

# Data-Intensive Distributed Computing

CS 431/631 (Fall 2021)

## Part 3: From MapReduce to Spark (1/3)

Ali Abedi

These slides are available at <https://www.student.cs.uwaterloo.ca/~cs451/>



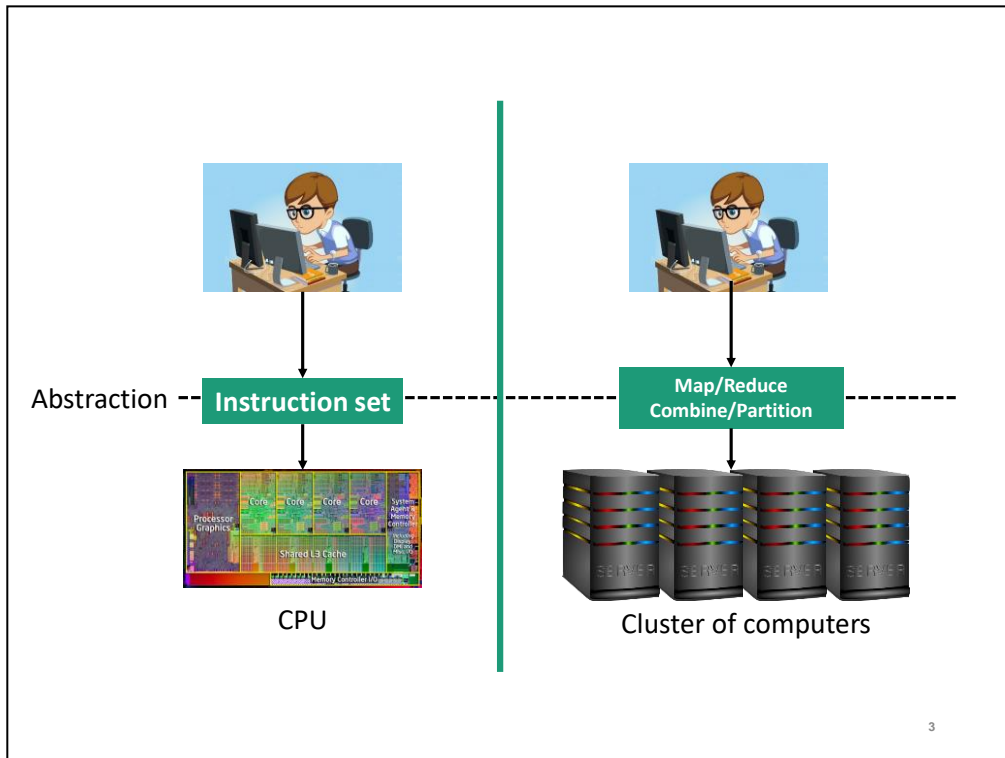
This work is licensed under a Creative Commons Attribution-NonCommercial-Share Alike 3.0 United States  
See <http://creativecommons.org/licenses/by-nc-sa/3.0/us/> for details



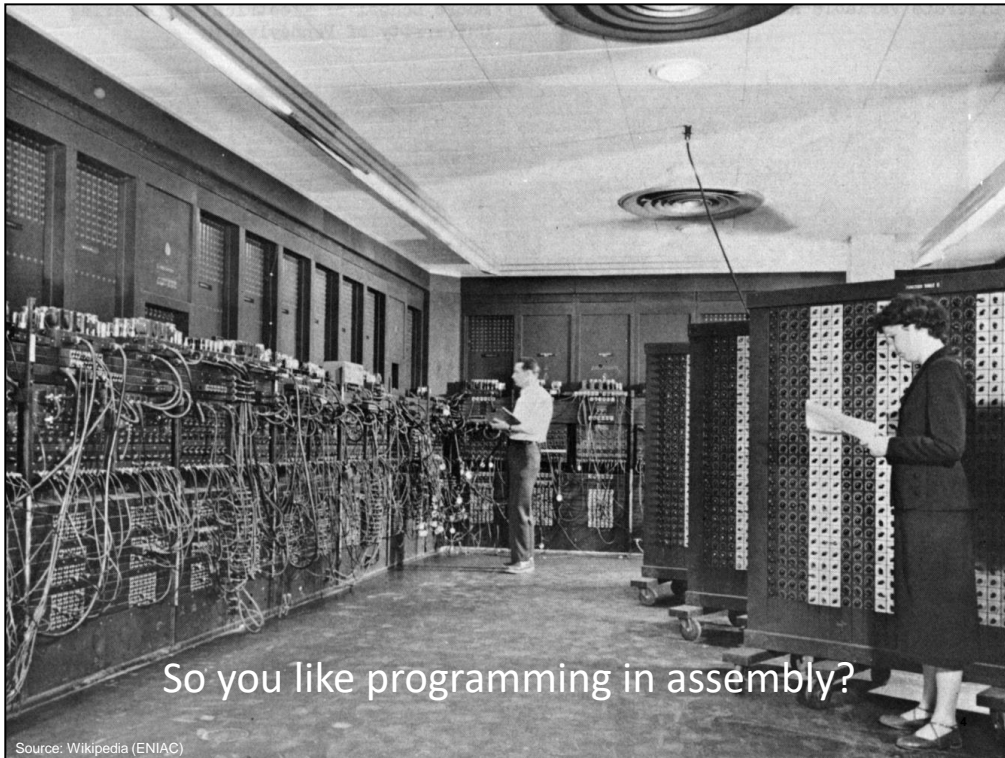
The datacenter *is* the computer!  
What's the instruction set?

Source: Google

2



We need a solution for both storage and computing.



So when we program in MapReduce is it like programming in assembly?! How can we do better?

What's the solution?

Design a higher-level language  
Write a compiler

Hadoop is great, but it's really waaaaay too low level!



What we really need is SQL!

Answer:



What we really need is a scripting language!

Answer:

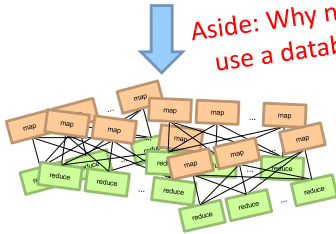


Yahoo and Facebook designed their own solutions on top of Hadoop to make it more flexible for their engineers.

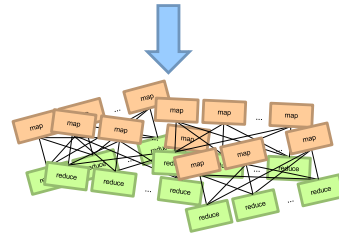


SQL

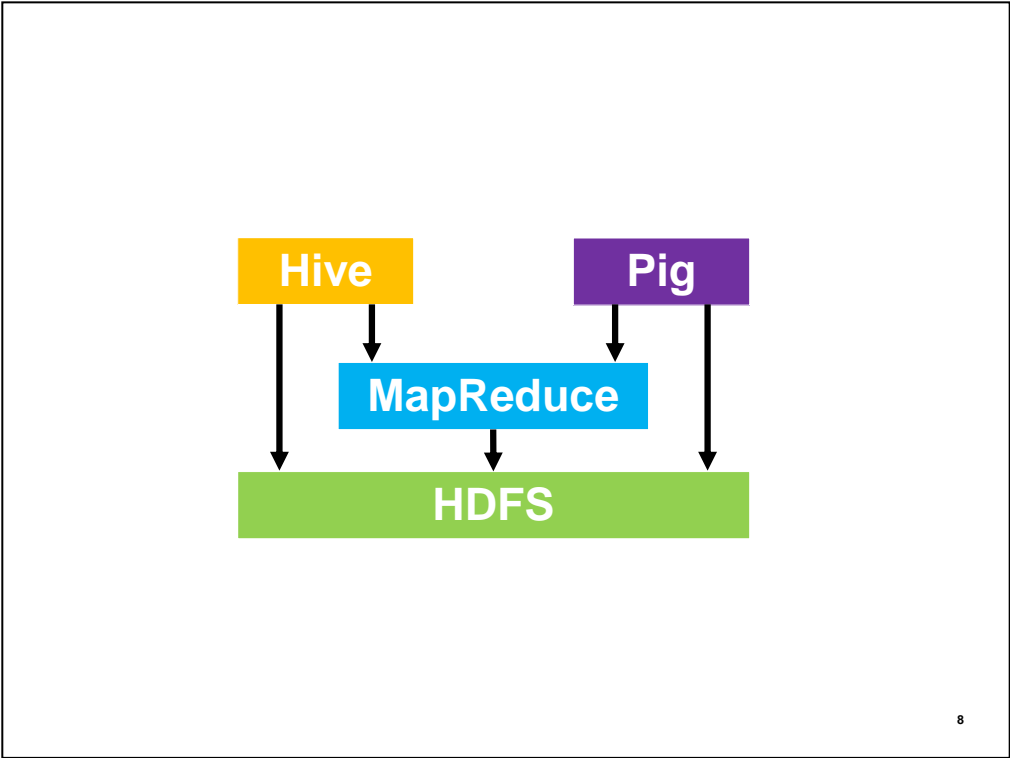
Aside: Why not just use a database?



Pig Scripts



Both open-source projects today!



Pig and Hive programs are converted to MapReduce jobs at the end of the day.





## Pig: Example

Task: Find the top 10 most visited pages in each category

Visits

User	Url	Time
Amy	cnn.com	8:00
Amy	bbc.com	10:00
Amy	flickr.com	10:05
Fred	cnn.com	12:00

•  
•  
•

URL Info

Url	Category	PageRank
cnn.com	News	0.9
bbc.com	News	0.8
flickr.com	Photos	0.7
espn.com	Sports	0.9

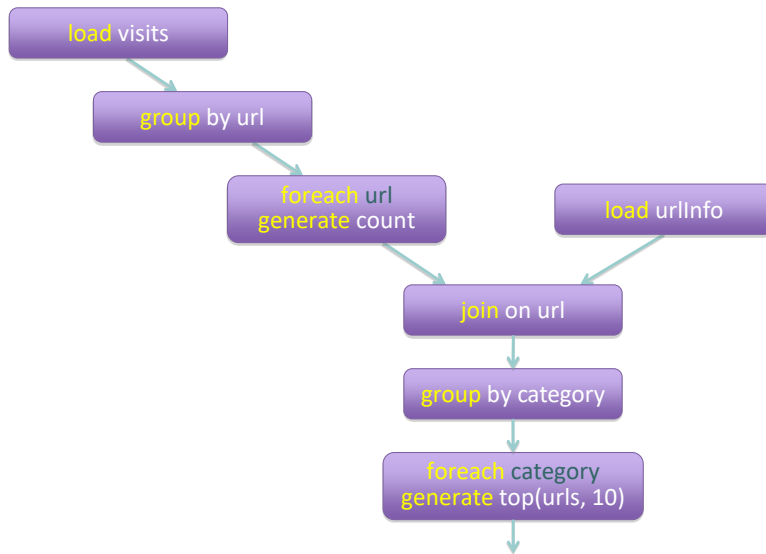
•  
•  
•

## Pig: Example Script

```
visits = load '/data/visits' as (user, url, time);
gVisits = group visits by url;
visitCounts = foreach gVisits generate url, count(visits);
urlInfo = load '/data/urlInfo' as (url, category, pRank);
visitCounts = join visitCounts by url, urlInfo by url;
gCategories = group visitCounts by category;
topUrls = foreach gCategories generate top(visitCounts,10);

store topUrls into '/data/topUrls';
```

# Pig Query Plan



Pig Slides adapted from Olston et al. (SIGMOD 2008)

12

# Fig: MapReduce Execution

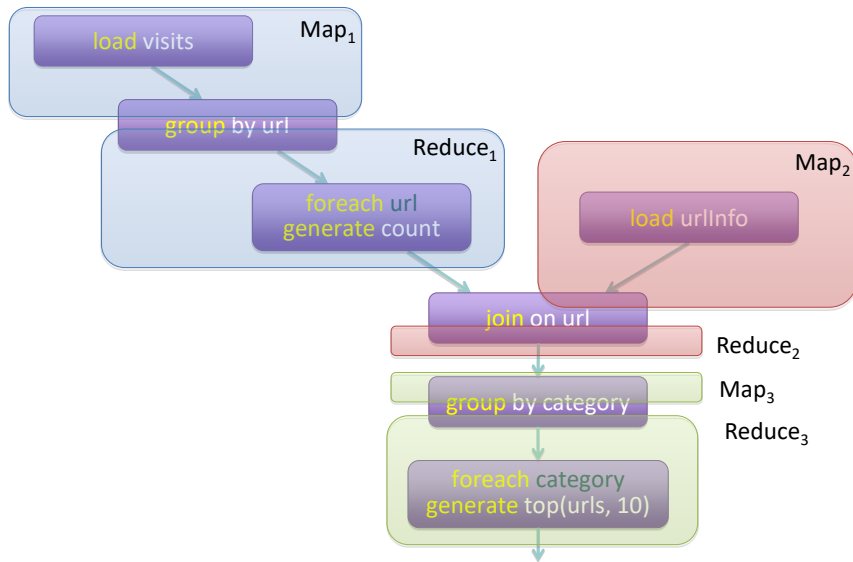


Fig Slides adapted from Olston et al. (SIGMOD 2008)

13

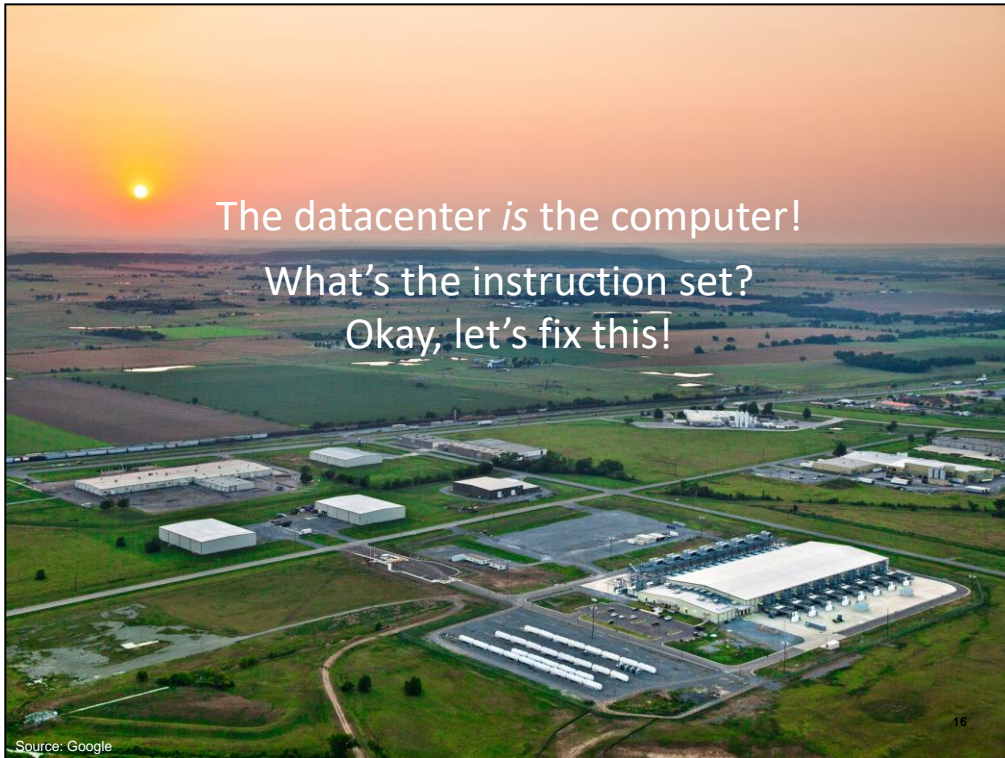


## But isn't Pig slower?

Sure, but c can be slower than assembly too...



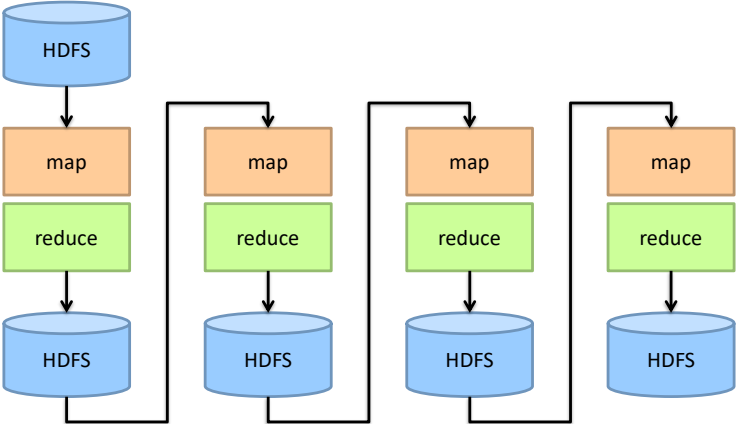
15



Having to formulate the problem in terms of map and reduce only is restrictive.



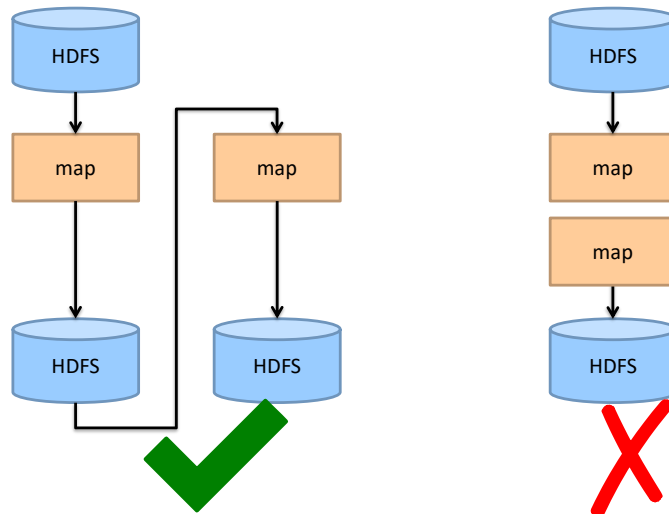
# MapReduce Workflows



What's wrong?

There is a lot of disk i/o involved which significantly reduces running MapReduce jobs like this.

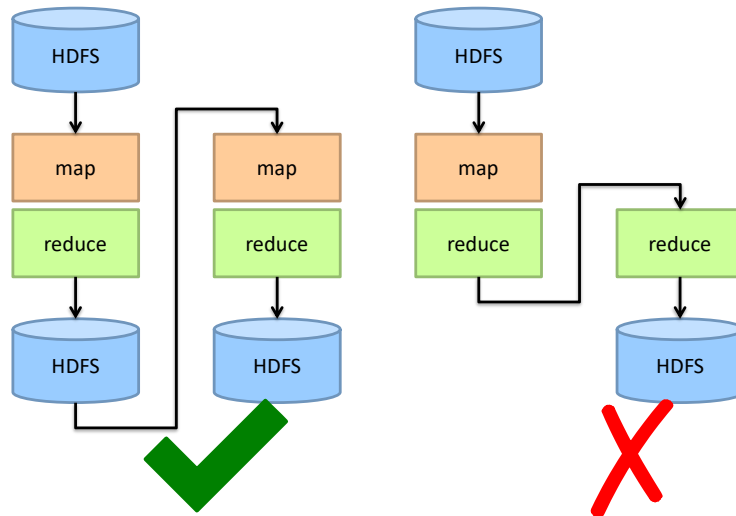
## Want MM?



18

It's okay not to have reduce but the output of map cannot go to another map.

## Want MRR?



19

Similarly we cannot directly move the output of reduce to another reduc.



Can we add more operations to make the instruction set more flexible?

# Spark

Answer to “What’s beyond MapReduce?”



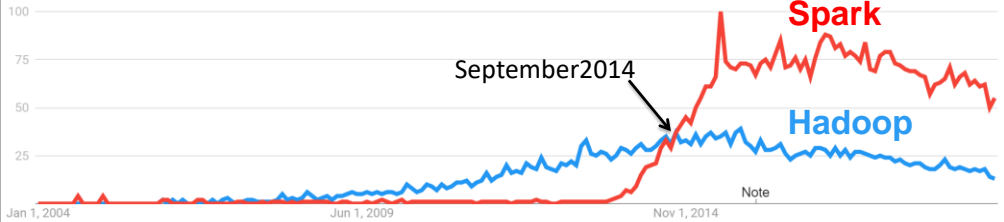
## Brief history:

Developed at UC Berkeley AMPLab in 2009

Open-sourced in 2010

Became top-level Apache project in February 2014

# Spark vs. Hadoop

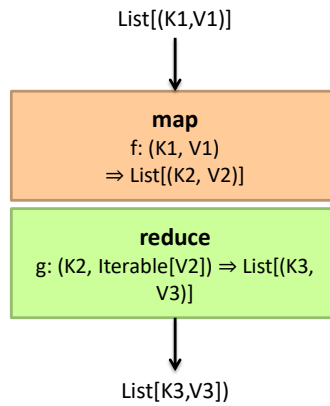


Google Trends

Spark is more popular than Hadoop today.



# MapReduce

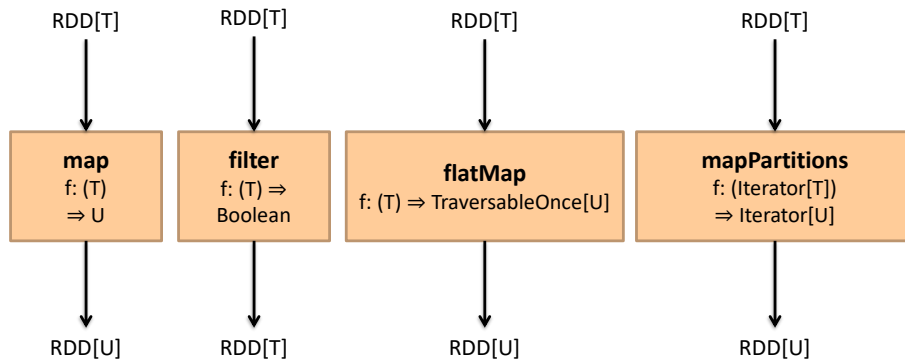


24

This is the only mechanism we had in MapReduce.



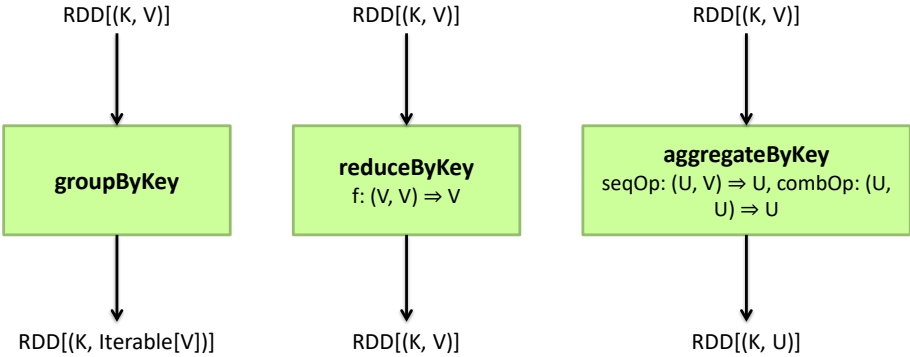
## Map-like Operations



25

But Spark provides many more operations (enriched instruction set).

# Reduce-like Operations



And many other operations!