

Database Data Layouts

Row Store vs. Column Store

34c3

@wieso86

Data Layout – Example Table “Person”

	First Name	Last Name	Country
Row #1	Paul	Smith	Australia
Row #2	Lena	Jones	USA
Row #3	Marc	Winter	Germany
Row #4	Hanna	Schulze	Germany

Data Layout – Example Table “Person”

Row #1 ↓	Paul
	Smith
	Australia
Row #2 ↓	Lena
	Jones
	USA
Row #3 ↓	Marc
	Winter
	Germany
Row #4 ↓	Hanna
	Schulze
	Germany

**Row based
data layout**

First Name ↓	Paul
	Lena
	Marc
	Hanna
Last Name ↓	Smith
	Jones
	Winter
	Schulze
Country ↓	Australia
	USA
	Germany
	Germany

**Columnar
data layout**

Data Layout – Micro Benchmarks

- **Table characteristics**

- Integer data type: 4byte
- 2^0 to 2^7 columns

- **Benchmarks**

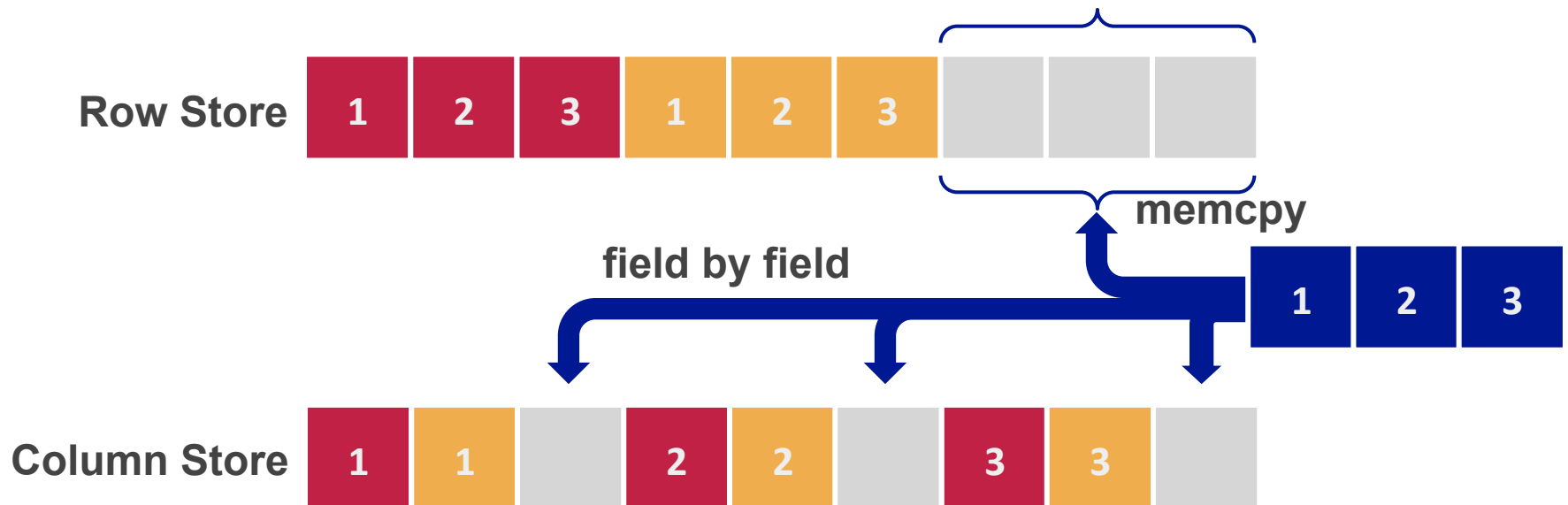
- INSERT INTO “Person” ...
- SELECT ... FROM “Person”
WHERE *First Name* = ...

- **VM (Benchmark Environment)**

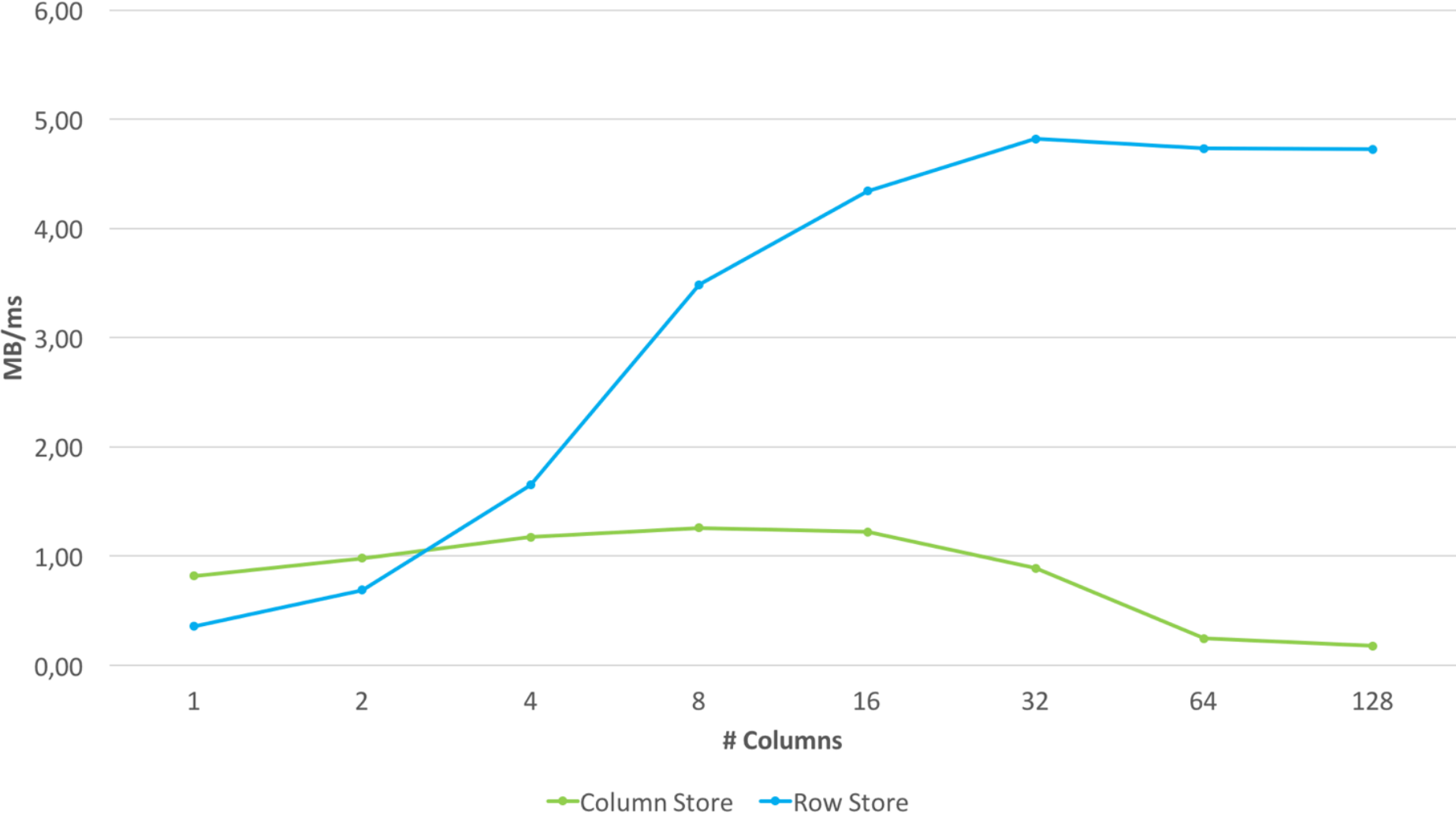
- Xeon E5-2697 v3 2,6-3,6 GHz
 - 4 Cores
- Cache
 - Cache line: 64 byte
 - L1 Cache: 32KB / core
 - L2 Cache: 256KB / core
 - L3 Cache: 35MB
- 32GB RAM

Data Layout – Micro Benchmark - Insert

- **Methods:** memcpy vs. field by field
- **Benchmark:** Insert 10M rows into empty database



Data Layout – Micro Benchmark - Insert



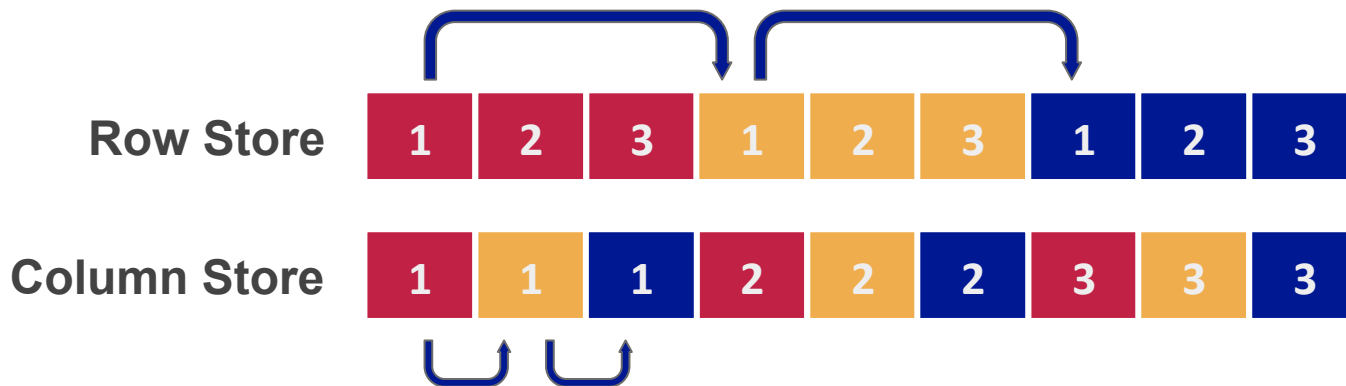
Data Layout – Micro Benchmark – Filter by *First Name*

- **Method**

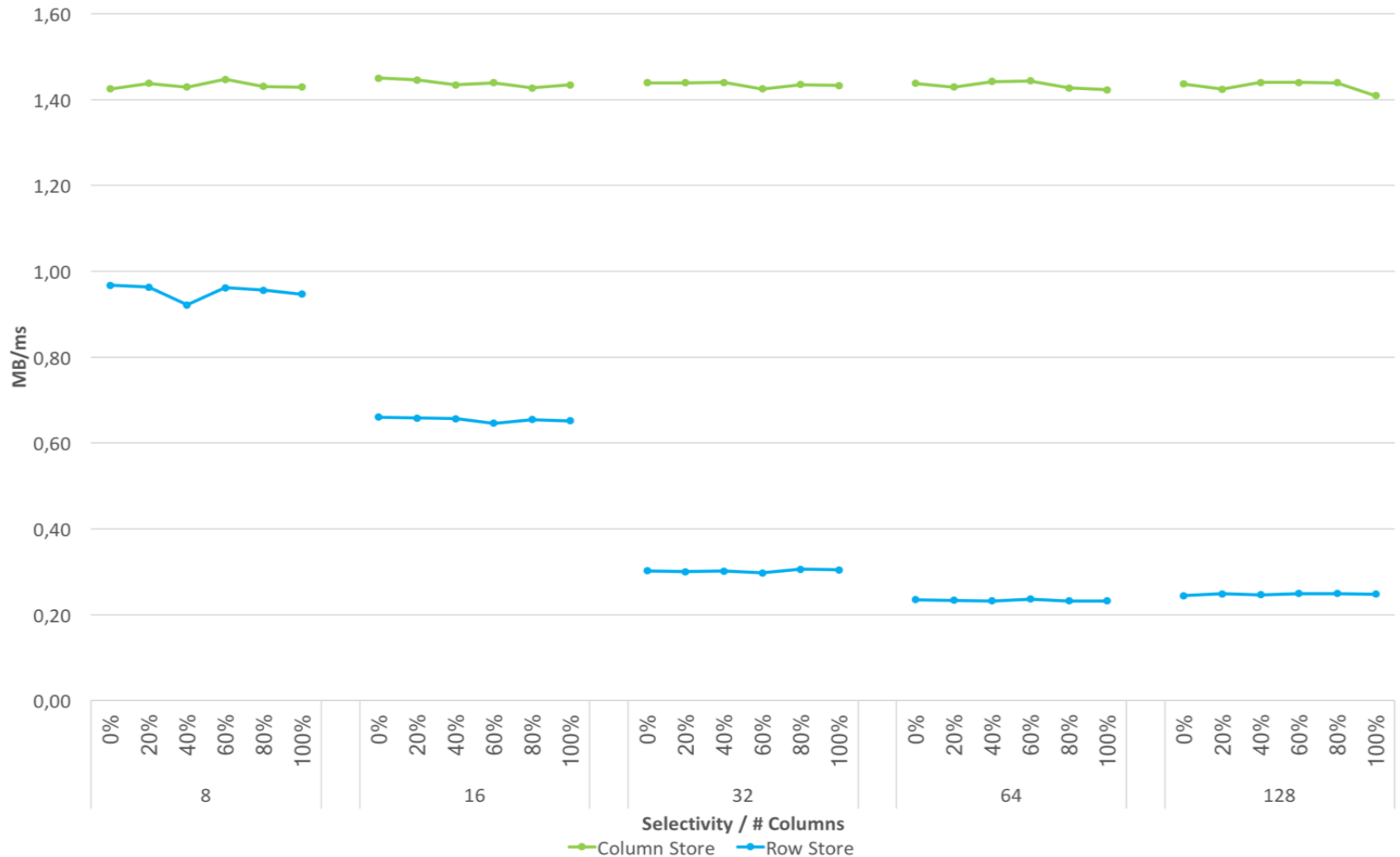
- Equality check for one specific attribute for all tuples
- Lookup positions (and build position list)

- **Benchmark**

- 10M rows
- {8,16,32,64,128} Columns
- {0,20,40,60,80,100}% Selectivity



Data Layout – Micro Benchmark – Filter by *First Name* (without writing results)



Data Layout – Micro Benchmark – Filter by *First Name* (with writing a position list)

