WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

CS 839: Topics in Database Management Systems

Lecture 5: Analytical Processing-2

Xiangyao Yu
9/20/2023

Update on the Schedule

Next Monday: Guest lecture from Jaso

“Transaction processing with disaggregation” starts next Wednesday (9/27)

Group discussion from last lecture

In the extreme, all computation can be pushed down to the storage, making the system shared-nothing again. Then, we lose the benefits of disaggregation. How do we strike the right balance between disaggregation and computation pushdown? Are there principles that computation pushdown must follow such that the system retain all the benefits of disaggregation?

Group discussion from last lecture

- Push minimal computation down to the storage
- Pushdown tasks that do not consume a lot of memory
- Push tasks that can reduce network
- Avoid communication among storage nodes (e.g., do not pushdown joins)
- Group-by aggregates are problematic
- Pushdown logic that is reusable across queries. Do not push user-specific logic.
- The query optimizer on the compute layer dynamically decide what to pushdown based on estimated cost
- If you can predict CPU usage of subplan, pushdown if CPU time is available

Analytical Processing-2 – Q/A

Extend Dremel to support joins and aggregations?

Why a single index lookup needs multiple accesses?

Is the metadata layer similar to compute layer in snowflake architecture?

Limited innovation and technical depth in the Lakehouse paper…

How are transactions supported in Lakehouse?

Discussion Question

Multiple papers suggest a new “layer” between compute and storage (e.g., spectrum layer in redshift, S3 select, metadata layer in Lakehouse). Please summarize the key functions that previous work proposed to push to such a layer.

Can you think of other functions that can be pushed to such a layer as well?

Can such a layer enable further performance optimizations or new use cases?