Relatório: An Improved Data Compression Method for General Data de Salauddin Mahmud

Autores:
Adriano Almeida[1]
Emanoel de Mmoura[1]
Leonardo Trindade[1]
Marinara Rübenich[1]
Professor: Célio Trois[2]

[1]Curso de Sistemas de Informação
[2]Departamento de Linguagens e Sistemas de Computação
Universidade Federal de Santa Maria

02 de Julho de 2019

informática
Sistemas de Informação

Roteiro

informática
Sistemas de Informação

Roteiro

informática
Sistemas de Informação

Introdução

- Por que devemos comprimir nossos dados?

informática
Sistemas de Informação

Tipos de Compressão de Dados

Compressão **COM** Perdas

- *Na descompactação, o arquivo será restaurado exatamente como quando foi compactado*
- *Nenhum bit do arquivo original será eliminado*

Fonte: [MAHMUD, 2012]

Compressão **SEM** Perdas

- *Na descompactação, o arquivo não será restaurado conforme o original*
- *Alguns bits do arquivo original serão eliminados*

Fonte: [MAHMUD, 2012]

informática
Sistemas de Informação

Roteiro

informática
Sistemas de Informação

Metodologia Proposta

- Os bits dos dados podem ser representados pela metade dos seus bits
- Arquivos de 64 bits podem ser representados por 32 bits e assim sucessivamente
- Basta dividir o número de bits do arquivo original por 2 (Exemplo: 16 bits / 2 = 8 bits)
- O que também significa que 1GB de dados podem ser representados por sua metade: 512MB de dados

informática
Sistemas de Informação

Tabela Verdade Lógica Proposta

Tabela Verdade:

Tabela1: Tabela Verdade proposta

A	B	Z
0	0	0
0	1	$\bar{0}$
1	0	$\bar{1}$
1	1	1

Fonte: [MAHMUD, 2012]

Definições:

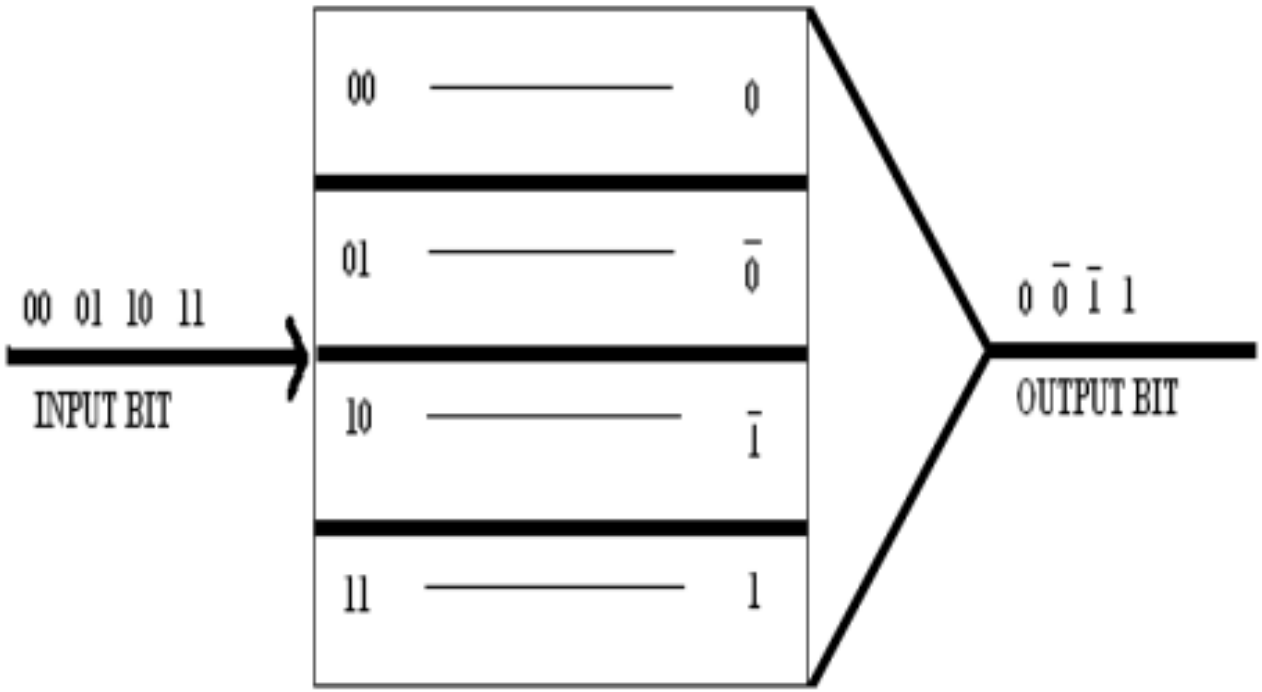
- **0:** significa que os dados A e B são 0 ($00 = 0$)
- $\bar{\mathbf{0}}$**:** significa que A é 0 e B é 1 ($01 = \bar{0}$)
- $\bar{\mathbf{1}}$**:** significa que A é 1 e B é 0 ($10 = \bar{1}$)
- **1:** significa que ambos são 1 ($11 = 1$)

Fonte: [MAHMUD, 2012]

informática
Sistemas de Informação

Representação da Técnica Proposta

Figura 1: Representação da técnica proposta



Fonte: [MAHMUD, 2012]

informática
Sistemas de Informação

Funcionamento

- Um valor X que recebe o array:
  X = {1,0,1,1,0,1,0,1,0,0,1,1,0,0,1,1,0,0,1,0}, onde X = 20
- Um array com 20 valores necessários para gerar o vetor Y:
  arr = array(0,1,0,1,0,0,1,1,0,1,1,0,1,0,1,0,1,0,1,0)
- Um valor Y que é gerado a partir do array arr randomizado (Y = arr[rand]): Y = {01010011011010101010}, onde Y = 20
- E finalmente o resultado Z que é proveniente do XOR (ou exclusivo) entre X e Y:
  Z = {1,1,1,0,0,1,1,0,0,1,0,1,1,0,0,1,1,0,0,0}, onde Z = 20.
  Aqui o arquivo já está criptografado e pronto para ser compactado

informática
Sistemas de Informação

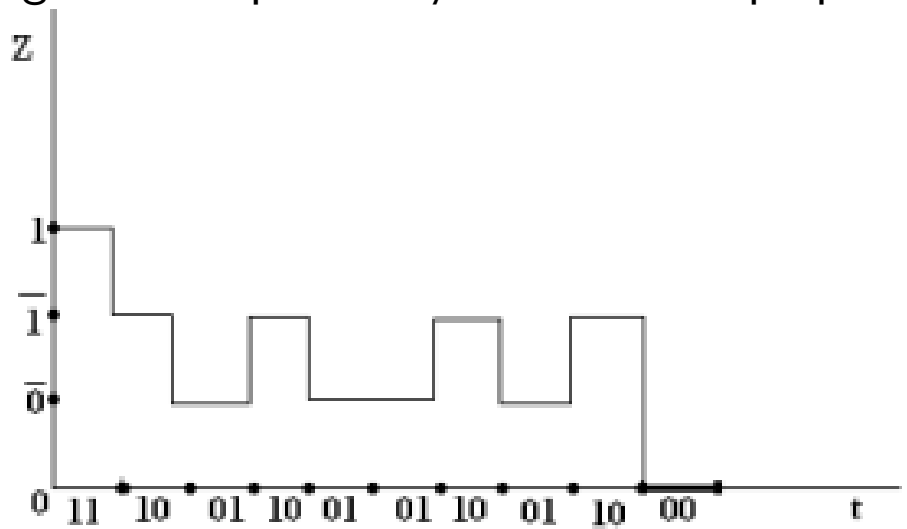
Resultado da Compactação

Tabela 2: representação da Compactação

11	10	01	10	01	01	10	01	10	00
1	$\overline{1}$	$\overline{0}$	$\overline{1}$	$\overline{0}$	$\overline{0}$	$\overline{1}$	$\overline{0}$	$\overline{1}$	0

Fonte:[MAHMUD, 2012]

Figura 2: Representação da técnica proposta



Fonte: [MAHMUD, 2012]

informática
Sistemas de Informação

- Para o cálculo de eficiência, o seguinte cálculo de taxa de compressão deve ser realizado:

$$\text{Taxa} = \frac{DadosCompactados}{DadosNãoCompactado} * 100\% \gg Taxa = \frac{10}{20} * 100\%$$

  - O que resulta em uma taxa de compressão de 50%

- Também existe o fator de compressão conforme mostrado abaixo:

$$\text{Compressão} = \frac{TamBitsEntrada}{TamBitsSaída} \gg Compressão = \frac{20}{10} = 2$$

  - Fatores acima de 1 significam que realmente houveram redução nos dados.

informática
Sistemas de Informação

Roteiro

informática
Sistemas de Informação

Desenvolvimento

- Foi desenvolvido um código na linguagem C que é capaz de fazer a compactação conforme a proposta de [Mahmud, 2012]

informática
Sistemas de Informação

Partes do Código Desenvolvido

Figura 3: main()

```
1 //PRINCIPAL
2 int main(){
3     int tamVetor, gerar;
4     int array[tamVetor];
5     int x[20] = {1,0,1,1,0,1,0,1,0,0,1,1,0,0,1,1,0,0,1,0};
6     int y[tamVetor]; //y gerado apartir do array
7     int z[tamVetor]; // vetor criptografado
8     int compacta[tamVetor/2]; // vetor comprimido
9     int descompacta[tamVetor]; // vetor descomprimido
```

Fonte: Autores

informática
Sistemas de Informação

Partes do Código Desenvolvido

Figura 4: Geração do Vetor Y

```
1  //NUMEROS RANDOMICOS
2  int serieRandomica(int min, int max){
3      return(rand() % (max - min)) + min;
4  }
5
6  // GERA ARRAY ALEATORIO
7  // O vetor aleatorio e gerado a partir de um random de 0 ou 1
8  int arrayAleatorio(int *array, int tamVetor){
9       int i;
10
11      for(i = 0; i < tamVetor; i++){
12          int random = serieRandomica(0,2);
13          array[i] = random;
14      }
15      return *array;
16 }
```

Fonte: Autores

informática
Sistemas de Informação

Partes do Código Desenvolvido

Figura 5: Geração do Vetor Z

```
1  // GERA O VETOR Z
2  // Vetor Z e resultado do (Xor X and Y)
3  int vetorZ(int *z, int tamVetor, int *x, int *y){
4      int i;
5
6      for(i = 0; i < tamVetor; i++){
7          int or = x[i]|x[i];
8          int and = (or != y[i]) ? 1 : 0;
9          z[i] = and;
10     }
11
12     printf("\n\t\t\t>>Vetor Z<<\n");
13     imprimeVetor(z, 0, tamVetor-1);
14
15         return *z;
16 }
```

Fonte: Autores

informática
Sistemas de Informação

Partes do Código Desenvolvido

Figura 6: Designação de valores Compactação / Descompactação

```
int comprimeDados(int val1, int val2){
    if(val1 == 0 && val2 == 0)
        return 0;
    else if(val1 == 1 && val2 == 1)
        return 1;
    else if(val1 == 0 && val2 == 1)
        return 2;
    else if(val1 == 1 && val2 == 0)
        return 3;
    else{
        printf("Error"); exit(0);
    }
}

int descomprimeDados(int val1){
    if(val1 == 0)
        return 00;
    else if(val1 == 1)
        return 11;
    else if(val1 == 2)
        return 01;
    else if(val1 == 3)
        return 10;
    else{
        printf("Error"); exit(0);
    }
}
```

Fonte: Autores

informática
Sistemas de Informação

Partes do Código Desenvolvido

Figura 6: Funções de Compactação / Descompactação

```
1  //RESULTADO DA COMPACTACAO
2  int compactacao(int *compacta, int tamVetor, int *z){
3      int i, j = 0;
4
5      for(i = 0; i < tamVetor; i+=2){
6          compacta[j] = comprimeDados(z[i], z[i+1]);
7          j++;
8      }
9
10     printf("\n\n\t\t\t>>Vetor Compactado<<\n");
11     imprimeVetor(compacta, 0, ((tamVetor-1)/2));
12     return *compacta;
13 }
14
15 //RESULTADO DA DESCOMPACTACAO
16 int descompactacao(int *descompacta, int tamVetor, int *compacta)
       {
17     int i, j = 0;
18
19     for(i = 0; i < tamVetor; i++){
20         descompacta[j] = descomprimeDados(compacta[i]);
21         j++;
22     }
23
24     printf("\n\t\t\t>>Vetor Descompactado<<\n");
25     imprimeVetor(descompacta, 0, ((tamVetor-1)/2));
26     return *descompacta;
27 }
```

Fonte: Autores

informática
Sistemas de Informação

Roteiro

informática
Sistemas de Informação

Conclusão

- A técnica proposta realmente funciona e traz resultados eficientes e satisfatórios
- A implementação e o entendimento do mesmo é bem simples, o que torna seu uso bem acessível e praticável

informática
Sistemas de Informação

Roteiro

informática
Sistemas de Informação

Referências

[MAHMUD, 2012] MAHMUD, SALAUDDIN. 2012. **An improved data compression method for general data**.

International Journal of Scientific Engineering Research, 3(3), 2.

Relatório: An Improved Data Compression Method for General Data de Salauddin Mahmud

Autores:
Adriano Almeida[1]
Emanoel de Mmoura[1]
Leonardo Trindade[1]
Marinara Rübenich[1]
Professor: Célio Trois[2]

[1]Curso de Sistemas de Informação
[2]Departamento de Linguagens e Sistemas de Computação
Universidade Federal de Santa Maria

02 de Julho de 2019