Let's shrink “bloated Debian repository”

Hideki Yamane
(Debian Project: Debian Developer)
<henrich @ debian.org/or.jp>
http://wiki.debian.org/HidekiYamane
Today's Agenda

- How large is Debian Repository
- One day, I found a solution...
- Is it really effective?
- Problem on slower Arch
- How much can we shrink it?
Debian supports...

- Many many packages
- Many CPU architectures
- Some kernels
How large is Debian Repository?

- Arch: source, all, amd64, armel, armhf, hurd-i386, i386, ia64, kfreebsd-amd64, kfree-bsd-i386, mips, mipsel, powerpc, s390, s390x, sparc
How large is Debian Repository?

- Arch: source 52GB, all 57GB, amd64 53GB, armel 38GB, armhf 26GB, hurd-i386 14GB, i386 50GB, ia64 42GB, kfreebsd-amd64 37GB, kfreebsd-i386 36GB, mips 35GB, mipsel 34GB, powerpc 42GB, s390 36GB, s390x 24GB, sparc 39GB...

- Total?  

(http://www.debian.org/mirror/size)
How large is Debian Repository?

- Arch: source 52GB, all 57GB, amd64 53GB, armel 38GB, armhf 26GB, hurd-i386 14GB, i386 50GB, ia64 42GB, kfreebsd-amd64 37GB, kfreebsd-i386 36GB, mips 35GB, mipsel 34GB, powerpc 42GB, s390 36GB, s390x 24GB, sparc 39GB...

- Total: 615GB!!

(http://www.debian.org/mirror/size)
How can we improve this?
Can we shrink this?

Yes, in some ways...

Drop support architectures
Delete packages from archive
Can we shrink this?

However, we don't want these solutions

Drop support architectures
Delete packages from archive
Use XZ!

Default compression is gzip. xz can reduce file size.
Use XZ!

ex)
fonts-horai-umefont (I'm maintainer :-)

By gzip -9 : 43,664kb
By xz : 25,476kb
Use XZ!

ex)
fonts-horai-umefont  (I'm maintainer :-)

By gzip -9 : 43,664kb
By xz -9 : 25,476kb
→ 5,916kb
The archive software now accepts packages using xz for compression in addition to gzip and bzip2 for both source and binary packages.

Additionally please only use xz (or bzip2 for that matter) if your package really profits from its usage (for example, it provides a significant space saving). While those methods may compress better they often use more CPU time to do so and a very small decrease in package size is hardly worth the extra effort placed on slower systems. Think of both user systems and the Debian buildds which will waste more time – an especially bad problem on slower architectures.

("The archive now supports xz compression" by Ansgar Burchardt <anggar@debian.org> http://lists.debian.org/debian-devel-announce/2011/08/msg00001.html)
The archive software now accepts packages using xz for compression in addition to gzip and bzip2 for both source and binary packages.

(snip)

Additionally please only use xz (or bzip2 for that matter) if your package really profits from its usage (for example, it provides a significant space saving). While those methods may compress better they often use more CPU time to do so and a very small decrease in package size is hardly worth the extra effort placed on slower systems. Think of both user systems and the Debian buildds which will waste more time – an especially bad problem on slower architectures.
XZ on Slower arch is problem...

It'll eat most CPU time
XZ on **Slower** arch is problem...

Then...

if only on **Powerful** arch?
XZ on Powerful arch is NOT problem

assumption:

use XZ on Intel/AMD arch by default
Before XZ...

![Graph showing size (GB) for various architectures before XZ.]

- **all**: 57 GB
- **i386**: 52 GB
- **amd64**: 55 GB
- **hurd-i386**: 14 GB
- **ia64**: 42 GB
- **kfreebsd-amd64**: 39 GB
- **kfreebsd-i386**: 37 GB

The graph illustrates the size in GB for different architectures before using XZ compression.
After XZ!

The chart shows the size comparison before and after using XZ for various architectures:

- **All**: Before 57 GB, After 45 GB
- **i386**: Before 52 GB, After 32 GB
- **amd64**: Before 55 GB, After 34 GB
- **hurd-i386**: Before 14 GB, After 10 GB
- **ia64**: Before 42 GB, After 24 GB
- **kfreebsd-amd64**: Before 39 GB, After 24 GB
- **kfreebsd-i386**: Before 37 GB, After 23 GB

The chart indicates a significant reduction in size for most architectures after using XZ.
How much can we shrink it?

![Bar chart comparing before and after xz compression for different architectures.](chart.png)
How much can we shrink it?

<table>
<thead>
<tr>
<th>architecture</th>
<th>before</th>
<th>after xz</th>
<th>difference</th>
<th>Reduction Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>57</td>
<td>???</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>i386</td>
<td>52</td>
<td>???</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>amd64</td>
<td>55</td>
<td>???</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>hurd-i386</td>
<td>14</td>
<td>???</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>ia64</td>
<td>42</td>
<td>???</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>kfreebsd-amd64</td>
<td>39</td>
<td>???</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td>kfreebsd-i386</td>
<td>37</td>
<td>???</td>
<td></td>
<td>---</td>
</tr>
<tr>
<td><strong>total</strong></td>
<td><strong>296</strong></td>
<td>???</td>
<td></td>
<td>---</td>
</tr>
</tbody>
</table>
How much can we shrink it?

<table>
<thead>
<tr>
<th>architecture</th>
<th>before</th>
<th>after xz</th>
<th>difference</th>
<th>Reduction Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>57</td>
<td>45</td>
<td>-12</td>
<td>21%</td>
</tr>
<tr>
<td>i386</td>
<td>52</td>
<td>32</td>
<td>-20</td>
<td>38%</td>
</tr>
<tr>
<td>amd64</td>
<td>55</td>
<td>34</td>
<td>-21</td>
<td>38%</td>
</tr>
<tr>
<td>hurd-i386</td>
<td>14</td>
<td>10</td>
<td>-4</td>
<td>29%</td>
</tr>
<tr>
<td>ia64</td>
<td>42</td>
<td>24</td>
<td>-18</td>
<td>43%</td>
</tr>
<tr>
<td>kfreebsd-amd64</td>
<td>39</td>
<td>24</td>
<td>-15</td>
<td>38%</td>
</tr>
<tr>
<td>kfreebsd-i386</td>
<td>37</td>
<td>23</td>
<td>-14</td>
<td>38%</td>
</tr>
<tr>
<td><strong>total</strong></td>
<td><strong>296</strong></td>
<td><strong>192</strong></td>
<td><strong>-104</strong></td>
<td><strong>35%</strong></td>
</tr>
</tbody>
</table>
Get the Fact!!  
(Log tells the truth...)

- **Date**: 2011/06/01-2012/05/31
- **Site**: ftp.jp.debian.org
  (actually ftp.jaist.ac.jp and it uses CDN system but most of traffic goes to jaist)
- **Log**: 105,902,720 lines
Which arch? (%)

= i386, amd64, all and source
Which arch? (size)

- Total: 83TB
  - all: 34
  - i386: 25
  - amd64: 18
  - source: 3

<table>
<thead>
<tr>
<th>architecture</th>
<th>download (TB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>34.40</td>
</tr>
<tr>
<td>alpha</td>
<td>0.02</td>
</tr>
<tr>
<td>amd64</td>
<td>17.80</td>
</tr>
<tr>
<td>arm</td>
<td>0.03</td>
</tr>
<tr>
<td>armel</td>
<td>0.66</td>
</tr>
<tr>
<td>armhf</td>
<td>0.02</td>
</tr>
<tr>
<td>hppa</td>
<td>0.01</td>
</tr>
<tr>
<td>hurd-i386</td>
<td>0.03</td>
</tr>
<tr>
<td>i386</td>
<td>25.10</td>
</tr>
<tr>
<td>ia64</td>
<td>0.15</td>
</tr>
<tr>
<td>kfreebsd-amd64</td>
<td>0.22</td>
</tr>
<tr>
<td>kfreebsd-i386</td>
<td>0.23</td>
</tr>
<tr>
<td>m68k</td>
<td>0.00</td>
</tr>
<tr>
<td>mips</td>
<td>0.10</td>
</tr>
<tr>
<td>mipsel</td>
<td>0.13</td>
</tr>
<tr>
<td>powerpc</td>
<td>0.80</td>
</tr>
<tr>
<td>s390</td>
<td>0.08</td>
</tr>
<tr>
<td>s390x</td>
<td>0.01</td>
</tr>
<tr>
<td>sh4</td>
<td>0.00</td>
</tr>
<tr>
<td>source</td>
<td>2.87</td>
</tr>
<tr>
<td>sparc</td>
<td>0.13</td>
</tr>
<tr>
<td></td>
<td>82.79</td>
</tr>
</tbody>
</table>
How much can we cut?

- If we'll apply xz...

- Cut **24TB**!
  
  - It's benefit for mirror admins

<table>
<thead>
<tr>
<th>architecture</th>
<th>download cut (TB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>7.24</td>
</tr>
<tr>
<td>alpha</td>
<td>0.00</td>
</tr>
<tr>
<td>amd64</td>
<td><strong>6.80</strong></td>
</tr>
<tr>
<td>arm</td>
<td>0.00</td>
</tr>
<tr>
<td>armel</td>
<td>0.00</td>
</tr>
<tr>
<td>armhf</td>
<td>0.00</td>
</tr>
<tr>
<td>hppa</td>
<td>0.00</td>
</tr>
<tr>
<td>hurd-i386</td>
<td>0.01</td>
</tr>
<tr>
<td>i386</td>
<td><strong>9.66</strong></td>
</tr>
<tr>
<td>ia64</td>
<td>0.06</td>
</tr>
<tr>
<td>kfreebsd-amd64</td>
<td>0.08</td>
</tr>
<tr>
<td>kfreebsd-i386</td>
<td>0.09</td>
</tr>
<tr>
<td>m68k</td>
<td>0.00</td>
</tr>
<tr>
<td>mips</td>
<td>0.00</td>
</tr>
<tr>
<td>mipsel</td>
<td>0.00</td>
</tr>
<tr>
<td>powerpc</td>
<td>0.00</td>
</tr>
<tr>
<td>s390</td>
<td>0.00</td>
</tr>
<tr>
<td>s390x</td>
<td>0.00</td>
</tr>
<tr>
<td>sh4</td>
<td>0.00</td>
</tr>
<tr>
<td>source</td>
<td>0.00</td>
</tr>
<tr>
<td>sparc</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td><strong>23.94</strong></td>
</tr>
</tbody>
</table>
Download speed issue

- **Source: 2011**

- **Global Download Study**
  - http://chartsbin.com/view/2484

- **You can check your download speed at**
  http://www.speedtest.net/
Best 5 countries

1. Korea : 2202KBps
2. Romania : 1909
3. Bulgaria : 1611
4. Lithuania : 1462
5. Latvia : 1377
Download speed average

- **United States**: 616KBps
- **Germany**: 647KBps
- **Japan**: 1364KBps (My result: 5.98 MB/s it's enough :-)
- **Nicaragua**: 180KBps

- **World Average**: 580KBps
  - North America = 500-600KBps
  - South America = 100-200KBps
  - Europe = eastern is better than western
Cut download time

- If we would update Desktop/Laptop everyday in unstable
  - Download 10-15MB (maybe) for each
    - It takes 2-3 mins
    - Xz cut 1min

- It's benefit for Debian users
  (including developers, of course :-)


Conclusion?

- How large is Debian Repository: 615GB
- One day, I found a solution... : use xz
- Is it really effective? : YES!
- Problem on slower Arch : x86 + all
- How much can we shrink it? : 100GB!
- It'll cut download traffic : 24TB/year
  - It's benefit for mirror admins
  - Also for Debian Users/Developers
Trade-off (vs decompression)

better compression vs increase decompression time
Trade-off (vs decompression)

Test Machine Spec
Intel Core i5
16GB Mem
ex1) fonts-horai-umefont_440-1_all.deb

$ du -k data.tar.*
43664  data.tar.gz
5780   data.tar.xz

$ time tar xf data.tar.gz
real  0m0.897s
user  0m0.880s
sys   0m0.104s

$ time tar xf data.tar.xz
real  0m0.619s
user  0m0.564s
sys   0m0.144s
```bash
$ cat decomp.sh
#!/bin/sh

i=0

while [ $i -lt 100 ]
do
    i=`expr $i + 1`
    tar xf $1
done

$ time ./decomp.sh data.tar.gz

real 1m43.487s
user 1m39.706s
sys  0m14.121s

$ time ./decomp.sh data.tar.xz

real 1m12.126s
user 1m5.780s
sys  0m18.169s
```
ex2) openclipart-png

$ du -k data.tar.*
621368  data.tar.gz
611520  data.tar.xz

$ time ./decomp.sh data.tar.gz
real   10m24.567s
user   9m55.829s
sys    2m16.849s

$ time ./decomp.sh data.tar.xz
real   69m28.146s
user   65m39.686s
sys    3m4.028s
ex3) non-all package – linux-image-3.2.0-3-amd64_3.2.21-3_amd64.deb (*)

$ time ../decomp.sh data.tar.gz

real  1m23.894s
user  1m20.993s
sys   0m21.061s

$ time ../decomp.sh data.tar.xz

real  3m0.363s
user  2m56.699s
sys   0m24.258s

*) linux-image-3.2.0 has already been applied xz
ex4) on non-x86 arch

... sorry, not checked yet ;-}
ex5) installing package

Good case

root@hp:/tmp/buildd# time dpkg -i fonts-horai-umefont_439-1_all.deb

real 0m0.751s
user 0m0.888s
sys 0m0.116s

root@hp:/tmp/buildd# time dpkg -i fonts-horai-umefont_440-3_all.deb

real 0m0.764s
user 0m0.848s
sys 0m0.120s
Normal case

root@hp:/tmp/buildd# time dpkg -i poppler-data_0.4.5-1_all.deb

real  0m0.129s
user  0m0.144s
sys   0m0.032s

root@hp:/tmp/buildd# time dpkg -i poppler-data_0.4.5-8_all.deb

real  0m0.233s
user  0m0.236s
sys   0m0.036s

download time =  almost same
install time     +0.104s
Worst case

root@hp:/tmp/buildd# time dpkg -i openclipart-png_2.0-2_all.deb

real  0m4.736s
user  0m6.180s
sys   0m1.568s

root@hp:/tmp/buildd# time dpkg -i openclipart-png_2.0-2.1_all.deb

real  0m40.695s
user  0m41.779s
sys   0m1.620s

download time = almost same
install time    +36s (x8)
Test tells us...

- xz decompression is slower than default gz (at most time)
  - rarely faster than gz
  - usually 2-8 times slower than gz

- it depends on its own data.
  - good compression rate = faster decompression

- it doesn't depend on running arch?
  - Not checked
Log tells the truth (again)

<table>
<thead>
<tr>
<th>package name</th>
<th>total download size(GB)</th>
<th>package name</th>
<th>numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 linux-2.6</td>
<td>4,830</td>
<td>krb5</td>
<td>723,923</td>
</tr>
<tr>
<td>2 openoffice.org</td>
<td>2,853</td>
<td>eglibc</td>
<td>683,543</td>
</tr>
<tr>
<td>3 libreoffice</td>
<td>2,346</td>
<td>linux-2.6</td>
<td>679,016</td>
</tr>
<tr>
<td>4 eglibc</td>
<td>1,566</td>
<td>cups</td>
<td>613,836</td>
</tr>
<tr>
<td>5 texlive-extra</td>
<td>1,432</td>
<td>openoffice.org</td>
<td>591,510</td>
</tr>
<tr>
<td>6 mesa</td>
<td>1,223</td>
<td>mono</td>
<td>580,730</td>
</tr>
<tr>
<td>7 evolution</td>
<td>1,199</td>
<td>evolution-data-server</td>
<td>537,474</td>
</tr>
<tr>
<td>8 freepats</td>
<td>1,111</td>
<td>bind9</td>
<td>513,989</td>
</tr>
<tr>
<td>9 texlive-base</td>
<td>1,022</td>
<td>libreoffice</td>
<td>507,735</td>
</tr>
<tr>
<td>10 samba</td>
<td>1,018</td>
<td>avahi</td>
<td>497,764</td>
</tr>
</tbody>
</table>
Top 50 packages

- They ate 47% of all traffic (39 / 82TB)
  - First target?
...then, how to apply it?

- Apply top 50 packages?
- Modify `debhelper`? (to apply `xz` for all/i386/amd64 by default)
- Modify build daemon?
- Mass-rebuild for i386/amd64/all arch?
- **Thoughts?**
  (after this presentation, welcome YOUR comment :-)

Conclusion (really)

- How large is Debian Repository: 615GB
- One day, I found a solution... : use xz
- Is it really effective? : YES!
- Problem on slower Arch : x86 + all
- How shrink : 100GB!
- **It'll cut download traffic** : 24TB/year

So, recommend to apply XZ to all, *i386 and *amd64 if we can (surely exclude “Priority:require”)
Also, Thanks to nice pictures

• SpaceFun
  http://wiki.debian.org/DebianArt/Themes/SpaceFun
  By Valessio Brito
  licensed under GPL-2

• Debian Theme (etch?)

• Debian Theme (by @nogajun)

• Thinking
  http://www.flickr.com/photos/nachoissd/3499105933/
  By Victor Pérez :: victorperezp.com
  licensed under Creative Commons Attribution 2.0 Generic (CC BY 2.0)

• A successful tool is one that was used to do something undreamed of by its author.
  http://www.flickr.com/photos/katerha/5746905652/
  By katerha
  licensed under Creative Commons Attribution 2.0 Generic (CC BY 2.0)