Serialized operator test framework

Major functionality lives in `serialized_test_util.py`

How to use

1. Extend the test case class from `SerializedTestCase`
2. Change the `@given` decorator to `@serialized_test_util.given`. This runs a seeded hypothesis test instance which will generate outputs if desired in addition to the unseeded hypothesis tests normally run.
3. [Optional] Add (or change a call of `unittest.main()` to) `testWithArgs` in `__main__`. This allows you to generate outputs using `python caffe2/python/operator_test/my_test.py -G`.
4. Run your test `python -m pytest caffe2/python/operator_test/my_test.py -G` to generate serialized outputs. They will live in `caffe2/python/serialized_test/data/operator_tes` one zip file per test function. The zip file contains an `inout.npz` file of the inputs, outputs, and meta data (like device type), a `op.pb` file of the operator, and `grad_#.pb` files of the gradients if there are any. Use `-O` to change the output directory. This also generates a markdown document summarizing the coverage of serialized tests. We can disable generating this coverage document using the `-C` flag.
5. Thereafter, runs of the test without the flag will load serialized outputs and gradient operators for comparison against the seeded run. The comparison is done as long as you have a call to assertReferenceChecks. If for any reason the seeded run's inputs are different (this can happen with different hypothesis versions or different setups), then we'll run the serialized inputs through the serialized operator to get a runtime output for comparison.

Coverage report

`SerializedTestCoverage.md` contains some statistics about the coverage of serialized tests. It is regenerated every time someone regenerates a serialized test (i.e. running an operator test with the `-G` option). If you run into merge conflicts for the file, please rebase and regenerate. If you'd like to disable generating this file when generating the serialized test, you can run with `-G -C`. The logic for generating this file lives in `coverage.py`.

##Additional Notes

If we'd like to extend the test framework beyond that for operator tests, we can create a new subfolder for them inside `caffe2/python/serialized_test/data`.

Note, we currently don't support using other hypothesis decorators on top of `given_and_seeded`. Hypothesis has some handling to explicitly check that `@given` is on the bottom of the decorator stack.

If there are multiple calls to assertReferenceChecks in a test function, we'll serialize and write the last one. The actual input checked may then differ if we refactor a test function that calls this multiple times, though the serialized test should still pass since we then use the serialized input to generate a dynamic output.