Leveraging Mesos as the Ultimate Distributed Data Science Platform

(such a long title,) by @DataFellas
@Noootsab, 8th Oct. ‘15 @MesosCon

However, “Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb” is a rather long title, yet the best movie ever (IMHO)
OUTLINE

- (Legacy) Data Science Pipeline/Product
- What changed since then
- Distributed Data Science (today)
- Luckily, we have Mesos and friends
- Going beyond (productivity)
DATA FELLAS
5 MONTHS OLD BELGIAN STARTUP

Andy Petrella
Maths
scala
Apache Spark
Spark Notebook
Data Banana

Xavier Tordoir
Physics
Bioinformatics
Scala
Spark
(Legacy) Data Science Pipeline
Or, so called, Data Product

Sampling  Modelling  Tuning  Report  Interpret

Static Results
Lot of information lost in translation
Sounds like Waterfall
ETL look and feel
(Legacy) Data Science Pipeline
Or, so called, Data Product

Sampling → Modelling → Tuning → Report → Interpretation

Mono machine!

CPU bounds

Memory bounds
Our World Today

No, it wasn’t better before

Facts

Data gets bigger or, precisely, the amount of available source explodes

Data gets faster (and faster), only even consider: watching Netflix over 4G ôô
Our world today
No, it wasn’t better before

Consequences

Sampling

Hard (or will be too big...)

Report

Ephemeral

Restricted view
Our world today
No, it wasn’t better before

Consequences

Interpretation
⇒ Too slow to get real ROI out of the overall system

How to work that around?
Our world today

No, it wasn’t better before

Needs

Alerting system over descriptive charts

More accurate results

More or harder models (e.g. Deep Learning)

More data

Constant data flow

Online interactions under control (e.g. direct feedback)
Our world today
No, it wasn’t better before

Needs

Distributed Systems
DISTRIBUTED DATA SCIENCE
System/Platform/SDK/Pipeline/Product/... whatever you call it

“CREATE” CLUSTER

FIND AVAILABLE SOURCES (CONTEXT, CONTENT, QUALITY, SEMANTIC, ...)

CONNECT TO SOURCES (STRUCTURE, SCHEMA/TYPES, ...)

CREATE DISTRIBUTED DATA PIPELINE/MODEL

TUNE ACCURACY

TUNE PERFORMANCES

WRITE RESULTS TO SINKS

ACCESS LAYER

USER ACCESS
DISTRIBUTED DATA SCIENCE
System/Platform/SDK/Pipeline/Product/... whatever you call it

“CREATE” CLUSTER

FIND AVAILABLE SOURCES (CONTEXT, CONTENT, QUALITY, SEMANTIC, ...)

CONNECT TO SOURCES (STRUCTURE, SCHEMA/TYPES, ...)

CREATE DISTRIBUTED DATA PIPELINE/MODEL

TUNE ACCURACY

TUNE PERFORMANCES

WRITE RESULTS TO SINKS

ACCESS LAYER

USER ACCESS
DISTRIBUTED DATA SCIENCE
System/Platform/SDK/Pipeline/Product/... whatever you call it

“CREATE” CLUSTER

FIND AVAILABLE SOURCES (context, content, quality, semantic, ...)

CONNECT TO SOURCES (structure, schema/types, ...)

CREATE DISTRIBUTED DATA PIPELINE/MODEL

TUNE ACCURACY

TUNE PERFORMANCES

WRITE RESULTS TO SINKS

ACCESS LAYER

USER ACCESS
DISTRIBUTED DATA SCIENCE
System/Platform/SDK/Pipeline/Product/... whatever you call it

“CREATE” CLUSTER

FIND AVAILABLE SOURCES (CONTEXT, CONTENT, QUALITY, SEMANTIC, ...)

CONNECT TO SOURCES (STRUCTURE, SCHEMA/TYPES, ...)

CREATE DISTRIBUTED DATA PIPELINE/MODEL

TUNE ACCURACY

TUNE PERFORMANCES

WRITE RESULTS TO SINKS

ACCESS LAYER

USER ACCESS
DISTRIBUTED DATA SCIENCE
System/Platform/SDK/Pipeline/Product/… whatever you call it

“CREATE” CLUSTER

FIND AVAILABLE SOURCES (context, content, quality, semantic, …)

CONNECT TO SOURCES (structure, schema/types, …)

CREATE DISTRIBUTED DATA PIPELINE/MODEL

TUNE ACCURACY

TUNE PERFORMANCES

WRITE RESULTS TO SINKS

ACCESS LAYER

USER ACCESS
DISTRIBUTED DATA SCIENCE
System/Platform/SDK/Pipeline/Product/… whatever you call it

“CREATE” CLUSTER

FIND AVAILABLE SOURCES (context, content, quality, semantic, …)

CONNECT TO SOURCES (structure, schema/types, …)

CREATE DISTRIBUTED DATA

TUNE ACCURACY

TUNE PERFORMANCES

WRITE RESULTS TO SINKS

ACCESS LAYER

USER ACCESS

YO!
AREN’T WE TALKING ABOUT “BIG” DATA?
FAST DATA?

SO COULD REALLY (ALL) RESULTS BEING NEITHER BIG NOR FAST?

ACTUALLY, RESULTS ARE BECOMING THEMSELVES “BIG” DATA!
FAST DATA!
DISTRIBUTED DATA SCIENCE
System/Platform/SDK/Pipeline/Product/… whatever you call it

“CREATE” CLUSTER

FIND AVAILABLE SOURCES (context, content, quality, semantic, …)

CONNECT TO SOURCES (structure, schema/types, …)

CREATE DISTRIBUTED DATA PIPELINE/Model

TUNE ACCURACY

TUNE PERFORMANCES

WRITE RESULTS TO SINKS

ACCESS Layer

USER Access

HOW DO WE ACCESS DATA SINCE 90’S? REMEMBER SOA?

→ SERVICES!

NOWADAYS, WE’RE TALKING ABOUT MICRO SERVICES.

HERE WE ARE, ONE SERVICE FOR ONE RESULT.
DISTRIBUTED DATA SCIENCE
System/Platform/SDK/Pipeline/Product/… whatever you call it

“CREATE” CLUSTER

FIND AVAILABLE SOURCES (CONTEXT, CONTENT, QUALITY, SEMANTIC, …)

CONNECT TO SOURCES (STRUCTURE, SCHEMA/TYPES, …)

CREATE DISTRIBUTED DATA PIPELINE/Model

TUNE ACCURACY

TUNE PERFORMANCES

WRITE RESULTS

ACCESS LAYER

USER ACCESS

C’MON, CHARTS/TABLES CANNOT ONLY BE THE ONLY VIEWS OFFERED TO CUSTOMERS/CLIENTS RIGHT?

WE NEED TO OPEN THE CAPABILITIES TO UI (DASHBOARD), CONNECTORS (THIRD PARTIES), OTHER SERVICES (“SOA”) …

OTHER PIPELINES !!!
**WHERE IS MESOS?**

(Almost) EVERYWHERE!

**"CREATE" CLUSTER**

Find available sources
- (context, content, quality, semantic, ...)

**CONNECT TO SOURCES**
- (structure, schema/types, ...)

**CREATE DISTRIBUTED DATA PIPELINE/MODEL**

**TUNE ACCURACY**

Implies Scalability

**TUNE PERFORMANCES**

Implies Scalability

**WRITE RESULTS TO SINKS**

Implies Deployment

**ACCESS LAYER**

Implies Deployment

**USER ACCESS**

Implies Scalability
Why Mesos?
Because it can... (and even more)

Mesos
Allocate
Deploy
Access
Configure
Scale
Schedule

Marathon

Chronos

DCOS
What about Productivity?
Streamlining development lifecycle most welcome

“Create” Cluster

Find Available Sources (context, content, quality, semantic, ...)

Connect to Sources (structure, schema/types, ...)

Create Distributed Data Pipeline/Model

Tune Accuracy

Tune Performances

Write Results to Sinks

Access Layer

User Access
What about Productivity?
Streamlining development lifecycle most welcome

➔ Longer production line
➔ More constraints (resources sharing, time, ...)
➔ More people
➔ More skills

Overlooking these points and you’ll be kicked soon or sooner

So, how to have:

● Results coming fast enough whilst keeping accuracy level high?
● Responsivity to external/unpredictable events?
What about Productivity?
Streamlining development lifecycle most welcome

At Data Fellas, we think that we need **Interactivity and Reactivity to tighten the frontiers (within team and in time).**

Hence, Data Fellas

- Extends the **Spark Notebook** (interactivity)
- In the **Shar3** product (integrated Reactivity)
That’s all folks
Thanks for listening/staying

Poke us on

@DATAFELLAS

@SHAR3_FELLAS

@SPARKNOTEBOOK

@XTORDOIR & @NOOOTSAB

Now @TYPESAFE:  http://t.co/01Bt6dQtGh

Follow up soon on  http://NOETL.ORG

(Hi5 to @CHIEFSCIENTIST for that)