A Glimpse into a Smoother Version Control Experience

Pierre-Yves David

OCTOBUS

FOSDEM 2025

In a nutshell

- Distributed Version Control System
- Sibling of Git
- 20 years old, still active
- Inspiration for multiple new VCS (Sapling, Jujitsu)
- Great scaling, featureful, extensible
- Distributed history rewriting, etc.

This lighting talk will

NOT

cover all this

What's is it like to work with Mercurial?

Let **us** edit a small script!

(with Mercurial)

```python
#!/usr/bin/python3
# Fibonacci Sequence
#
# Invented by Leonardo of Pisa in 1202

def fibonacci(n, sequence=None):
    """returns the value of the fibonacci sequence at given index"""
    # bootstrap values
    if sequence is None:
        sequence = [0, 1]
    # computes up to the target index
    if len(sequence) <= n:
        # compute the next value
        sequence.append(sum(sequence[-2:]))
        fibonacci(n, sequence)
    # return the result
    return sequence[n]

if __name__ == '__main__':
    import sys
    print(fibonacci(int(sys.argv[1])))
```

A developer writes a simple patch

```
$ hg diff
diff --git a/fibonacci.py b/fibonacci.py
@@ -5,6 +5,8 @@

def fibonacci(n, sequence=None):
     """returns the value of the fibonacci sequence at given index"""
+    if n <= 0:
+        raise ValueError(f"invalid fibonaci index: {n}")
     # bootstrap values
     if sequence is None:
         sequence = [0, 1]
```

A developer writes a simple patch

```
$ hg diff
diff --git a/fibonacci.py b/fibonacci.py
@@ -5,6 +5,8 @@

def fibonacci(n, sequence=None):
     """returns the value of the fibonacci sequence at given index"""
+    if n <= 0:
+        raise ValueError(f"invalid fibonaci index: {n}")
     # bootstrap values
     if sequence is None:
         sequence = [0, 1]
```

```
$ hg topic robustness
marked working directory as topic: robustness
```

A developer writes a simple patch

```
$ hg diff
diff --git a/fibonacci.py b/fibonacci.py
@@ -5,6 +5,8 @@

def fibonacci(n, sequence=None):
     """returns the value of the fibonacci sequence at given index"""
+    if n <= 0:
+        raise ValueError(f"invalid fibonaci index: {n}")
     # bootstrap values
     if sequence is None:
         sequence = [0, 1]

$ hg topic robustness
marked working directory as topic: robustness

$ hg commit
                          <TYPE COMMIT MESSAGE>
```

A developer writes a simple patch

```
$ hg diff
diff --git a/fibonacci.py b/fibonacci.py
@@ -5,6 +5,8 @@

def fibonacci(n, sequence=None):
     """returns the value of the fibonacci sequence at given index"""
+    if n <= 0:
+        raise ValueError(f"invalid fibonaci index: {n}")
     # bootstrap values
     if sequence is None:
         sequence = [0, 1]

$ hg topic robustness
marked working directory as topic: robustness

$ hg commit
                      <TYPE COMMIT MESSAGE>
$ hg push
added 1 changesets with 1 changes to 1 files
```

(And two other patches)

```
$ hg stack
### topic: robustness
### target: default (branch)
s3@ Remove the full sequence list (current)
s2: Avoid function recursion
s1: Reject negative indexes
s0^ Small implementation of the Fibonnaci sequence (base)
```

Message from your Reviewer

Looks good to me.
I fixed a typo.
I rebased it.
This is now merged.
Thanks

Someone touched my branch ?

Collaboration on my branch !

Before

```
$ hg stack --verbose
### topic: robustness
### target: default (branch)
s3(27eb7f00e3cb)@ Remove the full sequence list (current)
s2(5901b3018028): Avoid function recursion
s1(fbf409861668): Reject negative indexes
s0(72e4e2e8380f)^ Small implementation of the Fibonacci sequence
(base)
```

```
$ hg pull
added 2 changesets with 2 changes to 1 files
obsoleted 1 changesets
2 new orphan changesets
```

After

```
$ hg stack -v
### topic: robustness
### target: default (branch), 2 behind
s2(1b4e9c516975)@ Remove the full sequence list (current orphan)
s1(71a068c4b394)$ Avoid function recursion (orphan)
s0(1f159ed15f9b)^ Reject negative indexes (base)
```

First Patch was Touched

```
$ hg obslog --rev s0
o  5c0bc9f4a836 (5) Reject negative indexes
|    rebased(parent) from 7c78a3d0e24a using rebase
|    by "Victor <rat@x.es>" (Sun Jan 02 13:33:00 2025 +0000)
|
x  7c78a3d0e24a
|    amended(content) from fbf409861668 using amend
|    by "Victor <rat@x.es>" (Sun Jan 02 13:32:00 2025 +0000)
|
x  fbf409861668 (1) Reject negative indexes
```

First Patch was Changed

```
$ hg obslog --rev s0 -f --patch
o  5c0bc9f4a836 (5) Reject negative indexes
|    rewritten(parent, content) from fbf409861668
|    using amend, rebase by "Victor <rat@x.es>" (Sun Jan 02 13:33:00 2025 +0000)
|      diff --git a/fibonacci.py b/fibonacci.py
|      @@ -6,7 +6,7 @@
|       def fibonacci(n, sequence=None):
|           """returns the value of the fibonacci sequence at given index"""
|           if n <= 0:
|      -        raise ValueError(f"invalid fibonaci index: {n}")
|      +        raise ValueError(f"invalid fibonacci index: {n}")
|           # bootstrap values
|           if sequence is None:
|               sequence = [0, 1]
|
|
x  fbf409861668 (1) Reject negative indexes
```

So, about these orphans?

```
$ hg stack -v
### topic: robustness
### target: default (branch), 2 behind
s2(1b4e9c516975)@ Remove the full […] (current orphan)
s1(71a068c4b394)$ Avoid function recursion (orphan)
s0(1f159ed15f9b)^ Reject negative indexes (base)
```

Automated Resolution

```
$ hg evolve
move:[s1] Avoid function recursion
atop:[default] Reject negative indexes
merging fibonacci.py
move:[s2] Remove the full sequence list
merging fibonacci.py
working directory is now at c546abe53372
```

Ready to push

```
$ hg stack
### topic: robustness
### target: default (branch)
s2@ Remove the full sequence list (current)
s1: Avoid function recursion
s0^ Reject negative indexes (base)

$ hg push
added 2 changesets with 2 changes to 1 files
```

Message from your Reviewer

First changeset:

Replace the `while` loop by a `for` loop

Second changeset:

Use `a, b = b, a +b`

Typo in the commit message

I added types annotation to the script

Moving to the first patch

```
$ hg previous
1 files updated
[s1] Avoid function recursion
```

Moving to the first patch

```
$ hg previous
1 files updated
[s1] Avoid function recursion

$ hg stack
### topic: robustness
### target: default (branch)
s2: Remove the full sequence list
s1@ Avoid function recursion (current)
s0^ Reject negative indexes (base)
```

Updating the First Patch

```
$ hg diff
diff --git a/fibonacci.py b/fibonacci.py
@@ -10,9 +10,7 @@
     # bootstrap values
     sequence = [0, 1]
     # computes up to the target index
-    i = 1
-    while i < n:
-        i += 1
+    for _ in range(1, n):
         # compute the next value
         sequence.append(sum(sequence[-2:]))
     # return the result

$ hg amend
```

Updating the First Patch

```
$ hg diff
diff --git a/fibonacci.py b/fibonacci.py
@@ -10,9 +10,7 @@
     # bootstrap values
     sequence = [0, 1]
     # computes up to the target index
-    i = 1
-    while i < n:
-        i += 1
+    for _ in range(1, n):
         # compute the next value
         sequence.append(sum(sequence[-2:]))
     # return the result

$ hg amend
1 new orphan changesets
```

Moving to the Second Patch

```
$ hg stack
### topic: robustness
### target: default (branch)
s2$ Remove the full sequence list (orphan)
s1@ Avoid function recursion (current)
s0^ Reject negative indexes (base)
```

Moving to the Second Patch

```
$ hg stack
### topic: robustness
### target: default (branch)
s2$ Remove the full sequence list (orphan)
s1@ Avoid function recursion (current)
s0^ Reject negative indexes (base)

$ hg next
move:[s2] Remove the full sequence list
atop:[s1] Avoid function recursion
merging fibonacci.py
working directory is now at 6ab17facd802
```

Updating the Second Patch

```
$ hg pdiff
diff --git a/fibonacci.py b/fibonacci.py
@@ -8,13 +8,14 @@
     if n <= 0:
         raise ValueError(f"invalid fibonacci index: {n}")
     # bootstrap values
-    sequence = [0, 1]
+    current = 1
+    next = 1
     # computes up to the target index
     for _ in range(1, n):
         # compute the next value
-        sequence.append(sum(sequence[-2:]))
+        current, next = next, current + next
     # return the result
-    return sequence[n]
+    return current

 if __name__ == '__main__':
     import sys
$ hg amend --edit
```

<FIX COMMIT MESSAGE>

Getting Upstream Changes

```
$ hg pull
added 1 changesets with 1 changes to 1 files (+1 heads)
```

Getting Upstream Changes

```
$ hg pull
added 1 changesets with 1 changes to 1 files (+1 heads)

$ hg stack
### topic: robustness
### target: default (branch), 1 behind
s2@ Remove the full sequence list (current)
s1: Avoid function recursion
s0^ Reject negative indexes (base)
```

Rebasing on Upstream Changes

```
$ hg rebase
rebasing 8:bb733eacf86a robustness "Avoid function recursion"
merging fibonacci.py
warning: conflicts while merging fibonacci.py! (edit, then use 'hg resolve --mark')
unresolved conflicts (see 'hg resolve', then 'hg rebase --continue')
                                           <RESOLVE>
```

Rebasing on Upstream Changes

```
$ hg rebase
rebasing 8:bb733eacf86a robustness "Avoid function recursion"
merging fibonacci.py
warning: conflicts while merging fibonacci.py! (edit, then use 'hg resolve --mark')
unresolved conflicts (see 'hg resolve', then 'hg rebase --continue')
                                           <RESOLVE>
$ hg continue
rebasing 8:bb733eacf86a robustness "Avoid function recursion"
rebasing 10:765305cd9f41 robustness "Remove the full sequence list"
merging fibonacci.py
```

Rebasing on Upstream Changes

```
$ hg rebase
rebasing 8:bb733eacf86a robustness "Avoid function recursion"
merging fibonacci.py
warning: conflicts while merging fibonacci.py! (edit, then use 'hg resolve --mark')
unresolved conflicts (see 'hg resolve', then 'hg rebase --continue')
                                              <RESOLVE>
$ hg continue
rebasing 8:bb733eacf86a robustness "Avoid function recursion"
rebasing 10:765305cd9f41 robustness "Remove the full sequence list"
merging fibonacci.py

$ hg push
added 2 changesets with 2 changes to 1 files
5 new obsolescence markers
obsoleted 2 changesets
```

From the Reviewer's Seat

Getting the Latest Version

```
$ hg pull
added 2 changesets with 2 changes to 1 files (2 drafts)
5 new obsolescence markers
obsoleted 2 changesets
```

Inspecting Conflict Resolutions

```
$ hg obslog -f --patch "robustness#stack[1]"
o  086d52f0e12c (8) Avoid function recursion
|    rewritten(parent, content) from 460ffb9a873f using amend, rebase
|    by "Babar <b@b.ar>" (Thu Jan 01 00:00:00 1970 +0000)
|      diff --git a/fibonacci.py b/fibonacci.py
|      @@ -3,22 +3,14 @@
|       #
|       # First known discovery by Acharya Pingala around -200
|
|      -<<<<<<< predecessor:        460ffb9a873f - b: Avoid function recursion
|      -def fibonacci(n):
|      -||||||| predecessor-parent: 5c0bc9f4a836 - b: Reject negative indexes
|      -def fibonacci(n, sequence=None):
|      -=======
|      -def fibonacci(n: int, sequence: list[int]|None = None) -> int:
|      ->>>>>>> successor-parent:   b6506edd1b5f - rat: Add typing information
|      +def fibonacci(n:int ) -> int:
|           """returns the value of the fibonacci sequence at given index"""
|           if n <= 0:
|               raise ValueError(f"invalid fibonacci index: {n}")
|           # bootstrap values
|           sequence = [0, 1]
|           # computes up to the target index
|      -    i = 1
|      -    while i < n:
|      -        i += 1
|      +    for _ in range(1, n):
|               # compute the next value
|               sequence.append(sum(sequence[-2:]))
|           # return the result
|
|
x  460ffb9a873f (5) Avoid function recursion
```

Inspecting Commit Message Changes

```
$ hg obslog -f --patch "robustness#stack[2]"
o  7e391b6e23bb (9) Remove the full sequence list
|    rewritten(description, parent, content) from c546abe53372 using amend, evolve, rebase
|    by "Babar <b@b.ar>" (Thu Jan 01 00:00:00 1970 +0000)
|      diff -r c546abe53372+ -r 7e391b6e23bb changeset-description
|      @@ -1,3 +1,3 @@
|       Remove the full sequence list
|
|      -We never use value older than the first two, so let us same some memory.
|      +We never use values older than the first two, so let us save some memory.
|
|      diff --git a/fibonacci.py b/fibonacci.py
|      --- a/fibonacci.py
|      +++ b/fibonacci.py
|      @@ -13,9 +13,7 @@
|           # computes up to the target index
|           for _ in range(1, n):
|               # compute the next value
|      -        tmp = current
|      -        current = next
|      -        next = tmp + current
|      +        current, next = next, current + next
|           # return the result
|           return current
|
|
|
x  c546abe53372 (6) Remove the full sequence list
```

Did you notice?

obslog --patch

```
diff --git a/fibonacci.py b/fibonacci.py
@@ -13,9 +13,7 @@
     # computes up to the target index
     for _ in range(1, n):
         # compute the next value
-        tmp = current
-        current = next
-        next = tmp + current
+        current, next = next, current + next
     # return the result
     return current
```

obslog --patch

```
diff --git a/fibonacci.py b/fibonacci.py
@@ -13,9 +13,7 @@
     # computes up to the target index
     for _ in range(1, n):
         # compute the next value
-        tmp = current
-        current = next
-        next = tmp + current
+        current, next = next, current +
next
     # return the result
     return current
```

diff --from α --to ω

```
diff --git a/fibonacci.py b/fibonacci.py
@@ -1,23 +1,19 @@
 #!/usr/bin/python3
 # Fibonacci Sequence
 #
-# Invented by Leonardo of Pisa in 1202
+# First known discovery by Acharya Pingala around -200

-def fibonacci(n):
+def fibonacci(n:int ) -> int:
     """returns the value of the fibonacci sequence at given
index"""
     if n <= 0:
-        raise ValueError(f"invalid fibonaci index: {n}")
+        raise ValueError(f"invalid fibonacci index: {n}")
     # bootstrap values
     current = 1
     next = 1
     # computes up to the target index
-    i = 1
-    while i < n:
-        i += 1
+    for _ in range(1, n):
         # compute the next value
-        tmp = current
-        current = next
-        next = tmp + current
+        current, next = next, current + next
     # return the result
     return current
```

Epilogue

```
$ hg pull
2 local changesets published
active topic '//robustness' is now empty
```

Conclusion

- **Simple commands** **(and few flags)**

Conclusion

- **Simple commands** **(and few flags)**
- **Propagation of history rewriting**

Conclusion

- **Simple commands (and few flags)**
- **Propagation of history rewriting**
- **History of each commit (with clear diffs**

Conclusion

- **Simple commands** **(and few flags)**
- **Propagation of history rewriting**
- **History of each commit** **(with clear diffs)**
- **Automated instability resolution**

Conclusion

- **Simple commands (and few flags)**
- **Propagation of history rewriting**
- **History of each commit (with clear diffs)**
- **Automated instability resolution**
- **Scoped feature branch**

Conclusion

- **Simple commands (and few flags)**
- **Propagation of history rewriting**
- **History of each commit (with clear diffs)**
- **Automated instability resolution**
- **Scoped feature branch**
- **Safe and simple collaboration on drafts**

Interested in Mercurial ?

- **https://www.mercurial-scm.org/**
- **IRC: #mercurial @liberachat**
- **Matrix: #mercurial:matrix.org**

Commercial Support:
https://octobus.net/

Hosting: https://heptapod.net/

Questions?

Meet us outside!