

Apache2.4版本新特性

1. MPM支持在运行时装载，需要指定编译选项，`--enable-mpms-shared=all` `--with-mpm=event`
2. 新增了一些模块：mod_proxy_fcgi, mod_ratelimit, mod_request, mod_remoteip
3. 在每个模块和每个目录上指定日志级别
4. 支持配置文件中自定义变量
5. 支持更精确的keepalive timeout，单位是毫秒
6. 完全支持MPM Event模型
7. 支持异步读写
8. 增强版的表达式分析器

编译安装前准备

下载相关软件包

```
1 wget http://mirrors.hust.edu.cn/apache//httpd/httpd-2.4.25.tar.gz
2 wget http://mirror.bit.edu.cn/apache//apr/apr-1.5.2.tar.gz
3 wget http://mirror.bit.edu.cn/apache//apr/apr-util-1.5.4.tar.gz
```

安装依赖库及编译工具

```
1 yum groupinstall "Development Tools"
2 yum install pcre-devel
3 yum install zlib zlib-devel
4 yum update openssl
5 yum install openssl-devel
```

安装apr及apr-util

APR(Apache Portable Runtime)。Apache可移植运行库，APR提供了一组API，映射到底层操作系统。如果底层OS不支持某个特别的函数，APR自己会提供一个替代品，APR可以确保程序的跨平台可移植性。apr-util在apr的基础上提供了更多的数据结构和操作系统封装接口。所以需要先安装apr及apr-util。

方法一

可以先行进行编译安装安装apr、apr-util：

```
1 #编译apr
2 tar xf apr-1.5.0.tar.gz
3 cd apr-1.5.0
4 ./configure --prefix=/usr/local/apr
5 make && make install
6 #编译apr-util
7 tar xf apr-util-1.5.3.tar.gz
8 cd apr-util-1.5.3
9 ./configure --prefix=/usr/local/apr-util --with-
  apr=/usr/local/apr/
10 make && make install
```

然后在编译安装Apache时，编译配置参数指定：

```
--with-apr=/usr/local/apr
```

```
-with-apr-util=/usr/local/apr-util
```

方法二

将apr及apr-util的源码放进Apache源码目录src/lib目录中，在进行编译Apache的时，会自动到此目录下查找apr文件，然后进行编译。

```
1 tar -xf apr-1.5.2.tar.gz
2 mv apr-1.5.2 httpd-2.4.25/src/lib/apr
3
4 tar -xf apr-util-1.5.4.tar.gz
5 mv apr-util-1.5.4 httpd-2.4.25/src/lib/apr-util
```

配置编译

具体参数含义参考：configure配置源代码树文档：<http://httpd.apache.org/docs/2.4/programs/configure.html>

Optional Features: 选项中，`--disable`的选项默认是开启的，添加此选项则关闭。`--enable`的选项默认是关闭的，添加此选项开启。

`--enable-mpms-shared=MPM-LIST`

表示将MPM模块编译为可动态加载的模块，MPM-LIST可以是MPM模块名称例如`--enable-mpms-shared="worker event prefork"`或者`--enable-mods-shared=all`

`--enable-mods-shared=MODULE-LIST`

定义要[启用]并[构建]为动态共享模块的模块列表。MODULE-LIST可以是模块名称，也可以是 `all` `most` `few` `reallyall` 这几个值。

配置参数

此处选择将apr和apr-util的源码复制到Apache源码目录内，然后编译。

```
1 ./configure \  
2 --prefix=/usr/local/apache24 \  
3 --enable-so \  
4 --enable-mods-shared=most \  
5 --enable-mpms-shared=all \  
6 --with-mpm=event \  
7 --enable-rewrite \  
8 --enable-cgi \  
9 --enable-ssl \  
10 --enable-deflate \  
11 --with-pcre \  
12 --with-apr=/usr/local/apr \  
13 --with-apr-util=/usr/local/apr-util
```

`--prefix=/usr/local/apache24` ---->指定安装路径

`--enable-so` ----->开启DSO功能，默认将自动开启

`--enable-mods-shared=most` ----->将编译大多数模块并构建为DSO模块

`--enable-mpms-shared=all` ----->支持当前平台上动态加载的所有MPM，并将其构建为DSO模块

`--with-mpm=event` ----->选择服务器的默认MPM模块

`--enable-rewrite` ----->开启URL的重写

`--enable-cgi` ----->支持CGI

`--enable-ssl` ----->支持SSL/TLS

`--enable-deflate` ----->压缩压缩模块

`--with-pcre` ----->支持正则表达式

编译安装

```
1 | make && make install
```

后续配置

输出二进制文件

```
1 | #新增文件
2 | vim /etc/profile.d/httpd.sh
3 | export PATH=/usr/local/apache24/bin:$PATH
```

导出头文件(非必要)

```
1 | ln -sv /usr/local/apache24/include
   | /usr/include/httpd
```

导出库文件

```
1 echo "/usr/local/apache2/lib" >
   /etc/ld.so.conf.d/httpd.conf
```

```
1 #重新加载库文件
2 ldconfig -v
3
4 #显示当前加载的库文件
5 ldconfig -p
```

导出man文档

```
1 #添加
2 vim /etc/man_db.conf
3 MANDATORY_MANPATH /usr/local/apache24/man
```

创建apache用户和用户组

```
1 useradd -s /sbin/nologin -M apache
2 #修改apache配置文件httpd.conf中User、Group配置指令为
   apache
```

添加systemctl服务脚本

```
1 [Unit]
2 Description=The Apache HTTP Server
3 After=network.target remote-fs.target nss-
   lookup.target
4 Documentation=man:httpd(8)
5 Documentation=man:apachectl(8)
6
7 [Service]
8 Type=simple
9 ExecStartPre=/usr/local/apache24/bin/httpd -t
10 ExecStart=/usr/local/apache24/bin/httpd $OPTIONS -
   DFOREGROUND
11 # ExecStartPost=/home/status.sh
12 ExecReload=/usr/local/apache24/bin/httpd $OPTIONS -k
   graceful
```

```
13 ExecStop=/bin/kill -WINCH ${MAINPID}
14 KillSignal=SIGCONT
15 PrivateTmp=true
16
17 [Install]
18 WantedBy=multi-user.target
```

注意：手动编写服务脚本时，注意服务类型Type的选择，否则可能会无法启动。修改完服务脚本后要重新加载systemctl daemon-reload。此处可以选择Type=simple

Type=simple: (默认值) systemd认为该服务将立即启动。服务进程不会 fork 。

Type=forking: systemd认为当该服务进程fork，服务启动后，父进程退出。使用此启动类型应同时指定 PIDFile=，以便 systemd 能够跟踪服务的主进程。

Type=oneshot: 这一选项适用于只执行一项任务、随后立即退出的服务。

关于服务脚本Type参考：<https://www.freedesktop.org/software/systemd/man/systemd.unit.html>