

QEMU - A Crucial Building Block in Digital Preservation Strategies

Dirk von Suchodoletz, Klaus Rechert, and Achille Nana Tchayep

Albert-Ludwigs University Freiburg, Hermann-Herder Str. 10, 79104 Freiburg i. B.,
Germany

Abstract. Emulation as long-term preservation strategy depends on usable, reliable and sustainable emulators to preserve old or obsolete original digital environments. The most common approach to achieve this is to preserve the hardware layer of the different computer architectures. The open source project QEMU is in many ways a digital preservation tool that the community of memory institutions depends on.

The requirements for emulation in preservation differ from those that exist for most virtualization tasks. QEMU has to bridge a widening gap between past and current digital environments by keeping the emulated hardware constant and reliable over time. To ensure reliability special test suites should be run on updated versions of QEMU. A number of software and unit tests could be run totally automated using the QEMU VNC and monitor interfaces on prepared system images of all preservation relevant original environments.

QEMU as a cooperative endeavor should extend its user base into the digital preservation domain. It could bring (financial) support and thus foster more sustainability.

Emulators in Digital Preservation Emulation is an extremely versatile and durable solution for retaining access to any kind of digital content. For dynamic and interactive digital objects like educational software, research applications or computer games, emulation is actually the last remaining option, as these digital objects usually cannot be migrated [9, 10] to a more current format.

When compared to the emulation of applications, emulating hardware is a much more efficient and effective approach in terms of the number of emulators required. For instance an application emulator like Scumm-VM¹ supports a small range of specific older computer games in contrast to virtually all objects for X86 systems supported by QEMU. Nevertheless, the longevity of emulation solutions in digital preservation is major concern: While X86 emulators and virtualization tools have been available for more than 10 years they have not yet proven their long-term sustainability or availability. The future of software development in the virtualization domain is thus worryingly difficult to predict.

While the virtualization and emulation community is more interested in system behavior simulation and provision of test environments for modern/current

¹ Script Creation Utility for Maniac Mansion Virtual Machine, <http://www.scummvm.org> [1/8/2011]

operating systems, the digital preservation community depends on long-term availability of old hardware in virtualization/emulation software. This hardware has to be an exact replicate and be correct and accountable in regards to its code-base stability.

Emulation does not avoid migration, but moves it to a more abstract level. As computer development advances, emulators as mostly platform dependent applications also need to be adapted. The major challenge with this state of affairs is in updating the "outer" software shell of the emulator application without changing any inner components at all. This has been accomplished very well in the case of MESS² for "dead" architectures like the old Apple Macintosh or home computers of the 1980s and early 1990s. In contrast to this commercially available tools such as the VMware X86 virtualization suite have mostly ignored the long-term preservation domain. Such commercially available tools have changed their emulated hardware configurations significantly. In the course of this they have deprecated old operating systems like Windows 3.X from Version 4.X on. Additionally, the container formats of the virtual hard disk have been regularly altered, rendering current *Workstation* versions unable to access containers of earlier 3.X versions [10].

QEMU as a Digital Preservation Tool Emulation in digital preservation is seldom used standalone in direct user interaction but often part of a whole workflow integrating other tasks like object type determination or verification. As many workflows imply repeated tasks executed on several hundred up to million objects of similar type they need to be able to be run unattended. There is potential for completely automated migration workflows deploying original tools and software environments to generate, for example, PDF output from Ami Pro document input. Thus preservation frameworks require predetermined and reliable ways to interface to emulators [6].

From the feature set point of view, QEMU is "digital preservation aware" as it implements the different interfaces which could be used for automated migration-by-emulation services as outlined before. Beside the traditional direct GUI in- and output a VNC interface is available which offers an appropriate abstract layer to automatically interface to display content and provide automatic keyboard and mouse input replacing the human user [7].

Another important feature in QEMU is the "monitor" interface implementing an API to emulator controls like power and reset buttons or on removable devices. It offers a channel to send commands during emulator operation for e.g. mounting removable devices, sending special keystrokes like CTRL-ALT-DEL and allows suspending and shutting down of the emulated environment.

A X86 emulator has to consider a range of additional challenges beside reproducing the deprecated hardware for actual computer architectures. As long as the number of objects to be processed is manageable, or just a few individual

² The Multi Emulator Super System is a source-available project which documents the hardware for a wide variety of (mostly vintage) computers, video game consoles, and calculators through software emulation, <http://www.mess.org> [1/8/2011]

users interact with emulation environments, the required computing power and wall clock time consumed for those processes is minor. If large scale preservation systems are to be run, and preservation planning tools like PLATO [1] are to be used to evaluate runtimes and give reasonable cost estimates, more information is needed [3] such as the memory consumption or CPU load.

For large scale migration scenarios requirements like predictability and accountability of actions taken play an important role. The time required for an action (calculated e.g. in wall time or CPU cycles consumed) should be predictable to give archive operators a base to calculate costs and the amount of time consumed for a particular preservation actions [3]. Additionally the monitor interface should be more usable to operate complete guest sessions by sending sequences of keystrokes or mouse events.

Preserving the Emulated Hardware Digital preservation depends on stable virtual machine hardware, because of this the emulator plays a crucial role. It bridges the widening gap between the digital past and today's current environments. As an operating system like DRDOS 6.0 or Windows 3.1 is not maintained any more, there will be no support for newer hardware like actual gigabit network adaptors or high resolution 3D graphic cards. Thus, the reproduction of a certain 20 year old software environment depends on emulators like QEMU providing exactly the hardware configuration in use in those times. In order to be able to deploy such an environment in 20 years from now the emulated hardware must remain exactly the same across time e.g. the virtual 386 ISA bus machine of the early nineties being equipped with a NE2000 network adaptor, 16 bit Soundblaster sound card and simple VGA should remain unchanged.

Until now the version history of QEMU was pretty good maintaining the once introduced hardware components but rather volatile regarding the reliable support of the older operating systems like Windows 3.X, 95 or 98. For some versions the floppy support did not work correctly, in others Windows 9X could not be started at all and in newer QEMUs the mouse does not operate as expected e.g. caused by the changeover from the Bochs to the SeaBIOS. A major issue is the *virtual hardware available driver mismatch*: At the moment the Cirrus Logic VGA is unusable in Windows 3.11 and the PIIX3 IDE is not properly recognized by the Windows 95 or official Intel drivers. Such changes and flaws are a major problem for digital preservation as the long-term reliability and stability of emulators and the hardware they emulate is crucial. As long as there are no abstract emulator-specific video or IDE drivers provided for outdated operating systems a complete reproduction of original environments depends on the proper hardware emulation.

Automated Emulator Testing Software testing is often a very elaborate task involving a lot of manual and repetitive labour. As the part of the QEMU community interested in the support of older operating systems is comparatively small we suggest other means to solve the problem. The tests should be run

mostly automatically by running a large number of different original environments and checking on certain components, like:

- Booting off different media like hard disk, floppy, optical media images and network.
- Checking on screen resolution, network availability, proper keyboard translation and mouse support.
- Booting the operating systems which are required to work properly to set up preservation workflows. Check that they reach a defined level.
- Starting applications and execute certain actions.

Currently at the University of Freiburg, a test framework for QEMU and other VNC enabled emulators has been created using the results of research into the automation of interactive sessions [7, 5]. The idea is to begin by interactively recording a particular workflow using a modified VNCplay,³ such as the boot process of an old operating system from hard disk or installation medium. Such a recording can serve as base for the generation of a machine script for a later completely automated repetition. The recordings are comprised of an ordered list of interactive events like mouse movements or key strokes which are passed on to the emulated environment through the VNC interface. Each of these events is linked with an expected outcome which can be observed as a state of the emulated environment. Before this effect is seen, the next event cannot occur. The link with expected outcome of a previous event is necessary since certain actions will take different amounts of time to run.

The Black-box-Testing-principle [4] was selected to perform the emulator-testing because all the tests are realized without knowledge of the emulator's code or internal structure. Thus the selected emulator will be viewed as a Black-Box with only the inputs and outputs are taken into account. Through the VNC interface events (input data) are sending to the emulator and which produces one or more outputs. Like in a ordinary software test these output is compared to the expected outputs. To ensure reliability all the same tests cases are performed on every updated emulator version. In this way it is easier to check if the previous emulated systems, applications, and hardware in the new emulator version are running properly. This principle is knowing in the software engineering as regression test [4].

Thus the framework in development defines and executes usage scenarios to verify the behavior of the emulator on a predefined checklist. This list should be compiled in a way to tell as precisely as possible which component failed if a certain task does not perform as expected. It is to be sorted in such a way as to start with small tasks such as booting a simple DOS from floppy disk, ISO and hard disk before starting more complex operating systems like Windows 9X or running complex games in high screen resolutions on top of them. All results are compared to expected, pre-recorded outcomes. The emulator testing framework is programmed in Java to be included into the existing toolsets dealing e.g. with software management for the original environments. In a further step it will be

³ See <http://suif.stanford.edu/vncplay> [1/8/2011]

extended to verify other VNC enabled X86 emulators like the newest version of Dioscuri [8, 2] with the same set of tests.

Conclusion and Future Development In many aspects QEMU seems as a optimal digital preservation tool, implementing most of the APIs required for (automated) digital preservation actions missing only more advanced monitoring features. Being Open Source and possessing a large supporting community it has a good chance of long-term sustainability in contrast to commercial solutions. Nevertheless it lacks the reliability required for durable preservation solutions yet [11]. The original hardware needs to be supplied in a way that the original operating system drivers could directly be used as long as no other options (for example, high resolution screen output) are made available. Here the different user communities should cooperate more tightly e.g. to discuss and develop automated quality assurance for well defined test sets of standard (deprecated) environments. Another important issue is the long-term code serviceability and the definition of crucial hardware components to support, e.g. for migrating deprecated machines directly to the emulator.

To support the QEMU project the digital preservation community should extend the model used to finance the Dioscuri development. A joint effort of the national memory institutions could bring in relevant contribution without overstraining the resources of single partners [12].

References

1. Christoph Becker, Hannes Kulovits, Mark Guttenbrunner, Stephan Strodl, Andreas Rauber, and Hans Hofman. Systematic planning for digital preservation: evaluating potential strategies and building preservation plans. *International Journal on Digital Libraries*, 10:133–157, 2009.
2. Evgeni Genev. Vnc-interface for java x86-emulator dioscuri. October 2010.
3. Brian Hole, Li Lin, Patrick McCann, and Paul Wheatley. Life: A predictive costing tool for digital collections. In Andreas Rauber, Max Kaiser, Rebecca Guenther, and Panos Constantopoulos, editors, *7th International Conference on Preservation of Digital Objects (iPRES2010) September 19 - 24, 2010, Vienna, Austria*, volume 262, pages 359–364. Austrian Computer Society, 2010.
4. ISTQB. International software testing qualification board (istqb). Online, <http://www.istqb.org>, 2010.
5. Klaus Rechert, Dirk von Suchodoletz, and Randolph Welte. Emulation based services in digital preservation. In *JCDL '10: Proceedings of the 10th annual joint conference on Digital libraries*, pages 365–368, New York, NY, USA, 2010. ACM.
6. Klaus Rechert, Dirk von Suchodoletz, Randolph Welte, Felix Ruzzoli, and Isgandar Valizada. Reliable preservation of interactive environments and workflows. In Mounia Lalmas, Joemon M. Jose, Andreas Rauber, Fabrizio Sebastiani, and Ingo Frommholz, editors, *Research and Advanced Technology for Digital Libraries, 14th European Conference, ECDL 2010, Glasgow, UK, September 6-10, 2010. Proceedings*, volume 6273 of *Lecture Notes in Computer Science*, pages 494–497. Springer, 2010.

7. Klaus Rechert, Dirk von Suchodoletz, Randolph Welte, Maurice van den Dobbelen, Bill Roberts, Jeffrey van der Hoeven, and Jasper Schroder. Novel workflows for abstract handling of complex interaction processes in digital preservation. In *Proceedings of the Sixth International Conference on Preservation of Digital Objects (iPRES09)*, 2009.
8. Jeffrey van der Hoeven. Dioscuri: emulator for digital preservation. *D-Lib Magazine*, 13(11/12), 2007.
9. Jeffrey van der Hoeven and Dirk von Suchodoletz. Emulation: From digital artefact to remotely rendered environments. In *Proceedings of the Fifth International Conference on Preservation of Digital Objects (iPRES08)*, pages 93–98, The British Library, St. Pancras, London, 2008. The British Library.
10. Dirk von Suchodoletz. Requirements for emulation as a long-term preservation strategy. Online, <http://eprints.rclis.org/18984/>, July 2009.
11. Dirk von Suchodoletz. A Future Emulation and Automation Research Agenda. In Jean-Pierre Chanod, Milena Dobрева, Andreas Rauber, and Seamus Ross, editors, *Automation in Digital Preservation*, number 10291 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2010. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, Germany.
12. Dirk von Suchodoletz, Klaus Rechert, Jeffrey van der Hoeven, and Jasper Schroder. Seven steps for reliable emulation strategies – solved problems and open issues. In Andreas Rauber, Max Kaiser, Rebecca Guenther, and Panos Constantopoulos, editors, *7th International Conference on Preservation of Digital Objects (iPRES2010) September 19 - 24, 2010, Vienna, Austria*, volume 262, pages 373–381. Austrian Computer Society, 2010.