**TFS安装文档**

说明：

安装TFS时在Centos6.5环境下安装，其它版本不保证顺利安装。在安装时需要root超级用户来安装、并且关闭防火墙。要求以Centos6.5环境为准，并且以此文档为准,不保证其他环境能安装成功。

虚拟机要求有2块硬盘，一块为系统默认盘，另一块将被用作数据盘。

环境：

* + Centos 6.5 64
  + Automake
  + Libtool
  + Realine
  + Libz-devel
  + Uuid-devel
  + Tcmalloc google

VI详解：

|  |
| --- |
| http://wenku.baidu.com/link?url=lwNWeoLEG0UnzoMhdw7X2Vek0sNkIqAaRVk0Uh7CkB51yAbXXD8ZarQuIZjuKfGRtEoRXBwn5Yq5zF74EcA7SHIVs84ocI57Vpnid7ruFQ7 |

**安装依赖的软件包**

* 安装编译软件

|  |
| --- |
| yum install make gcc gcc-c++ |

* Automake TFS基于automake工具构建

|  |
| --- |
| yum install automake.noarch |

* libtool automake需要使用libtool

|  |
| --- |
| yum install libtool |

* realine 用于命令行编辑的库

|  |
| --- |
| yum install readline-devel |

* libz-devel 用于数据压缩/解压缩

|  |
| --- |
| yum install zlib-devel |

* uuid-devel 用于生成全局唯一ID

|  |
| --- |
| yum install e2fsprogs-devel  yum install libuuid-devel |

* tcmalloc google的内存管理库

|  |
| --- |
| wget http://gperftools.googlecode.com/files/gperftools-2.0.tar.gz  tar zvxf gperftools-2.0.tar.gz  cd gperftools-2.0  ./configure --enable-shared --enable-frame-pointers  make  make install |

**安装tb-common-utils**

TFS使用tb-common-utils软件包，tb-common-utils包含淘宝使用的基础系统库tbsys和网络库tbnet两个组件；安装tb-common-utils前需要设置环境变量TBLIB\_ROOT，tbsys和tbnet将会被安装TBLIB\_ROOT对应的路径（必须是绝对路径）下，TFS会在这个路径下查找tbsys、tbnet头文件和库。

设置TBLIB\_ROOT环境变量

|  |
| --- |
| 在vi ~/.bashrc文件中加入  export TBLIB\_ROOT=/usr/local/utils  export CLASSPATH=.:$TBLIB\_ROOT/lib:$CLASSPATH  export PATH=$PATH  然后执行source ~/.bashrc和 |

下载源码

|  |
| --- |
| 先安装subversion  yum install subversion  #svn co -r 18 http://code.taobao.org/svn/tb-common-utils/trunk tb-common-utils  注意： 这里不要checkout最新版本，version18以后的修改导致部分接口不能前向兼容。 |

编译安装

|  |
| --- |
| cd tb-common-utils/  chmod 777 \* -R  sh build.sh |

如果一切顺利，tb-common-utils已经安装成功到$TBLIB\_ROOT路径下；如遇到问题请先阅读后面的编译FAQ。

**安装TFS**

TFS开源用户大都只使用TFS的基本功能，所以这个版本我们默认只编译TFS的nameserver，dataserver，client和tool，以去除对mysql的依赖，需要使用到rcserver（全局资源管理服务），metaserver(TFS自定义文件名服务）的用户请自行编译安装这两个服务。

下载源码

|  |
| --- |
| svn co http://code.taobao.org/svn/tfs/branches/dev\_for\_outer\_users tfs |

编译安装

|  |
| --- |
| chmod 777 tfs -R  cd tfs/  sh build.sh init  ./configure --prefix=/usr/local/tfs --with-release  make  make install  cd /root/tb-common-utils/tfs/conf  cp \* /usr/local/tfs/conf |

--prefix 指定tfs安装路径，默认会被安装到~/tfs\_bin目录

--with-release 指定按release版本的参数进行编译，如果不指定这个参数，则会按开发版本比较严格的参数编译，包含-Werror参数，所有的警告都会被当错误，在高版本gcc下会导致项目编译不过，很多开源用户反馈的编译问题都跟这个有关，因为gcc高版本对代码的检查越来越严格。

至此，TFS已经安装成功了。

在成功安装TFS之后，在你的安装目录（默认为/usr/local/tfs)，包含几个子目录bin、conf、include、lib、logs、script等。

* bin：包含tfs所有的可执行程序文件，如nameserver(NS)、dataserver(DS)、tfstool。
* conf：包含tfs的配置文件，如NS的配置文件ns.conf，DS的配置文件ds.conf。
* include：包含TFS客户端库相关的头文件，应用程序使用TFS需要包含这些头文件。
* lib： 包含TFS客户端的静/动态库，应用程序使用TFS需要连接libtfsclient。
* logs：用于存储TFS运行过程中的日志。
* scripts：包含tfs常用的一些运维脚本，如stfs用于格式化DS， tfs启动/停止NS、DS。

配置tfs

vi /usr/local/tfs/conf/ns.conf修改

|  |  |
| --- | --- |
| work\_dir=/home/tfs  dev\_name= eth0（网卡名）   |  | | --- | | Ifconfig |   ip\_addr = 10.172.44.209（IP地址是网上面的IP地址）  ip\_addr\_list = 10.172.44.209 |192.168.0.2（第一项配置与ip\_addr相同，第二项可以随便配置一个无效的地址）  max\_replication = 1（单机环境测试，只能设置为1，否则不能存数据）  block\_max\_size = 75497472此参数与ds.conf配置文件的block\_max\_size = 75497472、mainblock\_size = 75497472参数一致 |

启动Nameserver

|  |
| --- |
| cd /usr/local/tfs  echo "/usr/local/lib" > /etc/ld.so.conf.d/usr\_local\_lib.conf  /sbin/ldconfig  ./scripts/tfs start\_ns |

如果没有提示错误，则NS就已经在后台开始运行了，可通过ps查看相应进行，或进入logs下，查看nameserver.log，如包含“nameserver running”则说明启动正常

查看方法：

|  |
| --- |
| ps aux    Logs下查看  cd /home/tfs/logs  cat nameserver.log |

挂载硬盘（虚拟机至少要有2块硬盘）：

|  |  |  |
| --- | --- | --- |
| fdisk -l  显示结果   |  | | --- | |  |   对数据盘进行分区，一块盘分一个区就行。  执行“fdisk /dev/sdb”命令，对数据盘进行分区；  根据提示，依次输入“n”，“p”“1”，两次回车，“wq”，分区就开始了，很快就会完成  -----------格式化----------  mkfs.ext4 /dev/sdb1  -----------挂载--------------  建好目录，第一块磁盘就挂载在第一个目录上。目录数据取决于你磁盘的数量。  mkdir -p /data/tfs1  mount /dev/sdb1 /data/tfs1  -----------加入启动时自动挂载----------------  设置开机自动挂载  [root@loctfs scripts]# vi /etc/fstab  加入以下信息在最后一行加入  /dev/sdb1 /data/tfs1 ext4 defaults 1 2  --------------重启系统----------------------------- 查看mount情况   |  | | --- | | /dev/mapper/VolGroup-lv\_root on / type ext4 (rw)  proc on /proc type proc (rw)  sysfs on /sys type sysfs (rw)  devpts on /dev/pts type devpts (rw,gid=5,mode=620)  tmpfs on /dev/shm type tmpfs (rw,rootcontext="system\_u:object\_r:tmpfs\_t:s0")  /dev/sda1 on /boot type ext4 (rw)  /dev/sdb1 on /data/tfs1 type ext4 (rw)  none on /proc/sys/fs/binfmt\_misc type binfmt\_misc (rw) | |

dataserver配置

vi /usr/local/tfs/conf/ds.conf修改

|  |
| --- |
| work\_dir=/usr/local/tfs  dev\_name= eth0  #ip addr  ip\_addr = 10.172.44.209  [dataserver]  #nameserver ip addr(vip)  ip\_addr = 10.172.44.209  ip\_addr\_list = 10.172.44.209|192.168.0.2  #nameserver port  port = 8100  backup\_path = /data/tfs  mount\_name = /data/tfs  # KB  mount\_maxsize = 9000000 |

------------存储区预分配------------------------------------

注意：挂载目录是/data/tfs1至/data/tfs(i)，其中i为磁盘号。

而ds.conf配置文件中的配置 mount\_name = /data/tfs 没有加（i）

执行scripts下的stfs format n (n为挂载点的序号，具体用法见stfs的Usage)。例如stfs format 2,4-6 则会对/data/tfs2,/data/tfs4,/data/tfs5,/data/tfs6，进行预分配。运行完后会在生成/data/tfs2, /data/tfs4,/data/tfs5,/data/tfs6下预先创建主块，扩展块及相应的统计信息。

|  |
| --- |
| cd /usr/local/tfs/scripts  ./stfs format 1 |

运行dataserver

|  |
| --- |
| cd /usr/local/tfs/scripts  ./tfs start\_ds 1  显示  dataserver 1 is up SUCCESSFULLY pid: 2563  一台服务器上启动一个dataserver进程     ./tfs stop\_ds 1 |

开始TFS存储

|  |  |  |  |
| --- | --- | --- | --- |
| 以下内容需要根据你的实际事情来操作。因为获得的存储TFS地址不一样。  cd /usr/local/tfs/bin  vi testfile创建testfile文件内容为  1  :wq保存退出  将本地testfile存入tfs，-s指定nameserver的ip:port。如果成功会打印put testfile => T19RETByJT1RCvBVdK success.类似的消息，其中T开头一串字符是TFS为这个文件生成的文件名，一共18个字符said从TFS取出该文件时，需要指定该文名。  ./tfstool -s 10.172.44.209:8100 -i "put testfile"   |  | | --- | |  |   从TFS读取T19RETByJT1RCvBVdK，存储到本地localfile中；如果成功会打印fetch T19RETByJT1RCvBVdK => localfile success.类似消息。  ./tfstool -s 10.172.44.209:8100 -i "get T19RETByJT1RCvBVdK localfile"   |  | | --- | |  |   删除文件  ./tfstool -s 10.172.44.209:8100 -i "rm T19RETByJT1RCvBVdK"   |  | | --- | |  |   说明：我这里删除的文件是tfs\_file所以（T19RETByxT1RCvBVdK）不一样。 |

编译FAQ

|  |
| --- |
| Q: 使用TFS一定需要64bit Liunx？  A: 是的，否则整个项目不能正常编译通过。  Q: 编译TFS过程中出现出现类似tbnet.h:39: fatal error: tbsys.h: No such file or directory的错误提示？  A: 需要先安装tb-common-utils软件包。  Q: 在安装tb-common-utils过程中，提示设置please set TBLIB\_ROOT varialbe first!!？  A: 需要先设置TBLIB\_ROOT环境变量，再编译安装tb-common-utils。  Q: 编译过程中出现类似警告：格式 ‘%lu’ 需要类型 ‘long unsigned int’，但实参 3 的类型为 ‘size\_t’ ？  A: 你的机器使用的应该是32bit OS，如果你坚持要编译，可以自行修改代码或者直接忽略这些警告。  Q: 执行./configure时，提示configure: error: readline header files not found, --disable-readline or install gnu readline library?  A: 你需要安装readline库，或在configure时加上--disable-readline参数，不使用readline库。  Q: 执行./configure时，提示configure: error: tcmalloc link failed (--without-tcmalloc to disable)？  A: 你需要安装tcmalloc库, 或在configure时加上--without-tcmalloc参数，不使用tcmalloc库。  Q: 编译过程中出现大量的错误信息，类似于warnings being treated as error？  A: 请确认在./configure的时候是否加了--with-release参数。  Q: 编译过程中提示类似client\_request\_server.cpp:722:38: error: no matching function for call to ‘atomic\_inc(volatile uint64\_t\*)’的错误信息？  A: 你的系统时32位的，请在64bit Linux编译安装TFS；因为32bit系统不支持对64bit整数的原子操作。  Q: 编译时遇到session\_util.cpp:2:23: fatal error: uuid/uuid.h: No such file or directory？  A: 你需要先安装uuid库, 用于生成全局唯一ID, uuid库是e2fsprogs包工具的一部分；  Q: 编译时提示tfs\_meta\_helper.cpp:15:18: fatal error: zlib.h: No such file or directory？  A: 你需要先安装zlib,用于压缩/解压缩。  Q: format TFS数据目录的过程中失败了？  A: 可能的原因：你对这个目录写权限；目录空间不足，先检查mount\_maxsize的配置；目录已经被格式化过，先clear，再format；磁盘换了； 具体错误，根据提示的errno来分析。  Q: 启动服务的过程中，提示ip 'xxxx' is not local ip, local ip: yyyy类似错误？  A: 确认dev\_name对应的网卡是否存在；ip\_addr的配置是否是该网卡上的ip地址。  Q: 通过tfstool存数据一致不能成功，错误码-5001？  A: 检查max\_replication的配置，机器数一定要大于副本数。  Q: ns.conf、ds.conf中为什么要配置ip\_addr\_list，而不是都直接使用vip？  A: 主备NS为了实现宕机切换后，备NS能够立即服务；主ns会异步将操作日志同步到备NS；同时所有的DS会同时向主备NS发心跳；这样备NS上拥有的信息与主NS基本相同（异步同步会有延时，做不到完全相同），可随时提供服务。基于此原因，nameserver必须知道备NS的real ip，DS也必须知道vip对应的主备NS的real ip。  Q: TFS HA怎么配置？  A: 这个让google回答你吧，淘宝HA的使用方式和配置可能并不适合你；tfs对HA软件及配置没有特殊要求，只要能实现主NS宕机vip飘移到备NS即可。  Q: 我部署好了TFS服务，有压力测试工具么，我想测试下性能？  A: TFS源代码下test/batch目录里有test\_batch\_write、test\_batch\_read、test\_batch\_mix三个工具。  示例：./test\_batch\_write -d ns-ip:port -c 1000 -r 10240:20480 -t 4  该命令代表往ns-ip:port对应的集群，开启4个线程，每个线程写1000个10K-20K范围内大小的文件。 |