CS 265

BIG DATA SYSTEMS

NoSQL | Neural Networks | SQL | Graph | Data Science
Leibniz, Let us calculate
How many and which structures are possible?

Can we compute performance w/o coding?
Can’t we do the same for other data-data-intensive problems?

statistics?    neural networks?
How to efficiently store and access data?

Design custom systems tailored for these actions
How to efficiently store and access data?

PROBLEM: algorithms and processes are changing literally everyday…
TAILORED SYSTEMS, HARDWARE OR EVEN EXTENSIONS ARE BOUND TO "FAIL"

Design custom systems tailored for these actions
Data Canopy

Statistical queries

Library of building blocks

Avoid redundant data access to accelerate statistical analysis
Statistic

Basic Aggregates
Statistic
Basic Aggregates

Data

a chunk
Q: Monthly Variance
Q: Monthly Variance

\[
\text{Variance} = \left( \frac{1}{n} \sum x^2 \right) - \left( \frac{1}{n} \sum x \right)^2
\]
Q: Monthly Variance

Variance = $\left( \frac{1}{n} \sum x^2 \right) - \left( \frac{1}{n} \sum x \right)^2 \rightarrow \sum x \quad \sum x^2$

Chunk size: 7 (a week)
Q: Monthly Variance \( \left( \frac{1}{n} \sum X^2 \right) - \left( \frac{1}{n} \sum X \right)^2 \)
Mean (first week) \[ \frac{1}{7} \sum x_i \]  
Monthly mean \[ \frac{1}{31} \sum x_i \]  
Var (first week) \[ \frac{1}{7} \sum x_i^2 - \left( \frac{1}{7} \sum x_i \right)^2 \]  
Reuse between ranges
Reuse between statistics
Mixed

Chunk size: 7 (a week)
Var (first week) \( \frac{1}{7} \sum x_i^2 - \left( \frac{1}{7} \sum x_i \right)^2 \)  Reuse between ranges

Monthly mean \( \frac{1}{31} \sum x_i \)  Reuse between statistics

Mean (first week) \( \frac{1}{7} \sum x_i \)  Mixed

Chunk size: 7 (a week)
Var (first week) \[ \frac{1}{7} \sum x_i^2 - \left( \frac{1}{7} \sum x_i \right)^2 \]

Reuse between ranges

Monthly mean \[ \frac{1}{31} \sum x_i \]

Reuse between statistics

Mean (first week) \[ \frac{1}{7} \sum x_i \]

Mixed

Chunk size: 7 (a week)
Var (first week) \[ \frac{1}{7} \sum x_i^2 - \left( \frac{1}{7} \sum x_i \right)^2 \] Reuse between ranges

Monthly mean \[ \frac{1}{31} \sum x_i \] Reuse between statistics

Mean (first week) \[ \frac{1}{7} \sum x_i \] Mixed

Chunk size: 7 (a week)
Basic Aggregates

Data

Decompose

Statistic

Store

Chunk size

Synthesize

a chunk

Synthesize

Store

Chunk size

Decompose

Statistic

Basic Aggregates
A basic aggregate

Data column
A basic aggregate

\[ \tau(x) = x \mid \tau(x) = x^2 \]
A basic aggregate

$\tau \circ f(S)$

Example: max, min, sum, product
\[ \sum X^2 \]

- 126
- 16
- 36
- 64
- 9

\[ f(\bullet) = \sum \bullet \quad \tau(\bullet) = \bullet^2 \]
\[ \sum X^2 \]

\[
\begin{bmatrix}
126 \\
16 \\
36 \\
64 \\
9
\end{bmatrix}
\]

\[ f(\bullet) = \sum \bullet \quad \tau(\bullet) = \bullet^2 \]

\[ \max \]

\[
\begin{bmatrix}
8 \\
6 \\
8 \\
3
\end{bmatrix}
\]

\[ f(\bullet) = \max \{ \bullet \} \quad \tau(\bullet) = \bullet \]
Statistics are decomposed into building blocks
\[ \frac{1}{n} \sum X \]

Arithmetic Mean

\[ 4.4 \]
>90% of statistics supported by NumPy and SciPy, and >75% of statistics supported by Wolfram can be expressed in this form.

### Table 1: Data Canopy terms.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache line size (bytes)</td>
<td></td>
</tr>
<tr>
<td>ST node size (bytes)</td>
<td></td>
</tr>
<tr>
<td>Chunk size (bytes)</td>
<td></td>
</tr>
<tr>
<td>Number of chunks</td>
<td></td>
</tr>
<tr>
<td>Number of rows</td>
<td></td>
</tr>
<tr>
<td>Number of columns</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2: Data Canopy terms.

<table>
<thead>
<tr>
<th>Type</th>
<th>Formula</th>
<th>(\Sigma x)</th>
<th>(\Sigma x^2)</th>
<th>(\Sigma xy)</th>
<th>(\Sigma y^2)</th>
<th>(\Sigma y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean (avg)</td>
<td>(\frac{\sum x_i}{n})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Root Mean Square (rms)</td>
<td>(\sqrt{\frac{1}{n} \cdot \sum x^2})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variance (var)</td>
<td>(\frac{\sum (x_i - \text{avg}(x))^2}{n})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard Deviation (std)</td>
<td>(\sqrt{\frac{\sum (x_i - \text{avg}(x))^2}{n}})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample Kurtosis (kur)</td>
<td>(\frac{1}{n} \sum (\frac{x_i - \text{avg}(x)}{\text{std}(x)})^4 - 3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample Covariance (cov)</td>
<td>(\frac{\sum x_i y_i}{n} - \frac{\sum x_i \sum y_i}{n^2})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simple Linear Regression (slr)</td>
<td>(\frac{\text{cov}(x,y)}{\text{var}(x)}, \text{avg}(x), \text{avg}(y)})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample Correlation (corr)</td>
<td>(\frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Basic Aggregates

Chunks

\[ F \]

Statistic function

Aggregate function

Transformation

Basic aggregates

\[ f \]

\[ \tau \]

\[ \text{Data column} \]

\[ \text{Chunk} \]
Basic Aggregates

Chunks

Statistic type

Aggregate function

F

Statistic function

f

Aggregate function

τ

Transformation

Basic aggregates
Basic Aggregates

Chunks

Statistic type

Aggregate function

Statistic function

Basic aggregates

Transformation

Overlapping ranges

Sub-ranges
Basic Aggregates

Statistic type

Mixed

Chunk

Transformation

Data column

Basic aggregates
Statistic

Basic Aggregates

Data

Decompose

Synthesize

Store

Chunk size

a chunk

Synthesize

Store

Chunk size

Data
Statistic
Basic Aggregates

Data

Decompose

Synthesize

Store

Chunk size

a chunk
Segment trees

Basic aggregates

Transformations

Chunks

Data column
Data

Logical chunk

| 5 | 4 | 2 | 5 | 4 | 4 | 6 | 8 | 0 | 1 | 7 | 2 | 2 | 1 | 3 | 5 |

DASlab
@ Harvard SEAS
Basic aggregates

<table>
<thead>
<tr>
<th>Leaves</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>7</td>
<td>8</td>
<td>14</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>

Data

<p>| | | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
42
One segment tree per basic aggregate per column

\[ \sum X \quad \sum X^2 \quad \sum X \quad \sum X^2 \]

Column a

Column b
One segment tree per basic aggregate per column

\[
\sum X, \quad \sum X^2, \quad \sum X, \quad \sum X^2
\]

Column a

Column b
Decompose

Statistic

Basic Aggregates

Synthesize

Store

Chunk size

Data

a chunk

Statistic

Basic Aggregates

Synthesize

Store

Chunk size

Data

a chunk
Calculate variance in temperature between May 15 and Oct. 15
Calculate variance in temperature between May 15 and Oct. 15
Calculate **variance in temperature** between May 15 and Oct. 15

\[
\text{variance} \quad \rightarrow \quad \frac{1}{N} \left( \sum x - \frac{1}{N} \sum x^2 \right)^2
\]

**Recipe**

May 15 and Oct. 15
Calculate variance in temperature between May 15 and Oct. 15

\[
\text{variance} \rightarrow \frac{\sum x - \left(\frac{\sum x^2}{N}\right)^2}{N-1}
\]

Recipe

May 15 and Oct. 15

Chunk size = 1 month
Calculate variance in temperature between May 15 and Oct. 15

\[
\text{variance} \quad \rightarrow \quad \sum x \quad \sum x^2 \quad N
\]

Recipe

May 15 and Oct. 15

a chunk

May | June | July | August | Sept. | Oct

temperature

base data

segment trees

base data

Chunk size = 1 month
Calculate variance in temperature between May 15 and Oct. 15

Plan

\[ \sum x \quad \sum x^2 \quad N \]

Recipe

base data \quad segment trees \quad base data

fractal size = 1 month
Statistic → Basic Aggregates

- Decompose
- Store
- Synthesize

Data

.chunk

Chunk size

?
DASlab
@ Harvard SEAS

Segment trees

base data

Chunk size = 1 month
Segment tree traversal

\[ O(\log(n/c)) \]

\( n \mid \text{column size}; \ c \mid \text{chunk size} \)
Segment tree traversal \(O(\log(n/c))\)                              Residual range scan \(O(c)\)

\[\begin{array}{c}
\text{base data} \quad \text{Segment trees} \quad \text{base data} \\
\end{array}\]

\(n\) | column size; \(c\) | chunk size
Total cost = $O(\log(n/c) + c)$

Segment tree traversal

Residual range scan

$O(\log(n/c))$

$O(c)$

n | column size; c | chunk size
Total cost = $O(\log(n/c) + c)$

Query cost

Chunk size

$n|$column size; $c|$chunk size
Total cost = $O(\log(n/c) + c)$
Total cost = $O(\log(n/c) + c)$
Total cost = $O(\log(n/c) + c)$
Total cost = \( O(\log(n/c) + c) \)
Total cost = $O(\log(n/c) + c)$
Given q, calculate the optimal query depth

Traverse the optimal depth, then scan the data

\[ d_q \]

\[ d_{\text{max}} \]
Operation Modes

Offline

Online

Speculative
Range Uniform Workload

1 thread  In-memory  40M rows, 100 columns, 5 statistic types
MonetDB
Modeltools (R)
NumPy (Python)

Latency (ms) vs Query Sequence

1 thread  In-memory  40M rows, 100 columns, 5 statistic types
Range Uniform Workload

Latency (ms)

MonetDB

Modeltools (R)

NumPy (Python)

Online Data Canopy

Query Sequence

1 thread  In-memory  40M rows, 100 columns, 5 statistic types
Total execution time (s)

<table>
<thead>
<tr>
<th></th>
<th>No DC</th>
<th>Online</th>
<th>Offline</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Range-Uniform</strong></td>
<td>217.44</td>
<td>45.89</td>
<td>1.05</td>
</tr>
<tr>
<td><strong>Range-Zoom-in</strong></td>
<td>432.53</td>
<td>31.42</td>
<td>1.05</td>
</tr>
</tbody>
</table>

* 2000 Queries

1 thread | In-memory | 40M rows, 100 columns, 5 statistic types
1 thread  In-memory  40M rows, 100 columns, 5 statistic types
Total execution time (s)

No DC

Individual

Sequential

Collaborative Filtering
Bayesian Classification
Simple Linear Regression

1 thread  In-memory  40M rows, 100 columns, 5 statistic types
Total execution time (s)

No DC

Individual

Sequential

Online

Collaborative Filtering

Bayesian Classification

Simple Linear Regression

1 thread  In-memory  40M rows, 100 columns, 5 statistic types
Total execution time (s)

- No DC
- Individual
- Sequential

Online

Collaborative Filtering
Bayesian Classification
Simple Linear Regression

1 thread  In-memory  40M rows, 100 columns, 5 statistic types
Data Canopy: Accelerating Exploratory Statistical Analysis

Statistics are everywhere!

Repetitive statistics and data access

Data Canopy synthesizes statistics from basic ingredients
BIG DATA SYSTEMS

NoSQL | Neural Networks | SQL | Graph | Data Science