

# BOINC クラスタを利用した KNOPPIX 構築環境の開発

阿部 大将<sup>†</sup> 北川 健司<sup>†</sup> 渡辺 義人<sup>†</sup>  
濱野 裕樹<sup>†</sup> 千葉 大作<sup>†</sup> 須崎 有康<sup>††</sup>

## 1. はじめに

CD ブート Linux “KNOPPIX” は他 OS がインストールされた PC 環境に影響を与えず、且つ自身の OS 環境に高いロバスト性を持ち、カスタマイズも他 Linux ディストリビューションと比較して容易であることから教育利用等のアプライアンス OS として注目を集めており、PC グリッド環境構築のための利用研究<sup>1)2)</sup> も行われている。現状の KNOPPIX における大きな問題としては、アプライアンス OS の構築に多くの時間、システムリソースを必要とすることが挙げられ、普及の妨げになっている。この問題を解決するため、筆者は遊休 PC のリソースを活用したグリッドフレームワーク BOINC<sup>3)</sup> ベースの KNOPPIX 作成のための PC グリッドの開発を行っている。本項では、この開発についての現状について報告する。

## 2. 概要

KNOPPIX は CD-ROM 内に isolinux, grub 等を利用したブート領域と起動ディスク (minirt.gz), システム領域 (CLOOP) を持っている。KNOPPIX の構築時間全体の中で最も大きな割合を必要とするのが CLOOP の作成であり、全体の作成時間内訳の約 80% 以上の時間がかかる。CLOOP 圧縮のほかに、CD イメージ作成のための mkisofs コマンドの実行などがあるが、CLOOP 圧縮に比べると非常に短い時間で済む。CLOOP の構築には PC の性能が向上した現在でも規模、設定によっては数十分もの時間が必要となる場合がある。

CLOOP 作成時間をさらに詳しく見ると、必要となる時間のほとんどが zlib によるデータ圧縮の時間となっている。表 1 は、KNOPPIX5.1.1CD 日本語版の CLOOP イメージを展開した時、そのまま CLOOP 圧縮した場合に必要な所要時間の割合を示したも

表 1 CLOOP 作成時間の内訳  
Table 1 inside of making loop time

処理	所要時間 (sec)	割合 (%)
全体	795.55	100
イメージ読み込み	15.10	1.90
データ圧縮 (zlib)	754.41	94.83
CLOOP 書き出し	10.69	1.34
その他	15.75	1.93

CPU: AMD Athlon(tm) 64X2 3800+ Mem: 3GB  
OS: Debian GNU/Linux 4.0r1 cloop ver: 2.05

のである。このときの CLOOP 圧縮の処理は、1.8GB のイメージをディスクから読み込み、データ圧縮処理を行った後、700MB 程度となったデータをディスクに書き出している。

データ圧縮処理性能は CPU の性能に大きく依存する。マルチコアの CPU を利用することで圧縮処理を分割することも可能だが、全力で CLOOP の構築を行わせるとすると、長い占有時間を割かなければならなくなる。この問題を解決するため、CLOOP を作成するためのグリッド環境を構築し、KNOPPIX の作成時間の短縮を検討することにした。グリッドは我々のオフィス上に用意することを想定し、フレームワークはマルチプラットフォームで実行できる BOINC を採用することとした。

## 3. 実装

CLOOP の作成はイメージファイルの分割、分割されたブロックデータの圧縮、圧縮されたブロックデータのまとめの 3 つのフェーズに分けられる。この内、BOINC に任せる部分は、データブロックを圧縮する部分とし、イメージの分割、ブロックデータのまとめは BOINC 外に別デーモンプログラムとして実装している。

KNOPPIX の CLOOP ブロックのサイズは 64 ~ 256KB が目安とされているが、このサイズをそのまま BOINC の一度の仕事量とすると、通信回数が増大してしまう。そこで、複数個の CLOOP ブロックをまとめて一度の通信で送信できるようにした。

通常 BOINC に渡されるデータは複数回計算し内容

<sup>†</sup> 株式会社アルファシステムズ  
Alpha Systems Inc

<sup>††</sup> 独立行政法人 産業技術総合研究所  
Advanced Industrial science and technology

の照合を行うが、一回の計算にデータ照合を省略した。ターゲット環境が社内 LAN のため SSL によるクライアント認証を切り、サーバにかかる負荷を軽くすることも試みたが、SSL なしではクライアントの接続ができず、計算が行えなかった。BOINC には zlib を利用して送信するデータを圧縮する機能があるが、CLOOP のブロック圧縮を複数回行うことになるため、この作業は省略した。

#### 4. 計測結果

CLOOP イメージの分解からデータの圧縮完了までについての実装を行い、実際にイメージを処理させた時のサーバ、クライアントの状態を計測した。

CLOOP ブロックのサイズを 64KB、一度に送出するブロックのサイズを 2MB になるように調整し、KNOPPIX5.1.1CD 日本語版の CLOOP イメージを展開したものを再度圧縮しなおす処理を行った。

サーバは Debain GNU/Linux 4.0r1, CPU: PentiumD 3.20GHz, Linux Kernel: 2.6.18, Memory:2GB, BOINC 6.1.0(revision:13895), クライアントは CentOS 4.4, CPU: Pentium4 2.00GHz, Linux kernel: 2.6.9-42.EL, Memory:512MB, BOINC クライアント 5.4.11 の LinuxBOX を利用した。ネットワークは 100Base HUB で構築された LAN を構築して行った。クライアントは 4 台用意し、1 台、2 台、4 台それぞれの場合での圧縮時間を計測した。得られた結果を、図 1 にグラフとして示す。

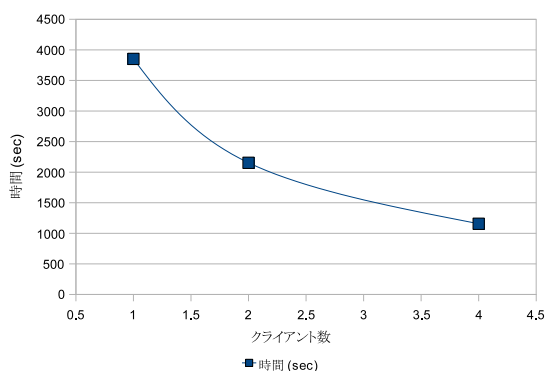


図 1 クライアント数毎の圧縮時間

Fig. 1 Compression time as client nums.

CLOOP の作成に必要なとした時間はクライアント 1 台の際には 3853 秒となる。ローカルでの CLOOP 圧縮時間と比較して 4.84 倍の時間を必要とし、BOINC のオーバーヘッドは非常に大きいと言える。クライア

ントを 2 台、4 台と増やすと、処理時間はそれに伴いリニアに減少していることから、BOINC のオーバーヘッド分の負荷をかけても、ある程度の台数のマシンが参加することができれば台数に見合った処理速度を得ることが期待できる。

#### 5. まとめ、今後の展開

グリッドフレームワーク BOINC を利用した KNOPPIX 作成のための PC グリッドの開発について、現状までに実施した内容を報告した。現状は極小の環境での動作実験のみを行っただけに過ぎないので、今後はさらにスケールアップして CLOOP の構築速度が高速化できるかどうかを実証していく必要がある。現状で、全体として不足している機能としては、CLOOP の圧縮ブロックを CLOOP イメージへ結合する機能がある。

今後は、CD-ROM へ組み込むための CLOOP 作成の仕組みを構築し、KNOPPIX の作成を行うことが出来るように開発を進めていく。特に、Windows 向けのクライアント開発を進め、実際の構成に近い Linux, Windows 混成体制での性能測定を行っていく。

開発が完了したのちは、社内での KNOPPIX 構築に利用する他、CD カスタマイズの Web サービスとしても展開を行う構想である。また、BOINC 環境を KNOPPIX に収録し、knoppix-terminalserver から BOINC プロジェクトのホストを簡単に増やしたり、同一 CD で BOINC クライアントとしての動作も出来るようにし、簡単に台数を増やせるようにすることで、BOINC ネットワークの構築を容易に行えるようにする。

#### 参考文献

- 1) 柴田良一, 林 憲一, 山方和昭, 澁谷寿夫: SunONEGridEngine と KNOPPIX を用いた教育用 PC 群による大規模グリッド構築に関する基礎的研究, 情報処理学会, VOL. IPSJ 66, 論文集 p.p. pp.5-69 : (2004)
- 2) 小西 史一: “インスタントコンピューティング技術で構築するハイスループットコンピューティング環境”, 日本化学会情報化学部会誌, Vol. 25, 34 (2007) .
- 3) D. Anderson. Boinc: A system for public-resource computing and storage. In Fifth IEEE/ACM International Workshop on Grid Computing (GRID '04), p.p. 4-10, Pittsburgh, PA, Nov 2004. IEEE Computer Society (CS) Press.