UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Revista Científica

http://revistas.udistrital.edu.co/ojs/index.php/revcie/index

CIDC

Artículo de revisión

Plataforma computacional sobre Amazon Web Services (Aws) de renderizado distribuido

Computational Platform About Amazon Web Services (Aws) Distributed Rendering

Plataforma de computação sob Amazon Web Services AWS de renderização distribuida

Gabriel Rojas-Albarracín[1]
Jorge Páramo-Fonseca[2]
Cindy Hernández-Merchán[3]

Recibido: mayo de 2017 **Aceptado:** agosto de 2017

Para citar este artículo: Rojas-Albarracín, G., Páramo-Fonceca, J., y Henández-Merchán, C. (2017). Plataforma computacional sobre Amazon Web Services (Aws) de renderizado distribuido. *Revista Científica*, 30 (3), 252-262.
Doi: https://doi.org/10.14483/23448350.12362

Resumen

En la actualidad se ha creado una dinámica en la cual las personas exigen una mayor calidad de imagen en diferentes medios (juegos, películas, animaciones, entre otros). Por lo general, una mejor definición requiere el procesamiento de imágenes de mayor tamaño, esto trae consigo la necesidad de aumentar la capacidad de cálculo. El presente artículo expone un caso de estudio en el cual se muestra la implementación de una plataforma de bajo costo, sobre la nube de Amazon, para el procesamiento (renderizado) de imágenes y animaciones de forma paralela.

Palabras clave: AWS, alto procesamiento (HPC), LuxRender, procesamiento de imágenes (renderizado).

Abstract

Today has created a dynamic in which people require higher image quality in different media formats (games, movies, animations). Further definition usually requires image processing larger; this brings the need for increased computing power. This paper presents a case study in which the implementation of a low-cost platform on the Amazon cloud for parallel processing of images and animatios.

Keywords: AWS, high processing (HPC), LuxRender, image processing (Render).

Resumo

Hoje em dia tem-se criado uma dinâmica em que as pessoas exigem maior qualidade de imagem em diferentes meios (jogos, filmes, animações). O aumento da qualidade de definição geralmente requer processamento de imagem maior tamanho, isso traz consigo a necessidade de aumentar a capacidade de cálculo computacional. Este artigo apresenta um estudo de caso com a implementação de uma plataforma de baixo custo sob a computação em nuvem

1. Universidad de Cundinamarca. Bogotá-Colombia. Contacto: lgabrielrojas@ucundinamarca.edu.co
2. Universidad de Cundinamarca. Bogotá-Colombia. Contacto: jparamo@ucundinamarca.edu.co
3. Universidad de Cundinamarca. Bogotá-Colombia. Contacto: chcindyl3@gmail.com

da Amazon Web Services (AWS) para o processamento de imagens e animações de forma paralela.

Palavras-chaves: AWS, Alto processamento (HPC), LuxRender, processamento de imagens (Renderização).

Introducción

Para procesar una imagen o animación sintética, es decir, creada enteramente por ordenador, es necesario aplicar diferentes ecuaciones matemáticas y físicas que permiten obtener resultados tan impresionantes que pueden ser confundidos con fotos reales. Como es de esperarse, tanto mayor sea la resolución o definición de un contenido multimedia, mayor será la cantidad de operaciones que debe realizar. Cálculo de efectos lumínicos, sombras y texturas, por mencionar algunas, son las tareas habituales en el proceso de renderización de un contenido digital.

Todo lo anterior trae consigo un aumento en los tiempos necesarios para obtener los resultados del procesamiento de una imagen o animación. Para disminuir esos tiempos se pensó en utilizar arquitecturas computacionales de altas prestaciones, que permiten utilizar sistemas de multiprocesamiento simétrico o en paralelo, dividiendo tareas en núcleos conectados a una red que trabaja a una alta velocidad. El presente artículo se divide en cuatro secciones: 1) antecedentes, donde serán presentados estudios similares tanto en temática como en el uso de las tecnologías aquí expuestas; 2) enfoque técnico, donde se presentará el modelo arquitectónico de la plataforma implementada y el flujo de procesos que realiza; 3) pruebas y resultados, en esta sección serán presentados con detalle las características del caso de estudio, las fuentes, y los datos obtenidos luego de realizar el procesamiento; y 4) conclusiones, donde finalmente serán analizados los resultados generados en la sección anterior.

Antecedentes

Para el desarrollo de este trabajo se identificaron proyectos similares tanto en el resultado final como en el uso de las tecnologías.

Diseño de clústeres computarizados con algoritmos de renderizado para la construcción de modelos tridimensionales a través del uso de herramientas de OpenSource

El objetivo de este trabajo consistió en optimizar el tiempo de renderizado de un recorrido virtual 3D utilizando un clúster de computadoras. Se utilizó cómputo paralelo, además de herramientas basadas en software libre como: 1) OpenMosix, que permite la construcción de un clúster para paralelizar la carga de trabajo en el proceso de renderizado y 2) Blender para diseñar modelos 3D; la compatibilidad de las dos herramientas se basa en ser bajo licencia GPL (Olivares Pinto *et al.*, 2014).

Searching for Extraterrestrial Intelligence: SETI Past, Present, and Future

En 1995 David Gedye se propuso convertir radio Seti en un súper computador virtual compuesto de un gran número de máquinas conectadas a través de internet y organizó el proyecto Seti@home para explorar esta idea. Fue lanzado originariamente en mayo de 1999.

El proyecto Seti@home es un proyecto diseñado para la búsqueda de inteligencia extraterrestre. Recientemente se han venido creando versiones de Seti que se ejecutan en las unidades de procesamiento gráfico (GPU) y son capaces de operaciones altamente paralelas. Seti@home puede calcular en la GPU hasta 30 veces más rápido que la CPU en los sistemas que contienen una GPU compatible (Seti@home, 1999).

Out of core sort-first parallel rendering for cluster based tiled displays

Este proyecto permite renderizar imágenes de alta resolución de modelos extensos usando PCs con poca memoria. Para ello se usó un algoritmo de preprocesamiento distribuido para generar una representación jerárquica en disco del modelo. En tiempo de ejecución cada PC renderiza la imagen para un cuadro de visualización, usando un

enfoque de renderización distribuida que emplea múltiples hilos para sobreponer los tiempos de renderizado, visualización y operaciones en disco. El sistema puede operar en un modo aproximativo para tiempo real de renderizado, o en un modo conservador para renderizar con precisión garantizada. En modo aproximativo con 16 PCs, cada uno con 512 Mb de RAM, se puede renderizar imágenes de 12 megapixeles de un modelo de 13 millones de triángulos con un 99.3% de precisión a 10.8 frames por segundo. Esto demuestra que un clúster hecho con PCs de bajo costo es una alternativa para sistemas que requieren alto procesamiento (Correa, Klosowski y Silva, 2002).

Free DistributedRenderFarmfor Blender

RenderFarm es un grupo de ordenadores conectados entre sí para completar una tarea grande. En el caso de la renderización en 3D, la mayoría de las veces la granja de renderización dividirá los fotogramas de una animación en varios equipos, y en lugar de tener un solo equipo trabajando durante 100 días se puede tener 100 equipos que trabajarán durante un día en la renderización.

Cuando el dueño del proyecto añade su escena para trabajos que hacer, el servicio se divide la animación en fotogramas individuales para renderizar. Se envía cada trama a un ordenador conectado y tiene como objetivo optimizar esta elección basado en la memoria disponible, así como la potencia de la CPU/GPU (Sheep it!, s.f.).

The New 'Cloud' Era

Con 2.000 poderosos nodos para renderizar, Fox Renderfarm tiene como objetivo proporcionar la renderización a través de la nube con toda la industria del CG.

Mediante el uso de la tecnología en la computación en la nube, Fox Renderfarm es capaz de satisfacer las demandas de los clientes y ayudarles a obtener trabajos de calidad antes de la fecha límite.

Aunque la tecnología en la nube se desarrolla y cambia rápidamente, Fox Renderfarm mantiene el ritmo de la era de la nube y ofrece un mejor y más rápido servicio de renderización a todos sus clientes.

Actualmente, Fox Renderfarm despliega miles de estaciones de trabajo y tiene pensado crear más en el futuro.

Cada máquina cuenta con un procesador Intel Xeon E55XX series, 24-64GB de RAM, 10Gbps de red local. En cuanto a conexión de red, cuenta con acceso a internet de 200Mbps y Hardware Firewall (Iori y Gordon, s.f.).

Enfoque técnico

Para lograr el objetivo final, es decir, el renderizado de imágenes o animaciones, se planteó un diseño basado en arquitectura clúster corriendo en la plataforma AWS de Amazon y su desarrollo se basó en componentes, presentándolo como un servicio web para ser consumido por diseñadores individuales o empresas de diseño de cualquier tamaño.

A continuación, será presentada la propuesta arquitectónica, donde se muestran y detallan los componentes (y sus relaciones), que conforman el servicio de renderizado.

Arquitectura propuesta

La arquitectura del clúster que se utilizó fue adaptable, es decir, varió en el número de nodos que se usaron para las diferentes pruebas, utilizándose un servidor maestro, dos nodos, cuatro nodos y 10 nodos según la prueba a realizar. Esta arquitectura fue montada sobre la nube provista por Amazon utilizando la plataforma AWS. Esta plataforma permite una gran escalabilidad y flexibilidad en su uso para ampliar o reducir el número de servidores y nodos.

La figura 1 permite ver de forma amplia como está compuesta la plataforma, la cual tiene un servidor maestro y n cantidad de nodos de procesamiento.

Figura 1. Modelo general de la plataforma.
Fuente: Elaboración propia.

El servidor maestro se encarga de proveer el servicio de renderizado, lo que permite a cualquier cliente usar la plataforma consumiendo el servicio al enviar trabajos a realizar (archivos a renderizar). Este servidor es el encargado de delegar, coordinar y distribuir los diferentes procesos entre los nodos que estén en uso.

Los nodos de procesamiento se encargan de realizar las tareas de cálculos físicos. La plataforma propuesta permite que el número de nodos sea variable, incluso en medio de un procesamiento, ya que la plataforma tiene una alta tolerancia a fallos.

Tanto el servidor maestro como los nodos requieren de piezas de software (componentes) para su funcionamiento. La figura 2 presenta los componentes instalados en el servidor maestro.

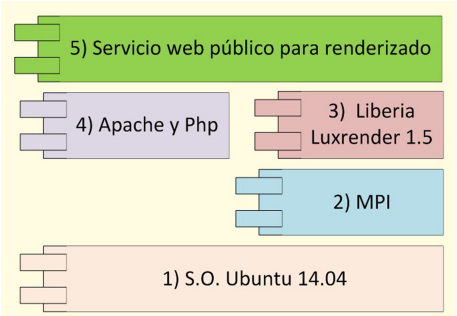


Figura 2. Modelo de capas del servidor maestro.
Fuente: Elaboración propia.

1. S.O.: el servidor maestro tiene como sistema operativo a Ubuntu server en su versión 14.04.
2. MPI: este componente es en realidad un conjunto de librerías basadas en un estándar que permiten el paso de mensajes para realizar tareas de forma distribuida. Sus principales características son: no precisan memoria compartida y permite la sincronización entre procesos.
3. Luxrender 1.5: es un motor de renderizado de uso libre bajo licencia APL2. Seleccionado por su capacidad de distribución de trabajo en nodos. Además, Luxrender recibe una gran variedad de tipos de archivo o fuentes para procesar, entre los que se encuentran: Blender, Maya, Daz Studio, 3DS Max y SKetchUp generalmente exportados a XLS.
4. Adicional a lo anterior, Luxrender tiene una serie de librerías en C++ que permite ser incrustado en aplicaciones de terceros y cuenta con versiones de Linux, Windows y Mac.
5. Apache y Php: esta combinación es una de las más comunes en internet y, para el presente trabajo, se utilizó como plataforma sobre la cual se ejecuta el servicio/aplicación web.
6. Servicio web: finalmente, se construyó un componente que permite dotar a la plataforma de funcionalidad.

Y la figura 3 presenta los componentes alojados en los nodos

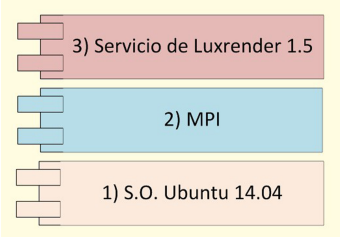


Figura 3. Piezas se software en cada nodo.
Fuente: Elaboración propia.

Las capas 1, 2 y 3 en todos los nodos son las mismas que el servidor maestro, con la única diferencia que en cada nodo se inicia Luxrender como un servicio para que pueda recibir las tareas a procesar.

La figura 4 muestra una captura de pantalla del aplicativo desarrollado.

Flujo de procesos

El flujo de procesos es de vital importancia tanto para asegurar el funcionamiento de la plataforma ofreciendo el servicio, como para entenderla. La figura 5 muestra un diagrama con los pasos más importantes que realiza la plataforma.

Como se muestra en el diagrama de flujo, el cliente sube un archivo a renderizar en un comprimido (.zip), dicho archivo es descomprimido por el aplicativo y entregado a Luxrender. Una vez en el servidor maestro tiene un nuevo trabajo que realizar en un archivo XLS con las instrucciones y características de la imagen o animación sintética, Luxrender distribuye el procesamiento entre todos los nodos de proceso disponibles y queda a la espera las constantes actualizaciones generadas por cada uno de ellos.

Aunque el diagrama presenta un modelo ideal, en el cual cada proceso es terminado, la realidad es que el proceso de renderizado puede terminar siendo un ciclo infinito, pues los cálculos de luces, sombras, reflejos, etc. se ven afectados por el rebote de sí mismos, por tal razón el presente caso de estudio se planteó con un límite de tiempo definido, que será explicado en detalle en la sección de pruebas y resultados.

El web service utiliza un clúster que permite dividir las tareas de renderizado y así obtener un procesamiento más rápido de la imagen o animación.

Figura 4. Aplicativo que permite el uso de la plataforma.
Fuente: Elaboración propia.

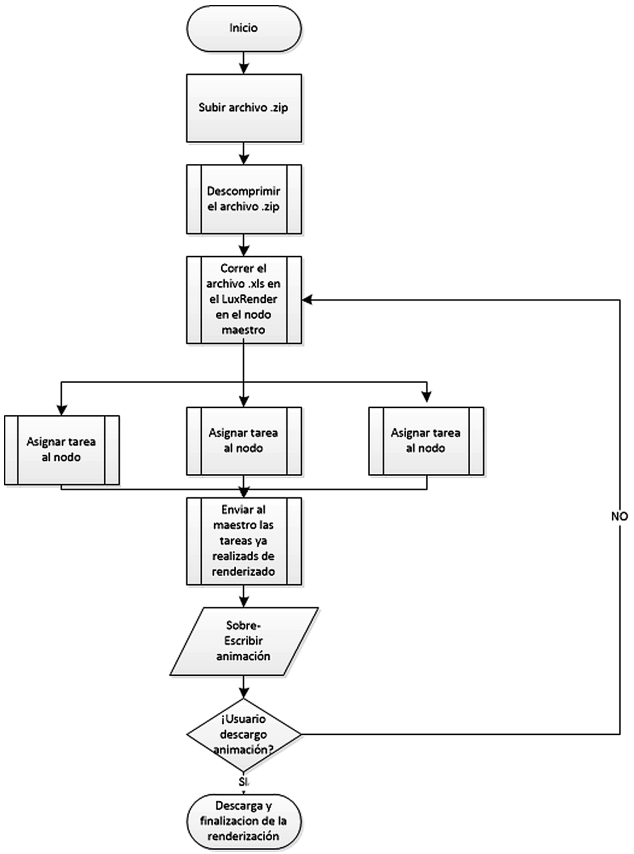


Figura 5. Diagrama de flujo de la plataforma.
Fuente: Elaboración propia.

Pruebas y resultado

En este capítulo se muestran los diferentes resultados de las pruebas que se realizaron durante el proyecto por medio de matrices y gráficos de resultados, con los que se puede inferir tanto los tiempos óptimos de renderizado, así como el número de nodos recomendado y las características de estos.

Descripción de la prueba

Las pruebas se hicieron con una imagen sintética diseñada en la herramienta Blender. A continuación, se presenta la ficha técnica de la imagen caso de prueba:

Tabla 1. Ficha técnica del archivo a renderizar.

Resolución de la imagen de salida	800 x 600 pixeles
Autor	Pitta
	http://www.luxrender.net/wiki/User:Piita
Link de descarga del archivo	https://www.cg.tuwien.ac.at/~zsolnai/luxrender-scenes/
Nombre del archivo	Conference_room.7z

Fuente: Elaboración propia.

Las pruebas consistieron en 24 escenarios, construidos con seis diferentes configuraciones de recursos físicos (plataformas). En cada una de las plataformas se realizaron pruebas con diferente

número de procesos concurrentes definidos en cuatro grupos de 32, 128, 512 y 1024 hilos de procesamiento en paralelo:

Es importante mencionar que, durante la etapa de preparación del entorno, se evidenció que el proceso requerido podría ser indefinido debido a los cálculos cíclicos que se realizan. Por tal motivo, se estableció un tiempo límite fijo de dos horas y 15 minutos, para luego comparar los resultados obtenidos en las diferentes pruebas.

Durante la realización de las pruebas se definió capturar la imagen de resultado cada 15 minutos (en total nueve muestreos por cada prueba), a partir del inicio del procesamiento.

A continuación, se muestra la composición técnica de las plataformas

A partir de la plataforma 2, todas las plataformas usaron un servidor maestro con las siguientes características:

- S.O.: Ubuntu server 14.04.
- Ram: 60.2 GB.
- Números de CPU: 32.

También es importante mencionar que el clúster de nodos de procesamiento de cada plataforma usa un modelo homogéneo, donde se cuenta con las mismas configuraciones y recursos para cada nodo.

Toma de resultados

Debido a que la simple inspección visual es subjetiva, como valores para el análisis comparativo se consideraron el tamaño de la imagen y el ruido. Para esto, se capturó el tamaño de la imagen en cada muestreo y su histograma. Con esto se pudo evidenciar que a medida que aumentaba la calidad, el ruido se reducía y con ello la disminución del tamaño de archivo generado.

Como evidencia de lo anterior, se presentan las figuras 6 y 7, donde se muestran dos imágenes con diferente resultado respecto a la calidad o nitidez de la imagen generada. En esta comparación se evidencia la disminución del ruido que se puede verificar por el histograma de cada imagen, notándose en ellos que a mejor calidad en

Tabla 2. Datos técnicos de las seis plataformas utilizadas.

	1) Plataforma 1 (E) *	2) Plataforma 2 (P2)	3) Plataforma 3 (B2)	4) Plataforma 4(B4)	5) Plataforma 5(B10)	6) Plataforma 6(G2)
Número de nodos de proceso (NP)		2	2	4	9	2
Nodo maestro (NM)		1	1	1	1	1
Sistema operativo	Windows 7 x64	Ubuntu server 14.04	Ubuntu server 14.04	Ubuntu server 14.04	Ubuntu server 14.04	Ubuntu server 14.04
RAM	3.90 GB	63.9 GB	181.5 GB	302.5 GB	605 GB	180.5 GB
Número de CPU	1	34	32	160	320	96
Instancia aws:	0	m1.small NP cc2.8xlarge NM	m1.small NP cc2.8xlarge NM	m1.small NP cc2.8xlarge NM	m1.small NP cc2.8xlarge NM	g2.8xlarge NP cc2.8xlargeNM
Número de GPU	0	0	0	0	0	8

* un equipo portátil (usado como referencia del peor resultado esperado)

Fuente: Elaboración propia.

las imágenes sintéticas (menos ruido) se tendrán picos más pronunciados en el histograma. Esto es debido a que los colores convergen en grupos concretos del histograma, mediante la representación gráfica del número de píxeles en cada nivel de intensidad de color.

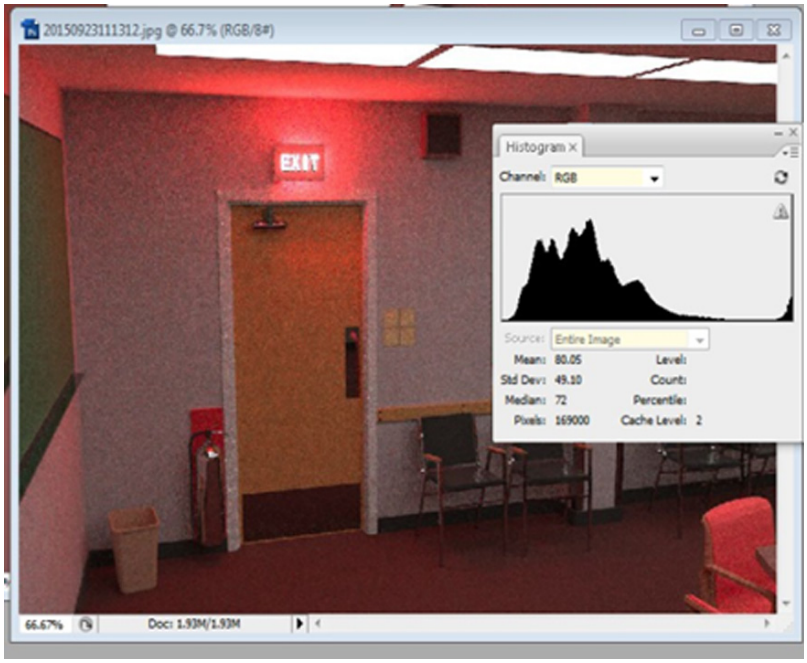


Figura 6. Imagen de regular calidad. Tamaño: 1.25 MB.
Fuente: Elaboración propia.

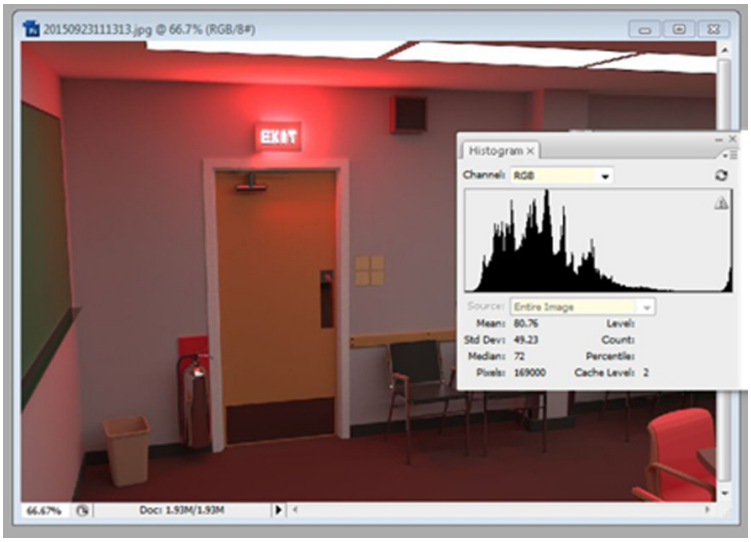


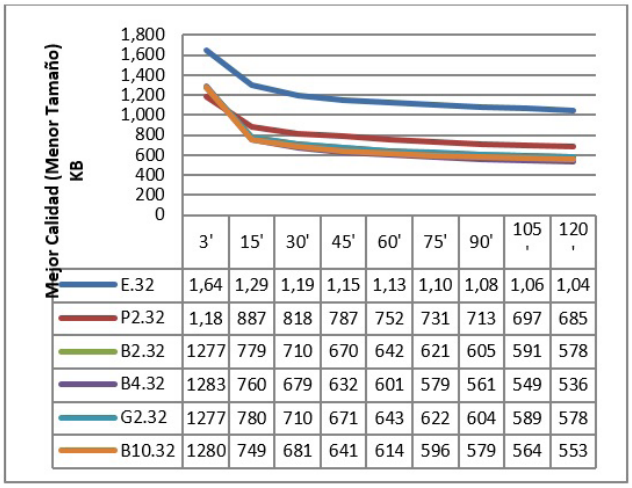
Figura 7. Imagen de buena calidad. Tamaño: 554 KB.
Fuente: Elaboración propia.

Resultados

Se obtuvieron un total de 216 imágenes renderizadas, ya que en cada uno de los 24 escenarios propuestos se capturaron nueve imágenes.

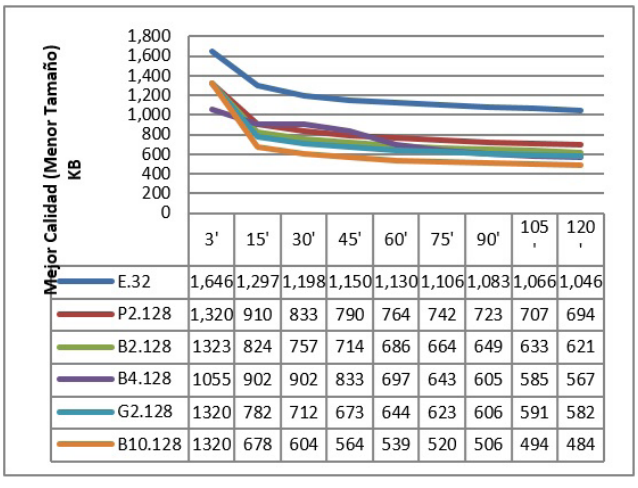
Para facilitar la interpretación de los resultados obtenidos, se optó por resumir estos resultados en cuatro gráficos de líneas, uno por cada número de hilos utilizado, donde se comparan los tamaños de las nueve imágenes capturadas en cada una de las plataformas.

A continuación, se presentan los gráficos de líneas obtenidos y una observación sobre estos:



	3'	15'	30'	45'	60'	75'	90'	105'	120'
E.32	1,64	1,29	1,19	1,15	1,13	1,10	1,08	1,06	1,04
P2.32	1,18	887	818	787	752	731	713	697	685
B2.32	1277	779	710	670	642	621	605	591	578
B4.32	1283	760	679	632	601	579	561	549	536
G2.32	1277	780	710	671	643	622	604	589	578
B10.32	1280	749	681	641	614	596	579	564	553

Figura 8. Resultados para cada plataforma con con 32 procesos concurrentes (hilos).

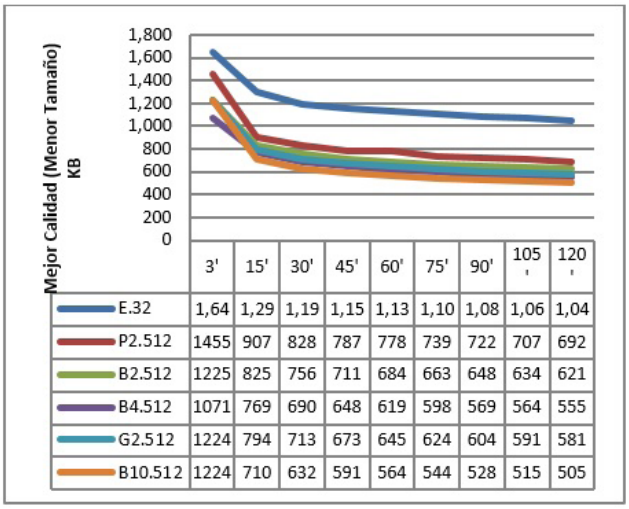


	3'	15'	30'	45'	60'	75'	90'	105'	120'
E.32	1,646	1,297	1,198	1,150	1,130	1,106	1,083	1,066	1,046
P2.128	1,320	910	833	790	764	742	723	707	694
B2.128	1323	824	757	714	686	664	649	633	621
B4.128	1055	902	902	833	697	643	605	585	567
G2.128	1320	782	712	673	644	623	606	591	582
B10.128	1320	678	604	564	539	520	506	494	484

Figura 9. Resultados para cada plataforma 128 procesos concurrentes (hilos).

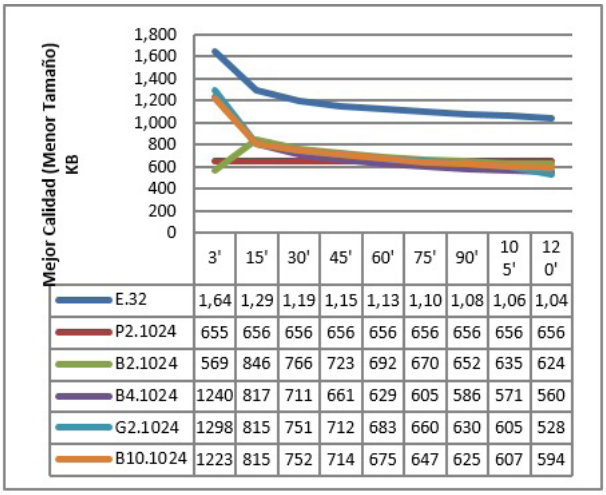
La figura 8 muestra una pronta caída en el tamaño de la imagen en los primeros 15 minutos y un mejor resultado para la plataforma B4.

En la figura 9 se evidencia una superioridad por parte de la plataforma con más nodos paralelos, es decir, la Plataforma B10.



	3'	15'	30'	45'	60'	75'	90'	105'	120'
E.32	1,64	1,29	1,19	1,15	1,13	1,10	1,08	1,06	1,04
P2.512	1455	907	828	787	778	739	722	707	692
B2.512	1225	825	756	711	684	663	648	634	621
B4.512	1071	769	690	648	619	598	569	564	555
G2.512	1224	794	713	673	645	624	604	591	581
B10.512	1224	710	632	591	564	544	528	515	505

Figura 10. Resultados para cada plataforma con plataforma con 512 procesos concurrentes (hilos).



	3'	15'	30'	45'	60'	75'	90'	105'	120'
E.32	1,64	1,29	1,19	1,15	1,13	1,10	1,08	1,06	1,04
P2.1024	655	656	656	656	656	656	656	656	656
B2.1024	569	846	766	723	692	670	652	635	624
B4.1024	1240	817	711	661	629	605	586	571	560
G2.1024	1298	815	751	712	683	660	630	605	528
B10.1024	1223	815	752	714	675	647	625	607	594

Figura 11. Resultados para cada 1024 procesos concurrentes (hilos).

En la figura 10 se evidencia también una superioridad por parte de la plataforma con más nodos paralelos, la Plataforma B10, pero con imágenes 2.0% más pesadas.

La figura 11 muestra una alteración en los datos registrados, específicamente en la plataforma P2 en toda su muestra y en B2 en el primer muestreo. En ambos casos se generó un error en el resultado de la imagen debido a que la cantidad de resultados entregados por cada nodo era superior a lo que podía procesar la plataforma, generando imágenes mal formadas como la mostrada en la figura 12.

Considerando lo anterior, y ya tabulados todos los resultados, se seleccionaron los mejores resultados de cada plataforma. A continuación, se presenta la tabla con los mejores resultados.

Como evidencia la tabla 2, los mejores resultados se obtuvieron con una cantidad baja de hilos, esto es debido a que a mayor cantidad de hilos se aumenta la cola de procesos para cada CPS. Sin embargo, para el caso de usar GPU se obtienen mejores resultados ya que este tipo de plataforma está diseñada para un gran número de procesos simultáneos.

A continuación, se muestra el resumen compilando de los mejores resultados para cada número de procesos concurrentes y su análisis.

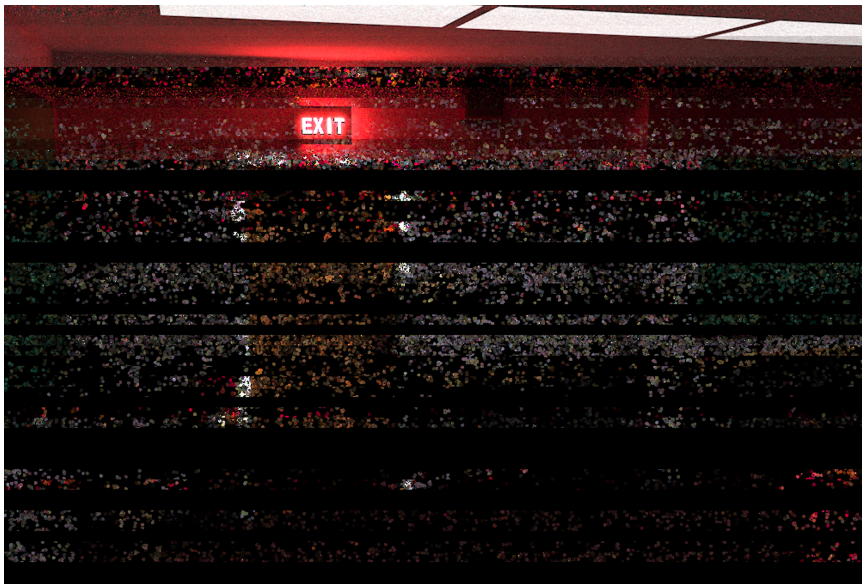
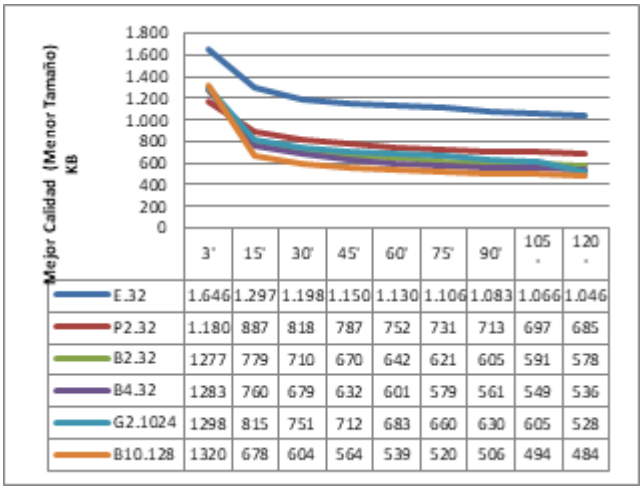


Figura 12. Resultado cuando la plataforma no soporta el número de hilos.
Fuente: Elaboración: Elaboración.

Tabla 3. Mejores resultados por plataforma luego de dos horas de procesamiento continuo.

	ID	Número de hilos	Tamaño
Plataforma 1	E.32	32	1046 KB
Plataforma 2	P2.32	1024	686 KB
Plataforma 3	B2.32	32	578 KB
Plataforma 4	B4.32	32	536 KB
Plataforma 5	B10.128	128	484 KB
Plataforma 6	G2.1024	1024	528 KB

Fuente: Elaboración propia.



	3'	15'	30'	45'	60'	75'	90'	105 '	120 '
E.32	1.646	1.297	1.198	1.150	1.130	1.106	1.083	1.066	1.046
P2.32	1.180	887	818	787	752	731	713	697	685
B2.32	1277	779	710	670	642	621	605	591	578
B4.32	1283	760	679	632	601	579	561	549	536
G2.1024	1298	815	751	712	683	660	630	605	528
B10.128	1320	678	604	564	539	520	506	494	484

Figura 13. Mejores resultados de cada plataforma.
Fuente: Elaboración propia.

El mejor resultado se obtuvo con una plataforma con 10 nodos, 605 Gb de RAM y 320 CPUs.

El mejor resultado se obtuvo con una plataforma con 10 nodos, 605 Gb de RAM y 320 CPUs.

Conclusiones

Varios equipos pueden disminuir los tiempos al realizar una tarea o proceso que requería una cantidad significativa de cálculos y recursos computacionales.

Es posible determinar el nivel de ruido de una imagen sintética y, por tanto, la calidad en el resultado de un proceso de renderización, mediante el análisis de sus histogramas. A mejor calidad, los histogramas tendrán picos más pronunciados, pues los colores convergen en grupos concretos.

Para la renderización de imágenes pequeñas que no requieren un mayor nivel de procesamiento se puede utilizar un clúster pequeño con las instancias gratuitas que ofrece Amazon AWS y con 32 hilos de procesamiento.

Utilizar dos instancias con GPU con 180.2 GB de RAM y 1024 hilos de procesamiento solo disminuye el nivel de calidad de la imagen en un 9%, comparado con utilizar 10 instancias con 602 GB de RAM y 128 hilos de procesamiento.

De igual forma, cuatro instancias con 301 GB de RAM y 32 hilos de procesamiento solo disminuyen la calidad de la imagen en un 10% en comparación con 10 instancias de 602 GB de RAM y 128 hilos de procesamiento. Por tanto, esta se convierte en la opción más adecuada en cuanto a la relación calidad-precio.

Referencias

Sheep it! (s.f.). *Sheep it! Render farm*. Recuperado de: https://www.sheepit-renderfarm.com/index.php

Seti@home. (05 de 1999). *Acerca de SETI@home*. Recuperado de: http://setiathome.berkeley.edu/sah_about.php

Iori y Gordon. (s.f.). *The New 'Cloud' Era*. Recuperado de: http://www.foxrenderfarm.com/

Olivares Pinto, U., Medina, F., Vega Lebrún, C. A., Cendejas Valdez, J. L. y Rosano Ortega, G. (01 de 2014). *Diseño de clústeres computarizados con algoritmos de renderizado para la construcción de modelos tridimensionales a través del uso de herramientas open source*. Recuperado de: http://www.redalyc.org/articulo.oa?id=90430816012

T. Correa, W., T. Klosowski, J. y T. Silva, C. (2002). *Out Of Core Sort First Parallel Rendering for Cluster Based Tiled Displays*. Recuperado de: http://www.sci.utah.edu/~csilva/papers/egpgv02.pdf