論文

# 複数台コンピュータ協調による動画データ圧縮処理高速化の研究

高橋 邦博　　諏訪 敬祐

近年，様々なデジタル動画サービスが普及している．デジタル動画データはそのままの状態だと非常に大きなサイズになるので，データを圧縮し利用する．しかし，動画データ圧縮処理には非常に長い処理時間がかかる．本研究では，グリッドコンピューティングの技術を用いて動画データの高速圧縮処理を実現する．具体的には，ネットワークに繋がれた複数台のコンピュータに動画データ圧縮処理を分散することにより処理時間を低減するプログラムを作成した．実証実験を行い，２台で最大178%，３台で最大245%と高速化することに成功した．さらに，処理性能が異なる等複雑な情報環境において，データの処理単位となる分割数を端末数と同一とした従来の方式と比較し，分割数を端末台数より十分大きくすることにより最大139%高速化が図れることを明らかにした．

キーワード：グリッドコンピューティング，インスタントコンピューティング，動画データ圧縮，KNOPPIX

## 1　はじめに

近年，地上波デジタル放送やDVD・HDDへの録画等デジタル動画が普及しているが，そのままの状態ではデータサイズが大きいため，データの圧縮を行って運用する．しかし，デジタル動画データの圧縮処理は扱うデータサイズが大きいこともあり，処理時間が非常に長くかかる．また，低価格化によりコンピュータの普及が進み，現在では家庭において複数台のコンピュータを所有する場合も少なくない．グリッドコンピューティング技術により，ネットワーク上の複数台のコンピュータを組み合わせて安価／簡単に高性能なシステムを構築することが可能になり，KNOPPIXのような1CD-Linux等を用いることによって，一時的にグリッド環境を構築する技術も生まれてきている．

本研究では，グリッドコンピューティングの技術を用いてネットワークに繋がれた複数台のコンピュータに動画データ圧縮処理を分散して処理時間を低減するプログラムを作成する．特に本プログラムはグリッドコンピューティング技術において，これまでほとんど実績のない家庭という新規フィールドへの対応を主目的として開発し，家庭情報環境のようなネットワークや処理性能について各端末間で大きな差異のある画一的でない情報環境に対応する方法を明らかにするとともに，対応法をプログラムへと実装することを目標とする．作成したプログラムを利用した実証実験を行い，動画データ圧縮処理時間低減の効果を確かめるとともに，家庭環境を模した画一的でない情報環境での高速化の効果を検証する．

## 2　グリッドコンピューティングと動画データ圧縮処理

### 2.1 グリッドコンピューティング

グリッドコンピューティングとは，図１に表すように「遠隔地のリソース（資源）をネットワークを通じて利用すること」であり，ネットワーク上の複数のリソースに処理を分散し，高性能を達成する技術である[１]．主に地球シミュレータのようなスーパーコンピュータや大

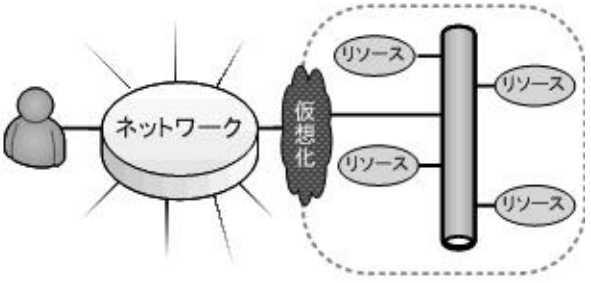


図１　グリッドコンピューティング模式図

TAKAHASHI Kunihiro
武蔵工業大学大学院環境情報学研究科博士前期課程 2007 年度修了生
SUWA Keisuke
武蔵工業大学環境情報学部情報メディア学科教授

規模データベースなどの分野で利用されている．ただ家庭のようなフィールドで実用化されている例は非常に少なく，今後の普及が見込まれている．

スーパーコンピュータの分野では，従来の同機種のみ（ホモジニアス）の分散処理から異機種混住（ヘテロジニアス）の分散処理へと移ってきているが，家庭のような特に複雑性の高い異機種混在環境を想定して定量的なデータを示した研究はほとんどない．本研究では，家庭のような特に複雑性の高い情報環境に対応する方法を明らかにするとともに対応法をプログラムとして実装し，実験に基づいた実測値として定量的なデータを示す．

### ２．２ インスタントコンピューティング

グリッドコンピューティング技術の１つとして CD ブートの Linux, KNOPPIX を用いたインスタントコンピューティングという技術がある．インスタントコンピューティングとは，既存の Windows 機や Macintosh 機などハードウェアや OS が混在している環境において KNOPPIX を用いてグリッドを構築するもので，一時的に分散処理環境とするものである[２]．利用用途としては建築の構造計算やバイオインフォマティクス解析[３]などの例があり，現在注目を集めている．

KNOPPIX を用いたインスタントコンピューティングは各コンピュータのハードウェア･OS･所有者の違いなど情報環境の複雑さを隠匿し，自動起動･自動設定にも優れているため，一般的な家庭環境と非常に親和性が高い．よって本研究では，KNOPPIX 上にアプリケーションを作成し，一時的な分散処理を行う．

### ２．３ 動画データ圧縮処理

表１　主な動画圧縮方式とデータサイズ

| 圧縮方式 | 標準的な圧縮率 | １時間でのデータサイズ | 画質 |
|---|---|---|---|
| なし | 約 250Mbps | 約 100GB | 最高 |
| MPEG1 | 1.5Mbps | 約 700MB | 低 |
| MPEG2 | 5Mbps | 約 2.2GB | 高 |
| MPEG4 | 3Mbps | 約 1.3GB | 高 |

表１は主な動画圧縮方式と標準的なビットレート（圧縮率）で録画した際の１時間あたりのデータサイズを表したものである．無圧縮では１時間で約 100GB と非現実的なサイズになってしまうため，表１のような方式で圧縮して利用する[４]．現在では，地上波デジタルが MPEG2 規格を利用して圧縮された状態で放送されているが，保存やデータの持ち運び（モバイル）用にさらにデータを圧縮するというニーズがある．しかし，圧縮処理には実記録時間の数倍もかかってしまう場合があり問題となっている．本研究では，複数台のコンピュータで分散して処理することにより圧縮処理を高速化する．

## ３　システムの概要

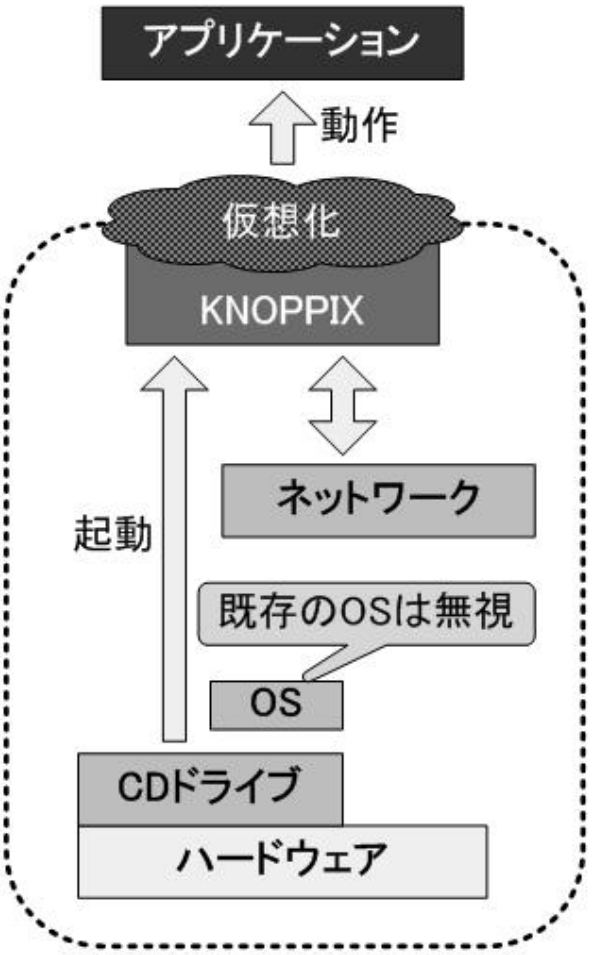


図２　システム全体の構造

図２はシステム全体の構造を表したものである．CD ドライブから KNOPPIX が起動し，既存の HDD に入っている OS については無視する．KNOPPIX の優れたハードウェア・ネットワーク自動認識・自動設定機能によって，KNOPPIX はアプリケーションに対し，ハードウェア, OS, ネットワークについて隠ぺいすることができる．実質的に KNOPPIX によってハードウェア，OS，ネットワークなどの環境について仮想化した状態となり，ユーザはサーバで圧縮処理を行うのみでよく，実際の家庭環境で想定されるノード側コンピュータの所有者やハードウェア，OS などがそれぞれ異なる環境でもその差異を意識することなく，ただの計算資源（リソース）として利用することができる．また，既存の HDD に影響を与えないため，一時的な分散処理環境の構築・利用にとどめることができ，この点もオーナーの違う端末の多い家庭環境での利

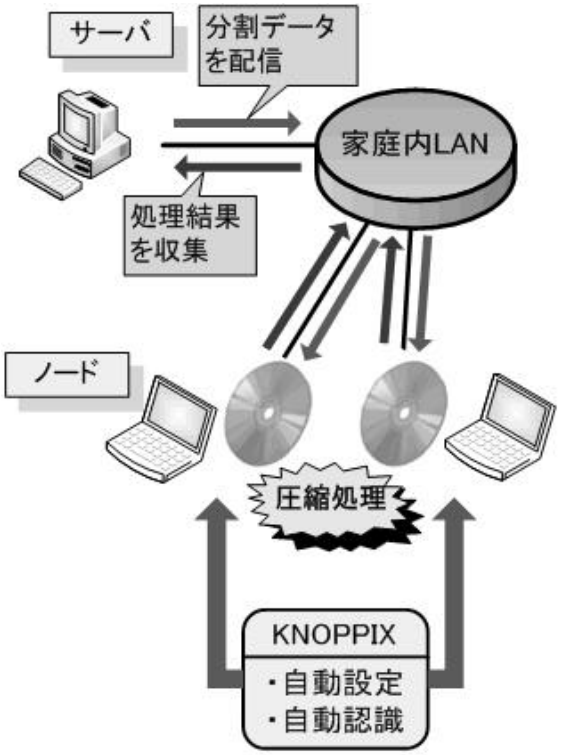


図３　家庭でのシステム利用モデル図

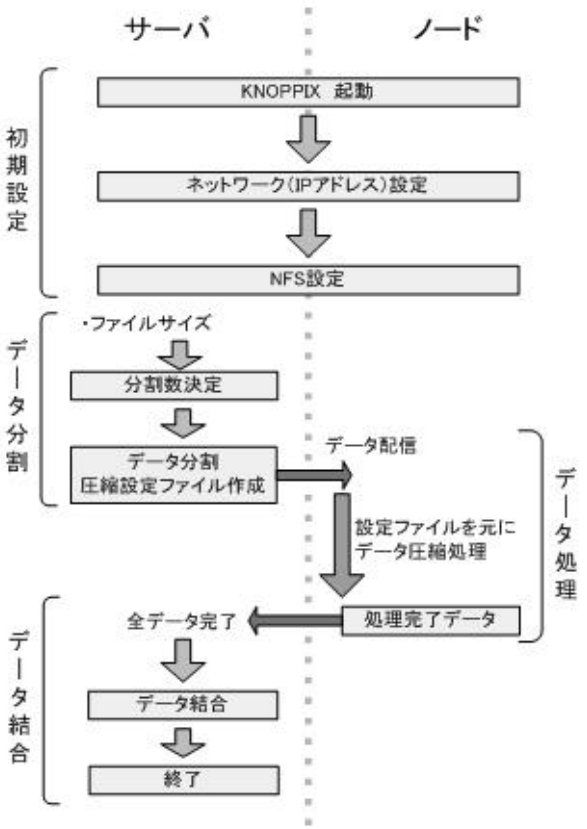


図４　ソフトウェアのデータ処理の流れ

便性の向上に有効である．

家庭において，実際にシステムを利用する際の処理の流れは図３のようになる．まず，圧縮処理を行うノード側のコンピュータにKNOPPIXを入れて起動する．この際，KNOPPIX がハードウェア・ネットワークについて自動認識・自動設定を行い，ハードウェア・ネットワークをすぐに使える状態にする．その後，サーバでユーザの操作を受け元データ分割後，分割データをノード側コンピュータに配信，ノードでは受け取った分割データについて設定に従い実際に圧縮処理を実行し，サーバへ処理結果のデータを返す．サーバでは，収集した処理結果データを結合し，ユーザへ最終結果ファイルとして出力する．

## 4　ソフトウェアの構成

図４はKNOPPIX 上で動作し，圧縮処理をする際のソフトウェアの具体的なデータフローである．

KNOPPIX の起動後，まず，通信をするためのネットワークの設定（IP アドレスなど）を行う．その後，UNIX のファイル共有システムであるNFSを設定する．以下，分割データ，処理結果データ，圧縮設定ファイル等のやり取りのサーバ・ノード間の通信は，基本的にNFS上のファイルにて行う．これは不必要なデータの送受信をなくして性能向上を望み，かつ処理開始，処理終了のイベント処理を簡潔・迅速に行うため，そして，動作の安定性を実現するためである．以上がソフトウェアの初期設定部にあたる．

次に，本研究で作成したソフトウェアの最も特徴的な部分であるデータの分割部である．分散処理においては，３台なら３つ，４台なら４つ，というように台数分に処理を分割して実行することが一般的である．しかし，本研究では，家庭において想定される複雑な情報環境に対応するために，あえて台数分よりも多く分割数を設定する．なお，各データは，分割数に応じて等分割される．分割数を台数分よりも多く設定することにより，複雑な環境に対応することができる理由は，以下の通りである．

図５は分散処理において一般的な分割数が台数分の場合のデータの処理の流れを示したもので，図６は分割数を十分多くした場合のデータの処理の流れである．台数分の分割数とした場合では，マシンα，β，γの内，α，βに比べて高性能なマシンγでは，他より速く処理が終了し，後に処理待ちの無駄な時間が生じてしまっている．せっかくのマシンγの高い処理能力を活かすことができていない．これに対し，分割数が十分多い場合では，処理が終了しても次々に処理が割り振られるため，各ノードの性能にあった処理量となり，複雑な環境に対応し，全体として効率化を図ることができる．

また，分割数増加の副次的効果として，データの処理

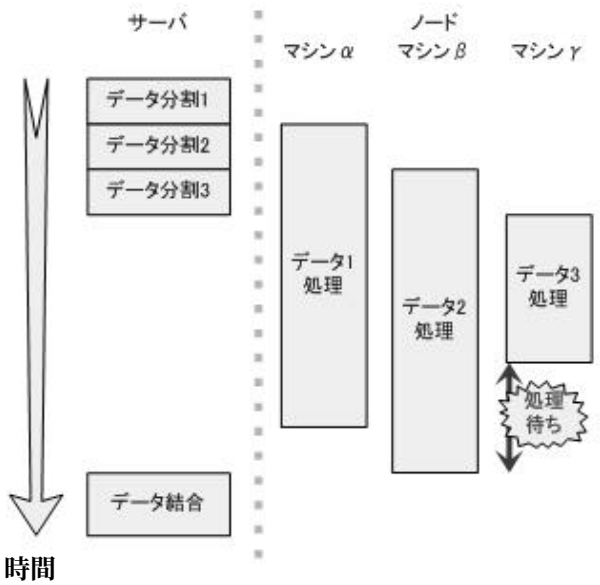


図５　分割数が台数分の場合のデータ処理

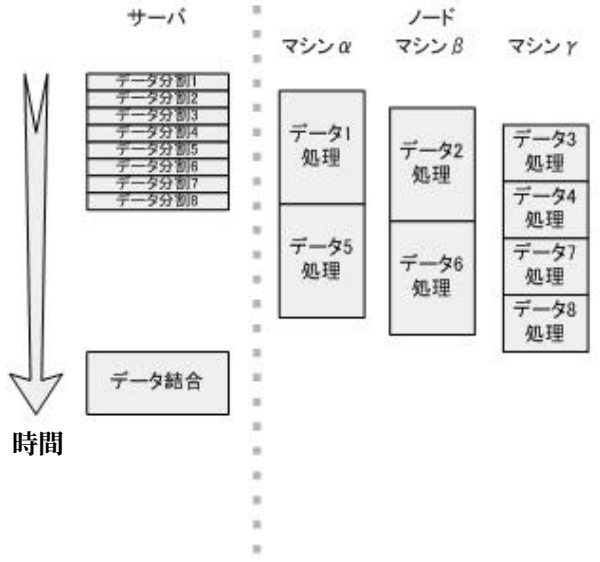


図６　分割数が十分多い場合のデータ処理

単位が短くなる．処理単位の短縮化は，ノード側のマシンの最初の処理開始までの待ち時間を少なくすることができる．よって，分割数の増加は，最初の処理までの時間短縮という点でも，全体の効率を向上させることができる．

データ分割後，分割データを受け取ったノードでは，NFS によって共有された設定ファイルを元に，共有ディレクトリ上の分割データについてデータ圧縮処理を実行し，処理完了データを共有ディレクトリに書き込む．以上の処理がデータ処理部である．

最後に，データ結合部では，サーバで全データの処理が完了したことを受け取り，最後のデータ結合の処理を行う．実際に圧縮処理が行われた処理結果データを結合し，最終的な１つの出力ファイルとして書き出す．ユーザ側へ処理の完了を通知し，中間ファイルや設定を消去した後，プログラムが終了する．

## 5　実証実験

作成したソフトウェアを使用し，実際に動画圧縮処理を分散し，高速化の効果を見るとともに本研究の特徴である分割数増加について考察する．使用したシステムの環境は表２の通りである．圧縮処理の内容は表３の通りである（一部実験によって可変）．なお，圧縮処理対象素材としては，すべてテレビ放送のバラエティ番組をキャプチャした動画データを利用した．

表２　実験に使用したコンピュータの仕様

| | サーバ | ノードA | ノードB |
|---|---|---|---|
| CPU | Pentium4 3GHz | Celeron 2.93GHz | PentiumD 3.0GHz |
| メモリ | 2GB | 1GB | 2GB |
| ネットワーク | 1Gbps | 1Gbps / 100Mbps | 1Gbps |

表３　実験で行った圧縮処理の内容

| | |
|---|---|
| 元ファイル１ | MPEG2　5Mbps　１時間（約2.2GB） |
| 元ファイル２ | MPEG2　5Mbps　30 分（約1.1GB） |
| 元ファイル３ | MPEG2　2.5Mbps　30 分（約 570MB） |
| 結果ファイル | MPEG4　3Mbps　１時間（約1.3GB） |

### ５．１ 基本実験

まず，処理環境をサーバ１台とノードA ２台という最も基本的な構成で，家庭環境において最も一般的であるMPEG2 のファイル（元ファイル１）から同程度の品質でより圧縮効率の高いMPEG4 へ処理するという基本実験を行った．

### ５．２ 分割数決定の確認実験

次に，基本実験の結果を受けて，分割数決定の予測を確かめ，基本実験の結果と比較するため，元ファイル１からファイルの長さを1/2 としたファイル（元ファイル２），圧縮率およびファイルの長さともに1/2 とし，結果的にファイルサイズを1/4 としたファイル（元ファイル３）を用意し，それぞれ実験を行った．実験の環境は，基本実験と同じ構成（サーバ１台とノードA ２台）で行った．

### ５．３ 複雑環境下での処理効率確認実験

最後に分割数増加による本研究の目的である複雑な情報環境についての対応を確かめるため，表４に示すよう

な，ノードが3台の場合（環境1），ノード3台のうち1台のみ他よりも遅い100Mbpsで接続した場合（環境2），ノードAに対し2倍程度高速なノードBを加えた場合(環境3)，という3種類の環境で元ファイル1を用いて実験を行った．

表4　処理効率確認実験の環境

| | マシン | ネットワーク |
|---|---|---|
| 環境1 | サーバ1台，<br>ノードA 3台 | 全て1Gbps |
| 環境2 | サーバ1台，<br>ノードA 3台 | ノードAのうち<br>1台のみ100Mbps |
| 環境3 | サーバ1台，ノードA<br>2台，ノードB 1台 | 全て1Gbps |

## 6　実験結果及び検討・考察

### 6．1 基本実験結果及び考察

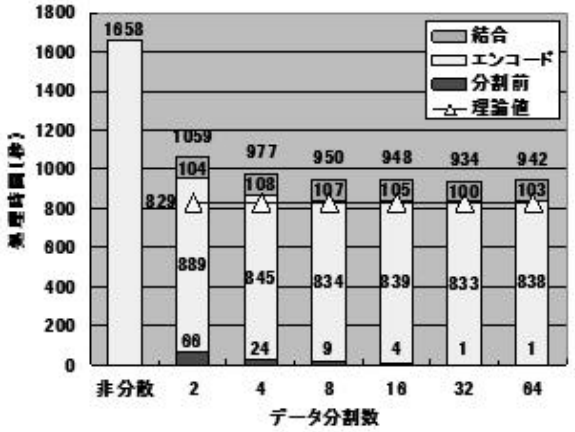


図7　基本実験結果

5．1の基本実験の結果である図7に示すように，分散処理を行わず1台で 1658 秒かかっていた処理時間について，2台で最少934秒と最大178%(1658/934×100%)の高速化が図れた．分割数8以上ではほぼ誤差範囲内の最大効率で動作しており，中でも32分割以降では，図5，図6にて示した分割数増加による副次的効果の処理単位の短縮化により，分割前の時間が1秒と最少化し，オーバーヘッドがほぼなくなっていることがわかる．よって，実験の処理性能環境，ファイルサイズにおいては処理単位を 32 分割以上にすることでさらに効率よくプログラムを動作させることができると予測される．

### 6．2 分割数決定の確認実験結果及び考察

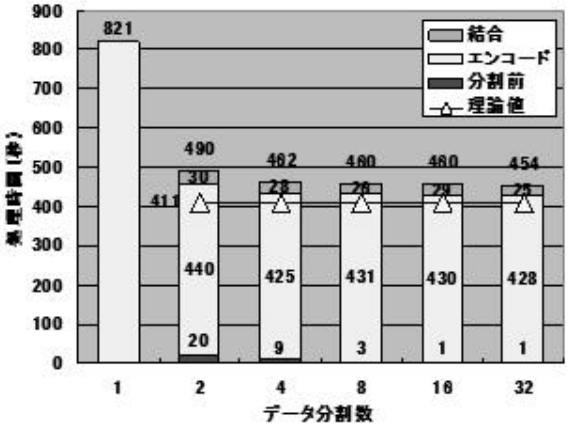


図8　分割数決定の確認実験結果（元ファイル2）

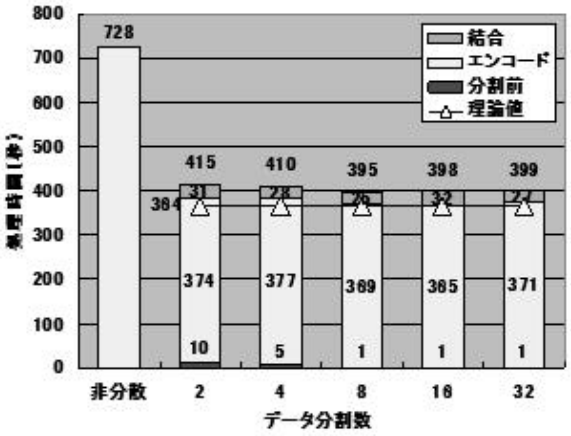


図9　分割数決定の確認実験結果（元ファイル3）

実際に分割数決定の予測を確かめるため，5．2の実験を行った結果が図8，図9である．基本実験と同じハードウェア構成で，図8に示すファイルの長さ，ファイルサイズともに1/2のデータでの実験では，基本実験の結果である32分割の1/2である16分割以降で分割前の時間が1秒と最少化し，図9に示すファイルの長さ，ビットレートともに1/2とし，ファイルサイズが1/4のデータでの実験では，32分割の1/4である分割数8以降で分割前の時間が1秒と最少になった．

以上の結果から，以下の式に示すように元データサイズをサーバ側で1秒で分割可能なサイズ（実験の処理環境ではおよそ70MB）で割った値（実験環境では32，16，8）を用いることで効率的な圧縮が可能である．

$$\text{分割数} \geqq \frac{\text{元データサイズ}}{\text{1秒で分割可能なサイズ（70MB）}} = 32,16,8 \ > \ \text{端末台数}$$

以上の式を目安に各処理環境に合わせて分割数を増加させることにより，分割数増加の副次的効果である分割前の時間の最少化を達成した上で，複雑な処理環境に対応するという双方の効果を最大化することができるということが分かった.

### ６．３ 複雑環境下の処理効率確認実験結果及び考察

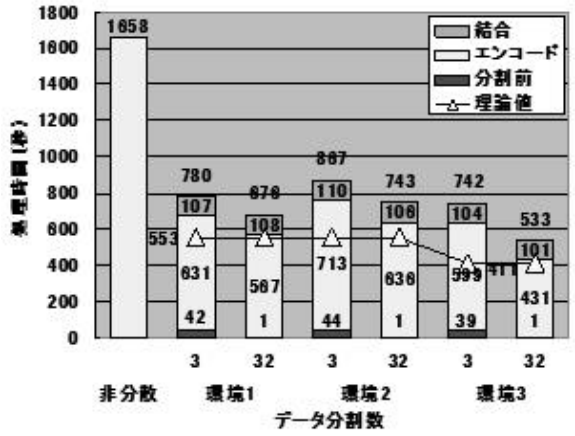


図10　複雑環境下での処理効率確認実験結果

分割数増加が家庭のような複雑な情報環境において実際に効果を表すか確認するための実験結果を図 10 に示す．分割数を台数分（３）ではなく，６．２で表した式をもとに算出し，十分に大きな値（32）とすることにより，環境１のノードが３台の場合では，780 秒かかっていた処理時間を 676 秒と 115%の高速化を達成し，環境２のネットワークに一部遅いものが含まれる場合では，867秒かかっていた処理時間を743秒と117%の高速化を達成，各ノード間に最も大きな処理性能の差異がある環境３のノードに一部性能の違うものが含まれる場合では，742秒かかっていた処理時間を533秒と139%の高速化を達成，といずれの環境においても複雑な環境に対応して高速化を図ることができた．元の分散処理を実行しない１台で処理した場合から見ると，1658 秒かかっていた処理時間をそれぞれ245%，223%，311%高速化することができた.

## ７　おわりに

### ７．１ まとめ

本論文では，ネットワーク上の複数台のコンピュータに動画データ圧縮処理を分散することにより処理時間を低減するプログラムを作成し，実証実験を行った．その結果２台による分散処理で最大178%，３台で最大245%と高速化することに成功した．さらに，一般的な家庭情報環境のようなネットワーク環境や個々に性能差がある端末でも，データの処理単位となる分割数を従来の端末台数分よりも増加させることにより，最大で 139%高速化できることが分かった.

### ７．２ 今後の課題

今後は，更なる処理の効率化を進めるとともに，６．２で示した分割数決定の式について，様々な環境で実験を繰り返して高度化を図り，家庭環境での実用を目指した GUI の実装・セキュリティ等の向上などに向け研究を進めていく．また，本研究では，予測されるオーバーヘッドの少なさから処理においてデータを等分割したが，圧縮処理そのものについてはより高速化が望める可変長分割についても，比較検討を行っていく.

## 謝辞

本研究にあたり，貴重なご意見をたくさんいただいた武蔵工業大学環境情報学部情報メディア学科奥平雅士教授，大谷紀子准教授に深く感謝致します.

## 参考文献

[１] 誰でもできるグリッドコンピューティング―さまざまな計算機環境でのグリッド構築―：
http://www.ipab.org/Presentation/sem05/05-01-02.pdf
[２] インスタントコンピューティングのツールとしての KNOPPIX：
http://www.ipab.org/Presentation/sem05/05-01-01.pdf
[３] KNOPPIX for Bio:1CD 起動 Linux によるバイオインフォマティクス解析環境の構築：
http://www.ipab.org/Presentation/sem05/05-01-03.pdf
[４] Overview of the MPEG-4 Standard：
http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm