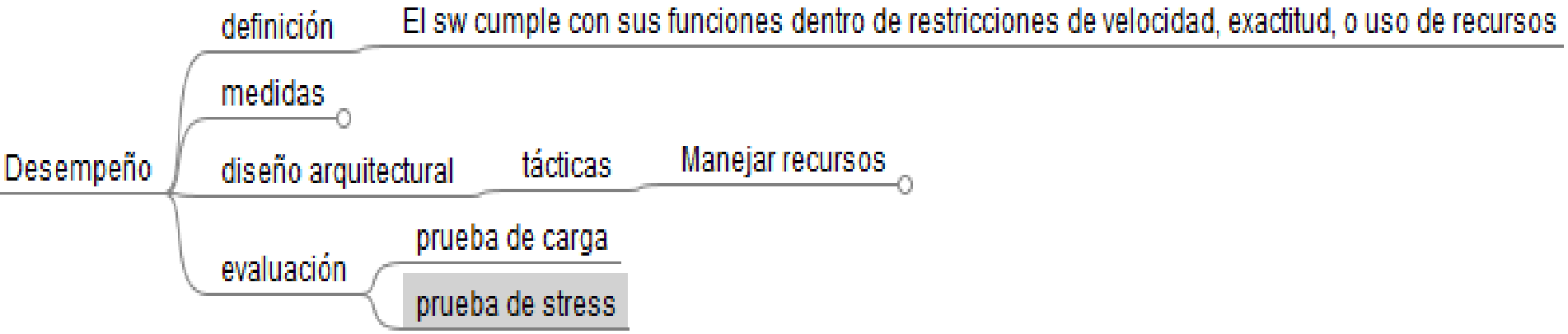
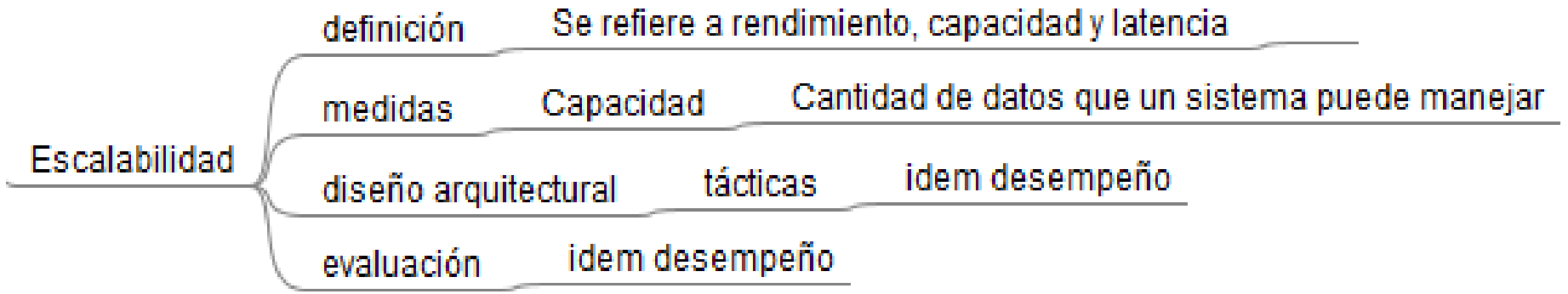
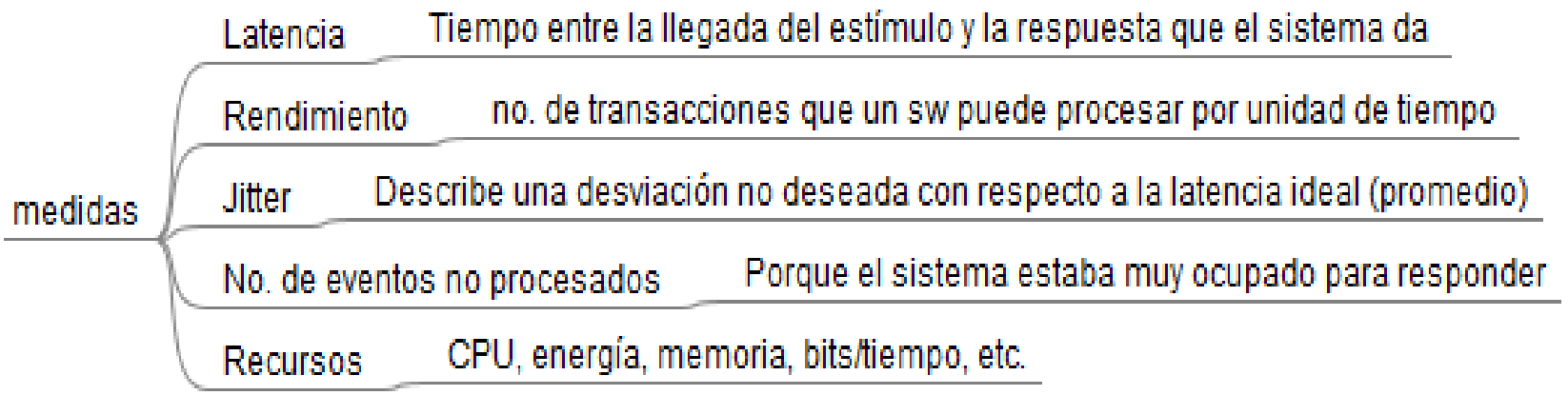
Desempeño y Escalabilidad

Definición de cada requerimiento





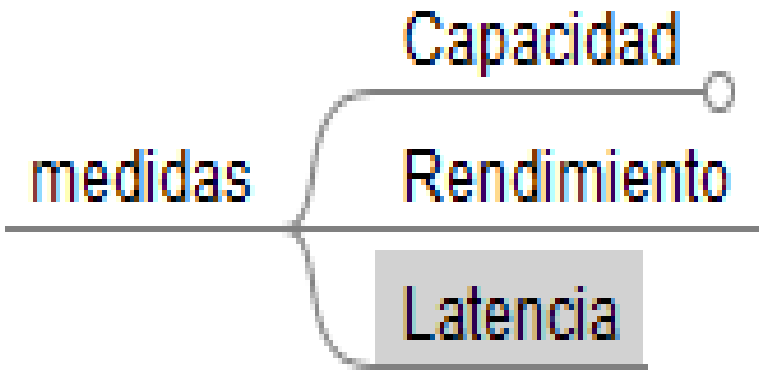
Medidas de desempeño



Escenario de calidad de desempeño

- El sistema envía al paciente y/o personal médico una alarma en menos de 1.5 segundos

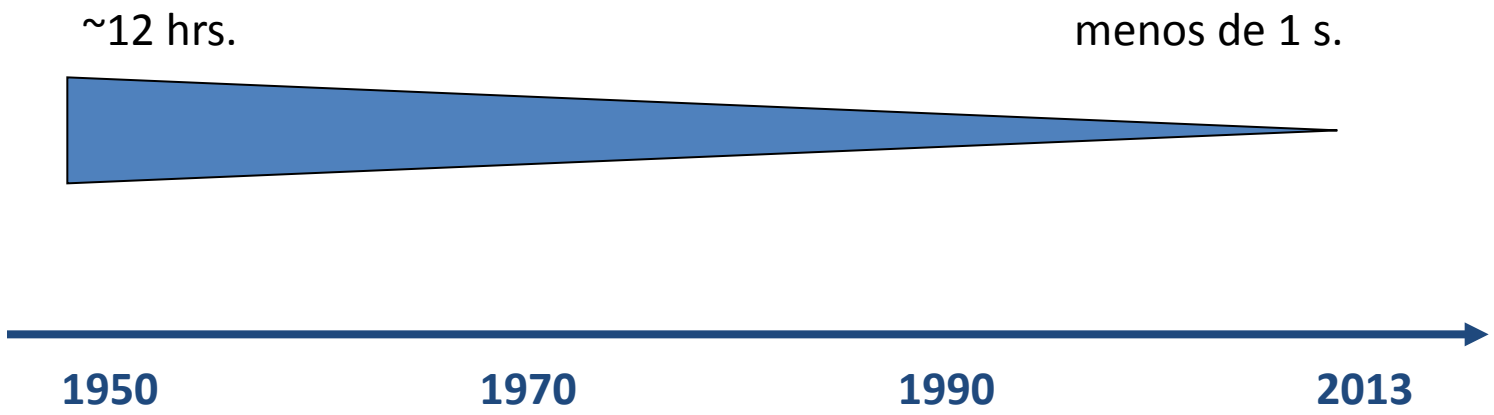
Medidas de escalabilidad



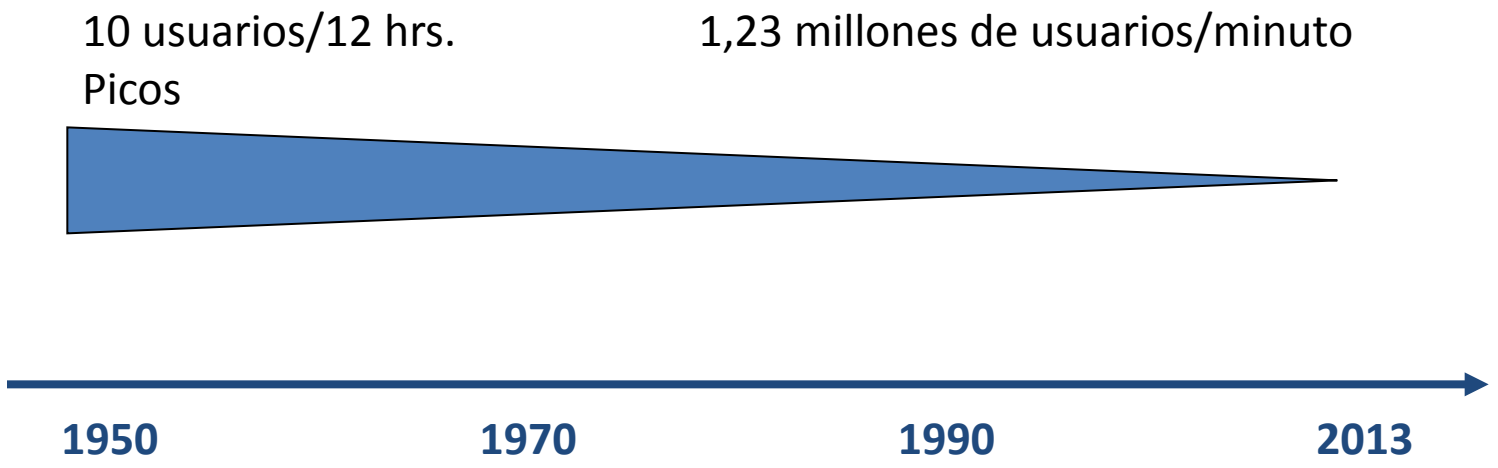
Escenario de calidad de escalabilidad

- El sistema debe manejar 1M de transacciones en 1 minuto

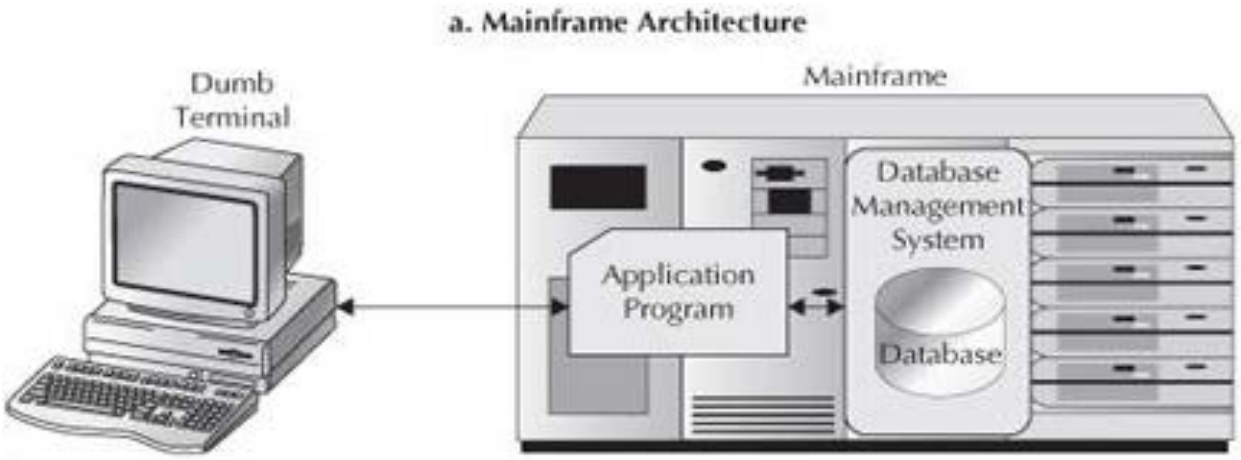
Evolución en la expectativa de los clientes latencia



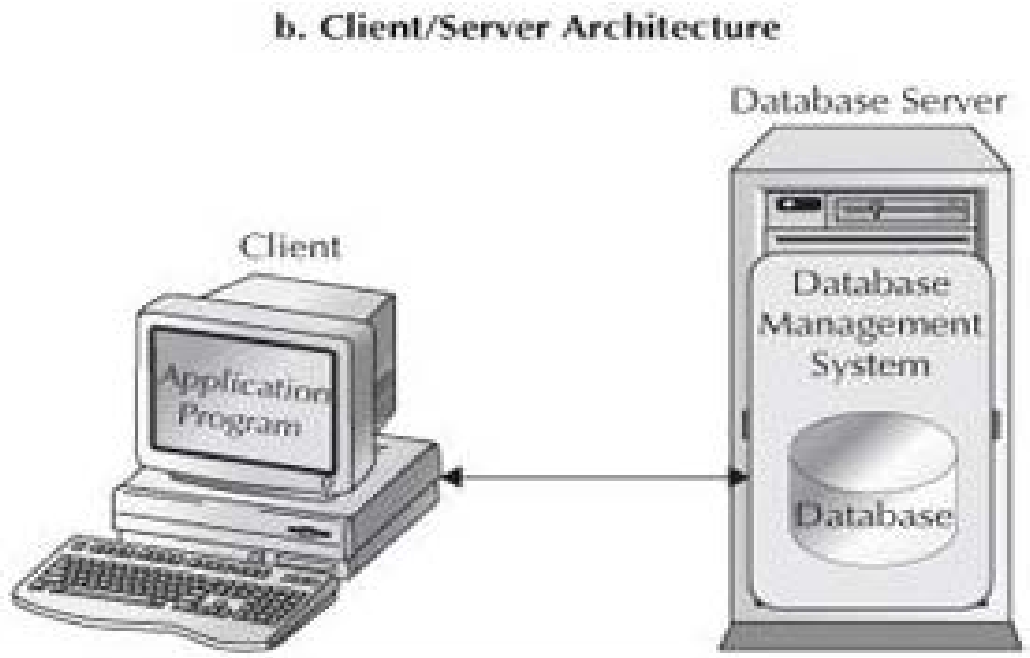
Evolución en la expectativa de los clientes rendimiento



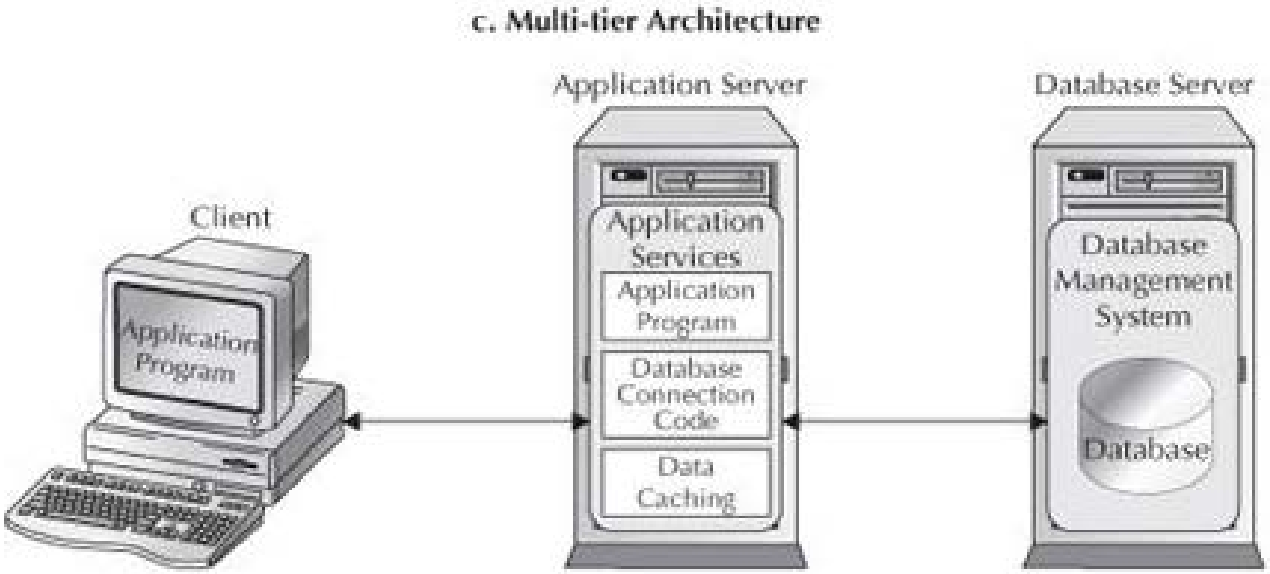
Arquitectura mainframe



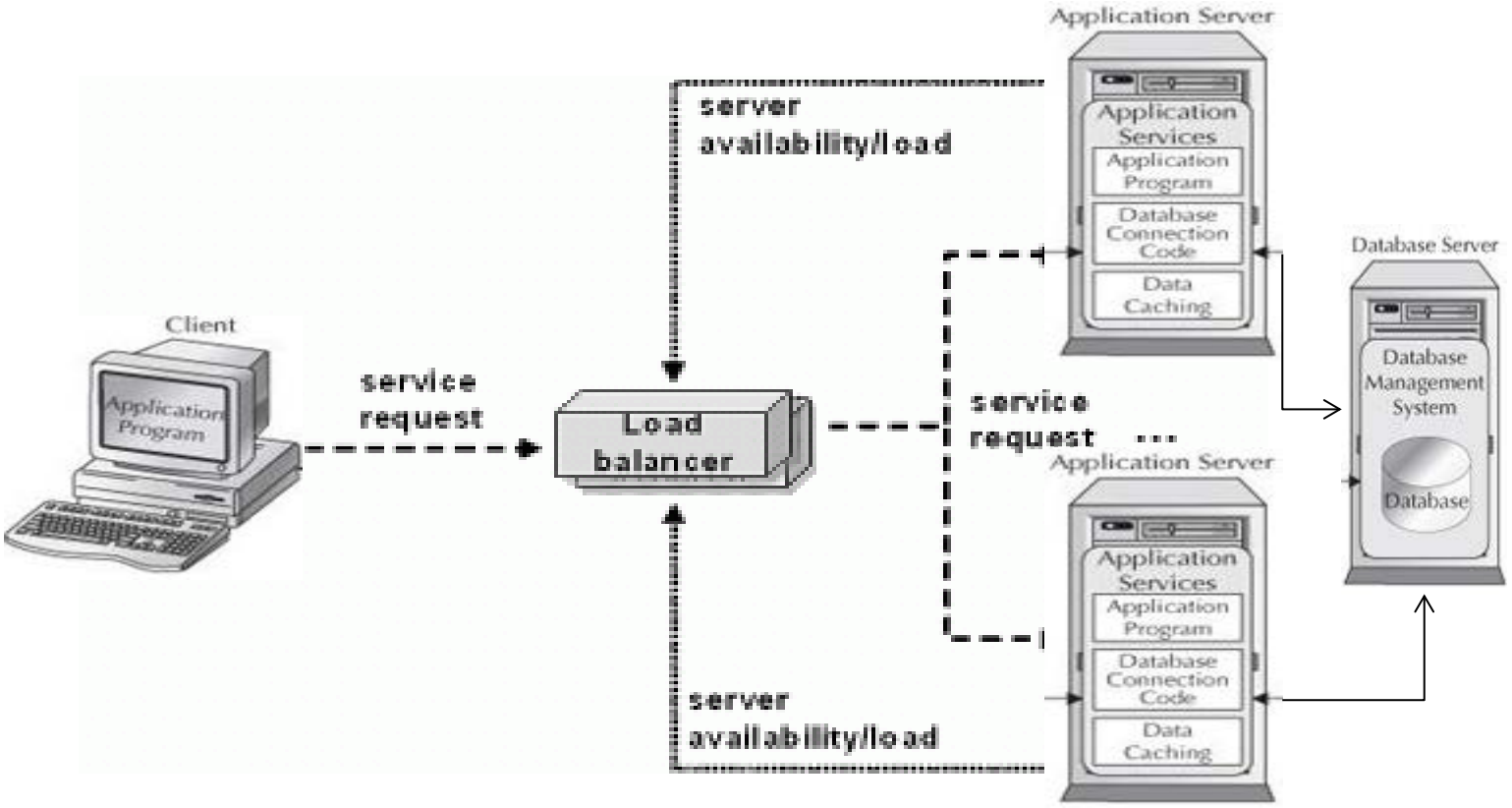
Arquitectura cliente/servidor (C/S)



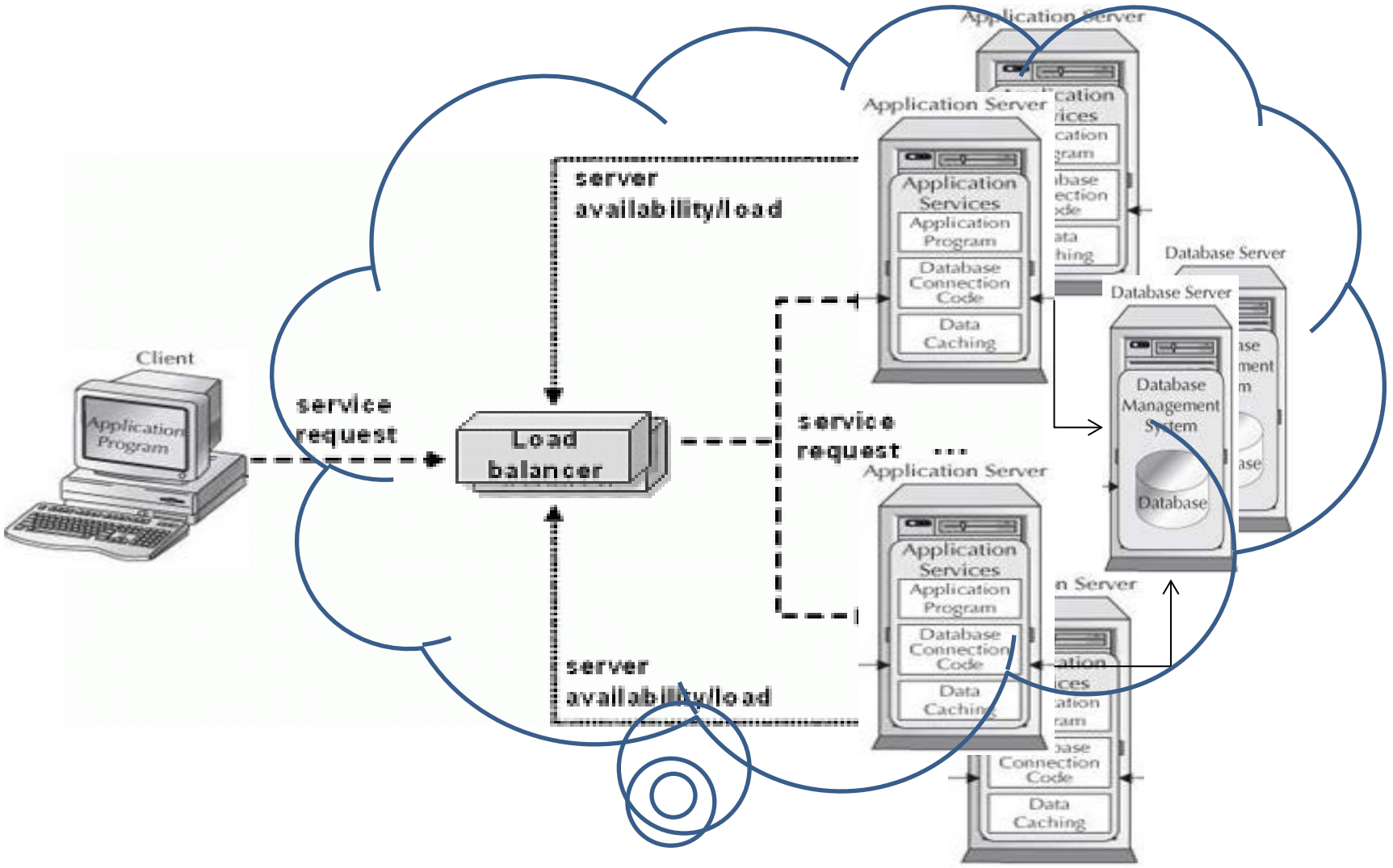
Arquitectura multi-tier



Arquitectura multi-tier con balanceo de carga



Arquitectura multi-tier con balanceo de carga en la nube



Pruebas de desempeño y escalabilidad

- Prueba de carga

Como el sw maneja niveles de carga anticipada que están en incremento (de carga ridícula a carga que el sistema es incapaz de manejar)

- Manejo de múltiples usuarios
- Integridad de las sesiones de usr.
- Latencia

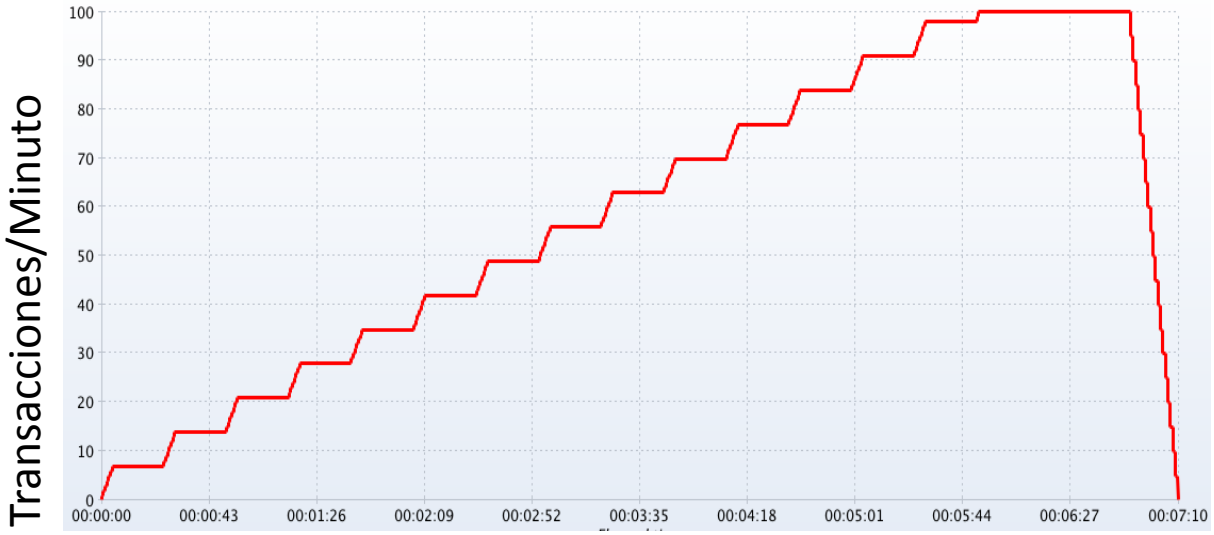
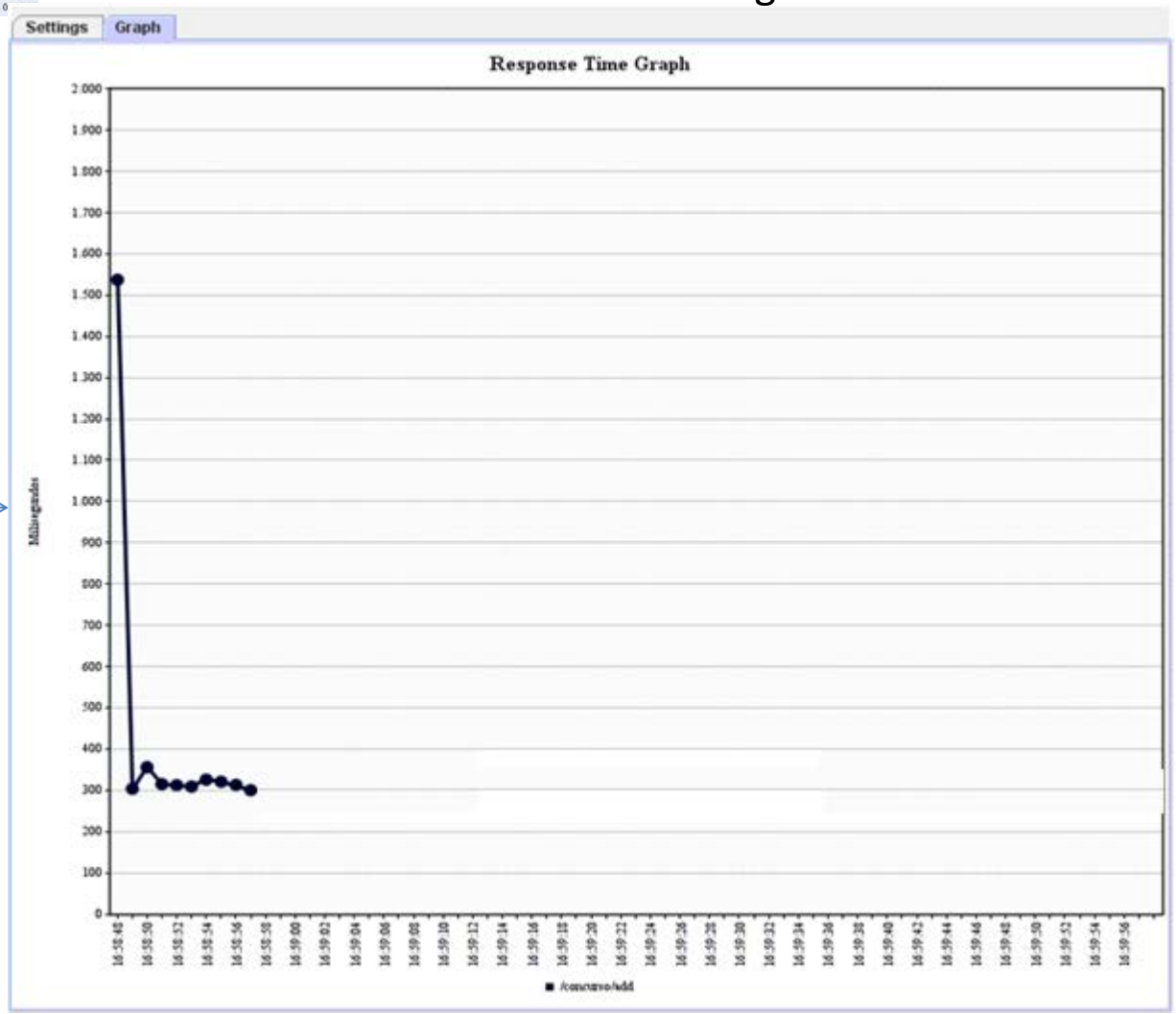


Fig. 1

Por cada escalón de la fig. 1, se podría obtener un gráfico de tiempo de respuesta similar al de la fig. 2

Fig. 2



Parámetros de la prueba

1 iteración
No. de usuarios: 10
Ramp-up= 10 s.

2 iteración
No. de usuarios: 20
Ramp-up= 10 s.

…

14 iteración
No. de usuarios: 100
Ramp-up= 10 s.

Fig. 1

Cuándo parar de probar?

1) El tiempo de respuesta sobrepasa la medida del escenario
2) Hay errores
3) La herramienta de prueba se bloquea

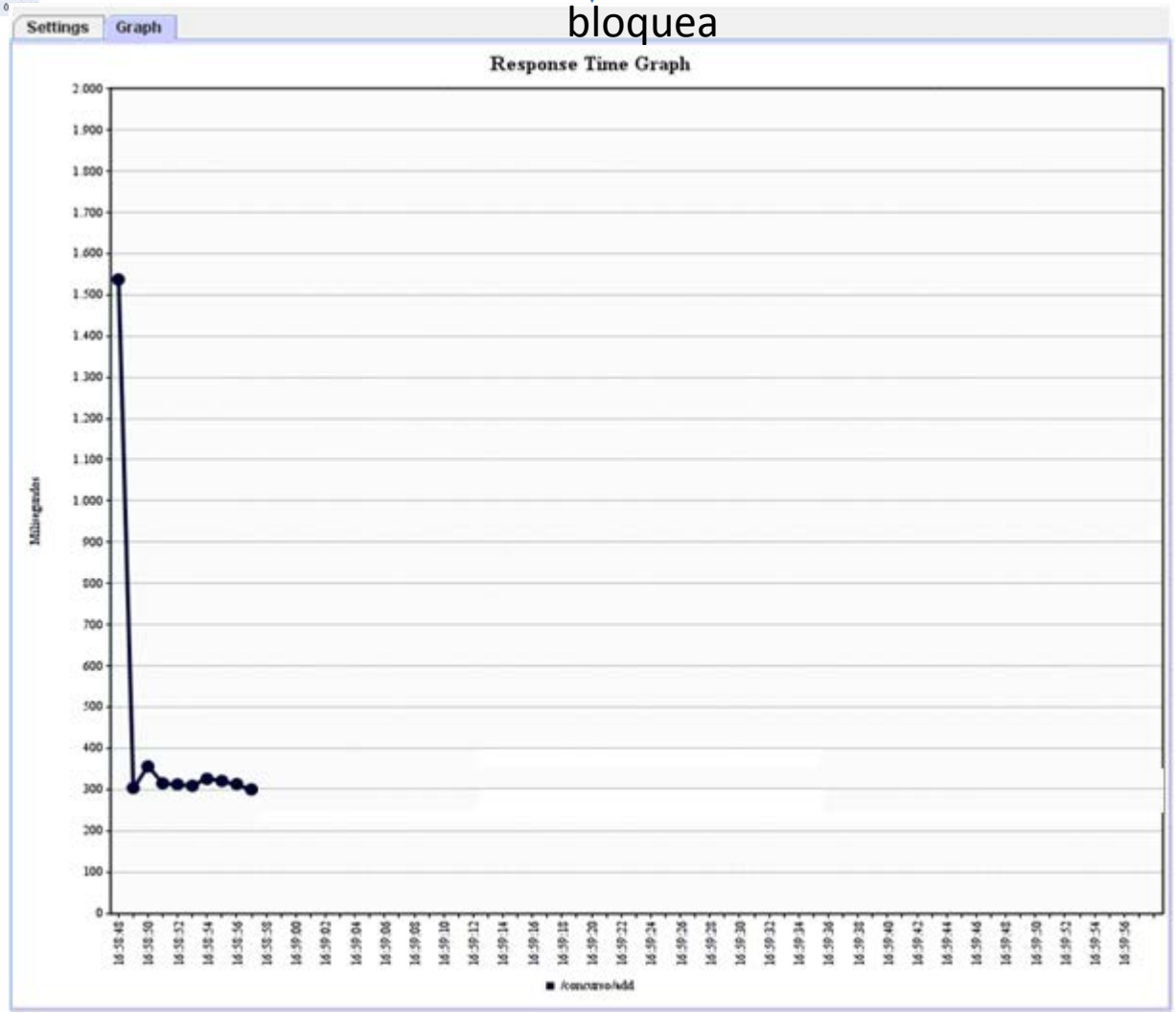


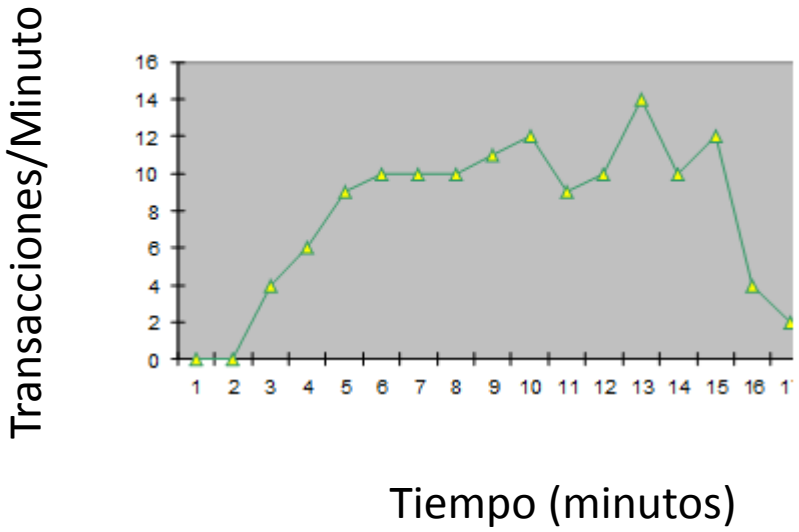
Fig. 2

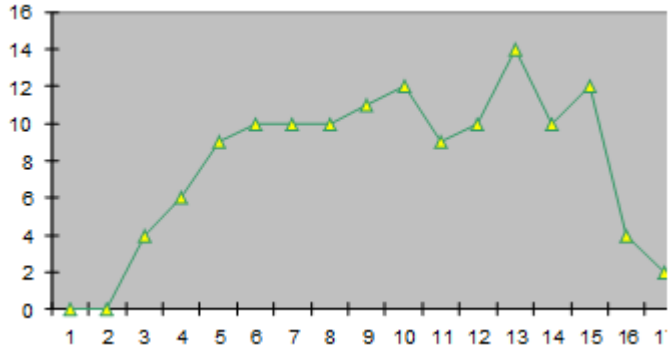
Prueba de desempeño y escalabilidad

- Prueba de stress

Como el sw maneja un carga extrema repentina. Antes/después del pico puede haber una carga baja

- Manejo de picos
- Habilidad para liberar los recursos





Por la prueba, modelada en la fig. 1, se podría obtener un gráfico de tiempo de respuesta similar al de la fig. 2

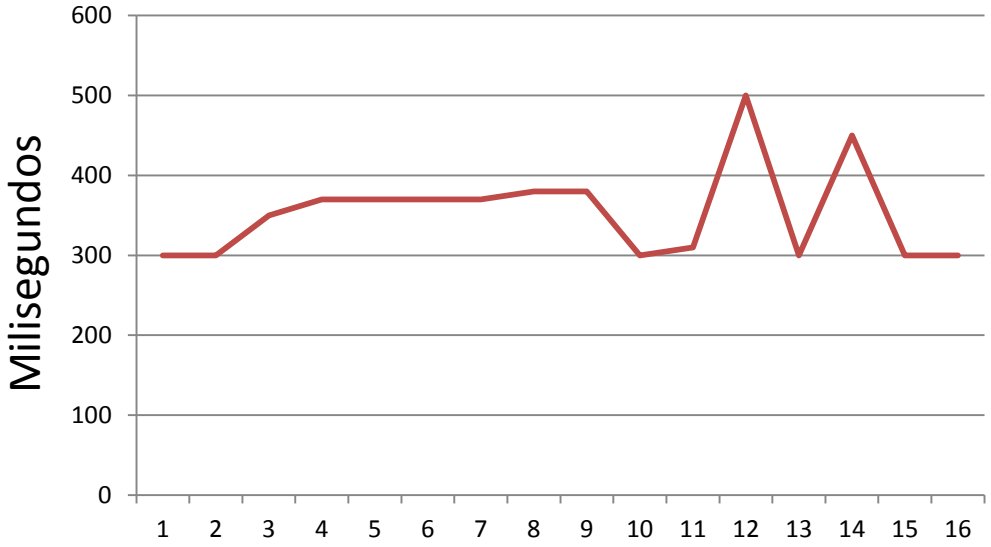
Parámetros de la prueba

1 Iteración

2-3 min
No. de usuarios: 0-4

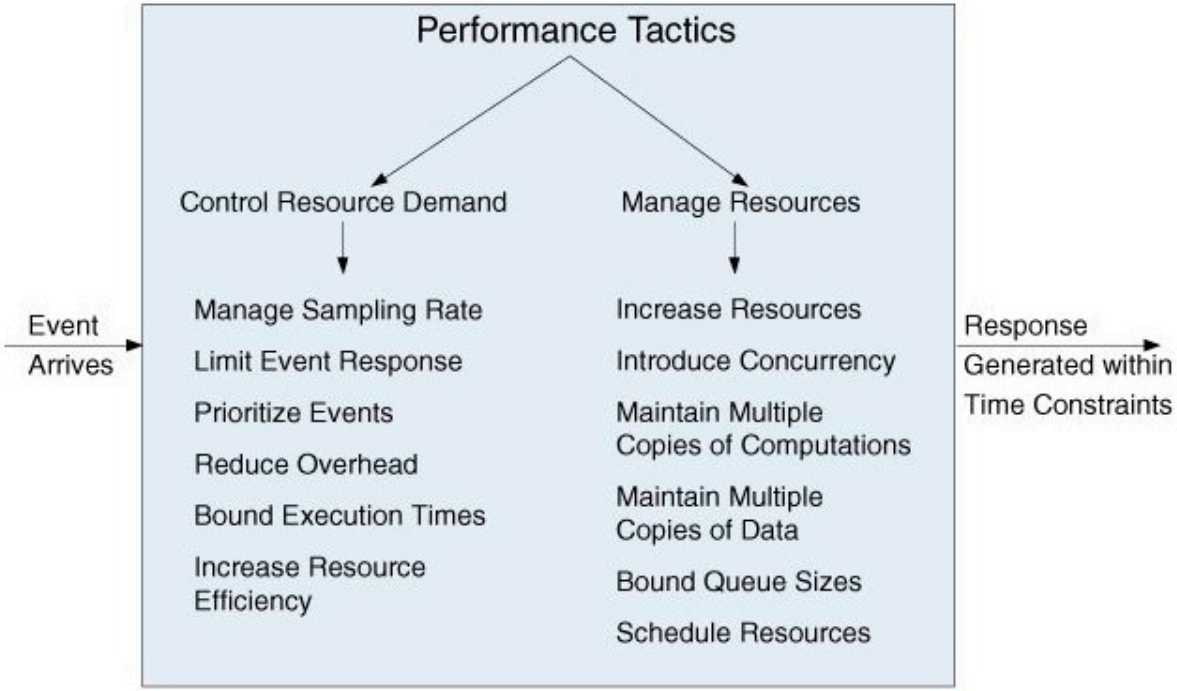
…
12-13 min
No. de usuarios: 10-16

Response time graph



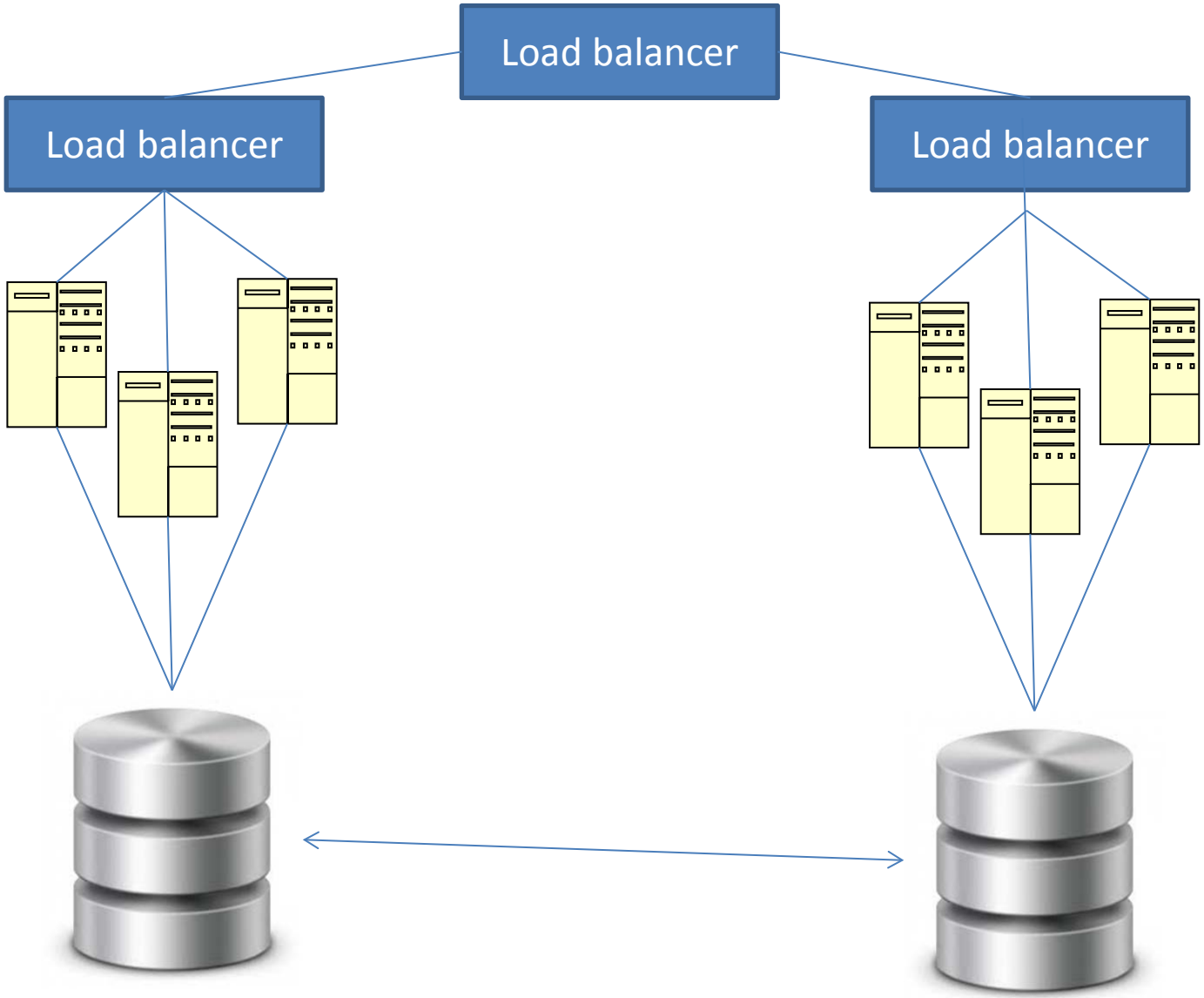
Tácticas

- Desempeño



Tomado de Software Architecture in Practice (3rd Edition). Len Bass

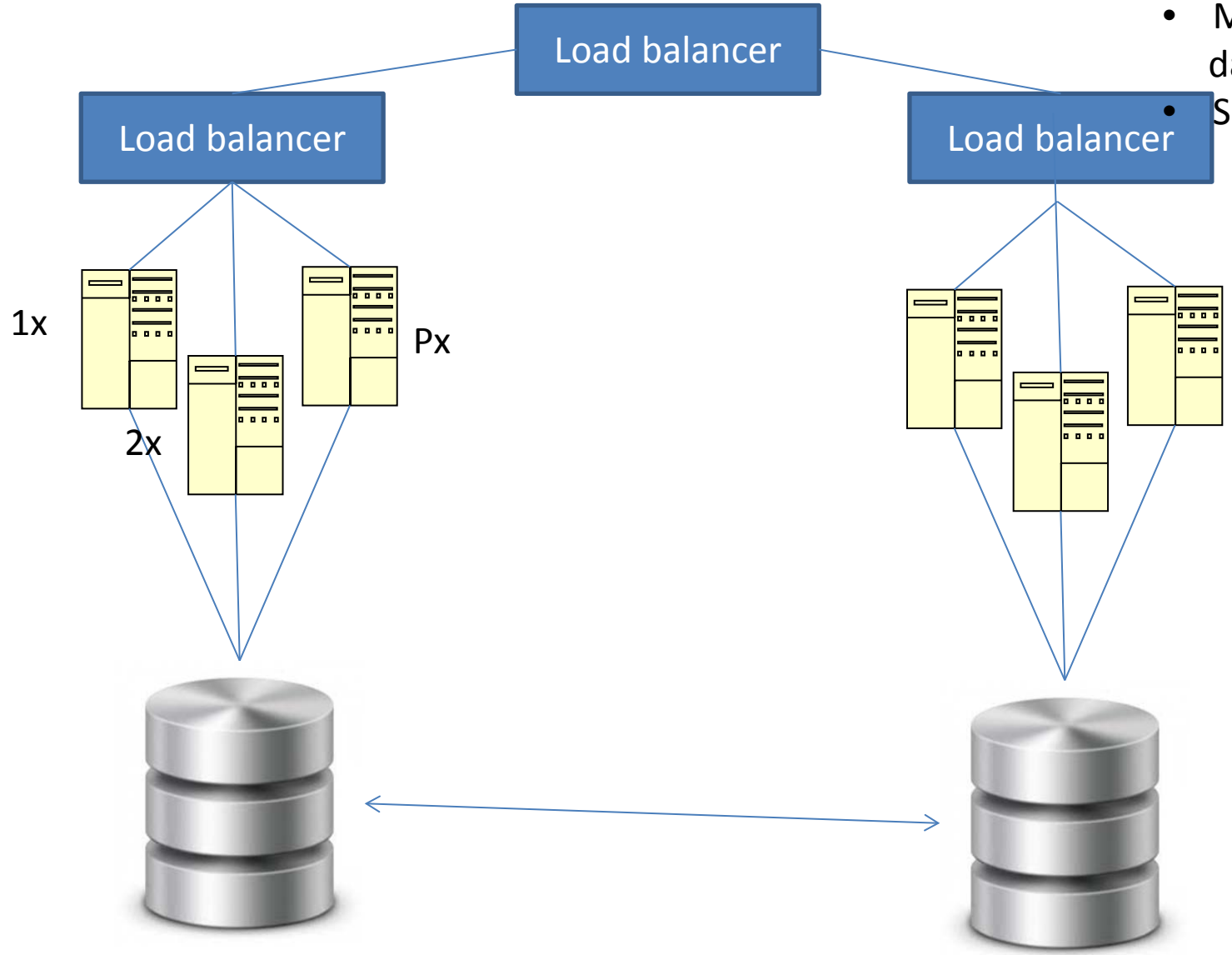
Data center architecture



Tactics

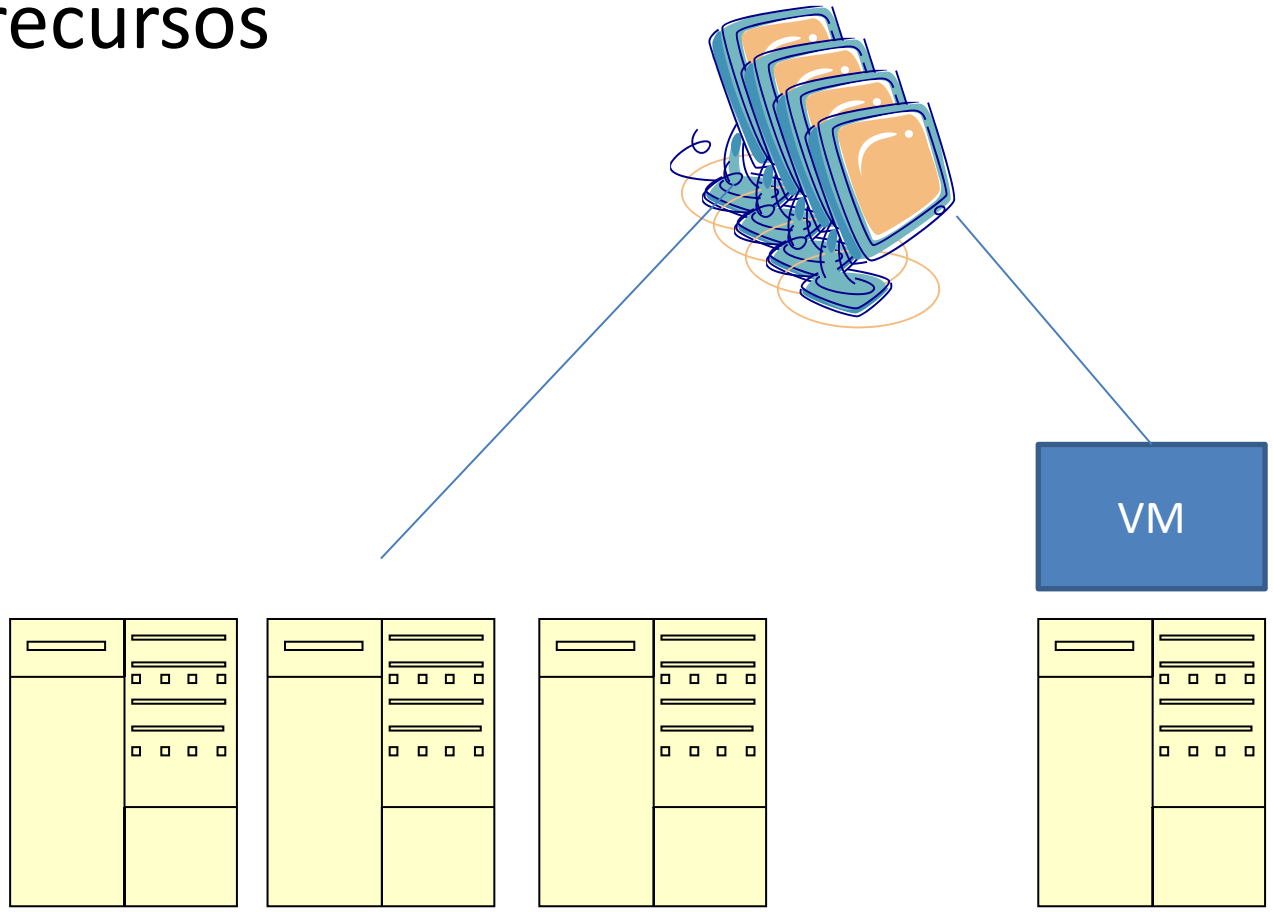
Data center architecture

- Multiple copies of computation
- Multiple copies of data
- Schedule resources

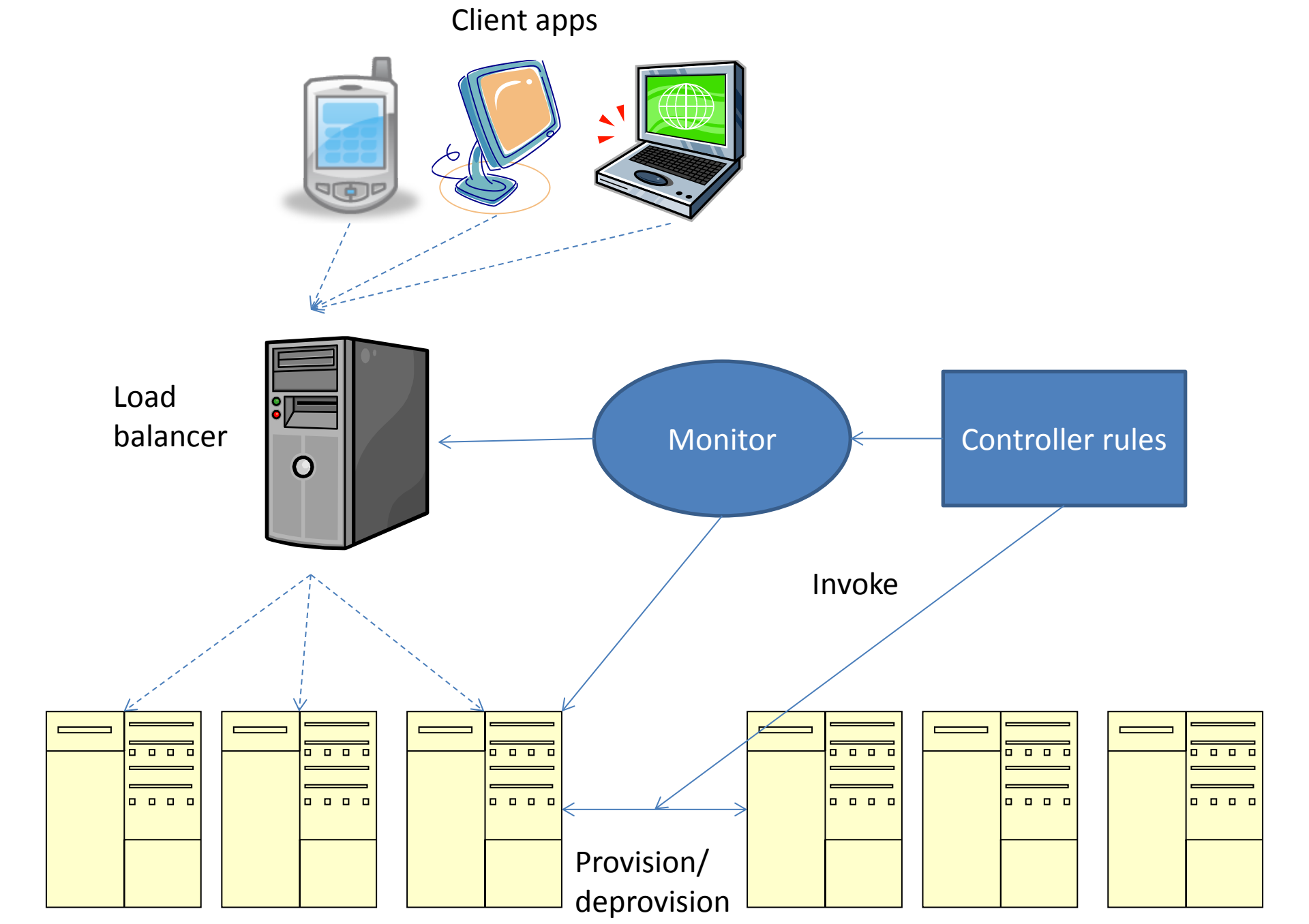


Virtualización en la nube

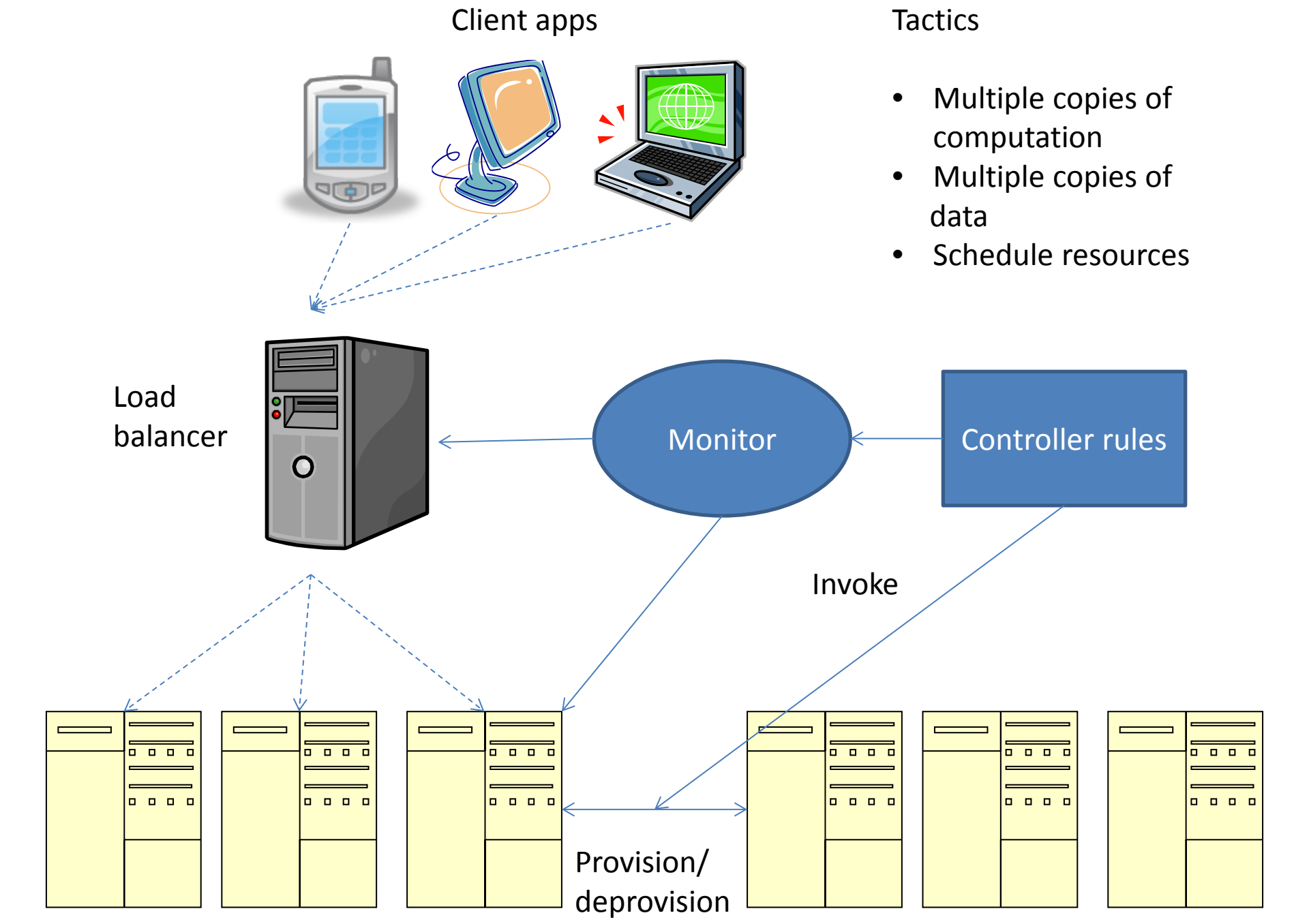
- Creación de instancias virtuales de algo
- Clientes terminan utilizando los mismos recursos



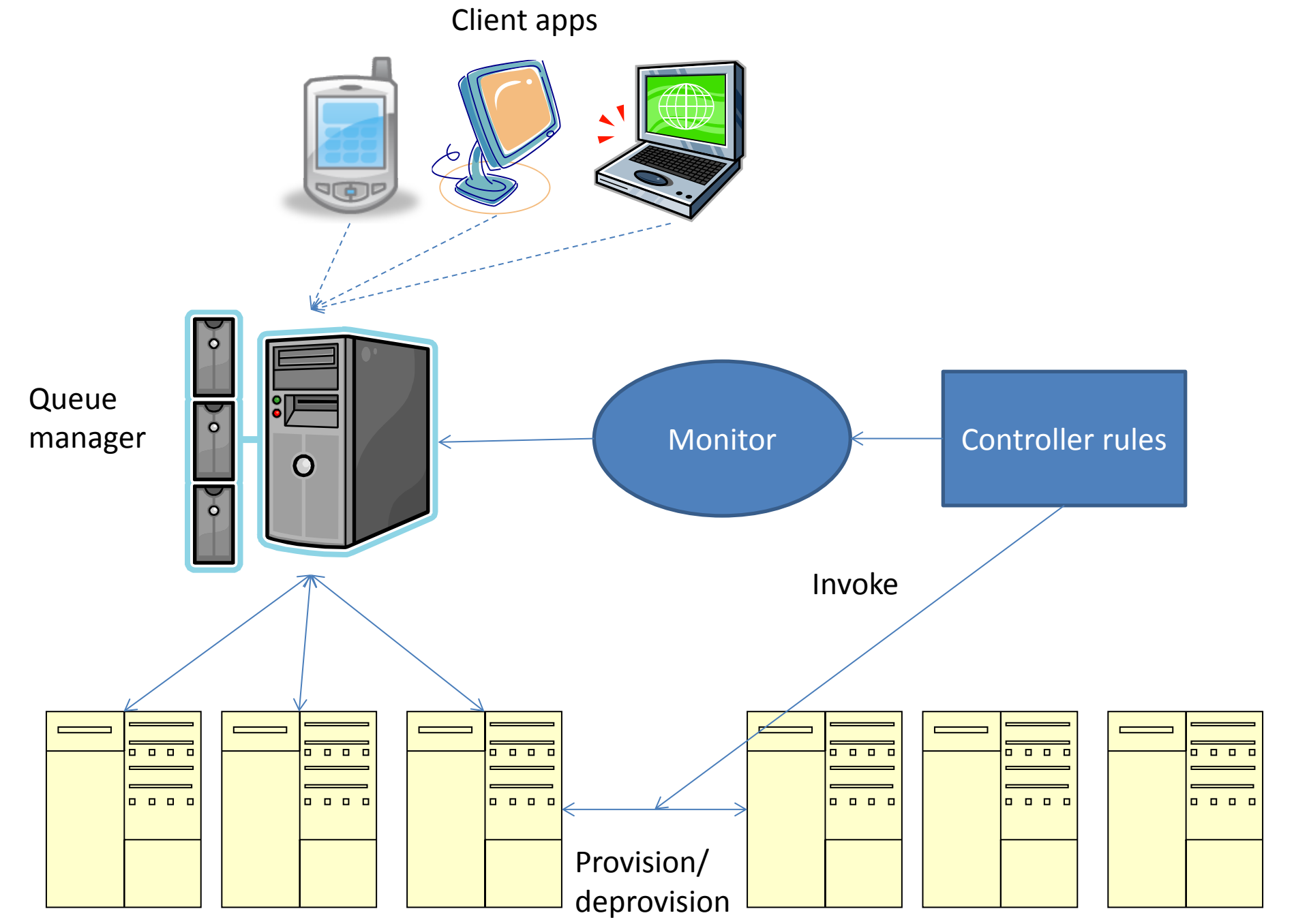
Push Architecture Pattern



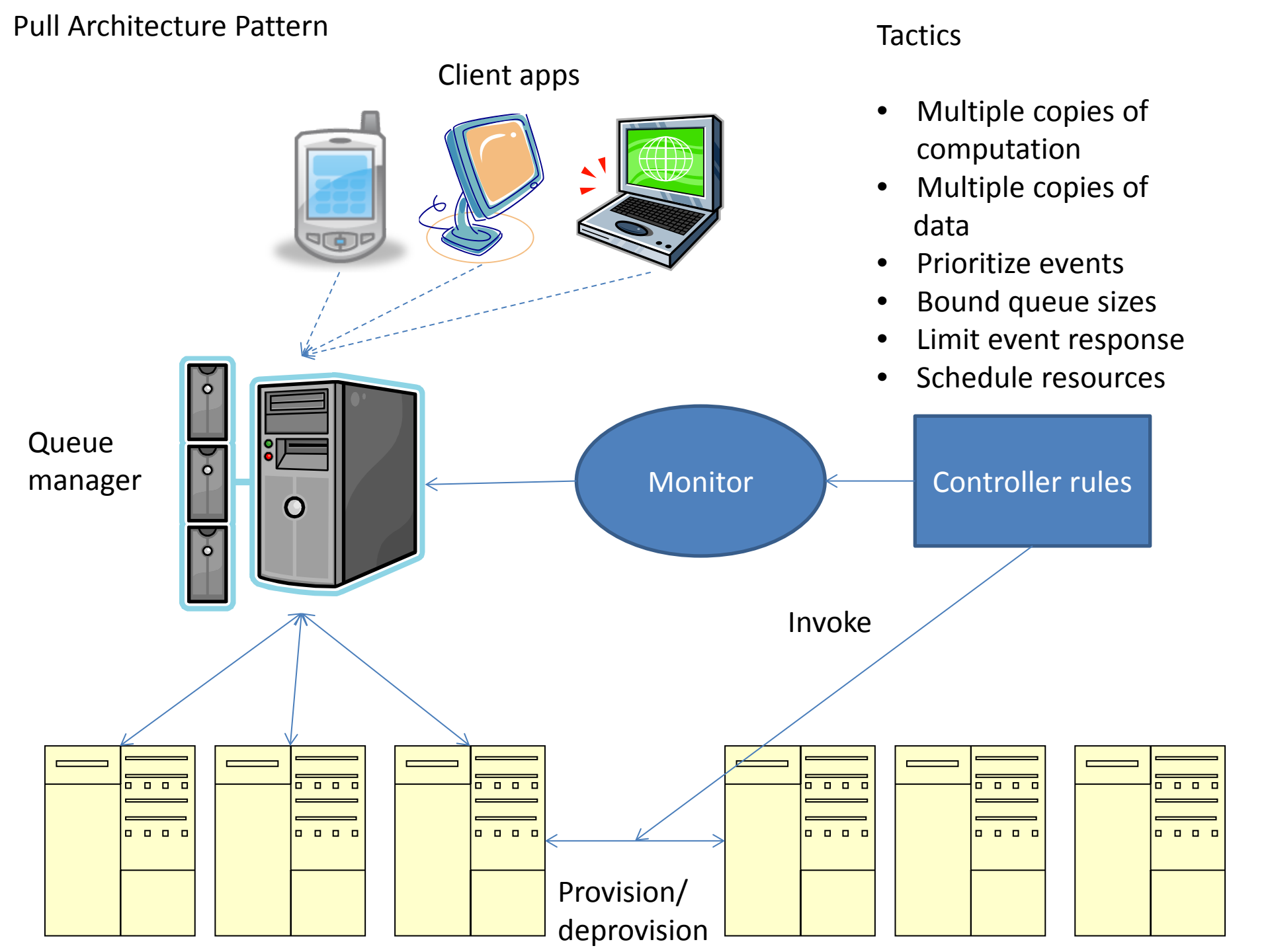
Push Architecture Pattern



Pull Architecture Pattern



Pull Architecture Pattern



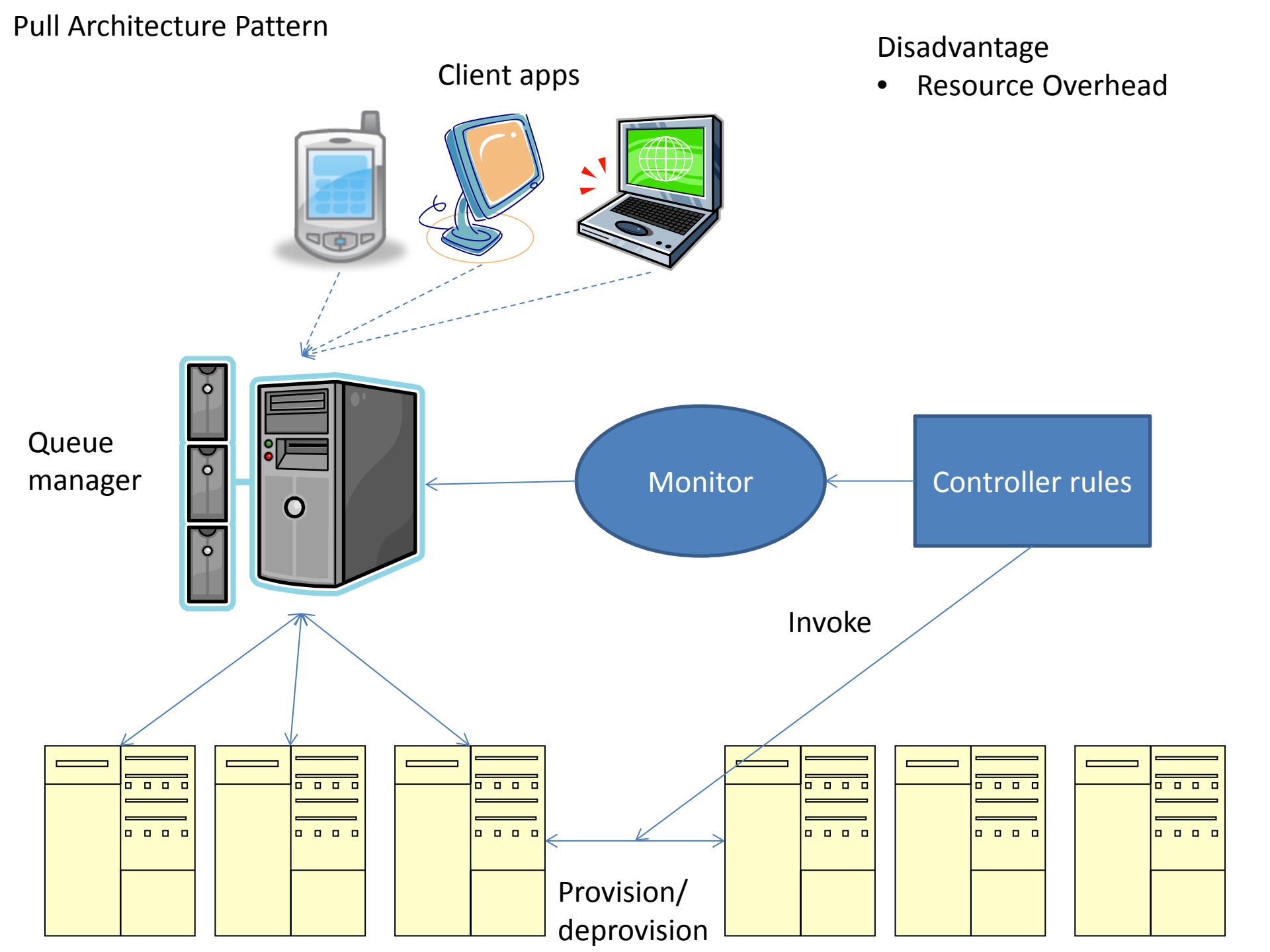
Tactics

- Multiple copies of computation
- Multiple copies of data
- Prioritize events
- Bound queue sizes
- Limit event response
- Schedule resources

Pull Architecture Pattern

Disadvantage

- Resource Overhead



Políticas de programación de recursos (scheduling)

- Simple
  - FIFO: orden de llegada, misma prioridad
- "Inteligentes"
  - Prioridades a peticiones
  - Estado de los recursos
  - Híbridas

“Inteligentes”

- Prioridades a peticiones
  - Peticiones tipadas
    - Preestablece prioridades para cada tipo de petición y atiende las peticiones en ese orden
  - Peticiones no tipadas
    - Asigna prioridades variables según lo que llegue
      - Round robin
      - Earliest deadline

“Inteligentes”

- Round robin

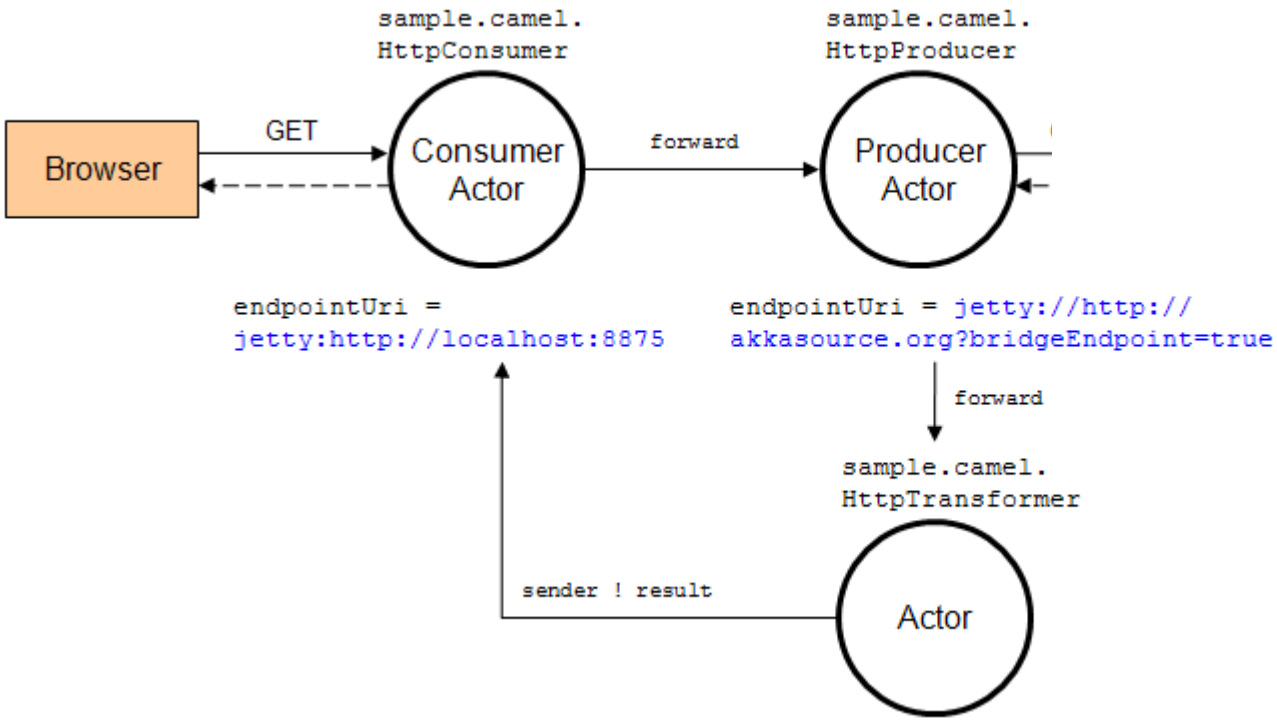
Process	Duration	Order	Arrival Time
P1	3	1	0
P2	4	2	0
P3	3	3	0

Suppose time quantum is: 1 unit, P1, P2 & P3 never block

“Inteligentes”

- Estado de los recursos
  - Contar no. de peticiones
    - Asegura que cada recurso obtiene la proporción de carga que se le asignó
  - Peticiones pendientes
    - Sabe el no. de peticiones que tiene un recurso actualmente y asigna la próxima petición al recurso que tenga el menor número

Actors architecture in Play



Actors architecture in Play