

PostgreSQL 19 新機能

- REPACK コマンド -

2026-10-03

オープンソースカンファレンス 2026 Online/Fall

特定非営利活動法人 日本 PostgreSQL ユーザ会

理事 高塚 遥

本講演について

- 2026 年秋にリリース予定の PostgreSQL 19 の新機能の 1つ 「REPACK」 コマンドについて、解説します。
- 従来からあったいくつかのサードパーティ拡張との比較や、動作の詳細について、掘り下げます。

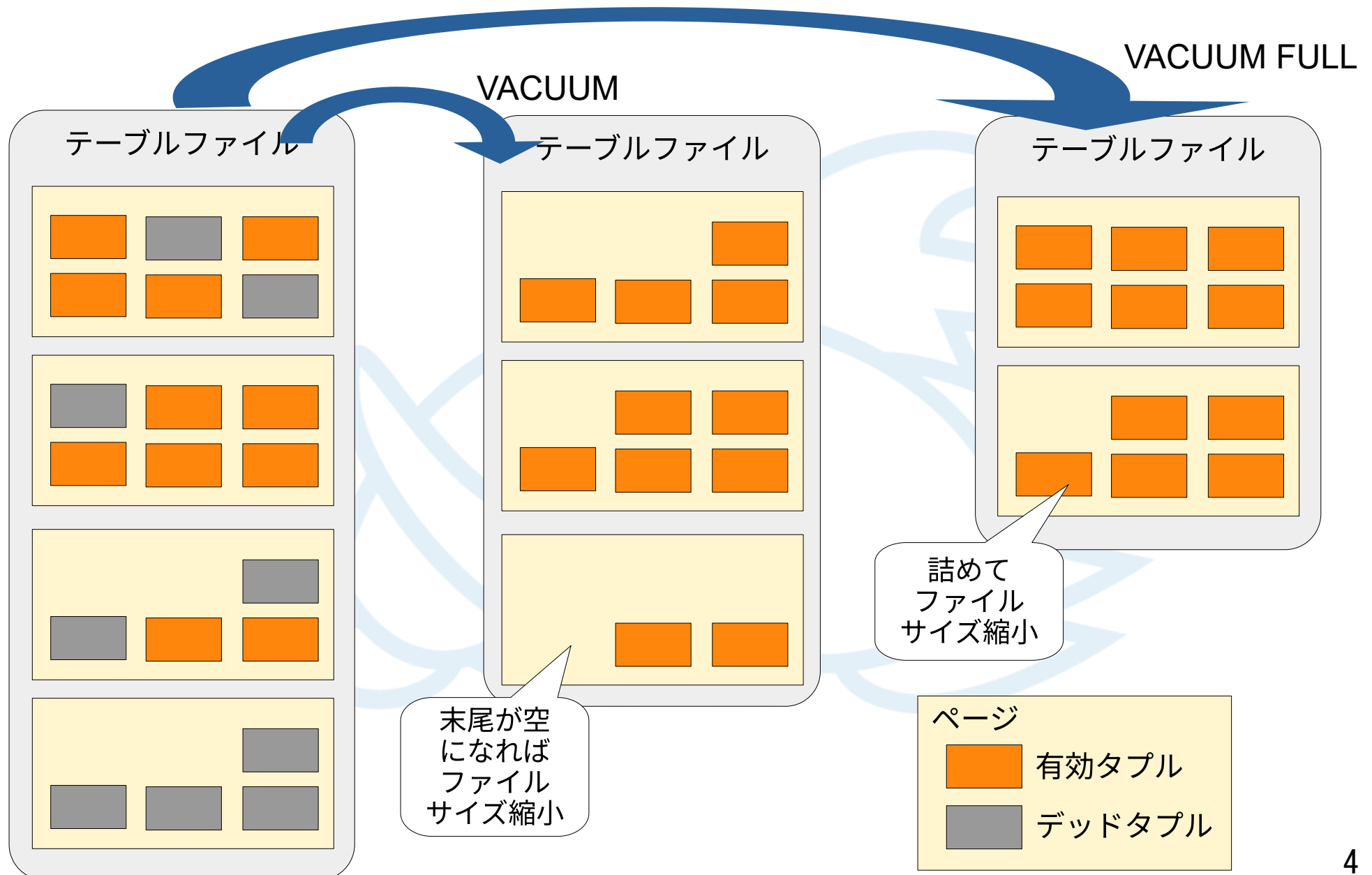
VACUUM の呪い

- 大昔から「VACUUM があってダメ」と言われてきた
 - VACUUM: 不要領域を片付けて、古い XID を凍結する
 - PostgreSQL のテーブルは追記型
 - 即時上書き削除せずロールバックに使うので不要領域がでる

VACUUM の改善はされているが...	
7.2 (2002 年)	コンカレント VACUUM
8.1 (2005 年)	autovacuum デフォルト有効
8.3 (2008 年)	HOT update - VACUUM 負荷軽減
8.4 (2009 年)	可視性マップ - VACUUM 負荷軽減
9.6 (2016 年)	XID 凍結情報付き可視性マップ
13 (2020 年)	インデックスパラレル VACUUM
17 (2024 年)	VACUUM の大幅な省メモリ・高速化



VACUUM と VACUUM FULL



VACUUM FULL をオンラインで

- **VACUUM FULL には依然として排他ロックが発生**
 - オンライン実行には外部ツールが必要
- pg_repack (pg_reorg)
 - 読み書きと並行して VACUUM FULL
 - 実行中の変更をトリガで一時テーブルに貯めて最後に適用
 - 短時間の排他ロックでテーブルファイル入れ替え
- pg_squeeze
 - pg_repack と類似のアーキテクチャ
 - トリガではなくロジカルデコーディングで実現
 - 自動実行機能

PostgreSQL 19 では

- VACUUM FULL
 - "This option, which is **deprecated**, makes VACUUM behave like **REPACK** without a USING INDEX clause."
 - VACUUM FULL と CLUSTER コマンドは REPACK コマンドに統合された
 - CLUSTER は指定インデックス順に物理順を並び変えるコマンドで VACUUM FULL と同様の効果
- REPACK [CONCURRENTLY]
 - オンライン実行が標準機能に組み込まれた
 - ロジカルデコーディング方式

REPACK コマンド

```
db1=# REPACK tbl01;  
REPACK
```

→ VACUUM FULL コマンドと同じ

```
db1=# REPACK tbl02 WITH INDEX idx01_tbl02;  
REPACK
```

→ CLUSTER コマンドと同じ、行の並び替えも行なわれる

```
db1=# REPACK (CONCURRENTLY) tbl03;  
REPACK
```

→ オンライン VACUUM FULL 実行になる

```
db1=# REPACK (CONCURRENTLY, VERBOSE) tbl04;  
INFO:  repacking "public.tbl04" in physical order  
INFO:  "public.tbl04": found 0 removable, 1000000 nonremovable row  
       versions in 18240 pages  
DETAIL:  0 dead row versions cannot be removed yet.  
CPU: user: 0.49 s, system: 0.26 s, elapsed: 1.23 s.  
REPACK
```

VERBOSE 指定
で結果情報が
報告される

進捗ビューー pg_stat_progress_repack

```
db1=# SELECT * FROM pg_stat_progress_repack;
```

```
-[ RECORD 1 ]-----+-----
```

pid	1957
datid	25870
datname	db1
relid	25878
command	REPACK
phase	seq scanning heap
repack_index_relid	0
heap_tuples_scanned	853491
heap_tuples_inserted	853491
heap_tuples_updated	0
heap_tuples_deleted	0
heap_blks_total	18240
heap_blks_scanned	15519
index_rebuild_count	0

進捗情報は
ビューから
確認できる

REPACK の処理フェーズ

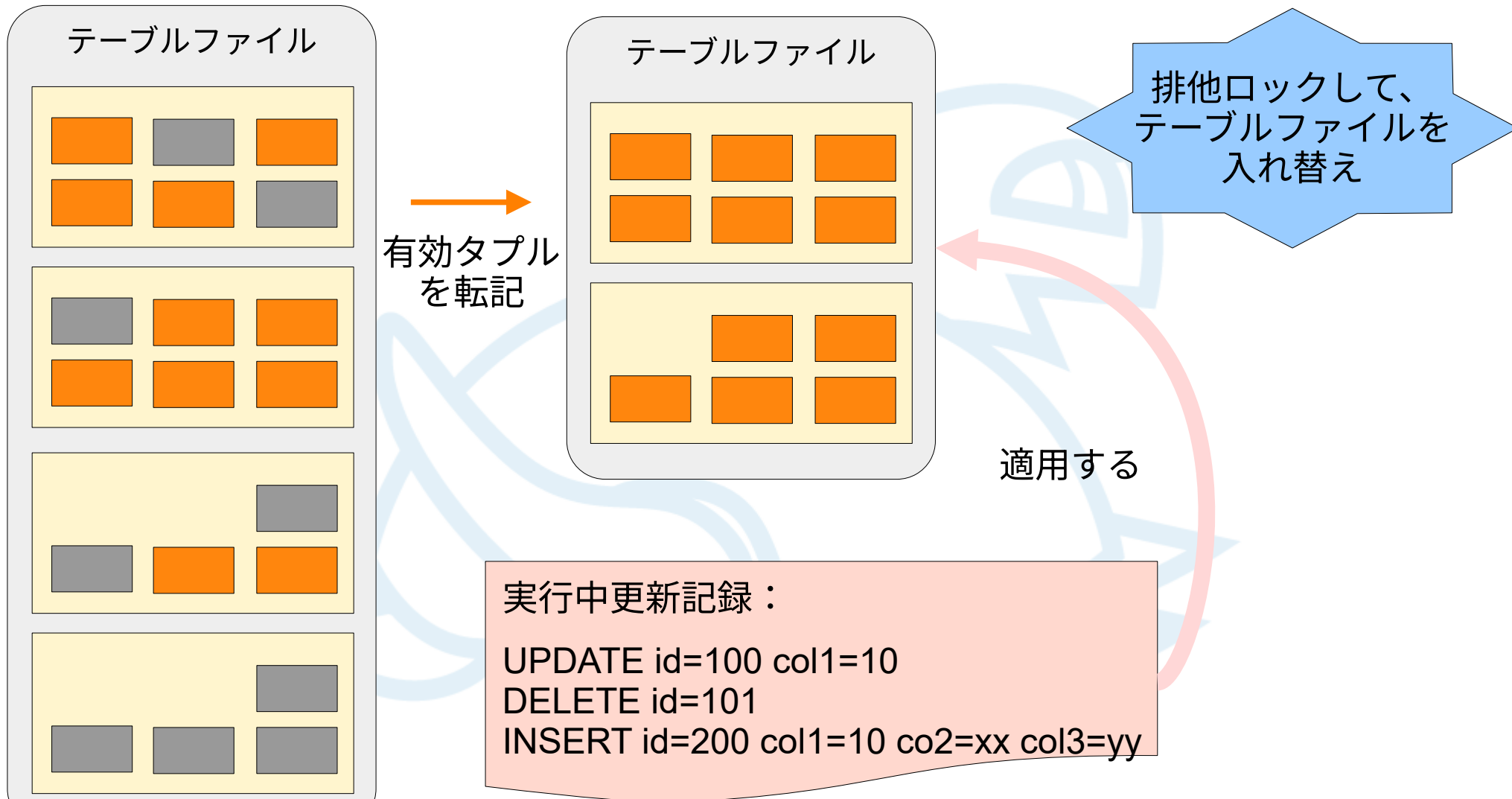
Phase	Description
initializing	The command is preparing to begin scanning the heap. This phase is expected to be very brief.
seq scanning heap	The command is currently scanning the table using a sequential scan.
index scanning heap	REPACK is currently scanning the table using an index scan.
sorting tuples	REPACK is currently sorting tuples.
writing new heap	REPACK is currently writing the new heap.
catch-up	REPACK CONCURRENTLY is currently processing the DML commands that other transactions executed during any of the preceding phases.
swapping relation files	The command is currently swapping newly-built files into place.
rebuilding index	The command is currently rebuilding an index.
performing final cleanup	The command is performing final cleanup. When this phase is completed, REPACK will end.

USING
INDEX
かつ
Seq Scan
の場合

CONCURR
ENTLY で
は、ここで
排他ロック

インデック
ス再構築も
行なわれる

CONCURRENTLY 動作イメージ



データベース全体を REPACK

```
db1=# REPACK;  
REPACK
```

CONCURRENTLY 指定には
個別テーブル指定が必要

→ データベース内の全テーブルを REPACK
(システムテーブルも対象)

```
db1=# ALTER TABLE tb101 CLUSTER ON tb101_pkey ;  
ALTER TABLE
```

→ REPACK USING INDEX コマンドで使うインデックスを指定

```
db1=# REPACK USING INDEX;  
REPACK
```

→ 前項コマンドで使用するインデックスが指定されたテーブルのみ、
並び替え付きの REPACK 実行 (CLUSTER コマンド相当)

```
db1=# ALTER TABLE tb101 SET WITHOUT CLUSTER;  
ALTER TABLE
```

→ REPACK USING INDEX コマンドで使うインデックス指定を解除

REPACK CONCURRENTLY 性能影響

- pgbench と共に実行してみる

```
$ pgbench -i -s 10 -F 90 db1
$ pgbench -c 40 -T 60 -n db1 &> pgbench.out &
$ (sleep 10 ; repack_concurrently.sh ; \
    : 3 テーブルを REPACK するスクリプトを 4 回実行
    sleep 10 ; repack_concurrently.sh ;) &> repack.out &
```

- 通常 : 2177 tps ⇒ 裏で REPACK 実行 : 1900 tps
- log_min_duration_statement = 1s に該当は ?
 - REPACK コマンド自体のみ

```
LOG:  duration: 1470.640 ms  statement: REPACK (CONCURRENTLY) pgbench_accounts ;
LOG:  duration: 1399.822 ms  statement: REPACK (CONCURRENTLY) pgbench_accounts ;
LOG:  duration: 2084.614 ms  statement: REPACK (CONCURRENTLY) pgbench_accounts ;
LOG:  duration: 1818.306 ms  statement: REPACK (CONCURRENTLY) pgbench_accounts ;
```

REPACK (CONCURRENTLY) の注意点

- REPACK の注意点

- 注意点は VACUUM FULL/CLUSTER と同じ
 - 実行中に対象テーブル用に 2 倍サイズ容量が必要
 - テーブルサイズ程度の WAL 発生 / レプリケーション負担
 - ソート用に一時領域を使うことがある

- REPACK (CONCURRENTLY) の注意点

- 主キーがないと使えない
- UNLOGGED テーブルには使えない
- スロットを使う (スロット数上限に注意)
- 実行中の更新記録に一時領域を使う

俯瞰的な話

- PostgreSQL ネイティブのデータ格納方式**以外**が使われることも増えてきた
 - Aurora, Alloy, Neon, YugabyteDB, 各種 Lake 系
- ネイティブテーブルファイル (heap) に対する新機能の需要はある？
 - ⇒ VACUUM を必要とするタイプの互換ソフトウェアでは、ネイティブ PostgreSQL 同様に有用
 - ⇒ ネイティブ PostgreSQL もまだまだ使われている

まとめ

- PostgreSQL 19 から追加された REPACK
- VACUUM FULL 、 CLUSTER コマンドを統合
- CONCURRENTLY で低負荷なオンライン実行が可能

VACUUM の呪いはそろそろ終わりにしましょう！

PostgreSQL 19 について

- 今回は Revert が多かった：
 - グラフ問い合わせ、テンポラルテーブル、
バイナリ pg_dumpall、パーティション分割・統合、
オンラインでのチェックサム有効 / 無効切り替え、
GROUP BY ALL
- その他の主な新機能
 - プランナヒント機能
 - 論理レプリケーションでシーケンスに対応
 - 外部キー制約の検査が高速化
 - ストリーミングレプリケーションむけ WAIT FOR コマンド

PostgreSQL 19 について

リリース予定

- PostgreSQL 19 RC1 ("Release Candidate 1") is scheduled for Thursday, Oct 15, 2026.

Unless there are significant issues found during the release candidate period, the GA release is scheduled for Thursday, **Oct 29, 2026**.

- AI によるバグ検出の効率化でバグ報告が増えている
 - リリース済みバージョン系列での脆弱性報告も急増

日本 PostgreSQL ユーザ会 (JPUG) とは

- PostgreSQL の普及を促進し、技術情報を公開し、ユーザ相互の情報交換を支援する
 - メーリングリストや Slack スペースの運営
postgresql-jp
 - 勉強会、カンファレンスの主催や開催支援
 - マニュアル翻訳
 - PostgreSQL グローバル開発グループ（PGDG）の日本における advocacy 代表窓口

PostgreSQL カンファレンス 2026

日時：11 月 27 日（金）

会場：「AP 東京八重洲」
（東京駅から徒歩 6 分）

- 10 月 26 日まで EventRegist
で早割半額チケット発売中
（10 月 27 日～当日までは通常価格チケット）

