storm脚本：是一个python脚本

调用storm脚本时，执行其中的main方法，mian方法源代码如下：

```
def main():
    if len(sys.argv) <= 1:
        print_usage()
        sys.exit(-1)
    global CONFIG_OPTS
    config_list, args = parse_config_opts(sys.argv[1:])
    parse_config(config_list)
    COMMAND = args[0]
    ARGS = args[1:]
    (COMMANDS.get(COMMAND, unknown_command))(*ARGS)
```

main方法中有两个参数：

COMMAND = args[0]

ARGS = args[1:]

所有的COMMAND：

COMMANDS = {

"jar": jar, "kill": kill, "shell": shell, "nimbus": nimbus, "ui": ui, "logviewer": logviewer,
"drpc": drpc, "supervisor": supervisor, "localconfvalue": print_localconfvalue,
"remoteconfvalue": print_remoteconfvalue, "repl": repl, "classpath": print_classpath,
"activate": activate, "deactivate": deactivate, "rebalance": rebalance, "help": print_usage,
"list": listtopos, "dev-zookeeper": dev_zookeeper, "version": version, "monitor": monitor
}

执行命令，提交Topology任务到集群：

命令如下：

storm jar WordCount.jar com.stone.WordCountMain wordcount

脚本的执行流程如下：

（1）获得COMMAND = args[0]参数，COMMAND = "jar"
（2）在COMMANDS 中找到对应的命令jar
（3）调用def jar 方法

```
def jar(jarfile, klass, *args):
    exec_storm_class(
        klass,
        jvmtype="-client",
        extrajars=[jarfile, USER_CONF_DIR, STORM_DIR + "/bin"],
        args=args,
        jvmopts=JAR_JVM_OPTS + ["-Dstorm.jar=" + jarfile])
```

（4）在def jar方法中，调用`exec_storm_class`

```
def exec_storm_class(klass, jvmtype="-server", jvmopts=[], extrajars=[], args=[],
fork=False):
    global CONFFILE
    storm_log_dir = confvalue("storm.log.dir",[CLUSTER_CONF_DIR])
```

```
    if(storm_log_dir == None or storm_log_dir == "nil"):
        storm_log_dir = STORM_DIR+"/logs"
    all_args = [
        JAVA_CMD, jvmtype, get_config_opts(),
        "-Dstorm.home=" + STORM_DIR,
        "-Dstorm.log.dir=" + storm_log_dir,
        "-Djava.library.path=" + confvalue("java.library.path", extrajars),
        "-Dstorm.conf.file=" + CONFFILE,
        "-cp", get_classpath(extrajars),
    ] + jvmopts + [klass] + list(args)
    print("Running: " + " ".join(all_args))
    if fork:
        os.spawnvp(os.P_WAIT, JAVA_CMD, all_args)
    else:
        os.execvp(JAVA_CMD, all_args) # replaces the current process and
        # never returns
```

（5）在`exec_storm_class`中最终操作系统调用`os.spawnvp(os.P_WAIT, JAVA_CMD, all_args)`或者`os.execvp(JAVA_CMD, all_args)`完成命令的执行。

也就是执行：

storm jar WordCount.jar com.stone.WordCountMain wordcount

实际上是调用：

java -client WordCount.jar com.stone.WordCountMain wordcount