

# A study of practical deduplication

**Dutch T. Meyer**

*University of British Columbia*

*Microsoft Research Intern*

**William Bolosky**

*Microsoft Research*

# Why Dutch is Not Here





# A study of practical deduplication

**Dutch T. Meyer**

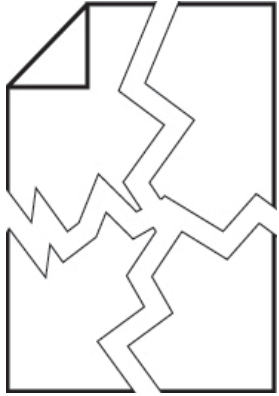
*University of British Columbia*

*Microsoft Research Intern*

**William Bolosky**

*Microsoft Research*

# Why study deduplication?



**9ms**  
**per seek**



**\$0.039**  
~~**\$0.046**~~  
**per**  
**GB**



# When do we exploit duplicates?

## It Depends.

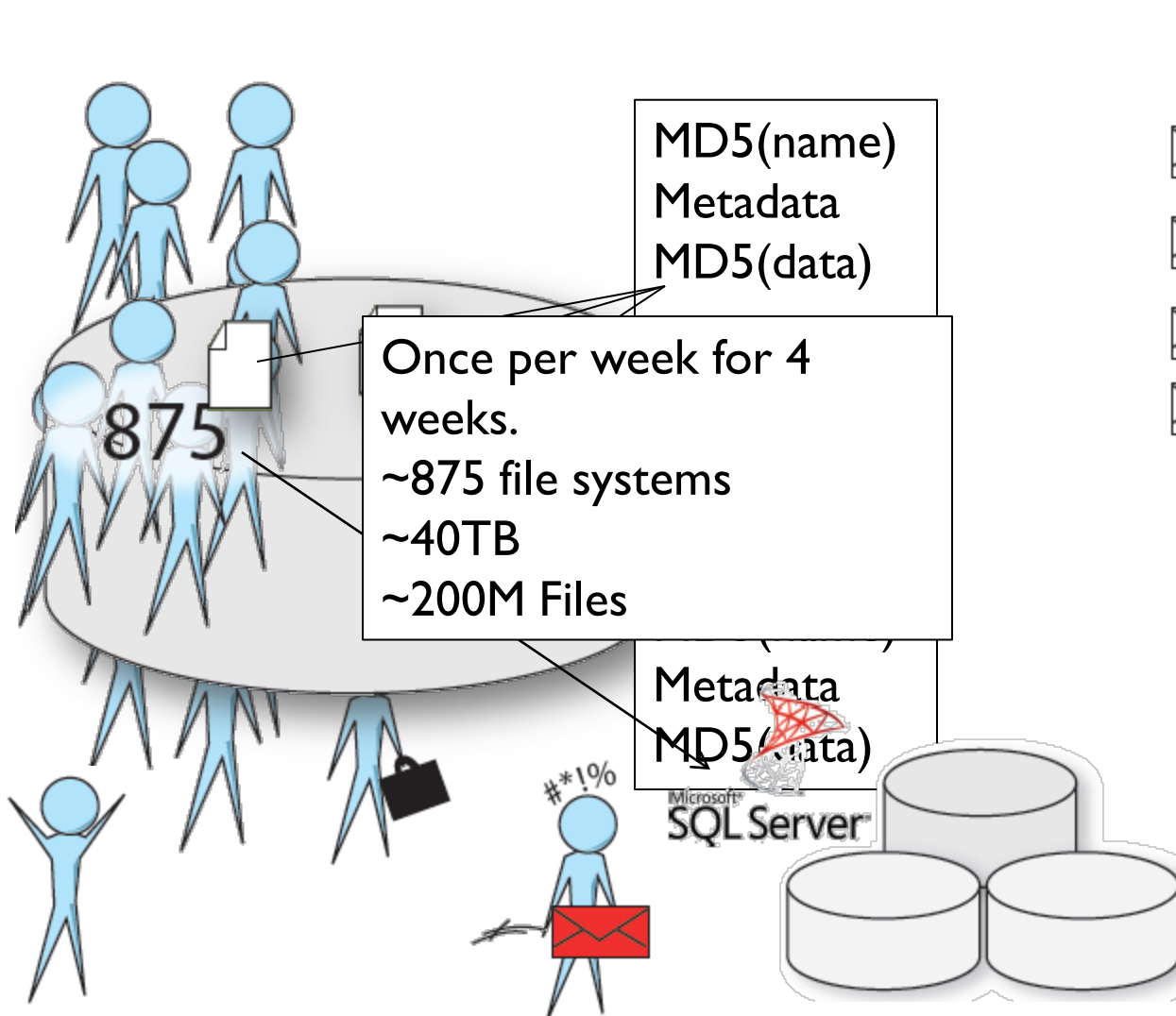
- ❑ How much can you get back from deduping?
- ❑ How does fragmenting files affect performance?
- ❑ How often will you access the data?

- Intro
- Methodology
- “There’s more here than dedup” teaser

(intermission)

- Deduplication Background
- Deplication Analysis
- Conclusion

# Methodology

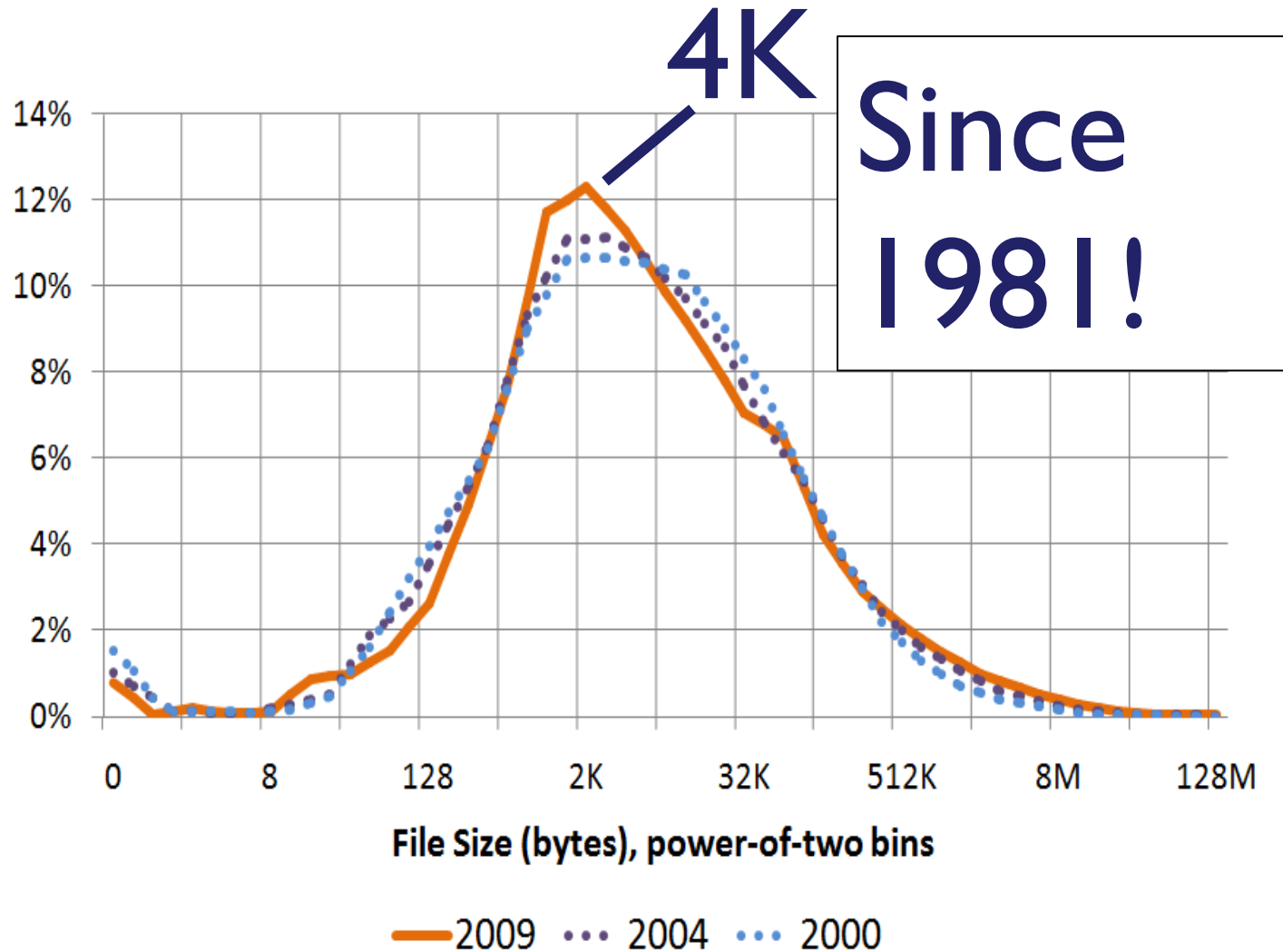


# There's more here than dedup!

- ❑ We update and extend filesystem metadata findings from 2000 and 2004
- ❑ File system complexity is growing
- ❑ Read the paper to answer questions like:

**Are my files bigger now than they used to be?**

# Teaser: Histogram of file size



# There's more here than dedup!

## How fragmented are my files?

# Teaser: Layout and Organization

- ❑ High linearity: only 4% of files fragmented in practice
  - ❑ Most windows machines defrag weekly
- ❑ One quarter of fragmented files have at least 170 fragments

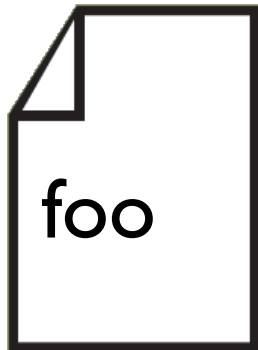
- Intro
- Methodology
- “There’s more here than dedup” teaser

(intermission)

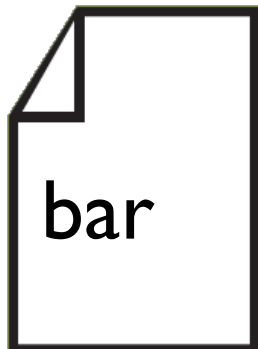
- Deduplication Background
- Deplication Analysis
- Conclusion

# Dedup Background

## Whole file Deduplication



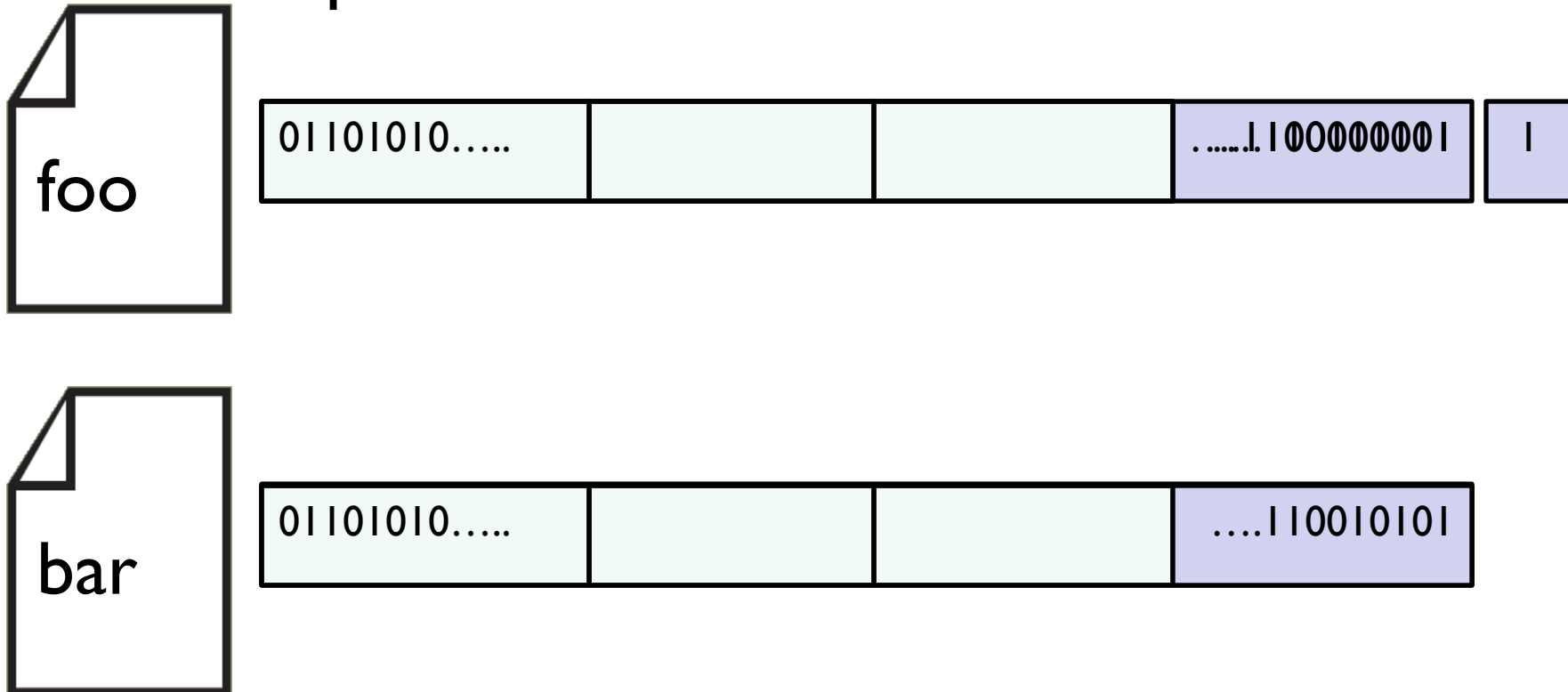
01101010..... ..110010101



01101010..... ..110010101

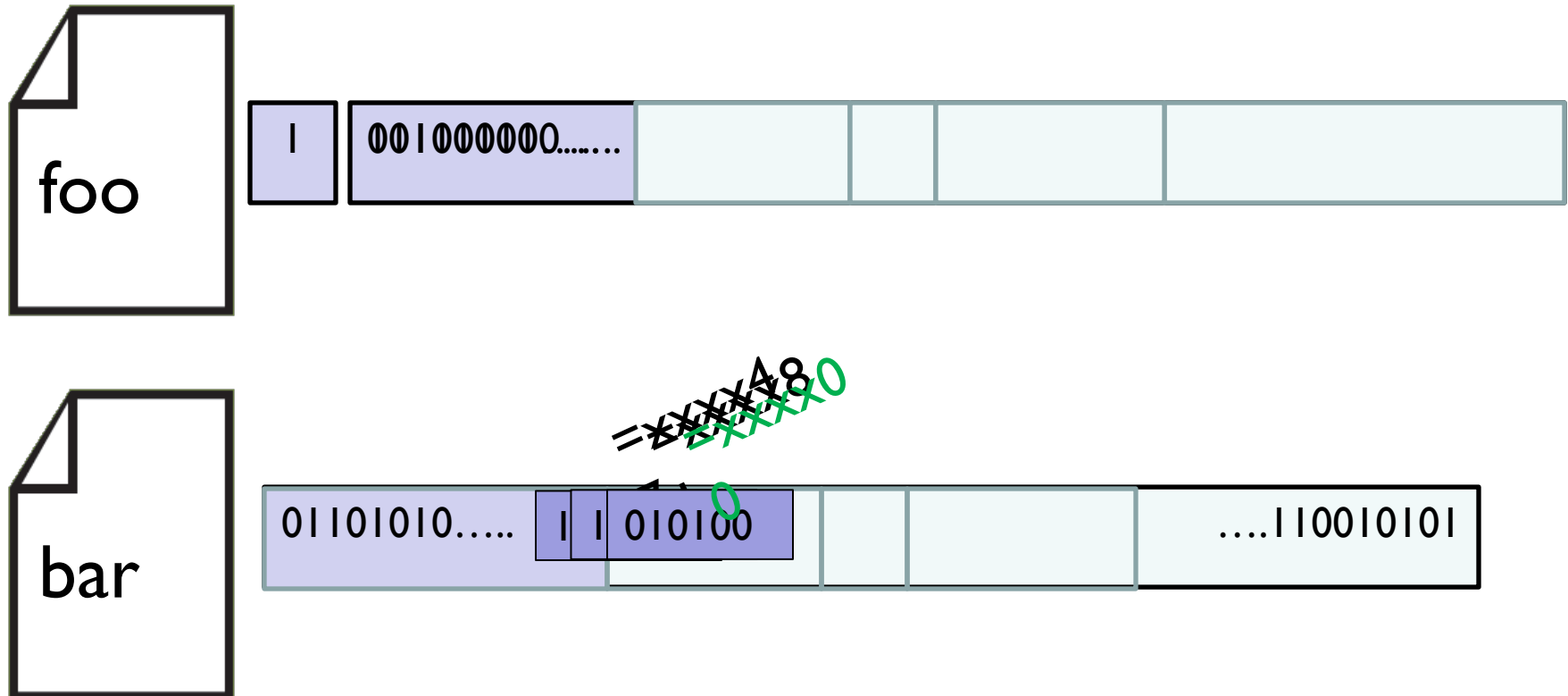
# Dedup Background

## Fixed Chunk Deduplication



# Dedup Background

## Rabin Fingerprinting

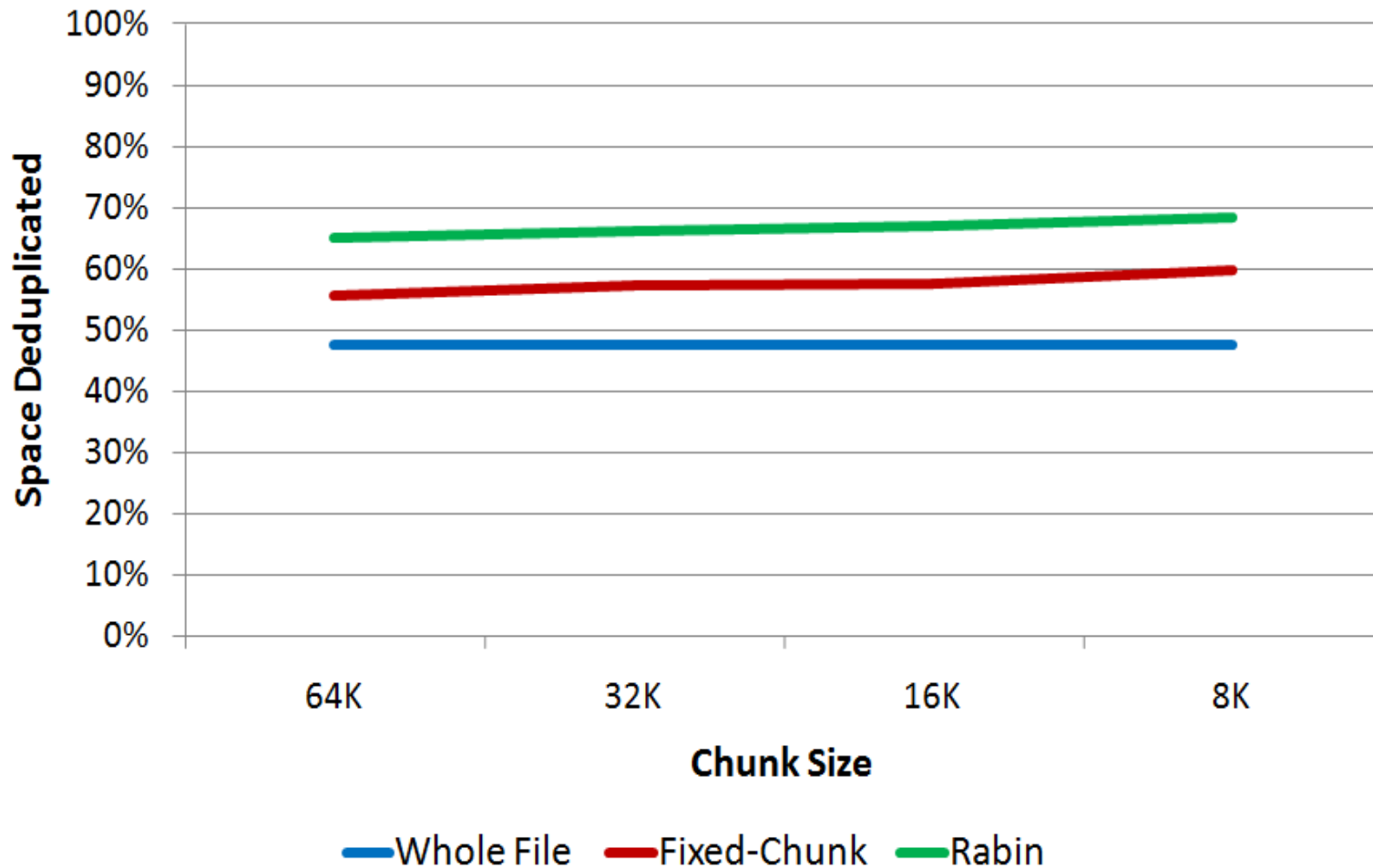


# The Deduplication Space

Algorithm	Parameters	Cost	Deduplication effectiveness
Whole-file		Low	Lowest
Fixed Chunk	Chunk Size	Seeks CPU Complexity	Middle
Rabin fingerprints	Average Chunk Size	Seeks More CPU More Complexity	Highest

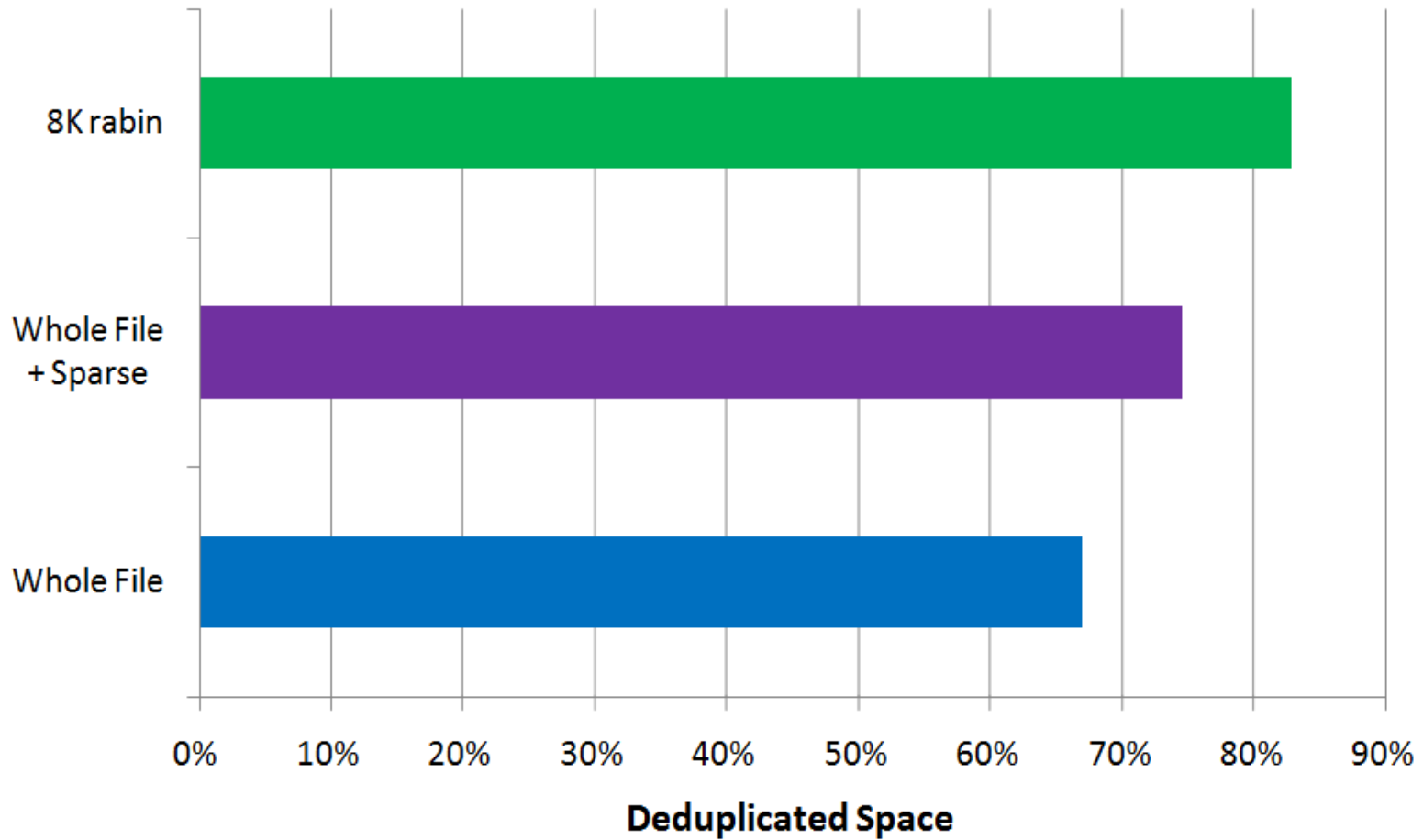
# What is the relative deduplication rate of the algorithms?

# Dedup by method and chunk size



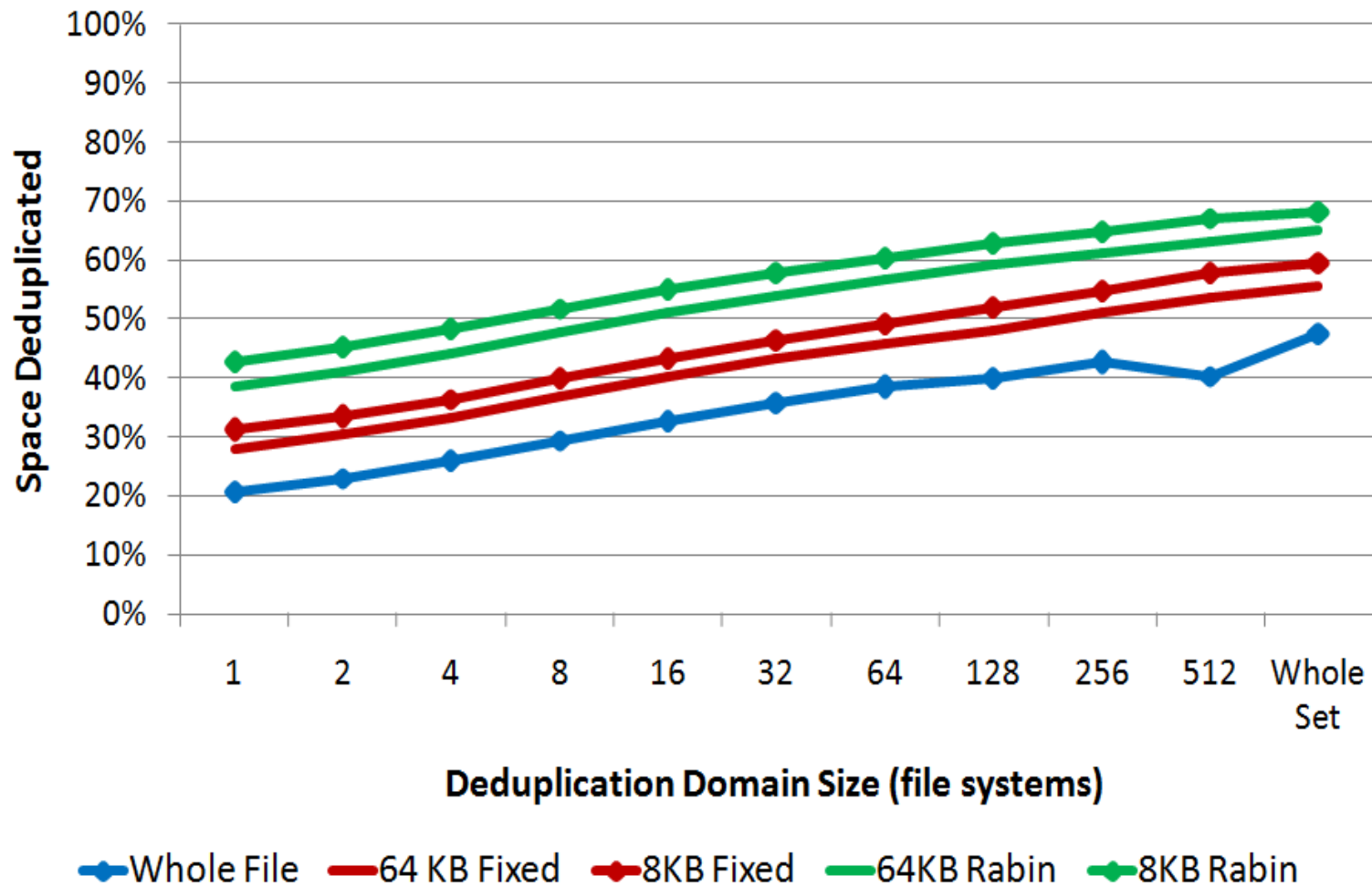
# What if I was doing full weekly backups?

# Backup dedup over 4 weeks



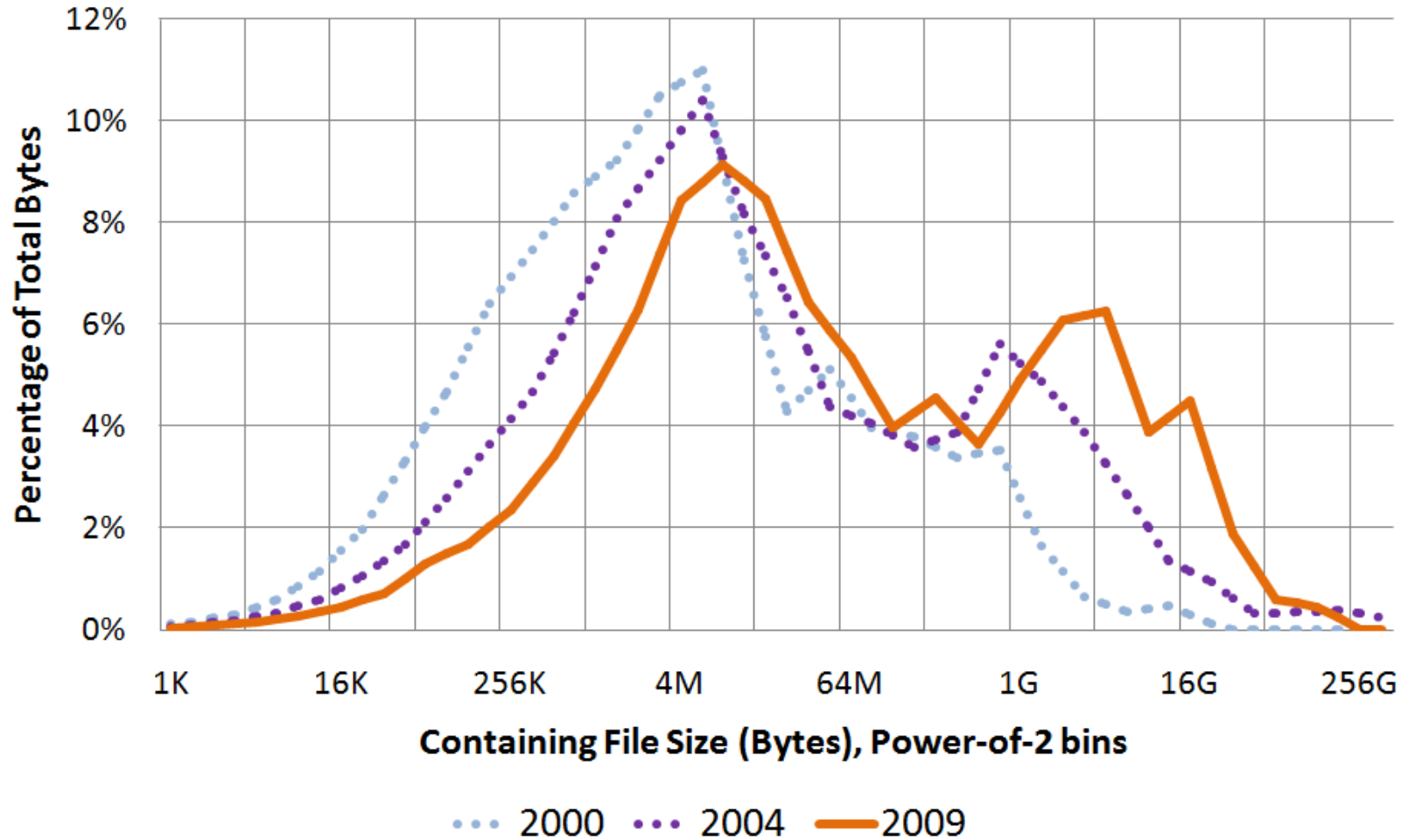
# How does the number of filesystems influence deduplication?

# Dedup by filesystem count



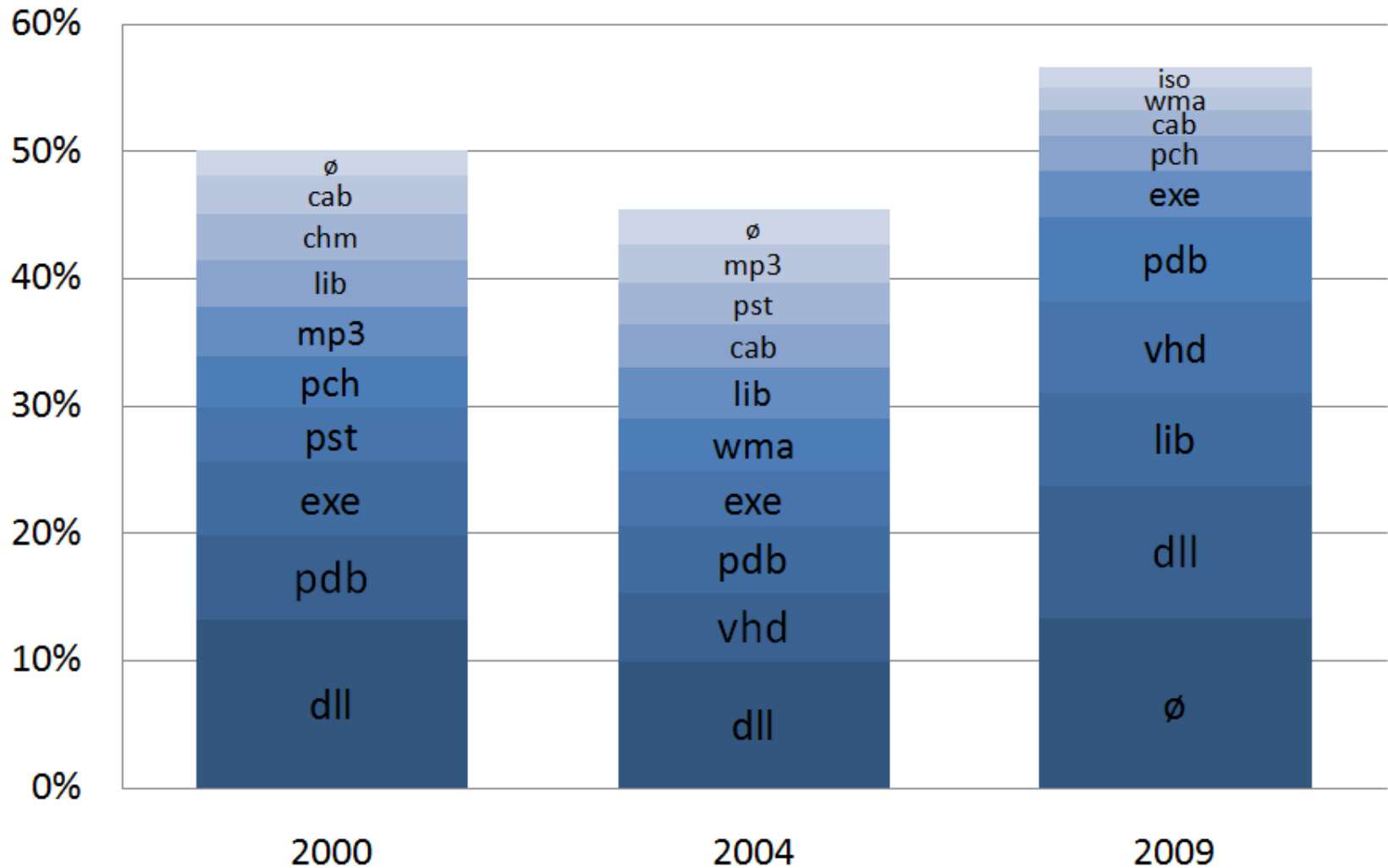
**So what is filling up all this space?**

# Bytes by containing file size

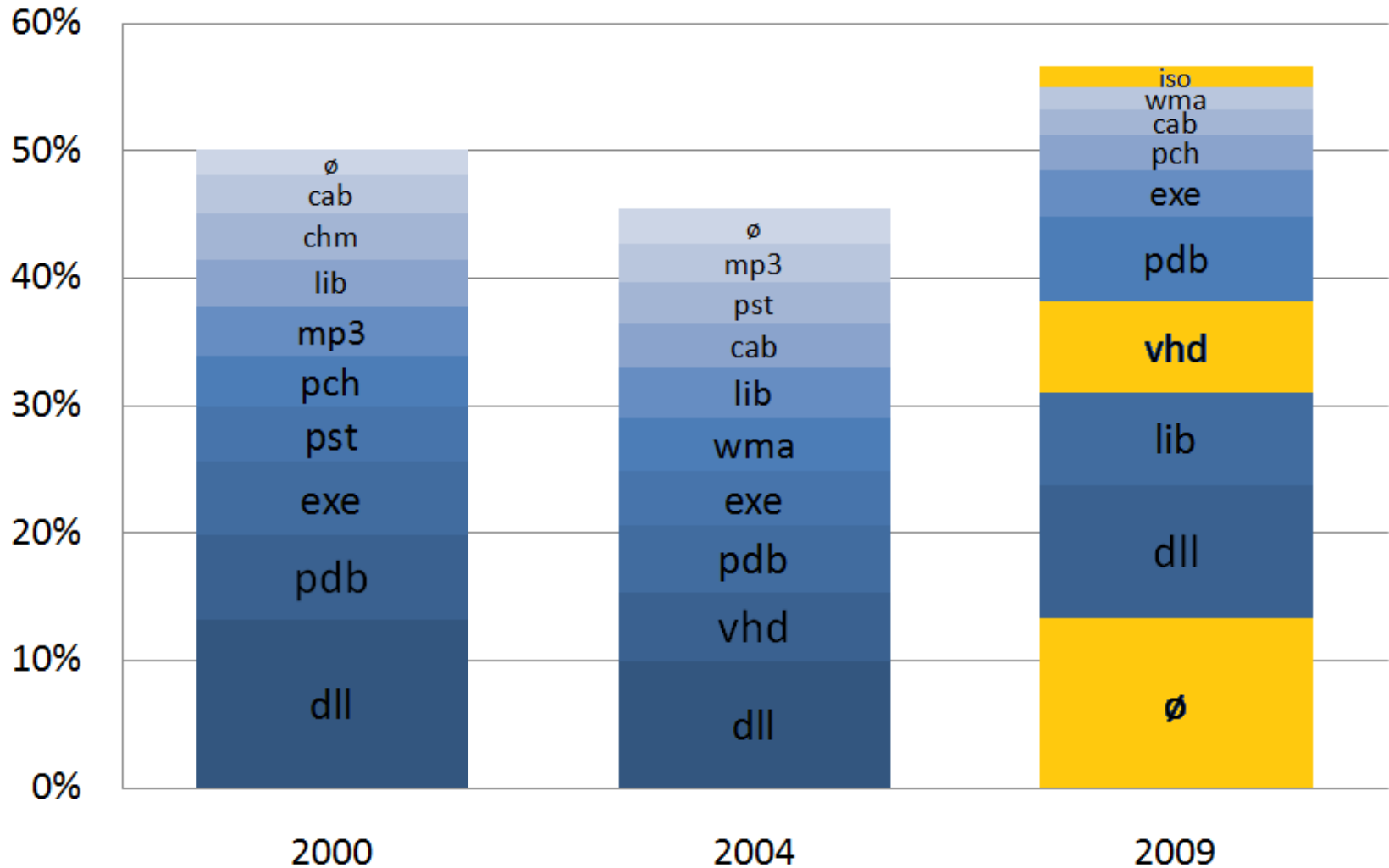


# What types of files take up disk space?

# Disk consumption by file type



# Disk consumption by file type



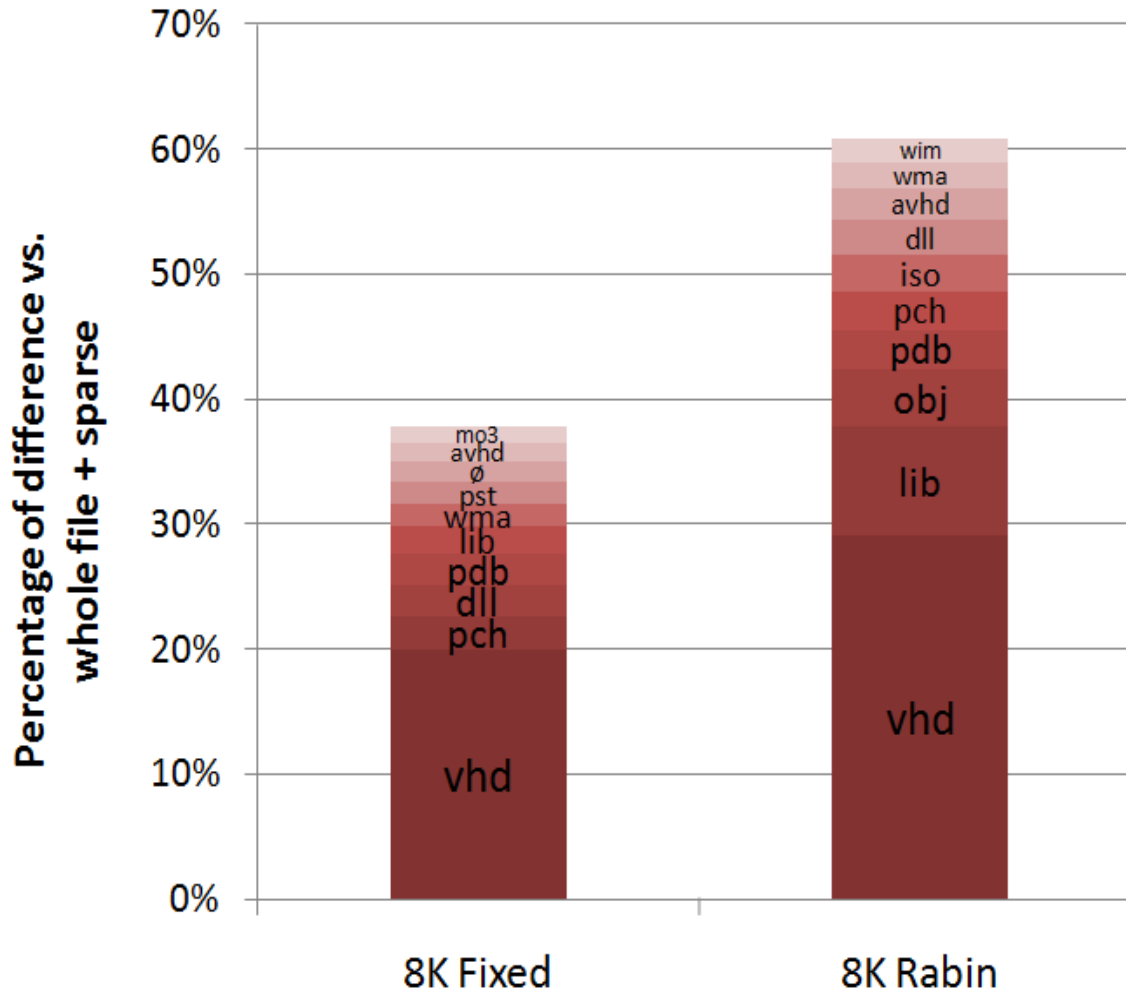
**Which of these types deduplicate well?**

# Whole-file duplicates

Extension	% of Duplicate Space	Mean File Size (bytes)	% of Total Space
dll	20%	521K	10%
lib	11%	1080K	7%
pdb	11%	2M	7%
<none>	7%	277K	13%
exe	6%	572K	4%
cab	4%	4M	2%
msp	3%	15M	2%
msi	3%	5M	1%
iso	2%	436M	2%
<a guid>	1%	604K	<1%

**What files make up the 20% difference between whole file dedup and sparse file, as compared to more aggressive deduplication?**

# Where does fine granularity help?



# Last plea to read the whole paper

- ❑ ~4x more results in paper!
- ❑ Real world filesystem analysis is hard
  - ❑ Eight machines months in query processing
  - ❑ Requires careful simplifying assumptions
  - ❑ Requires heavy optimization

- ❑ The benefit of fine grained dedup is  $< 20\%$ 
  - ❑ Potentially just a fraction of that.
- ❑ Fragmentation is a manageable problem
- ❑ Read the paper for more metadata results

We're releasing this dataset  
(when I finish the  
anonymization)