Distributed Chunk-Based Data Processing System

KEY FINDINGS: The Underlying Dynamics

Bottleneck: Fixed Network Overhead (RTT/Session Setup) dominates processing time for small datasets (<10K rows).

Sweet Spot: 1.66x Speedup achieved for datasets between 100K and 1M rows. (might be Optimal for medium data: processing savings justify network cost.)

System Limits ("Fails When"):

- Too Small (<10K): Connection setup cost (3-6ms) outweighs data retrieval savings.
- Too Large (10M / 1.2GB): Dataset exceeds the 300MB Cache Limit.

Insight: Caching needs "just right" data: large enough to be worth the network trip, small enough to stay in cache.

Number of Rows	Cold Start (ms)	Warm Cache (ms)	Speedup (Cold start / Warm Cache)	Throughput (MB/s)
1K	125	98	1.28x	9.44
10K	132	118	1.12x	9.55
100K	1,257	840	1.50x	10.74
200K	2,856	1,904	1.50x	10.36
500K	8,147	5,092	1.60x	9.77
1M	12,661	8,203	1.66x	9.98
10M	95,059	87,736	1.08x	12.30

Cold Start: Includes Disk I/O, parsing, and memory allocation. Warm Cache: Direct streaming from pre-allocated RAM (Zero-Copy).

Tested on: *2 Windows/WSL + Ethernet | 8GB RAM (on each PC) having*
Cross-machine RTT: *~3-4 ms*
Tools for measurements: *chrono, gRPC timestamps, htop, Wireshark*

Summary

A 6-node hierarchical pipeline (Leader to Team Leaders to Workers) demonstrates that session-aware caching reduces end-to-end latency by 1.66x for mid-to-large workloads (100K–1M rows). The system scales linearly up to 1M rows (122 MB), fitting entirely within the allocated 300MB cache. However, at 10M rows (1.2 GB), the dataset exceeds cache capacity, triggering LRU eviction and reverting performance to baseline I/O speeds.