Follow Ups…

Date: 10-07-2015

Pragmas (C/C++ only)

- **`#pgragma fenv_access`**
  Informs about possibly changed FP environment; requires **`strict`** FP model (see **`fenv.h`**)
- Block-wise control:
  `#pragma float_control(…,[on|off])`
  Turn on/off FP model settings

Examples:

- **`#pragma float_control(except,[on|off])`**
  Compiler has to expect/handle FP exceptions
  Alternative: use **`strict`** or **`except`** FP model
- **`#pragma float_control(fma,[on|off])`**
  FP contractions are allowed/disallowed
  Alternative: use **`strict`** FP model; **`-no-fma`** or **`/Qfma-`**

Optimization Notice

Copyright © 2015, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Pragmas (C/C++ only) - Mapping

- `#pragma float_control(…,[on|off])` currently does not map to `-fp-model` options directly…
- Use the following mapping as workaround:
  - `-fp-model fast`:
    `#pragma float_control (precise, off)`
  - `-fp-model precise`:
    `#pragma float_control (precise, on)`
  - `-fp-model strict`:
    `#pragma fenv_access (on)`
    `#pragma float_control (except, on)`

7/10/2015

Optimization Notice
Copyright © 2015, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Loop Blocking Pragma/Directive

- Syntax:

C++

```
#pragma block_loop
[clause[,clause]...]
#pragma noblock_loop
```

Fortran

```
!DIR$ BLOCK_LOOP [clause[[,]
clause]...]
!DIR$ NOBLOCK_LOOP
```

```
clause:
  factor ( expr )
  level ( levels )
  private ( var1
    [,var2 ]...
```

- BLOCK_LOOP enables greater control over optimizations on specific DO/for loop inside a nested loop
- Uses loop blocking technique to separate large iteration counted loops into smaller iteration groups
- Smaller groups can increase efficiency of cache space use and augment performance
- Works seamlessly with other directives including SIMD

Cache/Loop blocking:
https://software.intel.com/en-us/node/540516
https://software.intel.com/en-us/articles/cache-blocking-techniques

Optimization Notice
Copyright © 2015, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Loop Blocking Sample

Original Source Code:

```
#pragma block_loop factor(250) level(2)
  for (i=0; i < m; i++)
  {
     for (j=0; j < m; j++)
       {
         c[i]+=a[i][j]*b[j];
       }
  }
```

Outline of code after compiler loop transformations:

```
for (jj=0;jj<m/250+1;jj++)
{
  for (i=0; i < m; i++)
   {
    for (j=jj*250; j < min((jj+1)*250,m);j++)
      {
         c[i] += a[i][j]*b[j];
      }
     }
}
```

Note: It is not always safe to interchange the iteration variables due to dependencies between statements for the order they execute. This safety check will be performed by the compiler!

Optimization Notice
Copyright © 2015, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Legal Disclaimer & Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Copyright © 2015, Intel Corporation. All rights reserved. Intel, Pentium, Xeon, Xeon Phi, Core, VTune, Cilk, and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

Copyright © 2015, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.