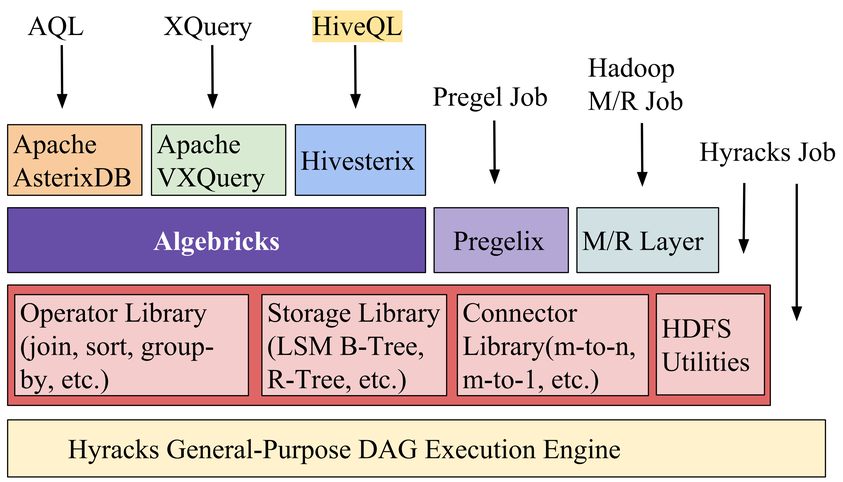
Algebricks: A Data Model-Agnostic Compiler Backend for Big Data Languages

Vinayak Borkar, Yingyi Bu, E. Preston Carman, Jr., Nicola Onose, Till Westmann, Pouria Pirzadeh, Michael J. Carey, Vassilis J. Tsotras

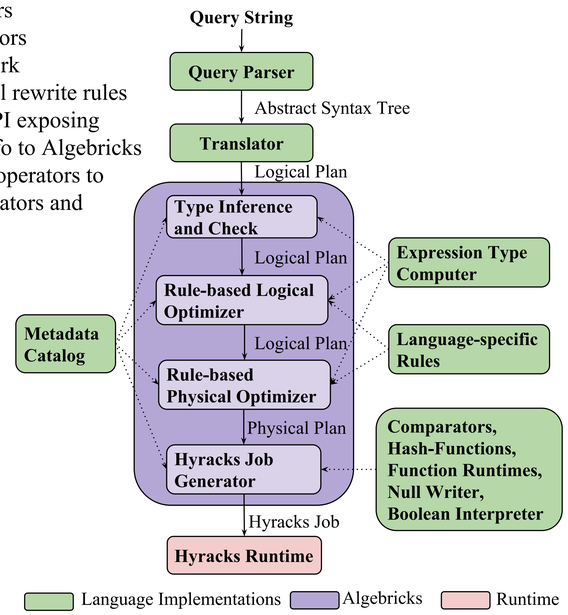
UC Irvine, X15 Software, Inc., UC Riverside, Oracle Labs https://asterixdb.ics.uci.edu https://asterixdb.incubator.apache.org

Software Stack



The Algebricks Framework

- Set of logical operators
- Set of physical operators
- Rewrite rule framework
- Set of generally useful rewrite rules
- Metadata provider API exposing metadata (catalog) info to Algebricks
- Mapping of physical operators to Hyracks runtime operators and connectors



Algebricks Nuts and Bolts

- Metadata interface
  - Data source metadata
  - Access path binding
  - Function metadata
- Logical operators
  - Function calls
    - Scalar
    - Aggregate
    - Stateful
    - Unnesting
- Physical operators
  - Exchange operators
    - One-to-One
    - Range
    - Broadcast
    - Hash
    - Random
- Rule-based optimizer
  - Logical optimizations
  - Physical optimizations
- Rewrite rules
  - Language-agnostic rules
  - Examples:
    - Pushing selects
    - Introducing projects
    - Query decorrelation
    - Used/produced variables
    - Functional dependencies, data properties
    - Equivalence classes

Hivesterix Example (HiveQL)

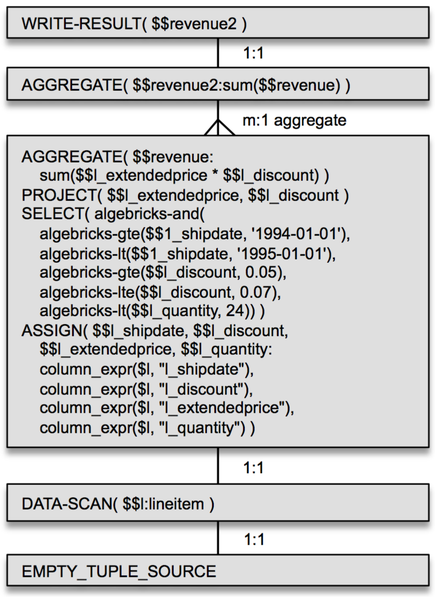
HiveQL Query

```
select sum(l_extendedprice*l_discount) as
revenue
from lineitem
where l_shipdate >= '1994-01-01'
  and l_shipdate < '1995-01-01'
  and l_discount >= 0.05
  and l_discount <= 0.07
  and l_quantity < 24;
```

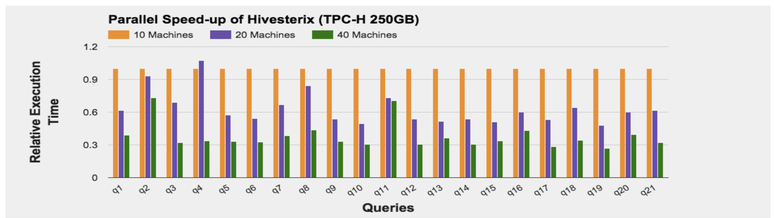
Algebricks Plan

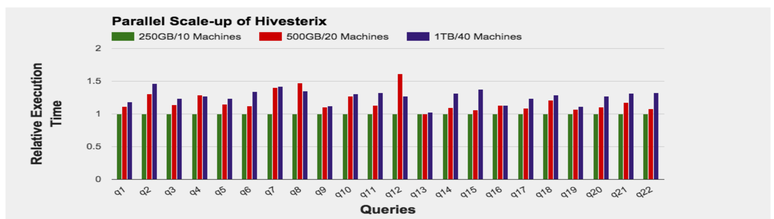
```
WRITE_RESULT( $$revenue )
AGGREGATE( $$revenue:sum(
   $$l_extendedprice*$$l_discount) )
SELECT( algebricks-and(
   algebricks-gte($$1_shipdate, '1994-01-01'),
   algebricks-lt($$1_shipdate, '1995-01-01'),
   algebricks-gte($$l_discount, 0.05),
   algebricks-lte($$l_discount, 0.07),
   algebricks-lt($$l_quantity, 24)) )
ASSIGN( $$l_shipdate, $$l_discount,
   $$l_extendedprice, $$l_quantity:
     column_expr($l, "l_shipdate"),
     column_expr($l,"l_discount"),
     column_expr($l, "l_extendedprice"),
     column_expr($l,"l_quantity") )
UNNEST( $$l:dataset(lineitem) )
EMPTY_TUPLE_SOURCE
```

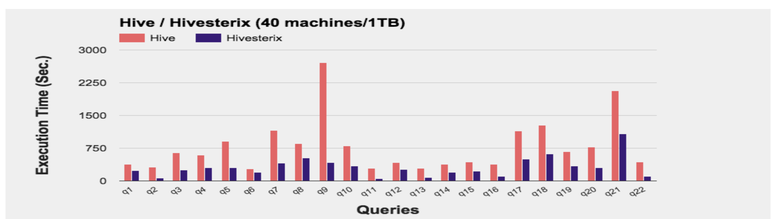
Hyracks Job



Hivesterix Experiments

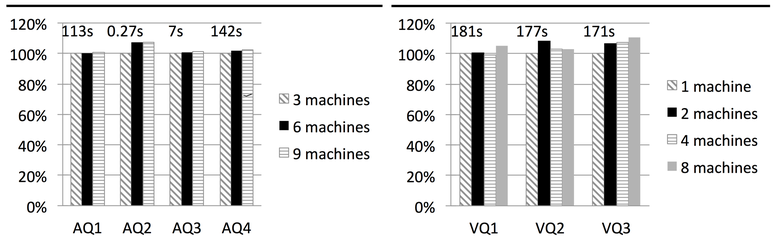






See Our Paper For More Information

- Design, implementation, use cases, and performance evaluation of Algebricks
- Hivesterix, Apache AsterixDB, and Apache VXQuery all built using Algebricks (with similarly good performance and scale-up results for both AQL and XQuery)



- While Algebricks is based on Hyracks, similar ideas could be used by other Big Data stacks (e.g., Spark, Flink, or Tez)
- Algebricks is available in open source under the Hyracks repository of AsterixDB (https://github.com/apache/incubator-asterixdb-hyracks)
- We hereby invite other Big Data researchers to download and try the system! (Array- or graph-based languages might be especially interesting to try....)
- *Future thoughts:* Add cost-based optimization and enhance the interaction between Algebricks and Hyracks to support dynamic query re-optimization

Sponsors: eBay, Facebook, Google, HTC, Microsoft, NSF, Oracle Labs, Yahoo!, Amazon Web Services, Infosys