

Transactional Outbox: push- CDC

(TL;DR)

« = , = » — ,
 . «Wakeup Outbox» outbox ,
 : (1) post-commit (0 ,), (2) pg_notify
outbox — PostgreSQL , 10 (DEV Community) , (3)
fallback- (1 → 60) , (sequinstream.com) .
 , (
) : WAL (Debezium, Sequin) —
55–260 (Github) .
 , , NOTIFY, .

1. polling-

1.1 « »

outbox T.
T/2, — T: 500 (glukhov.org) .
1/T — , .
100 100 ,
() , —
« » .
 , :
 , — . , « » :
 , polling- , (Medium) .
« » : ,
(0,) — .

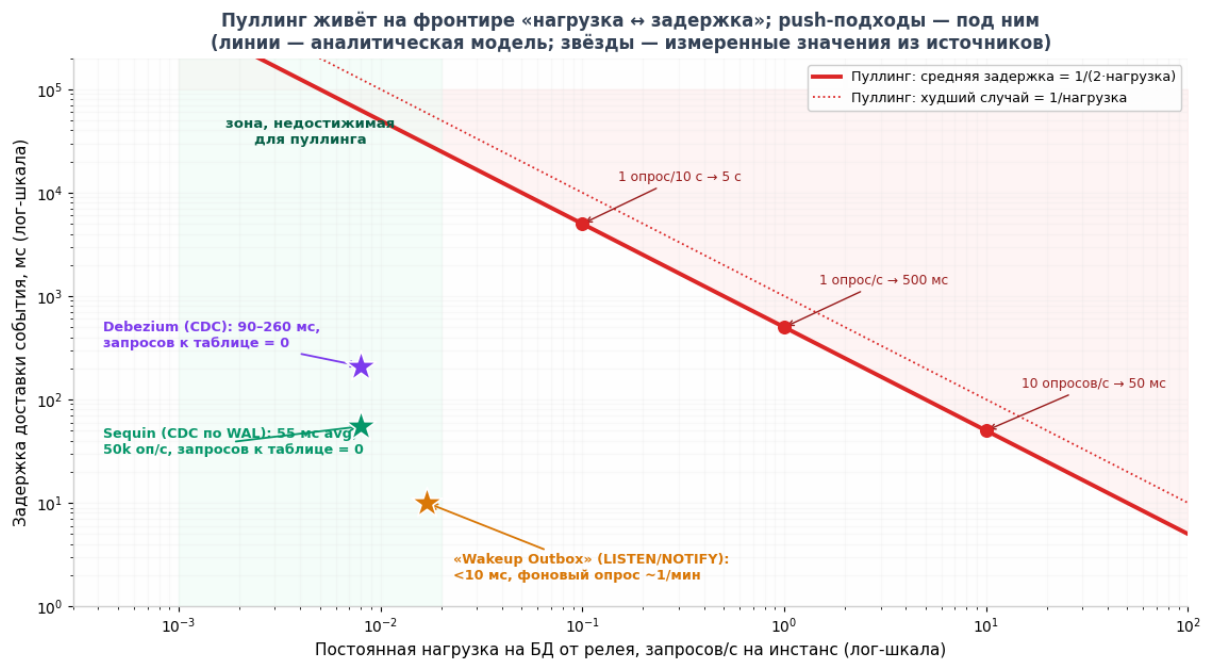


Figure 1: « »

T	()	()	10
50	25	50	20 / 200 /
100	50	100	10 / 100 /
1	500 (glukhov.org)	1	1 / 10 /
10	5	10	0,1 / 1 /

1.2

: (WHERE published_at IS NULL) —
(Gold Lapel) .

SELECT, . - , ,
SELECT ... FOR UPDATE SKIP LOCKED ,
« » ,
(singhajit.com) . - , (MySQL/InnoDB REPEATABLE READ)
gap- , - — ,
outbox- lock wait timeout ; PostgreSQL MVCC
(madewithlove) .

— : ()
, — « », «
, , ». PostgreSQL : LISTEN/
NOTIFY (push-) WAL (change data capture), .

2. PostgreSQL

2.1 LISTEN/NOTIFY:

LISTEN/NOTIFY — PostgreSQL pub/sub, [\(DEV Community\)](#). outbox: , ,
; —
(PostgreSQL) . , , , : ()
(PostgreSQL) . payload, — , 100
outbox , « » (PostgreSQL) ; ()
(The Art of PostgreSQL) ; () payload **8000** , (PostgreSQL)
.
— : at-most-once, , ; ,
— , (sequinstream.com) . NOTIFY
:
10 (DEV Community) . : — 8 ,
(— ,), NOTIFY ;
pg_notification_queue_usage() (PostgreSQL) . NOTIFY,
(PostgreSQL) . , : (PgBouncer transaction mode)
LISTEN advisory- (Gold Lapel) .

2.2 NOTIFY:

NOTIFY , PostgreSQL ,
.
NOTIFY, (AccessExclusiveLock «database 0»)
(recall.ai) . Recall :
, group commit,
fsync — CPU , (recall.ai) . ,
PostgreSQL 19, (dbos.dev) .
DBOS , , ,
— . NOTIFY
:
— **60 000** (**20**) **15–100** (dbos.dev) .
— - , DBOS fallback- (dbos.dev) .
:
(), — **CDC**,
NOTIFY ; —

2.3 WAL pg_logical_emit_message

— : PostgreSQL (WAL)
(INSERT/UPDATE/DELETE) — , ,
WAL- , (packagemain.tech) . wal_level = logical,
CREATE PUBLICATION outbox_pub FOR TABLE outbox (packagemain.tech) ;
pgoutput (PostgreSQL 10) , wal2json
(singhajit.com) . : **outbox** — ,
OLTP- ; CDC- Debezium,

polling- (Medium) . - : Debezium 500
 90 , (Hacker News) .

« » : **pg_logical_emit_message()** **WAL,**
 — (), CDC- (InfoQ) .

outbox: , , . — SQL- ,
 CDC- , , ,
 (InfoQ) . : pg_logical_emit_message — (- ,
 CDC-), : « »,
 CDC- .

3. «Wakeup Outbox»

3.1 :

: **outbox** « → → »,
 , , — :
 , (,), -
 in-memory (goroutine channel, BlockingCollection,). — ,
 — ; , . — - : AFTER INSERT
 outbox pg_notify('outbox_wake', ''), ~10 (DEV Community)
 ; payload , (PostgreSQL) .
 — : , — (,
 LISTEN- ,).
 : (,),
 , , — LISTEN - —
 outbox Go PostgreSQL: , « —
 , » (glukhov.org) . — : (,
 ,), backoff fallback- , NOTIFY
 (3.6), CDC. .

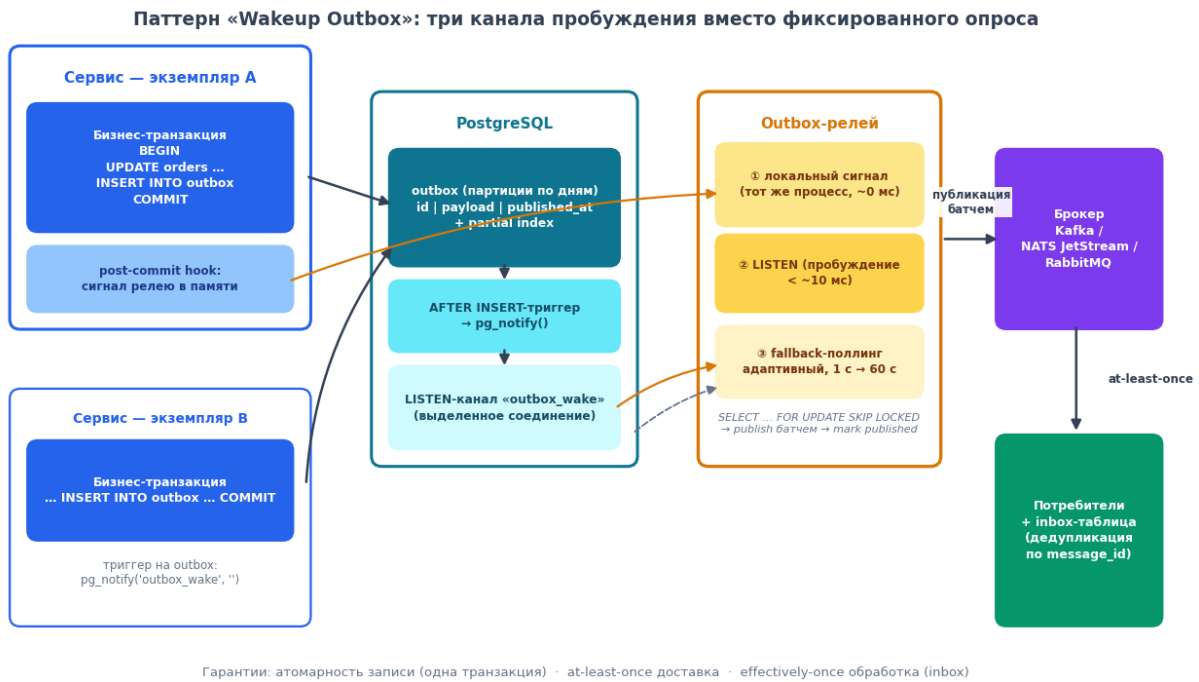


Figure 2: «Wakeup Outbox»

3.2

outbox append-only (4), WAL. (3.5), published_at, « » , :

```
CREATE TABLE outbox (
  id bigserial,
  aggregate_type text NOT NULL, -- для роутинга в топик
  aggregate_id text NOT NULL, -- ключ партиционирования у брокера
  event_type text NOT NULL,
  payload jsonb NOT NULL,
  created_at timestamptz NOT NULL DEFAULT now(),
  published_at timestamptz,
  PRIMARY KEY (created_at, id)
) PARTITION BY RANGE (created_at);

ALTER TABLE outbox REPLICA IDENTITY DEFAULT; -- важно для CDC-варианта [(singhajit.com)](https://singhajit.com/debezium-outbox-postgres-database-impact/)
CREATE INDEX outbox_unpublished_idx ON outbox (id) WHERE published_at IS NULL;

-- Канал : пробуждение слушателей (одна строка — одна точка изменения логики)
CREATE OR REPLACE FUNCTION outbox_wake() RETURNS trigger AS $$
BEGIN
  PERFORM pg_notify('outbox_wake', ''); -- пустой payload: 1 уведомление на транзакцию
  RETURN NULL;
END; $$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER outbox_wake_trg
AFTER INSERT ON outbox
FOR EACH ROW EXECUTE FUNCTION outbox_wake();
```

```
FOR UPDATE SKIP LOCKED —
,
:
,
(DEV Community) .
: → → published_at → ;
( at-least-once inbox — 6).
```

```
// Псевдокод релея (Go-подобный)
func (r *Relay) Run(ctx context.Context) {
    wake := merge(r.localSignal, // [] post-commit hook приложения
                  r.pgListener.Channel(), // [] LISTEN outbox_wake
                  r.backoff.Ticks()) // [] адаптивный fallback-таймер

    for {
        select {
        case <-ctx.Done(): return
        case <-wake:
            n := r.drain(ctx) // цикл захвата, пока батчи не пустеют
            r.backoff.Update(n > 0) // нашли работу → 1 с; пусто → ×2 до 60 с
        }
    }
}

func (r *Relay) claimBatch(ctx context.Context) {
    tx := r.db.Begin()
    rows := tx.Query(`SELECT id, aggregate_type, aggregate_id, event_type, payload
FROM outbox WHERE published_at IS NULL
ORDER BY id LIMIT $1 FOR UPDATE SKIP LOCKED`, batchSize)
    r.publisher.PublishAll(rows) // батч-публикация в брокер
    tx.Exec(`UPDATE outbox SET published_at = now() WHERE id = ANY($1)`, ids(rows))
    tx.Commit()
}
```

3.3 : fallback-

```
Fallback- — NOTIFY ,
; , LISTEN- ,
(8 ; pg_notification_queue_usage()) (PostgreSQL) .
— « », .
: fallback- ,
. Symfony Messenger, ,
~1 , « », — (Gold Lapel) .
: 1 , ( backoff)
30-60 , 1 — , LISTEN ( ,
). push- : ,
, , , —
, 2-3 , .
```

3.4 : ,

— : () (Milan Jovanović) . : SKIP LOCKED , ; « + 5 » ~1 350 **32 500** (Milan Jovanović) . (linger, acks,), (Milan Jovanović) . «Wakeup Outbox» : push- , .

: **OLTP-** , - . , WAL (singhajit.com) . NOTIFY (2.2). : **outbox** — **NOTIFY SKIP LOCKED** ; **WAL** (4), .

3.5 : **DELETE**

outbox , DELETE FROM outbox WHERE published_at IS NOT NULL AND ... — : WAL , , **OLTP-** (singhajit.com) . **created_at** (,) : DROP TABLE outbox_2026_07_30 — - , (singhajit.com) . pg_partman + pg_cron, (simplethread.com) . : (, N-k (DEV Community) . — , (DEV Community) . CDC- (4) : WAL , , (Medium) .

3.6 : **NOTIFY**

«Wakeup Outbox» , — NOTIFY (2.2): (recall.ai) . , DBOS, — : NOTIFY -« », , 50–100 max(id) outbox , (dbos.dev) . group commit. ~50–100 , — 20 per-tx NOTIFY (dbos.dev) . : (SELECT max(id)). : trigger (,), coalesced (,), off (, CDC). — .

3.7

— PostgreSQL, N .

(LISTEN/NOTIFY) — : PostgreSQL , — ,

pg_notify (PostgreSQL) . « » N — ,

— : FOR UPDATE SKIP LOCKED ,

outbox , (DEV Community)

— , « » , —

leader election (,)

work stealing: , — .

() — : , .

— , NOTIFY , « » (

— , ,) , (fallback-)

— backoff 60 N , .

: - , —

work stealing; - , — (,),

aggregate_id, 6.2 (Milan Jovanović) . «

» advisory- (pg_try_advisory_lock), « + »,

work stealing : , .

4. : CDC-

4.1 Debezium, Sequin

CDC- — outbox , SQL- ,

WAL . — Debezium outbox.EventRouter (SMT):

outbox, SMT Kafka aggregate_type, aggregate_id

() id (redhat.com) .

polling- CDC- : , ,

SKIP LOCKED () (Medium) . : Sequin Debezium

23 . / 210 (AWS MSK — 6 . / 258), Sequin

— 50 . / 55 253 p99 (Github) . ()

Debezium ~90 , (Hacker News) .

Kafka Connect , — Sequin :

Kafka, NATS, RabbitMQ, Redis Streams, SQS, ,

— Prometheus (Github) . — -

: pglogrepl/pgx Go (go-pq-cdc , heartbeat) (Github) , pgjdbc

replication API Java, psycopg logical replication Python. , «outbox

→ » : pgoutput- , LSN .

3.2: REPLICA IDENTITY DEFAULT (FULL,

WAL), FOR TABLE outbox - , (singhajit.com)

.

4.2 pg_logical_emit_message: outbox

pg_logical_emit_message(true,

'order_events', payload), WAL — CDC-

```

-- (InfoQ) . «outbox outbox»: , (
Debezium Flink CDC) (InfoQ) . —
, , - . : SQL ( « »
), , payload , CDC-
WAL (InfoQ) .
: pg_logical_emit_message , CDC-
, . outbox , ;
, , «Wakeup Outbox», CDC,
— .

```

4.3

```

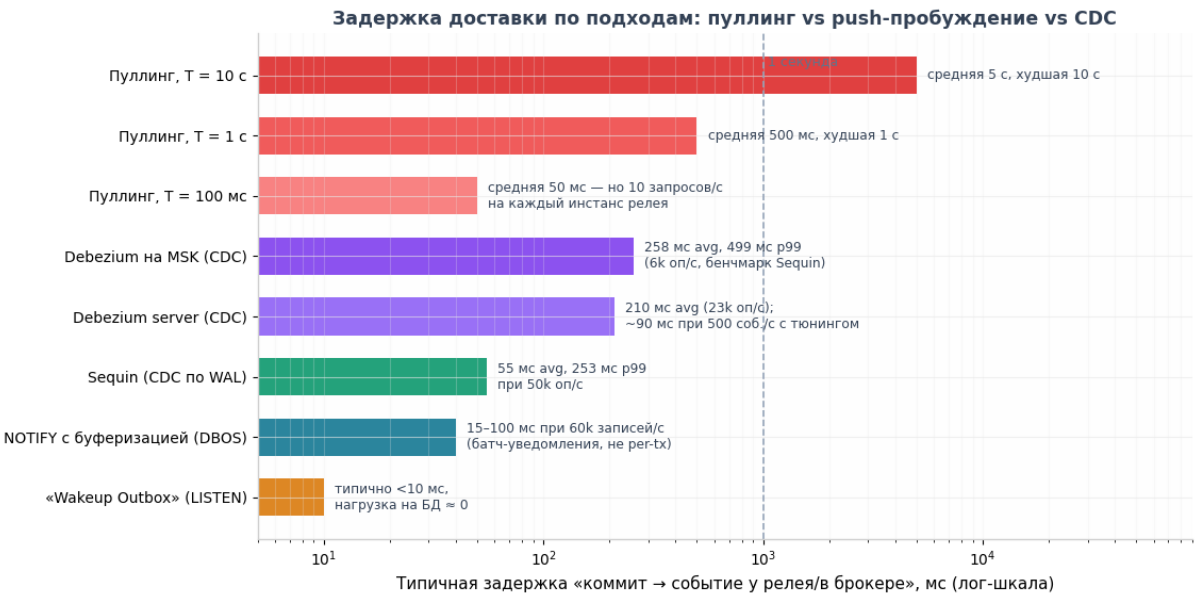
CDC- — WAL: , ,
; Debezium+outbox (peerdb.io) .
: max_slot_wal_keep_size ( , ),
(pg_wal_lsn_diff(pg_current_wal_lsn(), restart_lsn) pg_replication_slots) ,
heartbeat- , (peerdb.io) . — ,
LSN ; pg_stat_activity (singhajit.com) . —
: reorder buffer , logical_decoding_work_mem
(peerdb.io) .

- : WAL 1
5 ( max_slot_wal_keep_size), 5 , pg_wal/ 60/80 %,
MilliSecondsBehindSource Debezium (singhajit.com) . :
( , « » CDC « ») —
; WAL , (singhajit.com) .
: , , — .

```

5.

. — (: = $T/2$),
 ; , , , SLA.



WAL.					
(T		«Wakeup	CDC-	pg_logical_emit_m	
= 1)	(backoff)	Outbox» ()	(Debezium / Sequin)		
~500 (glukhov.org)	0,5-30 (backoff)	<10 (DEV Community) ; 15-100 (dbos.dev)	55 (Sequin) - 260 (Debezium/ MSK) (Github) ; ~90 500 ./ (Hacker News)	CDC	
1 /	~1/ , 1/	~1/ (fallback) + ()	0 ; CPU 0 ; WAL		
- ./ ; ~32,5 . ./ 5 (Milan Jovanović) id ; (Milan Jovanović) (-)		(fallback)	/	/	
			(Github) ; (singhajit.com) (Medium)		
			(= durable-) Kafka Connect / Sequin + CDC-		
× ,		tx			
(madewithlove)		(recall.ai)	(peerdb.io)		

Table 2 – continued

	(T = 1)	(backoff)	«Wakeup Outbox» ()	CDC- (Debezium / Sequin)	pg_logical_emit_m
	, <1 ./		PostgreSQL: -	+ ./ Kafka	CDC- ,

: «Wakeup Outbox» , —
 , LISTEN/NOTIFY, , SKIP LOCKED PostgreSQL. CDC-
 , : , ,
« » . (9), « » .

6.

6.1 outbox at-least-once effectively-once

: outbox at-least-once ,
« » — (event-driven.io) .
 : ACK, - , visibility timeout
(Matheus Palma) . « » «at-least-once +
», Inbox ():
consumer_inbox (consumer_name, message_id), -
 ; ON CONFLICT DO NOTHING no-op (Matheus Palma) .
«Wakeup Outbox» . : id outbox
— Kafka (SMT EventRouter), JetStream
Nats-Msg-Id (redhat.com) . : inbox-
 , « » processing , failed, — TTL-
(Matheus Palma) . : inbox , (, email);
 , effectively-once (Matheus Palma) .

6.2 :

. , ORDER BY id,
— bigserial . SKIP LOCKED,
 : , B A (Milan Jovanović) .
 , : = aggregate_id,
(Kafka: →), , /
(risingwave.com) . , CDC- :
(Medium) .
 : inbox , — OrderCreated OrderCancelled
 , - (Matheus Palma) .
 : , — ,

```
«outbox → → inbox» ;
( CDC) , .
```

7.

```
 , outbox- . — : —
- , ( , read- ),
- retention replay — Kafka NATS JetStream; « - », RabbitMQ
(BackendBytes) . : Kafka — +
(offset — ); JetStream 200–400 . / , —
, Nats-Msg-Id; RabbitMQ — 50–100 .
/ , 5–20 , (Premium Cloud Hosting Redefined) .
: «Wakeup Outbox» producer- , Debezium
Kafka, Sequin Kafka, NATS, RabbitMQ (Github) .
: NATS JetStream ( , , ),
Kafka — (Kafka Connect, Schema Registry, - , retention)
, RabbitMQ — task-queue (BackendBytes) .
, 6 : , message_id , inbox .
```

8.

```
 : ,
, : , ( ,
, ) —
(Medium) . : outbox ( published_at IS NULL — )
( — - ; = ) (DEV Community) ; lag
= now() — created_at
(Medium) . : LISTEN- ( / ), pg_notification_queue_usage()
, fallback- , — push- .
CDC- : (pg_replication_slots), pg_wal/,
(MillisecondsBehindSource Debezium), ; «1 / 5 »
«60/80 %» (singhajit.com) . runbook: -
( , ) (PostgreSQL) ; « » advisory-
LISTEN PgBouncer transaction- ( ) (Gold Lapel) ;
(heartbeat) (singhajit.com) ; REPLICA IDENTITY FULL, outbox ( WAL UPDATE)
(singhajit.com) ; DELETE ( ) (singhajit.com) .
```

9. MySQL: outbox LISTEN/NOTIFY

9.1 «Wakeup Outbox»,

```
MySQL LISTEN/NOTIFY, PostgreSQL- — .
: outbox , FOR UPDATE SKIP LOCKED (
MySQL 8.0.1) (tusacentral.com) , DROP PARTITION (MySQL RANGE- ; ,
, PK (id, created_at)) (keboola.com) ,
```

inbox fallback- . : push- ,
 - pub/sub (Redis,), binlog-CDC, MySQL
 « », WAL PostgreSQL.
 , MySQL- :
 (, ,) — « »
 fallback : 1-5 .
 , - post-commit « » Redis Pub/Sub
 (— « »): , ;
 fallback. (),
 « » , .

9.2 MySQL: gap- InnoDB

MySQL- , PostgreSQL , — outbox- . InnoDB
 — REPEATABLE READ, SELECT ... FOR UPDATE
 , gap- — « » (oneuptime.com)
 : WHERE published_at IS NULL ... FOR UPDATE SKIP LOCKED
 (published_at, created_at) gap NULL- — ,
 outbox; 100 , INSERT outbox ,
 innodb_lock_wait_timeout (50) Lock wait timeout exceeded (madewithlove) . ,
 INSERT- , EXPLAIN , slow log , —
 TRANSACTIONS SHOW ENGINE INNODB STATUS (madewithlove) .
 : READ COMMITTED (SET TRANSACTION ISOLATION LEVEL READ
 COMMITTED BEGIN) — InnoDB gap- , SKIP LOCKED
 (madewithlove) . gap- : ,
 (madewithlove) . (—
). 3.2: MySQL , WHERE published_at IS NULL
 — (published_at, id), ,
 (), «outbox () + outbox_pending ()»,
 pending — DELETE InnoDB ,
 OPTIMIZE TABLE - .

9.3 CDC- MySQL: binlog WAL

MySQL , : (binlog) —
 , WAL, — MySQL. Debezium MySQL-
 binlog ; outbox.EventRouter SMT PostgreSQL-
 — aggregate_type, aggregate_id, id (Medium) .
 : binlog_format = ROW (; STATEMENT/MIXED), binlog_row_image = FULL,
 gtid_mode = ON (binlog_expire_logs_seconds, 7
) (PipeCode) . : SELECT, RELOAD, SHOW DATABASES, REPLICATION SLAVE, REPLICATION
 CLIENT (PipeCode) . — Canal Maxwell (binlog-), — Sequin
 MySQL.
 PostgreSQL: CDC- binlog- (singhajit.com) . outbox
 — , pg_wal/), MySQL
 CDC : , — MySQL

DROP PARTITION , (PipeCode) . :
MySQL 8.x → « fallback + + READ COMMITTED + SKIP LOCKED »
, Redis/ - - , binlog-CDC (Debezium/Canal/Sequin)
— , PostgreSQL, (« ») ,
(CDC) MySQL.

10.

— , . 1 (): backoff
90 %+ . 2 (): pg_notify payload
LISTEN- — «Wakeup Outbox» <10 ;
. 3: post-commit hook , — .
4: , . 5: inbox —
. 6 (): Kafka
— CDC (Sequin Debezium), , .
, , : (
NOTIFY, 3.6), (CDC , 6.2), (
, 6.1). :
, — , , outbox, — -
, .

- ([glukhov.org](https://www.glukhov.org/app-architecture/integration-patterns/transactional-outbox-pattern-go/)): [https:// www.glukhov.org/ app-architecture/ integration-patterns/ transactional-outbox-pattern-go/](https://www.glukhov.org/app-architecture/integration-patterns/transactional-outbox-pattern-go/)
- ([risingwave.com](https://risingwave.com/blog/debezium-outbox-pattern-microservices/)): <https://risingwave.com/blog/debezium-outbox-pattern-microservices/>
- ([Medium](https://medium.com/event-driven-utopia/sending-reliable-event-notifications-with-transactional-outbox-pattern-7a7c69158d1b)): [https:// medium.com/ event-driven-utopia/ sending-reliable-event-notifications-with-transactional-outbox-pattern-7a7c69158d1b](https://medium.com/event-driven-utopia/sending-reliable-event-notifications-with-transactional-outbox-pattern-7a7c69158d1b)
- ([DEV Community](https://dev.to/preshyjones/the-outbox-pattern-in-payment-systems-58dj)): <https://dev.to/preshyjones/the-outbox-pattern-in-payment-systems-58dj>
- ([madewithlove](https://madewithlove.com/blog/the-query-that-was-correct-but-broke-everything-else/)): <https://madewithlove.com/blog/the-query-that-was-correct-but-broke-everything-else/>
- ([FreeCodecamp](https://www.freecodecamp.org/news/how-to-implement-the-outbox-pattern-in-go-and-postgresql/)): [https:// www.freecodecamp.org/ news/ how-to-implement-the-outbox-pattern-in-go-and-postgresql/](https://www.freecodecamp.org/news/how-to-implement-the-outbox-pattern-in-go-and-postgresql/)
- ([PostgreSQL](https://www.postgresql.org/docs/current/sql-notify.html)): <https://www.postgresql.org/docs/current/sql-notify.html>
- ([Medium](https://medium.com/@nustianrwp/the-transactional-outbox-pattern-a-rigorous-examination-for-distributed-systems-engineers-9c189836f470)): [https:// medium.com/ @nustianrwp/ the-transactional-outbox-pattern-a-rigorous-examination-for-distributed-systems-engineers-9c189836f470](https://medium.com/@nustianrwp/the-transactional-outbox-pattern-a-rigorous-examination-for-distributed-systems-engineers-9c189836f470)
- ([DEV Community](https://dev.to/polliog/building-real-time-log-streaming-with-postgresql-listennotify-4cbj)): [https:// dev.to/ polliog/ building-real-time-log-streaming-with-postgresql-listennotify-4cbj](https://dev.to/polliog/building-real-time-log-streaming-with-postgresql-listennotify-4cbj)
- ([sequinstream.com](https://blog.sequinstream.com/all-the-ways-to-capture-changes-in-postgres/)): <https://blog.sequinstream.com/all-the-ways-to-capture-changes-in-postgres/>
- ([Milan Jovanović](https://milanjovanovic.tech/blog/scaling-the-outbox-pattern)): <https://milanjovanovic.tech/blog/scaling-the-outbox-pattern>

- ([singhajit.com](https://singhajit.com/debezium-outbox-postgres-database-impact/)): <https://singhajit.com/debezium-outbox-postgres-database-impact/>
- ([redhat.com](https://docs.redhat.com/en/documentation/red_hat_build_of_debezium/1.9.7/html/debezium_user_guide/applying-transformations-to-modify-messages-exchanged-with-kafka)): https://docs.redhat.com/en/documentation/red_hat_build_of_debezium/1.9.7/html/debezium_user_guide/applying-transformations-to-modify-messages-exchanged-with-kafka
- ([InfoQ](https://www.infoq.com/articles/wonders-of-postgres-logical-decoding-messages/)): <https://www.infoq.com/articles/wonders-of-postgres-logical-decoding-messages/>
- ([Github](https://github.com/sequinstream/sequin)): <https://github.com/sequinstream/sequin>
- ([Github](https://github.com/Trendyol/go-pq-cdc)): <https://github.com/Trendyol/go-pq-cdc>
- ([Hacker News](https://news.ycombinator.com/item?id=42582203)): <https://news.ycombinator.com/item?id=42582203>
- ([peerdb.io](https://blog.peerdB.io/overcoming-pitfalls-of-postgres-logical-decoding)): <https://blog.peerdB.io/overcoming-pitfalls-of-postgres-logical-decoding>
- ([packagemain.tech](https://packagemain.tech/p/real-time-database-change-tracking)): <https://packagemain.tech/p/real-time-database-change-tracking>
- ([Matheus Palma](https://matheuspalma.com/blog/consumer-inbox-pattern-idempotent-message-processing)): <https://matheuspalma.com/blog/consumer-inbox-pattern-idempotent-message-processing>
- ([Milan Jovanović](https://milanjovanovic.tech/blog/implementing-the-inbox-pattern-for-reliable-message-consumption)): <https://milanjovanovic.tech/blog/implementing-the-inbox-pattern-for-reliable-message-consumption>
- ([Hacker News](https://news.ycombinator.com/item?id=44490510)): <https://news.ycombinator.com/item?id=44490510>
- ([dbos.dev](https://www.dbos.dev/blog/postgres-listen-notify-scalability)): <https://www.dbos.dev/blog/postgres-listen-notify-scalability>
- ([simplethread.com](https://www.simplethread.com/beyond-delete/)): <https://www.simplethread.com/beyond-delete/>
- ([event-driven.io](https://event-driven.io/en/outbox_inbox_patterns_and_delivery_guarantees_explained/)): https://event-driven.io/en/outbox_inbox_patterns_and_delivery_guarantees_explained/
- ([Gold Lapel](https://goldlapel.com/grounds/laravel-php/symfony-messenger-postgres-listen-notify)): <https://goldlapel.com/grounds/laravel-php/symfony-messenger-postgres-listen-notify>
- ([recall.ai](https://www.recall.ai/blog/postgres-listen-notify-does-not-scale)): <https://www.recall.ai/blog/postgres-listen-notify-does-not-scale>
- ([BackendBytes](https://backendbytes.com/articles/message-queue-comparison/)): <https://backendbytes.com/articles/message-queue-comparison/>
- ([Premium Cloud Hosting Redefined](https://onidel.com/blog/nats-jetstream-rabbitmq-kafka-2025-benchmarks)): <https://onidel.com/blog/nats-jetstream-rabbitmq-kafka-2025-benchmarks>
- ([vitalvas.com](https://blog.vitalvas.com/post/2025/11/06/kafka-vs-nats-jetstream/)): <https://blog.vitalvas.com/post/2025/11/06/kafka-vs-nats-jetstream/>
- ([The Art of PostgreSQL](https://tapoueh.org/blog/2018/07/postgresql-listen/notify/)): <https://tapoueh.org/blog/2018/07/postgresql-listen/notify/>
- ([Medium](https://medium.com/@gaddamnaveen192/stop-polling-your-outbox-table-this-is-how-debezium-actually-works-in-kafka-6f0e4a3d3d12)): <https://medium.com/@gaddamnaveen192/stop-polling-your-outbox-table-this-is-how-debezium-actually-works-in-kafka-6f0e4a3d3d12>
- ([oneuptime.com](https://oneuptime.com/blog/post/2026-03-31-mysql-skip-locked-and-nowait-mysql-8/view)): <https://oneuptime.com/blog/post/2026-03-31-mysql-skip-locked-and-nowait-mysql-8/view>
- ([oneuptime.com](https://oneuptime.com/blog/post/2026-03-31-mysql-innodb-gap-locks/view)): <https://oneuptime.com/blog/post/2026-03-31-mysql-innodb-gap-locks/view>
- ([PipeCode](https://pipecode.ai/blogs/debezium-connectors-snapshot-schema-history-outbox)): <https://pipecode.ai/blogs/debezium-connectors-snapshot-schema-history-outbox>
- ([tusacentral.com](http://www.tusacentral.com/joomla/index.php/mysql-blogs/224-using-skip-lock-in-mysql-for-queue-processing)): <http://www.tusacentral.com/joomla/index.php/mysql-blogs/224-using-skip-lock-in-mysql-for-queue-processing>
- ([keboola.com](https://500.keboola.com/improving-mysql-performance-with-table-partitioning/)): <https://500.keboola.com/improving-mysql-performance-with-table-partitioning/>