

Media CI: Saving \$\$\$ & Holidays

How jobs are assigned, infrastructure tweaks, and keeping Hans and Mauro happy.

Act I: The "Good" Old Days

- ▶ Testing used to mean running **massive shell scripts** locally.
- ▶ Hans and Mauro utilizing beefy machines right before pushing to the media branch.
- ▶ The unmistakable smell of **burning silicon** in the air.
- ▶ The scripts lived in the local machine of the maintainers.
- ▶ *Does anyone miss those days?*



Enter GitLab



Modernize!

We decided to step into the future. Let's make it parallel! Let's use GitLab to automate everything.



Reality Check

We inherited GitLab's quirks (shoutout to shadow clones). But the pipelines live proudly at freedesktop.org.



Maintained

Scripts are maintained before each kernel release. Drivers are fixed with latest static tools. SLO is very good

How a Pipeline is Executed



1. Push

A **branch is pushed** to a project. This triggers GitLab to wake up and look for a **.gitlab-ci.yml** file.



2. Resolve

GitLab **checks the CI config** to map out the pipeline structure, stages, and dependencies.



3. Create

Jobs are created based on logic. **Eg:** Don't run **virtme** unless core/virtual drivers change. Don't build if basic checks fail.



4. Assign

GitLab finds a **free runner** and assigns the work. The runner then pulls the environment and starts testing.

Act II: Unoptimized cost

\$107k

USD per year

The Initial Bill 🤯

Our pipelines are highly parallel (5-6 jobs).

Multiple developers plus peak times meant we needed **100 machines** for a decent Quality of Service.

Each machine is beefy: **32 cores** and **128 GiB of RAM**.

The Hacks to Save Money

Hack 1: Spot Instances

Playing Russian Roulette: Spot machines are cheap because they can be killed at any time.

- ▶ Stateless jobs = we don't care!
- ▶ Allow 2 retries if preempted.
- ▶ **Savings:** Down to **\$48K**.

Hack 2: The Autoscaler

On-Demand Magic: Why keep machines running if no one is pushing code?

- ▶ Small server hosts an autoscaler.
- ▶ Instantiates a new server on demand.
- ▶ **Savings:** Down to **\$5K USD!** 🏆

The Jevons Paradox in Action

”

We will inevitably use as much resources as we have. With costs down, we extended Media CI to check every patch in the mailing list.

”

– Scaling to fill the void

Act III: Priority Drama

”

GitLab runners are dumb. I send 100 pipelines to improve checkpatch warnings. Hans has to change his holiday plans... Hans is not happy.

”

– The priority problem

The Multiple System for Runners

Only committers affect the workflow of other committers. They can take revenge on each other later.

Runner Type	Target Tree	Description
VIP	media-committers	Hans's fast track for urgent merges.
VIP nested	media-committers vivid	Nested environment for committers.
Big	users/	Standard pipeline for general developers.
Small	patchwork	Used for patchwork jobs.

Act IV: The AI Apocalypse

AI is Everywhere

- ▶ Capacity has become extremely scarce.
- ▶ Spot instance preemptions have increased significantly.

Counter-Measures

- ▶ **Multizone Support:** Egress fees went up, but we actually get servers now.
- ▶ **Guaranteed VIPs:** Non-spot, expensive servers reserved for the last week.



The Stable Tree Fight



Throttled

As Media CI increased its use, the stable tree started to throttle our automated requests.



The Solution

We created a new local database containing the patches that are in stable and backported.



Peace Restored

No more spamming the stable tree! Efficient, local checking keeps everyone happy.

Act V: The Future (Who Pays?)



The Funding & Team Currently funded by Google (\$5-10K budget). Work done mainly by two people (Hidenori and me). Time overflows into our free time. If funding stops, we need a new sponsor!



Report Issues Help us improve by actively reporting issues or bugs via IRC, GitLab, or the mailing list.



Review Patches Prioritize reviewing patches related to static analysis fixes to keep the codebase clean.

Questions?

Thank you for your time.