Габдулханов Марсель РИМ-181228

Лабораторные работы по дисциплине «Алгоритмы обработки данных во внешней памяти»

ЛР №: 1,2,3,3,4,4,6

1 Compiling sources

```
gabmars@gabmars:~$ sudo apt-get install git
[sudo] password for gabmars:
Sorry, try again.
[sudo] password for gabmars:
Sorry, try again.
[sudo] password for gabmars:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-arch git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 99 not upgraded.
Need to get 3 932 kB of archives.
After this operation, 25,6 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ru.archive.ubuntu.com/ubuntu xenial/main amd64 liberror-perl all 0.17-1.2 [19,6 kB]
Get:2 http://ru.archive.ubuntu.com/ubuntu xenial-updates/main amd64 git-man all 1:2.7.4-0ubuntu1.6 [736 kB]
Get:3 http://ru.archive.ubuntu.com/ubuntu xenial-updates/main amd64 git amd64 1:2.7.4-0ubuntu1.6 [3 176 kB]
Fetched 3 932 kB in 0s (4 426 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 177102 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17-1.2_all.deb ...
Unpacking liberror-perl (0.17-1.2) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.7.4-0ubuntu1.6_all.deb ...
Unpacking git-man (1:2.7.4-0ubuntu1.6) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.7.4-0ubuntu1.6_amd64.deb ...
Unpacking git (1:2.7.4-0ubuntu1.6) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up liberror-perl (0.17-1.2) ...
Setting up git-man (1:2.7.4-0ubuntu1.6) ...
Setting up git (1:2.7.4-0ubuntu1.6) ...
gabmars@gabmars:~$ mkdir postgres
gabmars@gabmars:~$ cd postgres/
gabmars@gabmars:~/postgres$ git clone https://github.com/gabmars/postgres
Cloning into 'postgres'...
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 733625 (delta 0), reused 0 (delta 0), pack-reused 733624
Receiving objects: 100% (733625/733625), 427.19 MiB | 2.67 MiB/s, done.
Resolving deltas: 100% (603805/603805), done.
Checking connectivity... done.
Checking out files: 100% (5568/5568), done.
```

```
gabmars@gabmars:~/postgres$ ls
postgres
gabmars@gabmars:~/postgres$ cd postgres/
gabmars@gabmars:~/postgres/postgres$ ls
aclocal.m4  config  configure  configure.in  contrib  COPYRIGHT  doc  GNUmakefile.in  HISTORY  Makefile  README  README.git  src
gabmars@gabmars:~/postgres/postgres$ git checkout REL9_6_STABLE
Checking out files: 100% (4253/4253), done.
Branch REL9_6_STABLE set up to track remote branch REL9_6_STABLE from origin.
Switched to a new branch 'REL9_6_STABLE'
gabmars@gabmars:~/postgres/postgres$ git checkout -b lab1
Switched to a new branch 'lab1'
gabmars@gabmars:~/postgres/postgres$ git push origin -u lab1
Username for 'https://github.com': gabmars
Password for 'https://gabmars@github.com':
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'lab1' on GitHub by visiting:
remote:      https://github.com/gabmars/postgres/pull/new/lab1
remote:
To https://github.com/gabmars/postgres
 * [new branch]      lab1 -> lab1
Branch lab1 set up to track remote branch lab1 from origin.
```

```
gabmars@gabmars:~/postgres/postgres$ ls
aclocal.m4  config  config.log  config.status  configure  configure.in  contrib  COPYRIGHT  doc
gabmars@gabmars:~/postgres/postgres$ cd ..
gabmars@gabmars:~/postgres$ ls
bin  include  lib  postgres  share
gabmars@gabmars:~/postgres$ ls bin
clusterdb    createuser  dropuser  pg_archivecleanup  pg_config       pg_dump     pg_receivexlog
createdb     dropdb      ecpg      pg_basebackup      pg_controldata  pg_dumpall  pg_recvlogical
createlang   droplang    initdb    pgbench            pg_ctl          pg_isready  pg_resetxlog
gabmars@gabmars:~/postgres$ export PATH=$HOME/postgres/bin:$PATH
gabmars@gabmars:~/postgres$ export PGDATA=$HOME/postgres/DemoDb
gabmars@gabmars:~/postgres$ initdb
The files belonging to this database system will be owned by user "gabmars".
This user must also own the server process.

The database cluster will be initialized with locales
  COLLATE:  en_US.UTF-8
  CTYPE:    en_US.UTF-8
  MESSAGES: en_US.UTF-8
  MONETARY: ru_RU.UTF-8
  NUMERIC:  ru_RU.UTF-8
  TIME:     ru_RU.UTF-8
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

creating directory /home/gabmars/postgres/DemoDb ... ok
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting dynamic shared memory implementation ... posix
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

    pg_ctl -D /home/gabmars/postgres/DemoDb -l logfile start

gabmars@gabmars:~/postgres$ cd DemoDb/
gabmars@gabmars:~/postgres/DemoDb$ nano pG_VERSIO
gabmars@gabmars:~/postgres/DemoDb$ cat PG_VERSIO
cat: PG_VERSIO: No such file or directory
gabmars@gabmars:~/postgres/DemoDb$ nano PG_VERSIO
gabmars@gabmars:~/postgres/DemoDb$ ls
base    pg_clog      pg_dynshmem  pg_ident.conf  pg_multixact  pg_replslot  pg_snapshots  pg_sta
global  pg_commit_ts  pg_hba.conf  pg_logical     pg_notify     pg_serial    pg_stat       pg_sub
gabmars@gabmars:~/postgres/DemoDb$ cat PG_VERSION
9.6
gabmars@gabmars:~/postgres/DemoDb$ cd ..
gabmars@gabmars:~/postgres$ cd bin
gabmars@gabmars:~/postgres/bin$ ./postgres ~/postgres/DemoDb
postgres: invalid argument: "/home/gabmars/postgres/DemoDb"
Try "postgres --help" for more information.
gabmars@gabmars:~/postgres/bin$ ./postgres -D ~/postgres/DemoDb
LOG:  database system was shut down at 2019-05-11 20:05:50 +05
LOG:  MultiXact member wraparound protections are now enabled
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
FATAL:  database "gabmars" does not exist
ERROR:  relation "information_schems.tables" does not exist at character 22
STATEMENT:  select count(*) from information_schems.tables;
ERROR:  relation "information_schemas.tables" does not exist at character 22
```

```
gabmars@gabmars:~$ psql postgres
psql (9.6.13)
Type "help" for help.

postgres=# select 1
postgres-#
postgres-# ;
 ?column?
----------
        1
(1 row)

postgres=# select count(*) from information_schems.tables;
ERROR:  relation "information_schems.tables" does not exist
LINE 1: select count(*) from information_schems.tables;
                             ^
postgres=# select count(*) from information_schemas.tables;
ERROR:  relation "information_schemas.tables" does not exist
LINE 1: select count(*) from information_schemas.tables;
                             ^
postgres=# select count(*) from information_schema.tables;
 count
-------
   177
(1 row)

postgres=# \q
```


=====================
All 167 tests passed.
=====================
make[1]: Leaving directory '/home/gabmars/postgres/postgres/src/test/regress'
gabmars@gabmars:~/postgres/postgres$

2 Patching source code

```
gabmars@gabmars:~/postgres/postgres$ rm -rf ~/postgres/DemoDb
gabmars@gabmars:~/postgres/postgres$ initdb ~/postgres/DemoDb
The files belonging to this database system will be owned by user "gabmars".
This user must also own the server process.

The database cluster will be initialized with locales
  COLLATE:  en_US.UTF-8
  CTYPE:    en_US.UTF-8
  MESSAGES: en_US.UTF-8
  MONETARY: ru_RU.UTF-8
  NUMERIC:  ru_RU.UTF-8
  TIME:     ru_RU.UTF-8
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

creating directory /home/gabmars/postgres/DemoDb ... ok
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting dynamic shared memory implementation ... posix
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

    pg_ctl -D /home/gabmars/postgres/DemoDb -l logfile start

gabmars@gabmars:~/postgres/postgres$ postgres -D ~/postgres/DemoDb/
LOG:  database system was shut down at 2019-05-11 21:38:06 +05
LOG:  MultiXact member wraparound protections are now enabled
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
FATAL:  database "gabmars" does not exist
ERROR:  syntax error at or near "from" at character 92
STATEMENT:  create table dataTable
        as
        select x as group_id, y as entry_id, 1::decimal(19,6) as heavy0, from generate_series(1,1e2,1) x, generate_series(1,1e5,1) y;
ERROR:  current transaction is aborted, commands ignored until end of transaction block
STATEMENT:  create table dataTable
        as
        select x as group_id, y as entry_id, 1::decimal(19,6) as heavy0 from generate_series(1,1e2,1) x, generate_series(1,1e5,1) y;
ERROR:  current transaction is aborted, commands ignored until end of transaction block
STATEMENT:  begin transaction;
LOG:  server process (PID 29622) was terminated by signal 11: Segmentation fault
```

```
gabmars@gabmars:~/postgres/postgres$ git checkout .
gabmars@gabmars:~/postgres/postgres$ git clean -f
Removing src/test/regress/expected/index_including.out
Removing src/test/regress/sql/index_including.sql
Removing test.sql
gabmars@gabmars:~/postgres/postgres$ git apply ~/Downloads/including_columns_9.7_v1.patch
error: patch failed: src/backend/optimizer/util/plancat.c:687
error: src/backend/optimizer/util/plancat.c: patch does not apply
/home/gabmars/Downloads/including_columns_9.7_v1.patch:3440: new blank line at EOF.
+
gabmars@gabmars:~/postgres/postgres$ git status
HEAD detached at 70715e6
nothing to commit, working directory clean
gabmars@gabmars:~/postgres/postgres$ git checkout REL9_6_STABLE
Checking out files: 100% (2206/2206), done.
Previous HEAD position was 70715e6... Fix whitespace
Switched to branch 'REL9_6_STABLE'
Your branch is up-to-date with 'origin/REL9_6_STABLE'.
gabmars@gabmars:~/postgres/postgres$ git status
On branch REL9_6_STABLE
Your branch is up-to-date with 'origin/REL9_6_STABLE'.
nothing to commit, working directory clean
gabmars@gabmars:~/postgres/postgres$ git checkout `git rev-list -n 1 --before="2016-08-15" master`
Checking out files: 100% (1533/1533), done.
Note: checking out '05d8dec690e9719ff9a1830f5492864104275b5e'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at 05d8dec... Simplify the process of perltidy'ing our Perl files.
gabmars@gabmars:~/postgres/postgres$ git apply ~/Downloads/including_columns_9.7_v1.patch -3
/home/gabmars/Downloads/including_columns_9.7_v1.patch:3440: new blank line at EOF.
+
warning: 1 line adds whitespace errors.
gabmars@gabmars:~/postgres/postgres$ git status
HEAD detached at 05d8dec
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
```

```
gabmars@gabmars:~/postgres/postgres$ rm -rf ~/postgres/DemoDb/
gabmars@gabmars:~/postgres/postgres$ initdb ~/postgres/DemoDb/
The files belonging to this database system will be owned by user "gabmars".
This user must also own the server process.

The database cluster will be initialized with locales
  COLLATE:  en_US.UTF-8
  CTYPE:    en_US.UTF-8
  MESSAGES: en_US.UTF-8
  MONETARY: ru_RU.UTF-8
  NUMERIC:  ru_RU.UTF-8
  TIME:     ru_RU.UTF-8
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

creating directory /home/gabmars/postgres/DemoDb ... ok
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting dynamic shared memory implementation ... posix
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

WARNING: enabling "trust" authentication for local connections
You can change this by editing pg_hba.conf or using the option -A, or
--auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

    pg_ctl -D /home/gabmars/postgres/DemoDb/ -l logfile start

gabmars@gabmars:~/postgres/postgres$ postgres -D ~/postgres/DemoDb/
LOG:  database system was shut down at 2019-05-11 22:27:37 +05
LOG:  MultiXact member wraparound protections are now enabled
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
^CLOG:  received fast shutdown request
LOG:  aborting any active transactions
LOG:  autovacuum launcher shutting down
LOG:  shutting down
LOG:  database system is shut down
gabmars@gabmars:~/postgres/postgres$
```

```
gabmars@gabmars:~/postgres/postgres$ touch test.sql
gabmars@gabmars:~/postgres/postgres$ vim test.sql
gabmars@gabmars:~/postgres/postgres$ cat test.sql
\timing

begin transaction;

create table dataTable as select x as group_id, y as entry_id, 1::decimal(19,6) as heavy0

create index idx on dataTable(group_id,entry_id) including(heavy0);

select pg_size_pretty(pg_relation_size('idx'));

rollback;
gabmars@gabmars:~/postgres/postgres$ psql postgres<./test.sql
Timing is on.
BEGIN
Time: 0,144 ms
SELECT 10000000
Time: 28509,627 ms
ERROR:  index row requires 158288 bytes, maximum size is 8191
Time: 28121,220 ms
ERROR:  current transaction is aborted, commands ignored until end of transaction block
Time: 0,347 ms
ROLLBACK
Time: 2,259 ms
gabmars@gabmars:~/postgres/postgres$ cd ~/
gabmars@gabmars:~$ touch test.sql
gabmars@gabmars:~$ gedit test.sql
gabmars@gabmars:~$ cat test.sql
\timing

begin transaction;

create table dataTable as select x as group_id, y as entry_id, 1::decimal(19,6) as heavy0

create index idx on dataTable(group_id,entry_id) including(heavy0);

select pg_size_pretty(pg_relation_size('idx'));

rollback;

gabmars@gabmars:~$ cd postgres/postgres/
gabmars@gabmars:~/postgres/postgres$ psql postgres<~/test.sql
Timing is on.
BEGIN
Time: 0,147 ms
SELECT 10000000
Time: 27052,531 ms
CREATE INDEX
Time: 54018,994 ms
 pg_size_pretty
----------------
 378 MB
(1 row)

Time: 0,600 ms
ROLLBACK
Time: 516,940 ms
gabmars@gabmars:~/postgres/postgres$
```

3 Paged memory

```
gabmars@gabmars:~$ cat test.sh
#!/bin/sh
export PGDATA=~/postgres/DemoDb/
rm -rf ~/postgres/DemoDb/
cd ~/postgres/postgres
make
make install
~/postgres/bin/initdb
echo "listen_addresses = '*'">>~/postgres/DemoDb/postgresql.conf
echo "work_mem = 512MB">>~/postgres/DemoDb/postgresql.conf
echo "maintenance_work_mem = 512MB">>~/postgres/DemoDb/postgresql.conf
sed -i '113s/.*/shared_buffers = 2048MB/' ~/postgres/DemoDb/postgresql.conf

~/postgres/bin/pg_ctl -w start

~/postgres/bin/pgbench postgres -i -s 70
~/postgres/bin/pgbench postgres -c 4 -j 2 -T 60

~/postgres/bin/pg_ctl stop
```

```
vacuum...
set primary keys...
done.
starting vacuum...end.
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 70
query mode: simple
number of clients: 4
number of threads: 2
duration: 60 s
number of transactions actually processed: 17015
latency average: 14.105 ms
tps = 282.472374 (including connections establishing)
tps = 282.494607 (excluding connections establishing)
LOG:  received fast shutdown request
LOG:  aborting any active transactions
LOG:  autovacuum launcher shutting down
waiting for server to shut down....FATAL:  terminating connection due to administrator command
LOG:  could not send data to client: Broken pipe
LOG:  shutting down
.............LOG:  database system is shut down
 done
server stopped
```

```
gabmars@gabmars:~/postgres/postgres/src$ grep -r 'BLCKSZ' --include='*.h'
include/access/tuptoaster.h:    MAXALIGN_DOWN((BLCKSZ - \
include/access/slru.h: * Define SLRU segment size.  A page is the same BLCKSZ as is used everywhere
include/access/spgist_private.h:        MAXALIGN_DOWN(BLCKSZ - \
include/access/xlog_internal.h:         ((xlrp) % XLOG_BLCKSZ >= SizeOfXLogShortPHD)
include/access/htup_details.h: * header and MAXALIGN alignment padding.  Basically it's BLCKSZ minus the
include/access/htup_details.h:#define MaxHeapTupleSize  (BLCKSZ - MAXALIGN(SizeOfPageHeaderData + sizeof(ItemIdData)))
include/access/htup_details.h:  ((int) ((BLCKSZ - SizeOfPageHeaderData) / \
include/access/nbtree.h:         * BLCKSZ, so it can hold as much as a full page's worth of tuples.
include/access/gin_private.h:           MAXALIGN_DOWN(((BLCKSZ - \
include/access/gin_private.h:   (BLCKSZ - MAXALIGN(SizeOfPageHeaderData) \
include/access/gin_private.h:   ( BLCKSZ - SizeOfPageHeaderData - MAXALIGN(sizeof(GinPageOpaqueData)) )
include/access/brin_page.h:     (BLCKSZ - MAXALIGN(SizeOfPageHeaderData) - \
include/access/xlogreader.h:     * shall return the number of bytes read (never more than XLOG_BLCKSZ), or
include/access/xlogreader.h:     * Buffer for currently read page (XLOG_BLCKSZ bytes, valid up to at least
include/access/gist_private.h:#define PAGE_IS_EMPTY(nbp) (nbp->freespace == BLCKSZ - BUFFER_PAGE_DATA_OFFSET)
include/access/gist_private.h:  GISTSearchHeapItem pageData[BLCKSZ / sizeof(IndexTupleData)];
include/access/gist_private.h:  ( BLCKSZ - SizeOfPageHeaderData - MAXALIGN(sizeof(GISTPageOpaqueData)) )
include/access/xlogrecord.h: * present is BLCKSZ - the length of "hole" bytes.
include/access/xlogrecord.h: * from BLCKSZ, basically it needs to be stored as an extra information.
include/access/xlogrecord.h: * BLCKSZ - the length of "hole" bytes - the length of extra information.
include/access/itup.h:  ((int) ((BLCKSZ - SizeOfPageHeaderData) / \
include/pg_config_manual.h: * pg_proc's index; BLCKSZ larger than 8K would allow more).  Values larger
include/pg_config.h:    BLCKSZ must be a power of 2. The maximum possible value of BLCKSZ is
include/pg_config.h:    Changing BLCKSZ requires an initdb. */
include/pg_config.h:#define BLCKSZ 8192
include/pg_config.h:    the maximum size of a single file is RELSEG_SIZE * BLCKSZ; relations bigger
include/pg_config.h:    than that are divided into multiple files. RELSEG_SIZE * BLCKSZ must be
include/pg_config.h:/* Size of a WAL file block. This need have no particular relation to BLCKSZ.
include/pg_config.h:    XLOG_BLCKSZ must be a power of 2, and if your system supports O_DIRECT I/O,
include/pg_config.h:    XLOG_BLCKSZ must be a multiple of the alignment requirement for direct-I/O
include/pg_config.h:    buffers, else direct I/O may fail. Changing XLOG_BLCKSZ requires an initdb.
include/pg_config.h:#define XLOG_BLCKSZ 8192
include/pg_config.h:    and larger than XLOG_BLCKSZ (preferably, a great deal larger than
include/pg_config.h:    XLOG_BLCKSZ). Changing XLOG_SEG_SIZE requires an initdb. */
include/storage/fsm_internals.h:#define NodesPerPage (BLCKSZ - MAXALIGN(SizeOfPageHeaderData) - \
include/storage/fsm_internals.h:#define NonLeafNodesPerPage (BLCKSZ / 2 - 1)
include/storage/bufmgr.h:               (Block) (BufferBlocks + ((Size) ((buffer) - 1)) * BLCKSZ) \
include/storage/bufmgr.h:       (Size)BLCKSZ \
include/storage/bufpage.h:#define PageSizeIsValid(pageSize) ((pageSize) == BLCKSZ)
include/storage/bufpage.h:      Assert(((PageHeader) (page))->pd_special <= BLCKSZ);
include/storage/large_object.h: * We could set this as high as BLCKSZ less some overhead, but it seems
include/storage/large_object.h:#define LOBLKSIZE               (BLCKSZ / 4)
include/storage/checksum_impl.h:        checksum = pg_checksum_block(page, BLCKSZ);
include/storage/off.h:#define MaxOffsetNumber                 ((OffsetNumber) (BLCKSZ / sizeof(ItemIdData)))
include/utils/rel.h:    (BLCKSZ * RelationGetFillFactor(relation, defaultff) / 100)
include/utils/rel.h:    (BLCKSZ * (100 - RelationGetFillFactor(relation, defaultff)) / 100)
```

```
/* Size of a disk block --- this also limits the size of a tuple. You can set
   it bigger if you need bigger tuples (although TOAST should reduce the need
   to have large tuples, since fields can be spread across multiple tuples).
   BLCKSZ must be a power of 2. The maximum possible value of BLCKSZ is
   currently 2^15 (32768). This is determined by the 15-bit widths of the
   lp_off and lp_len fields in ItemIdData (see include/storage/itemid.h).
   Changing BLCKSZ requires an initdb. */
#define BLCKSZ 8192
```

```
7000000 of 7000000 tuples (100%) done (elapsed 33.61 s, remaining 0.00 s)
vacuum...
set primary keys...
done.
starting vacuum...end.
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 70
query mode: simple
number of clients: 4
number of threads: 2
duration: 60 s
number of transactions actually processed: 18066
latency average = 13.305 ms
tps = 300.639656 (including connections establishing)
tps = 300.662387 (excluding connections establishing)
LOG:  received fast shutdown request
LOG:  aborting any active transactions
LOG:  autovacuum launcher shutting down
waiting for server to shut down....LOG:  shutting down
.............LOG:  database system is shut down
 done
server stopped
```

3 Query plan analisys

```
gabmars@gabmars:~$ cat tenk.sql
CREATE TABLE tenk1 (
        unique1         int4,
        unique2         int4,
        two                     int4,
        four            int4,
        ten                     int4,
        twenty          int4,
        hundred         int4,
        thousand        int4,
        twothousand     int4,
        fivethous       int4,
        tenthous        int4,
        odd                     int4,
        even            int4,
        stringu1        name,
        stringu2        name,
        string4         name
);


COPY tenk1 FROM '/home/gabmars/postgres/postgres/src/test/regress/data/tenk.data';
create table tenk2 as select * from tenk1;
```

```
gabmars@gabmars:~$ psql postgres<tenk.sql
CREATE TABLE
COPY 10000
SELECT 10000
```

```
gabmars@gabmars:~$ psql postgres
psql (9.6beta4)
Type "help" for help.

postgres=# EXPLAIN ANALYZE SELECT *
postgres-# FROM tenk1 t1, tenk2 t2
postgres-# WHERE t1.unique1 < 100 AND t1.unique2 = t2.unique2 ORDER BY t1.fivethous;
                                                                                                                  QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------------------------------------
 Sort  (cost=98197.89..98644.23 rows=178538 width=488) (actual time=114.290..114.304 rows=100 loops=1)
   Sort Key: t1.fivethous
   Sort Method: quicksort  Memory: 77kB
   ->  Merge Join  (cost=1815.81..4511.13 rows=178538 width=488) (actual time=84.911..114.001 rows=100 loops=1)
         Merge Cond: (t1.unique2 = t2.unique2)
         ->  Sort  (cost=677.10..685.73 rows=3450 width=244) (actual time=5.305..5.320 rows=100 loops=1)
               Sort Key: t1.unique2
               Sort Method: quicksort  Memory: 77kB
               ->  Seq Scan on tenk1 t1  (cost=0.00..474.38 rows=3450 width=244) (actual time=0.023..5.198 rows=100 loops=1)
                     Filter: (unique1 < 100)
                     Rows Removed by Filter: 9900
         ->  Sort  (cost=1138.71..1164.58 rows=10350 width=244) (actual time=79.582..85.090 rows=10000 loops=1)
               Sort Key: t2.unique2
               Sort Method: external sort  Disk: 2520kB
               ->  Seq Scan on tenk2 t2  (cost=0.00..448.50 rows=10350 width=244) (actual time=0.012..22.001 rows=10000 loops=1)
 Planning time: 0.781 ms
 Execution time: 122.463 ms
(17 rows)
```

4 Btree

```
gabmars@gabmars:~/postgres/postgres$ psql postgres -c "create table sample(id text, value text); create index idx on sample(id);
_series(1,1e5) x;"
INSERT 0 100000
gabmars@gabmars:~/postgres/postgres$
```

```
/*
 * Binary search to find the first key on the page >= scan key, or first
 * key > scankey when nextkey is true.
 *
 * For nextkey=false (cmpval=1), the loop invariant is: all slots before
 * 'low' are < scan key, all slots at or after 'high' are >= scan key.
 *
 * For nextkey=true (cmpval=0), the loop invariant is: all slots before
 * 'low' are <= scan key, all slots at or after 'high' are > scan key.
 *
 * We can fall out when high == low.
 */
high++;                                         /* establish the loop invariant for high */

cmpval = nextkey ? 0 : 1;       /* select comparison value */

while (high > low)
{
        OffsetNumber mid = low + ((high - low) / 2);
        static int compareCalls = 0;

        /* We have low <= mid < high, so mid points at a real slot */

        compareCalls++;
        elog(NOTICE, "_bt_compare cal %d", compareCalls);
        result = _bt_compare(rel, keysz, scankey, page, mid);

        if (result >= cmpval)
                low = mid + 1;
        else
                high = mid;
}
```

```
NOTICE:  _bt_compare cal 2801
   id     |    value
---------+-----------
 key7777 | value7777
(1 row)

gabmars@gabmars:~/postgres/postgres$
```

```
NOTICE:  _bt_compare cal 323
   id     |    value
---------+-----------
 key7777 | value7777
(1 row)

gabmars@gabmars:~/postgres/postgres$
```

4 Streaming replication

```
                if (!RecoveryInProgress())
                        ereport(FATAL,
                                        (errmsg("cannot continue WAL streaming, recovery has already ended")));

                /* Process any requests or signals received recently */
                ProcessWalRcvInterrupts();

                if (got_SIGHUP)
                {
                        got_SIGHUP = false;
                        ProcessConfigFile(PGC_SIGHUP);
                        XLogWalRcvSendHSFeedback(true);
                }

                /* See if we can read data immediately */
                len = walrcv_receive(&buf, &wait_fd);
                elog(LOG,"_data_len %d",len);
                if (len != 0)
                {
```

```
gabmars@gabmars:~$ psql postgres
psql (9.6.13)
Type "help" for help.

postgres=# create role repl with replication password 'password' login;
CREATE ROLE
```

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     trust
# IPv4 local connections:
host    all             all             127.0.0.1/32            trust
host    replication     repl            127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local   replication     gabmars                                 trust
#host    replication     gabmars         127.0.0.1/32            trust
```

```
s@gabmars: ~/postgres/DemoDb                                En  00:01
# WRITE AHEAD LOG
#------------------------------------------------------------------------------

# - Settings -

wal_level = replica                     # minimal, replica, or logical
                                        # (change requires restart)
```

```
s@gabmars: ~/postgres/DemoDb                                En  00:03
# Set these on the master and on any standby that will send replication data.

#max_wal_senders = 0            # max number of walsender processes
                                # (change requires restart)
wal_keep_segments = 50          # in logfile segments, 16MB each; 0 disables
#wal_sender_timeout = 60s       # in milliseconds; 0 disables
```

```
s@gabmars: ~/postgres/DemoDb                                En  00:03
# Set these on the master and on any standby that will send replication data.

max_wal_senders = 4             # max number of walsender processes
                                # (change requires restart)
wal_keep_segments = 50          # in logfile segments, 16MB each; 0 disables
#wal_sender_timeout = 60s       # in milliseconds; 0 disables
```

```
gabmars@gabmars:~/postgres/bin$ ./pg_basebackup --xlog-method=stream -D ~/postgr
es/DemoDbSlave/ -U repl -h 127.0.0.1
Password:
gabmars@gabmars:~/postgres/bin$
```

```
port = 5433                             # (change requires restart)
```

```
hot_standby = on                        # "on" allows queries during recovery
```

```
gabmars@gabmars:~/postgres/share$ cp recovery.conf.sample ~/postgres/DemoDbSlave
/recovery.conf
```

```
standby_mode = on
```

```
primary_conninfo = 'host=localhost port=5432 user=repl password=passwrod'
```

```
gabmars@gabmars:~/postgres/bin$ ./psql postgres -c "create table test(x int);"
CREATE TABLE
gabmars@gabmars:~/postgres/bin$
```

```
gabmars@gabmars:~/postgres/bin$ ./psql postgres -c "create table test(x int);"
CREATE TABLE
gabmars@gabmars:~/postgres/bin$ ./psql postgres -c "create table test1(x int);"
CREATE TABLE
gabmars@gabmars:~/postgres/bin$
```

```
gabmars@gabmars:~/postgres/bin$ ./psql postgres -p 5433 -c "create table test1(
 int);"
ERROR:  cannot execute CREATE TABLE in a read-only transaction
gabmars@gabmars:~/postgres/bin$
```

```
gabmars@gabmars:~/postgres/bin$ ./psql postgres -p 5433 -c "select * from test;"
 x
---
(0 rows)

gabmars@gabmars:~/postgres/bin$
```

6 GiST penalty function

```
gabmars@gabmars:~$ postgres/bin/psql postgres < gisttest.sql
Timing is on.
CREATE EXTENSION
Time: 86,129 ms
BEGIN
Time: 0,154 ms
 setseed
---------

(1 row)

Time: 0,526 ms
CREATE TABLE
Time: 2,071 ms
CREATE INDEX
Time: 1,143 ms
INSERT 0 300000
Time: 171676,186 ms
 pg_size_pretty
----------------
 22 MB
(1 row)

Time: 0,517 ms
ROLLBACK
Time: 18,352 ms
```

```
gabmars@gabmars:~$ postgres/bin/psql postgres < gisttest.sql
Timing is on.
CREATE EXTENSION
Time: 83,444 ms
BEGIN
Time: 0,236 ms
 setseed
---------

(1 row)

Time: 0,628 ms
CREATE TABLE
Time: 6,220 ms
CREATE INDEX
Time: 1,114 ms
INSERT 0 300000
Time: 165293,223 ms
 pg_size_pretty
----------------
 22 MB
(1 row)

Time: 0,517 ms
ROLLBACK
Time: 22,737 ms
```

```
gabmars@gabmars:~$ postgres/bin/psql postgres < gisttest.sql
Timing is on.
CREATE EXTENSION
Time: 90,989 ms
BEGIN
Time: 0,259 ms
 setseed
---------

(1 row)

Time: 0,518 ms
CREATE TABLE
Time: 1,940 ms
CREATE INDEX
Time: 1,074 ms
INSERT 0 300000
Time: 166388,461 ms
 pg_size_pretty
----------------
 20 MB
(1 row)

Time: 0,538 ms
ROLLBACK
Time: 13,849 ms
```