Emulation & Virtualization as Preservation Strategies

David S. H. Rosenthal
LOCKSS Program, Stanford University Libraries

Figure 1: About 4 hours and the Mini vMac emulator was all it took Nick Lee to get OS7 running on his Apple Watch [86]

Executive Summary

Between the two fundamental digital preservation strategies, migration has been strongly favored. Recent developments in emulation frameworks make it possible to deliver emulations to readers via the Web in ways that make them appear as normal components of Web pages. This removes what was the major barrier to deployment of emulation as a preservation strategy. Barriers remain, the two most important are that the tools for creating preserved system images are inadequate, and that the legal basis for delivering emulations is unclear, and where it is clear it is highly restrictive. Both of these raise the cost of building and providing access to a substantial, well-curated collection of emulated digital artefacts beyond reach.

If these barriers can be addressed, emulation will play a much greater role in digital preservation in the coming years. It will provide access to artefacts that migration cannot, and even assist in migration where necessary by allowing the original software to perform it. The evolution of digital artefacts means that current artefacts are more difficult and expensive to collect and preserve than those from the past, and less suitable for migration. This trend is expected to continue.

Emulation is not a panacea. Technical, scale and intellectual property difficulties make many current digital artefacts infeasible to emulate. Where feasible, even with better tools and a viable legal framework, emulation is more expensive than migration-based strategies. The most important reason for the failure of current strategies to collect and preserve the majority of their target material is economic; the resources available are inadequate. The bulk of the resources expended on both migration and emulation strategies are for ingest, especially metadata generation and quality assurance. There is a risk that diverting resources to emulation, with its higher per-artefact ingest cost, will exacerbate the lack of resources.

Areas requiring further work if emulation is to achieve its potential as a preservation strategy include:

- Standardization of the format of preserved system images, the way they are obtained by emulators, and the means by which emulations of them are exposed to readers. This would enable interoperability between emulation components, aiding contributions and support from the open-source community.

- Improvements to the tools for associating technical metadata with preserved software to enable it to be emulated, and the technical metadata databases upon which they depend. This would reduce the cost of preserved system images.

- Clarification, and if possible relaxation, of the legal constraints on the creation and provision of access to collections of preserved system images. This would encourage institutions to collect software.

1 Introduction

After briefly describing the history of emulation and virtualization technologies, and defining some terminology, this report is divided into two main sections, separated by an interlude, and followed by a coda:

- The first section, *Looking Backward*, summarizes the state of the art in applying these technologies to the preservation of digital artefacts and their delivery in forms accessible to the unskilled. It uses three frameworks for delivery of emulations as exemplars. It concludes that these current techniques are technically effective and generally usable to replicate the experience of those individual digital artefacts and environments typical of the last century (*legacy* digital artefacts) for users of currently available systems. Their use is however hampered by intellectual property and economic issues, and by the lack of tools to ease the process of creating emulated environments.

- The interlude discusses the evolution of digital artefacts and their hardware infrastructure since before the turn of the century.

- The second section, *Looking Forward*, contrasts the effectiveness of current emulation and virtualization technology in providing future access to *legacy* versus *current* digital artefacts, and identifies a set of technological problems that require solutions.

- The coda discusses the sustainability of current emulation efforts, and presents a to-do list of actions to improve the effectiveness and reduce the cost of emulation as a strategy.

Unless specifically noted, all screen captures are of emulations accessed via the Chromium browser on a current Ubuntu Linux system over a 3Mb/s DSL connection.

1.1 History

Emulation & virtualization technologies have been a feature of the information technology landscape for a long time, but their importance for preservation was first brought to public attention in Jeff Rothenberg's seminal 1995 *Scientific American* article *Ensuring the Longevity of Digital Documents* [112]. As he wrote, Apple was using emulation in the transition of the Macintosh from the Motorola 68000 to the Power PC. The experience he drew on was the rapid evolution of digital storage media such as tapes and floppy disks, and of applications such as word processors each with their own incompatible format. His vision can be summed up as follows: documents are stored on off-line media which decay quickly, whose readers become obsolete quickly, as do the proprietary, closed formats in which they are stored. If this isn't enough, operating systems and hardware change quickly in ways that break the applications that render the documents.

Rothenberg identified two techniques by which digital documents could survive in this unstable environment, contrasting the inability of format migration to guarantee fidelity with emulation's ability to precisely mimic the behavior of obsolete hardware.

Rothenberg's advocacy notwithstanding, most digital preservation efforts since have used format migration as their preservation strategy. The isolated demonstrations of emulation's feasibility, such as the collaboration between the UK National Archives and Microsoft [22], had little effect. Emulation was regarded as impractical because it was thought (correctly at the time) to require more skill and knowledge to both create and invoke emulations than scholars wanting access to preserved materials would possess.

Emulation advocates including Rothenberg, Raymond Lorie of IBM, and the Dutch KB, instead of focusing on building emulations of the physical computers in use at the time, were sidetracked by the goal of building a "Universal Virtual Computer" (UVC) [134]. A UVC can be thought of as a meta-emulator, or a specification language for emulators.

The UVC was of interest only for preservation, and was thus unable to leverage the emulators that were concurrently being developed by hardware developers, game enthusiasts, and others. The Utopian goal of a UVC diverted resources that could have been applied to making the existing emulations easily usable and thus addressing the valid criticisms of their practicality.

1.2 Terminology

Terminology in this area is somewhat confusing. Preservation is not the main use of either emulation or virtualization, and terms have been re-used with somewhat different meanings to those useful for preservation. The following definitions are slightly idiosyncratic, but should make this report clearer.

- A **virtual machine** (VM) is a computer that has no separate physical existence, but is part of the behavior of a physical computer, called the VM's **host** computer. VMs mimic the instruction set and hardware configuration of some physical machine, or an abstract machine such as the Java Virtual Machine [70].

- **Virtualization** is a technique for implementing a VM on a host computer. It depends on the host computer's instruction set being the *same* as (strictly, mostly a superset of) the VM's instruction set, and having certain specific hardware properties that enable virtualization. Almost all instructions executed by the VM are directly executed by the host computer's CPU; only a few instructions are intercepted, using the specific properties, and performed by host software. Virtualized systems run *unmodified* software binaries.

- **Paravirtualization** is a close relative of virtualization, often used to substitute for obsolete I/O devices [11]. Software binaries, typically the operating systems, are modified by replacing the drivers for the obsolete I/O devices by newly constructed ones for virtual I/O devices that translate I/O to the host system's devices.
- A **virtual machine monitor** (VMM) is the name of the host software that enables virtualization by creating and monitoring the VM, and performing the intercepted operations. An alternate name is a **hypervisor**.
- **Emulation** is a technique for implementing a VM on a host computer whose instruction set is *different* from the host computer's. None of the instructions executed by the VM are directly executed by the host computer's CPU; all are translated by host computer software from the VM's instruction set to the host computer's instruction set by host software before being executed. Emulated systems run *unmodified* software binaries.
- The host computer software that does this translation, mimicking the VM's instruction set and its virtual hardware configuration, is called an **emulator**.
- A **preserved system image** is a set of stored data that is input to an emulator or a VMM and thereby executed[1]. The set of data would normally include metadata describing the VM's configuration, the contents of its storage media, and optionally the state of the virtual machine's memory.
- A preserved system image might be the result of **imaging** a physical computer's disks, as for example when accessioning a deceased faculty member's "papers". Or it might be created by **installing** from vendor's install media for the operating system and applications needed, together with preserved files requiring that environment.

As can be seen, the difference between emulation and virtualization is not absolute. In virtualization, some VM instructions are executed in host software. It is common to run an emulator to execute a VM whose instruction set is a subset of the host's. Emulating an early PC with an Intel 386 CPU on a modern Intel CPU is an example. Some emulators are implemented using just-in-time compilation, in which the first time a VM instruction sequence is encountered it is in effect replaced by its translation into the corresponding host machine sequence. This can greatly reduce the cost of emulation.

Thus, except where specifically indicated otherwise, in the following "emulation" will be used as shorthand for "emulation or virtualization depending on the overlap between the VM and host instruction sets".

Also, this report uses the term "digital artefact" rather than the usual "digital object" to emphasize that what is being emulated is an aggregation of many digital components into a form designed by humans to convey information. The OED defines artefact as:

> An object made or modified by human workmanship, as opposed to one formed by natural processes. [89]

2 Looking Backward

A digital preservation system that uses emulation will consist of three main components:

- One or more *emulators* capable of executing preserved system images.
- A *collection* of preserved system images, together with the metadata describing which emulator configured in which way is appropriate for executing them.
- A *framework* that connects the user with the collection and the emulators so that the preserved system image of the user's choice is executed with the appropriately configured emulator connected to the appropriate user interface.

Emulators have long been available for most systems of interest for preservation purposes, but they have been little used. The barriers to their use have been the complexity and cost of preparing a suitable preserved system image containing the digital artefact to be preserved, selecting and configuring a suitable emulator, representing the result of all this work as generally usable metadata, and delivering the resulting emulated digital artefact to a reader.

Recent progress in emulation frameworks has demonstrated that these barriers can be significantly reduced. In particular, in the Web environment, emulated digital artefacts can be delivered to readers transparently, so that they experience the preserved artefact without realizing that it is being emulated. Readers see no difference between the emulated artefact and other components of a Web page.

[1] As an example of confusion, in the context of virtualization, these sets of data are often called "virtual machines"

2.1 State of the Art: Emulators

All current frameworks depend on the same set of open source emulator projects, of which there are many. Those commonly used for preservation include:

- QEMU [95].
- MAME and its derivatives [73, 79].
- Basilisk II [20].
- DOS Box [40].

2.1.1 QEMU

QEMU is free and open-source software that describes itself thus [95]:

> QEMU is a generic and open source machine emulator and virtualizer.
>
> When used as a machine emulator, QEMU can run OSes and programs made for one machine (e.g. an ARM board) on a different machine (e.g. your own PC). By using dynamic translation, it achieves very good performance.
>
> When used as a virtualizer, QEMU achieves near native performances by executing the guest code directly on the host CPU. QEMU supports virtualization when executing under the Xen hypervisor or using the KVM kernel module in Linux. When using KVM, QEMU can virtualize x86, server and embedded PowerPC, and S390 guests.

QEMU emulates x86 and x86-64 systems, and these other systems [94]:

- PowerPC
- Sparc32
- Sparc64
- MIPS
- ARM
- ColdFire
- Cris
- Microblaze
- SH4
- Xtensa

QEMU is mainstream Linux software, part of most Linux distributions. QEMU is a member of the Software Freedom Conservancy [122], a not-for-profit organization that provides legal and administrative assistance to open source projects. QEMU interacts with the Conservancy through a Leadership Committee, with six members.

The set of graphics devices QEMU supports will become important later (Section 4.2.1). The available devices include:

- `cirrus`, an emulation of a Cirrus SVGA card. These were the market leaders in the early-to-mid 90s before being displaced by Graphics Processing Units (GPUs) as discussed in Section 3.2.1, and as such were well-supported at the time. For example, Windows NT 4.0 shipped with a driver for this hardware.
- `stdvga`, an emulation of a generic VGA card with paravirtualized extensions for higher resolutions. These extensions are well supported by current BIOS and driver software, so almost all guest systems work with `stdvga`. However, it has no graphics acceleration; all rendering is performed in host software. Modern desktop software expects a GPU that accelerates rendering. Its emulated performance is acceptable but noticeably less smooth than hardware. Emulations of games that expect a GPU using `stdvga` are not usable.
- `virtio-gpu`, a paravirtualization of the 3D capabilities of a GPU, which is discussed in Section 4.2.1.

2.1.2 MAME, MESS and derivatives

MAME describes itself thus [73]:

> MAME stands for Multiple Arcade Machine Emulator. When used in conjunction with images of the original arcade game's ROM and disk data, MAME attempts to reproduce that game as faithfully as possible on a more modern general-purpose computer. MAME can currently emulate several thousand different classic arcade video games from the late 1970s through the modern era.
>
> The source code to MAME is available for development and learning purposes. Most of it is free and open source.
>
> The main purpose of MAME is to be a reference to the inner workings of the emulated arcade machines. This is done both for educational purposes and for preservation purposes, in order to prevent many historical games from

> disappearing forever once the hardware they run on stops working. Of course, in order to preserve the games and demonstrate that the emulated behavior matches the original, you must also be able to actually play the games. This is considered a nice side effect, and is not MAME's primary focus.

Mess, the Multi Emulator Super System, was originally a separate project [79]:

> Prior to version 0.162 (May 2015), MAME only supported arcade machines, and MESS was a separate emulator for all other types of systems, built on the same code base, although there has been increasingly close cooperation between the teams for many years.
>
> AS OF VERSION 0.162, MAME AND MESS HAVE BEEN COMBINED INTO A SINGLE EMULATOR! ... As of version 0.163, MESS supports 1,081 unique systems with 2,199 total system variations and is growing all the time ... However, not all of the systems in MESS are fully functional.

JSMESS is a way of running the MESS (strictly now MAME) emulator inside a Web browser. The project describes itself thus [114]:

> JSMESS is a pipeline, using the Emscripten environment to compile the MAME/MESS emulator into JavaScript. The goal is to be able to get the latest improvements and bug fixes for all outside components into our distribution as quickly as possible. It also reduces our workload; we leave the emulation to the emulation authors, the conversion to the conversion authors, and simply work on using the best aspects of these environments to make the fastest and most flexible browser-based emulator we can.

This entire toolchain will shortly be consistently open source under a GPL license, thanks to a major effort to contact the entire history of contributors to MAME and MESS for their approval.

2.1.3 Basilisk II

Basilisk II describes itself thus [20]:

> Basilisk II is an Open Source 68k Macintosh emulator. That is, it allows you to run 68k MacOS software on your computer, even if you are using a different operating system. However, you still need a copy of MacOS and a Macintosh ROM image to use Basilisk II.
>
> Emulates either a Mac Classic (which runs MacOS 0.x thru 7.5) or a Mac II series machine (which runs MacOS 7.x, 8.0 and 8.1), depending on the ROM being used.

2.1.4 DOSBox

Wikipedia states [8]:

> DOSBox is an emulator program that emulates an IBM PC compatible computer running a DOS operating system. Many IBM PC compatible graphics and sound cards are also emulated. This means that original DOS programs (including PC games) are provided an environment in which they can run correctly, even though the modern computers have dropped support for that old environment. DOSBox is free software written primarily in C++ and distributed under the GNU General Public License. DOSBox has been downloaded over 25 million times since its release on SourceForge in 2002.
>
> ...
>
> The original DOSBox has not been updated in a long time. Active development is happening on forks of DOSBox. Forks such as SVN Daum and DOSBox-X provide additional features, which include support for save states and long filenames.

DOSBox [40] is used extensively by the commercial retro-gaming business.

2.1.5 Other Emulators

- *SheepShaver* Christian Bauer, who supports Basilisk II for 68K Mac emulation, also supports SheepShaver [21], a related emulator which is one of the few options for emulating PowerPC Macs. It is said to be in need of additional support.
- *UAE*, or the Unusable Amiga Emulator [13] is no longer unusable, it forms the basis of Basilisk IIs Motorola 68K emulation. It is also the basis for a business selling legal ROMs, operating systems, Amiga software, and even support at amigaforever.com.

2.2 State of the Art: Frameworks

The three frameworks described here are:

- **bwFLA**, developed at the University of Freiburg [69].
- **Olive**, developed at Carnegie Mellon University [113].
- the framework underlying the **Internet Archive**'s software library [57].

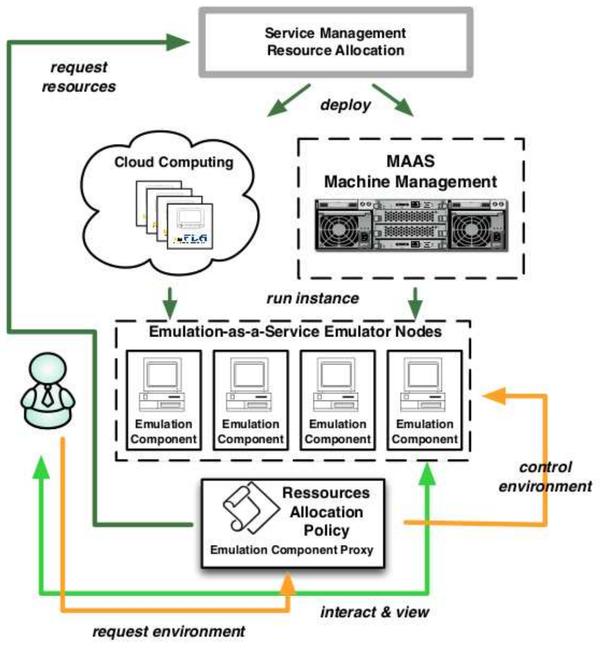


Figure 2: bwFLA overall architecture [69]

2.2.1 bwFLA

bwFLA was developed and is supported by a team at the University of Freiburg to provide Emulation As A Service (EAAS). Their framework runs in "the cloud" to provide comprehensive management and access facilities wrapped around a selection of emulators such as those in Section 2.1. Figure 2 shows the overall architecture of the system.

As a cloud service, bwFLA has to manage creating and destroying virtual machine instances within the cloud, and monitoring their behavior. Thus, the user interacts initially with a service manager (Emulation Component Proxy). This looks to the user like the emulation being requested, but in fact performs tasks preparatory to the emulation such as creating a session for the user, queuing the session while resources are requested, assigned and configured, and finally re-directing the user to the newly-created emulation.

The sequence of events that takes place when a user clicks on a link to an emulation, for example to the *Chop Suey* CD-ROM [43] (Figure 9), is as follows:

- The browser connects to the EC-proxy, which notices that this is a new session.
- Normally the EC-proxy would authenticate the user, but because this CD-ROM emulation is open access it doesn't need to.
- Assigns a VM to run the session's EC. If no VM is available when the request comes in it can take up to 90 seconds to start another.
- The EC-proxy starts the EC on the assigned VM with metadata telling it what to run.
- The emulator starts. After a short delay the user sees the Mac boot sequence, and then the CD-ROM starts running.
- At intervals, the EC sends the EC-Proxy a keep-alive signal. ECs that haven't sent one in 30 seconds are presumed dead, and their resources are reclaimed to avoid paying the cloud provider for unused resources.

In large-scale deployments such as Rhizome's, that may be faced with huge spikes in demand [97], things are a bit more complex. Rhizome implemented a "front-end" in node.js that sits between the user's browser and the EC-Proxy. It does three things:

- Implements a queue of incoming requests for emulation sessions, batching requests to the EC-Proxy for two seconds.
- When the EC-Proxy replies with some sessions, handing them out to the requests at the head of the queue.
- Showing the requests in the queue a "waiting" screen.
- Checking that the emulation is actually visible to the user and being used. After three minutes of invisibility or inactivity, the session is terminated, again to avoid paying the cloud provider for unused resources.

In order to use a variety of emulators, bwFLA encapsulates each of them as shown in Figure 3. The three sets of interfaces are:

- Data I/O, connecting the emulator to data sources such as disk images, user files, an emulated network containing other emulators, and the Internet.
- Interactive Access, connecting the emulator to the user.
- Control, providing a Web Services interface that bwFLA's resource management can use to control the emulator.

The communication between the emulator and the user takes place via standard HTTP on port 80, in two parts:

- Command and control uses a Web Service REST API.

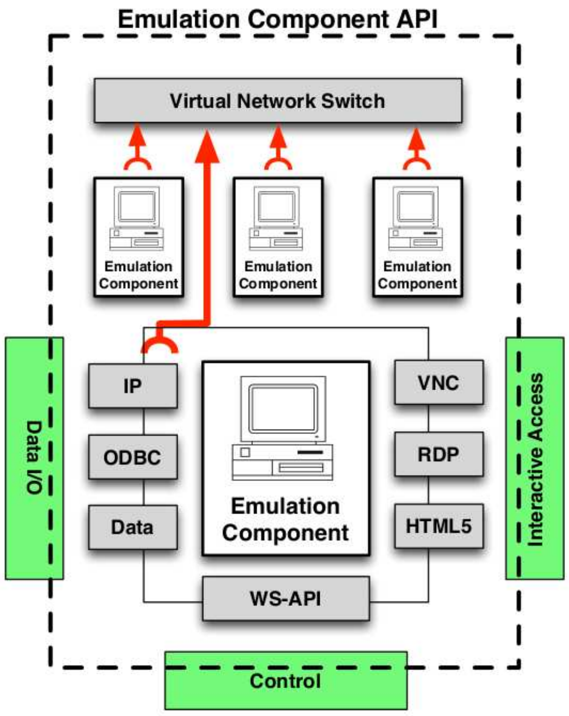


Figure 3: bwFLA emulator encapsulation [69]

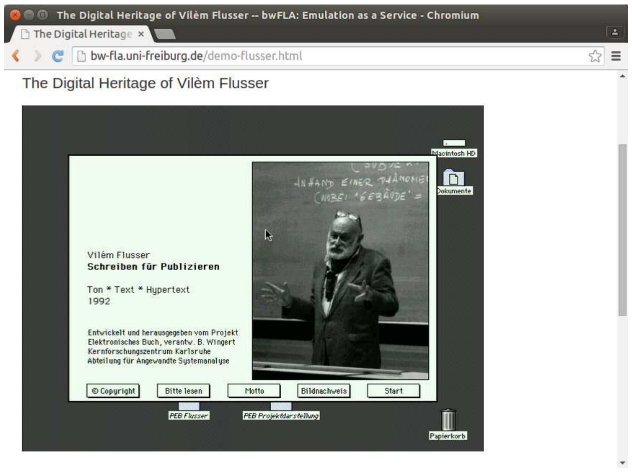
- Graphics and audio are encoded by the SDL video/audio driver into HTML5 standard formats.

Thus there is no need for a user to install software, or browser plugins, and no need to use ports other than 80. Both of these are important for systems targeted at use by the general public.

bwFLA's preserved system images are stored as a stack of overlays in QEMU's "qcow2" format [78]. Each overlay on top of the base system image represents a set of writes to the underlying image. For example, the base system image might be the result of an initial install of Windows 95, and the next overlay up might be the result of installing Word Perfect into the base system. Each overlay contains only those disk blocks that differ from the stack of overlays below it. The stack of overlays is exposed to the emulator as if it were a normal file system via FUSE.

These preserved system images currently represent only the system's disk, not its memory, so execution has to start by booting the system. Users have requested the ability to store memory images. This would allow emulations to behave as if resumed from a suspended state, for example after the emulated system had booted and the preserved artefact been started. Some emulators do not support this, and even the ones that do have technical issues (see Section 2.2.2). The team regards CRIU [36] as a potential path to this functionality but lacks resources

Figure 4: Flusser exhibit [30].



The Digital Heritage of Vilèm Flusser -- bwFLA: Emulation as a Service - Chromium

bw-fla.uni-freiburg.de/demo-flusser.html

The Digital Heritage of Vilèm Flusser

Vilém Flusser
Schreiben für Publizieren

Ton * Text * Hypertext
1992

Entwickelt und herausgegeben vom Projekt
Elektronisches Buch, verantw. B. Wingert
Kernforschungszentrum Karlsruhe
Abteilung für Angewandte Systemanalyse

Figure 5: Flusser exhibit using emulated CRT [30].

to work on it.

The technical metadata that encapsulates the system disk image is described in a paper to be presented to the iPres conference in November 2015 [98], using the example of emulating CD-ROMs. Broadly, it falls into two parts, describing the software and hardware environments needed by the CD-ROM in XML. The XML refers to the software image components via the Handle system [33], providing a location-independent link to access them. The paper describes an initial implementation of a tool for compiling and packaging this metadata which worked quite well for the restricted domain of CD-ROMs.

Although the bwFLA framework is designed to be deployed in the cloud, because the components communicate using network protocols they can be, and have been, all deployed on a single system. One application of this is in exhibition settings such as the Flusser exhibit at the ZKM, Karlsruhe, where this technique used a bootable USB flash disk to emulate a Mac Performa 630 running

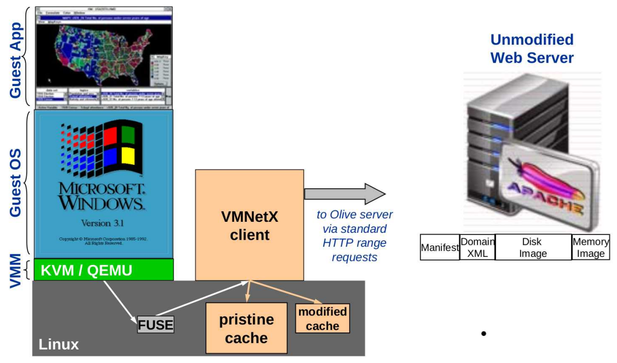


Figure 6: Olive Architecture [113]

a HyperCard stack [30]. The exhibit used a real CRT (Figure 4), the emulation on the Web uses graphical techniques to simulate the appearance of a CRT on today's LCD displays (Figure 5).

Freiburg plan to release their source code under an open source license but the exact license has not yet been decided upon.

2.2.2 Olive

The canonical configuration of the Olive framework is shown in Figure 6. The user's computer runs Linux with some standard components (KVM, QEMU and FUSE) installed. Also installed is VMNetX, the Olive client software (see Figure 6).

When a user clicks on a link to an emulation VMNetX obtains metadata describing the desired emulation from Olive's Web server and configures it, providing the emulator via FUSE with the illusion of file systems fully populated with the data of the desired emulated system image and file systems. In detail the following sequence of events takes place:

- The browser is redirected to a URL with a "vmnetx+https" URL scheme.
- The installation process for VMNetX has registered it as an external URL handler for that scheme, so the browser launches it and passes it the URL.
- VMNetX strips off the "vmnetx+" part of the URL scheme; its only function is to support the handoff above. What's left is an HTTPS URL to a VMNetX package file (.nxpk) on the server.
- Using HTTP range requests, VMNetX reads the ZIP file header in the package file and determines the offset + length of each ZIP file member within the NXPK. It then fetches the byte ranges corresponding to the XML manifest and domain XML.
- VMNetX starts a helper process, vmnetfs, which mounts a FUSE virtual filesystem containing "files" mirroring the contents of the disk and memory images for the emulation.
- Because QEMU will sequentially read the entire memory image, if it isn't already present in the cache (see below) vmnetfs starts a background thread to pre-load the cache by streaming its contents from the server.
- VMNetX updates the domain XML with local configuration (such as the path to the FUSE filesystem) and passes it to libvirt, which launches QEMU on the disk and memory image virtual files.
- QEMU starts loading the memory image.
- VMNetX shows its UI. Using an I/O trace exported by vmnetfs, it watches QEMU read through the memory image, and updates the "Loading" progress bar accordingly.
- When the memory image has been loaded, VMNetX starts a viewer widget for the SPICE thin client protocol and connects it to the listening socket exposed by QEMU.

The NXPK is a ZIP file containing:

(a) an XML manifest file,

(b) the "domain XML" containing the configuration parameters for the VM,

(c) the disk image in compressed qcow2 format [78],

(d) optionally, the memory image as a libvirt-wrapped compressed QEMU save file.

Crucially, all ZIP file members are uncompressed ("stored" rather than "deflated"). Thus the compression of (c) and (d) uses mechanisms provided by the file formats of those individual members rather than by ZIP.

In fact, the file systems VMNetX provides to the emulator are not fully populated. As the emulator accesses data blocks from the FUSE file system, the blocks are demand-paged from the Olive Web server using standard HTTP range queries and cached locally.

VMNetX maintains two layers of caching above the Olive Web server. The lower cache contains unmodified copies of the corresponding data in the Olive Web server (the *pristine* cache). As the emulated execution proceeds, it will write to and thus modify the contents of the emulated system's memory and file system so that the lower cache no longer represents the state of the emulated system (c.f. [111]). The upper (*modified*) cache captures these writes and thus reflects the state of the emulated system where it differs from that on the Olive Web server. When the emulation needs data it is fetched from the upper cache if it is present there. If not it is fetched from the lower cache if it is present there, and if not it is fetched via HTTP range query from somewhere in the NXPK ZIP file on Olive's Web server.

Because many users' systems are expected to be at the edge of the Internet, with significant latency and restricted bandwidth, even though data once fetched is cached the user would be faced with delays as data was fetched for the first time:

> Last-mile networks such as 4G cellular networks pose special challenges for Olive. Their low bandwidth and high latency make demand paging of Olive VMs over the Internet unacceptably slow. We have conducted experiments with history-based prefetching of VM state over last-mile networks in an experimental version of Olive called vTube. To generate accurate prefetching hints, vTube uses fine-grained analysis of disk and memory state access traces from previous executions. Our preliminary results show that despite wide variances from execution to execution and from user to user, vTube can identify short segments of state access that once activated, are exceptionally stable across multiple executions and can thus provide high-quality predictive hints. [113]

Because each time the emulation is invoked it executes the same program in the same environment, differing only in the user's inputs, many execution segments are the same between invocations. Recognizing them, and pre-loading the cache with the data they typically access, can make it much more likely that needed data will be found in the cache, and the emulation thus be able to proceed without delay.

> Qualitatively, the user experience in vTube during these experiments is comparable to viewing video over a last-mile network. [113]

Except in controlled "reading room" situations the proportion of users accessing Olive's emulations via systems running Linux powerful enough to run the emulators, and willing to install non-mainstream software such as VMNetX, will be small. Thus Olive also supports a configuration more akin to bwFLA's in which a Linux system close to the user in what they call a "cloudlet" runs the emulation, and communicates with another instance of VMNetX installed on the user's device (versions are available for Linux, Windows and Android). The communication between the two instances of VMNetX uses a custom protocol to transmit command and control messages, and to encapsulate graphics and audio via the SPICE remote desktop protocol [123]. However, it is still not possible for users to access Olive emulations without installing some software on their device, a barrier to wide adoption.

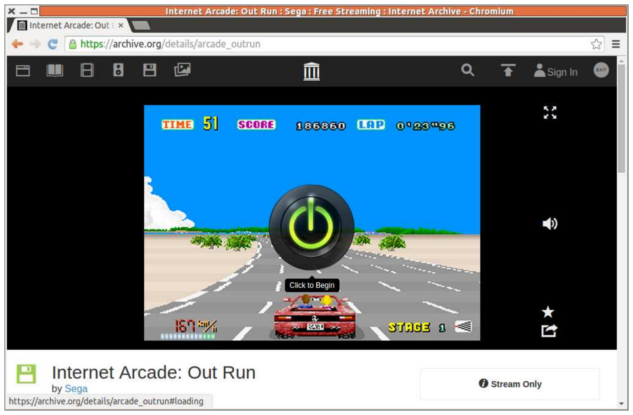

Internet Arcade: Out Run : Sega : Free Streaming : Internet Archive - Chromium
Internet Arcade: Out
https://archive.org/details/arcade_outrun
Sign In
TIME 51
SCORE 186860
LAP 0'23"96
Click to Begin
STAGE 1
Internet Arcade: Out Run
by Sega
Stream Only
https://archive.org/details/arcade_outrun#loading

Figure 7: Before emulator loading

Olive's code is released under the GPLv2 license.

2.2.3 Internet Archive

The Internet Archive's framework is different from bwFLA, because its emulators run not on some other computer accessed via the network, but on the user's own computer. It is different from Olive's canonical configuration in that the emulator runs not directly on the user's Linux system, but inside the user's browser. The browser's JavaScript environment is effectively independent of the user's underlying operating system, thus unlike Olive, and like bwFLA, the Internet Archive's framework does not require any special software on the user's machine. Nor, unlike bwFLA or Olive's "cloudlet" configuration, does it require the archive to pay for infrastructure to run the emulations, merely to store them and serve them to user's browsers.

When the user clicks on the "Click to begin" power button (Figure 7), JavaScript in that area of the window is executed to load metadata describing the emulation, the emulator itself, and the game to be emulated (Figure 8). The emulators are implemented in JavaScript, so they can be executed by the browser's JavaScript engine. They are typically created by compiling the emulator implemented in some other language into JavaScript using Emscripten [137, 115].

It might be thought that the performance of running the emulator locally by adding another layer of virtualization (the JavaScript virtual machine) would be inadequate, but this is not the case for two reasons. First, the user's computer is vastly more powerful than the computer being emulated, and second, the performance of the JavaScript engine in a browser is critical to its success, so large resources are expended on optimizing it.

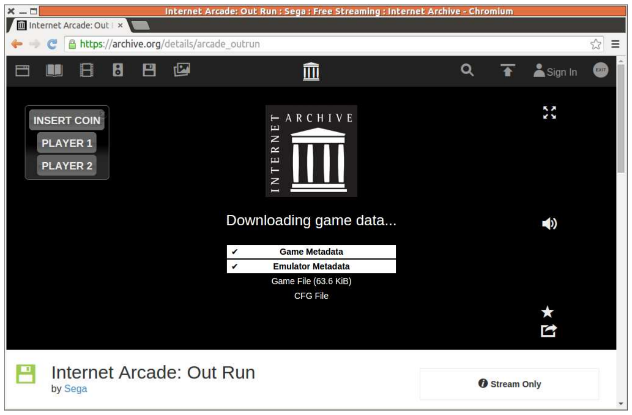


Figure 8: During emulator loading

2.2.4 Other Frameworks

Two successive EU-funded projects produced emulation frameworks:

- The KEEP project ran from 2009 to 2012 and produced an Emulation Framework [61] which featured:
  - 6 platforms supported: x86, C64, Amiga, BBC Micro, Amstrad, Thomson TO7
  - 7 emulators included: Dioscuri, Qemu, VICE, UAE, BeebEm, JavaCPC, Thomson
  - 30+ file formats supported: PDF, TXT, XML, JPG, TIFF, PNG, BMP, Quark, ARJ, EXE, disk/tape images and more
  - Integration with format identification tool FITS
  - Web services for software and emulator archives
- The UK Web Archive's Interject [131] framework was developed under the SCAPE project, which ended in 2014, as a research prototype. The goal was to demonstrate how emulation and migration could cooperate to provide access to a large Web archive. It was, for example, able to automatically emulate software for the ZX Spectrum found in the UK Web Archive.

2.3 State of the Art: Collections

This section examines the collections of preserved system images available at some institutions. The key to emulation's usefulness as a preservation strategy will be large collections of preserved system images; so far we have one large collection at the Internet Archive, a small one at Rhizome, and demo collections at Carnegie Mellon and Freiburg.

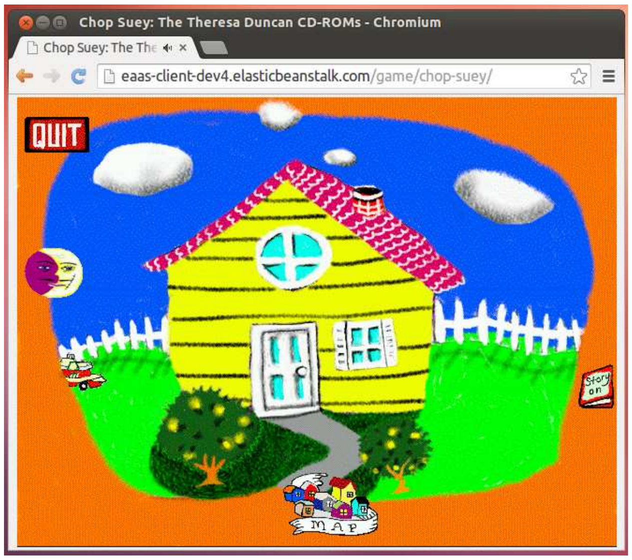

Chop Suey: The Theresa Duncan CD-ROMs - Chromium
eaas-client-dev4.elasticbeanstalk.com/game/chop-suey/
QUIT
Story on
MAP

Figure 9: Theresa Duncan and Monica Gesue's *Chop Suey* on MacOS 7.5 (1995) [43]

2.3.1 Rhizome

Rhizome's collection of digital artworks includes a number of preserved system images, including:

- The Theresa Duncan CD-ROMS [84], three seminal feminist games on Mac OS 7.5; *Chop Suey* from 1995 (Figure 9), *Smarty* from 1996 (Figure 10), and *Zero Zero* from 1997 (Figure 11).
- *Bomb Iraq* [44] is a system image of a Macintosh TV bought at a Salvation Army store in 2005, previously used by a student, who created a HyperCard stack game called *Bomb Iraq*. The disk image has been redacted to protect the student's identity, but is otherwise as it was when it was discarded.
- *untitled[scrollbars]* by Jan Robert Leegte [66], an exploration of browser scrollbars from 2000. The artist's intended experience requires a contemporary Internet Explorer [67]. Increasingly, scrollbars are not permanent but pop up when needed. Viewing the piece with, for example, Safari on OS X is baffling because the scrollbars are not visible.

The launch of the Theresa Duncan collection garnered significant media attention (e.g. [100]) and thus a big spike in traffic (Figure 12). Dragan Espenschied writes:

> From launch of the project April 17 to June 23, 4644 emulation sessions were served, from that 976 sessions during release day. During the launch phase, users mostly tried the games out very briefly. For the plateau phase, the usage pattern changed to less users that were

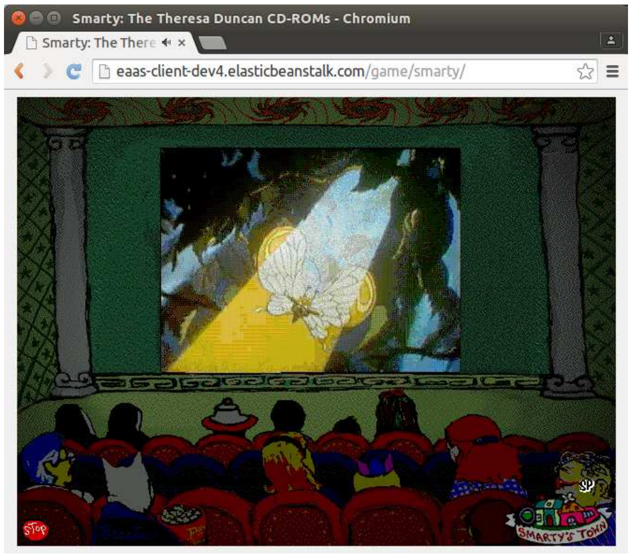

Smarty: The Theresa Duncan CD-ROMs - Chromium
Smarty: The There
eaas-client-dev4.elasticbeanstalk.com/game/smarty/
STOP
SMARTY'S TOWN

Figure 10: Theresa Duncan's *Smarty* on MacOS 7.5 (1996) [41]

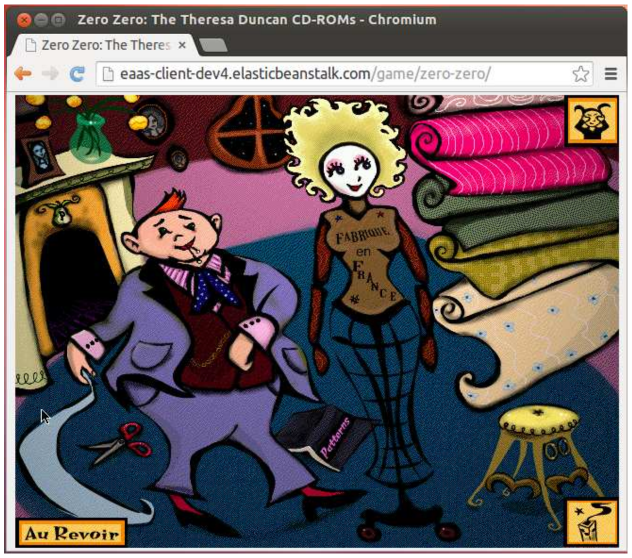

Zero Zero: The Theresa Duncan CD-ROMs - Chromium
Zero Zero: The Theres
eaas-client-dev4.elasticbeanstalk.com/game/zero-zero/
FABRIQUE en FRANCE
Patterns
Au Revoir

Figure 11: Theresa Duncan's *Zero Zero* on MacOS 7.5 (1997) [42]

> more "devoted" and played the games for up to two hours. The median session time was 99 seconds, with a wide variance between users. Top-20 users' session time was at least 109 minutes.

Rhizome's current plans include setting up their own emulation infrastructure rather than sharing bwFLA's. To this end they are evaluating Google's services as an alternative to Amazon's Elastic Beanstalk, and finding some advantages in speed of response to peaks.

2.3.2 Internet Archive

The Internet Archive's software collection currently holds nearly 36,000 items, including more than 7,300 for MS-DOS, 3,600 for Apple, 2,900 console games and 600 arcade games. Some can be downloaded, but most can only be streamed.

The oldest is an emulation of a PDP-1 with a DEC 30 display running the Space War game from 1962 [49], more than half a century ago. As I can testify having played this and similar games on Cambridge University's PDP-7 with a DEC 340 display seven years later, this emulation works well (Figure 13) [2].

The quality of the others is mixed. Resources for QA and fixing problems are limited; with a collection this size problems are to be expected. Jason Scott crowdsources most of the QA; his method is to see if the software boots up and if so, put it up and wait to see whether visitors who remember it post comments identifying problems, or whether the copyright owner objects.

[2]The Computer History Museum has restored PDP-1 hardware [35] that runs Space War [12].

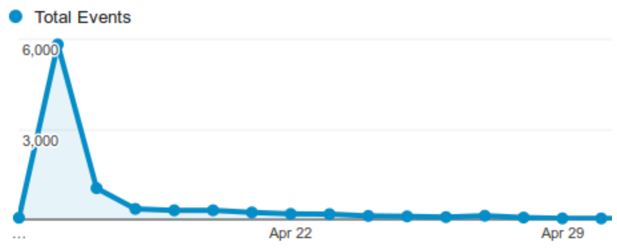


Figure 12: Daily count of emulations invoked after launch of Theresa Duncan CD-ROMs.

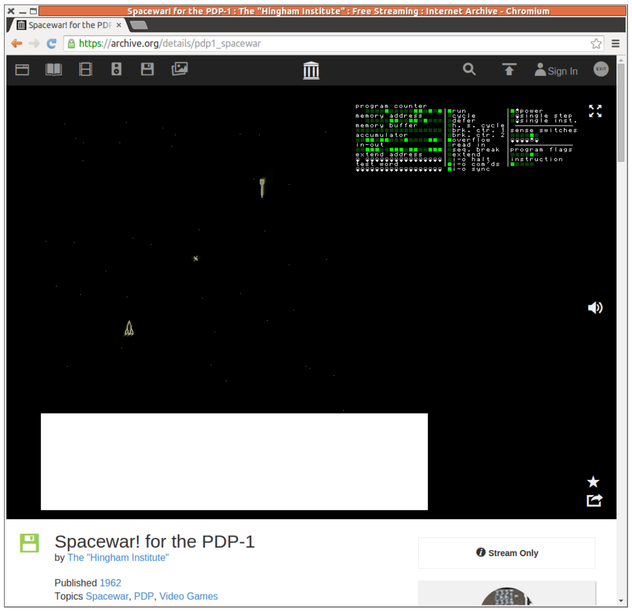
For example, VisiCalc for Apple][from 1979 [26] is perfectly usable once you have found the reference card for it [25] (Figure 14).

Out Run [116] is a Sega arcade game from 1986 that is perfectly playable, thanks to a helpful comment from user danpritts setting out the key assignments (Figure 15).

Wiz n' Liz [93] is a game for the Sega Mega Drive console from 1993 that is playable once you have figured out the key assignments, there is no helpful comment for this one (Figure 16).

The audio quality of many of these emulations is disappointing. Jason Scott writes:

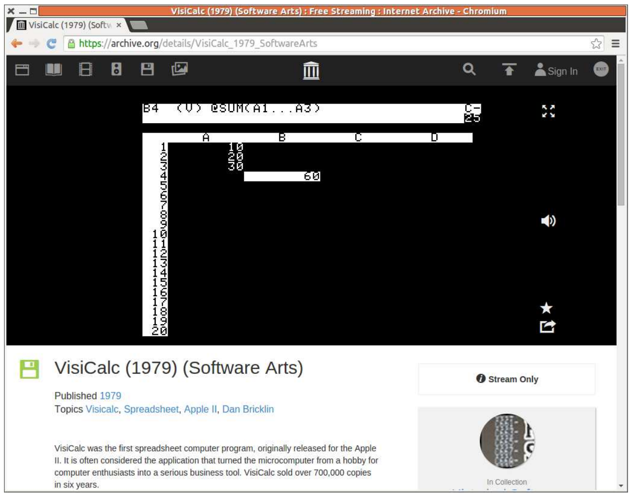
> Sound has turned out to be one of the more difficult issues with JavaScript programs. Different browsers handle sound differently, and the human ear notices tiny differences in sound quality much more than they noticed microscopic slowdowns in video rendering. This is definitely on the list to be improved as soon as


Spacewar! for the PDP-1 : The "Hingham Institute" : Free Streaming : Internet Archive - Chromium
https://archive.org/details/pdp1_spacewar
Sign In
Stream Only

Spacewar! for the PDP-1
by The "Hingham Institute"

Published 1962
Topics Spacewar, PDP, Video Games

Figure 13: Space War on the PDP-1 (1962)


VisiCalc (1979) (Software Arts) : Free Streaming : Internet Archive - Chromium
https://archive.org/details/VisiCalc_1979_SoftwareArts
Sign In
Stream Only
In Collection

VisiCalc (1979) (Software Arts)

Published 1979
Topics Visicalc, Spreadsheet, Apple II, Dan Bricklin

VisiCalc was the first spreadsheet computer program, originally released for the Apple II. It is often considered the application that turned the microcomputer from a hobby for computer enthusiasts into a serious business tool. VisiCalc sold over 700,000 copies in six years.

Figure 14: VisiCalc on the Apple][(1979)

possible, as soon as we have the best solution in place.

2.3.3 Carnegie Mellon

The Olive Archive's collection [90] currently contains 17 preserved system images including:

- *Mystery House*, a graphical game for the Apple][from 1982.
- *Oregon Trail 1.1*, a 1990 educational game for the Macintosh.
- *Great American History Machine*, visualization software from 1991 for American census and election data on Windows 3.1.
- *NCSA Mosaic 1.0*, an early Web browser for the Macintosh from 1993 (Figure 17).
- *TurboTax97*, the 1997 version of the popular tax preparation software on Windows 3.1 (Figure 18).
- *Chaste 3.1*, a 2013 simulation package for computationally demanding problems in biology and physiology, for Ubuntu 12.04 (Figure 19).

The collection is available on the CMU campus. It illustrates the range of problems that Olive can address, from long-obsolete games to software deposited with academic papers that has complex dependencies on underlying libraries.

2.3.4 Yale

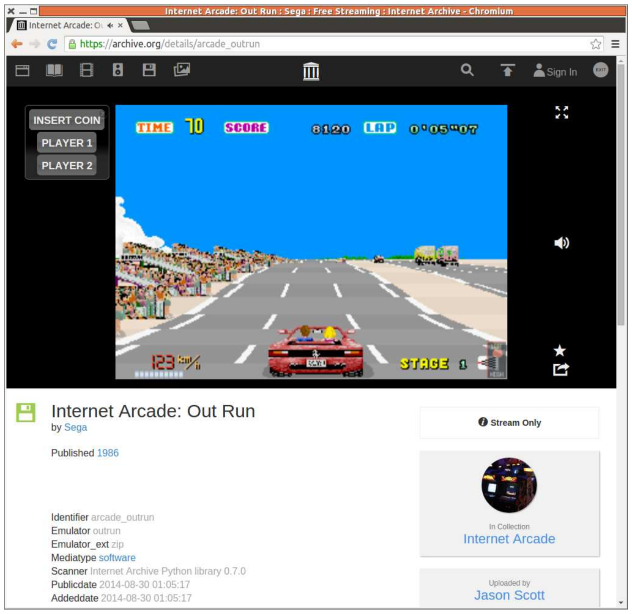
Yale's emulation infrastructure is similar to Rhizome's, using bwFLA. A pilot program was described in [34] but it is not yet in production use. When it starts the collection will be available to anyone with a Yale login, and to the public on terminals in a Yale library reading room. It is expected to start with about 440 CD-ROMs and a few system images. Long-term goals include providing all files ingested to the digital preservation system with an emulated contemporary environment, and exploring the use of emulation as a way of restricting access to digital artefacts.

2.3.5 Deutsche Nationalbibliothek

The bwFLA team are in the early stages of a two-year grant from DFG to work with the German National Library (DNB), the Bavarian State Library and the University of Arts in Karlsruhe to implement pilot reading-room access to multimedia content, such as the DNB's collection of about 500,000 physical carriers of digital artefacts, such as CD-ROMs. This uses the joint work between bwFLA and DNB in extracting technical metadata from CD-ROMs for emulation [98].

2.3.6 British Library

The British Library are in the early stages of an evaluation of preservation strategies for digital artefacts from their collection, comparing migration and two emulation frameworks, bwFLA (Section 2.2.1) and Interject (Section 2.2.4) as preservation strategies for a carefully constructed sample of 50 items from the BL's collection ranging from the late 1970s to the mid-2000s including both CD-ROMs and floppy disks. The goal is to develop preservation and access facilities for the various types of digital assets that the BL has accessioned incidentally,

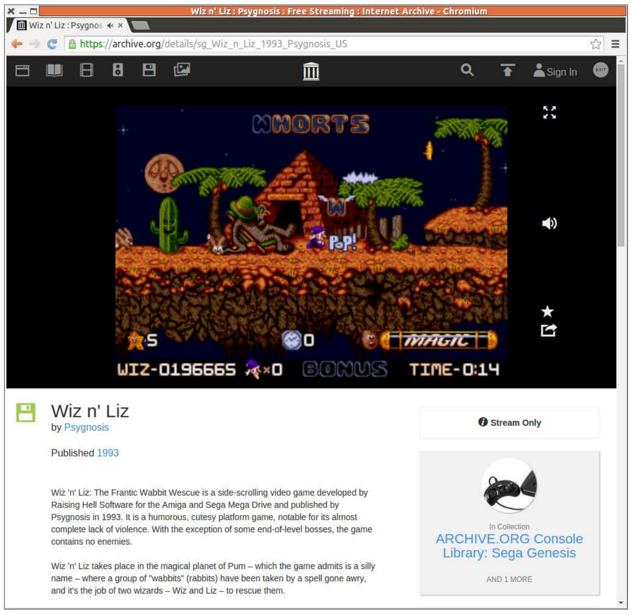

Internet Arcade: Out Run : Sega : Free Streaming : Internet Archive - Chromium
https://archive.org/details/arcade_outrun
Sign In
INSERT COIN
PLAYER 1
PLAYER 2
TIME 70 SCORE 8120 LAP 0'05"07
STAGE 1
Stream Only
In Collection
Internet Arcade
Uploaded by
Jason Scott

Internet Arcade: Out Run
by Sega

Published 1986

Identifier arcade_outrun
Emulator outrun
Emulator_ext zip
Mediatype software
Scanner Internet Archive Python library 0.7.0
Publicdate 2014-08-30 01:05:17
Addeddate 2014-08-30 01:05:17

Figure 15: Out Run on Sega's arcade system (1986)


Wiz n' Liz : Psygnosis : Free Streaming : Internet Archive - Chromium
https://archive.org/details/sg_Wiz_n_Liz_1993_Psygnosis_US
Sign In
WIZ-0196665 ×0 BONUS TIME-0:14
MAGIC
Stream Only
In Collection
ARCHIVE.ORG Console Library: Sega Genesis
AND 1 MORE

Wiz n' Liz
by Psygnosis

Published 1993

Wiz 'n' Liz: The Frantic Wabbit Wescue is a side-scrolling video game developed by Raising Hell Software for the Amiga and Sega Mega Drive and published by Psygnosis in 1993. It is a humorous, cutesy platform game, notable for its almost complete lack of violence. With the exception of some end-of-level bosses, the game contains no enemies.

Wiz 'n' Liz takes place in the magical planet of Pum – which the game admits is a silly name – where a group of "wabbits" (rabbits) have been taken by a spell gone awry, and it's the job of two wizards – Wiz and Liz – to rescue them.

Figure 16: Wiz n' Liz on Sega Genesis (1993)

for example via donation, because they were included with print materials, etc.

2.4 Shared Concerns

2.4.1 Emulator Support

Concern about the level of support for the emulators needed for preservation was universal. Each site had stories of minor but irritating inadequacies in the emulators they used. Developing and fixing bugs in emulators requires a high level of programming skill and motivation.

Preservation is not a significant use of QEMU in comparison with software and hardware development, and other commercial uses. Thus, although it is a mainstream open source project under active development, it is difficult to get issues of concern for preservation addressed by the QEMU team. For example, both the Rhizome and Olive teams report difficulties running early versions of Windows (from 95 to 98) under current QEMU versions. It appears that code needed for this worked some time ago but has not been fully maintained because these early Windows versions are not significant for the QEMU team.

Olive supports resuming rather than rebooting an emulation, but QEMU's support for this is inadequate. QEMU's developers are not committed to memory image interoperability across versions; version mismatches cause QEMU to exit. In that case, VMNetX will restart the VM without the memory image (cold-booting the guest OS) and will show a warning icon in the status bar to tell you this has happened. The thin-client server has the same version of QEMU as Olive's curation machines, so for emulations launched in thin-client mode the memory image will load properly. Efforts will be needed to improve this, either by maintaining a version of QEMU with consistent versioning, or by trying to get upstream QEMU to improve their cross-version compatibility.

Even mainstream users of QEMU have concerns. Five significant vulnerabilities have been discovered so far this year [117, 77, 119, 118]. which has to be focusing the team on security problems rather than enhancements.

Maintaining emulators for long-obsolete systems poses special problems. Their targets are static, so the rate at which bugs in their emulations are found will drop through time. Typically, the libraries upon which they depend are not under active development, so also will the rate at which bugs in their support from the operating system are found. Developers will see the bug rate drop, and will move on to other, more active projects.

We see this in many of the emulators for early PCs, Macs, and game systems. These were typically developed and supported by game enthusiasts; they don't have current commercial uses. In many cases their license specifically forbids commercial use in order to ease the IP issues of running old games [79].

Dragan Espenschied writes:

> Emulation development is either driven by hobbyists or the needs of big business. For example, hobbyists are making sure they can play their favorite arcade games in emulation; this means that software that is of interest to tech-savvy hobbyists gets prioritized in preservation, mostly very popular video games. Busi-

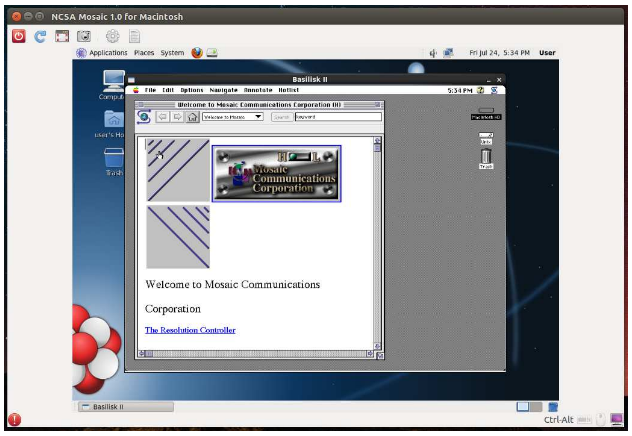
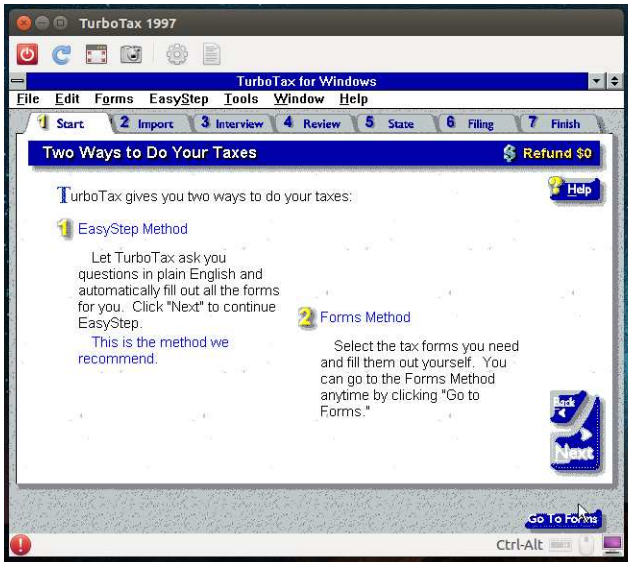

NCSA Mosaic 1.0 for Macintosh
Applications Places System
Fri Jul 24, 5:34 PM User
Basilisk II
File Edit Options Navigate Annotate Hotlist
Welcome to Mosaic Communications Corporation
Mosaic Communications Corporation
Welcome to Mosaic Communications Corporation
The Resolution Controller
Trash
Ctrl-Alt

Figure 17: NCSA's Mosaic 1.0 browser on MacOS (1993)


TurboTax 1997
TurboTax for Windows
File Edit Forms EasyStep Tools Window Help
1 Start 2 Import 3 Interview 4 Review 5 State 6 Filing 7 Finish
Two Ways to Do Your Taxes
Refund $0
Help
Back
Next
Go To Forms
Ctrl-Alt

TurboTax gives you two ways to do your taxes:

1 EasyStep Method

Let TurboTax ask you questions in plain English and automatically fill out all the forms for you. Click "Next" to continue EasyStep.

This is the method we recommend.

2 Forms Method

Select the tax forms you need and fill them out yourself. You can go to the Forms Method anytime by clicking "Go to Forms."

Figure 18: TurboTax on Windows 3.1 (1997)

> ness driven emulation projects like qemu are focusing on interoperability with host systems for certain business-relevant guest systems and don't focus on media fidelity or consistency in between releases. There is a lot of overlap with the interest of cultural and memory institutions, but in general, everybody seems to be working with by-products.

Jason Scott at the Internet Archive has been remarkably effective at encouraging emulator developers to work on problems relevant to preservation, but this is a personal rather than institutional effort. If emulation is to become a mainstream preservation technology some mechanism for commissioning fixes for critical problems from the emulation community will be needed.

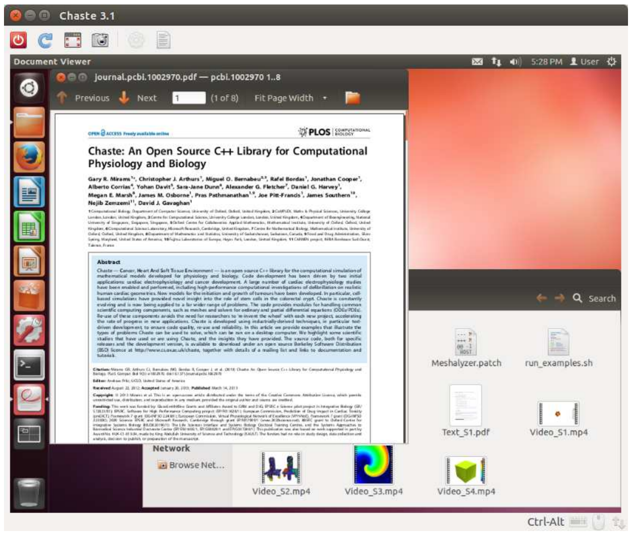

Chaste 3.1
Document Viewer
journal.pcbi.1002970.pdf — pcbi.1002970 1..8
Previous Next (1 of 8) Fit Page Width
Chaste: An Open Source C++ Library for Computational Physiology and Biology
Meshalyzer.patch
run_examples.sh
Text_S1.pdf
Video_S1.mp4
Network
Browse Net...
Video_S2.mp4
Video_S3.mp4
Video_S4.mp4
Search
Ctrl-Alt

Figure 19: Cloud emulation of Chaste 3.1 on Ubuntu 12.04 (2013)

2.4.2 Metadata

Migration-based preservation needs both *technical* metadata to enable migration and *bibliographic* metadata to assist users in finding preserved artefacts. Emulation-based preservation has similar requirements.

Tools, including JHOVE [48], DROID [129] and Siegfried [68], have been developed to extract technical metadata, and the PRONOM [130] database set up to unify recording of the results. None of this metadata is adequate for the needs of emulation. The tools cannot identify or record the dependencies that specify the emulator, operating system and application needed to emulate a preserved artefact.

The Freiburg team has worked with the German National Library (DNB) to test a workflow that generates the technical metadata needed for emulation access to CD-ROMs in the DNB's approximately 500,000 item collection of physical digital media. They classified a sample of 69 CDs, selected to reflect the diversity of the collection. The file systems on the CDs were typically ISO9660, or a hybrid of ISO9660 and Apple's HFS, and the first step of the workflow was to identify it. FITS [45] was used, but it had to be extended to identify HFS. Then FITS was used to identify the formats of the files in the file system(s). Then a matching process similar to John Ockerbloom's Typed Object Model for format migration [87] matched the formats with entries in their database of emulation environments that were compatible with all the observed formats. At least one suitable environment was found for 66 of the 69 CD-ROMs.

The metadata databases involved were custom XML. But efforts are under way to extend metadata standards such as the PREMIS data dictionary [88], and registries such as PRONOM [130], and TOTEM [62], for this purpose.

This represents an initial step towards the kinds of techncial metadata tools that would be needed to create and manage large collections of preserved system images. Without such tools the cost of, for example, creating technical metadata allowing all of DNB's 500,000 item collection to be emulated would be disproportionate to the benefits that allowing access would provide.

"Bibliographic" metadata is also needed so that the emulated artefacts can be found. Extracting this automatically will be even more challenging than it is for less diverse digital artefacts such as academic journals and e-books [105]. The Internet Archive's software collection shows that crowd-sourcing can be used to enhance such metadata provided some initial hand-created metadata is presented (Section 2.3.2).

Two other kinds of metadata may be important. The Internet Archive's collection shows the importance of *usability* metadata (see Section 2.3.2). Things like key bindings are essential information for an emulation's audience. They may be defined by, but not easily extracted from, the emulated software, or they may instead be properties of the emulation. Generating this metadata may be costly, or it may require crowd-sourcing as at the Internet Archive.

Usage metadata, such as those reported by bwFLA (see Section 2.4.4), are also important both for provisioning and for justifying the archive's funding, but if they are too detailed they raise privacy issues. The Internet Archive's praiseworthy concern for user privacy does make it difficult to know exactly how much use is being made of their emulations.

2.4.3 Fidelity

Fidelity to the original is a concern. There are two types of fidelity to consider, *execution* fidelity, whether the emulation executes the program correctly, and *experiential* fidelity, how close a user's experience is to the original user's experience.

In a Turing sense all computers are equivalent to each other, so it is possible for an emulator to replicate the behavior of the target machine's CPU and memory exactly, and most emulators do that. The only significant difference is in performance, and Moore's Law has meant that current CPUs and memories are so much faster than the systems that ran legacy digital artefacts that they may need to be artificially slowed to make the emulation realistic.

But a physical computer is more than a Turing machine. It has peripherals, which may appear to the CPU as I/O registers or memory locations, but whose behavior is not captured by that appearance. User interface peripherals have real-world attributes such as displays,

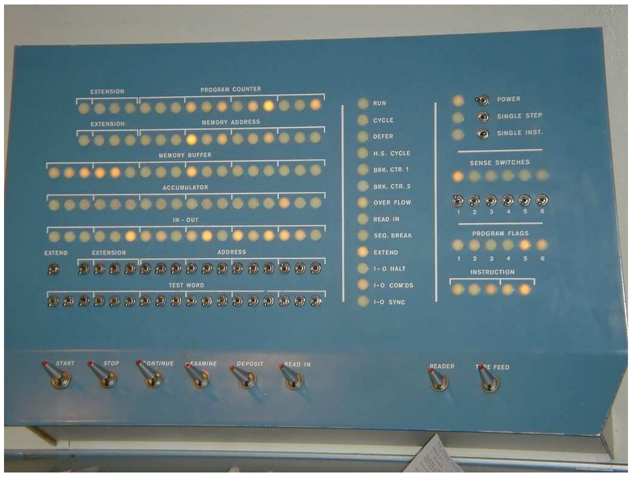


Figure 20: PDP-1 Control Panel, by fjarlq / Matt. Licensed under CC BY 2.0.

mice, keyboards, speakers and so on whose behavior is analog rather than digital.

Consider the emulation of Space Wars on the PDP-1 [49] (Figure 13). The experience of pointing and clicking at the Internet Archive's web page, pressing LEFT-CTRL and ENTER to start, watching a small patch in one window on your screen among many others, and controlling your spaceship from the keyboard is not the same as the original. That experience included loading the paper tape into the reader, entering the paper tape bootstrap from the switches, and pressing the Start switch (Figure 20). The program displayed on a large, round, flickering CRT. The player controls were idiosyncratic:

> Player controls include clockwise and counterclockwise rotation, thrust, fire, and hyperspace. Initially these were controlled using the front-panel test switches, with four switches for each player, but these proved to wear out very quickly under normal gameplay, and the location of the switches left one player off to one side of the CRT display and visually disadvantaged as a result. Most sites used custom control boxes wired into the same switches, although joysticks and other inputs were also used. [12]

Similarly, the key assignment issues with the Internet Archive's emulations of arcade and console games show that the experience of, for example, driving an emulation of a simulated race car [116] via the keyboard is not the same as the original experience of driving it with a suitable game controller. With games such as Wiz n' Liz [93], where accurate timing is important, this can significantly impact the emulation user's experience.

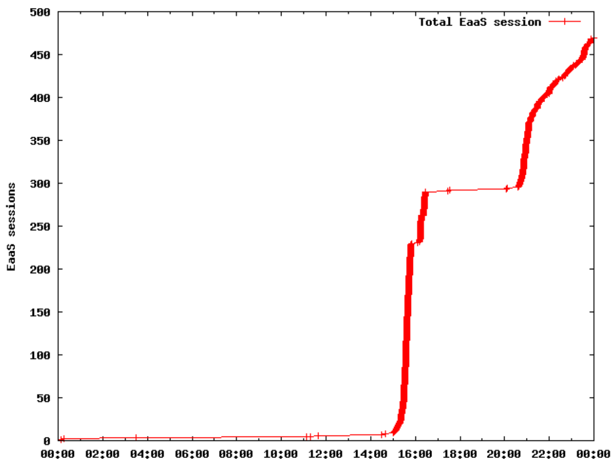


Figure 21: Release of *Bomb Iraq* [97]

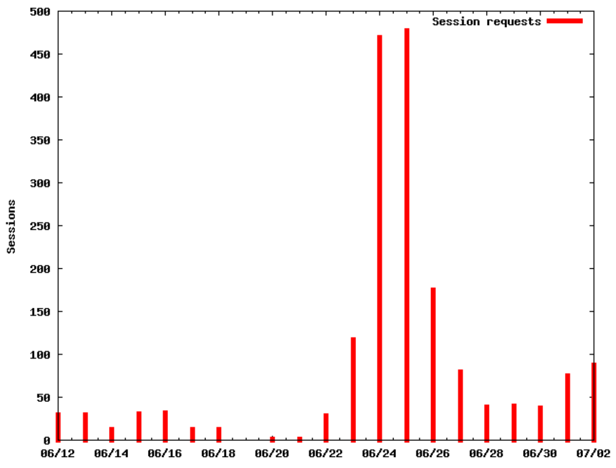


Figure 22: Daily load on bwFLA test/demo service [97]

2.4.4 Loads and Scaling

One advantage of frameworks such as the Internet Archive's (Section 2.2.3) and Olive's canonical configuration (Section 2.2.2) is that each additional user brings along with them the compute power needed to run their emulation. In such cases, the load on the infrastructure is merely that of delivering a quantity of static content, and thus equivalent to the load imposed by delivering the same quantity of migrated content. Frameworks in which the emulation runs remotely, however, must add resources to support added users.

Just as with the Theresa Duncan CD-ROMs (Section 2.3.1), when Rhizome released the *Bomb Iraq* emulation load spiked. Klaus Rechert described how the spike overloaded bwFLA's test and demo infrastructure [97] on which it was running, as shown by the flat section of Figure 21. Figure 22 shows that the peak was about 2 days at around twenty times previous typical load, a daily average rate around an emulation request every 3 minutes.

These experiences led Rhizome to develop the "front-end" described in Section 2.2.1, and to deploy their infrastructure on Amazon's highly scalable Elastic Beanstalk [4] infrastructure. Klaus Rechert computes:

> Amazon EC2 charges for an 8 CPU machine about €0.50 per hour. In case of [*Bomb Iraq*], the average session time of a user playing with the emulated machine was 15 minutes, hence, the average cost per user is about 0.02€ if a machine is fully utilized. [97]

In the peak, this would have been about €10/day, ignoring Amazon's charges for data out to the Internet. Nevertheless, automatically scaling to handle unpredictable spikes in demand always carries budget risks, and limits such as the queuing implemented by Rhizome's "front-end" are essential for cloud deployment.

2.4.5 Intellectual Property

Warnings: I Am Not A Lawyer, and this section is US-specific

Most library and archive institutions, the Internet Archive is an exception, are very reluctant to operate in ways whose legal foundations are less than crystal clear. There are two areas of law that affect using emulation to re-execute preserved software, copyright and, except for open-source software, the end user license agreement (EULA). This is a contract between the original purchaser and the vendor. Institutions generally lack a clear understanding of exactly what rights they acquired when they purchased software licenses, whether the rights cover execution in emulators and VMs, whether the rights cover software acquired other than by purchase such as by donation, and how long these rights last. They are thus motivated to err on the side of caution.

Software must be assumed to be copyright, and thus absent specific permission such as a Creative Commons or open source license, making persistent copies such as are needed to form collections of preserved system images is generally not permitted. The Digital Millennium Copyright Act (DMCA) contains a "safe harbor" provision under which sites that remove copies if copyright owners send "takedown notices" are permitted; this is the basis upon which the Internet Archive's collection operates. Further, under the DMCA it is forbidden to circumvent any form of copy protection or Digital Rights Management (DRM) technology. These constraints apply independently to every component in the software stack contained in a preserved system image, thus there may be many parties with an interest in an emulation's legality.

One often overlooked component is the font in which text is rendered, which is copyright. The owner of the

Item	Count
File extensions	5757
Operating systems	783
Application types	600+
Manufacturers	3807

Table 1: Cabrinety-NIST Metadata Stats [128]

copyright on the software probably licensed the fonts it uses, and thus cannot give permission for their emulated use. An example of this was the resurrection of *The Crossing*, a Pulitzer-winning web series that vanished from the Web when *The Rocky Mountain News* folded [65]. It took four years of negotiation with the newspaper's owners and the Denver Public Library to get permission to resurrect it, and then further negotiation with the font designer to get permission to use the newspaper's proprietary font. In this case the rights holders were known, it wasn't the "orphan font" problem [109].

In 1998 Stanford Libraries acquired the Cabrinety collection of pre-1995 software, hardware and other materials related to microcomputers. Fifteen years later, through the first two years of a project funded by the National Software Reference Library (NSRL), NIST and Stanford cooperated to image the systems and media, ingest the images and create metadata for them (Table 1). Now, the next phase is to contact copyright owners to ask their permission to provide access. This is a huge undertaking, made much worse by the "orphan works" problem for software[75]. The collection includes 12-15,000 software items, from 934 publishers [127]:

> Although we have made contact with some of the major publishers, there are still a significant number of titles for which we have not been able to identify rights holders. Discovery of rights holders is a difficult and time-consuming process. For many of the items in the Cabrinety collection it has been nearly impossible to determine whom to contact.

The Copyright Office's proposal to address "orphan works" [99] solves none of the problems they pose for software preservation [18], let alone the issues posed by EULAs, and by DRM, such as currently proposed for the widely used JPEG standard [72].

The Internet Archive and the Electronic Frontier Foundation have repeatedly worked through the triennial "Section 1201" process to obtain an exemption to the circumvention ban for:

> "computer programs and video games distributed in formats that have become obsolete and that require the original media or hardware as a condition of access, when circumvention is accomplished for the purpose of preservation or archival reproduction of published digital works by a library or archive. A format shall be considered obsolete if the machine or system necessary to render perceptible a work stored in that format is no longer manufactured or is no longer reasonably available in the commercial marketplace." [60]

The result of the most recent review was just announced [92], extending the exemption slightly to cover allowing single-player modes of multi-player games whose servers had been shut down.

This exemption appears to cover the Internet Archive's circumvention of any DRM on their preserved software, and its subsequent "archival reproduction" which presumably includes execution. It does not, however, exempt the archive from taking down preserved system images if the (claimed[3]) copyright owner objects, and the Internet Archive routinely does so. Similar notices are served on many sites, for example:

> This week, GitHub posted a takedown notice it received from Nintendo of America's legal representation. The Mario makers believe that a popular JavaScript-based Game Boy Advanced emulator hosting its source on GitHub violated the company's copyright for the games involved. [76]

Neither does the DMCA exemption cover the issue of whether the emulation violates the EULA. Given this legal uncertainty, it is not surprising that, apart from the Internet Archive's collection, only Rhizome among established cultural memory institutions allows public access to their preserved system images.

Digital media companies are notorious for their aggressive approach to enforcing their copyrights. Nevertheless, streaming media services such as Spotify, which do not result in the proliferation of copies of content, have significantly reduced although not eliminated intellectual property concerns around access to digital media. Software companies have similar attitudes to emulation. For example, although Microsoft upgrades physical systems from Windows 8 to Windows 10 for free, virtualizations such as Apple's Parallels and VMware require a paid upgrade [28]. "Streaming" emulation systems that do not result in proliferation of copies should have a similar effect on access to preserved digital artefacts. In permission-based structures, the service provider is the single point of negotiation with the rights owner. In objection-based structures, the service provider is the single point of contact for takedown notices.

[3]"claimed" because it frequently turns out that the claim is bogus. See for a recent example [74]. Contesting such claims would be a massive resource sink.

The success of the Internet Archive's collections, much of which can only be streamed, and Rhizome's is encouraging in this respect. Nevertheless, it is clear that institutions will not build, and provide access even on a restricted basis to, collections of preserved system images at the scale needed to preserve our cultural heritage unless the legal basis for doing so is clarified.

2.4.6 Curation Tools

Even if the legal basis for such collections were clarified, another barrier would remain, the cost of creating the preserved system images. At present, this is a heavily manual process. depending on the nature of the software in the image.

There are reports that base OS images might take 2-20 hours, depending on how well the OS is supported by QEMU, and the curator's familiarity with the OS. Commercial packaged software for an existing base VM might take an hour or two. Research software, which is typically of much lower quality than OSes and their packaged software, can take much longer; in some cases a few person-weeks.

2.5 Conclusions

As we see, emulation is an effective technique for preserving legacy digital artefacts. It is practical, having in at least two instances been used in public-facing deployments that have attracted substantial audiences. Both cloud- and browser-based emulations have demonstrated their ability to scale to meet demand peaks.

Despite emulation's ability to present a wide range of content types, from spreadsheets (Figure 14) and early Web browsers (Figures 17, 25) to recent scientific applications (Figure 19), the vast majority of preserved system images created so far, and the vast majority of emulation sessions have been of video games. This should not be a surprise. Using emulation for preservation was pioneered by video game enthusiasts. This reflects a significant audience demand for retro gaming which, despite the easy informal availability of free games, is estimated to be a \$200M/year segment [46] of the \$100B/year video games industry [81]. Commercial attention to the value of the game industry's back catalog is increasing:

> For the vast majority of video games that exist, though, the only way to legally obtain a copy is to track down original hardware and used software that may not have been produced for decades. Digital Eclipse is looking to change that, using a mix of technology and attention to historical detail to ensure that the classics of gaming remain in circulation in a cost-effective, accurate, and respectful manner.
>
> "Classic games are being devalued in the way they're released," Digital Eclipse's Head of Restoration Frank Cifaldi told Ars in an interview ... Cifaldi compared the company's efforts to The Criterion Collection, which makes definitive remastered prints of hundreds of classic movies, loads them with extra historical content, and keeps them in circulation through Blu-Ray and DVD sales. [91]

The video games industry is at least as big as the movie industry [124]; no-one thinks efforts to preserve movies are controversial. Major art museums have mounted shows treating video games as art, including the Smithsonian's American Art Museum [121] and the Victoria and Albert [15]. Because preserving content for scholars lacks the business model and fan base of retro gaming, it is likely that it will continue to be a minority interest in the emulation community.

There are relatively few preserved system images other than games for several reasons:

- The retro gaming community has established an informal *modus vivendi* with the copyright owners. Most institutions require formal agreements covering preservation and access and, just as with academic journals and books [104], identifying and negotiating individually with every copyright owner in the software stack is prohibitively expensive.

- If a game is to be successful enough to be worth preserving, it must be easy for an unskilled person to install, execute and understand, and thus easy for a curator to create a preserved system image. The same is not true for artefacts such as art-works or scientific computations, and thus the cost per preserved system image is much higher.

- A large base of volunteers is interested in creating preserved game images, and there is commercial interest in doing so. Preserving other genres requires funding. Techniques have been developed for mass preservation of, for example, Web pages, academic journals, and e-books, but research has shown that the available resources are sufficient to preserve less than half the artefacts that should be preserved [108]. No such mass preservation technology is available for emulations, and the cost per artefact preserved is many orders of magnitude higher.

Just as migration doesn't provide perfect mimicry of the original user's experience, neither does emulation. However perfectly the emulator mimics the original's behavior in the digital domain, the digital-to-analog and analog-to-digital channels connecting the emulator's digital domain to its user are different, in some cases radically different, from the original's (see Section 2.4.3). Of course, except for arcade systems, these channels were not identical between instances of the original either. But the differences now are significant enough in some cases to greatly impair the experience of the preserved digital artefact.

The bwFLA and Olive systems are strikingly similar. They use the same set of emulators, store their disk images in the same qcow2 format, use FUSE to render their images visible to the emulators, can be used in very similar configurations, and use XML metadata to describe the emulation environment of a preserved system image. Unfortunately, their XML metadata is system-specific, so they cannot share preserved system images. The key differences are:

- bwFLA has a more sophisticated, multi-layer way of composing their preserved system image from components, and does not require software to be installed on the user's device.
- Olive has a more sophisticated way of caching the blocks of their preserved system image at the system running the emulator, allowing local emulation over network connections whose bandwidth or latency would be inadequate for bwFLA.

The Olive team argue that representing the preserved system image as a single "bag of bits" is more robust; the multiple components of the bwFLA approach provide more potential modes of preservation failure. On the other hand, the bwFLA team argue that their approach can reduce the effort needed to create a functional preserved system image, allowing it to be built up by referring to pre-existing partial stacks.

3 The Last Two Decades of Evolution

Rothenberg's description of digital artefacts and their infrastructure was an accurate representation of the IT world at the time, but it bears little relation to the world in which digital artefacts are currently being created and used. Among the changes over the last two decades are the Web, the rise of interpreted programming languages, the advent of multimedia hardware (GPUs), and "the Cloud". These developments mean that preserving current digital artefacts requires techniques and faces problems that differ significantly from preserving digital artefacts from earlier eras.

3.1 Digital Artefacts

The five changes in the nature of digital artefacts over the last two decades with the greatest impact on their preservation are:

- The evolution of digital formats from being private to an individual application to being network protocols.
- The Web-enabled interconnectedness of digital artefacts.
- The evolution of the Web from a document model to a programming environment.
- The vast scale of the space of digital artefacts, and the resulting rise of "The Cloud".
- "Big Data" techniques.

3.1.1 Format Stability

One effect of the Web was to change the role of the formats in which information is encoded. As applications such as desktop publishing initially developed in the absence of network connectivity, the same application was the means by which the information was both *created* and *interpreted*. The format of the information was thus a private matter for the application. It could be changed at will, often as a way of motivating customers to upgrade. The "increasing returns" economics of technology markets [17] imply dominance by one, or at most a few, players. Thus most applications were doomed to fail in the market, leaving content created in their private format stranded. This was the common experience on which Rothenberg based his predictions.

Content on the Web is *published*, the application creating it cannot know what application will interpret it. The format must be standardized, either formally or informally. These standards must preserve *backwards compatibility*; there is no way to update previously published content. This effect applies not just to formats, such as HTML, intended for Web use, but also to widely used formats intended for other uses, such as the Microsoft Office formats. There is no way to prevent publishers using these formats.

Thus the advent of the Web was predicted to cause a massive reduction in the rate at which formats became obsolete [101]. Research [103] has confirmed this for Web formats, even for audio-visual formats from the early days of the Web which would have been expected to be especially vulnerable. As Microsoft discovered when they tried to remove support for some very old formats from the Office suite [102], their customers would no longer tolerate the costs of format obsolescence even for non-Web content.

3.1.2 Interconnectedness

Before the advent of the Web digital artefacts had easily identified boundaries. They consisted of a stack of components, starting at the base with some specified hardware, an operating system, an application program and some data. In typical discussions of digital preservation, the bottom two layers were assumed and the top two instantiated in a physical storage medium such as a CD.

The connectivity provided by the Internet and subsequently by the Web makes it difficult to determine where the boundaries of a digital object are. For example, the full functionality of what appear on the surface to be traditional digital documents such as spreadsheets or PDFs can invoke services elsewhere on the network, even if only by including links. Link resolution involves two network services:

- Domain Name Service (DNS), to map from the name of the Web server to its IP address.
- The Web server at that IP address, to deliver the content pointed to by the link.

The crawlers that collect Web content for preservation have to be carefully programmed to define the boundaries of their crawls. Doing so imposes artificial boundaries, breaking what appears to the reader as a homogeneous information space into discrete digital "objects".

Indeed, what a reader thinks of as "a web page" typically now consists of components from dozens of different Web servers [64], most of which do not contribute to the reader's experience of the page. They are deliberately invisible, implementing the Web's business model of universal fine-grained surveillance.

Although it is still possible to create digital artefacts that have well-defined boundaries and do not depend on any network services, it is no longer either easy or common to do so. It takes a careful application programmer to avoid using any library that uses a network service. Disconnected operation is now an unusual requirement, and when it is implemented it is often an after-thought that restricts the functionality of the application. Disconnected operation breaks the business model of pervasive surveillance, so it is implemented reluctantly.

3.1.3 Activity

Sir Tim Berners-Lee's original Web [23] was essentially an implementation of Vannevar Bush's Memex hypertext concept [29], an information space of passive, quasi-static hyper-linked documents. The content a user obtained by dereferencing a link was highly likely to be the same as that obtained by a different user, or by the same user at a different time.

It is worth noting that the very first US website, at SLAC in 1991, was a front-end to a dynamic database [106]. Since then, the Web has gradually evolved from the original static linked document model whose language was HTML, to a model of interconnected programming environments whose language is JavaScript. Indeed, none of the emulation frameworks described here would be possible without this evolution. The probability that two dereferences of the same link will yield the same content is now low, the content is dynamic. This raises fundamental questions for preservation; what exactly does it mean to "preserve" an artefact that is different every time it is examined?

Although no-one would argue that JavaScript is an ideal programming environment, its ubiquity allows some to argue that over time it will displace most other programming environments [24]. JavaScript is already widely used in browsers and servers for purposes far distant from its original task:

> This kind of wide-ranging usage led Microsoft's Scott Hanselman to dub JavaScript the "assembly language for the Web," a sentiment largely shared by people such as Brendan Eich, who invented JavaScript, and Douglas Crockford, who invented JSON, widely used for JavaScript-based data interchange.
>
> But the people calling for a bytecode for the browser never went away, and they were never entirely wrong about the perceived advantages. And now they're going to get their wish. WebAssembly is a new project being worked on by people from Mozilla, Microsoft, Google, and Apple, to produce a bytecode for the Web.
>
> WebAssembly, or wasm for short, is intended to be a portable bytecode that will be efficient for browsers to download and load, providing a more efficient target for compilers than plain JavaScript or even asm.js. ... The people behind wasm have not forgotten that JavaScript is supported everywhere and wasm is currently not supported anywhere. Their plan is to fill the gap with a polyfill; a JavaScript script that will convert wasm to asm.js for those browsers that don't have native wasm support. Either the browser will interpret the wasm directly, or it will load the polyfill and execute the resulting asm.js. Native handling should be faster, but the polyfill means that a developer can be sure that a wasm program will always work. [27]

WebAssembly has the potential to become a very widely accepted virtual machine for all sorts of digital artefacts, and it could significantly improve the performance of JavaScript emulation technology such as JSMESS.

3.1.4 Scale and The Cloud

In 1995, a typical desktop 3.5" hard disk held 1-2GB of data. Today, the same form factor holds 4-8TB, about 4000 times as much. In 1995, there were estimated to be 16 million Web users, Today, there are estimated to be over 3 billion, nearly 200,000 times as many [58]. At the end of 1996, the Internet Archive estimated the total size of the Web at 1.5TB [59], but today they ingest that much data roughly every 30 minutes [54].

The technology has grown, but the world of data has grown much faster, and this has transformed the problems of preserving digital artefacts. Take an everyday artefact such as Google Maps. It is simply too big and worth too much money for any possibility of preservation by a third party such as an archive, and its owner has no interest in preserving its previous states.

3.1.5 Big Data

Traditionally, scholars accessed archived material by using metadata to locate individual items [1]. Once vast collections of data became available, tools were needed to make them useful. Although these "Big Data" tools were initially developed for commercial and intelligence applications such as search engines, scholars rapidly found that the ability to ask a question of an entire collection of information in one operation, rather than of a single document at a time, could transform research. Text search and aggregate statistical analysis are increasingly the favored access methods.

3.2 Digital Infrastructure

The developments in the digital infrastructure with the biggest impact on preservation in general and emulation in particular include the evolution of the processing hardware to include a parallel processing element, the change in the systems in which the processing hardware is embedded to mobile devices, the slowing of Moore's Law, the dominance of a few instruction set architectures, and the rapid increase in the threats posed by malware.

3.2.1 GPUs

As Rothenberg was writing, PC hardware was undergoing a major architectural change. The connection between early PCs and their I/O devices was the ISA bus [9], whose bandwidth and latency constraints made it effectively impossible to deliver multimedia applications such as movies and computer games. This was replaced by the PCI bus [7], with much better performance. This opportunity led us[4] to start NVIDIA in 1993, and in 1995 deliver our first chip. NV1 [10] was the first hardware capable of running arcade games such as Sega's *Virtua Fighter* at full frame rate on a PC [126].

The demands of the highly competitive games market forced a division of the PC architecture into a Central Processing Unit (CPU) and what became known as Graphics Processing Units (GPUs). The reason was that CPUs were essentially sequential processors, incapable of performing the highly parallel task of rendering the graphics fast enough to deliver an acceptable user experience. Now, much of the silicon in essentially every device with a user interface implements a massively parallel GPU whose connection to the display is both very high bandwidth and very low latency. Most high-end scientific computation now also depends on the massive parallelism of GPUs rather than traditional supercomputer technology.

[4]Jen-Hsun Huang, Curtis Priem and Chris Malachowsky as founders, myself as #4

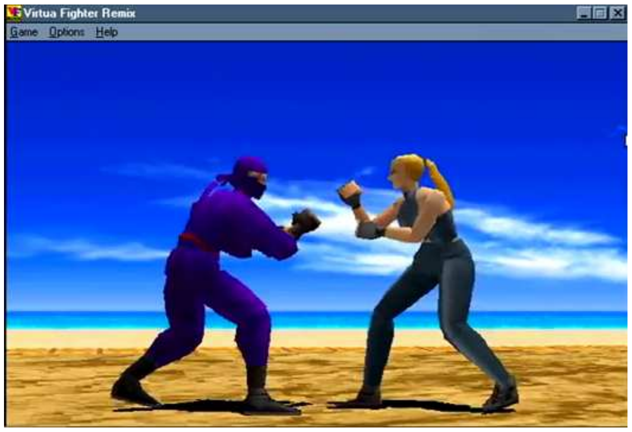


Figure 23: Sega's *Virtua Fighter* on NVIDIA NV1 hardware (1995) [126]

Vendor	2Q15	2Q14	YoY Growth
Lenovo	13,444	14,535	-7.5%
HP	12,253	13,675	-10.4%
Dell	9,560	10,466	-8.7%
Apple	5,136	4,423	16.1%
Acer Group	4,334	5,932	-26.9%
ASUS	4,330	4,693	-7.7%
Others	17,082	21,274	-19.7%
Total	66,140	74,998	-11.8%

Table 2: IDC: PC Shipments (Kunits) [120]

NV1 is a case where even emulation is not a practical way to preserve the content. The games produced for it were tightly coupled to the NV1 chip's hardware interface. That interface was actually designed to be easy to emulate; it consisted of hardware that virtualized all the on-chip resources [107]. But the functionality of those resources was highly proprietary, and protected as trade secrets. NVIDIA developed the chip's drivers using an emulation of the chip, but it was very slow, never released and likely hasn't survived.

It would have been extraordinarily difficult to create a clean-room emulator for the chip, and it is unlikely that even much faster modern GPUs would help much. The reason that NV1 was so fast was that, unlike its competitors and all successor GPUs, it did not render surfaces by dividing them into huge numbers of tiny triangles, but into a small number of curved primitives called quadric patches. Thus there is a mis-match between its application's data structures and the capabilities of subsequent hardware.

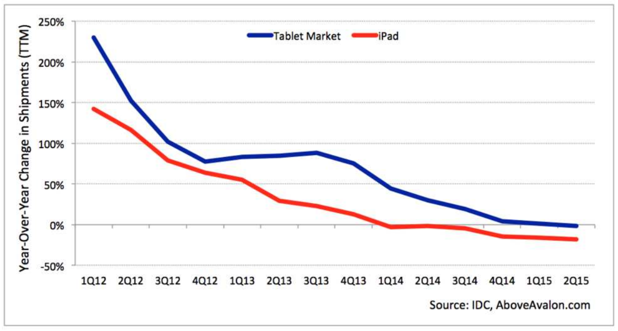


Figure 24: Tablet & iPad shipment growth rate [37]

3.2.2 The Rise of the Smartphones

In most cases, the hardware by which digital artefacts were experienced two decades ago was a desktop PC or Mac, laptops were at that time expensive and a small part of the market. Now, both desktop and laptop PC sales are in free-fall (Table 2), and even tablet sales are no longer growing (Figure 24).

Smartphones are the hardware of choice. They, and tablets, amplify the interconnectedness of Section 3.1.2; they are designed not as autonomous computing resources but as interfaces to the Internet.

The concept of a stand-alone "application" is no longer really relevant to these devices. Their "App Store" supplies custom front-ends to network services, as these are more effective at implementing the Web's business model of pervasive surveillance. Apps are notoriously difficult to collect and preserve. Emulation can help with their tight connection to their hardware platform, and indeed most apps are developed in emulators, but not with their dependence on network services.

Both desktop and laptop PCs provided a very homogeneous set of user interface technologies. A display with a gradually increasing resolution through time, a keyboard with a fairly standard set of keys, a tracking device such as a mouse or a track-pad, speakers and a microphone. An application usable with one PC's user interface hardware would be highly likely to be usable with another's.

The user interface hardware of mobile devices is much more diverse. In some cases the hardware is technically compatible with traditional PCs, but not functionally compatible. For example, mobile screens typically are both smaller and have much smaller pixels, so an image from a PC may be displayable on a mobile display but it may be either too small to be readable, or if scaled to be readable may be clipped to fit the screen. In other cases the hardware isn't even technically compatible. The physical keyboard of a laptop and the on-screen virtual keyboard of a tablet are not compatible. For example, try using EMACS on a tablet. It depends on holding down combinations of modifier keys such as SHIFT, CTRL and ALT, while pressing character keys. The emulations used to develop apps depend on the greater UI resources of desktop environments. As desktop environments become less available, these emulations become less useful for preservation.

3.2.3 Moore's Law

Gordon Moore predicted in 1965 [82] that the number of transistors per unit area of a state-of-the-art integrated circuit would double about every two years. For about the first four decades of Moore's Law, what CPU designers used the extra transistors for was to make the CPU faster. This was advantageous for emulation; the modern CPU that was emulating an older CPU would be much faster. The computational cost of emulating the old hardware in software would be swamped by the faster hardware being used to do it.

Although Moore's Law continued into its fifth decade, each extra transistor gradually became less effective at increasing CPU speed. Further, as GPUs took over much of the intense computation, customer demand evolved from maximum performance per CPU, to processing throughput per unit power. The extra transistors were used to provide multiple CPU cores per chip, and to reduce power consumption per unit computation, rather than to make individual CPUs faster. Emulation is a sequential process, so the fact that the CPUs are no longer getting rapidly faster is disadvantageous for emulation.

Virtualization does not incur the software overhead of emulation, and is thus less impacted by the slowing of CPU speed increase. Compute-intensive digital artefacts have evolved to either exploit the parallelism of multi-core CPU chips, or to use GPUs. They are thus less amenable to emulation and more to virtualization, if GPUs could be virtualized.

3.2.4 Architectural Consolidation

W. Brian Arthur's 1994 book *Increasing Returns and Path Dependence in the Economy* [17] described the way the strongly increasing returns to scale in technology markets drove consolidation. Over the past two decades this has happened to system architectures. Although it is impressive that MESS emulates nearly two thousand different systems from the past, going forward emulating only two architectures (Intel and ARM) will capture the overwhelming majority of digital artefacts. Much of the remaining minority will be game consoles, and these are likely to remain un-emulated. Their CPU is likely to be emulate-able, but they depend heavily on their GPU and their I/O devices, (e.g. Kinect, Wii). Emulating a current game console's GPU is likely to remain beyond the state of the art (See Section 4.2.1).

3.2.5 Threats

Although the Morris Worm [63] took down the Internet in 1988, the Internet environment two decades ago was still fairly benign. The tools for distributed denial of service (DDoS) attacks [38] had yet to appear; Trinoo [39] in 1997 [83] was probably the first. There were so few Internet users that commerce hadn't taken over the Internet, so the money to be made from theft and fraud wasn't significant.

Now, Internet crime is one of the world's most profitable activities, as can be judged by the fact that a single zero-day iPhone exploit sells for $1M [47]. Because users are so bad at keeping their systems up-to-date with patches, once a vulnerability is exploited it becomes a semi-permanent feature of the Internet. For example, the Conficker worm appeared in November 2008 [6], and by January 2009 was estimated to have infected at least 10M hosts. In mid-2011 Microsoft was still detecting 1.7M infections each quarter [80].

This threat persistence is a particular concern for emulation as a preservation strategy. *Familiarity Breeds Contempt* by Clark *et al.* [31] shows that the interval between discoveries of new vulnerabilities in released software *decreases* through time. Thus the older the preserved system image, the (exponentially) more vulnerabilities it will contain.

4 Looking Forward

Looking forward, each of these developments will have an impact on emulation as a preservation strategy.

4.1 Impact of Artefact Evolution

The greatly reduced rate of format obsolescence caused by the Web greatly reduces the need for either migration or emulation as a way of interpreting individual documents. But the fact that current digital artefacts are no longer individual static documents raises a set of problems for preservation in general and emulation in particular that need to be addressed.

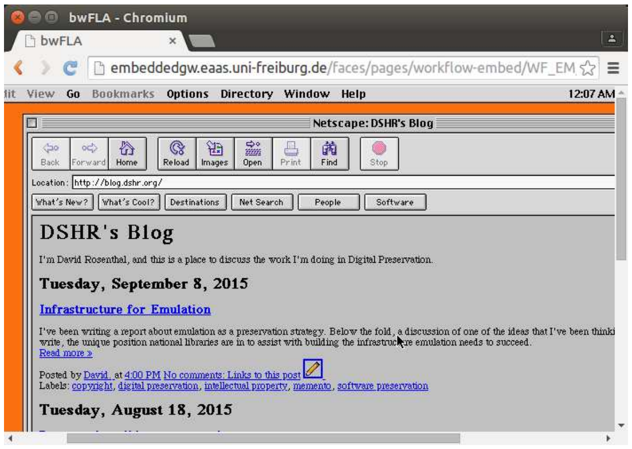
4.1.1 Interconnectedness

The original user experience of a current digital artefact is probably generated by the execution of a program, the content of some local files, and the set of Internet resources that it accesses. The desired user experience of emulating the preserved artefact could be generated, among others, by one of the following combinations:

1. The original program, local files, and Internet resources as they existed at the time the artefact was ingested. An example would be Rhizome's art piece "untitled[scrollbars]" [66].

2. The original program, local files, and same set of Internet resources but as they exist at the time of emulation. An example would be investigating how the evolution of Web pages affects browser rendering if users do not update their browsers.

3. The original program, local files, and a different set of the Internet resources as they exist at the time of emulation. An example would be investigating how old Web browsers render current Web content. Dragan Espenschied quickly put together such an example mostly from existing pieces, an emulation of Netscape 3 (1997) on MacOS 9 (1999) displaying blog.dshr.org (2015) (see Figure 25).


bwFLA - Chromium
bwFLA
embeddedgw.eaas.uni-freiburg.de/faces/pages/workflow-embed/WF_EM
lit View Go Bookmarks Options Directory Window Help 12:07 AM
Netscape: DSHR's Blog
Back Forward Home Reload Images Open Print Find Stop
Location: http://blog.dshr.org/
What's New? What's Cool? Destinations Net Search People Software

DSHR's Blog

I'm David Rosenthal, and this is a place to discuss the work I'm doing in Digital Preservation.

Tuesday, September 8, 2015

Infrastructure for Emulation

I've been writing a report about emulation as a preservation strategy. Below the fold, a discussion of one of the ideas that I've been thinki write, the unique position national libraries are in to assist with building the infrastructure emulation needs to succeed.
Read more »

Posted by David at 4:00 PM No comments: Links to this post
Labels: copyright, digital preservation, intellectual property, memento, software preservation

Tuesday, August 18, 2015

Figure 25: Current DSHR's blog via Netscape 3 on Mac OS 9 (1999)

Case 1 requires that the set of Internet resources accessed by the emulated program be ingested and preserved with it. Identifying this set is problematic, it must be based on running the program during ingest via an archiving proxy such as the Institut National de l'Audiovisuel's Live Archiving Proxy [53]. The emulation cannot be connected directly to the Internet because, during emulation, accesses to Internet resources must be redirected to the versions acquired during ingest.

4.1.2 Activity

Even if the set of resources accessed by the program as executed during ingest is correctly identified and preserved, user inputs and environmental state changes may result in the emulation attempting to access resources that were not preserved. Thus, to support case 1 above, an "Internet Emulator" must be interposed between the emulation and the live Internet. The most effective way to implement this would be to follow the example of Ilya Kreymer's prototype page reconstructor. The ingest phase would identify the resources accessed by the program and submit them for preservation to one or more Web archives. The Internet Emulator would use Memento (RFC7089 [135]) to redirect the emulated program's access to the preserved version of the resource closest to the time of ingest. If the resource was in the set identified during ingest, this version would be the one collected at that time. Otherwise, it would be the most appropriate version collected by a Web archive in its normal crawling.

If the emulated program is connected directly to the Internet, the result will be case 2 or 3. But, as we shall see in Section 4.2.3, there are other reasons an Internet Emulator will be needed.

Note that Olive's preserved VMs are themselves Web resources. Memento could be used by Olive clients to obtain the preserved VM closest in time to the date the emulation is intended to replicate.

4.1.3 Scale

From April to October of 2009 the ArchiveTeam succeeded in preserving 640GB of GeoCities before Yahoo shut the service down [16]. It isn't clear how much of the site this represents, but it is clearly a significant fraction. In addition, the Internet Archive did a deep crawl before the shutdown [55], these pages are available via the Wayback Machine. Other projects also preserved parts of the site, so overall there is a high probability that a GeoCities page can be re-accessed.

A recent study of the "islands" many Universities created in the heyday of Second Life reported:

> Most of these virtual universities are gone — it costs almost $300 per month to host your own island — but it turns out a handful remain as ghost towns. I decided to travel through several of the campuses, to see what's happening in Second Life college-world in 2015.
>
> First, I didn't see a a single other user during my tour. They are all truly abandoned. [52]

Even assuming that the Universities had continued paying for their islands, would similar efforts to those for GeoCities be effective in preserving even just the University islands if Second Life announced a shutdown? Clearly not:

- It required an intensive, crowd-sourced and distributed effort to obtain at most a few Terabytes of GeoCities data in six months of 2009. The data underlying the islands in Second Life is far larger.
- More significantly, the underlying data is not accessible to users; all users see is a rendering of that data.
- Even if the underlying data were accessible, it is useless without Linden Labs software to render it, and the software is unlikely to be available for preservation.
- The scale of the compute infrastructure needed to re-animate the islands is much larger than anything that has been attempted in preservation emulation.
- Finally, preserving the deserted architecture of the islands preserves only a small fraction of their significance. Far more of the meaning of the islands was contained in the social interactions they engendered, which were not preserved by Second Life, let alone by a third party.

Although in theory both bwFLA and Olive can emulate the multiple machines needed to implement the kinds of distributed systems being built today, neither has done so at a scale matching commonplace digital artefacts such as Google Maps, or important scientific computations such as climate models [32]. It isn't clear that the resources available for digital preservation could support this scale of effort.

4.1.4 Big Data

Emulation is a technique for accessing individual items; it does not lend itself to the kinds of aggregate access that "Big Data" techniques need. However, if large collections of preserved system images in standardized formats became available, they might themselves be the object of study using such techniques. Although this would not need emulation as such, it could be a useful side-effect of the adoption of emulation for preservation.

4.2 Impact of Infrastructure Evolution

4.2.1 GPUs

The NV1 example in Section 3.2.1 shows the problems GPUs pose for preservation. The QEMU project is making some efforts to support GPUs via their paravirtualized `virtio-gpu` virtual device (Section 2.1.1), based on earlier work from Red Hat [2], but this has significant limitations:

- As a paravirtualized device it needs a `virtio-gpu` driver, so far available only for Linux. Similar drivers could be written for other operating systems but they are a substantial amount of work for highly skilled programmers.
- It supports only the use of GPUs for 3D rendering via OpenGL, not for other tasks such as scientific computing.
- It offloads 3D rendering to the host's GPU, so it requires that the host have a GPU somewhat faster than that of the system being emulated.

The QEMU team's goals for `virtio-gpu` are realistic but limited. Really emulating the hardware of a state-of-the-art GPU as of a few years ago, even given a different but current GPU, is infeasible both because it would be far too much work for the available resources, and because much of the information that would be needed is trade secret, not to be released for this purpose even under non-disclosure.

Thus, although emulation as a strategy for preserving GPU-based but non-graphical artefacts such as large-scale scientific simulations is a theoretical possibility, lack of resources and intellectual property restrictions mean it will remain impractical, and these artefacts will remain unpreservable.

It is worth noting that the MIAOW project at the University of Wisconsin [19] is prototyping an open source hardware GPU with a partial copy of AMD's GPU architecture by implementing it on an FPGA[5]. This is only for testing; it is much slower than a current GPU. The IP issues of actually cloning an AMD GPU in real hardware with competitive performance appear as yet unresolved. But it is an indication that FPGA technology, such as implemented in Bunnie Huang's Novena laptop [51], could enable emulation of some sufficiently old GPUs with adequate performance.

4.2.2 Mobile

Given the sales figures from Section 3.2.2, it seems clear that in the medium term the hardware currently used to access emulations of legacy digital artefacts will become a specialist item. It will probably be available in library "reading rooms" and academics' offices, but the general public will expect to access these artefacts using a smartphone.

Current emulations are generally unusable on a smartphone. Even with considerable attention to emulating the basic I/O devices such as the keyboard on-screen (bwFLA and Olive can do this for some emulators), the experiential fidelity is very low. The keyboard takes up much of the limited screen real-estate, using the modifier keys is painfully awkward, and typing is much slower and more error-prone.

The diversity of user interface experience between, say, a high-end MacBook driving a 27-inch 2560 by 1440 display with a keyboard and touchpad, and an iPhone 5s with a 4-inch 1136 by 640 display and a touchscreen, is vastly larger that the diversity designers of legacy digital artefacts expected to handle. Attempts to span that gap will inevitably impair experiential fidelity. This is an area where, because it is not expected to replicate nearly as much of the user's experience, migration is much less impacted.

4.2.3 Threats

The fact that the functions of most current digital artefacts are either degraded or non-existent without an Internet connection poses many problems for preservation, but perhaps the most serious for emulation is that of security. The software to be emulated will have vulnerabilities that were not known at the time it was ingested. These vulnerabilities will have been discovered and exploited subsequent to ingestion. Exploits for these vulnerabilities will be present in the Internet at the time of emulation, so the emulated software will be compromised. There are two possibilities, but neither of them represent the user's experience at the time of ingestion, which is the goal of emulation:

- Either these vulnerabilities were fixed, and the software patched in the field subsequent to ingestion, and the archive were (implausibly) able to collect the patches and apply them to the software it ingested,

- Or the software was emulated in its original state but encountered exploits unknown at the time of ingest.

Compromised software is likely to be an accurate replication of the original user experience; there is no automated way to distinguish between the positive and negative aspects of that experience. Olive explicitly states that they regard the original software's vulnerabilities as an essential part of the emulation, as they may themselves be the object of study. Other frameworks adopt the same position.

Unfortunately, the goal of many compromises is to use the victim as a weapon for attacking other systems. Thus, even if the user of the emulation were untroubled by the compromise of their emulated system, it would not be ethical to allow their emulated system to attack others.

Thus an important if minimal pre-condition for making emulation safe for networked digital artefacts is the development of encapsulation techniques for Internet Emulators capable of preventing emulations of old software being compromised in ways that affect other systems. Clearly, technologies such as encrypted communications mean that doing this perfectly is beyond the state of the art. Institutions providing emulations will need to provide them with Internet Emulators with very restrictive default policies, to which exceptions would be granted on a case-by-case basis. This would seem to be the least they could do to avoid liability for knowingly putting other systems at risk.

The threat space is also one where emulation may be important for preservation. Web formats suffer obsolescence when the browsers no longer support them. It used to be argued that there was little motivation for browsers to drop support for old formats widely used on the Web [101], but recently the motivation has appeared. Adobe's inability to maintain the security of Flash at an acceptable level is leading other browser developers to follow Steve Jobs lead in deprecating Flash, and sites to stop using it:

[5] Field Programmable Gate Array.

> Five years ago, 28.9% of websites used Flash in some way, according to Matthias Gelbmann, managing director at web technology metrics firm W3Techs. As of August, Flash usage had fallen to 10.3%.
>
> But larger websites have a longer way to go. Flash persists on 15.6% of the top 1,000 sites, Gelbmann says. That's actually the opposite situation compared to a few years ago, when Flash was used on 22.2% of the largest sites, and 25.6% of sites overall. [85]

If browsers won't support Flash because it poses an unacceptable risk to the underlying system, much of the currently preserved Web will become unusable. It is true that some of that preserved Web is Flash malware, thus simply asking the user to enable Flash in their browser is not a good idea. But if Web archives emulated a browser with Flash, either remotely or locally, the risk would be greatly reduced. Even if the emulation fell victim to the malware, the underlying system would be at much less risk. If the goal of the malware was to use the compromised system as part of a botnet, the emulation's short lifecycle would render it ineffective. Users would have to be warned against input-ing any sensitive information that the malware might intercept, but it seems unlikely that many users would send passwords or other credentials via a historical emulation. And, because the malware was captured before the emulation was created, the malware authors would be unable to update it to target the emulator itself rather than the system it was emulating.

4.3 Conclusions

It is clear that preserving today's digital artefacts is far more difficult and costly than preserving the legacy objects for which migration, and to a much lesser extent, emulation is currently used. Neither emulation nor migration is a panacea for current digital artefacts. Nevertheless, if some areas could be addressed, emulation could play a much greater part in the preservation of current artefacts than it has so far for legacy artefacts.

5 Sustainability

Experience with preservation of e-journals, for which there are two dominant technologies (Portico's and LOCKSS'), and the Web, for which there is one dominant technology (Internet Archive's), suggests that "increasing returns" economics applies to digital preservation just as it does to other aspects of information technology. Thus it is likely that if emulation becomes an important aspect of digital preservation, it will be dominated by one, or at most a few, generic technologies that are "good enough" for most use cases. Niche technologies and customized solutions for particular use cases are unlikely to be viable.

5.1 Emulators

The pool of open source emulators is actively maintained for reasons unconnected with formal preservation. Legacy system emulators are maintained by retro gaming enthusiasts, and this is likely to remain the case. The maintenance load will tend to decrease through time (see Section 2.4.1). The emulator for current systems (QEMU) is part of mainstream Linux and BSD development, and this is likely to remain the case. The concern here is that as QEMU evolves to meet current demands, its ability to emulate older systems tends to degrade (see Section 2.4.1). There is currently no effective mechanism for supporting QEMU development specifically for preservation requirements (see Section 6.6).

5.2 Frameworks

Olive's sustainability plan is to seek grant funding for a 3-year pilot program that would involve committed users from a number of Universities and other memory institutions. Goals for this program would be to understand the range of use cases, develop processes for crowdsourcing preserved system images, and work out how long-term sustainability would be achieved, and which parties would be involved. The current concept is that some non-profit organization like Ithaka would provide the stable service on a subscription basis. Further development of the service, for example emulation of GPUs, would require research-level skills and research grant funding.

A number of funders including DFG, Baden-Württemberg and NEH have granted bwFLA research and pilot funding covering the next 2-3 years. Beyond that they envisage a similar future to Olive; a non-profit to operate the system and research funding for major developments.

Neither Olive nor bwFLA appears to have a detailed enough view of issues involved in their chosen sustainability path, including the need for startup funding to get the subscription service up to viability, and the legal complications that the service would encounter as it sold access to run software it did not own.

If more recent and larger digital artefacts were to be successfully emulated, pricing models for the services would be a tricky issue. At present, the cost of performing an individual emulation is small, not much larger than the cost of delivering a journal article from an archive such as JSTOR (see Section 2.4.4). So an all-you-can-eat pricing model based on institution size similar to JSTOR's might be appropriate. But the cost of performing an emulation of a large, distributed scientific computation would be significant, and thus usage-based pricing would be needed. This is much more complex and expensive to manage, and much more difficult to sell.

Unlike Olive and bwFLA, the Internet Archive is not primarily promoting its framework, which is in any case less formal and leverages other open source technologies such as Emscripten to greater extent.

5.3 Collections

The only current example of a large collection is the Internet Archive's, and it is this collection that they are promoting. The framework technology in use is subsidiary. Providing public access to the emulations is just one aspect of the institution's core mission, and is sustained as other parts are by a mix of staff resources, in-house volunteers and crowd-sourcing. It is hard to imagine any other institution being able to replicate this model. The collection is to a large extent the result of dynamic leadership and its curation is of very variable quality.

Given the current tools and legal frameworks, collecting, preserving and providing access to large, well-curated collections of preserved system images would be very expensive in staff time. The DNB's efforts to use bwFLA and automated metadata tools on its 500,000 item collection of physical digital media will be an interesting example [98]. The restricted scope of both the collection and the access mechanism (”reading room” only), give grounds for optimism.

5.4 Business Models and Competition

There is already a small market supporting emulation for retro gaming. Swiss company Stromasys [5] has been in business for over a decade providing emulations of PDP-11, VAX, AlphaServer, HP 3000, and SPARC environments for companies to run legacy software. The success of QEMU shows that there is significant market demand for emulation as a tool for developing current digital artefacts. Amazon's Device Farm [3] uses real rather than emulated smartphones and tablets to allow developers to test their apps against a wide range of devices.

Nevertheless, it is an open question as to whether the market would support a product from a major vendor such as Amazon or Google aimed at emulating legacy environments. If the market demand justified it, these companies would have leverage in negotiating licenses from the copyright owners to allow their customers to run legacy software that cultural heritage institutions could only dream of. They would have the technical resources to support the tools and emulators. Their existing charging models would support payments to the copyright owners where cultural heritage institutions would have to create new models, and would encounter pricing issues (see Section 5.2).

Another set of commercial considerations relate to the fact that most of the emulators are supported by retro gaming enthusiasts. To the extent that the market for retro gaming motivates companies such as Digital Eclipse [91] to re-issue old games for new systems, the motivation to support running the original games in emulation, and thus the motivation to support the emulators, decreases. Retro gaming is already estimated to be a $200M/yr market [46], so this effect could be a significant problem in the future.

6 The To-Do List

If emulation is to become a major digital preservation strategy the cost per preserved system image must be greatly reduced:

> Automation is essential, since the huge number of objects prevent manual handling of each object, in particular in view of the determination of technical metadata required for emulation. [98]

This leads to the following goals:

- The processes for extracting and encoding the technical metadata of a digital artefact, and packaging the artefact and the metadata into a preserved system image, must be as cheap as possible.
- Once packaged, the preserved system image must be reused as much as possible, thus maximizing the return on the initial investment. The artefact is unchanging, so it should not be necessary to re-extract and re-encode its metadata.
- There will be multiple emulators and emulation frameworks, and they will evolve through time. Re-extracting or re-packaging preserved artefacts for different, or different versions of, emulators or emulation frameworks would be wasted effort.
- The most appropriate framework configuration for a given user will depend on many factors, including the bandwidth and latency of their network connection, and the capabilities of their device. Thus the way in which emulations are advertised to users, for example by being embedded in a Web page, should not specify a particular framework or configuration; this should be determined individually for each access.

The last of these is both especially important and also dependent upon the others for its realization. Lets take the Theresa Duncan CD-ROMs at `http://emulators.rhizome.org/theresa-duncan-cdroms` as an example. That URL redirects to `http://theresa-duncan-eaas.s3-website-us-east-1.amazonaws.com/`. Several Mementos of this, probably evanescent, page are preserved at the

Internet Archive, the first from May 2, 2015 `https://web.archive.org/web/20150502122945/http://theresa-duncan-eaas.s3-website-us-east-1.amazonaws.com/`. but the Play links were not followed and collected. Even if they had been, they point, for example, to `http://eaas-client-dev4.elasticbeanstalk.com/game/chop-suey`, which is clearly an evanescent target. *If the access paths to the emulations link directly to evanescent services emulating the preserved artefacts, not to the artefacts themselves, the preserved artefacts are not themselves discoverable or preservable.*

At present, if the same original bits forming a digital artefact are to be accessed via different frameworks, *even if those frameworks use the same underlying emulator*, they must be separately packaged to create a different preserved system image for each framework, and a different access path, such as a link from a Web page, created for each.

The tasks implied by these goals are as follows.

6.1 Standardize Preserved System Images

Despite the considerable similarities between the ways bwFLA and Olive package the bits of a preserved system image, the results are not interoperable. A standard, framework-independent format for the bits and the associated metadata should be agreed. This will require agreement on:

- A format for the bits (bwFLA and Olive both use qcow2).
- A format for the metadata (bwFLA and Olive both use XML)
- A controlled vocabulary for the metadata.
- Whether the metadata is packaged with the bits (Olive) or links to the bits (bwFLA). The capability of linking to the bits is required for bwFLA's stacking approach to representing the bits (see Section 2.2.1).

6.2 Standardize Access To System Images

Once the format of the preserved system images has been standardized, there needs to be a standard way for browsers to recognize that when a link to the preserved image is resolved, that the way the content is to be rendered is via emulation. A Mime-Type binding to a process that selected and invoked an appropriate emulator, based on hints in the XML metadata, the user's browser and other configuration information, would be one way to do this.

6.3 Standardize Invoking Emulators

The emulator selection process needs a standard way to invoke the selected emulator, which means that the various emulators need to be encapsulated in a standard wrapper as bwFLA does (see Figure 3).

6.4 Improving Tools For Preserving System Images

One critical barrier to widespread adoption of emulation is the cost of creating a preserved system image. Tools to automate the process are essential. Because there is no agreement among the emulation systems on the output format of these tools, this interface is effectively internal to the emulation system. The tools can thus in practice only be developed by that emulator's team, so there is no competition to spur development.

If there were agreement on the desired output format, tool development could proceed independently of emulator development, and potentially with much broader involvement and more rapidly.

6.5 Metadata Databases

Both these tools and the emulator selection process of Section 6.2 require access to databases of technical metadata. There was general agreement that the existing PRONOM format metadata did not provide the detail needed for emulation, and that the process of adding to and updating it was too cumbersome. Some less formal mechanism is needed to maintain a database of the metadata discussed in Section 6.1 "in front of" the PRONOM database; entries would migrate to PRONOM as they stabilized over time, and as PRONOM was extended to support emulation-specific technical metadata.

PRONOM supports file format identification tools whose state even for non-emulated Web formats is less than ideal [71] Their use to support emulation is discussed in Section 2.4.2. Dragan Espenschied writes:

> Existing PRONOM tools, like DROID, JHOVE, etc, are very clunky and monolithic. Siegfried is a nice exception. For emulation, disk images are much more important than classic collection of files. And these disk images might be created with many different file systems. The PRONOM tools can only read very simple archives like ZIP, no way they could read ISO or COW images, let alone HFS file systems or more exotic specimen. Siegfried has planned to implement scanning WARC files, which would deserve support.
>
> I have the feeling that the fixation on the tools having to be available on Windows systems and as "cross platform GUI" holds the field

back. For Linux/BSD systems, there are multiple free/libre implementations of file system and archive mounters available.

Siegfried, with its lightweight implementation and JSON output is a potential basis for a tool of this kind.

6.6 Emulator Support

The emulation efforts reviewed all draw from the same pool of open source emulators, and all share concerns about the ability of the emulation community to support their use for preserving the broad range of digital artefacts needed. The preservation community could address these concerns by:

- Pooling resources to employ and direct (some time of) the skilled programmers needed.
- Pooling resources to institute a "bounty" program that would redirect some effort from the existing community to preservation needs.
- Raising money to host a "hackathon" event that would bring the emulation community together to target preservation needs.

6.7 Internet Emulators

As discussed in Section 4.2.3, emulations of current digital artefacts will need an Internet Emulator more sophisticated than bwFLA's "software wire". Two features in particular are needed:

- The ability at the user's choice to use Memento [135] to obtain the most time-appropriate version of network resources accessed by an emulated artefact, instead of the current version. Ilya Kreymer's page reconstructor could be a starting point for an implementation.
- "Sandboxing" capabilities with configurable policies and a highly restrictive default to, at a minimum, prevent emulated artefacts being used to attack other systems.

6.8 Legalities

The unclear legal environment is the major barrier to developing the large collections of preserved system images that would make emulation broadly useful.

Discussions have been under way for some time between lawyers from Carnegie Mellon and a software vendor to explore a licensing agreement that would allow the Olive archive to build and provide some limited access to a substantial collection of commercial software. The hope is that if a license could be drafted with which both sides could live, it could act as a template for other software vendors.

UNESCO has a mandate to assist its member states in preserving their national heritage, including the variety of born digital content. A 2012 international conference entitled Memory of the World in the Digital Age: Digitization and Preservation [132] led to the UNESCO/UBC Vancouver Declaration [133] and started an on-going series of discussions called PERSIST in which both CMU and Microsoft are active. The next meeting is in Paris in November 2015. The discussions cover three areas related to digital artefacts:

- Criteria for selection and retention of culturally significant artefacts.
- Technical issues in providing persistent access to collections of preserved digital artefacts.
- Policy issues surrounding building and providing access to collections of preserved digital artefacts.

The goal of the initial discussions is to create a consensus that such collections are important, and that all parties should work together to figure out how to make them a reality.

As regards the technical issues, emulation is clearly an essential part of the solution, and its importance will increase as digital objects become more complex and dynamic through time. Fortunately, as the examples above show, the remaining technical issues posed by emulation are relatively trivial.

As regards the policy issues, clarification of some viable legal basis for building and providing some form of access to these collections is essential. In 2010 the Blue Ribbon Task Force on Sustainable Digital Preservation and Access [14] pointed out that the only real justification for preservation is to provide access.

The legal and economic barriers to the archive of software apparently envisaged by PERSIST are formidable. The intellectual property frameworks, primarily copyright and the contract law underlying the End User License Agreements (EULAs), under which software is published differ from country to country At least in the US, where much software originates, these frameworks make collecting, preserving and providing access to collections of software impossible except with the specific permission of every copyright holder. The situation in other countries is similar. International trade negotiations such as the TPP are being used by copyright interests to make these restrictions even more onerous [125].

For the hypothetical operator of the software archive to identify the current holder of the copyright on every software component that should be archived, and negotiate permission with each of them for every country involved, would be enormously expensive. Research has shown that the resources devoted to current digital preservation efforts, such as those for e-journals, e-books and the

Web, suffice to collect and preserve less than half of the material in their scope. Absent major additional funding, diverting resources from these existing efforts to fund a global software archive would be robbing Peter to pay Paul.

Worse, the fact that the global software archive would need to obtain permission *before* ingesting each publisher's software means that there would be significant delays before the collection would be formed, let alone be effective in supporting scholars' access.

An alternative approach worth considering would separate the issues of permission to collect from the issues of permission to provide access. Software is copyright. In the paper world, many countries had copyright deposit legislation allowing their national library to acquire, preserve and provide access (generally restricted to readers physically at the library) to copyright material. Many countries, including most of the major software producing countries, have passed legislation extending their national library's rights to the digital domain.

The result is that most of the relevant national libraries already have the right to acquire and preserve digital works, although not the right to provide unrestricted access to them. Many national libraries have collected digital works in physical form. For example, the DNB's CD-ROM collection includes half a million items [98]. Many national libraries are crawling the Web to ingest Web pages relevant to their collections [50].

It does not appear that national libraries are consistently exercising their right to acquire and preserve the software components needed to support future emulations, such as operating systems, libraries and databases. A simple change of policy by major national libraries could be effective immediately in ensuring that these components were archived. Each national library's collection could be accessed by emulations on-site in "reading-room" conditions, as envisaged by the DNB. No time-consuming negotiations with publishers would be needed. This concept is discussed in more detail in [110].

If the legal basis for such collections is clarified, institutions other than possibly national libraries would need to defray the costs of maintaining and providing access. Whatever form of access might eventually be permitted must allow for a sustainable business model, but the PERSIST discussions do not appear to have reached that level of detail.

One idea that might be worth exploring as a way to mitigate the legal issues is lending. The Internet Archive has successfully implemented a lending system for their collection of digitized books [56]; readers can check a book out for a limited period, and each book can be checked out to at most one reader at a time. This has not encountered much opposition from copyright holders.

Task	Difficulty	Impact	Cost
Standard System Images	Low	Medium	Low
Standard Access to Images	Low	Medium	Low
Standard Invoking	Low	Low	Low
Improved Tools	Medium	High	Medium
Metadata Databases	Low	Low	Low
Emulator Support	Low	Medium	Medium
Internet Emulators	Low	Low	Low
Legalities	High	High	High

Table 3: Ranking To-Do Items

A similar system for emulation would be feasible; readers would check out an emulation for a limited period, and each emulation could be checked out to at most one reader at a time. One issue would be dependencies. An archive might have, say, 10,000 emulations based on Windows 3.1. If checking out one blocked access to all 10,000 that might be too restrictive to be useful.

6.9 Ranking

Table 3 shows my personal ranking of the To-Do items above on three axes, each with values Low, Medium and High.

- *Difficulty* is my assessment of the intrinsic difficulty of the item. For example, the striking similarity between the formats of current system images suggests that standardizing this should be easy.
- *Impact* is my assessment of the short-term effect on the use of emulation of successfully completing the item. For example, improved tools for creating preserved system images should lead to a significant increase in their availability.
- *Cost* is my assessment of the likely cost of successfully completing the item. For example, the protracted and difficult negotiations needed to clarify the legal basis for collecting and emulating preserved system images, and the expensive legal talent needed to undertake them, suggest a very high cost.

7 Conclusion

Research has shown that current digital preservation efforts, such as those for academic journals and the Web, save less than half the material they target [108]. Their preservation strategies are overwhelmingly based on migration. The evolution of digital artefacts means that current artefacts are more difficult and expensive to collect and preserve than those from the past, and less suitable for migration. This trend is expected to continue.

Recent developments in emulation frameworks make it possible to deliver emulations to readers via the Web in ways that make them appear as normal components of Web pages. This removes what was the major barrier to deployment of emulation as a preservation strategy. Barriers remain, the two most important are that the tools for creating preserved system images are inadequate, and that the legal basis for delivering emulations is unclear, and where it is clear it is highly restrictive. Both of these raise the cost of building and providing access to a substantial, well-curated collection of emulated digital artefacts beyond reach.

If these barriers can be addressed, emulation will play a much greater role in digital preservation in the coming years. It will provide access to artefacts that migration cannot, and even assist in migration where necessary by allowing the original software to perform it.

Emulation for preservation has a significant overlap with the techniques needed to ensure reproducibility of *in silico* science, such as those of the Workflow4Ever project [136]. Olive's preservation of Chaste 3.1 (Figure 19, Section 2.3.3) is an example of emulation applied to this problem. Increasing pressure for reproducibility might direct resources to areas synergistic with emulation for preservation, but both areas are hampered by similar legal and organizational problems. The overlap is in any case less than it appears, since the techniques being proposed involve significant constraints on how the "research object" is created [96]. Attempts to impose constraints on the production of general digital artefacts to improve their preservability have not been effective.

Emulation is not a panacea. Technical, scale and intellectual property difficulties make many current digital artefacts infeasible to emulate. Where feasible, even with better tools and a viable legal framework, emulation is more expensive than migration-based strategies. The most important reason for the failure of current strategies to collect and preserve the majority of their target material is economic; the resources available are inadequate. The bulk of the resources expended on both migration and emulation strategies are for ingest, especially metadata generation and quality assurance. There is a risk that diverting resources to emulation, with its higher per-artefact ingest cost, will exacerbate the lack of resources.

Acknowledgments

This work was funded by the Andrew W. Mellon Foundation on behalf of themselves, the Sloan Foundation and the US Institute of Museum and Library Services. Grateful thanks are due to all those who spent time explaining and demonstrating their systems, and answering my many questions. Despite all this help, any opinions expressed outside attributed quotes are those of the author alone.

Particular thanks to Jason Scott, Prof. Mahadev Satyanarayanan, Benjamin Gilbert, Gloriana St. Clair, Euan Cochrane, Dragan Espenschied, Klaus Rechert, Gerd Hoffman, Henry Lowood, Natasa Milic-Frayling, and the anonymous reviewers.

References

[1] Ian F. Adams, Ethan L. Miller, and Mark W. Storer. Analysis of workload behavior in scientific and historical long-term data repositories. Technical Report UCSC-SSRC-11-01, University of California, Santa Cruz, March 2011.

[2] Dave Airlie. Virgil 3D GPU project. `https://virgil3d.github.io/`.

[3] Amazon. AWS Device Farm. `https://aws.amazon.com/device-farm/`.

[4] Amazon. AWS Elastic Beanstalk. `https://aws.amazon.com/elasticbeanstalk/`.

[5] Anon. Charon (software). `https://en.wikipedia.org/wiki/Charon_(Software)`.

[6] Anon. Conficker. `https://en.wikipedia.org/wiki/Conficker`.

[7] Anon. Conventional PCI. `https://en.wikipedia.org/wiki/Conventional_PCI`.

[8] Anon. DOSBox. `https://en.wikipedia.org/wiki/DOSBox`.

[9] Anon. Industry Standard Architecture. `https://en.wikipedia.org/wiki/Industry_Standard_Architecture`.

[10] Anon. NV1. `https://en.wikipedia.org/wiki/NV1`.

[11] Anon. Paravirtualization. `https://en.wikipedia.org/wiki/Paravirtualization`.

[12] Anon. Spacewar (video game). `https://en.wikipedia.org/wiki/Spacewar!`

[13] Anon. UAE (emulator). `https://en.wikipedia.org/wiki/UAE_(emulator)`.

[14] Anon. Blue Ribbon Task Force on Sustainable Digital Preservation and Access. `http://brtf.sdsc.edu/`, 2008.

[15] Sebastian Anthony. *Hearthstone* exhibited as modern art at the V&A Museum in London. `http://arstechnica.co.uk/gaming/2015/09/hearthstone-exhibited-as-modern-art-at-the-va-museum-in-london/`, September 2015.

[16] Archive Team. GeoCities Project. http://www.archiveteam.org/index.php?title=Geocities_Project.

[17] W. Brian Arthur. *Increasing Returns and Path Dependence in the Economy*. University of Michigan Press, 1994.

[18] Lila Bailey. Digital Orphans: The Massive Cultural Black Hole On Our Horizon. https://www.techdirt.com/articles/20151009/17031332490/digital-orphans-massive-cultural-black-hole-our-horizon.shtml, October 2015.

[19] Raghuraman Balasubramanian, Vinay Gangadhar, Ziliang Guo Chen-Han Ho, Cherin Joseph, Jaikrishnan Menon, Mario Paulo Drumond, Robin Paul, Sharath Prasad, Pradip Valathol, and Karthikeyan Sankaralingam. MIAOW - An Open Source RTL Implementation of a GPGPU. In *IEEE Cool Chips*, 2015.

[20] Christian Bauer. Basilisk II: an Open Source 68K Macintosh Emulator. http://basilisk.cebix.net/.

[21] Christian Bauer. SheepShaver: An Open Source PowerMac Emulator. http://sheepshaver.cebix.net/.

[22] BBC. Warning of data ticking time bomb. http://news.bbc.co.uk/1/hi/technology/6265976.stm.

[23] Tim Berners-Lee. WorldWideWeb: Summary. http://goo.gl/KKrqSJ, August 1991.

[24] Gary Bernhardt. The Birth & Death of JavaScript. https://www.destroyallsoftware.com/talks/the-birth-and-death-of-javascript, April 2014.

[25] Dan Bricklin. VisiCalc: A Visible Calculator for the APPLE][. http://www.bricklin.com/history/refcard1.htm.

[26] Dan Bricklin and Bob Frankston. VisiCalc (1979) (Software Arts). https://archive.org/details/VisiCalc_1979_SoftwareArts.

[27] Peter Bright. The Web is getting its bytecode: WebAssembly. http://arstechnica.com/information-technology/2015/06/the-web-is-getting-its-bytecode-webassembly/, June 2015.

[28] Jon Brodkin. Year-old Parallels and VMware software won't be updated for Windows 10. http://arstechnica.com/information-technology/2015/08/year-old-parallels-and-vmware-software-wont-be-updated-for-windows-10/, August 2015.

[29] Vannevar Bush. As we may think. *The Atlantic Monthly*, July 1945.

[30] bwFLA. The Digital Heritage of Vilèm Flusser. http://bw-fla.uni-freiburg.de/demo-flusser.html, August 2015.

[31] Sandy Clark, Stefan Frei, Matt Blaze, and Jonathan Smith. Familiarity breeds contempt. *ACSAC 2010*, pages 251–260, 2010.

[32] ClimatePrediction.net. Top Teams. http://climateapps2.oerc.ox.ac.uk/cpdnboinc/top_teams.php.

[33] CNRI. Handle.net. http://handle.net/.

[34] Euan Cochrane. Emulation as a Service (EaaS) at Yale University Library. http://blogs.loc.gov/digitalpreservation/2014/08/emulation-as-a-service-eaas-at-yale-university-library/, August 2014.

[35] Computer History Museum. PDP-1 Restoration Project. http://www.computerhistory.org/pdp-1/index.php.

[36] CRIU Development Team. Checkpoint/Restore In Userspace. http://criu.org/Main_Page, April 2015.

[37] Neil Cybart. Finding iPad's Future. http://www.aboveavalon.com/notes/2015/8/3/finding-ipads-future, August 2015.

[38] Sven Dietrich, Neil Long, and David Dittrich. Analyzing distributed denial of service tools: The shaft case. In *USENIX LISA 2000*, December 2000.

[39] David Dittrich. The DoS Project's "trinoo" distributed denial of service attack tool. https://staff.washington.edu/dittrich/misc/trinoo.analysis, October 1999.

[40] DOSBox Crew. DOSBox. http://www.dosbox.com/.

[41] Theresa Duncan. Smarty. http://emulators.rhizome.org/theresa-duncan-cdroms.

[42] Theresa Duncan. Zero Zero. http://emulators.rhizome.org/theresa-duncan-cdroms.

[43] Theresa Duncan and Monica Gesue. Chop Suey. http://emulators.rhizome.org/theresa-duncan-cdroms.

[44] Dragan Espenschied. Acknowledgement, Circulation, Obscurity, System Ambience. http://rhizome.org/editorial/2014/jun/24/emulating-bomb-iraq-arcangel/, June 2014.

[45] FITS Development Team. File Information Tool Set (FITS). http://projects.iq.harvard.edu/fits.

[46] Timothy Geigner. Retro Games Industry Booming Despite Pirate-Options Being Super Available. https://www.techdirt.com/articles/20150817/09563431981/retro-games-industry-booming-despite-pirate-options-being-super-available.shtml, August 2015.

[47] Dan Goodin. Security firm pledges $1 million bounty for ios jailbreak exploits. *Ars Technica*, September 2015.

[48] Harvard University Library. Format-Specific Digital Object Validation. http://hul.harvard.edu/jhove/, 1999.

[49] Hingham Institute. Spacewar! for the PDP-1. https://archive.org/details/pdp1_spacewar.

[50] Helen Hockx-Yu. Ten years of archiving the UK Web. In *10th International Digital Curation Conference*, February 2015.

[51] Bunnie Huang. Novena. https://www.crowdsupply.com/sutajio-kosagi/novena.

[52] Laura Hudson. Exploring the abandoned digital campuses of second life universities. *Boing Boing*, August 2015.

[53] INA. LiveArchivingProxy. https://github.com/INA-DLWeb/LiveArchivingProxy.

[54] Internet Archive. Bandwidth. https://archive.org/stats/.

[55] Internet Archive. GeoCities Special Collection 2009. https://archive.org/web/geocities.php.

[56] Internet Archive. Internet Archive Lending Library. https://archive.org/details/lendinglibrary.

[57] Internet Archive. Software Library. https://archive.org/details/softwarelibrary.

[58] Internet World Stats. Internet Growth Statistics. http://www.internetworldstats.com/emarketing.htm.

[59] Brewster Kahle. Preserving the internet. *Scientific American*, 276(3):82–83, 1997.

[60] Brewster Kahle. Internet Archive Helps Secure Exemption To The Digital Millennium Copyright Act. https://archive.org/post/82097/internet-archive-helps-secure-exemption-to-the-digital-millennium-copyright-act, November 2006.

[61] KEEP Project. KEEP Emulation Framework [EF]. http://emuframework.sourceforge.net/.

[62] KEEP Project. Trustworthy Online Technical Environment Metadata Registry. http://www.keep-totem.co.uk/.

[63] Brendan P. Kehoe. The Internet Worm. http://www.cs.indiana.edu/docproject/zen/zen-1.0_10.html#SEC91, January 1992.

[64] Georgios Kontaxis and Monica Chew. Tracking protection in firefox for privacy and performance. In *IEEE Web 2.0 Security & Privacy*, January 2015.

[65] Adrienne LaFrance. Raiders of the Lost Web. *The Atlantic*, October 2015.

[66] Jan Robert Leegte. untitled[scrollbars]. http://www.leegte.org/works/online/composition_blue/index.htm, 2000.

[67] Jan Robert Leegte. untitled[scrollbars] (emulated). http://hdl.handle.net/11270/5a23663e-67da-43ba-9a35-83ab89bb5bed, 2000.

[68] Richard Lehane. Siegfried. http://www.itforarchivists.com/siegfried.

[69] Thomas Liebetraut, Klaus Rechert, Isgandar Valizada, Konrad Meier, and Dirk Von Suchodoletz. Emulation-as-a-Service - The Past in the Cloud. In *Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on*, pages 906–913. IEEE, 2014.

[70] Tim Lindholm, Frank Yellin, Gilad Bracha, and Alex Buckley. *The Java virtual machine specification*. Pearson Education, 2014.

[71] LOCKSS Program. LOCKSS: Format Migration. `http://documents.clockss.org/index.php/LOCKSS:_Format_Migration`.

[72] Jeremy Malcolm. There's No DRM in JPEG—Let's Keep It That Way. `https://www.eff.org/deeplinks/2015/10/theres-no-drm-jpeg-lets-keep-it-way`, October 2015.

[73] MAME Development Team. Multi Arcade Machine Emulator. `http://www.mame.net/`.

[74] Mike Masnick. Boston Public Broadcaster WGBH Files Bogus DMCA Notice On Public Domain Video Uploaded By Carl Malamud. `https://www.techdirt.com/articles/20150821/16573132031/boston-public-broadcaster-wgbh-files-bogus-dmca-notice-public-domain-video-uploaded-carl-malamud.shtml`, August 2015.

[75] Mike Masnick. Happy Birthday And The Problem With The Copyright Office's 'Orphan Works' Plan. `https://www.techdirt.com/articles/20151005/18115432443/happy-birthday-problem-with-copyright-offices-orphan-works-plan.shtml`, October 2015.

[76] Nathan Matisse. Nintendo asks GitHub to make Javascript-based Game Boy emulator disappear. `http://arstechnica.com/gaming/2015/07/nintendo-asks-github-to-make-javascript-based-game-boy-emulator-disappear/`, July 2015.

[77] Neil McAlister. Heartbleed, eat your heart out: VENOM vuln poisons countless VMs. `http://www.theregister.co.uk/2015/05/13/heartbleed_eat_your_heart_out_venom_vuln_poisons_countless_vms/`.

[78] Mark McLoughlin. The QCOW2 Image Format. `http://people.gnome.org/~markmc/qcow-image-format.html`, September 2008.

[79] MESS Development Team. Welcome to the MESS Wiki. `http://www.mess.org/`.

[80] Microsoft. Microsoft Security Intelligence Report: Volume 11. `http://goo.gl/dUhvPm`, 2011.

[81] Brett Molina. Gartner: Global video game market to hit $93B this year. `http://www.usatoday.com/story/tech/gaming/2013/10/29/gartner-worldwide-video-game-market/3294591/`, October 2013.

[82] Gordon E Moore et al. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):82–85, 1998.

[83] Viki Navratilova. Trinoo: First publicly available ddos tool (c. 1997) (slide 11). In *Uniforum Chicago*, August 2000.

[84] New Museum. Explore the worlds of three feminist video games from the 1990s in this online exhibition. `http://www.newmuseum.org/exhibitions/view/the-theresa-duncan-cd-roms`.

[85] Jared Newman. The agonizingly slow decline of adobe flash player. *Fast Company*, August 2015.

[86] Shaun Nichols. Guy puts 1990s MacOS 7 on an Apple Watch – without jailbreaking it. *The Register*, June 2015.

[87] J. Ockerbloom. Mediating Among Diverse Data Formats. Technical Report CMU-CS-98-102, Carnegie-Mellon University, 1998.

[88] OCLC. PREMIS (PREservation Metadata: Implementation Strategies) Working Group. `http://www.oclc.org/research/projects/pmwg/`, 2005.

[89] OED. Artefact. `http://www.oed.com/view/Entry/11133`.

[90] Olive Archive. Virtual Machines in Our Collection. `https://olivearchive.org/docs/collection/`.

[91] Kyle Orland. The new tech making game preservation more authentic and future-proof. `http://arstechnica.com/`

gaming/2015/08/the-new-tech-making-game-preservation-more-authentic-and-future-proof/, August 2015.

[92] Kyle Orland. US gov't grants limited right to revive games behind "abandoned" servers. http://arstechnica.com/gaming/2015/10/u-s-govt-grants-limited-right-to-revive-games-behind-abandoned-servers/, October 2015.

[93] Psygnosis. Wiz n' Liz. https://archive.org/details/sg_Wiz_n_Liz_1993_Psygnosis_US, 1993.

[94] QEMU. QEMU Emulator User Documentation. http://qemu.weilnetz.de/qemu-doc.html#QEMU-System-emulator-for-non-PC-targets.

[95] QEMU. QEMU: Open Source Processor Emulator. http://wiki.qemu.org/Main_Page.

[96] Andreas Rauber, J. Binder, T. Miksa, R. Mayer, S. Pröll, S. Strodl, and M. Unterberger. Digital curation and preservation of large and complex scientific objects: Preserving Complex Scientific Objects - Process Capture and Data Identification. https://indico.cern.ch/event/332370/session/3/contribution/31, June 2015.

[97] Klaus Rechert. EaaS in Action — And a short meltdown due to a friendly DDoS. http://openpreservation.org/blog/2014/07/09/eaas-action-and-short-meltdown-due-friendly-ddos/, July 2014.

[98] Klaus Rechert, Thomas Liebetraut, Oleg Stobbe, Isgandar Valizada, and Tobias Steinke. Characterization of CD-ROMs for Emulation-based Access. In *iPRES*, November 2015.

[99] Register of Copyrights. Orphan Works and Mass Digitization. http://copyright.gov/orphan/reports/orphan-works2015.pdf, June 2015.

[100] Adi Robertson. The girl game archival project that's rewriting geek history. http://www.theverge.com/2015/4/17/8436439/theresa-duncan-chop-suey-cd-rom-preservation, April 2015.

[101] David S. H. Rosenthal. Format Obsolescence: Scenarios. http://blog.dshr.org/2007/04/format-obsolescence-scenarios.html, April 2007.

[102] David S. H. Rosenthal. Format Obsolescence: Right Here Right Now? http://blog.dshr.org/2008/01/format-obsolescence-right-here-right.html, January 2008.

[103] David S. H. Rosenthal. Formats through time. http://blog.dshr.org/2012/10/formats-through-time.html, October 2012.

[104] David S. H. Rosenthal. "Preservation at Scale" at iPRES2013. http://blog.dshr.org/2013/09/preservation-at-scale-at-ipres2013.html, September 2013.

[105] David S. H. Rosenthal. Talk on LOCKSS Metadata Extraction at IIPC 2013. http://blog.dshr.org/2013/04/talk-on-lockss-metadata-extraction-at.html, April 2013.

[106] David S. H. Rosenthal. First US Web Page. http://blog.dshr.org/2014/11/first-us-web-page.html, November 2014.

[107] David S. H. Rosenthal. Hardware I/O Virtualization. http://blog.dshr.org/2014/12/hardware-io-virtualization.html, December 2014.

[108] David S. H. Rosenthal. The Half-Empty Archive. http://blog.dshr.org/2014/03/the-half-empty-archive.html, March 2014.

[109] David S.H. Rosenthal. Are format specifications important for preservation? http://blog.dshr.org/2009/01/are-format-specifications-important-for.html, January 2009.

[110] David S.H. Rosenthal. Infrastructure for Emulation. http://blog.dshr.org/2015/09/infrastructure-for-emulation.html, September 2015.

[111] David S.H. Rosenthal. It takes longer than it takes. http://blog.dshr.org/2015/02/it-takes-longer-than-it-takes.html, February 2015.

[112] Jeff Rothenberg. Ensuring the Longevity of Digital Documents. *Scientific American*, 272(1), 1995.

[113] Mahadev Satyanarayanan, Gloriana St. Clair, Benjamin Gilbert, Yoshihisa Abe, Jan Harkes, Dan Ryan, Erika Linke, and Keith Webster. One-Click Time Travel. Technical report, Computer Science, Carnegie Mellon University, June 2015.

[114] Jason Scott. JSMESS Wiki. https://github.com/jsmess/jsmess/wiki.

[115] Jason Scott. Still Life, With Emulator: The JSMESS FAQ. https://blog.archive.org/2013/12/31/still-life-with-emulator-the-jsmess-faq/.

[116] Sega. Internet Arcade: Out Run. https://archive.org/details/arcade_outrun, 1986.

[117] Simon Sharwood. Guest-host escape bug sees Xen project urge rapid upgrade. http://www.theregister.co.uk/2015/06/29/xen_project_urges_urgent_hypervisor_upgrade/.

[118] Simon Sharwood. QEMU may be fro-Xen out after two new bugs emerge. http://www.theregister.co.uk/2015/08/04/xen_project_ponders_breakup_with_qemu_after_two_new_bugs_found/.

[119] Simon Sharwood. Xen reports new guest-host escape, this time through CD-ROMs. http://www.theregister.co.uk/2015/07/28/xen_reports_new_guesthost_escape_this_time_through_cdroms/.

[120] Simon Sharwood. PC sales go OFF A CLIFF to under 300 million a year. http://www.channelregister.co.uk/2015/07/10/pc_sales_go_off_a_cliff_to_under_300_million_a_year/, July 2015.

[121] Smithsonian Institution. The Art of Video Games. http://www.americanart.si.edu/exhibitions/archive/2012/games/, March 2012.

[122] Software Freedom Conservancy. QEMU is Conservancy's Newest Member Project. http://sfconservancy.org/news/2015/jul/23/qemu-joins/http://www.dosbox.com/.

[123] SPICE Development Team. SPICE. http://www.spice-space.org/.

[124] Statista. Filmed entertainment revenue worldwide from 2015 to 2019. http://www.statista.com/statistics/259985/global-filmed-entertainment-revenue/.

[125] Maira Sutton. Users to USTR: Don't Sign Away Our Ability to Fix the Orphan Works Problem. https://www.eff.org/deeplinks/2015/08/users-ustr-dont-sign-away-our-ability-fix-orphan-works-problem, August 2015.

[126] swaaye. NVIDIA NV1 (Virtua Fighter Remix for NV1). https://www.youtube.com/watch?v=VK9sg_93iCE.

[127] Charlotte Thai. Copy That. https://web.stanford.edu/group/htgg/cgi-bin/drupal/?q=node/1189, April 2014.

[128] Charlotte Thai. What the Hell is It and What Do I Do with It? http://www.slideshare.net/charthai/what-the-hell-is-it-and-what-should-i-do-with-it-cataloging-challenging-collections, May 2014.

[129] UK National Archives. Digital Record Object Identification. http://sourceforge.net/projects/droid/, February 2010.

[130] UK National Archives. The Technical Registry PRONOM. http://www.nationalarchives.gov.uk/PRONOM/Default.aspx, February 2010.

[131] UK Web Archive. Interject. http://www.webarchive.org.uk/interject/.

[132] UNESCO. The Memory of the World in the Digital age: Digitization and Preservation. http://www.unesco.org/new/en/communication-and-information/events/calendar-of-events/events-websites/the-memory-of-the-world-in-the-digital-age-digitization-and-preservation/, September 2012.

[133] UNESCO. UNESCO/UBC Vancouver Declaration. http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/CI/CI/pdf/mow/unesco_ubc_vancouver_declaration_en.pdf, September 2012.

[134] J. R. van der Hoeven, R. J. van Diessen, and K. van der Meer. Development of a universal virtual computer (uvc) for long-term preservation of digital objects. *Journal of Information Science*, 31(3):196–208, 2005.

[135] Herbert van der Sompel, Michael Nelson, and Robert Sanderson. RFC 7089: HTTP Framework for Time-Based Access to Resource States – Memento. `http://tools.ietf.org/html/rfc7089`, December 2013.

[136] Workflow4Ever. Workflow4Ever Project. `http://www.wf4ever-project.org/`.

[137] Alon Zakai. Emscripten: An LLVM-to-JavaScript Compiler. `https://github.com/kripken/emscripten`.