Work Progress

kNN Search with Parallel Incremental Query Answering

Jaouhara Chanchaf

Tuesday Jan 19th, 2023

1. Summary (01/02/2023 - 01/18/2023)

Done:

AI 1 - AI 9	Khasif and PEXESO performance evaluation, Kashif performance on normalized vs unnormalized data, etc.
AI 10	Why PEXESO out performed Kashif (in query time)
AI 11	Why Kashif recall is low when considering PEXESO as a ground truth.

In progress: AI 4: Run joinable table search using JOSIE and use results to measure Kashif recall.

2. Kashif performance on normalized data

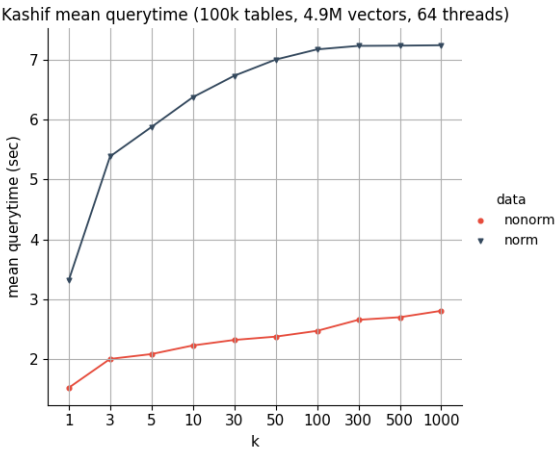

Figure 1: Kashif mean query time

2. Kashif performance on normalized data

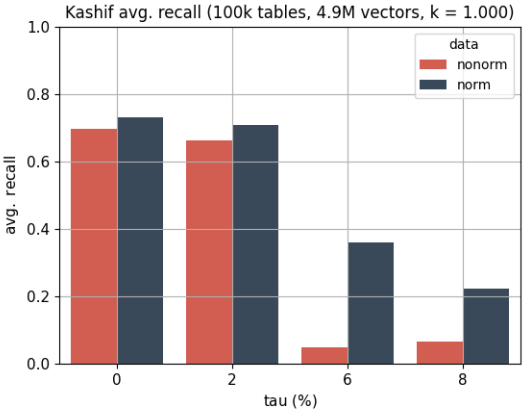

Figure 2: Kashif average recall

- Kashif has poor performance on normalized data.
- One possible explanation is that the splitting policies become ineffective when all data vectors are normalized.
- Kashif recall improves significantly when performing search over normalized data vectors.

3. Mean query time Kashif vs PEXESO (AI 1)

Why I thought PEXESO outperformed Kashif (in query time):

- I combined all query columns in one grid structure.
- I measured time for block() and verify() and not for query index building and quick browsing.
- PEXESO performance could be improved if better pivots are chosen. In all conducted experiments PEXESO does not employ any of the matching lemmas (lemma 2, 5 and 6).

3. Mean query time Kashif vs PEXESO (AI 1)

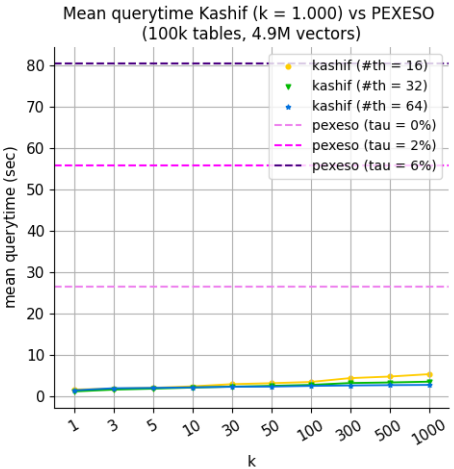

Figure 3: Mean query time PEXESO vs Kashif

- **Dataset:** 100k tables, 4.9 M vectors.
- **Query:** 5 Queries of size 10.
- **Kashif settings:** k = 1000, #threads = {16, 32, 64}
- **PEXESO settings:** T = 1%, τ = {0%, 2%, 6%}

3. Mean query time Kashif vs PEXESO (AI 1)

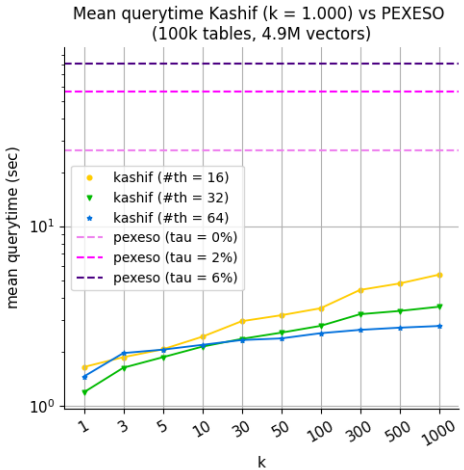

Figure 4: Mean query time PEXESO vs Kashif

- **Query:** 5 Query columns of size 100.
- **Kashif settings:** k = 1000, #threads = {16, 32, 64}
- **PEXESO settings:** T = 1%, τ = {0%, 2%, 6%}

3. Mean query time Kashif vs PEXESO (AI 1)

- Kashif (#th = 64) is 10x faster than PEXESO in all three settings.
- Note that PEXESO performance could be improved if better pivots are chosen. In all conducted experiments PEXESO does not employ any of the matching lemmas (lemma 2, 5 and 6).
- What would happen if we increase k to 10.000?

3. Mean query time Kashif vs PEXESO (AI 1)

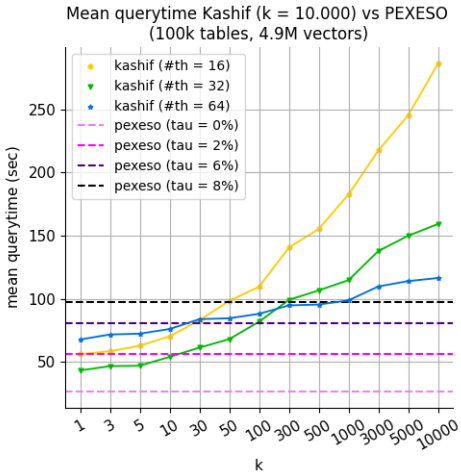

Figure 5: Mean query time PEXESO vs Kashif

- Kashif (#th = 64) takes less time to answer the query when $k < 30$ compared to PEXESO (tau = 6%) and $k < 1000$ compared to PEXESO (tau = 8%).

3. Mean query time Kashif vs PEXESO (AI 1)

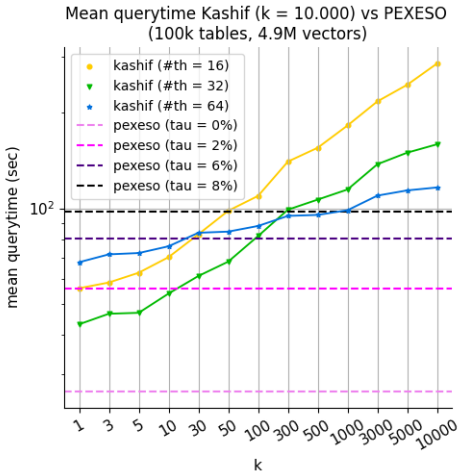

Figure 6: Mean query time PEXESO vs Kashif

- Kashif (#th = 64) takes less time to answer the query when $k < 30$ compared to PEXESO (tau = 6%) and $k < 1000$ compared to PEXESO (tau = 8%).

4. Kashif recall (AI 8)

We consider PEXESO as our ground truth and measure Kashif recall in three settings:

- **Mode 0:** Return all KNN results.
- **Mode 1:** Only return NNs of the same distance to the query vector.
- **Mode 2:** Return results of the same distance and extra results retrieved in the last increment.

4. Kashif recall (AI 8)

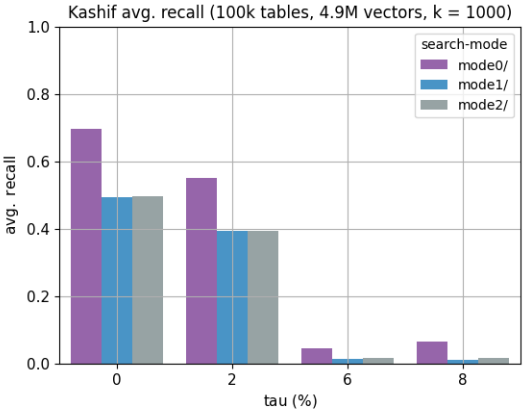

Figure 7: Kashif average recall

- **Query:** 5 Query columns of size 100.
- **Kashif settings:** k = 1000, #threads = 64
- **Ground truth:** PEXESO(P = 3, m = 4, T = 1%, τ = [0 - 8]%)

- Kashif recall decreases as we increase τ. This is because as τ increases PEXESO retrieves more results. One possible explanation is that the k value in Kashif is too small or some of the results returned by PEXESO are irrelevant.

4. Kashif recall (AI 8)

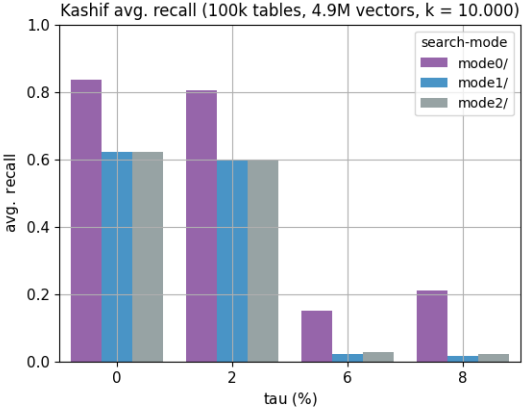

Figure 8: Kashif average recall

- **Query:** 5 Query columns of size 100.
- **Kashif settings:** k = 1000, #threads = 64
- **Ground truth:** PEXESO(P = 3, m = 4, T = 1%, τ = [0 - 8]%)

- Increasing k to 10.000 increases the average recall.

5. PEXESO impact of pivot selection on data filtering (AI 1, AI 2)

- PEXESO pivot selection algorithm has an impact of its filtering power, good pivots should map the original vectors and make them scattered in the pivot space. This allows the query region to cover less vectors (hence more vectors would be filtered)
- To evaluate the pivot vectors quality we measure the (%) of non empty leaf cells in PEXESO's leaf level.

5. PEXESO impact of pivot selection on data filtering (AI 1, AI 2)

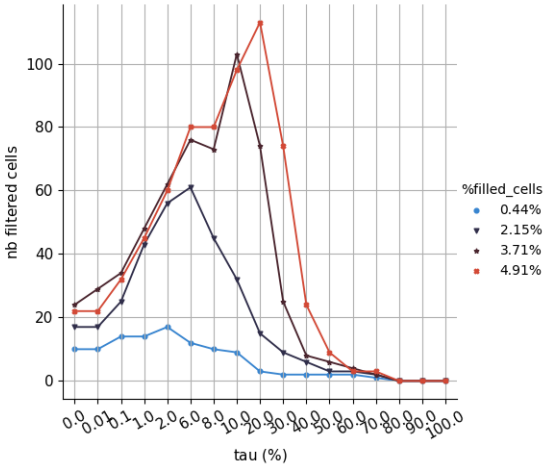

Figure 9: PEXESO: impact of τ and (%) of non empty leaf cells of the number of filtered cells

- **Dataset:** 10 Tables, 833 vectors.
- **Query:** 1 Query column of size 1.

5. PEXESO impact of pivot selection on data filtering (AI 1, AI 2)

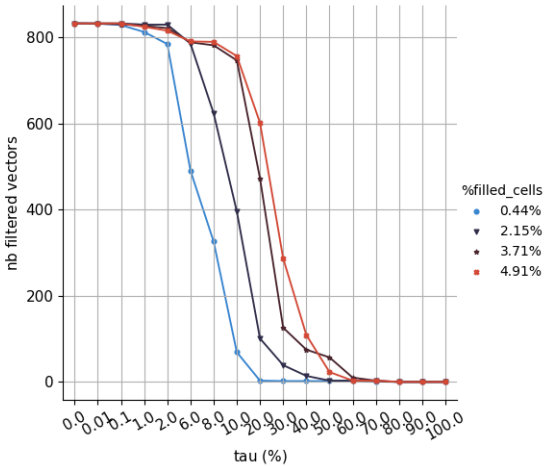

Figure 10: PEXESO: impact of τ and (%) of non empty leaf cells of the number of filtered vectors

- **Dataset:** 10 Tables, 833 vectors.
- **Query:** 1 Query column of size 1.

5. PEXESO impact of pivot selection on data filtering (AI 1, AI 2)

- When τ is very small ($\tau \leq 0.01$), the filtering power of PEXESO is at its highest. Consequently, cell filtering happens at the top levels where the number of cells is very small [1] which explains the low number of filtered cells.
- As we continue to increase τ the filtering happens at lower levels where the number of cells is much higher which explains the increase in the number of filtered cells.
- On the long run and since the number of levels m is finite the number of filtered cells drops to zero for $\tau > 20\%$.
- The higher the (%) of non empty cells the higher the number of filtered cells/vectors.

[1] The number of cells in a level $i = 2^{|P|*i}$, $|P|$ is the number of pivot vectors.

5. PEXESO impact of pivot selection on data matching

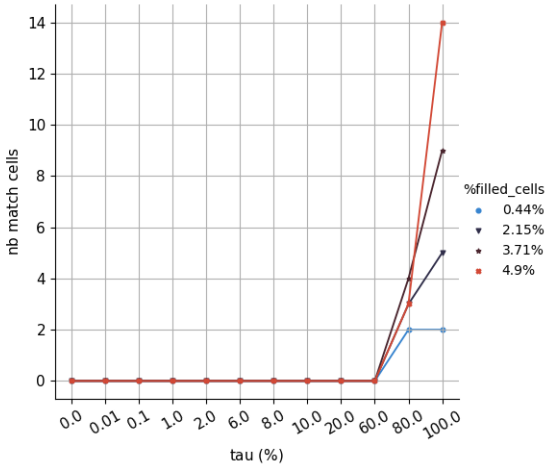

Figure 11: PEXESO: impact of τ and (%) of non empty leaf cells of the number of match cells

- **Dataset:** 10 Tables, 833 vectors.
- **Query:** 1 Query column of size 1.

5. PEXESO impact of pivot selection on data matching

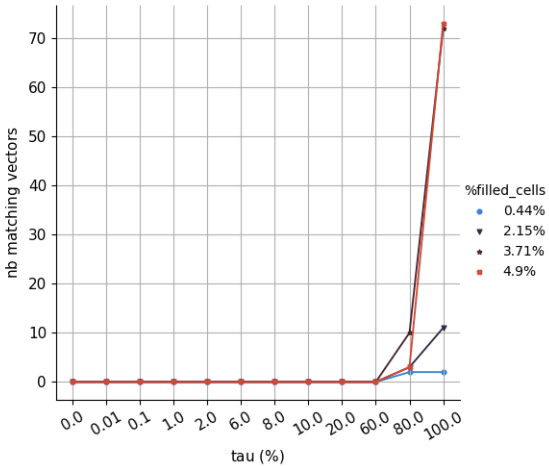

Figure 12: PEXESO: impact of τ and (%) of non empty leaf cells of the number of match vectors

- **Dataset:** 10 Tables, 833 vectors.
- **Query:** 1 Query column of size 1.

5. PEXESO impact of pivot selection on data matching

- PEXESO does not perform any cell/vector matching for $\tau \leq 60\%$.
- The higher the (%) of non empty cells the higher the number of match cells/vectors.