Git

(silly names is what we do best)

Source code control the way it was meant to be!

Linus Torvalds
torvalds@linux-foundation.org

Credits

(because Linus likes to make people think he's humble)

- CVS:
  - "WWCVSND":
    - What Would CVS Not Do?
- BitKeeper and Larry McVoy
  - Showing what distributed can be
- Junio Hamano & co
  - current git maintainer
    - ... and a lot of other helpers: there's about 280 distinct names in the author logs

Content

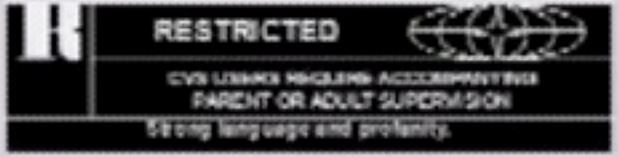
(bah. He wrote this in a hurry last night. Don't expect miracles)

- Implementation and design of a
  - Reliable
  - High-performance
  - Distributed
  - Content Manager
- ... and anything you bring up

Content Advisory

(you've been warned)

- Technical terms
  - SCM: Source Code Management
    - *part of* Software Configuration Management
  - CVS: Concurrent Versions System
    - Aka "the devil"
  - Wrong, Stupid: disagrees with Linus
    - See also: butt-ugly, "brain damaged"
- Strong opinions



Background

(aka "Didn't he have anything better to do?")

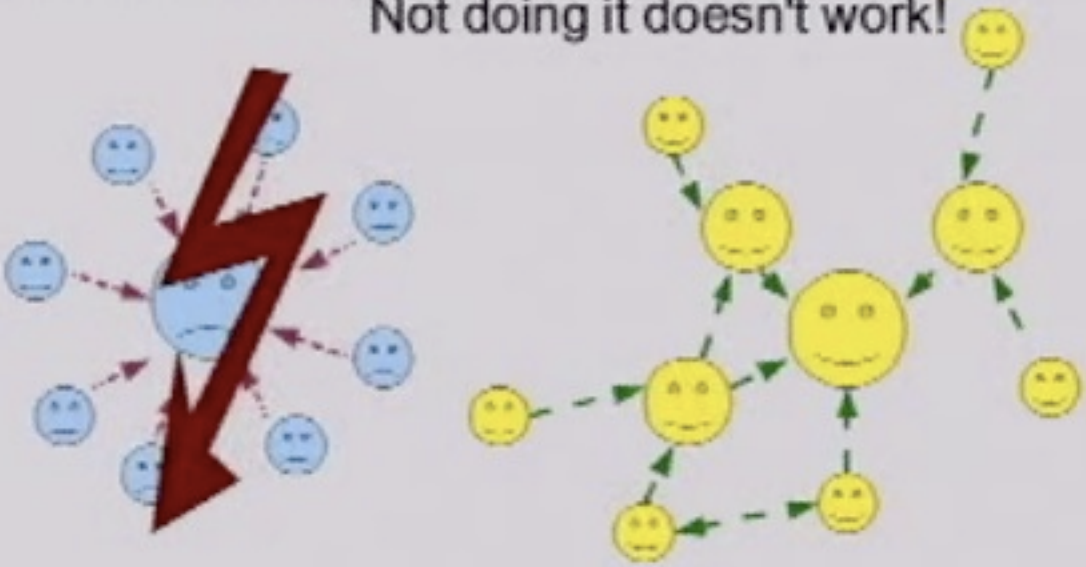
- I'm not an SCM person
  - Not heavily influenced by legacy designs
    - aka "CVS hasn't rotted my brain"
- Needed replacement for BitKeeper
  - Alternatives sucked
    - Centralized and/or absolutely horrid performance
- Some fundamental requirements
  - Performance and distributed nature
  - Security and trust issues

Distribution

(Look, ma, really cheesy graphics)

It's not just a good idea.

Not doing it doesn't work!



Distribution

(It's so much more than just off-line work)

- Collaboration
  - Commit changes without disturbing others
    - Branches that affect the main repository is a no-no!
  - Trust your data
    - ... without having to implicitly trust everybody else
      - Get away from the "commit access" mentality
        - No "special write access" class
        - No politics, and no granting/revoking of access
    - ... without having to implicitly trust your hosting
- Release engineering
  - Concurrent development/test/release cycles

Performance

("The need for speed")

- Performance is ***NOT*** secondary!
  - ... it affects how you work
  - ... and it affects quality
- I apply series of 200+ patches
  - ... must not be a "time to get a coffee" moment
- I do several merges *per day!*
  - Merging 22,000 files is something you should be doing all the time without even thinking about it. And it should take less than a second.

Reliable: trust and security

(does your data actually match what it's supposed to be?)

- Distributed systems mostly inherently safer
  - Not a single point of failure
  - Natural replication of data
    - ... and natural security boundaries!
      - Maintain the main repository behind three levels of firewalls
    - You can let people make changes
      - ... without actually allowing them direct write access!
- But you ***really*** want to trust your data
  - So you checksum everything with a strong hash...
  - ... and actually check it at every use.

Content Management

(It's so much more than just a random collection of files)

- It is ***not*** about "file revisions"
  - "Overview" vs "detail view"
    - Bad: "What happened to that file?"
    - Good: "What happened to [part of] the project?"
  - Single files largely uninteresting
    - Collections of files, however.

      `gitk v2.6.20.. drivers/scsi/ include/scsi/`
  - History must always be seen on a project basis
    - "Per-file" history is useless and wrong!
      - Content is about so much more than files that moved!