Batching in Executor

Amit Langote

The Problem

Storage / TAM already batches: 8KB pages hold ~40-100 tuples. Executor fetches and returns one tuple at a time.

Per-tuple (current)

```
ExecProcNode()
  ExecSeqScan()
    ExecScanExtended()
      SeqNext()
        table_scan_getnextslot()
          heap_getnextslot()
            heapgettup_pagemode()
```

× 10M times

Per-batch (with patch)

```
ExecProcNode()
  ExecSeqScan()
    ExecScanExtendedBatch()
      SeqNextBatch()
        table_scan_getnextbatch()
          heap_getnextbatch()
            heapgettup_pagemode_batch()
```

× 156K times (for 10M rows)

Result: 22-28% faster (scan without qual), 33-42% with batched qual

TupleBatch

Reducing TAM crossings requires containers on **both sides** — TAM and executor.

Batch fetch without TupleBatch?

HeapBatch has N tuple pointers

But only ONE ss_ScanTupleSlot

→ **N calls to populate the slot**

No reduction in TAM crossings!

With TupleBatch

HeapBatch has N tuple pointers

TupleBatch has N inslots[]

→ **1 call to materialize_all()**

Bulk transfer: N tuples, 1 crossing

TupleBatch is the executor-side container that enables bulk materialization

HeapBatch (TAM) + **TupleBatch** (executor) + **materialize_all** (callback) = actual call reduction

This is why TupleBatch cannot be a separate "optimization" patch — it's essential to the design.

Patch Status

Target for v19

Table AM batch API + heapam: HeapBatch, scan_begin / getnext / end_batch

SeqScan batching + TupleBatch: inslots[], materialize_all, executor_batch_rows GUC

In development (shows potential)

Batched qual evaluation: new EEOPs, ExecQualBatch(), separate interpreter (ExecInterpQualBatch()), WIP.

Future work

ExecProcNodeBatch(): node interface returning batches

Aggregate batching: use ExecProcNodeBatch(), batched agg transitions

Discussion Points

Open Questions

- Batch size tuning: GUC vs adaptive?
- Zero-overhead when batching disabled (executor_batch_rows=0)?
    - Especially with batched expression evaluation
- Batched qual: deform-heavy column handling?

Current Limitations

- Forward scans only
- Pagemode required (SO_ALLOW_PAGEMODE)
- Single-page batches
- No batched projection (forces per-slot output)

pgsql-hackers thread: "Batching in executor"