64-BIT C/C++ COMPILER

EKOPath 2.1

Biagio Lucini likes Mflops. He really, really likes them. Question is, does this compiler give him enough?

BUYER INFO

A compiler suite highly optimised for AMD64 and EM64T architectures. Also consider: *GCC*, Intel compilers.

- **DEVELOPER** PathScale
- **WEB** www.pathscale.com/ekopath.html
- **PRICE** See box below

AMD64 and the newer EM64T architecture have easily superseded the 32-bit derivations of the x86 processor. Given their relatively low price and the higher speeds they guarantee, it's no wonder that these new 64-bit systems have already taken a substantial share of the high-performance computing market. However, although having a fast processor is key to speeding up applications, there is another contributing factor: the compiler. And while there are several compilers around that can generate 64-bit code, a large number of them are simply a port of their 32-bit counterpart.

A notable exception is the *EKOPath* compiler suite, produced explicitly for 64-bit x86-compatible architectures. Its developers at PathScale say their compilers produce the fastest code on AMD64-based processors. Ever alert to the whiff of hyperbole, we installed the latest version of the C/C++ compiler to test their claim.

The *EKOPath* compiler suite is based on the open source Itanium compiler *Open64*, which in turn is based on the once-popular *SGI* compiler suite. The suite itself consists of the C/C++ compiler that we're testing here and a Fortran90/95 compiler. The latest release available at the time of writing is 2.1, which supports various recent Red Hat Enterprise and SUSE versions as well as Fedora Core. It doesn't support the most recent releases such as Fedora Core 3 and SUSE Pro 9.3, and other popular choices such as Debian are not even mentioned, which is a shame. However, if your distro isn't supported, it doesn't necessarily mean that the compilers won't work on them: just that you're on your own if things go wrong. If you do have an unsupported distribution you could try the 30-day demo version of *EKOPath*.

We should stress that although they can generate generic x86 code, the two compilers are intended to be installed on AMD64 and EM64T architectures. The suite requires a lot of RAM (512MB will do, but 2GB is recommended), which most if not all x86-compatible 64-bit machines available on the market can easily provide. Feature-wise, the *EKOPath* suite offers support for everything a developer might want. We like its compatibility with *GCC*, with the definition of macros and built-in constructs typical of the GNU compiler. This allows you to compile a great deal of open source software out of the box.

Also noticeable are the range of C/C++ standards that can be enforced at compilation time and the support for *OpenMP 2.0* (for C only), which will please any number crunchers.

Speed at a price?

There are two ways of installing the software: use *RPM* (for which you need to be root) or extract it from a tarball (for a non-root installation). The latter method is a bit more involved, and this is the route we took – you'd be disappointed if we didn't! The tarball can be extracted in any directory, but to use the compiler you need to set the **LD_LIBRARY_PATH** variable to point to an appropriate location or update **/etc/ld.so.conf**. For further comfort the **PATH** and **MAN** environment variables can be updated.

You will need to buy a licence for every single developer for each node you plan to use the compiler on – so a user who has accounts on two development nodes needs two licences. Licences are not perpetual: the subscription needs to be renewed each year. After 30 days from the end of the subscription period the compiler will stop working. With this licensing model, PathScale compiler suite could become pretty expensive, even for a medium-sized company.

After copying the licence file to the compiler library directory, you're ready to use the compiler. The accompanying documentation is very detailed. Well-written summaries highlight the important steps and all relevant information, and a debugger application is also provided.

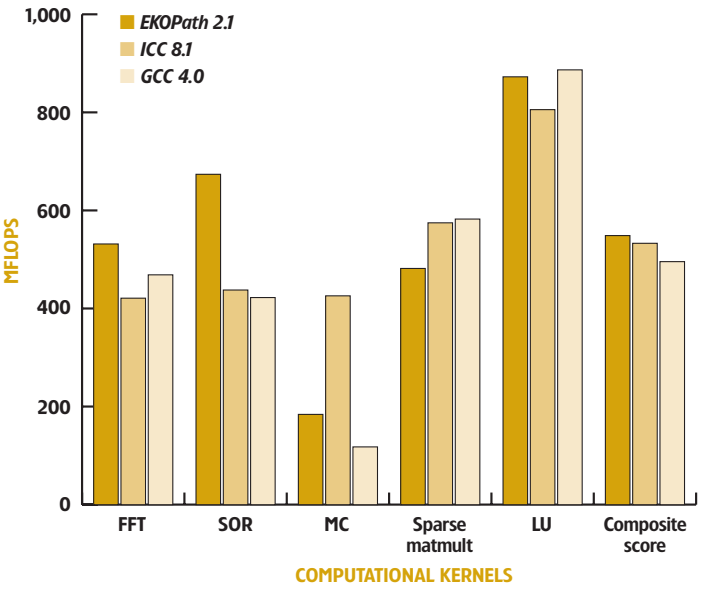
The test

We compared the PathScale C/C++ compiler with two strong performers on the AMD64 platform: the *Intel C/C++ Compiler* 8.1 (we took the 32-bit version) and the new *GCC 4.0*. We executed the SciMark2 benchmarks on a system with two AMD Opteron 244 processors and 4GB of RAM. The fastest executables were obtained with the **-Ofast -static** compiler flags. Results are shown above.

The claim that the *EKOPath* suite produces the fastest code on AMD64 systems is generally true, as is shown by the composite score and about half of the individual benchmark scores.

However, the rest of the benchmarks suggest that the C/C++ PathScale compiler might not always be the best choice. Given the high price, we'd suggest you try it out on your real-world source code before you splash out. PathScale says that losing a benchmark is comparable to finding a bug in the software and that it will fix any such issue as a high priority. We're sure it will – but while we're waiting we can only give *EKOPath* a partial thumbs-up. LXF

The performance of the *EKOPath C/C++ Compiler* measured by the benchmark suite SciMark2, compared with those of its closest rivals.



KEY

Mflops = Floating point operations per second, in millions
FFT = Fast Fourier Transformation
SOR = Jacobi successive over-relaxation
MC = Monte Carlo integration
Sparse matmult = Sparse matrix vector multiplications
LU = LU factorisation
Details at http://math.nist.gov/scimark2

EKOPATH PRICES

	1 year	2 years	3 years
Entire suite	$ 1,495	$ 2,695	$ 3,795
Fortran	$ 1,095	$ 1,990	$ 2,790
C/C++	$ 595	$ 1,080	$ 1,495

Prices are a subscription for a single developer on a single node. They exclude VAT, shipping and handling taxes. Volume and academic discounts are available.

LINUX FORMAT VERDICT

FEATURES	8/10
PERFORMANCE	9/10
EASE OF USE	7/10
VALUE FOR MONEY	5/10

Very good overall, but the performance does not justify the high price tag. Perhaps their Fortran scores are higher?

RATING 7/10