

Athena History

Modular Debathena

Debian Packages

An example diversion

Other Athena customizations

Athena

- ▶ Project Athena started at MIT in 1983
- ▶ grant from IBM and Digital

Mission statement:

By 1988, create a new educational computing environment at MIT built around high performance graphics workstations, high speed networking and servers of various types

Athena

Technical objectives:

1. design and implement the new computing environment with the following properties:
 - 1.1 provides computing resources needed to support educational uses at MIT
 - 1.2 accomodates heterogeneous hardware
 - 1.3 provides users and programmers with machine independent interfaces
 - 1.4 maximizes exportability and importability of software
 - 1.5 extends so that by 1990 each student can be provided with a workstation at a total system cost of about 10% of MIT tuition
2. foster innovative projects by the MIT faculty that demonstrate the educational value of the new computing environment
3. put in place and operate a system of about 1500 workstations for use by the MIT faculty, students and staff.¹

¹Berkeley UNIX on 1000 workstations: Athena changes to 4.3BSD

Athena Software

Bringing graphical(!) computing to the masses required a lot of new software:

X Window System, Kerberos, Hesiod, discuss, zephyr, athena widgets, remote virtual disk, olh, larvnet, athinfo.

Also use other software: NFS, AFS.

Make third-party software available to users (e.g. MACSYMA).

Consistent experience across multiple platforms

Project Athena was created simultaneously for microVAX and RT-PC.

Over the years, Athena on a great many other machines, usually with at least two different classes of machine in the offering at any given time: VAXstation, DECstation, RS/6000, various SPARC/Ultra, SGI Indy/O2, SunBlade/SunFire, Sun Netra, RHEL Linux, NetBSD, Debian/Ubuntu.

Athena today

- ▶ Infrastructure (mail, AFS, moira, LDAP, Kerberos, . . .)
- ▶ IS&T-run dialup pool (Ubuntu Lucid, 64-bit)
- ▶ SIPB-run dialup (Debian lenny, 32-bit)
- ▶ Public cluster machines (475)
- ▶ Auto-upgraded (private) workstations (270)
- ▶ Other privately-maintained machines (ca. 450)

Athena History

Modular Debathena

Debian Packages
An example diversion

Other Athena customizations

As much Athena as you want ... or as little

Traditional Athena was monolithic: one set of packages, thoroughly tested, with only a handful of toggles available to workstation owners (ISPUBLIC, mksrv).

Debathena splits Athena into about 150 different packages — install just debathena-kerberos-config, or get the full Athena experience with debathena-cluster.

With this great flexibility; choice can be hard — Metapackages:

- ▶ debathena-login
- ▶ debathena-login-graphical
- ▶ debathena-workstation
- ▶ debathena-cluster

Dependencies

Metapackages can be just “equivs packages”, which just depend on other packages and have no other content. (We are moving away from this, though, as it limits our ability to support multiple Debian/Ubuntu releases simultaneously.)

Other Debathena packages depend on each other (e.g. debathena-alpine-config).

Sometimes the dependencies aren't quite right (AppArmor).

Athena History

Modular Debathena

Debian Packages
An example diversion

Other Athena customizations

Anatomy of a Debian package

- ▶ **debian/changelog** List of changes; authoritative source of version number. Installed to `/usr/share/doc/package/changelog.Debian.gz+`
- ▶ **debian/compat** Just a numer, but signifies the debhelper compatibility level to indicate what features are used/needed/available
- ▶ **debian/control** The Debian-specific metadata for the package. Much of the required components of a FreeBSD port Makefile goes here.
- ▶ **debian/copyright** Debian does have its license lawyers. . .
- ▶ **debian/rules** Essentially, a Makefile. Build the package.

debian/control

```
Source: python-moira
Section: python
Priority: extra
Maintainer: Evan Broder <broder@mit.edu>
Build-Depends: cdb (>= 0.4.43), debhelper (>= 5.0.37.2), p
Standards-Version: 3.8.4
```

```
Package: python-moira
Architecture: any
Depends: ${shlibs:Depends}, ${misc:Depends}, ${python:Depen
Provides: ${python:Provides}
Description: Python bindings for Moira
    PyMoira provides a set of Python bindings for Moira, the A
    Service Management system.
```

How does Athena use maintainer scripts

Four maintainer scripts: {pre,post}{inst,rm}

Of our ~150 packages:

- ▶ postinst: 47 packages
add users, /etc/services, rc script games, gconf, printers
- ▶ postrm: 14 packages
users, rc scripts
- ▶ preinst: 11 packages
groups, rcscripts, and configuration files
- ▶ prerm: 21 packages
update-inetd, PAM stack updates, printers, users/sudoers

Diversions

Debathena makes very heavy use of diversions, a dpkg feature not used many other places.

<http://debathena.mit.edu/config-package-dev/>
DEB_DIVERT_FILES, DEB_REMOVE_FILES,
DEB_TRANSFORM_FILES, DEB_TRANSFORM_SCRIPT,
DEB_CHECK_FILES_SOURCE, DEB_UNDIVERT_FILES,
DEB_UNREMOVE_FILES

Why divert?

Suppose that you want to prevent unprivileged users from being able to spam other users terminals users with text. One way that a use could do this is by using the syslog function to send a message at loglevel LOG_EMERG to the system log (CVS sometimes does this when it crashes)). One way to disable this feature is to comment the line in `/etc/syslog.conf` of the form:

```
*.emerg *
```

One can implement a `config-package-dev` package to implement this change by transforming `syslog.conf`.

Provides/conflicts

If we want to replace a file, we need to conflict with anything else that might want to have it, and provide our own version:

```
Source: debathena-transform-example
```

```
Section: config
```

```
[...]
```

```
Package: debathena-transform-example
```

```
Architecture: all
```

```
Depends: cdb, ${misc:Depends}, elinks
```

```
Provides: $diverted-files
```

```
Conflicts: $diverted-files
```

```
Description: Example config-package-dev package
```

```
This is an example config-package-dev package.
```


debian/rules

```
#!/usr/bin/make -f

DEB_DIVERT_EXTENSION = .debathena
DEB_TRANSFORM_FILES_debathena-transform-example += \
/etc/syslog.conf.debathena

include /usr/share/cdb/1/rules/debhelper.mk
include /usr/share/cdb/1/rules/config-package.mk
```

The transformation script

Use whatever language you want, even perl: more rope for foot-shooting than you could possibly want!

```
debian/transform_syslog.conf.debathena:
```

```
#!/usr/bin/perl -0p  
s/^(\\*\\.emerg\\.\\*)/# $1/m or die;
```

Use of diversions

- ▶ supply our own AFS configuration: CellServDB, etc.
- ▶ Tell LDAP (and others) to use the MIT CA for x.509 certificates
- ▶ Add nss_nonlocal use in nsswitch.conf
- ▶ All sorts of stuff to control the configuration of cluster machines
- ▶ Stop AppArmor from being dumb with AFS homedirs (and more)
- ▶ ...

Athena History

Modular Debathena

Debian Packages
An example diversion

Other Athena customizations

Remotely administer clusters

- ▶ Cluster workstations are not like servers in a data facility!
- ▶ Individually visiting 500 workstations all over campus is not practical for maintenance. (Nor is a standard remote console setup.)
- ▶ We need machines to stay current, get updates that patch vulnerabilities, etc..

Automatically take updates, both from upstream Debian/Ubuntu and from our debathena repository.

Auto-update

- ▶ 30 machines hitting an apt repo at the same time over a shared 10Mbit link doesn't end well
- ▶ Don't want all machines in a cluster updating leaving none free for users
- ▶ Don't boot a user just to take updates
- ▶ read machine-specific cluster information from hesiod

athena-auto-update, a shell script run from cron

The atpocalypse

1. Desynchronization implemented with at(1).
2. Upstream updates at
3. new at maintainer script kills all at jobs before updating
4. Go run around to all 500 machines and fix them by hand

Recovery Hook

- ▶ Don't want to have to hit 500 machines by hand again!
- ▶ As part of auto-update, fetch (over SSL) a shell script from a known URL.
- ▶ Gives a lot of flexibility for when upstream throws a curveball
- ▶ Also, be sure to have machines in development repos **and actually use them so that testing works**

Questions

(Note to self: athinfo is interesting, but not packaging)

(Other note to self AFS is neat, but ports/152467 already exists)