Introduction to Berkeley DB

Jui-Nan Lin <jnlin@pixnet.tw>
www.pixnet.net

Agenda

- Introduction to Berkeley DB
- Programming with Berkeley DB
  - Creating Environment/Database
  - Set/Get/Delete Values
  - Searching Values with Cursors
- Secondary Databases
- Transaction
- Replication
- Performance Evaluation
- References

Berkeley DB

- Key-Value Based Database
- Developed by U.C. Berkeley
- Acquired by Oracle in 2006
- Transaction and Replication

Why Berkeley DB in Web 2.0?

- RDBMS
  - Widely used (lots of experienced users)
  - Huge
  - Too powerful
- BerkeleyDB
  - Key-Value Based
  - Fit most Web 2.0 application

Programming in BDB

- Programming Language
  - C, C++, Java
  - PHP, Perl, Python
- Environment
  - Directory of databases
- Database
  - Key
  - Value

Creating An Environment

```
#include <db.h>
DB_ENV *myEnv;
u_int32_t env_flags;

db_env_create(&myEnv, 0); /* return 0 for success */
env_flags = DB_CREATE | DB_INIT_MPOOL;

/* return 0 for success */
myEnv->open(myEnv, "/directory/to/environment",
   env_flags, 0);
```

Creating a Database

```
DB *dbp;
u_int32_t db_flags;

db_create(&dbp, myEnv, 0);
db_flags = DB_CREATE;

dbp->open(dbp, NULL, "dbname.db”, NULL, DB_BTREE,
   db_flags, 0);
```

Set/Get/Delete Values (1)

```
DBT key, value;
int intvalue = 0;
char *charkey = “mykey”;

memset(&value, 0, sizeof(DBT));
memset(&data, 0, sizeof(DBT));

key.data = charkey;
key.size = strlen(charkey) + 1;
value.data = &intvalue;
value.size = sizeof(int);
```

Set/Get/Delete Values (2)

```
/* Put Data into Database */
dbp->put(dbp, NULL, &key, &value, DB_NOOVERWRITE);

/* Get Data from Database */
dbp->get(dbp, NULL, &key, &value, 0);

/* Delete Data from Database */
dbp->get(dbp, NULL, &key, 0);
```

- Very Simple, huh?
- If your value is a C structure, remember to use your own memory.

```
data.data = &my_struct;
data.ulen = sizeof(my_struct);
data.flags = DB_DBT_USERMEM;
```

- If your structure has a pointer member, remember to marshal the data.

Cursors

- Get Key-Value Pairs Larger than Specified Key

```
DBC *cp;
dbp->cursor(dbp, NULL, &cp, 0);

key.data = charkey;
key.size = strlen(charkey) + 1;

cp->get(cp, &key, &value, DB_SET_RANGE);
```

- You can set your own compare function while creating database

```
dbp->set_bt_compare(dbp, compare_func);
```

Secondary Databases

- Key-Value => Value->Key
- No manual write into a secondary database.
  - You can read from a secondary database.
- You must associate secondary databases to a primary database.

Transaction (1)

- Enable Transaction in Environment and Database

```
env_flags = DB_CREATE | DB_INIT_TXN |
  DB_INIT_LOCK | DB_INIT_LOG | DB_INIT_MPOOL;
db_flags = DB_CREATE | DB_AUTO_COMMIT;
```

Transaction (2)

- Use Transactions

```
DB_TXN *txn;
myEnv-> txn_begin(myEnv, NULL, &txn, 0);
dbp->put(dbp, txn, &key, &value, 0);
txn->commit(txn, 0);
```

Replication

- Transaction Support Needed
- Master-Slave Architecture
  - Manual Specified Master, *Voted Master*
- TCP/IP Based communication-layer provided in BDB code

Replication Example (1)

```
unsigned short listen_port;
char *listen_host;
unsigned short other_port;
char *other_host;

MyEnv->repmgr_set_local_site(MyEnv, listen_host,
  listen_port, 0);
MyEnv->rep_set_priority(MyEnv, 100); /* Master */

MyEnv->repmgr_add_remote_site(MyEnv, other_host,
  other_port, NULL, 0); /* Slave */
MyEnv->rep_set_nsites(MyEnv, 2);
```

Replication Example (2)

```
env_flags = DB_CREATE | DB_INIT_MPOOL | DB_INIT_TXN
   | DB_INIT_LOCK | DB_INIT_LOG | DB_THREAD |
   DB_INIT_REP;
myEnv->open(…);

MyEnv->repmgr_start(MyEnv, 3, DB_REP_MASTER);
```

Performance

- Transaction (Auto Commit), without Replication
  - ~ 2000 inserts per second
  - ~ 3500 queries per second
- Still Developing

Q&A

- http://jnlin.org/wp-content/downloads/20080412_intro_bdb.pdf

References

- Getting Started with Data Storage (C) http://www.oracle.com/technology/documentation/berkeley-db/db/gsg/C/index.html
- Getting Started with Transaction Processing http://www.oracle.com/technology/documentation/berkeley-db/db/gsg_txn/C/index.html
- Getting Started with Replication
- http://www.oracle.com/technology/documentation/berkeley-db/db/gsg_db_rep/C/index.html
- Programming APIs http://www.oracle.com/technology/documentation/berkeley-db/db/api_c/frame.html