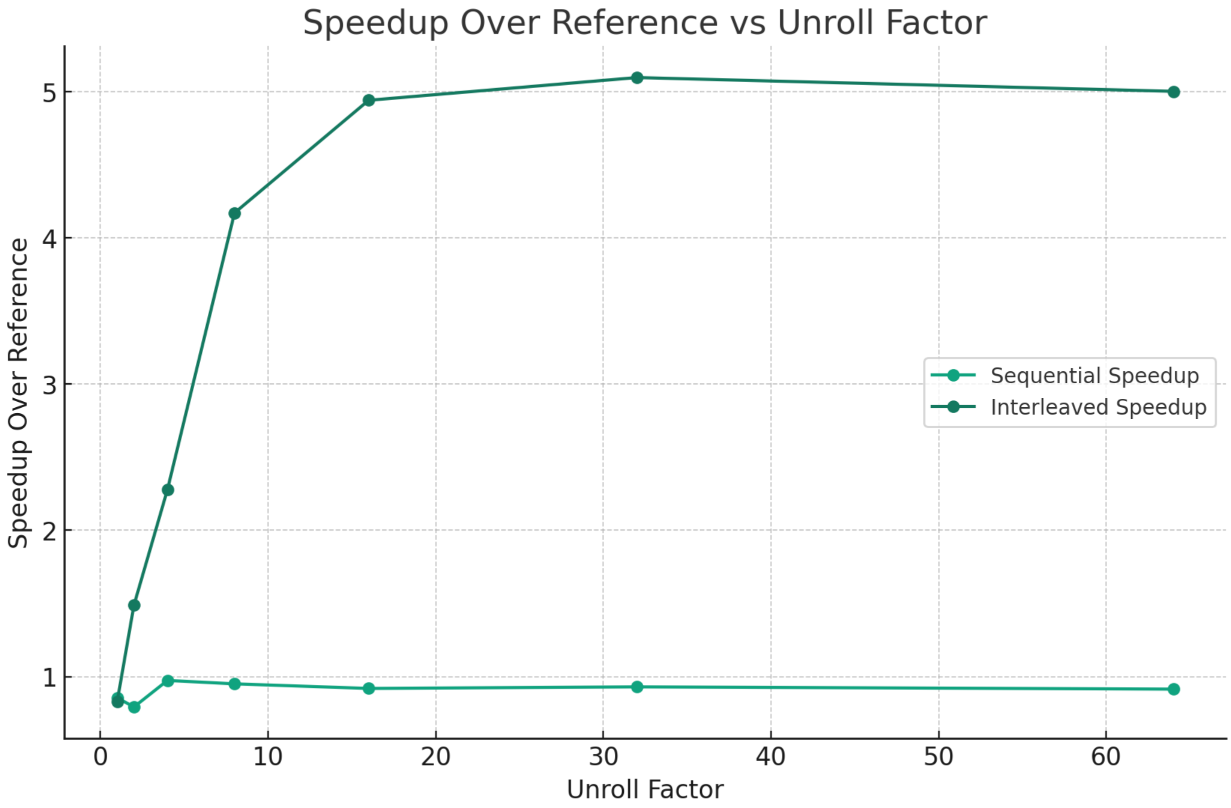
Loops with Independent Iterations

Chain Length = 64

Unroll Factor	Sequential Speedup Over Reference	Interleaved Speedup Over Reference
1	0.855342	0.827655
2	0.791946	1.48812
4	0.973484	2.27922
8	0.950103	4.17152
16	0.918697	4.94203
32	0.929575	5.09749
64	0.913995	5.00322

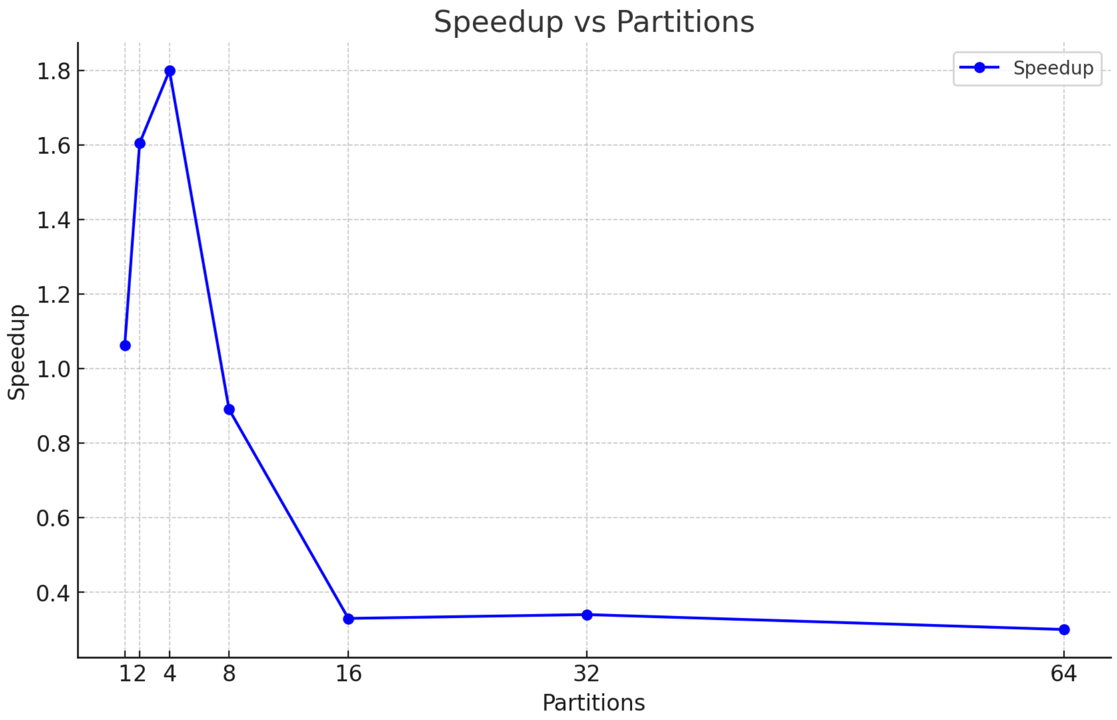


The interleaved loop consistently exhibits an increasing speedup over the reference loop. This pattern is particularly noticeable at an unroll factor of 64, where the speedup reaches its peak at 5.00322. In contrast, the performance of the sequential loop relative to the reference loop is more variable. It generally remains below 1, indicating that its performance is slower compared to the reference loop. This fluctuation suggests that the sequential loop's efficiency is less consistent and more sensitive to changes in the unroll factor.

The data strongly suggests that loop unrolling, when coupled with interleaving, significantly enhances performance as the unroll factor increases, especially when compared to the reference loop. This improvement is most pronounced at higher unroll factors. Although the sequential loop does not achieve the same level of speedup as the interleaved loop, it still displays varying degrees of performance enhancement under different unroll factors. These findings underline the effectiveness of loop unrolling and interleaving as methods to optimize loop performance in computational tasks.

Reduction loops

Partitions	New Loop Time (s)	Reference Loop Time (s)	Speedup
1	0.0599185	0.0636093	1.0616
2	0.0317657	0.050964	1.60437
4	0.0298001	0.0536121	1.79906
8	0.0603709	0.0537333	0.890052
16	0.158215	0.0521214	0.329434
32	0.159813	0.0543007	0.339775
64	0.209783	0.0628822	0.299749



The speedup achieved through loop partitioning varies significantly with the number of partitions. The most notable speedup, recorded at 1.79906, is observed when the loop is divided into 4 partitions. However, as the number of partitions increases beyond this optimal point, there is a noticeable decrease in speedup, signaling a decline in performance. This downward trend becomes especially pronounced with 64 partitions, where the speedup drops to its lowest at 0.299749. This suggests that excessive partitioning can adversely affect performance, leading to significant degradation rather than improvement.

The data suggests that while loop partitioning can enhance performance, it does so only up to a specific threshold. Beyond this optimal point, which in this test appears to be 4 partitions, additional partitioning becomes counterproductive, resulting in decreased speedup. This phenomenon indicates a clear trade-off between achieving parallelism through partitioning and the associated overhead that comes with it.