Vol.2011-SLDM-149 No.1
Vol.2011-EMB-20 No.1
2011/3/18

消費エネルギーを意識した可逆圧縮データ受信

横山哲郎[†1] 藤井勝之[†1] 神山剛[†2]
冨山宏之[†3] 高田広章[†4]

情報通信端末において，データ受信と可逆データ解凍を逐次的に行うよりも，パイプライン化した方が消費エネルギーの最適化が実現されるケースが存在することが明らかになった．パイプライン化によって，マルチパスの圧縮コマンド `bunzip2` では消費エネルギーは悪化してしまったが，ストリーム処理が可能な LZ(Lempel-Ziv) 系のアルゴリズムの場合，顕著なエネルギー最適化が達成された．データ受信が安定的に行える環境下で，情報通信端末の消費電力を電力測定器によって計測したところ，逐次処理と比較してパイプライン処理は最大 22.8 %の消費エネルギー削減が達成された．

Energy-Aware Lossless Data Compression and Wireless Data Receiving

Tetsuo Yokoyama,[†1] Katsuyuki Fujii,[†1]
Takeshi Kamiyama,[†2] Hiroyuki Tomiyama[†3]
and Hiroaki Takada[†4]

It has been shown that there is a certain case in which the pipeline processing is more energy efficient than the sequential processing. Because of pipelining processes, LZ (Lempel-Ziv) family algorithms achieved significant energy savings while multi-path compression command `bunzip2` suffered energy degradation. The experimentation has been performed on a commercial data transmission device connected to a digital power meter under the stable data transmission. A pipelining processing outperformed a sequential processing by 22.8 % in terms of energy consumption.

1. はじめに

情報通信端末において，サーバからデータ取得時の消費エネルギー最適化を考える．無線データ受信では，データ通信量の削減のために，データの可逆圧縮が広く行われている[2)]．予め用意された圧縮データをサーバから情報通信端末に無線データ受信して，情報通信端末の上で解凍を行うのである．無線データ受信と比較して端末における計算に必要な消費エネルギーは遙かに小さいため，これは消費エネルギー最適化にも繋がる．

現在，無線データ受信と解凍の消費エネルギー最適化を考えるとき，多くの学術論文でこの 2 者は独立であるという理想的な仮定がなされている．すなわち，無線データ受信と解凍を別々に実行したときの消費エネルギーの和は，両者を同時に実行したときの消費エネルギーに等しいという仮定である．しかし，本稿では，パイプライン化を行うことで無線データ受信と解凍にかかるエネルギーを最適化することが可能であることを実験により示す．これは，与えられた通信環境・計算機環境によっては，逐次処理よりパイプライン処理がエネルギー最適であることを示している．

もちろん，パイプライン化がすべての場合で消費エネルギーを最適化するわけではない．パイプライン化による CPU 使用率の上昇は CPU の動作周波数を増加させ消費電力を増加させる．パイプライン化された複数タスク切換えには新たなコストが掛かる．データ受信の低速度だと低 CPU 使用率であるのに対して，解凍は高 CPU 使用率である．2 タスクの切換え後，CPU を最適周波数に設定するのにタイムラグがあるので，消費エネルギーが悪化するのである．高周波数と低周波数がそれぞれ最適なタス

†1 南山大学情報理工学部, Faculty of Information Sciences and Engineering, Nanzan University
†2 株式会社 NTT ドコモ先進技術研究所, Research Laboratories, NTT DOCOMO, Inc.
†3 立命館大学理工学部, College of Science and Engineering, Ritsumeikan University
†4 名古屋大学大学院情報科学研究科, Graduate School of Information Science, Nagoya University

Vol.2011-SLDM-149 No.1
Vol.2011-EMB-20 No.1
2011/3/18

クを頻繁に切り換えるとき，このことはより顕著になる．後に見るように，実際，4 章の実験でも `bunzip2` では，パイプライン化によって消費エネルギーが悪化することが観察された．

しかし，同実験では，パイプライン化によって，LZ(Lempel-Ziv) 系の圧縮コマンドにおいて顕著な消費エネルギー最適化が実現できた．主な理由は次の 2 つである．第 1 に，データ受信のタスクのみでは CPU を使い切っていない場合，余力を用いて解凍を同時にすることで実行時間を短縮することができた．第 2 に，データ受信と解凍をパイプライン化することで，中間データをフラッシュメモリに格納する必要がなくなった．

本稿の概要は次の通りである．2 章では，圧縮・解凍を用いてデータ送受信の消費エネルギーを削減することを目標にした関連研究を述べる．3 章では実験で用いたコーパスや通信を安定化させるために構築した実験環境を説明する．実験結果は 4 章で述べる．最後に 5 章でまとめを述べる．

2. 関連研究

近年，データを圧縮して無線データ通信量を削減することで消費エネルギーを最適化する研究が行われている．これまでの研究では，無線データ受信と解凍の消費エネルギー最適化を考えるとき，通常この 2 者が独立であるという理想的な仮定がなされてきた．無線データ受信と解凍の消費エネルギーはそれぞれ独立に計測され，ネットワーク全体の最適化問題のパラメータとして用いられる．例えば，以下に述べる 3 つの研究の中でもこの仮定が行われている．Barr と Asanović[1] は圧縮送信と受信解凍にかかる消費エネルギーをそれぞれ最適化するために，両者に同じものを用いる対称的圧縮を用いるより，別のものを用いる非対称圧縮 (asymmetric compression) を用いた方が効果的であるケースを示した．Yu，Krishnamachari，Prasanna[7] は，可変圧縮率の圧縮を行えるセンサネットワークのデータ集約問題を考えた．最大圧縮率の圧縮を行うことは必ずしも消費エネルギー最適化につながらず，無線通信と圧縮にかかる消費エネルギーのトレードオフを考慮することが重要であることを示した．

表 1　圧縮コマンドとアルゴリズム
Table 1　Compression commands and their algorithms.

Command	Version	Main algorithm
`lzop`	1.08	LZO1X
`gzip`	1.4	LZ77
`zip`	3.0	Various
`lzma`	4.32.7	LZMA
`bzip2`	1.0.6	BWT and Huffman coding

Tavli，Bagci，Ceylan[5] は，あるセンサノードでデータの圧縮率を変更する動的圧縮フロー均衡（dynamic compression and flow balancing）戦略がネットワーク全体の寿命を伸ばすことを示した．データの圧縮率の変更は，一時解凍後に別の圧縮率で再圧縮することで実現していた．後に見るように，パイプライン化による消費エネルギー最適化が時に高効果であるので，以上の文献における問題についても，パイプライン化の効果を考慮するとさらに消費エネルギーの最適化が行えると考えられる．

3. 実験環境

実験を再現できるように，誰でもインターネットから容易に入手できる圧縮コマンドやコーパスを用い，通信条件が安定化するよう環境構築した．

3.1 圧縮コマンドとコーパス

UNIX 環境において広く使われている圧縮コマンドの中から，なるべく異なるアルゴリズムを採用しているものを選択して，本稿で用いた（表 **1**）．圧縮コマンドのオプションは何も指定せずにデフォルトのまま実行した．各アルゴリズムの詳細は，例えば，6) に詳しい．Barr と Asanović[1] も各アルゴリズムについて詳しく述べている．

`gzip` は，Lempel-Ziv coding (LZ77) によって与えられたファイルの圧縮を行う．LZ77 は，HTTP プロトコルのフィルタにも用いられている[2]．`lzop` は，`gzip` よりも圧縮・解凍の速度を向上することを目標として開発されたアプリケーションである．LZMA(Lempel-Ziv-Markov chain-Algorithm) は，LZ77 を改良したものである．解凍時において高速度・省

Vol.2011-SLDM-149 No.1
Vol.2011-EMB-20 No.1
2011/3/18

表 2　各圧縮コマンドによるファイルサイズ
Table 2　File sizes of each compression command.

command	Size(MB)	CR
(original)	10.00	100 %
lzop	2.73	366 %
gzip	1.77	565 %
zip	1.76	568 %
lzma	1.20	833 %
bzip2	1.17	855 %

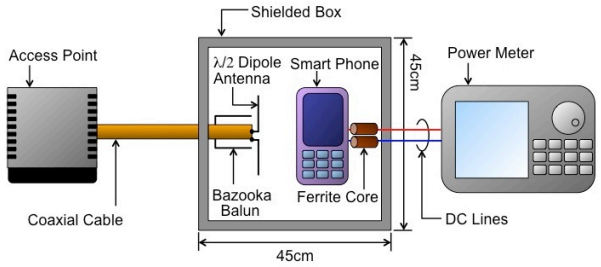

図 1　実験装置の概要
Fig. 1　Overview of the experimental setup.

メモリーでありながら，圧縮率を向上させている．bzip2 は，Burrows Wheeler 変換やハフマン符号化を順次適用することで，LZ 系のアプリケーションよりも高い圧縮率を実現している．ただし，複数パスであり実行時間は LZ 系のアプリケーションよりも遅い．zip は，様々なアリゴリズムを用いて圧縮・解凍を行うユーティリティーである．

DBLP[3](Digital Bibliography & Library Project) の XML レコードをダウンロードして先頭の 10MB を切り出したものを実験用コーパスとする．このコーパスを各コマンドで圧縮したファイルのサイズと圧縮率（CR: Compression Ratio）を，圧縮率の昇順に整列して表 2 に示す．ただし，本稿では，圧縮率は，$CR = \frac{\text{Original File Size}}{\text{Compressed File Size}}$ とする．

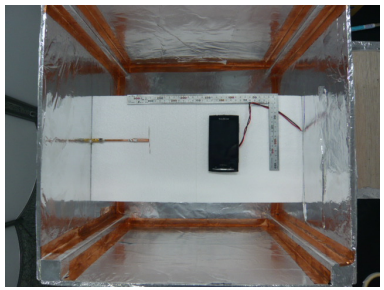

図 2　シールドボックスに設定された無線通信端末と無線 LAN 親機に接続されたアンテナ
Fig. 2　Wireless transmission device and the antenna connected to the wireless LAN station in the shield box.

3.2　実 験 装 置

実験装置の概要を図 1 に示す．無線 LAN 親機から伸びた同軸ケーブルの先端に λ/2 ダイポールアンテナを設置して，情報通信端末と無線 LAN 通信を行った．情報通信端末の電気は直流安定化電源菊水 PMC-18-5A から 4.00 V で供給され，電力測定器 WT1600 によってその電圧・電流が測定された．電流の入力範囲を 0.00-1.00 A とした．本稿におけるすべての実験はこの範囲内に収まった．測定電力の確度は読み値誤差 0.1 %，測定レンジ誤差 0.2 %（10 mW）であった．データ更新レートは最速の 50 ms に設定した．なお，附属電池パックには 3.6 V と記載してあったが，情報通信端末はこの電圧では起動しなかった．このため，完全充電された電池パックの実際の電圧 4 V に設定した．

再現性ある実験にするため，通信速度を一定にする必要があった．実験室内のサーバから単純に情報通信端末にデータ受信をすると，通信速度は大きくばらついた．これは無線 LAN 親機と情報通信端末の位置関係や環境によって，電波の反射・干渉・減衰が起こり，無線の信号強度が不安定であったためと思われる．また，情報通信端末の数ミリの移動や周囲での電波状況で信号強度 10 dB 以上の変化が観察された．

無線の信号強度を物理的に安定化させるために，情報通信端末と無線

Vol.2011-SLDM-149 No.1
Vol.2011-EMB-20 No.1
2011/3/18

表 3　実験で用いる情報通信端末
Table 3　Specification of the data transmission device.

Target	Xperia (SO-01B)
OS	Android 2.1
CPU	1 GHz Qualcomm Snapdragon
Instruction set	ARMv7
Frequency (MHz)	{245,384,576,768,998.4}

LAN 親機のアンテナをシールドボックスの中に発泡スチロールで固定した（図 **2**）．このことにより，シールドボックスの外部からの電波を遮断できた．無線 LAN 親機とアンテナを単に同軸ケーブルで接続した場合，信号強度が強すぎた．そこで，無線 LAN 親機の送信出力を 25%にして，さらにアッテネータで 39 dB 信号強度を減衰させた．その結果リンククオリティは 33-37 まで減少して安定した．

本稿で用いた情報通信端末のスペックを表 **3** に示す．Linux カーネルに含まれる CPUfreq[4] によって，CPU の動作周波数は制御された．本稿で扱う情報通信端末では，周波数を制御する governor は `ondemand` が設定されていた．`ondemand` は，本稿で用いたデフォルトの設定では，CPU 使用率を周期 0.5 s で確認して，上閾値（90 %）を上回ると最大周波数に，下閾値（60 %）を下回ると現在の周波数が 60 % 以下になるような中で最大の周波数に設定するようになっていた．なお，実験中，バックライトの輝度は最小に設定した．

4. 実験結果

解凍と無線データ受信を単独に行った結果について述べた後，両者をパイプライン化することでどういう変化があったかについて述べる．

4.1 解　　凍

表 **4** に，各コマンドで作成された圧縮ファイルの解凍に必要であったエネルギー（J），実行時間（s），電力（W），平均 CPU 使用率，平均周波数（MHz）を示す．エネルギー，実行時間，電力は，30 回の連続実行を 10 回以上計測して 1 回の実行に対応する値を計算したものであり，それぞれの値の直後の括弧の中に示された値はそれぞれの不偏標準偏差である．不偏標準偏差を求めるとき，連続実行した 30 回に関する測定値は独立同一分布に従うと仮定した．CPU 使用率はコマンドの実行前後の `/proc/stat` の値から求めた．平均周波数（MHz）は，各周波数に設定された時間が格納されているファイル[⋆1] の値から計算した．具体的には，各周波数に設定された時間にその周波数を掛けて合計を求めて，その合計を実行時間で除したものである．すなわち，各周波数に設定された時間にはアイドル時間も含まれてしまっている．この意味で，この値は，誤差を含んでいる可能性はある．なお，以降の表の測定値も同様にして求めた．

表 4　圧縮 XML ファイルの解凍
Table 4　Decompression of the compressed XML file.

Command	Energy(J)	Time(s)	Power(W)	Util.	Freq.(MHz)
(cp)	0.24 (0.01)	0.26 (0.024)	0.93 (0.024)	59.5 %	700
unlzop	0.48 (0.02)	0.51 (0.027)	0.95 (0.027)	77.4 %	743
gunzip	0.64 (0.01)	0.61 (0.007)	1.06 (0.007)	84.5 %	876
unzip	0.66 (0.01)	0.62 (0.011)	0.95 (0.011)	83.8 %	873
unlzma	1.34 (0.01)	1.08 (0.007)	1.24 (0.007)	88.0 %	906
bunzip2	4.72 (0.03)	4.98 (0.003)	0.95 (0.003)	97.0 %	875

測定消費電力は，すべて電力測定器の確度の誤差の範囲に収まった．測定消費エネルギーは，変動係数が最大の 4.2 %であった `unlzop` においても，約 95 %（$\pm 2\sigma$ の範囲．ここで σ は標準偏差）のデータが表に示した平均値の ±8.3% に収まる高精度であった．圧縮コマンドの実行時間のばらつきも，最大の `unlzop` でたかだか変動係数 5.3 %であった．

CPU 使用率は，高圧縮率のコマンドほど高くなった．同様に，周波数も，`unlzma` と `bunzip2` が逆転していることを除けば，高圧縮率のコマンドほど高くなった．もっとも，この 2 つの値の積で表される仕事量は，高圧縮率のコマンドほど高くなった．実行時間も高圧縮率のコマンドほど高くなった．したがって，本実験においては，高圧縮率の圧縮ファイルの解

⋆1 `/sys/devices/system/cpu/cpu0/cpufreq/stats/time_in_state`

Vol.2011-SLDM-149 No.1
Vol.2011-EMB-20 No.1
2011/3/18

表 5　圧縮 XML ファイルの無線データ受信
Table 5　Wireless Data receiving of the compressed XML file.

Command	Energy(J)	Time(s)	Power(W)	Util.	Freq.(MHz)
(original)	4.60 (0.13)	4.99 (0.025)	0.92 (0.025)	31.7 %	687
unlzop	1.14 (0.04)	1.19 (0.048)	0.95 (0.048)	33.5 %	879
gunzip	0.76 (0.03)	0.81 (0.046)	0.94 (0.046)	39.9 %	722
unzip	0.76 (0.06)	0.83 (0.054)	0.92 (0.054)	38.6 %	787
unlzma	0.53 (0.02)	0.55 (0.054)	0.96 (0.054)	36.5 %	800
bunzip2	0.51 (0.03)	0.55 (0.064)	0.95 (0.064)	38.7 %	789

凍ほど，CPU 使用率，CPU 使用量の両方で増大傾向があった．

4.2　データ受信

表 5 に，圧縮 XML ファイルの無線データ受信においての測定値を示した．データのばらつきは，圧縮ファイルの解凍を行った表 4 の場合よりも，大幅に大きかった．最大の変動係数は，共に bunzip2 においてであり，測定エネルギーで 5.9 %，実行時間で 11.6 %であった．

実験結果から，圧縮ファイルの無線データ受信は，電力はほぼ一定であり，消費エネルギー・実行時間・CPU 使用量は，ファイルサイズからだけで説明できた．通信速度は約 3.5 Mb であった．CPU 使用率は 40 %以下であり，他タスクの同時実行によりスループット向上が見込まれる．

4.3　データ受信＋解凍

表 6 に，圧縮 XML ファイルの無線データ受信・解凍を逐次的に，およびにパイプライン的に実施した結果を示した．なお，参考データとして，無圧縮で無線データ受信を行った結果を raw data として示した．パイプライン化できない unzip は，逐次処理の結果のみを示した．

図 3 に, 表 6 に示された実行時間と消費エネルギーをプロットし，また逐次処理の結果からパイプライン処理の結果へ矢印を引いた．

逐次処理の消費エネルギーと実行時間の実測値（表 6）は，bunzip2 を除くと，独立に測定した表 4 と表 5 の値の合計よりも悪化した．原因としては，最適ではない周波数で実行することによるオーバヘッド，タスク切り替えオーバヘッド，2 つのタスクに最適な周波数に変更するためのオーバヘッドなどが考えられる．

表 6　圧縮 XML ファイルの無線データ受信と解凍の逐次・パイプライン処理の比較
Table 6　Sequential and pipelining processing of wireless data receiving and decompression of the compressed XML files.

Command	Energy(J)	Time(s)	Power(W)	Util.	Freq.(MHz)
(raw data)	4.60 (0.13)	4.99 (0.024)	0.92 (0.024)	31.7 %	687
unlzop	1.78 (0.19)	2.08 (0.051)	0.86 (0.051)	50.2 %	602
unlzop(pipe)	1.56 (0.06)	1.53 (0.057)	1.00 (0.090)	56.5 %	714
gunzip	1.49 (0.16)	1.68 (0.099)	0.90 (0.099)	42.6 %	701
gunzip(pipe)	1.39 (0.10)	1.16 (0.083)	1.20 (0.083)	54.8 %	710
unzip	1.70 (0.15)	2.11 (0.050)	0.80 (0.050)	52.5 %	585
unlzma	2.16 (0.26)	2.21 (0.109)	1.00 (0.109)	65.3 %	707
unlzma(pipe)	2.02 (0.07)	1.49 (0.075)	1.36 (0.075)	64.8 %	769
bunzip2	4.66 (0.20)	4.81 (0.019)	0.97 (0.019)	81.5 %	829
bunzip2(pipe)	6.04 (0.55)	5.15 (0.064)	1.18 (0.064)	85.1 %	871

bunzip2 以外では，パイプライン処理化により性能向上とエネルギー最適化が達成できた．その度合いは，各圧縮コマンドにより異なった．最も性能が向上したのは，unlzop であった．unlzop は静的割り当てメモリが少なく，ハッシュテーブルがキャッシュに収まる様に意図されている．したがって，パイプライン化により中間データのメモリ書き出しが不必要になることで大幅に性能が向上したと思われる．実際，エネルギー遅延積では最良の結果を示した．gunzip や unlzma も window の一部をメモリに書き出す必要はあるかも知れないが，パイプライン化により少なくとも一部の中間データのメモリ書き出しが不必要になり，大幅に性能が向上したと思われる．最大の消費エネルギー削減は，unlzma の 22.8 %であった．これらはタスクの切り替えによるオーバヘッドが加わり，また単位時間当たりの仕事量が増加したため電力が増加したが，エネルギー効率の良い高周波数・高 CPU 使用率で実行でき，性能も向上したために，結果として消費エネルギーが最適化された．

bunzip2 がパイプライン化によって性能悪化や消費電力・消費エネルギー悪化に至った一因は，連長圧縮，ブロックソート，Move To Front，ハフマン符号化などを順次適用していくマルチパスのアルゴリズムである

Vol.2011-SLDM-149 No.1
Vol.2011-EMB-20 No.1
2011/3/18

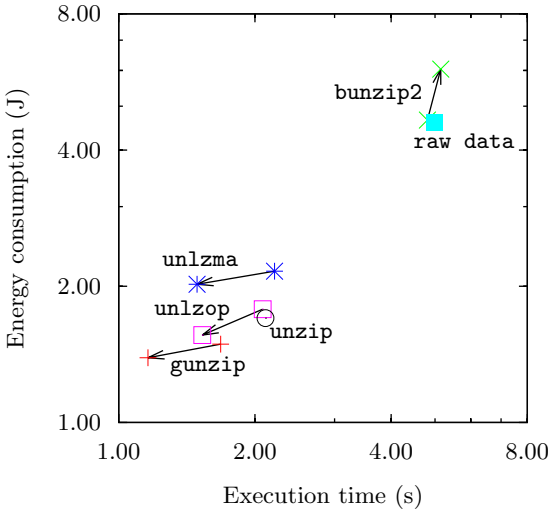

図 3 圧縮 XML ファイルの無線データ受信と解凍にかかった実行時間と消費エネルギーの実測値のパイプライン化による向上

Fig. 3 Improvement by pipelining wireless data receiving and decompression of the compressed XML files.

ことである．このため，パイプライン化してもタスクの並行実行が行えないため，タスク切り替えのオーバヘッドが悪影響を及ぼしたと考えられる．

図 3 の `bunzip2` の矢印の傾きが急であることから実行時間悪化以上に消費電力が悪化したことが読み取れる．このように，矢印の傾きが異なるコマンドがあるということから，単純に実行時間に定数を掛けることで消費エネルギーを見積もれないことが確認できた．

5. おわりに

LZ 系のアルゴリズムを採用している圧縮コマンドではパイプライン化により，消費エネルギーが大幅に最適化され得ることが観察された．各圧縮コマンドの各アルゴリズムの特性により消費エネルギーの削減比率と性能の向上比率の比は大きく異なった．すなわち，アルゴリズムによりパイプライン化による消費エネルギー削減効果の大小が存在した．

現在，通信ネットワークの消費エネルギー最適化にデータ可逆圧縮を用いる場合，圧縮解凍・送受信に必要な消費エネルギーは独立に測定されて最適化問題に用いられることが多い．しかし，本稿の結果は，パイプライン化を考慮すればさらに最適化が行えることを示唆している．

謝辞 本研究の一部は，財団法人中島記念国際交流財団の助成および 2010 年度南山大学パッヘ研究奨励金 I-A-2 によった．

参 考 文 献

1) Barr, K.C. and Asanović, K.: Energy-aware lossless data compression, *ACM Trans. Comput. Syst.*, Vol.24, No.3, pp.250–291 (2006).
2) Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Berners-Lee, T.: Hypertext transfer protocol – HTTP/1.1, Technical report (1999).
3) Ley, M.: The DBLP computer science bibliography: Evolution, research issues, perspectives, *Proceedings of the 9th International Symposium on String Processing and Information Retrieval*, Springer-Verlag, pp.1–10 (2002).
4) Linux kernel CPUfreq subsystem. `http://www.kernel.org/pub/linux/utils/kernel/cpufreq/cpufreq.html`.
5) Tavli, B., Bagci, I.E. and Ceylan, O.: Optimal data compression and forwarding in wireless sensor networks, *Comm. Letters.*, Vol.14, pp.408–410 (2010).
6) Witten, I. H., Moffat, A. and Bell, T. C.: *Managing gigabytes: Compressing and indexing documents and images*, Academic Press, second edition (1999).
7) Yu, Y., Krishnamachari, B. and Prasanna, V.K.: Data gathering with tunable compression in sensor networks, *IEEE Trans. Parallel Distrib. Syst.*, Vol.19, pp.276–287 (2008).