【RAC】rac 中如何指定 job 的运行实例

1.1 BLOG 文档结构图

1.2 前言部分

1.2.1 导读和注意事项

各位技术爱好者，看完本文后，你可以掌握如下的技能，也可以学到一些其它你所不知道的知识，~O(∩_∩)O~：

① 如何指定 job 的运行实例（重点）

② 代码获取 rac 所有节点的 IP 地址

Tips：

① 若文章代码格式有错乱，推荐使用 QQ、搜狗或 360 浏览器，也可以下载 pdf 格式的文档来查看，pdf 文档下载地址：http://yunpan.cn/cdEQedhCs2kFz （提取码：ed9b）

② 本篇 BLOG 中命令的输出部分需要特别关注的地方我都用灰色背景和粉红色字体来表示，比如下边的例子中，thread 1 的最大归档日志号为 33，thread 2 的最大归档日志号为 43 是需要特别关注的地方；而命令一般使用黄色背景和红色字体标注；对代码或代码输出部分的注释一般采用蓝色字体表示。

```
  List of Archived Logs in backup set 11
  Thrd Seq     Low SCN    Low Time            Next SCN   Next Time
  ---- ------- ---------- ------------------- ---------- ---------
  1    32      1621589    2015-05-29 11:09:52 1625242    2015-05-29 11:15:48
  1    33      1625242    2015-05-29 11:15:48 1625293    2015-05-29 11:15:58
  2    42      1613951    2015-05-29 10:41:18 1625245    2015-05-29 11:15:49
  2    43      1625245    2015-05-29 11:15:49 1625253    2015-05-29 11:15:53



[ZFXXDB1:root]:/>lsvg -o
T_XDESK_APP1_vg
rootvg
[ZFXXDB1:root]:/>
00:27:22 SQL> alter tablespace idxtbs read write;


====》2097152*512/1024/1024/1024=1G
```

本文如有错误或不完善的地方请大家多多指正，ITPUB 留言或 QQ 皆可，您的批评指正是我写作的最大动力。

1.2.2 本文简介

记得之前写健康检查脚本的时候需要统计 rac 的 IP 地址，要用到一个包 utl_inaddr.GET_HOST_ADDRESS，这样的话就得去每个实例去查询，想了想只能是 job 来完成了，但当时并不知道如何在 rac 环境中指定相应的实例去运行 job，今天又特意研究了一下这个问题终于解决了，小麦苗迫不及待的分享给大家。

1.3 相关知识点扫盲(摘自网络)

oracle 自从 10g 开始有 2 种 job，dbms_job 和 DBMS_SCHEDULER，那么相应的就分 2 种情况下的指定实例了，先摘抄一点简单的 job 知识吧。DBMS_SCHEDULER 是 Oracle 10G 中新增的一个包，与老版本的 dbms_job 包相比，dbms_scheduler 有很多新特性。

所谓出于 job 而胜于 job，说的就是 Oracle10g 后的新特性 Scheduler 啦。在 10g 环境中，ORACLE 建议使用 Scheduler 替换普通的 job，来管理任务的执行。其实，将 Scheduler 描述成管理 job 的工具已经太过片面了，10G 版本中新增的 Scheduler 绝不仅仅是创建任务这么简单。。。。

ORACLE 中管理 Scheduler 是通过 DBMS_SCHEDULER 包。。。

DBMS_JOB 和 DBMS_SCHEDULER 之间的主要区别如下：

1. DBMS_SCHEDULER 可以执行存储的程序、匿名块以及 OS 可执行文件和脚本(包括 linux 系统的 shell 脚本)，而 DBMS_JOB 只可以执行存储的程序或匿名的 PL/SQL 块。
2. 考虑到增强的组件重用，调度程序的程序单元作为模式对象存储。DBMS_JOB 只有一种组件，即作业；而调度程序具有组件层次结构。
3. 可以使用 DBMS_SCHEDULER 更具描述性地定义作业或进度表间隔。DBMS_ SCHEDULER 也具有更详细的作业运行状态以及故障处理和报告功能。

1.4 dbms_job 下指定实例运行 job

1.4.1 相关知识简介(摘自网络)

一个 JOB 在何级别运行是可以定制的。如果把 job 定义在 db 级，job 可以运行在任何活动的 instance 上，并遵循 job 的调度机制；如果把 job 定义在 instance 级别上，job 将运行在指定的实例上，如因某种异常导致创建 job 的实例当机，

那 job 将运行在存活的实例上。

1、目前我们的 rac 数据库是通过查询语句 select job,instance,what from dba_jobs 可以看到 instance=0，这表示该 job 是 db 级，可以运行在任何活动的 instance 上，由 job 的调度机制决定在哪个实例上运行。也就是说 RAC 会根据两台服务器的运行状态来调度 JOB 在不同的节点实例中运行，一个 JOB 可以在 A 机，下一次有可能在 B 机运行；

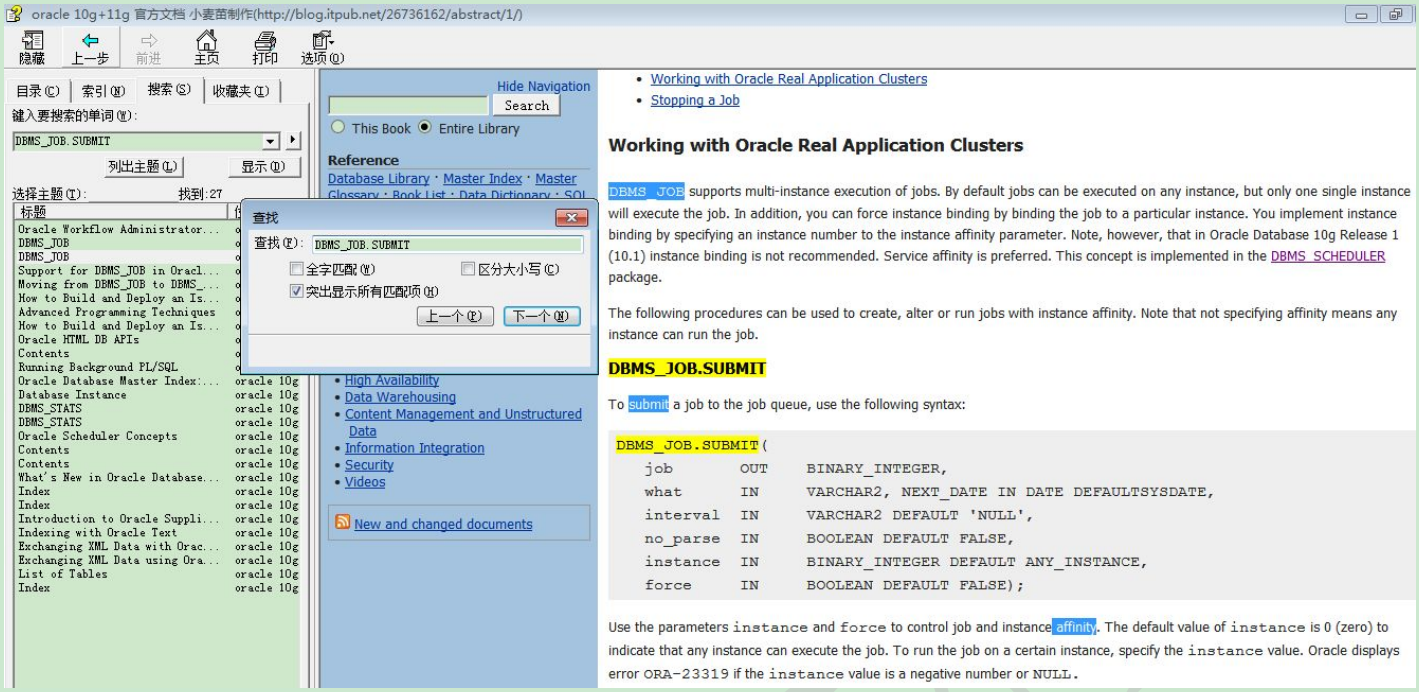
2、通过在调度中指定 instance 参数，可以指定 job 只在某个特定实例上运行，但是如果该实例的服务器出现故障时，发现 job 在实例 A 上不再运行，也不会切换到其它实例。如果 job 建立时没有指定运行在某个实例上，在 job 当前运行的实例关掉后，却可以切到其他活动的实例上。

```
var job number;
begin
  sys.dbms_job.submit(job => :job,
                      what => 'p_t1;',
                      next_date => to_date('27-03-2013 12:10:00', 'dd-mm-yyyy hh24:mi:ss'),
                      interval => 'trunc(sysdate,"mi")+1/24/60',
                      instance => 2);
  commit;
end;
/
```

指定实例编号

1.4.1.1 官方文档内容

利用小麦苗提供的工具搜索官方文档，看到如下的内容：



1.4.1.2 Working with Oracle Real Application Clusters

DBMS_JOB supports multi-instance execution of jobs. By default jobs can be executed on any instance, but only one single instance will execute the job. In addition, you can force instance binding by binding the job to a particular instance. You implement instance binding by specifying an instance number to the instance affinity parameter. Note, however, that in Oracle Database 10g Release 1 (10.1) instance binding is not recommended. Service affinity is preferred. This concept is implemented in the DBMS_SCHEDULER package.

The following procedures can be used to create, alter or run jobs with instance affinity. Note that not specifying affinity means any instance can run the job.

一、 DBMS_JOB.SUBMIT

To submit a job to the job queue, use the following syntax:

```
DBMS_JOB.SUBMIT(
   job       OUT    BINARY_INTEGER,
   what      IN     VARCHAR2, NEXT_DATE IN DATE DEFAULTSYSDATE,
   interval  IN     VARCHAR2 DEFAULT 'NULL',
   no_parse  IN     BOOLEAN DEFAULT FALSE,
   instance  IN     BINARY_INTEGER DEFAULT ANY_INSTANCE,
   force     IN     BOOLEAN DEFAULT FALSE);
```

Use the parameters *instance* and *force* to control job and instance affinity. The default value of *instance* is 0 (zero)

to indicate that any instance can execute the job. To run the job on a certain instance, specify the *instance* value.

Oracle displays error *ORA-23319* if the *instance* value is a negative number or *NULL*.

The *force* parameter defaults to *false*. If *force* is *TRUE*, any positive integer is acceptable as the job instance. If *force* is *FALSE*, the specified instance must be running, or Oracle displays error number *ORA-23428*.

二、 DBMS_JOB.INSTANCE

To assign a particular instance to execute a job, use the following syntax:

```
DBMS_JOB.INSTANCE(  JOB IN BINARY_INTEGER,
  instance                IN BINARY_INTEGER,
  force                   IN BOOLEAN DEFAULT FALSE);
```

The *FORCE* parameter in this example defaults to *FALSE*. If the instance value is 0 (zero), job affinity is altered and any available instance can execute the job despite the value of force. If the *INSTANCE* value is positive and the *FORCE* parameter is *FALSE*, job affinity is altered only if the specified instance is running, or Oracle displays error *ORA-23428*.

If the *force* parameter is *TRUE*, any positive integer is acceptable as the job instance and the job affinity is altered. Oracle displays error ORA-23319 if the *instance* value is negative or *NULL*.

说的还是比较详细的。

1.4.2 一个测试案例(获取所有节点的 IP 地址)

代码如下，`dbms_network_acl_admin` 主要是为了解决 ORA-24247 错误，很多年不做开发，写 plsql 的能力还是存在的，o(∩_∩)o ：

```
--- ORA-24247, "network access denied by access control list (ACL)"
begin
  dbms_network_acl_admin.drop_acl(acl => 'UTL_INADDR_LHR.xml');
  commit;
end;
/
begin
```

```
  dbms_network_acl_admin.create_acl(acl         => 'UTL_INADDR_LHR.xml',
                                    description => 'UTL_INADDR',
                                    principal   => 'MDSYS',
                                    is_grant    => TRUE,
                                    privilege   => 'resolve');
  commit;

  dbms_network_acl_admin.add_privilege(acl       => 'UTL_INADDR_LHR.xml',
                                       principal => 'MDSYS',
                                       is_grant  => TRUE,
                                       privilege => 'connect');
  commit;

   dbms_network_acl_admin.assign_acl(acl         => 'UTL_INADDR_LHR.xml',
                                     host        => '*');
  commit;

end;
/

drop table t_ipaddress_lhr;
create table t_ipaddress_lhr(INST_ID number, host_name varchar2(255), host_ip
varchar2(255) );

create or replace view vh_ipaddress_lhr as
SELECT a.HOST_ID || ': ' || a.HOST_ADDRESS host_ip1,
      a.HOST_ADDRESS host_ip2,
  a.HOST_ID host_name2
       FROM v$diag_alert_ext a
      WHERE a.COMPONENT_ID = 'rdbms'
        AND upper(a.FILENAME) =
            (SELECT upper(substr(d.VALUE, 1, (length(d.VALUE) - 5)) ||
                          'alert' || substr(d.VALUE, -6, 1) || 'log.xml')
               FROM v$parameter d
              WHERE d.NAME = 'background_dump_dest')
        and a.INDX =
            (SELECT max(b.INDX)
               FROM v$diag_alert_ext b
              WHERE b.COMPONENT_ID = 'rdbms'
                and upper(b.FILENAME) =
                    (SELECT upper(substr(d.VALUE, 1, (length(d.VALUE) - 5)) ||
                                  'alert' || substr(d.VALUE, -6, 1) ||
                                  'log.xml')
                       FROM v$parameter d
                      WHERE d.NAME = 'background_dump_dest'));
```

```
/

create or replace procedure ph_ip_lhr as

  V_SQL VARCHAR2(4000);
begin

  insert into t_ipaddress_lhr
    (inst_id, host_name)
    SELECT v.INSTANCE_NUMBER, v.HOST_NAME FROM v$instance v;
  commit;

  V_SQL := 'update t_ipaddress_lhr t
     set t.host_ip = utl_inaddr.GET_HOST_ADDRESS
   where t.inst_id = userenv(''instance'')';

  EXECUTE IMMEDIATE V_SQL;

  commit;

exception
  when others then

    V_SQL := 'update t_ipaddress_lhr t
       set t.host_ip =
           (SELECT v.host_ip2 FROM vh_ipaddress_lhr v)
     where t.inst_id = userenv(''instance'')';
    EXECUTE IMMEDIATE V_SQL;
    commit;

end ph_ip_lhr;
/

DECLARE
  X NUMBER;
begin

  for cur in (select b.JOB
                from dba_jobs b
               where b.WHAT = 'ph_ip_lhr;') loop

    sys.dbms_ijob.remove(cur.JOB);
    COMMIT;
  end loop;
```

```
  for cur in (select b.INST_ID from gv$instance b) loop

    SYS.DBMS_JOB.SUBMIT(job       => X,
                        what      => 'ph_ip_lhr;',
                        next_date => SYSDATE+cur.inst_id/8640,
                        INTERVAL  => 'null',
                        no_parse  => FALSE,
                        instance  => cur.inst_id);
    COMMIT;
  end loop;
END;
/
```

查看结果：

```
SELECT * FROM t_ipaddress_lhr;
```

	INST_ID	HOST_NAME	HOST_IP
1	2	ZFXDESKDB2	22.188.194.66
2	1	ZFXDESKDB1	22.188.194.64

1.5 DBMS_SCHEDULER 下指定实例运行 job

1.5.1 相关知识简介

1.5.1.1 JobClasses

JobClasses 相当于创建了一个 job 组，DBA 可以将那些具有相同特性的 job，统统放到相同的 JobClasses 中，然后通过对 JobClass 应用 ORACLE 中的"**资源使用计划**"特性，就可以对这些 job 执行过程中所需要的资源分配情况进行管理。

1、创建 JobClasses

使用 DBMS_SCHEDULER 包的 CREATE_JOB_CLASS 过程创建 JobClasses，该过程支持的参数如下，用 plsql developer 的命令行查看：

```
16:45:22 SQL> desc dbms_scheduler.create_job_class;
Parameter               Type           Mode Default?
```

```
------------------------ -------------- ---- --------
JOB_CLASS_NAME           VARCHAR2       IN
RESOURCE_CONSUMER_GROUP  VARCHAR2       IN   Y
SERVICE                  VARCHAR2       IN   Y
LOGGING_LEVEL            BINARY_INTEGER IN   Y
LOG_HISTORY              BINARY_INTEGER IN   Y
COMMENTS                 VARCHAR2       IN   Y
```

其中：

- JOB_CLASS_NAME：要创建的 JobClass 的名称，注意指定的长度不要超过 30 个字符，也不要与现有 JobClass 同名；
- RESOURCE_CONSUMER_GROUP：指定创建的 JobClass 所在的 RCG；

  提示：啥是 ResourceConsumerGroup

  你可以将其理解成一个资源分配的方式，处于相同 RCG 组中的用户、会话、或者对象共用一组资源，这组资源中可供分配的资源按照 DBA 指定的方式分配给 RCG。如果设计合理，通过这种方式，可以更有效的利用服务器的资源。
- SERVICE：指定创建的 JobClass 所在 Service，本选项常见于 RAC 环境，我们都知道 RAC 环境由多实例+数据库组成，此处所指定的 Service 实际就是指 JobClass 会在哪个实例上运行。

  注意：本参数与 RESOURCE_CONSUMER_GROUP 参数相互冲突，同一个 JobClass 只同设置两个参数中的一个值。
- LOGGING_LEVEL：指定日志记录的级别，有下列三种级别，是 DBMS_SCHEDULER 包中的三个常量：
  1. DBMS_SCHEDULER.LOGGING_OFF：关闭日志记录功能；
  2. DBMS_SCHEDULER.LOGGING_RUNS：对该 JobClass 下所有任务的运行信息进行记录；
  3. DBMS_SCHEDULER.LOGGING_FULL：记录该 JobClass 下任务的所有相关信息，不仅有任务运行情况，甚至连任务的创建、修改等也均将记入日志。

一、 官方文档对 service 的解释

service	This attribute specifies the database service that the jobs in this class have affinity to. In an Oracle RAC environment, this means that the jobs in this class only run on those database instances that are assigned to the specific service. Note that a service can be mapped to a resource consumer group, so you can also control resources allocated to jobs by specifying a service. See *DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING* for details. If both the *resource_consumer_group* and *service* attributes are specified, and if the service is mapped to a resource consumer group, the *resource_consumer_group* attribute takes precedence. If no service is specified, the job class belongs to the default service, which means it has no service affinity and any one of the database instances within the cluster might run the job. If the service that a job class belongs to is dropped, the job class will then belong to the default service. If the specified service does not exist when creating the job class, then an error occurs.

1.5.1.2 创建 service

基于 RAC 环境中使用的应用程序，有时候希望某个特定的应用程序仅仅运行在 RAC 的子节点，或者说为某些应用程序分配一个首要节点。对此，Oracle 可以使用 services 来实现。

这个具体可以参考官方文档：**Oracle® Database PL/SQL Packages and Types Reference 11*g* Release 2 (11.2)** E40758-03

1.5.2 一个测试案例(获取所有节点的 IP 地址)

```
--- ORA-24247, "network access denied by access control list (ACL)"
begin
  dbms_network_acl_admin.drop_acl(acl => 'UTL_INADDR_LHR.xml');
  commit;
end;
/
begin
  dbms_network_acl_admin.create_acl(acl         => 'UTL_INADDR_LHR.xml',
                                    description => 'UTL_INADDR',
                                    principal   => 'MDSYS',
                                    is_grant    => TRUE,
                                    privilege   => 'resolve');
  commit;

  dbms_network_acl_admin.add_privilege(acl       => 'UTL_INADDR_LHR.xml',
                                       principal => 'MDSYS',
                                       is_grant  => TRUE,
                                       privilege => 'connect');
  commit;

   dbms_network_acl_admin.assign_acl(acl         => 'UTL_INADDR_LHR.xml',
                                     host        => '*');
  commit;
```

```
end;
/

drop table t_ipaddress_lhr;
create table t_ipaddress_lhr(INST_ID number, host_name varchar2(255), host_ip
varchar2(255) );

create or replace view vh_ipaddress_lhr as
SELECT a.HOST_ID || ': ' || a.HOST_ADDRESS host_ip1,
      a.HOST_ADDRESS host_ip2,
  a.HOST_ID host_name2
       FROM v$diag_alert_ext a
      WHERE a.COMPONENT_ID = 'rdbms'
        AND upper(a.FILENAME) =
            (SELECT upper(substr(d.VALUE, 1, (length(d.VALUE) - 5)) ||
                          'alert' || substr(d.VALUE, -6, 1) || 'log.xml')
               FROM v$parameter d
              WHERE d.NAME = 'background_dump_dest')
        and a.INDX =
            (SELECT max(b.INDX)
               FROM v$diag_alert_ext b
              WHERE b.COMPONENT_ID = 'rdbms'
                and upper(b.FILENAME) =
                    (SELECT upper(substr(d.VALUE, 1, (length(d.VALUE) - 5)) ||
                                  'alert' || substr(d.VALUE, -6, 1) ||
                                  'log.xml')
                       FROM v$parameter d
                      WHERE d.NAME = 'background_dump_dest'));
/

create or replace procedure ph_ip_lhr as

  V_SQL VARCHAR2(4000);
begin

  insert into t_ipaddress_lhr
    (inst_id, host_name)
    SELECT v.INSTANCE_NUMBER, v.HOST_NAME FROM v$instance v;
  commit;

  V_SQL := 'update t_ipaddress_lhr t
     set t.host_ip = utl_inaddr.GET_HOST_ADDRESS
   where t.inst_id = userenv(''instance'')';
```

```
  EXECUTE IMMEDIATE V_SQL;

  commit;

exception
  when others then

    V_SQL := 'update t_ipaddress_lhr t
       set t.host_ip =
           (SELECT v.host_ip2 FROM vh_ipaddress_lhr v)
     where t.inst_id = userenv(''instance'')';
    EXECUTE IMMEDIATE V_SQL;
    commit;

end ph_ip_lhr;
/

begin

  for cur in (select v.INST_ID,
                     v.INSTANCE_NAME,
                     'INST_LHR_' || v.inst_id service_name,
                     'LHR_RAC' || v.INST_ID || '_JOB_CLASS' job_class_name,
                     'RAC_LHR_' || V.INST_ID JOB_NAME
                from gv$instance v) loop
    begin
      dbms_service.stop_service(service_name  => cur.service_name,
                                instance_name => cur.instance_name);
      dbms_service.delete_service(service_name => cur.service_name);
    exception
      when others then
        null;
    end;
    begin
      dbms_scheduler.drop_job_class(job_class_name => cur.job_class_name,
                                    force          => true);
    exception
      when others then
        null;
    end;
    begin
      dbms_scheduler.drop_job(job_name => CUR.JOB_NAME, force => TRUE);
    exception
      when others then
        null;
```

```
    end;
  end loop;

  for cur in (select v.INST_ID,
                     v.INSTANCE_NAME,
                     'INST_LHR_' || v.inst_id service_name,
                     'LHR_RAC' || v.INST_ID || '_JOB_CLASS' job_class_name,
                     'RAC_LHR_' || V.INST_ID JOB_NAME
                from gv$instance v) loop

    dbms_service.create_service(service_name => cur.service_name,
                                network_name => cur.service_name);
    dbms_service.start_service(service_name  => cur.service_name,
                               instance_name => cur.instance_name);

    dbms_scheduler.create_job_class(job_class_name => cur.job_class_name,
                                    service        => cur.service_name);

    DBMS_SCHEDULER.create_job(job_name        => 'RAC_LHR_' || CUR.INST_ID,
                              job_type        => 'STORED_PROCEDURE',
                              job_action      => 'ph_ip_lhr',
                              repeat_interval => NULL, --'FREQ=MINUTELY;INTERVAL=1'
                              job_class       => cur.job_class_name,
                              end_date        => NULL,
                              enabled         => TRUE);

  end loop;
end;
/
```

查看结果：

```
SELECT * FROM t_ipaddress_lhr;
```

	INST_ID	HOST_NAME	HOST_IP
1	2	ZFXDESKDB2	22.188.194.66
2	1	ZFXDESKDB1	22.188.194.64

1.6 总结

简单点：

① dbms_job 下指定实例运行 job 的方法是执行 SYS.DBMS_JOB.SUBMIT 包创建 job 的时候指定 instance 参数，

很简单

② DBMS_SCHEDULER 下指定实例运行 job 稍微有点复杂，创建 service，创建 job_class，然后创建 job 才可以，具体参考案例的代码。

About Me

.....

.....

本文作者：小麦苗，只专注于数据库的技术，更注重技术的运用

ITPUB BLOG：http://blog.itpub.net/26736162

本文地址：http://blog.itpub.net/26736162/viewspace-2072635/

本文pdf版：http://yunpan.cn/cdEQedhCs2kFz （提取码：ed9b）

QQ：642808185 若加 QQ 请注明您所正在读的文章标题

于 2016-03-30 09:00~ 2016-03-30 19:00 在中行完成

.....

.....