



DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE



INTRODUCCIÓN

13169

SISTEMAS DISTRIBUIDOS



Motivación

- Inicio → Sistemas centralizados.
- Redes de área local (LAN).
- Redes de área amplia (WAN).
- Factibilidad de computo en red.
 - Redes de computadoras o ***sistemas distribuidos***.



Sistema distribuido

Un sistema distribuido es una colección de computadores independientes que dan al usuario la impresión de construir un único sistema coherente.



Sistemas distribuidos

- Consta de componentes.
- Usuarios interactúan con un único sistema.
 - Necesidad de colaboración entre los componentes.
- Computadores y forma de comunicación **ocultas** al usuario.
- Fáciles de expandir o escalar.
 - Computadores independientes.
 - No saben que forman parte de un sistema mayor.

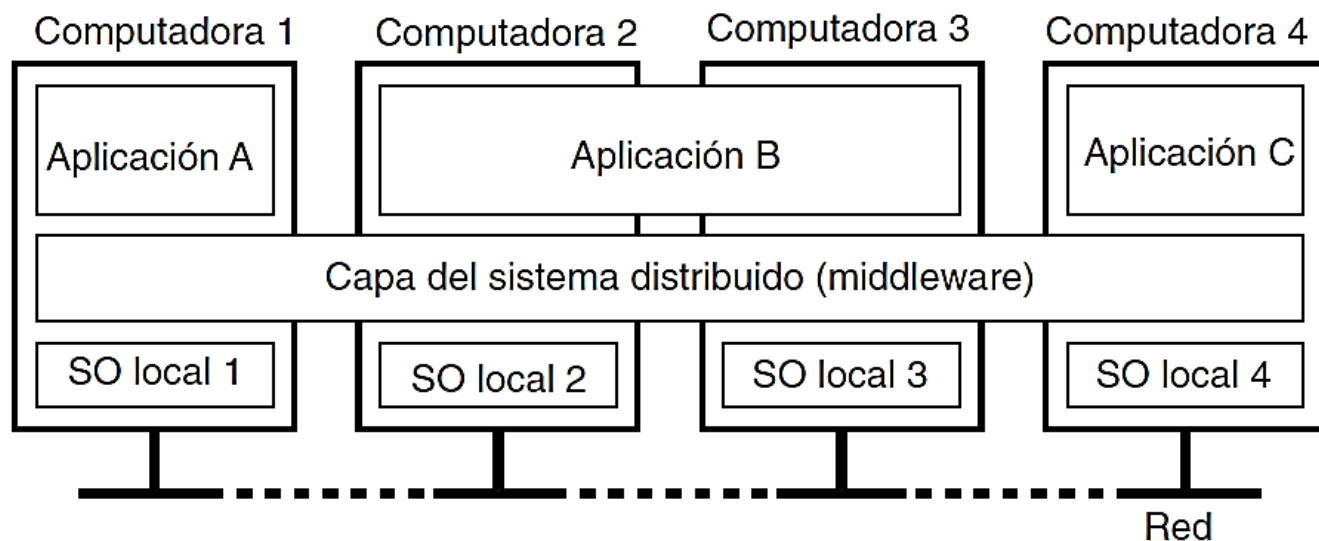


Figura 1-1. Un sistema distribuido organizado como middleware. La capa de middleware se extiende sobre diversas máquinas, y ofrece a cada aplicación la misma interfaz.



Middleware

- Objetivo de dar soporte a computadores y redes heterogéneas.
- Vista de un sistema único.
- Capa intermedia de software
 - Capa de alto nivel → Usuarios y aplicaciones.
 - Capa subyacente → SO y recursos básicos de comunicación.



Objetivos

- ***Facilitar el acceso a recursos.***
 - Razones económicas y de colaboración.
 - Conlleva problemas de seguridad y sincronización.
- ***Trasparencia en la distribución.***
 - Acceso.
 - Ubicación.
 - Migración.
 - Reubicación.
 - Replicación.
 - Concurrencia.
 - Falla



Transparencia	Descripción
Acceso	Oculto diferencias en la representación de los datos y la forma en que un recurso accede a los datos
Ubicación	Oculto la localización de un recurso
Migración	Oculto el que un recurso pudiera moverse a otra ubicación
Reubicación	Oculto el que un recurso pudiera moverse a otra ubicación mientras está en uso
Replicación	Oculto el número de copias de un recurso
Concurrencia	Oculto que un recurso puede ser compartido por varios usuarios que compiten por él
Falla	Oculto la falla y recuperación de un recurso

Figura 1-2. Distintas formas de transparencia aplicables en un sistema distribuido (ISO, 1995).

Transparencia	Descripción
Acceso	Oculta diferencias en la representación de los datos y la forma en que un recurso
Ubicación	
Migración	
Reubicación	
Replicación	
Concurrencia	
Falta	Oculta la falla y recuperación de un recurso

- Si bien todo esto es deseable, sobre todo desde el punto de vista del diseño, implica potenciales pérdidas de rendimiento y comprensión del sistema.
- Dependiendo del caso, podría ser mejor transparentar algunas características para poder entender de mejor forma el comportamiento del sistema.

Figura 1-2. Distintas formas de transparencia aplicables en un sistema distribuido (ISO, 1995).



Objetivos: Grado de apertura

- Sistema distribuido abierto es aquel que ofrece servicios bajo reglas estándar (sintaxis, semántica).
- **Interface Definition Language (IDL)**, lenguaje de especificación de componentes del sistemas.
 - Advanced Programming Interface (API).
 - Llamadas a procedimientos remotos (RPC).
- **Interoperabilidad**: Permite que procesos se comuniquen y generen implementaciones diferentes.
- **Portabilidad**: Extensiones pueden ser usadas en otros sistemas.
- **Fácil extensión**: Agregar más componentes.



Objetivos: Grado de apertura

- En la práctica muchos sistemas son monolíticos.
- Básicamente, son cerrados.
- Por ejemplo, **Web Cache**, donde los parámetros no pueden ser cambiados.
 - Tiempos de vida del documento.
 - Documento a cambiar cuando cache se llena.
 - No se toma decisiones respecto al contenido del documento.
- Separar política de la mecánica.

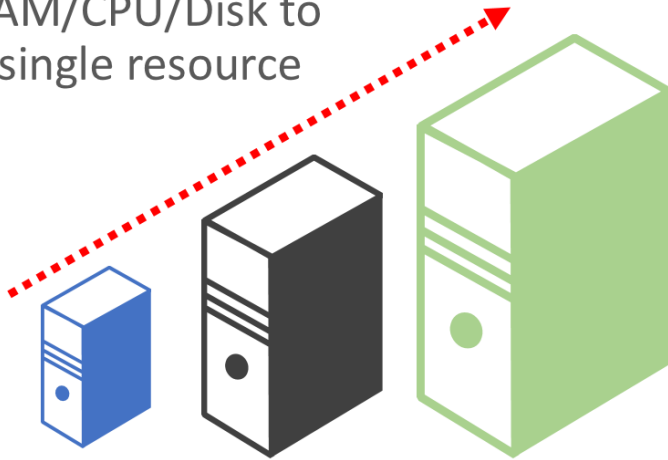


Objetivos: Escalabilidad

- Capacidad de adaptación y respuesta de un sistema con respecto al rendimiento del mismo a medida que aumentan de forma significativa el número de usuarios.
- *Escalabilidad vertical*
 - Aumentar capacidad de la máquina.
- *Escalabilidad horizontal*
 - Incluir más máquinas.
 - Conexión entre nuevas máquinas.

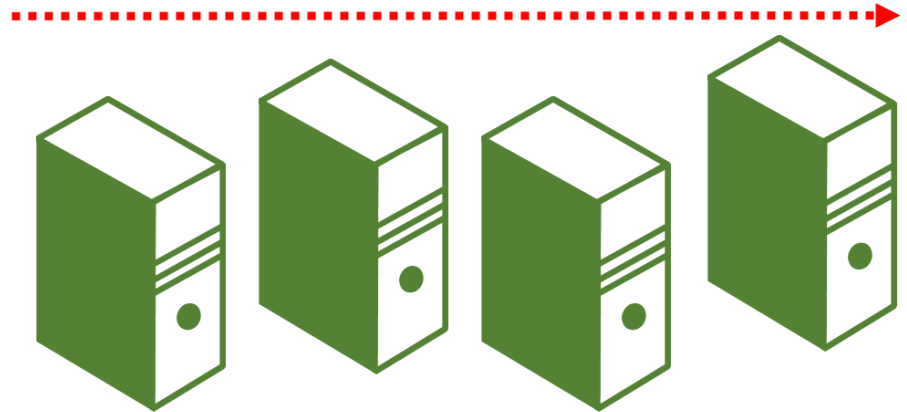
Scale Up (vertical scaling)

Increase capacity by adding
RAM/CPU/Disk to
a single resource



Scale Out (horizontal scaling)

Increase capacity by adding resources





Objetivos: Escalabilidad

- Tres técnicas para efectuar el escalamiento.
- *Ocultar latencia de sincronización*
 - Evitar lo más posible la espera de respuesta de peticiones remotas.
 - **Comunicación asíncrona.**
- *Distribución*
 - Dividir componentes en partes más pequeñas.
 - Dispersar las partes dentro del sistema.
- *Replicación*
 - **Caché** → Forma especial de replicación.
 - Potencial problema de **consistencia**.

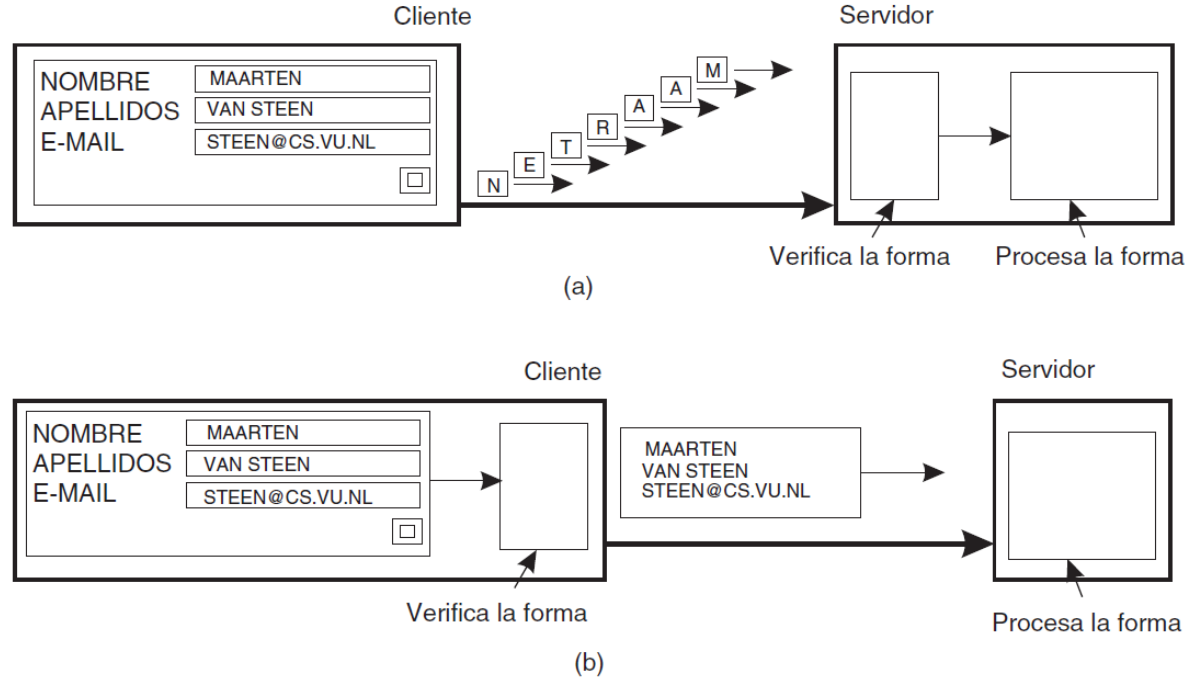


Figura 1-4. Diferencia entre dejar que (a) un servidor o (b) un cliente verifique formas mientras las llena.

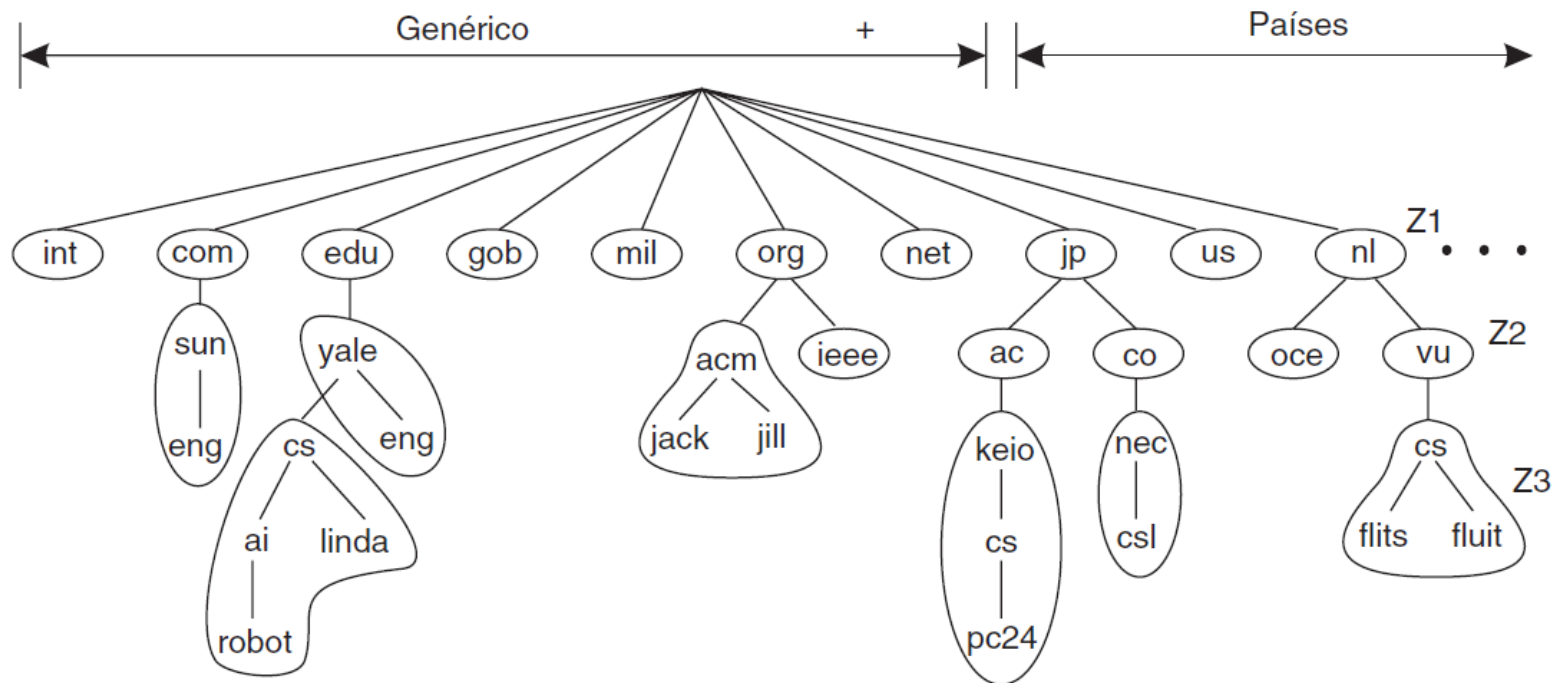
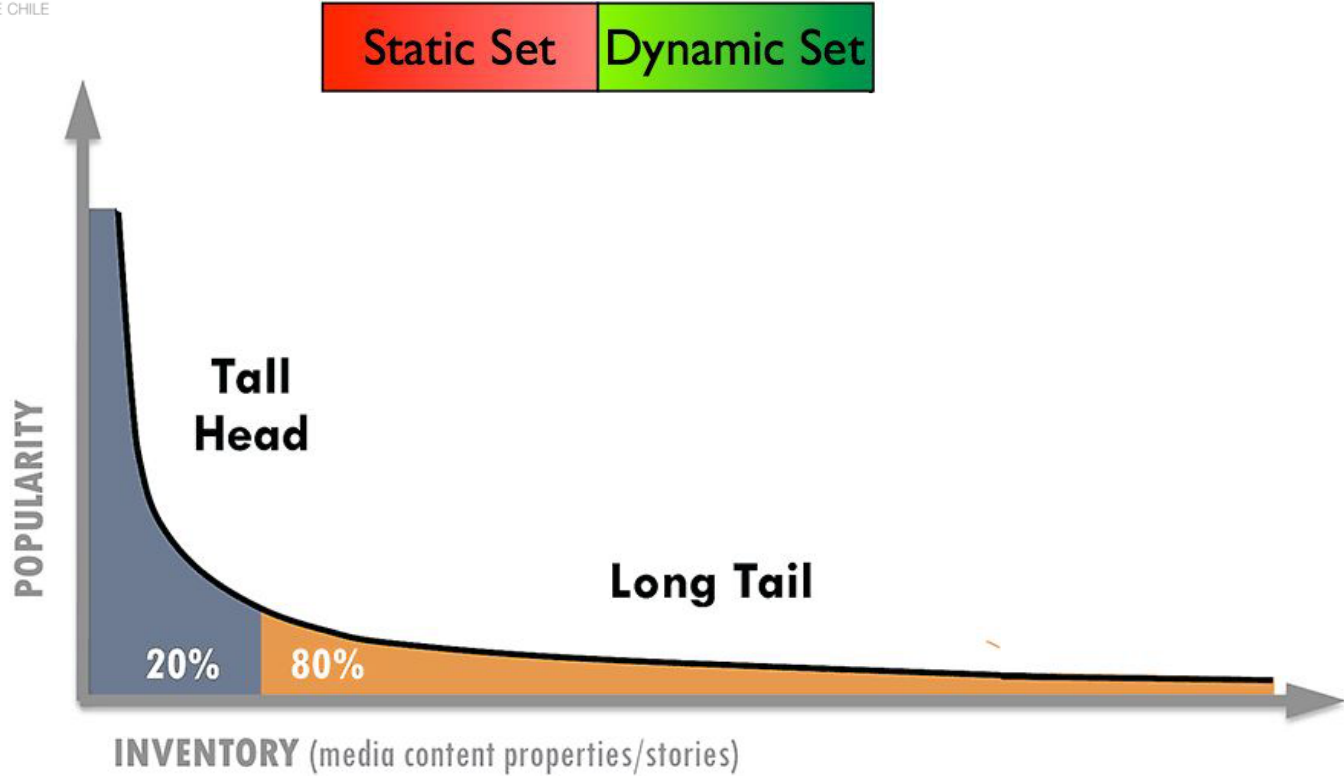


Figura 1-5. Ejemplo de cómo dividir el espacio de nombre DNS en zonas.





Trampas

- Componentes dispersos en la red
 - No considerar la dispersión causa problemas innecesarios.
-
1. La red es confiable.
 2. La red es segura.
 3. La red es homogénea.
 4. La topología no cambia.
 5. La latencia es igual a cero.
 6. El ancho de banda es infinito.
 7. El costo de transporte es igual a cero.
 8. Existe un administrador.



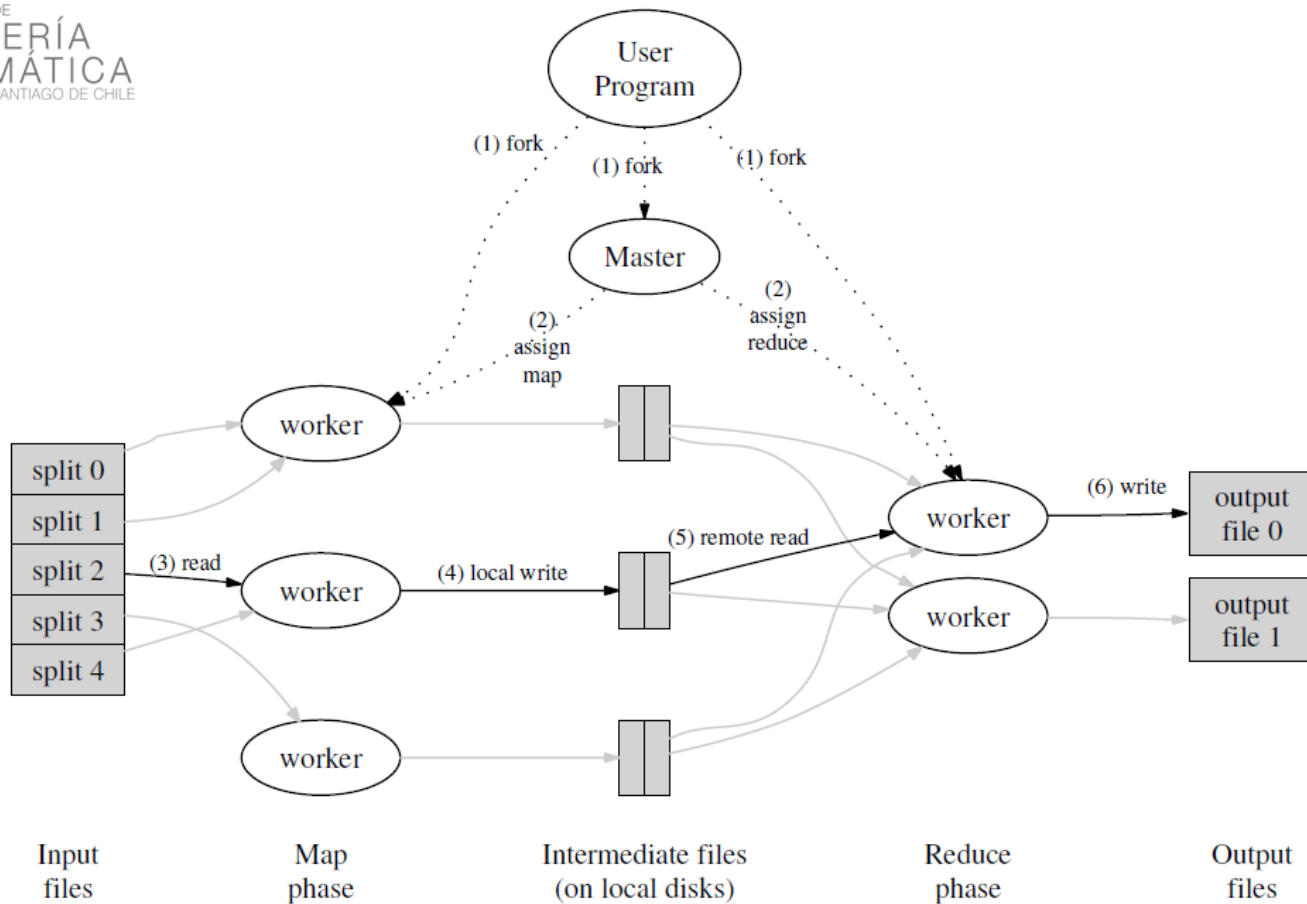
Caso estudio: Map Reduce

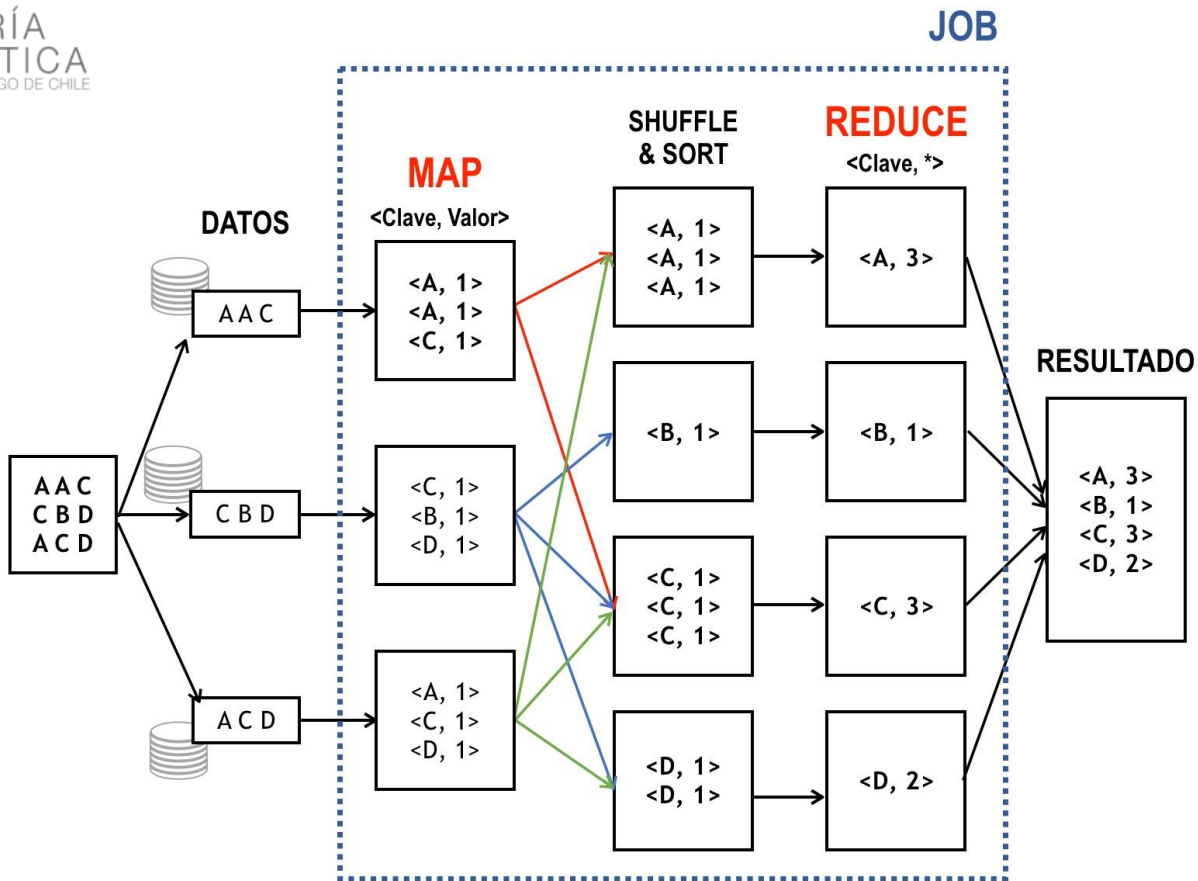
- Problema → Procesar grandes cantidades de datos.
- ***Buscan transparentar***
- Paralelización.
- Tolerancia de fallo.
- Distribución de los datos.
- Balance de carga.
- Generación de Map Reduce → Interfaz que automatiza la paralelización y distribución de computación de gran escala.



Caso estudio: Map Reduce

- *Función **Map***
 - Toma un par de valores y retorna valores en un dominio diferente.
 - $\text{Map}(k_1, v_1) \rightarrow \text{list}(k_2, v_2)$
- *Función **Reduce***
 - Produce una colección de valores para cada dominio.
 - $\text{Reduce}(k_2, \text{list}(v_2)) \rightarrow \text{list}(v_2)$
- Ambas funciones son programables por el usuario.







DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Departamento de Ingeniería Informática
Universidad de Santiago de Chile

¿CONSULTAS?