

Cluster Computing using Apache Spark on Raspberry Pi's



@GarethStep

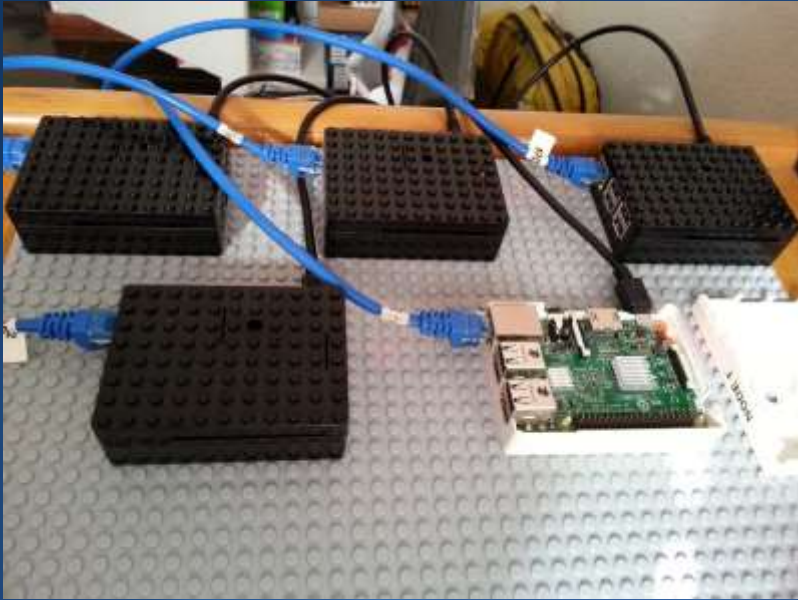
Talk structure

Chapter 1: How to build a Pi cluster

Chapter 2: What is Apache Spark

Chapter 3: The code

Chapter 1: How to build a Pi cluster



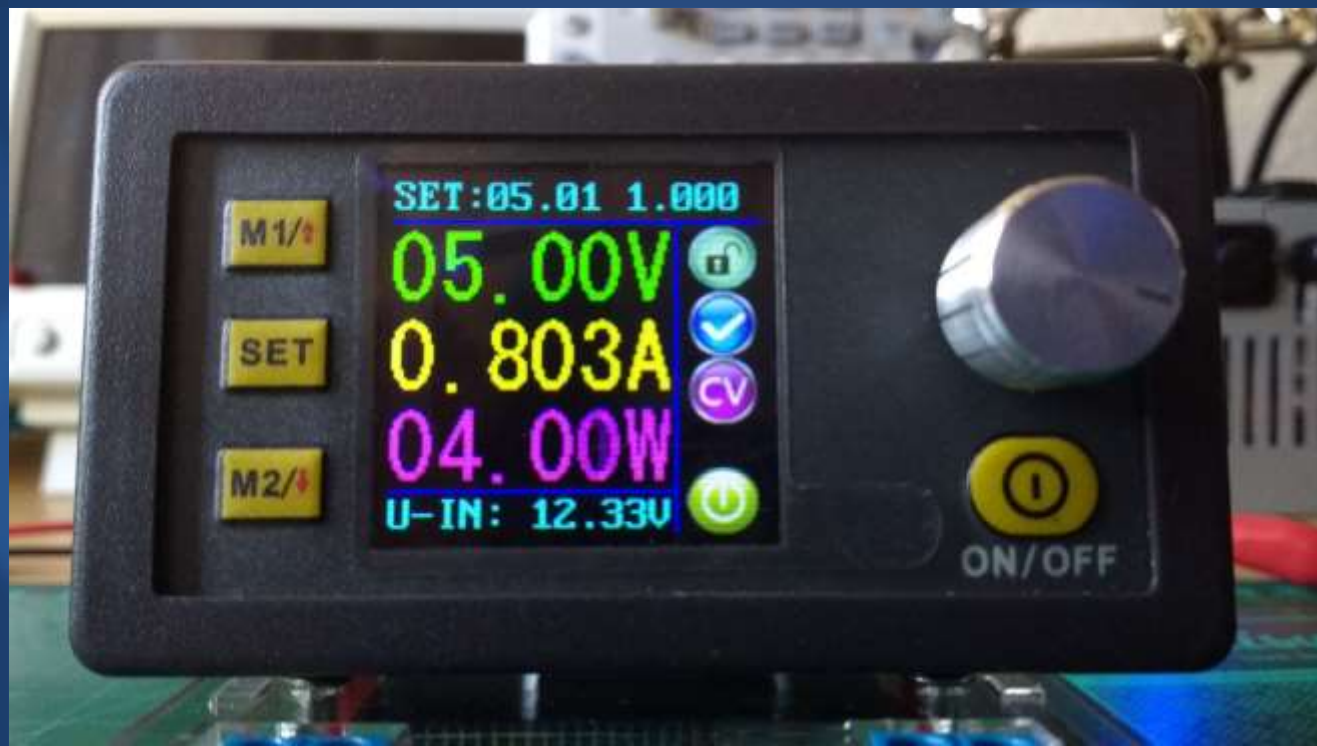
Pi Cluster Version 1.0

5x Pi's: 1 master, 4 slaves

2x Five port network switches

1x 450 watt ATX power supply





The minimum voltage and amperage required
to run a Raspberry Pi 3 at 100% CPU

Pi Cluster Version 2.0

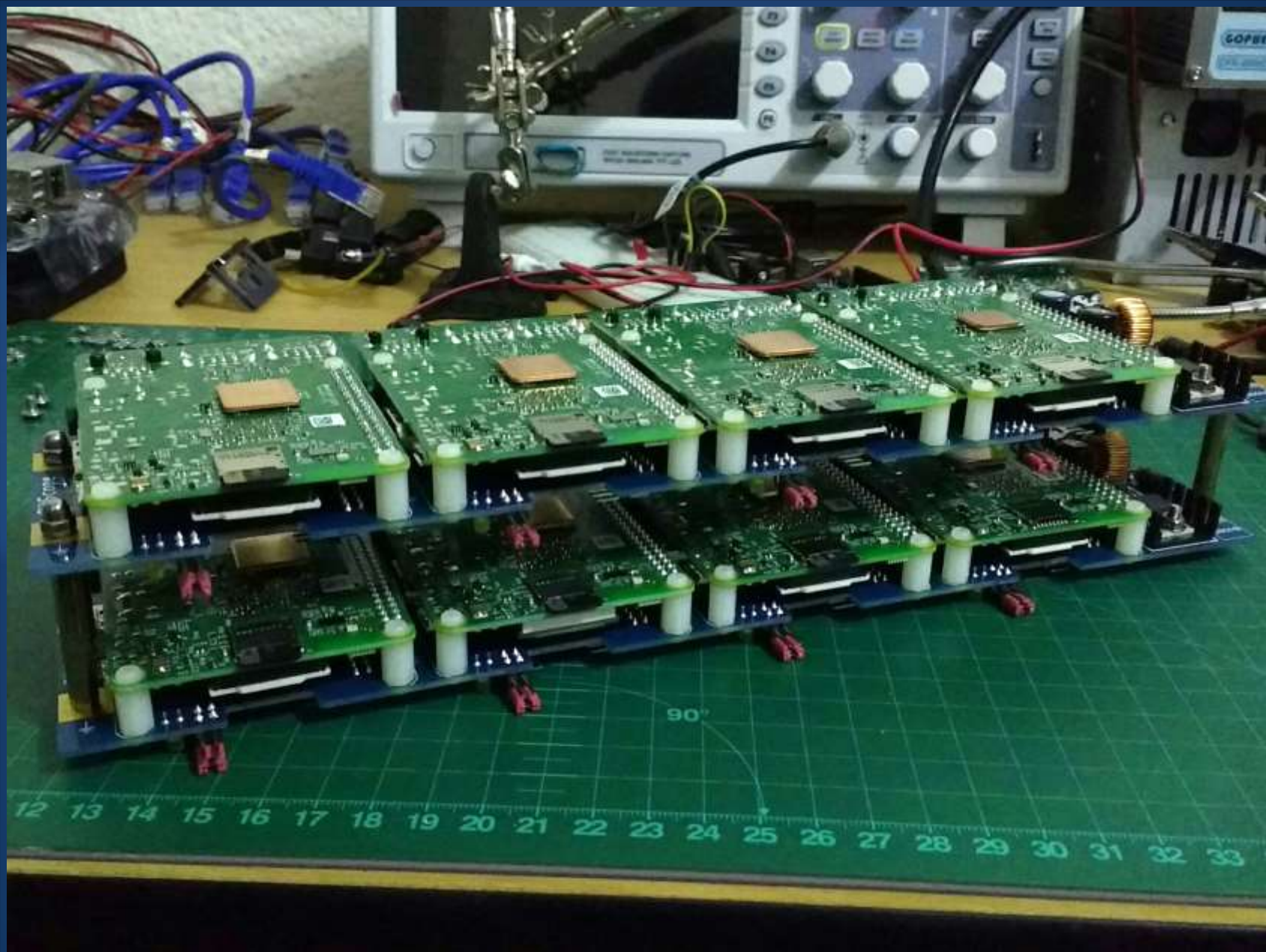
6x Pi's: 1 master, 5 slaves

2x Five port network switches

1x 450 watt ATX power supply
with 2.4A capable USB cables



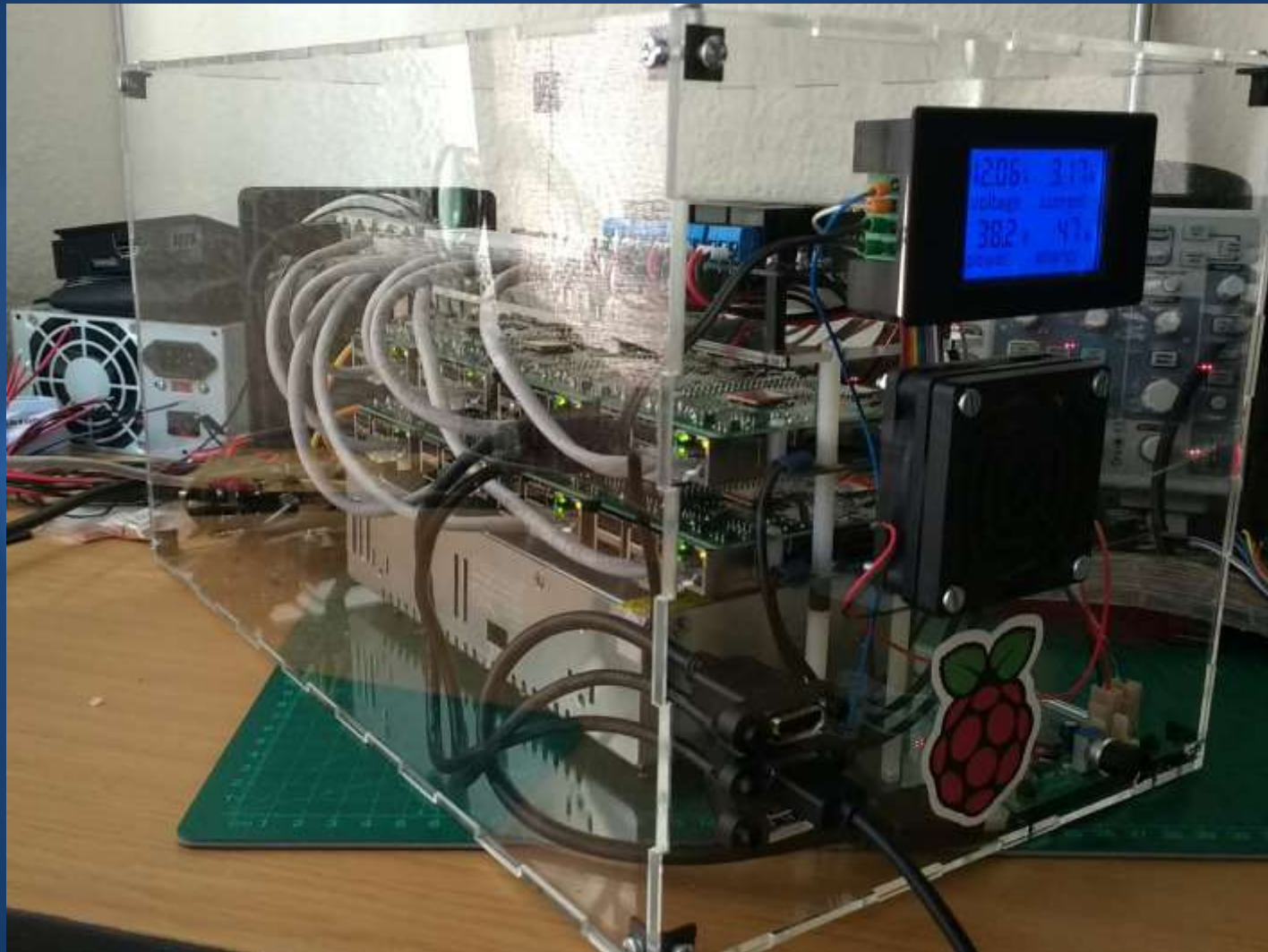








Pi Cluster Version 3.0



12.07 V

voltage

3.17 A

current

38.2 W

power

107 Wh

energy

12.00 V

voltage

7.00 A

current

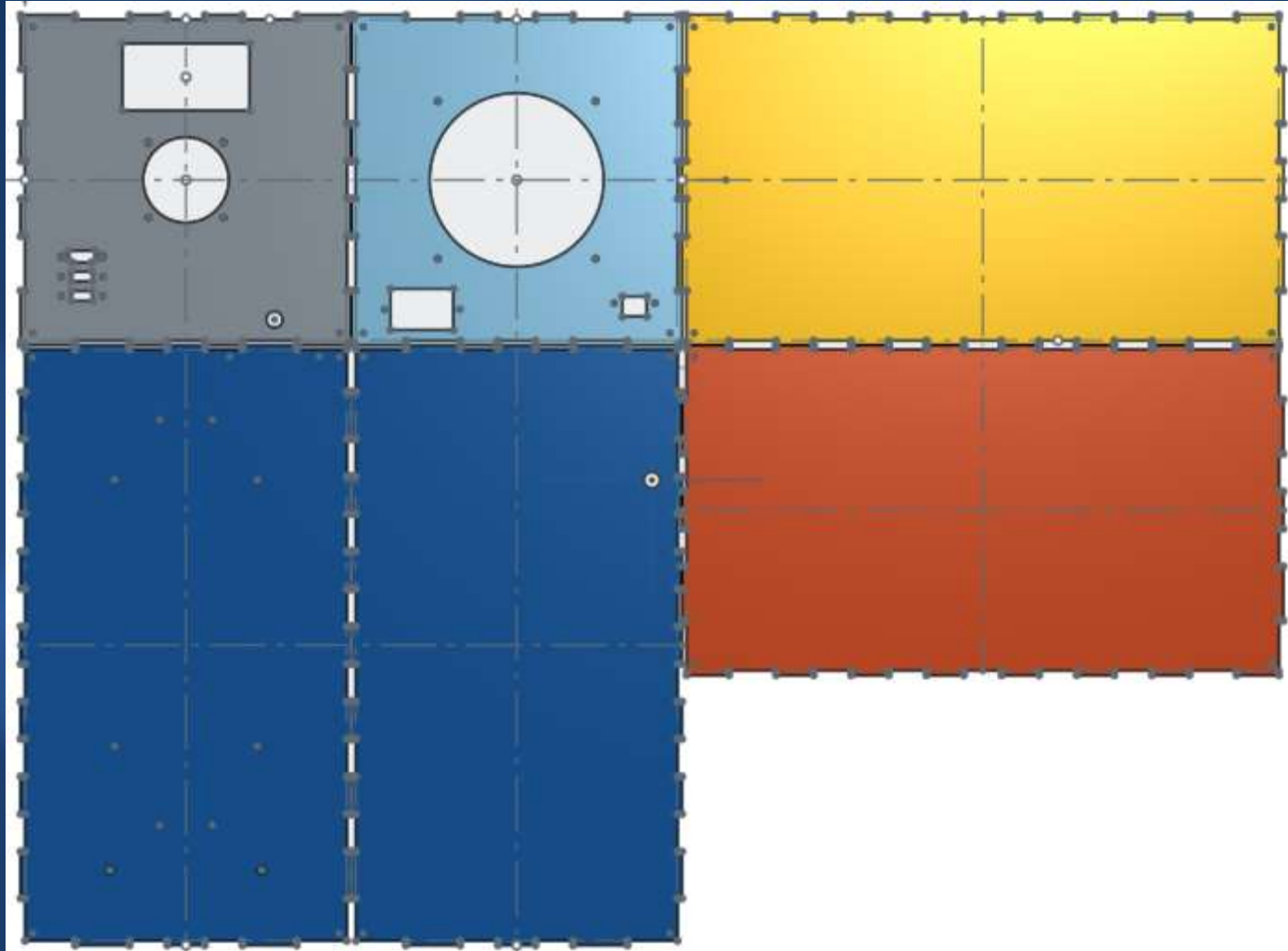
95.1 W

power

27 Wh

energy



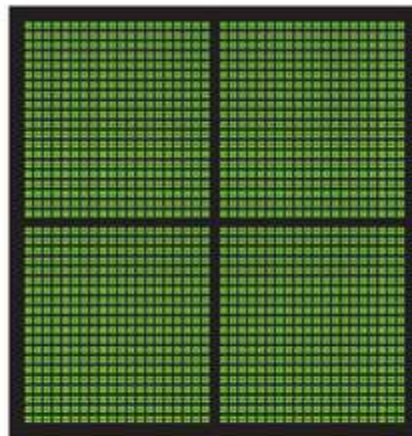


<http://bit.ly/2mOfACC>

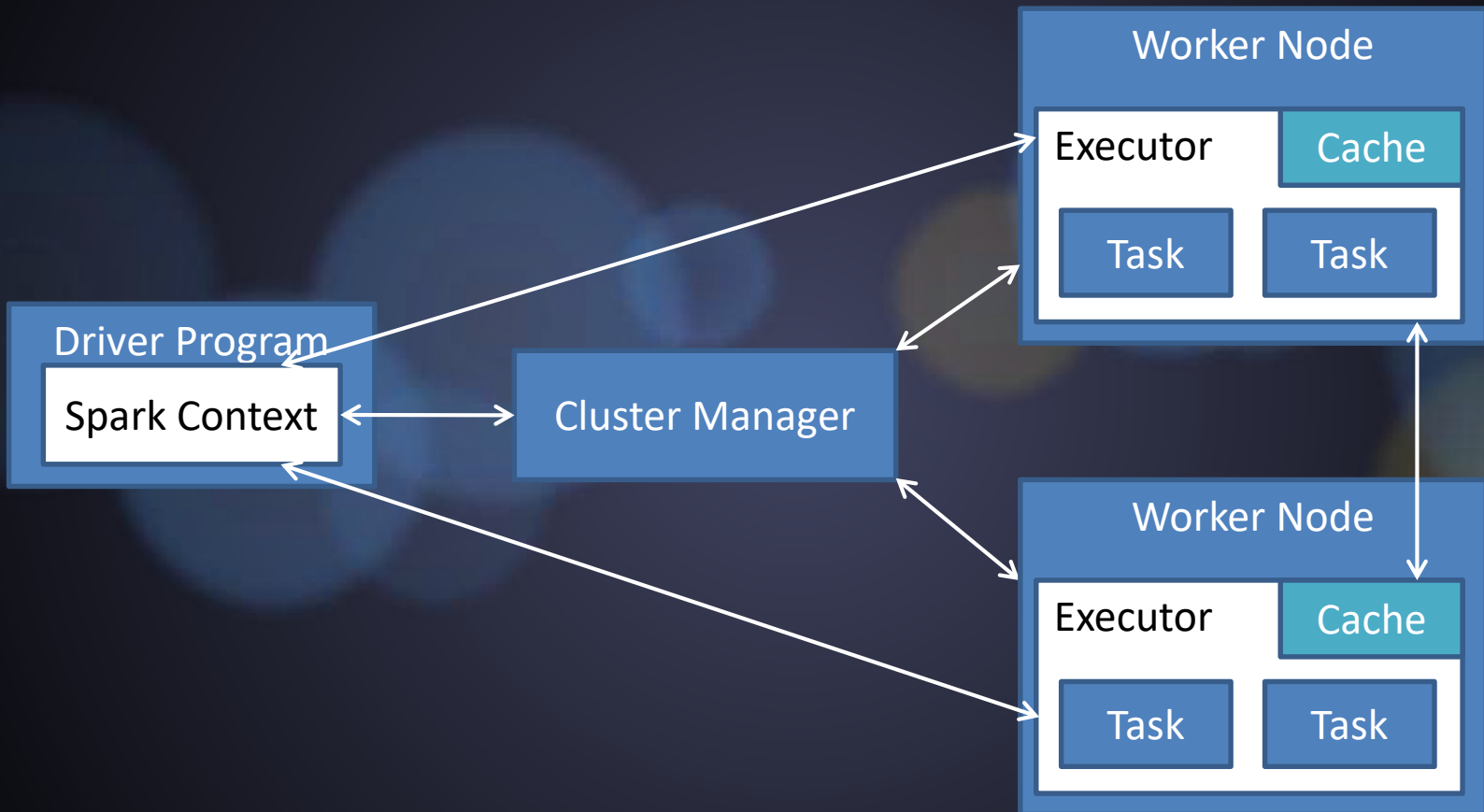
Chapter 2: What is Apache Spark

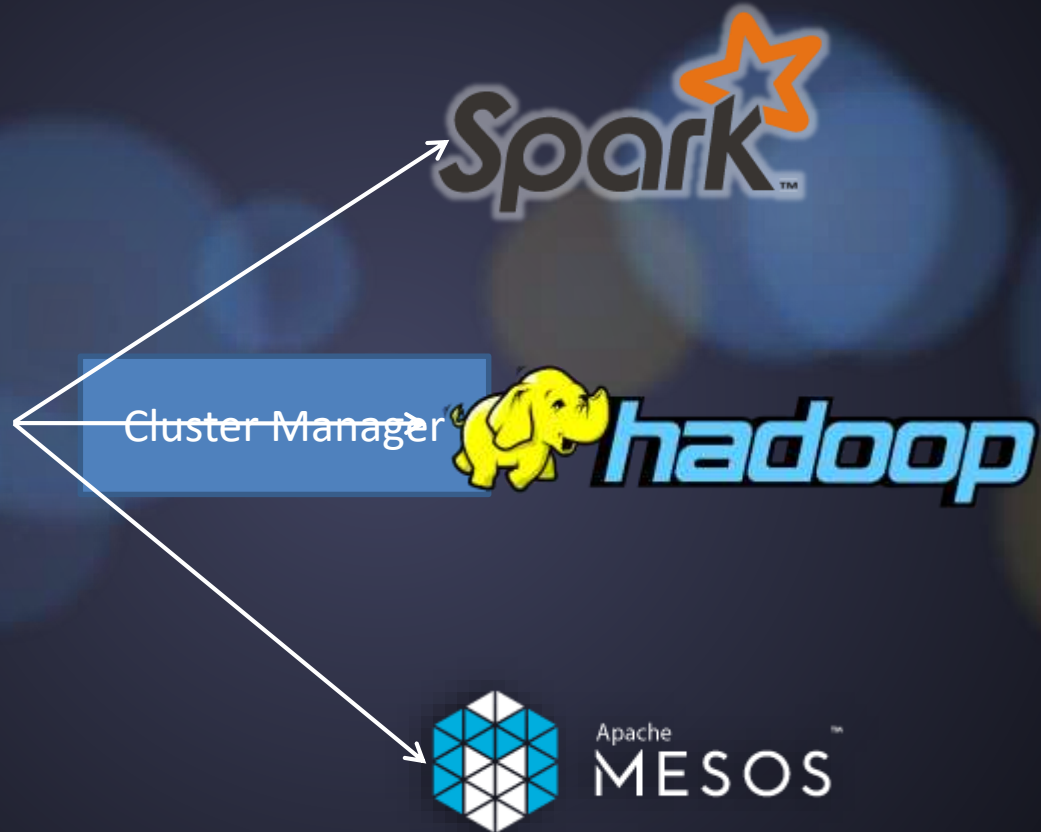


CPU
MULTIPLE CORES



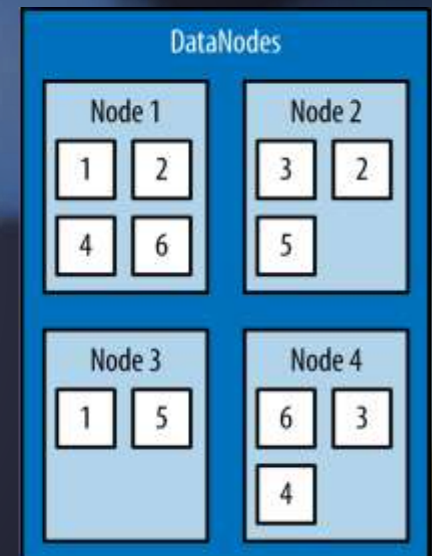
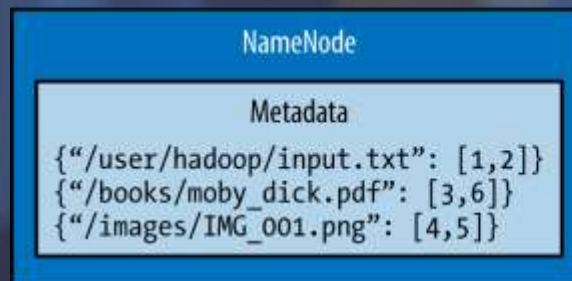
GPU
THOUSANDS OF CORES





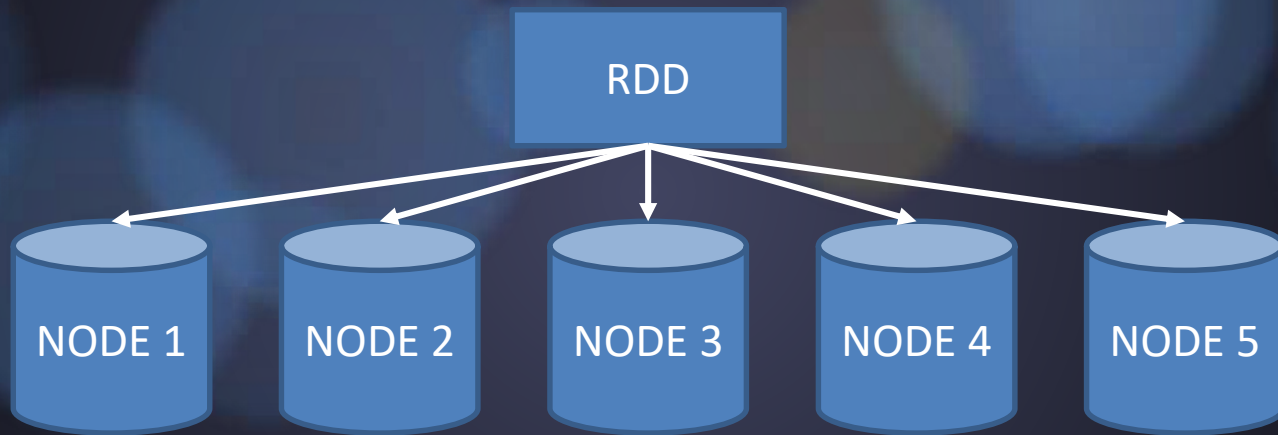
Apache Spark can access various data sources, such as:

- Hadoop's HDFS
- Apache HBASE
- Cassandra
- Amazon S3
- GlusterFS
- Hive
- Tachyon
- Simple NFS
- Or any Hadoop data source...



...in a unified manner, via a mechanism called a
Resilient Distributed Dataset (RDD)

It is an **abstraction** that makes working on distributed collections of data **completely transparent** to the user.



Immutable: Values cannot change.

Distributed: The values in their collections are located on many nodes

Resilient: They are fault-tolerant

Spark SQL

Streaming

MLib

GraphX

Spark Core API

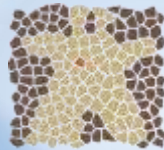
Java

Scala

Python

SQL

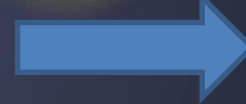
R



A P A C H E
G I R A P H



STORM



Chapter 3: The code

Pi Cluster Power Control



<https://github.com/garethstephenson/PiBladePowerControl>

Apache Spark Compute Job in Scala



https://github.com/garethstephenson/ELEN7046_Group2_2016

The Old Code...

Completed Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
app-20160912073505-0000	Category Count Per Hour []	12	6.0 GB	2016/09/12 07:35:05	spark	FINISHED	3.3 h

Completed Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
app-20160911104344-0000	Category Count Per Hour []	20	768.0 MB	2016/09/11 10:43:44	spark	FINISHED	12.5 h

The New Code...

Completed Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
app-20160822184752-0000	Category Count Per Hour []	12	6.0 GB	2016/08/22 18:47:52	spark	FINISHED	42 s

Completed Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
app-20160822174306-0000	Category Count Per Hour []	20	768.0 MB	2016/08/22 17:43:06	spark	FINISHED	1.6 min

Questions?

Thank you!

