PERCONA

Databases run better with Percona

About Me

JOBIN AUGUSTINE (47)

- Working for Percona
- Life with databases

PERCONA

PostgreSQL Indexing

Common Pitfalls and Misconceptions Unveiled

JOBIN AUGUSTINE

Why This topic is important - human factor

- Microscopic view
- Looses larger picture
- Unaware about the consequences

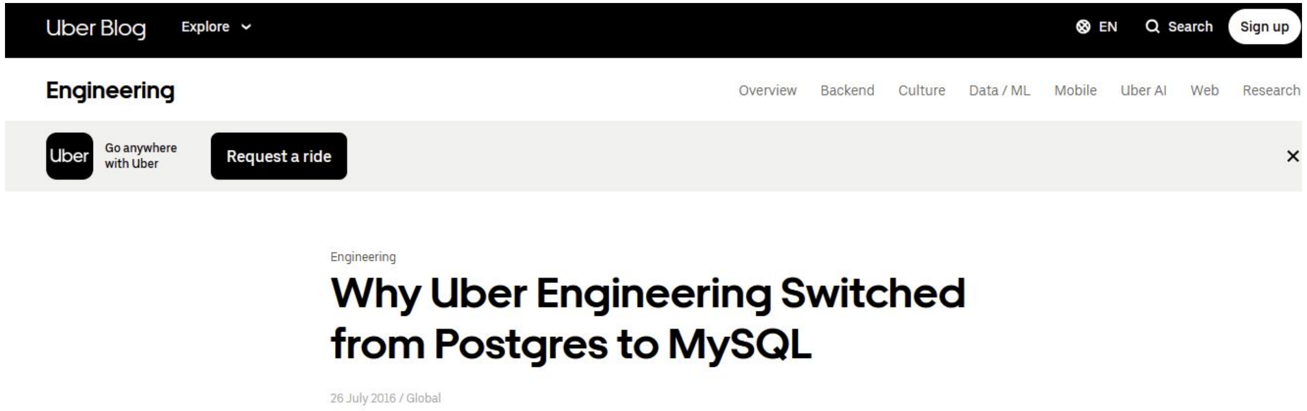
- Typically, the focus is on the positive aspects, with little attention given to potential negative impacts.
- Indexes are a common area where this oversight occurs, often leading to significant damage.
- Maintaining a 360-degree awareness is crucial.

1. Indexes penalize the transactions

- Indexes could speed up data retrieval (SELECT)
- But, at the cost of transaction processing (DMLs)
  - It can jeopardize the TPS objectives
- Each transaction need to update indexes

- Create a mental picture of what could be happening
- Avoid falling prey to testimonials.

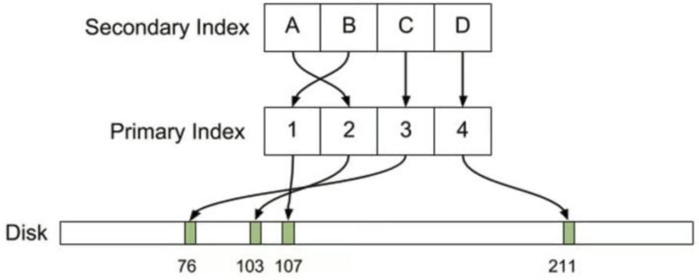
Write Amplification



Engineering

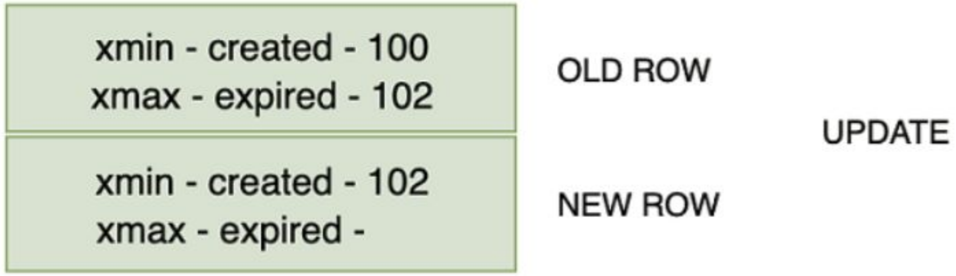
Why Uber Engineering Switched from Postgres to MySQL

26 July 2016 / Global



“a small logical update (say, writing a few bytes) becomes a much larger”

2. Write amplification in secondary index



xmin - created - 100
xmax - expired - 102

OLD ROW

UPDATE

xmin - created - 102
xmax - expired -

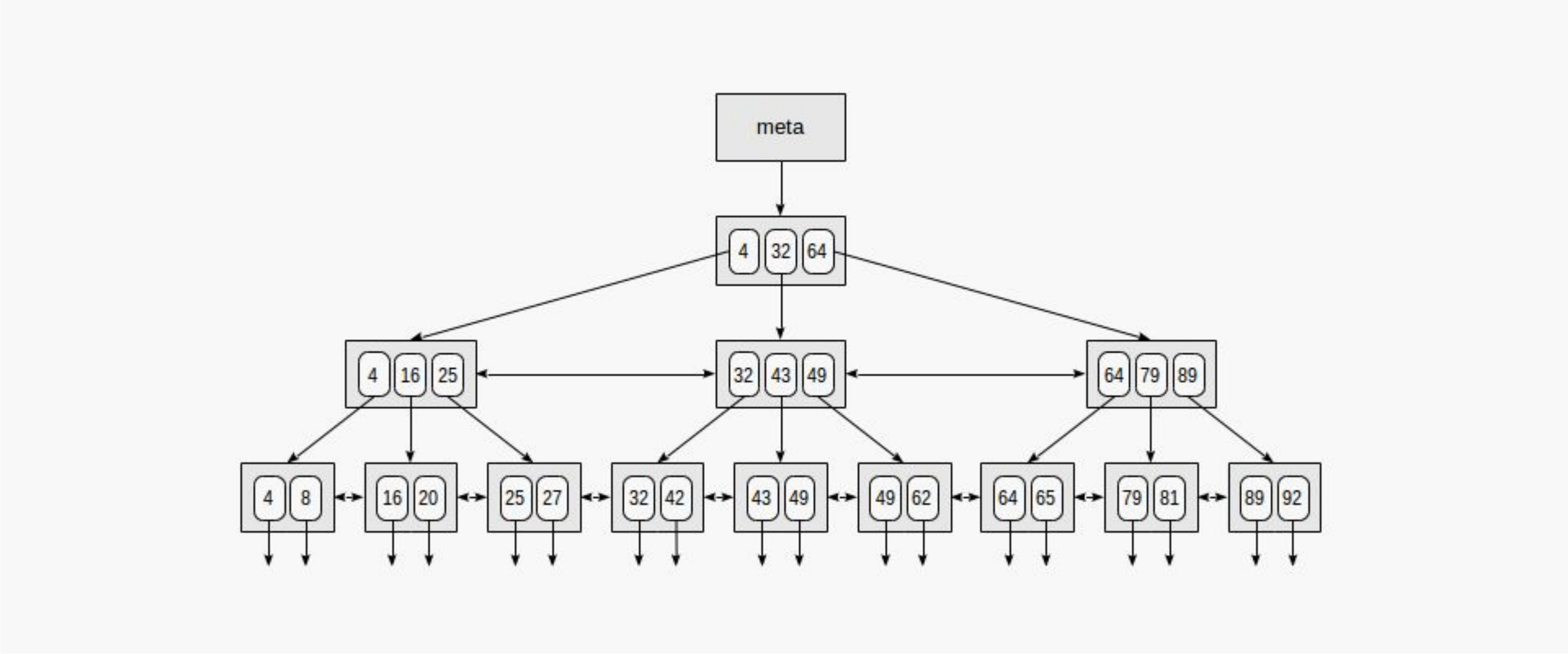
NEW ROW

- UPDATEs causes new tuple in Table
- All indexes need to be updated pointing to new tuple

Logically, PostgreSQL need to update every index, irrespective whether the updated column is part of index.

3. Write Amplification due scattered storage

Btree

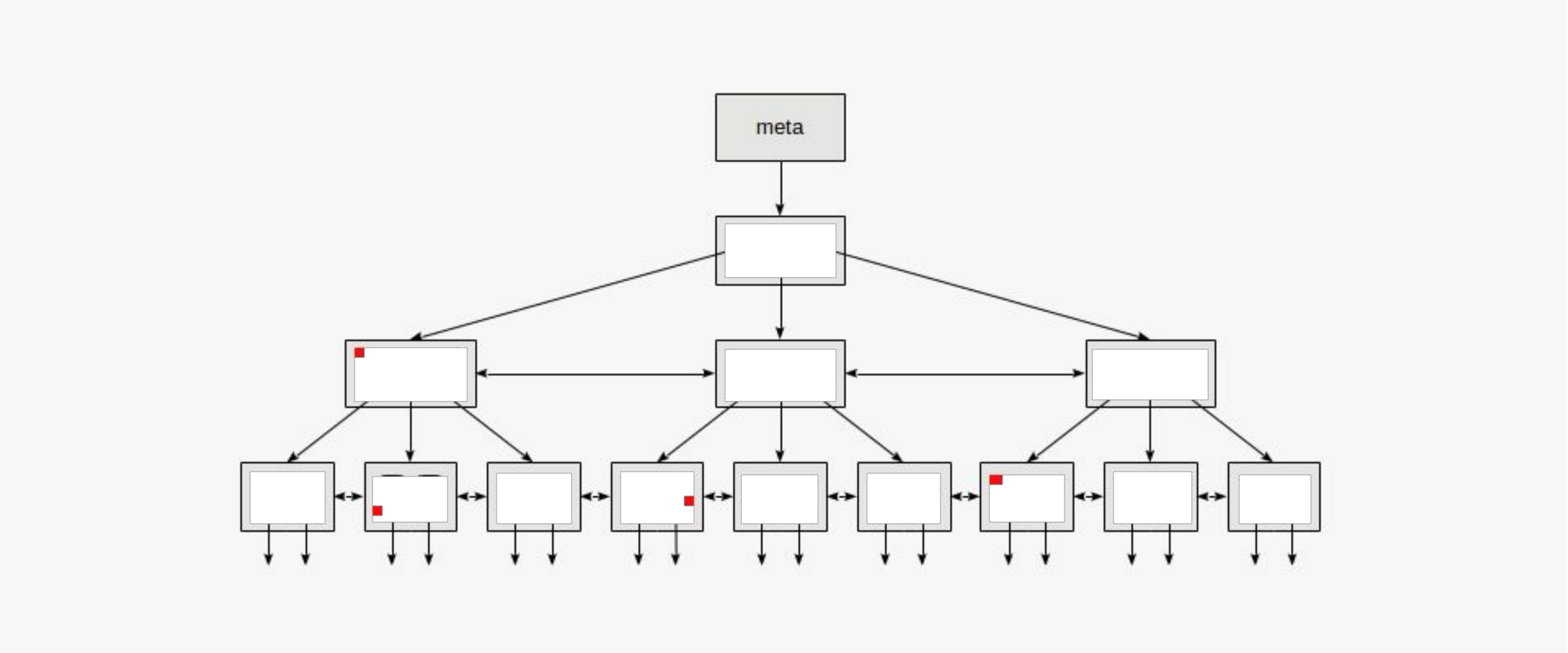


Even continuous data goes to different locations

Index access is Random Write

Egor Rogov, Indexes in PostgreSQL — 4 (Btree) https://postgrespro.com/blog/pgsql/4161516

4. Write amplification due to page size



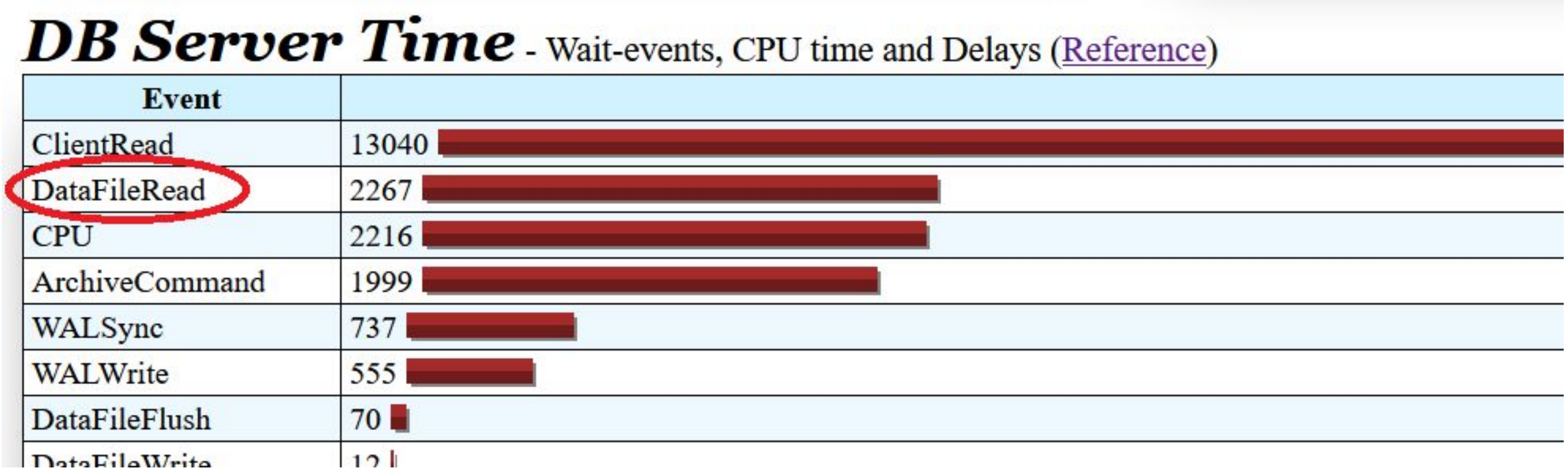
5. Read Amplification due to Read-Ahead

- **PostgreSQL don’t do Direct IO.**
- **Uses Linux Page Cache**
- **Read-Ahead:** The kernel employs a read-ahead mechanism to prefetch data into the page cache. When an application reads a certain amount of data, the kernel will typically read additional pages beyond the requested amount.
- **Random Read** Requests for updating Index

6. Needs more memory to cache

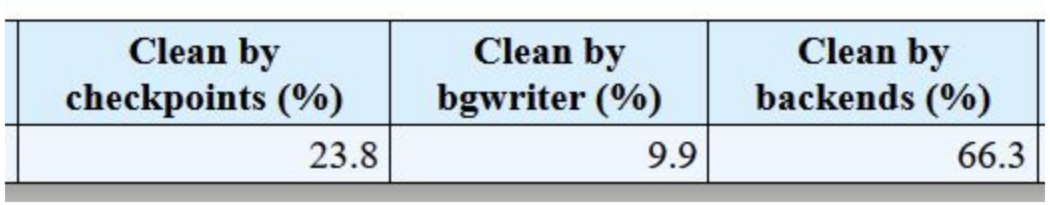
- Random read and write requires more memory for cache

Indications of poor cache



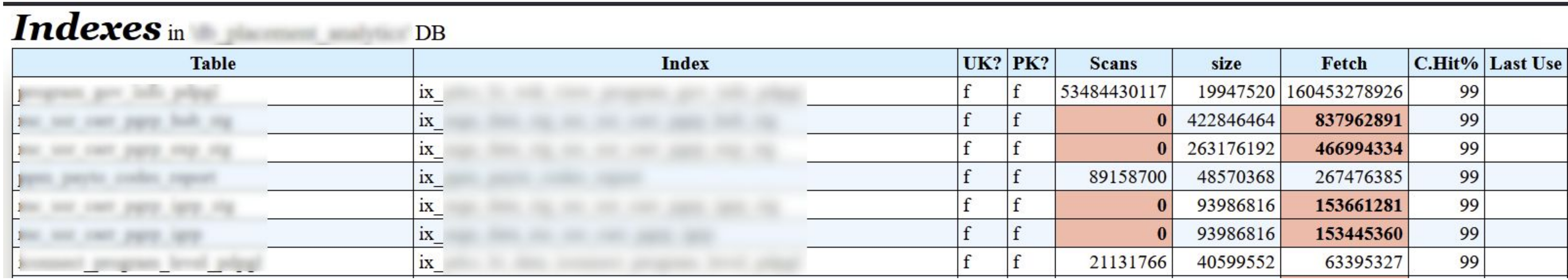
DB Server Time - Wait-events, CPU time and Delays (Reference)

Event	
ClientRead	13040
DataFileRead	2267
CPU	2216
ArchiveCommand	1999
WALSync	737
WALWrite	555
DataFileFlush	70
DataFileWrite	12



Clean by checkpoints (%)	Clean by bgwriter (%)	Clean by backends (%)
23.8	9.9	66.3

Misunderstanding about Unused Indexes



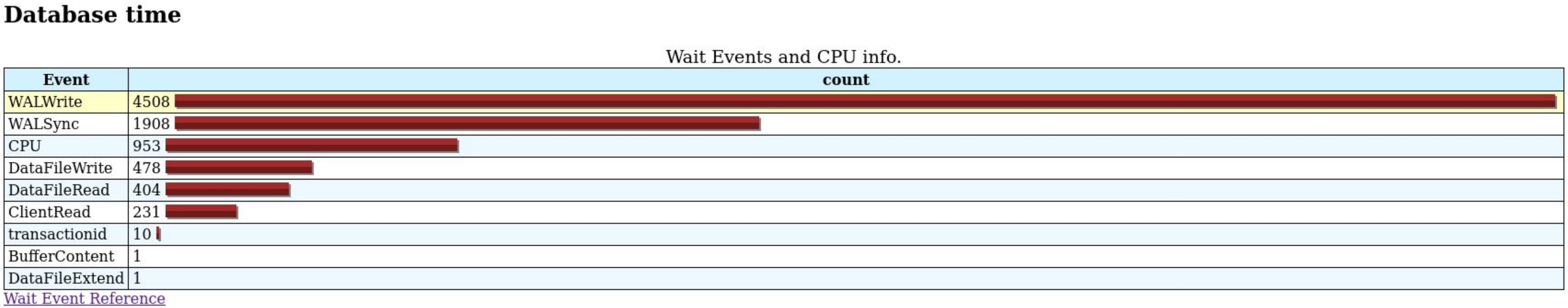
Indexes in [redacted] DB

Table	Index	UK?	PK?	Scans	size	Fetch	C.Hit%	Last Use
[redacted]	ix_[redacted]	f	f	53484430117	19947520	160453278926	99	
[redacted]	ix_[redacted]	f	f	**0**	422846464	**837962891**	99	
[redacted]	ix_[redacted]	f	f	**0**	263176192	**466994334**	99	
[redacted]	ix_[redacted]	f	f	89158700	48570368	267476385	99	
[redacted]	ix_[redacted]	f	f	**0**	93986816	**153661281**	99	
[redacted]	ix_[redacted]	f	f	**0**	93986816	**153445360**	99	
[redacted]	ix_[redacted]	f	f	21131766	40599552	63395327	99	

- Even Unused indexes need to be read into memory for update
- Unused Indexes also need to be updated by every DML
- Unused indexes also consumes memory

All expenses without any yield / benefit

7. Increased WAL generation



Database time

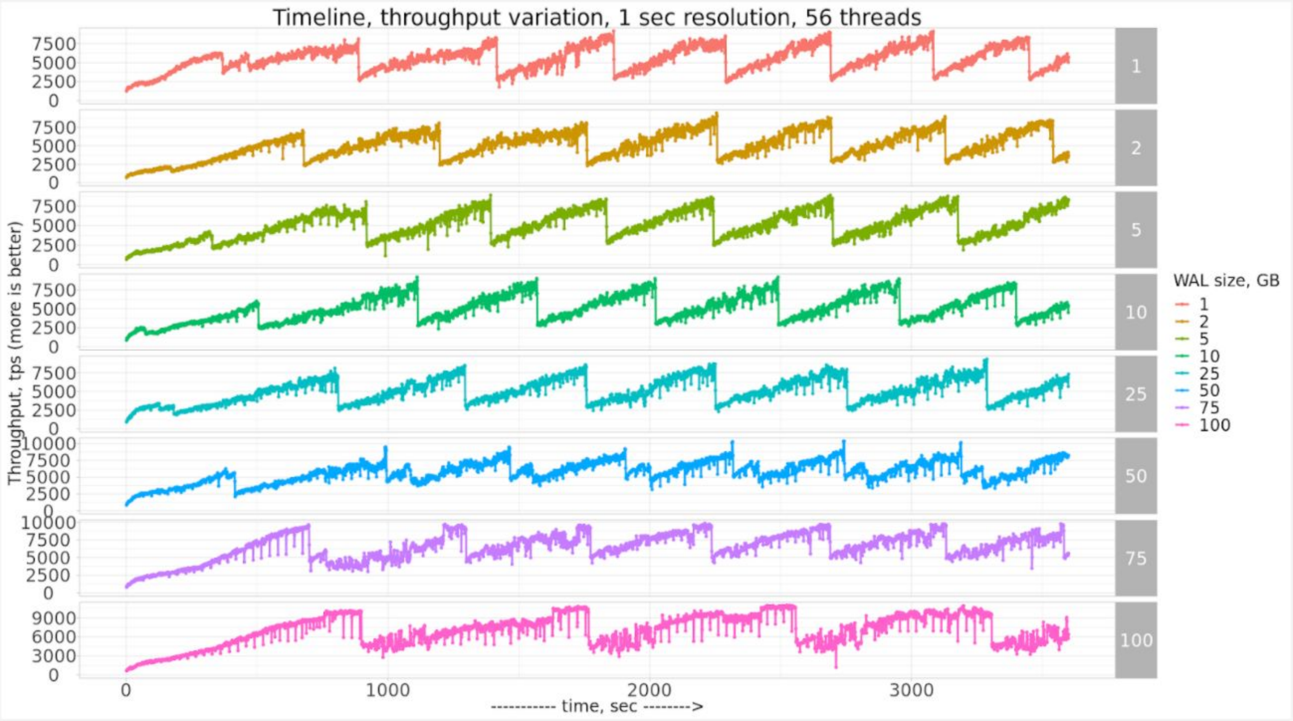
Wait Events and CPU info.

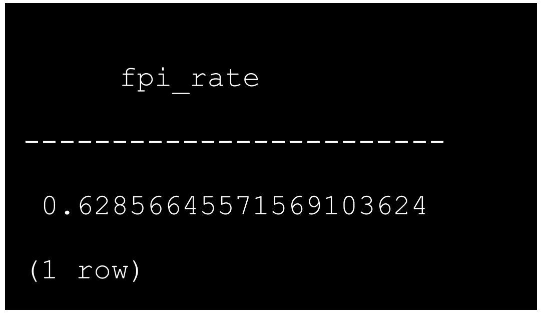
Event	count
WALWrite	4508
WALSync	1908
CPU	953
DataFileWrite	478
DataFileRead	404
ClientRead	231
transactionid	10
BufferContent	1
DataFileExtend	1

Wait Event Reference

- WAL generation rate has direct effect on TPS performance

8. Amplification of WAL Writes because of FPW





```
      fpi_rate
------------------------
 0.62856645571569103624
(1 row)
```

- Effect is multiplied due to **Full Page Writes.**
  - FPW depends on the number of pages dirited after checkpoint

Consequence Excessive WAL generation

- Replication Need more data to be transferred over network
- Replication apply has more work to do.
- Bigger backups, High Backup storage consumption
- WAL archiving not able to catch up with generation

9. Index bloating and Autovacuum

- Indexes also gets bloated
- Gradually indexes becomes inefficient.
- Indexes also need to be vacuumed as part of table vacuum.
- Generally takes more time and effort

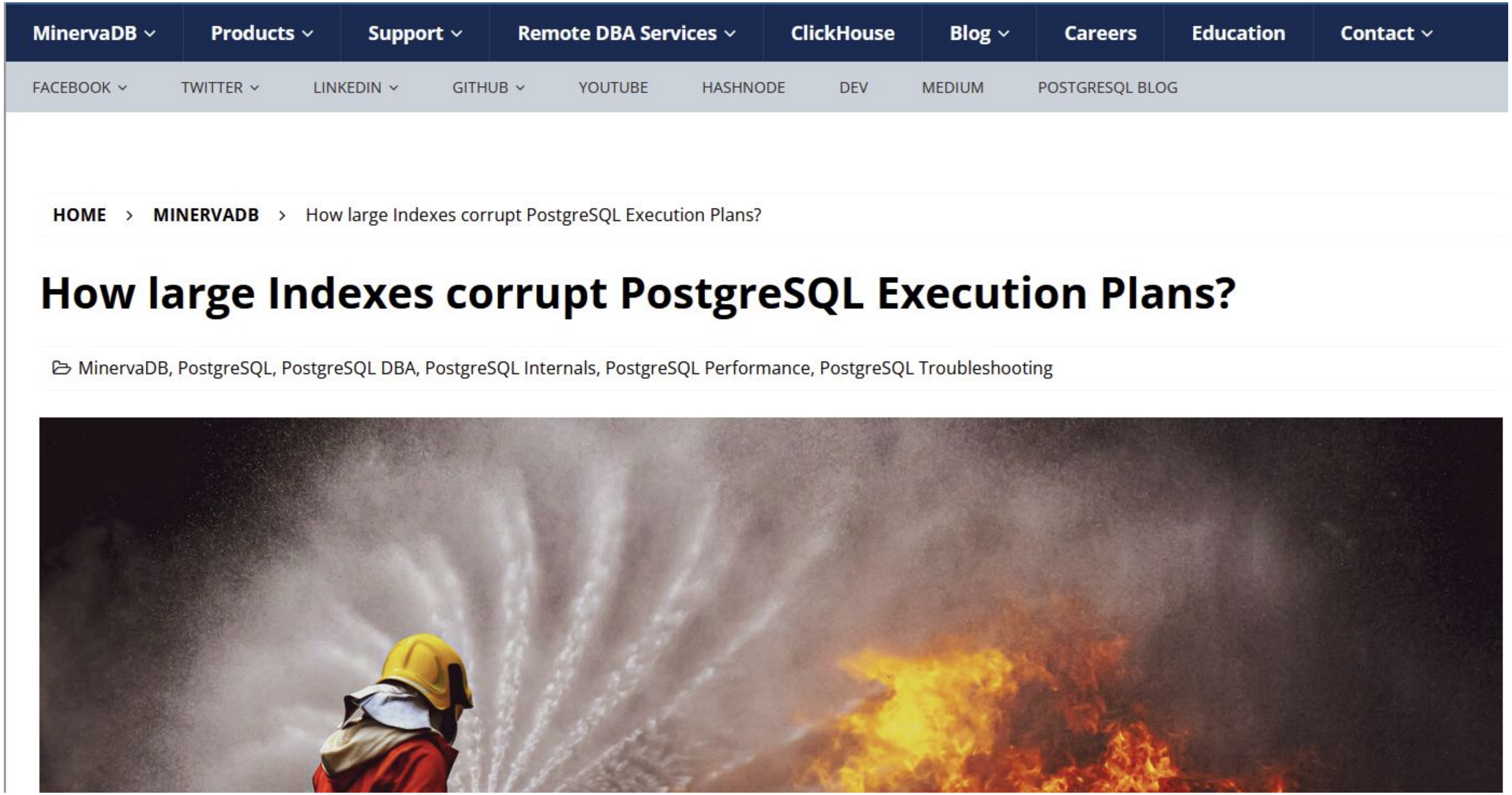
- More processing
- More WAL generation and datafile writes
- More IO.

Indexes prevents HOT update

HOT update optimization is possible only when

- The update **does not modify any columns referenced by the table's indexes**, not including summarizing indexes.

Big Index, Big problem



HOME › **MINERVADB** › How large Indexes corrupt PostgreSQL Execution Plans?

How large Indexes corrupt PostgreSQL Execution Plans?

MinervaDB, PostgreSQL, PostgreSQL DBA, PostgreSQL Internals, PostgreSQL Performance, PostgreSQL Troubleshooting

https://minervadb.xyz/how-large-indexes-corrupt-postgresql-execution-plans/

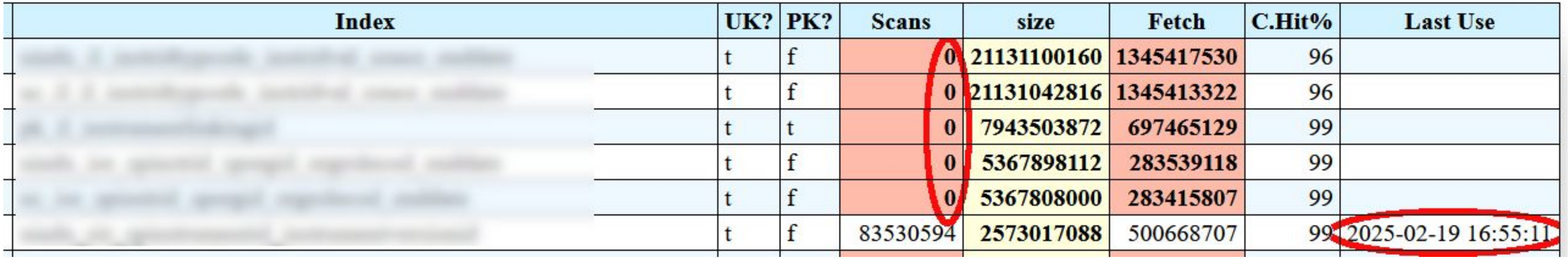
Solutions

Awareness

- Use Indexes only when there is no other way to meet the query performance expectations.
- Be aware about all negative side effects.
- Avoid copying Indexes from other systems. Because performance characteristics of PostgreSQL is different.
- Use tools / Scripts to identify effectiveness of indexes
  - Unused indexes are the first to be removed from the system

Use new versions of PostgreSQL

- Many Index optimizations are added in new features
- Now it is easy to find out when was last time an index is used



Index	UK?	PK?	Scans	size	Fetch	C.Hit%	Last Use
[redacted]	t	f	0	21131100160	1345417530	96	
[redacted]	t	f	0	21131042816	1345413322	96	
[redacted]	t	t	0	7943503872	697465129	99	
[redacted]	t	f	0	5367898112	283539118	99	
[redacted]	t	f	0	5367808000	283415807	99	
[redacted]	t	f	83530594	2573017088	500668707	99	2025-02-19 16:55:11

Thank You!

Title here

Poppins Medium size 18

- Poppins Normal size 18
- Poppins Normal size 18
- Poppins Normal size 18

Poppins Medium size 18

- Poppins Normal size 18
- Poppins Normal size 18
- Poppins Normal size 18

Title here - Poppins Medium size 34 colour #oe1a53

Poppins Normal size 20 - colour #oe1a53

Title Here

Text - Poppins Normal size 18

Title Here

Text - Poppins Normal size 18

Title Here

Text - Poppins Normal size 18

Text - Poppins Normal size 18

Section Cover

Subhead

Section Cover

Subhead

Section Cover

Subhead

Section Cover

Subhead

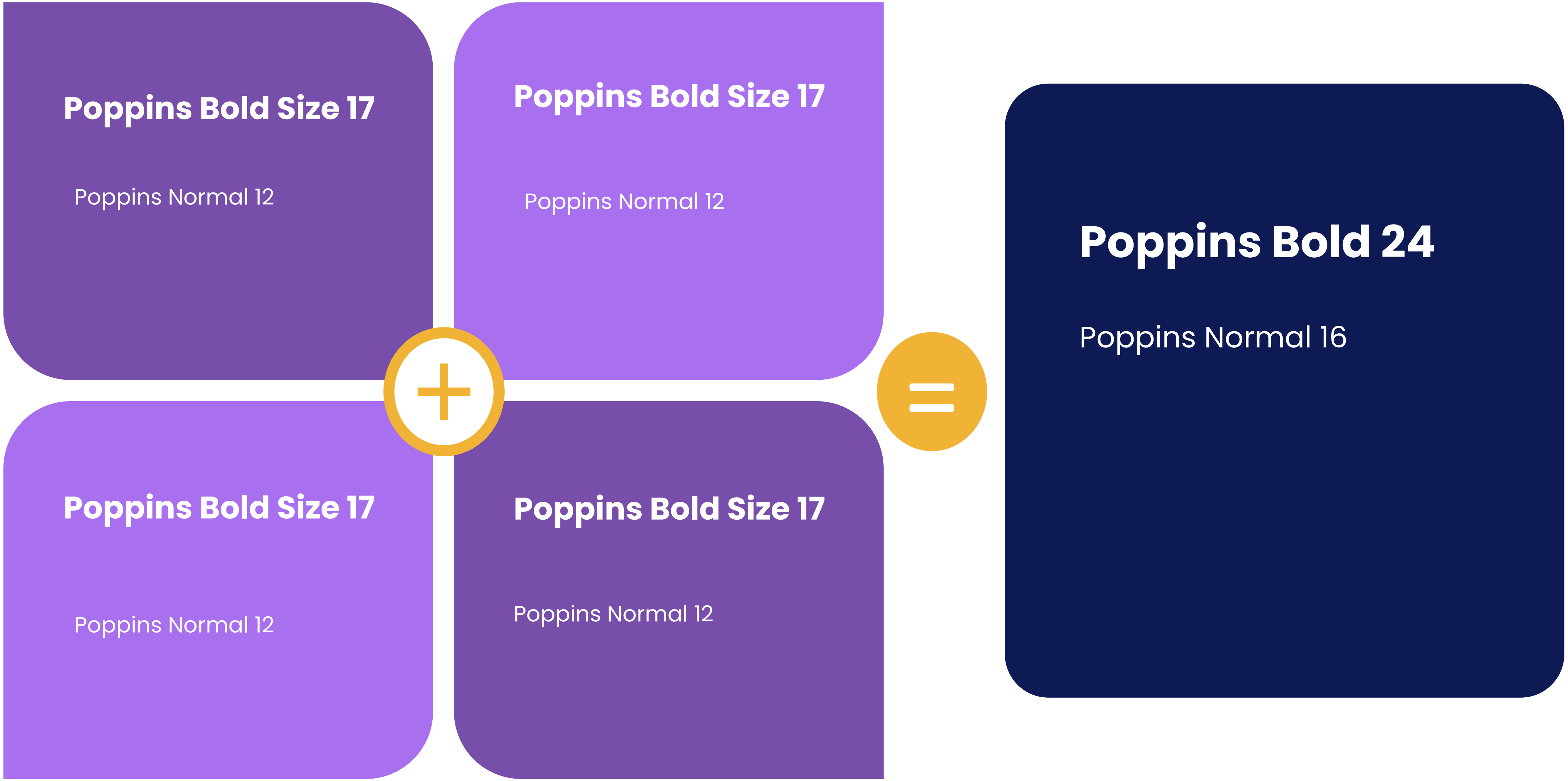
Section Cover

Subhead

Title Goes Here

Subtitle Goes Here

Extras



Poppins Bold Size 17

Poppins Normal 12

Poppins Bold Size 17

Poppins Normal 12

+

Poppins Bold Size 17

Poppins Normal 12

Poppins Bold Size 17

Poppins Normal 12

=

Poppins Bold 24

Poppins Normal 16

Title here -Poppins Medium Size 34

Subtitle - Poppins Semibold Size 18

Poppins Normal size 13

Poppins Normal size 13

Poppins Normal size 13

Poppins Normal size 13

Poppins Normal size 13

Poppins Normal size 13

Title

Subhead

Poppins Normal size 13
Poppins Normal size 13
Poppins Normal size 13
Poppins Normal size 13
Poppins Normal size 13

- Poppins Normal size 13
- Poppins Normal size 13
- Poppins Normal size 13
- Poppins Normal size 13

Title

Subhead

Poppins Normal size 13
Poppins Normal size 13
Poppins Normal size 13
Poppins Normal size 13
Poppins Normal size 13

- Poppins Normal size 13
- Poppins Normal size 13
- Poppins Normal size 13
- Poppins Normal size 13

Title

Subhead

Poppins Normal size 13
Poppins Normal size 13
Poppins Normal size 13
Poppins Normal size 13
Poppins Normal size 13

- Poppins Normal size 13
- Poppins Normal size 13
- Poppins Normal size 13
- Poppins Normal size 13

Title here

Subtitle

Poppins Normal size 13

Poppins Normal size 13

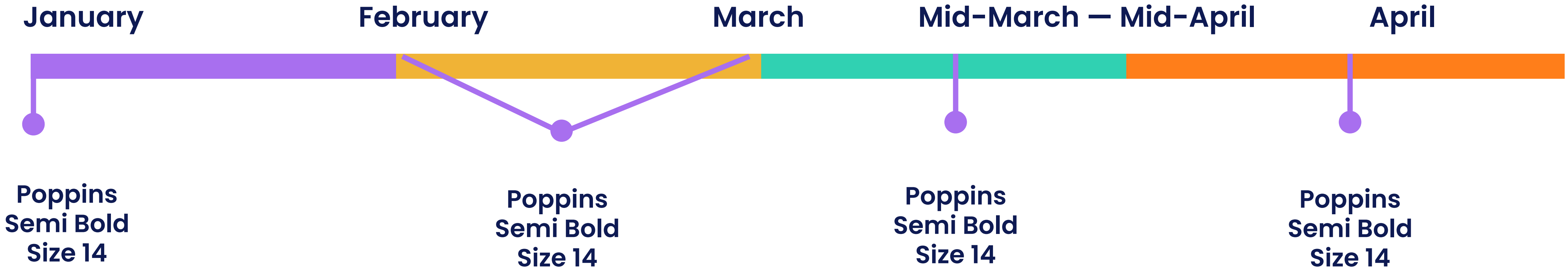
Poppins Normal size 13

Poppins Normal size 13

Poppins Normal size 13

Poppins Normal size 13

Timeline



January

February

March

Mid-March — Mid-April

April

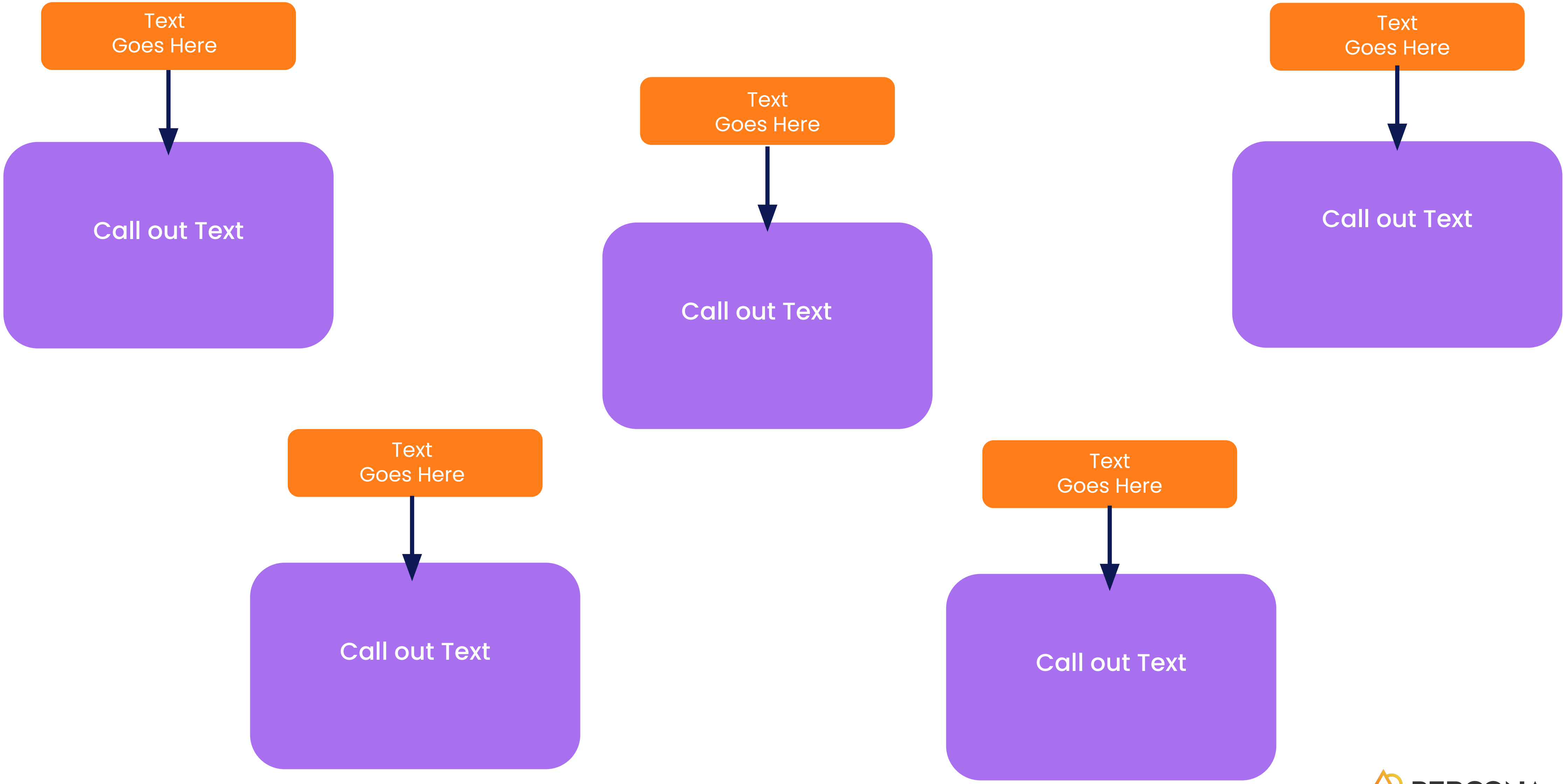
Poppins
Semi Bold
Size 14

Poppins
Semi Bold
Size 14

Poppins
Semi Bold
Size 14

Poppins
Semi Bold
Size 14

Title here



Text
Goes Here

Call out Text

Text
Goes Here

Call out Text

Text
Goes Here

Call out Text

Text
Goes Here

Call out Text

Text
Goes Here

Call out Text