

# QConf集中配置中心介绍

小林

2016.3.24

# 简介

基于zookeeper实现

配置实时更新到所有节点

服务器宕机、网络中断、集群迁移等都对用户使用透明

读取配置高效

本机测试：7W QPS

# 演示

usage: qconf command key [idc]

command: can be one of below commands:

version : get qconf version

get\_conf : get configure value

get\_host : get one service

get\_allhost : get all services available

get\_batch\_keys : get all children keys

key : the path of your configure items

idc : query from current idc if be omitted

example:

```
qconf get_conf "demo/conf"
```

```
qconf get_conf "demo/conf" "corp"
```

# 架构

Zookeeper



.....



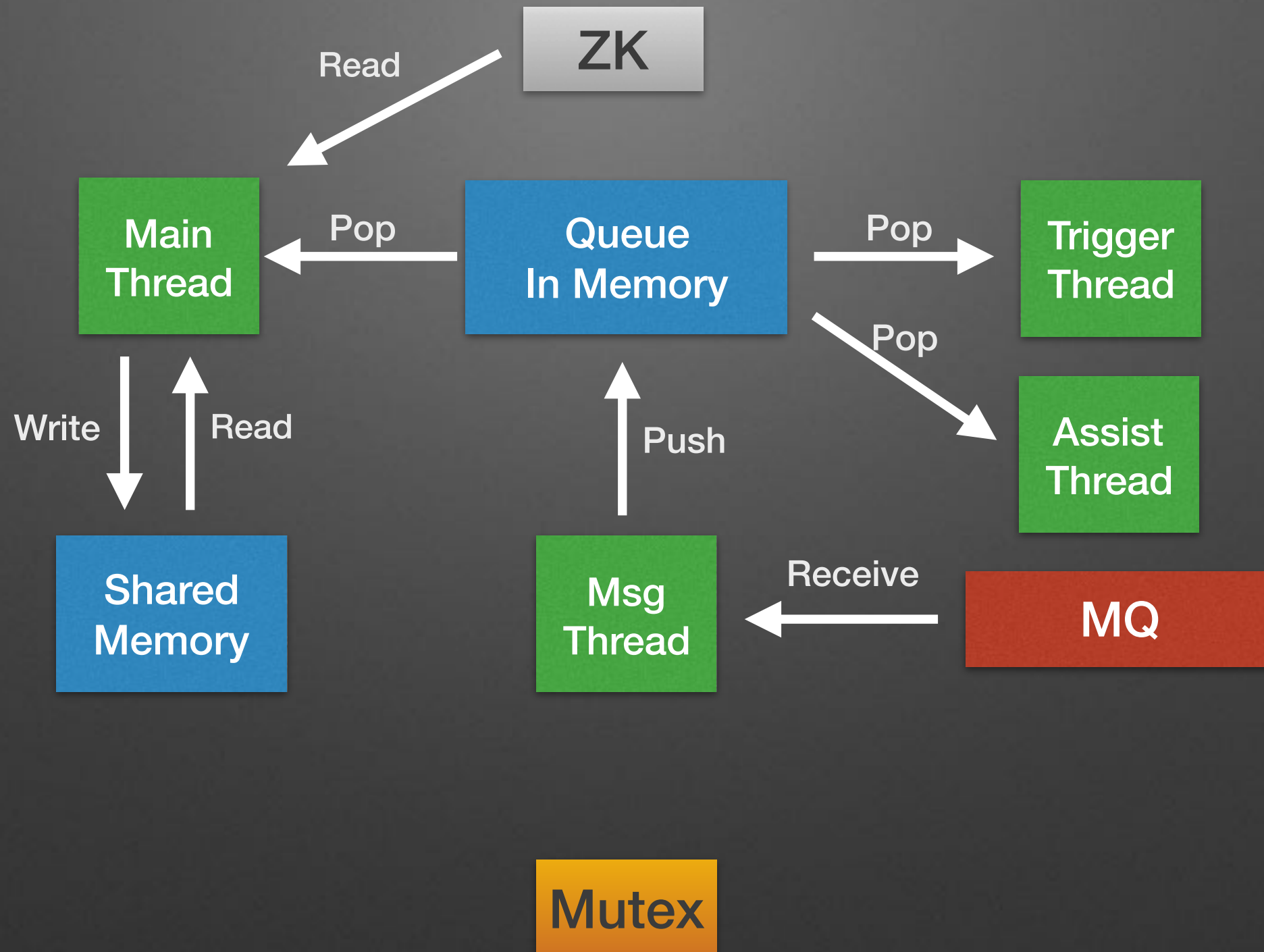
分布式

# QConf Agent

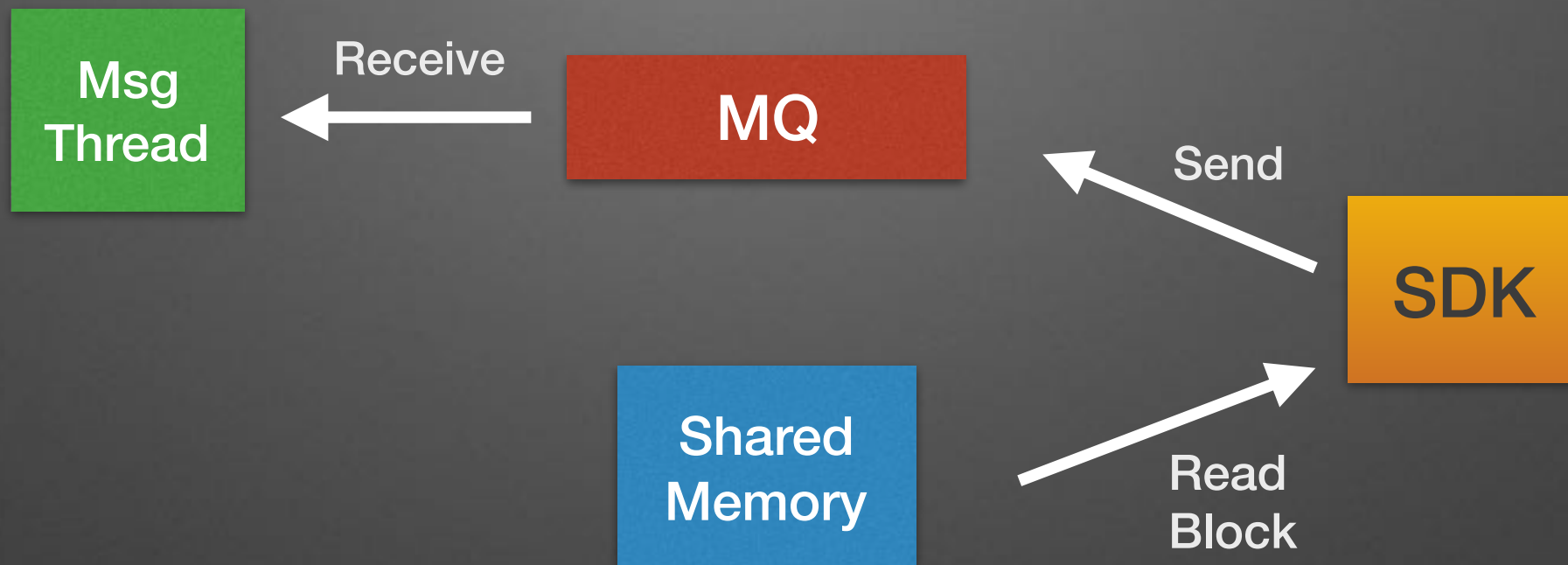
## System V IPC 机制

- Shared Memory
- Message Queues
- Semaphores

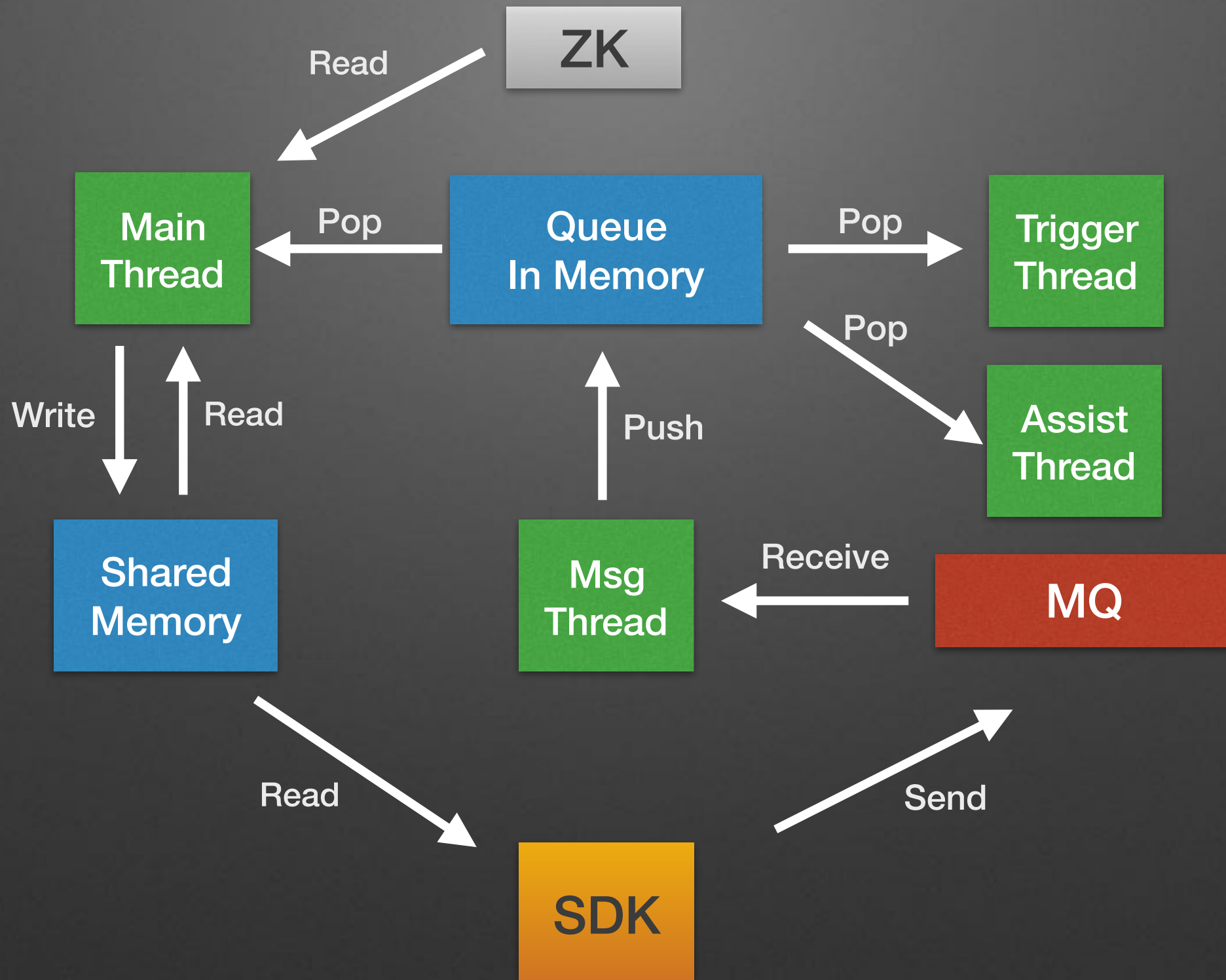
# QConf Agent



# QConf SDK



# QConf Arch





「源码面前，了无秘密」

侯捷

# 源码

## QConf

- CMakeLists.txt
- **agent** QConf Agent 源码
- **base** 公用库
- deps 编译依赖库, zookeeper、gdbm、curl等
- doc
- driver SDK源码
- manager
- test

# 源码-agent

```
agent
├── CMakeLists.txt
├── bin
│   └── agent-cmd.sh
├── conf
│   ├── agent.conf
│   ├── idc.conf
│   └── localidc
├── qconf_agent.cc
├── qconf_cmd.cc
├── qconf_cmd.h
├── qconf_config.cc
├── qconf_config.h
├── qconf_const.h
├── qconf_daemon.cc
├── qconf_daemon.h
├── qconf_dump.cc
├── qconf_dump.h
├── qconf_feedback.cc
├── qconf_feedback.h
├── qconf_gray.cc
├── qconf_gray.h
├── qconf_lock.cc
├── qconf_lock.h
├── qconf_script.cc
├── qconf_script.h
├── qconf_watcher.cc
├── qconf_watcher.h
├── qconf_zoo.cc
├── qconf_zoo.h
├── script
│   └── nginx_template.sh
```

# 源码-qconf\_agent.cc

```
int main(int argc, char **arg)
{
    // check whether agent is running
    .....
    // load configure
    ret = qconf_load_conf(agent_dir);
    .....
    // daemonize
    .....
    signal(SIGPIPE, SIG_IGN);
    signal(SIGINT, sig_handler);
    .....
    // Environment initialize
    ret = qconf_agent_init(agent_dir, log_dir);
    // main
    watcher_setting_start();
    qconf_agent_destroy();
    return QCONF_OK;
}
```

# 源码-qconf\_agent.cc

## qconf\_agent\_init

- qconf\_init\_dump\_file 初始化dump配置的环境
- qconf\_init\_shm\_tbl 初始化共享内存
- qconf\_init\_msg\_key 初始化消息队列

# 源码-qconf\_agent.cc

## watcher\_setting\_start

- assist\_watcher\_thread

定期扫描shm的所有key，保证key最新和监听事件的建立

- msg\_thread

将消息队列中的key读入到内存queue中

- main\_thread:deque\_process()

从内存queue中获取key，然后连接zk，获取最新值，写入shm

# 源码-SDK C++

## driver/c++

- CMakeLists.txt
- include
  - driver\_common.h
  - qconf.h
  - qconf\_errno.h
- src
  - driver\_api.cc
  - driver\_api.h
  - driver\_common.cc
  - qconf.cc

# 源码-SDK C++

## driver\_api.cc      qconf\_get\_

```
// get value from tbl
ret = hash_tbl_get(_qconf_hashtbl, tblkey, tblval);
if (QCONF_OK == ret) return ret;

// Not get batch keys from share memory, then send message to agent
int ret_snd = send_msg_to_agent(_qconf_msqid, tmp_idc, path, dtype);

// If not wait, then return directly
if (QCONF_NOWAIT == flags) return ret;

while (count < QCONF_MAX_GET_TIMES)
{
    usleep(5000);
    count++;

    ret = hash_tbl_get(_qconf_hashtbl, tblkey, tblval);
    if (QCONF_OK == ret)
    {
        LOG_ERR("Wait time:%d*5ms, type:%c, idc:%s, path:%s",
                count, dtype, tmp_idc.c_str(), path.c_str());
        return ret;
    }
}

if (count >= QCONF_MAX_GET_TIMES)
{
    LOG_FATAL_ERR("Failed to get value! wait time:%d*5ms, type:%c, idc:%s, path:%s, ret:%d",
        count, dtype, tmp_idc.c_str(), path.c_str(), ret);
}
```



# 后续工作

缺少一个可用的配置web界面(3天)

PHP项目配置代码需要修改(1天)

Q&A

# 参考资料

- Qconf
- System V IPC