

git_comments:**git_commits:**

1. **summary:** fix some typo in docs/cn/design.md (#1503)
message: fix some typo in docs/cn/design.md (#1503)
label: documentation

github_issues:

1. **title:** Some typo in docs/cn/design.md
body: ****BUG REPORT**** As title, there're some typo in `docs/cn/design.md`. RocketMQ消息队列集群主要包括 ****NameServe****、Broker in [2 通信机制](https://github.com/apache/rocketmq/blob/master/docs/cn/design.md#2-%E9%80%9A%E4%BF%A1%E6%9C%BA%E5%88%B6). It should be updated to ****NameServer****. 相当于 ****requestId****, 在同一个连接上 in [2.2 协议设计与编解码] (https://github.com/apache/rocketmq/blob/master/docs/cn/design.md#22-%E5%8D%8F%E8%AE%AE%E8%AE%BE%E8%AE%A1%E4%B8%8E%E7%BC%96%E8%A7%A3%E7%A0%81). It should be updated to ****requestId****. RocketMQ**这么做是还是在于**其Producer端写入消息和**Consomer**端订阅消息采用分离存储的机制来实现的 in [3 消息过滤] (https://github.com/apache/rocketmq/blob/master/docs/cn/design.md#3-%E6%B6%88%E6%81%AF%E8%BF%87%E6%BB%A4). I suggest it should be updated to ****这么做是在于**** and ****Consumer****. That's all.
2. **title:** Some typo in docs/cn/design.md
body: ****BUG REPORT**** As title, there're some typo in `docs/cn/design.md`. RocketMQ消息队列集群主要包括 ****NameServe****、Broker in [2 通信机制](https://github.com/apache/rocketmq/blob/master/docs/cn/design.md#2-%E9%80%9A%E4%BF%A1%E6%9C%BA%E5%88%B6). It should be updated to ****NameServer****. 相当于 ****requestId****, 在同一个连接上 in [2.2 协议设计与编解码] (https://github.com/apache/rocketmq/blob/master/docs/cn/design.md#22-%E5%8D%8F%E8%AE%AE%E8%AE%BE%E8%AE%A1%E4%B8%8E%E7%BC%96%E8%A7%A3%E7%A0%81). It should be updated to ****requestId****. RocketMQ**这么做是还是在于**其Producer端写入消息和**Consomer**端订阅消息采用分离存储的机制来实现的 in [3 消息过滤] (https://github.com/apache/rocketmq/blob/master/docs/cn/design.md#3-%E6%B6%88%E6%81%AF%E8%BF%87%E6%BB%A4). I suggest it should be updated to ****这么做是在于**** and ****Consumer****. That's all.
label: documentation
3. **title:** Some typo in docs/cn/design.md
body: ****BUG REPORT**** As title, there're some typo in `docs/cn/design.md`. RocketMQ消息队列集群主要包括 ****NameServe****、Broker in [2 通信机制](https://github.com/apache/rocketmq/blob/master/docs/cn/design.md#2-%E9%80%9A%E4%BF%A1%E6%9C%BA%E5%88%B6). It should be updated to ****NameServer****. 相当于 ****requestId****, 在同一个连接上 in [2.2 协议设计与编解码] (https://github.com/apache/rocketmq/blob/master/docs/cn/design.md#22-%E5%8D%8F%E8%AE%AE%E8%AE%BE%E8%AE%A1%E4%B8%8E%E7%BC%96%E8%A7%A3%E7%A0%81). It should be updated to ****requestId****. RocketMQ**这么做是还是在于**其Producer端写入消息和**Consomer**端订阅消息采用分离存储的机制来实现的 in [3 消息过滤] (https://github.com/apache/rocketmq/blob/master/docs/cn/design.md#3-%E6%B6%88%E6%81%AF%E8%BF%87%E6%BB%A4). I suggest it should be updated to ****这么做是在于**** and ****Consumer****. That's all.
label: documentation

github_issues_comments:

1. @duhenglucky please review it.

github_pulls:

1. **title:** [ROCKETMQ-14]invoke callback shoule be invoked in an executor rather than in current thread.
body: For success callback, processResponseCommand will execute invoke callback in a separate executor rather than the time thread. But for timeout problem , the timeout callback will be invoked directly in timer thread/netty selector

- thread. If the user put some time-cost task when timeout, the other timeout callback will be blocked to schedule. JIRA issue: <https://issues.apache.org/jira/browse/ROCKETMQ-14>
2. **title:** [ROCKETMQ-14]invoke callback shoule be invoked in an executor rather than in current thread.
body: For success callback, processResponseCommand will execute invoke callback in a saparate executor rather than the time thread. But for timeout problem , the timeout callback will be invoked directly in timer thread/netty selector thread. If the user put some time-cost task when timeout, the other timeout callback will be blocked to schedule. JIRA issue: <https://issues.apache.org/jira/browse/ROCKETMQ-14>
 3. **title:** [ROCKETMQ-14]invoke callback shoule be invoked in an executor rather than in current thread.
body: For success callback, processResponseCommand will execute invoke callback in a saparate executor rather than the time thread. But for timeout problem , the timeout callback will be invoked directly in timer thread/netty selector thread. If the user put some time-cost task when timeout, the other timeout callback will be blocked to schedule. JIRA issue: <https://issues.apache.org/jira/browse/ROCKETMQ-14>
 4. **title:** [ROCKETMQ-14]invoke callback shoule be invoked in an executor rather than in current thread.
body: For success callback, processResponseCommand will execute invoke callback in a saparate executor rather than the time thread. But for timeout problem , the timeout callback will be invoked directly in timer thread/netty selector thread. If the user put some time-cost task when timeout, the other timeout callback will be blocked to schedule. JIRA issue: <https://issues.apache.org/jira/browse/ROCKETMQ-14>
 5. **title:** [ROCKETMQ-14]invoke callback shoule be invoked in an executor rather than in current thread.
body: For success callback, processResponseCommand will execute invoke callback in a saparate executor rather than the time thread. But for timeout problem , the timeout callback will be invoked directly in timer thread/netty selector thread. If the user put some time-cost task when timeout, the other timeout callback will be blocked to schedule. JIRA issue: <https://issues.apache.org/jira/browse/ROCKETMQ-14>
 6. **title:** [ROCKETMQ-14]invoke callback shoule be invoked in an executor rather than in current thread.
body: For success callback, processResponseCommand will execute invoke callback in a saparate executor rather than the time thread. But for timeout problem , the timeout callback will be invoked directly in timer thread/netty selector thread. If the user put some time-cost task when timeout, the other timeout callback will be blocked to schedule. JIRA issue: <https://issues.apache.org/jira/browse/ROCKETMQ-14>
 7. **title:** [ROCKETMQ-14]invoke callback shoule be invoked in an executor rather than in current thread.
body: For success callback, processResponseCommand will execute invoke callback in a saparate executor rather than the time thread. But for timeout problem , the timeout callback will be invoked directly in timer thread/netty selector thread. If the user put some time-cost task when timeout, the other timeout callback will be blocked to schedule. JIRA issue: <https://issues.apache.org/jira/browse/ROCKETMQ-14>

github_pulls_comments:

1. @Jaskey Thanks for your PR, Could you paste some unit test for these question :-). please see our contributing checklist, <https://github.com/apache/incubator-rocketmq/blob/master/CONTRIBUTING.md>
2. @vongosling please check the committed unit test .
3. @Jaskey Thanks your PR Could you abide our PR process(modify your PR subject and mention your jira address in the description), please follow this PR. #5
4. Source code seems ok, please @zhouxinyu @lollipopjin review it
5. Hi, This PR has been merged, but next time please use JIRA issue name(`ROCKETMQ-14`) as branch name instead of `master`. Please refer to [here](<http://rocketmq.apache.org/docs/pull-request/>) for more details. Thanks.

github_pulls_reviews:

jira_issues:

1. **summary:** Remoting invoke callback shoule be invoked in an executor rather than in current thread.
description: For success callback, processResponseCommand will execute invoke callback in a saparate executor rather than the time thread. But for timeout problem , the timeout callback will be invoked directly in timer thread/netty selector thread. If the user put some time-cost task when timeout, the other timeout callback will be blocked to schedule.
2. **summary:** Remoting invoke callback shoule be invoked in an executor rather than in current thread.
description: For success callback, processResponseCommand will execute invoke callback in a saparate executor rather than the time thread. But for timeout problem , the timeout callback will be invoked directly in timer thread/netty selector thread. If the user put some time-cost task when timeout, the other timeout callback will be blocked to schedule.
3. **summary:** Remoting invoke callback shoule be invoked in an executor rather than in current thread.
description: For success callback, processResponseCommand will execute invoke callback in a saparate executor rather than the time thread. But for timeout problem , the timeout callback will be invoked directly in timer thread/netty selector thread. If the user put some time-cost task when timeout, the other timeout callback will be blocked to schedule.

jira_issues_comments:

1. github pr: <https://github.com/apache/incubator-rocketmq/pull/2>
2. Github user vongosling commented on the issue: <https://github.com/apache/incubator-rocketmq/pull/2> Source code seems ok, please @zhouxinyu @lollipopjin review it
3. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-rocketmq/pull/2>

4. Github user zhouxinyu commented on the issue: <https://github.com/apache/incubator-rocketmq/pull/2> Hi, This PR has been merged, but next time please use JIRA issue name(`ROCKETMQ-14`) as branch name instead of `master`. Please refer to [here](<http://rocketmq.apache.org/docs/pull-request/>) for more details. Thanks.
5. Github user vongosling commented on the issue: <https://github.com/apache/incubator-rocketmq/pull/19> IMO, we could close this pr without any more works @lollipopjin thanks @Jaskey
6. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-rocketmq/pull/19>
7. Github user Jaskey commented on the issue: <https://github.com/apache/incubator-rocketmq/pull/19> @vongosling Is there any problem on this unit test code?