

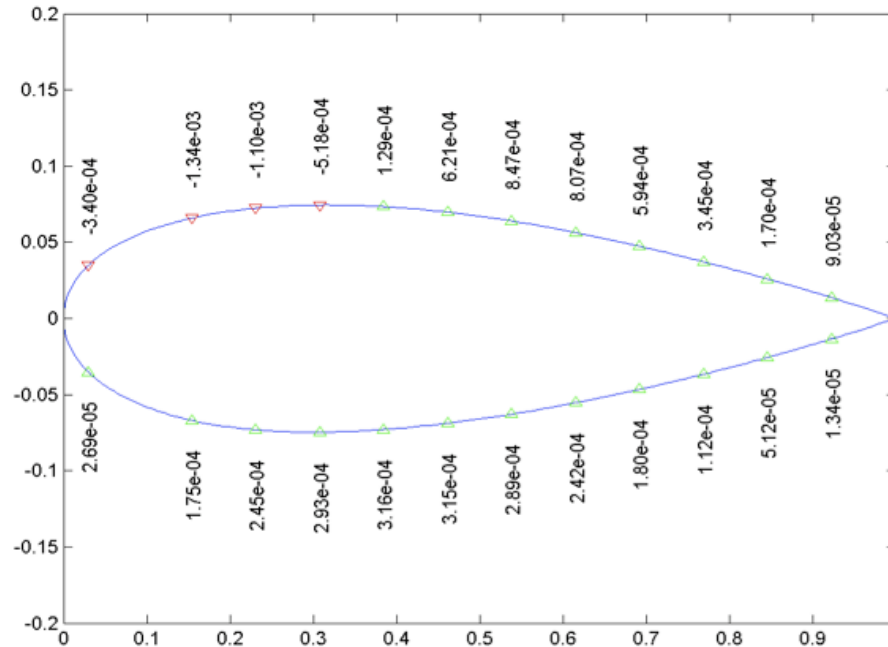


Benchmark of an airfoil simulation

Solving the Euler equations
using numerical methods.

By Kevin Veragoo

Airfoil Simulation



The algorithm computes the drag and lift at transonic regime with Mach value of 0.7 and Angle of Attach (AoA) of 4 degrees for an airfoil. The airfoil geometry is characterized using 24 design variables as depicted above. The lift and drag coefficients are found by solving the 2-dimensional Euler equations by using the finite volume method for compressible flows.

As far as computation is concerned, **lots of floating point additions**

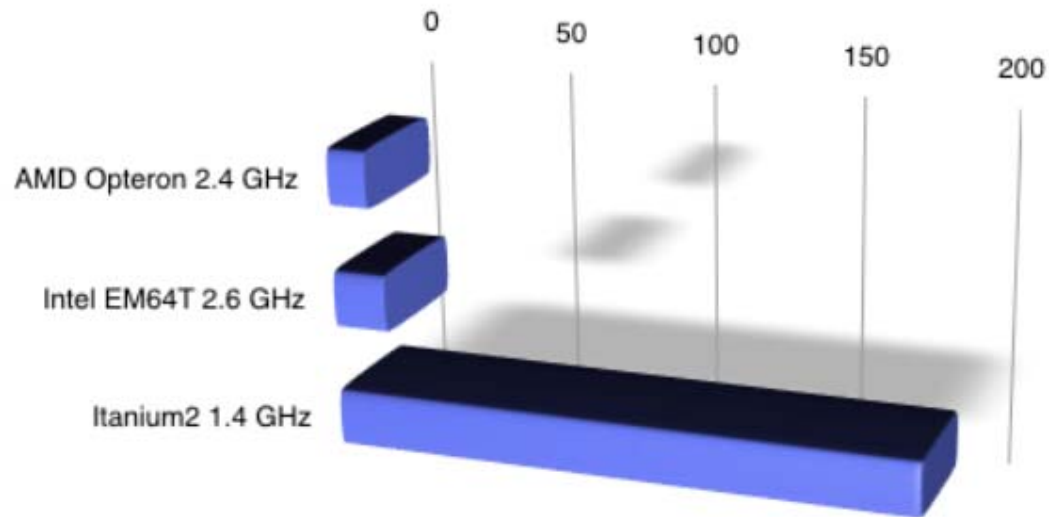
Computer Architectures

Operating System : CentOS 64 bit

Processors : Intel EM64T 2.6GHz
: AMD Opteron 2.4 GHz
: Intel Itanium2 1.4 GHz

Compilers : gcc and g77
: icc and ifort

gcc and g77 Performance



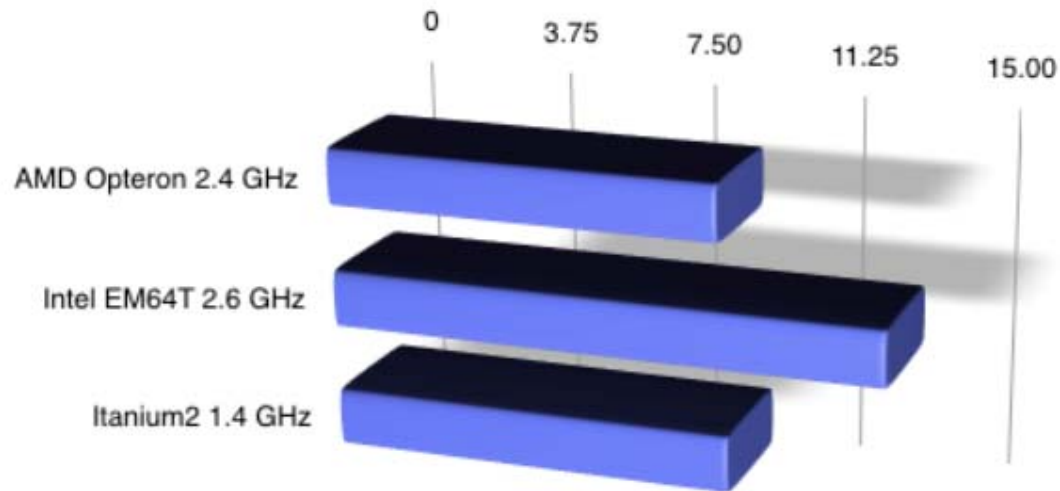
Comp Architecture	Time
Opteron 2.4 GHz (4.8 Gflops)	12.271
Xeon 2.6 GHz (5.2 Gflops)	16.806
Itanium2 1.4 GHz (5.6 Gflops)	185.351

Porting to Intel Compilers

Intel compilers need to link the executable using ifort and not icc, which causes trouble if your main function is written in C.

This is resolved, with a little hack, compiling all files as libraries and linking them all at the end with “-nofor_main” flag to ifort

icc and ifort Performance



Comp Architecture	Time
Opteron 2.4 GHz (4.8 Gflops)	9.261
Xeon 2.6 GHz (5.2 Gflops)	13.074
Itanium2 1.4 GHz (5.6 Gflops)	9.488

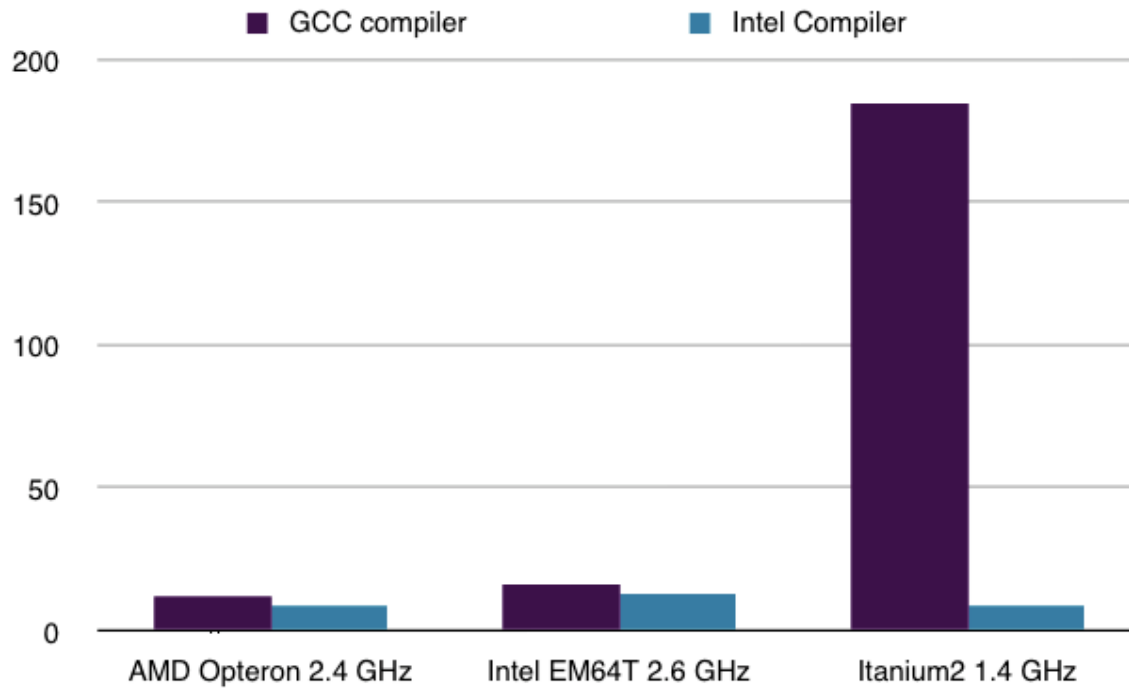
Itanium2 Performance

with Intel compilers

Architecture	Performance (More is better)	Clockspeed (More is faster)
Opteron	1x	1x
Itanium2	0.976x	0.58x
Xeon	0.708x	1.08x

To note that the Itanium2 is slightly outperformed by the Opteron, despite it's heavy burdened lower clockspeed.

Itanium2 Benchmark



Airfoil on Itanium2 Summary

We believe that the gcc/ g77 compiled airfoil performs badly because of it's inability to make full use of the 128 floating point registers.

In contrast, the icc/ ifort compiled version really seems to perform quite well given the Itanium2's clock speed.