

1. MocoTest-SysFault 测试工具使用手册

1. 工具安装
2. 配置
3. 测试
4. 多节点测试

MocoTest-SysFault 测试工具使用手册

工具安装

工具安装非常简单，解压后，将 `general_test_main`、`fuzzer`、`event_control`和 `strace`四个可执行文件所在的路径加入环境变量 `PATH`中。例如：

```
export PATH=解压路径:$PATH
```

主要运行脚本均在 `bin`目录中，在测试时应该进入该目录。

配置

工具需要知道如何启动应用以及如何调用应用的接口。

- 启动应用

工具通过 `get_configure.sh`得到程序的启动方式，该脚本需要将启动命令打印到标准输出上，一般情况下，直接输出启动命令即可，如

```
echo 'java -jar spring.jar' # spring.jar 为被测的Java程序
```

- 调用应用接口

工具运行时会执行 `client.sh`，用户可以在此脚本中加入接口调用，如

```
curl -XPOST http://localhost:8080/app/api
```

- 停止应用

一般情况下，工具会自动停止启动的应用，但如果工具无法关闭或应用启动后还启动了其他应用，还可以使用 `stop.sh` 脚本，用户可以在此脚本中加入停止命令，如

```
killall java
```

- 错误检查

应用如果异常停止，则工具会在 `log_test` 文件中打印 `"check failed!"` 日志。测试结束后，工具执行 `backup_test_case.sh` 脚本，将日志信息放入 `test_cases` 目录。

如果测试没有错误，该脚本会删除相关日志；如果发现了 `"check failed!"` 日志，则会保留。用户可以自行添加其他检查，例如，用户可以添加对应用日志的检查，如果发现相关日志，则同样保留日志，以待测试完成后查看。

测试

配置完成后，直接启动 `fuzz.sh` 即可。

测试过程中，需要根据日志观察应用是否启动正常，需要查看 `log_test_0_0` 和 `log_app_err_0_0` 等日志文件。注意如果应用程序配置了如何输出日志，则需要查看应用本身的日志。

测试过程中可以经常查看 `test_cases` 目录，观察是否有错误出现。测试一段时间后，如果长时间没有出现缺陷，则可以停止脚本，修改相关配置文件或检查日志。

多节点测试

如果应用支持多节点测试，则可以修改工具中的 `run_test.sh` 脚本，将脚本中的 `node_count` 参数改为需要的数量。多节点参数可以通过修改 `get_configure.sh` 进行配置，该脚本在测试时的第一个参数为节点序号，因此根据节点序号进行配置即可。