Theme: Shanghai library FOLIO project
Time: August 11, 2020 07:00pm (EST) / August 12, 2020 07:00am (GMT+8)

Link to Zoom meeting
https://zoom.us/j/98935865023?pwd=S0tQL25VWGhUMk5NbWRnaXJDTkdGUT09

Attendees:
Vincent Bareau (Enterprise Architect, EBSCO)
Gang Zhou (Project manager, Shanghai library)
Sha Jiang (Technical Director, Jiatu)
Lucy Liu (Product Owner, Folio China)

Notes:

1. **ES post-/pre- filter support provided by Okapi vs. direct message from storage (UXPROD-2592)**
   Sha Jiang: **Mikhail** had concerns over using pub-sub because there is a possibility that we will lose data. He recommended sending/receiving Kafka messages instead of using any inter-modules to do this.
   Vince:
   - It makes sense. The mod-pubsub can't guarantee the delivery of messages.
   - You said before that you don't want to require every module to implement something. If you want to add a module, you don't want to make modifications to other modules.
   - We have a mechanism which is part of folio. That is Okapi where you can create a filter. You can have Okapi pass the message to the filter either before or after it responds to the request. For example, if you call the module, you have the module receive the data and the module does something. If you have code that the module is required to directly make a call from the message queue or whatever to be done, if Okapi is able to connect with the module in first place, Okapi is able to send a message to the filter. You set the filter once, and you don't have to worry about the modules to implement or adding a library in order to be able to communicate with your client.
   - There are costs with using the filter. One cost is that it's going to slow down the performance a little bit because Okapi needs to make additional transactions.
   - Questions:
     - Did you consider the option of using a post and pre filter provided by Okapi to do the communications?
     - How would you expect the packaging implementation to work? How do you do it with the module? Do you provide a library that is included in the module?

Sha Jiang:
- We modified the RMB runtime module to make routes to listen to the APIs so that the APIs can send messages to Kafka.
- We can share the code with you on Gitlab to help you understand the details.

Vince:
- **Mikhail** has access to Gitlab and he can see the code.
- You put a listener in RMB. The Listener needs to have knowledge as to what to listen for. How is the decision made? Does the filter decision exist in the RMB code, or the module itself uses a connector that RMB provides to Kafka and the business logic is in the inventory module?

Sha Jiang: We extend the RMB module instead of changing RMB code directly. We write a subclass for the RMB module. The decision you mentioned is made in the configuration file.
Vince: When do you read the configuration file?
Sha Jiang: Once the module starts.
Vince: Is this for all tenants?
Sha Jiang: Yes.
Vince: There are a lot of different moving projects right now in folio, including yours with ElasticSearch and Kafka. I want to make sure they're all consistent with each other. So we don't have to repeat code, duplicate and maintain multiple versions of things.

2. **The slow SRS and data import - mod-pubsub threw exceptions after importing 17,000 Marc records.**
Lucy: I translated and shared the test results with you in the agenda email. Did you have a chance to read it?
Vince:
- Yes, I've seen it. This problem has been seen by others in other tests. The explanation is the way that mod-pubsub is working in this particular solution. It takes a certain amount of time for SRS to do its work. So you want to load in 17,000 records. SRS will parse the file and insert the records. Notifications are made to inventory so that inventory can update itself. Messages will go out from SRM into pubsub and the subscribers will be mod-inventory. The mod-inventory receives the messages. The new Marc records are already in the payload. So it's already being delivered into the inventory. And the inventory will proceed to process and do whatever needs to be done. When you start doing that, it starts to work. But mod-pubsub doesn't know whether or not there's capacity left in the inventory. So it will push messages one after another. It will push 17000 records to inventory, which will try its best to start processing but will run out of resources very quickly and produce the 500 errors that you've seen. So this is a known issue right now with the way that the workflow is set up.

- We have been working very recently (in the last two weeks) with the Folijet team (with Taras and some others) to prototype a different solution. We decided that mod-pubsub is not adding enough value for being a convenient wrapper on Kafka. So the team prototyped a solution that basically does direct Kafka integration instead of integrating the mod-pubsub. Now what happens is the messages go out onto Kafka when SRS is done with all this loading of the data from the files. SRM sends out a message, notifying a bunch of Records available. The inventory will go to retrieve the message. It will only retrieve the next message when it is done with this current processing. So we can limit the processing of inventory so that it doesn't get overwhelmed with too many tasks being pushed out of it. The inventory has to pull the task from Kafka directly. And there are some other details of implementing in terms of modifications of adding some ability to check whether there is capacity and so forth. The approach we were taking is that we slowed down in order to gain stability in the design. Once we have that, we can deal with the performance problem because now we can scale horizontally. We can't scale with the existing mod-pubsub solution because if we have multiple instances of inventory, mod-pubsub doesn't know which pubsub is overwhelmed and it cannot target the message to any particular system.
- The team ran their last test this morning and succeeded in doing 30000 records in 20 minutes without any problems or errors. This is also good because it has the ability to not lose the messages because if, for example, inventory crashes, nothing is lost so long as you have three or four inventories running. If one inventory crashes, the others can pick up. The one recovers and gets back on line and can restore, because messages are not pushed to the inventory that crashed. It requires it to pick up messages to do its work.
- They are going to make changes to inventory, SRS, SRM, and they are going to package up the Kafka connector in the form of the library.

Gang Zhou: When will this be released?
Vince: It will be released Q3. We will probably see it in the next Sprint review.
Gang Zhou: Are there documents we can refer to?
Vince: The issue is on Jira. But documentation of this solution is not ready yet. It's a rapid development effort. We will probably see the documentation next week.
Vince:

- This is also related to the modification you made to RMB because this is another version of the Kafka connector. I would like to see we have one solution. Now we have two different solutions with a connector to Kafka:
    - Assuming we pass this to the community, the proposed solution is to add the ability of modules to include Kafka connector which is what

    happens in this solution. In the near future we will have inventory that already has a connector built into it.
    - You have a solution with a modified RMB that has its own Kafka connector that does filter in the business logic and pushes things into ElasticSearch for indexing.
- In addition to that, there is another effort being done where we have elasticsearch being used for the circulation log project. We have one of the Epics or features to construct the circulation log. There's active work with that. Vasily is working on its sign with that team. He wants to see Kafka in ES. So he wants to push events into the message queue using Kafka, then have the actual circulation log maintained as a topic in elasticsearch index. So we have the third case where we have events going on in Kafka with elasticsearch integration.
- So I think we can have the two that I described for data import and the circulation log use the same amount of work. Vasily's solution for circulation log could make use of the Kafka connector library created. And it will use the ES as well. How would you solve your problem for your ElasticSearch use with the Kafka connector that lives in the library? So that might require putting the business logic that decides the point of filter in mod-inventory, for example. Inventory has a connector so it already knows how to talk to Kafka, send the messages and so forth. Potentially you could use the same library on your side to pull the same messages. What do you think of that?

Sha Jiang: We can use the same library. But we don't have the documents. I think we should complete our project with our solution. If the library is released, we can change to use that library.
Vince: That's fine. You have your own deadlines. I am making you aware that there is a parallel effort going on now in regards to the Kafka integration. It's not just a matter of simply changing your connector with a different connector. There is a slightly different implementation in terms of the responsibility so the filtering itself, the decision is not being driven by a configuration file that is read by RMB but rather it's something that is being done by mod-inventory. So it's a different approach. If you adopt it, later on you will have the client in mod-inventory and business logic to decide what needs to be filtered. I don't think it's going to change very often. It's a nice configuration file. But I think that once you have established which transactions are critical for re-indexing, it probably is not going to be very difficult to do that.
Sha Jiang: We can have multiple choices for the integration of Kafka. We can leave the library or the modification of RMB both available for the developers. They can have their own choices on how to integrate Kafka into their modules.
Vince: Correct. But there is potential rework of RMB. There's discussion of replacing parts of RMB with Spring, updating RMB to Java 11, which is the next LTS. We have

to do that by October. So RMB is not maintained. Keep in mind that everything in this project is changing and nothing is permanent.

3. **Any plan for the serials module?**
   Gang Zhou: We are planning how many modules will go online after we deliver the circulation module. We haven't seen the whole functional module in the demo or documents. Could you give us some information?
   Vince: Not familiar with this. I think we have supported serials in inventory already.
   Gang Zhou: We didn't see that. Serials are different from monographs. They have patterns and additional information on the item, such as volume, issue, etc. In general, we use a different module.
   Vince: EBSCO has a module for electronic serials, eholdings, which is backed up by the knowledge base. There is another ERM project which supports its own local databases, which also have support for electronic serials. Are you looking for sth for print serials?
   Gang Zhou: Yes, for print serials.
   Vince: don't know the plan for that. Charlotte might know this. I can inquire for you.
   Lucy:
   - Here are some meeting notes from the Resource Management Group. https://wiki.folio.org/display/RM/2020-05-08+Resource+Management+Meeting+Notes and https://wiki.folio.org/display/RM/2020-05-15+Meeting+notes There are relevant discussions.
   - There is also a ticket on Jira (UXPROD-194). It will be discussed in the implementers group meeting next Tuesday.
   - Maybe Gang Zhou can create a list describing the needs and share with Vince and Lucy so that we both can reach out to the community to find answers. Lucy will also bring it to the implementers group meeting for discussion next week.

   Gang Zhou: Sure.
   Note: Lucy received the list through WeChat. The list includes:
   - Serials acquisition
     - vendor catalog management
     - managing volumes and issues
     - Data receiving
     - Cataloging
     - Renewal (batch or individual)
   - Serials check-in and bounding
     - Receiving upon arrival
     - Multi-tenant receiving
   - Archival of bound volumes
     - Archival of back volumes and issues

        - Management of bound volumes

Vince: Do you have specialized needs for serials support in China, something like Marc, BIBFRAME, etc.?
Gang Zhou: That module should support different models. We use Marc in general. Prediction of the serials patterns, receiving and check-in, supporting marc holdings for LC, print holdings, etc.