Skipping Repetitive Dirty Pages for Restore-Express Incremental Checkpointing

Purushottam Sigdel and Nian-Feng Tzeng §

Center for Advanced Computer Studies, University of Louisiana, Lafayette

1. Abstract and Motivations

- Utilizing unused cores, REX reduces checkpoint data volume drastically via two insights:
  - Majority of modified pages ***exists*** in previous checkpoint files
  - Abundant data patterns stay ***unchanged*** over successive checkpoints.
- REX quickens restore through only last copy of every dirty page to create full system states.

Observed insights

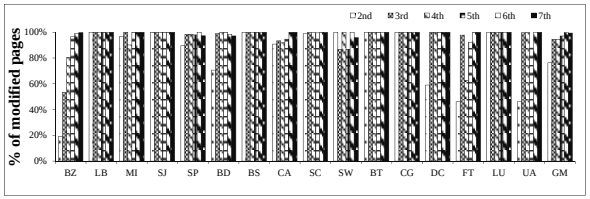

Figure 1: Percentage of pages in j^{th} incremental checkpoint file that had been modified in any prior (i^{th}, with $i < j$) checkpoint.

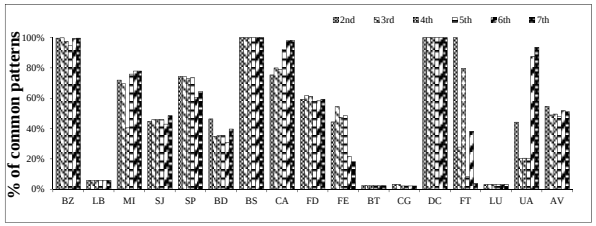

Figure 2: Data patterns in a modified page at (j-1)th incremental checkpoint repeated in corresponding page at j^{th} checkpoint.

- Vast majority of modified pages (i.e., ≥ 87%) also exist in prior checkpoints (taken by AIC [1]).
- Modified page has > 50% of its 32-byte data patterns same as those in prior checkpoint page.
- Vast majority of data patterns (over 80%) for benchmarks of BZ, BS, CA, DC stays unchanged over successive incremental checkpoints.

2. Proposed Approach

a. From checkpoint files, Page Coalescer generates a list of distinct pages, as recorded in C-List.

b. Deduplicating contents, the page-aware deduplication unit produces Pointer File (PF), Reference Pattern Sequence (RPS) File, and P-Table.

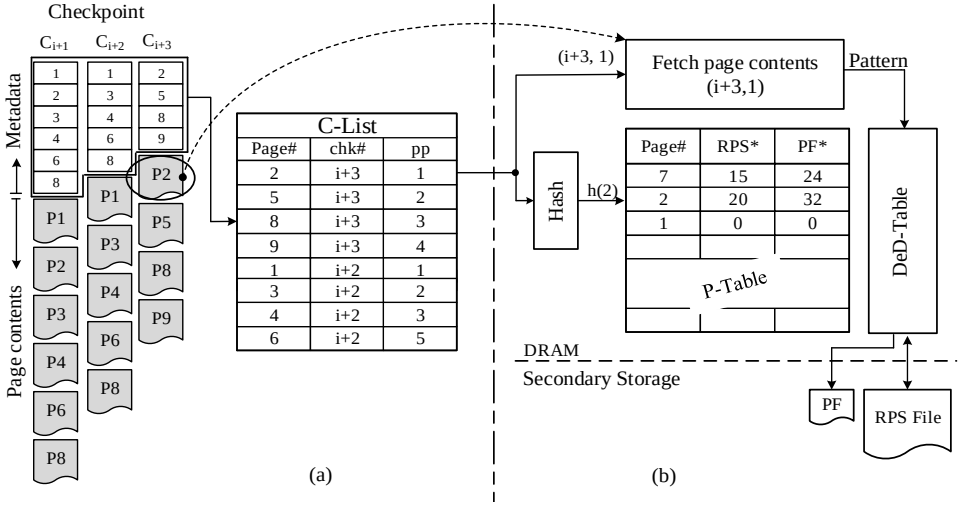

Figure 3: Overall block diagram of REX.

3. Summary

Result Highlights

- Restore latency under REX is shortened by a factor of 7.4× and 1.7×, in comparison to those of incremental checkpointing (IC) and full checkpointing (FC), respectively.
- REX reduces total volume of data writes to remote storage by 2.8× and 3.3×, when compared to FC and IC, respectively.
- REX's de-referencing approach yields the best overall decompression rate of 400 MB/s (i.e., 20.9× and 4.1× faster than those of bzip2 [2] & Xdelta3 [3], respectively), for express restores.
- REX's page-aware compression module gives rise to a comparable compression ratio of 2.4× (versus 2.6× by bzip2 [2] and 3.0× by Xdelta3 [3]).

Result Details

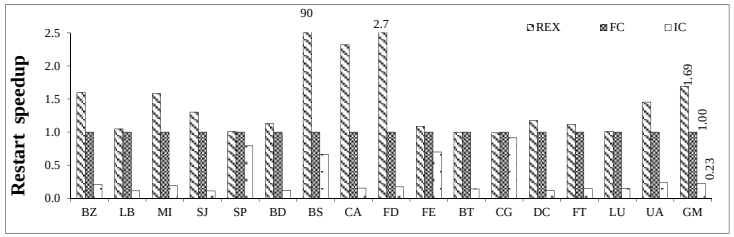

Figure 4: Restore speedup, taking the restore speed under FC as base.

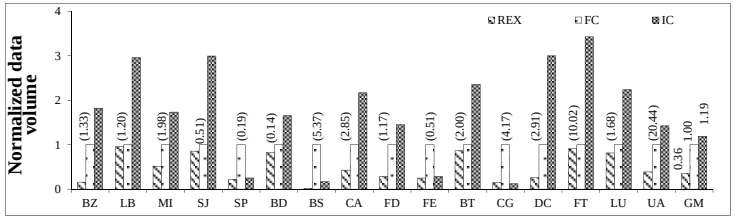

Figure 5: Total data volume, normalized against that under FC (in GB).

[1] I. Jangjaimon and N.-F. Tzeng, "Adaptive incremental checkpointing via delta compression for networked multicore systems," *Proc. of IEEE Int'l Parallel & Distributed Processing Symp.* (*IPDPS*), pp. 7-18, May 2013.

[2] J. Seward, "bzip2 and libbzip2, version 1.0.5: A program and library for data compression," 2007, URL — http://www.bzip.org/1.0.5/bzip2-manual-1.0.5.pdf.

[3] J. MacDonald, "File system support for delta compression," *M.S. Thesis*, Univ. of California, Berkeley, May 2000.

§ Support in part by NSF under **CNS-1527051**.