2 um par de mensagens. Então,

$$E(m_1) \circ E(m_2) \equiv E(m_1 \diamond m_2).$$

Seja E a função de cifração de um esquema homomórfico e m_1 e m_2 um par de mensagens. Então,

$$E(m_1) \circ E(m_2) \equiv E(m_1 \diamond m_2).$$

No esquema de **ElGamal**, $\circ$ = multiplicação e $\diamond$ = adição.

Criptossistemas parcialmente homomórficos

Suportam adição **ou** multiplicação.

Criptossistemas completamente homomórficos

Suportam adição **e** multiplicação.

Criptossistemas parcialmente homomórficos

Suportam adição **ou** multiplicação.

Criptossistemas completamente homomórficos

Suportam adição **e** multiplicação.

Criptossistemas completamente homomórficos em nível

Suportam uma quantidade limitada de operações de adição **e** multiplicação.

Criptossistemas parcialmente homomórficos

Suportam adição **ou** multiplicação.

Criptossistemas completamente homomórficos

Suportam adição **e** multiplicação.

Criptossistemas completamente homomórficos em nível

Suportam uma quantidade limitada de operações de adição **e** multiplicação.

PROBLEMAS - RING LEARNING-WITH-ERRORS (RLWE)

Problema de decisão *RLWE*

Sejam **m** pares **(a, b)** amostrados da distribuição $A_{s,\chi_{err}}$ ou da distribuição uniforme.

O problema de decisão RLWE consiste em distinguir em qual caso cada par se enquadra.

Problema DSPR

Sejam

- d e q inteiros,
- $R_q = \mathbb{Z}_q/(x^n + 1)$ um anel de polinômios com coeficientes em $\mathbb{Z}_q$,
- χ uma distribuição sobre R_q,
- $y_i \in R_q$ e $z_i = -y_i t^{-1} \mod q$, para $i \in \{1, 2\}$.

Problema DSPR

Sejam

- d e q inteiros,
- $R_q = \mathbb{Z}_q/(x^n + 1)$ um anel de polinômios com coeficientes em $\mathbb{Z}_q$,
- χ uma distribuição sobre R_q,
- $y_i \in R_q$ e $z_i = -y_i t^{-1} \mod q$, para $i \in \{1, 2\}$.

Define-se $a = y_1 + t \cdot \chi_{z_1}$ e $b = y_2 + t \cdot \chi_{z_2}$.

O problema DSPR consiste em distinguir elementos da forma $h = a/b$ de elementos amostrados uniformemente de R_q.

São considerados **difíceis**.

YASHE - YET ANOTHER SOMEWHAT HOMOMORPHIC ENCRYPTION

- Criptossistema completamente homomórfico em nível.

YASHE - YET ANOTHER SOMEWHAT HOMOMORPHIC ENCRYPTION

- Criptossistema completamente homomórfico em nível.
- Baseado na proposta de Stehlé e Steinfeld.
  - Herda a segurança sobre os problemas *RLWE* e *DSPR*.
  - Oferece variante dependente apenas do *RLWE*.

YASHE - YET ANOTHER SOMEWHAT HOMOMORPHIC ENCRYPTION

- Criptossistema completamente homomórfico em nível.
- Baseado na proposta de Stehlé e Steinfeld.
  - Herda a segurança sobre os problemas RLWE e DSPR.
  - Oferece variante dependente apenas do RLWE.
- Criptogramas moram em um anel gerado pelo n-ésimo polinômio ciclotômico.

YASHE - YET ANOTHER SOMEWHAT HOMOMORPHIC ENCRYPTION

- Criptossistema completamente homomórfico em nível.
- Baseado na proposta de Stehlé e Steinfeld.
  - Herda a segurança sobre os problemas RLWE e DSPR.
  - Oferece variante dependente apenas do RLWE.
- Criptogramas moram em um anel gerado pelo n-ésimo polinômio ciclotômico.
- Resistente a ataques quânticos.

TEOREMA CHINÊS DO RESTO - CRT

TEOREMA CHINÊS DO RESTO - CRT

Um polinômio com coeficientes **grandes**

$\Updownarrow$

Vários polinômios com coeficientes **pequenos**

TEOREMA CHINÊS DO RESTO - CRT

Seja um conjunto de primos $\{p_0, p_1, p_2, \cdots, p_{l-1}\}$ e $P \in \mathbb{Z}_q/(x^{n+1}+1)$.

$$P(x) = \sum_{i=0}^{n} a_i \cdot x^i \iff \left\{ \begin{array}{ll} P(x) & \mod p_0, \\ P(x) & \mod p_1, \\ P(x) & \mod p_2, \\ \cdots & \\ P(x) & \mod p_{l-1}. \end{array} \right\}$$

Seja um conjunto de primos $\{p_0, p_1, p_2, \cdots, p_{l-1}\}$ e $P \in \mathbb{Z}_q/(x^{n+1}+1)$.

$$P(x) = \sum_{i=0}^{n} a_i \cdot x^i \iff \left\{ \begin{array}{ll} P(x) & \mod p_0, \\ P(x) & \mod p_1, \\ P(x) & \mod p_2, \\ \cdots & \\ P(x) & \mod p_{l-1}. \end{array} \right\}$$

$$M = \prod_{i=0}^{l-1} p_i > n \cdot q^2$$

TEOREMA CHINÊS DO RESTO - CRT

- Suporta:
  - Adição.
  - Multiplicação.
  - Redução polinomial por um polinômio ciclotômico.

TEOREMA CHINÊS DO RESTO - CRT

- Suporta:
  - Adição.
  - Multiplicação.
  - Redução polinomial por um polinômio ciclotômico.
- Não suporta:

TEOREMA CHINÊS DO RESTO - CRT

- Suporta:
  - Adição.
  - Multiplicação.
  - Redução polinomial por um polinômio ciclotômico.
- **Não** suporta:
  - Redução modular.

TRANSFORMADA DISCRETA DE FOURIER - DFT

- Reduz a complexidade de uma multiplicação polinomial.

TRANSFORMADA DISCRETA DE FOURIER - DFT

- Reduz a complexidade de uma multiplicação polinomial.
- $\Theta\left(N^2\right) \Rightarrow \Theta(N)$ no espaço da transformada.

TRANSFORMADA DISCRETA DE FOURIER - DFT

- Reduz a complexidade de uma multiplicação polinomial.
- $\Theta\left(N^2\right) \Rightarrow \Theta(N)$ no espaço da transformada.
- Utiliza uma raiz primitiva e N-ésima da unidade ω_N.
  - $\omega_N^N = 1$.
  - Para i variando de 1 até $N - 1$, $\omega_N^i \neq 1$.

TRANSFORMADA DISCRETA DE FOURIER - DFT

- Reduz a complexidade de uma multiplicação polinomial.
- $\Theta\left(N^2\right) \Rightarrow \Theta(N)$ no espaço da transformada.
- Utiliza uma raiz primitiva e N-ésima da unidade ω_N.
  - $\omega_N^N = 1$.
  - Para i variando de 1 até $N - 1$, $\omega_N^i \neq 1$.
- $\omega_N = e^{i\frac{2\pi}{N}}$.

TRANSFORMADA DISCRETA DE FOURIER - DFT

- Reduz a complexidade de uma multiplicação polinomial.
- $\Theta\left(N^2\right) \Rightarrow \Theta(N)$ no espaço da transformada.
- Utiliza uma raiz primitiva e N-ésima da unidade ω_N.
  - $\omega_N^N = 1$.
  - Para i variando de 1 até $N - 1$, $\omega_N^i \neq 1$.
- $\omega_N = e^{i\frac{2\pi}{N}}$.
- Custo quadrático para aplicação da transformada.

TRANSFORMADA DISCRETA DE FOURIER - DFT

- Algoritmos alternativos:

- Algoritmos alternativos:
  - Transformada Rápida de Fourier (FFT): $\omega_N = e^{i\frac{2\pi}{N}}$.

- Algoritmos alternativos:
  - Transformada Rápida de Fourier (**FFT**): $\omega_N = e^{i\frac{2\pi}{N}}$.
  - *Number-Theoretic Transform* (NTT): $\omega_N \in \mathbb{Z}_p$.

- Algoritmos alternativos:
  - Transformada Rápida de Fourier (**FFT**): $\omega_N = e^{i\frac{2\pi}{N}}$.
  - Number-Theoretic Transform (**NTT**): $\omega_N \in \mathbb{Z}_p$.
    - Na nossa construção: $\omega_N = r^k \mod p$, onde **r** é a raiz primitiva de um primo **p** e $k = \frac{p-1}{N}$.

- Algoritmos alternativos:
  - Transformada Rápida de Fourier (**FFT**): $\omega_N = e^{i\frac{2\pi}{N}}$.
  - Number-Theoretic Transform (**NTT**): $\omega_N \in \mathbb{Z}_p$.
    - Na nossa construção: $\omega_N = r^k \mod p$, onde **r** é a raiz primitiva de um primo **p** e $k = \frac{p-1}{N}$.
- Ambas tem custo log-linear para a aplicação da transformada.

IMPLEMENTAÇÃO

CUYASHE

- C++.

- C++.
- Palavras com precisão dupla.

- C++.
- Palavras com precisão dupla.
- Implementação comparativa da FFT e NTT.

- C++.
- Palavras com precisão dupla.
- Implementação comparativa da FFT e NTT.
- Paralelismo com CUDA.
  - Adição polinomial.
  - Multiplicação polinomial.
  - *CRT* e *ICRT*.
  - Aplicação das transformadas.
  - Redução polinomial e modular.

- C++.
- Palavras com precisão dupla.
- Implementação comparativa da FFT e NTT.
- Paralelismo com CUDA.
  - Adição polinomial.
  - Multiplicação polinomial.
  - CRT e ICRT.
  - Aplicação das transformadas.
  - Redução polinomial e modular.
- *NTL* para inteiros grandes na *CPU*.

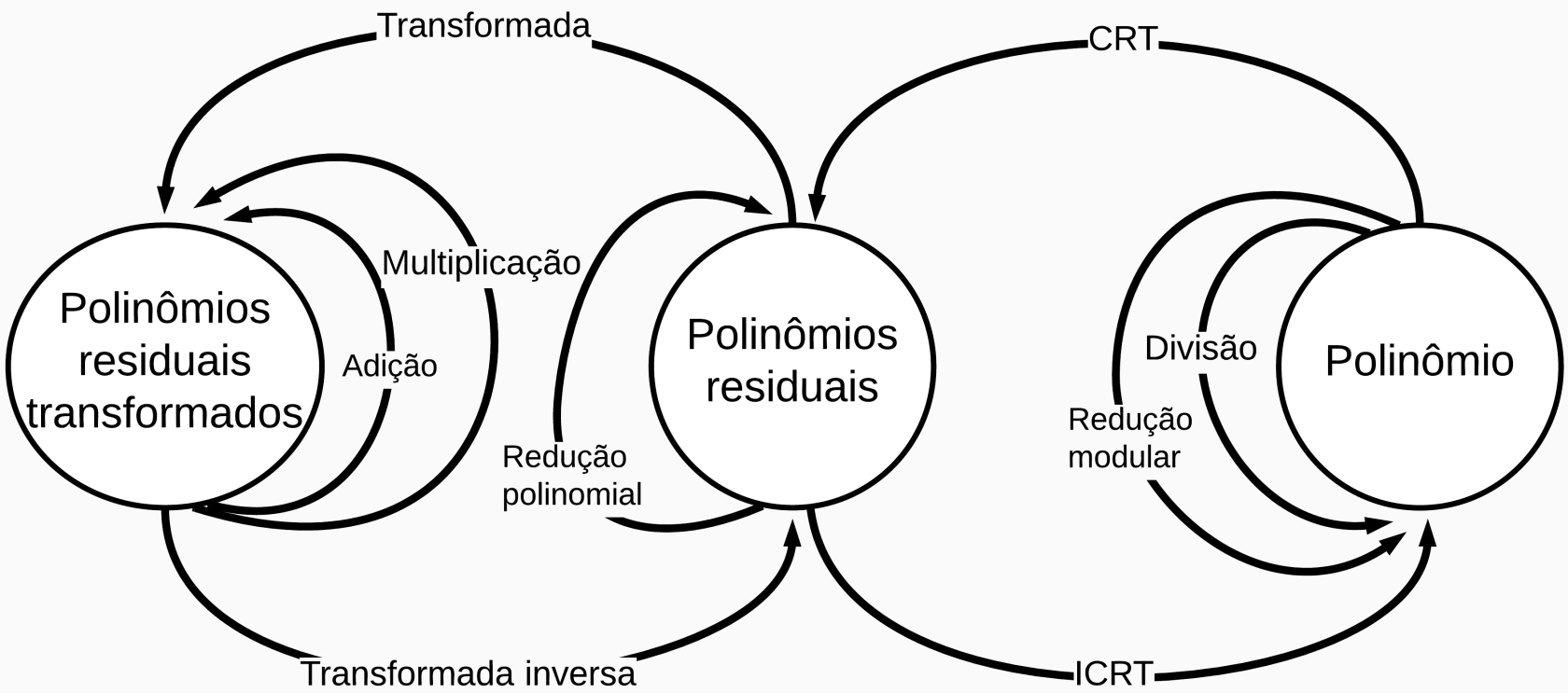
CUYASHE

- C++.
- Palavras com precisão dupla.
- Implementação comparativa da FFT e NTT.
- Paralelismo com CUDA.
  - Adição polinomial.
  - Multiplicação polinomial.
  - CRT e ICRT.
  - Aplicação das transformadas.
  - Redução polinomial e modular.
- NTL para inteiros grandes na CPU.
- Código portado da *RELIC* para inteiros grandes na *GPU*.
  - CRT e ICRT.
  - Redução modular.

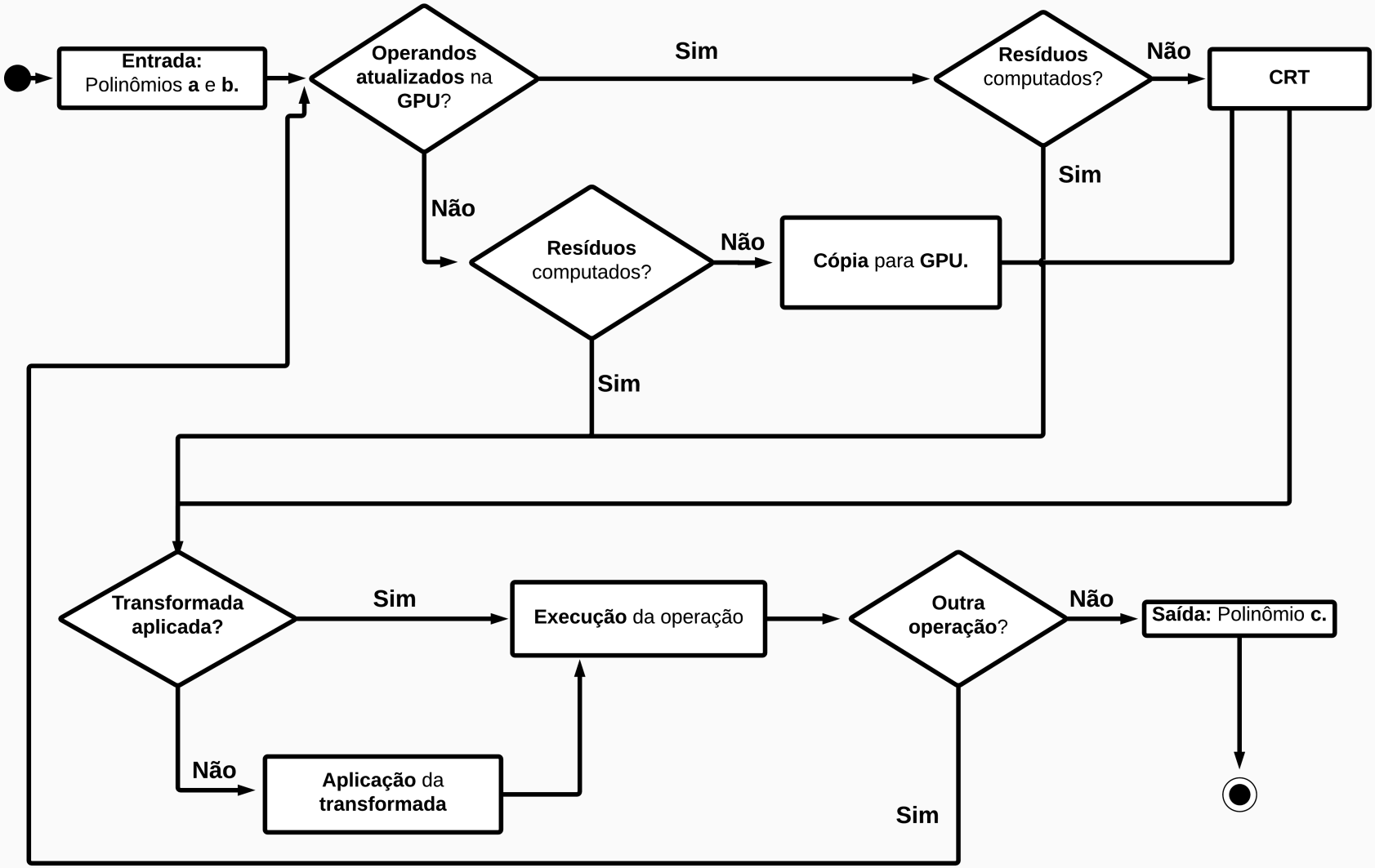
CUYASHE

- C++.
- Palavras com precisão dupla.
- Implementação comparativa da FFT e NTT.
- Paralelismo com CUDA.
  - Adição polinomial.
  - Multiplicação polinomial.
  - CRT e ICRT.
  - Aplicação das transformadas.
  - Redução polinomial e modular.
- NTL para inteiros grandes na CPU.
- Código portado da RELIC para inteiros grandes na GPU.
  - CRT e ICRT.
  - Redução modular.
- Distribuições probabilísticas construídas com a *cuRAND*.

CUYASHE - MÁQUINA DE ESTADOS



CUYASHE - DIAGRAMA DE FLUXO



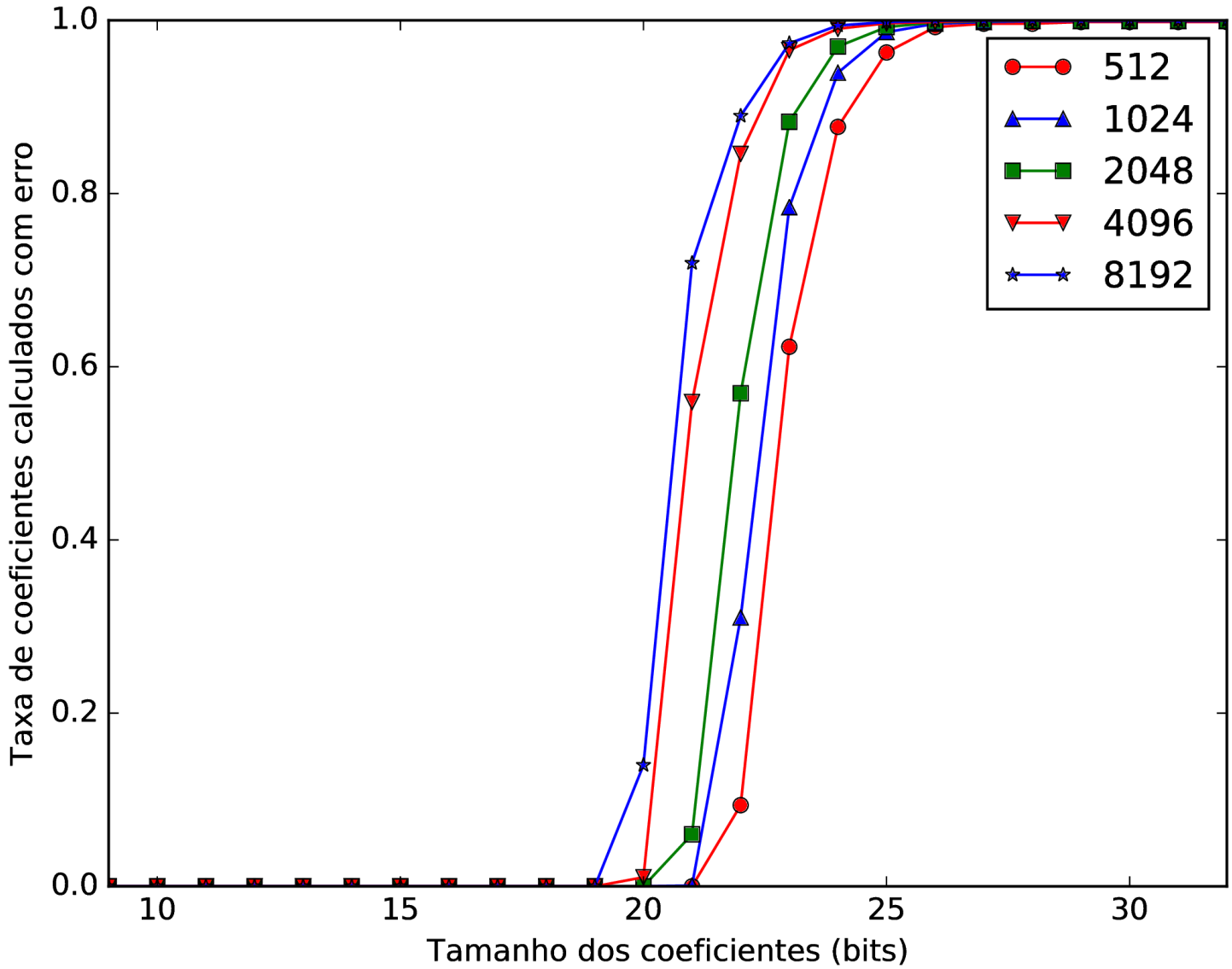
CUYASHE - ARITMÉTICA

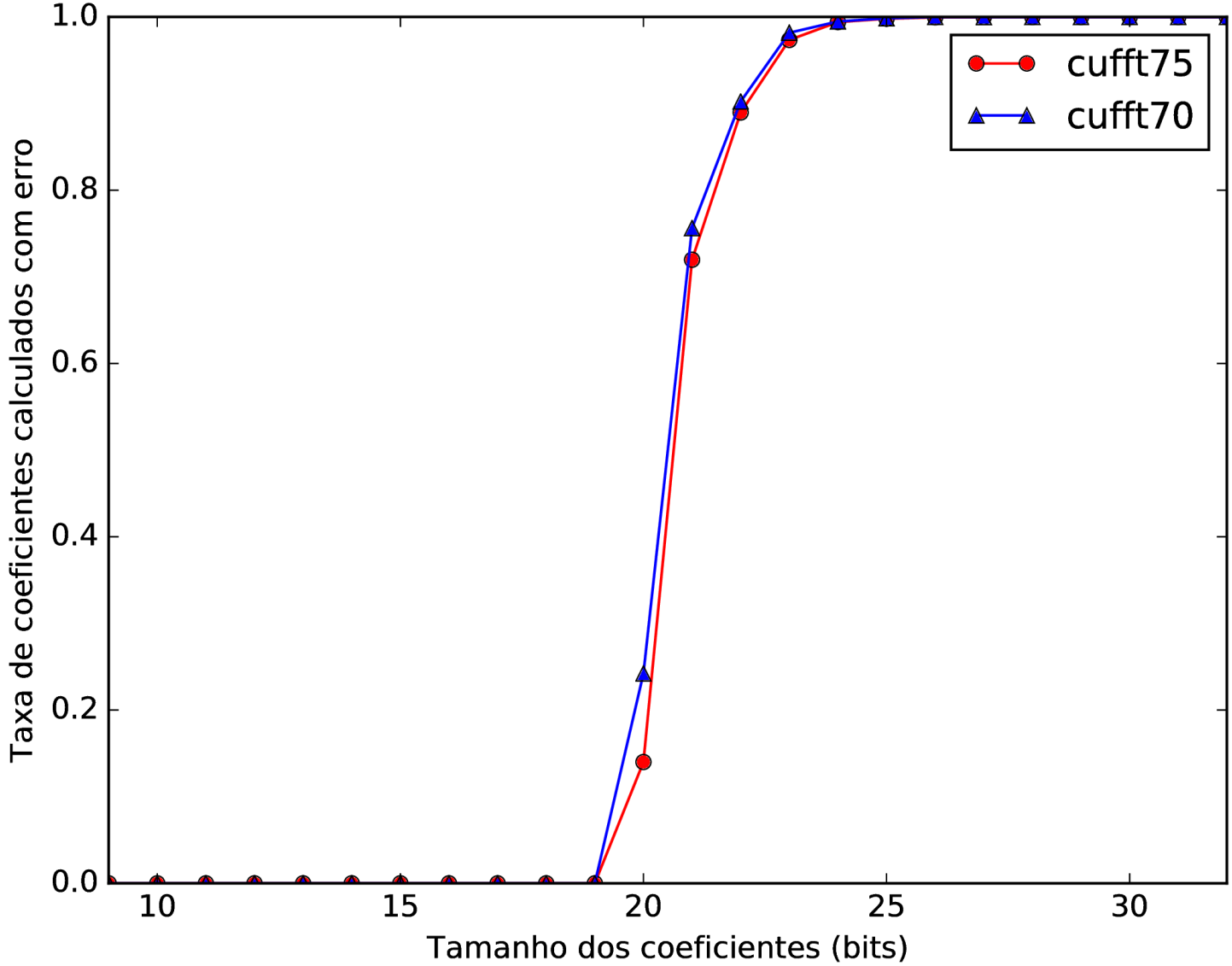
- Adição e subtração polinomial implementadas trivialmente.

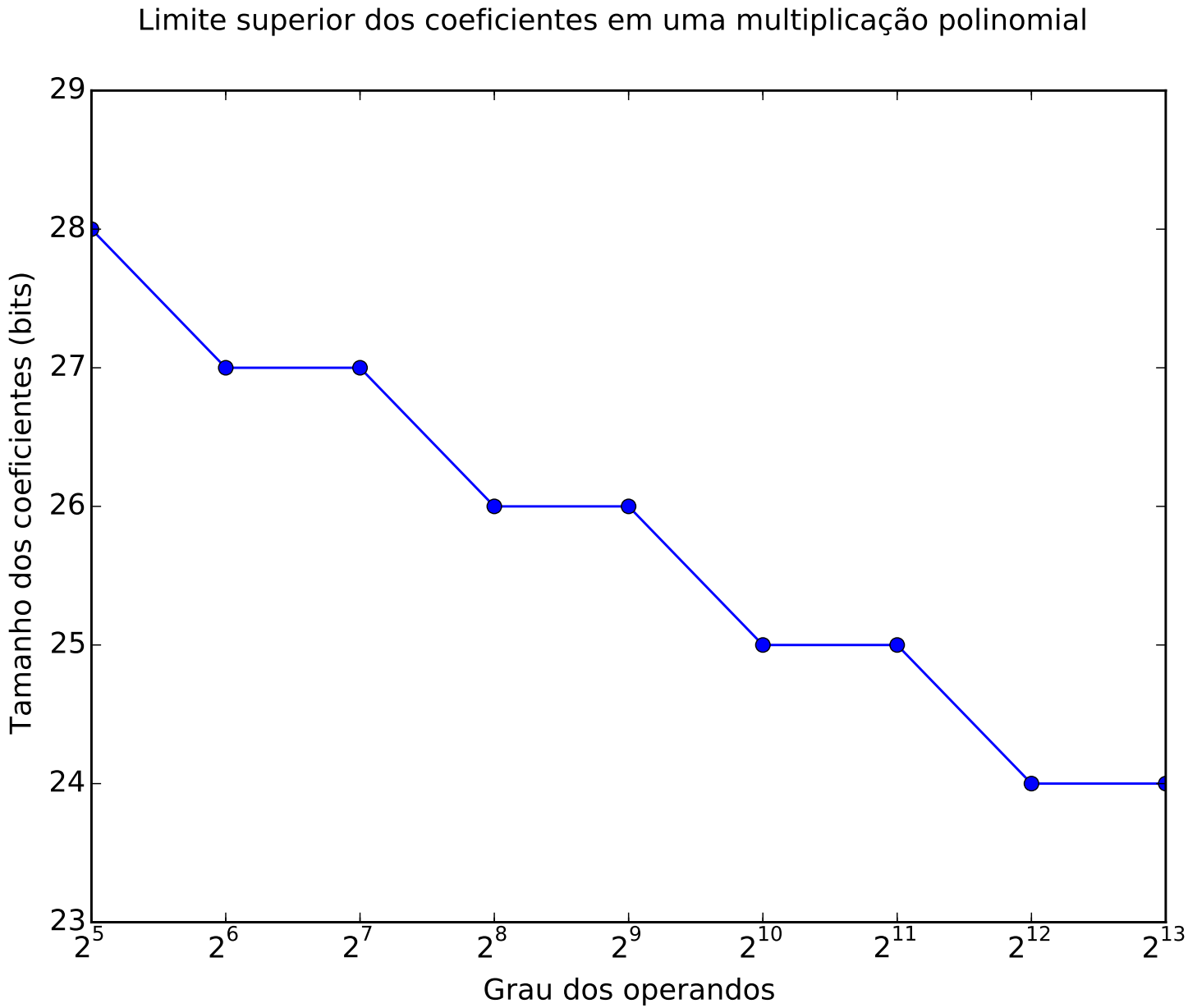
- Adição e subtração polinomial implementadas trivialmente.
- Implementação comparativa para a multiplicação polinomial entre FFT e NTT.
  - FFT fornecida pela biblioteca cuFFT.
  - NTT implementada com código próprio baseado na formulação de Stockham.
    - Algoritmo iterativo.
    - Dividir para conquistar.
    - Requer uma etapa de sincronização.

- Adição e subtração polinomial implementadas trivialmente.
- Implementação comparativa para a multiplicação polinomial entre FFT e NTT.
  - FFT fornecida pela biblioteca cuFFT.
  - NTT implementada com código próprio baseado na formulação de Stockham.
    - Algoritmo iterativo.
    - Dividir para conquistar.
    - Requer uma etapa de sincronização.
- Proposta pela utilização de um primo de Mersenne para gerar R_q.

- Adição e subtração polinomial implementadas trivialmente.
- Implementação comparativa para a multiplicação polinomial entre FFT e NTT.
  - FFT fornecida pela biblioteca cuFFT.
  - NTT implementada com código próprio baseado na formulação de Stockham.
    - Algoritmo iterativo.
    - Dividir para conquistar.
    - Requer uma etapa de sincronização.
- Proposta pela utilização de um primo de Mersenne para gerar R_q.
- Aproveitamento das propriedades de polinômios ciclotômicos em reduções polinomiais.


Erros em multiplicações de polinômios residuais
Taxa de coeficientes calculados com erro
Tamanho dos coeficientes (bits)
0.0
0.2
0.4
0.6
0.8
1.0
10
15
20
25
30
512
1024
2048
4096
8192


Comparação da taxa de erros para polinômios de grau 8192
Taxa de coeficientes calculados com erro
Tamanho dos coeficientes (bits)
cufft75
cufft70
1.0
0.8
0.6
0.4
0.2
0.0
10
15
20
25
30


Limite superior dos coeficientes em uma multiplicação polinomial
Tamanho dos coeficientes (bits)
29
28
27
26
25
24
23
2^5
2^6
2^7
2^8
2^9
2^{10}
2^{11}
2^{12}
2^{13}
Grau dos operandos

RESULTADOS

	CPU	GPU
Bos et al. 2013 (BLLN)	Core i7-3520M @ 2,89GHz	-
Lepoint and Naehrig 2014 (LN)	Core i7-2600 @ 3,4GHz	-
Nathan et al. 2016 (SEAL)	-	-
Dai and Sunar 2015 (cuHE)	Core i7 3770k @ 3,5GHz	GTX690/GTX770 @ 1GHz
Pöppelmann et al. 2015* (PNPM)	-	-
Alves and Aranha 2016 (cuYASHE)	Xeon E5-2630 @ 2,60GHz	GTX TITAN Black @ 0,98GHz

Tabela: Trabalhos utilizados como comparação.

*Utilizou a FPGA **Altera Stratix V 5GSND5H**.

RESULTADOS - PARÂMETROS DE CONFIGURAÇÃO DO YASHE

- $R = \mathbb{Z}[X] / \left(x^{4096} + 1\right)$,
- $q = 2^{127} - 1$,
- $w = 2^{32}$,
- $t = 2^{10}$.

Estes parâmetros definem um nível de segurança de 80 bits de acordo com Lepoint-Naehrig.

Função/Grau do operando	CPU (ms)	GPU (ms)	CPU/GPU
CRT / 1024	0,55	0,03	18x
ICRT/ 1024	7,74	0,43	18x
CRT / 2048	0,89	0,03	30x
ICRT / 2048	13,30	0,38	35x
CRT / 4096	1,80	0,05	36x
ICRT / 4096	26,86	0,54	50x

Tabela: Comparação entre os tempos de execução dos algoritmos direto e indireto do CRT em uma CPU e em uma GPU. A implementação na CPU faz uso de paralelismo pela biblioteca OpenMP.

RESULTADOS - COMPARAÇÃO ENTRE IMPLEMENTAÇÕES DA NTT

Grau	cuHE (ms)	cuYASHE (ms)	Ganho
8192	0,84	0,12	7x
16384	1,78	0,51	3,5x
32768	6,24	1,57	4x

Tabela: Comparação dos tempos necessários para a aplicação da NTT em um único polinômio residual.

RESULTADOS - COMPARAÇÃO ENTRE AS IMPLEMENTAÇÕES DA FFT E NTT

Grau	NTT (ms)	cuFFT (ms)	Ganho
1024	2,16	0,3	**7,2x**
2048	2,06	0,53	**3,9x**
4096	4,61	0,9	**5,1x**
8192	9,05	1,71	**5,2x**

Tabela: Comparação dos tempos necessários para a multiplicação de dois polinômios de certo grau utilizando a cuFFT e a implementação da NTT baseada na formulação de Stockham. O CRT foi executado com primos de 19 bits para a cuFFT e 24 para a NTT.

RESULTADOS - YASHE

Operação	cuYASHE(ms)	LN(ms)	Ganho	BLLN(ms)	Ganho
Cifração	1,85	16	**8,6x**	27	**14,6x**
Decifração	2,01	15	**7x**	5	**2,5x**
Adição H.	0,02	0,7	**35x**	0,02	-
Multiplicação H.	5,49	49	**8,9x**	31	**5,6x**

Tabela: Comparação entre a cuYASHE e as implementações LN e BLLN com tempos fornecidos pelos autores. Os tempos para multiplicação homomórfica incluem o custo de relinearização.

RESULTADOS - YASHE

Operação	cuYASHE(ms)	LN(ms)	Ganho	SEAL(ms)	Ganho
Cifração	1,85	15,4	**8,3x**	34,93	**18,9x**
Decifração	2,01	13,71	**6,9x**	34,1	**17x**
Adição H.	0,02	0,59	**30x**	0,18	**9x**
Multiplicação H.	5,49	31,07	**5,6x**	194,94	**35x**

Tabela: Comparação entre a cuYASHE e as implementações LN e SEAL avaliadas na mesma máquina. Os tempos para multiplicação homomórfica incluem o custo de relinearização.

Operação	cuYASHE(ms)	PNPM (ms)
Adição Homomórfica	0,02	0,19
Multiplicação Homomórfica	5,49	6,75

Tabela: Tempos para o cuYASHE e comparação com os resultados fornecidos pelo trabalho de PNPM. Os tempos para multiplicação homomórfica incluem o custo de relinearização.

CONSIDERAÇÕES FINAIS

CONSIDERAÇÕES FINAIS

- Implementação da biblioteca cuYASHE, disponível à comunidade sob uma licença GNU GPLv3*.

*https://github.com/cuyashe-library/cuyashe

CONSIDERAÇÕES FINAIS

- Implementação da biblioteca cuYASHE, disponível à comunidade sob uma licença GNU GPLv3*.
- Obtenção de ganho de desempenho em relação ao estado da arte.
  - Multiplicação homomórfica até 35 vezes mais rápida.
  - Cifração e decifração até 19x mais rápidas.

*https://github.com/cuyashe-library/cuyashe

CONSIDERAÇÕES FINAIS

- Publicações:
  - SBSeg 2015.
  - CSBC 2016.
  - Submissão para uma conferência internacional - Em processo de revisão.

CONSIDERAÇÕES FINAIS

- Publicações:
  - SBSeg 2015.
  - CSBC 2016.
  - Submissão para uma conferência internacional - Em processo de revisão.
- Trabalhos futuros:
  - Estudo da substituição do CRT pelo RNS.
  - Melhorar a implementação da NTT em comparação a cuFFT.
  - Aplicação da cuYASHE na implementação de um protocolo que preserve privacidade.

AGRADECIMENTOS

- Unicamp e Instituto de Computação
- Capes e CNPq, pelo financiamento.
- LMCAD, pelo ambiente de pesquisa.

Obrigado!

COMPUTAÇÃO SOBRE DADOS CIFRADOS EM GPGPUS

Pedro Geraldo M. R. Alves
Orientador: Diego F. Aranha

9 de junho de 2016

Instituto de Computação
Universidade Estadual de Campinas