

Practicality Verification of an Application Virtualization System Assuming Use of BYOD Environment

著者	Hayashi Toyohiro, Fukuda Yutaka, Satoh Akihiro, Kai Kyoko, Kushiro Noriyuki
journal or publication title	Open Journal of Social Sciences
volume	8
number	3
page range	482-492
year	2020-03-31
URL	http://hdl.handle.net/10228/00007838

doi: <https://doi.org/10.4236/jss.2020.83040>

Practicality Verification of an Application Virtualization System Assuming Use of BYOD Environment

Toyohiro Hayashi, Yutaka Fukuda, Akihiro Satoh, Kyoko Kai, Noriyuki Kushiro

Information Science and Technology Center, Kyushu Institute of Technology, Fukuoka, Japan

Email: toyohiro@isc.kyutech.ac.jp

How to cite this paper: Hayashi, T., Fukuda, Y., Satoh, A., Kai, K., & Kushiro, N. (2020). Practicality Verification of an Application Virtualization System Assuming Use of BYOD Environment. *Open Journal of Social Sciences*, 8, 482-492. <https://doi.org/10.4236/jss.2020.83040>

Received: March 10, 2020

Accepted: March 28, 2020

Published: March 31, 2020

Abstract

From 2019, We - Kyushu Institute of Technology started to bring and use students' laptops (called "BYOD on education") in some lectures. In the same timing, we have replaced our information systems for education and research use (we call "Education systems"). The new system is turned to support our BYOD environment on education scene. For one thing that, "Application virtualization system" has been adopted. It behaves as "the application was installed on PCs" without some installation procedures. It can be expected to enhance lectures such as some exercise using software. Before providing application virtualization system, some practicality verification is needed. We adopt this system to 30 different software for education and research use. After verification, we judged that 25 of 30 software can be adopting for BYOD environment on education scene.

Keywords

BYOD on Lectures, Application Virtualization System, Application Delivery, Cloudpaging

1. Bring and Use Students' Laptops on Our Institute

BYOD (Bring Your Own Device) is a trend way to bring own information equipment and use for business scene. Recently, BYOD is often adopting for education on universities, there are some reports of BYOD by using students' laptops on national university of Japan (Ohashi et al., 2018). On the other hand until 2017, in our institute, Kyushu Institute of Technology, information related lectures are opened on "PC classrooms" that PC terminal (Windows and Ubuntu can use) are equipped. In the latest system (2014-2018), Ubuntu Desktop was

operated mainly on a PC terminal, and collective education such as exercises was opened. This is the reason that some performances (CPU speed, RAM capacity etc.) are under spec for information related lectures on a college of technology, so we avoid adopt of BYOD for a long time.

However, the trend of adopting became stronger around 2016. Some middle class of laptops released on 2018, benchmarks of computation (CPU speed etc.) are same or upper than the PC terminal (desktop style PC) on our classrooms introduced on 2014. In addition, these laptops have rich capacity of batteries, so it can be worked all of day on fully charged (Tomishige et al., 2019).

Actually, over than 90% of students in our institute have own laptops. Focusing on information infrastructure and services for our institute's members. Over 400 of Wi-Fi access points in our campus are set, so users can access to campus network from almost buildings/classrooms. Essential tools for taking lecture (such as LMS, course registration system) are built as web application, and we utilize many public IaaS such as e-mail system, online storage system. Student can use many tools via Wi-Fi and own laptops.

As described above, performances of laptops, ownership ratio of students' laptops, status of information infrastructure are higher degree.

So BYOD by using students' laptops were considered by our executive committee in 2016. Finally, we decided that starts BYOD from 2018 (Kyushu Institute of Technology, 2019) (Kai et al., 2019).

Preparation of Information Infrastructure, Web Services, Software Licenses for Education

In order to educate utilized BYOD smoothly, we think the following should be prepared. These are:

- 1) Information network infrastructure such as Wi-Fi APs, high speed campus network,
- 2) Web services for education,
- 3) Comprehensive software licensing and management.

Status of these preparations in our institute is below.

- 1) Information network infrastructure

High speed information network is prepared as campus network. Connection speed about inter-building in campus is 10 Gbps - 40 Gbps range. In all of classroom, Wireless LAN (Wi-Fi, IEEE802.11ax or IEEE802.11ac) access points are set.

There are about 470 APs at Sep.2019. A network connection via Wi-Fi is provided as "Information outlet service", and this access is controlled by IEEE802.1X (EAP).

All of staff and student can connect campus network by their unified account (Fukuda et al., 2019).

- 2) Web services for education

Most of services for education and research use are deployed as Web Application, their services can be accessed by generic WWW browsers with some exten-

sions. We provide web services for students such as LMS (Moodle, OSS), course registration system (Live Campus, NTT Data Inc.), self assessment of learning recording system (Custom made), e-mail and online storage (Office 365 A1, Microsoft), online manual of information infrastructure and services (published by Joomla (CMS)).

3) Comprehensive software licensing and management

We provide comprehensive licensing and management for software that will be in high demand in lectures and exercises. All of staff and student can install and use their software in own PCs under controlled condition. For example, major Microsoft products (Windows OS, Office suite, Visual Studio, etc.), end-point Anti-Virus products, Engineering software (MATLAB, Mathematica) are provided as comprehensive.

2. Supporting BYOD on Education by “Education Systems”

To support information related lectures, education, and researches in our institute, we have exclusive information systems named “education systems”. These are replaced every 4 - 5 years, and these are aimed an advanced architecture in every replaced scene. But basically, these mainly provided an environment using PC terminals (host terminal, UNIX terminal, net boot PC) in classrooms.

Recent replacing timing was set to 2017-2019. Concretely, it was planned to start design in 2017, start building systems in 2018, and switch to new system and start operation in April 2019.

This replacing timing will overlap with adopt BYOD (starts from 2018) on our institute. So commission and executive committee has suggested that education systems should support BYOD. Based on this suggestion, education systems are turned to support our BYOD environment on education scene. Concretely, deploying number of PC terminals in classrooms will be reduced to 50%. On the other hand, providing high performance and availability web services, application platforms,

VM infrastructure, storage system is pushing ahead (**Figure 1**) (Hayashi et al., 2019).

3. The Modern Application Virtualization System

In some lectures, exercise format will be applied by using particular engineering application. In our institute, programming language (compiler), CAD, scientific computing platform are often used. To prepare an environment for providing these applications, below approaches has been adopted in the past.

- 1) Installing particular application to OS images for PC terminal
- 2) Preparation VDI (Virtual Desktop Infrastructure) environment for execution of particular application
- 3) Distribution of essentials for installs particular application (installer, license files etc.) to students

Exercise in every lecture can be done which approach is adopted. On the other hand, there are some problems on service operation.

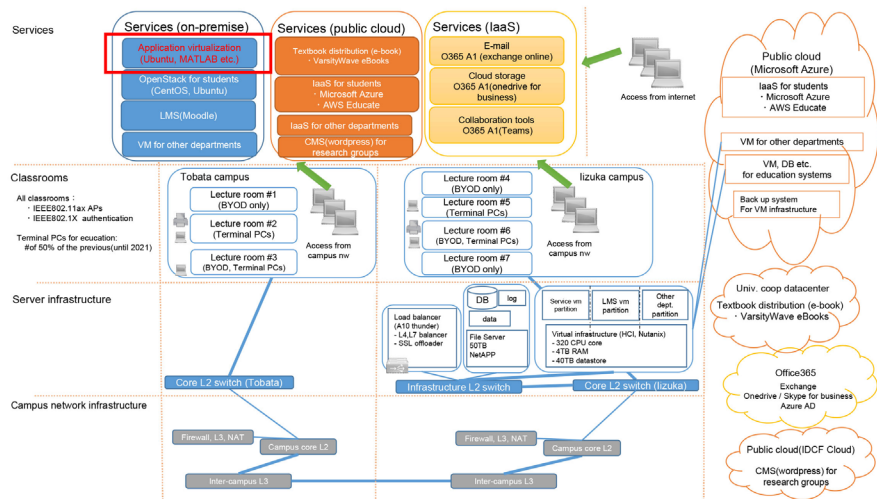


Figure 1. System configuration diagram of education systems (since 2019-).

In approach 1, making up and operation check test of OS images are required. In addition, this approach is applied only PC terminal, it is too limitedly.

In approach 2, particular application can be used from various remote terminals than approach 1. But in generally, operation of VDI service will be taken high operation cost, some application cannot be executed on VDI in license issue. And the weakest point is that performances of PC terminals or students' laptops do not contribute for running applications on VDI.

In approach 3, by taking software installer or license files to students will cause illegal taken out to outside, access control will be difficult, and inventory of software (uninstalling of software after graduation is needed, etc.) will be difficult. These will be risk of license management.

To resolve above problems, we have applied “Application virtualization system” in newer education systems. In next Section 3.1, we will describe more detail.

3.1. Essentials of Application Virtualization System

Application virtualization system will package (called “virtualization” or “encapsulation”) essentials (such as Files, Device drivers, environment variables, registry information) to execute an application, and deliver packages to users' computing environment such as PC terminal, users' own PCs.

In users' computing environment, agent software for Application virtualization system behaves as if the delivered essentials were installed. Thus, an application can execute normally without installation process on users' environment. It can be expected to avoid installation hassle.

Typical software/hardware configuration of application virtualization system (shown on **Figure 2**) will be below elements.

1) Packaging software/dedicated PC

Packaging software detects execution of software installer, file allocation, registry creation etc. These detected elements are captured and packaged as

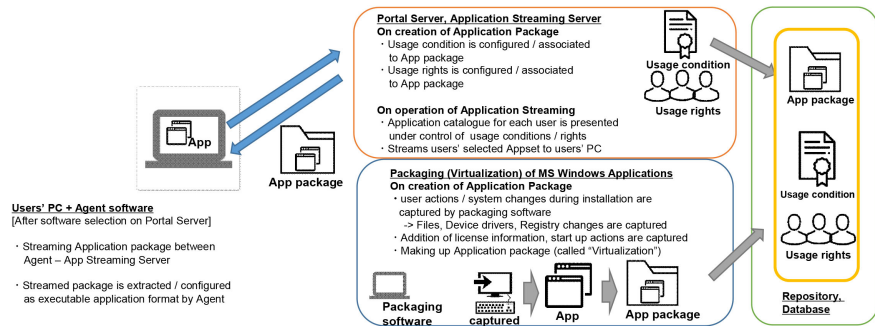


Figure 2. Typical configuration of application virtualization system.

“Application package”. Packaging software and installer of application which users want to install should run on dedicated PC.

2) Portal server, Application streaming server

It provides functions for software (application package) selection web portal, delivery an chosen application package to users' environment. In app streaming server, usage condition (expiration date of application etc.) and usage rights (accessible users, computers etc.) are set for each application package. In addition, some products have functions about control delivery method for each package (delivery as “demand streaming” or “download” etc.).

3) Agent Software

Agent software will install on users' environment. It cooperates with application streaming server to control application execution. And it extracts application package and manipulate file system/registry of users' environment to behave as “an application is installed on users' environment directly”.

By adopting an application virtualization system, an application can be executed on users' PCs natively. In addition, access control and license management for applications will be guaranteed. Making up of OS images, preparation of VDI environment, distribution of installer for applications are not needed. Problems described in the previous section are expected to be solved. After all, the approach of “an application can be executed on users' PCs natively” has a higher affinity for BYOD on education. It is an effective service to support BYOD.

But in fact, application virtualization system is not newer solution. It already existed in the early 2000s, and was introduced in our educational system operated at 2009-2013. It was used to distribute CAD software to classroom PC terminals (Nakamura, 2010). However, the system has serious restrictions, such as installation drive cannot be selected (cannot be installed on the C drive), and some of device drivers could not be included to application package.

3.2. Our Selection: Numecent Cloudpaging: A Modern System

As an implementation of application virtualization system to support BYOD, we applied Cloudpaging (by Numecent Inc.) for our educational system operated at 2019 (Numecent Inc., 2018). An system configuration is shown on **Figure 3**.

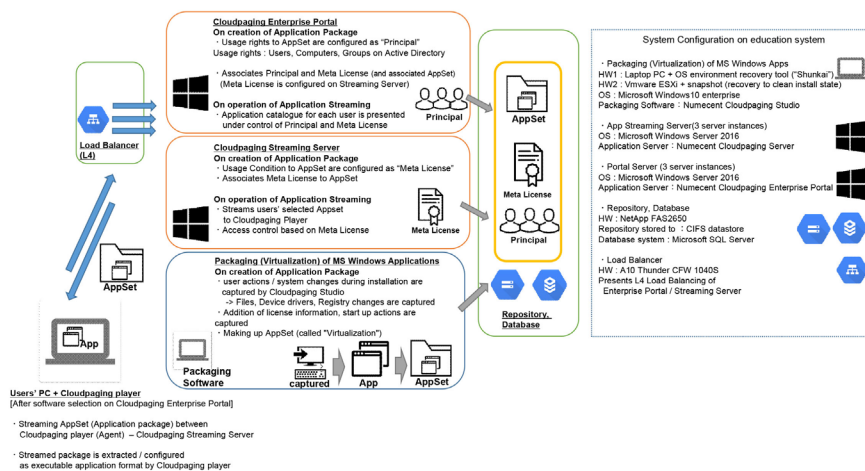


Figure 3. Configuration on application virtualization system on education systems.

System configuration is almost same as other application virtualization systems. In this system, it is noted that the development cost is applied to the application packaging method. It contributes to the improvement of the packaging success rate.

The application package on Cloudpaging is called "Appset". In AppSet, capturing of low level device drivers into application package, control of allocation files (Whether to prioritize existing or delivered files/execute files, called "Layer"), granting privilege escalation to executable files in AppSet (it requires changing Windows Defender settings) are supported. These features contribute for rating up of packaging success rate.

On streaming server, setting of usage condition (called "Meta license") such as usable number of AppSet, delivery method (download or streaming) are supported. In addition, settings of usage right management (for users/groups/PCs, called "Principal") for using permission of Appset are also supported. These will be contribute of access control and license management for application. Major parameters on meta license and Principal is shown on **Figure 4**.

4. Utilization of Cloudpaging in BYOD on Education

By using Cloudpaging system, expected that distribution of software and materials used in classes, and provision of an self study environment will be made easier for students' own laptops. It will contribute to enhance lectures such as some exercise using software. But the first, we define some policies for distribution of software.

These are: 1) What applications should be prepare, 2) Parameters for Appset (These are related to license control etc.).

Regarding policies for distribution of software, the usage form BYOD environment in our institute is considered.

As follows,

1) Comprehensive license contract software can be installed on the student's laptops, but strict license management is required,

Meta license : usage condition	Behavior	Example scenario
Maximum Seats	Set a available number of AppSet (called "Seat"). If it exceeds, AppSet will not be delivered	Control the number of applications that can be started according to the software contract
Expiration Time	Set an expiration date of AppSet. If it exceeds, AppSet will not be delivered	Inhibit execution of software after student has been graduated
Offline Duration	Set the number of grace days where Cloudpacing player cannot communicate with the server	Reduce long term Seat consumption (equivalent to license consumption)
Concurrent per user	Set the number of executable same AppSet concurrently by one user	Control the number of applications that can be started according to the software contract (one instance per one user, etc.)
Patch Version	Set the version of updated AppSet (called "Patch Version") to be delivered	Force to update software
License Type : Floating	If set, one seat is consumed where AppSet is "Started" by Cloudpacing player	Perform the operation of streaming the application from cloudpacing server
License Type : Fixed	If set, one seat is consumed where AppSet is "Delivered" by Cloudpacing server	Perform the operation of download the application from cloudpacing server. Essential parameter when Appset runs on BYOD.
Run Option : Prefetch For Offline	If set, Cloudpacing player performs download all AppSet data instead of streaming	AppSet will be downloaded (it stop to streaming) from cloudpacing server. Essential parameter when Appset runs on BYOD.
Run Option : Remove on Automatically	If set, delivered AppSet will be deleted where it exceeds "Expiration Time" parameter	To simple operation, eliminates the need for user to delete AppSet

Principal : usage right management	Behavior	Example scenario
User / User Group	Use and browse permission of AppSet to particular users / groups	Make AppSet available to students of a specific lecture or exercise
Machine / Machie Group	Use and browse permission of AppSet to particular PCs	Make AppSet available to specific classrooms

Figure 4. Major parameters on meta license and principal.

2) Network connection must be campus wireless LAN.

We define the following as a Cloudpacing Meta license and Principal suitable for this usage.

Applications should be prepared

Applications matches any of the following are selected. In the practicality verification in the next section, selection of applications is made based on this criterion.

- Want to use as teaching materials, but the installation procedure is complicated and requires know-how
- Comprehensive license agreement, high usage rate as teaching materials
- Includes installation files/license information that cannot be show to students
- Requires management of usage rights (to reduce the number of simultaneous activations, uninstallation is required at graduation, etc.)
- An environment almost same as served on a PC terminal of the past educational systems

Parameters for AppSet

Parameters for AppSet (shown on **Figure 4**) are determined as follows.

- Prefetch for Offline: "True", License Type: "Fixed"

It is not realistic to stream AppSet to laptop under wireless LAN connection from the viewpoint of bandwidth and stability. Therefore, operated by downloading AppSet to laptop will be determined.

- Expiration Time: Set to "End of school year"
Set expiration date to prevent taking out software
- Concurrent per user: Set to "1 (For example)"

Reduce the number of simultaneous executable same AppSet per user depending on the number of software licenses

4.1. Selection of Applications to Make AppSet

Based on the above criterion for "Applications should be prepared", select applications and make AppSet. Since application virtualization service has just

been launched to staffs/students of our institute, it is in the stage of practicality verification. Therefore, the selection of applications is based on the selection by administration team of educational systems. In order to improve the practicality, it is necessary to make selections based on interviews with the teacher in charge of lectures. This is major future work.

There are 30 types of applications that actually created AppSet, as shown on **Figure 5**. 26 types of 30 AppSet (**Figure 5**. Type 1, 3) was created assuming general application distribution by installer operations, file layout, process access right control, etc.

For the remaining 4 types (**Figure 5**. Type 2) was created assuming an environment distribution for running Ubuntu Desktop on Oracle VirtualBox. Because Ubuntu Desktop was used for collective education environment on the past educational systems. Ubuntu on VirtualBox will be alternatives that.

4.2. Practicality Verification/Judgement

As practicality verification of application virtualization system, some measurements and judgements are made to each AppSet.

These are: 1) Measurement of delivery time (bandwidth) of AppSet to users' environment, 2) Observation of delivery successful/failed delivery AppSet on users' environment, 3) Observation of successful/failed launch of delivered AppSet as windows application on users' environment.

As users' environment, Cloudpaging Player was installed on a laptop PC (FMV-UH90, Fujitsu Limited). The network connection for users' environment was a wired LAN (1 Gbps) and wireless LAN infrastructure of our institute (IEEE802.11ac, 5 GHz band, 300 Mbps). Results for verification/judgement are shown on **Figure 5**.

On **Figure 5**, time to delivery and boot up application, software profiles (name, version, purpose), size of AppSet, requirements for licensing, some important remarks except software installation process of each virtualized application are shown.

Also, Category 1 is the application that is determined to be usable, Category 2 is the distribution result of running Ubuntu Desktop on Oracle VirtualBox, and Category 3 is the application that is determined to be unusable.

4.3. Summary of Practicality Verification

It was determined that 25 of the 30 types could be delivered and the application could operate normally. However, even applications that are determined to be able to operate normally, some software does not operate correctly even if an AppSet is created solely by the installer. As an example, setting file allocation and access control to local programs (**Figure 5**, 1 - 14, Eclipse) was needed. From these results, it is a fact that the creation of AppSet is based on trial and error.

For Microsoft Office, some software and functions did not work (such as font loading and ODBC cannot be used), and problems such as destroying the Office

Category 1 : Determined that delivered application can be executable properly
License Type Fixed + Run Option : Prefetch For Offline / Network connection : Wired LAN(1Gbps), IEEE802.11ac (5GHz, 300Mbps)

	Software name, version, purpose	File size of AppSet(.stp)	Requirements for activating software (using license server, activation procedure etc.)	Some remarks except software installation process	Time to download Appset + boot up application (Wired.wifi)(sec.)
1-1	7-Zip, x64 18.05, zip file archiver	3.6MB			[2sec,2sec]
1-2	Adash professional + Java Runtime (JRE), 8.1.0, UML modeling tool	290MB		- Operation on PC terminals (in classrooms, net boot windows10 desktop PC) has been also confirmed	[13sec,13sec]
1-3	Android Studio, 1.8.2, IDE to build Android application	1.0GB			[68sec, 104sec]
1-4	ANSYS, 11.0, CAE	1.2GB(main) + 128MB(addon)	Installer requests license server information	- Not recognized even if language patch (separate addon installer) is applied after main installation - After installing ANSYS, install the patch and capture only the patch installation (separate AppSet will be created)	[101sec, 141sec]
1-5	Arduino IDE, 1.8.9, IDE to build Arduino board	178MB		- Binary transfer to Arduino board is not included this verification	[18sec,23sec]
1-6	ImageJ + Java, 1.52a, Image processing tool	69MB		- Normally, an installation method in which Zip file is expanded and copied in Program Files folder, so capture is performed after conversion to exe format using zip file archiver	[10sec, 13sec]
1-7	JW-cad, 8.10b, 2D CAD	25MB			[8sec, 8sec]
1-8	KICAD, 5.1.2, EDA CAD	964MB			[110sec,189sec]
1-9	Rufus, 3.5, bootable USB stick creation tool	1.0MB		- Normally, an installation method in which Zip file is expanded and copied in Program Files folder, so capture is performed after conversion to exe format using zip file archiver - Elevated to administrator privileges at startup (can be elevated)	[3sec,3sec]
1-10	SOLIDWORKS, 2019SP02, 3D CAD and solid modeler	4.4GB	Installer requests license server information		[346sec, 490sec]
1-11	Tableau Reader, 2019.2.1, BI visualization tool (reader)	1.1GB			[69sec, 108sec]
1-12	VESTA, 3.4.7, 3D visualization tool for electronic	16MB		- Normally, an installation method in which Zip file is expanded and copied in Program Files folder, so capture is performed after conversion to exe format using zip file archiver	[8sec,9sec]
1-13	XAMPP, 7.3.6, Web Application (AMP) environment for Windows	167MB		- Windows Firewall setting dialog is displayed at startup (can be changed)	[15sec,24sec]
1-14	Eclipse, 2019-06, IDE	390MB		- Need to Include OpenJDK capture - In some cases, only using the installer does not recognize the capture of the shared library, so place essential files and import it into AppSet is needed. - Allow Windows Defender setting change operation process (MsMgEng.exe) in cloudpaging studio settings is needed	[92sec,90sec] time to boot up is longer than time to downloading AppSet
1-15	Terapad, 1.0.9, Text editor	0.57MB			[2sec,2sec]
1-16	WinSCP, 5.15.2, scp protocol file transfer tool	51MB			[8sec,6sec]
1-17	JMAG, 17.1, FEM tools	2.27GB	Installer requests license server information	- Verified on floating license version (use of Intra FlexLM license server)	[124sec,181sec]
1-18	Mathematica, 12.0, Mathematical tool for technical computing	7.75GB	Installer requests license file	- Verified on floating license version (use of Intra FlexLM license server)	[418sec,735sec]
1-19	MATLAB, 9.6, Scientific calculation tools	12.9GB	Installer requests license file	- Verified on floating license version (use of Intra FlexLM license server)	[850sec,1685sec]
1-20	Sicstus prolog, 4.5.1 staff, Prolog language environment	55MB	Serial number registration required after installation	- After installation, execute the serial number registration command	[9sec,10sec]
1-21	OpenJDK, 12.0.2, Java Development tools, runtime tools	181MB			[10sec,15sec]
1-22	TeX Live, 2019-03, LaTeX environment	3.75GB			[242sec,422sec]

Category 2 : AppSets built to execute ubuntu VM image on Oracle VirtualBox
License Type Fixed + Run Option : Prefetch For Offline / Network connection : Wired LAN(1Gbps), IEEE802.11ac (5GHz, 300Mbps)

	Software name, version, purpose	File size of AppSet(.stp)	Requirements for activating software (using license server, activation procedure etc.)	Some remarks except software installation process	Bootup : success or fail Time to download Appset + boot up application (Wired.wifi)(sec.)
2-1	Virtualbox 5.1 with ubuntu 18.04 LTS Virtualbox VM, 201902, Assuming Ubuntu environment distribution to students' PCs - pattern 1	13.5GB		- AppSet of the virtual machine (ubuntu 18.04 LTS, NAT mode) installed on Virtualbox - If another virtual environment (VMWare Player etc.) has been installed, installed environment will be broken - Device driver (USB, etc.) is unstable depending on any verification environment	Failed Device driver (USB, etc.) is unstable
2-2	Virtualbox 5.1 + ubuntu 18.04 LTS ova image + ova deployment batch program, 201902, Assuming Ubuntu environment distribution to students' PCs - pattern 2	13.5GB		- After Virtualbox (installer) + ubuntu18.04 (ova) is delivered, a batch to deploy ova starts - Ubuntu virtual machine can work well, but long delivery time	OK [850sec,2056sec]
2-3	Virtualbox 5.2 + download and deployment program for ubuntu ova image, 5.2.3.2, Assuming Ubuntu environment distribution to students' PCs - pattern 3	107MB 13.4GB(ova)		- In addition to Virtualbox 5.2, install a program (.NET) to download and deploy ubuntu 18.04 LTS(NAT mode) ova - Elevated to administrator privileges at startup (can be elevated) - Fine tuned for USB and network device driver stabilization	OK Appset delivery [18sec,24sec] Download + install ova [390sec,940sec]
2-4	Virtualbox 6.1 + download and deployment program for ubuntu ova image, 6.1.1.0, Assuming Ubuntu environment distribution to students' PCs - pattern 4	195MB 13.4GB(ova)		- In addition to Virtualbox 6.1, install a program (.NET) to download and deploy ubuntu 18.04 LTS(NAT mode) ova - Elevated to administrator privileges at startup (can be elevated) - Fine tuned for USB and network device driver stabilization	OK Appset delivery [20sec,26sec] Download + install ova [410sec,948sec]

Category 3 : can not executed or with condition

	Software name, version, purpose	File size of AppSet(.stp)	Requirements for activating software (using license server, activation procedure etc.)	Some remarks except software installation process
3-1	Microsoft Office, 2019 Professional, Office Suite	1.2GB		NG (can not execute) - Conflict with pre-installed Office2016, out of memory in Word, font reading failure, ODBC not available in Excel, etc.
3-2	Microsoft Visual Studio, 2017, IDE for Windows App	3.7GB	Product ID registration required	NG (can not execute) - Product ID installation command (StorePID.exe) is failed to execute
3-3	Microsoft Visual Studio, 2019, IDE for Windows App	3.3GB	Product ID registration required	NG (can not execute) - Product ID installation command (StorePID.exe) is failed to execute
3-4	Distribution of documents for 1st. year students, 2018	62MB	No installer, but the folder containing document data will be delivered	Executable with conditionally - Do not work on some verification environment

Figure 5. Virtualized applications and verification results.

environment installed in the user environment were confirmed. We concluded that virtualization was difficult. For Microsoft Visual Studio, it failed in executing the command to register the product ID when creating the AppSet. We also concluded that virtualization was similarly difficult.

With regard to the distribution of Ubuntu Desktop on Oracle VirtualBox, there are some cases where the problem is not resolved (in particular issue of USB device driver) even after trial and error (**Figure 5, 2-1**), or distribution is possible but it takes much distribution time (**Figure 5, 2-2**). Judgment of both practicality and operational policy requires careful consideration, so will be our future work.

In the wireless LAN environment, the time required from distribution to boot up application is about 2 to 3 seconds for small scale (AppSet file size is several MB) software and about 3 minutes for medium scale (AppSet file size is 1 GB) software.

However, it takes about 27 minutes for MATLAB (12.9 GB) with the largest AppSet file size. It is possible to use in a lecture if it is necessary to complete the software distribution before the lecture.

5. Conclusion

In this paper, we described an application virtualization system (Cloudpaging) adopted in an educational system to support bring and use students' laptops (BYOD) in our institute. Since practicality needs to be verified in an actual system, in this paper 30 types of applications were selected and their availability was confirmed. As a result, the delivery of the application in various installation forms and usage forms was confirmed, and the effectiveness of the application virtualization system was shown.

In the future, it will be necessary to create further operation examples, select software, and establish an operation check procedure.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- Fukuda, et al. (2019). Wireless LAN Communication Experiments for the Development of a BYOD Environment. The Internet and Operation Technologies by Constructing Resilient Information Systems. *IPSJ Journal of Information Processing*. (In Japanese)
- Hayashi, et al. (2019). Updating the Computer System for Education and Research with a Focus on Supporting the Need for Laptops. *IPSJ SIG Technical Report, 2019-IOT-46*. (In Japanese)
- Kai, et al. (2019). Report on the Initial Setup Workshop 2018 for Laptop Users. *Bulletin of Kyushu Institute of Technology Information Science Center No. 31*. (In Japanese)
- Kyushu Institute of Technology (2019). *Kyutech BYOD (Bring Your Own Device)*. (In Japanese) http://www.kyutech.ac.jp/student-cheer/kyutech_byod.html
- Nakamura, T. (2010) Introduction of Windows Applications Using Diskless PCs. *Bulletin of Kyushu Institute of Technology Information Science Center, No. 22*. (In Japanese)
- Numecent Inc. (2018). *Cloudpaging*. <https://www.numecent.com/cloudpaging/>
- Ohashi, et al. (2018). BYOD in Kyushu Institute of Technology. *Bulletin of Kyushu Insti-*

tute of Technology Information Science Center No. 30. (In Japanese).

Tomishige, et al. (2018). An Attempt to Evaluate the Performance of the Recommended Specification of the Notebook Computer. *2018 Annual Conference of AXIES, MP-11.* (In Japanese).