TEAPOT: A Toolset for Evaluating Performance, Power and Image Quality on Mobile Graphics Systems

Jose-Maria Arnau
Joan-Manuel Parcerisa

Computer Architecture Department
Universitat Politecnica de Catalunya

Polychronis Xekalakis

Intel Labs
Intel Corporation
Motivation

- Mobile GPU Simulator Requirements
Motivation

- Mobile GPU Simulator Requirements

1. Support for Mobile Applications
Motivation

- Mobile GPU Simulator Requirements

1. Support for Mobile Applications

![Angry Birds on Android](image)

2. Full-System GPU Simulation

![Pie chart showing % of GPU time](image)
Motivation

- Mobile GPU Simulator Requirements

1. Support for Mobile Applications

2. Full-System GPU Simulation

3. GPU & Screen Power Models

![Mobile GPU Simulator Requirements Diagram]

![Full-System GPU Simulation Diagram]

![GPU & Screen Power Models Diagram]
Motivation

- Mobile GPU Simulator Requirements

1. Support for Mobile Applications

2. Full-System GPU Simulation

3. GPU & Screen Power Models

4. Flexible GPU Timing Simulator

- System Memory
- GPU
- Screen

% of energy

- Tile-Based Deferred Rendering
- Immediate Mode Rendering

- On-Chip Memory
- External Memory

Jose-Maria Arnau
Motivation

- Mobile GPU Simulator Requirements

1. Support for Mobile Applications

2. Full-System GPU Simulation

3. GPU & Screen Power Models

4. Flexible GPU Timing Simulator

Not supported by any publicly available GPU simulator
Motivation

- Mobile GPU Simulator Requirements

1. Support for Mobile Applications

   ![Angry Birds on an Android device](image)

   % of GPU time

   - Android
   - Advertisement
   - Angry Birds

2. Full-System GPU Simulation

   ![Pie chart showing percentage of GPU time](image)

   Not supported by any publicly available GPU simulator

3. GPU & Screen Power Models

   ![Diagram of System Memory, GPU, Screen percentage of energy](image)

4. Flexible GPU Timing Simulator

   ![Diagram of Immediate Mode Rendering vs. Tile-Based Deferred Rendering](image)

   Tailored towards desktop-like power hungry GPUs

Jose-Maria Arnau
Outline

1. Motivation

2. Simulation Infrastructure
   2.1. OpenGL ES Trace Generation
   2.2. GPU Functional Simulation
   2.3. Cycle-Accurate Timing Simulation
   2.4. Power Model
   2.5. Image Quality Assessment

3. Conclusions
Simulation Infrastructure - Overview

- Tools unmodified
- Tools adapted
- Tools created from scratch
- Trace files
- Statistics

Mobile Applications

Android 4.2 Jelly Bean

Android Emulator

Virtual GPU

Desktop GPU Driver

OpenGL ES Trace Generator

OpenGL ES Trace
- Vertex/Fragment programs (GLSL)
- Textures
- Geometry

Tools unmodified
Tools adapted
Tools created from scratch
Trace files
Statistics
Mockup 1: Simulation Infrastructure - Overview

![Simulation Infrastructure Diagram]

**Mobile Applications**
- Android 4.2 Jelly Bean
- Android Emulator
- Virtual GPU

**Desktop GPU Driver**
- OpenGL ES Trace Generator
- OpenGL ES Trace
  - Vertex/Fragment programs (GLSL)
  - Textures
  - Geometry

**OpenGL Trace Generation**
- Tools unmodified
- Tools adapted
- Tools created from scratch
- Trace files
- Statistics

**GPU Functional Simulation**
- GPU Trace
  - GPU assembly instructions
  - Memory addresses
- Instrumented Gallium3D (softpipe driver)
- Frames
Simulation Infrastructure - Overview

**Mobile Applications**
- Android 4.2 Jelly Bean
- Android Emulator
- Virtual GPU

**Desktop GPU Driver**
- OpenGL ES Trace Generator
- OpenGL ES Trace
  - Vertex/Fragment programs (GLSL)
  - Textures
  - Geometry

**OpenGL Trace Generation**

**GPU Functional Simulation**
- GPU Trace
  - GPU assembly instructions
  - Memory addresses
- Instrumented Gallium3D (softpipe driver)
- Frames

**GPU Timing Simulation**
- Cycle-Accurate GPU Simulator
- McPAT
- Screen Power Model
- Image Quality Assessment

**Tools**
- Tools unmodified
- Tools adapted
- Tools created from scratch
- Trace files
- Statistics

Jose-Maria Arnau
Simulation Infrastructure - Overview

Mobile Applications
- Android 4.2 Jelly Bean
- Android Emulator
  - Virtual GPU

Desktop GPU Driver

OpenGL Trace Generation
- OpenGL ES Trace
  - Vertex/Fragment programs (GLSL)
  - Textures
  - Geometry

GPU Functional Simulation
- GPU Trace
  - GPU assembly instructions
  - Memory addresses
- Instrumented Gallium3D (softpipe driver)
- Frames

GPU Timing Simulation
- Cycle-Accurate GPU Simulator
- McPAT
- GPU Execution Time
- GPU Energy
- Screen Power Model
- Image Quality Assessment
- Screen Energy
- Image Quality Assessment

Tools unmodified
Tools adapted
Tools created from scratch
Trace files
Statistics
OpenGL ES Trace Generator

Mobile Game

Android 4.2 Jelly Bean
  - GPU Driver

Android Emulator
  - Virtual GPU

Desktop GPU
  - Desktop GPU Driver
OpenGL ES Trace Generator

Mobile Game

Android 4.2
Jelly Bean

GPU Driver

Android Emulator

Virtual GPU

Desktop GPU

Desktop GPU Driver

glDrawArrays(...)

Jose-Maria Arnau
OpenGL ES Trace Generator

Mobile Game

Android 4.2 Jelly Bean

GPU Driver

Android Emulator

Virtual GPU

glDrawArrays(…)

Desktop GPU

Desktop GPU Driver
OpenGL ES Trace Generator

Mobile Game

Android 4.2 Jelly Bean

GPU Driver

Android Emulator

Virtual GPU

Desktop GPU

Desktop GPU Driver

glDrawArrays(...)
OpenGL ES Trace Generator

- Mobile Game
  - `glDrawArrays(...)`
- Android 4.2
  - Jelly Bean
  - GPU Driver
- Android Emulator
  - Virtual GPU
- Desktop GPU
  - Desktop GPU Driver
OpenGL ES Trace Generator

Mobile Game

Android 4.2 Jelly Bean

GPU Driver

Android Emulator

Virtual GPU

glDrawArrays(…)

Desktop GPU

Desktop GPU Driver

Jose-Maria Arnau
OpenGL ES Trace Generator

void glDrawArrays(...) {
    saveCommandInfo(...);
    real_glDrawArrays(...);
}

Android 4.2
Jelly Bean

Android Emulator

Virtual GPU

GPU Driver

Mobile Game

glDrawArrays(...)

Desktop GPU

Desktop GPU Driver
OpenGL ES Trace Generator

Mobile Game

Android 4.2
Jelly Bean

GPU Driver

Virtual GPU

OpenGL ES Trace Generator

void glDrawArrays(...) {
    saveCommandInfo(...);
    real_glDrawArrays(...);
}

Desktop GPU

Thread: Mobile Game
OpenGL Context: A

- OpenGL ES Command List
- Shaders
- Geometry
- Textures
OpenGL ES Trace Generator

Virtual Buttons → Mobile Game → glDrawArrays(…)

Android 4.2 Jelly Bean → GPU Driver

Android Emulator → Virtual GPU

OpenGL ES Trace Generator

thread: Mobile Game
OpenGL Context: A

- OpenGL ES Command List
- Shaders
- Geometry
- Textures

Desktop GPU

Desktop GPU Driver

void glDrawArrays(...) {
  saveCommandInfo(...);
  real_glDrawArrays(...);
}
OpenGL ES Trace Generator

- Virtual GPU
- Mobile Game

Android 4.2
Jelly Bean

- GPU Driver

Android Emulator

Thread: Mobile Game
OpenGL Context: A
- OpenGL ES Command List
- Shaders
- Geometry
- Textures

Thread: Virtual Buttons
OpenGL Context: B
- OpenGL ES Command List
- Shaders
- Geometry
- Textures

OpenGL ES Trace Generator
void glDrawArrays(...) {
    saveCommandInfo(...);
    real_glDrawArrays(...);
}
OpenGL ES Trace Generator

Thread: **Mobile Game**
OpenGL Context: A
- OpenGL ES Command List
- Shaders
- Geometry
- Textures

Thread: **Virtual Buttons**
OpenGL Context: B
- OpenGL ES Command List
- Shaders
- Geometry
- Textures

void glDrawArrays(...)
{ saveCommandInfo(...);
real_glDrawArrays(...);
}
OpenGL ES Trace Generator

- **Virtual Buttons**
- **Mobile Game**
  - `glDrawArrays(...)`
  - OpenGL ES Trace

- **Android 4.2 Jelly Bean**
  - **Surface Flinger**
  - **GPU Driver**
  - **Virtual GPU**

- **Android Emulator**
  - **Virtual GPU**

- **Desktop GPU**
  - **Desktop GPU Driver**

**Thread: Mobile Game**
- OpenGL Context: A
  - OpenGL ES Command List
  - Shaders
  - Geometry
  - Textures

**Thread: Virtual Buttons**
- OpenGL Context: B
  - OpenGL ES Command List
  - Shaders
  - Geometry
  - Textures

**Thread: Surface Flinger**
- OpenGL Context: C
  - OpenGL ES Command List
  - Shaders
  - Geometry
  - Textures

**OpenGL ES Trace**

-Jose-Maria Arnau
OpenGL ES Trace Generator

OpenGL ES Trace Generator

Thread: **Mobile Game**
OpenGL Context: A
- OpenGL ES Command List
- Shaders
- Geometry
- Textures

Thread: **Virtual Buttons**
OpenGL Context: B
- OpenGL ES Command List
- Shaders
- Geometry
- Textures

Thread: **Surface Flinger**
OpenGL Context: C
- OpenGL ES Command List
- Shaders
- Geometry
- Textures

Support for multiple applications and OpenGL ES contexts

Virtual GPU

Android 4.2 Jelly Bean
Surface Flinger
Android Emulator

Desktop GPU

void glDrawArrays(...)
{  
  saveCommandInfo(...);
  real_glDrawArrays(...);
}

OpenGL ES Trace

Desktop GPU Driver

Jose-Maria Arnau
GPU Functional Simulation

- **OpenGL ES Trace**
  - Vertex/Fragment programs (GLSL)
  - Textures
  - Geometry

- **OpenGL ES front-end**

- **Intermediate Representation:** TGSi
  - Tungsten Graphics Shader Infrastructure

- **Instrumented Softpipe Driver**
  - Software Rasterizer
  - TGSI Emulator

- **Gallium3D**

- **GPU Trace**
  - Information stored per GPU command:
    - Thread ID
    - OpenGL ES Context ID
    - GPU Assembly Instructions (TGSI)
    - Memory addresses referenced for fetching vertices, texels and pixels
    ...
GPU Timing Simulator

- Immediate-Mode Rendering
• Immediate-Mode Rendering
GPU Timing Simulator

• Tile-Based Deferred Rendering
GPU Timing Simulator

- Tile-Based Deferred Rendering


**GPU Timing Simulator**

- **Tile-Based Deferred Rendering**

---

**Diagram Details**:
- **GPU Trace**: GPU Command 0, GPU Command 1, GPU Command 2
- **Command Processor**
- **Vertex Cache**
- **Vertex Fetcher**
- **Vertex Processor**
- **Primitive Assembly**
- **Tiling Engine**
- **Tile Cache**
- **Polygon List Builder**
- **Tile Scheduler**
- **Z-Buffer**
- **Early Depth Test**
- **Rasterizer**
- **Color Buffer**
- **Texture Cache**
- **Fragment Processor**
- **ALU**
- **LD/ST**
- **L2 Cache**
- **Memory Controller**
- **Fixed-Function Stage**
- **Programmable Stage**
- **Memory Hierarchy**
GPU Timing Simulator

- Fragment/Vertex Processors
  - Simple in-order 4-stage pipeline
  - Multi-warp execution
  - Vectorial ISA
  - Texture Sampling Units
GPU Power Model

• Based on McPAT
  – TEAPOT extensions:
    • Multiple data caches per core
    • Read-only caches
    • Specialized graphics hardware (texture sampling units...)
  – Output file in JSON format
  – Directly called from timing simulator

Configuration file
num_raster_units
num_geometry_units
num_fragment_procs
num_vertex_procs
num_warps_per_proc
...

Cycle-Accurate GPU Simulator

McPAT

GPU description
Area, Leakage
Activity Factors
Dynamic Power
Screen Power Model

- OLED displays
  - Consume different energy depending on the colors
  - Screen energy depends on the output generated by the GPU

- OLED-based displays power model
  - Provides three functions, $f(R)$, $f(G)$, $f(B)$, that map pixel intensity into energy consumption

Image Quality Assessment

• Image Quality Metrics
  – Based on **per-pixel errors**
    • MSE (Mean-Squared Error)
    • PSNR (Peak Signal-to-Noise Ratio)
  – Based on the **human visual perception system**
    • MSSIM (Mean Structural SIMilarity Index)

Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli.
“Image Quality Assessment: from Error Visibility to Structural Similarity”.

– Require **reference** noise-free image for comparison
– Evaluate distortion when trading quality for energy
1. Motivation

2. Simulation Infrastructure
   2.1. OpenGL ES Trace Generation
   2.2. GPU Functional Simulation
   2.3. Cycle-Accurate Timing Simulation
   2.4. Power Model
   2.5. Image Quality Assessment

3. Conclusions
Conclusions

• The TEAPOT toolset is tailored towards the mobile segment since it:
  – Runs unmodified **Android applications**
  – Estimates performance, **energy**, area and image quality of mobile graphics systems
  – Provides a **flexible** timing simulator, supporting Immediate-Mode and Tile-Based Deferred Rendering
  – Reports statistics:
    • Per Application, including Android OS (**full-system**)  
    • Per Frame  
    • Per Component: GPU, System Memory and Screen
TEAPOT: A Toolset for Evaluating Performance, Power and Image Quality on Mobile Graphics Systems

Jose-Maria Arnau  
Joan-Manuel Parcerisa

Polychronis Xekalakis

Computer Architecture Department  
Universitat Politecnica de Catalunya

Intel Labs  
Intel Corporation