

**“Development of speed improvement driver of
CD/DVD start Linux”**

Live CD boot acceleration manual

6-June-2006

Alpha Systems Inc.

Copyright and Trademark Notice

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

Alpha Systems Inc. are maintaining the copyright in this document. The reproduction, the re-distribution, and the correction have been permitted based on GNU General Public License(GPL). The only page that is not allowed to be modified is this page"Copyright and Trademark Notice"

Please contact us if you have modified the document.It will be reflected in the official version if necessary.

- (R)Linux is a registered trademark of Linus Torvalds in several countries.
- (R)Linux International is a registered trademark in the USA used pursuant to a license from Linux Mark Institute, authorized licensor of Linus Torvalds, owner of the Linux trademark on a worldwide basis
- (R)UNIX is a registered trademark of The Open Group in the USA and other countries.
- In general, the system name, the product name, and the brand name taken up in other this book are registered trademark/brand name of each development manufacturer.
- The TM mark is not described clearly in the text.

First edition issue on February 28,2006

Translation version issue on March 31,2006

Author: Alpha Systems Inc.

Issue: Alpha Systems Inc.

Printed in Japan

This book might be revised without a previous notice improving additionally of the product.

Update history

Date	Content of update
28-Feb-2006	· First edition
31-Mar-2006	· Translate version

Definitions

Term	Meaning
Live CD	Operating system that starts by CD or DVD1. Especially,it mentions about the one,that is able to start GNU/Linux in this document.
KNOPPIX	A kind of live CD. Mr. Klaus Knopper in Germany has developed based on Debian GNU/Linux.
Compression loop device (cloop)	A virtual block device.

Notations

◆The example of the UNIX input command is described in the frame below.

ef.)

```
# cloopprofiler
```

◆The example of the file output is described in the frame below.

ef.)

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

◆The example of the start option of Linux is described in the frame below.

ef.)

```
boot: kernel
```

◆The example of the start option of Linux is described in the frame below.

Element of window	Mark	example
Button of window	() Is bundled	(Play) , (Next)
Button of pull-down menu and pop-up menu	[[]] Is bundled	[[File]], [[View]]
Selection button	<<>> Is bundled	<<select>>

◆The example of the parentheses for each element is shown below.

- It operates it 1.
- It operates it 2.
- It operates it 3.

Contents

1. Outline.....	1
1.1. Start acceleration of live CD.....	1
1.1.1. Start speed problem of live CD.....	1
1.1.2. Live CD boot acceleration technique.....	2
1.1.3. Flow of start acceleration.....	4
1.2. Live CD boot acceleration toolkit “Live CD Acceleration Tool kit(LCAT)”	5
1.3. About this document.....	5
2. Development environment for boot acceleration.....	6
2.1. Recommended hardware.....	6
2.2. Needed software.....	6
3. Preparation for development environment.....	7
3.1. Installation from deb package.....	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. Installation from source archive.....	8
4. Live CD Acceleration Tool kit (LCAT).....	9
4.1. Outline of LCAT.....	9
4.2. Various utility commands.....	10
4.2.1. Block reference situation profiler tool (clooprofiler).....	10
4.2.2. Image optimization tool(cloopoptimizer).....	25
4.2.3. Block reading program(cloopreadahead, rblk2bl, appblk2bl).....	26
4.3. Driver for optimization image(cloop.ko).....	30
4.4. Patch files for KNOPPIX various scripts.....	32
5. Boot acceleration of KNOPPIX.....	34
5.1. The cloop driver's for optimization image introduction.....	35
5.2. Acquisition of block reference situation data by start in profiling mode.....	39
5.3. Analysis of block reference situation.....	40
5.4. Arrangement of block of data of existing cloop image file optimization.....	42
5.5. Making of lookahead specification block list when starting.....	42
5.6. Making of iso image file that stores optimization cloop image file.....	44
5.7. acceleration of further start.....	44
6. Address of thanks.....	45

1. Outline

This document is the manual of "Live CD Acceleration Tool kit(LCAT)". "LCAT" is the result of the one of the infrastructure construction plan for open source software executed,"development of the acceleration driver of the CD/DVD starting Linux" in the first half of 2005 undertaken by the Information-technology Promotion Agency,Japan(IPA) and Alpha Systems,Inc. LCAT is a live CD starting high speed toolkit, and can perform starting improvement in the speed of the existing live CD.This manual explains the procedure of performing starting improvement in the speed of Live CD using LCAT.

The function of toolkit LCAT that enables to accelerate boot speed is briefly explained in this document.

1.1. Start acceleration of live CD

In this section, it mentions how live CD works to accelerate boot speed.

1.1.1. Start speed problem of live CD

Live CD can start up the complete system set of OS from a CD-ROM or a DVD-ROM. Live CD has the following feature: CD media can be mass-produced at a low price,Live CD has high robustness since the CD is read-only.Moreover,the system has a comparatively strong device detecting function, and it is possible to be used by almost every PC. Live CD makes the best use of these features and could be used in many cases,such as education fields,system recovery in emergency, supplement to a magazine,distribution of trial version software etc.

However,live CD uses optical media which is known to be slow in seek so that the start time is much longer compare with the PC that uses OS which is directly installed in HD and users have to wait until it is finished.

- Reading from CD

All the files which is necessary for system start up are stored in the CD-ROM. Some CD-ROM drives have a notation that says "This drive can read at high speed in x24". However,the actual speed is much slower since it is only telling the Maximum speed of the continuous reading time.Especially, the seek of picking up caused in the random access to the file is very slow compared with the storage such as HD. This is one of the causes lengthening the average access time of the data that wants to be read.

- Boot sequence

Live CD enables to connect peripheral devices that can be used by Linux without any specific operation from user. This automatic detection and setting are processed in the start sequence of the system.But start sequence is not efficient since the equipments can only process one by one in the current implementation.Furthermore during the start sequence,data will be read by CD-ROM whenever it is needed.Therefore, when the I/O re

quests issued by process are congested, the picking up will increase that cause I/O to wait longer and lengthen the duration of the start sequence.

1.1.2. Live CD boot acceleration technique

This chapter explains the solution of the two issues that will cause the delay during start sequence.

- Optimization of file system image

To solve the CD reading problem in live CD, we optimize the image of the filesystem of live CD. So as not to generate the random access as little as possible, the datablock of the filesystem image of live CD is arranged. The seek time of CD-ROM drive will be decreased by arranging the order as it has been read by live CD. As a result, it comes to be able to transmit data that is necessary at a dash from CD-ROM to which the file system image has been optimized.

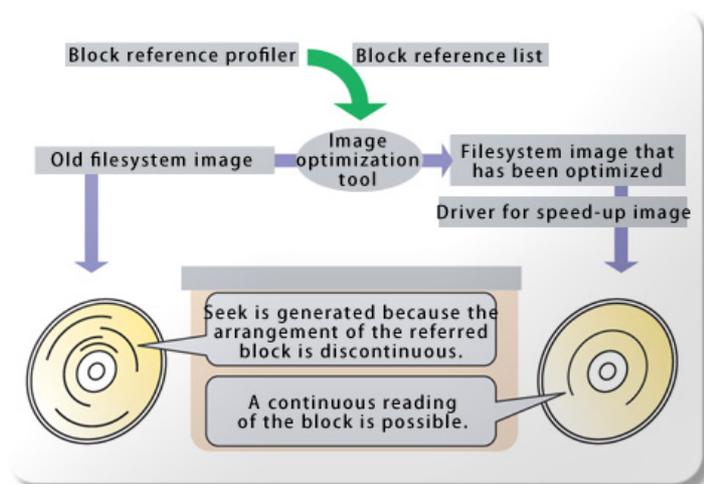


Figure 1: Outline of file system image optimization technique

Figure 1 shows the outline of the optimization of the file system image. "Block reference profiler" investigate the state of filesystem image read when live CD has started up, and outputs the result as the profile. "Image optimization tool" is permuted in order according to the reading order of data block filing the cloop image by using the obtained profiling data. The optimization filesystem image will be read by using "Driver for the optimization image", since the file format differs slightly.

- Optimization of boot sequence

Figure 1 shows the outline of the optimization of the filesystem image. "Block reference profiler" investigates the state of filesystem image read when live CD has started up, and outputs the result as the profile. "Image optimization tool" is permuted in order by which the data block filing the cloop image by using the data of the obtained profiling is read. The optimization filesystem image will be read by using "Driver for the optimization image", since

the file format differs slightly.

Figure 2 is the one that the outline of the start sequence optimization technique was shown. Advance the detection of peripherals by running parallel beginning the lookahead of the block of data in CD-ROM at the same time as starting and processing the start sequence. Moreover, the start time of the application can be shortened by reading the data needed to start the application ahead. The start sequence is optimized the entire reviewing based on these ideas.

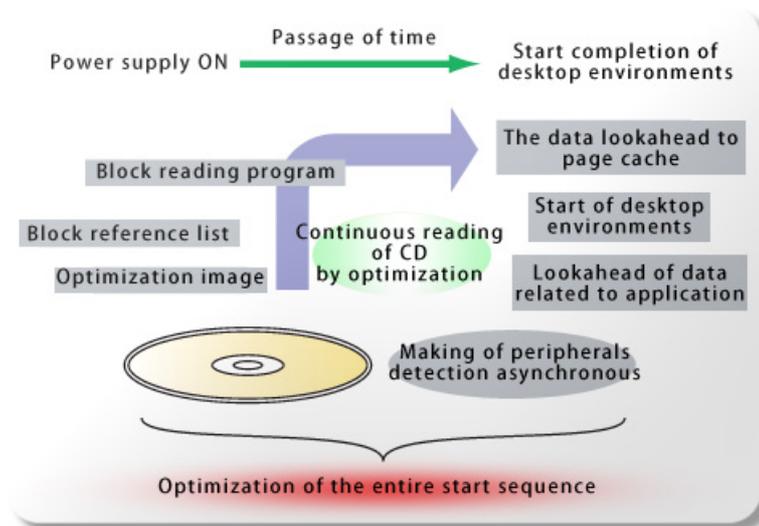


Figure 2: Outline of start sequence optimization technique

1.1.3. Flow of start acceleration

The flow of the start accelerated live CD is shown as follows.

- I. Create the ISO image that contains the driver for optimization image.
- II. Acquire and analyze the block reference situation.
- III. Block arrangement optimization
- IV. Create the start accelerated ISO image .

Figure 3 shows the concrete flow when the start acceleration is given to KNOPPIX by using LCAT. The blue colored part show the modified parts for acceleration.

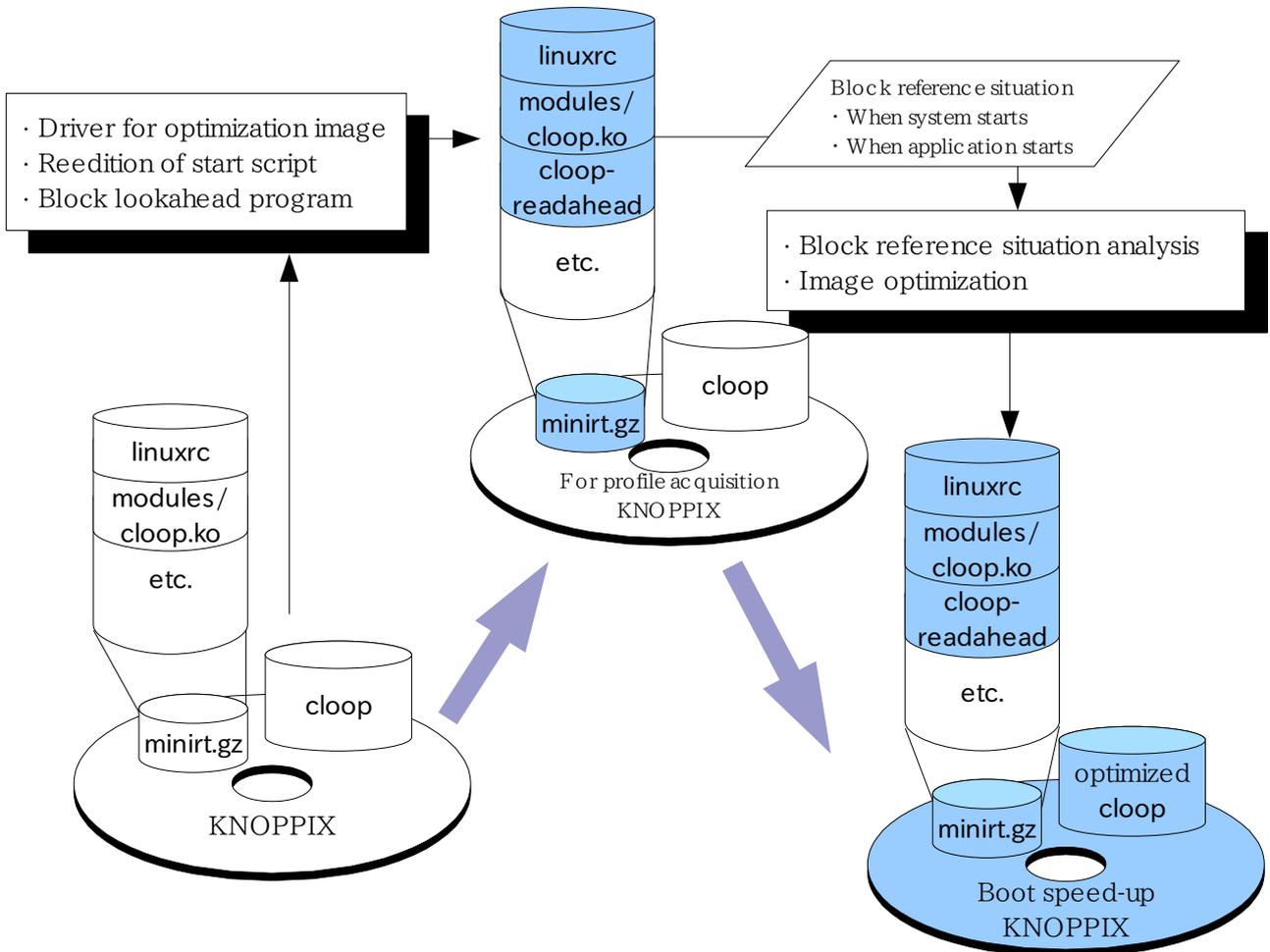


Figure 3:Flow of live CD boot acceleration

1.2. Live CD boot acceleration toolkit “Live CD Acceleration Tool kit(LCAT)”

LCAT is a toolkit which has been mounted to the live CD to apply the start acceleration technique. LCAT can optimize the block arrangement of the image file of compressed loop back block device cloop that has been adopted by such as KNOPPIX live CD. Also boot script, which is optimized for a boot sequence of KNOPPIX, is enclosed in LCAT package. The boot acceleration of live CD can be achieved by applying the tool in LCAT to the cloop image file and replacing the start script in initrd that is a temporary root file system for the boot

1.3. About this document

This document is the manual of “Live CD Acceleration Tool kit(LCAT)” which is the result of the first half open-source-software practical use infrastructure improvement enterprise "development of the acceleration driver of the CD/DVD starting Linux" of the 2005 fiscal year undertaken by the Information-technology Promotion Agency, Japan(IPA) and Alpha Systems, Inc.

This document is assumption of the use when a live CD developer applies the start acceleration to the KNOPPIX derivative. A live CD developer can do the start acceleration of the KNOPPIX derivative by referring to each README file collected in the package of this document and LCAT. Moreover, even when the start acceleration targets live CD other than KNOPPIX, this technique can be applied.

2. Development environment for boot acceleration

This section covers the hardware of the development environment recommended when the start acceleration of CD version KNOPPIX is done by using LCAT and the needed software.

2.1. The recommended hardware

Table 1 shows recommended environment that goes well with LCAT.

Table 1: Recommended system requirements

CPU	PC/AT compatible (i386) and 1GHz or more
Memory	512MB or more
Storage area	10GB or more
CD Drive	CD-R drive of 24 X speed or more

Note that if the hardware capacity doesn't fulfill the recommended environment, program would prone to delay.

2.2. The required software

Table 2 shows the software needed though all programs included in LCAT are operated.

Table 2: Necessary software list for LCAT

OS	Linux kernel 2.6.12(indispensability since 2.6)
Necessary library	libc6, development package libc6-dev (since 2.3.2) gtk2.0, development package libgtk2.0-dev (since 2.4) glib2.0, and development package libglib2.0-dev(since 2.0)

The behavior of LCAT has been confirmed in the following environments.

- KNOPPIX4.0.2 DVD version
- KNOPPIX4.0.2 DVD version HD installation
- Debian GNU/Linux Sarge 3.1
- Fedora Core 4 (Exclude kernel module)

3. Preparation for development environment

This section shows two methods for the installation of LCAT. One is to install from deb package, and the other is to compile from source file.

3.1. Installation from deb package

The deb package of LCAT consists of the following two packages.

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

Utility programs are included in the lcat_1.0.0_i386.deb. And, the cloop driver supporting optimized image (i. e. kernel module) is included in the lcat-cloop-module-2.6.12_1.0.0_i386.deb package.

This kernel module is gcc3.3 arranged for kernel2.6.12. In case using the cloop driver without this kernel, it is necessary to compile from source in suitable environment. The following sections show how to install each package.

3.1.1. lcat_1.0.0_i386.deb

This is the package including utility programs of LCAT but it doesn't contain the kernel module for the cloop driver. FOR installation, execute the following command by the root authority.

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

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

This is the package including kernel module for the cloop driver supporting optimized image. For installation, execute the following command by the root authority. Note that if a package cloop-module has been already installed, then it is necessary to uninstall the package of cloop-module to prevent packet crash.

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

The utility of LCAT is arranged in “/usr/local/bin/” when installing it from the deb package and the cloop driver for the optimization image is arranged in “/lib/modules/2.6.12/kernel/drivers/block/”.

3.2. Installation from source archive

Execute following commands in order to install from source archive data.

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

With the operation above, the decompressed file will be located into “/usr/local/bin/.” However, the installation directory can be arranged by using configure command with an option, “--prefix=<install path>”. Refer to the command below.

```
# make install
```

With the operation above, the utility and kernel module of LCAT will be installed. In case installing only utility, execute “make install” after changing the current directory to “lcat_1.0/util”. Kernel module of the cloop driver will be made as “lcat_1.0/cloop-2.02-opt/cloop.ko”.

4. Live CD Acceleration Tool kit (LCAT)

This section covers the outline of CD boot acceleration toolkit LCAT, cloop driver for the optimization image, and various utilities.

4.1. Outline of LCAT

Various utilities, drivers, scripts, and the functions to compose LCAT are shown below.

- Block reference situation profiler tool (clooprofiler)

The tool that visualizes the reference situation of the block by animation based on the block reference situation data file obtained by the profiling mode.

- Image optimization tool (cloopoptimizer)

The tool that relocates the data block in the cloop image according to the block reference situation data file.

- Block reading program (cloopreadahead, rblk2bl, appblk2bl)

Cloopreadahead is a program that reads the data block filing optimization cloop image file according to the reading block specification list. “rblk2bl” and “appblk2bl” are programs that create the reading block specification list from the block reference situation data file.

- Cloop driver for optimization image (cloop.ko)

A part of the cloop image file that has been relocated the datablock by image optimization tool differs from the default cloop image file in format. The cloop driver is a compressed loop device driver that corresponds to the format. Furthermore, it has the function to keep the reference situation of the data block which is profiling mode.

- Patch files for KNOPPIX various scripts

The process of the boot sequence has been arranged for the boot acceleration.

4.2. Various utility commands

This section shows the various utilities that compose LCAT.

4.2.1. Block reference situation profiler tool (clooprofiler)

Outline

Clooprofiler is the tool that visualize a block reference situation in the cloop image file. Clooprofiler will visualize a block reference situation by coloring the blocks and will animate a block reading state. And it can also be used as boot acceleration wizard of Linve CD since it has equivalent functions to other LCAT command line utilities such as cloopoptimizer, rblk2bl and appblk2bl.

Figure 4 will be displayed when you start the clooprofiler.

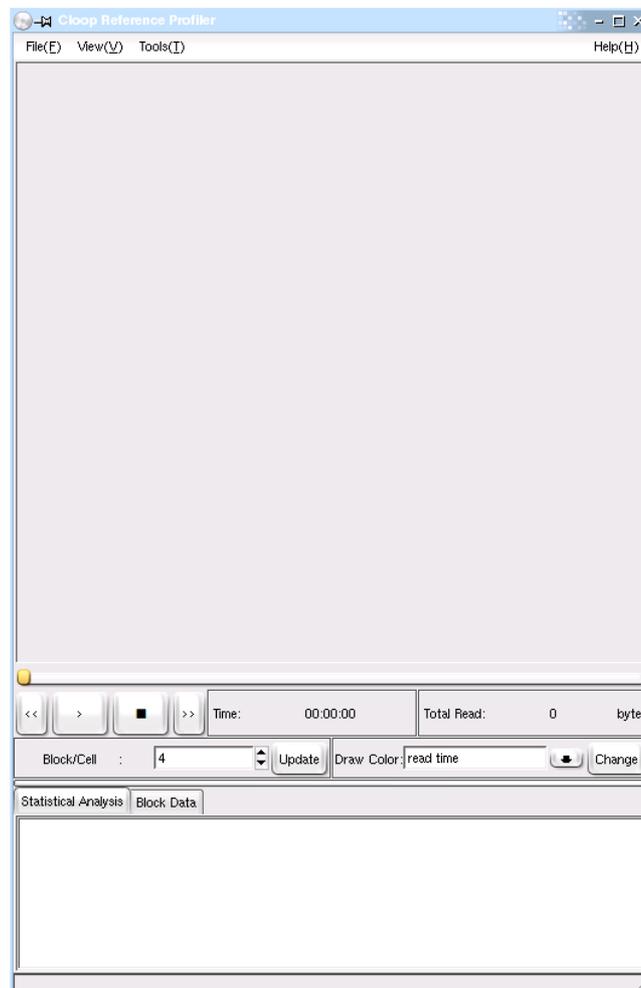


Figure 4: clooprofiler

The upper half of the window shows a block area that indicates a block arrangement of cloop. The block drawing area will be animated and change with time when block reference situation was chosen to play.

Cloopprofiler has two prominent functions that are block reference situation visualizing function and Live CD boot acceleration function.

- **Block reference situation visualizing function**
 - ◆ Animation display by block ID of block reference situation.
 - ◆ Animation display by block offset value of block reference situation.
 - ◆ Display of header information of cloop image file.
 - ◆ Control play, pause, fast-forward and rewind of the animation.
 - ◆ Color coding blocks separated by access time.
 - ◆ Color coding blocks separated by the number of accesses.
 - ◆ Change the graphic mode between amount of data and number of blocks.
 - ◆ Monitor a detail of a data block in the blockrendering area by pop-up window.
 - ◆ List the block reference situation
 - ◆ Choose the element and emphasize the block drawing area.
 - ◆ Change the playback duration of the animation of block reference situation by slider
- **Live CD boot acceleration function**
 - ◆ Create the optimized cloop image file by the cloop image optimization wizard.
 - ◆ Create the lookahead specification block list by the lookahead specification block list create wizard.
 - ◆ Initialize the block reference situation data of the cloop driver, supporting optimized image, starts by profiling mode.

Usage

Clooptimizer starts by the following commands.

```
$ clooprofiler
```

It explains the main use of clooprofiler.

- Animation display by block ID of block reference situation

Select `[[File]] -> [[Open read_block]]`(Figure 5) and block reference situation data file will open. It will visualize the drawing mode that shows Block reference situation based on Block ID.

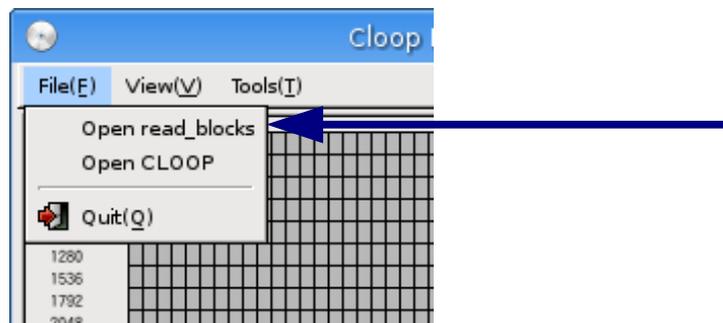


Figure 5: Select `[[Open read_blocks]]`

Control animation by the button and the slider in a window (Figure 6).



Figure 6:Button of control of animation of 'clooprofiler'

- Animation display by offset position of block reference situation

After opening block reference situation data file, select `[[file]] -> [[Open CLOOP]]` and Cloop image file will open(Figure 7) at file dialog. It will visualize the offset position of the block reference situation.

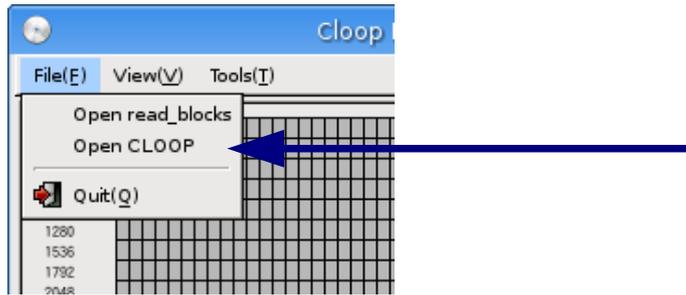


Figure 7: Select [[Open CLOOP]]

Select [[View]] -> [[draw resource]] and choose [[read blocks log]] or [[cloop header]] to switch between “block ID of Block reference situation” and “drawing mode at offset position”(Figure 8).

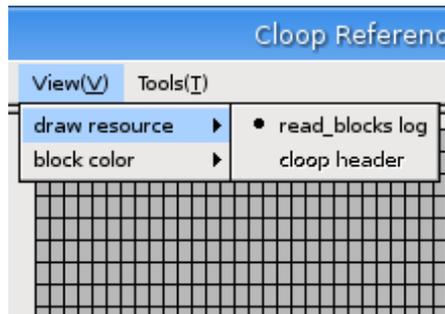


Figure 8: Select [[draw resource]]

- Optimization of cloop image

Select [[Tools]] -> [[Accelerate Wizard]](Figure 9). and Cloop Optimize Wizard dialog will open(Figure 10).

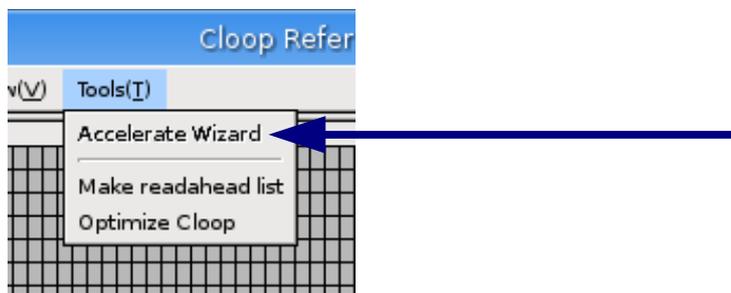


Figure 9: Select [[Accelerate Wizard]]

In the cloop image optimization wizard dialog, execute optimization will be executed by pressing |(Next)| button after specifying for the lookahead specification block list file to the “Block List” and specifying the save point of the optimized cloop image file in “Optimized Cloop”.

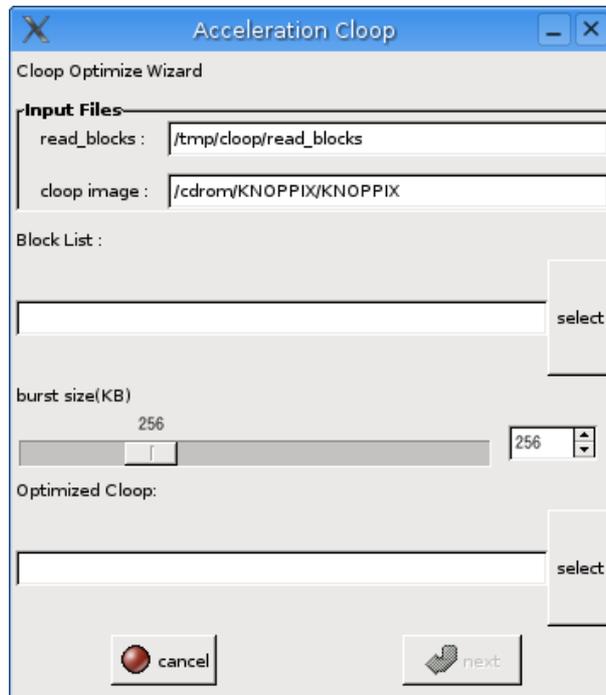


Figure 10: “Cloop Optimize Wizard” Dialog

Menubar list

This section explains the menubar of clooprofiler.

- [[File]] menu

[[File]] In the menu (Figure 11), can open the dialog that specifies the block reference situation data file and the cloop image file.

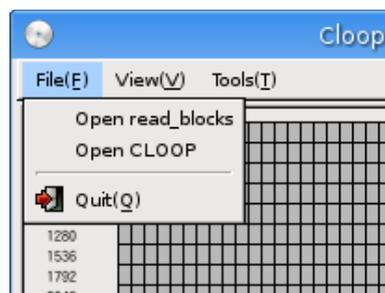


Figure 11: [[File]] menu

- ◆ [[Open read_blocks]]

It opens the dialog that specifies the block reference situation data file. Before selecting 'Open CLOOP', it is necessary to specify the block reference situation data file corresponding to the cloop image file.

The block reference situation data file that is selectable from this menu is only a block reference situation data file corresponding to the cloop image file.
- ◆ [[Open CLOOP]]

It opens the dialog that specifies the cloop image file. When the cloop image file is read, it comes to be able to perform selection in block display mode from menu bar [[View]] -> [[draw resource]]. Also, menubar [[Tools]] -> [[Accelerate Wizard]] becomes selectable, and ends up able to execute the cloop image optimization wizard dialog.

When you want to change the visualizing cloop image file, it is necessary to rearrange the block reference situation data file of the cloop image file that becomes the object.
- ◆ [[Quit]]

End this application.
- [[View]]menu

[[View]]The menu can adjust the display of the block reference situation.

 - ◆ [[draw resource]] container

Switch the block reference situation drawing mode (Figure 8).

 - [[read_blocks log]]

This mode is to render the block reference situation by utilizing the block ID obtained from the block reference situation data.
 - [[cloop header]]

This mode is to render the block reference situation by utilizing the block offset position obtained from the header information of cloop image file.
 - ◆ [[block color]] container

Customize the color of the drawing block. By this operation, the data block where time was required for reading or where it has been referred frequently can be distinguished at one view(Figure 12).

 - [[read time]]

This mode will separate the color of blocks according to the duration required for reading. The longer duration, the deeper color is applied.

- [[read count]]

This mode will separate the color of blocks according to reading frequency. The higher frequency, the deeper color is applied.

- [[time/count]]

The rendering block will be divided into tow areas. The Upper triangular part displays the [[read time]]mode and the lower triangular part displays the [[read count]].

- [[count/time]]

The rendering block will be divided into tow areas. The Upper triangular part displays the [[read count]]mode and the lower triangular part displays the [[read time]].

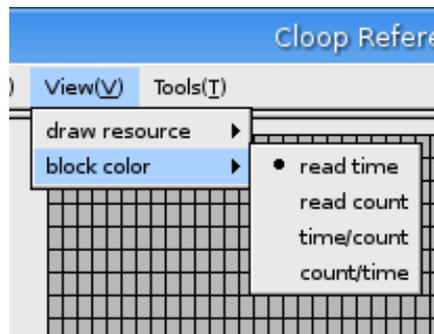


Figure 12: [[block color]] container

- [[Tools]] menu

[[Tools]] menu can start-up cloop image optimization wizard and, lookahead specification block list which are utilized for Live CD boot acceleration(Figure 13). These wizards can be selected only when the cloop image file is specified.

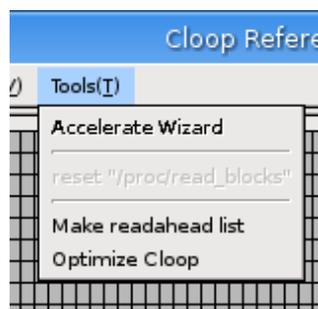


Figure 13: [[Tools]] menu

- ◆ [[Accelerate Wizard]]

It can start-up the wizard that can optimize the block arrangement of the open cloop image file and will create the lookahead specification block list based on the block reference data file.
- ◆ [[Make readahead list]]

Start-up the lookahead specification block list creating wizard.
- ◆ [[Optimize Cloop]]

Start-up the cloop image optimization wizard.
- [[Help]]menu (Figure 14)
 - ◆ [[Info]]

Display information of this program.



Figure 14: [[Help]] menu

Interface list

This section explains the function of each interface. Refer to Figure 6 for the arrangement of the main button.

- |(>)|(|(||)|) button

Play the block reference situation in animation when the block reference situation data file is being opened. It can pause and replay the animation while it's been played.
- |(■)| button

Stop the animation, and return the time counter to an initial value.
- |(>>)| button

Fast-forward the animation.
- |(<<)| button

Rewind the animation.

- Block drawing area

It is an area where animations of the block reference situation are played. While playing animation, the block being read will be indicated in blue frame. The details of the drawing block (i. e. referred block) will be popped up by focussing the mouse pointer on the drawing block area and click the right mouse button.

- Progress slider of time

The slider displayed right under the block drawing area shows the elapsed time of the boot progress. Elapsed time can be adjusted by dragging the toggle.

- |(Block/Cell)| select button

It can adjust the graphic mode and amount of data in one drawing block. The amount of data for one drawing block will be the number of blocks when the graphic mode is shown from block ID and when it is shown from offset position it will be the size of data (2^x). After changing the amount of data, it will be reflected in the block drawing area by pressing |(>) |(| |)) button, |(■) | button, |(change) | Button.

- [[Draw Color]] Pull-down list (Figure 15)

This function similar to [[View]] → [[draw color]]

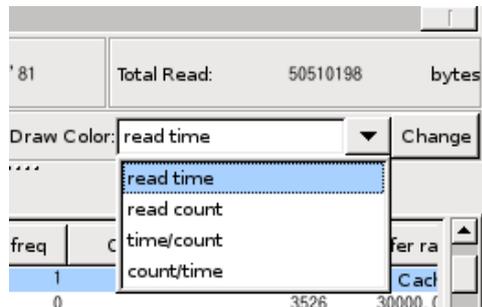


Figure 15: [[Draw Color]] Pull-down list

- [[Statistical Analysis]] list

A statistical analysis result is displayed from the block reference situation data file.

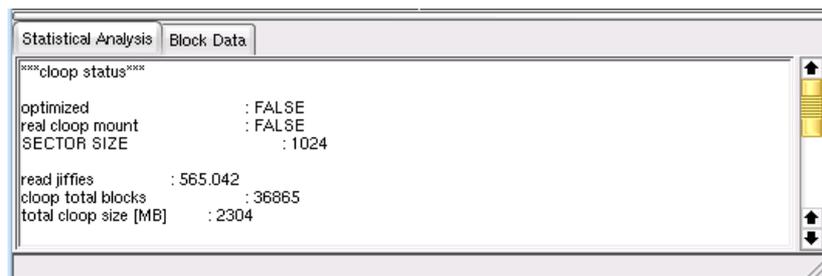


Figure 16: [[Statistical Analysis]] list

- [[Block Data]] list

The block reference situation will be displayed in refereced order. The drawing block of the block drawing area will be emphasized by clicking the element of the list.

Read No.	Block No.	Ref time(s)	Read freq	Offset	Size(bytes)	Transfer rate
0	0	0.022	0	650660269	5435	225.22
1	32	0.000	0	650681382	26021	260210.00
2	34816	0.013	0	620300996	6584	465.11
3	4096	0.005	0	653163777	5132	1063.83
4	20480	0.003	0	660699066	8774	2352.94
5	12288	0.003	0	652491490	9691	2727.27
6	21528	0.005	0	70222416	5872	1174.40

Figure 17: [[Block Data]] list

[[Tools]] Wizard list started from menu

This section explains will be about each wizard that starts from the 'Tools' menu.

- [[Accelerate Wizard]]

Display the Cloop Optimize Wizard dialog (Figure 10) by selecting [[Accelerate Wizard]]. The drawing block of the block drawing area will be emphasized by clicking the element of the list.

- ◆ read_blocks

Display pass the block reference situation data file which is currently open.

- ◆ cloop image

Display pass the cloop image file that is currently open.

- ◆ Block List

Redirect the output of the lookahead specification block list.

- ◆ burst size

Specify the reading size of the data in kilobyte for each request at the time of lookahead data block.

- ◆ Optimized Cloop

Redirect the output of the optimized cloop image file.

- **[[Make readahead list]]**

Display the Create Block List Wizard dialog (Figure 18) by selecting **[[Make readahead list]]**. As for Create Block List Wizard, it creates the lookahead block lists from an arbitrary block reference situation data file. It behaves the same way as LCAT utility command `rbk2bl`.

- ◆ **Input File**

Specify the input block reference situation data file.

- ◆ **Output File**

Specify the output lookahead specification block list.

- ◆ **burst block**

Specify the size of the data in kilobyte for each request at the time of lookahead data block

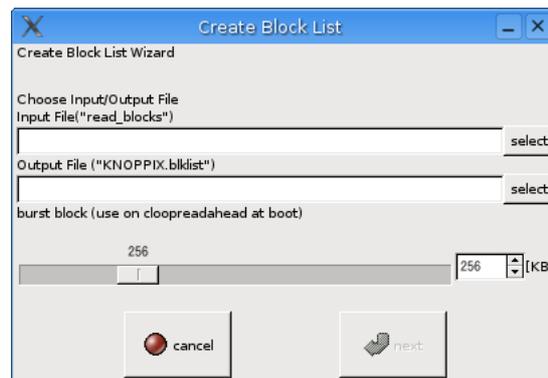


Figure 18: Make readahead list dialog

- **[[Optimize Cloop]]**

Display the Optimized Cloop Image Wizard dialog (Figure 19) by selecting **[[Optimize Cloop]]**. Optimized Cloop Image Wizard optimizes the block arrangement of an arbitrary cloop image file based on an arbitrary block reference situation data file. The interface of an Optimized Cloop Image Wizard dialog is summarized into below.

- ◆ **Input Cloop**

Specify the cloop image file of the target for optimization.

- ◆ **Input Block List**

Specify the block reference situation data file of the cloop image file of the target for optimization.

- ◆ **Output Cloop**

Specify the optimized cloop image file output.

- ◆ **butst block**

Specify the size of the data read by the request once in the data block lookahead in each kilobyte.

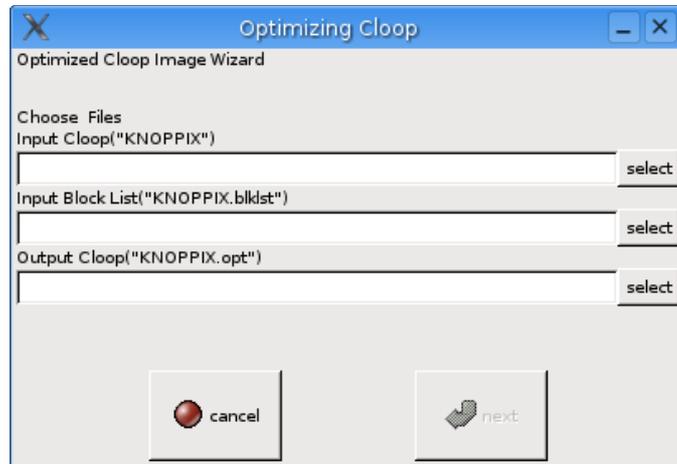


Figure 19: Optimized Cloop Image Wizard dialog

4.2.2. Image optimization tool(cloopoptimizer)

Outline

Cloopoptimizer creates the file of the cloop image of the optimization format out of default cloop image file. Optimization cloop image file made with cloopoptimizer can be read only from the cloop driver for the optimization image included in LCAT.

Cloopoptimizer can be used only in the image created in the format after the cloop version 2 onward. Which means the former version of the cloop image file cannot be used.

Usage

Cloopoptimizer takes the following arguments.

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

The path of cloop to optimize is specified as <original cloop file>. The block reference situation data file at the time of boot is specified as <boot block reference data>. The block reference situation data file of application is specified as <application reference data>. <application reference data> is omissible.

The result of cloopoptimizer is output to a STDOUT, and the progress report is output to the STDERR. Figure 20 shows the example of executing cloopoptimizer. In the execution example, cloopoptimizer is executed on KNOPPIX that started in the profiling mode (Refer to “Acquisition of the block reference situation data by the start in the profiling mode 5.2”).

```
$ 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
:
```

Figure 20: example of cloopoptimizer execute

4.2.3. Block reading program(cloopreadahead, rblk2bl, appblk2bl)

- cloopreadahead

Outline

Cloopreadahead is a program that reads the data block filing optimization cloop image based on the lookahead specification block list. The data of the read block will be stored in page cache that is managed by the virtual memory subsystem of kernel. The I/O waiting time of the process generated in the boot sequence can be shortened by calling up cloopreadahead in initrd during the the system startup of live CD.

This program is only for optimization cloop image file.

Usage

Cloopreadahead takes the following arguments.

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

The path to an optimization cloop image file is specified as <optimized cloop file>. A prediction specification block list is specified as <block list file>.

Cloopreadahead outputs the block number of the cloop block that has been read to a STDOUT.

- rblk2bl

Outline

rblk2bl is the program which changes the block reference situation data file acquired by the cloop driver corresponding to an optimization image into the prediction specification block list which can be used by cloopreadahead.

Usage

rblk2bl takes the following arguments.

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

A block reference situation data file is specified as <block reference data>. The size of a reading unit is specified as <burst size>. <burst size> becomes a particle size at the time of reading a block by cloopreadahead. It seems to work negatively in terms of speed in system boot when the value of <burst size> is too large or too small. Default <burst size> is 262144B (256KB) .

rblk2bl outputs the lookahead specification block list to the STDOUT. Figure 21 shows the example of the lookahead specification block list that rblk2bl outputs.

```
# 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
:
```

Figure 21:lookahead specification block list example

- appblk2bl

Outline

The program that converts the block reference situation data file that has been obtained from the cloop driver supporting optimization image into the lookahead specification block list to which it refers with cloopreadahead before the application starts.

Usage

Appblk2bl takes the following arguments.

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

The block reference situation data file at the time of a system startup is specified as <boot block reference data>. The size of a reading unit is specified as <burst size>. The block reference situation data file at the time of application starting is specified as <application block reference data>. The block reference situation data file at the time of two or more application startings can be specified.

appblk2bl outputs the lookahead specification block list to the STDOUT. The output format of appblk2bl is the same as rblk2bl.

4.3. Driver for optimization image(cloop.ko)

Outline

As for optimization cloop image file, a part of file format is different from the default cloop version 2 image file because of the maintenance of the location information on the cloop file system of data block. The cloop driver corresponding to this format is a driver for the optimization image. Moreover, the function to maintain the block reference situation data needed to relocate the data block is added.

It comes to be able to file a start in the profiling mode described later and an existing cloop image by building this driver into initrd used when live CD starts, and to treat the cloop image file that has been optimized as a root file system device.

Usage

The driver for this optimization image Kernel modules and is offered. Take the following module parameters.

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

The path of the file set up as a device image at the time of modular loading is specified as <cloop image file>. The maximum holding block reference situation data is specified as <chkblks>. When <chkblks> is zero or more, the driver corresponding to an optimization image operates in profiling mode. procs is used for the interface of a block reference situation.

Cloop image file “/cdrom/KNOPPIX/KNOPPIX” is specified after LCAT from the source is compiled, and the following execution example is loaded by the setting that maintains the block reference situation data for 10,000 blocks.

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

When module parameter “chkblks” is specified, the directory of cloop is made under the control of “/proc”. Bring the file under the control of this directory together as follows.

- /proc/cloop/read_blocks

The block reference situation can be acquired by reading this file. Figure 22 shows the example of the block reference situation. This file is only for reading.

```

cloop device : 0
total blocks : 36865
file format  : optimized
now jiffies  : 35549.0288
cloop kernel thread mode.

```

```

-----
call cloop_request      : 135551 times
call load_buffer()     : 485448 times
call read_from_file()  : 42441 times
read_time_max          : 0.0039 ms
read_time_min          : 0.0000 ms
read_time_sum          : 12.0818 ms
read length sum        : 820170874 bytes
extract_time_max       : 0.0009 ms
extract_time_min       : 0.0000 ms
extract_time_sum       : 21.0215 ms
extract length sum     : 2781413376 bytes
-----

```

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      -> read ID
# blknum  -> block ID
# size    -> block size
# access  -> jiffies when access to the block
# read    -> time that hung to reading
# extract -> time that hung to extract

```

Figure 22: Example of block reference situation/`/proc/cloop/read_blocks`

- `/proc/cloop/reset_read_blocks`

Clear the maintained block reference situation by writing it in this file. Write this file only in the root authority.

- `/proc/cloop/reset_profile`

Clear statistical data of the maintained block reference situation by writing it in this file. Write this file only in the root authority. In reset of block reference situation data “`/proc/cloop/read_blocks`”, execute the following commands by the root authority.

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

4.4. Patch files for KNOPPIX various scripts

Outline

In order to perform starting Acceleration of Live CD, a cloop image is optimized and optimization of a starting sequence including circumference detection etc. is also performed. All of each script file executed by starting the system are stored in the file system in the cloop image for KNOPPIX excluding linuxrc. With this, the process of making the cloop image at the trial to acceleration is generated, and the work efficiency decreases. Then, processing that replaces each script file that executes the processing of the start sequence is added to an initial stage of the start of the system. Moreover, the command that is called from each script also has changed part.

Munged script list(order of start)

Here, it explains the munged file speeding up knoppix-4.02 Japanese version CD the start.

- linuxrc

This is a script that it is in initrd that is a temporary root file system when starting, and kernel starts most first.

Start the block reading program when the lookahead specification block list exists in the CD-ROM device after detecting the cloop image file. Moreover, analyze the character string of the boot option to acquire the block reference situation data of cloop, and change cloop driver's module parameter. Moreover, install various scripts that optimize the start sequence when you construct the root file system.

- inittab

This manages the orchis level and is put on “/etc/inittab” in the file that sets init that becomes a parent process in all the processes.

There was processing for which it waited for two seconds immediately before the orchis level's becoming five and starting X server that was the environment of GUI in an existing configuration file. The processing was removed this time.

- knoppix-autoconfig

This is put on “/etc/init.d” in the script file that starts X server to lie at the bottom of the GUI environment of Linux.

Mounting for which it waited with sleep simply in an existing mounting until the processing that became a critical path ended. This time, it mounted like judging the processing had ended positively. Moreover, it was started it did not start one by one but bringing the process

of service together. Hwsetup (following description) newly prepared was executed in the peripherals detection. Moreover, the command that made the configuration file of X server was made to execute mkxf86config (following description) newly prepared, too.

- xsession

This is put on “/etc/init.d” in the script file that starts X server to lie at the bottom of the GUI environment of Linux.

This also stopped waiting s processing with sleep in the part of the processing that became a critical path as well as above-mentioned knoppix-autoconfig, and did the start confirmation in the process positively.

- 45xsession

This is put on “/etc/X11/Xsession.d/45xsession” in the script to start desktop environments after the start of X server is completed.

It was started bringing the one that each one of the process had been one by one executed together within the range without the influence. Moreover, the script generation processing to read the data block to which it referred when the application started being executed after a desktop start was completed ahead was added.

Replaced command list

Here, it is called from the munged script file speeding up knoppix-4.02 Japanese version CD the start, and it explains the command newly prepared.

- hwsetup

It has been understood that hwsetup called from an existing knoppix-autoconfig script as part of the peripherals detection often requires time to detect the cereal bus when executing it. Then, hwsetup to be able to separate each bus detection processing was prepared. Changed knoppix-autoconfig does the detection processing of the cereal bus in another process corresponding to new hwsetup.

- mkxf86config

mkxf86cofnig called from an existing knoppix-autoconfig script for the configuration file generation of X server occasionally required time for the monitor's detection. Because the processing had been executed twice in existing mkxf86config, the execution part of the second times was changed to recycle the processing result of the first time.

5. Boot acceleration of KNOPPIX

In this section, the actual procedure for accelerating boot sequence using live CD that is KNOPPIX will be explained. The main operation for the boot acceleration of KNOPPIX is to optimize the cloop image file stored in CD-ROM.

In order to optimize cloop image file, it needs to know how data blocks are read in cloop. As it mentioned previously, the block reference situation data which is the information of reading condition will be acquired by actually starting the system. And optimized cloop driver that has profiling function has to be implemented in KNOPPIX to acquire the block reference situation data.

Next, by activating the KNOPPIX in profiling mode, it acquires the block reference situation data in cloop image.

Finally, optimize the block arrangement of the cloop image according to block reference situation data, and commute it into ISO 9660 image file. And boot acceleration will be applied into KNOPPIX.

*** Attention ***

- Needs LCAT to operate this chapter. Refer to Chapter 3 for the method of installing LCAT.
- All the operations in this chapter have to be performed by root authority since there is fear to destroy the development environment by the operation mistake. Please read through at first and understand this chapter procedure before you perform the start acceleration.
- Confirm the hard disk space since it treats comparatively large file. When the applied media is CD, it is necessary to have about 10GB of an empty area.

5.1. The cloop driver's for optimization image introduction

This section covers the requirement for embedding the cloop driver supporting optimized image into KNOPPIX for acceleration.

Firstly, copy all files in the ISO image file of KNOPPIX into work directory. The following exmample shows when ISO image file is "knoppix_4.0.2cd_j.iso".

```
# 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
```

By the operation above, all files in the ISO image file will be copied under directory "./MASTER".

Secondly, replace the 'minirt.gz' image file in route file system, which is referred by the cloop dirver during system boot.

```
# 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
```

By the operation above, loop back image 'miniroot' will be mounted "./tmp_mount".

Next replace the excisting cloop driver with the cloop driver supporting optimized image. If binary installing, then copy "/lib/modules/2.6.12/kernel/drivers/block/cloop.ko" to under "./tmp_mount/modules". If compiling from source, copy under the compiled directory, "lcat_1.0/cloop-2.02-opt/cloop.ko", to under "./tmp_mount/modules".

```
# mv ./tmp_mount/modules/cloop.ko \
    ./tmp_mount/modules/cloop.ko.org
# cp <dest>/cloop.ko ./tmp_mount/modules
```

Then, modify linuxrc in order to be able to use profiling mode of the cloop driver supporting optimized image. Specifically, add argument processes to the part of process where the cloop driver is loaded in insmod. See list1.

Line number in list1 shows linuxrc of knoppix_4.0.2cd_jp.iso.

```
# 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 \
    "$1/$KNOPPIX_DIR/$KNOPPIX_NAME"; then
---- Here: start comment
399   # # DEBUG
400   # # echo "6" > /proc/sys/kernel/printk
401   # $INSMOD -f /modules/cloop.ko file="$1/$KNOPPIX_DIR/$KNOPPIX_NAME"
---- Here : end comment
---- Here : Addition
    # 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 $i; \
          [ -n "$chkblk" ] && CHKBLK="$chkblk" ;; esac
        case "$i" in *noclpthread*|*NOCLPTHREAD*) THREAD="" ;; esac
    done
    echo ""
    $INSMOD -f /modules/cloop.ko chkblks=$CHKBLK \
      "$THREAD" file="$1/$KNOPPIX_DIR/$KNOPPIX_NAME"
    echo "0" > /proc/sys/kernel/printk
---- Here : Addition
402   mountit /dev/cloop /KNOPPIX "-o ro" || FOUND_KNOPPIX=""
403   # Allow multi-image KNOPPIX mounts
404   if [ -n "$FOUND_KNOPPIX" -a -x "$DYNLOADER" \
    -a -x /KNOPPIX/sbin/losetup ]; then
---- Here : start comment
405   #   echo ""
---- Here : end comment
406     echo -n "${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

```

list1 : Correction of linuxrc(addition of cloop driver's for optimization image profiling mode)

Then, modify linuxrc in order to be able to use profiling mode of the cloop driver supporting optimized image. Specifically, add argument processes to the part of process where the cloop driver is loaded in ismod. See list1. Line number in list1 shows linuxrc of knoppix_4.0.2cd_jp.iso.

```

496 copy_to()
497 {
    |
573 }
574
---- Here : start move
577 COPYTO=""
578 BOOTFROM=""
579 DO_REMOUNT=""
580 REAL_TARGET=""
581 UNIONFS=""
582
583 case "$CMDLINE" in *toram*) DO_REMOUNT="yes";COPYTO="ram"; ;; esac
584 case "$CMDLINE" in *tohd=*) DO_REMOUNT="yes";COPYTO="hd"; ;; esac
585 case "$CMDLINE" in *bootfrom=*) DO_REMOUNT="yes"; \
    BOOTFROM="yes" ;; esac
586
---- Here : end move
---- Here : addition
#
# accelerated-knoppix : cloop blocks readahead
#
if test -z "$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 "$CLOOPREADAHEAD" -a -f \
            /cdrom/${KNOPPIX_DIR}/${KNOPPIX_NAME}.boot.lst ; then
            echo ""
            echo -n " ${GREEN}Reading cloop 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
---- Here : Addition
---- Here : start move
575 mount_knoppix /cdrom
576
---- Here : end move
587 # Remount later after copying/isoloading/driverloading?
588 # pre-test if everything succeeded
589 if test -n "$DO_REMOUNT" -a -n "$FOUND_KNOPPIX"

```

List 2: Correction of linuxrc(addition of processing that reads data block imaging cloop ahead)

Furthermore, add the process that can read-ahead a data block of the cloop image. See list2. In KNOPPIX4.0.2CD Japanese edition, the part(line 577 to 586) that process the boot options "toram", "tohd" and "bootfrom" will be executed at first, then the read-ahead process of the data block will be determined to activate based on the result. Moreover, the cloop image will be mounted while reading a data block(line 575-576).

directory when installing it from the deb package. "lcat_1.0/util/cloopreadahead" is copied onto the "./tmp_mount/accel" directory when compiling from the source.

```
# mkdir ./tmp_mount/accel
# cp <dest copy>/cloopreadahead ./tmp_mount/accel
```

After having finished all the work above, release the mount of loop back image 'miniroot', and put it as 'minirt.gz'

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

By the In KNOPPIX4.0.2CD Japanese edition, the part(line 577 to 586) that process the boot options "toram", "tohd" and "bootfrom" will be executed at first, then the read-ahead process of the data block will be determined to activate based on the result. Moreover, the cloop image will be mounted while reading a data block(line 575-576).job above, the file set of KNOPPIX for profile has been put under "./MASTER". Then, create ISO image file(knoppix_4.0.2cd_j_prof.iso) by gathering the file.

```
# 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
```

By the job above, ISO image file(knoppix_4.0.2cd_j_prof.iso) of KNOPPIX introducing the cloop driver supporting optimized image with profile functionality will be created under work directory.

In the following section, start the KNOPPIX introducing the cloop driver supporting optimized image on a real machine, and profile.

5.2. Acquisition of block reference situation data by start in profiling mode

This section shows the procedure to acquire block reference situation data using the KNOPPIX based on Section 5.1.

KNOPPIX can start from either a real machine or a virtual machine such as QEMU,etc. In such case,the following boot operation is specified to activate profiling mode.

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

Open the console after desktop has been activated and get the block reference situation data of at the period of system boot by the following operation.

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

The block reference situation data will be stored in "/tmp/boot.read_blocks" by this operation.

The application can be sped up if it is the high use application that is collected in KNOPPIX.

Here,it shows the example of accelerating OpenOffice.org.

Reset the current block reference situation by following operation before it starts to store the block reference situation data during application boot.

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

Launch Open Office.org and wait until it is completely activated. Get the block reference situation data by filling out following command syntax in the console.

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

The block reference situation data is now stored in "/tmp/appli.read_blocks". It should be noted that all the necessary applications have to be activated before getting the block reference situation data.

Put the two files,(i. e. boot.read_blocks and appli.read_blocks) under "./MASTER" directoy which was shown in Section5.1 by using USB memory or by network.

5.3. Analysis of block reference situation

This section covers the procedure to analyze block reference situation by using 'clooprofiler' for each block reference situation data file, that has been collected during system boot and application boot based on section 5.2.

Since the transferred files (boot.read_blocks, appli.read_blocks) are text files, they can be seen by most editors. Seeking frequency of CD-ROM drive can be seen by using clooprofiler which is a visualization tool for block reference situation.

Activate clooprofiler as follows:

```
# clooprofiler
```

Then, select the transferred block reference situation data file by using file dialog selected in menu bar [[File]] -> [[Openread_blocks]]. After having read data file, display animation of cloop block reference in real time by pressing [[>]] button. When you change block resolution in the displayed area, use [[block/Cell]] spin button.

Usually, an animation of block reference situation are drawn by block ID. In order to draw the animation in real offset position in the cloop image file, the selected cloop image file when collecting block reference situation data is needed.

When selecting a cloop image file, select the cloop image file by using file dialog selected in menu bar [[File]] -> [[OpenCLOOP]]. The cloop image file is under a work directory "/.MASTER/KNOPPIX/KNOPPIX". The mode will be changed to drawing mode in cloop image file by selecting in menu bar as [[View]] -> [[draw resource]] -> [[cloop header]].

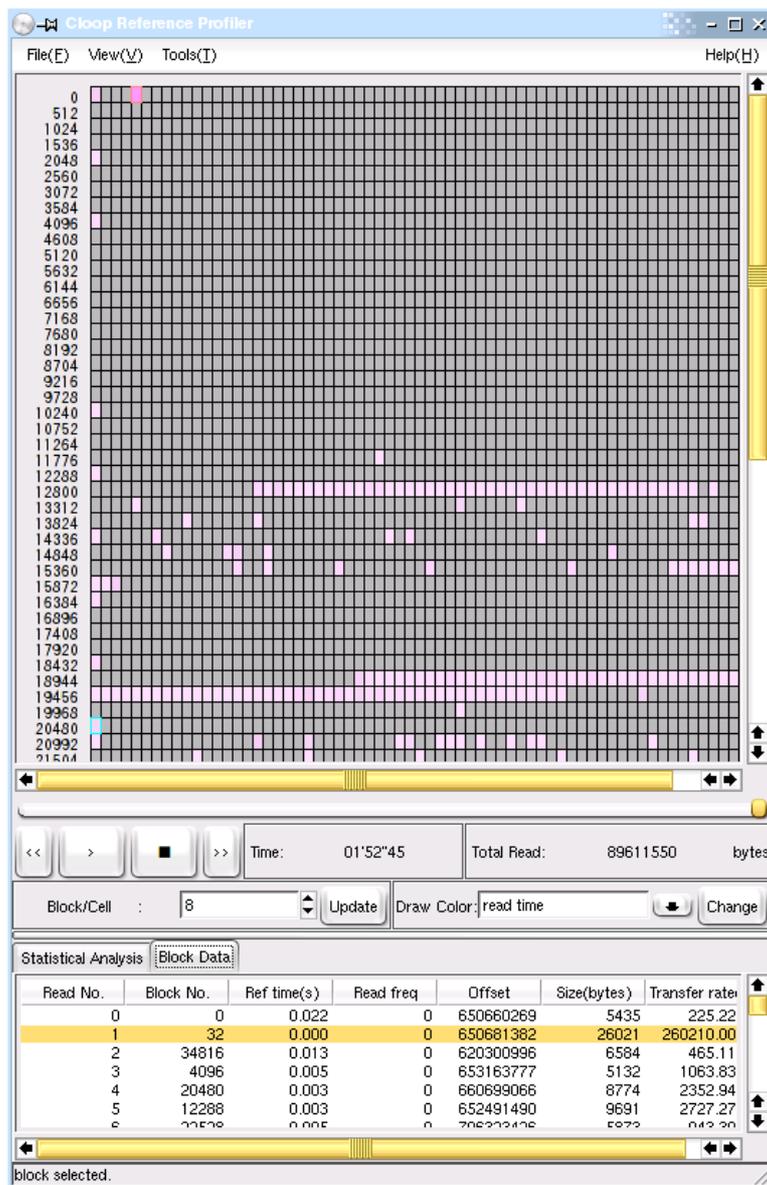


Figure 23: Making of block reference situation by clooprofiler visible

5.4. Arrangement of block of data of existing cloop image file optimization

This section shows how to optimize block location during cloop image according to the block reference situation files that were taken during system start and application start.

In order to optimize cloop image block position, there are two ways to perform it: One is to use utility command of LCAT, cloopoptimizer. The other is to use clooprofiler GUI. Only the former is shown in this section. As for the latter, refer to section4.2.1.

By the operation below, make optimized cloop image file, "./MASTER/KNOPPIX/KNOPPIX" by executing "cloopoptimizer" command after the existing cloop image file name has been changed another name(KNOPPIX.normal).

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

By the operation above, the optimized cloop image file will be put under "./MASTER".

5.5. Creating read-ahead specification block list during system boot.

This section covers the procedure to make data block list foreseen in cloop from the block reference situation data, in starting system, collected by the procedure of section5.2.

In order to make read-ahead block list, there are two ways to perform it: One is to use utility command of LCAT, rblk2bl. The other is to use clooprofiler GUI. Only the former is shown in this section. As for the latter, refer to section4.2.1.

By the operation below a data block list and read-ahead specification block list in cloop image which is foreseen from the block reference situation data file in during starting system boot "./MASTER/KNOPPIX/KNOPPIX.boot.lst" are created.

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

By the operation above, the read-ahead specification block list during system boot will be put under "./MASTER".

5.6. Making of iso image file that stores optimization cloop image file

By the procedure of section 5.1, section 5.4 and section 5.5, the cloop driver supporting optimized image, the optimized cloop image file and read-ahead block list during system boot are put under "./MASTER".

Then, get them together as ISO image file (knoppix_4.0.2cd_j_AC.iso) by the following operations.

```
# 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
```

By the operation above, the KNOPPIX ISO image file "./knoppix_4.0.2cd_j_AC.iso" will be completed

5.7. acceleration of further start

This chapter mainly explained the optimization of the cloop image. When accelerating live CD start-up, there are some ways such as environmental device detection process is to be asynchronization. Note that a way is different for each sort of or version of CD.

For the application example applying KNOPPIX 4.0.2CD Japanese version, refer to the LCAT archive file under "lcat_1.0/KNOPPIX/".

6. Address of thanks

LCAT was developed by “The open source software use substrate maintenance business 'Development of the speed improvement driver of CD/DVD start Linux’” project of the Information-technology Promotion Agency,Japan(IPA) fiscal year 2005. We got a lot support from IPA about the plan of this project and the progress of development.We express gratitude for thick support by them.

This development got a chance to introduce this project in The kernel reading party that opened by Yokohama Linux Users Group(YLUG) sponsored , KNOPPIX-BOF in open source conference at the Kansai open forum , and more.We express my gratitude also to the people who gave the talked place and the participating people, and We express my gratitude to the people to have commented to this project by the bulletin board of the people who discussed it, Blog, and the Internet for offering useful information with KOEDO LUG-ML including YLUG-ML and KNOPPIX-ML.

Finally, gratitude is shown to Mr. Klaus Knopper who developed KNOPPIX, Mr. Linus Torvalds who is the real parent of Linux, people in the world where Linux has been raised. and all people concerning an open source.

Thank you very much.

Team Accelerated KNOPPIX

Advanced Technology Research Dept.

Alpha System, Inc.

mailto:knoppix_at_alpha.co.jp



http://www.alpha.co.jp/ac-knoppix/index_en.html