

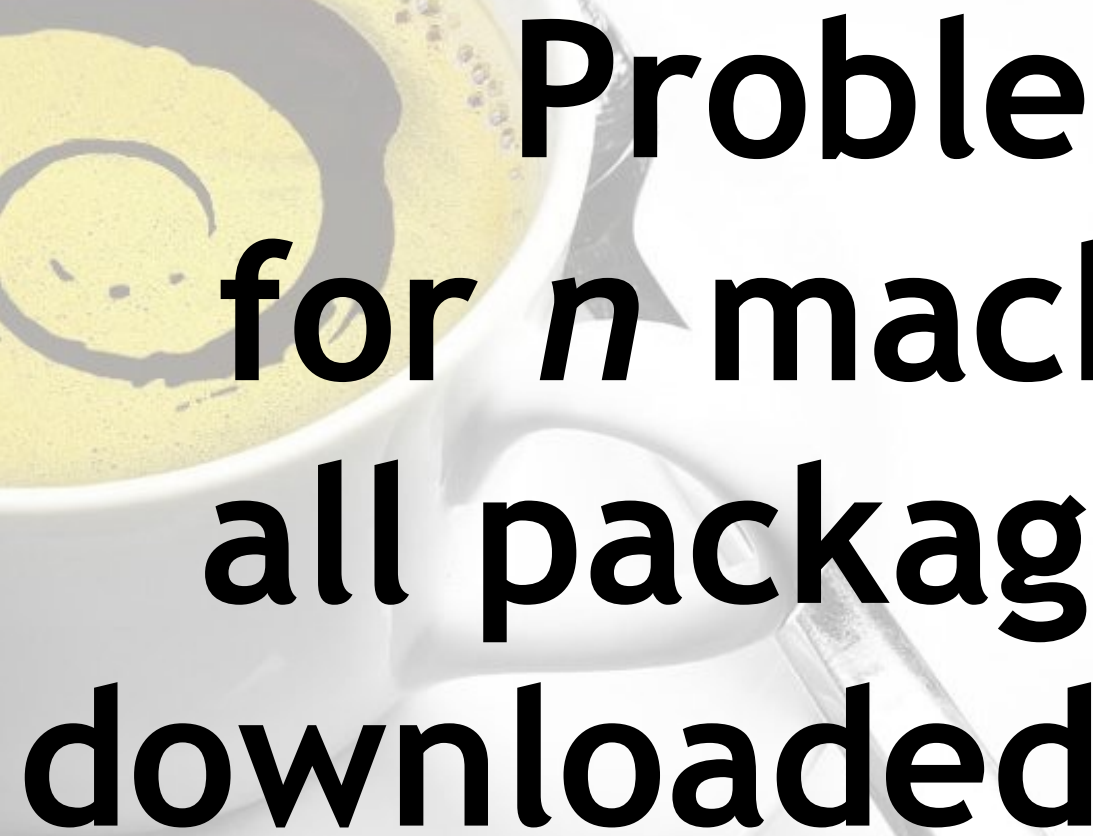


Package Caching Solutions

Jonathan Oxer

Debian Miniconf 7
January 28, 2008
Melbourne, Australia

debian

A close-up photograph of a white ceramic coffee cup filled with a light-colored latte. The surface of the coffee is decorated with a circular latte art design, possibly a 'candy' or 'heart' pattern. The background is slightly blurred, showing a white surface and a portion of a white object, possibly a sugar dispenser or a creamer.

Problem:
for n machines,
all packages are
downloaded n times



**So run a local
mirror, right?
Not so fast!**



**A mirror is a dumb
copy of the
entire archive...**



**...including every
package you've
never installed!**



ftp.au.debian.org

**Dec 16th 2007 - Jan 19th 2008:
1,200,598 package downloads**



ftp.au.debian.org

Dec 16th 2007 - Jan 19th 2008:

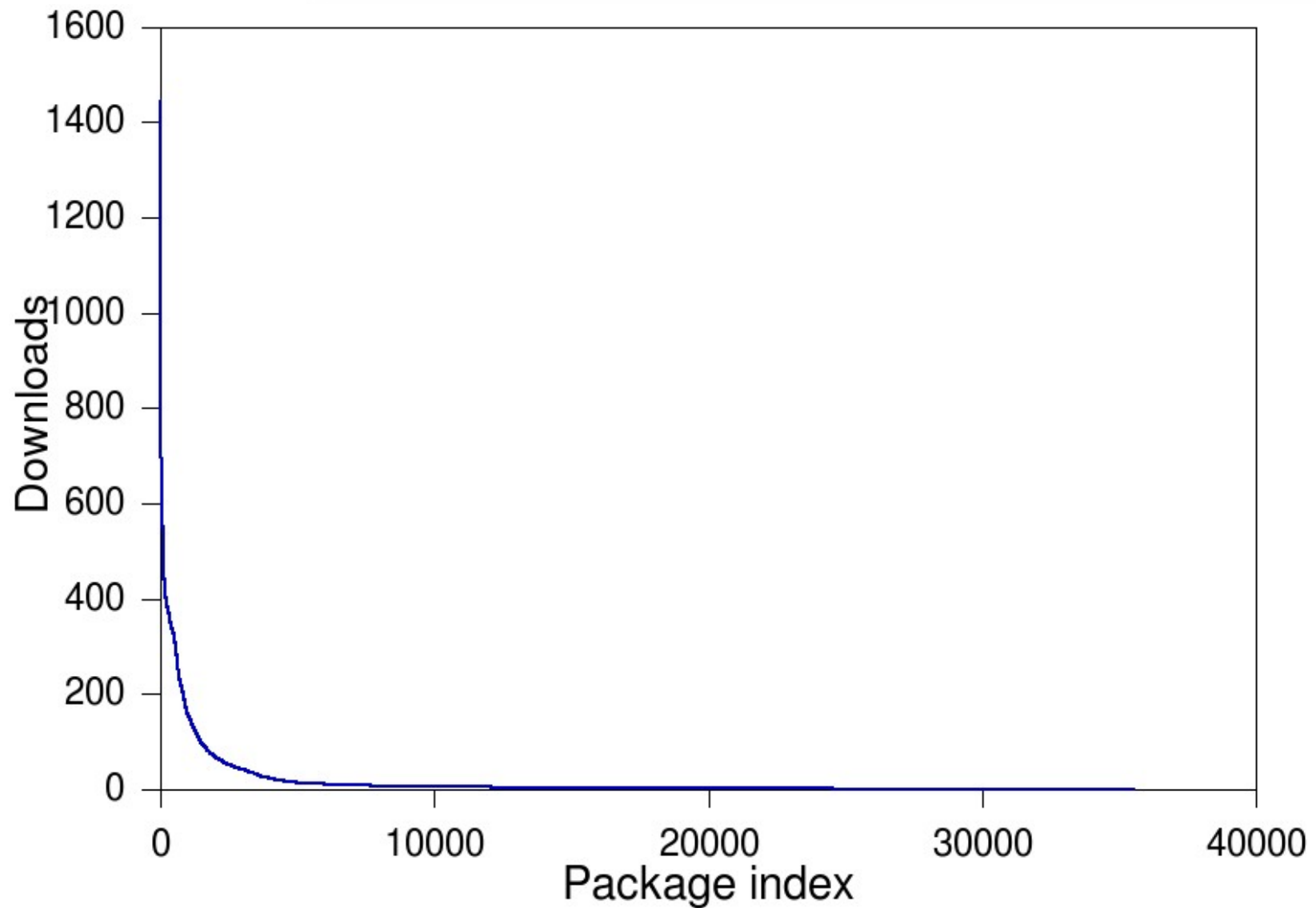
1,200,598 package downloads

277,110 different .debs

(35,513 in i386 alone)

16,296 packages in i386

Downloads / .deb: Debian i386



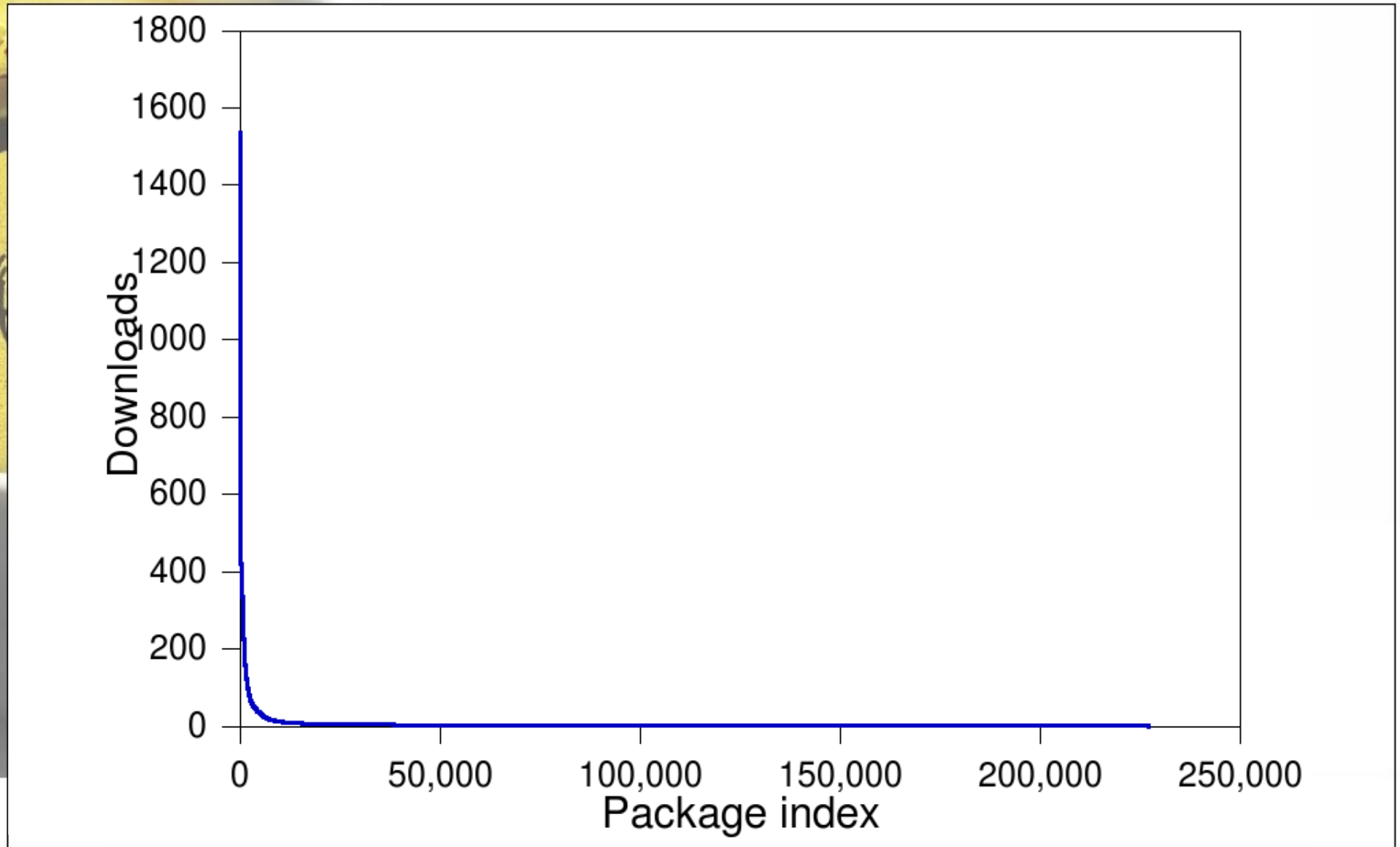
A cup of beer with a white teacup and saucer in the background.

**In Debian/i386
10% of packages
account for 89.8%
of downloads**

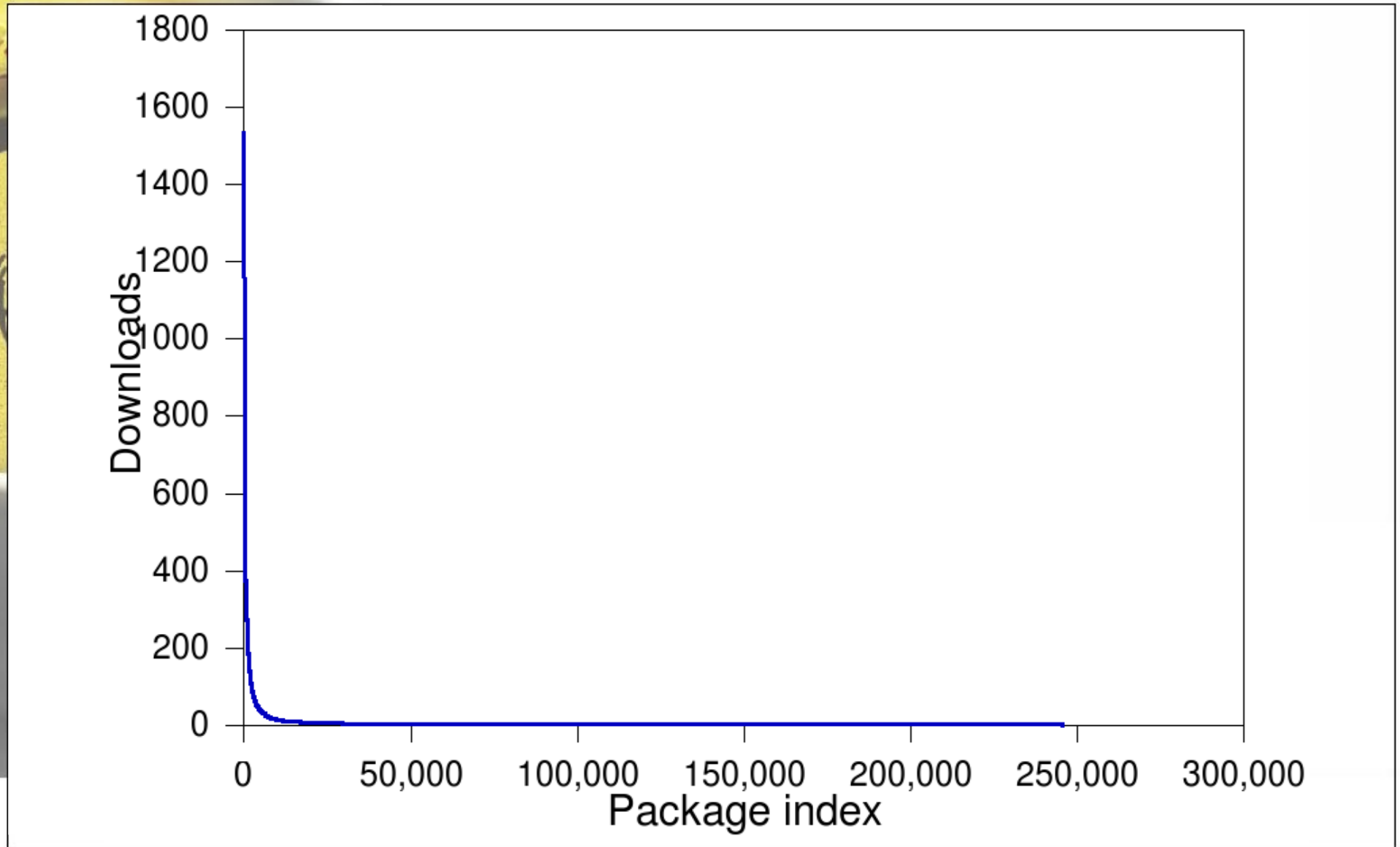


**10,989 .debs
(31%) were
downloaded
only once!**

Downloads / .deb: Debian all architectures



Downloads / .deb: Debian, Ubuntu, Backports

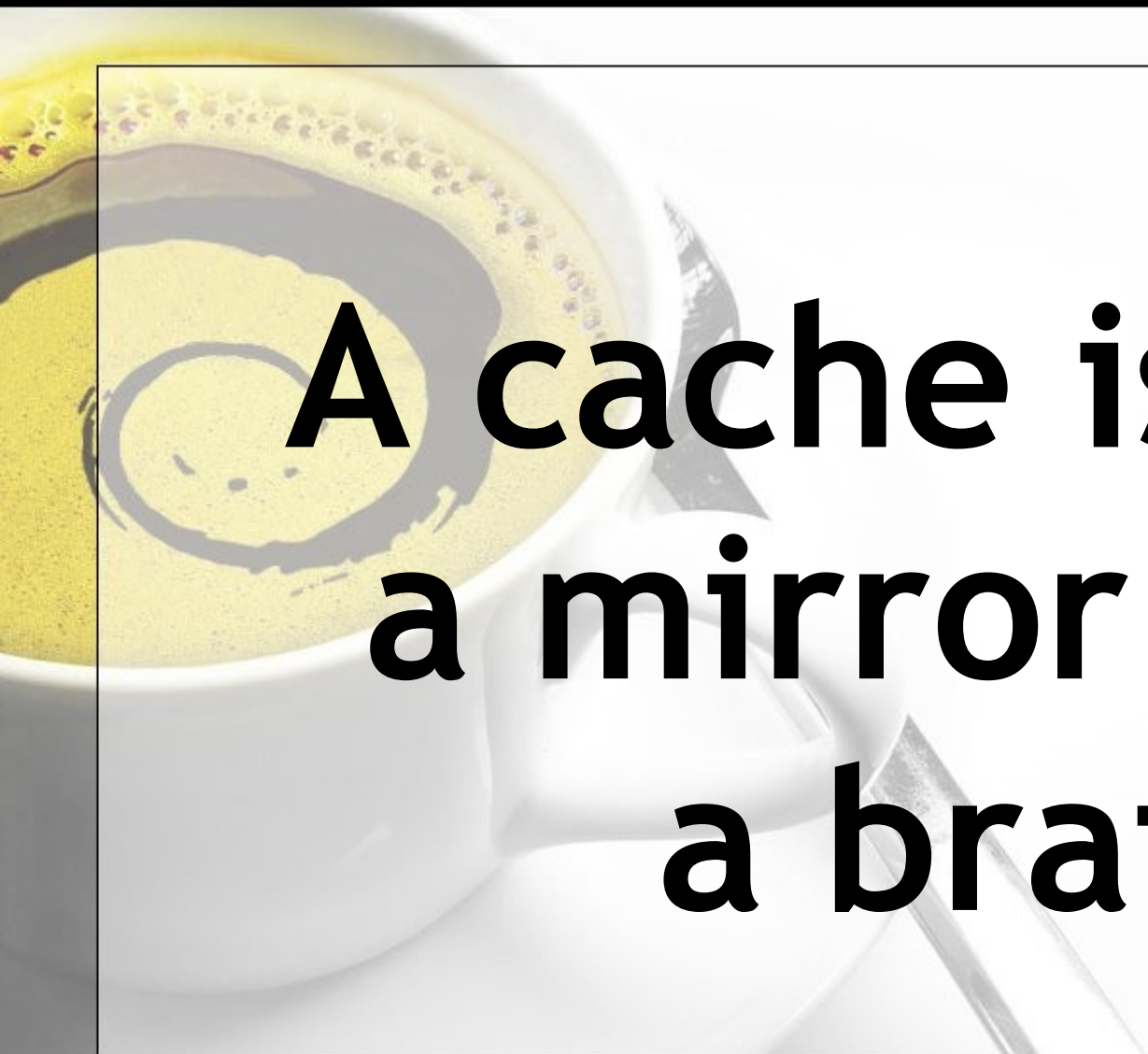




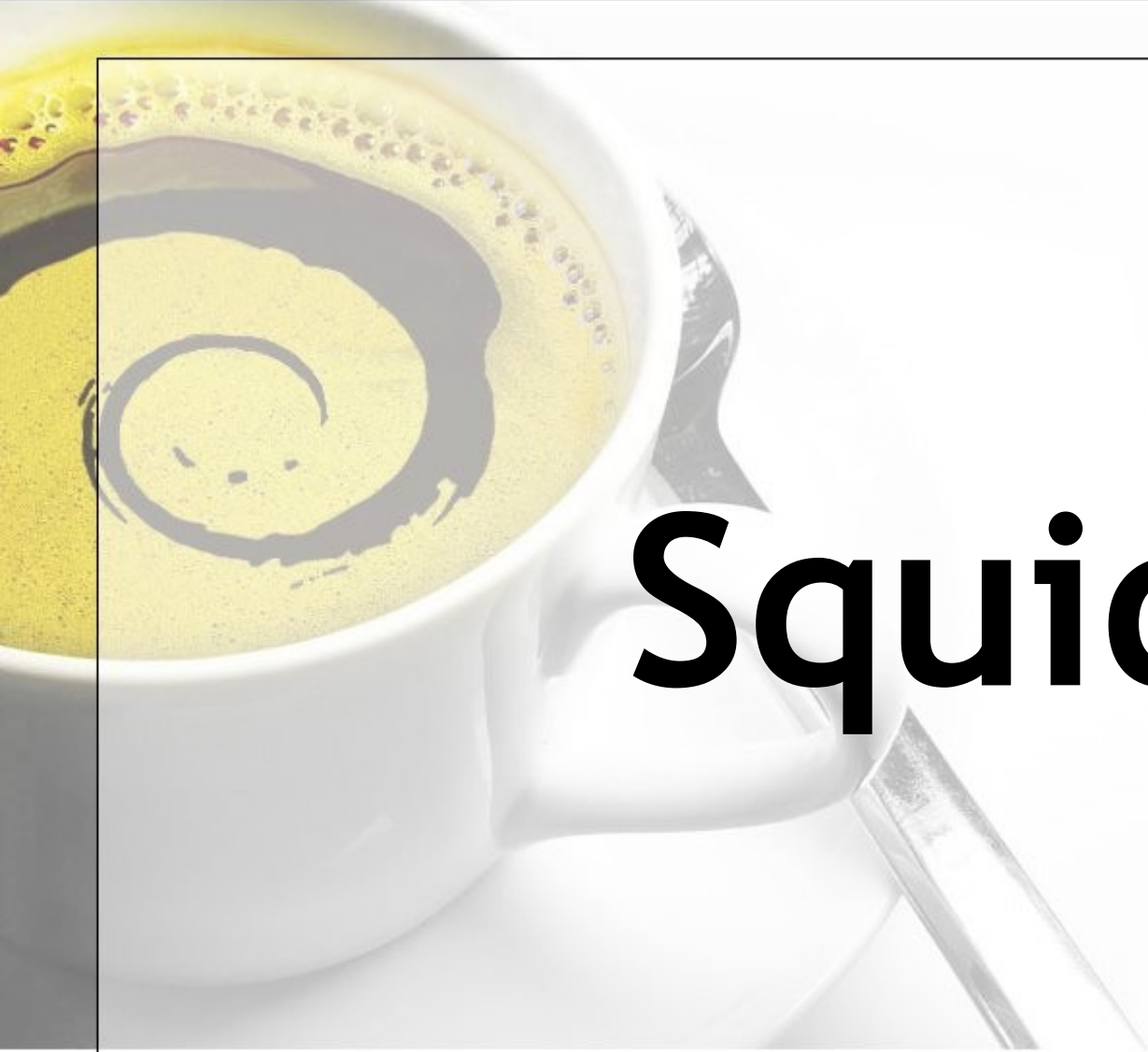
**The moral:
don't download
what you
don't need**

A white ceramic coffee cup filled with a latte, topped with a layer of foam. A silver spoon rests on the white saucer. The background is a soft, out-of-focus light color.

**Mirror bad:
cache good!**



**A cache is like
a mirror with
a brain**



Squid?



:- (Not tuned for large files.

:- (Cache expiry algorithm doesn't understand DPKG.



apt-proxy
apt-cacher
approx



approx

Written in Ocaml.

Runs as a daemon.

Forces clients to use specified repos.

Setting Up Approx

1. Install on server:

```
# aptitude install approx
```

2. Edit `/etc/approx/approx.conf` to set repo locations such as:

```
debian    http://ftp.au.debian.org
```

```
non-US    http://ftp.au.debian.org/non-US
```

```
security http://security.debian.org/debian-security
```

3. Edit `/etc/sources.list` on clients:

```
deb http://mycache:9999/debian stable main contrib
```

```
deb http://mycache:9999/security stable main
```

Logging goes in `/var/log/apt-proxy.log`



apt-proxy

Written in Python.

Runs as a daemon.

Forces clients to use specified repos.

v2 is a complete rewrite.

Setting Up Apt-proxy

1. Install on server:

```
# aptitude install apt-proxy
```

2. Edit /etc/apt-proxy/apt-proxy-v2.conf to set backend sources such as:

```
[debian]  
backends =  
    http://ftp.au.debian.org  
    http://ftp.us.debian.org
```

3. Edit /etc/sources.list on clients:

```
deb http://mycache:9999/debian stable main contrib  
deb http://mycache:9999/security stable main
```

Logging goes in /var/log/syslog



apt-cacher

Written in Perl.

Runs as a daemon or CGI.

**Clients can use any repo they like.
Logging; reporting; access control;
upstream proxy (with auth); client
advisories in return value.**

Setting Up Apt-cacher

1. Install on server:

```
# aptitude install apt-cacher
```

2. Edit /etc/apt-cacher/apt-cacher.conf to set options such as:

```
allowed_hosts=192.168.0.0/24  
admin_email = webmaster@example.com
```

3. Edit /etc/sources.list on clients:

```
deb http://cache/apt-cacher/ftp.debian.org \  
    stable main contrib
```

Logging goes in /var/log/apt-cacher/[access|error].log



These slides are online now at:
<http://jon.oxer.com.au/talks/>

Image by Andre Gonzalez:
<http://nostromo2k3.deviantart.com/gallery/>

debian