L-023

通信時の条件に応じたソフトウェアダウンロード方式の検討
An Optimization Method for Downloading Software Based on Communication Conditions

中原 大貴† 北山 健志† 西村 健† 古澤 康一†
Daiki Nakahara Kenji Kitayama Takeshi Nishimura Koichi Furusawa

1. はじめに

近年、製造業では PLM[1](Product Lifecycle Management)と呼ばれるソリューションが広まっている。PLM では、製品開発期間の短縮や生産工程の効率化のために、企画から設計、生産、出荷後のサポートやメンテナンスまで、製品に関わる全ての過程を包括的に管理する。これに伴い、複数のソフトウェアが連携し、設計や生産に関する多種多様なデータを扱う必要がある。これらの作業に関わる人員は多く、製品の関係者が使用するソフトウェアのバージョンの統一やソフトウェアで扱うデータの共有が必要になる。関係者間の連携作業のため、ソフトウェアを含めたこれらの多種多様なデータをサーバで集中管理し、クライアントへダウンロードして利用することが考えられる。しかし、これらのデータをダウンロードする場合、環境によっては多くの時間を要する。

そこで、本稿では通信時の条件に応じてソフトウェアのダウンロードに要する時間を短縮する方式を提案する。

2. ダウンロード時の課題

サイズが大きいデータをダウンロードする際は、一般的にサーバ側が特定の圧縮アルゴリズムでデータを圧縮してデータのサイズを小さくすることが多い。しかし、特定の圧縮アルゴリズムでデータを圧縮すると、条件によっては圧縮しない場合よりもダウンロードに要する時間が増大する。例えば、データの種類によっては圧縮効果が低く、データのサイズが小さくならないことが考えられる。このような場合、圧縮しない場合と比較して転送、圧縮/解凍に要する合計時間が増え、データの圧縮によってダウンロードの性能が悪化する。

このため、データを圧縮してダウンロードに要する時間を短縮する際は、転送、圧縮/解凍に要する合計時間を考慮しなければならない。

3. 提案方式

2章で述べたような課題を解決するために、データの種類、通信性能、サーバ及びクライアントの演算性能といった通信時の条件に応じてデータの種類ごとに圧縮アルゴリズムを選択する方式を提案する。

圧縮アルゴリズムはそれぞれ圧縮率や圧縮/解凍速度といった性能が異なり、これらの性能は互いにトレードオフの関係になっていることが多く、どの性能に重点を置くかで最適な圧縮アルゴリズムが異なる[2]。

一方、データの種類によっても圧縮効果は異なる。一般的にデータの可逆圧縮に用いられるアルゴリズムは、圧縮対象のデータから冗長性を取り除くことによってデータのサイズを小さくする。例えば、ソフトウェアのバイナリ形式の実行ファイルは冗長性が高く、圧縮効果が高い傾向がある。一方、PDF ファイルなどの既に圧縮されているファイルは冗長性が低く、圧縮効果が低い傾向がある。

以上のことから、通信時の条件に応じてデータの種類ごとに圧縮アルゴリズムを選択することにより、特定の圧縮アルゴリズムを使用する場合と比較して転送、圧縮/解凍に要する合計時間の短縮が見込める。

4. 提案方式の適用例

提案方式について、TCP/IP を適用した一般的なクライアント・サーバモデルにおいてソフトウェアをダウンロードするシステムを例として説明する。この例では、通信性能(転送速度)を RTT(Round Trip Time)から求める。また、演算性能を CPU のクロック周波数と使用率から求める。このシステムの構成と処理の順序を図 1に示し、以下で説明する。なお、図中の( )内の番号は処理の順序に対応しており、矢印はデータの流れを表す。

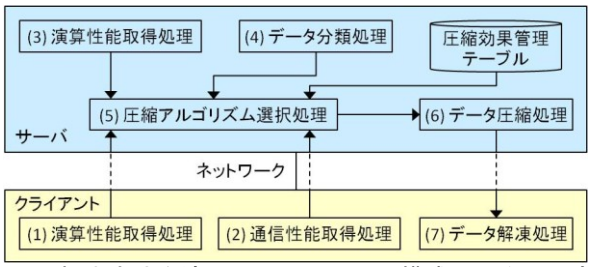

図1 提案方式を適用したシステムの構成と動作の順序

(1) クライアントの演算性能取得処理において、クライアントの CPU のクロック周波数と使用率を取得する。
(2) クライアントの通信性能取得処理において、クライアント・サーバ間の RTT を取得する。(1)と(2)で取得した値は両方ともクライアントからサーバに送信される。
(3) サーバの演算性能取得処理において、サーバの CPU のクロック周波数と使用率を取得する。
(4) サーバのデータ分類処理において、送信対象のファイルを、データの種類に応じて分類する。分類の仕方に関しては後述する。
(5) サーバの圧縮アルゴリズム選択処理において、(1)から(3)で収集した性能に関する情報を利用し、(4)で分類したデータの種類ごとに圧縮効果管理テーブルを参照して圧縮アルゴリズムの選択をする。圧縮効果管理テーブルには、各圧縮アルゴリズムと各データの種類に応じた圧縮効果が格納されている。圧縮アルゴリズムの選択方法に関しては後述する。
(6) サーバのデータ圧縮処理において、(5)で選択された圧縮アルゴリズムで圧縮対象のデータを圧縮し、圧縮データをクライアントに送信する。
(7) クライアントのデータ解凍処理において、受信した圧縮データを解凍する。 (4)で分類したデータの種類の数

† 三菱電機株式会社 情報技術総合研究所,
Information Technology R&D Center, Mitsubishi Electric Corporation

だけ(5)から(7)を実行し、全てのデータをクライアントで受信して解凍する。

(4)のデータ分類処理について、図 2を例にして説明する。この例では、Software というディレクトリの中に A、B、C というサブディレクトリが存在し、それらの中に合計 6 個のファイルが存在する。この内、拡張子から、a.pdf、b.pdf、d.chm(ヘルプファイル)を圧縮効果が低いファイルとして分類する。一方、c.dll、e.dll、f.exe は、圧縮効果が高いファイルとして分類する。このように送信対象のファイルを、データの拡張子単位で圧縮効果の傾向が似たもので分類し、それぞれに対して圧縮アルゴリズムを選択する。

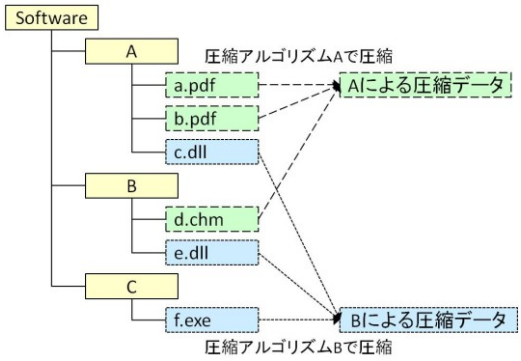

図 2 データ分類の例

(5)の圧縮アルゴリズム選択処理では、データの種類ごとに最適な圧縮アルゴリズムを選択するために、各アルゴリズムを使用した場合の圧縮データの転送、圧縮/解凍に要する合計時間を計算する。圧縮データの転送、圧縮/解凍に要する合計時間は以下の式で見積もる。

$$\frac{\text{元のファイルサイズ}\times\text{圧縮率}}{\text{転送速度}}+\frac{\text{元のファイルサイズ}}{\text{圧縮速度}}+\frac{\text{元のファイルサイズ}\times\text{圧縮率}}{\text{解凍速度}}$$

この式に含まれる圧縮率、圧縮/解凍速度は各圧縮アルゴリズムと各ファイルの種類から傾向が決まる。このため、各圧縮アルゴリズムと各データの種類に応じた圧縮効果が格納されている圧縮効果管理テーブルを予め持っておき、このテーブルから圧縮率と圧縮/解凍速度を取得する。圧縮/解凍速度には演算性能が影響するため、クライアントとサーバでそれぞれ取得した演算性能を考慮する。

5. 実例による考察

5.1 測定方法と測定結果

本方式の効果を考察するため、実例として 2 種類のデータに対して 3 種類の圧縮アルゴリズムで圧縮と解凍を実行し、圧縮/解凍速度と圧縮率を測定した。圧縮速度は（元のファイルサイズ / 圧縮時間）、解凍速度は（圧縮後のファイルサイズ / 解凍時間）、圧縮率は（圧縮後のファイルサイズ / 元のファイルサイズ）で求めた。測定環境を表 1に示す。

表 1 測定環境

項目	内容
CPU	Core™ i7 3.40GHz
メモリ	8GB(3.24GB 使用可能)
OS	Windows 7 32bit

圧縮と解凍には多数の種類の圧縮アルゴリズムを使用可能である 7-Zip[3]を使用した。圧縮アルゴリズムとして、速度重視、圧縮率重視、中間の性能を持つ、7-Zip で使用可能な以下の 3 つの圧縮アルゴリズムを使用した。

(1) 速度重視：Deflate
(2) 圧縮率重視：LZMA
(3) 中間：BZip2

圧縮と解凍の対象として、当社製のエンジニアリングツールを以下の 2 種類に分けて測定した。

(1) PDF、CHM(ヘルプファイル)などの圧縮効果が低いファイル
(2) DLL や EXE などの圧縮効果が高いファイル

3 種類の圧縮アルゴリズムによる測定結果を表 2に示す。

表 2 測定結果

	ファイルの種類	圧縮率	圧縮速度	解凍速度
Deflate	圧縮効果が低い	0.93	27MB/s	64MB/s
	圧縮効果が高い	0.24	15MB/s	6MB/s
LZMA	圧縮効果が低い	0.93	5.2MB/s	14MB/s
	圧縮効果が高い	0.13	5.6MB/s	4.3MB/s
BZip2	圧縮効果が低い	0.93	18MB/s	18MB/s
	圧縮効果が高い	0.20	18MB/s	4.7MB/s

5.2 考察

表 2の結果から、以下のことがわかる。

(1) 圧縮効果が低いファイルに関してはどの圧縮アルゴリズムでも圧縮率が変わらない。このため、速度重視の Deflate もしくは無圧縮にすることで、特定の圧縮アルゴリズムを使用する場合と比較してダウンロード時間の短縮が見込める。
(2) 圧縮効果が高いファイルに関しては、それぞれ圧縮率が異なる。このため、転送速度、圧縮/解凍速度を考慮し、4章で述べた計算式で処理の合計時間を見積もって最適な圧縮アルゴリズムを選択することでダウンロード時間の短縮が見込める。

6. おわりに

本稿では、通信時の条件に応じてデータの種類ごとに圧縮アルゴリズムを選択する方式を提案し、特定の圧縮アルゴリズムを使用する場合と比較してダウンロード時間の短縮が見込めることを実例によって考察した。本稿の例では、データを 2 種類にしか分けていないが、圧縮効果の違いによって更に細分化することによって本方式の効果を高めることができると考えられる。今後の課題として、本方式のプロトタイプの実装と評価、分類の細分化による効果の検証がある。

なお、Windows 7 は米国 Microsoft Corporation の米国及びその他の国における登録商標である。Core™ i7 は米国及びその他の国における Intel Corporation の登録商標である。

参考文献

[1]Saaksvuori Antti, Anselmi Immonen, “Product lifecycle management”, Springer (2008).
[2]山本 博資, “ユニバーサルデータ圧縮アルゴリズムの変遷—基礎から最新手法まで—”, IBIS2001 予稿集, pp.339-348 (2001).
[3]Igor Pavlov, “7-zip”, http://www.7-zip.org/ja/ .