Tails - Feature #12002

Feature # 5630 (Resolved): Reproducible builds

Estimate hardware cost of reproducible builds in Jenkins

11/28/2016 08:00 AM - bertagaz

<table>
<thead>
<tr>
<th>Status:</th>
<th>Resolved</th>
<th>Start date:</th>
<th>11/28/2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>High</td>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
<td>% Done:</td>
<td>100%</td>
</tr>
<tr>
<td>Category:</td>
<td>Infrastructure</td>
<td>Estimated time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Target version:</td>
<td>Tails_3.1</td>
<td>Spent time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Feature Branch:</td>
<td>Research</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type of work:</td>
<td>Research</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blueprint:</td>
<td><a href="https://tails.boum.org/blueprint/reproducible_builds/hardware/">https://tails.boum.org/blueprint/reproducible_builds/hardware/</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

Adding reproducible builds in Jenkins will certainly require more hardware resources (disk space and RAM at least). Before deploying it for real, we need to estimate how much and take action if necessary.

Related Issues:

- Related to Tails - Feature #12576: Have Jenkins use basebox:clean_old instead... Resolved 05/22/2017
- Related to Tails - Bug #12595: Not enough space in /var/lib/jenkins on isobui... Resolved 05/25/2017
- Related to Tails - Bug #12599: /var/lib/libvirt/images gets filled on isobui... Resolved 05/25/2017
- Related to Tails - Bug #12725: Sort out the apt-snapshots-disk partition situ... Resolved 06/16/2017
- Related to Tails - Bug #13177: Sort out the bitcoin-disk partition situation ... Resolved 06/27/2017
- Blocks Tails - Feature #11806: Update server storage planning needs for at le... Resolved 09/19/2016

Associated revisions

Revision 8cc1ccba - 05/24/2017 07:20 AM - intrigeri

Bump the APT snapshots used by the Vagrant build box to the ones used in Tails 3.0~rc1.

This will allow me to drop the 2017051002 snapshots, that take disk space uselessly. We're running short of disk space and are having a hard time estimating how much more we need (refs: #12002).

Revision d25114cd - 07/01/2017 01:21 PM - bertagaz

Add blueprint about reproducible builds hardware cost for lizard.

Refs: #12002

Revision 93fee2bf - 07/01/2017 01:28 PM - bertagaz

Fix few typos and add link to the main reproducible builds blueprint.

Refs: #12002

Revision d9993532 - 07/04/2017 08:40 AM - intrigeri

Reprobuilds hardware: improve TOC to clarify what things are about (refs: #12002).
Revision b5a4ad64 - 07/04/2017 08:41 AM - intrigeri
Reprobuilds hardware: move the "more isobuilders" discussion to a more suitable place, and drop premature conclusions (refs: #12002).

The previous reasoning assumed we would run 8 isobuilders with the exact same performance as our current 4 ones. I doubt it's the best thing to do, so I'm moving the problem definition to the place where we're designing our next hardware upgrade, and dropping these premature conclusions.

Revision 7250022b - 07/04/2017 08:43 AM - intrigeri
Reprobuilds hardware: make TOC more detailed (refs: #12002).

Revision 76e0523 - 07/04/2017 08:49 AM - intrigeri
Reprobuilds hardware: fix calculations (refs: #12002).

Revision ef97690a - 07/04/2017 08:50 AM - intrigeri
Reprobuilds hardware: clarify (refs: #12002).

Revision 8dda1386 - 07/04/2017 09:04 AM - intrigeri
Reprobuilds hardware: improve the section about memory (refs: #12002).

Revision 444f8782 - 07/04/2017 09:13 AM - intrigeri
Reprobuilds hardware: add a sub-total (refs: #12002).

Revision a436a64e - 07/05/2017 09:20 AM - bertagaz
Add time-based snapshots estimate to the reproducible build hardware blueprint.

Refs: #12002

Revision 945027e0 - 07/05/2017 10:08 AM - intrigeri
Release process: don't keep time-based APT snapshots longer than needed.

We had switched (commit:68eb4074f6d31da641db5e86b7633e407a6f084) to 6 months as we thought we would manually update the basebox ourselves. This doesn't make sense anymore: we have the basebox updated automatically every time we prepare a new major release.

This reverts commit:68eb4074f6d31da641db5e86b7633e407a6f084 and improves the phrasing of the re-introduced text.

refs: #12002

History
#1 - 04/05/2017 01:33 PM - bertagaz
- Status changed from Confirmed to In Progress
Let assume that a basebox size is around 300M (higher than actually, but better this way). Let also assume that per year we have a delay where we host 2 baseboxes (when we update it), and maybe one other when we need to change the build system (as a safety in the estimation). Does it seem sound?

Based on that, a first partial estimation would be:

- 3 x 300M -> 1G
- 20G per isobuilder for the basebox build process -> 80G

So let say 100G in total for the baseboxes hosting/building.

Now we have to consider that we need to keep the same amount of partial APT snapshots than baseboxes. For now we don't really have a way to estimate how much disk space it means, as we have no mechanism in the basebox build system to have a list of Debian packages needed for that. I'll fill a ticket about that. In the meantime, I'll do a basebox build with an empty APT cache so that we get an idea.

---

#2 - 04/05/2017 05:19 PM - intrigeri

Let also assume that per year we have a delay where we host 2 baseboxes (when we update it), and maybe one other when we need to change the build system (as a safety in the estimation). Does it seem sound?

This seems to implicitly assume at least:

- that we're going to host baseboxes in some central place (unclear given #12409)
- that we're going to bother garbage collecting old baseboxes: IIRC last time we discussed this, our conclusion was that it was not worth the effort

I suggest you write down your assumptions explicitly, so that we can ensure we update the estimates if/when these assumptions change, and more generally if I'll help understanding where our conclusions come from.

Now we have to consider that we need to keep the same amount of partial APT snapshots than baseboxes. For now we don't really have a way to estimate how much disk space it means, as we have no mechanism in the basebox build system to have a list of Debian packages needed for that. I'll fill a ticket about that. In the meantime, I'll do a basebox build with an empty APT cache so that we get an idea.

It seems that whey you write "partial APT snapshots", you mean "tagged APT snapshots" (given you're discussing the list of needed Debian packages). Now I'm lost: do we have plans to actually generate such snapshots? I see no such thing on the blueprint, and I don't recall any discussion about it. The blueprint says we're going to use frozen (but not tagged) APT snapshots, and that we'll keep them around for 6 months. So if I'm remembering + reading things right, generating a list of packages installed when creating a basebox won't teach us anything useful. Please clarify if I got some of this wrong.
intrigeri wrote:

This seems to implicitly assume at least:

- that we're going to host baseboxes in some central place (unclear given #12409)

Right, so that's:

- \(4 \times 3 \times 300\text{M} \rightarrow 4\text{G}\)
- \(4 \times 20\text{G} \rightarrow 80\text{G}\)

\(\rightarrow\) still roughly around 100G

- that we're going to bother garbage collecting old baseboxes: IIRC last time we discussed this, our conclusion was that it was not worth the effort

Well, in #12409#note-15, we're discussing of a rake basebox:clean_old option about which you ask if why we can't run it every build. Unless I'm confused that's precisely its role.

Not that I absolutely want it to be implemented, but I'm wondering: once we have that, our beloved sysadmins won't have to regularly grow the libvirt partition of the isobuilders \(\rightarrow\) less maintenance for us. It's also useful for people building Tails not to get their libvirt partition bloated with old baseboxes. So unless that's very costly, I see some advantages to have that.

I suggest you write down your assumptions explicitly, so that we can ensure we update the estimates if/when these assumptions change, and more generally it'll help understanding where our conclusions come from.

It seems that whey you write "partial APT snapshots", you mean "tagged APT snapshots" (given you're discussing the list of needed Debian packages).

Yes.

Now I'm lost: do we have plans to actually generate such snapshots? I see no such thing on the blueprint, and I don't recall any discussion about it. The blueprint says we're going to use frozen (but not tagged) APT snapshots, and that we'll keep them around for 6 months. So if I'm remembering + reading things right, generating a list of packages installed when creating a basebox won't teach us anything useful. Please clarify if I got some of this wrong.

There hasn't been such a formal discussion. I just remember a reaction of yours when we quickly realized we forgot to count the APT snapshot, and that a frozen one is quite huge in term of disk space. So I've been bold and thought that maybe we'll want a tagged snapshot (hence the creation of a ticket I mention about generating a manifest of the list of Debian packages). Now I don't specially want it, so if you think that's overkill, I'm fine.
Hi!

intrigeri wrote:

- that we're going to bother garbage collecting old baseboxes: IIRC last time we discussed this, our conclusion was that it was not worth the effort

Well, in [#12409#note-15](#12409#note-15), we're discussing of a rake basebox:clean_old option about which you ask if why we can't run it every build. Unless I'm confused that's precisely its role.

This second comment of mine was based on your previous implicit assumption that "we're going to host baseboxes in some central place": if we did that, then your numbers work only if we garbage collect baseboxes stored in that central place, which is not what rake basebox:clean_old is about, so it doesn't come for free. But you implicitly dropped this assumption in the comment of yours I'm replying to (and I agree), so my comment is now obsolete :)

Not that I absolutely want it to be implemented, but I'm wondering: once we have that, our beloved sysadmins won't have to regulary grow the libvirt partition of the isobuilders -> less maintenance for us. It's also useful for people building Tails not to get their libvirt partition bloated with old baseboxes. So unless that's very costly, I see some advantages to have that.

Absolutely, as agreed on [#12409](#12409) already.

Now I'm lost: do we have plans to actually generate such snapshots? I see no such thing on the blueprint, and I don't recall any discussion about it. The blueprint says we're going to use frozen (but not tagged) APT snapshots, and that we'll keep them around for 6 months. So if I'm remembering + reading things right, generating a list of packages installed when creating a basebox won't teach us anything useful. Please clarify if I got some of this wrong.

There hasn't been such a formal discussion. I just remember a reaction of yours when we quickly realized we forgot to count the APT snapshot, and that a frozen one is quite huge in term of disk space.

This does ring a bell, indeed :)”

So I've been bold and thought that maybe we'll want a tagged snapshot (hence the creation of a ticket I mention about generating a manifest of the list of Debian packages). Now I don't specially want it, so if you think that's overkill, I'm fine.

Well, it's not really a matter of what I want or not, and sadly my guessing skills are limited :)”

IMO this discussion is premature as we have no data that shows us that there's a real problem to solve, especially given the solution to this potential problem can be quite hard to get right, and can have a number of drawbacks.

To get you started: assuming the only frozen APT sources we need to build/upgrade a basebox are Debian stable + backports, then keeping a 6-months-old time-based snapshot costs us (in terms of storage) the total size of packages that were in Debian 6 months ago, and are not anymore, because we store the common ones and the newer ones already anyway. I see at least two ways to get this data:

- simply try: bump by N months Valid-Until for the oldest time-based snapshot of the debian repository we have, and monitor the disk space that's saved when this snapshot is automatically garbage collected

  - pros: gives us accurate data, modulo Debian is frozen so there's less upload churn than usual

  - cons: we need to wait a bit (the oldest snapshot we currently have is only 3 months old), and to be very reactive when it's time to gather the data (or add some cronjob to store disk usage information somewhere); thankfully we don't need this data so urgently, and I suspect that you can be plenty busy with your other reproducible builds tasks until the N months have passed

- do some clever computation, either using reprepro dumpreferences (same as above, gives us data that's as relevant as the age of our oldest snapshot), or using actual APT indices (e.g. from snapshots.d.o so we can immediately tell what it would cost us today to store a 6 months old snapshot, and we can even tell the same for some time in the past outside of a Debian freeze); in both cases some relatively simple scripting is required
Once we have this data, we can decide whether it's worth doing tagged snapshots or not: doing it would require spending quite some additional developer's time, and would either add some complexity to the basebox building & upgrading (having different sources depending on whether it uses a tagged or time-based set of snapshots), or add some painful limitations and lack of flexibility (if we only support tagged snapshots). I suspect that storage (time-based snapshots) will be cheaper than developers' time (tagged snapshots), but as I said it's only a guess and not relevant.

Now that the expectations have been clarified, I'm going to shut up (unless you need more info from me) and let you do your job :)

#5 - 04/18/2017 03:30 PM - intrigeri
- Description updated
- QA Check deleted (Info Needed)

(Added RAM to the list of things to evaluate, as you raised this on another ticket.)

#6 - 05/24/2017 07:00 AM - intrigeri
- Blocked by Bug #12574: isobuilders system_disks check keeps switching between OK and WARNING since the switch to Vagrant added

#7 - 05/24/2017 07:01 AM - intrigeri
- Blocked by Feature #12576: Have Jenkins use basebox:clean_old instead of basebox:clean_all added

#8 - 05/24/2017 07:04 AM - intrigeri
- Blocks Feature #11806: Update server storage planning needs for at least 2017 added

#9 - 05/25/2017 06:59 AM - intrigeri
- Blocked by Bug #12595: Not enough space in /var/lib/jenkins on isobuilders added

#10 - 05/27/2017 09:00 AM - intrigeri
- Blocked by deleted (Feature #12576: Have Jenkins use basebox:clean_old instead of basebox:clean_all)

#11 - 05/27/2017 09:00 AM - intrigeri
- Blocked by Feature #12576: Have Jenkins use basebox:clean_old instead of basebox:clean_all added

#12 - 06/01/2017 06:39 AM - intrigeri
- Blocked by deleted (Bug #12595: Not enough space in /var/lib/jenkins on isobuilders)

#13 - 06/01/2017 06:41 AM - intrigeri
- Blocked by deleted (Feature #12576: Have Jenkins use basebox:clean_old instead of basebox:clean_all)

#14 - 06/01/2017 06:41 AM - intrigeri
- Related to Feature #12576: Have Jenkins use basebox:clean_old instead of basebox:clean_all added

#15 - 06/01/2017 06:41 AM - intrigeri
- Related to Bug #12595: Not enough space in /var/lib/jenkins on isobuilders added

#16 - 06/01/2017 06:42 AM - intrigeri
- Related to Bug #12599: /var/lib/libvirt/images gets filled on isobuilders added

#17 - 06/01/2017 06:43 AM - intrigeri

(Relaxed relationship with related tickets: the research leading to the numbers we need might happen here, or on other tickets that are about specific issues. If the former, then this ticket would block the others; if the latter, then it's the opposite. So let's stick to "Related to" for now.)
(Our deadline is before the end of the year.)

Wrt. the storage space needed for APT snapshots used by the build boxes, I think the problem has been greatly simplified now that updating the basebox is part of our release process, and we use the same snapshots as in the ISO.

I've made another estimation, now that things have settled a bit. There's still some things to discuss/evaluate. I've grown a bit the numbers compared to what we have to get some room. Please have a first look for numbers that are already known, if they make sense for you, and if you have inputs for the remaining open questions.

## Disk space

### root

Was: 4G

Add 500M to have some margin for system upgrades. Won't grow that much in the future. Only for Buster upgrade, which may be in quite a long time if we use Stretch LTS.

\[5G (\pm 1G \times 4)\]

### /var/lib/jenkins

Was: 6G

13 baseboxes 13 * 1.5G \implies 20G
artifacts 5G
1 basebox build 25G

\[50G (\pm 44G \times 4)\]

h3. /var/lib/libvirt/images

Was: none

1 basebox 1.5G
1 snapshot 2G

\[5G (\pm 5G \times 4)\]

### Artifacts

Reproducible builds will add one ISO each time a build fails. Difficult to guess how often it will happen. Let's consider 50% of the time worst case?

\[-\text{Count number of base branches artifacts we keep now and multiply}.\]

### time-based APT snapshots

Probably one or two to keep sometimes when we update the basebox in the middle
of the 4 months. Size unknown for now, would need evaluation.

**Memory**

Current: 14.5G * 4

We did not bump it a lot in the past, biggest one was due to Vagrant itself, but it should stabilize in the future. Maybe count 20G to have some margin still?

**CPUs**

We use 4 per isobuilders.

Projecting if it will grow depends on the question below.

**Will we need more isobuilders?**

With the reproducibles builds we may have a lower output time, meaning we may want to add more isobuilders in the future if it's getting too slow.

-> Need to evaluate how much time it may delay the output we have, meaning having a look to the number of base branches builds in the past, or maybe assume we'll want N more?

1 more isobuilder requires:

- 4 CPUs
- 14.5G RAM
- 60G HDD

#22 - 06/05/2017 12:20 PM - bertagaz

I've adapted the above numbers with what was decided on #12574#note-6 regarding isobuilders root partition.

#23 - 06/05/2017 12:27 PM - bertagaz

bertagaz wrote:

**Artifacts**

Reproducible builds will add one ISO each time a build fails. Difficult to
guess how often it will happen. Let's consider 50% of the time worst case?

-> Count number of base branches artifacts we keep now and multiply.

Will we need more isobuilders?

-> Need to evaluate how much time it may delay the output we have, meaning having a look to the number of base branches builds in the past, or maybe assume we'll want N more?

So here are the number of base branches builds (stable + devel + testing + feature/stretch) in the past:

<table>
<thead>
<tr>
<th>month</th>
<th>base_branches</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-02:</td>
<td>86</td>
<td>263</td>
</tr>
<tr>
<td>2015-03:</td>
<td>134</td>
<td>358</td>
</tr>
<tr>
<td>2015-04:</td>
<td>92</td>
<td>351</td>
</tr>
<tr>
<td>2015-05:</td>
<td>147</td>
<td>268</td>
</tr>
<tr>
<td>2015-06:</td>
<td>91</td>
<td>248</td>
</tr>
<tr>
<td>2015-07:</td>
<td>50</td>
<td>101</td>
</tr>
<tr>
<td>2015-08:</td>
<td>92</td>
<td>195</td>
</tr>
<tr>
<td>2015-09:</td>
<td>87</td>
<td>470</td>
</tr>
<tr>
<td>2015-10:</td>
<td>106</td>
<td>935</td>
</tr>
<tr>
<td>2015-11:</td>
<td>105</td>
<td>611</td>
</tr>
<tr>
<td>2015-12:</td>
<td>117</td>
<td>809</td>
</tr>
<tr>
<td>2016-01:</td>
<td>154</td>
<td>757</td>
</tr>
<tr>
<td>2016-02:</td>
<td>126</td>
<td>603</td>
</tr>
<tr>
<td>2016-03:</td>
<td>89</td>
<td>839</td>
</tr>
<tr>
<td>2016-04:</td>
<td>85</td>
<td>779</td>
</tr>
<tr>
<td>2016-05:</td>
<td>143</td>
<td>1055</td>
</tr>
<tr>
<td>2016-06:</td>
<td>146</td>
<td>1022</td>
</tr>
<tr>
<td>2016-07:</td>
<td>126</td>
<td>1178</td>
</tr>
<tr>
<td>2016-08:</td>
<td>128</td>
<td>658</td>
</tr>
<tr>
<td>2016-09:</td>
<td>141</td>
<td>404</td>
</tr>
<tr>
<td>2016-10:</td>
<td>147</td>
<td>575</td>
</tr>
<tr>
<td>2016-11:</td>
<td>145</td>
<td>482</td>
</tr>
<tr>
<td>2016-12:</td>
<td>151</td>
<td>476</td>
</tr>
<tr>
<td>2017-01:</td>
<td>172</td>
<td>625</td>
</tr>
<tr>
<td>2017-02:</td>
<td>159</td>
<td>590</td>
</tr>
<tr>
<td>2017-03:</td>
<td>239</td>
<td>674</td>
</tr>
<tr>
<td>2017-04:</td>
<td>217</td>
<td>638</td>
</tr>
<tr>
<td>2017-05:</td>
<td>161</td>
<td>478</td>
</tr>
</tbody>
</table>

Some datapoints:

- Tails 2.0 was released end of 2016 January
- reproducible build jobs (at least for feature/stretch) were added end of 2016 November.
- porting to Stretch started on 2016 August
- the high peaks during 2016 May to July were mostly due to a lot of work happening in the test suite side, tagging a lot of them as fragile and fixing a lot of them.
artifacts 5G

What's this about?

Artifacts

Reproducible builds will add one ISO each time a build fails.

I'll assume below you instead mean "each time a build is not reproducible". Correct? Otherwise, please clarify as I don't get it.

Difficult to guess how often it will happen. Let's consider 50% of the time worst case?

I seriously hope that we won't be breaking reproducible builds that commonly. Let's say 20%?

-> Count number of base branches artifacts we keep now and multiply.

This seems to be based on the implicit assumption that we're going to do reproducibility testing on all active branches: I agree we should, because we should not merge branches that break reproducibility (once our ISO is reproducible); but I don't remember any ticket about this. Please create one if we have none, so we base on hardware cost estimates on actual plans :)

Once this is clarified, I agree with the proposed way of calculating this.

time-based APT snapshots

Probably one or two to keep sometimes when we update the basebox in the middle of the 4 months. Size unknown for now, would need evaluation.

Err, we currently keep the snapshots used for major releases for 6 months, not 4:

6. Make it so the time-based APT repository snapshots are kept around long enough, by bumping their `Valid-Until` to 6 months from now:
   [[APT_repository/time-based_snapshots#bump-expiration-date-for-all-snapshots]]

Please check if that's still relevant, or if someone forgot to update release_process.mdwn when we decided to keep baseboxes for 4 months only.

Memory

Current: 14.5G * 4
We did not bump it a lot in the past, biggest one was due to Vagrant itself,

Here, we need to account for the additional memory we already had to allocate when switching to Vagrant: it's been taken on our spare reserve and has already forced us to do some trade-offs, so it doesn't come for free.

but it should stabilize in the future.

I wonder what makes you think so. Every time we work on porting Tails to a new version of Debian, we have to bump the amount of RAM needed for in-memory builds. I suggest you look at this growth rate and take it into account.

**Will we need more isobuilders?**

With the reproducibles builds we may have a lower output time,

I guess this again implicitly relies on the assumption that we're going to do reproducibility testing on all active branches. Correct?

With this assumption in mind, well, "may" seems half-assed: we're simply gonna build twice as many ISO images, so it'll definitely make the feedback loop longer.

-> Need to evaluate how much time it may delay the output we have, meaning having a look to the number of base branches builds in the past, or maybe assume we'll want N more?

Do you mean "active" branches instead of "base" branches?

Anyway, I think we can assume that building twice as many ISO images without affecting the developer/RM experience too much roughly requires doubling the throughput of our CI for ISO builds. This can be done in various ways, by making builds faster and/or having more isobuilders. I propose you simply add this info to the blueprint for [#11680](#11680), and we don't bother trying to find a more precise estimate here about the needed throughput increase. OK?
So here are the number of base branches builds (stable + devel + testing + feature/stretch) in the past:

I'm sorry I don't understand how it's relevant here. Can you please clarify?

This being said, great job, congrats!

Also, it would be nice if this was done during the 3.1 cycle, if you can: this ticket is blocking #11806 that has become somewhat urgent while we've allocated space needed here and there... including to handle the unplanned needs of the reproducible builds system. I didn't look closely, but I think we won't have enough space anymore to grow storage volumes to match our needs in a couple months, and I would dislike having to delete more isotesters/isobuilders again.

intrigeri wrote:

/var/lib/jenkins

artifacts 5G

What's this about?

We need disk space in this partition to retrieve the two ISOs.

Now I agree that with the disk space we have to build the basebox, it should be enough without adding more for the artifacts. So let's forget this.

Artifacts

Reproducible builds will add one ISO each time a build fails.
I'll assume below you instead mean "each time a build is not reproducible". Correct? Otherwise, please clarify as I don't get it.

Yes, that's what I meant.

Difficult to guess how often it will happen. Let's consider 50% of the time worst case?

I seriously hope that we won't be breaking reproducible builds that commonly. Let's say 20%?

As you wish. I don't know how to estimate this right now.

-> Count number of base branches artifacts we keep now and multiply.

This seems to be based on the implicit assumption that we're going to do reproducibility testing on all active branches: I agree we should, because we should not merge branches that break reproducibility (once our ISO is reproducible); but I don't remember any ticket about this. Please create one if we have none, so we base on hardware cost estimates on actual plans :)

Ok, in my mind, the assumption was: we try to reproduce base branches only, as we've discussed to set up reproducible jobs for this branches only for now. I've created #12715 to discuss that and decide something.

Once this is clarified, I agree with the proposed way of calculating this.

**time-based APT snapshots**

Probably one or two to keep sometimes when we update the basebox in the middle of the 4 months. Size unknown for now, would need evaluation.

Err, we currently keep the snapshots used for major releases for 6 months, not 4:

[...]

Please check if that's still relevant, or if $someone forgot to update release_process.mdwn when we decided to keep baseboxes for 4 months only.

Right, I think the release process is not up to date, as we've decided since then to bump this snapshots at every releases, so we though 4 months were enough.

**Memory**

Current: 14.5G * 4

We did not bump it a lot in the past, biggest one was due to Vagrant itself,
Here, we need to account for the additional memory we already had to allocate when switching to Vagrant: it's been taken on our spare reserve and has already forced us to do some trade-offs, so it doesn't come for free.

but it should stabilize in the future.

I wonder what makes you think so. Every time we work on porting Tails to a new version of Debian, we have to bump the amount of RAM needed for in-memory builds. I suggest you look at this growth rate and take it into account.

Ack.

**Will we need more isobuilders?**

With the reproducibles builds we may have a lower output time,

I guess this again implicitly relies on the assumption that we're going to do reproducibility testing on all active branches. Correct?

Nop, I was assuming we'll reproduce base branches only.

Anyway, I think we can assume that building twice as many ISO images without affecting the developer/RM experience too much roughly requires doubling the throughput of our CI for ISO builds. This can be done in various ways, by making builds faster and/or having more isobuilders. I propose you simply add this info to the blueprint for [#11680](#11680), and we don't bother trying to find a more precise estimate here about the needed throughput increase. OK?

Let see how [#12715](#12715) goes.

So here are the number of base branches builds (stable + devel + testing + feature/stretch) in the past:

I'm sorry I don't understand how it's relevant here. Can you please clarify?

That's because I was considering reproduce base branches only.

Also, it would be nice if this was done during the 3.1 cycle, if you can: this ticket is blocking [#11806](#11806) that has become somewhat urgent while we've allocated space needed here and there… including to handle the unplanned needs of the reproducible builds system. I didn't look closely, but I think we won't have enough space anymore to grow storage volumes to match our needs in a couple months, and I would dislike having to delete more isotesters/isobuilders again.

That was my intent.
intrigeri wrote:

Artifacts

Reproducible builds will add one ISO each time a build fails.

I'll assume below you instead mean "each time a build is not reproducible". Correct? Otherwise, please clarify as I don't get it.

Yes, that's what I meant.

OK, good. Please update this in the proposal once it's been moved to a blueprint.

Difficult to guess how often it will happen. Let's consider 50% of the time worst case?

I seriously hope that we won't be breaking reproducible builds that commonly. Let's say 20%?

As you wish. I don't know how to estimate this right now.

If we wanted to do any kind of serious estimate, we could start by looking at the history of the #5630 branch builds and see how often we've broken it by merging other branches. But I doubt it's worth the hassle at this point.
time-based APT snapshots

Probably one or two to keep sometimes when we update the basebox in the middle of the 4 months. Size unknown for now, would need evaluation.

Err, we currently keep the snapshots used for major releases for 6 months, not 4:

[...]

Please check if that’s still relevant, or if $someone forgot to update release_process.mdwn when we decided to keep baseboxes for 4 months only.

Right, I think the release process is not up to date, as we’ve decided since then to bump this snapshots at every releases, so we though 4 months were enough.

Then please ensure this is fixed => ticket++

Memory

Current: 14.5G * 4

We did not bump it a lot in the past, biggest one was due to Vagrant itself.

Here, we need to account for the additional memory we already had to allocate when switching to Vagrant: it’s been taken on our spare reserve and has already forced us to do some trade-offs, so it doesn’t come for free.

but it should stabilize in the future.

I wonder what makes you think so. Every time we work on porting Tails to a new version of Debian, we have to bump the amount of RAM needed for in-memory builds. I suggest you look at this growth rate and take it into account.

Ack.

I’ll let you update this proposal on the blueprint then :) 

Also, it would be nice if this was done during the 3.1 cycle, if you can: this ticket is blocking #11806 that has become somewhat urgent while we’ve allocated space needed here and there… including to handle the unplanned needs of the reproducible builds system. I didn’t look closely, but I think we won’t have enough space anymore to grow storage volumes to match our needs in a couple months, and I would dislike having to delete more isotesters/isobuilders again.

That was my intent.
Great!

#30 - 06/25/2017 11:43 AM - intrigeri
- Related to Bug #12725: Sort out the apt-snapshots-disk partition situation on apt.lizard added

#31 - 06/25/2017 11:56 AM - intrigeri
- Priority changed from Normal to High
- Target version changed from Tails_3.2 to Tails_3.1

We're short on disk space on several partitions, and can't grow them as much as planned (#11806) due to the Vagrant thing having been deployed without taking the big picture of storage into account. So at least the storage aspect of this ticket has become urgent: the faster it's done, the earlier we can do #11806 and purchase the storage we need. Feel free to split storage/memory/CPU aspects into dedicated subtasks if you want to prioritize memory/CPU less high than storage.

#32 - 06/27/2017 09:48 AM - intrigeri
- Related to Bug #13177: Sort out the bitcoin-disk partition situation on lizard added

#33 - 07/01/2017 01:35 PM - bertagaz
- % Done changed from 10 to 20
- Blueprint set to https://tails.boum.org/blueprint/reproducible_builds/hardware/

intrigeri wrote:

We're short on disk space on several partitions, and can't grow them as much as planned (#11806) due to the Vagrant thing having been deployed without taking the big picture of storage into account. So at least the storage aspect of this ticket has become urgent: the faster it's done, the earlier we can do #11806 and purchase the storage we need. Feel free to split storage/memory/CPU aspects into dedicated subtasks if you want to prioritize memory/CPU less high than storage.

I've added everything into a blueprint with updates related to our discussions. On the disk side we're almost done. The remaining estimate to do is the APT snapshots size one. There's also an update and a pending question related to the memory part of the estimate.

#34 - 07/04/2017 09:44 AM - intrigeri
I've taken a look and improved the blueprint a bit, please have a look at my changes.

#35 - 07/04/2017 09:58 AM - intrigeri
- Blocked by deleted (Bug #12574: isobuilders system_disks check keeps switching between OK and WARNING since the switch to Vagrant)

#36 - 07/04/2017 12:04 PM - bertagaz
intrigeri wrote:

I've taken a look and improved the blueprint a bit, please have a look at my changes.
Looks good. So the only remaining is APT snapshots. I wonder how realistic it is to settle on 40G for one time-based snapshot given that's what the 2.12 one used (as shown in #12725)? If we do, then as we stated on the blueprint that would mean 4 * 40G we'd have to keep during two release cycles.

### #37 - 07/05/2017 08:41 AM - intrigeri

So the only remaining is APT snapshots. I wonder how realistic it is to settle on 40G for one time-based snapshot given that's what the 2.12 one used (as shown in #12725)?

Yes, but please apply to this number the ratio I've computed on #12111, and apply a 1.3 ratio of top of that to account for the growth of the Debian archive.

If we do, then as we stated on the blueprint that would mean 4 * 40G we'd have to keep during two release cycles.

The release says 3 extra snapshots, not 4, so perhaps I'm confused or not looking at the right place? Once this is clarified, this sounds good to me (I'm too lazy to find the reasoning behind this "3" number, which is not on the blueprint, but I'll trust you have copied the result correctly).

### #38 - 07/05/2017 09:23 AM - bertagaz

- Assignee changed from bertagaz to intrigeri
- % Done changed from 20 to 50
- QA Check changed from Dev Needed to Ready for QA

intrigeri wrote:

Yes, but please apply to this number the ratio I've computed on #12111, and apply a 1.3 ratio of top of that to account for the growth of the Debian archive.

Ack.

The release says 3 extra snapshots, not 4, so perhaps I'm confused or not looking at the right place?

Nop, I've made the mistake of adding the one we already keep, but that's not necessary as it's already taken into account. So you're right, 3 that is.

Once this is clarified, this sounds good to me (I'm too lazy to find the reasoning behind this "3" number, which is not on the blueprint, but I'll trust
you have copied the result correctly).

Updated the blueprint. I guess we're good here then, and we can go on with #11806.

#39 - 07/05/2017 10:12 AM - intrigeri
- Blocked by deleted (Feature #12715: Decide what builds we will try to reproduce in Jenkins)

#40 - 07/05/2017 10:13 AM - intrigeri
Deleted the "blocked by #12715" relationship (as the current estimates are about the "worst" case situation) so we can close this ticket.

#41 - 07/05/2017 10:16 AM - intrigeri
- Status changed from In Progress to Resolved
- QA Check changed from Ready for QA to Pass

The release says 3 extra snapshots, not 4, so perhaps I'm confused or not looking at the right place?

Nop, I've made the mistake of adding the one we already keep, but that's not necessary as it's already taken into account. So you're right, 3 that is.

OK, good.

Once this is clarified, this sounds good to me (I'm too lazy to find the reasoning behind this "3" number, which is not on the blueprint, but I'll trust you have copied the result correctly).

Updated the blueprint. I guess we're good here then,

Well, no: there was still the snapshots lifetime thing left to handle first, otherwise these estimates are pretty much off with the real world. I see no related ticket created since I've asked you to do so (in order to avoid forgetting this bit) so I assume this disappeared from your radar. Tracking this takes me more time / mental space / energy than handling it myself, so I went ahead: please review 945027a0d908a3120af873cd3f817f1c10c5a31.

and we can go on with #11806.

Yes :)
intrigeri wrote:

Well, no: there was still the snapshots lifetime thing left to handle first, otherwise these estimates are pretty much off with the real world. I see no related ticket created since I've asked you to do so (in order to avoid forgetting this bit) so I assume this disappeared from your radar. Tracking this takes me more time / mental space / energy than handling it myself, so I went ahead: please review 945027e0d908a3120af873cd3f817ff1c10e5a31.

Oooch, yes I forgot that. Looks good to me, thanks for the fix.