Hanging build results are showed on issue page when server restarts.

<table>
<thead>
<tr>
<th>Status:</th>
<th>新規(New)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>通常(Normal)</td>
</tr>
<tr>
<td>Assignee:</td>
<td>% Done: 0%</td>
</tr>
<tr>
<td>Category:</td>
<td>Hudson Plugin</td>
</tr>
<tr>
<td>Target version:</td>
<td>backlog</td>
</tr>
</tbody>
</table>

**Description**

**What steps will reproduce the problem?**

1. Install redmine 2.4.2
2. Configure redmine project repository.
3. Install and configure Redime Hudson plugin version 2.1.2
4. Set autofetch built details.
5. Restart redmine during hudson build is in progress, just before build is finished (I restarted apache).
6. Wait a while - to be ensured that autofetch job has been launched.

**What is the expected output? What do you see instead?**

I've got 'hanging' build results on issue page with text "built by <JobName> #228 at about 44 years ago".

And that's what I see in the database:

```sql
mysql> select * from hudson_builds;
+-----+---------------+---------+---------------------+----------+-------+-----------+--------+
| id  | hudson_job_id | result  | finished_at         | building | error | caused_by | number |
+-----+---------------+---------+---------------------+----------+-------+-----------+--------+
| 645 | 5             | SUCCESS | 2014-02-18 13:03:43 | false    |       | 1         | 204    |
| 646 | 7             | SUCCESS | 2014-02-18 13:03:43 | false    |       | 1         | 181    |
| 647 | 23            |         | 1970-01-01 03:00:00 |         |       | 1         | 228    |
| 648 | 9             | SUCCESS | 2014-02-18 18:17:00 | false    |       | 1         | 246    |
| 649 | 2             | SUCCESS | 2014-02-18 18:11:19 | false    |       | 1         | 357    |
+-----+---------------+---------+---------------------+----------+-------+-----------+--------+
```

**Please give us a result of following command.**

```bash
% ruby script/about

sh: darcs: not found
sh: cvs: not found
sh: bzr: not found
sh: git: not found

Environment:
Redmine version      2.4.2.stable
Ruby version         2.1.0-p0 (2013-12-25) [i686-linux]
Rails version        3.2.16
Environment          production
Database adapter     Mysql2
SCM:
Subversion           1.6.12
Mercurial            1.7.5
Filesystem
Redmine plugins:
  clipboard_image_paste 1.8
```

---

06/21/2020
Please provide any additional information below.

I fixed this on my system adding one more condition to HudsonJob.do_fetch? method:

diff -r -c -w /tmp/redmine_hudson/app/models/hudson_job.rb /usr/share/redmine/plugins/redmine_hudson/app/models/hudson_job.rb
*** /tmp/redmine_hudson/app/models/hudson_job.rb 2013-09-23 14:50:56.000000000 +0400
***************
*** 230,235 ****
--- 230,236 ----
+ latest_build = get_build(self.latest_build_number)
 return true if latest_build.is_a? HudsonNoBuild
 return true if latest_build.building?
 return true if !latest_build.finished_at || latest_build.finished_at == Time.at(0)
end

With patch provided above there is an issue with changeset duplication: changeset is added every fetch operation.

This fix it all...This fix it all...

diff -r -C 10 -w /tmp/redmine_hudson/app/models/hudson_job.rb /usr/share/redmine/plugins/redmine_hudson/app/models/hudson_job.rb
*** /tmp/redmine_hudson/app/models/hudson_job.rb 2013-09-23 14:50:56.000000000 +0400
***************
*** 207,242 ****
--- 207,241 ----
 build = get_build(build_number)
 if build.is_a?(HudsonNoBuild)
   build = new_build
 end

 # 0000000000
 ! build.add_changesets_from_xml buildelem if self.project.repository != nil

 # 00000000
 build.add_testresult_from_xml buildelem

 # 00000000
 build.add_artifact_from_xml buildelem
end
end
def do_fetch?
  latest_build = get_build(self.latest_build_number)
  return true if latest_build.is_a? HudsonNoBuild
  return true if latest_build.building?

  return false
end

def add_latest_build
  build = new_build
  build.number = self.latest_build_number
  build.result = ""
  build.finished_at = ""
  build.building = "true"
--- 207,244 ----
  build = get_build(build_number)
  if build.is_a?(HudsonNoBuild)
    build = new_build
  end

  build.update_by_api(buildelem)
  build.save

# 0000000000
! # Adding changesets only with last build update (when build.building changes to "false")
! build.add_changesets_from_xml buildelem if self.project.repository != nil && build.building === "false"

# 00000000
  build.add_testresult_from_xml buildelem

# 00000000
  build.add_artifact_from_xml buildelem
end

def do_fetch?
  latest_build = get_build(self.latest_build_number)
  return true if latest_build.is_a? HudsonNoBuild
  return true if latest_build.building?
+  return true if !latest_build.finished_at || latest_build.finished_at == Time.at(0)

  return false
end

def add_latest_build
  build = new_build
  build.number = self.latest_build_number
  build.result = ""
  build.finished_at = ""
  build.building = "true"