# **RESEARCH PLAN**

## **DP1: APACHE FLINK**

|  |
| --- |
| **Group name:** Messiu  **List of members:**   * 20127406 – Thái Cẩm Phong * 20127333 – Nguyễn Duy Thịnh * 20127062 – Nguyễn Khải Phú * 20127411 – Đỗ Đạt Thành |

|  |
| --- |
| **Keywords:** stream-processing, fault-tolerance, scalability, low latency, APIs  **Description:** Apache Flink is an open-source, data processing framework that supports batch-processing and stream-processing. It is a distributed processing engine designed for all common cluster environments, with a powerful runtime system focusing on low event latency and very high throughput.  **List of references:**   * <https://flink.apache.org> – *Flink* * <https://www.youtube.com/@ververica1483> – *Ververica* * <https://nexocode.com/blog/posts/what-is-apache-flink/> - *nexocode* * https://www.oreilly.com/library/view/introduction-to-apache/9781491977132/ch01.html - *O’Reilly* * <https://github.com/apache/flink> - *Apache Flink Github* * <https://en.wikipedia.org/wiki/Apache_Flink> - *Wikipedia* * <http://stratosphere.eu> – *Stratosphere* * <https://data-flair.training/blogs/apache-flink-ecosystem-components/> - *DataFlair* * <https://forum.huawei.com/enterprise/en/fi-components-basic-principle-of-flink/thread/599282-893> - HUAWEI Forum * <https://cwiki.apache.org/confluence/display/FLINK/Powered+by+Flink> – *Atlassian Confluence* * <https://spark.apache.org> – *Apache Spark* * https://cloudinfrastructureservices.co.uk/apache-spark-vs-flink-whats-the-difference/ - *Cloud Infrasturcture Services* * <https://www.tutorialspoint.com/apache_flink/apache_flink_conclusion> - *tutorials-point* * <https://www.ververica.com/blog/blink-flink-alibaba-search> – *Ververica* * <https://www.ververica.com/blog/flink-at-bouygues> - *Ververica* |

|  |
| --- |
| **Presentation outline**   1. **Introduction:**   Many streaming applications and systems in the world are designed to operate continuously with little downtime. From large-scale systems like transaction processing, weather forecasting, to small, pocket-size devices like smartphones, sensors, all of them create and receive seamless data streams every day. Companies from different industries all want to collect these data streams for analytical purposes.  Apache Flink is the right solution for this need. It is an open-source software, designed to perform computations and analysis over unbounded and bounded data streams. Historically, processing streams of data on a very large scale, across multiple sectors, has always been a challenge. But now, with the introduction of new technologies and architecture such as Flink, one can wish to analyze an extensive amount of real-time data in a timely fashion.  Apache Flink originated from Stratosphere, a collaboration of Technical University of Berlin, Humboldt University and Hasso Plattner Institute (all located in Berlin, Germany). In March 2014, Stratosphere was adopted by Apache Incubator, and got accepted as an Apache Software Foundation’s top level project in December 2014. Stratosphere’s main goal was to advance the state-of-the-art distributed, fault-tolerant data processing. In its 0.6 release (August 2014), Stratosphere was renamed to Flink, a general-purpose data processing engine for clusters. Throughout the years, more and more programmers joined Flink’s open-source community and have greatly contributed to its development. Flink is integrated by multiple big companies and enterprises around the world, such as Amazon, Alibaba, ebay, HUAWEI,…   1. **A deeper insight to the selected solution**    1. Major components and main functionalities:   The figure below shows the technology stack of Flink with 3 layers:    As illustrated in the figure, Flink does not come with a storage system. Users can integrate external databases/streaming systems (for example: HDFS, MongoDB, Hbase,…) and use Flink to collect and process data from those sources.  The lowest level of Flink is the **Deploy layer**. Flink can be deployed in 3 forms:   * Local – single node, single JVM * Cluster – cluster of multiple nodes, can integrate with all common cluster resource managers, such as YARN, Apache Mesos, Kubernetes, or as a stand-alone * Cloud – on Google’s GCE or Amazon’s EC2   The middle level of Flink is the **Runtime layer**, also the kernel of Apache Flink. This layer provides distributed processing with fault-tolerance, high reliability. Flink runtime consists of 2 types of processes:   * JobManagers: schedule tasks, coordinate checkpoints and failure recovery * TaskManagers: execute the tasks   The client is not a part of the Runtime layer, but is used to submit jobs to the JobManager  The last layer is **APIs and libraries**, providing 2 core APIs along with a lot more tools for different kinds of needs.   * DataSet and DataStream API: implement transformations on data sets, batch (using DataSet) or data streams (using DataStream) and return the output via sinks, which write the data to files or standard output, like command line terminal. * TableAPI: SQL-like expression language for relational stream and batch processing that can be embedded in DataSet and DataStream APIs. * Gelly: Graph API for Flink. It contains a set of methods and utilities that allow users to transform, modify graphs and simplify the development of graph analysis applications * FlinkML: new product from Flink’s open-source community, providing scalable ML algorithms and intuitive APIs to handle ML applications in Flink * FlinkCEP: stands for complex event processing. This library allows easy detection of complex event patterns in a stream of endless data   1. Its applications in academic and/or industry activities   Apache Flink excels in high-concurrency stream-processing with millisecond-level latency, high throughput, fault-tolerance. Flink is currently a top open-source stream processing engine in the industry. Dozens of companies use Flink to handle their processes involving real-time data, such as anomaly detection, business process monitoring, financial analytics, log aggregation,… Here are some examples of business-critical applications powered by Apache Flink:   * Amazon Kinetic Data Analysis: Amazon’s cloud service for stream processing. It uses Flink to power its Java application capability * Gojek: uses Flink to improve data-driven decisions across functions * OPPO: integrates Flink into its real-time data warehouse in order to “analyze the effects of operating activities and short-term interests of users” * SK Telecom: uses Flink for several applications, such as smart factory, mobility applications * Uber: built AthenaX, its internal streaming analytics platform, on Flink   Not only companies, many universities and research institutes around the world use Flink for research and educational purposes. For example: Technical University of Berlin, Leipzig University, University of Zagreb, Big Data Europe,…   * 1. Popularity   Over 50 companies around the globe, including big names like Amazon, Alibaba, HUAWEI, choose Flink to facilitate their business operations. A lot more universities and institutes are using Flink in their lectures, as well as doing research on Flink, mainly on its capabilities in the Big Data field.   * 1. Identify other solutions that have similar functionalities and compare them on different aspects   **Apache Spark**  Spark is an open-source, multi-language engine designed to process large amounts of real-time data, execute large-scale data analytics. It is deployed in a stand-alone mode or on top of many other distributed computing frameworks.  Spark is well-developed, has wider usage than Flink and is used by a lot more companies and enterprises. Flink, while being less mature than Spark, is superior in terms of performance, latency, scalability and automaticity. Both technologies are compatible with locally adapted applications in one unified region.  **Apache Storm**  Storm is an open-source, distributed, stream-processing system. Storm’s pipelined engine looks a bit similar to Flink, and they both aim for low-latency stream-processing. However, Storm does not come with batch capabilities, unlike Flink being suitable for both forms. Moreover, Flink offers higher level APIs, which means a lot of Flink’s APIs must be manually implemented when using Storm.   1. **Demonstration**   A good way to learn what we can achieve from Apache Flink’s excellent features set is to look into how companies and organizations are working with Flink.  **Alibaba**  Alibaba, the world’s largest e-commerce retailer, used a fork of Flink, called Blink to power Alibaba Search, its own search and recommendation platform. Here are some applications of Blink in Alibaba’s search infrastructure:   * ***Real-time A/B testing framework:*** Online logs (impressions, clicks, transactions,…) are collected and aggregated. Then, Alibaba engineers use Blink to perform OLAP analysis on the data to evaluate different search algorithms’ performance in real-time * ***Real-time feature updates:*** Alibaba’s search rankings system is a critical aspect to the search engine, contributing greatly to the company’s online revenue. With the aid of Blink, they can perform computations on real-time data like product click-through rate, product inventory, total clicks,… to give their customers the most relevant results. Flink’s powerful stream-processing cability gives a significant boost on the performance of this system * ***Machine learning (ML):*** There are days in the year where numerous products are discounted, such as Black Friday. In these days, user count and transaction increases dramatically, many times higher than in normal days. Normal ML models usually do not take these special occassions into account. To solve this problem, Alibaba utilizes Flink to build online ML models trained with real-time data, thus greatly increase performance on theses rare but important sale days.   **Boygues Telecom**  Boygues Telecom is the one of the largest mobile providers in France. They want to build a system that can perform analysis on log data, in order to search for indicators of user experience’s quality. This system, which they name as LUX (Logged User Experience), has to handle 2 billion events a day (500 thousand events per second) with a strict latency requirement of less than 200 miliseconds. After careful research, Apache Flink was selected to be the right framework for this project.    The Flink’s data transformation streaming topology receives the raw log data, then transforms and enriches the data before pushing it back to the system. From there, upstream systems will consume this data to perform analytics.  Flink’s low-latency, fault-tolerant and scalable engine makes it the best fit for Boygues’ requirements. Flink also supports visualization so the developers can understand how programs are running. Boygues Telecom is now exploring Flink applicability and potentials on other use cases in their business operations.   1. **Discussions and Conclusion**   Although Apache Flink is still somewhat new in the scene, there’s no argument that Flink is one of the top real-time processing frameworks. It’s hard to find another example with a combination of such powerful traits as Flink. Many big technology companies have adopted this powerful engine in production, and many more are planning to. Flink’s capabilities make it worth proper examination and studies, since it enables us to handle real-time data and computations on a large scale in a more flexible and affordable way than before. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Weekly schedule:** 〈what each member has done every week〉   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Week 01  (from 20/2 to 26/2) | Week 02  (from…to…) |  | Week n  (from…to…) | | Thái Cẩm Phong | Research Plan part 1 | Finish installation |  |  | | Nguyễn Khải Phú | Research Plan part 2 |  |  | | Nguyễn Duy Thịnh | Research Plan part 3 |  |  | | Đỗ Đạt Thành | Research Plan part 4 |  |  | |

|  |
| --- |
| **Group self-evaluation**   * **Advantages:** diligence, discipline, persistence * **Disadvantages:** the subject is unfamiliar; a variety of things to look for and study; new programming language |