Streaming Processing for RAG Architectures

Namita Sharma
Pablo Rodriguez Defino

BEAM SUMMIT

September 4-5, 2024

Sunnyvale, CA. USA

Agenda

- What is RAG
- Why Streaming for RAG
- Using Beam for RAG
- Example Tech Stack
- Beam Transforms
- Use Case example / Demo
- Closeup

BEAM SUMMIT

RAG (Retrieval Augmented Generation)

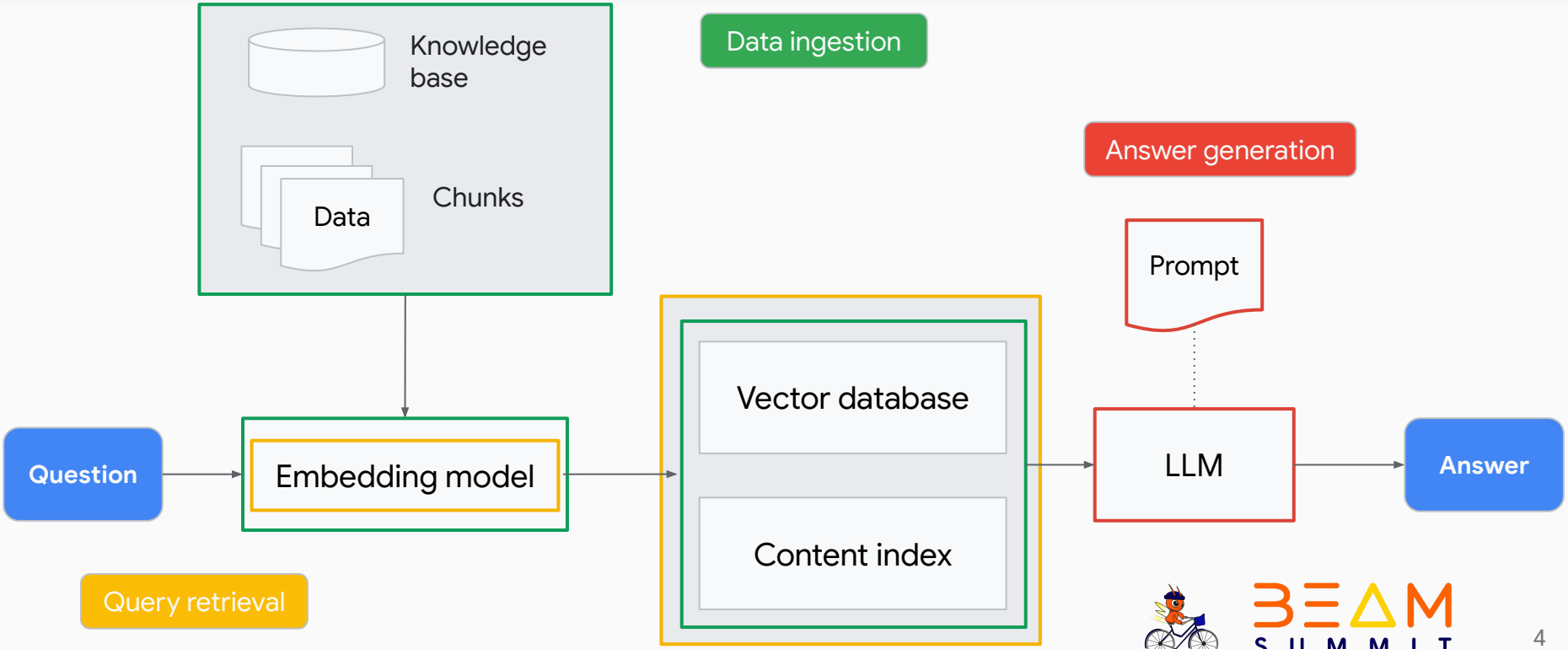
RAG is a NLP technique leverages both neural generative models & neural retrieval models to enhance the capabilities of conversation AI capabilities

The workflow of a RAG application:

- **Query:** The user inputs a question or a prompt.
- **Retrieval:** Similar to how Search works
- **Retrieval post processing [Optional]**: A specialised reader model extracts the most important snippets from the retrieved text.
- **Generation with context:** The generative language model is conditioned on both the original query and the extracted text snippets from the reader. This ensures the generated response is both fluent and factually grounded in the retrieved knowledge.

BEAM SUMMIT

RAG Pipeline



Why Streaming for RAG?

- Access to real-time data for more accurate and relevant responses
- Valuable context data being retrieved from Streaming sources (Kafka, PubSub, etc.)
- Scalability to handle large volumes of data can be problematic
  - Efficiency for real-time or near-real-time applications

BEAM SUMMIT

Why using Apache Beam for RAG?

- Apache Beam framework exposes a set of powerful constructs
  - Extensive IO support
  - Clear primitives for data processing requirements
    - windowing capabilities, time or session based
    - tunable completeness, latency and cost
- Multi-Language support
  - Python is the de facto language for data wrangling / prep on ML
- Runtime flexibility
  - Several available runners to choose from
- Unified programming model
  - Processing logic can be abstracted from the run modes: streaming or batch
  - Streaming for timely, low latency processing
  - Batch for offline (re)processing *(common on a fast paced tech area)

BEAM SUMMIT

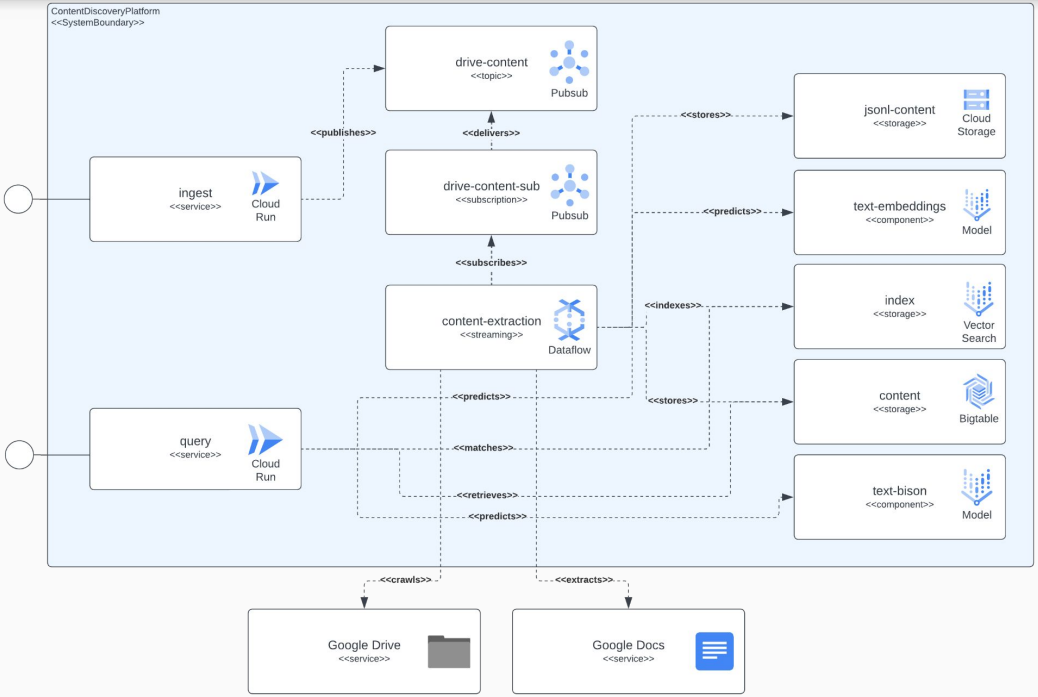
Use Case

- Conversational insights
  - Topics related with known, historical data
    - Models can be trained offline with large corpus of data, tuning for the expected cases
  - Some topics may be about recently generated data
    - External data events can enrich the model context:
      - social networks, news updates, document updates
      - world events
      - sports or election results, etc
- Data freshness, consistency and validation
  - Low latency requirements
  - Cross references are needed
  - Store information for future tuning
- Changes on volume, expected throughput
  - Rapid growth for hot topics can be problematic
  - Need to correctly handle back pressure, idempotency

BEAM SUMMIT

Example Pipeline

- Streaming source
  - decoupled, low latency input
- Content processing
  - extraction
  - embedding
  - indexing
- Storage destination
  - extracted content
  - embedding vectors
  - content/embedding mapping



Example Tech Stack

- Components
  - Streaming infrastructure
    - Pub/Sub
  - Apache Beam
    - Dataflow Runner
  - Low latency storage
    - Bigtable
  - ML / LLM interactions
    - Vertex AI
  - Large throughput storage
    - GCS
  - Service layer
    - Cloud Run

- Code is available for testing.
  - Includes pipeline, storage and demo query service layer
  - Infrastructure automation

BEAM SUMMIT

Beam Transforms

- PubSubIO
  - Reads incoming events
- Custom DoFns
  - Data retrieval from events
  - Format translations, error handling
- PythonExternalTransform
  - Custom code for chunking implementation
  - LLM interactions for embeddings
- BigTableIO
  - Stores mappings between contents and embeddings
- TextIO
  - Stores extracted content for later model's fine tuning

```
// Read the events with Google Drive identifiers and extract the documents contents
var maybeDocsContents =
    pipeline
        .apply(
            "ReadSharedURLs",
            PubsubIO.readMessages().fromSubscription(options.getSubscription()))
        .apply(
            "ApplyWindow",
            Window.<PubsubMessage>into(FixedWindows.of(Duration.standardMinutes(1)))
                .triggering(Repeatedly.forever(AfterWatermark.pastEndOfWindow()))
                .discardingFiredPanes()
                .withAllowedLateness(Duration.standardMinutes(1)))
        .apply("ExtractDocumentsContent", DocumentProcessorTransform.create());

// then we transform the document's content into JSONL format and store it on GCS
maybeDocsContents
    .output()
    .apply(
        "ToJSONLFormat",
        FlatMapElements.into(
                TypeDescriptors.kvs(TypeDescriptors.strings(), TypeDescriptors.strings()))
            .via(ExtractionUtils::docContentToKeyedJSONLFormat))
    .apply(
        "WriteJSONLToGCS",
        FileIO.<String, KV<String, String>>writeDynamic()
            .by(nameAndLineContent -> nameAndLineContent.getKey())
            .withDestinationCoder(StringUtf8Coder.of())
            .via(
                Contextful.fn(
                    nameAndLineContent ->
                        nameAndLineContent.getValue().replaceAll("[\\n\\r]", "")),
                TextIO.sink())
            .to(options.getBucketLocation())
            .withNaming(
                Contextful.fn(
                    name ->
                        ExtractionUtils.documentAndIdNaming(
                            "jsonl_content/" + name, ".jsonl"))));

// also we grab the content an create document chunks that will be used to extract embeddings
// and then they will be stored in MatchingEngine
maybeDocsContents
    .output()
    .apply("ProcessEmbeddings", FoundationModelsTransform.processEmbeddings())
    .apply("StoreEmbeddings", StoreEmbeddingsResults.create());

// also little bit of error handling.
// if the error is retriable (like lack of permissions on the docs) the events will be sent to
// the original PubSub Topic
maybeDocsContents
    .failures()
    .apply("ProcessErrorAndMaybeRetry", ErrorHandlingTransform.create());
```

BEAM SUMMIT

Let’s do some demo

BEAM SUMMIT

Thank you!

Questions?

Linkedin

- Namita: @namita1
- Pablo: @prodriguezdefino

Deeper insights at: Apache Beam Blog Post

BEAM SUMMIT

Title and body

You do not really need to use this template for all your slides.

As long as you use the title/cover slide, that is enough. You can use your own template for the rest of your presentation.

But if you want to use this, that is great!

BEAM SUMMIT

Only use images for which you have permission/copyright

BEAM SUMMIT

This is a section header

BEAM SUMMIT

Title on color

Maybe use this layout to show a square or vertical image on the right?

BEAM SUMMIT

Thank you!

Questions?

In the last slide (Q&A), please include some contact information.

Maybe your linkedin, X/twitter, email, blog, or wherever you preferred to be followed/reached for this topic.

BEAM SUMMIT