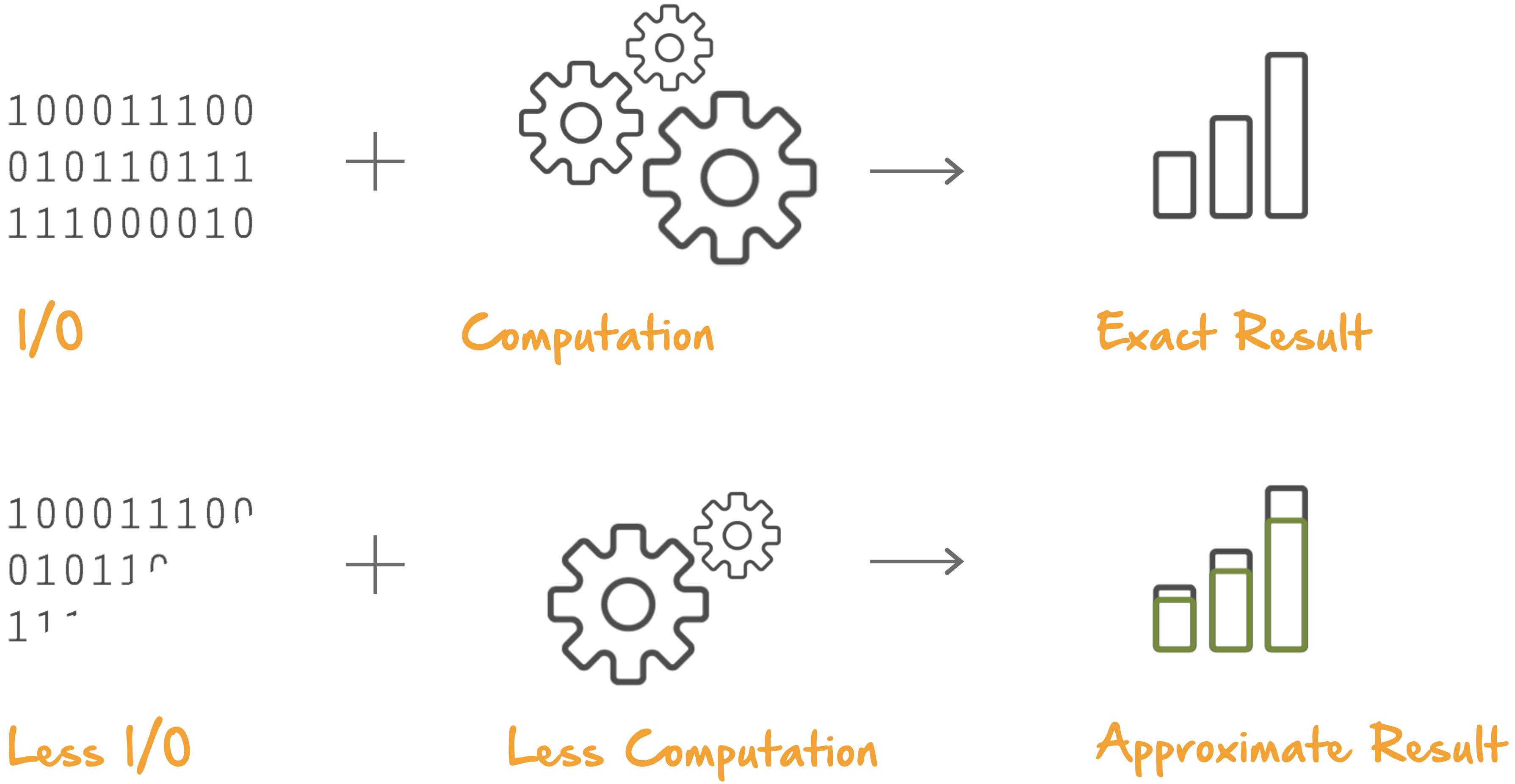
# Approximate Query Engines

## Commercial Challenges and Research Opportunities

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

# What Is Approximate Query Processing?



?

# 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!

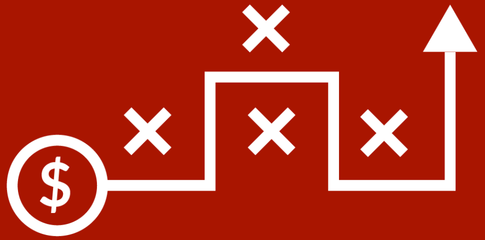
# Interactive Analytics: Myth or Reality?

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 | Software Inefficiencies | Hardware Limitations | Shared Infrastructures |
|---|---|---|---|
| faster than Moore's law | excessive copying/ serialization in modern apps | memory wall | higher concurrency |

Approximation seems to be a viable path to interactivity



# Commercial Challenges

?

# AQP: Where Are We Now?

| OLAP Workloads | TPC-H | TPC-DS | Facebook | Conviva Inc. | Customer |
|---|---|---|---|---|---|
| **System** | ABM [1] | QuickR [2] | BlinkDB [3] | [1] + [3] | Verdict [5] |
| **Unsupported Queries** | See paper | Full outer joins | Joins of multiple fact tables | Joins of multiple fact tables | Multiple fact joins, nested, textual filters |
| **Percentage of Supported Queries** | 68% | > 90% | > 96 % | 91% | 74% |
| **Speedup** | 10x | 2x | ? | 10-200x | 2-20x |

# 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**

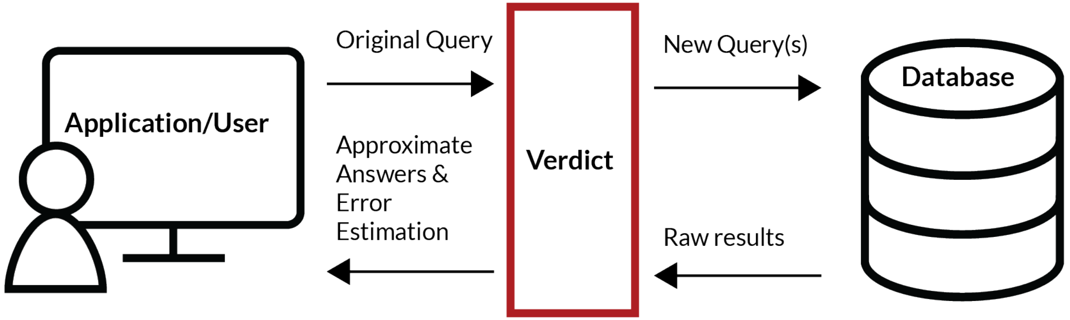
- Error estimation : BlinkDB, G-OLA, ...
- Query evaluation : Online aggregation, synopses, ...
- Overriding relational operators : ABS, ...

**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




Verdict Architecture (http://verdictdb.org)

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

| geo | avg(bid) |
|---|---|
| MI | 21.5 |
| WI | 42.3 |
| NY | 65.6 |
| … | … |

Qlik

DOMO

tableau

ZOOMDATA

ORACLE
BUSINESS INTELLIGENCE

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

| geo | avg(bid) | error | prob_existence |
|---|---|---|---|
| MI | 21.5 | ± 0.4 | 0.99 |
| CA | 18.3 | ± 5.1 | 0.80 |
| MA | 15.6 | ± 2.4 | 0.81 |
| … | … | … | …. |

# 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 *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

typical

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 <u>build the right stratified samples</u>

Challenge: Exploratory workloads constantly change

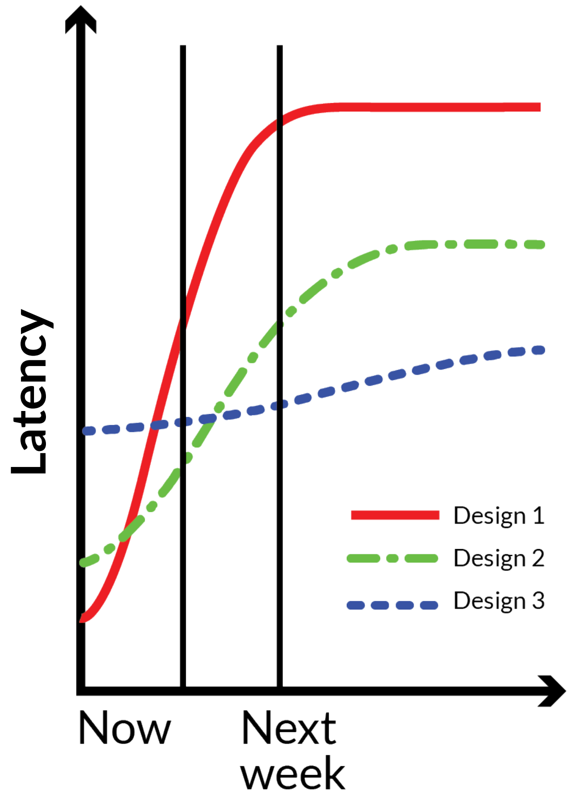
# Exploratory and Adhoc Workloads

| Major Customers of a major OLAP DB | What percentage of previous column-sets change? | |
|---|---|---|
| | After 1 week | After 1 month |
| Customer 1 | 71% | 86% |
| Customer 2 | 90% | 98% |
| Customer 3 | 80% | 100% |
| Customer 4 | 85% | 99% |
| Customer 5 | 69% | 59% |
| Customer 6 | 75% | 90% |

What's optimal now becomes useless next week

# One Possible Direction: Robust Optimzation (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

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

→

| geo | avg(bid) |
|---|---|
| MI | 10.5 |
| WI | 42.3 |
| NY | 65.6 |
| ~~IL~~ | ~~9.2~~ |

Exact Result

Subset error

| geo | avg(bid) | error | existence_prob |
|---|---|---|---|
| ~~MI~~ | ~~9.5~~ | ~~± 0.4~~ | 0.99 |
| WI | 40.8 | ± 5.1 | 1.00 |
| NY | 70.5 | ± 2.4 | 1.00 |
| IL | 10.2 | ± 1.1 | 0.90 |

Approx Result

Superset error

Aggregation error

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

BI Compatability!

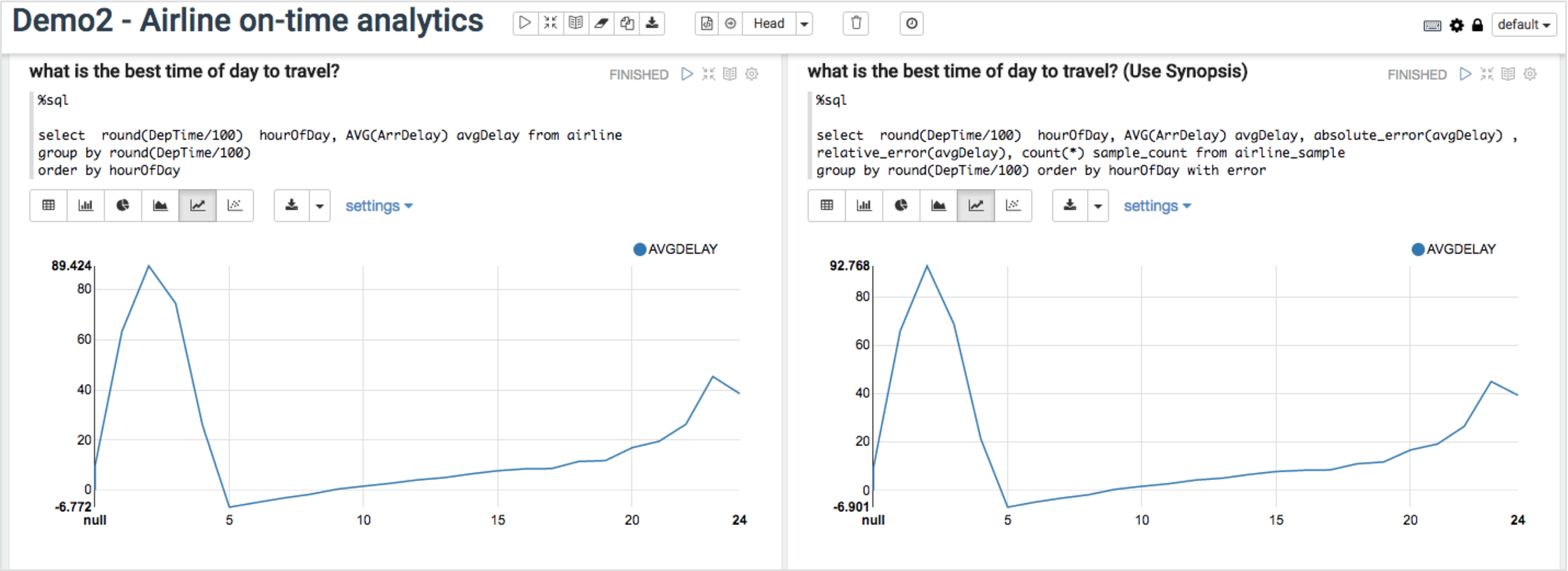
| geo | avg(bid) | error | existence_prob |
|---|---|---|---|
| MI | 9.5 | ± 0.4 | 0.99 |
| WI | 40.8 | ± 5.1 | 1.00 |
| NY | 70.5 | ± 2.4 | 1.00 |
| IL | 10.2 | ± 1.1 | 0.90 |

→

| geo | avg(bid) |
|---|---|
| MI | NULL |
| WI | 40.8 |
| NY | 70.5 |

# Possible Workaround II: Visualization

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



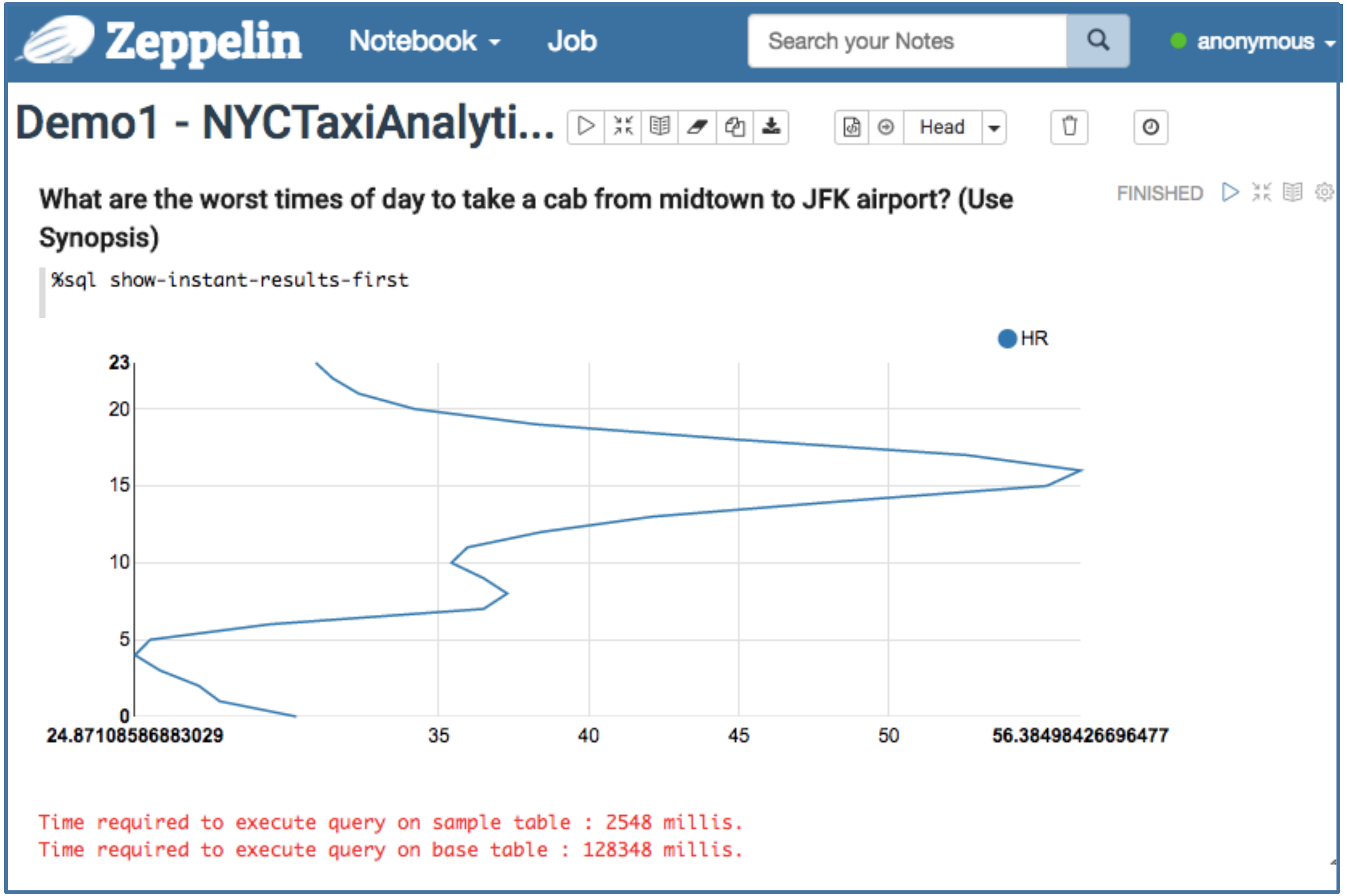

Time required to execute query : 27120 millis.

Time required to execute query : 230 millis.

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!



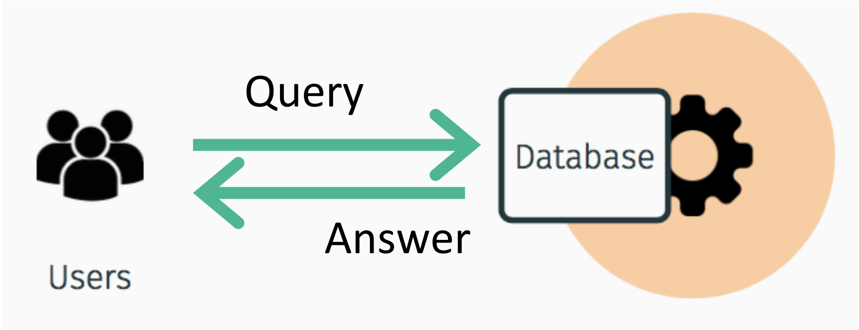

Try the full demo online:
http://snappydata.io/isight

# Research Opportunities

# Research Opportunity 1: Database Learning

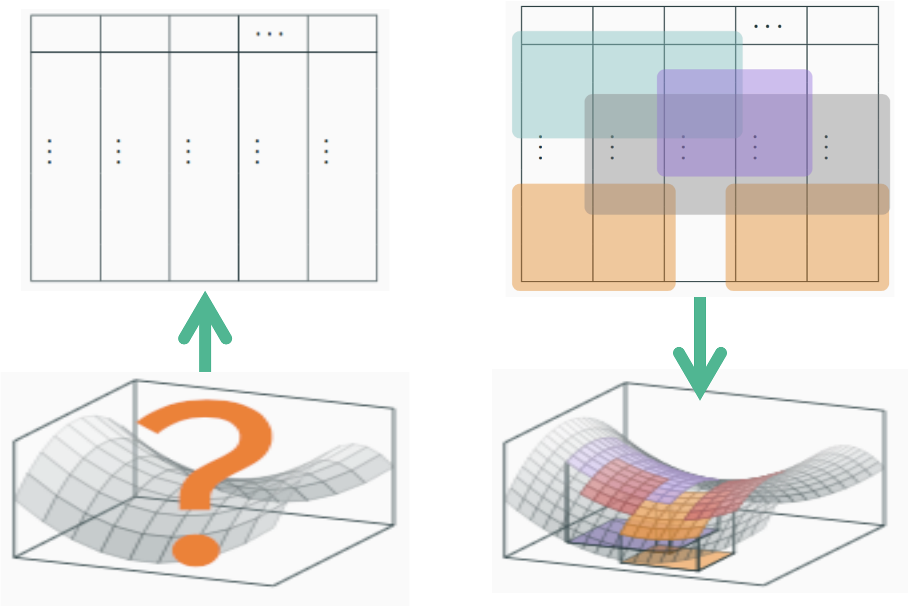
Main limitation of traditional DBs:

- 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

Supervised Learning

New Observation

$(x_1,y_1),\dots(x_n,y_n) \longrightarrow (x_{n+1},\ ?)$

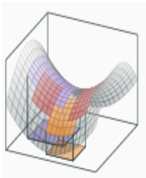
Past Queries & Their Answers

Database Learning

New Query

$(Q_1,A_1),\dots(Q_n,A_n) \longrightarrow (Q_{n+1},\ ?)$

Sample → Answer

Smaller Sample +  → Answer

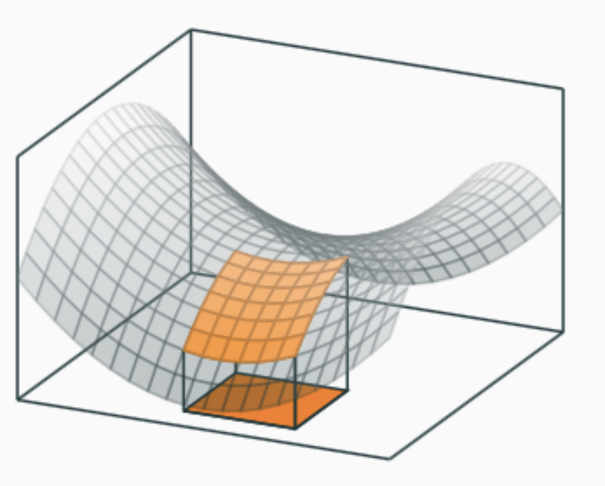
A DB that becomes smarter and faster every time it is queried…

* See Yongjoo's talk (Wed 11am)

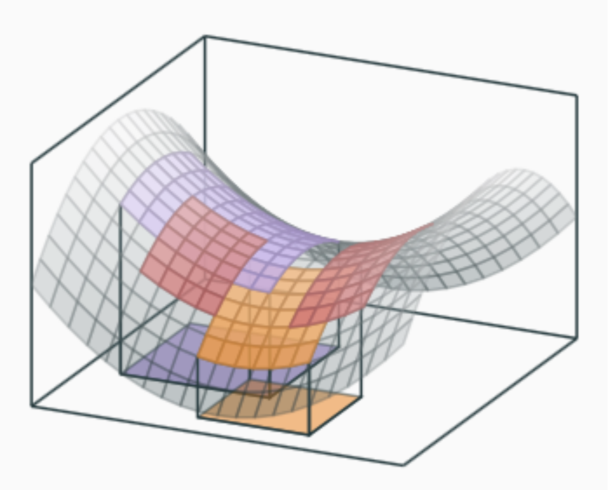
# Research Opportunity 2: Active Database Learning

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

Active Database Learning: Why wait for queries?



DB actively queries itself when idle



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,...
- Caliberate 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

[1] Kai Zeng, Shi Gao, Barzan Mozafari and Carlo Zaniolo. The Analytical Bootstrap: a New Method for Fast Error Estimation in Approximate Query, SIGMOD 2014

[2] Srikanth Kandula, Anil Shanbhag, Aleksandar Vitorovic, Matthaios Olma, Robert Grandl, Surajit Chaudhuri, Bolin Ding. Quickr: Lazily Approximating Complex Ad-Hoc Queries in Big Data Clusters, SIGMOD 2016

[3] Sameer Agarwal, Henry Milner, Ariel Kleiner, Ameet Talwalkar, Michael Jordan, Samuel Madden, Barzan Mozafari and Ion Stoica, Knowing When You're Wrong: Building Fast and Reliable Approximate Query Processing Systems, SIGMOD 2014

[4] Barzan Mozafari, Jags Ramnarayan, Sudhir Menon, Yogesh Mahajan, Soubhik Chakraborty, Hemant Bhanawat, Kishor Bachhav. SnappyData: A Unified Cluster for Streaming, Transactions and Interactice Analytics, CIDR 2017

[5] Yongjoo Park, Ahmad Shahab Tajik, Michael Cafarella, Barzan Mozafari. Database Learning: Toward a Database that Becomes Smarter Every Time, SIGMOD 2017