JBox Report

Research the performance of different file types on JBox

Yuan Su W1185751

Evaluation Design

User case

For different type files, we want to demonstrate the performance between File level deduplication and Chunk level deduplication - (1) Fixed chunk and (2) Variable chunk algorithm. For this deduplication experiment, we focused on File Update case. In addition, we provided multiple combinations of different power or different dividers with Fixed and Variable chunk algorithm. Therefore, we analyzed the results by adapting the two major performance indicators - transaction time and deduplication ratio.

Scenario

For File Update case, we attempted to simulate real case updating scenario. For each file types CSV, LOG, and PDF, by appending content regarding to the testing file type in front of the target file or appending content regarding to the testing file type at the end of the file, first we created five files which sizes are around 50M, 100M, 150M, 200M, 250M. For file type ZIP, we incremented the file size from 50M, 100M, 150M, 200M, 250M by appending new contents into ZIP file. Then in experiment, we copy the file size from small to large to JBox with an unified name to simulate file update case. The purpose is to verify bit-shifting issue for different deduplication algorithms.

For each file type, we also tested with multiple combinations as following table:

Algorithm	Parameters
Nor (Normal, no chunks)	
Fixed (Fixed Chunk Deduplication)	Power (Anchor): 2^{21}, 2^{22}, 2^{23}, 2^{24}, 2^{25}
	Divider: 4, 8, 16, 32, 64
Var (Variable Chunk Deduplication)	Power (Anchor): 2^{21}, 2^{22}, 2^{23}, 2^{24}, 2^{25}
	Divider: 4, 8, 16, 32, 64

DataSet

File type	Attribution	Content	Size
CSV	Sequential file	Support append at the end of the file Support append at the front of the file	50M, 100M, 150M, 200M, 250M
LOG	Sequential file	Support append at the end of the file Support append at the front of the file	50M, 100M, 150M, 200M, 250M

PDF	Random file	Support append at the end of the file Support append at the front of the file	50M, 100M, 150M, 200M, 250M
ZIP	Random file	Support append file to increment file size	50M, 100M, 150M, 200M, 250M

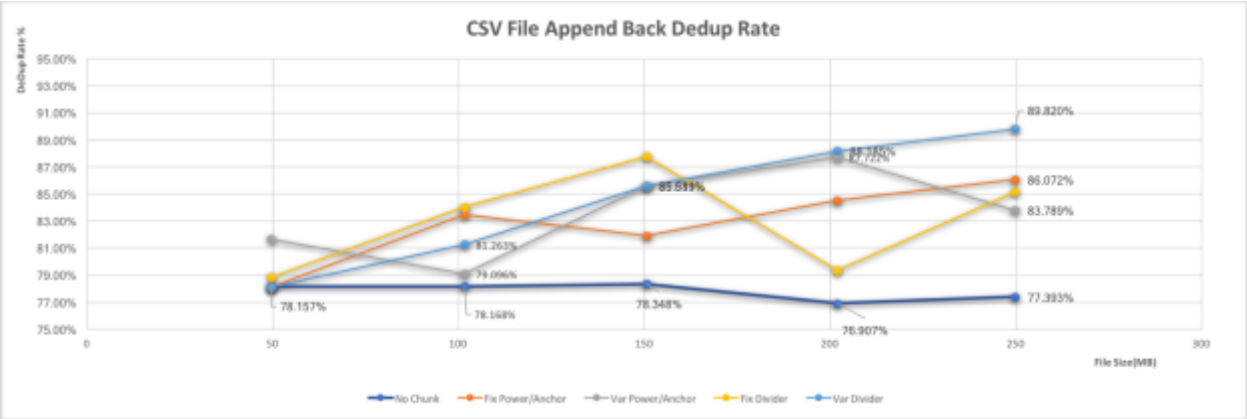
Testing Result and Evaluation

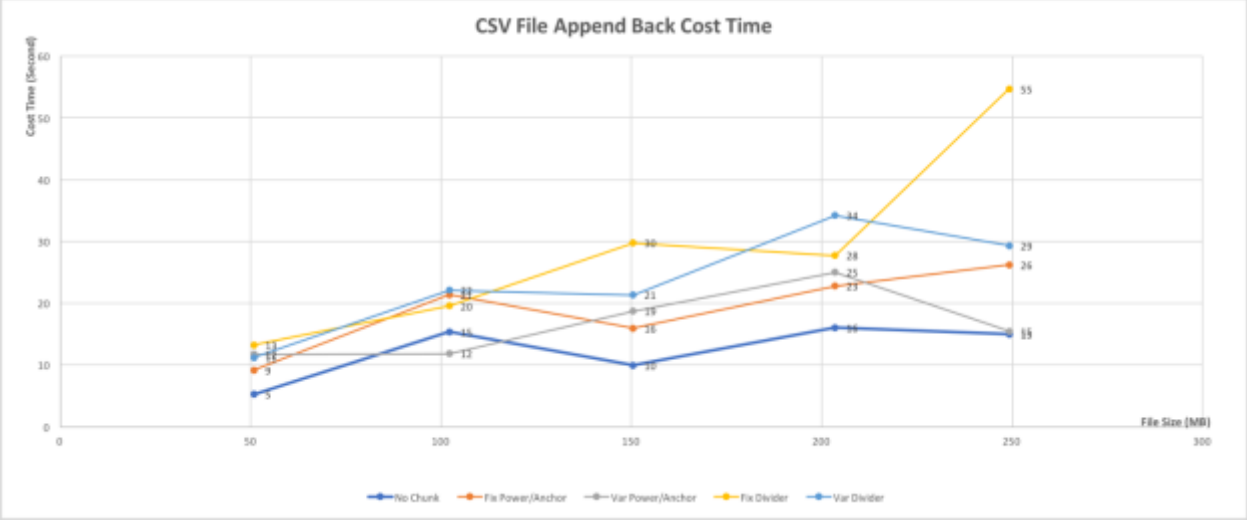
Entries (Series) name in the following deduplication ratio chart:
Nor: Normal, no chunks
Fix Power/Anchor: Average of Dedupe rate of fixed chunk size which compute from anchors
Variable Power/Anchor: Average of Dedupe rate of variable chunk size which compute from anchors
Fix Divider: Average of Dedupe rate of fixed chunk size which compute from dividers
Variable Divider: Average of Dedupe rate of variable chunk size which compute from dividers

Entries (Series) name in the following cost chart:
Nor: Normal, no chunks
Fix divider Time: Average of cost time of fixed chunk size which compute from anchors
Variable power/anchor Time: Average of cost time of variable chunk size which compute from anchors
Fix divider: Average of cost time of fixed chunk size which compute from dividers
Variable divider: Average of cost time of variable chunk size which compute from dividers

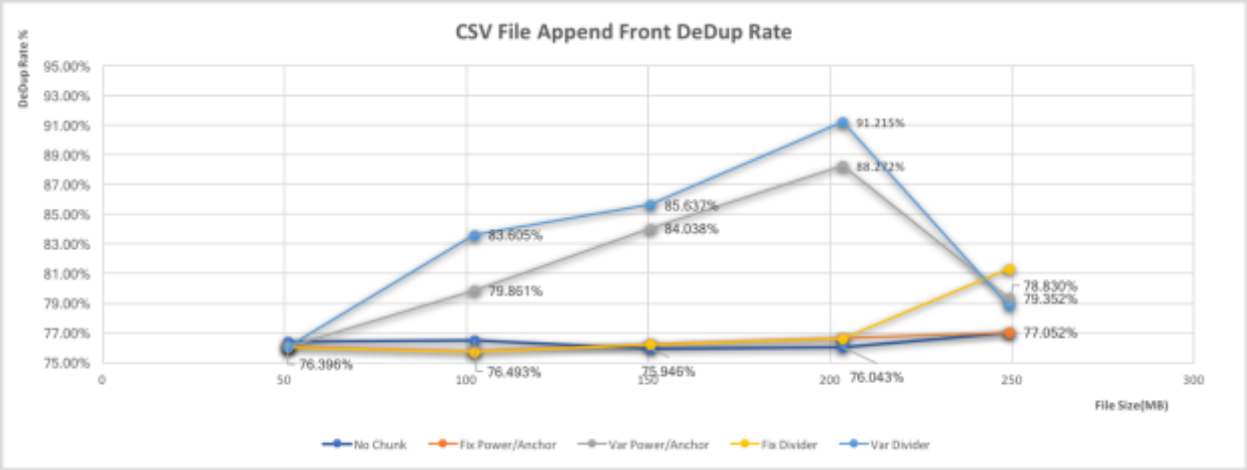
Experiment - CSV deduplication ratio and cost time

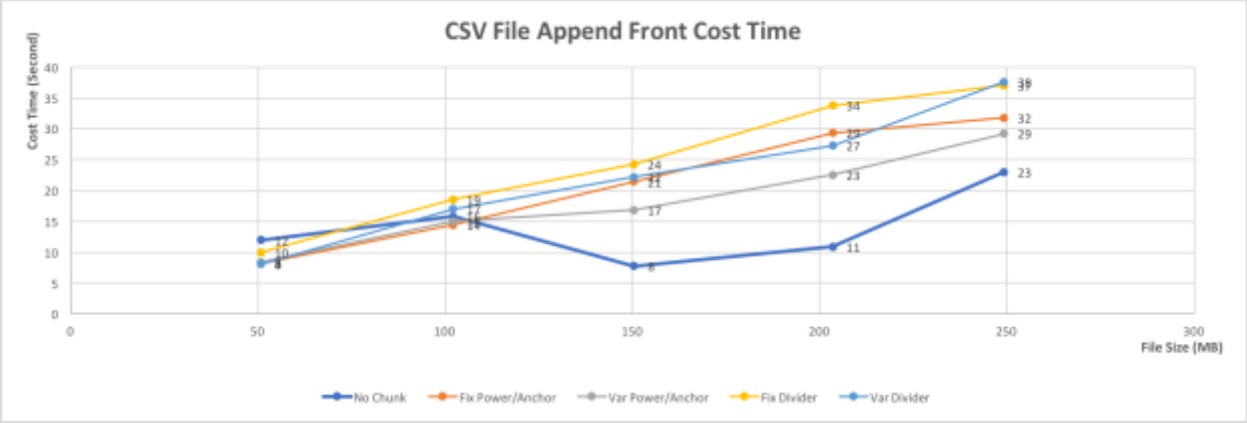
Append back scenario - deduplication ratio and cost time





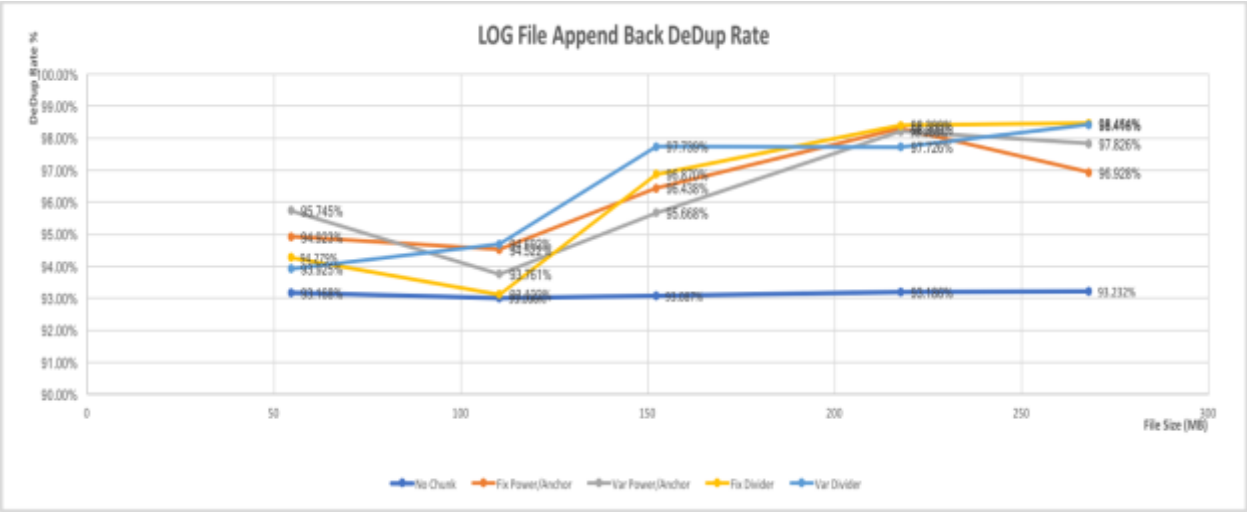
Append front scenario - deduplication ratio and cost time

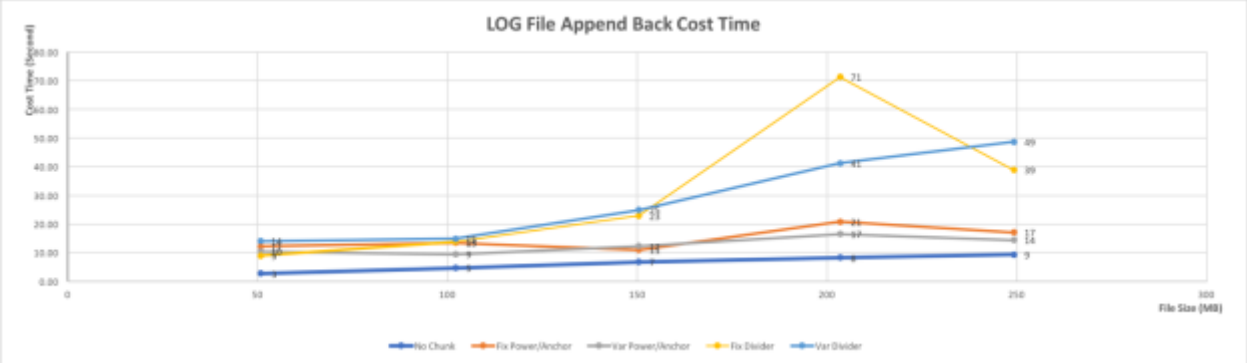




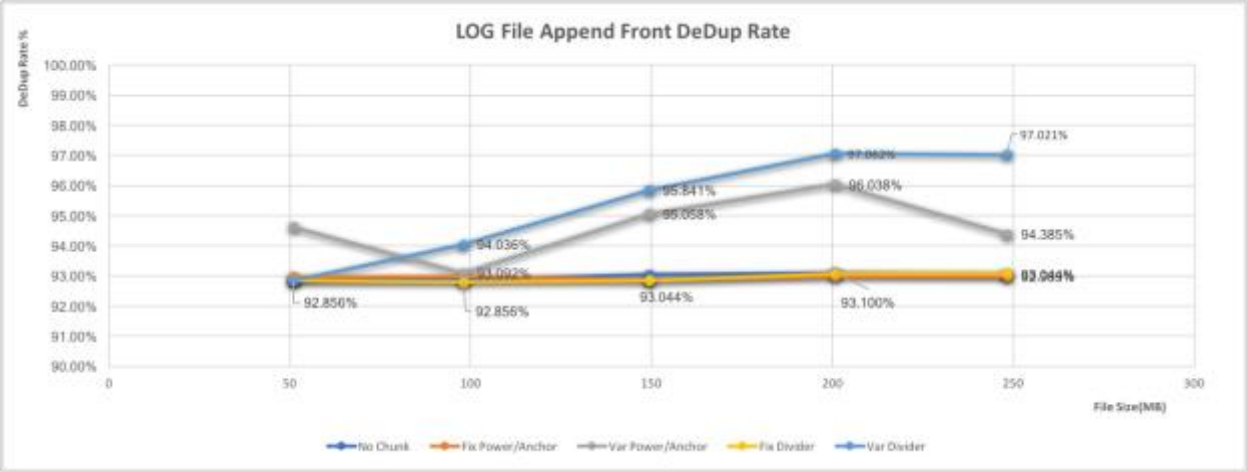
Experiment - LOG deduplication ratio and cost time

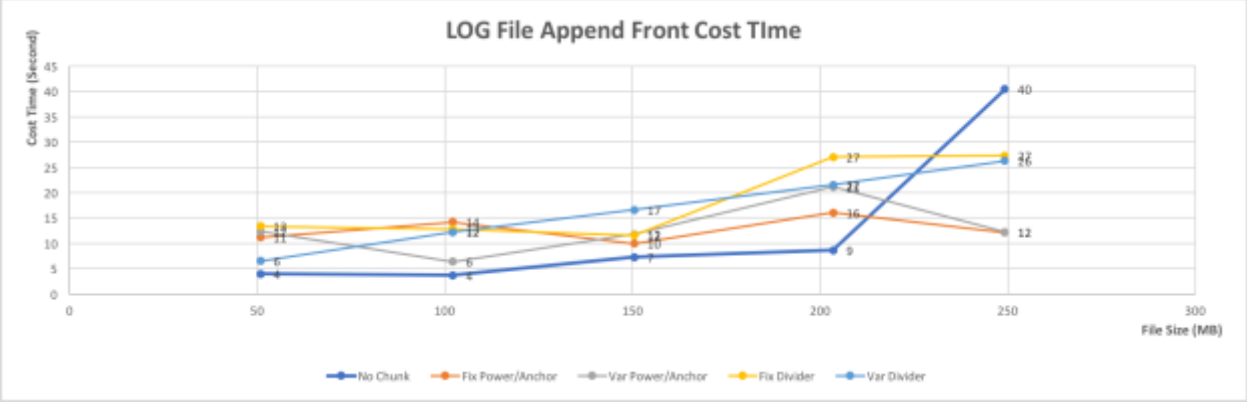
Append back scenario - deduplication ratio and cost time





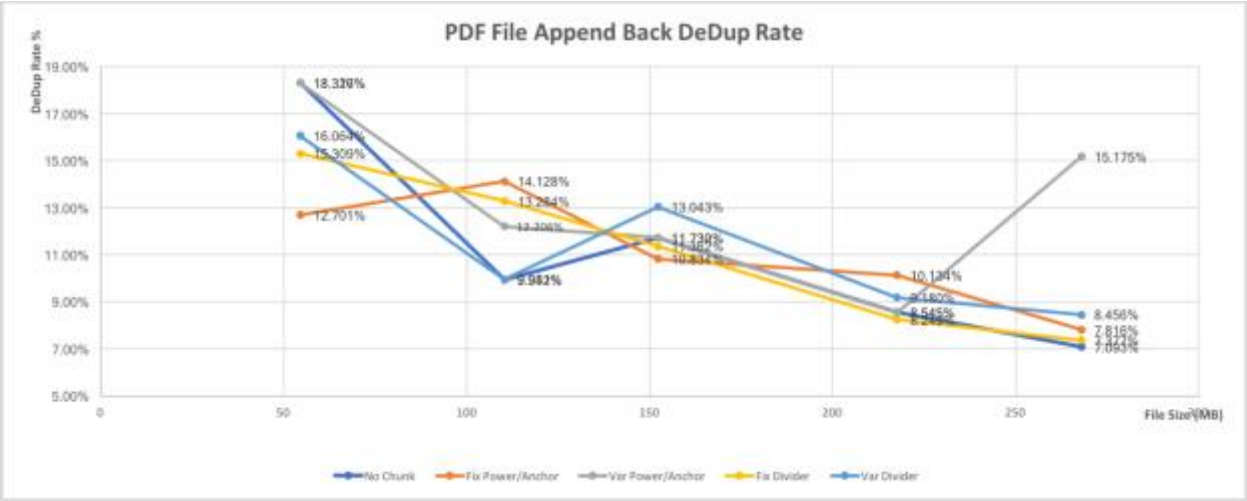
Append front scenario - deduplication ratio and cost time



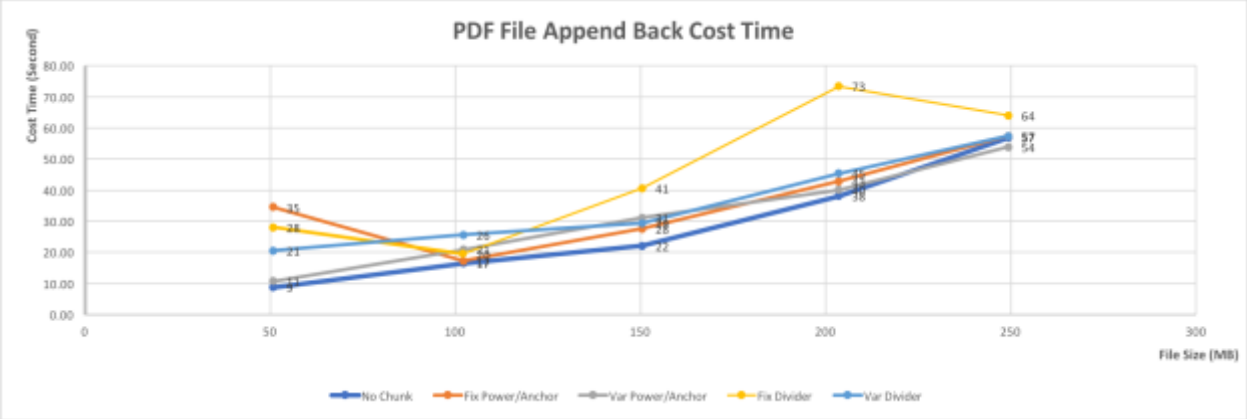


Experiment - PDF deduplication ratio and cost time

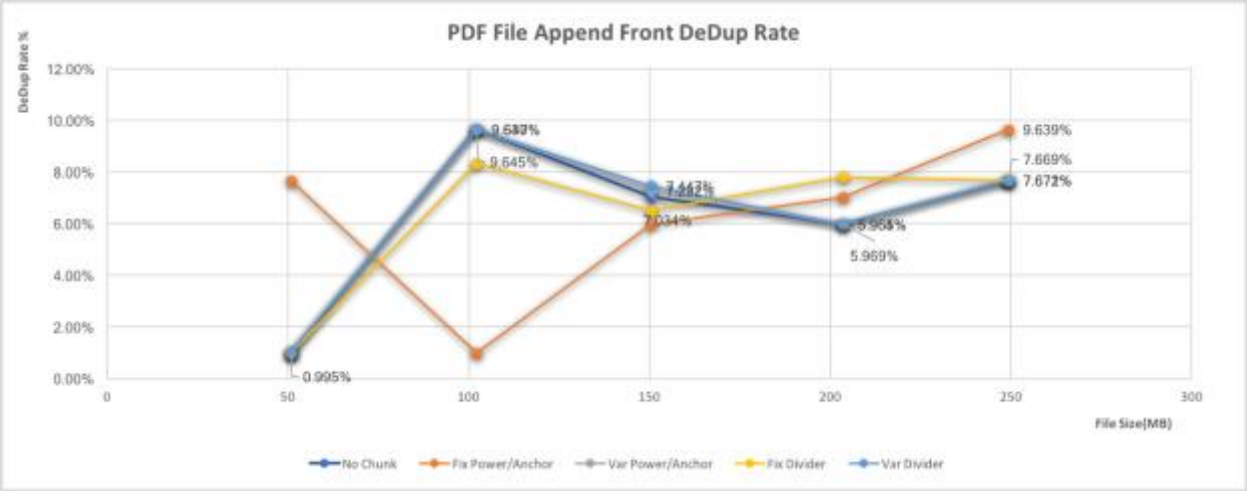
Append back scenario - deduplication ratio and cost time



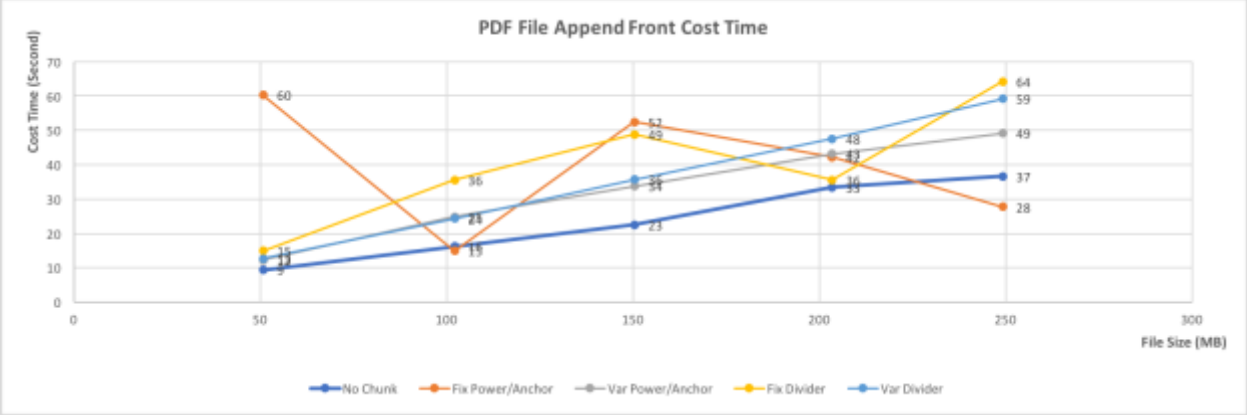
Generally, the deduplication ratio of chunk level deduplication is low. Furthermore, file level deduplication has the lowest. This result is reasonable, since the pdf format is similar to random file which is not readable.



Append front scenario - deduplication ratio and cost time



At appending front scenario, all the deduplication ratio are much lower than the appending back scenario. This is reasonable since the pattern is harder to detect.



Experiment - ZIP deduplication ratio and cost time

Increment file size scenario - deduplication ratio and cost time