

Spark安装配置

[Spark安装配置](#)

[下载](#)

[环境配置](#)

[多个节点的集群环境配置](#)

[运行测试](#)

[在yarn上的运行测试这里不再记录，请自行测试](#)

下载

进入[Apache Spark官网](#)的[下载页](#)，截止我写这篇文章的日期，spark的最新版本为2.0.0，因为我的hadoop版本是2.6.4的，所以我选择spark的版本是2.0.0，Hadoop是2.6。



Download Libraries Documentation Examples Community FAQ Apache Software Foundation

Download Apache Spark™

Our latest stable version is Apache Spark 2.0.1, released on Oct 3, 2016 ([release notes](#)) ([git tag](#))

- Choose a Spark release: 2.0.0 (Jul 26 2016)
- Choose a package type: Pre-built for Hadoop 2.6
- Choose a download type: Direct Download
- Download Spark: spark-2.0.0-bin-hadoop2.6.tgz
- Verify this release using the [2.0.0 signatures and checksums](#) and [project release KEYS](#).

Note: Starting version 2.0, Spark is built with Scala 2.11 by default. Scala 2.10 users should download the Spark source package and build with Scala 2.10 support.

Link with Spark

Spark artifacts are [hosted in Maven Central](#). You can add a Maven dependency with the following coordinates:

```
groupId: org.apache.spark
artifactId: spark-core_2.11
version: 2.0.1
```

Spark Source Code Management

Latest News

- Spark 2.0.1 released (Oct 03, 2016)
- Spark 2.0.0 released (Jul 26, 2016)
- Spark 1.6.2 released (Jun 25, 2016)
- Call for Presentations for Spark Summit EU is Open (Jun 16, 2016)

[Archive](#)

Download Spark

Built-in Libraries:

- [SQL and DataFrames](#)
- [Spark Streaming](#)
- [MLlib \(machine learning\)](#)
- [GraphX \(graph\)](#)

Third-Party Packages

环境配置

集群环境：

主机名	IP	hadoop环境	Scala环境	Java环境
master	192.168.146.146	/usr/local/hadoop-2.6.4	/usr/local/scala	/usr/local/jdk1.7.0_79
node1	192.168.146.145	/usr/local/hadoop-2.6.4	/usr/local/scala	/usr/local/jdk1.7.0_79
node2	192.168.146.144	/usr/local/hadoop-2.6.4	/usr/local/scala	/usr/local/jdk1.7.0_79
node3	192.168.146.143	/usr/local/hadoop-2.6.4	/usr/local/scala	/usr/local/jdk1.7.0_79

配置项	值	说明
将下载好的spark-2.0.0-bin-hadoop2.6.tgz拷贝到主机master上的/usr/local目录中，并执行命令解压到当前目录中：tar -zxf spark-2.0.0-bin-hadoop2.6.tgz，解压后的目录结构如下：		

```
总用量 88
drwxr-xr-x. 2 hadoop hadoop 4096 7月 20 05:28 bin
drwxr-xr-x. 2 hadoop hadoop 4096 7月 20 05:28 conf
drwxr-xr-x. 5 hadoop hadoop 47 7月 20 05:28 data
drwxr-xr-x. 4 hadoop hadoop 27 7月 20 05:28 examples
drwxr-xr-x. 2 hadoop hadoop 8192 7月 20 05:28 jars
-rw-r--r--. 1 hadoop hadoop 17811 7月 20 05:28 LICENSE
drwxr-xr-x. 2 hadoop hadoop 4096 7月 20 05:28 licenses
-rw-r--r--. 1 hadoop hadoop 24749 7月 20 05:28 NOTICE
drwxr-xr-x. 6 hadoop hadoop 4096 7月 20 05:28 python
drwxr-xr-x. 3 hadoop hadoop 16 7月 20 05:28 R
-rw-r--r--. 1 hadoop hadoop 3828 7月 20 05:28 README.md
-rw-r--r--. 1 hadoop hadoop 120 7月 20 05:28 RELEASE
drwxr-xr-x. 2 hadoop hadoop 4096 7月 20 05:28 sbin
drwxr-xr-x. 2 hadoop hadoop 41 7月 20 05:28 yarn
```

修改目录conf中的配置文件来配置spark的运行环境，conf目录中包含的文件有：

```
总用量 32
-rw-r--r--. 1 hadoop hadoop 987 7月 20 05:28 docker.properties.template
-rw-r--r--. 1 hadoop hadoop 1105 7月 20 05:28 fairscheduler.xml.template
-rw-r--r--. 1 hadoop hadoop 2025 7月 20 05:28 log4j.properties.template
-rw-r--r--. 1 hadoop hadoop 7239 7月 20 05:28 metrics.properties.template
-rw-r--r--. 1 hadoop hadoop 865 7月 20 05:28 slaves.template
-rw-r--r--. 1 hadoop hadoop 1292 7月 20 05:28 spark-defaults.conf.template
-rwxr-xr-x. 1 hadoop hadoop 3861 7月 20 05:28 spark-env.sh.template
```

我们主要修改spark-env.sh这个文件。拷贝spark-env.sh.template并重命名为spark-env.sh：
cp spark-env.sh.template spark-env.sh，结果如下：

```
总用量 36
-rw-r--r--. 1 hadoop hadoop 987 7月 20 05:28 docker.properties.template
-rw-r--r--. 1 hadoop hadoop 1105 7月 20 05:28 fairscheduler.xml.template
-rw-r--r--. 1 hadoop hadoop 2025 7月 20 05:28 log4j.properties.template
-rw-r--r--. 1 hadoop hadoop 7239 7月 20 05:28 metrics.properties.template
-rw-r--r--. 1 hadoop hadoop 865 7月 20 05:28 slaves.template
-rw-r--r--. 1 hadoop hadoop 1292 7月 20 05:28 spark-defaults.conf.template
-rwxr-xr-x. 1 hadoop hadoop 3861 10月 11 11:39 spark-env.sh
-rwxr-xr-x. 1 hadoop hadoop 3861 7月 20 05:28 spark-env.sh.template
```

接下来修改spark-env.sh，主要添加以下几项配置：

配置项	值	说明
JAVA_HOME	/usr/local/jdk1.7.0_79	指向jdk的安装路径
HADOOP_HOME	/usr/local/hadoop-2.6.4	指向hadoop的安装路径
SCALA_HOME	/usr/local/scala	scala的安装
HADOOP_CONF_DIR	/usr/local/hadoop-2.6.4/etc/hadoop	hadoop配置文件所在的目录
SPARK_MASTER_HOST	192.168.146.146	spark集群master运行主机

```
# - SPARK_NICENESS      The scheduling priority for daemons. (Default: 0)
JAVA_HOME=/usr/local/jdk1.7.0_79
SPARK_MASTER_HOST=192.168.146.146
HADOOP_HOME=/usr/local/hadoop-2.6.4
SCALA_HOME=/usr/local/scala
HADOOP_CONF_DIR=/usr/local/hadoop-2.6.4/etc/hadoop
~
```

执行sbin/start-all.sh命令，查看是否能够启动spark，查看是否有Worker和Master进程。

```
[hadoop@master spark2.0]$ jps
21655 Worker
3232 NameNode
3389 SecondaryNameNode
3569 ResourceManager
21704 Jps
21568 Master
[hadoop@master spark2.0]$
```

运行spark提供的示例检测：bin/run-example JavaSparkPi 5 2

```
[hadoop@master spark2.0]$ bin/run-example JavaSparkPi 5 2
16/10/11 12:29:36 INFO spark.SparkContext: Running Spark version 2.0.0
16/10/11 12:29:36 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
16/10/11 12:29:36 INFO spark.SecurityManager: Changing view acls to: hadoop
16/10/11 12:29:36 INFO spark.SecurityManager: Changing modify acls to: hadoop
16/10/11 12:29:36 INFO spark.SecurityManager: Changing view acls groups to:
16/10/11 12:29:36 INFO spark.SecurityManager: Changing modify acls groups to:
16/10/11 12:29:36 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(hadoop); groups with view permissions: Set(); users with modify permissions: Set(hadoop); groups with modify permissions: Set()
16/10/11 12:29:37 INFO util.Utils: Successfully started service 'sparkDriver' on port 50756.
16/10/11 12:29:37 INFO spark.SparkEnv: Registering MapOutputTracker
16/10/11 12:29:37 INFO spark.SparkEnv: Registering BlockManagerMaster
16/10/11 12:29:37 INFO storage.DiskBlockManager: Created local directory at /tmp/blockmgr-fa6a376a-ea7f-47c3-bda5-f28160292907
16/10/11 12:29:37 INFO memory.MemoryStore: MemoryStore started with capacity 413.9 MB
16/10/11 12:29:37 INFO spark.SparkEnv: Registering OutputCommitCoordinator
16/10/11 12:29:37 INFO util.log: Logging initialized @2505ms
16/10/11 12:29:38 INFO server.Server: jetty-9.2.z-SNAPSHOT
16/10/11 12:29:38 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@c3fc233{/jobs,null,AVAILABLE}
16/10/11 12:29:38 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@2de6a88d{/jobs/json,null,AVAILABLE}
16/10/11 12:29:38 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@243d886{/jobs/job,null,AVAILABLE}
16/10/11 12:29:38 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@343c4693{/jobs/job/json,null,AVAILABLE}
16/10/11 12:29:38 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@613d77af{/stages,null,AVAILABLE}
16/10/11 12:29:38 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@a7cb409{/stages/json,null,AVAILABLE}
16/10/11 12:29:38 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@10fc040{/stages/stage,null,AVAILABLE}
16/10/11 12:29:38 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@3116aa4b{/stages/stage/json,null,AVAILABLE}
```

```

16/10/11 12:29:41 INFO executor.Executor: Running task 1.0 in stage 0.0 (TID 1)
16/10/11 12:29:41 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 1923 ms on localhost (1/2)
16/10/11 12:29:42 INFO executor.Executor: Finished task 1.0 in stage 0.0 (TID 1). 945 bytes result sent to driver
16/10/11 12:29:42 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 653 ms on localhost (2/2)
16/10/11 12:29:42 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
16/10/11 12:29:42 INFO scheduler.DAGScheduler: ResultStage 0 (reduce at JavaSparkPi.java:52) finished in 2.341 s
16/10/11 12:29:42 INFO scheduler.DAGScheduler: Job 0 finished: reduce at JavaSparkPi.java:52, took 2.938450 s
Pi is roughly 3.14522
16/10/11 12:29:42 INFO server.ServerConnector: Stopped ServerConnector@7e9237f3{HTTP/1.1}{0.0.0.0:4040}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@14eee870{/stages/stage/kill,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@617c0410{/api,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@3f06f3fb{/,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@467dd9f{/static,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@84373ce{/executors/threadDump/json,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@2ad6fc23{/executors/threadDump,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@41685d91{/executors/json,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@7abbd432{/executors,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@28e3aa36{/environment/json,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@155b7aee{/environment,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@633a7ecf{/storage/rdd/json,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@3fc0329b{/storage/rdd,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@193e282f{/storage/json,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@57552dfc{/storage,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@6a124c63{/stages/pool/json,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@47063b1a{/stages/pool,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@3116aa4b{/stages/stage/json,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@10fc040{/stages/stage,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@a7cb409{/stages/json,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@613d77af{/stages,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@343c4693{/jobs/job/json,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@243d886{/jobs/job,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@2de6a88d{/jobs/json,null,UNAVAILABLE}
16/10/11 12:29:42 INFO handler.ContextHandler: Stopped o.s.j.s.ServletContextHandler@c3fc233{/jobs,null,UNAVAILABLE}
16/10/11 12:29:42 INFO ui.SparkUI: Stopped Spark web UI at http://192.168.146.146:4040
16/10/11 12:29:42 INFO spark.MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
16/10/11 12:29:42 INFO memory.MemoryStore: MemoryStore cleared
16/10/11 12:29:42 INFO storage.BlockManager: BlockManager stopped
16/10/11 12:29:42 INFO storage.BlockManagerMaster: BlockManagerMaster stopped
16/10/11 12:29:42 INFO scheduler.OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
16/10/11 12:29:42 INFO spark.SparkContext: Successfully stopped SparkContext
16/10/11 12:29:42 INFO util.ShutdownHookManager: Shutdown hook called
16/10/11 12:29:42 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-27729e64-3449-474d-a688-19f2689f0347
thaddeus@spark2-016:~$

```

至此，单个节点的Spark环境配置结束。

多个节点的集群环境配置

在上述单个节点启动成功的基础上，配置多个节点集群环境是比较简单的一件事情。

- 修改master上节点的slaves配置文件来配置Worker节点的位置，这里我将node1、node2、node3作为Worker节点的运行机器，在conf/slaves(复制slaves.template)中添加node1、node2和node3。

```

# A Spark Worker will be started on each of the machines listed below.
#
node1
node2
node3
~

```

- 将master上配置好的spark目录文件全部分别拷贝到node1、node2和node3所在机器上(可以通过ansible这个工具来操作)。

- 通过命令sbin/start-all.sh启动spark集群

```
[hadoop@master spark-2.0]$ sbin/start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /usr/local/spark-2.0/logs/spark-hadoop-org.apache.s
aster-1-master.out
node3: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark-2.0/logs/spark-hadoop-org.apa
orker.Worker-1-node3.out
node2: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark-2.0/logs/spark-hadoop-org.apa
orker.Worker-1-node2.out
node1: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark-2.0/logs/spark-hadoop-org.apa
orker.Worker-1-node1.out
[hadoop@master spark-2.0]$ jps
3232 NameNode
3389 SecondaryNameNode
3569 ResourceManager
21948 Master
22020 Jps
[hadoop@master spark-2.0]$
```

```
[hadoop@node1 root]$ jps
3057 NodeManager
11676 Jps
11618 Worker
2947 DataNode

[hadoop@node2 local]$ jps
11342 Worker
11393 Jps
3141 NodeManager
3030 DataNode

[hadoop@node3 root]$ jps
11107 Worker
3091 NodeManager
2982 DataNode
11155 Jps
```

- 也可以通过浏览器来查看集群状态，在浏览器中通过spark主节点的8080端口可以查看集群状态，在浏览器中输入：<http://master:8080>

Spark Master at spark://192.168.146.146:7077

URL: spark://192.168.146.146:7077
 REST URL: spark://192.168.146.146:6066 (cluster mode)
 Alive Workers: 3
 Cores in use: 3 Total, 0 Used
 Memory in use: 3.0 GB Total, 0.0 B Used
 Applications: 0 Running, 0 Completed
 Drivers: 0 Running, 0 Completed
 Status: ALIVE

Worker Id	Address	State	Cores	Memory
worker-20161011124201-192.168.146.143-36065	192.168.146.143:36065	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)
worker-20161011124202-192.168.146.144-60622	192.168.146.144:60622	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)
worker-20161011124205-192.168.146.145-41560	192.168.146.145:41560	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)

Running Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State
----------------	------	-------	-----------------	----------------	------	-------

Completed Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State
----------------	------	-------	-----------------	----------------	------	-------

运行测试

Standalone模式的测试

在shell环境下运行Spark提供的案例程序JavaSparkPi，通过如下命令：

```
bin/spark-submit --class org.apache.spark.examples.JavaSparkPi --deploy-mode cluster
examples/jars/spark-examples_2.11-2.0.0.jar 10 4
```

shell 界面 输出 如 下 信 息：

```
[hadoop@master spark-2.0]$ bin/spark-submit --class org.apache.spark.examples.JavaSparkPi --master spark://master:7077 --deploy-mo
de cluster examples/jars/spark-examples_2.11-2.0.0.jar 10 4
Running Spark using the REST application submission protocol.
16/10/11 12:58:34 INFO rest.RestSubmissionClient: Submitting a request to launch an application in spark://master:7077.
16/10/11 12:58:44 WARN rest.RestSubmissionClient: Unable to connect to server spark://master:7077.
Warning: Master endpoint spark://master:7077 was not a REST server. Falling back to legacy submission gateway instead.
16/10/11 12:58:45 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes
where applicable
[hadoop@master spark-2.0]$
```

从shell界面我们不能得到什么信息，我们可以通过浏览器来查看执行这个应用的具体信息，在浏览器中输入<http://master:8080>，我们将看到如下信息：

master:8080

Spark Master at spark://192.168.146.146:7077

URL: spark://192.168.146.146:7077
REST URL: spark://192.168.146.146:6066 (cluster mode)
Alive Workers: 3
Cores in use: 3 Total, 0 Used
Memory in use: 3.0 GB Total, 0.0 B Used
Applications: 0 Running, 1 Completed
Drivers: 0 Running, 1 Completed
Status: ALIVE

Workers

Worker ID	Address	State	Cores	Memory
worker-20161011125312-192.168.146.144-38386	192.168.146.144:38386	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)
worker-20161011125312-192.168.146.145-33906	192.168.146.145:33906	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)
worker-20161011125313-192.168.146.143-35015	192.168.146.143:35015	ALIVE	1 (0 Used)	1024.0 MB (0.0 B Used)

Running Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
----------------	------	-------	-----------------	----------------	------	-------	----------

Running Drivers

Submission ID	Submitted Time	Worker	State	Cores	Memory	Main Class
---------------	----------------	--------	-------	-------	--------	------------

Completed Applications

Application ID	Submitted Time	User	State	Duration
app-20161011125850-0000	2016/10/11 12:58:50	hadoop	FINISHED	6 s

Completed Drivers

Submission ID	Submitted Time	Worker	State	Cores	Memory	Main Class
driver-20161011125846-0000	Tue Oct 11 12:58:46 CST 2016	worker-20161011125312-192.168.146.144-38386	FINISHED	1	1024.0 MB	org.apache.spark.examples.JavaSparkPi

点击 Completed Applications 中的链接，我们可以查看运行这个应用所消耗的资源情况：

master:8080/app?appId=app-20161011125850-0000

Application: JavaSparkPi

ID: app-20161011125850-0000
Name: JavaSparkPi
User: hadoop
Cores: Unlimited (2 granted)
Executor Memory: 1024.0 MB
Submit Date: Tue Oct 11 12:58:50 CST 2016
State: FINISHED

Executor Summary

ExecutorID	Worker	Cores	Memory	State	Logs
------------	--------	-------	--------	-------	------

Removed Executors

ExecutorID	Worker	Cores	Memory	State	Logs
1	worker-20161011125312-192.168.146.145-33906	1	1024	KILLED	stdout stderr
0	worker-20161011125313-192.168.146.143-35015	1	1024	KILLED	stdout stderr

点击 Completed Drivers 下的超链来查看应用程序 driver 进程所在节点的信息，通过这个节点我们也可以查看整个应用程序的输入结果信息。

Spark Worker at 192.168.146.144:38386

ID: worker-20161011125312-192.168.146.144-38386
Master URL: spark://192.168.146.146:7077
Cores: 1 (0 Used)
Memory: 1024.0 MB (0.0 B Used)

Running Executors (0)

ExecutorID	Cores	State	Memory	Job Details	Logs
------------	-------	-------	--------	-------------	------

Finished Drivers (1)

DriverID	Main Class	State	Cores	Memory	Logs
driver-20161011125846-0000	org.apache.spark.examples.JavaSparkPi	FINISHED	1	1024.0 MB	stdout stderr

点击上图中的 stdout，我们可以查看整个应用程序的输出结果。如下：

192.168.146.144:8081/logPage/?driverId=driver-20161011125846-0000&logType=stdout

stdout log page for driver-20161011125846-0000

Back to Master

Showing 22 Bytes: 0 - 22 of 22

Pi is roughly 3.14438

在yarn上的运行测试这里不再记录，请自行测试