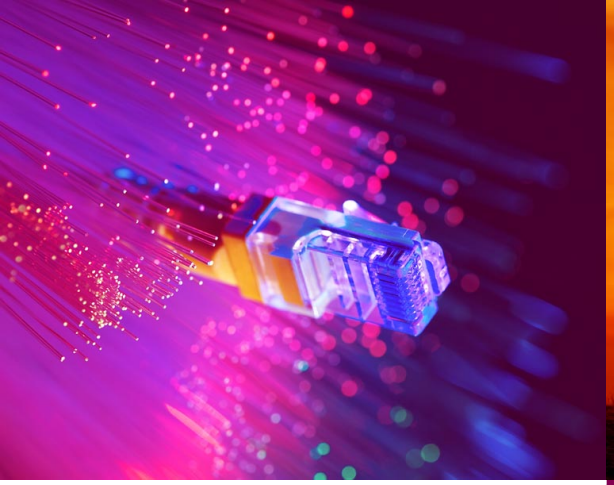


Webinar: Big data series: Apache Spark @ CSC - Introduction

Apurva Nandan

CSC – Finnish research, education, culture and public administration ICT knowledge center

Big Data: What's that?

Example – eCommerce company

- A basic approach to analyze the data in order to know about the customers – analyze the transactional data.
- The customer transaction activity should tell us what different customers are buying.
- But that's not enough, is it?
- What happens if the transaction never takes place?
  - Why did it happen?
  - What can the company do in that case?
  - **LOOK FOR NEW SOURCES OF DATA**

Big Data: What's that?

More sources of data:

- **Clickstream data:**
  - Storing and analyzing 'clicks' of visitors on a website.
  - Understanding their behavior
    - For eg. What other products were they looking at
- **Data from Shopping cart**
- **Social networking data:**
  - From all major social networking sites
  - Find out likes and dislikes of a customer
    - Finding even more customers that you don't currently have

- **Sensor data:**
  - Location informaton from smartphones
  - Can show regions where the product is used, where it isn't. Can also be used for monitoring supply

1 GB text file

100 GB text file

10 TB text file

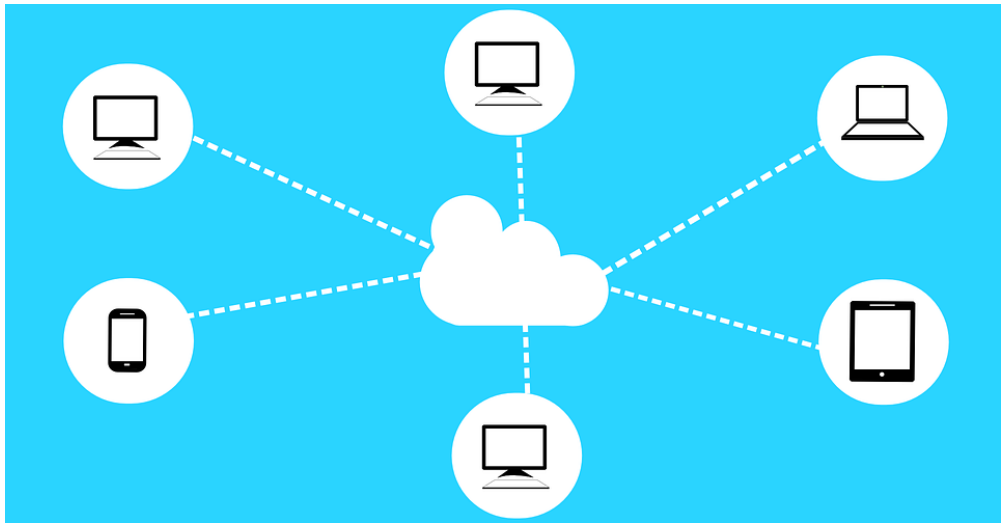
CRASH!!!!!

CRASH!!!!!

Analyze this

Cloud computing

- Computing resources accessed from virtual online 'cloud' rather than your own laptop or PC
  - CSC Cloud: pouta.csc.fi
- **Virtual Machine(VM)** – machine running in cloud, can be accessed by laptop/PC



Spark - Overview

Apache Spark

- Apache Spark is a fast, Open Source, big data based engine for large scale data analysis and distributed processing
- Can be Used with Scala, Java, Python or R
- Works on the Map-Reduce concept to do the distributed processing
- *Why should you care?*

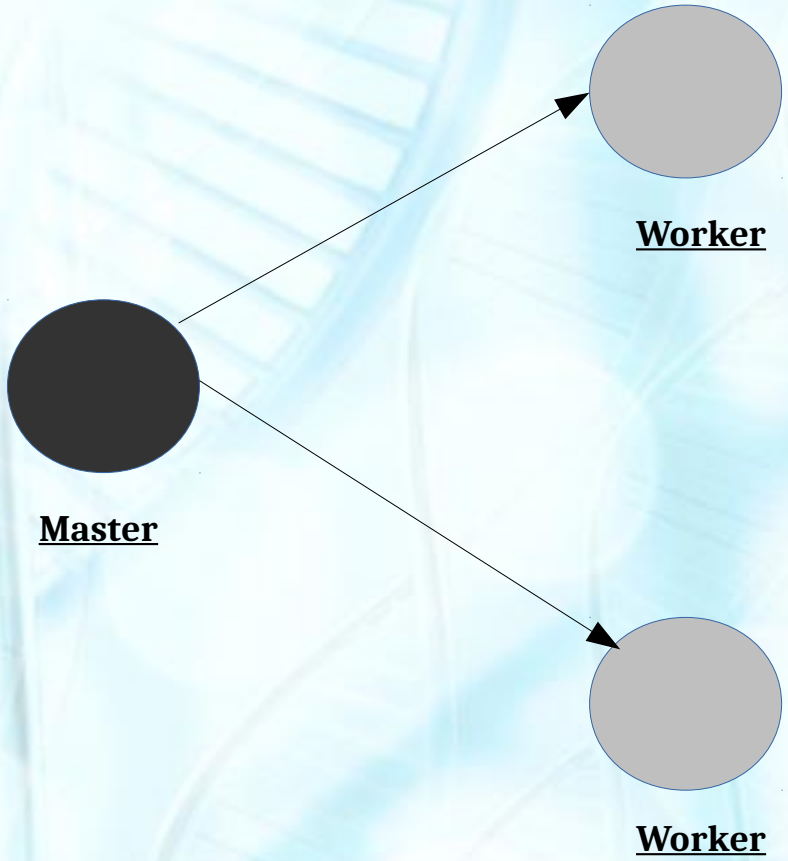
Because ..... DATA IS GROWING!

Apache Spark: Key Concepts

- **Stores the data into memory when needed, for rapid processing**
- Follows distributed processing concepts for distributing the data across the workers/nodes of the cluster

Apache Spark: Key Concepts

- **Cluster** : Set of machines grouped together
- Node: Individual machine in a cluster
- **Master**:
  - ➔ Entry point of a cluster
  - ➔ Delegates requests to workers
- **Worker**/Slave: Carries out the processing

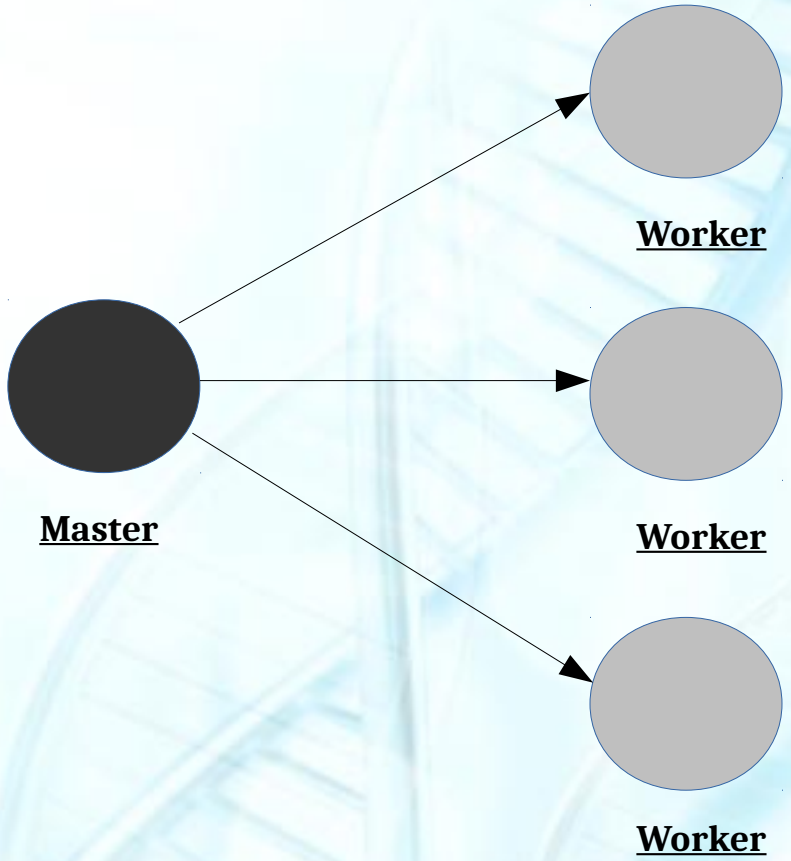


Why Run Spark on Cloud?

- You can request as much resource as you want
- You can pay per usage so adding more resources or deleting it would be flexible
- You can access it from anywhere
- You can use the same cluster for a group

Spark on Cloud: Scenario 1

- Worked well with 6GB
- Data increased to 9GB
- Processing time too high
- Add new worker



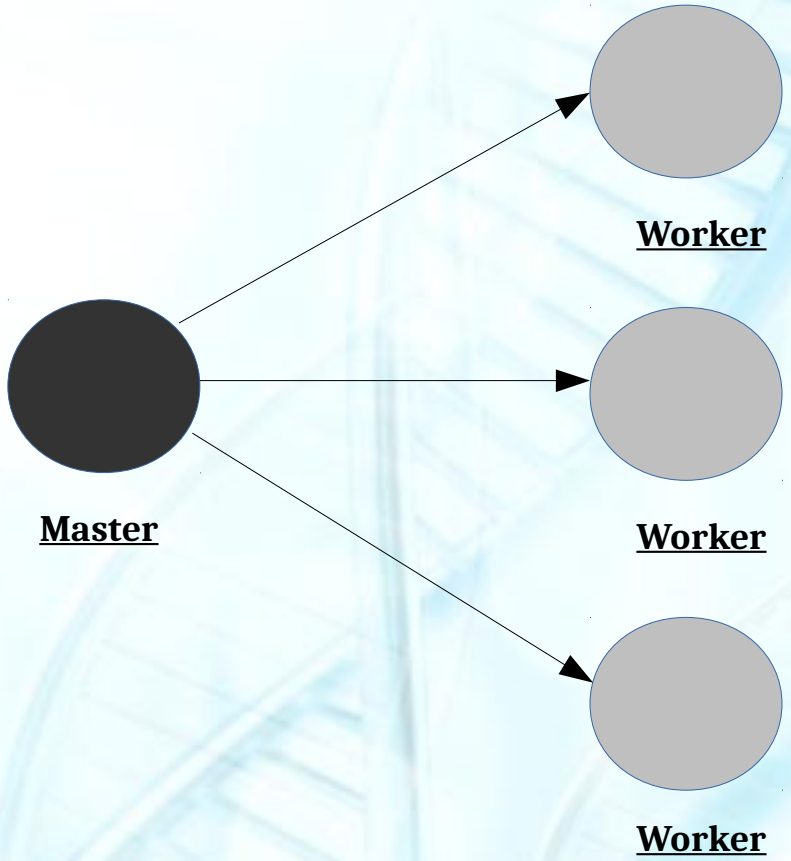
Each Worker has same configuration!

Worker config needs to be decided when deploying the cluster for the first time!

For eg.
Memory: 4GB
Cores: 4

Spark on Cloud: Scenario 2

- Works well with 9GB
- Data decreased to 6GB
- Processing time OK
- Delete Worker, Why?
- Save resources & Cost!



Each Worker has same configuration!

Worker config needs to be decided when deploying the cluster for the first time!

For eg.
Memory: 4GB
Cores: 4

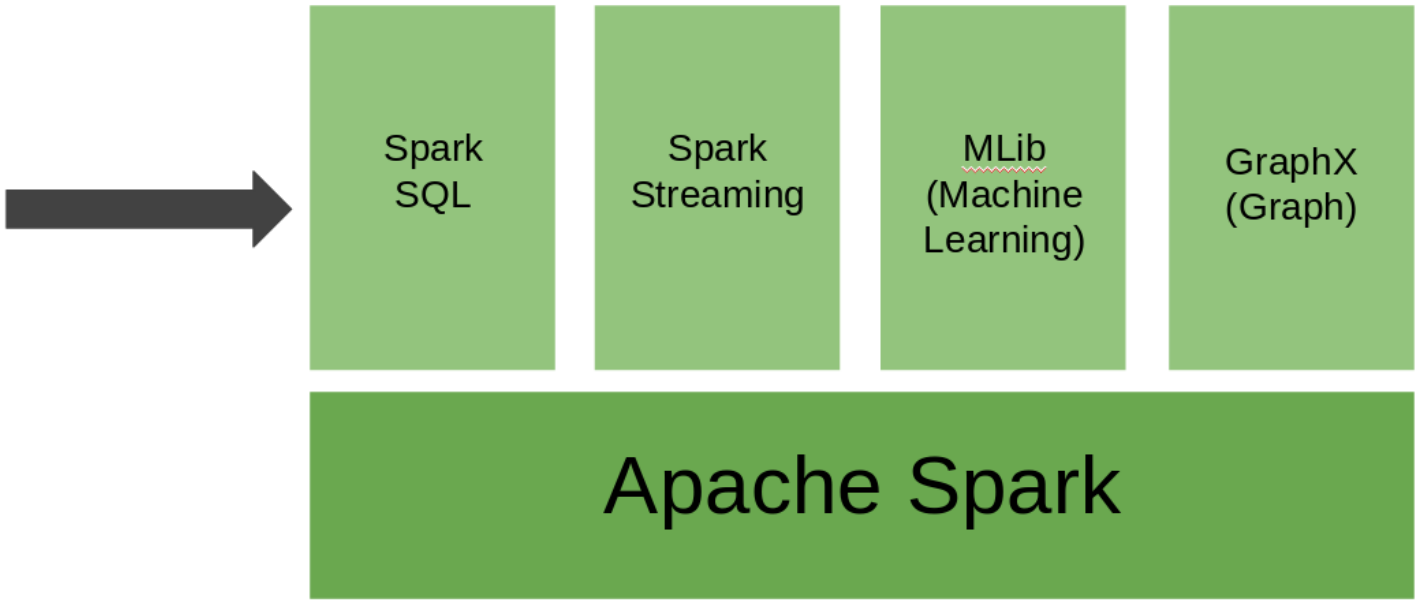
So, What's the point?

- Imagine you have a large VM for the processing and then things happen:
  - ➔ You realize you do not all of it, after sometime
  - ➔ OR, You need even a bigger VM
  - ➔ OR, It fails all of a sudden, ruining everything
- *“Ok, so I will just start with one Worker for now and wait to add more”* –
  - ✗ *Don't!*
  - ✗ *Because….*
  - ✗ *WTF! – Workers Too Fail*

Who uses Spark?

- Finance
- E-commerce
- Healthcare
- Media & Entertainment
- Travel industry

Spark Stack

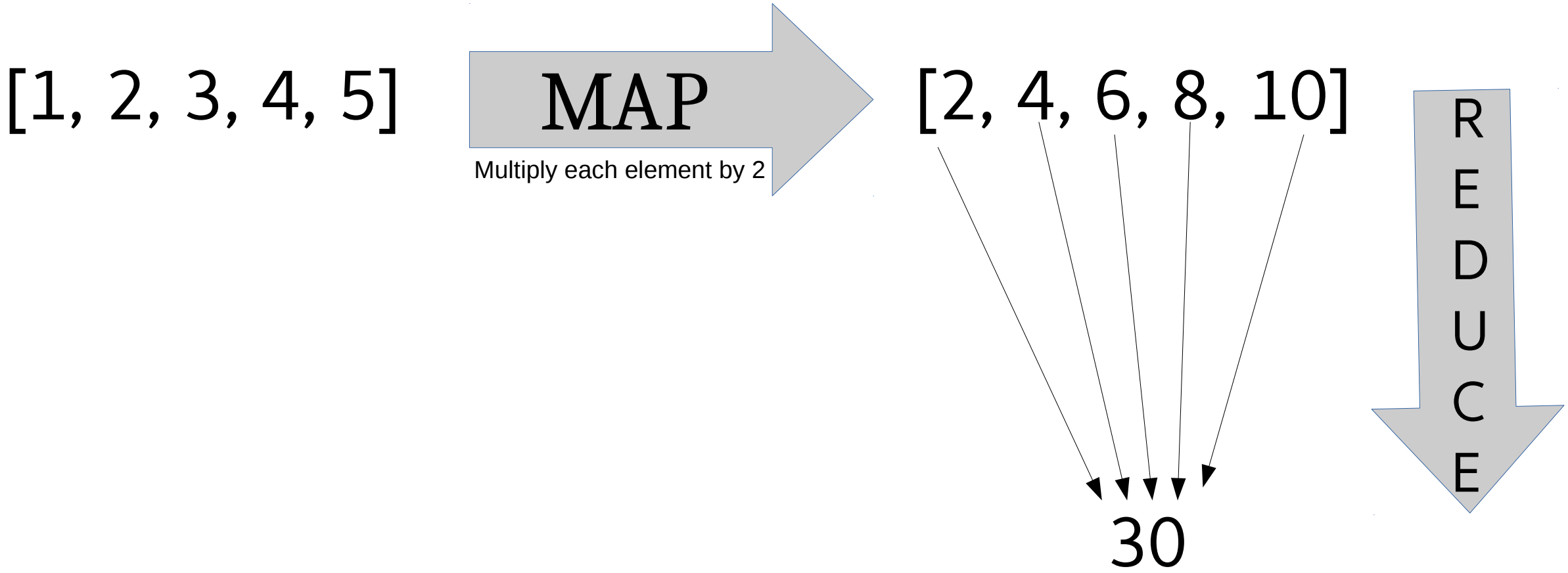

Spark SQL
Spark Streaming
MLib (Machine Learning)
GraphX (Graph)
Apache Spark

Spark Programming

Map-Reduce Paradigm

- Programming model developed by Google
- As the name suggests, there are two phases for processing the data – Map and Reduce
  - Map: Transform each element in the data set
  - Reduce: Combine the data according to some criteria
- Spark uses the Map Reduce programming model for distributed processing

Map-Reduce Paradigm



Machine learning in Spark

Spark Mllib – Machine learning in Spark

- **Classification**: logistic regression, naive Bayes,...
- **Regression**: generalized linear regression, survival regression,...
- **Decision trees**, random forests, and gradient-boosted trees
- **Recommendation**: alternating least squares (ALS)
- **Clustering**: K-means, Gaussian mixtures (GMMs),...
- **Topic modeling**: latent Dirichlet allocation (LDA)
- Frequent itemsets, association rules, and sequential pattern mining

- Feature transformations: standardization, normalization, hashing,...
- ML Pipeline construction
- Model evaluation and hyper-parameter tuning
- ML persistence: saving and loading models and Pipelines

- Other utilities include:
  - Distributed linear algebra: SVD, PCA,...
  - Statistics: summary statistics, hypothesis testing,...

Reminder: SQL Queries can run on Spark

Spark Streaming

Runs **on demand** whenever user requests it to run

Runs **continously** after a fixed interval of time

Jupyter Notebooks

- Open source web application for data analysis
- Supports languages like Python, R, Julia, Scala etc
- Interactive output in the web browser
- Good for demo and learning purposes
- Easy to download and share with others
- Integrates well with Spark

Apache Spark: Analysis

Deploying a Spark Cluster @ CSC via RAHTI

- Spark runs on top of CSC's container cloud known as RAHTI
- Webpage: http://rahti.csc.fi
- You need to have access to RAHTI even if you have a CSC account

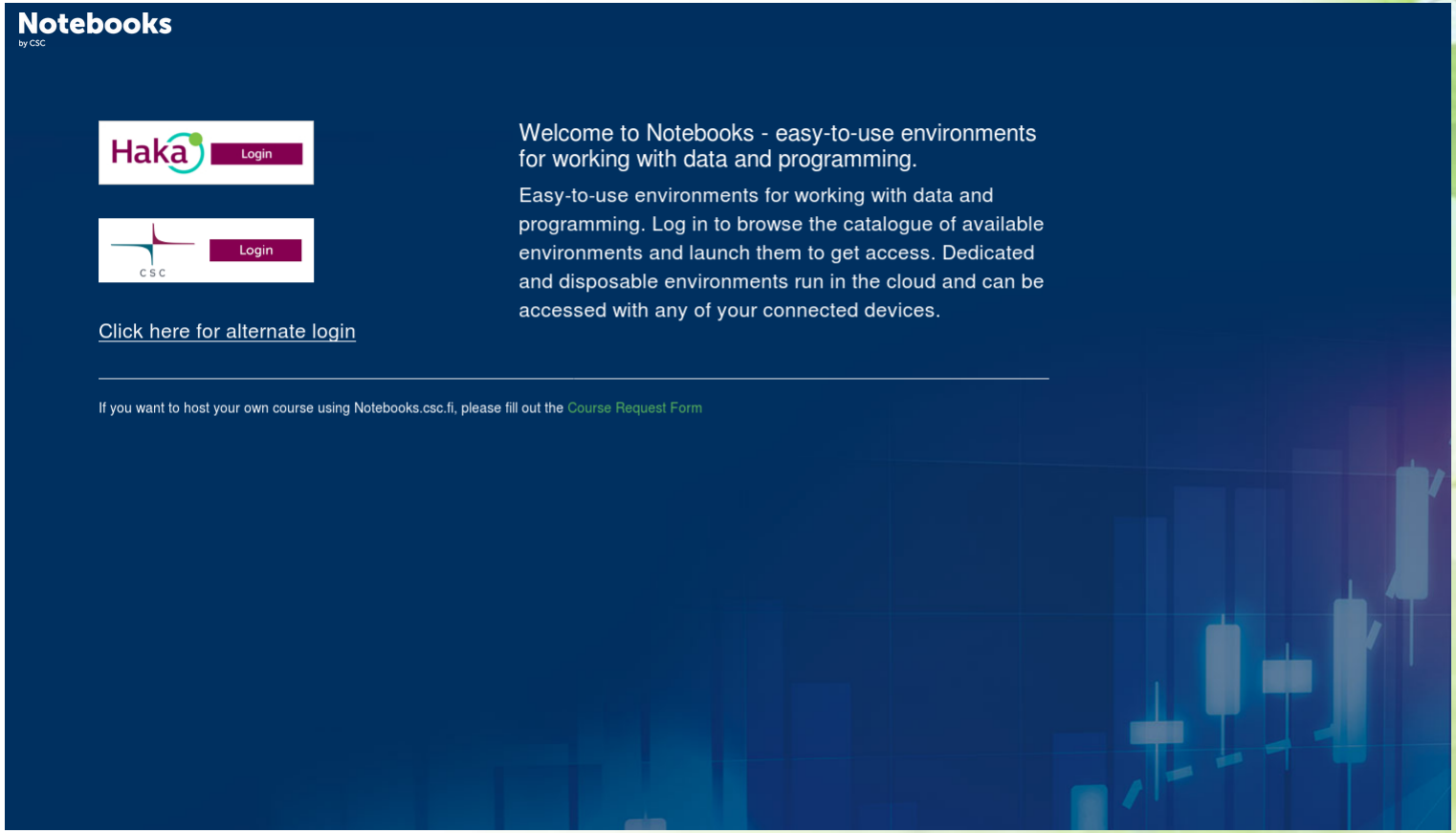
Deploying a Spark Cluster @ CSC via RAHTI

1. Get a CSC account
2. Ask for RAHTI access via rahti-support@csc.fi
3. Select 'Apache Spark' from the catalog and follow the instructions

Deploying a Spark Cluster @ CSC

- Notebooks.csc.fi : Easy to use environments for working with data and programming
- Used for training, learning, demos
- Login with Haka or CSC acc

- From early next month, it would be **possible to launch a small Spark Cluster via CSC Notebooks** for learning or testing purposes



Thank you!

- Spark Cluster Requests: servicedesk@csc.fi
- Databricks Spark Reference
- PySpark Dataframe Reference (for functions)
- Learning Spark: Lightning-Fast Big Data Analysis – Book (O'REILLY)


https://www.facebook.com/CSCfi


https://twitter.com/CSCfi

https://www.youtube.com/c/CSCfi


https://www.linkedin.com/company/csc---it-center-for-science

Spark vs Hadoop

- Faster than Hadoop in many cases specially iterative algorithms
- Spark Stores the data in memory which allows faster chaining of operations vs Hadoop Map Reduce which stores the output to disk for any map or reduce operation
- Much more simpler programming API as compared to Hadoop – Less time writing complex map reduce scripts
- Equally good fault tolerance