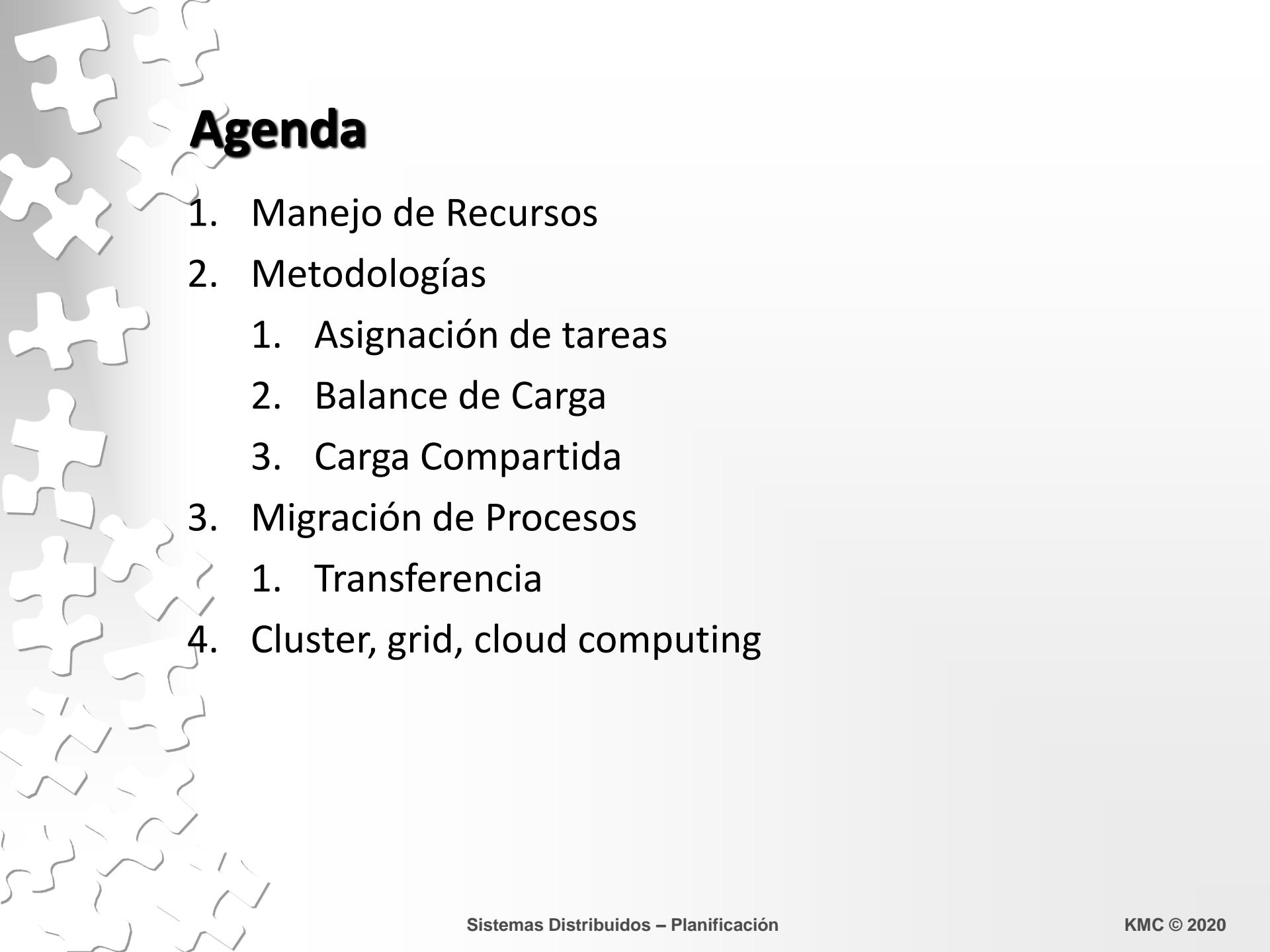




Sistemas Distribuidos

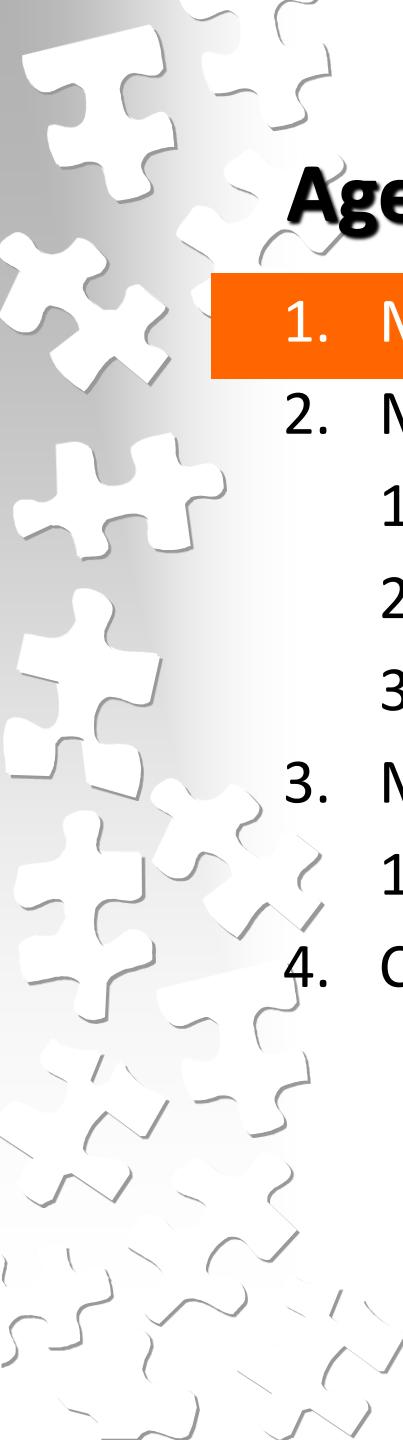
Módulo 5

Planificación de Procesos en Sistemas Distribuidos



Agenda

1. Manejo de Recursos
2. Metodologías
 1. Asignación de tareas
 2. Balance de Carga
 3. Carga Compartida
3. Migración de Procesos
 1. Transferencia
4. Cluster, grid, cloud computing



Agenda

1. Manejo de Recursos
2. Metodologías
 1. Asignación de tareas
 2. Balance de Carga
 3. Carga Compartida
3. Migración de Procesos
 1. Transferencia
4. Cluster, grid, cloud computing



Manejo de Recursos

Se diseña un manejador de recursos para:

- Control de asignación de recursos vs. procesos.
- Ruteo de procesos a sitios de acuerdo al asignación.

El objetivo es *optimizar*:

- El uso.
- Tiempo de respuesta.
- Congestión de la red.
- Sobrecarga de la planificación.

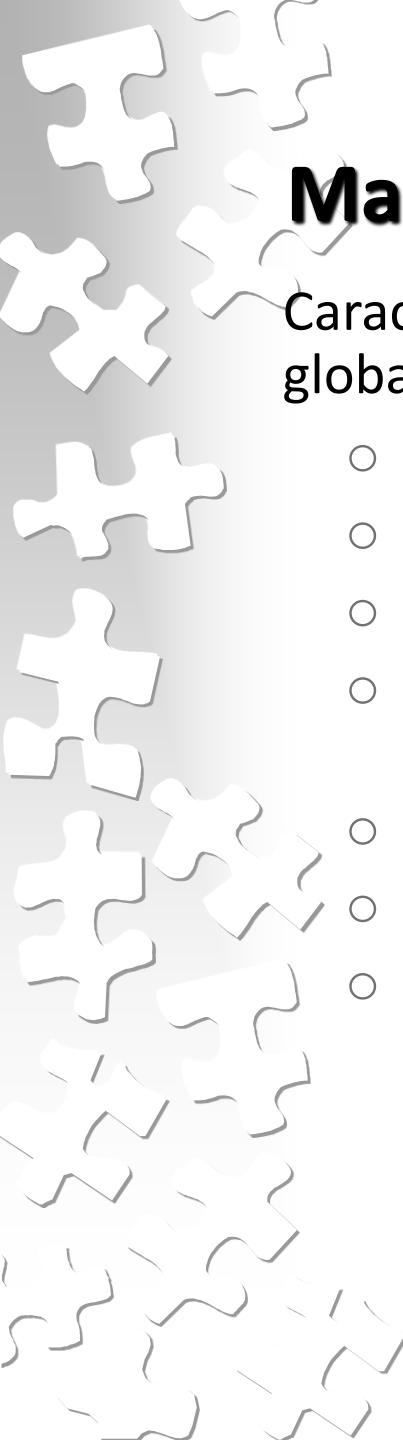


Manejo de Recursos

Técnica y metodologías para la planificación de procesos:

- Asignación de tareas
- Balance de carga
- Carga compartida

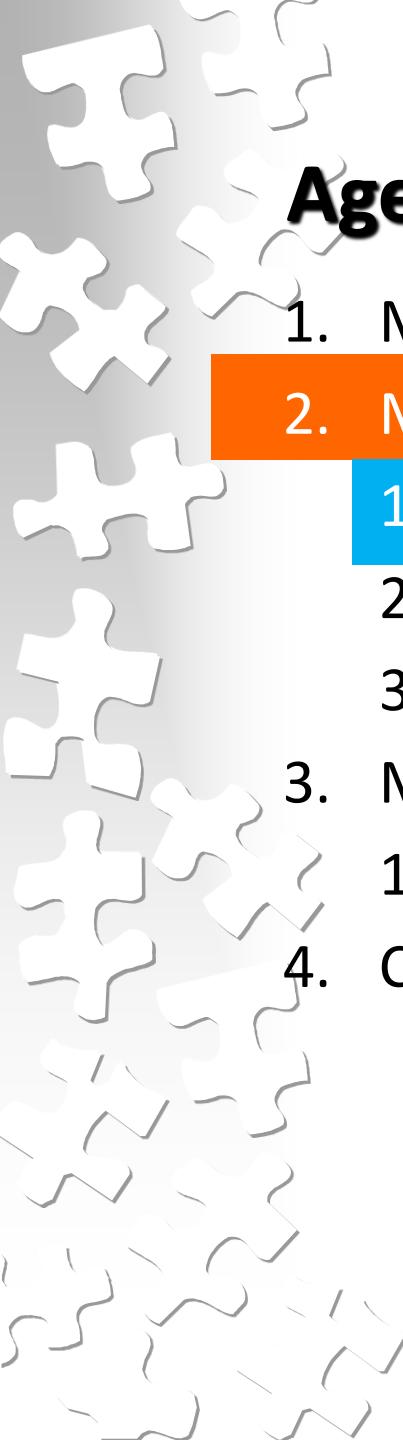
La primer técnica tiene limitada aplicabilidad.



Manejo de Recursos

Características deseables para un buen algoritmo de planificación global serían:

- No tener conocimiento “a priori” sobre los procesos.
- Dinámico en su naturaleza.
- Capacidad de tomar decisiones rápidas.
- Rendimiento de sistema y sobrecarga de la planificación balanceada.
- Estabilidad.
- Tolerancia a las fallas.
- Imparcialidad en el servicio.



Agenda

1. Manejo de Recursos
2. Metodologías
 1. Asignación de tareas
 2. Balance de Carga
 3. Carga Compartida
3. Migración de Procesos
 1. Transferencia
4. Cluster, grid, cloud computing



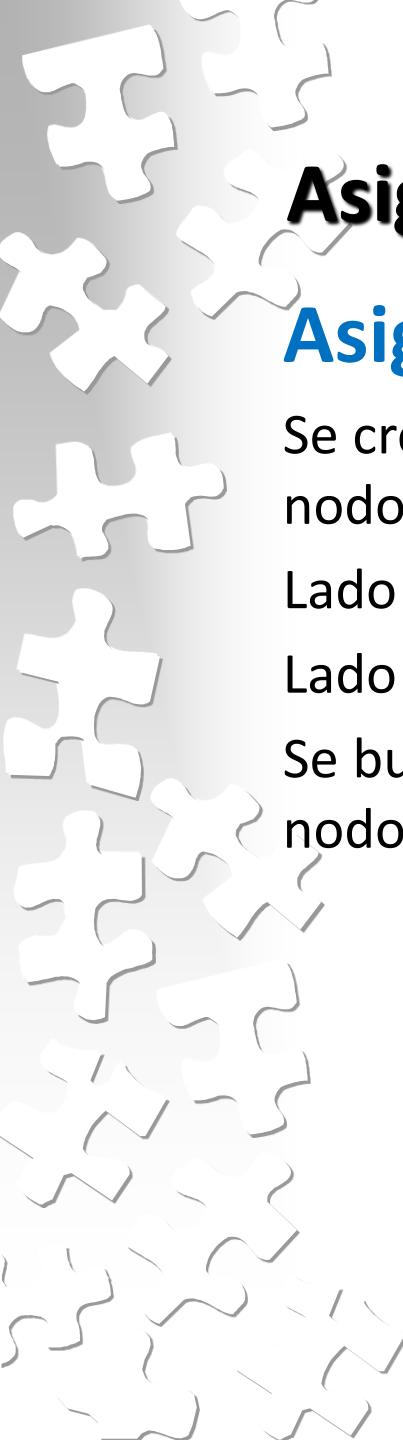
Asignación de Tareas

Se considera a un proceso compuesto por múltiples tareas.

Objetivo

- Minimización de costos de IPC
- Minimización de tiempos de retorno
- Alto grado de paralelismo
- Utilización eficiente de los recursos

Puede haber conflicto entre estos objetivos



Asignación de Tareas

Asignación óptima

Se crea un grafo estático de asignación con sitios y tareas como nodos.

Lado **tarea-nodo** es el costo de procesamiento.

Lado **tarea-tarea** es el costo de IPC.

Se busca un **cutset** que sea mínimo con la condición de que los nodos de una partición P_j son todos alcanzables desde n_i .



Asignación de Tareas

- Consideremos el siguiente sistema ejemplo, presentado por Stone [1977]:
 - Seis tareas $\{t_1, t_2, t_3, t_4, t_5, t_6\}$
 - Dos nodos $\{n_1, n_2\}$

Asignación de Tareas

COSTOS DE EJECUCIÓN		
	nodos	
	n_1	n_2
t_1	5	10
t_2	2	∞
t_3	4	4
t_4	6	3
t_5	5	2
t_6	∞	4

	t_1	t_2	t_3	t_4	t_5	t_6
t_1	0	6	4	0	0	12
t_2	6	0	8	12	3	0
t_3	4	8	0	0	11	0
t_4	0	12	0	0	5	0
t_5	0	3	11	5	0	0
t_6	12	0	0	0	0	0

Asignación de Tareas

Asignación serial

Tarea	Sitio
t_1	n_1
t_2	n_1
t_3	n_1
t_4	n_2
t_5	n_2
t_6	n_2

Costo de ejecución = 20

Costo de comunicación = 38

Costo total = 58

Asignación óptima

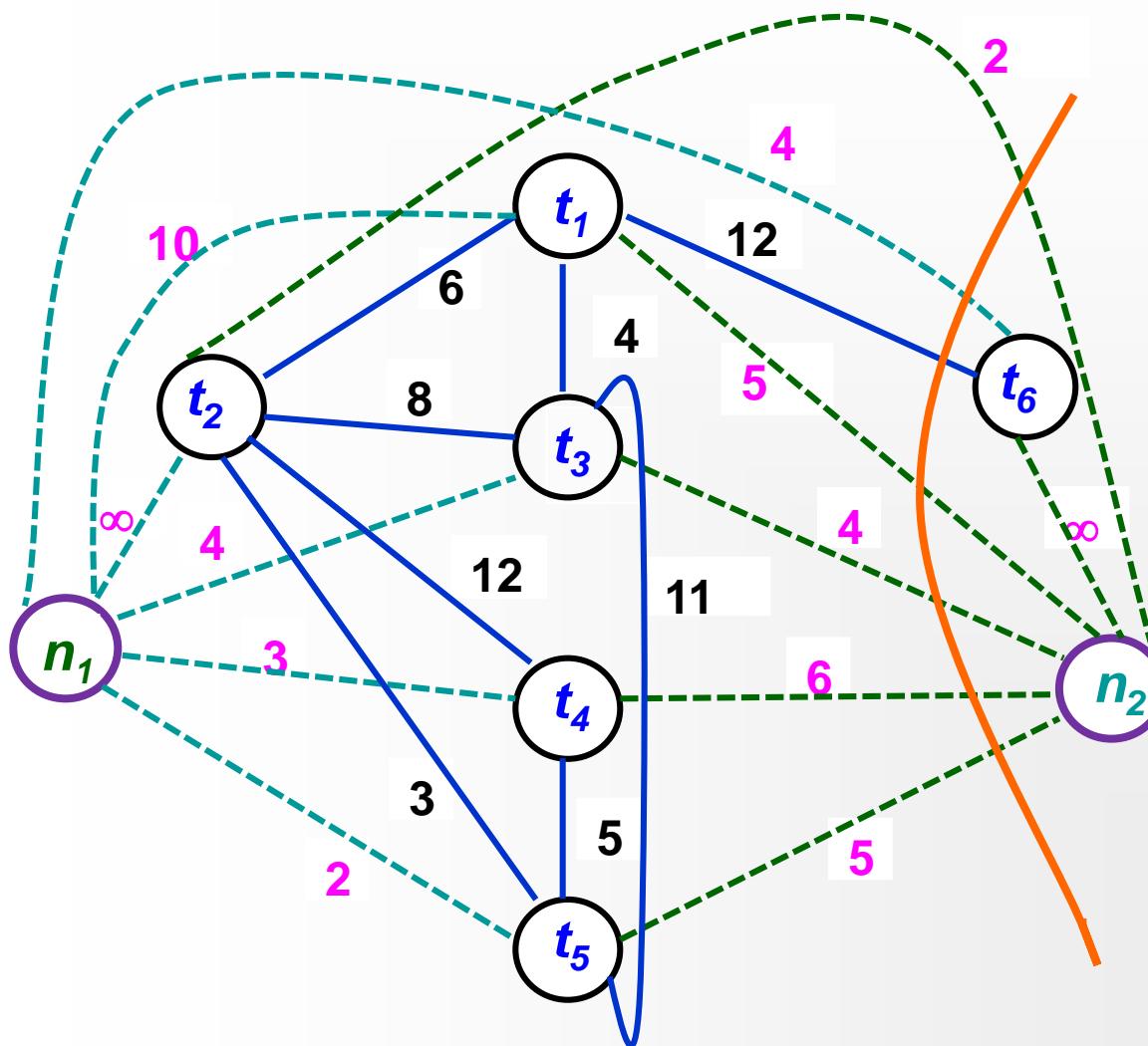
Tarea	Sitio
t_1	n_1
t_2	n_1
t_3	n_1
t_4	n_1
t_5	n_1
t_6	n_2

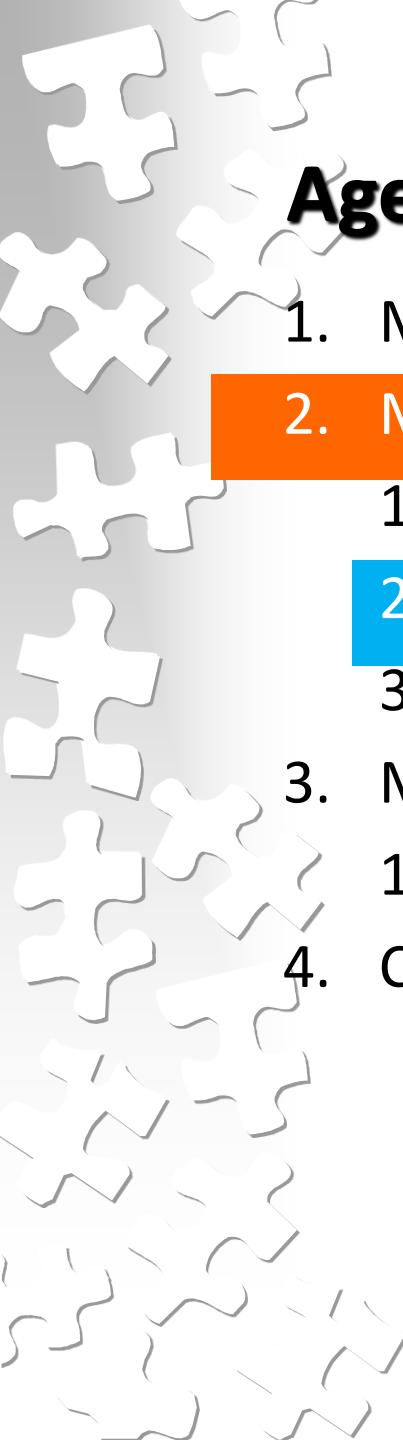
Costo de ejecución = 26

Costo de comunicación = 12

Costo total = 38

Manejo de Recursos – Asignación de Tareas





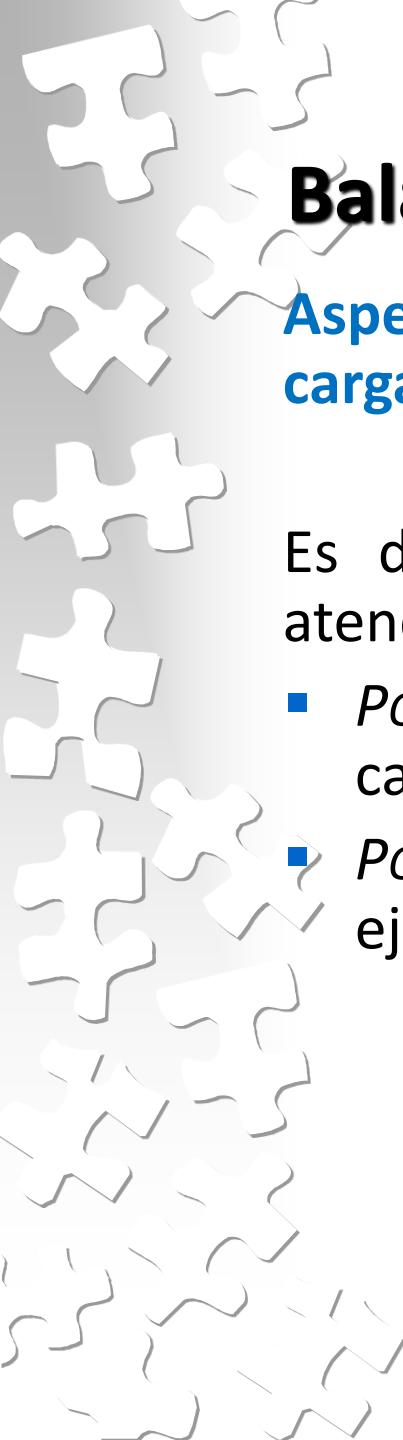
Agenda

1. Manejo de Recursos
2. Metodologías
 1. Asignación de tareas
 2. Balance de Carga
 3. Carga Compartida
3. Migración de Procesos
 1. Transferencia
4. Cluster, grid, cloud computing

Manejo de Recursos – Balance de Carga

Taxonomía





Balance de Carga

Aspectos a tener en cuenta en el diseño de balance de carga.

Es difícil obtener un “buen algoritmo” porque se debe atender a:

- *POLÍTICA DE ESTIMACIÓN DE CARGA* determina como estimar la carga de un sitio del sistema.
- *POLÍTICA DE TRANSFERENCIA DE PROCESOS* si el proceso se debe ejecutar local o remotamente.

Balance de Carga- POLÍTICA DE ESTIMACIÓN DE CARGA

La estimación puede basarse en factores dependientes del tiempo o del sitio:

- Número total de procesos en el sitio en el momento de estimación.
- Demanda de recursos de esos procesos.
- Arquitectura y velocidad del sitio procesador.



La mejor estimación es la utilización de CPU

Balance de Carga- POLÍTICAS DE TRANSFERENCIAS DE PROCESOS

Idea: *transferir procesos de sitios muy cargados a otros descargados*

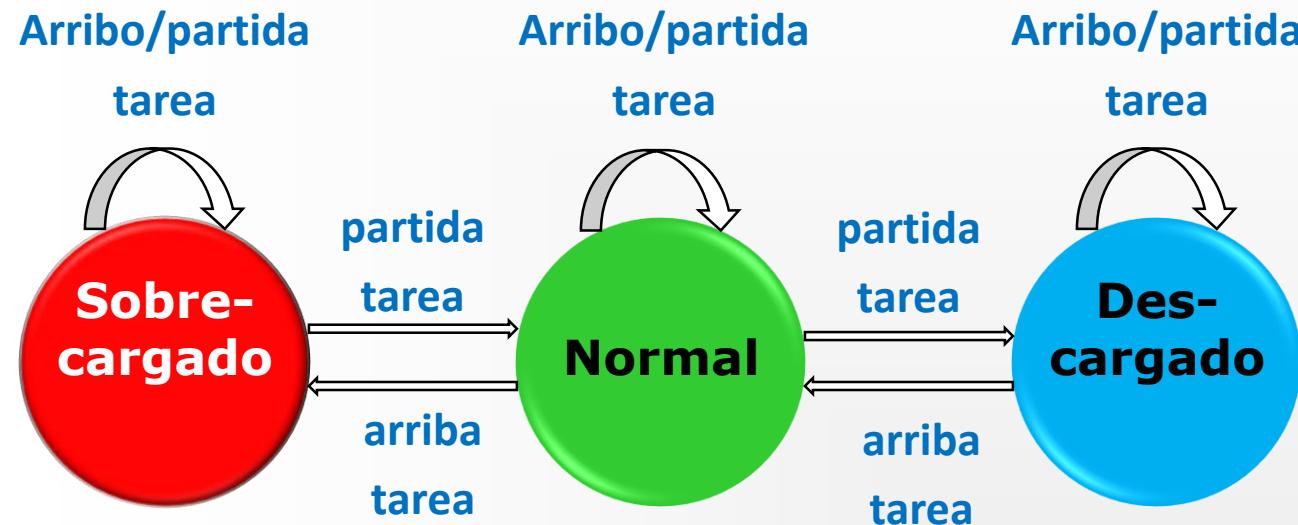
Problema: ¿cómo decidir que un sitio está más cargado que otro?

Se utiliza la política del **umbral**.

Este valor es el límite entre **pesados** y **livianos**.

POLÍTICA ESTÁTICA	cada sitio tiene un valor predefinido de umbral dependiendo de su capacidad de procesamiento. No requiere intercambio de información entre sitios.
POLÍTICA DINÁMICA	el umbral de cada sitio n_i se calcula como producto de la carga media de todos los sitios y una constante c_i predefinida (depende de la capacidad de procesamiento del sitio n_i relativo a los otros). Los sitios deben intercambiar información sobre sus estados.

Balance de Carga- POLÍTICAS DE TRANSFERENCIAS DE PROCESOS





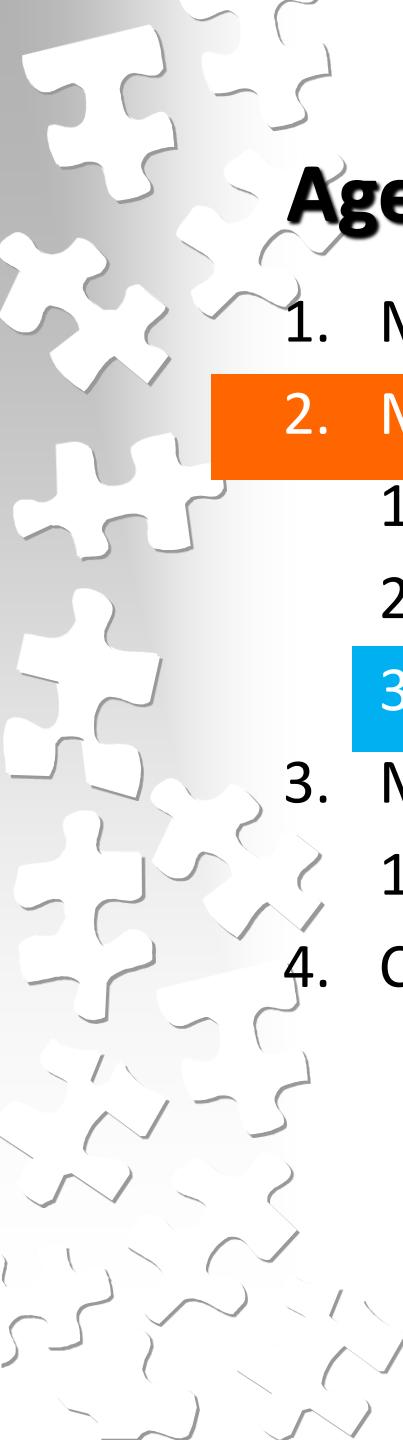
Balance de Carga

Aspectos a tener en cuenta en el diseño de balance de carga

...

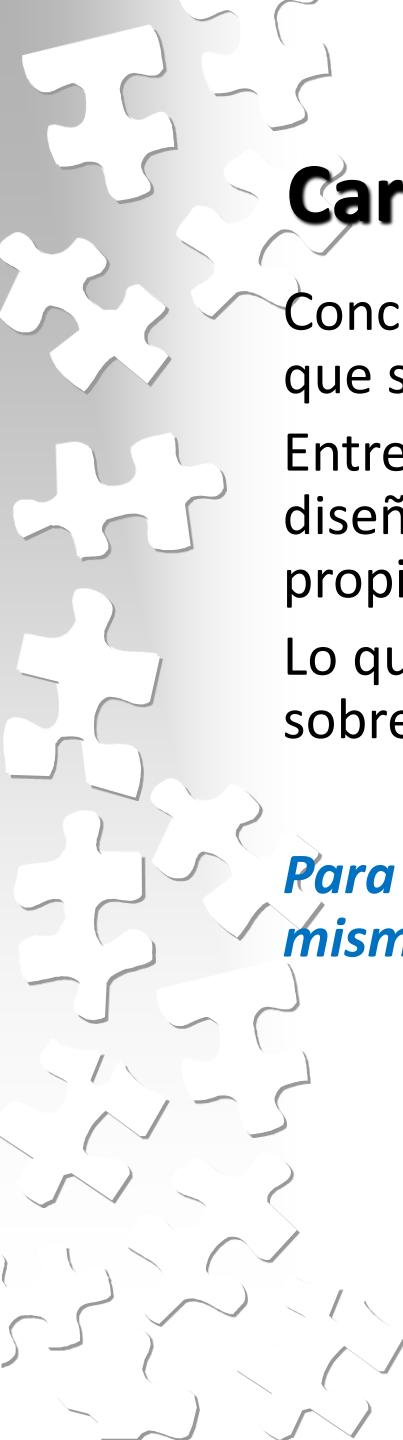
- *POLÍTICA DE LOCACIÓN* determina a que sitio debe enviarse el proceso seleccionado.
- *POLÍTICA DE INTERCAMBIO DE INFORMACIÓN* como intercambiar información entre los sitios.
- *POLÍTICA DE ASIGNACIÓN* determina la prioridad de ejecución de procesos locales y remotos en un sitio particular.
- *POLÍTICA QUE LIMITE LA MIGRACIÓN* determina el número de veces que un proceso puede migrar de un sitio a otro.

Son **procesos locales** a aquellos que se procesan en su sitio original y **procesos remotos** aquellos que se procesan en sitios distintos de los originales.



Agenda

1. Manejo de Recursos
2. Metodologías
 1. Asignación de tareas
 2. Balance de Carga
 3. Carga Compartida
3. Migración de Procesos
 1. Transferencia
4. Cluster, grid, cloud computing



Carga Compartida

Concepto diferente, no se hace un balance dinámico de carga sino que se reparte dinámicamente la carga.

Entre las consideraciones que deben tenerse en cuenta para el diseño de algoritmos de carga compartida se requieren decisiones propias acerca de las políticas presentadas en el balance de carga.

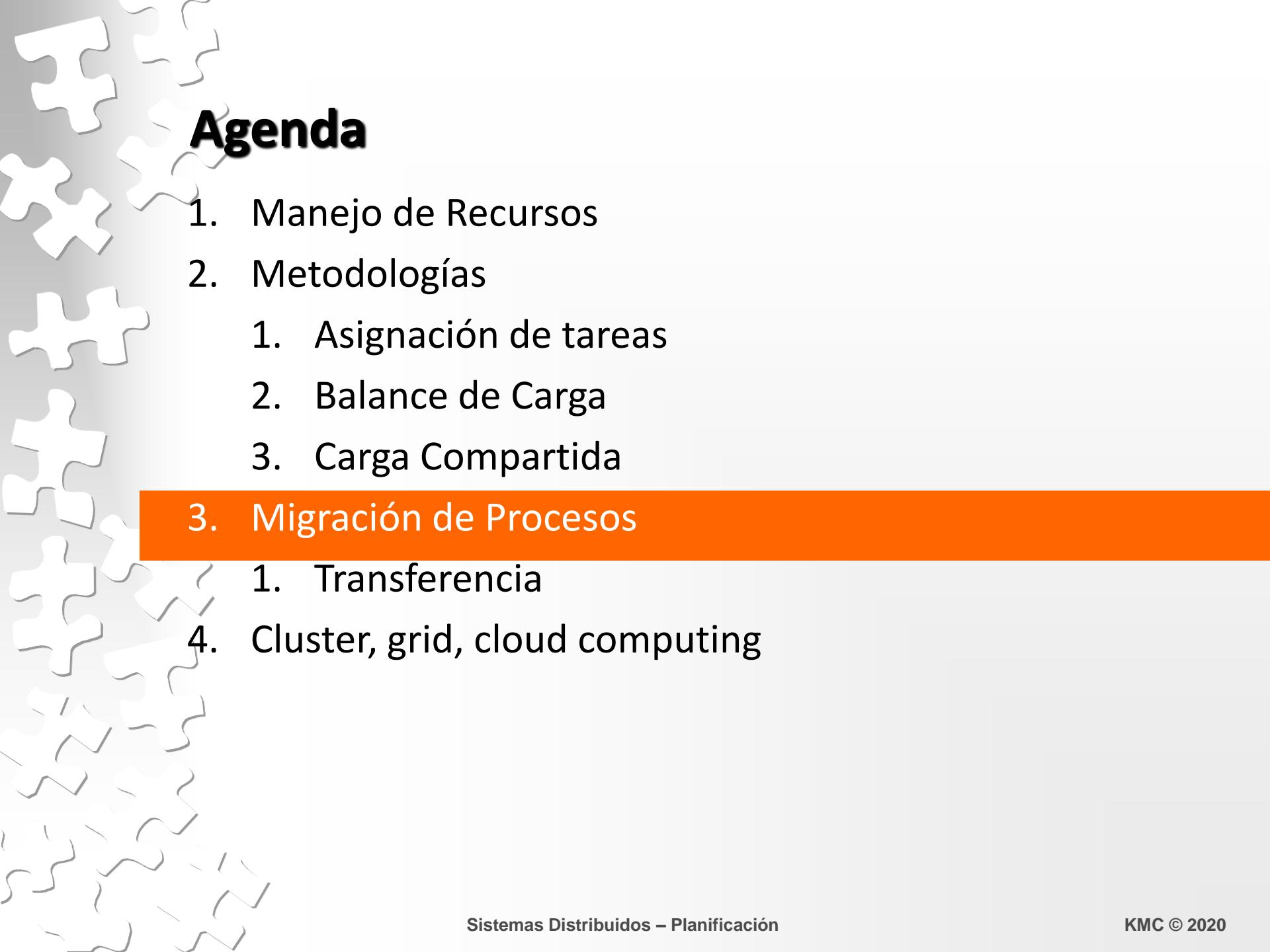
Lo que se busca es que ningún sitio esté ocioso cuando otros están sobrecargados.

Para el diseño de Carga Compartida se tienen en cuenta las misma políticas que para Balance de Carga.



Manejo de Recursos

- *POLÍTICA DE ESTIMACIÓN DE CARGA:* ídem
- *POLÍTICA DE TRANSFERENCIA DE PROCESOS:* política alto-bajo.
- *POLÍTICA DE LOCACIÓN:*
 - Política de remitente-inicia (*sender-initiated*): decide donde enviar.
 - Política de receptor-inicia (*receiver-initiated*): decide de donde tomar el proceso.
- *POLÍTICA DE INTERCAMBIO DE INFORMACIÓN ENTRE ESTADOS DE CARGA:*
 - Broadcast cuando el estado de carga cambia.
 - Polling cuando el estado de carga cambia.



Agenda

1. Manejo de Recursos
2. Metodologías
 1. Asignación de tareas
 2. Balance de Carga
 3. Carga Compartida
3. Migración de Procesos
 1. Transferencia
4. Cluster, grid, cloud computing



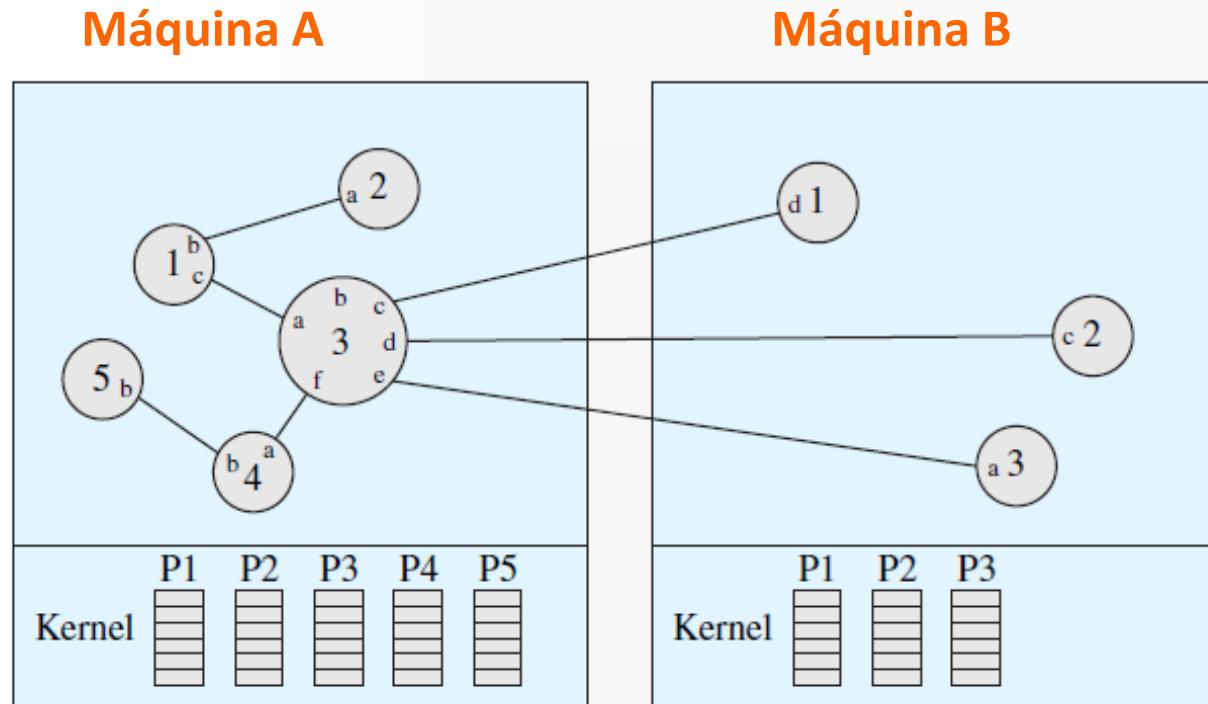
Administración de Procesos

Tres conceptos se usan para lograr este objetivo:

- **Alocación de procesos**: que proceso debe asignarse a que procesador.
- **Migración de procesos**: movimiento del proceso al procesador que ha sido asignado.
- **Threads**: paralelismo más fino para mejor utilización de la capacidad de procesamiento del sistema.

Administración de Procesos – Migración

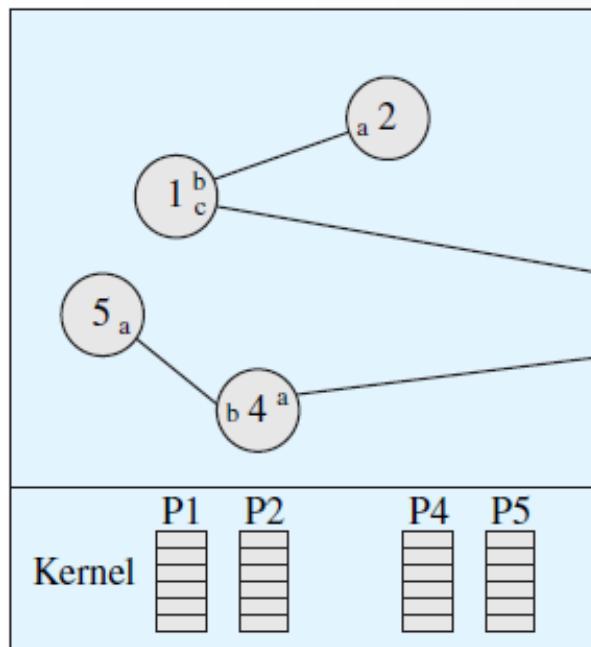
Ejemplo: Antes de la Migración



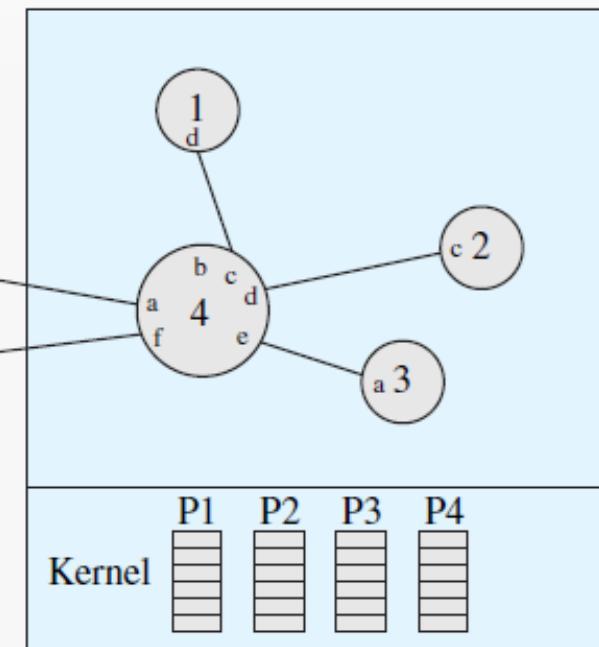
Administración de Procesos – Migración

Ejemplo: Después de la Migración

Máquina A

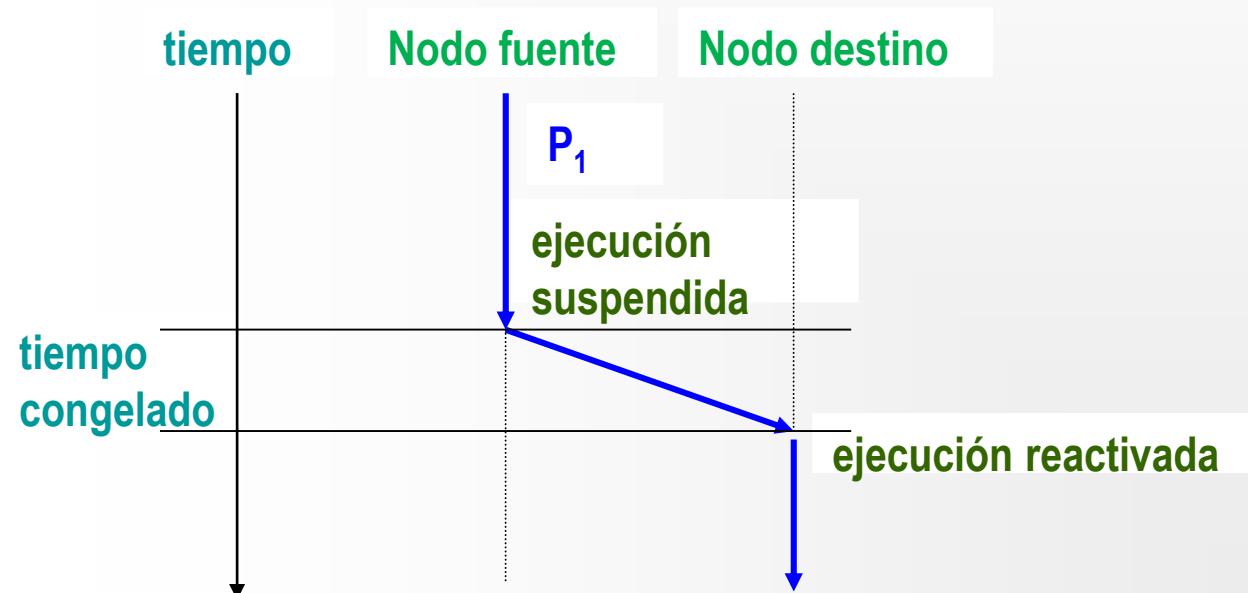


Máquina B



Administración de Procesos - Migración

Es la relocación de un proceso de su ubicación actual (*sitio fuente*) a otro sitio (*sitio destino*).





Administración de Procesos – Migración

Un proceso puede migrar antes de comenzar a ejecutar (*no apropiativo*) o durante el curso de su ejecución (*apropiativo*).

Involucra:

- Selección de un proceso a ser migrado.
- Selección de un sitio destino a donde el proceso debe ser migrado.
- Transferencia del proceso seleccionado al sitio destino.



Administración de Procesos – Migración

Características deseables de un buen mecanismo

- Transparencia.
- Mínima interferencia.
- Dependencias residuales mínimas.
- Eficiencia.
- Robustez.
- Comunicación entre coprocesos de un Job.

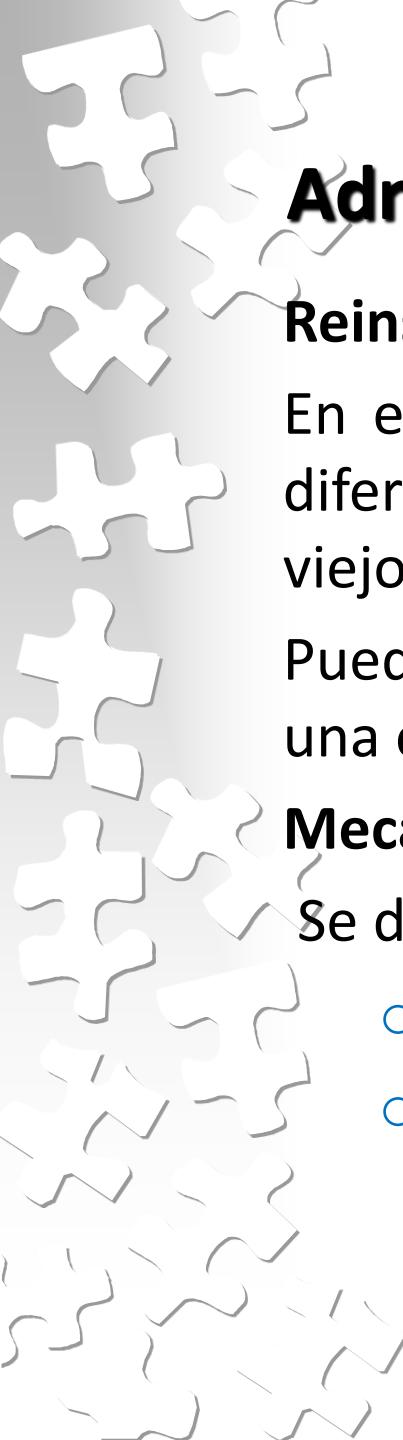


Administración de Procesos – Migración

Mecanismos

Involucra varias subactividades:

- Congelar el proceso en su sitio origen y reiniciarlo en su sitio destino.
- Transferir el espacio de direcciones correspondiente.
- Continuar los mensajes esperados por el proceso migrante.
- Manejar las comunicaciones entre procesos cooperativos que han sido separados como resultado de la migración.



Administración de Procesos – Migración

Reinstanciación del proceso en su sitio destino.

En el sitio destino se crea un PCB vacío, con identificación diferente para que existan dos copias, luego se cambia al viejo.

Puede ser necesario reiniciar los *system-calls* que contenía una operación de E/S lenta.

Mecanismos de transferencia del espacio de direcciones.

Se debe transferir:

- El estado del proceso
- El espacio de direcciones del proceso



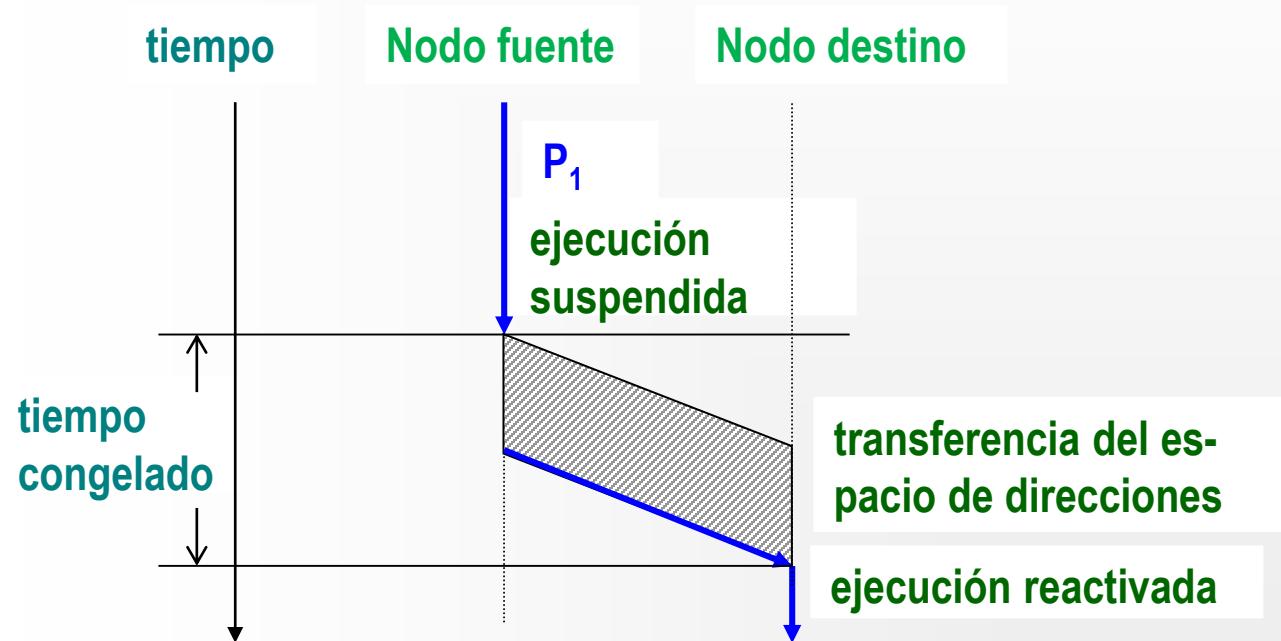
Administración de Procesos – Migración

Se usan diferentes mecanismos para la transferencia del espacio de direcciones:

- congelamiento total
- pretransferencia
- transferencia por referencia

Administración de Procesos – Migración

Congelamiento Total

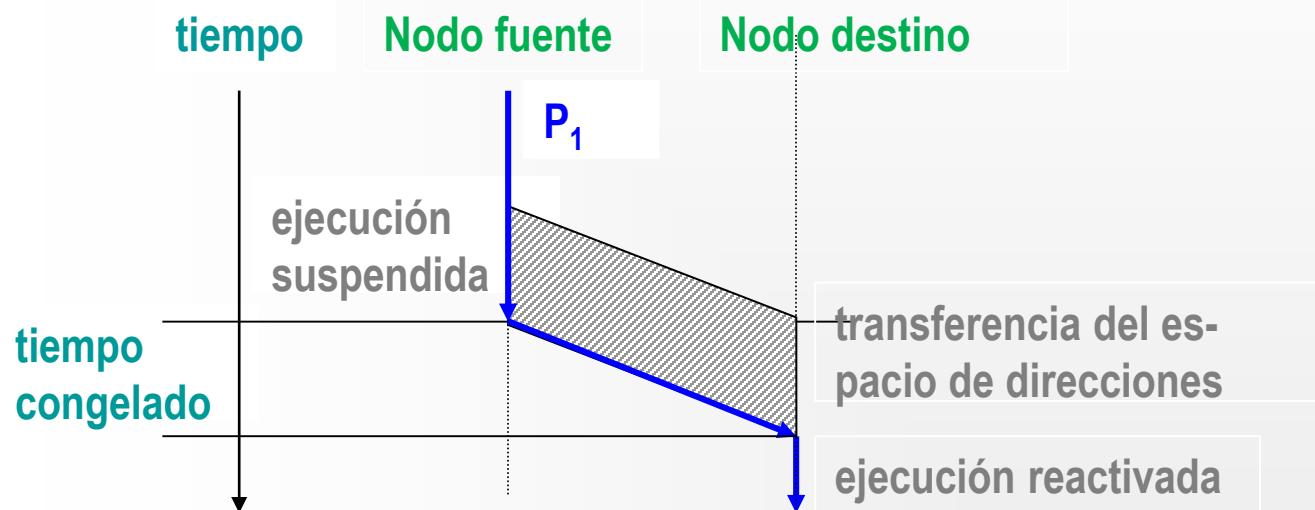


Ventaja: Fácil de implementar

Desventaja: Se pueden vencer los timeouts y los usuarios pueden notarlos.

Administración de Procesos – Migración

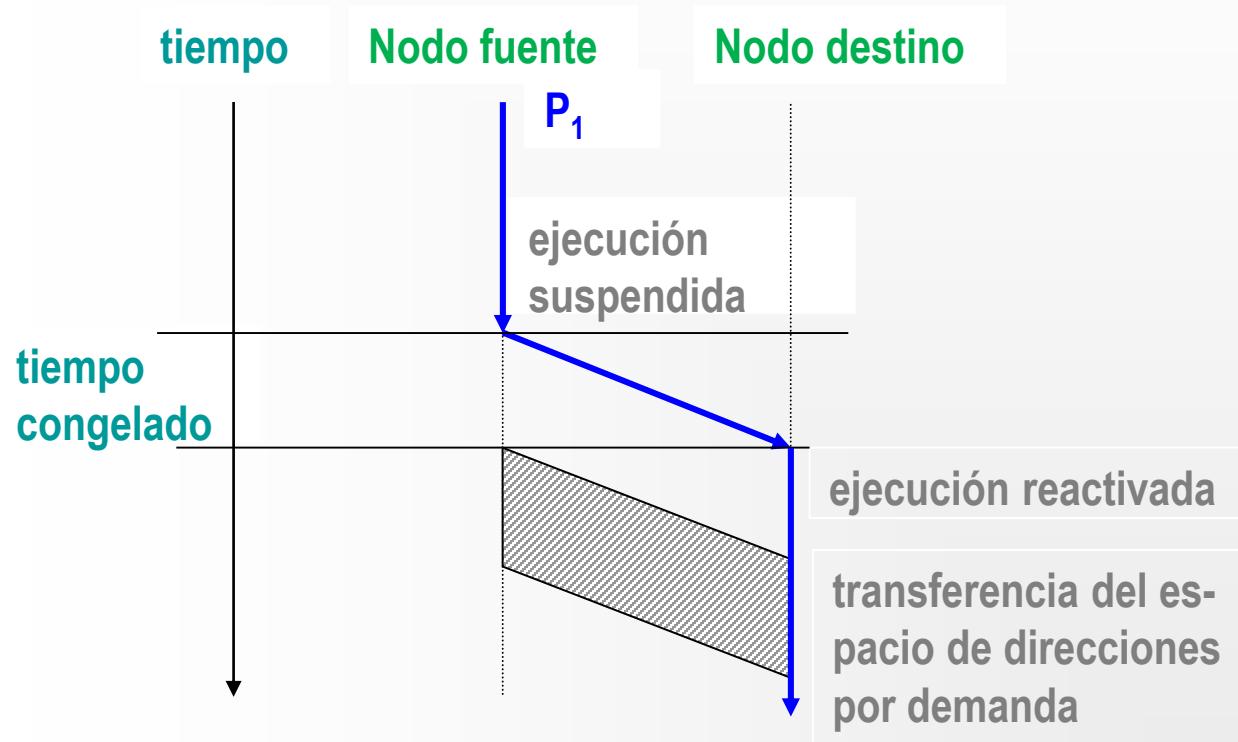
Pretransferencia



Puede transferir **páginas redundantes**

Administración de Procesos – Migración

Transferencia por demanda



Copia las páginas que necesita cuando ello ocurra.

Deja información en el sitio fuente, **esto es crítico**.

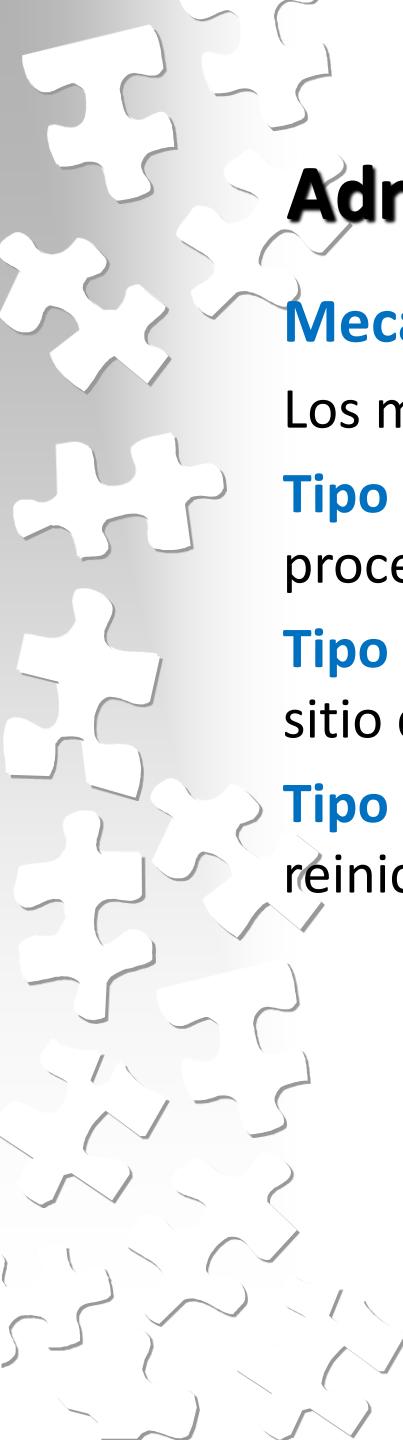


Administración de Procesos – Migración

Mecanismos para envío de mensajes

Cuando se mueven los procesos debe asegurarse que lleguen a su nueva locación:

- mensajes en ruta
- mensajes pendientes
- futuros mensajes



Administración de Procesos – Migración

Mecanismos para envío de mensajes

Los mensajes al proceso migrante pueden ser clasificados como:

Tipo 1: recibidos en el sitio fuente luego que se ha congelado el proceso y no se ha reiniciado en el sitio destino.

Tipo 2: recibidos en el sitio fuente cuando el proceso se inició en el sitio destino.

Tipo 3: enviados al proceso desde otros sitios luego que éste reinició en el sitio destino.



Administración de Procesos - Migración

Mecanismos para manejar coprocesos

Deshabilitar la separación de coprocesos

- a) No permitir migración de procesos que esperan que completen uno o más procesos (hijos).
- b) Asegurar que todos migren juntos.

Concepto del sitio origen

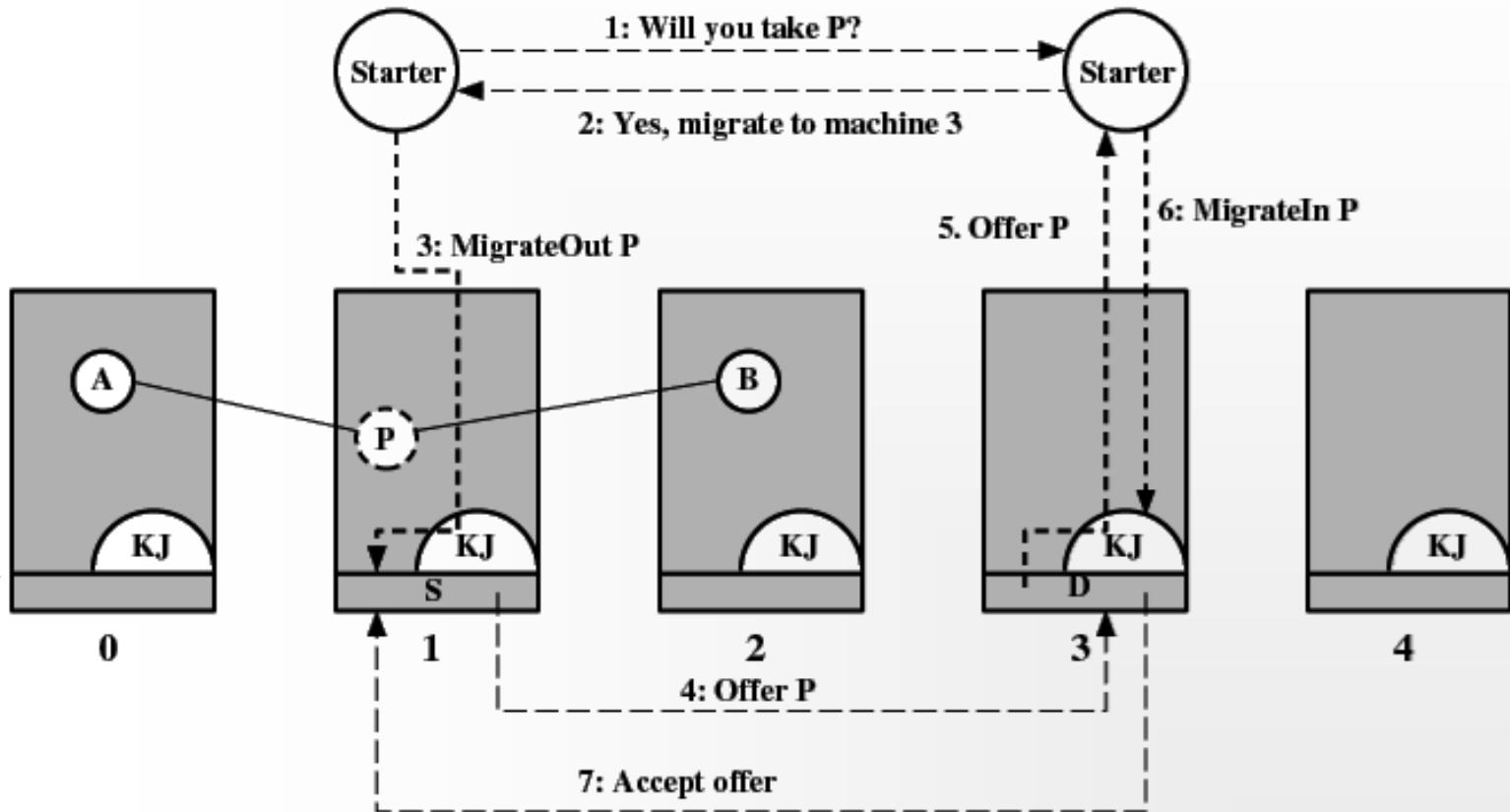
La comunicación se lleva a cabo por sitio origen.

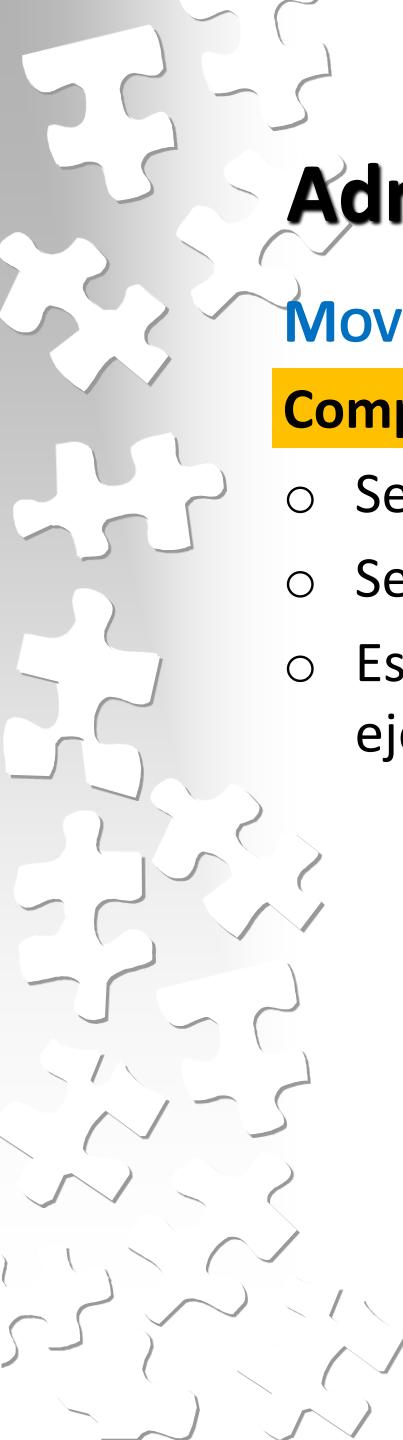


Ejemplo de Negociación en Migración

- La política de migración es responsabilidad del Starter (es un utilitario).
- El Starter es también responsable de la planificación de largo término y la alocación de memoria.
- La decisión de migrar debe ser alcanzada conjuntamente entre dos procesos Starter (uno en la fuente y otro en el destino)

Ejemplo de Negociación en Migración



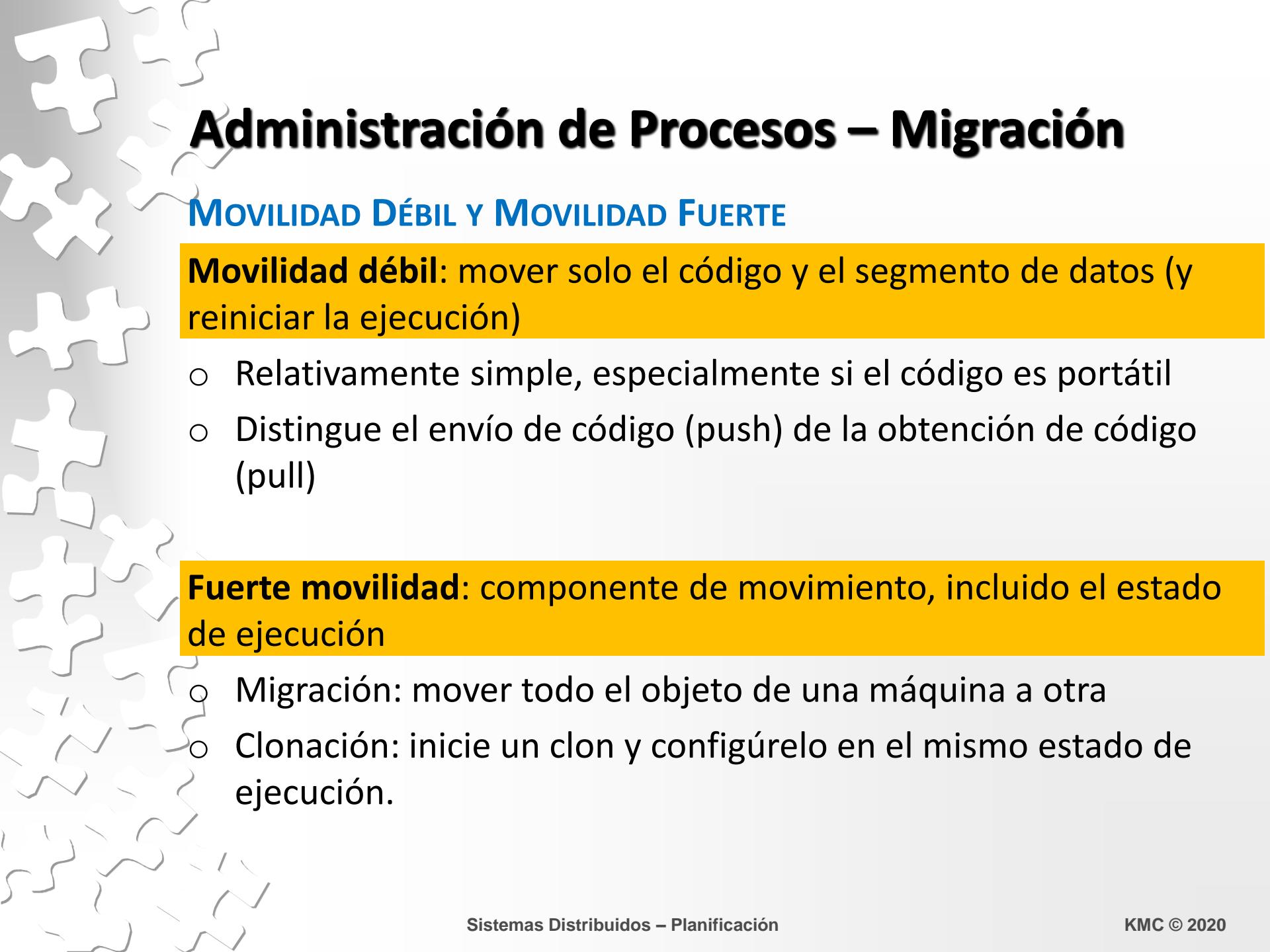


Administración de Procesos – Migración

MOVILIDAD DÉBIL Y MOVILIDAD FUERTE

Componentes del objeto

- Segmento de código: contiene el código real
- Segmento de datos: contiene el estado
- Estado de ejecución: contiene el contexto del subprocesso que ejecuta el código del objeto



Administración de Procesos – Migración

MOVILIDAD DÉBIL Y MOVILIDAD FUERTE

Movilidad débil: mover solo el código y el segmento de datos (y reiniciar la ejecución)

- Relativamente simple, especialmente si el código es portátil
- Distingue el envío de código (push) de la obtención de código (pull)

Fuerte movilidad: componente de movimiento, incluido el estado de ejecución

- Migración: mover todo el objeto de una máquina a otra
- Clonación: inicie un clon y configúrelo en el mismo estado de ejecución.



Administración de Procesos – Migración

Ventajas

- Reducción del tiempo medio de respuesta.
- Aceleración de jobs individuales.
- Ganancia de procesamiento total.
- Efectiva utilización de recursos.
- Reducción de tráfico en la red.
- Mejora de la confiabilidad del sistema.
- Mejora de la seguridad del sistema.



Agenda

1. Manejo de Recursos
2. Metodologías
 1. Asignación de tareas
 2. Balance de Carga
 3. Carga Compartida
3. Migración de Procesos
 1. Transferencia
4. Cluster, grid, cloud computing

Clusters

- Es una alternativa al multiprocesamiento simétrico (SMP)
- Es un grupo de computadoras interconectadas trabajando juntas como un recurso unificado.
 - La visión es como si fuera una única máquina.
 - Cada una puede correr su propio sistema.

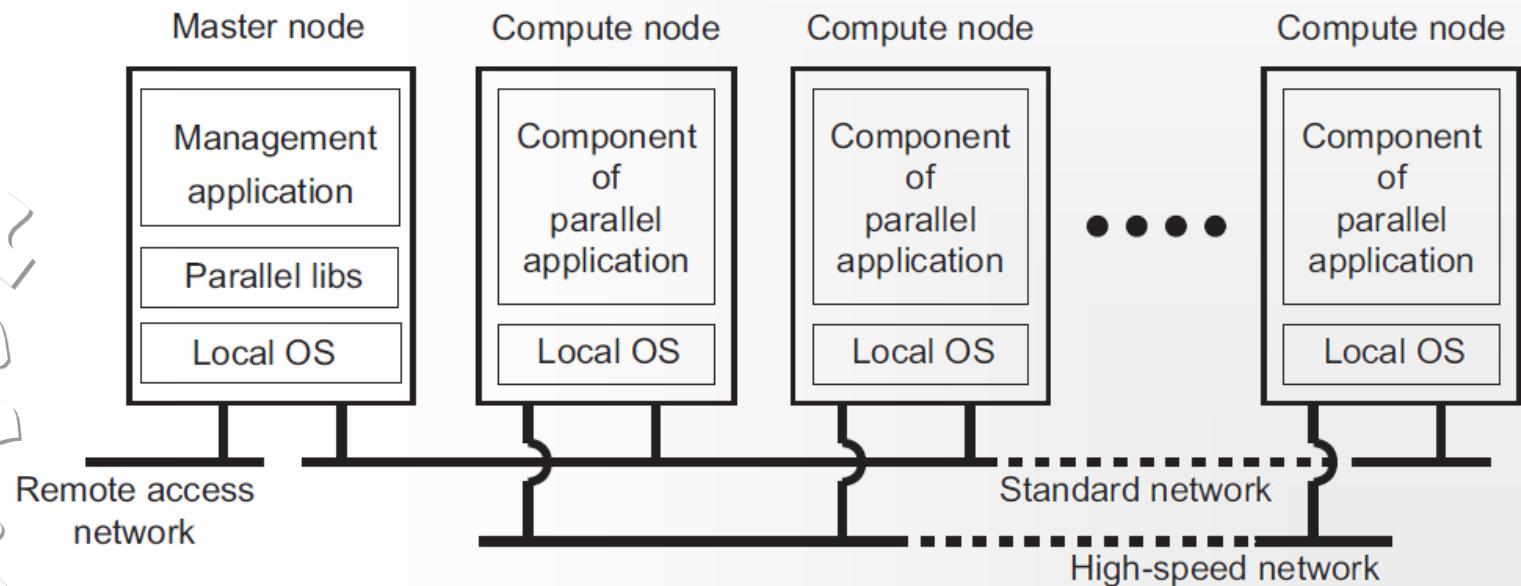
Homogeneidad

Heterogenidad

Arquitectura de Cluster de Computadoras

Esencialmente un grupo de sistemas de alta gama conectados a través de una LAN

Homogéneo: mismo sistema operativo, hardware casi idéntico
Nodo de gestión único

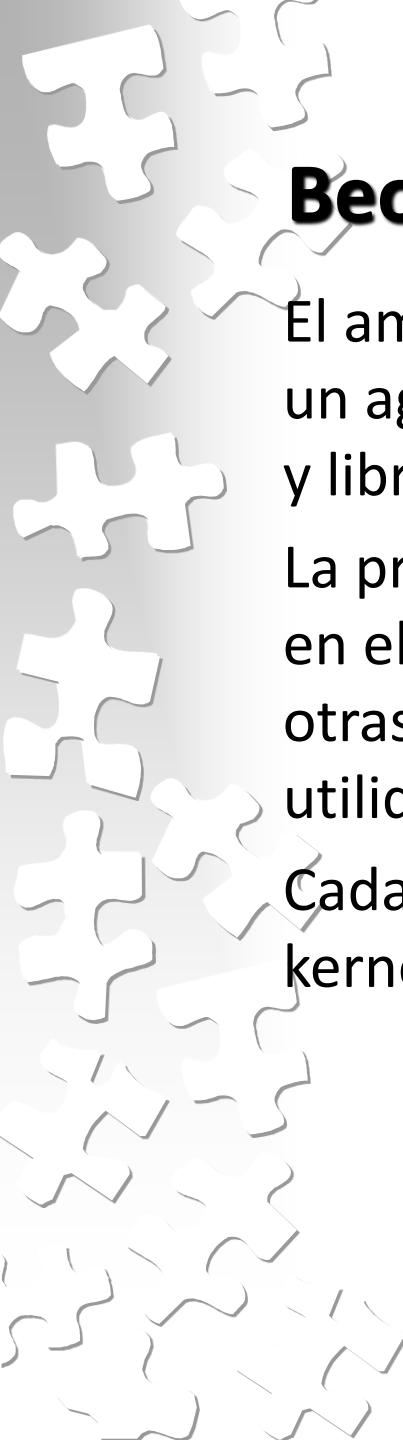


Arquitectura de Cluster de Computadoras

Servicios y funciones de middleware del cluster

- Único punto de entrada
- Jerarquía de archivos única
- Único punto de control
- Red virtual única
- Espacio de memoria unificado
- Administración de tareas de sistema unificado
- Interfaz de usuario única
- Espacio de E/S único
- Espacio de proceso único
- Checkpointing
- Migración de procesos





Beowulf y Clusters Linux

El ambiente de software Beowulf está implementado como un agregado a las distribuciones disponibles comercialmente y libres de “royalties” de Linux.

La principal fuente *open-source* de Beowulf está disponible en el sitio de Beowulf (www.beowulf.org) pero hay muchas otras organizaciones que ofrecen libremente herramientas y utilidades.

Cada nodo en el cluster Beowulf corre su propia copia del kernel de Linux y pueden funcionar en forma autónoma.



Grid

- Conjunto de nodos de diferentes lugares
 - Heterogeneidad
 - Disperso entre varias organizaciones
 - Puede abarcar fácilmente una red de área amplia (WAN)

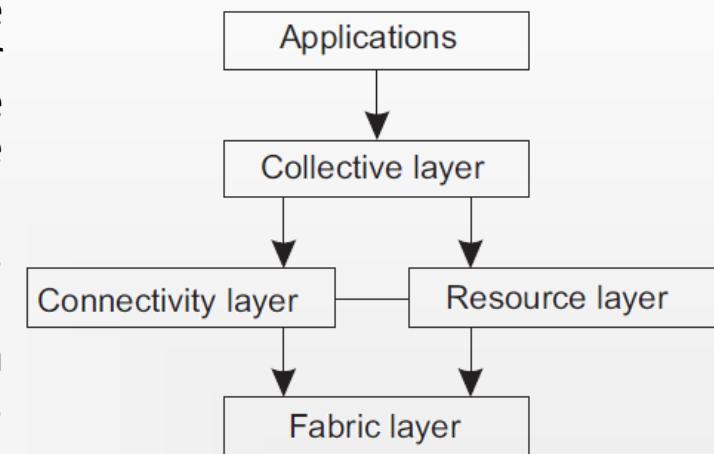
Para permitir colaboraciones, los grids generalmente usan organizaciones virtuales.

En esencia, esta es una agrupación de usuarios (o mejor: sus ID) que permitirá autorización para la asignación de recursos.

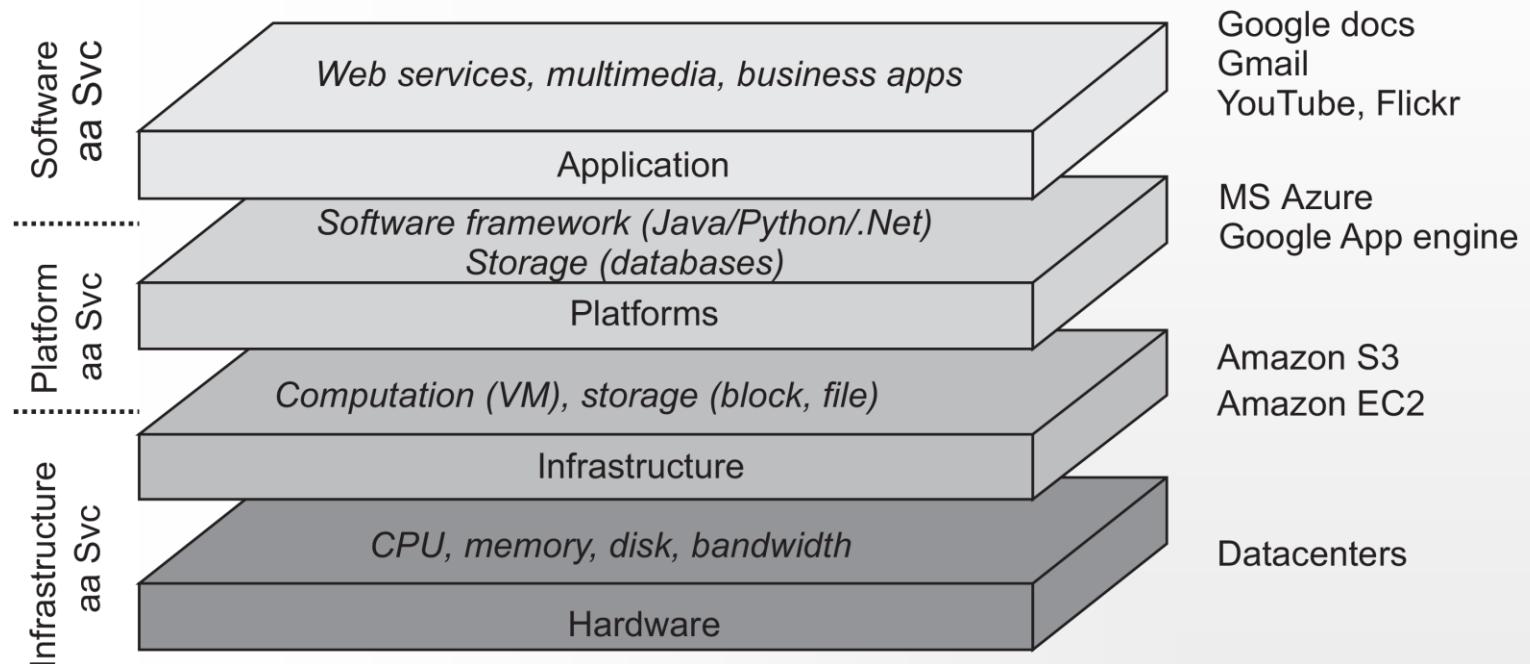
Grid - Arquitectura

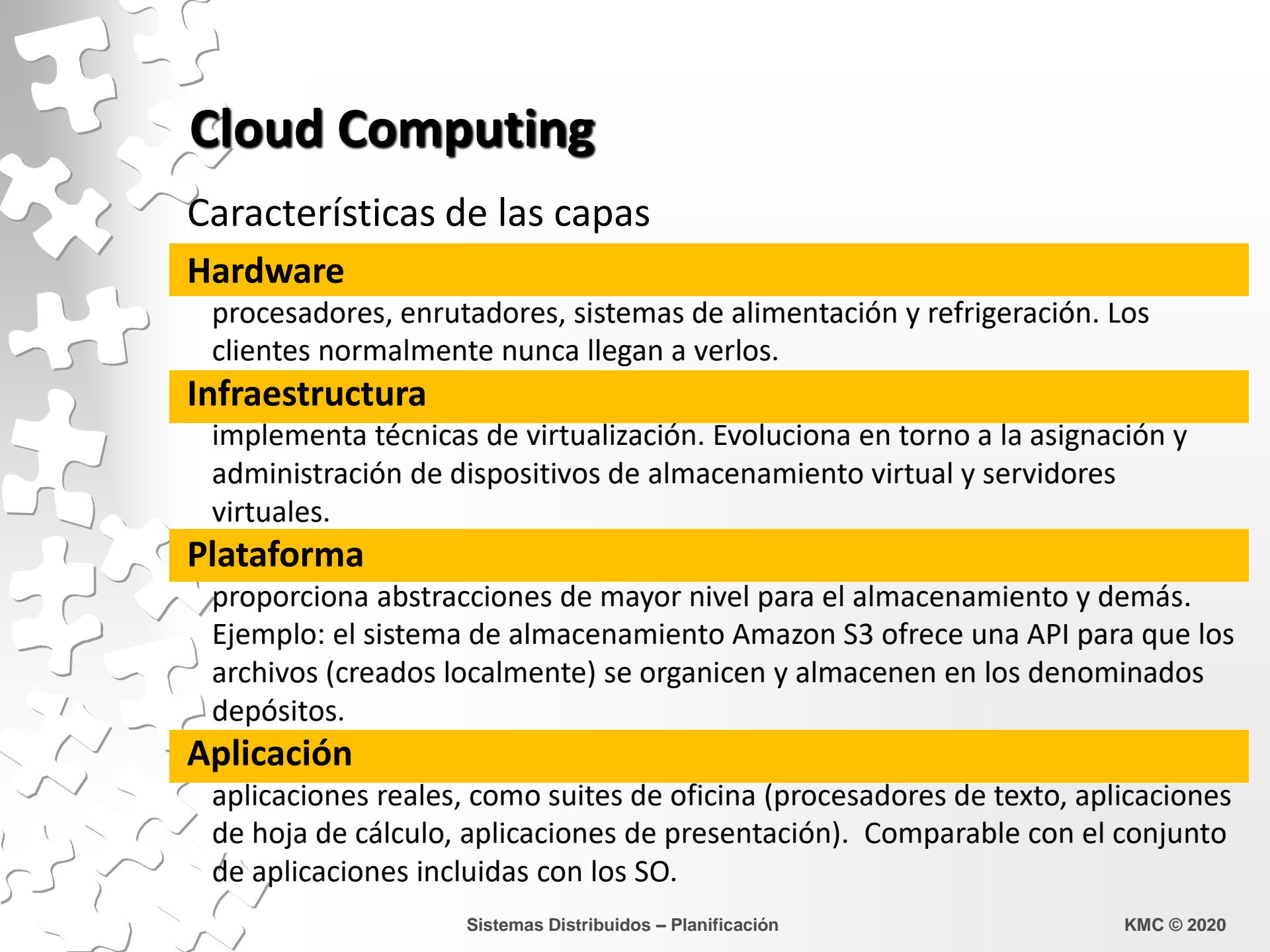
Capas (Layers)

- Fabric: proporciona interfaces a recursos locales (para consultar estado y capacidades, bloqueo, etc.)
- Connectivity: protocolos de comunicación/transacción, por ejemplo, para mover datos entre recursos. También varios protocolos de autenticación.
- Resource: administra un solo recurso, como crear procesos o leer datos.
- Collective: maneja el acceso a múltiples recursos: descubrimiento, programación, replicación.
- Aplicación: contiene aplicaciones de grid reales en una sola organización.



Cloud Computing





Cloud Computing

Características de las capas

Hardware

procesadores, enrutadores, sistemas de alimentación y refrigeración. Los clientes normalmente nunca llegan a verlos.

Infraestructura

implementa técnicas de virtualización. Evoluciona en torno a la asignación y administración de dispositivos de almacenamiento virtual y servidores virtuales.

Plataforma

proporciona abstracciones de mayor nivel para el almacenamiento y demás. Ejemplo: el sistema de almacenamiento Amazon S3 ofrece una API para que los archivos (creados localmente) se organicen y almacenen en los denominados depósitos.

Aplicación

aplicaciones reales, como suites de oficina (procesadores de texto, aplicaciones de hoja de cálculo, aplicaciones de presentación). Comparable con el conjunto de aplicaciones incluidas con los SO.



Bibliografía

- Sinha, P. K.; “Distributed Operating Systems: Concepts and Design”, IEEE Press, 1997.
- Van Steen, M; Tanenbaum, A; “Distributed Systems”, 3rd. Edition, 2017.
- Coulouris,G.F.; Dollimore, J. y T. Kindberg; “Distributed Systems: Concepts and Design”. 5th Edition Addison Wesley, 2011.
- Stallings, W. Operating Systems: Internals and Design Principles, 8th Edition, 2014 Prentice Hall.