



第 20 回 東京エリア Debian 勉強会 事前資料

Debian 勉強会会場係 上川純一*

2006 年 9 月 16 日

* Debian Project Official Developer

目次

1	Introduction To Debian 勉強会	2
1.1	講師紹介	2
1.2	事前課題紹介	2
2	Debian Weekly News trivia quiz	3
2.1	2006 年 XX 号	3
3	最近の Debian 関連のミーティング報告	4
3.1	東京エリア Debian 勉強会 19 回目報告	4
4	翻訳へのさそい	5
4.1	po 関連	5
4.2	Debian JP WWW メーリングリスト	5
4.3	参考文献	5
5	apt/dpkg のプロファイリング	6
5.1	oprofie のインストールと設定方法	6
5.2	oprofile が自分の利用している CPU をサポートしていない場合	6
5.3	デバッグシンボルを収集する : dpkg と apt をコンパイルしなおす	7
5.4	テストケースの作成	8
5.5	最適化の必要な部分の解析	8
5.6	最適化例	8
5.7	結果の検証	8
5.8	回帰テストの貢献	8
5.9	修正をフィードバック	8
5.10	参考文献	8
6	次回	9

1 Introduction To Debian 勉強会

上川純一

今月の Debian 勉強会へようこそ。これから Debian のあやしい世界に入るという方も、すでにどっぴりとつかっているという方も、月に一回 Debian について語りませんか？

目的として下記の二つを考えています。

- メールではよみとれない、もしくはよみとってられないような情報を情報共有する場をつくる
- まとまっていない Debian を利用する際の情報をまとめて、ある程度の塊として出してみる

また、東京には Linux の勉強会はたくさんありますので、Debian に限定した勉強会にします。Linux の基本的な利用方法などが知りたい方は、他でがんばってください。Debian の勉強会ということで究極的には参加者全員が Debian Package をがりがりと作りながらスーパーハッカーになれるような姿を妄想しています。

Debian をこれからどうするという能動的な展開への土台としての空間を提供し、情報の共有をしたい、というのが目的です。今回は違うこと言ってるかもしれませんが、御容赦を。

1.1 講師紹介

- 上川純一 宴会の幹事です。

1.2 事前課題紹介

今回の事前課題は「XXX」というタイトルで 200-800 文字程度の文章を書いてください。というものでした。その課題に対して下記の内容を提出いただきました。

1.2.1

1.2.2 上川

2 Debian Weekly News trivia quiz

上川純一

ところで、Debian Weekly News (DWN) は読んでいますか？Debian 界隈でおきていることについて書いている Debian Weekly News. 毎回読んでいるといろいろと分かって来ますが、一人で読んでいても、解説が少ないので、意味がわからないところもあるかも知れません。みんなで DWN を読んでみましょう。

漫然と読むだけではおもしろくないので、DWN の記事から出題した以下の質問にこたえてみてください。後で内容は解説します。

2.1 2006 年 XX 号

<http://www.debian.org/News/weekly/2006/XX/> にある 5 月 XX 日版です。

問題 1.

- A
- B
- C

問題 2.

- A
- B
- C

問題 3.

- A
- B
- C

問題 4.

- A
- B
- C

3 最近の Debian 関連のミーティング報告

上川純一

3.1 東京エリア Debian 勉強会 19 回目報告

4 翻訳へのさそい

えとーさん

4.1 po 関連

現状の ML についての Web は、<http://www.debian.or.jp/MailingList.html> です。

この中で翻訳関連を行なっているのは Debian JP Documentation メーリングリストです。

`debian-doc-ctl@debian.or.jp` に `fml` の方式で subscribe してください。過去記事は <http://lists.debian.or.jp/debian-doc/> にアクセスすると見ることができますので、ご参考にどうぞ。こちらは、`man`、`debconf-po`、`po`、および 付属ドキュメントなどの翻訳を行なっています。

Debian の `po` の各国のランキングです。 <http://www.debian.org/international/l10n/po/rank>

Debian の `debconf-po` の各国のランキングです。 <http://www.debian.org/international/l10n/po-debconf/rank>

ちなみに `debconf-po` についての国内での作業は作業がかぶらないように <http://kmuto.jp/debian/po-trans/> を使いながらやるとよいかと思われます。

4.2 Debian JP WWW メーリングリスト

`debian-www-ctl@debian.or.jp` に `fml` の方式で subscribe してください。過去記事は <http://lists.debian.or.jp/debian-www/> にアクセスすると見ることができますので、ご参考にどうぞ。

こちらは、主に `debian.org` の web サイトの翻訳および、`debian.or.jp` の web サイトについて活動しています。

以下私見ですなにかの参考になりましたら。偉そうかもしれませんが、ご容赦を、、、

おそらく、`debian-doc` などで活動を行なうと `ubuntu` のほうにも当然のように波及しますので、こちらの作業を行なうのもよいかなと思いました。

`ddtp` というプロジェクトで以前パッケージディスクリプション部分 (本日もう昨日か、見せていただいたアプリケーション一覧の説明部分とか?) を翻訳しているプロジェクトがありましたが、一旦停止しています。近々復帰しますので、DWN (Debian Weekly News) などを注視していただくとそのうちなんか出てくるかもです。

Debian プロジェクトはあなたの参加を心待ちにしております。でわでわ

4.3 参考文献

- ドキュメント翻訳手順 <http://kmuto.jp/d/index.cgi/debian/debian-doc-procedure.htm>

5 apt/dpkg のプロファイリング

上川

apt や dpkg のどの部分が一番遅いのか、実際にプロファイリングしてみます。この例をケーススタディーとして一般的にどういう作業をすればパフォーマンスチューニングができるのか、をあきらかにしてみましょう。

5.1 oprofile のインストールと設定方法

Debian のデフォルトのカーネルは oprofile をサポートしています^{*1}。もし、自分でコンパイルしていたりして oprofile サポートを追加していない場合は、カーネルを oprofile サポート付きでコンパイルしなおします。オプションは CONFIG_OPROFILE です。メニューでは

```
Instrumentation support : Profiling Support : Oprofile system profiling (experimental)*2
```

にあります。

カーネルがサポートしている場合、oprofile を利用するのに追加で必要なのは oprofile パッケージです。apt-get install oprofile でインストールしましょう。

5.2 oprofile が自分の利用している CPU をサポートしていない場合

残念ながら 7 月現在時点で、Intel core duo CPU 上では oprofile が動作しません。oprofile は認識できていない場合、cpu_type 変数が unset という値になります。カーネル側は cpu_type として i386/core を出力しているので、この時点でどうやらカーネル側のサポートは追加されているらしいということがわかります。

```
$ sudo opcontrol --init
cpu_type 'unset' is not valid
$ opcontrol --list-events
Unable to open cpu_type file for reading
Make sure you have done opcontrol --init
cpu_type 'unset' is not valid
$ cat /dev/oprofile/cpu_type
i386/core
$ uname -a
Linux coreduo 2.6.18-rc1dancer #2 SMP Sun Jul 9 09:57:01 JST 2006 i686 GNU/Linux
```

プロファイルを取得するという目的を考えると手段としてはいくつか考えられます。

- プロファイルの仕組みはあまりかわらないだろうと見込み、arch/i386/oprofile/nmi_int.c の ppro_init を修正、piii とかに見せてしまう
- まじめに oprofile のユーザ空間アプリケーションを修正、core duo の仕様書を読み、対応を追加
- おそらくすでに修正されていることを見越して、oprofile の CVS レポジトリをみにいく
- 実験することが目的なのでサポートされている CPU のマシンを準備する

今回は oprofile の開発メーリングリストを見たところ、5 月の時点でだれかがパッチを書いているのを発見したので、それととりこみます。念のため、今後作業する人のために BTS にも登録しました。380462^{*3}

^{*1} i386, amd64 などのアーキテクチャ以外での利用は現時点では難しい可能性があるので確認してください。

^{*2} 2.6.18-rc1 現在

^{*3} <http://bugs.debian.org/380462>

確認してみると、どうやら動作してくれてそうです。ここで、当面重要なのは、CPU_CLK_UNHALTED でしょう。CPU サイクルがどの関数で消費されているのかということをトラッキングできます。まず CPU の処理がかたまっている部分を目指して、何か問題がないかを眺めてみて、何も問題なく、それなりに問題が追求できにくくなった後に、L2 キャッシュのイベントの発生度合とかを確認していけばよいでしょう。

```
$ sudo opcontrol --init
$ sudo opcontrol --list-events
oprofile: available events for CPU type "Core Solo / Duo"

See Intel Architecture Developer's Manual Volume 3, Appendix A and
Intel Architecture Optimization Reference Manual (730795-001)

CPU_CLK_UNHALTED: (counter: all)
  Unhalted clock cycles (min count: 6000)
  Unit masks (default 0x0)
  -----
  0x00: Unhalted core cycles
  0x01: Unhalted bus cycles
  0x02: Unhalted bus cycles of this core while the other core is halted
INST_RETIRED: (counter: all)
  number of instructions retired (min count: 6000)
L2_RQSTS: (counter: all)
  number of L2 requests (min count: 6000)
  Unit masks (default 0xf)
  -----
  0x08: (M)odified cache state
  0x04: (E)xclusive cache state
  0x02: (S)hared cache state
  0x01: (I)nvalid cache state
  0x0f: All cache states
  0x10: HW prefetched line only
  0x20: all prefetched line w/o regarding mask 0x10.
```

[省略]

5.3 デバッグシンボルを収集する：dpkg と apt をコンパイルしなおす

まず、デバッグ情報がすでにあるパッケージについては、インストールします。今回は、大きいものとして、libc6-dbg パッケージがあるので、それはインストールします。プロファイルの結果、上位に出現するなど、必要そうであれば、あとでデバッグ情報のあるバージョンを追加します。

今回プロファイル対象の dpkg と apt はデフォルトではデバッグ情報がありません、プロファイル出力を確認しやすいように、デバッグシンボルを追加してコンパイルしなおします。

```
$ debuild -e DEB_BUILD_OPTIONS=nostrip
```

その

後、インストールします。

まず、デバッグ用のバイナリが正常に作成できているか簡単に確認します。まず、apt-get update をループでまわしてみます。libapt-pkg のシンボルレベルで確認できているので、デバッグシンボルが存在しているということがわかります。

```
$ while true; do sudo apt-get update ; done
```

```
Every 10.0s: opcontrol --dump && oprofile -l 2>/dev/null | head -30 ;
opcontrol --reset                                     Mon Jul
31 08:28:16 2006
CPU: Core Solo / Duo, speed 2061.84 MHz (estimated)
Counted CPU_CLK_UNHALTED events (Unhalted clock cycles) with a unit mask of 0x00 (Unhalted core cycles) count 180000
samples %      image name      app name      symbol name
43117   42.3446  libapt-pkg-libc6.3-6.so.3.11.0 apt-get      SHA1Transform(unsigned int*, unsigned char const*)
23991   23.5612  libapt-pkg-libc6.3-6.so.3.11.0 apt-get      MD5Transform(unsigned int*, unsigned int const*)
6551    6.4337    vmlinux      vmlinux      sys_set_thread_area
5027    4.9370    libc-2.3.6.so apt-get      (no symbols)
3280    3.2212    vmlinux      vmlinux      cfq_dispatch_requests
3264    3.2055    libfb.so     Xorg         fbCopyAreaMMX
739     0.7258    gpgv        gpgv         (no symbols)
614     0.6030    libc-2.3.6.so http         (no symbols)
606     0.5951    ld-2.3.6.so http         (no symbols)
510     0.5009    libc-2.3.6.so file         (no symbols)
```

dpkg についてもプロファイリングしてみます。しかしここで問題が出ます。libc6 の dbg パッケージの情報をoprofile が処理できていないようです。strace で解析してみましたが、ファイルをひらくところまでは何かできているようで、深入りすると深そうです。

```
$ while true; do sudo dpkg -i ../dselect_1.13.22_i386.deb ; done
```



```

CPU: Core Solo / Duo, speed 2061.84 MHz (estimated)
Counted CPU_CLK_UNHALTED events (Unhalted clock cycles) with a unit mask of 0x00 (Unhalted core cycles) count 180000
samples  %      image name      app name      symbol name
44592    33.3795  libc-2.3.6.so      dpkg          (no symbols)
24191    18.1083  vmlinux            vmlinux      sys_set_thread_area
16423    12.2935  dpkg              dpkg         parsedb
4416     3.3056   dpkg              dpkg         findnamenode
2904     2.1738   dpkg              dpkg         findpackage
2216     1.6588   dpkg              dpkg         f_dependency
1654     1.2381   dpkg-deb          dpkg-deb     inflate_fast
1647     1.2329   dpkg              dpkg         .plt

```

5.4 テストケースの作成

テストケースを作成します。今回は chroot 内部で大量の apt-get update, apt-get install と apt-get remove をループで実行してベンチマークをとってみましょう。

5.5 最適化の必要な部分の解析

プロファイル結果を利用して、解析します。

5.6 最適化例

今回の結果で適用できる最適化を分析します。

5.7 結果の検証

さきほどのベンチマークを利用して、状況が改善していることを確認します。

5.8 回帰テストの貢献

できるようであれば、今回の修正が二度と必要ないように、パフォーマンスのデグレードがすぐに検出しやすいよう、再現しやすいテストケースを追加してみるのもよいでしょう。

5.9 修正をフィードバック

パフォーマンスの改善のための修正をフィードバックします。Debian の場合、BTS にパッチを登録します。該当するバグ番号は下記です。

- XXX
- YYY

5.10 参考文献

- rpm のプロファイリング <https://www.redhat.com/magazine/012oct05/features/oprofile/>



未定です。内容は本日決定予定です。
参加者募集はまた後程。



Debian 勉強会資料

2006 年 9 月 16 日 初版第 1 刷発行

東京エリア Debian 勉強会（編集・印刷・発行）
