

# 基于 Jenkins CI 的 mugen 自动化测试框架研究

第三测试小队

桜風の狐 测试工程师 [caiweilin@iscas.ac.cn](mailto:caiweilin@iscas.ac.cn)

# 内容架构

- mugen 是什么
- mugen 的局限性
- mugen 的 deb 系兼容
- mugen 的用例编写
- mugen 的 Jenkins CI 自动化
- mugen 的 openEuler riscv 2403 测试

# mugen 是什么

mugen 是 openEuler 社区开放的测试框架，提供公共配置和方法以便社区开发者进行测试代码的编写和执行。

mugen 适合纯命令行程序的部署测试（配置测试）。

<https://gitee.com/openeuler/mugen>

# mugen 的局限性

## 上游 mugen 测试

1. openEuler x86-64/aarch64
2. 自动化测试流程未知
3. 许多用例在正常的系统上也无法通过

## 目标 mugen 测试

1. 在 Debian/Ubuntu/Fedora/openEuler 测试
2. 在 x86-64 和 riscv64 架构上测试
3. 自动触发 Jenkins CI 开展测试
4. 在整个测试范围使用同一套测试流程
5. 持续的维护

# mugen 的 deb 系兼容 扩大适用范围

以 RUYI 包管理的自动化测试范围为例：

- x86-64 QEMU openEuler 23.09
- x86-64 QEMU Fedora 38
- x86-64 QEMU Ubuntu 22.04 LTS
- riscv64 QEMU openEuler 23.09
- riscv64 systemd-nspawn LicheePi 4A RevyOS

之前有实习生做过 deb 系的兼容，但是代价是不再兼容 rpm 系，而这里的测试需求是同时支持两者。

mugen 有且仅有 **DNF\_INSTALL** 和 **DNF\_REMOVE** 函数与发行版有关，用途是调用 dnf 进行软件包的安装与卸载。

# mugen 的 deb 系兼容 DNF\_INSTALL 与 DNF\_REMOVE

```
DNF_INSTALL xz
```

```
# 测试逻辑
```

```
DNF_REMOVE
```

DNF\_INSTALL 安装并记录安装的包和引入的依赖

DNF\_REMOVE 则读取这个列表来将这些软件包移除干净

# mugen 的 deb 系兼容 DNF\_INSTALL 的实现

```
dnf --assumeno install xz 2>&1 |  
  grep -wE "$(echo $repoList | sed 's/ /|/g')\" |  
  grep -wE \"$(uname -m)|noarch\" | awk '{print $1}'
```

从 dnf 的输出获取。这是一部分代码，如果有升级的包它要进行额外的处理，把升级的包从输出中过滤掉。

# mugen 的 deb 系兼容 deb 系的实现

```
APT_INSTALL xz-utils  
DNF_INSTALL xz
```

```
# 测试逻辑
```

```
PKG_REMOVE
```

由于两类发行版同一软件包的包名会有差异，所以采用这样的方式。



# mugen 的 deb 系兼容 APT\_INSTALL 的实现

```
apt-get --simulate --no-show-upgraded install xz 2>&1 |  
    grep -iA 1000 "NEW packages will be installed" |  
    grep -E "^ "
```

但是在 apt 这里不能使用 `--assume-no`，因为不见得它每次都会问你的，应当使用 `--simulate`。

# mugen 的 deb 系兼容 总结

实践证明这样的兼容方式是可行的，只是测试用例将不再兼容上游 mugen 框架。

当然这种兼容也有其他解决方式，比如给定特定的文件名，由 `dnf provides` 和 `apt-file search` 反查软件包名。

# mugen 的用例编写

实现持续维护

测试要求用例有足够的灵活性。

# mugen 的用例编写

输出格式依赖

mugen 非常依赖命令的输出格式，比如寻找 `ruyi list` 输出中第一个支持当前架构的工具链：

```
pkgnames=$(ruyi list | grep -e "^* toolchain" | cut -d '/' -f 2)
for p in $pkgnames; do
    s=$(ruyi list |
        awk '/\* / {if (f==1) f=2} /./ {if (f==1) {print $0}}
        /\* toolchain\/'$p' / {if (f==0) f=1}' |
        grep -e "^ -" |
        grep -v "no binary for current host")
    if [ ! -z "$s" ]; then
        pkgname="$p"
        break
    fi
done
```

# mugen 的用例编写 返回值依赖

mugen 非常依赖命令的返回值，比如 ruyi 的一个非预期用法：

```
ruyi  
CHECK_RESULT $? 0 1 "Check ruyi empty cmdline help failed"
```

看起来非常合理，在没有给出任何参数的时候打印帮助信息然后返回非 0 值。

在某个测试版本这个命令的返回值从 1 变成了 0，只好修改了用例：

```
ruyi 2>&1 |grep usage  
CHECK_RESULT $? 0 0 "Check ruyi empty cmdline help failed"
```

# mugen 的用例编写

测试内容固定

以 ruyi venv 为例，当前 mugen 测试包含的编译环境有如下组合：

工具链	sysroot	预置配置
gnu-upstream	自带	generic
gnu-plct	自带	generic
gnu-plct	自带	milkv-duo
gnu-plct-xthead	自带	sipeed-lpi4a
llvm-upstream	gnu-upstrea	generic

这是由多个用例交叉覆盖的，比如在测试 qemu-user-riscv-xthead 时同时就测试了 gnu-plct-xthead 与 sipeed-lpi4a 的配置组合。

当前 RUYI 包管理的 mugen 用例主要覆盖了文档中载明的一些使用场景。我在写用例的时候是一边手测、一边写文档、一边写用例。

# mugen 的用例编写

ruyi venv 用例覆盖方案

但是前文所述逐个覆盖的方法是不长久的：

1. 工具链会越来越多，预置配置也会越来越多（如何验证合法组合能够构建 coremark，非法组合报 Error）
  - 工具链与预置配置的组合测试
  - 不带 sysroot 的工具链（主要是 llvm）与 sysroot 的组合测试
2. 遍历工具链二进制，保证工具链是可以运行的（比如不能缺动态链接库）
  - 大部分可以用 `--help` 测试
  - 少部分在用例中指定

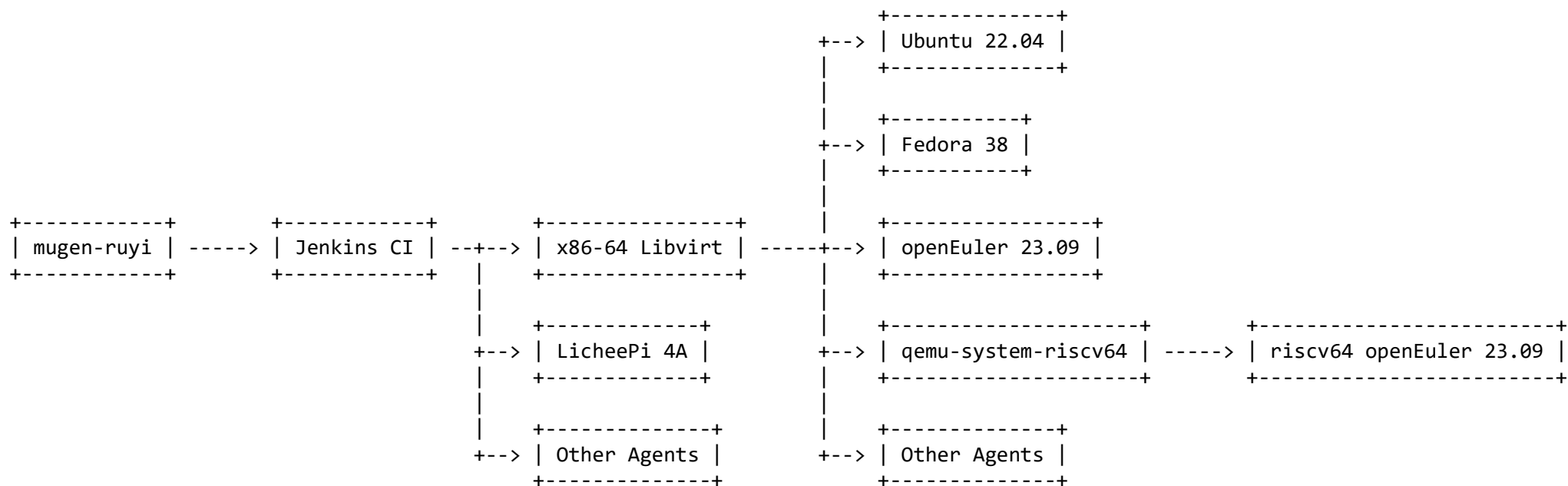
这通常使用 2~3 个用例即可覆盖完全。既可以以简单遍历的方式进行，也可以以自动生成和运行测试套的方式进行。

# mugen 的 Jenkins CI 自动化

完成了 mugen 的 deb 系兼容和测试用例的编写之后，就可以放心地将测试套扔进仓库，再由某个仓库触发 CI，实现整个自动化测试流程。



# mugen 的 Jenkins CI 自动化 还是以 ruyi 为例



Jenkins CI 根据 Pipeline 脚本调用指定的 Agents ，配置测试环境开展测试。  
其中 Libvirt 管理的虚拟机为本地虚拟机，而荔枝派 4A 则是远程连接到 CI 。

<https://gitee.com/weilinfox/mugen-ruyi>

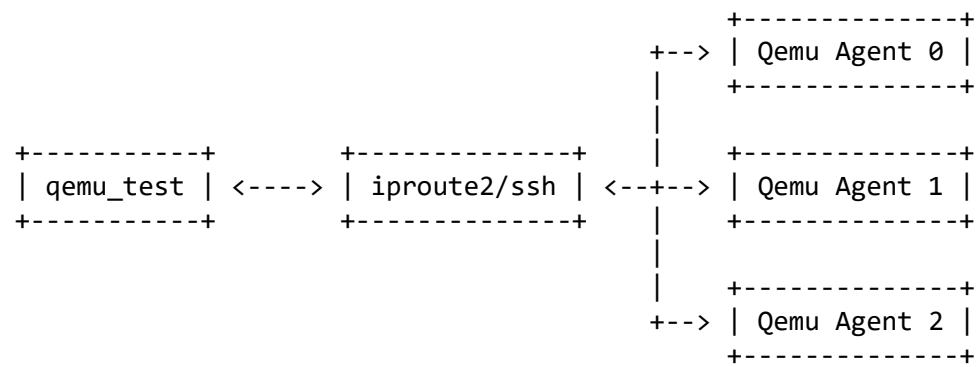
# mugen 的 openEuler riscv 2403 测试

这套测试流程最早是想应用在 openEuler 2309 独立发行版以及 2403 的 mugen 测试，主要用于替代全量测试中大量的手动重测。

但是随着第一次全量测试有了一个初步分析结果，80~90% 的失败是上游测试用例的问题（在 x86-64 无法通过），遗留重测数量并不多。所以预计 2403 的 mugen 测试依然以自动化测试为主、手动验证为辅的方式进行。

# mugen 的 openEuler riscv 2403 测试 2309 的残余

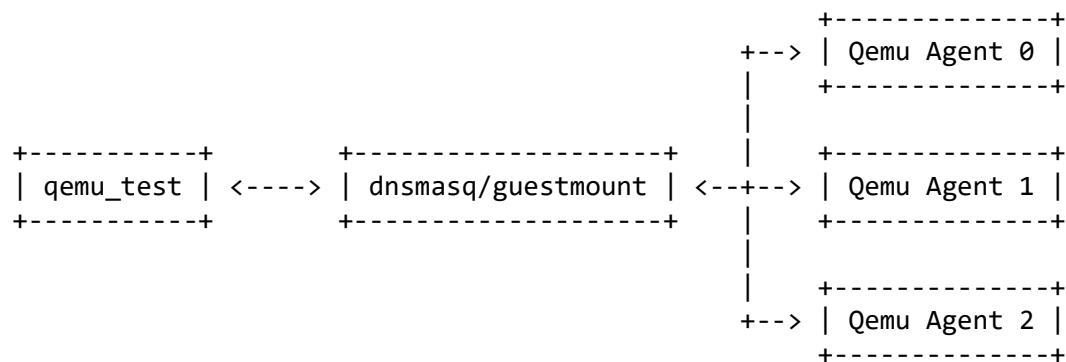
在 openEuler riscv 的 mugen 自动化测试中，测试是由 `qemu_test.py` 脚本来调度的，在 2309 发版测试中，遗留两个测试框架相关的问题：



- 1. 测试用例更改 ssh 配置导致测试机失联，日志取不出来
- 2. 用例对虚拟网卡进行的操作结果不符合预期导致测试失败

# mugen 的 openEuler riscv 2403 测试 2403 的方案

现在基于 Jenkins CI 的 mugen 自动化测试框架研究带来了新的灵感：



1. 直接挂载 qcow2 镜像抓取日志
2. 使用 dnsmasq 构建网桥

这将解决所有 2309 测试遗留问题， qemu\_test.py 的修改正在进行。

请大家指导