Replacing BOOST unit testing with eckit

JEDI meeting

March/21/2019

maryamao@ucar.edu

Changes in OOPS

feature/unit_testing_replacingBOOST

Some of the changes:

- Test.h
- All the tests OOPS (test/interface, …)
- Tests in Lorenz95

To use eckit unit testing:

- In CMakeLists.txt remove BOOST:

```
ecbuild_add_test( TARGET test_ioda_observationspace
                  BOOST
                  SOURCES mains/TestObservationSpace.cc
                  ARGS "testinput/iodatest.yml"
                  LIBS    ioda )
```

- Change tolerance in yaml files:

BOOST_CHECK_CLOSE(XX, ZZ, tol); |XX-ZZ|; // < XX*tol/100

EXPECT(is_approximately_equal(XX, ZZ, tol)); // |XX-ZZ| < tol

- If there is any independent test in your repository, you will need to change it. Examples in the following slides:

BOOST

```
#ifndef TEST_INTERFACE_GEOVALS_H_
#define TEST_INTERFACE_GEOVALS_H_

#include <cmath>
#include <string>
#include <vector>

#define BOOST_TEST_NO_MAIN
#define BOOST_TEST_ALTERNATIVE_INIT_API
#define BOOST_TEST_DYN_LINK
#include <boost/test/unit_test.hpp>


#include <boost/noncopyable.hpp>
#include <boost/scoped_ptr.hpp>


#include "eckit/config/LocalConfiguration.h"
#include "oops/base/ObsSpaces.h"
#include "oops/interface/GeoVaLs.h"
#include "oops/interface/Locations.h"
#include "oops/interface/ObsOperator.h"
#include "oops/runs/Test.h"
#include "oops/util/dot_product.h"
#include "test/TestEnvironment.h"


namespace test {
```

ECKIT

Example

```
#ifndef TEST_INTERFACE_GEOVALS_H_
#define TEST_INTERFACE_GEOVALS_H_

#include <cmath>
#include <string>
#include <vector>

#define ECKIT_TESTING_SELF_REGISTER_CASES 0


#include <boost/noncopyable.hpp>
#include <boost/scoped_ptr.hpp>


#include "eckit/config/LocalConfiguration.h"
#include "eckit/testing/Test.h"
#include "oops/base/ObsSpaces.h"
#include "oops/interface/GeoVaLs.h"
#include "oops/interface/Locations.h"
#include "oops/interface/ObsOperator.h"
#include "oops/runs/Test.h"
#include "oops/util/dot_product.h"
#include "test/TestEnvironment.h"


using eckit::types::is_approximately_equal;


namespace test {
```

Example

```
BOOST_CHECK(TestContainer.get());
// replace with
EXPECT(TestContainer.get());

//-------------------------------------

BOOST_CHECK_EQUAL(Nlocs, ExpectedNlocs);
// replace with
EXPECT(Nlocs == ExpectedNlocs);

//-------------------------------------

BOOST_CHECK_CLOSE(Vnorm, ExpectedVnorm, Tolerance);
// replace with
EXPECT(is_approximately_equal(Vnorm, ExpectedVnorm, Tolerance));

//-------------------------------------
BOOST_CHECK_SMALL( val, tolerance )
// replace with
EXPECT(val < tolerance)
```

** For more examples refer to OOPS branch *feature/unit_testing_replacingBOOST*

Example

BOOST

```
class ObsVector : public oops::Test {
 public:
  ObsVector() {}
  virtual ~ObsVector() {}
 private:
  std::string testid() const {return
"test::ObsVector<ioda::IodaTrait>";}


  void register_tests() const {
    boost::unit_test::test_suite * ts =
BOOST_TEST_SUITE("ObsVector");


    ts->add(BOOST_TEST_CASE(&testConstructor));
    ts->add(BOOST_TEST_CASE(&testCopyConstructor));
    ts->add(BOOST_TEST_CASE(&testNotZero));
    ts->add(BOOST_TEST_CASE(&testRead));
    ts->add(BOOST_TEST_CASE(&testSave));


    boost::unit_test::framework::master_test_suite().add(ts);
  }
};
```

ECKIT

```
class ObsVector : public oops::Test {
 public:
  ObsVector() {}
  virtual ~ObsVector() {}
 private:
  std::string testid() const {return
"test::ObsVector<ioda::IodaTrait>";}


  void register_tests() const {
    std::vector<eckit::testing::Test>& ts =
eckit::testing::specification();


    ts.emplace_back(CASE("ioda/ObsVector/testConstructor")
     { testConstructor; });
    ts.emplace_back(CASE("ioda/ObsVector/testCopyConstructor")
     { testCopyConstructor; });
    ts.emplace_back(CASE("ioda/ObsVector/testNotZero")
     { testNotZero; });
    ts.emplace_back(CASE("ioda/ObsVector/testRead")
     { testRead; });
    ts.emplace_back(CASE("ioda/ObsVector/testSave")
     { testSave; });
  }
};
```

** For more examples refer to OOPS branch *feature/unit_testing_replacingBOOST*

Example

BOOST

```
template <typename MODEL> class GeoVaLs : public oops::Test {
 public:
  GeoVaLs() {}
  virtual ~GeoVaLs() {}
 private:
  std::string testid() const {return "test::GeoVaLs<" +
MODEL::name() + ">";}


  void register_tests() const {
    boost::unit_test::test_suite * ts =
BOOST_TEST_SUITE("interface/GeoVaLs");


    ts->add(BOOST_TEST_CASE(&testConstructor<MODEL>));
    ts->add(BOOST_TEST_CASE(&testRead<MODEL>));


    boost::unit_test::framework::master_test_suite().add(ts);
  }
};
```

ECKIT

```
template <typename MODEL>
class GeoVaLs : public oops::Test {
 public:
  GeoVaLs() {}
  virtual ~GeoVaLs() {}
 private:
  std::string testid() const {return "test::GeoVaLs<" +
MODEL::name() + ">";}


  void register_tests() const {
    std::vector<eckit::testing::Test>& ts =
eckit::testing::specification();


    ts.emplace_back(CASE("interface/GeoVaLs/testConstructor")
      { testConstructor<MODEL>(); });
    ts.emplace_back(CASE("interface/GeoVaLs/testRead")
      { testRead<MODEL>(); });
  }
};
```

** For more examples refer to OOPS branch *feature/unit_testing_replacingBOOST*