What's Missing in Postgres?

BRUCE MOMJIAN



The presentation explains why some features are missing in Postgres. *Title concept from Melanie Plageman*

https://momjian.us/presentations



Creative Commons Attribution License

Last updated: September 2026

Outline

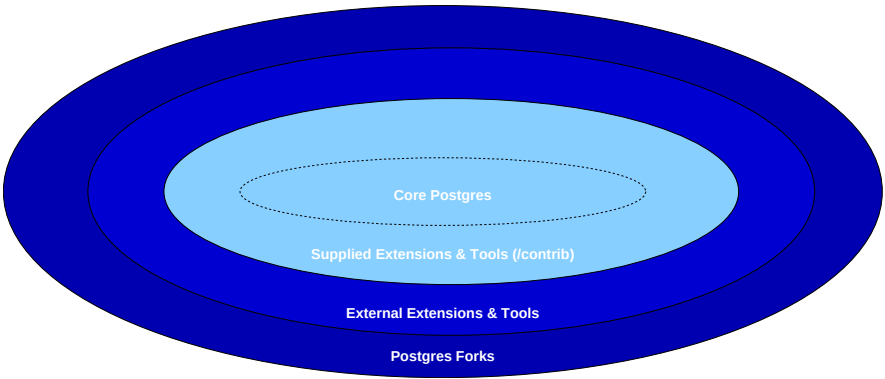
1. Postgres feature history
2. Feature sources
3. Cluster file encryption, i.e., TDE
4. Single host, performance
   - 4.1 64-bit transaction ids
   - 4.2 optimizer hints
   - 4.3 columnar storage
   - 4.4 global indexes
   - 4.5 direct I/O
   - 4.6 server-side threading
   - 4.7 internal connection pooler
5. Multi-host
   - 5.1 Oracle RAC-like
   - 5.2 multi-master replication
   - 5.3 logical replication of DDL
   - 5.4 sharding
6. Current status

1. Postgres Feature History Since 2010

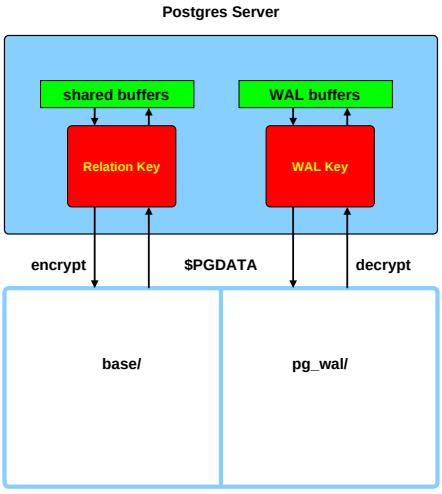
version	reldate	months	changes	C lines	C changes	% C change
9.0	2010-09-20		237	870790		
9.1	2011-09-12	12	203	932936	62146	7
9.2	2012-09-10	12	238	987460	54524	5
9.3	2013-09-09	12	177	1040813	53353	5
9.4	2014-12-18	15	211	1096707	55894	5
9.5	2016-01-07	13	193	1167110	70403	6
9.6	2016-09-29	9	214	1219720	52610	4
10	2017-10-05	12	189	1316447	96727	7
11	2018-10-18	12	170	1369590	53143	4
12	2019-10-03	11	180	1423215	53625	3
13	2020-09-24	12	178	1473738	50523	3
14	2021-09-30	12	220	1558178	84440	5
15	2022-10-13	12	184	1587763	29585	1
16	2023-09-14	11	206	1608031	20268	1
17	2024-09-26	12	182	1673116	65085	4
18	2025-09-25	12	210	1750814	77698	4
Averages		12	200			4.27

https://momjian.us/main/blogs/pgblog/2021.html#April_28_2021

2. Feature Sources



3. Cluster File Encryption, i.e., TDE



https://momjian.us/main/presentations/pending.html#cfe

Cluster File Encryption

Advantages

- Meets regulatory requirements, e.g., PCI
- Does not require coordination with infrastructure teams for storage encryption
- Automatically encrypts file system backups

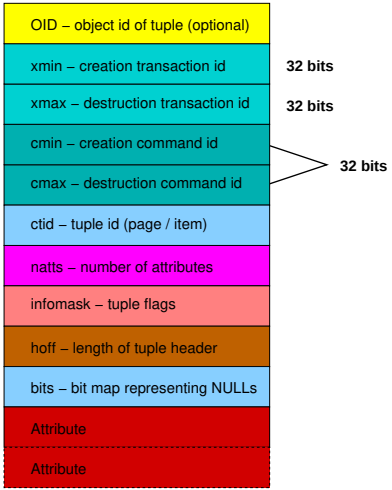
Disadvantages

- Of questionable security value, e.g., the key is in operating system memory
- Requires significant source code changes
- Client-side encryption is more secure

Percona is working on an external open source TDE extension that requires a fork of Postgres, and many Postgres forks support this.

https://momjian.us/main/blogs/pgblog/2025.html#February_22_2025
https://docs.percona.com/pg-tde/

4.1 64-Bit Transaction Ids



https://momjian.us/main/presentations/internals.html#mvcc

64-Bit Transaction Ids

Advantages

- Avoids the need to freeze tuples

Disadvantages

- Could increase the tuple header size by 50%
- Requires significant source code changes

The Postgres Pro fork of Postgres supports this.

4.2 Optimizer Hints

l	count	lookup_letter
p	342	Seq Scan on sample (cost=0.00..21.12 rows=342 width=2)
c	13	Bitmap Heap Scan on sample (cost=4.25..20.69 rows=13 width=2)
r	12	Bitmap Heap Scan on sample (cost=4.24..20.14 rows=12 width=2)
f	6	Bitmap Heap Scan on sample (cost=4.19..17.25 rows=6 width=2)
t	6	Bitmap Heap Scan on sample (cost=4.19..17.25 rows=6 width=2)
s	6	Bitmap Heap Scan on sample (cost=4.19..17.25 rows=6 width=2)
u	5	Bitmap Heap Scan on sample (cost=4.19..15.86 rows=5 width=2)
_	5	Bitmap Heap Scan on sample (cost=4.19..15.86 rows=5 width=2)
d	4	Bitmap Heap Scan on sample (cost=4.18..14.23 rows=4 width=2)
v	4	Bitmap Heap Scan on sample (cost=4.18..14.23 rows=4 width=2)
a	3	Bitmap Heap Scan on sample (cost=4.17..12.31 rows=3 width=2)
e	2	Bitmap Heap Scan on sample (cost=4.16..10.07 rows=2 width=2)
k	1	Index Only Scan using i_sample on sample (cost=0.15..8.17 rows=1 width=2)
i	1	Index Only Scan using i_sample on sample (cost=0.15..8.17 rows=1 width=2)

https://momjian.us/main/presentations/performance.html#optimizer

Optimizer Hints

Advantages

- Useful for quick fixes of optimizer mistakes

Disadvantages

- Locks query plans, preventing data distribution changes and optimizer improvements from using better plans
- While this can fix specific queries, the cause is often imperfect optimizer statistics or server settings
  - a more general fix can improve an entire class of queries, e.g., extended statistics, random_page_cost
- Often prevents the problem from being diagnosed and reported to the database project

pg_hint_plan is available as an external open source extension. Postgres 19's pg_plan_advice enables optimizer control. The EDB fork of Postgres supports this.

https://momjian.us/main/blogs/pgblog/2018.html#December_12_2018
https://github.com/ossc-db/pg_hint_plan

4.3 Columnar Storage

Column 1

Value 1	Row 2, 7, 9, 12
Value 2	Row 1, 5, 11, 14
Value 3	Row 4, 6, 8, 15
Value 4	Row 3, 10, 13, 16

Column 2

Value 1	Row 4, 6, 11, 16
Value 2	Row 3, 10, 12, 14
Value 3	Row 1, 5, 7, 9
Value 4	Row 2, 8, 13, 15

Column 3

Value 1	Row 4, 7, 11, 14
Value 2	Row 2, 5, 6, 13
Value 3	Row 3, 8, 10, 12
Value 4	Row 1, 9, 15, 16

Columnar Storage

Advantages

- Column values are only stored once per table, reducing storage requirements
- Ideal for columns with many duplicates

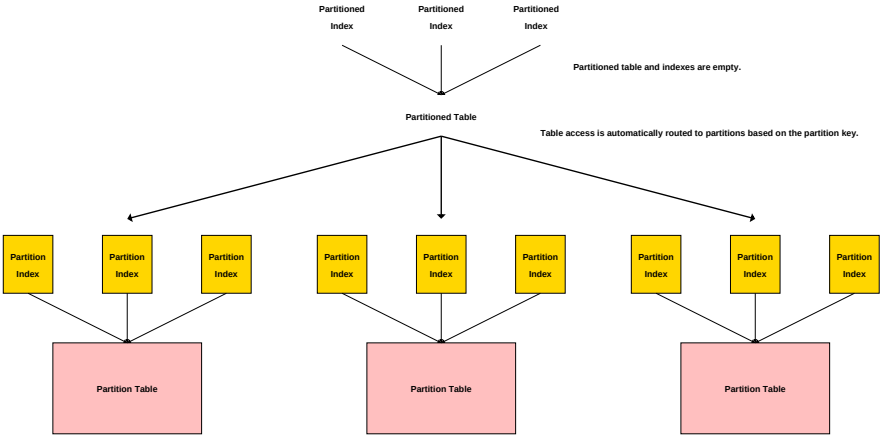
Disadvantages

- Accessing all columns of a row is expensive
- Updates and deletes are expensive
- Requires optimizer and storage changes

Citus is available as an external open source extension.

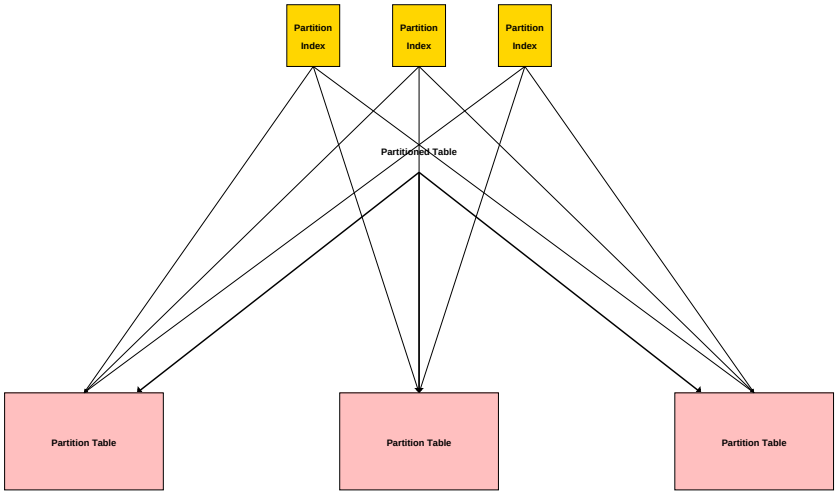
https://wiki.postgresql.org/wiki/ColumnOrientedSTorage
https://www.citusdata.com/product/community

4.4 Global Indexes: Per-Partition Indexes



https://momjian.us/main/presentations/performance.html#partitioning

Global Indexes



https://momjian.us/main/presentations/performance.html#partitioning

Global Indexes

Advantages

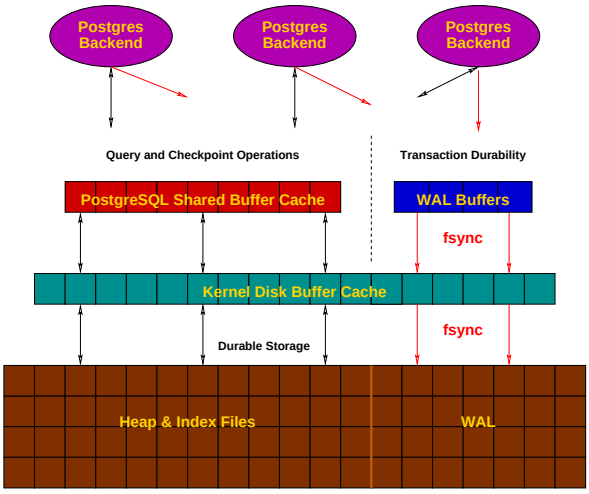
- Allows indexing of values that are not part of the partition key
- Allows unique constraints that are not part of the partition key

Disadvantages

- Partitioning is used to split very large tables, so global indexes would be very large
- Dropping partitions is expensive
- Requires significant source code changes

https://momjian.us/main/blogs/pgblog/2020.html#July_1_2020

4.5 Direct I/O



https://momjian.us/main/presentations/administration.html#wal

Direct I/O

Advantages

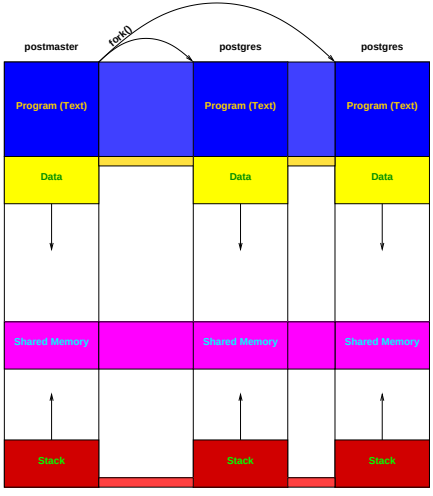
- Prevents double-buffering by the kernel and Postgres shared buffer cache
- Prevents copying of data from kernel buffers to shared buffers

Disadvantages

- Postgres-scheduled reads and writes might conflict with non-Postgres I/O
- Prevents sharing of kernel memory for I/O caching and per-process memory usage

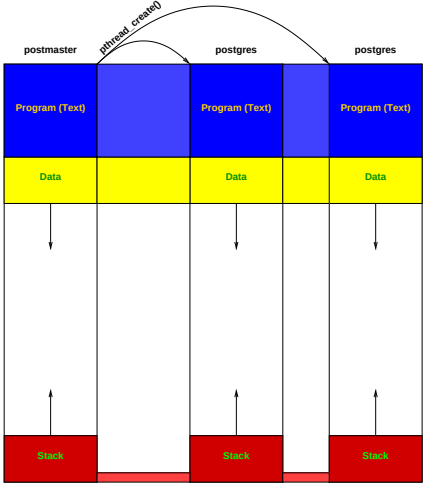
https://momjian.us/main/blogs/pgblog/2017.html#June_5_2017
https://momjian.us/main/blogs/pgblog/2018.html#December_7_2018

4.6 Server-Side Threading: Fork()



https://momjian.us/main/presentations/internals.html#shared_memory

Server-Side Threading



Server-Side Threading

Advantages

- Reduces task switching time

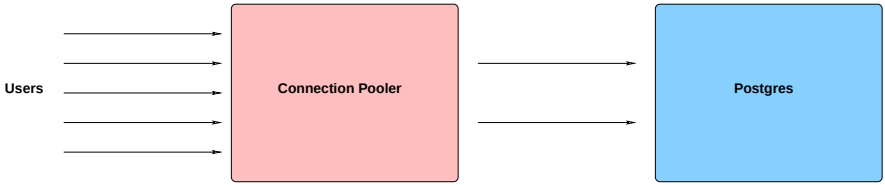
Disadvantages

- Makes Postgres sessions less resilient to session failure
- Requires significant source code changes

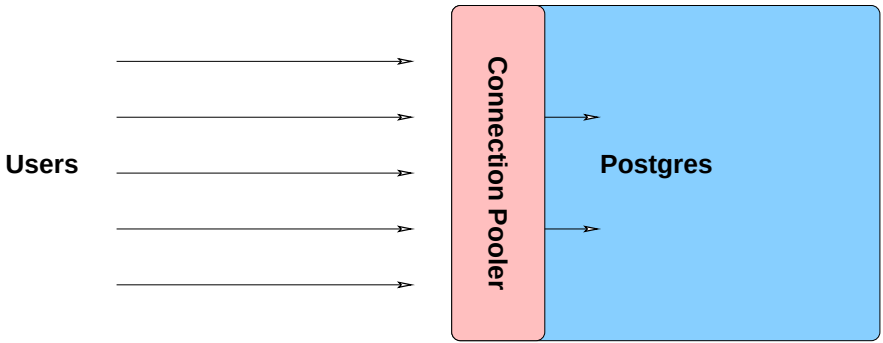
Another option is to keep one process per backend but allow each backend process to use multiple threads.

https://momjian.us/main/blogs/pgblog/2018.html#December_26_2018

4.7 Internal Connection Pooler: External Pooler



Internal Connection Pooler



Internal Connection Pooler

Advantages

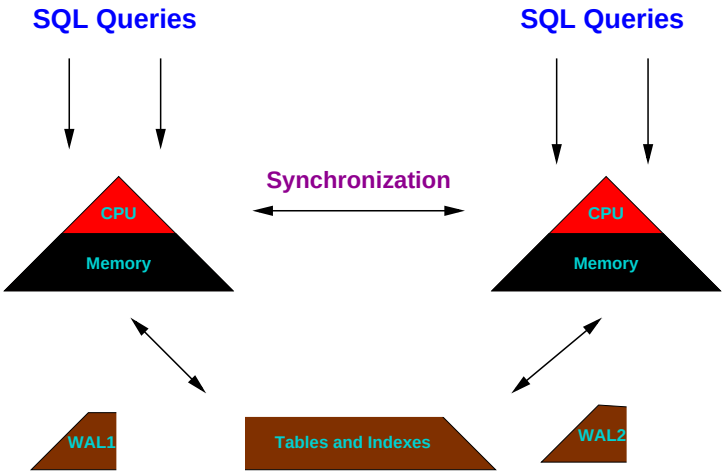
- Reduces latency
- More flexible authentication
- Simpler configuration

Disadvantages

- Insufficient for failover control

https://momjian.us/main/blogs/pgblog/2017.html#April_21_2017
https://momjian.us/main/blogs/pgblog/2019.html#January_25_2019

5.1 Oracle RAC-Like



https://momjian.us/main/presentations/performance.html#scaling

Oracle RAC-Like

Advantages

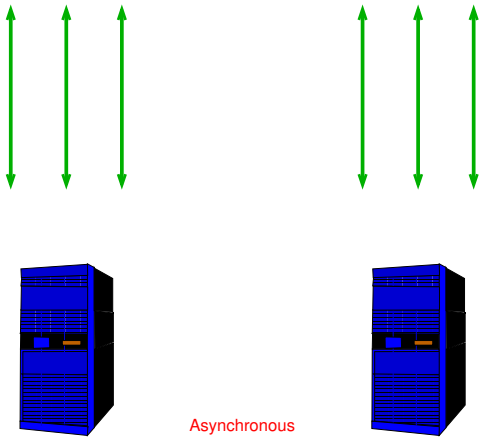
- Scales CPU and memory
- Partial reliability, partial scaling

Disadvantages

- Does not scale I/O
- Communication overhead between hosts requires application workload partitioning
- Complex architecture

https://momjian.us/main/blogs/pgblog/2018.html#July_9_2018

5.2 Multi-Master Replication



https://momjian.us/main/presentations/arch.html#replication

Multi-Master Replication

Advantages

- Allows simple draining of server traffic for maintenance
- Allows read-only scaling without traffic management

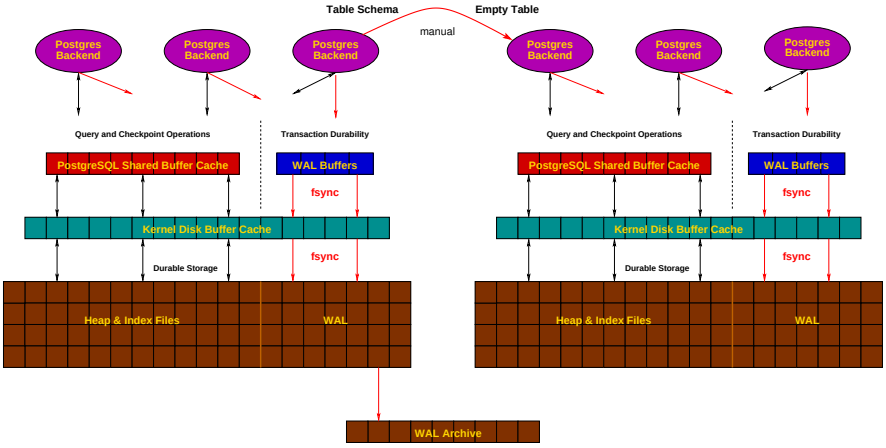
Disadvantages

- Requires conflict resolution management
- Requires DDL management when using Postgres logical replication; see next section

Some Postgres forks support this, e.g., EDB, pgEdge, Postgres Pro.

https://momjian.us/main/blogs/pgblog/2018.html#July_9_2018

5.3 Logical Replication of DDL



https://momjian.us/main/presentations/administration.html#wal

Logical Replication of DDL

Advantages

- Simplifies administration

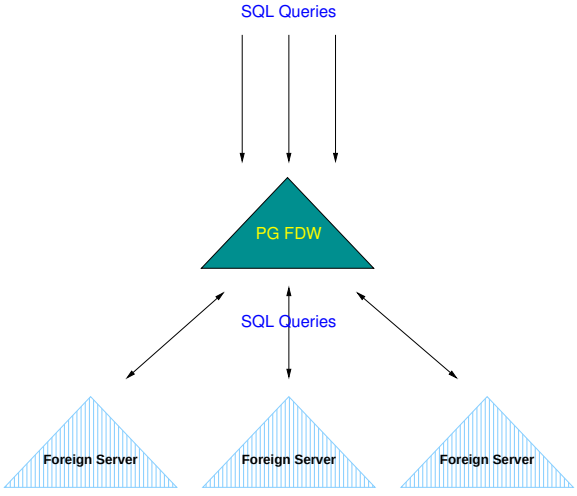
Disadvantages

- Requires regular source code updates to replicate new DDL

Some Postgres forks support this, e.g., EDB, pgEdge, Postgres Pro.

https://wiki.postgresql.org/wiki/Logical_replication_of_DDLs

5.4 Sharding



https://momjian.us/main/presentations/pending.html#sharding

Sharding

Advantages

- Allows writes to be scaled across multiple servers
- Allows data volumes to exceed a single server

Disadvantages

- Complex setup and administration
- Additional latency
- Limited value for queries that are counter to the sharding key

https://momjian.us/main/blogs/pgblog/2023.html#November_1_2023
https://wiki.postgresql.org/wiki/Built-in_Sharding

6. Current Status

- Cluster file encryption, i.e., TDE, fork
- Single host, performance
  - 64-bit transaction ids, fork
  - optimizer hints, external & fork
  - columnar storage, external
  - global indexes
  - direct I/O
  - server-side threading
  - internal connection pooler
- Multi-host
  - Oracle RAC-like
  - multi-master replication, fork
  - logical replication of DDL, fork
  - sharding

Green is in-progress for core Postgres and supplied extensions and tools; red is no progress. External extensions and Postgres forks are indicated.

The Pattern of Missing Features

While proprietary databases target the workloads of their largest paying customers, Postgres targets the workloads of general users.



https://www.flickr.com/groups/immemory

Conclusion



https://momjian.us/presentations

https://www.flickr.com/photos/bryanh/