

Modern Version Control: Mercurial Internals

Dirkjan Ochtman (@djco)
PyCon 2010

This is your host

- Big fan of Python
- Formerly a happy user of Subversion
- Mercurial crew member since 2008

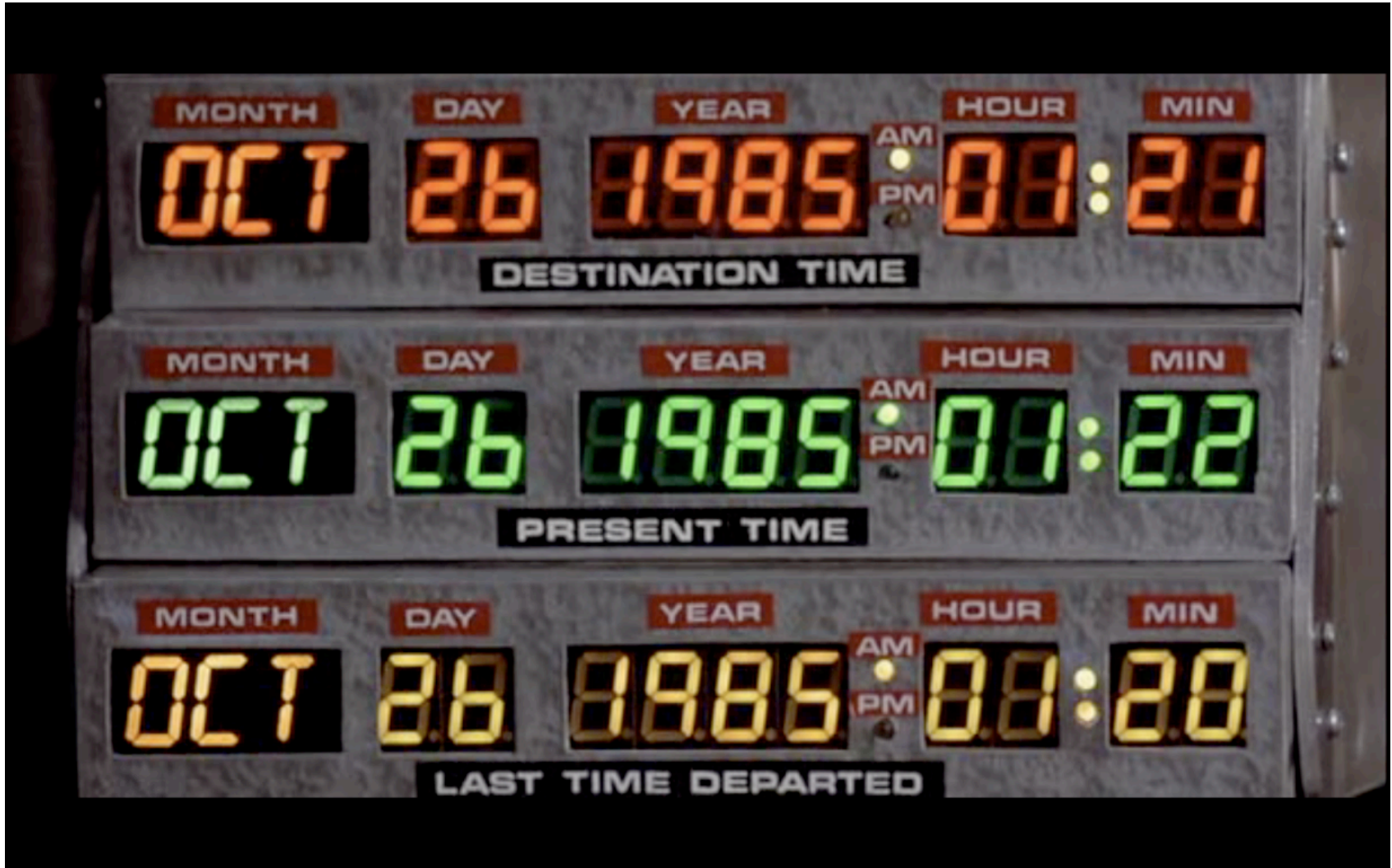
Where I'm going with this

- Problems & solutions
- How Mercurial does it
- Hooks and extensions
- Questions, hopefully with answers

Problems & Solutions

Problems: being distributed

- For a networked world
- No authority
- Bandwidth vs. latency
- Commit times are local, unsorted

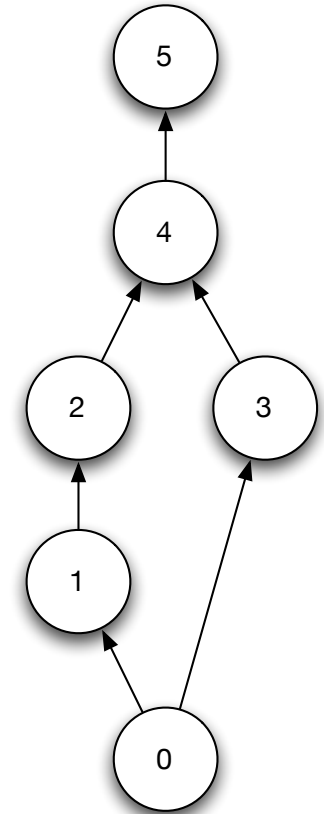


Problem: performance

- Slow tools discourage use
- Life in an I/O bound world
- Optimize for disk access / bandwidth
- $O(\text{revs})$, $O(\text{files})$, $O(\text{heads})$

Solutions: the DAG

- Directed Acyclic Graphs
- darcs' Patch Theory, optimized
- Heads: nodes without out-bound edges



Solutions in Mercurial

Revlogs (NG!)

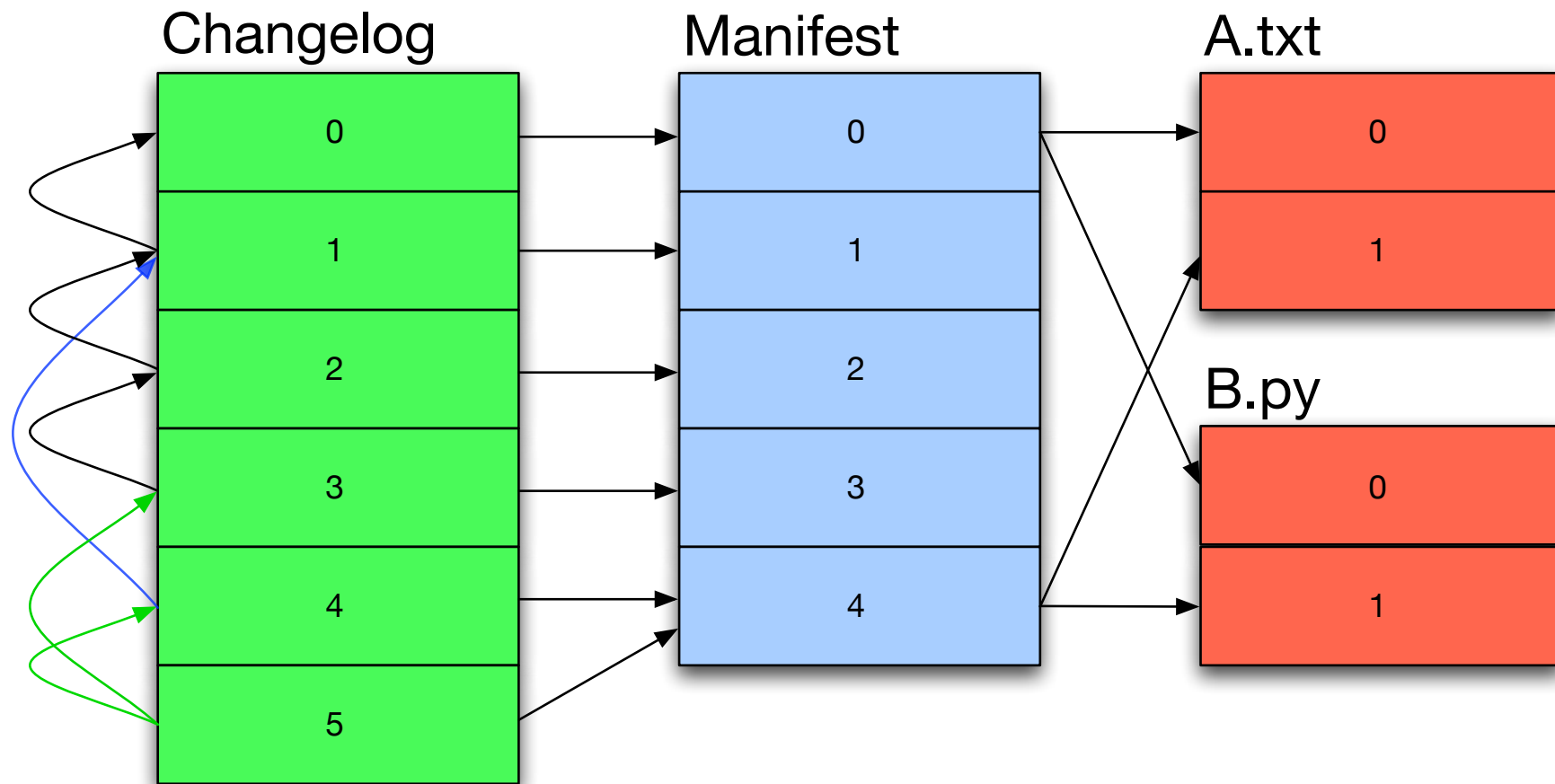
- Generic efficient storage mechanism
- Separate index for $O(1)$ access
- Append-only, light on seeks, DAG
- Optimize local decision, no second guessing
- Does not interpret stored data

The revlog index

rev	offset	length	base	linkrev	nodeid	p1	p2
10265	1925325	195	10265	10265	0d8577e1f56b	25e572394f5c	000000000000
10266	1925520	114	10266	10266	e2c24bb25b36	d6512b3e9ac0	0d8577e1f56b
10267	1925634	142	10266	10267	7e3a685be2f3	e2c24bb25b36	000000000000
10268	1925776	226	10266	10268	2fa95306d40c	0d8577e1f56b	000000000000
10269	1926002	295	10269	10269	acf001ee5ef8	2fa95306d40c	000000000000
10270	1926297	125	10270	10270	c2e27f7966a7	7e3a685be2f3	acf001ee5ef8
10271	1926422	288	10270	10271	9b38bec5dc29	c2e27f7966a7	000000000000

Three tiers

- Changelog: changeset metadata (hg log)
- Manifest: files/revs at changeset (hg up)
- Filelogs: per-file revisions (hg log <file>)



Changeset metadata

- user, message, date/time + timezone
- parent(s)
- branch name
- "extra"

.hgtags

- Just a file in the tree
- Ugly?
- Full history + merging capabilities

Hooks & Extensions

Hooks

- Can be in-process (Python) or outside
- Controlling vs informative
- pre-/post-command



Extensions

- Keeping the core simple
- Very powerful extension mechanism
- Just a simple Python module
- Controlled monkeypatching FTW

Mechanisms

- `uisetup()`
- `reposestap()`
- `wrapcommand()`
- `cmdtable`

Example code

```
def reposetup(ui, repo):
    if not ui.has_section('sshid'):
        return
    keys = identities()
    for k, v in ui.configitems('sshid'):
        if k.strip() in keys:
            ui.setconfig('ui', 'username', v.strip())
            break
```

[sshid]

djc-2010-02 = Dirkjan Ochtman <dirkjan@ochtman.nl>

xxx-2009-10 = Other User <other@example.org>

Questions & Answers

... or find me in the hallways