Approximate Query Engines

Commercial Challenges and Research Opportunities

Barzan Mozafari
University of Michigan, Ann Arbor | SnappyData Inc.

© SIGMOD 2017
What Is Approximate Query Processing?

- Exact Result
- Computation
- Exact I/O

- Approximate Result
- Less Computation
- Less I/O
Why Approximation?

1. Productivity
   Numerous studies: A latency >2 seconds is no longer interactive and negatively affects creativity!

2. Money (Time + Resources)
   Human time: Money
   Machine time: No one loves their EC2 bill!

Massive Market for Interactive-speed Analytics!
Interactive Analytics: Myth or Reality?

Q: What about in-memory & columnar DBs?

A: Try running a few OLAP queries concurrently on 100GB of data partitioned across a few nodes!

Data Explosion: faster than Moore's law

Software Inefficiencies: excessive copying/serialization in modern apps

Hardware Limitations: memory wall

Shared Infrastructures: higher concurrency

Approximation seems to be a viable path to interactivity
Commercial Challenges
## AQP: Where Are We Now?

<table>
<thead>
<tr>
<th>OLAP Workloads</th>
<th>TPC-H</th>
<th>TPC-DS</th>
<th>Facebook</th>
<th>Conviva Inc.</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unsupported Queries</td>
<td>See paper</td>
<td>Full outer joins</td>
<td>Joins of multiple fact tables</td>
<td>Joins of multiple fact tables</td>
<td>Multiple fact joins, nested, textual filters</td>
</tr>
<tr>
<td>Percentage of Supported Queries</td>
<td>68%</td>
<td>&gt; 90%</td>
<td>&gt; 96 %</td>
<td>91%</td>
<td>74%</td>
</tr>
<tr>
<td>Speedup</td>
<td>10x</td>
<td>2x</td>
<td>?</td>
<td>10-200x</td>
<td>2-20x</td>
</tr>
</tbody>
</table>
AQP: Academia vs. Industry

25 Years of Successful Research

Zero Market Share*

* few exceptions: SnappyData, InfoBright

WHY?

- Deployment Challenges
- Interface Challenges
- Planning Challenges
Deployment Challenge 1: Vendor Resistance

1. AQP solutions typically require modifications of DBMS internals
   - Error estimation: BlinkDB, G-OLA, ...
   - Query evaluation: Online aggregation, synopses, ...
   - Overriding relational operators: ABS, ...

2. Major vendors are slow in adopting ANYTHING, especially AQP
   - Users won’t abandon their existing DBMS just to use AQP

Possible Solution: Middleware-based AQP engines
Middleware-based AQP: Challenges & Opportunities

**Advantage:** Ultimate generality

- Drop-in solution: No changes to underlying DBMS
- Works with all DBMSs: Vertica, Impala, SparkSQL, Hive, ...

**Challenge:** Ensuring efficiency

- Bootstrap, online aggregation, co-partitioning, ...
Deployment Challenge 2: Incompatibility with BI Tools

```
select geo, avg(bid) from adImpressions group by geo having avg(bid)>10
```

<table>
<thead>
<tr>
<th>geo</th>
<th>avg(bid)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI</td>
<td>21.5</td>
</tr>
<tr>
<td>WI</td>
<td>42.3</td>
</tr>
<tr>
<td>NY</td>
<td>65.6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

```
select geo, avg(bid) from adImpressions group by geo having avg(bid)>10 with error 0.05 at confidence 95
```

<table>
<thead>
<tr>
<th>geo</th>
<th>avg(bid)</th>
<th>error</th>
<th>prob_existence</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI</td>
<td>21.5</td>
<td>± 0.4</td>
<td>0.99</td>
</tr>
<tr>
<td>CA</td>
<td>18.3</td>
<td>± 5.1</td>
<td>0.80</td>
</tr>
<tr>
<td>MA</td>
<td>15.6</td>
<td>± 2.4</td>
<td>0.81</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Planning Challenge 1: Runtime Prediction

Predicting error is hard; Predicting latency is even harder!

Which sample type/size to choose?

- **Error Target**: return a 99% accurate answer
  - **Bootstrap**: impossible to predict \textit{a priori}
  - **Analytical**: limited (and expensive with joins)
  - **Analytical bootstrap**: requires changes to DBMS

- **Latency Target**: return an answer within 2 secs
  - Performance prediction of DBMS still an open problem

We must invest in analytical approaches & perf. prediction
Planning Challenge 2: Offline Provisioning

A nice database story: once upon a time there was a workload...

- The columnar DB speeds up queries by 100x!
  - If you build the right projections
- DBMS-X speeds up queries by 100x!
  - If you build the right indexes and materialized views
- BlinkDB speeds up queries by 100x!
  - If you build the right stratified samples

Challenge: Exploratory workloads constantly change
Exploratory and Adhoc Workloads

<table>
<thead>
<tr>
<th>Major Customers of a major OLAP DB</th>
<th>What percentage of previous column-sets change?</th>
<th>After 1 week</th>
<th>After 1 month</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer 1</td>
<td></td>
<td>71%</td>
<td>86%</td>
</tr>
<tr>
<td>Customer 2</td>
<td></td>
<td>90%</td>
<td>98%</td>
</tr>
<tr>
<td>Customer 3</td>
<td></td>
<td>80%</td>
<td>100%</td>
</tr>
<tr>
<td>Customer 4</td>
<td></td>
<td>85%</td>
<td>99%</td>
</tr>
<tr>
<td>Customer 5</td>
<td></td>
<td>69%</td>
<td>59%</td>
</tr>
<tr>
<td>Customer 6</td>
<td></td>
<td>75%</td>
<td>90%</td>
</tr>
</tbody>
</table>

What’s optimal now becomes useless next week
One Possible Direction: Robust Optimization (RO) Theory

- **Nominal Optimization**
  - Performance falls off of a cliff when target workload changes

- **Robust Optimization**
  - Performance degrades more gracefully
  - Robust against workload changes

- **CliffGuard** (http://cliffguard.org)
  - Open-source framework for finding robust physical designs for DBs [SIGMOD’15]
Other Planning Challenges

- Approximation quality (e.g., error) adds a new dimension to our search space

- Need for:
  - Approximation-aware query scheduling
  - Approximation-aware query optimization
  - Approximation-aware dynamic code generation
Interface Challenges

Specifying and interpreting complex error statistics can overwhelm an average DB user.
Possible Workaround I: High-level Accuracy Contracts (HAC)

- User picks a single number \( p \), where \( 0 \leq p \leq 1 \) (default \( p=0.95 \))
- Engine guarantees that user only sees rows & values that:
  1. are at least \( p\% \) accurate with \( p\% \) probability; and
  2. exist with \( p\% \) probability
- Can be set at the JDBC/ODBC connection level
- No extra columns are returned

<table>
<thead>
<tr>
<th>geo</th>
<th>avg(bid)</th>
<th>error</th>
<th>existence_prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI</td>
<td>9.5</td>
<td>± 0.4</td>
<td>0.99</td>
</tr>
<tr>
<td>WI</td>
<td>40.8</td>
<td>± 5.1</td>
<td>1.00</td>
</tr>
<tr>
<td>NY</td>
<td>70.5</td>
<td>± 2.4</td>
<td>1.00</td>
</tr>
<tr>
<td>IL</td>
<td>10.2</td>
<td>± 1.1</td>
<td>0.90</td>
</tr>
</tbody>
</table>

\[ \text{BI Compatibility!} \]
Possible Workaround II: Visualization

- A tuple is important ONLY insofar as it affects a visible pixel! [Viz-Aware Sampling ‘16]

Explicit error no longer needed if two plots are reasonably similar
Possible Workaround III: Early Result

- Show approx results instantly while full query is running
- Allows user to terminate full query
- Tremendous savings!

Try the full demo online: http://snappydata.io/isight
Research Opportunities
Research Opportunity 1: Database Learning

Main limitation of traditional DBs:
• They cannot reuse work: I/O and computation done for a query is wasted afterwards

Observation in an AQP setting:
• Every query reveals a bit of information about the unknown underlying distribution
Research Opportunity 1: Database Learning

Past Observations
\((x_1, y_1), \ldots, (x_n, y_n)\)

Supervised Learning

New Observation
\((x_{n+1}, ?)\)

Past Queries & Their Answers
\((Q_1, A_1), \ldots, (Q_n, A_n)\)

Database Learning

New Query
\((Q_{n+1}, ?)\)

Sample

\(\text{Smaller Sample} + \)  \(\rightarrow\)  Answer

A DB that becomes smarter and faster every time it is queried...

* See Yongjoo’s talk (Wed 11am)
Research Opportunity 2: Active Database Learning

Active Learning: The model actively decides which items should be labeled & added to its training data

Active Database Learning: Why wait for queries?

1. Flexible Criteria: Uncertainty, Informativeness, ...
2. Overcomes limitations of Materialized Views
Research Opportunity 3: Stochastic Query Planning

Traditional databases
• Limit themselves to only correct and equivalent plans
• Choose a single plan (“the best plan”)

New opportunity in an AQP setting
• Plans do not have to be equivalent (better if they’re not!)
• Deliberately pursue multiple plans in parallel to obtain multiple estimates
  • Various sample types, synopses, histograms, correlations, regression models,…
• Calibrate and combine into a single, more accurate approximation
Conclusion
Conclusion

• **Traditional optimization:** Access all relevant tuples efficiently while skipping irrelevant tuples
  - better parallelism, indexing, materialization, compression, columnar formats, in-memory and in-situ processing.

• **AQP:** Access only a tiny fraction of relevant tuples
  - orthogonal and complementary to traditional opt.
  - can solve some of traditional limitations of DBs.
  - more viable in the long term

• Lots of real-world challenges; lots of rich research problems

• Commercialization opportunities are improving
  - Need for educational efforts focused on end-user experience
References


