

8/04/2019

TG2

**testing and developing**

Borja Ordóñez Bello, Javier Luque Moreno, Adrián López Godoy, Daniel Calin Stanus, Ignacio Burgos Lucha

**Scientific Cards**



Contenido

[1. Autores del trabajo, planificación y entrega 3](#_Toc5799652)

[1.1 Autores 3](#_Toc5799653)

[1.2 Planificación 3](#_Toc5799654)

[1.3 Entrega 3](#_Toc5799655)

[2. Descripción de las tecnologías 4](#_Toc5799656)

[2.1 Descripción de la tecnología A, CUDA 4](#_Toc5799657)

[2.2 Descripción de la tecnología B, OPENCL 4](#_Toc5799658)

[3. Criterios de comparación 5](#_Toc5799659)

[3.1 Categoría A: Genéricos 5](#_Toc5799660)

[3.1.1 Criterio A.1: Coste 5](#_Toc5799661)

[3.1.2 Criterio A.2: Escalabilidad 5](#_Toc5799662)

[3.1.3 Criterio A.3: Facilidad de implantación 5](#_Toc5799663)

[3.1.4 Criterio A.4: Comunidad de ayuda 5](#_Toc5799664)

[3.1.5 Criterio A.5: Idiomas 5](#_Toc5799665)

[3.1.6 Criterio A.6: Popularidad 5](#_Toc5799666)

[3.1.7 Criterio A.7: Madurez 5](#_Toc5799667)

[3.1.8 Criterio A.8: Coste de Formación 5](#_Toc5799668)

[3.2 Categoría B: Hardware 5](#_Toc5799669)

[3.2.1 Criterio B.1: Rendimiento 5](#_Toc5799670)

[3.2.2 Criterio B.2: Potencia gráfica 5](#_Toc5799671)

[3.2.3 Criterio B.3: Requisitos hardware 5](#_Toc5799672)

[3.3 Categoría C: Software 5](#_Toc5799673)

[3.3.1 Criterio C.1: Windows 5](#_Toc5799674)

[3.3.2 Criterio C.2: Mac 5](#_Toc5799675)

[3.3.3 Criterio C.3: Linux 5](#_Toc5799676)

[3.3.4 Criterio C.4: IDEs 5](#_Toc5799677)

[3.3.5 Criterio C.5: Lenguajes de programación 5](#_Toc5799678)

[3.3.6 Criterio C.6: Instalación 6](#_Toc5799679)

[3.3.7 Criterio C.7: Facilidad de uso 6](#_Toc5799680)

[3.3.8 Criterio C.8: Seguridad 6](#_Toc5799681)

[3.3.9 Criterio C.9: Licencia 6](#_Toc5799682)

[3.4 Categoría D: Soporte y mantenimiento 6](#_Toc5799683)

[3.4.1 Criterio D.1: Soporte 6](#_Toc5799684)

[3.4.2 Criterio D.2: Mantenimiento 6](#_Toc5799685)

[4. Evaluación de los criterios por tecnología 6](#_Toc5799686)

[4.1 Evaluación de los criterios para CUDA 6](#_Toc5799687)

[4.2 Evaluación de los criterios para OpenCL 7](#_Toc5799688)

[5. Comparación de las tecnologías 9](#_Toc5799689)

[6. Recomendaciones 14](#_Toc5799690)

[6.1 Situación 1 14](#_Toc5799691)

[6.1.1 Descripción de la situación 14](#_Toc5799692)

[6.1.2 Recomendación de tecnología a utilizar 14](#_Toc5799693)

[6.2 Situación 2 15](#_Toc5799694)

[6.2.1 Descripción de la situación 15](#_Toc5799695)

[6.2.2 Recomendación de tecnología a utilizar 15](#_Toc5799696)

[7.Conclusiones 16](#_Toc5799697)

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

El presente documento ha sido elaborado por:

* Daniel Stanus
* Borja Ordóñez
* Javier Luque
* Ignacio Burgos
* Adrián López

## 1.2 Planificación

Nuestra planificación ha sido desarrollada íntegra de forma online, y viene detallada en el siguiente enlace:

<https://app.teamweek.com/#pp/WJTiRMNrFoLTAAbn_iJHP6ZArunCmATd>

## 1.3 Entrega

El enlace de Github donde se encuentra el presente documento es el siguiente:

<https://github.com/algoy/DTE/tree/master/TG2T3>

# 2. Descripción de las tecnologías

El uso de las diferentes tecnologías que incorporan las tarjetas gráficas para sacar su potencial computacional se puede aplicar en diferentes ámbitos: matemático, científico, médico….

Las tecnologías que hemos investigado son CUDA y OpenCL. CUDA es para las tarjetas gráficas de la marca NVIDIA y OpenCL para cualquier marca de tarjetas gráficas, al ser un estándar abierto.

## 2.1 Descripción de la tecnología A, CUDA

CUDA es una arquitectura de cálculo en paralelo de NVIDIA que aprovecha la potencia de la GPU para proporcionar un incremento en el rendimiento del sistema. Esta potencia se utiliza para la computación en paralelo. CUDA también incluye un compilador y herramientas de desarrollo creadas por NVIDIA que ayudan a los programadores a usar diferentes lenguajes de programación como por ejemplo C, C++, Python y Java.

Esta tecnología funciona con todas las tarjetas de NVIDIA a partir de la serie G8X incluyendo las familias GeForce, Quadro, ION y Tesla.

CUDA intenta explotar las ventajas de las GPU frente a las CPU de propósito general utilizando el paralelismo que ofrecen sus múltiples núcleos, que permiten el lanzamiento de un altísimo número de hilos simultáneos. Por ello, si una aplicación está diseñada utilizando numerosos hilos que realizan tareas independientes (que es lo que hacen las GPU al procesar gráficos).

## 2.2 Descripción de la tecnología B, OPENCL

Junto a una interfaz de programación y de un lenguaje (C) permite crear aplicaciones con paralelismo a nivel de datos y tareas que pueden ejecutarse tanto en unidades de procesamiento (CPU) como en unidades de procesamiento grafico (GPU).

OpenCL provee del API que permite a los programas acceder a múltiples procesadores de forma simultánea para lograr el procesamiento en paralelo. Al distribuir el procesamiento ente múltiples procesadores mejora el rendimiento del programa.

Para que OpenCL funcione, el hardware y el software deben de soportar la API. Actualmente las tarjetas gráficas de AMD, NVIDIA e INTEL soportan este estándar.

# 3. Criterios de comparación

## 3.1 Categoría A: Genéricos

### 3.1.1 Criterio A.1: Coste

Nombre del criterio: Coste.

Descripción: Coste de implantación de los programas de programación.

Tipo de valor: Numérico.

### 3.1.2 Criterio A.2: Escalabilidad

Nombre del criterio: Escalabilidad.

Descripción: Capacidad del sistema para reaccionar y adaptarse sin perder la calidad, manejar el crecimiento continuo o para hacerse más grande sin perder calidad.

Tipo de valor: Escala 1-5, 1 menor 5 mayor.

### 3.1.3 Criterio A.3: Facilidad de implantación

Nombre del criterio: Implantación.

Descripción: Complejidad para instalar el sistema.

Tipo de valor: Texto libre (Alta, Media, Baja).

### 3.1.4 Criterio A.4: Comunidad de ayuda

Nombre del criterio: Comunidad.

Descripción: Es un espacio interactivo donde se comparte con otras personas sus inquietudes, sus ideas, sus sugerencias, hacer comentarios, reportar problemas y opinar.

Tipo de valor: Booleano (Si/No).

### 3.1.5 Criterio A.5: Idiomas

Nombre del criterio: Idiomas.

Descripción: Idiomas disponibles para los sistemas.

Tipo de valor: Texto libre.

### 3.1.6 Criterio A.6: Popularidad

Nombre del criterio: Popularidad.

Descripción: Es el grado de aceptación de los sistemas.

Tipo de valor: Texto libre (Baja, Media, Alta).

### 3.1.7 Criterio A.7: Madurez

Nombre del criterio: Madurez.

Descripción: Calidad y ventaja competitiva respecto otros sistemas.

Tipo de valor: Texto libre.

### 3.1.8 Criterio A.8: Coste de Formación

Nombre del criterio: Coste de Formación.

Descripción: Precio para formar un empleado para operar con el sistema.

Tipo de valor: Numérico.

## 3.2 Categoría B: Hardware

### 3.2.1 Criterio B.1: Rendimiento

Nombre del criterio: Rendimiento.

Descripción: Capacidad de un sistema para llevar a cabo una tarea.

Tipo de valor: Texto libre.

### 3.2.2 Criterio B.2: Potencia gráfica

Nombre del criterio: Potencia gráfica.

Descripción: Capacidad de la grafica para visualizar la información al usuario.

Tipo de valor: Texto libre (Baja, Media, Alta).

### 3.2.3 Criterio B.3: Requisitos hardware

Nombre del criterio: Requisitos hardware

Descripción: Condiciones que debe tener el sistema para operar con el hardware.

Tipo de valor: Texto libre.

## 3.3 Categoría C: Software

### 3.3.1 Criterio C.1: Windows

Nombre del criterio: Windows.

Descripción: Indica la disponibilidad de la aplicación en el sistema operativo.

Tipo de valor: Booleano (Si/No).

### 3.3.2 Criterio C.2: Mac

Nombre del criterio: Mac.

Descripción: Indica la disponibilidad de la aplicación en el sistema operativo.

Tipo de valor: Booleano (Si/No).

### 3.3.3 Criterio C.3: Linux

Nombre del criterio: Linux.

Descripción: Indica la disponibilidad de la aplicación en el sistema operativo.

Tipo de valor: Booleano (Si/No).

### 3.3.4 Criterio C.4: IDEs

Nombre del criterio: IDEs.

Descripción: Servicios que facilitan el desarrollo de software.

Tipo de valor: Texto libre.

### 3.3.5 Criterio C.5: Lenguajes de programación

Nombre del criterio: Lenguajes de programación.

Descripción: Lenguaje de programación que se va a emplear para el sistema.

Tipo de valor: Texto libre.

### 3.3.6 Criterio C.6: Instalación

Nombre del criterio: Instalación.

Descripción: Guía para instalar el software.

Tipo de valor: Texto libre.

### 3.3.7 Criterio C.7: Facilidad de uso

Nombre del criterio: Facilidad de uso

Descripción: Capacidad con la que se puede utilizar el software.

Tipo de valor: Texto libre.

### 3.3.8 Criterio C.8: Seguridad

Nombre del criterio: Seguridad.

Descripción: Confianza en el software y ausencia de riesgo.

Tipo de valor: Texto libre.

### 3.3.9 Criterio C.9: Licencia

Nombre del criterio: Licencia

Descripción: Autorización para hacer una serie de tareas o procesos con el software.

Tipo de valor: Texto libre.

## 3.4 Categoría D: Soporte y mantenimiento

### 3.4.1 Criterio D.1: Soporte

Nombre del criterio: Soporte.

Descripción: Capacidad de solucionar problemas del sistema.

Tipo de valor: Booleano (Si/No).

### 3.4.2 Criterio D.2: Mantenimiento

Nombre del criterio: Mantenimiento.

Descripción: El conjunto de operaciones necesarias para el funcionamiento correcto del sistema.

Tipo de valor: Booleano (Si/No).

# 4. Evaluación de los criterios por tecnología

## 4.1 Evaluación de los criterios para CUDA

Los criterios a seguir y evaluar para la presente tecnología están recogidos en la siguiente tabla:

|  |  |
| --- | --- |
| CRITERIO | EVALUACIÓN |
| A.1: Coste | 0€ (solo el hardware) |
| A.2: Escalabilidad | 3 |
| A.3: Facilidad de implantación | Simple |
| A.4: Comunidad de ayuda | Sí |
| A.5: Idiomas | Inglés |
| A.6: Popularidad | Alta |
| A.7: Madurez | Alta: desde junio de 2007 |
| A.8: Coste de formación | 115.60€ |
| B.1: Rendimiento | Alto |
| B.2: Potencia grafica | Limitada |
| B.3: Requisitos hardware | Soporte desde GeForce 9800 GTX (año 2008) |
| C.1: Windows | Soporte desde Windows Server 2012 R2 con arquitectura x86\_64 |
| C.2: Mac | Soporte desde Mac OS High Sierra 10.13 con arquitectura x86\_64 |
| C.3: Linux | Soporte con arquitecturas x86\_64 y ppc64le |
| C.4: IDEs | Visual Studio (2010, 2012, 2013, 2015, 2017) |
| C.5: Lenguajes de programación | C, C++, Fortran |
| C.6: Instalación | Simple |
| C.7: Facilidad de uso | Media |
| C.8: Seguridad | 4 |
| C.9: Licencia | Limitada (sujeto a EULA nVidia) |
| D.1: Soporte | Sí |
| D.2: Mantenimiento | Sí |

## 4.2 Evaluación de los criterios para OpenCL

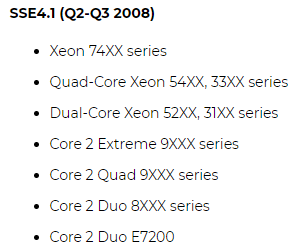
Los criterios a seguir y evaluar para la presente tecnología están recogidos en la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| CRITERIO | EVALUACIONES |  |
| A.1: Coste | 0€ (solo el hardware) |  |
| A.2: Escalabilidad | 5 |  |
| A.3: Facilidad de implantación | Media |  |
| A.4: Comunidad de ayuda | Si |  |
| A.5: Idiomas | Inglés |  |
| A.6: Popularidad | Baja |  |
| A.7: Madurez | Media: desde 2009 y tiene 6 versiones, la última es OpenCL 2.2 |  |
| A.8: Coste de formación | 1850€ |  |
| B.1: Rendimiento | Medio alto |  |
| B.2: Potencia grafica | Media |  |
| B.3: Requisitos hardware | Soporte desde GeForce 100 series en NVIDIA, Radeon 4350 para arriba en ATI/AMD. |  |
| C.1: Windows | Soporde desde Windows 8.1 32-bit Spring Refresh running on Intel HD Graphics 5500/6000 |  |
| C.2: Mac | Soporte desde Mac OS X 10.8 with Intel CPU |  |
| C.3: Linux | Soporte desde Linux 2.6 64-bit with AMD OpenCL 2.0 |  |
| C.4: IDEs | OpenCL Studio, Intel® SDK for OpenCL™ Applications, Intel® FPGA Development Tools, NVIDIA OpenCL SDK, Portable OpenCL (pocl) |  |
| C.5: Lenguajes de programación | C, C++, Python, Ruby, .NET |  |
| C.6: Instalación | Complicada |  |
| C.7: Facilidad de uso | Bajo |  |
| C.8: Seguridad | 2 |  |
| C.9: Licencia | Libre |  |
| D.1: Soporte | Si |  |
| D.2: Mantenimiento | No |  |

Imagen que contiene captura de pantalla

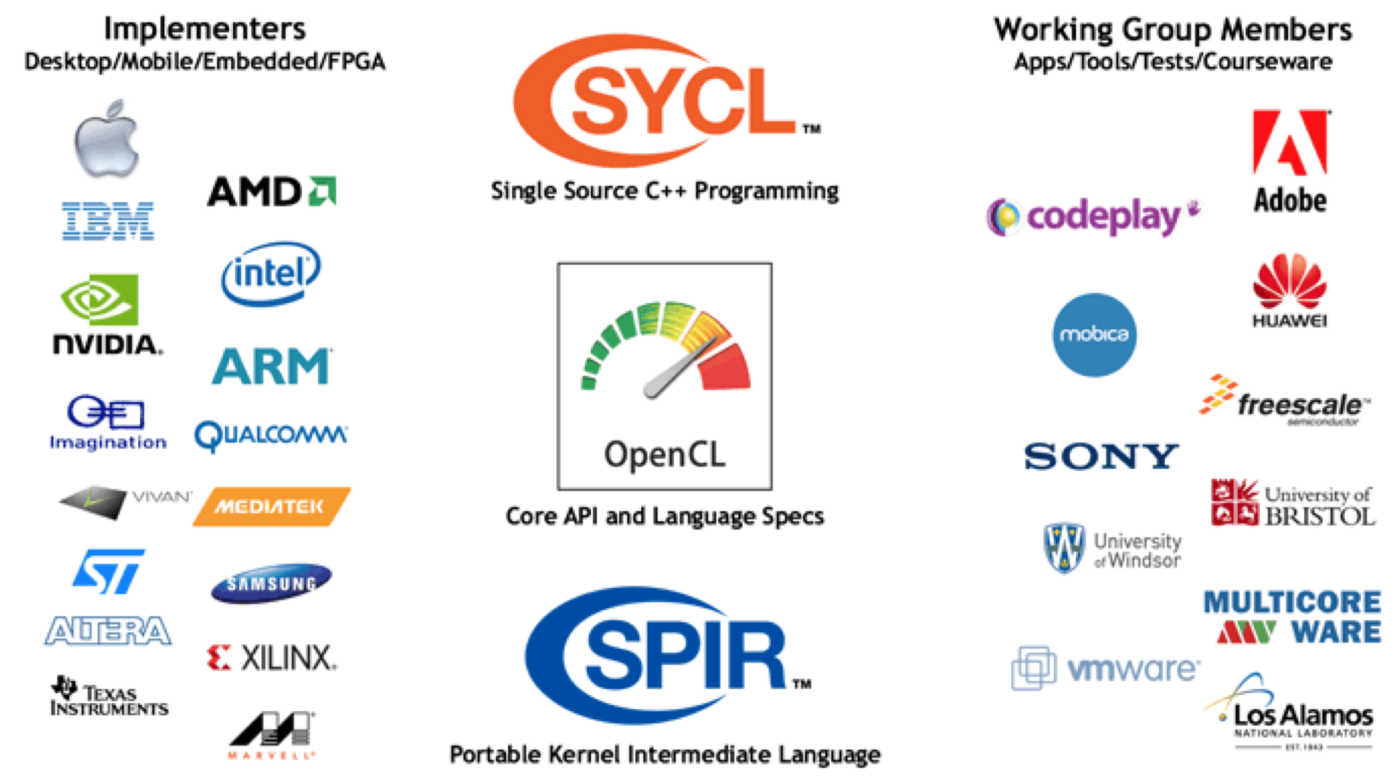
Descripción generada automáticamente Fuente:<https://medium.com/@stevenn.hansen/what-are-we-actually-doing-on-cuda-and-opencl-1020553bfcdb>

Requisitos hardware:

Para Intel: 

Coste formación:

Hemos encontrado como apropiado para profesionales, el siguiente curso, valorado en 1850€ - <http://www.ac6-formation.com/cours.php/cat_LANG/ref_L9.xphp>

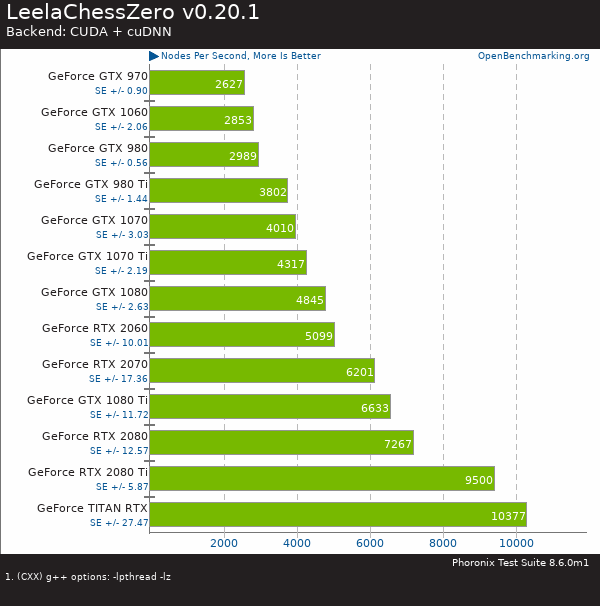
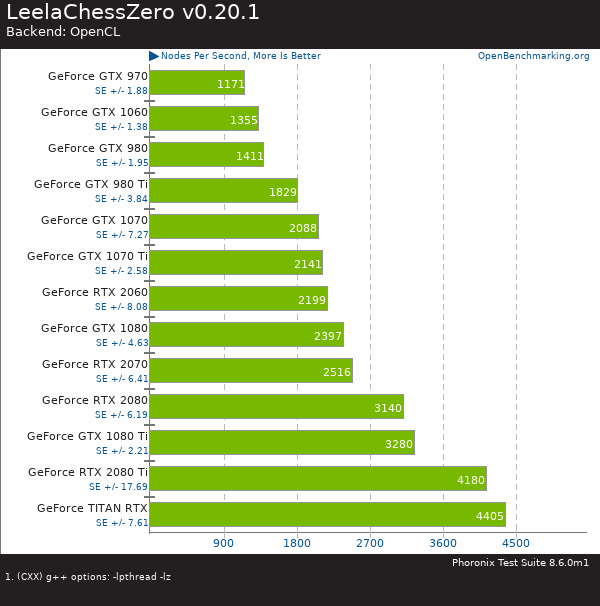


# 5. Comparación de las tecnologías

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CRITERIOS | CUDA | OpenCL | COMENTARIOS |  |
| A.1: Coste | 0€ | 0€ | Ambos son gratis, entonces nos da igual la elección. |  |
| A.2: Escalabilidad | 3 | 5 | Preferimos OpenCL por tener una mayor escalabilidad. |  |
| A.3: Facilidad de implantación | Simple | Media | CUDA es más fácil para implantarlo. |  |
| A.4: Comunidad de ayuda | Sí | Si | Ambas tienen comunidad de ayuda. |  |
| A.5: Idiomas | Inglés | Inglés |  |  |
| A.6: Popularidad | Alta | Baja | NVIDIA tiene la mayor cuota del mercado |  |
| A.7: Madurez | Alta: desde junio de 2007 | Media: desde 2009 y tiene 6 versiones, la última es OpenCL 2.2 | CUDA tiene mas tiempo en el mercado. |  |
| A.8: Coste de formación | 115.60€ | 1850€ |  |  |
| B.1: Rendimiento | Alto | Medio alto | CUDA se utiliza más en superordenadores por su mayor eficiencia. |  |
| B.2: Potencia grafica | Limitada | Media | NVIDIA tiene las graficas mas potentes del mercado, pero OpenCL también puede ser usada en graficas NVIDIA |  |
| B.3: Requisitos hardware | Soporte desde GeForce 9800 GTX (año 2008) | Soporte desde GeForce 100 series en NVIDIA, Radeon 4350 para arriba en ATI/AMD, admite Intel. | OpenCL funciona con un hardware menor. |  |
| C.1: Windows | Soporte desde Windows Server 2012 R2 con arquitectura x86\_64 | Soporte desde Windows 8.1 32-bit Spring Refresh running on Intel HD Graphics 5500/6000 | CUDA funciona con una versión de Windows más antigua. |  |
| C.2: Mac | Soporte desde Mac OS High Sierra 10.13 con arquitectura x86\_64 | Soporte desde Mac OS X 10.8 with Intel CPU | OpenCL funciona con una versión de Mac OS más antigua. |  |
| C.3: Linux | Soporte con arquitecturas x86\_64 y ppc64le | Soporte desde Linux 2.6 64-bit with AMD OpenCL 2.0 | CUDA funciona con una versión de Linux más antigua. |  |
| C.4: IDEs | Visual Studio (2010, 2012, 2013, 2015, 2017) | OpenCL Studio, Intel® SDK for OpenCL™ Applications, Intel® FPGA Development Tools, NVIDIA OpenCL SDK, Portable OpenCL (pocl) | OpenCL tiene más IDEs, más libertad |  |
| C.5: Lenguajes de programación | C, C++, Fortran | C, C++, Python, Ruby, .NET | En OpenCL soporta más lenguajes de programación. |  |
| C.6: Instalación | Simple | Complicada | Es más fácil hacer funcionar las gráficas con CUDA. |  |
| C.7: Facilidad de uso | Media | Bajo | Por lo anterior es más fácil utilizarlo. |  |
| C.8: Seguridad | 4 | 2 | Por la naturaleza abierta de OpenCL, es más vulnerable. |  |
| C.9: Licencia | Limitada (sujeto a EULA nVidia) | Libre | Es mejor uno libre. |  |
| D.1: Soporte | Sí | Si | Las dos reciben actualizaciones. |  |
| D.2: Mantenimiento | Sí | No | NVIDIA si ofrece soporte. |  |

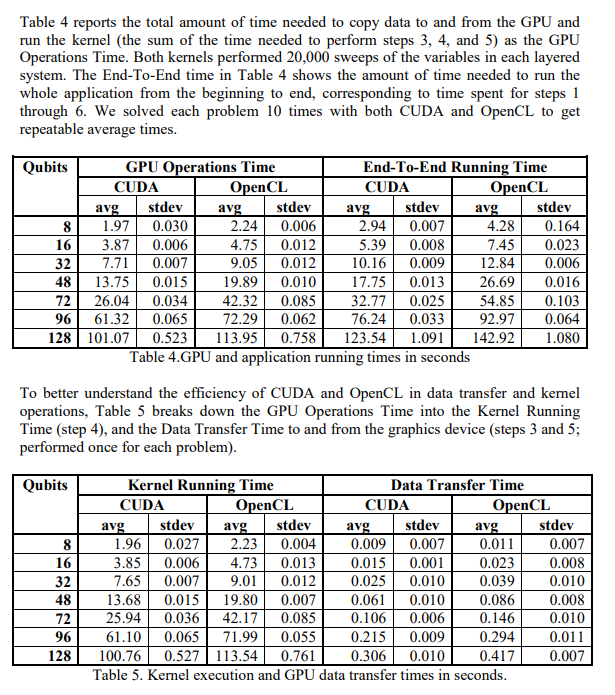
**Rendimiento:**

Según las siguientes pruebas, la tecnología CUDA posee un mayor rendimiento de media, al ejecutar la misma aplicación sobre el mismo hardware:



Fuente: <https://www.phoronix.com/scan.php?page=news_item&px=LCZero-NVIDIA-Benchmarks>

**Potencia gráfica:**

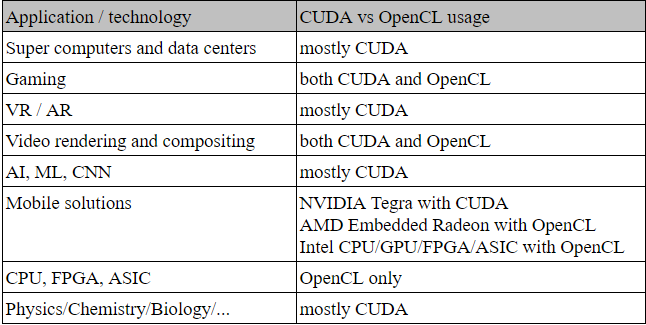


Fuente: <https://arxiv.org/ftp/arxiv/papers/1005/1005.2581.pdf>

# 6. Recomendaciones

Las dos tecnologías, aunque buscan un fin último similar (el aprovechamiento de la potencia de las GPUs modernas para realizar operaciones de cómputo), tienen diferentes ámbitos de uso principales.

Viene bastante bien resumiendo en la siguiente tabla, en inglés:



Fuente:<https://medium.com/@stevenn.hansen/what-are-we-actually-doing-on-cuda-and-opencl-1020553bfcdb>

Como vemos, para supercomputación científica se usa sobre todo CUDA, y para computación basado en CPU, OpenCL.

## 6.1 Situación 1

### 6.1.1 Descripción de la situación

Un laboratorio de la Universidad de Alcalá decide montar un servidor dedicado a investigar el bosón de Higgs. Necesita un equipo con alto poder de cómputo para investigar y adscribirse a redes científicas conjuntas de investigación. Va a ser un gran desembolso, por lo que investigan qué les conviene más.

### 6.1.2 Recomendación de tecnología a utilizar

En este caso, como vimos en la tabla anterior, la mejor opción es Nvidia CUDA, por su mayor madurez de versiones, y soporte de la comunidad oficial de Nvidia, así como por haber mayor documentación.

|  |  |  |
| --- | --- | --- |
| **Criterios relevantes para la decisión** | **Ventajas CUDA** | **Ventajas OpenCL** |
| Madurez | Alta, tiene más versiones y más depuradas que OpenCL. | - |
| Licencia | Sujeta al EULA de Nvidia (privativo) | Licencia y sistema abierto. |
| Soporte | Si | Si |
| Mantenimiento | Nvidia si ofrece soporte | - |
| Instalación | Sencilla, autoinstalador | - |
|  |  |  |

## 6.2 Situación 2

### 6.2.1 Descripción de la situación

El laboratorio de la asignatura optativa “Multimedia” de 4º, en la Universidad de Alcalá, está deseando comprar un equipo nuevo para el laboratorio, pero no dispone de gran presupuesto.

El uso principal será renderizado y composición de vídeo. Están sopesando qué equipo comprar, y qué tecnología usar, si CUDA u OpenCL.

### 6.2.2 Recomendación de tecnología a utilizar

En este caso, para este uso, lo recomendable sería comprar una GPU AMD Radeon, preferiblemente una RX580, por su menor precio que una Nvidia equivalente y menor precio. Además, OpenCL es bastante usada y eficiente en este aspecto de cómputo.

|  |  |  |
| --- | --- | --- |
| **Criterios relevantes para la decisión** | **Ventajas CUDA** | **Ventajas OpenCL** |
| Coste |  | Si, ofrece menor coste al poder usar soluciones AMD o incluso integradas (INTEL) |
| Licencia |  | Sistema totalmente abierto |
| Madurez | Alta, tiene más versiones |  |
| Mac |  | Si |

# 7.Conclusiones

A la hora de comparar dos tecnologías similares, debemos tener en cuenta una serie de factores que serán determinantes en cada situación. No podemos concluir que una sea mejor que otra, aún con los números en la mano:

En la situación descrita, donde la escuela necesitaba un equipo potente para renderizar y que sufría un problema de presupuesto (demasiado comedido), solo el coste y la mayor disponibilidad de opciones son variables más que recomendables para escoger esa opción sobre CUDA.

Sin embargo, si buscamos rendimiento puro en renderizado, deberíamos escoger la solución que más potencia bruta nos otorgue, que en este caso, a día de hoy, no deja de ser Nvidia con sus soluciones Quadro y su arquitectura CUDA.