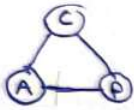


TEOREMA CAP



En un sistema distribuido NO podemos mantener

↓
SOLO 2 A LA VEZ

PA / PC / AC

CONSISTENCIA FUERTE (C)
DISPONIBILIDAD (A)
PARTICIONADO (P) → Tolerancia al particionado

AC → Sin Particiones: Garantizamos { Consistencia Fuerte, Disponibilidad }
↳ NO VIABLE :(

No admitimos que la red se particione
Habría que garantizar conectividad (replicación)
Difícil de asegurar → costoso para redes grandes

PC → Sin Disponibilidad: Garantizamos { Consistencia Fuerte, Particionado }
↳ ESCALA MAL :(

Al particionarse, el grupo con menos nodos deja de funcionar, hay usuarios sin atención para garantizar la consistencia.

TEMA 6

PA → Sin Consistencia: Garantizamos { Disponibilidad, Particionado }
↳ ESCALABLE → MEJOR OPCIÓN

Tras una partición → Siguen trabajando, podría haber divergencias que debían resolverse con una consistencia eventual.

REPLICACIÓN MULTI-MÁSTER

Los modelos de replicación Activa o Pasiva son FUERTEMENTE CONSISTENTES → Limita Escalabilidad
↳ Para mejorarlo aparece la Replicación Multi-Máster → Extensión del Modelo Pasivo

→ Cada 'cliente' tiene una réplica principal (Máster), cada copia se encarga de una actividad
Cada nodo es principal para algunas tareas y secundario para otras.

PEREZOSO → Los cambios NO se propagan inmediatamente, se acumulan y se propagan tras un número de cambios. Lo hace + rápido.

VENTAJAS → { Alta escalabilidad, Sobrecarga mínima, Operaciones NO deterministas }

INCONVENIENTES → { Si falla un master, todas sus peticiones pueden perderse → Inconsistencias, Gestión de fallos igual al modelo pasivo, NO soporta fallos bizantinos }

ALMACENES NoSQL

TIPOS

- o Almacenes clave-valor
 - { Esquema simple compuesto por dos campos: clave, valor
 - [La búsqueda se realiza por la clave.
- o Almacenes de documentos
 - { Esquema compuesto por objetos con no variable de atributos. (que pueden ser objetos)
 - [La búsqueda se puede realizar por atributos.
- o Almacenes de registros extensibles
 - { Esquema compuesto por tablas con no variable de columnas
 - [Los datos se separan por filas o columnas y estas en diferentes nodos.
 - [Sharding → para acceder a diferentes datos buscados en los nodos.

ELASTICIDAD

↑
escalable
+
adaptable

- Capacidad para adaptarse a variaciones de demanda → adaptarse a cambios de carga
- Asigna a cada aplicación la cantidad justa de recursos de manera autónoma
 - ↳ cubriendo la demanda existente con la mayor precisión posible
- REQUIERE
 - { ① Sistema de Monitorización → Supervisa la carga y rendimiento, identifica dónde se necesitan + o - recursos
 - { ② Sistema de Actuación → Automatiza la reconfiguración de recursos y servicios
- Ajusta la asignación de recursos en respuesta a los cambios de demanda.
- Pone en marcha el detector por el sistema de monitorización

CONTENCIÓN Y CUELLOS DE BOTELLA



Situación en la que un recurso se convierte en un cuello de botella para todo el sistema.

↳ Retiene todo el proceso

• Causas

→ Centralización:

Todas las tareas pasan por un nodo que realiza todas las tareas pesadas, ralentiza todo el sistema.

Solución: Evitar centralización de tareas pesadas.

→ Condiciones de:

carrera

Cuando compiten recursos

Solución: Usar herramientas de sincronización
Optar por asincronía

→ Tráfico

excesivo

Solución: Replicar recursos y mantener consistencia

Cambiar accesos remotos a locales para reducir latencia