



令和2年8月31日

GPUによるハフマン符号の展開処理を高速化する新しいデータ構造  
「ギャップ配列」の考案  
～国際会議 International Conference on Parallel Processing (ICPP)  
で最優秀論文賞受賞！～

【本研究成果のポイント】

- 多くのデータ圧縮方式で用いられているハフマン符号の並列展開処理を高速化する新しいデータ構造「ギャップ配列」を考案しました。
- 圧縮処理時にこのギャップ配列を付加することにより、ハフマン符号の並列展開処理を大幅に高速化することが可能になります。
- GPU 向けに「ギャップ配列」を含めたさまざまな高速化手法を開発・適用した結果、同じGPUを用いた従来の最速展開手法より、2.5倍から11000倍の高速化が達成できました。

【概要】

広島大学大学院先進理工系科学研究科の中野浩嗣教授らの研究チームは、株式会社富士通研究所と共同で、GPUによるハフマン符号の並列展開処理を高速化する新しいデータ構造「ギャップ配列」を考案しました。さまざまなファイルに対してNVIDIA社のTesla V100 GPUを用いて実験した結果、従来の最速展開プログラムに比べ、2.5倍から11000倍の高速化が達成できました。

本研究成果は2020年8月開催の国際会議 International Conference on Parallel Processing (ICPP)において発表し、その結果269件の投稿論文の中から最優秀論文賞(Best Paper Award)に選ばれました。

【背景】

ネットワークを通して大量のデータを転送する場合や、ディスク等のストレージに保管する際には、データサイズを縮小する圧縮処理が行われます。そして、そのデータの処理を行う際に展開処理により元のデータを復元します。これにより、ネットワーク転送時間の短縮や、ストレージの使用量の削減が可能となります。ハフマン符号は、最も基本的なデータ圧縮方式で、多くのデータ圧縮方式に組み込まれており、より大量のデータの処理を可能とするため、ハフマン符号の圧縮・展開処理の高速化が望まれています。

近年、回路素子技術の改良による計算機の高性能化は頭打ちになっており、高速処理のためには、複数の処理を同時に行う並列処理が必須です。GPUは数千個のプロセッサコアをもっており、大量のデータの並列処理を行うのに幅広く用いられています。ハフマン符号の圧縮処理は並列化が比較的容易であり、GPUで高速に行うことができます。ところが、ハフマン符号の展開処理は並列化が困難です。そのため、多くの場合、無圧縮のままデータ処理が行われているのが現状です。GPUによるハフマン符号の展開処理の高速化が望まれており、本研究成果はこの課題を解決しました。

## 【研究成果の内容】

ハフマン符号（図 1）では、コードブックを用いてバイト記号を可変長ビットの符号語に変換します。バイト記号列が元データ、可変長符号列が圧縮データに対応し、バイト記号を可変長符号列に変換する処理が圧縮処理、その逆が展開処理になります。

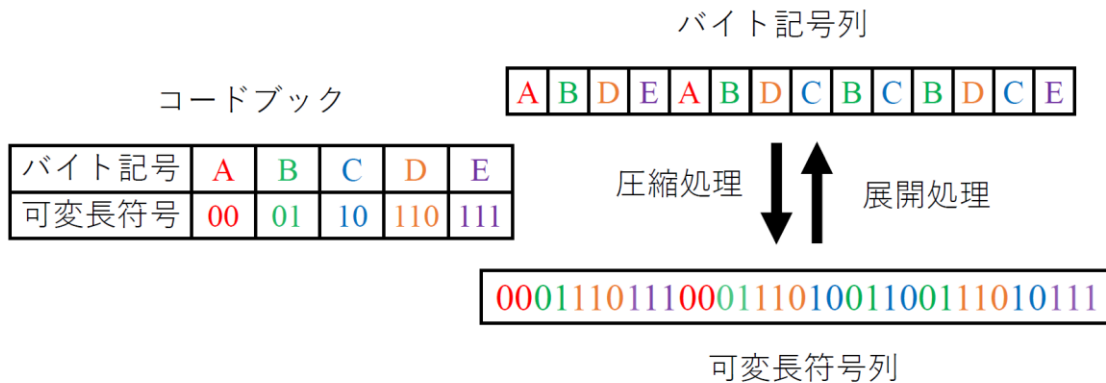
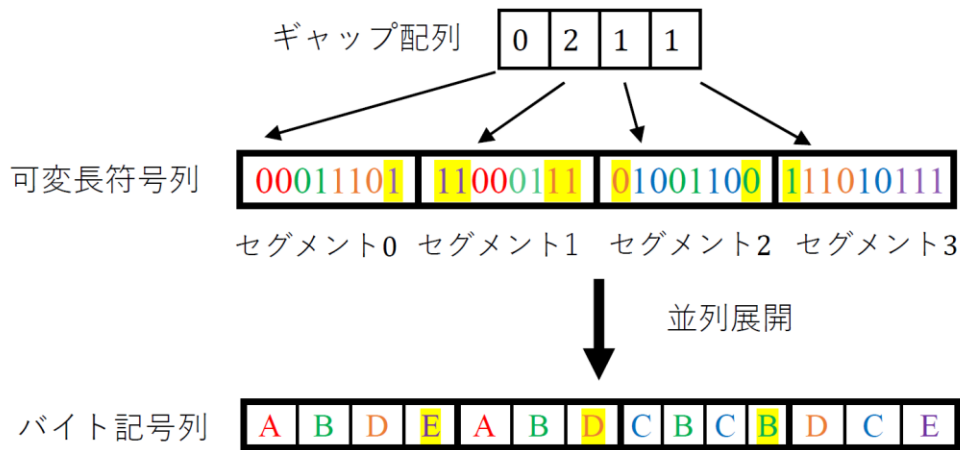


図 1：ハフマン符号の圧縮処理と展開処理

ハフマン符号の圧縮処理では各バイト記号を可変長符号に変換し、連結すればよいだけなので処理の並列化は容易です。一方、可変長符号列は 0 と 1 だけからなるビット列であり符号間の区切りはなく、個々の可変長符号を識別することができないため、ビット列の途中から展開処理を行うことができません。可変長符号列の先頭から逐次的にバイト記号列に変換する必要があり、GPU による並列展開処理が極めて困難でした。ハフマン符号の自己同期性をもちいて、GPU により投機的に並列展開処理を行う方法が 2018 年に Weißenberger らにより提案されましたが (Massively Parallel Huffman Decoding on GPUs. In Proc. of International Conference on Parallel Processing, 2018, DOI :10.1145/3225058.3225076), 同じ部分の展開処理を繰り返し行う必要があり、効率が悪いものでした。また、自己同期が起らない悪意のあるデータに対しては、膨大な計算時間が必要となってしまいます。

広島大学大学院先進理工系科学研究科の中野浩嗣教授らの研究チームは、株式会社富士通研究所と共同で、ハフマン符号の GPU による展開処理を高速化する新しいデータ構造「ギャップ配列」(図 2) を考案しました。ギャップ配列では、可変長符号列の同じ大きさのセグメント (256 ビット程度) への分割を考えます。1 つの可変長符号が 2 つのセグメントにまたがる場合がありますが、セグメントを構成するビット列だけではそれを認識することができません。そこでこのまたがる可変長符号の次の符号のビット位置 (ギャップ) を保存しておけば、セグメントの先頭の符号を識別することができます。全てのセグメントのギャップ値を並べたものがギャップ配列であり、その値を用いれば全てのセグメントを同時に展開することができます。圧縮処理時にギャップ配列を生成しておく処理が必要となりますが、その処理に必要な処理時間の増加は、わずか 4%-18% 程度です。また、ギャップ配列の分だけ圧縮データのサイズが増加しますが、増加量はたったの 0.4%-1.5% 程度です。

GPU でギャップ配列付きのハフマン符号の圧縮・展開処理を高速に行えるよう GPU のプログラミング技術を駆使した処理プログラムを開発しました。そして、NVIDIA 社の Tesla V100 GPU を用いて 10 個のさまざまなタイプのファイルを対象に性能評価を行いました。その結果、ギャップ配列のない従来のハフマン符号の圧縮処理の既知の最速プログラムに比べ、2.5 倍～7.4 倍の処理時間の高速化が行えました。また、展開処理については、Weißenberger らの開発したプログラムに比べ、2.5 倍～11000 倍の高速化が達成できました。



### 【今後の展開】

処理の一部にハフマン符号を含んでいる圧縮方式（gzip など）に今回開発したギャップ配列を組み込んで、これまでではできなかったGPUによる圧縮・展開処理の高速化を実現します。また、ビッグデータ、機械学習、クラウドコンピューティングなどの実際のアプリケーションにギャップ配列付きハフマン符号によるデータ圧縮を利用して処理時間の短縮を目指します。

### 【用語説明】

#### ●GPU

Graphics Processing Unit の略で、本来はグラフィックス専用処理のための集積回路ですが、グラフィックス以外の汎用計算用を行えるよう拡張されており、計算処理能力の高さから、多くのスーパーコンピュータに搭載されています。

#### ●ハフマン符号

1バイト（8ビット）の記号を可変長のビット列の符号に置き換える変換表（コードテーブル）を用いて、バイト記号列をビット列（可変長符号列）に変換するデータの符号化方法。頻出するバイトに対して短いビット列を割り当てることにより変換後のビット列の長さを短くすることができる。もっとも基本的なデータ圧縮手法であり、多くのデータ圧縮方式（gzip, zip, png, jpg 等）に組み込まれている。

#### ●ハフマン符号の自己同期性

ハフマン符号の可変長符号列の途中のビットから展開を開始した場合、最初は間違ったバイト列となってしまうが、ある程度処理が進むと正しいバイト列が確率的に得られるという性質。

### 【発表情報】

国際会議名：International Conference on Parallel Processing (ICPP) 2020

論文タイトル：Huffman Coding with Gap Arrays for GPU Acceleration

著者：Naoya Yamamoto, Koji Nakano, Yasuaki Ito, and Daisuke Takafuji (Hiroshima University) and Akihiko Kasagi and Tsuguchika Tabaru (Fujitsu Laboratories Ltd.)

DOI 番号：10.1145/3404397.3404429

国際会議 Web ページ：https://jnamara.github.io/icpp20/

受賞：ICPP 2020 Best Paper Award (269 件投稿中 1 位)

【お問い合わせ先】

大学院先進理工系科学研究科 中野 浩嗣

Tel : 082-424-5363 FAX : 082-424-5363

E-mail : nakano@hiroshima-u.ac.jp

発信枚数 : A4版 4枚 (本票含む)