Continuous Integration in SoC Design: Challenges and Solutions

Liu Wei, Li Jianjun, Yang Liangfeng, Li Liang
SDTECH Information Technology

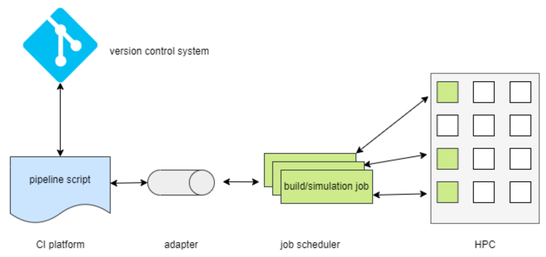
2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 24-27, 2025

INTRODUCTION

- Continuous Integration (CI): frequent integration of code changes into a shared repository with automated builds and tests
- CI has gradually been applied in the hardware development field, but implementing an efficient hardware continuous integration system still faces many challenges:
  - Incompatibility of HPC environments with mature CI tools, requiring additional dedicated computing resources, CI system with poor real-time performance and increased complexity
  - Deep integration of CI tools with Git, Requires many self-developed scripts in non-Git environments and cannot continuously follow the latest concepts and methods in the field of continuous integration, dedicated maintenance personnel
  - Performance degradation, binary file merge conflicts, and large file management challenges when using Git for chip design code

CI System Architecture for SOC design

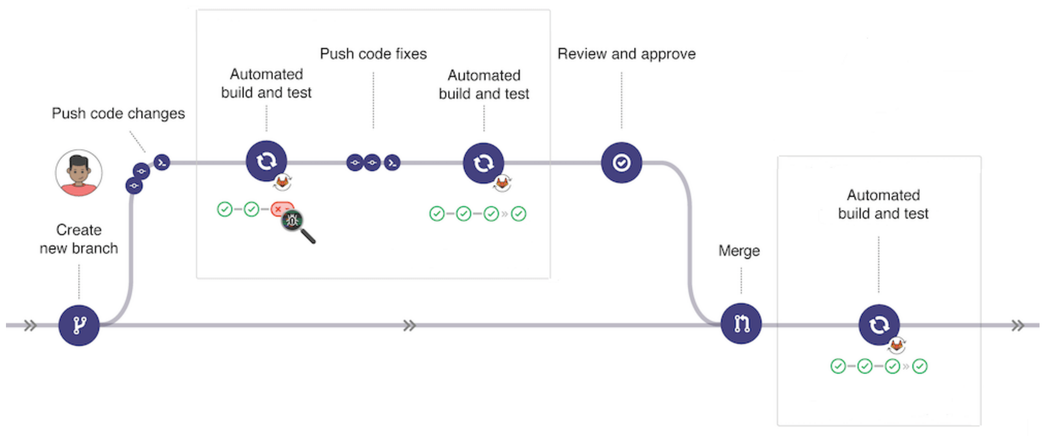
- Key Components of the CI System for SOC:



  - **Version Control System**, Stores chip design code, verification test cases, and configuration files, with Webhook trigger (e.g., GitLab, Bitbucket)
  - **CI Platform**: Provides pipeline scripts for executing tests, and monitors task status (e.g., GitLab CI, Jenkins)
  - **Job Scheduling System**: Manages and distributes parallel jobs to HPC (e.g., Slurm, LSF)
  - **Command Adapter**: Connects the CI system to the job scheduling system, translating CI platform commands to HPC commands (e.g., Jacamar CI)

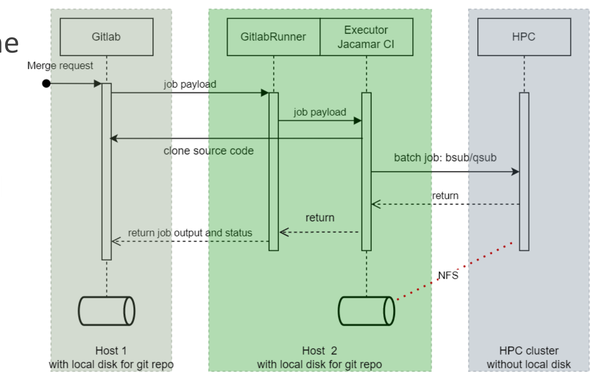
CI Process to Ensure Main Branch Availability

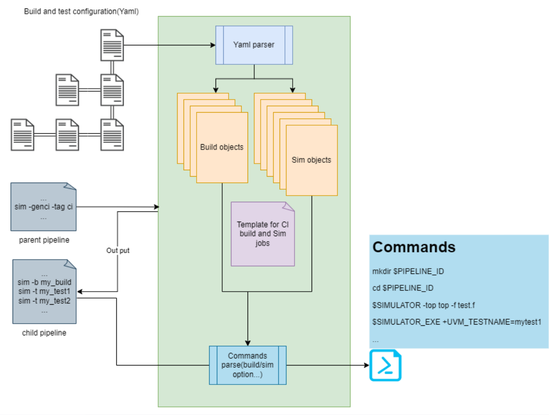
- Main objective: CI Process to Ensure Main Branch is Always Available
- Strategy: automatically run a set of tests before each code merge, only merging into the main line if all tests pass.
- CI tests cover multiple build and simulations at IP, subsystem, and SoC levels.



Implementation of the CI System Based on GitLab

- GitLab as the Git server, GitLab CI as the CI platform, and Jacamar CI as the adapter
- GitLab Runner clones code, Jacamar CI submits jobs to HPC
- Test results are returned to the GitLab page





- Enhanced existing regression scripts commonly used in chip design and verification for CI
- Use GitLab CI's parent-child pipeline mechanism
- A timed script executed for deleting old code and logs

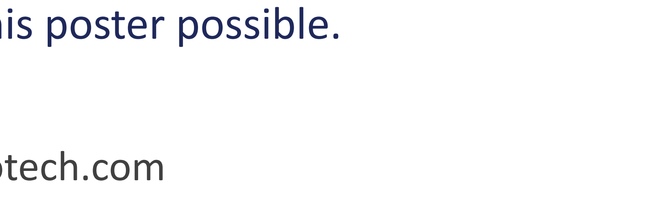
Git Challenges in SoC Projects and Resolutions

- Git requires the use of local hard drives, while the chip design industry extensively uses network drives for storage. Therefore, adjustments to the IT infrastructure are needed.
  - The Git server, CI runner and the user work server, have their own local storage for storing Git repositories locally.
  - the local storage of the CI runner and user work server needs to be shared to form the shared file system of the server cluster

- Use git scalar and sparse checkout to optimize performance. Need Git version 2.38 or later.
- Use Git LFS to manage large files.
- Use LFS locking to resolve binary file conflicts

Result and Conclusion

- Implementation in a real project:
  - 6Gb repository 41 developers
  - performed 3,303 merges
  - triggering 8,213 pipelines
  - with 6,743 successful completions and 1,344 failures.



the 1,344 early-caught failures represented over 3,360 hours of saved engineering time.

- Conclusion:
  - The current Git can already meet the SOC design verification requirements for code version management, including large repositories, large files, and binary files.
  - CI tools based on Git are highly mature, with many open-source tools readily available in the Git ecosystem, eliminating the need for extensive development. Only minimal effort is required to create a user-friendly, robust, and efficient CI system CI system, which requires no dedicated maintenance.

The authors would like to thank the creators of the open-source software we used, our design verification engineers and IT engineers for their cooperation, as well as the strong support from our boss, which made this poster possible.

数渡科技

Presenter: Liu Wei
Contact: liuwei@sudoinfotech.com