# 脚本分析报告

## 第五小组

刘浩龙、李嘉晨、莫泽威、祝鹏富、郑宜静

# 目录/Contents

01

# 脚本基本介绍

# 脚本基本介绍

build_chain.sh是FISCO BCOS给用户提供的用于快速搭建FISCO BCOS联盟链的脚本，用于快速生成一条链中节点的配置文件。
用户想快速体验可以使用-l命令选项指定节点IP和数目或者通过-f命令选项使用一个指定格式的配置文件，支持创建各种复杂业务场景的FISCO BCOS链。-l和-f命令选项必须指定一个且不可共存。

02

# 辅助类方法分析

## 基本变量

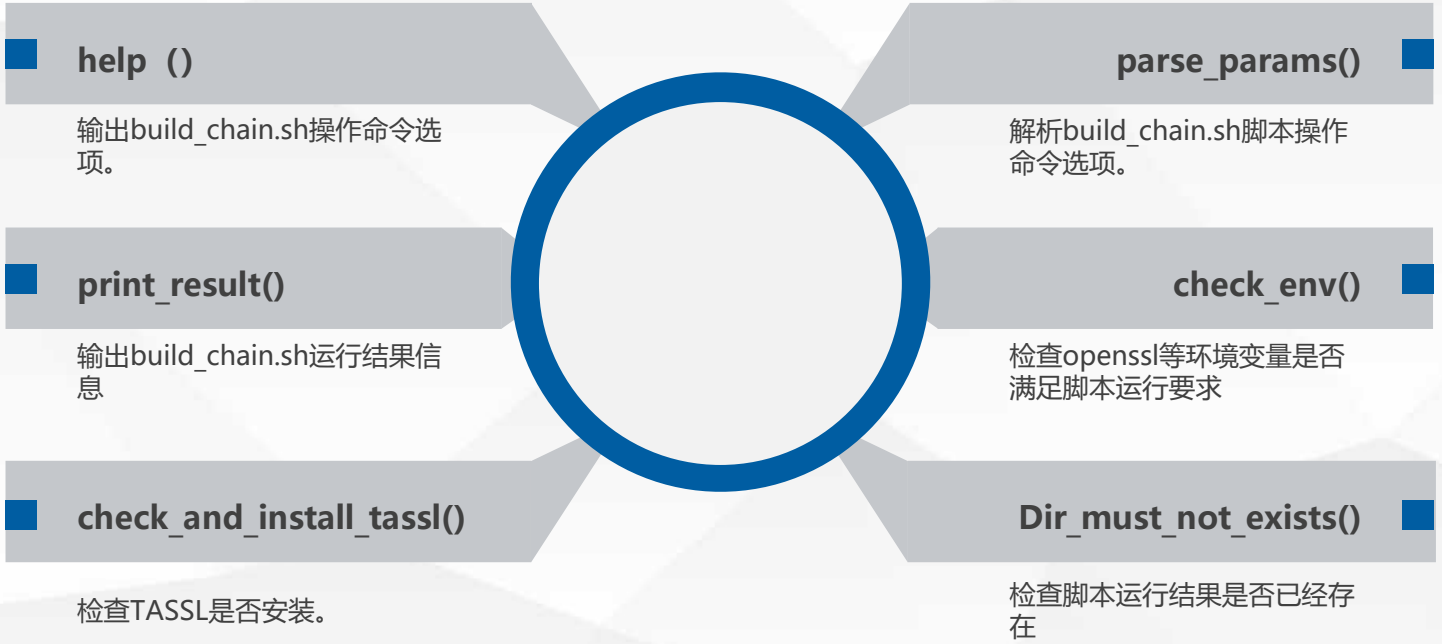
```
5 ca_file= #CA key
6 node_num=1
7 ip_file=
8 agency_array=
9 group_array=
10 ip_param=
11 use_ip_param=
12 ip_array=
13 output_dir=nodes
14 port_start=(30300 20200 8545)
15 state_type=storage
16 storage_type=LevelDB
17 conf_path="conf"
18 bin_path=
19 make_tar=
20 debug_log="false"
21 log_level="info"
22 logfile=build.log
23 listen_ip="127.0.0.1"
24 bcos_bin_name=fisco-bcos
25 guomi_mode=
26 docker_mode=
27 gm_conf_path="gmconf/"
28 current_dir=$(pwd)
29 consensus_type="pbft"
30 TASSL_CMD="${HOME}"/.tassl
31 auto_flush="true"
32 # trans timestamp from seconds to milliseconds
33 timestamp=$(($(date '+%s')*1000))
34 chain_id=1
35 fisco_version=""
36 OS=
```

## 默认值

代码的第5~36行显示了脚本里用到的各个变量，部分变量带有默认值，如节点数（node_num）、开始端口（port_start）、监听ip（127.0.0.1）等。

# 辅助相关方法分析




help（）
输出build_chain.sh操作命令选项。
parse_params()
解析build_chain.sh脚本操作命令选项。
print_result()
输出build_chain.sh运行结果信息
check_env()
检查openssl等环境变量是否满足脚本运行要求
check_and_install_tassl()
检查TASSL是否安装。
Dir_must_not_exists()
检查脚本运行结果是否已经存在

# 03

# 证书相关方法分析

# 证书相关方法分析

**联盟证书**

## gen_chain_cert()

生成联盟链证书。联盟链委员会使用openssl命令请求私钥ca.key，然后根据ca.key生成链证书ca.crt。

**机构证书**

## gen_agency_cert()

生成机构证书。机构使用openssl命令生成机构私钥agency.key，然后用机构私钥agency.key得到机构证书请求文件agency.csr，发送agency.csr给联盟链委员会，委员会使用ca.key根据agency.csr生成机构证书agency.crt并发送给对应机构。

**节点证书**

## gen_node_cert()

生成节点证书。节点生成私钥node.key和证书请求文件node.csr，机构管理员使用私钥agency.key和证书请求文件node.csr为节点/SDK颁发证书node.crt。

# 04

# 配置相关方法分析

generate_script_template()
- 生成脚本的模板，并导入头两行内容。

generate_node_scripts()
- 主要是利用generate_script_template()方法生成start.sh和stop.sh两个脚本。

genTransTest()
- 生成transTest.sh脚本。

generate_server_scripts()
- 生成server脚本，包括start_all.sh和stop_all.sh。

# 配置相关方法分析



主要生成每个节点的主配置文件

05

# 主函数分析

脚本会先根据命令选项-l或-f判断是使用命令输入的节点ip还是使用配置文件生成节点。

```
output_dir="$(pwd)/${output_dir}"
[ -z $use_ip_param ] && help 'ERROR: Please set -l or -f option.'
if [ "${use_ip_param}" == "true" ];then
    ip_array=(${ip_param//,/ })
elif [ "${use_ip_param}" == "false" ];then
    if ! parse_ip_config $ip_file ;then
        echo "Parse $ip_file error!"
        exit 1
    fi
else
    help
fi
```

随后检测输出文件目录是否存在，以及fisco版本情况，如果脚本运行结果已存在，会提示先删除旧的目录。

```
dir_must_not_exists ${output_dir}
mkdir -p "${output_dir}"

# get fisco_version
if [ -z "${fisco_version}" ];then
    fisco_version=$(curl -s https://raw.githubusercontent.com/FISCO-BCOS/F#
fi
```

然后检查运行环境，下载fisco-bcos。

```
# download fisco-bcos and check it
if [ -z ${docker_mode} ];then
    if [[ -z ${bin_path} && -z ${OS} ]];then
        bin_path=${output_dir}/${bcos_bin_name}
        package_name="fisco-bcos.tar.gz"
        [ ! -z "$guomi_mode" ] && package_name="fisco-bcos-gm.tar.gz"
        Download_Link="https://github.com/FISCO-BCOS/FISCO-BCOS/releases/d#
        LOG_INFO "Downloading fisco-bcos binary from ${Download_Link} ..."
        curl -LO ${Download_Link}
        tar -zxf ${package_name} && mv fisco-bcos ${bin_path} && rm ${pack#
        chmod a+x ${bin_path}
    elif [[ -z ${bin_path} && ! -z ${OS} ]];then
        echo "Please use docker mode to run fisco-bcos on macOS Or compile#
        exit 1
    else
        echo "Checking fisco-bcos binary..."
        bin_version=$(${bin_path} -v)
        if [ -z "$(echo ${bin_version} | grep 'FISCO-BCOS')" ];then
            LOG_WARN "${bin_path} is wrong. Please correct it and try agai#
            exit 1
        fi
        if [[ ! -z ${guomi_mode} && -z $(echo ${bin_version} | grep 'gm') #
            LOG_WARN "${bin_path} isn't gm version. Please correct it and #
            exit 1
        fi
        if [[ -z ${guomi_mode} && ! -z $(echo ${bin_version} | grep 'gm') #
            LOG_WARN "${bin_path} isn't standard version. Please correct i#
            exit 1
        fi
        echo "Binary check passed."
    fi
fi
```

再根据命令输入的节点信息或配置文件调用gen_chain_cert()方法生成链证书。

```
# prepare CA
echo "============================================================="
if [ ! -e "$ca_file" ]; then
    echo "Generating CA key..."
    dir_must_not_exists ${output_dir}/chain
    gen_chain_cert "" ${output_dir}/chain >${output_dir}/${logfile} 2>&1 |#
    mv ${output_dir}/chain ${output_dir}/cert
    if [ "${use_ip_param}" == "false" ];then
        for agency_name in ${agency_array[*]};do
            if [ ! -d ${output_dir}/cert/${agency_name} ];then
                gen_agency_cert "" ${output_dir}/cert ${output_dir}/cert/$#
            fi
        done
    else
        gen_agency_cert "" ${output_dir}/cert ${output_dir}/cert/agency >$#
    fi
    ca_file="${output_dir}/cert/ca.key"
fi
```

再调用gen_agency_cert()方法生成机构证书。

```
if [ -n "$guomi_mode" ]; then
    check_and_install_tassl

    generate_cert_conf_gm "gmcert.cnf"

    echo "Generating Guomi CA key..."
    dir_must_not_exists ${output_dir}/gmchain
    gen_chain_cert_gm "" ${output_dir}/gmchain >${output_dir}/build.log 2>&1
    mv ${output_dir}/gmchain ${output_dir}/gmcert
    gen_agency_cert_gm "" ${output_dir}/gmcert ${output_dir}/gmcert/agency >$
    ca_file="${output_dir}/gmcert/ca.key"****
fi
```

之后调用gen_node_cert()方法生成节点证书。

```
while :
do
    gen_node_cert "" ${output_dir}/cert/${agency_array[${server_count
    mkdir -p ${conf_path}/
    rm node.param node.private node.pubkey agency.crt
    mv *.* ${conf_path}/

    #private key should not start with 00
    cd ${output_dir}
    privateKey=$(openssl ec -in "${node_dir}/${conf_path}/node.key" -
    len=${#privateKey}
    head2=${privateKey:0:2}
    if [ "64" != "${len}" ] || [ "00" == "$head2" ];then
        rm -rf ${node_dir}
        continue;
    fi

    if [ -n "$guomi_mode" ]; then
        gen_node_cert_gm "" ${output_dir}/gmcert/agency ${node_dir} >
        mkdir -p ${gm_conf_path}/
        mv ./*.* ${gm_conf_path}/

        #private key should not start with 00
        cd ${output_dir}
        privateKey=$($TASSL_CMD ec -in "${node_dir}/${gm_conf_path}/g
        len=${#privateKey}
        head2=${privateKey:0:2}
        if [ "64" != "${len}" ] || [ "00" == "$head2" ];then
            rm -rf ${node_dir}
            continue;
        fi
    fi
    break;
done
```

最后调用print_result()方法输出结果信息。