Preserving Personal Digital Artifacts with Foundation

Sean Rhea
Intel Research Berkeley

Alex Pesterev* Russ Cox
MIT CSAIL

We are "living in the midst of digital Dark Ages"

Users increasingly store their most valuable data only in digital form

Proliferation of online storage providers makes reliable storage feasible

But will we be able to *interpret* that data a generation from now?

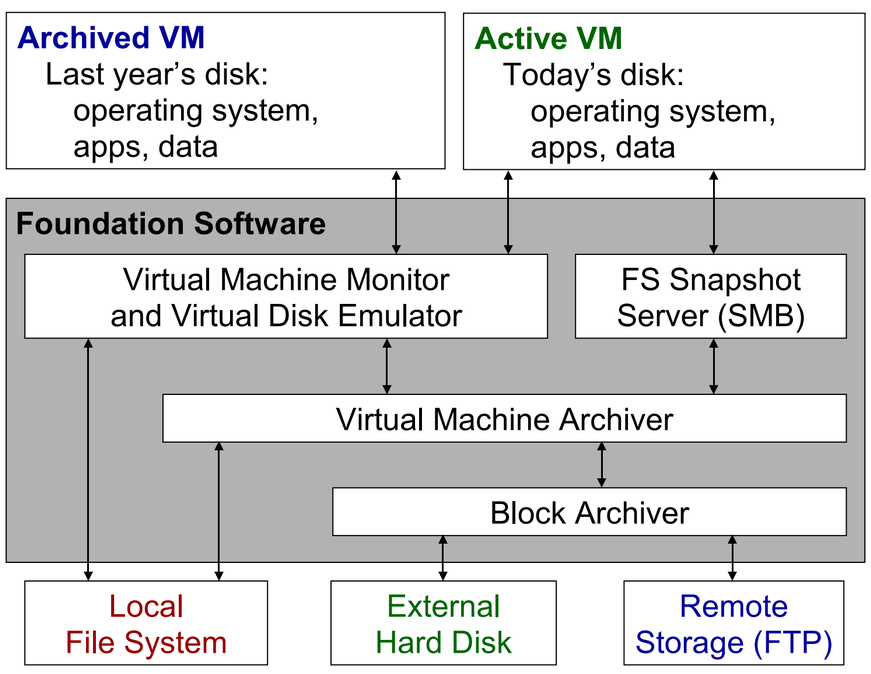
Motivating Examples

- Viewing an old PowerPoint presentation
  - Newer versions of PowerPoint mangle formatting
  - Does the presentation use non-standard fonts/codecs?
  - Does the original PowerPoint run on today's OS?
- Not just a Microsoft problem: consider a web page
  - Even current IE and Firefox don't agree on formatting
  - All kinds of plugins necessary: sound, video, Flash
- Compare to a (physical) scrapbook or 8mm movie
  - Degrades over time, but in a continuous manner
  - Digital content degrades abruptly: may not display at all

Foundation Preserves Personal Data

- Emulates old hardware, not old software
  - Not nearly as much variety in processors as in software
  - Existing emulators for Amiga, Nintendo, PDP-11, etc.
- Makes daily backups of *entire* software stack
  - Archives users' applications, OS, and configuration state
  - Boots stack in emulator to recover obscure file types
- Confines daily environment to a virtual machine (VM)
  - Ensures archived stack boots on emulated hardware
  - Today's VMM is the template for tomorrow's emulator
  - VMM snapshot facility assists in quick, consistent archival

System Architecture



1. Under normal operation, users' chosen operating system and applications all run in a virtual machine
2. Foundation takes nightly snapshots of all virtual disk blocks and archives them on an external hard disk and remote server, coalescing duplicates
3. To recover an old version of a file, a user looks in the file system snapshot server, mounted on /snapshot
4. If today's apps can no longer interpret an old file, boot older snapshot of operating system and applications to interpret it

Block Archiver

- Based on Venti [Quinlan & Dorward '02]
  - Identifies disk blocks in archive by SHA-1 hashes
  - Stores each block only once; can retain all nightly snapshots indefinitely
  - But removes spatial locality; read and write throughput seek-limited
- Foundation optimizes for common cases
  - Writing new blocks, reading and rewriting blocks in same order originally written
  - In 512-day trace:

    write tput 19 MB/s vs 800 kB/s in Venti

    read tput 13 MB/s vs 1.3 MB/s in Venti

Integration with VMM

- VMware snapshots give consistent disk images
  - Taking snapshot marks disk file read-only, sends new writes to a differences file
  - Provides consistent disk image to archival process
  - Allows user to continue working during archival
- Reuse differences file to speed next night's archival
  - Quickly identify what blocks have changed in last day
- Loopback file server provides old disk images for VMware to boot

*Work done in part while at Intel Research Berkeley