Opaque: An Oblivious and Encrypted Distributed Analytics Platform

Wenting Zheng, Ankur Dave, Jethro Beekman, Raluca Ada Popa, Joseph Gonzalez, and Ion Stoica

UC Berkeley
Complex analytics run on sensitive data
Complex analytics run on sensitive data

client

cloud provider
Complex analytics run on sensitive data
Cloud attackers

client

cloud provider

sensitive data
Cloud attackers

client

cloud provider

sensitive data
Cloud attackers

client

cloud provider

sensitive data
Cloud attackers

Client

Cloud provider

Sensitive data
How to protect data and computation while preserving functionality?
Cryptographic approaches

- Generic functionality: fully homomorphic encryption, ObliVM
  [RAD’78, Gentry’09]
Cryptographic approaches

- Generic functionality: fully homomorphic encryption, ObliVM
  [RAD’78, Gentry’09]

  too slow
Cryptographic approaches

- Generic functionality: fully homomorphic encryption, ObliVM [RAD'78, Gentry'09] too slow

- Specialized solutions: CryptDB, Arx, Seabed
Cryptographic approaches

- Generic functionality: fully homomorphic encryption, ObliVM
  [RAD’78, Gentry’09]
  too slow

- Specialized solutions: CryptDB, Arx, Seabed
  restricted functionality
Cryptographic approaches

- Generic functionality: fully homomorphic encryption, ObliVM
  [RAD’78, Gentry’09]
  
  too slow

- Specialized solutions: CryptDB, Arx, Seabed

  restricted functionality

Alternative: hardware enclaves
Hardware enclaves
(e.g., Intel SGX)
Hardware enclaves
(e.g., Intel SGX)

- Hardware-enforced secure execution environment
Hardware enclaves

(e.g., Intel SGX)

- Hardware-enforced secure execution environment
Hardware enclaves
(e.g., Intel SGX)

- Hardware-enforced secure execution environment
- Encrypted enclave memory called EPC (accessible only from the enclave)
Hardware enclaves
(e.g., Intel SGX)

- Hardware-enforced secure execution environment
- Encrypted enclave memory called EPC (accessible only from the enclave)
Hardware enclaves

(e.g., Intel SGX)

- Hardware-enforced secure execution environment
- Encrypted enclave memory called EPC (accessible only from the enclave)
- Protect against an attacker who has root access
Remote attestation

Client

Server

enclave

untrusted OS
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange.
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange.
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange.
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange.
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange
Remote attestation

Enables verifying which code runs in the enclave and performing key exchange
Enclave-based systems
Enclave-based systems

- Prior work: Haven [BMG '14], Scone [ATGL.. '16], VC3 [SCFGPMR '15]
Enclave-based systems

- Prior work: Haven [BMG '14], Scone [ATGKL.. '16], VC3 [SCFGPMR '15]
  - full functionality
Enclave-based systems

- Prior work: Haven [BMG ’14], Scone [ATGKL.. ’16], VC3 [SCFGPMR ’15]
  - full functionality
  - great performance
Enclave-based systems

- Prior work: Haven [BMG '14], Scone [ATGKL.. '16], VC3 [SCFGPMR '15]
  - full functionality
  - great performance
  - data access pattern leakage [XCP '15, OCFGKS '15]
Access patterns
Access patterns

machine 0

processor

addresses

memory

locked
Access patterns

machine 0

processor

addresses

memory

network messages

machine 1
Example: network access pattern leakage
Example: network access pattern leakage

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>
Example: network access pattern leakage

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>

SELECT count(*) FROM medical
GROUP BY disease
Example: network access pattern leakage
Example: network access pattern leakage
Example: network access pattern leakage
Example: network access pattern leakage
Example: network access pattern leakage

<table>
<thead>
<tr>
<th>ID</th>
<th>...</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>...</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>...</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>...</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>...</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>...</td>
<td>Cancer</td>
</tr>
<tr>
<td>32591</td>
<td>...</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>
Example: network access pattern leakage
Example: network access pattern leakage

Public information:
Diabetes twice as common as cancer
Example: network access pattern leakage

Public information: Diabetes twice as common as cancer
Example: network access pattern leakage

Public information: Diabetes twice as common as cancer
Example: network access pattern leakage

Public information:
Diabetes twice as common as cancer
Example: network access pattern leakage

- 12809 ... Diabetes
- 29489 ... Diabetes
- 13744 ... Cancer
- 18740 ... Diabetes
- 98329 ... Cancer
- 32591 ... Diabetes
- 12809 ... Diabetes
- 29489 ... Diabetes
- 13744 ... Cancer
- 18740 ... Diabetes
- 32591 ... Diabetes
- 13744 ... Cancer
- 98329 ... Cancer
Example: network access pattern leakage
Example: network access pattern leakage

| 12809 | ... | Diabetes |
| 29489 | ... | Diabetes |
| 13744 | ... | Cancer   |

| 18740 | ... | Diabetes |
| 98329 | ... | Cancer   |
| 32591 | ... | Diabetes |
Example: network access pattern leakage

12809 ... Diabetes
29489 ... Diabetes
13744 ... Cancer

98329 ... Cancer

18740 ... Diabetes
32591 ... Diabetes
Example: network access pattern leakage
Example: network access pattern leakage

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Diabetes</th>
<th></th>
<th></th>
<th>Cancer</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>...</td>
<td>Diabetes</td>
<td>29489</td>
<td>...</td>
<td>Diabetes</td>
<td>13744</td>
<td>...</td>
</tr>
<tr>
<td>18740</td>
<td>...</td>
<td>Diabetes</td>
<td>32591</td>
<td>...</td>
<td>Diabetes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Diabetes</th>
<th></th>
<th></th>
<th>Cancer</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>98329</td>
<td>...</td>
<td>Cancer</td>
<td>98329</td>
<td>...</td>
<td>Cancer</td>
<td>98329</td>
<td>...</td>
</tr>
<tr>
<td>98329</td>
<td>...</td>
<td>Cancer</td>
<td>98329</td>
<td>...</td>
<td>Cancer</td>
<td>98329</td>
<td>...</td>
</tr>
<tr>
<td>98329</td>
<td>...</td>
<td>Cancer</td>
<td>98329</td>
<td>...</td>
<td>Cancer</td>
<td>98329</td>
<td>...</td>
</tr>
<tr>
<td>98329</td>
<td>...</td>
<td>Cancer</td>
<td>98329</td>
<td>...</td>
<td>Cancer</td>
<td>98329</td>
<td>...</td>
</tr>
</tbody>
</table>
Example: network access pattern leakage

Learns that Alice has cancer
Leakage from prior work

- Memory access patterns attacks [XCP15] extracted complete text documents and photo outlines
- Network access patterns [OCF+15] extracted age, gender, address of individuals
Goal: oblivious distributed analytics
Goal: oblivious distributed analytics

access patterns are independent of data content
Opaque*: oblivious and encrypted distributed analytics platform

* Oblivious Platform for Analytic QUEries
Threat model
Threat model

• Powerful attacker who can compromise the server’s software stack (including the OS)
Threat model

- Powerful attacker who can compromise the server’s software stack (including the OS)
- Cannot compromise the trusted hardware or the client
Threat model

- Powerful attacker who can compromise the server’s software stack (including the OS)
- Cannot compromise the trusted hardware or the client
- Small region of oblivious memory
Security guarantees (informal)
Security guarantees (informal)

• Data encryption and authentication
Security guarantees (informal)

- Data encryption and authentication
Security guarantees (informal)

- **Data encryption** and **authentication**

- **Computation integrity**: the client can check that the computation result was not affected by an attacker
Security guarantees (informal)

- **Data encryption** and **authentication**

- **Computation integrity:** the client can check that the computation result was not affected by an attacker
Security guarantees (informal)

- **Data encryption** and **authentication**

- **Computation integrity**: the client can check that the computation result was not affected by an attacker

- **Obliviousness**: The memory and network accesses of a query is the same for any two inputs with the same size characteristics (input/outputs)
Security guarantees (informal)

- **Data encryption** and **authentication**

- **Computation integrity**: the client can check that the computation result was not affected by an attacker

- **Obliviousness**: The memory and network accesses of a query is the same for any two inputs with the same size characteristics (input/outputs)
  - When enabling padding, Opaque hides output sizes as well
Challenge: obliviousness is expensive
Challenge: obliviousness is expensive

Two-part solution:
Challenge: obliviousness is expensive

Two-part solution:
Distributed oblivious SQL operators
Challenge: obliviousness is expensive

Two-part solution:
Distributed oblivious SQL operators
Novel query planning techniques
Opaque components
Opaque components

Data encryption and authentication
Opaque components

Computation verification

Data encryption and authentication
Opaque components

Distributed oblivious operators

- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Opaque components

Oblivious query planning
- Cost model
- Rule-based opt.
- Cost-based opt.

Distributed oblivious operators
- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Opaque components

Oblivious query planning
- Cost model
- Rule-based opt.
- Cost-based opt.

Distributed oblivious operators
- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Query execution
Query execution
query = SELECT sum(*) FROM table
query = SELECT sum(*) FROM table
query = SELECT sum(*)
FROM table
query = SELECT sum(*) FROM table
query = SELECT sum(*)
FROM table
query = SELECT sum(*) FROM table
query = SELECT sum(*) FROM table
query = SELECT sum(*)
FROM table
query = SELECT sum(*) FROM table
query = SELECT sum(*) FROM table
query = SELECT sum(*) FROM table
query = SELECT sum(*)
FROM table

Client
Problem: cloud can alter distributed computation
Problem: cloud can alter distributed computation

- Drop data
Problem: cloud can alter distributed computation

- Drop data
- Modify data
Problem: cloud can alter distributed computation

- Drop data
- Modify data
- Skip task
Problem: cloud can alter distributed computation

- Drop data
- Modify data
- Skip task
- Replay old state
Example: drop data

query = SELECT sum(*) FROM table
Example: drop data

query = SELECT sum(*)
FROM table
Example: drop data

query = SELECT sum(*)
FROM table
Example: drop data

query = SELECT sum(*)
FROM table
Example: drop data

query = SELECT sum(*)
FROM table

Client

Server

Scheduler

Database

Opaque
Catalyst
Spark Driver
Example: drop data

query = SELECT sum(*)
FROM table
Self-verifying computation

Invariant: if computation does not abort, the execution completed so far is correct
Self-verifying computation

Invariant: if computation does not abort, the execution completed so far is correct
Self-verifying computation

Invariant: if computation does not abort, the execution completed so far is correct

If the computation is complete, then the entire query was executed correctly
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

Task 13

Task 14

Task 15

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Self-verifying computation

query = SELECT sum(*)
FROM table
Opaque components

- Oblivious query planning
  - Cost model
  - Rule-based opt.
  - Cost-based opt.

- Distributed oblivious operators
  - Oblivious Filter
  - Oblivious Aggregation
  - Oblivious Join

- Computation verification

- Data encryption and authentication
Opaque components

Oblivious query planning
- Cost model
- Rule-based opt.
- Cost-based opt.

Distributed oblivious operators
- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12809</td>
<td>Diabetes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>29489</td>
<td>Diabetes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13744</td>
<td>Cancer</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>18740</td>
<td>Diabetes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>98329</td>
<td>Cancer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32591</td>
<td>Diabetes</td>
<td></td>
</tr>
</tbody>
</table>
Oblivious aggregation

```sql
SELECT count(*) FROM medical GROUP BY disease
```

There can be many partitions
# Oblivious aggregation

```sql
SELECT count(*) FROM medical GROUP BY disease
```

---

**Map**

<table>
<thead>
<tr>
<th>12809</th>
<th>Diabetes</th>
</tr>
</thead>
<tbody>
<tr>
<td>29489</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Cancer</td>
</tr>
<tr>
<td>32591</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>

---

**Sort**

Oblivious sort

[CLRS, Leighton ‘85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Map

Sort

Oblivious sort
[CLRS, Leighton ‘85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Map

Sort

Oblivious sort
[CLRS, Leighton '85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Map

Sort

Oblivious sort
[CLRS, Leighton ‘85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Oblivious sort
[CLRS, Leighton ‘85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

<table>
<thead>
<tr>
<th></th>
<th>13744</th>
<th>Cancer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>98329</td>
<td>Cancer</td>
</tr>
<tr>
<td></td>
<td>12809</td>
<td>Diabetes</td>
</tr>
<tr>
<td></td>
<td>29489</td>
<td>Diabetes</td>
</tr>
<tr>
<td></td>
<td>18740</td>
<td>Diabetes</td>
</tr>
<tr>
<td></td>
<td>32591</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

```
SELECT count(*) FROM medical GROUP BY disease
```

The “Diabetes” group is split!
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

The “Diabetes” group is split!

How to aggregate obliviously and in parallel?
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

13744 ... Cancer
98329 ... Cancer
12809 ... Diabetes

29489 ... Diabetes
18740 ... Diabetes
32591 ... Diabetes

The “Diabetes” group is split!

How to aggregate obliviously and in parallel?
It can span over many partitions
Oblivious aggregation

```
SELECT count(*) FROM medical GROUP BY disease
```
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Scan
Oblivious aggregation

```
SELECT count(*) FROM medical GROUP BY disease
```
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Scan

Boundary processing

Partial agg.

Cancer;Diabetes:1

Diabetes;Diabetes:3
Oblivious aggregation

```
SELECT count(*) FROM medical GROUP BY disease
```

Scan

Boundary processing
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Scan

Boundary processing
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Scan

Boundary processing
Oblivious aggregation

```sql
SELECT count(*) FROM medical GROUP BY disease
```
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

```sql
SELECT count(*) FROM medical GROUP BY disease
```

<table>
<thead>
<tr>
<th>Disease</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancer</td>
<td>13744</td>
</tr>
<tr>
<td>Diabetes</td>
<td>12809</td>
</tr>
<tr>
<td>Diabetes</td>
<td>29489</td>
</tr>
<tr>
<td>Cancer</td>
<td>98329</td>
</tr>
<tr>
<td>Diabetes</td>
<td>18740</td>
</tr>
<tr>
<td>Diabetes</td>
<td>32591</td>
</tr>
</tbody>
</table>

Scan | Boundary processing | Scan
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

DUMMY
Cancer: 2
DUMMY

DUMMY
DUMMY
Diabetes: 4
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Oblivious sort
[CLRS, Leighton ‘85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Sort

Oblivious sort
[CLRS, Leighton ‘85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Sort

Oblivious sort
[CLRS, Leighton ‘85]

Final result

Cancer: 2
Diabetes: 4
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Aggregation has two sorts...

Oblivious sort
[CLRS, Leighton ‘85]
Oblivious aggregation

SELECT count(*) FROM medical GROUP BY disease

Aggregation has two sorts...

Can we do better?

Sort

Oblivious sort
[CLRS, Leighton '85]

Final result

Cancer: 2
Diabetes: 4
Opaque components

Oblivious query planning
- Cost model
- Rule-based opt.
- Cost-based opt.

Distributed oblivious operators
- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Opaque components

Oblivious query planning
- Cost model
- Rule-based opt.
- Cost-based opt.

Distributed oblivious operators
- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Rule-based optimization
Rule-based optimization

SELECT count(*)
FROM medical
WHERE age > 30
GROUP BY disease
Rule-based optimization

SELECT count(*)
FROM medical
WHERE age > 30
GROUP BY disease
Insight 1
Insight 1

1. Split each logical operator into smaller Opaque operators
Insight 1

1. Split each logical operator into smaller Opaque operators

2. Take a global view across the plan to remove some Opaque operators
Rule-based optimization

Logical op.

- Aggregation
- Filter
- medical
Rule-based optimization

Opaque op.

Logical op.

```
Aggregation

Filter

medical
```
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

medical
Rule-based optimization

Opaque op.

Logical op.

- Aggregation
- Filter
  - medical
  - medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

Scan

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

Scan

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
</tbody>
</table>
Rule-based optimization

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
</tbody>
</table>

Opaque op.

Logical op.

**Aggregation**

**Filter**

**O-sort**

**Project**

**Scan**

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

Filter

O-sort

Project

Scan

medical

Amanda D. Edwards 40 Diabetes
Robert R. McGowan 56 Diabetes
Kimberly R. Seay 51 Cancer
Dennis G. Bates 32 Diabetes
Donna R. Bridges 26 Diabetes
Ronald S. Ogden 53 Cancer
### Rule-based optimization

#### Logical op.

- Aggregation
- Filter
- O-sort
- Project
- Scan
- medical

#### Opaque op.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

Filter

O-sort

Project

Scan

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

Filter

O-sort

Project

Scan

medical

medical

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Project

Scan

medical

Filter

medical

12809  Amanda D. Edwards  40  Diabetes  0
12649  Robert R. McGowan  56  Diabetes  0
13744  Kimberly R. Seay  51  Cancer  0
18740  Dennis G. Bates  32  Diabetes  0
32591  Donna R. Bridges  26  Diabetes  1
98329  Ronald S. Ogden  53  Cancer  0
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

O-sort

Project

Scan

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

↑

Filter

↑

O-sort

↑

Project

↑

Scan

↑

medical

Filter

medical

12809  |  Amanda D. Edwards  |  40  |  Diabetes  |  0
129489 |  Robert R. McGowan  |  56  |  Diabetes  |  0
13744  |  Kimberly R. Seay   |  51  |  Cancer    |  0
18740  |  Dennis G. Bates    |  32  |  Diabetes  |  0
32591  |  Donna R. Bridges   |  26  |  Diabetes  |  1
98329  |  Ronald S. Ogden    |  53  |  Cancer    |  0
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

Filter

O-sort

Project

Scan

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Diagnosis</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

↑

Filter

↑

O-sort

↑

Project

↑

Scan

↑

medical

Filter

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Diagnosis</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>12949</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

Filter

O-sort

Project

Scan

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td></td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td></td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

<table>
<thead>
<tr>
<th>Medical</th>
<th>Filter</th>
<th>O-sort</th>
<th>Project</th>
<th>Scan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

↑

Filter

↑

O-sort

Project

Scan

medical

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Diagnosis</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Project

Scan

medical

Filter

medical

12809 Amanda D. Edwards 40 Diabetes 0
129489 Robert R. McGowan 56 Diabetes 0
13744 Kimberly R. Seay 51 Cancer 0
18740 Dennis G. Bates 32 Diabetes 0
98329 Ronald S. Ogden 53 Cancer 0
Rule-based optimization

Opaque op.

Logical op.

Aggregation

| 12809 | Amanda D. Edwards | 40 | Diabetes | 0 |
| 29489 | Robert R. McGowan | 56 | Diabetes | 0 |
| 13744 | Kimberly R. Seay  | 51 | Cancer   | 0 |
| 18740 | Dennis G. Bates   | 32 | Diabetes | 0 |
| 98329 | Ronald S. Ogden   | 53 | Cancer   | 0 |

Filter

O-sort

Project

Scan

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

medical

Rule-based optimization

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Age</th>
<th>Diagnosis</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
## Rule-based optimization

### Logical op.

- **Aggregation**
- **Filter**
- **Scan**
- **O-sort**
- **Project**

### Opaque op.

<table>
<thead>
<tr>
<th>Patient ID</th>
<th>Name</th>
<th>Age</th>
<th>Diagnosis</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

- Aggregation
- Filter
- O-sort
- Agg.
- Filter
- Project
- Scan

Medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

O-sort

Agg.

O-sort

Filter

O-sort

Project

Scan

medical

medical

13744   Kimberly R. Seay   51   Cancer   0
98329   Ronald S. Ogden   53   Cancer   0
129489  Robert R. McGowan 56   Diabetes 0
18740   Dennis G. Bates   32   Diabetes 0
12809   Amanda D. Edwards 40   Diabetes 0
Rule-based optimization

Can we remove any sort?
Rule-based optimization

Opaque op.

Logical op.

- O-sort
- Agg.
- Filter
- O-sort
- Project
- Scan

medical

medical

Aggregation

Filter

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

Scan

Project

O-sort

Agg.

O-sort

Filter

O-sort

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

medical

O-sort

Agg.

O-sort

Filter

O-sort

Project

Scan
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

O-sort

Agg.

Sort on 0/1 column

O-sort

Filter

Project

Scan

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

medical

O-sort

Agg.

O-sort

Filter

O-sort

Project

Scan

Sort on 0/1 column
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

Sort on Disease

Sort on 0/1 column
Rule-based optimization

Opaque op.

Logical op.

Aggregation

↑

Filter

↑

medical

↓

medical

Scan

Project

O-sort

Agg.

O-sort

Filter

Sort on Disease

+  

Sort on 0/1 column
Rule-based optimization

Opaque op.

Logical op.

Aggregation

↑

Filter

↑

medical

O-sort

Agg.

O-sort

Filter

O-sort

Project

Scan

↑

medical

→ Sort on Disease

+ 

→ Sort on 0/1 column

=
Rule-based optimization

Opaque op.

Logical op.

```
Aggregation

Filter

medical

O-sort

Agg.

O-sort

Filter

O-sort

Project

Scan

medical

Sort on Disease

+  

Sort on 0/1 column

=  

Sort on (0/1, Disease)
```
Rule-based optimization

Opaque op.

Logical op.

Aggregation

↑

Filter

↑

medical

↑

Aggregation

O-sort

↑

Filter

↑

O-sort

↑

O-sort

↑

Project

↑

Scan

↑

medical

↑

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

O-sort

Agg.

Filter

O-sort

Project

Scan

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

O-sort

Agg.

Filter

O-sort

Project

Scan

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

O-sort

Agg.

Filter

O-sort

Project

Scan

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

O-sorted

Aggregation

Filter

O-sorted

Project

Scan

medical
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

medical

12809 Amanda D. Edwards 40 Diabetes
129489 Robert R. McGowan 56 Diabetes
13744 Kimberly R. Seay 51 Cancer
18740 Dennis G. Bates 32 Diabetes
32591 Donna R. Bridges 26 Diabetes
98329 Ronald S. Ogden 53 Cancer
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

medical

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

<table>
<thead>
<tr>
<th>O-sort</th>
<th>Agg.</th>
<th>Filter</th>
<th>O-sort</th>
<th>Project</th>
<th>Scan</th>
<th>medical</th>
</tr>
</thead>
</table>

| 12809 | Amanda D. Edwards | 40 | Diabetes |
| 29489 | Robert R. McGowan | 56 | Diabetes |
| 13744 | Kimberly R. Seay | 51 | Cancer |
| 18740 | Dennis G. Bates | 32 | Diabetes |
| 32591 | Donna R. Bridges | 26 | Diabetes |
| 98329 | Ronald S. Ogden | 53 | Cancer |
Rule-based optimization

Opaque op.

Logical op.

Aggregation

↑

Filter

↑

medical

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Diagnosis</th>
<th>Diagnosis_Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Opaque op.

Logical op.

Aggregation

Filter

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

medical

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

![Diagram showing logical and opaque operators in a query execution plan. The diagram includes nodes for O-sort, Agg., Filter, Project, Scan, and medical operations. The data table at the bottom contains records for individuals with diabetes and cancer, showing names, ages, and diagnosis statuses.]
Rule-based optimization

Logical op.

- Aggregation
- Filter
- Project
- Scan
- O-sort
- Agg.
- Filter
- O-sort
- medical

Opaque op.

- multi-column sort

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Diabetes Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>129489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
</tbody>
</table>
Rule-based optimization

Logical op.

Opaque op.

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

multicolumn sort

13744  Kimberly R. Seay  51  Cancer  0
98329  Ronald S. Ogden  53  Cancer  0
12809  Amanda D. Edwards  40  Diabetes  0
18740  Dennis G. Bates  32  Diabetes  0
29489  Robert R. McGowan  56  Diabetes  0
32591  Donna R. Bridges  26  Diabetes  1

medical
Rule-based optimization

Logical op.

Opaque op.

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

multi-column sort

13744  Kimberly R. Seay  51  Cancer  0
98329  Ronald S. Ogden  53  Cancer  0
12809  Amanda D. Edwards  40  Diabetes  0
18740  Dennis G. Bates  32  Diabetes  0
29489  Robert R. McGowan  56  Diabetes  0
32591  Donna R. Bridges  26  Diabetes  1
Rule-based optimization

Logical op.

Opaque op.

O-sort
Agg.
Filter
O-sort
Project
Scan

medical

Filter

medical

Aggregation

multi-column sort

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Diagnosis</th>
<th>Overtreatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>32591</td>
<td>Donna R. Bridges</td>
<td>26</td>
<td>Diabetes</td>
<td>1</td>
</tr>
</tbody>
</table>
Rule-based optimization

Logical op.

Opaque op.

Filter

O-sort

Agg.

Aggregation

Project

Scan

medical

medical

13744  Kimberly R. Seay  51  Cancer  0
98329  Ronald S. Ogden  53  Cancer  0
12809  Amanda D. Edwards  40  Diabetes  0
18740  Dennis G. Bates  32  Diabetes  0
29489  Robert R. McGowan  56  Diabetes  0

multi-column sort
Rule-based optimization

Logical op.

Opaque op.

O-sort

Agg.

Filter

O-sort

Project

Scan

medical

medical

multi-column sort

Eliminated one oblivious sort!

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Age</th>
<th>Disease</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>13744</td>
<td>Kimberly R. Seay</td>
<td>51</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>98329</td>
<td>Ronald S. Ogden</td>
<td>53</td>
<td>Cancer</td>
<td>0</td>
</tr>
<tr>
<td>12809</td>
<td>Amanda D. Edwards</td>
<td>40</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>18740</td>
<td>Dennis G. Bates</td>
<td>32</td>
<td>Diabetes</td>
<td>0</td>
</tr>
<tr>
<td>29489</td>
<td>Robert R. McGowan</td>
<td>56</td>
<td>Diabetes</td>
<td>0</td>
</tr>
</tbody>
</table>
Opaque components

Oblivious query planning
- Cost model
- Rule-based opt.
- Cost-based opt.

Distributed oblivious operators
- Oblivious Filter
- Oblivious Aggregation
- Oblivious Join

Computation verification

Data encryption and authentication
Observation: not all tables are sensitive
Observation: not all tables are sensitive

Hospitalized patients
- P_ID
- D_ID
- Name
- Age

Disease
- D_ID
- Name
- G_ID

Medication
- M_ID
- D_ID
- Name
- Cost
Observation: not all tables are sensitive.
Observation: not all tables are sensitive

Opaque can operate in *mixed sensitivity*: sensitive tables are run with oblivious operators
Observation: not all tables are sensitive

A → B → C → D

C is locked
Observation: not all tables are sensitive
Observation: not all tables are sensitive

Not oblivious!
Observation: not all tables are sensitive
Observation: not all tables are sensitive

Sensitivity propagation: propagate obliviousness from leaf to root
Observation: not all tables are sensitive

Sensitivity propagation: propagate obliviousness from leaf to root
Observation: not all tables are sensitive

Sensitivity propagation: propagate obliviousness from leaf to root
Insight 2

Sensitivity propagation introduces a new dimension to query optimization
Cost-based optimization

Find the least costly medication for each patient
Cost-based optimization

Find the least costly medication for each patient

Assumption: $|P| < |D| < |M|$
Cost-based optimization

Find the least costly medication for each patient

Assumption: |P| < |D| < |M|

SELECT p_name, d_name, med_cost
FROM patient, disease,
    (SELECT d_id, min(cost) AS med_cost
     FROM medication
     GROUP BY d_id) AS med
WHERE disease.d_id = patient.d_id
    AND disease.d_id = med.d_id
Cost-based optimization

Find the least costly medication for each patient

Assumption: \(|P| < |D| < |M|\)

```sql
SELECT p_name, d_name, med_cost
FROM patient, disease,
    (SELECT d_id, min(cost) AS med_cost
     FROM medication
     GROUP BY d_id) AS med
WHERE disease.d_id = patient.d_id
  AND disease.d_id = med.d_id
```
Cost-based optimization

Find the least costly medication for each patient

Assumption: |P| < |D| < |M|

```
SELECT p_name, d_name, med_cost
FROM patient, disease,
     (SELECT d_id, min(cost) AS med_cost
      FROM medication
      GROUP BY d_id) AS med
WHERE disease.d_id = patient.d_id
  AND disease.d_id = med.d_id
```
Cost-based optimization

SQL optimizer with new cost:
Cost-based optimization

SQL optimizer with new cost:

More selective non-oblivious join
Cost-based optimization

SQL optimizer with new cost:

More selective
non-oblivious join
Cost-based optimization

SQL optimizer with new cost and sensitivity propagation:
Cost-based optimization

SQL optimizer with new cost and sensitivity propagation:

Fewer oblivious joins
Cost-based optimization

SQL optimizer with new cost and sensitivity propagation:

Fewer oblivious joins
Evaluation setup
Evaluation setup

- Single machine experiments:
  - Intel Xeon E3-1280 v5, 4 cores, 64 GB RAM
  - Intel SGX: 128 MB of enclave page cache (EPC)
Evaluation setup

- Single machine experiments:
  - Intel Xeon E3-1280 v5, 4 cores, 64 GB RAM
  - Intel SGX: 128 MB of enclave page cache (EPC)
- Distributed experiments
  - A cluster of 5 SGX machines
Evaluation
Evaluation

- How does Opaque compare to Spark SQL?
Evaluation

• How does Opaque compare to Spark SQL?
  • Big Data Benchmark (BDB); 4 queries total
Evaluation

• How does Opaque compare to Spark SQL?
  • Big Data Benchmark (BDB); 4 queries total
    • Queries 1, 2, 3: filter, aggregation, join
Evaluation

• How does Opaque compare to Spark SQL?
  • Big Data Benchmark (BDB); 4 queries total
    • Queries 1, 2, 3: filter, aggregation, join
    • 1 million records
Evaluation

• How does Opaque compare to Spark SQL?
  • Big Data Benchmark (BDB); 4 queries total
    • Queries 1, 2, 3: filter, aggregation, join
    • 1 million records

• How does Opaque compare to state-of-the-art oblivious systems?
Evaluation

• How does Opaque compare to Spark SQL?
  • Big Data Benchmark (BDB); 4 queries total
    • Queries 1, 2, 3: filter, aggregation, join
    • 1 million records

• How does Opaque compare to state-of-the-art oblivious systems?
  • GraphSC (oblivious graph analytics)
Evaluation

• How does Opaque compare to Spark SQL?
  • Big Data Benchmark (BDB); 4 queries total
    • Queries 1, 2, 3: filter, aggregation, join
    • 1 million records

• How does Opaque compare to state-of-the-art oblivious systems?
  • GraphSC (oblivious graph analytics)
    • PageRank
Big Data Benchmark (distributed)
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

Runtime (s)

- 100
- 10
- 1
- 0.1
- 0.01

Query number

Query 1  Query 2  Query 3

Spark SQL
Opaque
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

Runtime (s)

Query number

- Query 1
- Query 2
- Query 3

- Spark SQL
- Opaque
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

Runtime (s)

Spark SQL
Opaque

Query number

Query 1  Query 2  Query 3
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

Overhead: -0.47x to 2.3x
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

+ Obliviousness

Overhead: -0.47x to 2.3x
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

Runtime (s)

<table>
<thead>
<tr>
<th>Query number</th>
<th>Spark SQL</th>
<th>Opaque</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>0.01</td>
<td>0.1</td>
</tr>
<tr>
<td>Query 2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Query 3</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

+ Obliviousness

Runtime (s)

<table>
<thead>
<tr>
<th>Query number</th>
<th>Spark SQL</th>
<th>Opaque</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>0.01</td>
<td>0.1</td>
</tr>
<tr>
<td>Query 2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Query 3</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Overhead: -0.47x to 2.3x
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

Overhead: -0.47x to 2.3x
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

+ Obliviousness

Runtime (s)

<table>
<thead>
<tr>
<th>Query number</th>
<th>Spark SQL</th>
<th>Opaque</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>0.01</td>
<td>0.1</td>
</tr>
<tr>
<td>Query 2</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Query 3</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

Overhead: -0.47x to 2.3x
Big Data Benchmark (distributed)

Data encryption, authentication, computation verification

<table>
<thead>
<tr>
<th>Query number</th>
<th>Spark SQL</th>
<th>Opaque</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>Query 2</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>Query 3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Query 4</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>Query 5</td>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

+ Obliviousness

<table>
<thead>
<tr>
<th>Query number</th>
<th>Spark SQL</th>
<th>Opaque</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>0.01</td>
<td>0.02</td>
</tr>
<tr>
<td>Query 2</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>Query 3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Query 4</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>Query 5</td>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

Overhead: -0.47x to 2.3x

Overhead: 21x to 45x
PageRank: comparison with GraphSC (single machine)
Conclusion
Conclusion

• Opaque is an oblivious and encrypted distributed analytics platform
Conclusion

• Opaque is an oblivious and encrypted distributed analytics platform

• Open source: github.com/ucbrise/opaque
Conclusion

• Opaque is an oblivious and encrypted distributed analytics platform

• Open source: github.com/ucbrise/opaque

• IBM collaboration
Conclusion

• Opaque is an oblivious and encrypted distributed analytics platform

• Open source: github.com/ucbrise/opaque

• IBM collaboration

• Future work
Conclusion

- Opaque is an oblivious and encrypted distributed analytics platform
- Open source: github.com/ucbrise/opaque
- IBM collaboration
- Future work
  - Federated setting