Git as a Back Up Tool

Yes, No, Maybe?

The Pros

- Git is well understood (by me)
  - commands, .gitignore, iliminates *.bak files
- Git is fast (both for back up and recovery)
- Git archives are compact (when gc'ed)
- Git handles binary files
- Git provides secure synchronization with remote repositories
- Git backups do not require corporate support

The Cons

- Data is backed up twice (locally & remotely)
- Git does not track empty directories
  - file paths are tracked as attributes of content
- Git does not track file system metadata
  - it is considered harmful to git-based development
  - other than the execute bit
- Git does not track other Git projects
  - it will skip a directory with a .git subdirectory
  - unless the subdirectory is recorded as a submodule
    - and submodules are not independent of the main project!

The Maybes

- Can capture file system metadata
  - using a Git *pre-commit* hook
- Can restore file system metadata
  - using a Git *post-checkout* hook
- Other Git projects in the directory tree
  - might already be backed up (e.g. on github)
  - could be backed up to another directory
    - Using Git (an addition to the *pre-commit* hook)
    - Using another tool (such as rsync)

etckeeper

- Pros
  - Integrated with *apt*
  - Uses Git hooks
  - Easy to install (its a package in Debian, Ubuntu & Fedora)

- Cons
  - */etc* only
  - *metastore* app doesn't obey .gitignore
  - Does not handle Git projects
    - not expected in /etc

gibak

- Pros
  - Personal back up system for *$HOME*
  - Uses Git hooks
  - Handles Git projects
  - Available on github
  - Can checkout files based on time-based criterion

- Cons
  - Not easy to install
    - Requires *ocaml* (for *ometastore*)
  - Clumsy handling of Git projects
    - rsync'ed into *.git/*
    - not pushed
  - No tests
  - I can't type it!

gibak commands

```
 $gibak
   usage: gibak <action>

   <action> can be:
       help
       init
       commit
       eat <file_or_dir>
       show <file> as of
<time_spec>
       ls-changed-files
       ls-new-files
       ls-ignored-files
       ls-newly-ignored-files
       ls-stored-files [as of
<time_spec>]
       archive-to <file>
       extract-archive-to
<dir>
       rm-all
       rm-older-than
<time_spec>

   <time_spec> examples:
       5 days ago
       2 days 2 hours 3
seconds ago
       1979-02-26 18:30:00
```

Gib

- Pros
  - Fork of gibak
  - Can store the Git repository on a USB stick
  - Bash script rewritten in Python
- Cons
  - Mostly the same as gibak, including depency on *ocaml*
  - I have not used it

slug

- Pros
  - Fork of gibak
  - Backing up to a remote site is easy
  - Handles Git projects nicely
    - uses Git tool set
    - only if needed
  - Ocaml not required
    - using perl instead
  - On github

- Cons
  - Only partially implemented
  - No tests yet
  - *ocaml* dependency not yet removed
  - Name doesn't capture its positive qualities
    - Inspired by Git
  - I'm the author

Slug command

```
henry$ slug
usage:  <action>

<action> can be:
   help
   init
   it                              # to make a snaphot
   away /absolute/path             # copies the repository to a new local 'bare' repository
   away user@host:path             # copies the repository to a new remote 'bare' repository
   away                            # updates the previously copied <away-repo>
   clone <away-repo> <new-dir> # clones your home directory using an 'away' repository
   fix-hooks                       # must be run from parent directory of a slug repository
   eat <file_or_dir>
   show <file> as of <time_spec>
   ls-changed-files
   ls-new-files
   ls-ignored-files
   ls-newly-ignored-files
   ls-stored-files [as of <time_spec>]
   archive-to <file>
   extract-archive-to <dir>
   rm-all
   rm-older-than <time_spec>

<time_spec> examples:
   5 days ago
   2 days 2 hours 3 seconds ago
   1979-02-26 18:30:00
```

Statistics

- My mail directory
  - 1.7 GB
  - 40,357 files
  - 200-400 file operations a day (new, mv, delete)
  - 30 secs total
    - 12.5 secs for archiving
    - 17.0 secs for data transfer (over a 6 Mbs ADSL line)
- Faster than *cp -a*

Conclusion

- etckeeper could be used in production
  - http://joey.kitenet.net/code/etckeeper/
  - sudo apt-get install etckeeper
- giback not recommeded if you have Git projects
  - http://eigenclass.org/hiki/gibak-0.3.0
  - http://github.com/ChrisJohnsen/gibak
- gib not recommended if you have Git projects
  - http://github.com/mhl/gib
- slug marginally better than gibak, but wait for it