Big Data Processing in Pharo

Matteo Marra

ESUG - August 2019 - Köln

VRIJE UNIVERSITEIT BRUSSEL

Software Languages Lab

fwo

Big Data Frameworks

Programming Model
To express large data processing jobs in terms of simple functions (e.g., Map/Reduce)

Parallel Execution Model
To execute in parallel computation on large amount of data

Why Big Data?

Google MapReduce

Index the WWW
More than 20 billion web pages (> 400 TB)

Do it fast
With 1000 machines: < 3 hours, with one machine, 4 months.

Keep it cheap
Use a lot of recycled hardware, but with robust error recovery

MapReduce: Simplified Data Processing on Large Clusters
Dean J. and Ghemawat S. OSDI '04

Map/Reduce by Example

Dataset	KeyBy	Map
Rossi Abruzzo 1522706400	(Rossi, Abruzzo ...)	NIL
Verdi Toscana 1527976800	(Verdi, Toscana ...)	(Verdi, Toscana ...)
Bianchi Toscana 1525298400	(Bianchi, Toscana ...)	(Bianchi, Toscana ...)
Verdi Lombardia 1527976800	(Verdi, Lombardia ...)	NIL
Gialli Toscana 1525298400	(Gialli, Toscana ...)	(Gialli, Toscana ...)
Rossi Abruzzo 1525298400	(Rossi, Abruzzo ...)	NIL
Gialli Veneto 1527976800	(Gialli, Veneto ...)	NIL
Bianchi Sardegna 1522706400	(Bianchi, Sardegna ...)	NIL
Gialli Toscana 1527976800	(Gialli, Toscana ...)	(Gialli, Toscana ...)
Bianchi Lombardia 1527976800	(Bianchi, Lombardia ...)	NIL
Bianchi Toscana 1522706400	(Bianchi, Toscana ...)	(Bianchi, Toscana ...)
Gialli Sicilia 1522706400	(Gialli, Sicilia ...)	NIL
Gialli Sicilia 1527976800	(Gialli, Sicilia ...)	NIL
Bianchi Sardegna 1527976800	(Bianchi, Sardegna ...)	NIL

GroupBy

(Verdi, Toscana)

(Bianchi, Toscana ...)
(Bianchi, Toscana ...)

(Gialli, Toscana ...)
(Gialli, Toscana ...)

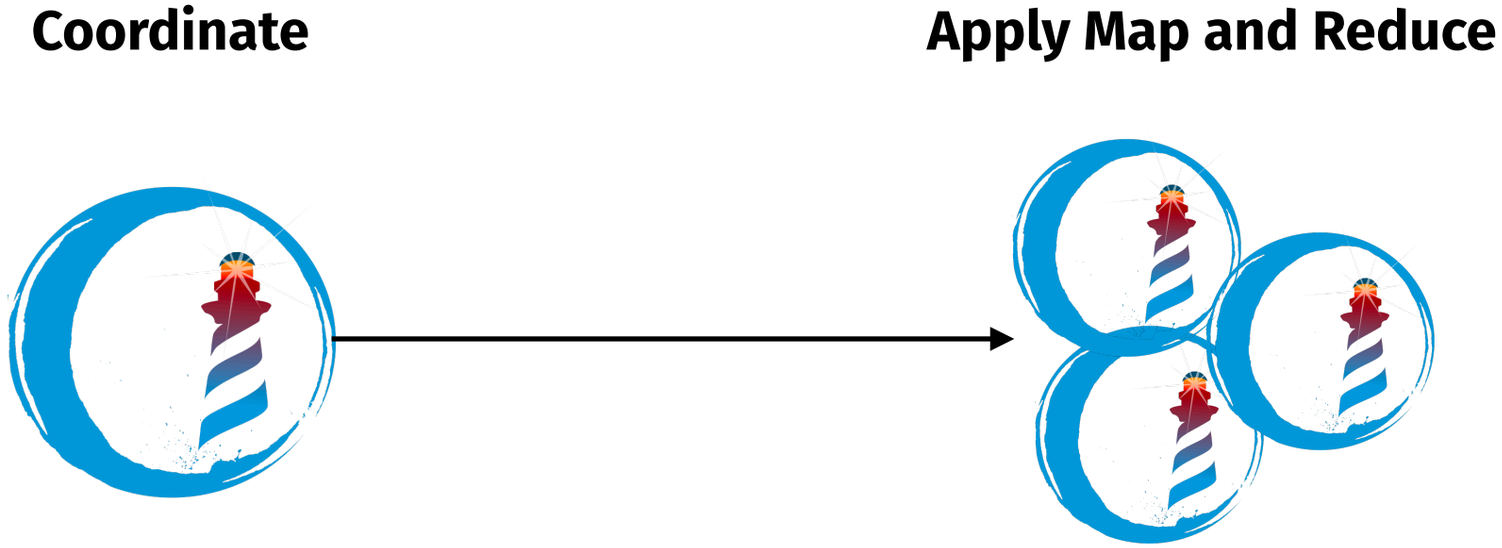
Reduce

(Verdi, 1)
(Bianchi, 2)
(Gialli, 2)

A Map/Reduce application in Pharo

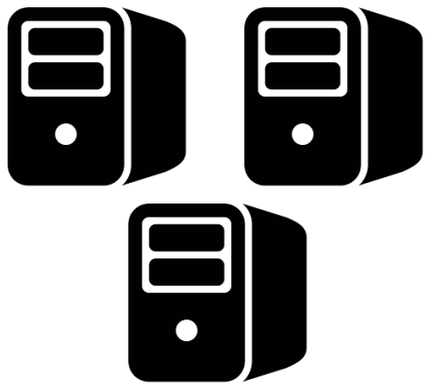
```
VoteCountingApp>>map: aLine
| splitted|
 (line includesSubstring: ‘Toscana')
          ifTrue: [ splitted := line substrings: ' '.
               (DateAndTime fromUnixTime: splitted at: 3)
                    > DateAndTime yesterday
                    ifTrue: [ ^ (splitted at: 2) -> 1 ] ].
     ^ nil -> nil

VoteCountingApp>>reduceByKey: values
    ^ values inject: 0 into: [:sum :current | sum + current]
```

Port: a Map/Reduce Framework for Pharo

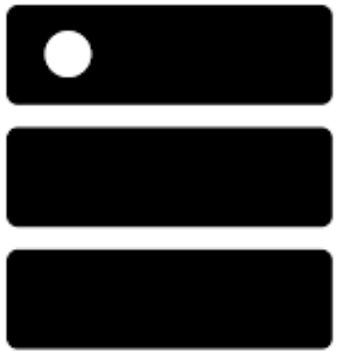


Deploying Port

Local
Deploy on one machine using multiple cores.

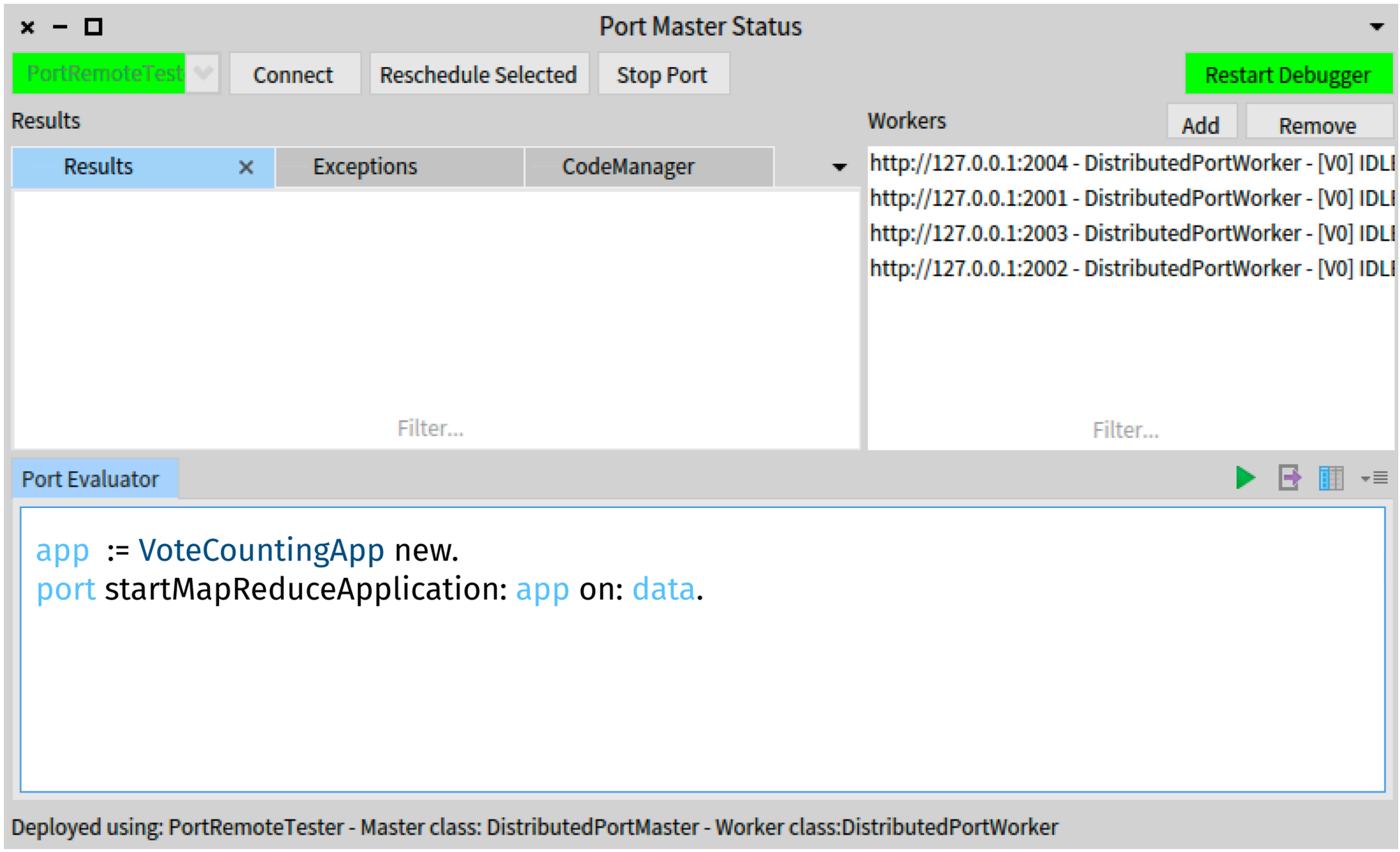


Standalone
Deploy on multiple machines in the same network.



Yarn Mode
Deploy on a cluster using Hadoop YARN.

Using Port

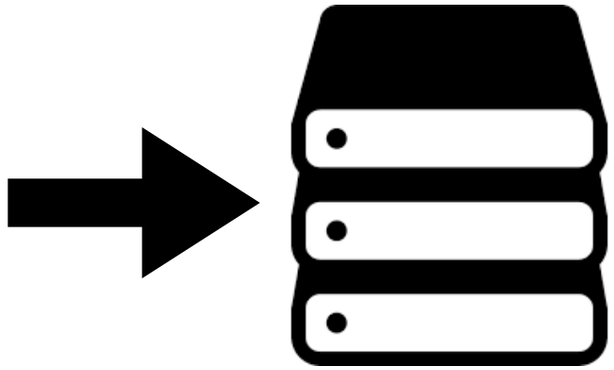

Port Master Status
PortRemoteTest
Connect
Reschedule Selected
Stop Port
Restart Debugger
Results
Workers
Add
Remove
Results
Exceptions
CodeManager
http://127.0.0.1:2004 - DistributedPortWorker - [V0] IDLE
http://127.0.0.1:2001 - DistributedPortWorker - [V0] IDLE
http://127.0.0.1:2003 - DistributedPortWorker - [V0] IDLE
http://127.0.0.1:2002 - DistributedPortWorker - [V0] IDLE
Filter...
Filter...
Port Evaluator
app := VoteCountingApp new.
port startMapReduceApplication: app on: data.
Deployed using: PortRemoteTester - Master class: DistributedPortMaster - Worker class:DistributedPortWorker

Handling Results

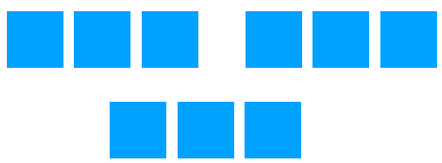
Asynchronous Execution
The application is executed asynchronously.

Asynchronous Result Handling
The result is handled asynchronously by the Map/Reduce Master

```
VoteCountingApp>>handleResult: resultPair
    self storeToHDFS: resultPair
```



From Map/Reduce to Spark



Distributed Collection

Instead of executing Map/Reduce on a Dataset, you load the dataset in memory

aggregate:
filter:
map:
reduce:
groupBy:

Broader API

Not only execute map or reduce, but a broad set of methods applicable to the distributed collection

map:  **reduce:**

In-Memory operations

Avoid heavy storing of intermediate results by explicitly keep them in memory

The Spark-like API in Port

```
port startMapReduceApplication: app
    on: dataSet .
```

```
VoteCountingApp>>run

data := port distribute: dataSet.
```

```
VoteCountingApp>>map: aLine
  ...
```

```
result := data map: [...].
```

```
VoteCountingApp>>handleResult: res
    self storeToHDFS: res
```

```
result storeToHDFSPath: '...' .
```

The Polls Analyser in Port

(V2)

```
port := Port withServerURL: 'http://myServerURL'.
data := port  distribute: dataSet.
data := data filter: [:line | line includesSubstring: 'Toscana'].
data map: [:line | splitted := line substrings: ' ' .
                  (DateAndTime fromUnixTime: splitted at: 3)
                    > DateAndTime yesterday
                    ifTrue: [  (splitted at: 2) -> 1 ]]
result := data reduceByKey: [:value :sum| sum + value]
```

```
result getCollection.
result storeToHDFSPath: '...'
```

DEMO

Uses of Port

Blockchain Analysis
In collaboration with Santiago at RMoD Lille.
Full blockchain in 6 hours vs 2 days!

Genetic algorithms
In collaboration with Alexandre Bergel at Universidad de Chile

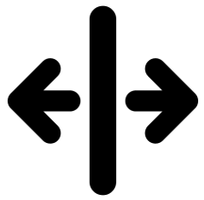
Towards Live Big Data Development Tools

Online Debugging

Debug the system as the bug is happening

Global View

Centralised debugging of the distributed system



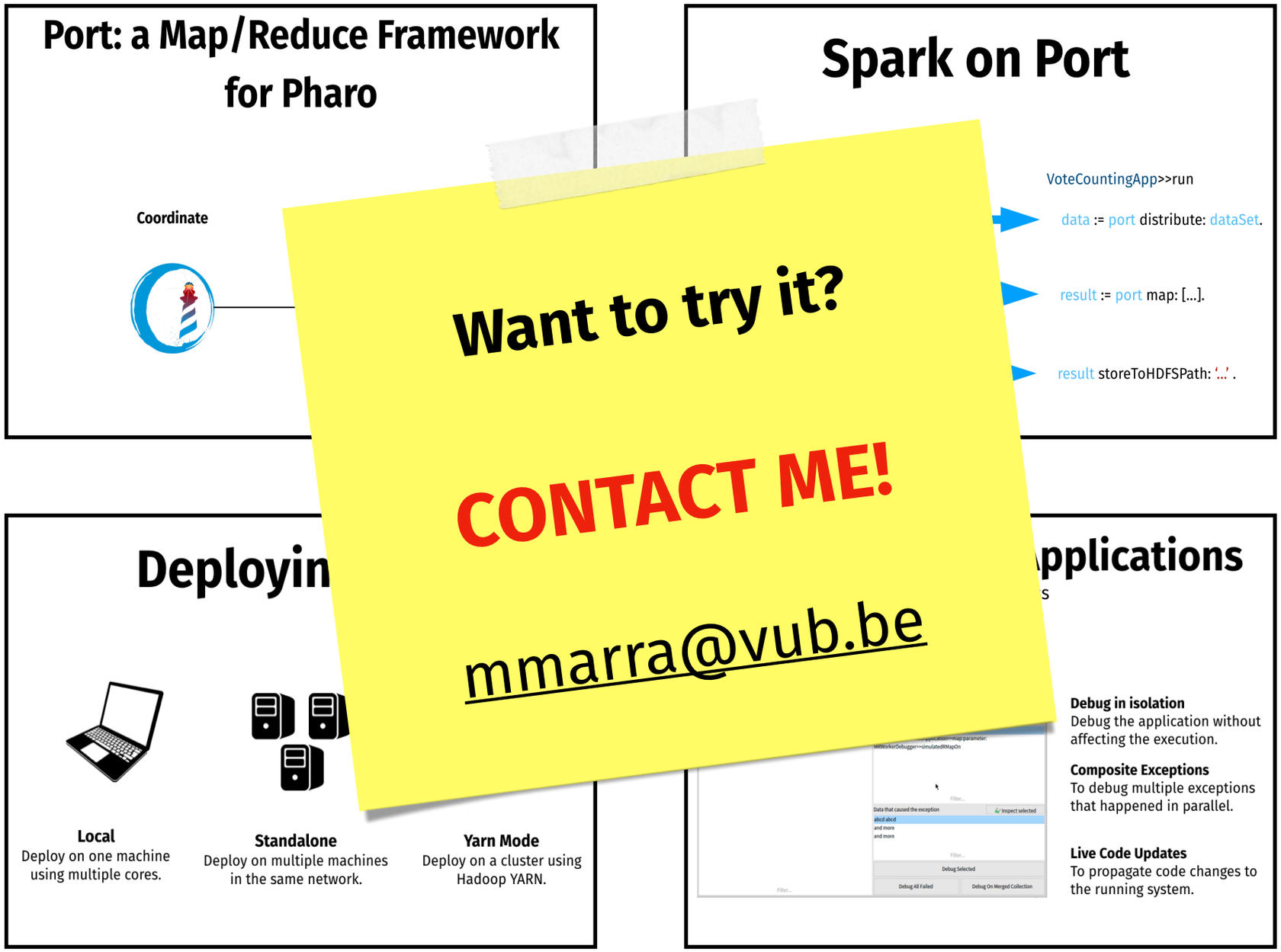
Isolation

Debug the system without interfering with its execution

Update of the Running System

Deploy code-fixes without restarting the whole system

Conclusion



Want to try it?

CONTACT ME!

mmarra@vub.be

Matteo Marra - Vrije Universiteit Brussel

mmarra@vub.be gitlab.soft.vub.ac.be/Marra github.com/Marmat21