

Parallelizing Git Checkout

A Case Study of I/O Parallelism on Desktop Applications

Matheus Tavares

July, 13, 2022

Advisor: Prof. Dr. Alfredo Goldman

Computer Science Department

IME USP

USP



- Version Control System (VCS)
 - ▶ *“a system that records changes to a file or set of files over time so that you can recall specific versions later”.*
- *Pro Git. S. Chacon, B. Straub.*
- Created in 2005 by Linus Torvalds, to version control the Linux kernel.

Repo	Size (GiB)	Files
Numpy	0.14	2 K
Gentoo	1.60	87 K
Linux	5.10	77 K
Chromium	36.57	401 K
Windows ⁽²⁰¹⁷⁾	300.00	3.5 M

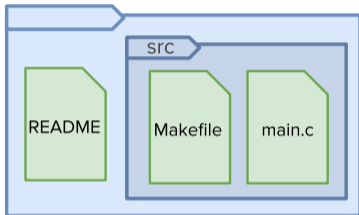
Table 1: *Examples of Git repositories (06/2022).*

Git Checkout

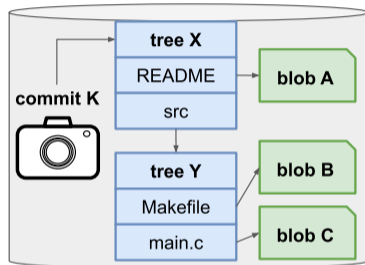
- Checkout: restores files from specific versions.
- Used by clone, checkout, switch, reset, sparse-checkout, etc.
- A local Linux clone takes:
 - ▶ 6 to 8 seconds on SSD
 - ▶ 20s to 2m on HDDs
 - ▶ **5 to 15 minutes on NFS!**

Structures Involved at Checkout

Working Tree



Object Store

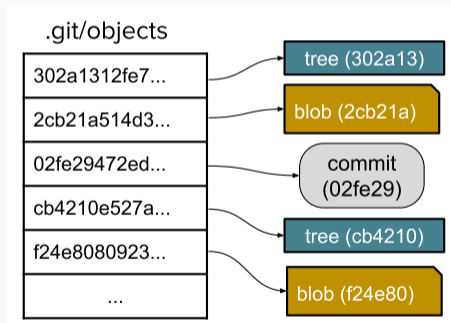


Index

Name	Object	Type	Other stat() data
README	blob A	Regular file	ctime, mtime, inode, etc.
src/Makefile	blob B	Regular file	
src/main.c	blob C	Regular file	

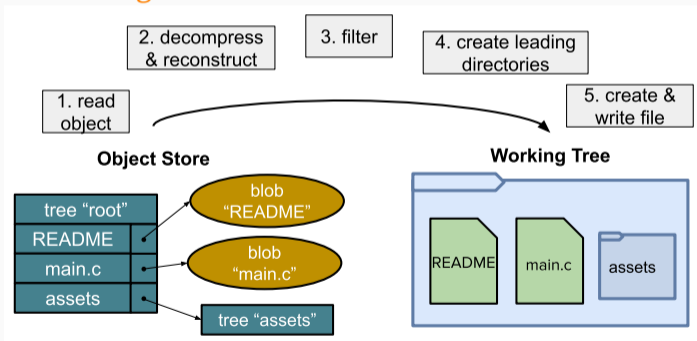
Git Objects

- Compressed using zlib (“deflate”).
- Referenced by their SHA-1 hash.
- Formats:
 - ▶ Loose: one object per file, lower compression.
 - ▶ Packed: many objects per file, higher compression, and *deltification*.



Checkout

1. Read the index file.
2. Update the in-memory index entries.
3. Update the working tree to match the new index.



4. Write the new index to disk.

- “git checkout .” on empty working tree of Linux repo (v5.12).
- Creation of about 72K files.
- Linux eBPF: on- and off-CPU.
- Two copies:
 - ▶ Full repository in a single packfile.
 - ▶ Shallow clone (depth=1) with loose objects only.

Checkout profile on Mango - SSD

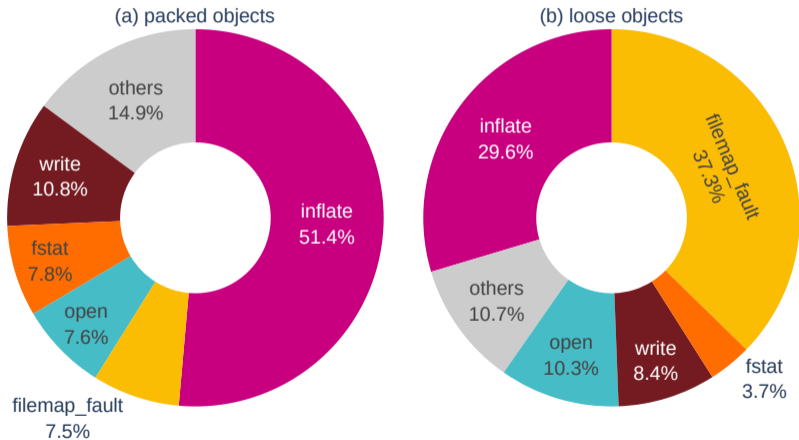


Figure 1: Machine Mango: i7-7700HQ (8 threads), 16 GB RAM, NVMe SSD.

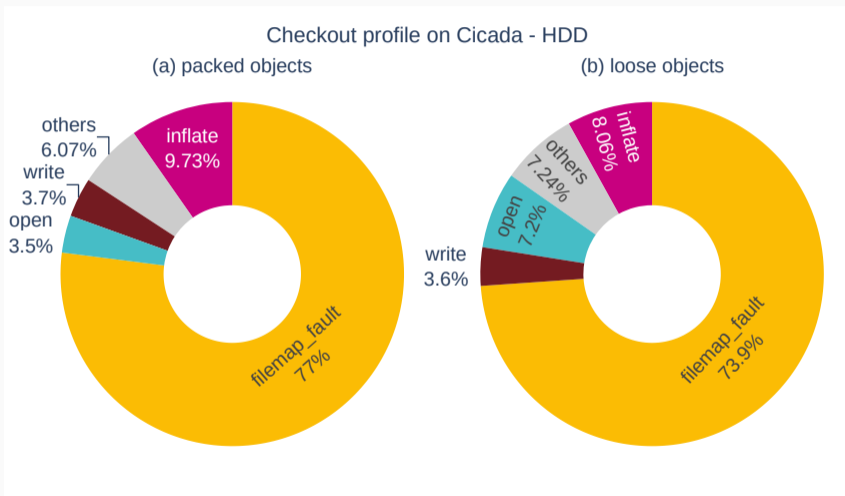


Figure 2: Machine Cicada: i5-3317U (4 threads), 6 GB RAM, SATA 2.6 HDD.

Checkout profile on NFS EBS gp3 - SSD

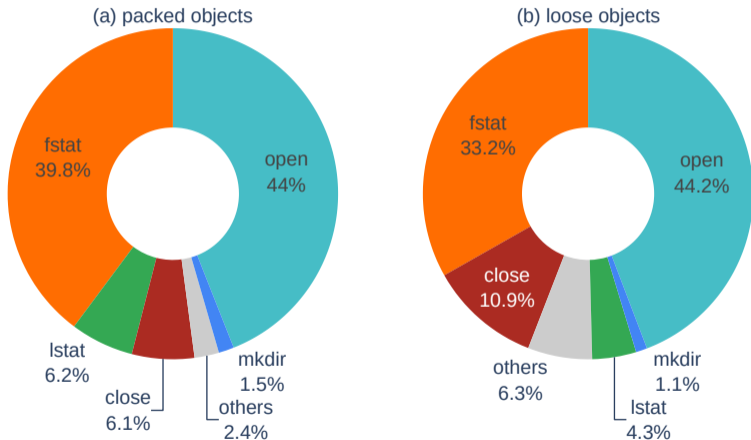
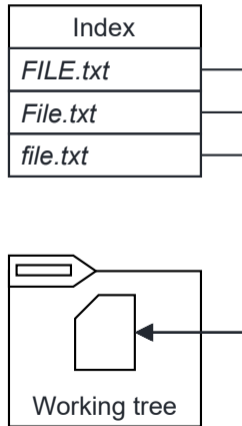


Figure 3: NFS: client and server on AWS EC2 instances. Storage: EBS gp3 SSD. 10

Challenges

- **Threads:** *“Multi-threading anything in git is fraught with challenges as much of the code base is not thread safe.”* - Jeff King
- **Filters:**
 - ▶ Attributes reading is optimized for sequential access.
 - ▶ External filters might assume single instance.
- **Path collisions:**
 - ▶ Concurrent writes to the same file.
 - ▶ Race on file creation vs. removal.
 - ▶ Leading dir replaced by a symlink: **security risk!**



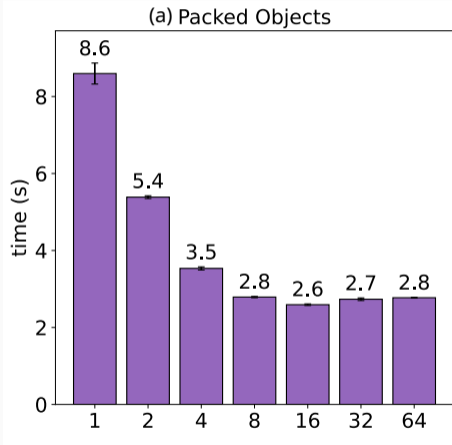
Main contributions

James Pickens (2008)	Nguyễn Thái Ngọc Duy (2016)	Jeff Hostetler (2020)	Matheus Tavares (2021)
<ul style="list-style-type: none">● Initial (threaded) prototype and NFS tests.● Thread-safe discussions and work queue idea.	<ul style="list-style-type: none">● General framework and work queue.● Internal API and <code>poll()</code> loop.● Parallelism on CPU (decompression) and I/O.	<ul style="list-style-type: none">● Entry classification based on filter requirements.● External filters are ineligible for parallelism.● Send attributes to the workers.● IPC, technical doc, and tests.● Adjustment for path collision report.	<ul style="list-style-type: none">● Study and merge of previous works.● <code>lstat()</code> cache analysis and CVE-2021-21300.● More tests: collisions, submodules, <code>--force</code>, etc.● Parallel write on non-empty working tree.● Performance and memory benchmarks.

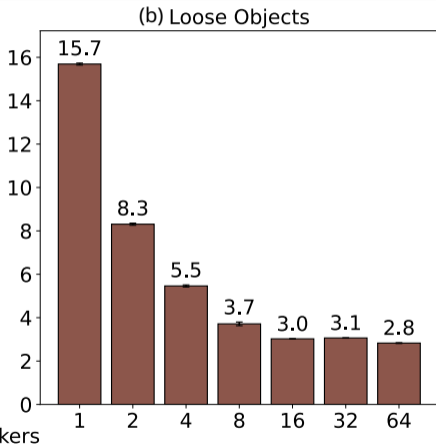
Results: Linux

Mango: i7-7700HQ (8 ths), 16 GB RAM, NVMe SSD (ext4)

Manjaro Linux - Linux repo



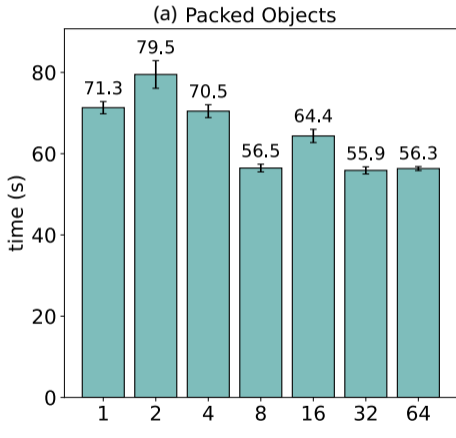
3.3x speedup



5.6x speedup

Cicada: i5-3317U (4 ths), 6 GB RAM, SATA 2.6 HDD (ext4)

Manjaro Linux - Linux repo



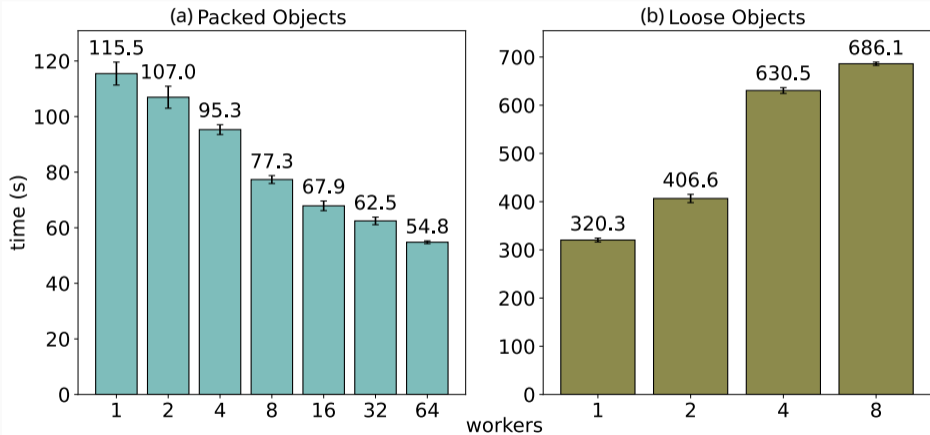
1.27x speedup



slowdown

Wall-e: i5-4210U (4 ths), 8 GB RAM, SATA 3.0 **HDD** (ext4)

Manjaro Linux - Linux repo

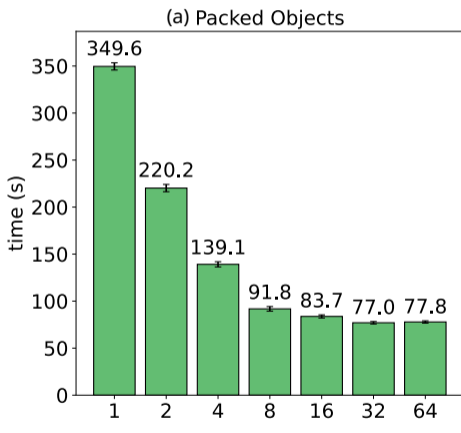


2.1x speedup

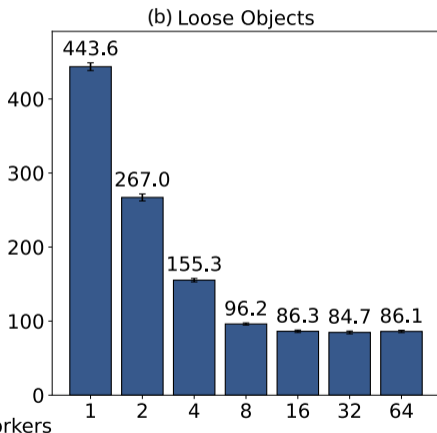
slowdown

NFS gp3: on AWS EC2 instances, EBS gp3 **SSD** (XFS)

Amazon Linux - Linux repo

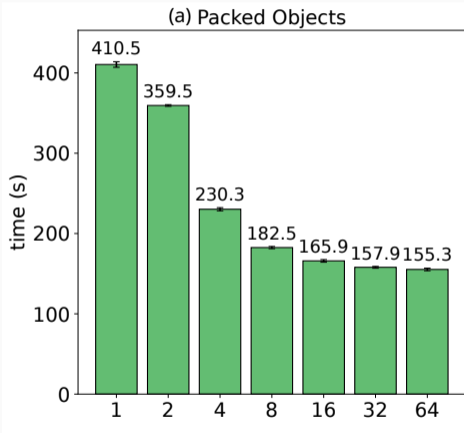


4.5x speedup

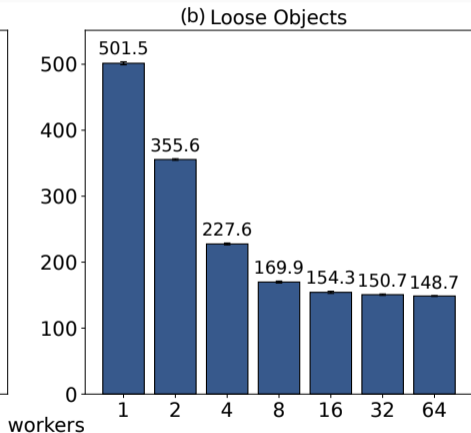


5.2x speedup

NFS LAN: client on Mango, server on Cicada HDD
Git repo



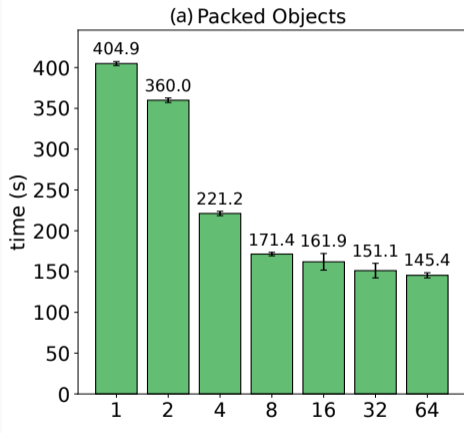
2.6x speedup



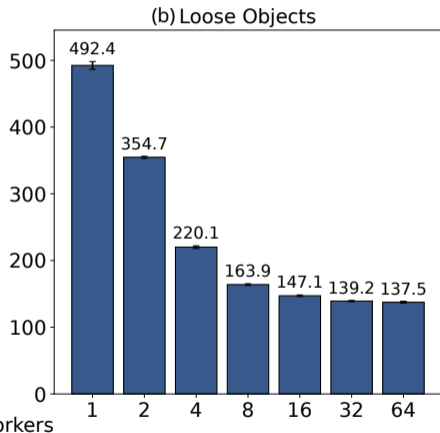
3.4x speedup

NFS LAN - One core: client on Mango, server on Cicada HDD

Git repo



2.8x speedup

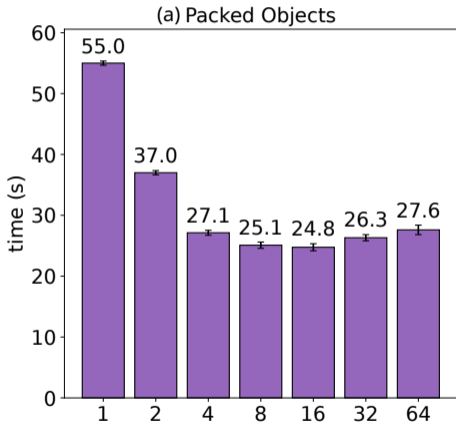


3.6x speedup

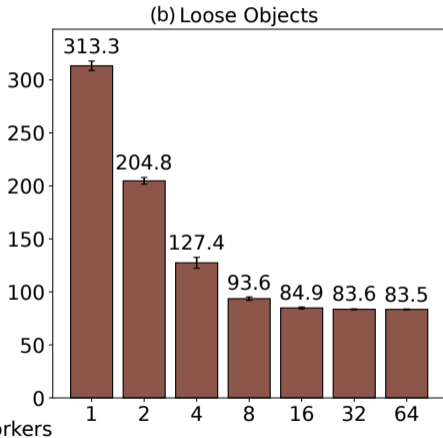
Results: Windows

Mango: i7-7700HQ (8 ths), 16 GB RAM, NVMe SSD (ext4)

Windows 10 - Linux repo



2.2x speedup



3.8x speedup

Cicada: i5-3317U (4 ths), 6 GB RAM, SATA 2.6 HDD (ext4)

Windows 10 - Linux repo



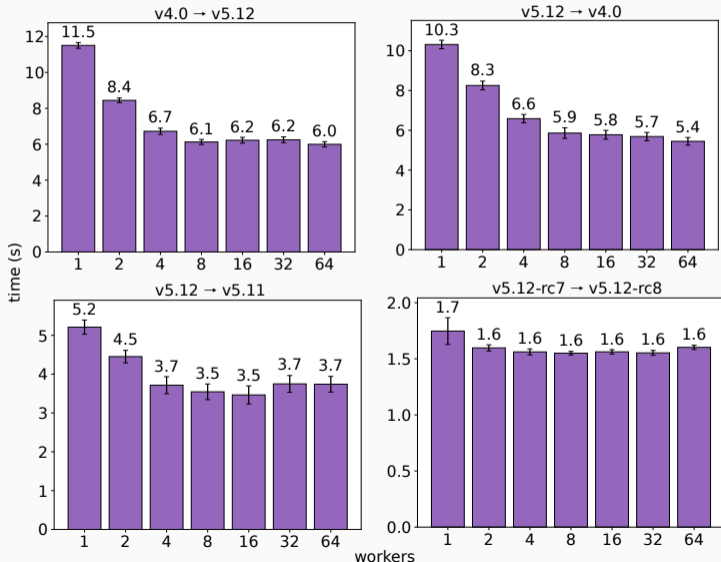
1.2x speedup

1.7x speedup

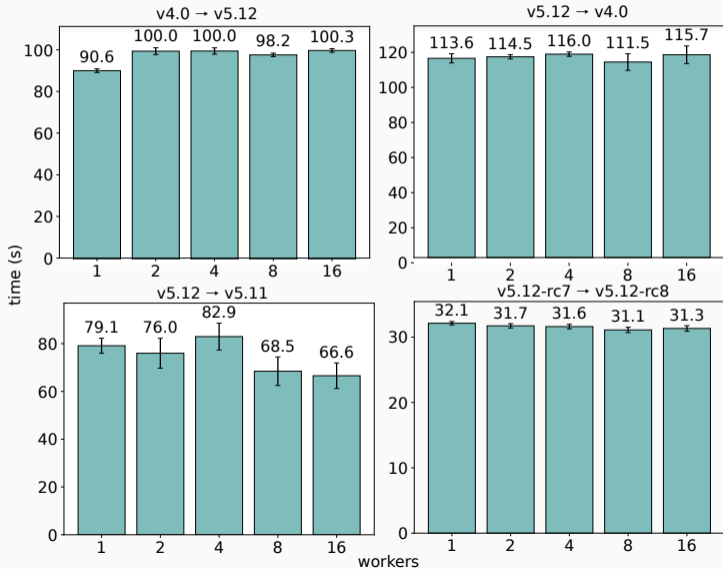
Extra Benchmarks

Linux

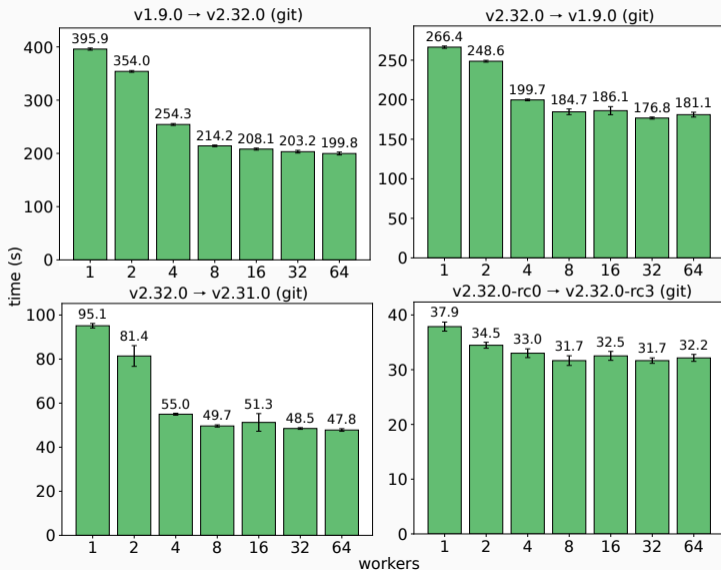
Mango: i7-7700HQ (8 ths), 16 GB RAM, NVMe SSD (ext4) - Linux repo



Cicada: i5-3317U (4 ths), 6 GB RAM, SATA 2.6 HDD (ext4) - Linux repo



NFS LAN: client on *Mango*, server on *Cicada* HDD - Git repo



Conclusions

- Up to 4.5x speedups on NFS and 3.6x on SSDs.
- Merged upstream and released on Jun 6th, 2021.
- Found and fixed a couple bugs in the checkout code.
 - ▶ RCE vulnerability: [CVE-2021-21300](#).

Thank you!