

CD/DVD から起動する Linux の起動高速化

北川 健司[†] 丹 英之[†] 阿部 大将[†] 千葉 大作[†] 須崎 有康[‡] 飯島 賢吾[‡] 八木 豊志樹[‡]

[†] {kitagake, tanh, abeda, chibad}@alpha.co.jp

[‡] {k.suzaki, k-ijima, yagi-toshiki}@aist.go.jp

[†] 株式会社 アルファシステムズ

[‡] 独立行政法人 産業技術総合研究所

概要： KNOPPIX に代表されるライブ CD は、CD-ROM ドライブのある PC をインストール無しで即 Linux マシンとして起動できる。また、リードオンリーな特性をもつ CD メディアにシステムが格納されているため、トラブル発生時には電源のリセットが可能であり家電製品に近い感覚で使用できる。更に CD メディアは、安価に大量複製が可能であるため、幅広く私たちの生活に浸透している。このように CD から起動できる OS には多くの可能性と影響力があると言える。しかし、ライブ CD を使用した人の大半はその遅さにストレスを感じ、非常用 OS としての用途に終始してしまうのが現状である。そこで我々は、ライブ CD の遅さを解決し、利便性を向上させる事にした。

1. はじめに

ライブ CD の特徴は、物理的に書き換えのできないメディアからの OS 起動ができる事である。つまり、システムに異常があっても電源を入れ直せば障害が残る事もなく簡単に復帰できる。この高ロバストな性質に加え、CD メディアは安価に大量プレスが可能である。

このような特徴を持ったライブ CD を学校教育現場に持ち込み実証実験を行った[1]。実験終了後、被験者から回収したアンケートには起動速度の遅さを指摘するものが多かった。ライブ CD からの起動は、CD ドライブという起動デバイスの性質上、システムが利用できるまでに時間を要する。ライブ CD でデスクトップ OS を実現するには起動速度改善は必須項目であると言える。そこで、我々はアプリケーションを含めたシステム全体の起動時間を改善することにした。今回は、複数存在するライブ CD の中から周辺機器検出に優れていると評判のある KNOPPIX[2]について起動時間改善の検討を行った。

2. KNOPPIX とは

KNOPPIX とは、ドイツの Klaus Knopper 氏により開発が進められている Debian GNU/Linux ベースの 1CD Linux ディストリビューションであり、産業技術総合研究所(AIST)で日本語化、及び配布[3]が行われている。

KNOPPIX の CD には、ブートローダ(isolinux)とカーネル(linux), initrd(minirt.gz), そして、ルートファイルシステム(cloop イメージファイル)の4つが格納されている。

KNOPPIX 起動の手順は、まず、ブートローダにより起動されたカーネルが minirt.gz を RAM ディスクとして展開し、CD ドライブをカーネルからマウントさせるなど、CD 内の cloop ファイルをマウントする為の最低限の環境を整え、最終的に cloop ファイルをルートファイルシステムとしてループバックマウントする。その後は通常の Linux のようにデスクトップ環境を整える。

cloop とは、圧縮ループバックデバイスの事で、圧縮ブロック伸張機能を付加したループバックデバイスである。KNOPPIX では通常 64KB 単位

でルートイメージを圧縮し、**cloop** イメージを作成する。ただし、圧縮後のブロックはシーケンシャルに配置され、ブロックサイズの変化に対応できない為、作成後には変更ができないブロックデバイスとなっている。KNOPPIX では、ルートファイルシステムを **cloop** ファイルとして扱うことで、CD メディア(700MB)という限られたリソースに約 1.8GB の情報を格納することを可能とする。

3. CD 起動が遅い原因

我々は、まず KNOPPIX の起動速度のボトルネックとなる箇所を特定するため、システムの起動シーケンス、物理デバイス(CD ドライブ)、論理デバイス(cloop)の観点から調査を行った。

➤ 起動シーケンスの解析

我々はシステム起動時の遅い原因を探る為、Bootchart[4]を組み込み、起動シーケンスの解析を行った。Bootchart とは、システム起動時の top や iostat の結果を記録し、CPU や DiskI/O、プロセス生成状況を時系列に視覚化するシステム起動プロファイラである。KNOPPIX 起動の様子を Bootchart により視覚化した結果、大きく 2 つのボトルネックが明らかとなった。

- (1) 起動プロセス前半に周辺機器の自動認識・設定を行う **hwsetup** 処理中に殆ど DiskI/O が発生していない。
- (2) 起動プロセス後半の X サーバ、デスクトップ環境 KDE 起動時には CPU 利用時に I/O 待ちが多発している。

➤ 物理デバイスの解析

CD ドライブの性能は一般的に n 倍速という表現が用いられている。1 倍速時のデータ転送速度は 150KB/sec である。また、CD ドライブの回転制御方式には、大別して線速度一定方式(CLV,

Constant Linear Velocity)と角速度一定方式(CAV, Constant Angular Velocity)がある。CLV 方式は音楽 CD(CD-DA)など常に同じ速度でデータを読み込む必要がある場合に用いられ、ピックアップが記録面をなぞる速度が一定になるようにピックアップの位置によって回転数を制御する。それに対し、CAV 方式は回転速度が一定である為、外周程、線速度が増し、データの読込速度が向上する。また、音楽 CD と異なりランダムアクセスが多い場合にも、CAV 方式の方がピックアップの移動だけで記録メディアのあらゆる場所のデータを高速に読込可能であるため都合が良い。

つまり、 n 倍速と謳っている CD ドライブが、常にその転送速度で読み込めるとは限らない。

ここで、実際にいくつかのパソコンの CD ドライブについて、ピックアップの位置と単位トラック(50MB)あたりの読込時間を計測したものを図 1 に示す。測定に使ったライブ CD(Linux 2.4.27, DMA: on, block size: 1024KB)は、50MB のファイルを内側から 13 トラック(650MB)配置したもので、容量は実際の KNOPPIX に格納されている **cloop** ファイルと同じ割合を占める。

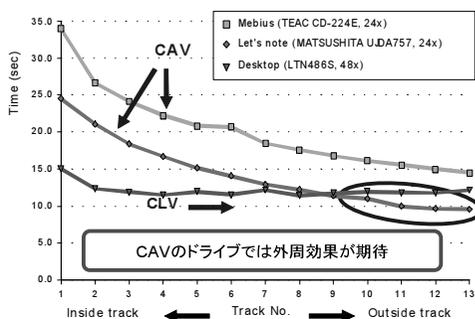


図 1. 回転制御方式とピックアップの位置による転送速度の推移

CLV 方式のドライブはピックアップの位置に依らずほぼ横ばいである。一方、CAV 方式では、

外周程その転送速度が向上しており、内側と外側の転送速度差は2.5倍程になる。つまり、起動に必要なファイルがCDの内側に集中していた場合、それだけで起動速度に弊害をもたらすと言える。

▶論理デバイスの解析

起動シーケンスの解析(2)についてももう少し掘り下げてみる。KNOPPIXではシステム起動時に必要なファイルをcloopから参照する。仮想ブロックデバイスであるcloopはアクセス要求のあったセクタを含んだcloopブロックを随時伸張することで、圧縮ファイルをファイルシステムとして透過的に利用できる機能を提供している。つまり、CDへのアクセス単位はcloopブロック単位であることが判る。

そこで我々は、cloopイメージへのアクセス状況を解析するためCloop Reference Profilerを作成した。図2はKNOPPIX起動時のcloopイメージへのアクセス状況をCloop Reference Profilerにより視覚化したものである。アクセス状況を動画再生すると読込順はバラバラであり、大量のシークが発生している事が判る。ここでcloopイメージはKNOPPIXのCDの大部分を占

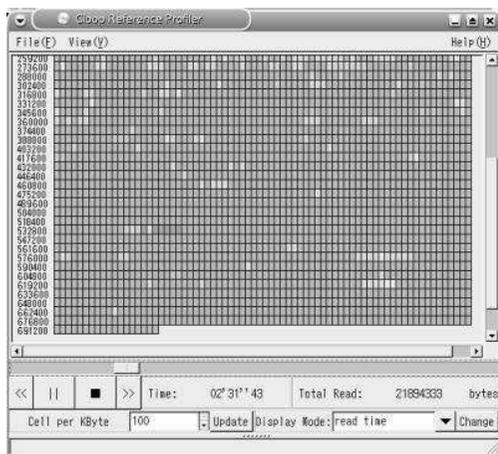


図2. Cloop Reference Profilerによる
cloopイメージファイルの可視化

表1. KNOPPIX3.8.2 起動時の統計情報

総参照ブロック数	3,668
総参照サイズ (KB)	68,156
伸張後のサイズ (KB)	234,752
総参照時間 (sec.)	91.971
平均転送速度 (KB/sec.)	723
ユニーク参照ブロック数	2,172
ユニーク参照サイズ (KB)	45,074

める為、cloopへのアクセスはそのままCDへのアクセスと捉える事ができる。つまり、起動後半のI/O待ちの原因はシークの大量発生によるCDからのデータ読み込みの遅さに起因する。

4. 解決策の検討

前章での解析結果を基に、KNOPPIX起動速度改善の為の解決策を検討した。主な検討箇所は、CD全体を占めるcloopブロックのシークを抑える事、それをCDドライブからより効率的に読み込む事、そして、リソースを有効に利用する事である。

▶cloopブロックの最適化

表1はKNOPPIX3.8.2日本語版をCPU:P4 2GHz, RAM:512MB, CD:40xのPCで起動した時のKDE起動完了までの統計情報である。システムが使用できるまでにcloopブロック3,668個、容量に換算すると約68MBをCDから読み込む。単に情報量だけで判断すると24倍速のCDドライブ(実効速度17倍速(2,550KB/sec)と仮定)であれば27秒弱で読み込む事になる。しかし、実際にはcloopイメージ全体にブロックが散在しており、シーク・回転待ちが多量に発生する。その為、本来の転送速度が発揮できず、平均転送速度は5倍速程度(723(KB/sec))まで低迷する。

起動時には一部のハードウェア構成の違いによるドライバを除けば、ほぼ同じファイルが必要とするため、ブロックの配置を過去のシステム起動時に読み込まれた順に並べ替える事でシークが削減できると考えた。

➤cloop ブロックの最適配置

2章3節で述べた通り、CDドライブの制御方式にはいくつか種類がある。現在流通しているCDドライブは、CAV、CLVをはじめ、ディスクを内周から幾つかのゾーンに分けゾーン単位で制御を行うZCAV、ZCLV方式や、ディスクの内側はCAV、外側をCLVで制御するPCAV方式などの複合型も存在する。我々は、どのCDドライブの制御方式の場合でも、転送速度は「内側≦外側」であるととし、起動時に参照するブロックは外周配置することにした。

➤ファイルの先読み

起動シーケンスの解析結果(1)で述べた通り、周辺機器検出中にはディスクアクセスが殆ど行われていなかった。そこで、このI/Oの空き時間に以後必要となるファイルを先読みする事で、リソースを有効に利用できると考えた。

過去にKNOPPIXの起動時にシステムコール`readahead()`を呼び出し、以後必要となるファイルを先読みするという試みがあった。`readahead()`とは、指定したファイルをページキャッシュに読み込ませるLinux固有のシステムコールである。図3は、`knoppix_v3.7_20041208-20050107+bootchart.iso(linux-2.6.9)`をCPU: Celeron 2GHz, RAM:768MB, CD:24xのPCで起動した時の`readahead`の有無による起動時間を示す。また、物理メモリは`boot`オプション`mem=`により指定した。十分なメモリを搭載するPCでは速度向上が見られるが、搭載するメモリ量が少ないと先読みしたファイルがキャッシュから無くなってしまふ為、再度読み込む事となり、逆に起動が遅くなる。

すぐにキャッシュが無くなるのは、`readahead()`が指定のあったファイルを全てキャッシュに読み込んでしまうためである。ファイルサイズの大きいライブラリなどファイルの一部

しか参照しない場合でも、全てキャッシュに蓄えようとする為、潤沢なメモリを必要としてしまう。

この経験を踏まえ今回は、先読みする単位をファイルではなく、`cloop`ブロックとすることで、必要なメモリを抑える事ができるのではないかと考えた。

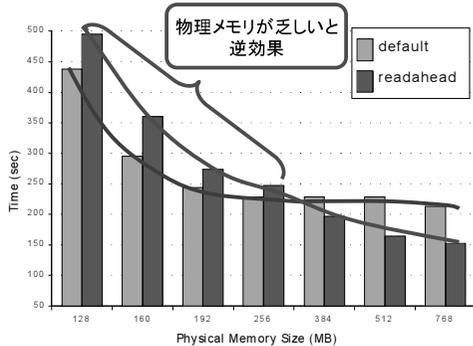


図3. readaheadの有無による起動時間

5. 実装と評価

解決策として検討したブロックの最適化、外周配置、ブロック先読みについて実装を行った。

まず、過去のブロック参照情報を取得する為、`cloop`ドライバにブロック番号、参照時刻、CD読み込みに掛かった時間等を保持する機能を追加した。これらのデータはシステム起動後に`procf`s経由で取得できる。このデータを基に起動に必要なブロックがシーケンシャルになるように`cloop`イメージファイルの最適配置を行う`cloop`イメージファイルコンバータの実装も行った。その際、CDドライブがCAVであった場合の特長を生かす為、外周配置を行う。また、ブロックの最適化によりイメージファイルのフォーマット変更が必要であった為、`cloop`ドライバの拡張を行った。

ブロック先読みに関しては、`cloop`イメージの指定されたブロックを読み込むコマンドを実装した。このコマンドは`cloop`デバイスのマウント

直後にブロック参照リストをパラメータとして実行され、カーネルの仮想メモリサブシステムのページキャッシュに格納する。

今回、ブロック配置の最適化を行ったライブCDはKNOPPIX Edu4.1(KNOPPIX3.7日本語版ベース)で、kernel-2.6.13.3とcloopドライバを改変し、cloopイメージファイルのユーザーランドはKNOPPIX Edu4.1のままである。そして、ベースとなるCDを作成し、そのCDの起動時のブロック参照情報をVMware5.0.0上で取得し、そのリストを基にブロック最適化処理を施して作成したKNOPPIXをAccelerated-KNOPPIXと呼ぶ。

Accelerated-KNOPPIXとそのベースとなったKNOPPIX Edu4.1とEdu4.1のベースとなったKNOPPIX3.7日本語版のシステム起動時間を図4に示す。計測したPCはCPU:PentiumIII-933MHz, RAM:256MB, CD:24xである。

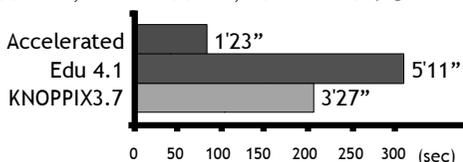


図4. 解決策を実装したKNOPPIXと各KNOPPIXの起動時間の比較

起動時間は、Accelerated KNOPPIXで83秒、KNOPPIX Edu4.1で311秒、KNOPPIX3.7で207秒であった。Edu4.1では、KNOPPIX3.7に比べ起動時にホームディレクトリへのファイルのコピーが数多く発生する為、その読込に余計に時間を要する。しかし、このEdu4.1を最適化したAcceleratedではEdu4.1の約27%の時間で起動が完了した。最適化前後のCDからの転送速度を表2に示す。

回転・シーク時間を抑えて、今までの約5倍の速度で必要な情報を読み込む事ができた。また、ファイルの先読みが十分に連続して行われており高速起動に繋がったと言える。

表2. 最適化前後の転送速度

	最適化前	最適化後
総参照ブロック数	4,920	
総参照サイズ (KB)	95,972	
総参照時間 (sec.)	124.056	26.288
平均転送速度(KB/sec.)	773	3,650

6. 更なる高速化の為の検討事項

ここまでの実装によりCDドライブの能力は十分に発揮出来るようになったと考える。今後更なるKNOPPIXの高速化を行う為に幾つかの事を検討している。

➤周辺機器検出の非同期化

周辺機器検出を行うhwsetupは、バス毎の検出を逐次処理しており、比較的時間を要する。この処理はクリティカルパス上に存在する為、システム起動時間に直接関係する。この処理を並列化、及びバックグラウンドで実行できるように改良を加え、非同期化する事で高速化に繋がると考えている。

➤cloop専用elevatorアルゴリズムの採用

現在のcloopはループバックデバイスと同様にブロックI/Oスケジューラを利用していない。ライブCDの起動に要する時間の殆どはシーク・回転待ち時間であることから、I/O要求を効率的に並べ替え、デバイスにI/O要求をディスパッチすることは、cloopブロックを最適配置していない箇所について有効に機能すると考えている。

➤cloopブロックのキャッシュ

システム起動の際、同じcloopブロックが何度も読み込まれる事が判っている。各レイヤにはそれぞれキャッシュがあり、これを排他的に制御できれば、キャッシュが重複しないため無駄なメモ

りを抑え、キャッシュヒット率を向上させる事ができると考えている。

▶参照する cloop ブロック数の削減

cloop 上に乗るファイルシステムを最適化する事で、伸張する cloop ブロックの絶対数を減らし、高速で効率的な起動ができるのではないかと考えた。cloop は仮想ブロックデバイスであるため、この階層上のファイルシステムは種類を問わない。今回は、ブロックアルゴリズムを採用している事、ネットワークから透過的にルートファイルシステムを利用できる HTTP-FUSE KNOPPIX[5]と相性が高い事等から、Ext2 ファイルシステム(以後 Ext2)に注目した。

Ext2 レイヤで利用されるデータブロック数を把握する為、kernel-2.6.11 の Ext2 に要求のあったブロック情報(間接ブロックを含む)を `procf`s 経由で取得できるように改良を加え KNOPPIX3.8.2 に実装した。表 3 は、CPU:PentiumIII 2GHz, RAM:512MB, CD:24x の PC にて KDE が起動完了するまでに読み込むデータ量をそれぞれのレイヤについて測定したものである。ただし、Ext2 における i ノードなどデータブロック以外の情報は計測対象外とする。cloop ブロックは同じブロックを何回も読み込む為、累計とユニークな値を表記した。また、cloop Layer のサイズは伸張後の容量である。

cloop ブロックを伸張して得られる有効バイトはかなり低めであると言える。仮に有効なデータブロックだけを詰めた cloop ブロックを作成できれば、cloop ブロック 1,379 個でシステムが起動で

表 3. レイヤ別の読み込みデータ量

	サイズ(KB)	ブロック数
Ext2 Layer (4KB/block)	88,228	22,057
cloop Layer (64KB/block)	(ユニーク) 139,200	2,175
	(累計) 235,584	3,681
iso9660 Layer	68,054	—

きる事になる。これは、読み込むブロックが 63%程になり、高速化が期待できると考えている。

7. まとめ

今回、KNOPPIX に cloop ブロックの最適化、最適配置、cloop ブロックの先読みを実装した。その結果、従来の KNOPPIX の半分以下の時間で起動するライブ CD を実現できた。また、更なる起動速度を改善するため、幾つかの改良点について考察した。特に Ext2 レイヤで実際に利用するデータブロック数を調べたところ、ユニークな cloop ブロックの 63%程のサイズでシステム起動できる事が判った。つまり、CD ドライブからの読込時間やブロックの伸張時間が削減される為、高速化に繋がると期待できる。

これらの起動時間を改善したライブ CD は、Accelerated-KNOPPIX として順次公開[6]する。

謝辞

本研究の一部は、IPA(情報処理推進機構) 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」[7]による。ご支援に深謝する。

参考文献&URL

- [1] IPA2004 年度, “学校教育現場におけるオープンソースソフトウェア活用に向けての実証実験”
<http://www.ipa.go.jp/software/open/2004/stc/eduseika.html>
- [2] knoppix,
<http://www.knopper.net/knoppix/>
- [3] KNOPPIX 日本語版
<http://unit.aist.go.jp/itri/knoppix/index.html>
- [4] Bootchart
<http://www.bootchart.org/>
- [5] 須崎, 八木, 飯島, 丹, “HTTP-FUSE KNOPPIX”, Linux Conference2005,
<http://lc.linux.or.jp/paper/lc2005/CP-02.pdf>
- [6] Accelerated-KNOPPIX
<http://www.alpha.co.jp/ac-knoppix/>
- [7] IPA2005 年度上半期, “オープンソースソフトウェア活用基盤整備事業”
<http://www.ipa.go.jp/software/open/2005/result.html>