OpenSWR: A Scalable High-Performance Software Rasterizer for SciVis

Intel HPC Developer Conference

November 14, 2015

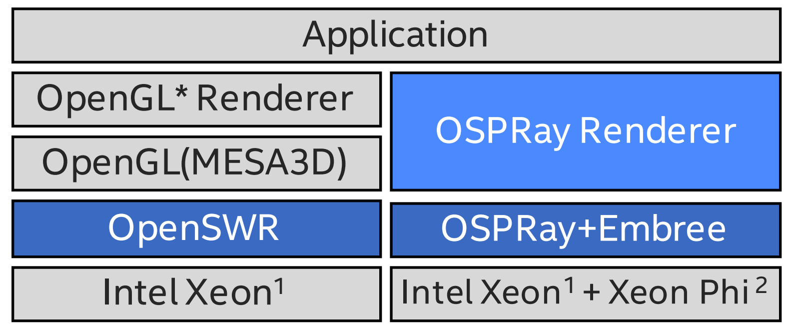
Software Defined Visualization: Our Approach

Option 1: Support existing APIs (OpenGL*)

- Works with existing applications
- No code changes or recompilation required
- ➔ **OpenSWR** software rasterizer

Option 2: Enable new functionality and improved performance through a new API

- Good option for new applications
- Integration underway for existing Key applications (ParaView*, VisIt, VMD, ...)
- ➔ **OSPRay** ray tracing based rendering engine [*built on Embree kernel library*]



[1] Intel® Xeon® processor,
[2] Intel® Xeon Phi™ coprocessor.

Our SDVis solutions support BOTH options!

Outline

- OpenSWR
  - Motivation and goals
  - Alpha Progress
  - SWR Core (a peek under the hood)
  - Early Results
  - Next steps

OpenSWR: Why Another Software Rasterizer?

	llvmpipe	SWR
Threading model	Single threaded vertex processing, up to 16 threads for rasterization	Common thread pool (1 to N threads) that pick up frontend (vertex) or backend (fragment) work as available
Vertex Processing	Entire draw call processed in single pass	Large draws chopped into segments that can be operated on in parallel
Frontend/Backend coupling	Separate binning pass in the single threaded frontend	Frontend vertex processing and binning combined in a single pass
Primitive Assembly and Binning	JIT, hand-coded SSE, and Scalar C One primitive at a time	x86-intrinsics (Intel® Streaming SIMD Extensions, Intel® Advanced Vector Extensions) working on vectors of primitives

OpenSWR: Alpha1 Release

- December 2014
- SWR core rasterizer with prototype OpenGL* driver
- Roughly OpenGL* 1.4 equivalent
- Fixed function pipeline, no exposed shaders
- Just enough functionality to run key applications (ParaView, VisIt)
- Source available on github (https://github.com/OpenSWR/openswr)

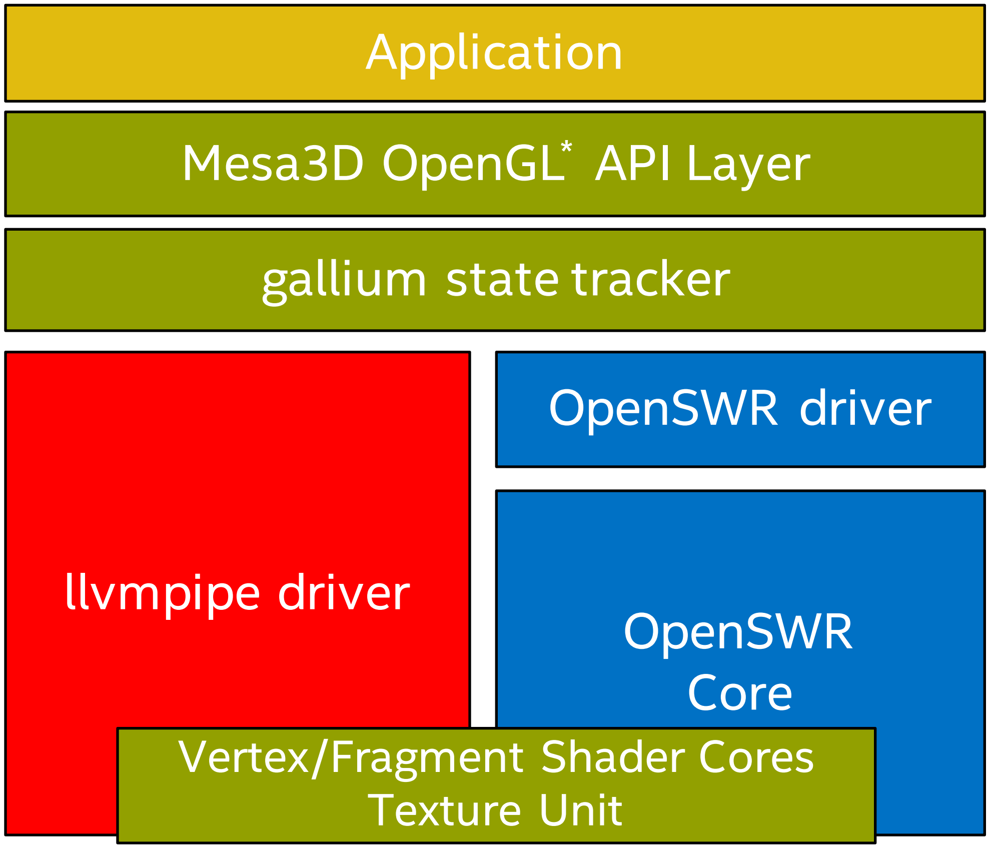
OpenSWR: Alpha2 Release

- September 2015
- SWR core rasterizer with Mesa3D 11.0 driver layer
- OpenGL* 3.x functionality with stream-out
- Vertex and Fragment shaders using GLSL version 130.
- Key applications supported: ParaView, VisIt, EnSight
- Began Mesa community review process by posting source to github (https://github.com/OpenSWR/openswr-mesa)

OpenSWR: Alpha1 vs Alpha2

- Current performance is still quite good, but lost a lot in switching from prototype OpenGL* driver to Mesa – 70% in some cases!
  - due to interface issues that need tuning
  - differences in shader code generation
  - due to conformance and feature additions to the SWR core
- Functionality improvements were well worth the switch
  - a modern OpenGL* driver from scratch is a large and complex task!
- Hopeful to recover most of this performance back

OpenSWR: Architecture Within Mesa3D



SWR Core

(a peek under the hood)

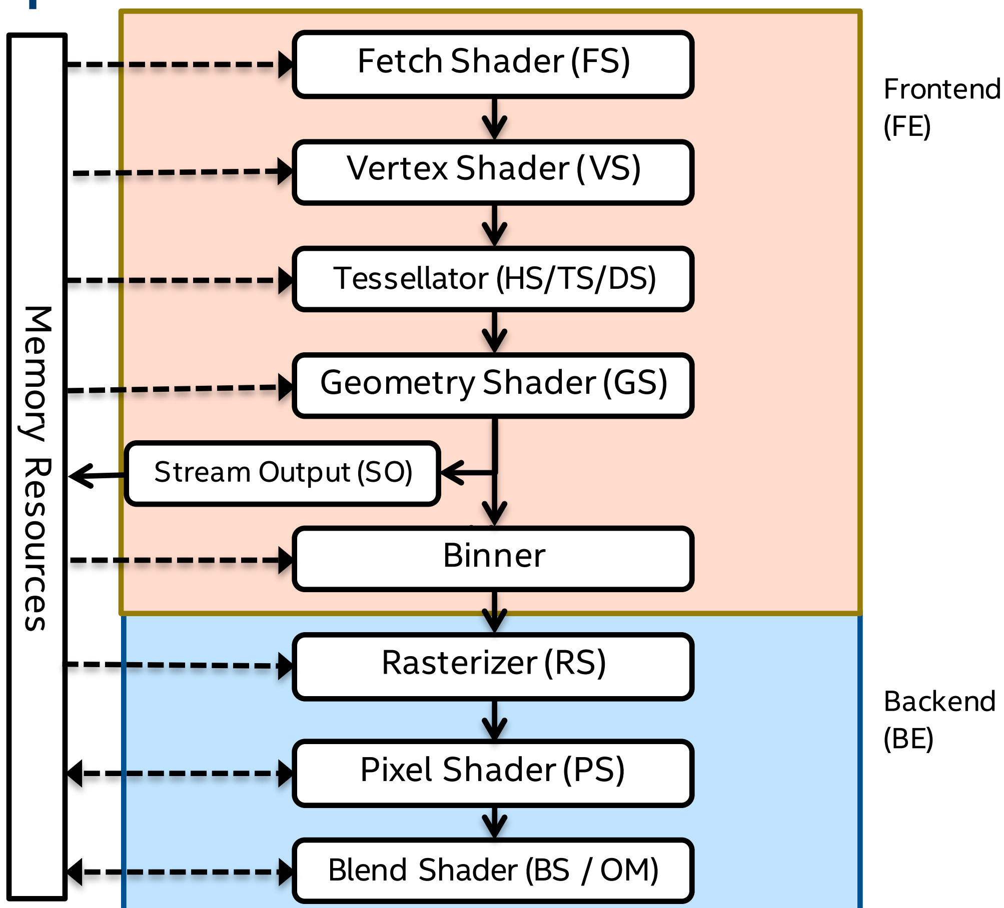
SWR Core: Features

- High performance, highly scalable software rasterizer
- Core provides threading model and fixed-function rasterization units
  - Features include tessellation, stream-output, 16x MSAA
- Driver layer provides shader callbacks
- Goal is to become a conformant, full-featured rasterizer
- Can support any API with appropriate driver

SWR Core: API

- SWR Core has a modern GPU-like API
  - Device/Context
  - State objects
  - Queries
  - Draw functions for execution
- Serves as a “hardware interface” on which to write a standard driver

SWR Core: Pipeline Overview



SWR Core: Frontend Features

- Fully multithreaded vertex processing
  - Common thread pool (1 to N threads)
  - All threads can work on frontend (vertex) or backend (fragment) tasks
  - All threads can participate in vertex processing – crucial for HPC
  - Large draws broken up into smaller draws to aid parallelism

- Vertex pipeline is fully vectorized up to AVX2
  - SIMD abstraction and LLVM use enables path to AVX512
  - Frontend optimized for each primitive type

SWR Core: Frontend Features (continued)

- Vertex memory management designed for low overhead
  - Only requires storage for vertices of 8 primitives
  - Custom arena allocator
  - Small binning memory for each triangle (40 bytes)

- Advanced vectorized culling techniques
  - Necessary for large triangle count workloads seen in HPC/SciVis
  - Zero area
  - Viewport/scissor cull
  - Cull between pixel centers (very small triangles)

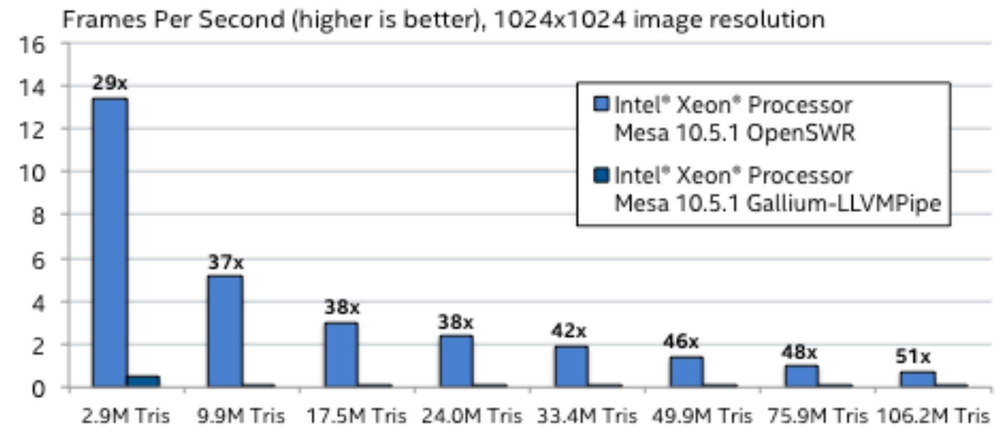
SWR Core: Frontend Features (continued)

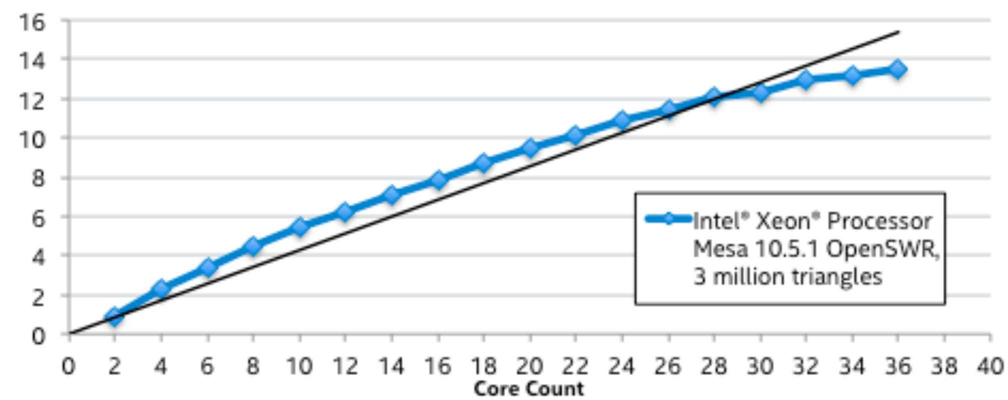
- Quick early-rasterization phase to further cull primitives around pixel centers
  - Triangles that get processed through this phase pass the resulting coverage mask to the backend to completely skip rasterization stage
  - Have seen > 90% triangles of large workloads culled, significantly reducing unnecessary binning/ rasterization

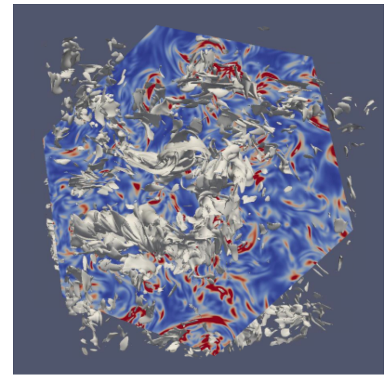
- Other ideas in development
  - Position-only shading
  - Deferred attribute shading
  - NUMA aware rasterization

OpenSWR Early Results

Performance: OpenSWR vs MESA* llvmpipe



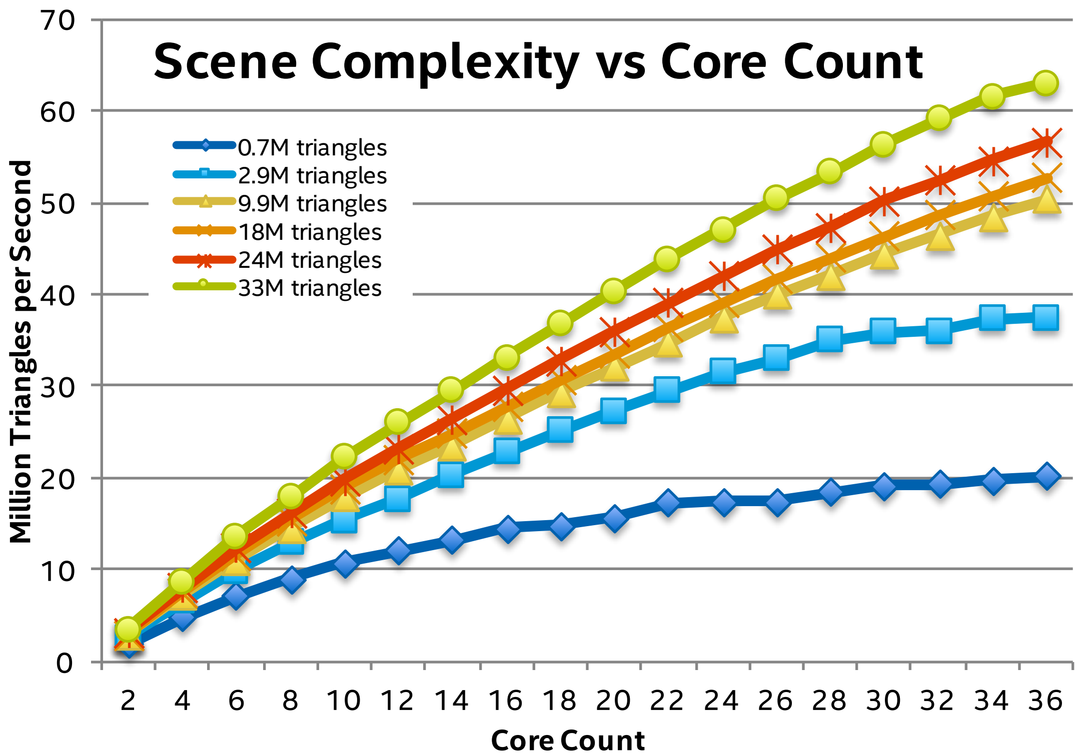


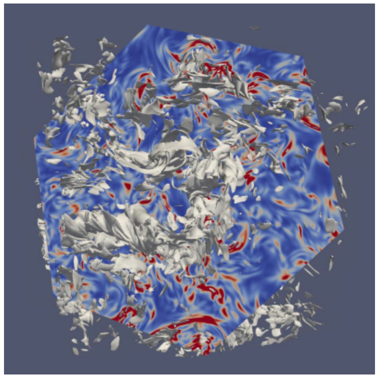


- Intel® Xeon® E5-2699 v3 Processor 2 x 18 cores, 2.3 GHz
- ParaView* 4.3.1
- OpenSWR "alpha 2"
- (full system configuration on slide 20)

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. For more information go to http://www.intel.com/performance.

Performance: OpenSWR





- Intel® Xeon® E5-2699 v3 Processor 2 x 18 cores, 2.3 GHz
- ParaView* 4.3.1
- OpenSWR "alpha 2"
- (full system configuration on slide 20)

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. For more information go to http://www.intel.com/performance.

Performance Test Configuration

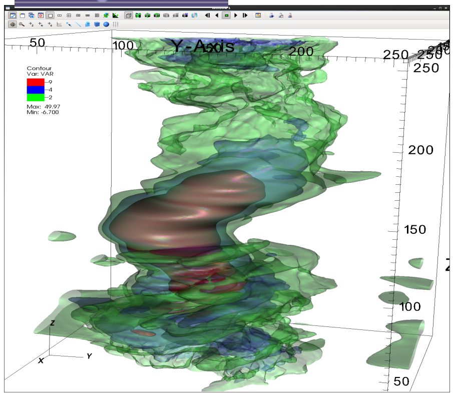
Node count	1
Platform	Cottonwood Pass Platform (Intel)
CPU	Intel® Xeon® processor E5-2699 v3 LGA2011 2.3GHz 45MB 145W (DP) Dual socket 18 core
RAM	128 GB total 8*16GB 2133MHz Reg ECC DDR4
BIOS	Vendor: Intel Corporation Version: SE5C610.86B.01.01.0005.101720141054 Release Date: 10/17/2014 BIOS Configuration: default
Hard drive	Intel® SSD SA2M160G2GC 1x160 GB SATA* SSD
NVIDIA Co-Processor	NVIDIA* GeForce* GTX* Titan X 3072 CUDA Cores 12GB memory Software Details: CUDA Version 7.0.28 OptiX Version 3.8.0 NVIDIA Driver Version 346.46
OS / Kernel	CentOS release 6.6 / 2.6.32-504.23.4.el6.x86_64

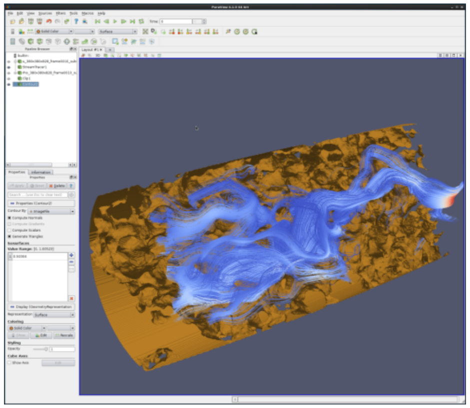
OpenSWR Testing Results

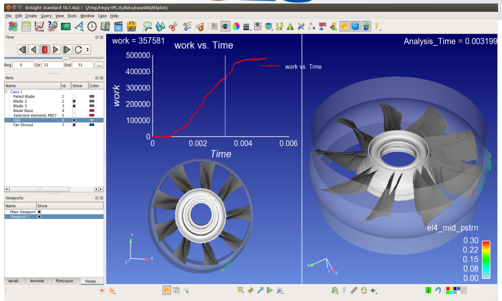
- VTK (OpenGL2* and OSMesa backends)
  - ctest suite passing 99% (~1600 tests)
- ParaView (OpenGL2* backend)
  - ctest suite passing 96% (~1150 tests)
- VisIt v2.8.1 conformance suite
  - Passing 85% automated testing
  - Some real issues that we're fixing.
  - But, many are minor pixel differences between OpenSWR and gold images.
- EnSight*
  - Autotests passing at 95% (failing 31/679 tests)
  - "Almost all failed tests are due to line rendering issues" versus gold images.
  - Usage testing has uncovered some issues that we're debugging.
- OpenGL* conformance and Mesa piglit testing will follow
  - Results are currently poor due to missing GS support.

OpenSWR: Apps Experience

- ParaView
  - IPCC Lightening Talk
  - Kitware booth demo
- VisIt
  - IPCC Lightening Talk
  - University of Tennessee booth demo
- EnSight*
  - Sean Ahern, CEI
    “Rendering in EnSight with OpenSWR”
  - SuperMicro* booth demo







Roadmap

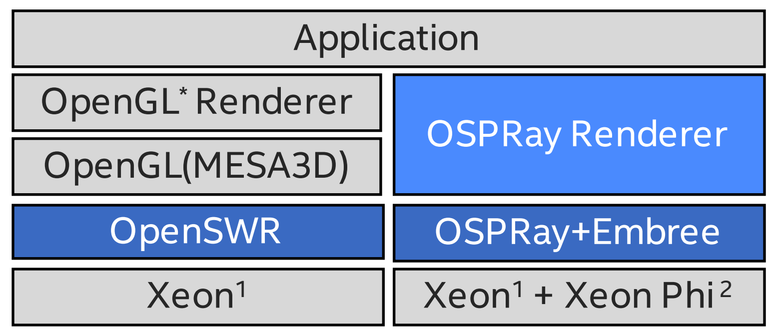
2015	Alpha2 – Source available Now • OpenGL* 3.x+, GLSL • Mesa3D Environment • VTK 6, 7 / ParaView* 4, 5 / VisIt 2.10
	Beta1 – Q4 • Initial Intel AVX-512 support • Planned source upstream to Mesa3D • CEI EnSight* support
2016	Beta2 – 1H 2016 • Optimized Intel AVX-512 • Additional app support in planning
	1.0 Release • Additional features / apps in planning

All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Intel® Advanced Vector Extensions (Intel® AVX);

Summary

- OpenSWR offers a compelling advantage over the standard Mesa software renderers for large-data visualization applications
- Both driver and core development are fully supported
- Planned open source upstream to Mesa master branch
- Works with existing OpenGL* applications
- No code changes or recompilation required



[1] Intel® Xeon® processor, [2] Intel® Xeon Phi™ coprocessor.

Q&A

Call to Action

http://www.openswr.org

Source available on github (https://github.com/OpenSWR/openswr-mesa)

Information about all of the Software Defined Visualization open source products can be found on http://sdvis.org/

Legal Notices and Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Statements in this document that refer to Intel's plans and expectations for the quarter, the year, and the future, are forward-looking statements that involve a number of risks and uncertainties. A detailed discussion of the factors that could affect Intel's results and plans is included in Intel's SEC filings, including the annual report on Form 10-K.

Cost reduction scenarios described are intended as examples of how a given Intel- based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Intel, the Intel logo, Xeon, Xeon Phi are trademarks of Intel Corporation in the U.S. and/or other countries.

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

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 SSE3 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

Risk Factors

The above statements and any others in this document that refer to plans and expectations for the second quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Words such as "anticipates," "expects," "intends," "plans," "believes," "seeks," "estimates," "may," "will," "should" and their variations identify forward-looking statements. Statements that refer to or are based on projections, uncertain events or assumptions also identify forward-looking statements. Many factors could affect Intel's actual results, and variances from Intel's current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be important factors that could cause actual results to differ materially from the company's expectations. Demand for Intel's products is highly variable and could differ from expectations due to factors including changes in business and economic conditions; consumer confidence or income levels; the introduction, availability and market acceptance of Intel's products, products used together with Intel products and competitors' products; competitive and pricing pressures, including actions taken by competitors; supply constraints and other disruptions affecting customers; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Intel's gross margin percentage could vary significantly from expectations based on capacity utilization; variations in inventory valuation, including variations related to the timing of qualifying products for sale; changes in revenue levels; segment product mix; the timing and execution of the manufacturing ramp and associated costs; excess or obsolete inventory; changes in unit costs; defects or disruptions in the supply of materials or resources; and product manufacturing quality/yields. Variations in gross margin may also be caused by the timing of Intel product introductions and related expenses, including marketing expenses, and Intel's ability to respond quickly to technological developments and to introduce new products or incorporate new features into existing products, which may result in restructuring and asset impairment charges. Intel's results could be affected by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Results may also be affected by the formal or informal imposition by countries of new or revised export and/or import and doing-business regulations, which could be changed without prior notice. Intel operates in highly competitive industries and its operations have high costs that are either fixed or difficult to reduce in the short term. The amount, timing and execution of Intel's stock repurchase program could be affected by changes in Intel's priorities for the use of cash, such as operational spending, capital spending, acquisitions, and as a result of changes to Intel's cash flows or changes in tax laws. Product defects or errata (deviations from published specifications) may adversely impact our expenses, revenues and reputation. Intel's results could be affected by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust, disclosure and other issues. An unfavorable ruling could include monetary damages or an injunction prohibiting Intel from manufacturing or selling one or more products, precluding particular business practices, impacting Intel's ability to design its products, or requiring other remedies such as compulsory licensing of intellectual property. Intel's results may be affected by the timing of closing of acquisitions, divestitures and other significant transactions. A detailed discussion of these and other factors that could affect Intel's results is included in Intel's SEC filings, including the company's most recent reports on Form 10-Q, Form 10-K and earnings release.

Rev. 4/14/15