Using z-checker-installer to install/run Z-checker

Sheng Di[1] and Robert Underwood[2]
[1]Argonne National Laboratory
[2]Clemson Unviersity
sdi1@anl.gov, robertu@g.clemson.edu

Introduction

Z-checker-installer is a very helpful utility that can help you automatically install not only Z-checker library but also all its dependent libraries including different compressors (such as SZ, ZFP, FPZIP, MGARD), data plotting libraries (such as gnuplot, sam2p), compressor adapter (such as libpressio), and analysis modules (such as LibpressioOPT), and so on.

You have two alternative options when doing the installation: you can either install with enabled LibpressioOPT (option A) or install the z-checker without LibpressioOPT (option B).

- As for option A, you need to do all the following three steps.
- As for option B, you can ignore Step 1 and start from Step 2 directly.

What is LibpressioOPT?
LibpressioOPT is a library which provides a facility for automatically configuring a compressor based system on one or more metrics. This function is useful when we want to compare different compressors by aligning the compression ratios in between and observing the reconstructed data's visual quality or so. Without LibpressioOPT, Z-checker also provides a simple search method, while it might not be as accurate as LibpressioOPT does.

Installation Instruction

Step 1. Install Libpressio OPT:
1) Configure passwordless github access: copy the content of ~/.ssh/id_rsa.pub to github's public key field.
Note: The current LibpressioOPT's copyright is under processing, so this is a pre-release version requiring a particular access permission. If you are interested to test it, please email the authors (Sheng or Robert). Once the LibpressioOPT clears the copyright process, the LipbressioOPT will be available for downloading without permission access requirement.

2) Install spack:
```
git clone https://github.com/spack/spack
source ./spack/share/spack/setup-env.sh
```

3) Download Robert's spack env package:
```
git clone https://github.com/robertu94/spack_packages robertu94_packages
```

4) Load Robert's spack repository to the spack:
```
spack repo add ./robertu94_packages
```

5) Modify the spack.yaml in z-checker-installer/libpressio-opt if necessary (please see details in the following).

git pull http://github.com/CODARcode/z-checker-installer
cd libpressioo-opt

for gcc 8.0 or older:

```
specs: [libpressio-opt ^libpressio+sz+zfp ^openblas@0.3.10 ^libstdcompat+boost]
```

otherwise,

```
specs: [libpressio-opt]
```

6) In the z-checker-installer directory, create init.sh with the following content:

```
#!/bin/bash
echo source SPACK/share/spack/setup-env.sh
source SPACK/share/spack/setup-env.sh
echo cd libpressio-opt
cd libpressio-opt
echo spack env activate ./
spack env activate ./
```

Note: SPACK needs to be replaced by the spack's path on your machine.

Then, execute "source init.sh".

7) In the libpressio-opt directory, execute "spack install"

Note: In the future, whenever you need to run z-checker with libpressioOPT, you just need to activate the spack env by running "source init.sh"

After installing LibpressioOPT by doing the above steps, it's ready to install z-checker with LibpressioOPT.

Step 2. Install z-checker-installer:

Option A: If you already did Step 1 (i.e., installed LibpressioOPT), then do the following:

```
git clone http://github.com/CODARcode/z-checker-installer
cd z-checker-installer
./z-checker-install.sh /lcrc/project/ECP-EZ/shdi/LibpressioOpt/libpressio-opt/.spack-env/view
```

Option B: If you don't plan to use LibpressioOPT but use the z-checker's built-in algorithm to search the bestfit error bound setting based on compression ratio, then:

```
git clone http://github.com/CODARcode/z-checker-installer
cd z-checker-installer
./z-checker-install.sh
```

Step 3. Perform test:

In the z-checker-installer directory, modify zc.config by changing plotDecSliceImage ---> 1, and also plotDecSliceMode to be DECVIS_ERROR_LIBPRESSIO_OPT
(The other two options for using the builtin search algorithm in Z-checker are DECVIS_ERROR_SELECT_NEAREST and DECVIS_ERROR_LINEAR_APPROX.)

Then, createZCCase.sh Hurricane. Please see z-checker-installer's readme for more details.

Then, runZCCase.sh -f REL Hurricane /lcrc/project/ECP-EZ/public/compression/Hurricane-ISABEL/test2 bin 500 500 100