Proteus:
Autonomous Adaptive Storage for Mixed Workloads

Michael Abebe mtabebe@uwaterloo.ca
Horatiu Lazu
Khuzaima Daudjee

June 2022

tiny.cc/proteus

UNIVERSITY OF WATERLOO

Mixed Workloads

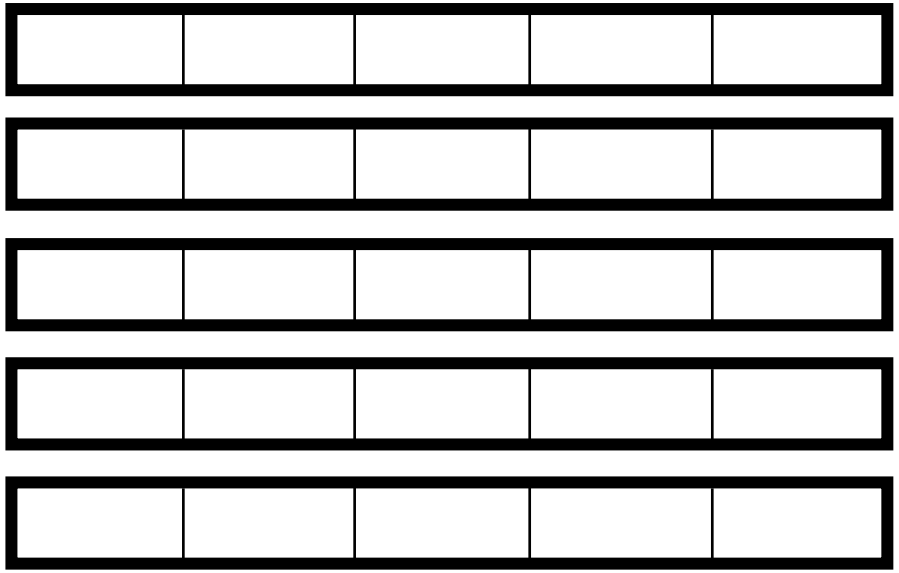
SELECT
book, **SUM**(qnt)
GROUP BY book

INSERT
(104, LoTR, John,
2, $21)

UNIVERSITY OF WATERLOO

Mixed Workloads and Storage

Row Layout



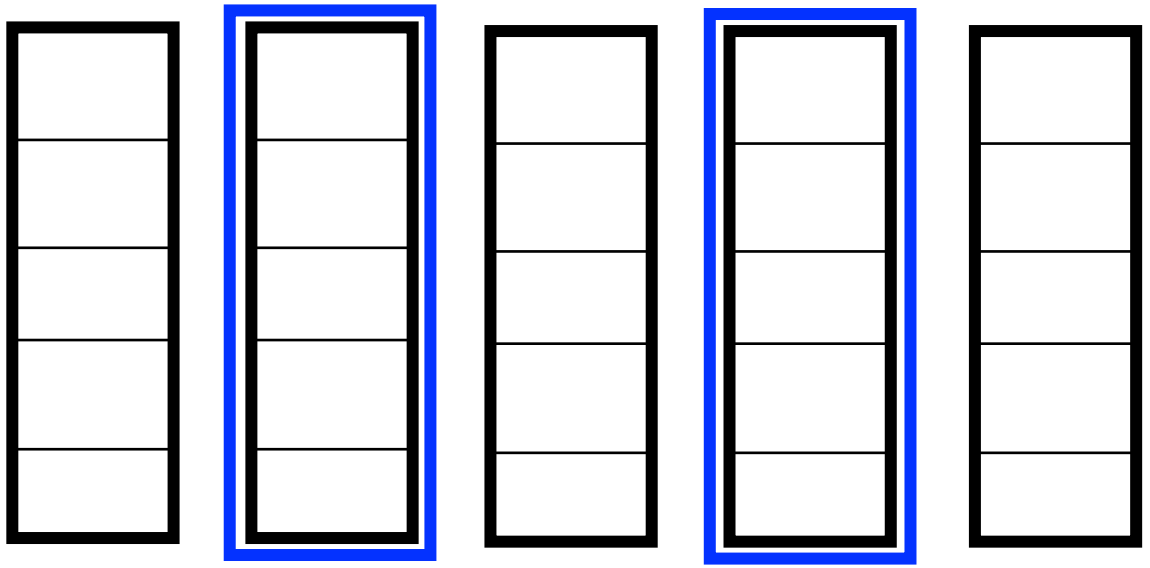
Updates (OLTP)

INSERT
(104, LoTR, John,
2, $21)

UNIVERSITY OF WATERLOO

Mixed Workloads and Storage

Columnar Layout

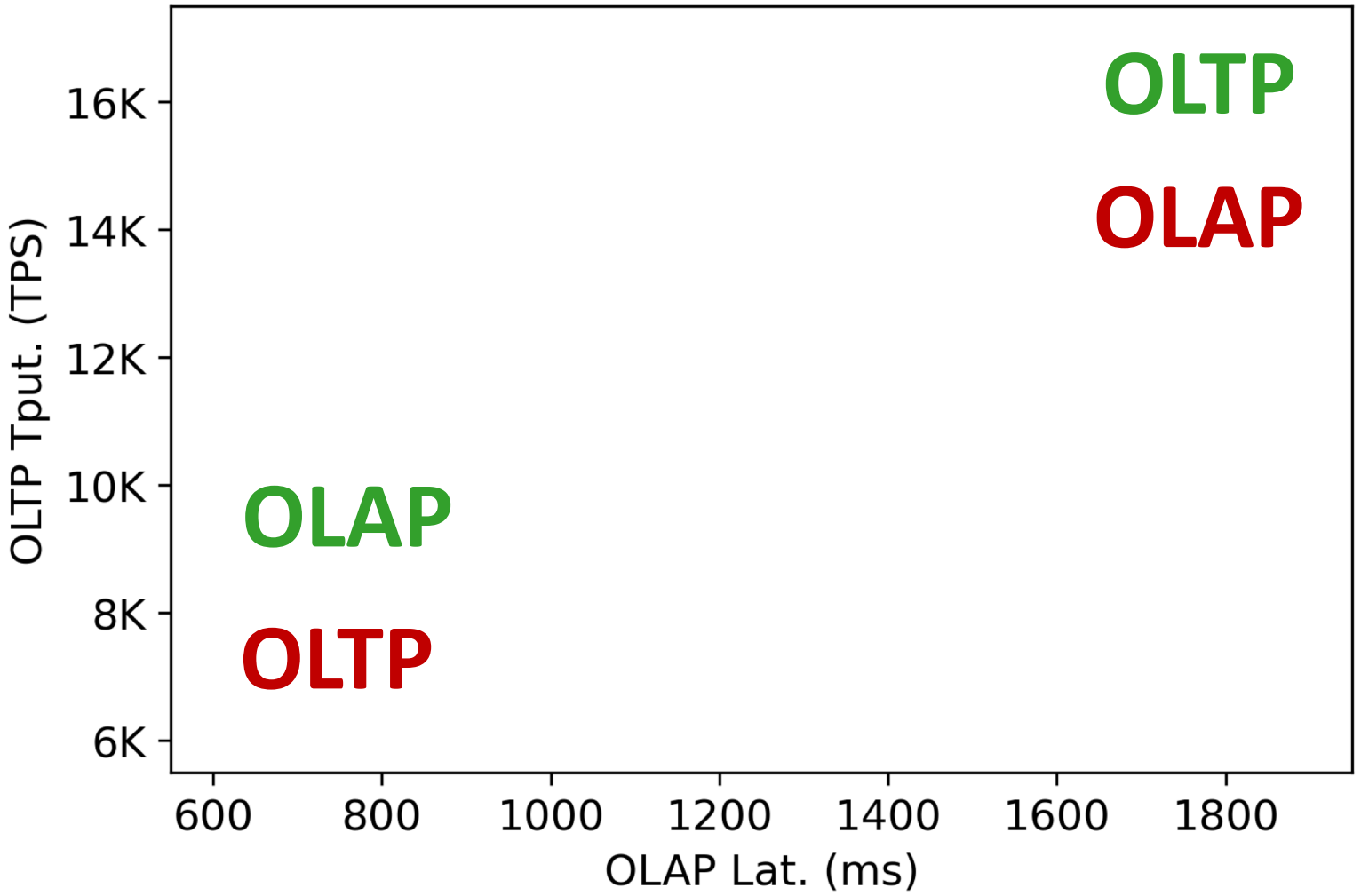


SELECT
book, **SUM**(qnt)
GROUP BY book

Analytics (OLAP)

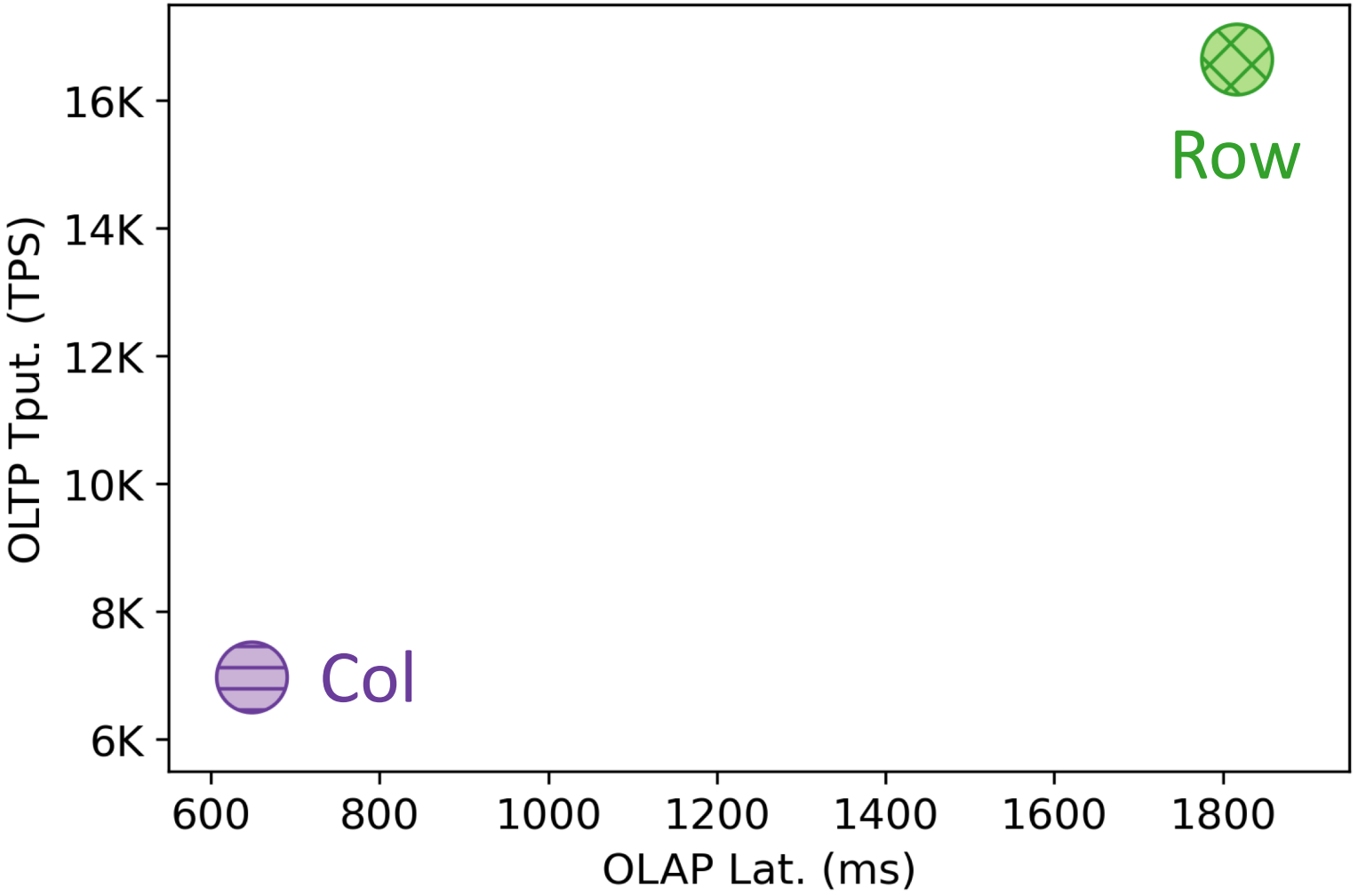
UNIVERSITY OF WATERLOO

Performance Trade-Off



UNIVERSITY OF WATERLOO

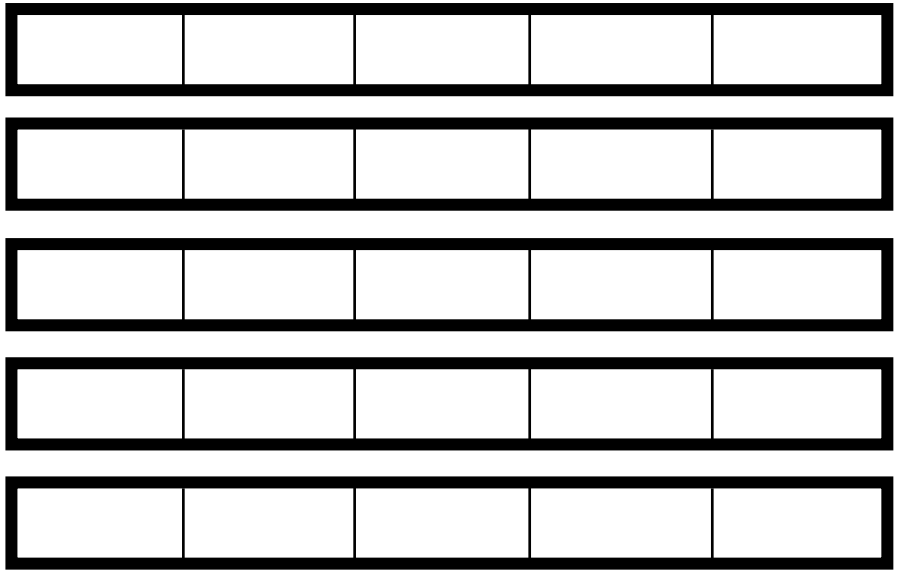
Performance Trade-Off



UNIVERSITY OF WATERLOO

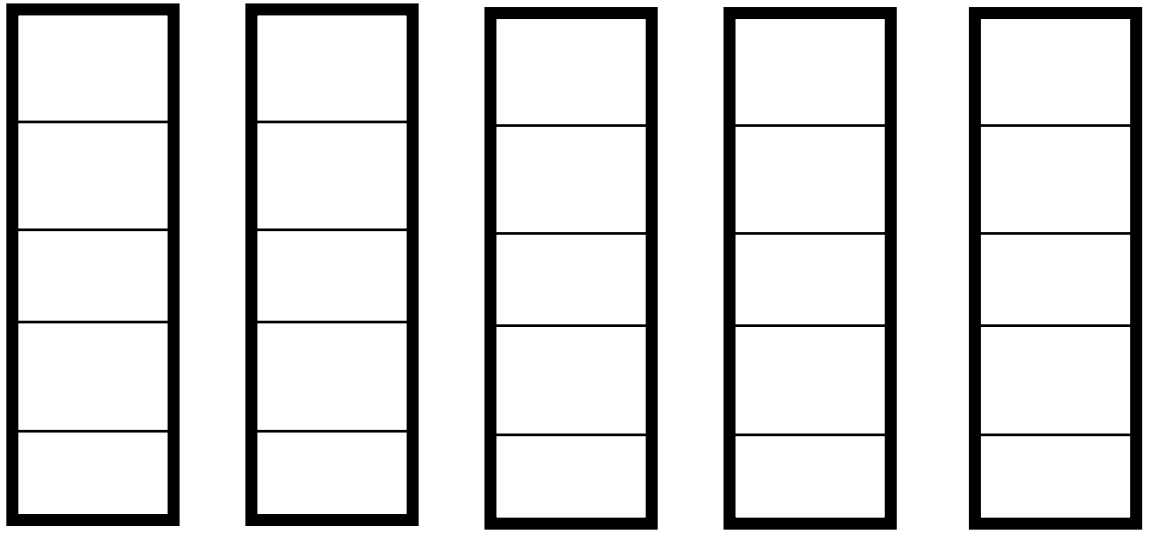
Keep Data in Both Layouts

Row Layout



Updates

Columnar Layout



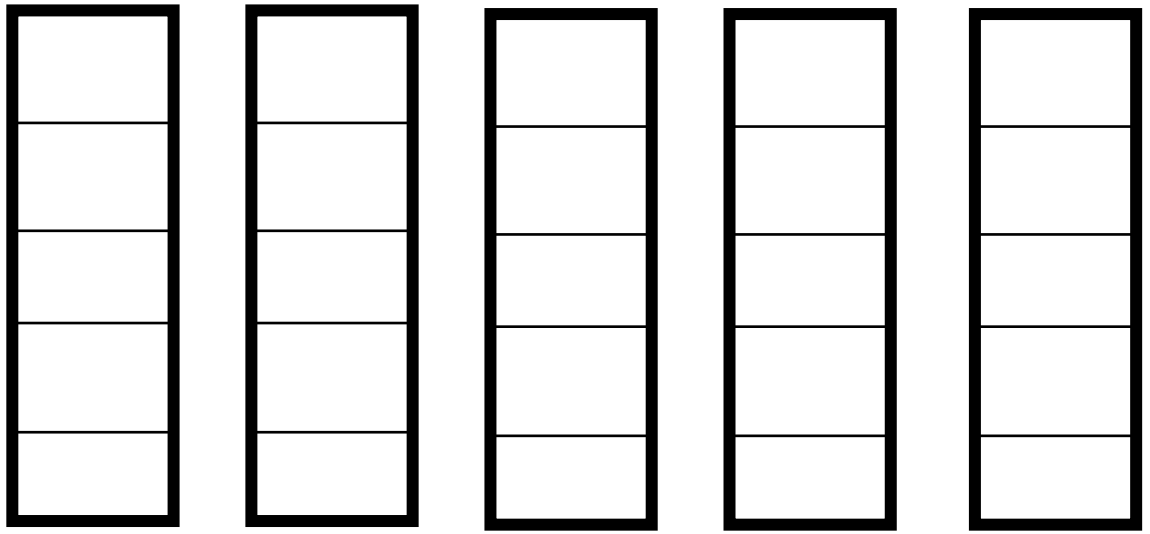
Analytics

UNIVERSITY OF WATERLOO

Keep Data in Both Layouts

Row Layout

Columnar Layout



Missing data

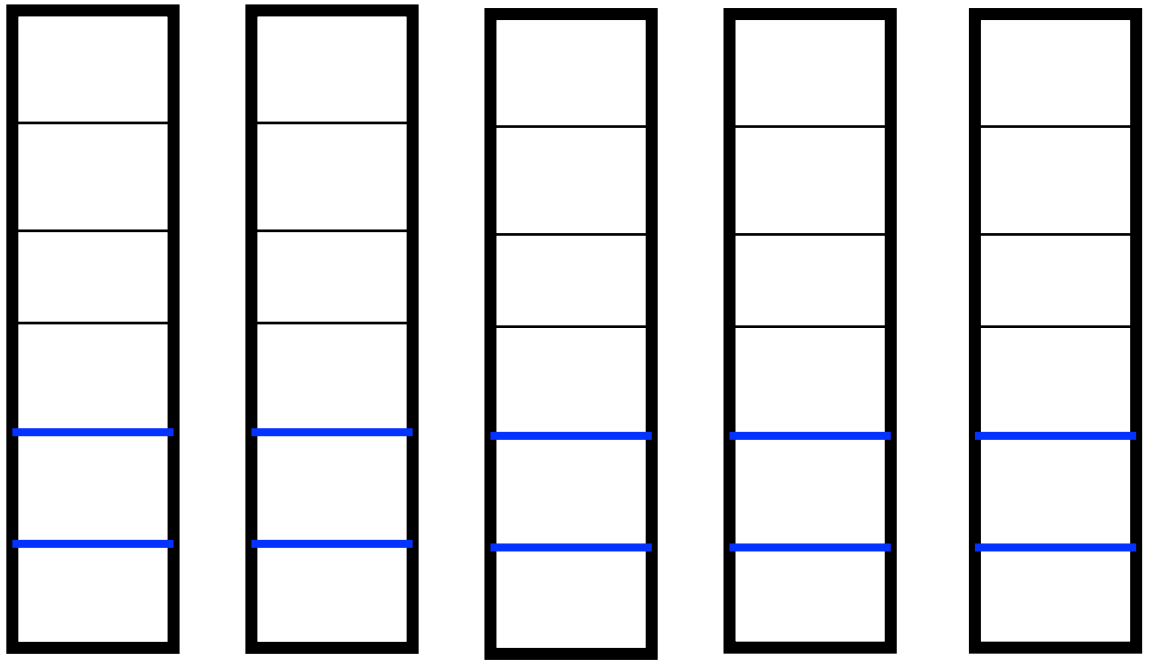
UNIVERSITY OF WATERLOO

Keep Data in Both Layouts

Row Layout

Slow Updates

Columnar Layout

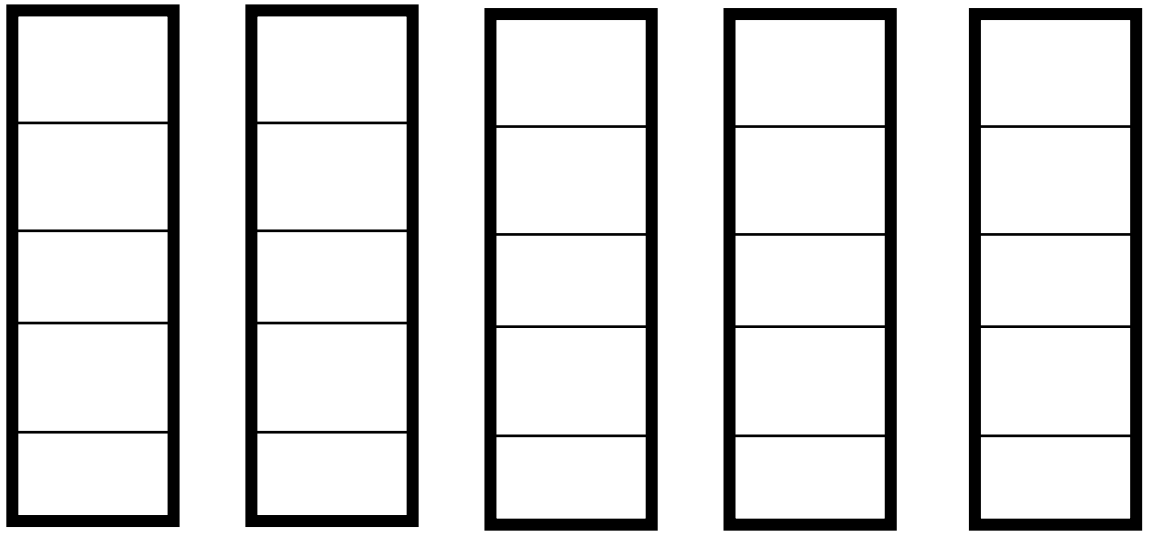


UNIVERSITY OF WATERLOO

Keep Data in Both Layouts

Row Layout

Columnar Layout



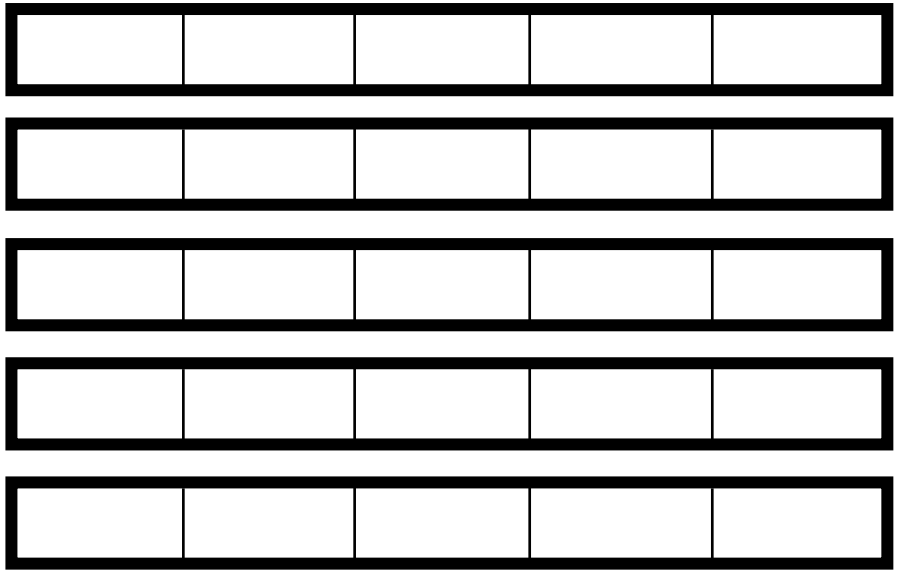
Increased Storage Costs

Missing recent data

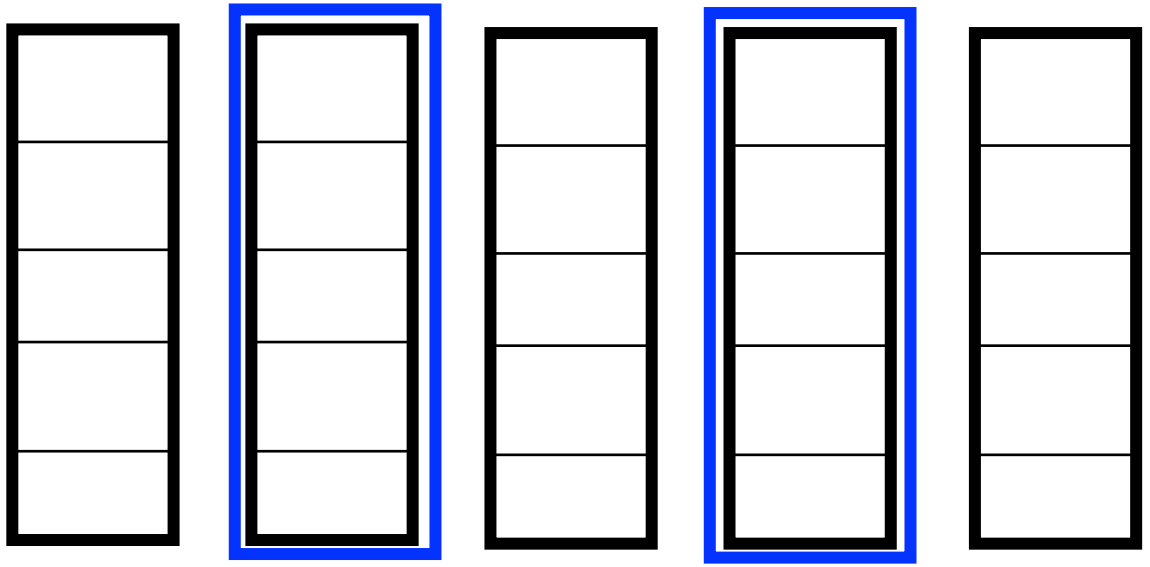
UNIVERSITY OF WATERLOO

Which data in which layout?

Row Layout



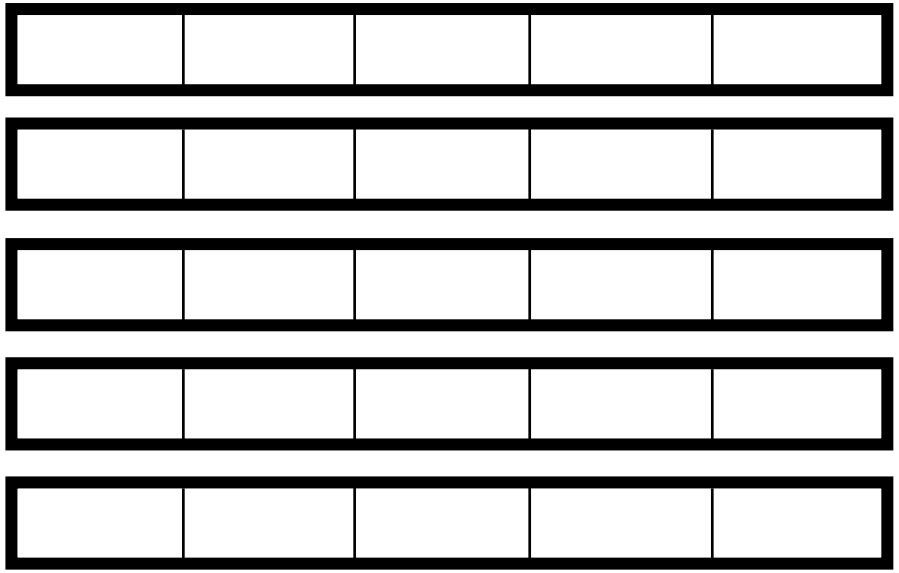
Columnar Layout



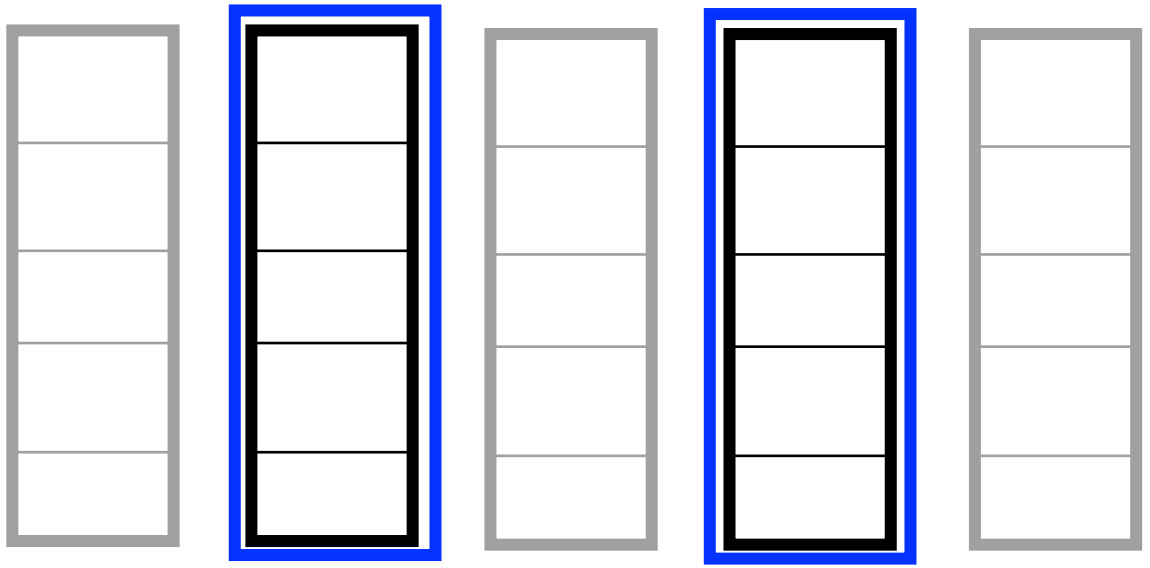
```
SELECT
   book, SUM( qnt )
GROUP BY book
```

Which data in which layout?

Row Layout



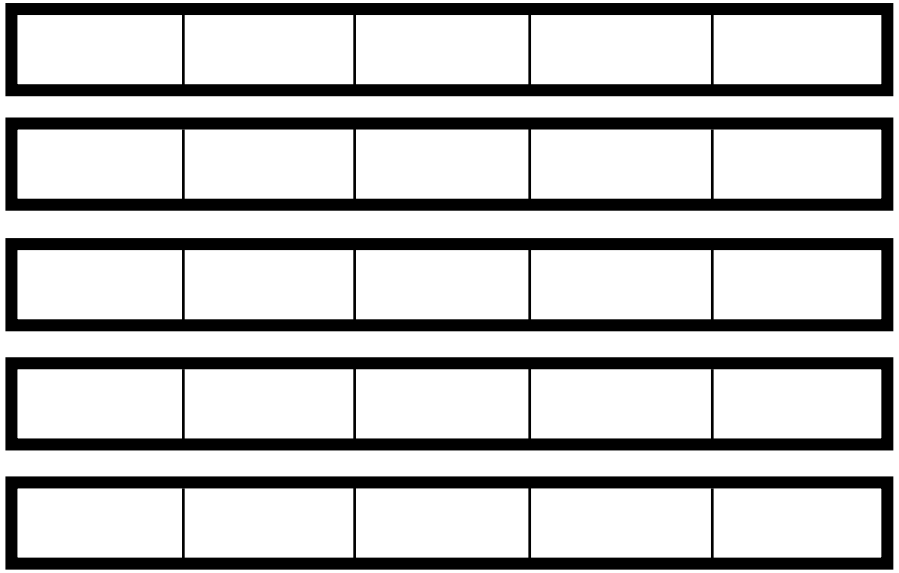
Columnar Layout



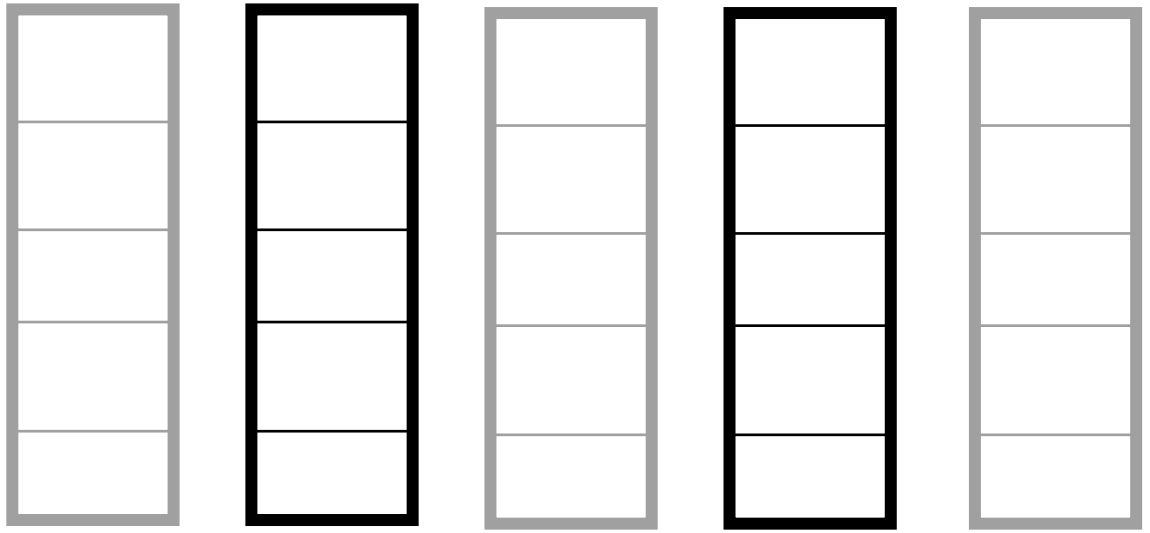
```
SELECT
   book, SUM( qnt )
GROUP BY book
```

Which data in which layout?

Row Layout



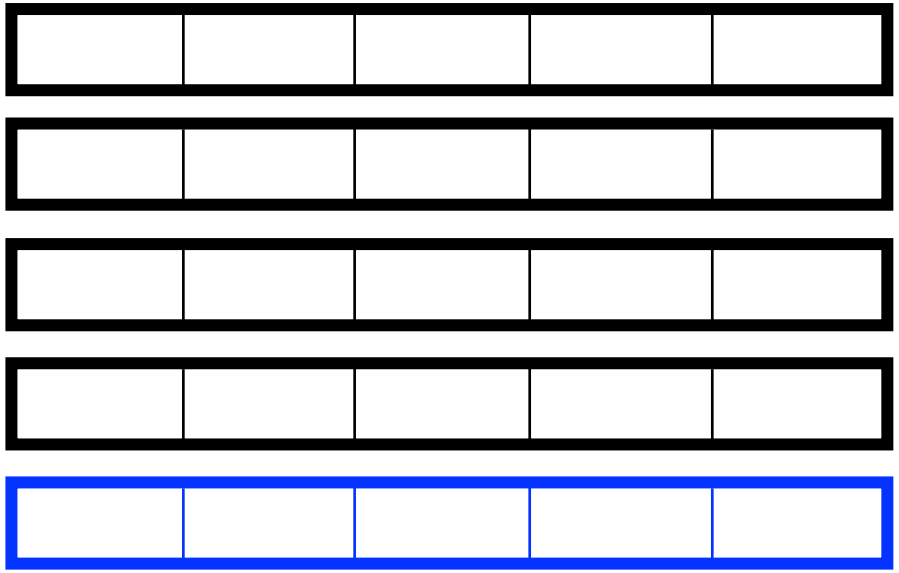
Columnar Layout



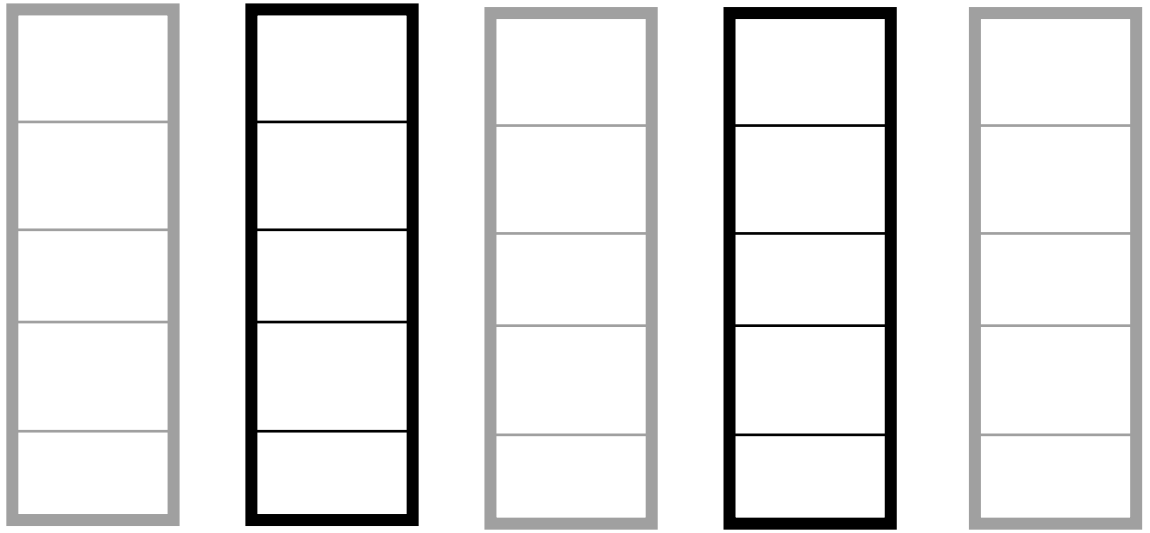
UNIVERSITY OF WATERLOO

Which data in which layout?

Row Layout



Columnar Layout

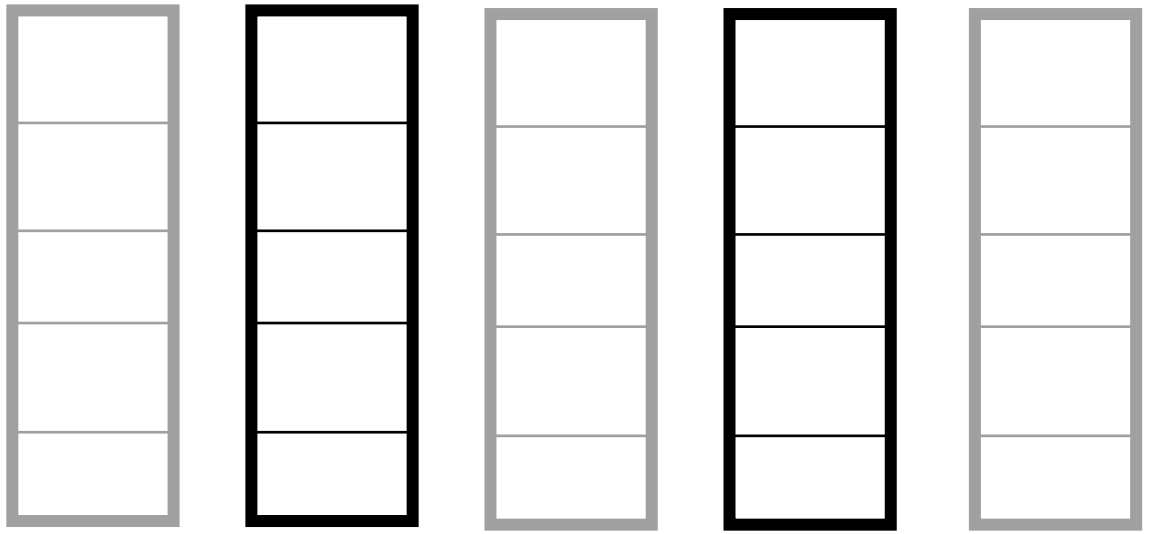


UNIVERSITY OF WATERLOO

Which data in which layout?

Row Layout

Columnar Layout

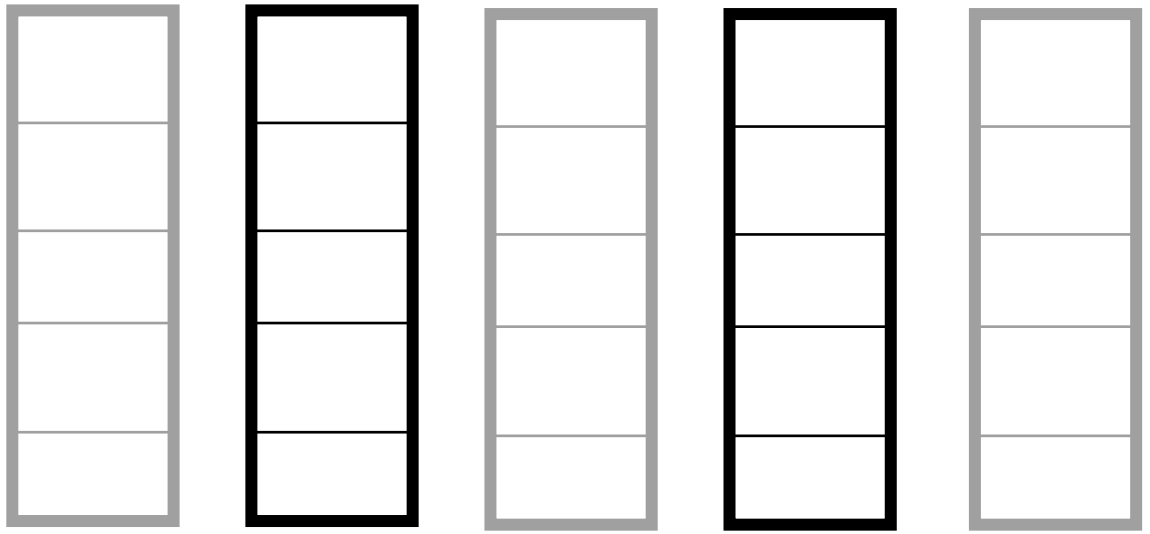


UNIVERSITY OF WATERLOO

Which data in which layout?

Row Layout

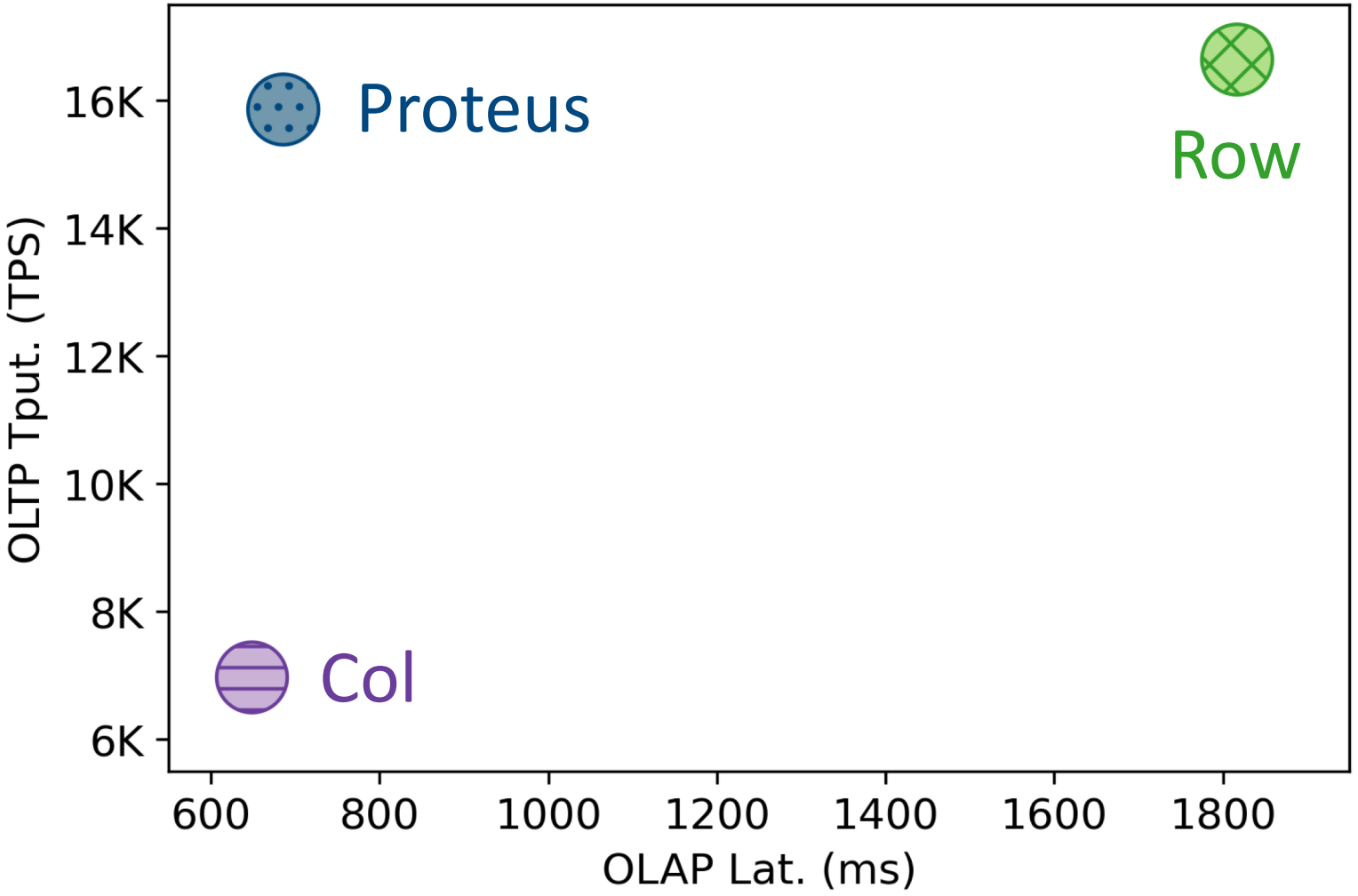
Columnar Layout



Need for autonomous adaptation

UNIVERSITY OF WATERLOO

Proteus – Best of Both



UNIVERSITY OF WATERLOO

Proteus Goals

Selectively store data in **different** layouts

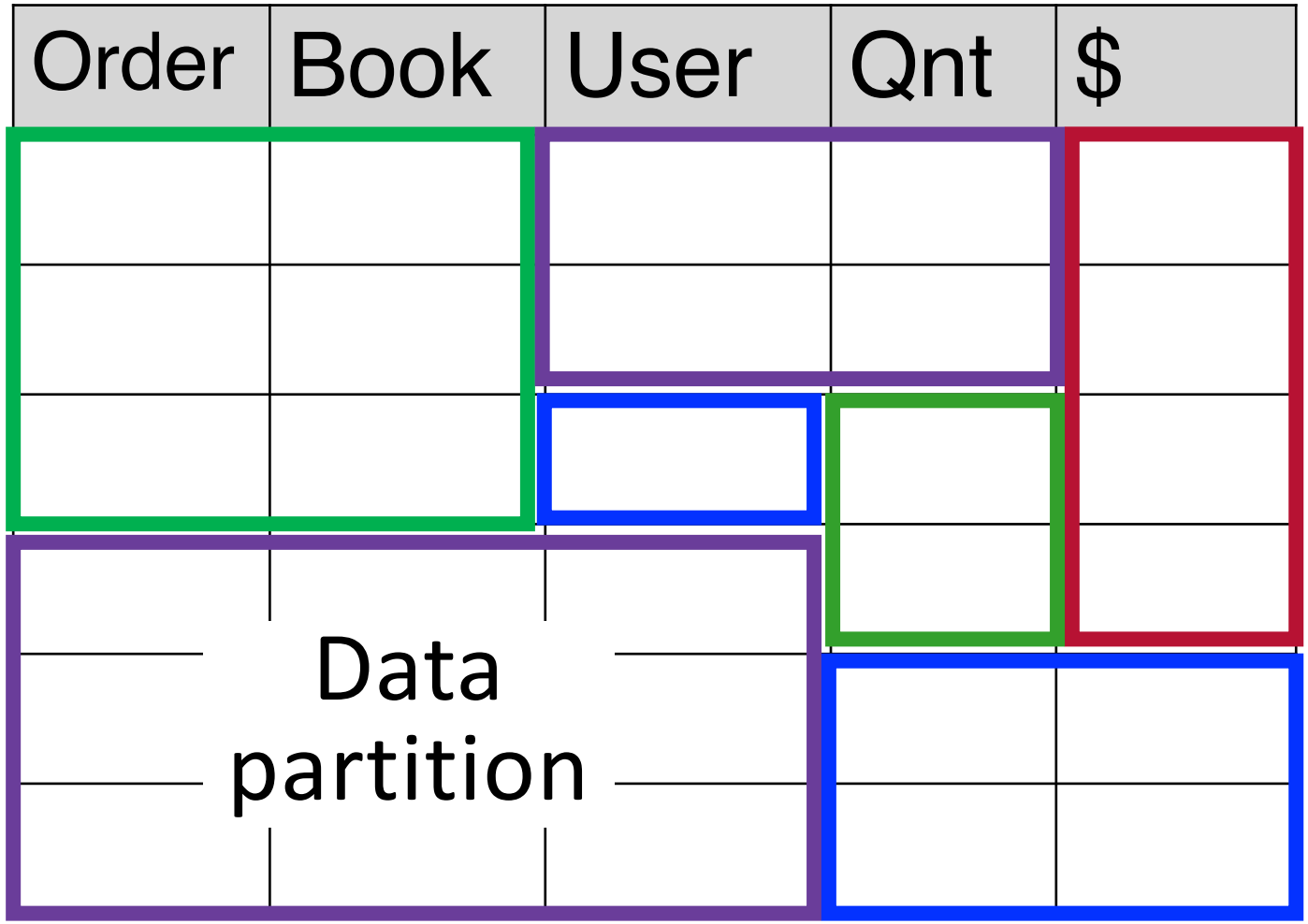
Leverage layout **specific optimizations**

System makes **decisions** based on **workload**

Scale-Out Distributed System

UNIVERSITY OF WATERLOO

Proteus Decisions



Storage layout

Master/replica(s)

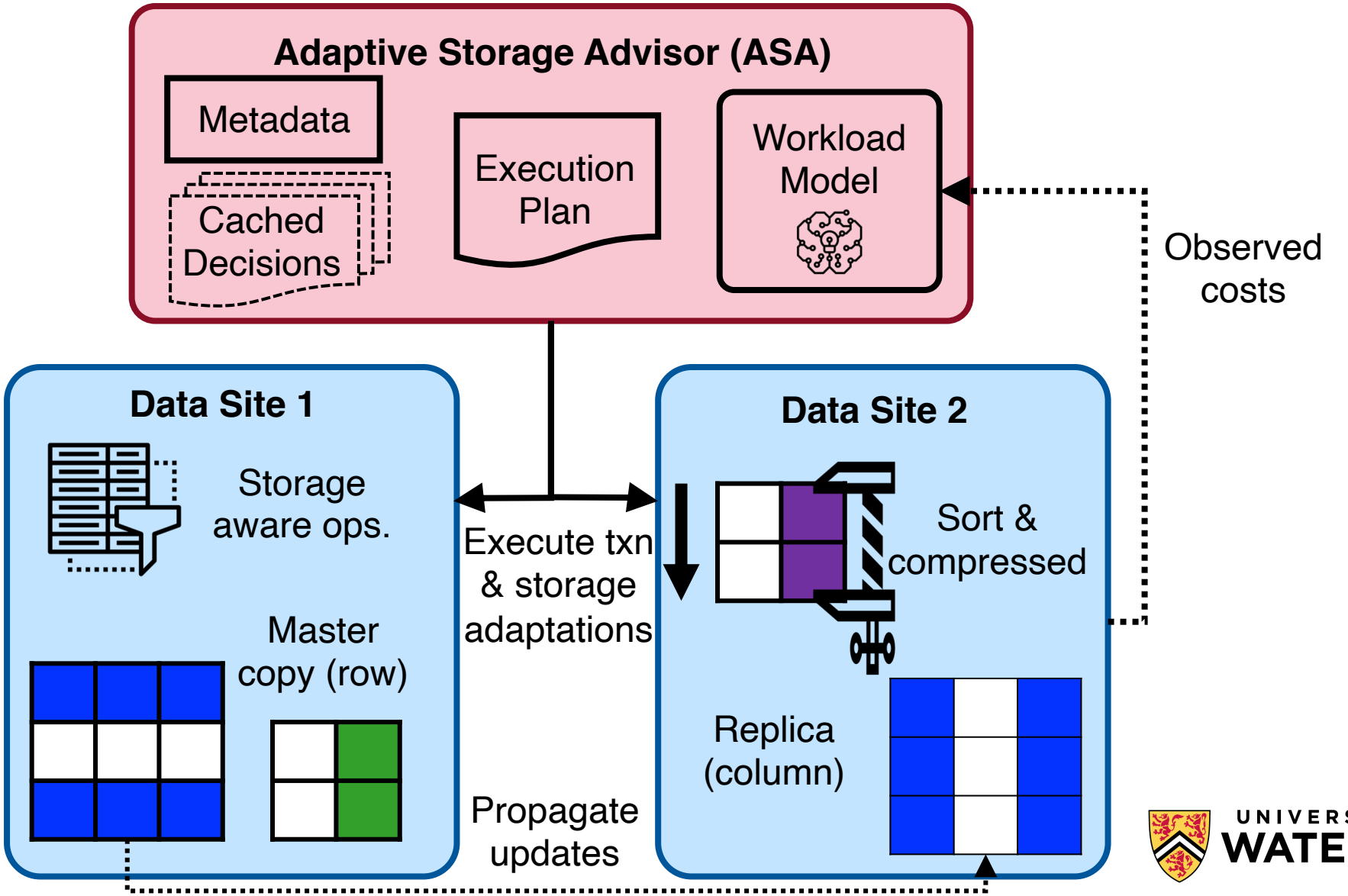
Txn execution

How to **partition?**

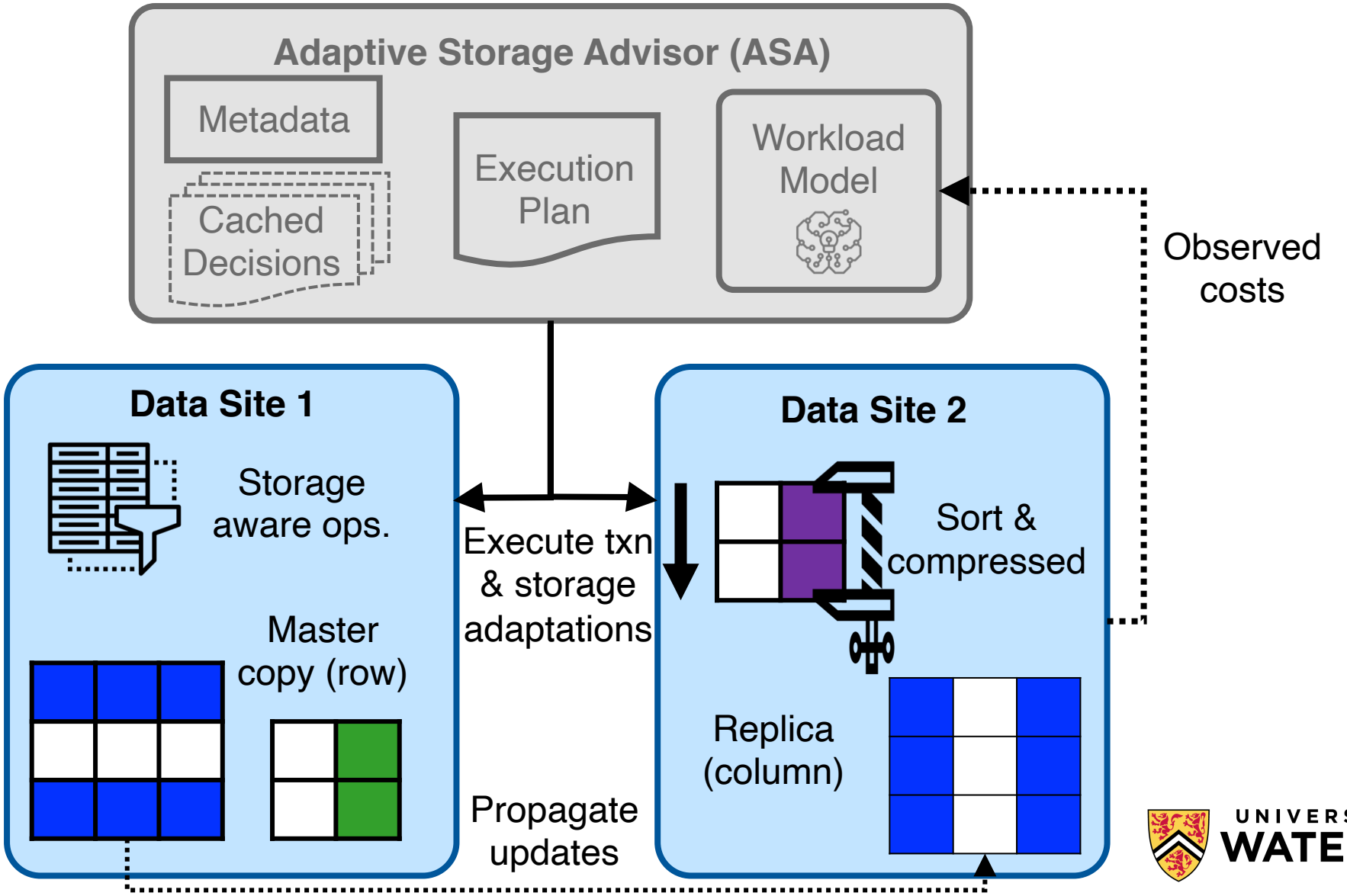
When & **what** to change

UNIVERSITY OF WATERLOO

Proteus Architecture



Efficient Execution



Storage-Aware Operators

Order	Book	User	Qnt	$
100	HP	Alice	7	$15
101	Drac	Ted	90	$3
102	SQL	Geoff	1	$100
103	HP	Jean	3	$15

```
SELECT $
WHERE
  book == 'HP'
  or  QNT > 50
```

Storage-Aware Operators

Use **column-specific** operators

Merge bitmaps for predicates, extract data

Book	Qnt	Book == HP	Qnt > 50	Merged
Drac	90	0	1	1
HP	7	1	0	1
HP	3	1	0	1
SQL	1	0	0	0

```
SELECT $
WHERE
  book == 'HP'
  or  QNT > 50
```

Storage-Aware Operators

Order	Book	User	Qnt	$
100	HP	Alice	7	$15
101	Drac	Ted	90	$3
102	SQL	Geoff	1	$100
103	HP	Jean	3	$15

```
SELECT
   book, SUM( qnt )
GROUP BY book
```

Book	Qnt

UNIVERSITY OF WATERLOO

Storage-Aware Operators

Order	Book	User	Qnt	$
100	HP	Alice	7	$15
101	Drac	Ted	90	$3
102	SQL	Geoff	1	$100
103	HP	Jean	3	$15

SELECT
book, **SUM**(qnt)
GROUP BY book

Book	Qnt
HP	7

UNIVERSITY OF WATERLOO

Storage-Aware Operators

Order	Book	User	Qnt	$
100	HP	Alice	7	$15
101	Drac	Ted	90	$3
102	SQL	Geoff	1	$100
103	HP	Jean	3	$15

SELECT
book, **SUM**(qnt)
GROUP BY book

Book	Qnt
HP	7
Drac	90

UNIVERSITY OF WATERLOO

Storage-Aware Operators

Order	Book	User	Qnt	$
100	HP	Alice	7	$15
101	Drac	Ted	90	$3
102	SQL	Geoff	1	$100
103	HP	Jean	3	$15

SELECT
book, **SUM**(qnt)
GROUP BY book

Book	Qnt
HP	7
SQL	1
Drac	90

UNIVERSITY OF WATERLOO

Storage-Aware Operators

Order	Book	User	Qnt	$
100	HP	Alice	7	$15
101	Drac	Ted	90	$3
102	SQL	Geoff	1	$100
103	HP	Jean	3	$15

```
SELECT
   book, SUM( qnt )
GROUP BY book
```

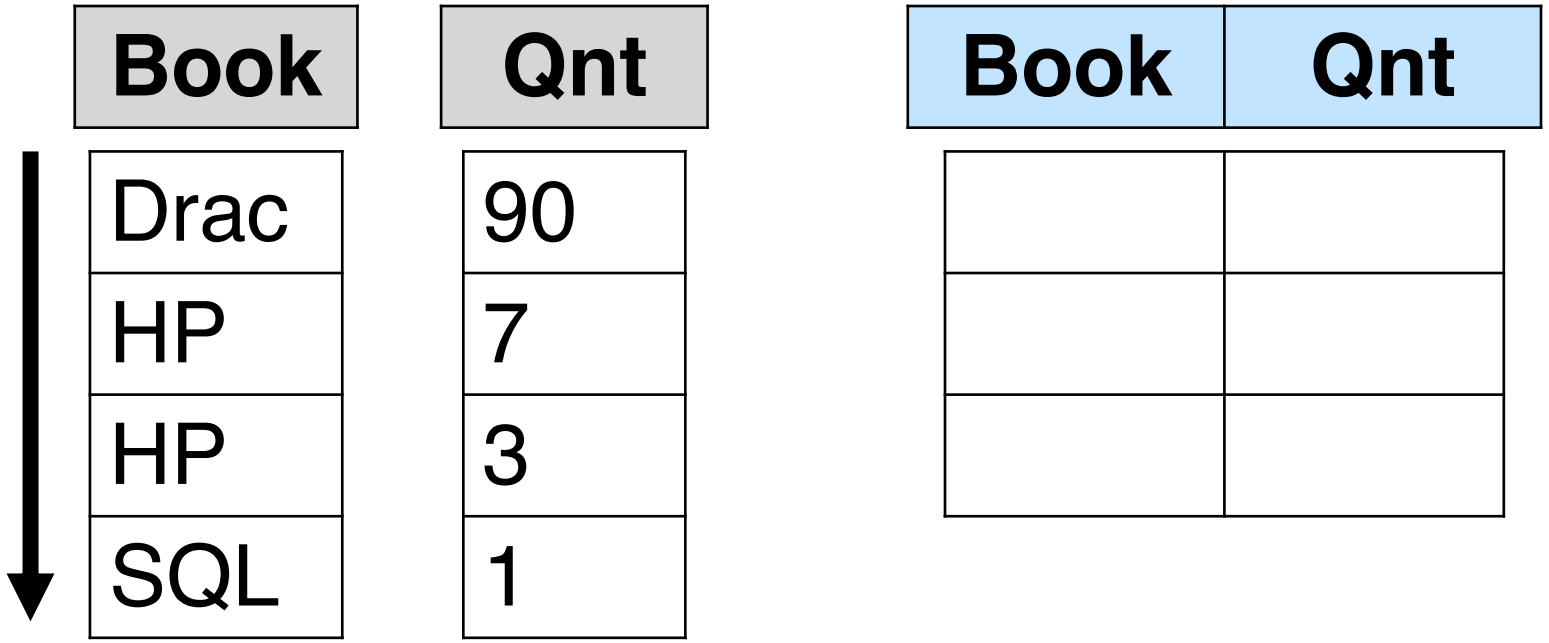
Book	Qnt
HP	10
SQL	1
Drac	90

UNIVERSITY OF WATERLOO

Storage-Aware Operators

Use **column-specific** operators

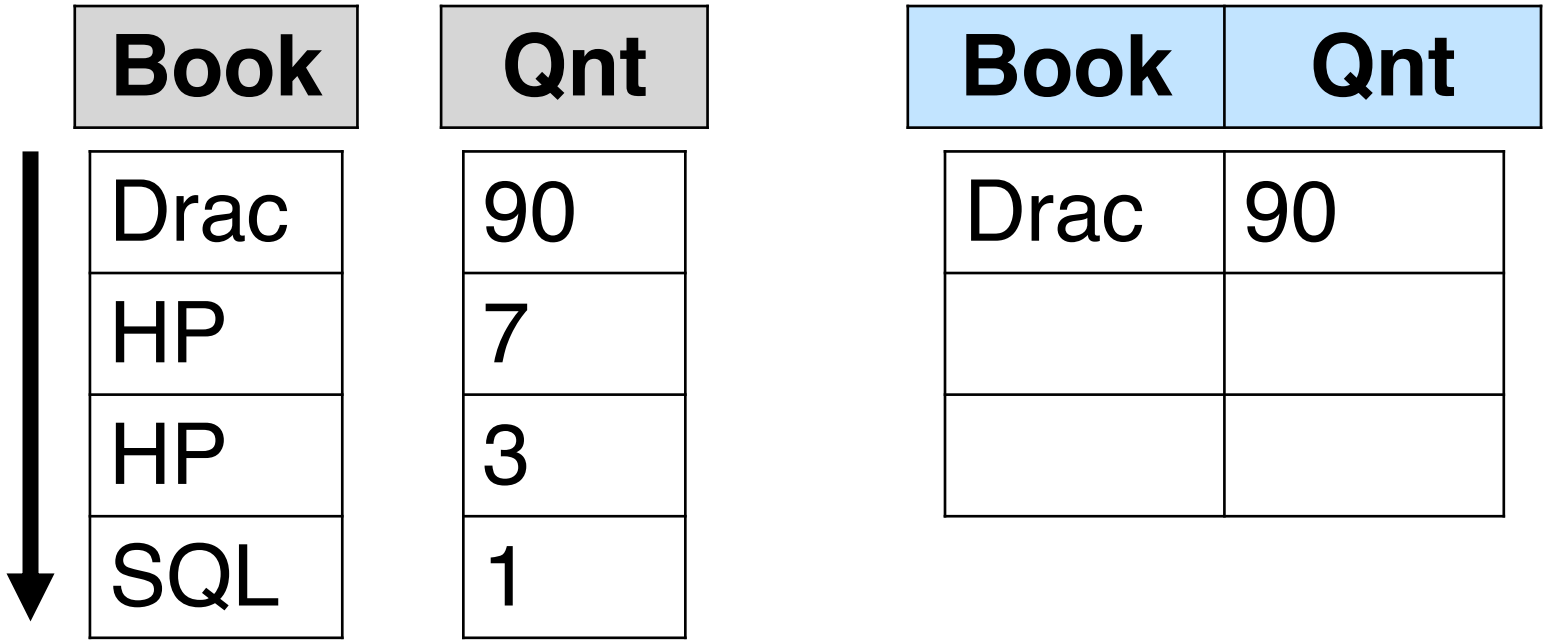
```
SELECT
   book, SUM( qnt )
GROUP BY book
```



Storage-Aware Operators

Use **column-specific** operators

```
SELECT
   book, SUM( qnt )
GROUP BY book
```



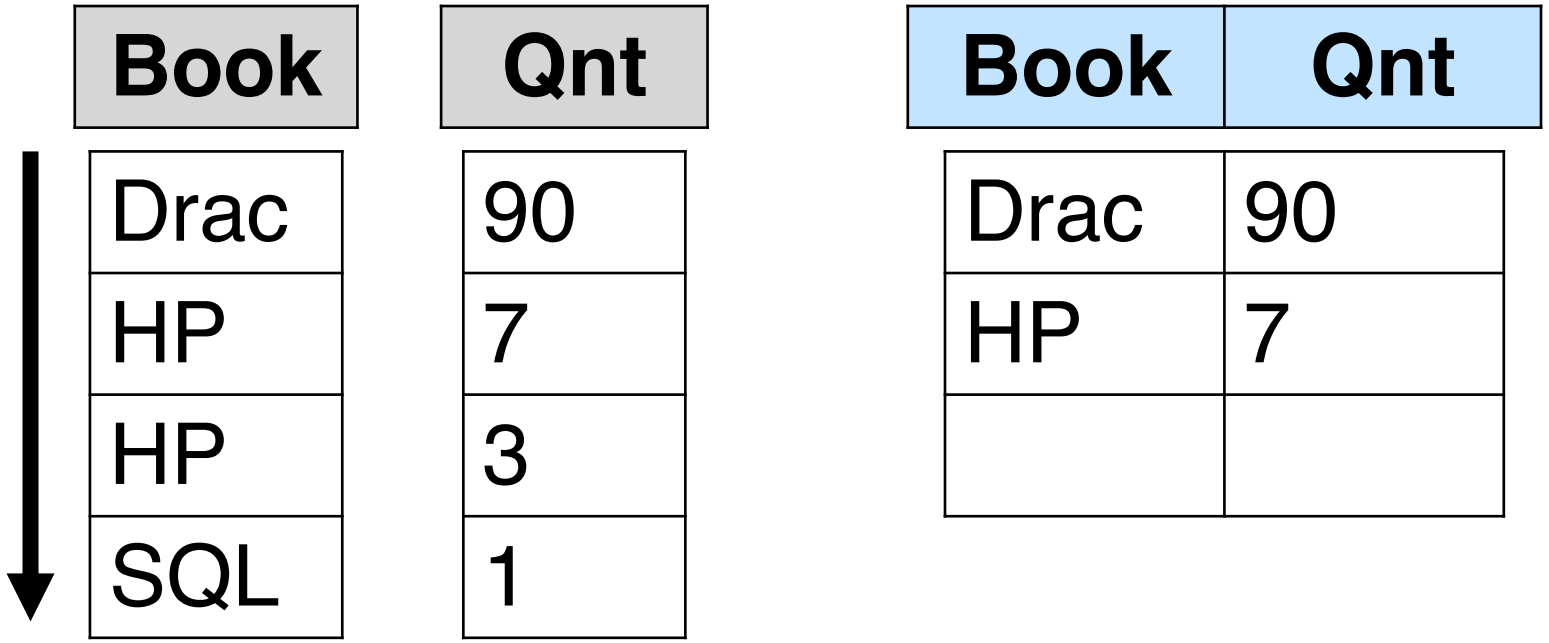
Book	Qnt
Drac	90
HP	7
HP	3
SQL	1

Book	Qnt
Drac	90

Storage-Aware Operators

Use **column-specific** operators

```
SELECT
   book, SUM( qnt )
GROUP BY book
```



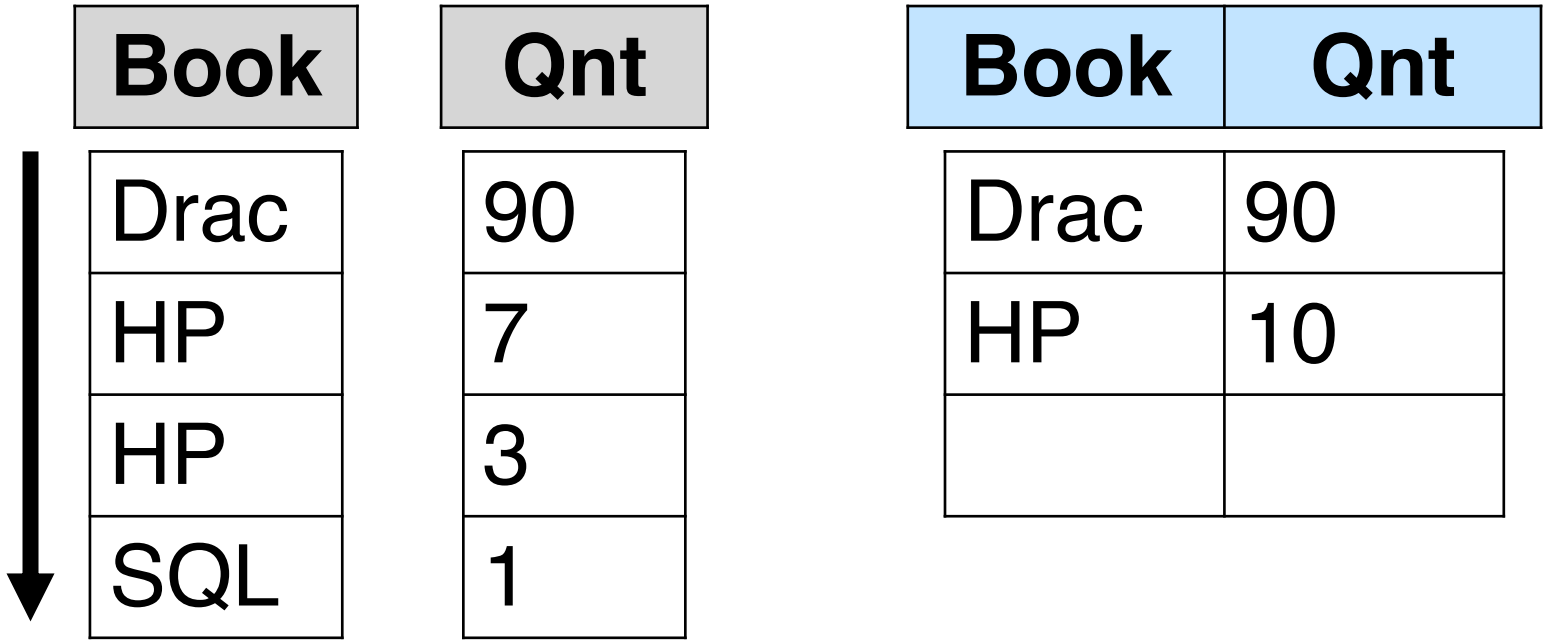
Book	Qnt
Drac	90
HP	7
HP	3
SQL	1

Book	Qnt
Drac	90
HP	7

Storage-Aware Operators

Use **column-specific** operators

```
SELECT
   book, SUM( qnt )
GROUP BY book
```



Book	Qnt
Drac	90
HP	7
HP	3
SQL	1

Book	Qnt
Drac	90
HP	10

Storage-Aware Operators

Use **column-specific** operators

```
SELECT
   book, SUM( qnt )
GROUP BY book
```

Operate **directly** over **sorted** and **compressed** data

Book	Qnt
Drac	90
HP	7
HP	3
SQL	1

Book	Qnt
Drac	90
HP	10
SQL	1

UNIVERSITY OF WATERLOO

Storage-Aware Operators

Use **column-specific** operators

Operate **directly** over **sorted** and **compressed** data

Per layout implementations of operators

- Update
- Read
- Scan
- Join
- Aggregate

UNIVERSITY OF WATERLOO

Zone Maps

Order	Book	User	Qnt	$
100	HP	Alice	7	$15
101	Drac	Ted	90	$3
102	SQL	Geoff	1	$100
103	HP	Jean	3	$15
Min	100	Drac	Alice	1	$3
Max	103	SQL	Ted	90	$100

SELECT Book
WHERE
QNT > 100

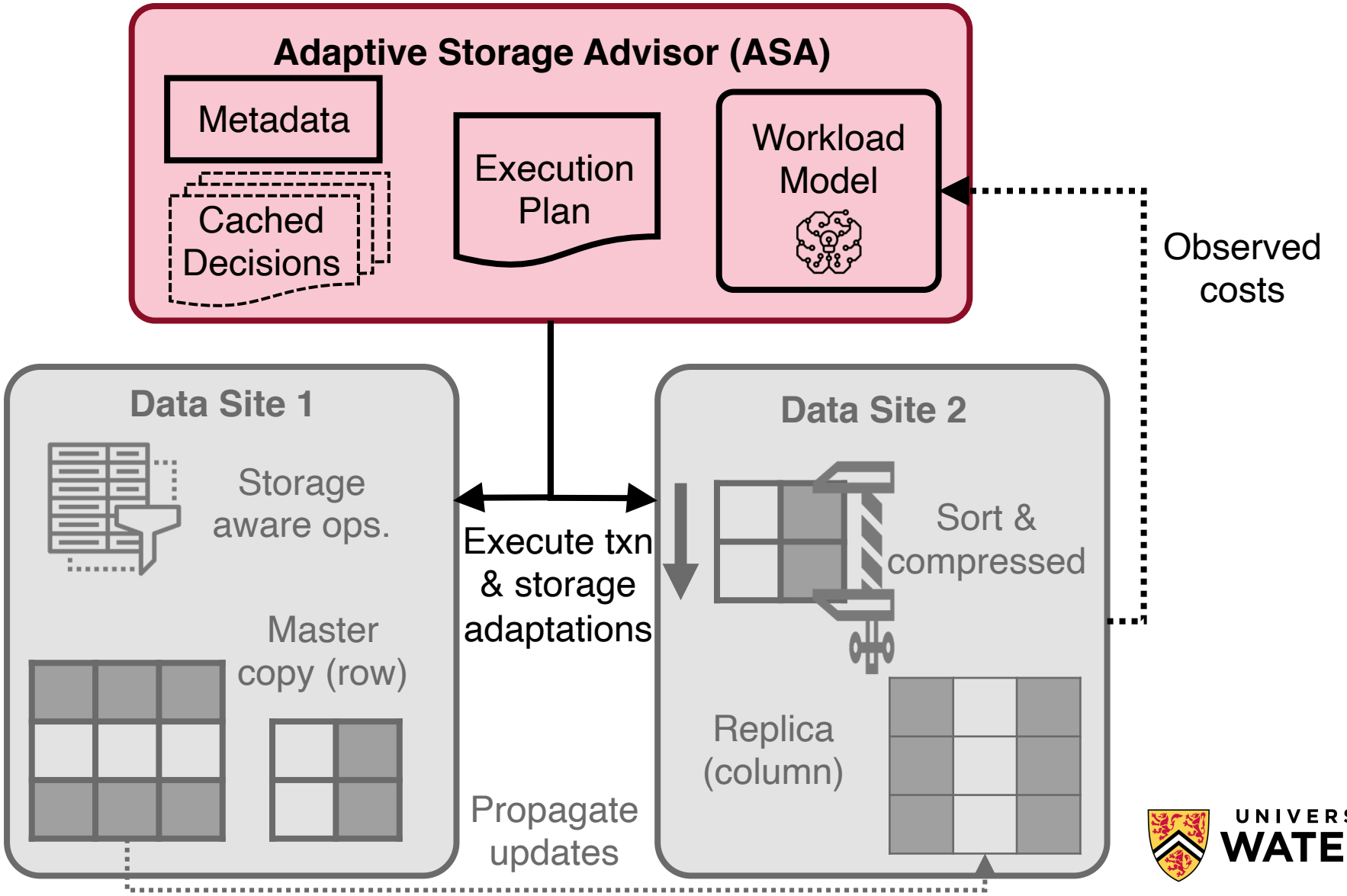
Zone Maps

Kept in memory and on a per partition basis

Allows data skipping, minimizing I/O

Min	100	Drac	Alice	1	$3
Max	103	SQL	Ted	90	$100

Storage Adaptation Decisions



Adaptive Storage Advisor

Should we **execute** a proposed change?

What **changes** should we propose?

Adaptive Storage Advisor

Should we **execute** a proposed change?

What **changes** should we propose?

UNIVERSITY OF WATERLOO

Making Layout Decisions

Quantify if a layout change is **beneficial**

Order	Book	User	Qnt	$
100	HP	Alice	7	$15
101	Drac	Ted	90	$3
102	SQL	Geoff	1	$100
103	HP	Jean	3	$15

SELECT book, **SUM**(qnt)
GROUP BY book

Should Proteus use **column** storage **sorted** by **book**?

UNIVERSITY OF WATERLOO

Quantifying Layout Decisions

Quantify if a layout change is **beneficial**

Txn **latency** with **current** vs **proposed** layouts

Current	**Proposed**
(row)	(Sorted col)
52 ms	15 ms

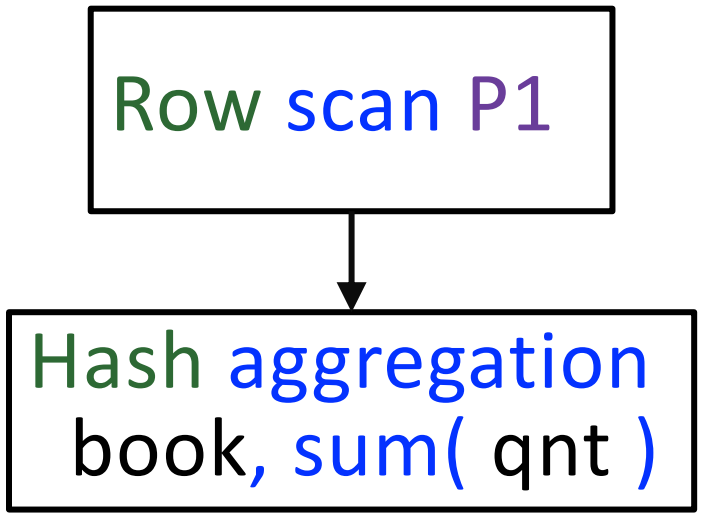
SELECT book, **SUM**(qnt)
GROUP BY book

Latency of a Transaction

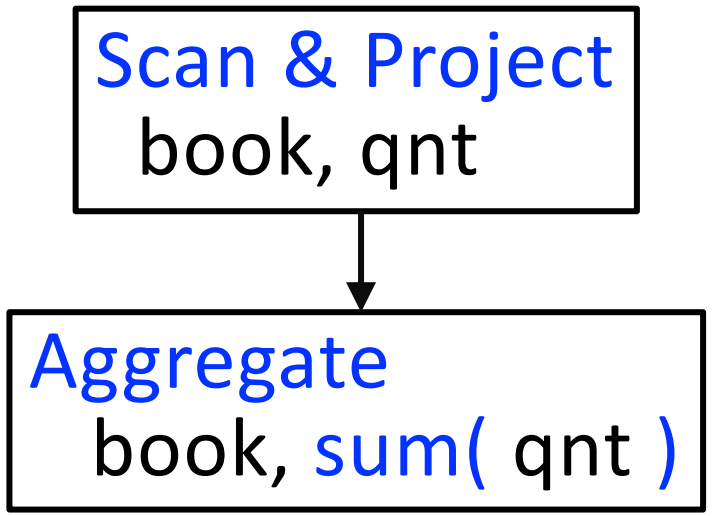
Breakdown transaction into **physical** operators

SELECT book, **SUM**(qnt) **GROUP BY** book

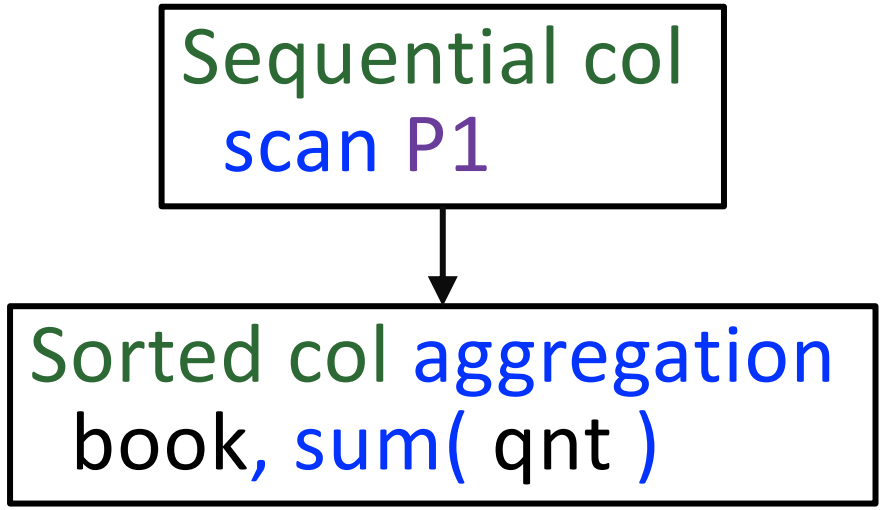
Row layout



Logical Plan



Sorted column layout



UNIVERSITY OF WATERLOO

Latency of a Transaction

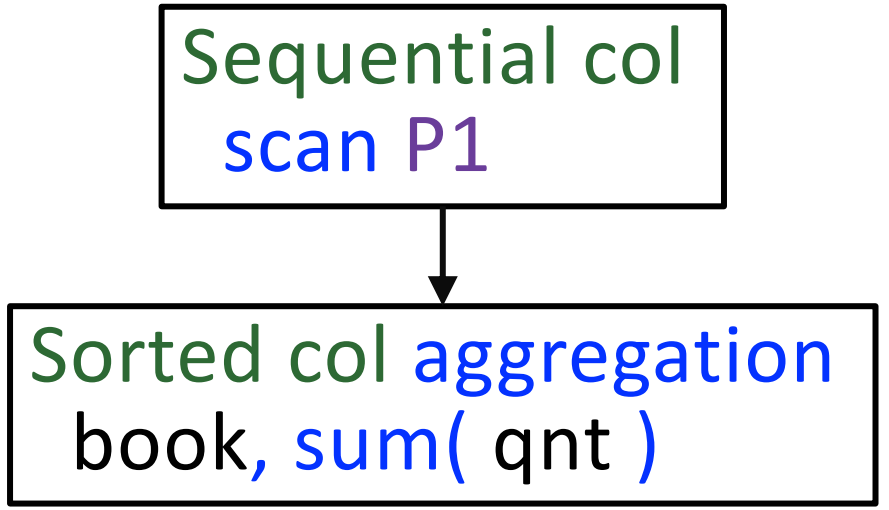
Breakdown transaction into **physical** operators

SELECT book, **SUM**(qnt) **GROUP BY** book

	Predictions
Sequential col scan P1	14 ms
Sorted col aggregation book, sum(qnt)	1 ms

Total predicted latency: 15 ms

Sorted column layout



UNIVERSITY OF WATERLOO

Predicting Latency of a Transaction

Breakdown transaction into **physical** operators

Predict physical operator latency

Per layout with workload stats as parameters

Cardinality
Data Width
Est Selectivity

Predicted Latency

Seq col scan

UNIVERSITY OF WATERLOO

Predicting Latency of a Transaction

Breakdown transaction into **physical** operators

Predict physical operator latency

Per layout with workload stats as parameters

Cardinality
Data Width
Est Selectivity

Row scan

Predicted Latency

UNIVERSITY OF WATERLOO

Quantifying Layout Decisions

Quantify if a layout change is **beneficial**

Txn **latency** with **current** vs **proposed** layouts

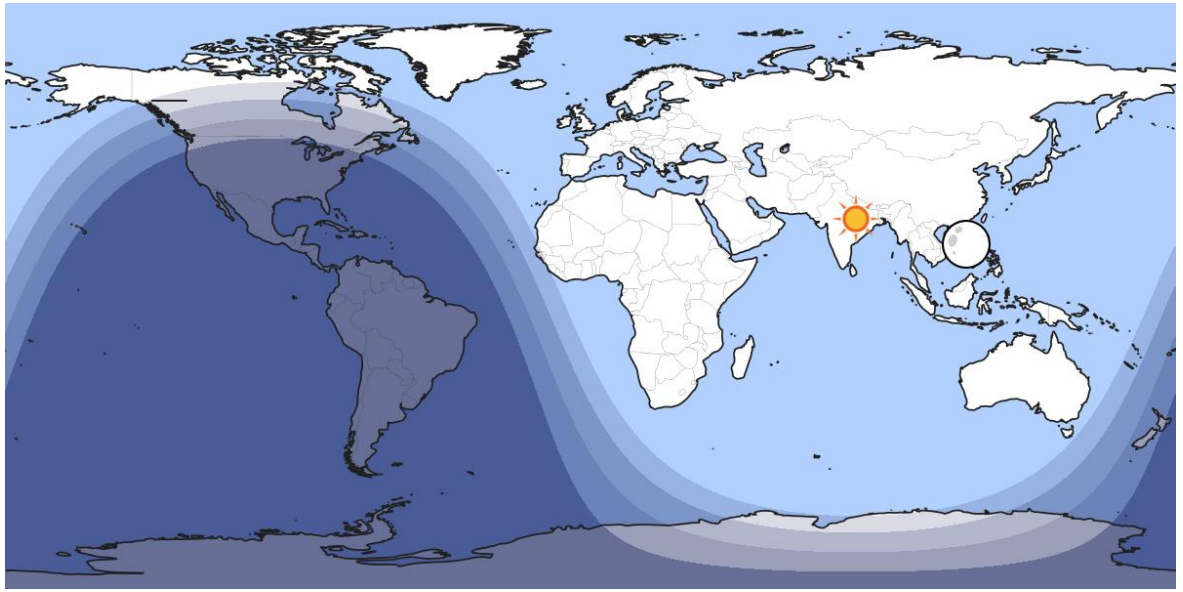
Current (row)	**Proposed** (Sorted col)	
52 ms	15 ms	**SELECT** book, **SUM**(qnt) **GROUP BY** book
14 ms	28 ms	**INSERT** (104, LoTR, John, 2, $21)

Weight by **likelihood** of transaction

UNIVERSITY OF WATERLOO

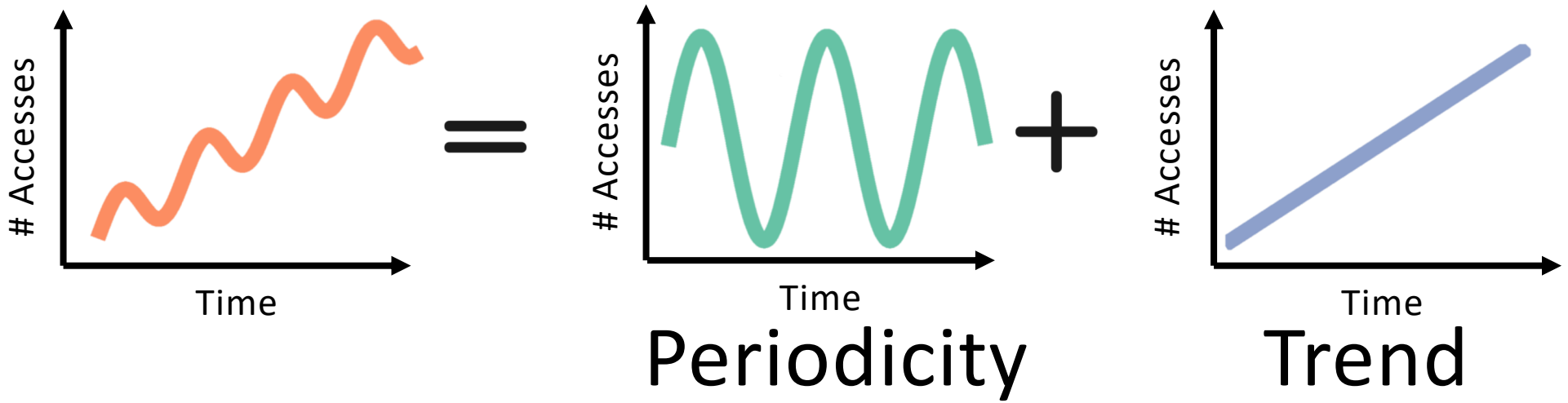
Likelihood of a Transaction

Data **accesses** to **storage**
often follow **predictable** pattern



Likelihood of a Transaction

Data **accesses** to **storage**
often follow **predictable** pattern



Quantifying Layout Decisions

Expected Benefit of Change

Txn **latency** with **current** vs **proposed** layouts

Weight by likelihood of txn

Cost incurred to perform change

Adaptive Storage Advisor

Should we **execute** a proposed change?

What **changes** should we propose?

Selecting What to Change

Based on **ongoing transaction**

Change high cost plan operators

Storage Layout	Decisions	
Format	row	column
Tier	memory	disk
Optimization	sort	compress
Replication	split or merge	horizontal or vertical
Mastership	Which site?	

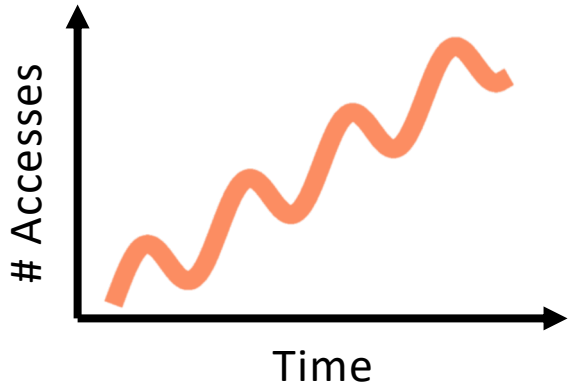
Selecting What to Change

Based on **ongoing transaction**

Change high cost plan operators

Predictively based on upcoming accesses

Simulate transactions



Based on **storage constraints**

tiny.cc/proteus

UNIVERSITY OF WATERLOO

How well does Proteus work?

CH BenCHmark (TPC-C & TPC-H)

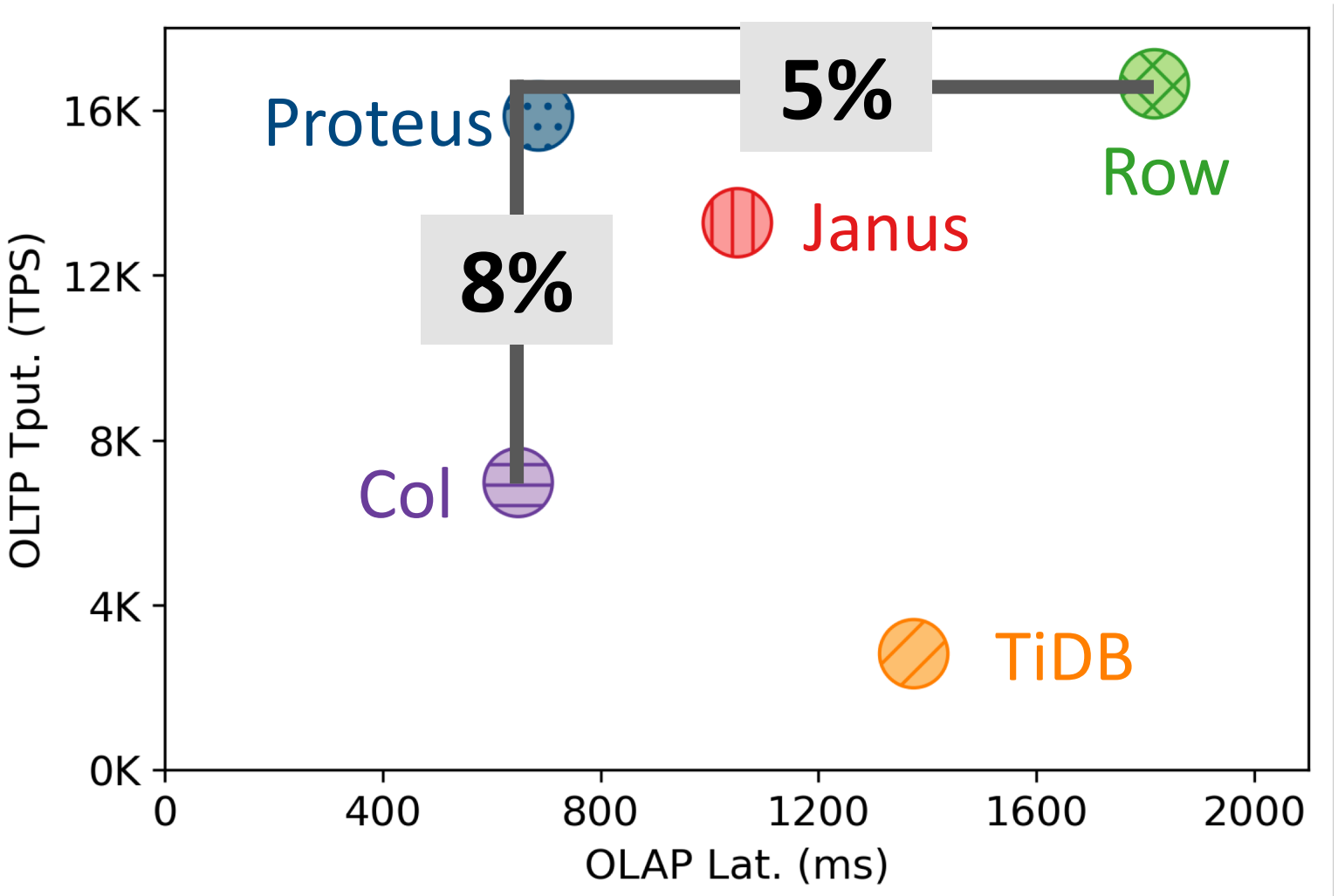
Comparisons

Row Store (RS)
Column Store (CS)
} *Single Copy*

Janus
TiDB
} *Maintain row & column*

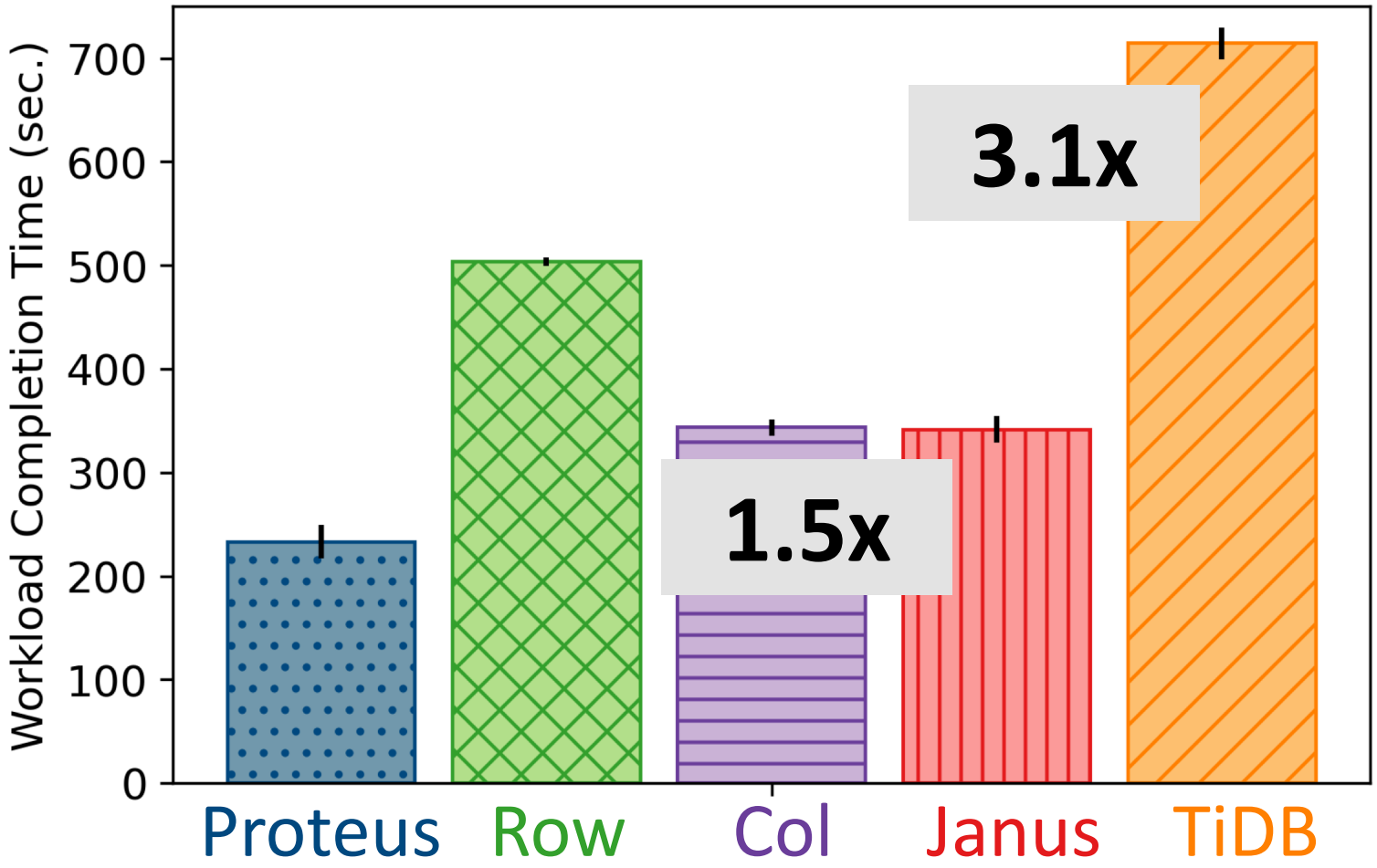
UNIVERSITY OF WATERLOO

CH BenCHmark



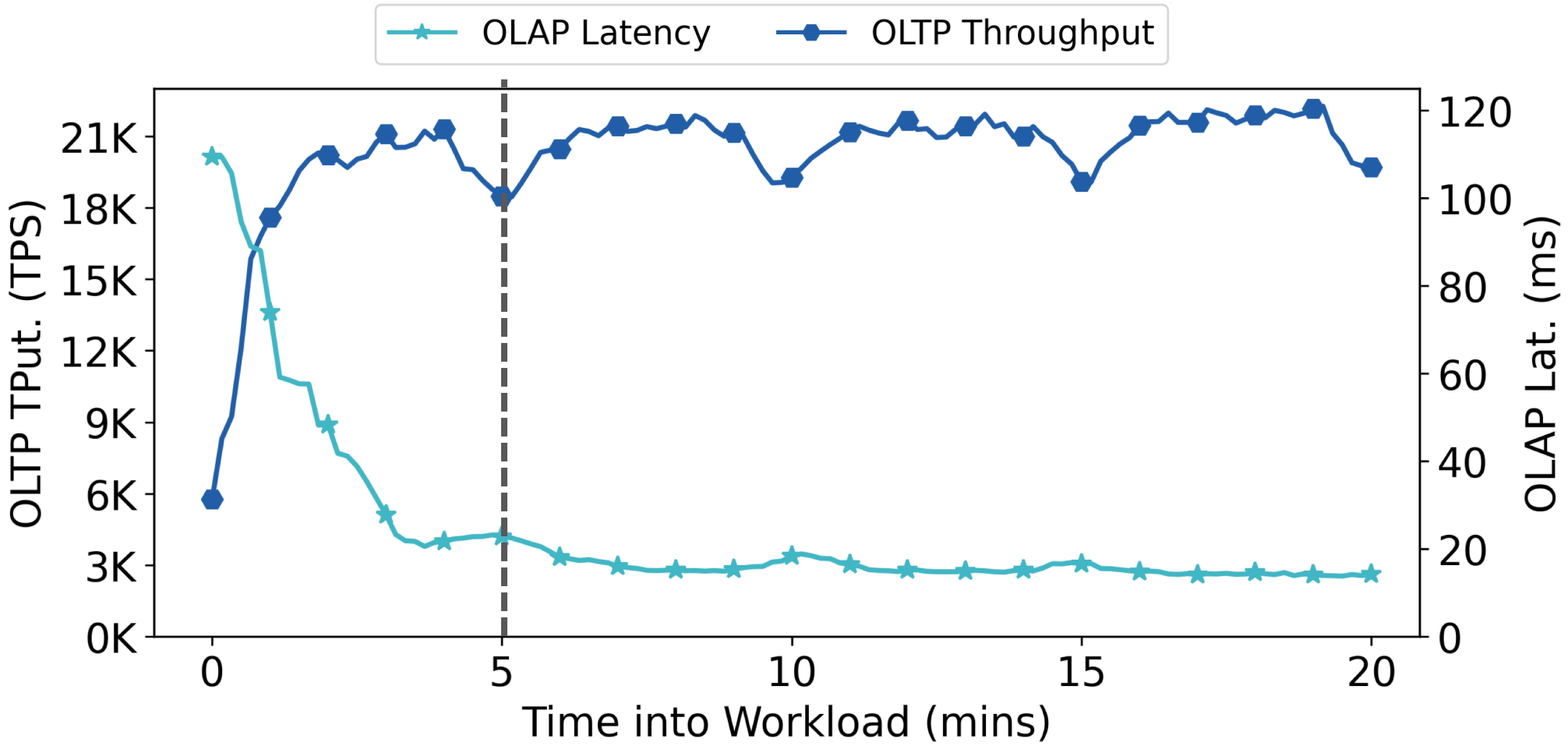
UNIVERSITY OF WATERLOO

CH BenCHmark



UNIVERSITY OF WATERLOO

Predictive Adaptation



UNIVERSITY OF WATERLOO

More Details

tiny.cc/proteus

More experimental workloads: Twitter, YCSB

Shifting workloads, Ablation studies

ML models (cost & access arrival)

Storage format and layout change execution

Efficient replication and concurrency control

UNIVERSITY OF WATERLOO

Proteus Takeaways tiny.cc/proteus

Adaptive Scale-Out Distributed DBMS

Selectively stores data in different formats

Proteus makes **decisions** based on **workload**

Superior performance on **mixed** workloads

UNIVERSITY OF WATERLOO