

Debian Package Caching: Apt-cacher

Jonathan Oxer

Debian Mini-Conf 2

Perth, Australia

Jan 20-21, 2003

Overview

- Package caching background
- Options for package storage
- Apt-cacher's internal mechanism
- Cache cleaning and reporting
- Setting up apt-cacher
- Mirror or cache?
- Package popularity curve

Apt-cacher Background

- Debian now widely deployed in larger networks
- Homogenous installations quite common
- Frequent duplicate requests slow and expensive
- Packages should be stored locally for re-use

Package Storage Options

- Running a local mirror
- NFS mounting `/var/cache/apt`
- Moving packages with scripts (`apt-move`)
- Traditional HTTP proxy such as Squid
- Dedicated caching system (`apt-cacher`, `apt-proxy`, `apt-cached`)

Apt-cacher Background

- Written by Nick Andrew to maintain two Debian boxes on a modem connection
- Looked at alternatives: Squid, copying `/var/cache/apt`, apt-proxy, decided to re-invent the wheel

Insult Rusty

- Rusty, stop playing FreeCiv!

Apt-cacher Structure

- Runs as a CGI under Apache
- Uses simple disk-based cache
- Calls Wget to fetch new packages

Big Fat Bug

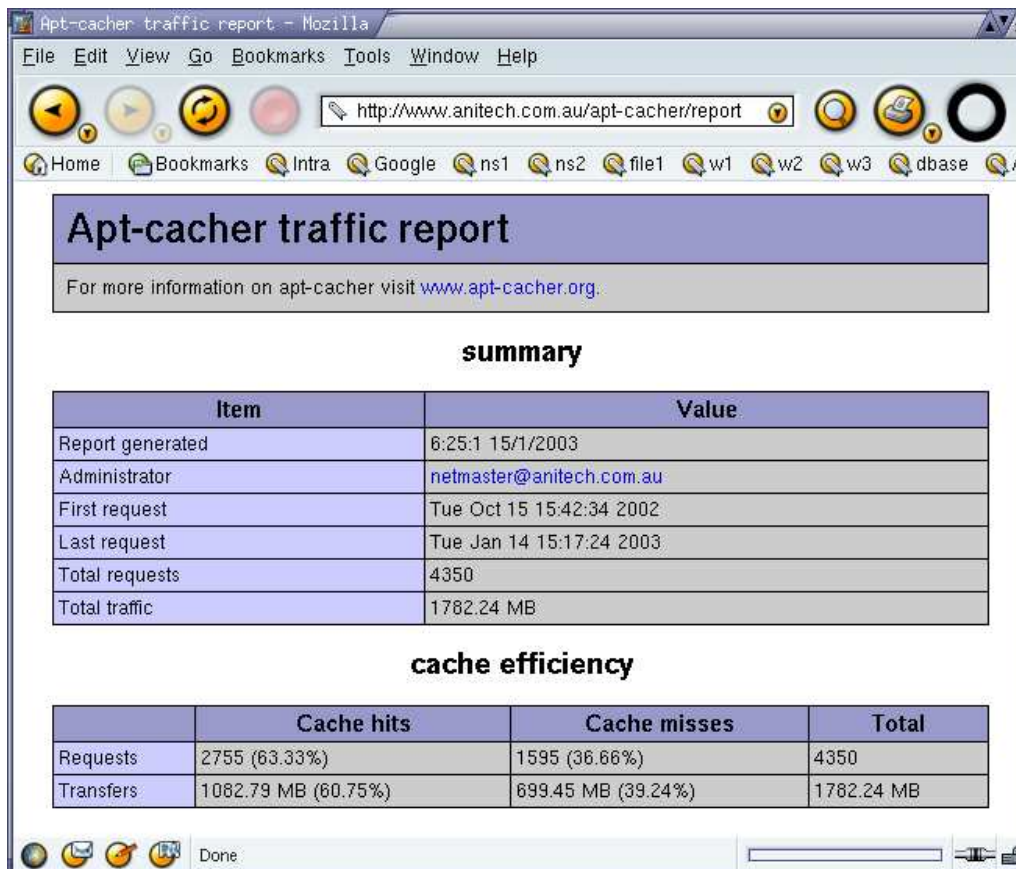
- Apache reports 500 internal server error
- Apt-cacher reports 404 file not found
- Apt-cacher gets the package anyway!
- Second request works fine

Cache Structure Questions

- Why not use `/var/cache/apt`?
- Can the cache be primed?

Reporting and Cleaning

- Reports generated by apt-cacher-report.pl



Apt-cacher traffic report - Mozilla

File Edit View Go Bookmarks Tools Window Help

http://www.anitech.com.au/apt-cacher/report

Home Bookmarks Intra Google ns1 ns2 file1 w1 w2 w3 dbase /.

Apt-cacher traffic report

For more information on apt-cacher visit www.apt-cacher.org.

summary

Item	Value
Report generated	6:25:1 15/1/2003
Administrator	netmaster@anitech.com.au
First request	Tue Oct 15 15:42:34 2002
Last request	Tue Jan 14 15:17:24 2003
Total requests	4350
Total traffic	1782.24 MB

cache efficiency

	Cache hits	Cache misses	Total
Requests	2755 (63.33%)	1595 (36.66%)	4350
Transfers	1082.79 MB (60.75%)	699.45 MB (39.24%)	1782.24 MB

Done

Reporting and Cleaning

- Reports generated by apt-cacher-report.pl
- Report processing extremely fast (~0.01 secs)
- Cleaning done by apt-cacher-cleanup.pl
- Checks objects against package lists
- Cleaning is slow (~14 seconds)

Setup

- Apt-cacher only goes on one machine
- Client machines have their sources.list modified:

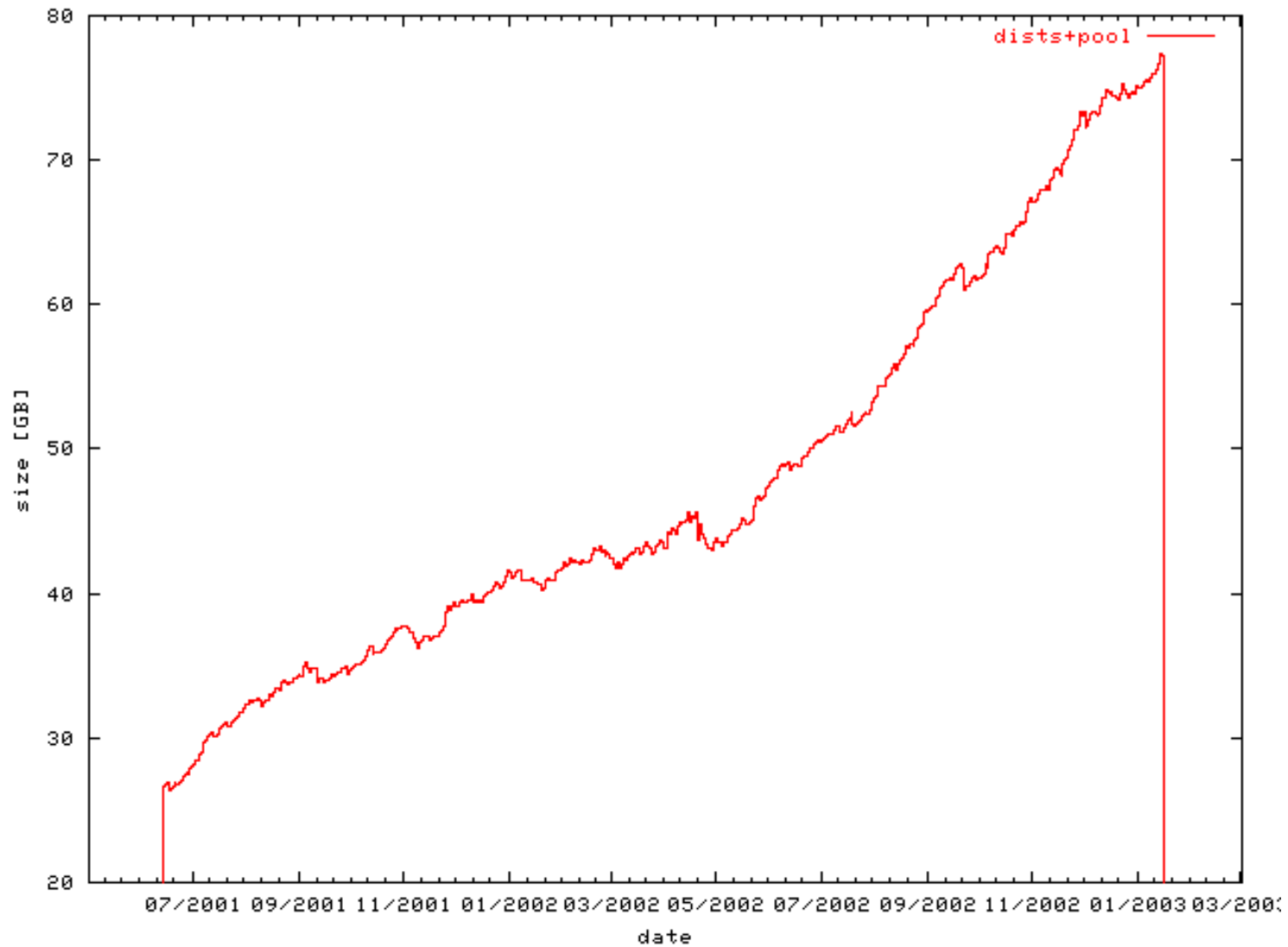
Future

- Remove reliance on HTTP headers
- Add ability to parse normal requests – mirror mimic!

Mirror or Cache?

- Actually more similar than people think
- Mirroring is pre-emptive, at tree level
- Caching is on-demand, at object level
- Cache is like a self-pruning, self-grafting mirror
- Mirror provides true redundancy – Satie!
- Cache can mimic a mirror
- Caches have more even load
- Currently about 270 mirrors for Main

Mirror Size



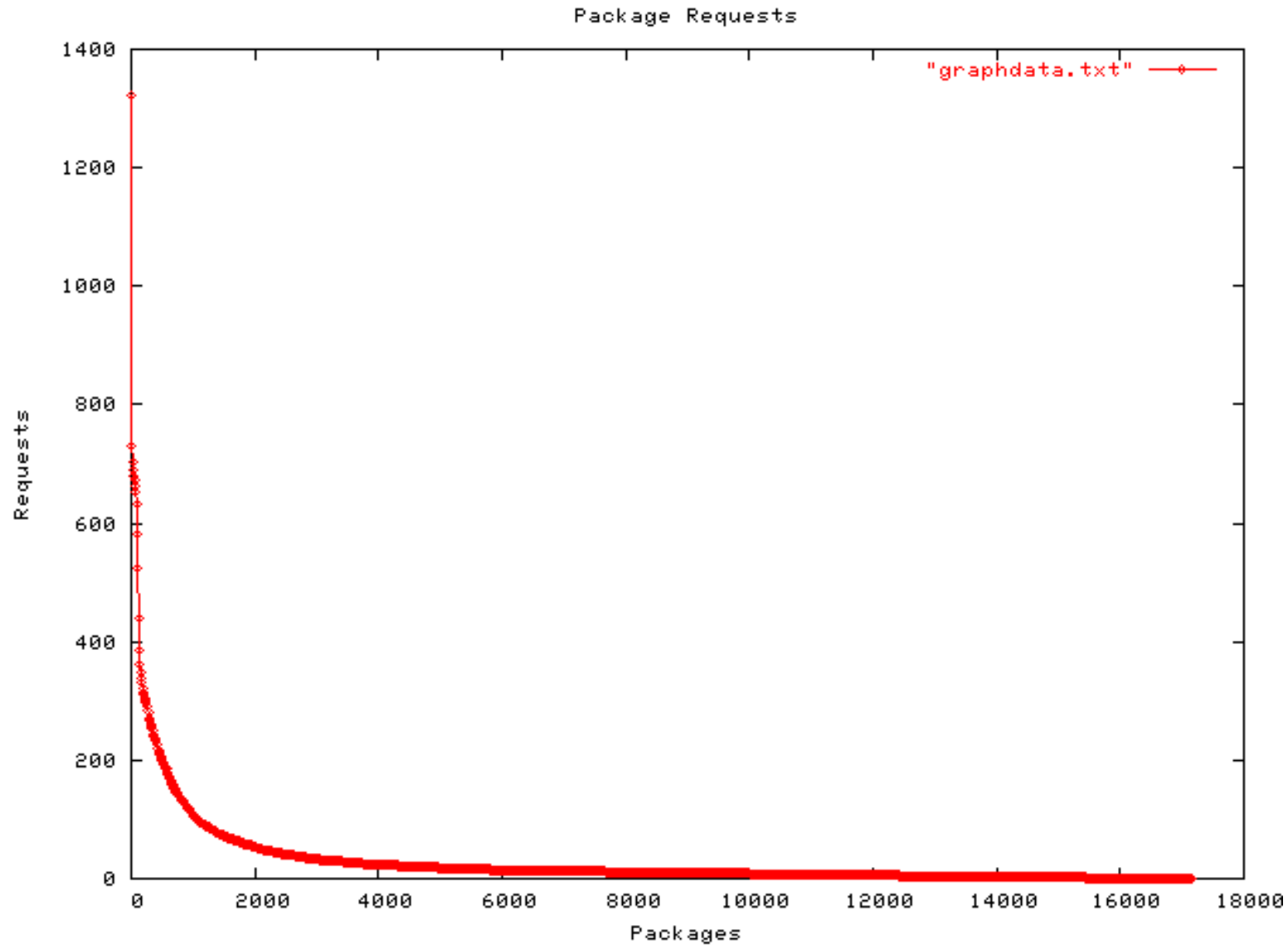
Mirror Size

- As popularity grows, so does infrastructure load

Mirror Stats: ftp.it.debian.org

- December: 505,000 HTTP package requests
- 17,170 distinct items
- foobar_1.1-1_all.deb <> foobar_1.1-2_all.deb
- Don't know about unrequested packages
- Mean average 29.5 requests / object
- Request range from 0 to 1300+

Mirror Stats: ftp.it.debian.org



Uninformed Opinion

- In many places that a mirror is run, a cache could be used with potentially better efficiency and less load
- Smaller networks should definitely use caches
- High level mirrors feeding multiple levels of caches

Game Over

Thanks for playing

Please insert 20c to continue