Distributed Version Control System

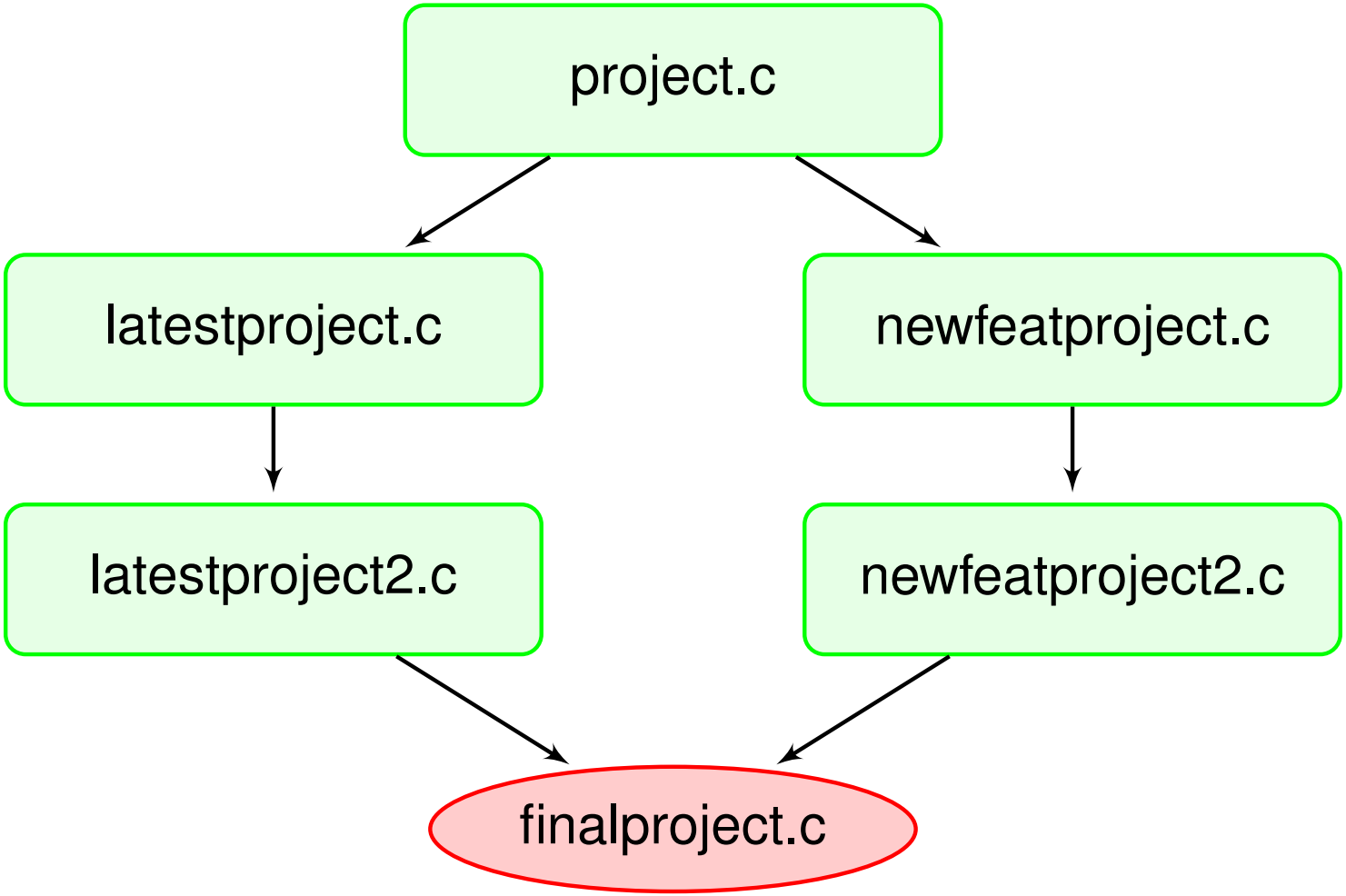
Rahul Ajmera
Satvik Chauhan

Department of Computer Science and Engineering
Indian Institute of Technology , Kanpur

November 11, 2012

Outline

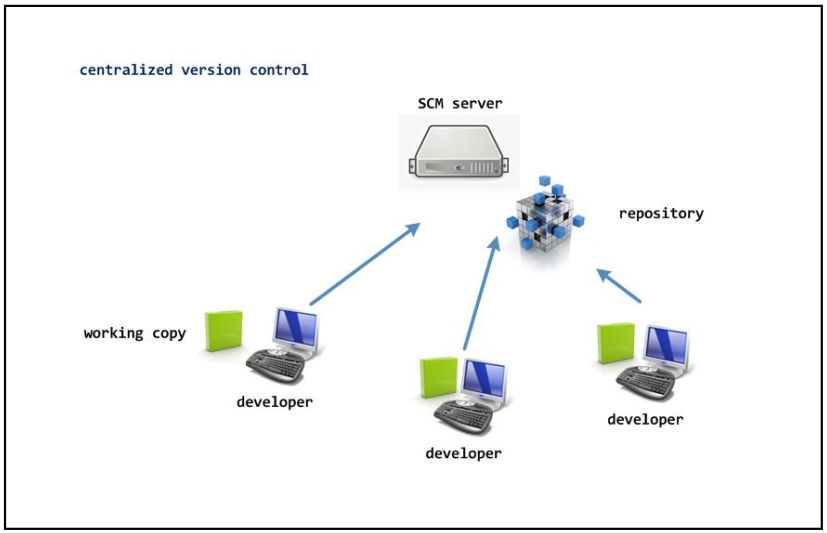
Directory history of a person not using Version Control

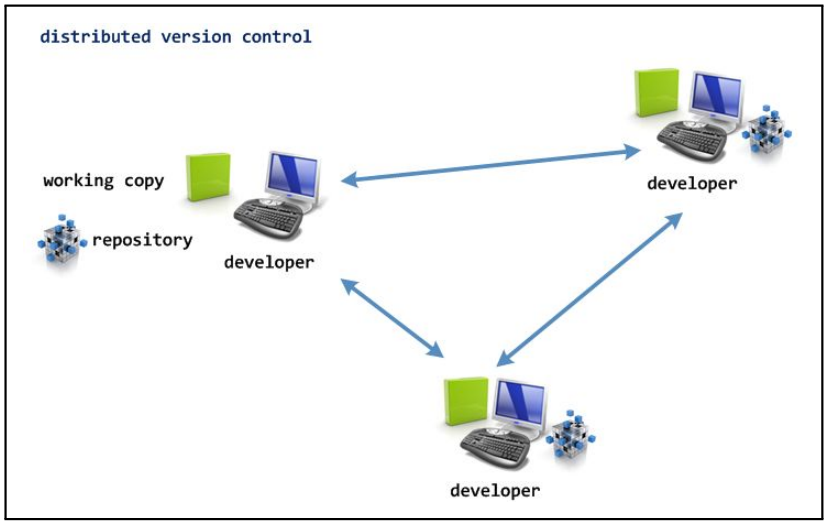


Motivation

- Keep track of changes you made.
- Revert changes to earlier versions.
- Share them with collaborators.
- Work on multiple features simultaneously.
- Merge two different versions of the files.

Centralized Vs Distributed

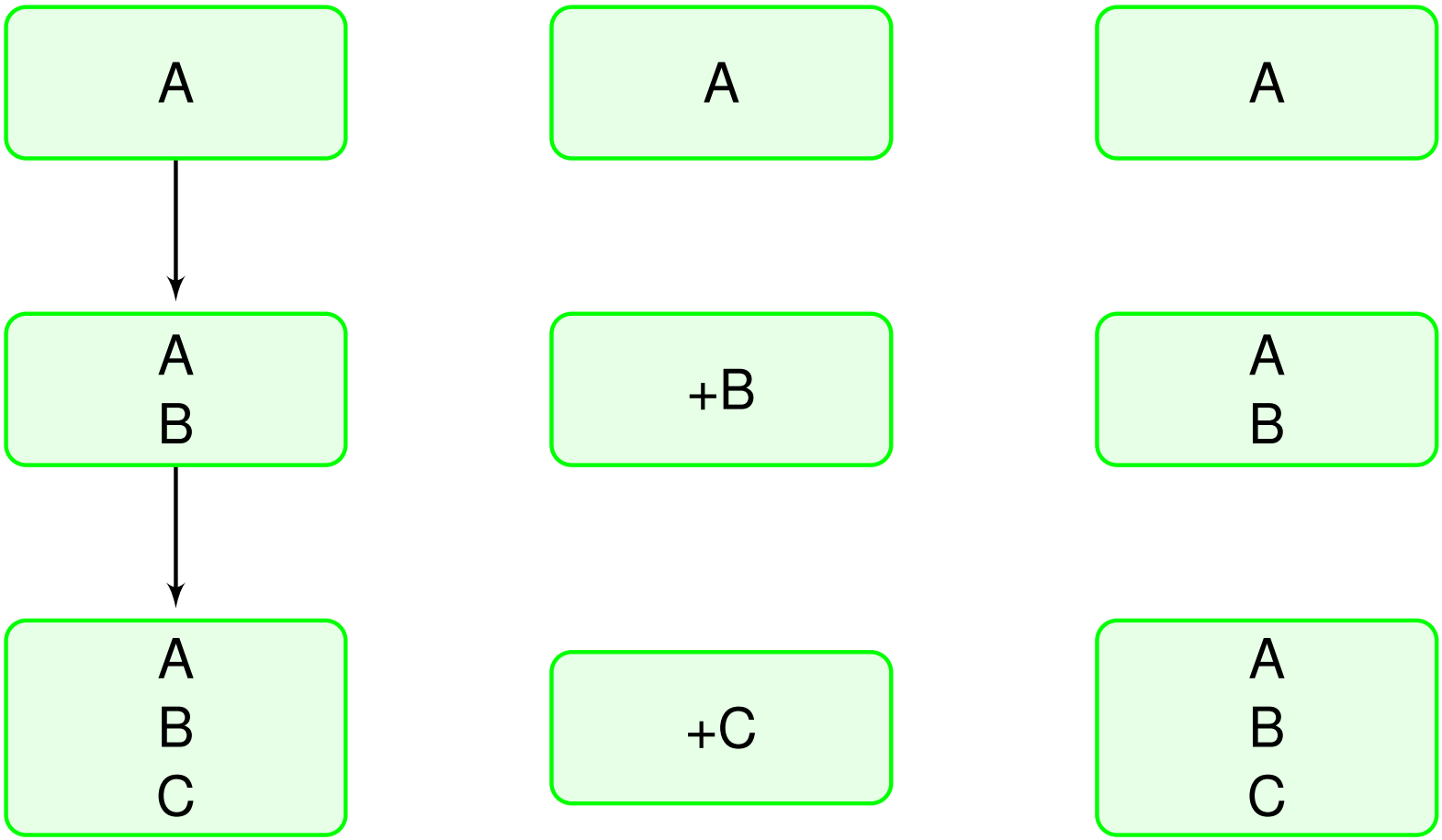




Changeset Choices

- Keep some snapshot versions and incremental diffs from these versions.
- Keep each commit as separate snapshot.
- Pros and Cons for each way.

Changeset Choices



- KISS principle.
- Modified II version + aggressive compression strategies.

Advantages

- It is seen that few files are modified with lots of changes between two commit.
- Use hash of files as their filename while taking snapshot.
- Unchanged files between two commits are not copied.
- Advantages
  1. Easy Rollbacks.
  2. Corruption of one file don't effect its other versions.
  3. Aggressive text compression Algorithms can be used to further compress the space.

Sending Commits

- Minimize number of communications between peers.
- Only 3 messages required at the time of pull.
  1. Commits to fetch.
  2. Commits compressed and sent.
- 3-Way merge startegy to merge the files.

Sending Commits

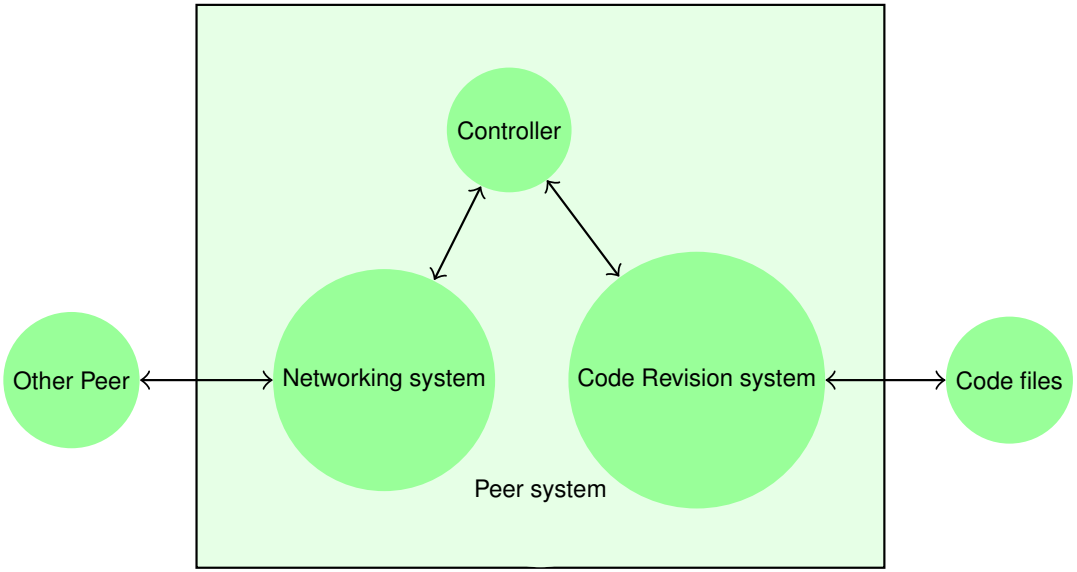


Figure : Highlevel design

- Command line UI.
- Graphical User Interface (Not very flexible).

Conclusion and Future work

- Covers basic features of distributed version control system.
- Enhancements
  1. Branch support.
  2. Automatic fetching of commits from added hosts.
  3. More flexible GUI and command line UI.