

2005 年度 上期

オープンソースソフトウェア活用基盤整備事業

「CD/DVD 起動 Linux の速度改善ドライバの開発」

起動高速化適用マニュアル

2006/2/28

株式会社アルファシステムズ

著作権について

Copyright(c) 2006 Alpha Systems Inc. All rights reserved.

このドキュメントは、株式会社アルファシステムズが著作権を保持しています。GNU Public License(GPL)にもとづき、複製と再配布および修正を許可しています。変更不許可部分は、この「著作権について」のページです。

あなたの変更を行なった場合、できるだけ我々にも連絡して下さい。たいていの変更は、適切なものであると判断されれば、その他の変更とともに原本に反映されます。

- ・ UNIX は The Open Group の登録商標です。
- ・ Linux は Linus Torvalds 氏 及び米国その他の国における登録商標あるいは商標です。
- ・ その他本書で取り上げられているシステム名、製品名、商品名は、一般に各開発メーカーの登録商標／商品名です。
- ・ 本文中に、TM マークは明記しておりません。

2006年02月28日 初版発行

著者：株式会社アルファシステムズ

発行：株式会社アルファシステムズ

Printed in Japan

本書は製品の改良その他により、予告無しに改定されることがあります。

更新履歷

年月日	更新內容
2006. 02. 28	· 初版

用語定義

用語	意味
ライブ CD	CD もしくは DVD 一枚で起動するオペレーティングシステムのこと。この文書では特に GNU/Linux システムを起動できるものを指す。
KNOPPIX	ライブ CD の一種。ドイツの Klaus Knopper 氏が Debian GNU/Linux をベースに開発している。
圧縮ループデバイス (cloop)	圧縮されたブロックデバイスイメージを扱うことができる仮想ブロックデバイス。

表記について

◆UNIXのコマンド入力の指定は,下のような枠の中に記述しています.

例)

```
# cloopprofiler
```

◆ファイル出力の例は,下のような枠の中に記述しています.

例)

```
cloop device : 0  
total blocks : 36865
```

◆Linuxの起動オプションの例は,次のような枠の中に記述しています.

例)

```
boot: kernel
```

◆ウィンドウの要素毎に,括る括弧を統一しています.

ウィンドウの要素	表記	表記例
ウィンドウのボタン	【】で括る	【Play】 , 【Next】
プルダウンメニュー, ポップアップメニューのボタン	『』で括る	『File』 , 『View』
選択ボタン	《》で括る	《select》

◆マウスの操作は,以下のように記述しています.

左ボタンを押下 → 左クリック
左ボタンを二回連続で押下 → 左ダブルクリック
右ボタンを押下 → 右クリック
右ボタンを二回連続で押下 → 右ダブルクリック

◆ウィンドウの操作で,複数の操作を行う場合,以下のような箇条書きで示しています.

操作1をします
操作2をします
操作3をします

目次

1. 概要.....	1
1.1. ライブ CD の起動高速化.....	1
1.1.1. ライブ CD の起動速度問題.....	1
1.1.2. ライブ CD 起動高速化手法.....	2
1.1.3. 起動高速化の流れ.....	4
1.2. ライブ CD 起動高速化ツールキット「Live CD Acceleration Tool kit(LCAT)」.....	5
1.3. この文書について.....	5
2. 起動高速化向け開発環境.....	6
2.1. 推奨ハードウェア.....	6
2.2. 必要となるソフトウェア.....	6
3. 開発環境の準備.....	7
3.1. deb パッケージからのインストール.....	7
3.1.1. lcat_1.0.0_i386.deb.....	7
3.1.2. lcat-cloop-module-2.6.12_1.0.0_i386.deb.....	7
3.2. ソースアーカイブからのインストール.....	8
4. Live CD Acceleration Tool kit (LCAT).....	9
4.1. LCAT の概要.....	9
4.2. 各種ユーティリティコマンド.....	10
4.2.1. ブロック参照状況プロファイラツール(cloopprofiler).....	10
4.2.2. イメージ最適化ツール(cloopoptimizer).....	21
4.2.3. ブロック読み込みプログラム(cloopreadahead, rblk2bl, appblk2bl).....	22
4.3. 最適化イメージ対応ドライバ(cloop.ko).....	24
4.4. KNOPPIX 起動スクリプト各種に対するパッチ.....	26
5. KNOPPIX の起動高速化.....	28
5.1. 最適化イメージ対応 cloop ドライバの導入.....	29
5.2. プロファイリングモードでの起動によるブロック参照状況データの取得.....	33
5.3. ブロック参照状況の解析.....	34
5.4. 既存 cloop イメージファイルのデータブロック配置最適化.....	36
5.5. 起動時の先読み指定ブロックリストの作成.....	36
5.6. 最適化 cloop イメージファイルを格納した iso イメージファイルの作成.....	37
5.7. 更なる起動の高速化.....	37
6. 謝辞.....	38

1. 概要

この文書は、2005 年度上期オープンソースソフトウェア活用基盤整備事業「CD/DVD 起動 Linux の速度改善ドライバの開発」にて開発されたライブ CD 起動高速化ツールキット「Live CD Acceleration Tool kit(LCAT)」を利用し、既存のライブ CD に対して起動の高速化を適用する際の手順を説明したマニュアルです。

ここでは、ライブ CD の起動高速化の概要、及びライブ CD の起動を高速化を行うツールキット LCAT の機能を簡単に説明します。

1.1. ライブ CD の起動高速化

ここでは、ライブ CD の起動高速化の概要を説明します。

1.1.1. ライブ CD の起動速度問題

ライブ CD は一枚の CD-ROM もしくは DVD-ROM から OS のシステム一式を立ち上げることができます。CD メディアが安価に大量生産が可能であることや、メディア自体が読み込み専用であり高いロバスト性をもっています。また、システムには比較的強力なデバイス検出機能を有しており、ほとんどの PC で利用することができます。これらの特徴を活かし、ライブ CD は教育分野での利用や、緊急時のシステムリカバリ、雑誌の付録、試供版ソフトウェアの配布など、多岐渡り利用されています。

しかしながら、ライブ CD はシークが遅いとされる光学式メディアから起動するため、通常の HD にインストールされた OS を利用する PC と比較すると、起動時間は大変長く、ユーザは PC を道具として利用できる状態になるまで待っている必要があります。それ故、ライブ CD を常用 OS として利用される頻度は低く、非常用の OS として用いられる場合が大半を占めています。

ライブ CD の起動が遅い原因は、大きく分けて二つあります。一つは、起動時の CD 読み出しに起因するもの、そしてもう一つは、周辺機器の自動検出処理などを含めた起動シーケンスに依存するものです。以下にこの二点をまとめます。

- CD の読み出し

ライブ CD では、システムの起動に必要なファイル全てを CD-ROM から読み出します。CD-ROM ドライブには、高速での読み出し対応などと謳っている製品もあります。しかし、あくまでも連続読み出し時での瞬間最高速度での表記であり、システムの起動など、様々なファイルを参照する使用状況下では最高速度の恩恵に与ることはほぼありません。特に、ファイルのランダムアクセスで生じるピックアップのシークは HD などのストレージと比較しても非常に遅く、読み出したいデータへの平均アクセス時間を長くしている要因の一つとなっています。

- 起動シーケンス

ライブ CD では Linux で使用できる周辺機器が、自動検出・自動設定によりユーザの手を煩わせることなく使用可能な状態になります。この自動検出・設定の処理はシステムの起動シーケンス中で行われますが、現在の実装では Linux 対応機器を一つずつ順番に処理しているので効率的ではありません。また、起動シーケンス中では、データが必要となった時点において CD-ROM メディアから読み出す処理を行います。そのため、プロセスが発行する I/O 要求が輻輳した場合、ピックアップのシークが増え、結局 I/O 待ちの状態が長く続き、起動シーケンスの処理に時間を要する事となります。

1.1.2. ライブ CD 起動高速化手法

ここでは,前節で挙げたライブ CD の起動時間に関する要因である二点について,それぞれの解決策について説明します.

- ファイルシステムイメージの最適化

ライブ CD の CD 読み出し問題の解決として,ライブ CD のファイルシステムイメージを最適化します.ライブ CD のファイルシステムイメージのデータブロックの並びを,できるだけランダムアクセスが発生しないように配置します.この読み込む順番に連続して配置することで,CD-ROMドライブのピックアップのシーク時間を減らします.これにより,ファイルシステムイメージの最適化が行われた CD-ROM からは一気に必要とするデータを転送することができるようになります.

図 1 は,ファイルシステムイメージの最適化の概要を示したものです.「ブロック参照プロファイラ」でライブ CD のファイルシステムイメージが起動時にどのように読み込まれているかをプロファイリングします.得られたプロファイリングのデータを元に「イメージ最適化ツール」で cloop イメージファイル中のデータブロックを読み出した順番に並び替えます.並び替えされた最適化済みファイルシステムイメージはファイルフォーマットが一部異なりますので,「最適化イメージ対応ドライバ」を用いて読み出します.

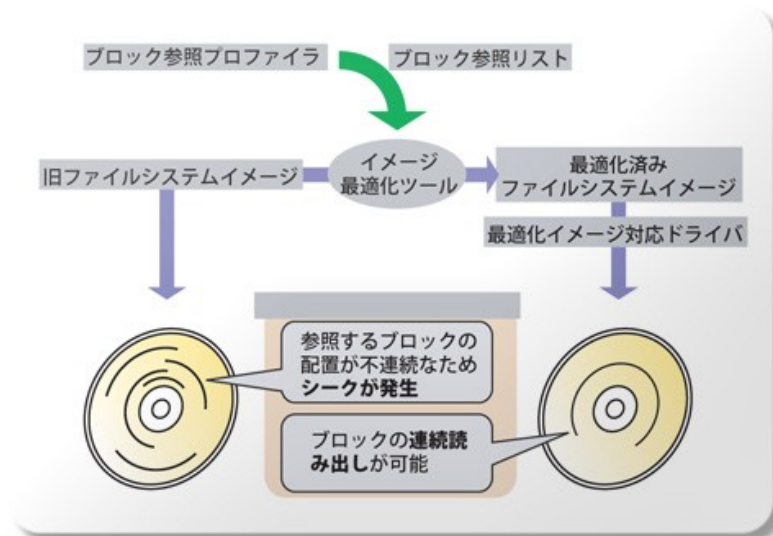


図 1:ファイルシステムイメージ最適化手法の概要

・ 起動シーケンスの最適化

起動シーケンスでの逐次処理問題の解決として、周辺機器の検出を並列化します。これによって、システムは早く起動することになります。また、起動シーケンス中で必要となるデータを予め先読みしてキャッシュに貯めておくことで、各プロセス起動時における CD-ROM ドライブへの I/O 要求の競合を低減できます。このデータ先読みと同時に周辺機器の検出を並行した起動シーケンスを実行し、起動時におけるリソース利用の効率化を図ります。

図 2 は、起動シーケンス最適化手法の概要を示したものです。起動と同時に、CD-ROM 内のデータブロックの先読みを開始しつつ、並列して起動シーケンスの処理を行い周辺機器の検出を進めます。また、アプリケーションの起動に必要なデータも先読みしておくことで、アプリケーションの起動時間も短縮できます。また、これらの着想を元に、起動シーケンスを見直し全体最適化を行います。

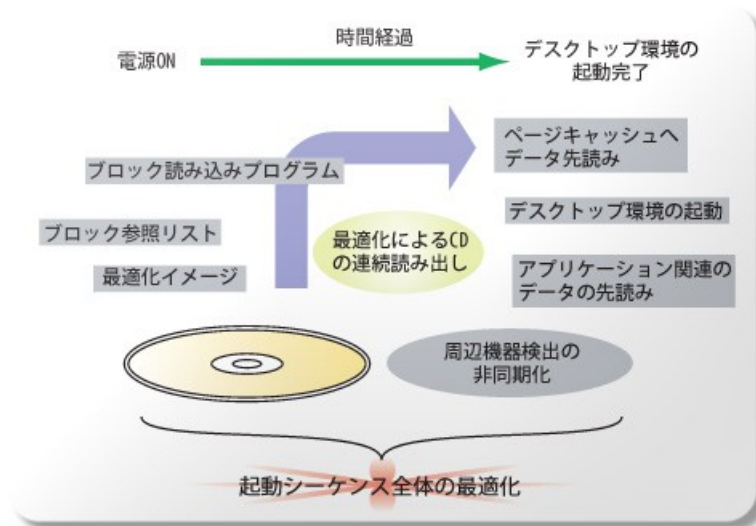


図 2: 起動シーケンス最適化手法の概要

1.1.3. 起動高速化の流れ

ライブ CD の起動高速化を行う流れは、以下のようになります。

- I. 最適化イメージ対応ドライバを導入した ISO イメージの作成
- II. ブロック参照状況の取得, 解析
- III. ブロック配置最適化
- IV. 起動高速化 ISO イメージの作成

図 3 は, LCAT を利用して KNOPPIX に起動高速化を施す場合の具体的な流れを示したものです. 図中, 着色されている部分が, 起動高速化によって変更する部分となります。

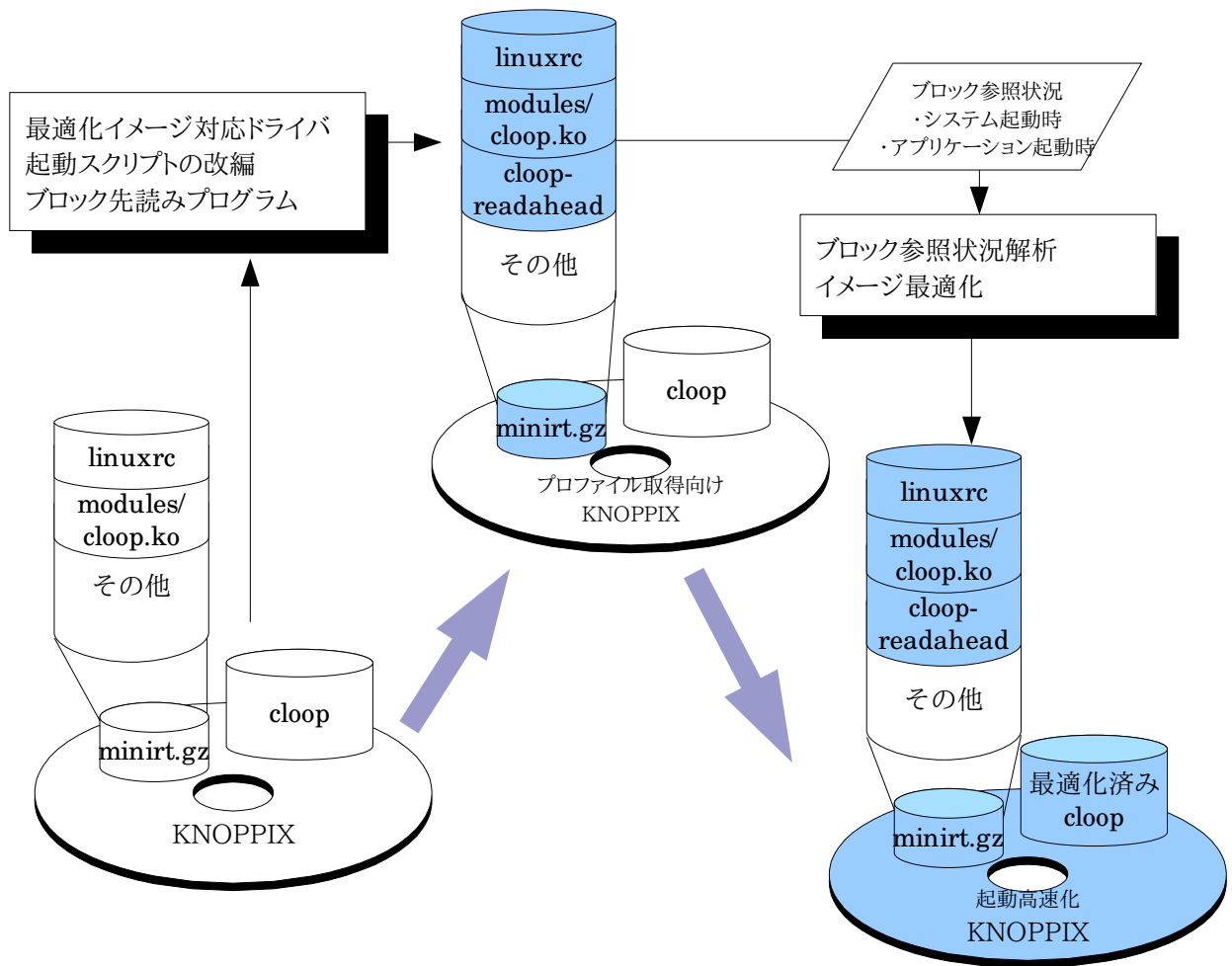


図 3:ライブ CD 起動高速化の流れ

1.2. ライブ CD 起動高速化ツールキット「Live CD Acceleration Tool kit(LCAT)」

LCAT は、これまでに述べたライブ CD の起動高速化手法を適用するために実装されたツールキットです。LCAT は KNOPPIX 等のライブ CD が採用している圧縮ループバックブロックデバイス cloop のイメージファイルのブロック配置を最適化することができます。また、LCAT のパッケージ内には KNOPPIX の起動シーケンスを最適化した起動スクリプトが同梱されています。LCAT 内のツールを cloop イメージファイルに適用し、起動用の仮ルートファイルシステムである initrd 内の起動スクリプトを差し替えることによって、ライブ CD の起動高速化を実現できます。

1.3. この文書について

この文書は、2005 年度上期 オープンソースソフトウェア活用基盤整備事業「CD/DVD 起動 Linux の速度改善ドライバの開発」の成果であるライブ CD 起動高速化ツールキット LCAT を用いて、ライブ CD の起動高速化を行う方法を解説したものです。

この文書は、ライブ CD 開発者が KNOPPIX 派生物に起動高速化を適用する際に利用することを想定したものです。ライブ CD 開発者はこの文書と LCAT のパッケージ内に収録された各 README ファイルを参考にすることで KNOPPIX 派生物の起動高速化を行うことができます。また、KNOPPIX 以外のライブ CD を起動高速化の対象とする場合でも、この手法を適用することができます。

2. 起動高速化向け開発環境

ここでは,LCATを用いて CD 版 KNOPPIX の起動高速化を行う際に推奨する開発環境のハードウェア及び,必要となるソフトウェアについて説明します.

2.1. 推奨ハードウェア

LCAT を用いて円滑に開発を行う為の推奨環境を表 1 に示します.

表 1:推奨動作環境

CPU	PC/AT 互換機(i386), 1GHz 以上のクロック周波数
メモリ	512MB 以上
記憶領域	10GB 以上
CD ドライブ	24 倍速以上の CD-R ドライブ

ハードウェアの性能が充分でない場合,プログラムの挙動に若干の遅延が見られるようになります.

2.2. 必要となるソフトウェア

LCAT に含まれるプログラムを全て動作させるのに必要となるソフトウェアを表 2 に示します.

表 2:LCAT に必要なソフトウェア一覧

OS	Linux kernel 2.6.12 以降(2.6 以降必須)
必要ライブラリ	libc6 及び開発パッケージ libc6-dev(2.3.2 以降) gtk2.0 及び開発パッケージ libgtk2.0-dev(2.4 以降) glib2.0 及び開発パッケージ libglib2.0-dev(2.0 以降)

LCAT は,以下の環境で動作確認をしています.

- ・ KNOPPIX4.0.2DVD 版(knoppix4.0.2DVD_20050923-20051006+IPAFont.iso)
- ・ KNOPPIX4.0.2DVD 版 HD インストール
- ・ Debian GNU/Linux Sarge 3.1
- ・ Fedora Core 4(kernel module 除く)

3. 開発環境の準備

ここでは,LCATのインストールについて説明します.インストールには,debパッケージを導入する方法,そしてソースからコンパイルする方法の二通りあります.

3.1. deb パッケージからのインストール

LCATのdebパッケージは,次の二つのパッケージから成り立っています.

- ・ lcat_1.0.0_i386.deb
- ・ lcat-cloop-module-2.6.12_1.0.0_i386.deb

lcat_1.0.0_i386.debには,ユーティリティプログラムが収録されています.そしてlcat-cloop-module-2.6.12_1.0.0_i386.debには,最適化イメージ対応cloopドライバ(カーネルモジュール)が収録されています.このカーネルモジュールは,gcc3.3で,linux kernel 2.6.12用に構築しています.このカーネル以外で最適化イメージ対応cloopドライバを利用する場合は,適合する環境下でソースから構築する必要があります.以下にそれぞれのパッケージのインストール方法を説明します.

3.1.1. lcat_1.0.0_i386.deb

LCATのユーティリティプログラムのパッケージです.このパッケージには最適化イメージ対応cloopドライバのカーネルモジュールは含まれていません.インストールには,root権限で以下のコマンドを実行します.

```
# dpkg -i lcat-1.0.0_i386.deb
```

3.1.2. lcat-cloop-module-2.6.12_1.0.0_i386.deb

最適化イメージ対応cloopドライバのカーネルモジュールのパッケージです.このパッケージ導入前,パッケージcloop-moduleがインストールされている場合にはパッケージの衝突が発生します.この場合,パッケージcloop-moduleをアンインストールしてください.インストールには,root権限で以下のコマンドを実行します.

```
# dpkg -i lcat-cloop-module-2.6.12_1.0.0_i386.deb
```

debパッケージからインストールを行った場合,LCATのユーティリティは“/usr/local/bin/”に,最適化イメージ対応cloopドライバは,“/lib/modules/2.6.12/kernel/drivers/block/”に配置されます.

3.2. ソースアーカイブからのインストール

作業用ディレクトリ配下にてソースアーカイブ `lcat_1.0.tar.gz` を展開します。作成されたディレクトリ `./lcat_1.0` に移動し、コンパイルを行います。コンパイルには、以下のコマンドを実行します。

```
$ tar xvfz ./lcat_1.0.tar.gz
$ cd ./lcat_1.0
$ ./configure
$ make
```

`configure` コマンドは、引数 `--prefix=<install path>` をとります。この引数を指定することで任意のディレクトリに LCAT のユーティリティをインストールできます。デフォルトのインストール先は、`"/usr/local/bin/"` です。

インストールには、`root` 権限で以下のコマンドを実行します。

```
# make install
```

“`make install`”を実行することで LCAT のユーティリティ及び、カーネルモジュールがインストールされます。ユーティリティのみインストールする場合は、カレントディレクトリを `“lcat_1.0/util”` に移した後、

“`make install`”を実行してください。最適化イメージ対応 `cloop` ドライバのカーネルモジュールは、`“lcat_1.0/cloop-2.02-opt/cloop.ko”` として作成されます。

4. Live CD Acceleration Tool kit (LCAT)

ここでは、ライブ CD 起動高速化ツールキット LCAT の概要、最適化イメージ対応 cloop ドライバ、そして各種ユーティリティについて説明します。

4.1. LCAT の概要

LCAT を構成する各種ユーティリティ、ドライバ、各種スクリプトとその機能を以下に示します。

- ブロック参照状況プロファイラツール(cloopprofiler)
プロファイリングモードによって得られたブロック参照状況データファイルを元に、ブロックの参照状況をアニメーションで可視化するツールです。
- イメージ最適化ツール(cloopoptimizer)
ブロック参照状況データファイルを元に、cloop イメージ中のデータブロックを再配置するツールです。
- ブロック読み込みプログラム(cloopreadahead, rblk2bl, appblk2bl)
cloopreadahead は、読み込みブロック指定リストを元に、最適化 cloop イメージファイル中のデータブロックを読み込むプログラムです。また、rblk2bl, appblk2bl は、ブロック参照状況データファイルから読み込みブロック指定リストを作成するプログラムです。
- 最適化イメージ対応 cloop ドライバ(cloop.ko)
イメージ最適化ツールによってデータブロックを再配置した cloop イメージファイルは、既存の cloop イメージファイルと一部フォーマットが異なります。このフォーマットに対応した、圧縮ループデバイスドライバです。また、プロファイリングモードである、データブロックの参照状況を保持する機能もあります。
- KNOPPIX 起動スクリプト各種に対するパッチ
起動高速化に伴い、起動シーケンスの処理を整理しました。これらは、システム起動時に参照される仮ルートファイルシステム initrd に格納することでシステム起動時に既存のスクリプトを置き換え、最適化された起動シーケンスを適用できます。また、周辺機器検出の非同期化を企図し、調整したデバイス認識プログラム hwsetup, kudzu も LCAT に同梱されています。

4.2. 各種ユーティリティコマンド

ここでは,LCAT を構成する各種ユーティリティについて説明します.

4.2.1. ブロック参照状況プロファイラツール(cloopprofiler)

概要

cloopprofiler は,cloop イメージファイル中のブロック参照状況を可視化するツールです.cloopprofiler はシステム起動時やアプリケーション起動時のブロック参照状況を,色付きのブロックで可視化し,ブロック読み出しの状況をアニメーション表示します.また,後述する clooptimizer や rblk2bl, appblk2bl など,LCAT の他コマンドラインユーティリティと同等の機能を持ち,ライブ CD の起動高速化ウィザードとしても利用できます.

cloopprofiler を起動すると,図 4 のウィンドウを表示します.

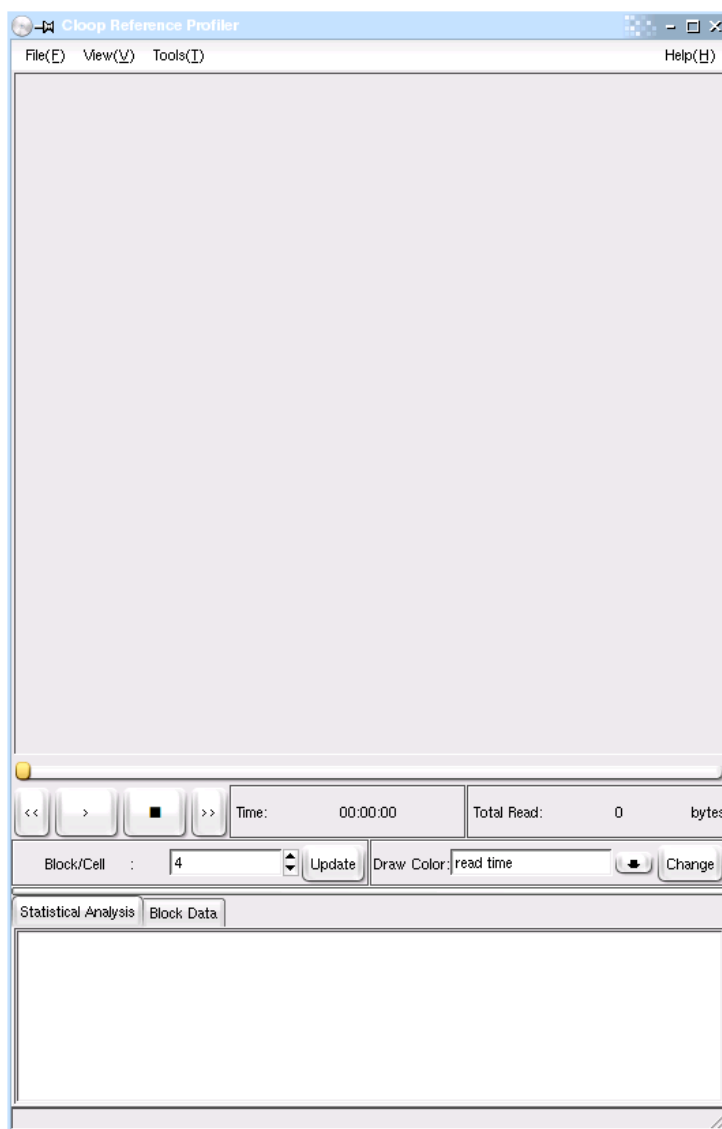


図 4: cloopprofiler メインウィンドウ

画面上半分が cloop のブロック配置を表示するブロック領域です。ブロック参照状況を再生すると、このブロック描画領域がアニメーションしながら変化します。

cloopprofiler は大きく分けて、ブロック参照状況可視化機能、ライブ CD 起動高速化機能の二つの機能を持ちます。

- ブロック参照状況可視化機能
 - ◆ ブロック参照状況のブロック ID によるアニメーション表示。
 - ◆ ブロック参照状況のブロックオフセット値によるアニメーション表示。
 - ◆ cloop イメージファイルのヘッダ情報表示。
 - ◆ アニメーションの、再生、一時停止、早送り、巻戻しなどの制御。
 - ◆ ブロック読み出しに要した時間でのブロック表示の色分け。
 - ◆ ブロック読み出し回数でのブロック表示の色分け。
 - ◆ 描画ブロックのデータ量/ブロック数の解像度変更。
 - ◆ ブロック描画領域でのポップアップウィンドウによるデータブロックの詳細表示。
 - ◆ ブロック参照状況のリスト表示。
 - ◆ 上記リスト表示の要素選択による、ブロック描画領域の強調表示。
 - ◆ スライダーによるブロック参照状況アニメーションの再生時間変更。
- ライブ CD 起動高速化機能
 - ◆ cloop イメージ最適化ウィザードによる、最適化 cloop イメージファイルの作成。
 - ◆ 先読み指定ブロックリスト作成ウィザードによる、先読み指定ブロックリストの作成。
 - ◆ プロファイリングモードで起動する最適化イメージ対応 cloop ドライバのブロック参照状況データの初期化。

使用方法

clooptimizer は,以下のコマンドで起動します.

```
$ clooprofiler
```

clooprofiler の主な使用方法を説明します.

- ブロック参照状況のブロック ID によるアニメーション表示

メニューバー『File』→『Open read_blocks』を選択し(図 5),ファイルダイアログにてブロック参照状況データファイルを開くと,ブロック参照状況をブロック ID での描画モードで可視化できる状態になります.

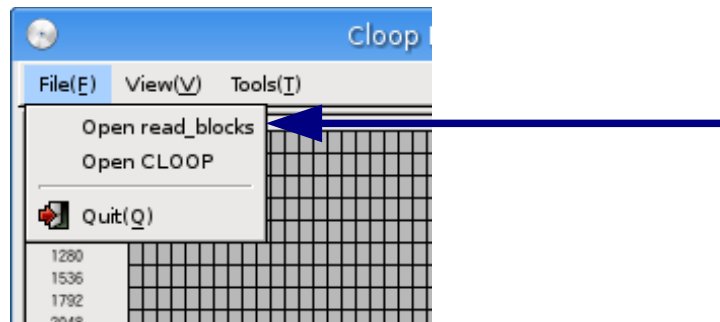


図 5: 『Open read_blocks』の選択

アニメーションの制御はウインドウ中段のボタンやスライダーで行います(図 6).

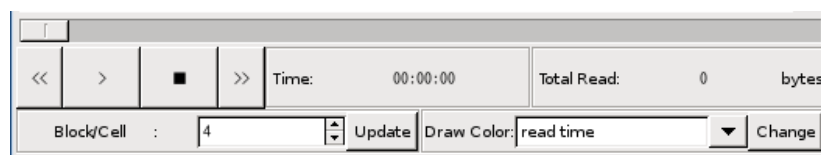


図 6: 『clooprofiler』のアニメーション制御ボタン

- ブロック参照状況のオフセット位置によるアニメーション表示

ブロック参照状況データファイルを開いた後,メニューバー『File』→『Open CLOOP』を選択し(図7),ファイルダイアログにて cloop イメージファイルを開くと,ブロック参照状況をオフセット位置での描画モードで可視化できる状態になります.

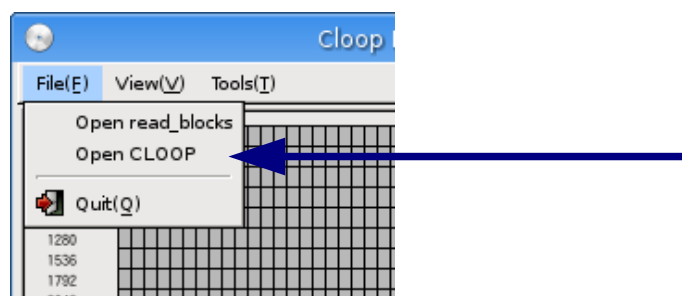


図 7: 『Open CLOOP』の選択

ブロック参照状況のブロック ID / オフセット位置での描画モード切り替えは、メニューバー『View』→『draw resource』の『read blocks log』, 『loop header』で行います(図 8).

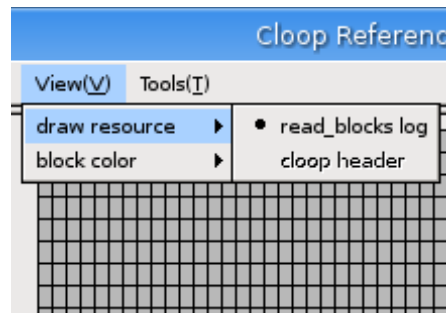


図 8: 『draw resource』の選択

- loop イメージの最適化

loop のイメージ最適化は、メニューバー『Tools』→『Accelerate Wizard』を選択し(図 9), Cloop Optimize Wizard ダイアログ(図 10)を開きます。

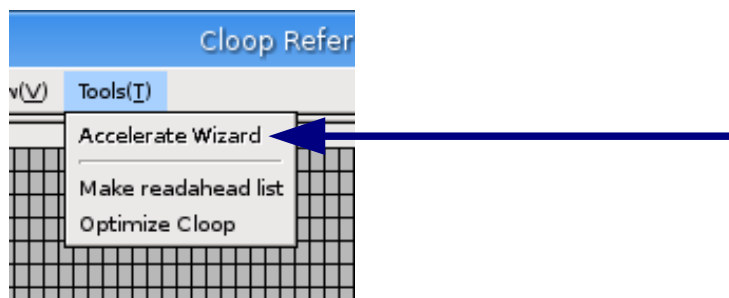


図 9: 『Accelerate Wizard』の選択

loop イメージ最適化ウィザードダイアログでは、“Block List”に先読み指定ブロックリストファイル，“Optimized Cloop”に最適化 loop イメージファイルの保存先を指定し，【Next】ボタンを押下することで最適化を実行します。

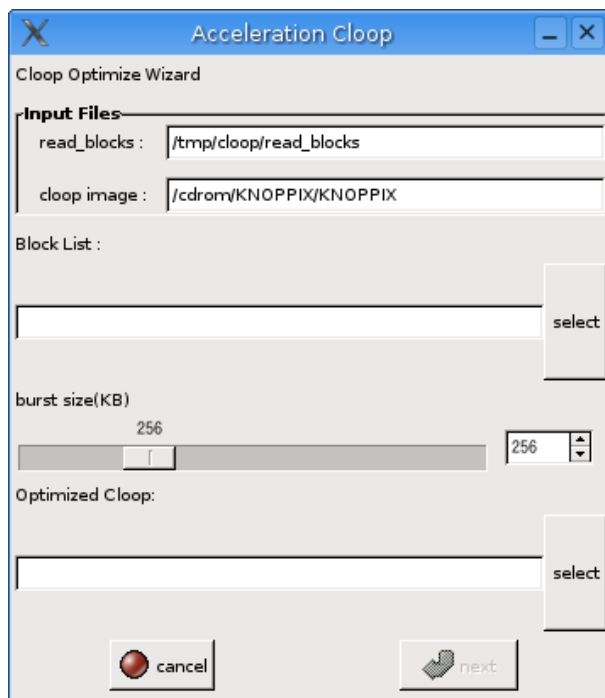


図 10: 『Cloop Optimize Wizard』ダイアログ

メニューバー一覧

ここでは,cloopprofiler のメニューバーについて説明します。

- 『File』メニュー

『File』メニュー(図 11)では,ブロック参照状況データファイルや cloop イメージファイルを指定するダイアログを開くことができます。

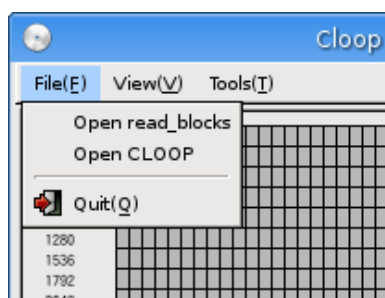


図 11: 『File』メニュー

- ◆ 『Open read_blocks』

ブロック参照状況データファイルを指定するダイアログを開きます。『Open CLOOP』メニュー選択の前に,まずこのメニューで可視化の対象となる cloop イメージファイルに対応するブロック参照状況データファイルを指定する必要があります。

このメニューで指定できるブロック参照状況データファイルは,LCAT1.0 以降の最適化イメージ対

応 cloop ドライバから出力されたブロック参照状況データファイルだけです。

- ◆ 『Open CLOOP』

cloop イメージファイルを指定するダイアログを開きます。cloop イメージファイルを指定することによって、メニューバー『View』→『draw resource』からブロック表示モードの選択ができるようになります。また同様に、メニューバー『Tools』→『Accelerate Wizard』の選択ができるようになり、cloop イメージ最適化ウィザードダイアログを実行できるようになります。

可視化する cloop イメージファイルを変更する場合には、その対象となる cloop イメージファイルのブロック参照状況データファイルを再指定する必要があります。

- ◆ 『Quit』

このアプリケーションを終了します。

- 『View』メニュー

『View』メニューは、ブロック参照状況の表示の調整を行います。

- ◆ 『draw resource』コンテナ

ブロック参照状況描画モードの切り替えを行います(図 8)。

- 『read_blocks log』

ブロック参照状況の描画を、ブロック参照状況データから得られた、ブロック ID を用いて描画するモードです。

- 『cloop header』

ブロック参照状況の描画を、cloop イメージファイルのヘッダ情報を元に、ブロックのオフセット位置を用いて描画するモードです。

- ◆ 『block color』コンテナ

描画ブロックの色の付け方を変更します。これにより、頻繁に参照されたデータブロックや、読み出しに時間を要したデータブロックが一目で判別できます(図 13)。

- 『read time』

読み込みに要した時間で、ブロックの描画領域の色を別けて描画するモードです。読み込みに要した時間が長い程、色が深くなります。

- 『read count』

読み込み回数で、ブロックの描画領域の色を別けて描画するモードです。読み込みの回数が多い程、色が深くなります。

- 『time/count』

描画ブロックの領域を対角線で半分に別け、上の三角部分を『read time』モード、下の三角部分を『read count』モードで描画します。

- 『count/time』

描画ブロックの描画領域を対角線で半分に分け、上の三角部分を『read count』モード、下の三角部分を『read time』モードで描画します。

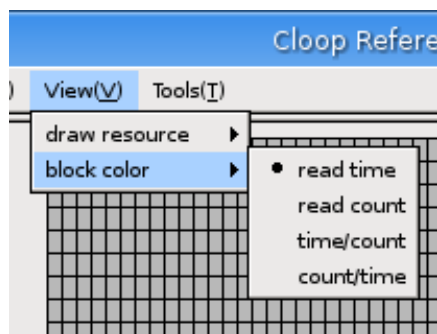


図 12:『block color』コンテナ

- 『Tools』メニュー

『Tools』メニューでは、ライブ CD の起動高速化で用いる、cloop イメージ最適化ウィザードや、先読み指定ブロックリスト作成ウィザードの起動を行います(図 14)。これらは、cloop イメージファイルが指定されている場合のみ選択できます。

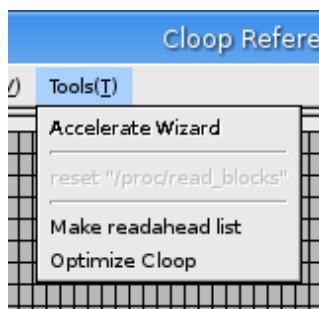


図 13:『Tools』メニュー

- ◆ 『Accelerate Wizard』

開いているブロック参照状況データファイルを元に、開いている cloop イメージファイルのブロック配置最適化及び、先読み指定ブロックリストを作成するウィザードを起動します。

- ◆ 『Make readahead list』

先読み指定ブロックリスト作成ウィザードを起動します。

- ◆ 『Optimize Cloop』

cloop イメージ最適化ウィザードを起動します。

- 『Help』メニュー(図 15)

- ◆ 『Info』

このプログラムの情報を表示します。

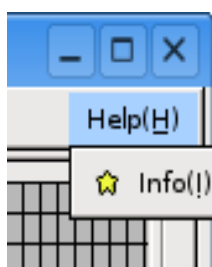


図 14:『Help』メニュー

インターフェイス一覧

ここでは、各インターフェースの機能について説明します。主なボタンの配置は図 6 を参照してください。

- 【>】([||]) ボタン
ブロック参照状況データファイルが開かれた状態であるとき、ブロック参照状況をアニメーションで再生します。アニメーションの再生中は、一時停止ボタンとなります。
- 【■】 ボタン
アニメーションの再生を停止し、タイムカウンターを初期値に戻します。
- 【>>】 ボタン
アニメーションの再生を早送りします。
- 【<<】 ボタン
アニメーションの再生を巻戻しします。
- ブロック描画領域
ブロック参照状況のアニメーションが描画される領域です。アニメーション中、現在読み込んでいるブロックは、青い枠で囲まれて表示されます。描画済みブロック(つまり、参照されたデータブロック)の領域にマウスポインタを移動し、右クリックすることで、その描画ブロックの領域に相当するデータブロックの詳細をポップアップして表示します。
- 時間進捗スライダー
ブロック描画領域の真下に表示されているスライダーは、再生時間の進み具合を示します。トグルをドラッグすることで、再生時間を変更できます。
- 【Block/Cell】 選択ボタン
ひとつの描画ブロックに含まれるデータ量を変更します。ブロック ID による表示モードではブロック数、オフセット位置による描画モードではブロックのデータサイズ(2 の選択値乗バイト)が、一つの描画ブロックに対するデータ量となります。データ量の変更後、【>】([||]) ボタン、【■】 ボタン、【change】 ボタンを押下することによってブロック描画領域に反映されます。このデータ量の変更には、アニメーションの停止を伴います。
- 『Draw Color』プルダウンリスト(図 16)

メニューバー『View』→『draw color』と同様の機能です。

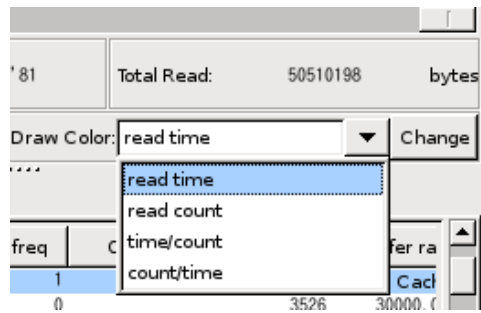


図 15: 『Draw Color』プルダウンリスト

- 『Statistical Analysis』リスト

ブロック参照状況データファイルから、統計的な分析結果を表示します。

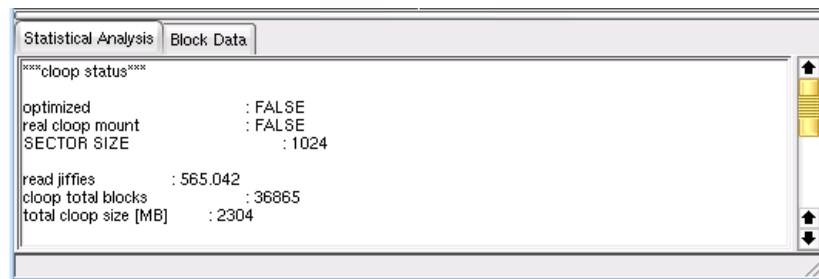


図 16: 『Statistical Analysis』リストの表示

- 『Block Data』リスト

ブロック参照状況を参照順にリスト表示します。リストをクリックすることで、該当するデータブロックに相当するブロック描画領域の描画ブロックが強調表示されます。

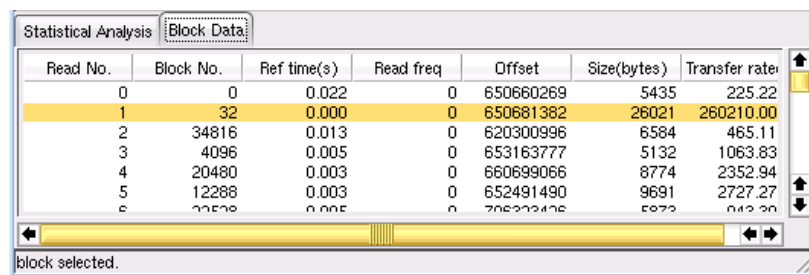


図 17: 『Block Data』リストの表示

『Tools』メニューから起動されるウィザード一覧

ここでは、『Tools』メニューから起動する各ウィザードについて説明します。

● 『Accelerate Wizard』

『Accelerate Wizard』を選択することで、Cloop Optimize Wizard ダイアログ(図 10)を表示します。この Cloop Optimized Wizard は、現在開いている、ブロック参照状況可データファイルを元に、現在開いている cloop イメージファイルの最適化を行います。最適化の処理により、最適化済み cloop イメージファイルと先読み指定ブロックリストを生成します。Cloop Optimize Wizard ダイアログのインタフェースについて以下にまとめます

◆ read_blocks

現在開いているブロック参照状況データファイルのパスを表示します。

◆ cloop image

現在開いている cloop イメージファイルのパスを表示します。

◆ Block List

先読み指定ブロックリストの出力先を指定します。

◆ burst size

データブロック先読みの際に 1 回のリクエストで読み込むデータのサイズを、キロバイト単位で指定します。

◆ Optimized Cloop

最適化した cloop イメージファイルの出力先を指定します。

● 『Make readahead list』

『Make readahead list』を選択することで、Create Block List Wizard ダイアログ(図 19)を表示します。Create Block List Wizard は、任意のブロック参照状況データファイルから、先読みブロックリストを作成します。これは、LCAT ユーティリティコマンド rblk2bl と同様の動作をします。Create Block List Wizard ダイアログのインタフェースについて、以下にまとめます。

◆ Input File

入力するブロック参照状況データファイルを指定します。

◆ Output File

出力する先読み指定ブロックリストを指定します。

◆ burst block

データブロック先読みの際に 1 回のリクエストで読み込むデータのサイズを、キロバイト単位で指定します。

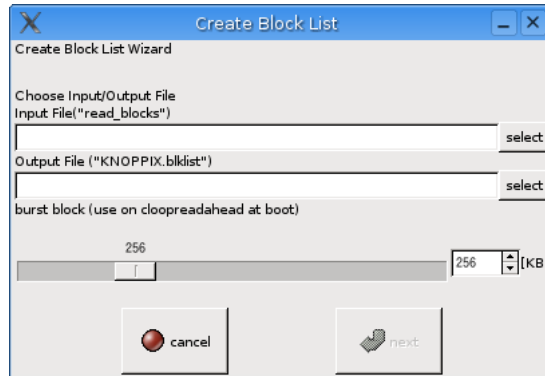


図 18: Make readahead list ダイアログ

- 『Optimize Cloop』

『Optimize Cloop』を選択することで,Optimized Cloop Image Wizard ダイアログ(図 20)を表示します.Optimized Cloop Image Wizard は,任意の cloop イメージファイルのブロック配置を,任意のブロック参照状況データファイルを元に最適化します.Optimized Cloop Image Wizard ダイアログのインタフェースについて,書きにまとめます.

- ◆ Input Cloop

最適化の対象となる cloop イメージファイルを指定します.

- ◆ Input Block List

最適化の対象となる cloop イメージファイルのブロック参照状況データファイルを指定します.

- ◆ Output Cloop

出力される最適化した cloop イメージファイルを指定します.

- ◆ butst block

データブロック先読みの際に 1 回のリクエストで読み込むデータのサイズを,キロバイト単位で指定します.

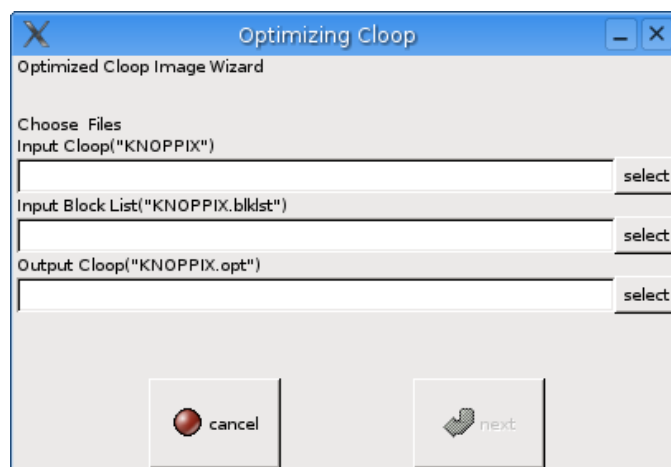


図 19: Optimized Cloop Image Wizard ダイアログ

4.2.2. イメージ最適化ツール(cloopoptimizer)

概要

cloopoptimizer は、既存フォーマットの cloop イメージファイルから最適化フォーマットの cloop イメージファイルを作成します。cloopoptimizer で作成した最適化 cloop イメージファイルは、LCAT に含まれる最適化イメージ対応 cloop ドライバからのみ、読み込み可能となります。

cloopoptimizer で扱える cloop イメージファイルは、既存フォーマットの V2.0 以降に対応しています。それ以前のバージョンの cloop イメージファイルは使用できません。

使用方法

cloopoptimizer は、以下の引数をとります。

```
$ cloopoptimizer <original cloop file> <boot block reference data> [<application reference data>]
```

最適化する cloop のパスを<original cloop file>、ブート時のブロック参照状況データファイルを<boot block reference data>、アプリケーションのブロック参照状況データファイルを<application reference data>に指定します。<application reference data>は省略可能です。

cloopoptimizer の結果は標準出力に出力され、進捗状況は標準エラー出力に出力されます。cloopoptimizer の実行例を図 19 に示します。実行例では、プロファイリングモード(5.2「プロファイリングモードでの起動によるブロック参照状況データの取得」を参照)で起動した KNOPPIX 上で、cloopoptimizer を実行しています。

```
$ cloopoptimizer /cdrom/KNOPPIX/KNOPPIX /proc/cloop/read_blocks >
./KNOPPIX.opt
block_size = 65536
num_blocks = 29859
header write.
start_offset = 477896
opt_blk_info_sum = 721645729, (704732KB)
block info table write.
block data write.
00001/29859: 0.000%, optidxno=00001, optidxsize=30220, write=30220
00002/29859: 0.003%, optidxno=00002, optidxsize=30491, write=30491
00003/29859: 0.007%, optidxno=00003, optidxsize=32705, write=32705
00004/29859: 0.010%, optidxno=00486, optidxsize=10347, write=10347
00005/29859: 0.013%, optidxno=00487, optidxsize=9412, write=9412
00006/29859: 0.017%, optidxno=00488, optidxsize=10374, write=10374
:
```

図 20: cloopoptimizer の実行例

4.2.3. ブロック読み込みプログラム(cloopreadahead, rblk2bl, appblk2bl)

- cloopreadahead

概要

cloopreadahead は、先読み指定ブロックリストを元に、最適化 cloop イメージファイル中のデータブロックを読み込むプログラムです。読み込まれたブロックのデータは、kernel の仮想メモリサブシステムが管理するページキャッシュに納められます。ライブ CD のシステム起動の際、起動用の仮ルートファイルシステム initrd 中で呼び出すことによって、起動シーケンス中で生成されたプロセスの I/O 待ち時間を短縮することができます。

このプログラムは、最適化 cloop イメージファイル専用です。

使用方法

cloopreadahead は、以下の引数をとります。

```
$ cloopreadahead <optimized cloop file> <block list file>
```

最適化 cloop イメージファイルへのパスを<optimized cloop file>、先読み指定ブロックリストを<block list file>に指定します。

cloopreadahead は読み込んだ cloop ブロックのブロック番号を標準出力に出力します。

- rblk2bl

概要

プロファイリングモードで動作している最適化イメージ対応 cloop ドライバによって得られた、ブロック参照状況データファイルを、システム起動時に cloopreadahead で参照する先読み指定ブロックリストへ変換するプログラムです。

使用方法

rblk2bl は、以下の引数をとります。

```
$ rblk2bl <block reference data> [<burst size>]
```

ブロック参照状況データファイルを<block reference data>、読み込み単位のサイズを<burst size>に指定します。この値は cloopreadahead でブロックを読み込む際の粒度になり、大きすぎても小さすぎても起動高速化の弊害となるようです。デフォルトでは 262144B(256KB)となっています。

rblk2bl は標準出力に先読み指定ブロックリストを出力します。rblk2bl が出力する先読み指定ブロックリストの例を、図 20 に示します。

```

# burst_size = 262144 (256 KB)
burst_start
0
32
34816
36352
4096
4480
20480
20992
12288
12928
22528
23296
16384
17408
24576
26112
# block = 17, size = 262346
burst_end
burst_start
26624
28160
:

```

図 21:先読み指定ブロックリストの例

● appblk2bl

概要

プロファイリングモードで動作している最適化イメージ対応 cloop ドライバによって得られた、ブロック参照状況データファイルを、アプリケーション起動前に cloopreadahead で参照する先読み指定ブロックリストへ変換するプログラムです。

使用方法

appblk2bl は、以下の引数をとります。

```

$ appblk2bl <boot block reference data> <burst size> \
  <application block reference data> ...

```

システム起動時のブロック参照状況データファイルを<boot block reference data>、読み込み単位のサイズを<burst size>、アプリケーション起動時のブロック参照状況データファイルを<application block reference data>に指定します。複数のアプリケーション起動時のブロック参照状況データファイルを指定できます。

appblk2bl は標準出力に先読み指定ブロックリストを出力します。appblk2bl の出力フォーマットは、rblk2bl と同一です。

4.3. 最適化イメージ対応ドライバ(cloop.ko)

概要

最適化 cloop イメージファイルはデータブロックの位置情報保持のため、既存 cloop イメージファイルとは一部ファイルフォーマットが異なっています。このフォーマットに対応した cloop ドライバが、最適化イメージ対応ドライバです。また、データブロックの再配置のために必要となる、ブロック参照状況データを保持する機能が追加されています。

このドライバを、ライブ CD が起動する際に用いる仮ルートファイルシステム initrd に組み込むことによって、後述するプロファイリングモードでの起動や、既存 cloop イメージファイル、最適化済み cloop イメージファイルをルートファイルシステムデバイスとして扱うことができます。

使用方法

この最適化イメージ対応ドライバはカーネルモジュールとして提供されます。以下のモジュールパラメータをとりまします。

```
# insmod cloop.ko file=<cloop image file> \  
chkblks=<max check block number>
```

モジュールのロード時にデバイスイメージとして設定するファイルのパスを<cloop image file>、ブロック参照状況データを保持する最大値を<chkblks>に指定します。この値が 0 以上の場合、最適化イメージ対応ドライバは、プロファイリングモードで動作します。ブロック参照状況のインタフェースには、procfs を利用しています。

次の実行例は、ソースからの LCAT コンパイル後、cloop イメージファイル “/cdrom/KNOPPIX/KNOPPIX” を指定し、ブロック参照状況データを 10,000 ブロック分保持する設定でロードしています。

```
# insmod ./cloop-2.0.2-opt/cloop.ko \  
file=/cdrom/KNOPPIX/KNOPPIX \      chkblks=10000
```

モジュールパラメータ“chkblks”が指定されている場合、“/proc”配下に cloop のディレクトリが作成されます。このディレクトリ配下のファイルを、以下にまとめます。

- /proc/cloop/read_blocks

このファイルを読み出すことで、ブロック参照状況を取得できます。ブロック参照状況の例を図 23 に示します。このファイルは読み出し専用です。

- /proc/cloop/reset_read_blocks

このファイルへ書き込むことで、保持しているブロック参照状況をクリアします。このファイルは、root 権限のみで書き込みできます。

- /proc/cloop/reset_profile

このファイルへ書き込むことで、保持しているブロック参照状況の統計的データをクリアします。このファイルは、root 権限のみで書き込みできます。

```

cloop device : 0 # cloop デバイス ID
total blocks : 36865 # 総 cloop ブロック数
file format : optimized # 最適化済み / 既存フォーマット
now jiffies : 35549.0288 # 取得時の uptime
cloop kernel thread mode. # thread モードで起動
-----
call cloop_request : 135551 times # cloop 読み込みリクエスト数
call load_buffer() : 485448 times # load_buffer() コール数
call read_from_file() : 42441 times # read_from_file() コール数
read_time_max : 0.0039 ms # 最大読み込み時間
read_time_min : 0.0000 ms # 最小読み込み時間
read_time_sum : 12.0818 ms # 総読み込み時間
read length sum : 820170874 bytes # 総読み込みサイズ
extract_time_max : 0.0009 ms # 最大伸張時間
extract_time_min : 0.0000 ms # 最小伸張時間
extract_time_sum : 21.0215 ms # 総伸張時間
extract length sum : 2781413376 bytes # 総伸張後サイズ
-----
no blknum size access read extract
-----
0 0 5435 25113.0688 0.0000 0.0001
1 32 26021 25137.0298 0.0001 0.0000
2 33 28640 25139.0024 0.0058 0.0002
:
# no -> 読み込み ID
# blknum -> ブロック ID
# size -> ブロックサイズ
# access -> アクセス時の jiffies
# read -> 読み込みに要した時間
# extract -> ブロックの伸張に要した時間

```

図 22: ブロック参照状況/proc/cloop/read_blocks の例

ブロック参照状況データ“/proc/cloop/read_blocks”のリセットには、以下のコマンドを root 権限で実行します。

```
# echo 1 > /proc/cloop/reset_read_blocks
```

4.4. KNOPPIX 起動スクリプト各種に対するパッチ

概要

ライブ CD の起動高速化には, cloop イメージのデータブロック配置最適化の他にもう一つ, 周辺検出などを含む起動シーケンスの最適化があります. KNOPPIX の場合, システムの起動で実行される各スクリプトファイルは linuxrc を除き, 全て cloop イメージ上のファイルシステムに格納されています. これでは, 高速化への試行の度に cloop イメージを作成する工程が発生し, 作業効率が低下します. そこで, システムの起動の初期段階に, 起動シーケンスの処理を実行する各スクリプトファイルを差し替える処理を追加しています. また, 各スクリプトから呼ばれるコマンドも一部変更しています.

変更を加えたスクリプト一覧(起動順)

ここでは, knoppix-4.02 日本語版 CD の起動高速化に伴い, 変更を加えたファイルについて説明します.

- linuxrc

これは, 起動時の仮ルートファイルシステムである initrd 中にあり, kernel が一番最初に起動するスクリプトです.

CD-ROM デバイス中に cloop イメージファイルを検出後, 先読み指定ブロックリストが存在した場合, ブロック読み込みプログラムを起動します. また, cloop のブロック参照状況データを取得するためブートオプションの文字列を解析し, cloop ドライバのモジュールパラメータを変更します. また, 起動シーケンスを最適化した各種スクリプトをルートファイルシステム構築時にインストールします.

- inittab

これは, ランレベルを管理し全てのプロセスの親プロセスとなる init の設定を行うファイルで, “/etc/inittab” に置かれます.

既存の設定ファイルでは, ランレベルが 5 となり GUI の環境である X サーバを起動する直前に, 2秒待つ処理が入っていました. 今回はその処理を取り除きました.

- knoppix-autoconfig

これは, 起動したハードウェア環境に合わせた各ソフトウェアの設定ファイルの作成や, 各サービスを行うデーモンを起動するスクリプトファイルで, “/etc/init.d/knoppix-autoconfig” に置かれます.

既存の実装では, クリティカル・パスとなる処理が終わるまで単純に sleep で待つ実装となっていました. 今回は, 積極的にその処理が終了したことを判断するように実装しました. また, サービスのプロセスを一つずつ起動するのではなく, まとめて起動するようにしました. 周辺機器検出では新しく用意した hwsetup(後述)を実行するようにしました. また, X サーバの設定ファイルを作成するコマンドも, 新しく用意した mkxf86config(後述)を実行するようにしました.

- xsession

これは、Linux の GUI 環境の根底をなす、X サーバを起動するスクリプトファイルで、“/etc/init.d”に置かれます。

これも、上記 knoppix-autoconfig と同様、クリティカル・パスとなる処理の部分にあった sleep での待ち s 処理をやめ、積極的にプロセスの起動確認を行うようにしました。

- 45xsession

これは、X サーバの起動完了後、デスクトップ環境を起動するためのスクリプトで、“/etc/X11/Xsession.d/45xsession”に置かれます。

プロセス一つずつを逐次実行していたものを、影響の無い範囲でまとめて起動するようにしました。また、デスクトップ起動完了後に実行される、アプリケーション起動時に参照するデータブロックを先読みするためのスクリプト生成処理を追加しました。

置き換えたコマンド一覧

ここでは、knoppix-4.02 日本語版 CD の起動高速化に伴い、変更を加えたスクリプトファイルから呼び出される、新たに用意したコマンドについて説明します。

- hwsetup

既存の knoppix-autoconfig スクリプトから周辺機器検出の一環として呼び出される hwsetup は、実行時シリアルバスの検出に時間を要することが多いことが判りました。そこで、各々のバス検出処理を切り離せるようにした hwsetup を用意しました。変更した、knoppix-autoconfig は新しい hwsetup に対応し、シリアルバスの検出処理を別のプロセスで行います。

- mkxf86config

既存の knoppix-autoconfig スクリプトから X サーバの設定ファイル生成のために呼び出される mkxf86config は、モニターの検出に時間を要する場合があります。既存の mkxf86config では、その処理を 2 回実行していたので、2 回目の実行部分を、1 回目の処理結果を再利用するように変更しました。

5. KNOPPIX の起動高速化

ここでは、ライブ CD である KNOPPIX を対象として実際に起動高速化を行う手順について説明します。KNOPPIX の起動高速化では、CD-ROM に格納する cloop イメージファイルの最適化を行うことが主な作業となります。

cloop イメージファイルの最適化には、cloop 中のデータブロックがどのように読み出されているかの情報が必要になります。この読みだし具合の情報である、ブロック参照状況データは、前述の通り、実際にシステム起動して取得します。このブロック参照状況データを取得するため、プロファイリング機能の付いた最適化イメージ対応 cloop ドライバを起動高速化の対象とする KNOPPIX に導入します。

次に、このプロファイリング機能のついた cloop ドライバを組み込まれた KNOPPIX をプロファイリングモードで実際に動かし、cloop イメージ中のデータブロックの参照状況を取得します。

そして、この取得したブロック参照状況を元に cloop イメージのブロック配置を最適化し、再び ISO9660 のイメージファイルにまとめることで、起動高速化が適用された KNOPPIX が完成します。

☆ 注意 ☆

- この章の操作を行うには、LCATが必要となります。LCATのインストール方法は、第3章を参照してください。
- この章の操作は全て root 権限で行います。操作ミスにより、開発環境を破壊する恐れもあります。起動高速化を行う際には、この章手順を一読し、理解してから作業して下さい。
- 比較的サイズの大きなファイルを扱います。ハードディスクの空き容量を確認してください。適用するメディアが CD の場合、約 10GB 程空き領域が必要です。

5.1. 最適化イメージ対応 cloop ドライバの導入

ここでは、最適化イメージ対応 cloop ドライバを高速化を施す対象となる KNOPPIX に組み込む手順を説明します。

まず、適用前 KNOPPIX の ISO イメージファイル(ここでは, knoppix_4.0.2cd_j.iso) に格納されているファイル群を作業用ディレクトリにコピーします。

```
# mkdir ./tmp_mount
# mount -t iso9660 -o loop,ro ./knoppix_4.0.2cd_j.iso ./tmp_mount
# cp -pRv ./tmp_mount ./MASTER
# umount ./tmp_mount
```

これで、“./MASTER” 配下に適用前 KNOPPIX の ISO イメージ中に格納されているファイルが揃いました。

KNOPPIX では、cloop ドライバが起動時に参照する仮ルートファイルシステム initrd のイメージファイル minirt.gz に収録されていますので、このイメージファイルを差し替える準備を行います。

```
# mv ./MASTER/boot/isolinux/minirt.gz
    ./MASTER/boot/isolinux/minirt.gz.org
# gzip -cd ./MASTER/boot/isolinux/minirt.gz.org > ./miniroot
# mount -t ext2 -o loop ./miniroot ./tmp_mount
```

これで、ループバックイメージ miniroot が、“./tmp_mount” マウントされました。

まず、既存の cloop ドライバを最適化イメージ対応 cloop ドライバを差し替えます。バイナリインストールの場合は、“/lib/modules/2.6.12/kernel/drivers/block/cloop.ko”，ソースからコンパイルした場合は、コンパイルを実行したディレクトリ配下“lcat_1.0/cloop-2.02-opt/cloop.ko”を、“./tmp_mount/modules” 配下にコピーします。

```
# mv ./tmp_mount/modules/cloop.ko \
    ./tmp_mount/modules/cloop.ko.org
# cp <コピー元>/cloop.ko ./tmp_mount/modules
```

次に、最適化イメージ対応 cloop ドライバのプロファイリングモードを利用できるよう、linuxrc を修正します。具体的には、insmod で cloop のドライバをロードする処理の部分に、引数処理の追加を行います(リスト 1)。リスト 1 の行番号は、KNOPPIX4.0.2CD 日本語版の linuxrc です。

```
# cp -a ./tmp_mount/linuxrc ./tmp_mount/linuxrc.bak
# emacs -nw ./tmp_mount/linuxrc
```

```

396 mount_knoppix()
397 {
398   if test -n "$FOUND_KNOPPIX" -a -f \\\ ← 一行
"$1/$KNOPPIX_DIR/$KNOPPIX_NAME"; then
----   ここから： コメントアウト
399   # # DEBUG
400   # # echo "6" > /proc/sys/kernel/printk
401   # $INSMOD -f /modules/cloop.ko \\\ ← 一行
file="$1/$KNOPPIX_DIR/$KNOPPIX_NAME"
----   ここまで： コメントアウト
----   ここから： 追加
        # accelerated-knoppix : cloop-opt.ko
        #
        echo "6" > /proc/sys/kernel/printk
        THREAD="thread_mode=1"
        CHKBLK="0"
        for i in $CMDLINE; do
            case "$i" in chkblk=*|CHKBLK=*) eval $i; \\\ ← 一行
[ -n "$chkblk" ] && CHKBLK="$chkblk" ;; esac
            case "$i" in *noclpthread*|*NOCLPTHREAD*) THREAD="" ;; esac
        done
        echo ""
        $INSMOD -f /modules/cloop.ko chkblks=$CHKBLK \\\ ← 一行
"$THREAD" file="$1/$KNOPPIX_DIR/$KNOPPIX_NAME"
        echo "0" > /proc/sys/kernel/printk
----   ここまで： 追加
402   mountit /dev/cloop /KNOPPIX "-o ro" || FOUND_KNOPPIX=""
403   # Allow multi-image KNOPPIX mounts
404   if [ -n "$FOUND_KNOPPIX" -a -x "$DYNLOADER" \\\ ← 一行
-a -x /KNOPPIX/sbin/losetup ]; then
----   ここから： コメントアウト
405   #   echo ""
----   ここまで： コメントアウト
406   echo -n "${CRE} ${GREEN}Found primary KNOPPIX compressed
image at
${MAGENTA}$1/$KNOPPIX_DIR/$KNOPPIX_NAME${GREEN}.${NORMAL}"
407   for c in 1 2 3 4 5 6 7; do

```

リスト 1 : linuxrc の修正(最適化イメージ対応 cloop ドライバのプロファイリングモードの追加)

また,cloopイメージ中のデータブロックを先読みする処理を追加しておきます(リスト2).
KNOPPIX4.0.2CD日本語版では,"toram", "tohd", "bootfrom"のブートオプションを処理する部分(577-586行目)を先に実行し,その結果を用いてデータブロックの先読み処理の実行を判断します.また,データブロックの先読みを行いながらcloopイメージファイルをマウント(575-576行目)します.

```

496 copy_to()
497 {
    |
573 }
574
---- ここから： 移動
577 COPYTO=""
578 BOOTFROM=""
579 DO_REMOUNT=""
580 REAL_TARGET=""
581 UNIONFS=""
582
583 case "$CMDLINE" in *toram*) DO_REMOUNT="yes"; \ \ ← 一行
COPYTO="ram"; ;; esac
584 case "$CMDLINE" in *tohd=*) DO_REMOUNT="yes"; \ \ ← 一行
COPYTO="hd"; ;; esac
585 case "$CMDLINE" in *bootfrom=*) DO_REMOUNT="yes"; \ \ ← 一行
BOOTFROM="yes" ;; esac
586
---- ここまで： 移動
---- ここから： 追加
#
# accelerated-knoppix : cloop blocks readahead
#
if test -z "$DO_REMOUNT" -a -n "$FOUND_KNOPPIX" ; then
if test -x /accel/cloopreadahead ; then
CLOOPREADAHEAD="yes";
case "$CMDLINE" in *nocbr*|*NOCBR*) CLOOPREADAHEAD=""; ;; esac
if test -n "$CLOOPREADAHEAD" -a -f // ← 一行
/cdrom/${KNOPPIX_DIR}/${KNOPPIX_NAME}.boot.lst ; then
echo ""
echo -n " ${GREEN}Reading cloop // ← 一行
blocks...${BLUE}(Backgrounding)${NORMAL}"
/accel/cloopreadahead /cdrom/${KNOPPIX_DIR}/${KNOPPIX_NAME} \
/cdrom/${KNOPPIX_DIR}/${KNOPPIX_NAME}.boot.lst \
> /.cloopreadahead.log 2>&1 &
fi
fi
fi
---- ここまで： 追加
---- ここから： 移動
575 mount_knoppix /cdrom
576
---- ここまで： 移動
587 # Remount later after copying/isoloading/driverloading?
588 # pre-test if everything succeeded
589 if test -n "$DO_REMOUNT" -a -n "$FOUND_KNOPPIX"

```

リスト2 : linuxrc の修正(cloop イメージ中のデータブロックを先読みする処理の追加)

linuxrc の編集が終了したら, cloop イメージ中のデータブロックを先読みするコマンド cloopreadahead を導入します。バイナリインストールの場合は, “/usr/local/bin/cloopreadahead”, ソースからコンパイルした場合は, コンパイルを実行したディレクトリ配下 “lcat_1.0/util/cloopreadahead” を “./tmp_mount/accel” 配下にコピーします。

```
# mkdir ./tmp_mount/accel
# cp <コピー元>/cloopreadahead ./tmp_mount/accel
```

上記作業の全てが終わったらループバックイメージ miniroot のマウント解除を行い, minirt.gz として配置します。

```
# umount ./tmp_mount
# gzip -c ./miniroot > ./MASTER/boot/isolinux/minirt.gz
```

これで, プロファイル用 KNOPPIX のファイル一式が ./MASTER 配下に揃いました。これを ISO イメージファイル (knoppix_4.0.2cd_j_prof.iso) にまとめます。

```
# mkisofs -l -r -J -V "KNX402CDJ_PROF" -hide-rr-moved -v \
  -b boot/isolinux/isolinux.bin -c boot/isolinux/boot.cat \
  -no-emul-boot -boot-load-size 4 -boot-info-table \
  -o ./knoppix_4.0.2cd_j_prof.iso ./MASTER
```

プロファイル機能の付いた最適化イメージ対応 cloop ドライバを導入したプロファイル用 KNOPPIX の ISO イメージファイル (knoppix_4.0.2cd_j_prof.iso) が作業ディレクトリ下に生成されます。

次は, この最適化イメージ対応 cloop ドライバを導入した KNOPPIX を実機で起動し, プロファイリングを行います。

5.2. プロファイリングモードでの起動によるブロック参照状況データの取得

ここでは、前節 5.1.「最適化イメージ対応 cloop ドライバの導入」で作成したプロファイル用 KNOPPIX を用いて、ブロック参照状況データを取得する手順を説明します。

前説で作成したプロファイル用 KNOPPIX を用い、実機もしくは QEMU などの仮想 PC を用い起動します。その際、次のブートオプションを指定することでプロファイリングモードで起動します。

```
boot: knoppix chkblk=10000 nocbr
```

デスクトップの起動完了後コンソールを開き、以下の操作でシステム起動時のブロック参照状況データを採取します。

```
# cat /proc/cloop/read_blocks > /tmp/boot.read_blocks
```

これで、システム起動時のブロック参照状況データが“/tmp/boot.read_blocks”に保存されました。

KNOPPIX に収録されているアプリケーションの中で、利用頻度の高いものがあればそのアプリケーションも高速化できます。ここでは、OpenOffice.org の起動高速化を行います。

アプリケーション起動時のブロック参照状況データ採取の前に、今までのブロック参照状況を以下の操作でリセットします。

```
# echo 1 > /proc/cloop/reset_read_blocks
```

次に、ランチャアイコン、メニューなど、なんらかの方法で対象とするアプリケーションである、OpenOffice.org を起動します。アプリケーション起動の完了を確認した後、コンソールでの以下の操作でアプリケーション起動時のブロック参照状況データを採取します。

```
# cat /proc/cloop/read_blocks > /tmp/appli.read_blocks
```

これで、アプリケーション起動時のブロック参照状況データが“/tmp/appli.read_blocks”に保存されました。数種類のアプリケーションが対象となる場合は、それぞれのアプリケーションに対して起動・終了を行った後、ブロック参照状況データを採取してください。

このプロファイルモードで採取した、システム起動時のブロック参照状況データファイル (boot.read_blocks)、アプリケーション起動時のブロック参照状況データファイル (appli.read_blocks) を、USB メモリーキーなどのリムーバブルメディア、もしくはネットワークを用い、開発環境である、前節 5.1「最適化イメージ対応 cloop ドライバの導入」で参照した参照した、“./MASTER”のある作業ディレクトリ配下に転送します。

5.3. ブロック参照状況の解析

ここでは、前節 5.2.「プロファイリングモードでの起動によるブロック参照状況データの取得」で採集したシステム起動時、及びアプリケーション起動時の各ブロック参照状況データファイルについて clooprofiler を用いブロック参照状況を解析する手順を説明します。

前節で転送したファイル(boot.read_blocks, appli.read_blocks)はテキストファイルですのでそのまま閲覧できますが、ブロック参照状況可視化ツールである clooprofiler を用いることで CD-ROM ドライブのシークの発生具合が判ります。

clooprofiler を以下の操作で起動します。

```
# clooprofiler
```

clooprofiler の起動完了後、メニューバー『File』→『Openread_blocks』を選択し、ファイルダイアログを用い転送したブロック参照状況データファイルを指定します。データの読み込み後、[>] ボタンを押下することによって、実時間で cloop 中のブロック参照をアニメーションで表示します。表示領域のブロック解像度の変更は、[block/Cell] スピンボタンで行います。

ブロック参照状況のアニメーションは、デフォルトではブロック ID で描画しています。この表示を実際の cloop イメージファイル中のオフセット位置で描画するには、ブロック参照状況データを採取する際に対象とした cloop イメージファイルが必要になります。cloop イメージファイルの指定は、メニューバー『File』→『Open CLOOP』を選択し、ファイルダイアログを用い cloop イメージファイルを指定します。cloop イメージファイルは、作業用ディレクトリ配下、“./MASTER/KNOPPIX/KNOPPIX”にあります。cloop イメージの読み込み後、メニューバー『View』→『draw resource』→『cloop header』を選択することにより、cloop イメージファイル中の位置で描画するモードになります。

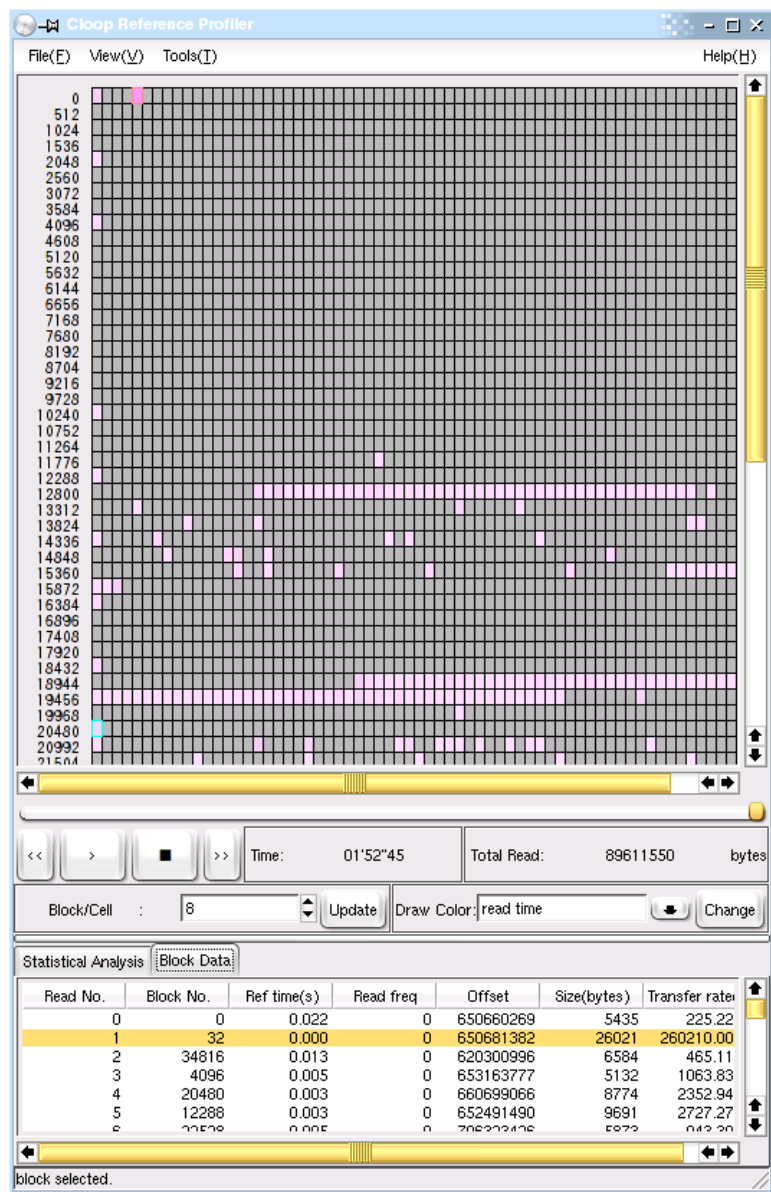


図 23:clooprofiler によるブロック参照状況の可視化

5.4. 既存 cloop イメージファイルのデータブロック配置最適化

ここでは、前節 5.2「プロファイリングモードでの起動によるブロック参照状況データの取得」で採集したシステム起動時、及び、アプリケーション起動時の各ブロック参照状況データを元に、既存 cloop イメージ中のブロック配置を最適化する手順を説明します。

cloop イメージのブロック配置を最適化するには、LCAT のユーティリティコマンド `cloopoptimizer` を用いる方法、もしくは、`cloopprofiler` の GUI を用いる方法の二通りあります。ここでは、コマンド `cloopoptimizer` を用いる方法で行います。`cloopprofiler` を用いる場合は、4.2.1「ブロック参照状況プロファイラツール (`cloopoptimizer`)」を参照してください。

以下の操作で、既存 cloop イメージファイルを別名 (`KNOPPIX.normal`) に変更した後、`cloopoptimizer` を実行し、最適化 cloop イメージファイル “`./MASTER/KNOPPIX/KNOPPIX`” を作成します。

```
# mv ./MASTER/KNOPPIX/KNOPPIX ./KNOPPIX.normal
# cloopoptimizer ./KNOPPIX.normal ./boot.read_blocks \
  ./appli_read_blocks > ./MASTER/KNOPPIX/KNOPPIX
```

これで、最適化済み cloop イメージファイルを “`./MASTER`” 配下に用意できました。

5.5. 起動時の先読み指定ブロックリストの作成

ここでは、前節 5.2「プロファイリングモードでの起動によるブロック参照状況データの取得」で採取したシステム起動時のブロック参照状況データから、先読みする cloop 中のデータブロックのリストを作成する手順を説明します。

先読み指定ブロックリストを作成するには、LCAT のユーティリティコマンド `rblk2bl` を用いる方法、もしくは、`cloopprofiler` の GUI を用いる方法の二通りあります。ここでは、コマンド `rblk2bl` を用いる方法で行います。`cloopprofiler` を用いる場合は、4.2.1「ブロック参照状況プロファイラツール (`cloopoptimizer`)」を参照してください。

以下の操作で、システム起動時のブロック参照状況データファイルから、起動時に先読みする cloop イメージ中のデータブロックリスト、先読み指定ブロックリスト “`./MASTER/KNOPPIX/KNOPPIX.boot.lst`” を作成します。

```
# rblk2bl ./boot.read_blocks > ./MASTER/KNOPPIX/KNOPPIX.boot.lst
```

これで、起動時の先読み指定ブロックリストを “`./MASTER`” 配下に用意できました。

5.6. 最適化 cloop イメージファイルを格納した iso イメージファイルの作成

前節 5.1.「最適化イメージ対応 cloop ドライバの導入」、5.4.「既存 cloop イメージファイルのデータブロック配置の最適化」、及び、5.5.「起動時の先読み指定ブロックリストの作成」によって、最適化イメージ対応 cloop ドライバ、最適化済み cloop イメージファイル、起動時の先読み指定ブロックリストが“./MASTER”配下に揃いました。

これを、以下の操作で、ISO イメージファイル(knoppix_4.0.2cd_j_AC.iso)にまとめます。

```
# mkisofs -l -r -J -V "KNX402CDJ_AC" -hide-rr-moved -v \  
-b boot/isolinux/isolinux.bin -c boot/isolinux/boot.cat \  
-no-emul-boot -boot-load-size 4 -boot-info-table \  
-o ./knoppix_4.0.2cd_j_AC.iso ./MASTER
```

これで、cloop イメージを最適化した KNOPPIX の ISO イメージファイル “./knoppix_4.0.2cd_j_AC.iso” が完成しました。

5.7. 更なる起動の高速化

本章では、cloop イメージの最適化を中心に説明しました。ライブ CD の起動高速化には、周辺機器検出の非同期化など起動シーケンスを最適化する手法もあります。これは、ライブ CD の種類やバージョンによって対処が異なります。

KNOPPIX4.0.2CD 日本語版での適用例を LCAT アーカイブ“lcat_1.0/KNOPPIX/”配下に納めました。参考にしてください。

6. 謝辞

LCATの開発は、独立行政法人情報処理推進機構(IPA)2005年度「オープンソースソフトウェア活用基板整備事業『CD/DVD起動Linuxの速度改善ドライバの開発』」プロジェクトの一部として行われました。IPAからは、本プロジェクトの計画、開発、検取などの進行について幅広い御支援を頂きました。厚い御支援に感謝いたします。

この開発がきっかけとなり、オープンソースカンファレンスや関西オープンフォーラムのKNOPPIX-BOF、そして横浜Linux Users Group(YLUG)が主催するカーネル読書会で本プロジェクトを紹介させて頂く機会がありました。お話の場を与えて下さった方々、参加して頂いた方々にも感謝致します。また、YLUG-MLをはじめ、小江戸LUG-ML、KNOPPIX-MLで有用な情報を提供し議論して下さいました方々、あとBlogやインターネットの掲示板で本プロジェクトにコメントして頂いた方々にも感謝致します。

ありがとうございました。

Team Accelerated KNOPPIX

Advanced Technology Research Dept.

Alpha System, Inc.

<mailto:knoppix@alpha.co.jp>

