

Infraestructura computacional

Infraestructura de TI: diseño - procesamiento

Diseño: Clusters

- Propósito:
 - Alta disponibilidad: recuperarse rápidamente de fallas menores; puede haber breve discontinuidad en el servicio
 - Tolerancia a fallas: continuar con el servicio a pesar del surgimiento de fallas; eventualmente con algún grado de degradación
 - Escalabilidad: responder a condiciones variables de carga
 - Desempeño: incrementar la capacidad de cómputo

Diseño: Clusters

- Tipos:
 - Failover: el servicio migra automáticamente ante una falla
 - Balanceo de carga: el mismo servicio es prestado por un conjunto de servidores
 - Cómputo: un conjunto de servidores se reparten los cálculos que se deben realizar
 - Aplicación: como los anteriores pero implementados en una aplicación específica (bases de datos, etc).

Diseño: Clusters - Alta disponibilidad

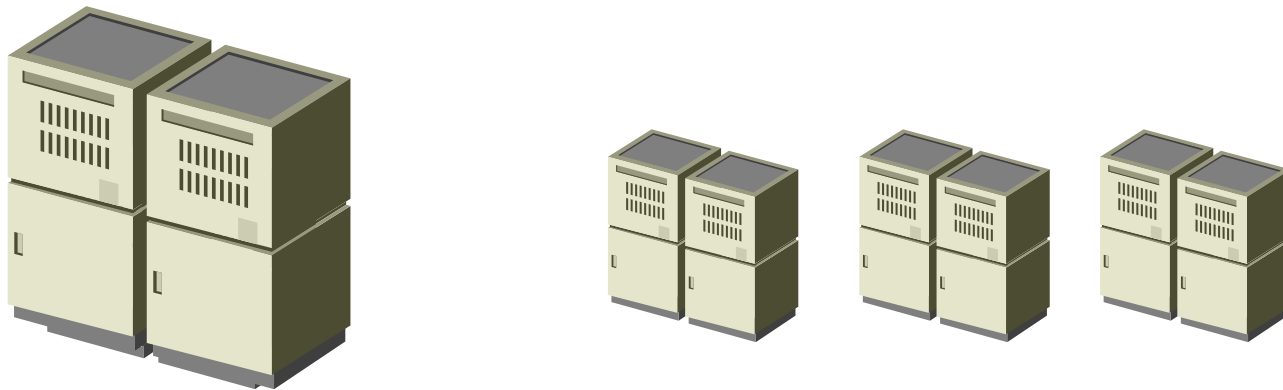
- Tener alta disponibilidad en procesamiento implica:
 - Redundancia de servidores
 - Conjunto de servidores presta un conjunto de servicios
 - Las relaciones servicio-servidor no son estáticas
 - Equilibrio en escalabilidad horizontal-vertical

Diseño: Clusters - Alta disponibilidad

- Manejo del estado:
 - Instancias independientes: sin estado compartido, crecimiento horizontal.
 - Instancias con alguna compartición: estado compartido por memoria
 - Estado compartido por disco: cluster de failover, crecimiento vertical.

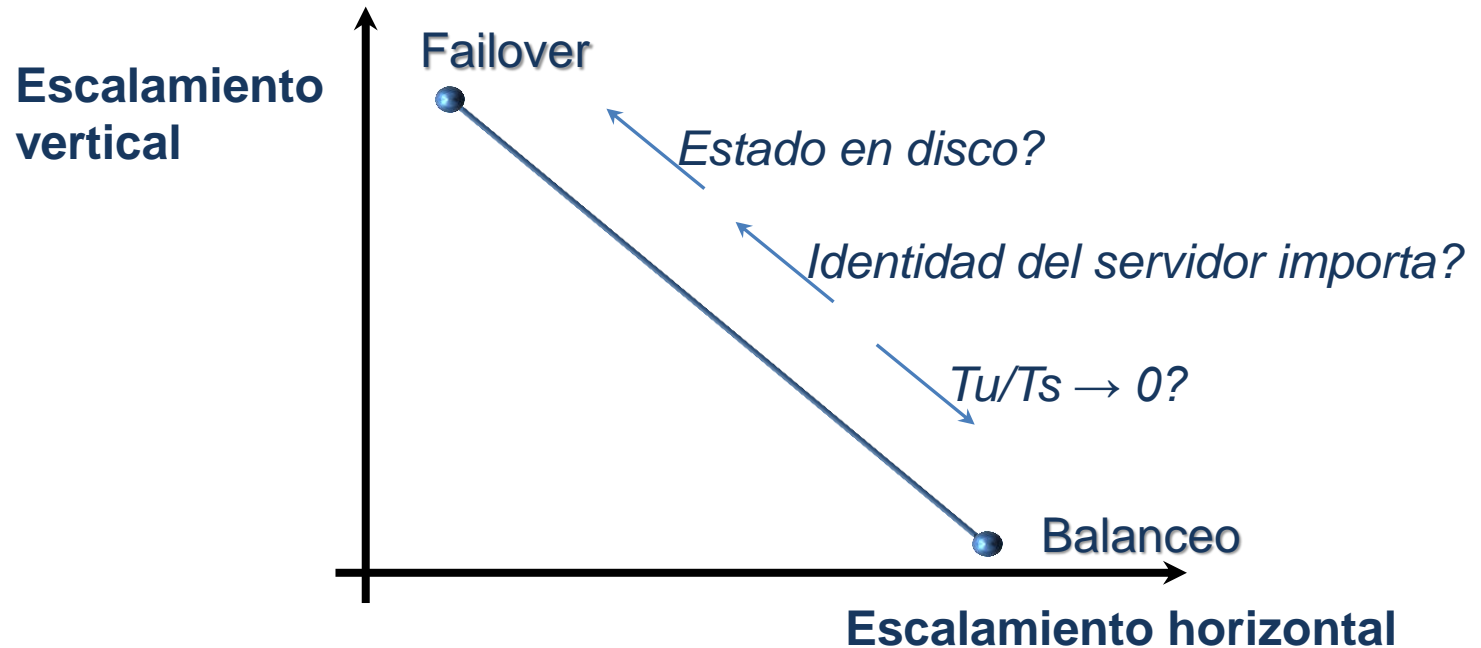
Diseño: Clusters - Alta disponibilidad

- Escalamiento:
 - Vertical: aumentar recursos en una máquina
 - Horizontal: aumentar número de máquinas
 - Diagonal



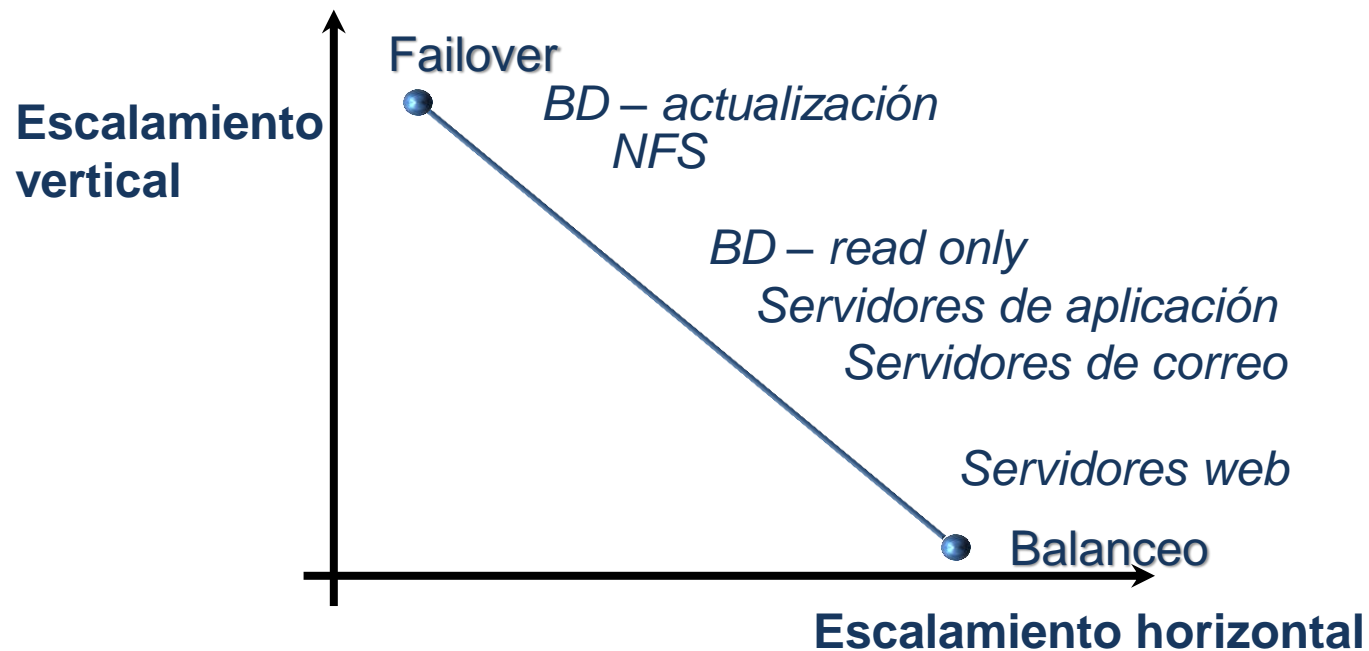
Diseño: Clusters - Alta disponibilidad

- Failover vs. balanceo



Diseño: Clusters - Alta disponibilidad

- Failover vs. balanceo



Diseño: Clusters - Alta disponibilidad

- **Failover:**
 - Recuperación ante fallas
 - Persistencia por almacenamiento compartido
 - Solución genérica
 - Escala verticalmente
 - Grandes servidores con hardware redundante
- **Balanceo de carga:**
 - Recuperación ante fallas
 - Sin estado
 - Solución específica
 - Escala horizontalmente
 - Servidores sencillos y baratos

Diseño: Clusters - failover

- Reacción ante fallas usual
 - Hardware:
 - reparar y reboot
 - SO
 - reboot
 - Aplicación
 - relanzar
- Reacción ante fallas cluster
 - migrar el servicio
 - migrar el servicio
 - Relanzar, eventualmente migrar el servicio

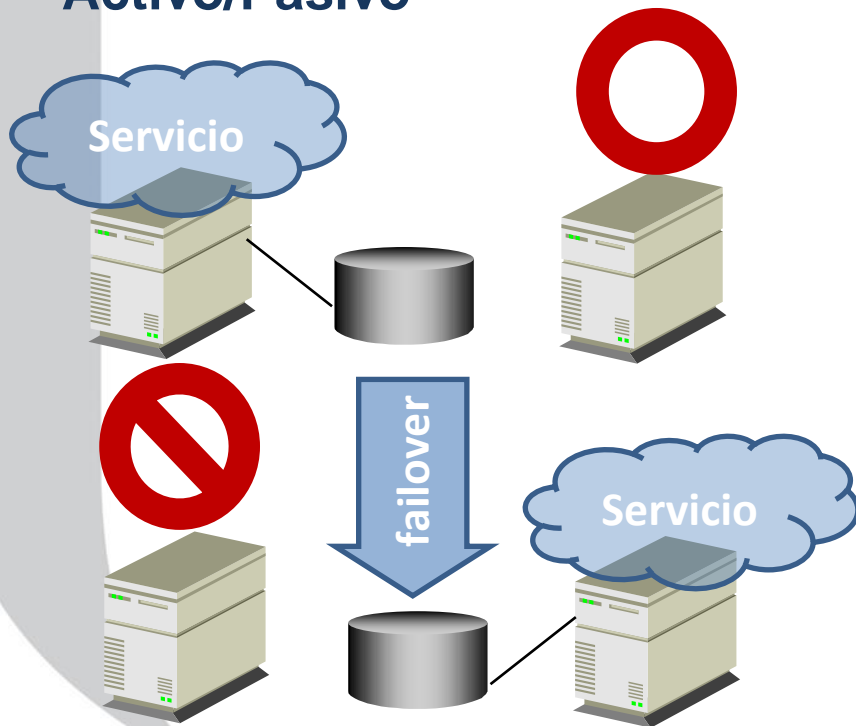
Diseño: Clusters - failover

- Respuestas ante falla de aplicación:
 - Ignorar, y descartar aplicación
 - Notificar a quien puede decidir
 - Intentar volver a arrancar un cierto número de veces, y si no es posible:
 - Ignorar y descartar aplicación
 - Enviar notificación
 - Empezar failover
 - Reboot del servidor
 - Detener servidor
 - Empezar failover sin intentar volver a arrancar

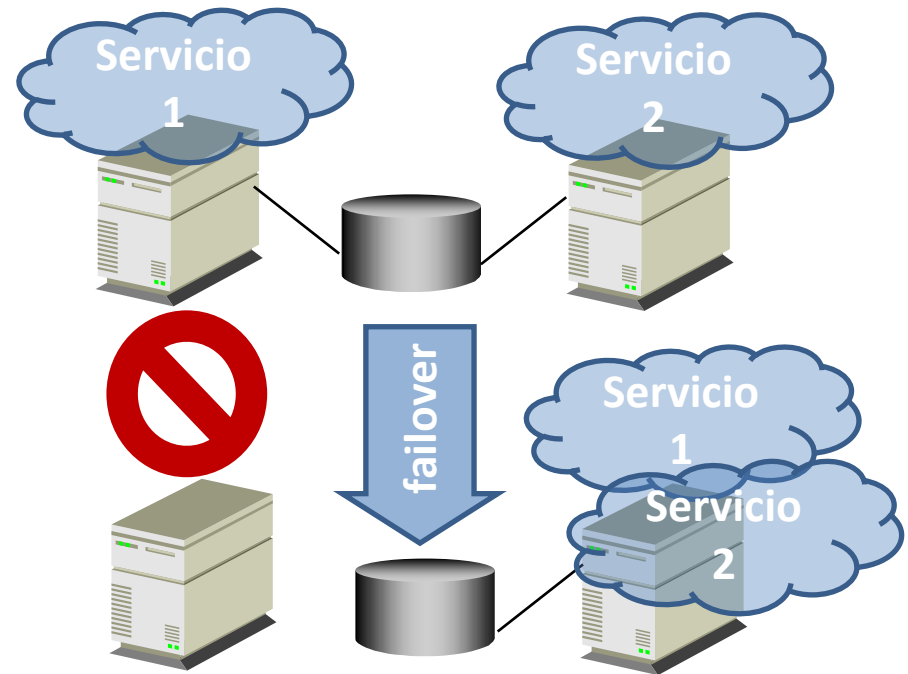
Diseño: Clusters - failover

- Activo - Pasivo:

Activo/Pasivo

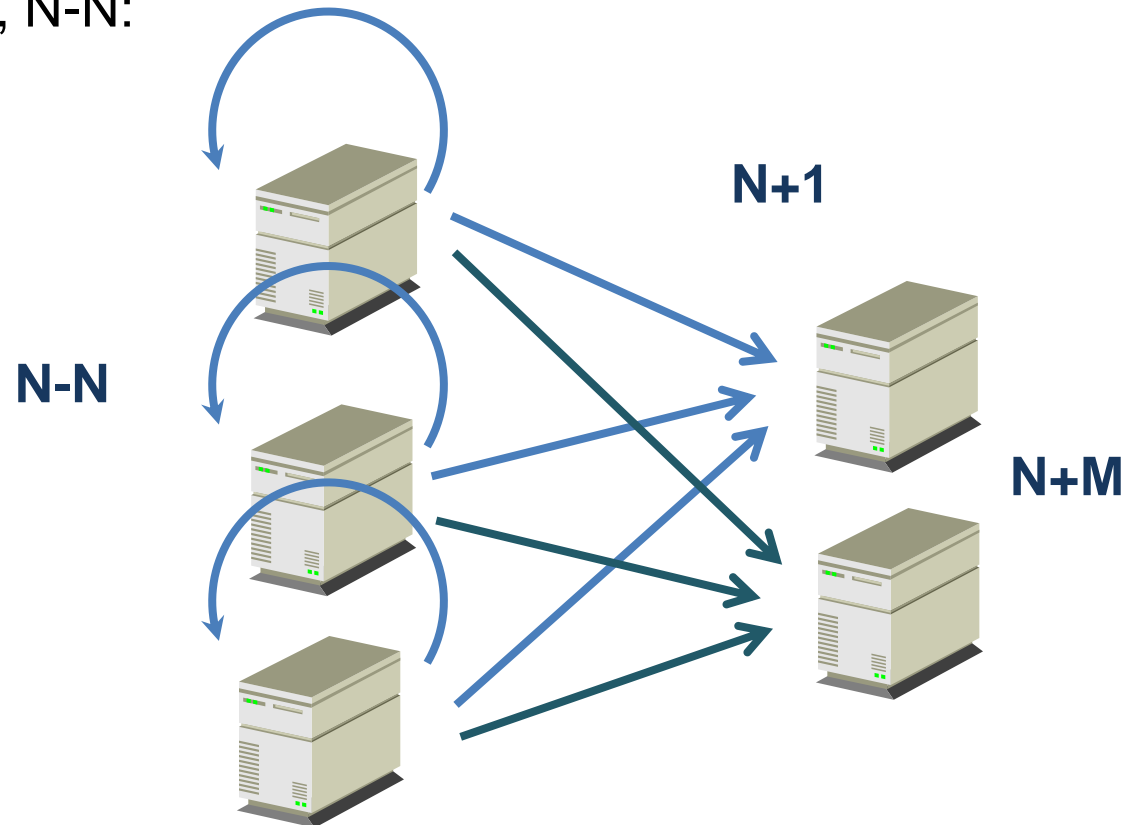


Activo/Activo



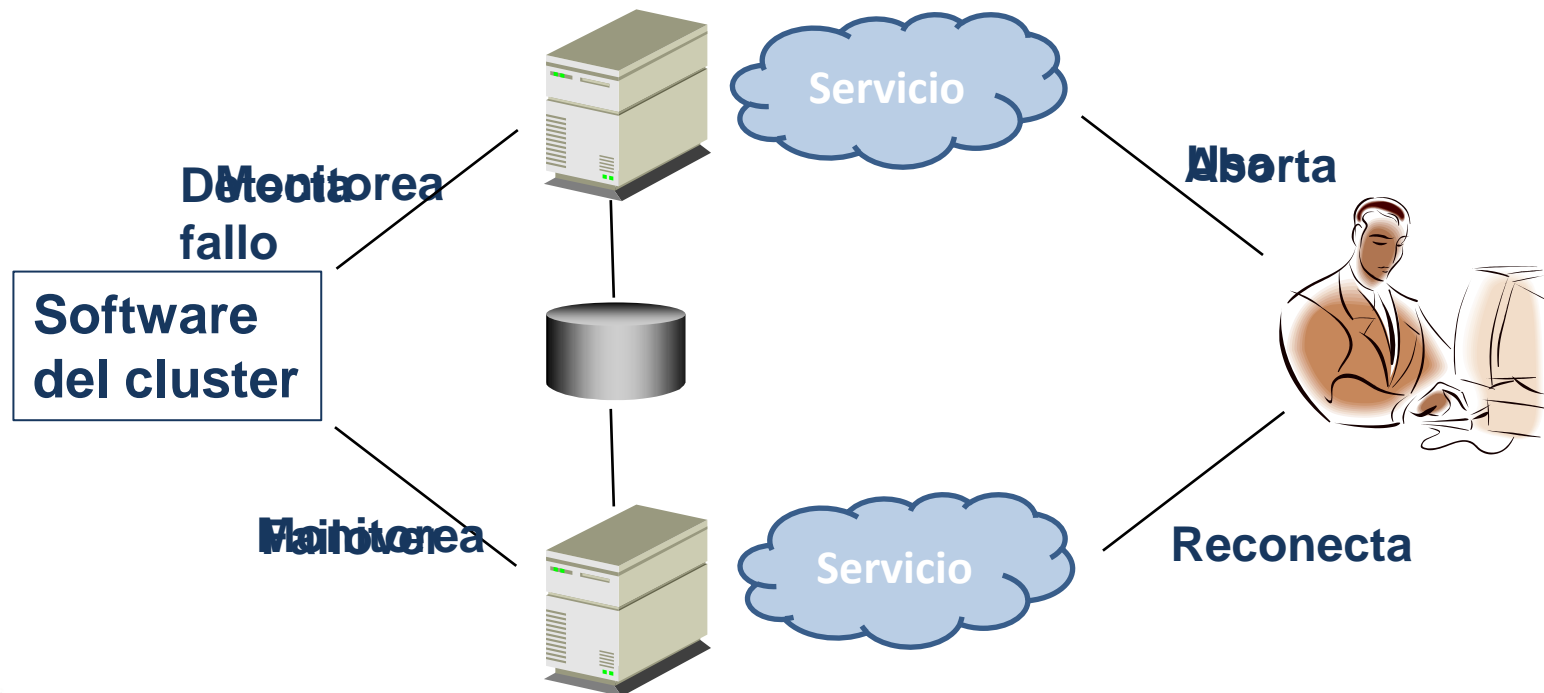
Diseño: Clusters - failover

- $N+1$, $N+M$, $N-N$:



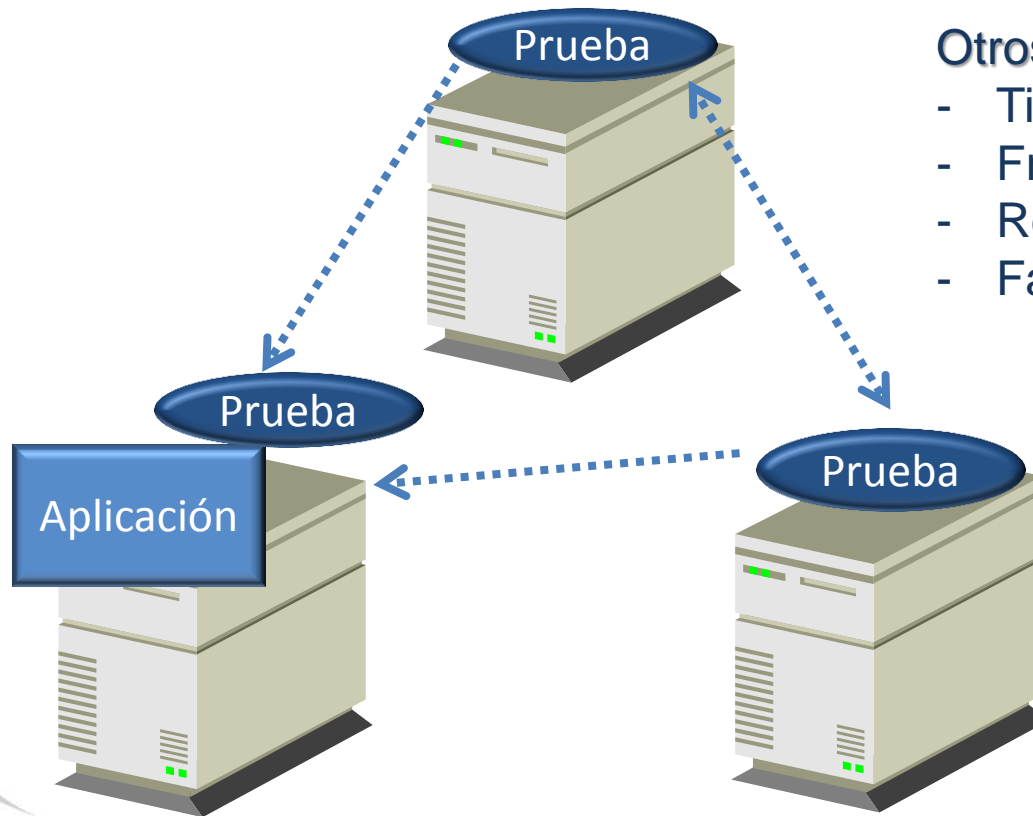
Diseño: Clusters - failover

- Funcionamiento:



Diseño: Clusters - failover

- Funcionamiento - Monitores



Otros aspectos:

- Timeout de las pruebas
- Frecuencia de las pruebas
- Renuencia de las aplicaciones
- Falla de la prueba

Diseño: Clusters - failover

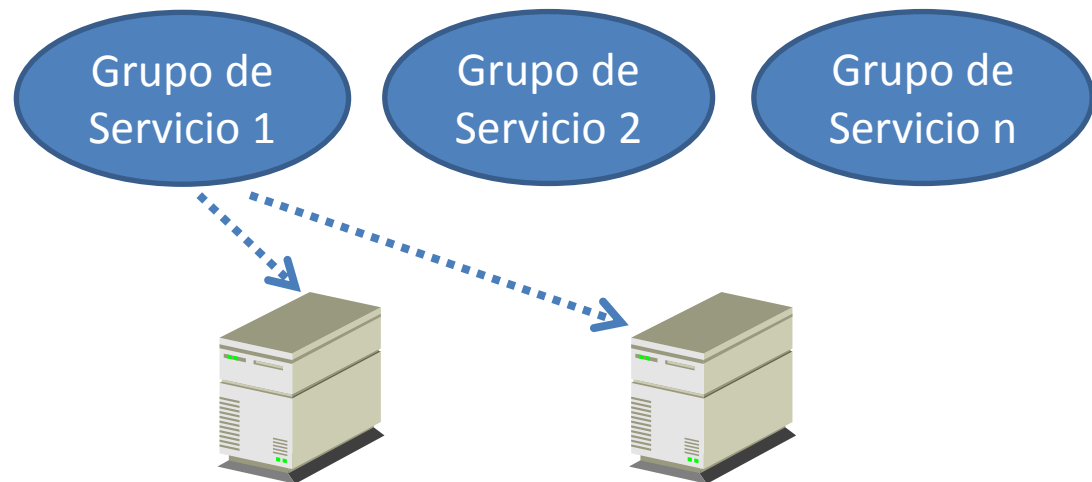
- Funcionamiento:
 - Nodo preferido: los servicios tienen un “nodo base”
 - Estado del cluster compartido: todo nodo conoce la localización de todos los servicios
 - Comunicación intracluster: servicio de comunicación dedicado
 - Latido: monitoreo sistemático del servidor
 - Síndrome de cerebro dividido: falla del sistema de clustering (del latido)
 - Dispositivos de quorum: mecanismo redundante de monitoreo
 - Conectividad externa: el cluster debe revisar las comunicaciones con el mundo externo
 - Dependencias entre los servicios: puede ser necesario levantar los servicios en un cierto orden

Diseño: Clusters - failover

- Administración de los servicios:
Diseño centrado en servicios

Servicios:

- Identidad de red
- Almacenamiento
- Procesos



Servidores

Diseño: Clusters - failover

- Administración de los servicios:
 - Declaración del servicio: procesos permanentes y esporádicos que se deben arrancar, volúmenes de almacenamiento, direcciones IP, etc.
 - Activación del servicio: es realizada por el software del cluster; el servicio no necesita proveer código para esto
 - Chequeo de los servicios: el software del cluster revisa la disponibilidad del servicio a intervalos definidos.
 - Desactivación del servicio: se detienen los procesos del servicio, y se le retiran los recursos (volúmenes, interfaces de red, etc.)
 - Relanzamiento del servicio: desactivación del servicio, y reactivación en el mismo nodo.
 - Migración del servicio: se detiene el servicio y se lanza en otro nodo

Diseño: Clusters - failover

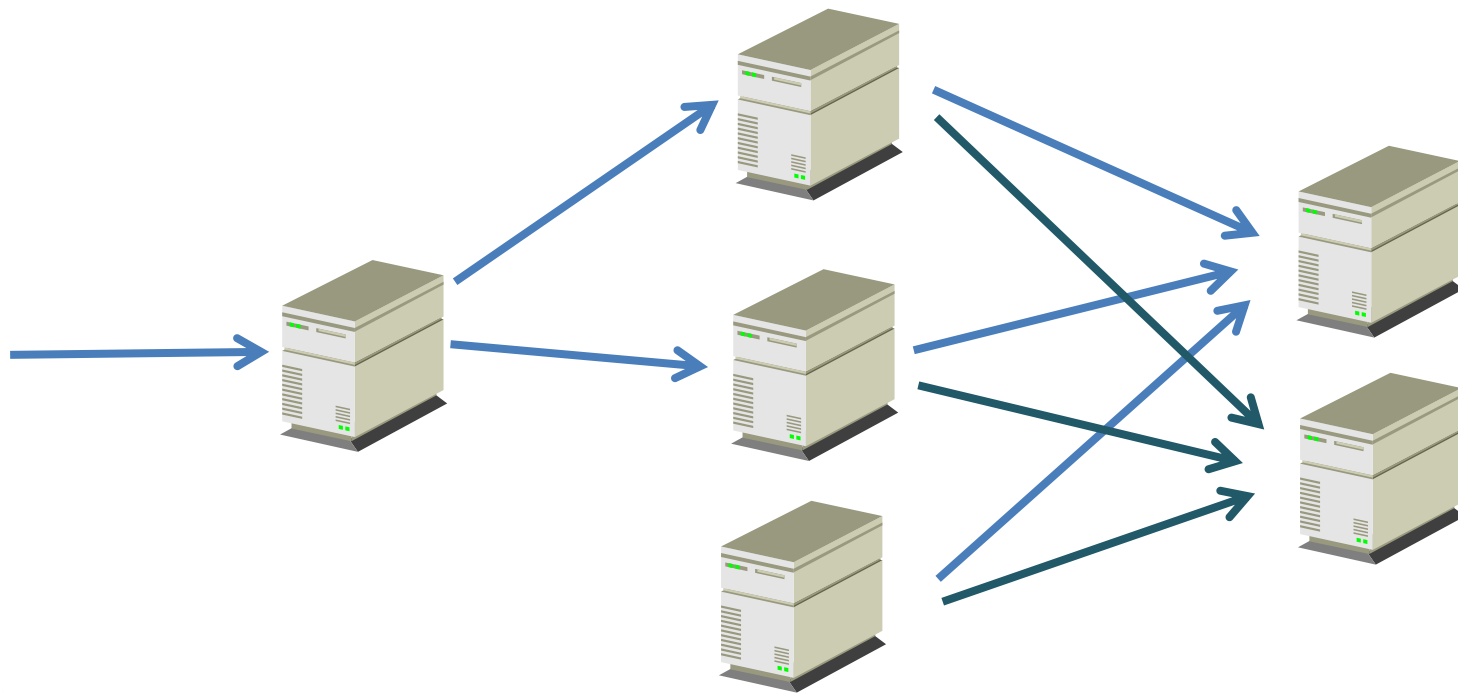
- Administración de los servicios:
Requerimientos sobre la aplicación:
 - Independencia del host físico: las aplicaciones no deben usar características específicas de la máquina física (hostid, IP)
 - Localización de archivos: los archivos deben estar en el almacenamiento compartido
 - Aprovisionamiento: la aplicación no debe depender de acciones durante el arranque del sistema operativo
 - Acciones de start-stop-restart disponibles para cada recurso

Diseño: Clusters - balanceo de carga

- Servicios sin estado: un requerimiento puede asignarse a cualquier nodo
- Dispositivo de balanceo de carga: para distribuir la carga entre los nodos
- Mejora de la productividad y del desempeño
- Se pueden combinar con failover clusters

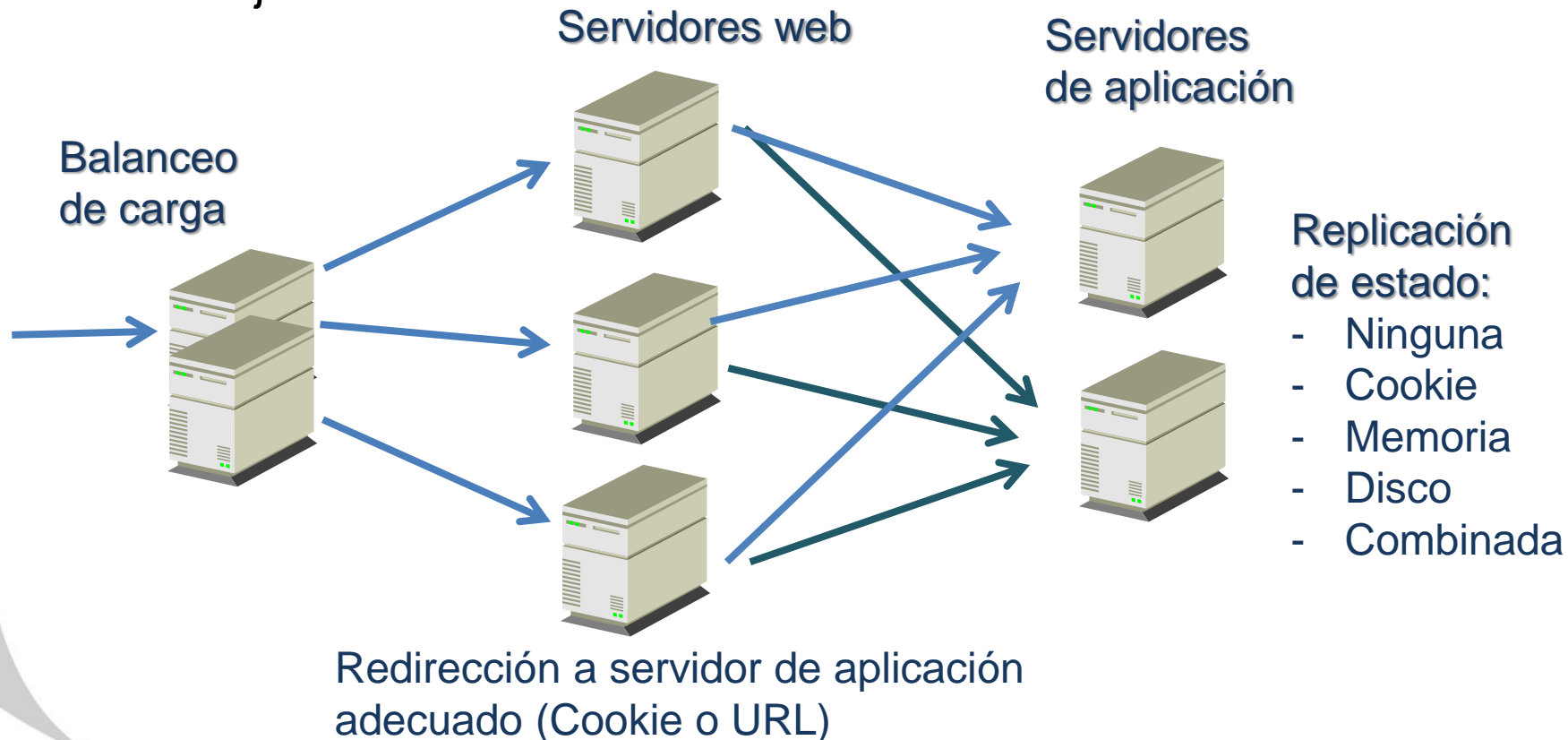
Diseño: Clusters - balanceo de carga

- Balanceo con failover:



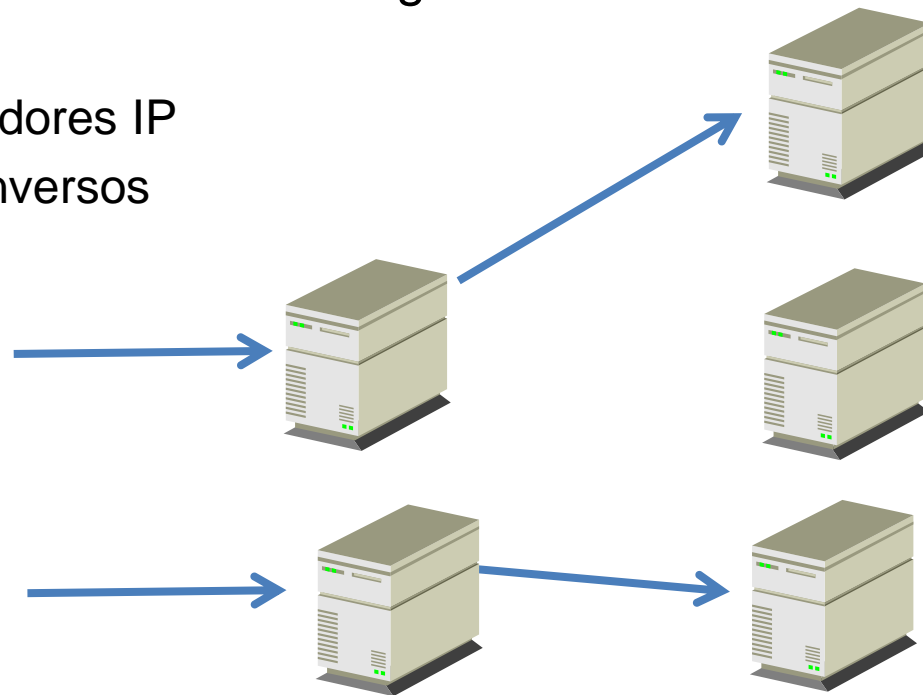
Diseño: Clusters - balanceo de carga

- Manejo de estado



Diseño: Clusters - balanceo de carga

- Técnicas de balanceo de carga
 - DNS
 - Balanceadores IP
 - Proxies inversos



Diseño: Clusters - balanceo de carga

- DNS
 - Directorio distribuido
 - Nombre de host – dirección IP
 - Nombre de dominio – servidor de nombres
 - Admite redundancia:
 - Direcciones IP
 - Servidores de nombres
 - “Distribución” más que “balanceo”

Diseño: Clusters - balanceo de carga

- Balanceadores IP
 - Conmutadores de nivel 3 de alto nivel
 - Enrutan direcciones IP
 - Hacen traducción de direcciones de red

Diseño: Clusters - balanceo de carga

- Proxies inversos
 - Trabajan a nivel de aplicación
 - Reciben requerimientos y los reenvían (eventualmente los reescriben)
 - Tipos
 - Transparentes
 - Semitransparentes
 - Visibles

Diseño: Clusters - balanceo de carga

- Servicios adicionales de los balanceadores:
 - Asignación de cargas diferenciales: configurar la carga que recibe cada servidor
 - Activación y retiro de servidores: según si se requieren más (por aumento de la carga) o si alguno ha fallado
 - Prioridad diferencial al tráfico: asignar prioridades dependiendo del tipo de tráfico
 - Encaminamiento dependiente del URL
 - Filtrado de contenidos

Diseño: Clusters - balanceo de carga

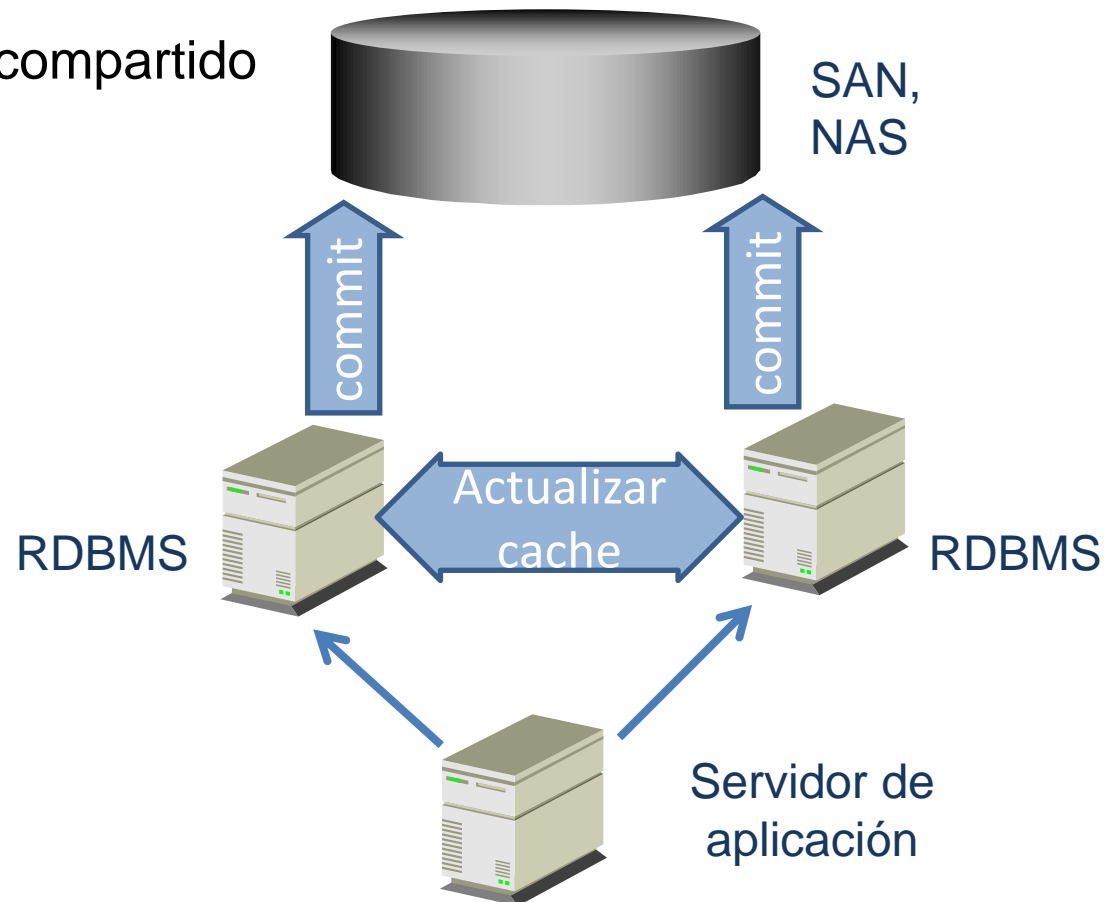
- Selección de blancos
 - Aleatorio: se selecciona un destino al azar
 - Circular: se seleccionan los destinos en orden uno tras otro
 - Circular ponderado: la capacidad del servidor determina la frecuencia con la que se le envían requerimientos
 - Más rápido: se selecciona el que responda más rápido
 - Menor número de conexiones: se selecciona el que tenga menor carga
 - Menor número de conexiones ponderado: se envía según la capacidad del servidor pero morigerado por su carga
 - Balanceo adaptativo: se determina la carga dinámicamente

Diseño: Clusters – bases de datos

- Almacenamiento compartido (shared disk)
- Multi-maestro
- Maestro – esclavo
- *Shared-nothing*

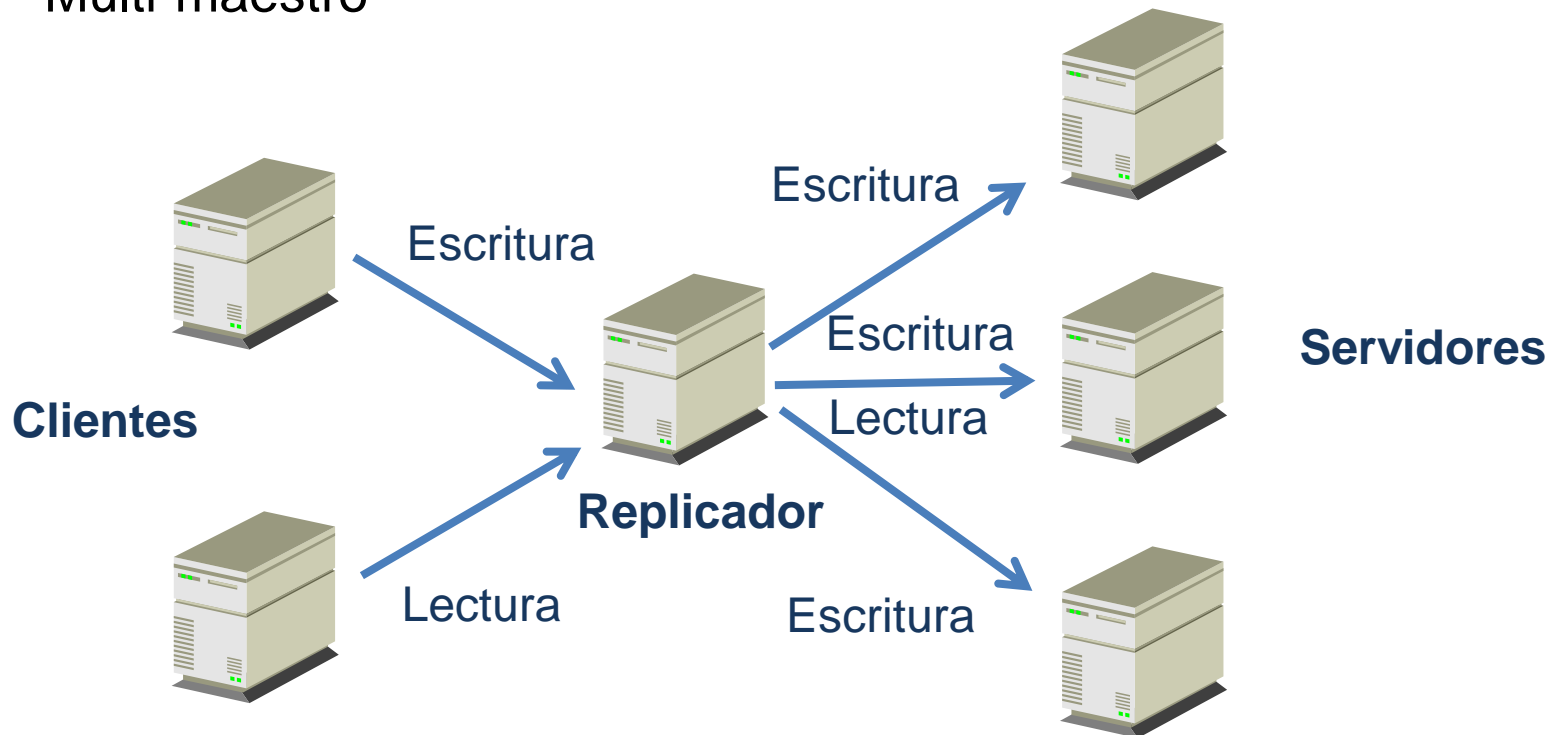
Diseño: Clusters – bases de datos

- Almacenamiento compartido (shared disk)



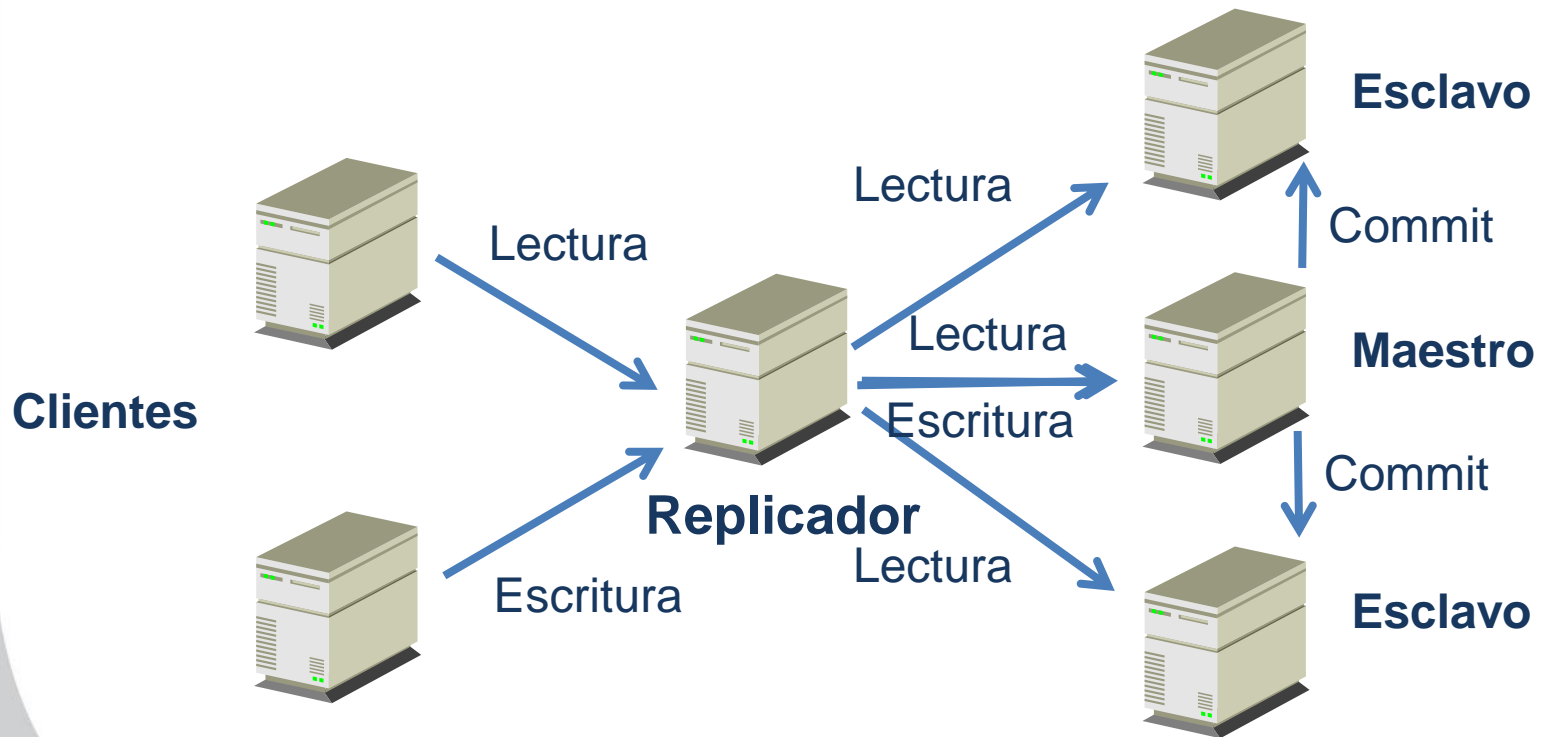
Diseño: Clusters – bases de datos

- Multi-maestro



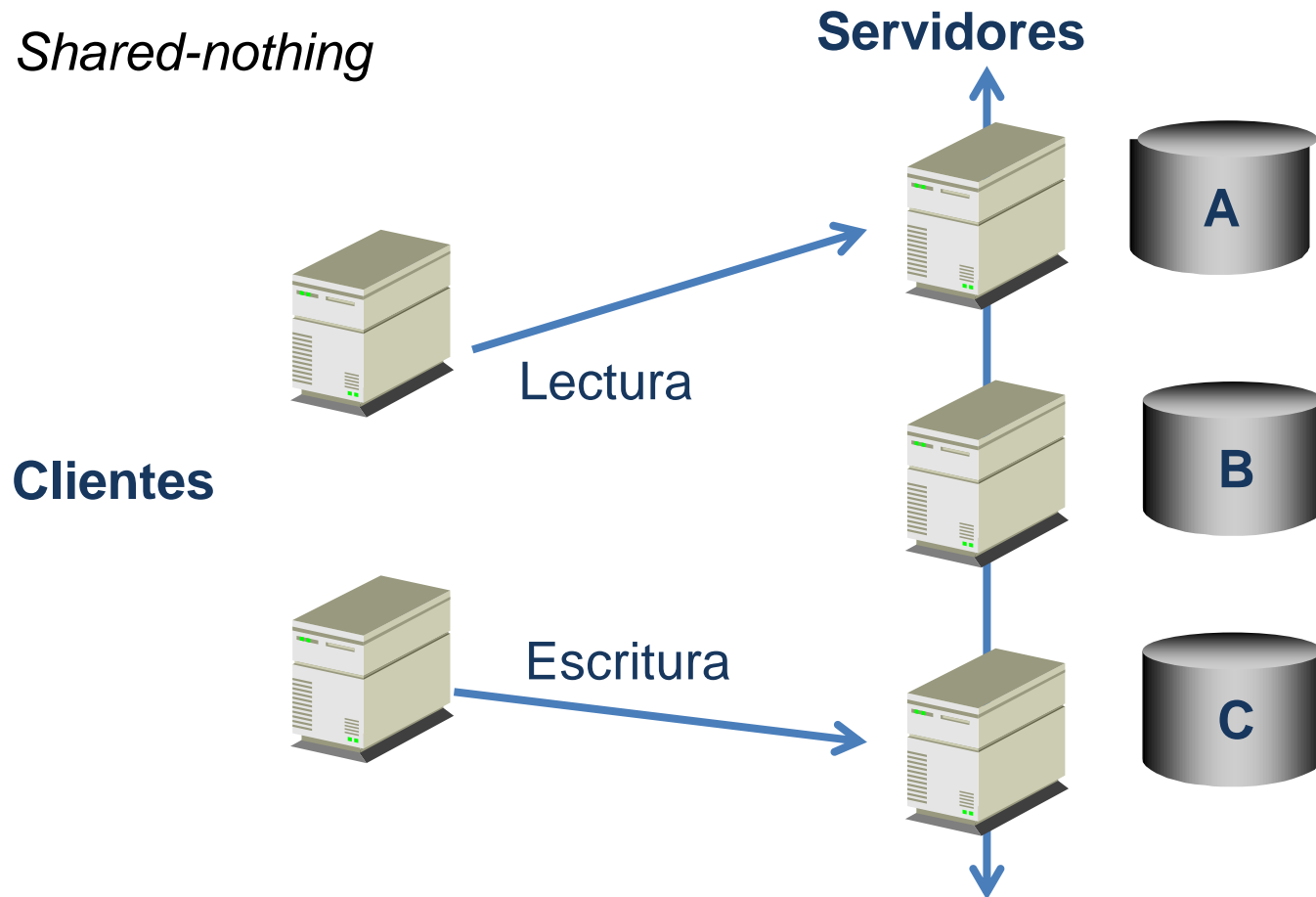
Diseño: Clusters – bases de datos

- Maestro - esclavo



Diseño: Clusters – bases de datos

- *Shared-nothing*



Alternativas de diseño

- Disponibilidad:

Disponibilidad	Definición	Descripción
99%	Local	Stand alone
99.5%	Local + RD estándar	Réplica local
99.9%	AD local	Cluster
99.95%	AD Remota	Cluster extendido
99.99%	AD local + Remoto	Cluster local con failover remoto
99.999%	AD local + AD Remoto	Cluster local – cluster remoto

RD = Recuperación de Desastres

AD = Alta Disponibilidad

Referencia:

Infrastructure Architecture, IASA Associate Course

Alternativas de diseño

- Clusters:
 - Escalamiento vertical:
 - Mejores recursos o mayor capacidad
 - Multiprocesamiento (solución concurrente)
 - Cluster de procesamiento (solución concurrente)
 - Cluster de failover
 - Cluster de balanceo de carga
 - Sin redundancia en el balanceador
 - Con redundancia en el balanceador

Alternativas de diseño

- Clusters:

Solución	Desempeño	Capacidad		Disponibilidad	Escalabilidad
		Productividad	Carga		
Esc. vertical					
más recursos	X	X	X		
multiproc.	X	X		X	X
Cluster					
cómputo	X	X	X		
failover				X	
balanceo					
con redund.		X	X	X	X
sin redund.		X	X		X

Alternativas de diseño

- Balanceo de carga
 - DNS
 - Fácil
 - No es flexible (distribución más que balanceo)
 - Disponibilidad del balanceador incorporada
 - No garantiza disponibilidad automática de servidores

Alternativas de diseño

- Balanceo de carga

Balanceo	Flexibilidad	Velocidad	Cuello de botella
IP			
Enrut. directo	-	+	-
NAT	-	+	+
Proxy inverso	+	-	+

Alternativas de diseño

- Bases de datos

Solución	Desempeño	Balanceo	Disponibilidad	Escalabilidad
<i>Shared disk</i>	+	+	+	+/(6)
Replicación				
multi-maestro	+(1)	+(1)	+	+(1)
maestro-esclavo	+(1)	+(1)	+(4)	+(1)
Shared nothing	+(2)	+/(3)	+(5)	+(2)

(1)= para las lecturas

(2)= depende del shipping y la distribución

(3)= estático; depende de patrones de acceso

(4)= puede ser lento

(5)= solo si hay esclavos; puede ser lento

(6)= limitada por sincronización

Referencias por temas

Alta disponibilidad. Clusters. High availavility and disaster recovery, Klaus Schmidt, Ed. Springer, 2006. Capítulo 6 (hasta página 165 y 176-187)