

Apache Beam is an open source, unified programming model for defining both batch and streaming data-parallel processing pipelines

Basics:

**Youtube:**

<https://www.youtube.com/watch?v=-6xpDRiurmA>

<https://cloud.google.com/dataflow/docs/concepts/beam-programming-model>

**Source**: <https://www.xenonstack.com/blog/apache-beam/>

**Good link basics**: <https://www.xenonstack.com/blog/apache-beam/>

<https://www.linkedin.com/pulse/from-lambda-architecture-kappa-using-apache-beam-siddharth-mittal>

<https://sanjayasubedi.com.np/apachebeam/beam-ptransforms/>

<https://cloud.google.com/dataflow/docs/concepts/beam-programming-model>

<https://medium.com/dailymotion/realtime-data-processing-with-apache-beam-and-google-dataflow-at-dailymotion-7d1b994dc816>

**JSON Processing :**

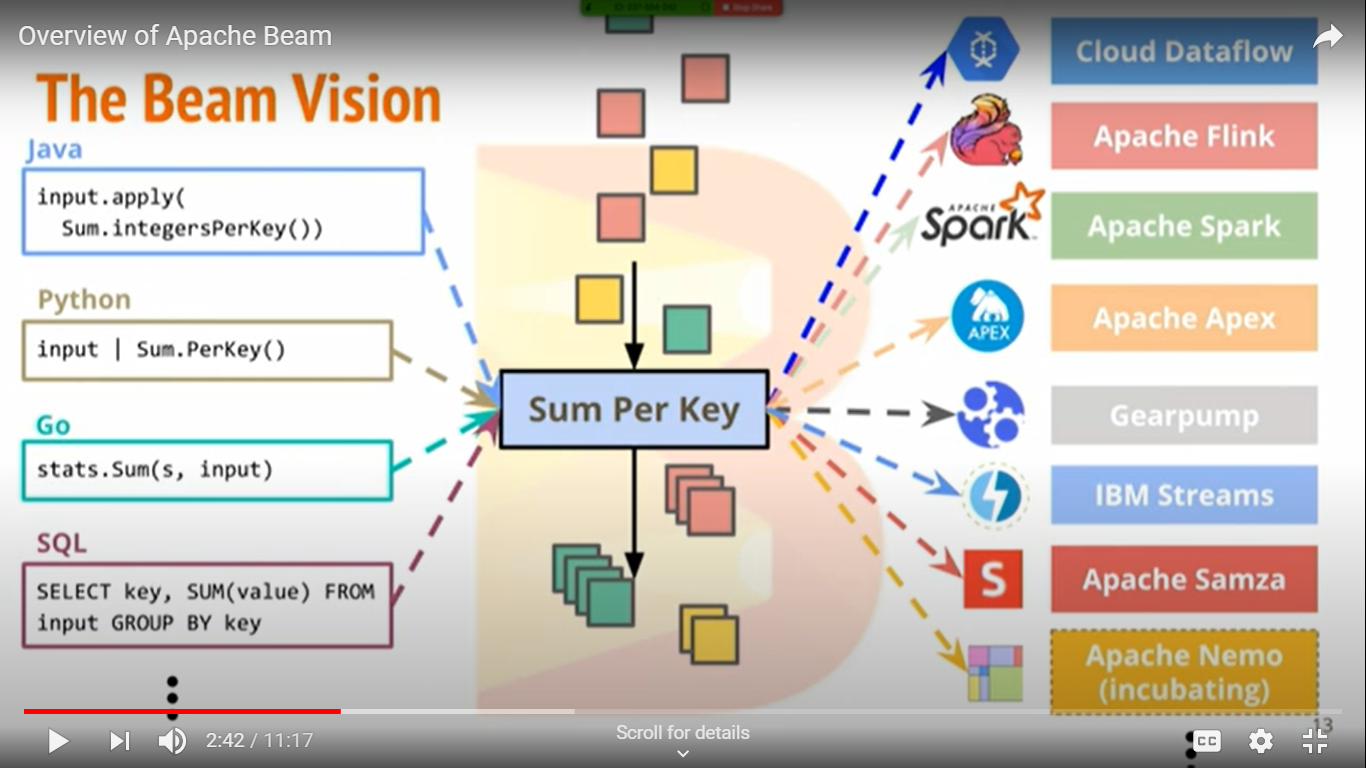
<https://github.com/apache/beam/blob/06128f27d1780f25c23ca65cc7ace693a78dac80/sdks/java/core/src/main/java/org/apache/beam/sdk/transforms/JsonToRow.java#L98>

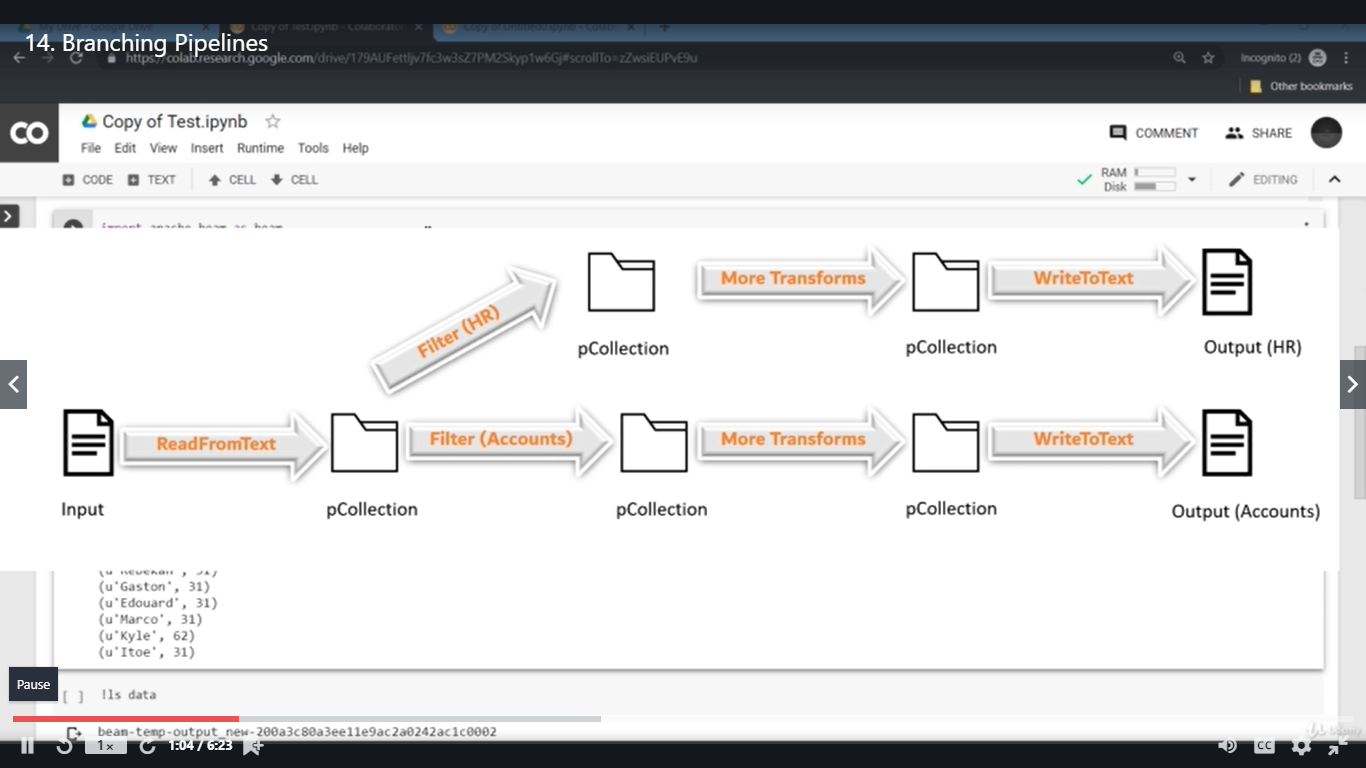
**Java sample:**

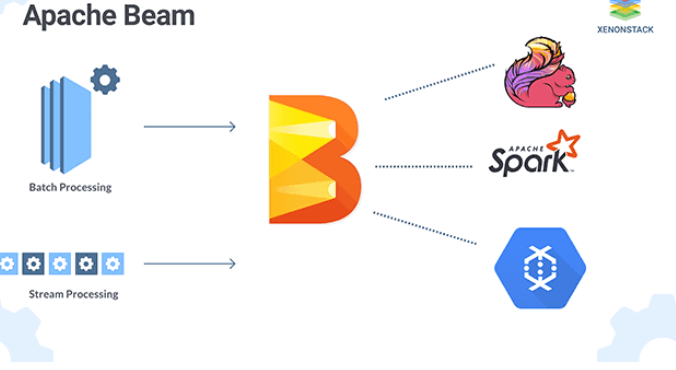
<https://dzone.com/articles/how-to-develop-a-data-processing-job-using-apache>

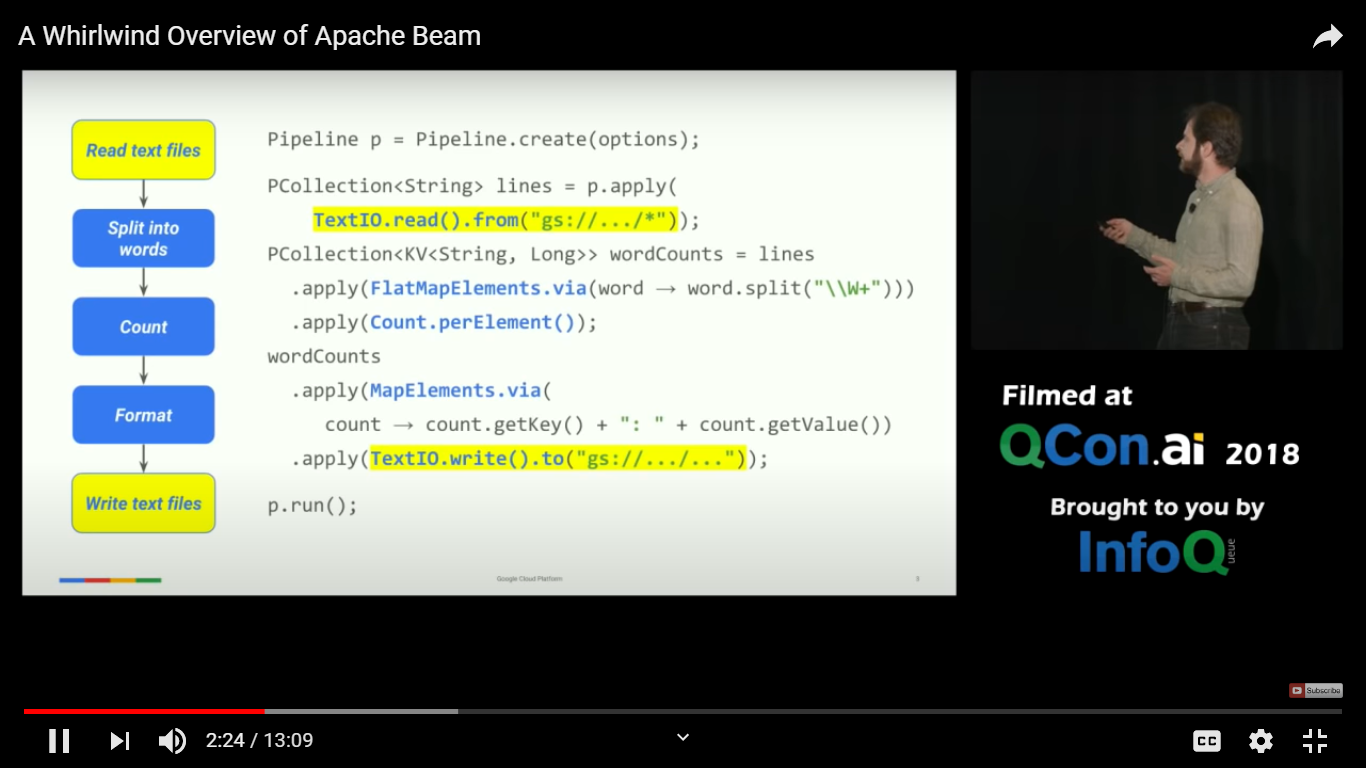
GIT:

<https://gitlab.com/saikrishnasahu/beneficiary-owner-data-processor>  
<https://gitlab.com/saikrishnasahu/dummy-rest>









**Pros of Apache Beam:**

* **Abstraction over different execution backends**. Data Engineers can now develop transformation API which can be executed on a different execution engine without changing even one line of code.
* **Clean and Simple programming model**. Easy to understand, implement and maintain.
* **Same data pipeline** for batch processing as well as for stream processing.

Along with above-mentioned advantages, we also observed some disadvantages or we can say we observed some limitations with Apache Beam SDK.

**Cons of Apache Beam:**

* **Lesser transparency and control** over what exactly is getting executed over the execution engine.
* **The limited scope of performance improvement** tricks as compared to Apache Spark API.
* **Apache Beam is a new API**, its latest SDK 2.0.0 which is its first stable version came in first half of 2017 and you will find very limited community comments and contribution on various sites and blogs.
* **The API has many open bugs.** Like with spark runner they are not supporting non-blocking pipeline execution. You will find an unresolved Jira for the same.
* **The API has a significant number of issues** when you execute your pipeline on windows environment. For example, you will face an error while you try to read a file from windows folder using TextIO. As work around you have to keep that file in your class path.

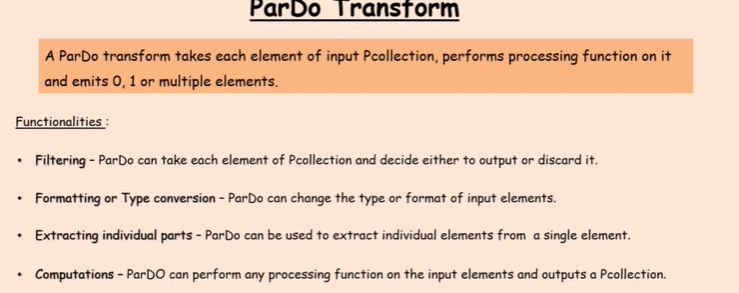
**Conclusion:**

***Apache Beam is a very promising SDK*** for developing distributed processing applications. As of now, ***it's stable and successful with Google Cloud Dataflow*** but it's little immature while working on other execution engines. **Many open Issues\Jira** needs to be addressed. In future releases, we are looking forward to more transparent yet abstract API.

***“Write Once and Execute Anywhere.”***

**“Undoubtedly Apache Beam is future of Streaming as well as Batch data processing applications.”**

Pardo:



ParDo is the core parallel processing operation in the Apache Beam SDKs, invoking a user-specified function on each of the elements of the input PCollection. ParDo collects the zero or more output elements into an output PCollection. The ParDo transform processes elements independently and possibly in parallel.