

UNIVERSITYof **HOUSTON**
CENTER FOR ADVANCED COMPUTING & DATA SCIENCE

Source/Version Control and Checkpointing

Amit Amritkar

Vistas in Advanced Computing

Outline

- Version Control Management
- Basic Git Usage
- Checkpointing/restarts

See:

<http://git-scm.com>

<https://bitbucket.com>

Vistas in Advanced Computing

University of HOUSTON | CACDS

What is Version control

From Oxford Dictionary

- The task of keeping a software system consisting of many versions and configurations well organized.
- also known as **revision control** or **source control**

Why use Version control?

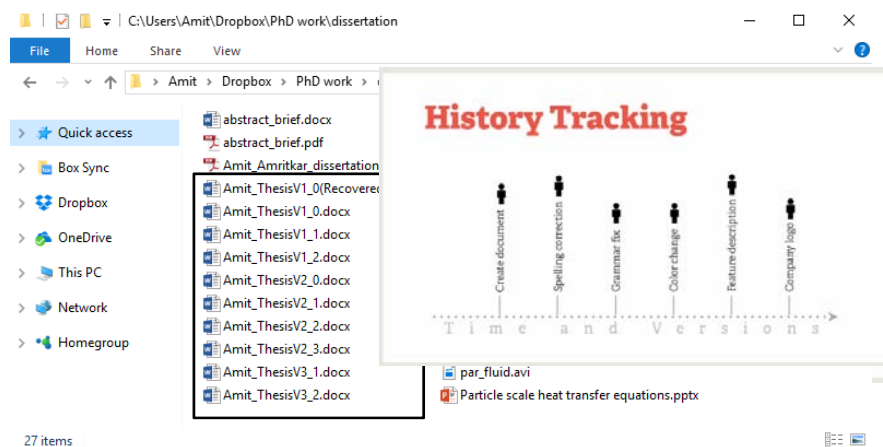
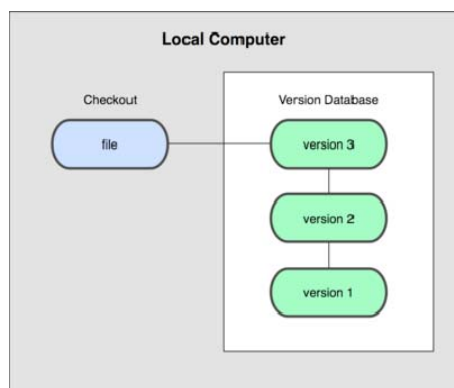


Image credit: <https://git-scm.com/video/what-is-version-control>

Why use Version control?

- **Reproducibility** - know what code was run when
- **Traceability** - know when things were added
- **Collaboration** - allow contributions without risking code breakage
- **Organization** - enforces a method of organization

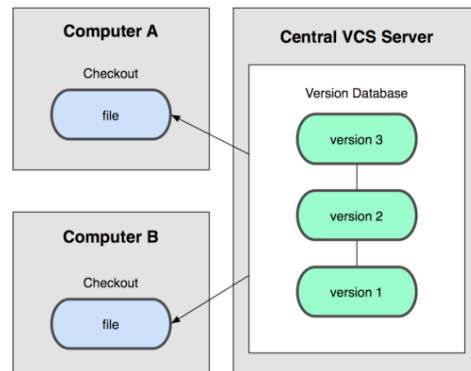
Local Source Control Management



[Image credit: <http://git-scm.com>]*

- 'Database' keeps versions of the file that can be "checked out"
- Edit and revise local files
- Use smart tools to see differences in the files

Centralized Source Control Management



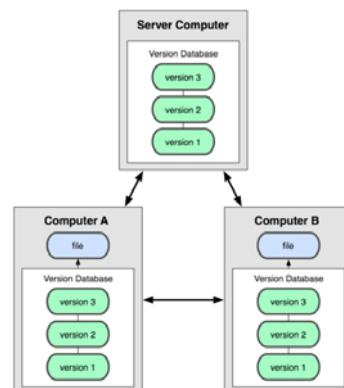
Examples

- CVS (1989)
- SVN (2000)
- ClearCase
- Perforce

[Image credit: <http://git-scm.com>]*

Vistas in Advanced Computing * Creative Commons Attribution 3.0 Unported License

Distributed Source Control Management



Examples

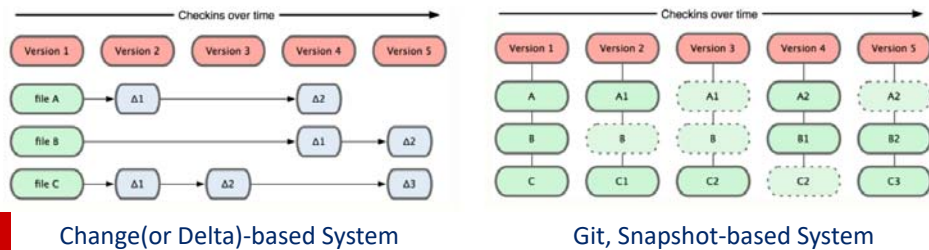
- Bitkeeper (2000)
- Darcs (2003)
- **Git (2005)**
- Bazaar (2005)
- Mercurial (2005)
- Often called a DVCS: Distributed Version Control System

[Image credit: <http://git-scm.com>]*

Vistas in Advanced Computing * Creative Commons Attribution 3.0 Unported License

Git is Different

- A commit creates a version for your file changes (common to most software configuration management (SCMs))
- A commit of your project or file is a snapshot at that moment which has a reference to it. (There is a separate copy, not just a "delta".)



[Image credit: <http://git-scm.com>]*

Vistas in Advanced Computing * Creative Commons Attribution 3.0 Unported License

Outline

- Version Control Management
- Basic Git Usage
- Checkpointing/restarts

Vistas in Advanced Computing

Git

- Available on windows, mac and linux
 - <https://git-scm.com/downloads/guis>
- Command line and GUI interface

Vistas in Advanced Computing

Local Commands

git help gives list of commands. You can use **man pages**, too
git help <command> gives details of a command

```
$ git help
usage: git ...
$ git help init
GIT-INIT(1)
...
```

Git Manual

Vistas in Advanced Computing

Local Commands

git init: Create an empty git repository or reinitialize and existing one

```
$ mkdir git_test  
$ cd git_test  
$ git init  
Initialized empty Git repository in  
/home1/01392/amit/git_test/.git/
```

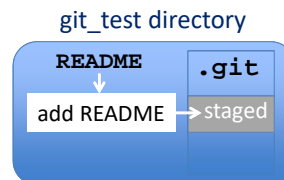
.git is your local repository.

Vistas in Advanced Computing

Local Commands

git add: Add file contents to the index (of files) and stages present copy for commitment.

```
$ echo "Hello Git World" >> README  
$ git add README
```



Vistas in Advanced Computing

Local Commands

git status: Show the working tree status

```
$ git status
# On branch master
#
# Initial commit
# 1
# Changes to be committed:
# (use "git rm --cached <file>..." to unstage)
# 2
# new file:   README
```

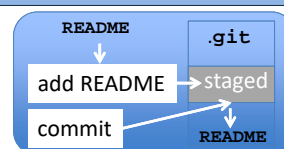
Shows no commitments (1) and a staged file (2) .

Vistas in Advanced Computing

Local Commands

git commit: Record changes to the repository

```
$ git commit -m "Adding README"
[master (root-commit) 774c810] Adding README
1 file changed, 1 insertion(+)
create mode 100644 README
```

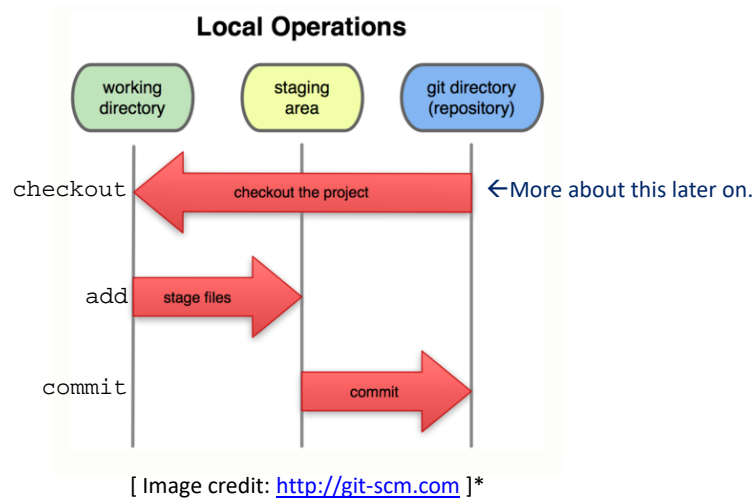


May get message to set your user name and email– so that it knows details of the author.

```
$ git config --global user.name "Your Name"
```

Vistas in Advanced Computing

Add and Commit



Vistas in Advanced Computing * Creative Commons Attribution 3.0 Unported License

Local Commands

git log: Show the commit logs

```
$ git log
commit 774c81087d052e43a630db7f676cfd9a6b006772
Author: Amit Amritkar <aramritk@central.uh.edu>
Date: Tue Jul 24 17:53:04 2012 -0500

1 Adding README
```

Note comment from **commit -m** option (1 Adding README).
Make your comments (history) meaningful.

Vistas in Advanced Computing

Local Commands

```
$ echo "Line 2" >> README
$ git add README
$ git commit -m "Adding Line 2"
$ echo "Line 3" >> README
$ git add README
$ git commit -m "Adding Line 3"
$ echo "Clear file" > README
$ git commit -am "Clear file"
```

> deletes previous contents of README

add & commit combined, all modified and indexed files

```
$ git commit -m "Clear file" README #add/commit a file
$ git commit -p -m "Clear file" # query add/commit files
```

alternate forms

Vistas in Advanced Computing

Local Commands

```
$ git log
commit c0513dbf6b609715f1510c438b9d00f065f7f3f4
Author: Amit Amritkar <aramritk@central.uh.edu>
Date: Tue Jul 24 17:58:53 2012 -0500
    Clear file
commit 88d4a87be3e7444d06463108e98ca78802f4859e
Author: Amit Amritkar <aramritk@central.uh.edu>
Date: Tue Jul 24 17:58:25 2012 -0500
    Adding Line 3
commit 43a446bedd92946d0ccf6fa2218f623284695f8b
Author: Amit Amritkar <aramritk@central.uh.edu>
Date: Tue Jul 24 17:58:01 2012 -0500
    Adding Line 2
commit 774c81087d052e43a630db7f676cfd9a6b006772
Author: Amit Amritkar <aramritk@central.uh.edu>
Date: Tue Jul 24 17:53:04 2012 -0500
    Adding README
```

ID

Author

Date

Comments

Note checksum (blobs or "id") for next slide.

```
$ git log README #can view log for individual files.
```

Vistas in Advanced Computing

Local Commands

git diff: Show changes between commits, commit and working tree, etc.

```
$ git diff README #--- staged +++ modified
diff --git a/README b/README
index d5c15a2..fcb6062 100644
--- a/README
+++ b/README
@@ -1,3 +1 @@
-Hello Git World
-Line 2
-Line 3
+Clear file
```

Vistas in Advanced Computing

Local Commands

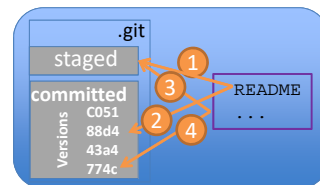
Types of differences: **Comparison**

`$ git diff README #staged file with modified README` 1

`$ git diff 88d4 README #ver 88d4... file with modified README` 2

- `$ git diff #staged files with modified files` 3
- `$ git diff 774c #ver 88d4... files with modified files` 4

If there are no staged files,
diff occurs on latest version.



Vistas in Advanced Computing

Local Commands

git checkout: Checkout a branch or paths to the working tree

```
$ git checkout 88d4a87be3e7
Note: checking out '88d4a87be3e7'.
```

Reverts (files) to snapshot 88..

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -b with the checkout command again. Example:

```
git checkout -b new_branch_name
```

More on branches (-b)
<http://bit.ly/1YLHDLA>.

```
HEAD is now at 88d4a87... Adding Line 3
```

Revert back to master snapshot (path)

```
$ git checkout master
Previous HEAD position was 88d4a87... Adding Line 3
Switched to branch 'master'
```

These are the comments!

Vistas in Advanced Computing

Remote Commands

git clone: Clone a repository into a new directory

git pull: Fetch from and merge with another repository or a local branch

git push: Update remote refs along with associated objects

Vistas in Advanced Computing

Remote Commands

Bitbucket Repository:

- Supports git and other protocols
- After creating empty repository:
 - Import at bitbucket or push files from local system
- For convenience name local directory of repository and remote repository the same name

```
$ #@bitbucket create repository SC
$ mkdir SC; cd SC #create local repo
$ date > README
$ git commit -am "new README"
```

Vistas in Advanced Computing

Push to a Server Repository

```
$ git remote add origin \
  ssh://git@bitbucket.org/amritkar/SC.git
  <protocol>://<site>/<user>/<repo_name>.<repo_type>
$ git push -u origin -all
  First time: push ALL up to site, declare local as upstream
$ echo '// No line return' >>p.c
$ git commit -am '2nd commit'
$ git push origin master
  Subsequent pushes: from local master to origin.
```

Vistas in Advanced Computing

Summary of Useful Commands

`git status` - Show the working tree status

`git log` - Show commit logs

`.gitignore` - Include `*.o *.a` `.gitignore` (1 line each)

`git diff` - Shows changes between commits, commit, and working tree, etc.

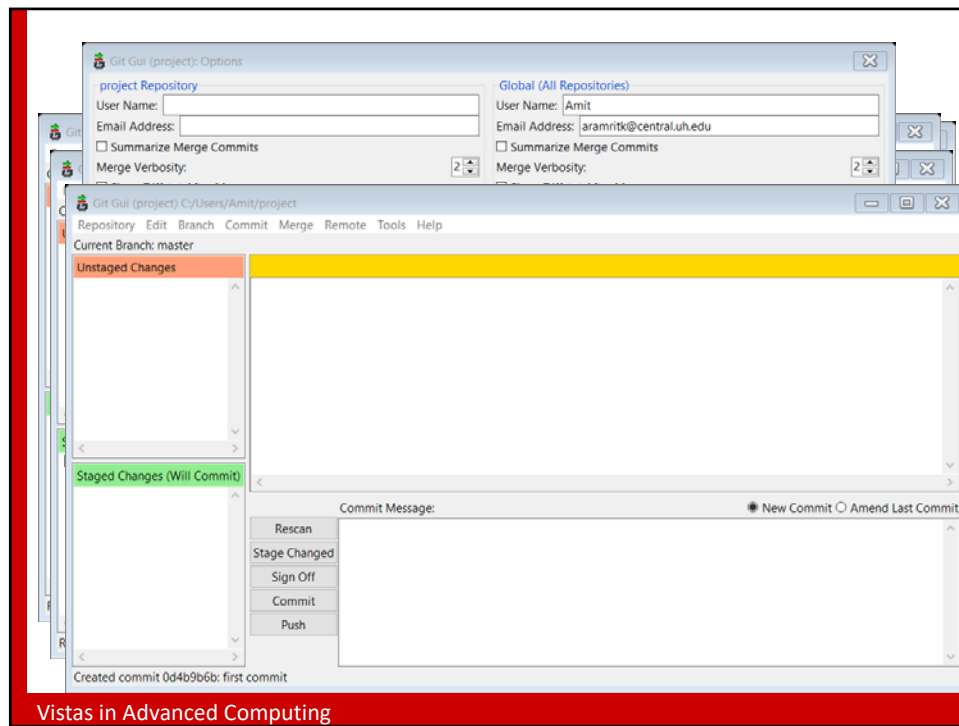
`git remote add <rem_nam> <sit>` adds a remote branch

Vistas in Advanced Computing

Tips and Tricks

- You must add a new file, and then commit it. `git commit -a` will not work for a new file.
- www.gitguys.com/topics

Vistas in Advanced Computing



Vistas in Advanced Computing

Outline

- Version Control Management
- Basic Git Usage
- Checkpointing/restarts

Vistas in Advanced Computing

University of HOUSTON | CACDS

Checkpointing/Restarts

- A technique to add fault tolerance into computing systems.
- Steps to follow,
 - Periodically save a snapshot of a running application's state
 - In case of failure use the last saved snapshot to restart the calculations
- Important for long running applications

MATLAB Checkpointing

Data files

- **save** *filename* *var1* *var2* ...
 - >> save myfile.mat x y → binary
 - >> save myfile.dat x y -ascii → ascii
- **load** *filename*
 - >> load myfile.mat → binary
 - >> load myfile.dat -ascii → ascii

```
if (mod(step,1000)==0)
save restart.mat var1 var2 var3;
end
```


References

"*Science and Technical Computing* course materials by The Texas Advanced Computing Center, 2014. Available under a Creative Commons Attribution Non-Commercial 3.0 Unported License"

- https://en.wikipedia.org/wiki/Application_checkpointing