

RPM Support - Task #3528

Investigate which library to use for dep solving

03/27/2018 02:35 PM - ttereshc

Status:	CLOSED - COMPLETE	Start date:	
Priority:	Normal	Due date:	
Assignee:	milan	% Done:	0%
Category:		Estimated time:	0:00 hour
Sprint/Milestone:		Tags:	Pulp 2
Platform Release:		Sprint:	Sprint 38
Groomed:	Yes	Quarter:	
Sprint Candidate:	Yes		
Description			
<p>Current implementation of a dep solving supports only strong (Requires:) simple (package_name) dependencies. In order to support strong (Requires:) rich (aka boolean) dependencies an external library should be used. The suggestion is to drop the current implementation of dep solving and for any kind of a dep solving start using:</p> <ul style="list-style-type: none">- either libdnf via its Python interface (python2-dnf)- or libsolv via its Python bindings (python2-solv) <p>libdnf uses libsolv and exposes a higher level API to a user.</p> <p>Both libdnf and libsolv requires repodata to perform dep solving. Some temporary working directory should be provided to a solver to store its cache there.</p> <p>Functionality of a solver we are looking for:</p> <ul style="list-style-type: none">• list dependencies for a given rpm package (can be a part of a synchronous task)• list fulfilled dependencies (from a given list of available packages) for a given rpm package• validate that all the dependencies (from a given list of available packages) for a given rpm package are fulfilled.• perform dep solving against multiple repositories <p>The actions mentioned above will be a part of an asynchronous task (see a copy example below).</p> <p>An example of the dep solving use case in Pulp is a recursive flag during a copy: https://docs.pulpproject.org/plugins/pulp_rpm/user-guide/recipes.html?highlight=recursive#copy-errata-from-one-repository-to-another</p> <p>Dep solving related functionality to support: https://pulp.plan.io/issues/2478</p> <p>Example usage of the libsolv Python bindings: https://github.com/fedora-modularity/depchase Ping ttereshc for more details about examples for libdnf case.</p> <p>Figure out which library supports all required functionality. Keep in mind that performance is important as well. There could be repositories with 100K+ packages against which dep solving can be requested.</p>			
Related issues:			
Related to RPM Support - Issue #4152: Regression Pulp 2.17.1: recursive copy ...		CLOSED - CURRENTRELEASE	

History

#1 - 03/27/2018 02:39 PM - ttereshc

- Description updated

#2 - 03/28/2018 10:27 AM - ttereshc

- Description updated

#3 - 03/28/2018 02:31 PM - dkliban@redhat.com

- Groomed changed from No to Yes

- Sprint Candidate changed from No to Yes

Both the source and target repos will most likely have to be locked during the copy task run. Out of sync solver pools&repos might happen in case cached pools are being used and the copy task has been interrupted: there might be inconsistencies between Mongo and the solver pools.

#9 - 05/11/2018 11:02 PM - rchan

- Sprint changed from Sprint 36 to Sprint 37

#10 - 05/14/2018 03:29 PM - milan

Thinking out loud about the requirements from the description section.... I think overall a sat solver isn't a good fit for what the requirements are; the main theme of a sat solver being to give a yes (no) answer with a (counter-)proof i.e steps how (not) to install a package, all the requirements here imply some adjusting of the solver for a task it wasn't really meant to be used for.

Functionality of a solver we are looking for:

list dependencies for a given rpm package (can be a part of a synchronous task)

A solver works in a context of all available packages --- a content pool. Considering the transitive nature of content unit dependencies, to give an answer, the solver has to work in the context of all available packages from all relevant repos. The solver then returns a list of jobs to fulfill to successfully *install* a content unit, which may in turn contain *removals* of obsoleted, outdated or conflicting units.

In an empty pool, the list of dependencies of a package equals to the top-level Requires field of the package but it most certainly misses dependencies of the dependencies. Unless the Requires field dependencies are all leafs in the dependency tree of the package. Thus the answer is context-dependent and blurred by various content conflicts but maybe the question itself is the same.

list fulfilled dependencies (from a given list of available packages) for a given rpm package

In the context of a (multi-)source--target repository copy task, to give an answer, the sat solver has to be *tricked* to consider the target repository content to be *already installed*. This can be achieved by copying the target repo into the source repo Pool and marking all the units installed. An answer to an install request will then give a list of units to install. When compared to the list of requirements (from the previous bullet), one can obtain list of fulfilled dependencies. However the conflicts might blur the picture again.

validate that all the dependencies (from a given list of available packages) for a given rpm package are fulfilled.

Similar to the previous bullet; asserting the list of packages to install to be empty. Mind the conflicts though.

perform dep solving against multiple repositories

This is naturally supported by both the solvers; a pool can have multiple repos.

Note please that all the answers can be fine-tuned (and changed) by setting various solver options and/or the priorities of the pool repos. Maybe stating the justification of the requirements could give more context to understand why those are needed and figure out how (better) to address them.

#11 - 05/17/2018 02:46 PM - milan

- File *pulpsolv.py* added

Attaching a POC Pulp-fed depsolving thru direct libsolv calls.

Unfortunately it segfaults at the solving :-/

```
[vagrant@pulp2 tmp]$ python pulpsolv.py -h
usage: pulpsolv.py [-h] [--repo-name REPO_NAME] [--unit-name UNIT_NAME]
```

optional arguments:

```
-h, --help            show this help message and exit
--repo-name REPO_NAME
--unit-name UNIT_NAME
```

```
[vagrant@pulp2 tmp]$ python pulpsolv.py --repo-name zoo --unit-name lion
```

```
skipping erratum
skipping erratum
skipping erratum
skipping erratum
skipping package_category
skipping package_group
skipping package_group
loaded rpm: lion-0-0.4-1-noarch-sha256-2c5df6b51df167eb01246dce3049668cbe688033b9abff24464b0e05cdeb20ac
processing unit rpm: lion-0-0.4-1-noarch-sha256-2c5df6b51df167eb01246dce3049668cbe688033b9abff24464b0e05cdeb20
ac attribute name value lion
processing unit rpm: lion-0-0.4-1-noarch-sha256-2c5df6b51df167eb01246dce3049668cbe688033b9abff24464b0e05cdeb20
ac attribute epoch value 0
processing unit rpm: lion-0-0.4-1-noarch-sha256-2c5df6b51df167eb01246dce3049668cbe688033b9abff24464b0e05cdeb20
```

```

ac attribute version value 0.4
processing unit rpm: lion-0-0.4-1-noarch-sha256-2c5df6b51df167eb01246dce3049668cbe688033b9abff24464b0e05cdeb20
ac attribute release value 1
processing unit rpm: lion-0-0.4-1-noarch-sha256-2c5df6b51df167eb01246dce3049668cbe688033b9abff24464b0e05cdeb20
ac attribute arch value noarch
processing unit rpm: lion-0-0.4-1-noarch-sha256-2c5df6b51df167eb01246dce3049668cbe688033b9abff24464b0e05cdeb20
ac attribute vendor value None
processing unit rpm: lion-0-0.4-1-noarch-sha256-2c5df6b51df167eb01246dce3049668cbe688033b9abff24464b0e05cdeb20
ac attribute requires value [{u'release': None, u'epoch': None, u'version': None, u'flags': None, u'name': u'wolf'}]
processing {u'release': None, u'epoch': None, u'version': None, u'flags': None, u'name': u'wolf'} (dict) attribute name value wolf
processing {u'release': None, u'epoch': None, u'version': None, u'flags': None, u'name': u'wolf'} (dict) attribute epoch value None
processing {u'release': None, u'epoch': None, u'version': None, u'flags': None, u'name': u'wolf'} (dict) attribute version value None
processing {u'release': None, u'epoch': None, u'version': None, u'flags': None, u'name': u'wolf'} (dict) attribute release value None
processing unit rpm: lion-0-0.4-1-noarch-sha256-2c5df6b51df167eb01246dce3049668cbe688033b9abff24464b0e05cdeb20
ac attribute provides value [{u'release': u'1', u'epoch': u'0', u'version': u'0.4', u'flags': u'EQ', u'name': u'lion'}]
# ----->%-----
chimpanzee-0 0.21 1.<NULL>
Found unit lion: 0bebcf20-b999-4707-9809-656f0e38e5cd, solv: 2
Segmentation fault
[vagrant@pulp2 tmp]$

```

```

Core was generated by `python pulpsolv.py --repo-name zoo'.
Program terminated with signal SIGSEGV, Segmentation fault.
#0 0x00007f29da3ed893 in solver_addpkgrulesforsolvable (
    solv=solv@entry=0x560d60f65b90, s=0x560d60f45d00, m=m@entry=0x7ffc5f931fa0)
    at /usr/src/debug/libsolv-0.6.30/src/rules.c:1034
1034         if (m && pool->implicitobsoleteusescolors && (s->arch > pool->lastarch || pool->id2arch[s->arch]
    != 1))
[Current thread is 1 (Thread 0x7f29dcfa2440 (LWP 19137))]
Missing separate debuginfos, use: dnf debuginfo-install python2-2.7.14-4.fc26.x86_64
(gdb) where
#0 0x00007f29da3ed893 in solver_addpkgrulesforsolvable (
    solv=solv@entry=0x560d60f65b90, s=0x560d60f45d00, m=m@entry=0x7ffc5f931fa0)
    at /usr/src/debug/libsolv-0.6.30/src/rules.c:1034
#1 0x00007f29da3b538e in solver_solve (solv=solv@entry=0x560d60f65b90,
    job=0x560d60f65b98, job@entry=0x7ffc5f932080)
    at /usr/src/debug/libsolv-0.6.30/src/solver.c:3652
#2 0x00007f29da88451f in Solver_solve_helper (jobs=..., self=0x560d60f65b90)
    at /usr/src/debug/libsolv-0.6.30/build/bindings/python/solv_python.c:5066
#3 _wrap_Solver_solve_helper (self=<optimized out>, args=<optimized out>)
    at /usr/src/debug/libsolv-0.6.30/build/bindings/python/solv_python.c:14932
#4 0x00007f29dcaaf53e in PyEval_EvalFrameEx () from /lib64/libpython2.7.so.1.0
#5 0x00007f29dcaadb19 in PyEval_EvalFrameEx () from /lib64/libpython2.7.so.1.0
#6 0x00007f29dcaadb19 in PyEval_EvalFrameEx () from /lib64/libpython2.7.so.1.0
#7 0x00007f29dcab0198 in PyEval_EvalCodeEx () from /lib64/libpython2.7.so.1.0
#8 0x00007f29dcab03a9 in PyEval_EvalCode () from /lib64/libpython2.7.so.1.0
#9 0x00007f29dcab639f in run_mod () from /lib64/libpython2.7.so.1.0
#10 0x00007f29dcab634a in PyRun_FileExFlags () from /lib64/libpython2.7.so.1.0
#11 0x00007f29dcab623e in PyRun_SimpleFileExFlags ()
    from /lib64/libpython2.7.so.1.0
#12 0x00007f29dcabc519 in Py_Main () from /lib64/libpython2.7.so.1.0
#13 0x00007f29dbc5488a in __libc_start_main () from /lib64/libc.so.6
#14 0x0000560d5f98377a in _start ()
(gdb)

```

#12 - 05/22/2018 10:57 PM - milan

- File *pulpsolv.py* added

Attaching a working POC; it still needs some polishing tho

#14 - 05/25/2018 04:47 PM - milan

- File *deleted (pulpsolv.py)*

- File *deleted (pulpsolv.py)*

Removing the outdated attachments and replacing with my [Github POC repo](#).

#15 - 05/25/2018 04:57 PM - milan

Random POC Notes

- not calling pool.setarch() [causes a segfault](#) (see also the note 11 above). libsolv requires setting an architecture which might be an issue as our repos can contain units with different architectures. libdnf being build on top of libsolv most likely exposes the same behavior.
- pool.parsemprichdep is needed to parse rich dependencies. Fedora26 however didn't ship with libsolv >=0.6.33 so I had to workaround on my dev box; for demo purposes, one can use e.g [ignatenkobrain's copr repo](#)
- The files attribute of an rpm unit isn't processed yet but should be trivial to add.
- Just the rpm unit type supported; drpm, srpm units should be trivial to add
- me not knowing what the errata type depsolving should look like I feel it might be a challenge to implement; we might consider evaluating this
- to support recursive copying of particular module dependencies while maintaining the user-expected [module vs system upgrade workflow semantics](#) we'd need to expose the pool->considered C-API function to Python. This function operates over a custom-sized bitmap vector of enabled--disabled content units that we'd have to implement an adapter for, between the Python and C runtimes. This would be a focused patch against the [upstream libsolv repo](#)
- libsolv exposes a decent unit/solvable lookup interface, that supports e.g the [globing syntax](#)

#16 - 05/29/2018 03:42 PM - milan

TODO

Yum content types not covered yet:

- ~~Erratum (easy) POC~~
- ~~srpm (easy) POC~~
- drpm (tricky) might not be needed
- ~~package category (trivial), a set of groups~~
- ~~package group (trivial), a set of (RPM) packages~~

Yum features not covered yet:

- ~~weak dependencies (done in POC)~~
- file-provides (tricky)

Modularity/App-streams support:

- expose pool->considered to Python bindings (tricky)
- module as a solvable (dunno, probably tricky)

#17 - 06/01/2018 11:12 PM - rchan

- Sprint changed from Sprint 37 to Sprint 38

#19 - 06/08/2018 08:58 AM - milan

- Status changed from ASSIGNED to CLOSED - COMPLETE

Marking as done since all the POC items have been implemented and the investigation was concluded, favoring the libsolv library. See the issues [#3715#3740](#) and #3741 for the Pulp2 follow-up stories.

#20 - 11/13/2018 09:28 PM - milan

- Related to Issue #4152: Regression Pulp 2.17.1: recursive copy of RPMs does not copy partially resolvable dependencies added

#21 - 04/15/2019 10:12 PM - bmbouter

- Tags Pulp 2 added