

Turn the library inside the NT2 project in Boost.SIMD and improving it's support for the AltiVec instruction set

Faïçal Tchirou

April 8, 2011

1 Personal Details

- Name : Faïçal Tchirou
- University : African Institute of Computer Science (Institut Africain d'Informatique)
- Degree Program : B. Sc.
- Email : tfeysal@hotmail.com
- Homepage : Not available yet
- Availability :
 - How much time do you plan to spend on your GSoC ?
I will have my final exam at the end of the month of June. So, between July and September, I could spend at least 6 hours per day on my GSoC project.
 - What are your intended start and end dates ?
I've already started. I mean that I start reading some documentation and references manuals, especially boost documentation, the nt2 project source code and it's documentation and some manuals about SIMD and AltiVec instruction set. I plan to continue familiarizing myself with this stuff before the community bonding period and to finish my work in the time imparted to me.
 - What other factors affect your availability (exams, courses, moving, work, etc.) ?
It is expected in my school that my final exam will be at the end of the month of June. If no unexpected event occurs, I will be fully available to work in the summer.

2 Background Information

2.1 Education Background

I am in second year at the African Institute of Computer Science (it's real name in French is Institut Africain d'Informatique). My school is not a really a Computer Science school but a Software Engineering one. We have basic courses on programming languages (Pascal, C/C++ and Java), on basic algorithms, on management and software design. We don't have real courses on algorithms (their analysis with asymptotic notations, their design, their proof, ...), data structures like trees, graphs, heaps (we were only basically taught about linked lists), programming methodology, etc. because we are primarily formed to work with RAD applications like WinDev or Delphi. However, me and several students who are primarily interested in computer science rather than software engineering learn by ourselves by reading books like Structure and Interpretation of Computer Programs (available online), the courses available online by the OpenCourseWare initiative and a lot of tutorials on web.

2.2 Programming Background

I started learning programming in 2006 with HTML but I really start programming in 2009 with the C language by trying to solve mathematical problems available on projecteuler.net. When attempting to learn functional programming, I discover Common Lisp which I also used to solve some projecteuler.net problems. I switched to C++ when trying to solve problems available at uva.onlinejudge.org. I used it first only for the STL because it provides very good built-in data-structures I used when solving algorithmics problems. But now I have a good knowledge of object oriented and template programming in C++. I also learned some other programming languages like Python, Java, ... and of course web oriented programming languages like PHP and a little of Ruby

So my programming background can be summarized in solving algorithmics problems in C, Lisp, and C++ and doing some academic projects like inventory management software, websites,

2.3 Programming interests and interest in Boost

I am particularly interested in algorithmics problems especially dynamic programming, graphs and mathematical ones and system programming. I first discover Boost when searching a library which implements graph algorithms. I read about it but I did not use it at that time. It was clear that this library is written by very good programmers and if I had the chance to work with it's authors and it's contributors I will learn a lot.

2.4 Interest in the Boost.SIMD library

I really want to learn by working on a challenging project with very good programmers and get involved in an open source project.

I hope if the GSoC ends, I will continue to work with them to benefit from their knowledge.

2.5 Do you have any previous work in this area before or on similar projects?

No.

2.6 What are your plans beyond this Summer of Code time frame for your proposed work?

I think a GSoC work is not only a 3 months project. After the deadline, I will remain in contact with my mentor and the project community and continue contributing to the project.

2.7 Please rate, from 0 to 5, your knowledge of the following languages, technologies or tools:

- C++ : 3
- C++ Standard Library : 3
- Boost C++ Libraries : 1
- Subversion : 1 (I'm more familiar with Git)

2.8 What software development environments are you most familiar with ?

I principally use Emacs on my Fedora. I am also familiar with Eclipse and Code::Blocks on GNU/Linux. If I have to work on Windows, I can work both on Visual Studio, Eclipse or Code::Blocks.

2.9 What software documentation tool are you most familiar with ?

I am most familiar with Doxygen.

3 Project Proposal

SIMD is a class of instruction sets in processors that allow to execute an operation on multiple data elements simultaneously. The future Boost.SIMD library already exists as a part of the NT2 project. The project consists of

- Making the library inside the NT2 project more boost compliant by following the boost coding and naming conventions.
- Improving the support of the AltiVec instruction set.
- Writing users and developers documentations.

As an example of making the library more compliant, this file (comments ommited)

```
1 #ifndef NT2_SDK_SIMD_DETAILS_IMPL_VMX_ALTIVEC_BITWISE_AND_HPP_INCLUDED
2 #define NT2_SDK_SIMD_DETAILS_IMPL_VMX_ALTIVEC_BITWISE_AND_HPP_INCLUDED
3
4 #include <nt2/sdk/meta/as_integer.hpp>
5 #include <nt2/sdk/simd/native_cast.hpp>
6
7 NT2_REGISTER_DISPATCH ( tag::bitwise_and_, tag::cpu_, (A0)(A1)
8                       , ((simd_<arithmetic_<A0>,tag::altivec_>))
9                       , ((simd_<arithmetic_<A1>,tag::altivec_>))
10                      );
11
12 namespace nt2 { namespace ext
13 {
14   template<class Dummy>
15   struct call< tag::bitwise_and_ ( tag::simd_<tag::arithmetic_,tag::altivec_>
16                                   , tag::simd_<tag::arithmetic_,tag::altivec_>
17                                   )
18                                   , tag::cpu_, Dummy
19                                   >
20   : callable
21   {
22     template<class Sig> struct result;
23     template<class This, class A0, class A1>
24     struct result<This(A0,A1)> : meta::strip<A0> {};
25
26     NT2_FUNCTOR_CALL(2)
27     {
28       A0 other = simd::native_cast<A0>( a1 );
29       A0 that = { vec_and(a0(),other()) };
30       return that;
31     }
32   };
33 } }
34
35 #endif
```

can become (comments ommited)

```
1 #ifndef BOOST_SIMD_BITWISE_AND_HPP
2 #define BOOST_SIMD_BITWISE_AND_HPP
3
4 #include <boost/meta/as_integer.hpp>
5 #include <boost/simd/native_cast.hpp>
6
7 BOOST_REGISTER_DISPATCH (tag::bitwise_and_, tag::cpu, (A0)(A1),
8                       ((simd_<arithmetic_<A0>, tag::altivec_>))
9                       , ((simd_<arithmetic_<A1>, tag::altivec_>))
10                      );
11
12 namespace boost {
13   namespace ext
```

```

14 {
15     template<class Dummy>
16     struct call<tag::bitwise_and_ (tag::simd_<tag::arithmetic_, tag::altivec_>,
17                                   tag::simd_<tag::arithmetic_, tag::altivec_>
18                                   ),
19                                   tag::cpu_, Dummy
20                                   >
21     {
22         template<class Sig> struct result;
23         template<class This, class A0, class A1>
24         struct result<This(A0,A1)> : meta::strip<A0> {};
25
26         BOOST_FUNCTOR_CLASS(2)
27         {
28             A0 other = simd::native_cast<A0>(a1);
29             A0 that = { vec_and(a0(), other()) };
30             return that;
31         }
32     };
33 } // namespace ext
34 } // namespace boost
35
36 #endif // BOOST_SIMD_BITWISE_AND_HPP

```

3.1 Proposed Milestones and Schedules

- Before April 25
 - Continue reading Boost, AltiVec and SIMD documentations.
 - Familiarize myself with Boost coding and naming conventions.
 - Familiarize myself with Boost.Proto.
 - Learning Boost documentation tool.
- April 25 - May 23
 - Remain in contact with my mentor.
 - Asking all necessary clarifications to the mentor.
 - Remain active on the mailing lists and on IRC.
- May 23 - July 11
 - Defining the library structure inside of Boost (folder tree, etc.)
 - Boostification of the library inside of the NT2 project.
 - Writing comments in the code according to the Boost documentation tool to later generate the documentation.
- July 11 - July 15
 - Define the necessary modules needed to consolidate the support of AltiVec instruction set.
- July 15 - August 15
 - Coding the necessary modules to consolidate the support of AltiVec instruction set.
- August 15 - August 22
 - Check the code written.
 - Polishing the final Boost.SIMD library.
 - Working on tests and bug fixes.
 - Writing user documentation, examples and how-tos.

I will remain involved in the community after the project.