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
- Trainer
- Data Banana

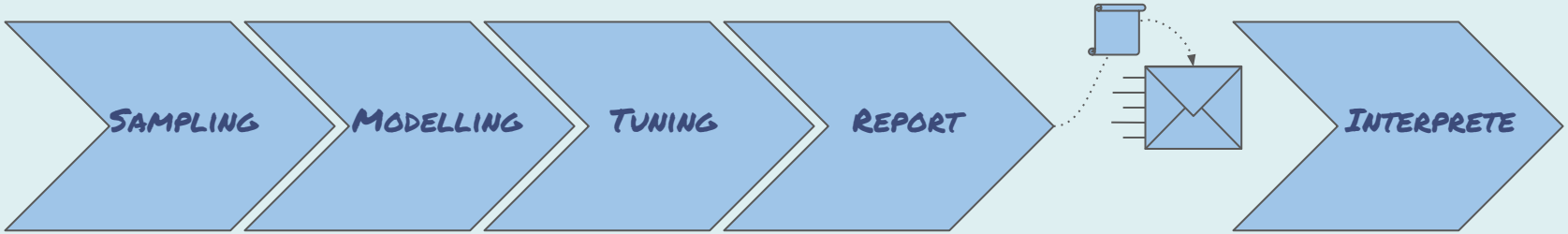
Xavier Tordoir

- Physics
- Bioinformatics

- Scala
- Spark

(Legacy) Data Science Pipeline

Or, so called, Data Product



Static Results

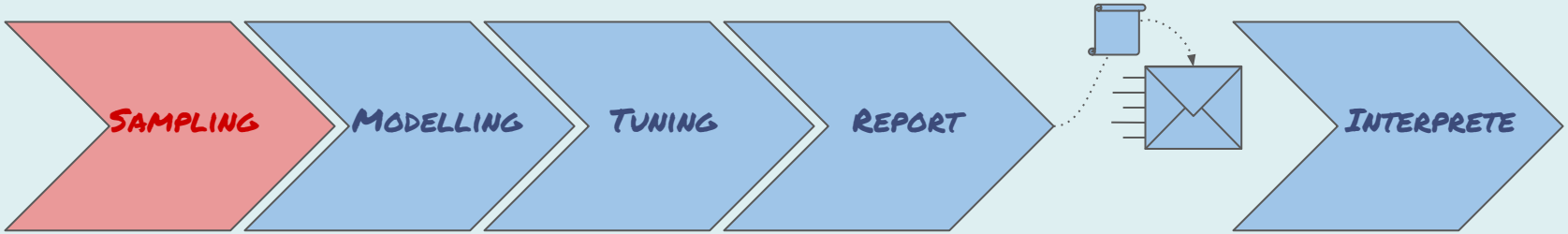
Lot of information lost in translation

Sounds like Waterfall

ETL look and feel

(Legacy) Data Science Pipeline

Or, so called, Data Product



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 ôÖ

Our world Today

No, it wasn't better before

Consequences

~~Sampling~~ → HARD (or will be too big...)

~~Sampling~~ → Ephemeral

~~Report~~ → Ephemeral

~~Report~~ → 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

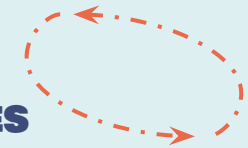
Find available sources (context, content, quality, semantic, ...)

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 (connect, security, 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

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 to sinks

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 — Implies Allocation

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

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

Create distributed data pipeline/Model

Tune accuracy

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)

Marathon

↓

Allocate

Deploy

Access

Configure

Scale

Schedule ← Chronos

Mesos

DCOS

What about Productivity?

Streamlining development lifecycle most welcome

"Create" Cluster	OPS
Find available sources (context, content, quality, semantic, ...)	DATA
Connect to sources (structure, schema/types, ...)	OPS DATA
Create distributed data pipeline/Model	SCI
Tune accuracy	SCI
Tune performances	SCI OPS
Write results to Sinks	OPS DATA
Access Layer	WEB OPS DATA
User Access	WEB OPS DATA SCI

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 

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)