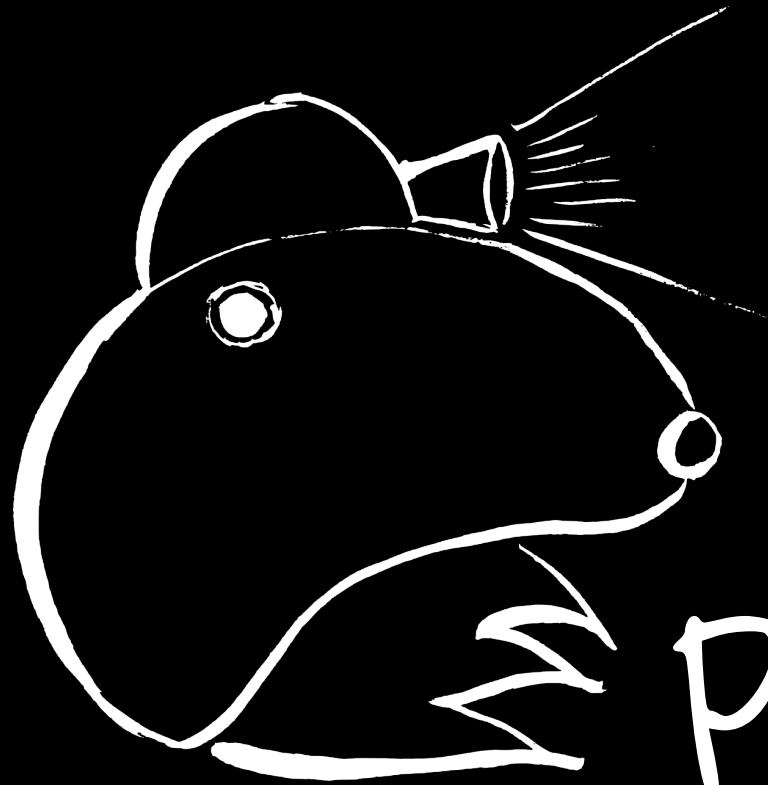


Distributing a (Linux) native  
application with



penMOLE

# Prerequisite

You should have:

- OpenMOLE (5.4)
- A Linux computer  
(to practice packaging)
- Python

# Get the archives

Go to [www.openmole.org/files/csdc](http://www.openmole.org/files/csdc)

Get packaging.tgz and Native.tgz

Extract packaging.tgz

# A native application

```
reuillon@simplet: /home/reuillon/Documents/Recherche/Work/OpenMOLE/python
import sys
import numpy
from numpy import *
from array import *
import csv

input = open(sys.argv[1], 'r')
n = float(sys.argv[2])

print("reading the matrix")
data = csv.reader(input)

array = numpy.array(list(data)).astype(float)

print(array)
print(n)
mult = array * n

print("saving the matrix")
numpy.savetxt(sys.argv[3], mult, fmt='%g')
2,1 33%
```

Content of "matrix.py"

# Run it

```
[reuillon:~/Documents ... ond/application] 2s $ python matrix.py data.csv 2 out.csv
reading the matrix
[[ 8.  8.  9.]
 [ 7.  6.  8.]
 [ 5.  7.  8.]]
2.0
saving the matrix
[reuillon:~/Documents ... ond/application] 6s $ ls
care data.csv matrix.py out.csv
[reuillon:~/Documents ... ond/application] 5s $
```

Reads data.csv

Multiply it by 2

Write out.csv

# Make it portable

This application depends on:

- python
- numpy
- libc...

The solution: use CARE

<http://reproducible.io/>

# Make it portable

```
$/care -o python.bin python matrix.py data.csv 2 out.csv
care info: concealed path: $HOME /home/reuillon
care info: concealed path: /tmp
care info: revealed path: $PWD /home/reuillon/Documents/Recherche/Work/OpenMOLE/python/second/application
care info: revealed path: /usr/bin/python2.7
care info: -----
reading the matrix
[[ 8.  8.  9.]
 [ 7.  6.  8.]
 [ 5.  7.  8.]]
2.0
saving the matrix
care info: -----
care info: Hints:
care info:   - search for "conceal" in `care -h` if the execution didn't go as expected.
care info:   - run `./python.bin` or `care -x python.bin` to extract the output archive correctly.
$ls
care data.csv matrix.py out.csv python.bin
$
```

# Make it portable

What is python.bin?

- An auto-extractible archive
- Contains every single files accessed by the program (and a bit more)

How do I use it?

- Extract: `./python.bin`
- Run: `./python/re-execute.sh`



# Make it portable

## What's inside python.bin?

```
reuillon@simplet: /home/reuillon/Documents/Recherche/Projects/open... x reuillon@simplet: /home/reuillon/Documents/Recherche/Work/OpenM... x reuillon@simplet: /home/reuillon/Documents/Recherche/Work/OpenM... x
$tree python
python
├── concealed-accesses.txt
├── proot
├── README.txt
├── re-execute.sh
├── rootfs
│   ├── dev
│   ├── etc
│   │   ├── alternatives
│   │   │   ├── libblas.so.3 -> /usr/lib/libblas/libblas.so.3
│   │   │   └── liblapack.so.3 -> /usr/lib/lapack/liblapack.so.3
│   │   ├── ld.so.cache
│   │   ├── localtime
│   │   ├── python2.7
│   │   │   └── sitecustomize.py
│   └── home
│       ├── reuillon
│       │   ├── Documents
│       │   │   ├── Recherche
│       │   │   │   ├── Work
│       │   │   │   │   ├── OpenMOLE
│       │   │   │   │   │   └── python
│       │   │   │   │   │       ├── data.csv
│       │   │   │   │   │       ├── matrix.py
│       │   │   │   │   │       └── out.csv
```

Emulates your system

Re-executes your program

rootfs/ is your file system

# Make it portable

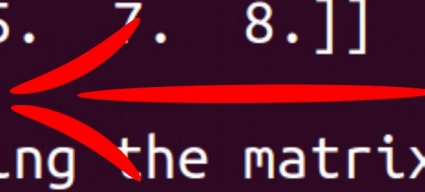
```
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/core.pyc
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/extras.py
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/extras.pyc
info: extracted: python/rootfs/usr/lib/python2.7/csv.py
info: extracted: python/rootfs/usr/lib/python2.7/csv.pyc
info: extracted: python/rootfs/usr/lib/python2.7/lib-dynload/_csv.x86_64-linux-gnu.so
info: extracted: python/rootfs/home/reuillon/Documents/Recherche/Work/OpenMOLE/python/second/application/data.csv
info: extracted: python/rootfs/home/reuillon/Documents/Recherche/Work/OpenMOLE/python/second/application/out.csv
info: extracted: python/re-execute.sh
info: extracted: python/concealed-accesses.txt
info: extracted: python/README.txt
info: extracted: python/proot
$ ./python/re-execute.sh
reading the matrix
[[ 8.  8.  9.]
 [ 7.  6.  8.]
 [ 5.  7.  8.]]
2.0
saving the matrix
$
```

It will work on any other Linux machine

# Make it portable

And you can change the parameters

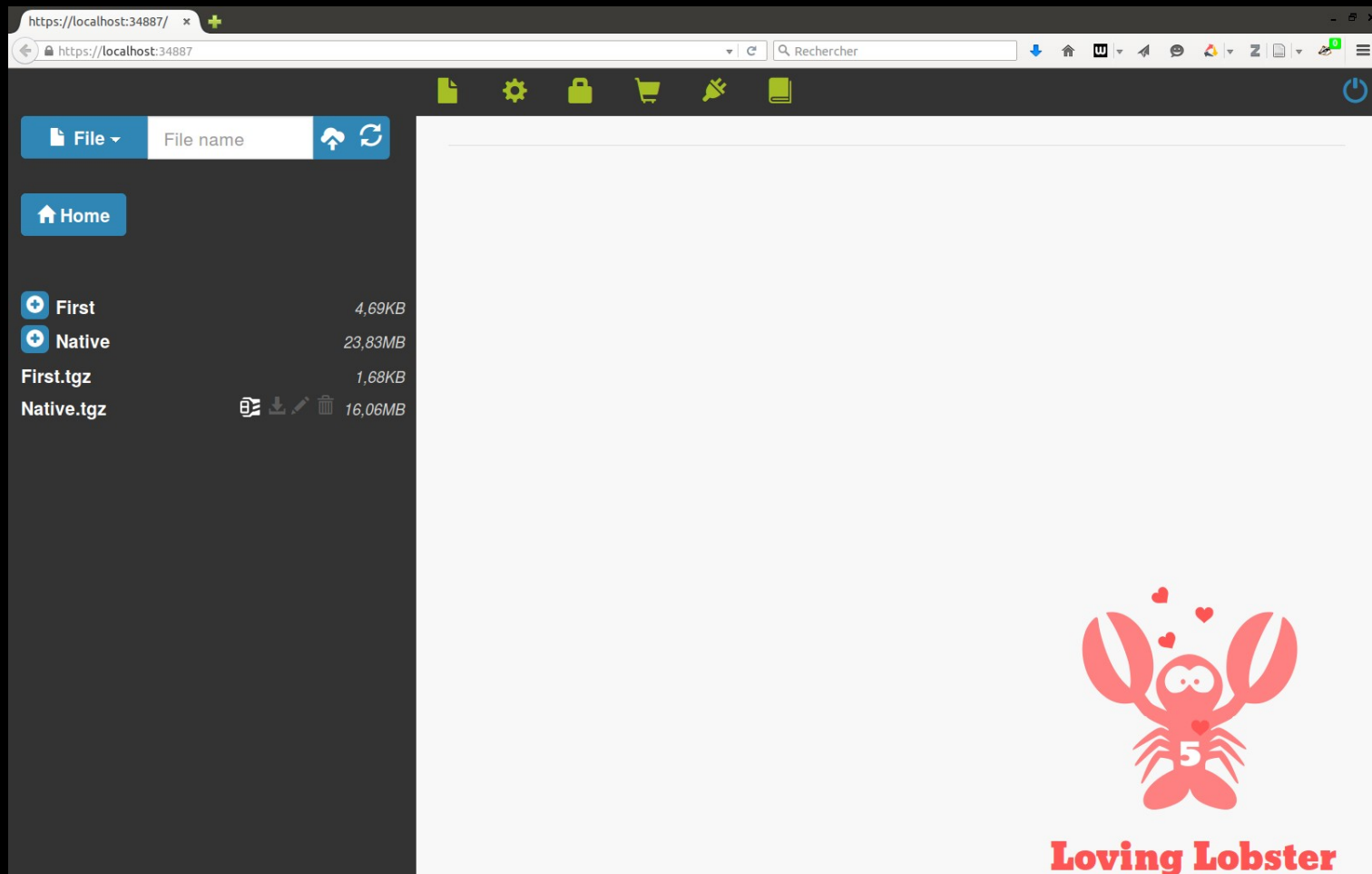
```
$/python/re-execute.sh python matrix.py data.csv 4 out.csv  
reading the matrix  
[[ 8.  8.  9.]  
 [ 7.  6.  8.]  
 [ 5.  7.  8.]]  
4.0  
saving the matrix  
$
```



Now it's 4.0

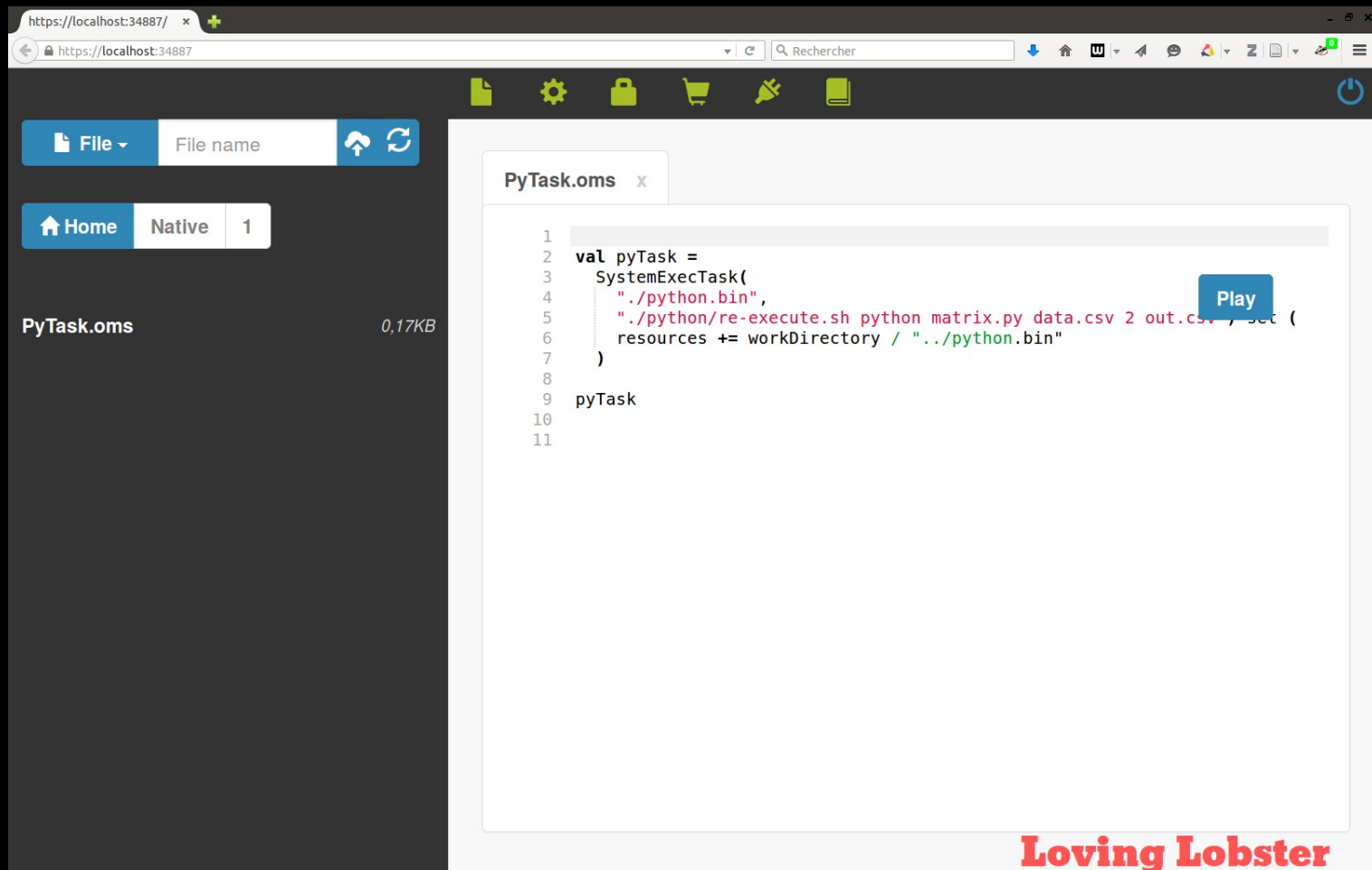
# Run it in OpenMOLE

Launch OpenMOLE, upload native.tgz and extract it.



# Run it in OpenMOLE

Go to "Native/1" and edit PyTask.oms



# Run it in OpenMOLE

## Use a SystemExecTask:

```
2  val pyTask =  
3    SystemExecTask(  
4      "./python.bin",  
5      "./python/re-execute.sh python matrix.py data.csv 2 out.csv") set (  
6        resources += workDirectory / "../python.bin"  
7    )
```

Commands



Required files

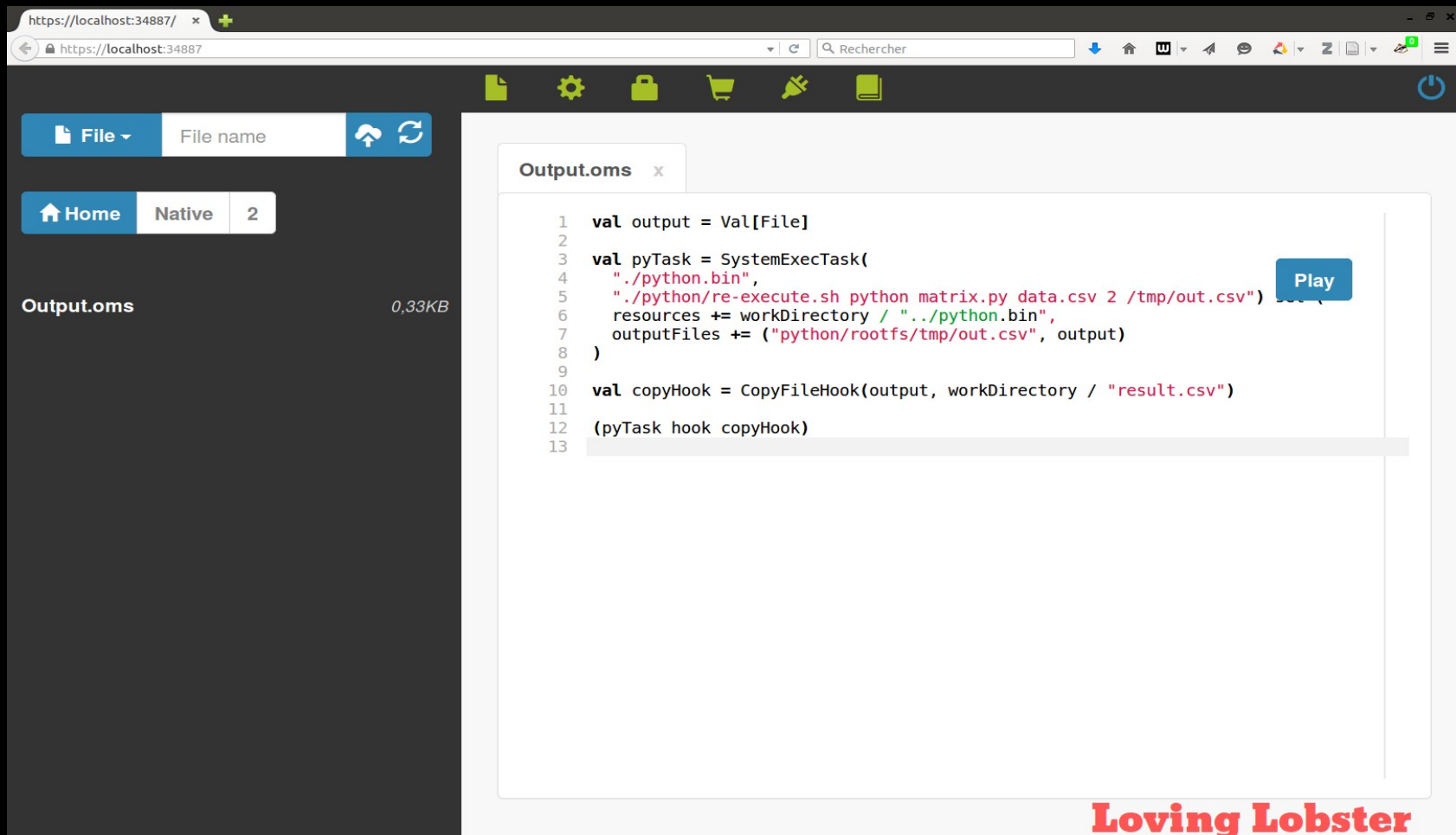


# Run it in OpenMOLE

The screenshot displays the OpenMOLE web interface in a browser. The address bar shows `https://localhost:34887/`. The interface includes a top navigation bar with icons for file, settings, task, and other functions. A sidebar on the left contains a home icon and a task icon. The main content area is titled "Executions" and features a control bar with "Environment error level" (set to "ERROR"), "Output history" (set to "500"), and a task status bar. The task status bar for "PyTask.oms" shows a timestamp of "30/9/2015, 14:33:20", a lightning bolt icon, "1 / 1", a duration of "0:00:00", and a status of "finished". Below this, a scrollable log displays the following text:

```
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/__init__.py
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/__init__.pyc
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/core.py
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/core.pyc
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/extras.py
info: extracted: python/rootfs/usr/lib/python2.7/dist-packages/numpy/ma/extras.pyc
info: extracted: python/rootfs/usr/lib/python2.7/csv.py
info: extracted: python/rootfs/usr/lib/python2.7/csv.pyc
info: extracted: python/rootfs/usr/lib/python2.7/lib-dynload/_csv.x86_64-linux-gnu.so
info: extracted: python/rootfs/home/reuillon/Documents/Recherche/Work/OpenMOLE/python/data.csv
info: extracted: python/rootfs/home/reuillon/Documents/Recherche/Work/OpenMOLE/python/out.csv
info: extracted: python/re-execute.sh
info: extracted: python/concealed-accesses.txt
info: extracted: python/README.txt
info: extracted: python/proot
reading the matrix
[[ 8. 8. 9.]
 [ 7. 6. 8.]
 [ 5. 7. 8.]]
2.0
saving the matrix
```

Get the output file  
Go to "2" and edit output.oms



The screenshot shows a web browser window at <https://localhost:34887/>. The application has a dark sidebar on the left with a 'File' menu, a search bar, and tabs for 'Home', 'Native', and '2'. The 'Native' tab is selected, and the file 'Output.oms' (0.33KB) is listed. The main area shows the file's content in a code editor. The code is in Scala and defines a task to execute a Python script. A 'Play' button is visible next to the code. The text 'Loving Lobster' is at the bottom right.

```
1 val output = Val[File]
2
3 val pyTask = SystemExecTask(
4   "./python.bin",
5   "./python/re-execute.sh python matrix.py data.csv 2 /tmp/out.csv"
6   resources += workDirectory / "../python.bin",
7   outputFiles += ("python/rootfs/tmp/out.csv", output)
8 )
9
10 val copyHook = CopyFileHook(output, workDirectory / "result.csv")
11
12 (pyTask hook copyHook)
13
```

Loving Lobster



# Get the output file

```
1 val output = Val[File]
2
3 val pyTask = SystemExecTask(
4   "./python.bin",
5   "./python/re-execute.sh python matrix.py data.csv 2 /tmp/out.csv" set {
6     resources += workDirectory / "../python.bin",
7     outputFiles += ("python/rootfs/tmp/out.csv", output)
8   }
9 )
10 val copyHook = CopyFileHook(output, workDirectory / "result.csv")
11 (pyTask hook copyHook)
12
13
```

A file variable

Location of  
the output file

Put the output file  
in the dataflow

Use a hook to extract  
the file from the dataflow

# Get the output file

## Run it and refresh.

The screenshot shows a web application interface with a dark sidebar and a light main content area. The sidebar contains a 'File' menu, a 'File name' input field, and a 'Refresh' button (circular arrow icon). Below this, there are tabs for 'Home', 'Native', and '2'. The main content area displays a file list on the left with 'Output.oms' (0,33KB) and 'result.csv' (0,03KB). A red arrow points from the 'Refresh' button to the 'result.csv' file, with the handwritten text 'Result file' next to it. Another red arrow points from the 'Refresh' button to the code editor area. The code editor shows a Scala script for running a Python task. A 'Play' button is visible next to the code. The footer of the page reads 'Loving Lobster'.

File name Refresh

Home Native 2

Output.oms 0,33KB  
result.csv 0,03KB

Result file

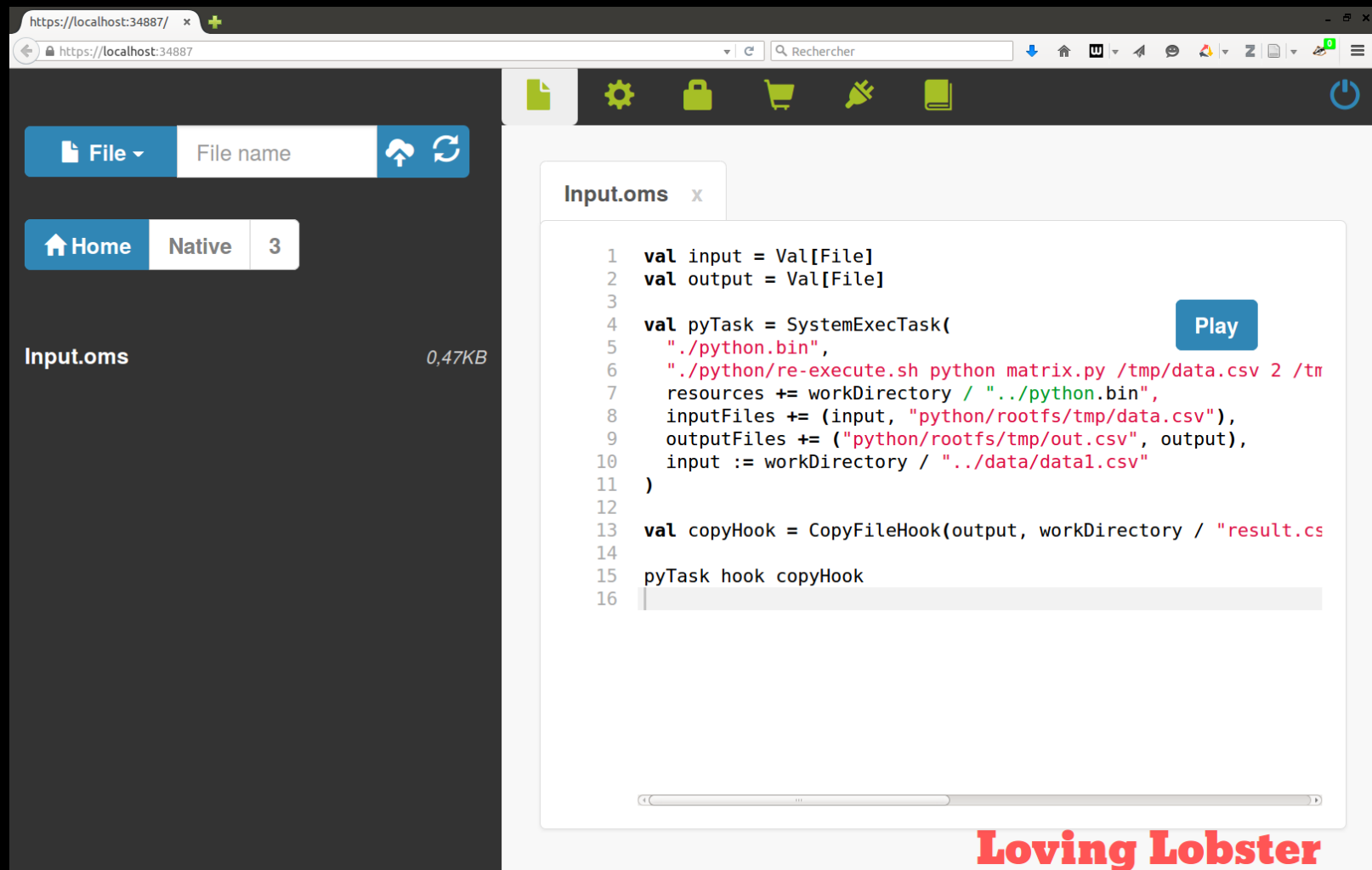
```
1 val output = val[File]
2
3 val pyTask = SystemExecTask(
4   "./python.bin",
5   "./python/re-execute.sh python matrix.py data.csv 2 /tmp/out
6   resources += workDirectory / "../python.bin",
7   outputFiles += ("python/rootfs/tmp/out.csv", output)
8 )
9
10 val copyHook = CopyFileHook(output, workDirectory / "result.cs
11
12 (pyTask hook copyHook)
13
```

Play

Loving Lobster

# Inject an input file

Go to "3" and edit Input.oms



The screenshot shows a web application interface for editing a file named `Input.oms`. The interface includes a sidebar with navigation buttons (File, Home, Native, 3) and a main editor area with a code editor and a Play button.

The code editor displays the following code:

```
1 val input = Val[File]
2 val output = Val[File]
3
4 val pyTask = SystemExecTask(
5     "./python.bin",
6     "./python/re-execute.sh python matrix.py /tmp/data.csv 2 /tn
7     resources += workDirectory / "../python.bin",
8     inputFiles += (input, "python/rootfs/tmp/data.csv"),
9     outputFiles += ("python/rootfs/tmp/out.csv", output),
10    input := workDirectory / "../data/data1.csv"
11 )
12
13 val copyHook = CopyFileHook(output, workDirectory / "result.cs
14
15 pyTask hook copyHook
16
```

The code is written in Scala and uses the `SystemExecTask` class to execute a Python script. The code defines input and output files, sets up a task, and defines a copy hook.

The interface also includes a sidebar with navigation buttons (File, Home, Native, 3) and a main editor area with a code editor and a Play button.

**Loving Lobster**

# Inject an input file

A new  
file variable

Changed input  
file location

```
1 val input = Val[File]
2 val output = Val[File]
3
4 val pyTask = SystemExecTask(
5   "./python.bin",
6   "./python/re-execute.sh python matrix.py /tmp/data.csv 2 /tmp/out.csv") set (
7   resources += workDirectory / "../python.bin",
8   inputFiles += (input, "python/rootfs/tmp/data.csv"),
9   outputFiles += ("python/rootfs/tmp/out.csv", output),
10  input := workDirectory / "../data/data1.csv"
11 )
12
13 val copyHook = CopyFileHook(output, workDirectory / "result.csv")
14
15 pyTask hook copyHook
16
```

Copy the file in the task  
execution directory

set the default value  
of the input variable

# Inject an input file

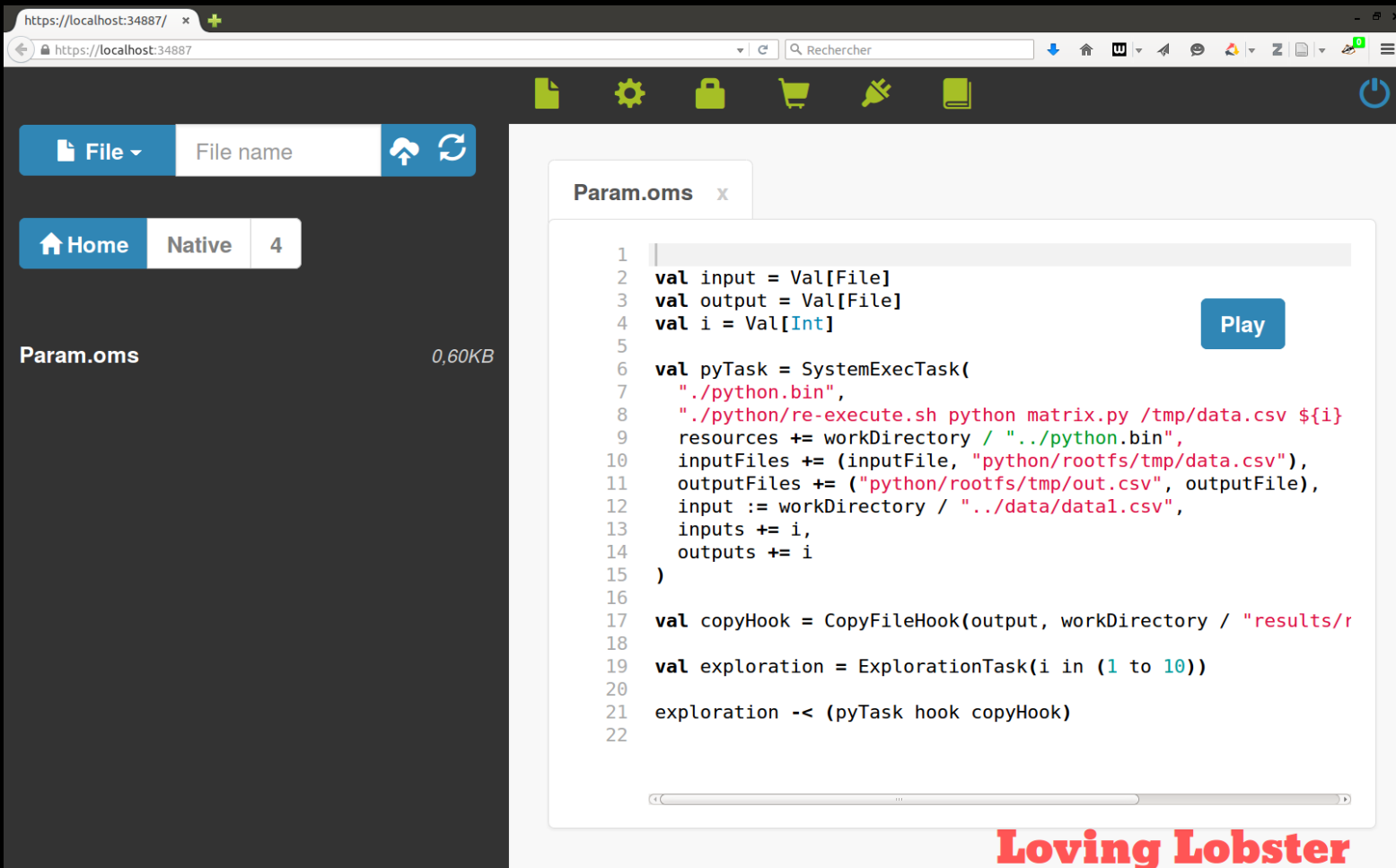
Run it!

You can change the file manually:

- data1.csv
- data2.csv
- data3.csv

# Explore a parameter

Go to "4" and edit Param.oms



The screenshot shows a web application interface for editing a parameter file. The browser address bar shows `https://localhost:34887/`. The application has a dark sidebar on the left with a "File" button, a "File name" input field, and a "Home" button. The main content area displays the file "Param.oms" with a size of "0.60KB". The file content is shown in a code editor with line numbers 1 through 22. The code is written in a functional programming style, likely F#, and includes a "Play" button for execution. The code defines variables for input, output, and a task, and then executes a loop over a range of values.

```
1  
2 val input = Val[File]  
3 val output = Val[File]  
4 val i = Val[Int]  
5  
6 val pyTask = SystemExecTask(  
7     "./python.bin",  
8     "./python/re-execute.sh python matrix.py /tmp/data.csv ${i}  
9     resources += workDirectory / "../python.bin",  
10    inputFiles += (inputFile, "python/rootfs/tmp/data.csv"),  
11    outputFiles += ("python/rootfs/tmp/out.csv", outputFile),  
12    input := workDirectory / "../data/data1.csv",  
13    inputs += i,  
14    outputs += i  
15 )  
16  
17 val copyHook = CopyFileHook(output, workDirectory / "results/r  
18  
19 val exploration = ExplorationTask(i in (1 to 10))  
20  
21 exploration -< (pyTask hook copyHook)  
22
```

**Loving Lobster**

# Explore a parameter

```
2 val input = Val[File]
3 val output = Val[File]
4 val i = Val[Int]
5
6 val pyTask = SystemExecTask(
7   "./python.bin",
8   "./python/re-execute.sh python matrix.py /tmp/data.csv ${i} /tmp/out.csv") set (
9   resources += workDirectory / "../python.bin",
10  inputFiles += (input, "python/rootfs/tmp/data.csv"),
11  outputFiles += ("python/rootfs/tmp/out.csv", output),
12  input := workDirectory / "../data/data1.csv",
13  inputs += i,
14  outputs += i
15 )
16
17 val copyHook = CopyFileHook(output, workDirectory / "results/result${i}.csv")
18
19 val exploration = ExplorationTask(i in (1 to 10))
20
21 exploration -< (pyTask hook copyHook)
22
```

A new variable

The value of "i"  
is used in the  
command line

"i" is an input and an output

The value of "i"  
is used in the hook

"exploration"  
is used in  
the workflow

"i" varies from  
1 to 10



# Explore a parameter

Run it, refresh and go to "results" folder.

The screenshot shows a web application interface. The top bar displays the URL `https://localhost:34887/` and a search bar labeled "Rechercher". Below the top bar is a navigation bar with icons for file management, settings, and other functions. The main interface is divided into two panels. The left panel is a file explorer showing a list of files in the "results" folder. The right panel is a data viewer showing the contents of "result1.csv".

**File Explorer (Left Panel):**

- File name:
- Home | ... | 4 | results
- result1.csv 0,02KB
- result10.csv 0,03KB
- result2.csv 0,02KB
- result3.csv 0,03KB
- result4.csv 0,03KB
- result5.csv 0,03KB
- result6.csv 0,03KB
- result7.csv 0,03KB
- result8.csv 0,03KB
- result9.csv 0,03KB

**Data Viewer (Right Panel):**

Param.oms x result1.csv x

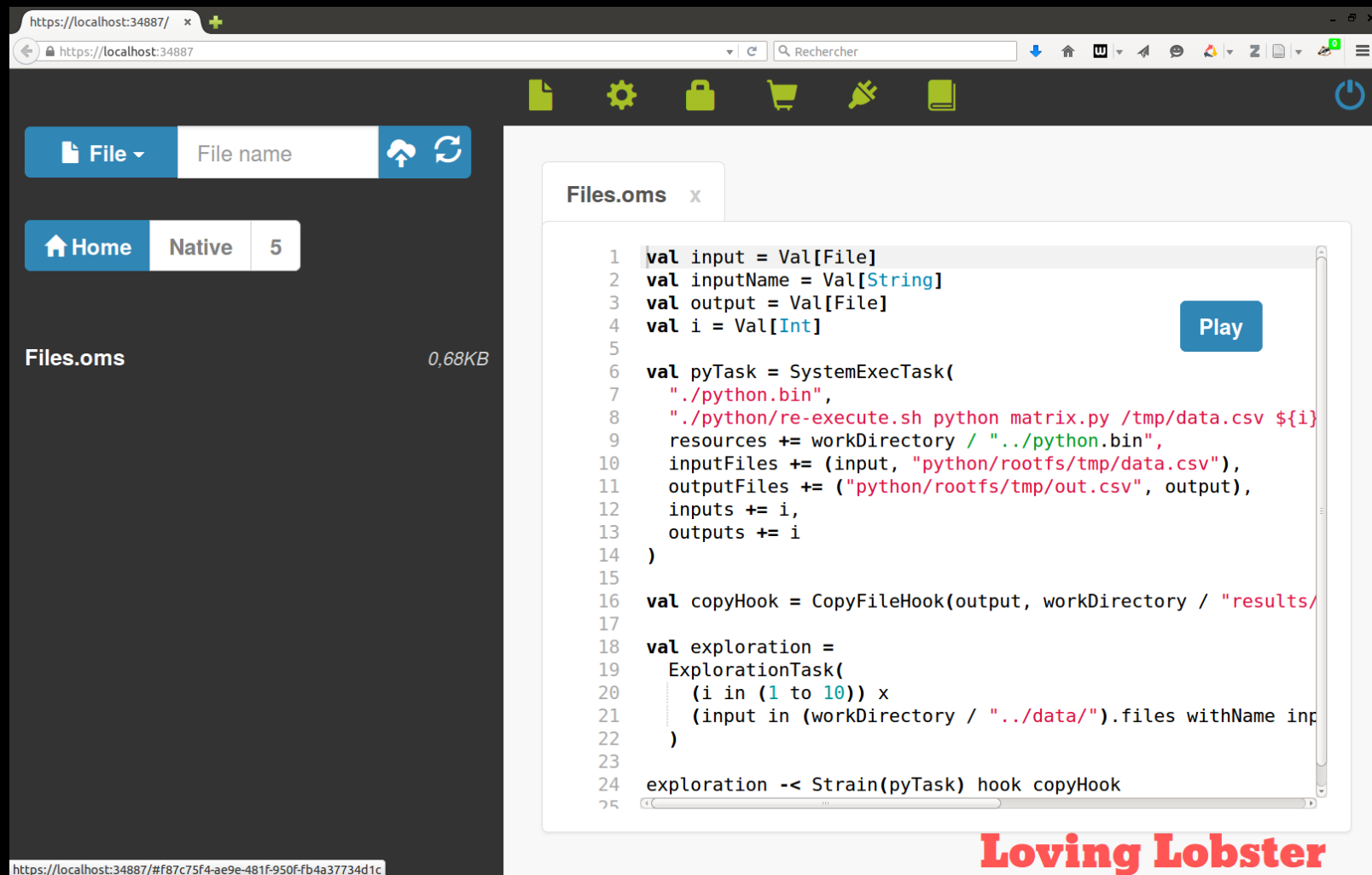
1	3	8	0
2	4	42	7
3	98	8	12
4			

**Loving Lobster**



# Explore the input file

## Go to "5" and edit Files.oms



The screenshot shows a web application interface for editing a file named `Files.oms`. The interface includes a sidebar with navigation tabs: `Home`, `Native`, and `5`. The `5` tab is selected. The main editor area displays the content of `Files.oms`, which is a Scala file. The code defines a task that runs a Python script and copies the output.

```
1 val input = Val[File]
2 val inputName = Val[String]
3 val output = Val[File]
4 val i = Val[Int]
5
6 val pyTask = SystemExecTask(
7   "./python.bin",
8   "./python/re-execute.sh python matrix.py /tmp/data.csv ${i}"
9   resources += workDirectory / "../python.bin",
10  inputFiles += (input, "python/rootfs/tmp/data.csv"),
11  outputFiles += ("python/rootfs/tmp/out.csv", output),
12  inputs += i,
13  outputs += i
14 )
15
16 val copyHook = CopyFileHook(output, workDirectory / "results/"
17
18 val exploration =
19   ExplorationTask(
20     (i in (1 to 10)) x
21     (input in (workDirectory / "../data/").files withName inp
22   )
23
24 exploration -< Strain(pyTask) hook copyHook
25
```

The interface also includes a `Play` button for running the code. The file size is indicated as `0,68KB`. The URL in the browser is `https://localhost:34887/`.

**Loving Lobster**

# Explore the input file

A variable to store  
the name of the input file

```
1 val input = Val[File]
2 val inputName = Val[String]
3 val output = Val[File]
4 val i = Val[Int]
5
6 val pyTask = SystemExecTask(
7   "./python.bin",
8   "./python/re-execute.sh python matrix.py /tmp/data.csv ${i} /tmp/out.csv" set (
9     resources += workDirectory / "../python.bin",
10    inputFiles += (input, "python/rootfs/tmp/data.csv"),
11    outputFiles += ("python/rootfs/tmp/out.csv", output),
12    inputs += i,
13    outputs += i
14  )
15
16 val copyHook = CopyFileHook(output, workDirectory / "results/${inputName.dropRight(4)}/result${i}.csv")
17
18 val exploration =
19   ExplorationTask(
20     (i in (1 to 10)) x
21     (input in (workDirectory / "../data/").files withName inputName)
22   )
23
24 exploration -< Strain(pyTask) hook copyHook
25
```

The path of the result file  
contains the name of the input file

Combine

Go through all the files of the folder  
"data", put the file in "input" and  
its name in "inputName"

All inputs are also outputs  
(they are used by the hook)

# Explore the input file

Run it and go in the "results" folder.

The screenshot shows a web application interface with a dark theme. At the top, there's a browser address bar showing 'https://localhost:34887/'. Below it is a navigation bar with icons for file, settings, folder, shopping cart, and a search bar labeled 'Rechercher'. The main interface is divided into a left sidebar and a main content area.

**Left Sidebar:**

- A 'File' dropdown menu with a 'File name' input field and a refresh icon.
- A breadcrumb navigation: 'Home' > '...' > '5' > 'results'.
- A list of files:
  - data1 (0,26KB)
  - data2 (0,27KB)
  - data3 (0,25KB)

**Main Content Area:**

A tab labeled 'Files.oms' is active. It contains a code editor with the following Scala code:

```
1 val input = Val[File]
2 val inputName = Val[String]
3 val output = Val[File]
4 val i = Val[Int]
5
6 val pyTask = SystemExecTask(
7   "./python.bin",
8   "./python/re-execute.sh python matrix.py /tmp/data.csv ${i} /tmp/out.csv") set (
9   resources += workDirectory / "../python.bin",
10  inputFiles += (input, "python/rootfs/tmp/data.csv"),
11  outputFiles += ("python/rootfs/tmp/out.csv", output),
12  inputs += i,
13  outputs += i
14 )
15
16 val copyHook = CopyFileHook(output, workDirectory / "results/${inputName.dropRight(1)}")
17
18 val exploration =
19   ExplorationTask(
20     (i in (1 to 10)) x
21     (input in (workDirectory / "../data/").files withName inputName)
22   )
23
24 exploration -< Strain(pyTask) hook copyHook
25
```

A 'Play' button is located to the right of the code editor.

**Footer:**

Loving Lobster

# Delegate to the Grid

The complex-systems propose 5000+ cores of computing for researchers.

Go to: <http://iscpif.fr/vo/> to learn how to access it.

# Delegate to the Grid

## Configure OpenMOLE

The screenshot shows the OpenMOLE web interface in a browser window. The address bar shows `https://localhost:34887/`. The interface has a dark sidebar on the left with a 'File' menu and a 'Home' button. The main area displays a code editor with Scala code. A modal dialog titled 'Authentications' is open in the center. It contains a 'Save' button at the top right, a dropdown menu for 'EGI P12 certificate', a 'Password' field with a masked input, a 'Key file' field with the value 'egi.p12', a 'Reset password' link, and a 'Close' button at the bottom right. A caution message 'Caution: all your preferences will be erased!' is displayed below the 'Reset password' link. The background code editor shows a Scala script with a 'Play' button to its right.

Authentications

Save

EGI P12 certificate

Password

Key file

egi.p12

Reset password

Caution: all your preferences will be erased!

Close

```
10 val copyHook = copyHook(output, workDirectory / results/${inputName.dropRight(1)}
11
12
13
14
15
16
17
18 val exploration =
19   ExplorationTask(
20     (i in (1 to 10)) x
21     (input in (workDirectory / "../data/").files withName inputName)
22   )
23
24 exploration -< Strain(pyTask) hook copyHook
25
```

Play

Loving Lobster

# Delegate to the Grid

Add one line to the script

```
23  
24  val env = EGIEnvironment("vo.complex-systems.eu")  
25  
26  exploration -< (Strain(pyTask) hook copyHook on env)  
27
```

# Delegate to the Grid

It runs!

The screenshot shows a web application interface with a browser window at the top. The browser has multiple tabs open, including 'User sup...', 'VOMS Adm...', 'Index of /fil...', 'OpenMOLE', 'Institut des ...', 'Institut ...', 'Zotero F...', and 'Commit...'. The address bar shows 'https://localhost:46447'. Below the browser, there is a navigation bar with icons for file, settings, lock, shopping cart, and a power button. The main content area is titled 'Executions' and features a table of task executions. The first row shows a task named 'Files.oms' with a timestamp of '30/9/2015, 17:55:19', a lightning bolt icon with the number 5, a flag icon with '6 / 31', a duration of '0:03:54', a status of 'running', and a bar chart icon labeled 'Env'. Below this row, there is a detailed view of the task, showing the environment name 'EGEnvironment@457476736', upload and download statistics, and a list of files. A 'Close' button is visible in the bottom right corner of the detailed view. The background of the detailed view shows a code editor with Scala code.

```
13  
16 val copyHook = CopyFileHook(output, workDirectory / "results/${inputName.dropRight(1)}")  
17  
18 val exploration =  
19   ExplorationTask(  
20     (i in (1 to 10)) x  
21     (input in (workDirectory / "../data/").files withName inputName)  
22   )  
23  
24 val env = EGEnvironment("vo.complex-systems.eu")  
25  
26 exploration -< (Strain(pyTask) hook copyHook on env)  
27
```

**Loving Lobster**