ＬＺＷテキスト圧縮における静的辞書の利用方式*

1Ｊ－9

服部 芳明†　　吉浦 裕†　　大津 豊††
（株）日立製作所　システム開発研究所‡　（株）日立製作所　ソフトウェア事業所‡‡

１．はじめに

狭帯域、盗聴容易なインフラを用いるインターネット、モーバイル通信では、圧縮と暗号の両者を実行する機会が多い。そこで、我々は、両者の統合により、処理性能を向上する方式を提案した(1)。本論文では、上記方式の結果、生じたＣＰＵ能力の余裕を活かしての圧縮率向上方式を検討する。

標準的な圧縮方法より圧縮率の大きい方法として、静的辞書を併用するものがある。ところが、この併用方式は、辞書が大規模なため、性能が低く、配布が困難であった。しかし、これらの問題点は、ＣＰＵ性能の向上、可搬媒体やネットワークの進歩により、ほぼ解決されている。

本論文では、動的辞書のみを用いていた従来方式に、静的辞書を追加することによって、圧縮率を向上する方式を提案する。

２．従来のデータ圧縮技術

２．１動的辞書のみを用いる方式

圧縮方式については、従来多数提案されている(2)が、ここでは、標準の一つであるＬＺＷ(3)を代表例として取り上げる。ＬＺＷでは、入力データ中の文字列を動的辞書に記録し、同じ文字列が再度現われた場合に、それを辞書の番号に置換するが、以下の問題点があった。

（１）辞書が十分成長する前、および入力データの内容が途中で変化した場合、辞書が有効に使用されず、圧縮率が悪い。

（２）複数の相手と通信する場合、相手毎に異なる辞書を用いる必要があるため、メモリ量が急激に増大する。

２．２動的辞書と静的辞書の併用方式

上記の問題は、静的辞書と動的辞書の併用により、原理的には解決可能である(4)。すなわち、

（１）静的辞書に、予め範囲の広い頻出文字列を登録しておけば、動的辞書が未成長でも、入力データの内容が変化しても、効率的な圧縮が可能である。

（２）静的辞書は変化しないので、通信相手毎に持つ必要はなく、計算機に１つだけ持てばよい。

そこで、この併用方式に基づいて圧縮率の高い方式を開発することにした。

３．静的辞書の併用方式の提案

３．１併用における問題点

辞書の併用に基づいた高圧縮率の実現を具体的に試みた結果、以下の問題点が判明した。

問題１：静的辞書のメモリ量が膨大

前記（１）を実現するには、静的辞書に一度に数万個の文字列を登録する必要がある。従来の動的辞書の文字列数は、一般に数千個なので、約１０倍程のメモリ量が必要となる。

問題２：動的辞書への重複文字列の登録

静的辞書には、元々頻出文字列が登録されているので、動的辞書に静的辞書と同じ文字列が登録されてしまう。我々の実験では、動的辞書の７０％は、静的辞書と重複し、無駄になる。一方、重複文字列を動的辞書に登録しないようにすると、動的辞書の成長が阻害される。たとえば、静的辞書に"the"が登録してあると、動的辞書に"there"は、登録されない。

３．２基本方針

（１）問題１の解法

本質的には、「親文字列の番号（２バイト）」と「差分文字（１バイト）」の計「３バイト／文字列」でよい。ところが、動的辞書は、文字列の登録・削除を効率的に行うための余分な情報（例えば、子文

---

*The method with Static Dictionary in LZW

†Hattori YOSHIAKI, Yoshiura HIROSHI
††Ohtsu YUTAKA
‡Systems Development Laboratory, Hitachi Ltd.
‡‡Software Development Center, Hitachi Ltd.

字列へのポインタなど）を覚える必要があった。

これに対し、静的辞書は、検索のみを高速に行えばよい。そこで、配列の中で、文字列を適切にソートしておけば、余分なデータを不要化でき、「３バイト／文字列」が可能である。本方式は、更に、辞書構成を工夫することで、「約２バイト／文字列」を実現した。

（２）問題２の解法

いくつかの手法がメモリ動的辞書に登録する文字列を静的辞書との差分で表現した。

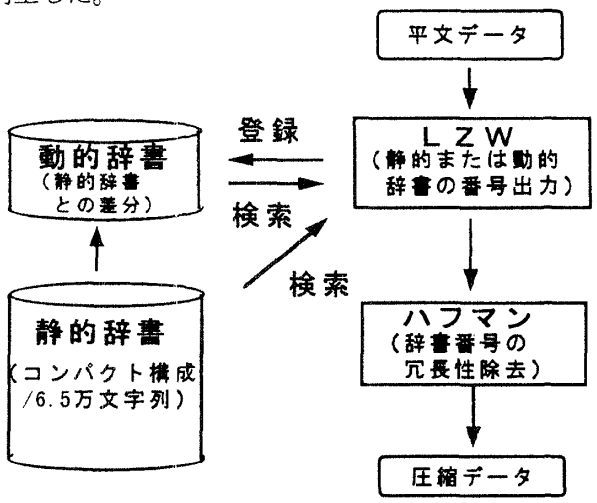
３．２静的辞書の利用方式（下図）

（１）出力辞書番号の選択

圧縮の各サイクルで、両者の辞書を利用し、圧縮率がより良くなる方の番号を出力する。

（２）ハフマン圧縮との連接

出力データは、「辞書に登録された文字列数に応じたビット表現」となり、冗長なビット数が残っている。しかし、「実際に使用された文字列数に応じたビット表現」にすれば、冗長ビット数を削減できる。そこで、ハフマン圧縮を連接することで、圧縮率を向上した。



４．評価

（１）システム概要

Ｃ言語約５Ｋステップのプログラムを Pentium133MHz のパソコン上で実行した。

（２）測定環境

２台のサーバと通信するクライアントを対象として測定した。この場合、従来は、大きな動的辞書を４つ必要とするが、本方式では、大きな静的辞書が１つあるので、４つの動的辞書を小さくできる。そのため、メモリ量が少なくとも、圧縮率は良くなった。

（３）結果

（ａ）圧縮率は、従来よりも約２０%向上。

（ｂ）メモリ量は、半分以下。

（ｃ）処理時間は、遅くなったが、ＩＳＤＮの通信速度よりも速く、ＣＰＵ性能の向上を前提とすると、処理時間短縮の改良を行えば、実行に耐えうる。

	入力データ（１００Ｋ）	従来のＬＺＷ		本方式	
圧縮率（出力/入力）	マニュアル	４８．１%		２５．０%	
	ニュース	５３．６%		３４．８%	
	論文	５１．８%		３２．４%	
メモリ量(byte)		１．８４Ｍｂｙｔｅ		０．３０Ｍｂｙｔｅ	
処理性能(bps)		圧縮	伸長	圧縮	伸長
		６Ｍ ｂｐｓ	９Ｍ ｂｐｓ	１６０Ｋ ｂｐｓ	２４０Ｋ ｂｐｓ

５．結論

本論文は、静的辞書と動的辞書の併用による圧縮率向上、メモリ量削減方式を提案した。

・静的辞書の特徴を活かした文字列の最適配置による省メモリ辞書

・静的辞書との差分による動的辞書の構成

により、従来のＬＺＷと比べ、圧縮率において約２０%向上した。

参考文献

1)吉浦 裕：モーバイル環境に適した圧縮／暗号通信方式（２）,第５２回情報処理学会全国大会講演論文集（１）,pp85-86,1995

2)M・ネルソン：データ圧縮ハンドブック（トッパン）

3)Welch, Terry,"A Technique for High-Performance Data Compression,"IEEE Computer, Volume 17, Number 6,June 1984, pages 8-19.

4)David K. Asano, Ryuji Kohno,"Intelligent Compression of English Text," The Proceedings of the 18th Symposium on Information Theory and Its Applications,Volume II of II,pp.577-580,1995