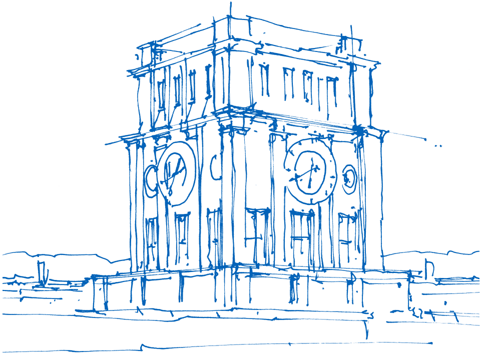
A Practical Approach to Groupjoin and Nested Aggregates

Philipp Fent, Thomas Neumann

Technische Universität München

VLDB 2021



Uhrenturm der TUM

Groupjoin

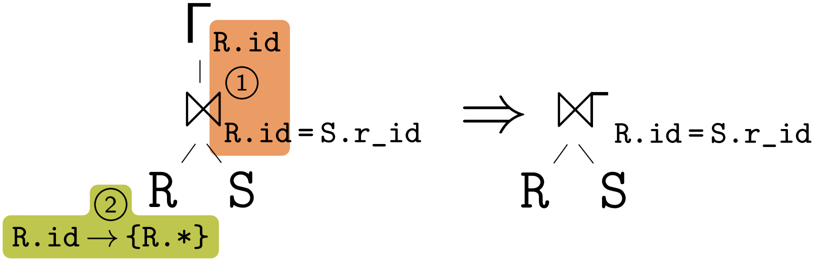
Combines Join and Group-By

$$\Gamma_{\texttt{R.id}}\left(\texttt{R} \bowtie_{\texttt{R.id}=\texttt{S.r\_id}} \texttt{S}\right) \quad \Rightarrow \quad \texttt{R} \bowtie\!\!\Gamma_{\texttt{R.id}=\texttt{S.r\_id}} \texttt{S}$$

$\texttt{R.id} \rightarrow \{\texttt{R.*}\}$

Groupjoin

Combines Join and Group-By



Groupjoin

The MD-join : An Operator for Complex OLAP

Damianos Chatziantoniou
Panakea Software Inc.
damianos@panakea.com

Michael Akinde
Dept. of Computer Science*
Aalborg University
strategy@cs.auc.dk

Theodore Johnson
Database Research Center
AT&T Labs - Research
johnsont@research.att.com

Samuel Kim
Dept. of Computer Science
Stevens Institute of Tech.
skim@cs.stevens-tech.edu

Main Memory Implementations for Binary Grouping

Norman May and Guido Moerkotte

University of Mannheim
B6, 29
68131 Mannheim, Germany
{norman,moer}@pi3.informatik.uni-mannheim.de

Accelerating Queries with Group-By and Join by Groupjoin

Guido Moerkotte
Universität Mannheim
Mannheim, Germany
moerkotte@informatik.uni-mannheim.de

Thomas Neumann
Technische Universität München
Munich, Germany
neumann@in.tum.de

Getting Swole: Generating Access-Aware Code with Predicate Pullups

Andrew Crotty
Brown University
crottyan@cs.brown.edu

Alex Galakatos
Brown University
agg@cs.brown.edu

Tim Kraska
MIT CSAIL
kraska@mit.edu

Groupjoin

The MD-join : An Operator for Complex OLAP

Damianos Chatziantoniou
Panakea Software Inc.
damianos@panakea.com

Michael Akinde
Dept. of Computer Science*
Aalborg University
strategy@cs.auc.dk

Theodore Johnson
Database Research Center
AT&T Labs - Research
johnsont@research.att.com

Samuel Kim
Dept. of Computer Science
Stevens Institute of Tech.
skim@cs.stevens-tech.edu

Main Memory Implementations for Binary Grouping

Norman May and Guido Moerkotte

University of Mannheim
B6, 29
68131 Mannheim, Germany
{norman,moer}@pi3.informatik.uni-mannheim.de

Accelerating Queries with Group-By and Join by Groupjoin

Guido Moerkotte
Universität Mannheim
Mannheim, Germany
moerkotte@informatik.uni-mannheim.de

Thomas Neumann
Technische Universität München
Munich, Germany
neumann@in.tum.de

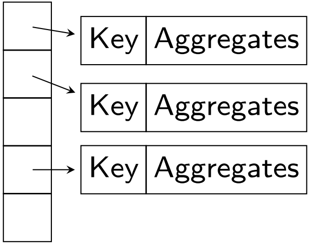
Getting Swole: Generating Access-Aware Code with Predicate Pullups

Andrew Crotty
Brown University
crottyan@cs.brown.edu

Alex Galakatos
Brown University
agg@cs.brown.edu

Tim Kraska
MIT CSAIL
kraska@mit.edu

Combined Hashtable



Groupjoin

The MD-join : An Operator for Complex OLAP

Damianos Chatziantoniou
Panakea Software Inc.
damianos@panakea.com

Michael Akinde
Dept. of Computer Science*
Aalborg University
strategy@cs.auc.dk

Theodore Johnson
Database Research Center
AT&T Labs - Research
johnsont@research.att.com

Samuel Kim
Dept. of Computer Science
Stevens Institute of Tech.
skim@cs.stevens-tech.edu

Main Memory Implementations for Binary Grouping

Norman May and Guido Moerkotte

University of Mannheim
B6, 29
68131 Mannheim, Germany
{norman,moer}@pi3.informatik.uni-mannheim.de

Accelerating Queries with Group-By and Join by Groupjoin

Guido Moerkotte
Universität Mannheim
Mannheim, Germany
moerkotte@informatik.uni-mannheim.de

Thomas Neumann
Technische Universität München
Munich, Germany
neumann@in.tum.de

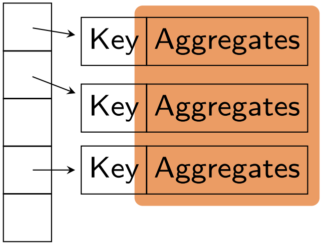
Getting Swole: Generating Access-Aware Code with Predicate Pullups

Andrew Crotty
Brown University
crottyan@cs.brown.edu

Alex Galakatos
Brown University
agg@cs.brown.edu

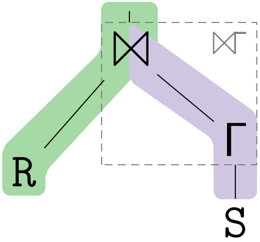
Tim Kraska
MIT CSAIL
kraska@mit.edu

Combined Hashtable



Groupjoin

Avoiding Contention with Eager Right Execution

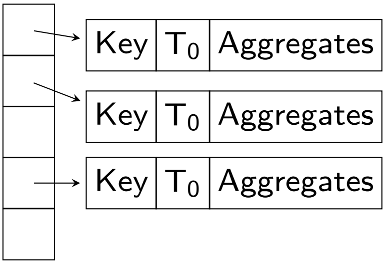


- Re-uses thread-local Γ logic
- Works well for $\sigma_S \approx 1$

Groupjoin

Memoizing Groupjoin

Global Hashtable



Thread-local Hashtables

T_1

$\Gamma_1 = \{\}$

T_2

$\Gamma_2 = \{\}$

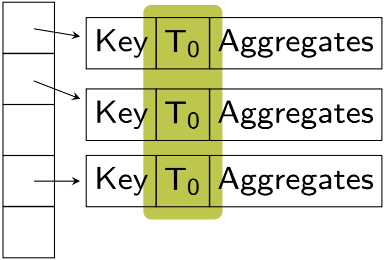
T_3

$\Gamma_3 = \{\}$

Groupjoin

Memoizing Groupjoin

Global Hashtable



Thread-local Hashtables

T_1

$\Gamma_1 = \{\}$

T_2

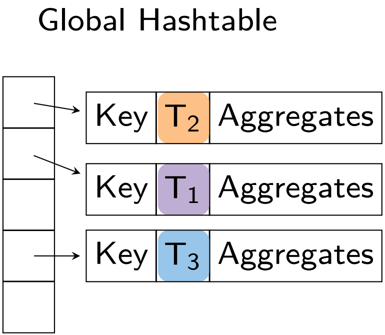
$\Gamma_2 = \{\}$

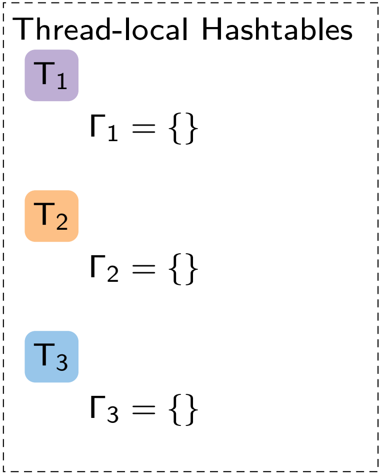
T_3

$\Gamma_3 = \{\}$

Groupjoin

Memoizing Groupjoin

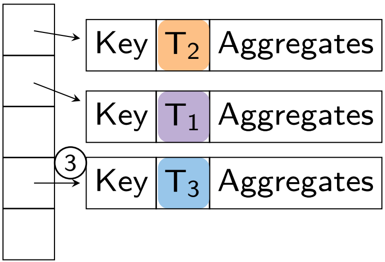




Groupjoin

Memoizing Groupjoin

Global Hashtable



Thread-local Hashtables

T_1

$$\Gamma_1 = \{③ : \gamma_1\}$$

T_2

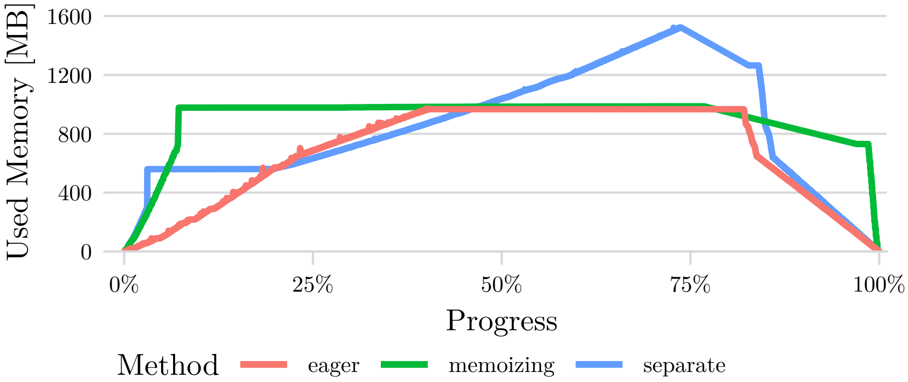
$$\Gamma_2 = \{③ : \gamma_2\}$$

T_3

$$\Gamma_3 = \{\}$$

Groupjoin

Comparison

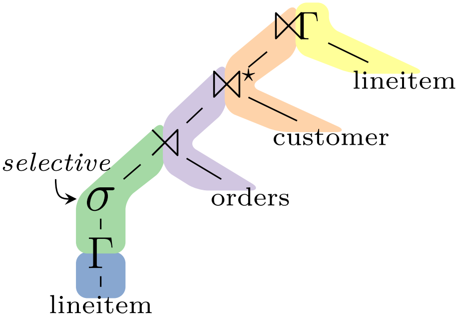


Nested Aggregates

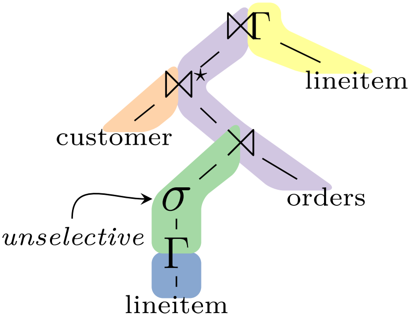
- Decorrelation supported by Groupjoin
- `HAVING` predicates are hard to estimate

Nested Aggregates

- Decorrelation supported by Groupjoin
- `HAVING` predicates are hard to estimate



a) Selective σ-Predicate



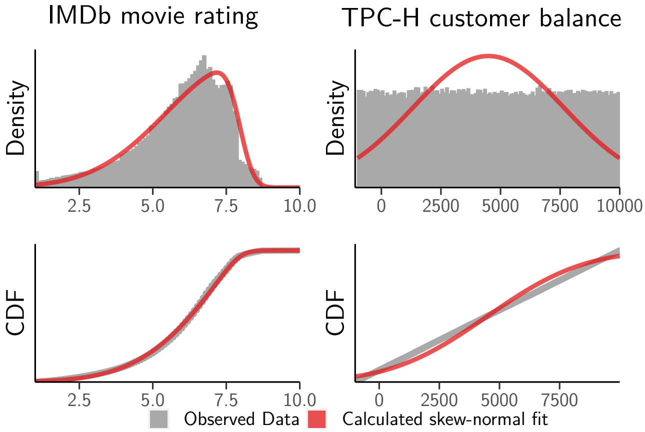
b) Unselective σ-Predicate

Estimating Aggregates

- Numerical columns $\sim \mathcal{N}(\mu, \sigma^2)$
- Cheap and generalizes nicely ... but inherently symmetric

Estimating Aggregates

- Numerical columns $\sim \mathcal{N}(\mu, \sigma^2)$
- Cheap and generalizes nicely ... but inherently symmetric



Estimating Aggregates

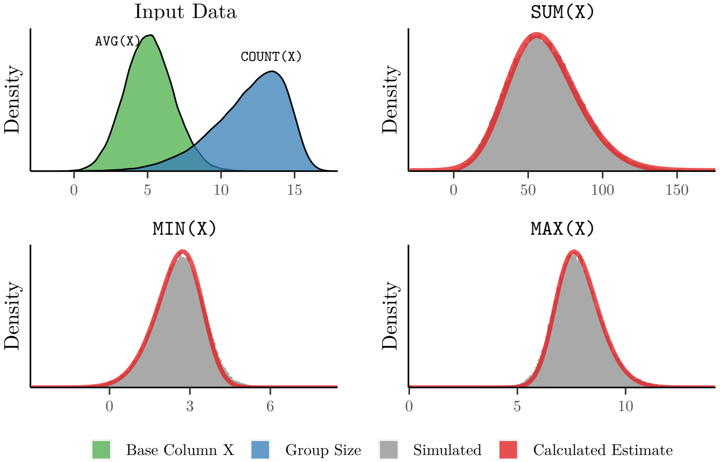
Using a skew-normal distribution

- $\Pr[x \leq c] = \Phi_{sn}(c)$
- Also approximates $(x \circ y)$, `AVG`/`SUM`/`MIN`/`MAX`

Estimating Aggregates

Using a skew-normal distribution

- $\Pr[x \leq c] = \Phi_{sn}(c)$
- Also approximates $(x \circ y)$, `AVG`/`SUM`/`MIN`/`MAX`



Practical Groupjoins and Nested Aggregates

Overall impact on TPC-H and TPC-DS

- Big improvements with cost model advised Groupjoin
- Slightly better query plans with `HAVING` estimates

Live queries: `https://umbra-db.com/interface`