首页 大大的技术(Michael) 小小的生活(Miumiu) 关于我们 留言板

站内搜索 搜索

micmiu – 软件开发+生活点滴

hadoop 2.2.0 集群模式安装配置和测试

作者: Michael 日期: 2014 年 1 月 22 日 查看评论 发表评论 (0)

本文详细记录Hadoop 2.2.0 集群安装配置的步骤,并运行演示一个简单的job。基本目录结构如下:

- 环境准备
- Hadoop安装配置
- 启动及演示

[一]、环境准备

本文所有集群节点的操作系统均为:CentOS 6.0 64位,不管是实体机还是虚拟机都可以,在这里统一叫做 "实例"吧,以4 台主机实例作为集群配置的演示,具体的划分如下:

hostname	IP	用途
Master.Hadoop	192.168.6.77	NameNode/ResouceManager
Slave5.Hadoop	192.168.8.205	DataNode/NodeManager
Slave6.Hadoop	192.168.8.206	DataNode/NodeManager
Slave7.Hadoop	192.168.8.207	DataNode/NodeManager

ps:如果是虚拟机可以把环境配置好后,copy多个实例即可,需要注意修改hostname

1、`vi /etc/hosts` 添加如下内容:

```
192.168.6.77    Master.Hadoop
192.168.8.205   Slave5.Hadoop
192.168.8.206   Slave6.Hadoop
192.168.8.207   Slave7.Hadoop
```

2、JDK

到Java 的官网下载jdk6 64位的版本,安装最基础的安装即可,当然由于CentOS6 自带了OpenJDK,本文直接用OpenJDK来演示(ps: OpenJDK的目录一般在`/usr/lib/jvm/` 路径下),该系统的JAVA_HOME 配置如下:`export JAVA_HOME = /usr/lib/jvm/java-1.6.0-openjdk.x86_64`

3、SSHD服务

确保系统已经安装了SSHD相关服务,并启动(CentOS默认已经安装好)。

4、创建用户

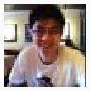
创建一个专用的账户:hadoop

```
$ useradd hadoop
```

5、配置SSH无密码登录

需要实现 Master到所有的Slave以及所有Slave 到Master的SSH无密码登录

微博



孙吴空的爸爸 上海 浦东新区

+加关注

@吴柚柚 //@电商行业: 【手机QQ也能发红包了! 来一发? 】7月15日-20日, 拍拍网联合手机QQ开展抢红包活动, 上亿全站通用无门槛代金券火热发放。京东接手运营半年时间, 拍拍网开始发力吸引客户了, 期待好体验!

拍拍网 V : 7月15日-20日, 拍拍网将联合手机QQ开展抢红包活动。上亿全站通用无门槛代金券火热发放, 小伙伴们速速来抢吧。今天, 小编就来告诉大家具体的攻略, 明天上午准时登陆手机QQ哦, 不

近期文章

- Spring AOP拦截Struts1记录操作日志
- Baby
- Hadoop2.x在Ubuntu系统中编译源码
- Ubuntu设置默认jdk
- Ubuntu12.04中Eclipse安装svn插件问题
- eclipse中开发Hadoop2.x的Map/Reduce项目
- Hadoop2.x eclipse plugin插件编译安装配置
- maven编译Spark源码
- java.net.ConnectException: to 0.0.0.0:10020 failed on connection exception
- ERROR tool.ImportTool: Imported Failed: Attempted to generate class with no columns!

浏览

- 图文介绍openLDAP在windows上的安装配置 - 12,237 views
- Thrift入门及Java实例演示 - 11,664 views
- SSO之CAS单点登录实例演示 - 11,648 views
- java keytool证书工具使用小结 - 9,516 views
- java自带的MD5、SHA1算法演示 - 9,217 views
- VirtualBox安装Ubuntu图文介绍 - 8,792 views
- Openfire服务端源代码开发配置指南 - 8,719 views
- Eclipse+Maven构建web项目及部署时Maven lib依赖问题的解决 - 8,147 views
- Mac OSX 10.8安装SVN的方法 - 7,050 views

有关SSH无密码登录的详细介绍可以参见:Linux(Centos)配置OpenSSH无密码登陆

6、配置时钟同步

```
$ cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
$ ntpdate us.pool.ntp.org
$ crontab -e
0-59/10 * * * * /usr/sbin/ntpdate us.pool.ntp.org | logger -t NTP
```

ps: 如果是实体机以上所有步骤需要在每个实例里都操作一遍;如果是虚拟机只需要一个实例中完成,其他实例复制即可。

[二]、Hadoop安装配置

1、下载源码编译本地库

由于官方的发布包中的本地库是32位的,不符合我们的要求,需要自己编译本地库,编译本地库的过程可以参考:Hadoop 2.x build native library on Mac os x ,大同小异,编译完成后,替换`<HADOOP_HOME>/lib/native/` 下的文件即可,注意lib文件名。

ps:这步只需要做一次即可,因为集群中的4个实例的环境一样。

2、下载发布包

打开官方下载链接 http://hadoop.apache.org/releases.html#Download ,选择2.2.0版本的发布包下载后解压到指定路径下:

```
$ tar -zxf hadoop-2.2.0.tar.gz -C /usr/local/
$ cd /usr/local
$ ln -s hadoop-2.2.0 hadoop
```

那么本文中 `HADOOP_HOME = /usr/local/hadoop/` .

3、配置hadoop用户的环境变量 `vi ~/.bash_profile` ,添加如下内容:

```
# set java environment
export JAVA_HOME=/usr/lib/jvm/java-1.6.0-openjdk.x86_64
export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
export PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin

# Michael@micmiu.com
# Hadoop
export HADOOP_PREFIX="/usr/local/hadoop"
export PATH=$PATH:$HADOOP_PREFIX/bin:$HADOOP_PREFIX/sbin
export HADOOP_COMMON_HOME=${HADOOP_PREFIX}
export HADOOP_HDFS_HOME=${HADOOP_PREFIX}
export HADOOP_MAPRED_HOME=${HADOOP_PREFIX}
export HADOOP_YARN_HOME=${HADOOP_PREFIX}
```

4、编辑 `<HADOOP_HOME>/etc/hadoop/hadoop-env.sh`

修改JAVA_HOME的配置:

```
export JAVA_HOME=/usr/lib/jvm/java-1.6.0-openjdk.x86_64
```

5、编辑 `<HADOOP_HOME>/etc/hadoop/yarn-env.sh`

修改JAVA_HOME的配置:

```
export JAVA_HOME=/usr/lib/jvm/java-1.6.0-openjdk.x86_64
```

6、编辑 `<HADOOP_HOME>/etc/hadoop/core-site.xml`

java.lang.OutOfMemoryError: GC overhead limit exceeded - 6,505 views

分类目录

- J2EE (41)
  - Hibernate (12)
  - Struts (5)
  - Spring (11)
  - JDBC (6)
  - JTA (3)
  - EJB (4)
- 海量数据 (28)
  - Hadoop (12)
  - Hive (8)
  - HBase (4)
  - Spark (2)
  - Sqoop (2)
- SOA (14)
  - webservice (12)
  - RPC (2)
- OpenSource (27)
  - openflashchart (2)
  - 生成文档 (5)
  - Openfire (4)
  - Security (1)
  - CORBA (4)
- 架构设计 (7)
  - 集群 (6)
  - 缓存 (1)
- 企业应用 (19)
  - SNMP (8)
  - SSO (4)
  - 服务器软件 (4)
- 编程语言 (34)
  - Java (24)
  - javascript (5)
  - Groovy (4)
  - nodejs (1)
- NoSQL (12)
  - Berkeley (6)
- 数据库 (10)
  - Oracle (7)
  - mysql (2)
- Android (2)
- web UI (1)
- 软件工具 (32)
  - 常用软件 (1)
  - 项目构建 (15)
  - 测试工具 (2)
- 操作系统 (21)
  - linux (15)
  - windows (1)
  - Mac (5)
- 小小 (12)
  - 心底软绵绵 (4)
  - 时间静悄悄 (5)
  - 肚皮圆滚滚 (3)

在<configuration>节点下添加或者更新下面的配置信息:

```
<!-- 新变量f:s.defaultFS 代替旧的:fs.default.name |micmiu.com-->
<property>
    <name>fs.defaultFS</name>
    <value>hdfs://Master.Hadoop:9000</value>
    <description>The name of the default file system.</description>
</property>
<property>
    <name>hadoop.tmp.dir</name>
    <!-- 注意创建相关的目录结构 -->
    <value>/usr/local/hadoop/temp</value>
    <description>A base for other temporary directories.</description>
</property>
```

7、编辑`<HADOOP_HOME>/etc/hadoop/hdfs-site.xml`

在<configuration>节点下添加或者更新下面的配置信息:

```
<property>
    <name>dfs.replication</name>
    <!-- 值需要与实际的DataNode节点数要一致,本文为3 -->
    <value>3</value>
    </property>
<property>
    <name>dfs.namenode.name.dir</name>
    <!-- 注意创建相关的目录结构 -->
    <value>file:/usr/local/hadoop/dfs/name</value>
    <final>true</final>
</property>
<property>
    <name>dfs.datanode.data.dir</name>
     <!-- 注意创建相关的目录结构 -->
    <value>file:/usr/local/hadoop/dfs/data</value>
</property>
```

8、编辑`<HADOOP_HOME>/etc/hadoop/yarn-site.xml`

在<configuration>节点下添加或者更新下面的配置信息:

```
<!-- micmiu.com -->
<property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
</property>
<property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

<!--  resourcemanager hostname或ip地址-->
<property>
    <name>yarn.resourcemanager.hostname</name>
    <value>Master.Hadoop</value>
</property>
```

9、编辑 `<HADOOP_HOME>/etc/hadoop/mapred-site.xml`

默认没有mapred-site.xml文件,copy mapred-site.xml.template 一份为 mapred-site.xml即可

在<configuration>节点下添加或者更新下面的配置信息:

```
<!-- micmiu.com -->
<property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
        <final>true</final>
</property>
```

[三**]**、启动和测试

1、启动**Hadoop**

- 异常处理 (19)
- 杂谈 (2)

文章归档

- 2014年七月 (1)
- 2014年五月 (1)
- 2014年四月 (12)
- 2014年三月 (4)
- 2014年二月 (6)
- 2014年一月 (8)
- 2013年十二月 (1)
- 2013年十月 (9)
- 2013年九月 (7)
- 2013年八月 (8)
- 2013年六月 (4)
- 2013年五月 (4)
- 2013年四月 (3)
- 2013年三月 (10)
- 2013年二月 (1)
- 2013年一月 (2)
- 2012年十二月 (1)
- 2012年十一月 (7)
- 2012年十月 (2)
- 2012年九月 (5)
- 2012年八月 (13)
- 2012年七月 (14)
- 2012年六月 (12)
- 2012年五月 (16)
- 2012年四月 (20)
- 2012年三月 (22)
- 2012年二月 (88)

标签云(3D)

技术链接

- 10→me@iteye
- 21→nosqlfan
- 22→mysqlops
- 23→OpenCloudDB

1.1、第一次启动需要在Master.Hadoop 执行format `hdfs namenode -format` :

```
[hadoop@Master ~]$ hdfs namenode -format
14/01/22 15:43:10 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = Master.Hadoop/192.168.6.77
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.2.0
STARTUP_MSG:   classpath =
........................................
...........micmiu.com............
........................................
STARTUP_MSG:   java = 1.6.0_20
************************************************************/
14/01/22 15:43:10 INFO namenode.NameNode: registered UNIX signal handlers for [TERM, HUP, INT]
Formatting using clusterid: CID-645f2ed2-6f02-4c24-8cbc-82b09eca963d
14/01/22 15:43:11 INFO namenode.HostFileManager: read includes:
HostSet(
)
14/01/22 15:43:11 INFO namenode.HostFileManager: read excludes:
HostSet(
)
14/01/22 15:43:11 INFO blockmanagement.DatanodeManager: dfs.block.invalidate.limit=1000
14/01/22 15:43:11 INFO util.GSet: Computing capacity for map BlocksMap
14/01/22 15:43:11 INFO util.GSet: VM type       = 64-bit
14/01/22 15:43:11 INFO util.GSet: 2.0% max memory = 888.9 MB
14/01/22 15:43:11 INFO util.GSet: capacity      = 2^21 = 2097152 entries
14/01/22 15:43:11 INFO blockmanagement.BlockManager: dfs.block.access.token.enable=false
14/01/22 15:43:11 INFO blockmanagement.BlockManager: defaultReplication         = 3
14/01/22 15:43:11 INFO blockmanagement.BlockManager: maxReplication             = 512
14/01/22 15:43:11 INFO blockmanagement.BlockManager: minReplication             = 1
14/01/22 15:43:11 INFO blockmanagement.BlockManager: maxReplicationStreams      = 2
14/01/22 15:43:11 INFO blockmanagement.BlockManager: shouldCheckForEnoughRacks  = false
14/01/22 15:43:11 INFO blockmanagement.BlockManager: replicationRecheckInterval = 3000
14/01/22 15:43:11 INFO blockmanagement.BlockManager: encryptDataTransfer        = false
14/01/22 15:43:11 INFO namenode.FSNamesystem: fsOwner             = hadoop (auth:SIMPLE)
14/01/22 15:43:11 INFO namenode.FSNamesystem: supergroup          = supergroup
14/01/22 15:43:11 INFO namenode.FSNamesystem: isPermissionEnabled = true
14/01/22 15:43:11 INFO namenode.FSNamesystem: HA Enabled: false
14/01/22 15:43:11 INFO namenode.FSNamesystem: Append Enabled: true
14/01/22 15:43:11 INFO util.GSet: Computing capacity for map INodeMap
14/01/22 15:43:11 INFO util.GSet: VM type       = 64-bit
14/01/22 15:43:11 INFO util.GSet: 1.0% max memory = 888.9 MB
14/01/22 15:43:11 INFO util.GSet: capacity      = 2^20 = 1048576 entries
14/01/22 15:43:11 INFO namenode.NameNode: Caching file names occuring more than 10 times
14/01/22 15:43:11 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
14/01/22 15:43:11 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
14/01/22 15:43:11 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension     = 30000
14/01/22 15:43:11 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
14/01/22 15:43:11 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
14/01/22 15:43:11 INFO util.GSet: Computing capacity for map Namenode Retry Cache
14/01/22 15:43:11 INFO util.GSet: VM type       = 64-bit
14/01/22 15:43:11 INFO util.GSet: 0.029999999329447746% max memory = 888.9 MB
14/01/22 15:43:11 INFO util.GSet: capacity      = 2^15 = 32768 entries
14/01/22 15:43:11 INFO common.Storage: Storage directory /usr/local/hadoop/dfs/name has been successfully formatted.
14/01/22 15:43:11 INFO namenode.FSImage: Saving image file /usr/local/hadoop/dfs/name/current/fsimage.ckpt_0000000000000000000 using no compression
14/01/22 15:43:11 INFO namenode.FSImage: Image file /usr/local/hadoop/dfs/name/current/fsimage.ckpt_0000000000000000000 of size 198 bytes saved in 0 seconds.
```

- 31→CrazyJvm
- 32→封神无度
- 33→董的博客

功能

- 登录
- 文章RSS
- 评论RSS
- WordPress.org

```
14/01/22 15:43:11 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
14/01/22 15:43:11 INFO util.ExitUtil: Exiting with status 0
14/01/22 15:43:11 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at Master.Hadoop/192.168.6.77
************************************************************/
```

1.2、在Master.Hadoop执行 `start-dfs.sh` :

```
[hadoop@Master ~]$ start-dfs.sh
Starting namenodes on [Master.Hadoop]
Master.Hadoop: starting namenode, logging to /usr/local/hadoop-2.2.0/logs/hadoop-hadoop-namenode-Master.Hadoop.out
Slave7.Hadoop: starting datanode, logging to /usr/local/hadoop-2.2.0/logs/hadoop-hadoop-datanode-Slave7.Hadoop.out
Slave5.Hadoop: starting datanode, logging to /usr/local/hadoop-2.2.0/logs/hadoop-hadoop-datanode-Slave5.Hadoop.out
Slave6.Hadoop: starting datanode, logging to /usr/local/hadoop-2.2.0/logs/hadoop-hadoop-datanode-Slave6.Hadoop.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop-2.2.0/logs/hadoop-hadoop-secondarynamenode-Master.Hadoop.out
```

在Master.Hadoop 验证启动进程:

```
[hadoop@Master ~]$ jps
7695 Jps
7589 SecondaryNameNode
7403 NameNode
```

在SlaveX.Hadop 验证启动进程如下:

```
[hadoop@Slave5 ~]$ jps
8724 DataNode
8815 Jps
```

1.3、在Master.Hadoop 执行 `start-yarn.sh` :

```
[hadoop@Master ~]$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop-2.2.0/logs/yarn-hadoop-resourcemanager-Master.Hadoop.out
Slave7.Hadoop: starting nodemanager, logging to /usr/local/hadoop-2.2.0/logs/yarn-hadoop-nodemanager-Slave7.Hadoop.out
Slave5.Hadoop: starting nodemanager, logging to /usr/local/hadoop-2.2.0/logs/yarn-hadoop-nodemanager-Slave5.Hadoop.out
Slave6.Hadoop: starting nodemanager, logging to /usr/local/hadoop-2.2.0/logs/yarn-hadoop-nodemanager-Slave6.Hadoop.out
```

在Master.Hadoop 验证启动进程:

```
[hadoop@Master ~]$ jps
8071 Jps
7589 SecondaryNameNode
7821 ResourceManager
7403 NameNode
```

在SlaveX.Hadop 验证启动进程如下:

```
[hadoop@Slave5 ~]$ jps
9013 Jps
8724 DataNode
8882 NodeManager
```

2、演示

2.1、演示hdfs 一些常用命令,为wordcount演示做准备:

```
[hadoop@Master ~]$ hdfs dfs -ls /
[hadoop@Master ~]$ hdfs dfs -mkdir /user
[hadoop@Master ~]$ hdfs dfs -mkdir -p /user/micmiu/wordcount/in
[hadoop@Master ~]$ hdfs dfs -ls /user/micmiu/wordcount
Found 1 items
```

```
drwxr-xr-x   - hadoop supergroup          0 2014-01-22 16:01
/user/micmiu/wordcount/in
```

2.2、本地创建三个文件 micmiu-01.txt、micmiu-03.txt、micmiu-03.txt, 分别写入如下内容:

micmiu-01.txt:

```
Hi Michael welcome to Hadoop
more see micmiu.com
```

micmiu-02.txt:

```
Hi Michael welcome to BigData
more see micmiu.com
```

micmiu-03.txt:

```
Hi Michael welcome to Spark
more see micmiu.com
```

把 micmiu 打头的三个文件上传到hdfs:

```
[hadoop@Master ~]$ hdfs dfs -put micmiu*.txt /user/micmiu/wordcount/in
[hadoop@Master ~]$ hdfs dfs -ls /user/micmiu/wordcount/in
Found 3 items
-rw-r--r--   3 hadoop supergroup         50 2014-01-22 16:06
/user/micmiu/wordcount/in/micmiu-01.txt
-rw-r--r--   3 hadoop supergroup         50 2014-01-22 16:06
/user/micmiu/wordcount/in/micmiu-02.txt
-rw-r--r--   3 hadoop supergroup         49 2014-01-22 16:06
/user/micmiu/wordcount/in/micmiu-03.txt
```

2.3、然后cd 切换到Hadoop的根目录下执行:

```
hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.2.0.jar
wordcount  /user/micmiu/wordcount/in /user/micmiu/wordcount/out
```

ps: hdfs 中 /user/micmiu/wordcount/out 目录不能存在 否则运行报错。

看到类似如下的日志信息:

```
[hadoop@Master hadoop]$ hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-
examples-2.2.0.jar wordcount  /user/micmiu/wordcount/in
/user/micmiu/wordcount/out
14/01/22 16:36:28 INFO client.RMProxy: Connecting to ResourceManager at
Master.Hadoop/192.168.6.77:8032
14/01/22 16:36:29 INFO input.FileInputFormat: Total input paths to process : 3
14/01/22 16:36:29 INFO mapreduce.JobSubmitter: number of splits:3
............................
.....micmiu.com........
............................
File System Counters
                FILE: Number of bytes read=297
                FILE: Number of bytes written=317359
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=536
                HDFS: Number of bytes written=83
                HDFS: Number of read operations=12
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=3
                Launched reduce tasks=1
                Data-local map tasks=3
                Total time spent by all maps in occupied slots (ms)=55742
                Total time spent by all reduces in occupied slots (ms)=3933
        Map-Reduce Framework
                Map input records=6
                Map output records=24
                Map output bytes=243
                Map output materialized bytes=309
                Input split bytes=387
                Combine input records=24
                Combine output records=24
                Reduce input groups=10
                Reduce shuffle bytes=309
                Reduce input records=24
                Reduce output records=10
                Spilled Records=48
                Shuffled Maps =3
                Failed Shuffles=0
                Merged Map outputs=3
                GC time elapsed (ms)=1069
                CPU time spent (ms)=12390
                Physical memory (bytes) snapshot=846753792
                Virtual memory (bytes) snapshot=5155561472
                Total committed heap usage (bytes)=499580928
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=149
        File Output Format Counters
                Bytes Written=83
```

到此 wordcount的job已经执行完成,执行如下命令可以查看刚才job的执行结果:

```
[hadoop@Master hadoop]$ hdfs dfs -ls /user/micmiu/wordcount/out
Found 2 items
-rw-r--r--   3 hadoop supergroup          0 2014-01-22 16:38 /user/micmiu/wordcount/out/_SUCCESS
-rw-r--r--   3 hadoop supergroup         83 2014-01-22 16:38 /user/micmiu/wordcount/out/part-r-00000
[hadoop@Master hadoop]$ hdfs dfs -cat /user/micmiu/wordcount/out/part-r-00000
BigData 1
Hadoop  1
Hi      3
Michael 3
Spark   1
micmiu.com      3
more    3
see     3
to      3
welcome 3
```

打开浏览器输入:http://192.168.6.77(Master.Hadoop):8088 可查看相关的应用运行情况。

———— EOF @Michael Sun ————

原创文章,转载请注明: 转载自***micmiu – 软件开发+生活点滴[http://www.micmiu.com/]***

本文链接地址***: http://www.micmiu.com/bigdata/hadoop/hadoop2x-cluster-setup/***

Hadoop   Hadoop

← hadoop 2.2.0 单节点安装和测试

Hive安装配置详解 →

发表评论?

0 条评论。

发表评论

昵称

邮箱

网址

注意 - 你可以用以下 HTML tags and attributes:

```
<a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote
cite=""> <cite> <code> <del datetime=""> <em> <i> <q cite=""> <strike>
<strong>
```

提交

↑ 回到顶部