Propuesta para el aprovechamiento de los recursos computacionales en la sala Alan Turing

Gregorio Ospina Arango

Proyecto de grado presentado como postulación para el titulo de
Ingeniero de Sistemas y Computación

Departmento de Ingenieria de Sistemas y Computación
Universidad de los Andes
Colombia
Junio 2020

Contents

1 Contexto

1.1 Introducción

El Departamento de Ingeniería de Sistemas y Computación de la Universidad de los Andes cuenta con unas salas de cómputo dedicadas para los estudiantes de pregrado y posgrado del departamento que tienen de un gran poder de cómputo. Estas salas tienen en mente soportar los procesos más exigentes que puedan requerir las distintas asignaturas del departamento. Entre estas capacidades de cómputo, en la sala Alan Turing, cada máquina está equipada con una tarjeta gráfica Nvidia Quadro P600 dedicada.
Sin embargo, se ha evidenciado que los recursos de estas salas no se aprovechan completamente, y la mayoría del tiempo están siendo subutilizados. Este proyecto busca estudiar la posibilidad de incrementar el aprovechamiento que se le hacen a los recursos de computación de la universidad, haciendo un enfoque especial en las tarjetas gráficas de la sala Alan Turing.

1.2 UnaCloud

El aprovechamiento de los recursos computacionales de la universidad ha sido un tema de investigación en el pasado, y existen varios proyectos que han hecho considerable progreso para resolver este problema. Uno de esos proyectos que se implementó sobre las salas de cómputo de la universidad y que sigue siendo utilizado hasta el día de hoy es UnaCloud.
UnaCloud es un programa de computación oportunista sobre la nube. Es una implementación del modelo de IaaS (Infrastructure as a Service [Infraestructura como servicio]) que detecta sobre cada una de las máquinas de las salas, los recursos ociosos, y de manera dinámica va consumiendo poder de cómputo a través de una máquina virtual instalada en el computador.(1)
De esta manera, se pueden paralelizar procesos de cómputo para ser ejecutados en este gran clúster de computadores, volviéndose así una excelente herramienta para investigadores, para el departamento o para participar en iniciativas académicas de computación voluntaria. Este proyecto ha tenido un enorme éxito, sin embargo no ha sido capaz de aprovechar completamente todos los recursos, puesto que no se puede usar los recursos de cómputo de las tarjetas gráficas. Las razones por las cuales esto no se ha podido, serán discutidas a lo largo de este documento.

1.3 Tarjetas Gráficas

Las tarjetas gráficas son unidades de cómputo especializadas en la computación paralela. Mientras que un CPU tiene poca cantidad de cores, con mucho poder de procesamiento. Las tarjetas gráficas tienen una gran cantidad de cores de poder computacional bajo, haciéndolas perfectas para el procesamiento de grandes cantidades de información cuando estos procesos pueden ser divididos en subtareas independientes. En computación *throughput* es la cantidad de pro-

cesos siendo computados al mismo tiempo. Estos procesadores gráficos gozan una enorme cantidad de throughput debido a la cantidad de procesos que pueden correr en paralelo. Esto hace de estos procesadores una herramienta muy especializada para áreas específicas como el desarrollo de juegos, el procesamiento de Big Data (2), y el estudio de inteligencia artificial(3).

1.4 Utilización actual de recursos

La afirmación de que los recursos de las salas de computación están siendo subutilizados es corroborada por una serie de estudios que fueron realizados en el 2010 y 2018. Estos estudios buscaban entender el porcentaje de utilización del CPU de los computadores de las salas Waira durante las horas del día.

1.4.1 Estudios previos de consumo

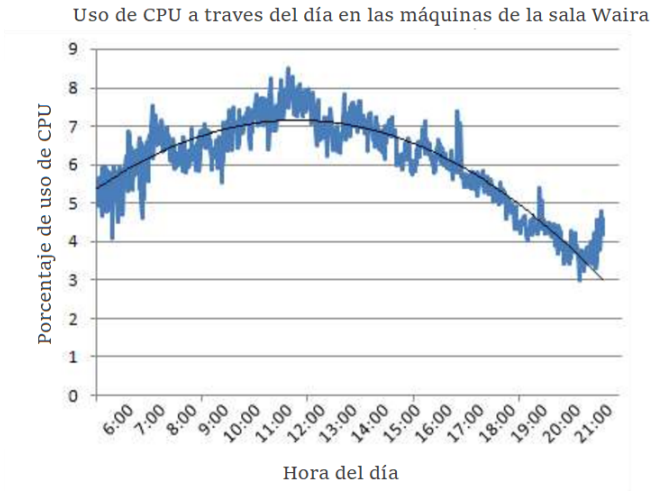

Figure 1: Estudio del 2010

En estas gráficas se evidencia que el consumo por parte de los estudiantes en términos de CPU es muy bajo, con picos de consumo de 6 8% al medio dia. Estos patrones resaltan la subutilización que se le está dando a estos recursos.

En un paper publicado en el 2015 para la HPCLATAM-CLCAR Latin America Joint Conference que habla sobre los resultados de UnaCloud para el aprovechamiento de los recursos ociosos, este presenta unas gráficas de un estudio de consumo de RAM que tienen las máquinas de las salas Waira, muy similar al estudio presentado anteriormente.(8)

Este estudio vio un pico de consumo de RAM de 30% Lo cual es considerablemente bajo y demuestra la necesidad que hubo de desarrollar el proyecto de

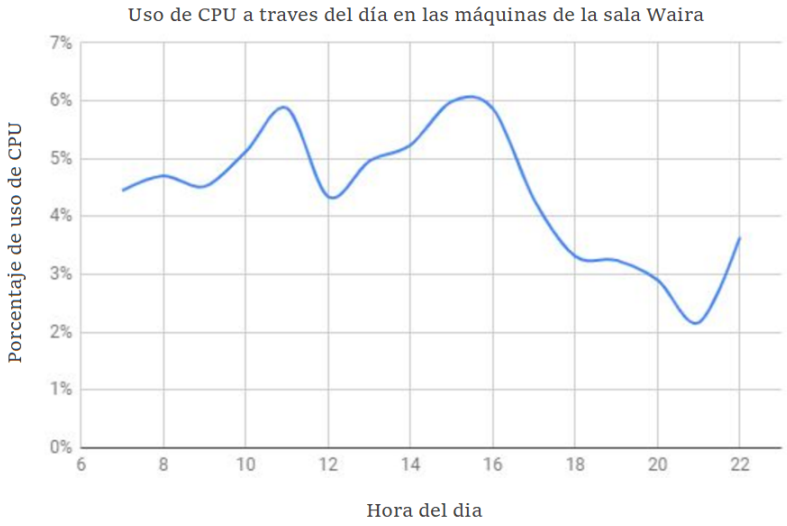

Figure 2: Estudio del 2018

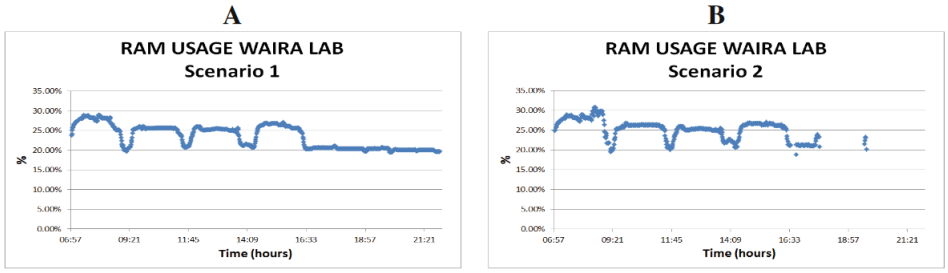

Figure 3: Estudio consumo RAM salas Waira [Osthoff, Barrios, Olivier, Silva, 2015]

UnaCloud.

1.4.2 Consumo en tarjetas gráficas

Para poder tomar decisiones sobre el aprovechamiento de los recursos computacionales, fue prioritario entender cuál es el consumo que se le esté dando a las tarjetas gráficas en las clases de la universidad. Obtuvimos información sobre dos cursos, Videojuegos producto cultural y Desarrollo de aplicaciones de realidad mixta. El profesor encargado de estos cursos, Pablo Figueroa, nos explicó que los usos que le dan a las tarjetas son:

- Unity: desarrollan videojuegos y demos en realidad virtual y aumentada
- Unreal Engine: desarrollan videojuegos y demos en realidad virtual y aumentada
- WebGL y OpenGL: desarrollan ejercicios del curso de computación gráfica interactiva
- Optix: desarrollan ejercicios del curso de computación gráfica interactiva
- CUDA: desarrollan ejercicios del curso de computación gráfica interactiva

Estas son herramientas que requieren mucho procesamiento gráfico, lo que nos hace pensar que, cuando se están utilizando las tarjetas en estas clases se les pide el total de su poder de cómputo. Sin embargo, si hacemos la suposición de que los usuarios generales de la sala en general utilizan más el CPU que la GPU, entonces podríamos concluir que estos instantes donde se le pide a la tarjeta gráfica el total de su poder de cómputo son anomalías. Para entender mejor los patrones de consumo y las necesidades que tiene la sala de estas tarjetas, hubiera sido oportuno tener un mejor entendimiento de la cantidad total de curso que hay que utilizan de estas tarjetas, para asi poder corroborar de manera mas definitiva la hipótesis de su sub-utilización.

2 Descripción y objetivos

Este proyecto se podrá concluir exitosamente mientras la propuesta que se derive de este cumpla con los siguientes objetivos.

2.1 Aprovechamiento de recursos

Como prioridad principal del proyecto, está el aprovechamiento de los recursos. El recurso que va a ser el enfoque principal del proyecto va a ser el de las tarjetas gráficas. Lo que se quiere poder asegurar es la posibilidad de estar utilizando las tarjetas gráficas mientras que la tarjeta no este siendo utilizada en su completitud por el estudiante. Con el propósito de poder poner la tarjeta a realizar procesos de cómputo similares a lo que hace UnaCloud.

2.2 Restricciones

2.2.1 Experiencia de usuario

La solución que se proponga deberá mantener el nivel de experiencia de usuario actual. No se puede aceptar que una solución para aprovechar mejor las máquinas destinadas a estudiantes dañe la experiencia de usuario que ya tienen actualmente estos estudiantes. Con experiencia usual nos referimos a que no queremos cambiar el sistema operativo, la manera de acceder a la máquina, el acceso a los archivos y el desempeño de la máquina.

2.2.2 Niveles de cómputo

La solución que se proponga debe permitir que cada computador tenga mínimo la misma capacidad de cómputo que tienen actualmente. Esto con el propósito de no interrumpir las clases que dependen del buen desempeño de estas máquinas.

3 Investigación

3.1 Arquitectura Inicial

La sala Turing cuenta actualmente con treinta y nueve máquinas que corren sobre el sistema operativo Windows 10, que tienen instalado el hipervisor de nivel dos Virtualbox. Estas máquinas cuentan con la siguiente arquitectura:

- Procesador: Intel (R) Core (TM) i7-7700 CPU @4.20GHz [8 core(s) x64]
- Disco duro: 1 TB
- Memoria: 64 GB DDR4 @1.20 GHz
- GPU: Nvidia Quadro P600
- Tarjeta de Red: Intel(R) (2) I219-LM Gigabit Network Connection 1 Gb/s
- Sistema Operativo: Windows 10 Enterprise.

3.2 Alternativas

A lo largo de la investigación, hubo varias tecnologías y herramientas que consideramos podían ser útiles para hacer la distribución de los recursos, a continuación vamos a hacer una descripción de las tecnologías y las razones por las cuales fueron descartadas.

3.2.1 Passthrough

La primera idea que tuvimos fue la de hacer un Passthrough. Passthrough permite al usuario usar un dispositivo PCI (Tarjetas gráficas o tarjetas de red) a través de un ambiente virtual para ser accedido por una máquina virtual (4). Esta es una solución que es popular en usuarios con ambientes Linux para evitar tener que hacer el inicio doble (double-boot) para acceder a programas Windows con la tarjeta gráfica. Se lanza una máquina virtual desde la máquina Linux y hacen el passthrough de la tarjeta gráfica a Windows.
Sin embargo, debido a que las máquinas de la sala Turing corren Windows en máquina anfitrión, esto no era una posibilidad inmediata. Previamente había habido intentos de hacer esto en las máquinas de la sala cuando estas corrían Windows Vista. Pero se concluyó que esto no era posible por la naturaleza del sistema operativo.
No obstante, estas máquinas corren ahora Windows 10 Enterprise, el cual es

una versión que ha tenido muchos cambios, y la dirección que está poniendo la compañía padre, Microsoft al Software Open Source (5) abría la posibilidad a que hubiese mayores libertades en el sistema operativo.
Para entender las razones por las cuales existen estas diferencias entre el sistema operativo Windows y Linux, es importante entender bien que compone a estos sistemas operativos, y cuáles fueron las decisiones de diseño que fueron tomadas, con sus implicaciones sobre nuestro estudio.

Kernel El kernel de un sistema operativo es el programa que se encuentra en todo el centro del sistema y tiene todo el control sobre todo el sistema. Siempre está cargado en memoria y sirve como la autoridad a la hora de: hacer repartición de la memoria RAM, manejar y administrar los dispositivos I/O, el manejo de recursos como espacio de direcciones en memoria, y la comunicación intraprocesal, conocida como IPC2.

Microkernels y los Kernel monolíticos. A lo largo de la historia, dos diseños de kernels han estado en el centro del desarrollo y debate sobre el mejor diseño para el sistema operativo, estos dos diseños son el kernel monolítico y el microkernel. El kernel de tipo monolítico fue el diseño inicial del desarrollo de sistemas operativos. Como lo sostiene su nombre es un programa el cual corre en un mismo espacio, es decir está programado enteramente desacoplado con todo el código necesario para poder ejecutar todas las tareas relacionadas con el kernel y el manejo del sistema operativo.
Este tiene varias ventajas: primero suele ser más pequeño en código, y esto indica que es más sencillo hacer la detección y corrección de errores. Al ser solo un programa, la velocidad de comunicación es máxima. Sin embargo, cuenta con algunas desventajas: Las incrementales exigencias de funcionalidades y servicios a los sistemas operativos hacen que los kernels monolíticos crezcan de manera desorganizada y pierdan esa mantenibilidad que los hizo atractivos en un comienzo. Por otro lado, ya que cada parte del kernel tiene autoridad máxima sobre el sistema, es posible que una parte del kernel corrompa los datos necesitados por otro proceso.

Para sobrepasar estas restricciones, se desarrollaron dos tipos de arquitectura de kernel, que han sido los primordiales diseños hasta la fecha. Microkernels fue la solución a muchos de los problemas de los kernels monolíticos, estos están constituidos bajo la idea que el bloque monolítico del kernel podía ser dividido por funciones en una unidad que se llamó "servidores", los cuales se comunican a través de un kernel de mínimo tamaño, así dejando en memoria la menor cantidad de información del sistema, consiguiendo que la memoria disponible para el usuario incremente considerablemente. Solamente las funcionalidades más básicas y primordiales para la ejecución del sistema son dejadas en memoria; manejo de memoria, multitasking e IPC. Cualquier otro servicio como manejo de red son delgados por servidores que residen en espacio de memoria del usuario y únicamente son accedidos cuando algún programa o proceso lo requiera. Este

diseño tiene como ventajas que sea mucho más fácil de mantener, puesto que es mucho más fácil encontrar fallas o bugs en un sistema si se tiene desacoplados los servicios y sus dependencias. Parches al kernel son mucho más fáciles de aplicar, puesto que no se tiene que hacer un reboot completo del sistema, sino simplemente de los servicios a los cuales se les quiere brindar atención. Desafortunadamente, este diseño no viene sin consecuencias, puesto que la comunicación entre estos servicios requiere interfaces de comunicación, este diseño potencialmente puede ser susceptible a fallas en desempeño.

Comparación del Kernel Windows y Linux Desde la aparición de Windows NT en 1993, Microsoft implementó un diseño derivado del microkernel, el kernel híbrido. Este parte del modelo de microkernel, pero agrega espacio libre en el espacio reservado para el kernel, donde coloca servicios que necesiten procesos, este espacio se va turnando por los distintos procesos y así, se logra una más rápida comunicación a diferencia del microkernel donde la comunicación siempre se hace con el espacio de memoria del usuario. De manera opuesta, Unix, el sistema operativo del cual se deriva Linux consiste en un kernel monolítico con la posibilidad de agregarle módulos.

La diferencia talvez más dramática que hay entre los dos kernels es la posibilidad que hay en el kernel Linux de modificarlo. El kernel Linux es un proyecto open source, lo que significa que cualquier persona puede descargar el código, mirar cómo funciona por adentro, hacerle cambios, compilarlo e instalarlo en su máquina, esto ha llevado a que la comunidad de Linux tenga mucha libertad en cuanto a el funcionamiento de su máquina. A diferencia de esto, Microsoft tiene un estricto control sobre el kernel de Windows, y este es inmodificable. Esto significa que Microsoft es el único actor con los permisos de hacer algún cambio al contenido del kernel.

Conclusiones relevantes al proyecto Después de estudiar esta posibilidad y ver las diferencias que tienen los dos sistemas operativos, llegamos a la conclusión que para Windows 10, la posibilidad de hacer Passthrough para virtualizar el uso de la tarjeta gráfica no es posible por las siguientes razones:

- **Immutabilidad del IOMMU**
  El IOMMU (Input Output Memory Management Unit) es un componente que se encarga de traducir direcciones provenientes de driver externos al sistema, a direcciones de memoria real en el sistema. Esto significa que cualquier dispositivo que quiera usar comunicación nativa con el computador debe pasar atreves del IOMMU, entre estas situaciones deseables esta una máquina virtual queriendo utilizar las capacidades de aceleración 3D de una GPU debe mandar sus instrucciones a través del IOMMU. Sin embargo, muchos dispositivos vienen con la opción IOMMU bloqueada, y la única manera de habilitar esto sería modificando el kernel. El cual, como hemos discutido, no se puede en sistemas operativos Windows.

- **Poca presión por la comunidad**
  Debido a que esta necesidad está concentrada en una porción muy pequeña de los usuarios, Microsoft ha mantenido la posibilidad de hacer passthroughs únicamente a las distribuciones empresariales de Windows, Windows Server 2016+. Estas son distribuciones de Windows hechas especialmente para el manejo de servidores, diferentes de la versión instaladas en las salas de la universidad que es hecha para máquinas individuales.

- **Kernel Windows**
  Debido a que el kernel de Windows es inmodificable, es la completa prerrogativa de Microsoft qué servicios provee el sistema operativo.

3.2.2 Contenedores Docker

Otra aproximación que tuvimos fue la de utilizar un contenedor Docker con un ambiente Linux, e intentar utilizar los recursos gráficos de la máquina a través del contendor. Sin embargo, Docker utiliza Hyper V para hacer la virtualización. Es posible hacer esta aproximación, pero únicamente si la máquina anfitriona corre sobre alguna distribución de Linux, ya que Docker tiene el soporte por medio de la aplicación nvidia-container-runtime. Pero para la infraestructura actual de la sala de computadores, no era viable.
Debido a estas razones, tocó descartar la posibilidad de mantener la infraestructura actual de las máquinas de las salas de cómputo de la universidad para realizar la virtualización gráfica. Sin embargo, en nuestras investigaciones sobre hipervisores, hubo un hipervisor de primer nivel que aparentaba tener la posibilidad de virtualizar dispositivos PCI.

3.3 ESXi

3.3.1 Propuesta de la plataforma

ESXi es el hipervisor de primer nivel de VMWare. Este reemplaza completamente el sistema operativo y sirve para desplegar entornos de virtualización. Lo que hace muy interesante a ESXi para la temática de esta investigación es la funcionalidad que ofrece ESXi de "Virtual Shared Graphics Acceleration" [Compartimiento Virtual de Aceleración Gráfica]. Sobre esta funcionalidad, VMWare provee tres distintas implementaciones que cubren diferentes casos de uso. Estas tres implementaciones son; MxGPU, vSGA y vDGA (6).

- **Virtual Dedicated Graphics Acceleration (vDGA)**
  vDGA es el mismo concepto de GPU Passthrough que llevamos explicando a lo largo del escrito, donde la tarjeta gráfica es compartida a una máquina virtual de manera dedicada. Esta implementación ofrece el mejor desempeño y rendimiento de la tarjeta gráfica. Sin embargo, es una virtualización completamente dedicada, lo que implica que queda imposible hacer un hotswap entre máquinas.

- **Virtual Shared Graphics Acceleration (vSGA)**
  vSGA permite compartir una sola GPU entre varias máquinas virtuales, "attractive solutions for users who require the GPUs full potential for brief periods of time." [atractiva solución para usuarios que requieren utilizar toda la potencia de la tarjeta gráfica por un breve periodo de tiempo]. Esta requiere de un uso cuidadoso de su uso en referencia a las aplicaciones usadas debido a que se pueden causar cuellos de botella. El mayor problema que presenta esta implementación es el hecho que las máquinas virtuales que quieran utilizar la tarjeta gráfica deberán correr sobre el driver propietario VMWare vSGA 3D que se comunica con VMWare vSphere. Además de esto, tiene limitado soporte del API.

- **Virtual Shared Pass-Through Graphics Acceleration (MxGPU)**
  Esta funciona similar a vSGA, con la diferencia de no correr sobre el driver de VMWare, y con la habilidad de correr la gran mayoría de las funcionalidades de la tarjeta gráfica. Dependiendo de la tarjeta, hasta 24 máquinas podrían utilizar los recursos de esta de manera concurrente. Esta solución tiene mejor rendimiento que vSGA.

De estas tres opciones, la que era más atractiva para el proyecto era MxGPU debido a su capacidad de correr sobre máquinas normales (que no corrieran sobre el driver de VMWare) y la capacidad de compartir el poder computacional de una misma tarjeta sobre varias máquinas. Esta fue con la que decidimos hacer unas pruebas piloto sobre los computadores del laboratorio ML340 de la universidad.

3.3.2 Prueba Piloto

En esta prueba se quería ver cómo era la experiencia de usuario y el rendimiento de la tarjeta gráfica al hacer una virtualización con ESXi tipo vSGA o MxGPU, con el propósito de estudiar la posibilidad de instalar ESXi en los computadores y servir una máquina Windows al usuario de la sala, y una o varias máquinas virtuales Linux que pueden utilizar los recursos de la tarjeta gráfica cuando esta esté en periodos de ocio. La prueba piloto la hicimos sobre una máquina del laboratorio ML 340 con la siguiente arquitectura:

Modelo	Dell Precision T1700
GPU	Nvidia Quadro K2200
Procesador	Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
Memoria	32GB RAM
Almacenamiento	500GB HDD

Sin embargo, llegamos a unos obstáculos que nos hicieron llegar a conclusiones tempranas sobre esta propuesta:

- La tarjeta gráfica del computador que utilizamos es compatible únicamente para utilizar vDGA, esto es el equivalente de hacer un passthrough normal desde Linux, perdiendo así toda la ventaja que habíamos ganado con la implementación de MxGPU.

- Por otro lado, las tarjetas gráficas de la sala Turing (la sala con las tarjetas gráficas donde se quiere implementar la solución) son de menor potencia que la de ML340 con una Nvidia Quadro P600. Por ende, no se puede utilizar ninguna de las opciones de ESXi sobre estas máquinas.

Debido a esto, y después de haber realizado los estudios sobre Docker, y Passthorugh, llegamos a la conclusión que la propuesta que íbamos a hacer no es posible con la infraestructura actual. Esto llevo al proyecto a cambiar de foco y empezamos a buscar alternativas que requirieran adquisición externa.
Lo lógico fue buscar las tarjetas gráficas que sí soportan MxGPU para pensar en proponer que cada máquina de la próxima generación de máquinas en Turing tuviera una de estas tarjetas. Sin embargo, estas son tarjetas de la gama más alta y son muy costosas. Pensar en comprar treinta y nueve de estas tarjetas representaría una inversión exorbitante.
La raíz del problema era que los recursos estuvieran fragmentados: que hubiera una tarjeta por cada computador y que a su vez se quisiera juntar todos esos computadores para hacer procesamiento computacional. Sin embargo, ya teníamos la solución para dividir el poder de computo, las tarjetas Nvidia con ESXi son capaces de servir de manera dinámica poder de cómputo a varias máquinas virtuales. Entonces decidimos cambiar el paradigma de la investigación: de cómo volver todas las tarjetas independientes un enorme proceso paralelo, a como coger un poder de computo centralizado y dividirlo de manera dinámica en varios nodos de procesamiento. Esto cambió el enfoque por completo del proyecto, y de ahí salió la idea de implementar un Desktop Pool.

4 Propuesta

4.1 Desktop Pool

4.1.1 Introducción

Un desktop pool es la arquitectura de un sistema donde desde un servidor central se sirven los escritorios y programas a los usuarios como un servicio (7). Los beneficios incluyen un poder escalable y centralizado de cómputo, donde se puede utilizar de manera directa los recursos no-utilizados de la sala de cómputo. La mantenibilidad de la sala mejoraría puesto que todo el mantenimiento se haría sobre un único servidor el cual se encarga de servirle el escritorio al usuario. Otra herramienta que se haría disponible seria la posibilidad de virtualizar el uso de las tarjetas gráficas de la sala, que en el momento son aprovechadas únicamente en un par de cursos del departamento, esta solución traería la posibilidad de, utilizando tecnologías como VMWare Horizon 7 y ESXi, la posibilidad de tener (en vez de una GPU por equipo) un número menor de tarjetas gráficas de gran capacidad de computo, cuyo poder de cómputo pueda ser repartido de manera elástica a las diferentes máquinas que estén siendo servidas y que estén requiriendo de este, el resto del poder de cómputo, podría estar siendo utilizado por los varios grupos de investigación que necesitan esta clase de procesamiento.

Esta implementación ha tenido unos casos de uso positivos en entornos mucho más grandes, lo que valida esta aproximación para un entorno pequeño como lo es la sala Turing.

4.1.2 Terminología

- VMware Horizon 7 Node (o simplemente un nodo)
  Una máquina host que contiene un número de escritorios virtuales en un despliegue de Horizon 7.

- Thin Clients
  La máquina a la que el usuario tiene acceso físicamente. Son escritorios muy livianos cuya única funcionalidad es poder manejar la máquina virtual que se despliega sobre ellos. Son dispositivos muy económicos que no requieren del mantenimiento usual requerido para un escritorio.

4.1.3 Estudio de beneficios

La sala Alan Turing tiene una serie de características que la hacen ideal para implementar un Desktop pool: Goza de una alta rotación de usuarios, tiene implementado un sistema de almacenamiento virtual que, aunque tiene altos requerimientos por parte de algunos usuarios, se mantiene subutilizada, no solo tienen los equipos 64 GB de RAM, sino que cada uno tiene una tarjeta gráfica. Debido a estas características, encontramos que hacer el cambio de escritorios individualizados a un desktop pool, le traería los siguientes beneficios a la sala y al departamento.
En primer lugar, es una sala donde los computadores son utilizados por una gran cantidad de personas, donde constantemente hay una rotación de usuarios sobre las máquinas. Adicionalmente, ya está implementado un almacenamiento virtual, donde lo que almacena cada estudiante en el computador, lo hace sobre la nube, quitando la restricción de necesitar usar el mismo computador para continuar con el mismo trabajo. Por ende, en términos de experiencia de usuario, no habría una diferencia notoria.

Por otro lado, si se implementa esta solución, el aprovechamiento de los recursos se maximizaría, puesto que una implementación de Desktop Pools sobre un ESXi que cuente con MxGPU podría tener un servidor central desde donde se sirven las máquinas virtuales y el cual tiene una cantidad de tarjetas gráficas que puedan alimentar cuando sea necesario, el procesamiento gráfico a los computadores de la sala, pero que cuando se esté en un periodo de inactividad, este poder de procesamiento pueda estar centralizado y disponible para ser utilizado por aplicaciones que puedan sacarle todo el provecho a los recursos.

En un tercer lugar, el mantenimiento de la sala y los costos a largo plazo se reducirían considerablemente, puesto que los dispositivos con los que los usuarios interactúan serían thin clients, no requerirían de hacer actualizaciones individuales, sino que estos cambios se hacen en la imagen que sirve el nodo.

Incluso se podría reutilizar las máquinas con las que ya cuenta la sala y dejarlas como thin clients para no necesitar la inversión inicial de los thin clients (Ver el Requerimientos de Infraestructura). Por ende, toda la inversión de cambio de máquinas se iría a la adquisición de un servidor con la potencia de servir 39 máquinas virtuales. Una vez pase un tiempo de la inversión inicial, se podría hacer una segunda inversión en thin clients, las cuales son máquinas muy simples, muy baratas que requieren un mantenimiento mínimo en comparación con las máquinas actuales. Además de la inversión inicial, este servidor no necesita ser reemplazado totalmente cada par de años, sino que puede gozar de actualizaciones graduales que lo permitan escalar de manera progresiva con los requerimientos computacionales de la sala.

Un cuarto componente que podría ser muy atractivo para el departamento sería el de ofrecer a los estudiantes que están viendo las clases que dependen de procesamiento gráfico acceso a estas máquinas virtuales de manera remota, ayudando así a estudiantes que no tengan acceso a estas tarjetas en sus computadores personales a seguir trabajando en sus clases. Por otro lado, más relevante a la situación actual, estas clases ya no tendrían problema de dictarse de manera remota, ayudando así a la universidad en sus esfuerzos de telestudio.

La quinta manera en la que esta implementación podría ser beneficiosa para mejorar el uso de la sala es en la identificación de diferentes tipos de workers. Horizon 7 permite establecer diferentes tipos de workers, Task Workers, Knowledge Workers, o Power Workers. La diferencia entre estos tipos de trabajadores es en sí las máquinas virtuales que les sirven tienen o no estado. "Knowledge workers must be able to create complex documents and have them persist on the desktop. Power users must be able to install their own applications and have them persist. Depending on the nature and amount of personal data that must be retained, the desktop can be stateful or stateless."[Knowledge workers debe tener la habilidad de crear complejos documentos que puedan persistir sobre el escritorio, Power Users deben poder instalar sus propias aplicaciones y que estas puedan persistir. Dependiendo en la naturaleza del trabajo y la cantidad de almacenamiento personal que debe ser retenida y persistida, los escritorios podrán tener o no estado](**?**). Teniendo esto en cuenta, se podrá optimizar de manera considerable el uso de almacenamiento y requerimientos computacionales, al identificar grupos de Power, Knowledge o Task workers para servir las máquinas virtuales que más se ajustan a las necesidades de estas, así evitando la asignación innecesaria de recursos.

4.1.4 Requerimientos de infraestructura

Ya teniendo una idea de los recursos que son utilizados de estas máquinas, es posible hacer unas aproximaciones de las necesidades mínimas que debería tener el servidor central de Horizon 7 para poder mantener la funcionalidad que provee actualmente a las máquinas de la sala.

Thin Clients Como ya fue discutido, una de las cosas más atractivas de esta solución es que el punto de entrada al sistema para un estudiante sentado en la sala no tiene que cambiar, puede ser la misma máquina que hay en este momento conectada a la máquina virtual. Una vez se quiera hacer la inversión adicional para adquirir thin clients, recomendamos utilizar los thin clients Wyse de Dell, específicamente el modelo **Wyse 5070 Thin Client**

- Intel® Pentium® Silver Processor J5005 1.5GHz (up to 2.80 GHz burst)
- 4GB 1x4GB, 2400MHz DDR Memory
- 16G eMMC Included in Chassis

Este tiene un costo de $589 por unidad.

CPU
Sobre CPU, VMware identifica dos tipos de Workers , el primero es el que más se apega al tipo de usuario que suelen utilizar la sala Turing, "Software developers or other power uses with high-performance needs might have much higher CPU requirements than knowledge workers and task workers. Dual or Quad virtual CPUs are recommended for 64-bit Windows 7 virtual machines running compute-intensive tasks such as using CAD applications, playing HD videos, or driving 4K display resolutions." [Desarrolladores u otros usuarios que requieran tareas con alto requerimiento de cómputo pueden necesitar mayor requerimiento concierne a CPU que otros knowledge o task workers. CPUs virtuales quad o duo, serían las recomendadas para usuarios utilizando máquinas virtuales Windows 10 de 64 bits corriendo aplicaciones como CAD, videos y alta definición o pantallas 4k]. No consideramos entrar en detalle sobre los otros tipos de workers puesto que este perfil define lo que necesita la sala en sus máquinas. Sobre esto, recomendamos los siguientes procesadores Intel que son compatibles con las tarjetas RAM recomendadas en el inciso siguiente. También cabe resaltar que la columna Max Memory es la capacidad máxima de memoria que se puede poner en cada uno de estos procesadores, y no está incluida en el precio.

Model	# Cores	Base Frequency (Ghz)	Turbo Frequency (GHz)	Cache (MB)	Max Memory Size (TB)	Memory Type	Price* (USD)
Intel® Xeon® W-2245 Processor [1] (9)	8 (16 Threads)	3.9	4.5	16.5	1	DDR4-2933	667
Intel® Xeon® W-2275 Processor (10)	14 (28 Threads)	3.3	4.6	19.25	1	DDR4-2933	1112
Intel® Xeon® W-2295 Processor (11)	18 (36 Threads)	3	4.6	24.75	1	DDR4-2933	1300

RAM

VMware enfatiza la importancia de tener una cantidad suficiente de RAM sobre las máquinas virtuales. No tener el RAM suficiente, lleva a una paginación excesiva, lo que causa problemas de desempeño y latencia e incrementa la carga sobre I/O. Sin embargo, ESXi y VMware tienen implementados sofisticados algoritmos que reducen la cantidad de memoria física que consume cada usuario considerablemente.

VMware recomiendo alocar inicialmente, durante un periodo de prueba 4 GB de RAM en máquinas Windows de 64 bits, y 2 Cores de CPU virtuales. Sin embargo, es válido afirmar que los usuarios de la sala Turing pueden correr aplicaciones pesadas, por lo que se recomendaría iniciar la etapa de prueba con 8GB alocadas físicamente por cada usuario.

39 Computadores x 8 GB = 312GB

A comparación de las actuales 32 GB x 39 Computadores = 1250 GB Usadas actualmente.

En el paper presentado anteriormente, fue demostrado que las máquinas de la sala Waira tienen picos de consumo de RAM de un 30%, esto es equivalente a

[1]

tener 9 GB. Por ende, poner a disposición mínima 8 GB por cada máquina sería un buen punto de partida para la capacidad de RAM de la sala. Sin embargo, las 8 GB serian el máximo cuando este en máxima ocupación la sala, en el resto de las configuraciones, la asignación de RAM es elástica según lo requiera la instancia. Esto es considerablemente menor que la memoria actualmente instalada por cada computador, pero al mismo tiempo, es una inversión mínima en comparación con la inversión que se hace normalmente a la memoria RAM, por esto, en la sección de Proyección de Inversión discutimos que infraestructura podría adquirir el departamento si se hiciera la misma inversión que se hizo para los computadores de la sala Turing anteriormente. No obstante cuanta memoria se vaya a comprar, nuestra recomendación de tarjetas es la siguiente

Model	DIMM Type	Memory Size (GB)	Memory Speed (GHz)	Voltage (V)	Manufacturer	Price* (USD)
4X70V98063 (12)	RDIMM	64 GB	DDR4-2933	1.2V	Samsung	324.00
01AG633 (13)	LRDIMM	64 GB	DDR4-2933	1.2V	Samsung	348.00

Tomando la memoria más cara, la inversión de 312GB de RAM constituiría de USD 1740.

GPU

Hay dos empresas que en este momento llevan la delantera en temas de tarjetas gráficas. AMD y Nvidia. Sin embargo, Nvidia se ha mostrado superior en el mercado de alta gama, con mejor compatibilidad debido a su driver nvidia-driver el cual sirve para hacer manipulaciones de bajo nivel sobre la tarjeta, además a eso, Nvidia ha tenido de manera consistente mejor desempeño que las AMD. Un último punto a favor de recomendar tarjetas Nvidia es el hecho que desde hace varios años el departamento ha comprado tarjetas Nvidia, y probablemente los cursos que usan de ellas dependen de que se mantenga la misma tecnología para no tener que ajustar las asignaturas a las nuevas herramientas.

Al momento de escribir esto, VMWare ESXi MxGPU tiene listado como tarjetas compatibles de Nvidia dos familias de tarjetas, la Tesla y la Quadro(16). Las tarjetas Tesla son tarjetas dedicadas únicamente a cómputo y no tienen ningún puerto de video. Estas son utilizadas en clusters industriales de cómputo y no les sirven a los laboratorios de la universidad puesto que las clases que utilizan esas tarjetas las necesitan para correr software como Unity y Unreal Engine. Por esto, la familia de tarjetas que mejor se adapta a las necesidades de la universidad son las Quadro puesto son hechas con procesos gráficos en mente.

Para que la implementación de Desktop Pool permita la virtualización de GPU's, es necesario adquirir tarjetas que sea compatibles con ESXi 6.5, y que sean de Nvidia, siendo que es la marca que lleva usando el departamento en las tarjetas anteriores y que: al ser Nvidia, pertenezcan a la familia Quadro, ya que estas son las tarjetas hechas para procesamiento gráficos.

Model	CUDA Parallel-Processing Cores	GPU Memory	Max Power Consumption	Display Connectors	Price* (USD)
Quadro RTX 6000 (14)	4,608	24 GB GDDR6	295 W	DP 1.4 (4), VirtualLink (1)	4,000
Quadro RTX 8000 (15)	4,608	48 GB GDDR6 with ECC	Total Graphics Power: 260W	DP 1.4 (4), VirtualLink (1)	5,500

Aunque esta inversión para una única tarjeta puede parecer exorbitante, es necesario resaltar que utilizando MxGPU, estas tarjetas llegan a una rata de consolidación de 1:32, lo que significa que una sola tarjeta la pueden utilizar de manera elástica hasta 32 Power Users. No obstante, para mantener el excelente desempeño de las máquinas de la sala, no se recomienda usar una única tarjeta, sino en vez manejar una rata de consolidación de 1:10, equivalente a 4 tarjetas.

Almacenamiento Al tiempo de este estudio, la sala ya cuenta con un sistema de almacenamiento en la nube, entonces no es claro si el almacenamiento individual de los usuarios sería parte de la propuesta de inversión de la plataforma.

4.1.5 Caso de estudio

Para aterrizar un poco lo que hemos discutido sobre el Desktop Pool, es relevante ver un caso de éxito de una implementación parecida. Aunque el caso de estudio que vamos a presentar tiene un segmento de usuarios diferente, ayuda mucho para ilustrar los beneficios de esta solución frente una solución de infraestructura más tradicional.

Carrol Hospital Carrol Hospital. El hospital Carrol Hospital Center quería simplificar el manejo de los escritorios empresariales mientras al mismo tiempo, tener más control sobre los recursos de la empresa y mejorar la seguridad de los datos provenientes de los usuarios del sistema. Para resolver esta necesidad, se decidió implementar la estructura de virtualización de VMware Horizon 7 para virtualizar 800 escritorios de la empresa, con la posibilidad de incrementar esta cifra a 1500 usuarios.
Las características que debía tener el sistema para poder satisfacer al hospital eran:

- **Escalabilidad**: Debe soportar 800 clientes, con la posibilidad de que suban a 1500 sin ninguna pérdida en desempeño.
- Costos: Reducción en costos a un largo plazo.

- Mantenibilidad: Debe poder ofrecer un fácil mantenimiento de los clientes, con despliegues automatizados.

Teniendo en cuenta estos requerimientos y características, concluyeron que para cada clúster de servidores, requerirían 8 nodos. Donde cada nodo tiene dos CPU's de 2.7 GHZ con dieciséis (16) cores lógicos y 288 GB de RAM. Cada uno de estos nodos estaba estimado poder soportar 128 usuarios.

Para esta estimación de cuanto poder de cómputo requería cada usuario, hicieron como es aconsejable según las buenas prácticas de construcción de arquitecturas, en hacer el estudio de workers y entender bien los tipos de usuarios que van a tener acceso al servicio y entender sus necesidades en términos de poder de cómputo. Este identificó dos tipos de usuario, el primero siendo el equipo de enfermería, que consiste en 800 personas y que requieren floating desktops. El segundo tipo de worker es el equipo de TI, que consiste en 25 personas y requieren dedicated desktops.

En una primera fase, se usaron los escritorios que ya tenía la empresa para servir las máquinas virtuales, pero está en el plan a largo plazo reemplazar todos estos escritorios por thin clients.

Al finalizada la implementación del ambiente de virtualización, el hospital contaba con una mejorada infraestructura tecnológica, donde la manutención de su infraestructura tecnológica es mucho más simple, teniendo que renovar los thin clients que, en comparación con un escritorio entero, son mucho más baratos y no requieren de mantenimiento individual, sino que las actualizaciones se hacen a los nodos que sirven a las máquinas virtuales.

4.1.6 Pruebas de estrés

Para probar que tan resistente es el sistema a cargas excesivas de trabajo, decidimos implementar un despliegue pequeño de un servidor vCenter sobre una máquina con las mismas especificaciones que la máquina donde realizamos las pruebas de ESXi.
Sobre esta máquina se configuró un hipervisor ESXi, y se desplegó un vCenter que se encarga de administrar 5 máquinas Virtuales, cuatro de ellas con sistema operativo Ubuntu 18, y una de ellas Windows 10 Pro. Cada máquina contaba con un máximo de 4 CPUs, 8 GB RAM y 40 GB de almacenamiento. La intención de esta prueba era poner bajo mucho estrés las cuatro máquinas Ubuntu y ver si este procesamiento excesivo tenía algún efecto sobre la máquina Windows.
Para hacer la prueba de estrés se utilizó la herramienta stress-ng que genera escenarios de estrés en sistemas Linux, esta herramienta permite simular distintas situaciones que generan estrés en la máquina, como operaciones I/O, alocar estrés sobre el CPU, generación de interrupciones, bloqueo de semáforos y memoria compartida, entre otras. Para esta prueba decidimos generar la siguiente carga

sudo stress-ng –cpu 2 –vm 4 –vm-bytes 2G –t 7m -v
Esto representa una carga sobre dos CPU's, dos operaciones exigentes en la memoria virtual con bloques de 2GB por 7 minutos.
Mientras esta prueba se ejecutaba sobre las máquinas virtuales Linux, en la máquina virtual Windows se navegaba por internet, se abría un editor de archivos, estudiando como reaccionaba y como se comportaba el sistema. En este caso debería llegar a máximo de CPU usado, ya que la máquina tiene máximo 8 cores virtuales para asignar. Y la memoria debería quedar en 40/32 GB utilizadas, entonces este era una prueba muy interesante para ver cómo se comportaba el sistema en una situación extraordinaria. Se esperaba que la máquina Windows fuera innavegable, que hubiera GUI las y excesiva paginación. Las gráficas a continuación muestran el consumo de recursos sobre el host, la máquina Windows, y los clones Linux. La simulación de carga comenzó a las 12.23. De

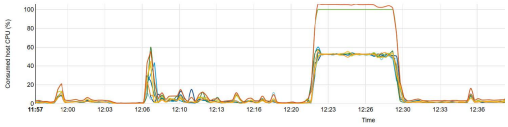

Figure 4: Consumo de la máquina host durante la prueba

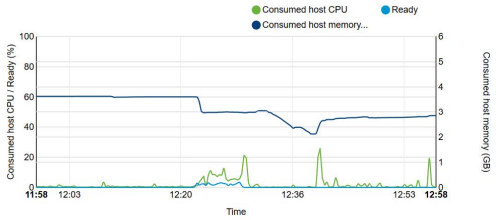

Figure 5: Consumo de la máquina Windows durante la prueba

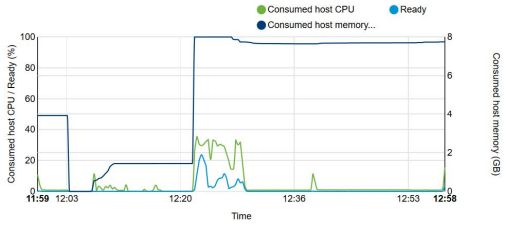

Figure 6: Consumo de la máquina Linux durante la prueba

estas gráficas podemos ver un interesante comportamiento del sistema. Primero podemos ver como la máquina Windows redujo su capacidad de memoria RAM

cuando empezó la carga sobre las otras máquinas, bajando de 4 GB de RAM a 3 e inclusive bajando hasta 2Gb. Esto muestra la elasticidad y el dinamismo que tiene el sistema para alocar recursos donde más se están solicitando. Por otro lado, una de las 4 máquinas Ubuntu (la que fue desplegada de última) le fue alocado únicamente 2GB de RAM, aun así en la configuración tuviera las 8GB alocadas a las otras copias.

Por otro lado, en la instancia Windows, no hubo excesiva paginación, aunque si se vio unos momentos de GUI lag. Por un lado, esta prueba valida lo que decía VMWare frente a el manejo de los CPUS virtuales, puesto que pudo soportar, con 4 cores físicos y 8 virtuales, a 5 máquinas realizando procesos exigentes para el computador. Sin embargo, el performance de la máquina Windows no pasó la prueba completamente, lo que nos deja con una conclusión muy importante; el sistema, aunque muy resistente y resiliente, debe ser dimensionado de la manera correcta para poder soportar una situación donde los recursos estén siendo puestos a la prueba.

4.1.7 Perfilamiento de usuarios

Sin embargo, esta simulación nos hizo caer en cuenta la importancia que tiene perfilar bien los distintos usuarios de los nodos de Horizon 7 para que se tenga un control sobre el poder de cómputo que se sirve a las instancias. Además de esto, la coyuntura en la que estamos viviendo en este momento, con un semestre 2020-2 virtual, obliga al departamento ser capaz de proveer máquinas virtuales capaces de correr los programas de las asignaturas a estudiantes que dependen de las máquinas de las salas Waira y Turing para hacer sus tareas y proyectos. Por eso definimos cuatro tipos de usuarios, los power workers, o nodos de investigación, knowledge workers, o los nodos de estudiantes autorizados, specialized knowledge workers o nodos de estudiantes autorizados especiales Y por último, los casual workers o los nodos livianos.

- **Nodos de investigación**:
  Grupos de investigación o proyectos con permisos especiales que tengan necesidades de cómputo superior a otros usuarios. Debería tener límite de procesamiento casi ilimitado y que sea restringido únicamente por los requisitos de los nodos de estudiantes o los nodos livianos. La disponibilidad de aplicaciones no debería estar restringida.

- **Nodos de estudiantes**:
  Estos nodos buscan emular el poder de cómputo que se encuentra en las máquinas de la sala Turing. A estos se puede acceder únicamente desde los thin clients en la Sala. La oferta de aplicaciones debería ser restringida para poder asegurar que se utilice para propósitos académicos. El poder de computo que debe ser alocado para estas máquinas se debería ir ajustando a lo que se va estudiando mejor la demanda de los estudiantes, pero se

podría comenzar con un límite de 8 GB físicas de RAM por usuario y 3 4 CPUs virtuales por cada máquina.

- **Nodos de estudiantes especial**:
  La idea de este perfil es poderle dar la posibilidad a estudiantes de clases con requerimientos extraordinarios a ser asignados máquinas con una mayor cantidad de recursos. Ampliando a 16GB de RAM y 6 CPUS por usuario. Se debería limitar el uso de estos para no saturar el sistema y limitar los recursos de los nodos de investigación, ni limitar mucho la cantidad de nodos livianos disponibles.

- **Nodos livianos**:
  La idea de este perfil, es poder alocar un número máximo de nodos livianos que puedan ser accedidos por usuarios del departamento de manera remota, esto para poder proveer máquinas virtuales específicas para estudiantes que tienen problemas en conseguir un computador con los programas necesarios o que no tiene la capacidad de correr los programas requeridos por sus clases. La idea es que esto sea una cantidad limitada y que se ajuste la oferta de nodos livianos dependiendo del porcentaje de uso que le estén dando a los otros nodos. Así, en horas cuando las máquinas físicas del laboratorio no se estén usando, más estudiantes tengan la posibilidad de seguir trabajando desde sus casas y así incrementar la tasa de uso de los recursos. Estas podrían ser máquinas virtuales livianas sin estado que permitan acceder a una lista restringida de aplicaciones que son utilizadas en las clases del departamento.

4.1.8 Proyección de inversión

Hasta ahora hemos intentado únicamente empatar el nivel de cómputo actual de la sala, para que los usuarios no pierdan ni experiencia de usuario, ni potencial computacional. Sin embargo, el costo de la infraestructura que hemos propuesto es mucho menor que el presupuesto que se ha designado a cada máquina.
Actualmente, cada máquina de Turing costó 1500 dólares, que para 39 máquinas equivalen a USD $58,500.00. En las tablas a continuación se desglosa los gastos en cada propuesta.
Propuesta de requerimientos mínimos

		RAM		
Cantidad	Memoria c/u	Memoria Total	Precio c/u	Precio Total
5	64 GB	320 GB	USD 324.00	USD 1,620.00

Utilizando el modelo (12)

		CPU		
Cantidad	# Cores c/u	# Cores Total	Precio c/u	Precio Total
4	18	72	USD 1,300.00	USD 5,200.00

Utilizando el modelo (10)

	GPU	
Cantidad	Precio	Precio Total
4	USD 4,000.00	USD 16,000.00

Utilizando el modelo QUADRO RTX 6000 (14)

Esta estimación es equivalente a lo siguiente

- Rata de consolidación de 1:10, por cada 10 máquinas, habría un nodo sirviendo esas máquinas.
- GB de RAM por cada máquina: 8.2GB
- Cores disponibles por cada máquina: 1.85
- Costo total: USD 22,920.00

Ahora, si se fuera a emplear un presupuesto similar al que se gastó para la compra de máquinas anterior, se podría adquirir la siguiente infraestructura:

Propuesta con Presupuesto similar al pasado

		RAM		
Cantidad	Memoria c/u	Memoria Total	Precio c/u	Precio Total
16	64 GB	1024 GB	USD 324.00	USD 5,184.00

Utilizando el modelo (12)

		CPU		
Cantidad	# Cores c/u	# Cores Total	Precio c/u	Precio Total
16	18	288	USD 1,300.00	USD 20,800.00

Utilizando el modelo (10)

	GPU	
Cantidad	Precio	Precio Total
8	USD 4,000.00	USD 32,000.00

Utilizando el modelo QUADRO RTX 6000 (14)

Esta estimación es equivalente a lo siguiente

- Rata de consolidación de 1:4.85, por cada 4.85 máquinas, habría un nodo sirviendo esas máquinas.

- GB de RAM por cada máquina: 26.3 GB
- Cores disponibles por cada máquina: 7.4
- Costo total: USD 57,940.00

Estas dos propuestas pueden sonar muy extremas, pero en realidad hay todo un espectro de configuraciones que se pueden armar para llegar a la infraestructura óptima tomando en cuenta las varias coyunturas en las que se encuentra el departamento y la universidad. Variables como la valoración de la moneda, y el presupuesto que se conceda al departamento van a afectar el poder adquisitivo, y por ende el tamaño de la infraestructura que se pueda adquirir. No obstante, esta solución de infraestructura tiene un buen margen de error, puesto que es fácilmente escalable según las necesidades que va teniendo la sala, no se compromete con un conjunto de máquinas por unos años, sino que esta infraestructura se puede escalar y desescalar dependiendo de las necesidades actuales.

Comparación Es pertinente hacer una comparación entre las tarjetas que hay en este momento en las máquinas de la sala Turing, contra las Quadro RTX 6000. Esto con el fin de poder afirmar que una de las nuevas tarjetas es capaz de reemplazar cuatro de las Quadro P600.

Quadro P600

- GPU Memory: 2 GB GDDR5
- NVIDIA CUDA Cores: 384
- Memory Interface: 128-bit
- Memory Bandwidth: 64GB

Quadro RTX 6000

- GPU Memory: 28 GB GDDR6
- NVIDIA CUDA Cores: 4,608
- Memory Interface: 384-bit
- Memory Bandwidth: 672 GB

La comparación entre estas dos tarjetas hace perfectamente claro que en todos los campos de comparación, la Quadro RTX 6000 es múltiples veces más poderosa que la P600, llevando a la conclusón de que, si la virtualización de VMWare de MxGPU es confiable, no debería haber ningún problema establecer una tasa de consolidación de 1:4. E inclusive se podrá notar un incremento en la capacidad de cómputo.

5 Conclusiones

5.1 Discusión

Consideramos que si esta propuesta es implementada en la sala Turing, tendría muchos beneficios sobre hacer una inversión de 39 máquinas muy poderosas que no serían apropiadamente aprovechadas. Primero, se lograría aprovechar los recursos computacionales de las tarjetas gráficas, logrando tener un centro de cómputo muy poderoso, que cuando solo haya nodos de investigación trabajando, van a ser capaces de gozar de un sistema robusto preparado para hacer cualquier tipo de procesamiento. Por otro lado, según la propuesta financiera, nuestra implementación representaría una inversión mucho menor puesto que no requiere hacer la compra de máquinas preparadas de manera independiente para soportar cargas de hasta 32GB por un únicos estudiante, sino que partimos de la base que la mayoría de los estudiantes utilizan muy pocos recursos de las máquinas, y contamos con la facilidad de aumentar los recursos para los estudiantes que sí están utilizando más poder de procesamiento de manera automática.

Por otro lado, consideramos que esta propuesta es muy oportuna con la condición actual de la universidad en torno a las medidas de distanciamiento que se están implementando en la Universidad, puesto que es muy alta la probabilidad de tener un semestre semipresencial, donde se va a restringir mucho el acceso a salas y laboratorios. Esta solución permitiría dar acceso a las máquinas de los laboratorios a estudiantes que no gozan de la posibilidad de tener buenas máquinas. Inclusive se podría pensar en alquilar los thin clients que únicamente se puedan conectar a la máquina virtual de la universidad y así se provee varias soluciones a estudiantes para poder mitigar un poco la disparidad de acceso a recursos entre los estudiantes del departamento.

Considero que esta investigación es un éxito desde el punto de vista de los objetivos que nos impusimos al comenzarla, primero que todo, encontramos la mejor manera de asegurar que los recursos de cómputo de la universidad no sean desperdiciados. Puesto que implementar el desktop pool en Turing daría lugar a situaciones como que si no hay ningún estudiante usando los computadores de la sala, estos se les dé de manera dinámica el 1% del poder de la sala y el restante este siendo utilizado en varios proyectos, iniciativas e investigaciones que pueda tener el departamento, o inclusive que la mitad de ese poder restante este siendo utilizado por estudiantes de manera remota. Esto flexibiliza y le da sentido a la inversión hecha por la universidad en términos de infraestructura computacional.

5.2 Trabajo a Futuro

Intentamos a lo largo de este proyecto cubrir todas las posibles variables que entran en juego en esto. Sin embargo, lo que estamos proponiendo es una idea nueva que representa un cambio considerable para el departamento. Esto lleva

a que haya mas trabajo por delante antes de poder tomar una decisión decisiva. Primero seria relevante entender los requisitos que le trae esta infraestructura a la red, esto con el fin de evitar la formación de cuellos de botella donde se pierda desempeño. Por otro lado, sería preciso estudiar mejor los patrones de consumo de las tarjetas gráficas por parte de clases en la universidad con el fin de tener una mejor idea de cuanto poder se necesita y entender también cuanto poder de cómputo se podría utilizar en asuntos diferentes al de los cursos.

References

[1] UnaCloud. (2020, April 21). Retrieved from https://github.com/UnaCloud

[2] O'Reilly. (2017). Introduction to GPUs for Data Analytics.

[3] Msv, J. (2017). In The Era Of Artificial Intelligence, GPUs Are The New CPUs. Forbes. Retrieved from https://www.forbes.com/sites/janakirammsv/2017/08/07/in-the-era-of-artificial-intelligence-gpus-are-the-new-cpus/81c65115d16e

[4] Pci passthrough - Proxmox VE. (2020, April 23). Retrieved from https://pve.proxmox.com/wiki/Pci_passthrough

[5] Warren, T. (2020). Microsoft: we were wrong about open source. Verge. Retrieved from https://www.theverge.com/2020/5/18/21262103/microsoft-open-source-linux-history-wrong-statement

[6] https://techzone.vmware.com/resource/deploying-hardware-accelerated-graphics-vmware-horizon-7#sec4-sub3

[7] VMWare. (2019, September 27). Advantages of Desktop Pools. Retrieved from https://docs.vmware.com/en/VMware-Horizon-7/7.1/com.vmware.horizon.virtual.desktops.doc/GUID-77C863FA-5D11-40D2-8867-37BA86D2AE73.html

[8] HPCLATAM-CLCAR Latin American Joint Conference (2nd : 2015 : Petropolis, Brazil). (2015). High performance computing : second latin american conference, carla 2015, petrøpolis, brazil, august 26-28, 2015, proceedings. (C. Osthoff, P. O. A. Navaux, C. J. Barrios Hernandez, & P. L. Silva Dias, Eds.) (Ser. Communications in computer and information science, 565). Springer. https://doi.org/10.1007/978-3-319-26928-3

[9] https://ark.intel.com/content/www/us/en/ark/products/198609/intel-xeon-w-2245-processor-16-5m-cache-3-90-ghz.html

[10] https://ark.intel.com/content/www/us/en/ark/products/198010/intel-xeon-w-2275-processor-19-25m-cache-3-30-ghz.html

[11] https://ark.intel.com/content/www/us/en/ark/products/198011/intel-xeon-w-2295-processor-24-75m-cache-3-00-ghz.html

[12] https://memory.net/product/4x70v98063-lenovo-1x-64gb-ddr4-2933-rdimm-pc4-23466u-r-dual-rank-x4-replacement/

[13] https://memory.net/product/01ag633-lenovo-1x-64gb-ddr4-2933-lrdimm-pc4-23466u-l-quad-rank-x4-replacement/

[14] https://www.nvidia.com/en-us/design-visualization/quadro/rtx-6000/

[15] https://www.nvidia.com/en-us/design-visualization/quadro/rtx-8000/

[16] https://www.vmware.com/resources/compatibility/search.php?deviceCategory=sptgdetails=1partners=6
[17]architectureplanninghttps://docs.vmware.com/en/VMware-Horizon-7/7.9/horizon-architecture-planning/GUID-5CC0B95F-7B92-4C60-A2F2-B932FB425F0C.html