NLC Embedded Processor Milieu

’mEl-"yü
the physical or social setting in which something occurs or develops)

Mark Crane
Overview - Requirements
OS - CS - Issues - Implementation
Overview

- Provide reliable, authenticated distribution of the real time environment
- Provide timely initialization of many nodes
- Provide incremental load of images/libraries and databases
- Allow future choice of hardware/communication platforms
- Extendable to local persistent storage devices such as BIOS changes, Xilinx serial proms, PLC code, PLC ladder logic, other...
- SLCTRIO looks like a test bed
The Steps

• Provide the milieu:
  – reset CPU and bootstrap to known condition
  – acquire specific node personality and boot parameters
  – acquire kernel and support functions
  – acquire support code, application libraries, and initial database
  – do it
  – support at least two environments; Development and Production
  – support evolving network interface cards (NIC) and CPUs
  – provide version information and ability to update remotely
Requirements

- Need SLC functional paper to document current system
- JJ’s requirements for NLC provides a good baseline
- A list
  - IOC are different from desktops but the goal is to save direct contact with the nodes wherever they are
  - Scale to 1000 nodes, 2 megabyte images + 500K initial data
  - Boot environment license control and purchase
  - Node security via authentication and encryption
  - OAM (Operations And Maintenance) interface
  - How about the thousands of flash devices for PLC/Xilinx/etc
VxWorks

- Example of the majority of embedded systems
  - local static memory for network and image parameters
  - uses simple TFTP to load image using UDP packets
  - loads a script which then pulls the rest of it’s world

- VxWorks is example of semi-standard interface; bootp works but breaks to get symbols to IOC

- Node identity is held locally on CPU, hard to administer

- PC based VxWorks users use local hard disks to boot
  - Think about 10 megabyte flash disks for local boot
  - Add utilities to remotely verify and update images on these disks

- No multicast TFTP clients known
Linux

• No built in local setup of parameters besides disks
• Identity can come from network or from local harddisk
• Most clusters have local disks
• Most distributed desktops use bootp/NFS
• Multicast tools available
• Use local hard disks to boot
  – Think about 10 megabyte flash disks for local boot
  – RPM utilities to remotely verify and update images on these disks
• No standard way to load the rest of the environment (yet)
• Lynx-OS will be one of the first to accomplish these goals
• Wind River is standoffish just now
EPICS

- Well known procedure to initialize
  - boot VxWorks (FTP/RSH)
  - local scripts to load libraries and database information
- EPICS is very specific but loads separate libraries as required and database download
- NFS disk mounts to transfer files (FTP also)
- Reset and OAM use networked terminal servers and hardware reset switches. Expensive to distribute!
- Is there a tool which gathers library and DB information?
TRIO

• Central “hive” a big seller for TRIO architecture
  – Remote nodes have OAM built into protocol (reset)
  – Local nodes can have OAM cards for easy copper based reset

• Remote TRIO concentrator boots local flash, update via network

• EPICS/VxWorks issues still here if used

• If non-VxWorks centric then more generic boot methods

• Booting of remote sector devices still accomplished via TCP/IP - TRIO used as TCP/IP communication gateway
Issues

- Bootp uses broadcast which is difficult on large networks
- Bootp requests may fill UDP queue on all* (?)
- TFTP servers can get overloaded, Multicast helps
- Network bumps from broadcast, ARP, multicasts, etc
  - note: S.Smith notes if MPS touched it’s down anyway
- SNMP is an answer again but we have no experience here
- Local booting when network is down?
- Incremental system integration booting issues (availability)
- Mother boards and NICs become unavailable monthly
- Fans and packet failures seem to be the major OAM issues
Issues

- I don’t know of VxWorks based Multicast solutions
- **Development and Production available from same network**
  - segment the networks or implement features to disallow crossover
  - Development node is always Development MAC/IP address so personality and connectivity comes from somewhere else
  - bootp/DHCP allows different servers to supply dev/prod info
  - VLANs on top of Ethernet can also help segregate traffic
- **Wireless nodes? Same ideas**
- **Watch license fees for bootroms $50 * 1000 adds up**
- **Each RTOS has specific boot and load requirements**
Standards Tract Implementation

- Intel and 3Com are the major players (Microsoft/Linux)
- TCO, WfM, PXE
- SNMP support for enterprise monitoring solutions
- The evolution
  - NC -> NetPC -> WfM
- PXE runs before OS from local static memory
  - local/run scripts, images, indicate boot type
- DMI for hardware software configuration
- Routers and other devices use TFTP to update images
Component Instrumentation

- Processor
- System memory settings
- Motherboard
- Parallel Ports
- Serial ports
- Disks
- Memory array mapped addresses
- Memory device mapped addresses
- Physical memory array
- Operating system
- Physical container global table
- System BIOS
- System cache
- System slots
- Network adapter 802 port
- Network adapter driver
- Memory devices
- Cooling devices
- Power supply
- Keyboard/pointing device/video
PXE Architecture -- Separate PXE Server

- DHCP Discover + options
- DHCP Offer (client IP addr)
- DHCP Offer (server addr) + options (PXE server tag)
- DHCP (alt. port) request + options
- DHCP (alt. port) reply (filename)
- TFTP request
- Network Bootstrap Program
- Multicast for extra credit

DHCP Server
proxy DHCP Service
Boot Service
TFTP Service
PXE Server

WfM Client

Copyright © 1998 Intel Corporation
Acronyms

- PXE/NSB
- BOOTP/DHCP
- DMI
- WfM
- DMTF
- TFTP/FTP
- RPC/NFS
- SNMP
- RSH
- BIOS
- UDP-TCP/IP
- NIC
- CPU
- ALL*
- OAM
- SLC
- EPICS
- DB
- PLC
- RTOS
- BSCT
- TRIO/SLCTRIO
- EPC/SKATER
- IPL

Slides at http://www/~sparky/nlc