Mining from version control systems

Yueqi Li
10/1/09
Outline

- Introduction to source code management.
- Preprocessing CVS data
- Mining from Git
- Discussion
Introduction to source code management

- Source code management (SCM) is also known as version control.
- SCM is used to:
  - Support concurrent development
  - Keep history of changes
Some concepts in SCM

- **Repository**: The repository is where files' current and historical data are stored, often on a server.
- **Revision**: Also version: A version is any change in form.
- **Commit**: Writing changes to a repository.
- **Checkout**: Creating a working copy from repository.

Definitions are adapted from wiki pedia
Some concepts in SCM

- Branch: Another copy that can be developed concurrently.
- Trunk: The unique line of development that is not a branch.
- Merge: Applying two sets of changes to same sets of files.
Outline

- Introduction to source code management.
- Preprocessing CVS data
  - Centralized SCM
  - Drawbacks of CVS
  - Preprocessing CVS data
- Mining from Git
- Discussion
Outline

- Introduction to source code management.
- Preprocessing CVS data
  - Centralized SCM
  - Drawbacks of CVS
  - Preprocessing CVS data
- Mining from Git
- Discussion
Centralized model

- The repository is on the server.
- Clients fetch copies from server and commit changes to server.
- Typical programs: CVS, SVN, Google Docs.
Work flow for CSCM

Some Initial Contents

Main()
{
    a();
    b();
}

Server

User A

User B
Work flow for CSCM

Main()
{
    a();
    b();
}

Server

User A

Main()
{
    a();
    b();
}

User B

b()
{
    ... 
}

Main()
{
    a();
    b();
}
Work flow for CSCM

Main()
{
    a();
    b();
}

Server

Main()
{
    a();
    b();
}

User A

Main()
{
    a();
    b();
}

User B

b()
{
    ...
}

Main()
{
    a();
    b();
}
Work flow for CSCM

Server

Main()
{ 
a();
b();
}

User A

Main()
{ 
a();
b();
}

User B

a()
{
...
}

Main()
{ 
a();
b();
}
Outline

• Introduction to source code management.
• Preprocessing CVS data
  – Centralized SCM
  – Drawbacks of CVS
  – Preprocessing CVS data
• Mining from Git
• Discussion
Drawbacks of CVS

• Some relevant drawbacks:
  – Transaction information is lost.
    • You can not tell the difference between
      – Committing a file and b file
      – Committing a file, then committing b file
    – Commit information before merge is lost.
  – Slow.
  – No distributed development support.
  – ...
Outline

- Introduction to source code management.
- Preprocessing CVS data
  - Centralized SCM
  - Drawbacks of CVS
  - Preprocessing CVS data
- Mining from Git
- Discussion
Preprocessing CVS data

- Most issues discussed in *Preprossing CVS Data for Fine-Grained Analysis @ MSR 04* are dealing with information loss.
- There are a few steps to preprocess CVS data before performing analysis on the data.
Step 1: Data extraction

- CVS is slow.
- To enable fast access, we need to extract data from CVS and store it in some database.
Step 2: Restoring transactions

- Fixed time windows
- Sliding time windows
  - Author
  - Log message
  - Time interval between each commission.
Step 3: Mapping changes to entities

- Mapping raw byte or text changes to syntactical changes.
  - Syntactical changes can be changes to functions.
Step 4: Data cleaning

- Large Transactions
  - Infrastructure change but not logical change can be considered as noise.

- Merge Transactions

This figure is copied from the paper.
Outline

- Introduction to source code management.
- Preprocessing CVS data
- Mining from Git
  - Distributed SCM
  - Demo
  - What Git brings to us
  - Why DCSM gains popularity
- Discussion
Distributed SCM

- No central server
- Each pear has a complete repository.
- Pears can merge their result by pushing data to or pulling data from each other.
A small example

Initially, Leo has one repository with functions A and B. A new developer Rubin joins. All change history in both Leo and Rubin are stored in Leo's repository now.
Outline

- Introduction to source code management.
- Preprocessing CVS data
- Mining from Git
  - Distributed SCM
  - Demo
  - What Git brings to us
  - Why DCSM gains popularity
- Discussion
Demo 1

- Initializing a repository
- Adding files and committing
- Reverting to a previous revision.
Outline

- Introduction to source code management.
- Preprocessing CVS data
- Mining from Git
  - Distributed SCM
  - Demo
  - What Git brings to us
  - Why DCSM gains popularity
- Discussion
What Git brings to us

- Benefits and limitations mentioned in *The Promises and Perils of Mining Git @ MSR 09*.
- We only illustrate some points mentioned in the paper.
Promise 1

- Recovering more history information from Git
  - Richer project history through commit, branch and merge from multiple repositories.

- Demo 2: branching and merging
Promise 2

- Explicit recording authorship information for all contributors.
- Demo 3: viewing authorship information in log.
Promise 3

- Time and space efficiency.
  - Entire Eclipse repository
    - CVS: 7-25 minutes per commit
    - Git: 1-2 minutes per commit
  - Entire Mozilla repository
    - SVN: 12G
    - GIT: 420 MB.
Peril 1

- It is not always possible to determine what branch a commit was made on.
  - Commit does not record the branch on which it was created.
  - Demo 4: showing branch info after merge.
Peril 2

- Implicit Branches
  - Developing in different machines creates implicit branches
  - Demo 5: merging by pull from different repositories.
Outline

- Introduction to source code management.
- Preprocessing CVS data
- Mining from Git
  - Distributed SCM
  - Demo
  - What Git brings to us
  - Why DCSM gains popularity
- Discussion
Why DCSM gains popularity

- The wild adoption of distributed development with growing use of the Internet.
- Development of open source projects.
  - Most high profile open source projects have developers from around the world.
  - People like to keep a local change history before committing to official repositories.
  - People like to keep history of each branches.
Growth Of Open Source Projects

The Total Growth of Open Source@Proceedings of the Fourth Conference on Open Source Systems 08
Outline

- Introduction to source code management.
- Preprocessing CVS data
- Mining from Git
- Discussion
Discussion

- Current recommending systems, predicting systems are for centralized model.
- We may need to refine current model for distributed development model so that people have incentive to cooperate the share information.
Thank you.