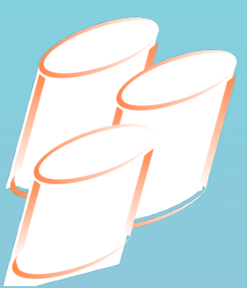
The Effectiveness of Deduplication on Virtual Machine Disk Images

Keren Jin & Ethan L. Miller
Storage Systems Research Center
University of California, Santa Cruz



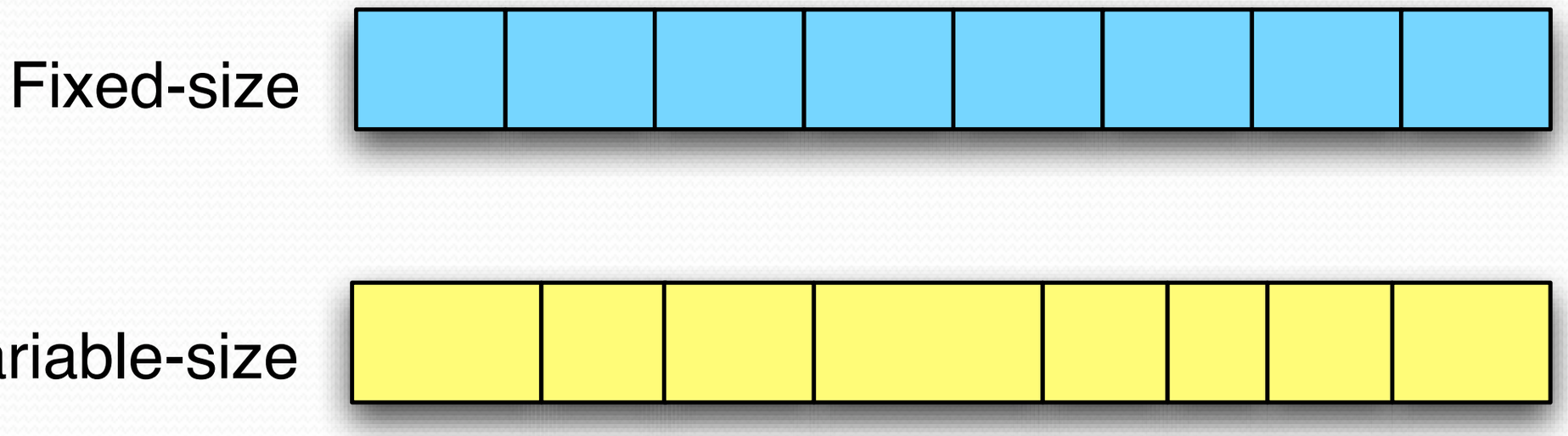
Motivation

- Virtualization is widely deployed for multiplexing services
  - Increases the demand for storage on a single node
- VM disk images have a lot of pages in common
  - Binaries shared between disk images
  - VM doesn't know the internal structure of the file system on the virtual disk
- Deduplication can find identical chunks in different VM disk images
  - Storage savings can range from 10–80% or more
- Which factors affect deduplication ratios?

Methodology overview

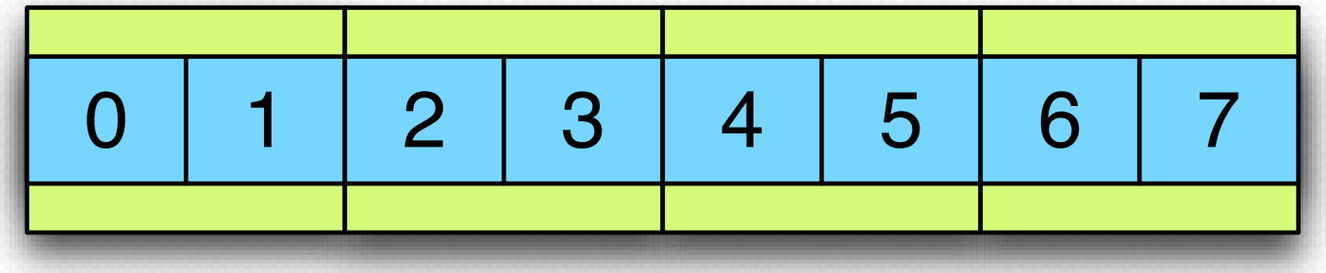
- VMs are downloaded from internet
  - Used freely-available images
  - Wide range of “pre-built” functionality
- VM disk images were chunked
  - Produced chunk lists for each image
- Deduplication experiments run against sets of chunk lists
  - Determined deduplication ratio and amounts of sharing
- Chunk-wise compression: deduplication + zip

Chunking



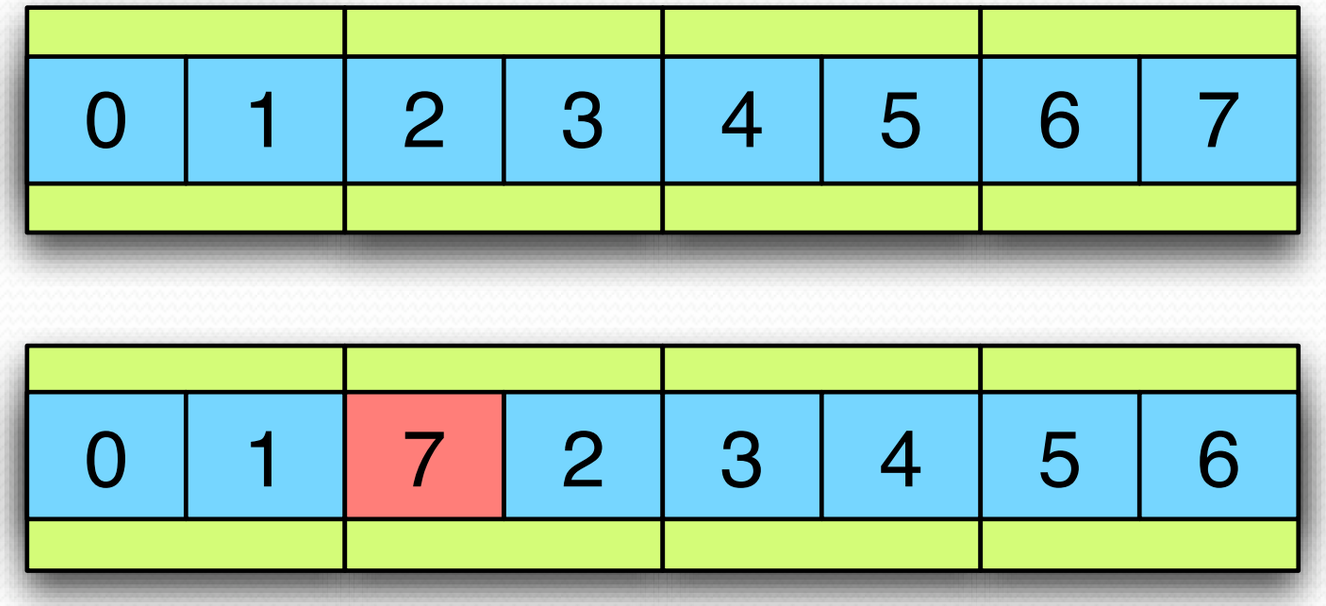
- Break VM disk images into chunks
  - Fixed-size chunking: constant chunk size.
  - Variable-size chunking: adjust Rabin fingerprint parameters to obtain desired size
  - Use secure hash of the chunk content for chunk ID
  - Zero-filled chunks are all identical
- Generate sorted list of chunks (easy to merge)
- A chunk store consists of chunk IDs from a group of VMs

Chunking VM disk images



- Disk image may be split into multiple image files
  - Some images use files ≤ 2GB long
- Fixed-size chunking has boundary-shifting problem
  - Adding a small amount of data may shift content
  - Chunk size is large (4KB): “small amount of data” may be a disk block
- Chunk each image file separately, as each image file has VMM-dependent header

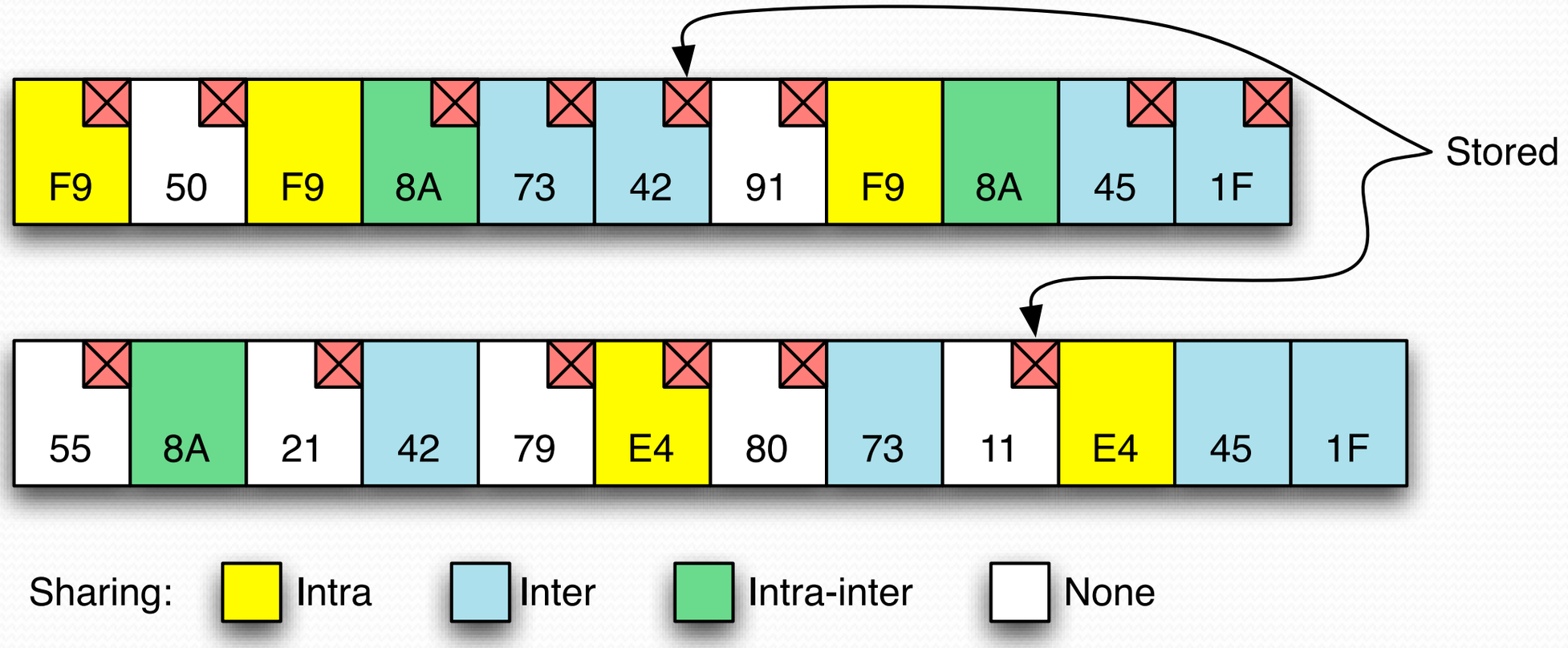
Chunking VM disk images



- Disk image may be split into multiple image files
  - Some images use files ≤ 2GB long
- Fixed-size chunking has boundary-shifting problem
  - Adding a small amount of data may shift content
  - Chunk size is large (4KB): “small amount of data” may be a disk block
- Chunk each image file separately, as each image file has VMM-dependent header

Deduplication categories

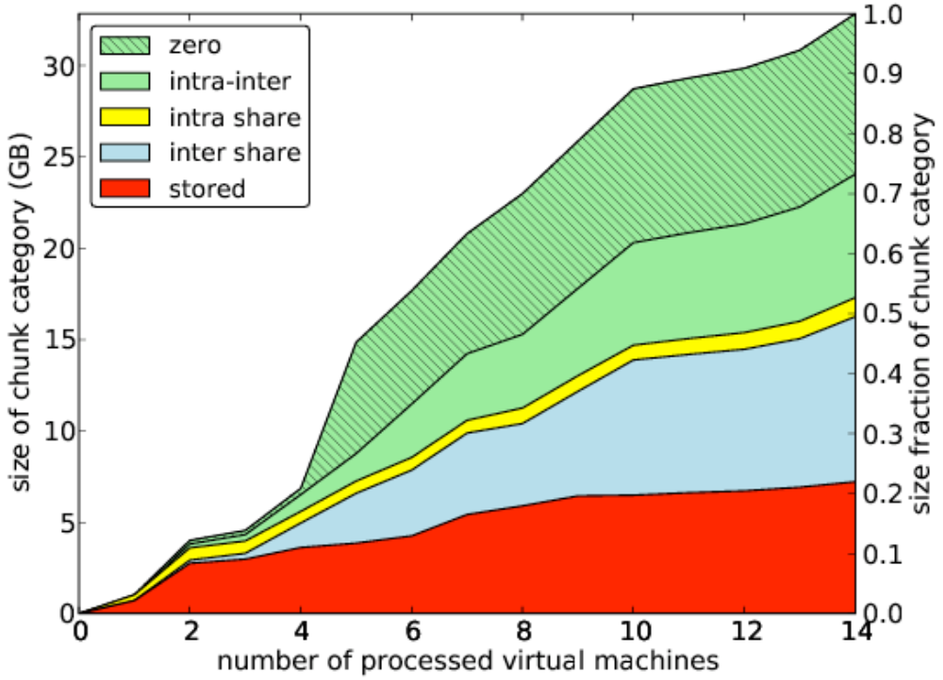
- Find identical chunk IDs in a chunk store
- Deduplication ratio = 1-*stored_bytes*/*original_bytes*
- Chunk categories:
  - Stored 14/23; intra 3/23; inter 4/23; intra-inter 2/23



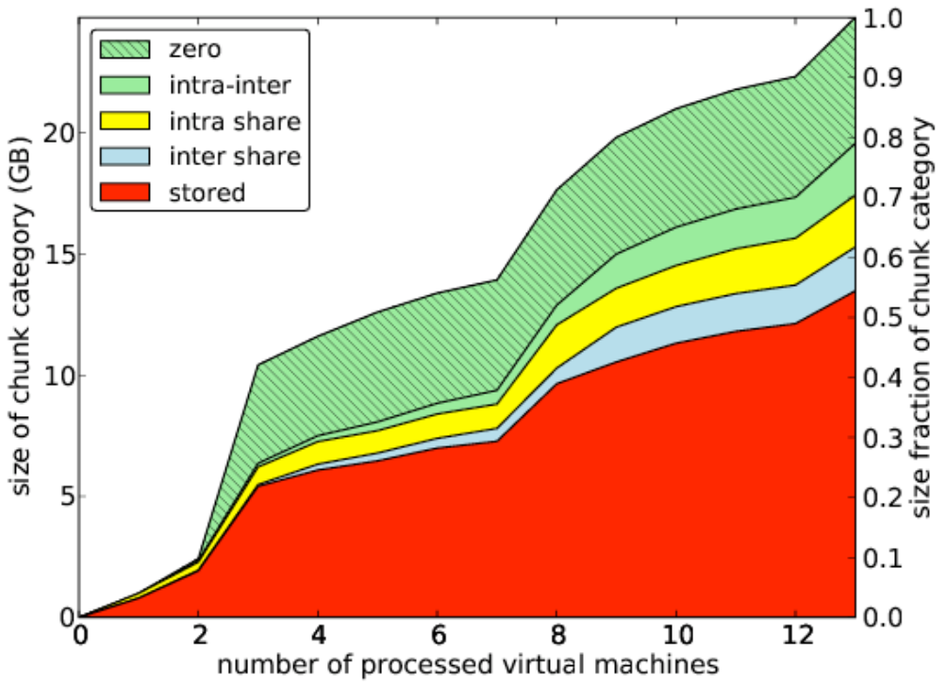
Homogeneous and heterogeneous VMs

- Effect of VM "similarity" on deduplication ratio
  - 14 Ubuntu 8.04 LTS
  - 13 various Unix and Linux instances
  - Variable-sized chunking: 1 KB
- Similar VMs: deduplication is very effective!
  - Slow marginal rate of increase
- Dissimilar VMs: less effective (but still helps!)
  - Marginal rate is higher
- Larger chunk size produces similar ratio and category distributions

Similar VMs

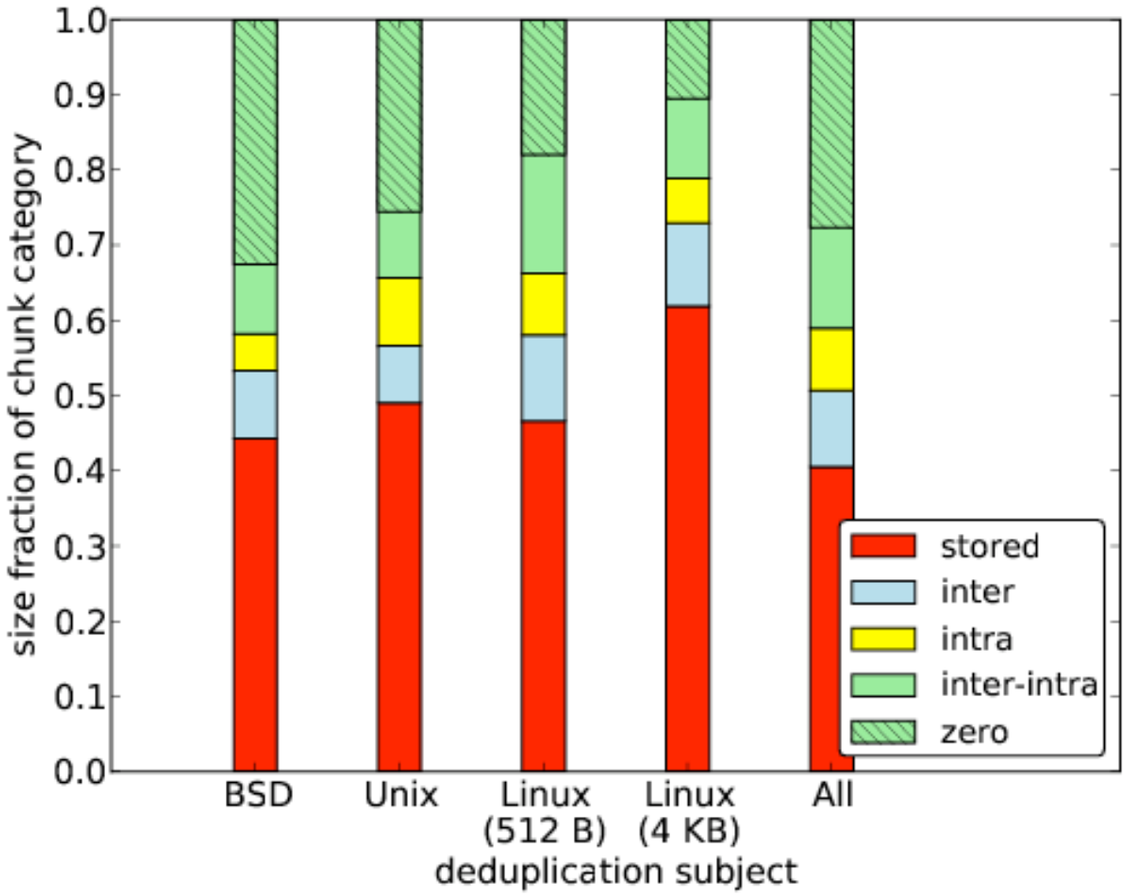


Dissimilar VMs



Chunk categories in different operating systems

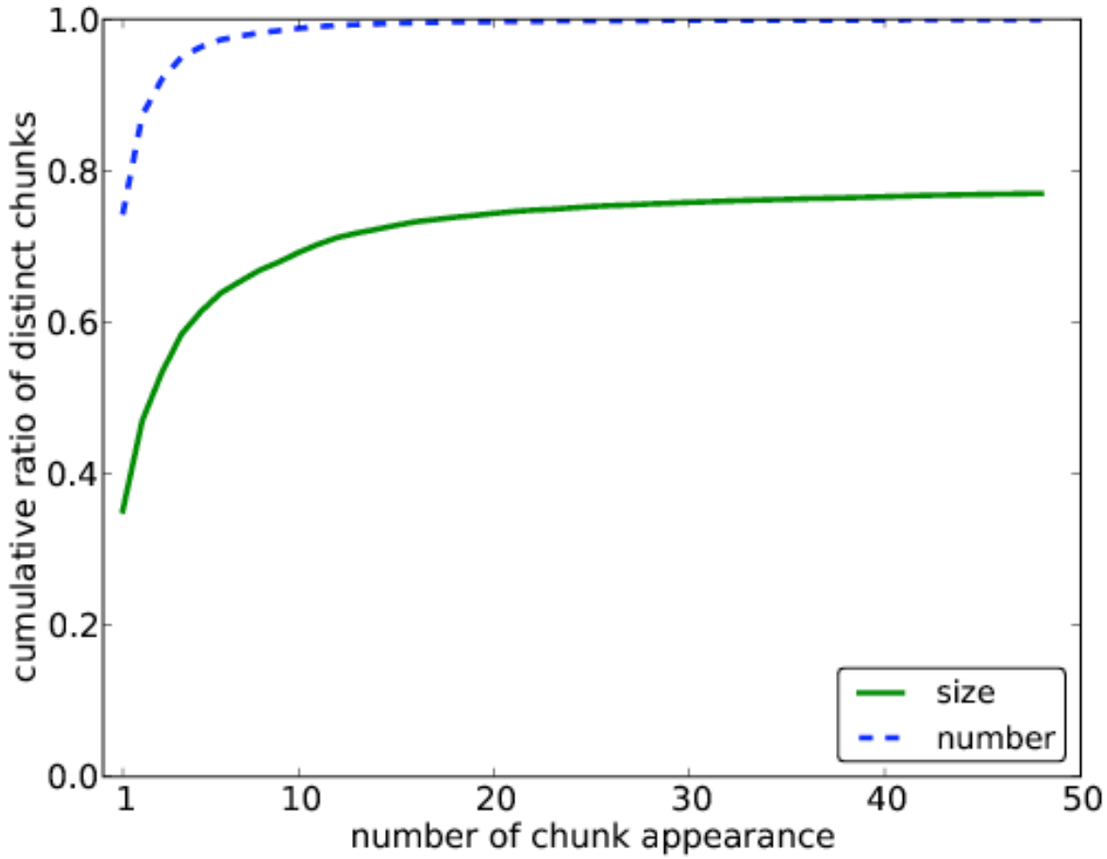
- More intra-inter sharing in Linux than BSD
  - Linux is more homogeneous
- Larger chunk sizes do not impact sharing ratio much
- Many small zero chunks in Linux
  - "Intrinsic" zero chunks due to file system or data files



Operating system groups
Variable-length chunks, avg. 512B

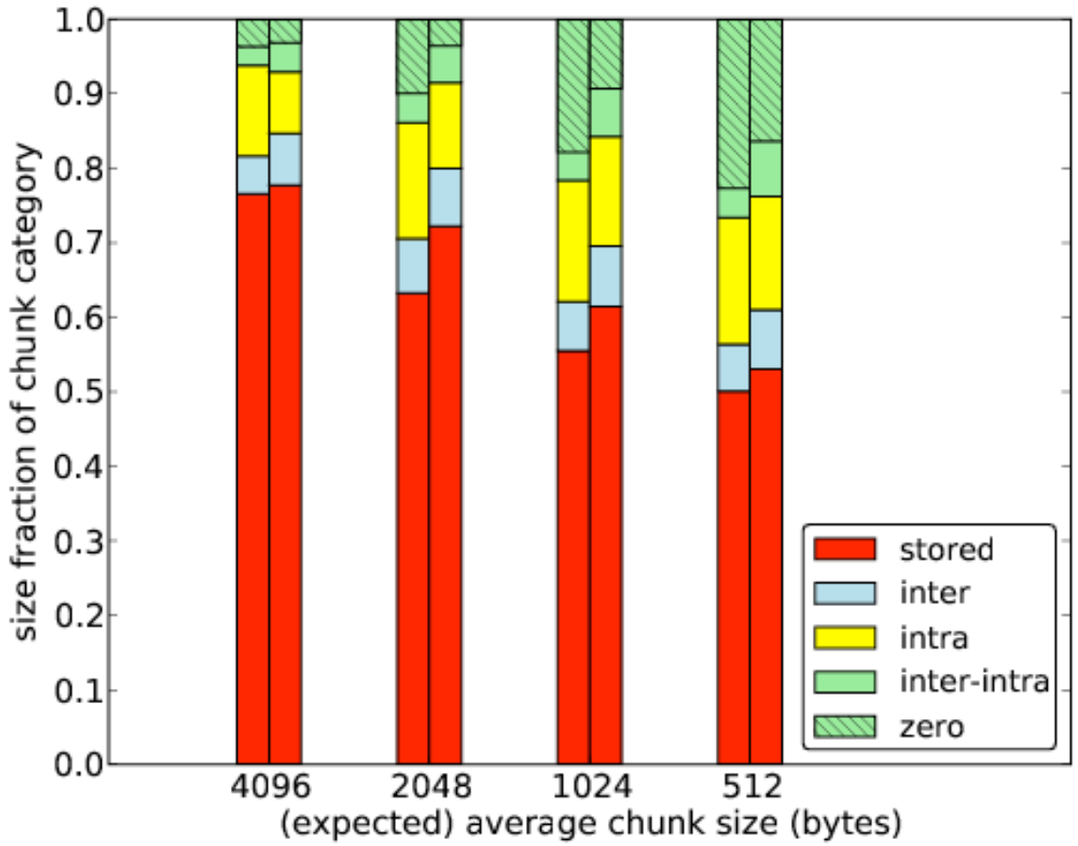
Chunk count distribution

- CDF of chunks by count and total size from Linux chunk store
- Space utilization
  - 70% of chunks are unique
  - 20% of space is zero-filled chunks
- Chunk reuse
  - Most chunks are used only a few times



Effect of chunk size

- Smaller chunk sizes → higher deduplication ratios
  - Less chance of the "avalanche effect" from rearranging chunks
- Fixed size chunking performs well
  - Much easier to implement in an online system
- Small zero chunks: caused by guest OS and apps
  - Empty disk space would cause larger chunks

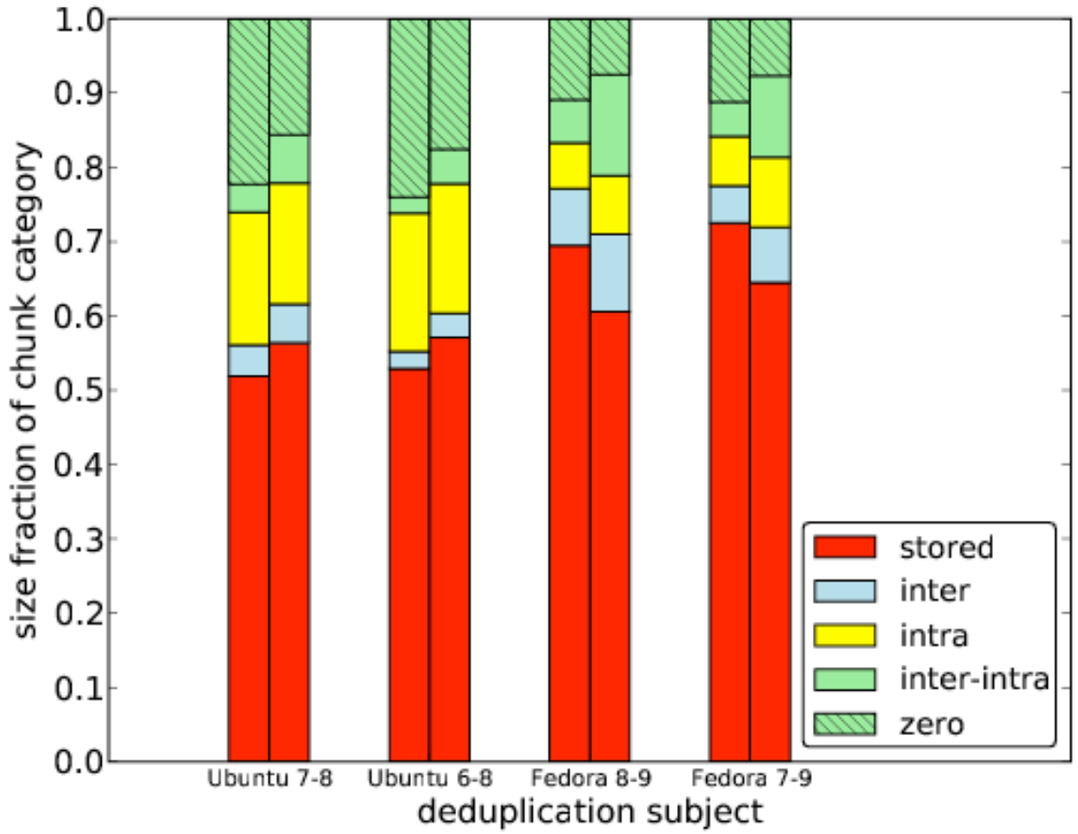


Ubuntu Server:
6.10, 7.04, 7.10 and 8.04.

Fixed and variable size chunking:
512, 1024, 2048, 4096 bytes

Effect of OS version

- OS versions closer together deduplicate better
  - Consecutive releases have higher sharing
- Still a high degree of deduplication even between non-consecutive releases
  - Mature operating systems change little across versions

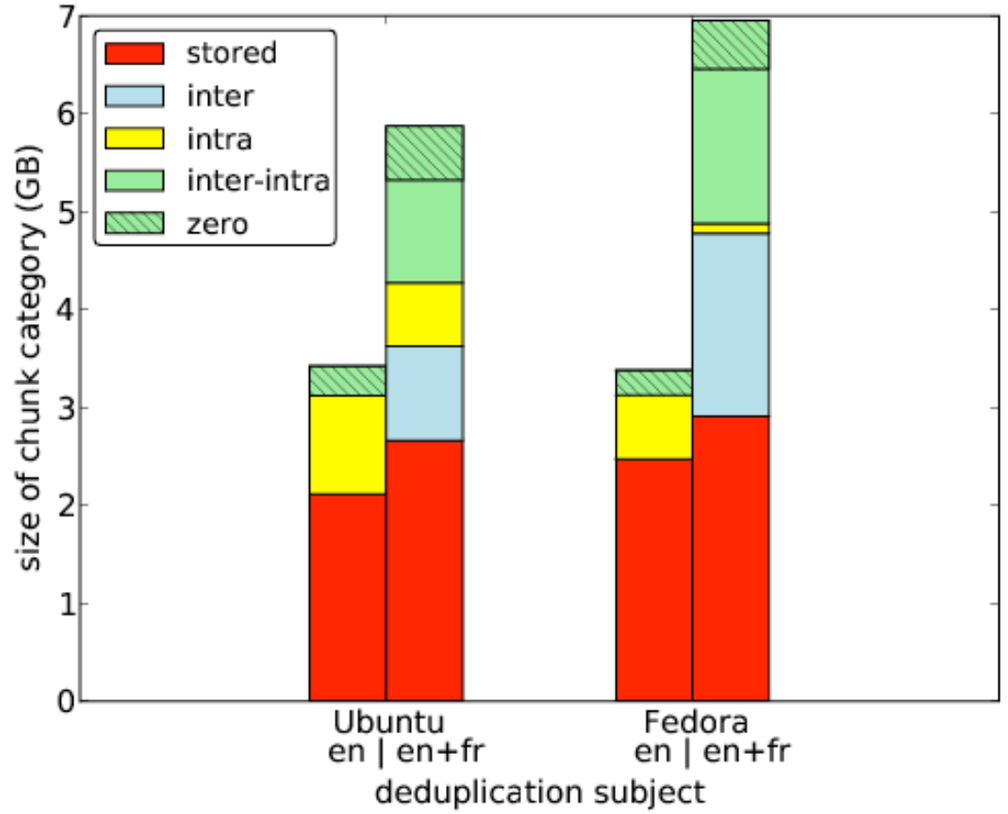


Ubuntu Server and Fedora: various versions

Fixed and variable size chunking: 512 bytes

Effects of locale on deduplication

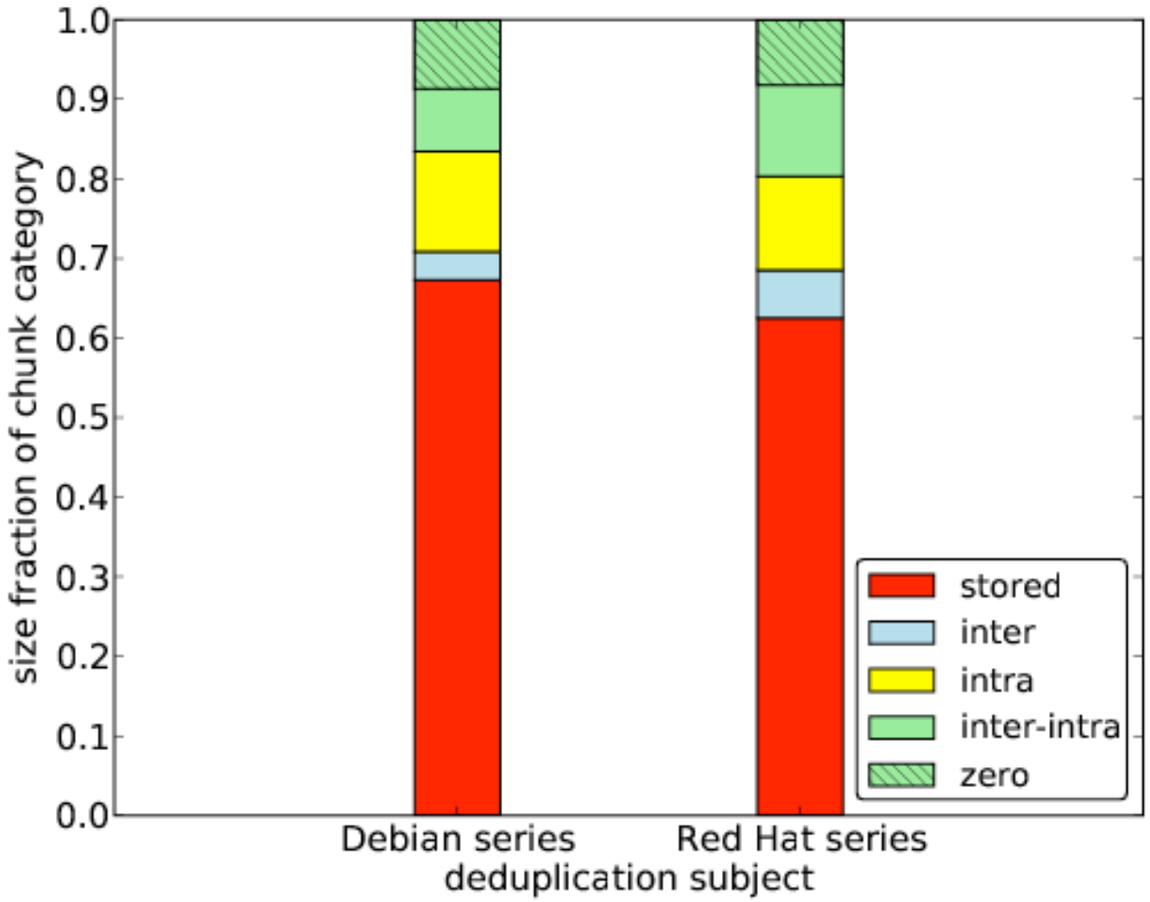
- Different locales deduplicate very well
  - Code remains the same
  - Distributions often include files for all languages
  - Config files determining locale simply select the right files for the language
- VM instances for different locales are highly similar



Ubuntu and Fedora server: English & French

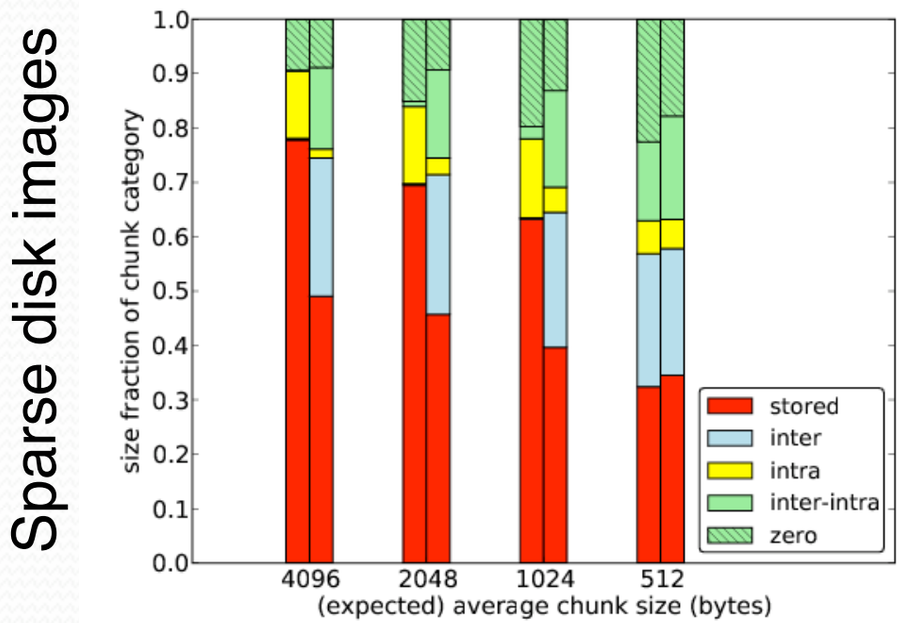
Effects of OS lineage

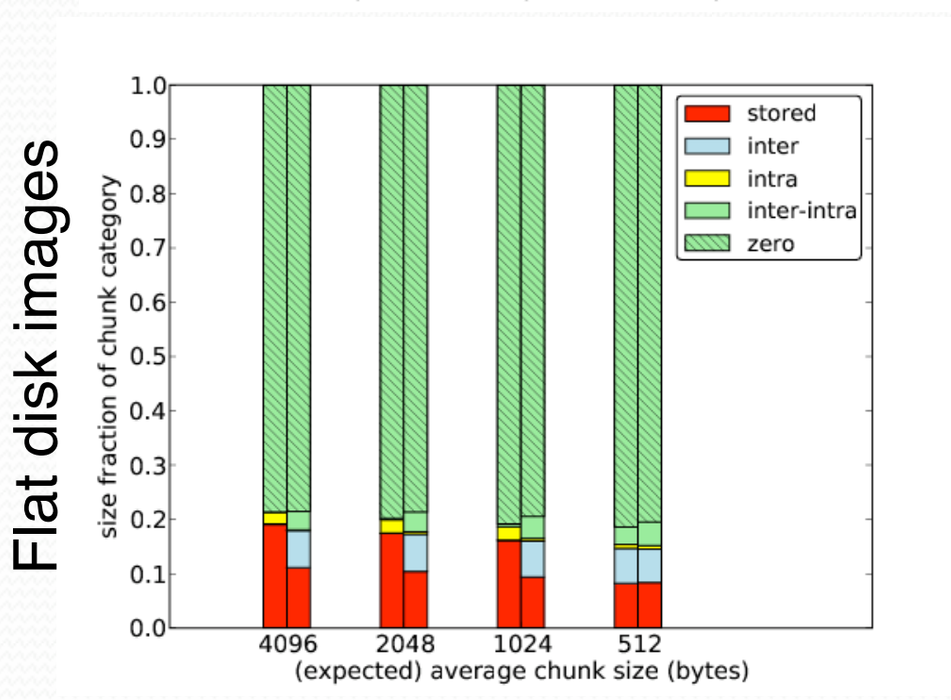
- Deduplicate a wide range of distributions from two lineages
  - Debian
  - Red Hat
- Result: deduplication isn't as effective as for other cases
  - Source code may be similar
  - Binaries differ significantly between distributions



Virtual Machine Managers: VMware vs. VirtualBox

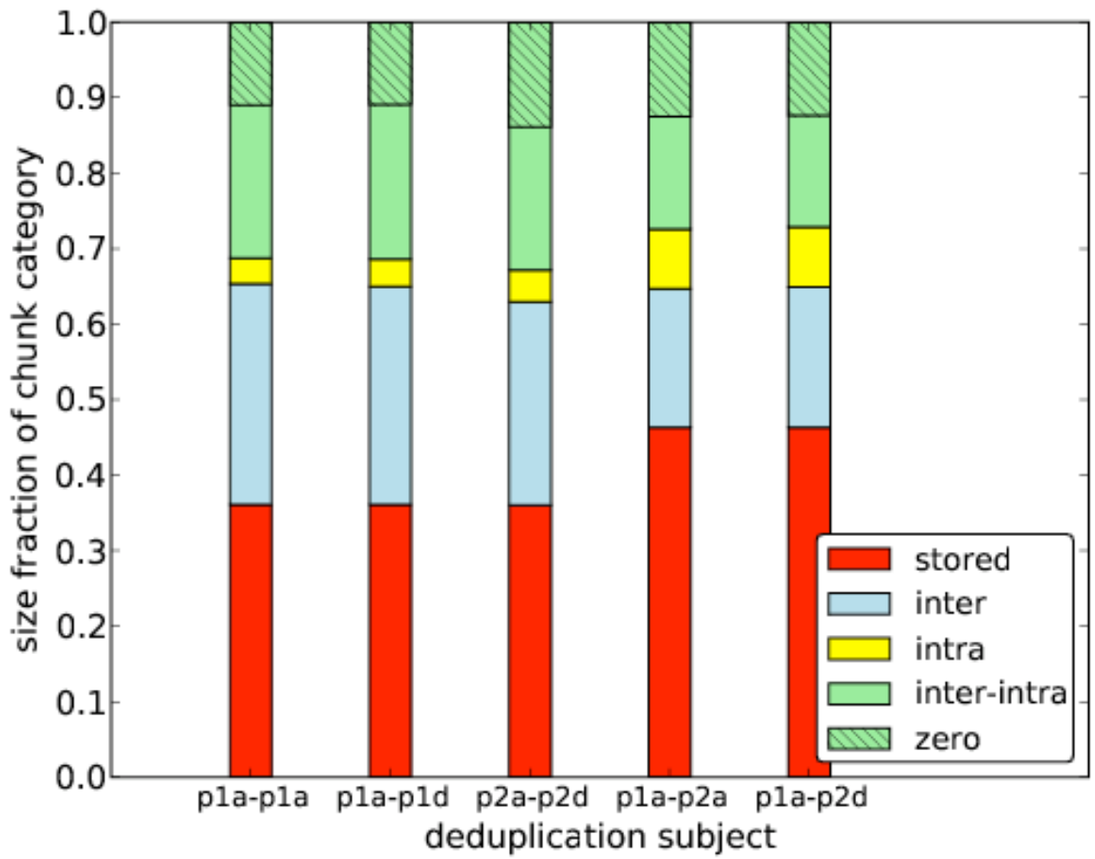
- Ubuntu Server 8.04.1 images in VMware & VirtualBox
  - Sparse images
  - Flat images
- Different VMMs generate differently aligned headers
- Actual guest data are duplicated
  - Good dedup effectiveness for both sparse and flat VM disks





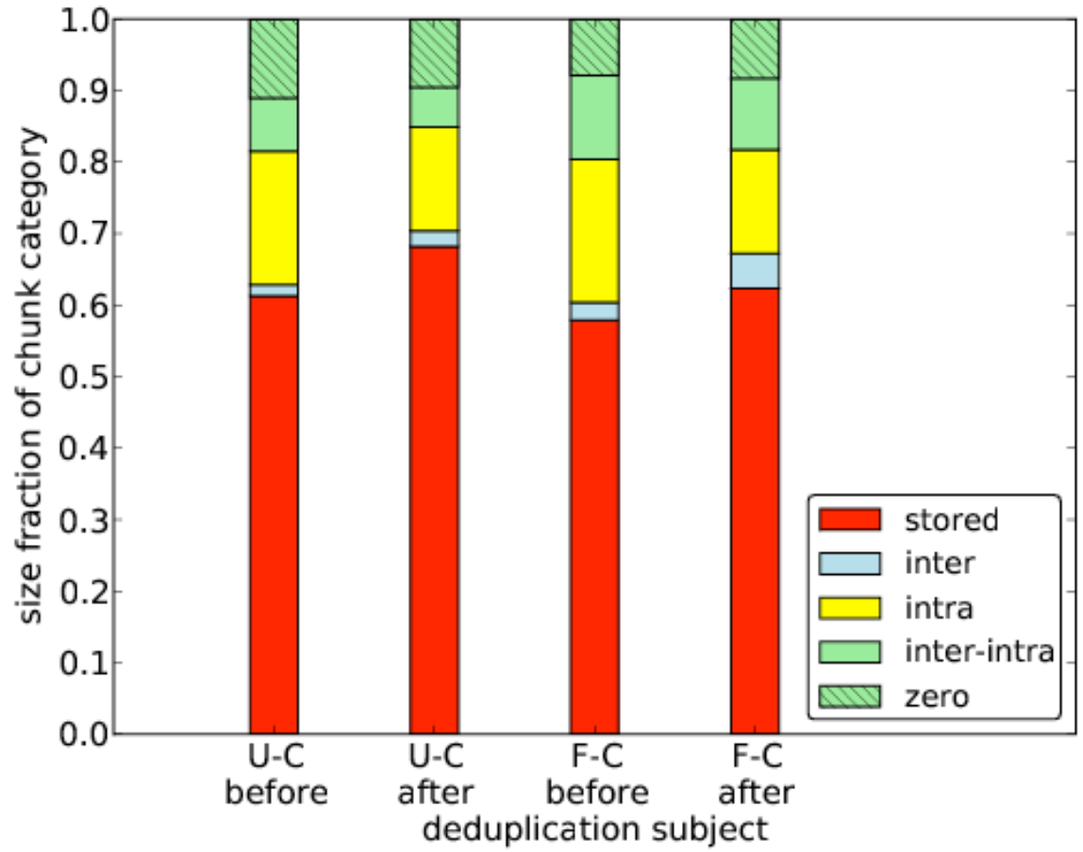
Result: package installation

- Install two package sets in common application areas
  - Vary installation order
  - Dedup pairs of images
- Installation order has little effect on deduplication ratio
- Different package sets don't have much new data
  - Common dependencies appear in both installations



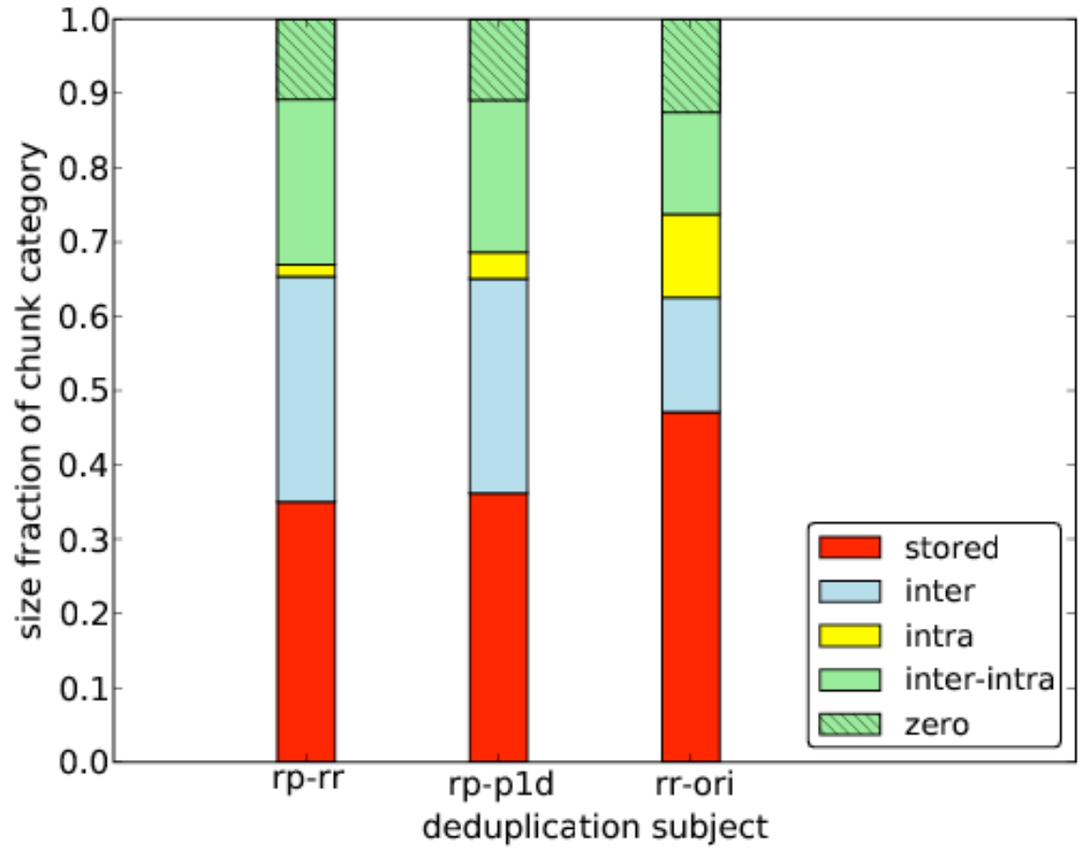
Packaging *systems* impact on deduplication effectiveness

- Compare packaging systems: install in same order for each experiment
  - `deb` (Ubuntu) vs. `rpm` (CentOS)
  - `rpm` (Fedora) vs. `rpm` (CentOS)
- Relatively little sharing: OS difference overwhelms package manager similarity



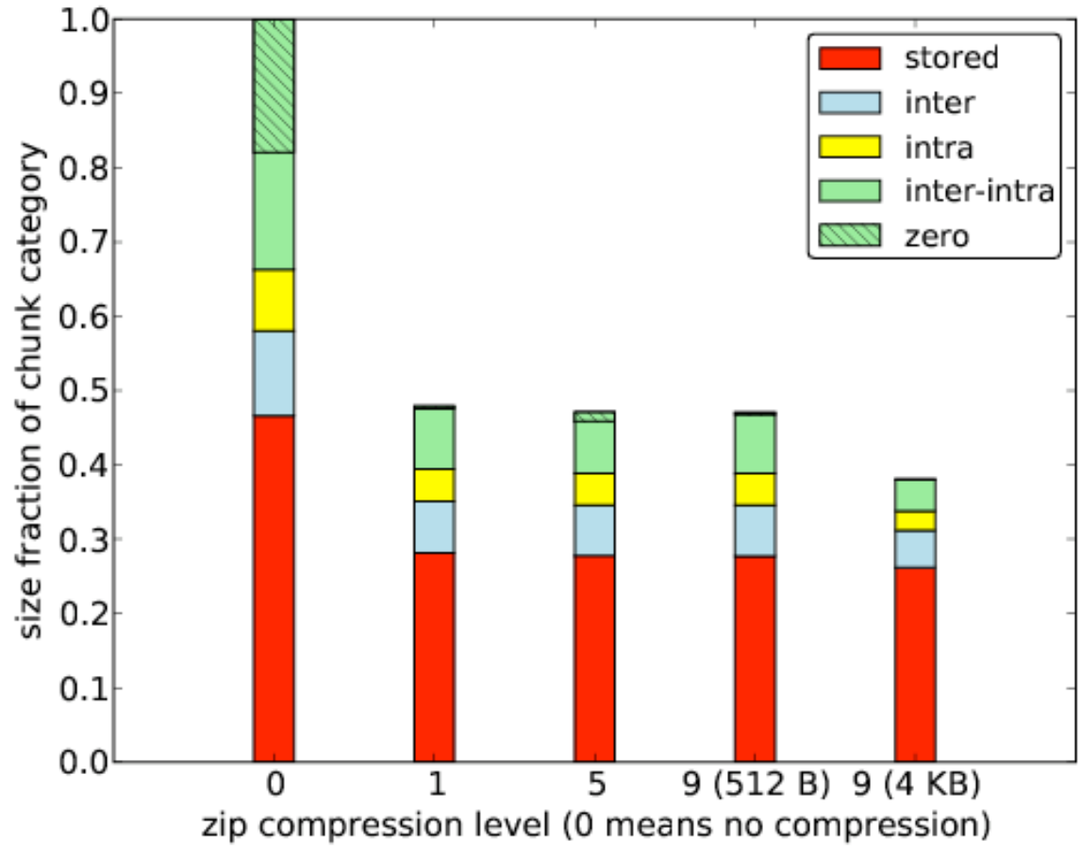
Does package removal matter?

- Removed original packages
  - Removed packages typically leave data in image
- Image with package removed resembles image with package installed
- However, reverted image *differs* from original image
- Don't bother removing packages from deduplicated VM disks…



How effective is chunk compression?

- Compress with zip after deduplication
- Reduces space by 40%
  - Compression level matters little: small window size
- Larger chunks → higher compression
  - Tradeoff: longer time to compress and decompress

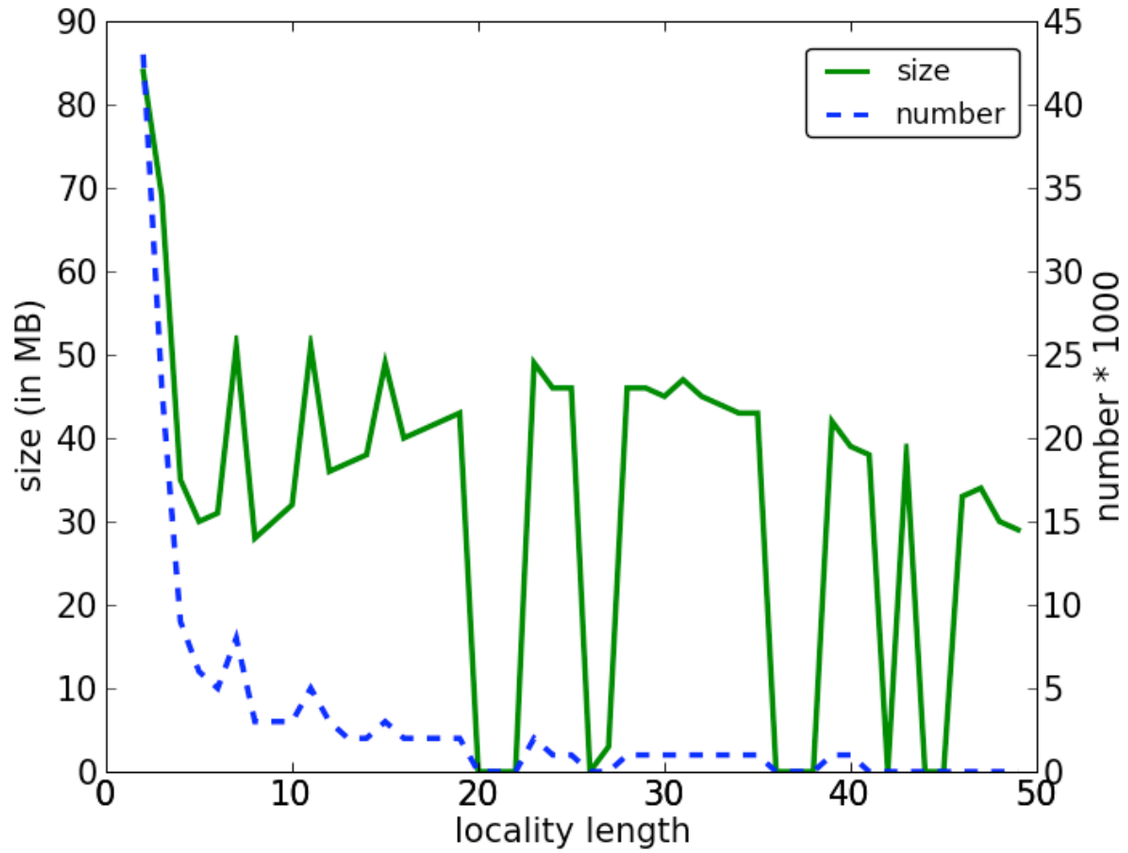


Ongoing work: spatial locality

- A *spatial locality* occurs if the same group of adjacent chunks appears multiple times in a chunk store
  - Don't count sub-localities if covered by super-localities.
- Methodology
  - Generate every possible locality by concatenating adjacent chunk IDs for specified length
  - Deduplicate
  - Remove covered sub-localities
- Detecting ALL localities is hard
  - Detecting certain lengths of localities is more tractable

Result: spatial locality

- Number of localities decreases sharply as locality length increases
- Long localities exist while their immediate predecessors and successors do not
- Different VM group might have different locality pattern
  - Working on showing this…



Ubuntu JeOS 8.04 (one instance)
Fixed-size chunking, 1KB

Future work

- Relationship between deduplication ratio and user-related actions
  - Examine disk images after users have configured them
  - Expect less deduplication, but still effective, especially for similar operating systems
- Explore locality issues: will deduplication hurt sequential I/O?
- Implement deduplication on the fly in VM manager

Conclusions

- Deduplication is efficient for virtual machines, often saving 50–80%
  - More virtual machine instances ➔ more savings
  - More homogeneity ➔ more savings
- Fixed size chunking works well!
- Deduplication works well across many differences
  - OS version
  - Package installations
  - Locale
  - Other differences reduce effectiveness more, but deduplication is still effective
- Integration with conventional data compression works, but limited by relatively small chunk sizes

Thanks!

Questions?

Research supported in part by Symantec & UC Micro