Item 324

**git_comments:**

**git_commits:**

1. **summary:** Add cmd flag of retain key ordering (#8886)
   **message:** Add cmd flag of retain key ordering (#8886) Signed-off-by: xiaolong.ran <rxl@apache.org> Fixes #6527 ### Motivation We already support `key_shared` mode on Pulsar Functions in #7647, However, this option is not exposed in the command line parameters, so users currently cannot use this function. ### Modifications In this pull request, we expose this cmd flag of `--retain-key-ordering` and add the docs for this. ### Verifying this change The original test can cover this option, so, in this change, no relevant code is added

**github_issues:**

1. **title:** Having a Key_Shared mode for functions
   **body:** **Is your feature request related to a problem? Please describe.** As functions can manipulate states, race conditions may occur if multiple instance of the same function are processed in parallel and update state for the same key. **Describe the solution you'd like** having a Key_Shared mode for a function's instances - similar to Key_Shared subscription mode for consumers - would provide the guarantee that only one function instance would run at any time for a given key **Describe alternatives you've considered** Another solution could be to have transactions enabled for states, but i'm not sure it's even possible with BookKeeper. **Additional context**
2. **title:** Having a Key_Shared mode for functions
   **body:** **Is your feature request related to a problem? Please describe.** As functions can manipulate states, race conditions may occur if multiple instance of the same function are processed in parallel and update state for the same key. **Describe the solution you'd like** having a Key_Shared mode for a function's instances - similar to Key_Shared subscription mode for consumers - would provide the guarantee that only one function instance would run at any time for a given key **Describe alternatives you've considered** Another solution could be to have transactions enabled for states, but i'm not sure it's even possible with BookKeeper. **Additional context**
3. **title:** Having a Key_Shared mode for functions
   **body:** **Is your feature request related to a problem? Please describe.** As functions can manipulate states, race conditions may occur if multiple instance of the same function are processed in parallel and update state for the same key. **Describe the solution you'd like** having a Key_Shared mode for a function's instances - similar to Key_Shared subscription mode for consumers - would provide the guarantee that only one function instance would run at any time for a given key **Describe alternatives you've considered** Another solution could be to have transactions enabled for states, but i'm not sure it's even possible with BookKeeper. **Additional context**
4. **title:** Having a Key_Shared mode for functions
   **body:** **Is your feature request related to a problem? Please describe.** As functions can manipulate states, race conditions may occur if multiple instance of the same function are processed in parallel and update state for the same key. **Describe the solution you'd like** having a Key_Shared mode for a function's instances - similar to Key_Shared subscription mode for consumers - would provide the guarantee that only one function instance would run at any time for a given key **Describe alternatives you've considered** Another solution could be to have transactions enabled for states, but i'm not sure it's even possible with BookKeeper. **Additional context**
5. **title:** Having a Key_Shared mode for functions
   **body:** **Is your feature request related to a problem? Please describe.** As functions can manipulate states, race conditions may occur if multiple instance of the same function are processed in parallel and update state for the same key. **Describe the solution you'd like** having a Key_Shared mode for a function's instances - similar to Key_Shared subscription mode for consumers - would provide the guarantee that only one function instance would run at any time for a given key **Describe alternatives you've considered** Another solution could be to have transactions enabled for states, but i'm not sure it's even possible with BookKeeper. **Additional context**
   **label:** documentation
6. **title:** Having a Key_Shared mode for functions
   **body:** **Is your feature request related to a problem? Please describe.** As functions can manipulate states, race conditions may occur if multiple instance of the same function are processed in parallel and

update state for the same key. **Describe the solution you'd like** having a Key_Shared mode for a function's instances - similar to Key_Shared subscription mode for consumers - would provide the guarantee that only one function instance would run at any time for a given key **Describe alternatives you've considered** Another solution could be to have transactions enabled for states, but i'm not sure it's even possible with BookKeeper. **Additional context**

7. **title:** Having a Key_Shared mode for functions
   **body:** **Is your feature request related to a problem? Please describe.** As functions can manipulate states, race conditions may occur if multiple instance of the same function are processed in parallel and update state for the same key. **Describe the solution you'd like** having a Key_Shared mode for a function's instances - similar to Key_Shared subscription mode for consumers - would provide the guarantee that only one function instance would run at any time for a given key **Describe alternatives you've considered** Another solution could be to have transactions enabled for states, but i'm not sure it's even possible with BookKeeper. **Additional context**
8. **title:** Having a Key_Shared mode for functions
   **body:** **Is your feature request related to a problem? Please describe.** As functions can manipulate states, race conditions may occur if multiple instance of the same function are processed in parallel and update state for the same key. **Describe the solution you'd like** having a Key_Shared mode for a function's instances - similar to Key_Shared subscription mode for consumers - would provide the guarantee that only one function instance would run at any time for a given key **Describe alternatives you've considered** Another solution could be to have transactions enabled for states, but i'm not sure it's even possible with BookKeeper. **Additional context**
9. **title:** Having a Key_Shared mode for functions
   **body:** **Is your feature request related to a problem? Please describe.** As functions can manipulate states, race conditions may occur if multiple instance of the same function are processed in parallel and update state for the same key. **Describe the solution you'd like** having a Key_Shared mode for a function's instances - similar to Key_Shared subscription mode for consumers - would provide the guarantee that only one function instance would run at any time for a given key **Describe alternatives you've considered** Another solution could be to have transactions enabled for states, but i'm not sure it's even possible with BookKeeper. **Additional context**

**github_issues_comments:**

1. @sijie Hi, is adding key shared mode on the roadmap soon?
2. @Lanayx I think @wolfstudy is picking that up already.
3. Yes, we already support key_shared mode on pulsar functions in #7647.
4. That's great news!
5. **body:** Does it mean this issue should be closed with 2.7? Is there somewhere in the documentation explaining how to use it this feature?
   **label:** documentation
6. > Does it mean this issue should be closed with 2.7? Yes, in 2.7.0, we can use this feature. > Is there somewhere in the documentation explaining how to use it this feature? @sijie It seems that we need to expose the corresponding parameters to users. Currently, when creating pulsar function, `--retain-key-ordering` is not exposed.
7. > @sijie It seems that we need to expose the corresponding parameters to users. Currently, when creating pulsar function, `--retain-key-ordering` is not exposed. Is there a turnaround to be able to use it? Is that enough to set `functionConfig.retainKeyOrdering = true` if using PulsarAdmin java?

**github_pulls:**

1. **title:** Add cmd flag of retain key ordering
   **body:** Signed-off-by: xiaolong.ran <rxl@apache.org> Fixes #6527 ### Motivation We already support `key_shared` mode on Pulsar Functions in #7647, However, this option is not exposed in the command line parameters, so users currently cannot use this function. ### Modifications In this pull request, we expose this cmd flag of `--retain-key-ordering` and add the docs for this. ### Verifying this change The original test can cover this option, so, in this change, no relevant code is added ### Documentation - Does this pull request introduce a new feature? (yes) - If yes, how is the feature documented? (docs) - If a feature is not applicable for documentation, explain why? - If a feature is not documented yet in this PR, please create a followup issue for adding the documentation
   **label:** documentation

## github_pulls_comments:

## github_pulls_reviews:

## jira_issues:

## jira_issues_comments: