Deploying Edge-based Virtual Desktop Infrastructure

James Blakley, Scott Haas[†], Victor Firoiu[‡],
Mahadevan Iyer[‡], Daniel Beveridge[‡], Girish Narkhede[‡], Shilpa George,
Thomas Eiszler, Jan Harkes, J. Ray Scott[†], Mahadev Satyanarayanan

[†]CMU Computing Services [‡]VMware, Inc.

January 2023
CMU-CS-23-101

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This research was supported by the National Science Foundation (NSF) under grant number CNS-2106862. This work was done in the CMU Living Edge Lab, which is supported by Intel, ARM, Vodafone, Deutsche Telekom, CableLabs, Crown Castle, InterDigital, Seagate, Microsoft, the VMware University Research Fund, and the Conklin Kistler family fund. Any opinions, findings, conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the view(s) of their employers or the above funding sources.

Keywords: edge computing, mobile networks, latency, bandwidth, edge-native applications, virtual machines, virtual desktop infrastructure

Abstract

Carnegie Mellon University's (CMU) *Virtual Andrew* service uses VMware Horizon Virtual Desktop Infrastructure (VDI) in coursework, research, and administration. It provides pre-configured, no-install access to a variety of restricted-license applications, such as computer-aided design (CAD) tools. This service is typically used from on-campus computing clusters and faculty/student offices with LAN connectivity to CMU's private cloud. The Covid-19 pandemic forced most members of the university community to work from their homes, with highly-variable last-mile connectivity. The change from on-campus LAN-access to off-campus WAN-access exposed limitations of VDI as a remoting service. These limitations led to a three-way collaboration between VMware, the CMU Living Edge Lab, and CMU Computing Service to investigate how Edge Computing could enable a better VDI experience. This technical report discusses our learnings.

1 Introduction

When the Covid-19 pandemic came to the Carnegie Mellon University (CMU) campus, the university community suddenly became more dependent on the CMU *Virtual Andrew* service. This service uses VMware Horizon Virtual Desktop Infrastructure (VDI) to provide remote access to a variety of applications. With most of the university community working away from campus during the pandemic, this dependence exposed the limitations of VDI as a remoting service. These limitations led to a three-way collaboration between VMware, the CMU Living Edge Lab [6], and CMU Computing Services to investigate how Edge Computing could enable a better VDI experience. This technical report discusses VDI background, the remote use case challenges, our work towards resolving these challenges in the context of Horizon, and open issues.

2 Virtual Desktop Infrastructure

For software, both learning by users and the rich ecosystem of compatible software and data formats constitute legacy. For many enterprises that use the software, incompatible change threatens to be highly disruptive. There is a high business value in creating new products that remain compatible with legacy software. Perhaps the best example of such business value is the IBM mainframe, whose legacy reaches back half a century. As recently as 2020, mainframes accounted for a substantial fraction of IBM's profits [13]. In early 2020, the website for these products stated [11]:

> *"The IBM Z family maintains full backward compatibility. This means that current systems are the direct, lineal descendants of System/360 announced in 1964, and System/370 from the 1970s. Many applications written for these systems can still run unmodified on the newest IBM Z system over five decades later."*

Of particular significance are authoring tools used by professional workers. Well-known examples of authoring tools include the Microsoft Office Suite, Adobe Acrobat, Adobe Photoshop, GIMP, AutoDesk Fusion 360, Blender, and software development tools such as the gcc compiler. Modern versions of these tools are used today on Windows, MacOS, and Linux desktops and laptops. They are the primary vehicles through which many millions of professional workers, such as architects, engineers, lawyers, creative artists, and software developers, deliver their productive value into the global economy. Although the market for authoring tools is much smaller in relative terms than the market for information consumption tools (tens of millions of users, rather than billions), it is still a large market in absolute terms. It is also a market with high profit margins because of the productive value of professional users.

For authoring tools of the personal computing era, Microsoft Windows is the dominant legacy environment. While software created for Apple MacOS, Linux, and other flavors of Unix are also important, we will focus on Windows for brevity. The number of distinct Windows applications that have been created lies between 16 million [16] and 35 million [9]. Even the smaller of these estimates is an enormous number. There have been a number of efforts to rewrite widely used authoring tools in a way that leverages cloud computing, but retains their user-facing and data-facing functionality and interfaces. Microsoft Office 365 is the best known of these transformations. Such rewriting requires crisp user interactions to be preserved, while splitting a

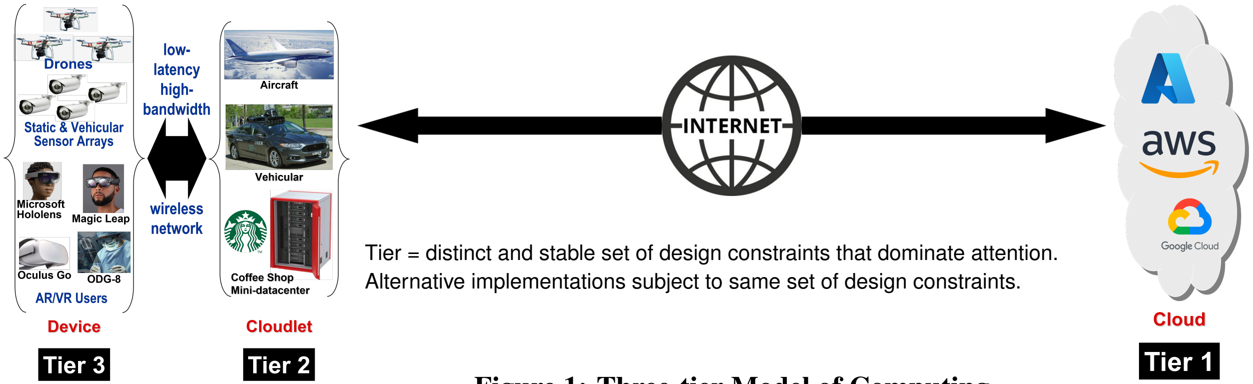
Drones
Static & Vehicular Sensor Arrays
Microsoft Hololens
Magic Leap
Oculus Go
ODG-8
AR/VR Users
Device
Tier 3
low-latency high-bandwidth
wireless network
Aircraft
Vehicular
Coffee Shop Mini-datacenter
Cloudlet
Tier 2
INTERNET
Tier = distinct and stable set of design constraints that dominate attention.
Alternative implementations subject to same set of design constraints.
aws
Google Cloud
Cloud
Tier 1

Figure 1: Three-tier Model of Computing

monolithic application into a cloud-based back-end separated by the high latency of the Internet from a Web browser.

This task is best understood relative to the 3-tier model of computing shown in Figure 1 [20]. In the context of this figure, the simplest way to represent standalone personal computing of the early 1990s is to remove Tier-1 and Tier-2, and to replace the devices at Tier-3 by desktops and laptops. Public and private cloud computing adds back Tier-1 to the figure. While the effort to refactor a personal computing application into a Tier-3 front-end and a Tier-1 back-end can be successful, it is very expensive and can only be profitable for the most widely used applications. For less widely used applications, the return on investment (ROI) is too low for the effort to be worthwhile.

Figure 2 illustrates the dilemma faced by the users and vendors of legacy authoring tools. Although we do not have sufficient empirical data to plot the exact shape of this curve (e.g., Zipf's Law, a Pareto distribution, or an exponential are all plausible), its salient features are high skew and an extremely long tail. In other words, a relatively small number of applications, possibly far below one percent of the 16 million or 35 million Windows applications mentioned earlier, are very heavily used. Only for these applications is the ROI of rewriting for a cloud-based implementation

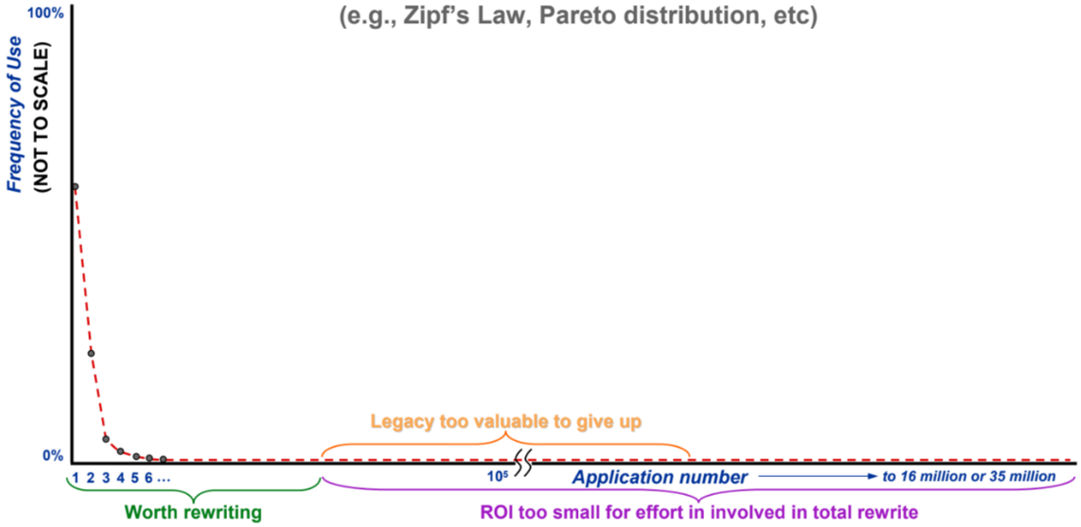

Figure 2: Target Applications for VDI (conceptual figure, not to scale)

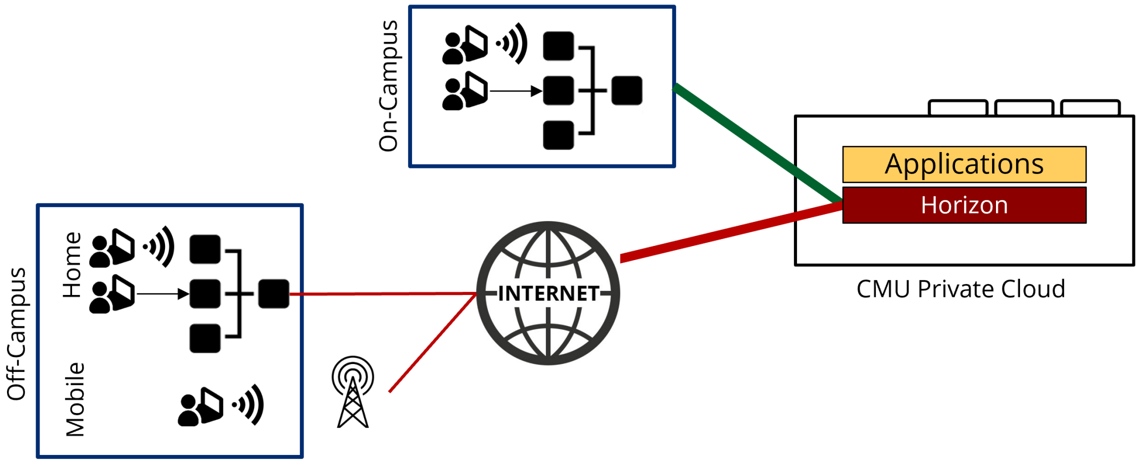

Figure 3: Virtual Andrew Use Cases

worthwhile. For the remaining millions of applications, many of which remain important to users and enterprises, rewriting is unlikely to be profitable.

VDI is a solution that leverages virtual machine (VM) technology to bring the benefits of cloud computing to these applications. At its simplest, VDI involves three steps:

- encapsulating an application and its operating system in a virtual machine (VM) image.
- launching an instance of this VM image at Tier-1, and managing it.
- interacting with the VM instance from Tier-3 using a remote desktop protocol. We use the generic term "RDP" for this remote desktop protocol, spanning a wide range of possibilities such as VNC [18], SPICE [15], PCoIP [7], VMware Blast [24], Microsoft RDP [7], etc.

VDI supports use of applications that cannot be executed at Tier-3. Example reasons for this constraint include license limitations (e.g., pooled licenses), device compatibility limitations (e.g., hardware instruction set, operating system/version, performance, GPU acceleration), and data movement restrictions (e.g., "crown jewel" design data and GDPR/HIPAA restrictions).

3 Virtual Andrew: VDI at Carnegie Mellon University

3.1 Overview

CMU Computing Services operates the *Virtual Andrew* VDI service [4] to enable access to applications required for course and lab work, research, business and education operations, and other uses. Virtual Andrew uses the VMware Horizon platform [25] deployed in CMU's private cloud to support Windows virtual desktops accessed via Windows, Mac, Linux, and browser clients. The currently supported applications are shown in Table 1, with the design and engineering applications highlighted in bold red.

CMU users typically access Virtual Andrew via wired and wireless LAN access from on-campus offices or computer labs. However, during the Covid pandemic, off-campus internet access from homes and other locations increased substantially. These access methods are shown in Figure 3.

7zip	**GIMP**	Python
ACDLabs ChemSketch	Google Chrome	Pyzo
ActivePerl	Google Workspace for Education	R
Adobe Acrobat Pro DC	GSView 5	**Rhinoceros**
Adobe Creative Cloud	IBM SPSS	RStudio
Adobe Reader DC	ImageJ	SAS 9.4
Adobe Spark	Inkscape	SAS On Demand for Academics
Alice 3	Java JDK	SimaPro
AMPAC	Java JRE	**Autodesk 3ds Max**
ANSYS	Jmol	**Autodesk AutoCAD**
ANSYS Granta Edupack	LightSIDE	**Autodesk Inventor Professional**
ANSYS Lumerical	LyX	**Autodesk Maya**
ArcGIS Pro	Maple	**Autodesk Mudbox**
Arduino	Mathematica	**Autodesk Inventor Nastran**
Audacity	Matlab	**Autodesk Revit**
Blender	Max 8	**Unity**
Campus Printing	Mendeley Desktop	**SolidWorks**
COMSOL	Microsoft Office	Texmaker
Concord	Minitab	VisIt
Cura	Netica	VLC
DNA Master	NVDA	Visual Studio Code
Eclipse	ParaView	Weka
Emacs	Pd	WinAVR
Emerald Cloud Lab (ECL)	Preform	X-Win32 2014
FastX	Processing	ZBrush
Firefox	PuTTY	

Bold red indicates engineering and design authoring applications

Table 1: Supported Virtual Andrew Applications

3.2 Limitations of Virtual Andrew

Virtual Andrew provides good service in on-campus use. However, off-campus use is subject to limitations that impact user experience negatively. In some cases, this impact can render the experience inadequate for sustained use. This section describes these limitations and describes how EdgeVDI mitigates them.

We have found that, for many VDI use cases and applications, consistent end-to-end latency less than 150 ms is sufficient to provide acceptable user experience [23, 12]. However, for engineering and design applications with *rich graphics* and *significant user interaction*, consistent end-to-end latency less than 50 ms is necessary to provide an experience nearly equivalent to on-campus access. For Virtual Andrew users, the on-campus experience (Figure 3) is adequate for most applications. Well designed Wi-Fi and wired LAN networks provide sub-10ms round trip latencies with high bandwidth, low jitter and low packet loss. Even highly interactive engineering and design applications give acceptable user experience in this setting.

In the off-campus use cases, VDI user experience is greatly impacted by *network proximity,* where "nearness" corresponds to low latency and high bandwidth. In-home use with good network proximity to CMU's private cloud can give a good user experience. Poor network proximity, however, can hurt user experience. In some off-campus use cases, round trip times (RTT) can be highly variable, with tails of the distributions stretching to 100's of milliseconds [3]. Improving network proximity by hosting desktop VMs on a nearby cloudlet (Tier-2), rather than the private cloud on campus (Tier-1), can mitigate the experience degradation.

It is important to note that physical proximity, by itself, is neither necessary nor sufficient to ensure network proximity. At the speed of light in fiber, one millisecond translates to 200 km of physical distance. With a 5G or Wi-Fi first hop below 5 milliseconds one-way, a wireless end-to-end RTT below 15 milliseconds can be achieved even with a cloudlet that is physically quite far away. In fact, this physically distant cloudlet may be a better choice than a nearby cloudlet with a heavily-loaded ingress network. That said, it is typically the case that more distant cloudlets are reached over more network hops. Each network hop is a potential bottleneck. Each adds some queueing delay, leading towards a heavy-tailed RTT distrbution. Hence, there is a weak correlation between physical and network proximity. By reducing the number of potential congestion points for network traffic, improved real-time network conditions such as end-to-end latency, jitter, and packet loss can be achieved. In many networks, connections between carriers may be well outside the user's metro area [3, 17] – adding many network hops between user and cloudlet. Carrier deployed *local breakout* [26] moves interconnect closer to the user and cloudlet and can yield a short-tailed RTT distribution.

Additionally, a first-hop mobile wireless network can add 30+ ms for LTE and 15-20 ms for 5G. Since this delay is mostly driven by the wireless link between the device and the mobile radio access network (RAN) [22], it can only be mitigated by reducing the wireless link latency. This would typically require migrating from LTE to 5G.

In summary, improving the Virtual Andrew experience for off-campus users involves two complementary steps. First, it involves placement of VM-hosting cloudlets in close network proximity to users, with the goal of reducing the number of networking hops on the end-to-end path. Second, it involves improved last-mile connectivity such as 5G.

3.3 User Mobility and Roaming

As a VDI user moves from one location to another, the network distance between the client and VM-hosting server can grow. Movement can mean both mobility (e.g., working continuously on a long-distance train trip) and roaming (e.g., a visit to another university in another city). In these cases, the Virtual Andrew user can see high latencies and experience degradation between their physical location and on-campus VM-hosting server. This latency can be mitigated by a movement of the desktop VM to a cloudlet closer to the user. To achieve mobility, a VM-migration and a session-migration capability between original and new cloudlet are needed [19]. Roaming is somewhat easier, requiring only "suspend/resume" capability [14, 21].

3.4 VMware Horizon VDI

As mentioned earlier (§ 3.1), Virtual Andrew uses VMware Horizon as its VDI platform. Like other existing commercial VDI offerings, Horizon is designed with a modular but centralized

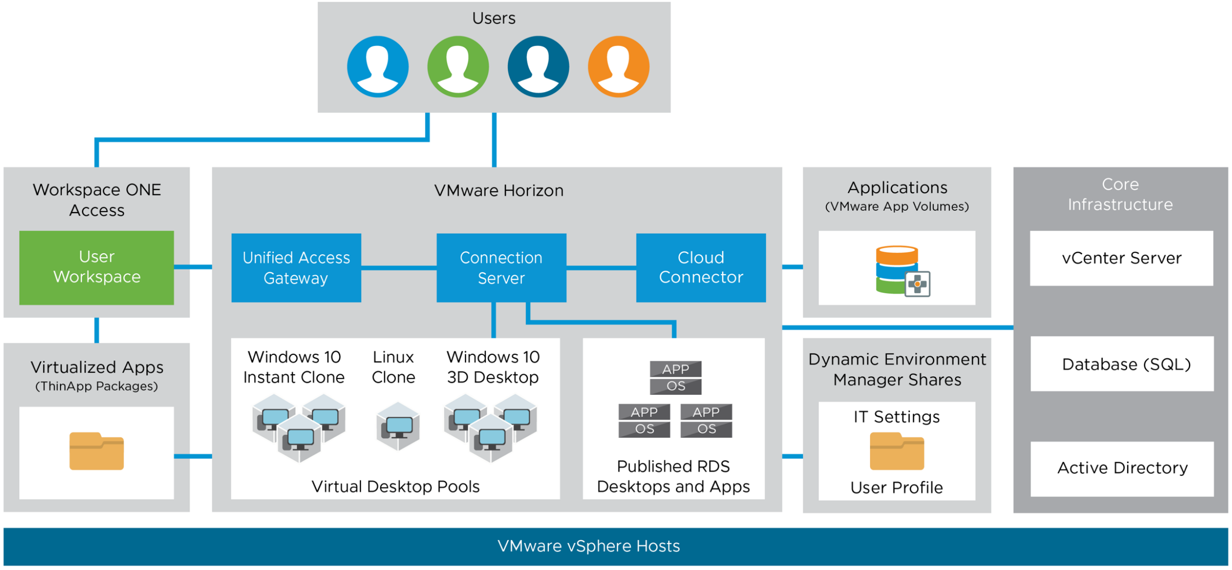

Figure 4: VMware Horizon Architecture

architecture. An underlying assumption is that deployment occurs in a relatively homogeneous environment, with excellent network proximity (i.e., low latency and high bandwidth) between VDI servers and users. Other important assumptions, driven by enterprise-scale deployment considerations, are the need for integration with services such as Active Directory(AD), ease of management, data security, and privacy. The VMware Horizon architecture, prior to EdgeVDI modifications, is shown in Figure 4.

When remote access causes violation of the network proximity assumption between the desktop VMs and users, the VDI architecture has to change to restore the assumption. This requires the *VMware Horizon* "pod" in Figure 4 to move closer to the user (i.e., to the "other side" of the Internet). As the pod is disaggregated from the remaining modules, the requirements for the other modules must be reconsidered. These considerations include module scope (e.g., an AD may be scoped to cover an entire enterprise, not just the VDI service), administrative domains (e.g., cloudlets may be deployed in a telco data center outside the direct control of enterprise IT), and module interconnect bandwidth requirements (e.g., some applications such as a large-scale CAD application may require substantial data transfers to the virtual desktop). An important goal of our work is to learn what modifications are needed to Horizon in order to support the new requirements and constraints of EdgeVDI.

4 The EdgeVDI Project

At the Living Edge Lab, we set out to explore how edge computing can benefit unmodified legacy applications that were originally written for a personal computing environment with a particular focus on the authoring tools discussed in the introduction.

4.1 Key Questions

There are a number of research and business questions that need to be answered before EdgeVDI can become a widely accepted solution to the VDI limitations described earlier (§ 3.2). Relative to the off-campus use cases shown in Figure 3, the salient questions are:

1. Which VDI user applications have degraded experience in the off-campus cases? Are there particular user tasks that experience more degradation than others? What system characteristics cause this degradation? Does edge computing mitigate this degradation? By how much?

2. Is it possible to automatically detect poor user experience without the need for explicit user input? Is this detection a good cue for triggering VM migration to a "better" cloudlet?

3. How must a commercial VDI offering such as Horizon be re-architected for edge deployment? What parts must be distributed to the edge and which should remain centralized?

4. How does user mobility impact VDI? Does the physical movement of a user from one location to another necessitate the migration of the corresponding VM? How can this movement be accomplished in a way that maintains user experience quality during the migration?

4.2 Experimental EdgeVDI Environment

To study these questions, we built an EdgeVDI environment at the Living Edge Lab (See Figure 5). This environment is built on the *PyEdgeSim* environment [10]. The EdgeVDI environment consists of the following components:

- A dedicated *lab VMware Horizon cloudlet* to host VDI VMs. This cloudlet runs the same version of Horizon used in the production Virtual Andrew implementation but provides a dedicated sandbox for experimental uses.

- Two networks are used to enable client access to the Horizon cloudlet and replicate the use cases in Figure 3. The first, connected over the CMU wired and Wi-Fi access network and routing through an *interference generator* (See Section 4.3), allows us to connect clients with minimal end-to-end latency, jitter, and bandwidth restrictions. We then use the interference generator to introduce additional network impacts under automated program control. This network also enables access to the environment through commercial wired and wireless network operators connected via commercial interconnect points to the CMU campus.

  The second network, the *Living Edge Lab Network* (LELN) [5], is a fully deployed indoor and outdoor private LTE Citizen's Band Radio Service (CBRS) [8] network on and around the CMU campus. This network delivers best-in-class LTE end-to-end latency and avoids long interconnect times between different carriers. Figure 6 shows the end-to-end latency of this network in comparison to a commercial LTE network. It allows us to test EdgeVDI in a real-world edge computing environment outside the boundaries of our lab.

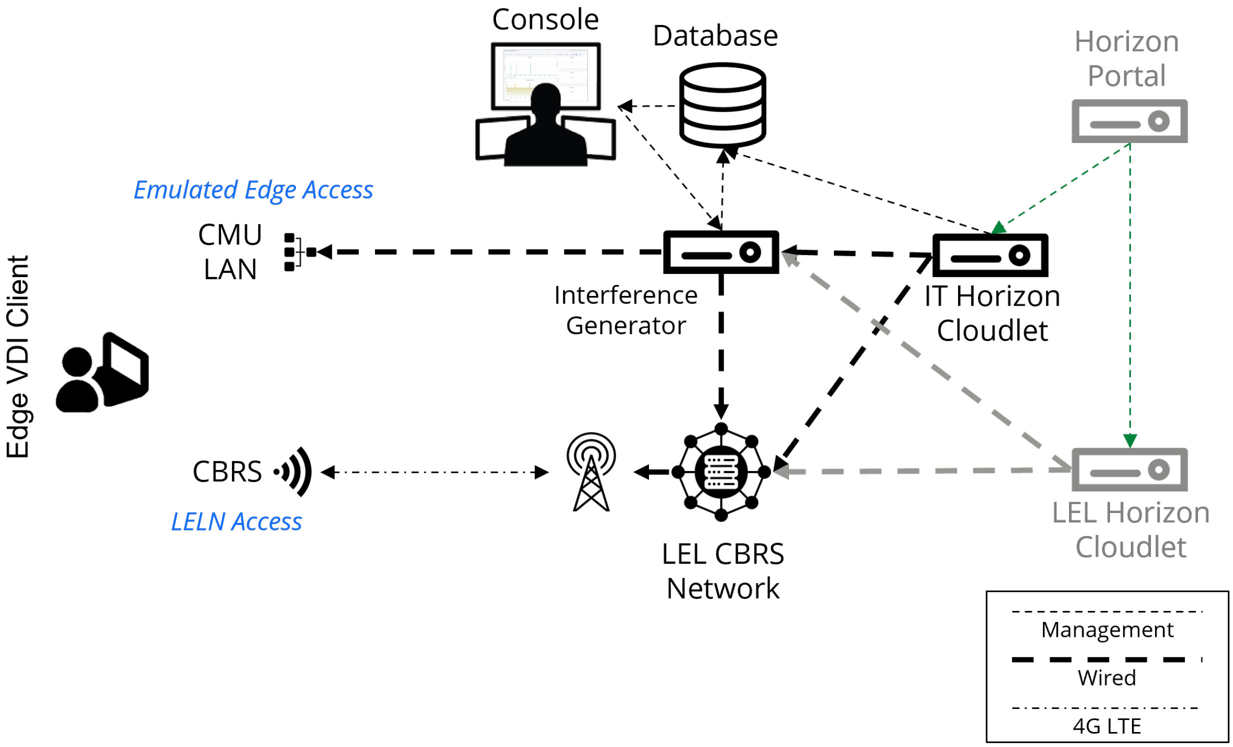

Figure 5: Living Edge Lab EdgeVDI Environment

- A specially configured Windows VM instance that includes our needed applications (e.g., AutoDesk Fusion 360 and Blender) and is launched by default on the lab cloudlet.
- Linux and Windows *Horizon clients*. These clients run the default production Horizon client software and are configured to gather VDI *perfmon* and *blast* performance information.
- A database (*InfluxDB*) and dashboard (*Grafana*) that are used to collect data from and monitor our experiments.
- FUTURE: A second Horizon cloudlet to be used in investigating issues related to user mobility between cloudlets.

4.3 Experience Interference Generator

The network access methods available in the Living Edge Lab – commercial LTE, our private LTE CBRS network, and the CMU wired and wireless campus networks – provide different but fixed performance profiles. They allow experimentation in real-world network conditions. However, testing under other network configurations (e.g., 5G networks) or under non-ideal network conditions (e.g, load, poor connectivity) is not readily controllable in a production network. Poor VDI user experience often occurs in conjunction with these exceptional network conditions. To enable creation of arbitrary network conditions at will, we implemented an *experience interference generator*. This interference generator uses the *AdvantEDGE* mobile edge emulation platform[10]

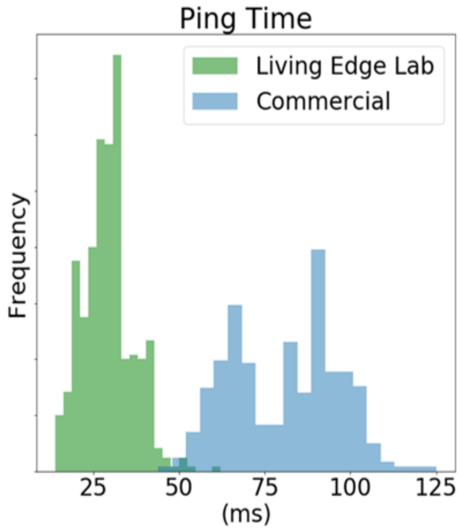

Figure 6: Measured LEL End-to-End Latency

Name	Description
Wired LAN aka "Zero"	No Added Interference 0ms RT Latency; 0ms Jitter; 0% Packet Loss
4G Living Edge Lab Network	4G LTE Network w/Local Breakout 30ms RT Latency; 2ms Jitter; 0% Packet Loss
4G Commercial Network with Local Breakout	4G LTE Network w/Local Breakout 50ms RT Latency; 2ms Jitter; 0% Packet Loss
4G Commercial Network with Remote Interconnect	4G LTE Network w/Remote Interconnect 140ms RT Latency; 2ms Jitter; 0% Packet Loss
5G Commercial Network with Local Breakout	5G CBRS Network w/Local Breakout 20ms RT Latency; 2ms Jitter; 0% Packet Loss
5G Commercial Network with Remote Interconnect	5G CBRS Network w/Remote Interconnect 120ms RT Latency; 2ms Jitter; 0% Packet Loss

Table 2: Baseline Network Profiles

with the automation framework described in [3]. Based on the network parameters in Tables 2 and 3, we created a tool to generate a random mix of interference profiles. Once generated, these profiles can be "played" while a Horizon user is interacting with the Horizon VM. The user experience impact can then be directly observed or detected with the Machine Learning Experience Predictor described in Section 4.4.

4.4 Automating User Experience Measurement

Human users are the subjective arbiters of "good" user experience. However, in scaled VDI environments, relying on human reporting of bad experience is unreliable. Users may never report bad experiences – choosing instead to suffer the degradation or abandon use of the service. The service provider, however, would prefer to have a good indicator of whether the service provides

Type	Approach	Ranges
Bad Signal	Randomly vary latency, jitter and packet loss	Latency: 200-1000ms Jitter: 5-50ms Packet Loss: 5-20%
Load	Randomly introduce UE-bots to congest the link to the Horizon client	On baseline 1Mbps link: Traffic: 50-3150 kbps
Combo	Combine Bad Signal and Load	Latency: 200-1000ms Jitter: 5-50ms Packet Loss: 5-20% On baseline 1Mbps link: Traffic: 50-3150 kbps

Table 3: Interference Profiles

good experience to all and under what conditions it does not. Service providers often choose proxy experience indicators (e.g., end-to-end latency, jitter, network throughput). We sought to determine whether a large number of proxy indicators could be combined into a machine learning model that would provide a strong predictor of human perceived experience.

4.4.1 Machine Learning Experience Predictor

Our *Machine Learning Experience Predictor (MLEP)* uses system data to determine in real time whether a specific user's experience is likely to be degraded. This prediction could be aggregated and used as an overall indicator for the system or used as a diagnostic tool to identify bad pockets of service. As an initial effort, we identified a candidate set of system features from the VDI desktop VM instance. These approximately 200 features were drawn from `perfmon` and the RDP protocol used by Horizon (i.e., the Blast protocol). We also identified design application use cases from Autodesk Fusion 360 [2] and Blender [1] to serve as dataset collection cases.

A subset of the authors, playing the role of users, interacted with these applications and tagged the experience as either good or bad. This was used as the basis of a preliminary case study to provide insights towards the design of a formal EdgeVDI user study. Since the subjects of the preliminary case study were the authors themselves, no IRB approval was needed.

During user interaction, the network characteristics were varied using the interference generator described in Section 4.3 to create randomly varying good and bad experiences. Tagging was accomplished using a USB-connected foot pedal. When the user detected a poor user experience, they depressed this pedal. When a good user experience resumed, they released it. This binary indicator is the classification that the MLEP seeks to replicate. This data was used to create the disjoint training and test sets for the MLEP. Using this dataset, we trained a support vector machine (SVM) to classify experiences. This classifier and our results are described in Section 4.4.2.

During the data collection process, we observed some interesting effects. First, users fatigued quickly during the tagging sessions. The simultaneous task of application interaction and foot pedal tagging was tiring. As a result, we kept those sessions to no more than 15-20 minutes. Second, there is a noticeable lag between the experience becoming bad (or returning to good) and the user response tagging the transition. Users tend to wait for a short period to confirm that the transition has actually happened before reacting. We do not believe that this lag impacted the performance of the predictor.

TP True positive: the number of data points correctly identified as poor experience.
FP False positive: the number of data points incorrectly identified as poor experience.
TN True negative: the number of data points correctly identified as good experience.
FN False negative: the number of data points incorrectly identified as good experience.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

Train data	Test data	Accuracy	Precision	Recall
80% UC1	20% UC1	97.4 (0.003)	96.8 (0.009)	96.1 (0.007)
80% UC2	20% UC2	99.2 (0.001)	98.6 (0.005)	99.2 (0.002)
80% UC1	100% UC2	89.9 (0.037)	81.3 (0.061)	93.7 (0.020)
80% UC2	100% UC1	79.4 (0.008)	64.6 (0.007)	96.6 (0.053)
5 users	1 user	80.5 (0.110)	76.7 (0.248)	82.2 (0.236)

Figures in parentheses are standard deviations from five runs.

Table 4: MLEP Result Summary

4.4.2 Experimental Results

As mentioned in Section 4.4.1, we conducted a preliminary case study to measure the performance of the MLEP in the EdgeVDI environment. For the study, we collected 15 – 20 minutes of user experience data from 6 users across two application use cases. For the prototype implementation of the MLEP we used a support vector machine (SVM) classifier to predict the user experience. Approximately 200 features drawn from perfmon and the Blast protocol were given as inputs to the predictor. We also used principal component analysis (PCA), to remove correlated features and lower the dimensionality of the input data. The foot pedal tags are used as ground truth labels, to measure the prediction accuracy of the MLEP. Table 4 presents the results from 5 runs of each experiment, varying the random seed across runs. Each row corresponds to a different configuration of training and test data.

Use case 1 (UC1): For the first application use case we use the Fusion 360 design software application, developed by Autodesk. This software is typically used to create 3D computer-aided designs (CAD) by many professions such as architects, mechanical, electrical, civil, and aerospace engineers. As mentioned earlier, we use the data collected from 6 users to train and evaluate the predictor. For each run of this experiment, we randomly shuffle the data and use 80% to train and remaining 20% to test the performance of the predictor. The first row of Table 4 gives results when training and testing are both done on UC1. With this configuration, the predictor has a mean accuracy of 97.4%, mean precision of 96.8% and a mean recall of 96.1%.

Use case 2 (UC2): For the second use case we use the Blender software, which is a graphics software tool used for creating animated films, 3D-printed models and video games. The experimental setup is same as the previous use case. The results for UC2 are shown in the second row of Table 4. The results show a mean accuracy of 99.2%, mean precision of 98.6% and a mean recall of 99.2%.

Prediction across use cases: Next we answer the question, “Is the MLEP application independent?” Specifically, can a predictor trained for a particular use case be used to predict the user experience for another use case? To test the hypothesis, we use the MLEP trained on `UC1` data to predict the user experiences of `UC2` data and vice versa. For each run of the experiment, we use 80% of the use case data to train the predictor and 100% of the other use case data to test it. Rows 3 and 4 of Table 4 gives the results across use cases. The average of these two rows gives a cross-use case predictor performance as follows: accuracy of 84.6%, precision of 73%, and recall of 95.2%. In other words, the predictor is able to correctly identify a true bad experience 95% of the time; however, only 73% of the experiences predicted as bad are truly bad.

Unseen User: In the previous experiments we used data from all users to train or test the predictor. We also explored the performance of the predictor for unseen users. For each of the experiments, we train the MLEP using data from five users and hold off data from the remaining user for testing. Across all six users, the predictor has a mean accuracy of 80.5%, precision of 76.7% and recall of 82.2%.

4.4.3 Limitations

While these initial results are promising, there are some limitations and unknowns:

- Can the results be generalized to a large number of different users and different application use cases?
- Can they be generalized across different system configurations? (e.g., different client and VM server types)?
- In a scaled system, will the same system features predict experience when many users are simultaneously active?

A full-fledged user study on a deployed EdgeVDI system could answer these questions.

4.5 Key Learnings

During the course of the project to date, we have had substantial opportunity to interact with and observe VDI in the current Virtual Andrew use cases of Figure 3 and in real and simulated EdgeVDI environments using a variety of applications. We have the following observations. These observations are made in the context of the baseline on-campus use cases which give Virtual Andrew users an acceptable user experience for all supported applications.

- There is a noticeable degradation in experience in all of the off-campus uses compared to on-campus uses.
- This degradation is most noticeable in the mobile use cases particularly in periods of poor coverage and high traffic load. The degradation is driven by both network distance between user and the CMU private cloud and the radio access network latency.
- However, even the in-home use cases experience some degradation despite high quality broadband access. We believe this to be driven mostly by high network distance.

- EdgeVDI can effectively mitigate the network distance issues. Radio access network latency can only be mitigated by moving to a lower latency wireless network (e.g., 5G)
- Authoring tools – especially those with high levels of user interactivity and rich graphical interfaces – are highly impacted by long RTTs. However, even lightly interactive streaming applications can suffer due to excessive queuing and packet drops.
- It appears feasible to create a reliable automated predictor of bad user experience from system measurements. However, much work remains to determine whether this can be generalized and scaled across applications and users.

5 Future Work and Conclusion

Now that the pandemic has evolved to an endemic, university life has moved to a new on-going normal of hybrid on-campus and off-campus. This means that our work in EdgeVDI will continue. There remain research questions to be answered and business problems to be solved.

On the research side, we plan continued work on automated detection of user experience, user mobility, and the interplay between mobile networks and applications. These are important issues for EdgeVDI and other interactive edge-native applications. In 2023, we are focused on expanding experience measurement work, desktop migration while mobile and roaming, and performing formal user experience testing with current Virtual Andrew users.

On the business side, our key focus will be on re-architecting the VDI stack for the edge and understanding the commercial viability of EdgeVDI by working with carriers and other ecosystem players. In 2023, we plan a joint test of a new edge-centric Horizon architecture in the CMU Computing Services environment and to engage with mobile and wired carriers to pilot EdgeVDI with students, faculty, and staff in their natural usage environments.

There are, also, natural extensions of this work into other edge-native applications beyond VDI – including virtual, augmented, and mixed reality – those applications also requiring high interactivity and rich interfaces.

We have learned enough about EdgeVDI already to believe that it will be a compelling edge-native application for users of interactive, graphically rich authoring applications. Remote work will not end during the new endemic reality and making people more productive regardless of where they work will be an important driver for the foreseeable future.

References

[1] BLAKLEY, J. Spiky Ball Sculpt Use Case Video, 2022. https://www.youtube.com/watch?v=oa-VSO3jh2o.

[2] BLAKLEY, J. Tippy Ashtray Orbit Use Case Video, 2022. https://www.youtube.com/watch?v=-1NCSn8enF4.

[3] BLAKLEY, J. R., IYENGAR, R., AND ROY, M. Simulating Edge Computing Environments to Optimize Application Experience. Tech. Rep. Technical Report CMU-CS-20-135, School of Computer Science Carnegie Mellon University, 2020.

[4] Carnegie Mellon University Computing Services. Virtual Andrew Service. https://www.cmu.edu/computing/services/endpoint/software/virtual-andrew.html.

[5] Carnegie Mellon University Living Edge Lab. The Quest for Lower Latency: Announcing the New Living Edge Lab Wireless Network, 2021. https://www.cmu.edu/scs/edgecomputing/articles/quest_for_low_latency.html.

[6] Carnegie Mellon University Living Edge Lab. Edge Computing @ CMU Living Edge Lab, 2022. https://www.cmu.edu/scs/edgecomputing/.

[7] Casanova, L., Kristianto, E., et al. Comparing RDP and PcoIP protocols for desktop virtualization in VMware enviroment. In *2017 5th International Conference on Cyber and IT Service Management (CITSM)* (2017), IEEE, pp. 1–4.

[8] Fierce Wireless. What is CBRS?, 2020. https://www.fiercewireless.com/private-wireless/what-cbrs.

[9] Fortin, M. Windows 10 Quality approach for a complex ecosystem, November 2018. http://bit.ly/3QQ0Jmv, Last Accessed January 7, 2023.

[10] Gazda, R., Roy, M., Blakley, J., Sakr, A., and Schuster, R. Towards Open and Cross Domain Edge Emulation–The AdvantEDGE Platform. In *2021 IEEE/ACM Symposium on Edge Computing (SEC)* (2021), pp. 339–344.

[11] IBM. IBM Z Platform, Accessed Feb 3, 2020. https://www.ibm.com/docs/en/linux-on-systems?topic=knpbptr-overview.

[12] Iyer, M. VMware Presentation @ The OEC Fall Workshop, 2021. https://www.cmu.edu/scs/edgecomputing/resources/oec-archive/.

[13] Jhonsa, E. IBM Gains as Mainframes Help Fuel an Earnings Beat: 6 Key Takeaways. *Real Money* (July 2020). https://bit.ly/3WpXPWx, Last accessed January 7, 2023.

[14] Kozuch, M., and Satyanarayanan, M. Internet Suspend/Resume. In *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications* (2002), pp. 40–46.

[15] Lan, Y., and Xu, H. Research on technology of desktop virtualization based on spice protocol and its improvement solutions. *Frontiers of Computer Science 8*, 6 (2014), 885–892.

[16] Myerson, T. Welcoming Developers to Windows 10, April 2015. https://bit.ly/3QQB4Kc, Last Accessed January 7, 2023.

[17] Open Edge Computing Initiative. Interconnect Workstream Results, 2021. https://www.cmu.edu/scs/edgecomputing/resources/oec-archive/interconnect-workstream/index.html.

[18] Richardson, T., Stafford-Fraser, Q., Wood, K., and Hopper, A. Virtual network computing. *IEEE Internet Computing 2*, 1 (1998), 33–38.

[19] Satyanarayanan, M., Eiszler, T., Harkes, J., Turki, H., and Feng, Z. Edge Computing for Legacy Applications. *IEEE Pervasive Computing 19*, 4 (2020), 19–28.

[20] Satyanarayanan, M., Gao, W., and Lucia, B. The Computing Landscape of the 21st Century. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (HotMobile '19)* (Santa Cruz, CA, 2019).

[21] Satyanarayanan, M., Gilbert, B., Toups, M., Tolia, N., Surie, A., O'Hallaron, D. R., Wolbach, A., Harkes, J., Perrig, A., Farber, D. J., Kozuch, M. A., Helfrich, C. J., Nath, P., and Lagar-Cavilla, H. A. Pervasive Personal Computing in an Internet Suspend/Resume System. *IEEE Internet Computing 11*, 2 (March/April 2007).

[22] Smith, S., Darwhekar, I., Blakley, J., Eiszler, T., and Harkes, J. Segmenting Latency in a Private 4G LTE Network. Tech. Rep. Technical Report CMU-CS-22-115, School of Computer Science Carnegie Mellon University, 2022.

[23] Tolia, N., Andersen, D. G., and Satyanarayanan, M. Quantifying interactive user experience on thin clients. *IEEE Computer 39*, 3 (2006), 46–52.

[24] VMware. VMware Blast Extreme Protocol. `http://bit.ly/3ZNswak`.

[25] VMware. VMware Horizon. `https://www.vmware.com/products/horizon.html`.

[26] Ye, Y., Vora, N., Sairanen, J.-P., and Sahasrabudhe, M. Enabling local breakout from eNodeB in LTE networks. In *2012 IEEE International Conference on Communications (ICC)* (2012), IEEE, pp. 6982–6986.