

データベース概説 第8回

宇陀則彦

1 トランザクションの有無

まず、実験用のディレクトリ benchmark を作成し、そこに移動する。

```
dragon:~$ mkdir benchmark
dragon:~$ cd benchmark
```

次のプログラム benchmark1.rb を入力し、実行しなさい。(実行は ruby benchmark1.rb とする)

```
benchmark1.rb

#!/usr/bin/ruby -Ku

require 'sqlite3'
require 'benchmark'

open("|rm -f test1.db test2.db")

db1 = SQLite3::Database.new('test1.db')
db2 = SQLite3::Database.new('test2.db')
sql = <<SQL
create table test(
  name varchar(255),
  mail varchar(255)
);
SQL

db1.execute(sql)
db2.execute(sql)

sql = "insert into test values('hoge', 'hoge@hoge.com')"
```

```
Benchmark.bm { |x|
  x.report('no transaction') { # 明示的な transaction 指定なし
    500.times {db1.execute(sql)} # 500回のトランザクション発生
  }
  x.report('transaction ') {
    db2.transaction { # 明示的に transaction を指定
      500.times {db2.execute(sql)} # トランザクションは1回
    }
  }
}
```

このプログラムは二つのデータベース test1.db と test2.db にそれぞれ、insert 文を 500 回実行し、どのくらいの時間がかかるかを表示するプログラムである。sqlite3 では明示的に transaction を指定しないと、自動的にトランザクションが宣言される。したがって、test1.db への insert 文は insert 文ごとにトランザクションが発生する。test2.db へのトランザクションは 1 回である。トランザクションは同時実行制御において処理の一貫性を保つためのものであるが、その一方でパフォーマンスにも大きな影響を与えるので、データベースプログラミングの際は注意が必要である。

2 インデックスの有無

次のプログラム benchmark2a.rb を実行しなさい。また、このプログラムを benchmark2b.rb という名前でもコピーし、6 行目の opac1.db を opac2.db に変更し、実行しなさい。

```
benchmark2a.rb
#!/usr/bin/ruby -Ku

require 'sqlite3'
require 'benchmark'

db1 = SQLite3::Database.new('/home/uda/benchmark/opac1.db')

sql = "select title from opac1 where title = 'メディアとしての紙芝居';"

Benchmark.bm { |x|
  x.report('index') {
    1.times {db1.execute(sql) }
  }
}

db1.close
```

```
dragon:~/benchmark$ ruby benchmark2a.rb
dragon:~/benchmark$ ruby benchmark2b.rb
```

opac1.db と opac2.db はともに約 90 万件の書誌データを格納している。opac1.db と opac2.db の違いは、インデックスを作成¹しているかいないかである。データの量やハードウェアの性能にもよるが、インデックスのあるなしでは検索にかなりの差がでる。

実験

8 行目の select 文を部分文字列の問い合わせに変更し、2 つのプログラムの実行結果を比較しなさい。

```
select title from opac1 where title like 'メディア%';
```

¹B-Tree インデックスが作成される。インデックスを作成する SQL 文はこの場合、create index idx1 on opac1(title); である。

3 準備（知識情報演習 I を履修していない学生用）

今回の演習は、宇陀が用意した `dragon.slis.tsukuba.ac.jp` というサーバで行う。以下、`dragon` で作業するための操作を説明する。

1. 端末の電源を入れなおし、Linux として起動する。（既に起動している場合は次に進む）
2. 「GNOME 端末」を起動する。
3. 「`ssh -Y dragon.slis.tsukuba.ac.jp`」を実行する。「Are you sure you want to continue connecting (yes/no)?」と聞いてきたら、`yes` とする。（この操作は最初の接続時のみ）
4. パスワードを聞いてくるので、配布した `dragon` 用のパスワードを入力する。
5. `dragon` 上で「`emacs`」と実行し、1 章,2 章のプログラムを入力する。`emacs` は `Meadow` とほぼ同じように操作できるエディタである。

その他、詳しい操作は、<http://klis.tsukuba.ac.jp/klis/index.php?KIRL-I> の「第 5 回（後半第 1 回）」のテキストを参照すること。