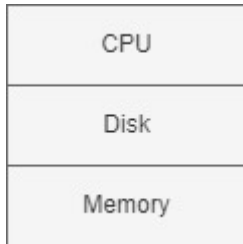


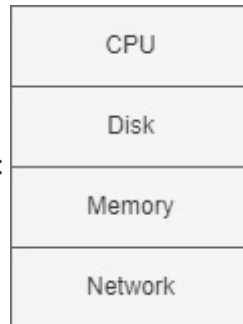
Clase del viernes 28 de julio del 2023

Al rededor de los años 2000, los componentes principales de una computadora eran 3: CPU, Disk

y Memory.



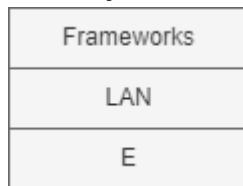
Posteriormente la parte de red tomó importancia:



Los 3 componentes principales establecen protocolos de bajo nivel, que especifica cómo se organizan cadenas de bits para transmitir un mensaje.

Actualmente se utilizan Frameworks, debajo de ellos están los lenguajes de alto nivel, más abajo

están los lenguajes ensamblador.

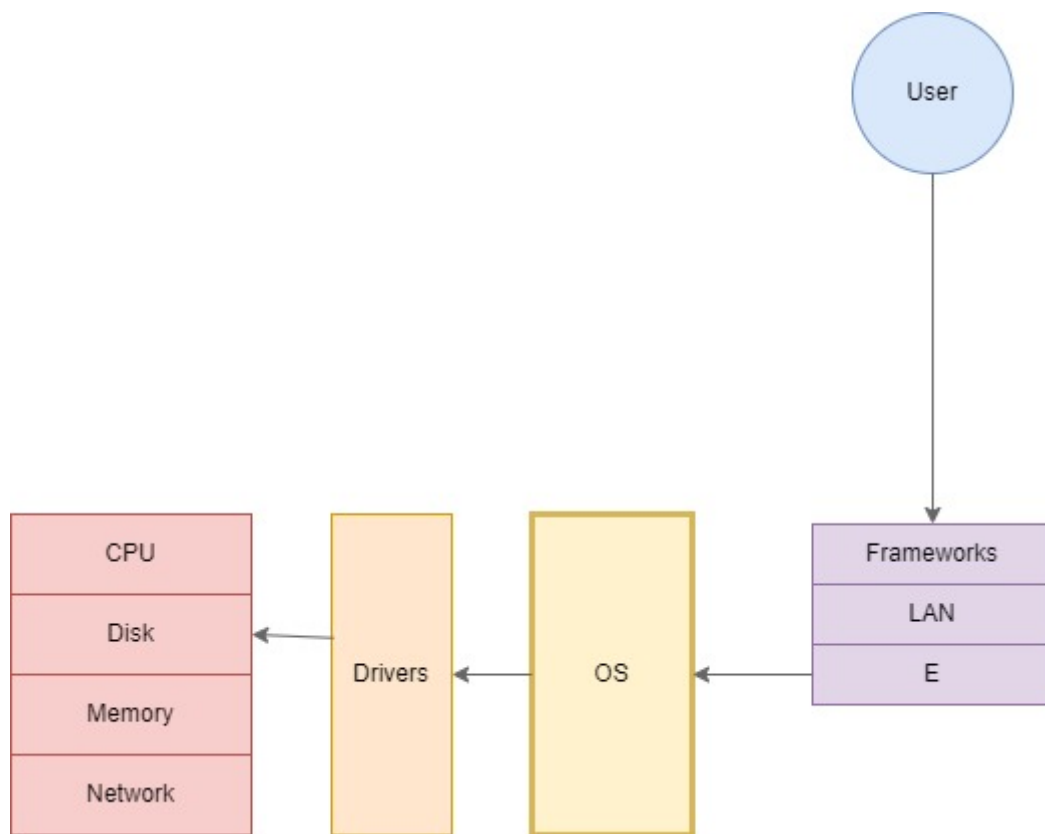


El sistema operativo es quien conecta las 2 partes, ya que los lenguajes de alto nivel, frameworks y demás no puede conversar de tú a tú con el hardware, otro componente que permite la comunicación junto con el sistema operativo serían los drivers.

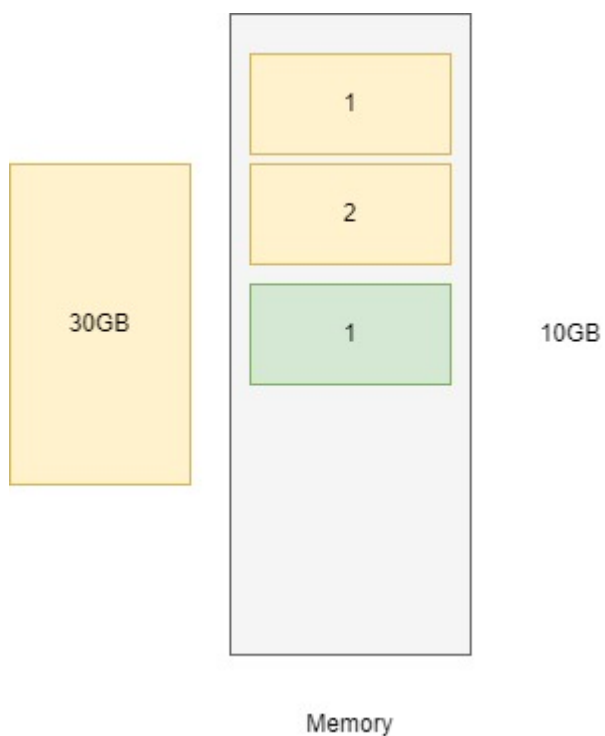
Cuando un usuario escribe un programa para llegar al disco duro, se relaciona con el lenguaje de alto nivel, estos lo hacen con el sistema operativo para solicitar permisos, el OS permite que se hable con el driver y se establecen las conexiones para la comunicación entre el programa y el dispositivo.

Esto conlleva a que el sistema operativo se convierta en un cuello de botella.

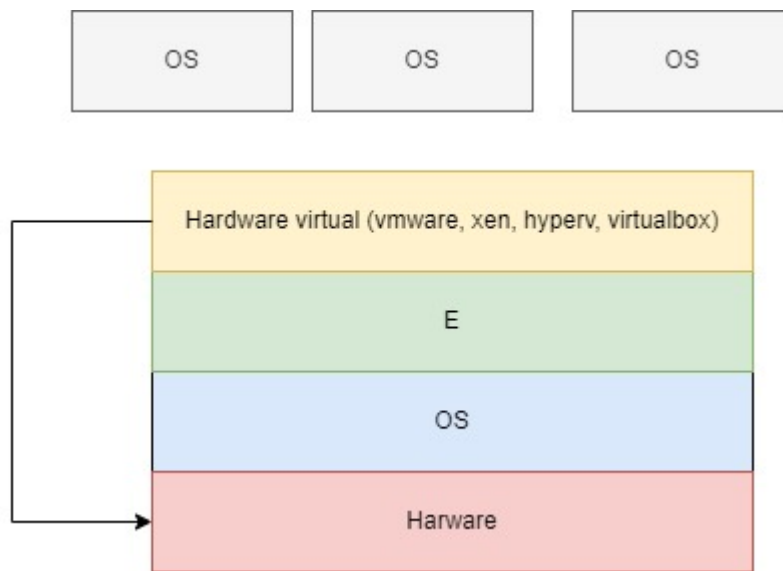
Estos sistemas operativos se conocen como monolíticos, esto significa que dicho sistema es dueño de todo lo que se hace en la computadora.



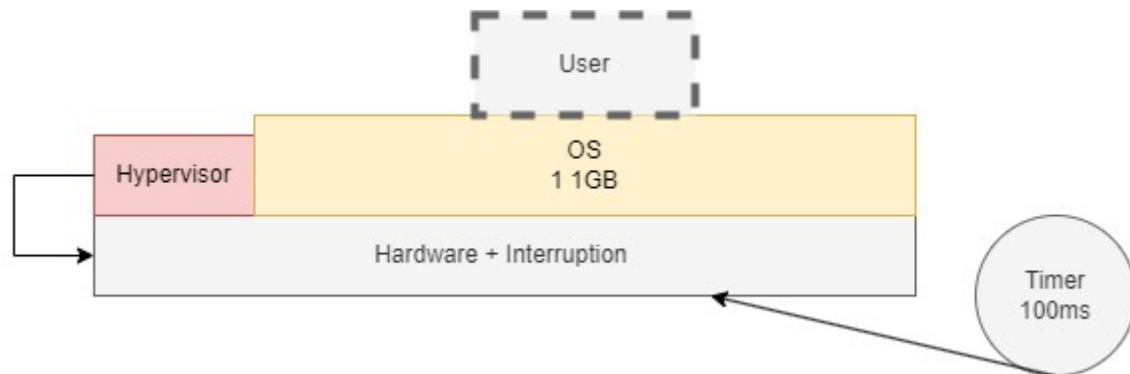
Cuando un archivo no cabe completo en memoria, se crean páginas del archivo en un proceso llamado "paginación", ahí es donde se ejecuta el ciclo de Fetch.



Los hardware virtuales hacen mímica de un hardware real, permitiendo correr distintos sistemas operativos repartiendo el tiempo para cada uno.

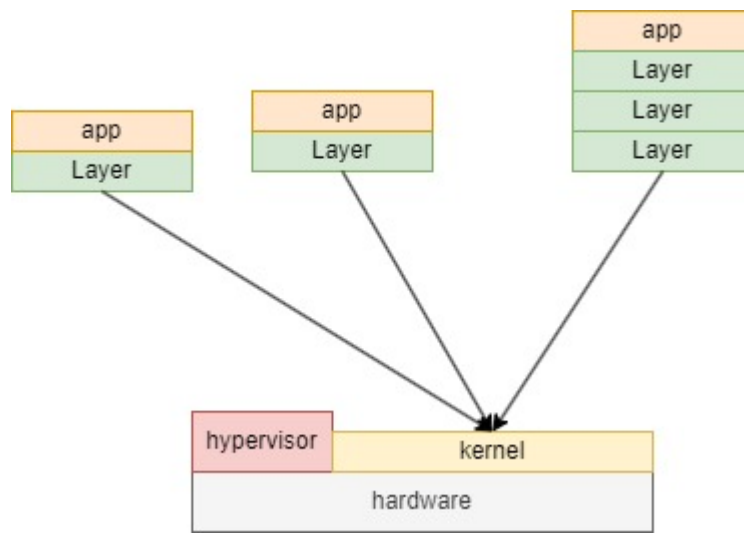


hypervisor es como un microkernel, toma el sistema operativo y le quita el monopolio de administrar todos los recursos de hardware de la máquina. Es de tamaño reducido porque solo contiene lo necesario para controlar el tiempo que utiliza cada sistema operativo.



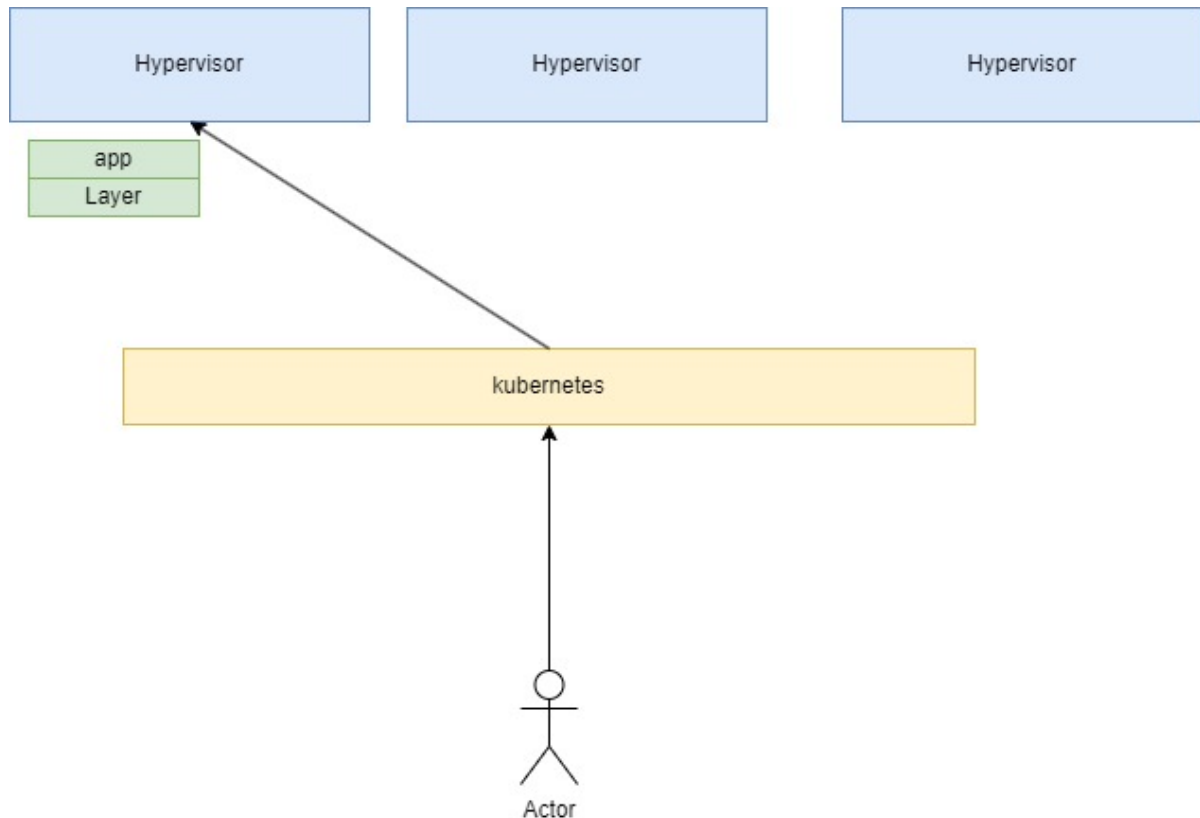
Esto presentaba un problema ya que los sistemas operativos que se utilizaban eran sistemas completos, que podían ser de gran tamaño y al cambiar de sistema operativo, el hypervisor pierde mucho tiempo. Del tamaño completo de los sistemas operativos para ejecutar el programa específico solo se necesita una parte, por lo que es mejor reducir su tamaño y que se convierta en un sistema operativo especializado.

El kernel es el núcleo del sistema operativo, la base de lo que se necesita para correr el sistema operativo, se pensó en poner "layers" que interactúan con el kernel y sobre dichos layers corre un app, con este modelo surgió la idea de un "container", su distribución es la siguiente:

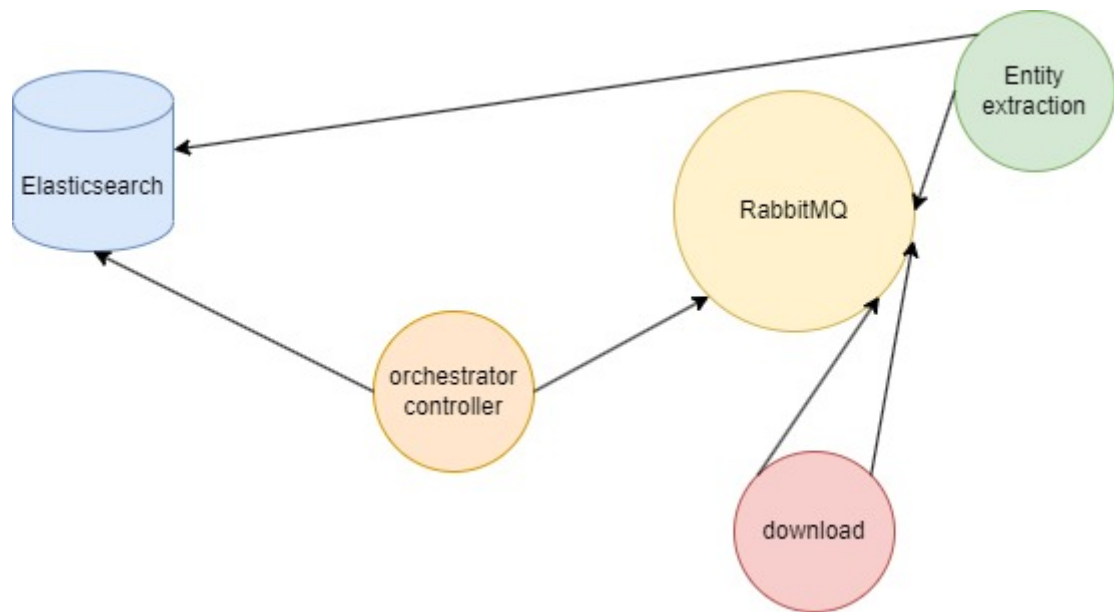


Docker fue el líder en contenedores por muchísimo tiempo, pero tenía desventajas como no tener alta disponibilidad.

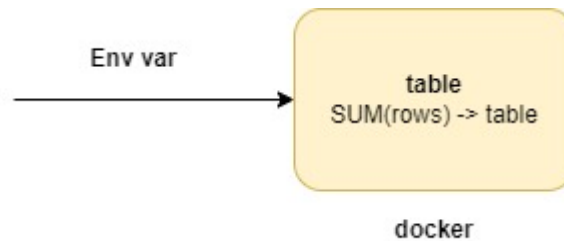
Posteriormente surge Kubernetes. Es un framework que administra contenedores.



Un controller observa un Elasticsearch, duerme por intervalos y revisa, en el momento en que encuentre algo, luego publica un mensaje a RabbitMQ, otro componente realiza el download para cuando el mensaje llega, después se publica otro mensaje a RabbitMQ y se realiza la extracción de la entidades y finalmente se escribe en Elasticsearch.



Es posible cambiar la forma en la que se comporta un contenedor mediante variables de entorno.



Es importante hacer unit test en las partes más importantes del código para asegurarse de que todo funcione bien después de hacer una modificación en el código, especialmente cuando varios desarrolladores modifican el mismo código.

La herramienta kubectl permite comunicarse con kubernetes API, normalmente la especificación de lo que se ejecuta dentro de kubernetes se escribe en lenguaje YAML, este es hardcoded por lo que no es posible hacer código reutilizable, por esto se utiliza un helm chart, que engloba los YAML y kubectl.

Helm permite automatizar la instalación de contenedores, para esto necesita un helm chart, el cual es como un módulo de programación, que tiene un template que es el código con parámetros y un archivo de valores con valores para los parámetros.

