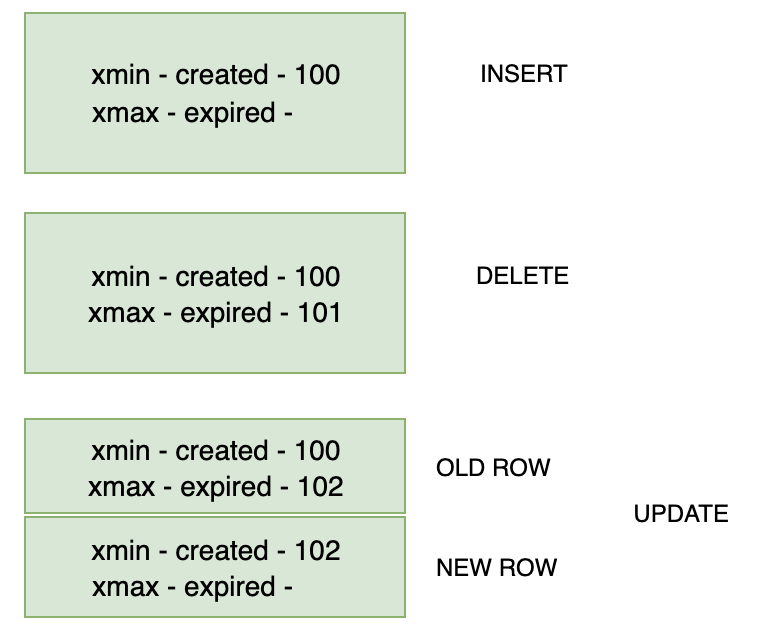
postgres=# CREATE TABLE test(ID int);  
CREATE TABLE  
postgres=# INSERT INTO test VALUES(1);  
INSERT 0 1  
postgres=# SELECT xmin, xmax, id FROM test;  
 xmin | xmax | id  
------+------+----  
 1739 | 0 | 1  
(1 row)



postgres=# select name, setting, context from pg\_settings where category ~ 'Autovacuum';

name | setting | context

-------------------------------------+-----------+------------

autovacuum | on | sighup

autovacuum\_analyze\_scale\_factor | 0.1 | sighup

autovacuum\_analyze\_threshold | 50 | sighup

autovacuum\_freeze\_max\_age | 200000000 | postmaster

autovacuum\_max\_workers | 3 | postmaster

autovacuum\_multixact\_freeze\_max\_age | 400000000 | postmaster

autovacuum\_naptime | 60 | sighup

autovacuum\_vacuum\_cost\_delay | 20 | sighup

autovacuum\_vacuum\_cost\_limit | -1 | sighup

autovacuum\_vacuum\_scale\_factor | 0.2 | sighup

autovacuum\_vacuum\_threshold | 50 | sighup

(11 rows)

SELECT schemaname,

relname,

now() - last\_autovacuum AS "noautovac",

now() - last\_vacuum AS "novac",

n\_tup\_upd,

n\_tup\_del,

pg\_size\_pretty(pg\_total\_relation\_size(schemaname||'.'||relname)),

autovacuum\_count,

last\_autovacuum,

vacuum\_count,

last\_vacuum

FROM pg\_stat\_user\_tables

WHERE (now() - last\_autovacuum > '7 days'::interval

OR now() - last\_vacuum >'7 days'::interval )

OR (last\_autovacuum IS NULL AND last\_vacuum IS NULL )

ORDER BY novac DESC;

SELECT pid,

state,

query

FROM pg\_stat\_activity

WHERE query like '%VACUU%

AND state = 'active';

select pg\_terminate\_backend(pid);

postgres=# set vacuum\_cost\_delay=10;

SET

postgres=# VACUUM VERBOSE pgbench\_branches ;

INFO: vacuuming "public.pgbench\_branches"

INFO: index "pgbench\_branches\_pkey" now contains 10 row versions in 2 pages DETAIL: 0 index row versions were removed.

0 index pages have been deleted, 0 are currently reusable.

CPU 0.00s/0.00u sec elapsed 0.00 sec.

INFO: "pgbench\_branches": found 0 removable, 10 nonremovable row versions in 1 out of DETAIL: 0 dead row versions cannot be removed yet.

There were 0 unused item pointers.

0 pages are entirely empty.

CPU 0.00s/0.00u sec elapsed 0.00 sec.

VACUUM

postgres=#

--set vacuum\_cost\_delay=

VACUUM VERBOSE ANALYZE table1;

VACUUM VERBOSE ANALYZE table2;

VACUUM VERBOSE ANALYZE table3;

etc...

Then, it can be run using psql from any host:

psql -h <server\_ip> -f vacuum.sql -U <user> -d <db> >> /log/vacuum.log

This could be done more dynamically. Create a sql file using below command (for example, gen\_vacuum\_list.sql)

SELECT 'VACUUM VERBOSE ANALYZE ' || schemaname || '.' || relname ||';'

FROM pg\_stat\_user\_tables

WHERE (now() - last\_autovacuum > '7 days'::interval

OR now() - last\_vacuum >'7 days'::interval )

OR (last\_autovacuum IS NULL AND last\_vacuum IS NULL );

Dump the output to a file, and execute the result against the database:

psql -h <server\_ip> -f gen\_vacuum\_list.sql -U <user> -d <db> >> /tmp/vacuum\_tables.sql

psql -h <server\_ip> -f /tmp/vacuum\_tables.sql -U <user> -d <db> >> /log/vacuum.log

In crontab:

\* \* \* \* \* /usr/bin/pgrep -f ’postgres: autovacuum’ | xargs --no-run-if-empty -I $ renice -n 20 -p $ >/dev/null 2>/dev/null

\* \* \* \* \* /usr/bin/pgrep -f ’postgres: autovacuum’ | xargs --no-run-if-empty -I $ ionice -c 3 -t -p $

In postgresql.conf:

autovacuum\_max\_workers = 20

autovacuum\_vacuum\_cost\_delay = 10

postgres=# create table index\_test (id int);

CREATE TABLE

postgres=# insert into index\_test values (generate\_series(1,10000000));

INSERT 0 10000000

postgres=# create index fragmented\_index on index\_test (id);

CREATE INDEX

postgres=#

postgres=# create extension pgstattuple ;

CREATE EXTENSION

postgres=# insert into index\_test values (generate\_series(1,10000000));

INSERT 0 10000000

postgres=# update index\_test set id = 1 where id < 5000000;

UPDATE 4999999

postgres=# SELECT \* FROM pgstatindex('fragmented\_index');

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

version | 2

tree\_level | 2

index\_size | 366919680

root\_block\_no | 412

internal\_pages | 177

leaf\_pages | 44612

empty\_pages | 0

deleted\_pages | 0

avg\_leaf\_density | 82.79

leaf\_fragmentation | 35.65

postgres=# reindex index fragmented\_index ;

REINDEX

Time: 7960.823 ms (00:07.961)

postgres=# SELECT \* FROM pgstatindex('fragmented\_index');

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

version | 2

tree\_level | 2

index\_size | 224641024

root\_block\_no | 290

internal\_pages | 98

leaf\_pages | 27323

empty\_pages | 0

deleted\_pages | 0

avg\_leaf\_density | 90.09

leaf\_fragmentation | 0

SELECT a.indexrelname, b.\*  
FROM pg\_stat\_user\_indexes a,  
LATERAL pgstatindex(indexrelname) b  
ORDER BY leaf\_fragmentation DESC;

SELECT (x.a).indexrelname, (x.b).\*  
FROM (SELECT a, pgstatindex(a.indexrelname) AS b  
 FROM pg\_stat\_user\_indexes a) x  
ORDER BY leaf\_fragmentation DESC;

SELECT relname AS table, indexrelname AS index, pg\_size\_pretty(pg\_relation\_size(indexrelid)) AS size  
FROM pg\_stat\_user\_indexes  
WHERE idx\_scan = 0  
ORDER BY pg\_relation\_size(indexrelid) DESC;

SELECT indrelid::regclass AS table, indkey AS column\_numbers, array\_agg(indexrelid::regclass) AS indexes, pg\_catalog.pg\_get\_expr(indpred, indrelid, true) AS expression  
FROM pg\_index  
GROUP BY indrelid, indkey, pg\_catalog.pg\_get\_expr(indpred, indrelid, true)  
HAVING count(\*) > 1;