

Paralldroid

Sergio Afonso
safonsof@ull.es

Alejandro Acosta
aacostad@ull.es

Francisco Almeida
falmeida@ull.es

ULL

Universidad
de La Laguna



Grupo de Computación de Altas Prestaciones

- Actualmente compuesto por 6 investigadores, de los cuales 4 son empleados permanentes en el Departamento de Ingeniería Informática de la Universidad de La Laguna, y 4 estudiantes de doctorado

Líneas de investigación

- Computación paralela en sistemas heterogéneos (CPUs multicore + GPUs + dispositivos móviles)
- Análisis y modelado de rendimiento
- Paralelización de aplicaciones científicas
- Herramientas y sistemas para la computación paralela
 - Programación basada en esqueletos: DPSKEL, LLC
 - Compiladores fuente-a-fuente: accULL, Paralldroid
 - Servicios web y computación en la nube para HPC: OpenCF
 - Herramientas para la instrumentación y análisis (TIA): CALL, EML

Información detallada en <http://cap.pcg.u11.es>

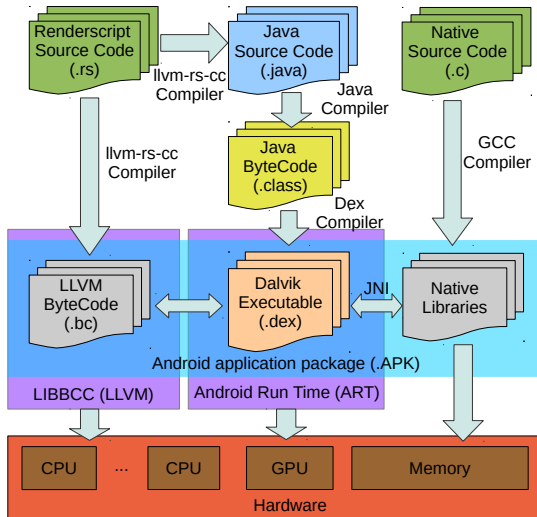
Introducción

- Mejora de las prestaciones de Systems on Chip debido al crecimiento del mercado de los smartphones
- Heterogeneidad entre diferentes dispositivos y entre los procesadores dentro de cada uno de ellos
- Dificultad de escritura de código de altas prestaciones multiplataforma
- Existen herramientas para obtener código acelerado automáticamente (OpenMP, OpenACC), pero no están diseñadas para su uso en el desarrollo de aplicaciones móviles
- Gran cantidad de aplicaciones de visión por computador y procesamiento de imágenes en móviles se benefician del procesamiento en paralelo

Samsung
Exynos
PROCESSOR

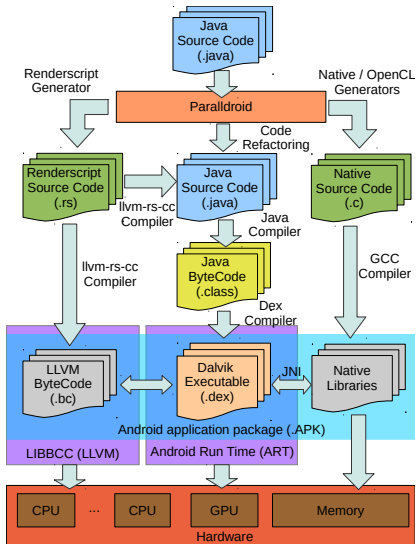


Android



- **Java:** Lenguaje principal y más sencillo de utilizar, con gran cantidad de herramientas disponibles
- **RenderScript:** Lenguaje para computación de altas prestaciones en Android, que permite paralelismo de datos
- **Nativo (JNI):** Útil para reutilizar código ya escrito en C/C++ y para acceder a librerías nativas del dispositivo

Paralldroid



- Unifica todos los modelos de desarrollo en Android: Puede generar código Renderscript, OpenCL y nativo
- Anotaciones ajustadas al paradigma de programación orientada a objetos que encajan más naturalmente en Java que OpenMP
- Facilita la escritura de métodos paralelos de forma intuitiva
- Extensión de OpenJDK que utiliza transformaciones de AST (Árbol Sintáctico Abstracto) para procesar, transformar y generar código

Ejemplo

```
@Target(OPENCL)
public class GrayScale {
    @Declare
    private float gMonoMult[] = {0.299f, 0.587f, 0.114f};

    @Map(T0)
    private int width;
    @Map(T0)
    private int height;

    public GrayScale(int width, int height) {
        this.width = width;
        this.height = height;
    }

    @Parallel
    public void run(@Input Bitmap src, @Output Bitmap out, @Index int x, @Index int y) {
        int pixel = src.getPixel(x, y);
        int acc;

        acc = (int)(Color.red(pixel) * gMonoMult[0]);
        acc += (int)(Color.green(pixel) * gMonoMult[1]);
        acc += (int)(Color.blue(pixel) * gMonoMult[2]);

        out.setPixel(x, y, Color.argb(Color.alpha(pixel), acc, acc, acc));
    }
}
```

Conclusiones

- Paralldroid facilita la aceleración de aplicaciones en Android a través de anotaciones Java y transformaciones de AST
- Nuestra implementación de las anotaciones resulta familiar para desarrolladores que conozcan con OpenMP, pero además se adaptan bien al paradigma de programación orientado a objetos
- Cada clase de una aplicación se puede implementar en un lenguaje separado, dependiendo del que proporcione mejores resultados para cada algoritmo concreto, de manera transparente
- Actualmente trabajamos en mejorar el rendimiento del backend para OpenCL

Agradecimientos: Este trabajo ha sido apoyado por el Ministerio de Educación, Cultura y Deporte a través de los proyectos TIN2011-24598 y TIN2016-78919-R, por la red CAPAP-H4 y la NESUS IC1315 COST Action.