Flink Notebook

Interactive power of Flink Streaming

Spring 2015

Flink Notebook

Interactive power of Flink

I. Introduction

1. The business idea

In the last few years Big Data generated a lot of buzz along with the launch of several successful big data products. Thanks to contribution from open source community and several giant Internet companies, the big data ecosystem has now reached a tipping point, where the basic infrastructure capabilities of supporting big data challenges and opportunities are easily available. Entering the next generation of big data - so-called Big Data 2.0 - two of its concentrated areas are Velocity and Applications, besides Data Quality. The cause for the former is that data is growing at an exponential rate and the ability to analyse it faster is more important than ever. For instance, sensors can generate data on millions of events per second and store all of those data and response in real-time is non trivial. The latter is helping to overcome the technical challenges of existing frameworks by making them easy to use and understand for everyone to benefit from big data.

As a result, the demand for ***streaming processing*** is increasing a lot these days. Processing big volume of data is not sufficient in the cases that infinite streaming data is arriving at high speed and users require a system to process fast and react to any incident immediately. In addition, although hardware price has plunged year over year , it's still expensive to equip a storage which is growing few terabytes every day for batch analysis. Streaming processing solutions are designed to handle high volume in real time with a scalable , high available and fault tolerant architecture. This enables analysis of data in motion.

One of the disadvantages to users is that many big data framework provide a rich imperative API codes only to process data stream on the fly. First, user must spend time learning API documentation properly since those APIs is fairly new to them. Therefore, the life cycle to develop products taking longer. Second, given that most big data applications are fairly simple application-wise, a block of API codes might be less optimal to use for most the popular queries. Third, the only way to share your work is to pack it as a library or service which is need to be deployed again. Fourth, the results

of streaming queries are keep changing upon the time. Thus, it demands a visual way to monitor the streaming instead of concrete array of numbers. Those above drawbacks apparently provide an unfriendly UX that inhibit both productivity and system performance. It is really against what we expect from "Application" aspect of Big Data 2.0.

We would like to promote "Flink Notebook", an interactive, collaborative web-based interface on top of Apache Flink which provide a complete streaming processing. Flink Notebook brings to developers a friendly, effective environment without scarifying the power of Flink Streaming.

Basically, Flink Notebook allows users to write standard SQL queries on webpages. The notebook issues corresponding commands to Flink to make it operate continuously on data as they arrive. The result is updated incrementally in real-time and displayed on Notebook with visual aid. In short, users need to know nothing about Flink API, except for very common SQL basis. The result is also brought to life by a powerful visualisation web pages.

Flink Notebook facilitates the productivity by allowing several colleagues to collaborate on the same notebook on the same work in real-time despite the physically location differences. The work can also be captured and exported to a portable format which can be viewed shared , restored later at ease.

2. My research background

Hint: just very shortly, why is it important, what happened before

// existing tools in Complex Event Processing (other name ? Marton), why still growing

During my 3-month internship I am working on Design and Implement an SQL dialect for Apache Flink, parts of the topic is related to online and streaming data processing and user UX. I realise an eager needs of a new application , which is presented in this minor thesis, to engage developers into building Big Data applications.

Fig 1: buzzword of Stream

Streaming Processing is not a new concept. Indeed the similar concept, Complex Event Processing (CPE) had been proposed from the 1990s by Event Simulation Research at Stanford [1]. Since that time, people have started generating a lot of different buzzword around it [2] and often reinventing ideas borrowing from other fields , but using a different vocabularies to describe the same concepts. Basically , the idea is to

analyse one or multiple data streaming to identify meaningful events and respond to them as quickly as possible.

According to CEP Tooling Market Survey 2014 [3], from 1996, there were more than 30 companies providing Streaming Processing solution. All the major software vendors (IBM, Oracle, Microsoft, SAP) also have good to excellent offerings in the CEP space for customers.

However, since a massive amount of data is growing rapidly every second, Hadoop is emerging distributed processing ecosystem today. Thanks to Hadoop, people can build a large scalable distributed system on Cloud. Even though Hadoop is designed to[1] [2]scale up to thousands of machines with very high degree of fault tolerance, it is optimised to run batch jobs with a huge load of computation. Because of time factor, Hadoop has limited value in online environment where fast processing is crucial. Therefore, existing CEP solution is barely compatible with Hadoop ecosystem. We demand a new sort of streaming framework which is able to integrate on top of Hadoop system. Apache Flink is one of these frameworks.

In term of cooperative work, iPython Notebook is web-based interactive environment where one can combine a working session into a single document. The document can be shared and converted to other common formats such HTML, pdf for quick review. Google Docs allows multiple users to collaborate on a document in realtime. Our Flink Notebook inherits both of these technologies to provide an interactive, collaborative environment along with super power of Flink.

3. Method to use in the Thesis

Hint: very shortly: literature review, some interviews, company consultations, my own major thesis research to use

YOU SPEAK HERE ABOUT A PROPOSED PROJECT. USE PROPER WORDS TO MAKE IT CLEAR THIS IS ONLY A PORJECT PORPOSAL: HERE YOU WILL NOT DEVELOP ANYTHING!

A project developing Flink Notebook requires us to have a deep understanding on Flink Streaming , interpreter theory and web programming skill. First, we study the principle of Stream Processing and data flow in Flink. Second, we need to design SQL-like syntax which user can use to issue command to Flink. Third, a web application helps to interact with user.

The project involves many big data developers during design process. According to regular project management methods , we have to observer , to monitor their behaviour during testing , also we need to interview them for valuable feedback.

[1] Hadoop is a free, Java-based programming framework that supports the processing of large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation

Finally, to shorten the developing cycle, we take the advantage of community to integrate many open source project into ours

II. The product environment //TODO

Present a short, concise, selected analysis of existing research which is relevant to your topic. It has to justify how your investigation may help answer some of the questions in this area. Your thesis review can't be a straightforward summary of everything you have read on the topic or even a chronological ("monographic") description : keep it in form of 2-3 main articles / models and use max. 3-4 more specific papers.

Important: After citations of 2-3 articles, opinions, results, please compare and criticize the information collected — contrast your approach, stress weaknesses or strengths in their methods used, findings published around, etc.

Hint:

literature research on compilers / streaming data and referring to the major thesis;

Software development methodologies, comparison to these;
1-2 GENERAL articles and 2-3 SPECIFIC ones to this topic;
have an opinion summarised about these articles

just place a short history plus definitions here; referring even to common textbooks

1. Data Stream
2. Continuous Query Language
3. CQL for Flink

Operational transformation

III. Marketability of the Idea

1. Market Needs

First and foremost, we take a glance on Big Data promising forecast in global market. According to well-known IDC forecast [4], the big data and analytics market will reach *$125 billion* worldwide in 2015. Rich media analytics such as video , audio and image will contribute as a crucial driver to prompt a rise in big data applications. 25% of top IT vendors may announce Data-as-a-Service solutions based on expanding cloud ecosystems (expected to reach $118 billion in 2015).

One of the major business lines, which could not be adopted successfully without Stream Processing supports, is Internet of Things (IoT). IoT back end as a service (BaaS) will emerge and require a mature Stream Processing platform. Expenditure on IoT may exceed $1.7 trillion, up 14% from 2014 and go up to $3 trillion by 2020 . In other words , it is growing with a five-year CAGR of 30%. Therefore, we expect IoT would bring a huge slice of the cake to Stream Processing in revenue and development.

How organisations embrace the Big Data trending? The report 'Big & Fast Data: The Rise of Insight-Driven Business' conduct by *Capgemini*, surveyed 1,000 senior decision makers in nine regions and nine industries. It reveals the fact that 65 % of organizations acknowledge they are at risk of becoming uncompetitive unless they embrace new data analytics solutions. Specifically , accessing Big Data faster is where C-suite executives see the most value – 77 percent state that decision makers increasingly require data in real-time. However, More than half (52 percent) of respondents reported that developing fast insights from data was hampered by limitations in the IT development process.

In short, the market is in strong demand of real-time Big Data Platform , along with supported tools to provide better development process.

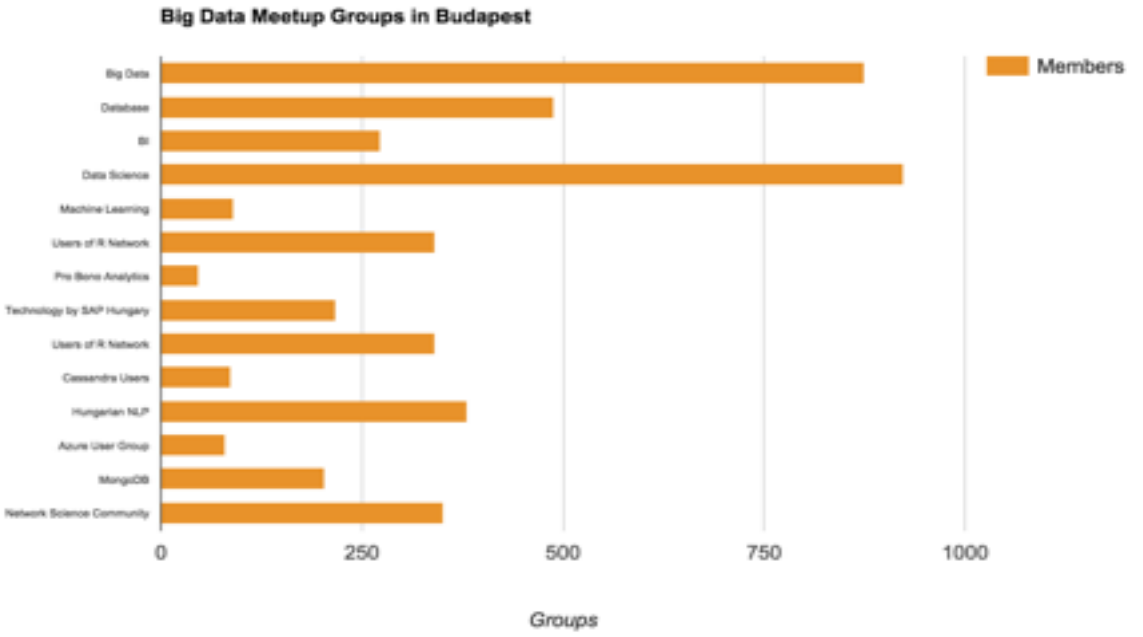
Fig 2: German Market

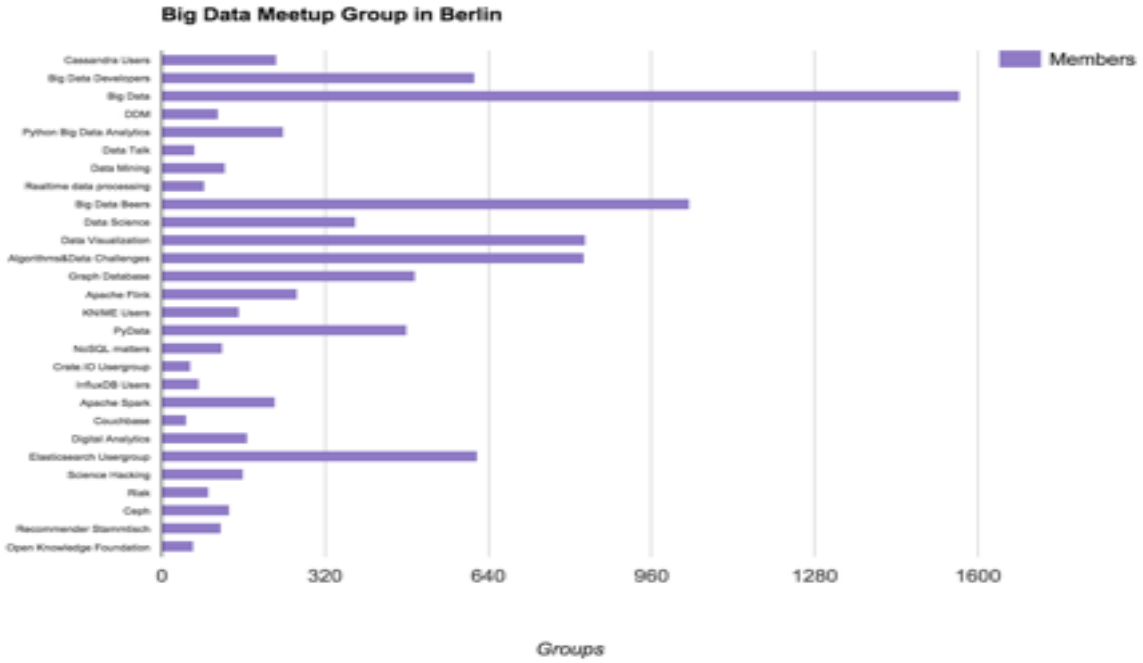
We are going to supply services to the German and Hungarian market at the beginning. The hint is that our teams are based on Berlin and Budapest so that we are capable of nurturing our initial market at the beginning. Even though the German Big Data market still appears to be at an early stage, it is expected to from EUR 650 million in 2013 to almost EUR 1.7 billion in 2016. Additionally, Software and Services would account for 70% of that value.

We did a research on [3]*Meetup* groups using their API to see how active Big Data and Data Analytics community is.

In Berlin, we found about 30 Data-related groups with more than 9300 members in total. In Budapest , we are also able to reach about 4700 members in total. Even each developers can join more than 1 group but the number is till quite considerable.

[3] http://www.meetup.com





2. **Competitors**

According to the CEP market players to end of 2014, conducted by Paul Vincent and his team, CEP tools have a long history (Fig 5) but many of them are acquired or left the market. Almost tools and platform support query languages which we also provide. Even all the major and medium software vendors offer their own solution but we target to the open source community with some emerging top projects such as Apache Storm , Apache Spark and Apache Samza.

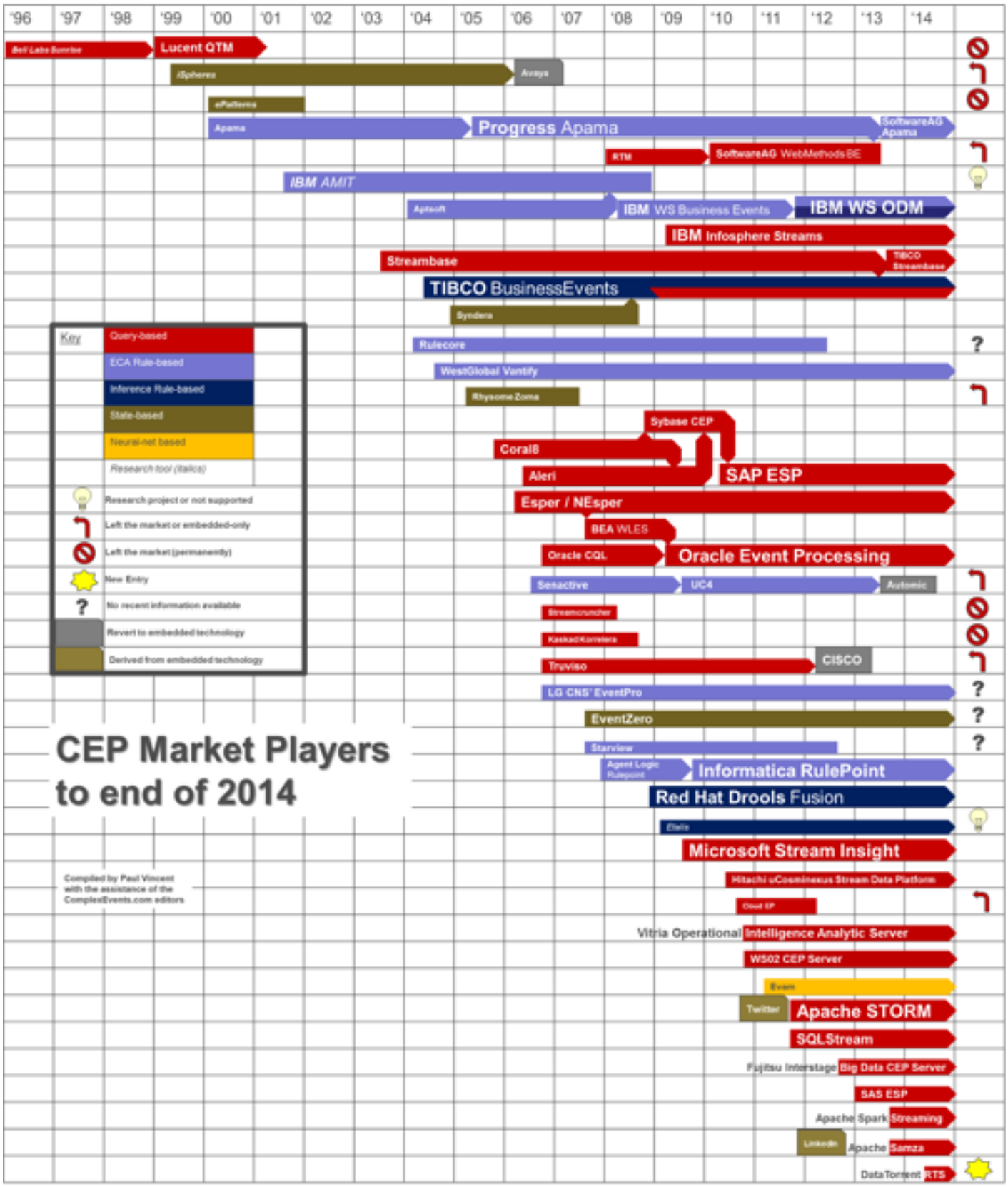


Fig 5: CPE Market Players to end of 2014

We briefly compare those tools to our solution in term of Streaming Processing in Table 1:

	Storm	Spark	Samza	Flink	(note)
Delivery Semantics	>1	1	>1	1	smaller is better
State Management	Stateless	Stateful	Stateful	Stateful	stateful is better
Latency	sub-second	second	sub-second	sub-second	smaller is better
Language support	Any	Scala, Java, Python	Scala, Java	Scala, Java	more is better
Query Language Support	Yes	Yes	Yes	Yes	
Notebook features	No	No	No	Yes	

Table 1: Direct Competitors

*Notebook features: *collaborative*, *interactive interface* ; *file format to share*

3. Value proposition

Hint: what we can offer
We would offer a difference to the above products; it will be a
- compiler, easy-to-use; more effective, etc etc
- a „Notebook" to use as a manual

Customer pains:

Working on most of the existing Big Data frameworks, developers encounter several disadvantages:

- They need to spend time learning API via documentations. The API is non-standardized and may not be well-designed , thanks to the committers' expertise. In addition , for most popular user-cases, the programs are conducted from API by developers may be less optimal in term of time and space cost.

Fig 6: Imperative code

- Result from querying data stream is continuously emitted, users have hard problems to observe the change to react.

- It is not easy to share your work s with colleagues except for packed

library or services with additional documents

Customer gains:

- All of developers is familiar with SQL syntax which is widely adopted as a standard for data query language. Therefore, It would be much more efficient if they could issue command to Big Data application using SQL-like syntax. No much effort to learn that syntax
- A visualized output would make users easier to examine and monitor the flow of data
- Developers wish to have a portable file to wrap up their work, share to people and show how the output is expected. Need not to recompile the program.

Products and Services

To relieve our developers' pain and accelerate their work, we would like to introduce Flink Notebook with several advanced features:

- It is a *web-based* UI environments helping users interact with Flink platform. As long as we are able to connect to Flink, we can run Flink notebook *everywhere with a browser*.
- Command the system with *SQL* syntax.
- An *interactive* UI : users are able to type their command and receive the result as real-time charts.
- Flink notebook save your work (commands and visualized outputs) to a portable **.fnb* file. We can *export* it, *hand over* to colleagues. They *could open the file at ease in any browser.* If your machine do not connect to Flink infrastructure, you can see all of issued command and its output non-interactively. Otherwise, you are empowered to re-run all of authenticated command and observe new results.
- Users can invite other people to work on the same *.fnb format collaboratively regardless of different physical location.

IV. The project proposal

1. The project idea

Hint: Repeating in details the starting text: why is it beneficial for the company to have this development

Stream Processing have been a very active and diverse area of research, commercial & open source development since 1990'. Along with traditional business event which is streaming stock data for financial services, streaming satellite data for the military or real-time vehicle-location data for transportation and logistics businesses , companies in multiple industries must handle large volumes of complex data in realtime.

Fig 7: What happens on Internet on 60 seconds

However, Stream Processing gains more and more attention recently, thanks to the explosion of mobile devices and trendy Internet of Things. The number of smartphone users in 2014 is more than 1.64 billion which account for 38.4% of mobile phone users[9]. According to market research in previous section, expenditure on IoT may exceed $1.7 trillion and reach to $3 trillion in 2020. The enormous number of Internet-connected devices and the ubiquity of high-speed connectivity add to explosion of mobile data. Figure ? depicts the number of events happens in 60 seconds on Internet. Even though the system receive an extremely high velocities of data and event throughput, users expect to receive response immediately without any delay. Those challenges can only be solved by a stream processing system, instead of traditional database technology.

At the same time, demand for business process agility and execution has only grown. Useful tools are largely available but inhibit the development process because of its poor User Experience. Most of them require user to write complicate API code, which is not trivial to many users. Scientific users tell IDC that these tools can be great for retrieving and moving through complex data, but they do not allow researchers to take the next step and pose intelligent questions.

Sophisticated tools for data integration and analysis on this scale are largely lacking today. There are opportunities to create tools and applications for Big Data. Vendors

that create tools and applications for use at this scale can use them as a lever to seize market leadership positions in the Big Data market.

We are bring out Flink Notebook to provide users a truly collaborative and interactive interface to help them deploy their solution as soon as possible. Furthermore, the beautiful Flink Notebook already hide the complicated but powerful analytic platform of Flink. The integrated solution bring users a powerful machine with beautiful UX.

2. The company background

Data Artisan: a spin-off company from Technische Universitat Berlin, Germany. The founder team has a high level of expertise in distributed data-intensive system. They are aiming to develop the next generation technology of Big Data Analytics and Processing via Apache Flink.

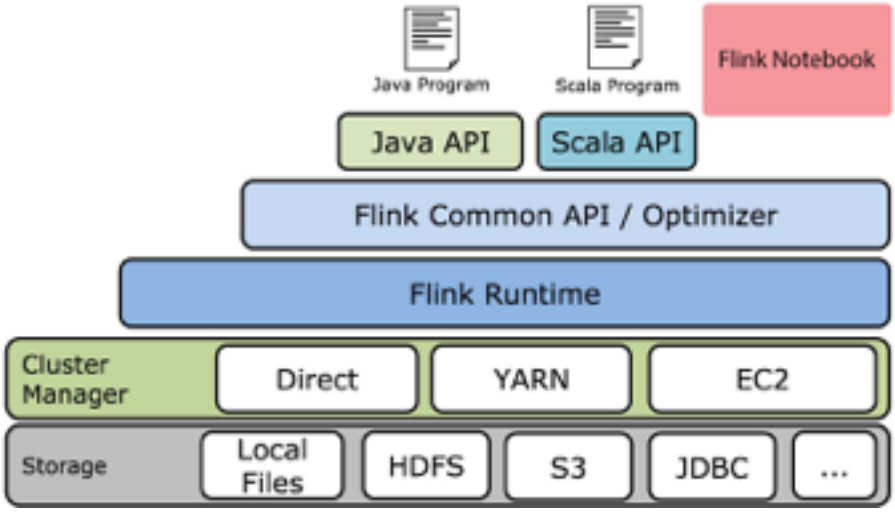


Fig 8: Apache Flink stack

Apache Flink: is a top-level project of the Apache Software Foundation with a community of over 100 contributors from industry and academia. Flink is an open source system for expressive, declarative, fast, and efficient Big Data analysis. It originates from EU-granted Stratosphere project since 2000s.

Basically, users are able to write a java/scala program on Apache Flink but it was painful and complicated as we mentioned before. *Flink Notebook* is considered as an application written on Scala or Java API. It receive user's commands as input and generate Flink API code automatically. User need not to care about Java/Scala API anymore.

We are highly competent engineers on Distributed System in general and concurrency in particular so that with only two engineers we can build up the collaborative feature at ease. However, Flink notebook serves a web-based application, thus we need a small team of one web developers and one web designer whose skills are complementary to ours.

3. Feasibility and Revenue models

Hint: Activities and resources needed for the project
SOME SENTENCES ABOUT PROJECT MANAGEMENT; ICT projects;
development process: resources (mandays, experts, software to use,
hardware needs, time, money, etc.
Marketing / promotion activities: building a website; pricing the tool
compared to other tools, etc

// CREATE A TABLE, A MATRIX HERE FOR SAY, 4 QUARTERS (ONE YEAR PROJECT), WITH ACTIVITIES, KEY FIGURES
If you create a 10-15 element in an Activity List like

1) Project Idea accepted

2) Activities detailed

3) Resource list accepted

4) Staff and infrastructure deployment

5) Development process

6) Testing and demo version distributed etc

and ADD a paragraph about price and revenue modelling,

a. Revenue Streams

//For what value are your customers willing to pay?

//What and how do they recently pay? How would they prefer to pay?

//How much does every revenue stream contribute to the overall revenues?

Be a part of Apache Flink, we are following the Open-source business model. We warmly welcome and encourage users to use our Flink Notebook freely. Additionally, we would like to support users to make the best out of Flink Notebook. Besides public supports on forum and mailing group, customers may require private supports from our dedicated experts.

Customers get support on:

- Trouble-shooting and reporting errors.
- Refactoring data and architecture model.
- Training on collaborative development cycle with Flink Notebook.
- Rich resource and materials

Customers can choose either session supports and annual subscription.

- Session supports, we charge customer according how many hours of support. The fee is about 50$/hour
- Annual subscription plan , customer receives 24/7 supports within a year which cost s 800$ /year.

b. Key Activities

//What key activities does your value proposition require?

//What activities are important the most in distribution channels, customer relationships, revenue stream…?

We put most of our resources to build a product which bring the best experience to users. Besides, marketing strategy is rather important to engage users and enlarge our communities.

Design and development

We strictly follow agile methods for building software to eliminate the cost during development. The process is broken into a consequence of 10-day *sprints*. The duration of a sprint depends on the difficulty of problems but no longer than 3 weeks. Each sprint consists of 5 steps:

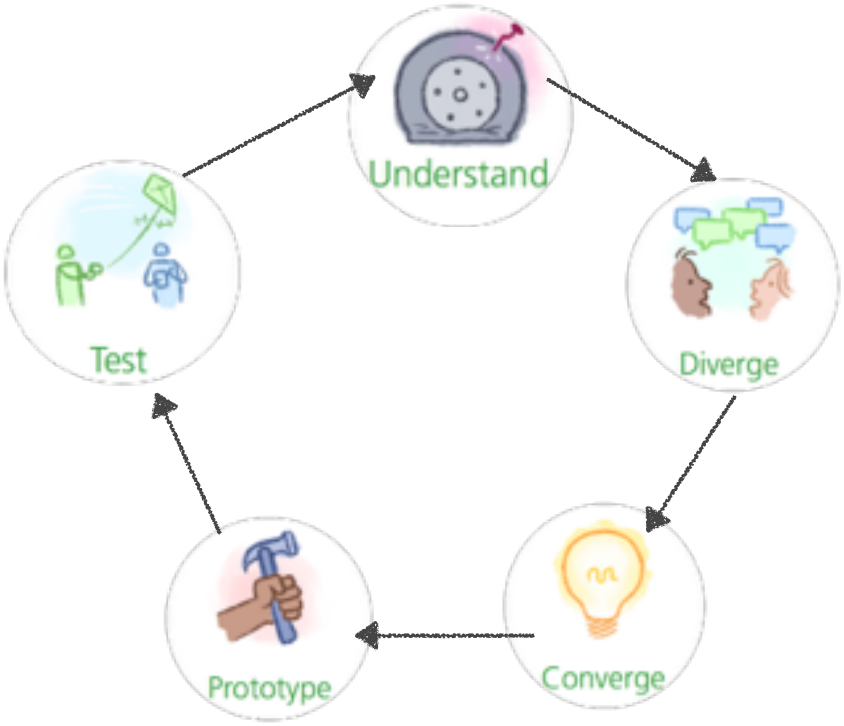


Fig 9: Design Sprint

To improve the quality of codes, we are applying Test-Driven Development method (TDD) . Instead of writing code at the early stage, we first write an automated test case that defines a desired improvement or new function, then produce the minimum amount of code to pass the test, finally refactor the program to acceptable standards.

	Duration	Desctiption	Tools
1. Understand	1 - 3 days	Discover insights and define the problem	Facts Assumptions, Discovery interview, Task Modeling, Empathy Map, Journey Maps, Who/Do
2. Diverge	1/2 - 2 day	Generate solutions	Challenge Maps, Scenarios, Storyboarding, Six-ups, Who/Do
3. Converse	1/2- 2 day	Rank and select solutions	Assumption Table, 3-12-3, Storyboarding, Dot-voting
4. Prototype	1 - 3 days	Build the solutions	Illustrator, Photoshop or Sketch, HTML/CSS, Keynote or PowerPoint, coding
5. Test	1/2 - 1 day	Observer the customer impact	• Testing Interview • Plus/ Delta

Table 2: Sprint design in details

Passing all test cases ensures that our program satisfies all of our designed requirements and minimise the number of bugs.

To collaborate on the project, we utilize [4]G*ithub* to host the source code repository and [5]Jira to capture and organize issues, assign work, and follow team activities.

Marketting strategies

Aligned with Apache Flink, we can propagate the idea of Flink Notebook through various Big Data Conferences (Strata + Hadoop , Apache Conference), invited talks & tutorials at *Meetup* group. As business model we also give 3 months of free support for the first 30 customers.

Building a screen cast channel on Vimeo, Youtube to give away basic tutorials and tips.

To stand out from other frameworks, we also establish an expert/evangelist network around the Flink to help raise many discussions/debates on the features on Flink in

[4] http://www.github.com

[5] https://www.atlassian.com/software/jira

general and Flink notebook in particular. We manage to keep Flink Notebook as a hot topic and mentioned frequently.

c. Key Resources

//What key resources does your value proposition require?

//What resources are important the most in distribution channels, customer relationships, revenue stream…?

Manpower: We already have a vast support from academic and industrial community to develop Flink (20 official committers, 100 contributors, academic supervisors from TU Berlin, Sztaki, KTH, Apache supports), but we delicate team of 3 full-time developers and 1 designer to make Flink Notebook developed.

Flink Notebook is also backed by a robust, sustainable developer community and the governance framework and processes of Flink and Apache Software Foundation.

Finance: Euro 20.000 for the team operation in 6 months.

d. Key Partnerships

We are glad to make a partner with Google to bring easy Flink deployment to the Google Cloud Platform, and enable Google Cloud Dataflow users to leverage Apache Flink as a backend. Now users even can use Flink under Google Cloud platform. And Flink Notebook would be integrate into it soon.

e. Cost Structure

Cost during development period:

Fixed Cost in Euro	**Cost (monthly)**		**Virable Cost**	**Cost**
Salary	2800		**Conference Fee**	~ 200 euro/ person/ conference
Office Rent	200			
Utilities	100			
Jira subscription	20			
Server & Domain	50			
Total	**3170**			

Table 3: Cost structure

4. Project planning

Hint: A simple scheduling of activities (based on list of 3.5); participants, milestones and deadlines; all in a visual plan like Gantt-diagram
Management problems: which type of project management is needed

and create a simple table how scheduling will go forward is weeks or in months

	05/01/15	10/30/15
Milestone 1 - Run SQL command via web form	05/01/15	06/02/15
Milestone 2: interactive web page	06/03/15	08/14/15
Milestone 3: collaborative web page	08/17/15	10/30/15

Fig 10: Milestone

// Milestone 1 : 3 sprints

// Milestone 2: 6 sprints

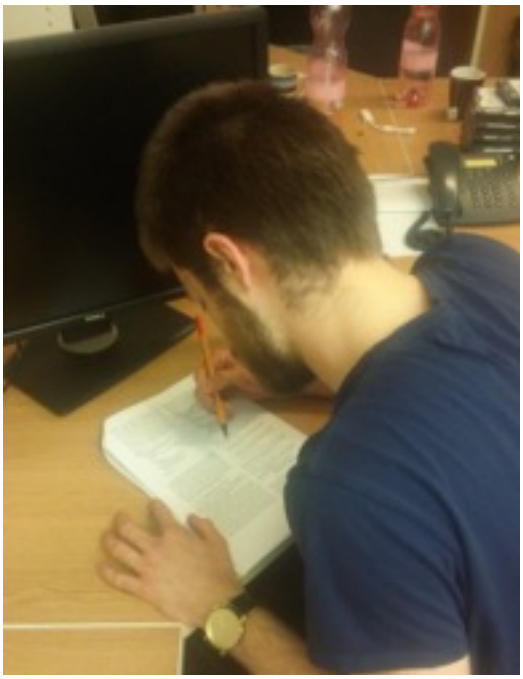
// Milestone 3: 6 sprints

5. Customer Feedback Subproject Proposal

Hint: Here you can write some words on what you could do: idea planning, programming, testing, presentation to beta-version users, etc

// Feedback via User Rating && Quizz

// Users: 2 developers at Sztaki



V. Summary //TODO

Hint: What was the target; how I worked, what is the result; recommendations to many internship company management; and: What I have learnt form this Thesis Work…

[1] "The Stanford Rapide™ Project", Stanford, retrieved on May 1, 2015

[2] Kleppmann, Martin. "Stream Processing, Event Sourcing, Reactive, CEP... and Making Sense of It All". January 29, 2015. Retrived on May 1,, 2015

[3] Vincent, Paul. "CEP Tooling Market Survey 2014" . December 3, 2014 . Retrived on May 1, 2015

[4] Pree, Gill. "6 Predictions For The $125 Billion Big Data Analytics Market in 2015" Forbes Tech. Dec 11, 2014. Retrieved on May 2 2015

[5] "New global study by Capgemini and EMC shows Big Data driving market disruption, leaving many organizations fearing irrelevance". PRNewswire. March 10, 2015. Retrieved on May 3 2015

[6] Siciliani, Tony. "Streaming Big Data: Storm, Spark and Samza". Java Dzone. Feb 2 2015

[8]Otávio M. de Carvalho, Eduardo Roloff, and Philippe O. A. Navaux. *A Survey of the State-of-the-art in Event Processing*. 11th Workshop on Parallel and DistributedProcessing (WSPPD), August 2013

[9] "2 Billion Consumers Worldwide to Get Smart(phones) by 2016". Emarketer. Dec 11, 2014. Retrieved on May 4 2015

http://whathappensontheinternetin60seconds.com/

http://whatsthebigdata.com/2014/07/18/what-happens-on-the-web-in-60-seconds-infographic/

http://businessmodelalchemist.com/blog/2012/08/achieve-product-market-fit-with-our-brand-new-value-proposition-designer.html

http://www.forbes.com/sites/louiscolumbus/2012/10/15/using-search-analytics-to-see-into-gartners-232b-big-data-forecast/

http://www.gartner.com/technology/research/it-spending-forecast/

http://www.prnewswire.com/news-releases/nosql-market-is-expected-to-reach-42-billion-globally-by-2020---allied-market-research-498476751.html

---

http://www.bi.hu/index.html

http://www.bi.hu/2015/03/30/a_bi_platformok_vezeto_szallitoi_2015ben.html

http://bievkonyv.hu/2011/evkonyv.shtml

http://bievkonyv.hu/2011/konferencia.shtml

https://www.pac-online.com/big-data-segments-market-figures-hungary-0

http://www.fujitsu.com/hu/solutions/business-technology/bigdata/

———————-

When I was working on Google Apps, we would often hear people ask, “Why launch Google Spreadsheets? It’s 20 years behind Microsoft Excel and 200 features short!” They didn’t realise that a driving mantra for Google Apps was “It’s the collaboration, People!” I have seen metrics, and still experience daily, how Google Apps’ real-time collaboration features boost team task productivity by a factor of 10x or more. It is collaboration among team members with diverse skill sets and points of view that yields these large gains in organizational smarts.

———————

it hard to do rich data exploration like you can do in MATLAB. So, we built and are open sourcing Scala Notebook — an interactive Web-based Scala evaluator.

Based on iPython Notebook, it lets you evaluate Scala expressions as with a REPL, but also supports rich HTML, graphics and persistent documents. It even offers two-way interactivity by connecting an FRP to Knockout JS. We'll demo the product, talk a bit about how it's built and how we're planning to evolve it on github.

———————-

Easy to view and edit past commands

Commands can return HTML or images, allowing richer interactivity (charts, for example)

Notebooks can be saved and loaded, providing a bridge between interactive REPL and classes in a project

Supports mixing Scala expressions and markdown, letting you create rich, interactive documents similar to Mathematica”’

Company:

Companies
actuated
Alteryx
Applix
Business Objects
Cloudera
Cognos
Comshare
DatAllegro
EMC
Eclipse BIRT
Greenplum
HP
HortonWorks
Hyperion
IBM
Infobright
Informatica
JasperSoft
Jedox Palo
KarmaSphere
MicroStrategy
Microsoft
Netezza
, Oracle
Panorama Software
ParAccel
Pentaho

QlikView

Rapid Miner

SAP

SAS

SPSS

Splunk

Sun

Sybase

TIBCO Spotfire

Tableau

Talend

Teradata,

Vertica

MODIFIED Thesis frame DUY **22nd March 2015**

1. Introduction

   1.1. The business idea: online processing of streaming data needs a new solution,
   a compiler ; the Thesis is offering a **PROJECT** to develop this tool
   terms, short idea

   1.2. My research background
   just very shortly, why is it important, what happened before

   1.3. Methods to use in the Thesis
   very shortly: literature review, some interviews, company consultations,
   my own major thesis research to use

2. Marketability of the idea

   2.1. The product environment: literature research on compilers / streaming data

and referring to the major thesis; Software development methodologies, comparison to these;
1-2 GENERAL articles and 2-3 SPECIFIC ones to this topic;
have an opinion summarized about these articles

2.2. Market need: existing customer problems
statistical data, how many project, programmers, etc could be interested
„I just focused in the Thesis to the Vietnamese market: product specialities
can be used at approximately XXX. Software development companies in VietNam

2.3. Competitors: similar products and services
List and put into a matrix all known parameters of leading products

2.4. Value proposition: what we can offer
We would offer a difference to the above products; it will be a
- compiler, easy-to-use; more effective, etc etc
- a „Notebook" to use as a manual

3. The ~~Business Model~~ Project Proposal

3.1. The Project Idea
Repeating in details the starting text: why is it beneficial for the company to have this development

3.2. Customer segment, targeted markets
here you can talk about „two specific markets, Italy, VietNam"
or number of small developers, etc etc as you like

~~3.3. Channels to build a user community on these markets~~

~~3.4. Contacting customers: customer relationship solutions, promotion~~

3.5. The company background

here you can write about the company, today product portfolio, etc
why they are interested in a new product

3.6. Feasibility: Activities and resources needed for the project

SOME SENTENCES ABOUT PROJECT MANAGEMENT; ICT projects;

development process: resources (mandays, experts, software to use, hardware needs, time, money, etc.
Marketing / promotion activities: building a website; pricing the tool
compared to other tools, etc

3.7. Project Planning

A simple scheduling of activities (based on list of 3.6.); participants, milestones and deadlines;
all in a visual plan like Gantt-diagram
Management problems: which type of management is needed

3.8. My participation in the proposed „Customer Feedback Subproject Proposal"

Here you can write some words on what you could do: idea planning, programming, testing, presentation to beta-version users, etc

~~3.9. Building a brand MAYBE~~

~~*however being small, any possibility to use a partner to be a „branded product"*~~

~~3.10. Revenue model~~

~~*pricing ideas: there will be a lecture about; building a brand*~~

~~3.11.~~

4. Summary

   What was the target; how I worked, what is the result;
   recommendations to mny internship company management;
   and: What I have learnt form this Thesis Work…