**功能&性能测试用例执行结果**

# 认证软件名称和版本（必选）

## 认证软件名称和版本

|  |  |  |  |
| --- | --- | --- | --- |
| 用例模块\*： | 功能测试 | 子模块： | 软件版本 |
| 用例编号： | 01 |
| 用例名称： | 软件名称和版本 |
| 用例目的\*： | 验证待测试软件的软件名称和版本号 |
| 预置条件\*： | 1. 待认证软件完成迁移和部署。
2. 待认证软件启动正常。
 |
| 测试步骤\*： | 1. 启动软件，查看软件名称和版本号信息。
2. 将1中信息截图保存，并附到测试结果中。
 |
| 预期结果\*： | 1. 软件名称与待认证软件名称一致。
2. 软件版本与待认证软件版本一致。
 |
| 测试结果\*：**(测试日志 或 截图)** |  |
| 测试结论\* | *通过/有条件通过/不通过* |
| 备 注： | *若不通过或有条件通过，在此备注说明* |

# 软件迁移完整性扫描（必选）

鲲鹏代码迁移工具支持x86 Linux到Kunpeng Linux的软件分析评估和重构适配。

工具包获取： <https://www.hikunpeng.com/developer/devkit/download>

## 软件迁移完整性扫描（非源码软件包）

|  |  |  |  |
| --- | --- | --- | --- |
| 用例模块\*： | 兼容性测试 | 子模块： | 软件迁移 |
| 用例编号： | 02 |
| 用例名称： | Porting\_001软件安装包可通过Porting Advisor的软件安装包分析 |
| 用例目的\*： | 对非源码软件包，执行迁移前和已迁移的软件安装包（rpm、deb、jar、war、tar、zip、gz、tar.gz、tar.bz，tgz、tbz，tbz2、egg、whl、ear）分析，确认已迁移的软件安装包已完成鲲鹏不兼容的依赖文件的替换。 |
| 预置条件\*： | 1. 目标服务器已配置相应的YUM/APT/ZYPPER源。2. 获取鲲鹏DevKit安装包，并解压进入。3. 执行命令安装：./install.sh ，并安装代码迁移工具完成。4. 拷贝安装完成后的工具URL地址到浏览器（Chrome/Edge）：https://$IP:$Port默认用户：devadmin 密码：自定义（IP和Port存在映射的需要映射场景） |
| 测试步骤\*： | 1. 登录鲲鹏DevKit首页。2. 点击新建“快速创建鲲鹏工程或任务”，默认选择“迁移”功能中“软件迁移评估”，点击“确认”。3. 默认任务名称，勾选“分析软件包”，选择“上传”的分析软件包，选择“目标操作系统”，点击“开始分析”，参考如下图： |
| 预期结果\*： | 成功生成软件迁移评估报告，且评估成功。分析报告评估结果一：1. 软件迁移评估成功，用户进行判断/修复/验证。2. 下载分析报告或截取分析结果截图（修改前/验证后）。分析报告评估结果二：1. 提示无需要评估的内容，则不需要做改动。2. 下载分析报告或截取分析结果截图。 |
| 测试结果\*：**(测试日志 或 截图)** |  |
| 测试结论\* | *通过/有条件通过/不通过* |
| 备 注： | *1. 非源码软件应用，本例必选。多组件单个软件包的场景，可以一次性分析；多组件多个软件包的场景，请分多次分析和执行本条用例的测试。**2. 本用例不涉及待认证应用的性能检查，用户能提供预期结果1中的截图（用户提供该截图时需包含自己电脑右下角屏幕的任务栏时间显示）或者预期结果2中的截图，都可以认为用户的分析是成功的。**3. 最初分析报告在迁移前分析，最后的分析报告在迁移完成后执行。迁移前后的两份报告文件保存时请在名称上做出区分。* |

## 源码迁移完整性扫描（C/C++/Fortran/Go/Python/Java/Scala/汇编等源码包）

|  |  |  |  |
| --- | --- | --- | --- |
| 用例模块\*： | 兼容性测试 | 子模块： | 软件迁移 |
| 用例编号： | 03 |
| 用例名称： | Porting\_002软件源代码可通过Porting Advisor的源码迁移分析 |
| 用例目的\*： | C/C++/ASM/Fortran/Go/Python/Java/Scala等语言软件的源代码在迁移前或已完成迁移后，检查源代码是否存在待修复的点或验证确认已完成迁移源代码的修改点。 |
| 预置条件\*： | 1. 目标服务器已配置相应的YUM/APT/ZYPPER源。2. 获取鲲鹏DevKit安装包，并解压进入。3. 执行命令安装：./install.sh ，并安装代码迁移工具完成。4. 拷贝安装完成后的工具URL地址到浏览器（Chrome/Edge）：https://$IP:$Port默认用户：devadmin 密码：自定义（IP和Port存在映射的需要映射场景） |
| 测试步骤\*： | 1. 登录鲲鹏DevKit首页。2. 点击新建“快速创建鲲鹏工程或任务”，选择“迁移”功能中“源码迁移”，点击“确认”。3. 默认任务名称，选择“上传”的压缩包（自动解压）或文件夹，勾选“源码类型”，选择“目标操作系统”、“编译器版本”、“构建工具”，输入对应的“编译命令”，最后点击“开始分析”，参考如下图： |
| 预期结果\*： | 成功生成软件迁移评估报告，且评估成功。源码分析报告结果一：1. 软件迁移评估成功，用户进行判断/修复/验证。2. 下载分析报告或截取分析结果截图（修改前/验证后）。源码分析报告结果二：1. 提示无需要评估的内容，则不需要做改动。2. 下载分析报告或截取分析结果截图。 |
| 测试结果\*：**(测试日志 或 截图)** |  |
| 测试结论\* | *通过/有条件通过/不通过* |
| 备 注： | *1. C、C++、Fortran、Go、Python、Java、Scala、汇编语言源码，本例必选。其它不在列表内的编译型语言应用，可不执行本例。多组件一次性编译的场景，可以一次性分析；需要多次编译的场景，请分多次分析和执行本条用例的测试。**2. 本用例不涉及待认证应用的性能检查，用户能提供预期结果1中的截图（用户提供该截图时需包含自己电脑右下角屏幕的任务栏时间显示）或者预期结果2中的截图，都可以认为用户的分析是成功的。**3. 最初分析报告在迁移前分析，最后的分析报告在迁移完成后执行。迁移前后的两份报告文件保存时请在名称上做出区分。* |

# 安全测试（必选）

## 防病毒扫描

|  |  |  |  |
| --- | --- | --- | --- |
| 用例模块\*： | 安全测试 | 子模块： | 防病毒扫描 |
| 用例编号： | 05 |
| 用例名称： | 软件防病毒扫描 |
| 用例目的\*： | 对产品及产品依赖的软件进行防病毒扫描 |
| 预置条件\*： | 以下条件二选一：1． 使用自己公司购买的商用杀毒软件进行扫描；2． 使用开源软件clamav进行扫描。 |
| 测试步骤\*： | 以下步骤二选一：1． 使用自己公司购买的商用杀毒软件进行扫描：按照平时产品扫描的步骤进行病毒扫描即可，然后提供扫描结果；2． 使用开源软件进行扫描：请参考旗舰店提供的clamav工具的使用指南，然后提供扫描结果。 |
| 预期结果\*： | 产品及产品依赖的软件没有病毒 |
| 测试结果\*：**(测试日志 或 截图)** |  |
| 测试结论\* | *通过/有条件通过/不通过* |
| 备 注： | *若不涉及性能，可在此备注说明* |

## CVE 漏洞扫描

|  |  |  |  |
| --- | --- | --- | --- |
| 用例模块\*： | 安全测试 | 子模块： | CVE漏洞扫描 |
| 用例编号： | 06 |
| 用例名称： | 软件CVE漏洞扫描 |
| 用例目的\*： | 检查产品及产品依赖软件的CVE漏洞 |
| 预置条件\*： | 以下条件二选一：1. 有自己的CVE漏洞检查工具；2. 使用旗舰店提供的CVE漏洞检查工具。 |
| 测试步骤\*： | 以下步骤二选一：1. 如果使用自己的CVE漏洞检查工具：按照产品平时漏洞扫描的步骤扫描即可，然后提供扫描结果；2. 使用旗舰店提供的CVE漏洞检查工具：参考随工具一起发布的使用指南，然后提供扫描结果。 |
| 预期结果\*： | 产品及产品依赖的软件没有CVE漏洞 |
| 测试结果\*：**(测试日志 或 截图)** |  |
| 测试结论\* | *通过/有条件通过/不通过* |
| 备 注： | *若不涉及性能，可在此备注说明* |

# 软件性能比对测试（必选）

鲲鹏性能分析工具支持鲲鹏平台上的系统性能分析、Java性能分析和系统诊断，提供系统全景及常见应用场景下的性能采集和分析功能，并基于调优专家系统给出优化建议。

工具包获取：<https://www.hikunpeng.com/developer/devkit/download>

## 软件性能对比测试（调优助手）

|  |  |  |  |
| --- | --- | --- | --- |
| 用例模块\*： | 性能测试 | 子模块： | 性能比对 |
| 用例编号： | 07 |
| 用例名称： | tuning\_001分析待认证程序，导出采集分析报告 |
| 用例目的\*： | 通过该测试用例，分析定位性能指标和瓶颈点，辅助用户快速定位和处理性能问题。 |
| 预置条件\*： | 1. 目标服务器已配置相应的YUM/APT/ZYPPER源。2. 获取鲲鹏DevKit安装包，并解压进入。3. 执行命令安装：./install.sh ，并安装性能分析工具完成。4、拷贝安装完成后的工具URL地址到浏览器（Chrome/Edge）：https://$IP:$Port默认用户：devadmin 密码：自定义（IP和Port存在映射的需要映射场景） |
| 测试步骤\*： | 1. 登录鲲鹏DevKit首页。2. 点击新建“快速创建鲲鹏工程或任务”，默认选择“调优”功能中“调优助手”，点击“确认”。3. 默认任务名称，选择待分析的“目标节点”，分析对象“应用”，根据情况选择Launch Application（由Hyper tuner启动待测软件）或者Attach to Process（待测软件已启动，根据进程名或者PID跟踪应用详情）模式，选择采样时长（默认15s）和采集文件大小（默认100m），开始分析任务采集，如下图：4. 待任务执行完成后查看调优建议，并参考建议进行调优。5. 调优后再次执行压测和调优助手分析，再次查看结果。 |
| 预期结果\*： | 1. 任务完成，查看分析报告，各页签收集到相应数据信息。右上方“建议范围”选择“经过阈值过滤的建议”，之后依次查看 “系统配置”、“热点函数”、“系统性能”、“进程/线程性能”的优化建议，并截图（调优前和调优后）2. 在侧边栏选择“查看任务基本信息”，查看后“任务状态”为“已完成”；“采集结束原因”为“应用运行结束”或“到达任务采集时间”。参考如下截图： |
| 测试结果\*：**(测试日志 或 截图)** |  |
| 测试结论\* | *通过/有条件通过/不通过* |
| 备 注： | *1. 性能分析用例为可选项，推荐用户执行。在选择执行性能分析用例的情况下，本条用例为必选用例。**2. 用户能提供性能分析报告“任务信息”页签截图及报告导出文件的情况下，只要预期结果中的截图“任务状态”、“采集结束原因”、“分析对象”几个字段与截图一致，即可认为用例结果是OK的。**3. 首次分析在优化前执行，最后分析在优化后执行。优化前后的两份报告文件保存时请在名称上做出区分。如果用户根据分析报告，未发现优化建议，请额外说明。* |

# 功能测试（必选）

功能测试用例需覆盖对功能清单中2级功能进行100%覆盖。其中测试用例的级别可以参考如下定义，具体级别定义可以根据业务规定进行修改。

|  |  |
| --- | --- |
| 级别 | 定义 |
| 0级 | 用例包括各特性的主要的，常用的功能点；目的是为了初验测试时使用。 |
| 1级 | 完成所有基本功能点的覆盖，覆盖正向操作。最常使用和重要性程度高的基本功能。 |
| 2级 | 主要包括非有效边界值、异常输入等；完成所有功能点全方位的覆盖。这类测试用例重要性程度高，使用频率一般。 |
| 3级 | 1 除了功能测试以外的测试用例，其中容错测试，疲劳测试，安全测试不算在专项测试的范围以内，因为它们是在功能测试执行的。2 根据测试特性划分，部分功能特性适合进行专项的测试。这部分用例一般不和功能测试一起执行，但是具有相当高的重要性。是否每个版本执行，需要根据测试类型，特性进行细分。 |
| 4级 | 属于功能测试的非常异常的，难以测试的用例，不在每个版本测试执行，同时很少使用，重要性程度低。 |

## Xxxx

**功能用例至少要10个**

|  |  |  |  |
| --- | --- | --- | --- |
| 用例模块\*： | 功能测试 | 子模块： | 无 |
| 用例编号： |  |
| 用例名称： |  |
| 用例目的\*： |  |
| 预置条件\*： |  |
| 测试步骤\*： |  |
| 预期结果\*： |  |
| 测试结果\*：**(测试日志 或 截图)** |  |
| 测试结论\* | *通过/有条件通过/不通过* |
| 备 注： | *若不涉及性能，可在此备注说明* |

提示：多个测试用例拷贝上面表格模板并填写即可， 注意表格之间不要连起来。**测试结论一定要填写（通过/有条件通过/不通过，三选一）**。

# 附：补充说明

## 模板填写规范

* 带\*符号为必填项
* 测试用例执行模板格式请勿修改，以及用例模块只有3个值功能测试、业务性能测试和基础性能测试，请勿修改，后续需在线识别此关键词
* 测试结果，建议提供按照测试步骤测试之后的测试结果日志或截图，以便更好说明
* 测试结论，一定要填写（通过/有条件通过/不通过，三选一）
* 备注，若某用例未测试通过或有条件通过，则在此备注说明具体原因即可

示例：

|  |  |  |  |
| --- | --- | --- | --- |
| 用例模块\*： | 业务性能测试 | 子模块： | 无 |
| 用例编号： |  |
| 用例名称： | Performance\_http\_csv\_01 或 性能测试-http |
| 用例目的\*： | 通过该测试用例，测试在X平台上的http加载数据写入性能 |
| 预置条件\*： | 1）X平台已正常部署2）性能监控工具已准备 |
| 测试步骤\*： | 1. 确保各服务正常工作
2. 执行XX操作
3. 启动多个http客户端程序，进行数据加载，通过dstat命令查看网络速率，速率不增之后稳定运行5min
4. 计算http接收数据性能
 |
| 预期结果\*： | http接收数据性能不低于100M/s |
| 测试结果\*：**(测试日志 或 截图)** | Max：350 MB/sMin：90 MB/sAvg：150 MB/s |
| 测试结论\* | 通过  |
| 备 注： | X86对标结果Max：380 MB/sMin：89 MB/sAvg：101 MB/s |

## 结论描述约定

|  |  |
| --- | --- |
| 测试结论 | 具体描述 |
| 通过 | 按照用例的预置条件和测试步骤，测试结果与预期结果完全符合。 |
| 有条件通过 | 按照用例的预置条件和测试步骤，测试结果与预期结果不符合，但因预置条件问题且可澄清为非被测目标软件自身问题导致的不符合。 |
| 不通过 | 按照用例的预置条件和测试步骤，测试结果与预期结果不符合。 |