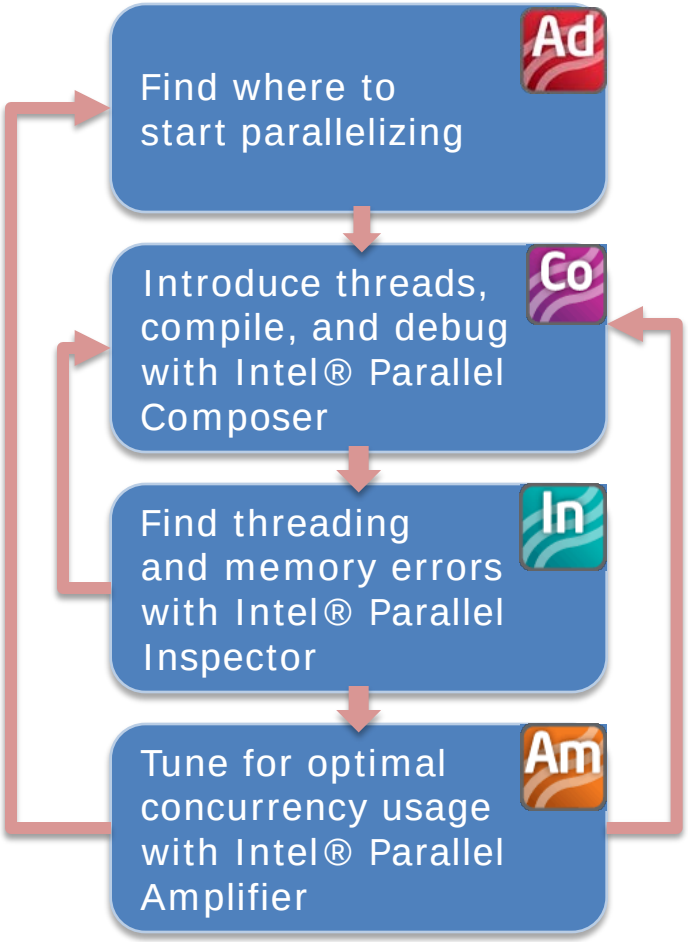
Image Processing: Stop Developing Code From Scratch

Walt Shands

Four Steps to a Parallel Application



DESIGN – Intel® Parallel Advisor Lite
Gain insight on where parallelism will most benefit existing source code – usually begins with a "hotspot"

CODE, DEBUG – Intel® Parallel Composer
Develop effective applications with a C/C++ compiler and comprehensive threaded libraries and API's, and a parallel debugger

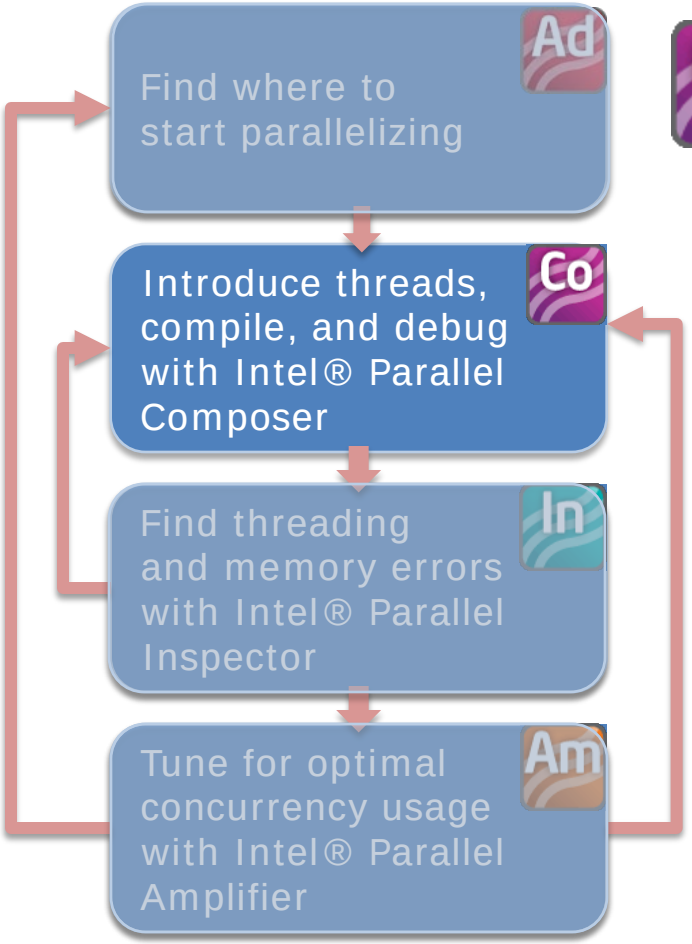
VERIFY – Intel® Parallel Inspector
Help ensure application reliability with proactive parallel memory and threading error checking

TUNE – Intel® Parallel Amplifier
Enhance applications with an intuitive performance analyzer and tuner

Intel Parallel Studio is now available.
Get free eval software:
intel.com/software/prodcuts/eval

www.intel.com/go/parallel

Intel® Parallel Composer



CODE & DEBUG PHASE

Add parallelism to Windows* applications more quickly, with better scaling!

- Highly optimizing C/C++ compiler with built-in parallelism features:
  - OpenMP 3.0*
  - Intel® Threading Building Blocks
  - Intel® Integrated Performance Primitives
  - Parallel Debugger Extension

Increase productivity in implementing parallelism

Intel Parallel Studio is now available.
Get free eval software:
intel.com/software/prodcuts/eval

www.intel.com/go/parallel

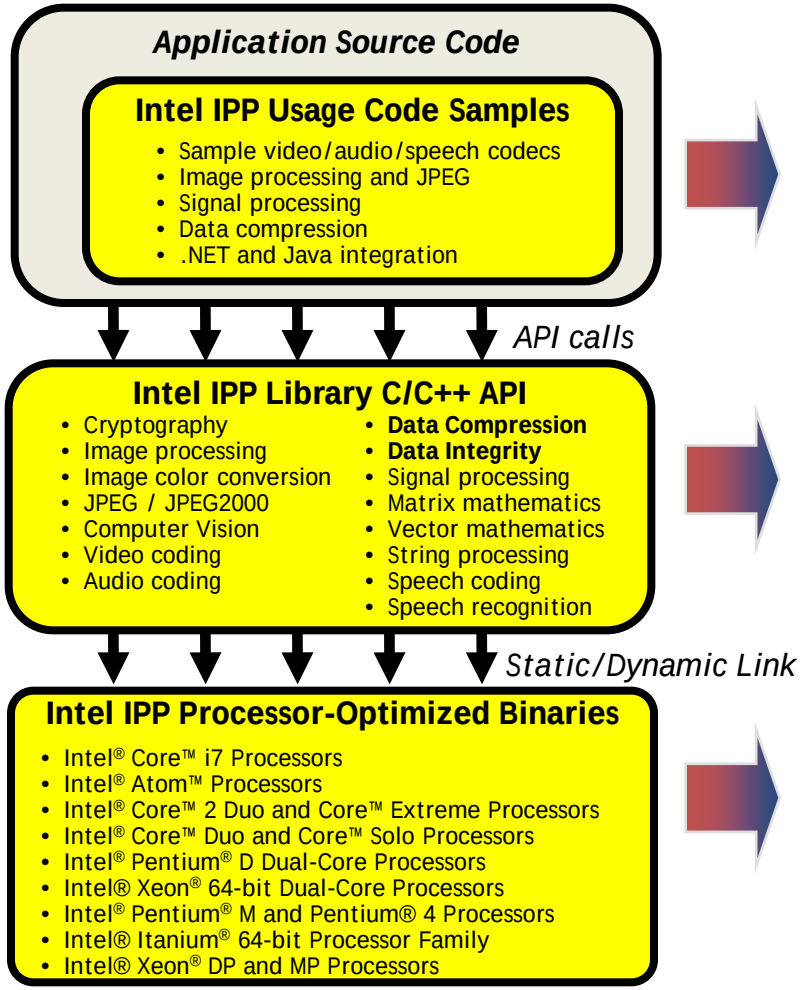
Intel Software Products

Intel® Integrated Performance Primitives (Intel® IPP) — Overview and Benefits

Free Code Samples

Cross-platform API

Processor-Optimized Implementation



Rapid Application Development

Compatibility and Code Re-Use

Outstanding Performance

Intel Software Products

Intel® IPP

Industry Leading Performance

- Efficient parallelism on multicore platforms
- Instruction set-level optimizations
- Intel-compatible platforms

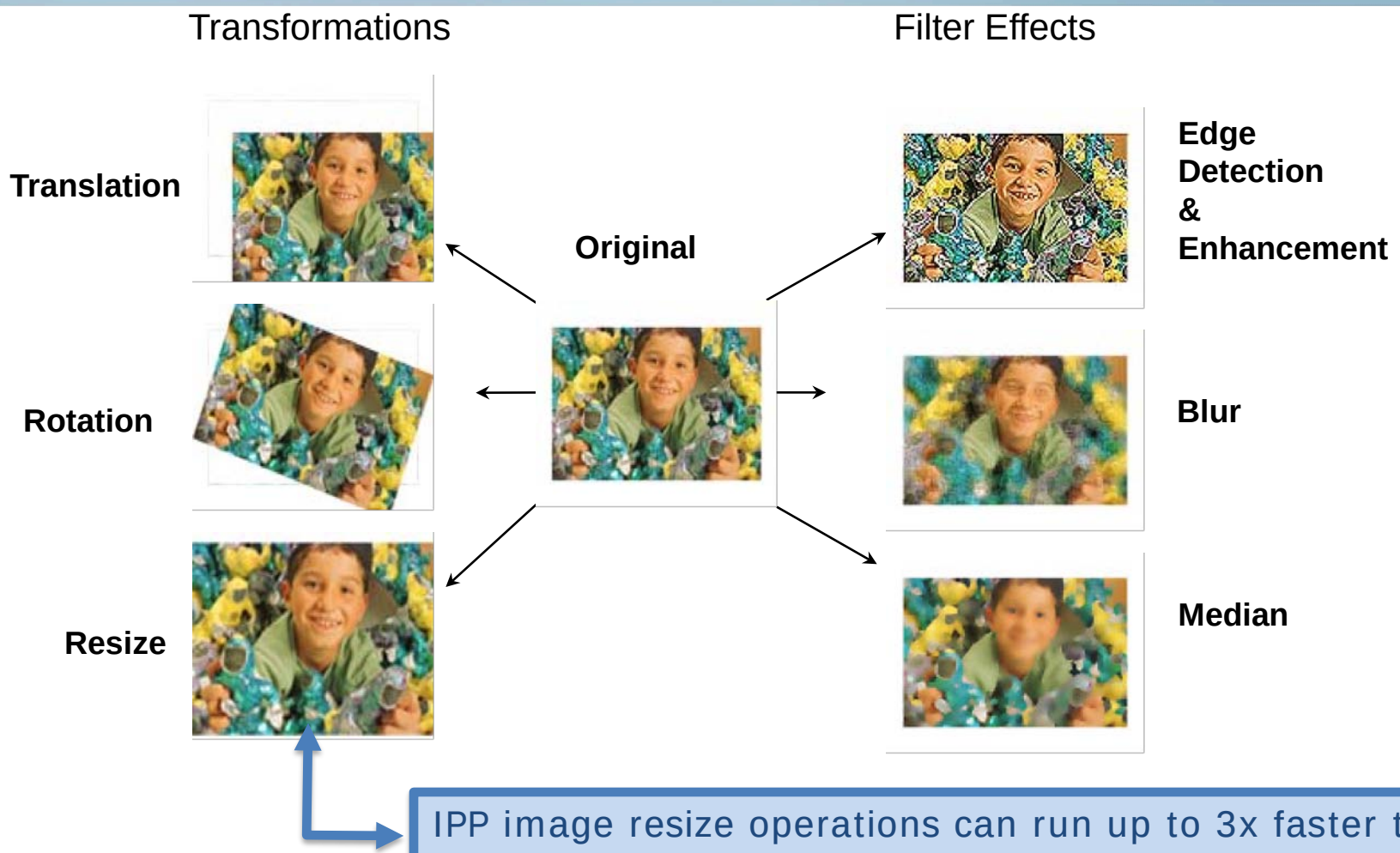
Enhances Developer Productivity

- Optimized and future-scaling multimedia, signal, and data processing routines
- Comprehensive high level APIs and code samples for ease of implementation
- Utilize one performance library for 32 & 64 bit Windows*, Linux* and Mac OS

Future Proof

- Optimized for current multicore and future manycore processors
- Ensures that applications benefit seamlessly from the latest architecture enhancements

Intel® Integrated Performance Primitives - Image Processing Examples



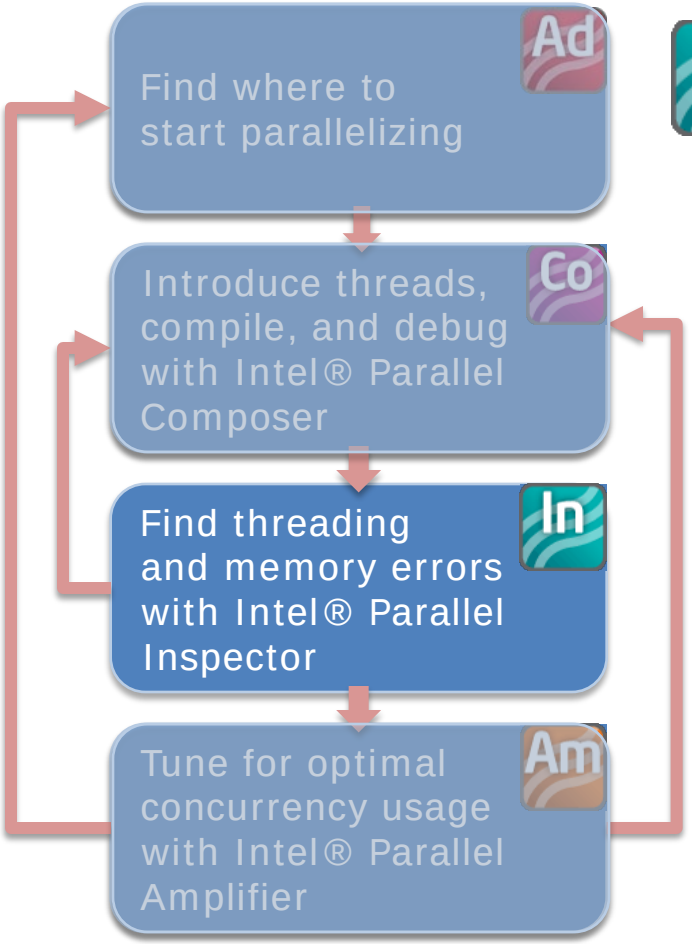
IPP image resize operations can run up to 3x faster than compiled C++ code on multicore systems.

Image Processing Functions

Image processing: Intel IPP is a premier library of image processing algorithms, and includes a rich selection of functions operating on images and regions within images

Transforms: Wavelet Fourier (FFT/DFT, real/complex) Windowing (Hamming, Bartlett) Discrete Cosine (DCT) **Filtering Functions:** General linear filtering Convolution/Deconvolution (LR and FFT) Box, Min, Max, Median Wiener Filters Fixed Filters (Prewitt, Sobel, Laplace, Gauss, Scharr, Roberts) Sharpening/Hipass/Lowpass Filters **Geometric Transforms:** Resize, Mirror, Rotate, Shear Affine transforms Perspective transformations Bilinear warping Coordinate remapping	**Image Statistics:** Sum, Integral, Tilted Integral Mean, Min, Max, Histogram, StDev Image Moments Image Norms (L1, L2, infinity) Image Quality Index calculation Proximity Measures (Cross-correlations, Square Distance) Threshold/Compare Operations **Image Arithmetic/Logic Operations:** Alpha composition Arithmetic operations (add/sub/mul/div/sqrt/sqr/ln/exp/abs) Logical operations (AND, OR, XOR, Shift, NOT) **Image Data Exchange/Initialization:** Copy/Set/Transpose Channel swapping Jaehne/Ramp/Zigzag initialization Memory allocation for multiple image types

Intel® Parallel Inspector



CODE CHECKING PHASE

Ensure parallel application reliability!

- Memory Error Detection
- Threaded Error Detection

Help ensure application reliability with proactive parallel memory and threading error checking

Intel Parallel Studio is now available.
Get free eval software:
intel.com/software/prodcuts/eval

www.intel.com/go/parallel

Intel® Parallel Inspector

Memory Error Detection

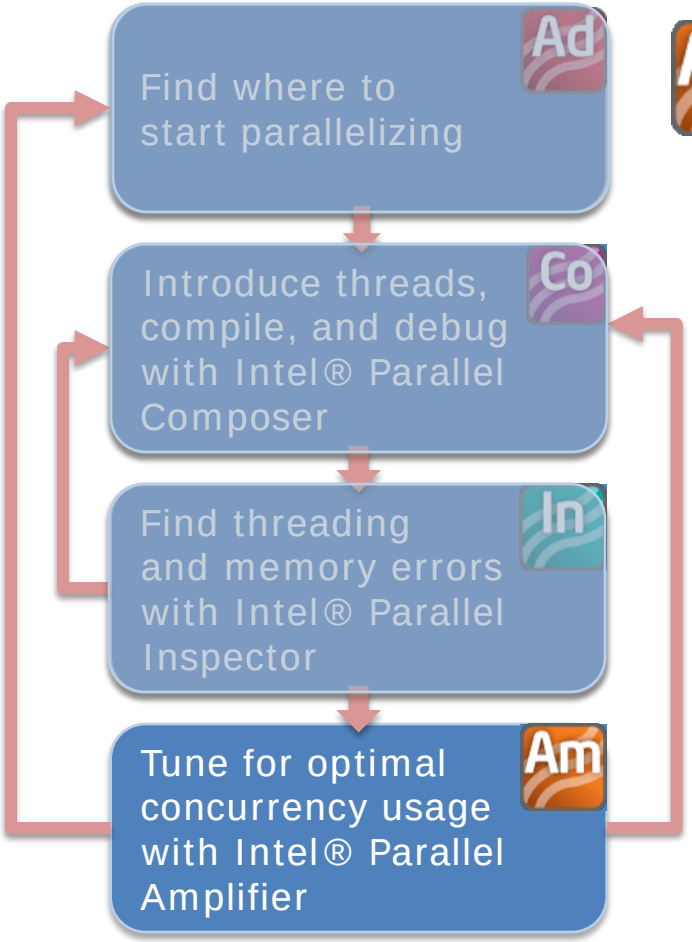
- Memory Leaks, i.e., allocation errors
- Memory Corruption, i.e., crashes, etc.
- Uninitialized Memory Accesses
- Dangling pointers
- Buffer Overflows, Stack Overflows
- Depth of error checking can be configured

Thread Error Detection

- Data race errors
- Deadlocks
- Finds latent (or likely to occur) errors and maps them to the source-code line, call stack, and memory reference
- Displays useful warnings for effective diagnosis, highlighting potentially severe errors

Works on standard debug builds

Intel® Parallel Amplifier



TUNING PHASE

Tune for multi-core performance scalability!

- **Hotspot Analysis**
  Where is my program spending time running?
- **Concurrency Analysis**
  Where is my program not concurrent?
- **Lock/Wait Analysis**
  Where is my program waiting on Sync or I/O?

Intel Parallel Studio is now available.
Get free eval software:
intel.com/software/prodcuts/eval

www.intel.com/go/parallel

Intel® Parallel Amplifier

Hotspot Analysis

- Where is my program spending time running?
  And how it got there
  Identify Hotspot functions

Concurrency Analysis

- Where is my program not concurrent?
  Useful when moving serial code to parallel
  Identify processor cores utilization

Lock/Wait Analysis

- Where is my program waiting on Sync or I/O?
  Identify locking problems that slow threaded software
  Identify objects limiting parallelism

Source View

- Shows data so it correlates with your code

Statistical Call Tree

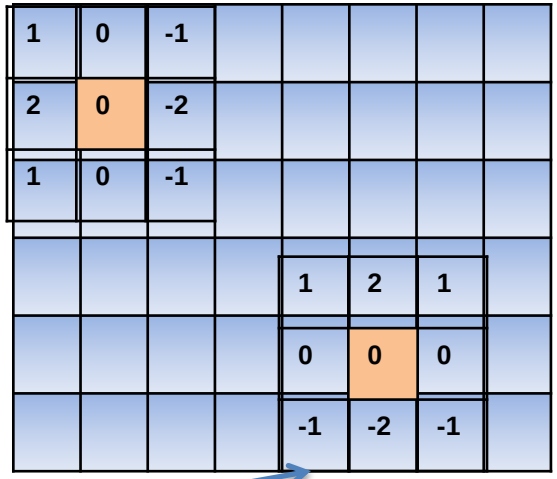
- Helps catch call path most impacting performance

Compare Multiple Runs

- Quickly see the impact of your changes

Sobel Edge Detection

Apply vertical edge filter to each pixel to find vertical edges



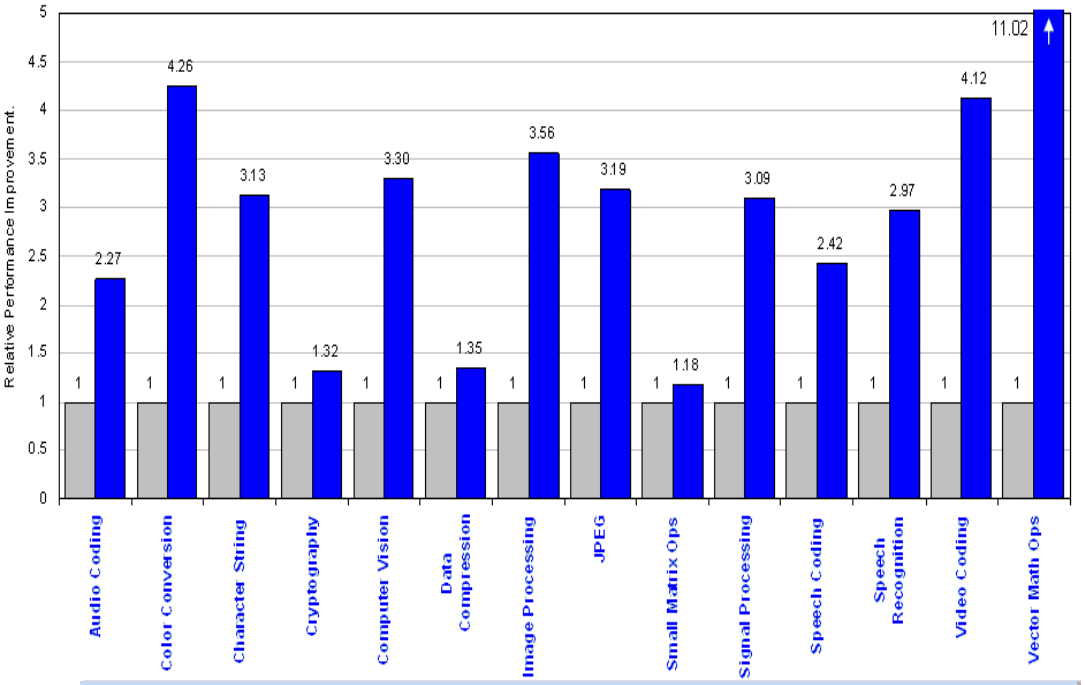
Apply horizontal edge filter to each pixel to find horizontal edges

Find vertical and horizontal edges

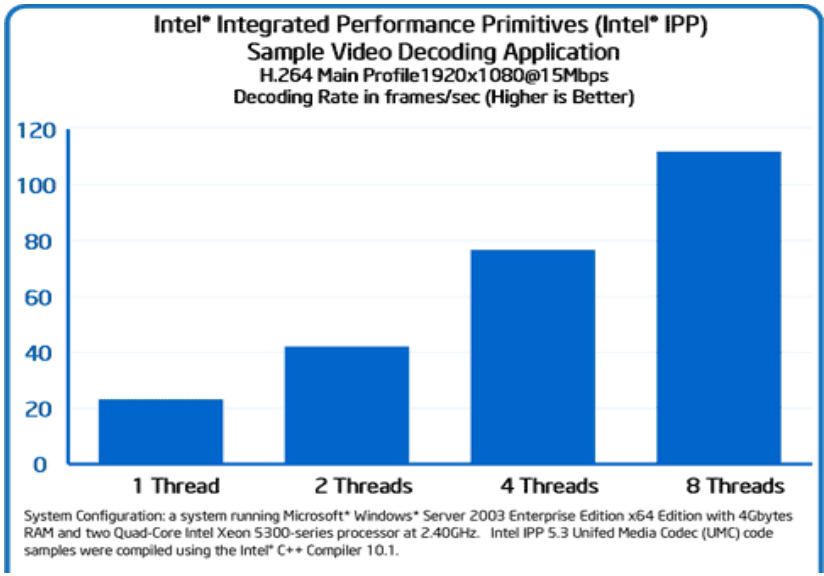
Intel® Integrated Performance Primitives (IPP)

Intel IPP vs. C on single processor

- 200% faster (average over all domains)
- Optimized C performance normalized to 1



System configuration: Intel® Xeon® 4 Processor, 2.8GHz, 2GB using Windows* XP

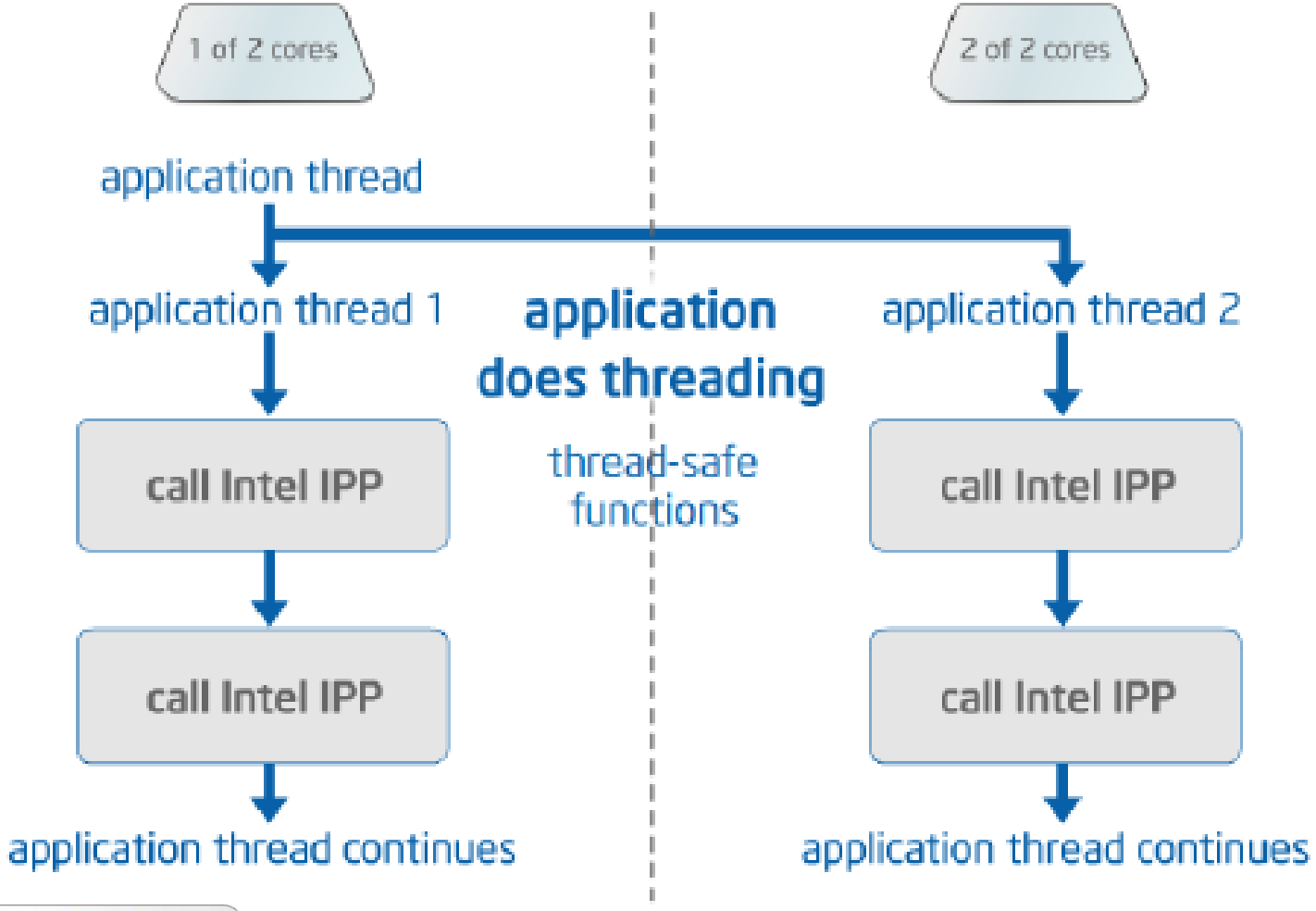


Multicore performance scaling example: H.264 decode

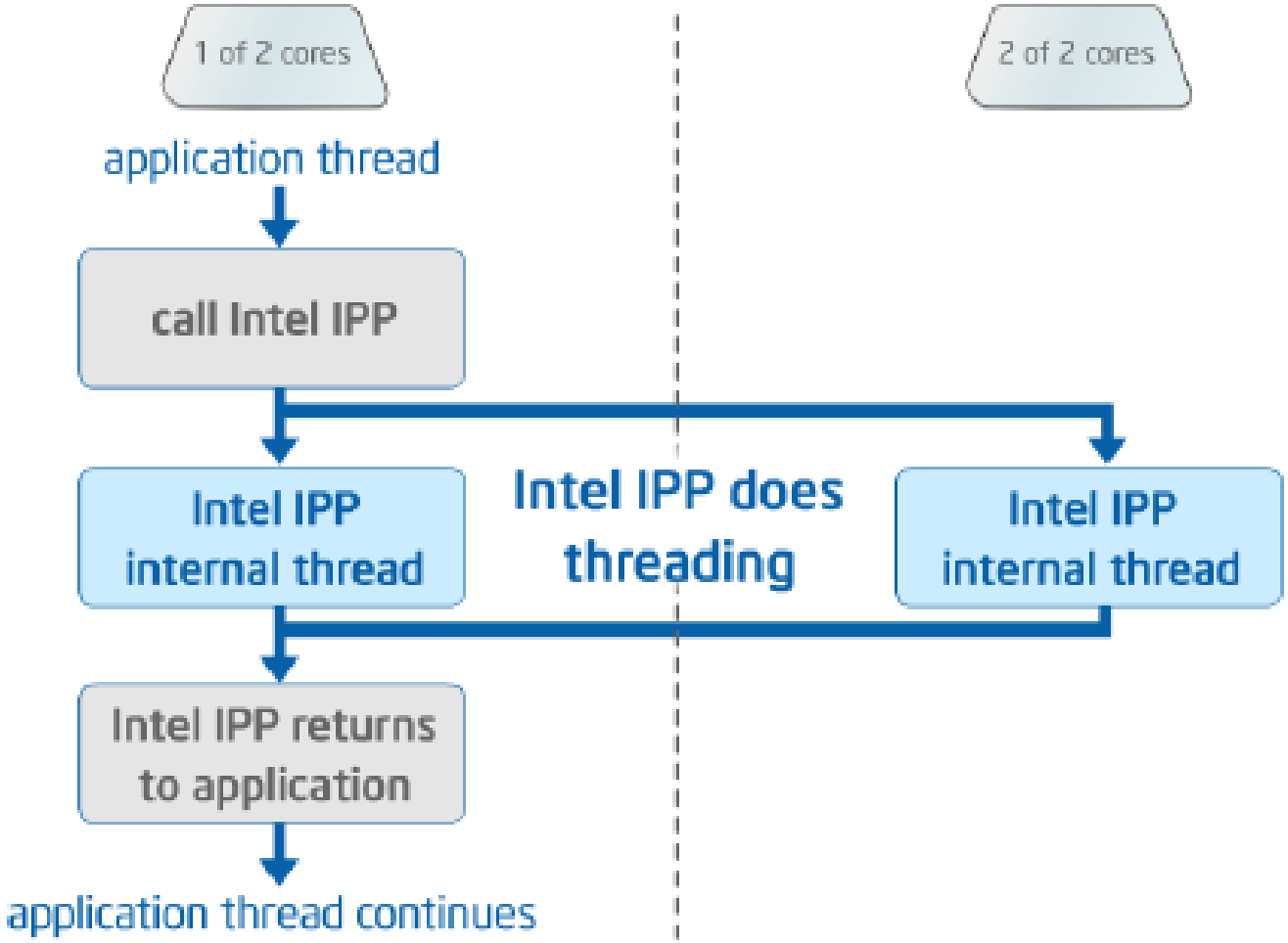
Source: Intel Corporation. Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. 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.

Intel Software Products

Threading In Application



Threading Inside Intel IPP



Intel® IPP Functions and Code Samples: Image Processing/Coding

Domain	Functions	Samples
Image Processing	* Geometry transformations, such as resize/rotate * Linear and non-linear filtering operation on an image for edge detection, blurring, noise removal and etc for filter effect. * Linear transforms for 2D FFTs, DFTs, DCT. * image statistics and analysis	* Tiled Image Processing / 2D Wavelet Transform /C++ Image Processing Classes/Image Processing functions Demo
Computer Vision	* Background differencing, Feature Detection (Corner Detection, Canny Edge detection), Distance Transforms, Image Gradients, Flood fill, Motion analysis and Object Tracking, Pyramids, Pattern recognition, Camera Calibration	* Face Detection
Color Models	* Convert image/video color space formats: RGB, HSV, YUV, YCbCr * Up/Down sampling * Brightness and contrast adjustments	
JPEG Coding	* High-level JPEG and JPEG2000 compression and decompression functions * JPEG/JPEG2000 support functions: DCT, Wavelet transforms, color conversion, downsampling	* Integration with the Intel® JPEG Library (IJL) / Integration with the Independent JPEG Group (IJG) library /JPEG2000 encoder/decoder /JPEG viewer
Realistic Rendering	* Acceleration Structures, Ray-Scene Intersection and Ray Tracing * Surface properties, shader support, tone mapping	* Ray Tracing

Intel® IPP Functions and Code Samples: Video/Audio/Speech

Domain	Functions	Samples
Video Coding	* VC-1, H.264, MPEG-2, MPEG-4, H.261, H.263 and DV codec support functions	* Simple Media Player/ Video Encoder / h.264/DV decoding/video transcoder /Reverberation Demo/ Virtual Multi-Channel Audio Player/
Audio Coding	* Echo cancellation and audio transcoding, BlockFiltering, Spectral Data prequantization.	* Audio Codec Console application
Speech Coding	* Adaptive/Fixed Codebook functions, Autocorrelation, Convolution, Levinson-Durbin recursion, Linear Prediction Analysis & Quantization, Echo Cancellation, Companding	* G.168, G.167, G.711, G.722, G.722.1, G.722.2, AMRWB, Extended AMRWB (AMRWB+), G.723.1, G.726, G.728, G.729, RT-Audio, GSM AMR, GSM FR
Speech Recognition	* Feature Processing, Model Evaluation/Estimation/Adaptation, Vector Quantization, Polyphase Resampling, Advanced Aurora, Ephraim-Malah Noise Supression, AEC, Voice Detection	* Aurora, Advanced Aurora, Audio Processing, Gaussian Mixture, Speech Processing
Signal Processing	* Transforms: DCT, DFT, MDCT, Wavelet (both Haar and user-defined filter banks), Hilbert * Convolution, Cross-Correlation, Auto-Correlation, Conjugate * Filtering: IIR/FIR/Median filtering, Single/Multi-Rate FIR LMS filters * Other: Windowing, Jaehne/Tone/Triangle signal generation, Thresholding	* Signal Processing Function Demo

Intel® IPP Functions and Code Samples: Data Processing/Compression

Domain	Functions	Samples
Data Compression	* Entropy-coding compression: Huffman, VLC * Dictionary-based compression: LZSS, LZ77 * Burrows-Wheeler Transform (BWT), MoveToFront (MTF), Run-Length-Encoding (RLE), Generalized Interval Transformation (GIT) * Compatible feature support for zlib and bzip2	* zlib, bzip2, gzip-compatible /General data compression examples
Cryptography	* Big-Number Arithmetic / Rijndael, DES, TDES, SHA1, MD5, RSA, DSA, Montgomery, prime number generation and pseudo-random number generation (PRNG) functions	* Intel IPP crypto usage in Open SSL*
String Processing	* Compare, Insert, change case, Trim, Find, Regexp, Hash	* “ippgrep” – regular expression matching
Vector Math	* Logical, Shift, Conversion,Power, Root, Exponential, Logarithmic, Trigonometric, Hyperbolic, Erf, Erfc	
Matrix Math	* Addition, Multiplication, Decomposition, Eigenvalues, Cross-product, transposition	
Common Functions	* CPUTypes, Thread number control, Memory Allocation	* Linkages/Different language support

Deferred Mode Image Processing

Image Processing Task using a Sequence of Basic Operations

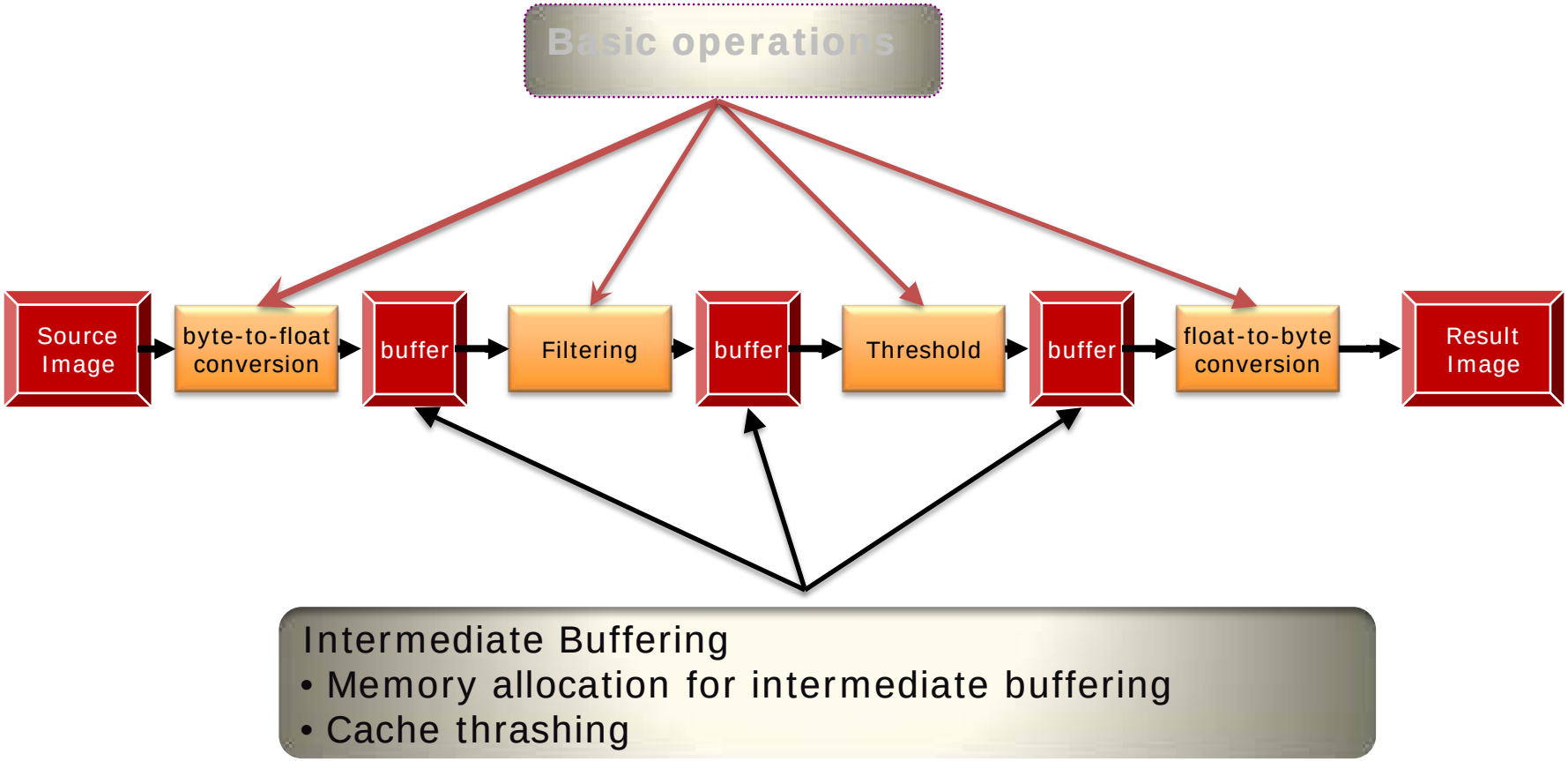
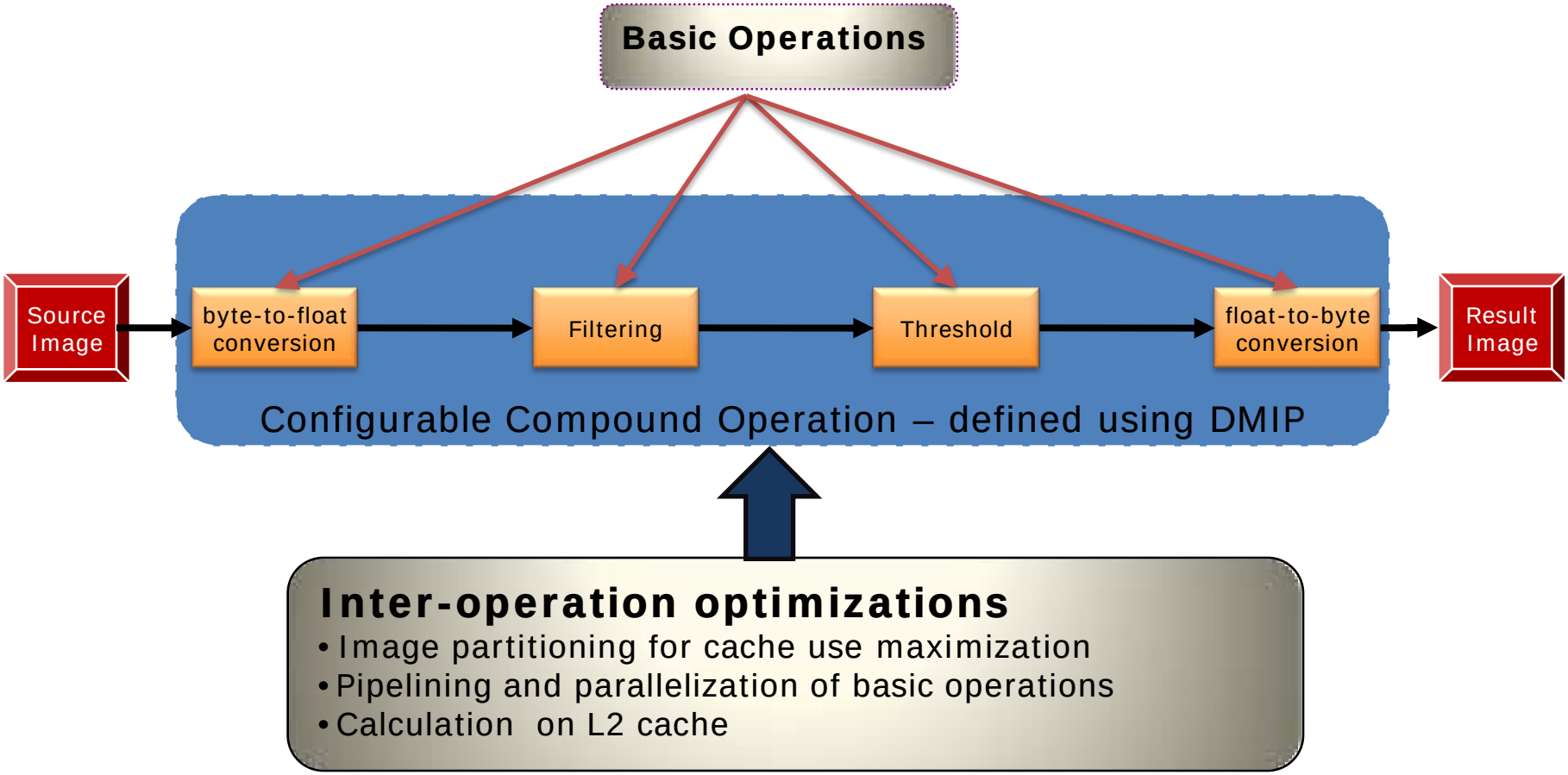


Image Processing Task, using Deferred Mode Image Processing (DMIP)



Intel's Deferred Mode Image Processing (DMIP)

- DMIP provides a mechanism to chain multiple image processing operations together
- DMIP performs scan line based processing to better utilize cache behavior
- DMIP shows significant performance benefit for large images
- Provided in Intel® IPP

DMIP Essentials

- Intuitive programming layer based on C++
  - .NET interface also supported

- Performance benefits
  - Utilize existing optimized IPP functions
  - Utilize parallel processing on multi-core platforms
  - Perform scan line based processing for better cache utilization

- Easy to extend with user defined operations

Sobel Edge Detector

DMIP implementation

```
Image A(s,Ipp8u,IppC3,roi,sStep)
Image D(d,Ipp8u,IppC1,roi,dStep)
Kernel KH(Horiz,k3x3,ipp8u,ipp16s)
Kernel KV(Vert,k3x3,ipp8u,ipp16s)
Graph G = ColorToGray(*A)
D = To8u(Abs(G*KH) + Abs(G*KV))
```

Upcoming webinars…

Live Date	Event	Title	Speaker
3/10/2009	Webinar	Go-Parallelism! Ease the Onramp for C/C++ Windows* Development	James Reinders
3/17/2009	Tech. Session	Solve Parallelism with Intel® Parallel Studio	Joe Wolf
3/24/2009	Webinar	Simplify Parallelism with Intel® Parallel Composer	Joe Wolf
3/31/2009	Tech. Session	Parallel Implementation Methods with Intel® Parallel Composer	Ganesh Rao
4/7/2009	Webinar	Debugging Parallel Code for Fast, Reliable Applications	Jay Desouza
4/14/2009	Tech. Session	Find Errors in Windows C++ Parallel Applications	Gerold Mueller or Robert Mueller
4/21/2009	Webinar	Easy Ways to Solve Parallel Performance Challenges	Gary Carleton
4/28/2009	Tech. Session	The Good, the Bad, and the Ugly: Improve Parallel Application Quality and Performance	Eric Moore
5/5/2009	Webinar	The Key to Scaling Applications for Multicore	Paul Petersen/Mark Davis
5/12/2009	Tech. Session	Identify and Address Threading Opportunities	Caroline Davidson
5/19/2009	Webinar	Image Processing: Stop Developing Code from Scratch	Walt Shands, Intel Corporation
5/26/009	Tech. Session	Simplifying Parallelism Implementation with Intel® Threading Building Blocks	Mike D'Mello
6/2/2009	Tech. Session	Static Analysis and Intel® C++ Compilers	Dmitry Putunin

Intel Parallel Studio is now available.
Get free eval software:
intel.com/software/prodcuts/eval

www.intel.com/go/parallel

Software Products

Backup

Optimization Notice

Intel® compilers, associated libraries and associated development tools may include or utilize options that optimize for instruction sets that are available in both Intel® and non-Intel microprocessors (for example SIMD instruction sets), but do not optimize equally for non-Intel microprocessors. In addition, certain compiler options for Intel compilers, including some that are not specific to Intel micro-architecture, are reserved for Intel microprocessors. For a detailed description of Intel compiler options, including the instruction sets and specific microprocessors they implicate, please refer to the "Intel® Compiler User and Reference Guides" under "Compiler Options." Many library routines that are part of Intel® compiler products are more highly optimized for Intel microprocessors than for other microprocessors. While the compilers and libraries in Intel® compiler products offer optimizations for both Intel and Intel-compatible microprocessors, depending on the options you select, your code and other factors, you likely will get extra performance on Intel microprocessors.

Intel® compilers, associated libraries and associated development tools 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 Intel® Streaming SIMD Extensions 2 (Intel® SSE2), Intel® Streaming SIMD Extensions 3 (Intel® SSE3), and Supplemental Streaming SIMD Extensions 3 (Intel® 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.

While Intel believes our compilers and libraries are excellent choices to assist in obtaining the best performance on Intel® and non-Intel microprocessors, Intel recommends that you evaluate other compilers and libraries to determine which best meet your requirements. We hope to win your business by striving to offer the best performance of any compiler or library; please let us know if you find we do not.

Notice revision #20101101