

COMS W6998

Wander join

Presenter: Xupeng Li

Scribe: Zachary Huang

Basic idea: Approximate aggregation result join.

Join is common but expensive. Join of small tables leads to large table.

Fast from hours to seconds with high confidence.

Ripple join vs Wander join

Ripple join dependent, uniform,

Ripple join maintain sample pool, record all samples and result,

Stored In memory,

Sample from A and B, and get the result,

Sample each row is uniform, but dependent on previous sampling.

Wander join independent, non-uniform,

Cases for cyclic or acyclic

1. Tree decomposition. 2 choose walk. 3. Sanity check

Experiment

1. converge time is much less for wander join for different queries

2. low time to achieve different confidence levels

Main contribution:

Propose a new, efficient algorithm, implemented in popular database.

Discussions:

Q: How to compute the confidence interval?

Because sample dependent, becomes very complex.

Wander join ~2015 won the best paper because blurs online algorithm, which is advanced for a long time.

Q: Why is ripple join bad? Is it because random? No Guide? What bad means?

Purpose of sampling is for performance.

Current semantics: sample from one final time, and further do query.

Use some magic gives us sample without runs join.

Sampling: how many times join and throw, which is inefficient.

One approach: random samples from left and right, do a join.

With likelihood, they will join, otherwise waste effort.

Likelihood depends on join and table.

$\text{Sample}(L) \text{ join } \text{Sample}(R) \text{ not equal } \text{Sample}(L \text{ join } R)$.

Also related to the hardware improvement.

80s: store on disk. Random access cost is large.

For disk only 5ms, page 8 kb, which you can sample. Scan 300 m/s.

Q: Why sample when can read and run the query?

Sample + Seek: know query, do sample ahead of time. Offline sampling.

Do the random seeking before won't be expensive.

Wander join: has large memory, can build indexes.

Index: receives a key, see records that have the key.

Therefore, can directly ask how many records match.

Likelihood of getting a good join much better.

Both online or offline indices are possible.

Random walk: hope that join, otherwise waste resource.

Q: What is independence sample? Why it is important? If so many matches for the last table, why not take all of them? Afterall, we want sample pool to be large!

Because they are not independent.

Random sample in memory is not cheap.

Filter will also cause waste if it's not in the index.

Group by is also filter. Filter by the group by values.

Indendence helps calculate the confidence. (Ripple has many long pages talking about calculating confidence interval, which is hard)

Q: How to use pangloss to run online aggregation?

Pangloss use sample seek, very quickly we get a good approximate result, wait a long time, and have the full result. Online aggregation gives you answer bit by bit. Wander join can increase every seconds.

Sample seek doesn't have anything in between. Get approximate quickly, and a long while for final result.

Q: How to visualize join?

No visualization currently considers join.

Wait ahead of time until the final result.

Why is visualization of join hard?

Because size, time...

Can we visualize it using data density for joins?

If joining tables inherently, there will be even larger tables.

Rendering table would also be a problem.

User may want to try join.

First join, if very large, then don't want them.

Query will shrink things down like in common website.

Therefore, interactive visualize join is very hard.