AMD

Fusion11

DEVELOPER SUMMIT

PROFILING AND PERFORMANCE ANALYSIS TOOLS FOR HETEROGENEOUS APPLICATIONS

Building tools and simulators for future devices

Perhaad Mistry, David Kaeli
Rafael Ubal, Dana Schaa,
Rodrigo Dominguez, Enqiang Sun

Northeastern University

Norman Rubin
AMD

Chris Gregg
University of Virginia

TOPICS

- **Part 1:** Building Performance Analysis Tools for Heterogeneous Applications ~ 20mins
- **Part 2:** Multi2Sim Simulation Framework - A CPU-GPU Model for Heterogeneous Computing ~ 10mins
- **Part 3:** Other interesting work at Northeastern ~ 5mins

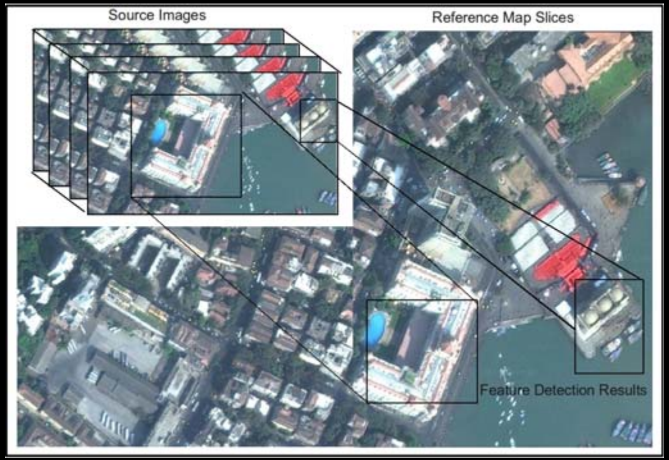
TOPICS

- ***Part 1: Performance Analysis Tools for Heterogeneous Applications***
  - Motivation for profiling tools, What does OpenCL provide?
  - OpenCL events and profiling usage
  - Speeded Up Robust Features (SURF)
  - Profiling SURF within the OpenCL interface
  - Profiling applications based on SURF
- **Part 2:** The Multi2Sim Simulation Framework - A CPU-GPU Model for Heterogeneous Computing
- **Part 3:** Other interesting work at Northeastern

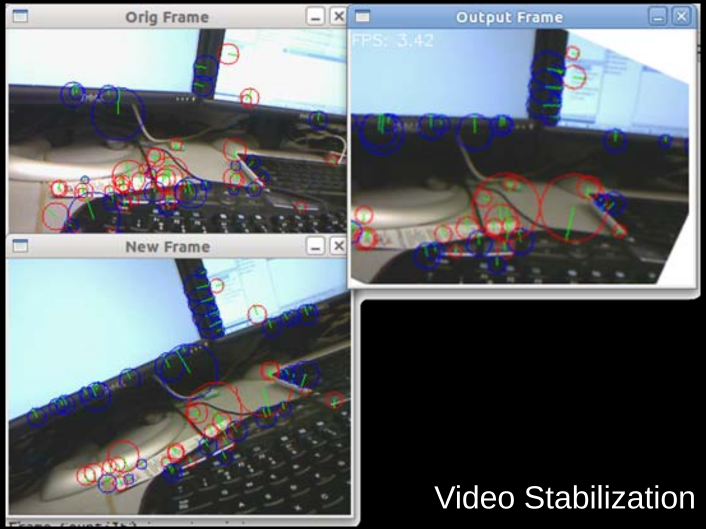
MOTIVATION FOR HETEROGENEOUS PROFILING TOOLS

Northeastern University

- Heterogeneous hardware running increasingly complex algorithms
  - Library developer cannot predict the application where his/her library will be used
- Algorithms whose performance is dependent on factors other than “data size”
  - Analysis is required at runtime by the library to learn about the application



Feature Based Image Search



Video Stabilization

AMD Fusion11 DEVELOPER SUMMIT

OPENCL EVENTS

- OpenCL provides not only cross platform applications, but also mechanisms to create tools for parallel computing
- ***Events*** are an interface to understanding OpenCL performance
  - Event objects (`cl_event`) used to determine command status
- OpenCL enqueue methods return event objects
  - Provides for command level control and synchronization

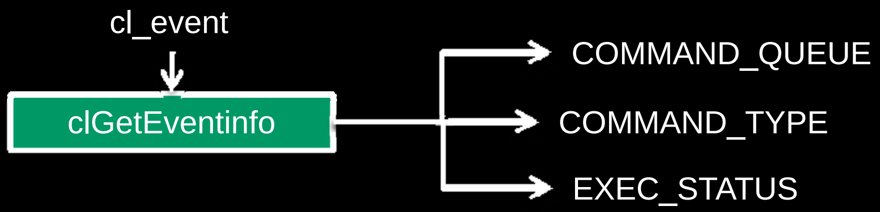
Command State	Description
CL_QUEUED	Command is in a queue
CL_SUBMITTED	Command has been submitted to device
CL_RUNNING	Command is currently executing on device
CL_COMPLETE	Command has finished execution

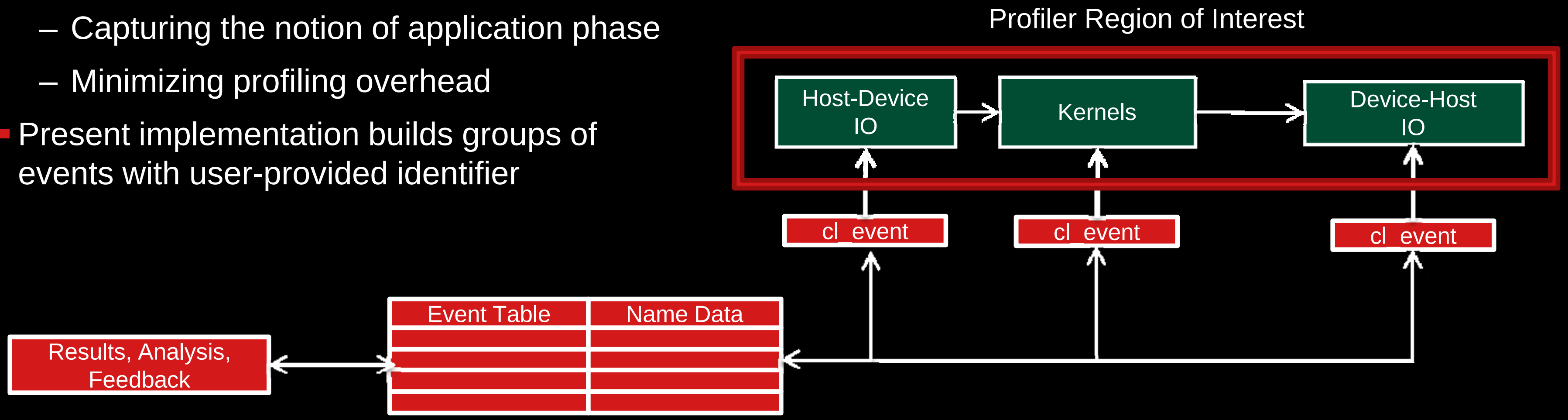
Command states as visible from OpenCL events

```
cl_int clEnqueueNDRangeKernel (
        cl_command_queue queue,
        cl_kernel kernel, cl_uint work_dim,
        const size_t *global_work_offset,
        const size_t *global_work_size,
        const size_t *local_work_size,
        cl_uint num_events_in_wait_list,
        const cl_event *event_wait_list,
        cl_event *event)
```

OPENCL PROFILING

- Events provide rich runtime information
  - Not just timestamps
- Supports schedulers across multiple families of different devices (CPUs, GPUs, APUs)
- Implementation challenges
  - Capturing the notion of application phase
  - Minimizing profiling overhead
- Present implementation builds groups of events with user-provided identifier





References to event objects

SPEEDED UP ROBUST FEATURES (SURF)

Northeastern University

- Motivating example to build a OpenCL-based profiler
- *Summarize* an image into a number of *interest points*
  - Robust features - Simple to compute, compact in size
  - Less sensitive to changes in image scale and rotation
- Common applications:
  - Object recognition – Face recognition
  - Tracking - Navigation
  - Image stitching - Building panoramas

SURF →

I-point

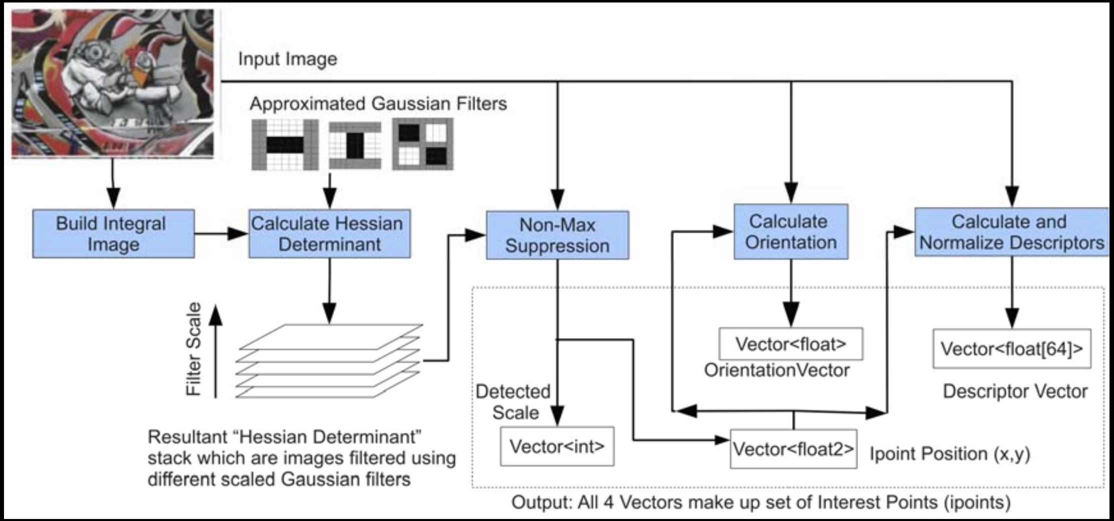
float2 Pixel Position
float Orientation
float Scale
float Descriptor[64]

Speeded-Up Robust Features (SURF), Herbert Bay et. al.

SPEEDED UP ROBUST FEATURES (SURF)

Northeastern University

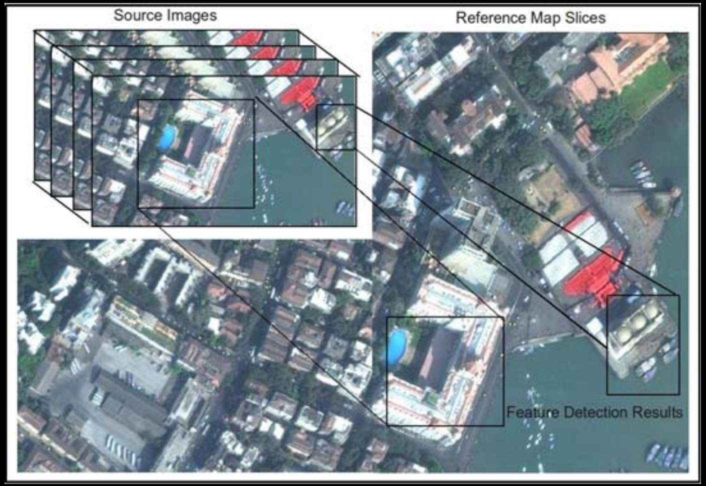
- Integral image: (2 kernels) 4 calls
  - Scan, transpose in 2 dimensions
- Hessian: (2 Kernels) 8 calls
  - Groups of convolutions
- Non max suppression: (1 kernel) 5 calls
  - Maxima and minima from convolution
- Orientation: (2 kernels) 2 calls
  - Local intensity gradients for rotation invariance
- Descriptors: (2 kernels) 2 calls
  - Haar descriptors around each i-point

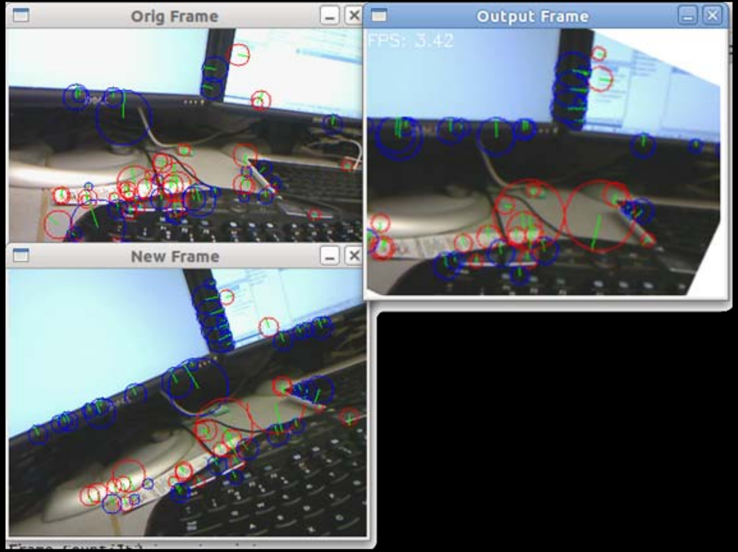


SURF is a multi-kernel pipeline where each stage contributes a part of each feature

SURF APPLICATIONS

- Simple applications using SURF's generated features
- Image Search - Compare descriptors of different features using simple Euclidean distance
- Video Stabilization - Compare orientation values of different features





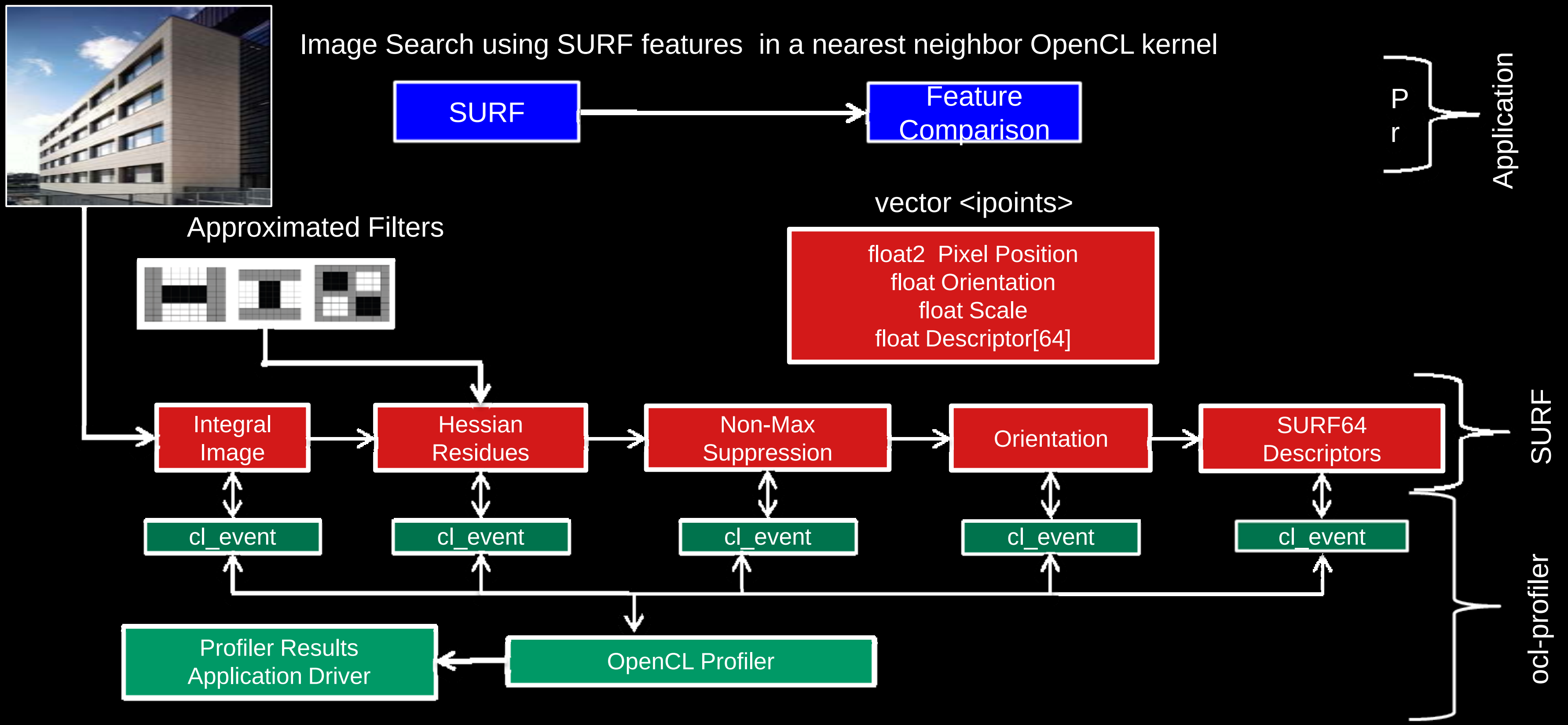
WHY ARE WE TALKING ABOUT SURF ?

Northeastern University

- Improve the state of the art in performance analysis tools for interesting workloads
  – We want to improve performance for complex and irregular applications and algorithms
- Performance Characteristics of SURF
  – Data driven performance necessitates profiling at runtime
  – Input arguments threshold determine performance
- Commonly used as a algorithm kernel within an application
  – Applications include stabilization of a video, image searching, motion tracking, etc.
  – The same algorithm is used for different applications with different input parameters
    - Number of convolutions
    - Thresholds

OPENCL PROFILER IN SURF APPLICATION

Northeastern University

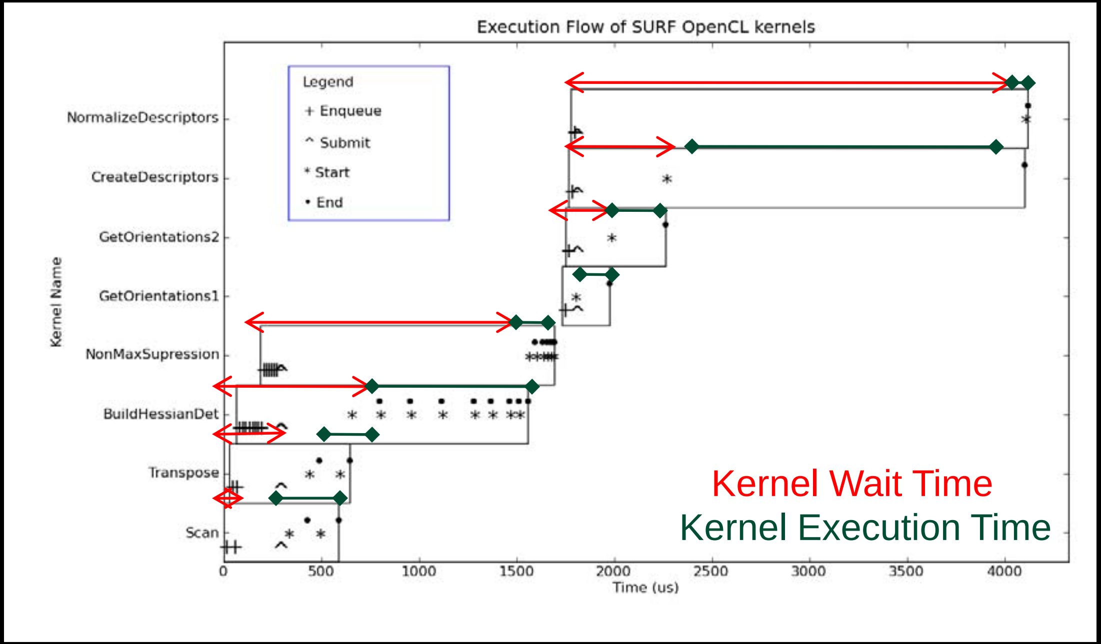


AMD Fusion11 DEVELOPER SUMMIT

KERNEL TIMELINE IN SURF

Northeastern University

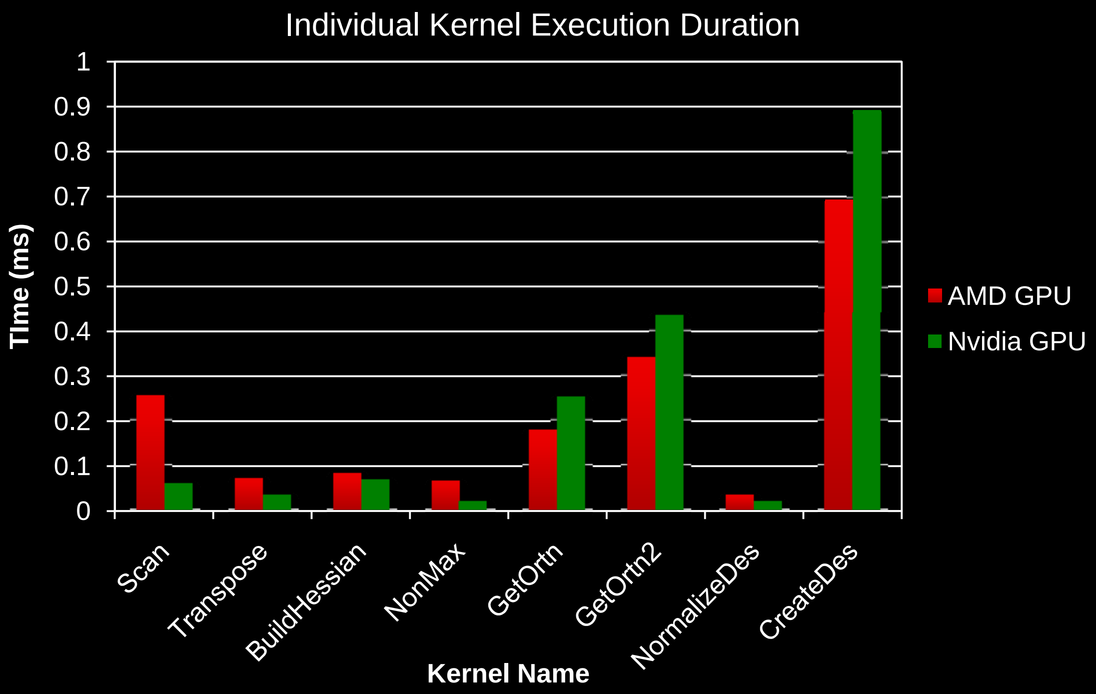
- Application view of SURF
  - Kernel pipelined over data set
  - Averaged event time stamps for a data set
- Exposes optimization opportunities
  - Cumulative time of small kernel
  - High kernel call count
  - Device – host IO duration is insignificant in pipeline
- Used to estimate host idle time once kernels are enqueued



Similar traces on any OpenCL compliant device ☺

INDIVIDUAL KERNEL PERFORMANCE

- Optimization steps for kernels
  - Timing of each kernel across frames
- Events show a consistent view across devices
- Individual timings are not representative
  - Createdescriptors is longest kernel
  - However BuildHessian is called more
  - Hard to find without profiling
- Reducing the number of kernel calls may be as beneficial as applying platform specific optimization
- Profiling allows us to pursue feedback-driven optimization

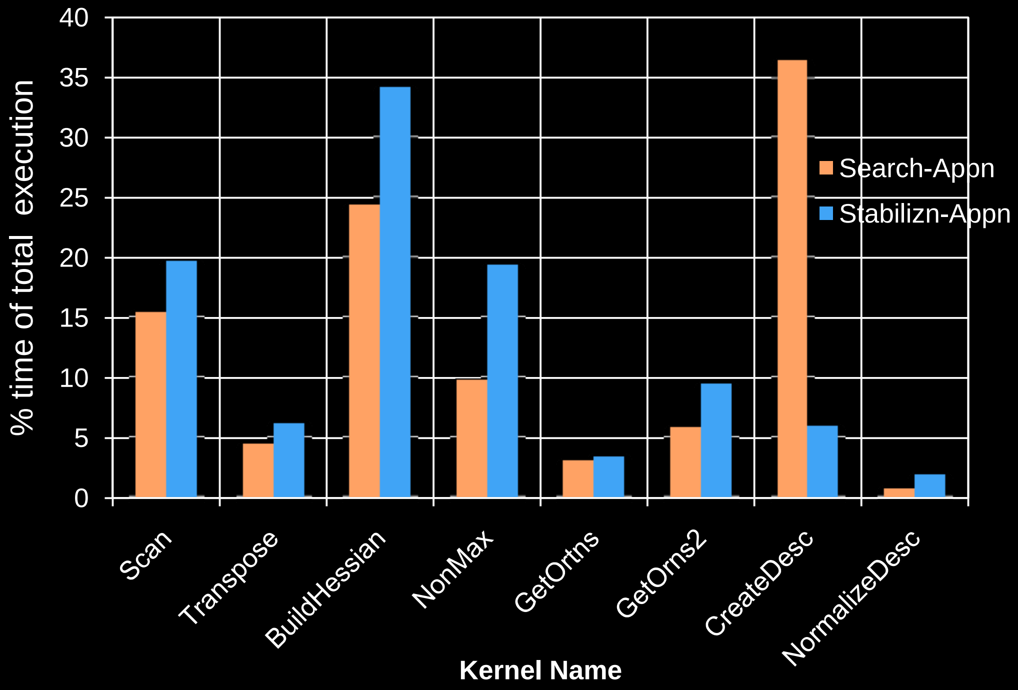


SURF PERFORMANCE FOR DIFFERENT APPLICATIONS

Northeastern University

- Different applications on top of SURF
  - Stabilization
  - Image Search
- Search Application:
  - Create-Descriptor is the bottleneck
  - Split kernel on multiple devices
- Stabilization Application:
  - Build-Hessian is the bottleneck
  - Reduce the number of kernel calls

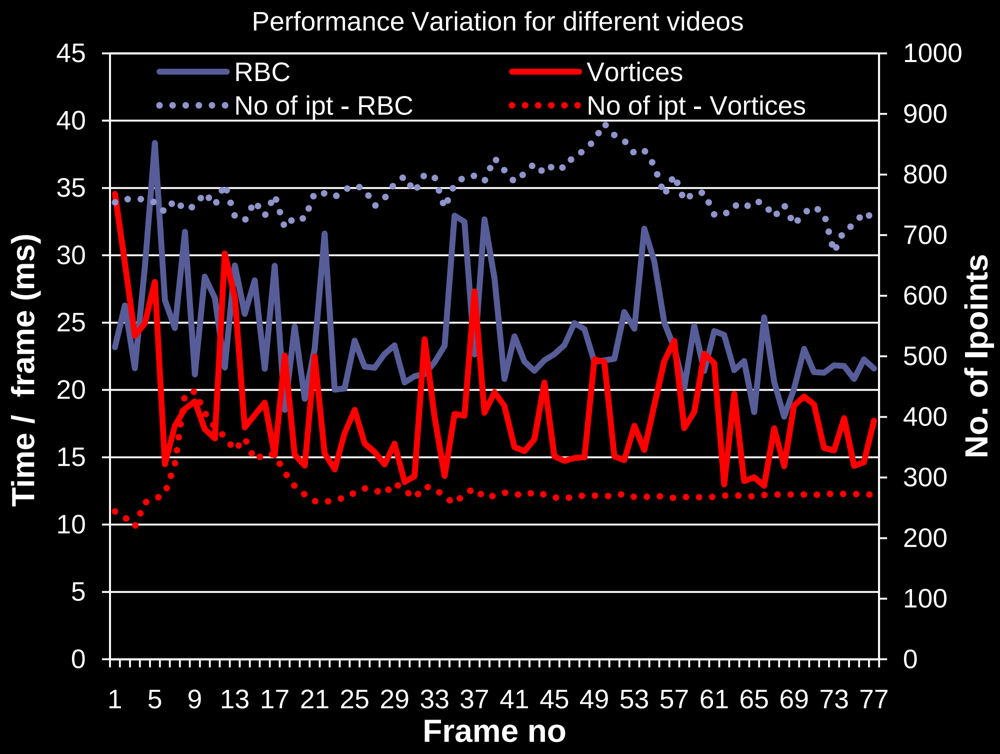
Percentage time of each kernel of SURF (AMD 5870)



SURF PERFORMANCE FOR DIFFERENT DATA SETS

Northeastern University

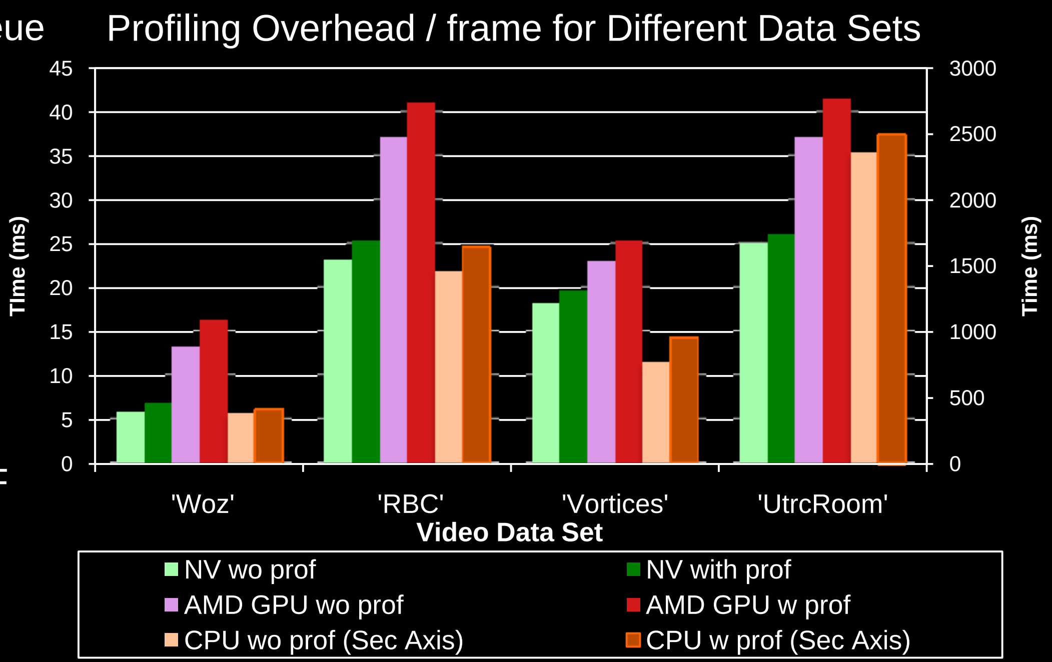
- Performance variation for videos of similar frame size
  – Use case for runtime performance analysis
- Same input parameters
  – Running a simple feature extraction
  – Variation due to differing feature count
  – Cannot predict the feature count
- Profiling enables performance analysis on a per data set basis
  – More than just “average time per frame”



AMD Fusion 11 DEVELOPER SUMMIT

PROFILER OVERHEAD

- Baseline: profiling disabled in command queue
  – Overhead for different videos
- Simple techniques to minimize overhead
  – Grow event list once and reuse data structures
- Query events after frame
  – Allows for variable granularity of performance measurement
- We show the worst case overhead for SURF
  – Profiling all kernels for every frame



Consistent overhead seen - per platform

SUMMARY

Northeastern University

- This work was motivated by an interesting case of data dependent parallelism performance
- SURF currently runs on CPUs, GPUs and APUs
  - Profiling plays an increasingly important role in heterogeneous environments
- The OpenCL specification provides a useful interface to understand application performance
  - Similar information provided for different devices
- Compliments existing such as the APP Profiler and Nvidia OpenCL Profiler
  - A common solution for multiple devices and vendors
  - Enables static and dynamic profiling and feedback optimization

AMD

Fusion11

DEVELOPER SUMMIT

THE MULTI2SIM SIMULATION FRAMEWORK

A CPU-GPU Model for Heterogeneous Computing

Rafael Ubal, Perhaad Mistry,
Dana Schaa, Rodrigo Dominguez
David Kaeli
Northeastern University

Norman Rubin
AMD

www.multi2sim.org

TOPICS

Northeastern University

- **Part 1:** Building Performance Analysis Tools for Heterogeneous Applications
- **Part 2:** The Multi2Sim Simulation Framework - A CPU-GPU Model for Heterogeneous Computing
  - Simulation needs for heterogeneous architectures
  - Introduction to Multi2Sim
  - The OpenCL callstack
  - OpenCL functional simulation of the Evergreen ISA
  - Usage scenarios for functional simulation
    - Instruction Mix
    - VLIW Packing
  - Status and future work

Part 3: Other interesting work at Northeastern ~ 5mins

CURRENT ARCHITECTURAL SIMULATION METHODOLOGY

Northeastern University

- Current simulation needs for performance analysis
  - Heterogeneous environments with CPU-GPU based systems
  - Tool for evaluation of new architectural proposals
  - Ability to model unique memory subsystems
  - Simulation of a GPU ISA

- Existing GPU simulation approaches
  - Barra: NVIDIA Tesla ISA
  - GPGPU-Sim: PTX architectural simulator
  - Ocelot: PTX emulator and optimizations
  - No publicly available architectural simulation or emulation of AMD ISAs
  - None of the above presently support heterogeneous simulation

Northeastern University

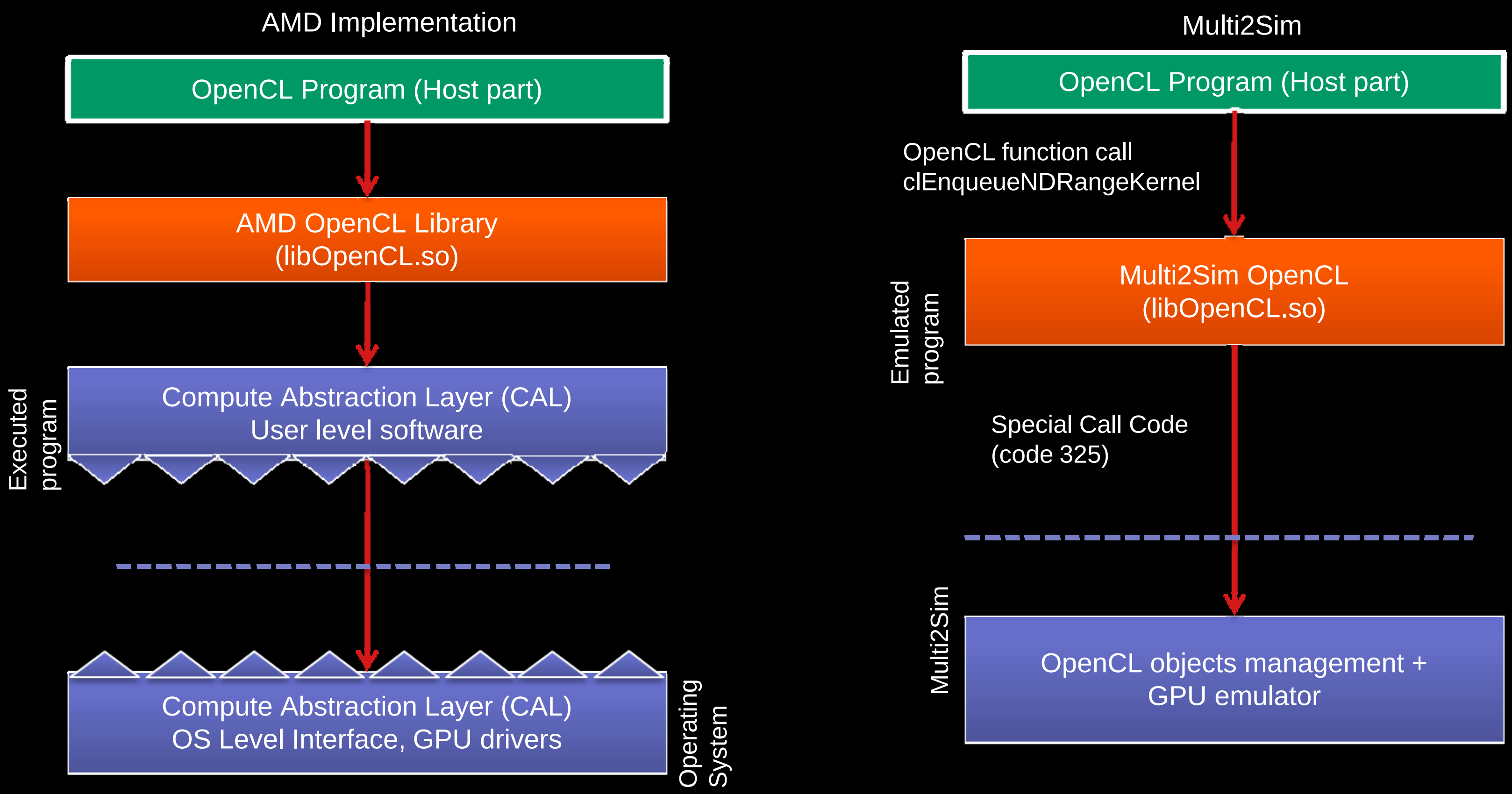
- History
  - Multi2Sim 1.x (MIPS) – Superscalar pipeline and multithreading
  - Multi2Sim 2.x (x86) – Multicore simulation with configurable memory hierarchy and interconnects
  - New Multi2Sim 3.x.x version series – Towards simulating heterogeneous computing
- Two different levels of accuracy
  - Functional simulation (or emulation): m2s-fast
  - Detailed (or timing) simulation: m2s

An Application Only Simulator

```
$ ./test-args hola que tal
    arg[0] = 'hola'
    arg[1] = 'que'
    arg[2] = 'tal'
```

```
$ ./m2s-fast test-args hola que tal
  <... Simulator output ...>
    arg[0] = 'hola'
    arg[1] = 'que'
    arg[2] = 'tal'
  <... Simulator statistics ...>
```

SIMULATING A GPU - THE OPENCL CALLSTACK



SIMULATING A GPU - KERNEL STATE

Northeastern University

1) **Global** Operations

a) OpenCL Binary Image Format (BIF)
b) Extract Evergreen machine code
c) Initialize device, constant memory
d) Set kernel arguments

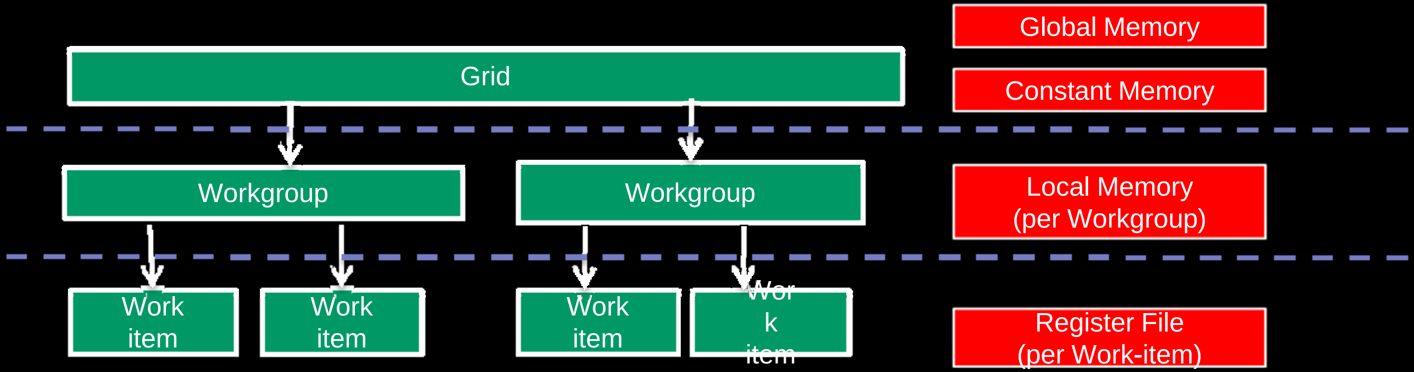
2) Per **work group** Operations

a) Initialize local memory
b) NDRange size.
c) Work-group size

3) Per **work item** Operations

a) Initialize registers
b) Work-item global coordinates
c) Work-item local coordinates

Functional View: Global and constant memories (per ND range). Local memory (per work-group). Registers (per work-item).



SIMULATING THE EVERGREEN VLIW-BASED ISA

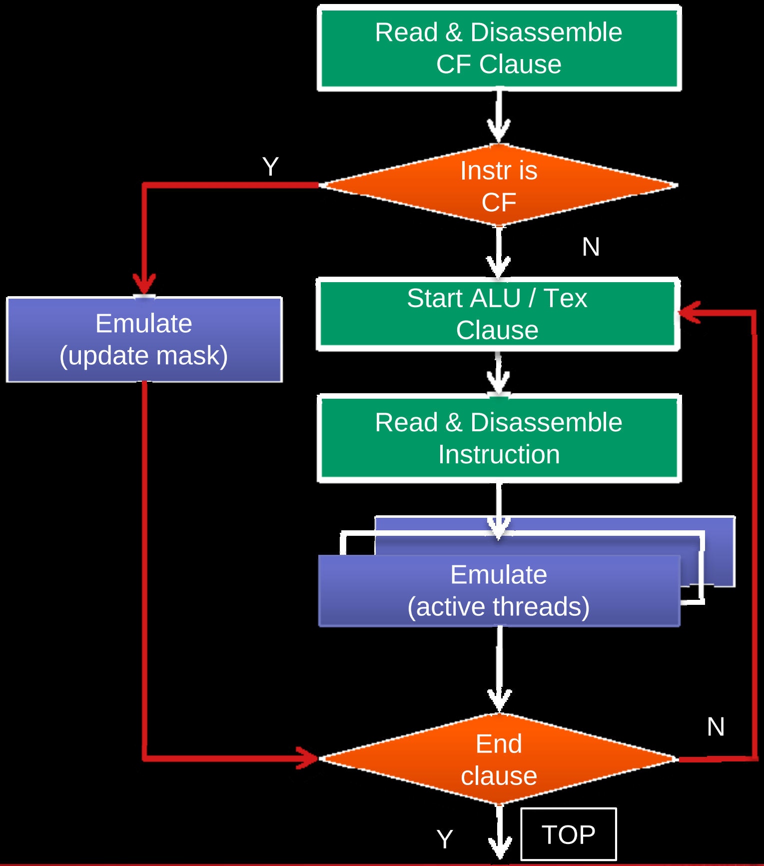
Northeastern University

- Evergreen program – Clause based format
  - Control flow (CF) clause
  - Arithmetic-logic (ALU) clause
  - Fetch-through-texture-Cache (TEX) clause
- Kernels handled as precompiled binaries
  - Precompiled kernel required
  - M2S cannot compile from source since simulator would be need to implement OpenCL compiler
- Compiler driver utility written as part of Multi2Sim tool chain to generate ISA trace

```
00 ALU_PUSH_BEFORE: ADDR(32) CNT(47) KCACHE0(CB0:0-15)

 0  x: MOV          R8.x,  0.0f
    y: MOV          R8.y,  0.0f
    z: ASHR         ____,  KC1[3].x,  (0x0000001F).x
    t: RCP_UINT__EG  T0.w,  KC0[1].x
 1  x: LSHL         R2.x,  KC0[1].x,  (0x00000005).x
    y: LSHR         T0.y,  PV0.z,  (0x0000001E).y
    z: MOV          R8.z,  0.0f
    w: MOV          R8.w,  0.0f
    t: MULLO_UINT  T0.z,  KC0[1].x,  PS0
 2  x: PREDNE_INT  ____,  R14.x,  0.0f

01 JUMP  ADDR(20)

02 ALU: ADDR(79) CNT(43) KCACHE0(CB0:0-15) KCACHE1(CB1:0-15)

 3  x: LSHL         R15.x,  R9.x,  (0x00000005).x
    y: LSHL         T0.y,  R0.y,  (0x00000002).y
    w: LSHL         T0.w,  KC0[1].x,  (0x00000004).z
    t: AND_INT      R16.x,  R1.x,  (0xFFFFFFFC).w
```

SIMULATING THE EVERGREEN ISA

Northeastern University

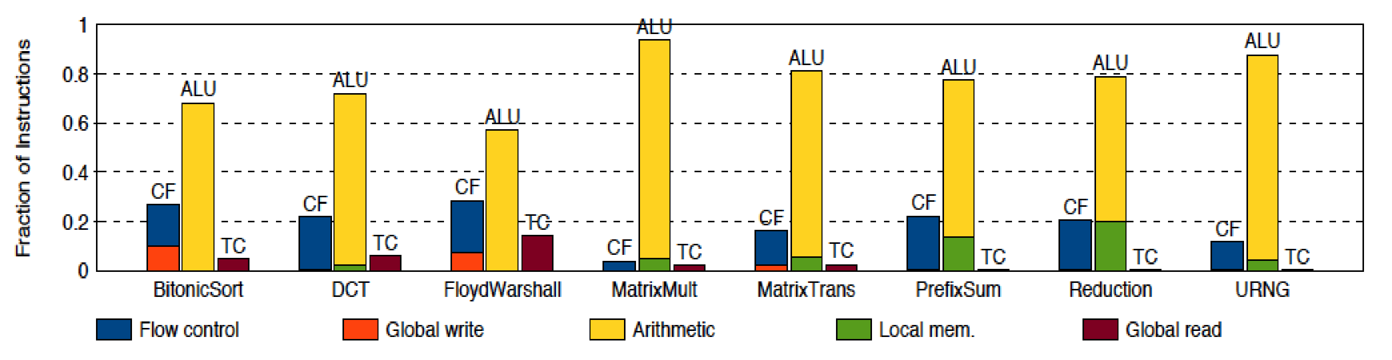



```
00 ALU_PUSH_BEFORE: ADDR(32) CNT(10) KCACHE0(CB0:0-15)

 0  x: MOV          R8.x,  0.0f
    y: MOV          R8.y,  0.0f
    z: ASHR         ____,  KC1[3].x,  (0x0000001F).x
    t: RCP_UINT__EG  T0.w,  KC0[1].x
 1  x: LSHL         R2.x,  KC0[1].x,  (0x00000005).x
    y: LSHR         T0.y,  PV0.z,  (0x0000001E).y
    z: MOV          R8.z,  0.0f
    w: MOV          R8.w,  0.0f
    t: MULLO_UINT  T0.z,  KC0[1].x,  PS0
2   x: PREDNE_INT  ____,  R14.x,  0.0f

01 JUMP  ADDR(20)

02 ALU: ADDR(79) CNT(43) KCACHE0(CB0:0-15) KCACHE1(CB1:0-15)

 3  x: LSHL         R15.x,  R9.x,  (0x00000005).x
    y: LSHL         T0.y,  R0.y,  (0x00000002).y
    w: LSHL         T0.w,  KC0[1].x,  (0x00000004).z
    t: AND_INT      R16.x,  R1.x,  (0xFFFFFFFC).w
```

GPU EMULATION USAGE SCENARIOS

Northeastern University

- Workload Characterization – Instruction Mix
- Instruction mix is not always a static analysis
  - A benchmark run with representative data
- Comparing the percentage of ALU, texture and control-flow clauses
  - Similar to Kernel Analyzer and APP profiler
- Statistics gathered by Multi2Sim and validated against AMD APP SDK Examples

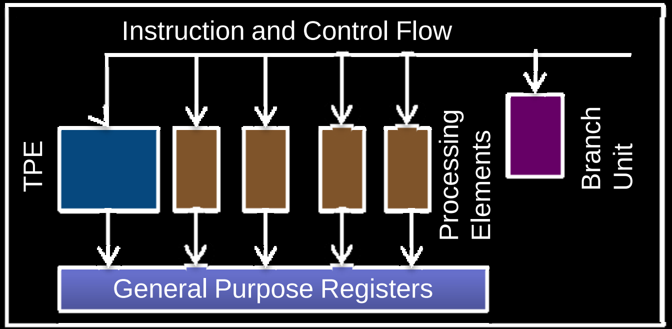


Percentage of ALU Clauses, CF Clauses and Texture clauses for examples in AMD APP SDK

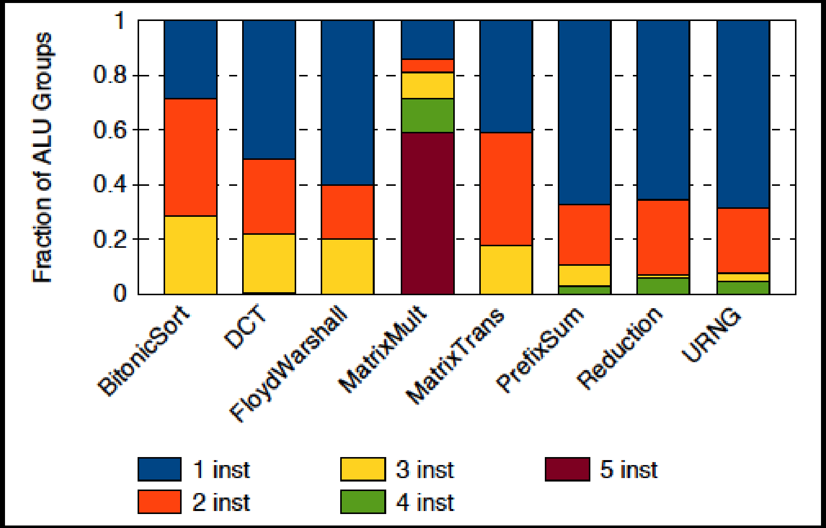
GPU EMULATION USAGE SCENARIOS

Northeastern University

- Workload Characterization – VLIW Packing
- Stream core of AMD GPU is a VLIW (Very Large Instruction Word) architecture
  - Upto 5 scalar instructions co-issued in a VLIW packet
- VLIW Packing handled by GPU shader-compiler
  - Improved by optimizations that increase arithmetic intensity (e.g. : loop unrolling, vectorization)



Stream core of AMD GPU



Breakdown of VLIW packing for AMD APP SDK examples

STATUS AND FUTURE WORK

Northeastern University

- Present Status
  - Validated against execution of the AMD APP SDK
- Ongoing Work
  - Architectural Simulation and exploration
  - Pipeline stages, Functional units and thread management
  - Full GPU memory subsystem
  - Pipeline visualization tool for heterogeneous architectures
- A complete heterogeneous simulation model by integration with the multicore model
  - First heterogeneous (x86 + Evergreen) architectural simulator for Fusion-like platforms

AMD Fusion 11 DEVELOPER SUMMIT

M2S MAILING LIST

Low traffic mailing list - Subscribe for updates

http://www.multi2sim.org/mailing

PACT tutorial (Friday October 14, 2011)

The Multi2Sim Simulation Framework.

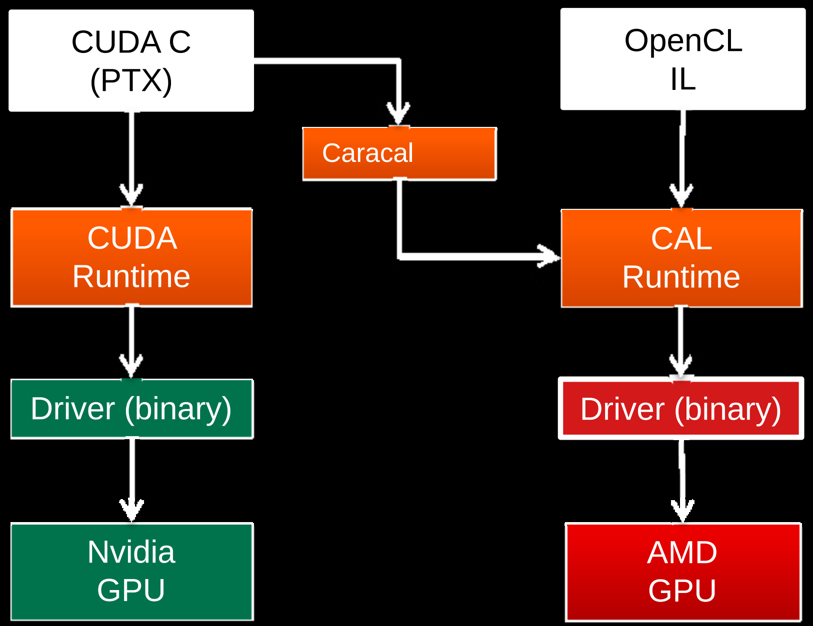
A CPU-GPU Model for Heterogeneous Computing

OTHER INTERESTING WORK IN NORTHEASTERN UNIVERSITY

Northeastern University

- Caracal
  - An open-source dynamic translator that can be used by compiler researchers
  - Allows CUDA C programs to run on AMD GPUs
- Motivation
  - Study intermediate representations
  - Study implications of translating architecture-dependent code
- Relevant URL

  http://code.google.com/p/gpuocelot



EXPERIENCES WITH MIGRATING APPLICATIONS TO A GPU

Northeastern University

- 3-D Cardiac CT Imaging
  - Iterative Least Squares Back Projection
- 3-D Breast Cancer Screening
  - Maximum Likelihood Estimation
- Intrusion Detection Systems
  - K-Nearest Neighbor Outlier Detection
- Physics-based Simulation for Surgical simulation
  - Data structures useful for physics simulation
- Ultrasound image processing pipeline

MASSACHUSETTS GENERAL HOSPITAL

MASSACHUSETTS GENERAL HOSPITAL

NUIC Technologies

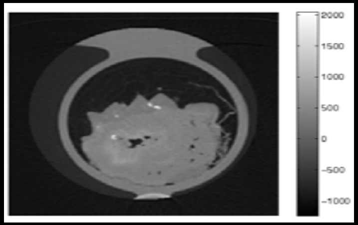
SimQuest

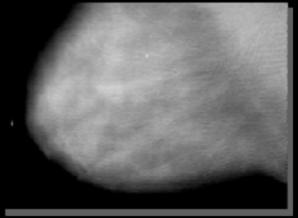
BK Medical

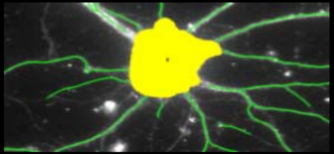
BIOMEDICAL IMAGE RECONSTRUCTION

Northeastern University

- Developing a suite of Biomedical Image Reconstruction Libraries
  – Implementations that can be tailored to different problems
- Target applications:
  – Deformable registration - radiation oncology
  – 3-D Iterative reconstruction – cardio-vascular imaging
  – Maximum likelihood estimation – Digital Breast Tomosynthesis
  – Motion compensation in PET/CT images - cardiovascular imaging
  – Hyperspectral imaging – skin cancer screening
  – Image segmentation – brain imaging
- $1.3M NSF Award EEC-0946463







AMD Fusion11 DEVELOPER SUMMIT

HOMEWORK FOR THE TRIP HOME ☺

Northeastern University

- SURF code download
  - http://code.google.com/p/clsurf
- Multi2Sim Download
  - www.multi2sim.org
- GPUOcelot
  - http://code.google.com/p/gpuocelot/
- Relevant Papers
  - **P. Mistry**, C. Gregg, N. Rubin, D. Kaeli, K. Hazelwood. **Analyzing program flow within a many-kernel OpenCL application**, *In Proceedings of the Fourth Workshop on General Purpose Processing on Graphics Processing Units, GPGPU-4*
  - R. Dominguez, D. Schaa, and D. Kaeli. Caracal: **Dynamic Translation of Runtime Environments for GPUs**. *In Proceedings of the Fourth Workshop on General Purpose Processing on Graphics Processing Units, GPGPU-4*
- For more information about GPU research in NUCAR
  - www.ece.neu.edu/groups/nucar/GPU/

AMD Fusion11 DEVELOPER SUMMIT

AMD Fusion11 DEVELOPER SUMMIT

THANK YOU !
QUESTIONS OR COMMENTS ?

Perhaad Mistry
pmistry@ece.neu.edu

Rafael Ubal (Author of M2S)
ubal@ece.neu.edu

Disclaimer & Attribution

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. There is no obligation to update or otherwise correct or revise this information. However, we reserve the right to revise this information and to make changes from time to time to the content hereof without obligation to notify any person of such revisions or changes.

NO REPRESENTATIONS OR WARRANTIES ARE MADE WITH RESPECT TO THE CONTENTS HEREOF AND NO RESPONSIBILITY IS ASSUMED FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

ALL IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED. IN NO EVENT WILL ANY LIABILITY TO ANY PERSON BE INCURRED FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AMD, the AMD arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. All other names used in this presentation are for informational purposes only and may be trademarks of their respective owners.

The contents of this presentation were provided by individual(s) and/or company listed on the title page. The information and opinions presented in this presentation may not represent AMD's positions, strategies or opinions. Unless explicitly stated, AMD is not responsible for the content herein and no endorsements are implied.