

# **HiOA Big Data Course**

## **Session 6 - Dealing With Big Data Quantities**

**Dirk Hesse**

# **Big data quantities?**

We're talking several dozen to several 100 TB.

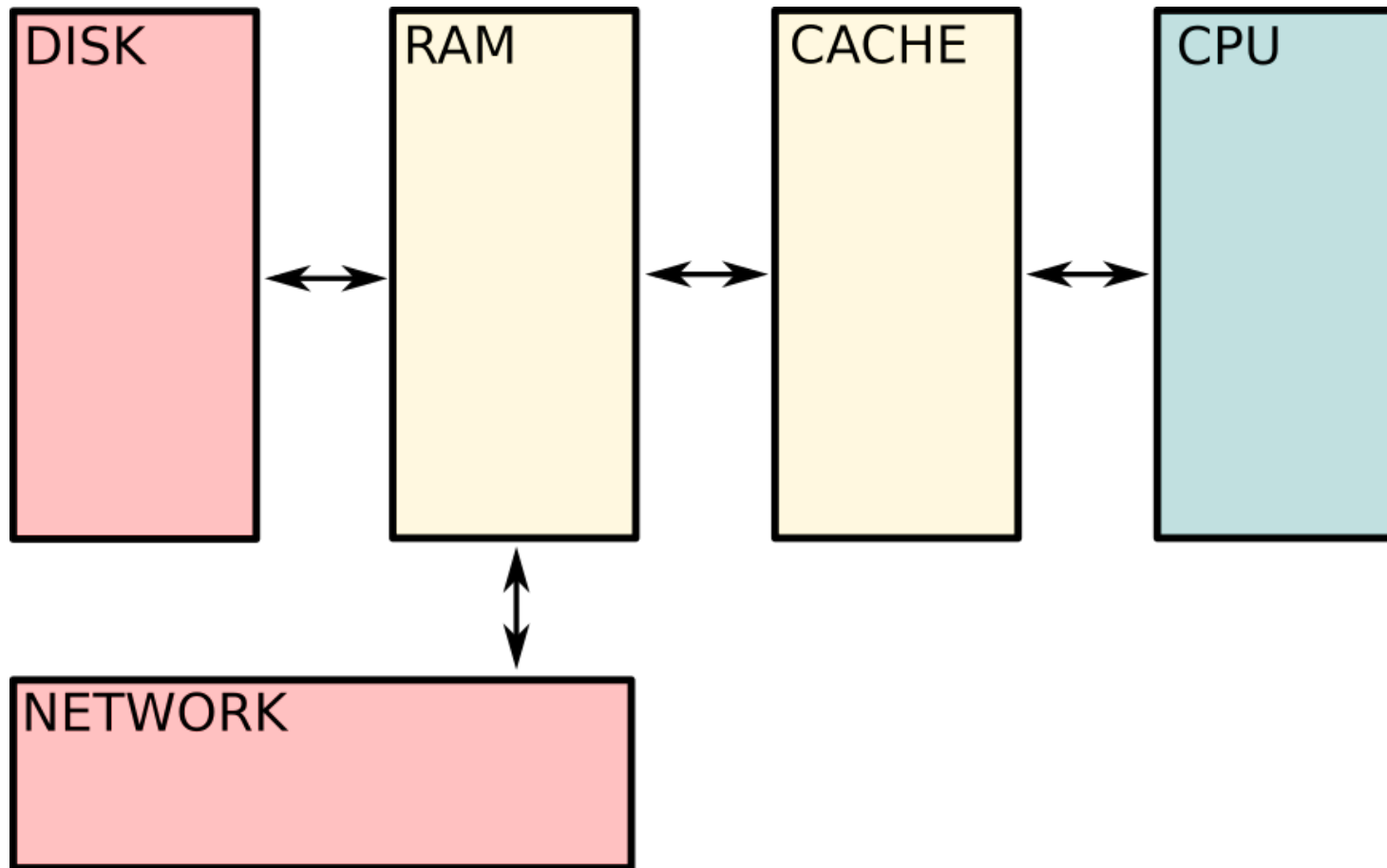
## **Example: AWS instances for rent**

### **X1.32xlarge (Feb. 2017)**

- 128 CPUs
- 1 952 GB of RAM
- 2 x 1920 TB SSD
- approx. 130 NOK/hour

**How does a computer work anyway?**

# Data flow in your computer



# Interesting properties

- **Latency**
  - How fast can data be accessed?
- **Bandwidth**
  - How much data per second can be transferred?
- **Size**
  - How much data can be stored?

These vary by *orders of magnitude*.

# Latencies

Time	Event	Equivalent
1ns	L1 cahce ref.	Human blink (150ms).
3ns	Branch mispr.	Light travels around the world twice.
4ns	L2 cache ref.	Arrow travels 50m (0.6s).
100ns	RAM ref.	Usain Bolt runs 150m (15s).
2 $\mu$ s	Zip 1KB	Make a coffe (5 min).
16 $\mu$ s	SSD read	Watch 2 episodes of Seinfeld (40m).
500 $\mu$ s	Net DC<- >DC	Watch all LOtR movies twice (20h).
3ms	Disk seek.	Watch all James Bond movies twice (5d).
150ms	Net CA<- >NL	Take STK-INF4000 twice (8.5 months).

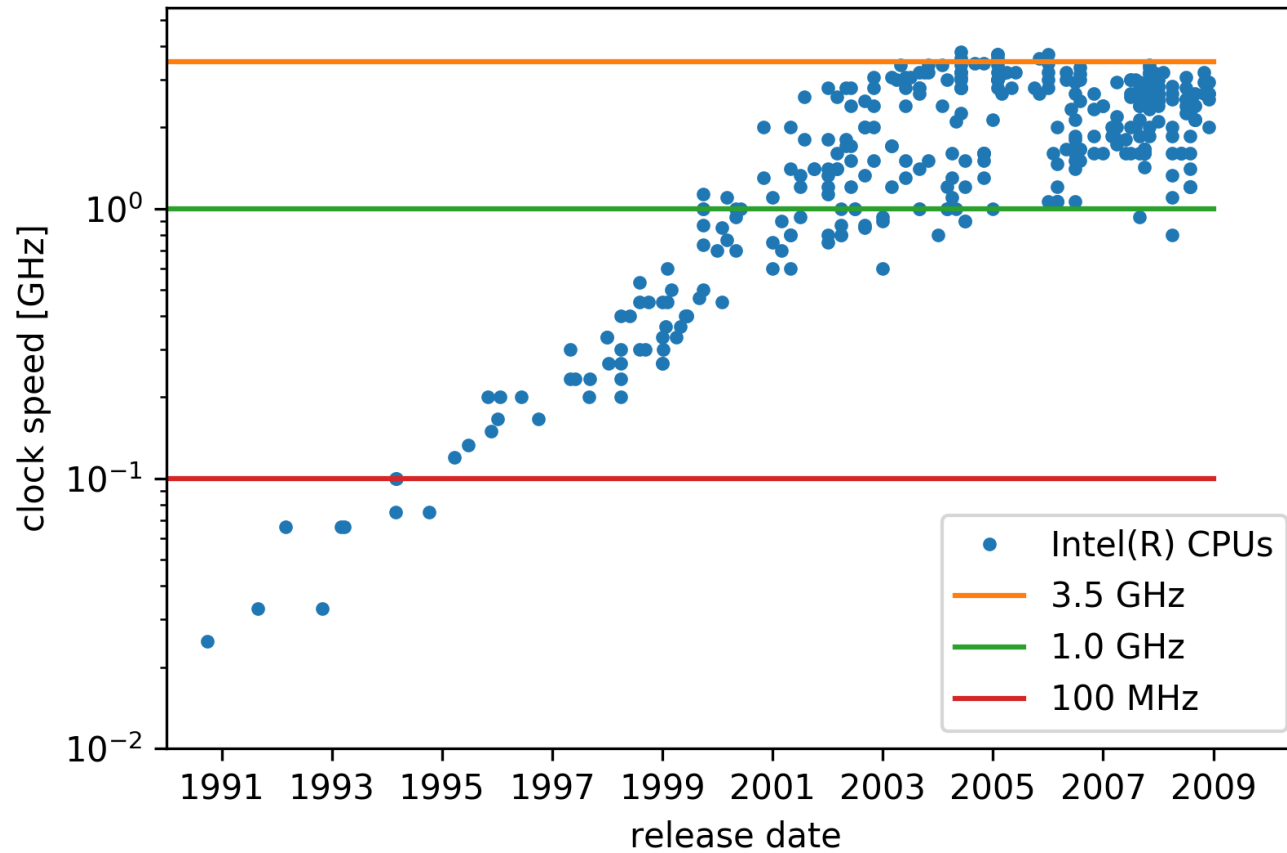


## Sizes and Bandwidths (Haswell Mobile)

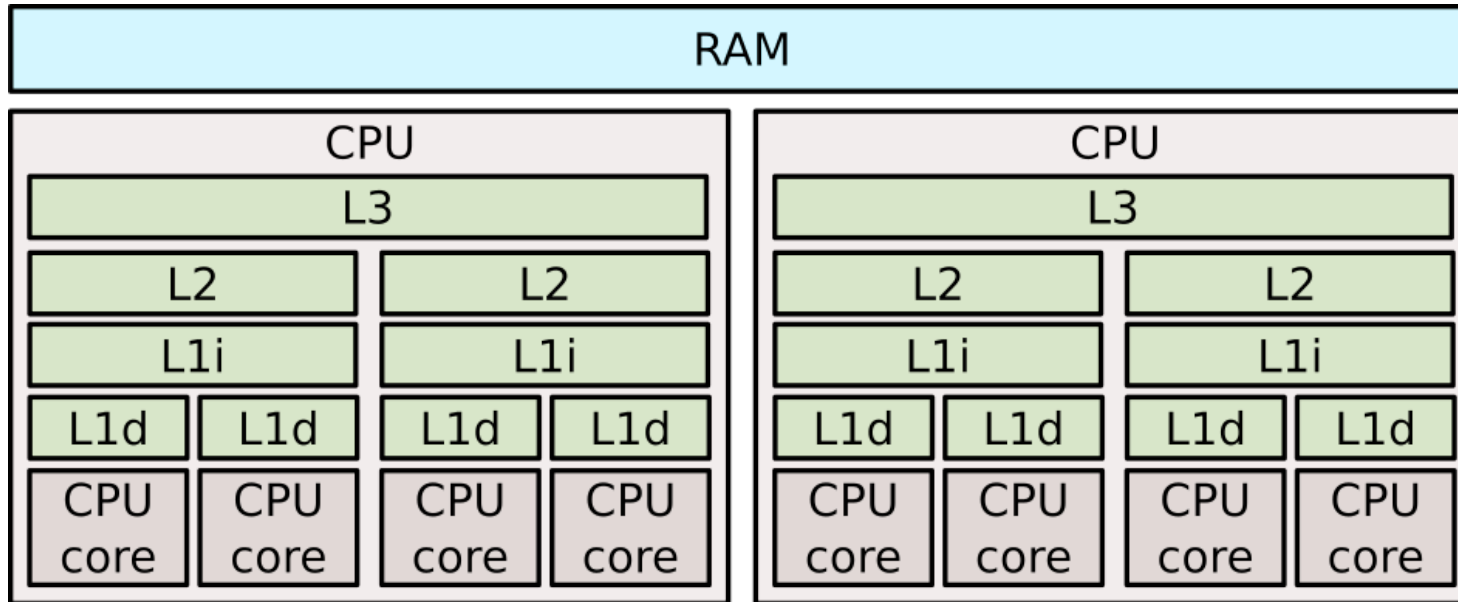
Medium	Size	Bandwidth
Level 1 Cache	128 KiB	700 GiB/s
Level 2 Cache	1 MiB	200 GiB/s
Level 3 Cache	6 MiB	100 GiB/s
RAM	16 GiB	10 GiB/s
SSD	500 GiB	600 MB/s
Ethernet		13 MB/s
Wifi		1.4 MB/s

# **Why not use a single CPU?**

# CPU Clock Speeds



# The Ugly Truth



# Parallelize the work

- Multiprocessing / Threading
  - Operate with multiple CPUs/cores on same memory.
- Cluster computing (i.e. Cray, Blue Gene)
  - Have multiple machines work on the same problem.
  - Communicate via network (slow).
  - Usually used for scientific simulations (fluid dynamic, quantum chromodynamics)
    - Tend to be communication intensive.
- Low-cost commodity hardware clusters.
  - Things break easily.
  - Communication not as efficient.
  - Cheap.

# Multiprocessing

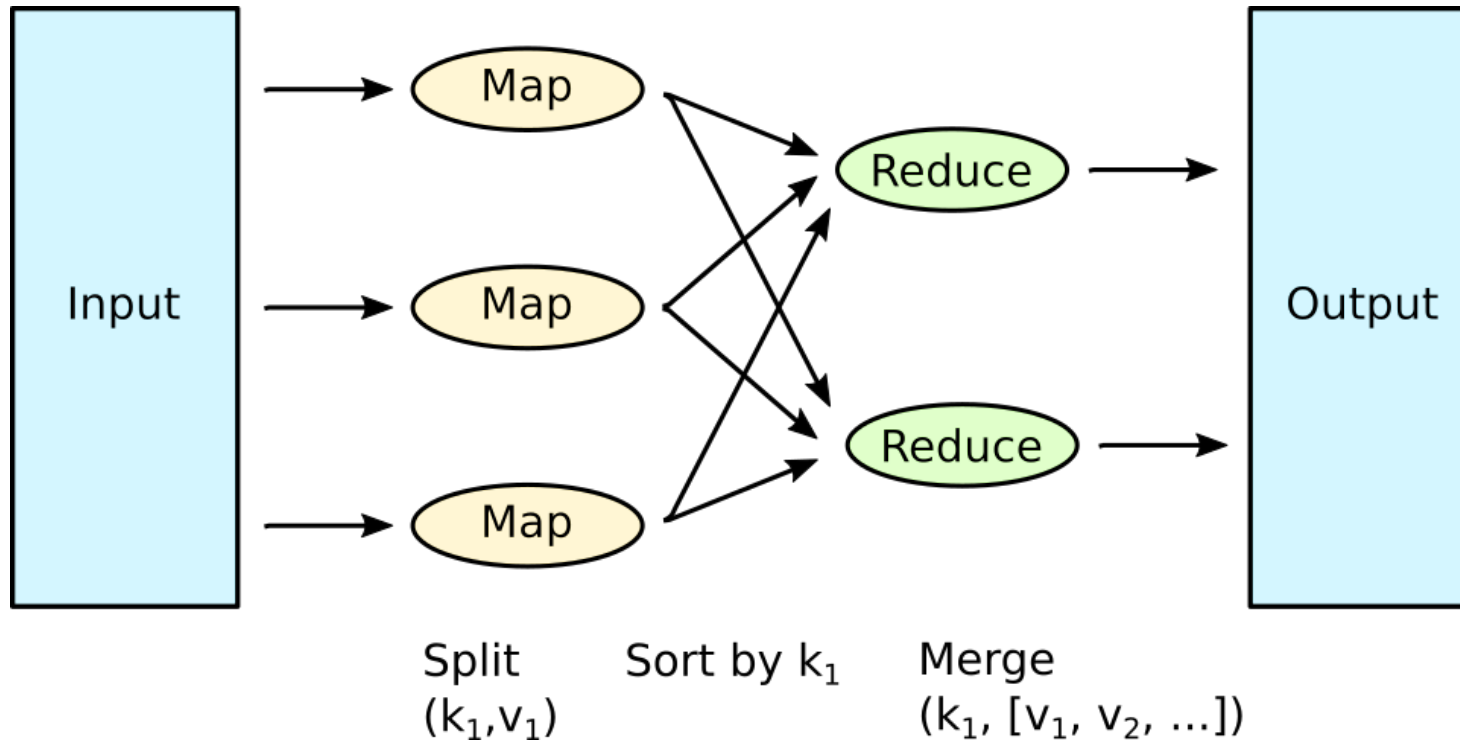
- Needs synchronization between threads.
- Needs coordination of memory accesses.
- Done automatically in some numpy backends.
  - And hence in `scikit-learn`.
- Manually: `multiprocessing`.
  - Will keep track of synchronization/coordination for you.

# Cluster computing

- In the data science world: MapReduce.
- In other fields: MPI.

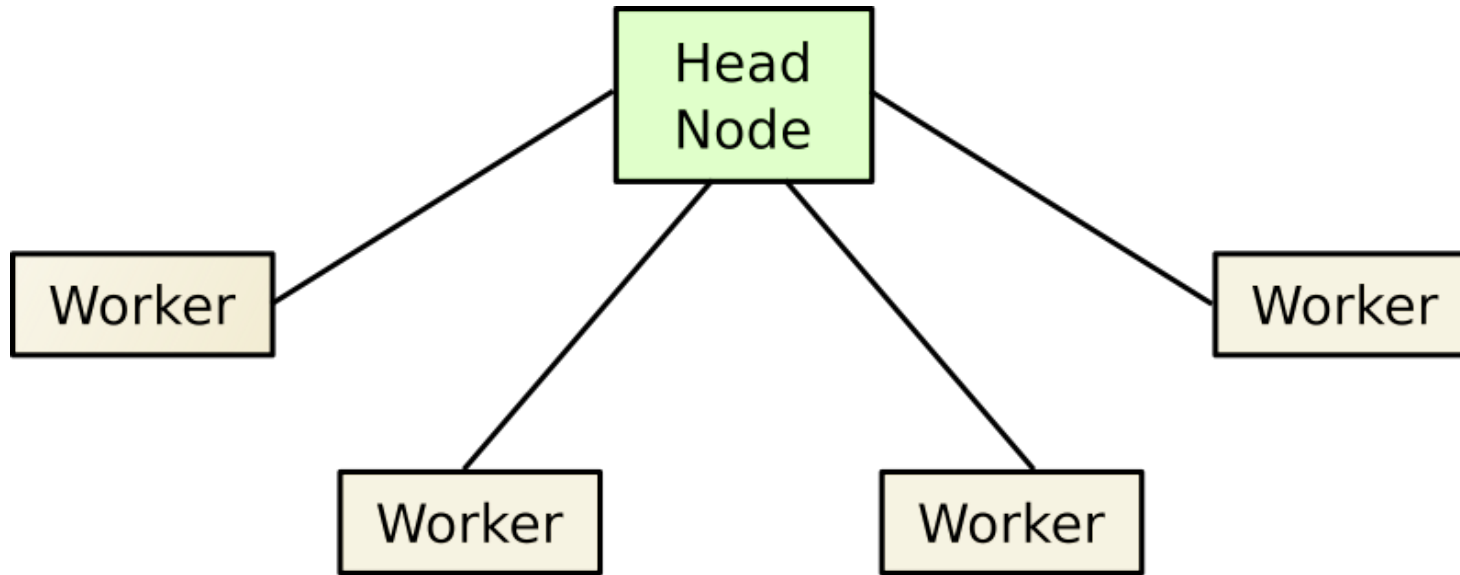
We won't talk about MPI here.

# MapReduce





# A MapReduce Cluster



# What's the fuss?

Google managed to sort 1 *petabyte* of data ...

... in **hours** ...

... in **2011!**

# Implementation

- Practical implementations take care of
  - Parallelization
  - Communication
  - Data distribution
  - Data collection
  - Node failure
- Most used/known implementation
  - Apache Hadoop
    - Comes with `hdfs`, a *distributed file system*.

# Apache Spark

- In-memory MapReduce.
- Data processing.
- Machine learning.
- Streaming.
- Used to sort 100TB in 23 minutes.
  - Using 207 machines on EC2 (2014).
- Used to sort 100TB for \$1.44/terabyte (2016).