



Designing Manageable Computer Systems

A New Framework for Informatics Division Computing

10th February 2000

Paul Anderson
Division of Informatics
<paul@dcs.ed.ac.uk>

<<http://www.dcs.ed.ac.uk/~paul/internal/icsa.pdf>>



Overview

- Background
 - What do we mean by "Manageable Computer Systems"?
 - What is "Systems Administration"?
 - The Informatics Context.
- Previous Work in Informatics
 - Machine installation & configuration.
 - Software updating & distribution.
- Why change?
 - Changes in the way that systems are being used.
 - Changes in technology.
- A "New System" for Informatics.



Manageable Systems

- From a Systems Administration point of view, a "Computer System" is a collection of individual machines and associated software interacting over a network.
- The perceived "quality" of the system depends on a number of important properties:
 - Usability, Predictability, Reliability,
 - Flexibility, Security, Efficiency.
- A "Manageable" system is one where these properties can be maintained without an unreasonable amount of effort.



Systems Administration

Systems Administration is the job of designing and maintaining manageable systems.

Hardware is generally not a major problem -
The important aspects include:

- Network Services
 - File System, Printing, Backup, Mail, Web, News.
 - DNS, YP, DHCP, Routing.
- Machine Installation & Configuration
 - Hardware devices, Services.
- Software Installation & Updating.



Systems Administration

- For a small homogeneous system that changes infrequently, the Systems Administration requirements can be very low:
 - Manual procedures are adequate.
 - The acceptable quality of service is often quite low.
- For large systems, it is usually the diversity and rate of change which present more difficulties than simply the scale.
 - Linear scaling is unacceptable.
 - Automation is essential.
 - Quality is more important.
 - Management is distributed



The Informatics Context

- Scale
 - 800 machines (35 servers).
 - 1000 software packages.
 - 3000 users.
- Diversity
 - 50 different types of machine.
 - Different hardware configurations.
 - Different software sets.
 - Different service configurations.
- Change
 - 5 new machines per week.
 - 50 machines per week reconfigured.
 - 10's of software packages updated daily.



Developments in Informatics

- Informatics COs have been responsible for development of several original tools for machine configuration and installation:
 - The "LCFG" machine configuration framework.
 - "ifu" provides software updates and synchronisation on Sun file-servers.
 - "UpdateRPM" is a newer mechanism, initially developed to support the Linux systems.
- <http://www.dcs.ed.ac.uk/~paul/>
 - [talks/Config.pdf](#)
 - [publications/LISA8_Paper.pdf](#)
 - [publications/LISA5_Paper.pdf](#)



Installation & Configuration

- Some aspects of configuration:
 - What is the machine hardware and what drivers does it need?
 - What is the machine's network address and which servers should it use?
 - What software should the machine carry?
 - What services should the machine provide and how should they be set up?
 - Who is allowed to access the machine?
- The LCFG system at KB currently contains about 2000 parameters.
- In a system of this size, configurations change continuously.

Configuration Systems



- Manual Configuration is not viable:
 - Too time-consuming.
 - Error-prone and unpredictable.
 - Uncertainty about configuration is insecure.
- Early systems used "Cloning" ("ghosting")
 - Does not support diversity or rapid change.
 - Static system.
- LCFG is a "dynamic" system
 - Configuration parameters are held in a central database and regularly checked by the client.
 - Every machine from portables to file-servers can be rebuild completely automatically with their own unique configuration.

LCFG



- Machines can be installed or rebuilt trivially from the database entry and the master copy of the complete software archive.
- Configurations are always up-to-date.
- Simple "inheritance" allows machines to be configured by specifying small variations from some standard configuration.
- High-level configuration of the entire system is possible.

An Evaluation of LCFG



- Some successful design decisions:
 - All configuration data is held independently of the client.
 - System is dynamic.
- Some successful implementation decisions:
 - Independent modules based on a common core allow distributed development.
 - Basic modules can be constructed very simply.
 - The "database" technology is simple.
- Some problems:
 - Coarse grain security prevents distributed management.
 - The "database" technology is simple.
 - Language choice (shell script) is poor.
 - Modularity of database entries is poor.

Software Distribution



- "Lfu" distributed software packages among a number of servers which were then accessed directly over the network, by the clients.
 - All clients had access to identical up-to-date software.
 - This is not appropriate for clients which may operate disconnected, and have big enough disks to hold all their own software.
- "Updaterrpm" uses software in "RPM packages".
 - Package sets to be carried on individual machines (or groups of machines) are specified in the LCFG.
 - Every night, the actual installed packages are compared with the specification and adjusted accordingly.

An Evaluation of UpdateRPM



- Some successful properties:
 - RPM is a "standard" package format.
 - (Many packages are available pre-built)
 - Integration with LCFG.
 - Easier to identify source of individual files.
 - Packages can be signed for security.
 - Easier to support multiple versions.
- Some problems:
 - Only currently supported under Linux.
 - User-installed software is difficult to control.

Why Change?



- There have been significant changes to the requirements of the system:
 - Scaling & distributed management.
 - Security.
 - Mobility.
 - New technologies.
- This requires major changes in many of our existing procedures and technologies, for example:
 - Directory services.
 - Filesystems.
- Merging incompatible technologies from the ex-departments

Directory Services



- Sun's NIS is currently used for:
 - User information (accounts).
 - Host network information.
 - LCFG.
 - User and machine groups.
 - Etc ...
- Problems
 - Update technology.
 - Replication.
 - Query possibilities.
 - Centralised master.
 - Lack of security.
 - Simple single-table model.

Filesystems



- Sun's NFS
 - Security issues.
 - Mobility.
 - Replication.
 - Portability.
- Global Namespaces
 - Automounters

A "New System" - Priorities



- There are difficult trade-offs to be made:
 - Between the various "desirable properties".
 - Between the quality of the system and the available resources.
- General priorities are likely to be:
 - Provide a solid Infrastructure.
 - Concentrate on "commodity" computing.
 - Delay the (re)implementation of high-level network services.
 - Provide a sustainable system that can evolve steadily.
- <http://www.informatics.ed.ac.uk/admin/committees/computing/meetings/99-07-26/newnet.pdf>

Initial Work



The most obvious areas for initial development include:

- Security mechanisms
 - Per-user authentication (Kerberos?)
- Directory services.
 - User accounts, LCFG (LDAP?)
- Filesystems.
 - Security, mobility (Coda?)
- Machine Configuration
 - LCFG.
- Software Distribution
 - UpdateRPM.

Some Other Issues



- Operating System Support
 - Initially Linux on PC hardware.
 - Capability to support other Unices.
 - "Inter-operate" with NT.
- Network Services
 - Current implementations are adequate.
 - Can be ported later.
 - (Mail, News, print service, etc.)
- Procedures
 - Account management, etc.
- Network Infrastructure

Designing Manageable Computer Systems

*A New Framework for
Informatics Division Computing*

10th February 2000

Paul Anderson
Division of Informatics
<paul@dcs.ed.ac.uk>

<<http://www.dcs.ed.ac.uk/~paul/internal/icsa.pdf>>