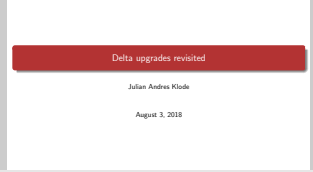
Delta upgrades revisited

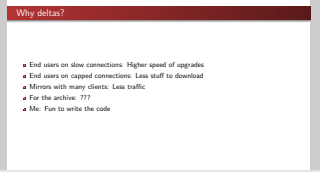
Julian Andres Klode

August 3, 2018



Why deltas?

- End users on slow connections: Higher speed of upgrades
- End users on capped connections: Less stuff to download
- Mirrors with many clients: Less traffic
- For the archive: ???
- Me: Fun to write the code



what about debdelta?

It's complicated.

what about debdelta?

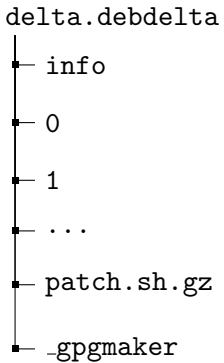


Figure: The debdelta format

what about debdelta?
delta.debdelta
info
0
1
...
patch.sh.gz
_gpgmaker
Figure: The debdelta format

1. debdelta regenerates a deb from a delta
2. a script stored in the delta builds the deb
3. - script could do anything
4. - cannot predict what it would do
5. patch data in numbered files
6. signature inline

Let's start from scratch

A delta deb is a deb of deltas.

Structure of a delta deb

```
delta.δdeb
├─ debian-delta
├─ control.tar.xz                same as in full deb
│  ├─ control
│  └─ ...
└─ data.tar.xz
   ├─ usr/lib/myfile             delta against old file
   └─ etc/my/conffile            delta against /dev/null
```

Figure: A delta deb

Structure of a delta deb
delta.δdeb
debian-delta
control.tar.xz same as in full deb
control
...
data.tar.xz
usr/lib/myfile delta against old file
etc/my/conffile delta against /dev/null
Figure: A delta deb

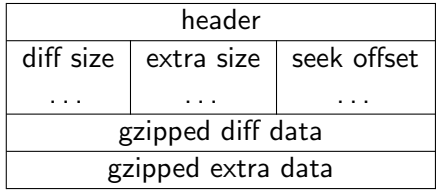
1. A delta deb looks like a normal deb
2. The control tarball is essentially unchanged
3. The data tarball contains deltas instead of actual files
4. conffiles are represented as deltas against /dev/null - they contain all the delta
5. Support for deltas only requires changing the bit in dpkg that writes dpkg-new files

How to apply a deltadeb?

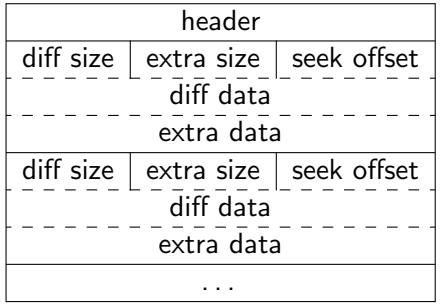
Just dpkg -i it.

1. not much to say here
2. honestly, just install it like a normal deb
3. only have to change the dpkg code that reads old file
4. to read regenerated file from delta applier instead

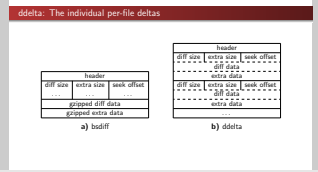
ddelta: The individual per-file deltas



a) bsdiff



b) ddelta



- ddelta is a fork of bsdiff, written by me
- bsdiff consists of a header followed by three blocks of data:
  - control data
  - the diff (new data - old data)
  - extra: new data to write to file
- Control data is a vector of triplets (diff, extra, seek) defining
  - diff: how many bytes to use from the diff block
  - extra: how many bytes to copy from the extra block
  - seek: how much to seek in the old file.
- Compressed with gzip (same contents in diff means diff = 0 =¿ compresses well)
- Requires us to keep entire delta in memory or on disk and seek in it

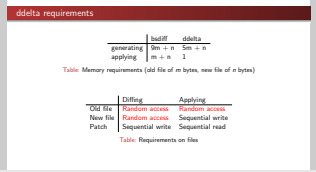
ddelta requirements

	bsdiff	ddelta
generating	9m + n	5m + n
applying	m + n	1

Table: Memory requirements (old file of *m* bytes, new file of *n* bytes)

	Diffing	Applying
Old file	Random access	Random access
New file	Random access	Sequential write
Patch	Sequential write	Sequential read

Table: Requirements on files



1. ddelta requires about half the memory of bsdiff when generating deltas, thanks to patches from chrome and android
2. and it only requires constant memory to apply vs linear
3.
4. ddelta requires random access on old and new file when generating the delta
5. but only on the old file when applying it

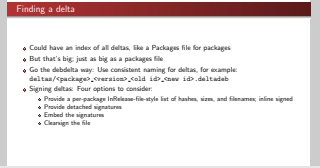
Package ID

- Might have two different packages with same version, but different contents
- Installing a delta deb would fail because contents are different
- Solution: A unique identifier for each package
- One approach: A hash of the list of files and their hashes (or well, a hash of the mtree)

Package ID
- Might have two different packages with same version, but different contents
- Installing a delta deb would fail because contents are different
- Solution: A unique identifier for each package
- One approach: A hash of the list of files and their hashes (or well, a hash of the mtree)

Finding a delta

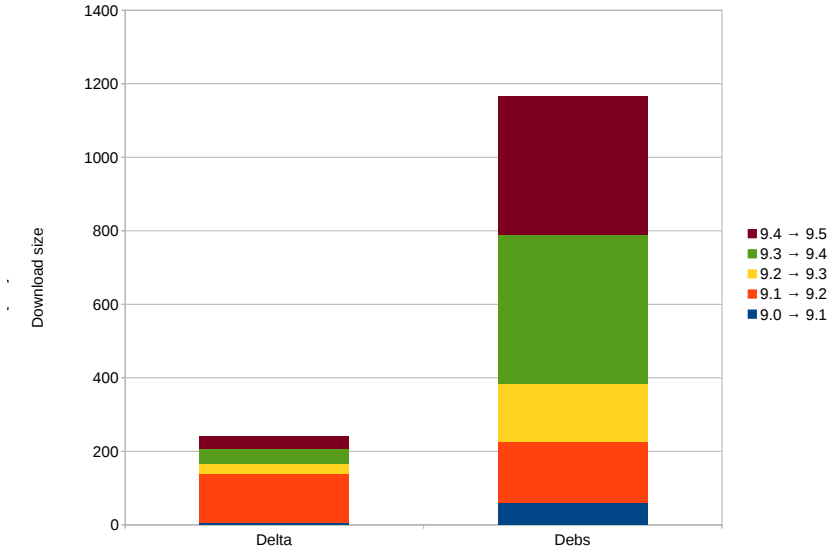
- Could have an index of all deltas, like a Packages file for packages
- But that's big; just as big as a packages file
- Go the debdelta way: Use consistent naming for deltas, for example: `deltas/<package>_<version>_<old id>_<new id>.deltadeb`
- Signing deltas: Four options to consider:
  - Provide a per-package InRelease-file-style list of hashes, sizes, and filenames; inline signed
  - Provide detached signatures
  - Embed the signatures
  - Clearsign the file

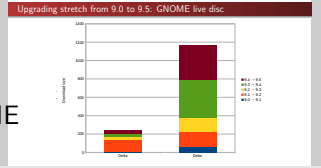


Let's talk numbers

Let's talk numbers

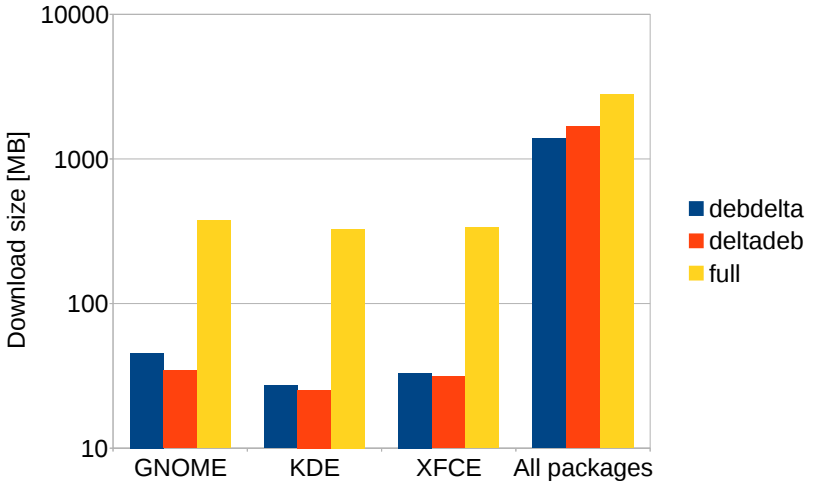
Upgrading stretch from 9.0 to 9.5: GNOME live disc

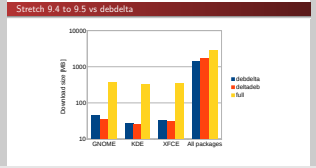




1. Debian stable is optimal for deltas - only few things change
2. here: a simple upgrade of a gnome desktop
3. downloaded 240 MB instead of 1165 MB
4. that's 924 MB saved, or about 80 per cent

Stretch 9.4 to 9.5 vs debdelta

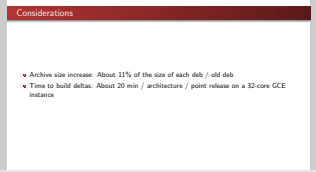




1. our deltas perform better in usual installations
2. perform slightly worse overall by 60% vs 50% of deb size

Considerations

- Archive size increase: About 11% of the size of each deb / old deb
- Time to build deltas: About 20 min / architecture / point release on a 32-core GCE instance



Questions?

Questions?