Redis as a Database Solution

Evan Dorsky and Ben Kahle

March 2014

1 GitHub Repository

https://github.com/EnigMoiD/softsys-3-db

2 Installation

Redis is open source software under the **three clause BSD license** (redis.io). Installation was straightforward, with

```
brew install redis
```

on Mac OS X and

```
apt-get install redis
```

on Ubuntu.

3 Immersion

It's not difficult to get started. Running

```
redis-server
```

starts the Redis server, and running

```
redis-cli
```

opens a Redis command line interface. The documentation on the Redis website was adequate, if not great, for getting up to speed.

4 Semantics

Redis is a flat key-value store. We can list the keys with

```
keys pattern
```

, and if the key holds a set, we can view its contents with

```
smembers key
```

For our purposes, this was very convenient as our schema mapped an actor to a list of actors that the actor shared a movie with. This would make writing our desired search algorithm relatively straightforward.

5 Performance

Redis stores the entire dataset in mememory. This means that reads and writes are very fast, but the dataset cannot be larger than the available memory. However, for our use case this should not be an issue with large server memory available. This is something to keep in mind as the movie database is constantly growing and may require an increasing memory store over time. As a result of in-memory storage, basic query processing is very fast. We can only imagine that the performance would scale well with multiple concurrent queries, though it is important to note that to maintain isolation and atomicity, a single Redis instance process all commands sequentially (one at a time). Initial population of the database was also relatively quick (Python took 10 minutes, counting the string parsing), but reworking the schema into a map from actors to lists of actors took multiple hours. We actually stopped the process because of time constraints, so we're not sure how long it would have taken. Once this precomputation was finished, however, queries would probably be fast.

6 ACIDity

6.1 Atomicity

All Redis transactions are atomic, so this is a big win for Redis.

6.2 Consistency

A single Redis instance is consistent, which should be adequate for our needs.

6.3 Isolation

Because Redis commands are serialized and executed sequentially, all commands in Redis are isolated.

6.4 Durability

Redis sacrifices durability for speed, so it is not ACIDic in this sense.

7 C Interface

The Redis C API, hiredis, is very low level and straight forward; it does not try to do too much out of the box, but provides some nice tools for a more advanced implementation. This is useful because it allows a simple interface that functions very quickly while minimizing potential error points. We were able to write up an interface rather quickly and it doesn't look like it would be difficult to flesh out the interface further.