



Accelerate your SCM



Pushing Git performance Over the *SPEED LIMIT*

Luca Milanese

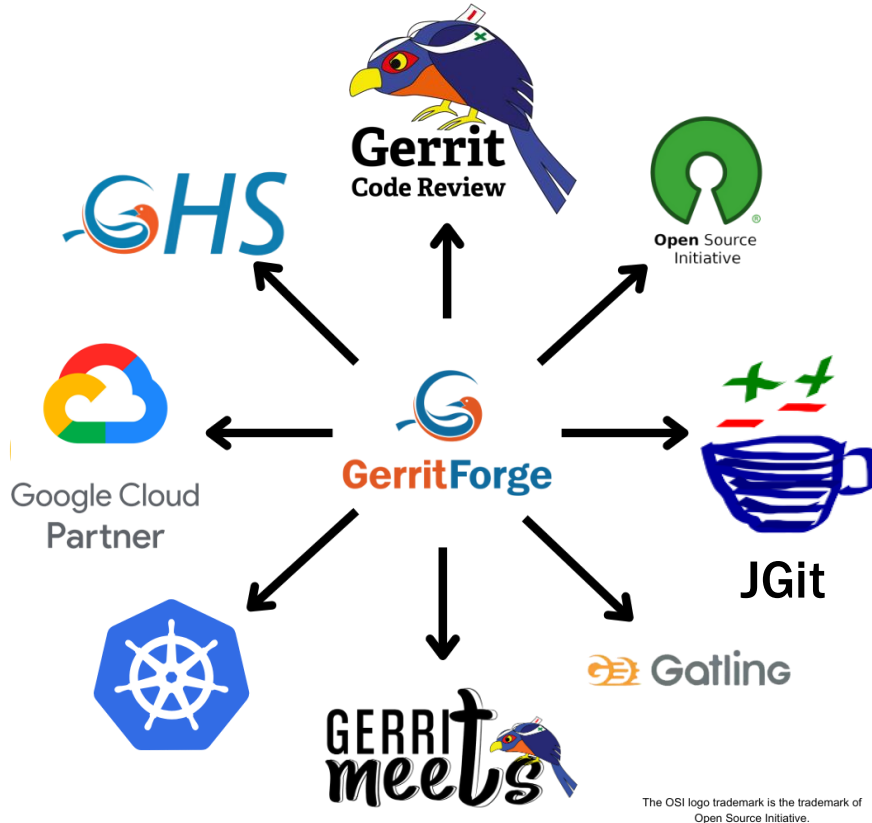
GERRITFORGE Inc. CEO

JGit Committer | Gerrit Maintainer

Gerrit Release Manager



About GerritForge Inc.



- HQ in Sunnyvale – California
- Distributed worldwide
USA, UK, Italy, Poland, Germany, Israel
- **JGit and Gerrit Code Review since 2012**
- 16k merged contributions, 1M LOCs
- 5x Gerrit Maintainers | 17x Contributors
- 2x JGit Committers
- 2x Gerrit Release Managers
- 1x Gerrit Engineering Steering Committee
- 1x Gerrit Community Managers





Luca Milanesio
GerritForge, Inc. CEO

- JGit Committer
- Gerrit Code Review Maintainer
- Gerrit Code Review Release Manager
- Gerrit Code Review ESC
- Created the Gerrit Plugins ecosystem
- Maintainer of GerritHub.io since 2014



JGit & Gerrit Contributions since 2012

GIT Commits - Overall Statistics

3,820

#Commits

97

#Projects

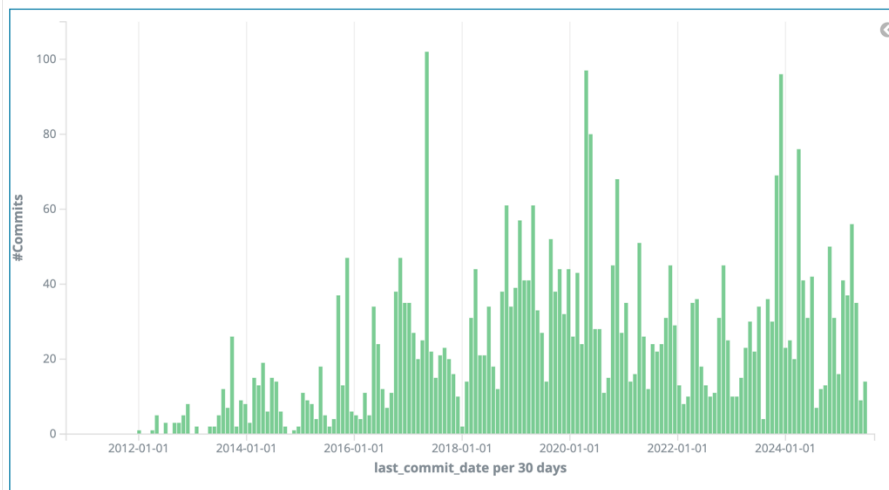
1

#Authors

1

#Organizations

GIT Commits - commits over time



Source: <https://analytics.gerrithub.io>

Me and the Git community



1. Luca Milanesio
GerritForge



2. Peff (aka Jeff King)
GitHub



3. Martin Fick
Nvidia



GitTogether 2011

📍 GooglePlex - Mountain View - CA on 📅 26th-28th Oct 2011



Today's agenda – Git and GHS performance racetrack

***“Performance
is a feature”***
[Shawn]

Git Bitmap
[Colby - 2012]

Git at High-Speed (aka GHS)
R&D [Luca, Ponch, Tony
Shane McIntosh – 2023/2024]

***GHS Research Paper
at the ICSE 2025***
[Shane McIntosh, Ponch]

***GHS for GitLab
and GitHub Enterprise***

2025

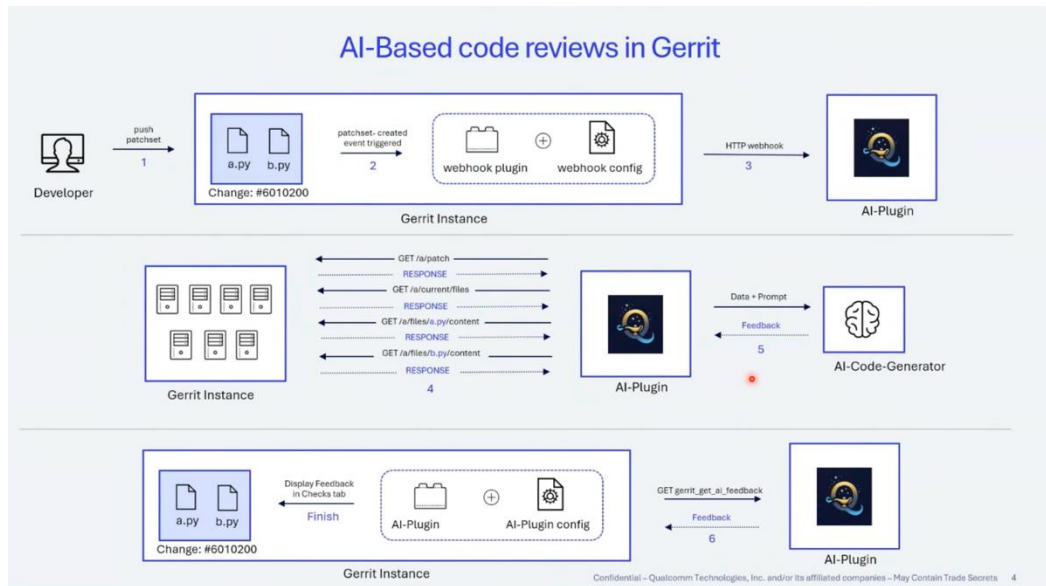


Disclaimer

GerritForge Inc.'s R&D over the years has produced numerous fixes and improvements to the **JGit** and **Gerrit Code Review** projects, which are all **Open-Source**. The “*Git at High-Speed*” (aka GHS) AI agent and engine, is intellectual property of GERRITFORGE Inc. patent pending in the USA (US 63/693,044) and the EU (PN857132EP)



Why going faster actually matters



AI-Based
Software
Development
Lifecycle

Amplifies
Git and Gerrit
traffic



Qualcomm



Why going faster actually matters

Challenges

- H2 Cache Management
- Stability around replication
- Notedb issues with NFS
- Renaming Projects post 3.x Upgrade
- Increased Load (New AI use-cases & others)

Future Roadmap

- OS Upgrades + Containerize Gerrit primaries
- Gerrit life cycle management (Upgrades/Plugins)
- Distributed caches
- Clones using HTTPs (Replicas)
- Vertically scaling K8 replicas (globally) + HPA
- Pull replication



Qualcomm



“Performance is a feature” – Where it all begun



Shawn Pearce (Google) - JGit and Gerrit Code Review projects founder

- True and unique leader
- Fueled innovation and performance on Git and Gerrit
- JGit WindowCache
- JGit DFS backend and Multi-Site at Google
- Smart HTTP protocol
- ... and much more !



Git reachability bitmap – 2012



Colby Ranger (Google) – Gerrit Hackathon @GooglePlex

- Introduces the idea of a sparse table for Git objects' reachability (Git bitmap)

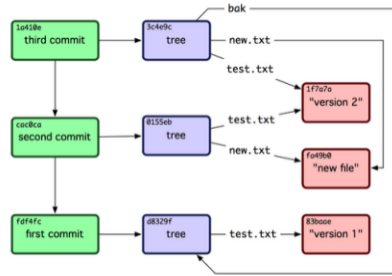


Image credit: ProGit, by Scott Chacon

	0155eb	1a410e	1f7a7a	3c439c	83baae	cac0ca	d8329f	fa49b0	fdf4fc
1a410e	1	1	1	1	1	1	1	1	1
cac0ca	1	0	1	0	1	1	1	1	1
fdf4fc	0	0	0	0	1	0	1	0	1

1a410e AND NOT fdf4fc =>
1 1 1 1 0 1 0 1 0



Git at High-Speed (GHS) academic research Team - 2023



Luca Milanesio
GerritForge CEO
Senior JGit Committer
Gerrit Maintainer & Release Manager



Shane McIntosh
Associate Professor
University of Waterloo
Ontario, Canada



Fabio Ponciroli (aka Ponch)
GerritForge
Gerrit Maintainer



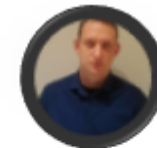
Antonio Barone (aka Tony)
GerritForge
Gerrit Maintainer & Release Manager



Jacek Centkowski
GerritForge | SoftwareMill
Gerrit Maintainer



Marcin Czech
GerritForge
Gerrit Contributor



R&D baseline repository and traffic simulation

Repository use-case baseline

- Developer “real-world” of a Git mono-repo
 - 100k+ refs, 1M+ Git objects, 1k+ committers, 80k+ commits per day
- CI/CD validation workflow on developer’s code
 - 500k+ fetches/day, 40k+ clones/day, 100k+ feedback/day



Gatling-Git : extension to the Gatling testing

- Abstract the Git commands from the underlying protocol
Git/HTTPS and Git/SSH implemented
- Ability to define the shape of users' sessions
- Introduce user session variables to isolate the user's actions



Git at High-Speed testing arena



HW specs for the tests:

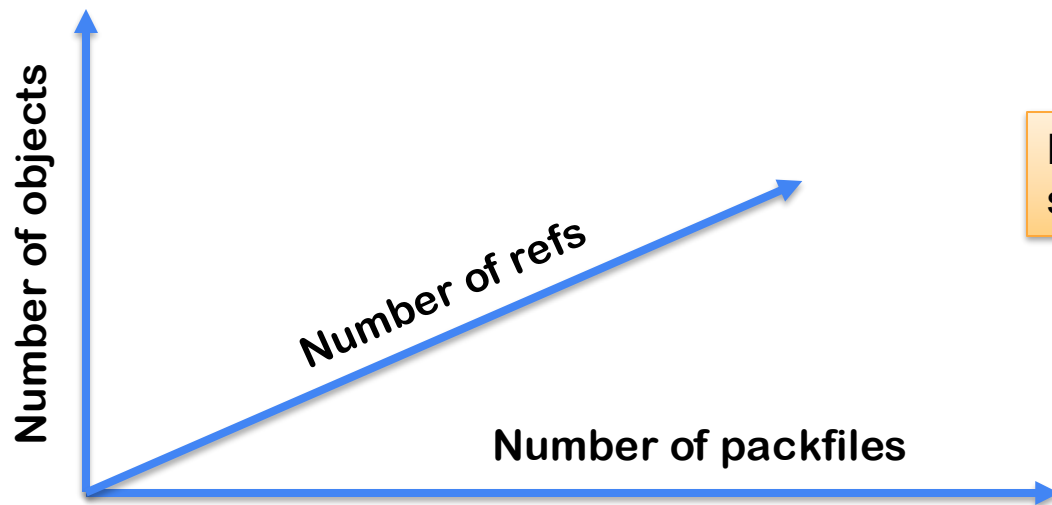
- Intel(R) Xeon(R) Gold 6438Y+
- Local SSD - 2GBps (write) 7GBps (read)
- 128 CPUs
- 128 GBytes RAM

Git servers under test:

- Gerrit Code Review v3.12.2
- GitLab CE v17.8.1
- GitHub Enterprise v3.17.2 (testing WIP)



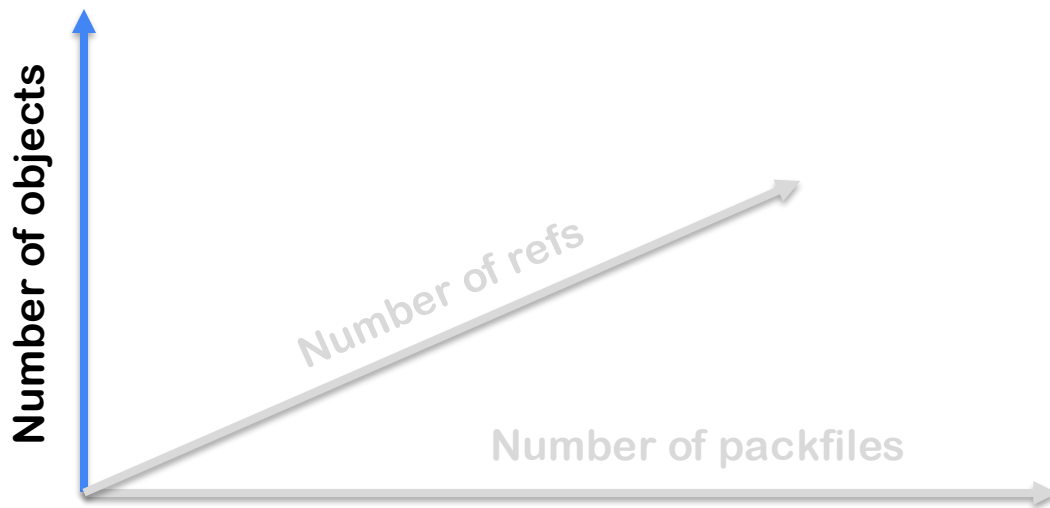
What dimensions influence Git performance?



Does the repository size **REALLY** matter?



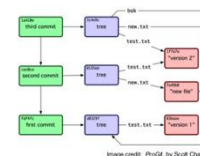
How the number of objects slows down a repo?



- Counting
- Rev walk
- Search for reuse
- Delta chains resolution



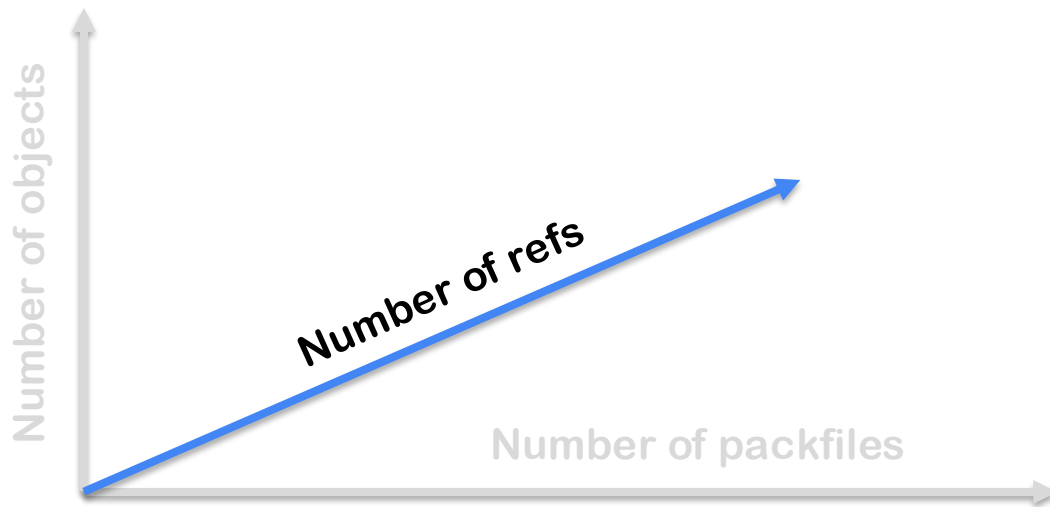
Bitmaps, commit-graphs, packfiles



```
015fab 1a410e 1f7a7a 3c439c 83bade cac0ca 08329f 2a49b0 fdf4fc  
1a410e 1 1 1 1 1 1 1 1 1  
cac0ca 1 0 1 0 1 1 1 1 1  
fdf4fc 0 0 0 0 1 0 1 0 1  
  
1a410e AND NOT fdf4fc =>  
1 1 1 1 0 1 0 1 0
```



How the number of objects slows down a repo?



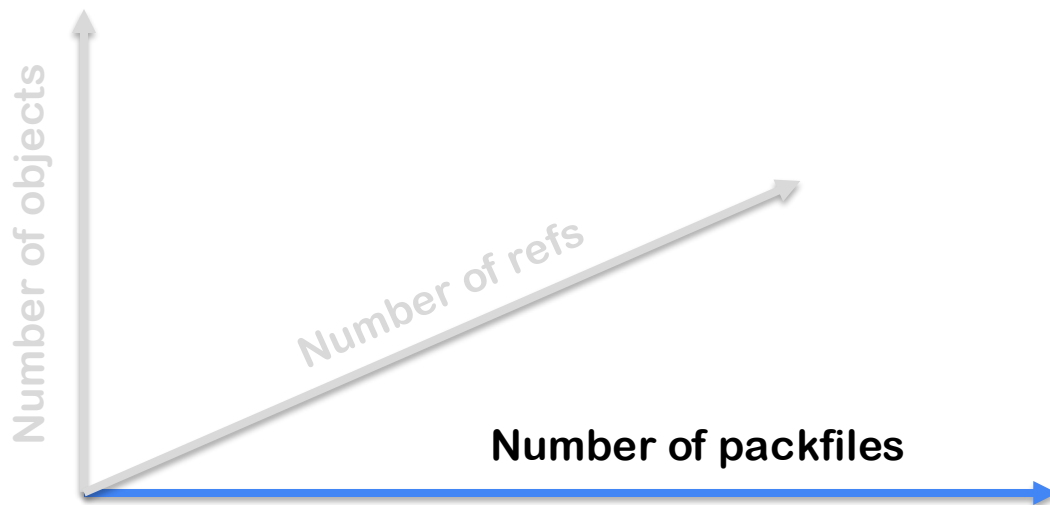
- Mutability
- inodes explosions
- Directory metadata slowdown
- Attribute caching
- Racy reads
- Filesystem time resolution
- Locking



Packed-refs
Reftable



How the number of objects slows down a repo?



- Objects duplication
- Navigation slowdown
- Search-for-reuse explosion
- Memory consumption
- Pruning and stale file handles



Cruft Packfiles
Preserved packfiles
Geometric repacking
Multi-pack index / bitmap



Good bitmaps



bad bitmaps



Anatomy of a good bitmap

- Smaller is better
- Dense for what's more needed
- Resolves 90% of the navigation needed



A **bad bitmap** makes the **Git repo slower** than no bitmaps at all

How many bitmaps?

JGit can control how many and how dense they are

- pack.bitmapExcessiveBranchCount
- pack.bitmapExcludedRefPrefixes
- pack.bitmapRecentCommitCount
- pack.bitmapRecentCommitSpan

Available on
JGit bitmap generation
ONLY



Git at High-Speed (aka GHS) research paper published



Shane McIntosh



Fabio Ponciroli

ICSE 2025
47th International Conference
on Software Engineering

Sun 27 April - Sat 3 May 2025

Ottawa, Ontario, Canada

Access the paper
and learn how GHS
achieves 100x Git performance



**Using Reinforcement Learning to Sustain the
Performance of Version Control Repositories**

<https://gerritforge.com/20250624.ghs.research.whitepaper.pdf>



Git at High-Speed (aka GHS) in a nutshell



Git with regular maintenance



Git with regular maintenance
plus **GHS** side-by-side



Keeps a good Bitmap
always available to serve
quick clones

Understands when to
prune without losing
friction on existing cached
packfiles



Harness reinforcement
learning to choose the
operation for maximum
speed and minimum
overhead

Optimize refs
organization



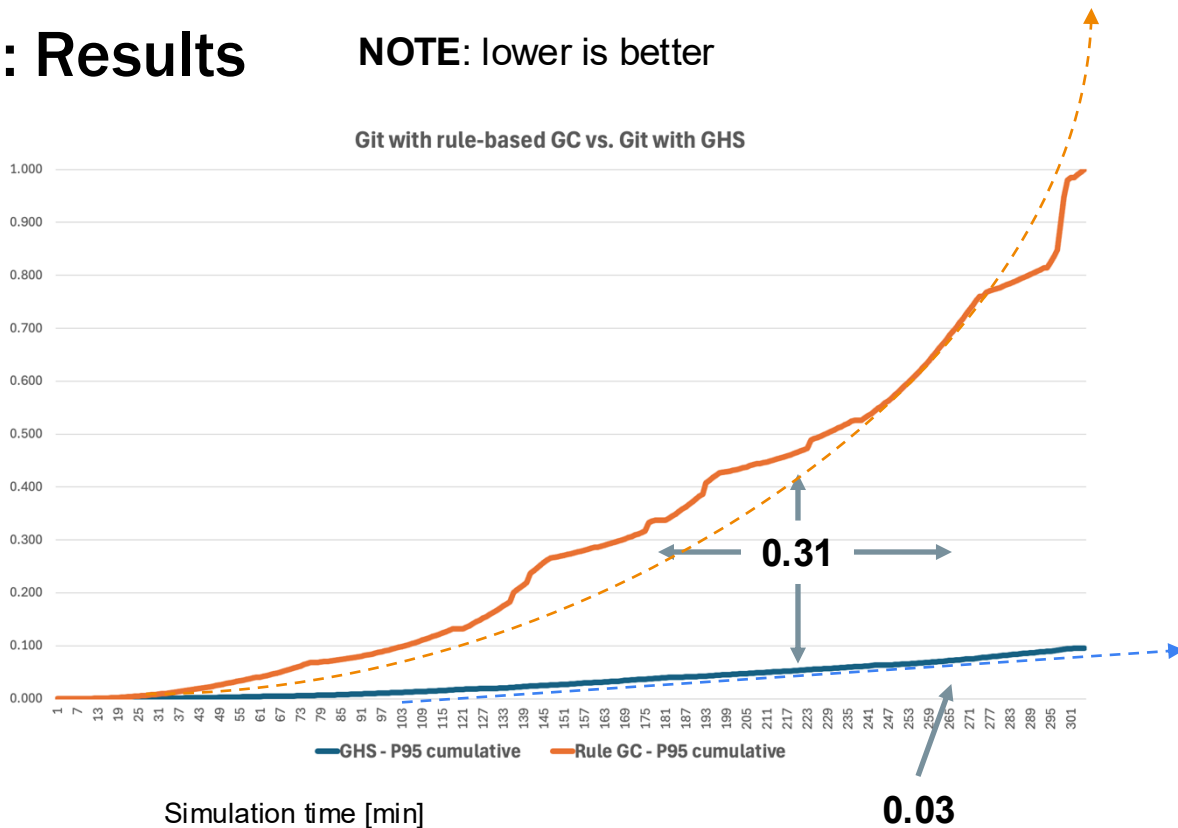
Gerrit Code Review: Results

NOTE: lower is better



Up to 10x faster

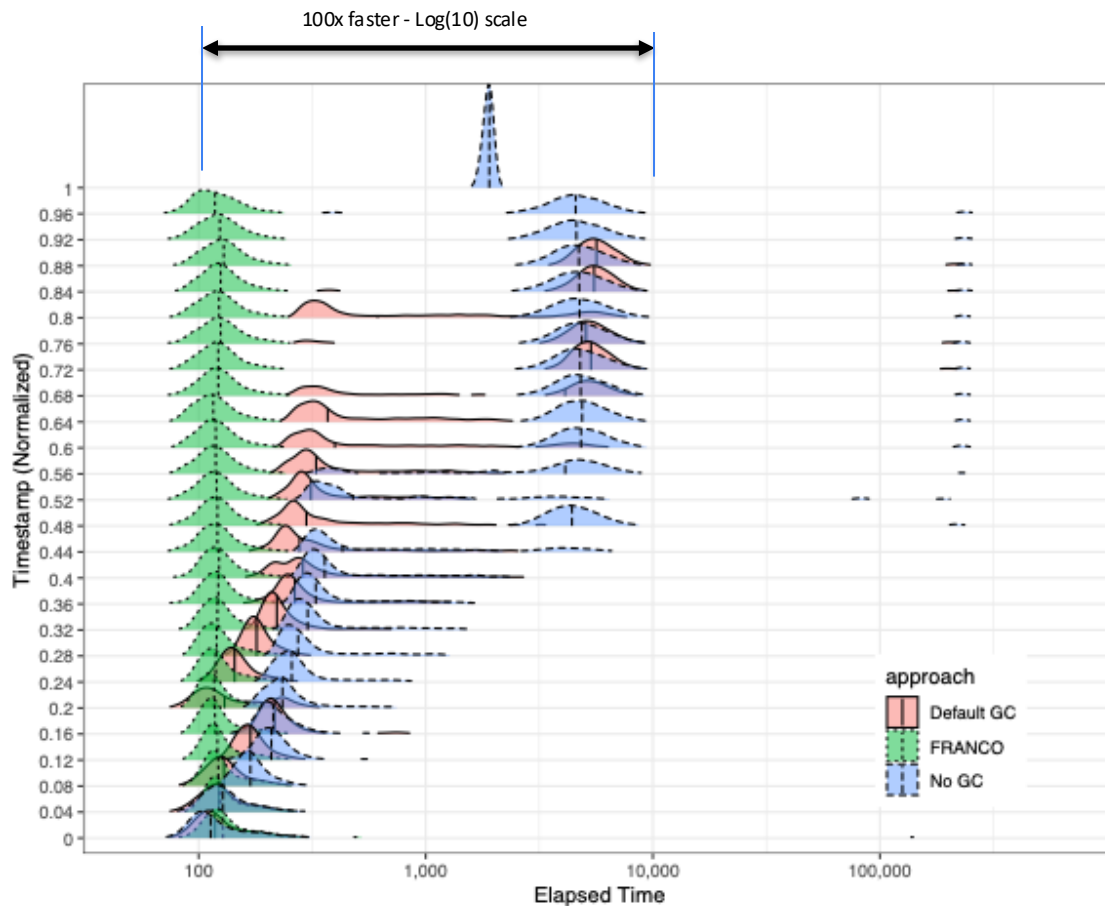
Cumulative
Execution times
(normalized)



GitLab: Results



Up to 100x Faster



GHS is running on GerritHub.io



Currently enabled on GerritForge's repositories

> To be extended to **all of the GitHub repos using Gerrit reviews** on GerritHub.io



A bright yellow background with a horizontal strip of white paper torn into the center. The paper has irregular, ragged edges, and the yellow background is visible on either side of the white strip. The text "WHAT'S NEXT?" is printed in a bold, black, sans-serif font on the white paper.

WHAT'S NEXT?

GHS test drive with **GitHub** Enterprise



GitHub:Enterprise requires a lot more resources than Git standalone

- GHS helps reduce the system load
- Increase throughput by two orders of magnitude for use with CI/CD
- Scientific testing with benchmarks tailored to the **GitHub** API and PR workflow
- OpEx cost reduction by at least 50% for **GitHub** Enterprise on Cloud Hosting



GHS new research paper with GitLab and GitHub real-life use-cases



A word cloud featuring the phrase 'ANY QUESTIONS?' in large, bold, dark blue capital letters. Surrounding this central text are various question words in different colors and sizes, including 'WHEN?', 'WHAT?', 'WHERE?', 'HOW?', 'WHO?', 'WHY?', and 'WHERE?'. The words are arranged in a circular pattern, creating a dynamic and engaging visual effect.



Do you want your Gerrit / GitLab / GitHub running 100x faster?



<https://gerritforge.com/ghs>

info@gerritforge.com