

Smart Pip (spip) - Parallel Matrix Testing Refactoring Summary

Antigravity (AI Coding Assistant)

January 30, 2026

1 Overview

The `spip matrix` command has been refactored to support high-performance parallel execution of package tests across multiple Python versions. This was achieved by introducing a thread pool architecture that ensures resource isolation and thread safety.

2 Key Achievements

1. **High-Performance Parallelism:** Implemented a worker pool using `std::thread` and `std::hardware_concurrency` allowing many versions to be tested simultaneously.
2. **Atomic Task Distribution:** Used `std::atomic<size_t>` for thread-safe task indexing, ensuring each package version is processed exactly once without complex locking.
3. **Complete Resource Isolation:** Each test run utilizes its own isolated Git worktree (`git worktree add`) in a unique directory. This prevents interference between parallel tests.
4. **Concurrency Fixes for Shared Cache:** Resolved race conditions where multiple threads attempted to fetch the same metadata or wheel simultaneously. Implemented a robust "look-lock-look" pattern with `std::mutex` and atomic temporary file renames to ensure cache integrity.
5. **Thread-Safe Logging:** Console output is protected by `std::mutex` to prevent interleaved or scrambled log messages from different threads.
6. **Automated Bootstrapping:** Parallel-safe logic for bootstrapping base Python versions ensures stable environments are shared correctly across threads.
7. **Robust Cleanup:** Added sophisticated cleanup logic to prune worktrees and remove temporary directories after each thread finishes its task.

3 Verification Results

End-to-end testing with the `colorama` package (matrix of 16 versions) showed extreme stability. The system correctly handled parallel resolution, installation, and testing with a total execution time of approximately **36 seconds**. The concurrency fixes successfully prevented the "Could not find wheel URL" errors previously observed during simultaneous dependency resolution.

4 Conclusion

The `spip` tool is now capable of performing large-scale matrix testing at maximum CPU utilization, maintaining the strict isolation and reproducibility guarantees that define the project, while ensuring high reliability in multi-threaded environments.