Programación concurrente y paralela

**Documentación: Proyecto Final**

Jens Wöhrle – A01670506

Salvador Pámanes – A01195166

**La idea en general**

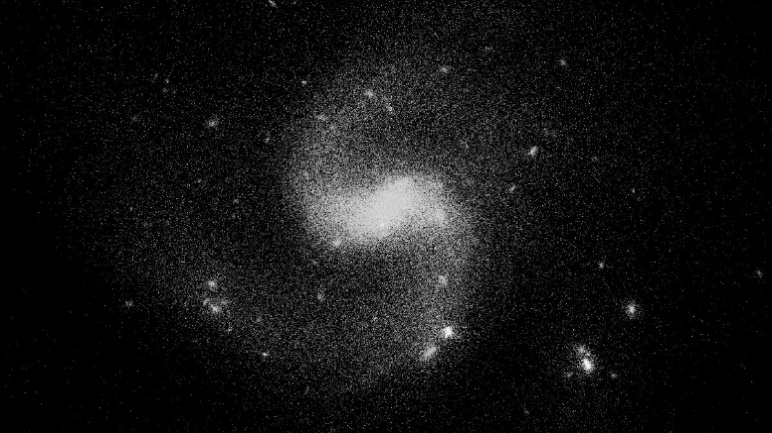
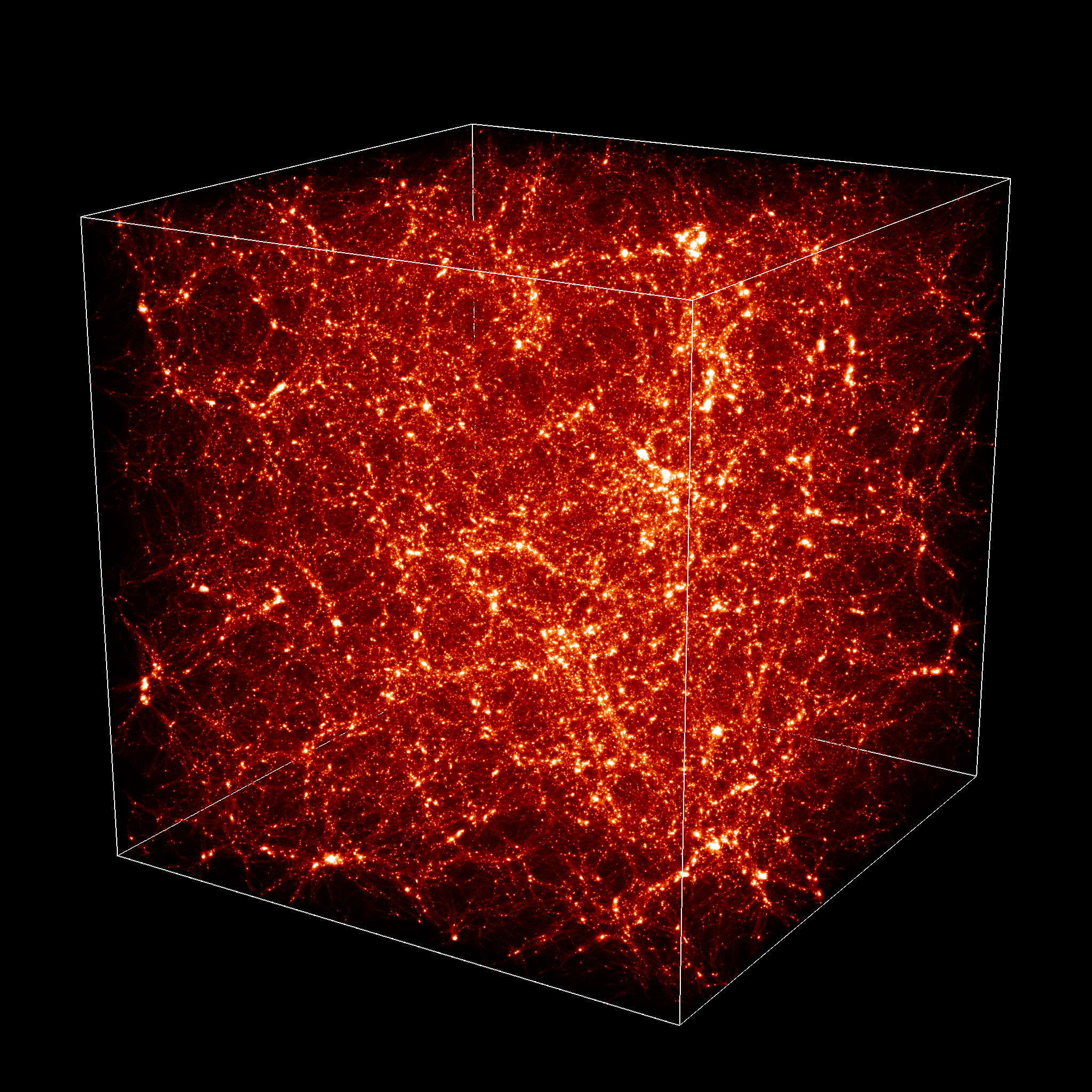
Proponemos una simulación física de partículas tipo n-Body. Creemos que ésta es una aplicación muy interesante que implica muchos cálculos sobre vectores de datos.

Una optimización útil para reducir el número de interacciones a calcular es usar octrees o algoritmos como Barnes-Hut, que agrupan cuerpos cercanos y usan su centro de masa para otros cálculos.

Quisiéramos usar CUDA u OpenCL para realizar este proyecto.

Nuestros objetivos iniciales son lograr una simulación en tiempo real, con un número de partículas mínimo de 1000, en 2D. De ser posible, queremos generalizar a 3D y lograr más partículas. El dibujo de las mismas se puede hacer en OpenGL con simples puntos en las coordenadas calculadas.

Algunas fuentes de información. Unas incluyen ejemplos de referencia, con programas cuadráticos (sin la optimización).



**Ejecución**

Hemos programado el proyecto en C++ con OpenCL y OpenGL de modo que para la compilación necesita estos paquetes. El Testing y la programación hicimos en Linux.

**Programación en detalle**

**Nbody.ocl:**

Aquí podemos encontrar el algoritmo de la simulación en general con la función nbody(…) . Esta función ejecutamos para cada estrella en la simulación que calcula su nueva posición con la dependencia de las posiciones de las otras estrellas. Para hacerlo, utilizamos la funcione “accDelta” que calcula, la nueva aceleración de la estrella.

También utilizamos “vectoring” aquí, de modo que trabajamos con “float4” que es un vector de 4 valores de float para su posición en X, Z, Y y su peso (no se cambia).

**NBody.cpp**

Aquí hacemos los “calls” de OpenCL para cada estrella. El código es muy similar como los ejemplos del semestre para crear el contexto, activar el Kernel y transferir los datos, etc.. Para una mejor estructura tenemos NBody.hpp donde declaramos los valores para trabajar en OpenCL

**Main.cpp**

Esta utiliza OpenGL para visualizar las estrellas y hace los calls a “NBody.cpp” para empezar la visualización.

**Referencias**

<http://http.developer.nvidia.com/GPUGems3/gpugems3_ch31.html>

<http://haifux.org/lectures/267/OpenCL_Dos_and_Donts.pdf>

<https://sites.google.com/a/nirmauni.ac.in/cudacodes/cuda-projects/proj-6-tree-based-barnes-hut-n-body-algorithm>