

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

MÁSTER EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN



“ESTUDIO DEL RENDIMIENTO EN LA
MIGRACIÓN EN VIVO DE UNA APLICACIÓN
EDGE EN DESPLIEGUE DE RED
VIRTUALIZADO”

TRABAJO FIN DE MÁSTER

Junio -2026

AUTOR: Sergio Sáez Heredia

DIRECTOR/ES: Katja Gilly de la Sierra
- Llamazares

Cristina Bernad Cantó

INDICE

INDICE	2
1 INTRODUCCIÓN	6
2 MARCO Y ARQUITECTURA DEL SISTEMA	8
2.1 Marco tecnológico	8
2.2 Migración en vivo de máquinas virtuales	9
2.3 Arquitectura del sistema	11
2.4 Visualización de la topología y servicios desplegados.....	13
2.5 Flujo de operación y escenarios de prueba	14
2.6 Componentes del sistema.....	15
2.6.1 Proxmox	15
2.6.2 MicroStack	16
2.6.3 OSM (Open Source MANO).....	16
2.6.4 Apache Benchmark.....	16
2.6.5 Herramientas de monitorización del sistema.....	17
2.7 Justificación del entorno experimental	17
2.8 Relación con los objetivos del trabajo	18
3 METODOLOGÍA	19
3.1 Configuración del entorno experimental	19
3.1.1 Infraestructura de virtualización	19
3.1.2 Versiones del software empleado	19
3.1.3 Configuración de la instancia del servicio	20
3.1.4 Herramientas de monitorización del sistema.....	21
3.1.5 Automatización y sincronización de las pruebas.....	22
3.2 Diseño del experimento	22
3.3 Escenarios de prueba	24
3.3.1 Escenario A: Migración en vivo gestionada por MicroStack	24
3.3.2 Escenario B: Migración en vivo orquestada mediante OSM	25
3.4 Creación y configuración de la instancia del servicio	26
3.5 Generación de carga	28
3.6 Ejecución de las migraciones	30
3.6.1 Migración en vivo gestionada por MicroStack	30
3.6.2 Migración en vivo orquestada mediante OSM.....	31
3.6.3 Coordinación con la generación de carga y monitorización	33
3.7 Monitorización y recogida de métricas.....	34

3.7.1	Monitorización del uso de CPU y memoria	34
3.7.2	Monitorización de operaciones de entrada/salida en disco	34
3.7.3	Métricas de rendimiento del servicio	35
3.7.4	Organización y almacenamiento de los datos experimentales	35
3.7.5	Automatización y sincronización del proceso experimental.....	36
4	RESULTADOS.....	37
4.1	Rendimiento del servicio (Apache Benchmark)	37
4.1.1	Tiempo total de ejecución del test	37
4.1.2	Tiempo medio total de conexión	39
4.1.3	Análisis detallado de tiempos de conexión	40
4.2	Tiempo de migración.....	42
4.2.1	Evolución del tiempo de migración	43
4.3	Impacto en el consumo de recursos del sistema.....	45
4.3.1	Uso de CPU (vmstat)	45
4.3.2	Utilización de disco (iostat).....	46
4.3.3	Evolución temporal de las métricas del sistema	47
4.3.4	Evolución temporal por dirección de migración	50
4.4	Análisis comparativo global.....	53
5	CONCLUSIONES	55
5.1	Limitaciones del estudio	56
5.2	Líneas futuras de trabajo	57
6	Anexo A – Comandos de configuración de la instancia del servicio.....	58
7	Anexo B – Script de automatización de las migraciones	59
8	Bibliografía	61

INDICE DE FIGURAS

Figura 2.1 Arquitectura del entorno experimental para la evaluación de migraciones en vivo de máquinas virtuales.	12
Figura 2.2 Topología de red del entorno MicroStack mostrando el router virtual, la red externa y la red interna utilizada durante los experimentos, así como la instancia del servicio web desplegada en la infraestructura cloud.....	13
Figura 2.3 Vista de instancias de servicio (NS Instances) en la interfaz de OSM mostrando el servicio desplegado y en estado operativo durante la ejecución de los experimentos	14
Figura 2.4 Flujo de operación seguido durante la ejecución de las pruebas	15
Figura 3.1 Máquinas virtuales desplegadas en la plataforma Proxmox para la ejecución del experimento.	24
Figura 3.2 Instancia en entorno OpenStack.....	27
Figura 3.3 Detalle de la instancia del servicio web desplegada.	28
Figura 3.4 Instancia de servicio gestionada por OSM durante la ejecución de las migraciones orquestadas,	32
Figura 3.5 Ejecución de una migración en vivo desde la interfaz de OSM a nivel de servicio.	32
Figura 4.1 Gráfica comparativa “Apache Benchmark” – tiempo total de ejecución del test	38
Figura 4.2 Gráfica comparativa tiempo medio total de conexión	39
Figura 4.3 Comparación de los tiempos medios de Connect, Processing, Waiting y Total obtenidos con Apache Benchmark para los escenarios MicroStack y OSM.	41
Figura 4.4 Gráfica comparativa tiempo de migración	42
Figura 4.5 Evolución del tiempo de migración para cada una de las 20 migraciones ejecutadas en los escenarios MicroStack y OSM.	44
Figura 4.6 Gráfica comparativa Uso del sistema de CPU durante la migración	45
Figura 4.7 Gráfica comparativa Utilización del disco durante la migración.....	46

Figura 4.8 Evolución temporal del uso de CPU en modo sistema durante las migraciones en vivo en el escenario MicroStack. Se muestra el valor medio junto con una banda de variabilidad.	48
Figura 4.9 Evolución temporal del uso de CPU en modo sistema durante las migraciones en vivo en el escenario OSM. Se representa el valor medio junto con una banda de variabilidad.	48
Figura 4.10 Evolución temporal de la utilización del disco (%util) durante las migraciones en vivo en el escenario MicroStack. Se muestra el valor medio con su variabilidad.	49
Figura 4.11 Evolución temporal de la utilización del disco (%util) durante las migraciones en vivo en el escenario OSM. Se representa el valor medio con su variabilidad.	49
Figura 4.12 Evolución temporal del uso de CPU en modo sistema durante las migraciones en vivo en el escenario MicroStack, diferenciando ambas direcciones de migración. Se muestra el valor medio junto con la variabilidad de los datos.	51
Figura 4.13 Evolución temporal del uso de CPU en modo sistema durante las migraciones en vivo en el escenario OSM, diferenciando ambas direcciones de migración. Se representa el valor medio junto con la variabilidad de los datos.	51
Figura 4.14 Evolución temporal de la utilización del disco (%util) durante las migraciones en vivo en el escenario MicroStack, diferenciando ambas direcciones de migración. Se muestra el valor medio con su variabilidad.	52
Figura 4.15 Evolución temporal de la utilización del disco (%util) durante las migraciones en vivo en el escenario OSM, diferenciando ambas direcciones de migración. Se representa el valor medio con su variabilidad.	52

1 INTRODUCCIÓN

La virtualización de recursos se ha consolidado como uno de los pilares fundamentales en las infraestructuras modernas de tecnologías de la información, permitiendo una utilización más eficiente del hardware, una mayor flexibilidad en la gestión de servicios y una reducción significativa de los costes operativos. En este contexto, las arquitecturas basadas en computación en la nube y en la virtualización de funciones de red (NFV) han adquirido un papel clave, especialmente en entornos de telecomunicaciones y en servicios con requisitos estrictos de disponibilidad y latencia.

Uno de los mecanismos más relevantes dentro de estos entornos virtualizados es la **migración en vivo de máquinas virtuales** (*live migration*), que permite trasladar una instancia en ejecución desde un nodo físico a otro sin necesidad de interrumpir el servicio ofrecido. Esta funcionalidad resulta esencial para tareas de mantenimiento, balanceo de carga o recuperación ante fallos. No obstante, durante el proceso de migración pueden producirse degradaciones temporales del rendimiento, como incrementos en la latencia del servicio o picos en el consumo de recursos del sistema, especialmente cuando la migración se realiza bajo carga.

En los últimos años, la evolución hacia arquitecturas de **edge computing** ha incrementado la relevancia de este tipo de mecanismos. En estos escenarios, los servicios se despliegan en infraestructuras próximas al usuario final con el objetivo de reducir la latencia y mejorar la calidad de experiencia. Tal y como se describe en la literatura especializada, el edge computing introduce restricciones adicionales en términos de recursos disponibles y sensibilidad al rendimiento, lo que hace que operaciones como la migración en vivo deban analizarse cuidadosamente para garantizar que no se compromete la calidad del servicio.

En paralelo, el uso de plataformas de orquestación en entornos NFV permite automatizar la gestión del ciclo de vida de los servicios virtualizados, introduciendo una capa adicional de abstracción y control sobre la infraestructura subyacente. Si bien estas plataformas aportan ventajas significativas en términos de gestión y escalabilidad, su utilización puede introducir un **overhead adicional** en determinadas operaciones, como la migración en vivo de máquinas virtuales. Resulta, por tanto, especialmente relevante analizar el impacto que tiene la inclusión de una capa de orquestación sobre el rendimiento de estos procesos frente a una gestión directa desde la infraestructura cloud.

El objetivo principal de este Trabajo Fin de Máster es **evaluar y cuantificar el overhead introducido por una plataforma de orquestación en los procesos de migración en vivo de máquinas virtuales**, comparándolo con el comportamiento de dichas migraciones cuando se gestionan de forma directa desde la infraestructura cloud. Para ello, se analiza el impacto de la migración en vivo sobre un servicio web desplegado en un entorno virtualizado, considerando métricas tanto a nivel de servicio como de consumo de recursos del sistema.

Con este fin, se ha diseñado un entorno experimental controlado basado en tecnologías de virtualización y orquestación, en el que se realizan migraciones en vivo de una máquina virtual bajo carga constante. El estudio compara dos modelos diferenciados: por un lado, la migración gestionada directamente por la plataforma cloud mediante mecanismos nativos; y por otro, la migración orquestada desde una capa superior de gestión del servicio. Los resultados obtenidos permiten analizar de forma objetiva el impacto de ambos escenarios y evaluar las implicaciones de introducir una capa adicional de virtualización en términos de rendimiento y calidad del servicio.



2 MARCO Y ARQUITECTURA DEL SISTEMA

2.1 Marco tecnológico

La evolución de las infraestructuras de tecnologías de la información ha estado marcada en los últimos años por la adopción generalizada de técnicas de virtualización, que permiten desacoplar los servicios del hardware físico subyacente. Este enfoque posibilita una utilización más eficiente de los recursos, una mayor flexibilidad en el despliegue de servicios y una simplificación de las tareas de gestión y mantenimiento de los sistemas.

En este contexto, las arquitecturas basadas en computación en la nube (*cloud computing*) han permitido consolidar y centralizar recursos de cómputo, almacenamiento y red, ofreciendo servicios escalables y dinámicos. Sin embargo, la creciente demanda de aplicaciones sensibles a la latencia y con requisitos estrictos de calidad de servicio ha impulsado la aparición de nuevos paradigmas de despliegue, entre los que destaca el **edge computing**.

El edge computing se basa en el despliegue de servicios y capacidades de procesamiento en infraestructuras situadas en proximidad al usuario final o a las fuentes de datos, reduciendo la dependencia de centros de datos centralizados. Permitiendo disminuir la latencia, optimizar el uso del ancho de banda y mejorar la calidad de experiencia del usuario, especialmente en aplicaciones de tiempo real o servicios críticos. No obstante, las infraestructuras de edge computing suelen presentar mayores restricciones en términos de recursos disponibles, lo que hace que la gestión eficiente de los mismos resulte especialmente relevante.

El paradigma de la **virtualización de funciones de red (NFV)** fue definido formalmente por el European Telecommunications Standards Institute (ETSI) en su marco arquitectónico MANO [1], estableciendo los componentes necesarios para la gestión y orquestación de servicios virtualizados. Mediante NFV, las funciones tradicionalmente implementadas sobre hardware dedicado se despliegan como instancias virtuales ejecutándose sobre infraestructuras cloud estándar. Este modelo permite una mayor flexibilidad, una reducción de costes y una gestión más ágil del ciclo de vida de los servicios.

Para facilitar la gestión de estos entornos virtualizados complejos, se emplean plataformas de **orquestación**, que automatizan el despliegue, la configuración y la operación de los

servicios virtualizados. Estas plataformas permiten tratar los servicios como entidades lógicas compuestas por uno o varios componentes virtuales, abstrayendo los detalles de la infraestructura subyacente y proporcionando mecanismos avanzados de control y automatización. Sin embargo, la introducción de estas capas de orquestación puede implicar un coste adicional en términos de consumo de recursos y tiempos de ejecución de determinadas operaciones.

Dentro de este marco, la **migración en vivo de máquinas virtuales** constituye un mecanismo esencial para garantizar la disponibilidad y la continuidad del servicio en entornos virtualizados. La posibilidad de trasladar una máquina virtual en ejecución entre nodos de cómputo permite realizar tareas de mantenimiento, balanceo de carga o recuperación ante fallos sin necesidad de interrumpir el servicio. No obstante, este proceso implica la transferencia de estados internos de la máquina virtual y puede generar un consumo adicional de recursos que afecte temporalmente al rendimiento del sistema.

En escenarios de edge computing, donde los servicios son especialmente sensibles a la latencia y a las variaciones de rendimiento, el impacto de una migración en vivo adquiere una relevancia aún mayor. Además, cuando estas migraciones se gestionan a través de plataformas de orquestación, se introduce una capa adicional de abstracción que puede influir en el comportamiento del proceso, tanto en términos de duración como de consumo de recursos.

Por todo ello, resulta fundamental analizar de forma sistemática el impacto que tienen las migraciones en vivo de máquinas virtuales sobre los servicios desplegados, así como evaluar el **overhead introducido por las plataformas de orquestación** frente a una gestión directa desde la infraestructura cloud. Este análisis permite determinar hasta qué punto la incorporación de una capa adicional de virtualización afecta al rendimiento del sistema y a la calidad del servicio, especialmente en entornos próximos al usuario final como los definidos por el paradigma de edge computing.

2.2 Migración en vivo de máquinas virtuales

La migración en vivo de máquinas virtuales (*live migration*) es un mecanismo fundamental en los entornos de virtualización modernos que permite trasladar una máquina virtual en ejecución desde un nodo físico origen a un nodo destino sin necesidad de interrumpir de forma explícita el servicio que ofrece. Esta técnica resulta

especialmente relevante en infraestructuras cloud y NFV, donde la disponibilidad del servicio y la continuidad operativa constituyen requisitos críticos.

El mecanismo de migración en vivo se encuentra ampliamente documentado en plataformas de virtualización basadas en KVM y OpenStack, donde se describe el procedimiento de pre-copy y stop-and-copy utilizado para minimizar el tiempo de interrupción del servicio [2], [3].

Desde un punto de vista técnico, la migración en vivo se basa en la transferencia progresiva del estado de ejecución de la máquina virtual, principalmente su memoria y el contexto de la CPU, desde el nodo origen al nodo destino. Este proceso se lleva a cabo de manera coordinada entre ambos nodos, permitiendo que la máquina virtual continúe atendiendo peticiones durante la mayor parte del proceso de migración.

El procedimiento de migración en vivo suele dividirse en varias fases diferenciadas. En una primera fase, conocida como *pre-copy*, el contenido de la memoria de la máquina virtual se copia desde el nodo origen al nodo destino mientras la máquina continua en ejecución. Durante esta fase, las páginas de memoria que se modifican son marcadas como “sucias” y se vuelven a transferir en iteraciones sucesivas, con el objetivo de reducir el volumen de memoria pendiente de sincronización.

Una vez que la cantidad de memoria pendiente es lo suficientemente reducida, se inicia la fase de *stop-and-copy*. En esta etapa, la ejecución de la máquina virtual se detiene de forma momentánea en el nodo origen, se transfieren las últimas páginas de memoria y el estado final de la CPU, y se reactiva la ejecución en el nodo destino. Aunque esta fase implica una breve interrupción, su duración suele ser lo suficientemente reducida como para resultar imperceptible para la mayoría de los servicios, siempre que el proceso esté correctamente gestionado.

En entornos donde no se dispone de almacenamiento compartido entre los nodos de la infraestructura, como es el caso del entorno experimental analizado en este trabajo, la migración en vivo debe incluir también la transferencia del almacenamiento asociado a la máquina virtual. Este tipo de migración, conocida como *block migration*, implica la copia de los bloques de disco de la instancia al nodo destino durante el proceso de migración. La inclusión de la transferencia de almacenamiento incrementa la complejidad del proceso y puede introducir un mayor consumo de recursos de entrada/salida, lo que hace especialmente relevante el análisis de su impacto sobre el rendimiento del sistema.

Durante la ejecución de una migración en vivo, el consumo de recursos del sistema puede experimentar variaciones significativas. El proceso de copia de memoria y almacenamiento genera carga adicional sobre la CPU, la red y el subsistema de disco de los nodos implicados, lo que puede derivar en picos temporales de utilización. En consecuencia, los servicios desplegados sobre la máquina virtual migrada pueden verse afectados por aumentos de latencia o variaciones en los tiempos de respuesta, especialmente cuando la migración se realiza bajo carga.

Este impacto resulta particularmente crítico en escenarios de *edge computing*, donde los servicios suelen estar sujetos a restricciones de recursos más estrictas y a requisitos de baja latencia. En estos entornos, incluso degradaciones temporales pueden tener un efecto perceptible en la calidad del servicio ofrecido al usuario final. Por este motivo, resulta esencial analizar de forma detallada el comportamiento de las migraciones en vivo y evaluar cómo distintos tipos de gestión influyen en su rendimiento.

En este contexto, la migración en vivo no debe considerarse únicamente como una operación infraestructural, sino como un proceso que puede verse influido por las capas de gestión y orquestación que intervienen en su ejecución. La introducción de plataformas de orquestación añade una capa adicional de control y abstracción sobre la infraestructura subyacente, lo que puede traducirse en un overhead adicional durante la ejecución de la migración. El análisis de este impacto constituye uno de los ejes centrales del presente trabajo y justifica la comparación entre migraciones gestionadas directamente desde la infraestructura cloud y aquellas orquestadas desde un nivel superior de gestión del servicio.

Sobre la base de los conceptos descritos, en el siguiente apartado se presenta la arquitectura del entorno experimental diseñado para analizar el impacto de las migraciones en vivo en un escenario controlado.

2.3 Arquitectura del sistema

La arquitectura empleada en este trabajo se basa en un entorno de laboratorio desplegado sobre una plataforma de virtualización Proxmox, diseñada para reproducir un escenario controlado de ejecución y migración de máquinas virtuales. Sobre esta infraestructura se han configurado varios nodos virtuales con funciones diferenciadas.

El sistema cuenta con **dos máquinas virtuales que ejecutan MicroStack**, las cuales actúan como nodos de la infraestructura cloud. Estas máquinas reciben la denominación de **Nodo Control** y **Node 1**, y desempeñan el rol de nodos de cómputo entre los que se realizan las migraciones en vivo de las instancias. MicroStack proporciona los mecanismos necesarios para administrar instancias y la ejecución de migraciones en vivo dentro del entorno OpenStack.

Adicionalmente, se han desplegado **dos máquinas virtuales auxiliares**. La primera de ellas aloja la plataforma de orquestación **OSM (Open Source MANO)**, utilizada para gestionar servicios virtualizados y ejecutar operaciones de migración desde un nivel de orquestación superior [5]. La segunda máquina virtual actúa como **generador de carga**, ejecutando la herramienta Apache Benchmark, desde la cual se envían peticiones HTTP al servicio desplegado durante las pruebas experimentales.

El servicio bajo estudio consiste en una **instancia virtual que ejecuta un servidor web**, desplegada inicialmente en uno de los nodos MicroStack. Durante la ejecución de las pruebas, esta instancia es sometida a carga continua desde la máquina de Apache Benchmark, mientras se realizan migraciones en vivo entre el Nodo Control y Node 1.

La arquitectura del entorno experimental se resume en la **Figura 2.1**, donde se distinguen los nodos de cómputo, el orquestador OSM y el generador de carga.

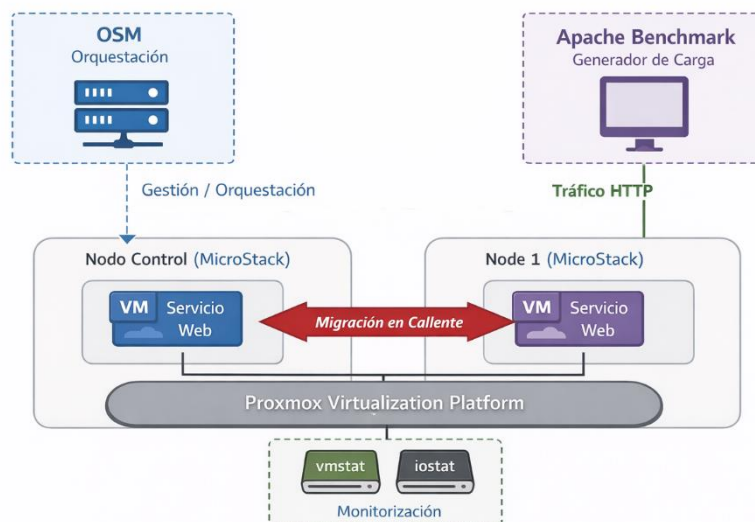


Figura 2.1 Arquitectura del entorno experimental para la evaluación de migraciones en vivo de máquinas virtuales.

2.4 Visualización de la topología y servicios desplegados

Con el fin de ilustrar el entorno experimental empleado en este trabajo, se muestran a continuación capturas de la topología de red de la infraestructura MicroStack y del estado de los servicios desplegados mediante la plataforma de orquestación OSM.

Estas representaciones permiten visualizar la organización de la red virtual utilizada durante los experimentos, así como verificar el correcto despliegue y ejecución del servicio gestionado por el orquestador.

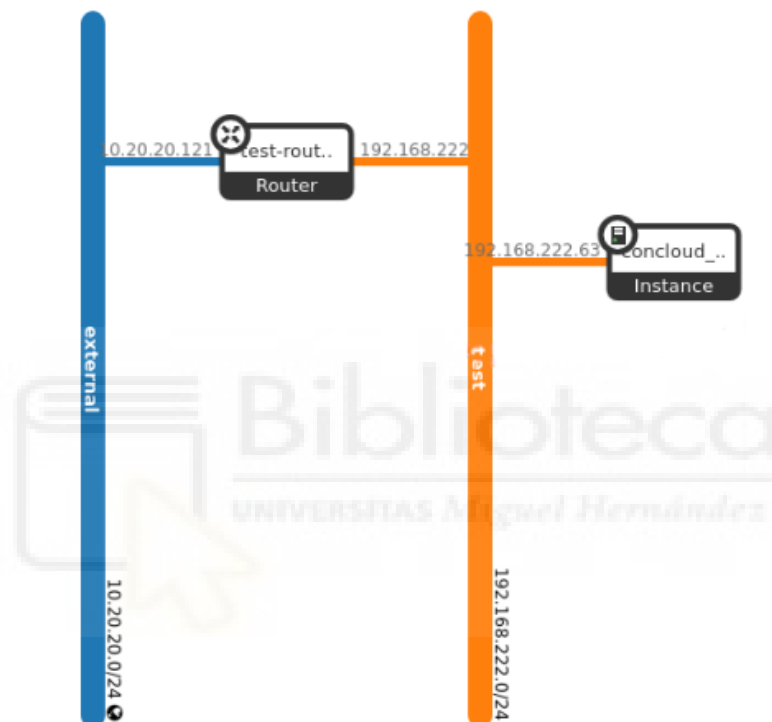


Figura 2.2 Topología de red del entorno MicroStack mostrando el router virtual, la red externa y la red interna utilizada durante los experimentos, así como la instancia del servicio web desplegada en la infraestructura cloud.

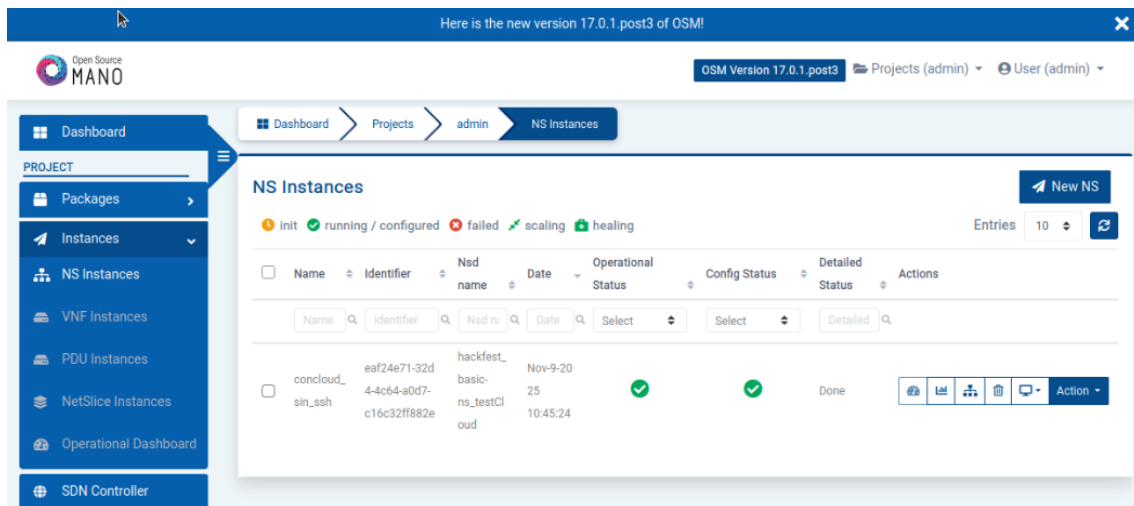


Figura 2.3 Vista de instancias de servicio (NS Instances) en la interfaz de OSM mostrando el servicio desplegado y en estado operativo durante la ejecución de los experimentos

2.5 Flujo de operación y escenarios de prueba

El proceso experimental seguido en este trabajo comienza con el despliegue de la instancia del servicio en uno de los nodos MicroStack. Una vez verificado el correcto funcionamiento del servicio, se inicia la generación de tráfico HTTP desde la máquina dedicada a Apache Benchmark, manteniendo una carga constante durante toda la prueba.

Con el servicio en funcionamiento y bajo carga, se procede a ejecutar una **migración en vivo de la máquina virtual** desde el nodo origen al nodo destino. Este proceso se realiza en dos direcciones: desde el Nodo Control hacia Node 1 y viceversa. Cada escenario se repite múltiples veces con el objetivo de obtener resultados representativos y reducir el efecto de posibles variaciones puntuales.

Durante la ejecución de las migraciones, se monitoriza el comportamiento del sistema mediante distintas herramientas. En los nodos de la infraestructura se recogen métricas de uso de CPU, memoria y disco utilizando las herramientas vmstat e iostat, mientras que la herramienta Apache Benchmark proporciona información sobre los tiempos de respuesta y el comportamiento del servicio desde el punto de vista del cliente.

El experimento se ha llevado a cabo utilizando **dos modelos distintos para la ejecución de las migraciones**. En el primer caso, las migraciones se realizan directamente desde MicroStack, utilizando los mecanismos nativos de la plataforma. En el segundo caso, las migraciones se ejecutan a través de la plataforma OSM, actuando como orquestador del servicio. Esta aproximación permite comparar ambos métodos y analizar cuál de ellos

ofrece un mejor comportamiento en términos de impacto sobre el servicio y consumo de recursos. En la figura 2.2 podemos apreciar el flujo de trabajo seguido para las pruebas realizadas:

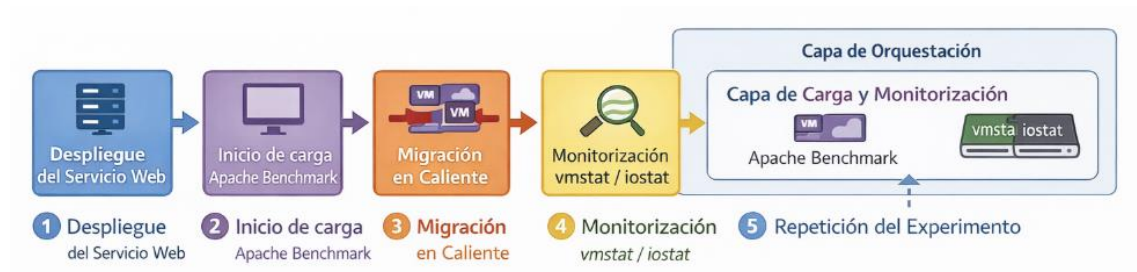


Figura 2.4 Flujo de operación seguido durante la ejecución de las pruebas

2.6 Componentes del sistema

2.6.1 Proxmox

Proxmox es una plataforma de virtualización de código abierto que integra tecnologías de virtualización basadas en KVM y contenedores, proporcionando una solución completa para la gestión de infraestructuras virtualizadas. Permite administrar máquinas virtuales, recursos físicos y redes desde una única plataforma, siendo ampliamente utilizada tanto en entornos académicos como profesionales.

La virtualización empleada en el entorno experimental se basa en KVM (Kernel-based Virtual Machine), una tecnología de virtualización integrada en el núcleo Linux ampliamente utilizada en entornos cloud [4].

En el contexto de este trabajo, Proxmox se emplea como la base de virtualización sobre la que se despliega todo el entorno experimental. Su función principal es alojar las distintas máquinas virtuales que conforman el sistema, garantizando la conectividad entre ellas y proporcionando un entorno estable y controlado para la ejecución de las pruebas. Gracias a Proxmox, es posible reproducir un escenario de laboratorio realista y fácilmente replicable.

2.6.2 MicroStack

MicroStack constituye una distribución ligera de OpenStack, basada en los servicios principales de la plataforma cloud definida por la Open Infrastructure Foundation [2]. Proporciona los servicios esenciales de OpenStack para la gestión de instancias, redes y almacenamiento, manteniendo una complejidad menor que las distribuciones completas.

En este trabajo, MicroStack se utiliza como núcleo de la infraestructura cloud, actuando como gestor de los nodos de cómputo y de las máquinas virtuales que alojan el servicio bajo estudio. A través de MicroStack se realizan las migraciones en vivo de las instancias entre los nodos Nodo Control y Node 1, lo que permite analizar el comportamiento de estos procesos desde la capa de infraestructura cloud, sin intervención de la orquestación

2.6.3 OSM (Open Source MANO)

OSM (Open Source MANO) es una plataforma de orquestación de código abierto alineada con las especificaciones ETSI MANO para la gestión del ciclo de vida de servicios virtualizados, implementando los componentes funcionales definidos en dicho marco arquitectónico [1], [5]. Su objetivo principal es automatizar el despliegue, la configuración y la gestión del ciclo de vida de los servicios virtualizados en entornos NFV.

Dentro del entorno experimental, OSM se emplea como capa de orquestación superior, permitiendo gestionar el servicio desplegado como una entidad lógica y no únicamente como una máquina virtual aislada. En este contexto, OSM se utiliza para desencadenar operaciones de migración en vivo desde el nivel de servicio, lo que posibilita comparar este escenario con la gestión directa proporcionada por MicroStack y evaluar el impacto de la orquestación en los procesos de migración.

2.6.4 Apache Benchmark

Apache Benchmark es una herramienta de generación de carga incluida en el servidor web Apache, diseñada para medir el rendimiento de servicios HTTP mediante el envío de un elevado número de peticiones concurrentes. Se utiliza habitualmente para evaluar la capacidad de respuesta y el comportamiento de servidores web bajo diferentes niveles de carga.

En este trabajo, Apache Benchmark se emplea como generador de tráfico hacia el servicio web desplegado en la máquina virtual objeto de estudio. La herramienta se ejecuta en una máquina virtual independiente, desde la cual se envían peticiones HTTP de forma continua durante la ejecución de las migraciones en vivo. Permitiendo simular el acceso de clientes reales al servicio y analizar el impacto de las migraciones sobre la calidad del servicio desde el punto de vista del usuario final.

2.6.5 Herramientas de monitorización del sistema

Para la observación del comportamiento del sistema durante las pruebas experimentales, se emplean herramientas de monitorización a nivel de sistema operativo, concretamente vmstat e iostat. Estas herramientas proporcionan información detallada sobre el uso de recursos del sistema, tales como CPU, memoria, entrada/salida de disco y actividad de procesos.

En el entorno experimental, estas herramientas se utilizan para recopilar métricas durante la ejecución de las migraciones en vivo, permitiendo analizar el impacto de los distintos modelos de gestión sobre el consumo de recursos de los nodos de la infraestructura. Los datos obtenidos a través de estas herramientas constituyen la base para el análisis comparativo presentado en capítulos posteriores.

2.7 Justificación del entorno experimental

El entorno experimental ha sido diseñado con el objetivo de equilibrar realismo, control y reproducibilidad. La utilización de un laboratorio virtualizado permite repetir los experimentos bajo condiciones similares y reducir la influencia de factores externos no controlados.

La separación de funciones en distintas máquinas virtuales (infraestructura, orquestación y generación de carga) evita interferencias entre componentes y permite obtener métricas más fiables durante la ejecución de las pruebas. Asimismo, la elección de MicroStack y OSM responde a su alineación con entornos cloud y NFV ampliamente utilizados tanto en el ámbito académico como industrial.

La realización de migraciones en vivo bajo carga representa un escenario exigente pero realista, en el que es posible evaluar el impacto de los distintos modelos de gestión sobre el rendimiento del servicio y el consumo de recursos del sistema.

2.8 Relación con los objetivos del trabajo

La arquitectura y el entorno experimental descritos en este capítulo han sido diseñados de forma explícita para dar respuesta a los objetivos planteados en este Trabajo Fin de Máster. La selección de las tecnologías empleadas, así como la organización de los distintos componentes del sistema, permite abordar de manera controlada el análisis del impacto de las migraciones en vivo de máquinas virtuales sobre servicios en ejecución.

En particular, la utilización de dos nodos de cómputo gestionados mediante MicroStack posibilita la realización de migraciones en vivo gestionadas directamente desde la infraestructura cloud, lo que permite evaluar el comportamiento nativo de este tipo de operaciones. De forma complementaria, la incorporación de la plataforma OSM introduce una capa de orquestación que permite gestionar las migraciones desde el nivel de servicio, alineándose con los principios de la virtualización de funciones de red y aportando un segundo modelo de gestión claramente diferenciable.

La separación funcional de los distintos componentes del sistema, incluyendo la infraestructura de cómputo, la plataforma de orquestación y el generador de carga, facilita la obtención de métricas objetivas y comparables durante la ejecución de los experimentos. Asimismo, la generación de tráfico constante hacia el servicio desplegado permite analizar el impacto de las migraciones en un escenario realista, en el que el servicio se mantiene activo y sometido a carga durante todo el proceso.

De este modo, la arquitectura propuesta no solo permite la ejecución de los experimentos definidos, sino que establece una base sólida para la comparación entre los distintos métodos de migración analizados. Esta relación directa entre el diseño del sistema y los objetivos del trabajo garantiza la coherencia del estudio y sienta las bases para el análisis de resultados presentado en los capítulos posteriores.

3 METODOLOGÍA

3.1 Configuración del entorno experimental

Con el objetivo de garantizar la reproducibilidad del estudio y facilitar la correcta interpretación de los resultados obtenidos, en este apartado se describe de forma detallada la configuración del entorno experimental empleado para la evaluación de las migraciones en vivo de máquinas virtuales. Se especifican tanto las características de la infraestructura utilizada como las versiones del software, la configuración de la instancia del servicio y las herramientas empleadas para la monitorización y automatización de las pruebas.

3.1.1 Infraestructura de virtualización

El entorno experimental se ha desplegado sobre una plataforma de virtualización basada en **Proxmox**, que actúa como hipervisor subyacente y aloja todas las máquinas virtuales necesarias para la ejecución del experimento. Sobre esta infraestructura se han definido cuatro máquinas virtuales principales, cada una con una función claramente diferenciada:

- Dos máquinas virtuales configuradas como nodos de cómputo, ejecutando **MicroStack**, denominadas *Nodo Control* y *Node 1*.
- Una máquina virtual dedicada a la plataforma de orquestación **OSM (Open Source MANO)**.
- Una máquina virtual destinada a la generación de carga mediante la herramienta **Apache Benchmark**.

Esta separación funcional permite aislar los distintos componentes del sistema y evitar interferencias entre la infraestructura cloud, la capa de orquestación y el proceso de generación de tráfico, contribuyendo a la obtención de métricas fiables durante la ejecución de las pruebas.

3.1.2 Versiones del software empleado

Las versiones del software utilizadas en el entorno experimental se detallan a continuación:

- **Proxmox VE:** versión 7.2-11
- **MicroStack (OpenStack):** snap microstack, basado en OpenStack Ussuri
- **OSM (Open Source MANO):** versión 17.0.1.post3
- **Sistema operativo de la instancia del servicio:** Ubuntu Server 18.04 LTS (bionic-server-cloudimg)
- **Apache Benchmark:** versión incluida en Apache HTTP Server 2.3
- **Sistema operativo de las máquinas auxiliares:** Ubuntu Server 18.04 LTS

La especificación de estas versiones resulta fundamental para garantizar la reproducibilidad del experimento, dado que el comportamiento de las migraciones en vivo puede verse influido por cambios en las implementaciones de las plataformas de virtualización y orquestación.

3.1.3 Configuración de la instancia del servicio

El servicio objeto de estudio se despliega sobre una máquina virtual gestionada por MicroStack, que aloja un servidor web accesible mediante peticiones HTTP. La instancia se crea a partir de una imagen base de **Ubuntu Server 18.04 LTS**, seleccionada por su estabilidad y compatibilidad con entornos cloud.

La configuración de recursos de la instancia se realiza mediante un *flavor* definido en MicroStack, que establece de forma explícita los siguientes parámetros (mantenidos constantes durante todas las pruebas):

- Número de vCPU: 1
- Memoria RAM: 1 GB
- Almacenamiento: 5 GB

La instancia se despliega a partir de una imagen cloud compatible con **cloud-init**. No obstante, en el entorno experimental no se empleó un fichero de inicialización personalizado mediante *user-data*. La configuración del servicio web se realizó manualmente tras el despliegue de la instancia, instalando y habilitando el servidor Apache mediante comandos ejecutados directamente en la máquina virtual. Los comandos utilizados para dicha configuración se incluyen en el **Anexo A**.

La creación de la instancia se lleva a cabo mediante la herramienta de línea de comandos de OpenStack proporcionada por MicroStack, utilizando un comando equivalente al siguiente:

```
microstack.openstack server create \  
  --flavor hackfest_basic-VM-vnf-1-flv \  
  --image bionic-server-cloudimg \  
  --network test \  
  <nombre_instancia>
```

Antes del inicio de cada conjunto de pruebas, se verifica el correcto arranque de la máquina virtual y el funcionamiento del servicio web, asegurando que la instancia se encuentra en condiciones operativas equivalentes en todos los escenarios experimentales.

3.1.4 Herramientas de monitorización del sistema

Para la monitorización del comportamiento del sistema durante la ejecución de las migraciones en vivo se emplean herramientas estándar a nivel de sistema operativo, ejecutadas directamente en los nodos de la infraestructura cloud.

El uso de CPU, memoria y actividad general del sistema se monitoriza mediante la herramienta **vmstat** [6], configurada para generar muestras con una periodicidad de un segundo. El comando utilizado es el siguiente:

```
vmstat 1
```

Este comando genera una salida en formato de texto que incluye métricas como el número de procesos en ejecución, el uso de memoria, la actividad de intercambio (*swap*) y la utilización de la CPU, permitiendo identificar variaciones asociadas al proceso de migración.

Para el análisis del comportamiento del subsistema de almacenamiento se emplea la herramienta **iostat** [7], también configurada con un intervalo de muestreo de un segundo, mediante el comando:

```
iostat -dx 1
```

La salida de esta herramienta proporciona métricas detalladas relacionadas con las operaciones de entrada/salida en disco, como tasas de lectura y escritura, tiempos de

espera y porcentaje de utilización del dispositivo, resultando especialmente relevante en un entorno donde se realizan migraciones en vivo con transferencia de almacenamiento (*block migration*).

Las salidas generadas por ambas herramientas se redirigen a ficheros de texto independientes para cada ejecución experimental, facilitando su posterior análisis y correlación con el proceso de migración.

3.1.5 Automatización y sincronización de las pruebas

Con el fin de garantizar la coherencia y repetibilidad de los experimentos, la generación de carga, la monitorización del sistema y el registro temporal de las migraciones se coordinan mediante un script de automatización desarrollado específicamente para este trabajo.

Este script permite iniciar de forma sincronizada las herramientas de monitorización, lanzar la generación de carga mediante Apache Benchmark y registrar el instante de inicio y finalización de cada migración en vivo. Asimismo, organiza automáticamente los ficheros de salida generados durante cada prueba, diferenciando los distintos escenarios experimentales y la dirección de las migraciones.

El uso de este mecanismo de automatización reduce la intervención manual durante la ejecución de las pruebas y minimiza posibles errores en la recogida de datos, contribuyendo a la fiabilidad de los resultados obtenidos. El código completo del script empleado se incluye en el Anexo B de este documento.

3.2 Diseño del experimento

El diseño experimental de este trabajo tiene como objetivo evaluar el impacto de las migraciones en vivo de máquinas virtuales sobre un servicio en ejecución, poniendo el foco en el **overhead introducido por una capa adicional de orquestación** frente a una gestión directa desde la infraestructura cloud. Para ello, se comparan dos modelos distintos para la ejecución de las migraciones: uno basado en la gestión directa desde la

infraestructura cloud, gestionado directamente por MicroStack, y otro orquestado, gestionado a través de la plataforma OSM.

El experimento se lleva a cabo sobre el entorno descrito en el apartado anterior, manteniendo constantes la infraestructura, la configuración de la instancia del servicio y los parámetros de generación de carga. Esta aproximación permite aislar el mecanismo de gestión de la migración como variable principal del estudio, garantizando que las diferencias observadas en los resultados se deban exclusivamente a la forma utilizada para ejecutar la migración en vivo.

El servicio objeto de estudio se despliega como una instancia gestionada por MicroStack que aloja un servidor web accesible mediante peticiones HTTP. Durante la ejecución de las pruebas, dicha instancia se somete a una carga continua generada desde una máquina virtual dedicada, mientras se realizan migraciones en vivo entre los nodos de cómputo *Nodo Control* y *Node 1*.

Cada uno de los modelos de migración analizados se ejecuta de forma repetida con el fin de obtener resultados estadísticamente representativos y reducir la influencia de posibles variaciones puntuales del sistema. En total, se realizan múltiples migraciones en vivo en ambos sentidos de migración, manteniendo idénticas condiciones de partida en todas las ejecuciones.

El diseño del experimento se ha concebido para reflejar un escenario exigente pero realista, en el que el servicio permanece activo y bajo carga durante todo el proceso de migración. De este modo, es posible analizar tanto el comportamiento del servicio desde el punto de vista del usuario final como el impacto de la migración en el consumo de recursos del sistema, permitiendo una comparación objetiva entre los escenarios analizados.

En la Figura 3.1 se ilustran las máquinas virtuales desplegadas sobre la plataforma Proxmox, mostrando la distribución de funciones entre los nodos de infraestructura, la plataforma de orquestación y el generador de carga.

Tipo ↑	Descripción	Uso de dis...	Memoria - ...	Uso de CPU	Tiempo de uso	Host CPU ...
node	asterix				-	
node	obelix				-	
qemu	115 (osm17v3)	0.0 %	90.5 %	9.2% of 4 ...	8 días 09:08:22	36.8% of 1...
qemu	116 (SergioBenchmark)	0.0 %	17.9 %	12.2% of 8 ...	8 días 08:22:33	97.7% of 1...
qemu	126 (testControl1)	0.0 %	48.7 %	1.1% of 8 ...	8 días 09:08:27	9.0% of 1C...
qemu	127 (testNode1)	0.0 %	10.4 %	10.9% of 8 ...	8 días 09:08:29	87.4% of 1...

Figura 3.1 Máquinas virtuales desplegadas en la plataforma Proxmox para la ejecución del experimento.

3.3 Escenarios de prueba

Con el objetivo de evaluar de forma comparativa el impacto de distintos modos de gestión de la migración en vivo de máquinas virtuales, se han definido dos escenarios de prueba claramente diferenciados. Ambos escenarios comparten la misma infraestructura, la misma configuración de la instancia del servicio y los mismos parámetros de generación de carga, de modo que las diferencias observadas en los resultados puedan atribuirse exclusivamente al mecanismo utilizado para gestionar la migración.

En todos los escenarios, el servicio web se despliega como una instancia gestionada por MicroStack y se somete a una carga continua de tráfico HTTP generada desde una máquina virtual dedicada. Durante la ejecución de las pruebas, la instancia se migra en vivo entre los nodos de cómputo *Nodo Control* y *Node 1*, manteniéndose constantes el resto de condiciones del entorno experimental descritas en el apartado 3.1.

La definición de escenarios diferenciados permite analizar el comportamiento de la migración en vivo desde dos perspectivas complementarias: una gestión directa desde la infraestructura cloud y una gestión mediada por una plataforma de orquestación. Esta aproximación resulta clave para cuantificar el **overhead introducido por la capa de orquestación** y evaluar su impacto sobre el rendimiento del servicio y el consumo de recursos del sistema.

3.3.1 Escenario A: Migración en vivo gestionada por MicroStack

En este escenario, la migración en vivo de la instancia del servicio se realiza directamente desde la infraestructura cloud utilizando los mecanismos nativos proporcionados por MicroStack. La operación de migración se ejecuta sobre la máquina virtual sin

intervención de capas superiores de orquestación, basado en la gestión directa de la máquina virtual desde la infraestructura cloud.

Este escenario se utiliza como referencia para caracterizar el comportamiento base de las migraciones en vivo en el entorno experimental. La gestión directa desde MicroStack permite observar el impacto de la migración cuando se ejecuta con la mínima capa de abstracción posible, sirviendo como punto de comparación frente al escenario gestionado mediante la plataforma de orquestación OSM.

Las migraciones se ejecutan de forma repetida en ambos sentidos (desde *Nodo Control* hacia *Node 1* y viceversa), manteniendo constante la configuración de la instancia, el nivel de carga aplicado y las herramientas de monitorización empleadas. Los resultados obtenidos en este escenario constituyen la base para evaluar el overhead introducido por la plataforma de orquestación en el escenario alternativo.

3.3.2 Escenario B: Migración en vivo orquestada mediante OSM

En el segundo escenario, la migración en vivo de la instancia del servicio se ejecuta a través de la plataforma de orquestación OSM, que gestiona el servicio como una entidad lógica dentro de su ciclo de vida. En este caso, la operación de migración se desencadena desde el nivel de servicio, alineándose con los principios de la virtualización de funciones de red y la gestión centralizada de servicios.

Desde el punto de vista funcional, la ejecución efectiva de la migración es delegada en la infraestructura cloud subyacente; sin embargo, la coordinación y el control del proceso se realizan desde la capa de orquestación. Esta aproximación introduce una capa adicional de abstracción frente al modelo basado en la gestión directa desde la infraestructura cloud, lo que puede traducirse en un incremento del consumo de recursos o en variaciones en el comportamiento temporal del proceso de migración.

Al igual que en el escenario anterior, las migraciones se ejecutan de forma repetida en ambos sentidos entre los nodos de cómputo, manteniendo idénticas condiciones de carga y monitorización. Esta homogeneidad en las condiciones experimentales permite comparar de forma directa los resultados obtenidos en ambos escenarios y analizar el impacto específico de la orquestación sobre la migración en vivo

3.4 Creación y configuración de la instancia del servicio

La instancia del servicio objeto de estudio se despliega como una máquina virtual gestionada por la plataforma MicroStack, ejecutándose sobre uno de los nodos de cómputo del entorno experimental. Dicha instancia aloja un servidor web que constituye el servicio sobre el cual se evalúa el impacto de las migraciones en vivo.

La creación de la instancia se realiza a partir de una imagen base de **Ubuntu Server 18.04 LTS**, adecuada para entornos cloud y ampliamente utilizada en escenarios de virtualización por su estabilidad y compatibilidad con las herramientas de gestión empleadas. La configuración detallada de la instancia, incluyendo la asignación de recursos de cómputo y el proceso de inicialización mediante *cloud-init*, se ha descrito previamente en el apartado 3.1, manteniéndose constante durante toda la ejecución de los experimentos.

Con el fin de garantizar la comparabilidad de los resultados obtenidos, la configuración de la máquina virtual no se modifica entre los distintos escenarios de prueba. La misma instancia se utiliza tanto en las migraciones en vivo gestionadas directamente por MicroStack como en aquellas orquestadas a través de la plataforma OSM, evitando así que cambios en la configuración puedan influir en las métricas recogidas.

La instancia se conecta a una red virtual común del entorno experimental, disponiendo de conectividad IP que permite el acceso desde la máquina virtual dedicada a la generación de carga mediante Apache Benchmark. Antes del inicio de cada conjunto de pruebas, se verifica el correcto funcionamiento del sistema operativo y del servidor web desplegado, asegurando que el servicio se encuentra operativo y accesible.

Este método permite aislar el proceso de migración en vivo como la variable principal del estudio, garantizando que las diferencias observadas en el rendimiento del servicio y en el consumo de recursos del sistema se deban exclusivamente al método de gestión de la migración y no a variaciones en la configuración de la instancia. La **Figura 3.2** muestra la instancia del servicio desplegada en el entorno OpenStack gestionado por MicroStack, evidenciando su estado operativo previo a la ejecución de las migraciones en vivo.

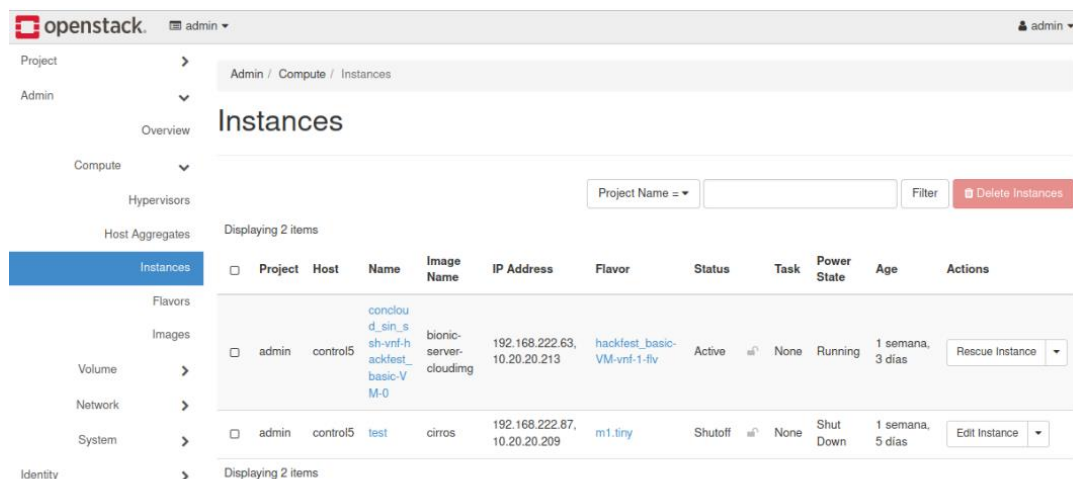
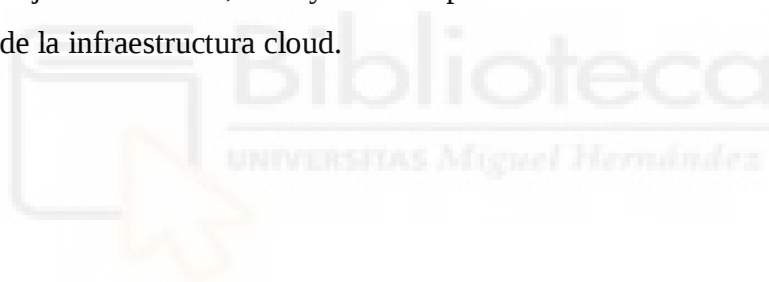


Figura 3.2 Instancia en entorno OpenStack.

En la **Figura 3.3** se presenta un detalle de la configuración de la instancia que aloja el servidor web objeto de estudio, incluyendo los parámetros de recursos asignados y su estado dentro de la infraestructura cloud.



```
ubuntu@controls:~$ microstack.openstack server show 5e52fb03-1934-4d90-b2c9-24a290b98744
```

Field	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-AZ:availability_zone	nova
OS-EXT-SRV-ATTR:host	controls
OS-EXT-SRV-ATTR:hypervisor_hostname	controls
OS-EXT-SRV-ATTR:instance_name	instance-00000003
OS-EXT-STS:power_state	Shutdown
OS-EXT-STS:task_state	None
OS-EXT-STS:vm_state	stopped
OS-SRV-USG:launched_at	2025-11-09T10:45:48.000000
OS-SRV-USG:terminated_at	None
accessIPv4	
accessIPv6	
addresses	test=192.168.222.63, 10.20.20.213
config_drive	
created	2025-11-09T10:45:31Z
flavor	hackfest_basic-VH-vnf-1-flv (c18728fe-47fe-4284-91f2-10ac2bab17b8)
hostId	984168575c938fd2c690f599112211bbc4839310674f64d161d58c44
id	5e52fb03-1934-4d90-b2c9-24a290b98744
image	blonic-server-cloudimg (df222c84-eab7-4c80-9c96-fd7d9cd28810)
key_name	None
name	concloud_sin_ssh-vnf-hackfest_basic-VH-0
project_id	2744885bc2024ee486788ce1dda7691f
properties	
security_groups	name='default'
status	SHUTOFF
updated	2025-12-02T08:46:34Z
user_id	920930666b594b73b33ce93d0e1878ac
volumes_attached	

Figura 3.3 Detalle de la instancia del servicio web desplegada.

3.5 Generación de carga

La generación de carga durante los experimentos se realiza mediante la herramienta **Apache Benchmark**, ejecutada desde una máquina virtual dedicada con el objetivo de simular el acceso concurrente de múltiples clientes al servicio web desplegado. El uso de una máquina independiente para la generación de tráfico permite aislar este proceso del

resto de componentes del sistema y evitar interferencias en las métricas recogidas durante las pruebas.

En cada ejecución experimental, Apache Benchmark envía un total de **1000 peticiones HTTP** al servicio, utilizando un nivel de **conurrencia de 10 conexiones simultáneas**. Estos parámetros se mantienen constantes en todos los escenarios de prueba con el fin de garantizar la comparabilidad de los resultados obtenidos. La elección de estos valores permite generar una carga suficiente para poner a prueba el comportamiento del servicio durante la migración en vivo, sin llegar a saturar de forma artificial la infraestructura subyacente.

La herramienta se ejecuta mediante un comando equivalente al siguiente:

```
ab -n 1000 -c 10 http://<ip_servicio>/
```

El servidor web desplegado en la instancia del servicio ofrece una página HTML sencilla utilizada como contenido de prueba durante los experimentos. Según la salida de Apache Benchmark, el documento servido tiene un tamaño de 10918 bytes. Este tamaño reducido permite que los tiempos de respuesta observados reflejen principalmente el comportamiento del servidor web y el impacto del proceso de migración en vivo, evitando que los resultados estén dominados por el tiempo de transferencia de grandes volúmenes de datos.

La generación de tráfico se inicia de forma coordinada con el proceso de migración en vivo, de manera que el servicio permanece bajo carga durante todo el intervalo en el que se lleva a cabo la migración. Permitiendo evaluar el impacto real de la migración sobre el rendimiento del servicio desde el punto de vista del usuario final, analizando posibles variaciones en los tiempos de respuesta o en la capacidad de atención de peticiones.

Los resultados proporcionados por Apache Benchmark incluyen métricas relacionadas con el número total de peticiones atendidas, los tiempos de respuesta y el comportamiento general del servicio bajo carga. Estas métricas se utilizan posteriormente para analizar el impacto de los distintos escenarios de la migración en vivo y para correlacionar el rendimiento del servicio con el consumo de recursos observado a nivel de sistema.

3.6 Ejecución de las migraciones

La ejecución de las migraciones en vivo de la instancia del servicio se realiza siguiendo dos modos diferenciados, con el objetivo de comparar el impacto de una gestión directa desde la infraestructura cloud frente a una gestión basada en orquestación. En ambos casos, la migración se lleva a cabo mientras el servicio permanece activo y sometido a carga continua generada mediante Apache Benchmark, tal y como se ha descrito en los apartados anteriores.

Para garantizar la comparabilidad de los resultados, las condiciones de partida se mantienen constantes en todas las ejecuciones experimentales: se utiliza la misma instancia del servicio, con idéntica configuración de recursos, conectividad de red y nivel de carga. Las diferencias entre los escenarios se limitan exclusivamente al mecanismo empleado para desencadenar y gestionar la migración en vivo.

3.6.1 Migración en vivo gestionada por MicroStack

En el escenario de migración gestionada directamente por MicroStack, la migración en vivo de la instancia del servicio se ejecuta mediante los mecanismos nativos proporcionados por OpenStack. La operación se inicia de forma manual desde el nodo de control, actuando directamente sobre la máquina virtual que aloja el servicio web.

Dado que el entorno experimental no dispone de almacenamiento compartido entre los nodos de cómputo, la migración se realiza utilizando un mecanismo de **migración en vivo con transferencia de almacenamiento (*block migration*)**, lo que implica la copia del contenido del disco de la máquina virtual desde el nodo origen al nodo destino durante el proceso de migración.

La ejecución de la migración se lleva a cabo mediante la herramienta de línea de comandos de OpenStack proporcionada por MicroStack, utilizando un comando equivalente al siguiente:

```
microstack.openstack server migrate \  
--live-migration concloud_sin_ssh-vnf-hackfest-basic-VM-0\  
--block-migration \  
--disk-overcommit
```

La opción `--live-migration` permite que la máquina virtual permanezca operativa durante todo el proceso, mientras que `--block-migration` habilita la transferencia del almacenamiento al nodo destino, eliminando la necesidad de un sistema de almacenamiento compartido. La opción `--disk-overcommit` permite una sobreasignación controlada de recursos de almacenamiento en el nodo destino, facilitando la ejecución de la migración en el entorno experimental.

Este modo representa un modelo de gestión basado en la infraestructura cloud, en el que la migración se ejecuta directamente sobre la máquina virtual sin intervención de la plataforma de orquestación. Las métricas recogidas durante estas pruebas se utilizan como referencia para evaluar el impacto de la migración en vivo cuando se gestiona sin la intervención de capas superiores de orquestación.

3.6.2 Migración en vivo orquestada mediante OSM

En el segundo escenario experimental, la migración en vivo de la instancia del servicio se ejecuta a través de la plataforma de orquestación **OSM (Open Source MANO)**, que gestiona el servicio como una entidad lógica dentro de su ciclo de vida. La máquina virtual no se considera un elemento aislado, sino que forma parte de un servicio virtualizado instanciado previamente y controlado desde el nivel de orquestación.

La **Figura 3.4** muestra la vista correspondiente a las instancias de servicio (*NS Instances*) en la interfaz web de OSM. En dicha vista se observa el servicio desplegado en estado operativo, correctamente configurado y bajo el control del orquestador. Esta vista confirma que la gestión del servicio se realiza desde una perspectiva lógica y centralizada, abstraída de los detalles específicos de la infraestructura cloud subyacente.

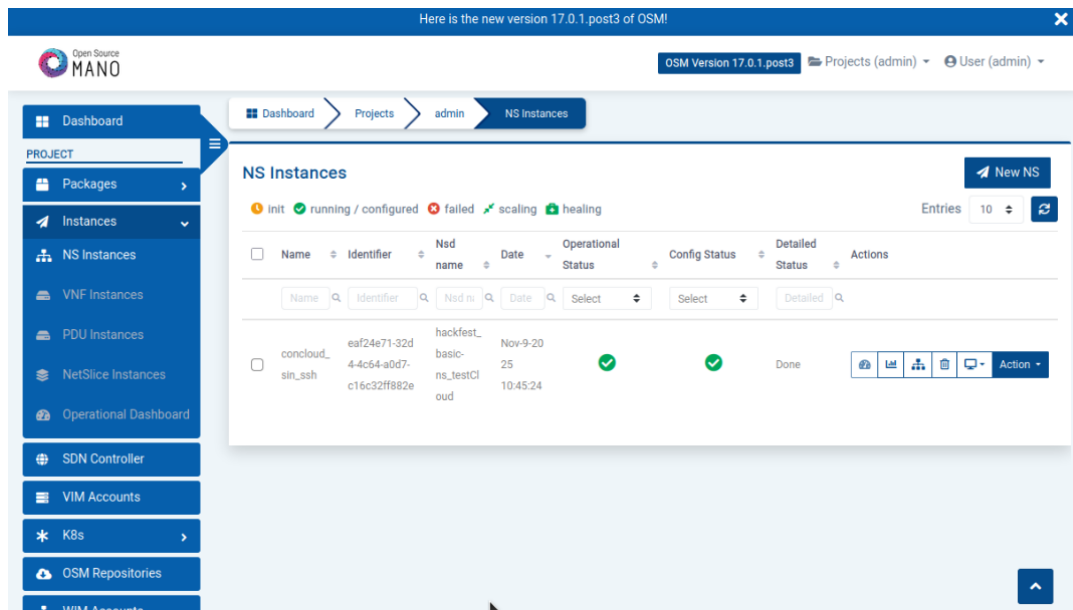


Figura 3.4 Instancia de servicio gestionada por OSM durante la ejecución de las migraciones orquestadas,

Desde esta vista, OSM permite ejecutar distintas acciones de gestión asociadas al ciclo de vida del servicio. Tal y como se muestra en la **Figura 3.5**, entre estas acciones se encuentra la opción de **migración de la máquina virtual** asociada al servicio. La migración se desencadena seleccionando dicha acción desde el menú de operaciones del servicio, lo que permite iniciar el proceso de migración en vivo desde el nivel de servicio y no directamente desde la infraestructura.

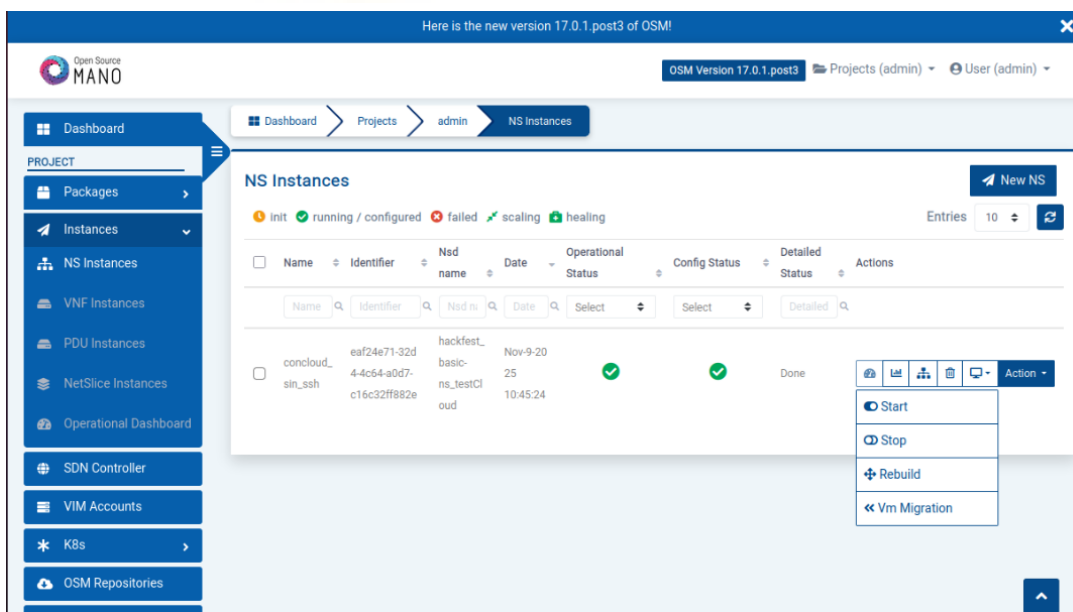


Figura 3.5 Ejecución de una migración en vivo desde la interfaz de OSM a nivel de servicio.

OSM actúa como una **capa de control superior**, encargada de coordinar la operación de migración con la infraestructura cloud gestionada por MicroStack. Aunque la ejecución efectiva de la migración en vivo es finalmente delegada en la plataforma cloud subyacente, el proceso se inicia y supervisa desde el orquestador, introduciendo una capa adicional de abstracción y control frente a la gestión directa desde la infraestructura cloud.

La inclusión de esta capa de orquestación puede implicar un **overhead adicional** en el proceso de migración, derivado de las operaciones de gestión, coordinación y comunicación entre OSM y la infraestructura cloud. El análisis de este posible impacto constituye uno de los objetivos principales del presente trabajo, motivo por el cual este escenario se compara directamente con la migración gestionada de forma nativa por MicroStack.

Al igual que en el escenario infraestructural, las migraciones en vivo orquestadas mediante OSM se ejecutan de forma repetida en ambos sentidos entre los nodos de cómputo, manteniendo constantes las condiciones de carga y monitorización. Esta homogeneidad permite comparar de forma objetiva el comportamiento de ambos escenarios y evaluar el impacto específico de la orquestación sobre el rendimiento del servicio y el consumo de recursos del sistema.

3.6.3 Coordinación con la generación de carga y monitorización

En ambos escenarios de migración, la ejecución del proceso se coordina de forma precisa con la generación de carga y la recogida de métricas. La carga de tráfico HTTP se mantiene activa durante toda la duración de la migración, mientras que las herramientas de monitorización registran el comportamiento del sistema desde el inicio hasta la finalización del proceso.

El instante de inicio y finalización de cada migración en vivo se registra de forma explícita, permitiendo medir la duración total del proceso y correlacionarla con las métricas de rendimiento del servicio y consumo de recursos obtenidas. Este procedimiento se repite de forma sistemática en todas las ejecuciones experimentales, garantizando la coherencia y comparabilidad de los datos recogidos.

3.7 Monitorización y recogida de métricas

Con el fin de analizar el impacto de las migraciones en vivo sobre el rendimiento del servicio y el consumo de recursos del sistema, se ha definido un mecanismo de monitorización que permite recopilar métricas relevantes durante la ejecución de cada prueba experimental. La recogida de métricas se realiza de forma simultánea a la generación de carga y a la ejecución de las migraciones, proporcionando una visión completa del comportamiento del sistema a lo largo de todo el proceso.

La monitorización se lleva a cabo mediante herramientas a nivel de sistema operativo, ejecutadas directamente en los nodos de la infraestructura cloud. Con el cual podemos observar de manera directa el consumo de recursos asociado a la migración de la máquina virtual, evitando dependencias de mecanismos externos de monitorización que puedan introducir interferencias adicionales.

3.7.1 Monitorización del uso de CPU y memoria

Para la monitorización del uso de CPU, memoria y actividad general del sistema se emplea la herramienta **vmstat**, configurada para generar muestras con una periodicidad de un segundo. Esta herramienta proporciona información relevante sobre la carga del sistema, el estado de la memoria, la actividad de intercambio (*swap*) y la utilización de la CPU durante la ejecución de las migraciones en vivo, tal y como se describe en su documentación oficial (manual del sistema Linux) [6].

El uso de **vmstat** permite identificar variaciones en el comportamiento del sistema asociadas al inicio y desarrollo del proceso de migración, como picos de carga en la CPU o cambios en el número de procesos en ejecución. Estas métricas resultan especialmente útiles para analizar el impacto de la migración en vivo sobre el rendimiento global del nodo de cómputo y detectar posibles efectos derivados de la introducción de una capa adicional de orquestación.

3.7.2 Monitorización de operaciones de entrada/salida en disco

El comportamiento del subsistema de almacenamiento se analiza mediante la herramienta **iostat**, también configurada con un intervalo de muestreo de un segundo. Esta herramienta proporciona métricas detalladas relacionadas con las operaciones de

entrada/salida en disco, incluyendo tasas de lectura y escritura, tiempos de espera y nivel de utilización del dispositivo de almacenamiento, según se especifica en su página manual del sistema Linux [7].

Dado que las migraciones en vivo realizadas en el entorno experimental incluyen la transferencia del almacenamiento de la máquina virtual (*block migration*), la monitorización de las operaciones de entrada/salida resulta especialmente relevante. Las métricas obtenidas permiten evaluar el impacto de este proceso sobre el rendimiento del sistema y analizar posibles diferencias entre los modos de gestión de la migración.

3.7.3 Métricas de rendimiento del servicio

Desde el punto de vista del servicio, el rendimiento se evalúa mediante los resultados proporcionados por la herramienta Apache Benchmark, ejecutada desde una máquina virtual dedicada. Esta herramienta ofrece métricas relacionadas con los tiempos de respuesta, el número de peticiones atendidas y el comportamiento general del servicio web bajo carga.

La combinación de estas métricas con las obtenidas a nivel de sistema permite correlacionar el comportamiento interno de la infraestructura con la percepción del servicio desde el punto de vista del cliente.

3.7.4 Organización y almacenamiento de los datos experimentales

Todas las métricas recogidas durante la ejecución de las pruebas se almacenan de forma estructurada y separada para cada escenario experimental. Para cada migración se generan ficheros independientes que contienen las salidas de las herramientas de monitorización, los resultados del benchmark y la duración total del proceso de migración.

Esta organización facilita el posterior tratamiento de los datos y permite realizar comparaciones directas entre los distintos escenarios de migración analizados. Asimismo, la repetición de las pruebas un total de 20 veces por cada escenario permite reducir la influencia de variaciones puntuales y obtener resultados representativos para el análisis presentado en los capítulos posteriores.

3.7.5 Automatización y sincronización del proceso experimental

Con el objetivo de garantizar la coherencia y repetibilidad de las pruebas, la generación de carga, la monitorización del sistema y el registro temporal de las migraciones se coordinan mediante un script de automatización desarrollado específicamente para este trabajo. Este script permite sincronizar el inicio de las herramientas de monitorización y del benchmark con el proceso de migración, así como registrar de forma precisa la duración total de cada prueba.

Durante cada ejecución, el script gestiona la puesta en marcha de las herramientas **vmstat** e **iostat**, lanza la generación de carga mediante Apache Benchmark y organiza automáticamente los ficheros de salida asociados a cada escenario experimental. Asimismo, permite diferenciar las migraciones realizadas en cada dirección, facilitando el posterior análisis comparativo de los resultados.

El uso de este mecanismo de automatización reduce la intervención manual durante la ejecución de las pruebas y minimiza posibles errores en la recogida de datos, contribuyendo a la fiabilidad de los resultados obtenidos. El código completo del script empleado se incluye en el **Anexo B** de este documento.

4 RESULTADOS

En este capítulo se presentan los resultados obtenidos tras la ejecución de 20 migraciones en vivo bajo cada uno de los modelos analizados: gestión directa mediante MicroStack y gestión orquestada mediante OSM.

Para cada métrica se representan las medias acompañadas de sus correspondientes intervalos de confianza al 95%, con el objetivo de evaluar la consistencia y variabilidad de los resultados y aportar rigor estadístico al análisis comparativo.

El intervalo de confianza al 95% se ha calculado mediante la expresión:

$$IC = \bar{x} \pm t_{0.975, n-1} \cdot \frac{s}{\sqrt{n}}$$

donde:

- $\bar{x}$ es la media muestral,
- s es la desviación estándar,
- $n = 20$ el número de repeticiones,
- $t_{0.975, n-1}$ el valor de la distribución t de Student para un nivel de confianza del 95%.

Este método permite estimar la precisión de las medias obtenidas y valorar la estabilidad de los resultados experimentales.

4.1 Rendimiento del servicio (Apache Benchmark)

El rendimiento del servicio web durante la ejecución de las migraciones se ha evaluado mediante la herramienta Apache Benchmark, considerando dos métricas principales: el tiempo total de ejecución del test y el tiempo medio total de conexión (*Connection Time – Mean Total*).

4.1.1 Tiempo total de ejecución del test

La Figura 4.1 muestra el tiempo total necesario para completar las 1000 peticiones HTTP para ambos escenarios.

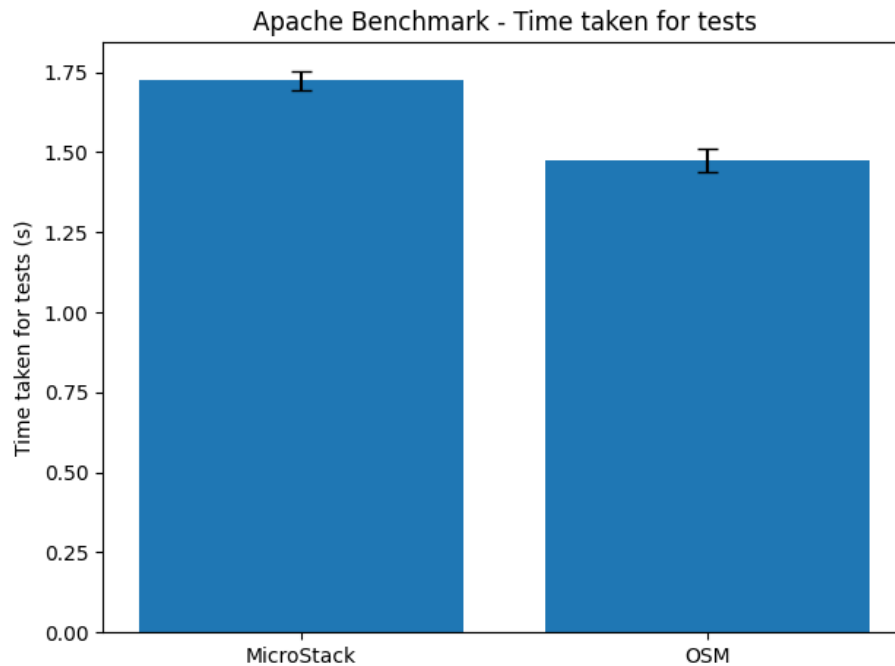


Figura 4.1 Gráfica comparativa “Apache Benchmark” – tiempo total de ejecución del test

Los valores medios obtenidos son:

- **MicroStack:** 1.724 s ± 0.031 (IC 95%)
- **OSM:** 1.475 s ± 0.037 (IC 95%)

Se observa que el tiempo total de ejecución del benchmark es ligeramente inferior en el escenario gestionado mediante OSM. La diferencia absoluta es de aproximadamente 0.25 segundos, lo que representa una mejora relativa cercana al 14 %. Esta diferencia indica que, en promedio, el conjunto de 1000 peticiones HTTP se completa en un tiempo ligeramente menor cuando la migración se gestiona mediante la plataforma de orquestación.

Aunque los intervalos de confianza presentan un solapamiento reducido, la diferencia es pequeña en términos absolutos y no implica una degradación del servicio en ninguno de los escenarios. Desde el punto de vista del usuario final, el rendimiento del servicio web se mantiene estable durante la migración, independientemente del método utilizado.

Este resultado indica que la inclusión de la capa de orquestación no penaliza el rendimiento global del servicio HTTP bajo carga.

4.1.2 Tiempo medio total de conexión

La Figura 4.2 presenta el tiempo medio total de conexión reportado por Apache Benchmark. Esta métrica corresponde al tiempo total durante el cual cada conexión TCP permanece activa, desde el establecimiento de la conexión hasta la recepción completa de la respuesta del servidor y el cierre de la misma.

Por tanto, este valor incluye no solo el tiempo de establecimiento de la conexión (*Connect*), sino también el tiempo de procesamiento de la petición y de espera de la respuesta (*Processing* y *Waiting*), constituyendo una medida global del tiempo de servicio percibido por el cliente para cada petición HTTP.

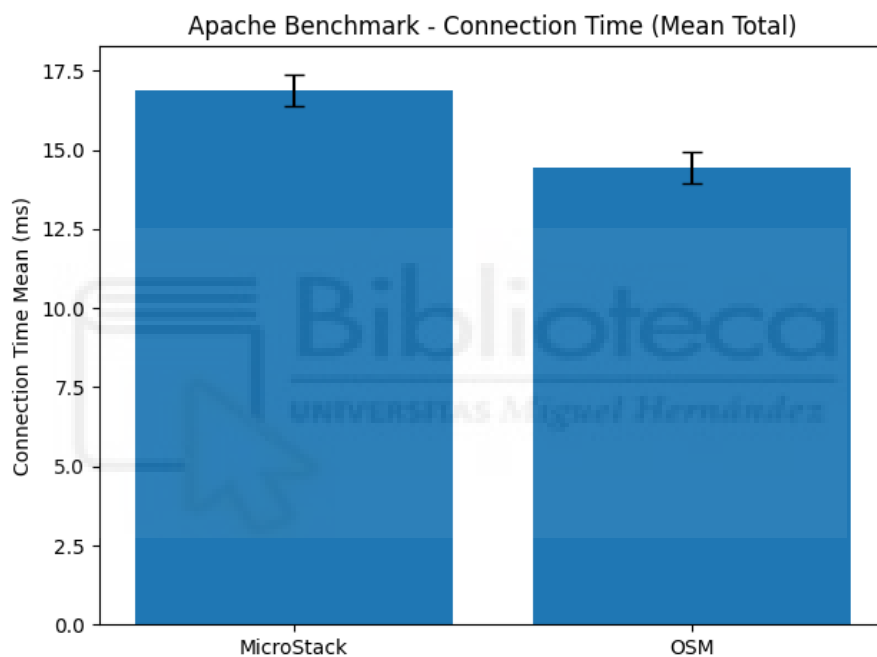


Figura 4.2 Gráfica comparativa tiempo medio total de conexión

Los valores medios obtenidos son:

- **MicroStack:** 16.9 ms
- **OSM:** 14.45 ms

Se aprecia nuevamente una ligera mejora en el escenario gestionado mediante OSM. La diferencia relativa es aproximadamente del 14%, manteniéndose ambos valores dentro de rangos bajos de latencia.

Este resultado confirma que, pese al mayor tiempo total del proceso de migración observado posteriormente en el escenario orquestado, la calidad de servicio percibida por el cliente HTTP no se ve degradada.

En consecuencia, la inclusión de la capa de orquestación no introduce penalizaciones apreciables en el rendimiento del servicio web bajo carga.

4.1.3 Análisis detallado de tiempos de conexión

Apache Benchmark proporciona un desglose detallado de los tiempos asociados a cada petición HTTP a través de la sección *Connection Times*, que incluye las métricas *Connect*, *Processing*, *Waiting* y *Total*. Estas métricas permiten analizar las distintas fases del ciclo de vida de una petición HTTP, desde el establecimiento de la conexión TCP hasta la recepción completa de la respuesta.

A partir de las salidas de Apache Benchmark obtenidas en las 20 ejecuciones realizadas en cada escenario experimental, se ha calculado el valor medio de estas métricas con el fin de comparar el comportamiento del servicio durante las migraciones gestionadas directamente por MicroStack y aquellas orquestadas mediante OSM.

La Tabla 4.1 muestra el agregado de los valores medios obtenidos para cada métrica.

El tiempo de conexión (*Connect*) reportado por Apache Benchmark corresponde al tiempo necesario para establecer la conexión TCP entre el cliente y el servidor, incluyendo el proceso de *three-way handshake* (protocolo de tres vías). Esta métrica es diferente del tiempo total de conexión analizado en el apartado 4.1.2, que representa el tiempo completo asociado a la petición HTTP.

En los resultados obtenidos, el valor de *Connect* es de 0 ms en ambos escenarios. Este valor no implica que el tiempo de establecimiento de la conexión sea nulo, sino que se encuentra por debajo de la resolución de medida de Apache Benchmark, que presenta los tiempos en milisegundos enteros. Dado que el cliente y el servidor se encuentran dentro del mismo entorno virtualizado y conectados mediante una red local de muy baja latencia, el tiempo real de establecimiento de la conexión es inferior a 1 ms, por lo que la herramienta lo redondea a 0 ms.

MÉTRICA	MICROSTACK	OSM
Connect	0 ms	0 ms
Processing	16.50 ms	14.00 ms
Waiting	15.95 ms	13.65 ms
Total	16.90 ms	14.45 ms

Tabla 4.1. Valores medios de los tiempos de conexión reportados por Apache Benchmark

La Figura 4.3 muestra la comparación de los tiempos medios de las distintas fases de una petición HTTP reportadas por Apache Benchmark para ambos escenarios experimentales. En ella se representan las métricas *Connect*, *Processing*, *Waiting* y *Total*, calculadas como el valor medio de las 20 ejecuciones realizadas en cada entorno.

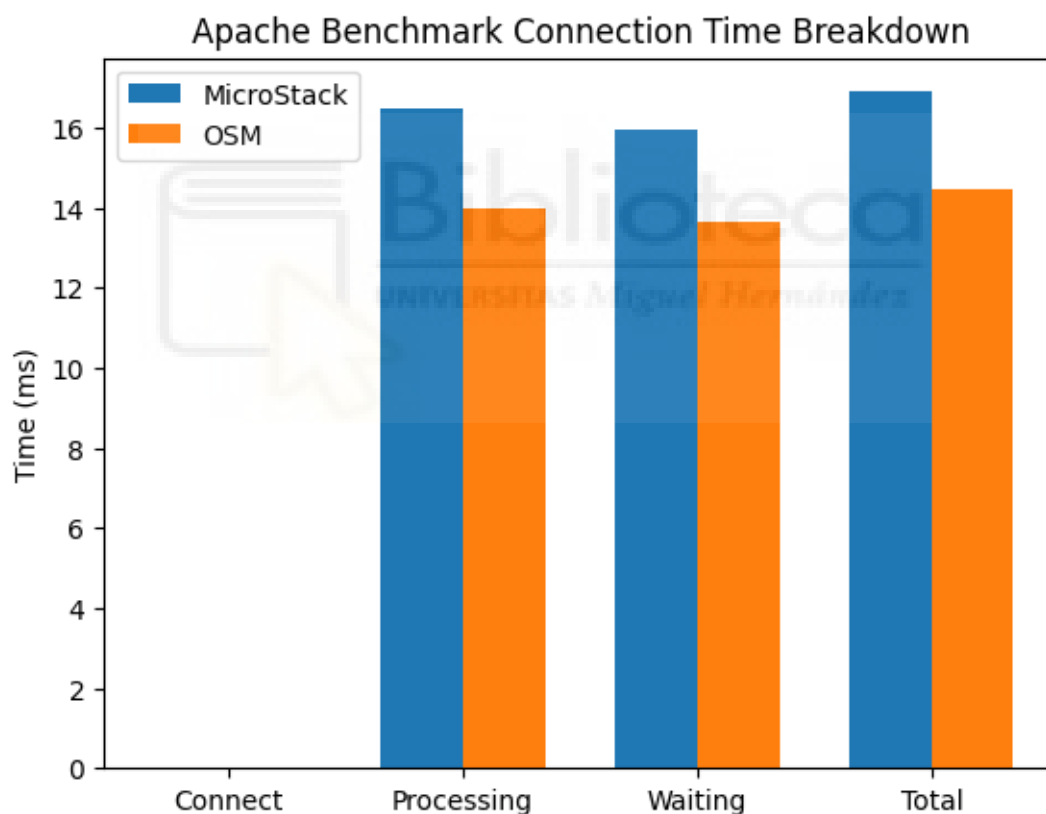


Figura 4.3 Comparación de los tiempos medios de *Connect*, *Processing*, *Waiting* y *Total* obtenidos con Apache Benchmark para los escenarios MicroStack y OSM.

Como puede observarse, los valores correspondientes a *Connect* son prácticamente nulos en ambos escenarios debido a que el cliente y el servidor se encuentran dentro del mismo entorno virtualizado y conectados mediante una red local de muy baja latencia. En cuanto

al resto de métricas, se observa que los tiempos de *Processing*, *Waiting* y *Total* presentan valores ligeramente inferiores en el escenario gestionado mediante OSM, aunque las diferencias se mantienen dentro de un rango reducido y no representan variaciones significativas desde el punto de vista del rendimiento del servicio.

4.2 Tiempo de migración

La duración total del proceso de migración constituye una de las métricas fundamentales para evaluar el overhead introducido por la plataforma de orquestación. La Figura 4.4 muestra la duración media del proceso completo de migración en vivo para ambos modelos.

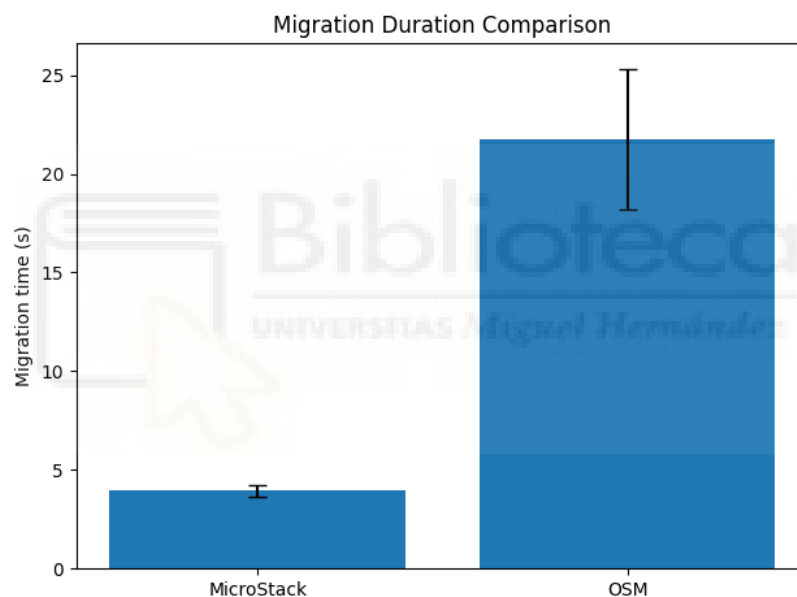


Figura 4.4 Gráfica comparativa tiempo de migración

Los valores medios obtenidos son:

- **MicroStack:** 3.95 s ± 0.28 (IC 95%)
- **OSM:** 21.75 s ± 3.58 (IC 95%)

La migración orquestada mediante OSM presenta una duración significativamente superior. En términos relativos, el incremento es aproximadamente:

$$\frac{21.75 - 3.95}{3.95} \times 100 \approx 451\%$$

Esto confirma la existencia de un **overhead considerable introducido por la capa de orquestación**, que multiplica por más de cinco la duración total del proceso de migración.

Sin embargo, como se ha mostrado en el apartado anterior, este incremento en el tiempo total del procedimiento no se traduce en una degradación perceptible del rendimiento del servicio HTTP.

4.2.1 Evolución del tiempo de migración

Con el fin de analizar si existe alguna tendencia en la duración de las migraciones a medida que se ejecutan múltiples operaciones consecutivas, se ha representado la evolución del tiempo de migración para cada una de las ejecuciones realizadas en ambos escenarios experimentales.

La Figura 4.5 muestra el tiempo registrado en cada una de las migraciones ejecutadas, tanto para el escenario gestionado directamente mediante MicroStack como para el escenario orquestado mediante OSM. Esta representación permite observar la variabilidad entre ejecuciones y evaluar si existe una tendencia de incremento del tiempo de migración a lo largo de la serie de experimentos.

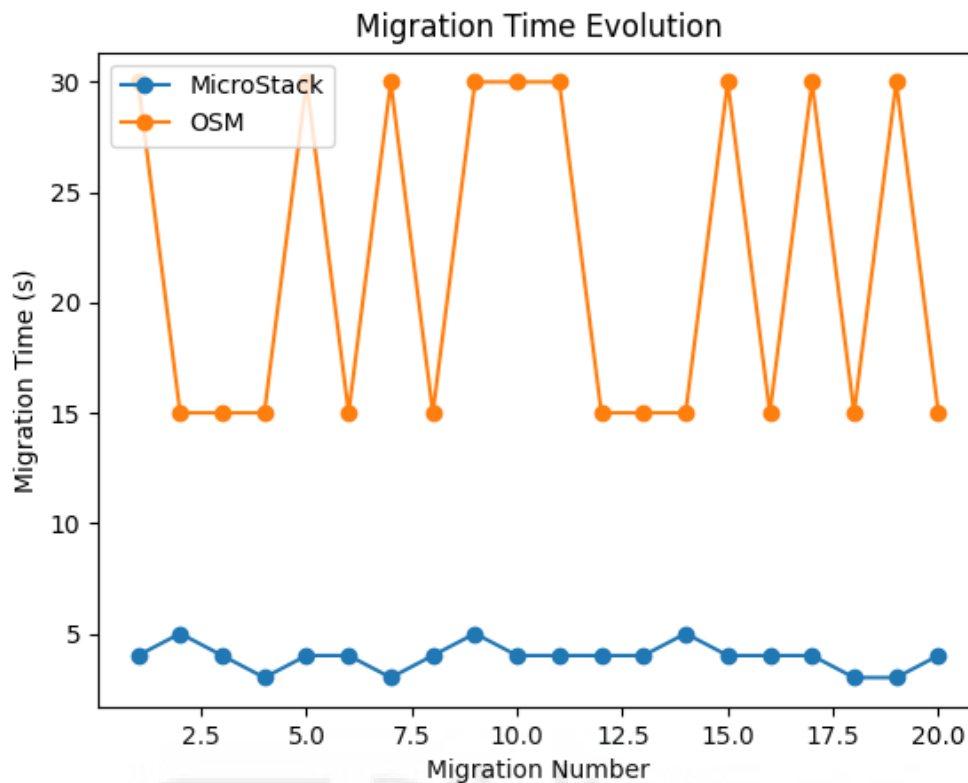


Figura 4.5 Evolución del tiempo de migración para cada una de las 20 migraciones ejecutadas en los escenarios MicroStack y OSM.

Como puede observarse, no se aprecia una tendencia clara de crecimiento del tiempo de migración a medida que se realizan más migraciones. No obstante, sí se observa una mayor variabilidad en el escenario orquestado mediante OSM, lo cual es coherente con la mayor complejidad del proceso de migración cuando interviene la capa de orquestación.

Adicionalmente, en el escenario gestionado mediante OSM, la evolución temporal del tiempo de migración presenta saltos discretos, con valores agrupados en intervalos aproximados de 15 segundos. Este comportamiento sugiere la existencia de mecanismos internos de temporización en la plataforma de orquestación, que realizan comprobaciones periódicas del estado de la migración en lugar de un seguimiento continuo del proceso.

Como consecuencia, la métrica de tiempo de migración obtenida a través de OSM presenta una menor granularidad temporal en comparación con la gestionada directamente por MicroStack, donde el tiempo se registra de forma más precisa y continua. Esto implica que la capa de orquestación introduce un efecto adicional en la

medición del tiempo, que no refleja únicamente la duración real del proceso de migración, sino también el comportamiento interno de los mecanismos de control de la plataforma.

No obstante, este efecto no se traduce en una degradación perceptible del rendimiento del servicio HTTP, tal y como se ha observado en los resultados obtenidos mediante Apache Benchmark.

4.3 Impacto en el consumo de recursos del sistema

Con el fin de analizar el comportamiento interno de la infraestructura durante las migraciones, se han estudiado métricas significativas obtenidas mediante las herramientas vmstat e iostat.

4.3.1 Uso de CPU (vmstat)

Se ha analizado el valor máximo del parámetro sy (CPU en modo sistema) durante cada migración.

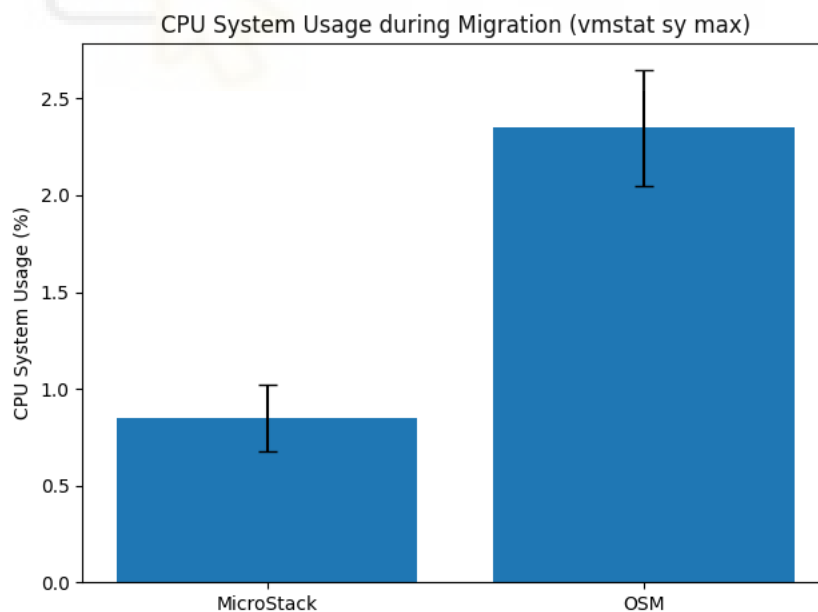


Figura 4.6 Gráfica comparativa Uso del sistema de CPU durante la migración

Los valores medios obtenidos son:

- **MicroStack:** 0.85 %
- **OSM:** 2.35 %

El escenario orquestado presenta un mayor consumo de CPU en modo sistema, lo que es coherente con la existencia de operaciones adicionales de coordinación y gestión entre OSM y la infraestructura cloud subyacente.

Aunque el incremento es proporcionalmente significativo, el valor absoluto continua siendo bajo, por lo que no supone una saturación del nodo de cómputo.

4.3.2 Utilización de disco (iostat)

Dado que las migraciones se realizan mediante *block migration*, el subsistema de almacenamiento juega un papel relevante. Se ha analizado el valor máximo de %util durante cada migración.

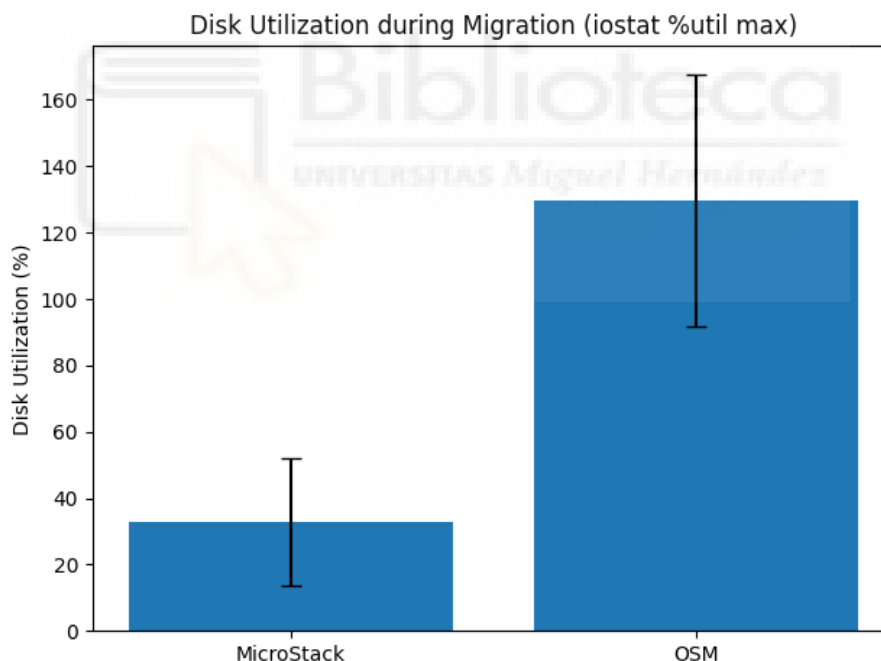


Figura 4.7 Gráfica comparativa Utilización del disco durante la migración

Los valores medios obtenidos son:

- **MicroStack:** 32.85 %
- **OSM:** 129.82 %

El escenario orquestado muestra picos de utilización de disco considerablemente superiores. El valor superior al 100% observado en el escenario OSM se debe a la

naturaleza virtualizada del entorno experimental, donde el porcentaje de utilización puede reflejar tiempos acumulados de espera en colas de entrada/salida y no exclusivamente uso físico directo del dispositivo.

Este resultado sugiere una mayor intensidad en las operaciones de transferencia o una coordinación menos eficiente del proceso de copia de almacenamiento cuando la migración es gestionada desde la capa de orquestación.

En términos relativos, el incremento es cercano al 295%, lo que confirma que el overhead de OSM se manifiesta de forma especialmente significativa en el subsistema de almacenamiento.

4.3.3 Evolución temporal de las métricas del sistema

Con el fin de analizar el comportamiento dinámico del sistema durante el proceso de migración, se han representado también las métricas capturadas mediante las herramientas **vmstat** e **iostat** en función del tiempo de observación. Dado que ambas herramientas se ejecutaron con un intervalo de muestreo de un segundo, cada punto de las gráficas representa el estado del sistema en un instante concreto durante la ejecución de la migración.

A diferencia de las representaciones basadas en una única ejecución, en este caso las gráficas muestran la evolución temporal media de las métricas a partir del conjunto de migraciones realizadas en cada escenario experimental. Esta aproximación permite obtener una visión más representativa del comportamiento del sistema, reduciendo la influencia de variaciones puntuales.

Estas representaciones permiten observar la evolución temporal del consumo de recursos del sistema y detectar posibles picos de utilización asociados al proceso de migración de la máquina virtual. En particular, permiten visualizar los momentos en los que se produce una mayor actividad de CPU o del subsistema de almacenamiento durante la transferencia de memoria y disco asociada a la migración en vivo.

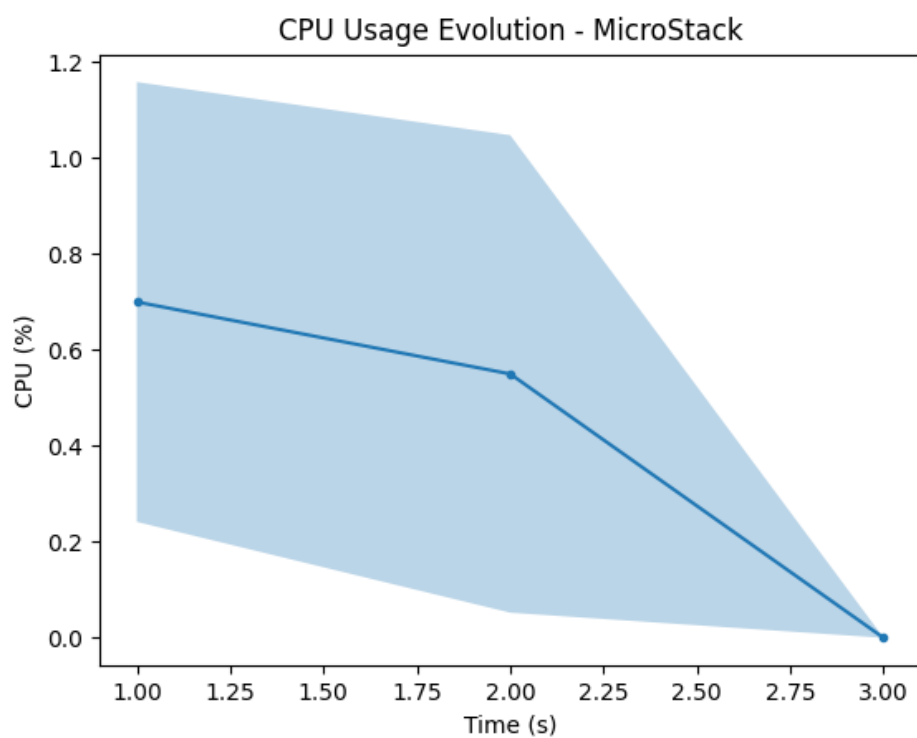


Figura 4.8 Evolución temporal del uso de CPU en modo sistema durante las migraciones en vivo en el escenario MicroStack. Se muestra el valor medio junto con una banda de variabilidad.

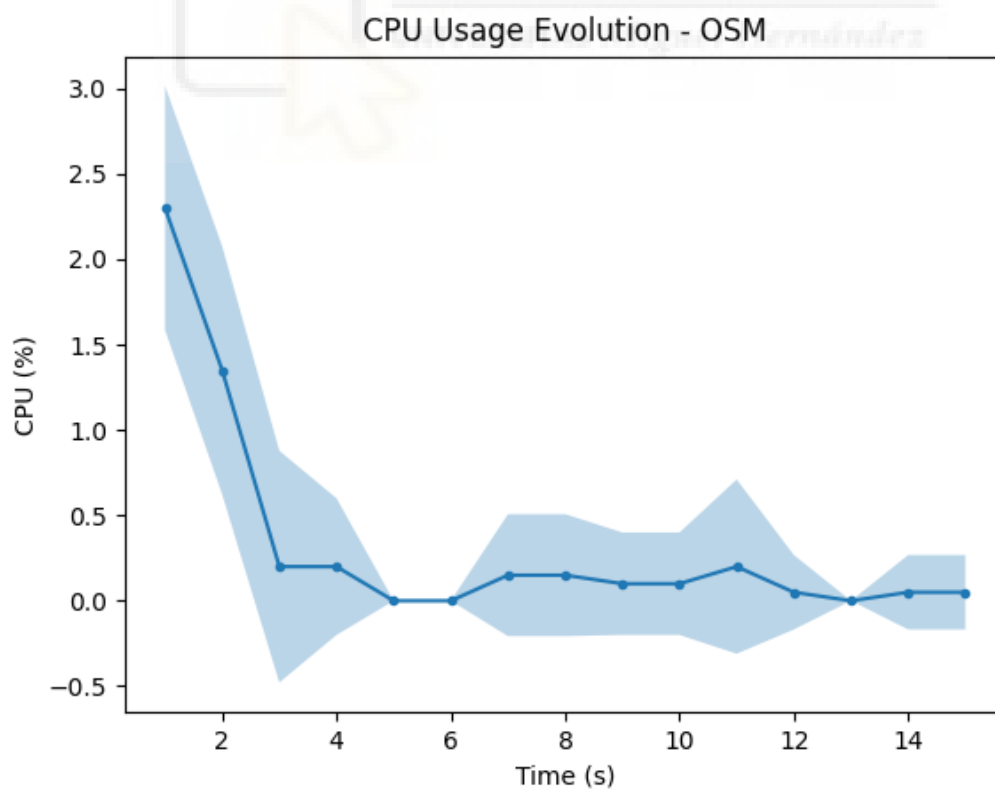


Figura 4.9 Evolución temporal del uso de CPU en modo sistema durante las migraciones en vivo en el escenario OSM. Se representa el valor medio junto con una banda de variabilidad.

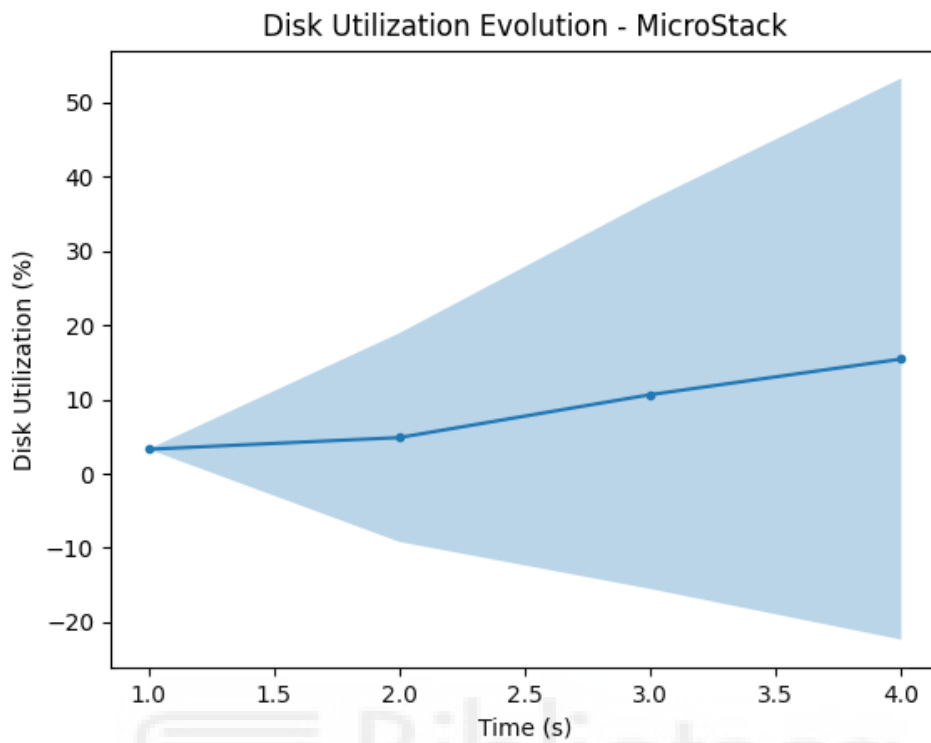


Figura 4.10 Evolución temporal de la utilización del disco (%util) durante las migraciones en vivo en el escenario MicroStack. Se muestra el valor medio con su variabilidad.

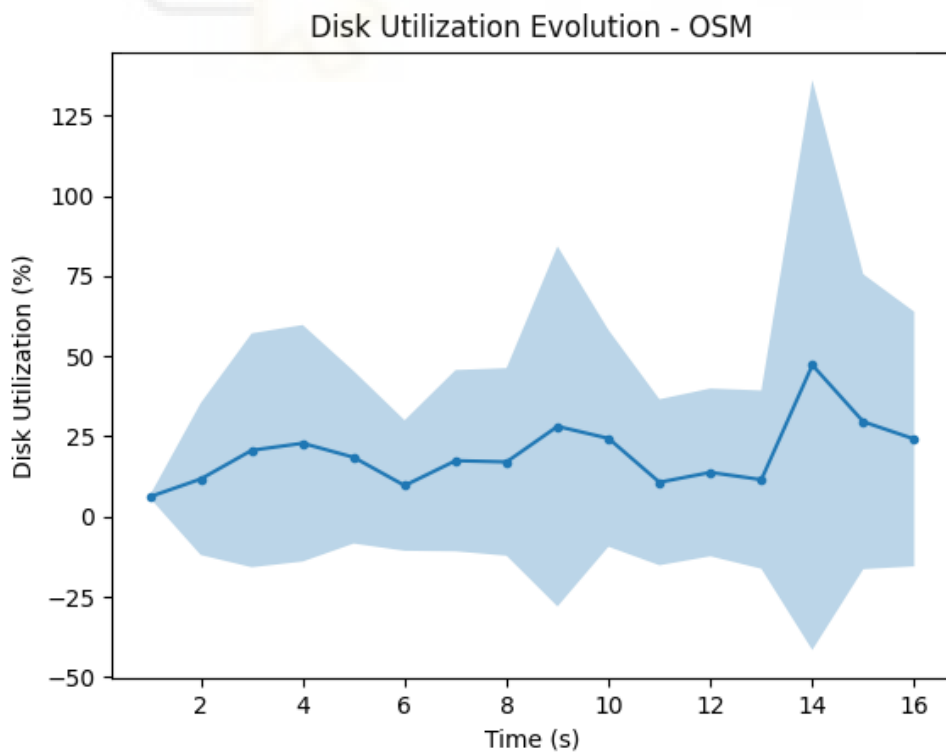


Figura 4.11 Evolución temporal de la utilización del disco (%util) durante las migraciones en vivo en el escenario OSM. Se representa el valor medio con su variabilidad.

Como puede observarse en las gráficas, las migraciones generan picos temporales de utilización tanto en CPU como en el subsistema de almacenamiento, siendo estos picos más prolongados e intensos en el escenario orquestado mediante OSM, lo cual es coherente con los resultados analizados en los apartados anteriores.

Cabe destacar que el intervalo temporal representado difiere entre los escenarios analizados. En el caso de MicroStack, la duración de las migraciones es significativamente menor, situándose en torno a unos pocos segundos, mientras que en el escenario orquestado mediante OSM el proceso se prolonga durante un intervalo considerablemente mayor.

Esta diferencia se debe directamente al mayor tiempo total de migración observado en el escenario OSM, tal y como se ha mostrado en el apartado 4.2. En consecuencia, el eje temporal de las figuras se ajusta a la duración media de las migraciones en cada escenario, lo que explica la diferencia en la longitud de las series representadas.

4.3.4 Evolución temporal por dirección de migración

Con el fin de analizar posibles diferencias en el comportamiento del sistema en función del nodo origen y destino, se ha realizado un estudio adicional considerando de forma separada las dos direcciones de migración: desde el Nodo Control hacia Node 1 y desde Node 1 hacia el Nodo Control.

Para cada dirección y para cada escenario experimental (MicroStack y OSM), se han agregado los resultados de las distintas ejecuciones, calculando el valor medio de las métricas capturadas mediante *vmstat* e *iostat* en cada instante temporal. Esta aproximación permite obtener una representación más completa del comportamiento del sistema, teniendo en cuenta tanto el modelo de gestión de la migración como la dirección en la que se realiza.

En las Figuras 4.12 a 4.15 se muestra la evolución temporal del uso de CPU en modo sistema y de la utilización del subsistema de almacenamiento, diferenciando las dos direcciones de migración en cada escenario. En cada caso, se representa el valor medio junto con una banda de variabilidad que refleja la dispersión de los datos entre ejecuciones.

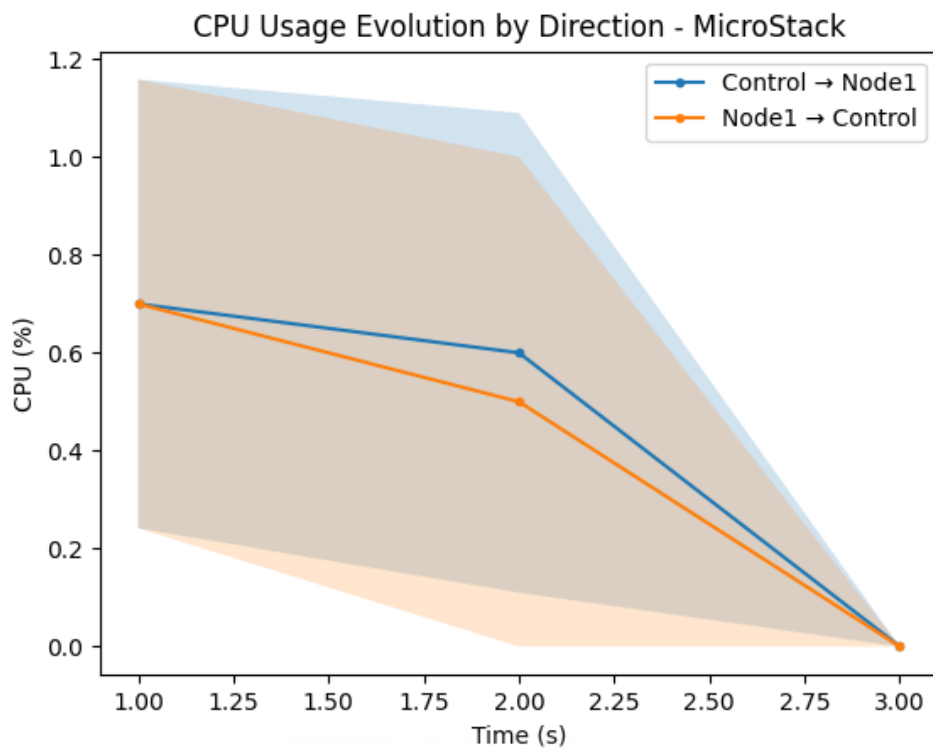


Figura 4.12 Evolución temporal del uso de CPU en modo sistema durante las migraciones en vivo en el escenario MicroStack, diferenciando ambas direcciones de migración. Se muestra el valor medio junto con la variabilidad de los datos.

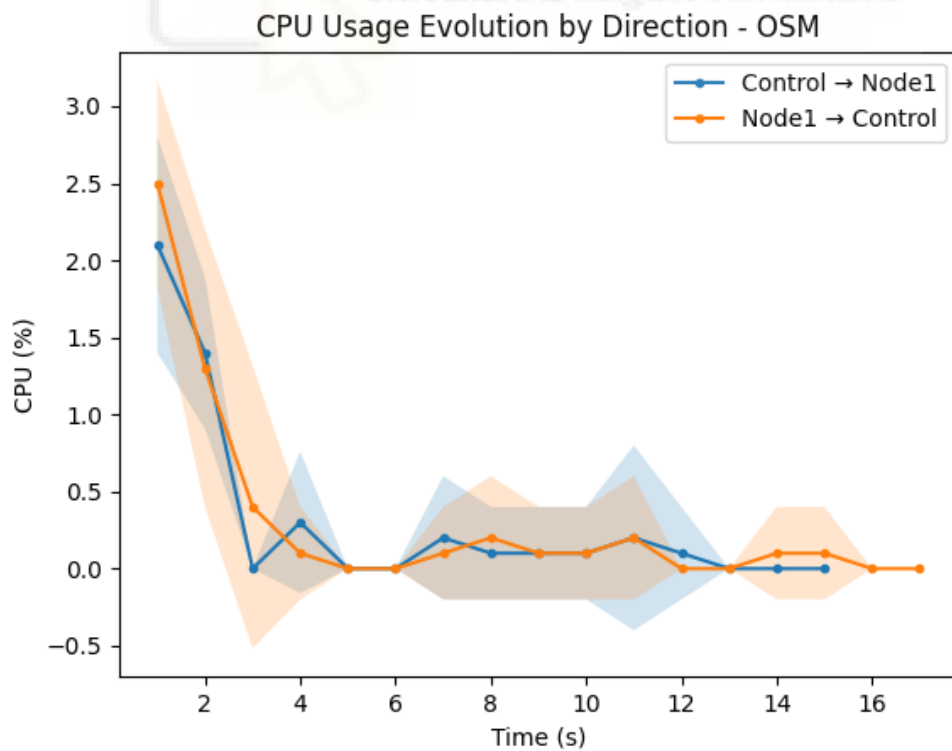


Figura 4.13 Evolución temporal del uso de CPU en modo sistema durante las migraciones en vivo en el escenario OSM, diferenciando ambas direcciones de migración. Se representa el valor medio junto con la variabilidad de los datos.

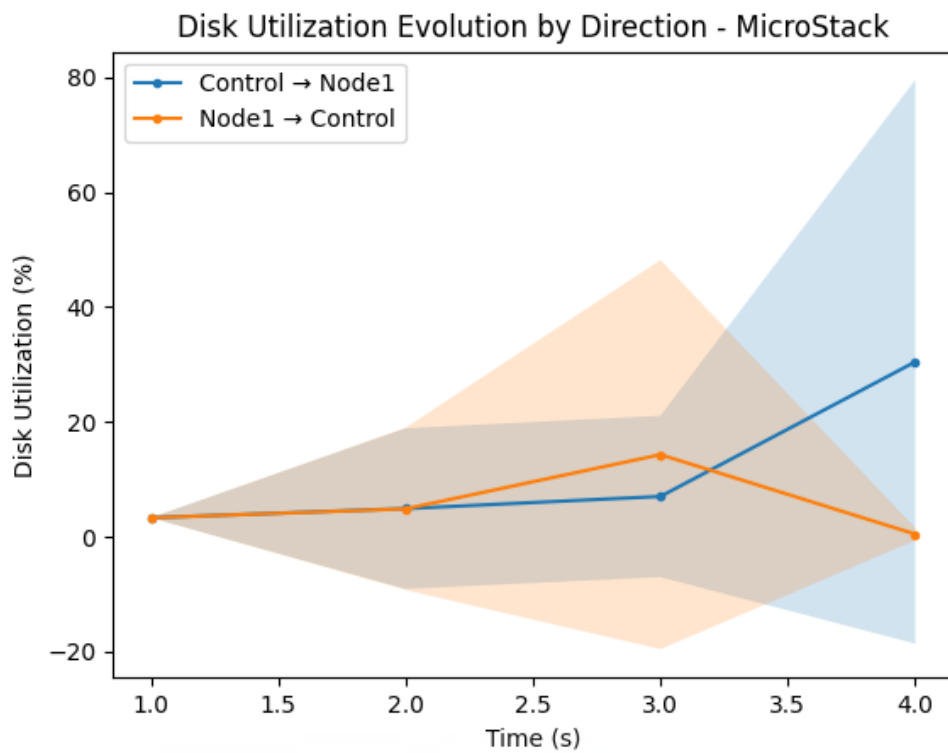


Figura 4.14 Evolución temporal de la utilización del disco (%util) durante las migraciones en vivo en el escenario MicroStack, diferenciando ambas direcciones de migración. Se muestra el valor medio con su variabilidad.

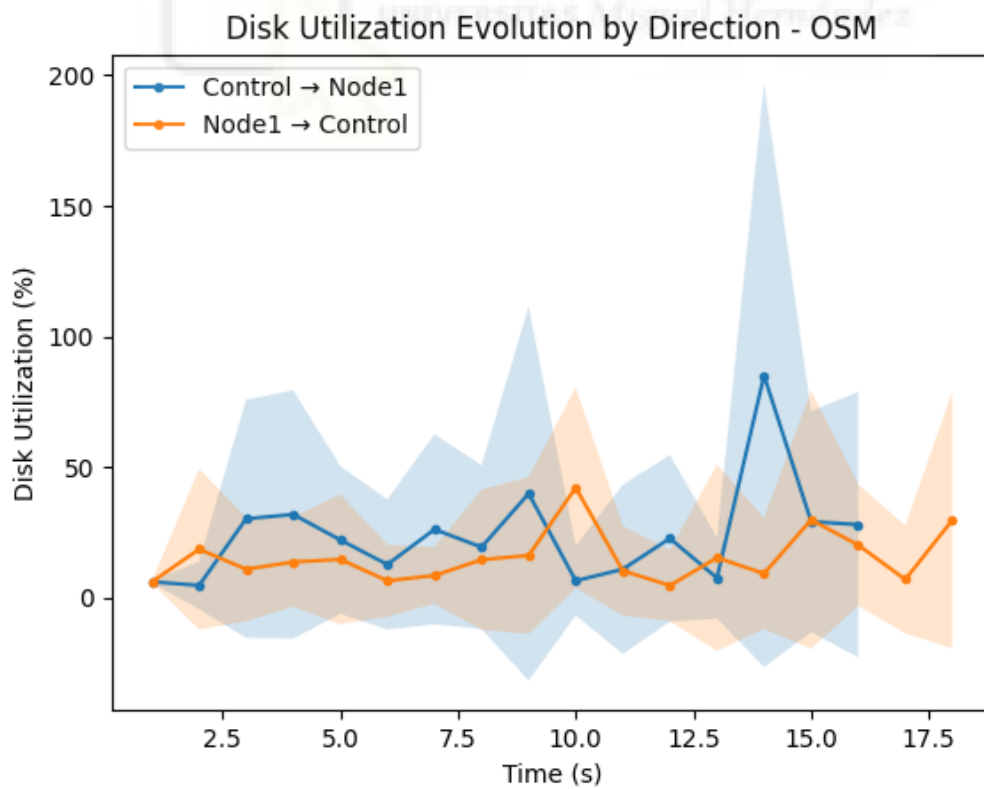


Figura 4.15 Evolución temporal de la utilización del disco (%util) durante las migraciones en vivo en el escenario OSM, diferenciando ambas direcciones de migración. Se representa el valor medio con su variabilidad.

Como puede observarse, se aprecian ligeras diferencias en la intensidad y duración de los picos de utilización en función de la dirección de migración, tanto en el escenario gestionado mediante MicroStack como en el orquestado mediante OSM. Estas diferencias pueden estar asociadas a variaciones en el estado del sistema o en la carga de los nodos en el momento de la migración.

No obstante, en ambos escenarios, los picos de utilización observados son temporales y no provocan una degradación sostenida del rendimiento del sistema, lo cual resulta coherente con los resultados obtenidos en el análisis del rendimiento del servicio presentado en los apartados anteriores.

4.4 Análisis comparativo global

La Tabla 4.2 resume las principales métricas obtenidas en ambos escenarios de migración.

MÉTRICA	MICROSTACK	OSM	DIFERENCIA APROXIMADA
Tiempo de migración	3.95 s	21.75 s	+450%
Tiempo total AB	1.724 s	1.475 s	-14%
Conexión media	16.9 ms	14.45 ms	-14%
CPU modo sistema	0.85%	2.35%	+176%
Disco %util	32.85%	129.82%	+295%

Tabla 4.2. Comparativa global de métricas

El análisis conjunto de los resultados permite extraer varias conclusiones relevantes:

1. La capa de orquestación introduce un overhead significativo en la duración total del proceso de migración.
2. El consumo de recursos internos (CPU en modo sistema y utilización de disco) es mayor bajo OSM.

3. No se observa degradación apreciable en la calidad de servicio HTTP percibida por el cliente.

Este comportamiento resulta especialmente interesante en el contexto de entornos edge computing. Aunque el tiempo total del procedimiento aumenta de forma notable cuando se emplea una plataforma de orquestación, la continuidad del servicio y la experiencia del usuario no se ven comprometidas de manera significativa.

Por tanto, puede concluirse que la inclusión de una plataforma de orquestación como OSM:

- Penaliza el tiempo interno de la operación de migración.
- Incrementa el consumo de recursos durante el proceso.
- No impacta negativamente en la calidad de servicio percibida.

Este equilibrio entre coste operacional y continuidad del servicio constituye uno de los principales hallazgos del presente trabajo y aporta evidencia empírica sobre el impacto real de introducir una capa adicional de gestión en entornos virtualizados basados en NFV y edge computing.



5 CONCLUSIONES

El presente Trabajo Fin de Máster ha tenido como objetivo principal evaluar y cuantificar el impacto que introduce una plataforma de orquestación en el proceso de migración en vivo de máquinas virtuales, comparando su comportamiento frente a la gestión directa desde la infraestructura cloud. Para ello, se ha diseñado y desplegado un entorno experimental controlado basado en tecnologías de virtualización (Proxmox y MicroStack) y orquestación (OSM), en el que se han ejecutado migraciones en vivo bajo carga constante de servicio.

Los resultados obtenidos permiten extraer conclusiones relevantes tanto desde el punto de vista infraestructural como desde la perspectiva del servicio.

En primer lugar, se ha comprobado que la migración en vivo gestionada mediante OSM introduce un overhead significativo en la duración total del proceso. Concretamente, el tiempo medio de migración pasa de 3.95 segundos en el escenario gestionado directamente por MicroStack a 21.75 segundos en el escenario orquestado, lo que supone un incremento aproximado del 450%. Este resultado confirma que la inclusión de una capa adicional de gestión y coordinación impacta de manera directa en el tiempo interno requerido para completar la operación de migración.

En segundo lugar, el análisis del consumo de recursos del sistema ha mostrado que la migración orquestada presenta un mayor uso de CPU en modo sistema y picos significativamente superiores en la utilización del subsistema de almacenamiento. Estos resultados son coherentes con la existencia de operaciones adicionales de coordinación, validación y comunicación entre la plataforma de orquestación y la infraestructura cloud subyacente. El overhead observado se manifiesta principalmente en el plano interno de la infraestructura, afectando al consumo de recursos y a la duración del procedimiento.

Sin embargo, uno de los hallazgos más relevantes del estudio es que este incremento en el tiempo total de migración y en el consumo de recursos no se traduce en una degradación apreciable de la calidad de servicio percibida por el usuario final. Las métricas obtenidas mediante Apache Benchmark muestran que el rendimiento del servicio HTTP bajo carga se mantiene estable durante la migración en ambos escenarios. Incluso se observan ligeras diferencias a favor del escenario orquestado en los tiempos de respuesta, si bien estas variaciones son reducidas y no representan cambios significativos desde el punto de vista funcional.

Este resultado es especialmente relevante en el contexto de arquitecturas de edge computing y entornos NFV, donde la continuidad del servicio y la experiencia de usuario constituyen factores críticos. Los datos obtenidos indican que, aunque la orquestación penaliza el tiempo interno del procedimiento de migración, no compromete la disponibilidad ni la calidad del servicio ofrecido. Desde una perspectiva operativa, esto implica que el coste adicional introducido por la capa de orquestación se mantiene en el plano de gestión interna del sistema, sin impactar negativamente en la percepción del usuario.

Por tanto, puede concluirse que la utilización de una plataforma de orquestación como OSM supone un incremento del coste operativo de las migraciones en vivo en términos de duración y consumo de recursos, pero no genera un deterioro en la calidad del servicio bajo carga. Este equilibrio entre mayor complejidad interna y mantenimiento de la experiencia de usuario refuerza la viabilidad de emplear soluciones de orquestación en entornos virtualizados, incluso en escenarios sensibles al rendimiento.

Desde el punto de vista metodológico, el trabajo ha permitido diseñar un entorno experimental reproducible y controlado, capaz de generar métricas comparables y estadísticamente representativas. La repetición sistemática de las pruebas y el análisis mediante intervalos de confianza al 95% aportan solidez a las conclusiones alcanzadas.

5.1 Limitaciones del estudio

Como todo estudio experimental, este trabajo presenta determinadas limitaciones que deben considerarse al interpretar los resultados.

En primer lugar, el entorno utilizado corresponde a un laboratorio virtualizado, donde todos los componentes se ejecutan sobre una infraestructura compartida. Aunque se ha buscado un equilibrio entre realismo y control experimental, el comportamiento en entornos productivos con mayor escala y recursos dedicados podría presentar variaciones.

En segundo lugar, el servicio evaluado corresponde a un servidor web sencillo sometido a carga HTTP moderada. Servicios con mayores requisitos de cómputo, memoria o almacenamiento podrían mostrar un comportamiento diferente durante el proceso de migración en vivo.

Finalmente, el estudio se centra en un único orquestador (OSM) y en una configuración específica de la infraestructura. Otros orquestadores o configuraciones alternativas podrían introducir comportamientos distintos en términos de overhead y consumo de recursos.

5.2 Líneas futuras de trabajo

A partir de los resultados obtenidos, se plantean varias posibles líneas de trabajo futuras:

- Evaluar el comportamiento de las migraciones en vivo bajo niveles de carga más elevados o variables, analizando el impacto en escenarios de mayor estrés.
- Extender el estudio a servicios más complejos, incluyendo aplicaciones multinstancia o arquitecturas distribuidas.
- Analizar el comportamiento en entornos con almacenamiento compartido frente a block migration, comparando su impacto en el subsistema de disco.
- Estudiar el impacto de diferentes orquestadores o versiones más recientes de las plataformas empleadas.
- Incorporar métricas adicionales, como consumo energético o latencia de red durante la migración.

En conclusión, el trabajo realizado demuestra que la orquestación introduce un overhead interno significativo en los procesos de migración en vivo, pero no compromete la continuidad ni la calidad del servicio bajo carga. Este hallazgo aporta evidencia empírica relevante para la evaluación de arquitecturas NFV y edge computing, y contribuye a una mejor comprensión del equilibrio entre automatización avanzada y rendimiento infraestructural en entornos virtualizados.

6 Anexo A – Comandos de configuración de la instancia del servicio

En este anexo se recogen los comandos utilizados para la configuración manual de la instancia del servicio tras su creación en el entorno MicroStack. Aunque la imagen seleccionada es compatible con cloud-init, en el experimento no se utilizó un fichero de inicialización personalizado, realizándose la instalación del servidor web de forma manual.

En primer lugar, se actualizó la lista de paquetes del sistema operativo:

```
sudo apt update
```

A continuación, se instaló el servidor web **Apache**, que se utilizaría para servir las peticiones HTTP generadas durante los experimentos:

```
sudo apt install -y apache2
```

Una vez instalado el servidor web, se habilitó el servicio para que se iniciara automáticamente junto con el sistema:

```
sudo systemctl enable apache2
```

Posteriormente, se inició el servicio Apache:

```
sudo systemctl start apache2
```

7 Anexo B – Script de automatización de las migraciones

En este anexo se incluye el script desarrollado para la automatización de las pruebas experimentales realizadas en el presente trabajo. Su finalidad es coordinar la generación de carga sobre el servicio, la monitorización del sistema y el registro temporal de cada ejecución, permitiendo la obtención de métricas de forma organizada y consistente.

El script integra la ejecución de la herramienta Apache Benchmark para la generación de tráfico HTTP, junto con las herramientas de monitorización del sistema **vmstat** e **iostat**, configuradas con un intervalo de muestreo de un segundo. De este modo, se obtiene información detallada sobre el comportamiento del sistema durante el proceso de migración en vivo.

Para cada ejecución, el script crea automáticamente un directorio específico en el que se almacenan los resultados generados, incluyendo las salidas de las herramientas de monitorización, los resultados del benchmark y la duración total de la prueba. Esta organización facilita el posterior análisis de los datos y la comparación entre los distintos escenarios experimentales.

Asimismo, el script permite identificar la dirección de cada migración, diferenciando entre las ejecuciones realizadas entre los nodos de cómputo del entorno experimental, lo que resulta especialmente útil para el análisis comparativo presentado en el capítulo de resultados.

El código completo del script utilizado se presenta a continuación:

```
#!/bin/bash

# CONFIGURACIÓN
TARGET_IP="10.20.20.68"
NUM_PRUEBAS=20
PETICIONES=1000
CONCURRENCIA=10

mkdir -p experimento

for i in $(seq 1 $NUM_PRUEBAS); do
    echo "===== PRUEBA $i ====="
    read -p "¿Esta migración es de control a node1 o al revés? (c->n / n->c): "
    direc

    DIR="experimento/mig$(printf "%02d" $i)_$direc"
```

```

mkdir -p "$DIR"

echo "[INFO] Esperando a que inicies la migración..."
read -p "Presioná Enter cuando INICIES la migración..."

START_TIME=$(date +%s)

# Iniciar monitorización
vmstat 1 > "$DIR/vmstat.txt" &
VMSTAT_PID=$!

iostat -dx 1 > "$DIR/iostat.txt" &
IOSTAT_PID=$!

echo "[INFO] Ejecutando benchmark (ab)..."
ab -n $PETICIONES -c $CONCURRENCIA http://$TARGET_IP/ >
"$DIR/ab_output.txt"

echo "[INFO] Esperando a que finalice la migración..."
read -p "Presioná Enter cuando TERMINE la migración..."

END_TIME=$(date +%s)
DURATION=$((END_TIME - START_TIME))

echo "[INFO] Finalizando monitorización..."
kill $VMSTAT_PID
kill $IOSTAT_PID

echo "Tiempo de migración: $DURATION segundos" | tee "$DIR/duracion.txt"
echo "[OK] Prueba $i completada."

echo ""
done

```

8 Bibliografía

- [1] ETSI, *Network Functions Virtualisation (NFV); Architectural Framework*, ETSI GS NFV-MAN 001, European Telecommunications Standards Institute, 2014.
- [2] OpenStack Foundation, *OpenStack Documentation – Live Migration of Instances*, Available: <https://docs.openstack.org/>
- [3] Clark, C. et al., “Live Migration of Virtual Machines,” *Proceedings of the 2nd Symposium on Networked Systems Design & Implementation (NSDI)*, 2005.
- [4] Kivity, A. et al., “KVM: the Linux Virtual Machine Monitor,” *Proceedings of the Linux Symposium*, 2007.
- [5] ETSI, *Open Source MANO (OSM) Documentation*, European Telecommunications Standards Institute, Available: <https://osm.etsi.org/>
- [6] Linux Manual Page, *vmstat(8) – Report virtual memory statistics*, Available: <https://man7.org/linux/man-pages/man8/vmstat.8.html>
- [7] Linux Manual Page, *iostat(1) – CPU and I/O statistics*, Available: <https://man7.org/linux/man-pages/man1/iostat.1.html>