

SMART LAB MODEL: APPLYING THE HIGH AVAILABILITY SYSTEM FOR USING IN EACH COMPUTER LAB

Suphanchai Punthawanunt¹, Sutapat Sappajak², Chanphen Tonsakun-aree³,
Jaruyaporn Kaewkate⁴ and Suphilak Suebsing⁵

¹Dean for Faculty of Science and Technology, Kasem Bundit University,
1761 Pattanakarn Road, Suan-Luang, Bangkok 10250, Thailand, suphanchai.pun@kbu.ac.th

²Associate Dean for Faculty of Science and Technology, Kasem Bundit University,
1761 Pattanakarn Road, Suan-Luang, Bangkok 10250, Thailand, sutapat.sap@kbu.ac.th

^{3,4,5}Instructor, Faculty of Business Administration, Kasem Bundit University,
1761 Pattanakarn Road, Suan-Luang, Bangkok 10250, Thailand

³chanphen.ton@kbu.ac.th, ⁴jaruyaporn.kae@kbu.ac.th, ⁵suphilak.sue@kbu.ac.th

ABSTRACT

This research presents about the previous research of the Diskless system for Computer Lab (DSCL) that the process is added more good solution in automatically procedures to enable arrange the correct used program according to each of the computer lab for managing the requested instructors of each course in each period by the High Availability (HA) System's application programming interface (API). They also have to protect virus, and no programs are deleted by accident. So the researchers realize the importance of benefits and values for students to use the programs in the system correctly and according to what are requested by instructor in terms of requested correct program, computer lab and requested time (faster for checking and loading and no staff at that time for assisting to keep, retrieve and send-receive the data via the internet system in high speed function.

KEYWORDS: High Availability (HA), Application Programming Interface (API)

1. Introduction

In general, there will be many IT staffs in the IT operation management center that are ready for giving assistance to support the teaching. The system management requires a lot of their time to manage and rearrange the whole programs, for example, checking the requested form to use the operation center and rearrange them according to the required

course in any curriculum on each semester, so that they can arrange the correct program according to each course. Also they have to monitor carefully no other irrelevant programs are downloaded by users or students. They also have to protect virus, and no programs are deleted by accident. These processes are amended initially from the research work result "Fully automated diskless deployment for university's lab" [1] from the past. This system mentioned about the Diskless System for Computer Lab (DSCL) that it still could not give good solution in automatic procedures to enable each computer lab to manage the program per request of the instructor of each course in the timely manner. Also, the coordination process among the IT staff and each instructor in each period of time can cause some errors and affect the teaching process. Therefore, the researchers realize the importance of benefits and values for students to use the programs in the system correctly and according to what are requested by instructor in terms of correct program requested in each computer lab and time requested.

2. Literature Review

High Availability System is a system which is almost "online" all the time (always on). This system is with 2 sites' servers. Both sites will always replicate the database so that it provides stability in working system, using High Availability system to enable effectiveness by connecting the both servers at real time. If there is any problem on main server, the system will immediately activate the auto failover. The basic function of the system is that the system will be automatic switched to another server, with so low RTO and RPO that even the users may not know that there is any system error because the users can still keep working on without any downtime [2].

Diskless System is the basic work without hard disk. The client contains no hard disk itself so the basic idea is to create a principle file to send to many clients at one time. Therefore, one client will read-write the principle file which is actually impossible that all machines will read-write principle file at one time, so it is important to arrange area to support the data writing from the clients in order to store all the data in server. The control process which between client and server, divides into 2 parts which are:

- 1) OBM Diskless V.1.9 x 64. This is the control on "ON Hard Disk System" by:
 - Assigning the distribution of IP Address in order to specify the position of the clients.

- Inspecting MAC Address of Card LAN in order to specify the Card LAN on board of each machine to download the images from the server to clients in order to store image files at the Partition image to open the Boot LAN of the clients.

- Also storing all data among server and clients.

2) Net Support school V.11.41.19 Smile_2017, which is the control by adapted software which can control the clients. Also specify many ways of the rights to use, for example, giving tools in teaching, controlling, pursuing and investigating the server. It also can monitor all clients, control the use on internet, printer, USB-port, can send the document files of the teaching media to support the instructors [3].

Application Programming Interface (API) is one of the connecting routes to connect the website of API from other places. It is the center which enables the adapted program to connect the other adapted program or connect the work with the operation system, i.e. Twitter. There are many websites connected the data to Twitter, both reading and sending the data to Twitter, which need the connection and exchange of data by API [4].

The Related researches are:

1) Pholpabu and Muangpool [5], have developed the inclusive managing system of the clients by using diskless system under Linux operation. The work under client-server by the clients will retrieve the operation data through computer network and compile the result at server by resizing and reducing the characteristic of the clients so that it is suitable for work. The storage of data, application services, software, operation system, adapted software and all controlling processes will be included at server which is the center in the network system as a whole.

2) Cox and Mott [6], have developed system and method for secure initial program load for diskless workstations: A client workstation generates a network request for an initial program load. The request is serviced by a server which preferably includes in the reply to the client, the address of an authentication server (AS), client, and a secure initial program load server (SECIPL). The SECIPL then provides a secure Kernel to the client, either encrypted with a key known to the SECIPL and client, or otherwise secured by a cryptographic checksum utilizing a key known to the client and the SECIPL. In this manner,

the client workstation is thereby assured that an authenticated boot image has been received through potentially non-secure communication links.

3) Phanpikhon et al. [7] have compared the Performance comparison of the diskless technology: controlling from the Central Control System will manage large number of computer systems, reducing the redundancy of computer system, reducing the energy consumption and costs. From the result, the link of aggregation structural connections is shown the high performance in terms of the minimum average of times.

4) Yang et al. [8] have developed the implementation of a Diskless Cluster Computing Environment in a Computer Classroom. The objective is to prove that Diskless Remote Boot in Linux (DRBL) has better performance than Bootable Cluster CD (BCCD) in this experiment. In order to achieve the objective of a platform for Free Software Teaching, DRBL is applied to the computer classroom, enabling PCs to be manually or automatically switched among different OS of Windows, Free Software Teaching.

3. Methodology

The researchers have specified 4 researching processes as follows:

3.1 The study of basic data collection. The researchers studied the basic working process of the network system. The basic work is the diskless management of clients, the servers and the high availability system's application programming interface. The study of software open soft, the work of Linux operation system, and the use of Python language program to design and develop the system from the related research data in order to exploit all basic principles to analyze and design the system.

3.2 For the analysis and design of system from the mentioned study result, the researchers have analyzed and designed the network structure of Smart Lab Model System: Applying the High Availability System for using in each computer lab, as specified is shown in figure 1.

Figure 1, the system network structure consists of ① KIS API, which is the system to send the number of the computer operation room from server of system to be checked with basic data of KIS in order to retrieve CourseID (`course_id = get_course_id (room_id)`), ② Images Pool is the storage of image's group which is created for those related courses, and

requested by clients, and ③ server is the part which controls the clients to retrieve the data from Images' Pool, ④ Each computer labs which is the group of Clients controlled by Server.

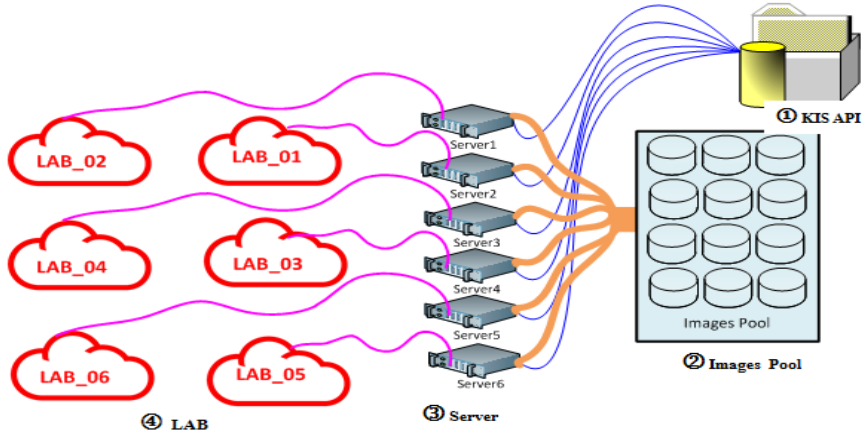


Figure 1 The network Structure of the Smart Lab Model: Applying the High Availability System for using in each Computer Lab

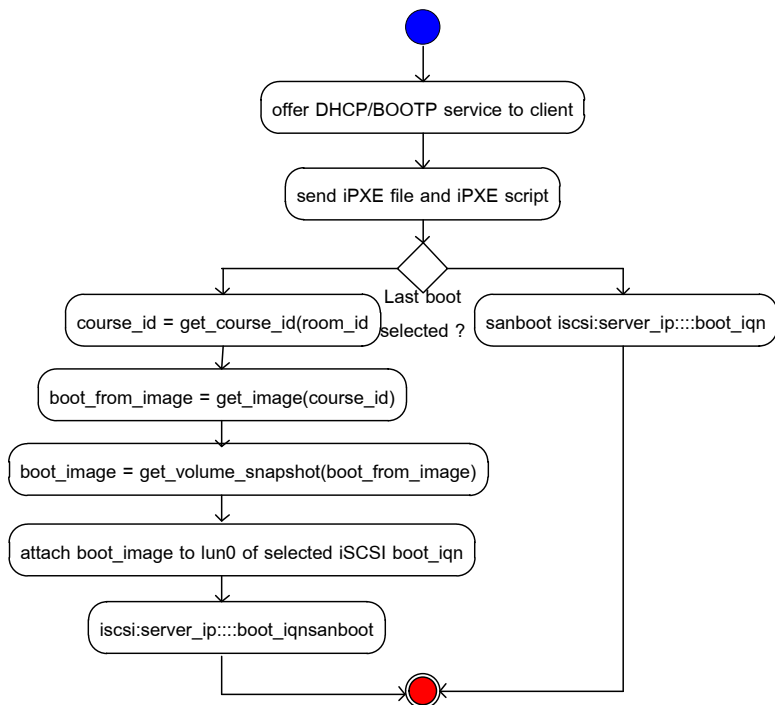


Figure 2 State Diagram of the Smart Lab Model: Applying the High Availability System for using in each computer Lab

The working of system of picture # 2 will be shown that when the users switch on each client in the computer lab (Lab_01), the client will be retrieved image-files to boot the system from server. When the server (Server 1) receives the request, it will connect and send the related file for booting the system and connect to the KIS website, API on screen of user will be shown the connecting status from images' pool and then the users can use that image for each course for that period.

The clients will retrieve files in order to boot the system from server. Then they connect and send the related files by booting the system and connecting to the requirement. The user's screen will be shown the connecting status from images' pool and the image of the course for that period, having steps of work of the system. (The example of the client of the users in computer lab (LAB_01) with server 1 with following steps:

- 1) When the user switches on the client in the computer lab (LAB_01), the client will retrieve image-files in order to boot the system from the network system (offer DHCP/BOOTP service to client).

- 2) When server (server 1) received the request from the client in lab 1, it will connect and send the related files to boot the system (send iPXE file and iPXE script)

- 3) If the users do not press any button within 5 seconds, the server (server 1) will connect with KIS API website to retrieve the CourseID (course_id = get_course_id (room_id))

- 4) The server (server 1) will bring Course ID (course_id = get_course_id (room_id)) to retrieve image for each course from images' pool (boot_from_image = get_image(course_id))

- 5) The server (server 1) will create boot_image from Snapshot of Image for each course (boot_image = get_volume_snapshot(boot_from_image))

- 6) The server (server 1) will connect boot_image with iSCSI (attach boot_image to lun0 of selected iSCSI boot_iqn)

- 7) The server (server 1) will assign clients to boot from iSCSI which is already prepared (iscsi:server_ip:::boot_iqnSANboot).

- 8) If the user selects last boot from selected menu, the client will be boot from original iSCSI in system (sanboot iscsi:server_ip:::boot_iqn)

3.3 The installation for system test. The researchers have installed the system in computer lab. There are 50 clients installed for system test. The test result of Smart Lab Model: Applying the High Availability System for using in each computer lab, the researchers have tested the system by analyzing the effectiveness of system as follows:

1) Testing on switching on the Clients without hard disk in order to check the time needed to enter the operation system Windows 10 in order to check the time using for booting the operation system and to enter other programs for users, totally 50 machines. The result showed that the clients can enter the operation system through all tools, comparing to the time used to boot in old way by using Hard disk is shown in figure 3.

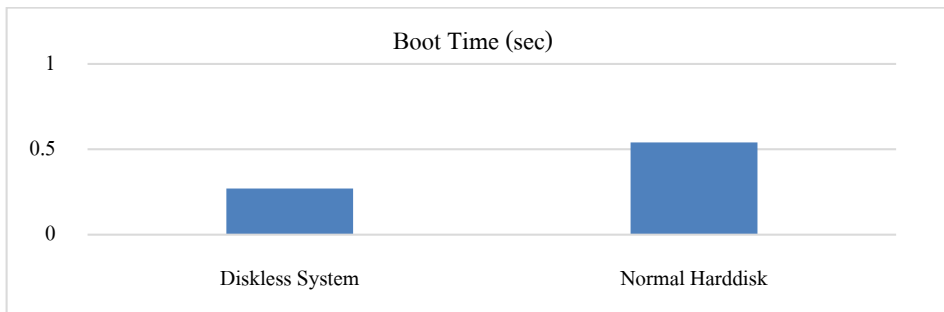


Figure 3 The data's rate using to boot the operation system Windows 10 by diskless system. Time consumption is 28 seconds, while it consumed 54 seconds booting through hard disk.

2) Testing the data's rate upload/download from the clients to server in 30 seconds period of the operation system Windows 10. The result showed that the client used the data to upload/download comparing with the time period in seconds is shown in figure 4.

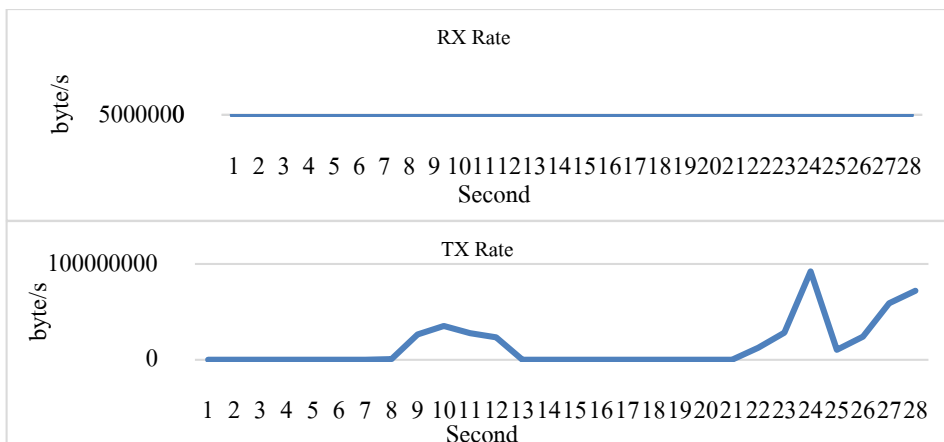


Figure 4 The data rate of upload/download from the clients to server in 30 seconds of operation system Windows 10.

3) Testing the data's rate Read/Write of image file on server in 30 seconds of the booting of operation system Windows 10. The result was found that the client can enter the operation system Windows of all machines when compared with the time period used to boot in the old way which needs to have. It will initially extract the data and then after the 22nd second will start to write the data as specified are shown in figures 5, 6.

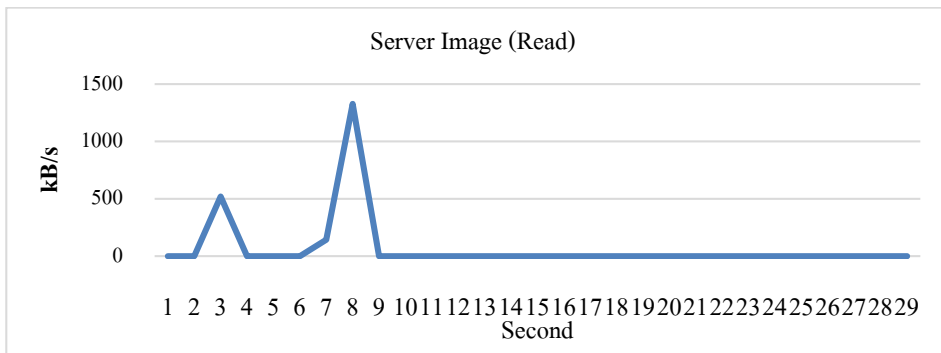


Figure 5 The data's rate of Reading of image files on server in 30 seconds of booting of operation system Windows 10.

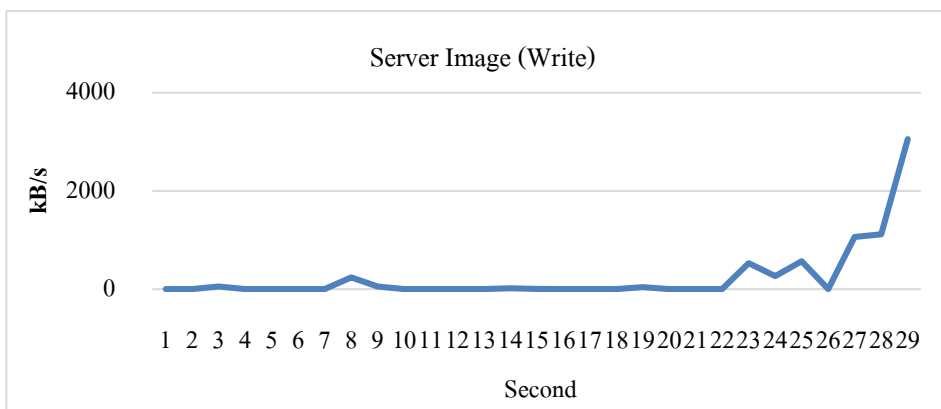


Figure 6 The data's rate of Writing of image files on Server in 30 seconds of booting of operation system Windows 10.

4) Testing the data's rate of Read/Write the data on Swap on server in 30 seconds of the booting of operation system Windows 10. The result showed that the client can enter the operation system Windows 10 of all machines when compared with the time period used to

boot in the old way which needs to have. There will be no writing of data in swap during the process of booting of the clients, as specified are shown in figures 7, 8.

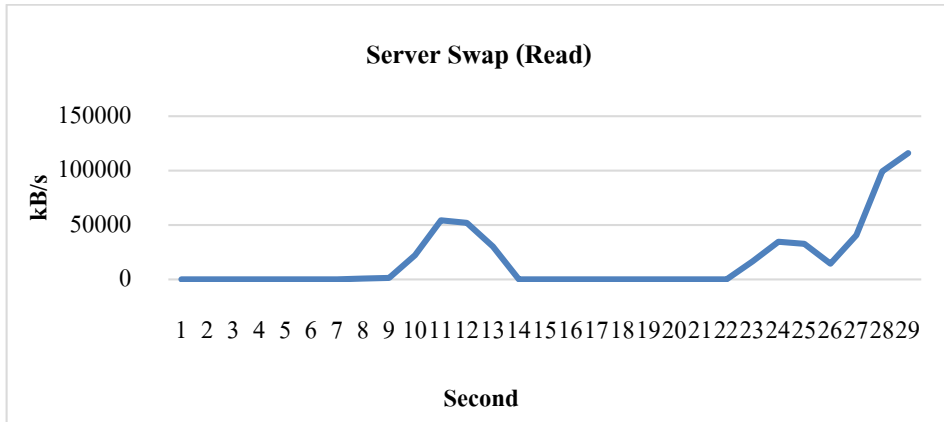


Figure 7 The data's rate of Reading data on Swap on server in 30 seconds of booting the operation system Windows 10.

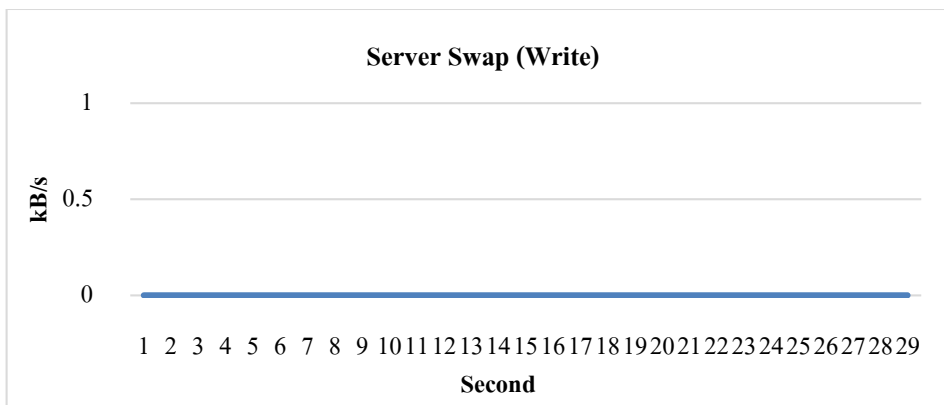


Figure 8 The data's rate of Writing the data on Server in 30 seconds of booting the operation system Windows 10.

5) Test the data of reading the data on snapshot on server in 30 seconds of booting the operation system Windows 10. The result showed the client can enter the operation system Windows 10 of all machines when compared with the time period used to boot in the old way which needs to have. The snapshot of clients will read the data periodically. The

writing of data will happen at the end of the booting process, as specified are shown in figures 9, 10.

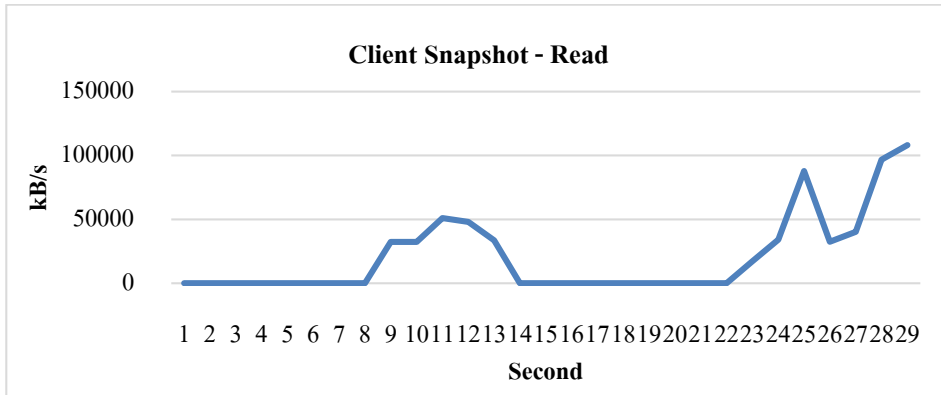


Figure 9 The data's rate of Reading the data on snapshot on server in 30 seconds of booting the operation system Windows 10.

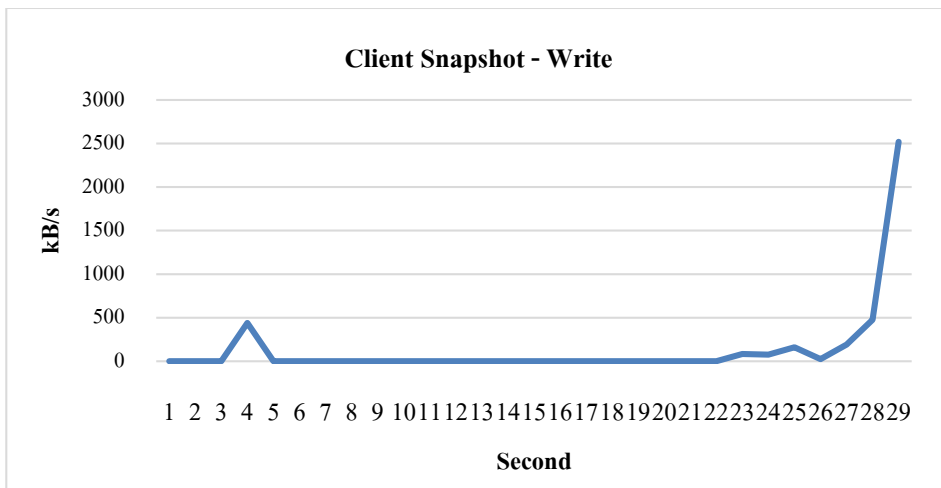


Figure 10 The data's rate of Writing of data on snapshot on Server in 30 seconds of booting the operation system Windows 10.

6) Testing the single-client-test_server-network-stat_correction. The result showed that the client can enter the operation system Windows 10 of all machines when compared with the time period used to boot in the old way which needs to have. Table indicates the other values incurred during the booting of the clients, as specified are shown in figure 11.

No.	Time	bond0				/dev/sda (Image)		/dev/sdb (Swap)		Client (Snapshot)	
		RX	TX	RX rate	TX rate	kB read	kB write	kB read	kB write	kB read	kB write
1	19/07/09 16:19:40	24926328284	15540567882	-	-	0	0	0	0	0	0
2	19/07/09 16:19:41	24926360360	15540568814	34076	932	0	0	0	0	0	0
3	19/07/09 16:19:42	24926395718	15540573023	35358	4209	520	52	0	0	0	0
4	19/07/09 16:19:43	24926506438	15540576079	110720	3056	0	0	0	0	0	440
5	19/07/09 16:19:44	24926540514	15540577061	34076	982	0	0	0	0	0	0
6	19/07/09 16:19:45	24926574590	15540577951	34076	890	0	0	0	0	0	0
7	19/07/09 16:19:46	24926734139	15540584219	159549	6268	144	0	0	0	0	0
8	19/07/09 16:19:47	24926773903	15540588230	39784	4011	1327	240	804	0	0	0
9	19/07/09 16:19:48	24926951135	15541308808	177232	720578	0	52	1388	0	32292	0
10	19/07/09 16:19:49	24928081105	15567666218	1129970	26357410	0	0	22240	0	32292	0
11	19/07/09 16:19:50	24929056397	15602906904	1424292	35239686	0	0	54368	0	51056	0
12	19/07/09 16:19:51	24930716017	15630713968	1209620	27809064	0	0	51940	0	47944	0
13	19/07/09 16:19:52	24931707265	15654099988	992248	23386020	0	0	30120	0	33536	0
14	19/07/09 16:19:53	24931741341	15654100878	34076	890	0	16	0	0	0	0
15	19/07/09 16:19:54	24931775417	15654101768	34076	890	0	4	0	0	0	0
16	19/07/09 16:19:55	24931809493	15654102658	34076	890	0	0	0	0	0	0
17	19/07/09 16:19:56	24931843815	15654103548	34122	890	0	0	0	0	0	0
18	19/07/09 16:19:57	24931877737	15654104438	34122	890	0	0	0	0	0	0
19	19/07/09 16:19:58	24931911859	15654105328	34122	890	0	40	0	0	0	0
20	19/07/09 16:19:59	24931945981	15654106218	34122	890	0	0	0	0	0	0
21	19/07/09 16:20:00	24931980057	15654107108	34076	890	0	0	0	0	0	0
22	19/07/09 16:20:01	24932014133	15654107998	34076	890	0	0	0	0	0	0
23	19/07/09 16:20:02	24932207891	15666680726	193758	12572728	0	528	16552	0	17152	84
24	19/07/09 16:20:03	24932561345	15695019222	353454	28338496	0	264	34540	0	34056	76
25	19/07/09 16:20:04	24933151377	15787296370	590032	92277148	0	568	32704	0	87856	160
26	19/07/09 16:20:05	24933427091	15797747692	275714	10451322	0	0	14388	0	32516	24
27	19/07/09 16:20:06	24933878831	15821785594	451740	24037902	0	1064	40560	0	40184	192
28	19/07/09 16:20:07	24935202775	15880893470	1323944	59107876	0	1116	99376	0	96712	476
29	19/07/09 16:20:08	24938442655	15952384493	3239880	71991023	0	3054	116096	0	108152	2520

Figure 11 Single-client-test_server-network-stat_correction

3.4 Test and store the data. After the installation, testing the system and collecting all data, concluding, analyzing, and testing the system, the researchers used the testing type Black Box [9] by dividing the testing of system into 2 parts as follows:

1) Testing the system by the researchers. To test the function of the required system by inputting and outputting to see if they get the required result as designed, so that they can improve the system to get the required the analyzed and design results.

2) Testing by experts. The researchers asked 3 experts to test the system with instructor and students in the computer lab, then assess the satisfactory result of the users, using the sample group of 50 students in the junior student class (the third year students).

4. Results

The result of the effectiveness of system, researchers gave the Smart Lab Model: Applying the High Availability System for using in each Computer Lab to 3 experts to assess

the performance of the system, then analyze with basic statistics value comparing to standard and concluded the results, as shown in table 1.

Table 1 The results from the experts

No	Detail	\bar{x}	S.D.	the result
1	Using in each computer lab is qualified	4.05	0.62	high
2	The capacity of system	4.02	0.57	high
3	The simpleness of use	4.00	0.63	high
4	The security of system	4.01	0.54	high
	Average	4.02	0.59	high

From Table 1, the results from the experts showed that Smart Lab Model: Applying the High Availability System for using in each computer lab is qualified as "high" ($\bar{x} = 4.05$, S.D. = 0.62), considering each aspect, it was found that the capacity of system is rated as "high" ($\bar{x} = 4.02$, S.D. = 0.57), other than that, the simpleness of use is rated as "high" ($\bar{x} = 4.00$, S.D. = 0.63) and the security of system is rated as "high" ($\bar{x} = 4.01$, S.D. = 0.54), respectively.

Table 2 The results of users' assessment

No	Detail	\bar{x}	S.D.	the result
1	The qualified in overall	4.04	0.65	high
2	The simpleness	4.07	0.85	high
3	The capacity of system	4.04	0.65	high
4	The security of system	4.01	0.90	high
	Average	4.04	0.80	high

From Table 2, the results of assessment was found that Smart Lab Model: Applying the High Availability System for using in each computer lab showed qualified in overall and rated as "high" ($\bar{x} = 4.04$, S.D. = 0.65), considering each aspect, it was found that simpleness is rated as "high" ($\bar{x} = 4.07$, S.D. = 0.85), other aspect as capacity of system is rated as "high"

(\bar{x} = 4.04, S.D. = 0.65) and the security of system is rated as "high" (\bar{x} = 4.01, S.D.=0.90), respectively.

5. Conclusions

5.1 The research was prepared to test the function of Smart Lab Model: Applying the High Availability System for using in each computer lab. From the testing, it was found that when the user switched on the client in computer lab, the server will be connected and sent the related files to boot the system, and then connect to the registration system by using application programming interface (KIS API). The screen of client will be shown the connecting status from images pool and showed the used image for that certainly course, which helps to reduce the hardware management. The system has good control the working of the clients. This research was prepared to give the instructors, the students and the IT's staffs more convenience to prepare the teaching processes in the computer lab. The performance testing and the comments of experts for the suitability of system was rated as "high" (\bar{x} = 4.02, S.D. = 0.59) in overall.

5.2 The result of Smart Lab Model: Applying the High Availability System for using in each computer lab was found that the sample group has satisfactory comment for the system, and rated "high".

6. Suggestion

We can use of the Virtual Desktop Infrastructure (VDI) which is the adapted technology for Virtualization to use with Desktop, to add for next time of the development of Smart Lab, in order to build up more capacity of management, with more control on security. This will increase the value of the use of computer lab, as nowadays many business aspects have brought into use of the VDI rather than PC but it is quite expensive.

Acknowledgement

I would like to thank Mr. Manatsarin Wichianlert and Mr. Paskon Intarasak, staffs of the Computing & Technology Center, Kasem Bundit University for their helpfuls in laboratory tests.

References

- [1] Punthawanunt S, Sappajak S and FUJI Y. Fully automated diskless deployment for university's lab. *Kasem Bundit Engineering Journal* 2018;8(2):104-16.
- [2] High Availability System [Internet]. 2018 [cited 2019 July 2]. Available from: <https://www.cloudhm.co.th/about/blog/high-availability/>
- [3] Client-Server-Network (Online) [Internet]. 2019 [cited 2019 June 7]. Available from: <https://www.techwalla.com/articles/what-is-a-client-server-network>
- [4] Wikipedia. Application programming interface [Internet]. 2019 [cited 2019 June 2]. Available from: https://en.wikipedia.org/wiki/Application_programming_interface
- [5] Pholpabu P, Muangpool T. The Testing of Client Controlled System by Diskless Server Management Software. *Journal of Project in Computer Science and Information Technology* 2018;4(2):44-50. (In Thai)
- [6] Cox JO, Mott JM, inventor; International Business Machines Corp., assignee. System and method for secure initial program load for diskless workstations. United States patent US5349643A. 1994 Sep 20.
- [7] Phanpikhor K, Khummanee S, Songram P, Jareanpon C. Performance comparison of the diskless technology. *Proceeding of the 2013 10th International Joint Conference on Computer Science and Software Engineering (JCSSE)*; 2013 May 29-31; Maha Sarakham, Thailand. p. 99-104.
- [8] Yang C, Hsieh W, Chen H. Implementation of a Diskless Cluster Computing Environment in a Computer classroom. *Proceedings of the 3rd IEEE Asia-Pacific Services Computing Conference*; 2008 Dec 9-12; Yilan, Taiwan. p. 819-24.
- [9] Black Box Testing [Internet]. 2019 [cited 2019 July 18]. Available from: <http://softwaretestingfundamentals.com/black-box-testing/>

Author's Profile



Suphanchai Punthawanunt, Dean for Faculty of Science and Technology, and Director for Computing and Technology Center. Kasem Bundit University, 1761 Pattanakarn Road, Suan-Luang, Bangkok 10250, Thailand. E-Mail: suphanchai.pun@kbu.ac.th

Education: BBA. Degree in Business Computer Assumption University, Bangkok, Thailand and M.Sc. Degree in Computer Information Science, Assumption University, Bangkok, Thailand.

Interested research area: Data Center, Computer Network Management, Internet of Things, Big Data and Cloud Computing.



Sutapat Sappajak, Associate Dean for Faculty of Science and Technology, Kasem Bundit University, 1761 Pattanakarn Road, Suan-Luang, Bangkok 10250, Thailand, E-Mail: sutapat.sap@kbu.ac.th

Education: M.Sc. Degree in Information Technology, Eastern Asia University, Pathumthani, Thailand and M.Ed. Degree in Educational Technology and Communications, Sukhothai Thammathirat University, Nonthaburi, Thailand.

Interested research area: Computer Graphic, Distributed Systems, Operating System and Software Engineering.



Chanphen Tonsakun-aree, Instructor, Faculty of Business Administration, Kasem Bundit University, 1761 Pattanakarn Road, Suan-Luang, Bangkok 10250, Thailand, E-Mail: chanphen.ton@kbu.ac.th

Education: M.Sc. Degree Computer Information System, Assumption University, Bangkok, Thailand.

Interested research area: Management Information System, Data Transformation and Data Analytic.



Jaruayporn Kaewkate, Instructor, Faculty of Business Administration, Kasem Bundit University, 1761 Pattanakarn Road, Suan-Luang, Bangkok 10250, Thailand, E-Mail: jaruayporn.kae@kbu.ac.th

Education: M.Sc. (Information Technology), King Mongkut's University of Technology Thonburi, Bangkok, Thailand.

Interested research area: Management information system, Computer Graphic and System analysis and design.



Suphilak Suebsing, Instructor, Faculty of Business Administration, Kasem Bundit University, 1761 Pattanakarn Road, Suan-Luang, Bangkok 10250, Thailand. E-Mail: suphilak.sue@kbu.ac.th

Education: BBA. Degree in Business Computer, Kasem Bundit University and M.Sc. Degree in Information Technology, Mahanakorn University of Technology, Bangkok, Thailand.

Interested research area: Programming/application development, Web Design, Web Programming and Web Marketing, Management Information Technology, Web E-Commerce, E-Marketing, Content Management System (CMS)

Article History:

Received: August 22, 2018

Revised: September 5, 2019

Accepted: September 6, 2019