Make our CI feedback loop shorter

This is about the follow-up work to #15501 that we want to do in 2019-2020, i.e.:

1. Reconsider our options → ditched the "hacker option"
2. Get in touch with ProfitBricks, that donate crazy amounts of VM "hardware" to the Reproducible Builds project. If they're happy to give us some, this would be, by far, the simplest and cheapest option. AFAIK that's a simple "here's a VM, you're root" setup so it does not have the drawbacks of more powerful & complex cloud systems, apart from the fact we don't run the hardware ourselves.
3. If ProfitBricks is happy to provide what we need: discuss on -summit@ if we want that (mostly a political decision), keeping this in mind: what happens if they suddenly pull the plug?
4. Wait for HPE's answer.
5. If HPE does not sponsor us, request Tails budget for the "Bare metal server dedicated to CI" option.
6. Implement the chosen solution.

Histories:

#1 - 08/09/2019 05:02 PM - intrigeri
- Blocks Feature #13284: Core work: Sysadmin (Adapt our infrastructure) added

#2 - 08/09/2019 05:02 PM - intrigeri
- Due date set to 04/09/2018
- Start date set to 04/09/2018
- Follows Feature #15501: Server hardware (2017-2019 edition): evaluate some of the options added

#3 - 08/09/2019 05:02 PM - intrigeri
- Related to Bug #11680: Upgrade server hardware (2017-2019 edition) added

#4 - 08/09/2019 05:04 PM - intrigeri
- Description updated
- Due date deleted (04/09/2018)
since we had to cancel our last meeting on this topic, I've taken some steps to keep the ball rolling.

I've updated the blueprint: specs, cost estimates (based on the work we did together on a pad last month), pros & cons.

First, as a baseline, the total budget estimate we had for this project was 8250€. But our more recent estimates (sticking with the hacker option™) are around 10.5k€.

I'm increasingly convinced that the hacker option™, while a very cool idea, has 2 major drawbacks IMO:

- It would eat lots of our limited time, and:
  - That time would be better put in places that require human brains that can't be replaced by $€¥.
  - The whole thing will get postponed to whenever we have that time available, i.e. not any time soon, I bet. Meanwhile, developers and release managers suffer from our slow CI, and the general atmosphere in turn suffers.
- It would be another instance of static resources allocation (in this case, between 4 nodes). My not-so-secret plan is to make our CI work in a different way at some point: that would not require 1 dedicated VM per Jenkins node, but instead run all CI stuff in one single VM, and it would thus benefit from dynamically allocated resources, i.e. when only one job is running at a given time, it is much faster; and if there are more jobs running concurrently, well, they share resources (but likely they don't need the exact same kind of resources all at the same time). It would make me sad to purchase hardware that commits us to the current static resources allocation scheme for the next 5+ years.

I'm under the impression that you felt the same way during our meeting last month wrt. the "eat lots of our limited time" aspect. So perhaps it's time to officially ditch the hacker option™, and focus on the other ones?

Furthermore, it is my understanding that we already ditched the cloud option: nobody on our team is knowledgeable nor super excited about this sort of stuff at this point, and here again, it would take a while until we learn.

Assuming we agree on this, the 2 remaining options are:

- Free VMs at ProfitBricks: I understand you were not overly excited at the idea of running stuff on hardware we don't control, hosted by a for-profit company; I'm not either. But it has the potential to save us 15k€ + 50€/month, and if instead we have to pay these costs, well, we'll have to find the corresponding money somehow, and it might not come from sources we find super great either. That's why I find it hard to justify dismissing the idea entirely. I propose I tentatively ask ProfitBricks, so we know if this option is actually on the table; and then, we can talk (either within the sysadmin team, or in the broader Tails community, because it's in great part a political decision). Would this work for you?
- Bare metal server dedicated to CI: a few more thousand more € than planned, but it's mostly business as usual and requires rather little work. IMO that's the way we should go if for some reason, free VMs are not an option. That hardware expense would be significantly larger than budgeted so this would take a discussion on -summit@.

What do you think?

If you prefer to discuss this in a synchronous matter, let me know, and we'll schedule a meeting :)

05/17/2020
For this week's meeting:

groente & intrigeri half-sadly agree we should ditch the hacker option.

intrigeri will ask ProfitBricks if they can provide what we need; if they can, we'll discuss on -summit@ if we want that (mostly a political decision), keeping this in mind: what happens if they suddenly pull the plug?

intrigeri wrote:

intrigeri will ask ProfitBricks if they can provide what we need

Asked someone if they felt like sharing their contacts at ProfitBricks, or writing an introduction email.

I've emailed someone at ProfitBricks (likely not the best person but hopefully they can redirect me adequately :)

While I was at it, I started one last time to ask around me about contacts at HPE: who knows, HPE might be happy to donate hardware to us.

I'd like to see if HPE answers us within a month from now. If they don't, IMO we should proceed with the next steps towards the "Bare metal server dedicated to CI" option.

I'd like to see if HPE answers us within a month from now. If they don't, IMO we should proceed with the next steps towards the "Bare metal server dedicated to CI" option.
A month has passed and given the economic context, I strongly doubt HPE will be in the mood to sponsor hardware for us ⇒ I'm all for investigating further our other options, starting with https://tails.boum.org/blueprint/hardware_for_automated_tests_take3/#gamer-option

I've completed the part that I wanted to do and groente took the lead on this new proposal (gamer option), so I'm reassigning accordingly. The "Target version" was set for the work I just completed, so feel free to set it to whatever works for you.

At first glance, I find the gamer option very, very attractive!

To confirm it would suit our needs, besides the base clock numbers I can see already, I'd love to see benchmarks, for example comparing it to:

- 4 × Intel E-2134 (the one we got to benchmark the hacker option)
- 4 × Xeon Gold 5222 (datacenter-grade hardware)

Ideally, such benchmarks would include both single-threaded and multi-threaded performance info: some of our workload is bound to 1 single core, while some takes advantage of multiple CPUs.

---

#21 - 04/25/2020 12:00 PM - intrigeri

- File benchmarks.ods added

tl;dr: my main worry is solved and there's no need to block on it anymore :)

intrigeri wrote:

To confirm it would suit our needs, besides the base clock numbers I can see already, I'd love to see benchmarks, for example comparing it to:

- 4 × Intel E-2134 (the one we got to benchmark the hacker option)
- 4 × Xeon Gold 5222 (datacenter-grade hardware)

Done! What I saw suggests that 1 × Ryzen TR 3960X should perform:

- similarly to ant01’s E-2134, for single-threaded workloads (per-CPU performance is the bottleneck for a fair bit of the latency of 1 given CI job)
- probably better than 4 × E-2134, aka. the hacker option, for multi-threaded workloads, which matters for:
  - throughput of our CI as a whole (multiple jobs running at a time)
  - some parts of our CI jobs use multiple cores

Awesome :) Of course, one should take such benchmark results with a grain of salt, and only real-world testing will tell.

For the curious, I'm attaching the raw data.

Another requirement I just thought about: nested KVM virtualization.
While I hope we can get rid of one layer of virtualization at some point, we're not there yet, so in order to immediately benefit from this new machine, without blocking on lots of significant amounts of dev work, it needs to support nested KVM virtualization.
I've found signs on the web that suggests nested virt works on Ryzen TR 3960X for vmware, but did not find reports about KVM, so I guess we can only try ourselves.
- Target version changed from Tails_4.6 to Tails_4.7

Files

<table>
<thead>
<tr>
<th>Filename</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>benchmarks.ods</td>
<td>17.9 KB</td>
<td>04/25/2020</td>
<td>intrigeri</td>
</tr>
</tbody>
</table>