We’ve shown the performance improvement of zheap over heap in a few different pgbench scenarios. All of these tests were run with data that fits in *shared_buffers (32GB), and 16 transaction slots per zheap page. Scenario-1 and Scenario-2 has used synchronous_commit = off and Scenario-3 and Scenario-4 has used synchronous_commit = on*

Scenario 1: A 15 minutes simple-update pgbench test with scale factor 100 shows 5.13% TPS improvement with 64 clients. The performance improvement increases as we increase the scale factor; at scale factor 1000, it reaches11.5% with 64 clients.

	Scale Factor	HEAP	ZHEAP (tables)*	Improvement
Before test	100	1281 MB	1149 MB	-10.30%
	1000	13 GB	11 GB	-15.38%
After test	100	4.08 GB	3 GB	-26.47%
	1000	15 GB	12.6 GB	-16%

* The size of zheap tables increase because of the insertions in pgbench_history table.

Scenario 2: To show the effect of bloat, we’ve performed another test similar to the previous scenario, but a transaction is kept open for the first 15 minutes of a 30-minute test. This restricts HOT-pruning for the heap and undo-discarding for zheap for the first half of the test. Scale factor 1000 - 75.86% TPS improvement for zheap at 64 client count. Scale factor 3000 - 98.18% TPS improvement for zheap at 64 client count.

	Scale Factor	HEAP	ZHEAP (tables)*	Improvement
After test	1000	19 GB	14 GB	-26.30%
	3000	45 GB	37 GB	-17.70%

* The size of zheap tables increase because of the insertions in pgbench_history table.

The reason for this huge performance improvement is that when the long-running transaction gets committed after 900 seconds, autovacuum workers start working and degrade the performance of heap for a long time. In addition, the heap tables are also bloated by a significant amount. On the other hand, the undo worker discards the undo very quickly, and we don't have any bloat in the zheap relations. In brief, zheap clusters the bloats in undo segments. We just need to determine the how much undo can be discarded and remove it, which is cheap.

Scenario 3: A 15 minutes simple-update pgbench test with scale factor 100 shows 6% TPS improvement with 64 clients. The performance improvement increases as we increase the scale factor to 1000 achieving 11.8% with 64 clients.

	Scale Factor	HEAP	ZHEAP (tables)*	Improvement
Before test	100	1281 MB	1149 MB	-10.30%
	1000	13 GB	11 GB	-15.38%
After test	100	2.88 GB	2.20 GB	-23.61%
	1000	13.9 GB	11.7 GB	-15.80%

* The size of zheap tables increase because of the insertions in pgbench_history table.

Scenario 4: To amplify the effect of bloats in scenario 3, we’ve performed another test similar to scenario, but a transaction is kept open for the first 15 minutes of a 30 minute test. This restricts HOT-pruning for heap and undo-discarding for zheap for the first half of the test.

	Scale Factor	HEAP	ZHEAP (tables)*	Improvement
After test	1000	15.5 GB	12.4 GB	-20%
	3000	40.2 GB	35 GB	-12.90%