INTRODUCCIÓN

FIABILIDAD

Daniel Blanco Calviño

EL RENDIMIENTO Y LA ESCALABILIDAD NO LO SON TODO

- Hemos visto por qué nuestro sistema debe:
  - Rendir bien.
  - Comportarse bien cuando la carga aumente.

- Pero esto **no es suficiente**.
  - Si tenemos fallos e interrupciones en el servicio la experiencia de usuario se verá afectada.

EL RENDIMIENTO Y LA ESCALABILIDAD NO LO SON TODO

- Hemos visto por qué nuestro sistema debe:
  - Rendir bien.
  - Comportarse bien cuando la carga aumente.

- Pero esto **no es suficiente**.
  - Si tenemos fallos e interrupciones en el servicio la experiencia de usuario se verá afectada.

- **Debemos asegurar la fiabilidad.**

FIABILIDAD

- Probabilidad de que el software se ejecute durante un determinado período de tiempo sin que se produzcan fallos.

- **Uptime**.
  - Tiempo que nuestro sistema está operativo.

- **Downtime**.
  - Tiempo que nuestro sistema no está operativo.

- **Disponibilidad (%) = uptime / tiempo total**

FIABILIDAD EN SISTEMAS A GRAN ESCALA

- ¿Qué sucede si tenemos múltiples servicios?

S1

S2

S3

FIABILIDAD EN SISTEMAS A GRAN ESCALA

- ¿Qué sucede si tenemos múltiples servicios?

S1

S2

S3

FIABILIDAD EN SISTEMAS A GRAN ESCALA

- ¿Qué sucede si tenemos múltiples servicios?

- uptime = (n° servicios operativos / n° servicios totales) * tiempo.
  - No tiene en cuenta el **porcentaje de uso de cada uno.**

FIABILIDAD EN SISTEMAS A GRAN ESCALA

- ¿Qué sucede si tenemos múltiples servicios?

- uptime = (n° servicios operativos / n° servicios totales) * tiempo.
  - No tiene en cuenta el **porcentaje de uso de cada uno.**

- uptime = (n° peticiones exitosas (est.) / n° peticiones totales (est.)) * tiempo.
  - No tiene en cuenta la **relevancia de cada servicio.**

FIABILIDAD EN SISTEMAS A GRAN ESCALA

- ¿Qué sucede si tenemos múltiples servicios?

- uptime = (n° servicios operativos / n° servicios totales) * tiempo.
  - No tiene en cuenta el **porcentaje de uso de cada uno.**

- uptime = (n° peticiones exitosas (est.) / n° peticiones totales (est.)) * tiempo.
  - No tiene en cuenta la **relevancia de cada servicio.**

- **Ponderando cada servicio según su uso y relevancia.**

OTRAS MÉTRICAS

- **Mean Time Between Failures (MTBF)**
  - Tiempo medio entre fallos de nuestro sistema. Nos interesa un **MTBF alto.**

- **Mean Time To Recover (MTTR)**
  - Tiempo medio para recuperar el sistema de algún fallo. Nos interesa un **MTTR lo más bajo posible.**

ALTA DISPONIBILIDAD

- ¿Qué significa un sistema con alta disponibilidad?
  - **¿90%?**
  - **¿95%?**
  - **¿99%?**
  - **¿100%?**

- Nuestros interesados quieren el 100%, pero no es algo realizable.
  - Fallos **humanos**.
  - Fallos **hardware**.
  - Problemas en **proveedores externos.**
  - **Mantenimiento**.

Availability %	Downtime por año	Downtime por mes	Downtime por semana	Downtime al día
90% ("un nueve")	36.53 días	73.05 horas	16.80 horas	2.40 horas
95% ("un nueve cinco")	18.26 días	36.53 horas	8.40 horas	1.20 horas
99% ("dos nueves")	3.65 días	7.31 horas	1.68 horas	14.40 minutos
99.9% ("tres nueves")	8.77 horas	43.83 minutos	10.08 minutos	1.44 minutos
99.99% ("cuatro nueves")	52.60 minutos	4.38 minutos	1.01 minutos	8.64 segundos
99.999% ("cinco nueves")	5.26 minutos	26.30 segundos	6.05 segundos	864.00 milisegundos

Fuente: Wikipedia

TOLERANCIA A LOS FALLOS

- No importa lo mucho que nos esforcemos, **los fallos son inevitables.**

- Debemos buscar la **tolerancia a los fallos.**
  - **Prevención.**
  - **Detección.**
  - **Recuperación.**

PREVENCIÓN A LOS FALLOS

- **Punto único de fallo (Single Point Of Failure)**
  - Elemento que si falla hace que el sistema al complete quede inoperativo.
    - Servidor.
    - BBDD.
    - Servicio de terceros.

- Es importante **eliminar** los máximos puntos únicos de fallo posibles.
  - Se debe aplicar **redundancia**.
    - Múltiples servidores (escalado horizontal).
    - Múltiples BBDD.

DETECCIÓN DE LOS FALLOS

- Una detección rápida de los fallos es vital.
  - Aviso a posibles **interesados**.
  - Puesta en marcha de **estrategias de contención y recuperación.**

- Nuestro sistema **debe detectar fallos parciales** por si mismo.
  - Envío de peticiones a un servidor saludable en lugar de uno con problemas.

- **Monitorización**.
  - Automática.
  - Equipo dedicado.

RECUPERACIÓN DE LOS FALLOS

- Recuperar el sistema cuanto antes para que los usuarios sufran el menor impacto posible.
  - **Apagar** por completo **el sistema.**
  - **Rollback** a versión anterior.
  - **Copia de seguridad** de los datos.

- La recuperación debe ser lo más sencilla posible.

- Una vez recuperado, se debe **documentar**.
  - **Post Mortem** con las causas, los pasos para recuperar el sistema y las estrategias de prevención para que no vuelva a ocurrir en el futuro si es posible.