

Master Thesis / Bachelor Thesis / Student Researcher Position

Benchmarking C++ Codes: Profiling Template Instantiation at Compile Time

Description: Advanced C++ programming is often based on template metaprogramming techniques. With ongoing developments in the C++ language standard, complex type deduction gets even more important. Template metaprogramming is heavily used within our Algorithmic Differentiation (AD) tools. AD is a program transformation technique for the automatic generation of *adjoint codes* for a given numerical simulation program. Adjoint codes play a crucial role in sensitivity analysis and optimization. The compilation times of those transformed codes are getting more and more problematic and depending on the compiler even infeasible.

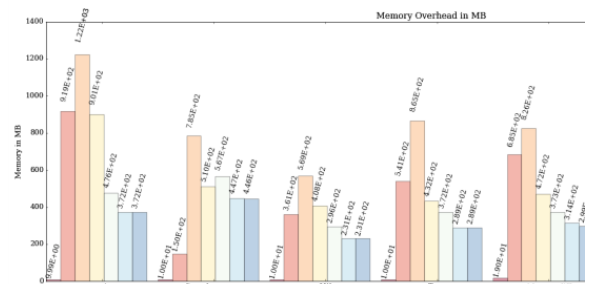
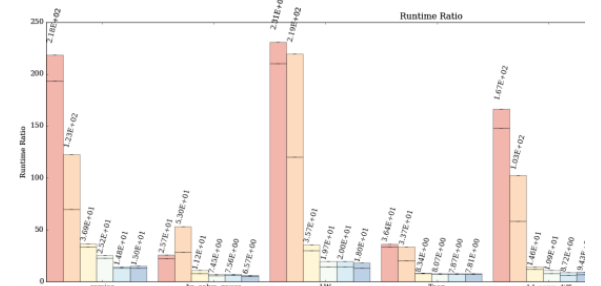
Goal: Though benchmarking the compilation process is very compiler and platform dependent, it is highly desirable in the wake of increasing template metaprogramming use. Getting reliable benchmarking and profiling data for the AD tools developed at our institute is the outcome of this thesis.

Profile: You should have worked with C/C++ and respective compilers a lot under Linux as well as Windows. In addition, knowledge in the field of numerical simulation and optimization is beneficial.

If you are interested in a bachelor or master thesis or a student researcher position (up to 19 hours a week) on this topic, please do not hesitate to contact us!

Contact: Dr. Johannes Lotz
ITC, Seffenter Weg 23, Room 124
lotz@stce.rwth-aachen.de

```
template <typename T, typename U>
typename std::enable_if
<std::remove_reference<T>::type::is_adjoint &&
std::remove_reference<U>::type::is_adjoint,
internal::bi<typename std::remove_reference<T>::type::base_t,
typename helper::node_struct<T>::type,
typename helper::node_struct<U>::type,
operations::dco_##OPNAME>>::type
FUNCTION (T&& x1, U&& x2) DCO_MAPI_NOEXCEPT {
static_assert(std::is_same<
typename std::remove_reference<T>::type::base_t, \
typename std::remove_reference<U>::type::base_t>::value, \
"\n\n" \
"\t \
\tDCO_MAP ERROR: Mixed precision computation is currently not sup \
\tbut is possible. If you require mixed precision in dco/map, pl \
\tcontact NAG.\n" \
"\n\n"); \
```



```
[=====] Running 11 benchmarks, skipping 1 benchmark.
[ RUN ] DeliveryMan.DeliverPackage (10 runs, 100 itera
[ DONE ] DeliveryMan.DeliverPackage (2069.081880 ms)
[ RUNS ] Average time: 206908.188 us (~334.506 u
Fastest time: 206250.788 us (-657.400 u
Slowest time: 207368.788 us (+460.600 u
Median time: 206923.288 us (1st quarti
```

```
Average performance: 4.83306 runs/s
Best performance: 4.84847 runs/s (+0.01540
Worst performance: 4.82233 runs/s (-0.01074
Median performance: 4.83271 runs/s (1st quart
```

```
[ITERATIONS] Average time: 2069.082 us (~3.345 us)
Fastest time: 2062.508 us (-6.574 us /
Slowest time: 2073.688 us (+4.606 us /
Median time: 2069.233 us (1st quartile
```