Intel® MKL Data Fitting component. Overview

Andrey Nikolaev

Intel

Agenda

- 1D interpolation problem statement
- Functional decomposition of the problem
- Application areas
- Data Fitting in Intel® MKL
- Data Fitting API and usage models
- Data Fitting performance

1D interpolation problem statement

For given table function {x(i), y(i)}, i=1,…,n where
x(i) - breakpoints in ascending order, y(i) – values

- Approximate function f(x): f(x(i))=y(i)

- Evaluate value f(t(j)) and derivative f' (t(j))
  - Site t(j) is any real value [x(1),x(n)], j=1,…,m

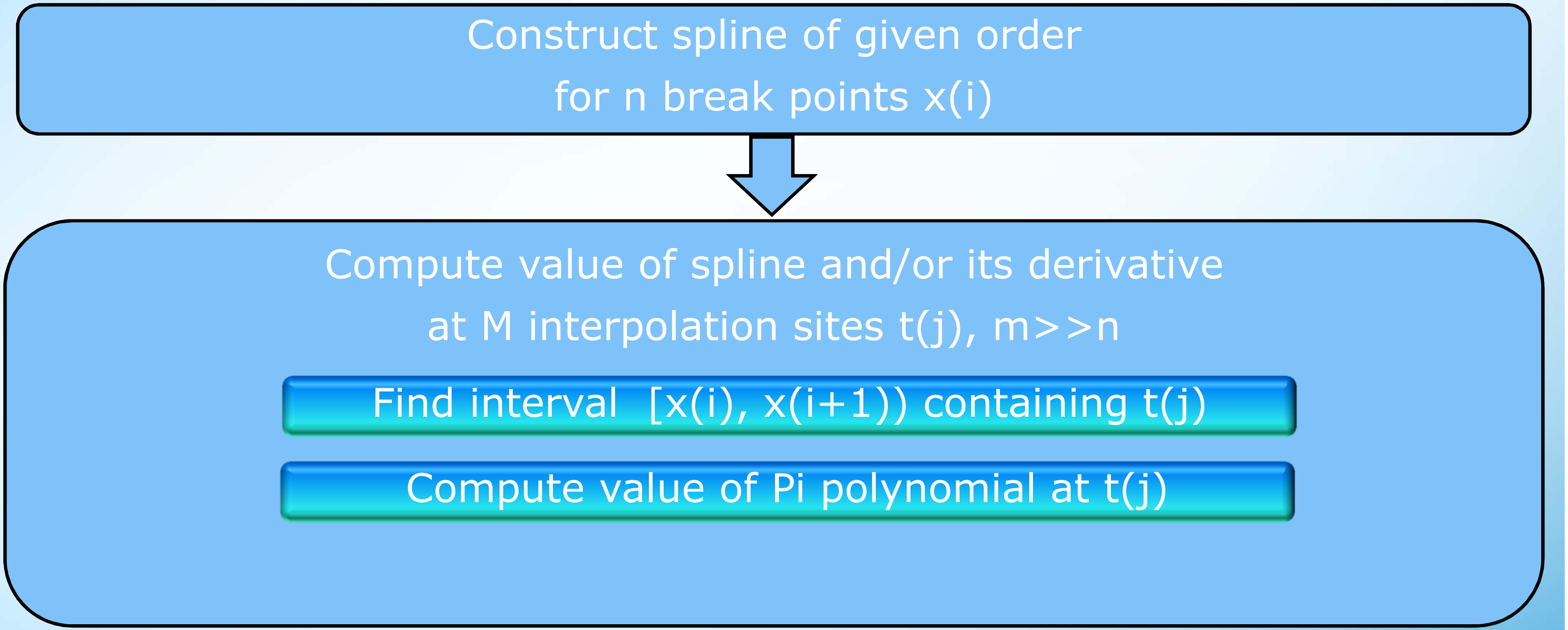
- Evaluate integral of f(x) over interval [a(j),b(j))
  - Integration limits a(j) and b(j) belong to or are outside of interpolation interval [x(1),x(n)], j=1,…,m

1D interpolation problem statement

- **Splines – methodology for solution of interpolation problem**
  - Can be preferable vs polynomial interpolation
    - Runge’s phenomenon
      - Interpolation error for g(x)=1/(1+25x^2) increases with order of the polynomial

- **Spline – piece-wise polynomial function**
  - g(x) := Pj(x), x belongs to [x(j), x(j+1))
    - Pj(x) -polynomial of degree k on the interval [x(j), x(j+1))
  - Spline - smooth up to order q at x(j) if values of derivatives up to order q for P(j-1) and Pj at x(j) exist and equal

Functional decomposition of the problem



Construct spline of given order
for n break points x(i)

Compute value of spline and/or its derivative
at M interpolation sites t(j), m>>n

Find interval [x(i), x(i+1)) containing t(j)

Compute value of Pi polynomial at t(j)

Integration has similar computational flow

Search – key building block

Application areas

- **Data analysis and analytics**
  - Approximation of statistical estimates like histogram
- **Manufacturing**
  - Geometrical modeling
  - *"B-spline recurrence relations … were used at Boeing, …, five hundred million times a day"* Carl de Boor, On Wings of Splines Newsletter of Institute for Mathematical Sciences, ISSUE 5 2004
- **Energy**
  - Surface approximation
- **ISV**
  - Software libraries

Data Fitting in Intel® MKL

- **Intel® MKL Data Fitting – SW solution for**
  - Spline construction
  - Spline based interpolation and computation of derivatives
  - Spline based integration
  - Cell Search

Data Fitting in Intel® MKL

- **Components of Intel® MKL Data Fitting**

Spline construction

Spline	Spline type
Linear	
Quadratic	Default, Subbotin
Cubic	Default, Natural, Hermite, Bessel, Akima
Look-up	
Stepwise constant	Continuous-right, Continuous-left
User-defined	

Boundary conditions	Internal conditions
Not-a-knot	1st derivative
Free-end	2nd derivative
1st derivative at the left/right endpoint	Knot array
2nd derivative at the left/right endpoint	
Periodic	
Function value at mid point of first cell	

Rich collection of splines that support different boundary or/and internal conditions

Data Fitting in Intel® MKL

- **Components of Intel® MKL Data Fitting**

Interpolation/extrapolation/integration

Feature	Comment
Computation of value, derivative of arbitrary order	• Support of a-priori information about structure of partition, and/or interpolation sites • In addition to default spline based interpolation library supports user-defined functions to • re-define default spline based computations on interpolation or/and extrapolation intervals • re-define cell search functions • Option to get results of cell search simultaneously with interpolation • User defined threading-friendly API
Computation of integrals	• Support of a-priori info about structure of partition, and/or integration limits • In addition to default spline based interpolation library supports user-defined functions to • re-define default integration on interpolation or/and extrapolation intervals • re-define cell search functions • User defined threading-friendly API

Flexibility in support of various usage models for spline based computations

Data Fitting in Intel® MKL

- **Components of Intel® MKL Data Fitting**

Search

Feature	Comment
Computation of cell indices containing given sites	• Support of a-priori information about structure of partition, and/or interpolation sites • In addition to default cell search computation library supports user-defined function to • re-define cell search functions • User defined threading-friendly API

Flexibility in support of various usage models for cell search

Data Fitting API and usage models

Step	Code example	Comment
Create a task	`status = dfdNewTask1D( &task, nx, x, xhint, ny, y, yhint );`	You can call the Data Fitting function several times to create multiple tasks
Modify the task parameters.	`status = dfdEditPPSpline1D( task, s_order, c_type, bc_type, bc, ic_type, ic, scoeff, scoeffhint );`	
Perform Data Fitting spline-based computations	`status = dfdInterpolate1D(task, estimate, method, nsite, site, sitehint, ndorder, dorder, datahint, r, rhint, cell );`	You may reiterate steps 2-3 as needed
Destroy the task or tasks	`status = dfDeleteTask( &task );`	

API and usage model similar to that in Vector Statistical component, Fourier Transforms in Intel® MKL

Optimization Notice

Data Fitting API and usage models

Cubic Spline-Based Interpolation

```
#include "mkl.h‟

int main(){
    /* Initialize the partition and set their values */
    nx = N;
    xhint = DF_NON_UNIFORM_PARTITION /* The partition is non-uniform. */
    /* Initialize the function and set their values */
    ny = 1; /* The function is scalar. */
    yhint = DF_NO_HINT; /* No additional information about the function is provided. */
    /* Create a Data Fitting task */
    status = dfdNewTask1D( &task, nx, x, xhint, ny, y, yhint );

    /* Initialize spline parameters */
    s_order = DF_PP_CUBIC; /* Spline is of the fourth order (cubic spline). */
    s_type = DF_PP_BESSEL; /* Spline is of the Bessel cubic type. */

    ic_type = DF_NO_IC; ic = NULL; /* Define internal conditions for cubic spline construction (none in this example) */
    bc_type = DF_BC_NOT_A_KNOT; bc = NULL; /* Use not-a-knot boundary conditions */
    scoeffhint = DF_NO_HINT; /* No additional information about the spline. */

    /* Set spline parameters in the Data Fitting task */
    status = dfdEditPPSpline1D( task, s_order, s_type, bc_type, bc, ic_type, ic, scoeff, scoeffhint );
    /* Construct a cubic Bessel spline: Pi(x) = c1,i + c2,i(x – xi) + c3,i(x – xi)^2 + c4,i(x – xi)^3; the library packs spline
        coefficients to scoeff: scoeff[4*i+0] = c1,i, scoef[4*i+1] = c2,i,  scoeff[4*i+2] = c3,i, scoef[4*i+1] = c4,i */
    status = dfdConstruct1D( task, DF_PP_SPLINE, DF_METHOD_STD );

    /* Initialize interpolation parameters and set site values */
    nsite = NSITE;
    sitehint = DF_NON_UNIFORM_PARTITION; /* Partition of sites is non-uniform */
    ndorder = 1; dorder = 1; /* Request to compute spline values */
    datahint = DF_NO_APRIORI_INFO;  /* No additional information about breakpoints or sites is provided. */
    rhint = DF_MATRIX_STORAGE_ROWS; /* The library packs interpolation results in row-major format. */
    cell = NULL; /* Cell indices are not required. */
    /* Compute the sline values at the points site(i), i=0,..., nsite-1 and place the results to array r */
    status = dfdInterpolate1D(task, DF_INTERP, DF_METHOD_STD,nsite, site, sitehint, ndorder, &dorder, datahint, r, rhint, cell );

    /* De-allocate Data Fitting task resources */
    status = dfDeleteTask( &task );
    return 0;
}
```

Optimization Notice

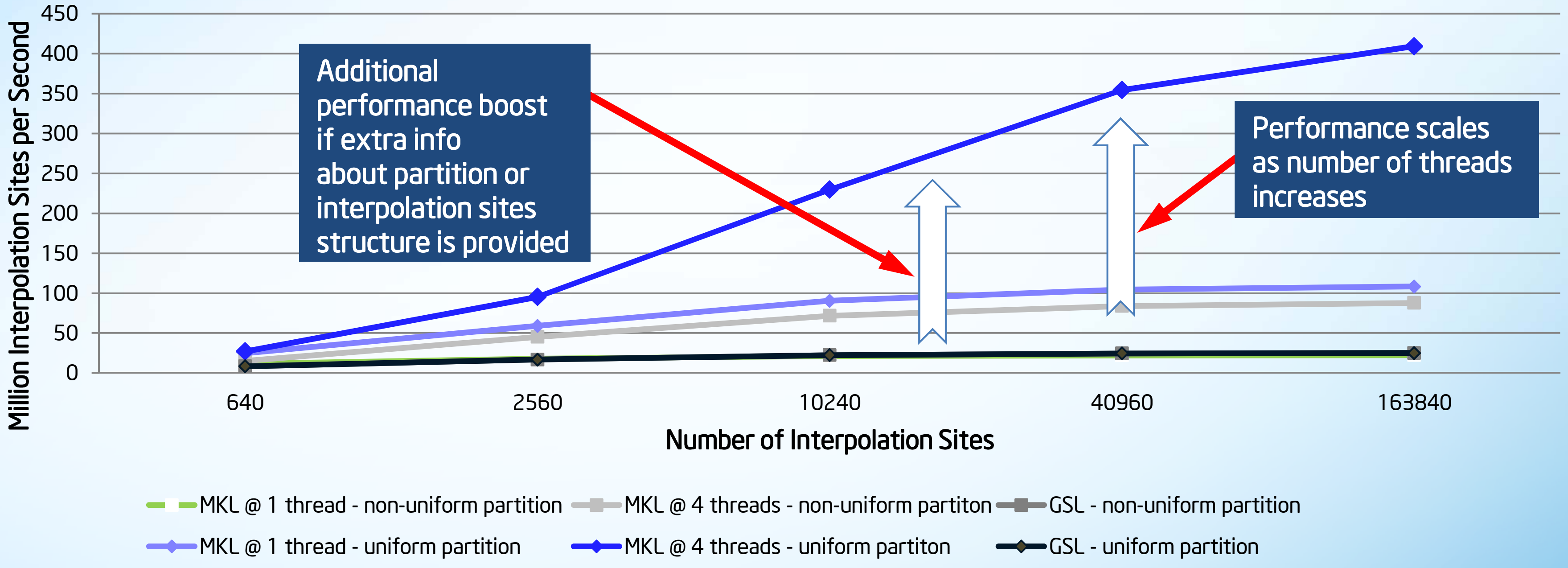
Data Fitting API and usage models

Cell Search

```
#include "mkl.h“

int main(){

   /* Initialize a uniform partition */
   nx = N;
   /* Set values of partition x: for uniform partition, provide end-points of the interpolation interval [-1.0,1.0] */
   x[0] = -1.0f; x[1] = 1.0f;
   xhint = DF_UNIFORM_PARTITION; /* Partition is uniform */

   /* Initialize function parameters; in cell search, function values are not necessary and are set to zero/NULL values
*/
   ny = 0;
   y = NULL;
   yhint = DF_NO_HINT;

   /* Create a Data Fitting task */
   status = dfdNewTask1D( &task, nx, x, xhint, ny, y, yhint );
   ...
   /* Initialize interpolation (cell search) parameters */
   nsite = NSITE;
   /* Set sites in the ascending order */
   ...
   sitehint = DF_SORTED_DATA; /* Sites are provided in the ascending order. */
   datahint = DF_NO_APRIORI_INFO; /* No additional information about breakpoints/sites is provided.*/

   /* Compute indices of the cells that contain interpolation sites. The library places the index of the cell containing
      site(i) to the cell(i), i=0,...,nsite-1 */
   status = dfsSearchCell1D( task, DF_METHOD_STD, nsite, site, sitehint, datahint, cell );

   /* Process cell indices */
   status = dfDeleteTask( &task );
   return 0;
}
```

Data Fitting performance

Data Fitting Performance Improvements using Intel® Math Kernel Library versus GSL* Spline Construction and Interpolation



Additional performance boost if extra info about partition or interpolation sites structure is provided

Performance scales as number of threads increases

Construction of natural cubic spline with free end boundary conditions for function defined on uniform and non-uniform partitions. Partition size is 1280. Spline-based values and first derivatives are computed.

Configuration Info - Versions: Intel® Math Kernel Library (Intel® MKL) 10.3.8 GSL 1.15; Hardware: Intel® Core® i7-2600 Processor, 3.40Ghz, 8 MB L2 cache, 4 GB Memory; Operating System: Fedora 14 x86_64; Benchmark Source: Intel Corporation.

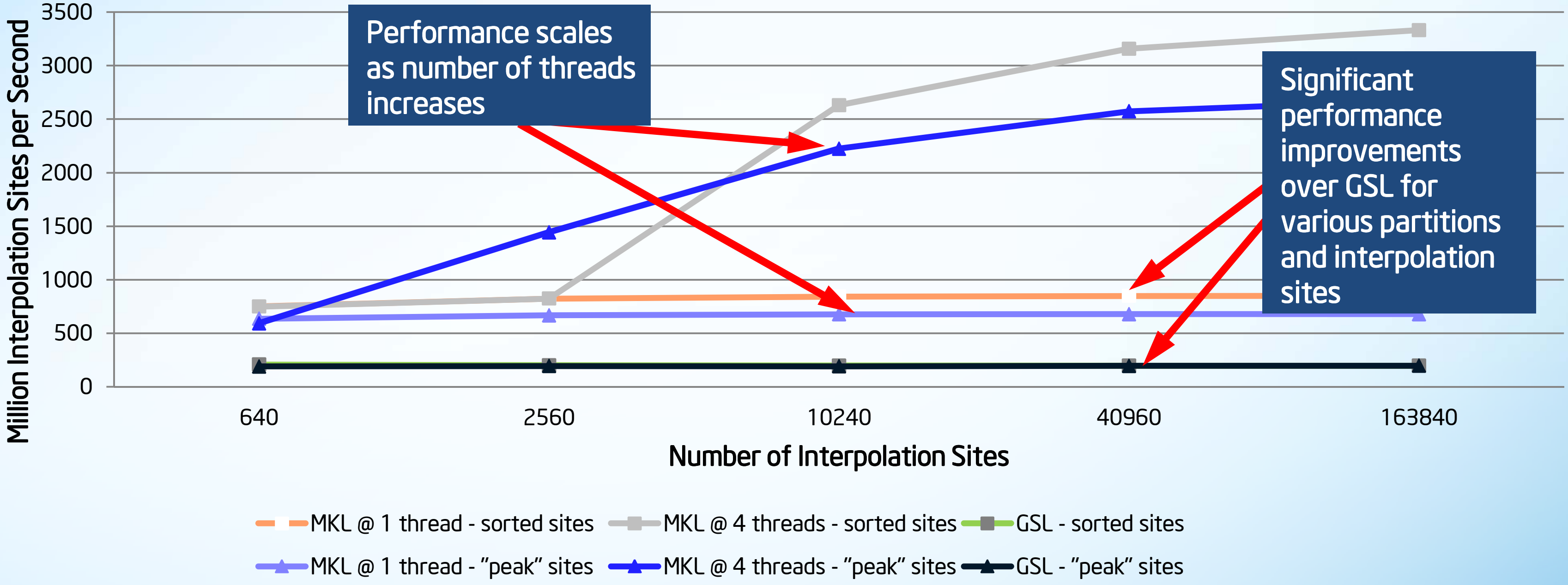
Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, refer to www.intel.com/performance/resources/benchmark_limitations.htm.

Refer to our Optimization Notice for more information regarding performance and optimization choices in Intel software products at : http://software.intel.com/en-us/articles/optimization-notice/.

* Other brands and names are the property of their respective owners

Data Fitting performance

Data Fitting Performance Improvements
using Intel® Math Kernel Library versus GSL*
Cell Search



Performance scales as number of threads increases

Significant performance improvements over GSL for various partitions and interpolation sites

Performing cells search on non-uniform partition. Partition size is 1280.
Sorted sites - interpolation sites are sorted ; "peak" sites - distribution of interpolation sites has a clear peak.

Configuration Info - Versions: Intel® Math Kernel Library (Intel® MKL) 10.3.8 GSL 1.15; Hardware: Intel® Core® i7-2600 Processor, 3.40Ghz, 8 MB L2 cache, 4 GB Memory; Operating System: Fedora 14 x86_64; Benchmark Source: Intel Corporation.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, refer to www.intel.com/performance/resources/benchmark_limitations.htm.

Refer to our Optimization Notice for more information regarding performance and optimization choices in Intel software products at : http://software.intel.com/en-us/articles/optimization-notice/.

* Other brands and names are the property of their respective owners

intel®

Software

Optimization Notice

Optimization Notice

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

Legal Disclaimer

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.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, reference www.intel.com/software/products.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Atom, Centrino Atom Inside, Centrino Inside, Centrino logo, Cilk, Core Inside, FlashFile, i960, InstantIP, Intel, the Intel logo, Intel386, Intel486, IntelDX2, IntelDX4, IntelSX2, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium Inside, skoool, Sound Mark, The Journey Inside, Viiv Inside, vPro Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

http://intel.com/software/products