

Programación distribuida

Que es?

Que es?

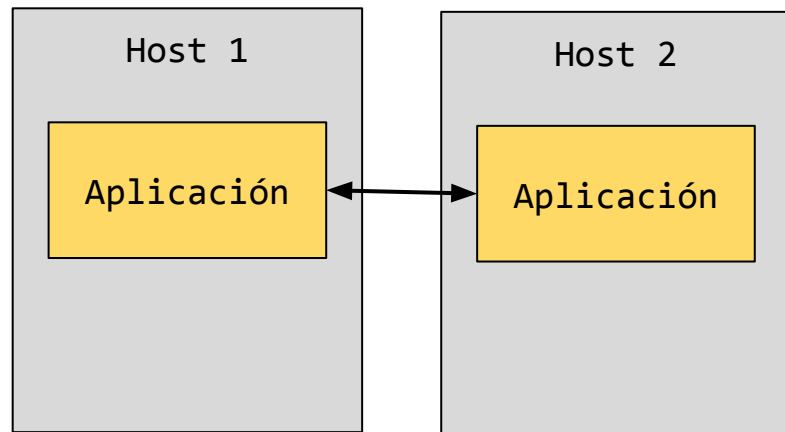
Técnica de programación ampliamente usada para desarrollar aplicaciones que se comunican entre sí mediante el pase de mensajes a través de una red.

Con la programación distribuida se pueden construir sistemas distribuidos.

Que es?

Técnica de programación ampliamente usada para desarrollar aplicaciones que se comunican entre si mediante el pase de mensajes a través de una red.

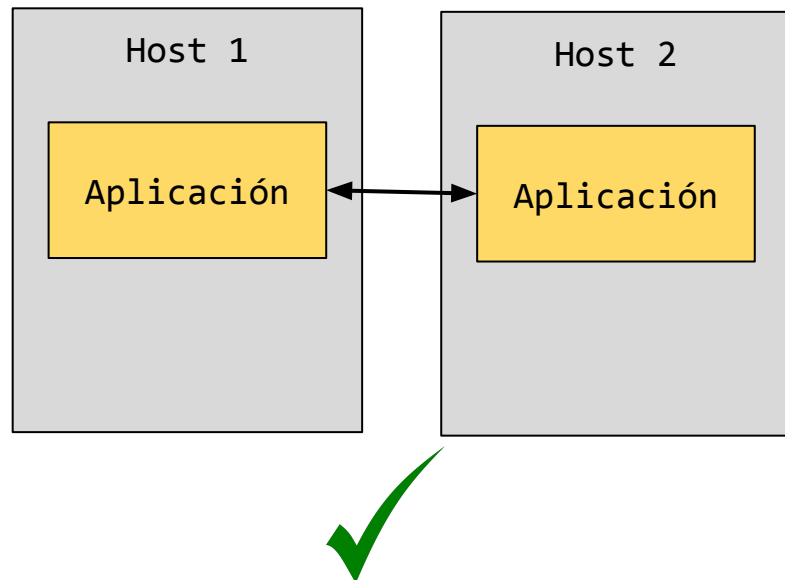
Con la programación distribuida se pueden construir sistemas distribuidos.



Que es?

Técnica de programación ampliamente usada para desarrollar aplicaciones que se comunican entre si mediante el pase de mensajes a través de una red.

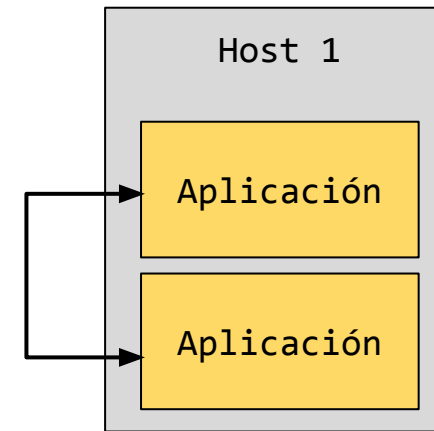
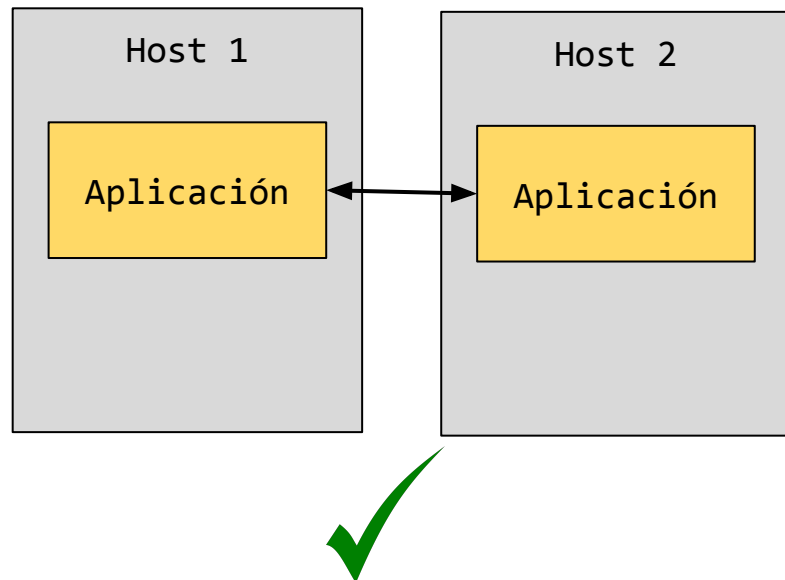
Con la programación distribuida se pueden construir sistemas distribuidos.



Que es?

Técnica de programación ampliamente usada para desarrollar aplicaciones que se comunican entre si mediante el pase de mensajes a través de una red.

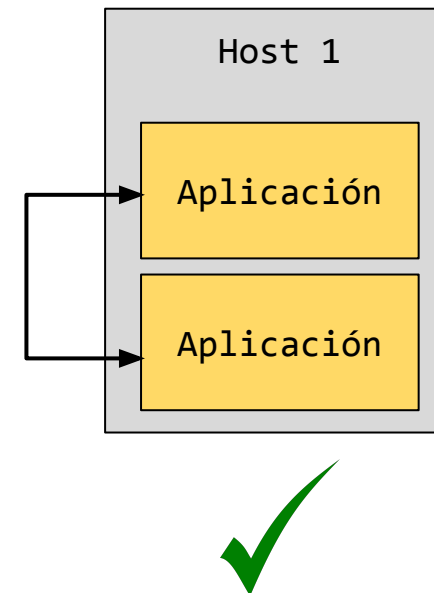
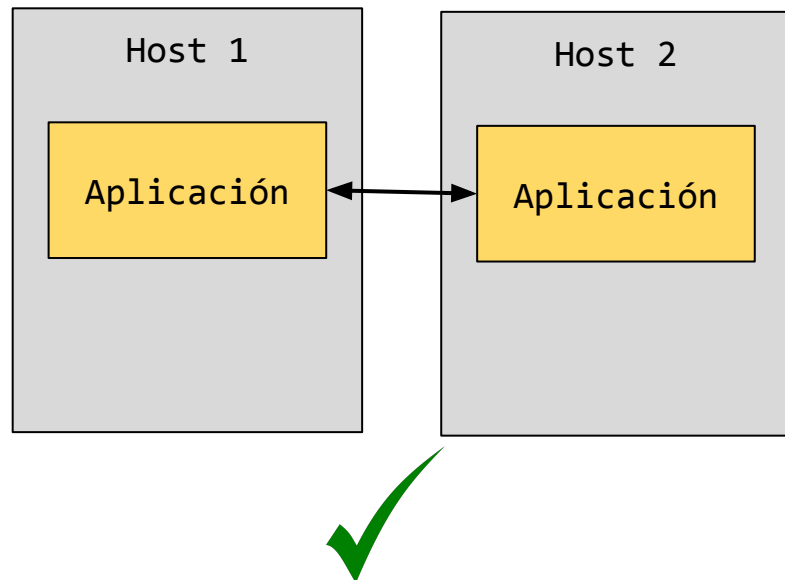
Con la programación distribuida se pueden construir sistemas distribuidos.



Que es?

Técnica de programación ampliamente usada para desarrollar aplicaciones que se comunican entre si mediante el pase de mensajes a través de una red.

Con la programación distribuida se pueden construir sistemas distribuidos.



Que es? – Nociones Básicas

- Computador paralelo
Capaz de ejecutar varias instrucciones simultáneamente
- Computación paralela
Uso de varios procesadores para resolver una tarea en común de forma simultánea
 - Cada procesador trabaja en una porción del problema
 - Los procesos pueden intercambiar datos
- Programación paralela
Considera aspectos conceptuales y las particularidades físicas de la computación paralela

Historia

Concurrencia: Paralelismo en potencia

- Múltiples procesadores
 - Programación paralela
 - C
 - Java
 - Scala
 - etc



Historia

Concurrencia: Paralelismo en potencia

- Múltiples procesadores
 - Programación paralela
 - C
 - Java
 - Scala
 - etc
 - MPI



Historia (MPI)

Concurrencia: Paralelismo en potencia

Message Passing Interface

- Estándar de envío de mensajes para programación paralela
- Librerías en Fortran y C principalmente
- Utilizado en universidades y “supercomputadoras”



Historia (MPI)

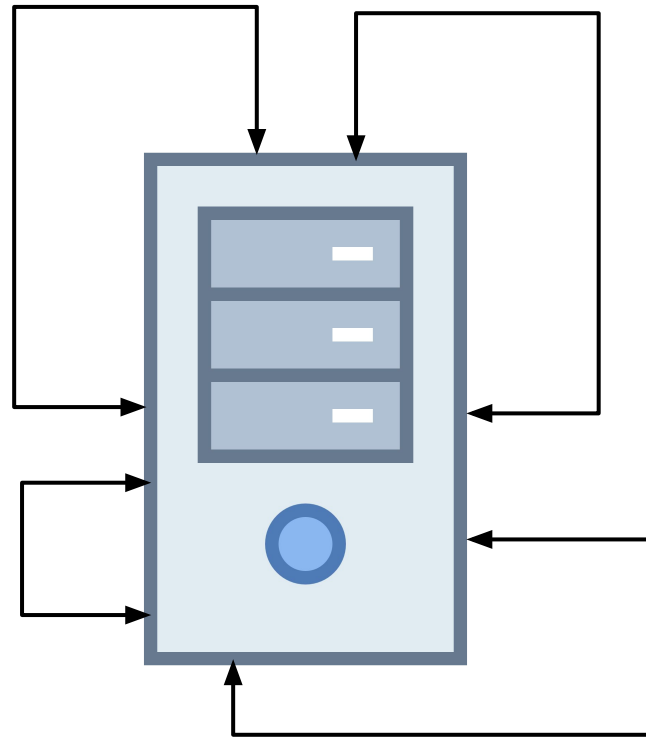
Concurrencia: Paralelismo en potencia

Message Passing Interface

Aprovechamiento de los recursos

Mayor capacidad de cómputo

Simple!



Historia (MPI)

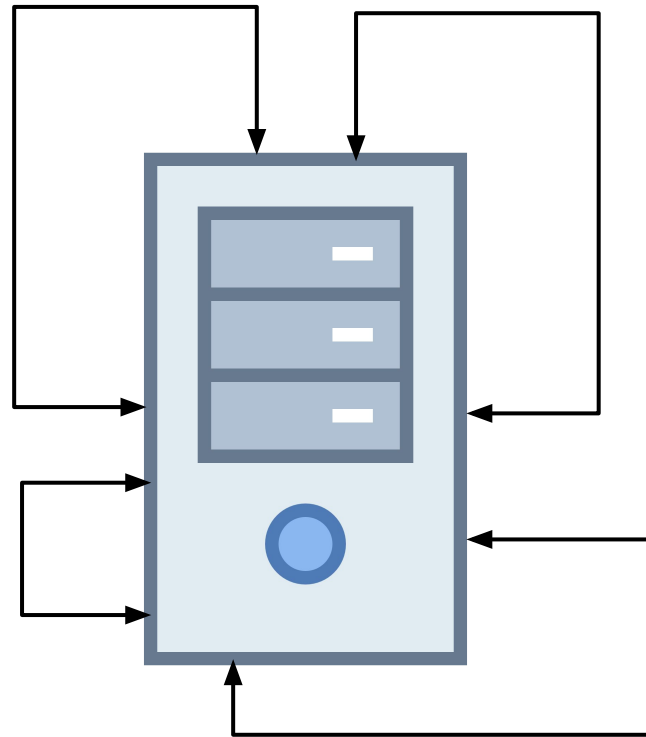
Concurrencia: Paralelismo en potencia

Message Passing Interface

Aprovechamiento de los recursos

Mayor capacidad de cómputo

Simple!



Una sola máquina

Retrabajo algorítmico

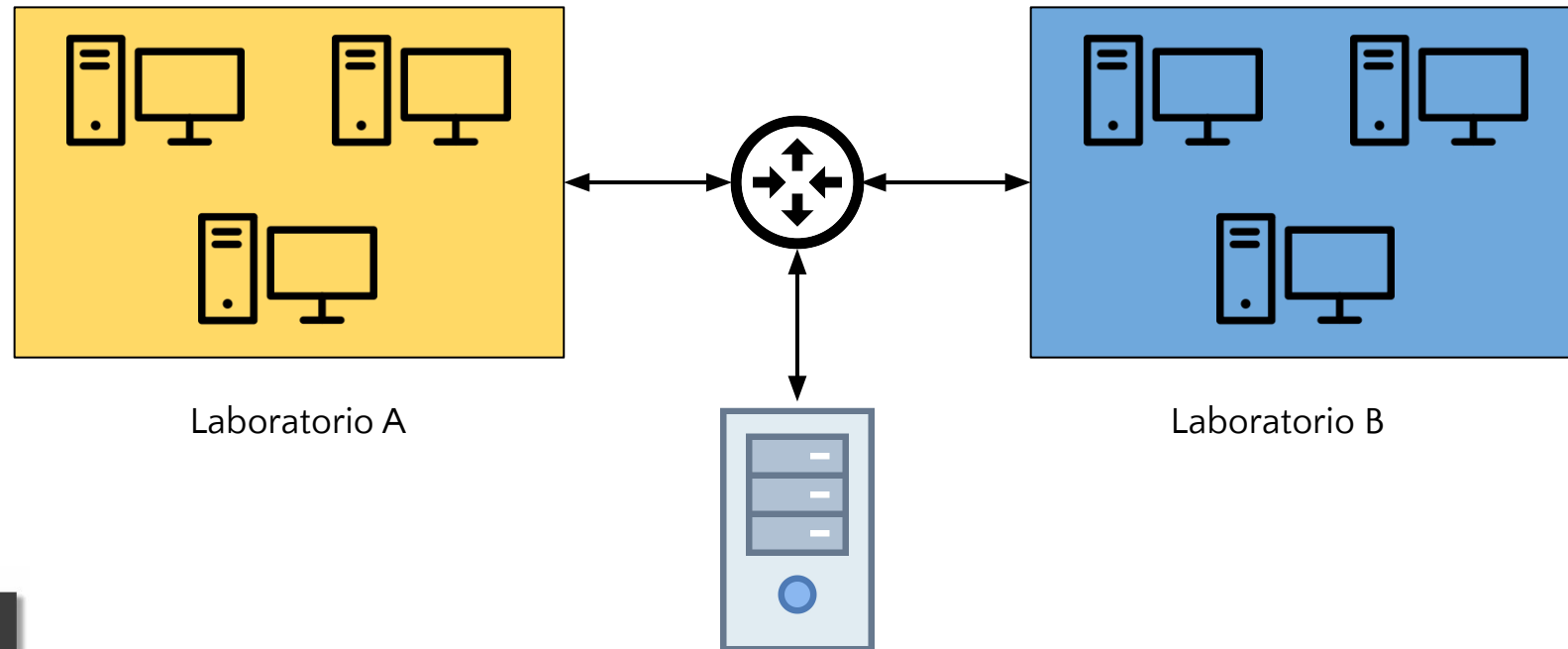
Complejo!



Historia (MPI)

Concurrencia: Paralelismo en potencia

Message Passing Interface (SCAN)



Historia

Concurrencia: Paralelismo en potencia

- Múltiples procesadores
 - Programación paralela
 - C
 - Java
 - Scala
 - etc



Historia

Concurrencia: Paralelismo en potencia

- Múltiples procesadores
 - Programación paralela
 - C
 - Java
 - Scala
 - etc
- Separación geográfica
 - Programación distribuida
 - Map reduce
 - Spark



Historia

Concurrencia: Paralelismo en potencia

- Múltiples procesadores
 - Programación paralela
 - C
 - Java
 - Scala
 - etc



- Separación geográfica
 - Programación distribuida
 - Map reduce
 - Spark
 - Zookeeper (Mención honorífica)



Que es? – Necesidad

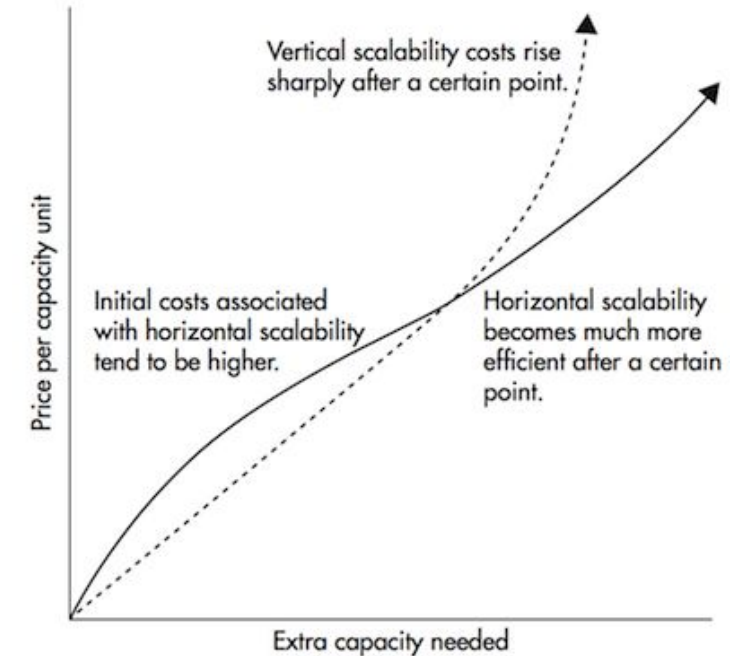
Limitaciones físicas

- Límite de la velocidad de la luz: Estancamiento en los incrementos en las frecuencias de reloj
- Límite de integración: Más frecuencia → Más consumo + Mayor temperatura

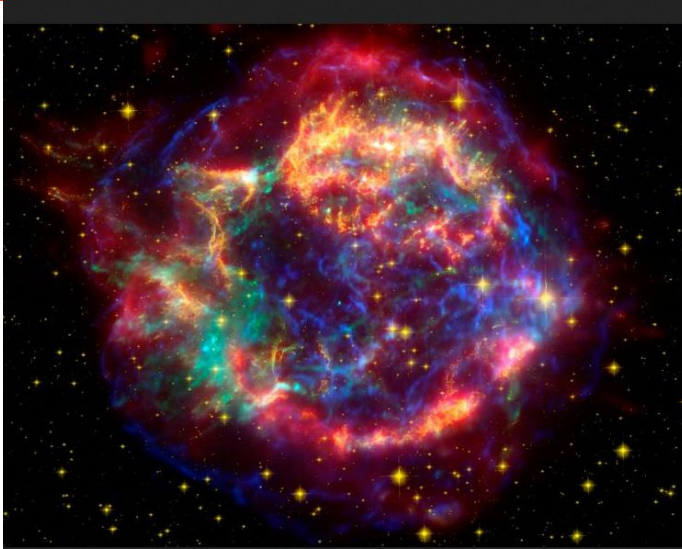
Problemas con alta complejidad

- Problemas de gran dimensión
- Problemas de tiempo real
- Problemas de gran desafío
 - Estudio del genoma humano
 - Problemas de complejidad NP-complete

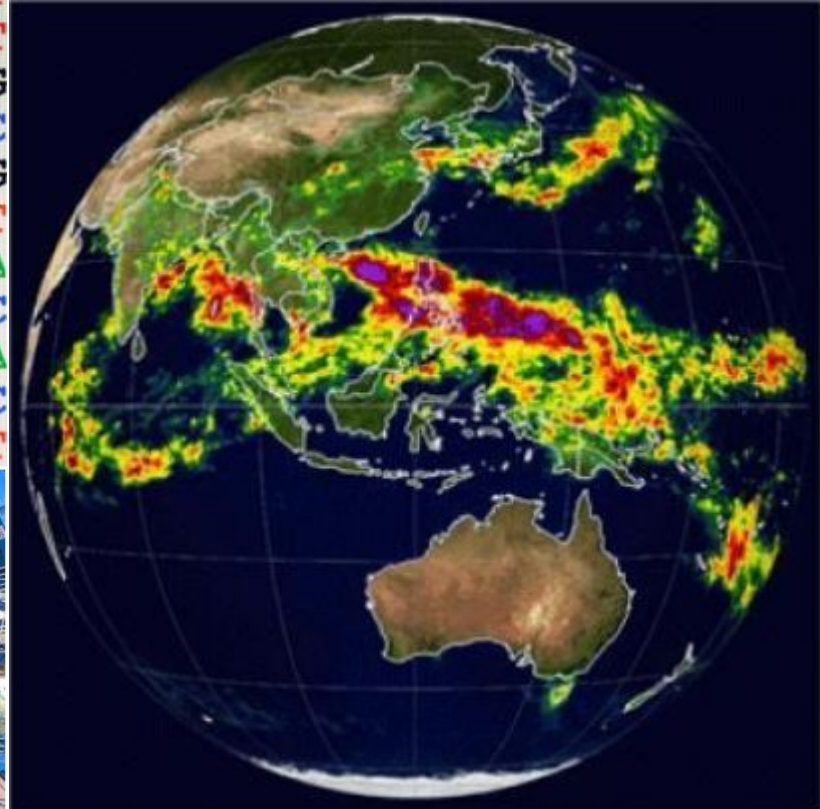
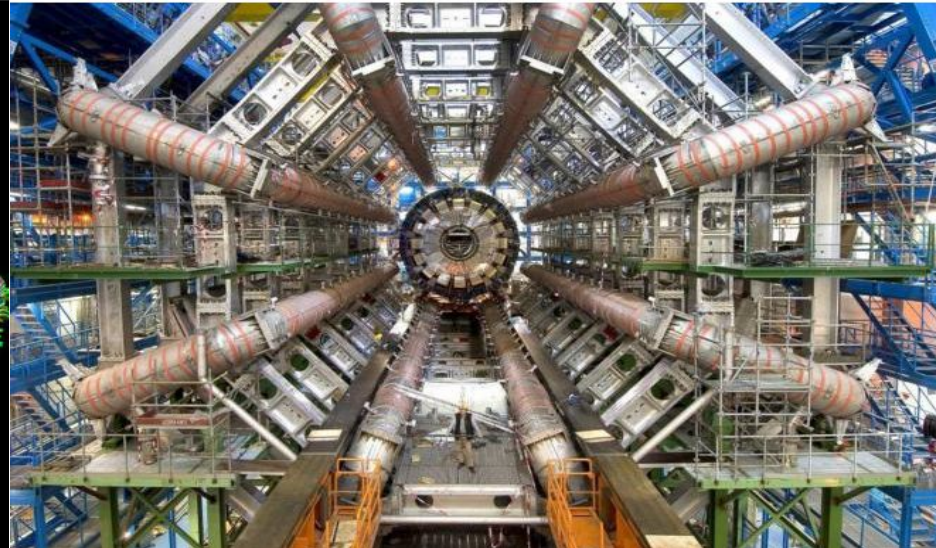
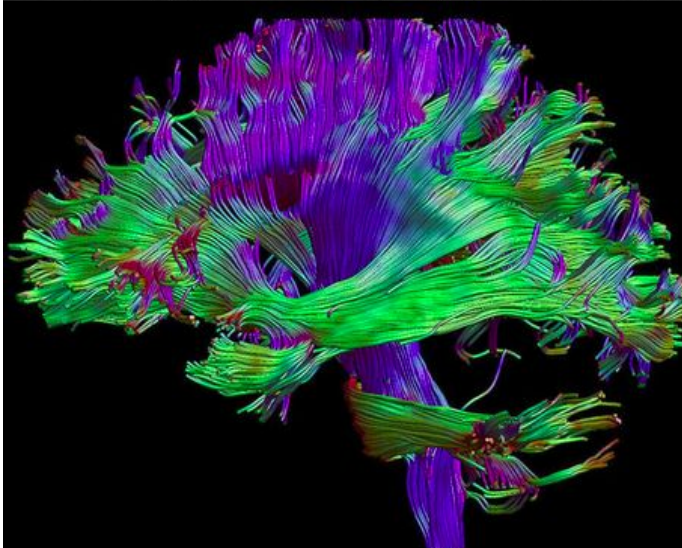
Costos



Que es? – Necesidad



ATCTCTTGGCTCCAGCATCGATGAAGAACGCA
TCATTTAGAGGAAGTAAAAGTCGTAACAAGGT
GAACGTGTCAAAACTTTTAAACAACGGATCTCTT
TGTTGCTTCGGCGGCGCCCGCAAGGGTGCCCG
GGCCTGCCGTGGCAGATCCCCAACGCCGGGCC
TCTCTTGGCTCCAGCATCGATGAAGAACGCAG
CAGCATCGATGAAGAACGCAGCGAAACGCGAT
CGATACTTCTGAGTGTTCTTAGCGAACTGTCA
CGGATCTCTTGGCTCCAGCATCGATGAAGAAC
ACAACGGATCTCTTGGCTCCAGCATCGATGAA
CGGATCTCTTGGCTCCAGCATCGATGAAGAAC
GATGAAGAACGCAGCGAAACGCGATATGTAAT



Que es?

La naturaleza de un sistema distribuido está dado por la filosofía de programación, no la separación física de sus componentes.

- No existe una definición formal que separe los conceptos de
 - Programación paralela
 - Programación concurrente
 - Programación distribuida

Arquitecturas de sistemas distribuidos:

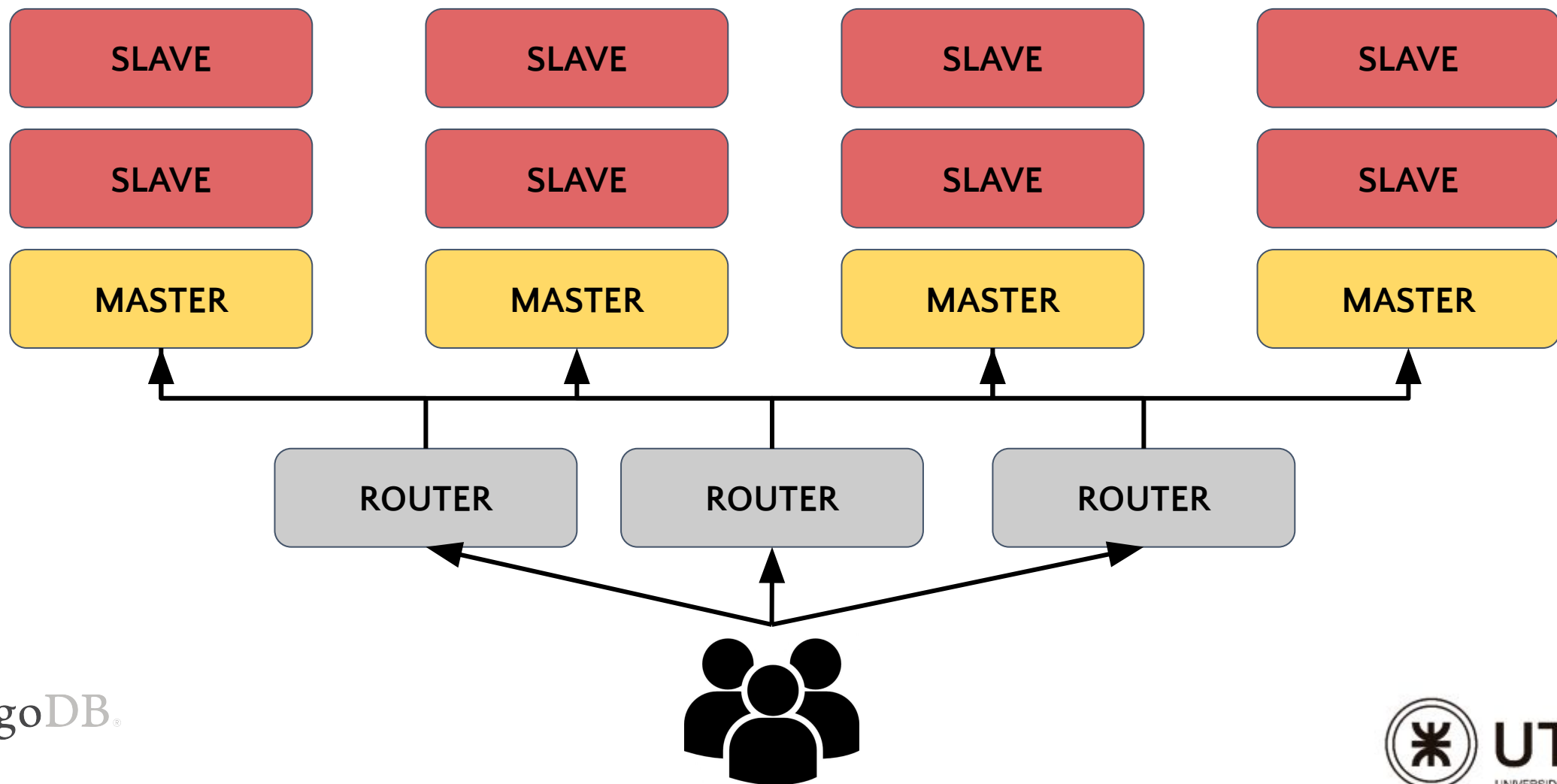
- Cliente-servidor
- 3 capas
- n capas
- peer to peer

Que es? – Ejemplos

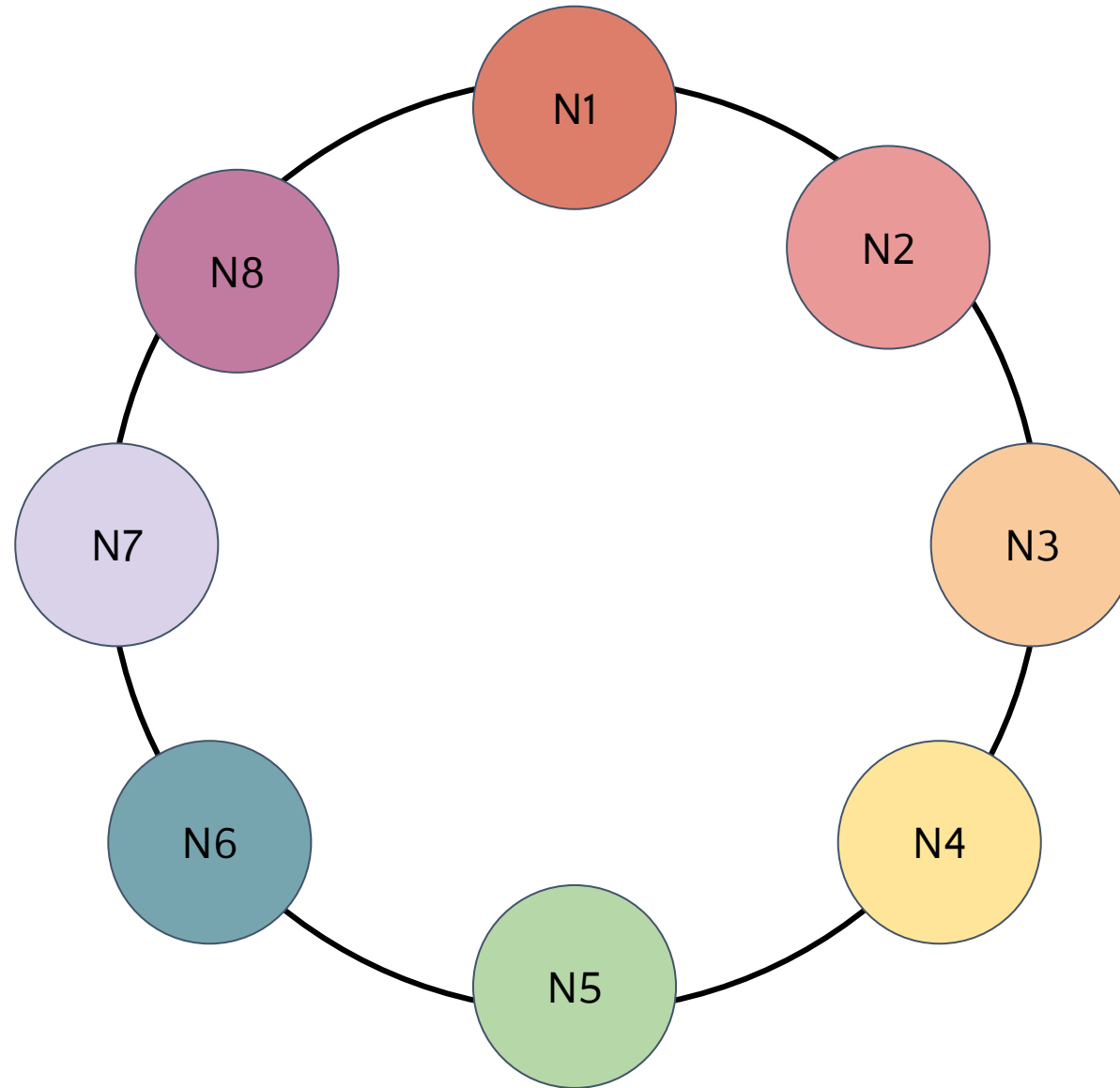
Otro tipo de sistemas distribuidos ampliamente usados son sistemas de storage y **bbdd NoSQL**:



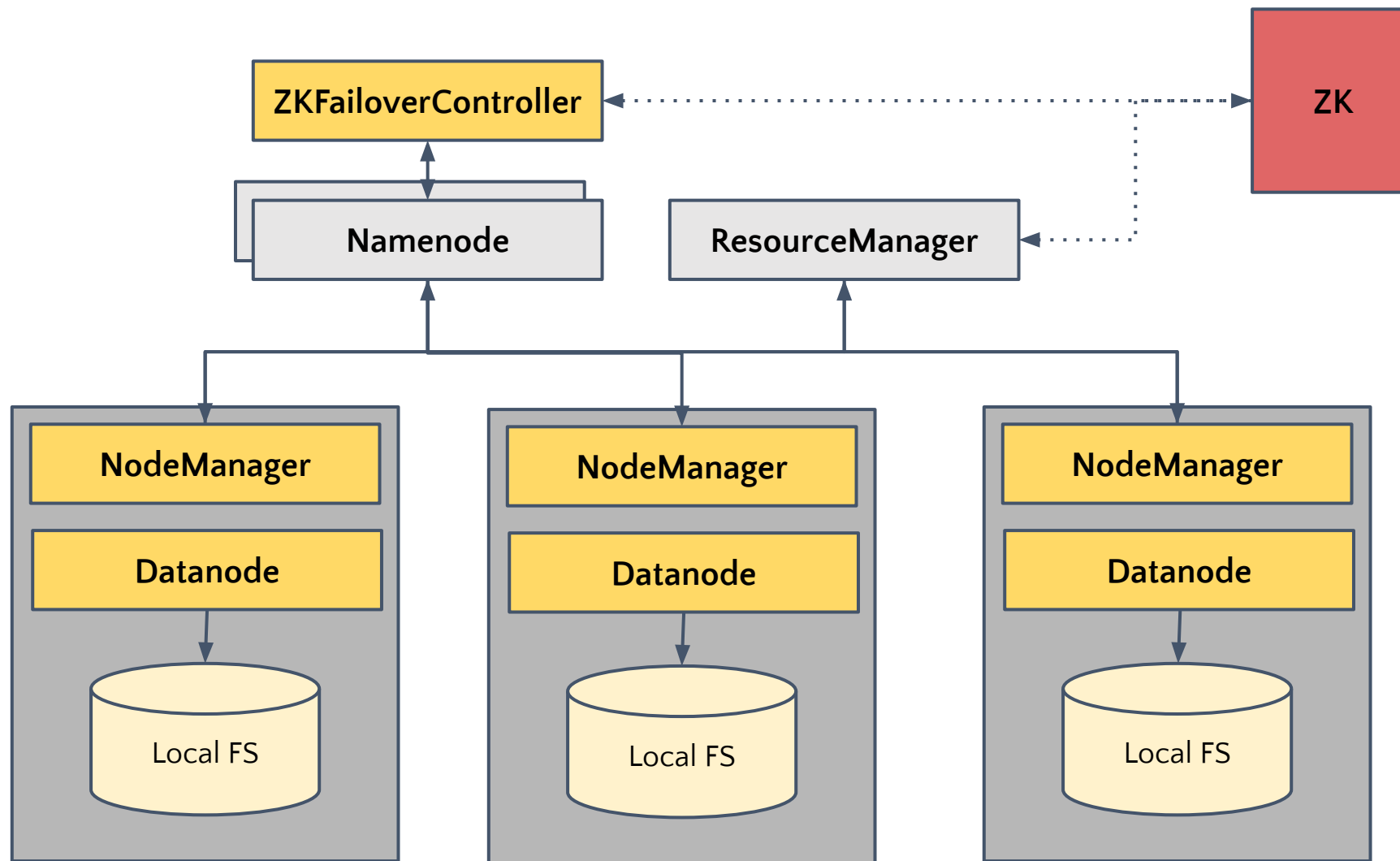
Que es?



Que es?

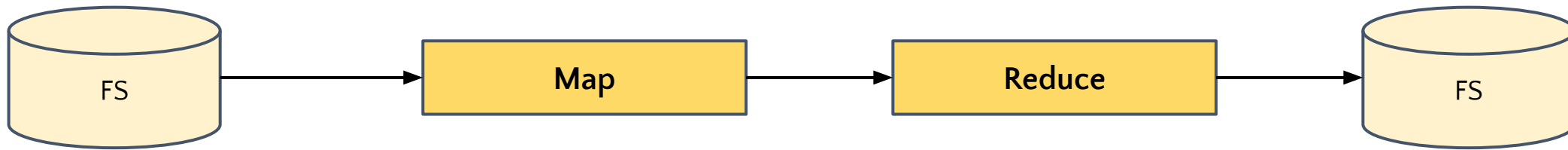


Que es?



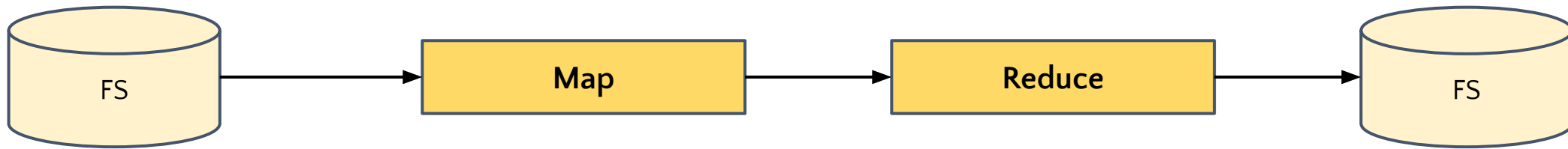
Map Reduce

Logicamente:

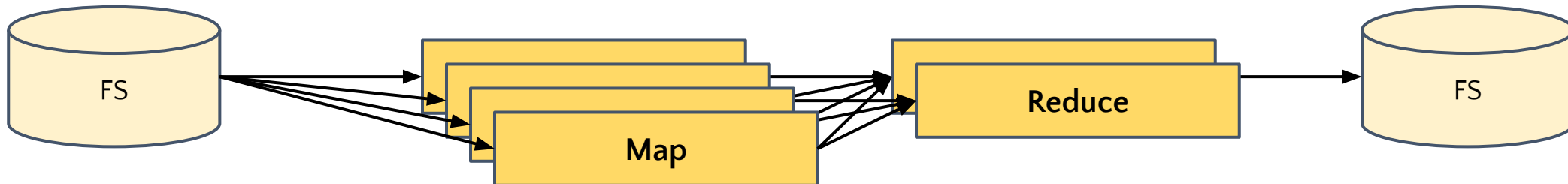


Map Reduce

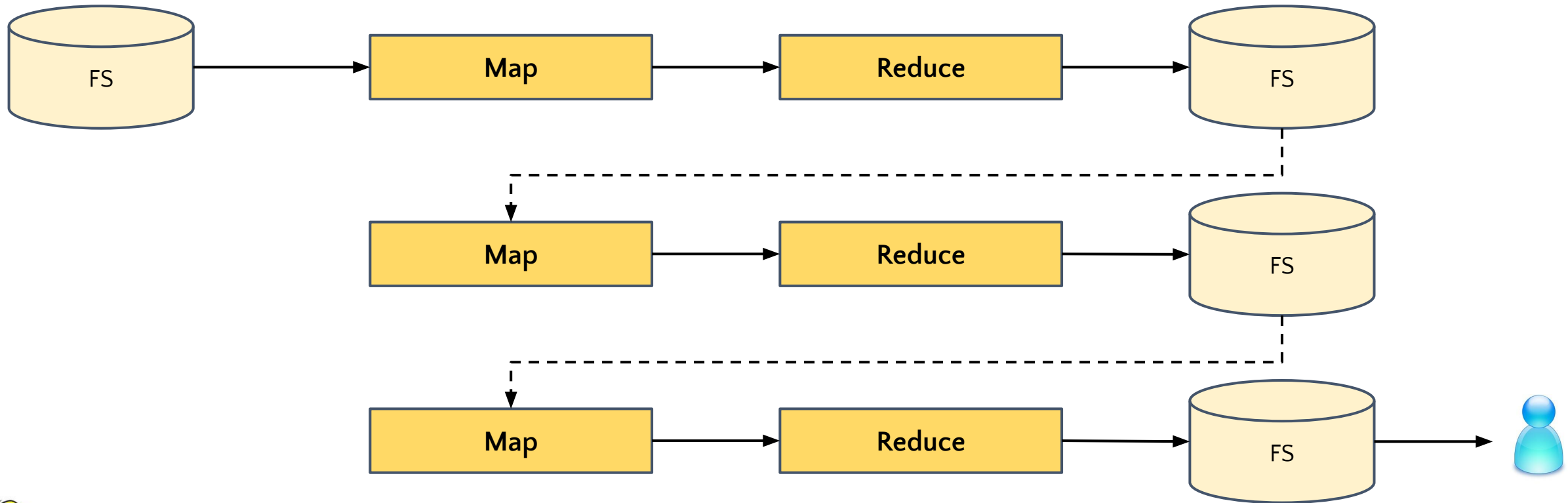
Logicamente:



Fisicamente:



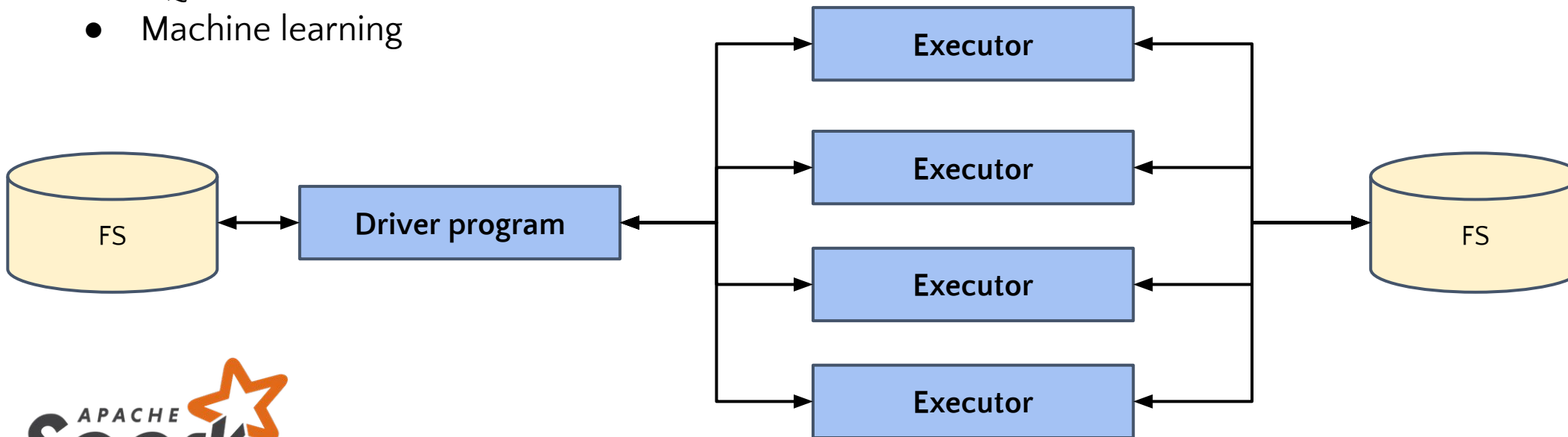
Map Reduce



Spark

Procesamiento distribuido de propósito general:

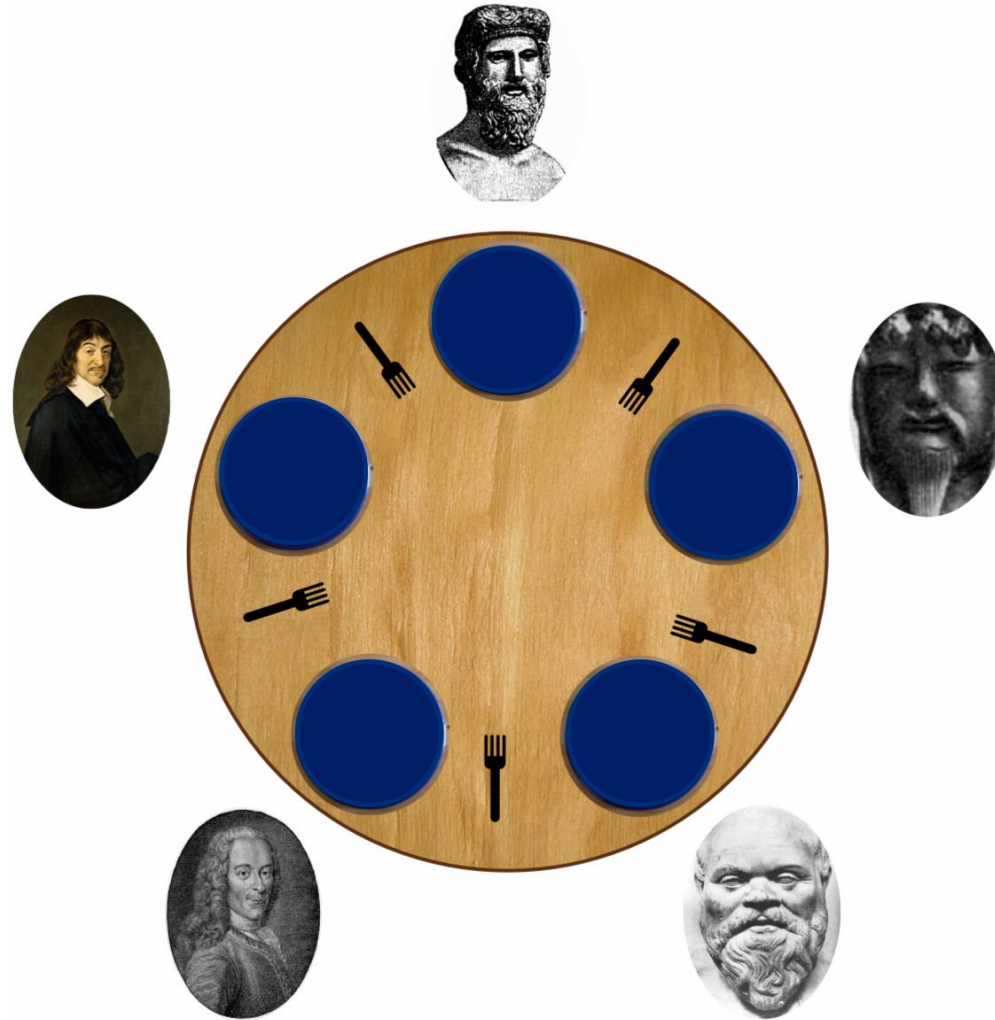
- Map reduce
- Métricas
- SQL
- Machine learning



Zookeeper



Dining Philosophers



Ventajas y desventajas

Ventajas

-

Desventajas

-

Ventajas y desventajas

Ventajas

- Ahorro de tiempo/dinero
- Capacidad para solucionar problemas más complejos
- Mejora en la eficiencia al dividir en subtarefas

Desventajas

- Mayor complejidad a la hora de encontrar una solución
- Se utilizan más recursos
- Problemas de sincronización entre procesos
- Dificultad de configuración de los nodos

Problemas actuales

- Wait-free consensus
- Oblivious Routing
- Stability of continuous consensus
- Complexity of implementing atomic Snapshots
- Pure Nash equilibria in Selfish Routing
- Adverse Cooperative Computing
- Distributed Approximations
- Sensor Networks: Locality for Geometric Graphs

Conclusiones

