特別記事

ロスレス圧縮の技術と特許

圧縮ソフトSLC/ELCのアルゴリズム

井谷宣子/吉田茂(富士通研究所)

現在使われている多くの圧縮技術には特許が存在し、開発ビジネスの現場で利用するためには権利的な問題を解決する必要がある。そのため、独自の圧縮技術を開発し、それを利用することで特許問題を回避するケースもある。本稿では、富士通研究所が開発したロスレス圧縮技術を利用した圧縮ソフト「SLC」「ELC」のアルゴリズムに注目し、ロスレス圧縮技術の概要と特許状況を解説していく。

技術の流れと特許問題

データ圧縮は、情報量を落とすことなくデータ量を減らす技術です。ロスレス(lossless)圧縮(可逆圧縮)とロッシー(lossy)圧縮(非可逆圧縮)に分けられ、本稿ではロスレス圧縮について紹介します。

ロスレス圧縮は、元の情報を完全に保ったままデータ量を1/5〜1/2に減らす技術です。1ビットの欠けもなく元のデータを完全に復元することができるため、テキストやプログラムの実行ファイルなどの格納・通信に使われます。GZIP(GNU zip)やLHA、UNIX COMPRESSがその代表例です。

ビジネスでのロスレス圧縮技術(アルゴリズム)の利用にあたっては、特許をめぐるトラブルが数多く発生しており、知的財産権を侵害して提訴された場合の差し止めや高額ロイヤリティ請求の脅威を払拭できることが重要です。

圧縮技術は、1970年以前に、あらかじめ対象とするデータの種類を定め、それに最適な符号を生成する理論が完成していました[文献①]。70年代後半には、どのような種類のデータにも適応して対応できる方式の理論が確立されました。80年代には、この理論の実用化が検討され、多数の特許が出願されています。現在使われているロスレス圧縮技術の基本部分は、この時期に固められました。実際に普及したのは90年代に入ってからで、PC分野においてフリーソフトが主導して利用が広がりました。

しかし、80年代にはすでに特許が出願されていたため、利用が進むにつれて特許に絡む訴訟が発生しました[文献②〜⑤]。たとえば、1994年、MS-DOS6のディスク圧縮をめぐり米スタック社が米マイクロソフトを訴えた特許裁判があり、80億円相当の補償で和解しています。また、GIFに使われているLZW型圧縮[注1]に対し、ユニシス社が所有している特許をもとに特許料が徴収されました。GIFに関しては、2004年の6月に特許期限が切れたことから再び話題に登っており、記憶に新しいことと思います。

このように、広く使われているからトラブルと無縁というわけではなく、ビジネスで安心して利用するためには、特許の権利関係が解決されていることが必要です。

富士通研究所では、ロスレス圧縮がコンピュータシステムでのデータハンドリングにおける基盤技術であるため、方式の分類調査も含め、広く研究を行ってきました[文献⑦〜⑮]。そして、市場に普及しているフリーの圧縮ソフトは特許面が懸念されることから、自社の特許をベースに新方式の圧縮ソフトを開発しました。そうして開発された「SLC(Super Lossless data Compression)」と「ELC(Embedded Lossless data Compression)」は、主要な富士通のミドルウェア製品に搭載されており、多様なソフトウェア製品からハードウェア製品まで幅広く利用されています[文献⑦〜⑨]。

[注1]過去に現れたデータパターンに対して符号を割りふっていく、という考え方に基づく圧縮技術の1つ。データパターンを収めた辞書データを利用する。後出するLZ78型の1つ。

ロスレス圧縮の方式と課題

ロスレス圧縮の処理は、おおまかにモデリング部と符号化部の2つの部分に分けることができます。

モデリング部は、繰り返しの探索や統計量の計算を行います。たとえば文章が記録されたテキストファイルを圧縮するには、元データ(文章など)内で繰り返し出てくる単語を探索したり、特定の単語に続く文字の確率を求めたりします。

符号化部では、モデリング部で取得した情報に対して符号を割り当てます。数学的に最適な方法が求められており、速度と圧縮率のバランスをいかにとるかが命題です。このバランスは、前段階のモデリング部で用いる方法に依存するため、モデリング方法とあわせて調整することになります。

圧縮方式の検討においては、モデリング部の検討が主体になります。多くのモデリング方法が存在していますが、それらは大きく「統計型」と「辞書型」の2種類に分けられます[文献①]。統計型は単語などの短い繰り返しに対して高い性能が得られ、辞書型は文章など長い繰り返しに対して高い性能が得られます。

統計型モデリング

統計型は、特定の文字列(文脈)に続いて出る文字(通常は1バイト)の確率を算出し、その確率に符号を割り当てることでデータ量を削減します。理論的には辞書型より高い圧縮率が得られますが[文献⑦⑨⑩]、アルゴリズムが複雑で実用化が進んでいません。符号化単位が1文字で出現確率の算出処理が煩雑であるため、実用的な速度が得がたいことが課題です。

確率計算を簡略化することで高速化ができますが、簡略化は圧縮率低下を招きます。統計型の実用化には、根本的な改善が必要です。

辞書型モデリング

辞書型は、過去に現れた文章をそのまま辞書として記憶し、辞書と新しく読み込んだ文章を比較して繰り返しを探索します。繰り返し文字列を前出のコピーとして符号を割り当てることでデータ量を削減します。

アルゴリズムが単純で実用化が進んでいるのですが、特許に絡むトラブルが多く発生しています。先にあげたMS-DOSやGIFの訴訟は、辞書型の特許に関するものです。

富士通の圧縮方式

SLCは、統計型をベースにした方式です[文献⑩]。統計型をベースに、長い繰り返しに対しては辞書型を併用することによって、圧縮率を保ちつつ処理速度を上げました。SLCは、圧縮が速く、圧縮・復元がほとんど同じ処理量でバランスがとれている点と、高い圧縮率が特徴です。データの圧縮・復元双方を使用する用途に向いています。PCやサーバでの利用に最適です。

ELCは、辞書型をベースにした方式です[文献⑤]。特許面で問題となる繰り返し探索の実装に、独自の探索方法を用いています。ELCは、復元が速く、復元時の動作メモリが少ないことが特徴です。あらかじめ外部で作成したデータを転送・格納しておき、復元して使用する用途に向いています。この性質は携帯電話、PDA、情報家電などの機器での利用に最適です。

SLC

SLCの圧縮技術について解説します。

統計型と辞書型の併用モデル

SLCのモデリング部分では、統計型をベースに辞書型を併用します[文献⑩]。単語などの短い繰り返しには、短い繰り返しが得意な統計型を、文章などの長い繰り返しには、長い繰り返しが得意な辞書型を使用して圧縮します。この併用方法は、短い文脈では次文字の候補が複数あるのに対して、長い文脈では、次文字の候補が1つに限定されると仮定しています。

具体的な処理手順をFig. 1に示します。元データに含まれる3文字の出現位置をハッシュ表に随時登録し、符号化位置から見て直前3文字の過去の出現位置を保有しているか否かで辞書型と統計型を切り替えます。

SLCの辞書型の部分を取り上げて、従来の辞書型との違いを説明します。Fig. 2では、「compression」が繰り返し現れるデータを例にとっています。辞書型(LZ方式)の代表的な方法であるLZ77型の場合、繰り返し部分を過去に現れた位置「6」と繰り返しの長さ「11」で置き換えて圧縮します。SLCでは、繰り返し文字列の先頭部分「com」を残し(統計型で符号化)、残りの「pression」を長さ「8」で置き換えて圧縮します。

復元のときは、残された文字列の「com」の最近出現位置から、繰り返し文字列の位置を検出して元の文章を復元します。

スプレイ符号

SLCの符号化部分では、スプレイ符号[文献⑥]という動的可変長符号を使用します。スプ

Fig. 1 SLCの処理手順(統計型と辞書型を切り替える方法)

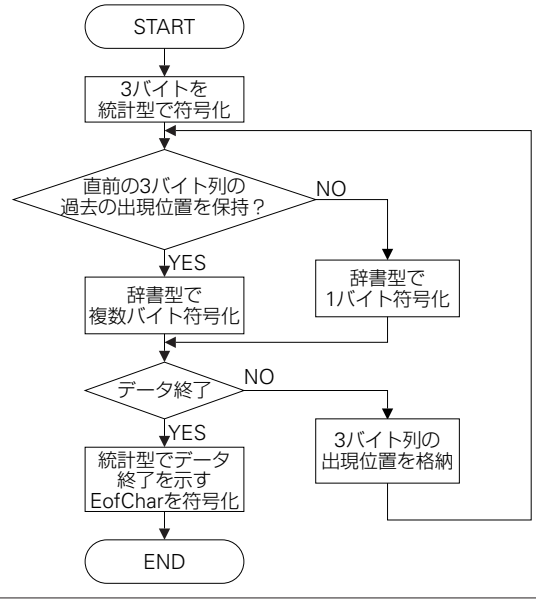
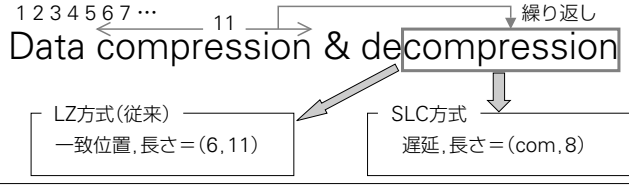


Fig. 2 SLCと従来方式の違い



レイ符号は、符号木と呼ばれる2分木を使用します。同様に符号木を用いるハフマン符号[注2]に比べて、シンプルで高速です。また、少ないデータ量でも、片寄りをつけて圧縮できることが特徴です。

符号木は、Fig. 3に示すように各葉に文字を割り当て、根から葉までの経路を符号とします。たとえば、Fig.3の符号木において文字Eは、根から経路をたどっていくことで「1100」という符号を得ることができます。

スプレイ符号は、出現した文字の符号長が半分になるように、順次、符号木を更新します。この更新を繰り返すことによって、出現頻度に応じた符号長が得られることになります。たとえば、Fig.4では文字Eが出現したことで、符号長が半分になるように符号木を更新しています。文字Eに対する符号「1100」は更新後「01」と4ビットから2ビットに短くなっています(右側の符号木)。更新のしかたは、葉から根へたどりながら、2つずつの組を作り、それぞれの組で目的の葉から根までの経路を1つ減らすように子供の節を交換します。Fig. 4では、節6と節5および節2と節0が子供の節を交換しています。

SLCでは、1バイトを1文字とみなし、0x0〜0xFFおよびファイルの最後を示す文字の257個の葉からなるスプレイ木を128個使用します。直前の7ビットの文脈に応じて木を切り替えて符号化します。

[注2]ハフマン木と呼ばれる木構造のデータを構築し、そこから各データに割り当てる符号を決定する圧縮技術。

ELC

次に、ELCの圧縮技術について解説します。

復元が高速な辞書型モデル

ELCのモデリング部分では、辞書型を使用します[文献3]。LZ77とLZ78の2つの辞書型方式のうち、高速復元に向いているLZ77型圧縮を使用します。LZ77型圧縮は、繰り返し文字列を探索し、過去に現れた出現位置と繰り返しの長さに置き換えて圧縮します。

LZ77型圧縮は、復元は高速なのですが、圧縮が遅いという課題があります。圧縮処理の大半を占めているのは、データ列の繰り返し探索です。従来は繰り返しの先頭部分を探索テーブルに随時登録し、先頭が一致した位置を軸に一致を求めることによって探索を高速化しています。この探索テーブルには、ハッシュ表や木構造などが使われていますが、LZ77型圧縮の実装には、多数の特許があり、争点になるところです。

ELCでは、探索テーブルを随時更新するのではなく、最初に一括作成することで、既存の特許と異なる構成にしています。使用する探索テーブルは、入力バッファにおける各アドレスに対して、もっとも近い過去に同じ3バイト列が出現した位置を格納したものです。以下では、「最近出現位置テーブル」と呼ぶことにします。

最近出現位置テーブルは、次に示す2段階の処理を経て生成します(Fig. 5)。

①3バイト列をコード順に並べ替える
②同じ3バイト列の1つ前の出現位置を求める

入力バッファから得られるすべての3バイト列をコード順に並べ替えます。たとえば、Fig. 5では、アドレス1「com」、アドレス2「omp」、アドレス3「mpr」……、をコード順に並べ替えます。並べ替えは、どんなソート法でもかまいませんが、ELCではいちばんシンプルに構成できるラディックスソート(Radix sort)を用いています。

②の段階では、①で求めたリストについて、頭から順に、隣接して格納している2つの3バイト列を比較し、最近出現位置を求めます。たとえば、Fig.5では、「com」の出現位置が1、15、24と並び、アドレス15「com」の1つ前の出現位置が1(相対位置は15-1=14)、アドレス24「com」の1つ前の出現位置が15(相対位置は9)であることが求められます。

こうして求めた「最近一致位置テーブル」から、符号化位置から始まる3バイト列の1つ前の出現位置、さらにその1つ前の位置と前方にたどると、同じ3バイト列が過去に出現した位置をすべて求めることができます。こうして得られた候補からもっとも長く一致する位置を探します。

固定長符号

ELCの符号化部分では、固定長符号を使用します。LHAやZIPでは、圧縮率を上げるために、探索した出現位置と一致長を、さらに可変長符号を使って符号化していま

Fig. 3 符号木

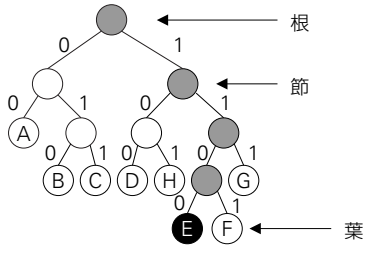
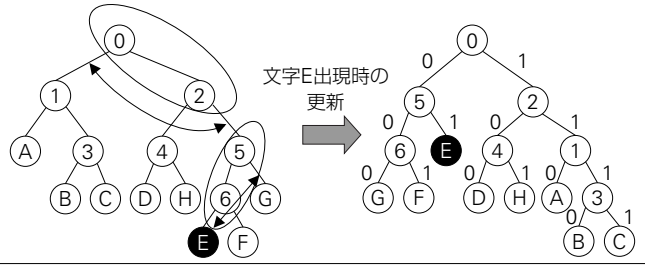


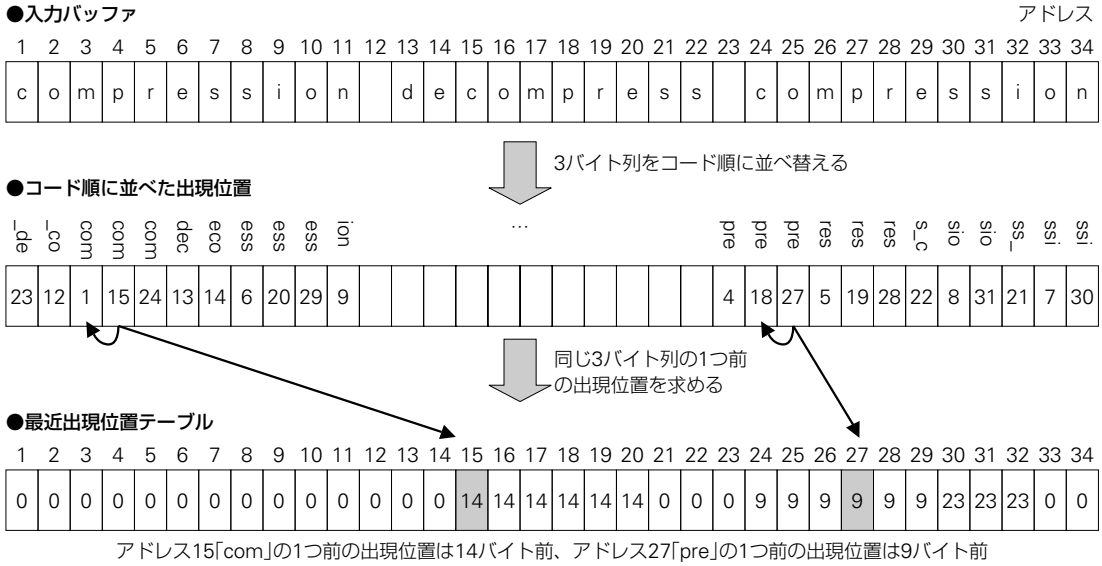
Fig. 4 スプレイ符号の更新



す。ELCでは、LZ77型圧縮の高速復元の特徴を生かすため、シンプルな固定長符号をベースに圧縮率を改善しました。Fig. 6にELCの符号構成を示します。基本構成として、シンプルな1ビットフラグ符号、8ビット単位の固定長符号を用います。一致長に3ビット、一致位置に13ビットを割り当てます。固定長符号の場合、このビットの割りふり方で圧縮率がかなり左右されます。一致長を長くしたほうが圧縮しやすいのですが、そのぶん一致位置のビットが短くなるため、探索領域が狭くなり長い繰り返しを見つけにくくなります。一般的には、一致長に4ビット、一致位置に12ビットがバランスがよいとされています。ELCでは、一致長拡張を用いて、基本構成のビット割り当てでは一致位置に13ビットを割り当て、より広い領域で候補を得るようにしています。

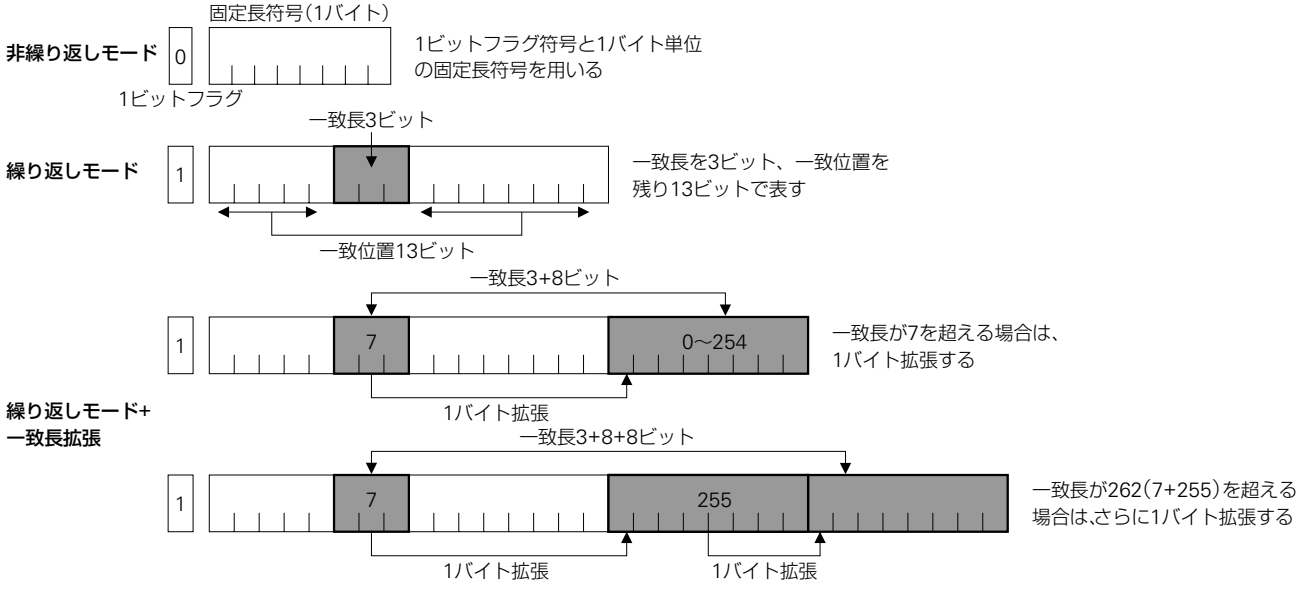
一致長拡張は、長く一致した場合には表現可能な一致長を長くできるように拡張したものです。一致長に割り当てたビットで表現可能な最大値(3ビットの場合には7(2進数で111))に達した場合、次に続く8ビッ

Fig. 5 最近出現位置テーブルの生成



アドレス15「com」の1つ前の出現位置は14バイト前、アドレス27「pre」の1つ前の出現位置は9バイト前

Fig. 6 固定長符号



トを一致長として用います。拡張した8ビットも同様に表現可能な最大値(255)の場合は、さらに次に続く8ビットも一致長として用います。

圧縮・復元速度と圧縮率

富士通の開発したSLCおよびELCと代表的な圧縮ソフト(GZIP、LHA、UNIX COMPRESS)の圧縮性能比較をFig. 7に示します。評価データには、ロスレス圧縮の性能評価によく用いられるcalgary corpusとcanterbury corpusを使用しました。圧縮率比較では、2つのcorpus(コーパス)から、圧縮しやすいものから圧縮しにくいものまで代表的な5つのデータを抜粋して測定結果を示します。処理速度比較では、5つのデータから圧縮しやすさで中間的なobj2の測定結果を示します。

SLCの特徴は、圧縮速度が速く、圧縮と復元で速度がほぼ等しい点です。圧縮がGZIPの1.2倍で最速となっています。圧縮・復元の双方が高速で、リアルタイム処理に向いています。リアルタイムのデータ転送では、圧縮と復元で遅いほうが全体の処理時間を決めることになるためです。また、統計型の特徴として、短い繰り返しが多い数値データで、ほかの圧縮ソフトと比較してもっとも高い圧縮率となっています。

ELCの特徴は、復元速度が速い点です。復元速度がGZIPの1.5倍で、最速となっています。また、圧縮速度もSLCに次いでELCが高速に動作しています。ELCをはじめ辞書型による圧縮の速度はデータ依存度が大きく、探索テーブルに登録した先頭文字列が多数回出現するようなデータでは、圧縮速度は低速になります。

まとめ

ロスレス圧縮が普及し始めた90年代に比べて、通信回線もHDDの容量も潤沢になりましたが、それに伴ってデータ量も増大しており、ますます圧縮技術が必要になっています。

ソフトウェア開発において、少ないリソースで動作させることが最大課題だった省リソース重視の時代から、潤沢なリソースを活用した信頼性・保守性重視の時代へと移ってきています。利用するデータ形式もXMLなどデータ量より運用しやすさを重視したものに移ってきています。これに伴って、データの冗長さが増え、圧縮しやすいデータが増えています。しかし、英文のプレインテキスト向けに発展してきた既存の圧縮技術では、冗長性に見合った圧縮が得られないケースも増えています。特定のデータ形式向けに前処理を用意することで圧縮率を改善できますが、データ形式ごとに試行錯誤が必要です。データ形式ごとに試行錯誤するのではなく、1つのモデリング方法で効率よく圧縮できる新しい技術の検討が今後の課題です。

◇参考文献一覧

①『文書データ圧縮アルゴリズム入門』
植松友彦、ISBN4-7898-3672-X、CQ出版(1994/10)

②「特許が見逃せない 米スタックと米IBMが基本押さえる」
浅見直樹ほか、日経エレクトロニクスNo. 580、P. 110-117(1993/5)

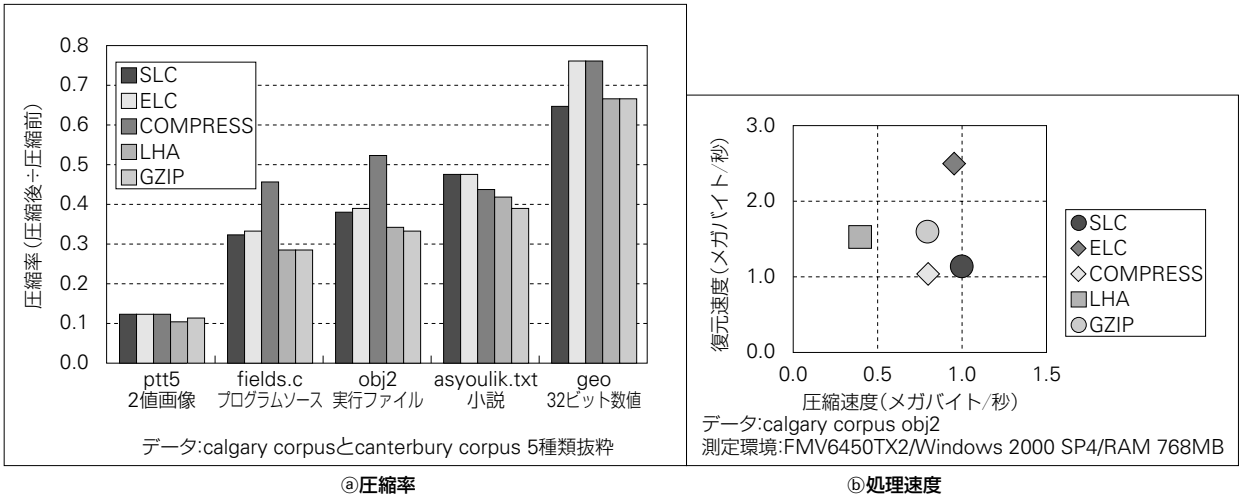
③「主流になるLZ方式、コンピュータの標準機能に」
加藤雅浩、日経エレクトロニクス別冊(98年版 データ圧縮とディジタル変調)、P. 229-233(1998)

④「データ圧縮アルゴリズム」
奥村晴彦、C MAGAZINE 1998年10月号、P. 52-63

⑤「データ圧縮の基礎から応用まで」
奥村晴彦、C MAGAZINE 2002年7月号、P. 13-35

⑥「Application of Splay Tree Data Compression」
Douglas W. Jones、Communication of ACM Vol. 31 No. 8、P. 996-1007(1998/8)

Fig. 7 圧縮性能の比較



⑦「**LZ方式を使わない独自方式 パソコンから大型機に対応**」
吉田茂ほか、日経エレクトロニクス別冊（98年版 データ圧縮とディジタル変調）、P. 235-242（1998）
⑧「**ロスレス圧縮アーカイバソフトSLCA**」
佐藤宣子ほか、雑誌FUJITSU 2001. 1、http://magazine.fujitsu.com/vol52-1/
⑨「**ロスレス圧縮技術　SLCAの特徴と製品紹介**」
http://www.labs.fujitsu.com/jp/gijutsu/lossless/product.html
⑩「**ブレンドスプレイ符号化方式の検討**」
村下君孝ほか、1994年電子情報通信学会秋季大会予稿論文番号SA-7-6（1994）
⑪「**Improvement of Sliding-Window Data Compression Using Splay-Tree Coding**」
吉田茂ほか、IEEE Data Compression Conf. 1994、P. 491
⑫「**High-Speed Statistical Compression using Self-Organized Rules and Predetermined Code Tables**」
村下君孝ほか、IEEE Data Compression Conf. 1994、P. 491
⑬「**ユニバーサル・データ圧縮の実用化動向と実現技法**」
吉田茂ほか、1994年電子情報通信学会秋季大会予稿 論文番号SA-7-3
⑭「**統計型圧縮とrepetition finderを併用する高速データ圧縮方式**」
佐藤宣子ほか、情報処理学会第59回 全国大会予稿 論文番号2G-3（1999）
⑮「**高速なLZ77型圧縮アルゴリズム**」
井谷宣子ほか、FIT2004第3回情報科学技術フォーラム予稿 論文番号A-036（2004）

ビジネス向けデータ圧縮の実装例

大野均（富士通デバイス）

SLC/ELCの実装製品

富士通デバイスでは、富士通研究所で研究開発したSLC、ELCを実装した製品の開発販売を行っています。

以下に示す製品が実際に販売されています。

①圧縮アーカイバ
「Arcmanager（アークマネージャー）」
②データ圧縮ライブラリ
「ESLC（イーエスエルシ：Embedded Super Lossless Compression）」
③データ圧縮ライブラリ
「RELC（レルク：Rapid Embedded Lossless data Compression）」

圧縮アーカイバArcmanager（SLC方式）の特徴を示します。

・複数ファイルの管理

複数のファイルやフォルダを1つにまとめて、一括で取り扱うことができます。

・自己復元（解凍）つき書庫ファイルの作成

Arcmanagerをインストールしていないマシンでも展開できる書庫を作成することができます。

・用途に応じて選べる製品構成

GUI、Console、DLLの3つの形態があります。画面操作による目視で取捨選択したデータを圧縮・復元する場合にはGUI製品を、バッチ処理で利用する場合にはConsole製品を、ソフトウェアシステムで利用する場合にはDLL製品を選択できます。

・パスワードの設定

書庫ファイルにパスワードをつけて、アクセスを制限できます。

※

次に、データ圧縮ライブラリESLC（SLC方式）/RELC（ELC方式）の特徴を示します。

・マルチプラットホーム対応

PC/ワークステーション/PDAなど、異機種間で圧縮データをやりとりできます。

・携帯電話・家電製品など組み込み機器対応

各種マイコン（FR、ARM、SH、MIPS）で利用できます。

・Java（iアプリDoJa）対応

サーバと携帯電話間のパケット削減に有効です。

圧縮製品の利用

データ圧縮ソフトの利用には、通信負荷を下げる効果や、従量課金の通信費用を安くする効果があります。また、必要とする記憶媒体の量を減らして費用を安くする効果があります。圧縮製品の利用例を以下に示します。

・ソフトウェアシステムへの組み込み

クライアント/サーバシステムで、CAD情報、販売・顧客情報など大容量のデータをFTP転送する通信負荷を低減します。また、Webサーバとクライアントシステム間のダウンロード、アップロードの通信パケット料を安くします。

・バッチ処理のデータ蓄積

データのバックアップ処理などで、転送時間を短縮します。

・外部記憶媒体によるデータ提供

マニュアルやアプリケーションプログラムをCD-ROMで提供するときの枚数を減らすことができます。

・機器への組み込み

機能アップして増えたプログラムやデータも容量の小さいフラッシュメモリに載せて機器の原価を下げることができます。

今後の展開

富士通デバイスでは、顧客の要望に応じたOS展開や仕様追加、カスタマイズなどにも対応していきます。最新情報および体験版のダウンロードは下記のURLを参照してください。

・http://www.fdi.fujitsu.com/
・http://www.arcmanager.com/
・http://www.arcmanager.jp/