Ticket #4410 (closed Patches: worksforme)

Sparse/Packed matrix assignment needs type conversion

Opened 2 years ago
Last modified 2 years ago

Reported by:	Marco Guazzone <marco.guazzone@...>	Owned by:	david.bellot
Milestone:	To Be Determined	Component:	uBLAS
Version:	Boost Development Trunk	Severity:	Problem
Keywords:		Cc:	

Description

In file *boost/detail/matrix_assign.hpp* there are two possible source of type-conversion error interesting both *sparse* and *packed* matrices.

Let:

```
typedef typename M::value_type value_type;
typedef F<typename M::..., typename E::value_type> functor_type;
```

Then:

1. *Comparison of an `E::value_type` with an `M::value_type`*

   ```
   if (v != value_type/*zero*/()) // where v is of type E::value_type
   ```

   E.g., `E::value_type` is `std::complex<float>` and `M::value_type` is `std::complex<double>`.

2. *Assignment of an `M::value_type` to an `E::value_type`*

   ```
   functor_type::apply(*it, value_type/*zero*/());
   ```

   E.g., `E::value_type` is `float` and `M::value_type` is `std::complex<float>`.

Attachments

- matrix_assign_problem.cpp (587 bytes) - added by *Marco Guazzone <marco.guazzone@...>* 2 years ago.
  A sample program for showing the problem (the program should not compile).
- matrix_assign-packed_sparse_storage-type_conversion.patch (13.4 KB) - added by *Marco Guazzone <marco.guazzone@...>* 2 years ago.
  Possible solution.
- test_ticket4410.cpp (799 bytes) - added by *Marco Guazzone <marco.guazzone@...>* 2 years ago.
  Test case: test copy-construction/-assignement of a sparse (symmetric) matrix. The test fails to compile if the patch is not applied.

Change History

Changed 2 years ago by Marco Guazzone <marco.guazzone@…>

- **attachment** *matrix_assign_problem.cpp* added

A sample program for showing the problem (the program should not compile).

Changed 2 years ago by Marco Guazzone <marco.guazzone@…> comment:1

I propose a possible patch (see attachment: *matrix_assign-packed_sparse_storage-type_conversion.patch*).

Essentially,

1. Expressions of the first type might be changed by casting an `E::value_type` to a `M::value_type`, like in this way:

```
if (static_cast<value_type>(v) != value_type/*zero*/())
```

Obviously, this does not work when `E::value_type` and `M::value_type` are not *castable* (e.g., `std::complex` and `double`, respectively).

2. Expressions of the second type might be changed in 2 ways:

- Option A (the one used in the proposed patch)

```
typedef typename matrix_traits<E>::value_type expr_value_type;
functor_type::apply(*it, expr_value_type/*zero*/()); // NOTE: use o
```

- Option B

```
typedef F<typename M::..., value_type> functor_type; // NOTE: use M.
functor_type::apply(*it, value_type/*zero*/()); // unchanged
```

Changed 2 years ago by Marco Guazzone <marco.guazzone@…>

- **attachment** *matrix_assign-packed_sparse_storage-type_conversion.patch* added

Possible solution.

Changed 2 years ago by Marco Guazzone <marco.guazzone@…> comment:2

There is also a companion post un uBLAS ml:

http://lists.boost.org/MailArchives/ublas/2010/07/4420.php

Changed 2 years ago by david.bellot comment:3

- **Owner** changed from *guwi17* to *david.bellot*
- **Status** changed from *new* to *assigned*

Changed 2 years ago by david.bellot comment:4

applied patch from Marco Guazzone. Should be OK, however it raises a concern about a possible security hole with static_cast<>, only when people are crazy enough to use ublas as a data storage having nothing to do with linear algebra (like I did once ;-))

Changed 2 years ago by david.bellot comment:5

- **Status** changed from *assigned* to *closed*
- **Resolution** set to *worksforme*

Changed 2 years ago by Marco Guazzone <marco.guazzone@…>

- **attachment** *test_ticket4410.cpp* added

  Test case: test copy-construction/-assignement of a sparse (symmetric) matrix. The test fails to compile if the patch is not applied.