DEVELOPER TOOLS ARSENAL FOR TEGRA PLATFORMS

Sebastien Domine, 4/4/2016
TEGRA PLATFORMS

GAMING

DRONES

ROBOTICS

IVA

AUTOMOTIVE
SOFTWARE DEVELOPMENT WORKFLOW

Software Development

Toolchain Setup
Cross-compilation
Porting

Remote Debugging

Remote Profiling

Running

Debugging
CPU/GPU

Profiling
System/CPU/GPU/IO/…

Ship it!

Desktop Tools

CodeWorks

JetPack Install.

Nsight EE

Nsight Tegra VSE

Tegra System Profiler

Tegra Graphics Debugger

Tegra Graphics Debugger

CUDA Visual Profiler

Nsight EE

Cuda-gdb

Cuda-memcheck

PerfWorks

CUPTI

nvprof

Install.

Nsight EE

Nsight Tegra

Tegra Graphics Debugger

Tegra Graphics Debugger

CUDA Visual Profiler

Desktop Tools
COMMON CORE TOOLS OFFERING
Specialization Where Needed

CUDA Debugging and Profiling
Graphics Debugging and Profiling
GPU Performance Counters Libraries
CPU Profiling and System Trace
NVIDIA Tools Extension (NVTX)
GETTING STARTED...

CodeWorks 1r4

Jump starts developing for SHIELD/Android platform
Installs Android NDK & SDK tool chain
Installs Developer tools, GameWorks, Libraries,...
Reference documentation and samples
Cross-compiles code samples, pushes them to devkit
And Runs one sample...
GETTING STARTED...
JetPack Installer For DriveCX/PX and Jetson

Jump starts developing for Embedded platforms
Installs Linux ARM cross-compilation tool chain
Installs Developer tools, CUDA, Libraries,...
Flashes Drive PX/CX, Jetson TK1/TX1 OS Image
Reference documentation and samples
Compiles code samples, pushes them to devkit
And Runs one sample...
JETPACK AND CODEWORKS

New SDK Component Manager

- Parallel downloads
- Component update notification
- Component dependency resolution
- Integrated terminal window
CUDA STANDALONE TOOLS

Visual Profiler
Trace CUDA activities
Profile CUDA kernels
Correlate performance instrumentation with source code
Expert-guided performance analysis

NVPROF
Collect performance events and metrics

NVDISASM, CUOBJDUMP

CUDA-MEMCHECK
Detect out-of-bounds memory accesses
Detect race condition in memory accesses
Detect uninitialized variable accesses
Detect incorrect GPU thread synchronization

CUDA-GDB
Debug CUDA kernels with CLI
Debug CPU and GPU code
CPU and GPU core dump support
NVIDIA® NSIGHT™ ECLIPSE EDITION
Homogeneous application development for CPU+GPU compute platforms

CUDA-Aware Editor  
CUDA Debugger  
CUDA Profiler

CPU+GPU
CUDA 8.0
For Tegra Platforms

Tegra Parker Support
CUDA Debugging with Compute Preemption
CUDA Profiling with advanced PC Sampling metrics
CUDA Visual Profiler with Critical Path Analysis
DEPENDENCY ANALYSIS

Provide insight into application-level performance limiters
Expose dependencies between activities according to the programming model
Identify waiting time due to inter-stream dependencies
Highlight activities on the critical application runtime path
Supports CUDA (Linux/Mac/Windows) and POSIX threads (Linux/Mac)
DEPENDENCY ANALYSIS EXAMPLE

Dependencies between events derived from programming model constraints

Allows to compute wait states and the critical path

- `cudaLaunch`
- `cudaStreamSynchronize`
- `Kernel`

Stalls CPU (waiting time)
**DEPENDENCY ANALYSIS IN NVPROF**

New option to run post-mortem dependency analysis

New option to trace POSIX threads for multi-threaded applications

<table>
<thead>
<tr>
<th>Critical path(%)</th>
<th>Critical path</th>
<th>Waiting time</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>94.61%</td>
<td>3.942181s</td>
<td>0ns</td>
<td>clock_block(long*, long)</td>
</tr>
<tr>
<td>5.20%</td>
<td>216.857718ms</td>
<td>0ns</td>
<td>cudaMalloc</td>
</tr>
<tr>
<td>0.16%</td>
<td>6.617667ms</td>
<td>0ns</td>
<td>&lt;Other&gt;</td>
</tr>
<tr>
<td>0.01%</td>
<td>293.028000us</td>
<td>0ns</td>
<td>cuDeviceGetAttribute</td>
</tr>
<tr>
<td>0.01%</td>
<td>235.154000us</td>
<td>0ns</td>
<td>cudaGetDeviceProperties</td>
</tr>
<tr>
<td>0.01%</td>
<td>221.116000us</td>
<td>0ns</td>
<td>cudaFree</td>
</tr>
<tr>
<td>0.00%</td>
<td>158.703000us</td>
<td>0ns</td>
<td>cudaStreamCreate</td>
</tr>
<tr>
<td>0.00%</td>
<td>35.252000us</td>
<td>0ns</td>
<td>cudaConfigureCall</td>
</tr>
<tr>
<td>0.00%</td>
<td>35.248000us</td>
<td>0ns</td>
<td>cuDeviceGetName</td>
</tr>
<tr>
<td>0.00%</td>
<td>33.139000us</td>
<td>0ns</td>
<td>cuDeviceTotalMem_v2</td>
</tr>
<tr>
<td>0.00%</td>
<td>20.298000us</td>
<td>0ns</td>
<td>cudaSetupArgument</td>
</tr>
<tr>
<td>0.00%</td>
<td>19.433000us</td>
<td>0ns</td>
<td>cudaGetDevice</td>
</tr>
<tr>
<td>0.00%</td>
<td>0ns</td>
<td>3.942147s</td>
<td>pthread_join</td>
</tr>
<tr>
<td>0.00%</td>
<td>0ns</td>
<td>3.942136s</td>
<td>cudaStreamSynchronize</td>
</tr>
<tr>
<td>0.00%</td>
<td>0ns</td>
<td>1.001459s</td>
<td>pthread_mutex_lock</td>
</tr>
<tr>
<td>0.00%</td>
<td>0ns</td>
<td>980.464357ms</td>
<td>pthread_cond_wait</td>
</tr>
</tbody>
</table>
DEMO - DEPENDENCY ANALYSIS IN VISUAL PROFILER
TEGRA GRAPHICS DEBUGGER

Next-gen graphics development tools

Supports OpenGL ES 2.0/3.0/3.1 + Android Extension Pack
Monitor key software and hardware performance metrics
Debug draw calls, related states and resources
Live capture of a single rendering frame
Edit and recompile shaders live
Automatic GPU bottleneck analysis
Advanced timings for draw calls and kernel dispatches
NEW WITH TEGRA GRAPHICS DEBUGGER 2.X

Graphics Range Profiler
Advanced Interposer for non-rooted devices
Highlight redundant API state changes
NVTX support for perf markers and ranges
Highlight drawcalls based on shader selection
TEGRA SYSTEM PROFILER
Multi-core CPU profiler for all Tegra platforms

Easily prepare a device and deploy application for profiling

Maximize multi-core CPU utilization

Quickly identify CPU “hot spots”, “hot paths” and L1/L2 cache issues

Visualize multi-core CPU activities with a new timeline view

Time range filtering
NEW WITH Tegra SYSTEM PROFILER 2.5/2.6

Tegra Parker support and Expanded system trace

NVIDIA Tools eXtension Support (NVTX)

Visualize thread state: running/ready/blocked

Trace CUDA kernel workload execution (Jetson/DriveCX/PX)

Trace OpenGL-ES API calls

Visualize CPU and EMC frequencies
Real-time HD SLAM with GPU processing @ 25Hz

2560 x 720 @ 60Hz
PERFWORKS
Next-gen GPU Performance Counter Data Collection

C API for collecting GPU performance counters and data from NVIDIA GPUs.

- CUDA, OpenGL, OpenGL ES, D3D11, and D3D12
- Cross-Platform, with support for Kepler, Maxwell, Pascal GPUs
- Target Audience: tools developers, engine developers

Schedule: Beta in 3Q 2016
PERFWORKS SDK

Successor to the NVIDIA Perfkit SDK (NVPMAPI)
• Collect GPU metrics for Performance Monitoring (e.g. HUD).
• Automated serialized drawcall bottleneck analysis

Adds range-based profiling
• Collect metrics per user defined range, draw calls, or dispatches (Perf Markers, RTs, …)

Supports multi-threaded GPU work submission
• Collect data on modern multi-threaded APIs
• Collect consistent metrics across multiple generations of NVIDIA GPUs
Differentiate annotations from libraries and application

Middleware libraries have their own domain

User-defined and named synchronization primitives
GAME DEVELOPERS

Enable Game Developer to easily port to SHIELD/Android

• CodeWorks 1r4
• NDK profiling
• Same Graphics Debugging and Profiling Tools as PC

Support consumer devices

• No rooted OS requirement
• Basic feature set support on non-Tegra

Visual Studio support
NVIDIA® NSIGHT™ Tegra Visual Studio Edition

Android NDK/JDK application development

- Project Management
- Android Debugging
- Logcat Filtering
NEW WITH NSIGHT TEGRA 3.3
Android Marshmallow and ARMv8 AArch64

Support for non-Tegra devices
Tegra Graphics Debugger Attach
SIGSEV Signal handler
NDK r10e win64 (Link massive games + new options)
GDB 7.9 win64
CMake 3.1 support
Visual Studio 2015
Android GDB debugging in Visual Studio

Set breakpoints in both Java and Native (C/C++)

Use the familiar Visual Studio Locals, Watches, Memory and Breakpoints windows.

Build Native Android projects in Visual Studio using vs-android, ndk-build or makefiles.
AUTOMOTIVE
Nsight Eclipse Edition - NextGen

Build, Debug and Profile CUDA applications

Required Eclipse version 4.4 or later

Developed based on Eclipse CDT/DSF framework.

Using Eclipse remote system explorer plugins to connect to the remote devices.

Nsight plugins delivered as archive file(zip) and installed using standard Eclipse

Can co-exist with other Eclipse plugins in the user environment.
SCRENNSHOTS

Launch shortcuts for cuda-gdb

Debug session views
FUTURE

- Multi-process support
  - PerfWorks
  - Tegra System Profiler
- Multi-Node
  - DrivePX2 with 2 SoCs
- LTTNg GPU event provider
  - GPU process trace
Monday 4/4 - S6111, NVIDIA CUDA® Optimization with NVIDIA Nsight™ Eclipse Edition: A Case Study 9:00 - Room 211A

Tuesday 4/5 - S6659, Perfworks: A Library for GPU Performance Analysis 15:00 - Room 211B

Wednesday 4/6 - L6135A/B, Jetson Developer Tools Lab 13:30/15:30 - Room 210C

Thursday 4/7 - S6810, Optimizing App. Performance w/ CUDA Profiling Tools 10:00 - Room 211B

https://developer.nvidia.com