NORTHWESTERN
UNIVERSITY

The Collective: A Cache-Based System Management Architecture

Ramesh Chandra, Nickolai Zeldovich,
Constantine Sapuntzakis, Monica S. Lam

Instructor: Fabian Bustamante
Presented by: Mario Sanchez

The focus of this paper

- The Collective: a cache-based system management model that combines the advantages of centralized management while taking advantage of inexpensive PCs.
- Prototype developed and deployed for almost a year

Personal Desktop Model

- Managing multiple desktops a Challenge: (a) Installing software, (b) troubleshooting errors, (c) performing upgrades, (d) performing backups, (e) recovering from viruses and spyware.

Move to the simplicity of centralized mainframe model!

Thin-client Computing

- Computation is performed on computers centralized in the data center.
- user's desk: special-purpose remote display terminal or a general-purpose personal computer returning remote display software.
- Drawbacks:
  - Higher hardware costs
  - Performance problems

The Collective

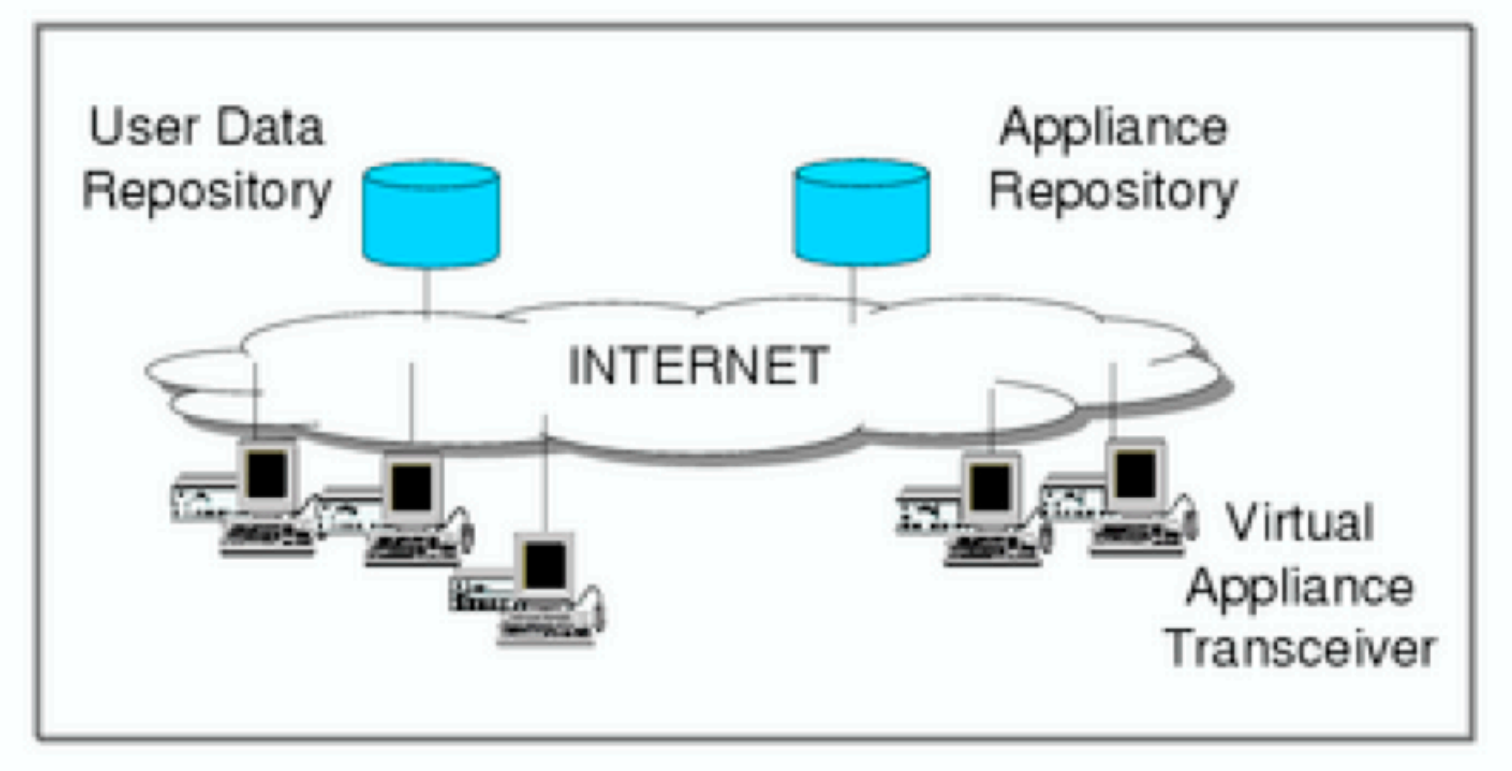
- System that delivers managed desktops to personal computer users.
- Sys Admin are responsible for the creation and maintenance of the desktop environments.
- Virtual Appliances include the operating system and all installed applications: VMware GSX Server
- Virtual Appliance Transceiver client software.

Cache-based system Model

Separate state of computer into two parts:

- System State:
  - Operating system and all applications.
  - Referred to as Virtual Appliances
- User State:
  - User profile
  - Preferences
  - Data files

The Collective Architecture



Store system and user states in separate network-accessible repositories

Virtual Appliance Transceiver

- VAT: caches and runs the latest copies of appliances locally and backs up changes to user data to a network repository.
- Suitable for managing computers on
  - LAN
  - WAN with broadband
  - Computers occasionally disconnected from the network like laptops.

Characteristics of VAT

- The VAT is a separate layer in the software stack devoted to management:
  - The VAT is protected from appliances it manages by the VMM, increasing security and reliability.
  - The VAT automatically updates itself: requires no management on the part of the user.
  - Cache design is kept simple with use of versioning scheme: every data item is referred to by unique name.

System Overview

1. Appliances
2. Repositories
3. Subscriptions
4. User Interface
5. Management Functions
6. Optimization for different Network connectivities

1. Appliances

Encapsulate a computer state into a virtual machine. Contain:

- System disks:
  - Created by administrator;
  - Hold the Operating System and Applications.
- User disks:
  - Hold persistent data private to user (files and settings)

1. Appliances

- Ephemeral disks:
  - Hold user data that is not backed up (browser cache, temporary files).
  - No reason to incur on additional traffic to back up such data
- Memory image:
  - Hold the state of a suspended Appliance

2. Repositories

- Each Appliance has a repository.
- Updates to an appliance get published as a new version in that repository.
- VATs automatically find and retrieve the latest version of the appliance in that repository
- Repository versions are immutable.
- Copy-on-write (COW) disks are used to express the differences between versions (save space, optimize data transfer).

3. Subscriptions

- User Account: to perform access control.
- Collective Profile: list of appliances the user has access to (located on network storage).
- Subscription:
  - Subscriptions are created in the profile.
  - Stores user state associated with each appliance.
  - Contains a repository for the user's disks associated with an appliance (to version user disks).

User Account -> User's Collective Profile -> Subscription

3. Subscriptions

- Some state associated with the subscription is stored only on storage local to the VAT.
- A COW of sys disk is created locally on the VAT.
- Suspended appliances write memory image to the VAT storage.

Suspended appliances cannot be migrated between VATs

4. User Interface

- Log-in window
- List of Appliances and their status
- Menu of options: start, stop, suspend, reset, delete, user disk undo, publish
- Indicates the amount of data that remains to be backed up from the local storage device.

5. Management Functions

- System Updates: Admin prepares new version of appliance and deposits it in the repository. User gets a copy after next reboot.
- Machine Lockdown: changes made to the system disk are stored locally and discarded when the appliance is shut down.
- Hardware Management: PCs running the VAT are interchangeable (simple).

5. Management Functions

- Backup:
  - COW snapshots of the user disk are created on reboot as well as periodically.
  - One backup for each version to the user repository.
  - Users can rollback to previous versions.
  - Only one VAT can upload snapshots at a time (to the repository).
  - Snapshots can accumulate at the client for later upload.

6. Optimizations for different Network Connectivities

- VAT Includes a large on-disk cache (gigabytes).
- Cache keeps local copies of the system and uses disk blocks.
- The system pre-fetches data on demand and into cache to ensure good performance.
- Appliance writes go to the VAT local disk

(a) LAN, (b) WAN, (c) disconnected Laptops, (d) portable VATs, (e) Remote display

VAT Design

- Modified KNOPIXX (Live CD version of Linux)
- VMware GSX server
- Uses SSH to authenticate to the profile server (sets up key pair to access storage afterwards)
- Managed as an appliance
- Automatically updates from repository
- Up to three versions stored locally

VAT Design

- 350MB uncompressed, 160MB compressed (cloop tool), Rsync reduces size to 10MB (only diff blocks are updated)
- Simple layout: each repository is a directory, each version is a subdirectory.
- Once published, files can’t be changed
- NFS for reads and writes from network storage

VAT Design

- Two different caches: a small object cache for small data and meta-data and a COW cache for COW disk blocks.
- Small objects (like list of appliances) are replicated entirely. User data snapshots are also stored completely before being uploaded to the server.
- COW is used to cache immutable versions of disks from repositories and are accessed block by block.

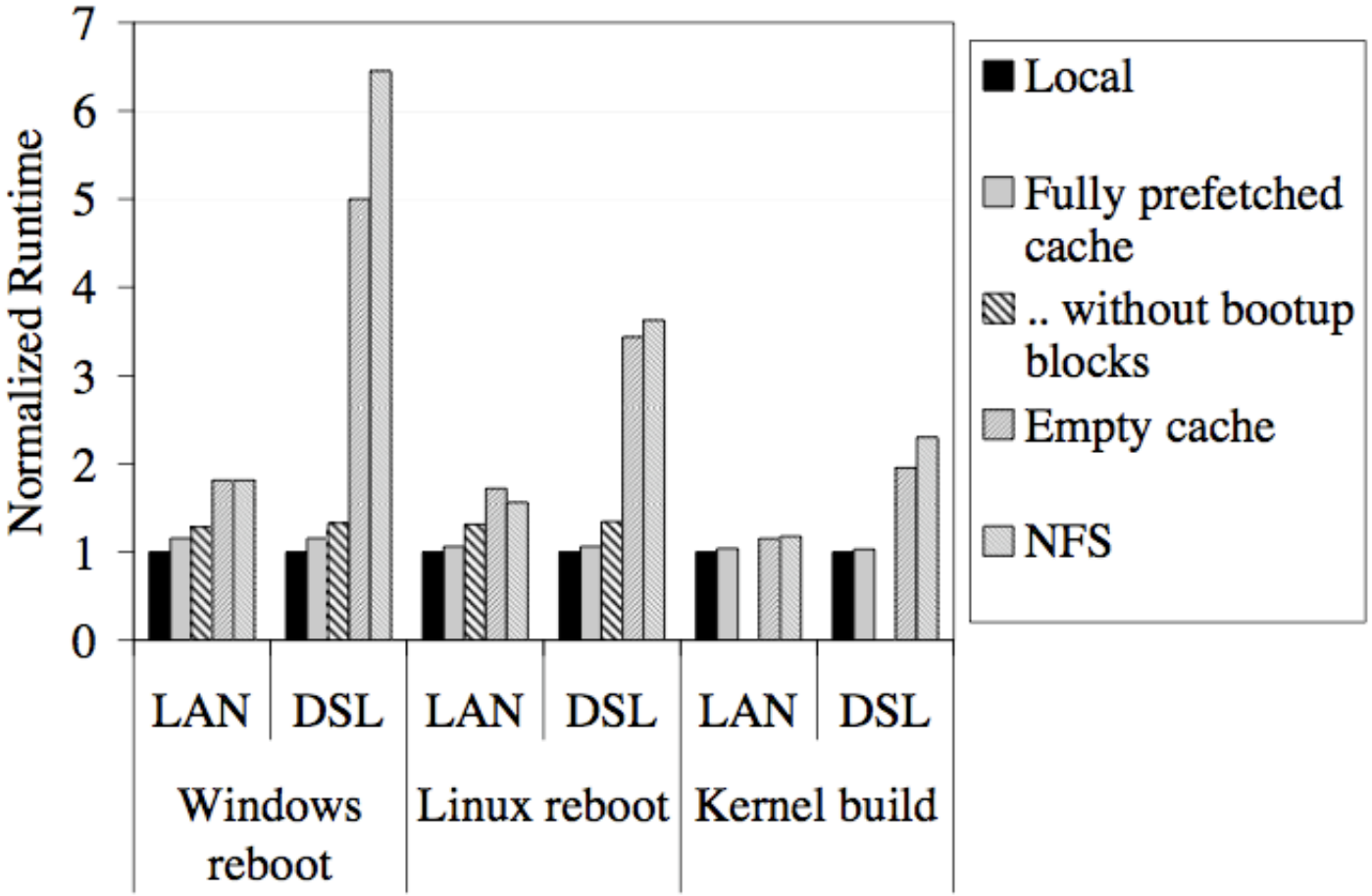
VAT Design

- Each 512-byte sector is protected by an MD5 hash; on fail the data is treated as not cached.
- Prefetcher process: fetches useful appliance blocks in the background.
- An appliance can be used in offline mode by prefetching it completely in cache.
- The prefetcher rate-limits itself.
- The prefetcher hides network latency from the appliance.

Evaluation

- Run the same virtual machine workloads on unmodified version of Vmware's GSX server and compare overhead of the Collective caching system.
- Two sets of experiments:
  - Local: the VM is copied to local disk and executes w/o demand-fetching.
  - NFS: the VM is stored on an NFS file server and is demand-feteched by VMWare

Runtime on different Cache configurations



LAN: 100 Mbps, DSL: 1.5 Mbps / 3.84 Kbps, 40 msec RTT

Effects of disk performance

	VAT startup	Windows reboot	Linux reboot	Kernel build
Lexar 1GB Flash Drive	53	129	42	455
IBM 4GB Microdrive	65	158	53	523
Hitachi 40GB 1.8” Drive	61	84	43	457
Fujitsu 60GB 2.5” Drive	52	65	40	446

* all time are in seconds

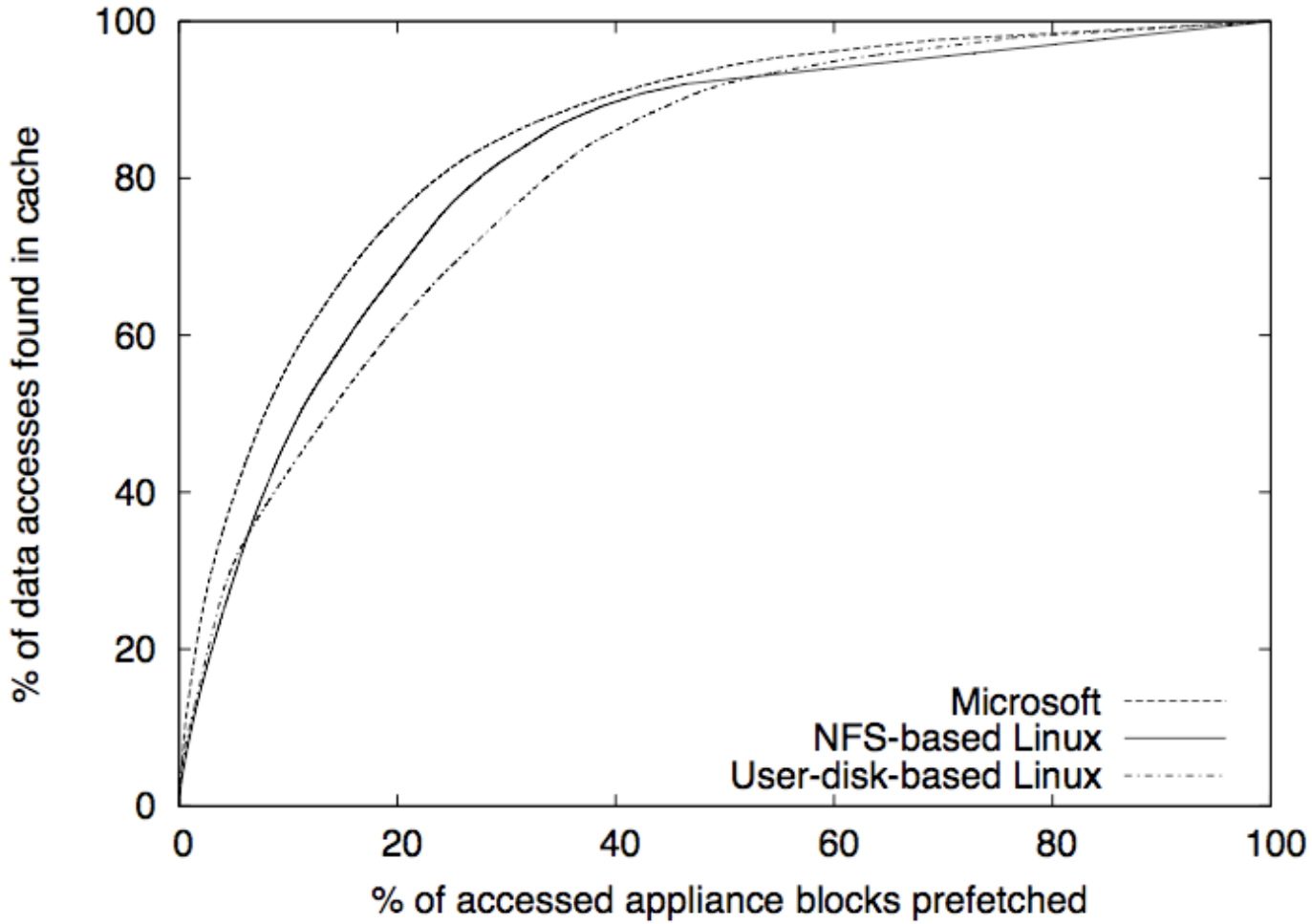
To emphasize disk performance, the VMs are fully prefetched into the cache, and the machines are connected over a 100 Mbps LAN.

Effectiveness of Prefetching

Appliance	Accessible Size	Accesses in Traces	Unique data Accessed
Windows XP	8.9 GB	31.1 GB	2.4 GB
NFS Linux	3.4 GB	6.8 GB	1.0 GB
User-disk Linux	6 GB	5.9 GB	0.5 GB

- Measure the access profile of appliance blocks.
- Prefetching based on the popularity of blocks.
- 15 days of traces from 9 users using 3 different appliances.
- Users only access 10% - 30% of the accessible data in the appliances.

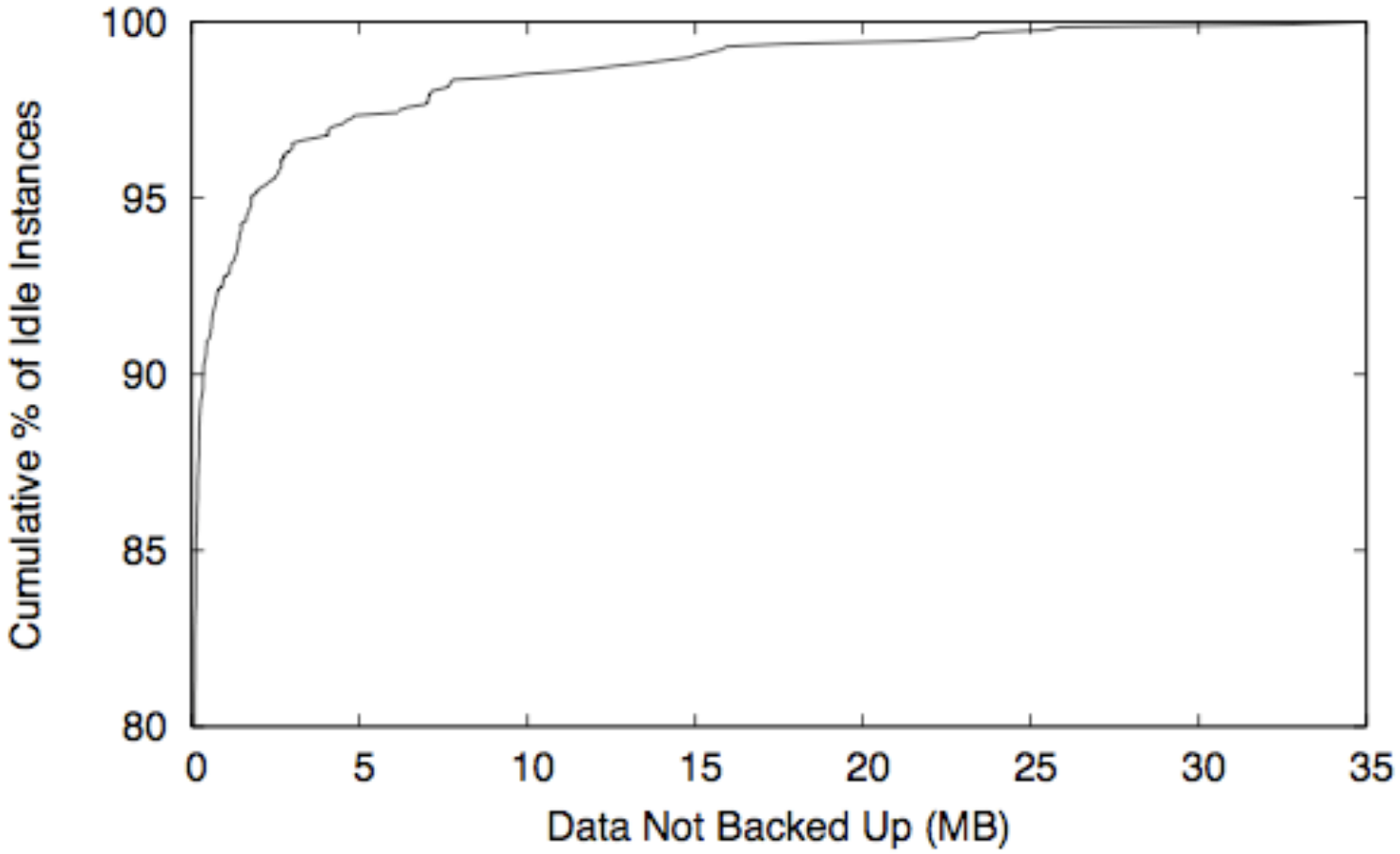
Block access profile



Takeaways

- A small amount of data accounts for most of the accesses
- Prefetching can greatly improve the responsiveness of an interactive workload (reducing cache misses).
- Populatiry of accessed appliance data is a good heuristic for prefetching

Backup of Data



Drawbacks?

- The model requires the user to subscribe to an entire appliance; big or small software upgrades use the same mechanism.
- Trade-off user's ability to customize their environment in return for uniformity, scalability, better security and lower cost of management.
- Performance not satisfactory for I/O intensive applications such as software builds and graphic intensive applications.

Comments

- What happens to users that need customized application settings? Users are not allowed to install applications other than those included in the appliance.
- Overhead comparison between the Collective and the actual non-virtualized system?
- Though the idea is very practical, the system doesn’t present any real contributions to the scientific community.

QUESTIONS?

Thank You