ANALYSIS OF N-BLOCK CACHING STRATEGIES
Rajashree Krishnadevamaiya | rk4509@rit.edu
Advisor: Dr. Hans-Peter Bischof

Goal
Analyze the performance of caching algorithms in a distributed network for N block replacement caching strategies. Compare cache replacement algorithms for varied size of data block in input traces.

Background
Distributed caching allows for sharing of cached data between clients or devices. Distributed caching provides better consistency and reliability compared to in-memory caches.

Traditional cache replacement strategies replace one block at a time, we study the effect of replacing N blocks at a time. We study the behavior of First In First Out, Least Recently Used, Least Frequently Used, and NChance forwarding replacement strategies.

The metrics used for evaluation are memory access time, disk access rate, cache hit rate, and cache miss rate.

System Design & Metrics
A cache simulator is designed using Java. Matlab is used for data manipulation and visualization.

Simulator Components include clients, content manager, server, main memory/disk, and a driver.

Simulator Architecture

Results & Discussions
Experimental Setup
Data: Input trace files with data requests that has 15% of the total requests as set of blocks.
Visualizations for cache performance of all algorithms for single block replacement v/s N-Block replacement.

Access Time
For small cache sizes, N-block replacement reduces the memory access time. As the cache size increases, the behavior is similar to single block replacement.

Miss Rate
For small cache sizes, N-block replacement reduces the miss rate. As the cache size increases, N-block replacement behaves similar to single block replacement. Large cache sizes reduces the number of replacements for both single and N-block replacements.

Conclusions
N-Block replacement is suitable if input data requests contains blocks of data that are always needed together.
N-Block replacement improves the performance of cache for small cache sizes. For large cache sizes, N-Block replacement behaves like single block replacement.
N-Block replacement improves the performance of cache for low clients and small cache sizes. For higher number of clients, N-Block replacement behaves like single block replacement.
There is no change in performance of cache for varied server cache size, both for single and N-block replacement.
Of the four algorithms evaluated, N-Chance replacement algorithm performs better than the other three.

Reference