

# Trabajo Práctico

## Radix Sort en OpenCL

Leonardo Scandolo (lscandolo@gmail.com)

R-222 Arquitectura del Computador

### Introducción

El objetivo del trabajo es que el alumno aprenda y adquiera experiencia en el uso de OpenCL para utilizar placas gráficas como dispositivos para hacer cómputos generales. Se espera que a través del desarrollo de este trabajo práctico comprenda las ventajas y desventajas del uso de dichos dispositivos, así como que obtenga el conocimiento necesario para poder juzgar cuando su uso es beneficioso para un proyecto. El trabajo consiste en implementar el algoritmo de ordenación llamado *Radix Sort* para que se ejecute puramente en OpenCL. La implementación buscada, que es detallada en el material que se le proveerá al alumno hace uso de todas las jerarquías de memoria, así como uso de algunas técnicas ampliamente usadas para programación de placas gráficas.

### Metodología

Para asegurar el acceso del alumno a una placa gráficas que compatible con OpenCL, se instalaron las librerías pertinentes en un servidor administrado por el Departamento de Ciencias de la Computación, y que todos los alumnos pueden acceder libremente a través de *ssh*.

También se le proveerá al alumno un programa de prueba con la inicialización básica de un dispositivo que implementa OpenCL, así como traspaso de memoria desde y hacia el dispositivo y un kernel básico para modificar los contenidos de la memoria alocada.

El alumno será provisto de artículos científicos y vínculos detallando la lógica del algoritmo de ordenamiento a implementar.

Una vez terminado el proyecto se le pedirá al alumno calcular los tiempos de ejecución del algoritmo para diferentes cantidades de elementos a ordenar, y compararlos con algoritmos de ordenación en procesadores convencionales (pueden utilizarse algoritmos provistos en la librería estándar de C para este propósito).

### Herramientas

Para ejecutar código en OpenCL el alumno deberá utilizar el servidor provisto para este propósito si no posee un dispositivo que implemente el estándar

OpenCL. Este servidor, llamado labdcc, se puede acceder desde el servidor *kleene* del departamento de computación (accesible por ssh en 'kleene.fceia.unr.edu.ar') a través de ssh. Pueden usarse las mismas credenciales existentes para conectarse a *kleene*.

Una vez en el directorio *home* del alumno, deberá agregarse la dirección de las librerías de OpenCL al ambiente de programación. Esto se hará ejecutando el comando:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64
```

o simplemente agregandolo al archivo `~/bashrc` para que se ejecute automáticamente cada vez que el alumno ingrese al servidor.

El alumno tendrá acceso a código de ejemplo que incluye un archivo *Makefile* para tener referencia acerca de cómo compilar el código que utilice OpenCL.

Asimismo contará con artículos que proveen una explicación de las características de OpenCL, el algoritmo de ordenación que se espera que implemente y el algoritmo de suma de prefijos que es central en la implementación de Radix Sort en OpenCL.

## Algunas ideas sobre el proyecto

- El primer paso debería ser tener una implementación del algoritmo de suma de prefijos (parallel scan), local y global, ya que se utilizará como parte del algoritmo de ordenamiento.
- El algoritmo debería utilizar el grupo de trabajo de tamaño más grande posible, de manera de obtener el mayor beneficio del paralelismo provisto por las placas gráficas.
- El código de los kernel a ejecutar debería estar en un archivo diferente al código C del proyecto, y no como un string dentro del código (como se presenta en el archivo de prueba), para facilidad de edición.
- Puede hacerse el algoritmo primero para ordenar arreglos cuya longitud sea una potencia de 2 para facilitar algunas partes del código, y luego modificarse para soportar arreglos de cualquier tamaño.

## Entrega del Trabajo

El trabajo será evaluado por la cátedra mediante una presentación en computadora. El alumno debe entregar un informe de al menos dos páginas incluyendo datos académicos (integrantes del grupo, legajos, fechas) y reportando problemas y soluciones encontradas durante la realización del trabajo y posibles extensiones al mismo.

## Material y Referencias

- <http://www.khronos.org/registry/cl/sdk/1.1/docs/man/xhtml/>
- [http://http.developer.nvidia.com/GPUGems3/gpugems3\\_ch39.html](http://http.developer.nvidia.com/GPUGems3/gpugems3_ch39.html)
- <https://code.google.com/p/back40computing/wiki/RadixSorting>