



CentOS 迁移适配工具说明书

麒麟软件有限公司

2022 年 6 月

目录

CENTOS 迁移适配工具说明书	1
引言.....	1
一. 工具集总体介绍.....	2
1. <i>x2kylin_ex</i> 工具说明.....	2
2. <i>kyreplace</i> 工具说明.....	3
3. <i>kycomp</i> 工具说明.....	4
4. <i>Kylinch</i> 工具说明.....	5
二. X2KYLIN_EX 使用指南.....	6
1. 工具特性描述.....	6
2. 安装依赖.....	7
3. 使用方法.....	7
4. 迁移评估报告内容说明.....	13
三. KYREPLACE 使用指南.....	16
1. 工具安装.....	16
2. 迁移初始化.....	16
3. 迁移前评估.....	16
4. 分析评估报告.....	17
5. 迁移实施.....	20
6. 处理桌面环境.....	20
7. 迁移后验证.....	21

8. 注意事项	21
9. 迁移案例	22
10. 后续升级计划	32
四. KYCOMP 使用指南	33
1. 使用方式	33
2. 工具的价值与效果	33
3. 获取方式	33
五. KYLINCH 使用指南	37
1. 测试环境	37
2. 测试要求	37
3. 使用步骤说明	38
4. 提交测试日志	54
5. 测试项说明	54
6. FAQ	92

引言

2020 年 12 月，CentOS 社区宣布 CentOS 服务器操作系统 8 和 7 系列分别于 2021 年底和 2024 年 6 月底停止服务。CentOS 是由开源社区免费提供的服务器操作系统，停服将导致系统安全漏洞无社区支持，给部署在服务器上的应用系统带来严重的安全隐患。CentOS 停服给用户带来重大系统安全和业务连续性风险、操作系统选择等问题。

针对 CentOS 操作系统的迁移成为未来用户需要面对的重要课题。由于对 CentOS 的迁移并不是简单的系统重新安装，而是需要对操作系统及其上搭载的应用软件 and 业务系统进行替代、适配、迁移和重构。CentOS 迁移适配工具是麒麟软件有限公司为配合 CentOS 迁移工作自主研发的迁移适配工具集，本文档是针对 CentOS 迁移适配工具形成的工具说明书，可以帮助用户在应用软件和业务系统进行替代、适配、迁移和重构中评估迁移工作量，降低风险，提升效率。

本文档麒麟软件提供 CentOS 迁移适配工具：

`Ky_replace` 工具能够对已部署业务的 CentOS 系统环境进行一键化就地系统迁移实施，保留原有业务软件，且系统完成了向目标麒麟操作系统的切换。

`x2kylin_ex` 能够在迁移实施前，为客户提供业务软件从 Centos7.6、8.2 向银河麒麟高级服务器操作系统 V10 迁移的软件包依赖及接口依赖分析。

`Kylinch` 提供对目标整机系统迁移到银河麒麟高级服务器操作系统 V10 的兼容性和性能进行迁移测试评估；

`kycomp` 是对原 CentOS 操作系统与银河麒麟高级服务器操作系统 V10 进行差异对比分析工具。

一. 工具集总体介绍

麒麟迁移评估工具集旨在为客户在操作系统替换及业务应用迁移等场景下提供有效、准确的迁移评估分析，以及高效、快捷的迁移实施方法。工具集包括 x2kylin_ex、ky_replace、kycomp、Kylinch 四款工具，工具集概况如下：

工具名称	x2kylin_ex	ky_replace	kycomp	kylinch
支撑场景	迁移前评估	迁移前评估 迁移实施	迁移前评估	迁移后测试
功能概述	系统差异分析、系统一键化替换	业务软件依赖分析、接口分析	系统差异分析	系统硬件兼容性测试
功能描述	业务软件与操作系统兼容性评估工具，帮助用户分析业务软件与目标麒麟操作系统的兼容度与差异识别	操作系统迁移评估与迁移实施工具，帮助用户快速平滑的将原系统替换至麒麟系统	操作系统软件包差异对比工具，帮助用户从软件包维度比较不同操作系统的差异	自动探测系统的硬件配置信息，对硬件设备通过执行相关的测试用例进行兼容性和性能等测试
CentOS 版本	CentOS 7.6 CentOS 8.2	Centos7.4~7.6 Centos7.7~7.9 Centos8.0~8.2	CentOS 7.6 CentOS 8.2	不涉及

1. x2kylin_ex 工具说明

x2kylin_ex 工具提供用户软件从 CentOS7.6 或 CentOS8.2 向 Kylin Linux

Advanced Server V10 (SP2)系统的迁移评估。该工具具备对待迁移软件包进行文件扫描、依赖识别、运行软件环境对齐和对运行结果分析等功能。特别地，此工具实现了对软件包的编写语言和架构的自动识别，用户无需再指定待迁移软件包语言，架构等信息便可进行迁移评估。迁移评估结束后将会生成数据来协助用户进行迁移影响分析及修改指导，生成的数据包括需要引入的依赖 rpm 包，需要修改的依赖 rpm 包中的具体函数原型声明等信息，由此实现了自动化的迁评估。

2. kyreplace 工具说明

麒麟 centos 替换工具——kyreplace，主要功能为将 centos 系统原地替换为麒麟操作系统。工具支持自动识别当前系统版本，支持自主设置目标迁移系统，支持迁移前评估（给出可视化评估报告）及一键式迁移（给出可视化迁移报告）功能。

待迁移系统	目标操作系统
Centos8.0~8.2	银河麒麟高级服务器操作系统(兼容版)V10
Centos7.7~7.9	银河麒麟高级服务器操作系统(兼容版)V10
	中标麒麟高级服务器操作系统软件 V7.0Update9
Centos7.0~7.6	银河麒麟高级服务器操作系统(兼容版)V10
	中标麒麟高级服务器操作系统软件 V7.0Update9
	中标麒麟高级服务器操作系统软件 V7.0Update6

参数说明：

```
[root@localhost ~]# kyreplace -h
usage: kyreplace [-h] [-init INIT] [-check] [-c] [-change2ukui] [-p]

optional arguments:
  -h, --help            show this help message and exit
  -init INIT            Set target os version and init the migration environment.
  -check                Pre-migration assessment.
  -c                    Continue to migration(only for centos7 to ns7.6/ns7.9 or
                        centos8 to ns8.2).
  -change2ukui          Change desktop to UKUI(after migration).
  -p                    Print migrate progress.
[root@localhost ~]#
```

- -h 查询帮助
- -init 初始化工具，配置目标系统为麒麟 7.6、7.9 或 8.2，如
kyreplace -init 7.9)
- -check 执行迁移前评估，并输出评估报告
- -c 支持迁移中断后继续迁移（目前仅支持 centos7.0-7.9 迁移
到 ns7.6/7.9，或 centos8.0-8.2 迁移到 8.2）
- -change2ukui 迁移后环境检查，并将桌面环境转换为麒麟的 UKUI
- -p 获取迁移进度百分比

3. kycomp 工具说明

kycomp 操作系统差异对比分析工具是对待迁移操作系统与麒麟服务器操作系统进行差异分析的工具。具备对比软件包差异能力（包名变化、子包数量变化）、接口差异能力（命令差异、服务差异、库差异、函数差异），并将差异对比结果通过网页访问页面进行展示。

4. Kylinch 工具说明

kylinch 整机认证测试是为了保证麒麟操作系统与硬件设备之间具有良好的协同性。它是一款操作灵活、功能强大的测试工具，它可以自动探测系统的硬件配置信息，并针对已检测到的硬件设备通过执行相关的测试用例进行兼容性和性能等测试。

kylinch 是我们对厂商产品进行整机认证时所使用的一款测试工具。主要包含客户端模块和服务端模块。客户端用于测试执行，服务端除了为网络测试提供 server 端以外，还提供了测试日志收集、在线查看以及下载。

二. x2kylin_ex 使用指南

此文档主要对服务器迁移评估工具 x2kylin_ex 的安装、使用以及测试用例进行了介绍。

1. 工具特性描述

x2kylin_ex 工具功能特性描述如下：

(1) 支持待分析软件单包 (rpm、tar.gz 压缩格式) 扫描或目录扫描，目录扫描用于评估一个软件含有多个软件包的情况，此时需用户将待分析的软件包组拷贝到扫描目录下；

(2) 工具支持从源 OS 到目标 OS 的软件迁移评估，源 OS 可选 CentOS7.6 (默认) 或 CentOS8.2，目标 OS 为 Kylin Linux Advanced Server V10 (SP2)；

(3) 工具支持自动识别软件包的编程语言，能够识别的有效语言包括 C/C++、Java、Python，支持识别软件使用多种语言实现的场景；

(4) 工具支持自动识别软件包适用的 CPU 架构，能够识别的有效架构包括 x86_64 (默认)、aarch64；

(5) 工具支持分析的迁移环境差异包括：直接依赖 rpm 包差异、rpm 中包含的 so 库差异和 so 库对外提供的代码级别的差异。这些差异信息将会以 html 文件的形式呈现，同时差异信息也会以 json 数据形式保存在以软件包命名的 json 文件中。

2. 安装依赖

```
python3 ( python3.7.x )  
sqlite ( 3.7.17 版本及以上 )  
python3-pip  
net-tools  
pciutils  
systemd  
dmidecode  
util-linux
```

3. 使用方法

工具建议由 root 用户或具有 sudo 权限的普通用户执行，具体使用方法如下：

3.1. 工具部署

(1) 获取到此工具的压缩包 x2kylin_ex-1.0-1.v2.0.tar.gz 后，执行以下命令，进行工具部署

```
# tar -zxvf x2kylin_ex-1.0-1.v2.0.tar.gz
```

```
[root@localhost x2kylin]# ls  
log4j  x2kylin_ex-1.0-1.v2.0.tar.gz  
[root@localhost x2kylin]# tar -zxvf x2kylin_ex-1.0-1.v2.0.tar.gz █
```

(2) 进入解压目录

```
# cd x2kylin_ex-1.0-1.v2.0
```

```
[root@localhost x2kylin]# ls
log4j  x2kylin_ex-1.0-1.v2.0  x2kylin_ex-1.0-1.v2.0.tar.gz
[root@localhost x2kylin]# cd x2kylin_ex-1.0-1.v2.0/
[root@localhost x2kylin_ex-1.0-1.v2.0]# ls
LICENSE  README.md  ReleaseNotes.md  x2kylin_ex  x2kylin_ex.py
[root@localhost x2kylin_ex-1.0-1.v2.0]#
```

3.2. 使用工具进行迁移评估扫描

3.2.1. 查看帮助手册

执行以下命令，查看帮助手册：

```
# ./x2kylin_ex.py -h
```

```
[root@localhost x2kylin_ex-1.0-1.v2.0]# ls
LICENSE  README.md  ReleaseNotes.md  x2kylin_ex  x2kylin_ex.py
[root@localhost x2kylin_ex-1.0-1.v2.0]# ./x2kylin_ex.py -h
usage: x2kylin_ex.py [-h] {scan,conf-analyse,conf-collect} ...

<===== x2kylin_ex Usage Manual =====>

positional arguments:
  {scan,conf-analyse,conf-collect}
                        sub-command help
  scan                  migration assessment tool
  conf-analyse          analyse system configures and generate report
  conf-collect          collect system configures

optional arguments:
  -h, --help            show this help message and exit

e.g.: x2kylin_ex /home/package/package.x86_64.rpm
[root@localhost x2kylin_ex-1.0-1.v2.0]#
```

参数说明：

(1) scan：迁移评估功能

必选参数：

- PATH，指定需要扫描的单个 rpm 包路径，或指定需要扫描的路径文件夹(不支持嵌套目录，只支持单层目录)。

可选参数:

-os , 指定迁移分析的源操作系统, 支持 CentOS7 (7.6)/CentOS8 (8.2)。

-lang , 指定迁移分析软件包的语言, 支持 c/java/python , 可以同时指定一种或以上语言类型。

-arch , 指定迁移分析的软件包架构, 支持 x86_64/aarch64。

-t , 指定迁移分析软件包的扫描方式类型, 支持 single/batch

- single , 扫描目录下所有软件包, 对扫描结果进行整合。

- batch , 扫描目录下所有软件, 每一个软件包分开展示。

(2) conf-analyse : 分析系统配置并生成报告

只有在物理机 + CentOS 系统上, 才能正常生成报告。

(3) conf-collect : 收集系统配置

收集系统信息, 如 linux 命令等。

3.2.2. 指定 -os 参数

准备待迁移评估软件包, 例如 git-2.27.0-5.ky10.x86_64.rpm :

执行以下扫描命令, 通过-os 参数指定迁移源操作系统, 若不指定, 默认为 CentOS7.6。

```
# ./x2kylin_ex.py scan git-2.27.0-5.ky10.x86_64.rpm -os centos8
```

```
[root@localhost x2kylin_ex-1.0-1.v2.0]# ls
git-2.27.0-5.ky10.x86_64.rpm LICENSE README.md ReleaseNotes.md x2kylin_ex x2kylin_ex.py
[root@localhost x2kylin_ex-1.0-1.v2.0]# ./x2kylin_ex.py scan git-2.27.0-5.ky10.x86_64.rpm -os centos8
centos8
None
None
single
*****
begin to check install_requires...
install_requires check passed
*****
begin to parse language...
package language : c
*****
begin to parse architecture...
package architecture : x86_64
----Scanning...
Decompressing package git-2.27.0-5.ky10.x86_64.rpm...
Updating result...
The results are saved: /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/x2kylin/x2kylin_8.2/output/git-2.27.0-5.ky10.x86_64-20220509173327.html
*****
output json file is saved in the /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/output/centos8-git-2.27.0-5.ky10.x86_64-20220509173327.html
output html file is saved in the /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/output/centos8-git-2.27.0-5.ky10.x86_64-20220509173327.html
===x2kylin_ex execute successful===
[root@localhost x2kylin_ex-1.0-1.v2.0]#
```

3.2.3. 指定 -lang 参数

执行以下命令，通过 -lang 参数指定待迁移的软件包语言，若不指定，工具内部自动识别。

```
# ./x2kylin_ex.py scan git-2.27.0-5.ky10.x86_64.rpm -os centos8 -lang c
```

```
[root@localhost x2kylin_ex-1.0-1.v2.0]# ls
git-2.27.0-5.ky10.x86_64.rpm LICENSE README.md ReleaseNotes.md x2kylin_ex x2kylin_ex.py
[root@localhost x2kylin_ex-1.0-1.v2.0]# ./x2kylin_ex.py scan git-2.27.0-5.ky10.x86_64.rpm -os centos8 -lang c
centos8
{'c'}
None
single
*****
begin to check install_requires...
install_requires check passed
*****
begin to parse architecture...
package architecture : x86_64
----Scanning...
Decompressing package git-2.27.0-5.ky10.x86_64.rpm...
Updating result...
The results are saved: /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/x2kylin/x2kylin_8.2/output/git-2.27.0-5.ky10.x86_64-20220509173449.html
*****
output json file is saved in the /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/output/centos8-git-2.27.0-5.ky10.x86_64-20220509173449.html
output html file is saved in the /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/output/centos8-git-2.27.0-5.ky10.x86_64-20220509173449.html
===x2kylin_ex execute successful===
[root@localhost x2kylin_ex-1.0-1.v2.0]#
```

3.2.4. 指定-arch 参数

执行以下命令，通过-arch 参数指定待迁移的软件包架构，若不指定，工具内部进行自动识别;

```
# ./x2kylin_ex.py scan git-2.27.0-5.ky10.x86_64.rpm -os centos8
-lang c -arch x86_64
```

```
[root@localhost x2kylin_ex-1.0-1.v2.0]# ls
git-2.27.0-5.ky10.x86_64.rpm LICENSE README.md ReleaseNotes.md x2kylin_ex x2kylin_ex.py
[root@localhost x2kylin_ex-1.0-1.v2.0]# ./x2kylin_ex.py scan git-2.27.0-5.ky10.x86_64.rpm -os centos8 -lang c -arch x86_64
git-2.27.0-5.ky10.x86_64.rpm
centos8
{'c'}
x86_64
single
*****
begin to check install_requires...
install_requires check passed
----Scanning...
Decompressing package git-2.27.0-5.ky10.x86_64.rpm...
Updating result...
The results are saved: /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/x2kylin/x2kylin_8.2/output/git-2.27.0-5.ky10.x86_64-ml
*****
output json file is saved in the /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/output/centos8-git-2.27.0-5.ky10.x86_64.js
/root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/output/centos8-git-2.27.0-5.ky10.x86_64-20220509173727.html
centos8-git-2.27.0-5.ky10.x86_64-20220509173727.html
output html file is saved in the /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/output/centos8-git-2.27.0-5.ky10.x86_64-20220509173727.html
===x2kylin_ex execute successful===
[root@localhost x2kylin_ex-1.0-1.v2.0]#
```

3.2.5. 扫描软件包组

准备待迁移评估的软件包组，例如 log4j:

```
[root@localhost x2kylin_ex-1.0-1.v2.0]# ls
git-2.27.0-5.ky10.x86_64.rpm LICENSE log4j README.md ReleaseNotes.md x2kylin_ex x2kylin_ex.py
[root@localhost x2kylin_ex-1.0-1.v2.0]# ls log4j/
log4j-2.17.0-3.p01.ky10.noarch.rpm log4j-jcl-2.17.0-3.p01.ky10.noarch.rpm log4j-taglib-2.17.0-3.p01.ky10.noarch.rpm
log4j-bom-2.17.0-3.p01.ky10.noarch.rpm log4j-jmx-gui-2.17.0-3.p01.ky10.noarch.rpm log4j-web-2.17.0-3.p01.ky10.noarch.rpm
log4j-help-2.17.0-3.p01.ky10.noarch.rpm log4j-slf4j-2.17.0-3.p01.ky10.noarch.rpm
[root@localhost x2kylin_ex-1.0-1.v2.0]#
```

执行以下命令，通过-t 参数，指定迁移分析软件包的扫描方式类型，-t single 扫

描目录下所有软件包并将扫描结果整合，-t batch 扫描目录下所有软件包，每个软件包扫描结果分开展示;

```
# x2kylin_ex ./log4j/ -t single
```

```
[root@localhost x2kylin_ex-1.0-1.v2.0]# ls
git-2.27.0-5.ky10.x86_64.rpm LICENSE log4j README.md ReleaseNotes.md x2kylin_ex x2kylin_ex.py
[root@localhost x2kylin_ex-1.0-1.v2.0]# ./x2kylin_ex.py scan ./log4j/ -t single
./log4j/
centos7
None
None
single
*****
begin to check install_requires...
install_requires check passed
*****
begin to parse language...
package language : java
*****
begin to parse architecture...
package architecture : x86_64
----Scanning...
Producing report...
The results are saved: /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/x2kylin_7.6/output/log4j-20220509173953.html
*****
output json file is saved in the /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/output/centos7-log4j.json
/root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/output/centos7-log4j-20220509173954.html
centos7-log4j-20220509173954.html
output html file is saved in the /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/output/centos7-log4j-20220509173954.html
===x2kylin_ex execute successfull===
[root@localhost x2kylin_ex-1.0-1.v2.0]#
```

./x2kylin_ex.py scan ./log4j/ -t batch

```
[root@localhost x2kylin_ex-1.0-1.v2.0]# ls
git-2.27.0-5.ky10.x86_64.rpm LICENSE log4j README.md ReleaseNotes.md x2kylin_ex x2kylin_ex.py
[root@localhost x2kylin_ex-1.0-1.v2.0]# ./x2kylin_ex.py scan ./log4j/ -t batch
./log4j/
centos7
None
None
batch
*****
begin to check install_requires...
install_requires check passed
*****
begin to parse language...
package language : java
*****
begin to parse architecture...
package architecture : x86_64
----Scanning...
Producing report...
The results are saved: /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/x2kylin_7.6/output/log4j-2.17.0-3.p01.ky10.noarch-20220509174036.html
Scanning...
Producing report...
The results are saved: /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/x2kylin_7.6/output/log4j-bom-2.17.0-3.p01.ky10.noarch-20220509174036.html
Scanning...
Producing report...
The results are saved: /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/x2kylin_7.6/output/log4j-help-2.17.0-3.p01.ky10.noarch-20220509174036.html
Scanning...
Producing report...
The results are saved: /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/x2kylin_7.6/output/log4j-jcl-2.17.0-3.p01.ky10.noarch-20220509174037.html
Scanning...
```

3.2.6. 简便操作

最简洁的操作命令为:

```
# ./x2kylin_ex.py scan git-2.27.0-5.ky10.x86_64.rpm
```

```
[root@localhost x2kylin_ex-1.0-1.v2.0]# ls
git-2.27.0-5.ky10.x86_64.rpm LICENSE log4j README.md ReleaseNotes.md x2kylin_ex x2kylin_ex.py
[root@localhost x2kylin_ex-1.0-1.v2.0]# ./x2kylin_ex.py scan git-2.27.0-5.ky10.x86_64.rpm
centos7
None
None
single
*****
begin to check install_requires...
install_requires check passed
*****
begin to parse language...
package language : c
*****
begin to parse architecture...
package architecture : x86_64
----Scanning...
Decompressing package git-2.27.0-5.ky10.x86_64.rpm...
Updating result...
The results are saved: /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/x2kylin/x2kylin_7.6/output/git-2.27.0-5.ky10.x86_64-20220509174219.html
*****
output json file is saved in the /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/output/centos7-git-2.27.0-5.ky10.x86_64.json
/root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/output/centos7-git-2.27.0-5.ky10.x86_64-20220509174221.html
centos7-git-2.27.0-5.ky10.x86_64-20220509174221.html
output html file is saved in the /root/test/x2kylin/x2kylin_ex-1.0-1.v2.0/x2kylin_ex/output/centos7-git-2.27.0-5.ky10.x86_64-20220509174221.html
==x2kylin_ex execute successful==
[root@localhost x2kylin_ex-1.0-1.v2.0]#
```

默认源操作系统为 CentOS7，工具会自动解析软件包语言，架构。

3.2.7. 收集系统配置

```
# ./x2kylin_ex.py conf-collect
```

```
[root@localhost x2kylin_ex-1.0-1.v2.0]# ./x2kylin_ex.py conf-collect
2022-05-09 17:44:37,689 INFO manager/get_param_config/174: Parameter configuration file loaded.
2022-05-09 17:44:37,705 INFO manager/get_regex_config/213: Regex pattern compiled.
2022-05-09 17:44:37,705 INFO manager/load_parsers/228: All builtin parsers loaded.
2022-05-09 17:44:37,705 INFO manager/load_parsers/231: All custom parsers loaded.
cat: '/usr/src/kernels/*/include/linux/syscalls.h': No such file or directory
2022-05-09 17:44:37,980 INFO time_utils/wrapper/21: 0.27s taken for running function [get_data]
2022-05-09 17:44:37,995 WARNING list/parse_content/47: no data in ls_dev
2022-05-09 17:44:38,047 INFO time_utils/wrapper/21: 0.07s taken for running function [get_parsed_content]
```

3.3. 输出说明

执行命令后会将评估报告保存在./x2kylin_ex/output 目录下

4. 迁移评估报告内容说明

4.1. 概要说明

html 报告的第一部分显示待迁移软件包的概要信息，包括软件包路径、软件包编写语言、软件包架构、迁移的源操作系统、迁移的目标操作系统，以及软

件包在源、目标系统上的兼容性对比结果。

【git-2.27.0-5.ky10.x86_64】迁移分析报告

一、概要信息

扫描路径	/root/test/git-2.27.0-5.ky10.x86_64.rpm
软件包语言	["c"]
软件包架构	x86_64
源操作系统	centos7.6
目标操作系统	Kylin-V10-SP2
兼容性结论	不兼容

4.2. 兼容性统计结果

兼容性结果分为依赖包兼容性和接口函数兼容性。

依赖包兼容统计包括依赖包的总数、在源/目标操作系统兼容的软件包数、不兼容的软件包数，以及在目标操作系统新增的依赖包数，依据这些数据，由公式：依赖包兼容率 = (兼容依赖包数) / (兼容依赖包数 + 不兼容依赖包数)，得出依赖包的兼容率。

接口兼容统计分为接口函数总数、在源/目标操作系统兼容的接口数和接口变化数，依据这些数据，由公式：接口兼容率 = 兼容接口总数 / 接口总数。

二、兼容性统计结果

依赖包结果统计：

依赖包总数	兼容依赖包数	不兼容依赖包数	新增支持依赖包数	依赖包兼容率
38	35	2	1	94.59%

接口函数结果统计：

接口函数总数	兼容接口数	接口变化数	接口兼容率
332	308	24	92.77%

4.3. 依赖软件包兼容性分析

此部分显示每个依赖包的详细信息，表格的结论栏展示此依赖包的兼容信息，为兼容、不兼容或新增；点击向左箭头，显示此依赖软件包提供的库文件。

三、依赖软件包兼容性分析

扫描的软件包: git-2.27.0-5.ky10.x86_64_c

No.	centos7.6	Kylin-V10-SP2	结论
> 20	psacct-0.0.1-13.el7.x86_64.rpm	psacct-0.0.4-4.ky10.x86_64.rpm	兼容
> 21	diffutils-3.3-4.el7.x86_64.rpm	diffutils-3.7-3.ky10.x86_64.rpm	兼容
> 22		openssl-libs-1.1.1f-4.p01.ky10.x86_6...	新增
∨ 23	libcurl-7.29.0-51.el7.x86_64.rpm	libcurl-7.71.1-2.ky10.x86_64.rpm	兼容
	libcurl.so.4()(64bit)	libcurl.so.4()(64bit)	package provide the corresponding d ependency
> 24	expat-2.1.0-10.el7_3.x86_64.rpm	expat-2.2.9-2.ky10.x86_64.rpm	兼容
> 25	glib2-2.56.1-2.el7.x86_64.rpm	glib2-2.62.5-2.ky10.x86_64.rpm	兼容
> 26	pcre2-10.23-2.el7.x86_64.rpm	pcre2-10.35-1.ky10.x86_64.rpm	兼容

4.4. 接口变化分析

第四部分展示接口变化信息，包括接口在源/，目标操作系统上的名称、调用此接口的进程，以及接口变化的描述。

四、接口变化分析

扫描的软件包: git-2.27.0-5.ky10.x86_64_c

No.	centos7.6	Kylin-V10-SP2	调用进程	差异描述
1	int pthread_cond_wait(pthread...	int pthread_cond_wait(pthread...	git	● The function parameter size changes
2	gpointer g_hash_table_lookup...	gpointer g_hash_table_lookup...	git-credential-libsecret	● The function parameter size changes
3	void g_hash_table_unref(GH...	void g_hash_table_unref(GH...	git-credential-libsecret	● The function parameter size changes
4	gboolean g_hash_table_insert...	gboolean g_hash_table_insert...	git-credential-libsecret	● The function parameter size changes
5	size_t *argv? get_quarter_n...	size_t *argv? get_quarter_n...	git-shell git-browse git...	● The function parameter size

三. kyreplace 使用指南

此文档主要对服务器迁移评估替换工具 kyreplace 的安装、使用以及测试用例进行了介绍。

1. 工具安装

命令行执行# rpm -ivh kyreplace-1.1.noarch.rpm

2. 迁移初始化

(1) 新建任务工作目录, 如/root/kyreplace, 并进入, 之后的所有操作均在该工作目录中。

(2) 命令行执行# kyreplace -init 7.6/7.9/8.2

注意: 用户可根据需求, 自主选择迁移到 7.6 (中标麒麟高级服务器操作系统软件 V7.0Update6)、7.9 (中标麒麟高级服务器操作系统软件 V7.0Update9) 或 8.2 (银河麒麟高级服务器操作系统(兼容版)V10), 只需在 init 参数后面加上目标系统即可。

3. 迁移前评估

命令行执行# kyreplace -check

4. 分析评估报告

评估报告有两种，分别为：

```
/var/tmp/kymigration/kyreplace/centos2kylin/assessment/assessment.html
```

```
/var/tmp/kymigration/kyreplace/leapp/assessment/assessment.html
```

具体报告位置会在评估结束后，显示在屏幕上。

(1) 若评估报告为：

```
/var/tmp/kymigration/kyreplace/centos2kylin/assessment/assessment.html
```

该评估报告的格式是 html 格式，包括概要信息、迁移风险评估、非 CentOS 软件包列表、迁移评估失败原因、迁移后依赖缺失 5 个部分。

第一部分“概要信息”，会列出环境信息，并给出是否可以迁移的提示，若提示为‘true’，则表示可以迁移；若提示为‘false’，则表示不可以迁移；

第二部分“迁移风险评估”，会列出当前环境的依赖缺失或软件包冲突问题，若没有此类问题，则该项不会列出；

第三部分“非 CentOS 软件包列表”，会列出不属于 CentOS 的软件包，告知用户这些软件包不会被替换，而是会被保存在迁移后的系统中；

第四部分“迁移评估失败原因”，会列出详细的迁移风险，若可以迁移，该项不会列出；

第五部分“迁移后依赖缺失”，会预估出业务软件在迁移完成后能否正常使用，若全部正常，该项不会列出。

总结：用户对迁移评估报告 `assessment.html` 进行分析，解决其中的迁移风险后可以重新执行 `# kyreplace -check` 进行评估，直到“是否可以迁移”为 `true`，方可正式执行迁移。

(2) 若评估报告为：

```
/var/tmp/kymigration/kyreplace/leapp/assessment/assessment.htm
```

|

评估命令执行完成后，会生成评估报告 `assessment.html`，报告中详细列举出当前 OS 与目标 OS 系统间的升级分析报告。该评估报告的格式是 `html` 格式，包括概要信息、评估风险提示、非 CentOS 软件包列表、迁移评估失败原因。

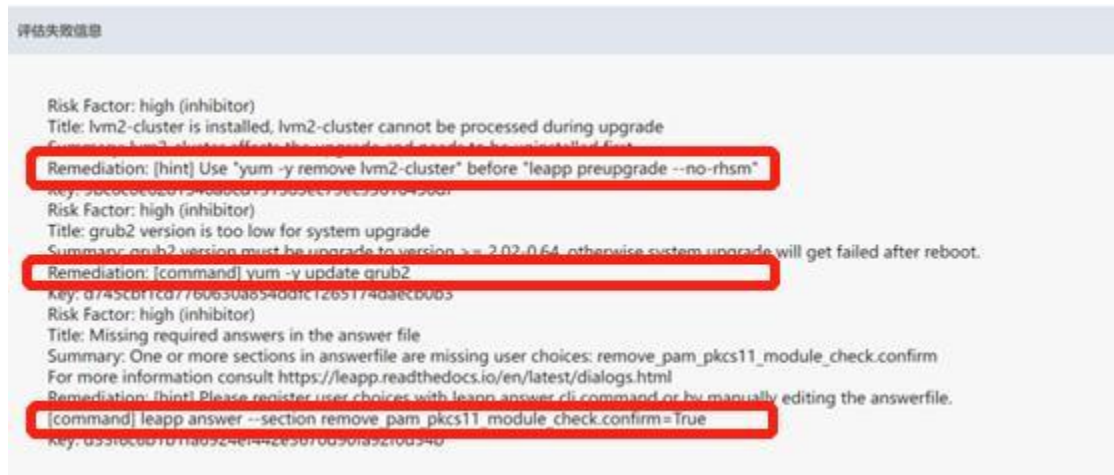
第一部分“概要信息”，会列出环境信息，并给出是否可以迁移的提示，若提示为 `'true'`，则表示可以迁移；若提示为 `'false'`，则表示不可以迁移；

第二部分“评估风险提示”，会列出迁移的风险，包括风险提示、风险等级及

解决建议；

第三部分“非 CentOS 软件包列表”，会列出不属于 CentOS 的软件包，告知用户这些软件包不会被替换，而是会被保存在迁移后的系统中；

第四部分“迁移评估失败原因”，会列出影响到操作系统迁移的问题。包含影响等级，影响概述，影响详细信息，推荐解决方案。这部分的问题必须解决，否则将无法迁移。以下图为例进行说明。



上图显示了某次迁移评估失败后，评估报告中的评估失败信息，详细列出了风险等级、问题描述及解决方案，只有将这部分问题全部解决，才可继续执行迁移。

参考其中 Remediation(修复建议)部分信息，分别执行以下三条命令：

```
yum -y remove lvm2-cluster
```

```
yum -y update grub2
```

```
leapp answer --section remove_pam_pkcs11_module_check.confirm
= True
```

不同版本的 CentOS 系统的问题也会不同，请参考 assessment.html 并根据实际问题进行修复。

解决了上述影响因素后，可以再次执行 `kyreplace -check` 再次评估是否有影响迁移的因素。若没有影响迁移的因素，assessment.html 报告如下：

非centos7软件包数

38

是否可以迁移

true

5. 迁移实施

```
# kyreplace
```

执行结束后 reboot 重启。

注意：在此过程中，可使用 `kyreplace -p` 命令来获取迁移进度。

6. 处理桌面环境

当迁移的目标系统为“银河麒麟高级服务器操作系统(兼容版)V10”时，如果原系统安装了图形桌面环境，需要通过此步骤处理桌面环境。

重启后，执行以下命令，对当前系统进行检查，若之前的系统有桌面环境，该命令会将其转换为 UKUI。

```
# kyreplace -change2ukui
```

如果无须处理桌面环境，则执行命令会打印 “There is no need to switch.” 的字样。

注意：若在执行转换的过程中遇到无法安装的问题，需要手动使用 `yum install` 进行安装，如：

```
完毕！
上次元数据过期检查：0:00:27 前，执行于 2022年04月24日 星期日 20时16分51秒。
错误：
问题：package libvirt-4.5.0-42.module_el8.2.0+97+a5dbdd5c.x86_64 requires libvirt-daemon-driver-qemu = 4.5.0-42.module_el8.2.0+97+a5dbdd5c, but none of the providers can be installed
- package libvirt-daemon-driver-qemu-4.5.0-42.module_el8.2.0+97+a5dbdd5c.x86_64 requires /usr/bin/qemu-img, but none of the providers can be installed
- conflicting requests
- package qemu-img-2.12.0-99.el8.x86_64 is filtered out by modular filtering
- package qemu-img-15:2.12.0-99.module_el8.2.0+204+7d93107e.ks8.01.x86_64 is filtered out by modular filtering
(尝试添加 '--skip-broken' 来跳过无法安装的软件包 或 '--nobest' 来不只显示最佳选择的软件包)
Something has gone wrong, please run kyreplace -change2ukui again.
[root@localhost ~]#
```

此时需要手动安装 `qemu-img` 软件包

```
yum install ;
```

```
http://update.cs2c.com.cn:8080/NS/V10/8U2/os/adv/lic/AppStream/x86\_64/Packages/qemu-img-2.12.0-99.module\_el8.2.0+204+7d93107e.ks8.
```

01.x86_64.rpm

```
已安装：  
  qemu-img-15:2.12.0-99.module_el8.2.0+204+7d93107e.ks8.01.x86_64  
完毕！  
[root@localhost ~]# kyreplace -change2ukui
```

安装成功后重新执行即可

```
# kyreplace -change2ukui
```

若该命令执行完毕之后，出现界面无法进入的问题，可以使用 ssh 连接到该机器，查看 lightdm 服务的状态，并尝试不断重启 lightdm 服务，直到出现麒麟的登陆界面。若多次重启 lightdm 服务后仍无法进入，则重启机器。

```
# systemctl status lightdm
```

```
# systemctl restart lightdm
```

7. 迁移后验证

迁移完成后可以使用 `nkvers` 查看版本信息和内核信息，并查看迁移报告 `/var/tmp/kymigration/kyreplace/centos2kylin/migration/migration.html`

或 `/var/tmp/kymigration/kyreplace/leapp/migration/migration.html`。

具体报告位置会在迁移结束后，显示在屏幕上。

8. 注意事项

(1) 用户在迁移前，必须明确自己的迁移目标，并通过 `kyreplace -init` 命

令去指定。

(2) centos7.0-7.2 系统,若想迁移到银河麒麟高级服务器操作系统(兼容版)V10,因磁盘格式默认 ftype=0,会导致工具无法正常使用,所以需要重新格式化磁盘修改 ftype=1,具体操作流程见《centos7.0-7.2 修改磁盘格式方案》



leapp7.0-7.2修改
磁盘格式方案.docx

由于 centos7 的软件包,与银河麒麟高级服务器操作系统(兼容版)V10 中的软件包版本差异较大,为防止某些软件在迁移后无法运行,不建议将 centos7 的系统迁移到银河麒麟高级服务器操作系统(兼容版)V10。

对于用户有自己搭建内网仓库并进行迁移需求的情况,参照《centos 迁移源配置方法-v1.0》进行迁移工具配置。

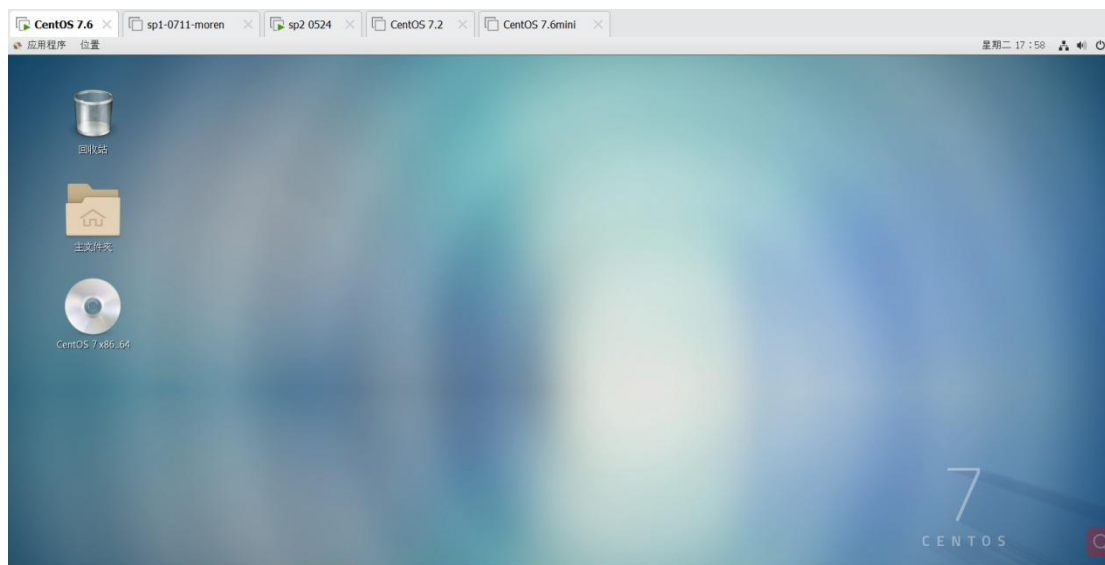


centos迁移源配置
方法-v1.0.docx

9. 迁移案例

9.1. CentOS 7.6 替换示例(目标系统麒麟 ns8.2)

CentOS 7.6(带 GUI 的服务器不勾选右侧安装)测试过程



安装并初始化

```
[root@localhost kyreplace]# kyreplace -init
*****
Initializing the environment for centos7
*****
已加载插件：fastestmirror, langpacks
Determining fastest mirrors
 * base: mirrors.aliyun.com
 * extras: mirrors.bfsu.edu.cn
 * updates: mirrors.aliyun.com
base                               | 3.6 kB    00:00
extras                             | 2.9 kB    00:00
updates                             | 2.9 kB    00:00
(1/4): extras/7/x86_64/primary_db   | 246 kB    00:00
(2/4): base/7/x86_64/group_gz       | 153 kB    00:02
(3/4): updates/7/x86_64/primary_db  | 15 MB     00:10
(4/4): base/7/x86_64/primary_db     | 6.1 MB    00:17
软件包 python-setuptools-0.9.8-7.el7.noarch 已安装并且是最新版本
无须任何处理
已加载插件：fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.bfsu.edu.cn
 * updates: mirrors.aliyun.com
正在解决依赖关系
--> 正在检查事务
--> 软件包 policycoreutils-python.x86_64.0.2.5-29.el7 将被 升级
--> 软件包 policycoreutils-python.x86_64.0.2.5-34.el7 将被 更新
--> 正在处理依赖关系 policycoreutils = 2.5-34.el7, 它被软件包 policycoreutils-python-2.5-34.el7.x86_64 需要
--> 正在检查事务
--> 软件包 policycoreutils.x86_64.0.2.5-29.el7 将被 升级
--> 软件包 policycoreutils.x86_64.0.2.5-34.el7 将被 更新
--> 解决依赖关系完成
```

执行 `kyreplace -check` 命令，进行升级前检测

```
[root@localhost kyreplace]# kyreplace -check
=> Processing phase `configuration_phase`
=> * ipu_workflow_config
    IPU workflow config actor
=> Processing phase `FactsCollection`
=> * persistentnetnames
    Get network interface information for physical ethernet interfaces of the original system.
=> * scan_sap_hana
    Gathers information related to SAP HANA instances on the system.
=> * persistentnetnamesdisable
    Disable systemd-udev persistent network naming on machine with single eth0 NIC
=> * scan_custom_repofile
    Scan the custom /etc/leapp/files/leapp_upgrade_repositories.repo repo file.
=> * authselect_scanner
    Detect what authselect configuration should be suggested to administrator.
=> * scan_kernel_cmdline
    No documentation has been provided for the scan_kernel_cmdline actor.
=> * system_facts
```

UPGRADE INHIBITED

Upgrade has been inhibited due to the following problems:

1. Inhibitor: Possible problems with remote login using root account
2. Inhibitor: Missing required answers in the answer file

Consult the pre-upgrade report for details and possible remediation.

检测报错，根据

<cc/var/tmp/kymigration/kyreplace/leapp/assessment/assessment.htm>

建议进行修复。

```
[root@localhost noarch]# leapp answer --section remove_pam_pkcs11_module_check.c
onfirm=True
```

```
[root@localhost ~]# sed -i 's/#PermitRootLogin yes/PermitRootLogin yes/' /etc/ss
h/sshd_config
```

再次执行命令，成功通过检测

```
[root@localhost kyreplace]# kyreplace -check
=> Processing phase `configuration_phase`
=> * ipu_workflow_config
    IPU workflow config actor
=> Processing phase `FactsCollection`
=> * persistentnetnames
    Get network interface information for physical ethernet interfaces of the original system.
=> * scan_sap_hana
    Gathers information related to SAP HANA instances on the system.
=> * persistentnetnamesdisable
    Disable systemd-udev persistent network naming on machine with single eth0 NIC
=> * scan_custom_repofile
    Scan the custom /etc/leapp/files/leapp_upgrade_repositories.repo repo file.
=> * authselect_scanner
    Detect what authselect configuration should be suggested to administrator.
=> * scan_kernel_cmdline
    No documentation has been provided for the scan_kernel_cmdline actor.
=> * system_facts
```

```

=====
                        REPORT
=====

A report has been generated at /var/log/leapp/leapp-report.json
A report has been generated at /var/log/leapp/leapp-report.txt

=====
                        END OF REPORT
=====
    
```

执行# kyreplace 进行升级

```

[root@localhost etc] # kyreplace
=> Processing phase `configuration_phase`
=> * ipu_workflow_config
    IPU workflow config actor
=> Processing phase `FactsCollection`
=> * scanlienablerepo
    Produce CustomTargetRepository based on the LEAPP_ENABLE_REPOS in config.
=> * removed_pam_modules_scanner
    Scan PAM configuration for modules that are not available in Kylin_ns8.2.
=> * root_scanner
    Scan the system root directory and produce a message containing
=> * network_manager_read_config
    Provides data about NetworkManager configuration.
=> * scancpu
    Scan CPUs of the machine.
=> * scanmemory
    Scan Memory of the machine.
=> * scan_sap_hana
    Gathers information related to SAP HANA instances on the system.
=> * storage_scanner
    Provides data about storage settings.
=> * read_openssh_config
    Collect information about the OpenSSH configuration.
=> * firewallld_facts_actor

(1362/1530): qgnomeplatform-0.4-3.el8.x86_64.rpm 542 kB/s | 135 kB    00:00
(1363/1530): qpdf-libs-7.1.1-10.el8.x86_64.rpm 610 kB/s | 335 kB    00:00
(1364/1530): qrencode-libs-3.4.4-5.el8.x86_64.r 613 kB/s | 58 kB    00:00
(1365/1530): qemu-kvm-common-2.12.0-99.el8.x86_ 609 kB/s | 931 kB    00:01
(1366/1530): qt5-qtbase-common-5.12.5-4.el8.noa 966 kB/s | 39 kB    00:00
(1367/1530): qt5-qtbase-5.12.5-4.el8.x86_64.r 482 kB/s | 3.4 MB    00:07
(1368/1530): quota-4.04-10.el8.x86_64.rpm      268 kB/s | 212 kB    00:00
(1369/1530): quota-nls-4.04-10.el8.noarch.rpm   833 kB/s | 94 kB    00:00
(1370/1530): radvd-2.17-14.el8.x86_64.rpm       610 kB/s | 103 kB    00:00
(1371/1530): rasdaemon-0.6.1-3.el8.x86_64.rpm  327 kB/s | 104 kB    00:00
(1372/1530): rdma-core-26.0-8.el8.x86_64.rpm   617 kB/s | 62 kB    00:00
(1373/1530): realmd-0.16.3-18.el8.x86_64.rpm   559 kB/s | 233 kB    00:00
(1374/1530): rest-0.8.1-2.el8.x86_64.rpm        90 kB/s | 70 kB    00:00
(1375/1530): rng-tools-6.8-3.el8.x86_64.rpm    361 kB/s | 58 kB    00:00
(1376/1530): rootfiles-8.1-22.el8.noarch.rpm    350 kB/s | 13 kB    00:00
(1377/1530): rpcbind-1.2.5-7.el8.x86_64.rpm    432 kB/s | 69 kB    00:00
(1378/1530): rsync-3.1.3-7.el8.x86_64.rpm      691 kB/s | 403 kB    00:00
(1379/1530): rsyslog-8.1911.0-3.el8.x86_64.r  336 kB/s | 730 kB    00:02
    
```

```

(1530/1530): flite-1.3-31.el8.x86_64.rpm | 525 kB/s | 6.1 MB | 00:11
-----
Total | 1.4 MB/s | 1.1 GB | 14:17
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      : | 1/1
Complete!
The downloaded packages were saved in cache until the next successful transaction.
You can remove cached packages by executing 'dnf clean packages'.
=> Processing phase `InterimPreparation`
=> * efi_interim_fix
    Adjust EFI boot entry for first reboot
=> * initram_disk_generator
    Creates the upgrade initram disk

=> * efi_interim_fix
    Adjust EFI boot entry for first reboot
=> * initram_disk_generator
    Creates the upgrade initram disk

=> * add_upgrade_boot_entry
    Add new boot entry for Leapp provided initramfs.
A reboot is required to continue. Please reboot your system.

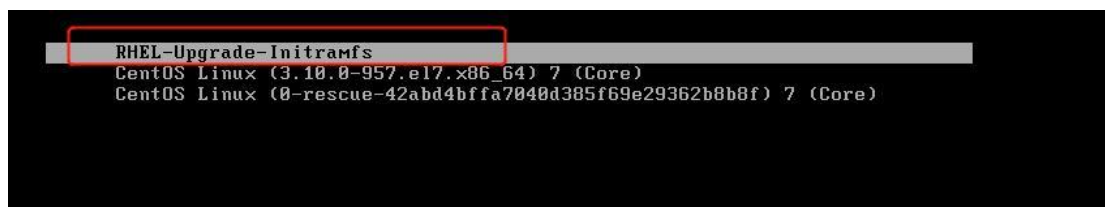
Debug output written to /var/log/leapp/leapp-upgrade.log

=====
                        REPORT
=====

A report has been generated at /var/log/leapp/leapp-report.json
A report has been generated at /var/log/leapp/leapp-report.txt

=====
                        END OF REPORT
=====
    
```

运行完成，执行# reboot 命令

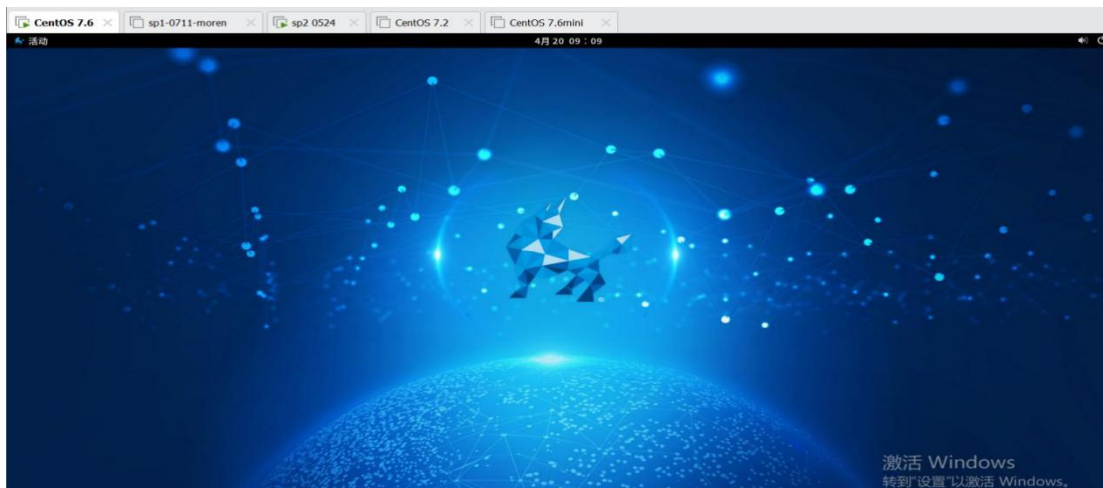


在这个界面需要等待

```
early console in extract_kernel
input_data: 0x00000000053893b1
input_len: 0x000000000066a55c
output: 0x0000000001000000
output_len: 0x000000000217caec
kernel_total_size: 0x0000000004a2c000
trampoline_32bit: 0x00000000009c000
booted via startup_32()
Physical KASLR using RDRAND RTSC...
Virtual KASLR using RDRAND RTSC...

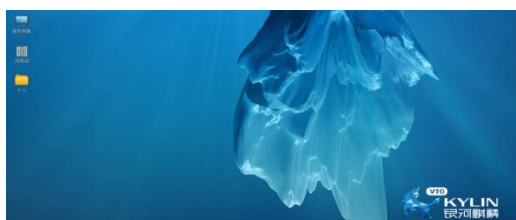
Decompressing Linux... Parsing ELF... Performing relocations... done.
Booting the kernel.
[ 1.220809] integrity: Unable to open file: /etc/keys/x509_ima.der (-2)
[ 1.220811] integrity: Unable to open file: /etc/keys/x509_evm.der (-2)
[ 3.377733] sd 2:0:0:0: [sda] Assuming drive cache: write through
[ 10.949173] selinux-autorelabel[786]: *** Warning -- SELinux targeted policy relabel is required.
[ 10.949653] selinux-autorelabel[786]: *** Relabeling could take a very long time, depending on file
[ 10.949942] selinux-autorelabel[786]: *** system size and speed of hard drives.
[ 33.688359] selinux-autorelabel[786]: Warning: Skipping the following R/O filesystems:
[ 33.688484] selinux-autorelabel[786]: /sys/fs/cgroup
[ 33.689019] selinux-autorelabel[786]: Relabeling / /boot /dev /dev/hugepages /dev/mqueue /dev/pts
/dev/shm /run /sys /sys/fs/cgroup/blkio /sys/fs/cgroup/cpu,cpuacct /sys/fs/cgroup/cpuset /sys/fs/cg
roup/devices /sys/fs/cgroup/freezer /sys/fs/cgroup/hugetlb /sys/fs/cgroup/memory /sys/fs/cgroup/net_
cls,net_prio /sys/fs/cgroup/perf_event /sys/fs/cgroup/pids /sys/fs/cgroup/rdma /sys/fs/cgroup/system
d /sys/fs/pstore /sys/kernel/debug
[ 95.615005] selinux-autorelabel[786]: Warning no default label for /dev/mqueue
[ 95.616348] selinux-autorelabel[786]: Cleaning up labels on /tmp
```

界面与系统内核信息均已升级为 kylin ns 8.2 系统



运行# kyreplace -change2ukui

完成后重启

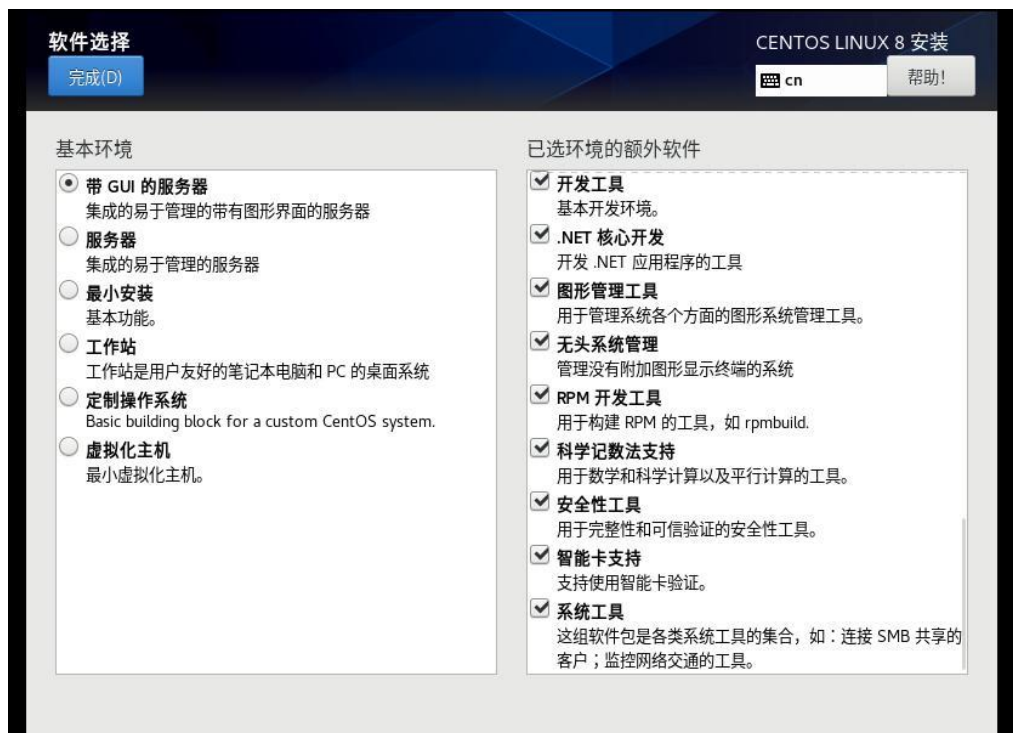


```
[root@localhost ~]# cat /proc/version
Linux version 4.19.91-24.8.el8.ks8.12.x86_64 (mockbuild@67df6a95a890499baba6108b
e189ba0c) (gcc version 8.3.1 20191121 (KyLin 8.3.1-5) (GCC)) #1 SMP Thu Nov 4 22
:16:58 CST 2021
```

9.2. CentOS 8.2 替换示例(目标系统麒麟 ns8.2)

iso:CentOS-8.2.2004-x86_64-dvd1.iso

安装方式:全部勾选安装



工具安装

```
[root@localhost ~]# rpm -ivh kyreplace-1.0-1.noarch.rpm
Verifying... ##### [100%]
准备中... ##### [100%]
正在升级/安装...
  1:kyreplace-1.0-1 ##### [100%]
[root@localhost ~]#
```

工具初始化

```
[root@localhost kyreplace]# pwd
/root/kyreplace
[root@localhost kyreplace]# kyreplace -init
*****
Initializing the environment for centos8
You can modify the repo file : /etc/kyreplace/centos2kylin/ass.repo
*****
Verifying... ##### [100%]
准备中... ##### [100%]
正在升级/安装...
  1:centos2kylin-1.0-1 ##### [100%]
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip3 install --user` instead.
Processing /etc/kyreplace/centos2kylin/modules/pexpect-4.8.0-py2.py3-none-any.whl
Processing /etc/kyreplace/centos2kylin/modules/ptyprocess-0.7.0-py2.py3-none-any.whl
Installing collected packages: ptyprocess, pexpect
Successfully installed pexpect-4.8.0 ptyprocess-0.7.0
[root@localhost kyreplace]#
```

迁移前评估

```
[root@localhost kyreplace]# kyreplace -check
```

查看评估报告

/var/tmp/kymigration/kyreplace/centos2kylin/assessment/assessment.html

是否可以迁移 **true**

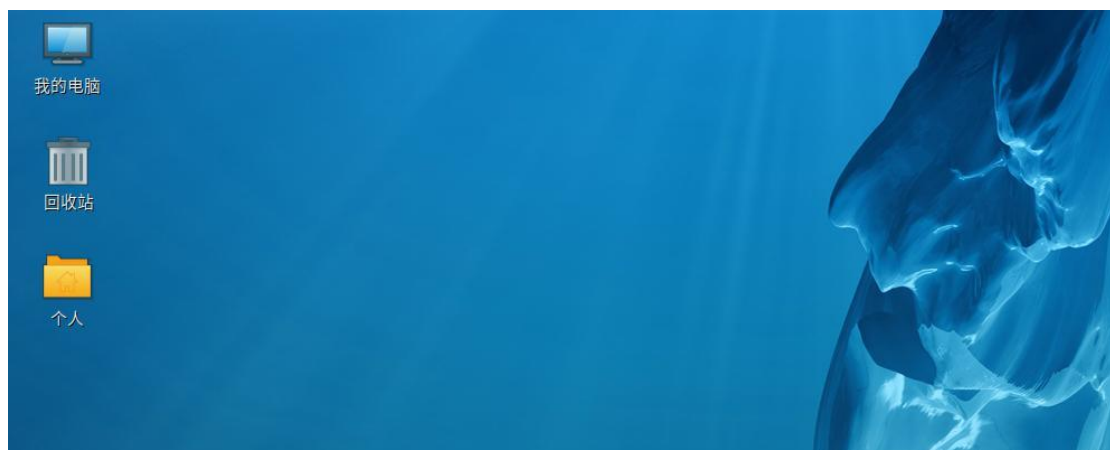
可以看到“是否可以迁移”为 true，则可以执行迁移。

执行迁移移完毕

```
[root@localhost kyreplace]# kyreplace
===== Start =====
Checking if the tool is executed by root user
Get current OS version.
Version is centos : 8.2
Checking the version of Kylin OS
Migration to Kylin OS 8.2
Checking required packages
Creating a list of RPMs installed before the migration.
Verifying RPMs installed before the migration against RPM database.
```

```
Complete!
Removing yum cache
Migrate successfully, update grub.cfg.
Generating grub configuration file ...
done
Creating a list of RPMs installed after the switch
Verifying RPMs installed after the switch against RPM database
Review the output of following files:
localhost.localdomain-rpms-list-after.log
localhost.localdomain-rpms-verified-after.log
0 files removed
Migrate complete. Kylin OS recommends rebooting this system. and check /etc/yum.repos.d/kylin_bak for old repo files.
The migration report path : /var/tmp/kymigration/kyreplace/centos2kylin/migration/migration.html
[root@localhost test]#
```

重启



稍等可进入麒麟的桌面。

迁移完毕后，可查看迁移报告

</var/tmp/kymigration/kyreplace/centos2kylin/migration/migration.html>。

9.3. CentOS7.6 替换示例(目标系统麒麟 ns7.6)

iso:CentOS-7-x86_64-DVD-1810.iso

安装方式:带 GUI 的服务器(不勾选右侧安装)

工具安装

```
[root@localhost test]# ls
kyreplace-1.0-1.noarch.rpm
[root@localhost test]# rpm -ivh kyreplace-1.0-1.noarch.rpm
准备中... ##### [100%]
正在升级/安装...
 1:kyreplace-1.0-1 ##### [100%]
```

若存在依赖问题，请使用 `yum install python3 -y` 安装 `python3`。

工具初始化

(1) 修改 `/usr/share/kyreplace/kyreplace.conf` 文件 将 `target_os = 8.2`

修改为 `target_os = 7.6`

(2) 命令行执行 `# kyreplace -init`

```
[root@localhost kyreplace]# kyreplace -init
*****
Initializing the environment...
*****
准备中... ##### [100%]
正在升级/安装...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 1863k 100 1863k 0 0 2427k 0 --:--:-- --:--:-- 2426k
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. pip
21.0 will drop support for Python 2.7 in January 2021. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/
development/release-process/#python-2-support pip 21.0 will remove support for this functionality.
Collecting pip<21.0
  Downloading pip-20.3.4-py2.py3-none-any.whl (1.5 MB)
    ##### 1.5 MB 1.4 MB/s
Collecting wheel
  Downloading wheel-0.37.1-py2.py3-none-any.whl (35 kB)
Installing collected packages: pip, wheel
Successfully installed pip-20.3.4 wheel-0.37.1
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. pip
21.0 will drop support for Python 2.7 in January 2021. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/
development/release-process/#python-2-support pip 21.0 will remove support for this functionality.
Processing /etc/centos2kylin/modules/pexpect-4.8.0-py2.py3-none-any.whl
Processing /etc/centos2kylin/modules/ptyprocess-0.7.0-py2.py3-none-any.whl
Installing collected packages: ptyprocess, pexpect
Successfully installed pexpect-4.8.0 ptyprocess-0.7.0
usage: centos2kylin.pyc [-h] [-V] [-c] [-check]

optional arguments:
-h, --help show this help message and exit
-V Verify RPM information before and after the migration.
-c Continue to migration after interruption.
-check Migration assessment.
```

迁移前评估

```
[root@localhost kyreplace]# kyreplace -check

The assessment report path : /var/tmp/kymigration/kyreplace/centos2kylin/assessment/assessment.html
[root@localhost kyreplace]#
```

查看评估报告

/var/tmp/kymigration/kyreplace/centos2kylin/assessment/assessment.h
tml

是否可以迁移 true

可以看到“是否可以迁移”为 true，则可以执行迁移。

执行迁移

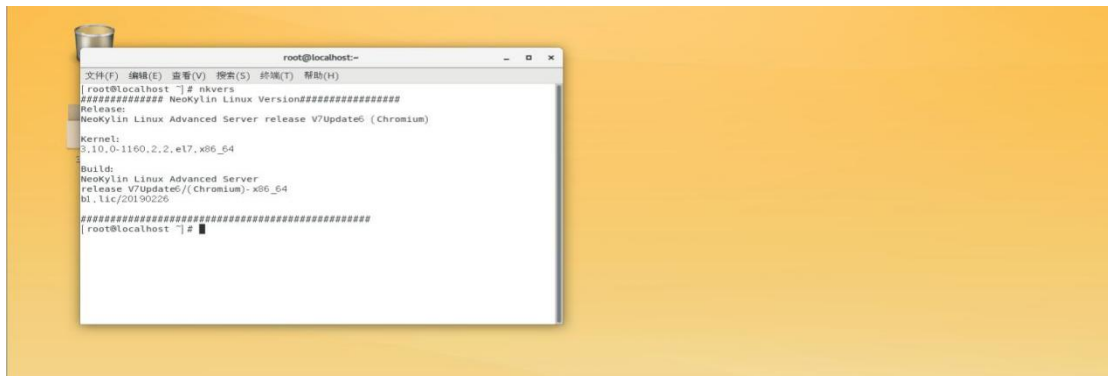
```
[root@localhost kyreplace]# kyreplace
===== Start =====
Checking if the tool is executed by root user
Get current OS version.
Version is centos : 7.6
Checking the version of Kylin OS
Migration to Kylin OS 7.6
```

迁移完毕

```
完毕!
Removing yum cache
Migrate successfully, update grub.cfg.
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.10.0-1160.2.2.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-1160.2.2.el7.x86_64.img
Found linux image: /boot/vmlinuz-3.10.0-957.el7.x86_64
Found initrd image: /boot/initramfs-3.10.0-957.el7.x86_64.img
Found linux image: /boot/vmlinuz-0-rescue-686d854f54474fcb89c99fa0e9089bf8
Found initrd image: /boot/initramfs-0-rescue-686d854f54474fcb89c99fa0e9089bf8.img
done
已加载插件: fastestmirror, langpacks
正在清理软件源: kylin_base kylin_updates
find: '/run/user/1000/gvfs': 权限不够
Migration report path : /etc/centos2kylin/output/html/migration.html
Migrate complete. Kylin OS recommends rebooting this system. and check /etc/yum.repos.d/kylin_bak for old repo files.
The migration report path : /var/tmp/kymigration/kyreplace/centos2kylin/migration/migration.html
```

重启# reboot

迁移后验证



还可查看迁移报告

[/var/tmp/kymigration/kyreplace/centos2kylin/migration/migration.html](#)。

10. 后续升级计划

计划 8 月底完成迁移平台化改造，实现批量替换，提供 web 可视化操作界面等功能，提升用户体验。

四. kycomp 使用指南

1. 使用方式

用户通过 web 访问方式，选择对比的操作系统，然后可以查看对比结果。

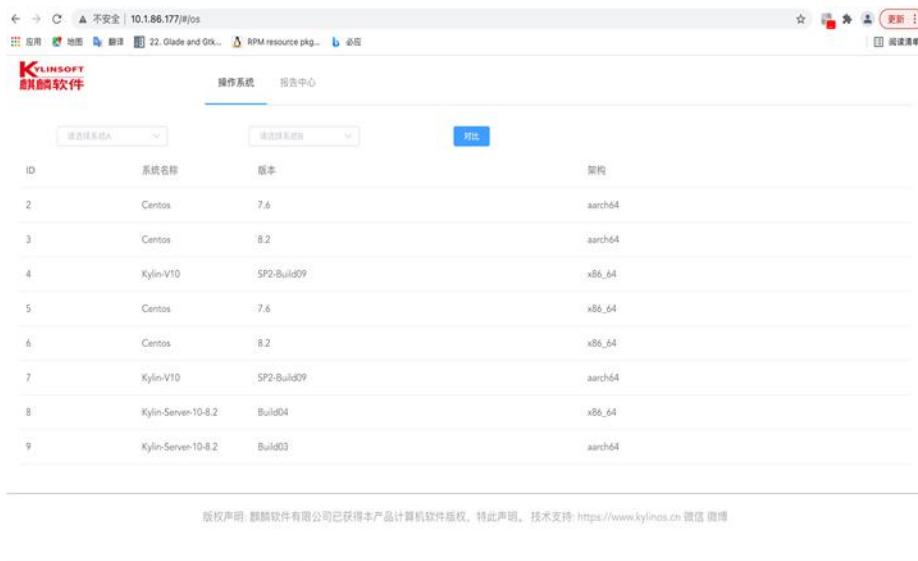
2. 工具的价值与效果

kycomp 操作系统差异对比分析工具能够收集系统命令、服务、函数及软件包变化差异，对迁移软件进行系统差异评估，对迁移软件中依赖的命令、服务、函数接口提供差异分析报告，以帮助迁移评估。

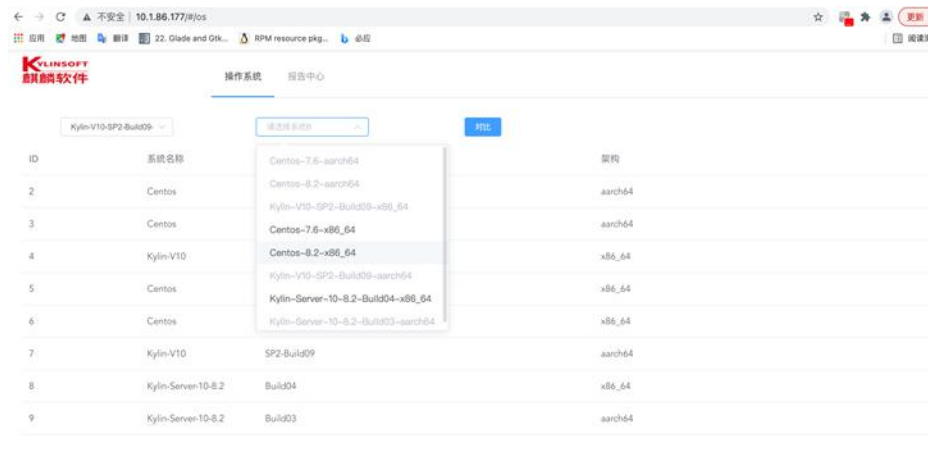
3. 获取方式

提供 web 服务访问。具体操作示例如下：

用浏览器打开地址：<http://部署 IP 地址>



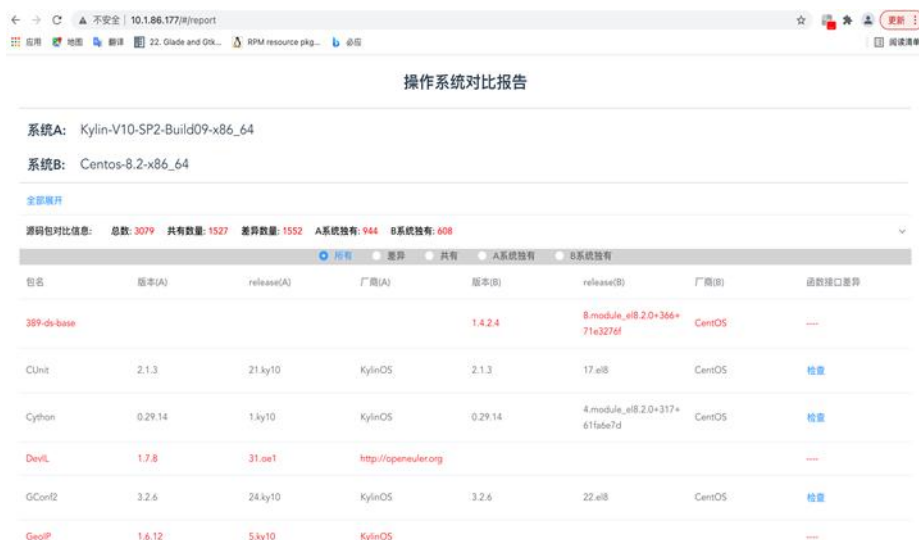
在页面上的两个下拉选框中选择要对比的两个系统,只能选相同架构的两个系统进行对比:



点击【对比】按钮后跳到报告中心页面,一共有源码包对比,RPM包对比,命令对比,服务对比和动态库对比五项,每一项都可以单击展开对比表格,表格可以分页显示:



在报告中心页面上可点击【全部展开】或单独点开某行的对比项,下图是点击源码包对比信息展开的表格:



操作系统对比报告

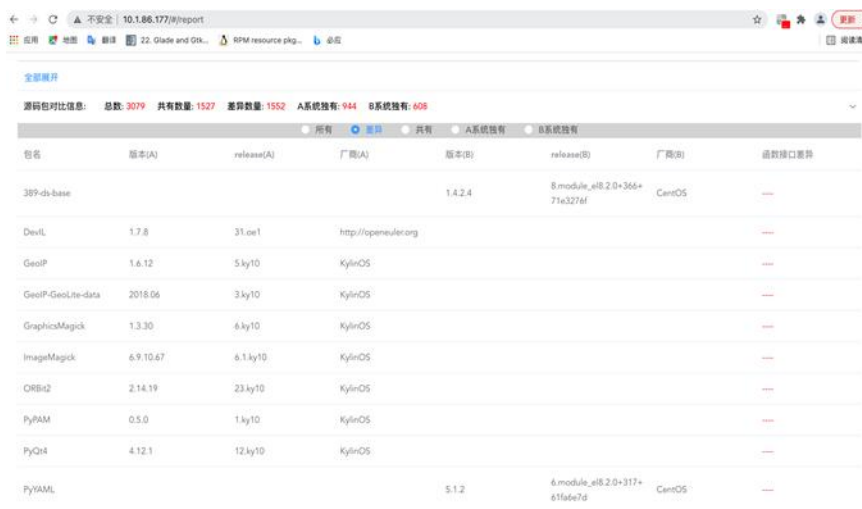
系统A: Kylin-V10-SP2-Build09-x86_64
系统B: Centos-8.2-x86_64

全部展开

源码包对比信息: 总数: 3079 共有数量: 1527 差异数量: 1552 A系统独有: 944 B系统独有: 608

包名	版本(A)	release(A)	厂商(A)	版本(B)	release(B)	厂商(B)	函数接口差异
389-ds-base				1.4.2.4	8.module_el8.2.0+366+71e3276f	CentOS	---
CUnit	2.1.3	21.ky10	KylinOS	2.1.3	17.el8	CentOS	检查
Cython	0.29.14	1.ky10	KylinOS	0.29.14	4.module_el8.2.0+317+61fa6e7d	CentOS	检查
DevIL	1.7.8	31.oel1	http://openeye.org				---
GConf2	3.2.6	24.ky10	KylinOS	3.2.6	22.el8	CentOS	检查
GeoIP	1.6.12	5.ky10	KylinOS				---

在具体展开的对比信息表中,可点击单选框来展示不同的数据,有"所有", "差异", "共有", "A系统独有", "B系统独有" 共5个单选框,下图是点击"差异"展开的表格:



全部展开

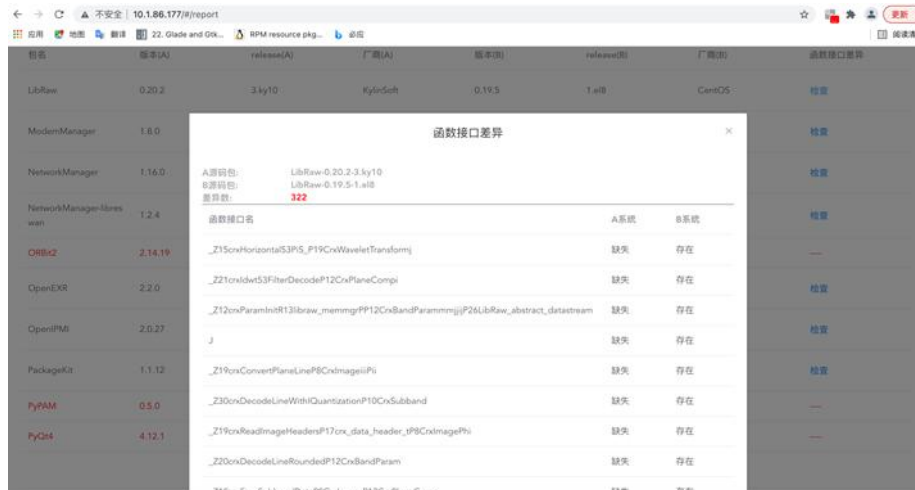
源码包对比信息: 总数: 3079 共有数量: 1527 差异数量: 1552 A系统独有: 944 B系统独有: 608

所有 差异 共有 A系统独有 B系统独有

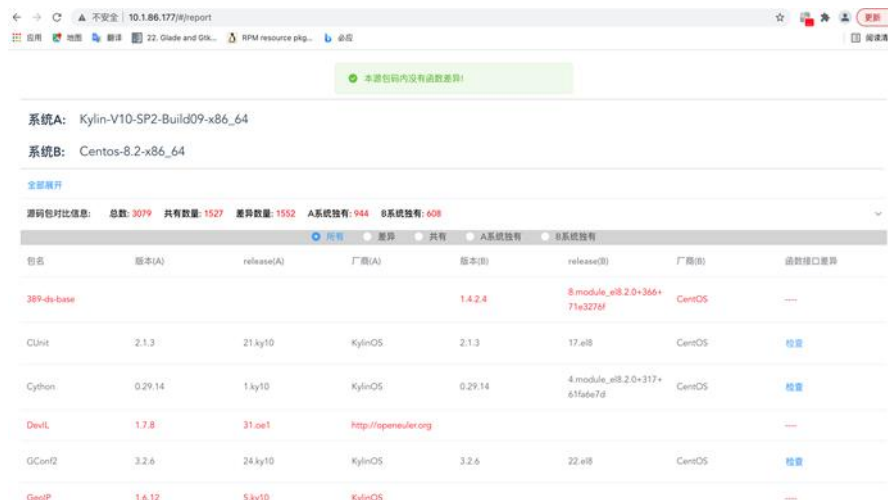
包名	版本(A)	release(A)	厂商(A)	版本(B)	release(B)	厂商(B)	函数接口差异
389-ds-base				1.4.2.4	8.module_el8.2.0+366+71e3276f	CentOS	---
DevIL	1.7.8	31.oel1	http://openeye.org				---
GeoIP	1.6.12	5.ky10	KylinOS				---
GeoIP-GeoLite-data	2018.06	3.ky10	KylinOS				---
GraphicsMagick	1.3.30	6.ky10	KylinOS				---
ImageMagick	6.9.10.67	6.1.ky10	KylinOS				---
ORBit2	2.14.19	23.ky10	KylinOS				---
PyBAM	0.5.0	1.ky10	KylinOS				---
PyQt4	4.12.1	12.ky10	KylinOS				---
PyYAML				5.1.2	6.module_el8.2.0+317+61fa6e7d	CentOS	---

在选中"所有"之后,表格中的红色行表示某个源码包只在一个系统中存在,表头中的A和B分别代表两个系统。

点击"检查"可以展示一个源码包在两个系统中的函数接口差异:



如果没有函数接口差异就不会展示对话框，会提示 "本源码包内没有函数差异":



其它对比项的表格和源码包对比表格采用的同样的布局和同样的操作方式。

五. Kylinch 使用指南

此文档主要对服务器整机认证测试套件 kylinch 的安装、使用以及测试用例进行了介绍。

1. 测试环境

硬件环境：x86_64、aarch64、mips64el

软件环境：Kylin Linux Advanced Server V10(SP1)

2. 测试要求

系统安装要求：

客户端和服务端安装 kylin 操作系统，安装时“软件选择”请选择“带 UKUI GUI 的服务器”，并勾选“已选环境的附加项”下的全部内容。

除了操作系统安装的硬盘之外，被测服务器还需要一颗额外的磁盘，用于进行磁盘的格式化以及读写测试。比如 Kylin 系统安装在 sda 上，那么被测服务器需要额外一颗 sdb 的盘进行硬盘相关测试。

其他要求：

最好保证测试系统有网络连接。

如设备不自带光驱，需要外接光驱测试 cdrom 项。

3. 使用步骤说明

系统安装完毕后，请根据以下步骤则可以完成安装测试工具 kylinch。

kylinch 执行测试主要包括三个步骤：

根据硬件情况自动列出需要执行的测试用例

执行测试

结果输出

工具安装以及测试步骤说明参见下述章节。

3.1. 客户端安装

需要进行认证测试的机器作为客户端，可以使用脚本自动安装客户端相关服务，或者通过手动安装的方式实现。

使用脚本安装会将需要使用的依赖包全部安装好并做好启动服务等操作，手动安装则需要自行安装工具包，以及开启服务等。

推荐使用脚本安装，如果被测服务器没有网络连接，只能进行离线测试，请直接使用脚本安装。

注意：如果加载用例集中包括性能测试集和稳定性测试集，需要手动将性能测试工具包解压到/opt/performance_server。

3.1.1. 手动安装

手动安装的操作步骤如下：

将 kylinch 的压缩包拷贝到测试机上并解压

安装 kylinch 客户端的 rpm 包

```
# rpm -ivh kylin-ch-<version>-<date>.<os>.<arch>.rpm
```

```
# rpm -ivh kylin-ch-testsuites-<version>-<date>.<os>.noarch.rpm
```

关闭防火墙和 selinux

```
# systemctl stop firewalld
```

```
# iptables -F
```

```
# setenforce 0
```

在对应架构的文件夹下，找到 fio、netperf 和 memtester 测试工具并安装

安装完成后，在终端输入 'kylinch'，如果可以正常运行，则表示安装成功。

如果加载用例集中包括性能测试集（如图 1）或整机压力测试集，需要手动将性能测试工具包解压到/opt/performance_server（目录结构见图 2）

需要先进行授权认证才会加载测试集，详情见 3.3 工具授权认证

Name	Status	Estimate	Device	Automated
[X] + 整机认证测试集				
[X] + 性能测试集(性能测试耗时较长, 请不要和其他测试集一起测试)				

图 1-加载测试集界面

```

[root@localhost performance_server]# pwd
/opt/performance_server
[root@localhost performance_server]# ll
总用量 292
-rwxrwxr-x 1 root root 2376 2月 22 2021 com_syslog.h
drwxr-xr-x 8 root root 88 8月 26 15:07 config
-rwxrwxrwx 1 root root 38056 8月 26 15:09 fio.sh.x
-rwxrwxrwx 1 root root 29240 8月 26 15:09 iozone.sh.x
-rwxrwxrwx 1 root root 28872 8月 26 15:09 lmbench.sh.x
-rwxrwxr-x 1 root root 70 2月 22 2021 netperf.h
-rwxrwxrwx 1 root root 38088 8月 26 15:09 netperf.sh.x
-rwxrwxr-x 1 root root 1779 4月 20 14:11 netserver.sh
drwxr-xr-x 5 root root 47 8月 26 15:07 script
-rwxrwxr-x 1 root root 124 2月 22 2021 speccpu.h
-rwxrwxrwx 1 root root 38472 8月 26 15:09 speccpu.sh.x
-rwxrwxrwx 1 root root 25512 8月 26 15:09 specjvm.sh.x
-rwx--x--x 1 root root 26168 8月 30 16:37 stream.sh.x
drwxr-xr-x 12 root root 284 8月 28 12:07 tools
-rwxrwxrwx 1 root root 30696 8月 26 15:09 Unixbench.sh.x
-rw-rw-r-- 1 root root 34 8月 25 17:11 version.txt
[root@localhost performance_server]#
    
```

图 2-performance_server 目录结构图

3.1.2. 脚本安装

脚本安装操作步骤如下：

将 kylinch 的压缩包拷贝到测试机上并解压

执行安装脚本进行工具安装

```
# bash install.sh -client
```

安装完毕显示“PASS”证明安装通过，若未显示“PASS”，屏幕会打印未安装的依赖包，可手动安装依赖包。

```

.....安装fio工具.....
fio已安装
[INFO]-----安装netperf工具-----
netperf已安装
-----安装memtester工具-----
memtester已安装
setenforce: SELinux is disabled
PASS
    
```

图 3-client 端安装 pass

```

netperf已安装
-----安装mentester工具-----
mentester已安装
setenforce: SELinux is disabled
----- error summary -----
stress has not been installed
    
```

图 4-client 端安装 fail

如果加载用例集中包括性能测试集（如上图 1）或整机压力测试集，需要手动将性能测试工具包解压到/opt/performance_server（目录结构见上图 2）

3.2. 服务端安装

服务端可以安装在与被测机器网络连通的任意服务器上，多台客户端可以共用同一服务端，但是连接同一服务端的多台客户端不能同时进行网络测试。

服务端可以通过脚本自动安装或者手动安装，使用脚本安装会将需要使用的依赖包全部安装好并做好启动服务等操作，手动安装则需要自行安装工具包以及开启服务等。

推荐使用脚本安装，如果服务端所在的服务器没有网络连接，只能进行离线测试，请直接使用脚本安装。

3.2.1. 手动安装

手动安装的操作步骤如下：

将 kylinch 的压缩包拷贝到测试机上并解压

安装 kylinch 服务端的 rpm 包

```
# rpm -ivh kylin-ch-server-<version>-<date>.<os>.<arch>.rpm
```

关闭防火墙和 selinux

```
# systemctl stop firewalld
```

```
# iptables -F
```

```
# setenforce 0
```

在对应架构的文件夹下，找到 netperf 测试工具并安装，用于为客户端进行网络测试时提供服务

服务端 web 功能需要安装如下依赖包

```
# pip3 install Flask Flask-bootstrap uwsgi
```

如果下载速度慢，可以配置国内 pip 源进行安装，比如：

```
# pip3 install Flask Flask-bootstrap uwsgi -i  
http://mirrors.aliyun.com/pypi/simple/ --trusted-host mirrors.aliyun.com
```

服务端默认使用 8080 端口，同时搭配 nginx（默认端口 80）提供 web 服务，请保证这些端口未被占用，之后通过如下命令启动服务：

```
# systemctl start kylinch-server.service
```

```
# systemctl start nginx.service
```

3.2.2. 脚本安装

脚本安装操作步骤如下：

将 kylinch 的压缩包拷贝到测试机上并解压

执行安装脚本进行工具安装

```
# bash install.sh -server
```

安装完毕显示“PASS”证明安装通过，若未显示“PASS”，屏幕会打印未安装的依赖包，可手动安装依赖包。

```
1:kylin-ch-server-1.0.0-20210805.ky##### [100%]
setenforce: SELinux is disabled
[INFO] kylinch-server 服务开启成功
[INFO] nginx 服务开启成功
PASS
```

图 5-server 端安装 pass

```
Failed to start nginx.service: Unit nginx.service not found.
[ERROR] nginx 服务开启失败
----- error summary -----
nginx has not been installed
python3-cio has not been installed
python3-flask has not been installed
Flask-bootstrap has not been installed
uwsgi has not been installed
nginx 服务开启失败
```

图 6-server 端安装 fail

3.3. 工具授权认证

测试前需要先认证工具才能进行测试。

在测试机上首次安装 kylinch，执行完以上的安装步骤之后，终端执行命令 kylinch 有如下图显示，表示需要 key 文件来进行授权认证，认证之后才能使用工具。

注意：key 文件有时间期限，到期后请重新获取 key 文件认证。

```
[root@localhost ~]# kylinch
[ERROR] 工具未认证，请先认证后再执行工具。
[root@localhost ~]#
```

图 7-kylinch 授权认证

kylinch --auth KylinchAuthorization.key 文件的路径

```
[root@localhost ~]# kylinch --auth /root/KylinchAuthorization.key
[INFO] 认证成功。
[root@localhost ~]#
```

图 8-kylinch 授权认证成功

注意事项：

1) 认证的 key 需要和工具匹配，否则报错

```
[root@localhost License]# kylinch --auth KylinchAuthorization.key
[ERROR] 当前工具版本为 1.0.1, 授权版本是 1.0.0, 版本不匹配, 请重新认证.
[INFO] 认证成功.
[root@localhost License]#
```

图 9-kylinch 工具版本与 key 不匹配

2) 如果出现如下提示工具已过期，请联系工具提供方重新获取 key

```
[root@localhost License]# kylinch
[ERROR] 您的工具认证已过期, 请重新认证.
[root@localhost License]# kylinch --auth KylinchAuthorization.key
[ERROR] 您的工具认证已过期, 请重新认证.
[ERROR] 认证失败.
```

图 10-kylinch 过期

3.4. 测试准备

3.4.1. 服务端测试准备

执行如下命令检查 netperf 是否正常安装，如果可以打印工具版本，则代表安装成功。

```
# netperf -V
```

```
[root@localhost ~]# netperf -V
Netperf version 2.7.0
```

图 11-netperf 查看版本

执行如下命令检查 kylinch-server 服务是否正常启动

```
# systemctl status kylinch-server.service
```

```
[root@localhost ~]# systemctl status kylinch-server.service
● kylinch-server.service - Kylin Hardware Compatibility Test Server
   Loaded: loaded (/usr/lib/systemd/system/kylinch-server.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2021-08-06 17:41:30 CST; 20h ago
     Process: 165000 ExecStartPre=/usr/share/kylinch/lib/server/kylinch-server-pre.sh (code=exited, status=0/SUCCESS)
    Main PID: 165002 (uwsgi)
      Status: "uWSGI is ready"
        Tasks: 4
       Memory: 52.8M
      CGroup: /system.slice/kylinch-server.service
             └─165002 /usr/local/bin/uwsgi --ini /usr/share/kylinch/lib/server/uwsgi.ini
             └─165005 /usr/local/bin/uwsgi --ini /usr/share/kylinch/lib/server/uwsgi.ini
             └─165006 /usr/local/bin/uwsgi --ini /usr/share/kylinch/lib/server/uwsgi.ini
             └─165007 /usr/local/bin/uwsgi --ini /usr/share/kylinch/lib/server/uwsgi.ini
```

图 12-查看 kylinch-server 服务状态

3.4.2. 客户端测试准备

执行如下命令检查 netperf、fio、memtester 是否正常安装，如果可以打印工具版本，则代表安装成功。

```
# netperf -V
```

```
[root@localhost ~]# netperf -V
Netperf version 2.7.0
```

图 13-查看 netperf 版本

```
# fio -V
```

```
[root@localhost ~]# fio -v
fio-3.20
```

图 14-查看 fio 版本

```
# memtester
```

```
[root@localhost ~]# memtester
memtester version 4.5.1 (64-bit)
Copyright (C) 2001-2020 Charles Cazabon.
Licensed under the GNU General Public License version 2 (only).

pagesize is 4096
pagesizemask is 0xffffffffffff000
need memory argument, in MB

Usage: memtester [-p physaddrbase [-d device]] <mem>[B|K|M|G] [loops]
```

图 15-查看 memtester 版本

kylinch 执行测试用例之前会检查所有待测项的依赖包是否安装，如果没有安装，会使用 yum 安装依赖包。如果有依赖包未安装成功导致退出测试，请手

动安装缺失的依赖包。

缺少依赖包退出测试如下图：

```

Selection (<number>|all|none|quit|run): r
Installing required packages: stress
Error: There are no enabled repositories in "/etc/yum.repos.d", "/etc/yum/repos.d", "/etc/distro.repos.d".
"yum install -y stress" None
Fail to install required packages.
Required rpm package not installed, test stopped.
----- Summary -----
core                                FAIL
system                              FAIL
    
```

图 16-kylinch 缺少依赖包退出

正式测试之前，针对不同测试项，请参见“表 1 测试用例前置条件”进行准备工作。

注意：性能测试集需要联网才能下载安装依赖包。

表 1 测试用例前置条件

测试项	测试模式	测试时间	前置条件
整机认证测试集			
acpi	自动	1min	被测机器支持 acpi
boottime	自动	1min	无
cdrom	交互	20min	1.如无内置光驱需要外接光驱测试 2.准备一张 CD 或 DVD 进行测试
core	自动	15min	无
cpufreq	自动	20min	1.存在 /sys/deivces/system/cpu/cpu0/cpufreq 文件 2. CPU 支持变频
disk	交互	100min*被 测分区数	至少存在一个可用分区，需要满足如下条件：

			1. 分区不能是 lvm (逻辑分区) 2. 分区不能被挂载
filesystem	交互	5min	至少存在可用分区，需要满足如下条件： 分区大小大于 2G 分区不能是 lvm (逻辑分区) 分区不能被挂载
hardwareinfo	自动	10min	无
ipmi	自动	2min	被测机器支持 ipmi
kdump	交互	15min	被测机器支持 kdump
logicalvolume	交互	5min	至少存在可用分区，需要满足如下条件： 分区大小大于 2G 分区不能是 lvm (逻辑分区) 分区不能被挂载
memory	自动	40min	无
mobileharddisk	交互	15min	准备移动硬盘
mobilehdhotplug	交互	10min	准备移动硬盘
network	自动	20min	服务端 kylinch-server 服务开启 客户端和服务端关闭防火墙 客户端与服务端直连或通过交换机连接

			客户端与服务端已安装 netperf 各网口配置好连通服务端的 IP
nvme	自动	20min	无
perf	自动	5min	无
system	自动	1min	无
udiskhotplug	交互	5min	准备 U 盘
usb	交互	10min	准备待测 usb 设备，比如 U 盘
usb_keyboard	交互	2min	被测机器需要插入键盘(此测试必须在图形化环境进行)
usb_mouse	交互	2min	被测机器需要插入鼠标(此测试必须在图形化环境进行)
usb_udisk	交互	15min	准备 U 盘
usbcdriverhotplug	交互	10min	准备 usb 光驱设备
video	交互	10min	被测机器需要连接显示器(此测试必须在图形化环境进行)
watchdog	交互	10min	无
性能测试集			
strem	自动	10min	准备/opt/performance_server
fio	自动	80min	准备/opt/performance_server
unixbench	自动	90min	准备/opt/performance_server
iozone	自动	360min	准备/opt/performance_server
lmbench	自动	900min	准备/opt/performance_server

specjvm	自动	600min	准备/opt/performance_server
speccpu	自动	2880min	准备/opt/performance_server
netperf	交互	15min	准备/opt/performance_server
性能测试集			
ltp	自动	48h	准备/opt/performance_server
rebootstress	交互	8h	无

3.5. 测试执行

请参照以下步骤开始执行测试：

客户端执行“kylinch”，首次输入需要按照提示信息，依次输入 Test ID，Product URL，Test Server IP。如果输入错误需要修改，请在终端执行“kylinch --clean”清除以下信息（注意：如果已测试了部分用例，请先打包再 clean，避免测试结果被清除）

Test ID：测试任务 ID，用于在服务端 web 查看在线报告

Product URL：被测产品网站地址，可不填

Test Server：kylinch 服务端 IP，服务端与被测机器的 IP 必须是连通的

```
[root@localhost ~]# kylinch
The Kylin Hardware Compatibility Test Suite
Please provide your Compatibility Test ID:0804
Please provide your Product URL:www.kylinos.cn
Please provide the Compatibility Test Server (Hostname or Ipaddr):192.66.1.28
Compatibility Test ID:      0804
Hardware Info:              Suma H520-C30 Dhyana
Product URL:                www.kylinos.cn
OS Info:                    Kylin Linux Advanced Server V10 (Tercel)
Kernel Info:                4.19.90-23.8.v2101.ky10.x86_64
Test Server:                192.66.1.28
```

图 17-首次运行 kylinch

进入测试用例选择与执行界面，在此界面，框架将自动扫描硬件并选取当前环境可供测试的测试项，每列表示意义如下：

第一列：是否选中

第二列：测试项名称

第三列：测试状态。NotRun、Fail (红色)、Pass (绿色)、Force (必测项 -黄色)

第四列：预计测试时间 (仅供参考)

第五列：设备 (与硬件设备相关的测试项会显示)

第六列：是否为自动项

Name	Status	Estimate	Device	Automated
[X] - 整机认证测试集				
[X] system	Force	1min		yes
[X] acpi	NotRun	1min		yes
[X] core	NotRun	12min		yes
[X] cpufreq	NotRun	20min		yes
[X] ethernet	NotRun	20min	enp5s0f0	yes
[X] ethernet	NotRun	20min	enp5s0f1	yes
[X] ethernet	NotRun	20min	enp5s0f2	yes
[X] ethernet	NotRun	20min	enp5s0f3	yes
[X] infiniband	NotRun	20min		yes
[X] ipmi	NotRun	15min		yes
[X] memory	NotRun	40min		yes
[X] nvme	NotRun	20min		yes
[X] perf	NotRun	5min		yes
[X] video	NotRun	10min	0000:02:00.0	yes
[X] cdrom	NotRun	20min		no
[X] disk	NotRun	100min		no
[X] filesystem	NotRun	5min		no
[X] logicalvolume	NotRun	5min		no
[X] usb	NotRun	10min		no
[X] usb_keyboard	NotRun	2min		no
[X] usb_mouse	NotRun	2min		no
[X] kdump	NotRun	15min		no
[X] watchdog	NotRun	10min		no
[] + 性能测试集 (性能测试耗时较长，请不要和其他测试集一起测试)				

图 18-kylinch 测试用例执行界面

根据需要选择测试项来执行测试，按键说明如下：

Enter/+/-：展开/收起测试集

空格：选中/取消选中 (测试项开头的[X]代表选中，[]代表未选中

s/S：全选

d/D：全不选

q/Q：退出测试

r/R：开始测试

m/M：打开测试项详情界面（Esc 退出）

h/H：显示帮助信息（Esc 退出）

选择完测试项后，输入“R”，开始执行测试。部分测试项需要进行人工交互，请按测试项提示信息进行操作。

测试执行完毕后，会提示是否需要收集日志并打包，如果有测试项失败，需要重测，可以在提示打包时选择“n”。

```
----- Summary -----
acpi                PASS
system             PASS

您有可以打包的测试日志，是否要收集日志并打包(请在全部待测项都执行完后再进行此操作)? (y|n) y
Log saved to /var/kylinch_results/kylinch-20210719184125-hvT10cNiaj success.
```

图 19-kylinch 提示是否收集日志并打包

如果对测试结果进行了打包，打包完毕后，会提示是否需要上传测试结果到服务端。输入“y”可以上传测试结果到服务端，在线查看测试结果和详细日志。如果上传失败，请检查网络配置，然后离线上传测试结果。

```
Log saved to /var/kylinch_results/kylinch-20210719184125-hvT10cNiaj success.
是否提交最后一次打包的测试结果到server端? (y|n) y
Uploading...
Successfully uploaded result to server 172.17.30.102.
```

图 20-kylinch 上传测试结果到 server 端

3.6. 测试结果

3.6.1. 客户端测试结果查看

客户端的日志存放路径如下：

未打包日志存放目录：`/usr/share/kylinch/logs/<jobid>`，如下图所示：

某一项 log 存放在以测试项命名的.log 文件中，`job.log` 存放所有的 log。

```
[root@localhost ~]# ll /usr/share/kylinch/logs/t5UYg6yum1/
总用量 2488
-rw-r--r-- 1 root root 596714 8月 7 15:00 acpi.log
-rw-r--r-- 1 root root 245 8月 7 15:11 compatibility.json
-rw-r--r-- 1 root root 1042477 8月 7 15:11 device.json
-rw-r--r-- 1 root root 5418 8月 7 15:11 factory.json
-rw-r--r-- 1 root root 16305 8月 7 15:11 filesystem.log
-rw-r--r-- 1 root root 743783 8月 7 15:11 job.log
-rw-r--r-- 1 root root 130503 8月 7 15:11 system.log
[root@localhost ~]#
```

图 21-客户端未打包日志展示

已打包日志存放目录：`/var/kylinch_results/`（打包完成后有日志存放路径打印见图 19）可以看到有同名的文件夹和压缩包，文件夹用于本地查看 log，压缩包用于离线上传以及认证测试提供日志。

```
[root@localhost ~]# ls /var/kylinch_results/
kylinch-20210722090045-ftBhXEZCrs  kylinch-20210804172113-lLhRoqfTvp  kylinch-20210804172403-QMP8BfiIyR
kylinch-20210722090045-ftBhXEZCrs.zip  kylinch-20210804172113-lLhRoqfTvp.zip  kylinch-20210804172403-QMP8BfiIyR.zip
[root@localhost ~]# ls /var/kylinch_results/kylinch-20210804172113-lLhRoqfTvp/
compatibility.json  cpufreq.log  device.json  factory.json  job.log  system.log
[root@localhost ~]#
```

图 22-客户端已打包日志展示

3.6.2. 服务端测试结果查看

kylinch 的服务端除了与客户端进行网络测试外，还提供 web 功能，用于在线展示上传到服务端的测试结果，并提供测试结果的下载、离线上传等功能。

可以参照如下步骤在线查看测试结果：

浏览器打开服务端 IP 地址，在“测试结果”界面，点击被测机器信息，找到对应的测试 ID，进入测试结果查看页面。测试 ID 即客户端执行测试时输入的 Test ID。

Kylin整机适配测试结果

Lemote-LM-LS3B4000-7A1000-2w-V01-server_A1906-Not-Specified ▾

Greatwall-QingTian-DF720-Undefined ▾

图 23-web 页面首页

在测试结果查看页面,可以查看环境信息、执行结果、运行日志等详细信息。

Kylin整机适配测试报告

测试机	Greatwall-QingTian-DF720-Undefined
ID	ft200-0721
Job ID	kylinch-20210722090638-CSO0YdHaL5

[基本信息](#)
[硬件信息](#)
[测试日志](#)
[下载附件](#)

测试机环境信息

Manufacturer	Greatwall
Product Name	QingTian DF720
Version	Undefined
OS	Kylin Linux Advanced Server V10 (Terce)
kernel	4.19.90-23.8.v2101.ky10.aarch64
ID	ft200-0721
Product URL	dd
server	172.17.30.101

测试用例	测试设备	测试结果
acpi		NotRun
core		NotRun
cpufreq		NotRun
disk		NotRun
ethernet-enp11s0f0	enp11s0f0	NotRun
ethernet-enp11s0f1	enp11s0f1	NotRun
filesystem		NotRun
ipmi		NotRun
kdump		PASS
logicalvolume		NotRun
memory		FAIL
perf		NotRun
system		FAIL
usb		NotRun
usb_keyboard		NotRun
usb_mouse		NotRun
video-0000:08:00.0	0000:08:00.0	NotRun
watchdog		PASS

图 24-web 测试结果查看页面

点击“测试设备”列可查看详细设备信息，点击“PASS”或“FAIL”可以查看详细测试日志

硬件信息：查看机器设备细腻些

测试日志：查看测试运行日志

下载附件：下载测试日志。注意不要更改下载附件的 zip 包名称

离线上传

如果客户端在测试完成后由于网络或 server 端服务未开启等原因导致上传测试结果失败，可以使用离线上传功能，将客户端测试日志压缩包上传到服务端并查看。

4. 提交测试日志

提交认证测试需要所有测试项测试 pass，不支持的测试项做出说明。

将所有测试项通过的测试结果日志压缩包（zip），提交认证。（可以是多份 zip 包）

可以通过服务端 web 页面下载或从客户端目录拷贝获取测试结果，客户端测试结果存放在/var/kylinch_results/目录下。

5. 测试项说明

acpi

测试范围：测试 acpidump

测试流程：

- (1) 自动列出测试计划；
- (2) 通过查找 “/sys/firmware/acpi/tables” 判断机器是否支持 acpi；
- (3) 执行命令 “acpidump”。

测试结果：

结果失败：机器不支持 acpi 或 acpidump 命令执行失败。

结果通过：测试过程不出错。

测试时间：1min

boottime

测试范围：抓取自 kernel 开始的启动时间

测试流程：

- (1) 执行命令：systemd-analyze；
- (2) 筛选启动时间。

测试结果：

结果失败：命令执行失败。

结果通过：命令执行通过。

测试时间：1min

cdrom

测试范围：

测试设备自带或外接光驱设备。包括：

- (1) 只读光盘：CD-ROM DVD-ROM

(2) 刻录光盘：刻录 CD-RW 光盘的 CD 刻录光驱，能刻录 DVD-RW、DVD+RW、DVD+/-RW (DVD-Dual)、DVD-RAM、DVD-Super Multi、CD-RW 光盘中的一种或几种或所有的 DVD 刻录光驱。

(3) 蓝光光盘：BluRay

测试流程：

注意：测试过程中需要用户交互（根据提示放入光盘，以及关闭光盘仓）

(1) 列出测试计划；

(2) 打开光驱，放入光盘；

(3) 根据提示放入对应的光盘。若是读写测试，放入可读写光盘，测试格式化，写测试，读测试；若是只读测试，放入有数据的只读光盘进行读测试，放入没有数据的只读光盘进行写测试和读测试；

(4) 读写测试方法如下：

① 读测试：将光盘数据拷贝到特定目录并与光盘数据进行对比，检查数据一致性。

② 写测试：生成 iso 文件数据并写入光盘，写入完成与原始数据对比，检查数据一致性。利用（1）来进行读测试。

③ 读写测试：格式化光盘，利用（2）进行写测试。

测试结果：

结果失败：测试流程中任何子测试项出错则会不能正常进行，最终测试结果则为失败。

结果通过：所有子测试项通过。

测试时间：20min

core

测试范围：

测试 CPU 时钟准确程度，检测系统 CPU 并且确保 CPU 在大负载下能正常工作。

测试流程：

(1) 时钟准确度测试

① RTC 硬件时钟基本稳定性测试

获取硬件时钟时间 `rtc_tm1`，睡眠 `sleeptime=120s`，再次获取硬件时钟时间 `rtc_tm2`，验证 $\text{delta}=\text{rtc_tm2}-\text{rtc_tm1}-\text{sleeptime}$ 。

结果失败：delta 不等于 0

结果通过: delta 等于 0

② 时钟抖动测试

抖动是系统时钟之间彼此不同步时出现的一种现象。循环测试 10000 次测试，设定平均抖动时间 `max_ave_jitter`，最大抖动时 `max_adj_jitter` 为 0.2s，每次测试循环计算机器内所有运行 CPU，测试每个 CPU 之间的纳秒级时间差，统计最大时间差为 `maxdelta`，平均时间差为 `ave_jitter`。

结果失败： $\text{ave_jitter} > 0.2\text{s}$ ， $\text{maxdelta} > 0.2\text{s}$

结果通过:10000 次循环测试都正确

③ 始终方向测试

记录 `starttime` 然后睡眠 `sleeptime=60s` 后记录 `stoptime`， $\text{delta}=\text{stoptime}-\text{starttime}-\text{sleeptime}$ 。

结果失败：delta 不等于 0

结果通过：delta 等于 0

(2) CPU 负载测试

使用 stress 工具进行 10 分钟压力测试。

结果失败：stress 命令执行失败

结果通过：stress 命令执行成功

测试结果：

结果失败：有测试项目执行失败

结果通过：所有测试项目执行成功

测试时间：15min

cpufreq

测试范围：

主要测试系统的变频功能，包括 userspace、powersave、performance、ondemand、conservative 模式。

测试流程：

(1) 获取 CPU 信息并打印，包括 CPU 型号、CPU 支持最大频率，CPU 支持最小频率，CPU 数量，按拓扑划分的 package，CPU 支持的 governor，CPU flags，CPU 驱动，当前 governor，是否支持 aperf。

(2) 按照 package 分组 CPU，分别测试 userspace，ondemand，conservative，powersave，performance 模式。

(3) userspace 模式变频测试：

- ① 设置所有 CPU 为 userspace 模式，并验证；
- ② 选择测试 package 中随机 CPU 设置随机频率，并打印此 CPU 的频率，

验证 governor 为 userspace ；

- ③ 设置 package 中所有 CPU 为最低频 ；
- ④ 运行负载测试，获取平均测试时间 min_freq_average_runtime ；
- ⑤ 设置 package 中所有 CPU 为最高频 ；
- ⑥ 运行负载测试，获取平均测试时间 max_freq_average_runtime ；
- ⑦ 判定：预期测试时间比 $expect_speed = \max_freq / \min_freq$

实际测试时间比 $measured = \min_freq_average_runtime / \max_freq_average_runtime$

允许存在 20% 的误差，即 $expect_speed * 80\% \leq measured \leq expect_speed * 120\%$

(4) ondemand 模式变频测试：

- ① 设置所有 CPU 为 powersave 模式 ；
- ② 设置所有 CPU 为 ondemand 模式，并验证 ；
- ③ 获取 package 的所有 CPU 频率，并打印 ；
- ④ 运行负载测试，获取 CPU 频率并判定是否为最高频，否则此项测试失

败 ；

- ⑤ 获取平均测试时间 average_runtime ；
- ⑥ 如果存在 userspace_min_freq_runtime，则进行测试时间判定

实际测试时间比 $measured = \min_freq_average_runtime / average_runtime$

允许存在 20% 的误差，即 $expect_speed * 80\% \leq measured \leq expect_speed * 120\%$

⑦ 如果 CPU 支持超频，则测试此项：

选择此 package 中任一 CPU 运行负载测试，获取测试时间 runtime，判定 runtime 需小于 average_runtime。

(5) conservative 模式变频测试：

- ① 设置所有 CPU 为 performance 模式；
- ② 设置所有 CPU 为 conservative 模式，并验证；
- ③ 获取 package 的所有 CPU 频率，并打印；
- ④ 运行负载测试，获取 CPU 频率并判定是否在最低频与最高频之间；
- ⑤ 获取平均测试时间 average_runtime；
- ⑥ 打印 package 中 CPU 的频率；
- ⑦ 如果存在 userspace_min_freq_runtime，则进行测试时间判定

实际测试时间比 $measured = min_freq_average_runtime / average_runtime$

允许存在 20% 的误差，即 $expect_speed*80\% \leq measured \leq expect_speed*120\%$

(6) powersave 模式变频测试：

- ① 设置所有 CPU 为 powersave 模式；
- ② 验证 package 中 CPU 模式为 powersave，且频率为最低频；
- ③ 运行负载测试，并获取平均测试时间 average_runtime；
- ④ 打印测试后的 CPU 频率；
- ⑤ 如果存在 userspace_max_freq_runtime，则进行判定：

实际测试时间比 $measured = average_runtime /$

userspace_max_freq_runtime

允许存在 20% 的误差，即 $expect_speed*80\% \leq measured \leq expect_speed*120\%$

⑥ 如果存在 userspace_min_freq_runtime，则进行判定：

$$0.9 \leq average_runtime/userspace_min_freq_runtime \leq 1.1$$

(7) performance 模式变频测试：

- ① 设置所有 CPU 为 powersave 模式；
- ② 设置所有 CPU 为 performance 模式，并验证；
- ③ 获取 package 所有 CPU 频率并打印，检查是否为最大频；
- ④ 运行负载测试，获取平均测试时间 average_runtime；
- ⑤ 优先取 powersave 的 runtime，若没有则取 userspace_min_freq 的 runtime，存在 min_freq_runtime 则进行判定：

$$\text{实际测试时间比 } measured = min_freq_runtime / average_runtime$$

允许存在 20% 的误差，即 $expect_speed*80\% \leq measured \leq expect_speed*120\%$

⑥ 如果存在 userspace_max_freq_runtime，则进行判定：

$$0.9 \leq average_runtime/userspace_max_freq_runtime \leq 1.1$$

(8) 循环测试 3~7 直到所有 package 测试完成。

测试结果：

结果失败：命令执行失败或没有达标

结果通过：所有子测试项执行成功

测试时间：20min

disk

测试范围：

- (1) 收集 LVM 磁盘信息，收集设备系统信息，包括内核信息、分区表、交换分区、磁盘状态、分区挂载信息、内核模块信息和 PCI 设备信息；
- (2) 验证 I/O 子系统数据，并获取性能数据。测试系统所有存储设备。

测试流程：

注意：进行 disk 测试之前，需要提前确认系统至少存在一个分区，需要满足不是 lvm 分区，分区没有被挂载。

- (1) 获取磁盘和操作系统信息，包括 LVM 磁盘信息、内核信息、分区表、交换分区、磁盘状态、分区挂载信息、内核模块信息和 PCI 设备信息；

结果失败：有任意命令失败不能正确输出结果

结果通过：所有命令正常运行，有输出结果

- (2) 测试人员选定待测设备，进行 fio 测试；

- (3) 下面开始使用独立的测试工具 fio 进行硬盘测试：

① 使用 fio 工具进行 RAW I/O 测试：其中最大测试文件为 1GB，块大小取 4K 到 64K 以 2 倍递增（5 个值），对应每个块大小进行顺序读写和随机读写测试。共 10 条指令和输出数据；

② 进行 VFS (buffered filesystem) 测试：其中最大测试文件为 1GB，块大小取 4K 到 64K 以 2 倍递增（5 个值），对应每个块大小进行顺序读写和随机读写测试。共 10 条指令和输出数据；

- ③ 格式化并重新挂载待测分区。

结果失败：fio 测试过程返回值有任意一个非零值则失败

结果通过：fio 测试的返回值均为 0

测试结果：

结果失败：子测试项任意一个失败

结果通过：所有子测试项通过

测试时间：100min

filesystem

测试范围：

(1) 收集 LVM 磁盘信息，收集设备系统信息，包括内核信息、分区表、交换分区、磁盘状态、分区挂载信息、内核模块信息和 PCI 设备信息；

(2) 验证分区格式化。

测试流程：

注意：进行 filesystem 测试之前，需要提前确认系统至少存在一个大于 2G 的分区，并且满足两个要求：分区不是 lvm，分区没有被挂载。

(1) 获取磁盘和操作系统信息，包括 LVM 磁盘信息、内核信息、分区表、交换分区、磁盘状态、分区挂载信息、内核模块信息和 PCI 设备信息；

结果失败：有任意命令失败不能正确输出结果

结果通过：所有命令正常运行，有输出结果

(2) 测试人员选定待测设备，进行格式化测试：

① 使用 mkfs 命令格式化 ext2/ext3/ext4/ntfs/xfs/vfat；

② 对格式化后的磁盘，挂载到/mnt；

③ 执行 dd 命令写入；

④ 卸载/mnt；

测试结果：

结果失败：子测试项任意一个失败

结果通过：所有子测试项通过

测试时间：5min

hardwareinfo

测试范围：

用于抓取系统各硬件信息。

测试流程：

(1) 使用命令 `cat /etc/os-release, cat /etc/.productinfo, nkvers` 抓取系统版本信息；

(2) 使用命令 `lscpu, /proc/cpuinfo, dmidecode -t processor` 抓取系统cpu 信息；

(3) 使用命令 `free, /proc/meminfo, dmidecode -t memory` 抓取系统内存信息；

(4) 使用命令 `fdisk -l, df -h, lsblk` 等抓取系统磁盘信息；

(5) 使用命令 `dmidecode -t baseboard` 查看主板信息；

(6) 使用命令 `lspci` 查看 VGA 信息；

(7) 使用命令 `lspci, ethtool` 查看网卡信息；

(8) 使用脚本查看 raid 卡信息；

(9) 使用 `dmidecode -t bios` 查看 BIOS 信息；

(10) 抓取 `sosreport`。

测试结果：

结果失败：命令执行返回值非 0

结果成功：命令执行返回值为 0

测试时间：10min

ipmi

测试范围：

用于测试操作系统与 BMC 兼容性，打印 fru 和传感器信息。

测试流程：

(1) 开启 ipmi 服务；

(2) 执行命令 `ipmitool fru`，打印内建的 Field Replaceable Unit (FRU) 信息；

(3) 执行命令 `ipmitool sensor`，打印详细的传感器信息；

(4) 执行命令 `ipmitool mc info`，打印 BMC 版本信息。

测试结果：

结果失败：服务开启失败，命令执行返回值非 0

结果成功：服务开始成功，命令执行返回值为 0

测试时间：2min

kdump

测试范围：

测试系统的 kdump 的本地转存能力以及生产 crash 文件的准确性。

注意：kernel-debug 版本需要和内核版本匹配，需要 kexec-tools。

测试流程：

- (1) 检查/proc/cmdline 设置了 crashkernel；
- (2) 配置/etc/kdump.conf，如下并重启 kdump 服务：

```
path /var/crash  
  
kdump_obj all
```
- (3) 询问是否重启；
是：继续执行
否：测试失败，退出测试
- (4) sync;echo c >/proc/sysrq-trigger；
- (5) 开机后验证/var/crash 目录下是否有 vmcore 文件。

测试结果：

结果失败：kernel-debug 与内核版本不符，没有设置 crashkernel，没有生成 vmcore 文件

结果通过：可以正常模拟 kdump 并能生成 vmcore 文件

测试时间：15min

loigcalvolume

测试范围：

- (1) 收集 LVM 磁盘信息，收集设备系统信息，包括内核信息、分区表、交换

分区、磁盘状态、分区挂载信息、内核模块信息和 PCI 设备信息；

(2) 创建逻辑卷，删除逻辑卷测试。

测试流程：

注意：进行 logicalvolume 测试之前，需要提前确认系统至少存在一个大于 2G 的分区，并且满足两个要求：分区不是 lvm，分区没有被挂载。

(1) 获取磁盘和操作系统信息，包括 LVM 磁盘信息、内核信息、分区表、交换分区、磁盘状态、分区挂载信息、内核模块信息和 PCI 设备信息；

结果失败：有任意命令失败不能正确输出结果

结果通过：所有命令正常运行，有输出结果

(2) 测试人员选定待测设备，进行逻辑卷测试：

① 创建逻辑卷；

② 查看逻辑卷；

③ 删除逻辑卷；

测试结果：

结果失败：子测试项任意一个失败

结果通过：所有子测试项通过

测试时间：5min

memory

测试范围：

主要测试系统的 RAM 和 swap 分区。

测试流程：

(1) 通过/proc/meminfo 获取内存相关信息：总内存 (System Memory)、空闲内存 (Free Memory)、可用内存 (Available Memory)、交换分区 (Swap Memory)、HugePages Total、HugePages Free、HugePages Size ；

(2) 内存读写测试

① 获取测试内存 test_mem=空闲内存*90/100，最大为 4G ；

② 执行命令测试：memtester test_memM 1。

(3) 内存压力测试

① 打印内存信息；

② 利用内存测试脚本进行进程加压测试，test_mem= 空闲内存 +extra_mem，extra_mem 最大为 512M ；

③ 执行测试./eatmem_test -m test_memM，测试时间 120s，创建 num_cpus*2 数量的线程用来加压，每个线程映射的内存大小为 test_mem/ (num_cpus*2)，最后返回一个值判定是否测试通过；

(4) 大页内存测试

① 打印大页内存信息；

② 利用脚本进行测试：./hugett1b。

测试结果：

结果失败：存在测试子项测试失败，返回值为非 0

结果通过：所有测试子项通过测试，返回值为 0

测试时间：40min

mobileharddisk

测试范围：

用于测试移动硬盘的识别、读写、卸载功能。

测试流程：

- (1) 提示连接移动硬盘，连接后按 y 继续测试；
- (2) 获取连接的移动硬盘盘符并打印 lsblk；
- (3) 卸载移动硬盘，格式化为 ext4 格式；
- (4) 挂载移动硬盘至/mnt；
- (5) 执行 `dd if=/dev/zero of=/mnt/test.img bs=1M count=1024 oflag=direct` 写入 1G 文件，展示写入速率；
- (6) 执行 `dd if=/mnt/test.img of=/dev/null bs=1M` 读出 1G 文件到内存，展示读取速率；
- (7) 执行 `umount` 卸载移动硬盘。

测试结果：

结果失败：移动硬盘未连接或命令执行返回值非 0

结果成功：命令执行返回值为 0

测试时间：15min

mobilehdhotplug

测试范围：

用于测试移动硬盘热插拔功能。

测试流程：

- (1) 提示将已连接的移动硬盘移除，按 “y” 继续测试；

- (2) 提示连接移动硬盘，连接后按 “y” 继续测试；
- (3) 识别是否有移动硬盘连接：
 - 是：打印 lsblk
 - 否：退出测试
- (4) 提示移除第 2 步连接的移动硬盘，按 “y” 继续测试；
- (5) 检查移动硬盘是否移除：
 - 是：打印 lsblk
 - 否：退出测试
- (6) 循环第 2 步到第 5 步 10 次。

测试结果：

结果失败：移动硬盘识别失败

结果成功：所有子项测试通过

测试时间：10min

network

测试范围：

测试系统所有的网卡设备。

测试流程：

- (1) 打印测试端口信息，包括测试端口，测试端 ip，服务端 ip，ethtool 信息，mac 地址，biosdevname -d；
- (2) 使用 ifconfig 和 ip link set 测试端口断开连接，确认能够连接服务端，获取测试端口速度；
- (3) icmp (丢包率) 测试，ping -q -c 500 -i 0 server_ip，没有丢包，则测

试通过；

(4) netperf 测试，最多重试 3 次：

- ① tcp stream 测试，带宽达到待测端口速度的 90%，则结果通过；
- ② tcp rr 测试，有测试结果打印，则结果通过；
- ③ tcp crr 测试，有测试结果打印，则结果通过；
- ④ udp stream 测试，有测试结果打印，则结果通过；
- ⑤ udp rr 测试，有测试结果打印，则结果通过；
- ⑥ omni 测试，有测试结果打印，则结果通过。

(5) http 测试，主要测试文件的上传及下载的时间及速率，以及验证下载文件与本地文件的一致性；

(6) 清理测试环境，恢复端口连接。

测试结果：

结果失败：获取信息失败，netperf 运行失败，丢包率不达标，http 上传下载失败

结果通过：所有信息收集正确且带宽和丢包率达标

测试时间：20min * 测试端口

nvme

测试范围：

测试 nvme 磁盘

测试流程：

注意：机器上连接有 nvme 磁盘才会加载此测试项。

(1) 确认待测设备未挂载可使用；

(2) 使用 `nvme-cli` 工具对磁盘进行 `format`、`write`、`read`、`smart-log`、`get-log`、`list` 测试。

测试结果：

结果失败：待测设备不可用或命令执行失败

结果通过：待测设备可用且命令执行成功

测试时间：20min

perf

测试范围：

测试 `perf` 工具是否能够正常使用。

测试流程：

注意：`perf` 版本要和内核版本一致。

(1) 验证 `perf` 版本与内核版本一致；

(2) 检查循环事件，`perf record -a -e cycles -o hwcompatible-perf.data`
`sleep 5`；

(3) 检查是否检测到 `cycles` 事件，`perf evlist -i hwcompatible-perf.data`；

(4) 检查样本是否被收集，`perf report -i hwcompatible-perf.data --stdio`。

测试结果：

结果失败：有命令执行失败

结果通过：所有命令执行成功

测试时间：5min

system

测试范围：

测试并收集系统信息：工具是否被修改、os 版本与内核是否匹配、内核是否被修改、selinux 状态。

测试流程：

- (1) 检查本工具是否被修改；
- (2) 检查 OS 版本和 kernel 版本是否匹配；
- (3) 检查内核是否被修改/感染；
- (4) 检查 selinux 是否正常启用；

测试结果：

结果失败：所有信息收集中任意一个失败。

结果通过：所有信息收集正确，没有内容被更改。

测试时间：1min

usb

测试范围：

测试 USB 端口热插拔功能，保证 USB 设备能够在系统下正常使用。

测试流程：

- (1) 利用 udevadm 工具获得所有 usb 设备；
- (2) 提示插入待测试的 USB 设备，选择一个 USB 端口插入设备：
 - 是：继续下一步
 - 否：重复 2

(3) 检测到 USB 设备，打印总线 id；

(4) 提示拔出 USB 设备，手动拔出设备：

是：继续下一步

否：重复 4

(5) 检测到 USB 设备拔出；

(6) 询问所有端口是否已测试：

是：退出

否：重复 1-6

测试结果：

结果失败：没有检查到 usb 设备连接或拔出

结果通过：所有项测试通过

测试时间：10min

udiskhotplug

测试范围：

测试 U 盘热插拔功能。

测试流程：

(1) 提示将已连接的 U 盘移除，按 “y” 继续测试；

(2) 提示连接 U 盘，连接后按 “y” 继续测试；

(3) 识别是否有 U 盘连接：

是：打印 lsblk

否：退出测试

(4) 提示移除第 2 步连接的 U 盘，按 “y” 继续测试；

(5) 检查 U 盘是否移除：

是：打印 lsblk

否：退出测试

(6) 循环第 2 步到第 5 步 10 次。

测试结果：

结果失败：U 盘识别失败

结果通过：所有子项测试通过

测试时间：5min

usb_keyboard

测试范围：

测试 usb 键盘输入，保证键盘能够在系统下正常使用。

测试流程：

(1) 提示插入键盘：

是：继续下一步

否：重复 1

(2) 在弹框里输入字符；

(3) 验证输入与显示一致，结果通过。

测试结果：

结果失败：键盘输入字符错误

结果通过：键盘输入字符正确

测试时间：2min

usb_mouse

测试范围：

测试 usb 鼠标点击，保证鼠标能够在系统下正常使用。

测试流程：

(1) 提示插入鼠标：

是：继续下一步

否：重复 1

(2) 点击弹框中的按钮关闭弹框。

测试结果：

结果失败：通过其他方式关闭弹框

结果通过：点击按钮关闭弹框

测试时间：2min

usb_udisk

测试范围：

测试 U 盘识别、读写、格式化、卸载功能。

测试流程：

(1) 提示连接移动硬盘，按 “y” 继续测试；

(2) 获取连接的 U 盘盘符并打印 lsblk；

(3) 在 U 盘挂载目录，创建 test 文件夹并删除；

(4) 卸载 U 盘，格式化为 ext4 格式；

- (5) 挂载移动硬盘至/mnt；
- (6) 检查 U 盘是否格式化为 ext4；
- (7) 执行 `dd if=/dev/zero of=/mnt/test.img bs=1M count=1024 oflag=direct` 写入 1G 文件，展示写入速率；
- (8) 执行 `dd if=/mnt/test.img of=/dev/null bs=1M` 读出 1G 文件到内存，展示读取速率；
- (9) 执行 `umount` 卸载移动硬盘。

测试结果：

结果失败：U 盘未识别到或命令执行返回值非 0

结果通过：命令执行返回值 0

测试时间：15min

usbcdriverhotplug

测试范围：

测试 usb 光驱热插拔功能。

测试流程：

- (1) 提示将已连接的 usb 光驱移除，按 “y” 继续测试；
- (2) 提示连接 usb 光驱，连接后按 “y” 继续测试；
- (3) 识别是否有 usb 光驱连接：
是：打印 `lsblk`
否：退出测试
- (4) 提示移除第 2 步连接的 usb 光驱，按 “y” 继续测试；

(5) 检查 usb 光驱是否移除：

是：打印 lsblk

否：退出测试

(6) 循环第 2 步到第 5 步 10 次。

测试结果：

结果失败：usb 光驱识别失败

结果通过：所有子项测试通过

测试时间：10min

video

测试范围：

测试视频连接，xserver 服务运行。

测试流程：

注意：此项测试不能通过远程 ssh 连接测试。

(1) 测试显示器连接执行 xrandr 命令；

结果失败：xrandr 命令执行失败

结果通过：xrandr 命令执行成功

(2) glxgears -display 测试 5 次；

(3) 查看模块和驱动信息。

测试结果：

结果失败：有测试项目执行失败

结果通过：所有测试项目执行成功

测试时间：10min

watchdog

测试范围：

触发 watchdog，测试系统是否可以正常复位。

测试流程：

- (1) 加载 softdog 模块；
- (2) 设置最大超时时间为 20s；
- (3) 如果是前台进程，询问是否重启，否则直接重启：
 - 是：执行测试
 - 否：退出并测试失败
- (4) 重启后检查，是否重启成功。

测试结果：

结果失败：触发 watchdog 超时

结果通过：所有测试项执行成功

测试时间：10min

stream

测试范围：

使用 stream 测试工具测试内存带宽。

测试流程：

- (1) 检查工具是否存在，不存在直接退出；

(2) 打印默认配置参数信息，并询问是否需要调整，超时时间 60s：

```

---- start to run test stream time: 2021-09-02 15:35:56 ----
[INFO]check stream tools exists...
[INFO] 检测到工具存在
当前配置测试参数:
使用工具版本: stream-5.9-1.tar.bz2
测试次数: 1
编译参数ntimes: 100
编译参数array_size: 80000000
测试最大线程: 满线程
-----
是否需要配置测试参数 (输入超时: 60 s) (y|n) y
请输入使用工具版本(默认stream-5.9-1.tar.bz2):
请输入测试次数(默认1): 3
请输入编译参数ntimes(默认100):
请输入编译参数array_size(默认80000000):
请输入测试最大线程(默认满线程): 8
-----
设置后配置测试参数:
使用工具版本: stream-5.9-1.tar.bz2
测试次数: 3
编译参数ntimes: 100
编译参数array_size: 80000000
测试最大线程: 8
-----
cd /opt/performance_server;./stream.sh.x -server_performance 2>&1
    
```

图 25-stream 配置参数

是：进入参数配置页面，完成后打印配置后的参数

否：使用默认参数测试

超时：使用默认参数测试

参数包括：

- ① 使用工具：str 类型，工具包名称
- ② 测试次数：int 类型，用于多次测试
- ③ 编译参数 ntimes：int 类型，用于编译，取测试 ntimes 次最优值
- ④ 编译参数 array_size：int 类型，用于编译至少为 4 倍 L3 缓存
- ⑤ 测试最大线程：int 类型，默认测试单线程和满线程，此处可以设置测

试满线程为指定值。

- (3) 开始测试 stream ;
- (4) 测试完成打印 json 格式信息 ;
- (5) 拷贝测试日志 , 测试结果到 log 存储路径的 stream 文件夹下。

测试结果 :

结果失败 : 工具不存在 , 执行测试失败

结果通过 : 所有测试项执行成功

测试时间 : 10min

fio

测试范围 :

使用 fio 测试工具测试磁盘读写速率。

测试流程 :

- (1) 检查工具是否存在 , 不存在直接退出 ;
- (2) 打印默认配置参数信息 , 并询问是否需要调整 , 超时时间 60s :
 - 是 : 进入参数配置页面 , 完成后打印配置后的参数
 - 否 : 使用默认参数测试
 - 超时 : 使用默认参数测试

参数包括 :

- ① 使用工具版本 : str 类型 , 工具包名称
- ② 测试线程数 : int 类型 , 默认为机器核数 , 限制最大为 16
- ③ 队列深度 : int 类型 , 默认 32
- ④ 测试路径 : str 类型 , 默认测试在根目录/fio_test

- ⑤ 测试次数：int 类型，默认 3，用于测试多次取平均值
- (3) 开始测试 fio；
- (4) 测试完成打印 json 格式测试结果信息；
- (5) 拷贝测试日志，测试结果到 log 存储路径的 fio 文件夹下。

测试结果：

结果失败：工具不存在，执行测试失败

结果通过：所有测试项执行成功

测试时间：80min

iozone

测试范围：

使用 iozone 测试工具测试磁盘读写速率。

测试流程：

- (1) 检查工具是否存在，不存在直接退出；
- (2) 打印默认配置参数信息，并询问是否需要调整，超时时间 60s：
 - 是：进入参数配置页面，完成后打印配置后的参数
 - 否：使用默认参数测试
 - 超时：使用默认参数测试

参数包括：

- ① 执行脚本参数：str 类型，默认 server_performance
 - a. desktop_performance 测试 16M 块大小文件大小分别为 2 倍内存、1 倍内存、0.5 倍内存的随机读写、顺序读写速率。

b. server_performance 默认测试 2 倍内存块范围 64K 到 16M 下的随机读写、顺序读写。

c. m2test 同上

d. r64kt4 测试命令 `iozone -s 1g -r 64k -i 0 -i 1 -t 4 -F /iozone1 /iozone2 /iozone3 /iozone 4`

e. r64kt1 测试命令 `iozone -s 1g -r 64k -i 0 -i 1 -f /iozone`

② 使用工具版本：str 类型，工具包名称

③ 测试路径：str 类型，默认根目录/

④ 测试次数：int 类型，默认 1

(3) 如果执行参数为 server_performance 或 m2test 则会触发此参数下的参数配置，打印默认配置参数，并询问是否需要配置：

是：进入参数配置页面，完成后打印配置后的参数

否：使用默认参数测试

参数包括：

① 全面测试块范围最小值：str 类型，默认 64K

② 全面测试块范围最大值：str 类型，默认 16M

③ 全面测试文件范围最小值：str 类型，默认 2 倍内存，需要带单位，例如 10g

④ 全面测试文件范围最大值：str 类型，默认 2 倍内存，需要带单位，例如 20g

(4) 开始测试 iozone；

(5) 测试完成打印 json 格式测试结果信息；

(6) 拷贝测试日志，测试结果到 log 存储路径的 iozone 文件夹下。

测试结果：

结果失败：工具不存在，执行测试失败

结果通过：所有测试项执行成功

测试时间：360min

Imbench

测试范围：

使用 Imbench 测试工具测试系统性能。

测试流程：

- (1) 检查工具是否存在，不存在直接退出；
- (2) 打印默认配置参数信息，并询问是否需要调整，超时时间 60s：
 - 是：进入参数配置页面，完成后打印配置后的参数
 - 否：使用默认参数测试
 - 超时：使用默认参数测试

参数包括：

- ① 使用工具版本：str 类型，工具包名称
 - ② 测试内存：int 类型，默认 16384，单位 M，限制最大值为 16384
- (3) 开始测试 Imbench；
 - (4) 测试完成打印 json 格式测试结果信息；
 - (5) 拷贝测试日志，测试结果到 log 存储路径的 Imbench 文件夹下。

测试结果：

结果失败：工具不存在，执行测试失败

结果通过：所有测试项执行成功

测试时间：900min

netperf

测试范围：

使用 netperf 测试工具测试网络性能。

测试流程：

- (1) 检查工具是否存在，不存在直接退出；
- (2) 打印默认配置参数信息，并询问是否需要调整：
 - 是：进入参数配置页面，完成后打印配置后的参数
 - 否：使用默认参数测试

参数包括：

- ① 使用工具版本：str 类型，工具包名称
- ② 执行脚本参数：不可输入多个参数
 - a. server : 测 试
tct_stream,udp_stream,tcp_rr,udp_rr,trans_rate,omni,tcp_crr
 - b. tcp_stream : 测试命令 netperf -H <serverip> -t TCP_STREAM -l
<time>
 - c. udp_stream :测试命令 netperf -H <serverip> -t UDP_STREAM -l
<time>
 - d. tcp_rr : 测试命令 netperf -H <serverip> -t TCP_RR -l <time>

- e. udp_rr : 测试命令 netperf -H <serverip> -t UDP_RR -l <time>
- f. ping_sh : 测试命令 ping <serverip> -c 10
- g. trans_rate : 测试命令 netperf -H <serverip> -l <time>
- h. omni :测试命令 netperf -H <serverip> -t omni -- -d rr -l <time>
-O
" THROUGHPUT,THROUGHPUT_UNITS,MIN_LATENCY,MAX_L
ATENCY,MEAN_LATENCY"
- i. tcp_crr : 测试命令 netperf -H <serverip> -t TCP_CRR -l <time>

- ③ 测试时间 : int 类型 , 默认 120 , 单位 s
- ④ server 端 IP : 需要输入 ip , 默认为 kylinch 服务端 ip
- ⑤ server 端用户名 : str 类型 , 默认 root
- ⑥ server 端密码 : str 类型 , 默认 123456
- ⑦ server 端安装目录 : str 类型 , 默认 /home/kylin , 用于在 server 端安

装 netperf

```

---- start to run test netperf time: 2021-09-02 19:32:35 ----
[INFO] 测试前请确保已连接服务端且关闭服务端的防火墙
[INFO]check netperf tools exists...
[INFO] 检测到工具存在
当前配置测试参数:
使用工具版本: netperf-2.7.0.zip
执行脚本参数[server tcp_stream udp_stream tcp_rr udp_rr ping_sh trans_rate omni tcp_crr]: server
测试时间-单位s: 120
server端IP: 192.66.1.234
server端用户名: root
server端密码: 123456
server端脚本安装目录: /home/kylin
-----
是否需要配置测试参数 (输入超时: 0 s) (y/n) y
请输入使用工具版本(默认netperf-2.7.0.zip):
请输入执行脚本参数[server tcp_stream udp_stream tcp_rr udp_rr ping_sh trans_rate omni tcp_crr](默认server): tcp_rr
请输入测试时间-单位s(默认120): 60
请输入server端IP(默认192.66.1.234):
请输入server端用户名(默认root):
请输入server端密码(默认123456): qwerr13
请输入server端脚本安装目录(默认/home/kylin): /opt
-----
设置后配置测试参数:
使用工具版本: netperf-2.7.0.zip
执行脚本参数[server tcp_stream udp_stream tcp_rr udp_rr ping_sh trans_rate omni tcp_crr]: tcp_rr
测试时间-单位s: 60
server端IP: 192.66.1.234
server端用户名: root
server端密码: qwerr13
server端脚本安装目录: /opt
-----
    
```

图 26-netperf 参数配置

- (3) 开始测试 netperf ;
- (4) 测试完成打印 json 格式测试结果信息 ;
- (5) 拷贝测试日志 , 测试结果到 log 存储路径的 netperf 文件夹下。

测试结果 :

结果失败 : 工具不存在 , 执行测试失败

结果通过 : 所有测试项执行成功

测试时间 : 15min

speccpu

测试范围 :

使用 speccpu 测试工具测试 CPU 性能。

测试流程 :

- (1) 检查工具是否存在 , 不存在直接退出 ;
- (2) 打印默认配置参数信息 , 并询问是否需要调整 :
 - 是 : 进入参数配置页面 , 完成后打印配置后的参数
 - 否 : 使用默认参数测试

参数包括 :

- ① 使用工具版本 : str 类型 , 工具包名称
 - ② 测试核数 : int 类型 , 默认满线程 , 此参数为设置多线程测试的核数
 - ③ 测试次数 : int 类型 , 默认 3
- (3) 开始测试 speccpu ;
 - (4) 测试完成打印 json 格式测试结果信息 ;

(5) 拷贝测试日志，测试结果到 log 存储路径的 speccpu 文件夹下。

测试结果：

结果失败：工具不存在，执行测试失败

结果通过：所有测试项执行成功

测试时间：2880min

specjvm

测试范围：

使用 specjvm 测试工具测试 java 基准性能。

测试流程：

- (1) 检查工具是否存在，不存在直接退出；
- (2) 打印默认配置参数信息，并询问是否需要调整：
 - 是：进入参数配置页面，完成后打印配置后的参数
 - 否：使用默认参数测试

参数包括：

- ① 执行脚本参数：str 类型，默认 all
 - a. base：测试命令 `java -jar SPECjvm2008.jar -ikv -base`
 - b. peak：测试命令 `java -Xms1024m -Xmx<2/3 内存>m -jar SPECjvm2008.jar -ikv -peak -Dspecjvm.benchmark.threads=<CPUN>`
 - c. all：包括 base 和 peak

- (3) 开始测试 specjvm；

(4) 测试完成打印 json 格式测试结果信息；

(5) 拷贝测试日志，测试结果到 log 存储路径的 specjvm 文件夹下。

测试结果：

结果失败：工具不存在，执行测试失败

结果通过：所有测试项执行成功

测试时间：600min

unixbench

测试范围：

使用 unixbench 测试工具测试系统性能。

测试流程：

(1) 检查工具是否存在，不存在直接退出；

(2) 打印默认配置参数信息，并询问是否需要调整：

是：进入参数配置页面，完成后打印配置后的参数

否：使用默认参数测试

参数包括：

① 执行脚本参数：str 类型，默认 base

a. single：测试命令 Run -c 1

b. multi：测试命令 Run -c <CPUN>

c. base：测试命令 Run -c 1 -c <CPUN>

d. graphics：Run graphics

e. all：包括 base、graphics

- ② 使用工具版本：str 类型，工具包名称
- ③ 测试线程数：int 类型，默认为 CPU 核数，仅对 multi 和 base 参数的

CPUN 生效

- (3) 开始测试 unixbench；
- (4) 测试完成打印 json 格式测试结果信息；
- (5) 拷贝测试日志，测试结果到 log 存储路径的 unixbench 文件夹下。

测试结果：

结果失败：工具不存在，执行测试失败

结果通过：所有测试项执行成功

测试时间：90min

Itp

测试范围：

使用 Itp 测试工具测试系统稳定性。

测试流程：

- (1) 检查工具是否存在，不存在直接退出；
- (2) 打印默认配置参数信息，并询问是否需要调整：
 - 是：进入参数配置页面，完成后打印配置后的参数
 - 否：使用默认参数测试

参数包括：

- ① 测试时间参数，int 类型，默认 48，单位小时
- (3) 开始测试 Itp；

- (4) 测试完成打印 ltp 每项测试结果信息；
- (5) 拷贝测试日志，测试结果到 log 存储路径的 ltp 文件夹下。

测试结果：

结果失败：工具不存在，执行测试失败

结果通过：所有测试项执行成功

测试时间：48h

rebootstress

测试范围：

系统重启压力测试。

测试流程：

- (1) 打印默认配置参数信息，并询问是否需要调整：
 - 是：进入参数配置页面，完成后打印配置后的参数
 - 否：使用默认参数测试
- 参数包括：
- ① 测试次数参数，int 类型，默认 100
 - (2) 清除系统日志 messages 和 dmesg
 - (3) 抓取系统磁盘、网卡、内存信息；
 - (4) 如果是前台进程，询问是否重启，否则直接重启：
 - 是：执行测试
 - 否：退出并测试失败
 - (5) 重启后检查是否重启成功，硬件信息是否与重启前抓取一致，不一致则

退出测试；

(6) 全部测试完成后，查看并收集系统日志，无 error 或 fail 信息。

测试结果：

结果失败：触发重启失败，重启前后硬件信息不一致。

结果通过：所有测试项执行成功

测试时间：8h

6. FAQ

6.1. Perf 测试 FAIL

报错信息：“[ERROR] perf 版本与内核版本不一致，无法测试 perf，请更新 perf”

解决方案：

1. `uname -a` 查看内核版本
2. 在“工具包/架构/”目录或“工具包/架构/updates”目录下寻找与内核版本一致的 `perf*.rpm` 包，离线安装。或者在 `yum 源/debug` 目录下寻找。
3. 如未找到对应内核版本的 `perf rpm` 包，请联系接口人获取。

6.2. kdump 和 watchdog 测试后没有自动打包测试日志

`kdump` 和 `watchdog` 涉及重启，需要等待测试完成后在终端输入“`kylinch`”进行日志打包上传。

6.3. 测试 fail 查看报错信息

如果 fail 项包含测试子项，可查看对应测试项 log 最底部的 subtest，如下

图：

```
-----
SubTest:  check certrpm                FAIL
SubTest:  check kernel                  FAIL
SubTest:  check selinux                  PASS
SubTest:  check cpuinfo                  PASS
SubTest:  check meminfo                  PASS
SubTest:  check VGA                      PASS
SubTest:  check ethernet                 PASS
```

图 27-测试子项

可以看到具体失败的测试子项（有些测试项比较简单，不包括测试子项，则不显示 SubTest）

在日志中搜索[ERROR]或[FAIL]可以看到错误输出，如下图

```
[+] Checking installed cert package...
S.5..... /usr/share/kylinch/lib/tests/cpufreq/cpufreq.so
[ERROR] Files in kylin-ch have been tampered.

[+] Checking kernel...
Kernel RPM: kernel-4.19.90-22.3.ky10.mips64el-rt35
OS Version: Kylin Linux Advanced Server V10 (Tercel)
Detailed OS Version: Kylin Linux Advanced Serverrelease V10 (SP1) /(Tercel)-mips64el-Build08/20201030

[INFO] Checking debug kernel...
[INFO] 内核未受到污染
[ERROR] Files from kernel-4.19.90-22.3.ky10.mips64el-rt35 were modified.

Boot Parameters: root=/dev/sda3 ro rhgb quiet console=tty quiet crashkernel=512M rd_start=0xffffffff887b000
```

图 28-测试日志中的报错信息

6.4. Intel cpufreq 测试问题

问题描述 :cpufreq 测试 pass 但是日志中 powersave 模式和 performance

模式的测试时间几乎一样。

解决方案：

查看 cpufreq 驱动，可以通过查看 log 或执行命令的方式查看

cpufreq 测试日志

```

---- start to run test cpufreq ----

##### CPU Info#####
CPU Model           : Loongson-3A R4 (Loongson-3B4000)
CPU Max Frequency   : 1800000
CPU Min Frequency   : 900000
CPU Nums            : 8
CPU Package 0      : [0, 1, 2, 3]
CPU Package 1      : [4, 5, 6, 7]
CPU Available Governors : ['conservative', 'ondemand', 'userspace', 'powersave', 'performance']
CPU Flags           : None
CPU Driver          : loongson3
    
```

图 29-测试日志查看 cpu 驱动

或者执行命令：

cpupower -c 0 frequency-info

```

[root@localhost ~]# cpupower -c 0 frequency-info
analyzing CPU 0:
driver: acpi-cpufreq
CPUs which run at the same hardware frequency: 0
CPUs which need to have their frequency coordinated by software: 0
maximum transition latency: Cannot determine or is not supported.
hardware limits: 1.60 GHz - 2.50 GHz
available frequency steps: 2.50 GHz, 2.00 GHz, 1.60 GHz
available cpufreq governors: conservative ondemand userspace powersave performance
current policy: frequency should be within 1.60 GHz and 2.50 GHz.
                    The governor "performance" may decide which speed to use
                    within this range.
current CPU frequency: 2.50 GHz (asserted by call to hardware)
boost state support:
  Supported: no
  Active: no
  Boost States: 0
    
```

图 30-命令查看 cpu 驱动

如果 driver 是 intel_pstate，那么此项 pass。因为使用 intel_pstate 驱动调频时 powersave 模式更类似于 ondemand 模式，频率不会保持最低，会随着 cpu 压力而自行调节，导致加压时 powersave 模式和 performance 模式的测试时间几乎一样。

如果 driver 不是 Intel_pstate，那么这是一个 bug。

6.5. 终端打开 kylinch 提示没有需要运行的测试集，退出测试

问题描述：

安装 kylin-ch 后，终端运行 kylinch，提示“没有需要运行的测试集，退出测试”如下图

```
[root@localhost ~]# kylinch
The Kylin Hardware Compatibility Test Suite
Please provide your Compatibility Test ID:00
Please provide your Product URL:kylin
Please provide the Compatibility Test Server (Hostname or Ipaddr):192.66.1.1
Compatibility Test ID:      00
Hardware Info:              GreatWall 擎天DF723 NA
Product URL:                 kylin
OS Info:                     Kylin Linux Advanced Server V10 (Tercel)
Kernel Info:                 4.19.90-23.13.v2101.ky10.aarch64
Test Server:                 192.66.1.1

[ERROR]没有需要运行的测试集，退出测试。
[root@localhost ~]#
```

图 31-没有需要运行的测试集

解决方案：

没有安装 kylin-ch-testsuites 导致此问题，安装后就可以加载出测试集了。

6.6. 测试完成后无法上传日志到 server 端

问题描述：客户端能够 ping 通 server 端 ip，客户端或其他机器无法打开 server 端网页。

解决方案：可能是由于 server 端重启过导致服务未开启或防火墙未关闭，手动执行以下命令：

```
# systemctl start kylinch-server
```

```
# systemctl start nginx
```

```
# systemctl stop firewalld
```

```
# iptables -F
```

6.7. 已知问题

机型	问题描述	问题原因
CPU:ZHAOXIN KaiSheng KH-37800D	perf 测试 fail : The sys_perf_event_open() syscall returned with 16 (设备 或资源忙) for event (cycles).	系统原因, V10SP1-0518 版本还未合入修改。
CPU:龙芯 3B4000	cpufreq 测试 fail : package0 测试 pass ,package1 测试 fail , package1 的低频与高 频的测试时间相同	CPU 设计如此, 3B4000 不支持除 package 0 以外 cpu 调频
CPU:龙芯 3B4000	acpi 测试 fail	机器不支持 acpi 测试
CPU:龙芯 3B4000	ipmi 测试 fail : ipmitool fru 测试失败 ,ipmitool sensor 测试通过	BMC 不支持 fru
CPU:龙芯 3B4000	kdump 测试 fail :kdump 服务开 启失败	机器不支持 kdump

<p>泰山 200 服务器</p> <p>CPU : kunpeng920</p>	<p>watchdog 测试失败 ,无法触发重 启</p>	<p>BIOS 问题</p> <p>环境中执行 rmmod</p> <p>sbsa_gwdt</p> <p>modprobe sbsa_gwdt</p> <p>action=1 之后测试可以通 过</p>
---	-----------------------------------	--