

Tarea #1

(Pág 1 y Pág 2)

- ¿Cuáles son los modelos de programación que Meiklejohn reconoce? **Mapa 1.**
- ¿Cuáles han realizado ustedes?
- Ha verdad nunca he implementado otro mas que MVC (Model View Controller) Pero el artículo me sirve para expandir mis horizontes. !!
- ¿Tienen idea de cómo acercarse a los demás?
- Leyendo, programando y equivocándose, es el único camino!

Mapa 1.

En el artículo se nos presentan:

2 Modelos de Programación

** Ver Mapa 1.1

** y uno extra que es la combinación de los 2.

LASP

Se enfoca en la programación de libre sincronización de aplicaciones a larga escala distribuidas.

- Usa **CRDT** (Conflict Free Replicated Data Type)

Operaciones determinísticas combinadas

Comparte estado de réplicas

Todas las réplicas convergen en un solo resultado.

- CRDT se usa de manera secuencial.
- Las aplicaciones se pueden juntar para que se combinen resultados.

LASP sacrifica consistencia para mantener su disponibilidad.

Austere

Los estados replicados y compartidos son sincronizados para cada operación en el Sistema.

- **Austere** sacrifica su disponibilidad en orden de mantener su consistencia.

• Usa **2PL** (Dos fases de bloqueo) (2 phase locking)

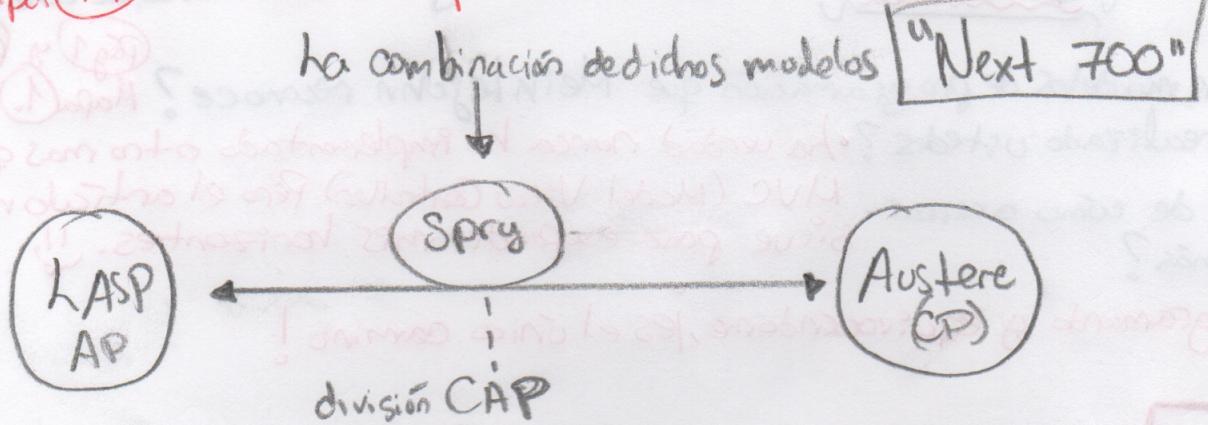
Para asegurar que los valores o datos son leídos sin discrepancias.

• Usa **2PC** (2 fases de compromiso) (2 phase commit)

Para publicar/comprometer todas las modificaciones.

Pág. 1

Mapa 1.1 *



- Spry hace un "intercambio" entre la disponibilidad y la consistencia basado en el nivel de semántica de las aplicaciones.

Segundo Punto

¿Qué dificultades nos trae la concurrencia?

- Cada nodo de ejecución se puede
 - Ejecutor en Paralelo (Varios procesadores)
 - Ejecutor de manera alterna (1 procesador)

Mapa Cuadro 2

■ La programación concurrente es difícil se debe tener cuidado en:

- Prevención no controlada de acceso a localidades de memoria.
- Exclusión mutua que es la adquisición de una localidad compartida de memoria

■ ¿Qué se desea para resolver el problema?

↳ Que el programador construya una
• Confluencia

↳ Se refiere a mantener el orden de un programa, y que no impacte en el resultado del mismo.

• ¿Que diferencias buenas o malas trae el cómputo distribuido sobre el concurrente?

Para empezar el cómputo distribuido es una forma particular de el cómputo concurrente.
* La extiende! de manera sustancial.

Sus diferencias

Contro 3

Cómputo Concurrente

- No determinista!
- La comunicación como se lleva a cabo en 1 sólo equipo es satisfactoria.
- Conjunto de acciones que pueden ser ejecutadas simultáneamente.
- Comparte una memoria común.

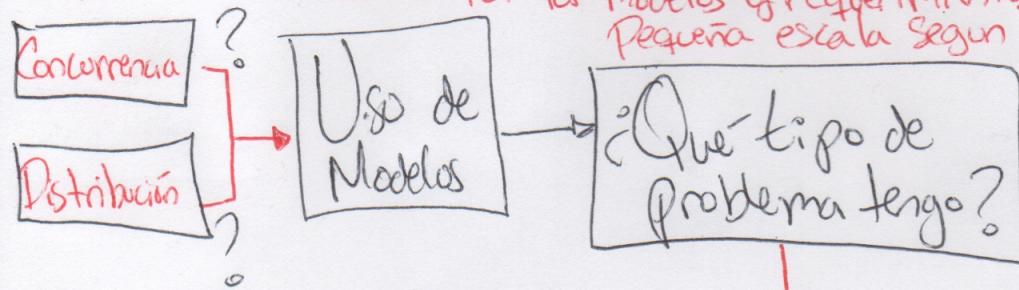
Cómputo Distribuido

- La comunicación puede ser No satisfactoria.
- Para ejecutarse en una Red de manera Paralela.
- No comparten una memoria común

¿Que estrategias principales describe el artículo para enfrentar a estos dificultades?

¿Porque se presentan como extremos en tensión?

Por los modelos y requerimientos. Si son de grande escala o Pequeña escala según el problema que tengamos enfrente



Pequeña escala

- Por lo general uso concurrente

Grande Escala

- Por lo general se soluciona mejor con sistemas distribuidos

Spry, LASP y Austere Pueden cohabitar en el mismo sistema que estamos diseñando.

Conjugando en uno sólo y podemos como Programadores CONTRARIA AL Pág. 3.