Redis-IR (Redis Instant Recovery)

Redis-IR is a database prototype that implements the Instant Recovery technique. This technique allows client applications to process transactions immediately upon system restart after a failure. The application may not need to wait for the recovery delay. The main idea of instant recovery is to process transactions during the recovery process, giving the impression that the recovery was instantaneous.

Summary

1 - How to install Redis-IR

Redis-IR can be builded on Ubuntu 18.04 or earlier. However the prototype was tested only on Ubuntu 18.04 and 16.04.
The following libraries are required for building:

- libdb4.8-dev, libdb4.8++-dev (Berkely DB)
- libconfig-dev

Install Berkely DB:

```
1. sudo add-apt-repository ppa:bitcoin/bitcoin
2. sudo apt-get update
3. sudo apt-get install libdb4.8-dev libdb4.8++-dev
```

OR

```
1. wget http://download.oracle.com/berkeley-db/db-4.8.30.zip
2. unzip db-4.8.30.zip
3. cd db-4.8.30
4. cd build_unix/
5. ../dist/configure --prefix=/usr/local --enable-cxx
6. make
7. make install
```

Install Libconfig:

```
1. sudo apt-get update -y
2. sudo apt-get install -y libconfig-dev
```

Install Redis-IR

1. Download Redis-IR at https://drive.google.com/drive/folders/1O4PC8dMTrhZPVhOWsCyAPy13AH71qfOw?usp=sharing
2. tar -xf redis-IR.tar.xz
3. cd redis-IR
4. make

2 - Using Redis-IR

Run Redis-IR server:
1. Enter on *redis-IR/src* directory in a prompt console.
2. ./redis-server

You can interact with Redis-IR using the built-in client:
1. Another console, enter on *redis-IR/src* directory.
2. ./redis-cli

Type some commands:
1. Insert a key/value: *set key value*.
2. Get a value form a key: *get key*.

Try to shout down the database system and run again to observe the recovery. Use the following command on *redis-cli* to shout down:
1. shoutdown

A relatively large workload is required to better observe the instant recovery process. Thus, it is necessary to put the system into production or use a workload simulation tool.

3 - Configuring Redis-IR recovery

Disabling instant recovery

The instant recovery can be disabled setting the *instant_recovery_state* field to *"OFF"* in the *redis_ir.conf* file in the Redis-IR root directory. Thus, the system will run the default Redis recovery, i.e., it will run a sequential log recovery. In this case, transactions can be performed only after the recovery process is complete.

```
instant_recovery_state = "OFF";
```

Enabling synchronous indexing

The instant recovery technique uses an asynchronous log indexing for performance reasons. However, the indexing can be synchronous, i.e, each transaction update must wait a log indexing to confirm its write. To enable the synchronous indexing, set the *instant_recovery_synchronous* field to *"ON"* in the *redis_ir.conf* file in the Redis-IR root directory.

```
instant_recovery_synchronous = "ON";
```

The instant recovery uses asynchronous log indexing by default, i.e., *instant_recovery_synchronous* has the value *"OFF"*. In this case, the system stores log records form sequential log file to indexed log file periodically and asynchronous to transaction processing.

4 - Benchmarking

Redis-IR can use Memtier Benckmark to simile workloads automatically. Memtier is a high-throughput benchmarking tool for Redis developed by Redis Labs. This tool has a command-line interface that provides a set of customization and reporting features to generate various workload patterns.

To install Memtier Benckmark

The following libraries are required for building Memtier:

- libevent 2.0.10 or newer.
- libpcre 8.x.
- OpenSSL (unless TLS support is disabled by ./configure --disable-tls).

On Ubuntu/Debian distributions, simply install all Memtier prerequisites as follows:

1. apt-get install build-essential autoconf automake libpcre3-dev libevent-dev pkg-config zlib1g-dev libssl-dev

Building and installing Memtier:

1. Enter in *redis-IR/src/memtier_benchmark* directory.
2. autoreconf -ivf
3. ./configure
4. make
5. sudo make install

Using Memtier automatically on Redis-IR

Before running the Redis-IR server, configure the following parameters in the *redis_ir.conf* file in the Redis-IR root directory:

1. Set *memtier_benchmark_state* to *ON* to start Memtier at database system restart automatically.
   memtier_benchmark_state = "ON";
2. Set *memtier_benchmark_parameters* to *" --hide-histogram -n 100000 --ratio 5:5 --randomize"*. This parameter generates a workload with 100,000 SET and GET random operations in a 5:5 ratio. The field *hide-histogram* makes the toll don't print detailed latency histogram.
   memtier_benchmark_parameters = " --hide-histogram -n 100000 --ratio 0:11 --randomize";
3. Set *start_benchmark_after* to *"STARTUP"* to start the workload execution at system restart.
   start_benchmark_after = "STARTUP";

At system restart and during recovery process, Memtier can be perform the workload. It is possible see the latency report on the server console. It is also possible to run another client applications.

5 - Using a pre-generated database

It is possible to generate a workload and back it up for later testing. Redis-IR implements persistence only by log files. The standard installation of Redis-IR creates a sequential log file called *sequentialLy.aof* and an indexed log (a B-tree) file called *indexedLog.db*. Both files are stored in the *redis-IR/src/logs* directory. They are required to provide the instant recovery. Therefore, it is necessary to store these two files together for backup purposes. The *finalLogSeek.dat* file synchronizes the indexing of the log records used by Redis-IR and must also be backed up with those files.

It is possible to back up only the sequential log, but it is necessary to generate a new indexed log upon system restart. However, the database restart will be delayed, as transactions cannot be performed during the generation of the log index. After the log index is created, recovery begins. The log index is created at system restart automatically. In addition, a new *finalLogSeek.dat* file also is generated.

You can download some pre-generated databases at
https://drive.google.com/drive/folders/1bbFHQSZdfi8jlMO6-NiMTaPo5jEFOb9w?usp=sharing.

Changing the name of the logs used by Redis-IR

To change the sequential log file name, set the *aof_filename* field in the *redis_ir.conf* file in the Redis-IR root directory.

```
aof_filename = "logs/sequentialLog1GB.aof";
```

To change the indexed log file name, set the *indexedlog_filename* field in the *redis_ir.conf* file in the Redis-IR root directory.

```
indexedlog_filename = "logs/indexedLog1GB.db";
```

6 - Reporting recovery

Redis-IR can produce a simple report with information about the system recovery and execution of a Memtir workload, such as: recovery time, execution time, number of operations performed, among others. In addition, features of each client operation performed on the database can be stored in a CSV file. The CSV file has the fields: key, command, startTime, finishTime, and type. The recovery report and the csv file may be enabled in *redis_ir.conf* file before the system restart.

Generating recovery report

Set the *generate_recovery_report* field to *"ON"* in the *redis_ir.conf* file in the Redis-IR root directory to enable the recovery report generation.

```
generate_recovery_report = "ON"
```

Run the following command in *redis-cli* to generate the recovery report:

```
saveRecoveryInfoToFile <filename>
```

You can clear the recovery information in the system by the command:

```
resetRecoveryInfo
```

Generating a CSV file of executed command features

Set the *generate_executed_commands_csv* field to *"ON"* in the *redis_ir.conf* file in the Redis-IR root directory to enable the csv file generation.

```
generate_executed_commands_csv = "ON"
```

Run the following command in *redis-cli* to generate the CSV file:

```
saveCommandsExecutedToCSV <filename>
```

You can clear the commands executed report in in the system by the command:

```
clearCommandsExecuted
```

Generating recovery report and CSV files automatically

Set the *generate_stats_automatically* field to *"ON"* in the *redis_ir.conf* file in the Redis-IR root directory to generate the recovery report and CSV files automatically after system recovery and/or Memtier workload performing.

```
generate_stats_automatically = "ON"
```

You can set the name of the recovery report file in the *recovery_report_filename* field in the *redis_ir.conf* file in the Redis-IR root directory.

```
recovery_report_filename = "recovery_report/recovery_report.txt";
```

You can set the name of the CSV file in the *executed_commands_csv_filename* field in the *redis_ir.conf* file in the Redis-IR root directory.

```
executed_commands_csv_filename = "datasets/datasets.csv";
```

Generating throughput graphic

The file *Graphics.ipynb* (in Redis-IR root directory) can generate a transaction throughput graphic by a CSV file of commands executed. The file may be used in Jupyter Notebook. It is necessary to know how program in python language to use the scripit.

Arlino Magalhães
arlino@ufpi.edu.br