Compatibility of Linux Architecture for Diskless Technology System

Aryanti Aryanti [1,2,*] Ade Silvia Handayani[2] Ibnu ziad[2] Ikhthison Mekongga[3]

Farid Jatri Abiyyu[2]

[1] *Electrical Engineering, Southern Taiwan University of science and Technology, Tainan, Taiwan*
[2] *Electrical Engineering, Politeknik Negeri Sriwijaya, Palembang, Indonesia*
[3]*Computer Engineering, Politeknik Negeri Sriwijaya, Palembang, Indonesia*
[*]*Corresponding author. Email: da72b201@stust.edu.tw*

ABSTRACT

The increasing need for internet-based computers with high accessibility power causes diskless servers to continue to develop. This paper shows the design and development of the diskless server used compatibility of Linux architecture. In the proposed system, the diskless technology system's creation consists of software design, network topology design, and device integration. Software designed a DHCP server as an automatic IP addressing for clients, developing the Linux Terminal Server Project (LTSP), creating the NFS server, and designing Epoptes. In this study, we added WINE as a compatibility layer to run applications developed for the Windows operating system. The design results provide a good Quality of Service value for each parameter. In the future, adding some functionality and the number of clients can do for further development.

Keywords: *Compatibility, Linux Architecture, Diskless server*

1. INTRODUCTION

Nowadays, information systems are growing and become an essential requirement for delivering information. Need for internet-based computers with high accessibility power is required. A computer network without a hard drive (diskless) can be a solution to this problem. Diskless can store the client's hard disk with the operating system [1]. Thus, clients who do not have a hard disk can still use the hard disk. The course will be safer from viruses and bug attacks than a cloud-based cluster computer network because it only comes from local storage and can be filtered by the host.

This diskless server continues to be designed and implemented because of the many benefits it provides. A diskless server's advantage is that client PCs do not require storage media such as hard disks to operate to simplify troubleshooting, data backup, and save costs in assembling server space. The host's data distributed by a switch using UTP (Unshielded Twisted Pair) cat six cables as the transmission medium.

Previous research related is C.T. Yang et al. [1] Implemented cluster computing of diskless server in a computer classroom. To determine an excellent computational performance, two cluster platforms BCCD and DRBL, are used where DRBL is known to have a better understanding than BCCD. S Prasath.S. et al. [2] proposed centralization of IT infrastructure system with diskless server boot. This study describes in detail the boot process carried out. Booting is done on the network using Network Boot. Phanpikhor et al. [3] presented a performance comparison of diskless technology. This study compared five diskless technology software and six diskless structures tested based on the average available time. The structural link connection obtained high achievement in terms of the minimum average of times.

Duncan, R., & Schreuders, Z. C [4] presented security implication that running in windows software using the Linux system. In this study, Wine is used as a compatibility layer for running windows and determining specific malware. The results of this study indicate that Wine managed to run five samples of the malware window on Linux.

N Absar, N et al. [5] presented star topology performance in small Internetworks. This study analyzes the star topology's performance using a standard LANs model and the OPNET IT GURU simulator. The server load changed based on the different statistics of nods and simulation time.

In [1,2,3], the authors implemented the diskless server, diskless system boot mode, and a comparison of diskless server performance. In [4] authors presented Wine performance as a compatibility layer running on windows. While [5] proposed understanding a star topology on a LAN network with the specified stimulator.

This research aims to design a diskless server used Ubuntu Linux OS with a compatibility system by added WINE as a compatibility layer to run applications developed for the Windows operating system.

2. DISKLESS TECHNOLOGY SYSTEM DESIGN

2.1. Overview

The proposed system designed diskless technology system operating on Linux systems. The Diskless server can store the client's hard disk with the operating system. Thus, clients who do not have a hard disk can still use the hard disk—flowchart diagram system, as shown in Figure 1.

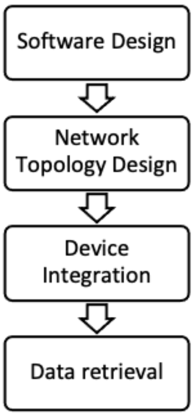


Figure 1. Flowchart Diagram

2.2. Software Design

Software design is the initial stage in building a diskless server. This stage includes designing a DHCP server as an automatic IP addressing for clients, developing the Linux Terminal Server Project (LTSP), designing the NFS server, designing Epoptes. In this study, we added WINE as a compatibility layer, so that it can run applications developed for the Windows operating system but does not support cross-platform to run on the Linux kernel operating systems. The client PC is monitored by the host using Epoptes. The administrator can find out which client is active, broadcast messages to the client, remotely and force-shutdown the client if needed. The Host PC to be built works as a control centre and monitor for the client. IP Address sharing is done on the DHCP server so that clients do not specify IP statically (static TCP / IP).

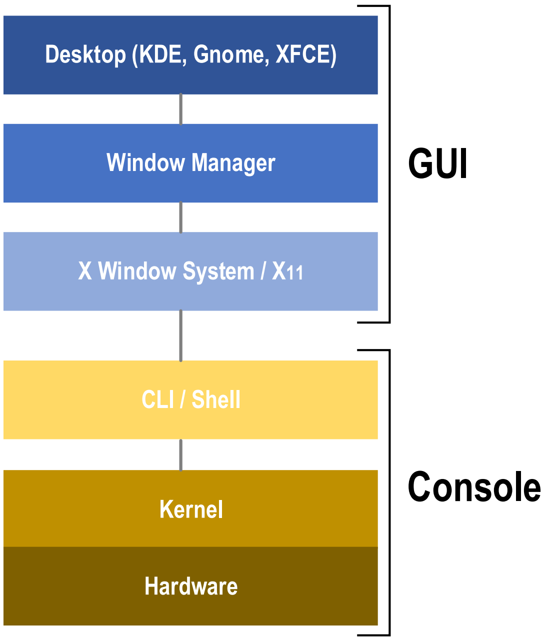


Figure 2. Linux Architecture

In Figure 2, the two main parts of the Linux architecture is GUI and Console. In the GUI, there is a window manager that manages applications running in the GUI. The window system is a unique graphic system for a Linux-based operating system, simplifying user activities. Desktop environment makes it easy to execute commands, no need to use the command line. Desktop environments that are commonly used by Linux are Gnome, KDE, LXDE, XFCE, and MATE. The Console consists of the Kernel and the CLI / Shell. The command line on a Linux-based system that acts as an interface between the user and the kernel. The kernel used for interface software and computer hardware. It also used memory settings for running processes, file settings and settings for input and output on the computer.

2.3. Network Topology Design

In this study, the topology used is Star topology. Star topology has the advantage that the PC server connected to all clients [6]. Clients not connected to other clients; this can avoid looping, which is a disturbance that occurs because switches combined with another switch or the switch itself. The star topology is also by server requirements. It can reduce network failure because all network nodes are connected to the switch to simplify the troubleshooting process if there is a problem.

2.4. Device Integration

This stage is the stage of connecting the host and client and then executing the client. If the host only has one client, the host can be directly related to the client using a UTP cable. If more than one client used, a switch is needed to connect all networks. The network used is local; the button used is a type of unmanageable control so that users only need to use the switch and do not need to configure anymore.

3. RESULT AND DISCUSSION

We implemented the proposed compatibility of Linux architecture for diskless workstation systems. Overall this system consists of a Main host that functions as four servers at once, switches, and end-users. The four servers namely, Secure Shell (SSH), Trivial File Transfer Protocol (TFTP), Dynamic Host Configuration Protocol (DHCP), and Network File System (NFS), as shown in Figure 3.

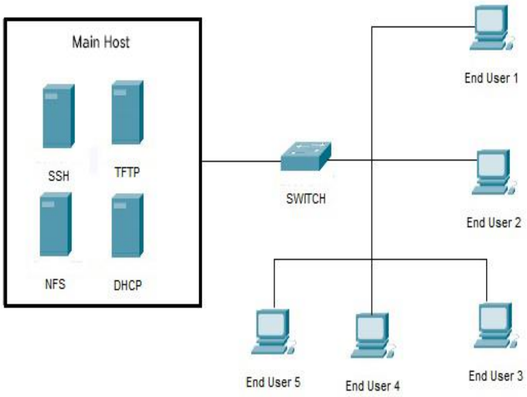


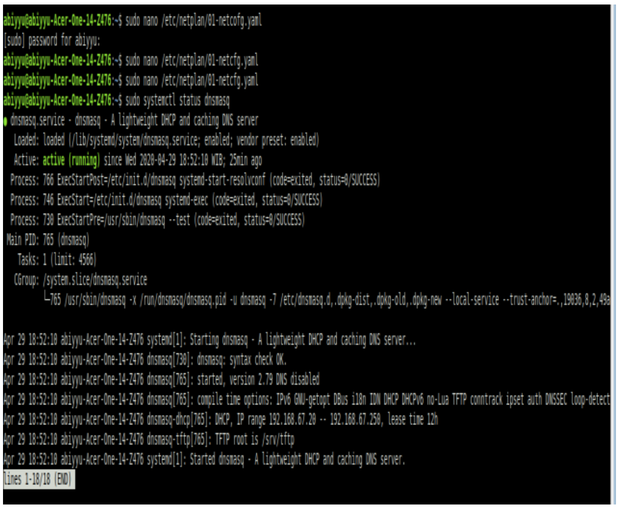
Figure 3. Linux Architecture for Dikless Technology System

In Architecture System, as shown in Fig 3, the client requests IP to the host using the TFTP protocol. The IP designed have the suffix / 24, so the maximum number of clients is 252 clients (IP 192.168.67.2 to 192.168.67.253). IP 192.168.67.1 also functions as a gateway. The client runs the host desktop image, so the computing process only occurs on the host. The SSH server permits clients to access directories so that clients do not need internal storage to operate. The NFS server gives different client permissions for operating systems (for example, Windows) to access the host, with special software, namely the NFS Client.

This research implements compatibility layer as an alternative to Virtual Machine to support cross-platform so that PCs run windows kernel-based applications, which basically cannot use on Linux-based servers. Wine Is Not an Emulator (WINE) open-source software that works like a virtual machine but does not run the operating system on top of the central operating system.

The diskless server design results show an active indication of the DHCP server, marked with a green dot and a green "active (running)" sign on the Linux terminal. Hosts can automatically share IP addresses to clients using the DHCP protocol not to need a router to share IP addresses. The correct LTSP indication indicated by a green dot and the word "active (exited). It means the LTSP is not in use. The DHCP server's functional expression, as shown in Figure 4. A dynamic representation of LTSP, as shown in Figure 5

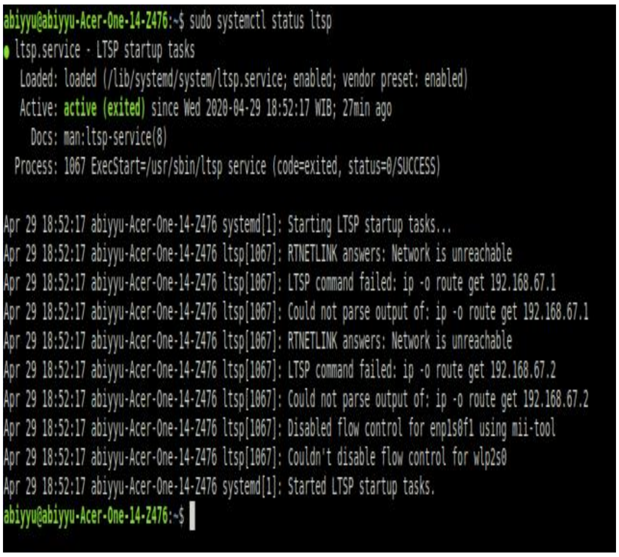
The client can be monitored, controlled, and sent messages by the host using Epoptes. Four active clients connect to the node. The host can broadcast, monitor, and force shut down the client, as shown in Figure 6.



```
abiyyu@abiyyu-Acer-One-14-Z476:~$ sudo nano /etc/netplan/01-netcofg.yaml
[sudo] password for abiyyu:
abiyyu@abiyyu-Acer-One-14-Z476:~$ sudo nano /etc/netplan/01-netcfg.yaml
abiyyu@abiyyu-Acer-One-14-Z476:~$ sudo nano /etc/netplan/01-netcfg.yaml
abiyyu@abiyyu-Acer-One-14-Z476:~$ sudo systemctl status dnsmasq
● dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
   Loaded: loaded (/lib/systemd/system/dnsmasq.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-04-29 18:52:10 WIB; 25min ago
  Process: 766 ExecStartPost=/etc/init.d/dnsmasq systemd-start-resolvconf (code=exited, status=0/SUCCESS)
  Process: 746 ExecStart=/etc/init.d/dnsmasq systemd-exec (code=exited, status=0/SUCCESS)
  Process: 730 ExecStartPre=/usr/sbin/dnsmasq --test (code=exited, status=0/SUCCESS)
 Main PID: 765 (dnsmasq)
    Tasks: 1 (limit: 4566)
   CGroup: /system.slice/dnsmasq.service
           └─765 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -7 /etc/dnsmasq.d,.dpkg-dist,.dpkg-old,.dpkg-new --local-service --trust-anchor=.,19036,8,2,49a

Apr 29 18:52:10 abiyyu-Acer-One-14-Z476 systemd[1]: Starting dnsmasq - A lightweight DHCP and caching DNS server...
Apr 29 18:52:10 abiyyu-Acer-One-14-Z476 dnsmasq[730]: dnsmasq: syntax check OK.
Apr 29 18:52:10 abiyyu-Acer-One-14-Z476 dnsmasq[765]: started, version 2.79 DNS disabled
Apr 29 18:52:10 abiyyu-Acer-One-14-Z476 dnsmasq[765]: compile time options: IPv6 GNU-getopt DBus i18n IDN DHCP DHCPv6 no-Lua TFTP conntrack ipset auth DNSSEC loop-detect
Apr 29 18:52:10 abiyyu-Acer-One-14-Z476 dnsmasq-dhcp[765]: DHCP, IP range 192.168.67.20 -- 192.168.67.250, lease time 12h
Apr 29 18:52:10 abiyyu-Acer-One-14-Z476 dnsmasq-tftp[765]: TFTP root is /srv/tftp
Apr 29 18:52:10 abiyyu-Acer-One-14-Z476 systemd[1]: Started dnsmasq - A lightweight DHCP and caching DNS server.
lines 1-18/18 (END)
```

Figure 4. DHCP Server that has been active



```
abiyyu@abiyyu-Acer-One-14-Z476:~$ sudo systemctl status ltsp
● ltsp.service - LTSP startup tasks
   Loaded: loaded (/lib/systemd/system/ltsp.service; enabled; vendor preset: enabled)
   Active: active (exited) since Wed 2020-04-29 18:52:17 WIB; 27min ago
     Docs: man:ltsp-service(8)
  Process: 1067 ExecStart=/usr/sbin/ltsp service (code=exited, status=0/SUCCESS)

Apr 29 18:52:17 abiyyu-Acer-One-14-Z476 systemd[1]: Starting LTSP startup tasks...
Apr 29 18:52:17 abiyyu-Acer-One-14-Z476 ltsp[1067]: RTNETLINK answers: Network is unreachable
Apr 29 18:52:17 abiyyu-Acer-One-14-Z476 ltsp[1067]: LTSP command failed: ip -o route get 192.168.67.1
Apr 29 18:52:17 abiyyu-Acer-One-14-Z476 ltsp[1067]: Could not parse output of: ip -o route get 192.168.67.1
Apr 29 18:52:17 abiyyu-Acer-One-14-Z476 ltsp[1067]: RTNETLINK answers: Network is unreachable
Apr 29 18:52:17 abiyyu-Acer-One-14-Z476 ltsp[1067]: LTSP command failed: ip -o route get 192.168.67.2
Apr 29 18:52:17 abiyyu-Acer-One-14-Z476 ltsp[1067]: Could not parse output of: ip -o route get 192.168.67.2
Apr 29 18:52:17 abiyyu-Acer-One-14-Z476 ltsp[1067]: Disabled flow control for enp1s0f1 using mii-tool
Apr 29 18:52:17 abiyyu-Acer-One-14-Z476 ltsp[1067]: Couldn't disable flow control for wlp2s0
Apr 29 18:52:17 abiyyu-Acer-One-14-Z476 systemd[1]: Started LTSP startup tasks.
abiyyu@abiyyu-Acer-One-14-Z476:~$ 
```

Figure 5. LTSP that has been active

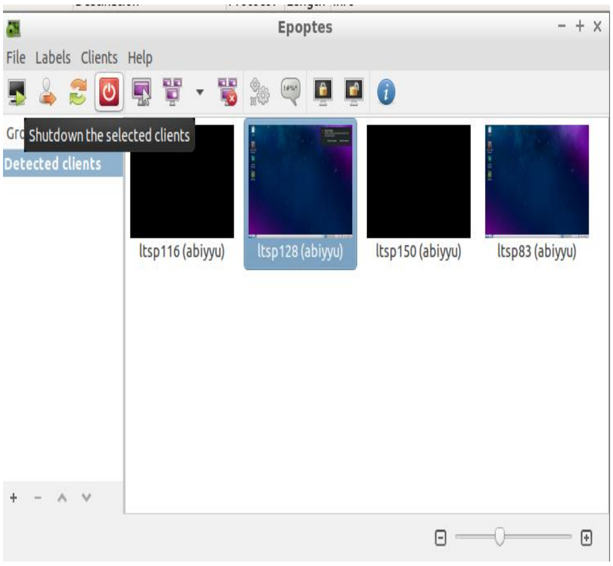


Figure 6. Monitoring results with 4 active clients

In this study, the Quality of Service (QOS) test carried out on Throughput, Packet Loss, Delay, and Jitter, as shown in Table 1. Two client conditions used were Idle and Non-Idle. The performance standard used is the TIPHON standard. In the throughput, when the client is idle, the value ranges from 65 Kbps to 388 Kbps (Bad to Poor), while in the non-idle state, the amount ranges from 11 Mbps to 23 Mbps. At Idle, the network is not too busy. It is not idle; the system becomes engaged to determine the actual client throughput value. Based on the TIPHON standard for throughput, the server has excellent throughput. Packet Loss obtained from measurement results ranges from 0% to 0.007%. These results obtained because the server only uses UTP cables as a transmission medium with a short distance to reduce noise and attenuation when transmitting data. Delay occurs in the range 0.3 to 13.05ms categorized as Perfect in the TIPHON standard. Wireshark's jitter measurement results achieved different jitter values ranging from 3x10-8ms to 12ms classified as Good.

4. CONCLUSION

Compatibility of Linux architecture for a Diskless technology system can have satisfactory performance. The Diskless server can store the client's hard disk with the operating system. WINE as Compatibility Layer support cross-platform so that PCs run windows kernel-based applications. Quality of Service for four clients in Idle and Non-idle conditions obtained good values for each parameter. This system developed by adding some functionality and the number of clients added even more in the future.

AUTHORS' CONTRIBUTIONS

Aryanti Aryanti conceived, searches literature, and drafted the manuscript. Ade Silvia Handayani, Ibnu Ziad, Ikhthison Mekongga supervised the analysis, reviewed the script, and contributed to the discussion. Farid Jatri Abiyyu conceived, designed, analyzed data. The authors approved the final manuscript.

Table 1. Quality of Services' diskless server

No	Client	Throughput (bps)	Packet Loss (%)	Delay (time)	Jitter (time)
1	Idle	65 Kbps	0 %	7.89 ms	0.05 ms
	Non-Idle	13 Mbps	0 %	0.61 ms	0.002 ms
2	Idle	134 Kbps	0 %	32.5 ms	0.01 ms
	Non-Idle	11.4 Kbps	0 %	0.62 ms	0.00000003 ms
3	Idle	388 Kbps	0 %	12.99 ms	0.000018 ms
	Non-Idle	11.97 Mbps	0 %	0.59 ms	0.0058 ms
4	Idle	377 Kbps	0 %	13.05 ms	12.886 ms
	Non-Idle	23 Mbps	0.007 %	0.3 ms	0.00011 ms

ACKNOWLEDGMENTS

We gratefully acknowledge funding provided by PPPM Politeknik Negeri Sriwijaya.

REFERENCES

[1] Yang, C. T., Hsieh, W. F., & Chen, H. Y. 2008. Implementation of a diskless cluster computing environment in a computer Classroom. Proceedings of the 3rd IEEE Asia-Pacific Services Computing Conference.

[2] Prasath S, S., Ravikumar, S., Suryawanshi, P., & Khade, A. 2020. Booting of Diskless Workstation. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. Vol. 1, Issue 1.

[3] Phanpikhor, K., Khummanee, S., Songram, P., & Jareanpon, C. 2013. Performance comparison of the diskless technology. Proceedings of the 2013 10th International Joint Conference on Computer Science and Software Engineering.

[4] Duncan, R., & Schreuders, Z. C. 2019. Security implications of running windows software on a Linux system using Wine: a malware analysis study. Journal of Computer Virology and Hacking Techniques. Vol. 5. Pp. 39-60.

[5] Absar, N., Jahangir Alam, M., & Ahmed, T. 2014. Performance Study of Star Topology in Small Internetworks. International Journal of Computer Applications. Vol. 107. No.2. pp 45-53.

[6] Barranco, M., Proenza, J., Rodríguez-Navas, G., & Almeida, L. (2006). An active star topology for improving fault confinement in CAN networks. IEEE Transactions on Industrial Informatics.