

*Plan de trabajo:*  
*Paralelización de algoritmos*  
*irregulares en arquitecturas*  
*GPU*

Luis Arija Gonzalez

# 1. Descripción General y Objetivos:

Las arquitecturas basadas en unidades de procesamiento de gráficos (GPUs) se han convertido en los últimos años en una de las formas más populares y asequibles de llevar a cabo computación de altas prestaciones.

Sus características las hacen ideales para algoritmos paralelos con muy regulares y con escasa interacción entre las diferentes tareas, siendo menos evidente cómo sacar partido de ellas en algoritmos menos regulares.

En este trabajo estudiaremos el problema de implementar en GPUs este tipo de algoritmos.

- Aprender a desarrollar algoritmos paralelos sobre GPUs
- Realizar una o varias implementaciones de un algoritmo no regular no trivial en GPUs y analizar la eficiencia de las mismas.
- Extraer conclusiones que ayuden a buscar patrones algorítmicos útiles para resolver problemas similares.

## 2. Lista de tareas:

- Familiarización con el proyecto y planificación: 27 horas.
- Elección y aprendizaje del marco de programación de GPUs a utilizar -- OpenCL, CUDA, etc.: 40 horas.
- Elección y estudio del algoritmo a paralelizar -- en principio sería un algoritmo para resolver el problema de los "N cuerpos" usando octrees: 20 horas.
- Desarrollo de las diferentes implementaciones a comparar, incluyendo la posible adaptación de alguno de estos algoritmos para resolver problemas similares: 100 horas.
- Desarrollo de un front-end de visualización de estos algoritmos: 20 horas.
- Análisis de la eficiencia de las diferentes implementaciones: 20 horas.
- Memoria y presentación: 70 horas.

### 3. Diagrama de Gantt

Tarea\Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Familiarización con el proyecto y planificación	■	■																
Elección y aprendizaje del marco de programación de GPUs a utilizar		■	■	■														
Elección y estudio del algoritmo a paralelizar					■	■												
Desarrollo de las diferentes implementaciones a comparar, incluyendo la posible adaptación de alguno de estos algoritmos para resolver problemas similares						■	■	■	■	■	■	■						
Desarrollo de un front-end de visualización de estos algoritmos												■	■					
Análisis de la eficiencia de las diferentes implementaciones													■	■				
Memoria y presentación															■	■	■	■

## 4. Copia de la Propuesta de trabajo escrita por el Tutor

Nombre del trabajo:

=====

Paralelización de algoritmos irregulares en arquitecturas GPU

Resumen general del trabajo:

=====

Las arquitecturas basadas en unidades de procesamiento de gráficos (GPUs) se han convertido en los últimos años en una de las formas más populares y asequibles de llevar a cabo computación de altas prestaciones.

Sus características las hacen ideales para algoritmos paralelos con cálculos muy regulares y con escasa interacción entre las diferentes tareas, siendo menos evidente cómo sacar partido de ellas en algoritmos menos regulares.

En este trabajo estudiaremos el problema de implementar en GPUs este tipo de algoritmos.

Lista de objetivos concretos del trabajo:

=====

- Aprender a desarrollar algoritmos paralelos sobre GPUs.
- Realizar una o varias implementaciones de un algoritmo no regular no trivial en GPUs y analizar la eficiencia de las mismas.
- Extraer conclusiones que ayuden a buscar patrones algorítmicos útiles para resolver problemas similares.

Desglose de la dedicación total del trabajo en horas (297 horas en los Grados):

=====

=====

- Familiarización con el proyecto y planificación: 27 horas.
- Elección y aprendizaje del marco de programación de GPUs a utilizar

- OpenCL, CUDA, etc.: 40 horas.
- Elección y estudio del algoritmo a paralelizar -- en principio sería un algoritmo para resolver el problema de los "N cuerpos" usando octrees: 20 horas.
- Desarrollo de las diferentes implementaciones a comparar, incluyendo la posible adaptación de alguno de estos algoritmos para resolver problemas similares: 100 horas.
- Desarrollo de un front-end de visualización de estos algoritmos: 20 horas.
- Análisis de la eficiencia de las diferentes implementaciones: 20 horas.
- Memoria y presentación: 70 horas.

Lista de conocimientos previos recomendados para realizar el trabajo:

=====

Haber aprobado las asignaturas obligatorias de la materia

Programación. Recomendable haber cursado la asignatura optativa "El Arte de Programar".

Alumno exclusivo para esta propuesta (introduzca el usuario

UPM/matrícula del alumno):

=====

190324