鲲鹏开发套件插件工具（IntelliJ）
2.3.T10
用户指南

文档版本	01
发布日期	2021-07-30

华为技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 工具概述

鲲鹏开发套件插件工具是基于IntelliJ提供给开发者面向鲲鹏平台进行应用软件开发、迁移、编译调试、性能调优等一系列端到端工具，即插即用。一体化呈现代码迁移插件、加速库插件、编译插件及性能分析插件的完整开发套件。

鲲鹏开发套件插件工具是一个工具集，由多个插件组成，支持IDE前端界面，支持一键式安装后端，代码编辑体验增强，自动检测安装鲲鹏编译器，编译调试，用例可视化，编码辅助，工程分析扫描。

用户可以通过安装Kunpeng DevKit插件集直接将四个插件都安装好，也可以单独选择个别插件安装使用。

图 1-1 逻辑模型

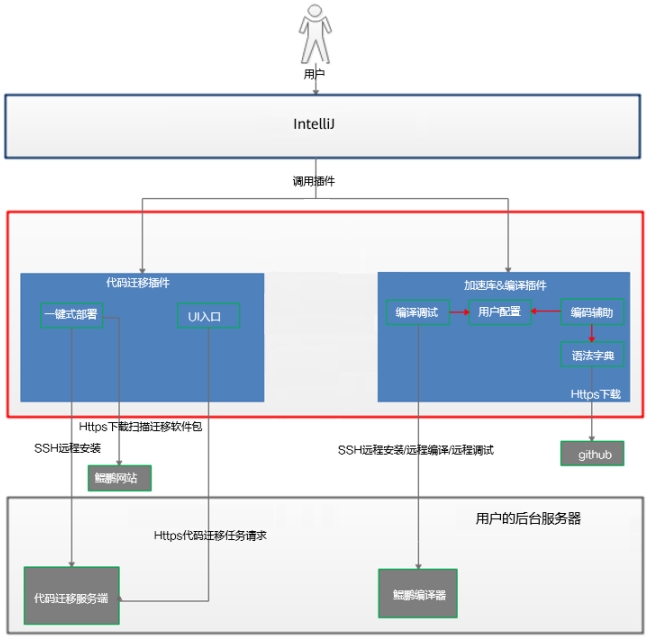


表 1-1 逻辑模型

模块名	功能
一键式部署	一键式安装/升级/卸载代码迁移工具。 后端工具安装包通过鲲鹏网站下载。
UI入口	工具的IntelliJ前端入口。
编译调试	通过SSH远程安装鲲鹏编译器。 ● 一键式安装鲲鹏GCC/毕昇编译器/毕昇JDK ● 可视化编译配置任务，一键式任务运行 ● 远程单步调试C/C++代码 ● 编译调试过程信息实时展示 ● gtest框架用例树渲染及状态展示
用户配置	用户可配置远程环境，编译/调试任务。
辅助编码	插件会自动下载鲲鹏的字典库数据，字典数据下载完成后，插件会启用针对加速库函数的语法高亮、函数定义跳转、以及代码自动补全。 语法字典在联网环境下自动从github上下载。

说明

IntelliJ/MindStudio系列插件是基于IntelliJ IDEA开发的，当前仅支持IntelliJ IDEA，其它共IntelliJ平台的IDE产品可能也可以安装此插件，但当前未进行过验证，无法保证完美兼容。

安装IntelliJ/MindStudio系列插件需要依赖kunpeng foundation插件，请在先下载kunpeng-foundation插件安装包并安装该插件，然后再安装IntelliJ鲲鹏开发套件插件。

2 安装

标题名称看看。包括安装 、升级、卸载。描述公共配置

2.1 安装鲲鹏开发套件

鲲鹏开发套件是IntelliJ的一款扩展工具。通常可将此类工具称作集成开发环境插件。

鲲鹏代码迁移插件是鲲鹏开发套件的子工具，作为客户端调用服务端的功能，完成扫描迁移任务，可以对待迁移软件进行快速扫描分析，并提供专业的代码迁移指导，极大简化客户应用迁移到鲲鹏平台的过程。当客户有软件需要迁移到鲲鹏平台上时，可先用该工具分析可迁移性和迁移投入，以解决客户软件迁移评估中分析投入大、准确率低、整体效率低下的痛点。

代码迁移工具能够自动检查并分析出用户源码、C/C++/Fortran/Python软件构建工程文件、C/C++/Fortran/Python软件构建工程文件使用的链接库、x86汇编代码中需要修改的内容，并给出修改指导，以解决用户代码兼容性排查困难、迁移经验欠缺、反复依赖编译调错定位等痛点。

说明

- IntelliJ鲲鹏代码迁移插件是基于IntelliJ IDEA开发的，当前仅支持IntelliJ IDEA，其它共IntelliJ平台的IDE产品可能也可以安装IntelliJ鲲鹏代码迁移插件，但当前未进行过验证，无法保证完美兼容。
- 仅支持x86 Linux到Kunpeng Linux的扫描与分析，不支持Windows软件代码的扫描、分析与迁移。

当前版本的开发套件已在如下操作系统上验证：

- Windows 10

- Ubuntu 18.04.4

鲲鹏代码迁移工具插件安装步骤如下：

- 在Intellij编辑器插件面板（Plugins）中根据关键字搜索“Kunpeng DevKit”或“Kunpeng Porting Advisor Plugin”，或者在Intellilj**应用商店网页**中找到Kunpeng Porting Advisor Plugin，单击“安装”。
- 也可选择在华为企业业务网站下载鲲鹏开发套件扩展安装包。步骤如下：

步骤1 登录**华为企业业务网站**下载鲲鹏开发套件扩展安装包和数字签名，确保与网站上的原始安装包一致。

说明

校验方法如下：

1. 在如下链接中获取校验工具和校验方法：
   https://support.huawei.com/enterprise/zh/tool/pgp-verify-TL1000000054
2. 下载以上链接中的《OpenPGP签名验证指南》，并根据指南进行软件包完整性检查。

步骤2 打开本地PC上的Intellij，在左上角单击“文件”，选择“设置”。

步骤3 在左侧菜单栏中先单击“插件”，然后单击顶部设置按钮，选择“从磁盘安装插件…”后选中已经下载到本地的扩展安装包，单击“确定”。

----结束

安装完成后，重新启动Intellij，在左下角单击 

打开鲲鹏代码迁移工具插件。

说明

- 该插件支持Intellij2020.2以上版本。
- 当用户在Intellij上安装了Material Theme UI等主题插件时，鲲鹏代码迁移工具插件中涉及密码的弹窗中会出现两个重复的“小眼睛”（隐藏或显示密码），两个“小眼睛”都能正常使用，不会影响整体的操作。

2.2 安装并登录鲲鹏代码迁移工具

须知

- 代码迁移工具进行代码迁移时，需要调用Linux下的rpm、deb等命令才能完成扫描和迁移相关任务，这些命令和逻辑必须在后端Linux运行。
- 插件只支持以Web模式使用root用户安装工具，不支持以CLI模式安装工具。
- 由于root用户拥有最高权限，直接使用root用户登录服务器可能会存在安全风险。建议您使用普通用户登录服务器后切换为root用户，再执行后续安装操作，并建议您通过配置禁止root用户SSH登录的选项，来提升系统安全性。具体配置如下：
  先以普通用户登录服务器，切换至root登录后检查/etc/ssh/sshd_config配置项PermitRootlogin，如果显示no，说明禁止了root用户登录；如果显示yes，则需要将配置项PermitRootlogin是否设置no。

可以通过以下两种方式安装鲲鹏代码迁移工具：

- 选择使用插件部署功能一键式安装：
  - 部署过程中，系统会自动下载鲲鹏代码迁移工具安装包。
  - 请确保部署工具的服务器可以访问外部网络，否则工具将部署失败。
- 选择手动下载并在服务器上安装鲲鹏代码迁移工具：
  - 详细操作请参见鲲鹏代码迁移工具中的“配置操作系统yum/apt/zypper源”和“安装”章节。
  - 手动将鲲鹏代码迁移工具的安装包下载到本地，并将安装包、安装包验证的**Keys.txt**和**.asc**文件上传到目标服务器“/tmp”路径，以时间戳命名的目录下，再使用插件的一键式安装。

  说明

  获取软件包后，需要校验软件包，确保与网站上的原始软件包一致。校验方法参见鲲鹏代码迁移工具中“安装包完整性校验”章节。

以使用部署功能安装工具为例，安装鲲鹏开发套件后，选择鲲鹏代码迁移插件 PortingAdvisor，单击"配置服务器"，打开如图2-1所示界面，参数描述如表2-1所示。配置参数后单击“保存”。

图 2-1 配置远端服务器

表 2-1 配置远端服务器参数说明

参数	说明
IP地址	安装工具的远程服务器IP地址。
端口	工具安装过程中设置的HTTPS端口。 **说明** 如果服务器已开通防火墙，使用代码迁移工具前请确认服务器OS防火墙已开通HTTPS端口（默认为8084）。
服务证书设置	• 选择“指定根证书”前，请管理员在“web服务端证书”设置中获取CSR文件并用该CSR文件在CA系统或自签名证书系统生成标准的X.509证书，完成签名后导入证书，完成后通过指定根证书建立安全连接。 • 选择“信任当前服务证书”则意味着用户信任同服务端建立的连接。

如果没有在服务器上部署工具，单击“点击此处部署”，打开“部署前必读”对话框。勾选“我已阅读以上文字”，单击“确定”，打开如图2-2所示界面，参数描述如表2-2所示。

配置参数后单击“检测连接”。若工具提示连接检测失败，可根据提示修改参数后重新尝试检测连接。工具提示“SSH连接检测成功”后，单击“开始部署”。

安装过程中需要：

- 再次输入服务器用户密码。
- 输入服务器root用户密码安装工具。
- 配置工具安装目录，默认为“/opt”。
- 配置Web Server的IP地址。
- 配置HTTPS端口，默认端口为8084。
- 配置tool端口，默认端口为7998。

说明

如果服务器已开通防火墙，使用代码迁移工具前请确认服务器OS防火墙已开通HTTPS端口（默认为8084）。

图 2-2 安装代码迁移工具

表 2-2 安装代码迁移工具参数说明

参数	说明
IP地址	待安装工具的服务器IP地址。
SSH端口	待安装工具的服务器SSH端口。
操作系统用户名	登录待安装工具的服务器操作系统的用户名。 **说明** 如果想通过普通用户安装，需要满足一些条件，具体参照**3.3.1 通过普通用户连接鲲鹏代码迁移工具**。

参数	说明
选择SSH连接方式	连接方式可选择： • 密码认证 • 密钥认证 **说明** 选择“密钥认证”后需要导入“id_rsa”私钥文件，详细操作请参见3.3.2 配置SSH密钥认证。
操作系统用户密码	操作系统用户密码。
说明 安装后如需卸载工具，使用SSH远程登录工具，以**root**用户登录Linux操作系统命令行界面，执行如下命令（“/opt/portadv”为工具安装目录，请根据实际情况替换）： **bash /*opt*/portadv/tools/uninstall.sh**	

工具部署完成后，单击“立即登录”，打开如图2-3和图2-4所示界面，参数描述如表2-3所示。输入用户名和密码，单击“确定”。

说明

默认连续5次登录失败，系统将对此用户进行锁定，锁定3分钟后可以重新登录。

图 2-3 首次登录

图 2-4 非首次登录

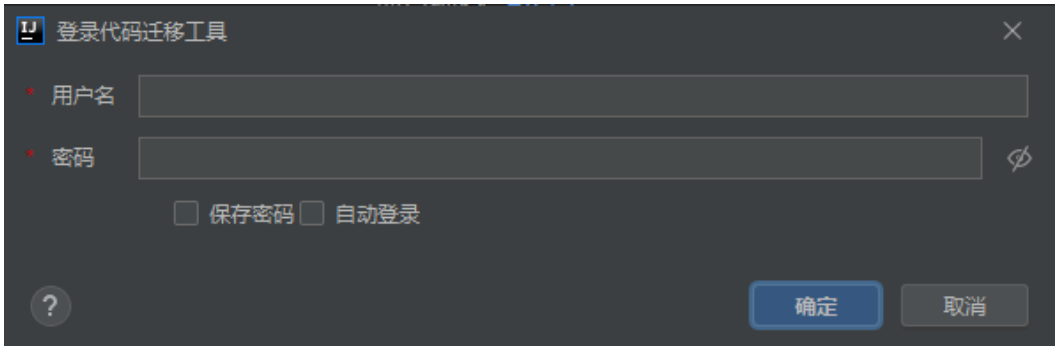


表 2-3 登录代码迁移工具参数描述

参数	说明
用户名	登录工具的用户。默认的管理员为**portadmin**。
密码	登录工具的用户密码。 工具安装完成后首次登录需要创建管理员密码，密码需要满足如下复杂度要求： ● 密码长度为8~32个字符 ● 必须包含大写字母、小写字母、数字、特殊字符（`~!@#$%^&*()-_=+|[{}];:'",<.>/?）中的两种及以上类型的组合 ● 密码不能是用户名 **说明** ● 首次登录的普通用户，系统提示修改初始密码，请按提示修改密码。 ● 为了保证安全，用户应定期修改自己的登录密码。
确认密码	再次输入设置的密码。
记住密码	当用户勾选“记住密码”登录后，再次登录该用户时，只需输入用户名，单击密码框会自动填充密码，以方便用户登录。记住密码功能只支持普通用户，不支持管理员账户（portadmin）。
自动登录	当用户同时勾选“记住密码”和“自动登录”登录后，插件重启后，无需输入用户名和密码，自动登录该用户。自动密码功能只支持普通用户，不支持管理员账户（portadmin）。

工具部署完成后，单击工具右侧的  按钮，选择服务器可切换其他已经部署了代码迁移工具的服务器。

2.3 使用加速库插件

鲲鹏开发套件是基于IntelliJ的一款扩展工具，加速库插件是其中的一个子工具。加速库插件即插即用，能够扫描代码文件中可使用鲲鹏加速库优化后的函数或汇编指令，生成可视化报告；编码时能够自动匹配鲲鹏加速库函数字典，智能提示、高亮、联想字典中可以替换的库和函数。关于鲲鹏加速库的详细介绍请参见**鲲鹏加速库**。

鲲鹏加速库插件支持的功能特性如下：

- 智能联想
  Coding时自动联想鲲鹏加速库优化后的相关函数
- 函数搜索
  支持鲲鹏加速库函数的代码定义跳转、函数搜索
- 语法高亮
  Coding时高亮鲲鹏加速库优化后的相关函数
- 加速分析

支持工程和文件扫描，识别出可以用鲲鹏加速库替换的函数

- 字典管理

  支持加速库函数字典管理，可线上（自动）和线下更新

- 在编辑器插件面板（Plugins）中根据关键字搜索“Kunpeng DevKit”或“Kunpeng Library Plugin”，或者在IntelliJ**应用商店网页**中找到Kunpeng Library Plugin，单击“安装”。（安装鲲鹏加速库插件需要依赖kunpeng foundation插件，点击弹窗中的“安装”自动下载kunpeng foundation插件。）
- 也可选择在**华为企业业务网站**下载并安装鲲鹏开发套件扩展安装包。步骤如下：

步骤1 登录**华为企业业务网站**下载鲲鹏开发套件扩展安装包和数字签名，确保与网站上的原始安装包一致。

说明

校验方法如下：

1. 在如下链接中获取校验工具和校验方法：

   https://support.huawei.com/enterprise/zh/tool/pgp-verify-TL1000000054

2. 下载以上链接中的《OpenPGP签名验证指南》，并根据指南进行软件包完整性检查。

步骤2 打开本地PC上的IntelliJ，在左上角单击File，选择Settings。

步骤3 在左侧菜单栏中先单击Plugins，然后单击顶部设置按钮，选择Install Plugin from Disk…后选中已经下载到本地的扩展安装包，单击OK。

----结束

鲲鹏加速库插件安装成功后，请参考以下图片使用加速库插件的功能。

说明

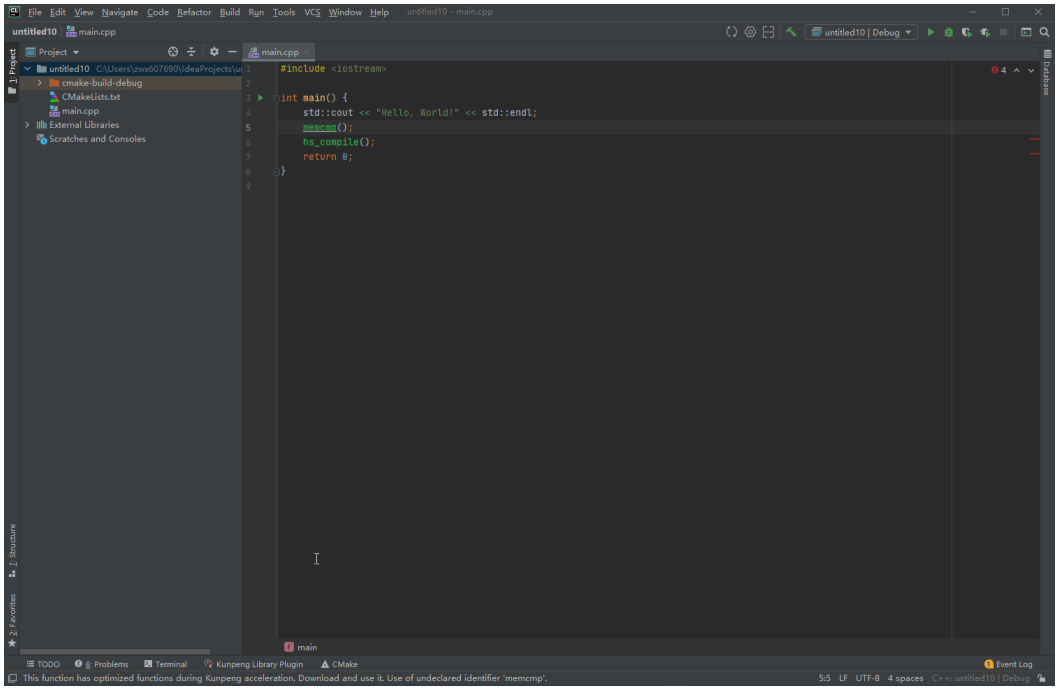
- IntelliJ鲲鹏加速库插件是基于CLion开发的，当前仅支持Clion，其它共IntelliJ平台的IDE产品可能也可以安装IntelliJ鲲鹏加速库插件，但当前未进行过验证，无法保证完美兼容。
- 安装鲲鹏加速库插件需要依赖kunpeng foundation插件，请在**华为企业业务网站**下载kunpeng-foundation插件安装包，先安装该插件，然后再安装鲲鹏加速库插件。
- 该插件支持IntelliJ2020.2以上版本。

加速分析

在IntelliJ的资源栏中，用户可以右键点击自己项目下的任意文件或者文件夹或者空白区域，此时会出现“鲲鹏加速分析”、“清除加速分析报告”、“查看加速分析报告”（英文环境下对应“Kunpeng accelerated analysis”、“Clear accelerated analysis report”、“View accelerated analysis report”）。

单击“鲲鹏加速分析”后，插件会分析工程里面依赖加速库的函数，并将扫描的结果在可视化面板以及IntelliJ的工具栏展示。点击工具栏中的问题可以跳转到函数方法，点击可视化界面中的“函数所在路径”可以跳转到函数方法。

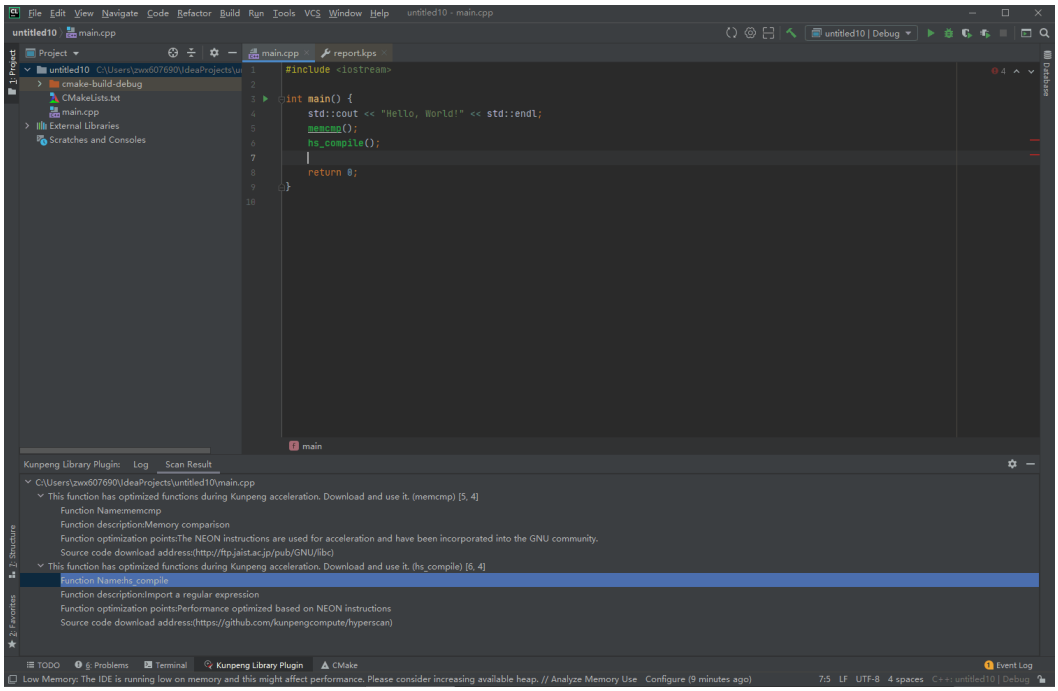
图 2-5 加速分析



编码辅助

当插件安装完成后，插件会自动下载鲲鹏的字典库数据，字典数据下载完成后，插件会启用针对加速库函数的语法高亮、函数说明、以及代码自动补全。

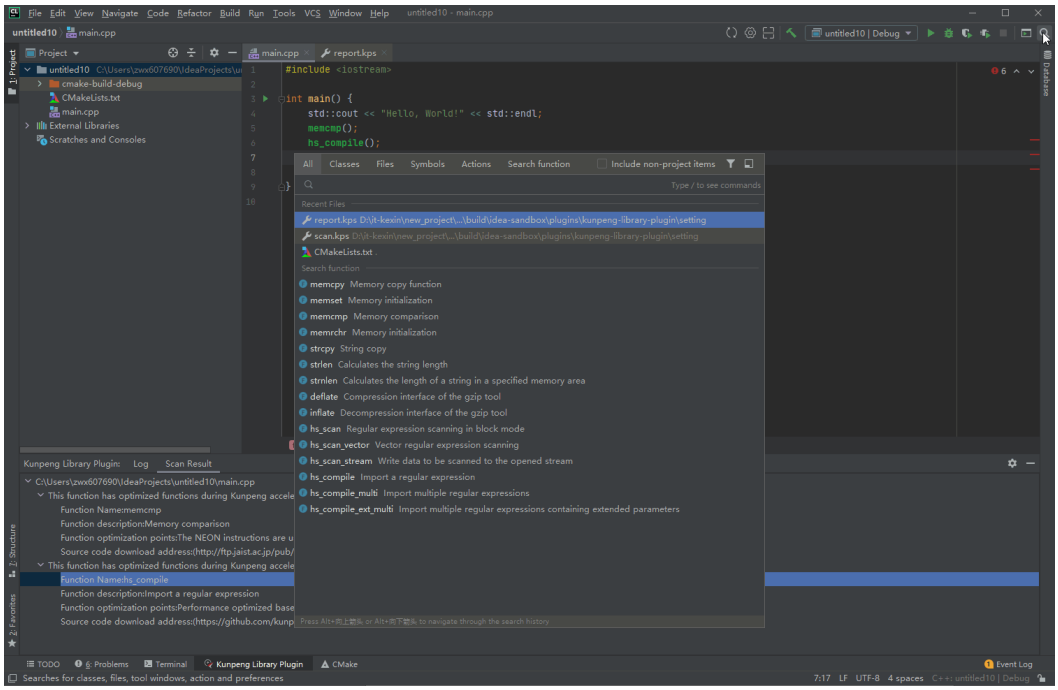
图 2-6 编码辅助（语法高亮/智能联想/智能提示）



函数搜索

单击编辑区右上角的  ，会出现函数搜索界面，提供对function函数的名称，以及Intrinsic函数的名称的搜索。单击function函数会跳转头文件定义，单击Intrinsic函数会打开Intrinsic的帮助文档地址。

图 2-7 加速库优化后的函数搜索



其他常用操作

- 支持配置远程字典地址，导入本地字典，是否开启加速库提示。

  说明

  - 用户启动鲲鹏加速库插件时会自动从远程字典地址获取最新字典数据。请提前确认网络代理是否正常配置，并能联网。
  - 如果要导入本地字典，请从以下连接获取：

    https://gitee.com/kunpengcompute/kunpengacclibdict/raw/master/dictionary.json

- 单击编辑区右上角  ，即可检测新版本。插件自身也存在自动检测机制。

2.4 安装鲲鹏编译插件

鲲鹏开发套件是基于Intellij的一款扩展工具，编译插件是其中的一个子工具。编译插件即插即用，支持一键安装鲲鹏GCC编译器，以及鲲鹏平台远程调试能力，支持的功能特性如下：

- 一键式部署

支持从Intellij编辑器插件面板（Plugins）中搜索“Kunpeng DevKit”或“Kunpeng Compiler”下载并在线安装插件，同时支持一键部署服务端鲲鹏GCC编译器

- 编译调试
  - 一键式安装鲲鹏GCC
  - 可视化编译配置任务，一键式任务运行
  - 远程单步调试C/C++代码
  - 编译调试过程信息实时展示
  - gtest框架用例树渲染及状态展示

当前已在如下服务端操作系统上验证：

- Ubuntu 18.04.4
- CentOS 7.0
- openEuler 20.03（LTS）（鲲鹏GCC编译器）

鲲鹏编译插件安装步骤如下：

- 在Intellij编辑器插件面板（Plugins）中根据关键字搜索“Kunpeng DevKit”或“Kunpeng Compiler Plugin”，或者在Intellij**应用商店网页**中找到Kunpeng Compiler Plugin，单击“安装”。（安装鲲鹏编译插件需要依赖kunpeng foundation插件，点击弹窗中的“安装”自动下载kunpeng foundation插件。）
- 也可选择在华为企业业务网站下载鲲鹏开发套件扩展安装包。步骤如下：

步骤1 登录**华为企业业务网站**下载鲲鹏编译插件安装包和数字签名，确保与网站上的原始安装包一致。

说明

校验方法如下：

1. 在如下链接中获取校验工具和校验方法：
   https://support.huawei.com/enterprise/zh/tool/pgp-verify-TL1000000054
2. 下载以上链接中的《OpenPGP签名验证指南》，并根据指南进行软件包完整性检查。

步骤2 打开本地PC上的Intellij，在左上角单击“文件”，选择“设置”。

步骤3 在左侧菜单栏中先单击“插件”，然后单击顶部设置按钮，选择“从磁盘安装插件…”后选中已经下载到本地的扩展安装包，单击OK。

----结束

安装完成后，重新启动Intellij，在左下角打开鲲鹏编译插件。

说明

- IntelliJ鲲鹏编译插件建议使用CLion环境，若用于IntelliJ IDEA环境，则无法显示C/C++语言一般性的语法关键字高亮，只能显示IntelliJ鲲鹏编译插件自身支持的鲲鹏语法关键字高亮。
- 安装鲲鹏编译插件需要依赖kunpeng foundation插件，请在**华为企业业务网站**下载kunpeng-foundation插件安装包，先安装该插件，然后再安装鲲鹏编译插件。
- 鲲鹏编译插件支持Intellij2020.2以上版本。

2.5 升级鲲鹏代码迁移工具

- 支持2.2.T1、2.2.T2、2.2.T2.SPC100、2.2.T2.SPC200、2.2.T2.SPC300版本升级到2.2.T3版本。

- 支持2.2.T3升级到2.2.1版本。
- 支持2.2.T3、2.2.1升级到2.2.T4版本。
- 支持2.2.T4、2.2.1升级到2.3.T10版本。

前提条件

- 联网状态下，系统会自动下载鲲鹏代码迁移工具**安装包**。
- 也可以手动将所需升级的鲲鹏代码迁移工具的软件包下载到本地，并上传到目标服务器用户主目录下，再使用一键式升级。

  获取软件包后，需要校验软件包，确保与网站上的原始软件包一致。详细步骤如下：

  a. 分别**下载**软件数字证书和软件。
  b. 在如下链接中获取校验工具和校验方法：

     https://support.huawei.com/enterprise/zh/tool/pgp-verify-TL1000000054
  c. 参见**2**中下载的《OpenPGP签名验证指南》进行软件包完整性检查。
- 升级前请确认鲲鹏代码迁移工具可以正常使用。
- 工具默认安装在“/opt/portadv”目录，升级前请确认安装空间至少保留8GB。

须知

- 不支持在分析任务执行过程中升级，请确保升级时没有任务在运行。
- 升级过程中请勿执行Ctrl+Z、Ctrl+C和重启系统操作。
- IDE插件只支持以Web模式升级工具，不支持以CLI模式升级工具。

升级操作

步骤1 单击“鲲鹏代码迁移插件”菜单右侧  ，在下拉菜单中选择“工具维护”，再选择“升级”。

步骤2 打开如**图2-8**所示界面，参数描述如**表2-4**所示。

配置参数后单击“检测连接”。若工具提示连接检测失败，可根据提示修改参数后重新尝试检测连接。工具提示“SSH连接检测成功”后，单击“开始升级”。

图 2-8 升级代码迁移工具

表 2-4 升级代码迁移工具参数说明

参数	说明
IP地址	待升级工具的服务器IP地址。
SSH端口	待升级工具的服务器SSH端口。
操作系统用户名	待安装工具的服务器操作系统的用户名。
选择SSH连接方式	连接方式可选择： ● 密码认证 ● 密钥认证 **说明** 选择“密钥认证”后需要导入“id_rsa”私钥文件，详细操作请参见**3.3.2 配置SSH密钥认证**。
操作系统用户名密码	待安装工具的服务器操作系统的用户密码。

步骤3 在终端输入服务器用户密码。

步骤4 在终端输入服务器root用户密码。

步骤5 在终端输入工具安装目录，默认为“/opt/portadv”。

步骤6 显示如下内容，确认是否升级鲲鹏代码迁移工具。

```
Please confirm that there are no running tasks(yes/no):
```

- 确认升级：输入**yes**后回车。
- 放弃升级：输入**no**后回车。

步骤7 在终端输入当前服务器的IP地址。

步骤8 在终端输入HTTPS端口，默认端口为8084。

步骤9 在终端输入tool端口，默认端口为7998。

步骤10 界面显示“工具升级成功”，单击“立即登录”按钮使用升级后的工具。

须知

升级失败的情况下，升级脚本会自动将工具回退到升级前的版本。

----结束

2.6 卸载鲲鹏代码迁移工具

须知

建议不要在分析任务执行过程中卸载，否则可能出现异常。
IDE插件只支持以Web模式卸载工具，不支持以CLI模式卸载工具。

前提条件

没有正在运行中的任务。

操作步骤

步骤1 单击工具右侧的  按钮，选择“工具维护”，单击“卸载”，打开如图2-9所示界面，参数描述如表2-5所示。

图 2-9 卸载代码迁移工具

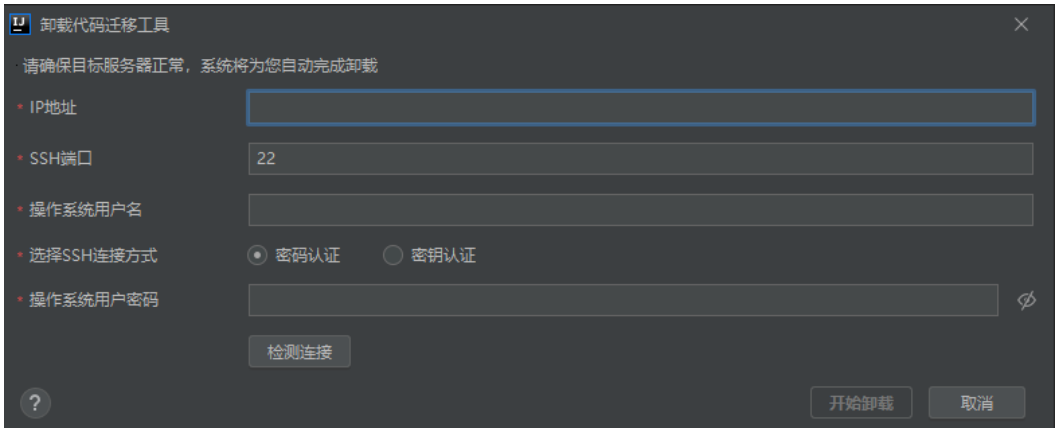


表 2-5 卸载代码迁移工具参数说明

参数	说明
IP地址	待卸载工具的服务器IP地址。
SSH端口	待卸载工具的服务器SSH端口。
操作系统用户名	待安装工具的服务器操作系统的用户名。

参数	说明
选择SSH连接方式	连接方式可选择： ● 密码认证 ● 密钥认证 **说明** 选择“密钥认证”后需要导入“id_rsa”私钥文件，详细操作请参见**3.3.2 配置SSH密钥认证**。
操作系统用户密码	操作系统用户密码。

步骤2 配置参数后单击“检测连接”。若工具提示连接检测失败，可根据提示修改参数后重新尝试检测连接。工具提示“SSH连接检测成功”后，单击“开始卸载”。

步骤3 在终端输入服务器普通用户密码。

步骤4 在终端输入服务器root用户密码。

步骤5 显示如下内容，确认是否卸载鲲鹏代码迁移工具。

```
Are you sure you want to uninstall porting advisor?(y/n)
```

- 确认卸载：输入**y** 后回车。
- 放弃卸载：输入**n**后回车。

步骤6 界面显示“工具卸载成功”，单击“完成”按钮完成卸载。

说明

当用户执行卸载命令时，如果有正在运行的任务，工具会给出提示“A task is running. Are you sure you want to uninstall porting advisor?(y/n)”。若用户仍选择卸载，当前运行的任务会直接中断。

----结束

3 代码迁移插件

3.1 介绍

鲲鹏代码迁移插件作为客户端调用服务端的功能，完成扫描迁移任务，可以对待迁移软件进行快速扫描分析，并提供专业的代码迁移指导，极大简化客户应用迁移到鲲鹏平台的过程。当客户有软件需要迁移到鲲鹏平台上时，可先用该工具分析可迁移性和迁移投入，以解决客户软件迁移评估中分析投入大、准确率低、整体效率低下的痛点。

代码迁移工具支持五个功能特性：

- 软件迁移评估：自动扫描并分析软件包（非源码包）、已安装的软件，提供可迁移性评估报告。
- 源码迁移：能够自动检查并分析出用户源码、C/C++/Fortran/Python软件构建工程文件、C/C++/Fortran/Python软件构建工程文件使用的链接库、x86汇编代码中需要修改的内容，并给出修改指导，以解决用户代码兼容性排查困难、迁移经验欠缺、反复依赖编译调错定位等痛点。
- 软件包重构：通过分析x86平台软件包（RPM格式、DEB格式）的软件构成关系及硬件依赖性，重构适用于鲲鹏平台的软件包。
- 专项软件迁移：基于鲲鹏解决方案的软件迁移模板，进行自动化迁移修改、编译、构建软件包，帮助用户快速迁移软件。
- 增强功能：支持软件代码质量的静态检查功能，如在64位环境中运行的兼容性检查、结构体字节对齐检查和内存一致性检查等增强功能。

  说明

  仅支持x86 Linux到Kunpeng Linux的扫描与分析，不支持Windows软件代码的扫描、分析与迁移。

3.2 特性指南

3.2.1 软件迁移评估

3.2.1.1 特性描述

软件迁移评估帮助用户分析用户X86环境上软件包安装路径中的SO库文件，并检查这些文件与鲲鹏平台的兼容性。

3.2.1.2 特性操作

软件迁移评估适用于x86服务器和基于鲲鹏916/920的服务器。

前提条件

已成功登录鲲鹏代码迁移工具。

说明

“/opt/portadv”为工具默认安装目录，下文以此默认路径为例，请根据实际情况替换。

操作步骤

步骤1 在页面左侧，单击  创建新任务。

步骤2 在创建分析任务区勾选“分析软件包”或“分析已安装软件”，并对以下参数进行配置。

说明

- “分析软件包”和“分析已安装软件”相互独立，可根据实际需求勾选其中的一个或所有。
- “分析已安装软件”只能在x86环境上勾选使用。

图 3-1 创建任务区参数项

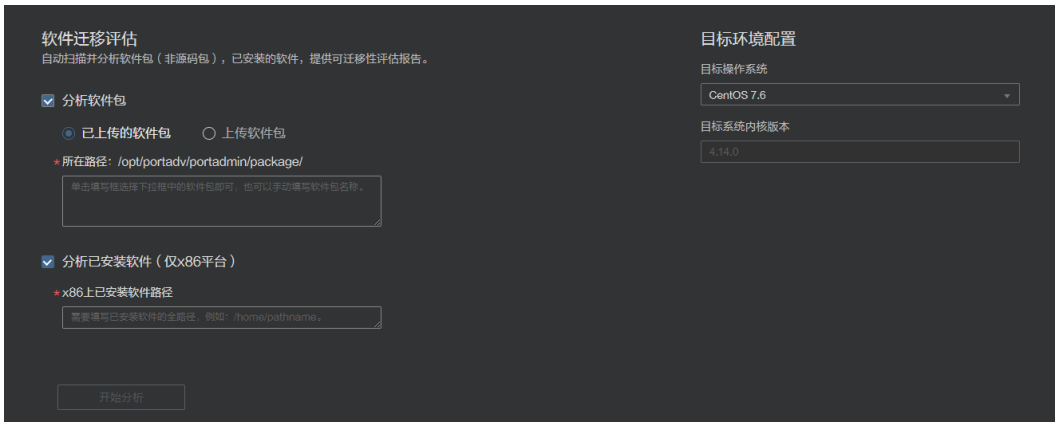


表 3-1 创建分析任务区参数项说明

参数	说明
分析软件包	
已上传的软件包	单击填写框选择下拉框中的软件包即可，也可以手动填写软件包名称，多个路径请通过“，”分隔。
上传软件包	单击“上传”按钮上传软件包。 **说明** ● 只允许同时上传一个软件包，软件包小于或等于1GB，且解压后小于或等于剩余磁盘空间的一半。 ● 软件包支持rpm、deb、jar、war、tar、zip、gz、tar.gz、tar.bz，tgz、tbz，tbz2、egg、whl类型。 ● 类Debian系统上可以扫描deb、jar、war、tar、zip、gz、tar.gz、tar.bz，tgz、tbz，tbz2、egg、whl类型。 ● 类RHEL系统上可以扫描rpm、jar、war、tar、zip、gz、tar.gz、tar.bz，tgz、tbz，tbz2、egg、whl类型。
分析已安装软件（仅x86平台）	
x86上已安装路径	需要填写已安装软件的绝对路径，例如：/home/pathname/ **说明** 需要保证porting用户对已安装软件的绝对路径有访问权限，权限不足会导致分析失败。
目标操作系统	选择目标系统版本。可选择： ● BC-Linux 7.6/7.7 ● CentOS 7.4/7.5/7.6/7.7/8.0/8.1/8.2 ● Deepin V15.2 ● Debian 10 ● EulerOS 2.8 ● iSoft 5.1 ● Kylin V10 SP1 ● LinxOS 6.0.90 ● NeoKylin V7U5/V7U6 ● openEuler 20.03 ● SUSE SLES 15.1 ● Ubuntu 18.04.x/20.04.x ● UOS 20 SP1 ● uosEuler 20 ● 更多 **说明** 点击“更多”后，根据页面上的步骤升级依赖字典，然后查看更新后的操作系统列表。

参数	说明
目标系统内核版本	选择目标系统内核版本。 • BC-Linux 7.6支持4.19.25 • BC-Linux 7.7支持4.19.25 • CentOS 7.4支持4.11.0 • CentOS 7.5支持4.14.0 • CentOS 7.6支持4.14.0 • CentOS 7.7支持4.18.0 • CentOS 8.0支持4.18.0 • CentOS 8.1支持4.18.0 • CentOS 8.2支持4.18.0 • Deepin V15.2支持4.19.34 • Debian 10支持4.14.0 • EulerOS 2.8支持4.19.36 • iSoft 5.1支持4.19.90 • Kylin V10 SP1支持4.19.90 • LinxOS 6.0.90支持4.19.0 • NeoKylin V7U5支持4.14.0 • NeoKylin V7U6支持4.14.0 • openEuler 20.03支持4.19.90 • SUSE SLES 15.1支持4.12.14 • Ubuntu 18.04.x支持4.15.0 • Ubuntu 20.04.x支持5.4.0 • UOS 20 SP1支持4.19.0 • uosEuler 20支持4.19.90

步骤3 单击“开始分析”，生成分析报告。

分析完成后，自动跳转至分析报告详情界面，如图3-2所示，参数描述如表3-2所示。

说明

- 用户可在任务进行过程中单击关闭，取消任务。
- 支持多个用户同时创建分析任务。
- 用户可以在分析报告的列表中，单击指定分析任务的报告名称查看分析报告。
- 分析报告的名称为生成报告的时间。

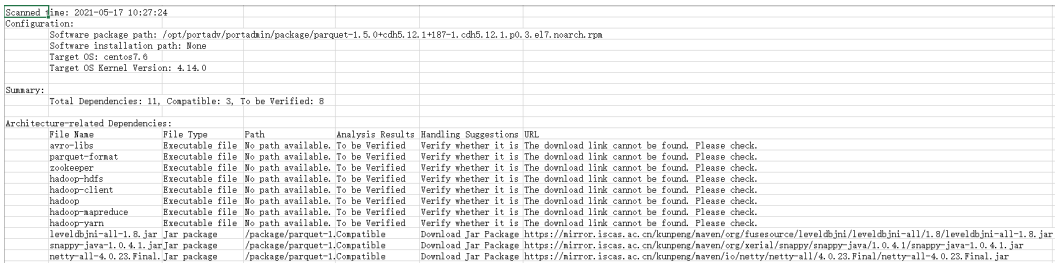
图 3-2 迁移评估报告结果

表 3-2 迁移评估报告结果参数说明

参数	说明
配置信息	显示软件安装包存放路径或软件包名称或x86上已安装软件路径，目标操作系统和目标系统内核版本。
与架构相关的依赖库文件	显示SO文件。 ● 针对兼容鲲鹏平台的动态库、静态库文件、软件包、可执行文件、Jar包等，用户可以直接单击处理建议中的“下载”，下载鲲鹏平台可用的文件，然后进行替换，或者下载鲲鹏版本源码，直接编译。对于一些未开源的依赖文件，工具无法提供下载URL，请自行获取后进行替换。 ● 针对鲲鹏平台兼容性未知的文件，请先在鲲鹏平台上验证。若不兼容，请联系供应方获取鲲鹏兼容版本，或获取源码并编译成鲲鹏兼容版本，或使用其他方案替代。

单击“下载报告（.csv）”，下载的分析报告如**图3-3**所示，参数描述如**表3-3**所示。

图 3-3 分析报告的基本信息



```
Scanned time: 2021-05-17 10:27:24
Configuration:
        Software package path: /opt/portadv/portadmin/package/parquet-1.5.0+cdh5.12.1+187-1.cdh5.12.1.p0.3.el7.noarch.rpm
        Software installation path: None
        Target OS: centos7.6
        Target OS Kernel Version: 4.14.0

Summary:
        Total Dependencies: 11, Compatible: 3, To be Verified: 8

Architecture-related Dependencies:
        File Name               File Type       Path            Analysis Results  Handling Suggestions  URL
        avro-libs               Executable file No path available. To be Verified  Verify whether it is The download link cannot be found. Please check.
        parquet-format          Executable file No path available. To be Verified  Verify whether it is The download link cannot be found. Please check.
        zookeeper               Executable file No path available. To be Verified  Verify whether it is The download link cannot be found. Please check.
        hadoop-hdfs             Executable file No path available. To be Verified  Verify whether it is The download link cannot be found. Please check.
        hadoop-client           Executable file No path available. To be Verified  Verify whether it is The download link cannot be found. Please check.
        hadoop                  Executable file No path available. To be Verified  Verify whether it is The download link cannot be found. Please check.
        hadoop-mapreduce        Executable file No path available. To be Verified  Verify whether it is The download link cannot be found. Please check.
        hadoop-yarn             Executable file No path available. To be Verified  Verify whether it is The download link cannot be found. Please check.
        leveldbjni-all-1.8.jar  Jar package     /package/parquet-1.Compatible     Download Jar Package https://mirror.iscas.ac.cn/kunpeng/maven/org/fusesource/leveldbjni/leveldbjni-all/1.8/leveldbjni-all-1.8.jar
        snappy-java-1.0.4.1.jar Jar package     /package/parquet-1.Compatible     Download Jar Package https://mirror.iscas.ac.cn/kunpeng/maven/org/xerial/snappy/snappy-java/1.0.4.1/snappy-java-1.0.4.1.jar
        netty-all-4.0.23.Final.jar Jar package  /package/parquet-1.Compatible     Download Jar Package https://mirror.iscas.ac.cn/kunpeng/maven/io/netty/netty-all/4.0.23.Final/netty-all-4.0.23.Final.jar
```


表 3-3 分析报告的基本信息说明

参数		说明
Scanned time		扫描时间。
Configuration	Software package path	软件包存放路径。
	Software installation path	软件已安装路径。
	Target OS	目标操作系统。
	Target OS Kernel Version	内核版本。
Summary	Total Dependencies: x, Compatible: x, To be Verified: x	显示依赖库文件总数，兼容鲲鹏平台的文件数，待验证文件数。
Architecture-related Dependencies		显示依赖库文件扫描的详细信息： • File Name：文件名称 • File Type：文件类型 • Path：存放路径 • Analysis Results：分析结果 • Handling Suggestions：处理建议 • URL：下载地址

----结束

3.2.2 源码迁移

3.2.2.1 特性描述

源码迁移功能分析用户C/C++/Fortran/Python软件的可迁移性。

3.2.2.2 特性操作

单击工具左侧的  ，阅读免责声明和GCC版本升级说明后，弹出如图3-4所示界面，参数描述如表3-4所示。配置参数后单击“开始分析”，右下角显示分析进度。

说明

以下内容中的“/opt/portadv”为服务端默认安装路径，如果为自定义安装，请使用对应的自定义安装路径替换。

图 3-4 新建源码迁移任务

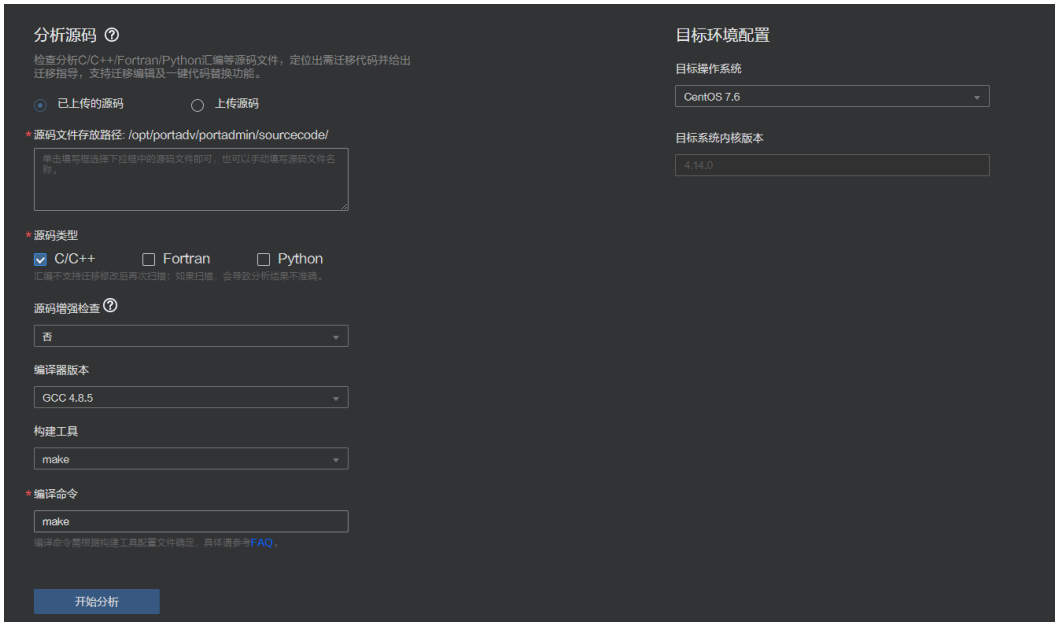


表 3-4 源码迁移参数说明

参数	说明
源码文件存放路径	需要填写相对路径，可以通过以下两种方式实现： ● 单击“上传”按钮上传压缩包（上传过程中自动解压）或文件夹。 ● 先将源码文件手动上传到服务器上本工具的指定路径下（例如：/opt/portadv/portadmin/sourcecode/），再单击填写框选择下拉框中的源码文件即可，也可以手动填写源码文件名称。 **说明** 在扫描含全汇编的源码时，用户需要先在源码文件的根目录下编译生成make.log，然后再打包上传。例如：用户要扫描含纯汇编的源码example，需要先在/opt/portadv/用户名/sourcecode/example/下生成make.log，再填写相对路径example。 在“上传”下拉菜单中可选择： ● 压缩包：选择上传源码文件压缩包，工具会自动解压。 **说明** ● 支持上传zip，tar，gz，tar.gz，tar.bz，tar.bz2，tar.xz，tgz，tbz，tbz2，txz格式的压缩包，只允许同时上传一个压缩包。源码文件压缩包小于或等于1GB，解压后小于或等于剩余磁盘空间的一半。 ● 工具会自动将压缩包解压至和压缩包同名的文件夹。例如上传的压缩包为test.zip，工具会自动解压至test文件夹。 ● 文件夹：选择上传本地解压的源码文件夹。 **说明** ● 只允许同时上传一个文件夹，文件夹小于或等于剩余磁盘空间的一半。 ● IE浏览器不兼容文件夹上传功能，上传文件夹需要使用其他浏览器，如Google Chrome，Microsoft Edge。
源码类型	选择源码类型。可选择： ● C/C++ ● Fortran ● Python
源码增强检查	选择是否对源码进行增强检查。默认为“否”。 启动源码增强检查可以提升源码在鲲鹏平台上运行的性能。例如检查结构体变量是否对齐鲲鹏Cache Line。 Suggestion: Set the __attribute__((__aligned__(128))) attribute for the struct structure in line XX.

参数	说明
编译器版本	选择编译器版本。 目标系统默认的编译器版本： ● BC-Linux 7.6默认为GCC 4.8.5 ● BC-Linux 7.7默认为GCC 4.8.5 ● CentOS 7.4默认为GCC 4.8.5 ● CentOS 7.5默认为GCC 4.8.5 ● CentOS 7.6默认为GCC 4.8.5 ● CentOS 7.7默认为GCC 4.8.5 ● CentOS 8.0默认为GCC 8.2 ● CentOS 8.1默认为GCC 8.3 ● CentOS 8.2默认为GCC 8.3 ● Deepin 15.2默认为GCC 6.3 ● Debian 10默认为GCC 8.3 ● EulerOS 2.8默认为GCC 7.3 ● iSoft 5.1默认为GCC 7.3 ● Kylin V10 SP1默认为GCC 7.3 ● LinxOS 6.0.90默认为GCC 6.3 ● NeoKylin V7U5默认为GCC 4.8.5 ● NeoKylin V7U6默认为GCC 4.8.5 ● openEuler 20.03默认为GCC 7.3 ● SUSE SLES 15.1默认为GCC 7.4 ● Ubuntu 18.04.x默认为GCC 7.3 ● Ubuntu 20.04.x默认为GCC 9.3 ● UOS 20 SP1默认为GCC 8.3 ● uosEuler 20默认为GCC 7.3 C/C++可选择： ● GCC 4.8.5/4.9.3/5.1/5.2/5.3/5.4/5.5/6.1/6.2/6.3/6.4/6.5/7.1/7.2/7.3/7.4/8.1/8.2/8.3/9.1/9.2/9.3 Fortran可选择： ● GFORTRAN 7 ● GFORTRAN 8 ● GFORTRAN 9
构建工具	选择构建工具。可选择： ● make ● cmake ● automake

参数	说明
编译命令	源码编译命令。 编译命令需根据构建工具配置文件确定，具体请参考通过构建工具配置文件识别编译命令。
目标操作系统	选择目标系统版本。可选择： ● BC-Linux 7.6/7.7 ● CentOS 7.4/7.5/7.6/7.7/8.0/8.1/8.2 ● Deepin V15.2 ● Debian 10 ● EulerOS 2.8 ● iSoft 5.1 ● Kylin V10 SP1 ● LinxOS 6.0.90 ● NeoKylin V7U5 ● NeoKylin V7U6 ● openEuler 20.03 ● SUSE SLES 15.1 ● Ubuntu 18.04.x ● Ubuntu 20.04.x ● UOS 20 SP1 ● uosEuler 20 ● 更多 **说明** 点击“更多”后，根据页面上的步骤升级依赖字典，然后查看更新后的操作系统列表。

参数	说明
目标系统内核版本	目标操作系统对应的内核版本。 • BC-Linux 7.6支持4.19.25 • BC-Linux 7.7支持4.19.25 • CentOS 7.4支持4.11.0 • CentOS 7.5支持4.14.0 • CentOS 7.6支持4.14.0 • CentOS 7.7支持4.18.0 • CentOS 8.0支持4.18.0 • CentOS 8.1支持4.18.0 • CentOS 8.2支持4.18.0 • Deepin V15.2支持4.19.34 • Debian 10支持4.14.0 • EulerOS 2.8支持4.19.36 • iSoft 5.1支持4.19.90 • Kylin V10 SP1支持4.19.90 • LinxOS 6.0.90支持4.19.0 • NeoKylin V7U5支持4.14.0 • NeoKylin V7U6支持4.14.0 • openEuler 20.03支持4.19.90 • SUSE SLES 15.1支持4.12.14 • Ubuntu 18.04.x支持4.15.0 • Ubuntu 20.04.x支持5.4.0 • UOS 20 SP1支持4.19.0 • uosEuler 20支持4.19.90
说明 如果用户所处的环境Glibc版本低于2.28，则无法使用最新的汇编文件自动翻译功能，如有需要，请单击页面上的“查看安装指导”，按照指示进行操作。	

分析完成后，自动跳转至分析报告详情界面，如图3-5所示，参数描述如表3-5所示。

说明

- 也可以在分析报告的列表中，单击指定分析任务的报告名称查看分析报告。
- 分析报告的名称为生成报告的时间。

图 3-5 分析报告

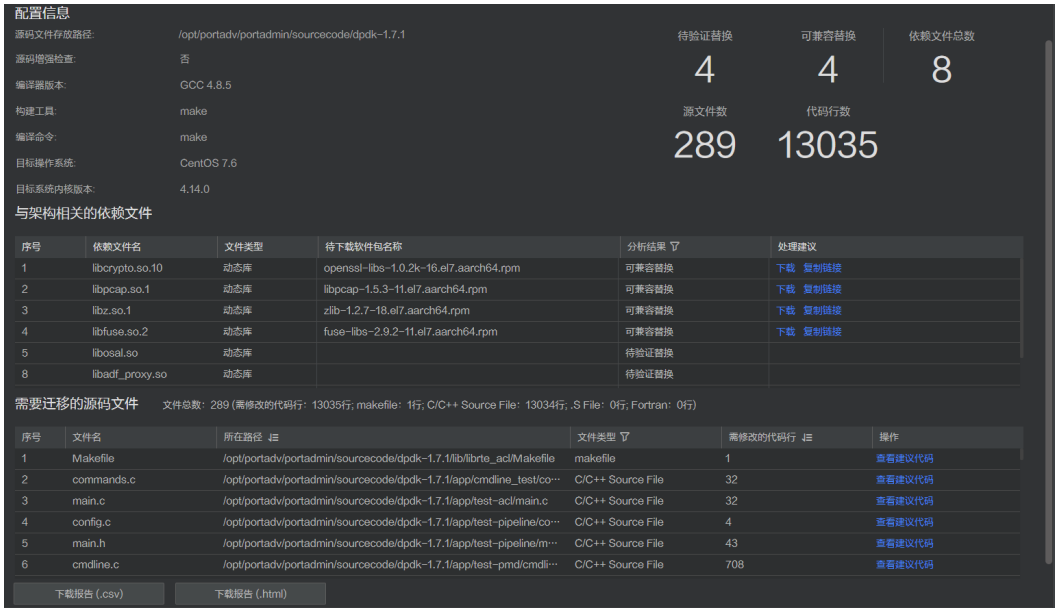
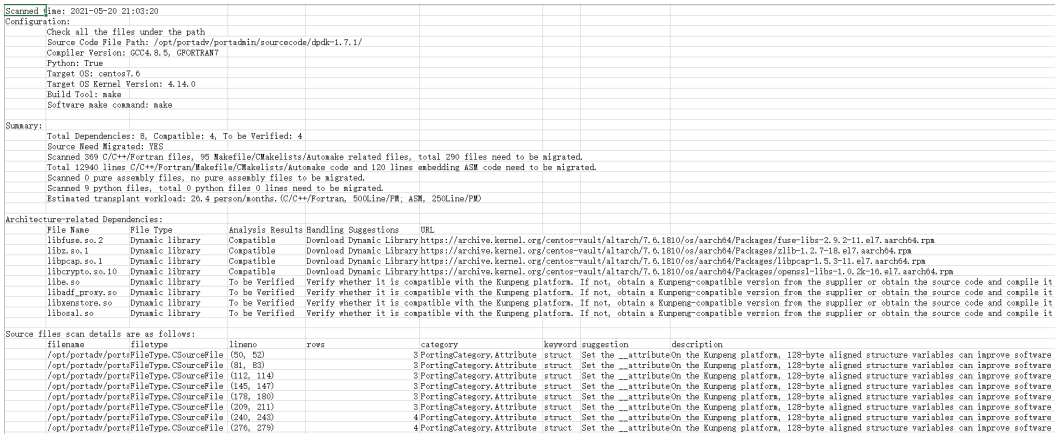


表 3-5 迁移报告结果参数说明

参数	说明
配置信息	显示源码文件存放路径、是否执行源码增强检查、编译器版本、构建工具、编译命令、目标操作系统和目标系统内核版本。
与架构相关的依赖库文件	显示SO文件。 • 针对兼容鲲鹏平台的动态库、静态库文件、软件包、可执行文件、Jar包等，用户可以直接单击处理建议中的“下载”，下载鲲鹏平台可用的文件，然后进行替换，或者下载鲲鹏版本源码，直接编译。对于一些未开源的依赖文件，工具无法提供下载URL，请自行获取后进行替换。 • 针对鲲鹏平台兼容性未知的文件，请先在鲲鹏平台上验证。若不兼容，请联系供应方获取鲲鹏兼容版本，或获取源码并编译成鲲鹏兼容版本，或使用其他方案替代。
需要迁移的源文件	显示需要迁移的源文件总数和需要修改的代码行数，通过单击操作列“查看建议源码”可以快速进入对应的源码迁移建议页面。 文件类型包括C/C++ Source File，Fortran，makefile，Python以及.S File，可通过筛选查看具体类型的文件。 文件所在的路径根据Unicode编码排序，需修改的代码行根据行数排序。

单击“下载报告 (.csv)”，下载的分析报告如**图3-6**所示，参数描述如**表3-6**所示。

图 3-6 下载的分析报告



Scanned time: 2021-05-20 21:03:20
Configuration:
Check all the files under the path
Source Code File Path: /opt/portadv/portadmin/sourcecode/dpdk-1.7.1/
Compiler Version: GCC4.8.5, GFORTRANT
Python: True
Target OS: centos7.6
Target OS Kernel Version: 4.14.0
Build Tool: make
Software make command: make

Summary:
Total Dependencies: 8, Compatible: 4, To be Verified: 4
Source Need Migrated: YES
Scanned 369 C/C++/Fortran files, 95 Makefile/CMakelists/Automake related files, total 290 files need to be migrated.
Total 12940 lines C/C++/Fortran/Makefile/CMakelists/Automake code and 120 lines embedding ASM code need to be migrated.
Scanned 0 pure assembly files, no pure assembly files to be migrated.
Scanned 9 python files, total 0 python files 0 lines need to be migrated.
Estimated transplant workload: 26.4 person/months.(C/C++/Fortran, 500Line/PM; ASM, 250Line/PM)

Architecture-related Dependencies:

File Name	File Type	Analysis Results	Handling Suggestions	URL
libfuse.so.2	Dynamic library	Compatible	Download Dynamic Library	https://archive.kernel.org/centos-vault/altarch/7.6.1810/os/aarch64/Packages/fuse-libs-2.9.2-11.el7.aarch64.rpm
libz.so.1	Dynamic library	Compatible	Download Dynamic Library	https://archive.kernel.org/centos-vault/altarch/7.6.1810/os/aarch64/Packages/zlib-1.2.7-18.el7.aarch64.rpm
libpcap.so.1	Dynamic library	Compatible	Download Dynamic Library	https://archive.kernel.org/centos-vault/altarch/7.6.1810/os/aarch64/Packages/libpcap-1.5.3-11.el7.aarch64.rpm
libcrypto.so.10	Dynamic library	Compatible	Download Dynamic Library	https://archive.kernel.org/centos-vault/altarch/7.6.1810/os/aarch64/Packages/openssl-libs-1.0.2k-16.el7.aarch64.rpm
libe.so	Dynamic library	To be Verified	Verify whether it is compatible with the Kunpeng platform. If not, obtain a Kunpeng-compatible version from the supplier or obtain the source code and compile it	
libadf_proxy.so	Dynamic library	To be Verified	Verify whether it is compatible with the Kunpeng platform. If not, obtain a Kunpeng-compatible version from the supplier or obtain the source code and compile it	
libxenstore.so	Dynamic library	To be Verified	Verify whether it is compatible with the Kunpeng platform. If not, obtain a Kunpeng-compatible version from the supplier or obtain the source code and compile it	
libosal.so	Dynamic library	To be Verified	Verify whether it is compatible with the Kunpeng platform. If not, obtain a Kunpeng-compatible version from the supplier or obtain the source code and compile it	

Source files scan details are as follows:

filename	filetype	lineno	rows	category	keyword	suggestion	description
/opt/portadv/ports	FileType.CSourceFile	(50, 52)	3	PortingCategory.Attribute	struct	Set the __attribute	On the Kunpeng platform, 128-byte aligned structure variables can improve software
/opt/portadv/ports	FileType.CSourceFile	(81, 83)	3	PortingCategory.Attribute	struct	Set the __attribute	On the Kunpeng platform, 128-byte aligned structure variables can improve software
/opt/portadv/ports	FileType.CSourceFile	(112, 114)	3	PortingCategory.Attribute	struct	Set the __attribute	On the Kunpeng platform, 128-byte aligned structure variables can improve software
/opt/portadv/ports	FileType.CSourceFile	(145, 147)	3	PortingCategory.Attribute	struct	Set the __attribute	On the Kunpeng platform, 128-byte aligned structure variables can improve software
/opt/portadv/ports	FileType.CSourceFile	(178, 180)	3	PortingCategory.Attribute	struct	Set the __attribute	On the Kunpeng platform, 128-byte aligned structure variables can improve software
/opt/portadv/ports	FileType.CSourceFile	(209, 211)	3	PortingCategory.Attribute	struct	Set the __attribute	On the Kunpeng platform, 128-byte aligned structure variables can improve software
/opt/portadv/ports	FileType.CSourceFile	(240, 243)	4	PortingCategory.Attribute	struct	Set the __attribute	On the Kunpeng platform, 128-byte aligned structure variables can improve software
/opt/portadv/ports	FileType.CSourceFile	(276, 279)	4	PortingCategory.Attribute	struct	Set the __attribute	On the Kunpeng platform, 128-byte aligned structure variables can improve software

表 3-6 分析报告的基本信息说明

参数		说明
Scanned time		扫描时间。
Configuration	Source Code File Path	扫描的源码文件存放路径。
	Compiler Version	编译器版本。
	Python	是否扫描Python源码。
	Target OS	目标操作系统。
	Target OS Kernel Version	内核版本。
	Build Tool	构建工具。
	Software make command	编译命令。
Summary	Total Dependencies: x, Compatible: x, To be Verified: x	显示依赖库文件总数，兼容鲲鹏平台的文件数，待验证文件数。
	Source Need ported	源码是否需要迁移。 ● Yes ● No
	Scanned xx C/C++/Fortran files, xx Makefile/ CMakelists/Automake related files, total xx files need to be ported.	显示需要迁移的C/C++/Fortran文件和Makefile/CMakelists/Automake related文件总数以及文件个数。
	Total xx lines C/C++/ Makefile/CMakelists/ Automake code and xx lines embedding ASM code need to be ported.	显示源码总行数以及需要迁移的内联汇编代码行数。

参数		说明
	Scanned xx pure assembly files, xx pure assembly files to be ported.	显示全汇编文件总数以及需要迁移的汇编文件个数和汇编代码行数。
	Scanned xx python files, total xx python files xx lines need to be ported.	显示Python文件总数以及需要迁移的Python文件个数和Python代码行数。
	Estimated transplant workload: xx person/ months.(C/C++/Fortran, 500 Line/PM; ASM, 250Line/PM)	显示源码迁移预估人力（x/人月）。 预估标准：1人月迁移工作量=500行C/C++/Fortran源代码，或250行汇编代码。
Architecture-related Dependencies		显示依赖库文件扫描的详细信息： ● File Name：文件名称 ● File Type：文件类型 ● Analysis Results：分析结果 ● Handling Suggestions：处理建议 ● URL：下载地址
Source files scan details are as follows:		显示源文件分析报告的详细信息： filename：扫描文件全路径。 filetype：扫描文件的类型。 lineno：函数在文件中的行号。 rows：函数在文件中的总行数。 category：关键字所属类型。 keyword：关键字名称。 suggestion：迁移建议或者提示建议。 description：关键字用法描述。

Module函数的描述说明如下：

```
current usage:
add_library(test1 STATIC attr_gcc5.1.c builtin_gcc5.1.c )
The general signature is:
  add_library(<name> [STATIC | SHARED | MODULE]
              [EXCLUDE_FROM_ALL]
              [source1] [source2 ...])
For details: https://cmake.org/cmake/help/v3.13/command/add_library.html
```

- current usage：当前文档中函数用法。
- The general signature is：官方通用函数用法。
- For details：对应Module函数的官方详细帮助文档链接。

在“迁移报告”的操作中直接单击“查看建议源码”，可以查看源码迁移建议，参考建议修改源码后，可使用快捷键“ctrl+s”保存修改。

3.2.3 软件包重构

3.2.3.1 特性描述

软件包重构能够分析用户提供的x86软件包，并根据用户提供的资源文件尝试构建鲲鹏平台软件包。软件包重构需要用户在鲲鹏平台使用。

3.2.3.2 特性操作

单击工具左侧的  ，打开如图3-7所示界面，参数描述如表3-7所示。配置参数后单击“下一步”执行重构。

说明

- “/opt/portadv”为默认安装目录，下文以此默认路径为例，请根据实际情况替换。
- RPM包只能在类RHEL系统上执行，重构过程中需要依赖系统组件rpmrebuild/rpmbuild/rpm2cpio，请提前检查系统环境是否已满足。

  安装rpmrebuild：自行下载rpmrebuild组件，并上传至服务器执行**rpm -ivh xxx.rpm**命令安装。

  安装rpmbuild：执行**yum list | grep rpm-build**命令检查yum源中是否包含rpmbuild的rpm包。如果存在，执行**yum install -y rpm-build**安装。如果不存在，请自行下载对应组件，并上传至服务器执行**rpm -ivh xxx.rpm**命令安装。
- DEB包只能在类Debian系统上执行，重构过程中需要依赖系统组件ar/dpkg-deb，请提前检查系统环境是否已满足。
- 如果RPM包或者DEB包里面包含JAR包，请检查系统是否存在JAR命令，如果不存在，请安装JDK工具。
- 软件包重构结果默认保存在"/opt/portadv/xx/report/packagerebuild/task_id/"路径（xx代表用户名，task_id即任务创建时间），执行完成后您可以进入该路径查看已重构的软件包。

图 3-7 软件包重构

表 3-7 软件包重构参数说明

参数	说明
已上传的软件包	单击填写框选择下拉框中的软件包即可，也可以手动填写软件包名称。 **说明** 如果构建的软件安装包大于1GB，则只能手动将软件安装包上传至服务器。
上传软件包	单击“上传”按钮上传软件安装包，上传成功自动填写安装包名称。 **说明** 只允许同时上传一个软件包，软件包不超过1GB。
软件包存放路径	软件包的保存路径，即当前登录用户的工作空间。
配置依赖文件	上传重构软件包需要的依赖文件（SO库，JAR文件）。
依赖文件存放路径	上传的依赖文件默认保存在“/opt/portadv/data”目录下。 单击“上传”按钮上传本地依赖文件。 **说明** • 工具仅提供部分JAR文件自动下载功能，文件将下载到“/opt/portadv/data”目录下，其他软件包重构过程中需要的依赖文件，请自行上传。 • 可同时上传多个依赖文件，框内如有同名文件时，将被选择的新文件覆盖，且每个最大不超过1GB。
授权访问外部网络获取重构软件包所需要的依赖文件	软件包重构过程中需要访问外部网络获取重构所需的依赖文件。如果是隔离网络环境请自行配置代理。

重构成功后，可下载重构包，查看执行结果。

若重构失败，可以点击页面上的历史报告，查看重构失败原因并按建议进行处理。

3.2.4 专项软件迁移

3.2.4.1 特性描述

专项软件迁移提供开源软件迁移、调优的工具化解决方案。用户可以按照解决方案的粒度选择相应的软件，进行工具化快速迁移软件。专项软件迁移包含软件下载、修改、编译和软件包构建功能，需要用户在基于鲲鹏的服务器环境使用。

3.2.4.2 特性操作

单击工具左侧  打开“迁移前必读”对话框，勾选“我已阅读以上文字”，单击“确认”，打开如图3-8所示。

图 3-8 专项软件迁移

进入专项软件迁移，支持迁移的软件如表3-8所示。

说明

专项软件迁移过程中可能会安装依赖组件，修改系统配置，下载迁移软件并进行修改、编译、构建等操作，迁移前请仔细阅读步骤描述。

表 3-8 专项软件迁移

鲲鹏解决方案	软件名称	版本
大数据	hdp-hadoop	3.1.1.3.1.0.0-78
	cdh-kudu	1.7.0-5.16.1
	hdp-hbase	2.0.2.3.1.0.0-78
	cdh-impala	5-2.12.0-5.16.1
	hdp-hive	3.1.0.3.1.0.0-78
	hdp-spark2	2.3.2.3.1.0.0-78
数据库	MySQL	8.0.17
Web	.NET-Core	3.1
	Nginx	1.14.2
	Tengine	2.2.2
高性能计算	OpenFOAM	v1906
注1：Web解决方案下的Nginx/Tengine不能和其他解决方案下的软件在同一服务器上迁移。 注2：Web解决方案下的.NET-Core只展示静态页面，不支持迁移功能。详细步骤请参见迁移描述中的链接。		

在表格上方选择指定的鲲鹏解决方案类型，在目标软件名称对应的“操作”列中单击“迁移”。打开专项软件迁移详情页面，如图3-9所示，参数描述如表3-9所示。

说明

专项软件迁移支持的所有解决方案软件的迁移模板都存放在服务器“/opt/portadv/resource/migration”路径下（“/opt/portadv”为工具安装目录，请根据实际情况替换），所有用户共享此路径下的模板。

图 3-9 专项软件迁移详情

表 3-9 基本信息区域参数说明

参数	说明
软件名称	显示软件名称。
版本	显示软件版本。
华为maven源仓库	显示maven源仓库地址。 **说明** 只有涉及到华为maven源仓库时才会显示。
描述	显示迁移描述，并提供鲲鹏社区详细迁移指导的链接地址。
操作	提供到迁移详情页的链接。

在“执行步骤”区域选择需要执行的步骤，单击“开始迁移”。

说明

- “环境检查”区域下的“前置条件”默认自动执行，无需手动检查确认。
- 专项软件迁移过程中需要访问外部网络获取所需的资源文件。如果是隔离网络环境请自行配置代理。
- 单击开始迁移后，选中的需要执行的步骤前会显示执行状态。
- 专项软件迁移会对您的软件进行自动化迁移修改、编译、构建软件包。
- 专项软件迁移过程中可以离开当前页面进行其他操作，但不能备份，恢复，升级软件迁移模板。
- 用户可在任务进行过程中单击关闭，取消任务。
- 执行专项软件迁移时的工作空间为“/opt/portadv/xx/migration”（xx代表用户名，“/opt/portadv”为工具安装目录，请根据实际情况替换），所有的执行过程文件和执行结果文件均存放在此工作空间下。
- 专项软件迁移的日志默认保存在 “/opt/portadv/logs/porting.log”文件中（“/opt/portadv”为工具安装目录，请根据实际情况替换），可在该日志文件中查看迁移失败原因并按建议进行处理。

3.2.5 增强功能

3.2.5.1 特性描述

鲲鹏代码迁移工具提供下面三种增强功能：

- 64位运行模式检查

  对用户C/C++软件从32位模式迁移到64位模式进行检查。工具强制以64位模式编译用户软件，并通过编译选项发现从32位模式迁移到64位模式的必要修改，并提示用户进行进一步检查。

- 结构体字节对齐检查

  用户软件中的结构体变量进行检查，分析其内存分配情况，并反馈用户。

- 内存一致性检查

  对用户软件迁移到鲲鹏平台可能存在的内存一致性问题进行检查、修复。自动修复工具需要更新用户使用的GCC编译器，随后在用户软件的编译过程中自动完成内存一致性问题的修复。

3.2.5.2 特性操作

64位运行模式检查和结构体字节对齐检查功能适用于x86服务器，支持x86平台GCC 4.8.5~GCC 9.3版本32位应用向64位应用的运行模式检查和字节对齐检查。内存一致性检查功能适用于鲲鹏平台，检查源码中存在的内存一致性问题。

3.2.5.2.1 64 位运行模式检查

64位运行模式检查就是将x86平台GCC4.8.5~GCC9.3版本原32位的应用迁移到64位平台上，进行迁移检查并给出修改建议。

前提条件

已成功登录鲲鹏代码迁移工具。

说明

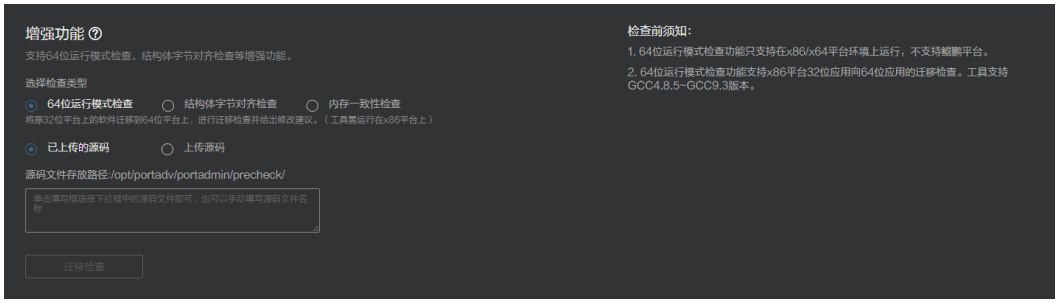
“/opt/portadv”为工具默认安装目录，下文以此默认路径为例，请根据实际情况替换。

操作步骤

步骤1 在页面左侧，单击  创建增强任务。

选择“64位运行模式检查”，如图3-10所示。

图 3-10 64 位运行模式检查



步骤2 可以通过以下两种方式选择待检查的源码：

- 选择“已上传的源码”：单击填写框选择下拉框中的源码文件即可，也可以手动填写源码文件。
- 选择“上传源码”：单击“上传”按钮上传压缩包（上传过程中自动解压）或文件夹。

  说明

  - 支持上传zip，tar，tar.gz，tar.bz，tar.bz2，tar.xz，tgz，tbz，tbz2，txz格式的压缩包，只允许同时上传一个压缩包。源码文件压缩包小于或等于1GB，解压后小于或等于剩余磁盘空间的一半。
  - 只允许同时上传一个文件夹，文件夹小于或等于剩余磁盘空间的一半。

步骤3 单击“迁移检查”，开始迁移检查。迁移检查完成后，自动跳转至迁移检查报告界面，如图3-11所示。

图 3-11 迁移检查报告

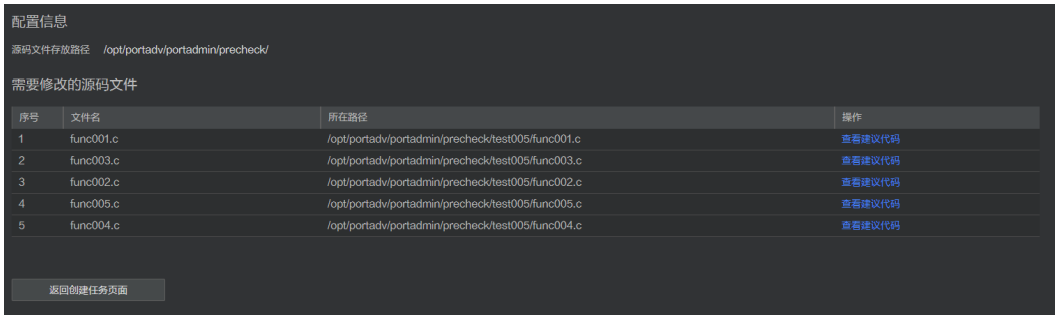
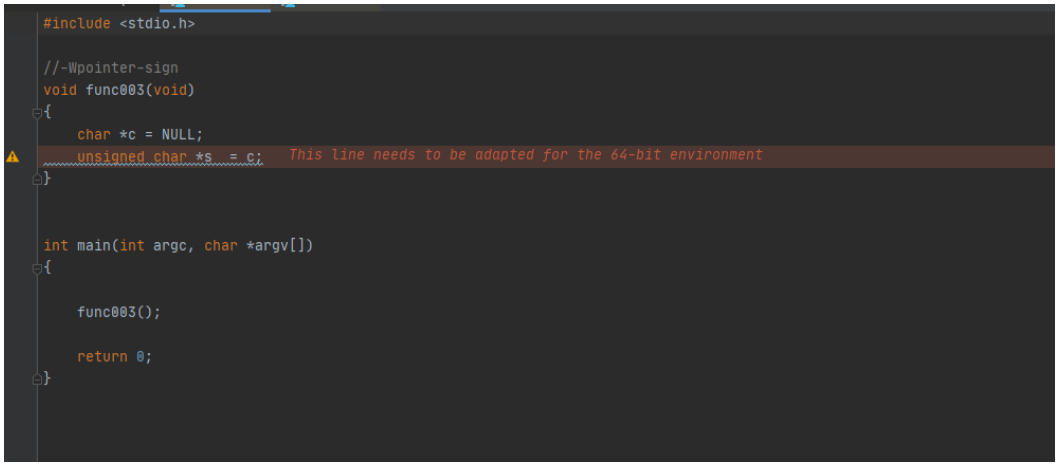


图 3-12 查看建议代码



说明

- 支持多个用户同时创建64位运行模式检查任务。
- 用户可在任务进行过程中单击关闭，取消任务。
- 若检测结果为源码不需要修改，则会弹出成功界面，不会出现预检报告界面。

----结束

3.2.5.2.2 结构体字节对齐检查

结构体字节对齐检查就是在需要考虑字节对齐时，检查源码中结构体类型变量的字节对齐检查。

前提条件

已成功登录鲲鹏代码迁移工具。

说明

“/opt/portadv”为工具默认安装目录，下文以此默认路径为例，请根据实际情况替换。

操作步骤

步骤1 在页面左侧，单击  创建增强任务。

选择“结构体字节对齐检查”，如图3-13所示。

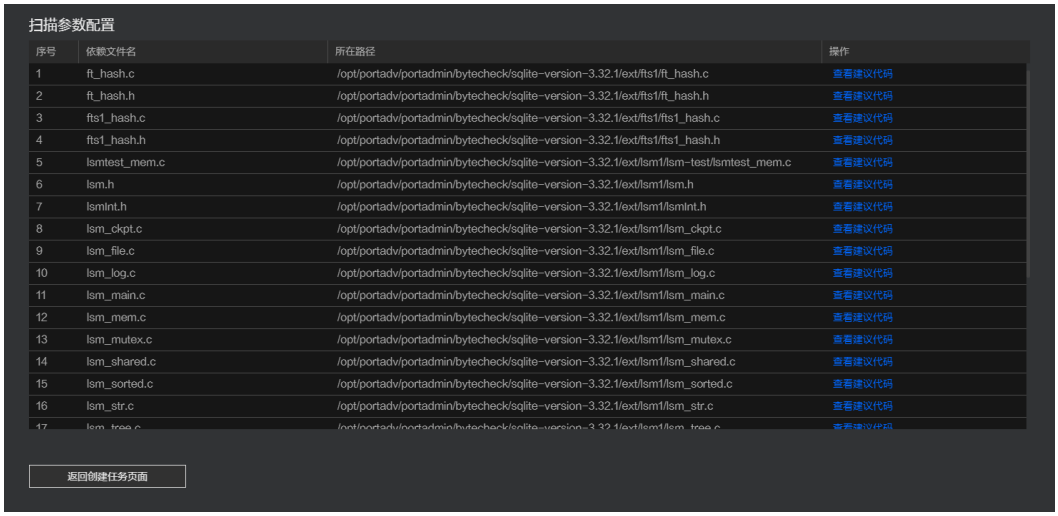
图 3-13 结构体字节对齐检查

表 3-10 结构类型定义对齐检查参数说明

参数	说明
源码文件存放路径	• 选择“已上传的源码”：单击填写框选择下拉框中的源码文件即可，也可以手动填写源码文件。 • 选择“上传源码”：单击“上传”按钮上传压缩包（上传过程中自动解压）或文件夹。 **说明** – 支持上传zip，tar，tar.gz，tar.bz，tar.bz2，tar.xz，tgz，tbz，tbz2，txz格式的压缩包，只允许同时上传一个压缩包。源码文件压缩包小于或等于1GB，解压后小于或等于剩余磁盘空间的一半。 – 只允许同时上传一个文件夹，文件夹小于或等于剩余磁盘空间的一半。
构建工具	选择构建工具。可选择： • make • cmake • automake
编译命令	源码编译命令。 编译命令需根据构建工具配置文件确定，具体请参考**3.3.6 通过构建工具配置文件识别编译命令**。

步骤2 单击“对齐检查”，开始字节对齐检查。检查完成后，自动跳转至迁移检查报告界面，如**图3-14**所示。

图 3-14 对齐检查报告



单击“查看建议代码”可进入“对齐检查报告”界面，如图3-15所示。

图 3-15 建议代码

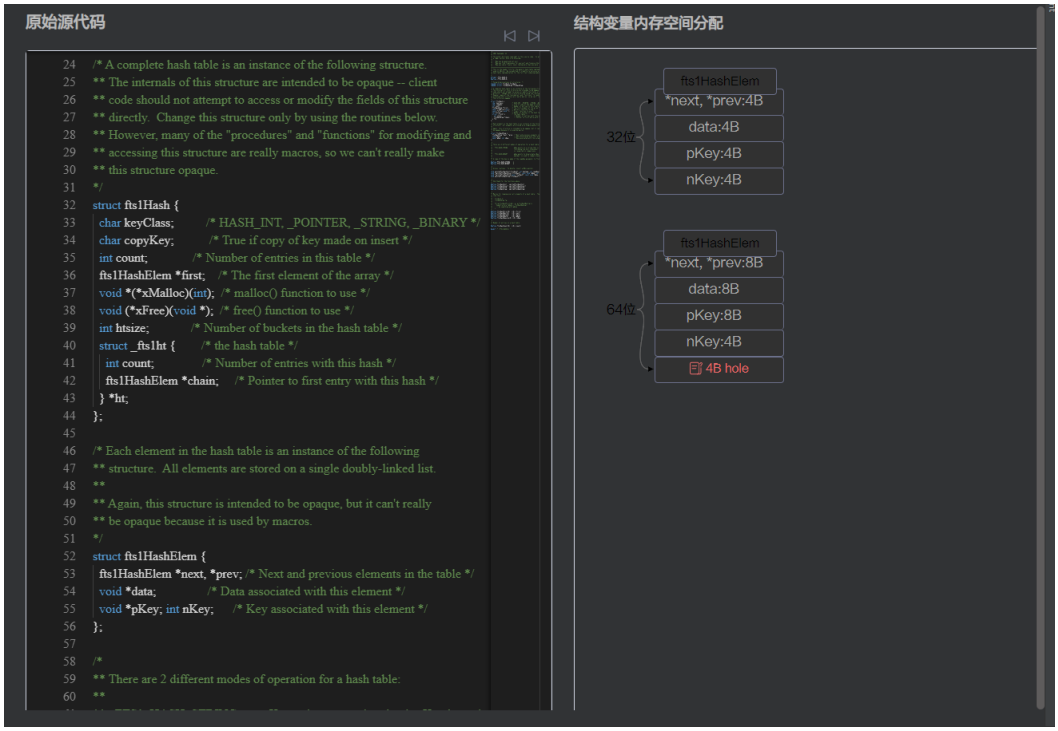


表 3-11 建议代码界面参数说明

参数	说明
原始源代码	原始源代码。
结构变量内存空间分配	显示代码中需要对齐的32位和64位字节内存空间。

说明

- 支持多个用户同时创建字节对齐检查任务。
- 用户可在任务进行过程中单击关闭，取消任务。
- 对于64位运行模式检查和结构体字节对齐检查，在同一时刻同一用户只能进行其中一个。
- 用户可以单击原始源代码模块右上角的上下键，进行上下切换。

----结束

3.2.5.2.3 内存一致性检查

内存一致性检查就是检查源码迁移在鲲鹏平台运行时可能存在的内存一致性问题，并提供插入内存屏障的建议。

说明

- 由于运行内存一致性检查时，生成的中间文件较大，如果用户想要使用该功能，请确保安装工具的所在磁盘空间足够大，根据经验数据，每10w行代码需要100G左右的磁盘空间。
- 源码规模过大可能会导致占用资源过多，建议代码量不超过10万行。

前提条件

已成功登录鲲鹏代码迁移工具。

说明

“/opt/portadv”为工具默认安装目录，下文以此默认路径为例，请根据实际情况替换。

操作步骤

步骤1 在页面左侧，单击  创建增强任务。

选择“内存一致性检查”，如图3-16所示。

图 3-16 内存一致性检查

步骤2 选择“检查模式”。

- 静态检查，请继续执行**步骤3**。
- 编译器自动修复，需要用户根据页面提供的操作步骤进行操作，详细请参见**编译器自动修复工具使用指导**。

  说明

  静态检查的结果误报少、修复率最高为60%，编译器自动修复结果不会漏报，误报率比静态检查高，修复率可保证100%。误报越多，性能影响相对越大。不同软件下编译的修复率不同，性能影响也不同。

步骤3 选择“文件上传类型”。

- 源码文件上传，请继续执行**步骤4**~**步骤6**。

说明

静态检查工具需要使用clang编译工具执行用户项目的构建，并生成BC文件，作为静态检查工具的输入，请确保上传代码可以使用clang编译工具编译通过。

- BC文件上传，请继续执行**步骤7~步骤8**。

步骤4 填写“源码文件存放路径”。可以通过以下两种方式实现：

- 选择“已上传的源码”：单击填写框选择下拉框中的源码文件即可，也可以手动填写源码文件。
- 选择“上传源码”：单击“上传”按钮上传压缩包（上传过程中自动解压）或文件夹。

说明

- 支持上传zip，tar，gz，tar.gz，tar.bz2，bz，bz2格式的压缩包，只允许同时上传一个压缩包。源码文件压缩包小于或等于1GB，解压后小于或等于剩余磁盘空间的一半。
- 只允许同时上传一个文件夹，文件夹小于或等于剩余磁盘空间的一半。

步骤5 填写编译命令，然后单击“下一步”。

解析编译命令后生成BC文件。

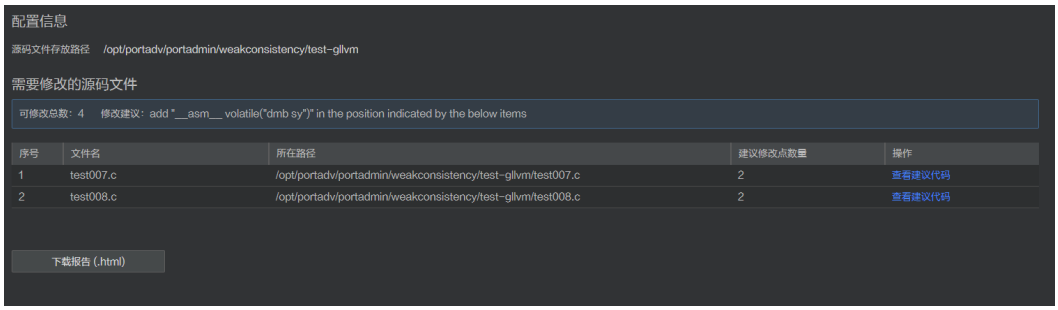
说明

- 编译命令支持make、cmake、configure、shell命令以及shell脚本，其中使用make命令时不支持make install。
- 构建命令或者构建脚本不能对用户空间（/opt/portadv/用户名/）之外的目录和文件进行创建或修改操作。
- 若检查过程中出现生成BC文件失败的情况，可以自行**生成BC文件**，然后再执行**步骤7~步骤8**通过“BC文件上传”直接分析BC文件，进行内存一致性检查。

步骤6 选择生成的BC文件后单击“确认检查”，开始内存一致性检查。

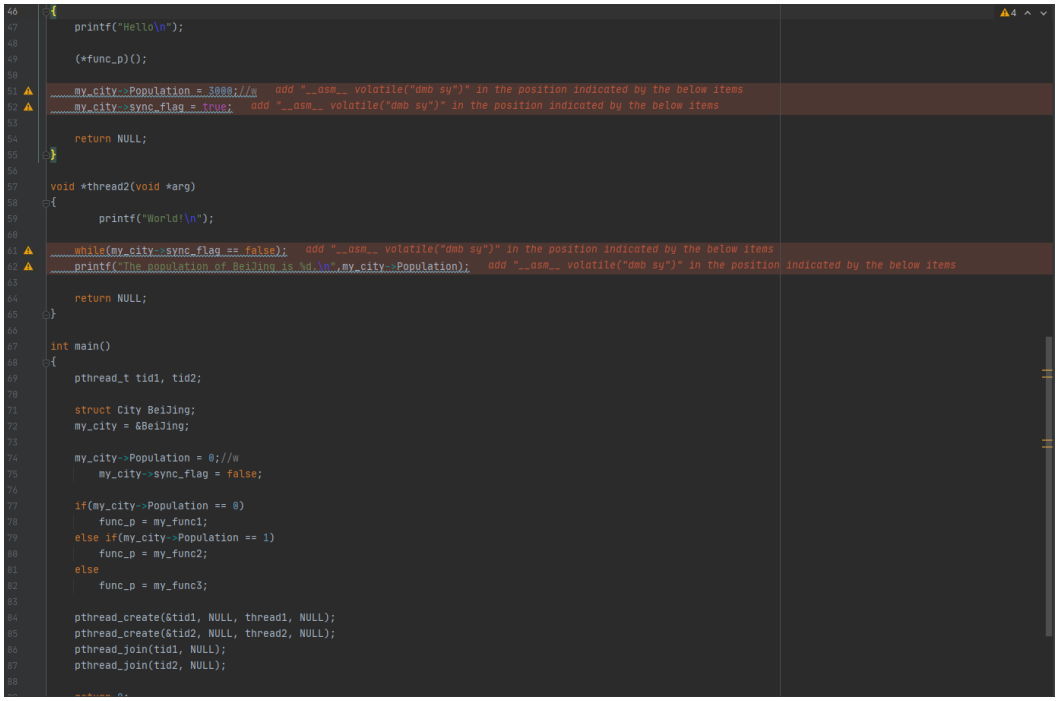
检查完成后，单击“查看报告”可进入“检查报告”界面，如**图3-17**所示

图 3-17 内存一致性检查报告



单击操作列表中的“查看建议源码”进入源码修改建议页面，如**图3-18**所示。

图 3-18 源码迁移建议



说明

- 支持多个用户同时创建内存一致性检查任务。
- 用户可在任务进行过程中单击关闭，取消任务。
- 用户可以单击原始源代码模块右上角的上下键，进行上下切换。

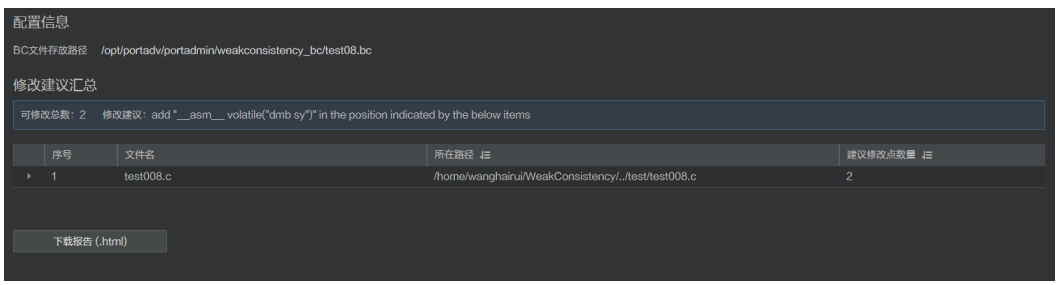
步骤7 填写“BC文件存放路径”。可以通过以下两种方式实现：

- 选择“已上传的BC文件”：单击填写框选择下拉框中的BC文件即可，也可以手动填写BC文件。
- 选择“上传BC文件”：单击“上传”按钮上传BC文件。

步骤8 单击“确认检查”，开始内存一致性检查。

检查完成后，单击“查看报告”可进入“检查报告”界面。如图3-19所示。

图 3-19 内存一致性检查报告



BC文件检查报告中会显示建议修改点数量和建议修改代码的具体位置，可根据这些信息进行排查和修改。

----结束

编译器自动修复工具使用指导

工具支持的操作系统和GCC版本：

表 3-12 工具支持的操作系统和 GCC 版本

操作系统	GCC版本
BC-Linux 7.6/7.7	GCC 4.8.5/4.9.3/5.1.0/5.2.0/5.3.0/5.4.0/5.5.0/6.1.0/6.2.0/6.3.0/6.4.0/6.5.0/7.1.0/7.2.0/7.3.0/7.4.0/8.1.0/8.2.0/8.3.0/9.1.0/9.2.0/9.3.0
CentOS 7.4/7.5/7.6/7.7	GCC 4.8.5/4.9.3/5.1.0/5.2.0/5.3.0/5.4.0/5.5.0/6.1.0/6.2.0/6.3.0/6.4.0/6.5.0/7.1.0/7.2.0/7.3.0/7.4.0/8.1.0/8.2.0/8.3.0/9.1.0/9.2.0/9.3.0
CentOS 8.0	GCC 8.2.0/8.3.0/9.1.0/9.2.0/9.3.0
CentOS 8.1/8.2	GCC 8.3.0/9.1.0/9.2.0/9.3.0
Debian 10	GCC 8.3.0/9.1.0/9.2.0/9.3.0
Deepin 15.2	GCC 6.3.0/6.4.0/6.5.0/7.1.0/7.2.0/7.3.0/7.4.0/8.1.0/8.2.0/8.3.0/9.1.0/9.2.0/9.3.0
iSoft 5.1	GCC 7.3.0/7.4.0/8.1.0/8.2.0/8.3.0/9.1.0/9.2.0/9.3.0
Kylin V10 SP1	GCC 7.3.0/7.4.0/8.1.0/8.2.0/8.3.0/9.1.0/9.2.0/9.3.0
LinxOS 6.0.90	GCC 6.3.0/6.4.0/6.5.0/7.1.0/7.2.0/7.3.0/7.4.0/8.1.0/8.2.0/8.3.0/9.1.0/9.2.0/9.3.0
NeoKylin V7U6	GCC 4.8.5/4.9.3/5.1.0/5.2.0/5.3.0/5.4.0/5.5.0/6.1.0/6.2.0/6.3.0/6.4.0/6.5.0/7.1.0/7.2.0/7.3.0/7.4.0/8.1.0/8.2.0/8.3.0/9.1.0/9.2.0/9.3.0
openEuler 20.03	GCC 7.3.0/7.4.0/8.1.0/8.2.0/8.3.0/9.1.0/9.2.0/9.3.0
SUSE SLES15.1	GCC 7.4.0/8.1.0/8.2.0/8.3.0/9.1.0/9.2.0/9.3.0
Ubuntu 18.04.x	GCC 7.3.0/7.4.0/8.1.0/8.2.0/8.3.0/9.1.0/9.2.0/9.3.0
Ubuntu 20.04.x	GCC 9.3.0
UOS 20 SP1	GCC 8.3.0/9.1.0/9.2.0/9.3.0
uosEuler 20	GCC 7.3.0/7.4.0/8.1.0/8.2.0/8.3.0/9.1.0/9.2.0/9.3.0

说明

- 以上列表以操作系统默认支持的GCC版本为准，如果服务器操作系统升级过GCC版本，可能存在兼容性问题。
- 以上支持的GCC版本都是指GNU官方的GCC分支，不推荐用户基于鲲鹏GCC来使用此编译器自动修复工具。

使用该功能需要配置对应的环境，可以通过以下步骤进行：

步骤1 下载相关软件包。

- GCC源码： **https://gcc.gnu.org/**（请从GCC官网下载对应版本的源码。）
- GCC修复工具patch：**https://github.com/kunpengcompute/devkitdriver/tree/main/gccchecker**（Debian系列和RHEL系列操作系统都使用此包。）
- 内存一致性修复组件：从Porting安装路径下，找到/tools/weakconsistency/gccchecker/gcctool.tar.gz。
- libstdc++6.so依赖库：**http://ftp.cn.debian.org/debian/pool/main/g/gcc-7/libstdc++6-7-dbg_7.4.0-6_arm64.deb**（Debian系列和RHEL系列操作系统都使用该依赖包。）

步骤2 安装内存一致性修复组件。

1. 解压安装包。

   ```
   tar xf gcctool.tar.gz
   ```

   解压后确认“gcctool/bin”目录下有以下文件：

   gcctool，gcctool-bin，add_libraries.sh

2. 安装libstdc++依赖库。

   ⚠ 注意

   libstdc++.so.6依赖库只能给gcctool使用，请勿替换系统原有库，替换后可能会存在兼容性问题。

   gcctool依赖特定版本的libstdc++，脚本gcctool/bin/add_libraries.sh可以自动将libstdc++6-xxx.deb包中的libstdc++.so.6安装到gcctool/bin目录下。

   libstdc++执行命令：

   ```
   bash gcctool/bin/add_libraries.sh -d /path/to/gcctool_root_dir -f /path/to/libstdc++6-7-dbg_7.4.0-6_arm64.deb
   ```

3. 配置环境变量。

   将“gcctool”放入安装目录，配置环境变量：

   ```
   export PATH=/path/to/gcctool/bin:$PATH
   ```

步骤3 合入GCC patch。

说明

如果提示“‘patch’command not found”，则请先安装：

Debian系列：

```
apt install patch
```

RHEL系列：

```
yum install patch
```

```
cd /gcc/source/root/dir
patch -p1 < /path/to/gcc/patch/file
```

步骤4 编译GCC。

GCC源码编译操作请参考GCC官方文档，合入的patch对GCC编译依赖组件和编译过程没有影响，任何GCC编译问题可以询问**GNU社区**。

----结束

准备好环境之后再执行以下步骤使用工具：

步骤1 设置内存一致性修复组件优化等级。

```
export HW_DEBUG=[ 0 | 1 | 2 ]
```

编译组件支持通过环境变量配置修复优化等级，默认级别为**0**。

- 0表示关闭内存一致性修复功能。
- 1表示应用组件优化规则，可以减少性能损失。
- 2表示使用最安全的修复策略，性能损失较大。

步骤2 （可选）定义自动修复源码范围。

1. 工具允许用户自定义源码修复范围，以文件或函数为单位。配置允许列表后，修复组件只对列表内的内容进行修复。允许列表的格式如下：
   - 文件列表以“files:”开头，每个文件独占一行。
   - 文件使用绝对路径。
   - 支持C/C++/Fortran格式的文件，不支持纯汇编文件。
   - 函数列表以“functions:”开头，每个函数独占一行。
   - 支持C/C++普通函数，不支持模板或具有abi_tag属性的函数。

   以下为允许列表的格式示例：

   ```
   files:
     /path/to/file/a
     /path/to/./file/b
     /path/to/../file/c
     /path/to/file/d
   functions:
     func_a
     func_b()
     func_c(int xxx)
     int func_d()
     classA::func_e
     ns::classB::func_f()
     std::string nsA::nsB::classC::func_g(int xxx)
   ```

2. 修复组件获取允许列表路径。

   通过设置环境变量，指定允许列表路径，默认不设置。

   “export AUTOFIXLIST=/path/to/allowlist”

步骤3 编译软件。

用户可以编译软件，编译过程没有变化。（原本编译过程如果使用了-pipe编译选项，需要移除，不会影响原编译结果。）

----结束

生成 BC 文件

BC（BitCode）文件：使用LLVM编译源代码生成中间文件（IR），BC文件是IR的二进制表示。

BC文件生成方式有以下两种：

- 方法一：首选。借助第三方工具gllvm生成BC文件。
  - gllvm工具使用go编译，并依赖clang编译器，使用详情请参考：**https://github.com/SRI-CSL/gllvm/blob/master/README.md**，推荐使用1.2.9版本。

– 使用gllvm工具构建时，需指定编译器类型为gclang和gclang+，增加编译选项“-g -fno-inline-functions”。

使用样例：

请确保系统当前环境满足待测试程序的构建条件。

a. 将clang编译器和编译后gllvm工具设置到环境变量中：

```
export PATH=/opt/portadv/tools/weakconsistency/staticcodeanalyzer/llvm-tools/bin:/path/to/gllvm/bin:$PATH
export LD_LIBRARY_PATH=/home/porting/lib:$LD_LIBRARY_PATH
```

b. 设置gllvm对clang的依赖：

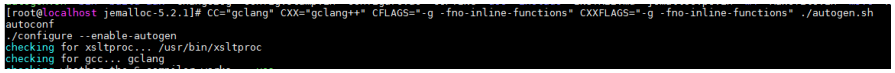
```
export LLVM_CC_NAME="clang"
export LLVM_CXX_NAME="clang++"
export LLVM_LINK_NAME="llvm-link"
```

c. 优化等级：将待测试工程的所有优化等级都修改为“-O0”。

d. 构建：将C编译器指定为gclang，C++编译器修改为gclang++，同时增加编译选项“-g -fno-inline-functions”，此操作不唯一，与待测试工程的构建工具和参数相关；其他构建步骤不变；可参考以下构建用例：

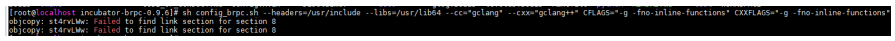
- 对于使用脚本构建，但未指定输入参数的工程，以jemalloc为例，可使用如下命令指定编译器类型：

  图 3-20 jemalloc

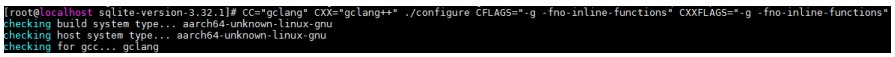
- 对于使用脚本构建，但指定输入参数的工程，以incubator-brpc为例，可根据incubator-brpc的参数要求，使用如下命令指定编译器类型：

  图 3-21 incubator-brpc

- 对于使用configure构建的工程，以sqlite为例，可使用如下命令指定编译器类型：

  图 3-22 sqlite

- 对于在配置文件中指定了编译器类型的工程，需修改配置文件，以bwa为例，使用make构建工具，需将Makefile第一行CC指定为glang后再执行构建命令

  图 3-23 修改 makefile 文件

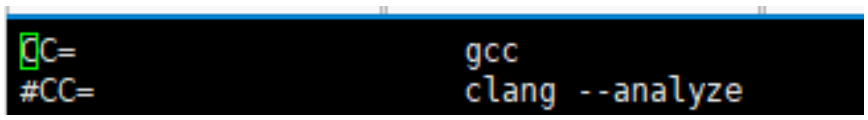
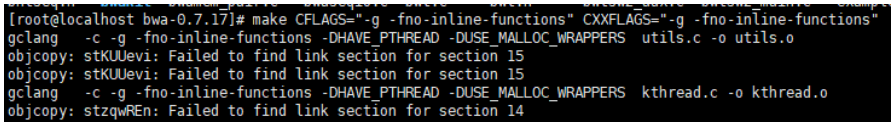
  

  图 3-24 bwa

- 生成BC文件：构建完成后，生成动态库(.so)文件或者可执行文件，使用gllvm的get-bc工具生成BC文件。以jemalloc为例：

  图 3-25 jemalloc

  ```
  [root@localhost lib]# ls
  libjemalloc.a  libjemalloc_pic.a  libjemalloc.so  libjemalloc.so.2
  [root@localhost lib]# get-bc libjemalloc.so.2
  Bitcode file extracted to: libjemalloc.so.2.bc.
  [root@localhost lib]# ls
  libjemalloc.a  libjemalloc_pic.a  libjemalloc.so  libjemalloc.so.2  libjemalloc.so.2.bc
  [root@localhost lib]#
  ```

- 方法二：使用工程构建工具生成BC文件，如make，cmake等。如果软件使用该方法，需修改相关的配置文件，如makefile。
  - 替换编译命令：使用clang替换gcc等，使用llvm-link替换ld等。
  - 调整编译选项：修改优化等级“-O0”，增加“-flto -g -fno-inline-functions”。

  上述操作完成后，通过构建获得BC文件。

  使用样例：

  须知

  - 在构建项目之前，需将gcc编译器替换成llvm相关的工具。
  - 构建时务必替换或增加指定的编译选项，防止部分指令信息缺失，从而影响分析结果准确性。
  - 将最终生成的目标文件修改为BC文件。

  a. 使用make工具完成项目构建。

  图 3-26 构建

  ```
  [root@localhost test]# ls
  main.c  Makefile  test.c
  ```

  b. 修改编译命令并替换编译选项。以下是原始的makefile文件和修改后的makefile文件：

图 3-27 原始的 makefile 文件

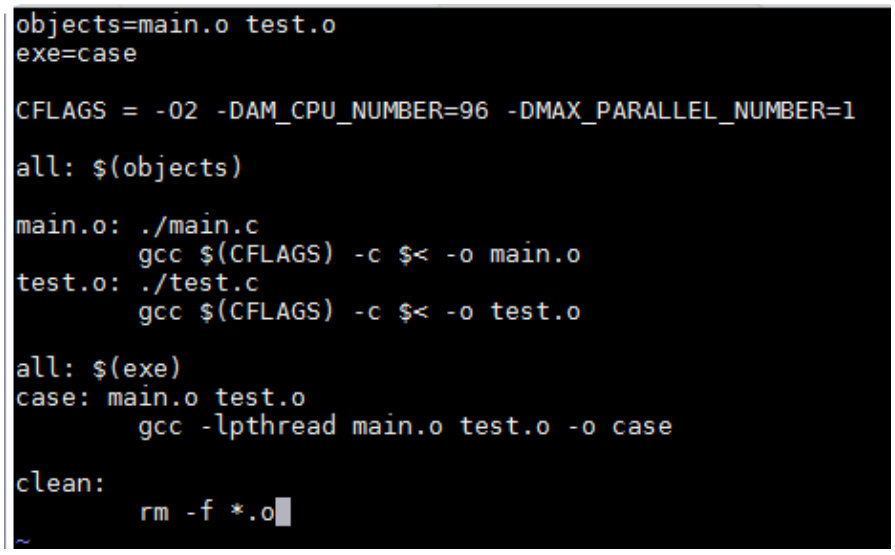
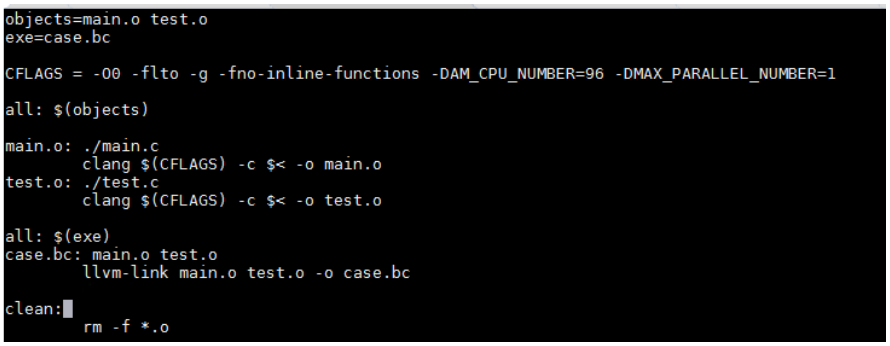


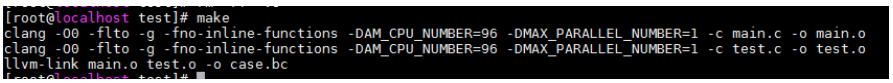
图 3-28 修改后的 makefile 文件



修改后的makefile文件，将编译工具修改为clang，链接生成目标文件的工具改为llvm-link。

c. 完成项目构建。

图 3-29 构建完成



说明

- 若分析结果不准确，可能是BC文件异常造成的。
- 生成的BC文件异常，可能是由于编译过程中涉及gfortran等编译器。
- 可通过以下方式验证BC文件是否异常：使用llvm的llc和本地的gcc将文件转为可执行程序，并验证该程序功能是否正常。

3.3 常用操作

3.3.1 通过普通用户连接鲲鹏代码迁移工具

当用户没有在服务器上部署工具时，单击插件界面上的“点击此处部署”开始部署安装鲲鹏代码迁移工具。

在安装过程中配置目标服务器参数时，若想通过操作系统的普通用户来安装工具，该普通用户需要满足以下要求：

1. 创建home目录。
   ```
   mkdir /home/xxx
   ```
2. 设置新创建的目录所属者为xxx。
   ```
   chown -R xxx:yy /home/xxx
   ```
3. 设置用户所在的用户组为wheel。
   ```
   usermod -G wheel xxx
   ```

说明

命令中xxx为普通用户的用户名，yy为普通用户的组名。

3.3.2 配置 SSH 密钥认证

以下步骤以Windows系统为例。

步骤1 打开本地cmd终端。

步骤2 执行如下命令生成生成公私钥对。

ssh-keygen -m PEM -t rsa -b 2048

过程中需要：

- （可选）输入保存的文件名，默认为在“C:\Users*username*/.ssh”目录下保存为“id_rsa”（私钥）和“id_rsa.pub”（公钥）文件。
- （可选）设置密钥的密码口令。

回显信息如下：

```
Generating public/private rsa key pair.
 Enter file in which to save the key (C:\Users\username/.ssh/id_rsa):
 Enter passphrase (empty for no passphrase):
 Enter same passphrase again:
 Your identification has been saved in C:\Users\username/.ssh/id_rsa.
 Your public key has been saved in C:\Users\username/.ssh/id_rsa.pub.
 The key fingerprint is:
SHA256:rCRpryf6uZU+dQd/S8WN1azvay58zi3gtb53gayhvO8 china\username@HGHY4USERNAME
 The key's randomart image is:
+---[RSA 2048]----+
|              ..|
|               +|
|              =.|
|    . . .   o +|
|   + . S  o. .o |
|  . + o. ..+o+..|
|     =....o+= +.|
|   .=o  o ..o=o=|
| .o=+.. .+E .OO+|
+----[SHA256]-----+
```

步骤3 普通用户登录服务器并上传公钥文件“id_rsa.pub”。

将公钥文件放至“/home/*操作系统用户名*/.ssh”目录下，将公钥文件“id_rsa.pub”名字改为“authorized_keys”。

如已存在此“authorized_keys”文件，将公钥文件内容复制至“authorized_keys”文件中即可。

步骤4 设置“authorized_keys”文件的权限为600。

chmod 600 /home/*操作系统用户名*/.ssh/authorized_keys

步骤5 查看服务器ssh配置文件。

cat /etc/ssh/sshd_config

检查以下两处开关：

PubkeyAuthentication yes

RSAAuthentication yes

如若不是“yes”将上述开关设为“yes”后保存配置文件并重启sshd服务。

步骤6 在IDE插件中选择密钥认证。

在界面上直接导入本地的私钥文件即可。

----结束

3.3.3 配置扫描参数

单击工具右侧的  按钮，选择“设置”，单击工具下面的“扫描参数配置”，打开如图3-30所示界面。参数描述如表3-13所示。

图 3-30 扫描参数设置

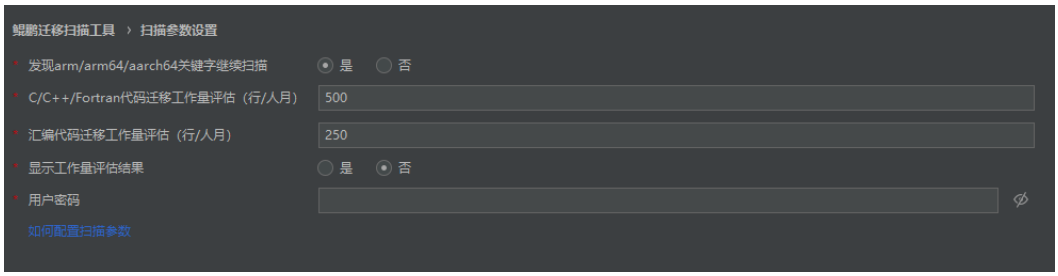


表 3-13 扫描参数说明

参数	说明
Arm/Arm64/AArch64关键字是否继续扫描	默认为“是”。
C/C++/Fortran代码迁移工作量评估（行/人月）	输入C/C++/Fortran代码迁移工作量评估标准。默认为500，取值范围1~99999。
汇编代码迁移工作量评估（行/人月）	输入汇编代码迁移工作量评估标准。默认为250，取值范围1~99999。
显示工作量评估结果	选择是否在分析报告详情页面显示“预估迁移工作量”。默认为“否”。
用户密码	输入当前登录用户的密码。

修改扫描参数配置后，单击“确定”。

3.3.4 管理依赖字典

单击工具右侧的按钮，选择“设置”，单击工具下面的“依赖字典”，打开如图3-31所示界面。参数描述如表3-14所示。

说明

- 只有管理员（**portadmin**）可以备份、恢复、升级依赖字典。
- 以下内容中的/opt为服务端默认安装路径，如果为自定义安装，请使用对应的自定义安装路径替换。
- 不支持2.3.T10以前的版本使用2.3.T10版本的依赖字典文件执行升级依赖字典的操作。

图 3-31 依赖字典管理

表 3-14 依赖字典管理参数说明

参数	说明
备份	备份依赖字典。 输入管理员密码，单击“确定”按钮备份依赖字典。 **说明** 系统上只会保留最新备份的依赖字典，历史备份依赖字典将会被替换。
升级	1. 单击“上传”按钮上传依赖字典，默认保存在“/opt/portadv/config/dependency_dictionary”目录下。 **说明** 请下载最新的依赖字典。 2. 输入管理员密码。 3. 单击“确定”按钮升级依赖字典。
恢复	恢复依赖字典。 输入管理员密码，单击“确定”按钮恢复已备份的依赖字典。

3.3.5 管理 web 服务端证书

Web服务端证书用于客户端浏览器和web服务器之间的通讯，实现数据信息在客户端和web服务器之间的加密传输，可以防止数据信息的泄露。为提高安全性，建议替换成自己的证书，并及时更新证书，保证证书的有效性。

前提条件

已成功登录鲲鹏代码迁移工具。

说明

只有管理员（**portadmin**）可以执行生成CSR文件、导入web服务器证书、下载根证书，重启和更换工作密钥的操作。普通用户只能查看web服务端证书信息。

查看当前 web 端服务证书

步骤1 单击“鲲鹏代码迁移插件”菜单右侧  ，在下拉菜单中选择“设置”，然后单击“Web服务端证书”。

步骤2 查看证书信息，如图3-32所示，界面参数描述如表3-15所示。

图 3-32 Web 服务端证书信息

表 3-15 web 服务端证书参数说明

参数	说明
证书名称	显示证书名称。
证书到期时间	显示证书到期时间。
状态	显示证书的当前状态。 有效：证书剩余有效时间超过证书过期告警阈值。 即将过期：证书剩余有效时间小于或等于证书过期告警阈值。 已过期：已过证书到期时间。 **说明** • 工具每天自动检查并更新证书状态。 • web服务证书自动告警时间默认为90天。管理员可在“证书过期告警阈值”选框中修改，可配置范围为7~180天。

参数	说明
操作	• 生成CSR文件 **说明** CSR（Certificate Signing Request）是证书请求文件。证书申请者在申请数字证书时，由CSP（加密服务提供者）在生成私钥的同时也生成证书请求文件。证书申请者把CSR文件提交给证书颁发机构后，证书颁发机构使用其根证书私钥签名就生成了证书公钥文件，即证书。 • 导入web服务端证书 • 重启服务 • 更新工作密钥

----结束

自定义服务器证书信息并导入

步骤1 在“Web服务端证书”界面的操作栏中单击“生成CSR文件”。

打开“生成CSR文件”对话框，如图3-33所示，界面参数描述如表3-16所示。

图 3-33 生成 CSR 文件

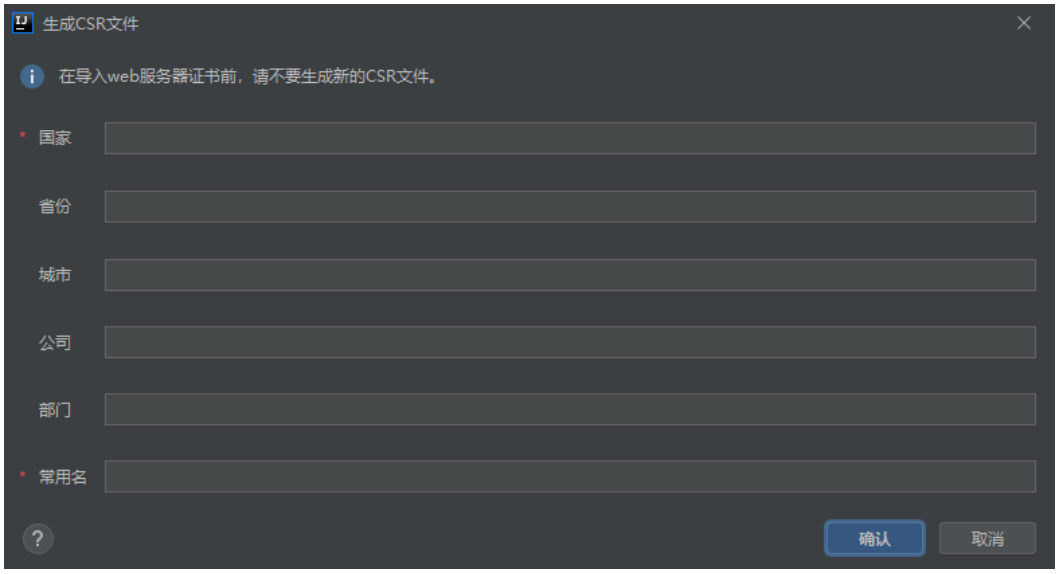


表 3-16 CSR 文件参数说明

参数	说明
国家	使用者所在的国家。 支持字母，长度为2个字符，为必填项。
省份	使用者所在的省份。 支持字母、数字、“-”、“_”、“.”和空格，最大长度128个字符。

参数	说明
城市	使用者所在的城市。 支持字母、数字、“-”、“_”、“.”和空格，最大长度128个字符。
公司	使用者所在的公司。 支持字母、数字、“-”、“_”、“.”和空格，最大长度64个字符。
部门	使用者所在部门。 支持字母、数字、“-”、“_”、“.”和空格，最大长度64个字符。
常用名	使用者的名称。 支持字母、数字、“-”、“_”、“.”和空格，最大长度64个字符，为必填项。

步骤2 单击“确认”生成CSR文件。

步骤3 将导出的CSR文件发往SSL证书颁发机构，并申请SSL证书。

获取到正式的证书后，保存到本地。

说明

用户也可以使用其根证书进行自签名获取正式的证书。

步骤4 单击“导入web服务端证书”。

弹出“导入web服务端证书”对话框。

步骤5 单击按钮，选择需要导入的证书文件。

说明

- 导入的证书文件不得大于1MB，支持的格式为*.crt、*.cer、*.pem。
- 步骤**步骤1**中生成的CSR文件与向CA机构申请的服务器证书是一一对应的，在导入服务器证书之前请不要再次生成新的CSR文件，否则需要向CA机构重新申请服务器证书。
- 导入的SSL证书如果不是从正式的证书颁发机构获取，而是用户自己使用工具生成，在导入该SSL证书后，还需要确认客户端浏览器中是否已存在对应的根证书。

步骤6 单击“上传”导入web服务端证书。

步骤7 单击“重启服务”使证书生效。

说明

用户手动在服务器上重启Nginx服务，证书不会生效，必须在web上进行重启操作。

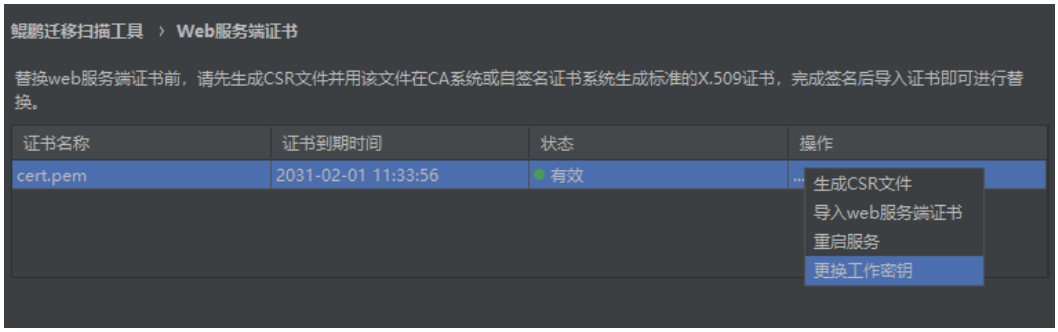
----结束

更新工作密钥

工作密钥用于加密启动Nginx服务的口令，根密钥用于加密工作密钥。为提高系统安全性，建议定期更新。

步骤1 在“Web服务端证书”界面的操作栏中单击“更换工作密钥”，如**图3-34**所示。

图 3-34 更新工作密钥



步骤2 单击“重启服务”使工作密钥生效。

----结束

3.3.6 通过构建工具配置文件识别编译命令

现象描述

在使用代码迁移工具过程中，存在对构建文件配置分析场景时，“编译命令”可能不是默认的make/cmake，用户需要根据自己的构建配置文件来确定“编译命令”，进而填写“编译命令”。

参考说明

- “构建工具”选择make时，默认为make，支持以make开头的自定义编译命令。make工具支持参数-C和-f，其中-C用于指定工作目录，-f用于指定makefile。如果用户的Makefile中使用了标签，则“编译命令”需要根据用户使用标签的情况来确定。
  - 常见场景下，用户会在Makefile中设置clean和all标签，如图3-35所示。

    图 3-35 场景一

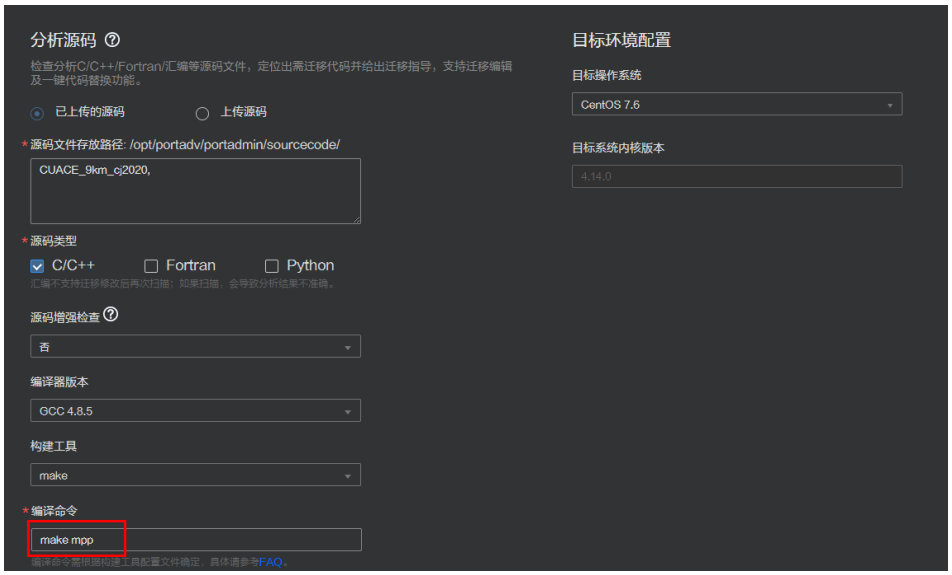
    ```
    all: pairaln wtpre wtcyc wtmer wtzmo wtobt wtclp wtext wtgbo wtlay wtcns wtmsa

    pairaln: $(GENERIC_SRC) pairaln.c
            $(CC) $(CFLAGS) -o pairaln file_reader.c ksw.c pairaln.c $(GLIBS)

    wtpre: $(GENERIC_SRC) wtpre.c
            $(CC) $(CFLAGS) -o wtpre file_reader.c wtpre.c $(GLIBS)

    wtcyc: $(GENERIC_SRC) wtcyc.c
            $(CC) $(CFLAGS) -o wtcyc file_reader.c ksw.c wtcyc.c $(GLIBS)

    wtmer: $(GENERIC_SRC) wtmer.c
            $(CC) $(CFLAGS) -o wtmer file_reader.c wtmer.c $(GLIBS)
    ```

    这种情况下，make clean表示执行clean标签下的编译命令，make表示执行all标签或者all标签关联的其它标签下的编译命令，此时，用户在“编译命令”中填写make即可。
  - 如果用户设置了自定义的mpp标签，如图3-36所示。

图 3-36 场景二

这种情况下，make mpp表示执行mpp标签下的编译命令，此时，用户需要在“编译命令”中填写make mpp，如图3-37所示。

图 3-37 make mpp



- “构建工具”选择cmake时，默认为cmake，支持以cmake开头的自定义编译命令。cmake工具支持的参数请参考cmake 3.13.4官方使用手册。
- “构建工具”选择automake时，默认为make，且不可修改。automake工具不支持任何参数。使用automake时，需要确保用户的项目源码中存在可以正常使用的软件构建配置文件Makefile，Makefile文件需要和源码文件一起上传，或者用户在上传待分析软件后手动执行automake相关命令，然后将生成的Makefile属主改为porting，以便工具能正常读取Makefile。

3.3.7 多任务并发异常处理

现象描述

当用户在创建源码迁移、增强功能任务时，可能会出现以下两个提示：

- 当前工具不能达到最大并发处理，多任务运行可能需要等待，可以参考用户指南使工具达到最大并发处理。
- 当前工具无法分析任务，请参考用户指南使工具正常运行任务。

可能原因

- 系统用户被篡改。
- 数据库被篡改。

处理步骤

- 方法一：若当前已安装的鲲鹏代码迁移工具存在更高版本，升级该工具。
- 方法二：卸载当前已安装的鲲鹏代码迁移工具，重新安装该工具。

3.4 FAQ

3.4.1 工具检测连接失败提示算法不匹配

现象描述

安装代码迁移工具，点击“检测连接”提示算法不匹配。

可能原因

客户端算法与服务器端不匹配。

处理步骤

步骤1 root用户登录服务器。

步骤2 执行**vim /etc/ssh/sshd_config**指令检测如下算法是否配置(每项满足其一即可)。

- KexAlgorithms: diffie-hellman-group-exchange-sha256
- MACs: hmac-sha2-256,hmac-sha1
- Ciphers: aes128-ctr,aes192-ctr,aes256-ctr
- HostKey: /etc/ssh/ssh_host_rsa_key

----结束

3.4.2 工具安装失败提示 OpenSSL 库未找到

现象描述

工具安装失败并提示OpenSSL库未找到。部分回显信息如下：

```
OpenSSL 1.1.1-dev xx XXX xxxx
openssl already installed!
openssl-devel already installed!
checking for OS
+ Linux 4.14.0-115.el7a.0.1.aarch64 aarch64
checking for C compiler ... found
+ using GNU C compiler
+ gcc version: 7.3.0 (GCC)
...
checking for OpenSSL library ... not found
...
./auto/configure: error: SSL modules require the OpenSSL library.
...
with nginx by using --with-openssl=<path> option.

make: *** No rule to make target 'build', needed by 'default'. Stop.
make: *** No rule to make target 'install'. Stop.
nginx模块安装错误!
cp: cannot create directory ‘/opt/portadv/tools/nginx-install/html/’: No such file or directory
...
hmod: cannot access ‘/opt/portadv/tools/nginx-install/html/’: No such file or directory
Installation failed!
```

可能原因

系统上升级过OpenSSL，但未对应升级OpenSSL依赖组件。

说明

OpenSSL组件包括openssl，openssl-libs，openssl-devel。

处理步骤

恢复OpenSSL组件到系统自带的OpenSSL 1.0.2版本或者完整升级OpenSSL组件到系统上安装的OpenSSL 1.1.1版本，然后重新安装工具。

恢复OpenSSL组件步骤如下（以CentOS 7.7 Arm为例）：

步骤1 从CentOS官网下载CentOS镜像文件“CentOS-7-aarch64-Everything-1908.iso”。

步骤2 使用SSH远程登录工具，将CentOS镜像上传至/root目录。

步骤3 使用SSH远程登录工具，进入CentOS操作系统命令行界面。

步骤4 执行如下命令将CentOS的ISO镜像文件挂载到本地目录下。

mount /root/CentOS-7-aarch64-Everything-1908.iso /media -o loop

须知

系统重启后需要重新挂载。

步骤5 执行如下命令进入挂载目录。

cd /media

步骤6 执行如下命令找出OpenSSL组件所在的位置。

find . -name "openssl*"

步骤7 执行如下命令安装OpenSSL组件。

rpm -ivh /media/Packages/openssl-devel-1.0.2k-19.el7.aarch64.rpm --nodeps --force

rpm -ivh /media/Packages/openssl-libs-1.0.2k-19.el7.aarch64.rpm --nodeps --force

rpm -ivh /media/Packages/openssl-1.0.2k-19.el7.aarch64.rpm --nodeps --force

----结束

注意下载对应版本的OpenSSL相关RPM包，在升级过程中需要挂载对应操作系统版本的ISO镜像文件安装相关依赖。

完整升级OpenSSL组件的步骤如下：

升级 OpenSSL

适用于以下版本操作系统：

- CentOS 6.5/6.9/6.10/7.2/7.3
- RHEL 6.5/6.9/6.10/7.2/7.3
- 深之度15.2 x86版本

步骤1 下载如下OpenSSL相关RPM包：

- **openssl-libs-1.0.2k-19.el7.x86_64.rpm**
- **openssl-devel-1.0.2k-19.el7.x86_64.rpm**
- **openssl-1.0.2k-19.el7.x86_64.rpm**

步骤2 将相关RPM包上传至服务器任意目录下，使用cd命令切换至该目录。

步骤3 分别执行如下命令安装OpenSSL。

rpm -ivh openssl-libs-1.0.2k-19.el7.x86_64.rpm

rpm -ivh openssl-devel-1.0.2k-19.el7.x86_64.rpm

rpm -ivh openssl-1.0.2k-19.el7.x86_64.rpm

说明

如果需要先安装依赖，执行**yum install**命令安装相关依赖，然后执行如下命令安装OpenSSL。

rpm -ivh openssl-libs-1.0.2k-19.el7.x86_64.rpm --force --nodeps

rpm -ivh openssl-devel-1.0.2k-19.el7.x86_64.rpm --force --nodeps

rpm -ivh openssl-1.0.2k-19.el7.x86_64.rpm --force --nodeps

步骤4 安装完后，执行如下命令查看OpenSSL版本。

openssl version

显示如下内容说明OpenSSL成功升级。

```
OpenSSL 1.0.2k-fips
```

----结束

3.4.3 工具安装时出现 nginx 编译错误而导致安装失败

现象描述

工具安装时出现nginx编译错误而导致安装失败。

可能原因

系统中存在加速库提供的libwd.so文件导致工具安装时nginx编译失败。

处理步骤

执行**find / -name libwd***指令查询系统中是否存在版本号为1.2.10的libwd.so文件，如果存在建议将此文件升级到最新版本。

3.4.4 工具在容器中安装失败

现象描述

工具在Docker中安装失败。

可能原因

Docker的/usr/include目录下缺少sys文件夹。

处理步骤

在Docker的/usr/include目录查找sys文件夹，若无此文件夹，则从Docker对应的操作系统中拷贝sys文件夹至Docker的/usr/include目录下，然后卸载并重新安装工具。

4 编译插件

4.1 介绍

鲲鹏开发套件是基于Intellij的一款扩展工具，编译插件是其中的一个子工具。编译插件即插即用，支持一键安装鲲鹏GCC编译器，以及鲲鹏平台远程调试能力，支持的功能特性如下：

- 一键式部署

支持从Intellij编辑器插件面板（Plugins）中搜索“Kunpeng DevKit”或“Kunpeng Compiler”下载并在线安装插件，同时支持一键部署服务端鲲鹏GCC编译器

- 编译调试
  - 一键式安装鲲鹏GCC
  - 可视化编译配置任务，一键式任务运行
  - 远程单步调试C/C++代码
  - 编译调试过程信息实时展示
  - gtest框架用例树渲染及状态展示

4.2 特性指南

4.2.1 快速服务器和编译器部署

4.2.1.1 特性描述

鲲鹏编译插件支持用户快速添加目标服务器并部署鲲鹏GCC、毕昇编译器和毕昇JDK。

4.2.1.2 特性操作

4.2.1.2.1 安装鲲鹏 GCC 编译器

如果您需要使用鲲鹏GCC编译器中一些加速库，则需要先在服务端安装鲲鹏GCC编译器。鲲鹏GCC编译器是针对鲲鹏平台的高性能编译器，它基于开源GCC开发的编译器工具链（包含编译器、汇编器、链接器），支持C、C++、Fortran语言及其运行库。

当前仅以下操作系统下支持安装鲲鹏GCC编译器：

openEuler 20.03（LTS）

openEuler 20.09

CentOS 7.6

Ubuntu 18.04

Ubuntu 20

Kylin V10

UOS 20

步骤1 在左侧菜单栏单击  ，然后单击插件下的  进入部署页面。

步骤2 配置目标服务器参数后，单击“开始部署”，一键式安装鲲鹏GCC编译器。

图 4-1 安装鲲鹏 GCC 编译器

表 4-1 安装鲲鹏 GCC 参数

参数	说明
服务器IP地址	待安装鲲鹏GCC的服务器IP地址。
SSH端口	待安装鲲鹏GCC的服务器SSH端口。
操作系统用户名	登录待安装鲲鹏GCC的服务器的操作系统用户名。 **说明** 如果想通过普通用户安装，需要满足一些条件，具体参照**4.3.2 通过普通用户安装编译器**。
选择SSH连接方式	密码认证或密钥认证。
操作系统用户密码	输入操作系统用户密码。
私钥	导入id_rsa私钥文件。

参数	说明
密码短语	（可选）输入生成密钥设置的密码短语。

安装过程中需要：

- 再次输入服务器用户密码
- 输入服务器root用户密码安装鲲鹏GCC

----结束

说明

由于root用户拥有最高权限，直接使用root用户登录服务器可能会存在安全风险。建议您使用普通用户登录服务器后切换为root用户，再执行后续安装操作，并建议您通过配置禁止root用户SSH登录的选项，来提升系统安全性。具体配置如下：

先以普通用户登录服务器，切换至root登录后检查/etc/ssh/sshd_config配置项PermitRootlogin，如果显示no，说明禁止了root用户登录；如果显示yes，则需要将配置项PermitRootlogin是否设置no。

4.2.1.2.2 安装毕昇编译器

如果您需要使用毕昇编译器，则需要先在服务端安装毕昇编译器。毕昇编译器基于开源LLVM开发，并进行了优化和改进，同时将flang作为默认的Fortran语言前端编译器，是针对鲲鹏平台的高性能编译器。

当前仅以下操作系统下支持安装毕昇编译器：

openEuler 20.03（LTS）

openEuler 20.09

CentOS 7.6

Ubuntu 18.04

Ubuntu 20

Kylin V10

UOS 20

步骤1 在左侧菜单栏单击  ，然后单击插件下的  进入部署页面。

步骤2 配置目标服务器参数后，单击“开始部署”，一键式安装毕昇编译器。

图 4-2 安装毕昇编译器

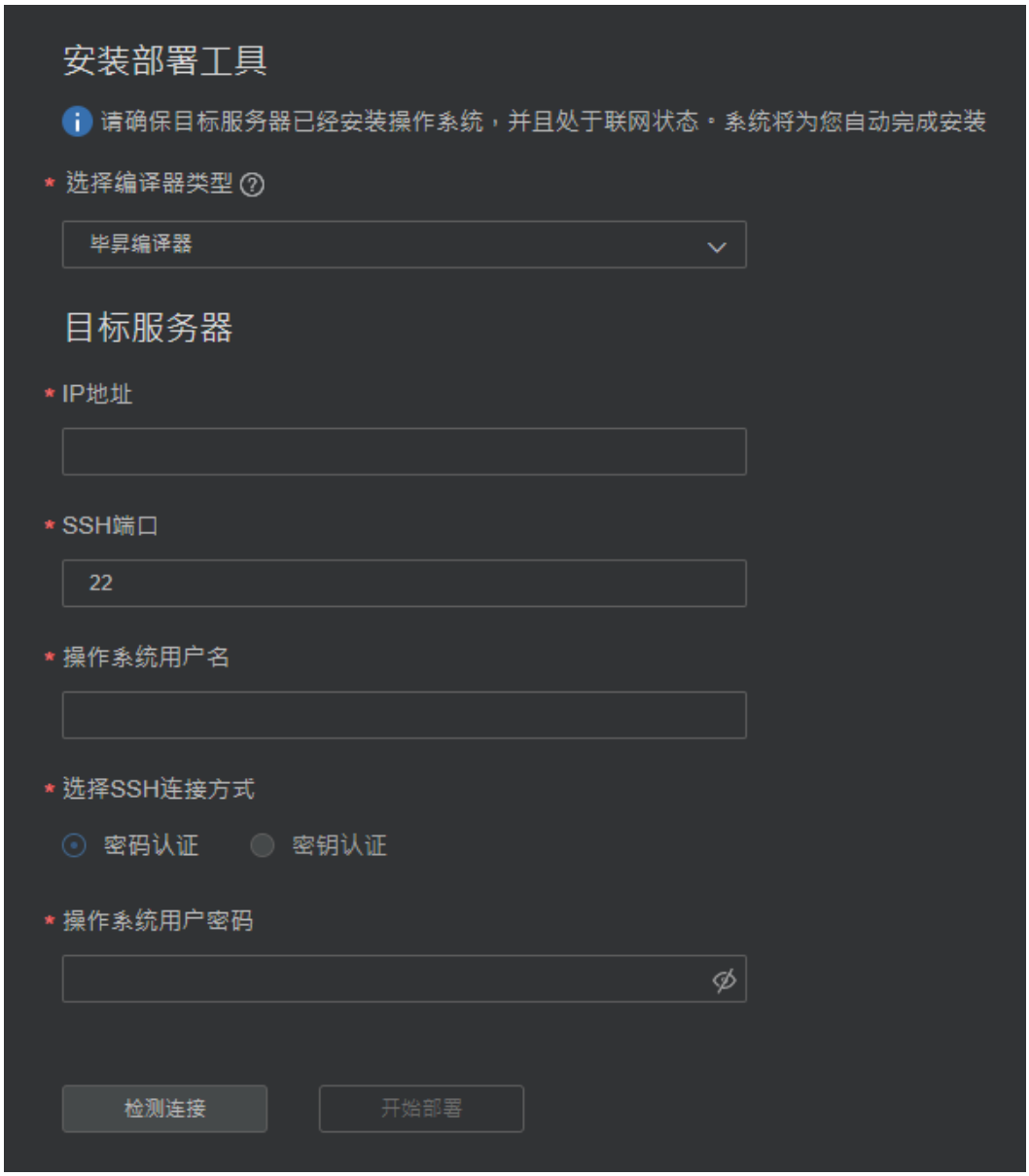


表 4-2 安装毕昇编译器参数

参数	说明
服务器IP地址	待安装毕昇编译器的服务器IP地址。
SSH端口	待安装毕昇编译器的服务器SSH端口。
操作系统用户名	登录待安装毕昇编译器的服务器的操作系统用户名。 **说明** 如果想通过普通用户安装，需要满足一些条件，具体参照**4.3.2 通过普通用户安装编译器**。
选择连接方式	密码认证或密钥认证。
操作系统密码	输入操作系统密码。

参数	说明
私钥	导入id_rsa私钥文件。
密码短语	（可选）输入生成密钥设置的密码短语。

安装过程中需要：

- 再次输入服务器用户密码
- 输入服务器root用户密码安装毕昇编译器

----结束

4.2.1.2.3 安装毕昇 JDK

如果您需要使用毕昇JDK，则需要先在服务端安装毕昇JDK。毕昇JDK基于OpenJDK开发，是一个高性能、可用于生产环境的OpenJDK发行版，它积累了大量使用场景和Java开发者反馈的问题和诉求，解决了业务实际运行中遇到的多个问题，并在ARM架构上进行了性能优化。

当前仅以下操作系统下支持安装毕昇JDK：

openEuler 20.03（LTS）

openEuler 20.09

CentOS 7.6

Ubuntu 18.04

Ubuntu 20

Kylin V10

UOS 20

步骤1 在左侧菜单栏单击  ，然后单击插件下的  进入部署页面。

步骤2 配置目标服务器参数后，单击“开始部署”，一键式安装毕昇JDK。

图 4-3 安装毕昇 JDK

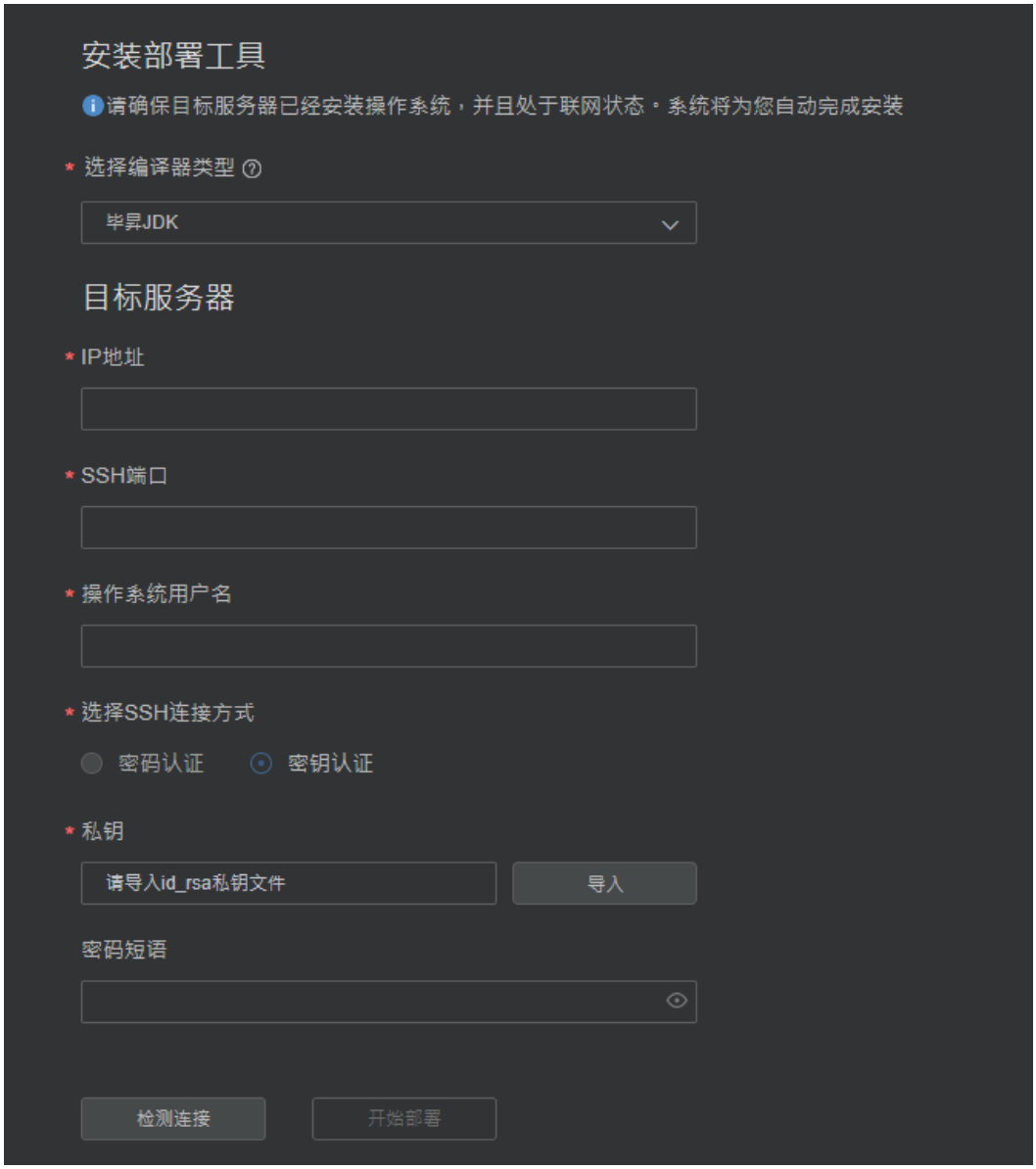


表 4-3 安装毕昇 JDK 参数

参数	说明
服务器IP地址	待安装毕昇JDK的服务器IP地址。
SSH端口	待安装毕昇JDK的服务器SSH端口。
操作系统用户名	登录待安装毕昇JDK的服务器的操作系统用户名。 **说明** 如果想通过普通用户安装，需要满足一些条件，具体参照**4.3.2 通过普通用户安装编译器**。
选择连接方式	密码认证或密钥认证。

参数	说明
操作系统密码	输入操作系统密码。
私钥	导入id_rsa私钥文件。
密码短语	（可选）输入生成密钥设置的密码短语。

安装过程中需要：

- 再次输入服务器用户密码
- 输入服务器root用户密码安装毕昇JDK

----结束

4.2.2 编译调试

4.2.2.1 特性描述

鲲鹏编译器插件支持用户配置代码编译和调试任务，然后选择右侧功能栏中对应的任务运行。

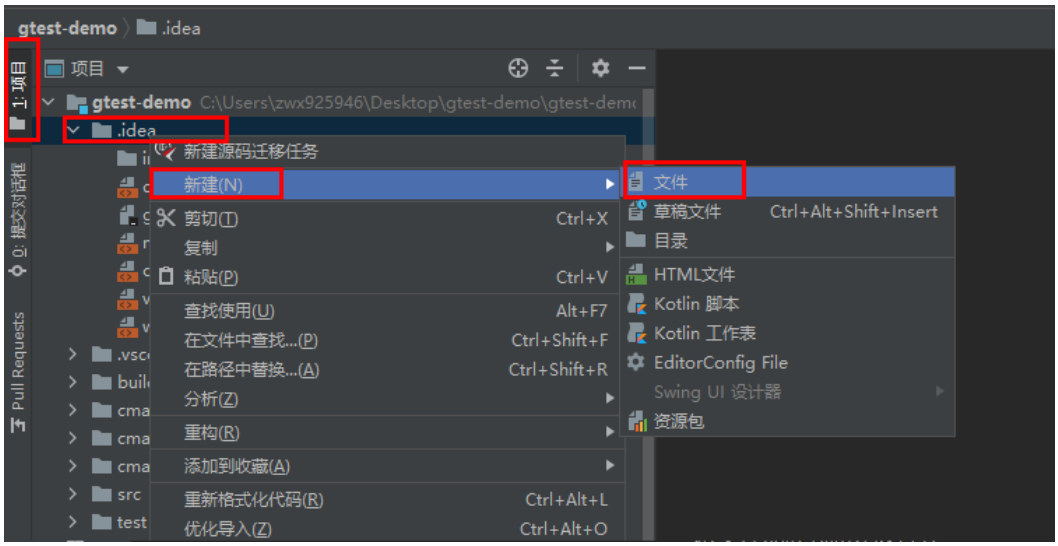
4.2.2.2 特性操作

配置远程环境

步骤1 生成settings.json文件。

在IntelliJ上选择“项目”，找到“.idea”文件夹，单击右键选择“新建>文件”， 输入文件名“settings.json”生成settings.json文件。如图4-4所示。

图 4-4 新建生成 settings.json 文件



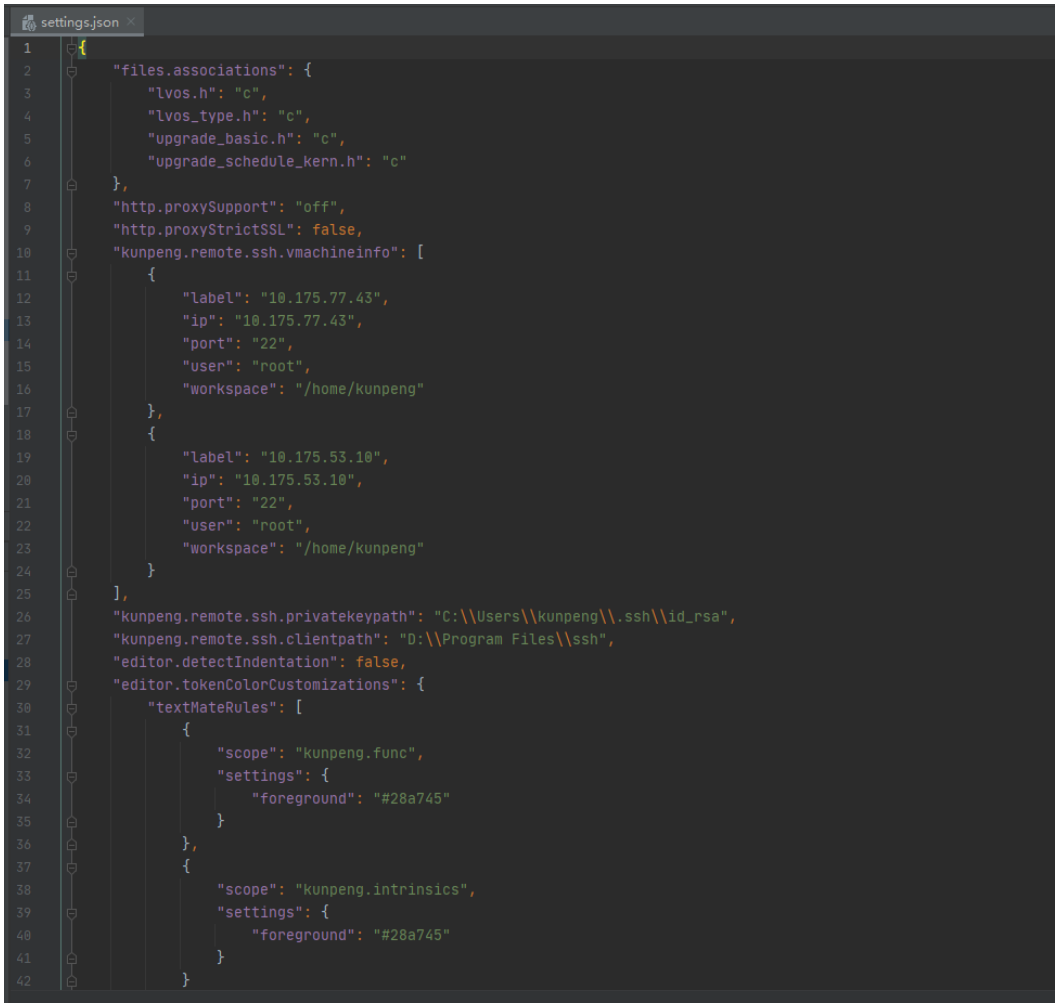
步骤2 配置远程环境信息。

复制以下代码至新创建的settings.json文件中，修改以下代码中的服务器名称、目标服务器的IP、目标服务器IP对应的端口、登录目标服务器的用户名和同步代码到服务器的路径。

```
"kunpeng.remote.ssh.machineinfo": [
 {
    "label": "xxx",/*服务器名称*/
    "ip": "xxx.xxx.xxx.xxx",/*目标服务器的IP*/
    "port": "xxx",/*目标服务器IP对应的端口*/
    "user": "xxx",  /*登录目标服务器的用户名*/
    "workspace": "xxx"/*同步代码到服务器的路径*/
 },
 {
    "label": "xxx",
    "ip": "xxx.xxx.xxx.xxx",
    "port": "xxx",
    "user": "xxx",  /
    "workspace": "xxx"
 }
]
```

配置样例如下：

图 4-5 settings.json



----结束

配置本地 SSH 公私钥登录

配置本地SSH公私钥登录（以Windows为例）

步骤1 制作公私钥对。

Windows下可以通过cmd执行**ssh-keygen -b 2048 -t rsa**命令生成公私钥对，默认回车即可。生成的公私钥对默认保存在“~/.ssh/目录”下的id_rsa、id_rsa.pub文件中。

步骤2 拷贝本地.ssh文件夹到SSH服务端（即要连接的远程环境），然后将公钥文件中的内容导入到authorized_keys文件中。

cd ~/.ssh cat id_rsa.pub >> authorized_keys

步骤3 配置目录权限。

- SSH服务端用户目录权限配置为700
  chmod 700 ~
- .ssh目录权限配置为700
  chmod 700 ~/.ssh
- authorized_keys权限配置为600
  chmod 600 ~/.ssh/authorized_keys

----结束

配置本地SSH私钥文件id_rsa的路径

在IntelliJ上打开settings.json文件，在文件中添加“kunpeng.remote.ssh.privatekeypath”，并填写私钥文件id_rsa的路径。

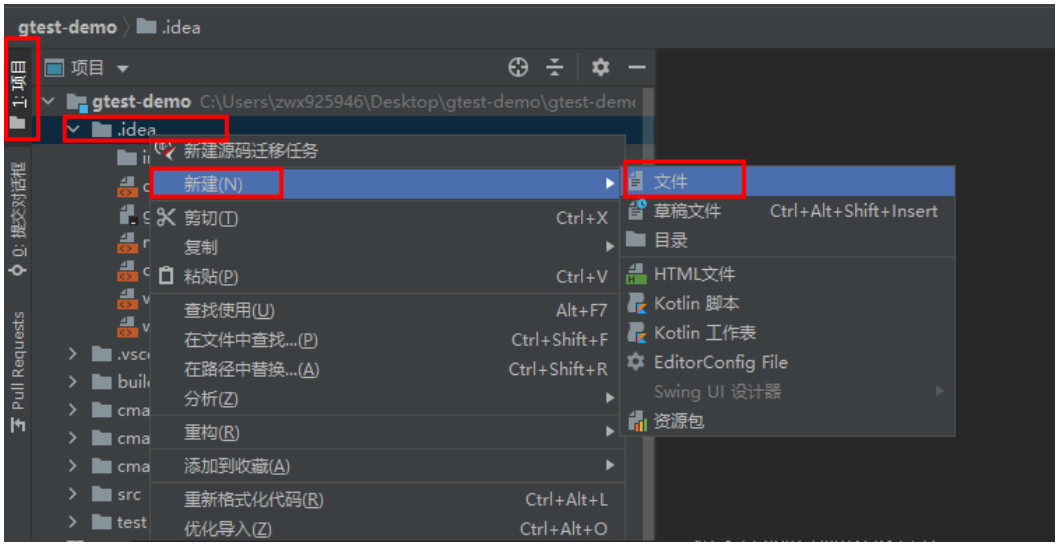
路径格式为：x:\xxx\xxx\xxx\id_rsa

配置编译任务（tasks.json）

步骤1 生成tasks.json文件。

在IntelliJ上选择“项目”，找到“.idea”文件夹，单击右键选择“新建>文件”， 输入文件名“tasks.json”生成tasks.json文件。如图4-6所示。

图 4-6 新建生成 tasks.json 文件



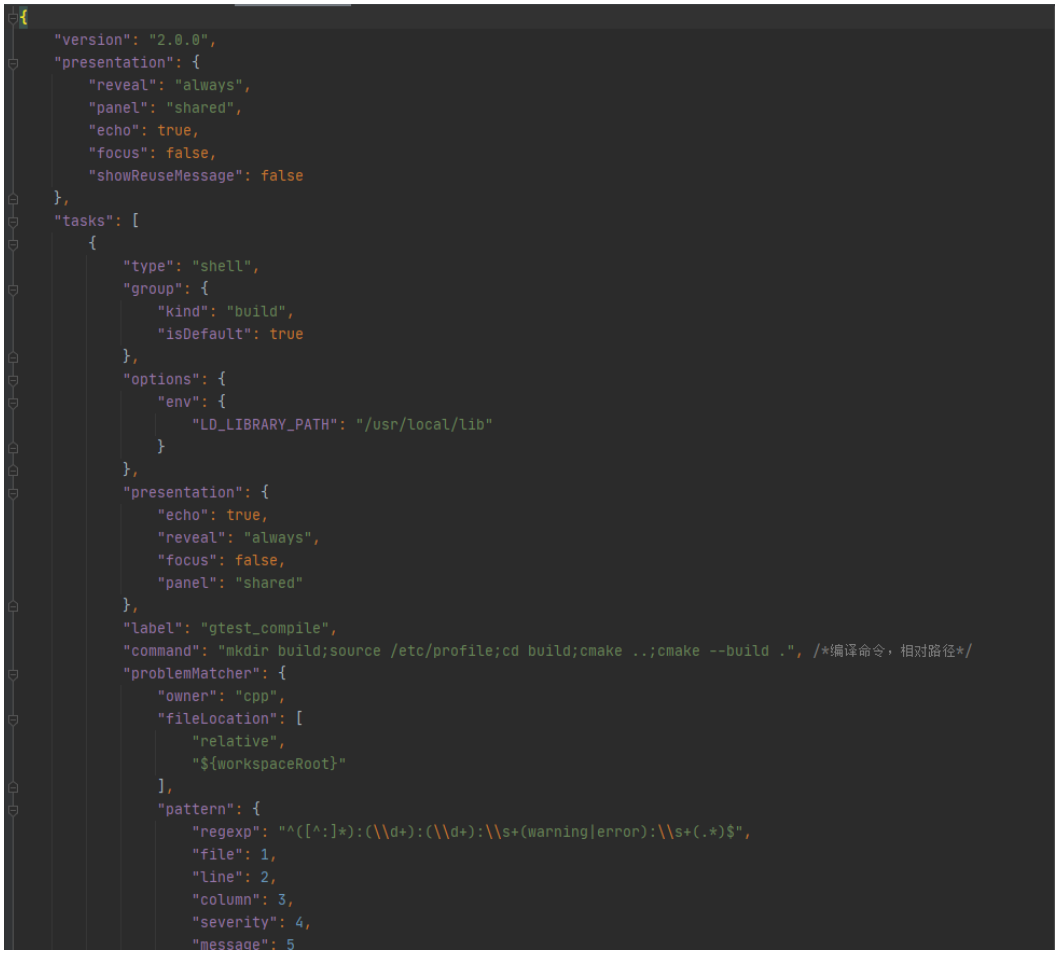
步骤2 配置编译任务信息。

复制以下代码至新创建的tasks.json文件中，修改以下代码中的编译任务名称和启动编译的命令（命令里包含的路径为相对于workspace的路径）,如图4-7所示。

```
{
    // See https://go.microsoft.com/fwlink/?LinkId=733558
    // for the documentation about the tasks.json format
    "version": "2.0.0",
    "tasks": [
        {
            "type": "shell",
            "group": {
                "kind": "build",
                "isDefault": true
            },
            "options": {
                "env": {
                    "xxx": "xxx"
                }
            },
            "presentation": {
                "echo": true,
                "reveal": "always",
                "focus": false,
                "panel": "shared"
            },
            "label": "xxx",    /*编译任务名称*/
            "command": "xxx", /*启动编译的命令，命令里包含的路径为相对于workspace的路径*/
            "problemMatcher": {
                "owner": "cpp",
                "fileLocation": [
                    "relative",
                    "${workspaceRoot}"
                ],
                "pattern": {
                    "regexp": "^([^:]*):(\\d+):(\\d+):\\s+(warning|error):\\s+(.*)$",
                    "file": 1,
                    "line": 2,
                    "column": 3,
                    "severity": 4,
                    "message": 5
                }
            }
        }
    ]
}
```

配置样例如下：

图 4-7 tasks.json

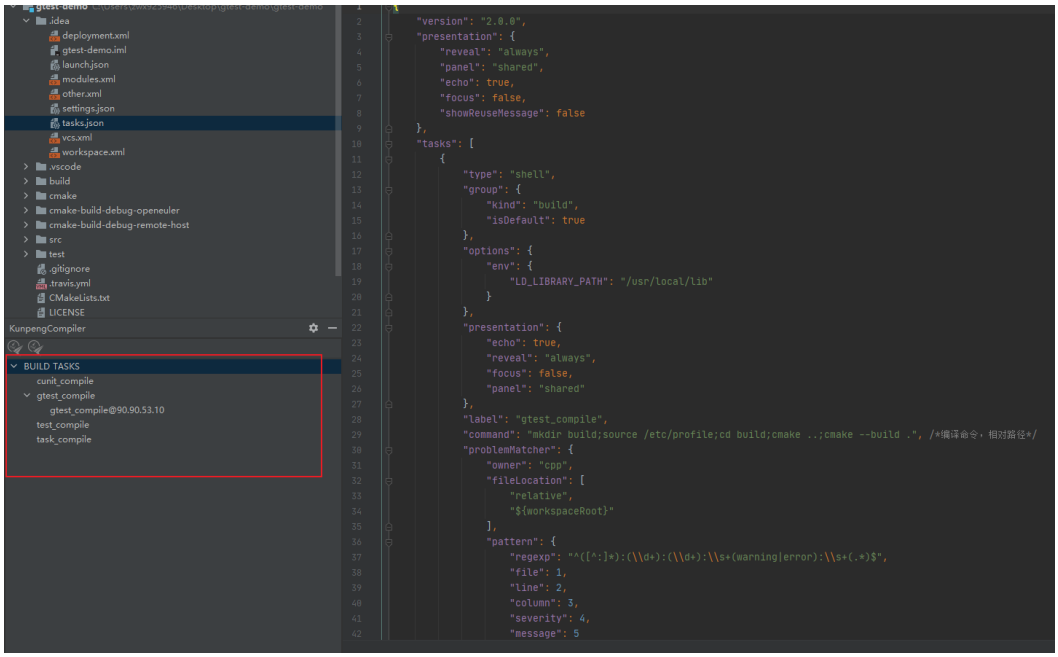


```
{
    "version": "2.0.0",
    "presentation": {
        "reveal": "always",
        "panel": "shared",
        "echo": true,
        "focus": false,
        "showReuseMessage": false
    },
    "tasks": [
        {
            "type": "shell",
            "group": {
                "kind": "build",
                "isDefault": true
            },
            "options": {
                "env": {
                    "LD_LIBRARY_PATH": "/usr/local/lib"
                }
            },
            "presentation": {
                "echo": true,
                "reveal": "always",
                "focus": false,
                "panel": "shared"
            },
            "label": "gtest_compile",
            "command": "mkdir build;source /etc/profile;cd build;cmake ..;cmake --build .", /*编译命令，相对路径*/
            "problemMatcher": {
                "owner": "cpp",
                "fileLocation": [
                    "relative",
                    "${workspaceRoot}"
                ],
                "pattern": {
                    "regexp": "^([^:]*):(\\d+):(\\d+):\\s+(warning|error):\\s+(.*)$",
                    "file": 1,
                    "line": 2,
                    "column": 3,
                    "severity": 4,
                    "message": 5
```

----结束

远程编译

插件安装完成后在左侧工具栏会显示一个视图，如下图所示。



开始启动编译前，需要同意同步代码到workspace 路径下（“kunpeng.remote.ssh.machineinfo”中定义的“workspace”），同步完成即可启动编译任务。

启动编译任务：选中要编译的文件，右键单击“启动编译”。

4.3 常用操作

4.3.1 安装编译调试工具

为满足编译调试需要，远端服务器需要预先安装编译调试工具（如已安装，可忽略），以openEuler 20.03（LTS）为例：

yum install make

yum install cmake

yum install gcc-c++

yum install git

yum install gdb

4.3.2 通过普通用户安装编译器

如果想要使用鲲鹏GCC编译器、毕昇编译器和毕昇JDK，需要先在服务器上安装鲲鹏GCC编译器、毕昇编译器和毕昇JDK。

用户可以在插件中配置目标服务器参数后，单击“开始部署”，一键式安装鲲鹏GCC编译器、毕昇编译器和毕昇JDK。

在配置目标服务器参数时，若想通过操作系统的普通用户来安装编译器，该普通用户需要满足以下要求：

1. 创建home目录。
   ```
   mkdir /home/xxx
   ```
2. 设置新创建的目录所属者为xxx。
   ```
   chown -R xxx:yy /home/xxx
   ```
3. 设置用户所在的用户组为wheel。
   ```
   usermod -G wheel xxx
   ```

说明

命令中xxx为普通用户的用户名，yy为普通用户的组名。

4.4 FAQ

5 加速库插件

5.1 介绍

鲲鹏开发套件是基于IntelliJ的一款扩展工具，加速库插件是其中的一个子工具。加速库插件即插即用，能够扫描代码文件中可使用鲲鹏加速库优化后的函数或汇编指令，生成可视化报告；编码时能够自动匹配鲲鹏加速库函数字典，智能提示、高亮、联想字典中可以替换的库和函数。关于鲲鹏加速库的详细介绍请参见**鲲鹏加速库**。

鲲鹏加速库插件支持的功能特性如下：

- 智能联想
  Coding时自动联想鲲鹏加速库优化后的相关函数
- 函数搜索
  支持鲲鹏加速库函数的代码定义跳转、函数搜索
- 语法高亮
  Coding时高亮鲲鹏加速库优化后的相关函数
- 加速分析
  支持工程和文件扫描，识别出可以用鲲鹏加速库替换的函数
- 字典管理
  支持加速库函数字典管理，可线上（自动）和线下更新

5.2 特性指南

5.2.1 加速分析

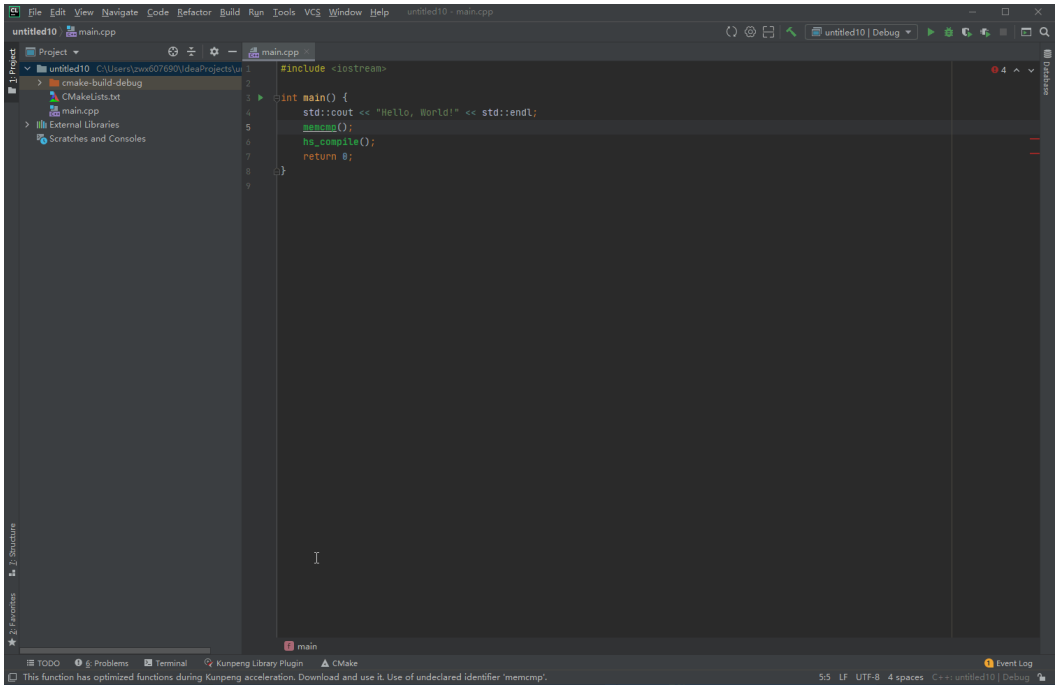
5.2.1.1 特性描述

在IntelliJ的资源栏中，用户可以右键点击自己项目下的任意文件或者文件夹或者空白区域，此时会出现“鲲鹏加速分析”、“清除加速分析报告”、“查看加速分析报告”。

5.2.1.2 特性操作

单击“鲲鹏加速分析”后，插件会分析工程里面依赖加速库的函数，并将扫描的结果在可视化面板以及IntelliJ的工具栏展示。点击工具栏中的问题可以跳转到函数方法，点击可视化界面中的“函数所在路径”可以跳转到函数方法。

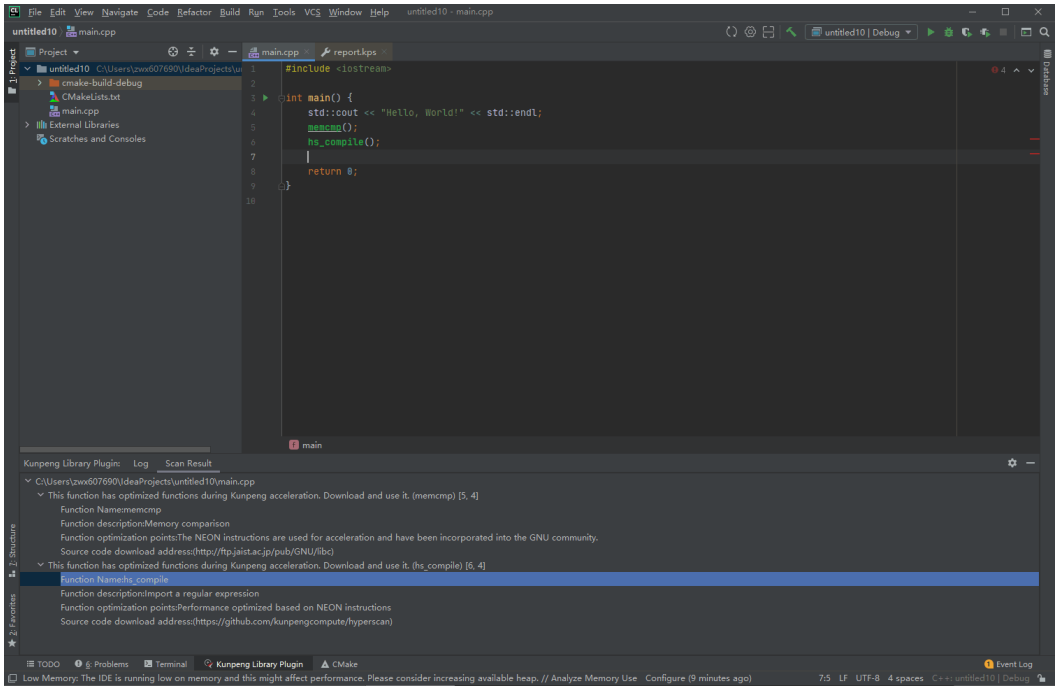
图 5-1 加速分析



5.2.2 编码辅助

当插件安装完成后，插件会自动下载鲲鹏的字典库数据，字典数据下载完成后，插件会启用针对加速库函数的语法高亮、函数说明、以及代码自动补全。

图 5-2 编码辅助（语法高亮/智能联想/智能提示）



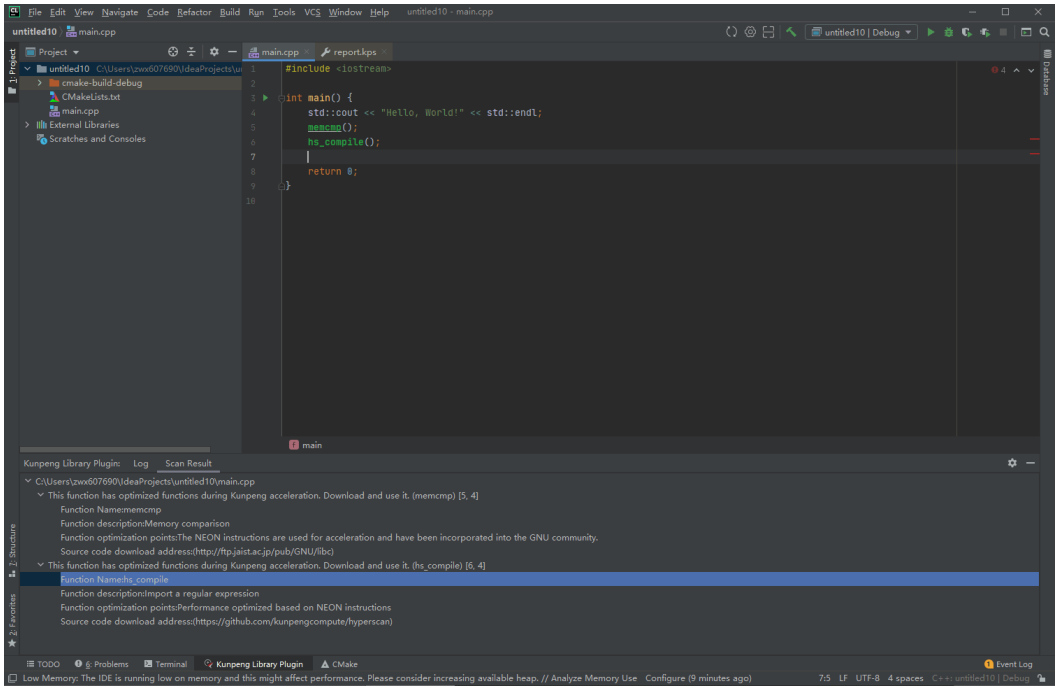
5.2.2.1 特性描述

当插件安装完成后，插件会自动下载鲲鹏的字典库数据，字典数据下载完成后，插件会启用针对加速库函数的语法高亮、函数定义跳转、以及代码自动补全。

5.2.2.2 特性操作

系统会根据输入的内容自动联想函数或补全代码。用户可以使用上下方向键高亮智能联想内容的说明，还可以点击说明框中的链接地址查看函数的详细定义。也可单击代码查看补全选项。

图 5-3 编码辅助（语法高亮/智能联想/智能提示）



5.2.3 函数搜索

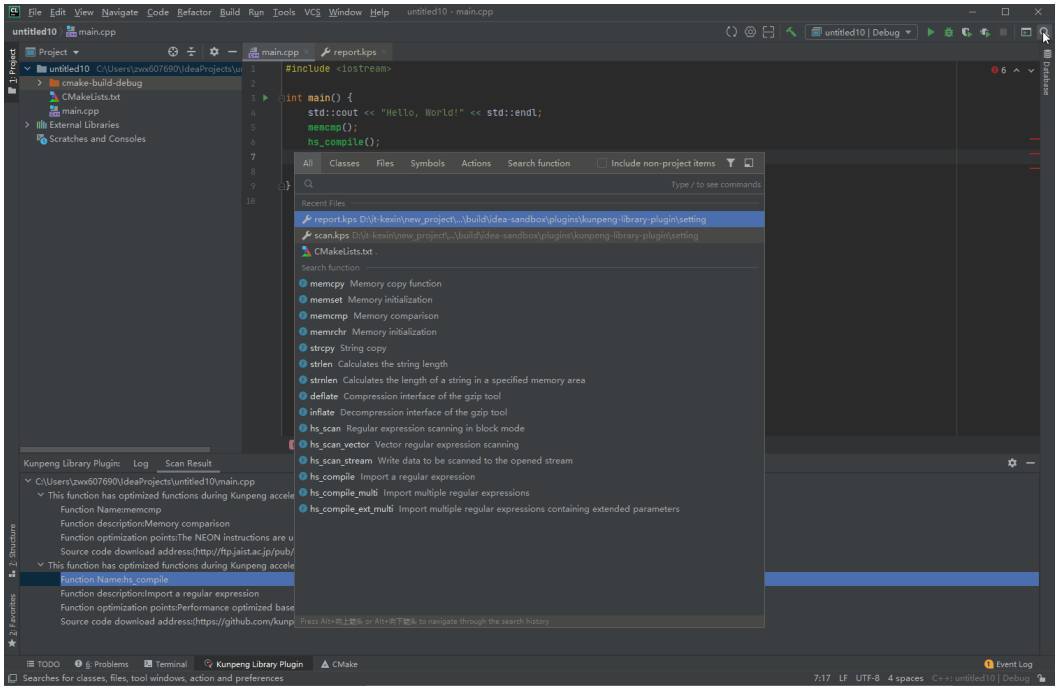
5.2.3.1 特性描述

加速库插件提供对function函数的名称、优化点、描述搜索，以及Intrinsic函数的名称、函数详细定义、Intel对应的Intrinsic功能函数名称、Intel对应的汇编指令名称、ARM对应的汇编指令名称的搜索。

5.2.3.2 特性操作

单击编辑区右上角的  ，在搜索框中输入搜索关键字，然后单击function函数会跳转头文件定义，单击Intrinsic函数会打开Intrinsic的帮助文档地址。

图 5-4 加速库优化后的函数搜索



5.3 常用操作

- 单击编辑区右上角  ，即可配置远程字典地址，导入本地字典，和配置加速库提示选项。

  说明

  - 用户启动鲲鹏加速库插件时会自动从远程字典地址获取最新字典数据。请提前确认网络代理是否正常配置，并能联网。
  - 如果要导入本地字典，请从以下连接获取：

    https://gitee.com/kunpengcompute/kunpengacclibdict/raw/master/dictionary.json

- 单击编辑区右上角  ，即可检测新版本。插件自身也存在自动检测机制。
- 单击编辑区右上角  ，选择加速分析的类型，然后点击“保存设置”，如图5-5所示。

图 5-5 选择加速分析类型

5.4 FAQ

5.4.1 无法下载字典文件

问题

使用插件时，提示“无法获取远程字典文件”，如何解决?

图 5-6 无法获取远程字典文件

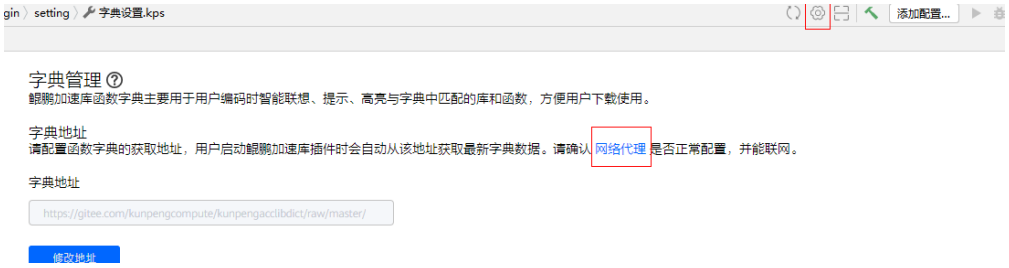
回答

步骤1 请单击“重置远程地址”，将字典远程地址重置为默认值。

步骤2 重置后如果还是不能下载字典文件，此时需要检测网络是否处于封闭网络，如果是则需要配置代理才能访问字典地址。

- 请单击错误提示框中的“检查网络代理”，进入IntelliJ网络代理配置界面。
- 或者单击工具类上的“字典管理”，进入字典管理界面，然后单击“网络代理”，进入网络代理配置界面。

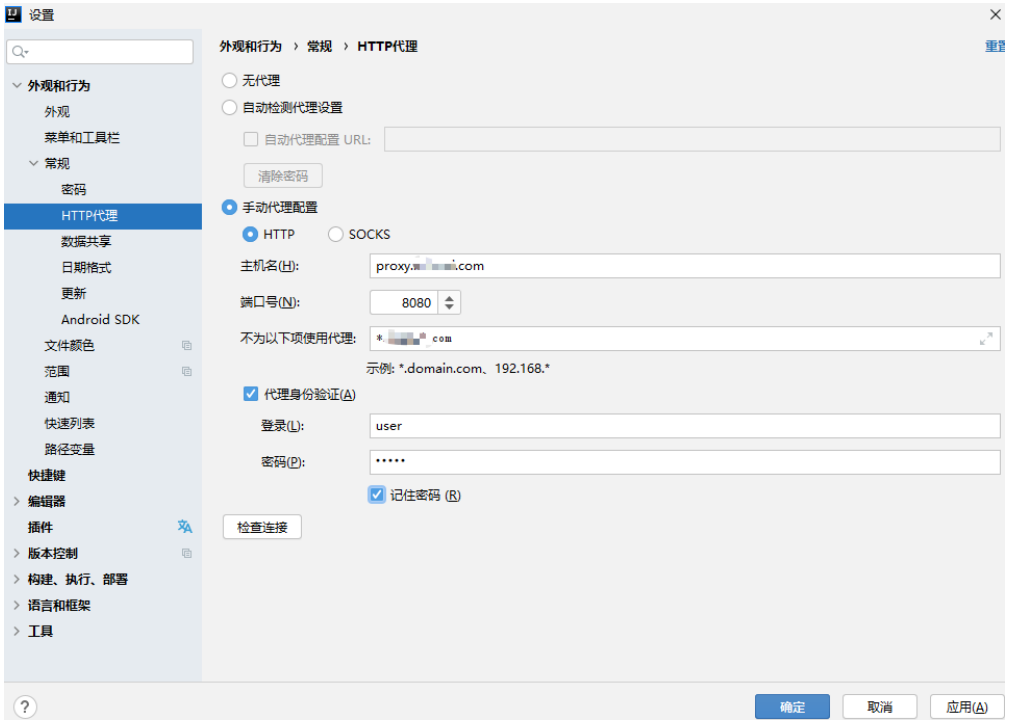
图 5-7 字典管理



配置网络代理，变量值配置格式：http://用户名:密码@proxy.xx.com:8080

- 如果您的代码服务器需要认证，则请配置用户名和密码，注意密码有特殊字符需要转义。
- proxy.xx.com为代理服务器的域名
- 8080为代理服务器的端口

图 5-8 配置网络代理



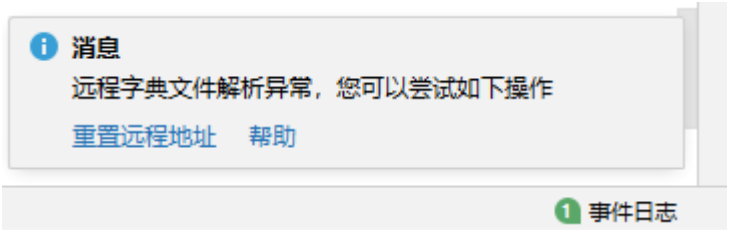
----结束

5.4.2 无法解析字典文件内容

问题

远程字典下载成功，但该字典文件不是有效的json格式，内容无法解析，如何解决?

图 5-9 无法解析



回答

可能由于用户修改过字典的默认远程地址导致的。请单击错误提示框中的“重置远程地址”进行重置。

5.4.3 无法获取远端校验文件

问题

远程字典下载成功，且字典文件也是有效的json格式，内容可以解析，但在下载校验文件时，提示“无法获取远程校验文件”，如何解决?

图 5-10 无法获取远程校验文件

说明

该校验文件用于对字典内容进行合法性校验，确认是否被篡改等。

回答

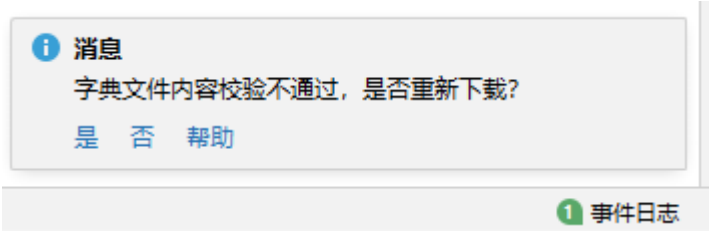
可能由于用户修改过校验地址导致的。请单击错误提示框中的“重置校验地址”进行重置。

5.4.4 字典文件校验不通过

问题

校验文件下载成功，但是字典文件与其不匹配，如何解决?

图 5-11 字典文件校验不通过



回答

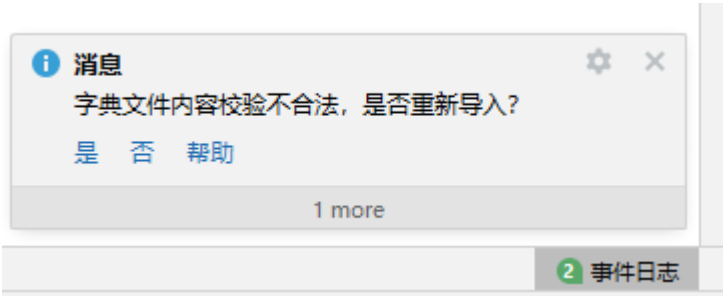
可能是由于字典文件下载后被篡改，或者校验文件下载后被篡改导致的。请单击错误提示框中的“是”，重新下载字典文件和校验文件。

5.4.5 字典文件校验不合法

问题

远程字典下载成功，且字典文件也是有效的json格式，内容可以解析，校验sha256通过，但里面的内容不合法，会弹出如下提示。如何解决?

图 5-12 字典文件内容校验不合法



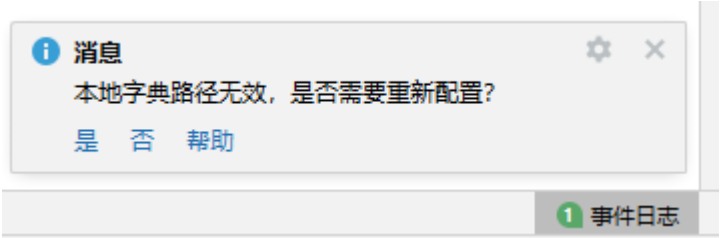
回答

由于字典文件内容被篡改导致的。请单击错误提示框中的“是”，重新下载字典文件和校验文件。

5.4.6 导入本地字典文件常见问题

问题1：导入本地字典文件时，提示“本地字典路径无效”，如何解决?

图 5-13 本地字典路径无效

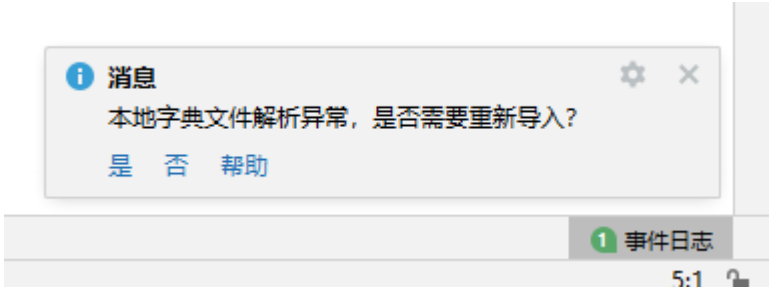


回答：由于填写的路径不规范，不是本地有效的磁盘路径导致的。请单击错误提示框中的“是”，重新选择有效的字典文件或输入有效的路径。

图 5-14 输入有效的路径

问题2：导入本地字典文件时，文件路径有效，但是无法正常解析json格式，如何解决?

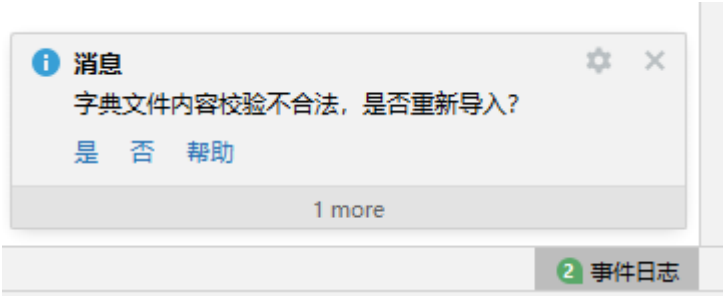
图 5-15 无法解析



回答：由于本地字典文件不是标准json格式导致的，例如文件中缺失冒号等。请打开要导入的文件，检查文件中是否有缺失冒号或逗号的情况，请修改为标准的json格式后重新导入。

问题3：导入本地字典文件时，文件路径有效且正常解析json格式，但提示“字典文件内容校验不合法”，如何解决?

图 5-16 字典文件内容校验不合法



回答：请从**字典远程地址**重新获取一份字典文件，然后重新导入。

5.4.7 函数搜索功能常见问题

使用函数搜索功能时，为什么有的跳转定义有的打开网站?

回答：字典中的Function函数会跳转定义，Intrinsic函数会打开帮助文档链接。

5.4.8 待扫描文件中函数数量与扫描结果中的函数数量不一致

问题

为什么待扫描文件中函数数量与扫描结果中的函数数量不一致?

回答

扫描分析只扫描分析字典里面的function函数，不分析Intrinsic函数。同时扫描也是根据用户所选的加速库类型来扫描的。