Accelerating Data Deduplication by Exploiting Pipelining and Parallelism with Multicore or Manycore Processors

Wen Xia， Hong Jiang， Dan Feng， Lei Tian

Background and Challenges

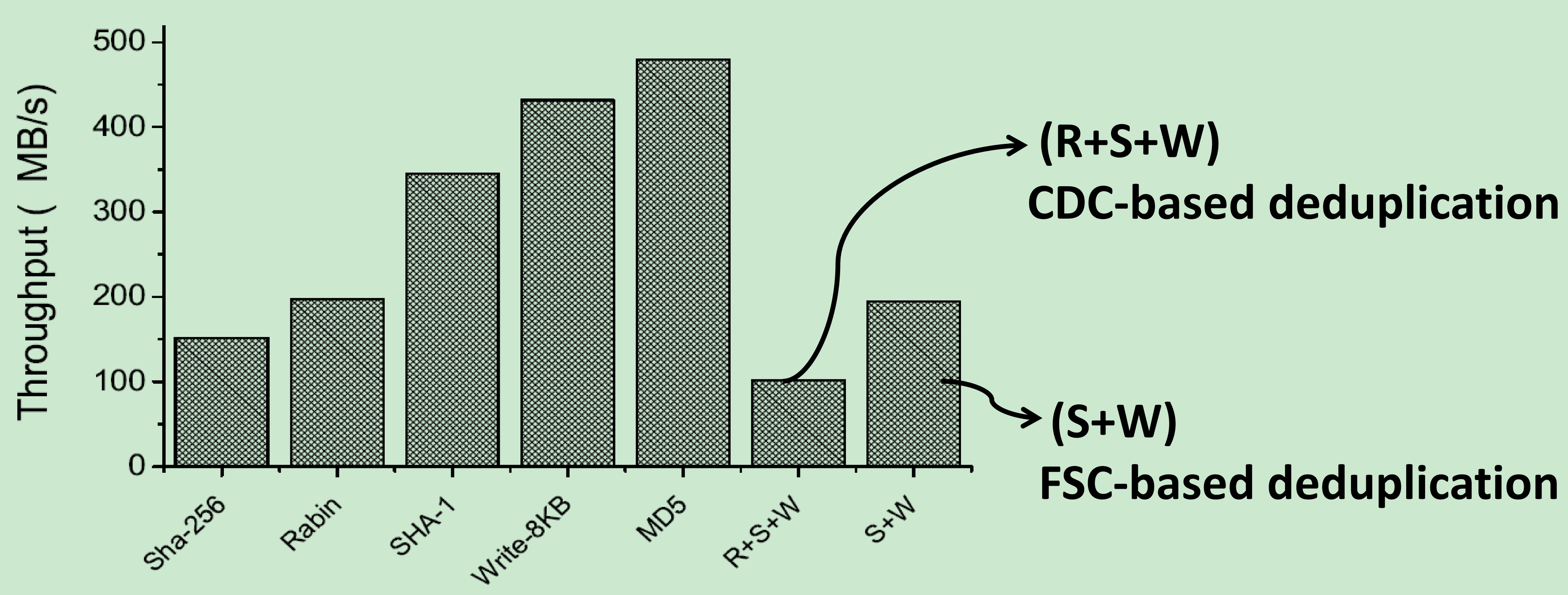
•Data deduplication

- Reduce storage space requirement by eliminating duplicate data
- Minimize the transmission of redundant data in storage systems

•Deduplication computation overheads

- Contend –Defined Chunking (Rabin)
- Fingerprinting (SHA1 or SHA256)

•Increasing compute resource with multicore or manycore



•Real world data deduplication

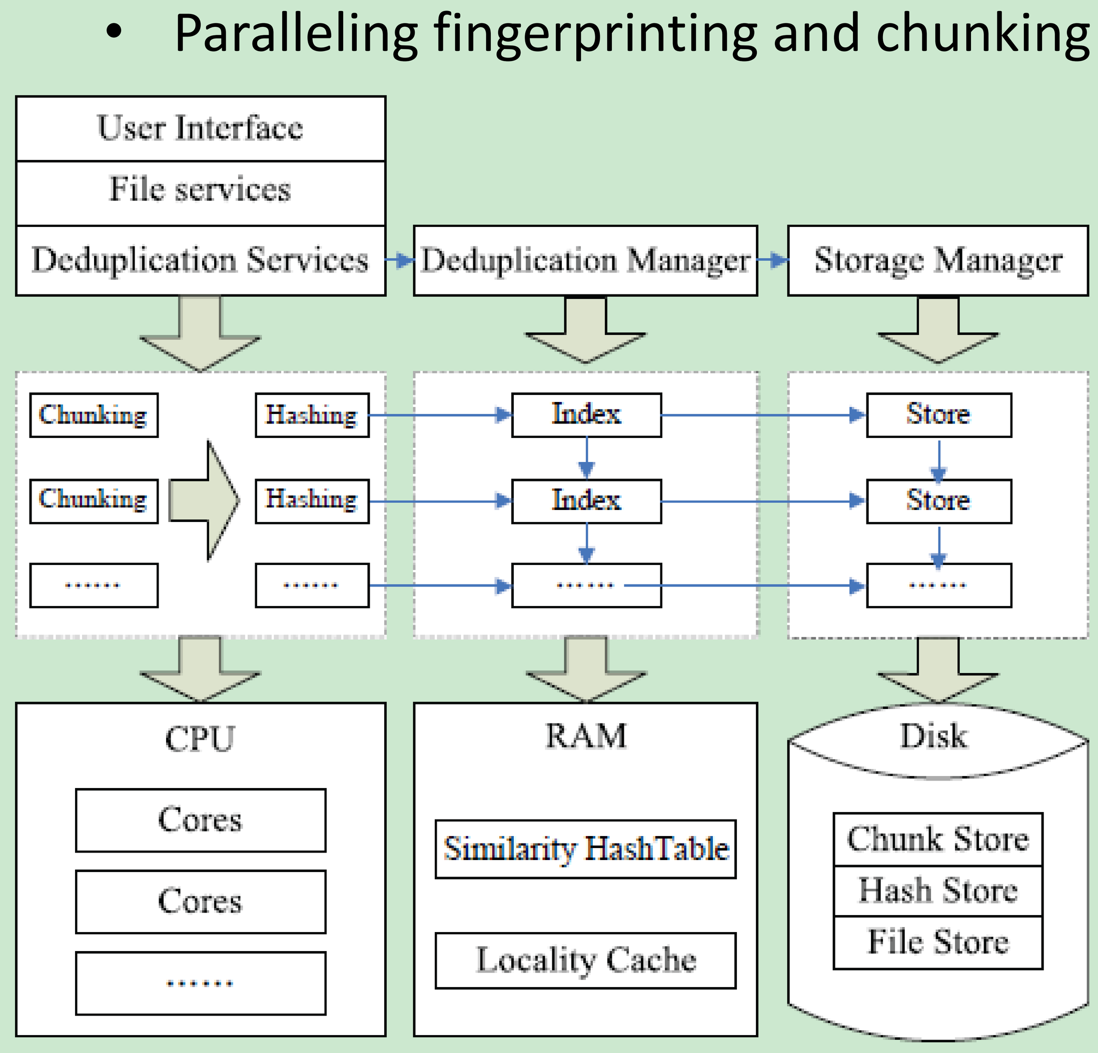
P-Dedupe Approaches

•Data deduplication process can be organized as:

- Data units (such as chunks and files)
- Functional units (i.e. *chunking, hashing, indexing and writing*)
- They are independent of one another

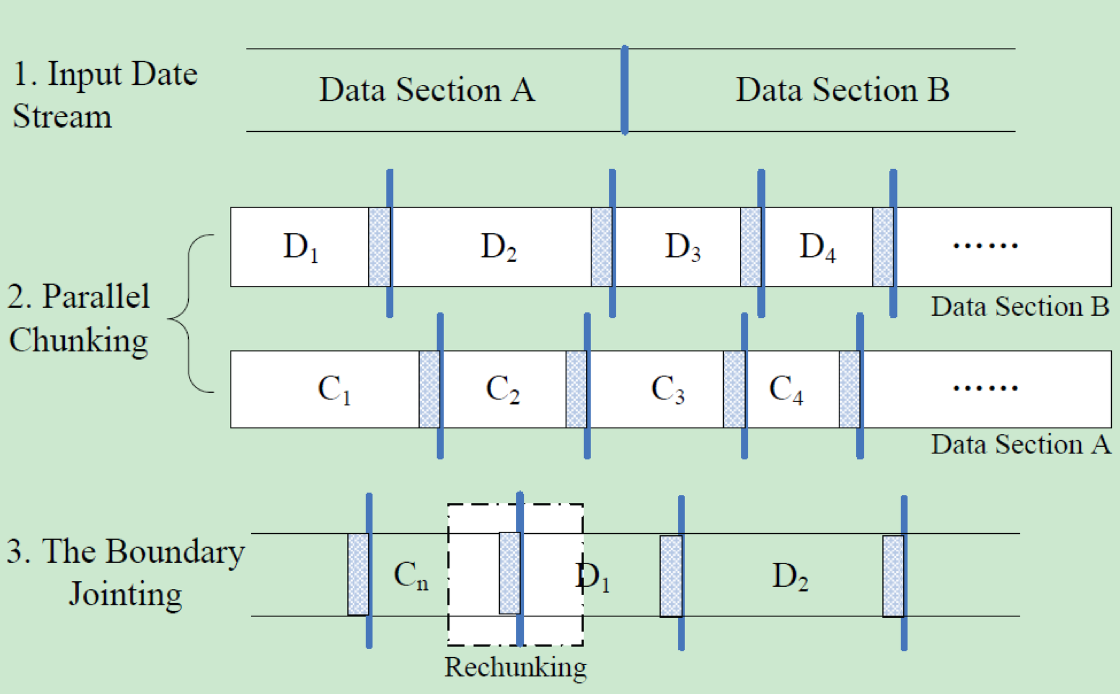
•Full exploitation of parallelism on data deduplication

- Pipelining of CDC based deduplication processes
- Paralleling fingerprinting and chunking



Deduplication pipeline
(S1) chunking ,
(S2) fingerprinting ,
(S3) indexing ,
(S4)writing chunk data and metadata.

•P-Dedupe system architecture



Only small bytes need to be re-chunked:
2 (slide window size)*

Or not re-chunking ?

•1. The parallel CDC algorithm runs with two threads
•2. The data stream is divided to Section A and Section B
•3. The boundaries of A and B need to be re-chunked

Observation and Motivations

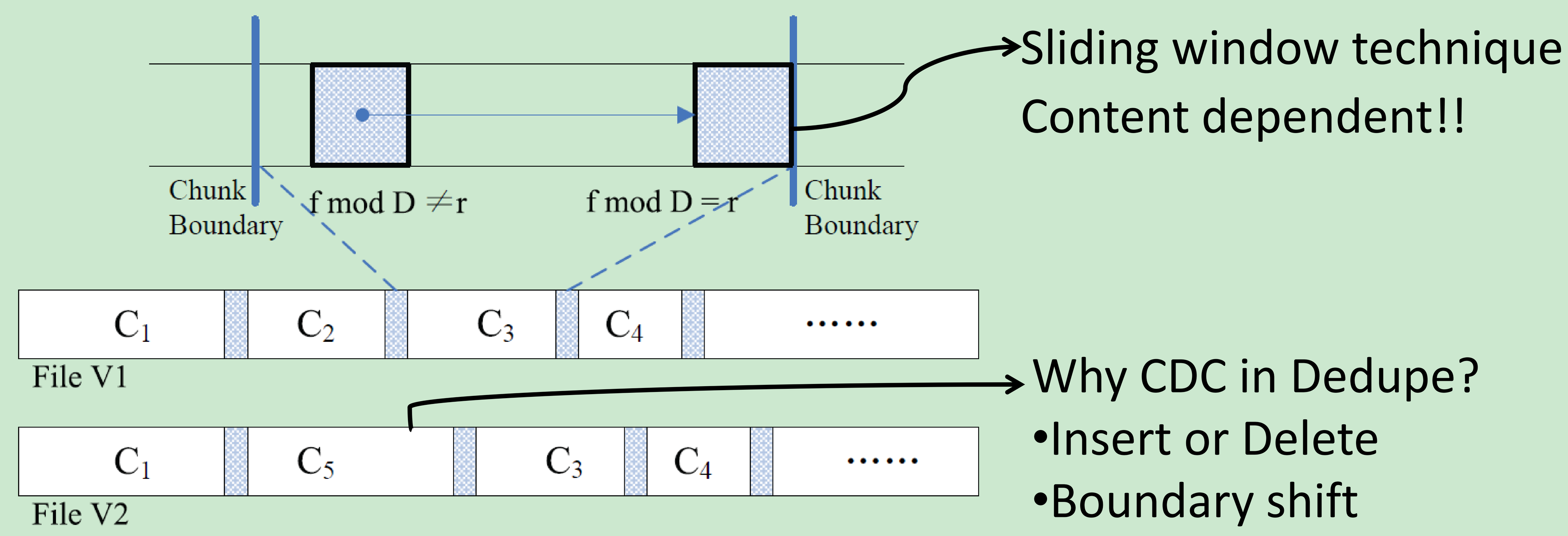
•Minimize the deduplication compute overheads

•Serial Dedupe $X_{put} = 1/(T_c + T_f + T_w/D)$

•Pipelining $X_{put} = 1/Max(T_c, T_f, T_w/D)$

•Parallelism $X_{put} = 1/Max(\frac{T_c}{N}, \frac{T_f}{N}, T_w/D)$

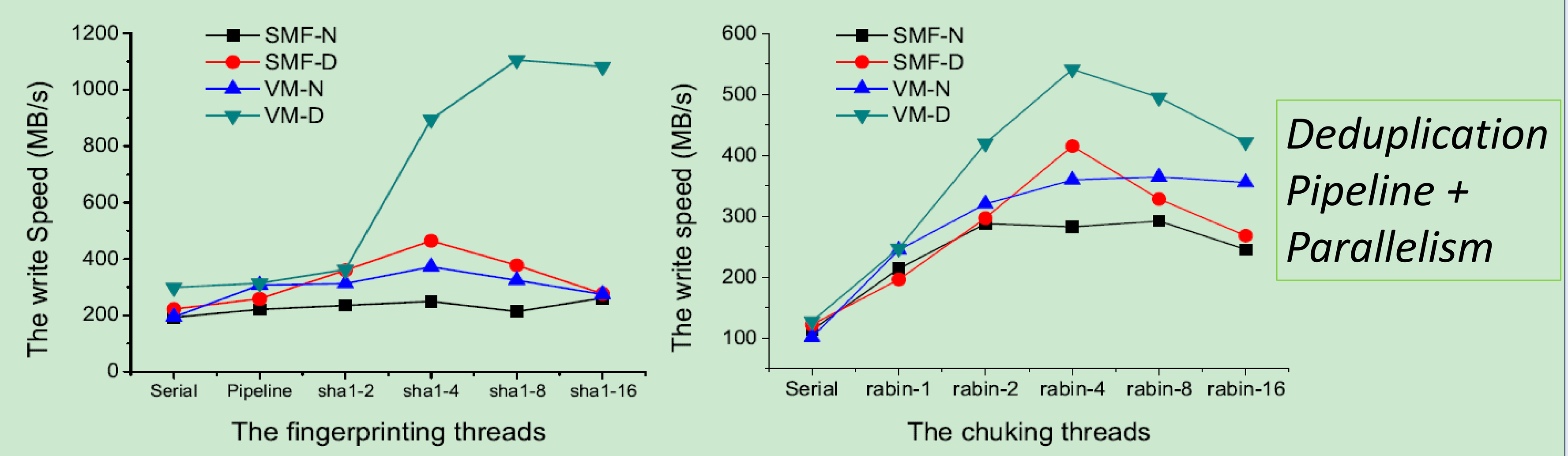
T_c: Chunking
T_f: Fingerprinting
T_w: Writing
D: Dedupe factor
N: Parallel threads



- **CDC based data deduplication.**

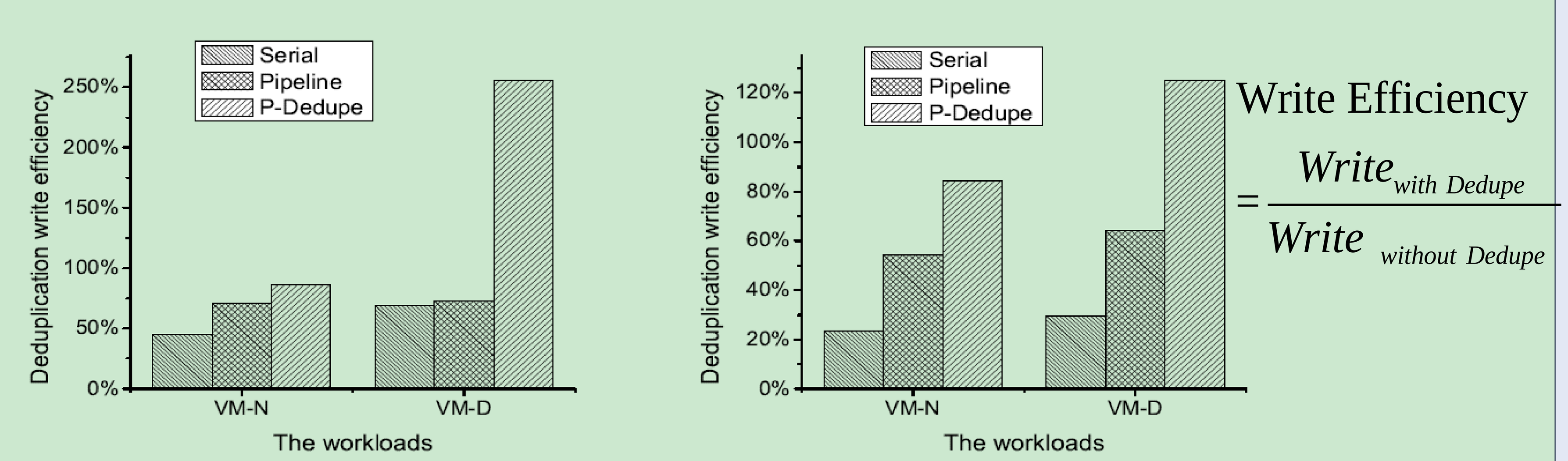
Preliminary Results

•Evaluate P-Dedupe on an Intel quad-core and eight-thread CPU



(a) Fix-Size Chunking. (b) Content Defined Chunking.

•Deduplication based writing throughput



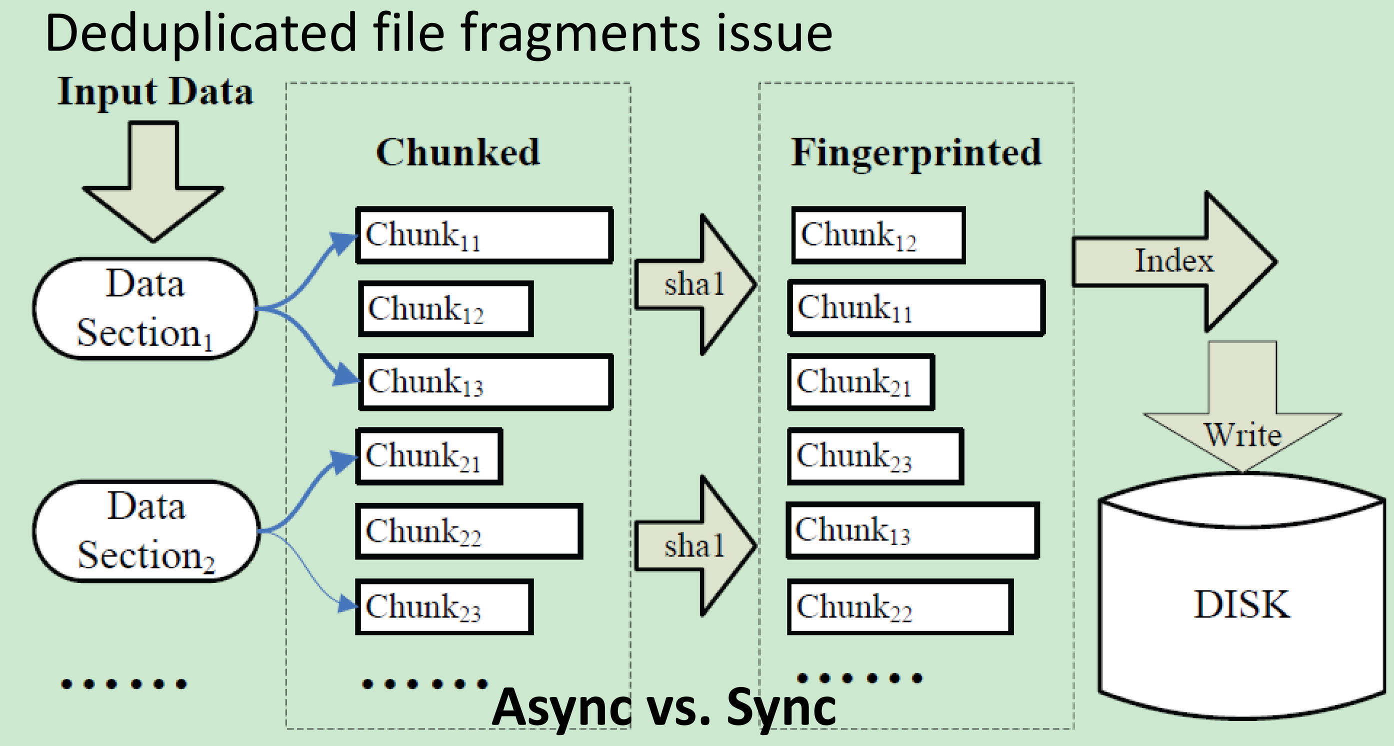
(a) Fix-Size Chunking. (b) Content Defined Chunking.

•Write efficiency among Serial , Pipeline and P-Dedupe

Ongoing Work

•Boost the performance with increasing numbers of cores

- Memory and cache management
- Choices of section size and chunk size
- Asynchronization or synchronization of parallelism
- Deduplicated file fragments issue



This work was supported by Chinese 973 Program of under Grant No.2011CB302301, and the US NSF under Grants CNS-1116606, CNS-1016609, IIS-0916859 and CCF-0937993.

@ 2012 Huazhong University Science and Technology and University of Nebraska-Lincoln. All right reserved.