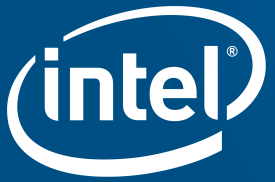

INTEL® HPC DEVELOPER CONFERENCE
FUEL YOUR INSIGHT
intel®



INTEL® HPC DEVELOPER CONFERENCE

FUEL YOUR INSIGHT

UPDATE ON OPENSWR: A SCALABLE HIGH-PERFORMANCE SOFTWARE RASTERIZER FOR SCIVIS

Jefferson Amstutz

Intel Corporation

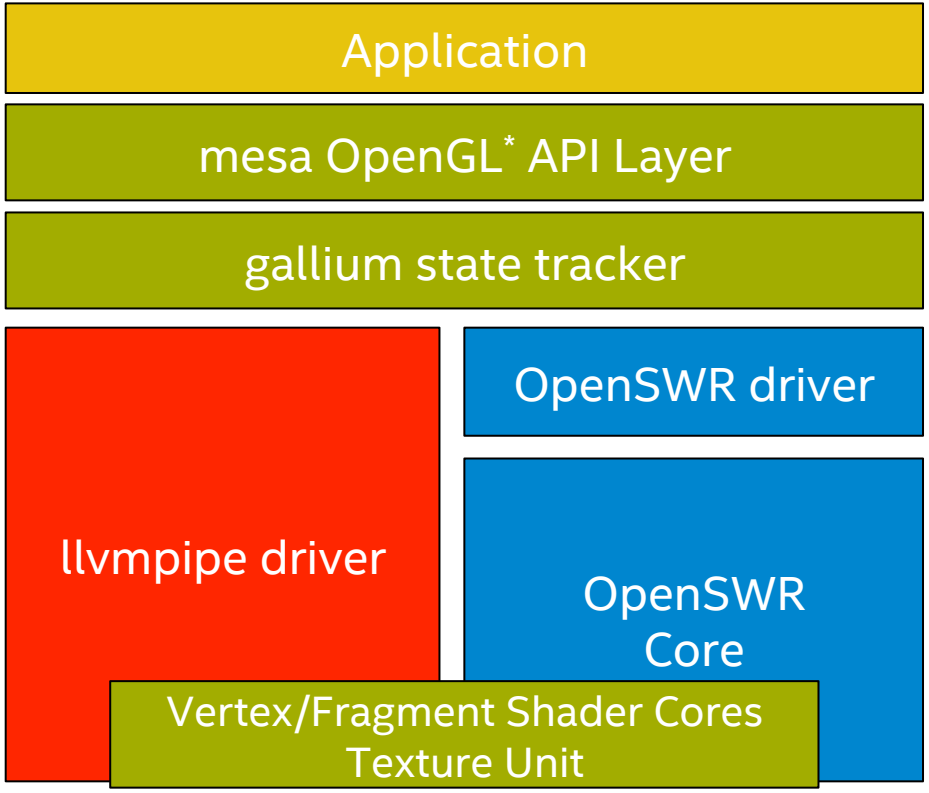
November 12, 2016

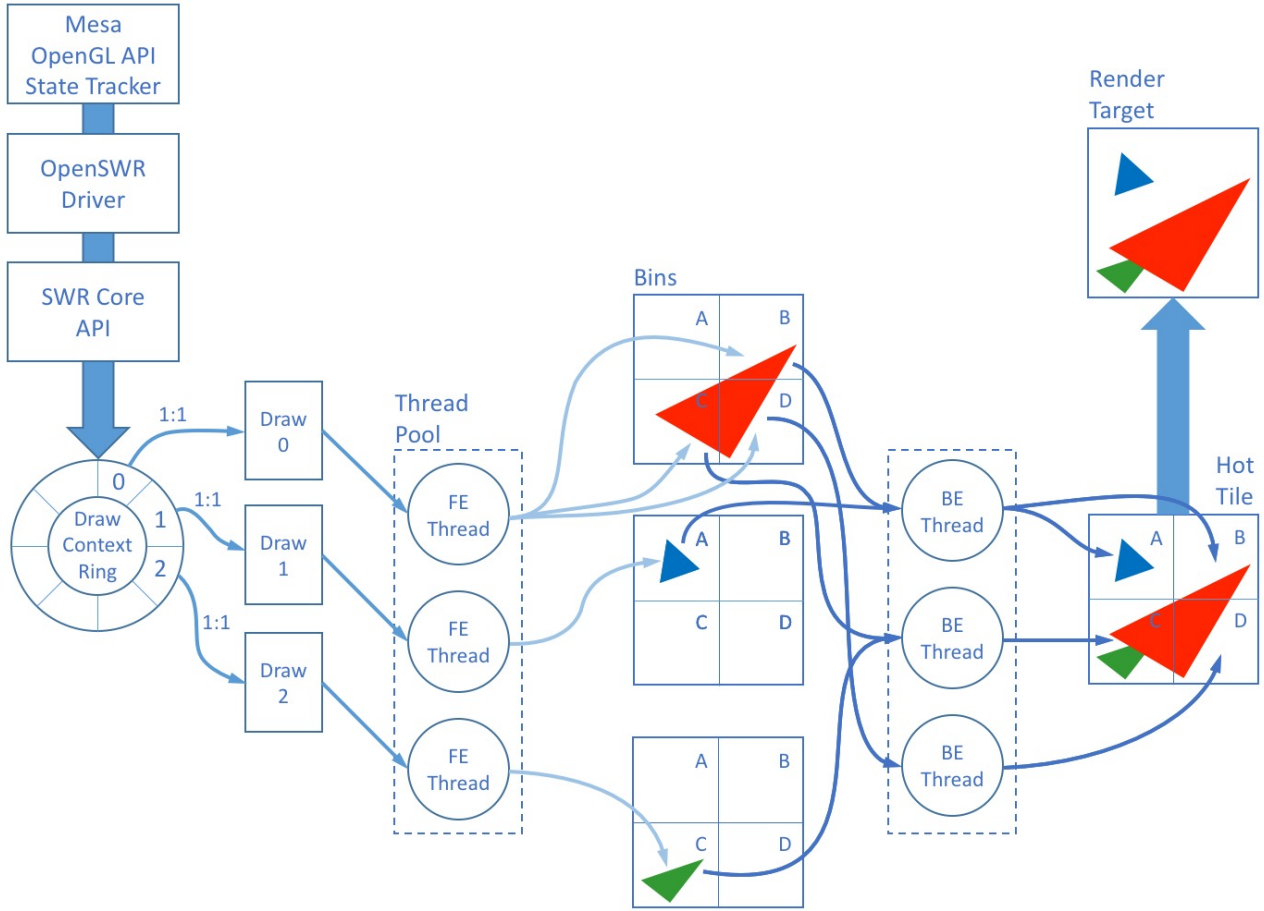
OpenSWR Software Rasterizer

OpenSWR is a highly scalable OpenGL software rasterizer that allows use of unmodified visualization software.

- OpenGL 3.2 functionality (supporting primary applications).
- Features full threading, LLVM jitted shaders, and Intel® Advanced Vector Extensions (AVX & AVX2) vectorization.
- Works with existing applications, no code changes or recompilation required.

Refresher: OpenSWR Architecture Within Mesa




Mesa
OpenGL API
State Tracker
OpenSWR
Driver
SWR Core
API
Draw
Context
Ring
0
1
2
1:1
1:1
1:1
Draw
0
Draw
1
Draw
2
Thread
Pool
FE
Thread
FE
Thread
FE
Thread
Bins
A
B
C
D
A
B
C
D
A
B
C
D
BE
Thread
BE
Thread
BE
Thread
Render
Target
Hot
Tile
A
B
C
D

Refresher: 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

HPC Optimizations

Specific optimizations necessary for high triangle count workloads:

- All threads can work on frontend (vertex) or backend (fragment) tasks
- Large draws broken up into smaller draws to aid parallelism
- Frontend is optimized for each primitive type
- Advanced vectorized culling techniques
  - Zero area
  - Viewport/scissor cull
  - Cull between pixel centers (very small triangles)

Progress since SC15

What's been keeping us busy:

- Upstreamed to Mesa
  - All development occurs in Mesa master branch
  - Mesa 12.0 was the first release to include OpenSWR
  - Additional features, performance & bug fixes for Mesa 13.0 release
- Optimization work in-progress for:
  - Intel® Xeon Phi™ Processor (previously code named Knights Landing)
  - Intel® Advanced Vector Extensions 512 (Intel® AVX-512)
- In production on TACC and LANL systems.

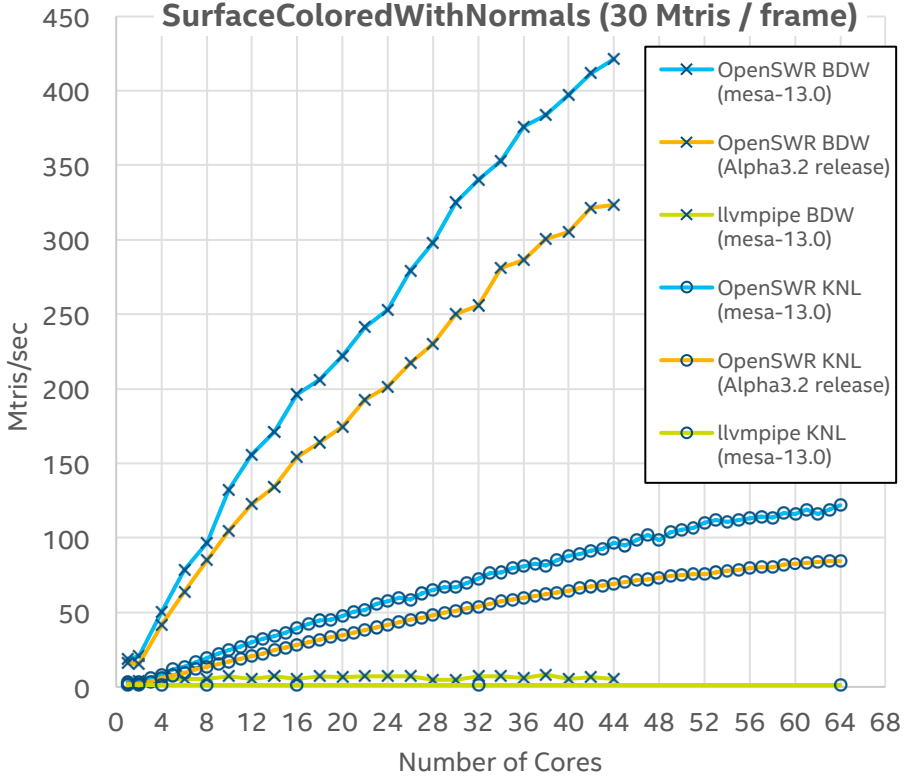
Broadwell workstation configuration – 2016-11-04

Node count	1
Platform	Intel Cottonwood Pass
CPU	Xeon DP Broadwell-EP E5-2699 v4 LGA2011 2.2GHz 55MB 145W Dual socket 22 core
RAM	128 GB total 8*16GB 2133MHz Reg ECC DDR4
BIOS	Vendor: Intel Corporation Version: SE5C610.86B.01.01.0016 Release Date: 03/31/2016 BIOS Configuration: default
Hard drive	INTEL SSDSC2BB480G601 1x160 GB SATA SSD
NVIDIA Co-Processor	eVGA GeForce GTX 970 NVIDIA Driver Version 361.42
OS / Kernel	Ubuntu 16.04 LTS / 4.4.0-45-generic

Xeon Phi™ workstation configuration – 2016-11-04

Node count	1
Platform	Ninja Developer Platform (DAP)
CPU	Xeon Phi 7210 1.30 GHz 215W 64 cores 16GB MCDRAM in quadrant cache mode
RAM	96 GB total 6*16GB 2133MHz Reg ECC DDR4
BIOS	Vendor: American Megatrends, Inc. Version: Supermicro K1SPE(-T) BIOS 1.0b CPLD 02.12.02 Release Date: 05/24/2016 BIOS Configuration: default
Hard drive	INTEL SSDPEDMX400G4 400 GB PCIe SSD
NVIDIA Co-Processor	NVIDIA* GeForce* GTX* 750 Ti
OS / Kernel	Ubuntu 16.10 / 4.8.0-26-generic

VTK Benchmark + OpenSWR



Application: VTK + OpenSWR

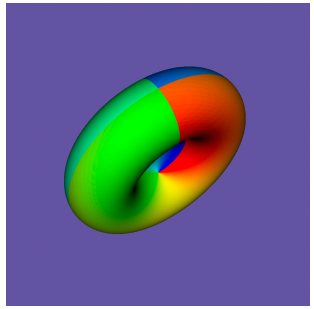
Description: Scientific visualization toolkit

Availability:

- **Code:** *http://openswr.org*
- **Recipe:** *http://openswr.org*

Usage Model:

- Native on Xeon, with full core utilization
- AVX/AVX2 Intrinsics (AVX-512 in-progress)



Highlights:

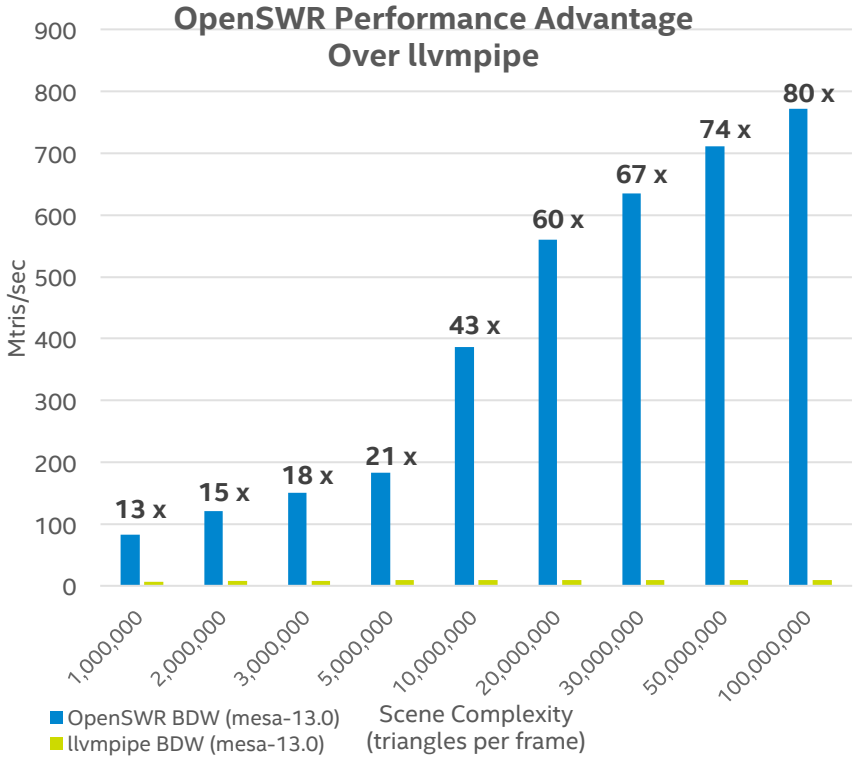
- OpenSWR enables simple OpenGL driver replacement, no app change
- VTK is the scientific visualization toolkit behind ParaView and VisIt, and a good platform for showing comparative performance across typical vis rendering workloads
- End-User benefits: Ability to achieve competitive performance and the flexibility of IA for rendering / visualization applications

Results:

- OpenSWR provides interactive rendering across tested workloads and dramatic performance improvements over mesa llvmpipe
- OpenSWR on Xeon exhibits strong core scaling to optimally utilize available resources. Further KNL optimizations are in progress.
- Significant performance improvements over Alpha 3.2 release a year ago. Many additional performance areas to explore.

KNL = The Intel® Xeon Phi™ Processor (previously code named Knights Landing)

VTK Benchmark + OpenSWR



Application: VTK GLBenchmarking

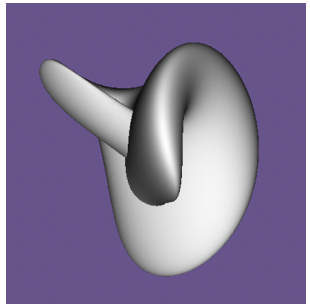
Description: Scientific visualization toolkit

Availability:

- **Code:** *http://openswr.org*
- **Recipe:** *http://openswr.org*

Usage Model:

- Native on Xeon, with full core utilization
- AVX/AVX2 Intrinsics



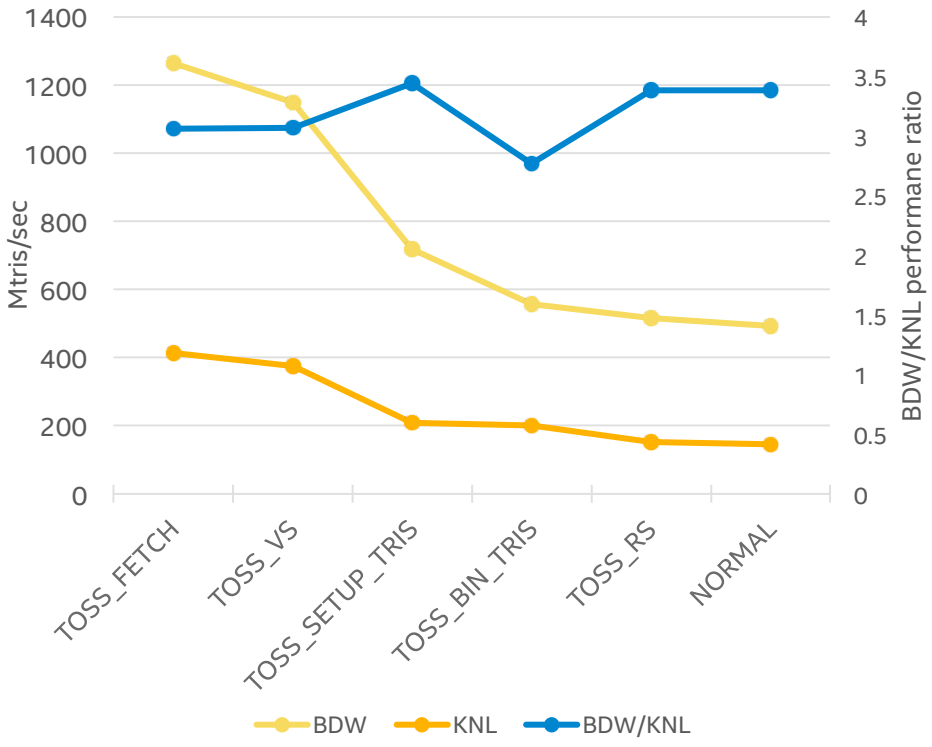
Highlights:

- OpenSWR enables simple OpenGL driver replacement, no app change
- VTK is the scientific visualization toolkit behind ParaView and VisIt, and a good platform for showing comparative performance across typical vis rendering workloads
- End-User benefits: Ability to achieve competitive performance and the flexibility of IA for rendering / visualization applications

Results:

- OpenSWR provides interactive rendering across tested workloads and dramatic performance improvements over mesa llvmpipe
- OpenSWR on Xeon exhibits strong core scaling to optimally utilize available resources. Further KNL optimizations are in progress.
- Significant performance improvements over Alpha 3.2 release a year ago. Many additional performance areas to explore.

KNL / BDW pipeline breakdown



- KNL is currently running AVX2 pipeline
- Some universal optimizations done based on KNL profiling
- Graph shows performance as measured at different points in the pipeline (discarding work after the indicated stage)
- Straightforward to expand fetch and vertex shading to AVX-512
- Triangle setup/binning could be vectorized
- Fragment shading parallelization needs rearchitecture to pack multiple triangles per vector

KNL / Intel® AVX-512

The Intel® Xeon Phi™ Processor (previously code named Knights Landing) is a many core array of smaller atom cores, with 512 bit wide vector units attached.

Challenges to performance tuning for this processor:

Instruction ordering and full vector utilization.

What needs to be done:

- Shader widening (fetch, VS, PS) and widen texture sampler to better utilize vector units
- Refactor pipeline stages to take advantage of Intel® Advanced Vector Extensions 512 (Intel® AVX-512) intrinsic instructions

Continued Work / Roadmap

- Enable MSAA
  - OpenSWR will be the first mesa software renderer with MSAA support
- ISA specific optimizations: KNL, AVX-512, and SKX
- Algorithmic performance improvements
  - Quick early-rasterization phase to further cull primitives around pixel centers
  - Position-only / deferred attribute shading
  - NUMA aware rasterization
- Additional Features as requested by major applications

Resources

- www.openswr.org
- SC15 presentation (openswr.org/talks/SC15DevConf-OpenSWR.pdf)
- mesa.freedesktop.org
- How to file bugs? Bugzilla on freedesktop.org

Q&A

Legal Notices and Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure.

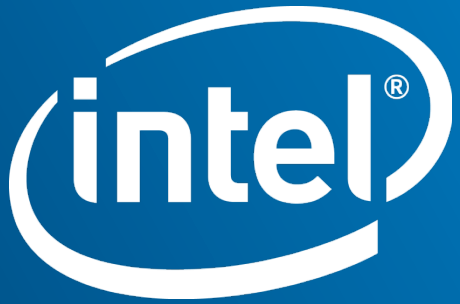
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may 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 complete information visit **http://www.intel.com/performance**.

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.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel, the Intel logo and others 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®