Bayou + weaker Consistency

"Strong"

Distribution
Replication

reliability

(but perf. too)

"Data" perspective

no longer true

"one copy"
reads
writes

prime reason: weaker models?

(performance, availability)

partition

C₁

R₁

R₂

C₂

R₃

no majority
One approach: (quorum-based)

\[ \implies \text{CAP theorem} \]

"Pick two"

\[ \text{Consistency} \]

Consistency

\[ \text{Availability} \]

 apparitions

byyou: early, important

\[ \implies \text{What is Ew/wc?} \]

\[ \implies \text{When can it be} \]

\[ \implies \text{Could weaken durability} \]

\[ \implies \text{Resolution of conflicts} \]

\[ \implies \text{Often automatic} \]

\[ \implies \text{Left to application} \]
Weaker consistency systems:
- Google Drive
- Dropbox

General:
- Version vectors
  - Vector clocks
  - File/object versions

Conflict resolution:
- Bayou approach
  - Dep. checks, merge proc.
granularity affects:
  \( \Rightarrow \) what conflicts are detected
  \( \Rightarrow \) amount of resolution

Dep Check: (app-specific constraint)

nice vs. "traditional" optimisitic?  
Specific \( \Leftrightarrow \) (generic) 
DB \( \Rightarrow \) exact state

Merge Procedure:
how hard is it to write this code?
(replication: make this harder)

\( \Rightarrow \) order
(determinism:)

  \( \downarrow \) how could this
anti-entropy: ordered
randomized
update

closed

primary: per dataset

convergence: from commit

use time (not commit)

why not? must see commit

versions

new alloc: must succeed