Data Management Stack Optimizations for Interactive User Interfaces

With interactive data analysis as the vision, today's user interfaces already provide the necessary tools for better interactivity. The big data systems, however, have workflows resembling the days of the punch cards: Jobs must be submitted to a scheduler, results arrive seconds or even minutes later. Optimizations to the data management stack partially solve the disparity between user interface advancements and big data systems. Additionally, providing a middleware solution enables users to implement these optimizations within their existing stack.

Programming

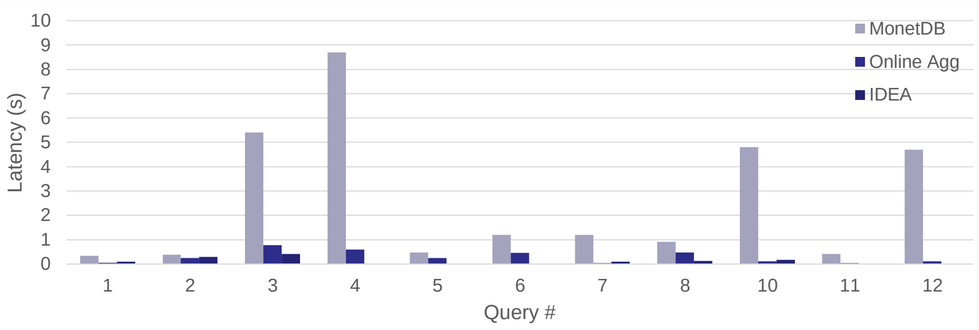
Visual

Language

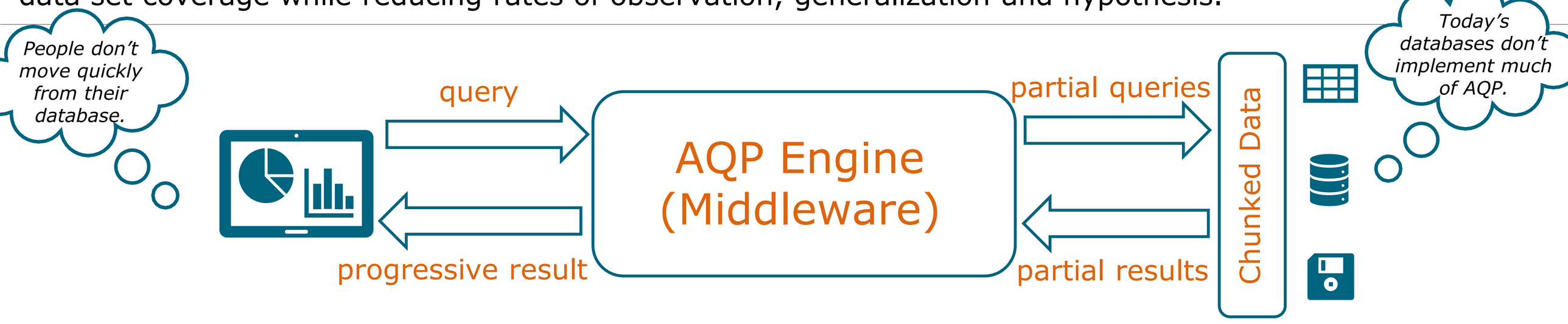
User Interfaces evolved from command lines to voice and touch interfaces, enabling more users to benefit from various applications. Big data systems lack in usability as most require the user to be a skilled programmer. Today's workflows most often involve a domain expert talking to a data scientist interacting with the big data systems, which is prone to misunderstanding.

#1	sex
#2	education
#3	education **WHERE** sex='Female'
#4	education **WHERE** sex='Male'
#5	sex, education
#6	sex **WHERE** education='PhD'

#7	salary
#8	salary **WHERE** education='PhD'
#9	sex, salary
#10	salary **WHERE** sex='Female'
#11	salary
#12	salary **WHERE** sex='Female'



Interface Latency can play an important role in shaping user behavior and impacts the outcomes of exploratory visual analysis. Delays of 500ms can incur significant costs, decreasing user activity and data set coverage while reducing rates of observation, generalization and hypothesis.



Approximate Query Processing (AQP) in the middleware can be done offline and online. Data sources are prepared with offline AQP, e.g. being split into chunks. With online AQP, queries are first divided into multiple queries and then merged to a result.

Sales

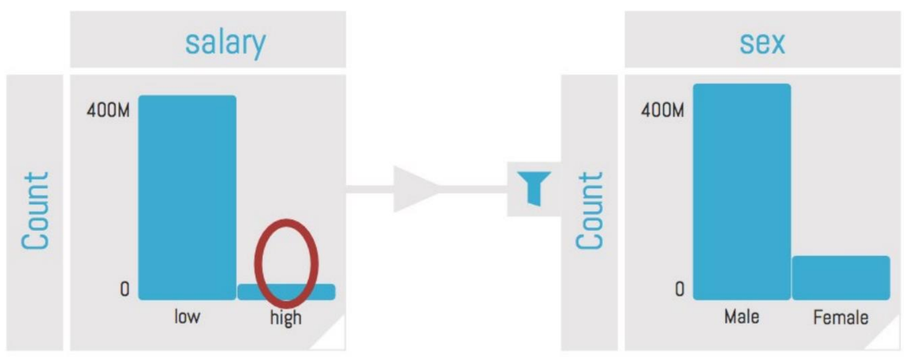
Product	Amount
CPU	1
CPU	1
CPU	2
CPU	3
CPU	4
Disk	1
Disk	2
Monitor	1

Sampling (Online OR Offline)

Sales-Sample

Product	Amount
CPU	1
CPU	2
CPU	3
Disk	2

On similar queries, **Result Caching** can be used to answer queries based on cached results of previous queries.



P_{male}	$\{0.70, \varepsilon_1\}$
P_{female}	$\{0.30, \varepsilon_2\}$
P_{high}	$\{0.20, \varepsilon_3\}$
P_{low}	$\{0.80, \varepsilon_4\}$
$P_{low \mid male}$	$\{0.75, \varepsilon_5\}$
$P_{low \mid female}$	$\{0.92, \varepsilon_6\}$
$P_{high \mid male}$	$\{0.25, \varepsilon_7\}$
$P_{low \mid female}$	$\{0.08, \varepsilon_8\}$

Fabian Heseding
Student B.Sc. IT-Systems Engineering

Lecture Series on Practical Data Engineering 2019/2020
Based on the lecture "Towards Interactive Data Analytics" by Carsten Binnig.
Hasso Plattner Institute, Potsdam, Germany

HPI Hasso Plattner Institut