



**Instituto Universitario Aeronáutico**

# **Comparación de Performance de Hipervisores**

**Rodrigo Javier Fernández**  
**Proyecto de Grado**  
**2015**



## Índice

1. Introducción .....	- 5 -
2. Hipervisores a Analizar .....	- 9 -
2.1. Vmware ESXi 6.0 .....	- 9 -
2.2. Microsoft Hyper-V Server 2012 R2 .....	- 10 -
2.3. Citrix Xenserver 6.5 .....	- 12 -
3. Metodologías de prueba existentes .....	- 15 -
4. Metodología de prueba utilizada .....	- 17 -
5. Aplicaciones incluidas en las pruebas .....	- 19 -
5.1. Servidor de Correo .....	- 19 -
5.2. Máquina Virtual Inactiva .....	- 20 -
5.3. Aplicación Web 2.0.....	- 21 -
5.4. Aplicación de Comercio Electrónico.....	- 21 -
6. VMmark Harness.....	- 22 -
7. Cliente VMmark .....	- 24 -
8. Máquinas Virtuales y Tiles .....	- 24 -
8.1. Configuración de Máquinas Virtuales .....	- 27 -
9. Hardware Utilizado .....	- 28 -
10. Despliegue de Sistemas .....	- 29 -



10.1. Instalación Hipervisor .....	29 -
10.1.1. Instalación ESXi 6.0.....	29 -
10.1.2. Instalar XenServer 6.5 .....	29 -
10.1.3. Instalar Hyper-V Server 2012 R2.....	30 -
10.2. Despliegue del Tile .....	30 -
10.2.1. Despliegue del Servidor de Correo .....	31 -
10.2.2. Despliegue de la Máquina Virtual Inactiva.....	33 -
10.2.3. Despliegue de la Aplicación Web 2.0 .....	34 -
10.2.4. Despliegue de la Aplicación de Comercio Electrónico .....	36 -
10.2.5. Despliegue Clientes para los Tiles .....	38 -
10.2.6. Despliegue del Primeclient.....	40 -
11. Ejecución de las pruebas .....	52 -
12. Evaluación.....	57 -
12.1. Resultados 1 Tiles .....	58 -
12.2. Resultados 3 Tiles .....	62 -
12.3. Resultados 4 Tiles .....	65 -
13. Conclusiones.....	69 -
14. Referencias.....	71 -



## **Resumen**

La virtualización de la infraestructura IT permite los recursos se consoliden y puedan ser compartidos sobre diversas aplicaciones para compensar sus limitaciones y las crecientes necesidades de negocio aprovechando al máximo dichos recursos. La virtualización abstrae los recursos físicos informáticos y crea entornos en los cuales no existen limitaciones dadas por las configuraciones y capacidades físicas de los equipos o su implementación. La virtualización es muy importante para la computación en la nube, dado que la prestación de servicios se simplifica al proporcionar una plataforma que utiliza los recursos IT de una manera óptima y escalable, disminuyendo los costos en infraestructura, energía y administración.

El Hipervisor juega un papel fundamental en la virtualización de servidores al ser el pilar de la misma. Es un software que proporciona un entorno de hardware virtualizado para soportar múltiples sistemas operativos corriendo al mismo tiempo en el mismo servidor físico. La computación en nube tiene que soportar múltiples ambientes heterogéneos y los Hipervisores permiten alcanzar este objetivo.

La intención de esta tesis es de comparar el desempeño de VMware ESXi 6, Citrix XenServer 6.5 y Microsoft Hyper-V Server 2012 R2. Para ello, se utilizará el benchmark de virtualización gratuito propietario de VMware, VMmark. Este deberá ser adaptado para funcionar en un solo Hipervisor de cualquier fabricante generando carga similar a encontrada en servidores de producción reales, y coleccionar los datos de desempeño correspondientes que serán posteriormente analizados y comparados.



## **1. Introducción**

La utilización de la informática ha crecido de manera exponencial en las últimas décadas acompañada por un importante desarrollo y crecimiento de las telecomunicaciones que le permitió alcanzar prácticamente todos los sectores de la sociedad.

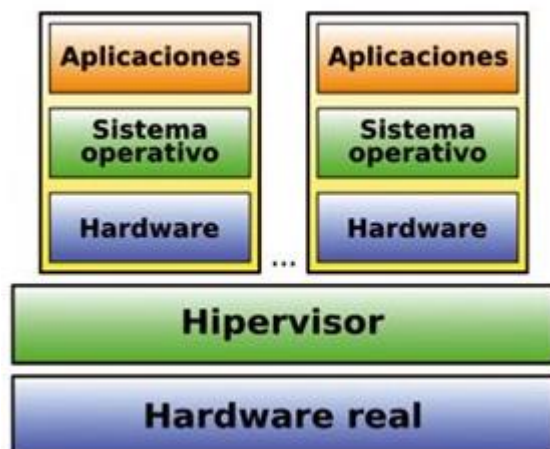
Este crecimiento estrepitoso de la demanda de servicios informáticos significó un crecimiento de igual medida de la infraestructura IT que soporta dichos servicios. Esto planteó el desafío de implementar y administrar datacenters a gran escala. Para la mayoría de las compañías la inversión que esto significa, redes de alta velocidad, sistemas de refrigeración, ups, generadores eléctricos, servidores y racks está más allá de sus posibilidades financieras. Incluso para aquellas que cuentan con los recursos necesarios, es imperativa la optimización de la infraestructura IT a las escalas requeridas para que sea financieramente viable. Esto requiere que no se desperdicien los recursos computacionales que no se utilizan de modo que se reduzcan los costos asociados.

Es aquí donde entra la virtualización, permitiendo que múltiples servidores virtuales corran de manera independiente sobre un servidor físico, hasta inclusive sobre aprovisionándolo. Esto permite aprovechar al máximo los recursos del servidor físico, reduciendo la cantidad de infraestructura física necesaria para soportar la demanda de servicios IT.

Una capa de virtualización provee el soporte de infraestructura usando los recursos de bajo nivel para crear múltiples máquinas virtuales que son independientes y están aisladas entre sí. Dicha capa de virtualización se llama Hipervisor. Si bien tradicionalmente Hipervisor se refiere a tradicionalmente a la capa de virtualización que está por encima del hardware y por debajo del sistema operativo, nos podemos referir a él como una capa genérica en muchos casos. [Nanda]

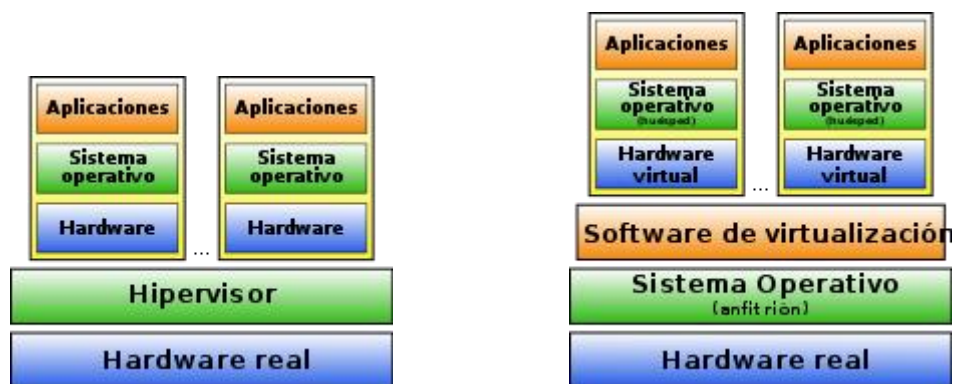


Los distintos niveles de virtualización son niveles de conjuntos de instrucciones: nivel de capa de abstracción de hardware (HAL), nivel de Sistema Operativo (Interfaz de llamada de sistema), nivel de aplicación o nivel de librerías de usuario. Una máquina virtual representa un ambiente operativo para un grupo de aplicaciones de usuario, las cuales incluyen librerías, interfaces/servicios de llamadas de sistema, configuraciones de sistema, servicios y estado del sistemas de archivos. En cualquier nivel de abstracción, el principio general es el mismo en las particiones de recursos de nivel más bajo usando técnicas novedosas para mapear a múltiples máquinas virtuales de niveles más elevados de forma transparente. [Nanda]



El hipervisor es el software de virtualización que corre entre el hardware físico y la máquina virtual, proveyéndole al sistema operativo de la misma una vista virtualizada del hardware. El sistema operativo huésped, el que corre en la máquina virtual, corre como si lo hiciera directamente sobre el hardware físico.

Existen dos tipos de hipervisores: los de tipo 1 que corren directamente sobre el hardware y del tipo 2 que corren sobre un sistema operativo huésped. Para este trabajo se utilizarán hipervisores del tipo 1 que son los de uso comercial.



Hipervisor Tipo I

Hipervisor Tipo II



Para la virtualización de servidores hay tres técnicas de virtualización:

- Virtualización completa: el hipervisor la máquina virtual corre un sistema operativo huésped sin modificar que no está al tanto de la capa de virtualización. Se utiliza traducción binaria y técnicas de ejecución directa para crear máquinas virtuales capaces de correr sistemas operativos huéspedes propietarios. Estas técnicas pueden generar mucho procesamiento adicional dado que las instrucciones que manipular recursos protegidos deben ser interceptadas y reescritas. Como resultado, los fabricantes de procesadores incorporaron soporte para virtualización a sus productos. De este modo el hipervisor puede delegar eficientemente el acceso a recursos restringidos.
- Para-virtualización: El sistema operativo huésped es modificado de modo que sabe que fue virtualizado y sabe cómo realizar solicitudes al hipervisor cuando necesita acceder a recursos restringidos. Esto simplifica la abstracción de hardware que debe ser provista, pero se dificulta el control de versión entre el hipervisor y el sistema operativo huésped para-virtualizado dado que son controlados por organizaciones distintas.
- Virtualización al nivel de Sistema Operativo: Esta técnica provee una arquitectura eficiente para cada sistema operativo que corre en una instancia distinta. Cada máquina virtual es tratada como un proceso y usa el planificador de CPU para recibir tiempo de CPU.



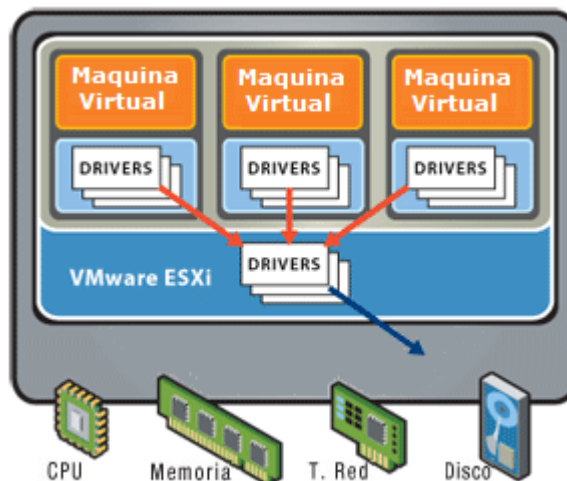
## 2. Hipervisores a Analizar

### 2.1. Vmware ESXi 6.0

Vmware ESXi es un hipervisor de tipo I capaz de migrar las máquinas virtuales en caliente y arrancarlas desde dispositivos conectados a la red. ESXi es una versión liviana, carece de la consola incluida en su predecesor ESX. ESXi 6.0 soporta virtualización completa.



Al ser un Hipervisor de tipo I se instala directamente sobre el servidor y administra todas las instrucciones I/O, para lo cual requiere la instalación de los controladores necesarios. Mantiene réplicas de las estructura de sistema, como la tabla de páginas y la mantiene consistente con las tablas virtuales interceptando cada instrucción que intenta actualizar estas estructuras. Así hay un nivel extra de mapeado en la tabla de páginas. Las páginas virtuales son mapeadas a las páginas físicas a través de la tabla de páginas del sistema operativo huésped de la máquina virtual. ESXi entonces traduce la página física (también llamada trama) a la página de la máquina virtual, la cual es la página en la memoria física. Esto ayuda al ESXi a manejar mejor la memoria y aumentar el rendimiento general del sistema. [Barham].



Este producto se utiliza para la consolidación de servidores, permitiendo reducir significativamente el tamaño de los datacenters. Utiliza otras técnicas como memory ballooning, páginas compartidas, compresión, swapping entre otras para mejorar la performance global y mantener las máquinas virtuales aisladas unas de otras. Esto lo hace un sistema confiable para uso comercial.

## 2.2. Microsoft Hyper-V Server 2012 R2

Este hipervisor también es de tipo I, tiene mucha funcionalidad nueva con respecto a su versión anterior, Hyper-V Server 2008 R2. Esta funcionalidad incluye mejoras en la administración, virtualización de red, replicación de las máquinas virtuales, mejora en la migración de las máquinas virtuales y migración de los archivos de las máquinas virtuales, asignación de adaptadores de red físicos a una máquina virtual, acceso a la topología NUMA. A diferencia de ESXi que ha evolucionado de manera más gradual desde su versión previa, Hyper-V Server 2012 R2 ha tenido una importante evolución reduciendo la brecha con el líder del mercado ESXi.

Hyper-V aísla las máquinas virtuales en particiones. Una partición es una unidad lógica que se utiliza para aislar, soportada por el hipervisor, en la cual cada sistema operativo huésped se ejecuta. El hipervisor tiene al menos una partición padre en donde

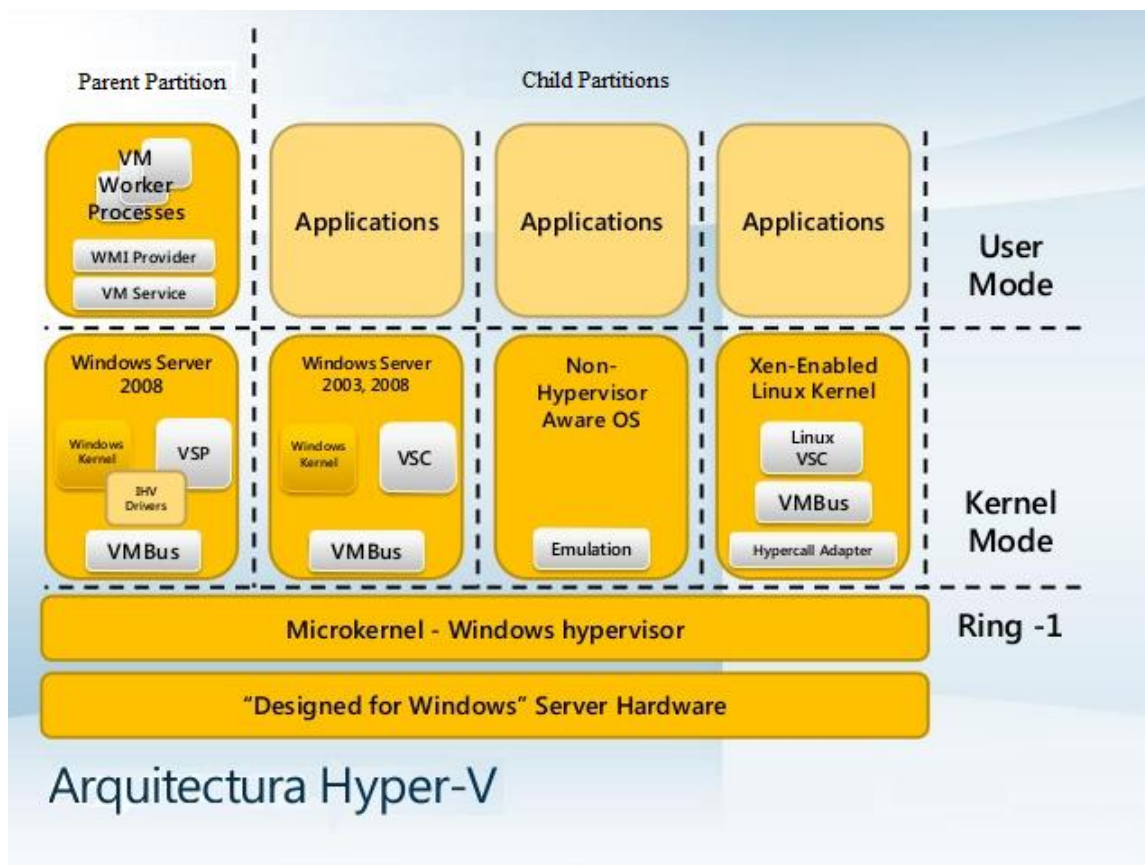


corre una versión soportada de Windows Server. La pila de virtualización corre en la partición padre y tiene acceso directo al hardware. La partición padre crea las particiones hijas, en donde corren los sistemas operativos huéspedes, usando la hypercall API expuesta por el hipervisor. [Wiki]

Las particiones hijas no tienen acceso directo a los recursos físicos. En cambio el hypervisor expone solo una parte de los recursos, dependiendo la configuración de la máquina virtual. Por ejemplo, el hypervisor expone a cada máquina virtual un subconjunto de procesadores físicos dependiendo de las configuraciones de las mismas.

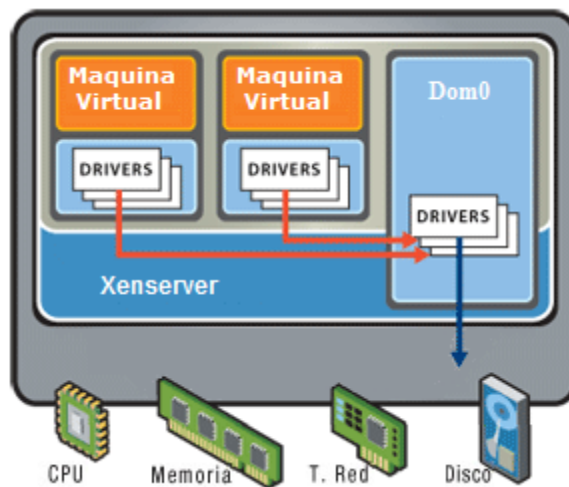
El hipervisor maneja las interrupciones para el procesador y las redirige a la partición respectiva usando el Controlador de Interrupciones Sintético (SynIC).

Las particiones tienen una vista virtual de los recursos físicos a través de dispositivos virtuales. Cualquier solicitud para los dispositivos virtuales es redirigida vía el VMBus a los dispositivos en la partición padre. El VMBus es un canal lógico que posibilita la comunicación entre particiones. La partición padre corre un Proveedor de Servicio Virtual (VSP), el cual se conecta al VMBus y maneja las solicitudes de las particiones hijas de acceso a dispositivos. Los dispositivos virtuales de las particiones hijas corren un Cliente de Servicio de virtualización (VSC), que redirige las solicitudes al VSP en la partición padre por el VMBus. Todo esto es transparente para el sistema operativo huésped. [Wiki]



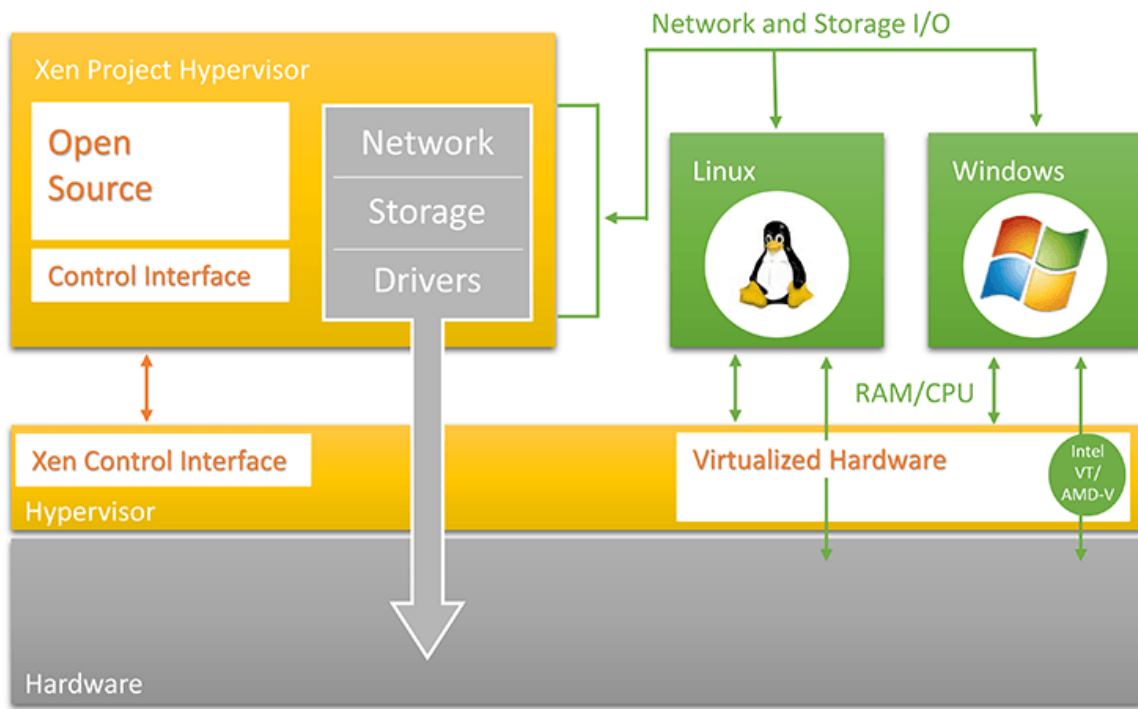
### 2.3. Citrix XenServer 6.5

Citrix XenServer 6.5 es una plataforma de virtualización open-source construida con el hipervisor Xen. Inicialmente estaba basada en para-virtualización, pero ahora también soporta virtualización asistida por hardware. De todas formas, aún mantiene un dominio de administración separado (Dom0) que controla las máquinas virtuales y negocia el acceso a los recursos.



En XenServer, el sistema operativo huésped es responsable de asignar la tabla de páginas físicas, pero solo tiene acceso de lectura y XenServer validar las actualizaciones de la tabla. Además los sistemas operativos huéspedes solo puede acceder memoria física no continua, ya que el hipervisor ocupa los primeros 64MB de cada espacio. [Che]

Dom0 es creado al momento de arranque y es responsable del control de creación, pausa, migración y eliminar los dominios huéspedes, parámetros del planificador de CPU y políticas de asignación de recursos.



Para virtualizar las operaciones de I/O, Xenserver propone modelo de anillo descriptor de buffer asíncrono y memoria compartida. En este modelo, dos aspectos deben ser considerados: transferir mensajes de I/O e información de I/O. Dos mecanismos de comunicación son provistos entre Xenserver y los sistemas operativos huéspedes: llamadas sincrónicas utilizando hiperllamadas para enviar mensajes desde los dominios huéspedes al dominio de administración, y eventos asíncronos usando interrupciones virtuales para enviar notificaciones desde el hipervisor al dominio huésped. Cuando la información es solicitada por el dominio huésped es llevada a memoria física, Dom0 enviará una interrupción virtual al dominio huésped correspondiente e intercambiará las páginas de memoria que contengan la información con un página libre presentada por el dominio huésped.



### **3. Metodologías de prueba existentes**

Al momento de seleccionar la metodología adecuada para comparar las plataformas descritas anteriormente se encontraron algunas alternativas en base a estudios realizados previamente.

- ❖ Basadas en benchmarks que comparan el impacto de los recursos adicionales requeridos para operaciones CPU, memoria y acceso a disco.
- ❖ Basadas en benchmarks que comparan el impacto de operaciones primitivas básicas y llevándolo a situaciones reales por predicción
- ❖ Basadas en benchmarks que comparan la performance generando carga similar a la encontrada en servidores reales.

Para la primera alternativa se revisó el trabajo "Performance comparison of Hypervisors" realizado por VMware en donde se comparan dos hipervisores, Xen y ESX. Para ello se realizaron los benchmarks:

- i. SPECcpu2000 en donde se simulan aplicaciones de uso intensivo de CPU.
- ii. Passmark que es conjunto de pruebas que representan pruebas orientadas a estaciones de trabajo.
- iii. Netperf que simula el uso de red de un datacenter.
- iv. SPECjbb2005 usada para representar aplicaciones Java usadas en un datacenter.
- v. SPECcpu2000 INT se utilizó para simular pruebas IT en un datacenter



El principal objetivo de este trabajo evaluar el desempeño de los hipervisores ESX y Xen. También se revisó el trabajo "Performance Evaluation of Hypervisors in the Private Cloud based on System Information using SIGAR Framework and for System Workloads using Passmark" de Vijaya Vardhan Reddy donde se utilizó el benchmark Passmark junto con el framework SIGAR para recolectar los datos. El objetivo de este trabajo fue medir la performance de hipervisores ESXi 4.1, XenServer 6.0 y KVM con Ubuntu 12.04

Para la segunda alternativa se analizó el trabajo "A Component-Based Performance Comparison of Four Hypervisors" por Jinho Hwang, Sai Zeng y Timothy Wood. Allí se realizaron benchmarks específico para cada componente: CPU (bytemark), Memoria (Ramspeed), red (Netperf) y discos (Bonnie++). En esta investigación se estudiaron los hipervisores Xenserver, Hyper-V, ESXi y KVM.

Por último, para la tercer alternativa donde se simulan cargas similares a las que se encuentran en servidores reales se analizó es trabajo "VMmark - A scalable Benchmark for Virtualized Systems" de Vikram Makhija y Bruce Herndon de Vmware del 2006 y su última versión "VMware VMmark® Benchmarking Guide" también de Vmware del 2014. El trabajo de Vmware presenta un método de benchmark basado en Tile, grupo de servidores con distintos roles que simular carga de escenarios reales. Cada rol está basado en un benchmark de un solo servidor. Este conjunto de cargas de trabajo se agrupan en una unidad de Trabajo llamada Tile. La performance de cada carga de trabajo se suma y se obtiene un puntaje total para el Tile. El puntaje total es calculado sumando los puntajes de cada Tile cuando se corren varios Tiles simultáneamente. [Makhija]

Tiles es la unidad de trabajo para un benchmark de consolidación de ambientes virtualizados y está definido como la colección de máquinas virtuales corriendo diversas cargas. El número total de Tiles da una medida de capacidad de consolidación de un servidor físico y la capa de virtualización. [Makhija]





Las siguientes son las cargas basadas en las cargas de un datacenter: Servidor de correo, Servidor Java, Servidor inactivo, Servidor Web, Servidor de base de datos y servidor de archivos. En lugar de desarrollar nuevas cargas, se utilizaron benchmarks existentes cuando fue posible para evitar redundancia el trabajo de implementación. Provee una base bien conocida sobre la cual se construye el benchmark, aunque se requirieron modificaciones para hacerlas apropiadas para un benchmark de múltiples máquinas virtuales dado que las reglas de ejecución de varios benchmark no estaban alineadas con los objetivos diseñados para Vmmark. [Makhija]

La metodología de puntuación utilizada en Vmmark es la descrita a continuación. Una vez que una prueba de Vmmark es completada, la performance cada de carga es reportada. Estas métricas son registradas en intervalos regulares durante la ejecución de la prueba. Una vez que todas las cargas terminan de correr se calcula el puntaje para cada Tile y finalmente se calcula el puntaje total en base a los puntajes de los Tiles.

#### **4. Metodología de prueba utilizada**

En base a las metodologías analizadas y la experiencia personal trabajando con virtualización se decidió utilizar la tercera alternativa generando carga similar a la encontrada en servidores reales y se comparan los resultados. El benchmark apropiado para la realización de la comparación de plataformas que en este trabajo se plantea es SPECvirt\_sc2013. Dado el elevado costo de este benchmark, inclusive con el descuento para instituciones educativas u\$s 750, se optó por otra alternativa. La modificación de un benchmark de uso gratuito de modo que se adapte a las necesidades de este trabajo. El benchmark seleccionado es Vmmark de VMware, el que fue diseñado para un ambiente multi-hipervisor, más específicamente un ambiente multi-hipervisor ESXi administrados por VMware Vcenter. Vmmark fue diseñado no solo para medir la performance de la plataforma con múltiples hipervisores ante la carga generada por uno o más Tiles, sino también para medir el impacto en la performance de la plataforma de operaciones de administración habituales como: la migración de una máquina virtual entre hipervisores,



la migración de los archivos de estas máquinas virtuales entre los contenedores del sistema de archivos y la creación de una máquina virtual a partir de una plantilla. También contempla la posibilidad de medir la energía consumida durante la ejecución de las pruebas por los equipos físicos participantes. Dado que estas capacidades exceden a las pruebas que este trabajo propone se modificarán Vmmark de modo que pueda correr con un solo hipervisor de cualquier fabricante y sin realizar operaciones propias de una plataforma multi-hipervisor. Para esto se modificarán las configuraciones, scripts y código embebido en XML. Además se simularán un Vcenter y dos ESXi de modo que Vmmark funcione.

Este benchmark está basado en cargas de trabajo relevante para un datacenter.

- Servidor de correo
- Sistema inactivo
- Simulación de carga para aplicación Web 2.0
- Simulación aplicación de comercio electrónico

Las cargas de trabajo que se usan en Vmmark son representativas de las aplicaciones más utilizadas. Las cuatro cargas utilizadas en cada Tile son:

Carga de Trabajo	Aplicación	Plataforma Máquina virtual
Servidor de Correo	Exchange 2007	Windows 2008 64-bit, 4 vCPU, 8GB RAM, 32GB arranque & 40GB disco datos
Sistema Inactivo	Ninguna	Windows 2003 32-bit, 1 vCPU, 512MB RAM, 4GB disco
Web 2.0	Olio	Back-end: SLES 11 64-bit, 2 vCPU, 2GB RAM, 10GB arranque & 4GB disco datos Front-end: SLES 11 64-bit, 4 vCPU, 6GB RAM, 10GB arranque & 70GB disco datos



Comercio Electrónico	DVD Store 2	Back-end: SLES 11 64-bit, 4 vCPU, 4GB RAM, 10GB arranque & 35GB disco datos Front-end (3): SLES 11 64-bit, 2 vCPU, 2GB RAM, 10GB disco
----------------------	-------------	---

Se correrán las pruebas utilizando 1,2 y 4 Tiles de forma de observar la respuesta de las plataformas con distintos niveles de sobre aprovisionamiento y como esto afecta las máquinas virtuales que corren en el hipervisor y sus respectivas aplicaciones. El procesador suele ser el recurso físico más valioso en un servidor y el más complicado de expandir.

Cantidad Tiles	Relación CPU Físico : CPU Virtual
1	1 : 1.625
2	1 : 3.541
4	1 : 4.5

## 5. Aplicaciones incluidas en las pruebas

### 5.1. Servidor de Correo

La mayoría de los negocios actuales otorgan a sus empleados de casillas de correo corporativos. Por esta razón, los servidores de correo generan mucha de la carga de los datacenters modernos, no solo por sus grandes requerimientos de recursos sino también por sus estrictos requerimientos de tiempo de respuesta. El servidor de correo más utilizado entre los clientes de Vmware es Microsoft Exchange. Por esta razón se utiliza Microsoft Exchange 2007 para representar la carga de un servidor de trabajo en Vmmark. [Vmware]

Microsoft provee una herramienta bien conocida para generar carga que simula usuarios de un servidor de correo Exchange llamada Loadgen. Para mantener la consistencia con el diseño de vmmark se realizaron algunos cambios en cómo se implemente esta herramienta. En su configuración por defecto Loadgen necesita



demasiado espacio para el propósito del benchmark por los que se redujo el tamaño inicial de las casillas de correo de los usuarios simulados. [Vmware]

Se limita la configuración a 1000 usuarios con casillas de correo de capacidad limitada en comparación con el tamaño por defecto. Esto permite que no haya cuellos de botella con ningún recurso en la máquina virtual Exchange. Vmmark está diseñado para registrar periódicamente métricas para cada carga de trabajo simulada. En vez de usar el número de usuarios como es habitual, se mide el número de transacciones ejecutadas por el servidor de correo. [Vmware]

La carga del servidor Microsoft Exchange 2007 corre en una máquina virtual de 32 bits con Microsoft Windows Server 2008 Enterprise Edition con cuatro CPU virtuales y 8 GB de memoria RAM.

## **5.2. Máquina Virtual Inactiva**

Muchos ambientes reales contienen sistemas con poca actividad o carga en forma de ráfagas. Estos sistemas con muy poca carga (esencialmente inactivos) son objetivos atractivos cuando se consolida servidores. Aun cuando están inactivos, estos sistemas aun requieren recursos de la capa de virtualización, lo que puede impactar la performance de las otras máquinas virtuales. Por esta razón, una máquina virtual inactiva es incluida en cada tile. Está máquina virtual no produce métricas que afecten el benchmark. Sin embargo se la requiere que periódicamente se reporte para considerar la prueba Vmmark válida. [Vmware]

La máquina virtual inactiva corre Microsoft Windows 2003 Enterprise Edition de 32 bits y tiene 1 CPU virtual y 512 MB de memoria RAM.



### **5.3. Aplicación Web 2.0**

Las aplicaciones web 2.0 tales como redes sociales se han vuelto ubicuas en los últimos años. Estas aplicaciones tienden a ser construidas sobre un amplio rango de herramientas open-source y aplicaciones distribuidas. La carga de trabajo simulada Olio (<http://incubator.apache.org/olio/>) imita una aplicación web 2.0 enfocada en redes sociales y eventos. Olio ha sido implementada en distintas tecnologías, incluidas PHP, Java y Ruby. Olio es un ejemplo de aplicación web 2.0 y soporta las siguientes operaciones habituales: Inicio, Loguearse, Buscar, Detalle de Evento, Detalles Personales, Agregar Persona y Agregar Evento. Vmmark incluye una implementación Java de nivel 2 de la carga de trabajo Olio. [Vmware]

Olio está configurada para simular 400 usuarios concurrentes. Periódicamente se mide la calidad de las operaciones realizadas. Se miden la cantidad de operaciones realizadas y los tiempos de respuesta de las mismas.

Olio utiliza en cada tile dos máquinas virtuales, que corren Novell SLES 11 de 64 bits. El front-end de la aplicación Server Java EE container con cuatro CPU virtuales y 6 GB de RAM. El back-end corre una base de datos MySQL en una máquina virtual con dos procesadores virtuales y 2 GB de memoria RAM.

### **5.4. Aplicación de Comercio Electrónico**

Las bases de datos transaccionales soportan una amplia cantidad de aplicaciones, normalmente parte de una arquitectura de varios niveles. Las bases de datos tienden a utilizar muchos recursos y a forzar a los la mayoría de los servidores donde corren y la componentes de la infraestructura. En muchos casos, estas bases de datos son sensibles a los tiempos de respuesta. En la mayoría de los casos el procesamiento de las transacciones es en ráfagas, por lo que la demanda de recursos varía en el tiempo. La



habilidad de la plataforma donde corre estos motores de soportar picos de uso de recursos es crítica para mantener una performance aceptable. [Vmware]

DVD Store Versión 2 (DS2) es una aplicación completa de prueba de comercio electrónico online con una aplicación web en un front-end, una base de datos en un back-end y un programa controlador que simula usuarios que realizan operaciones como consulta de inventarios o compras.

En cada tile DS2 utiliza cuatro máquinas virtuales, tres para servidores web y una para la base de datos. El primer servidor web genera carga de manera constante, los servidores web restantes lo hacen en forma de ráfagas demandando recursos de forma variada. Por cada servidor web, el programa controlar DS2 corre diez subprocesos que simulan usuarios. En estas pruebas se miden la cantidad de transacciones realizadas y los tiempos de respuestas de las mismas.

DS2 corren en cuatro máquinas virtuales que corren Novell SLES 11 de 64 bits. Los tres servidores web (DS2WebA, DS2WebB, y DS2WebC) corren apache 2.2 en máquinas virtuales de 2 CPU virtuales y 2 GB de memoria RAM. La cuarta máquina restante del tile corre una base de datos MySQL y utiliza cuatro procesadores virtuales y 4 GB de memoria RAM.

## **6. VMmark Harness**

Es una aplicación que corre en máquina virtual llamada Prime client y arranca y detiene las carga de trabajo en las máquinas de cada tile y recolecta los resultados de las pruebas.

Vmmark Harness está basada en el framework de automatización de pruebas STAF (<http://staf.sourceforge.net/index.php>) y su motor de ejecución STAX. Estas herramientas



soportan el desarrollo y ejecución de pruebas distribuidas y coordinadas entre sistemas heterogéneos. [Vmware]

The screenshot shows the STAX Job Monitor interface. The 'Active Job Elements' pane on the left lists various tasks such as 'InfraOps: Collect Events and Perf', 'Tile 0: Olioclient', and 'Tile 0: DS2WebA'. The 'Testcase Info' pane on the right displays a table with the following data:

Name	PASS: 2	FAIL: 0	Start Date-Time	Duration	Starts	Information
VMmark Initialization	1	0	20150723-22:28:24	00:00:38	1	
VMmark Setup	1	0	20150723-22:29:03	00:11:49	1	
VMmark Wkld Setup	0	0	20150723-22:40:52	<Pending>	1	
VMmark Run tiles	U	U	20150723-23:04:01	<Pending>	1	

The 'Messages' pane at the bottom shows a log of events, including 'VMmarkRMQmgr Clearing Queues and PS Sessions', 'Collecting EnvInfo', and 'VMmark2: Hosts In Cluster: [192.168.7.101, 192.168.7.102]'. It also contains error messages such as 'DS2WebA Abort: Tile 2: ds2webdrv Process Terminated Unexpectedly'.

Vmmark Harness consiste de muchos módulos STAX XML, el archivo de configuración VMMARK2.CONFIG y muchos archivos de configuración específicos para cada carga de trabajo. El módulo STAX principal, vmmark2\_main.xml procesa el archivo VMMARK2.CONFIG para configurar la prueba a correr. Cada carga de trabajo tiene su propio módulo que contiene código específico necesario para iniciar y correr la carga de trabajo y recolectar los resultados de esta. [Vmware]

El archivo de configuración VMMARK2.CONFIG define las pruebas que se correrán, identifica los clientes y máquinas virtuales involucradas en la prueba, el número de tiles y las cargas de trabajo para cada uno. Después de que la configuración ha sido procesada, la aplicación realizar un validación de los sistemas y sincronización previa y realiza la fase de configuración para cada carga de trabajo en cada tile. Una vez completada esta fase,



se inicial las cargas de trabajo en cada tile. Cuando estas finalizan, Vmware Harness vuelve a validar la sincronización y recolecta los resultados. [Vmware]

## 7. Cliente VMmark

Cada tile requiere un cliente que corre en una máquina virtual independiente.

Carga de Trabajo	Software Cliente
Servidor de Correo	LoadGen
Máquina Virtual Inactiva	IdleVMTest
Aplicación Web 2.0	Cliente Olio
Aplicación de Comercio Electrónico	Cliente DVD Store 2

Además de estas herramientas, todos los sistemas cliente deben correr la aplicación STAF, Java JDK, Cygwin y las herramientas de Administración de Microsoft Exchange. La máquina virtual prime client además debe correr la aplicación STAX, Powershell y VMware vSphere PowerCLI.

## 8. Máquinas Virtuales y Tiles

Cada tile está compuesto por 9 máquinas virtuales:

- Servidor correo: mailserver
- Máquina Virtual Inactiva: standby
- Aplicación Web 2.0: oliodb, olioweb
- Aplicación Comercio Electrónico: ds2db, ds2webA, ds2webB, ds2webC
- Cliente Vmmark: virtualclient



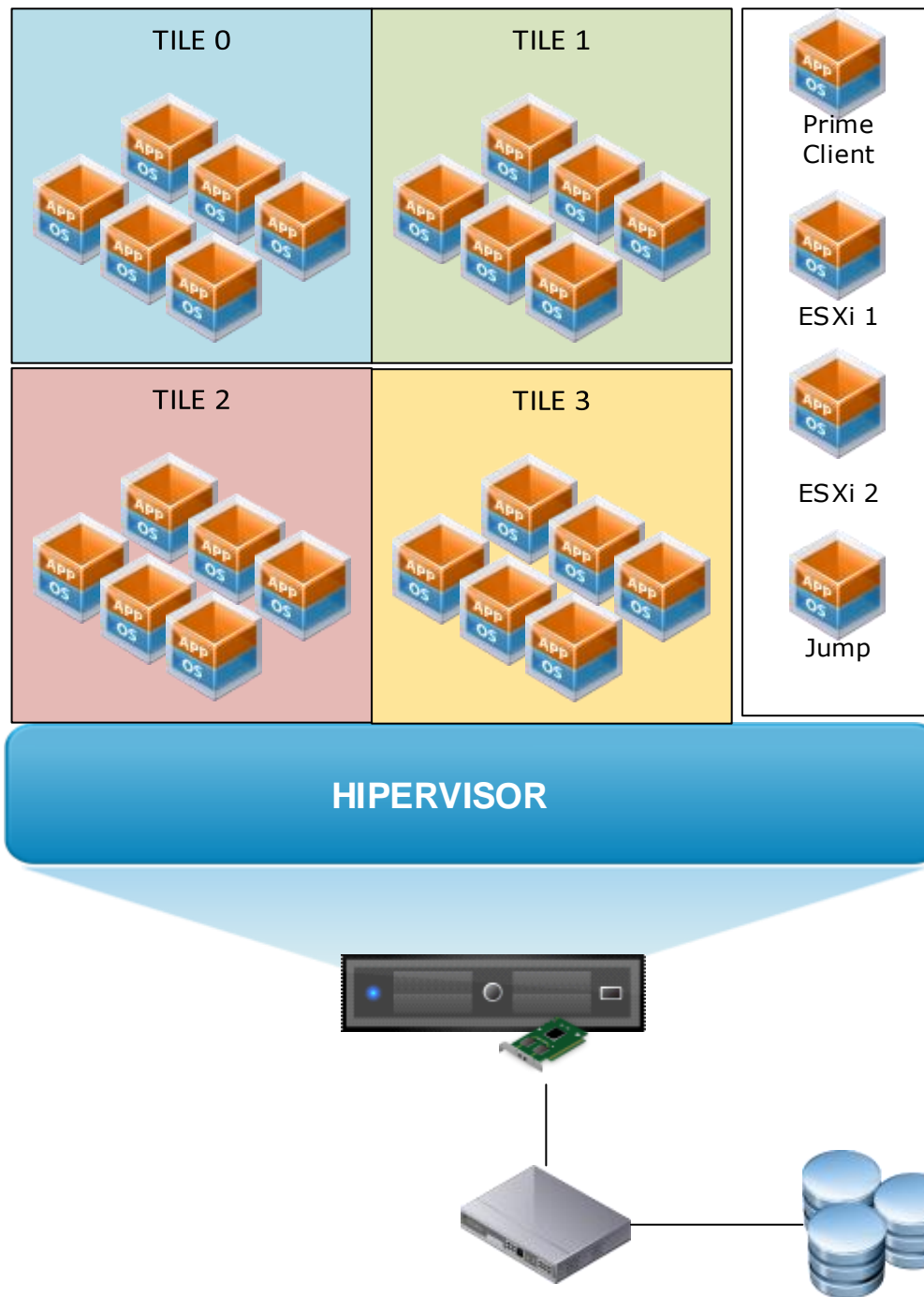


Además de las máquinas virtuales de los cuatro tiles utilizados, corren cuatro máquinas de infraestructura. Jump corre Windows 2008 Enterprise Edition y se utiliza para acceder por red a las máquinas virtuales dado que están conectadas a un red virtual interna, para correr Vmware Vcenter necesario para que Vmmark corra.

Se utilizaron dos máquinas más para simular los ESXi 6.0 conectados a la instancia de Vcenter mencionada anteriormente para poder correr la prueba. El cuarto servidor virtual llamado primeclient, en donde también corre Windows 2008 Enterprise Edition, es



donde se ejecuta la aplicación STAX que administra y coordina la ejecución de toda la prueba.





## 8.1. Configuración de Máquinas Virtuales

A continuación se detalla las configuraciones de las máquinas virtuales que se utilizaron en las pruebas que se realizaron.

Nombre	NumCpu	MemoryGB	MemoryMB	Storage (GB)	OS	IP
DS2DB0	4	4	4096	45	SLES 11	192.168.7.6/24
DS2DB1	4	4	4096	45	SLES 11	192.168.7.16/24
DS2DB2	4	4	4096	45	SLES 11	192.168.7.26/24
DS2DB3	4	4	4096	45	SLES 11	192.168.7.36/24
DS2WebA0	2	2	2048	10	SLES 11	192.168.7.7/24
DS2WebA1	2	2	2048	10	SLES 11	192.168.7.17/24
DS2WebA2	2	2	2048	10	SLES 11	192.168.7.27/24
DS2WebA3	2	2	2048	10	SLES 11	192.168.7.37/24
DS2WebB0	2	2	2048	10	SLES 11	192.168.7.8/24
DS2WebB1	2	2	2048	10	SLES 11	192.168.7.18/24
DS2WebB2	2	2	2048	10	SLES 11	192.168.7.28/24
DS2WebB3	2	2	2048	10	SLES 11	192.168.7.38/24
DS2WebC0	2	2	2048	10	SLES 11	192.168.7.9/24
DS2WebC1	2	2	2048	10	SLES 11	192.168.7.19/24
DS2WebC2	2	2	2048	10	SLES 11	192.168.7.29/24
DS2WebC3	2	2	2048	10	SLES 11	192.168.7.39/24
ESXi1	4	4	4096	5	ESXi 6.0	192.168.7.101/24
ESXi2	4	4	4096	5	ESXi 6.0	192.168.7.102/24
Jump	4	8	8192	80	Win 2008 Ent R2	192.168.7.100/24
mailserver0	4	8	8192	120	Win 2008 Ent R2	192.168.7.2/24
mailserver1	4	8	8192	120	Win 2008 Ent R2	192.168.7.12/24
mailserver2	4	8	8192	120	Win 2008 Ent R2	192.168.7.22/24
mailserver3	4	8	8192	120	Win 2008 Ent R2	192.168.7.32/24
OlioDB0	2	2	2048	14	SLES 11	192.168.7.4/24
OlioDB1	2	2	2048	14	SLES 11	192.168.7.14/24
OlioDB2	2	2	2048	14	SLES 11	192.168.7.24/24
OlioDB3	2	2	2048	14	SLES 11	192.168.7.34/24
OlioWeb0	4	6	6144	80	SLES 11	192.168.7.5/24
OlioWeb1	4	6	6144	80	SLES 11	192.168.7.15/24
OlioWeb2	4	6	6144	80	SLES 11	192.168.7.25/24
OlioWeb3	4	6	6144	80	SLES 11	192.168.7.35/24
primeclient	4	8	8192	80	Win 2008 Ent R2	192.168.7.210/24



standby0	1	0.5	512	5	Win 2003 Ent	192.168.7.3/24
standby1	1	0.5	512	5	Win 2003 Ent	192.168.7.13/24
standby2	1	0.5	512	5	Win 2003 Ent	192.168.7.23/24
standby3	1	0.5	512	5	Win 2003 Ent	192.168.7.33/24
virtualclient0	4	8	8192	80	Win 2008 Ent	192.168.7.1/24
virtualclient1	4	8	8192	80	Win 2008 Ent	192.168.7.11/24
virtualclient2	4	8	8192	80	Win 2008 Ent	192.168.7.21/24
virtualclient3	4	8	8192	80	Win 2008 Ent	192.168.7.31/24

## 9. Hardware Utilizado

Para la ejecución de las pruebas se utilizará un servidor IBM System x3850 que utiliza en grandes ambientes de producción para emular ambientes reales. También se utilizó almacenamiento SAN como el que se encuentran en los datacenter modernos. Se verifico la compatibilidad del hardware seleccionado con cada uno de los hipervisores a utilizar. A continuación se detalla el hardware utilizado.

System x3850 X5 7145	
Procesador	Intel(R) Xeon(R) X7542 2.67GHz L3 Cache 18 MB
	1066 MHz front-side bus (FSB)
	4 procesadores
	6 cores por socket
Memoria	256 GB (32 x 8 GB DDR III)
Almacenamiento	Local: 150 GB RAID 1
	SAN: 2 TB RAID 10
Red	6 x 1 GB nics
HBA	4 x 8 GB HBA



## **10. Despliegue de Sistemas**

### **10.1. Instalación Hipervisor**

#### **10.1.1. Instalación ESXi 6.0**

A continuación se nombran los pasos necesarios para instalar el hipervisor de Vmware en el servidor y el cliente de administración.

1. Descargar la imagen del instalador desde el sitio de Vmware
2. Bootear el servidor desde la imagen descargada
3. Instalar el Hipervisor en disco local
4. Reiniciar el sistema
5. Configurar el switch de administración
6. Configurar la interfaz vmkernel con la ip correspondiente
7. En sistema operativo que se use para administrar el hipervisor, acceder desde un navegador a la ip configurada anteriormente y descargar el cliente Vsphere
8. Instalar el cliente Vsphere

#### **10.1.2. Instalar XenServer 6.5**

Se siguieron los siguientes pasos para instalar Xenserver en el servidor.

1. Descargar la imagen del instalador desde el sitio de Citrix
2. Bootear el servidor desde la imagen descargada
3. Instalar el Hipervisor en disco local
4. Reiniciar el sistema
5. Configurar la interfaz de administración con la ip correspondiente
6. En sistema operativo que se use para administrar el hipervisor, acceder desde un navegador a la ip configurada anteriormente y descargar el XenCenter
7. Instalar XenCenter.



### **10.1.3. Instalar Hyper-V Server 2012 R2**

Finalmente se mencionan los pasos para instalar el hipervisor de Microsoft.

1. Descargar la imagen del instalador desde el sitio de Microsoft
2. Bootear el servidor desde la imagen descargada
3. Instalar el Hipervisor en disco local
4. Reiniciar el sistema
5. Configurar la interfaz de red con la ip correspondiente
6. En sistema operativo que se use para administrar el hipervisor, instalar las herramientas administrativas del rol Hyper-V

### **10.2. Despliegue del Tile**

Como se mencionó anteriormente cada Tile cuenta con nueve máquinas virtuales, de las cuales tres corren Windows Server 2003 o 2008. Las máquinas restantes corren SLES 11, Vmmark provee platillas para la creación de estos servidores. Para facilitar el trabajo de despliegue de los tiles, se iniciaran las pruebas sobre la plataforma de Vmware. De este modo se podrá aprovechar las platillas mencionadas para la creación de las máquinas correspondientes. Una vez completadas las pruebas sobre la plataforma ESXi, se sacara provecho de una característica muy importante de la virtualización, la portabilidad de las máquinas virtuales. Por lo que se realizará un proceso de migración conocido como V2V (virtual a virtual) desde ESXi a Hyper-V y finalmente de este último a XenServer. Esto nos permitirá facilitar el trabajo y ahorrar bastante tiempo de implementación. A continuación se resumirá el proceso de creación de las máquinas virtuales, el proceso detallado puede ser consultado en ["VMware VMmark® Benchmarking Guide 2.5.2"](#).



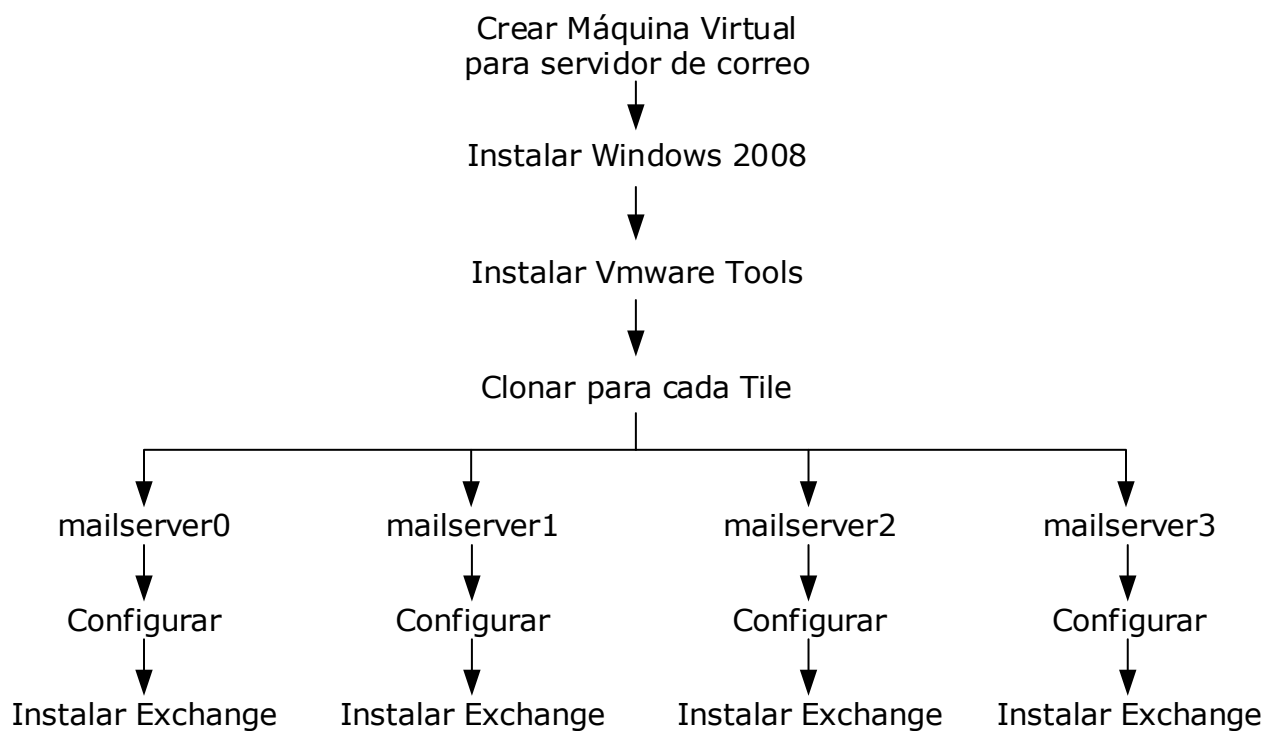
### **10.2.1. Despliegue del Servidor de Correo**

En esta sección describe la creación de la máquina virtual para servidor de correo. La primera de este tipo se debe crear desde cero, las subsiguientes serán clonadas de la primera.

1. Crear la máquina virtual con las configuraciones para el servidor de correo descrita en [Configuración de Máquinas Virtuales](#).
2. Instalar Microsoft Windows Server 2008 R2 Enterprise Edition.
3. Instalar Vmware Tools.
4. Instalar actualizaciones para Microsoft Windows Server 2008.
5. Configurar STAF para que arranque automáticamente.
6. Crear los volúmenes para backup y la base de datos de Exchange.
7. Instalar y configurar la aplicación STAF (la cual es utilizada para que vmmark pueda ejecutar comando remotamente en los servidores de cada Tile).
8. Desactivar el firewall de Windows.
9. Modificar el archivo host con las correspondientes ips detalladas en [Configuración de Máquinas Virtuales](#).
10. Instalar Powershell.
11. Instalar el rol Internet Information Services.
12. Configurar Internet Explorer para que no utilice ningún proxy.
13. Clonar la máquina virtual tantas veces como Tiles se utilicen.
14. Renombrar el servidor



- 15. Configurar la interfaz de red con la ip detallada en [Configuración de Máquinas Virtuales](#).
- 16. Promover el servidor a controlador de dominio creando un dominio nuevo.
- 17. Instalar Microsoft Exchange Server 2007.
- 18. Configurar Microsoft Exchange Server.
- 19. Actualizar Microsoft Exchange Server.
- 20. Realizar un backup de la base de datos de Exchange.
- 21. Incrementar el tamaño del cache de la base de datos de transporte.



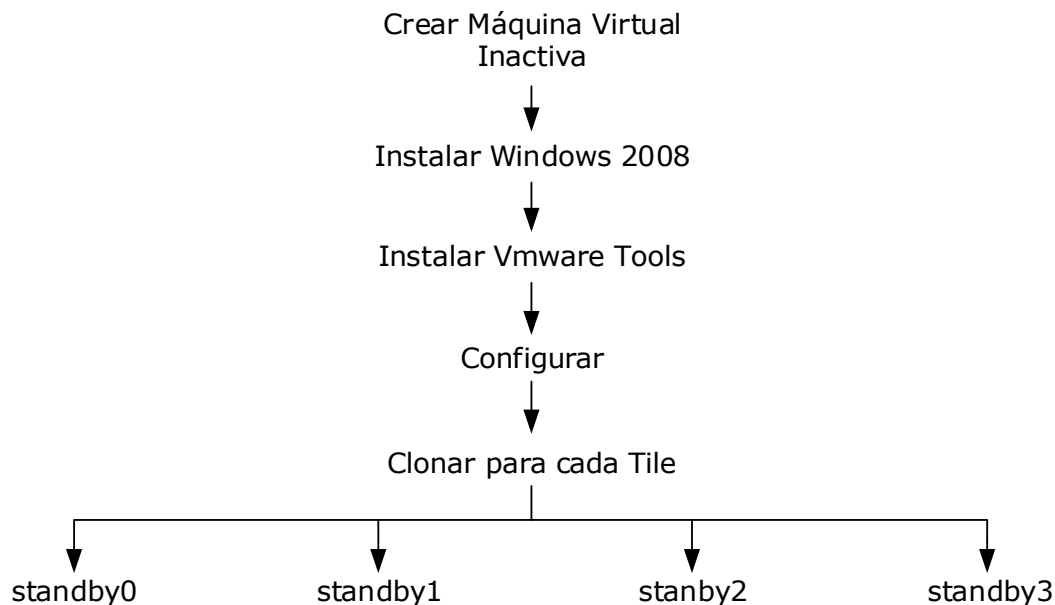




### **10.2.2. Despliegue de la Máquina Virtual Inactiva**

En esta sección se describe la creación de la máquina virtual inactiva del Tile. La primera de deberá crear desde cero y las restantes se pueden clonar de la primera.

1. Crear la máquina virtual con las configuraciones para el servidor de correo descrita en [Configuración de Máquinas Virtuales](#).
2. Instalar Microsoft Windows Server 2003 Enterprise Edition SP2.
3. Instalar Vmware Tools.
4. Instalar actualizaciones para Microsoft Windows Server 2003.
5. Instalar y configurar la aplicación STAF.
6. Desactivar el firewall de Windows.
7. Modificar el archivo host con las correspondientes ips detalladas en [Configuración de Máquinas Virtuales](#).
8. Configurar STAF para que arranque automáticamente
9. Clonar la máquina virtual tantas veces como Tiles se utilicen.
10. Renombrar el servidor.
11. Configurar la interfaz de red con la ip detallada en [Configuración de Máquinas Virtuales](#).



### **10.2.3. Despliegue de la Aplicación Web 2.0**

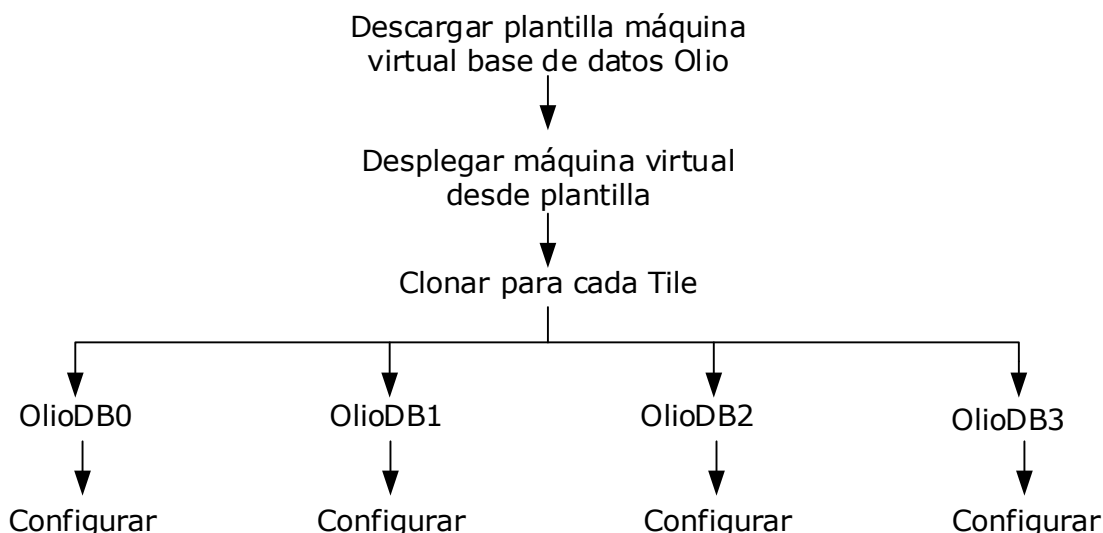
#### **10.2.3.1. Despliegue del Backend para la aplicación Web 2.0**

En esta sección se describe como se despliega la base de datos Olio utilizando la plantilla disponible junto a Vmmark.

1. Descargar la plantilla Base de Datos Olio desde el sitio de Vmware.
2. Desplegar la máquina virtual desde la plantilla descargada.
3. Configurar la aplicación STAF.
4. Clonar la máquina virtual tantas veces como Tiles se utilicen.
5. Renombrar el servidor.
6. Configurar la interfaz de red con la ip detallada en [Configuración de Máquinas Virtuales](#)



7. Modificar el archivo hosts con las correspondientes ips detalladas en [Configuración de Máquinas Virtuales](#).



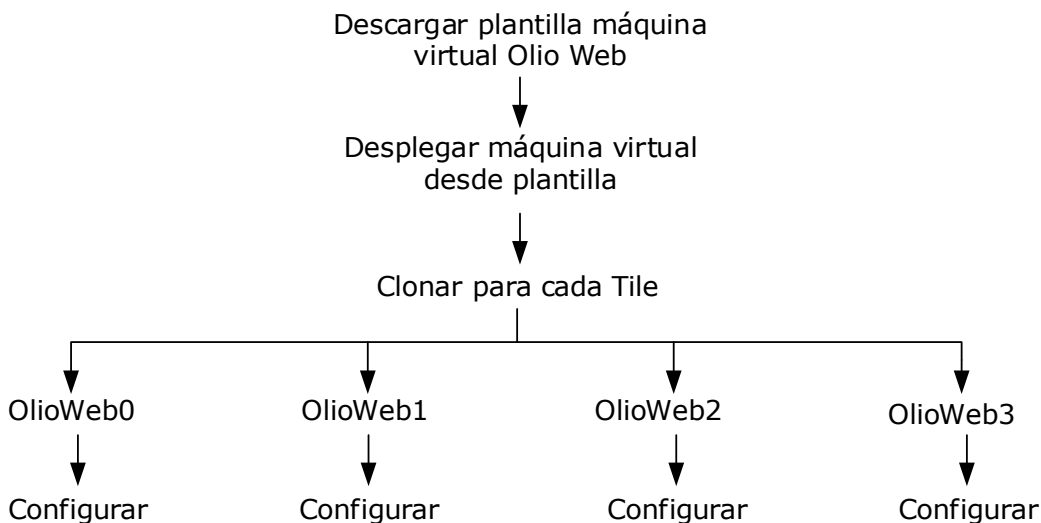
### **10.2.3.2. Despliegue del Frontend para la aplicación Web 2.0**

En esta sección se describe como se despliega la máquina virtual donde corre la Web de Olio utilizando la plantilla disponible junto a Vmmark.

1. Descargar la plantilla Olio Web desde el sitio de Vmware.
2. Desplegar la máquina virtual desde la plantilla descargada.
3. Configurar la aplicación STAF.
4. Clonar la máquina virtual tantas veces como Tiles se utilicen.
5. Renombrar el servidor.
6. Configurar la interfaz de red con la ip detallada en [Configuración de Máquinas Virtuales](#)



7. Modificar el archivo hosts con las correspondientes ips detalladas en [Configuración de Máquinas Virtuales](#).



## 10.2.4. Despliegue de la Aplicación de Comercio Electrónico

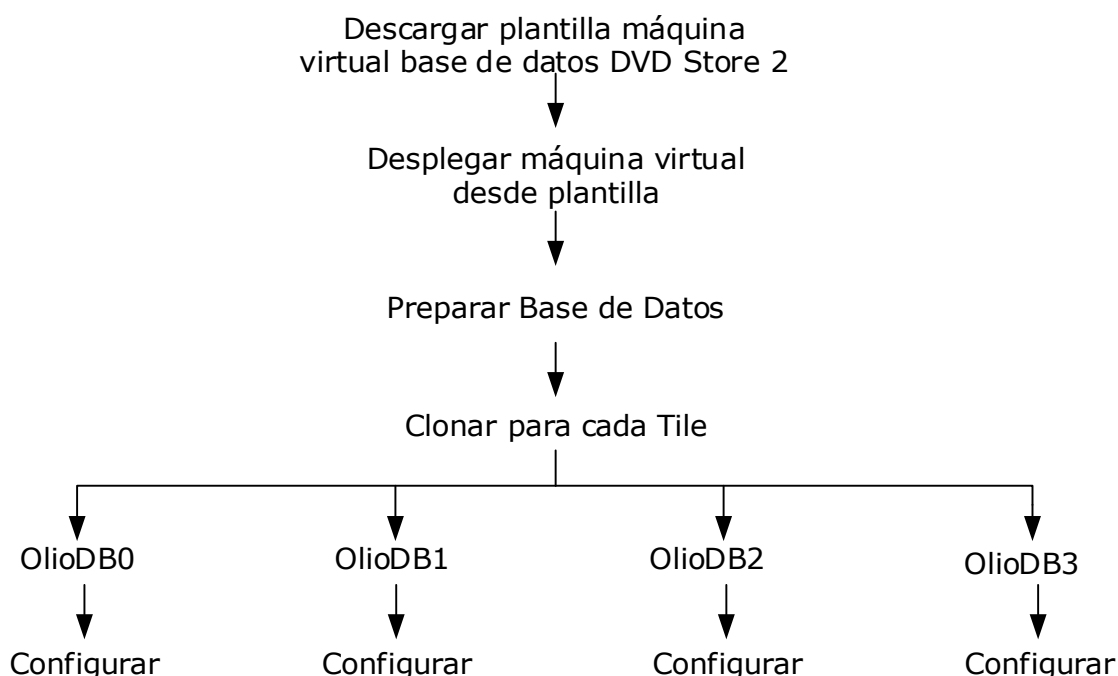
### 10.2.4.1. Despliegue del Backend para la aplicación de Comercio Electrónico

En esta sección se describe como se despliega la base de datos DVD Store 2 utilizando la plantilla disponible junto a Vmmark.

1. Descargar la plantilla Base de Datos DVD Store 2 desde el sitio de VMware.
2. Desplegar la máquina virtual desde la plantilla descargada.
3. Configurar la aplicación STAF.
4. Preparar base de datos DVD Store 2.
5. Clonar la máquina virtual tantas veces como Tiles se utilicen.



6. Renombrar el servidor.
7. Configurar la interfaz de red con la ip detallada en [Configuración de Máquinas Virtuales](#)
8. Modificar el archivo hosts con las correspondientes ips detalladas en [Configuración de Máquinas Virtuales](#).



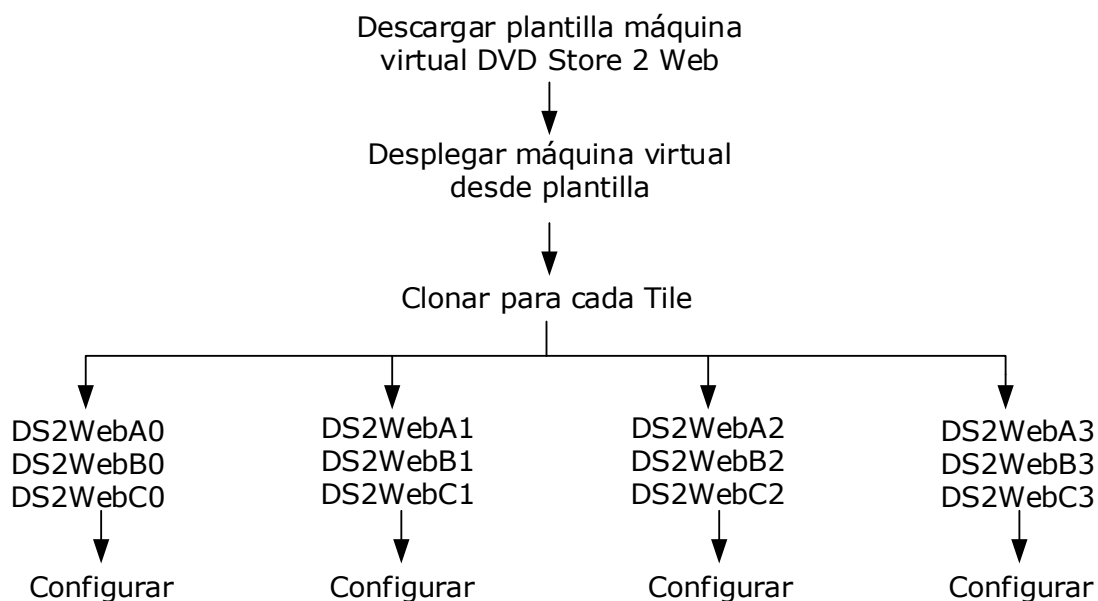
#### **10.2.4.2. Despliegue del Front-end para la aplicación de Comercio Electrónico**

En esta sección se describe como se despliega la máquina virtual donde corre la Web de DVD Store 2 utilizando la plantilla disponible junto a Vmmark.

1. Descargar la plantilla DVD Store 2 Web desde el sitio de Vmware.
2. Desplegar la máquina virtual desde la plantilla descargada.



3. Configurar la aplicación STAF.
4. Clonar la máquina virtual tantas veces como Tiles se utilicen.
5. Renombrar el servidor.
6. Configurar la interfaz de red con la ip detallada en [Configuración de Máquinas Virtuales](#)
7. Modificar el archivo hosts con las correspondientes ips detalladas en [Configuración de Máquinas Virtuales](#).



### 10.2.5. Despliegue Clientes para los Tiles

En esta sección describe la creación de las máquinas virtuales clientes.

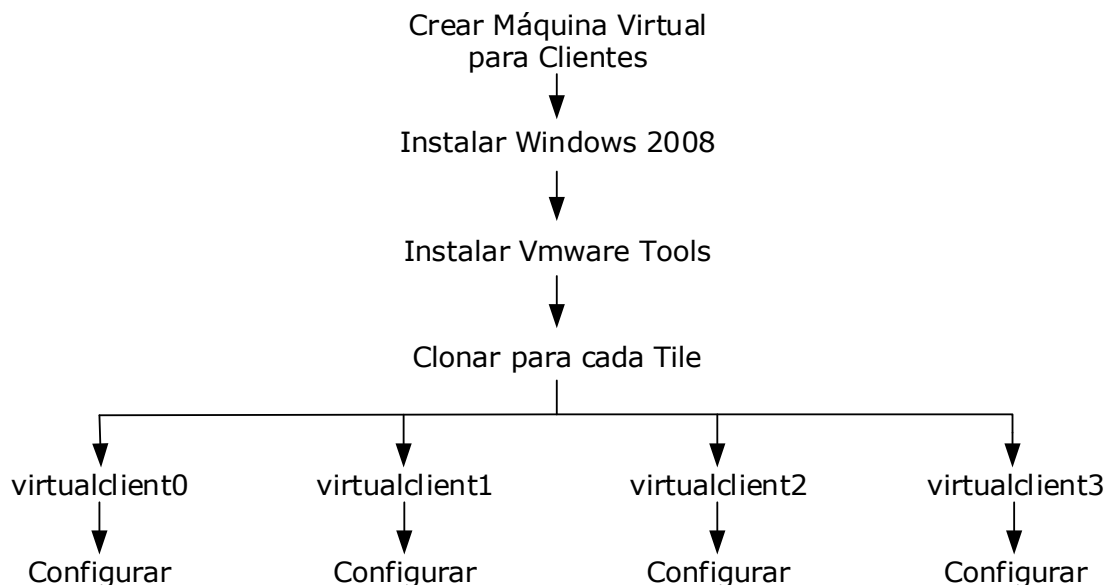
1. Crear la máquina virtual con las configuraciones para el cliente descrita en [Configuración de Máquinas Virtuales](#).



2. Instalar Microsoft Windows Server 2008 Enterprise Edition (esta versión no debe ser R2 por problemas de compatibilidad con una herramienta de Vmmark).
3. Instalar VMware Tools.
4. Instalar Java JRE.
5. Instalar actualizaciones para Microsoft Windows Server 2008.
6. Configurar STAF para que arranque automáticamente.
7. Instalar y configurar la aplicación STAF.
8. Desactivar el firewall de Windows.
9. Deshabilitar la característica "Prevención de Ejecución de Datos" (DEP).
10. Modificar el archivo host con las correspondientes ips detalladas en [Configuración de Máquinas Virtuales](#).
11. Instalar Powershell.
12. Instalar PowerCLI.
13. Instalar el servicio de eventos STAF.
14. Instalar el rol Internet Information Services.
15. Configurar Internet Explorer para que no utilice ningún proxy.
16. Instalar y configurar Cygwin.
17. Realizar modificaciones necesarias en el registro de Windows para evitar problemas con el uso de puertos.
18. Instalar VMmark Harness.
19. Clonar la máquina virtual tantas veces como Tiles se utilicen.
20. Renombrar el servidor



21. Configurar la interfaz de red con la ip detallada en [Configuración de Máquinas Virtuales](#).
22. Unir el servidor al dominio del Tile correspondiente.
23. Instalar las Herramientas para Microsoft Exchange Server 2007.
24. Instalar LoadGen.



### 10.2.6. Despliegue del Primeclient

En esta sección describe la creación de la máquina virtual Primeclient que corre Vmmark

1. Crear la máquina virtual con las configuraciones para el cliente descrita en [Configuración de Máquinas Virtuales](#).
2. Instalar Microsoft Windows Server 2008 Enterprise Edition (esta versión no debe ser R2 por problemas de compatibilidad con una herramienta de Vmmark).





3. Instalar Vmware Tools.
4. Instalar Java JRE.
5. Instalar actualizaciones para Microsoft Windows Server 2008.
6. Configurar la interfaz de red con la ip detallada en [Configuración de Máquinas Virtuales](#).
7. Configurar STAF para que arranque automáticamente.
8. Instalar y configurar la aplicación STAF.
9. Desactivar el firewall de Windows.
10. Deshabilitar la característica "Prevención de Ejecución de Datos" (DEP).
11. Modificar el archivo host con las correspondientes ips detalladas en [Configuración de Máquinas Virtuales](#).
12. Instalar Powershell.
13. Instalar PowerCLI.
14. Instalar el servicio de eventos STAF.
15. Instalar el rol Internet Information Services.
16. Configurar Internet Explorer para que no utilice ningún proxy.
17. Instalar y configurar Cygwin.
18. Realizar modificaciones necesarias en el registro de Windows para evitar problemas con el uso de puertos.
19. Instalar STAX.
20. Configurar servicio VMmark2 PowerCLI.
21. Instalar VMmark Harness.
22. Configurar VMmark Harness.



A continuación se muestra el archivo de configuración VMMARK2.CONFIG de Vmmark utilizando cuatro Tiles y se muestran las modificaciones realizadas.

```
#           Example VMmark Config File
# Copyright © 2006 - 2014 VMware, Inc. All rights reserved.
# -----
# VMmark Configuration Settings
# -----
# Number of Tiles to Run
TILES=4
# Enable Turbo Mode Timing and Scoring
# This will mark any run non-compliant and change the RUNTIME=1800
# Both PTD on and off are allowed.
# Score File = Score_Turbo_<NumTiles>_Tile_Test.out.
# Valid values [0/1] Disable/Enable
#TurboMode=1
# -----
# Power Configuration Settings
# PTD=0/1: Valid values [0/1] Disable/Enable
# PTD_Clients: STAF-enabled systems connected to power meters
#     Note: The number of PTD_Clients determines the total number of PTD
#           : processes activated. 2 clients would require 2 ports, etc.
# PTD_Target: Classification of metered items (SERVER or EXT_STOR [storage])
# PTD_Ports: Port used for each power meter.
# PTD_MeterTypes: Meter Type for each power meter.
# PTD_COMs: COM port for each power meter on its PTD client.
#     Note: When GPIB is enabled, via PTD_Enable_GPIB="1", PTD_COMs is used to
#           specify the GPIB address #.
#     Ex: PTD_COMs="8"
# PTD_Amp_Ranges: Allows defining of different Ampere ranges per meter.
```



```
# PTD_Volt_Ranges: Allows defining of different Volt ranges per meter.
# PTD_Enable_GPIB: Allows the use of GPIB devices, set to 1 to enable
#   Ex: PTD_Client:PTDclient1 -> Target:SERVER -> Port:8888 -> MeterType:8
(Yokogawa) -> COM:COM1 -> Range:4.0
#   Ex: PTD_Client:PTDclient2 -> Target:SERVER -> Port:8890 -> MeterType:8
(Yokogawa) -> COM:COM1 -> Range:4.0
# -----
```

### **PTD=0**

```
PTD_Clients="localhost"
PTD_Target="SERVER SERVER EXT_STOR EXT_STOR"
PTD_Ports="8888 8890 8892 8894"
PTD_MeterTypes="8 8 8 0"
PTD_COMs="COM1 COM1 COM2 COM4"
PTD_Amp_Ranges="5.0 5.0 5.0 5.0"
PTD_Volt_Ranges="120.0 120.0 120.0 120.0"
PTD_Enable_GPIB="0 0 0 0"
# - Optional Settings - defaults shown
# - Warning: Changing values may make resulting runs non-compliant or cause issues.
###PTD_RampTime=1800
###PTD_Local_Hostnames=""
###PTD_Local_Ports
# -----
# Growler Configuration Settings
# Note: Growler is a messaging system that can display info such as
# Errors, Warnings, and VMmark2 results to a Growl-configured
# system (Win/Mac/Linux), see: http://Growl.info for download/info.
# Additionally, these messages can be forwarded to smart phones
# via applications like prowl(iOS) or NotifyMyAndroid.
# Growler=0/1: Valid values [0/1] Disable/Enable
```



```
# GrowlerHost: System to notify
# GrowlerPassword: Password for GrowlerHost
# GrowlerSticky= 0/1: Disable/Enable option to force message confirm.
# GrowlerVerbose= 0/1: Disable/Enable additional error messages.
# -----
```

#### **Growler=0**

```
GrowlerHost="localhost"
GrowlerPassword=""
GrowlerSticky=0
GrowlerVerbose=1
# -----
# Infrastructure Configuration Settings
# -----
```

**VCServerHOSTNAME="vcserver0"**

**VCServerUSER="Administrator"**

**VCServerPASSWORD="Jsunday15"**

**VCServerCLUSTER="vmmark"**

**RMQServer="localhost"**

```
Deploy/Templates="StandbyTemplate1"
Deploy/OScustomizations="StandbyCust StandbyCust2"
Deploy/DeployLUNs="CX3-80-Lun24 CX3-80-Lun23"
Deploy/DeployVMs="DeployVM1 DeployVM2"
# - Optional Settings - defaults shown
# - Warning: Changing values may make resulting runs non-compliant
# Deploy/DELAYTIME="170"
# Deploy/UpgradeTools="1"
# Deploy/BurstQueueSize="auto"
# Deploy/DeployDelayBetween="300"
# - Optional Settings - defaults shown
# - Warning: Changing values may make resulting runs non-compliant
```



```
# Vmotion/Workloads="OlioDB"

# Vmotion/VMotionDelayBetween="180"
# Vmotion/BurstQueueSize="auto"
# Vmotion/DELAYTIME="150"
SVmotion/TargetLUNs="CX3-80-Lun21 CX3-80-Lun21"
# - Optional Settings - defaults shown
# - Warning: Changing values may make resulting runs non-compliant
# SVmotion/Workloads="Standby"
# SVmotion/SleepBetween="300"
# SVmotion/BurstQueueSize="auto"
# SVmotion/DELAYTIME="160"
# - Note: Vmotion and SVmotion infrastructure operations can not operate on the same
workload(s)
# - at the same time.
# -----
# Workload Configuration Settings
# -----
MailServer/MailPassword="Jsunday15"
MailServer/MailDomains="vmmark2-mail0 vmmark2-mail1 vmmark2-mail2
vmmark2-mail3"
MailServer/MailQualifier="eng.vmware.com"
# - Note: MailQualifier must be 3 parts : For example "eng.vmware.com"
#
# - Optional Settings - defaults shown
# - Warning: Changing values may make resulting runs non-compliant
# MailServer/SKIPRESTORE="0"
# MailServer/DELAYTIME="12"
# - Optional Settings - defaults shown
# - Warning: Changing values may make resulting runs non-compliant
# OlioWeb/SKIPRESTORE="0"
# OlioWeb/DELAYTIME="5"
```



```
# OlioWeb/RAMPUP="300"
# OlioWeb/RAMPDOWN="300"
# OlioWeb/SCALE="400"
# OlioWeb/FINALIZE="10"
# OlioWeb/OLIORAIN="1"
# OlioWeb/FinalizeGlassFish="0"
#
# OlioDB/SKIPRESTORE="0"
# OlioDB/SCALE="400"
# - Optional Settings - defaults shown
# - Warning: Changing values may make resulting runs non-compliant
# DS2WebA/SKIPRESTORE="0"
# DS2WebA/DELAYTIME="8"
#
# DS2WebB/SKIPRESTORE="0"
# DS2WebB/DELAYTIME="10"
# DS2WebB/IterationTimeSlice="240"
# DS2WebB/DelayTimeSlice="60"
# DS2WebC/SKIPRESTORE="0"
# DS2WebC/DELAYTIME="15"
# DS2WebC/IterationTimeSlice="180"
# DS2WebC/DelayTimeSlice="120"
# DS2DB/SKIPRESTORE="0"
# - Optional Settings - defaults shown
# - Warning: Changing values may make resulting runs non-compliant
# Standby/DELAYTIME="2"
# -----
# Testbed and Environment Configuration Settings
# -----
# Set up the list of client systems - must have
```



# at least as many clients as tiles.

**CLIENTS="virtualclient0 virtualclient1 virtualclient2 virtualclient3"**

# Workload-specific list of Server VM Hostnames

**MailServer/SERVERS="Mailserver0 Mailserver1 Mailserver2 Mailserver3"**

**Standby/SERVERS="Standby0 Standby1 Standby2 Standby3"**

**OlioWeb/SERVERS="OlioWeb0 OlioWeb1 OlioWeb2 OlioWeb3"**

**OlioDB/SERVERS="OlioDB0 OlioDB1 OlioDB2 OlioDB3"**

**DS2WebA/SERVERS="DS2WebA0 DS2WebA1 DS2WebA2 DS2WebA3"**

**DS2WebB/SERVERS="DS2WebB0 DS2WebB1 DS2WebB2 DS2WebB3"**

**DS2WebC/SERVERS="DS2WebC0 DS2WebC1 DS2WebC2 DS2WebC3"**

**DS2DB/SERVERS="DS2DB0 DS2DB1 DS2DB2 DS2DB3"**

# - Note: For best results follow the above naming convention

# Reporter Settings

**REPORTER=1**

REPORTERDIR="/cygdrive/c/vmmark2/tools/"

ScriptName="VMmark-ReporterMaster.sh"

#

# Error handling

# This will determine how the harness deals with errors during the phases

# before running. By setting ERRORIMMEDIATE to 1, the harness will exit

# upon finding the first error (as opposed to continuing through all the

# remaining work for that phase).

**ERRORIMMEDIATE=0**

#####

#####

# Standard Defaults

# All parameters below this line have the standard default

# values listed. Normally these values do not require

# changing. For the path values, we recommend that you

# configure the testbed using the standard path names as

# this helps facilitate debugging. There are several numeric



```
# values that have occasionally been shown to be useful for
# benchmarkers, but care should be taken when modifying these
# values. Uncommenting these parameters will be denoted as
# "Overriding Required Default". Changing key parameters
# such as RUNTIME will cause the test run to be non-compliant.
#####
#####
# Set the RUNTIME for the test. The Default for a Fully Compliant
# test is 10800 secs (3 hours). Changing this value can make your
# test non-compliant
###RUNTIME=10800
RUNTIME=3600
# TILEDELAY=seconds
# Delays the start of tile N by TILEDELAY * N (secs) and also
# increments the runtime of the workloads by TILEDELAY * (#TILES - N - 1)
# to compensate for the delay. No delay if set to 0.
###TILEDELAY=60
# FLAG to control display/logging of Info/Debug messages
# when debugSTAXUtilLogAndMsg is called
# if == 0: No messages are displayed/logged
# if != 0: Messages are displayed and logged
###DEBUGFLAG=0
# This is a list of the workloads to be invoked
# this is to allow a nonstandard list of workloads
# to be run with this harness as long as the template
# workload functions are in the vmmark2/xml directory.
# For a fully compliant test all 6 workloads must be
# used (default)
###WORKLOADLIST="Standby MailServer OlioWeb OlioDB DS2WebA DS2WebB
DS2WebC DS2DB"
```





```
# This is a list of the infrastructure operations to be invoked
# this is to allow a nonstandard list of infrastructure operations
# to be run with this harness as long as the template
# workload functions are in the vmmark2/xml directory.
# For a fully compliant test all 3 infrastructure operations must be
# used (default)
###INFRASTRUCTURELIST="Deploy Vmotion SVmotion"
INFRASTRUCTURELIST=""
# Flag to disable the Cluster VM Mapping routines
###DoClusterMapping=1
DoClusterMapping=1
# RMQ Settings
# Note: Changing this setting can have an adverse impact on your environment, use only
as needed.
# RMQ_MinInitDelay is the delay time allotted for workers to start
# RMQ_WorkerDelay is the delay time between worker spawning.
# RMQ_DisbleQueueRestart disables the automatic clearing of RMQ queues before each
run
# RMQ_PowerCLI_Delay is the delay time in seconds allotted for PowerCLI to Load and/or
write status
# messages, determining the state of the PowerCLI Service starting. Should be greater
than zero.
RMQ_MinInitDelay=60
RMQ_WorkerDelay=2
###RMQ_DisbleQueueRestart=0
RMQ_PowerCLI_Delay=30
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# Note: Please observe the special formatting used with
# some of the non-numeric character strings in various
# parameter settings. Quotes, double backslashes, %u,
```



```
# and use of path names of the form "C:\Progra~1\..."
# are required where used in the sample VMMARK2.CONFIG
# files to avoid character stripping by the shells or
# to avoid the use of path names with spaces, which are
# not supported in this file.
#!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# This directory is the location of the
# Harness code
###IMPORTDIR=c:\vmmark2\xml

# This is the location of any config files
# referenced by the harness code.
###CONFIGDIR=c:\vmmark2
# This is a tmp directory for scratch files
# the harness creates
###TMPDIR=c:\vmmark2\tmp
# This is the path on the VMmark clients that contains
# the individual workload directories.

# Note: The extra backslash is to escape the backslash-v
# that was being treated as a control char when passed
# to windows batch files.
###CLIENTROOTDIR=C:\\vclient
# Parallelize Workload Setup
# Set to 0 to disable
###PARALLELSETUP=1
PARALLELSETUP=1
# FLAG to control cleaning up results files after they've been
# copied into the vmmark2/results/Results_<datestamp> directory.
# If set to 0, cleanup of files will be skipped.
```



```
###CLEANUPFLAG=1

# Post Process Results
# Disabled by default except for full runs where:
# RUNTIME=10800 and WORKLOADS="<list of all 6>"
# Note: %u will be #Tiles in run
# The output is sent to: vmmark2/results/Results_<timestamp>/Score_#_tile.out
# The working directory is vmmark2/results/Results_<timestamp>
# To post process partial runs, set flag to 1 and modify the post processing
# command as needed.
#
###POSTPROCESSFLAG=0
###POSTPROCESSCMD="C:\\cygwin\\bin\\perl -f C:\\vmmark2\\tilescore.pl -t %u"
# Miscellaneous Parameters
# CancelTestBlockSize allows the throttling of the number of simultaneous tiles processed
by
# the canceltest application. Valid ranges are 1-500, default is 50.
##CancelTestBlockSize="50"
#
# ClusterMappingTimeout allows the modification of the expected amount of time (in
seconds) the harness
# will wait for the ClusterMapping routines to complete prior to aborting/terminating. Any
non-zero value
# in seconds is allowed. Default is 3600 seconds ("3600")
# Example : ClusterMappingTimeout="5400"
#ClusterMappingTimeout="3600"
#
# ScrubConfigFile (0/1) : If set to 1 will enable the scrubbing of all passwords from the
copy
# of the VMMARK2.config file in the current run's results folder
##ScrubConfigFile="0"
#
```



#01292014

## 11. Ejecución de las pruebas

Como se mencionó anteriormente se corrió Vmmark primero en el hipervisor de VMware, ESXi

**mlicesxcli06 - vSphere Client**

File Edit View Inventory Administration Plug-ins Help

Home Inventory Inventory

mlicesxcli06

- DS2DB0
- DS2DB1
- DS2DB2
- DS2DB3
- DS2WebA0
- DS2WebA1
- DS2WebA2
- DS2WebA3
- DS2WebB0
- DS2WebB1
- DS2WebB2
- DS2WebB3
- DS2WebC0
- DS2WebC1
- DS2WebC2
- DS2WebC3
- ESXi1
- ESXi2
- Jump
- mailserver0
- mailserver1
- mailserver2
- mailserver3
- oliodb0
- oliodb1
- oliodb2
- oliodb3
- OlioWeb0
- OlioWeb1
- OlioWeb2
- OlioWeb3
- primeclient
- standby0
- standby1
- standby2
- standby3
- virtualclient0
- virtualclient1
- virtualclient2
- virtualclient3

**mlicesxcli06.americas.manulife.net VMware ESXi, 6.0.0, 2494585**

Getting Started Summary Virtual Machines Resource Allocation Performance Configuration Users Events Permissions

**Configuration Issues**

ESXi Shell for the host has been enabled  
 SSH for the host has been enabled  
 System logs on host mlicesxcli06.americas.manulife.net are stored on non-persistent storage.

**General**

Manufacturer: IBM  
 Model: System x3850 X5-[7145AC1]-  
 CPU Cores: 24 CPUs x 2.66 GHz  
 Processor Type: Intel(R) Xeon(R) CPU X7542 @ 2.67GHz  
 License: VMware vSphere 6 Hypervisor - Licensed for 4 physical CP...  
 Processor Sockets: 4  
 Cores per Socket: 6  
 Logical Processors: 24  
 Hyperthreading: Inactive  
 Number of NICs: 6  
 State: Connected  
 Virtual Machines and Templates: 40  
 vMotion Enabled: N/A  
 VMware EVC Mode: Disabled  
 vSphere HA State: N/A  
 Host Configured for FT: N/A  
 Active Tasks:  
 Host Profile: N/A  
 Image Profile: ESXi-6.0.0-2494585-standard  
 Profile Compliance: N/A  
 DirectPath I/O: Supported

**Resources**

CPU usage: **1887 MHz** Capacity 24 x 2.66 GHz  
 Memory usage: **125862.00 MB** Capacity 262127.30 MB

Storage	Drive Type	Capacity
B-SRM-TEST06	Non-SSD	1.66 TB 69%

Network	Type
Isolated	Standard port group
vlan128	Standard port group

**Fault Tolerance**

Fault Tolerance Version: 6.0.0-6.0.0-6.0.0  
 Refresh Virtual Machine Counts  
 Total Primary VMs: 0  
 Powered On Primary VMs: 0  
 Total Secondary VMs: 0  
 Powered On Secondary VMs: 0

**Host Management**

Manage this host through VMware vCenter.

Commands

New Virtual Machine



misexcli06 - vSphere Client

File Edit View Inventory Administration Plug-ins Help

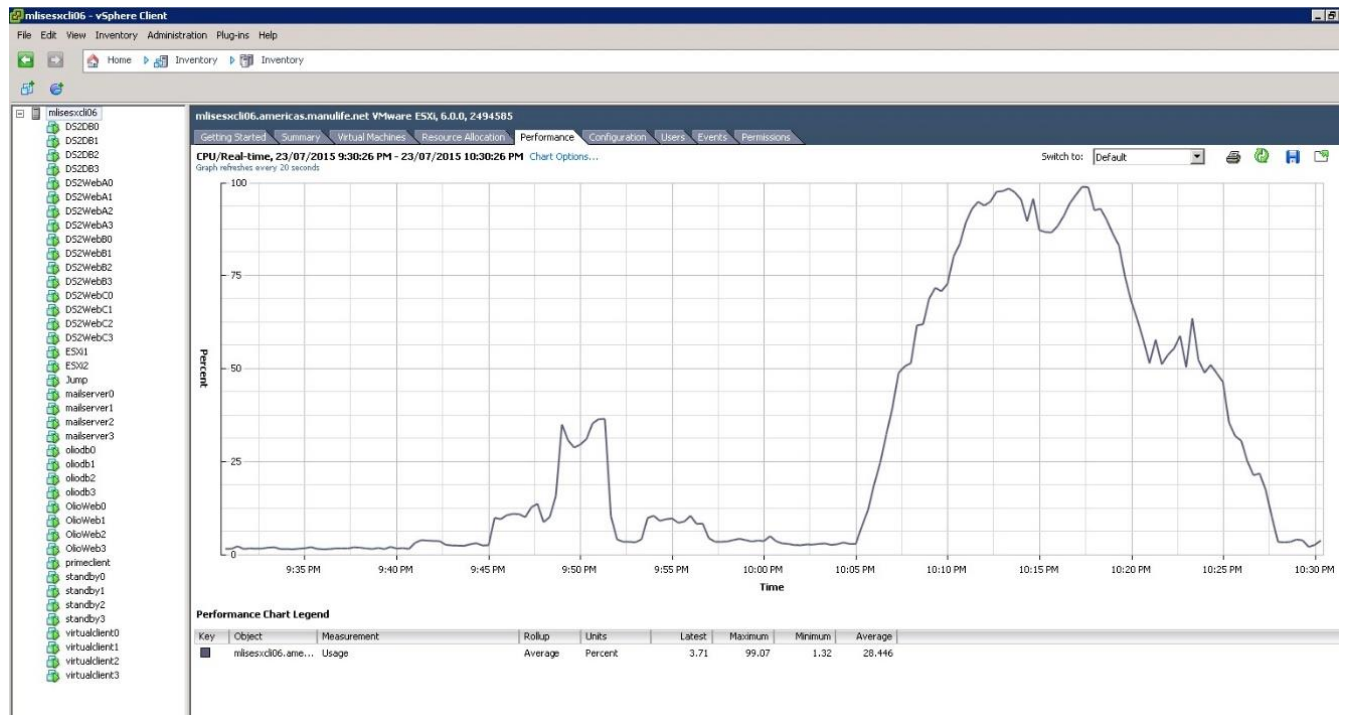
Home Inventory Inventory

misexcli06

misexcli06.americas.manulife.net VMware ESXi, 6.0.0, 2494585

Getting Started Summary Virtual Machines Resource Allocation Performance Configuration Users Events Permissions

Name	State	Provisioned Space	Used Space	Host CPU - ...	Host Mem - MB	Guest Mem - %	VMware Tools Runn...
DS2DB0	Powered On	162.85 GB	42.85 GB	4806	5848	27	Running
DS2DB1	Powered On	128.19 GB	35.57 GB	4740	5568	22	Running
DS2DB2	Powered On	128.19 GB	35.57 GB	4361	5299	12	Running
DS2DB3	Powered On	88.19 GB	50.26 GB	3809	8173	4	Running
DS2WebA0	Powered On	107.44 GB	33.08 GB	3315	8098	7	Running
DS2WebA1	Powered On	49.17 GB	31.96 GB	3296	3248	57	Running
DS2WebA2	Powered On	88.17 GB	30.25 GB	3186	7126	7	Running
DS2WebA3	Powered On	88.17 GB	30.25 GB	2914	3692	10	Running
DS2WebB0	Powered On	86.17 GB	78.65 GB	2870	6075	41	Running
DS2WebB1	Powered On	86.17 GB	9.23 GB	2846	5106	33	Running
DS2WebB2	Powered On	49.17 GB	49.17 GB	2832	3196	53	Running
DS2WebB3	Powered On	86.17 GB	9.23 GB	2673	5889	41	Running
DS2WebC0	Powered On	86.17 GB	9.24 GB	2669	5369	43	Running
DS2WebC1	Powered On	49.17 GB	31.93 GB	1803	3434	43	Running
DS2WebC2	Powered On	12.17 GB	4.78 GB	1277	384	3	Running
DS2WebC3	Powered On	12.17 GB	4.46 GB	1224	368	1	Running
ESX1	Powered On	12.17 GB	4.92 GB	1164	363	1	Running
ESX2	Powered On	12.17 GB	4.46 GB	932	390	0	Running
Jump	Powered On	12.17 GB	4.46 GB	884	343	1	Running
mailserver0	Powered On	16.17 GB	16.17 GB	807	1768	12	Running
mailserver1	Powered On	12.17 GB	4.51 GB	797	348	3	Running
mailserver2	Powered On	16.17 GB	16.17 GB	564	1790	15	Running
mailserver3	Powered On	16.17 GB	16.17 GB	552	1788	12	Running
oliodb0	Powered On	16.17 GB	16.17 GB	521	1801	16	Running
oliodb1	Powered On	138.19 GB	59.88 GB	408	8020	1	Running
oliodb2	Powered On	49.17 GB	31.94 GB	402	3173	35	Running
oliodb3	Powered On	123.37 GB	43.39 GB	393	8243	6	Running
OlioWeb0	Powered On	88.19 GB	39.17 GB	219	8179	10	Running
OlioWeb1	Powered On	5.17 GB	658.67 MB	128	3364	2	Not running
OlioWeb2	Powered On	5.17 GB	658.79 MB	103	4125	1	Not running
OlioWeb3	Powered On	5.67 GB	4.86 GB	30	276	4	Running
primeclient	Powered On	5.67 GB	4.86 GB	29	307	8	Running
standby0	Powered On	5.67 GB	4.86 GB	29	297	4	Running
standby1	Powered On	5.67 GB	4.86 GB	24	312	4	Running
standby2	Powered On	12.17 GB	4.45 GB	19	373	3	Running
standby3	Powered On						



Una vez que las pruebas con uno, dos y cuatro Tiles fueron completadas, se realizó un proceso de migración V2V hacia un servidor Hyper-V temporal. Una vez completado este proceso, se instaló Hyper-V Server 2012 R2 en el IBM System x3850 descrito anteriormente y se migraron las máquinas al hipervisor recientemente instalado. Por último, se instalaron en cada máquina virtual los Servicios de Integración de Hyper-V (equivalentes a Vmtools de Vmware) y se corrieron las pruebas sobre la plataforma de Microsoft.





Hyper-V Manager  
 WIN-6M4NPNL2AE8

Virtual Machines					
Name	State	CPU Usage	Assigned Memory	Uptime	Status
DS2DB2	Running	3 %	4096 MB	2.00:38:48	
OlioWeb0	Running	4 %	6144 MB	2.00:38:38	
virtualclient3	Running	3 %	8192 MB	2.04:24:11	
OlioWeb1	Running	3 %	6144 MB	2.00:38:37	
DS2DB0	Running	4 %	4096 MB	2.00:38:49	
virtualclient0	Running	4 %	8192 MB	3.05:42:33	
OlioWeb2	Running	4 %	6144 MB	2.00:37:08	
virtualclient1	Running	3 %	8192 MB	3.04:19:40	
virtualclient2	Running	4 %	8192 MB	3.04:19:38	
DS2DB3	Running	4 %	4096 MB	2.00:38:46	
DS2DB1	Running	1 %	4096 MB	2.00:38:47	
DS2WebA0	Running	1 %	2048 MB	2.00:38:28	
DS2WebA1	Running	1 %	2048 MB	2.00:38:24	
DS2WebA2	Running	1 %	2048 MB	2.00:38:25	
DS2WebB0	Running	1 %	2048 MB	2.00:38:26	
DS2WebB1	Running	0 %	2048 MB	2.00:38:21	
DS2WebC1	Running	0 %	2048 MB	2.00:38:39	
DS2WebC2	Running	0 %	2048 MB	2.00:38:41	
DS2WebA3	Running	1 %	2048 MB	2.00:38:43	
DS2WebC3	Running	0 %	2048 MB	2.00:38:32	
DS2WebB2	Running	1 %	2048 MB	2.00:38:31	
DS2WebC0	Running	0 %	2048 MB	2.00:38:33	
mailserver2	Running	0 %	3272 MB	2.00:38:11	

Checkpoints

Completadas las pruebas sobre Hyper-V, se realizó el V2V hacia un XenServer temporal. Luego se instaló XenServer 6.5 en el Servidor y me migraron las máquinas al hipervisor instalado en el servidor físico. Luego se instaló Xenserver Tools en las máquinas virtuales y se corrieron las pruebas.

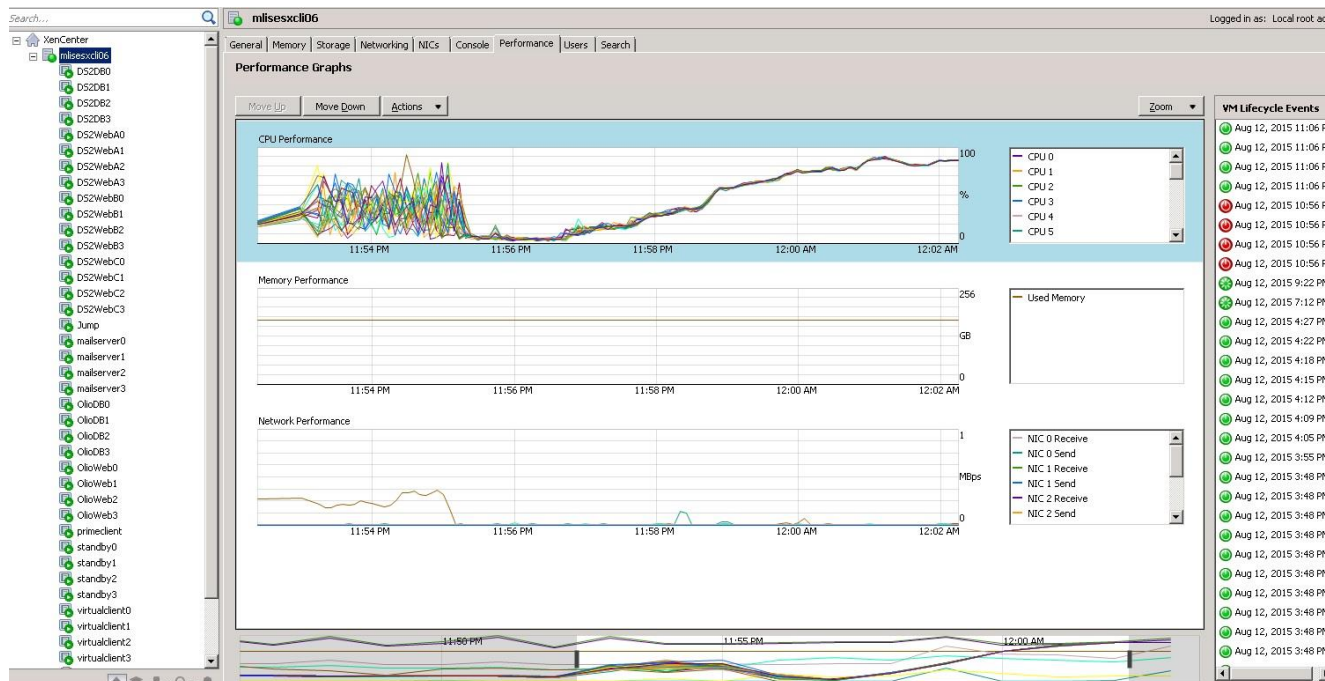


The screenshot shows the XenCenter interface for the VM 'misesxcli06'. The console window displays the XenServer 6.5 configuration menu. The menu options include: Customize System, Status Display (highlighted), Network and Management Interface, Authentication, Virtual Machines, Disks and Storage Repositories, Resource Pool Configuration, Hardware and BIOS Information, Keyboard and Timezone, Remote Service Configuration, Backup, Restore and Update, Technical Support, Reboot or Shutdown, Local Command Shell, and Quit. On the right side of the console, system information is displayed: IBM System x3850 X5 -I7145AC11-, XenServer 6.5.0-90233c, and Management Network Parameters: Device eth0, IP address 10.201.152.54, Netmask 255.255.255.192, Gateway 10.201.152.1. Navigation instructions at the bottom indicate: <Enter> OK <Up/Down> Select <Enter> Fingerprints <F5> Refresh.

The screenshot shows the XenCenter Overview for the VM 'misesxcli06'. It displays a table with columns for Name, CPU Usage, Used Memory, Address, Network (avg / max Kbps), Disks (avg / max Kbps), and Uptime. The table lists various VMs and their resource usage and uptime.

Name	CPU Usage	Used Memory	Address	Network (avg / max Kbps)	Disks (avg / max Kbps)	Uptime
misesxcli06	94% of 24 CPUs	174429 of 262127 MB	10.201.152.54, 10.2...	150/889	-	2 days 8 hours 8 minutes
Default install of XenServer						
mailserver2	62% of 4 CPUs	3351 of 8192 MB	fe80::f42f:5de9:c48...	452/452	3274/8931	1 day 8 hours 18 minutes
mailserver3	57% of 4 CPUs	2968 of 8192 MB	fe80::3c48:825e:a6...	347/347	2680/7383	1 hour 46 minutes
virtualclient0	46% of 2 CPUs	1178 of 8192 MB	fe80::84b2:db73:7a...	4935/4935	59/117	1 day 11 hours 13 minutes
virtualclient3	40% of 2 CPUs	1179 of 8192 MB	fe80::10fd:d017:b8...	4860/4860	33/66	1 day 6 hours 4 minutes
virtualclient2	37% of 2 CPUs	1214 of 8192 MB	fe80::40c9:362f:c8b...	4567/4567	39/79	1 day 6 hours 48 minutes
virtualclient1	29% of 2 CPUs	1206 of 8192 MB	fe80::5053:a439:af...	3142/3142	41/81	1 day 1 hour 8 minutes
mailserver1	25% of 4 CPUs	3441 of 8192 MB	fe80::aca3:f125:bd0...	500/500	3841/10795	1 day 9 hours 19 minutes
OloWeb0	21% of 4 CPUs	XenServer Tools not l...	-	-	-	1 day 6 hours 43 minutes
OloWeb3	21% of 4 CPUs	XenServer Tools not l...	-	-	-	1 day 7 hours 2 minutes
OloWeb2	21% of 4 CPUs	XenServer Tools not l...	-	-	-	1 day 6 hours 43 minutes
OloWeb1	19% of 4 CPUs	XenServer Tools not l...	-	-	-	1 day 6 hours 43 minutes
DS2WebA1	18% of 2 CPUs	XenServer Tools not l...	-	-	-	49 minutes
DS2WebA3	16% of 2 CPUs	XenServer Tools not l...	-	-	-	1 day 7 hours 2 minutes
OloDB3	16% of 2 CPUs	XenServer Tools not l...	-	-	-	1 day 6 hours 42 minutes
OloDB2	14% of 2 CPUs	XenServer Tools not l...	-	-	-	1 day 6 hours 43 minutes
OloDB0	13% of 2 CPUs	XenServer Tools not l...	-	-	-	1 day 6 hours 42 minutes
OloWeb1	13% of 4 CPUs	XenServer Tools not l...	-	-	-	7 hours 2 minutes
DS2DB0	12% of 4 CPUs	XenServer Tools not l...	-	-	-	





## 12. Evaluación

Al completarse cada ejecución de Vmmark, la aplicación recolectaba los resultados de cada aplicación evaluada. Al estar Vmmark diseñada solo para ser ejecutada utilizando varios hipervisores ESXi, se simuló esa infraestructura. En consecuencia las estadísticas de utilización de recursos de los hipervisores fueron recolectadas manualmente a la finalización de cada ejecución. En resumen, se obtuvieron estadísticas a nivel de la aplicación, con los mismos contadores para cada plataforma, y estadísticas a nivel hipervisor de la utilización de recursos tanto del hipervisor como de las máquinas virtuales. Desafortunadamente los contadores provistos por cada plataforma difieren enormemente en cantidad y calidad entre una y otra. El hipervisor ESXi dispone de gran cantidad de contadores adecuados que permiten realizar un análisis detallado de comportamiento del sistema. XenServer cuenta con una cantidad muy inferior, aunque con los mínimos necesarios para analizar el comportamiento del hipervisor las máquinas que corren sobre él. Por último, Hyper-V carece de los contadores esenciales para analizar el uso de recursos y la contención en la utilización de los mismos. Esta situación dificulta el análisis al no



disponer con suficientes contadores útiles comparables, al contar solo con los contadores uso de cpu y uso de memoria para comparar el desempeño de los hipervisores. Tratándose de sistemas sobre provisionados, en donde la sumatoria de los recursos virtuales supera la cantidad de recursos físicos del servidor, en nuestro caso en particular el procesador, se requieren contadores específicos para medir la contención de recursos (situación en la cual las máquinas virtuales compiten por los recursos físicos produciéndose demoras en la provisión de dichos recursos, que impacta directamente en su performance). Afortunadamente en adición al nivel utilización de los principales recursos, o al menos los que más frecuentemente son sobre utilizados, contamos con los resultados entregados por vmmark del desempeño de las aplicaciones. Estos tienen los mismos contadores en cada plataforma, por lo que su comparación será un proceso directo.

### 12.1. Resultados 1 Tiles

1 TILE							
	Duración	% Uso CPU			% Uso Memoria		
		Desv Est	Prom	Max	Desv Est	Prom	Max
ESXi	01:02:35	3,78	20,84	27,50	0,05	20,41	20,46
Hyper-V	01:03:11	6,71	32,58	42,97	0,01	17,34	17,35
XenServer	01:03:29	7,24	34,90	43,52	0,00	23,37	23,37

1 TILE			
	Mail		
	Tasks Dispatched	Tasks Completed	Send Mail Action Latency
ESXi	108412	108411	0
Hyper-V	108180	108180	0
XenServer	108258	108258	0

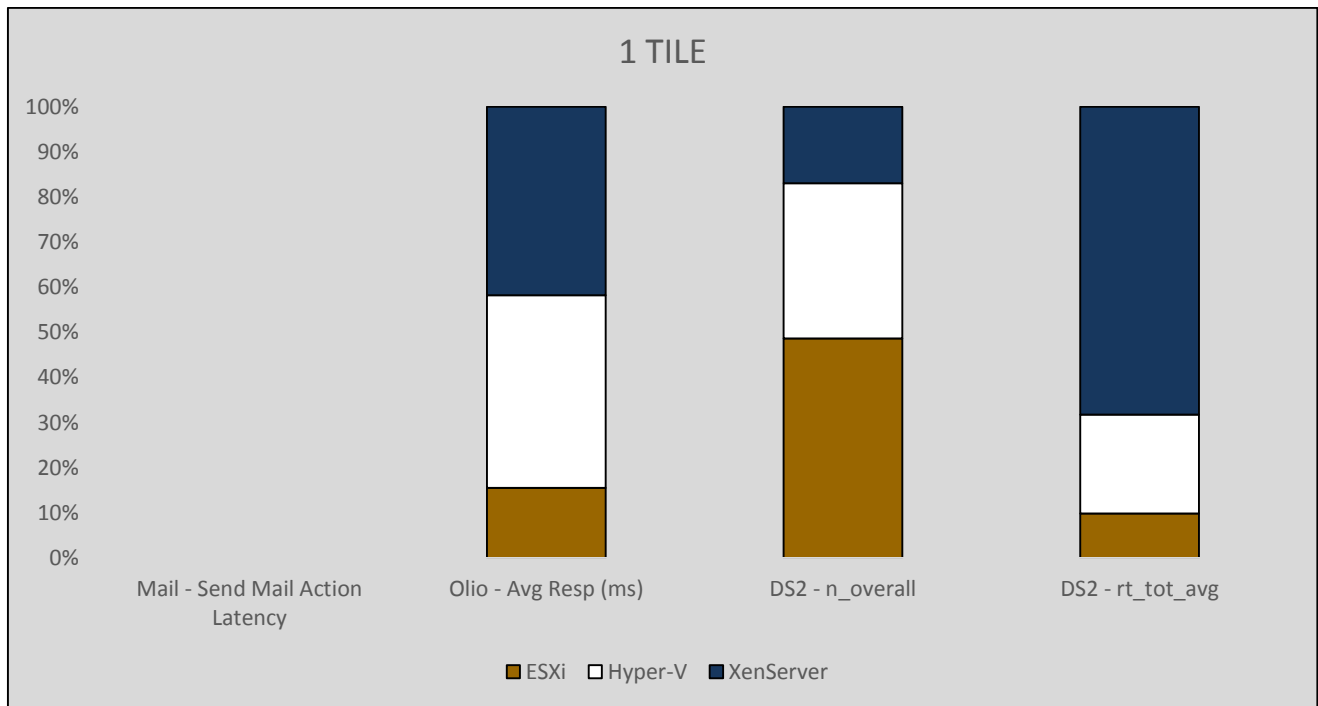
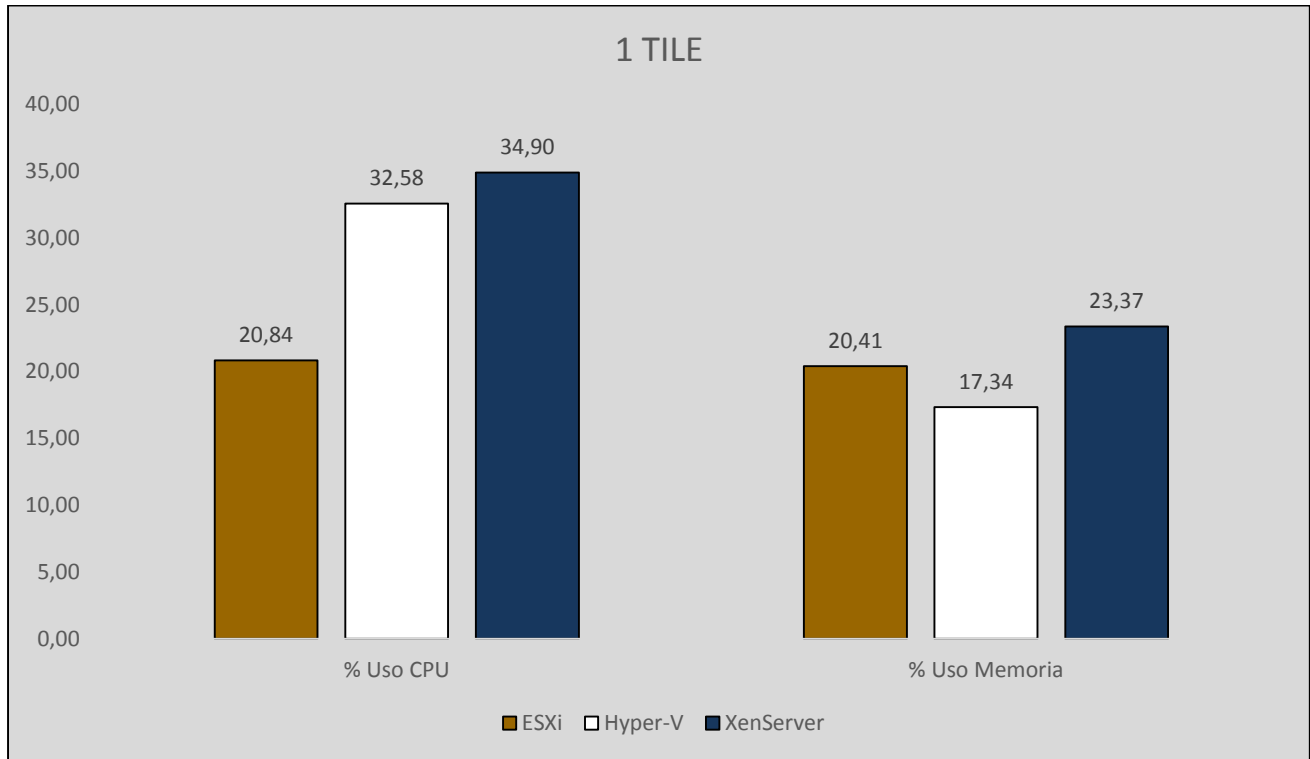


1 TILE					
	Olio				
	Operations initiated	Operations completed	Avg Resp (Sec)	Min Resp (Sec)	Max Resp (Sec)
ESXi	238942	238940	0,09	0,06	1,87
Hyper-V	232419	232401	0,24	0,13	2,79
XenServer	232914	232904	0,23	0,09	2,80

1 TILE			
	DS2		
	n_overall	rt_tot_lastn_max	rt_tot_avg
ESXi	669004	124,28	46,64
Hyper-V	472831	274,28	103,2
XenServer	233526	787,64	321,76

Referencias DS2

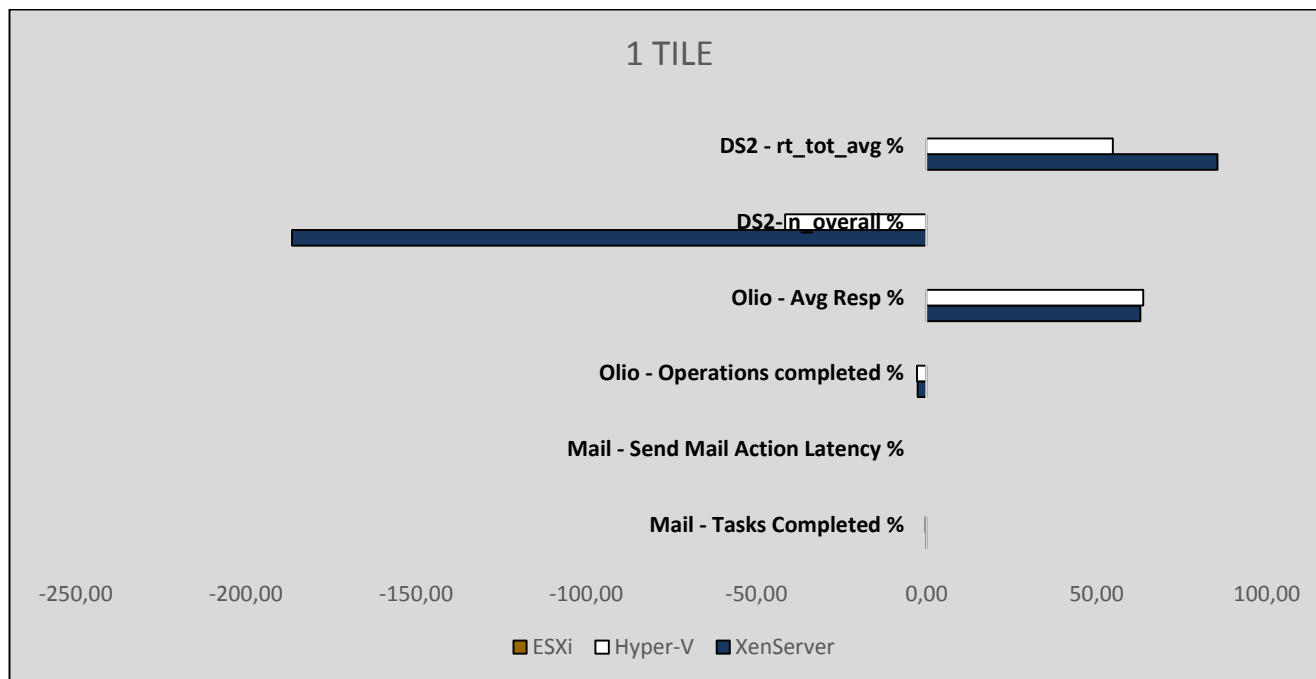
n_overall: total number of orders processed between Stats reset and Run over
rt_tot: total response time seen by a user during an order cycle
rt_tot_avg: average of rt_tot over all orders
rt_tot_lastn_max: maximum rt_tot seen





En estos gráficos podemos observar que un reducido nivel de sobre aprovisionamiento (1:1.6 CPU) en cuanto a la utilización memoria de memoria, Hyper-V tuvo menor consumo seguido por ESXi y Xenserver. La brecha entre cada plataforma no es elevada, aunque la diferencia entre Hyper-V y XenServer fue de poco más de 6%. Si bien esto no parecería elevado, debemos considerar que en servidores de gran capacidad este porcentaje puede representar por ejemplo 61 GB de RAM, lo cual puede llegar a ser utilizado por para aprovisionar máquinas virtuales adicionales. La diferencia de utilización de CPU es mucho más significativa, ESXi consumió mucho menos procesador que los otros hipervisores, casi 12% menos que Hyper-V y 14% que XenServer. Antes de continuar debemos recordar que Vmmark genera las mismas tareas en cada corrida y antes de iniciar una nueva restaura las bases de datos, logs y configuraciones de las aplicaciones de modo que las aplicaciones corran en el mismo entorno que la ejecución anterior de la prueba.

Ahora que hemos revisado la utilización de recursos de cada plataforma, veremos el desempeño de las aplicaciones en hipervisor. Podemos observar el servidor de correo tuvo un prácticamente la misma performance, casi la misma cantidad de tareas ejecutadas y sin latencia en el envío de correos. En cuanto a la aplicación Web Olio, observamos que ESXi ejecuto casi un 3% más de operaciones que las otras plataformas y con menor tiempo de respuesta para las operaciones, 63% menos que Hyper-V y 62% menos que Xenserver. Por último, para la aplicación de comercio electrónico DS2, el desempeño en fue muy superior en ESXi al de Hyper-V y muy por detrás estuvo XenServer. ESXi ejecuto 41% más de operaciones que Hyper-V y 186% más que XenServer. En cuanto al tiempo de respuesta en las operaciones, ESXi tuvo 55% menos que Hyper-V y 86% menos XenServer.



### 12.2. Resultados 3 Tiles

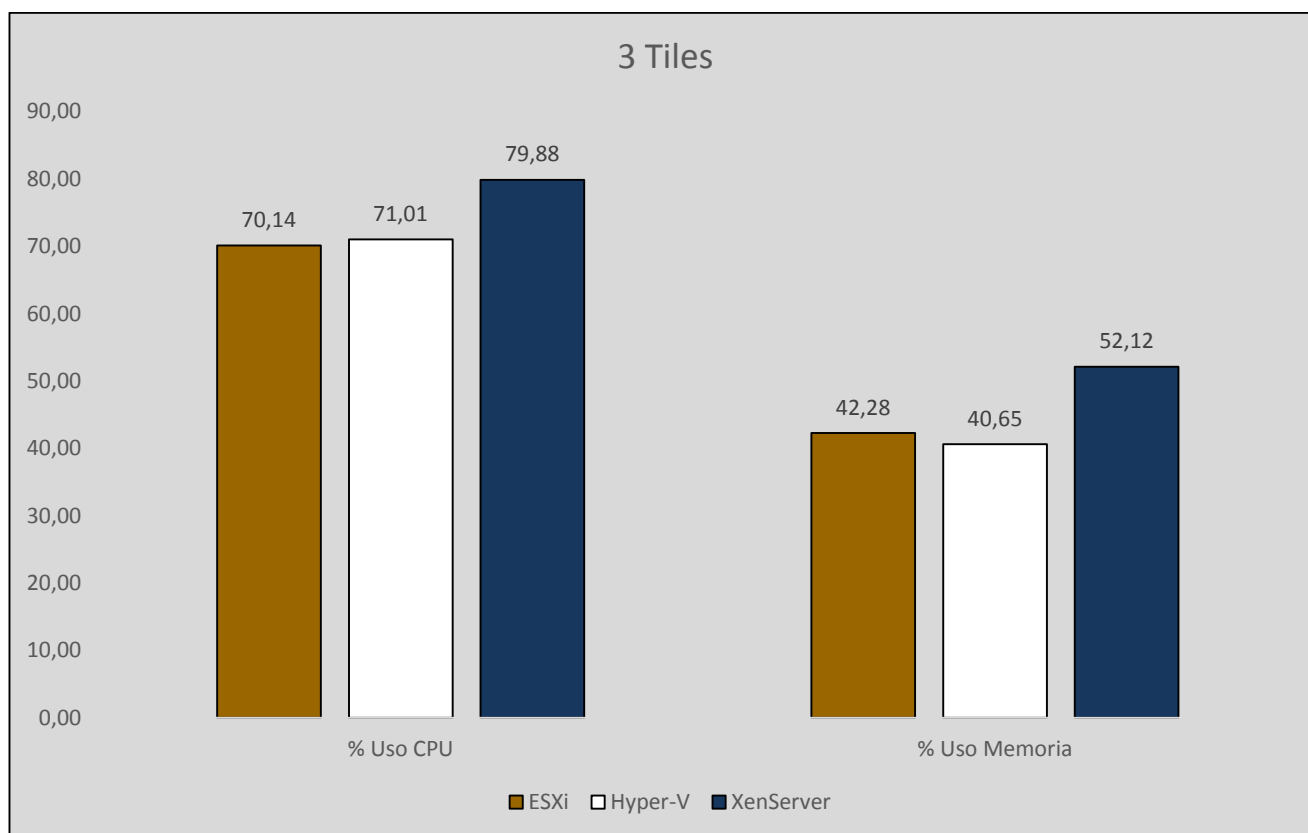
3 TILES							
	Duración	% Uso CPU			% Uso Memoria		
		Desv Est	Prom	Max	Desv Est	Prom	Max
ESXi	01:05:58	16,53	70,14	90,16	0,42	42,28	42,66
Hyper-V	01:06:49	18,02	71,01	83,12	0,05	40,65	40,76
XenServer	01:07:23	16,69	79,88	89,77	0,00	52,12	52,12

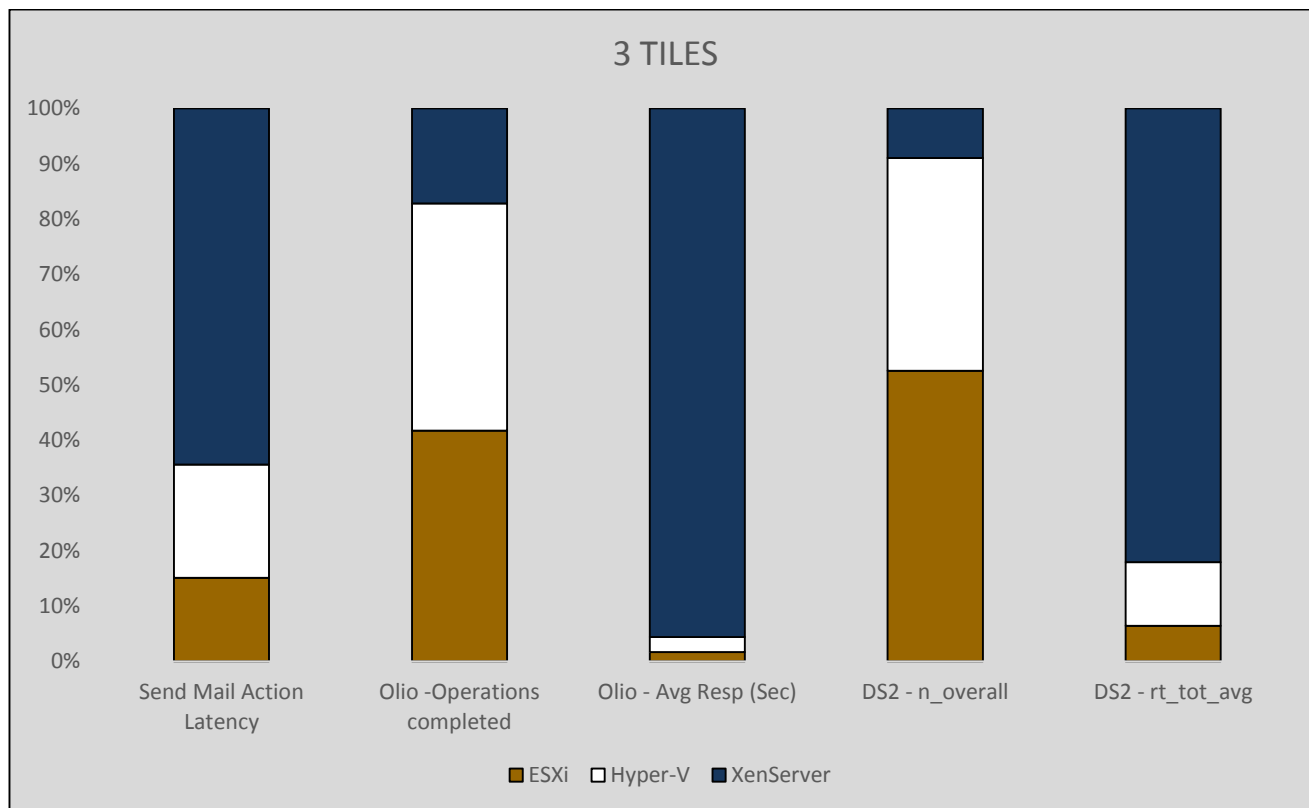
3 TILES			
	Mail		
	Tasks Dispatched	Tasks Completed	Send Mail Action Latency
ESXi	327795	327794	42,72
Hyper-V	326567	326567	57,85
XenServer	324283	324277	181,88



3 TILES					
	Olio				
	Operations initiated	Operations completed	Avg Resp (Sec)	Min Resp (Sec)	Max Resp (Sec)
ESXi	721699	721678	0,15	0,07	2,50
Hyper-V	711550	711526	0,24	0,13	2,58
XenServer	321475	296880	8,40	2,24	22,59

3 TILES			
	DS2		
	n_overall	rt_tot_lastn_max	rt_tot_avg
ESXi	1605064	214,17	75,70
Hyper-V	1175472	367,09	135,26
XenServer	273875	2226,97	965,16





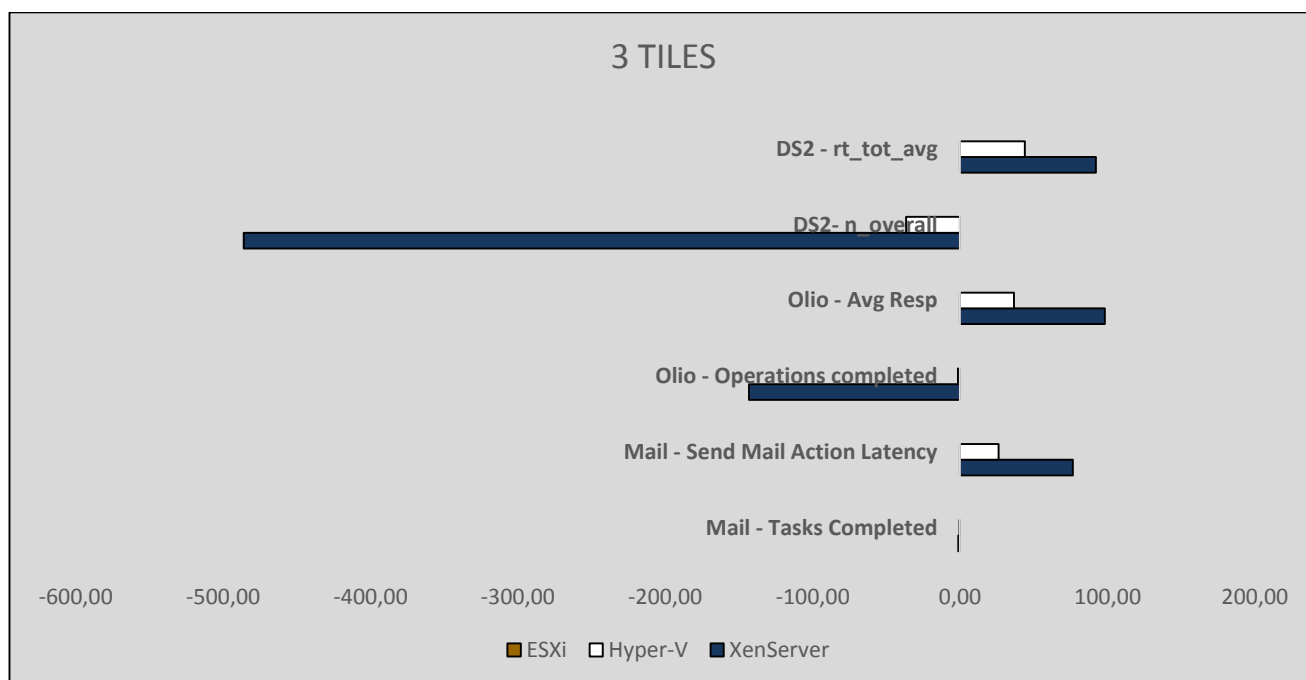
Con un sobre aprovisionamiento de CPU dentro de los parámetros utilizados en ambientes de producción, 1:3.5, observamos que la tendencia observada para 1 TILE se mantiene. ESXi apenas consume menos procesador que Hyper-V y XenServer utiliza mucho más que los otros. En cuanto a la memoria, nuevamente Hyper-V utiliza poco menos que ESXi, apenas 2% y el consumo de XenServer es mucho mayor superando a Hyper-V por casi 12%.

En cuanto al desempeño de las máquinas virtuales y las aplicaciones que corren sobre las mismas, la tendencia también se mantuvo. Para el servidor de correo, ESXi realizó 0.38% más operaciones que Hyper-V y 1.08% más que XenServer. Pero la diferencia principal se observó en la demora promedio de envío de correo, ESXi tuvo 26% menos de demora que Hyper-V y 77% menos que XenServer. Para la aplicación web Olio, ESXi continuo mostrando una performance muy superior, aunque se acorto la brecha con Hyper-V y la profundizo con XenServer. El hipervisor de Vmware realizo 1% más





operaciones que su par de Microsoft y 143% más que el de Citrix. Se observó el mismo resultado en la demora de las solicitudes para Olio, ESXi tuvo 37% menos de demora que Hyper-V y 98% menos que XenServer. Finalmente analizamos DS2, en donde ESXi mantuvo a tendencia, realizando 37% más de operaciones que Hyper-V y 486% más que XenServer. Los tiempos de respuesta para DS2 son coherentes con la cantidad de operaciones, ESXi tuvo respuestas 44% más rápidas que Hyper-V y 92% que XenServer.



### 12.3. Resultados 4 Tiles

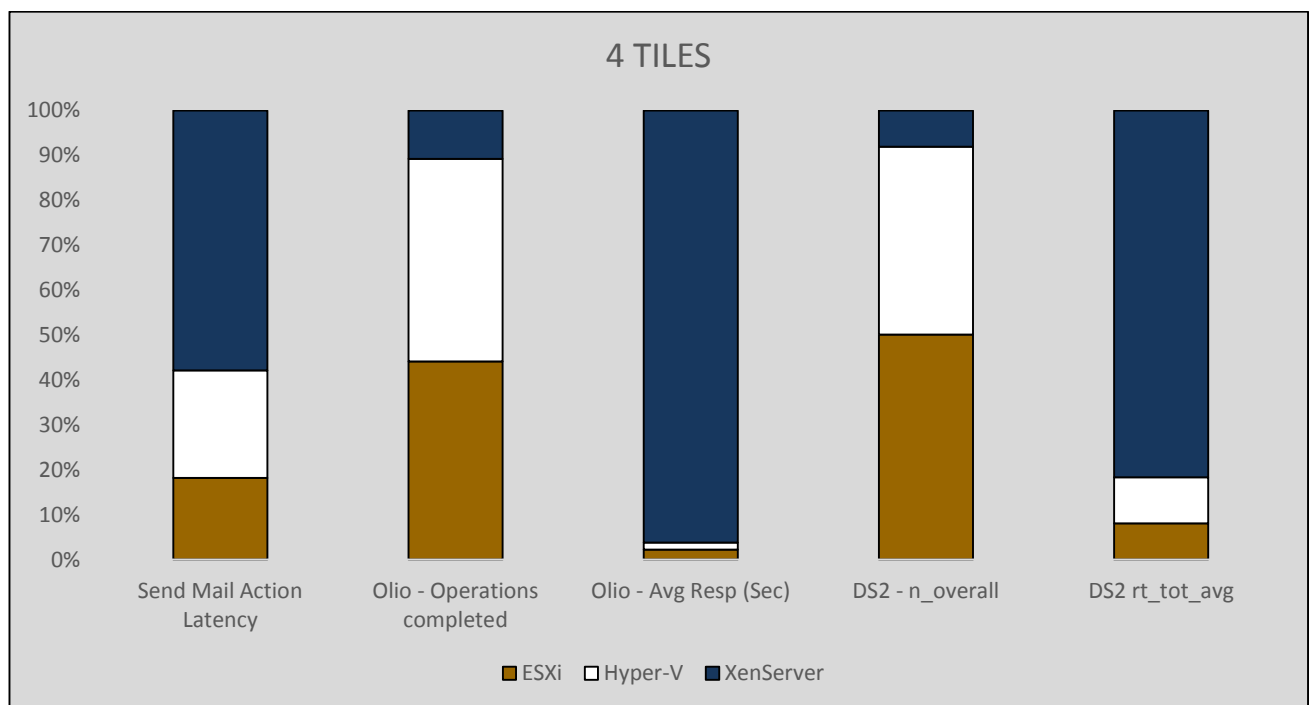
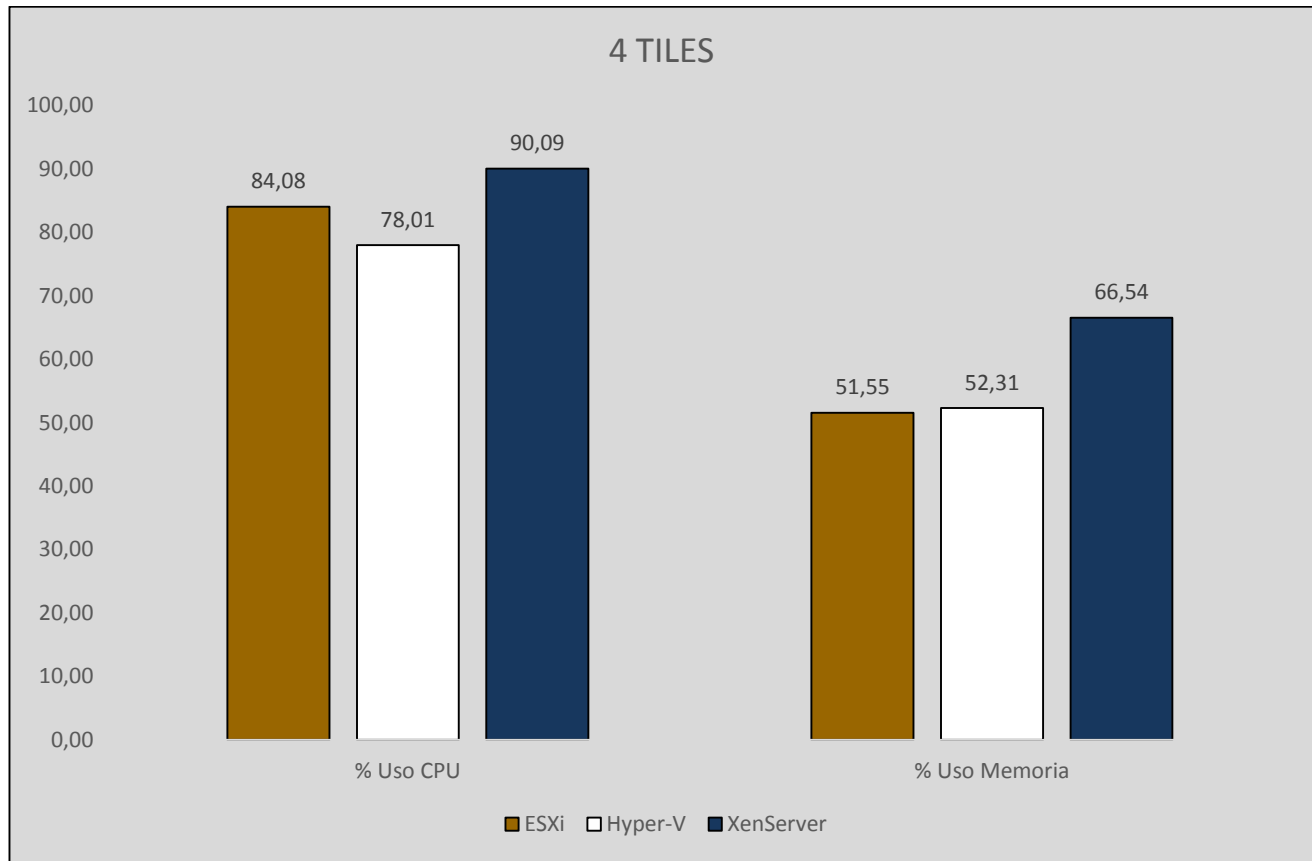
4 TILES							
	Duración	% Uso CPU			% Uso Memoria		
		Desv Est	Prom	Max	Desv Est	Prom	Max
ESXi	01:07:28	20,84	84,08	100,00	0,70	51,55	52,43
Hyper-V	01:07:51	17,88	78,01	97,17	0,08	52,31	52,46
XenServer	01:09:03	22,32	90,09	99,01	0,00	66,54	66,54



4 TILES			
	Mail		
	Tasks Dispatched	Tasks Completed	Send Mail Action Latency
ESXi	441887	441886	45,17
Hyper-V	438825	438823	59,43
XenServer	432726	432720	143,58

4 TILES					
	Olio				
	Operations initiated	Operations completed	Avg Resp (Sec)	Min Resp (Sec)	Max Resp (Sec)
ESXi	927290	927255	0,44	0,07	6,25
Hyper-V	948440	948371	0,31	0,15	2,58
XenServer	269720	227482	18,94	5,96	44,17

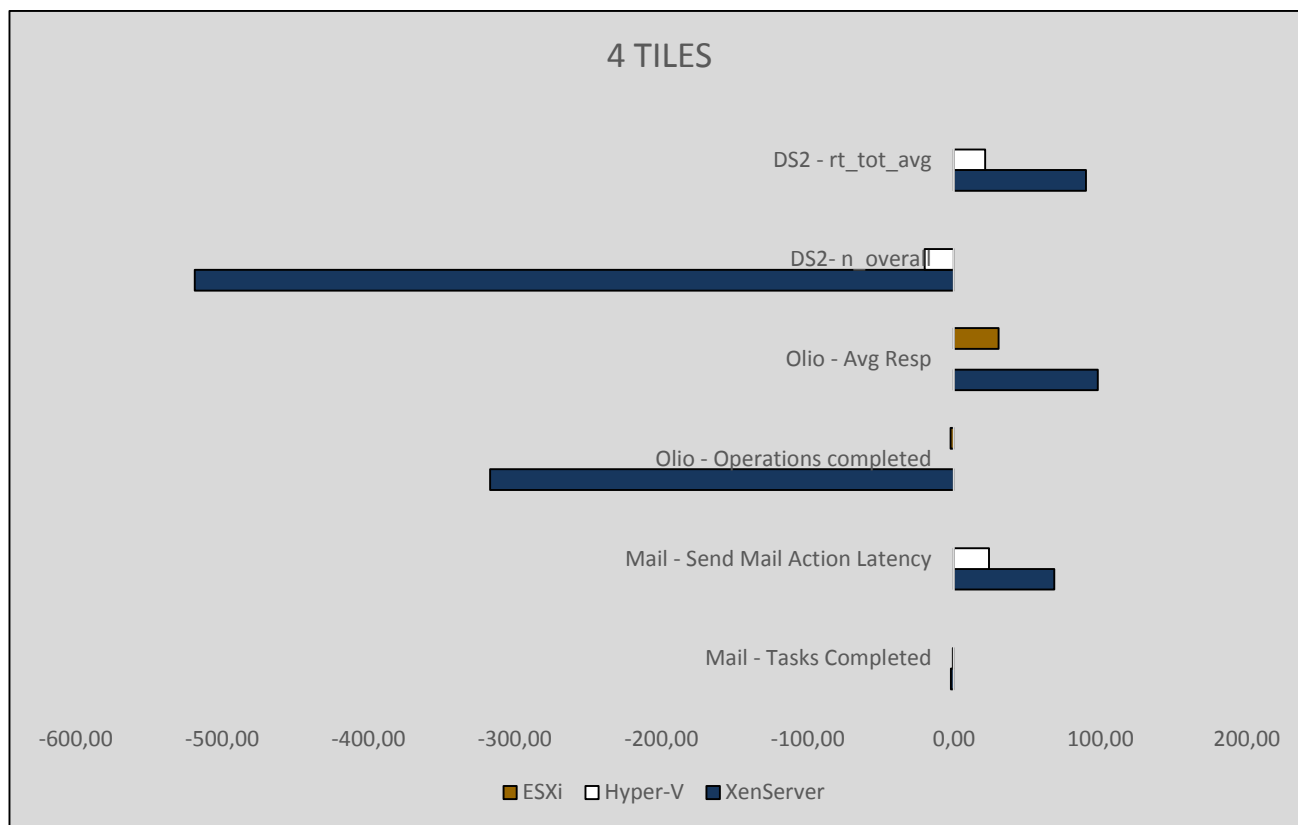
4 TILES			
	DS2		
	n_overall	rt_tot_lastn_max	rt_tot_avg
ESXi	1644797	380,17	131,55
Hyper-V	1370417	398,81	167,42
XenServer	265885	2691,15	1332,82





Finalmente para las pruebas con 4 TILES, con una proporción de 1:4.5 (Procesador Físico: Procesador Virtual) como la que se encuentra en ambientes de producción reales. Podemos ver un cambio en la tendencia de uso de CPU, esta vez ESXi utilizo en promedio 6% más que Hyper-V, Xenserver mantuvo la distancia con el hipervisor con menor uso demandando. En cuanto al uso de memoria se mantuvo la tendencia, ESXi utilizo 1% de memoria que Hyper-V y 15% menos que Xen Server.

Para el servidor de correo la performance fue similar, ESXi realizo 0,7% más de operaciones y 2% más que Xenserver. La demora de envío de correo no vario con respecto a las pruebas de 3 TILES, ESXi envió 24% más rápidos los correos que Hyper-V y 69% más rápido que XenServer. En cuanto a la aplicación Olio, a diferencia de las pruebas anteriores, Hyper-V realizo 2% más de operaciones que ESXi y 317% más que XenServer. En cuanto al tiempo de respuesta de Olio para las operaciones, Hyper-V atendió las solicitudes 31% más rápido que ESXi y 98% que XenServer. Por último, para DS2 se mantuvo la tendencia de las 2 pruebas anteriores, ESXi realizó 20% más de operaciones que Hyper-V y 519% que XenServer. ESXi atendió las solicitudes de DS2 21% más rápido que Hyper-V y 90% que XenServer.



### 13. Conclusiones

Se consiguió un servidor físico a con características de un servidor producción de ambientes reales compatible con los tres hipervisores que se iba a estudiar. Lo logro adaptar el benchmark de Vmware de uso libre, pero para uso exclusivo en ambientes multiservidores ESXi, para poder generar igual carga similar a la encontrada en ambientes reales en cada ejecución. Esto nos permitió tener una referencia que nos permita comparar el desempeño de los distintos hipervisores. Se corrió en cada plataforma en benchmark utilizando 1, 3 y 4 tiles para observar el comportamiento de las plataformas ante distintos niveles de sobre provisionamiento de máquinas virtuales.

Observamos claramente una utilización superior de recursos físicos por parte de



Xenserver para las distintas pruebas. También se observó que a pesar de utilizar más recursos las aplicaciones corriendo en Citrix tuvieron un rendimiento muy inferior al de las otras plataformas. En cuanto a ESXi y Hyper-V observamos un menor consumo de recursos del primero con un tile, pero que tendió a equipararse a medida que el sobre aprovisionamiento aumentaba. Esto ser reflejo también en la performance de las aplicaciones, mejor en ESXi en líneas generales aunque la brecha disuria conforme el nivel de sobre aprovisionamiento aumentaba. Esto último fu el caso contrario con XenServer, a medida que el nivel de sobre aprovisionamiento aumentaba, aumentaba la diferencia de performance en favor de ESXi.

Se trabajó con las versiones gratuitas de ESXi y Hyper-V. XenServer es de uso gratuito. Se observó que ESXi fue el más fácil de utilizar, más flexible y con mayor cantidad de funciones y herramientas para su administración. XenServer los sigue más atrás en los aspectos mencionados, y muy lejos está Hyper-V para el cual las tareas de despliegue, migración y fundamentalmente monitoreo fueron considerablemente más complicadas que en las otras plataformas.

El análisis de cada plataforma completa excede los límites de este trabajo, por lo que limitándonos solo al análisis del hipervisor aislado podemos concluir que ESXi a los otros hipervisores seguido por Hyper-V y bastante más atrás por XenServer. En ambientes de gran tamaño, con servidores de gran capacidad y altos niveles de sobre aprovisionamiento se recomienda el uso preferentemente de ESXi en primera instancia o de Hyper-V en su defecto. En ambientes más pequeños en donde el presupuesto para las correspondientes licencias para las plataformas de virtualización es limitado y los niveles de sobre aprovisionamiento de máquinas virtuales XenServer es una alternativa a ser considerada.



## 14. Referencias

A Performance Comparison of Hypervisors for Cloud Computing - Suganya Sridharan - University of North Florida - 2012

[Makhija] VMmark A Scalable Benchmark for Virtualized systems - Makhija, V., B. Herndon, P. Smith, L. Roderick, E. Zamost, J. Anderson - Technical Report VMware-TR-2006-002 - 2006

[Vmware] VMware VMmark® Benchmarking Guide- Technical Report VMware EN-000006-06 - 2014

A Performance Comparison of Hypervisors - Technical Report VMware - 2007

A Component-Based Performance Comparison of Four Hypervisors – J. Hwang, S. Zeng, F. Wu, T. Wood - 2014

Performance Evaluation of Hypervisors in the Private Cloud based on System Information using SIGAR Framework and for System Workloads using Passmark - P. Vijaya Vardhan Reddy, L. Rajamani – 2014

[Nanda] A Survey on Virtualization Technologies – Nanda S., T. Chiueh - Technical report, Department of Computer Science, SUNY at Stony Brook, New York - 2005

<https://en.wikipedia.org/wiki/Hyper-V>