Let's shrink “bloated Debian repository”

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

debian
The Universal Operating System

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?

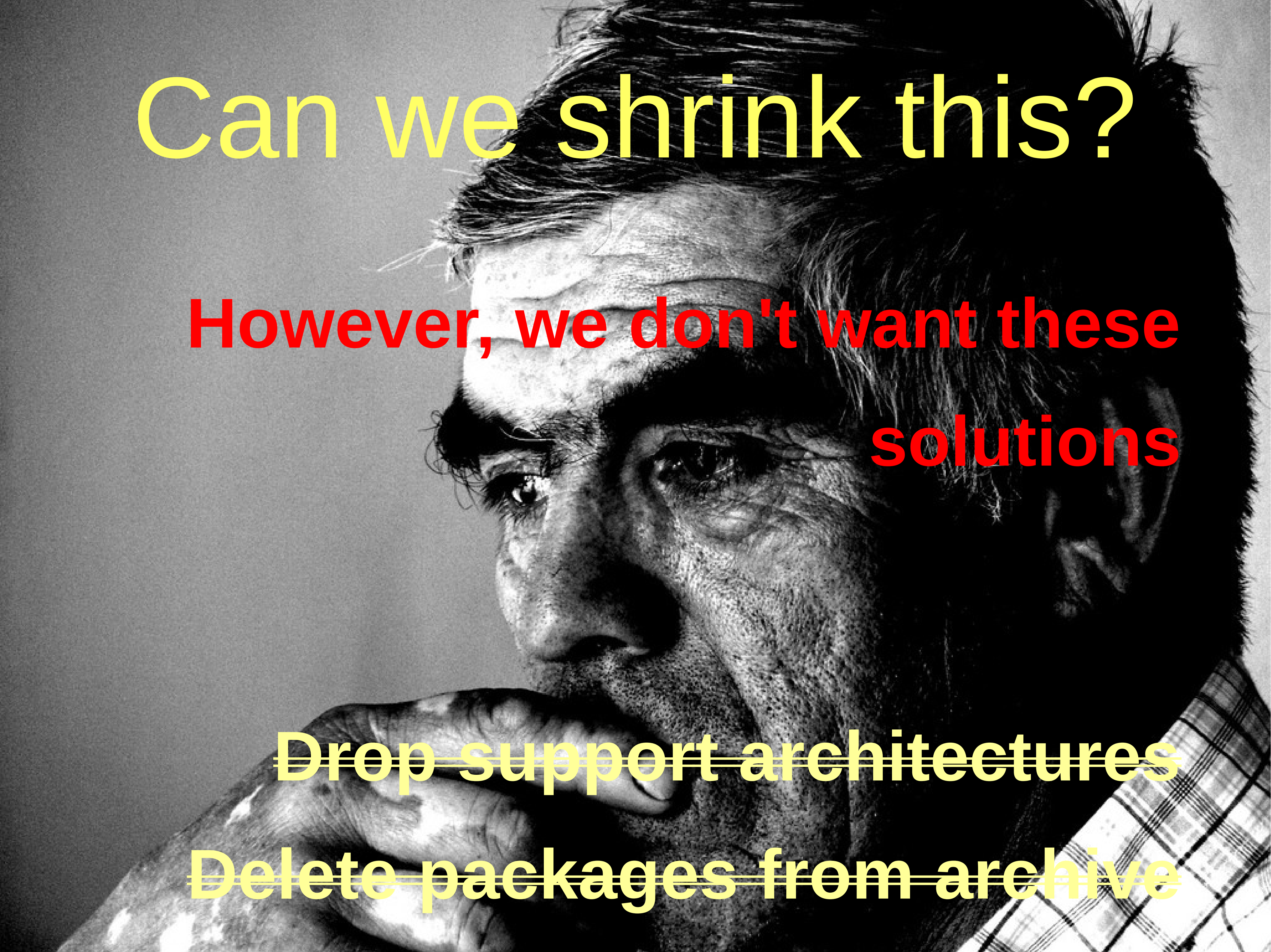
Can we shrink this?

Yes, in some ways...

Drop support architectures

Delete packages from archive

Can we shrink this?

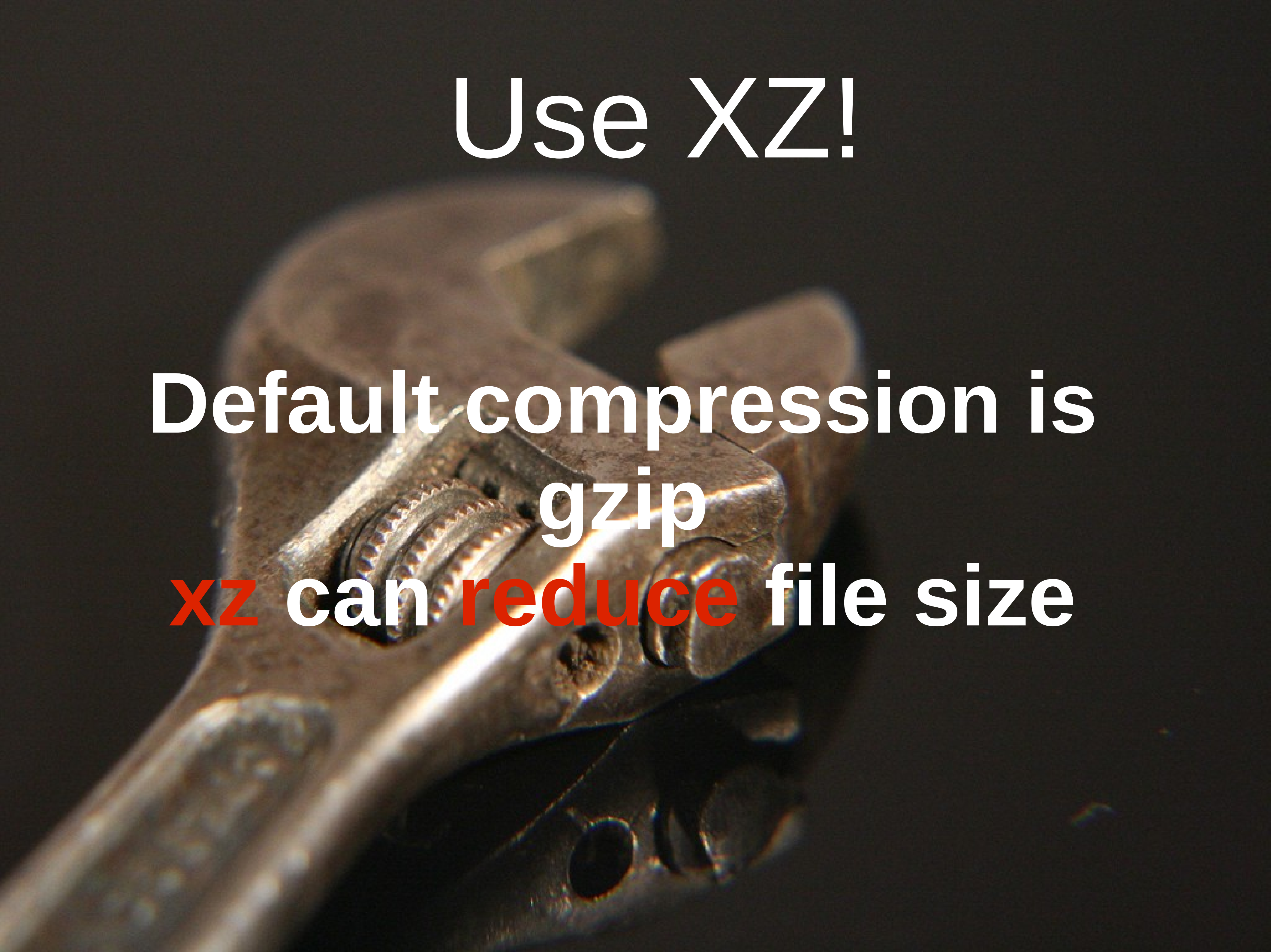


However, we don't want these solutions

~~**Drop support architectures**~~

~~**Delete packages from archive**~~

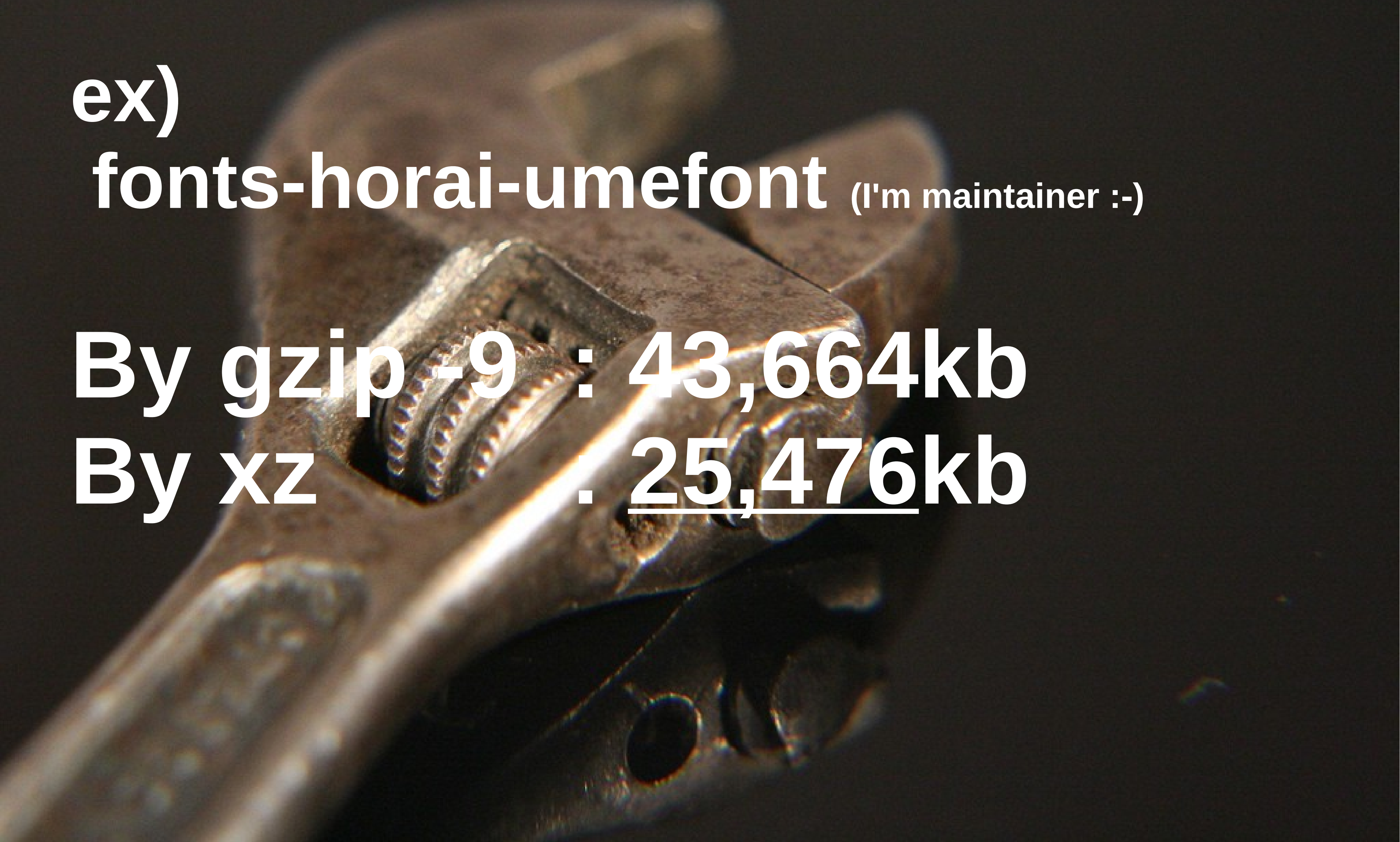
Use XZ!



Default compression is gzip

xz can reduce file size

Use XZ!

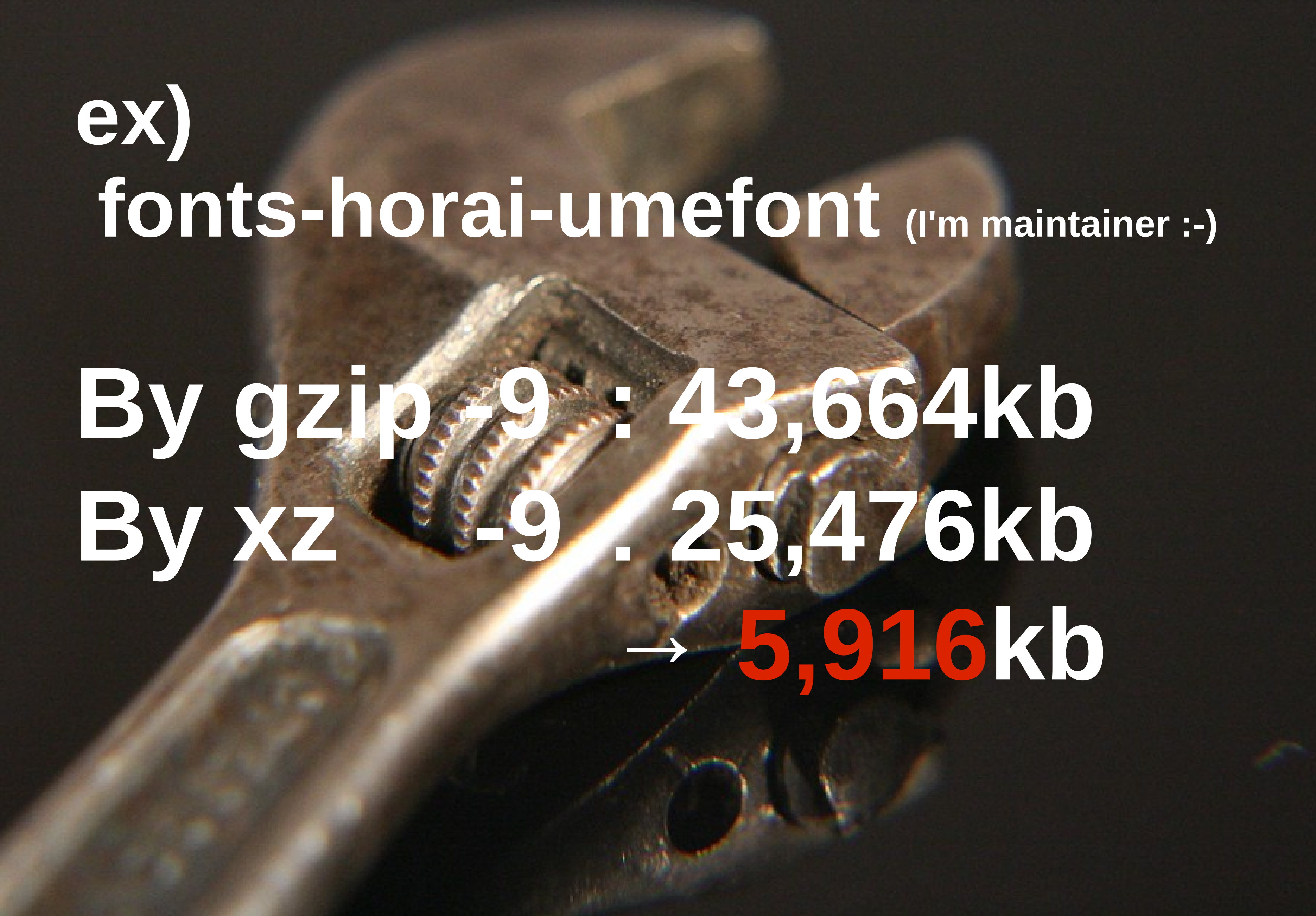


ex)

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

WARNING!

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.

("The archive now supports xz compression" by Ansgar Burchardt <ansgar@debian.org>
http://lists.debian.org/debian-devel-announce/2011/08/msg00001.html)

WARNING!

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.

("The archive now supports xz compression" by Ansgar Burchardt <ansgar@debian.org>
http://lists.debian.org/debian-devel-announce/2011/08/msg00001.html)

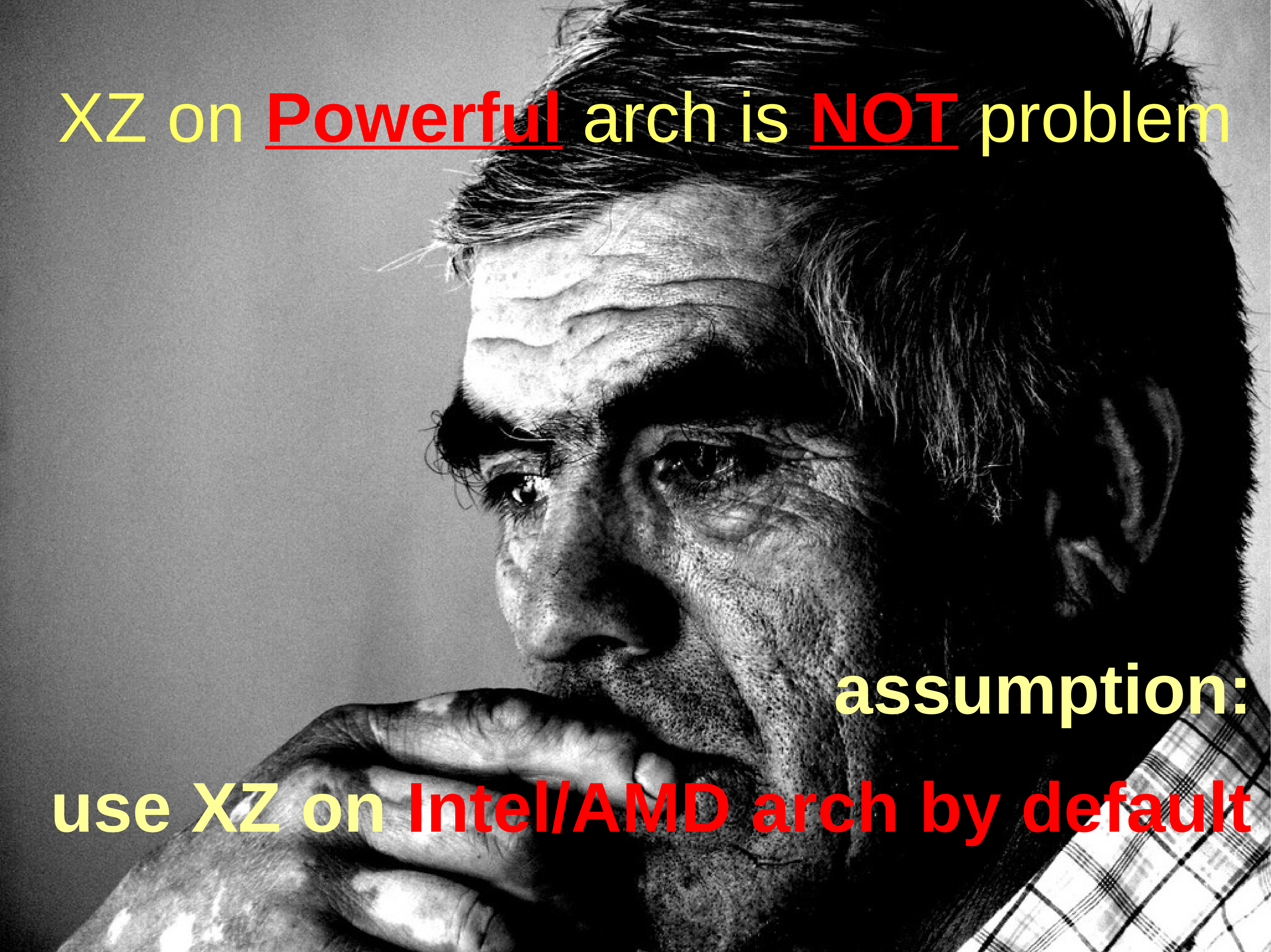
XZ on Slower arch is problem...

It'll eat most CPU time

XZ on **Slower** arch is problem...

Then...

if only on Powerful arch?

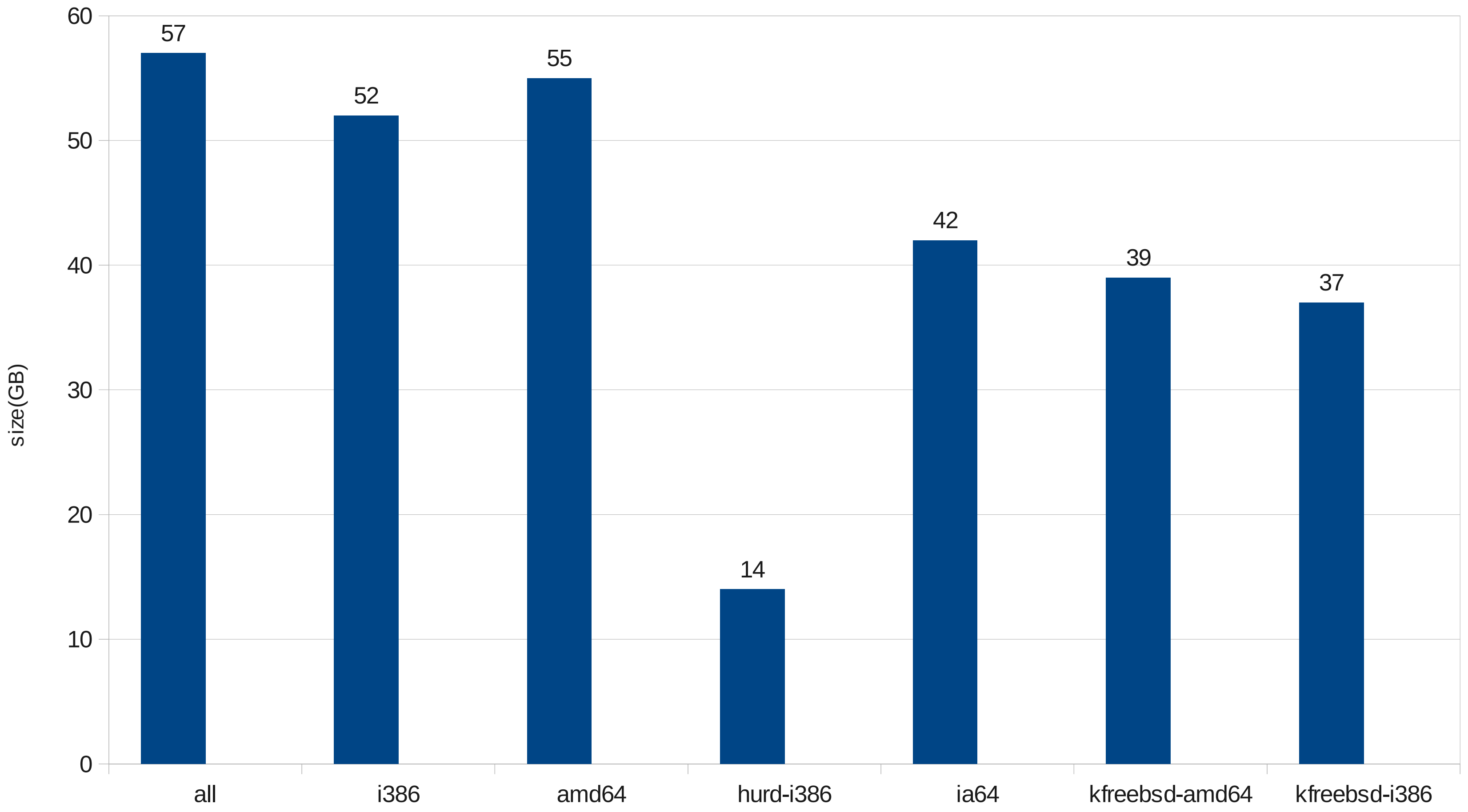


XZ on **Powerful** arch is **NOT** problem

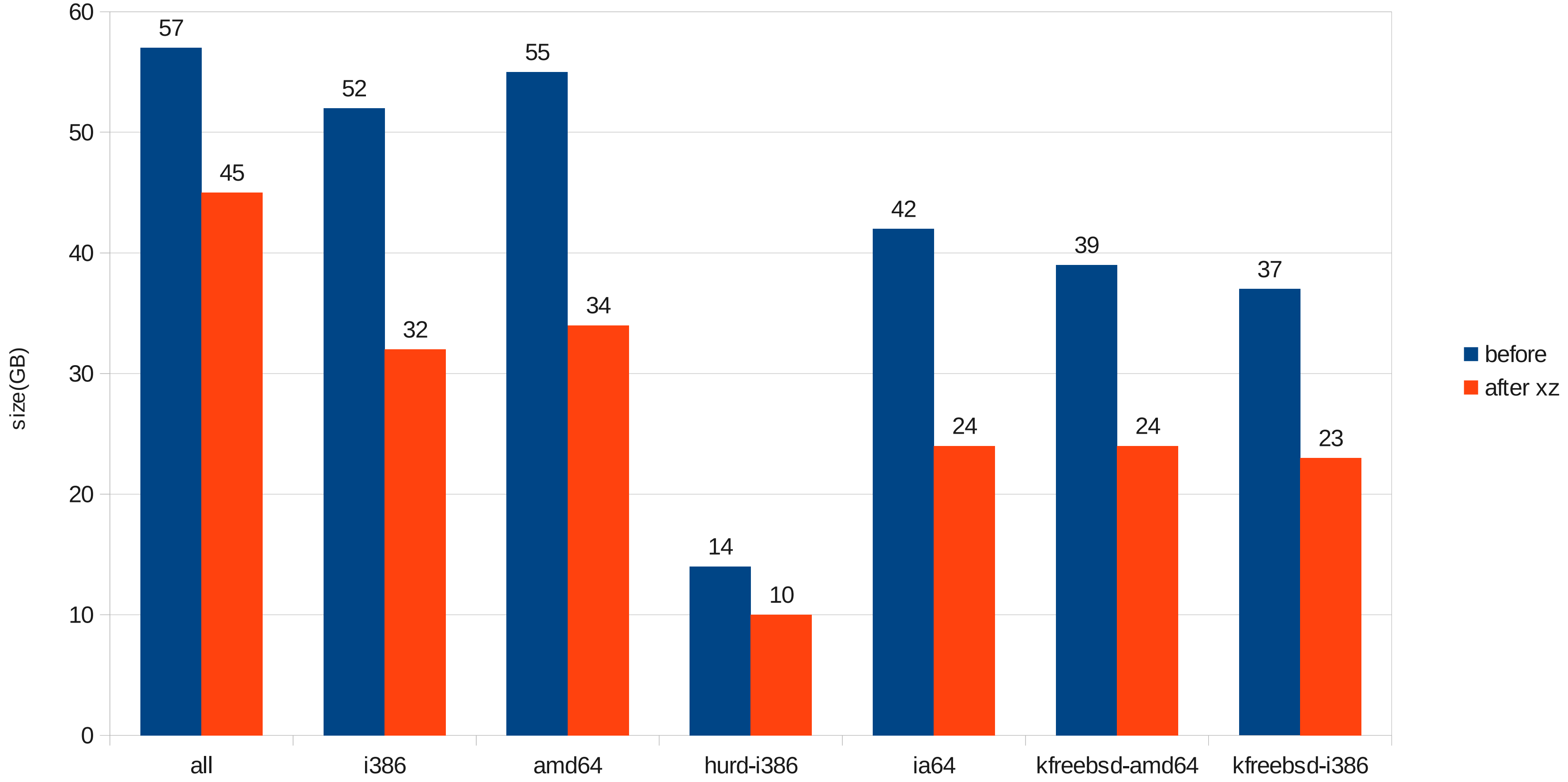
assumption:

use XZ on Intel/AMD arch by default

Before XZ...

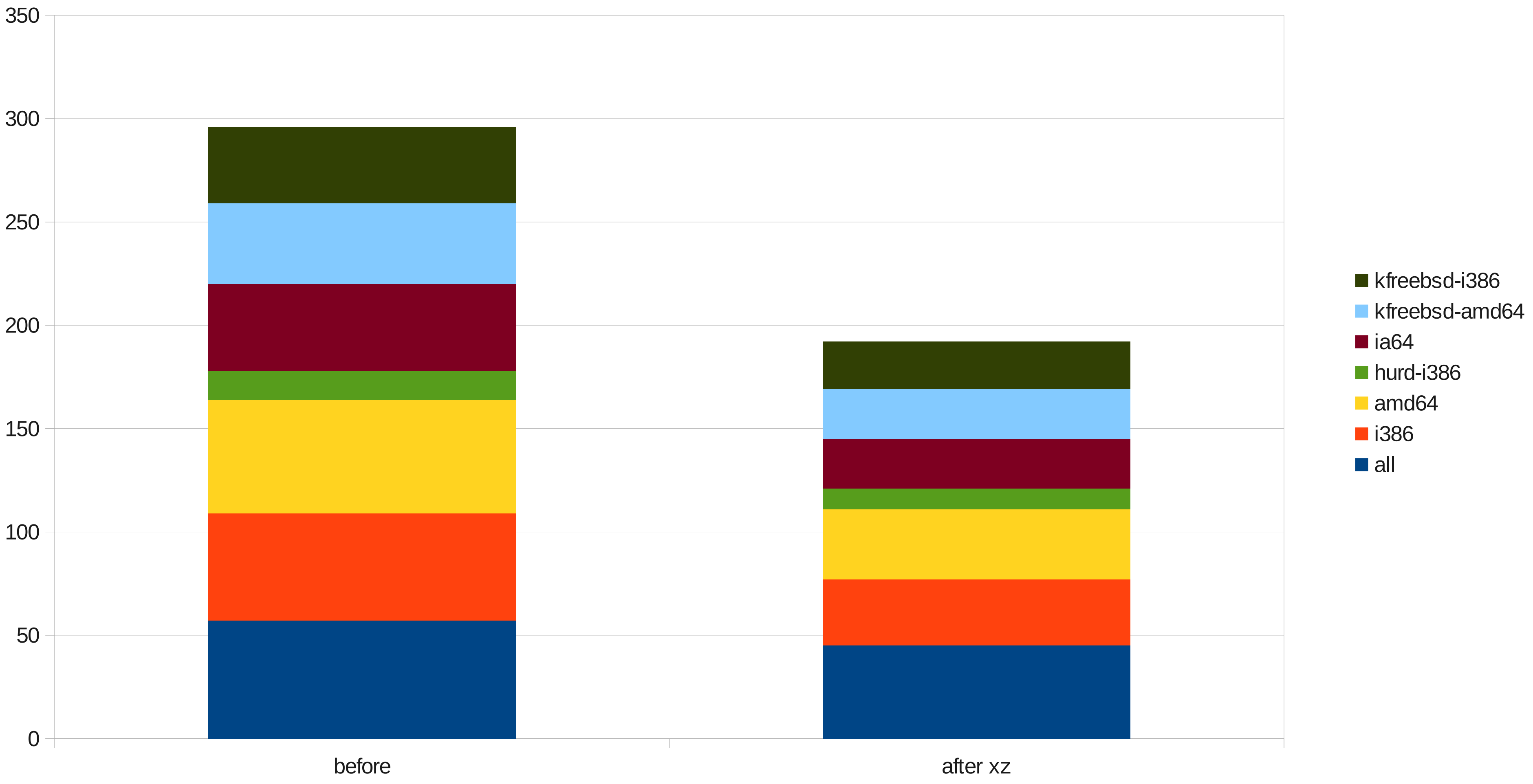

60
50
40
30
20
10
0
size(GB)
57
52
55
14
42
39
37
all
i386
amd64
hurd-i386
ia64
kfreebsd-amd64
kfreebsd-i386

After XZ!



60
50
40
30
20
10
0
size(GB)
57
45
52
32
55
34
14
10
42
24
39
24
37
23
all
i386
amd64
hurd-i386
ia64
kfreebsd-amd64
kfreebsd-i386
before
after xz

How much can we shrink it?



How much can we shrink it?

architecture	before	after xz	difference	Reduction Rate
all	57	???	---	---
i386	52	???	---	---
amd64	55	???	---	---
hurd-i386	14	???	---	---
ia64	42	???	---	---
kfreebsd-amd64	39	???	---	---
kfreebsd-i386	37	???	---	---
total	296	???	---	---

How much can we shrink it?

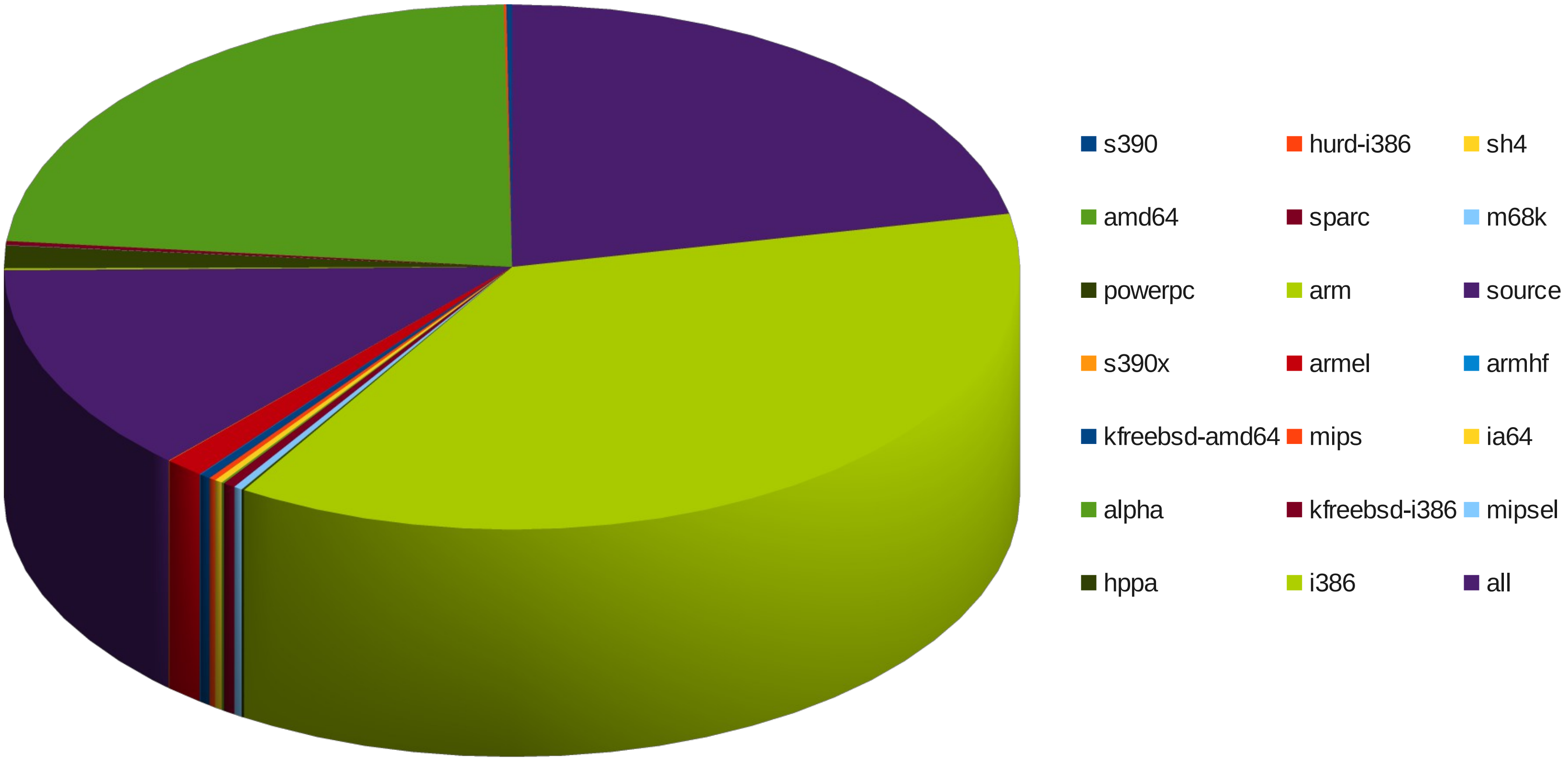
architecture	before	after xz	difference	Reduction Rate
all	57	45	-12	21%
i386	52	32	-20	38%
amd64	55	34	-21	38%
hurd-i386	14	10	-4	29%
ia64	42	24	-18	43%
kfreebsd-amd64	39	24	-15	38%
kfreebsd-i386	37	23	-14	38%
total	296	192	**-104**	**35%**

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

architecture	download (TB)
all	**34.40**
alpha	0.02
amd64	**17.80**
arm	0.03
armel	0.66
armhf	0.02
hppa	0.01
hurd-i386	0.03
i386	**25.10**
ia64	0.15
kfreebsd-amd64	0.22
kfreebsd-i386	0.23
m68k	0.00
mips	0.10
mipsel	0.13
powerpc	0.80
s390	0.08
s390x	0.01
sh4	0.00
source	**2.87**
sparc	0.13
	82.79

How much can we cut?

- **If we'll apply xz...**
  - Cut **24TB**!
    - It's benefit for mirror admins

architecture	download cut (TB)
all	**7.24**
alpha	0.00
amd64	**6.80**
arm	0.00
armel	0.00
armhf	0.00
hppa	0.00
hurd-i386	0.01
i386	**9.66**
ia64	0.06
kfreebsd-amd64	0.08
kfreebsd-i386	0.09
m68k	0.00
mips	0.00
mipsel	0.00
powerpc	0.00
s390	0.00
s390x	0.00
sh4	0.00
source	0.00
sparc	0.00
	23.94

Download speed issue

- **Source: 2011**
  - Pando Networks Releases Global Internet Speed Study, Pando Networks Inc 2011, viewed 22th September, 2011, <http://www.pandonetworks.com/Pando-Networks-Releases-Global-Internet-Speed-Study>.
- **Global Download Study**
  - http://chartsbin.com/view/2484
- **You can check your download speed at http://www.speedtest.net/**

Download speed average

- **Best 5 countries**

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

debian

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

debian

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

Test1

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
```

Test1.5

```
$ 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
```

Test2

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
```

Test3

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

Test4

ex4) on non-x86 arch

... sorry, not checked yet ;-)

Test5

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
```

Test5

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

Test5

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)

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

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)