Meet the Walkers

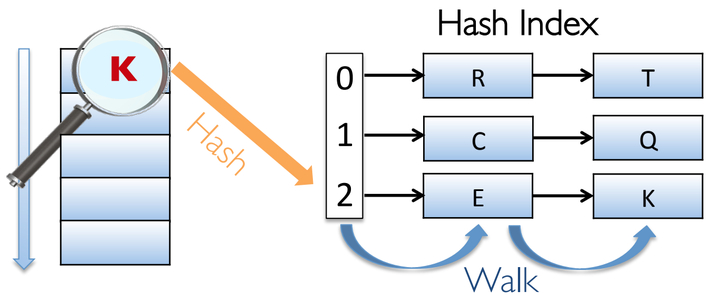
Accelerating Index Traversals for In-Memory Databases

Onur Kocberber[1], Boris Grot[2], Javier Picorel[1], Babak Falsafi[1], Kevin Lim[3], Parthasarathy Ranganathan[4]

[1]EcoCloud, EPFL [2]University of Edinburgh [3]HP Labs [4]Google, Inc.

Our World is Data-Driven!

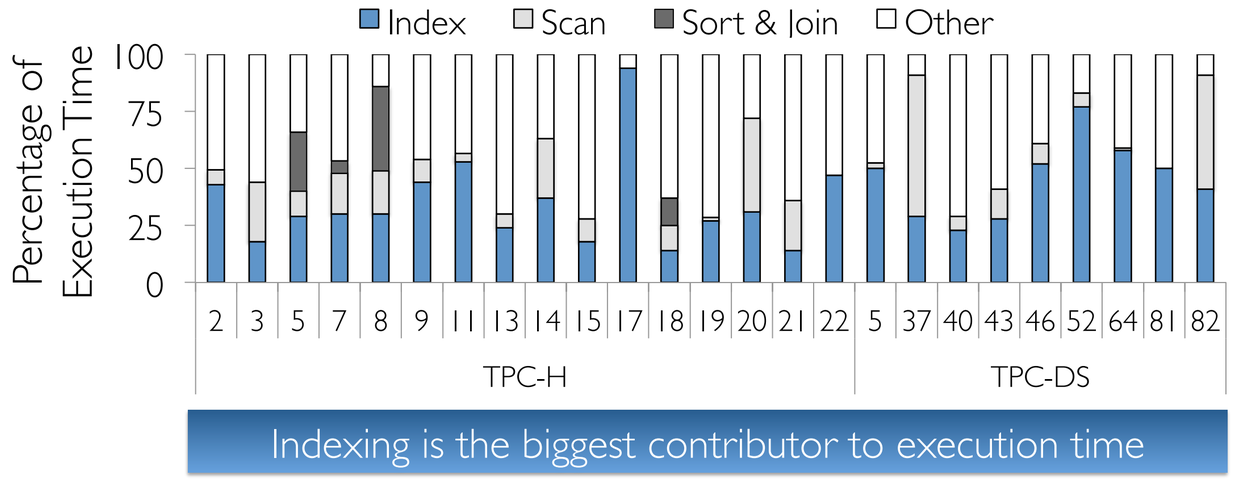
- Data resides in huge databases
- Indexes are essential for all database operators
- Hash index: a fundamental index structure



- Frequent use case: join via hash index

How Much Time is Spent Indexing?

Measurement on Xeon 5670 CPU with 100GB Dataset on MonetDB



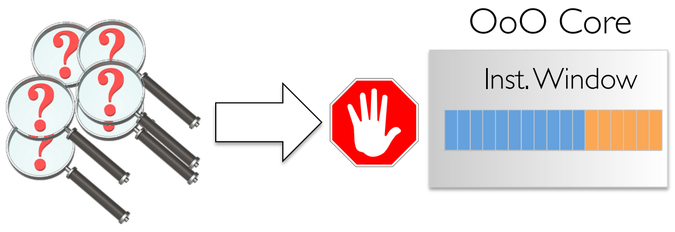
Indexing is the biggest contributor to execution time

Dissecting Index Lookups

Hash: Avg. 30% time of each lookup
- Computationally intensive, high cache locality

Walk: Avg. 70% time of each lookup
- Trivial computation, low cache locality

Next lookup: Inherently parallel
- But, beyond the inst. window capacity



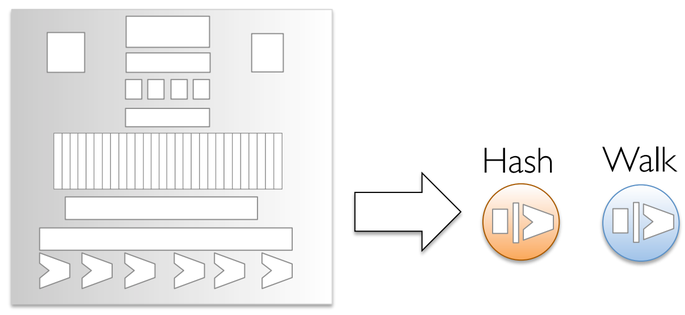
Roadmap

Roadmap for efficient and high-throughput index lookups

1. Specialize
   - Customize hardware for hash and walk
2. Parallelize
   - Perform multiple index lookups at a time
3. Generalize
   - Use a programmable building block

1. Specialize

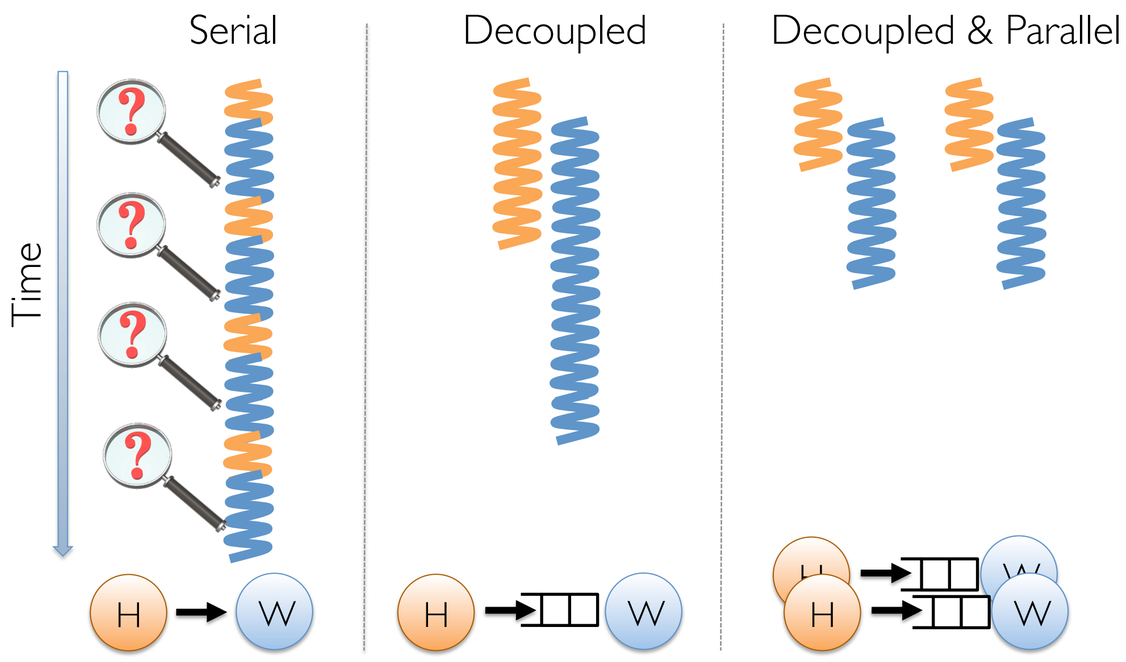
Design a dedicated unit for hash and walk



General-purpose OoO

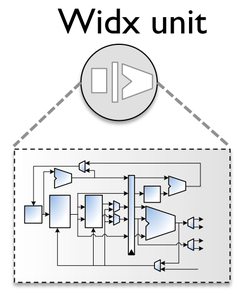
Specialized hash and walk hardware

2. Parallelize

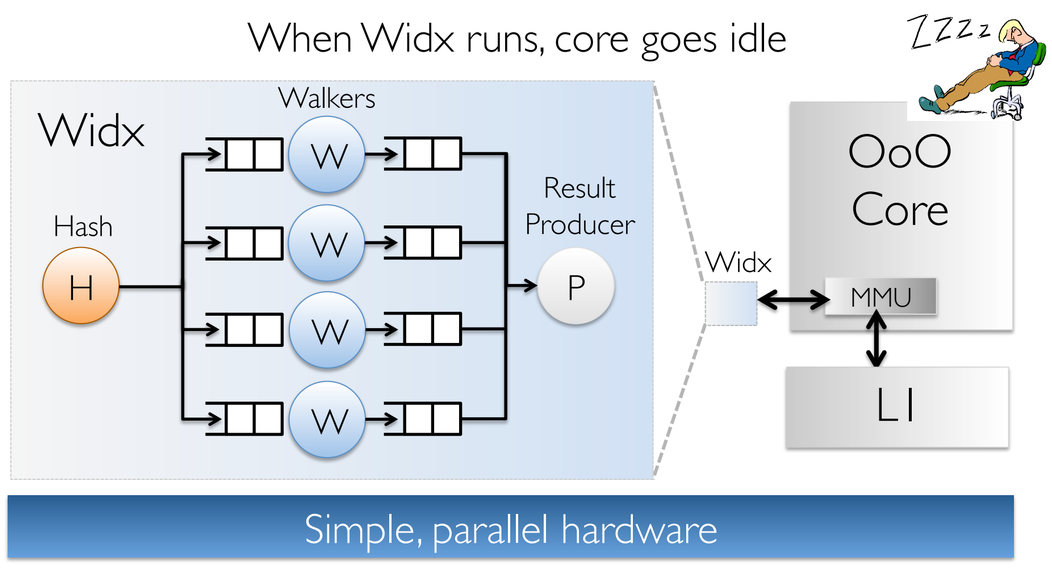


3. Generalize

- Widx unit: common building block for hash and walk
  - Two-stage RISC core
  - Custom ISA
- Widx units are programmable
  - Execute functions written in Widx ISA
  - Support limitless number of data structure layouts

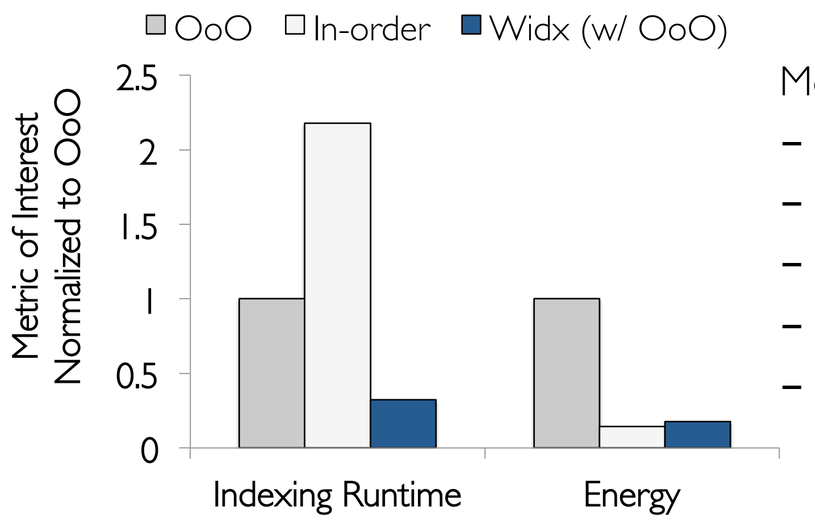


Putting it all together: Widx



Simple, parallel hardware

Result Highlights



More details in the paper!
- Per query results
- Join kernel evaluation
- Programming model
- Bottleneck analysis
- Widx cycle breakdowns

[2]This work was done while the author was at EPFL

[4]This work was done while the author was at HP Labs