Entornos de ejecución de computación para el análisis de datos

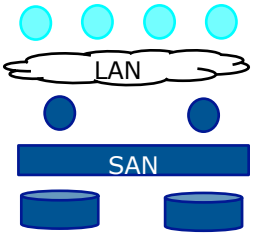
Paralelismo y Sistemas Distribuidos

Requerimientos (I)

- Eficiencia en el uso de CPU pierde relevancia
  - Trabajo sobre gran cantidad de datos
  - Importante aumentar la capacidad de almacenamiento y optimizar el acceso a datos
- Normalmente, la estructura de las aplicaciones es sencilla
  - Mayor parte del código es masivamente paralelo: misma función aplicada sobre diferentes subconjuntos de datos sin dependencias
  - Fase de reducción para combinar resultados parciales

Requerimientos (II)

- Arquitectura de procesado evita el uso de servidores centralizados
  - Datos almacenados en discos locales



IBM GPFS

- Importante
  - Escalabilidad horizontal
    - Rendimiento debe mejorar de manera proporcional si se añaden nuevos nodos de cálculo/almacenamiento
  - Tolerancia a fallos
    - Si un nodo cae, la aplicación debe poder seguir funcionando y los datos deben seguir accesibles (replicación)

1.3

Requerimientos (adicionales)

- Simplicidad de uso
  - Del entorno de ejecución
  - Del modelo de programación
- "Pagar" sólo por lo que se usa

1.4

Cloud computing

- *Cloud*: conjunto *compartido* de recursos de computación (ofrecidos como servicios)
- Tipos de Cloud
  - Privados: para uso interno de una entidad
  - Públicos: para ofrecer servicios a usuarios externos
    - Utility computing: proveedor ofrece servicios durante el tiempo contratado
  - Híbridos
- Permite utilizar/contratar la cantidad de servicios necesaria durante sólo el tiempo necesario y adaptarse dinámicamente a medida que cambian los requisitos (aumentando o disminuyendo la cantidad o cambiando el tipo)

1.5

Servicios ofrecidos en el cloud: HW

- Infrastructure As A Service (**IaaS**)
- Se solicita una cantidad de recursos hardware (cpu's, gpu's, memoria, disco,…)
- A partir de ahí el cliente se encarga de toda la instalación del software que necesita (desde el Sistema Operativo hasta las aplicaciones que quiere ejecutar)
- Ejemplos: Amazon Web Services EC2 y S3

1.6

Servicios ofrecidos en el cloud: Entorno

- Platform As A Service (**PaaS**): Entorno de ejecución (además del hw, Sistema operativo, entorno de desarrollo, bases e datos…)
- Se solicita una cantidad de recursos hardware con el software necesario para poder ejecutar las aplicaciones propias
- El cliente instala y pone en ejecución las aplicaciones que necesita pero no controla el hw ni el software ya instalado
- Ejemplos: Google Cloud, Microsoft Azure

1.7

Servicios ofrecidos en el cloud: SW

- Software As A Service (**SaaS**)
- El cliente puede utilizar aplicaciones que se ejecutan en el cloud
- No tiene control sobre la infrastructura (hw o sw) que se está utilizando
- Ejemplos: Google Docs, Gmail…

1.8

Servicios ofrecidos en el cloud: AlaaS

- Artificial Intelligence As A Service
- Ejemplos:
  - SageMaker (Amazon)
    - Interfaz Jupyter notebook
    - Cubre todas las fases del workflow de ML: permite construir modelos, entrenarlos y ponerlos en producción
  - Azure Machine Learning Studio (Microsoft)
    - Interfaz web o mediante Jupyter notebook
    - Cubre todas las fases del workflow de ML: permite construir modelos, entrenarlos y ponerlos en producción
    - Azure Intelligence Gallery: para compartir recursos relacionados con Azure (tutoriales, modelos, experimentos…)
  - Cloud Machine Learning Engine (Google)
    - Permite entrenar modelos y ponerlos en producción
  - Watson Analytics (IBM)

1.9

Virtualización

- La virtualización de hardware facilita la explotación de los datacenter
- En general, la virtualización consiste en extender o reemplazar una interfaz para mimetizar el comportamiento de otro sistema

1.10

Ventajas de la virtualización

- Crea la ilusión de múltiples sistemas dedicados sobre un único sistema físico
- Aisla los programas del sistema subyacente y de otros programas
- Los entornos virtuales pueden ser creados fácilmente y con poco overhead
- Facilita la portabilidad de los programas
- Permite que se haga asignación de recursos de grano fino y de una manera ágil
- Facilita la gestión de la tolerancia a fallos (más sencillo recuperarse de cualquier error)

Tipos de virtualización (I)

- Virtualización a nivel HW
  - Permite crear una abstracción del sistema completo (HW+SW), permitiendo que un SO *guest* se ejecute de manera aislada sobre un sistema nativo (*host*)
  - Ejemplos: KVM, Xen, VMWare Workstation, VirtualBox
  - Dos tipos
    - Virtualización completa
      - Capa de software que multiplexa el uso de los recursos físicos para crear la ilusión de que hay suficientes y emula el comportamiento del hardware (hypervisor o Virtual Machine Monitor (VMM))
      - Hypervisor intercepta la gestión de las excepciones /interrupciones / llamadas a sistema y cualquier operación privilegiada que intente ejecutar el SO guest
    - Paravirtualización: el SO guest está modificado para poder interactuar con el hypervisor (hypercalls ≈ llamadas a sistema)

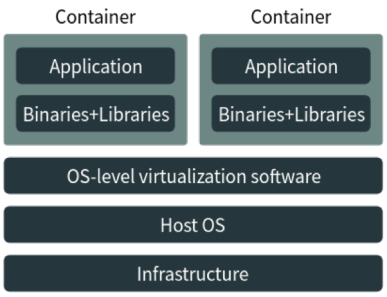
Tipos de virtualización (II)

- Desventajas
  - La inicialización es lenta
    - Creación de las máquinas virtuales
    - Carga del Sistema Operativo guest
  - Espacio de disco/memoria necesario para guardar las imágenes, y seguramente debido a replicación de código innecesario (por ejemplo, imágenes con el mismo SO)

1.13

Tipos de virtualización (III)

- Virtualización a nivel SO
  - Se basa en replicar únicamente el espacio a nivel de usuario compartiendo un único SO: *containers*
  - Ejemplos: docker, singularity,…
  - Cada *container* "se cree" una máquina virtual: la ilusión se consigue particionando el espacio de nombres y aislando qué parte puede ver cada *container* (identificadores de proceso, de usuario, interfaz de red, sistema de ficheros….)



1.14

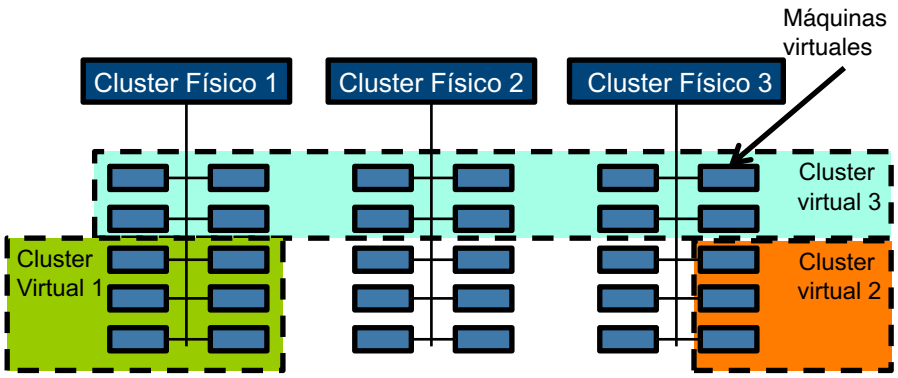
Tipos de virtualización (IV)

- Ventajas
  - Mínimo coste para las operaciones de iniciar y parar containers
  - Mínimo consumo de recursos y alta escalabilidad
- Desventajas
  - Todos los containers se ejecutan sobre el mismo SO host

1.15

Clusters virtuales

- Clusters construidos mediante máquinas virtuales ejecutadas en uno o más clusters físicos



1.16

Características de los clusters virtuales

- Se pueden tener N VM con diferentes SO sobre el mismo host
- El tamaño del cluster puede cambiar dinámicamente
- El fallo de un host físico sólo afecta a las VM que alberga
- El fallo de una VM no afecta al resto de VM's ni al host
- Se puede usar para
  - Ofrecer ejecución distribuida, tolerancia a fallos o recuperación de errores
  - Implementar compartición de hardware

1.17