Conflicts : File mod. in both R & L & L.
Different : File mod. in either R or L.

After initial comparison to
common

After

File modification

File system modification

Set of files

No other modifications are being

Recognize - Run as a transaction - atomic

Two file acts - Left & Right - which need reconciliation.

Last reconcile time: time of which last reconciliation

Assume list of files with path names (Almost to Gp)

Recognition Procedure:
Receiv. Recom. Time

1. Vis. Que. (New) + Mod. Fr. Time (Left) +
   Last. Recom. Time

    < o. Left. Que. + Recom. Time

2. For each f in common-Left do

   2. Conflic. Que. ∈ φ

3. Left. Que. only → Subtract (Left. Que. common-Left)


5. Right. Que. ∈ Enqueue (Right. Que.)

6. Right. Que. (Left. Que. common-left)

3. Left. Que. ∈ Enqueue (Left. Que.)


1. Queue all activity on Left & Right

Procedure: Recom. Enqueue (Left, Right, Recom. Que.)
for each f in right-order

e else delcete f from left

for each f in left-order

e else copy f from left to right

if mod.f.time (left.f) > less.remmark.

for each f in left-order

Then terminate

elseif f modulus.time (left.f) ≠ modulus.time

elseif right.near then copy f to right

elseif left.near then copy f to right

if left.near and right.near then add f to conflict-free
Manually resolve (after t, right + t)

For each f in conflict line do

3. once side deleted
   1. one side delete a file
   2. one create a new file
   4. commit a right end file
- System time diff. + 7

Solution:

- Clock skew
- Clock synchronization
- Physical clocks
- Unsync
- 7
- 6:45
- Clock
- Other side
- Bed
- Crooked
- Mold in one side & sticking
- Newly erected flue
- Decent + help to resolve the issue
- Too expensive
- It's better to do 6.1-6.9 - 6.1 component
Ensuring protocol communication:

Which update comes first?

System + System

- Consensus problem - impossible to solve in parallel
- All replicas should receive in same order
- Ensure consistent updates and严格执行
- Replicated State Machines - method to file system implementation.

If the system implements a problem: many have to change

Generation number (logical time)

- Don't use physical time but use
Two-generals problem

- If A1 & A2 attack at same time then A will win o/w will lose to E.
- Messengers have to be sent to coordinate time of attack - can be captured / killed / bought by E.
- Is there a terminate protocol to compute time of attack?