

From DataFrames to Tungsten: A Peek into Spark's Future

Reynold Xin @rxin
Spark Summit, San Francisco
June 16th, 2015



DataFrame

noun

Making Spark accessible to everyone (data scientists, engineers, statisticians, ...)

Tungsten

noun

Making Spark faster & prepare for the next five years.

The background of the slide is a teal watercolor wash, with darker, more saturated areas in the upper left and center, fading into lighter shades of teal towards the bottom and right edges. The texture is soft and painterly.

How do DataFrames and
Tungsten relate to each other?

Google Trends for “dataframe”

Single-node tabular data structure, with API for

relational algebra (filter, join, ...)

math and stats

input/output (CSV, JSON, ...)

ad infinitum



Data frame: lingua franca for “small data”

```
head(flights)
#> Source: local data frame [6 x 16]
#>
#>   year month day dep_time dep_delay arr_time arr_delay carrier tailnum
#> 1  2013     1   1      517         2      830         11      UA  N14228
#> 2  2013     1   1      533         4      850         20      UA  N24211
#> 3  2013     1   1      542         2      923         33      AA  N619AA
#> 4  2013     1   1      544        -1     1004        -18      B6  N804JB
#> .. ... ..
```

Spark DataFrame

Distributed data frame for Java, Python, R, Scala

Similar APIs as single-node tools (Pandas, dplyr), i.e. easy to learn

```
> head(filter(df, df$waiting < 50)) # an example in R
## eruptions waiting
##1      1.750      47
##2      1.750      47
##3      1.867      48
```

Spark
DataFrame



Existing
Single-node
Data Frames



KB

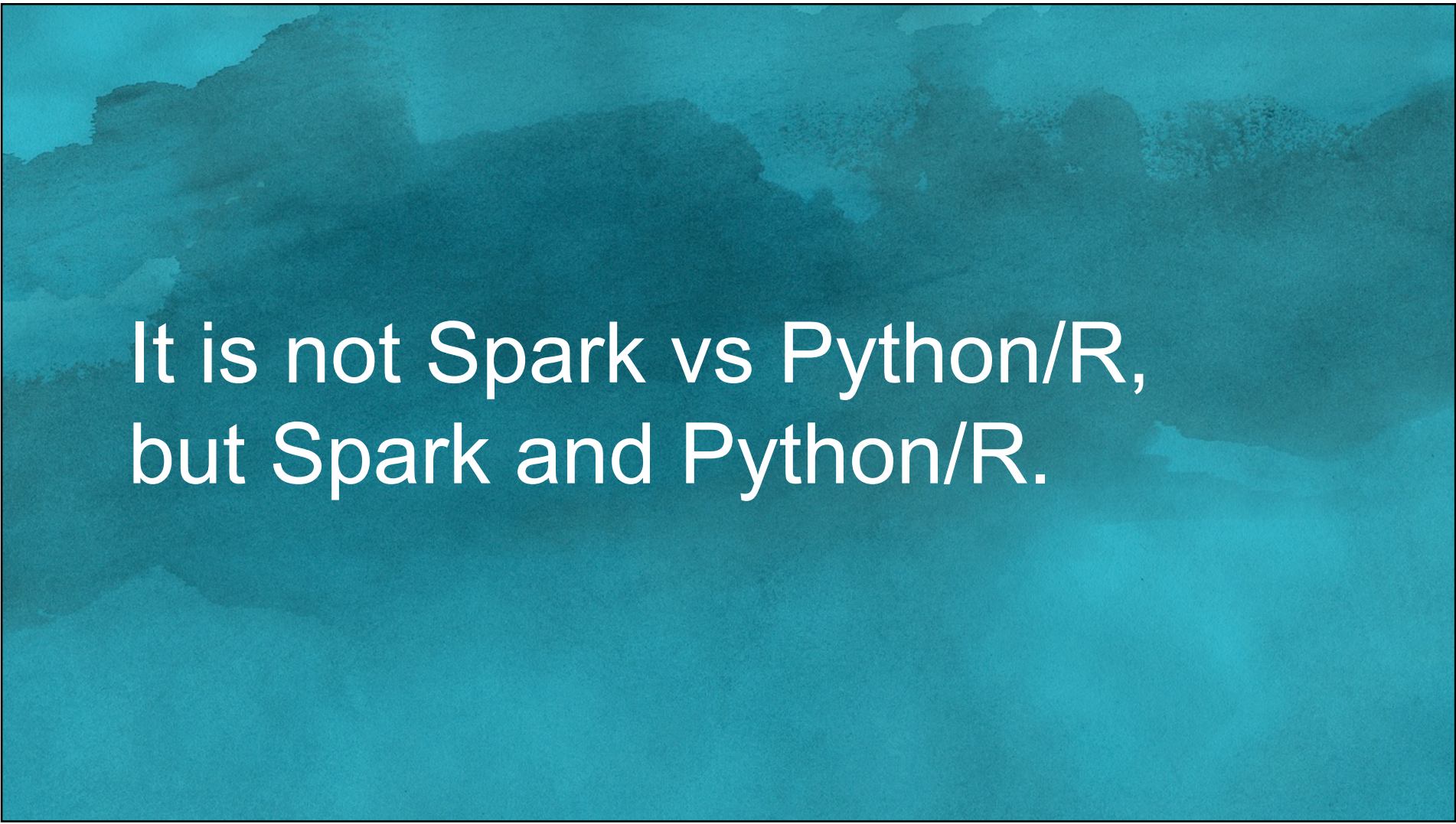
MB

GB

TB

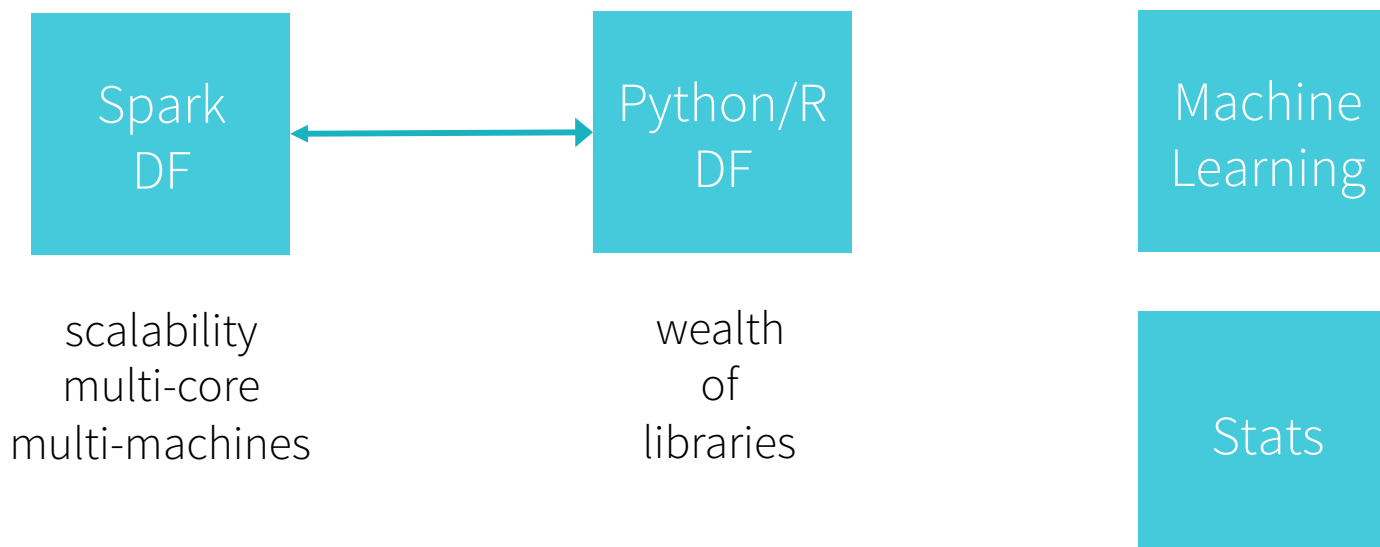
PB

data size

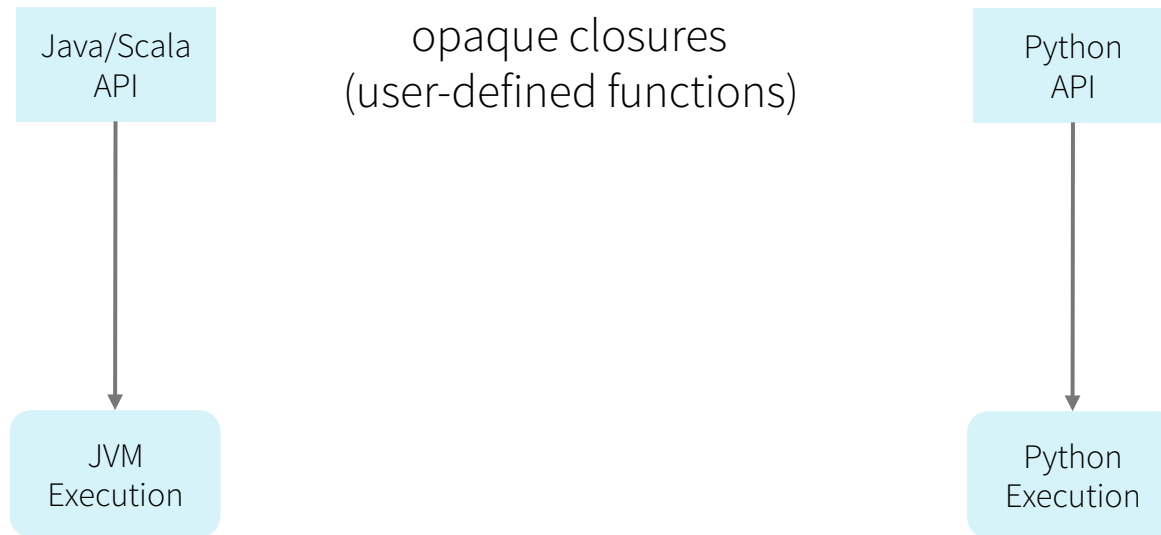
The background of the slide is a teal watercolor wash, with darker, more saturated areas at the top and bottom, and lighter, more diffused areas in the middle where the text is located.

It is not Spark vs Python/R,
but Spark and Python/R.

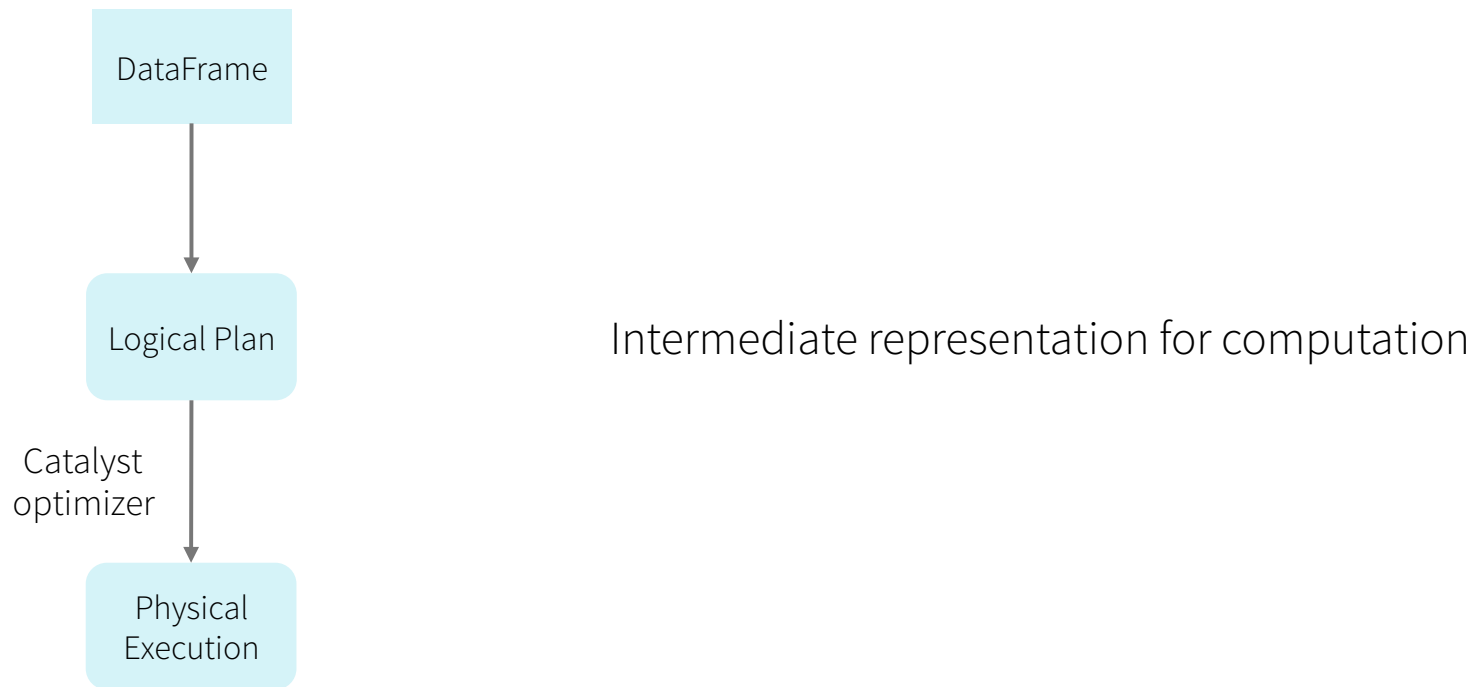
Spark **and** Python/R



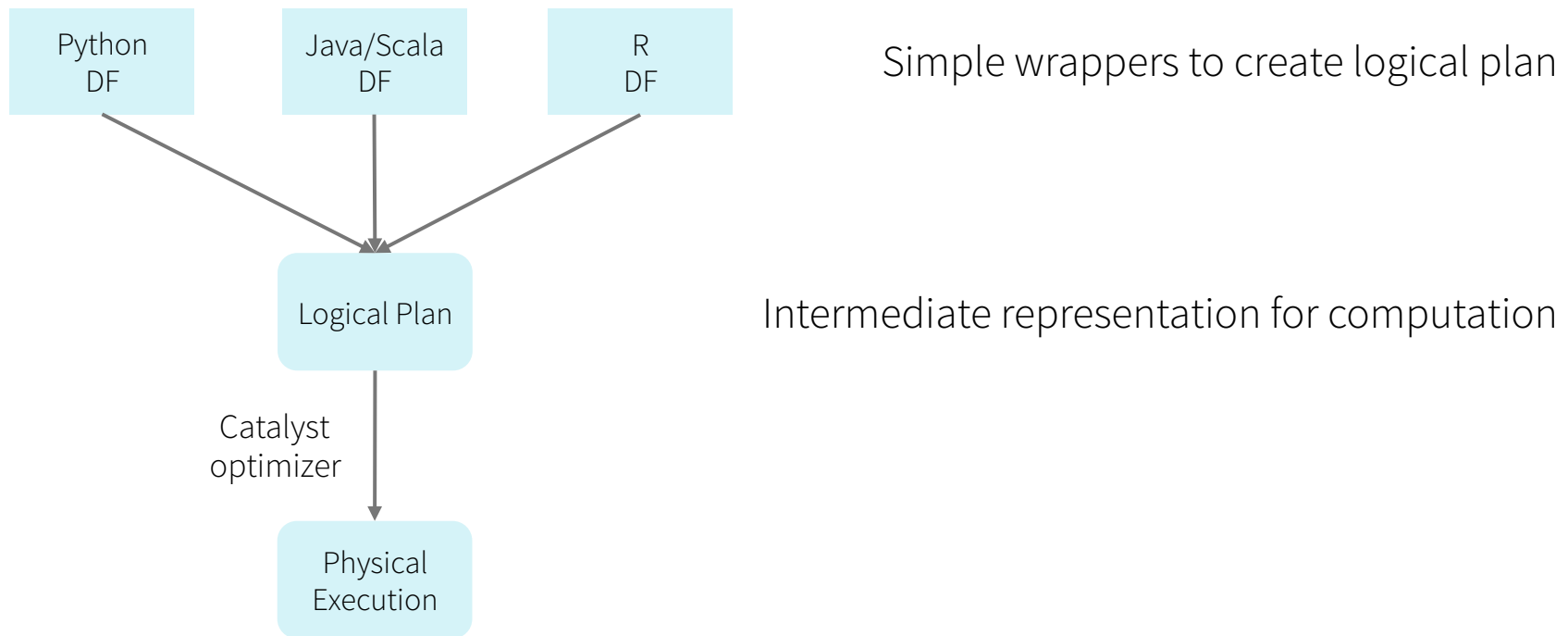
Spark RDD Execution



Spark DataFrame Execution



Spark DataFrame Execution



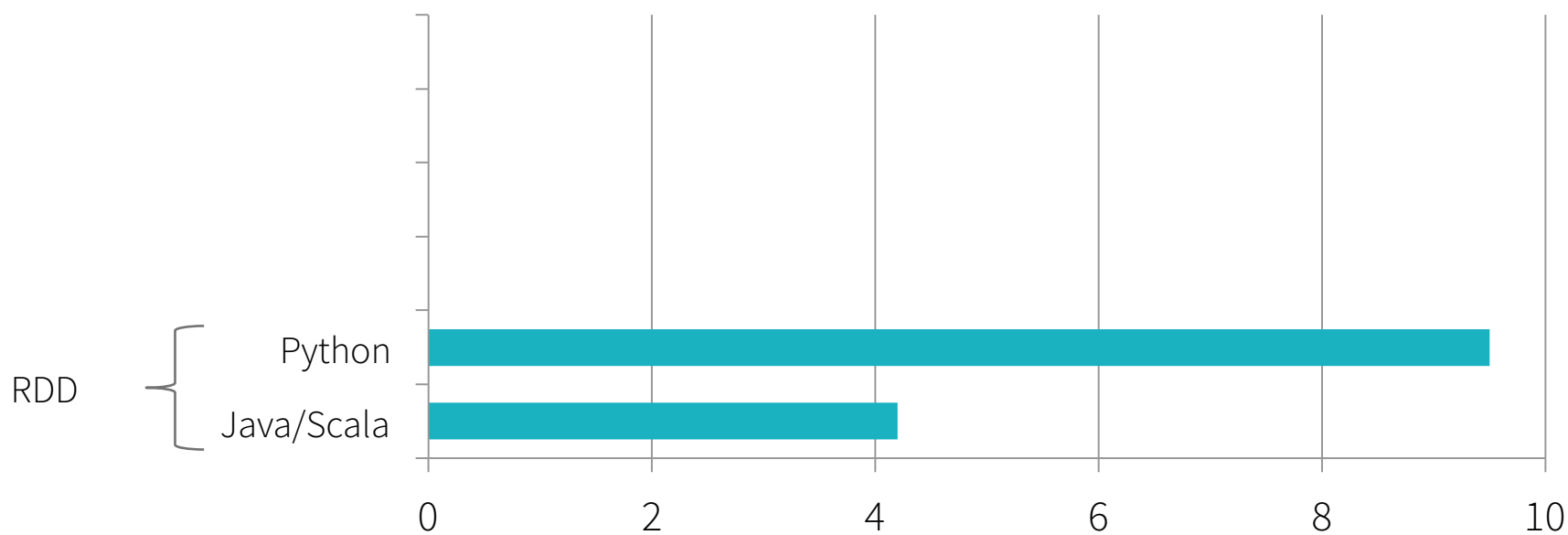
Benefit of Logical Plan: Simpler Frontend

Python : ~2000 line of code (built over a weekend)

R : ~1000 line of code

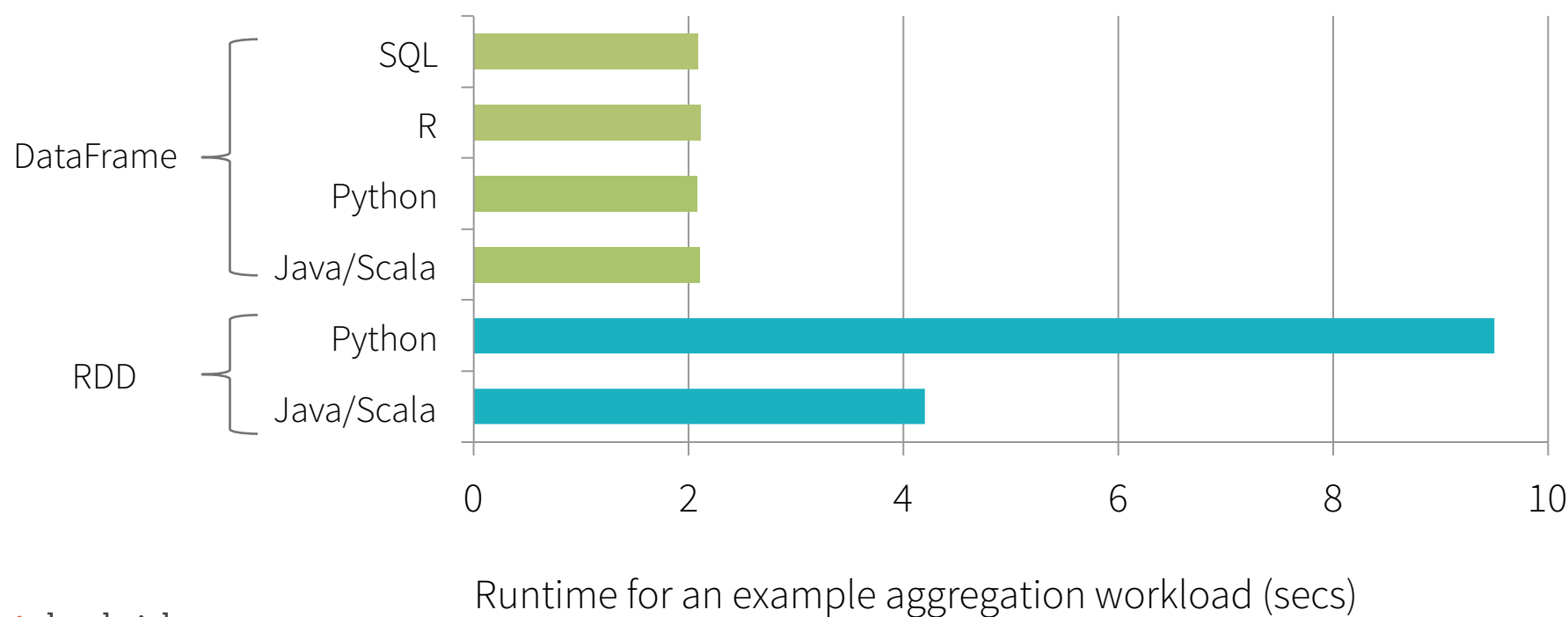
i.e. much easier to add new language bindings (Julia, Clojure, ...)

Performance



Runtime for an example aggregation workload

Benefit of Logical Plan: Performance Parity Across Languages



The background of the slide is a textured teal or turquoise color, created with a watercolor effect. It features soft, blended edges and subtle variations in tone, with some areas appearing slightly darker or more saturated than others, giving it an artistic, painterly feel.

What about Tungsten?

Hardware Trends

Storage

Network

CPU

Hardware Trends

2010

Storage 50+MB/s
(HDD)

Network 1Gbps

CPU ~3GHz

Hardware Trends

	2010	2015
Storage	50+MB/s (HDD)	500+MB/s (SSD)
Network	1Gbps	10Gbps
CPU	~3GHz	~3GHz

Hardware Trends

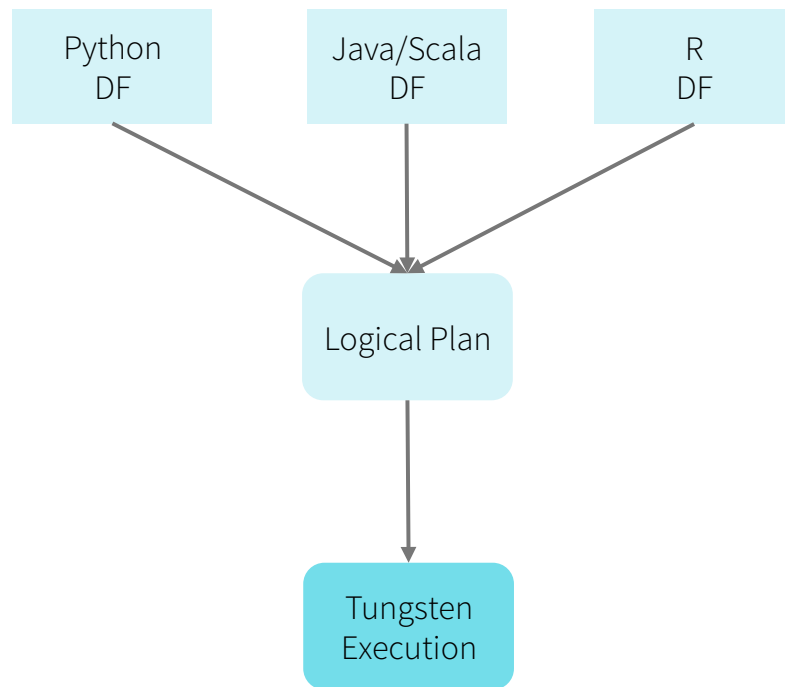
	2010	2015	
Storage	50+MB/s (HDD)	500+MB/s (SSD)	10X
Network	1Gbps	10Gbps	10X
CPU	~3GHz	~3GHz	☹️

Tungsten: Preparing Spark for Next 5 Years

Substantially speed up execution by optimizing CPU efficiency, via:

- (1) Runtime code generation
- (2) Exploiting cache locality
- (3) Off-heap memory management

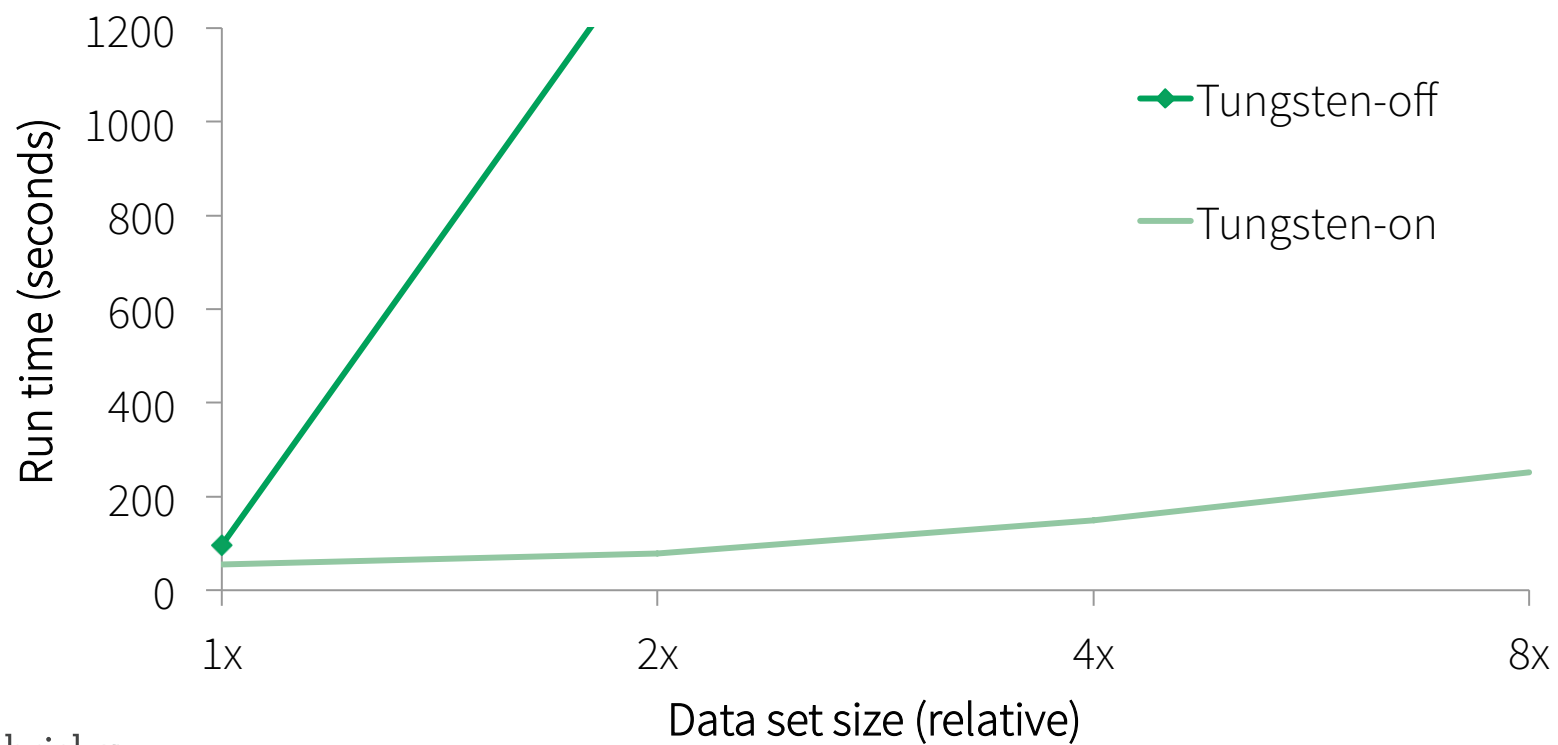
From DataFrame to Tungsten



5PM

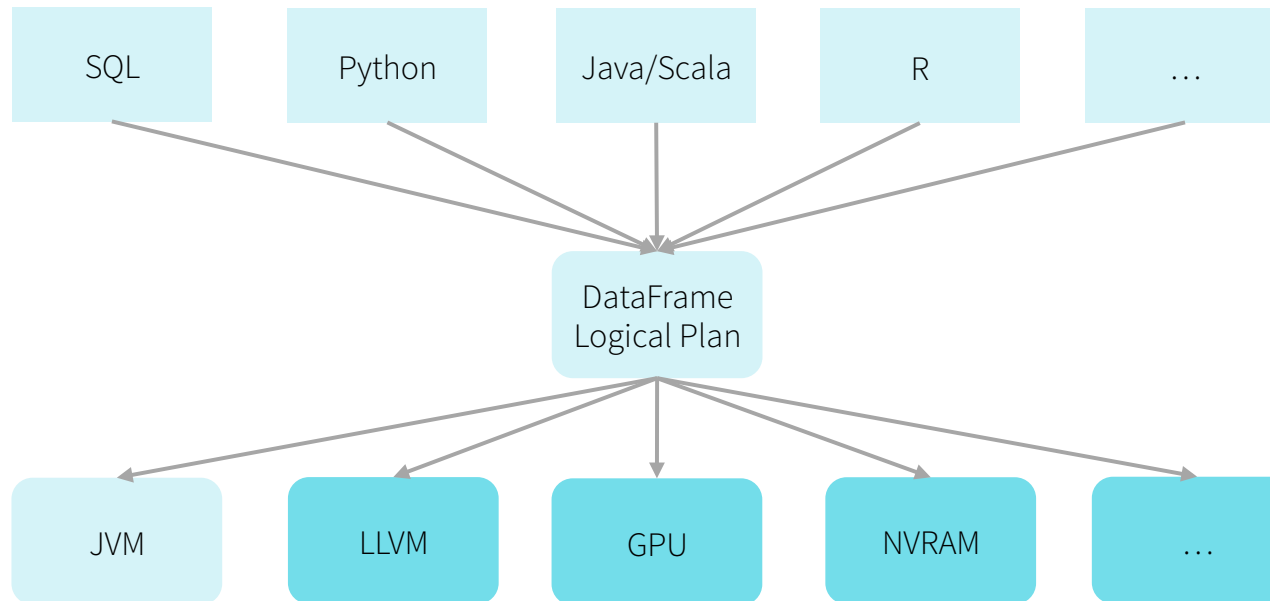
Deep Dive into Project Tungsten
Developer Track by Josh Rosen

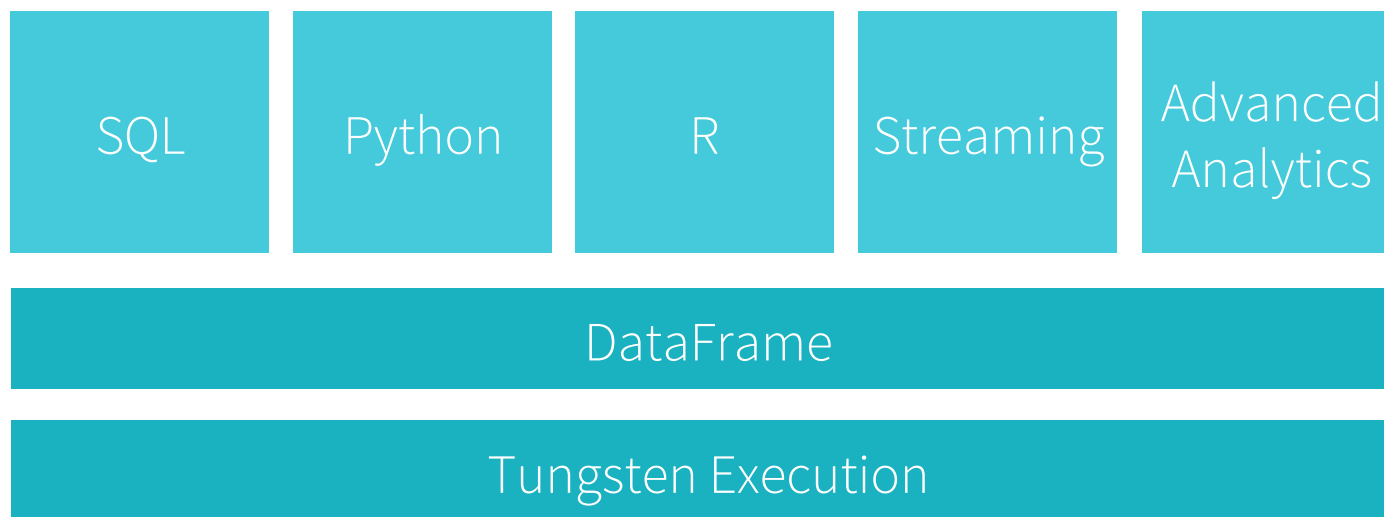
Initial Performance Results



Unified API, One Engine, Automatically Optimized

language
frontend





Spark Office Hours Today

Topic Area	
1:00-1:45	Core, YARN, Ops
1:45-2:30	Core/SQL/Data Science
3:00-3:40	Streaming
3:40-4:15	Core, Python, R
4:30-5:15	Machine Learning
5:15-6:00	Matei Zaharia

Databricks booth A1