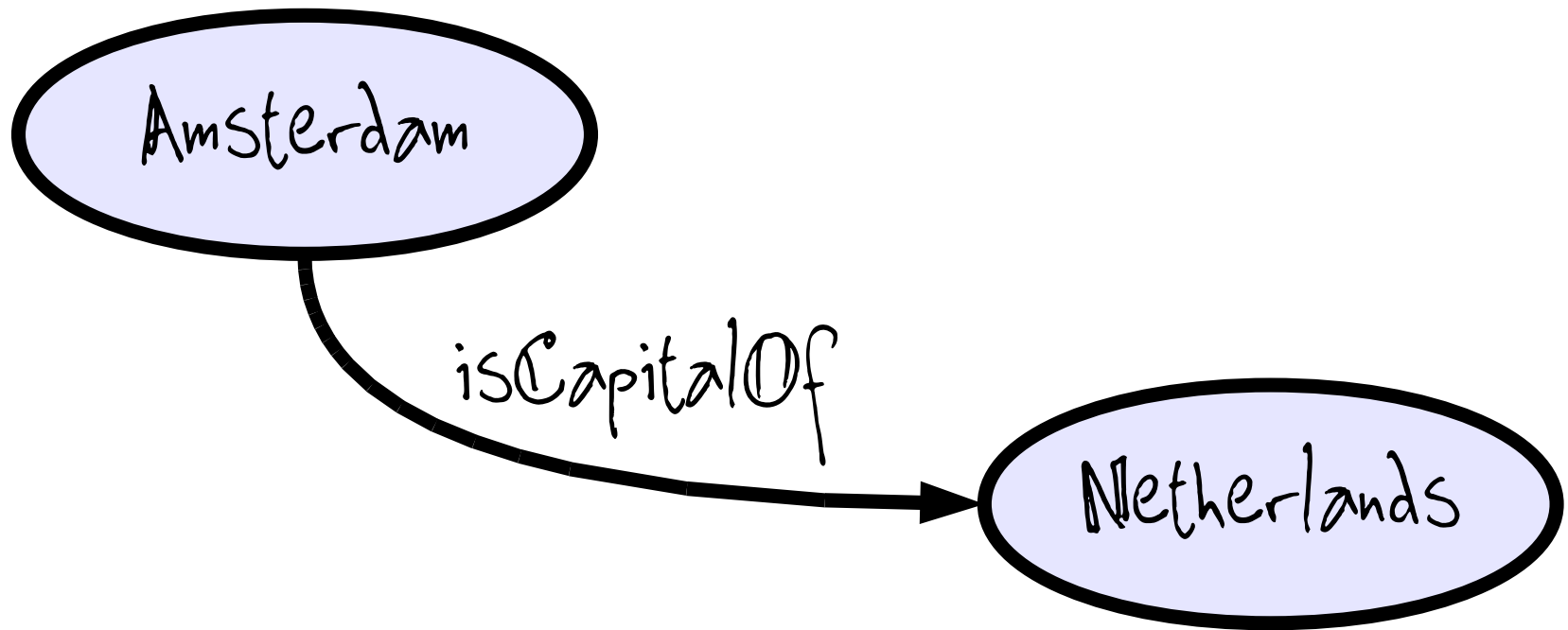


MARVIN: reasoning over a billion triples

# What's a triple?





How much is one billion?

Let's say this is one triple




[Vrandečić, 2007]

This

Suez:  $10^7$  triples  
(10 million)





The moon:  $10^8$  triples

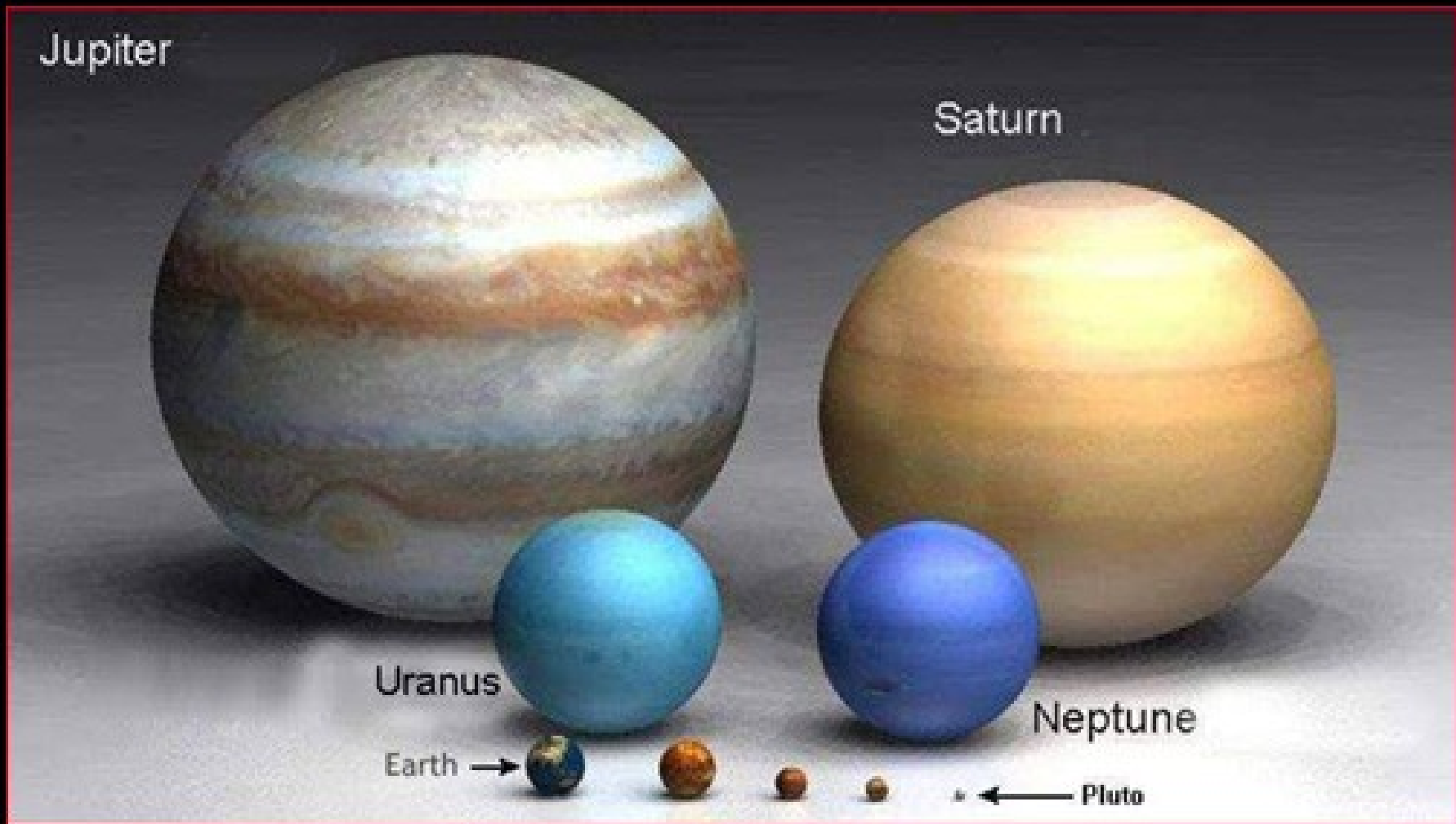
[Vrandecic, 2007]

The earth:  $10^9$  triples  
(1 billion)



[Vrandecic, 2007]

Jupiter:  $10^{10}$  triples  
(1 triple per webpage)



[Vrandečić, 2007]

We need to scale

We need to scale what?

We need to scale reasoning

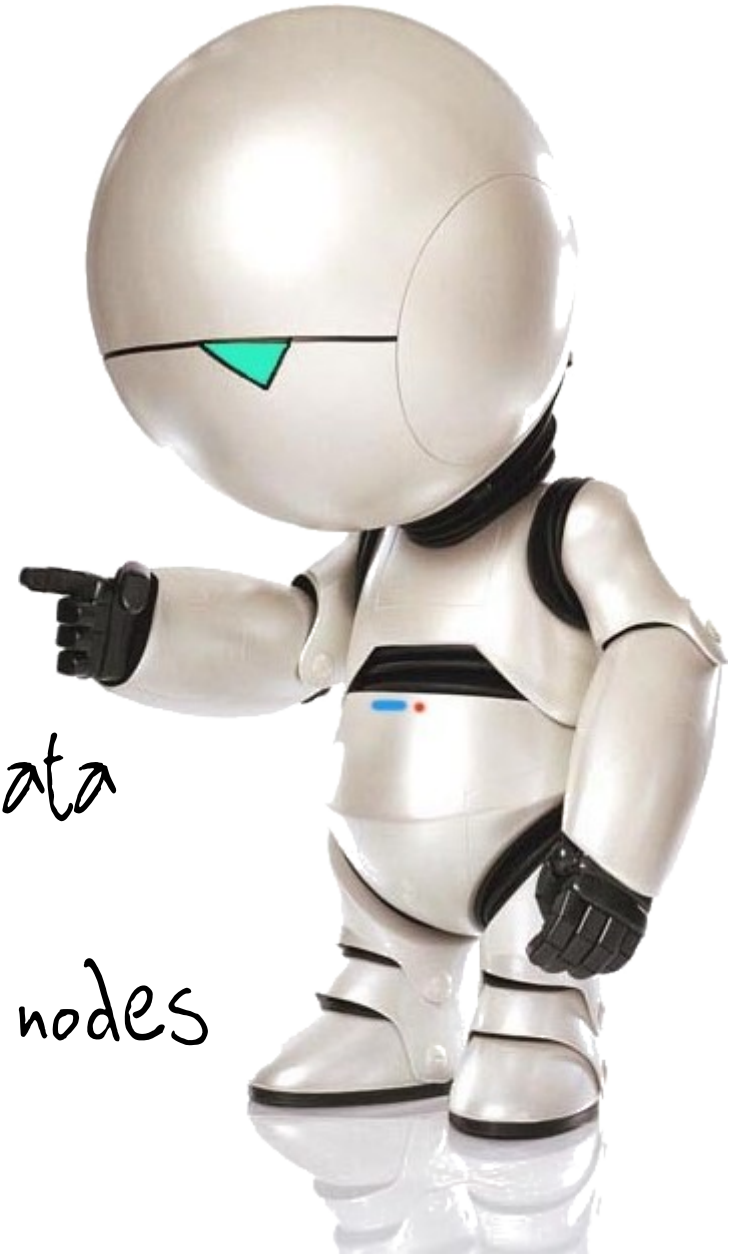
# What is reasoning?

- Amsterdam is a city
- All cities are big
- Thus, Amsterdam is big

Reasoning is: combining pieces of data

# MapReduce

- 1) we have many compute nodes
- 2) each gets piece of data
- 3) does all reasoning on that data
- 4) swaps some data with other nodes
- 5) and goes back to 2)



# What's hard about MapReduce?

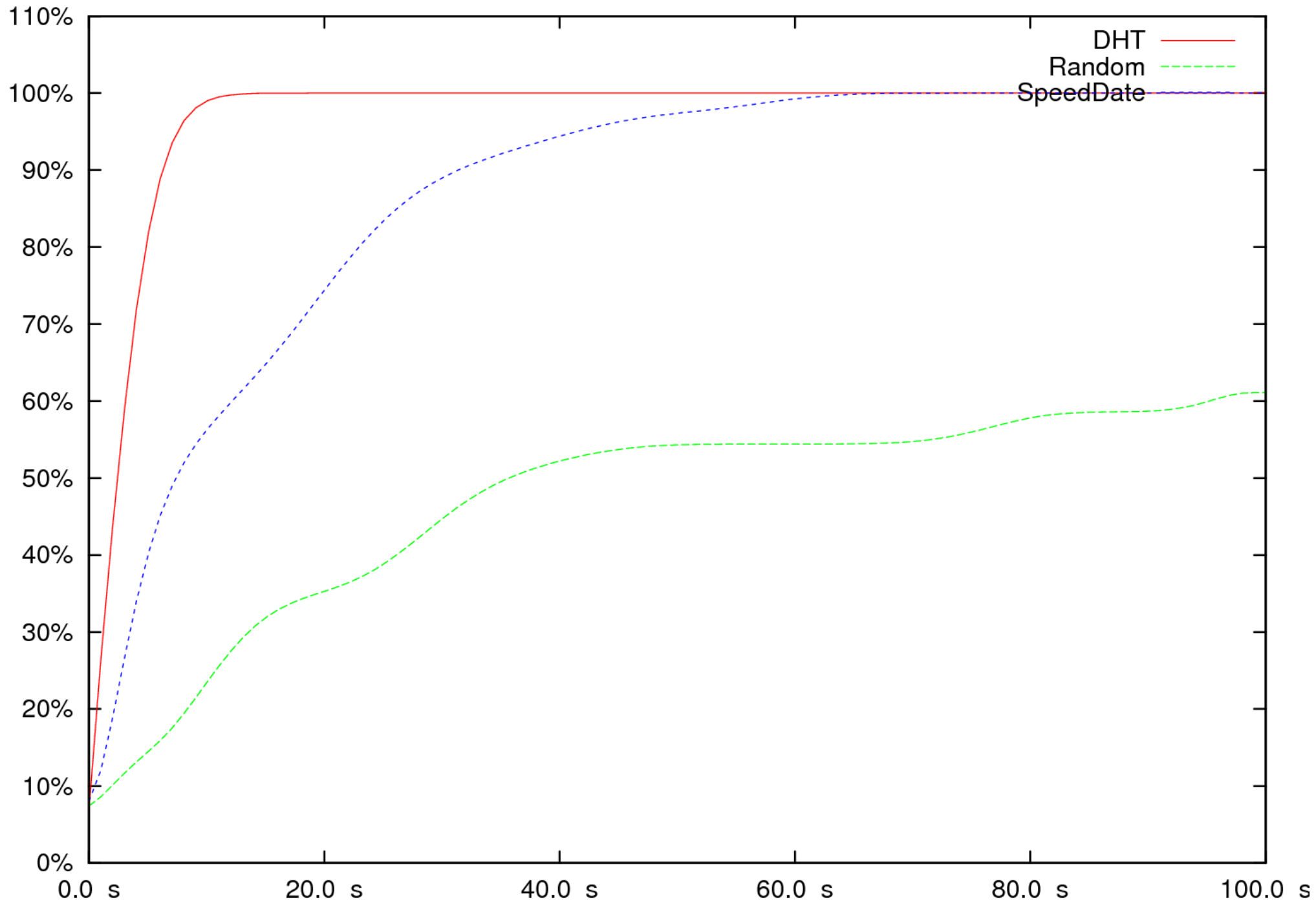
- not all combinations of data are useful  
(Amsterdam is a City & all Dogs are Animals)
- randomly exchanging data is inefficient
- centralised coordination expensive
- want: decentralised, efficient, and loadbalance

# Let's experiment ...

- Random: each triple sent to random peer
- DHT: each triple sent to responsible peer, based on hashcode of triple (push)
- SpeedDating: each peer gets triples closest to the area he's responsible for (pull)

# Let's see some animations

- Random
- DHT
- SpeedDating



# Summary

- billions of triples is a lot
- reasoning over them is not trivial
- we invent & experiment with distribution
- see <http://larkc.eu/marvin>