

使用Spack软件包管理器构建高性能计算软件环境

● 韦建文 王杰 文敏华 王一超 林新华

上海交通大学 网络信息中心 上海 200240

摘要：

为科研人员提供一个稳定易用的计算环境是高性能计算平台的重要任务。现有主流的基于手动编译的软件环境构建方法，由于缺少自动化编译和多版本共存等特性，难以有效管理日益增多的软件。本文介绍一个面向高性能计算的软件包管理器Spack，包括其主要用法、内部设计、在编译耗时和运行效率上的优势。实验结果表明，Spack有效降低了软件编译时间，且编译出的软件运行效率优于Docker，与手工编译相同。上海交通大学高性能计算平台基于Spack实现了分角色的软件管理方法，除了管理员提供的超过200种常用软件，用户可以很方便地自行构建专属的软件环境。新软件上线时间从5天缩短到2天，极大提高了系统管理员的维护效率和用户的使用效率。

关键词：高性能计算，软件包管理器

引言

上海交通大学高性能计算平台的计算能力位居全国高校前列^[1]，自2013年上线以来，常年满负荷运行，已累计为校内300多个课题组提供超过5亿核小时的计算服务。为了支持用户高效完成计算任务，除了稳定可靠的硬件环境，还需要灵活易用的软件环境。在我们每年两千多人技术咨询中，“如何安装软件”占了60%。为应对这种情况，我们配备了一名员工专职从事应用编译，在过去一年新增了71个计算软件和库，进行了205次版本更新，可用软件包总数超过200个。

即将于2019年9月上线的第二代校级高性能计算平台，计算能力提升了10倍，理论性能达到2PFlops。可以预期的是，用户必将对软件种类和易用性提出更高的要求。如何构建一个软件种类丰富、运行高效、调用方便的软件环境，是维护高性能计算平台的一个难题。本文将着重介绍构建高性能计算软件环境的主要挑战、软件包管理器的概念、如何使用Spack软件包管理器，以及使用Spack构建上海交大的超算软件环境。

1. 构建高性能计算软件环境的主要挑战

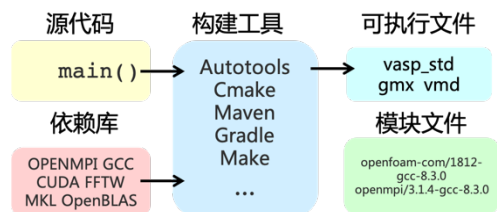


图1 软件编译流程

如图1所示，编译一个高性能计算软件需要准备好软件源代码以及依赖软件库，然后通过构建工具编译出可执行程序，最后整理出软件模块文件供用户调用。构建一个面向高性能计算的软件环境存在如下挑战：

软件数量多。常用的高性能计算软件多达上百种，若算上软件在版本、编译器、优化选项上的变种，构建软件环境需要重复上千次准备源代码、编译、撰写模块的操作。

编译流程差异大。常用的软件编译工具就有Autotools^[11]、CMake^[12]、Maven^[13]、SCons^[14]等十多种，在用法上差异较大，需要投入大量时间才能熟练掌握这些工具。

维护环境模块工作量大。供最终用户使用的环境模块文件，必须与软件版本一一对应，手动维护模

块文件难以保证内容的正确性。

软件版本共存。出于性能、稳定性和结果一致性的考虑，高性能计算平台需要提供同一个软件的多个版本供用户使用，这需要在安装时设计好安装路径，避免路径冲突。

软件管理角色不清晰。在现有方法中，系统管理员拥有特权可以部署全局可用的软件，普通用户因无特权而很难从基础库开始构建自定义的软件环境。

2. 软件包管理器

表1 软件包管理器对比

软件包管理器	软件数量	发布形式	能否非特权部署	能否多版本共存	是否提供环境模块文件
OpenHPC	2361	预编译包	否	否	否
Anaconda	9053	预编译包	是	是	否
Docker, Singularity	19549	预编译包	是	是	否
EsayBuild	1850	源代码	是	是	否
Spack	3247	源代码	是	是	是

OpenHPC^[3]是一个基于Yum的软件包管理工具，以预编译包提供x86和ARM平台的常用高性能计算库。OpenHPC需要使用管理员权限安装，系统中只能保留软件的一个版本。

Anaconda^[4]是基于Pip的软件包管理器，用于管理Python扩展包和一般的科学计算软件包。Anaconda可以把软件安装在用户家目录，不需要管理员特权，并且可以创建多个相互隔离的软件环境。

Docker^[5]和Singularity^[6]是基于容器技术的软件封装技术，他们将软件连同系统库一起打包，构建出一个共享宿主节点内核的运行环境。应用只需要打包一次就能在支持容器技术、同一指令集架构的操作系统上运行。

EasyBuild^[7]和Spack^[8]借鉴了BSD Ports，将软件编译过程中的关键步骤抽象成一系列函数，用户使用函数而非具体执行的命令描述编译过程，在实际执行时包管理器根据用户编写的流程完成下载源代码包、解压、编译、安装的流程。这两个工具都可以方便地以普通用户权限安装软件，且已支持超过1000种科学计算软件。Spack安装简单，命令行提供了丰富的编译选项，评估试用后被选中用于构建上海交大高性能计算平台的软件环境。

3. 使用Spack安装软件

3.1 安装Spack

为了应对构建软件环境的挑战，一类被称为软件包管理器 (Software Package Manager) 的工具应运而生。这类工具屏蔽了不同软件在获取、编译和加载过程中的差异，为用户提供了一个统一的软件包管理接口，首先在类UNIX操作系统上获得了广泛应用，例如Yum、Apt、BSD Ports等。一些针对高性能计算软件环境的包管理器在此基础上改进，如OpenHPC、Anaconda、EasyBuild、Spack等。此外，Docker和Singularity等基于容器技术的软件部署方法，也获得了越来越多的关注。这些软件包管理器的主要特性如表1所示。

Spack只需要Python和GCC编译器就能工作，安装过程非常简单，不需要管理员特权，只要将源代码克隆到本地即可使用。

```
$ git clone https://github.com/spack/spack.git
$ cd spack/bin
$ ./spack -v
```

3.2 使用Spack安装软件包

Spack使用install命令安装软件包，这个命令可指定软件版本、软件功能、优化选项、编译器、依赖软件包等编译参数。如图2所示，使用Spack安装aoflagger^[15]（一个用于射电天文学数据处理的开源软件）2.10.0版本，指定编译器优化参数(cflags)为“-O2”，使用符号“%”指定编译器为Intel编译器18.0.3，使用“^”符号指定依赖软件为casacore 2.4.1版本，并要求在casacore上开启对python的支持。

```
spack install \
  aoflagger@2.10.0 \<..... 软件名
  cflags=-O2 \<..... 优化级别
  ^casacore@2.4.1+python \<..... 依赖库
  %intel@18.0.3 \<..... 编译器
```

图2 Spack安装软件示例

3.3 Spack的工作流程

编译软件时，Spack根据命令行指定的选项，读

取与软件包对应的package.py包文件，解析软件包下载地址、特性选项、依赖和冲突、编译方式，然后编译软件。aoflagger软件的包文件片段如图3所示。

其中，class指定软件包类型，以决定使用什么工具编译这个软件，Spack支持的软件包类型包括AutotoolsPackage、CMakePackage、MakePackage以及一般的Package等；home和url分别用于指定软件主页和下载地址；version指定软件包的版本和校验和，确保用户下载的软件源码包是正确的；variant用于描述软件支持的特性，常见的如是否开启对openmp、mpi、python的支持等，使用者可以在spack install选项中通过“+”或“-”符号开启或关闭这些特性；depends_on和conflicts指令用于描述这个软件依赖的其他软件包，或是与之冲突的软件包、编译器等。

Spack将递归地构建出一个包含软件及其依赖包的有向无圈图(directed acyclic graph, DAG)，以此指导编译过程。编译aoflagger的有向无圈图如图4所示，Spack将从下到上依次编译依赖包最后编译aoflagger。

```
class Aoflagger(CMakePackage):
    """RFI detector and quality analysis
    for astronomical radio observations."""

    homepage="https://sourceforge.net/aoflagger/"
    url="https://downloads.sourceforge.net/
    project/aoflagger/aoflagger-2.10.0/
    aoflagger-2.10.0.tar.bz2"

    version('2.10.0',
           'f1df6f9cc3ea87a529a3a53da9bb3033')

    variant('python',
           default=True,
           description='Add python support')

    depends_on('boost')
    depends_on('casacore')

    conflicts('%gcc@9:')
```

图3 aoflagger的package.py示例片段

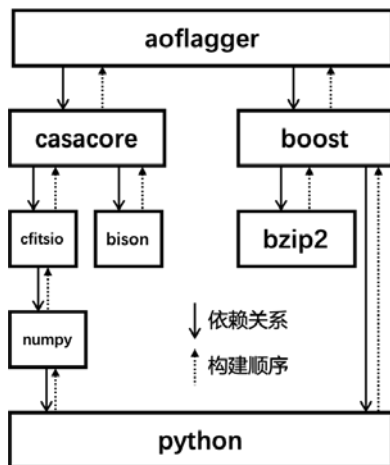


图4 aoflagger软件包依赖关系

3.4 调用Spack安装的软件

高性能计算系统上普遍使用环境模块^[2] (Environment Module) 调用软件，环境模块通过修改“PATH”、“CPATH”、“LD_LIBRARY_PATH”等环境变量动态地调用软件。但是，为软件的每一个版本及其依赖库手工编写模块文件的工作量很大。Spack通过RPATH和自动生成模块降低了软件调用的难度。首先，Spack在编译时使用RPATH参数将依赖库的路径写入了编译出的二进制程序中因而不需依赖环境模块提供的LD_LIBRARY_PATH。其次，每一个安装的软件唯一地对应了一个有向无圈图，Spack在软件安装路径中添加了该图的散列值，从而保证同一软件的不同编译版本或变种能唯一地确定，不会发生路径冲突。最后，Spack根据软件的构建选项和安装路径，可自动生成环境模块供用户调用。下面的示例展示了通过module命令加载和使用软件的流程。

```
$ module avail gromacs
gromacs/4.5.5-intel-19.0.4 gromacs/4.5.5-gcc-8.3.0
$ module load gromacs/4.5.5-intel-19.0.4
$ which mdrun_mpi
/lustre/spack/opt/sandybridge/intel-19.0.4/gromacs-
4.5.5-f239hfe823gf/bin
```

3.5 Spack性能测试

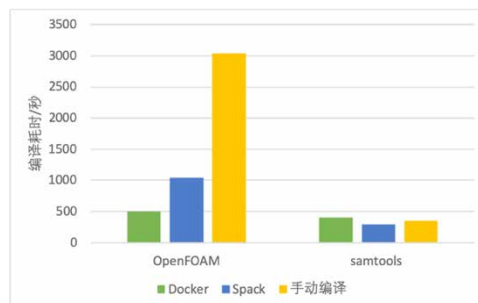


图5 编译时间

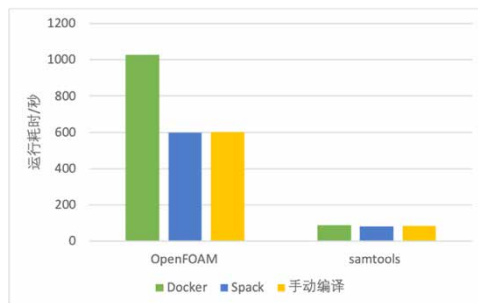


图6 运行时间

我们使用软件编译时间与软件运行时间来考察Spack性能，与之对比的是手动编译和Docker编译。用于测试的程序是基因数据处理软件Samtools^[9]和计算流体软件OpenFOAM^[10]，前者运行一个单机多线程

算例，后者运行一个4节点MPI算例。

编译耗时如图5所示。Samtools编译耗时差异很小，Docker因为需要下载镜像，最耗时。OpenFOAM编译耗时差异大，Docker可以直接使用操作系统预编译的OpenMPI库因而最快；Spack可以自动解决依赖包，可在无人值守状态下完成编译，也比手工方式快。

运行耗时如图6所示。单节点Samtools运行耗时差异很小，Docker多了首次启动需要的15秒时间。4节点OpenFOAM运行耗时差异大，手动编译和Spack安装的版本基本持平，分别需要603秒和598秒完成计算；而Docker版需要1028秒，慢了41%。经检查发现，Docker版本使用的操作系统自带OpenMPI库没有加入对Infiniband网络的支持，运行过程中使用了低速的千兆网导致多节点运行效率降低。

综上，手动编译和Spack编译都能发挥出了高性能计算平台的应有性能。而Docker为保证兼容性优先选择了操作系统预编译的软件包，这在配备了Infiniband等特殊硬件的高性能计算平台上并不是最优方案。使用Spack作为构建Docker镜像的工具，以解决Docker应用在高性能计算平台上多节点运行效率低的问题，是一个可行的方向。

4. 上海交通大学高性能计算中心的Spack实践

上海交大高性能计算中心摸索出了一套使用Spack构建高性能计算软件环境的方法，在国内率先使用Spack安装生产环境上的软件，构建流程如图7所示。

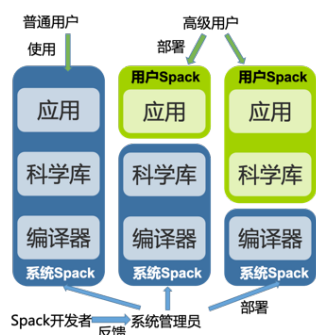


图7 基于Spack按角色分工的高性能计算软件环境构建方法

首先，Spack使用者被分为三类：1)Spack开发者，负责定期从Spack社区更新软件包，测试软件

包稳定性，并补充Spack缺失的软件包；2)系统管理员，根据Spack开发者反馈的测试结果，将软件的稳定版本部署到系统中供所有用户使用；3)普通用户，既可以直接使用Spack部署在系统中的应用，也可以只使用Spack部署的编译器或科学库，或者完全从头用Spack部署自己的软件环境。三个角色在相互隔离的环境中测试与合作，管理员只部署通过验证的软件，而开发者进行的测试不会影响生产环境，普通用户可自由选择复用或重新构建整个软件环境。

其次，按类别选择需要部署的软件。软件环境中的软件数量并非越多越好，在环境中共存的软件数量越多，在生成模块文件和指定依赖关系时产生冲突的几率也越大。我们根据软件类型、使用广泛程度等因素，选择一个稳定可靠的子集部署在软件环境中。对于编译器和科学库这类基础软件，如GCC、OpenBLAS等，会安装尽可能多的版本供用户选择，并保持一个版本能用至少半年；常用科学计算软件会保留最近的两到三个版本，更新频率要比基础库高；对于使用量小的软件将被部署在特定用户的家目录下。

这套分角色分类别的软件构建方法，很好兼顾了只需要运行计算软件的初级用户需求，和需要构建自己软件环境的高级用户的需求。软件部署前经过测试，超过生命周期后有序退出，管理员部署的软件也能最大限度地复用，避免了重复劳动。

5. 小结

本文总结了构建高性能计算软件环境遇到的软件数量多、编译流程差异大、维护模块工作量大、软件版本共存、软件管理角色不清晰等挑战，在对比数个软件包管理器后，重点介绍了Spack软件包管理器的安装方法、工作流程和软件调用方法，并通过实验展示了Spack优异的编译速度和高质量的编译结果。最后介绍了上海交通大学高性能计算平台基于Spack的分角色分类别的软件环境部署流程，这套流程能涵盖Spack开发者、系统管理员和普通用户的软件部署需求，为构建高性能计算软件环境提供了有效方法。Spack收录的软件包数量和功能仍在稳步增加，将为高性能计算用户构建特定领域的软件环境带来便利。

参考文献

- [1] 上海交通大学高性能计算中心[EB/OL]. 2019[2019-07-15]. <https://hpc.sjtu.edu.cn>.
- [2] Environment Modules[EB/OL]. 2019[2019-03-24]. <http://modules.sourceforge.net>.
- [3] OpenHPC: Community Building Blocks for HPC Systems[EB/OL]. 2019[2019-03-24]. <https://openhpc.community>.

- [4] Anaconda Cloud[EB/OL].2019[2019 - 03 - 24]. <https://anaconda.org>.
- [5] Docker: Empowering App Development for Developers[EB/OL]. 2019[2019 - 03 - 24]. <https://www.docker.com>. [6] Singularity [EB/OL].2019[2019 - 03 - 24]. <https://www.sylabs.io/>.
- [7] Hoste K, Timmerman J, Georges A , et al. EasyBuild: Building Software With Ease[C]//2012 SC Companion: the International Conference for High Performance Computing, Networking, Storage and Analysis, Salt Lake City, 2012 - 11 - 10至2012 - 11 - 16.
- [8] Gamblin T, Legendre M, Collette M R, et al. The Spack package manager: bringing order to HPC software chaos[C]//2015 SC Companion: the International Conference for High Performance Computing, Networking, Storage and Analysis, Austin, 2015 - 11 - 15至2015 - 11 - 20.
- [9] Li Heng, Handsaker Bob, Wysoker Alec, et al. The Sequence Alignment/Map format and SAMtools[J]. Bioinformatics, 2009, 25(16): 2078 - 2079.
- [10] Jasak, Hrvoje. OpenFOAM: Open source CFD in research and industry[J]. International Journal of Naval Architecture and Ocean Engineering, 2009, 1(2): 89 - 94.
- [11] GNU Autotools[EB/OL].2017[2019 - 03 - 24]. <https://www.gnu.org/software/automake/>. [12] CMake[EB/OL].2019[2019 - 03 - 24]. <https://cmake.org/>.
- [13] Maven[EB/OL].2019[2019 - 03 - 24] <https://maven.apache.org/>.
- [14] SCons: A software construction tool[EB/OL].2019[2019 - 03 - 24]. <https://scons.org/>.
- [15] AOFlogger[EB/OL].2019[2019 - 03 - 24]. <https://sourceforge.net/p/aoflogger/wiki/Home/>.

要闻集锦

日本“富岳”超级计算机找出数十种新冠候选药

新华社北京7月6日新媒体专电：日媒称，京都大学一研究团队7月3日宣布，他们利用理化学研究所拥有世界最快运算速度的“富岳”（Fugaku）超级计算机，发现了数十种有望成为新冠肺炎特效药的物质，这些物质有可能抑制细胞内病毒的繁殖。

据《日本经济新闻》7月4日报道，京都大学奥野恭史教授等人组成的研究团队宣布，研究人员将利用这些物质展开进一步实验，以确认治疗效果。

担任理化学研究所项目副主管的奥野教授及其团队利用“富岳”超算的高算力，以现有的2000多种药物为对象进行了模拟实验。实验的目的是确认这些相关药物是否能在某种程度上抑制病毒繁殖。

结果发现有数十种药物有望成为治疗新冠肺炎的特效药，其中就包括12种已经开始用于新冠肺炎治疗临

床试验的药物。比如用来驱除寄生虫的氯硝柳胺和硝唑尼特等，硝唑尼特已经在美国和墨西哥用于临床治疗。

此外，一些日本药企拥有专利的药物据说也在模拟实验中表现出不错的结果。奥野教授认为：“经过‘富岳’的计算形成的候选清单中含有一些正在用于临床治疗的药物，这说明计算结果是正确的。”

报道称，未来针对细胞开展的实验将会更加详细地确认药物的疗效。制药企业和研究人员将通力合作，在临床研究和治疗上展开合作。

据报道，在6月底的最新一届“TOP500”超算排名中，日本的“富岳”位列第一，其几天的运算量就相当于2011年曾经登顶的“京”超算工作一年的运算量。

原本“富岳”将在2021年正式投入使用，但为了参与新冠病毒的研究，提前到今年4月开始试验性应用。

（卢永捷）