Administrating Unix/Linux Systems in Server and Teaching Environments

Alexander Zangerl

Bond University

az@{bond.edu.au,debian.org}

What’s this all about?

- about choosing *your* Unix platform
- They all suck!
- ...some suck more, some less
- No "one size fits all" Unix
- choice depends on what you want to do with it
  - this talk is about my experiences
  - which may or may not be applicable to your situation

Who am I?

- system programmer-turned-sysadmin-turned-lecturer
  - now shocking students at Bond University, QLD
  - and administering a couple of servers and labs
- Linux bigot? maybe...
- MS opponent, more than likely,
- but - most important - a fan of all things Unix!

What to expect of this talk

some information for making your own judgment about which Unix flavour might suit your needs best.

- History of Unix at Bond
- Linux enters the stage
  - Which Linux distro?
  - Suse, Redhat, Debian
- Which Unix for which application?
  - student labs
  - teaching backends
  - software development servers
- Subjective comparison of the systems I’ve dealt with

#insert <disclaimer.h>

- I’m a Debian Developer, one of >1000 volunteers
- Debian *is* my personal favourite
- This is not to be a Debian advertisment show...
  - but I can’t deny my bias.
- I’m covering only systems I’ve had to work with recently
  - Solaris and Linux, but no AIX, HP-UX and no *BSD

Ancient History of UNIX at Bond

- mostly a MS shop
- until about 4 years ago:
  - UNIX confined to the backbone services
  - eg. email, DNS; on BSD, AIX, Solaris
- IT school had one Solaris box for teaching Java, C subjects
  - everything else done on Windows boxes

Bond is a Special Place

- commercial uni
- Multiple entities run infrastructure
  - ITS runs central things: network, email, DNS, most labs
  - IT school runs some servers for IT school only
  - (some) IT lecturers run their own servers
- but all in all things work remarkably well for such a setup!

shell.it.bond.edu.au

- the one Solaris box of the IT school
  - used for teaching Java, shell, C programming
  - drove Xterms, ran Email, NFS, Samba, X11, you name it...
- unpatched system, admin overworked, eventually RIP
- with students having shell accounts
- ...and CDE, all kinds of compilers plus too many bad ideas
- net result: very unstable, security sieve

Enter the Unix Guerilla

- 2000: a new lecturer brought in new ideas and services
- ITS, IT both not interested in supporting
  - then let’s roll our own services!
- later other Unix adepts join Bond IT, me included
- a general swing of interests towards Unix started

Let’s see: where do we want Unix?

(everywhere, of course!)

- Teaching/Research backend servers
- Student-accessible servers
- Student labs

Eventually we got some Unix into these areas, and the future looks good.

Beginnings of the Alternate Server Farm

- started out with a few surplus computers
- all different
- only commonality: slow and hardware way beyond EOL
- originally running Suse Linux
- mainly because of the lecturer’s experience with it

Backend Server Issues

- Stability most important
- Continuity provisions to ensure smooth future
- Administration (mostly) done by lecturers
  - Ease of administration important
  - Good automation support
  - Cooperative admin must be easy
  - Can’t affort to waste much time

Services Offered

- `http://james.bond.edu.au/`
  - homegrown teaching portal
  - combining all teaching-related services
  - testbed for *lots* of new ideas
- db and application servers
  - for TopicMap (knowledge engineering) research

Student Programming Server

- Foremost you want a consistent environment
  - something that makes sense to a newcomer
  - eg. file locations logical, software properly integrated
  - after all we try to teach Best Current Practice, not bad hunt-and-patch examples
- Then you need a fair amount of security and confinement
  - 70% of students being taught `fork()` reinvent the "fork-bomb" immediately.
- And finally you need robustness and sufficient performance

Solaris as a programming server?

- That’s what we had (and still have, to some extent).
- shortly after I joined I rebuilt `shell` from scratch
- didn’t choose Linux then:
  - kernel stability on UltraSparc (E250) unknown
  - availability of native, Sun-supported Java?
  - maturity of RAID software?
- One question was also how much time I’d have to play with this system later on.
- So we went for stability and the thing we knew.

Solaris

- But Solaris certainly doesn't make me happy:
  - The software quality is abysmal. Given an unknown `$TERM`, Sun's `vi` still dumps core.
  - to make a Solaris system bearable, you have to rip out 90% of the userland and replace it with (other|GNU) stuff.
  - and then, consistency? not exactly a highlight.
  - Programming on Solaris is not much fun: read any INSTALL document or `autoconf` script to see which systems need most workarounds and special care.
- all in all, not a good choice of platform for introducing new students into the magic of efficient programming or Unix.
- but it's fairly stable and not *too* bad performance-wise.

Linux in the Labs

- developed new subjects with more Unix focus
- eg. "System Security", "Unix Administration", "Internet Tech"
- new requirements for the computer labs:
  - hands-on experience with administration of Unix systems
  - destructive administration, too!
  - security exercises, vulnerability checks etc.
  - just not doable in "normal" Windows labs
  - nor on any of the existing student-accessible servers

Linux Lab to the Rescue!

- Linux because of
  - our preferences
  - availability of i386 boxes
  - open-source nature - "Look, ma, no license fee!"
- small lab, quite experimental
- setup in a firewalled environment
  - students can admin their own systems
  - but without adverse effects on main infrastructure

Linux Lab

- in the very beginning perceived by ITS, IT school as an experiment
  - completely handled by the Unix Guerilla Guys
  - so we implemented our preferences
- Firewall running Debian Linux from the beginning
- lab computers initially running Suse
- systems to be set up by students themselves
- very ad-hoc, not useful for non-admin subjects
- also limiting possible other uses of that lab

Lab Issues

- ad-hoc setup: no good for subjects needing ready-made systems
- what about reusing installations after semesters?
- needed either possibility to re-synchronise systems for reuse
- or efficient rollout of multiple boxes

Debian offered all necessary features:

- automated rollout via FAI plus cfengine
- good support for remote and bulk administration
- also BCP in consistency and distro design

Environment grows more mature

- Linux lab goes mainstream
  - more lecturers start to use it
  - added central auth environment, automated rollout
- Other people are starting to use Linux
- ITS switches to Redhat on some essential servers
  - Uni hardware acquisition policy now includes ‘must work with Linux’
- `james` portal needed growing server farm
- administration of multiple systems became more problematic
- eventual cleanup of homegrown stuff and switch to Debian

Debian changeover

- Suse, Redhat systems without maintenance contracts
- often recent software versions unavailable
- or bug-fixes n/a
- result: lots of locally-compiled, homegrown software installed
- often subtly different between machines, nightmare to migrate services

Benefits from changing to Debian:

- more software available, less local hacks needed
- all software installable via Internet
- security updates available, early and for everybody
- good support to roll out localised packages if needed

Current State of Affairs

- IT school alternate server environment:
  - a backup system on a Sparc
  - `james` web portal and db backend on i386
  - a small research cluster on UltraSparcs
  - firewall and auth system on i386

  all running Debian
- Linux Lab:
  - running Debian (most of the time)
- IT school main environment:
  - programming server, Solaris on UltraSparc
  - a couple of Redhat Linux boxes

The Comparison

I warned you about this being subjective!

- Solaris
- Suse and Redhat Linux
- Debian Linux

Solaris

- Administration
  - Package mgmt is not bad
  - but Sun doesn't stick to it, e.g. some Java stuff comes in `tar`balls or as zipfile with custom installer
  - Free software integration not great
  - Patches are no fun, often quite late
- Use
  - too inconsistent for students, eg. goodies in `/usr/ucb`
  - Basic offerings marginal, doesn't even include compiler anymore
  - lots of 3rd party software needed to make it useful
- Our main reason for Solaris: Java

Redhat and Suse Linux

- end-user friendly, but not admin-friendly
  - too much eye candy hides the essential concepts
- hard to administer without the (GUI) admin tools
  - contra-intuitive tools (I hate Suse's `yast`.)
  - manual configuration changes easily lost
- automation and bulk administration support lacking
- no guiding policy behind software integration
- less software packaged than for Debian
- lack of continuity biggest problem for server-use
  - updates after release EOL?
  - free versions restricted wrt. update availability
  - basically no upgrade path for customised environment

Debian Linux

- less foolproof installation
  - less eye-candy, more function
- consistent environment
  - all software must meet integration criteria
- flexible and easy to admin
  - automation, remote or central admin well-supported
- available for 11+ hardware platforms
- continuity
  - continuous updates if desired
  - but configuration is always preserved
- stability *and* security
  - security fixes are backported for stable distribution
  - no need to use bleeding edge software on server

Questions?

- Feel free to ask now!
- ...or later: `az@{bond.edu.au,debian.org}`
- These slides: `http://people.debian.org/~az/tasit-2004/`