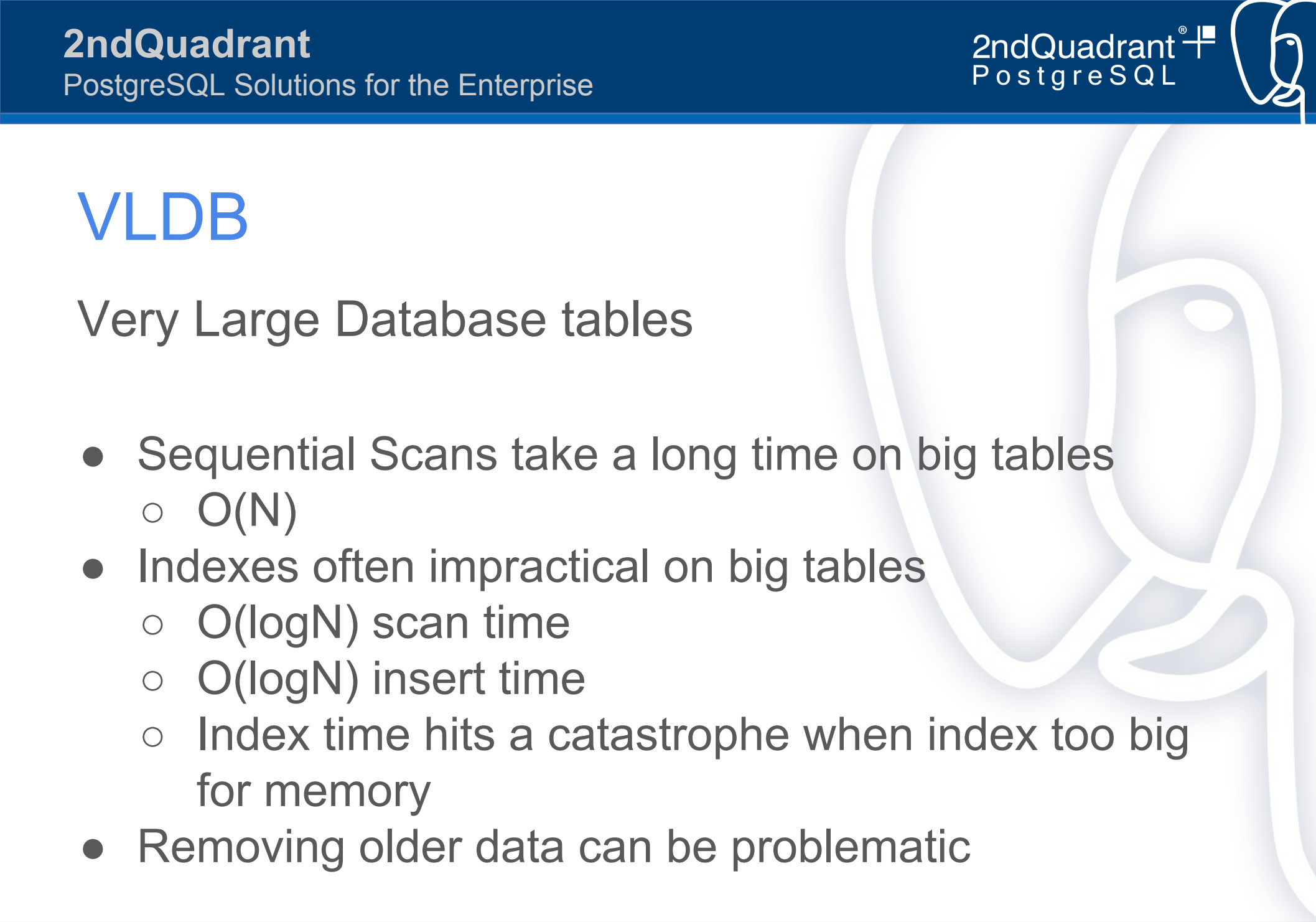
2ndQuadrant®
PostgreSQL

Partitioning & Time Series

15th Feb 2019

Simon Riggs, CTO, 2ndQuadrant
PostgreSQL Major Developer



VLDB

Very Large Database tables

- Sequential Scans take a long time on big tables
  - O(N)
- Indexes often impractical on big tables
  - O(logN) scan time
  - O(logN) insert time
  - Index time hits a catastrophe when index too big for memory
- Removing older data can be problematic

Opportunity!

Data organization

- By default, rows not organized within a table
- If there was some structure to our data we might be able to simply avoid scanning...
  … not a performance gain, just work avoidance

- Data frequently organizes itself
- Sometimes you want to force the organization

Solutions

Very Large Database tables

- Data frequently organizes itself
  e.g. OrderId increases over time
  e.g. LogTimestamp increases over time
  ⇒ BRIN indexes work very well for this case

- Sometimes you want to force the organization
  ⇒ Partitioning

Partitioning Options

- Inheritance-based partitioning
  First in PostgreSQL 8.1
  No longer recommended - but not deprecated, yet

- Declarative Partitioning (short summary!)
  First in PostgreSQL 10
  Improved in PostgreSQL 11
  Working better in PostgreSQL 12dev
  Working well in PostgreSQL 13 probably

Time-based Table Example

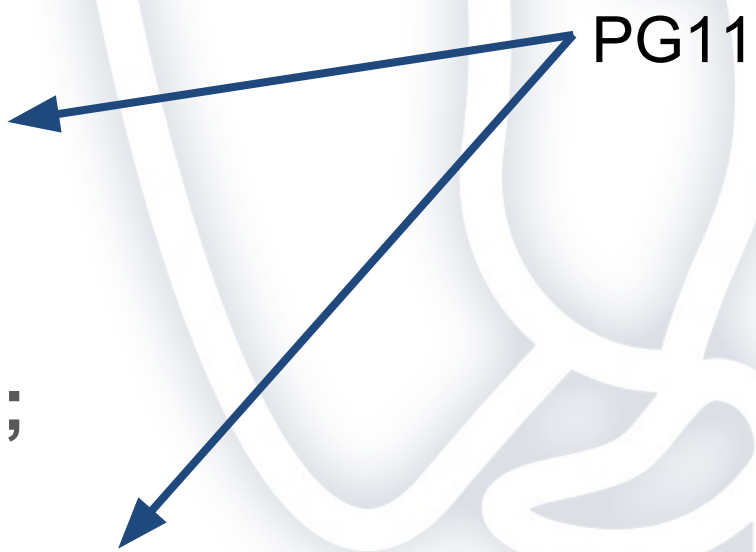
- CREATE TABLE measurement
  (deviceid BIGINT NOT NULL
  ,logts TIMESTAMP
  ,stuff JSONB);

- CREATE INDEX ON measurement
  USING BRIN (logts);

Time-Series Partitioning (Top-level)

- CREATE TABLE measurement
  (deviceid BIGINT NOT NULL
  ,logts TIMESTAMP
  ,stuff JSONB);
  PARTITION BY RANGE (logts);

- CREATE INDEX ON measurement
  USING BRIN (logts);



Time-Series Partitioning (Per Partition)

- CREATE TABLE **measurement_YYYY2019_MM01**
  PARTITION OF measurement
  FOR VALUES FROM (x) TO (y);
- Run something like this N times to create all partitions currently manual operation
- INTERVAL partitioning possible in PostgreSQL 13
  Automatically add a new partition every period

Partition Maintenance

- Add one new partition each time period
- Remove one new partition each time period

- Currently takes AccessExclusiveLock, so interrupts query and data loading
- Metadata only operation, so quick once we have lock
- Means we need to create partitions in advance

- Working on reducing lock levels for these actions

Default Partitions Warning

- Avoids the need to add new partitions, for a while
- When you do finally add a new partition, you get an AccessExclusiveLock

When a table has an existing DEFAULT partition and a new partition is added to it, the existing default partition must be scanned to verify that it does not contain any rows which properly belong in the new partition. If the default partition contains a large number of rows, this may be slow. The scan will be skipped if the default partition is a foreign table or if it has a constraint which proves that it cannot contain rows which should be placed in the new partition.

- Holds lock for long time… not much use for time-series

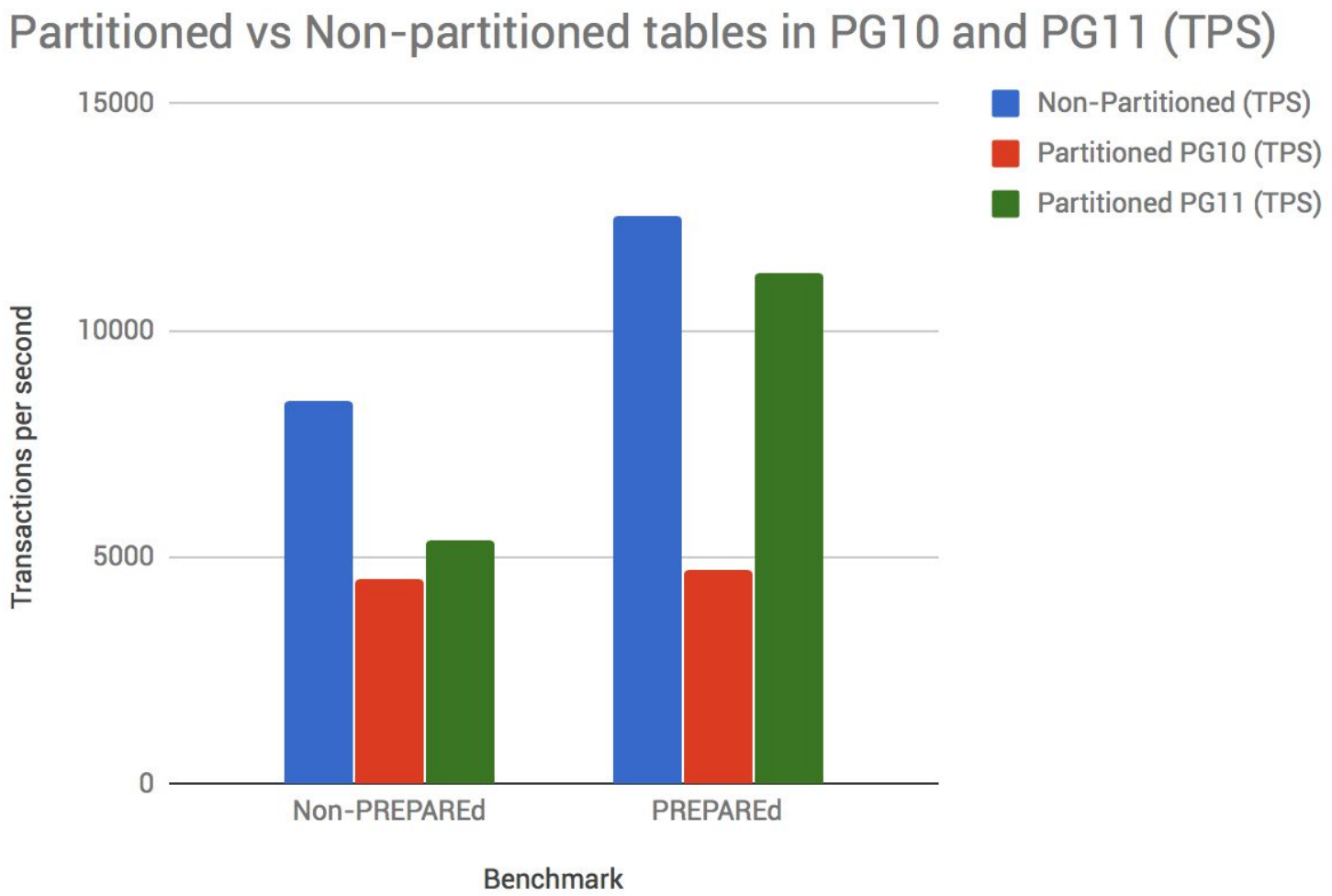
Partition Sizing

- How many partitions to have?

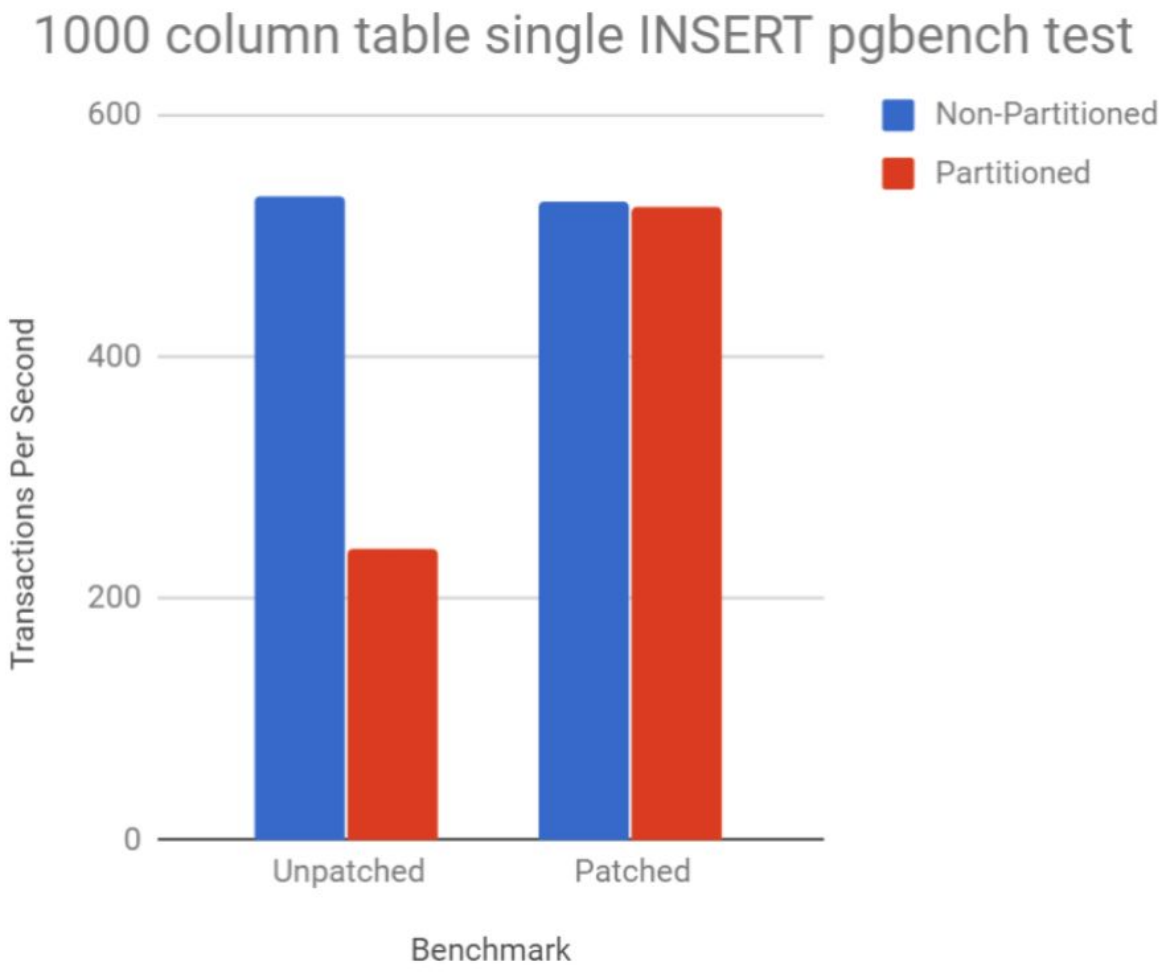
- 1 per minute
- 1 per hour
- 1 per day
- 1 per week
- 1 per month

- Depends on data volumes

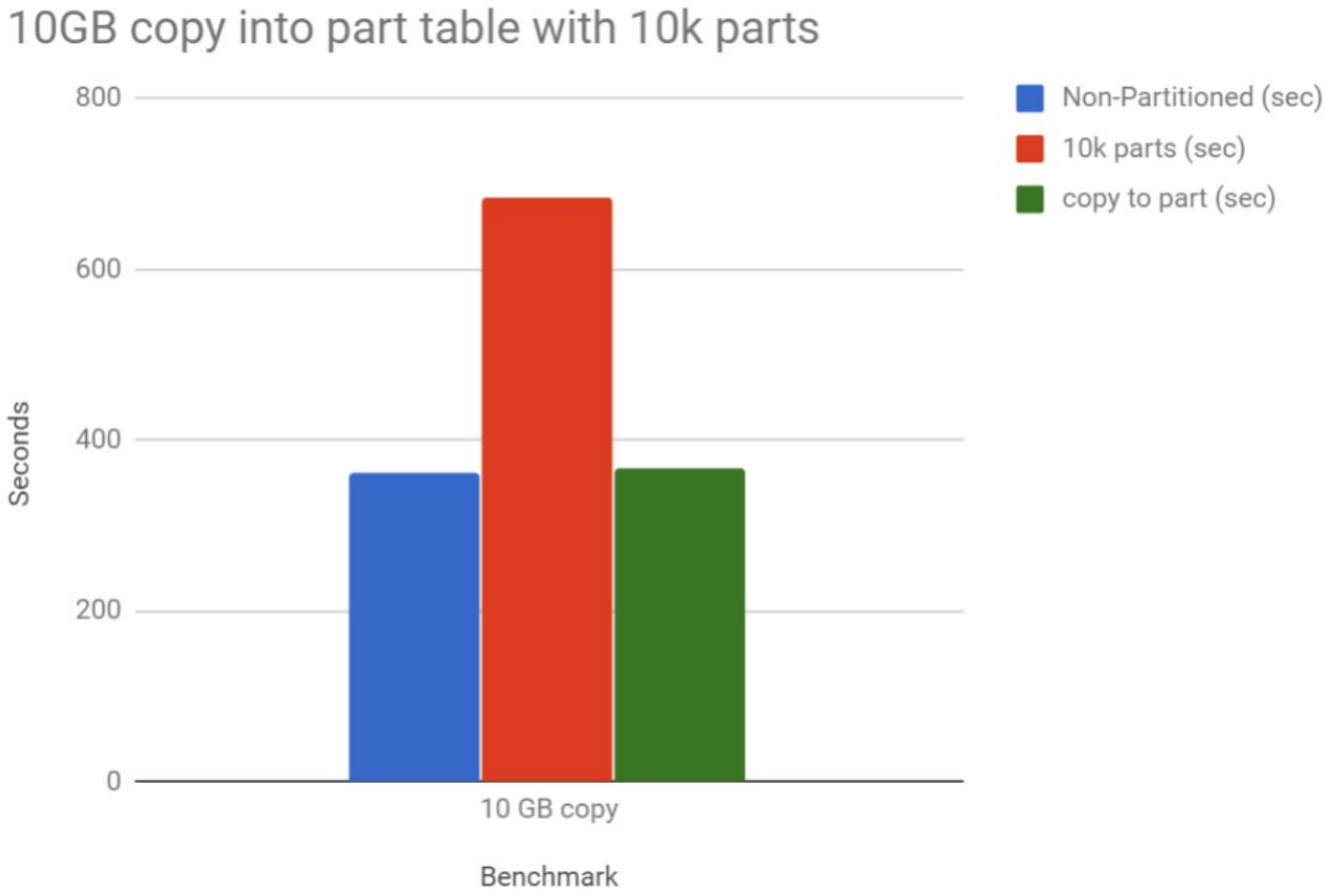
Perf: Prepared Query



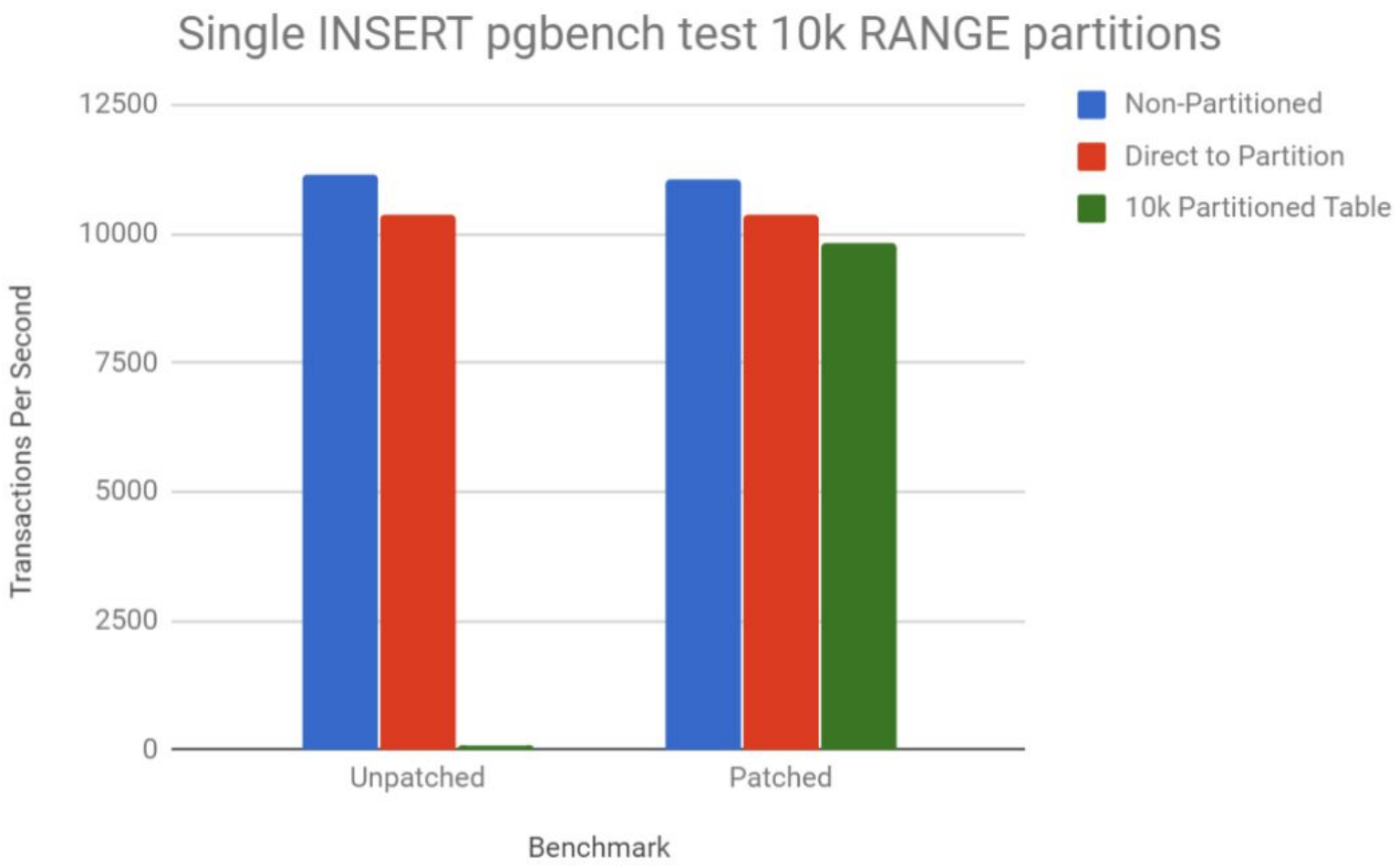
Perf: INSERTs



Perf: COPY with 10k partitions



Perf: INSERTs with 10k partitions



PostgreSQL 12 Situation

- Many patches under review
- Limitations still exist on number of partitions
- Relief will come in later versions

Choosing Solutions

BRIN or Partitioning?

- BRIN indexes simple; requires no DDL
  Fast and effective
- Partitioning requires complex DDL
  Current performance issues, explained here
  Allows you to easily drop data
- Suggest using both

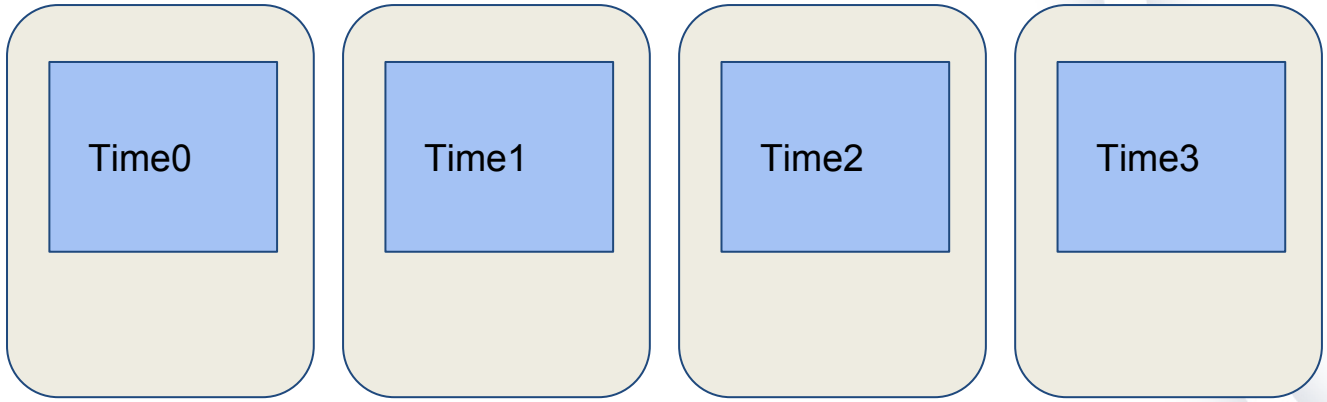
Scaling Time-Series

Spread partitions onto multiple nodes

- Various options are possible, with different characteristics
- Round Robin - partitions spread out across nodes
- Hash Partitions - each partition hashed across N nodes

Scaling - Round Robin

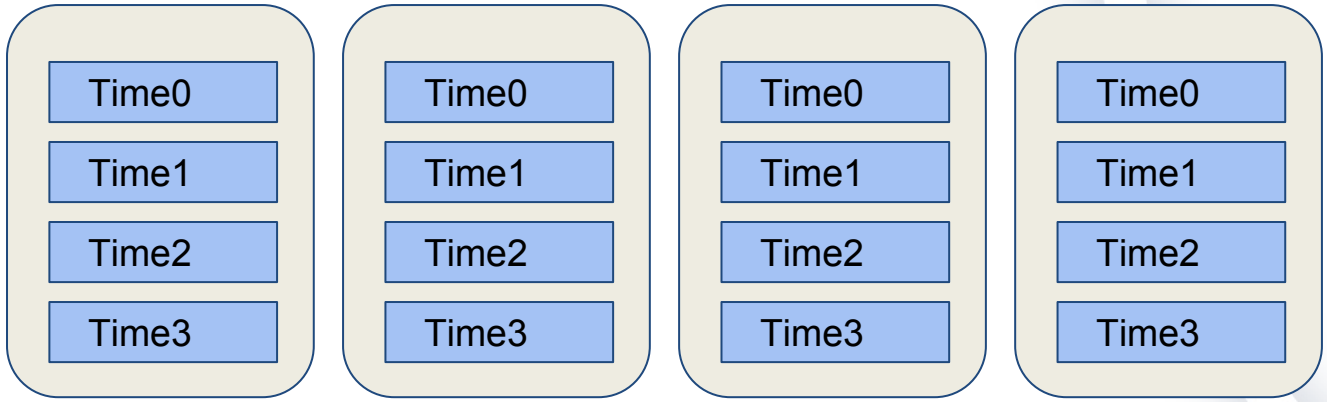
One partition goes to each node in turn



- No need for consistent query - works now
- Scales only for for multi-partition queries
- Fast access via 1 node for most-recent partition
- Easy upgrade path to add new hardware

Scaling - Hash Partitions

Each partition hashed across multiple nodes



- Needs consistent query
- Scales for big queries, even single node queries
- Higher latency for OLTP most-recent partition queries
- More complex options for upgrade, still possible

Multi-Level Partitioning

Each partition can itself have sub-partitions

- Top-Level PARTITION BY RANGE (logts)
- Each partition then further partitioned
  PARTITION BY HASH (deviceid)
- With sub-partitions defined like this

```
CREATE TABLE orders_p1 PARTITION OF orders
FOR VALUES WITH (MODULUS 4, REMAINDER 0);
CREATE TABLE orders_p2 PARTITION OF orders
FOR VALUES WITH (MODULUS 4, REMAINDER 1);
CREATE TABLE orders_p3 PARTITION OF orders
FOR VALUES WITH (MODULUS 4, REMAINDER 2);
CREATE TABLE orders_p4 PARTITION OF orders
FOR VALUES WITH (MODULUS 4, REMAINDER 3);
```

Multi-Level Partitioning

Each partition can itself have sub-partitions

- Top-Level PARTITION BY RANGE (logts)
- Each partition then further partitioned
  PARTITION BY HASH (deviceid)
- With sub-partitions defined like this

```
CREATE TABLE orders_p1 PARTITION OF orders
FOR VALUES WITH (MODULUS 4, REMAINDER 0);
CREATE TABLE orders_p2 PARTITION OF orders
FOR VALUES WITH (MODULUS 4, REMAINDER 1);
CREATE TABLE orders_p3 PARTITION OF orders
FOR VALUES WITH (MODULUS 4, REMAINDER 2);
CREATE TABLE orders_p4 PARTITION OF orders
FOR VALUES WITH (MODULUS 4, REMAINDER 3);
```

BDR Multi-node Query

Consistency and Performance

- Timestamp-based snapshots
- Allow consistent queries across nodes even with real-time replication of data
- Data verification between nodes

- Multi-node parallel query (MPP) across
  - Local clusters with remote DR nodes
  - Geo-distributed clusters

Solution Availability

- Tuning in PostgreSQL 12 and 13 contributed by 2ndQuadrant
- Available now in 2ndQuadrant Postgres (2QPG11)
- Multi-node query in Postgres-BDR
- Other solutions may be available from other vendors

2ndQuadrant PostgreSQL Solutions

- 24/7 Bug-fix Support and RemoteDBA
- Consulting and Training
- Product Stack focused on enhanced
  - Performance
  - High Availability
  - Security

  for your PostgreSQL apps

2ndQuadrant PostgreSQL Solutions

Website https://www.2ndquadrant.com/
Blog https://blog.2ndquadrant.com/
Email info@2ndQuadrant.com