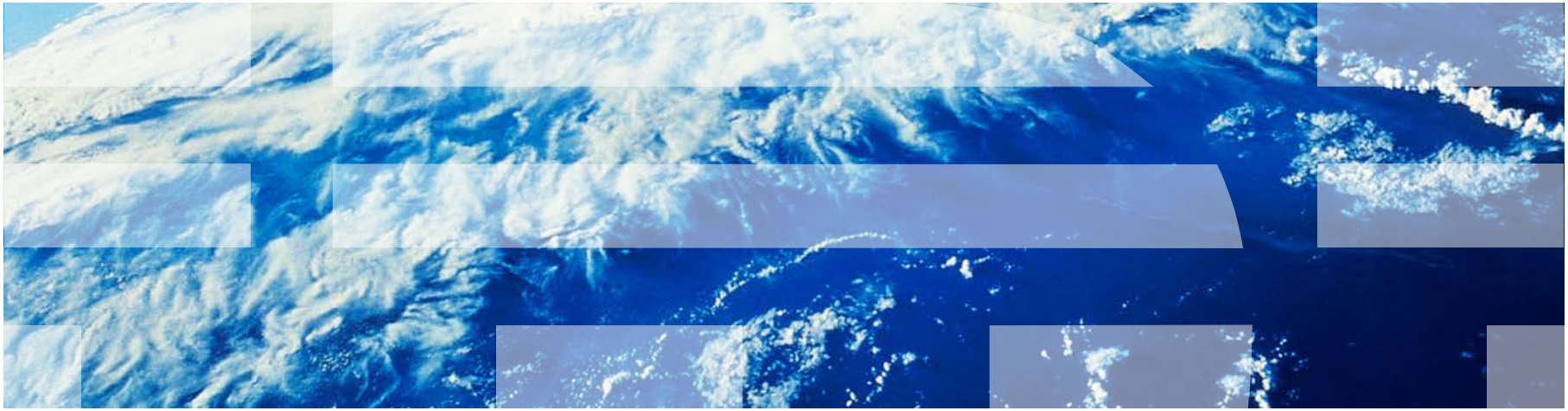
Improving the Performance of Backup Candidate File Selection using Inode Bitmap

Sosuke Matsui, Tsuyoshi Miyamura, Noriko Tanemura,
Terue Watanabe, Norie Iwasaki

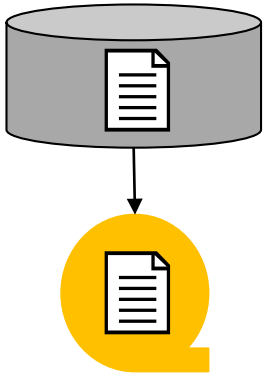
Tokyo Software & Systems Development Lab, IBM Japan

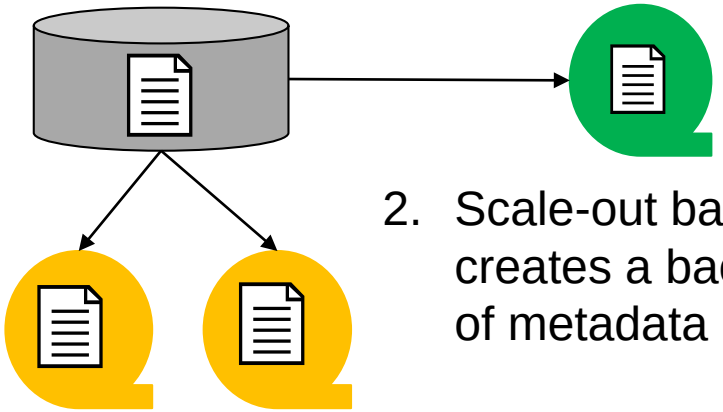


© 2017 IBM Corporation

HSM and Scale-out backup

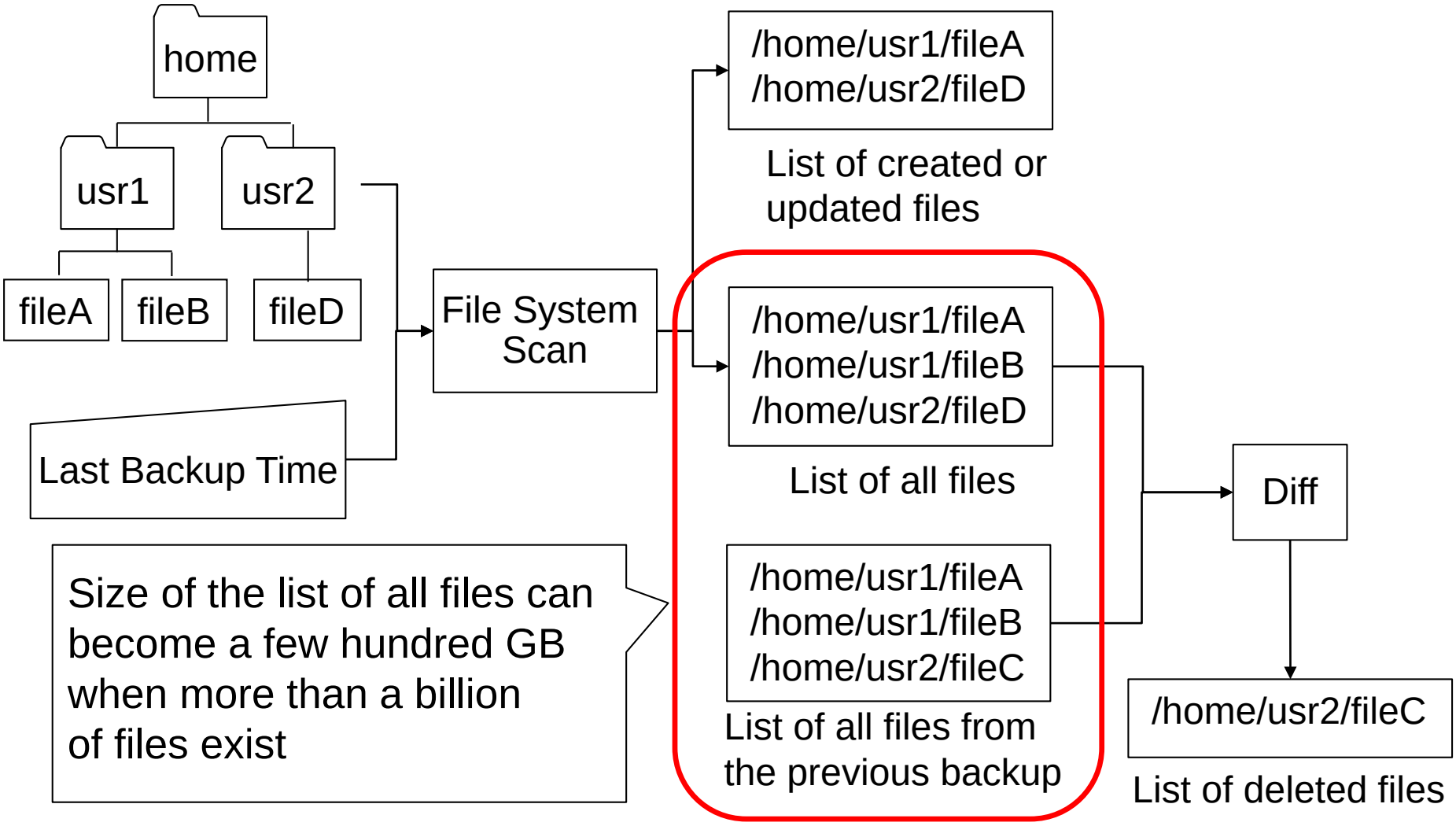
- What is Hierarchical Storage Management (HSM)
  - –Manages storage space efficiently using high-cost and low-cost media
- What is Scale-out backup
  - –A kind of backup used with HSM
  - –Assumes file data is already backed up by HSM
  - –Creates a backup of inodes and the directory structures of a file system when executed
  - –Faster than existing backup methods since a backup of file data is already done by HSM





Improving the performance of backup candidate file selection is the key for Scale-out backup

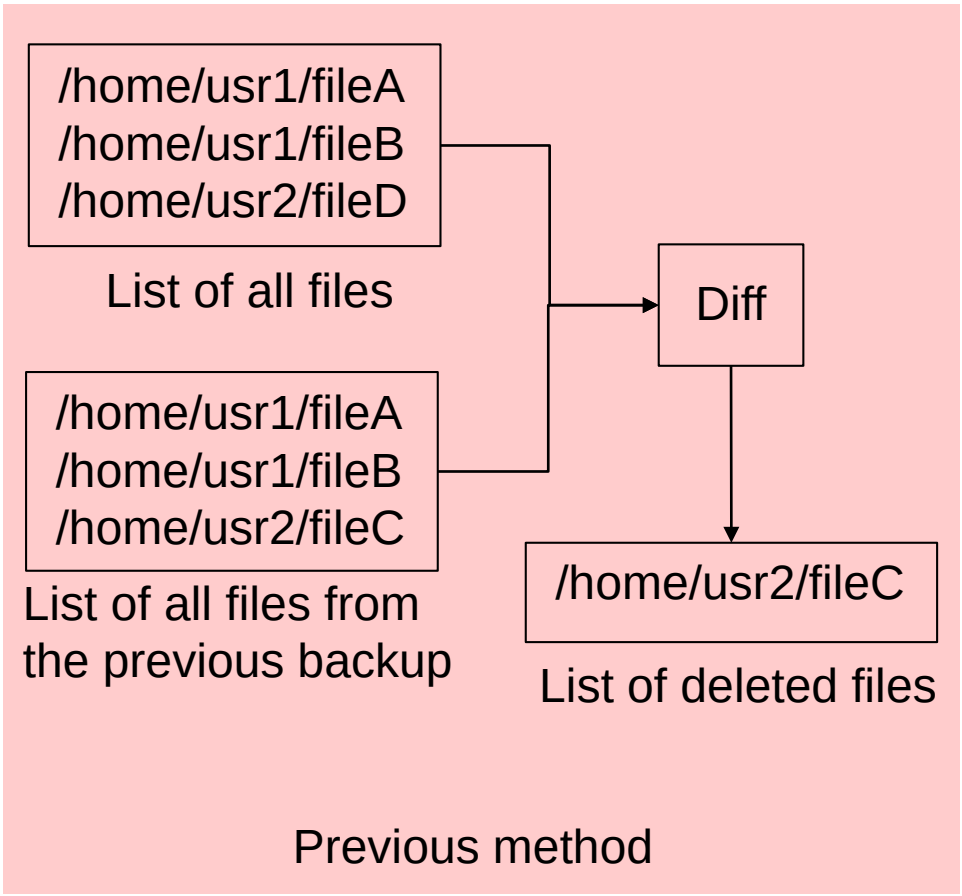
Backup candidate file selection

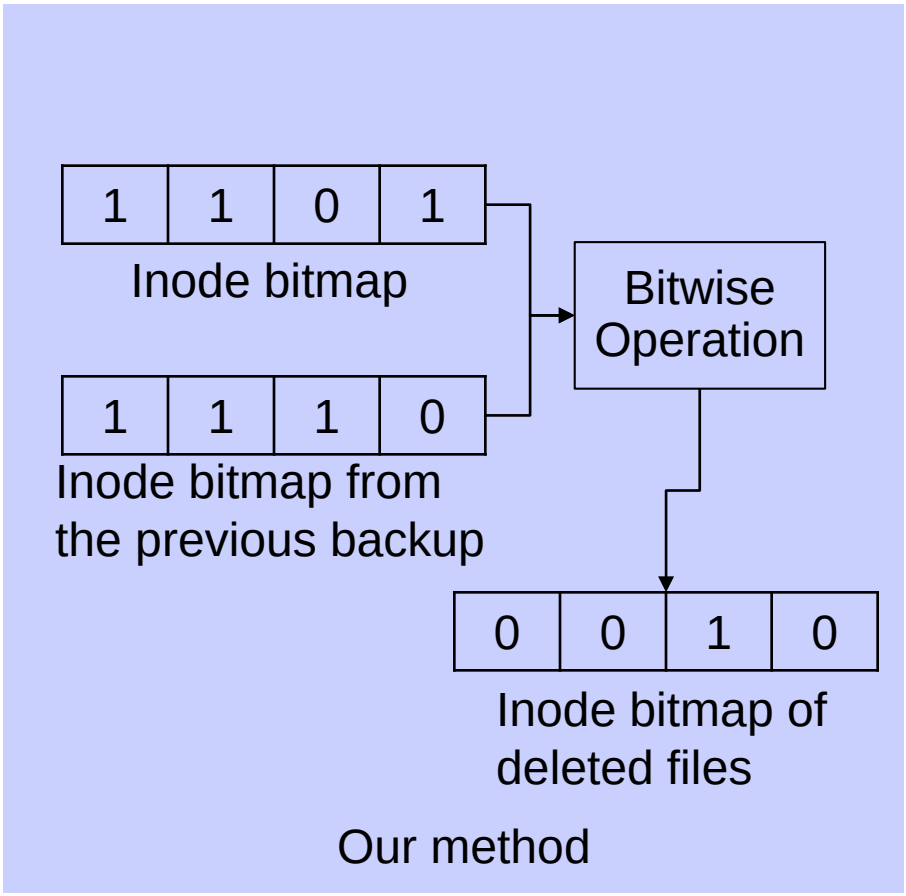


Cannot find backup candidate files efficiently because of large intermediate files when more than a billion of files exist

Our goal and approach

Goal: Improve the performance of backup candidate file selection

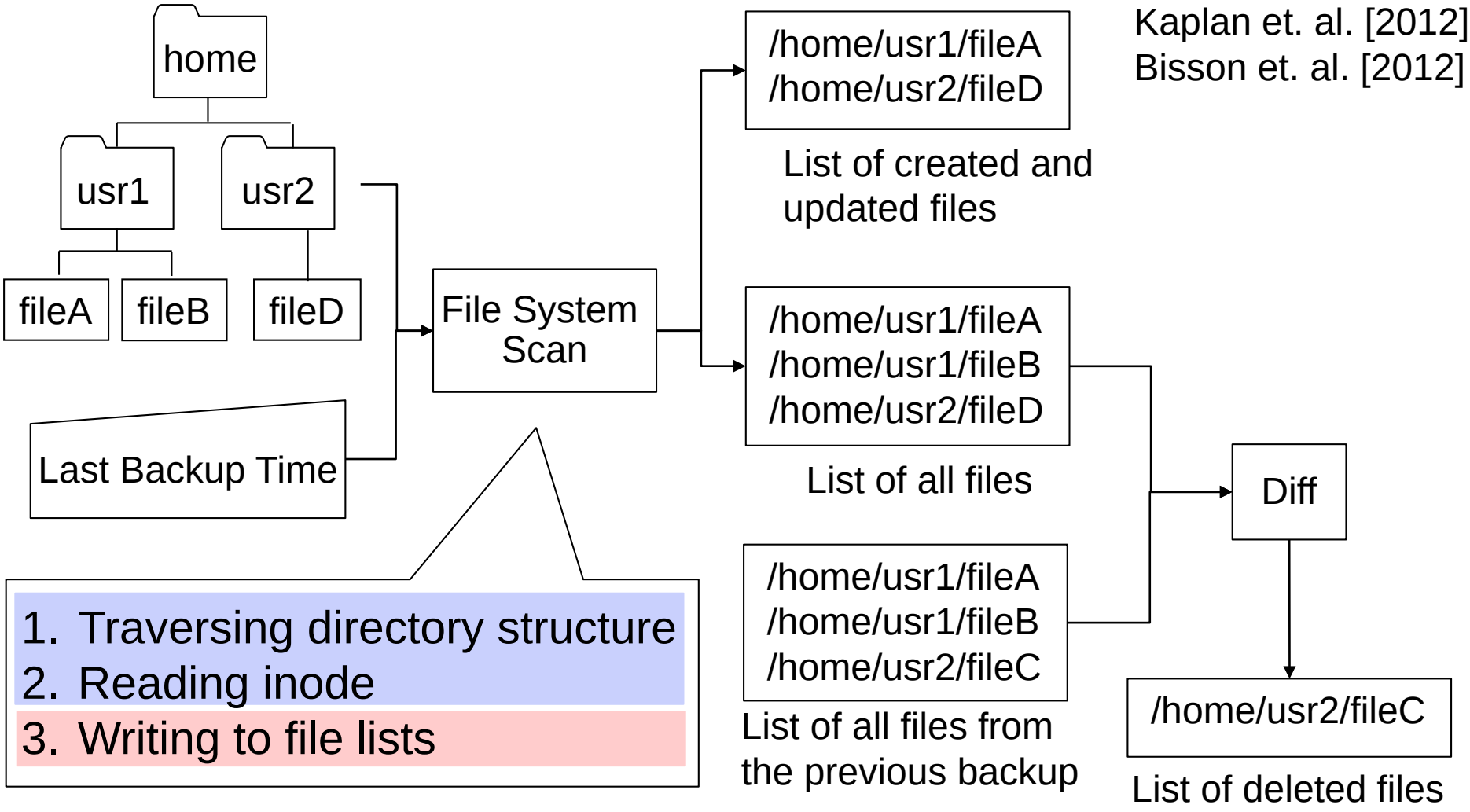




Our approach: Uses inode bitmaps for finding deleted files

Effect: Reduces the size of intermediate files for backups

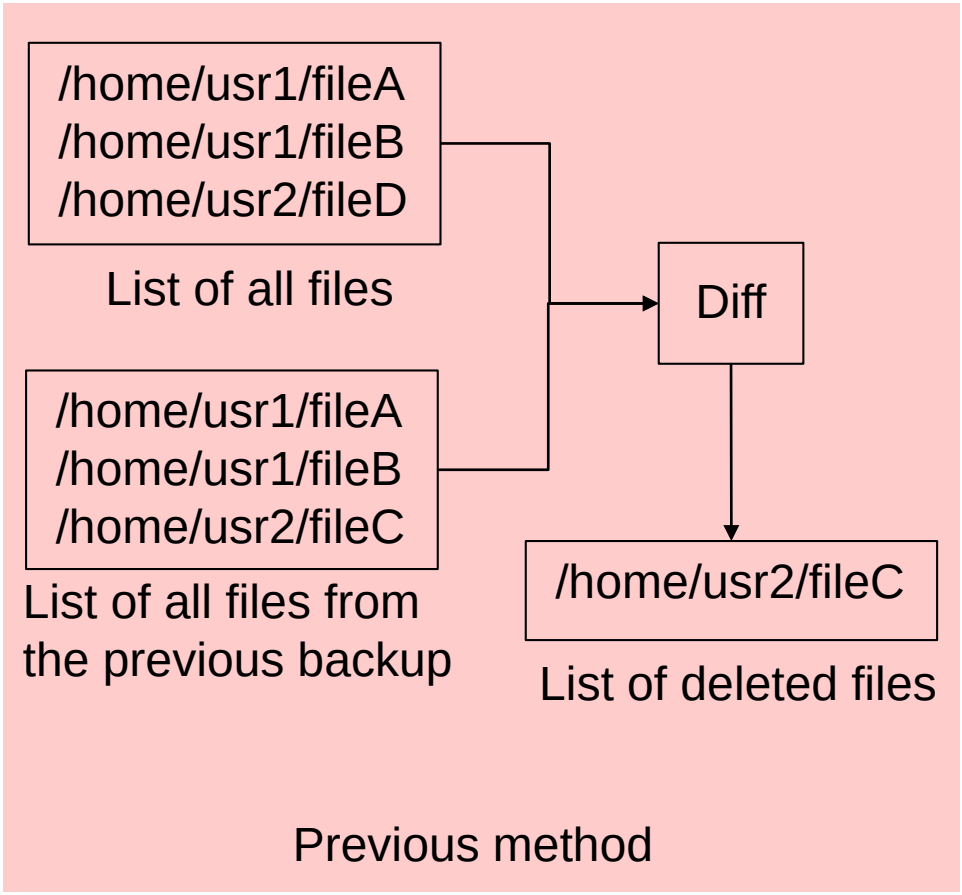
Related work

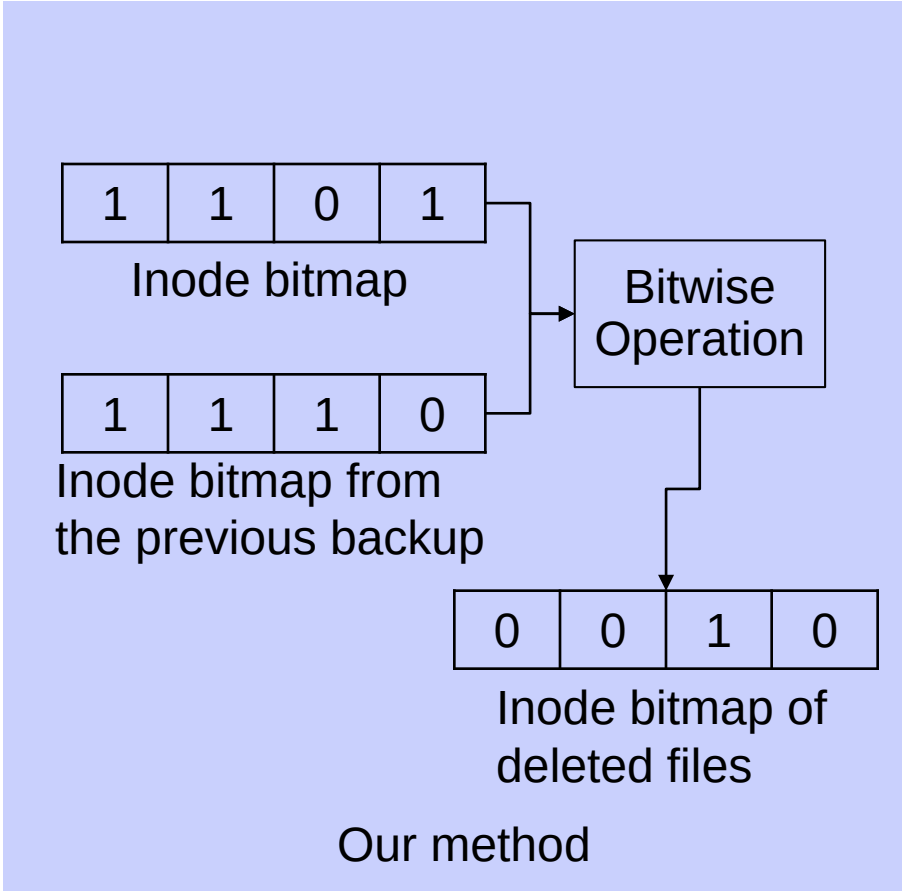


Improved the performance of first two steps in the FS scan

Size of list of all files has not been discussed

Basic idea of our approach





1. Reduces the size of intermediate files using inode bitmaps
2. Finds deleted files by bitwise operation of inode bitmaps

Preliminary experiments

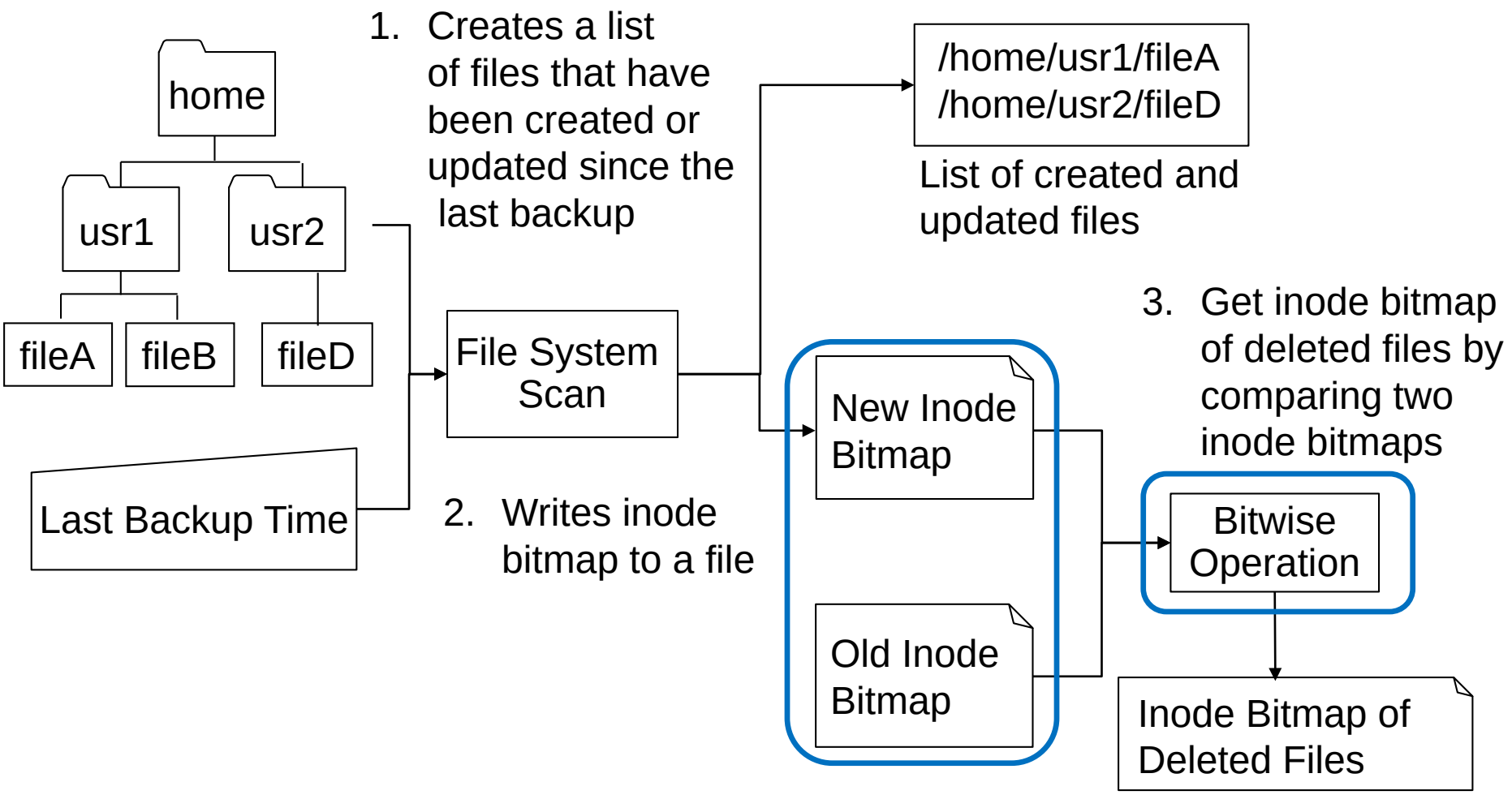
Measured the size of intermediate files and time to create them for file list and inode bitmap

- A list of 10 billion files
- 10 billion bits of an inode bitmap

Format of intermediate files	Size of intermediate files	Time to create intermediate files
File list	640 GB	21629.0 seconds
Inode bitmap	1.25 GB	42.4 seconds

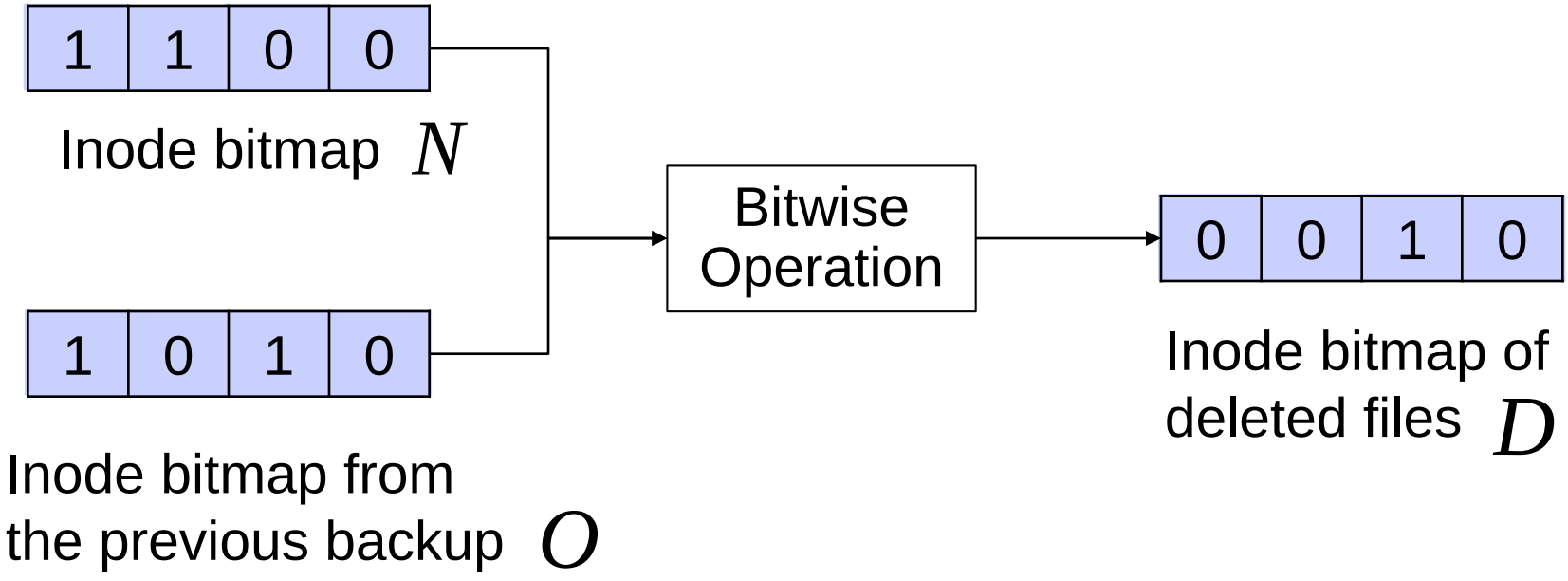
Using inode bitmaps as intermediate files can have an advantage

Flowchart of our approach



1. Reduces the size of intermediate files using inode bitmaps
2. Finds deleted files by bitwise operation of inode bitmaps

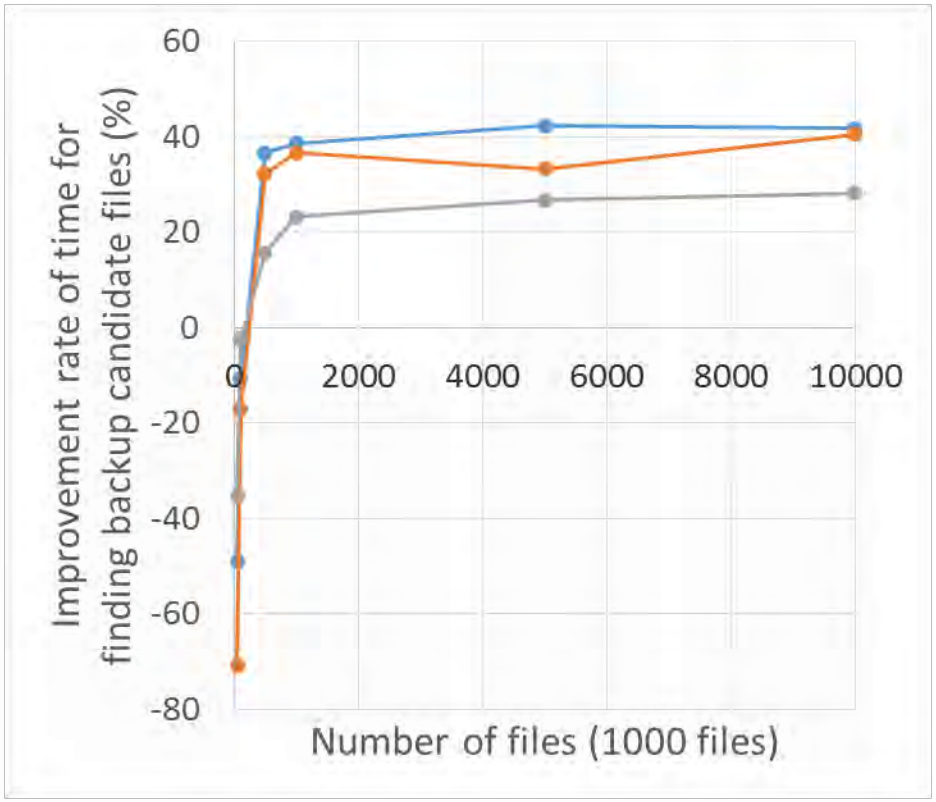
Finding deleted files from inode bitmaps



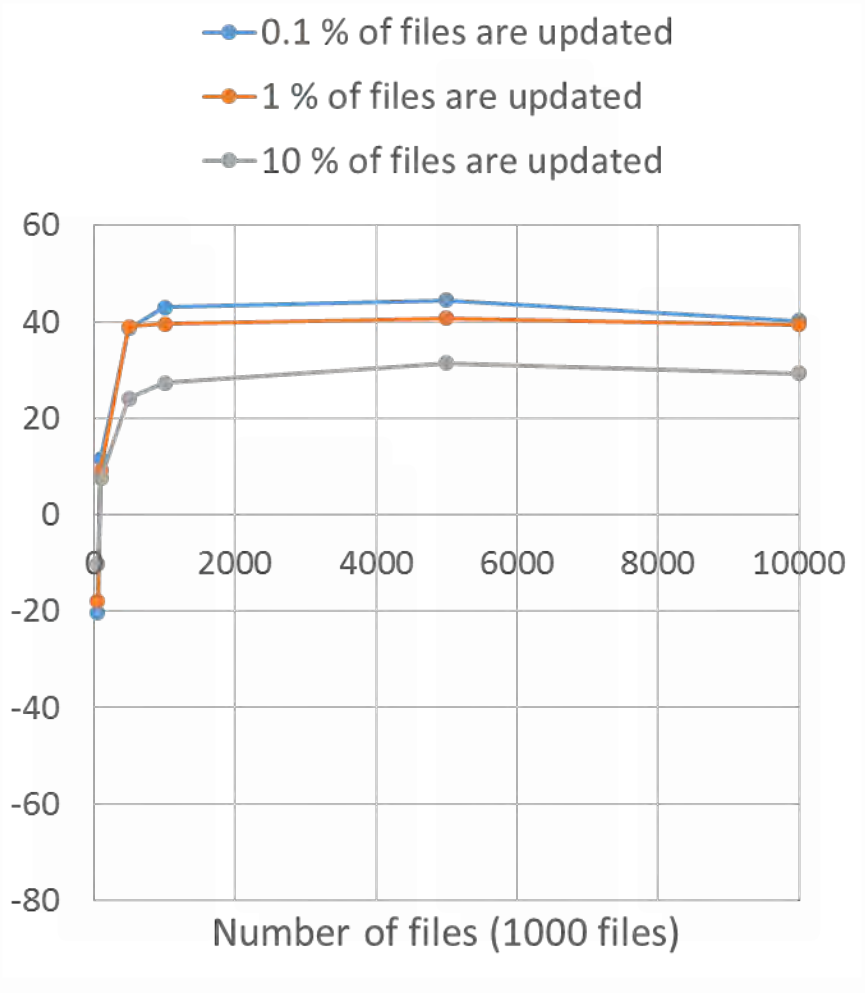
Inode bitmap of deleted files can be calculated by:

$$D = O \wedge \left(\neg N \right)$$

Experimental results



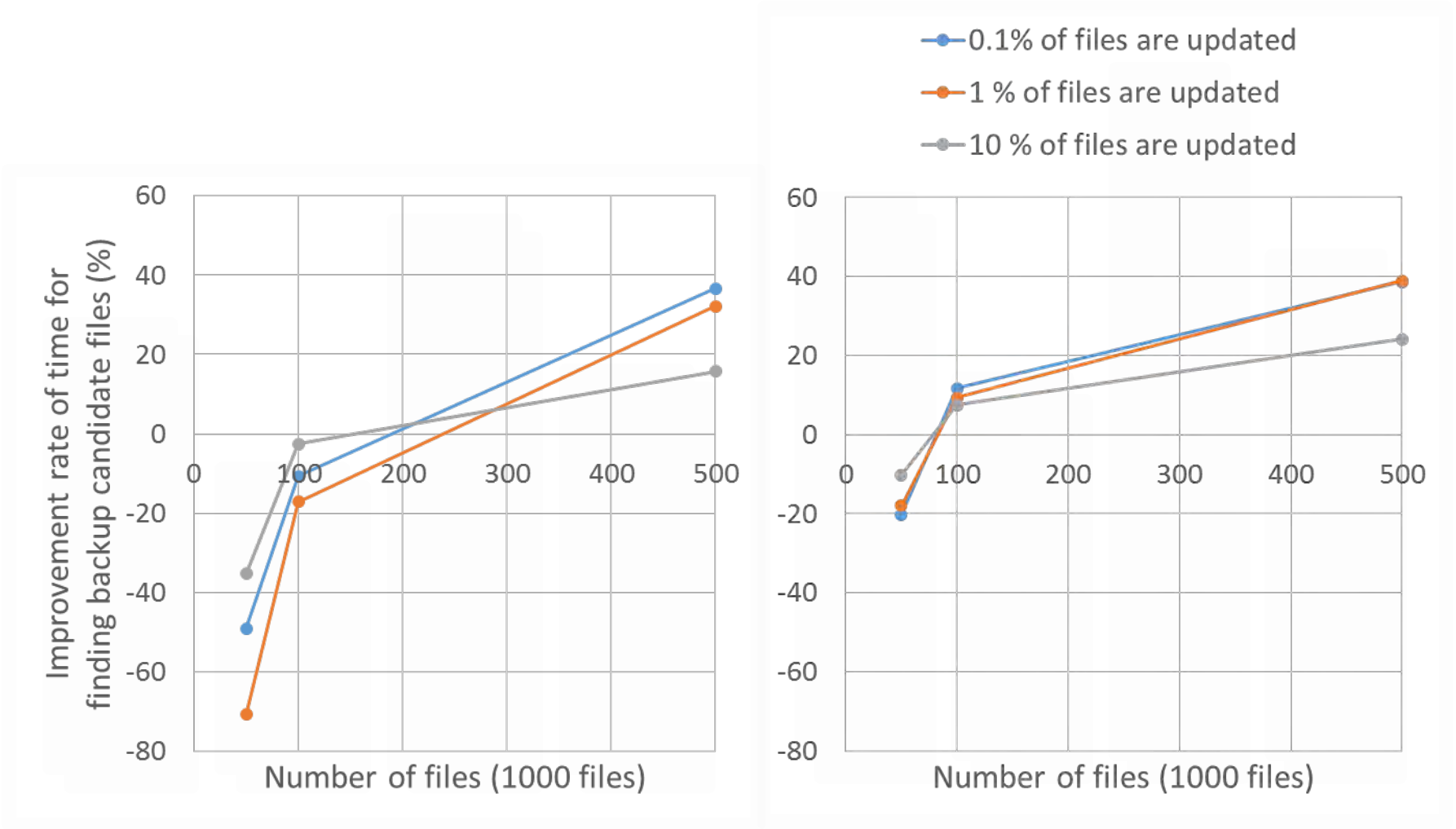
Average file path length: 64 bytes



Average file path length: 128 bytes

Our method improved the performance of backup candidate file section up to 44.5%

Results with small number of files



Previous method showed the better results when a number of files in a file system is small

Conclusion and future work

- Conclusion
  - Improved the performance of backup candidate file selection up to 44.5 %
  - Used inode bitmap as intermediate files
  - Calculated the bitwise operation of inode bitmaps to find deleted files

- Future work
  - Evaluate the performance of our method with 10 billion files