SAND2006-4376C

A Minimal Linux Environment for High Performance Computing Systems

James H. Laros III, Sandia National Laboratories
Cynthia Segura, High Performance Technologies, Inc. (HPTi)
Nathan Dauchy, High Performance Technologies, Inc. (HPTi)

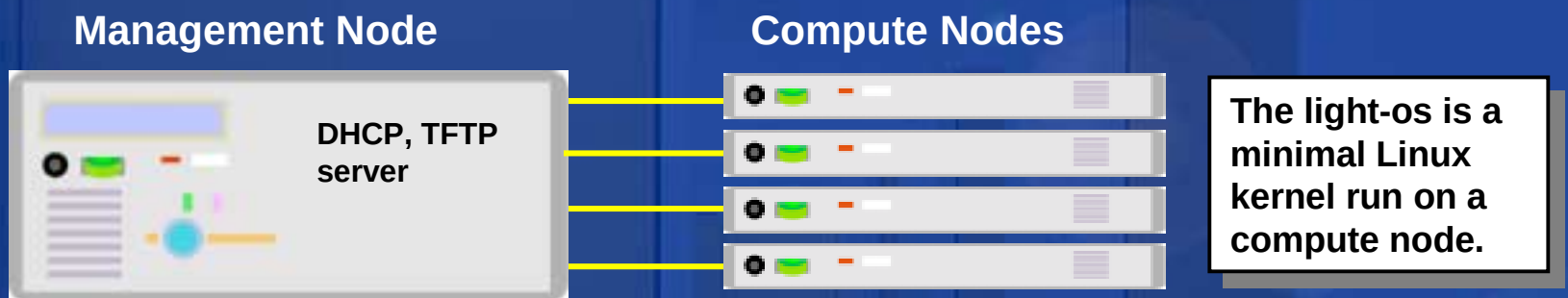
Presented by: Cynthia Segura, HPTi

July 17, 2006

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

What is the Light-os?

- An environment targeted to support parallel scientific applications on High Performance Computers (HPC)
- A scaled-down Linux that reduces system overhead to a bare minimum so that as many resources as possible are devoted to the scientific applications that run on the system
  - Not intended as a generic solution for all clusters



History of the Light-os

- Builds upon research at Sandia National Laboratories:
  - Light Weight Kernels (LWK)
    - Proven efficient on large HPC clusters 10K nodes
  - Diskless clusters
    - Tested on clusters with 2K nodes
- Light-os
  - Emulates a LWK environment using open source software
    - Supports everything that Linux supports
      - **Distributed filesystems (Panasas, Lustre)**
      - **Interconnects (Infiniband, Myrinet)**
      - **PCI, PCI-X**
  - Extends the benefits of diskless clusters (no dependency on server)

Light-os Philosophy

- As many computing resources as possible should be dedicated to the scientific application.
- More resources (memory, cpus) are dedicated to the scientific applications for which the cluster is designed.
- A less complex environment should require less maintenance and support.

In Short, Less is MORE!

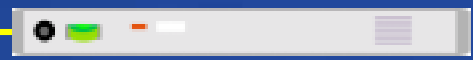
Light-os Requirements

- No local disk
- No remotely mounted root filesystem (NFS)
- No dependency on a Linux distribution
- No kernel source modification
- No Linux system daemons
- No static memory allocation

In a survey of related work, we did not find another project that met these criteria.

How does it work?

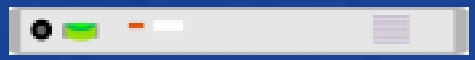
Compute node boots using PXE. Requests boot information from the server.



Server provides kernel and initrd over the network to the compute node

Once the compute node receives its packet, it begins the kernel initialization process in which the initrd is mounted and the linuxrc script is executed.

Light-os Initialization



At this point, we take advantage of the fact that the linuxrc can be any valid executable or script.

Our linuxrc script prepares a tmpfs filesystem as the final root filesystem and only copies what is necessary into that filesystem.

Once constructed, we pivot root into our new filesystem and unmount the initrd.

- After initialization, the compute note is completely independent from the server
- The entire root filesystem is contained in tmpfs, a filesystem which keeps all files in virtual memory
- All remaining resources are available to the applications
- Optionally, we load Busybox into the new root filesystem because of its small size, the tradeoff is acceptable.

Light-os Potential Benefits

- Less maintenance and support
  - No system services, No distribution updates
  - Kernel patches can still be applied
- Increase in mean time between failure (MTBF)
  - Realizes the benefits of a diskless solution
  - Reduces the complexity of the kernel and potential for errors
- Decrease in application wall clock times
  - Applications are not interrupted by services and daemons
  - With more system resources, applications should run faster
  - More predictable wall clock times
- Shortening boot times
  - No filesystem checks, mounting of additional filesystems, initializing of system services

Light-os Trade Offs

- Executables must be statically compiled
  - No shared libraries (could be avoided through the use of a parallel virtual filesystem)
- Specialized runtime environment
  - Ideally would require a mechanism to launch an executable
  - Can support MPICH
- Out of Band System Management
  - Should not require node or host processor involvement
  - IPMI, ILO, etc.

Light-os Summary

- The Light-os is a *compromise* between a LWK and a traditional cluster implementation
  - The light-os cannot achieve the performance of a LWK designed for a specific architecture
  - The light-os is a cheap, easy to implement alternative
- The light-os does not preclude the use of traditional monitoring, runtime tools and environments
  - Each additional service should be evaluated carefully to determine the impact on the system