BAE SYSTEMS

Linux In Defence:

Linux SOE Evolution At JORN

(by Jamie Birse)

BAE SYSTEMS

Introduction

- Evolution of the Linux SOE at JORN
- Why use a Linux SOE
- Origins of the Linux SOE, Future Trends and Expectations
- Implementation of System Hardware
- Designation of Engineering Timelines
- Example of Simplified Linux SOE
  - Linux SOE Deployment
  - Troubleshooting
- Execution of the Linux SOE at JORN
  - Components of the Linux SOE
  - Types of Automated System Builds
- Summary

BAE SYSTEMS

BACKGROUND

- JORN is a High Frequency radar system spread across Australia at 8 sites and requires significant computing power to achieve successful operation.
- The network is divided into 2 segments, an operational network and a development network.
  - The operational network consists of 3 independent radars consisting of 2 sites each, with a command and control site for co-ordination.
  - The development network is a single site consisting of multiple development, replica and prototype LANs.
  - Currently around 600 Linux systems
- The Operational Radar
  - Has a Key Performance Indicator uptime of 98%.
  - Is a ground based radar analogous in operation, support and maintenance to a jet fighter - there is no help desk support at Mach 2.

BACKGROUND – cont'

- Operations Support
  - Supported by radar support techs at each site.
  - No dedicated system administrators at operational sites.
  - Secondary support is given by development site personnel during working hours.

- To build, maintain, support and develop these systems there is a heavy reliance on:
  - Engineering
  - Configuration management and documentation
  - Training, logistics and obsolescence management

Why Use a Linux SOE

- Requirements
  - A standard operating system base e.g. RHEL.
  - Adherence to strict security requirements.
  - A standard directory/file structure.
  - A standard set of applications and libraries across OS versions.
  - Easily expanded when new requirements arise.
  - A centralised management of
    - Users, Logs, Updates, Procedures
  - A system that is configuration managed and updated by a change management process.
  - New systems must be interoperable with legacy systems until phased out or ported to newer systems.
- The Linux SOE and supporting engineered processes fulfil these requirements.

Linux SOE, where did it come from and where is it going?

- Former systems implemented (and still operational):
  - VAX and VMS
  - DEC/Compaq/HP Alpha and VMS
  - Compaq/HP Alpha and Tru64 5.1 and 4.1g.
- However, these systems were
  - Expensive to maintain with high support costs
  - Tending towards obsolescence with time
  - No clear future for the equipment or their corresponding Operating Systems
  - Hard to find people who want to work with them
- Linux is a well known and obvious choice as a successor
  - Runs on inexpensive commodity hardware (Intel and AMD)
  - “No” OS licensing costs

- **Linux distribution evolution**
  - Trial on Red Hat 7.1 32bit
    - Why – It was on a magazine?
  - Develop SOE, deploy Red Hat 7.3 32bit, but it was EOL.
  - Update SOE, deploy RHEL 4.7 32bit
  - RHEL is chosen for its stability
  - There is a trial version of RHEL 4.5 64bit
  - Minor updates to RHEL 4.8 32 bit due to newer hardware
  - Update SOE, deploy RHEL 5.6 64bit, as this occurs RHEL4 is EOL
  - RHEL 6.1 64bit is deployed for servers only at the same time
  - Future, trialling of Scientific Linux as a replacement/addition.

BAE SYSTEMS

RHEL Clone Migration

- The rationale for migrating towards a RHEL clone in the Linux SOE
  - Not the support cost per node directly, though the Australian Government wants value for the Tax Powers money
  - Cost to verify and acquire extra node support is disproportionately high.
  - Currently employ 6 Linux engineers to support and develop the SOE.
  - Of 3 Red Hat support calls, no satisfactory outcome provided.

Engineering Timelines

- The length of time from engineering requirements to SOE deployment:
  - RH 7.3: 3 years. However, time of RH 7.3 deployment, it was end of life.
  - RHEL4.7: 4 years. 32bit only employed due to software dependency on 64bit RHEL version.
    - Prototype of 64bit RHEL4.5 was in progress in parallel but encountered above software dependency issues.
    - Linux SDE released on RHEL 4.5 64bit prototype + admin on SUN equipment, adds SDE software tools clearcase, clearquest, eclipse.
    - Changes in engineering process greatly increased development time.
  - RHEL 5.6: 2 years.
  - RHEL 6.1: 6 months, servers only and 64bit only.
    - Move to 6.1 was due to intended future use of IPA, risk reduction for the next network wide update to RHEL 6.

Components to the Linux SOE within JORN

- RHEL iso's, 3rd party RPMS, TAR files, configuration files, scripts.
- 650+ scripts and configuration files.
- Hostname Structured configuration File Hierarchy.
- Automated System Builds.

System Hardware

- HP DL380 (starting G3 up to G7), DL360, DL320, XW6000, Z400, some white box systems, supermicro systems, bespoke hardware.
- SDE hardware: SUN Blades and SAN.
- Nvidia graphics cards (due to software constraints) and up to 4 HP 24 inch monitors per workstation.
- Networking: Primarily Cisco but Dell, Foundry and Enterasys also used.

BAE SYSTEMS

Configuration File Hierarchy

- Configuration files are named using hostnames or part thereof.
- Due to the structured nature of hostnames we are able to create a hierarchy of configuration files based on the hostname structure.
- Example hierarchy of configuration files for node coeslcp010:

  lconfig_jindalee.dat
  lconfig_coe.dat
  lconfig_coes.dat
  lconfig_coes__p.dat*
  lconfig_coeslcp.dat
  lconfig_coeslcp0.dat
  lconfig_coeslcp01.dat
  lconfig_coeslcp010.dat

*Note: Node Naming is not ideal for Linux SOE.

- Configuration files can be inserted below the base/default lconfig_jindalee.dat by inclusion in hostname configuration files. These are:
  lconfig_processor.dat
  lconfig_workstation.dat
  lconfig_jorn_operational_node.dat
  lconfig_jorn_development_node.dat
- Tools assist to generate a single configuration file for a hostname to:
  - Debug, Rationalise, Compare, Test and review configuration baselines

Configuration file hierarchy allows for changes:

- Change at the top applies to all systems across JORN
- Change can be applied to only one system
- Change can be applied to a subnet or group of systems
- Adding new systems – potentially as simple as adding it to the hosts file and its MAC Address to the install server's configuration file.

Types of Scripts

- Setup scripts: setup applications, services, configuration files, permissions, directories and files, common functions, wrapper scripts, etc.
  - Most setup scripts modify existing linux conf files or generate new ones from the lconfig_* dat.
  - Allows for conf files changes between RHEL version or even package updates.
- Install scripts: install packages, TAR files, build applications.
- Make scripts: These usually generate specific files, i.e hosts, hosts.equiv, etc.
- Utility scripts: common administration tasks.
- Create scripts: create install media or files, eg. install dvds, install usb keys, kickstart files and generated tar files from configuration management.
- Configured Linux files. e.g. licence files.

Types of Automated System builds

- Primary install server: NFS, NIS, NTP, TFTP, DHCP, DNS, Home directories for Linux and Unix, adding configured users and managed users.
- Failover server: DRBD and heartbeat, DRBD and corosync, rsync and manual scripts.
- Failover NFS server: DRBD and heartbeat, DRBD and corosync, rsync and manual scripts.
- Oracle Database server:
- Processor nodes: general, waveform generators, digital receivers.
- Workstation nodes: operator consoles, developer workstations, JIAB.
- Standalone system:

A Linux SOE Install Simplified

1. A customised kickstart RHEL install. Auto reboot.
2. Automatic start of the OTHR Linux SOE setup script that calls many other setup scripts. Auto reboot.

The Linux SOE Site Deployment

3. Generate an install USB key (32GB key) for the site install server (from another install server).
4. Prepare all hardware to be installed.
5. Boot the install server off the USB key. Return an hour later.
6. PXE boot the failover install server and any other primary servers. Wait 45 minutes.
7. PXE boot all other systems to be installed.
8. Set passwords.
9. Run confidence tests.

BAE SYSTEMS

Troubleshooting Unknown Problems

1. Reinstall it for a quick fix and back to a known baseline.
2. Site support techs do investigation.
3. Raise with Linux engineering team

BAE SYSTEMS

Summary

What does a Linux SOE give us?

1. A known base operating system and tool set.
2. Repeatability.
3. Easy expansion of the network.
4. Access to a great number of new tools.
5. A clear upgrade path, both hardware and operating system.
6. The ability for non Linux radar maintainers to maintain 600 Linux systems across 8 sites.

BAE SYSTEMS

Questions?