

# Un Modelo de Arquitectura para un Sistema de Virtualización Distribuido

## Tesis completa:

<http://sedici.unlp.edu.ar/handle/10915/71044>



## Tesista

Pablo A. Pessolani

## Directores

Dr. Toni Cortés

Dr. Fernando G. Tinetti

## CoDirector

Dr. Silvio Gonnet

## MOTIVACIÓN

Sin considerar cuál es la denominación del entorno de ejecución creado por la virtualización (Máquina Virtual, Contenedor, Prisión, etc.) según su tipo, queda claro que éste resulta del particionado de los recursos de un computador. Como consecuencia de esto, el poder de cómputo y la utilización de recursos de esos entornos estarán acotados por el computador que lo contiene.

Al menos dos preguntas surgen al considerar esta limitación:

- 1. ¿Cómo se pueden expandir el poder de cómputo y la utilización de recursos de una VM o Contenedor a varios computadores?**
- 2. ¿Cómo lograr mayores niveles de rendimiento y escalabilidad de las aplicaciones que se ejecutan en la Nube?**

Se propone un modelo de Arquitectura de un Sistema de Virtualización Distribuido (DVS: Distributed Virtualization System), el cual permite construir dominios de ejecución denominados Contenedores Distribuidos (DC: Distributed Containers) que pueden extenderse más allá de los límites de una máquina física y con ello lograr mayores niveles de rendimiento y escalabilidad requerido por las actuales aplicaciones que se desarrollan para ejecutar en la Nube.

## APORTES DE LA TESIS

Esta tesis propone un modelo de arquitectura de un DVS que da respuesta a las dos preguntas planteadas:

- 1. ¿Cómo se pueden expandir el poder de cómputo y la utilización de recursos de una VM o Contenedor a varios computadores?**

Los límites de un entorno aislado para la ejecución de aplicaciones DC pueden expandirse a todos los nodos del clúster (agregación) mejorando sus características de rendimiento, escalabilidad y disponibilidad. Además, los nodos de un cluster pueden compartirse entre varios DCs (partición), mejorando así la utilización de la infraestructura de hardware.

## 2. ¿Cómo lograr mayores niveles de rendimiento y escalabilidad de las aplicaciones que se ejecutan en la Nube?

La metodología actual para lograr que las aplicaciones obtengan mayores niveles de rendimiento y escalabilidad consiste en utilizar un gestor de contenedores para desplegar un “enjambre” de ellos en los cuales se ejecutan los componentes de las aplicaciones y sus réplicas. Con esta metodología, los programadores y administradores que deben implementar, desplegar, operar, gestionar y controlar cada aplicación carecen de una visión integrada, lo que incrementa los tiempos y costos de esas tareas. A diferencia de los Gestores de Contenedores, un VOS (Sistema Operativo Virtual) distribuido ejecutando dentro de un DC aporta una visión integrada de las aplicaciones equivalente a un sistema centralizado evitando que el programador deba considerar la localización de cada componente en el cluster, su monitoreo, su direccionamiento (direcciones IP, puertos TCP, etc.), su seguridad (reglas de firewall), etc. Esta visión integrada facilita la gestión reduciendo los tiempos y los costos de implementación, operación y mantenimiento de aplicaciones en la Nube. Para verificar el diseño, detectar errores y proponer mejoras se desarrolló un prototipo de DVS que permitió comprobar la viabilidad del modelo propuesto. Uno de los productos más importantes que arrojó el prototipo es M3-IPC, como mecanismo de comunicaciones entre procesos. M3-IPC resuelve algunos problemas relacionados con el soporte de subprocesos (threads), la transparencia de la ubicación, la redirección de mensajes en la migración de procesos, el protocolo de transporte de red independiente y el confinamiento de IPC (Comunicación entre Procesos) para la virtualización. Los resultados muestran que M3-IPC logra todos sus objetivos de diseño, con un alto rendimiento tanto para los mensajes y transferencias de datos intra-nodo e inter-nodo.

1. Ejecutar UML (User Mode Linux) como VOS dentro de un DC.

2. Ejecutar un VOS Unikernel generado con rumpkernel con soporte POSIX dentro de un DC.

Ambos podrían utilizar servicios distribuidos en otros nodos del mismo DC. Estos proyectos demostrarían que un Linux en modo usuario (UML) o un Unikernel basado NetBSD pueden adaptarse como VOS de un DVS, aumentando así la confianza en el modelo propuesto.

## LINEAS DE I/D FUTURAS

Las líneas de I/D futuras, en general, se enfocarán en la mejora del modelo y actualización su prototipo. Para los próximos años, se planean desarrollar los siguientes proyectos que pueden producir un gran impacto en la adopción del modelo de DVS: