OpenSourceWatershed

"watching software flow downstream"

http://oswatershed.org

Scott Shawcroft
April 25, 2009

Scott Shawcroft

- Last Quarter at University of Washington
- Computer Engineer
- Internships: Creative Commons and Google
- OS Projects: touchd, Menzies, Annoamp, denu
- Linux since spring '04.
- LFS -> Gentoo -> Ubuntu

Idea

Upstream software is better.

How can distribution developers work closer with upstream?

How can upstream developers work closer with downstream?

What to study

Which distributions are closer to upstream and why?

Which upstream packages are closer to downstream and why?

What to study

- Concretely determine which distribution is freshest.
- Freshest implies minimal lag behind upstream.
- Lag is the time between “now” and the oldest new release.

Subversion Lag Example

On 03/23/09 gentoo stable is at 1.5.5.

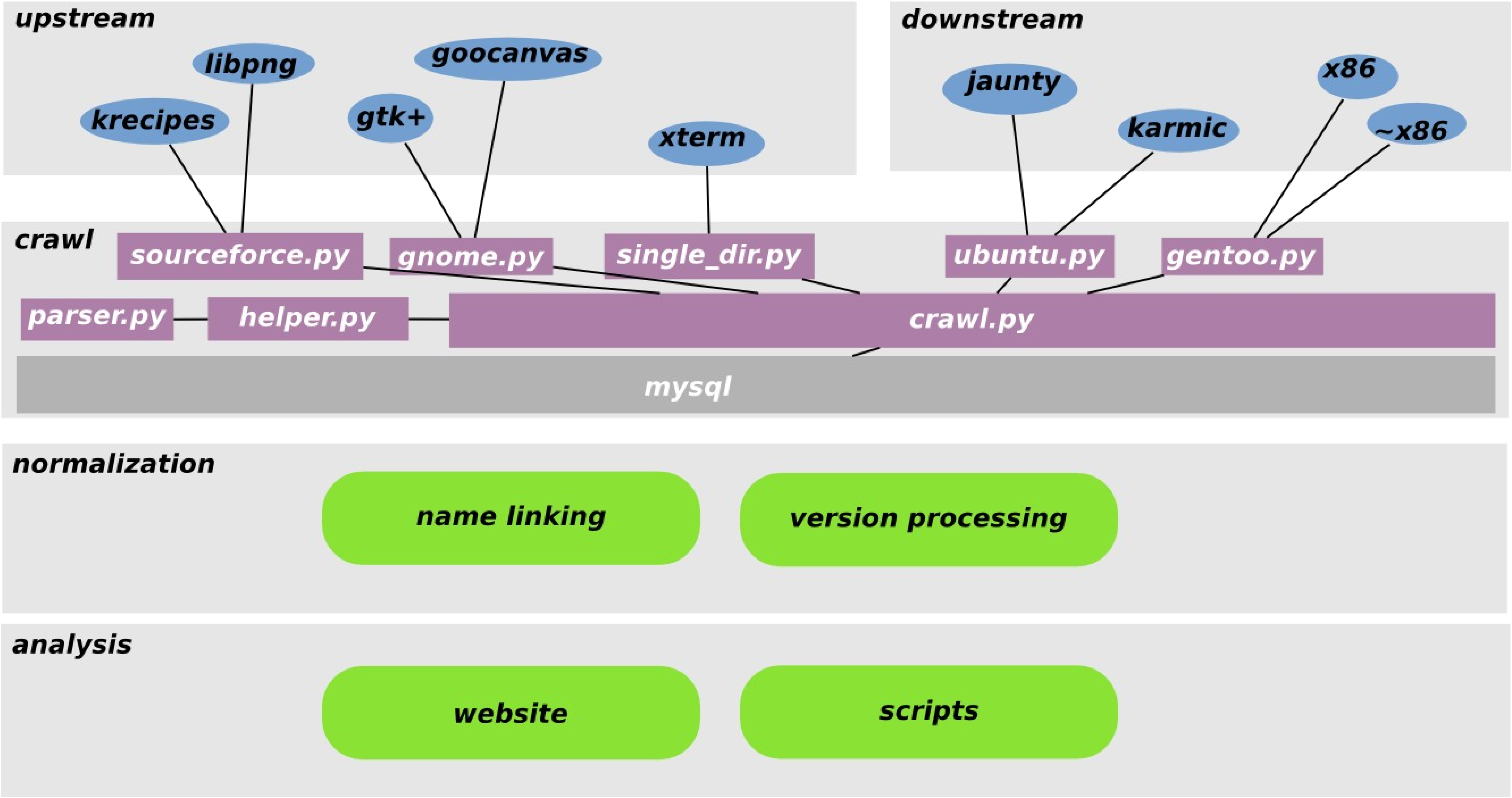
Then the lag is 3/23/09-2/27/09, the time between now and the oldest new release.

Thus, the lag is 24 days.

04/19/09	1.6.1
03/20/09	1.6.0
02/27/09	1.5.6
12/22/08	1.5.5
10/24/08	1.5.4

Subversion Lag Graph

Overview



upstream
libpng
goocanvas
krecipes
gtk+
xterm
downstream
jaunty
x86
karmic
~x86
crawl
sourceforce.py
gnome.py
single_dir.py
ubuntu.py
gentoo.py
parser.py
helper.py
crawl.py
mysql
normalization
name linking
version processing
analysis
website
scripts

Data Gathered

Gather release information from upstream and downstream.

- **Name**
- **Version**
- **Date**
- Revision
- Extra

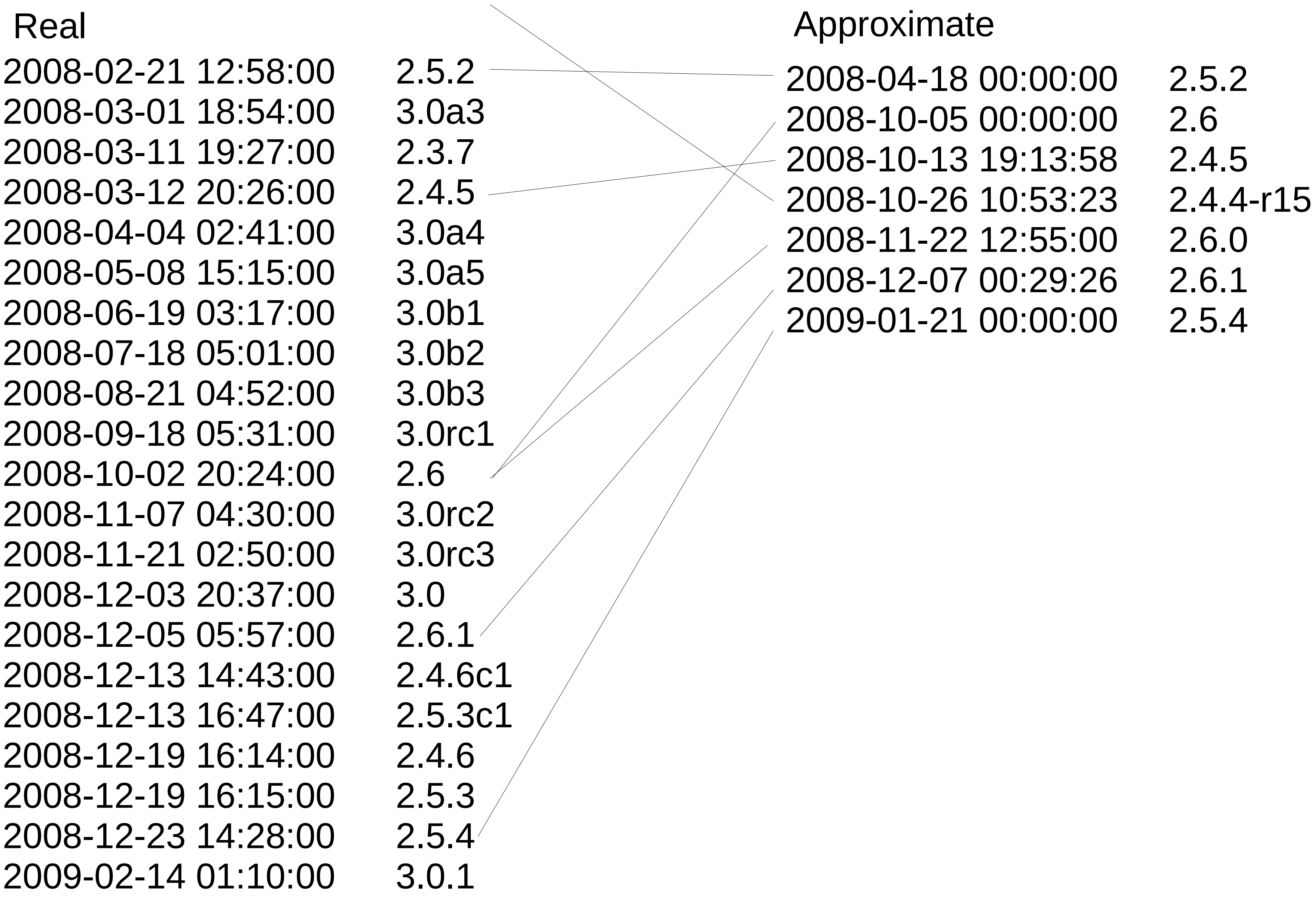
Computing Lag

Computing the lag is a trivial task in a perfect world. In the real world it comes down to two things:

- Approximating missing data
- Understanding the data
  - Package name mangling
  - Version branching

Approximating Upstream

Make a best guess based on its first appearance in any distribution.



Real

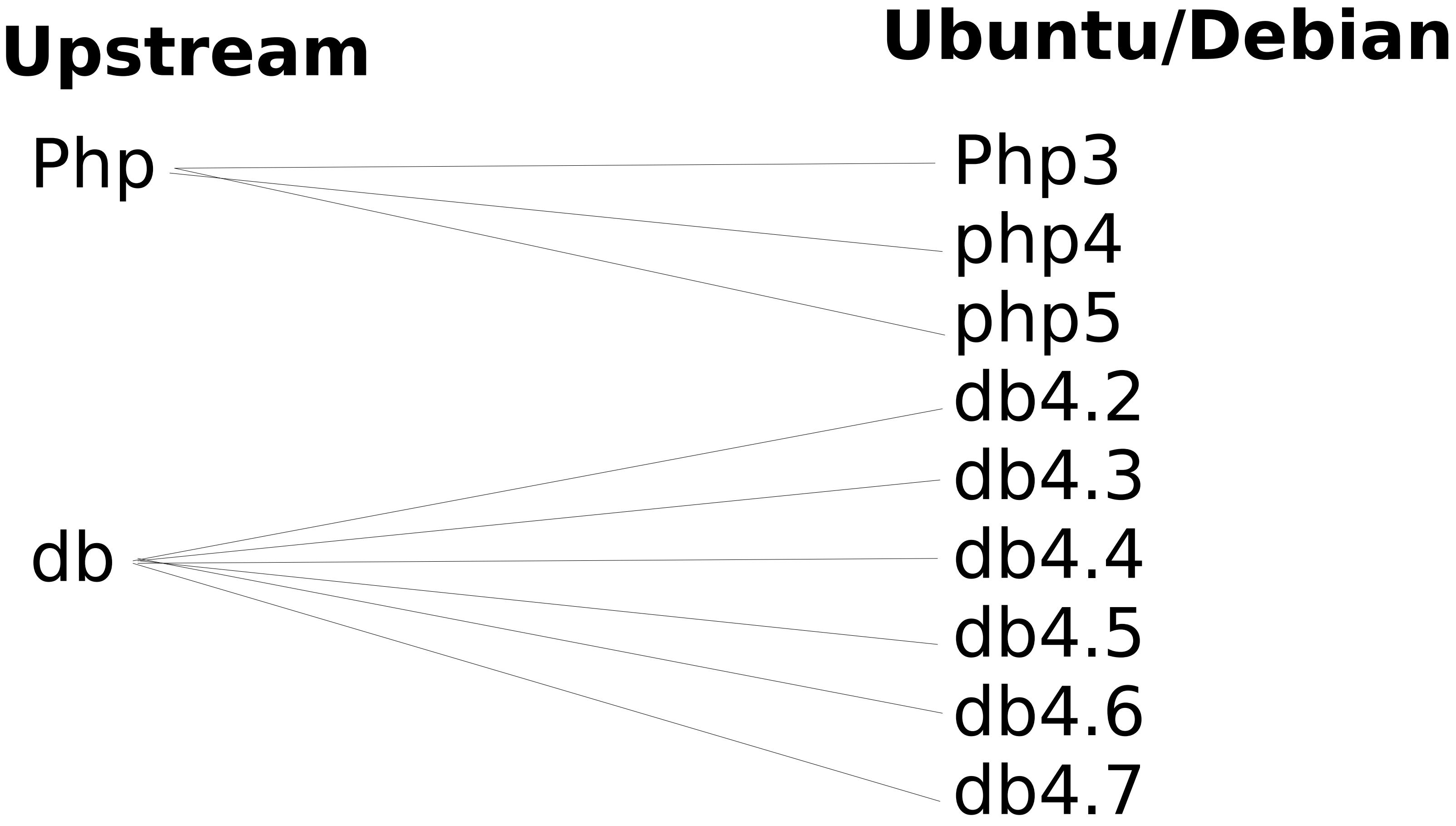
2008-02-21 12:58:00	2.5.2
2008-03-01 18:54:00	3.0a3
2008-03-11 19:27:00	2.3.7
2008-03-12 20:26:00	2.4.5
2008-04-04 02:41:00	3.0a4
2008-05-08 15:15:00	3.0a5
2008-06-19 03:17:00	3.0b1
2008-07-18 05:01:00	3.0b2
2008-08-21 04:52:00	3.0b3
2008-09-18 05:31:00	3.0rc1
2008-10-02 20:24:00	2.6
2008-11-07 04:30:00	3.0rc2
2008-11-21 02:50:00	3.0rc3
2008-12-03 20:37:00	3.0
2008-12-05 05:57:00	2.6.1
2008-12-13 14:43:00	2.4.6c1
2008-12-13 16:47:00	2.5.3c1
2008-12-19 16:14:00	2.4.6
2008-12-19 16:15:00	2.5.3
2008-12-23 14:28:00	2.5.4
2009-02-14 01:10:00	3.0.1

Approximate

2008-04-18 00:00:00	2.5.2
2008-10-05 00:00:00	2.6
2008-10-13 19:13:58	2.4.5
2008-10-26 10:53:23	2.4.4-r15
2008-11-22 12:55:00	2.6.0
2008-12-07 00:29:26	2.6.1
2009-01-21 00:00:00	2.5.4

Normalizing Names

Distros must also deal with package branches. Gentoo uses 'slotting', most use new package names.



Upstream

Ubuntu/Debian

Php

Php3

php4

php5

db

db4.2

db4.3

db4.4

db4.5

db4.6

db4.7

Ordering Versions

Original ordering based only on release date.

Problems:

- All new releases obsolete old ones.
- Any new downstream release that doesn't match an upstream release is completely fresh.

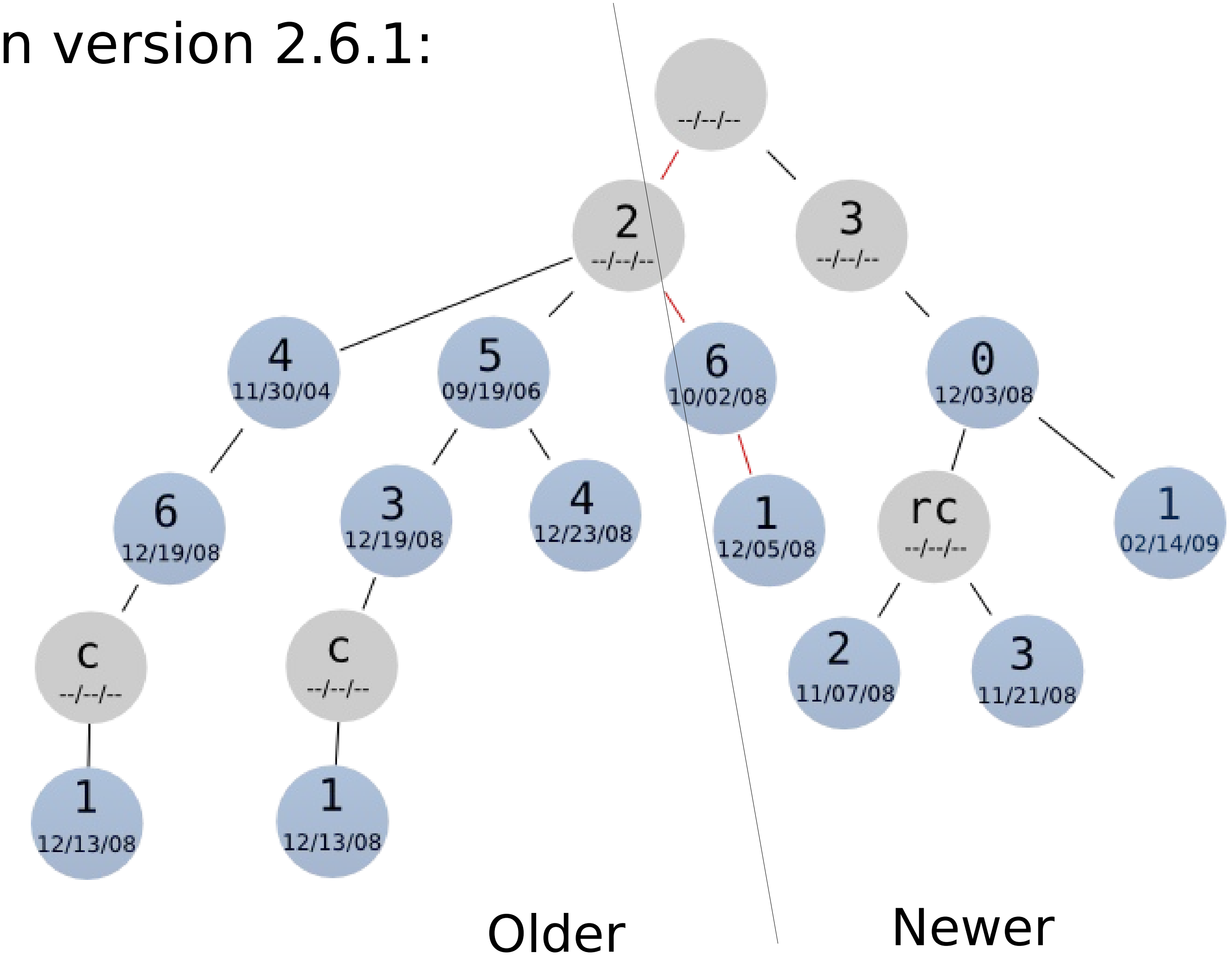
2008-10-02 20:24:00	2.6
2008-11-07 04:30:00	3.0rc2
2008-11-21 02:50:00	3.0rc3
2008-12-03 20:37:00	3.0
2008-12-05 05:57:00	2.6.1
2008-12-13 14:43:00	2.4.6c1
2008-12-13 16:47:00	2.5.3c1
2008-12-19 16:14:00	2.4.6
2008-12-19 16:15:00	2.5.3
2008-12-23 14:28:00	2.5.4
2009-02-14 01:10:00	3.0.1

Should we obsolete 2.6.1 with 2.4.6?

Ordering Versions

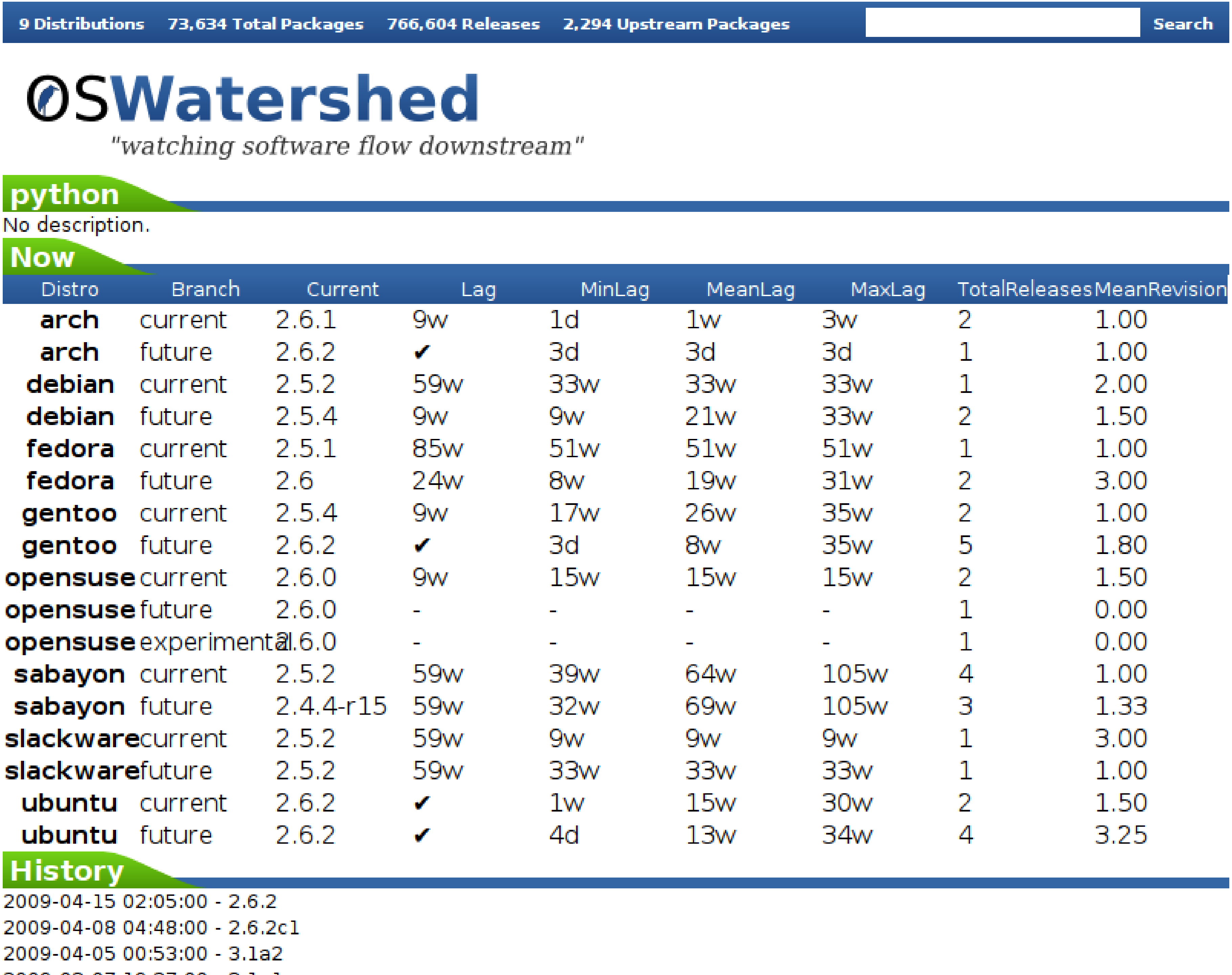
1) Split the version.
2) Build a tree with children sorted by release date.

For Python version 2.6.1:



Results ~ Python

http://oswatershed.org/pkg/python



python

No description.

Now

Distro	Branch	Current	Lag	MinLag	MeanLag	MaxLag	TotalReleases	MeanRevision
arch	current	2.6.1	9w	1d	1w	3w	2	1.00
arch	future	2.6.2	✔	3d	3d	3d	1	1.00
debian	current	2.5.2	59w	33w	33w	33w	1	2.00
debian	future	2.5.4	9w	9w	21w	33w	2	1.50
fedora	current	2.5.1	85w	51w	51w	51w	1	1.00
fedora	future	2.6	24w	8w	19w	31w	2	3.00
gentoo	current	2.5.4	9w	17w	26w	35w	2	1.00
gentoo	future	2.6.2	✔	3d	8w	35w	5	1.80
opensuse	current	2.6.0	9w	15w	15w	15w	2	1.50
opensuse	future	2.6.0	-	-	-	-	1	0.00
opensuse	experimental	2.6.0	-	-	-	-	1	0.00
sabayon	current	2.5.2	59w	39w	64w	105w	4	1.00
sabayon	future	2.4.4-r15	59w	32w	69w	105w	3	1.33
slackware	current	2.5.2	59w	9w	9w	9w	1	3.00
slackware	future	2.5.2	59w	33w	33w	33w	1	1.00
ubuntu	current	2.6.2	✔	1w	15w	30w	2	1.50
ubuntu	future	2.6.2	✔	4d	13w	34w	4	3.25

History

2009-04-15 02:05:00 - 2.6.2
2009-04-08 04:48:00 - 2.6.2c1
2009-04-05 00:53:00 - 3.1a2

Picking Packages for Analysis

- Real complete upstream data
- Present in every distribution
- No two versions share a release date

* Analysis in this presentation done with 137 “good” packages unless otherwise stated.

Presentation Packages

- a2ps
- acpid
- alsa-utils
- aspell
- aspell-en
- atk
- audacious
- autoconf
- automake
- bash
- bc
- binutils
- bison
- blackbox
- bridge-utils
- cairo
- ccache
- cdrdao
- clisp
- compiz
- coreutils
- cpio
- cscope
- curl
- cvs
- desktop-file-utils
- diffstat
- dirmngr
- dosfstools
- e2fsprogs
- ed
- emacs
- enscript
- esound
- ethtool
- expat
- file
- findutils
- flac
- flex
- fontconfig
- gawk
- gcc
- gdb
- gettext
- gimp
- git
- gkrellm
- gnome-icon-theme
- gnuchess
- gnuplot
- gperf
- gqview
- grep
- groff
- gucharmap
- gv
- gzip
- hal
- hdparm
- hicolor-icon-theme
- hplip
- icon-naming-utils
- imagemagick
- indent
- intltool
- iptables
- iptraf
- joe
- kbd
- less
- lftp
- libpng
- libtool
- libxml2
- m4
- mailx
- make
- mc
- mdadm
- mercurial
- module-init-tools
- mtr
- mtx
- mutt
- nano
- nasm
- netpbm
- nmap
- ntfsprogs
- obexftp
- openssh
- openssl
- openvpn
- pan
- parted
- patch
- pciutils
- pcmciautils
- perl
- pidgin
- pm-utils
- ppp
- python
- rcs
- rdesktop
- rsync
- samba
- scim
- scim-m17n
- screen
- sdparm
- sed
- shared-mime-info
- sharutils
- slrn
- smartmontools
- sox
- strace
- subversion
- sudo
- sysfsutils
- syslinux
- sysvinit
- tango-icon-theme
- tar
- tcpdump
- tcsh
- texinfo
- tin
- udev
- usbutils
- vorbis-tools
- wget
- xchat
- xcompmgr
- xterm

Stable Distros Over Time

Younger distributions are generally fresher.

🟦 *Arch*
🟥 *Debian*
🟦 *Fedora*
🟪 *Funtoo*
🟪 *Gentoo*
🟩 *OpenSuse*
🟦 *Sabayon*
🟦 *Slackware*
🟧 *Ubuntu*

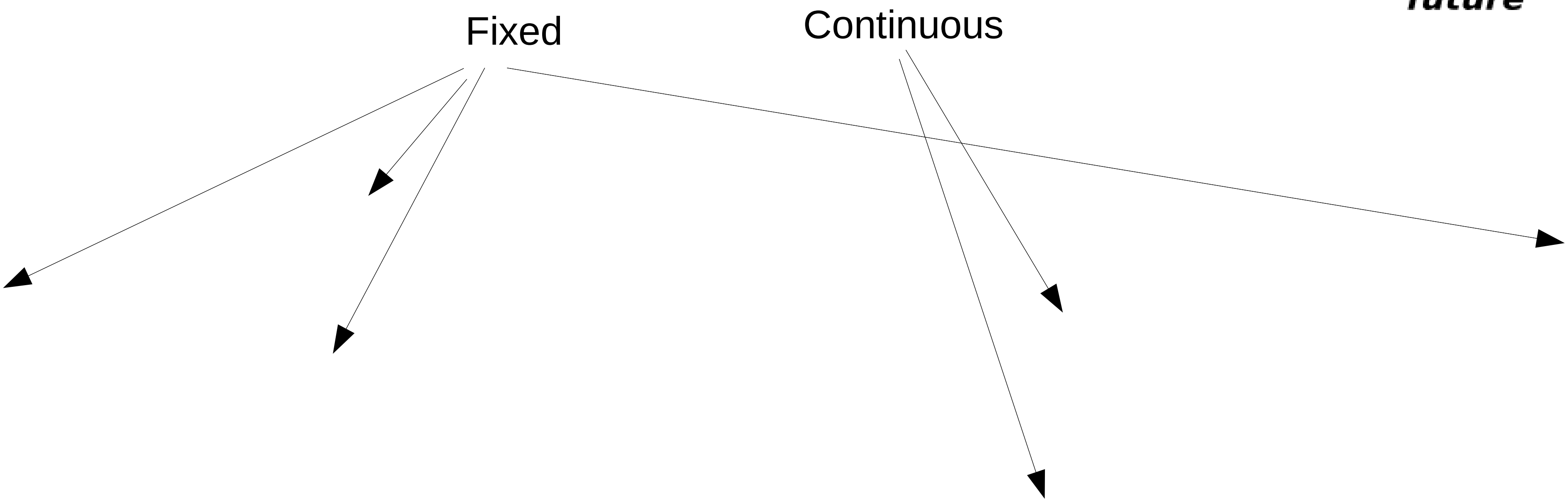
– – *past*
— *current*
⋯⋯ *future*

Stable Distros Over Time

Continuously released distros are fresher than fixed release cycle distros.

- 🟦 *Arch*
- 🟥 *Debian*
- 🟦 *Fedora*
- 🟪 *Funtoo*
- 🟪 *Gentoo*
- 🟩 *OpenSuse*
- 🟦 *Sabayon*
- 🟦 *Slackware*
- 🟧 *Ubuntu*

- – – *past*
- —— *current*
- .... *future*



Fixed

Continuous

Ubuntu vs. Gentoo

- Debian based
- Binary Distribution
- dpkg
- 6 Month Release Cycle
- Started October 20, 2004

- Source Distribution
- Portage
- Continuous Release Cycle
- Started March 31, 2002

Ubuntu vs. Gentoo

Data Problems

70k+ package names for 10k-20k packages.

“good” does not imply truly good.

“stable”

Distribution comparison grounded in a common notion of “stable”.

- Quantitative notion of stability.
- Common distribution developers notion of stability.
- Common user notion of stability.

Future Work

More of the same.

Increase the pool of “good” packages.

Create more methods of analysis.

Gather more data.

Gather data from more sources.

Goals

- Find and establish best packaging practices, upstream and downstream.
- Central source for distro package status.
- Aid in distro selection.
- Reduce lag, focus development upstream.

Keep distros fresh,
with newer features
and greater security.

Links

- scott.shawcroft @ gmail.com
- oswatershed.org
- trac.oswatershed.org
- github.com/tannewt/open-source-watershed

OpenSource**Watershed**

"watching software flow downstream"