

yum工作原理及配置

Yum 全称 (Yellow dog Updater, Modified), 是一款基于 rpm 包管理的软件包前端管理器。yum 的前身是 Yellow Dog Linux (开源的操作系统) 中的包管理器 Yellow Dog Updater, 后经由杜克大学的开发团队对 Yellow Dog Updater 改进修改成为现在的 yum。后被 Red Hat Enterprise Linux、CentOS 等众多发行版采用, 包括 Yellow Dog Linux 本身。

yum工作原理

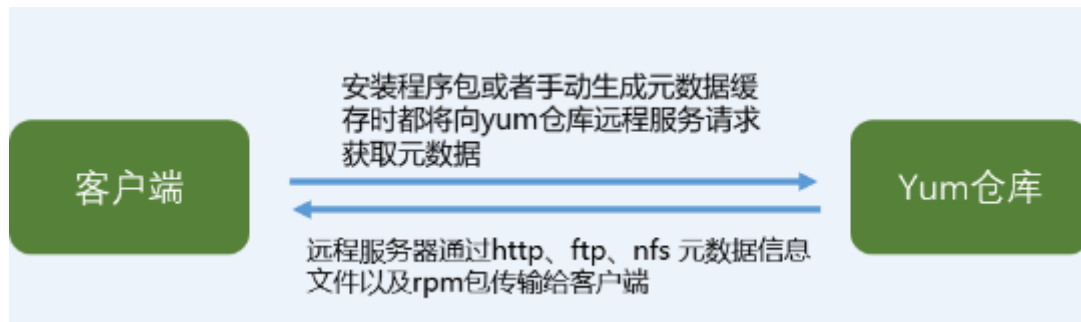
rpm 和 yum 都作为程序包管理工具, 它们都可以对程序包进行查询, 安装、卸载操作。众多的软件事实上都不是独立纯在的, 而是依赖于其他软件的组件完成它的工作, 使用 rpm 包安装程序包时, 如果软件包有依赖关系存在需要安装依赖包时, rpm 包管理器只会提醒用户所需要的依赖包, 并不会自动帮你解决依赖关系, 需要手动安装相关的依赖包; yum 作为 rpm 包的前端管理工具, 可以解决这一问题。当然实现这一功能的还有 debian 所发展的另外一个软件包管理器 APT (Advanced Package Tool), 以及有 yum 的改进版本 DNF (Dandified Yum)。

使用 yum 工具安装程序包的过程是 C/S 的形式, 即需要远程服务器, 以及 yum 客户端工具。因为 RPM 包内都记录着此安装包的依赖关系信息以及此安装包安装后生成的文件列表等诸多信息, 且 yum 服务器端 (yum 仓库-repository) 存放着许许多多的 rpm 包, 所以在创建 yum 仓库时, 就事先使用分析工具将每一个 rpm 包之间的依赖关系、文件列表信息、包名称、版本、描述等信息记录保存到服务器端特定目录下, 通常存放这些信息的目录是 repodata。

本地主机通过 repo 配置文件 (/etc/yum.repos.d/*.repo), 指定通过 yum 客户端工具安装程序时连接的远程 yum 服务器即 yum 源。远程服务器上存储着包与包之间的依赖关系等信息文件, yum 客户端每一次请求安装程序包时, 都会先将这些元数据下载并缓存到本地, yum 缓存默认保存在

`cachedir=/var/cache/yum/$basearch/$releasever` 路径下，yum客户端通过元数据信息按照依赖关系将rpm包从yum远程服务下载到本地，然后进行安装，默认安装完成后，yum工具会自动将安装包删除。

yum工作流程



yum配置

yum工具的配置文件的组成。 `main` 公共配置段 `/etc/yum.conf`，和 `repo` 仓库配置段 `/etc/yum.repos.d/*.repo`。主要配置段对所有仓库的行为属性进行定义，仓库配置段对单独的仓库进行配置定义。

main段 `/etc/yum.conf`

```
1 # main段 /etc/yum.conf
2 [main]
3 cachedir=/var/cache/yum/$basearch/$releasever
4     # 元数据信息缓存保存路径
5 keepcache=0
6     # 下载的rpm包是否保存到本地
7 debuglevel=2
8     # 调试级别，默认为2
9 retries=6
10    # 重试次数。指网络发生错误的情况下，yum重新尝试安装的
    # 次数，默认是6，如果是0，则表示无限的重试
11 logfile=/var/log/yum.log
12    # 日志文件保存
13 exactarch=1
14    # 对所要下载的rpm包是否与当前的系统架构平台严格匹配
```

```

15 obsoletes=1
16     # 允许更新旧版本的程序包
17 gpgcheck=1
18     # 是否检查来源合法性及包完整性
19 plugins=1
20     # 是否支持插件模式
21 installonly_limit=5
22     # 支持一次安装多少个程序包
23 bugtracker_url=http://bugs.centos.org/set_project.php?
    project_id=23&ref=http://bugs.centos.org/bug_report_
    page.php?category=yum
24 distroverpkg=centos-release
25     # 指定一个软件包，系统会根据这个包判断发行版本
26 pkgpolicy=newest
27     # 有两个选项newest|last。在配置了多个仓库的场景下，如
    果要安装的软件在多个仓库中存在，则newest选项表示选择安装最
    新版本的安装包，last表示按照仓库名称字母顺序进行选择仓库，
    然后进行安装

```

repo段 /rec/yum.repos.d/.repo

```

1  # repo段 /rec/yum.repos.d/*.repo
2  [base]
3      # 仓库名称ID，名称不能有空格，名称ID要唯一，不能出现同
    名的仓库
4  name=CentOS-$releasever - Base - mirrors.aliyun.com
5      # 仓库的描述信息
6  failovermethod=priority
7      # 默认值是priority。priority表示顺序从仓库下载，
    roundrobin表示随机选择一个仓库下载
8  baseurl=http://mirrors.aliyun.com/centos/$releasever
    /os/$basearch/
9      # 仓库的访问路径。可以指明多个仓库，repodata所在的路
    径就是仓库的路径
10
    http://mirrors.aliyuncs.com/centos/$releasever/os/$
    basearch/

```

```

11 mirrorlist=http://mirrorlist.centos.org/?
   release=$releasever&arch=$basearch&repo=os
12     # 镜像服务器的地址列表。将多个baseurl放到mirrorlist
   服务器上，yum工具从其中选择一个最快的仓库服务器下载
13 gpgcheck=1
14     # 是否检查包来源合法性和完整性
15 gpgkey=http://mirrors.aliyun.com/centos/RPM-GPG-KEY-
   CentOS-7
16     # 公钥文件（可以是本地，也可以是服务器路径）
17 enabled=1
18     # 是否启用此仓库，默认是启用
19 cost=1000 # 定义仓库开销，默认1000，越小优先使用

```

yum配置文件中的内置变量

内置变量的使用，增强了repo配置文件的通用性。

`$releaserver` 指当前操作系统的发行版的主版本号

`$basearch` 系统基础架构

`$arch` 系统CPU架构，X86_64或者i386

`$YUM[0-9]` 系统中定义的，用户可以在yum中使用

yum使用

yum工具使用

基本格式：`yum [option] COMMAND`

安装/升级/卸载

```

1 yum install package_name
2     # 安装包，后跟软件包名称，可指定安装包版本
3 yum reinstall package_name
4     # 重新安装
5 yum check-update
6     # 查看可升级的程序包
7 yum update package_name

```

```
8      # 升级指定的包，可以升级到指定版本：yum update X-
      1.2.2
9 yum downmgrade package_name
10     # 对程序包进行降级版本
11 yum [groupinstall|groupupdate] "group_name"
12     # 安装或升级指定包组
13 yum groupremove "group_name"
14     # 卸载指定包组
15 yum remove paceage_name
16     # 卸载指定的程序包
17 yum deplist package_name
18     # 显示程序的依赖包
19 yum localinstall package_name
20     # 从本地的rpm安装，如果有依赖关系，也会从仓库下载
21 yum --enablerepo=repoid
22     # 指定从某个仓库安装程序
```

yum仓库管理

```
1 yum repolist [all|enabled|disabled]
2     # 显示所有仓库|可用|禁用
3 yum clean [all|packages|metadata|expire-
      cache|rpmdb|plugins]
4     # 清理yum缓存
5 yum makecache
6     # 手动构建缓存
7 yum search COMMAND
8     # 搜索包含此工具的包名
```

信息查询

```
1 yum list [all|installed|available]
2     # 查看所有rpm包|已经安装|可用安装包
3 yum info package_name
4     # 查看包信息
5 yum grouplist
6     # 查看仓库包组信息
7 yum groupinfo "package_group_name"
8     # 查看指定包组的信息
9 yum whatprovides /path/fo/file
10    # 查询某文件是由那个安装包安装的
11 yum history
12    # 查看yum操作历史
```

1.显示@符号的表示已经安装好的; 2.显示anaconda的表示系统安装时安装的软件。

参考资料

- [developerWorks 中国](#)