The SPARC Console System

Console System

Architectures

Four different architectures have been analyzed for the SPARC console system:

- ✔ thin-clients (ex. Dafne)
- ✔ workstations (ex. TTF)
- ✔ OpenMosix based PCs Cluster (ex. PC farms)
- ✔ Linux Terminal Server Project

Console System

Architectures: thin-clients

- in use at Dafne (SunRay1: 100MHz-8MB)
- good scalability
- ease of management
  - centralized accounting
  - centralized software upgrade
- server is a single point of failure
- HW and SW are both proprietary
- expensive

Console System

Architectures: workstations

- most common in accelerator Control Systems (TTF)
- distributed system
- gives the capability to tune the single WS in order to accomplish a specific task
- complexity of management
  - auxiliary centralized accounting
  - auxiliary software upgrading system

Console System

Architectures: OpenMosix Cluster

Other more exotic configurations can be taken in account; the more attractive one is a cluster of PCs based on Mosix/OpenMosix:

- ✔ distributed system based on standard PCs
- ✔ computing power increases as PCs number raise
- ✔ very high load over the cluster network (intrinsic)
- ✔ our tests demonstrated the impossibility to migrate SHM and threaded bounded processes
- ✔ lots of kernel crashes demonstrated that OM is actually immature for our needs

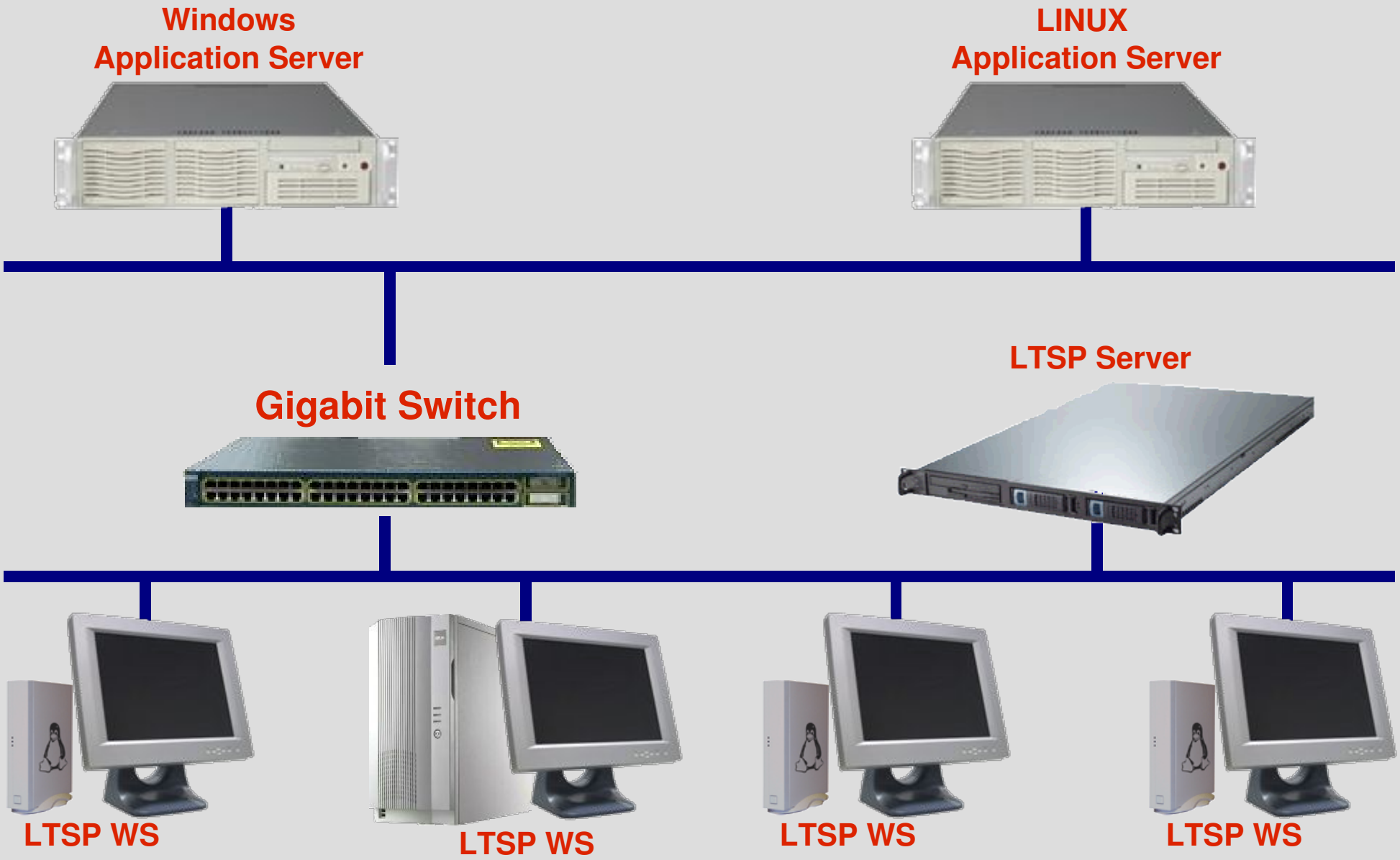
Console System

Architectures: LTSP

- OSS, very good support from developers
- centralized accounting and updating
- highly customizable
  - can act simultaneously as thin-client and/or WS
  - possibility of using OM in future, if it will be stable
  - a number of heterogeneous HW can be used
- very low cost hardware, free SW
- no moving parts configurations are possible
  - low consumes and noise
  - high durability

Console System

LTSP: working logical scheme



Console System

LTSP: working principles

- LTSP server has to offer some services:
  - DHCP server (dynamic IP assignment)
  - TFTP server (gives bootstrap kernel to clients)
  - NFS server (exports terminals root filesystem)
  - XDMCP server (the display manager, optional)
  - NIS server (authentication, optional)
- LTSP terminals bootstrap:
  - boots through PXE or etherboot
  - gets an IP address through a DHCP request
  - gets its Linux kernel though TFTP
  - mounts its filesystem through NFS
  - gives the user a session (a “SCREEN”)

Console System

LTSP: SCREENs

LTSP offers a number of built-in scripts in order to have different types of user sessions, they are referred as “SCREEN scripts”; the user can switch among them through simple shortcut keys:

- local console (on the terminal itself)
- X terminal on a remote display manager
- rdesktop on a Windows remote desktop server
- a telnet/ssh session on a remote machine

Custom scripts can easily be developed

- a labview program
- a mozilla client giving access to the e-LogBook

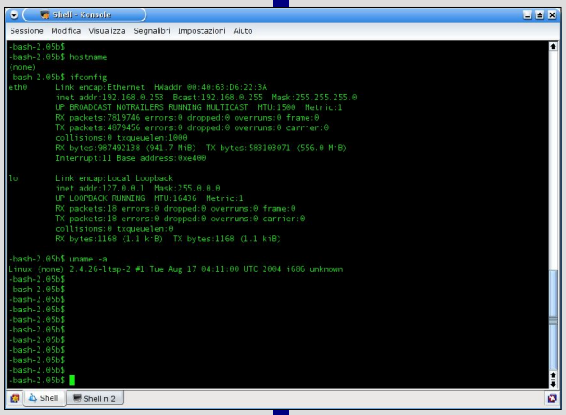
Console System

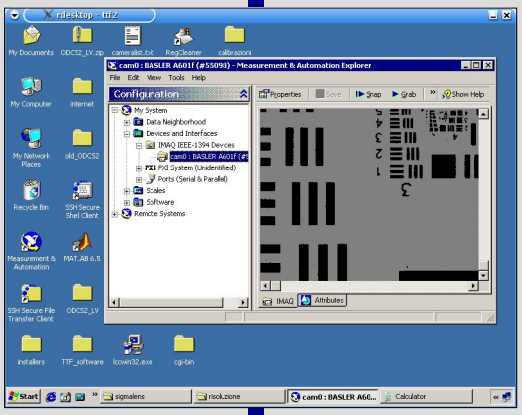
LTSP: SCREENs scheme

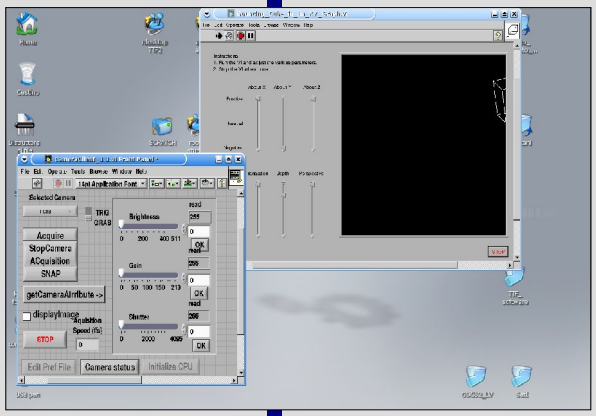
External SSH Server

Windows Application Server

LINUX Application Server







SCREEN_2

SCREEN_1

SCREEN_3

LTSP WS

Console System

LTSP: LOCAL_APPS

Applications can be run locally on the terminal itself. This is accomplished through ssh sessions from the LTSP server to the terminal.

- a terminal can be tuned to be powerful enough to run a particular application, not bearing down on the server
- the system gets very high scalability
- a terminal can be a WS and/or a thin-client as needed

Console System

e-LogBook

An e-LogBook, based on the TTF experience, has been developed and is operational on a temporary server (http://elog.roma2.infn.it/MYelog/index.jsp)

- ✔ readable and editable from remote
- ✔ based on OSS technologies: apache, JSP, HTML
- ✔ search engine
- ✔ files uploads
- ✔ attached files live preview
- ✔ project documents repository
- ✔ works both on Solaris and Linux servers

Console System

e-LogBook: snapshots