

# Tema 2

## Alta disponibilidad y escalabilidad



Pedro A. Castillo Valdivieso  
Depto Arquitectura y Tecnología de Computadores  
Universidad de Granada  
[pacv@ugr.es](mailto:pacv@ugr.es)

## Índice



### [ Introducción ]

Concepto de alta disponibilidad

Concepto de escalabilidad

Escalar un sitio web

Conclusiones

## Introducción

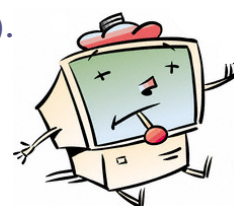
disponibilidad

escalabilidad

conceptos más importantes al diseñar una granja web

## Introducción

El éxito de una empresa depende de que los usuarios tengan buena experiencia al visitarla (su web).



TIRA.ECOL (CC some rights reserved) - Javier Meléndez



[Version Original] <http://tira.ecompositivus.org>



[English Version] <http://comic.ecompositivus.org>

## Introducción

Nuestros servidores deben dar el mejor servicio a todos los usuarios y deben estar todo el tiempo disponible (24/7).

- Disponibilidad
- Escalabilidad
- Balanceo de carga



## Índice



Introducción

[ Concepto de alta disponibilidad ]

Concepto de escalabilidad

Escalar un sitio web

Conclusiones

## Alta disponibilidad

Mala impresión al entrar en un sitio y está caído.

Esperamos que una web esté disponible siempre.

**Disponibilidad:** capacidad de aceptar visitas las 24h todos los días.



## Alta disponibilidad

Cuando un sitio no está disponible se dice que se ha caído o sufre un problema de no-disponibilidad:

- Tiempo de no-disponibilidad programado.
- Tiempo de no-disponibilidad no programado.

Sólo debería haber "tiempos de no-disponibilidad programados" (y lo más cortos posibles)

actualizaciones del SO, de aplicaciones o de hardware



## Alta disponibilidad

Medir la disponibilidad dando un porcentaje.

Escala “punto nueve”:

$$100 - (\text{tiempoCaido} / \text{periodoTiempo}) * 100$$

Por ejemplo:

caída de 1h en un día -> 95.83333% de disponibilidad

caída de 1h en una semana -> 99.404% de disponibilidad

Lo ideal es tener un 100% de disponibilidad.

## Alta disponibilidad

Un 100% de disponibilidad es no sufrir caídas no-programadas

Los sitios web se conforman con alcanzar un 99.9% ó 99.99%

Disponibilidad (%)	Periodo de un año
90%	36.5 días
95%	18.25 días
98%	7.3 días
99%	3.65 días
99.9%	8.76 horas
99.99%	52.56 minutos
99.999%	315 segundos
99.9999%	31.5 segundos

## Alta disponibilidad

<http://www.edgeblog.net/2007/in-search-of-five-9s/>

Otra forma de escribir la ecuación:

$$\text{Availability} = \text{Uptime} / (\text{Uptime} + \text{Downtime})$$

Horas en un año (periodo) = 8760

Horas caído (tiempo caído)= 1830

Disponibilidad =  $(8760 - 1830) / 8760 = 0.791$

Disponibilidad =  $100 - (1830 / 8760) * 100 = 79.1$

## Alta disponibilidad

<http://www.edgeblog.net/2007/in-search-of-five-9s/>

¿Cómo podemos calcular la disponibilidad de un sistema?

$$A_s = A_{c1} * A_{c2} * A_{c3} * \dots A_{cn}$$

Si tenemos dos servidores (web+BD) y cada uno tiene 99%,  
la disponibilidad del sistema será  $99\% * 99\% = 98.01\%$

Si cada uno puede estar caído 3.65 días,  
podemos esperar que el sistema  
esté caído un total de 7.3 días en  
un año.

Peor caso: que esté  
caído 3.65 días el  
servidor web y justo  
después caiga el de  
BD otros 3.65 días

Disponibilidad (%)	Periodo de un año
90%	36.5 días
95%	18.25 días
98%	7.3 días
99%	3.65 días
99.9%	8.76 horas
99.99%	52.56 minutos
99.999%	315 segundos
99.9999%	31.5 segundos

## Alta disponibilidad

Pero los sistemas reales son más complejos...  
Hay muchos más elementos y algunos redundantes.  
Necesitamos fórmulas algo más complejas:

(1) Para un sistema con N componentes, la disponibilidad del sistema se calcula como:

$$A_s = A_{c1} * A_{c2} * A_{c3} * \dots A_{cn}$$

Component	Availability
Web	85%
Application	90%
Database	99.9%
DNS	98%
Firewall	85%
Switch	99%
Data Center	99.99%
ISP	95%

Ejemplo de sitio de comercio electrónico con varios puntos de fallo

## Alta disponibilidad

En ese ejemplo, si cualquier componente falla, supondremos que el sistema falla.

La disponibilidad será:

$$85\% * 90\% * 99.9\% * 98\% * 85\% * 99\% * 99.99\% * 95\% = \underline{59.87\%}$$

¿Parece baja? Al usuario le importa que el sistema dé el servicio. Si está caído, le dará igual que sea por el cortafuegos o por fallo de una aplicación web.

¿cómo mejorar la disponibilidad de este sistema?

## Alta disponibilidad

¿y si mejoramos la disponibilidad del servidor web y cortafuegos?

(2) Si el sistema tiene un componente replicado, la disponibilidad de esa parte del sistema completo será:

$$A_s = A_{c1} + (1 - A_{c1}) * A_{c2}$$

Component	Availability
Web	85%
Application	90%
Database	99.9%
DNS	98%
Firewall	85%
Switch	99%
Data Center	99.99%
ISP	95%

En el ejemplo, añadir un segundo servidor web hará:

$$\text{disponibilidad\_web2} = 85\% + (1-85\%) * 85\% = 97.75\%$$

## Alta disponibilidad

Antes teníamos un 59.87% para todo el sistema.

¿Qué disponibilidad tendremos si replicamos el servidor web y el cortafuegos?

Cada uno de esos componentes tendrá ahora 97.75%

Y el sistema:

$$97.75\% * 90\% * 99.9\% * 98\% * 97.75\% * 99\% * 99.99\% * 95\% = \underline{79.10\%}$$

Hemos mejorado en 19.23%

Pasaríamos de unas 3500 horas de no-disponibilidad al año a unas 1830 horas de no-disponibilidad al año.



## Alta disponibilidad

Si replicáramos cada elemento de red, servidores e ISP, dejando un solo centro de datos:

$$97.75\% * 99\% * 99.9999\% * 99.96\% * 97.75\% * 99.99\% * 99.99\% * 99.75\% = 94.3\%$$

Component	Availability
Web	97.75%
Application	99%
Database	99.9999%
DNS	99.96%
Firewall	97.75%
Switch	99.99%
Data Center	99.99%
ISP	99.75%

Mejorado en 34.43%

Pasaríamos de unas 3500 horas de no-disponibilidad al año a unas 500 horas de no-disponibilidad al año.

## Alta disponibilidad

Si generalizamos la última ecuación para cuando replicamos dos componentes:

$$A_s = AC_{n-1} + (1 - AC_{n-1}) * AC_n$$

Así, si hemos añadido un tercer servidor web:

$$\text{disponibilidad\_web3} = 97.75\% + (1 - 97.75\%) * 85\% = 99.6625\%$$

Y si añadimos un cuarto servidor web:

$$\text{disponibilidad\_web4} = 99.6625\% + (1 - 99.6625\%) * 85\% = 99.949\%$$

## Alta disponibilidad

### Ejercicio T2.1:

*Calcular la disponibilidad del sistema si tenemos dos réplicas de cada elemento (en total 3 elementos en cada subsistema).*

## Alta disponibilidad

¿Cómo se consigue mejorar la disponibilidad?

El uso de subsistemas redundantes y monitorizarlos mejora la disponibilidad del sistema global.

Surgen conceptos derivados:

- disponibilidad de red
- disponibilidad de servidor
- disponibilidad de aplicación

Si la disponibilidad de red es baja, quizás haya que mejorar el ancho de banda, y no tenga sentido centrar esfuerzos en mejorar las aplicaciones.

## Alta disponibilidad

### Disponibilidad de la red

El diseño debe tener redundancia a todos los niveles:

- Conexión a Internet
- Routers/cortafuegos/balanceadores
- Servidores

Si hay que recortar costes, algún elemento puede ser único:  
p.ej. el router, si el proveedor se compromete a reemplazarlo en pocas horas.

## Alta disponibilidad

### Disponibilidad del servidor

Casi cualquier elemento hardware del servidor puede fallar:

- CPU, memoria, discos, placas, etc.

Existen productos en el mercado con cualquier elemento duplicado.



<http://bit.ly/YGCfuE>

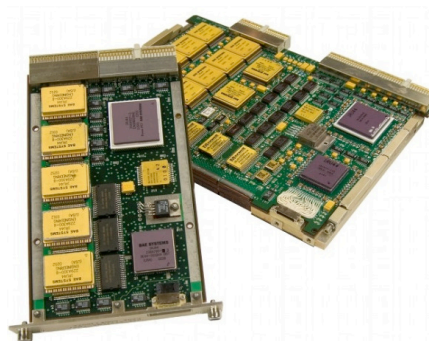
## Alta disponibilidad

### Disponibilidad del servidor (ejemplo)

Sistemas redundantes en el Curiosity (doble placa):

<http://bit.ly/15VxCBx>

“El día 27 de febrero de 2013 fue incapaz de guardar los datos adquiridos en la memoria flash [...] El fallo ocurrió mientras estaba usando la computadora A, por lo que desde control se ordenó pasar a usar el ordenador B para asegurar el correcto funcionamiento de los sistemas básicos“



## Alta disponibilidad

### Disponibilidad del servidor

Puesto que casi cualquier elemento hardware del servidor puede fallar...

Se pueden configurar los servidores con redundancia mediante el software.

Mejora de la escalabilidad.



## Alta disponibilidad

### Disponibilidad del servidor

Monitorizar la disponibilidad con las herramientas del SO.

El desarrollador ofrece herramientas para monitorizar hardware y software.

*Stratus servers are available only in the above hardware configurations for use with DeltaV software. The DeltaV version of the Fault-Tolerant Server is configured with a specific Stratus software revision level and setup that can be purchased only through Emerson Process Management. If other options are desired (such as additional memory) an ISR must be submitted to quote and purchase the server. Servers for use in a DeltaV system may not be purchased directly from Stratus.*

<http://bit.ly/YGCfuE>

## Alta disponibilidad

### Disponibilidad de las aplicaciones

Complicado medir las prestaciones de las aplicaciones.

El funcionamiento de unos módulos afectan al de otras aplicaciones (dependencias).

El desarrollador del SO suele facilitar herramientas para monitorizar también las aplicaciones en ejecución.

## Alta disponibilidad

### Disponibilidad de las aplicaciones

Dependencia entre aplicaciones o módulos de aplicación:

- Si un módulo falla, el proceso no se completa, y la experiencia del usuario es mala.

Desarrollar aplicaciones robustas -> hacerlas redundantes.

Si una parte del proceso falla, que haya una alternativa para completarlo.

## Alta disponibilidad

### Ejercicio T2.2:

*Buscar frameworks y librerías para diferentes lenguajes que permitan hacer aplicaciones altamente disponibles con relativa facilidad.*

*Como ejemplo, examina PM2*

*<https://github.com/Unitech/pm2>  
que sirve para administrar clústeres de NodeJS.*

## Índice



Introducción

Concepto de alta disponibilidad

[Concepto de escalabilidad]

Escalar un sitio web

Conclusiones

## Escalabilidad

Cuando una persona sufre estrés, su capacidad para afrontar tareas se ve mermada.

Cuando un sistema experimenta estrés, su capacidad para dar servicio también se ve afectada.



## Escalabilidad

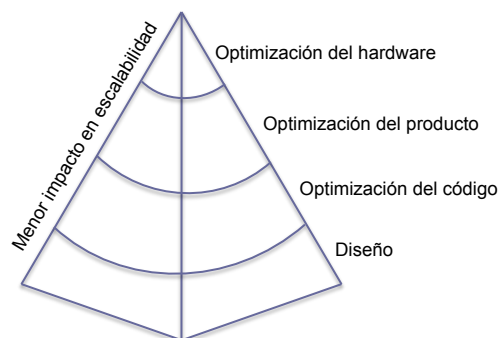
Incremento del nivel de estrés:

- Cambios en las aplicaciones
- Fallos o caídas de algunas partes del sistema
- Incremento del número de máquinas
- Incremento repentino del número de usuarios del sitio

La **escalabilidad** se refiere a la capacidad de un sistema de manejar la carga, y el esfuerzo para adaptarse al nuevo nivel de carga.

## Escalabilidad

**escalabilidad** = capacidad de un sistema de manejar la carga, y el esfuerzo para adaptarse al nuevo nivel de carga



<http://bit.ly/Z2II1G>



## Escalabilidad

Si un sitio gana popularidad, o si llega una fecha señalada, puede incrementarse su carga.

Para manejar esa carga, las empresas tienen más servidores de los necesarios normalmente.

Decidir cómo **añadir más recursos al sistema web** es crucial en el diseño inicial y en el mantenimiento.

En ocasiones, si la CPU del servidor está al 95% todo el tiempo, cambiándola puede ser suficiente para cierto nivel de carga. Pero si más adelante hay más carga, será insuficiente.

## Escalabilidad

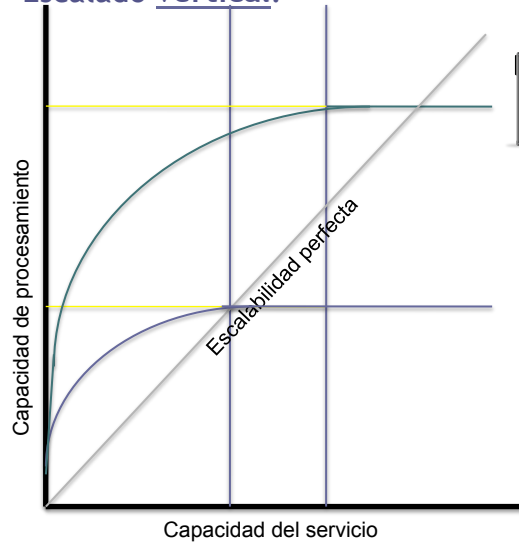
Vemos que hay dos tipos de escalado:

- Ampliación **vertical**:  
incrementar la RAM, CPU, disco de un servidor.
- Ampliación **horizontal**:  
añadir máquinas a algún subsistema (servidores web, servidores de datos, etc).

En ocasiones una ampliación vertical puede ser suficiente.

## Escalabilidad

### Escalado vertical:



Si ponemos una máquina más potente, podrá dar servicio hasta aquí, pero a partir de ese nivel ya no podrá atender más carga

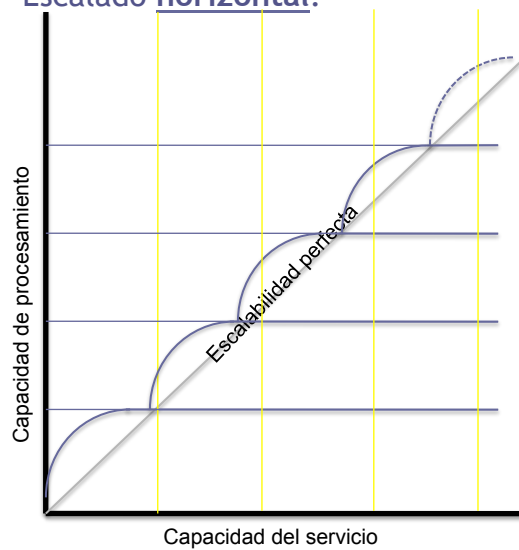


Ésta máquina sólo puede dar servicio hasta este punto.

<http://bit.ly/XfBRoR>

## Escalabilidad

### Escalado horizontal:



Cada máquina incrementa un poco la escalabilidad.

<http://bit.ly/XfBRoR>

## Escalabilidad

¿Cómo analizar la sobrecarga?

- Si la CPU está cerca del 100% todo el rato y el resto de subsistemas no está sobrecargado, sustituir por una CPU más potente.
- Si el uso de RAM es muy alto, veremos un uso alto de disco (por swapping). Incrementando la cantidad de RAM mejoraremos el rendimiento.
- Un ancho de banda insuficiente afectará al rendimiento. Contrataando una mejor conexión será suficiente.

## Escalabilidad

### **Ejercicio T2.3:**

*¿Cómo analizar el nivel de carga de cada uno de los subsistemas en el servidor?*

*Buscar herramientas y aprender a usarlas.*

*...¡o recordar cómo usarlas!*

## Escalabilidad

*Transformar el servidor web en una granja web*

- Proceso complejo.
- Preparar aplicaciones para distribuir la carga.
- Configurar la red para soportar tráfico creciente.
- Configuración del balanceo de carga para formar un cluster web para cada servicio.

Una granja web puede tener varios clusters web.

## Índice



Introducción

Concepto de alta disponibilidad

Concepto de escalabilidad

[ Escalar un sitio web ]

Conclusiones

## Escalar un sitio web

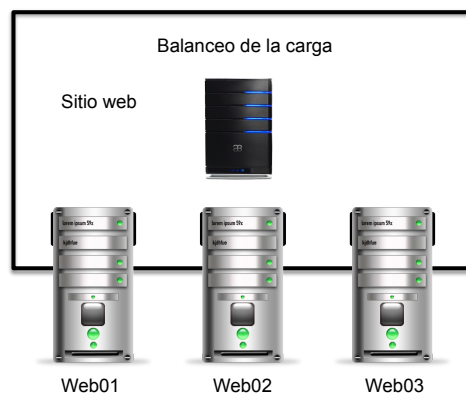
Tenemos que configurar tres niveles:

- máquinas como servidores web
- aplicaciones
- almacenamiento

## Escalar un sitio web

El nivel web se puede configurar *balanceando la carga*:

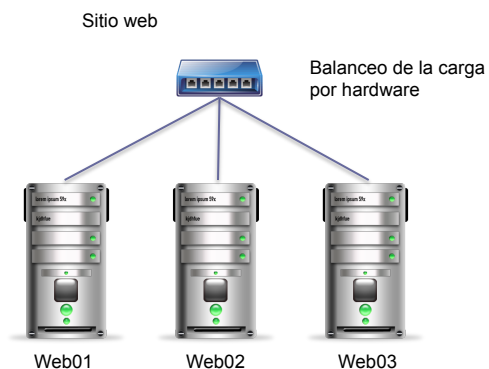
- uso de una máquina con software específico



## Escalar un sitio web

También se puede usar un *balanceador hardware*:

- Local Director (Cisco)
- ServerIron (Foundry)
- BigIP (F5)



## Escalar un sitio web

El balanceador pasa peticiones a los servidores según el tráfico de la red.

Hay varios algoritmos para decidir qué máquina final servirá cada petición:

- Por turnos (round-robin)
- Según el menor número de conexiones
- Por ponderación
- Por prioridad
- Según el tiempo de respuesta

## Escalar un sitio web

Escalar el *nivel de aplicaciones* requiere diseñar el software pensando en que se ejecute en varios servidores:

- Paralelismo
- Transparencia de ubicación: no debe haber dependencia de una máquina concreta para ejecutarse la aplicación.

Es importante diseñar las aplicaciones desde el principio para que se ejecuten en varios servidores.

Adaptar posteriormente una aplicación dependiente de cierto servidor puede ser costoso.

## Escalar un sitio web

Escalar el *nivel de almacenamiento* es complejo y depende del tipo de servicios a ofrecer:

- LDAP: Protocolo Ligero de Acceso a Directorios
- NFS: Sistema de archivos de red
- Bases de datos

Cada uno de estos mecanismos suele requerir mecanismos y configuraciones diferentes.

## Escalar un sitio web

### Ejercicio T2.4:

*Buscar ejemplos de balanceadores software y hardware (productos comerciales).*

*Buscar productos comerciales para servidores de aplicaciones.*

*Buscar productos comerciales para servidores de almacenamiento.*

## Índice



Introducción

Concepto de alta disponibilidad

Concepto de escalabilidad

Escalar un sitio web

[ Conclusiones ]



## Conclusiones

Conceptos clave: escalabilidad y alta disponibilidad.

**Monitorización** para detectar problemas y determinar posibles mejoras del sitio web.

La escalabilidad se suele implementar **replicando servidores** para las mismas tareas.

Conseguir disponibilidad y escalabilidad mediante **balanceo de carga**.