Student ID : #932262940 Name : Xuan He

Project Proposal

After searching from the Internet, I decide to select the "BTrees" library as the system under test. The "BTrees" library contains a set of persistent object containers built around a modified BTree data structure. The trees are optimized for use inside ZODB's "optimistic concurrency" paradigm, and include explicit resolution of conflicts detected by that mechannism.

The "BTrees" library can be found in the office website of Python, and the link is https://pypi.python.org/pypi/BTrees/4.1.4#downloads. In addition, in order to test "BTrees" library, the other two libraries - zope.interface and persistent - are required. They also can be found in the office website of Python, and the link is https://pypi.python.org/pypi/zope.interface/4.1.3 and https://pypi.python.org/pypi/persistent#downloads.

There are five modules (OOBTree, IOBTree, OIBTree, IIBTree, and IFBTree) that handle the different variants. The first two letters of the module name specify the types of the keys and values in mappings – O for any object, I for 32-bit signed integer, and F for 32-bit C float. There are four data structures (BTree, Bucket, TreeSet, and Set) provide by each module. For my test, I will focus on OOBTree module, which provides a mapping with arbitrary objects as key and values, and BTree data structure.

The functions I want to test in "BTrees" library are listed below:

1. insert(key, value)
   Insert a key and value into the B Tree. If the key was already in the B Tree, then there is no change and 0 is returned. If the key was not already in the B Tree, then the item is added and 1 is returned. For this function, I will try insert different kinds of values and keys. If inserting is successful, I will check whether it is a B Tree after insert operation by the function "_check()". The correctness means insert function return 0 and the key is already in the B Tree, or return 1 and "_check()" function does not alert error.

2. update(collection)
   Add or update the items from the given collection object to the B Tree. The input collection must be a sequence of (key, value) 2-tuples, or an object with an 'items' method that returns a sequence of (key, value) pairs. For this function, the testing method is similar with insert function. The difference is that it does not have return value, so the correctness means that both update and _check function does not alert error, and the data of B Tree is update correctly,

3. pop(key, d)
   Remove key and return the corresponding value. If key is not found, d is returned if given, otherwise KeyError is raised. For this function, I will try pop different kinds of keys. The correctness means return the corresponding value, the date of B Tree is updated, and _check function does not alert error.

4. get(key, default)

   Get the value associated with the given key.Return the default if has_key(key) is false. For this function, the testing method is similar with pop function. The correctness means that I get the correct value mapping with the key, or get the default value when has_key(key) is false.

5. maxKey(key=None)

   Return the maximum key. If a key argument if provided and not None, return the largest key that is less than or equal to the argument. Raise an exception if no such key exists. For this function, I will try test different kinds of keys and different kinds of B Tree. The correctness means that the return value is the maximum key of the B Tree in my selected range, or raise an exception when there is not satisfied key in the B Tree.

6. minKey(key=None)

   Return the minimum key.If a key argument if provided and not None, return the smallest key that is greater than or equal to the argument. Raise an exception if no such key exists. . For this function, the testing method is similar with maxKey function. The correctness also similar with maxKey function. The only difference is that it selects the minimum key.