

Create a library : Asterics HPC

Pierre Aubert



Minimal repository :

```
https://lappweb.in2p3.fr/~paubert/ASTERICS_HPC/ressource/build/  
Correction/ExampleMinimal.tar.gz
```

Correction :

```
https://lappweb.in2p3.fr/~paubert/ASTERICS_HPC/ressource/build/  
Correction/Examples.tar.gz
```

Minimal example



**Linked
Image
Not Found**

Minimal example



**Linked
Image
Not Found**

Minimal example

AstericsHPC

asterics_hpc
macro cmake



ExampleMinimal

**Linked
Image
Not Found**

Minimal example

AstericsHPC

```
asterics_hpc  
macro cmake
```

asterics_hpc



ExampleMinimal

**Linked
Image
Not Found**

Minimal example

AstericsHPC

```
asterics_hpc  
macro cmake
```

asterics_hpc



ExampleMinimal

build

Compilation here :

```
cmake ..  
make  
make run_all  
make plot_all
```

Linked Image Not Found

Minimal example

AstericsHPC

```
asterics_hpc  
macro cmake
```

asterics_hpc

1-Hadamard



build

```
Compilation here :  
cmake ..  
make  
make run_all  
make plot_all
```

Linked Image Not Found

Minimal example

AstericsHPC

```
asterics_hpc  
macro cmake
```

asterics_hpc



ExampleMinimal

```
build  
Compilation here :  
cmake ..  
make  
make run_all  
make plot_all
```

1-Hadamard

2-Saxpy

Linked Image

Not Found

Minimal example

AstericsHPC

```
asterics_hpc  
macro cmake
```

asterics_hpc

ExampleMinimal

build

Compilation here :

```
cmake ..  
make  
make run_all  
make plot_all
```

3-Reduction

1-Hadamard

2-Saxpy

Linked Image Not Found

Minimal example

AstericsHPC

```
asterics_hpc  
macro cmake
```

asterics_hpc

ExampleMinimal



```
build  
Compilation here :  
cmake ..  
make  
make run_all  
make plot_all
```

3-Reduction

1-Hadamard

2-Saxpy

4-Barycentre

Linked Image

Not Found

Minimal example

AstericsHPC

```
asterics_hpc  
macro cmake
```

asterics_hpc

ExampleMinimal



```
build  
Compilation here :  
cmake ..  
make  
make run_all  
make plot_all
```

3-Reduction

5-Sgemm

1-Hadamard

2-Saxpy

4-Barycentre

Linked Image

Not Found

Minimal example

AstericsHPC

```
asterics_hpc  
macro cmake
```

asterics_hpc

ExampleMinimal

```
build  
Compilation here :  
cmake ..  
make  
make run_all  
make plot_all
```

3-Reduction

5-Sgemm

1-Hadamard

2-Saxpy

4-Barycentre

hadamardpython

Linked Image Not Found

Minimal example

AstericsHPC
asterics_hpc
macro cmake

astericshpc



```
build
Compilation here :
cmake ..
make
make run_all
make plot_all
```

Linked Image

1-Hadamard

2-Saxpy

3-Reduction

4-Barycentre

5-Sgemm

hadamardpython

saxpypython

Not Found

Minimal example



AstericsHPC

asterics_hpc
macro cmake

astericshpc

```
build
Compilation here :
cmake ..
make
make run_all
make plot_all
```

3-Reduction

5-Sgemm

1-Hadamard

2-Saxpy

4-Barycentre

hadamardpython

reductionpython

saxpypython

Linked Image Not Found

Minimal example

AstericsHPC
asterics_hpc
macro cmake

astericshpc



```
build
Compilation here :
cmake ..
make
make run_all
make plot_all
```

Linked

1-Hadamard

2-Saxpy

3-Reduction

4-Barycentre

5-Sgemm

hadamardpython

reductionpython

saxpypython

barycentrepthon

Not Found

Minimal example



AstericsHPC

asterics_hpc
macro cmake

astericshpc

```
build
Compilation here :
cmake ..
make
make run_all
make plot_all
```

3-Reduction

5-Sgemm

1-Hadamard

2-Saxpy

4-Barycentre

hadamardpython

reductionpython

sgemmpython

saxpypython

barycentrepthon

Linked Image Not Found

Aims of Asterics HPC library

- ▶ To provide :
 - ▶ The **rdtsc** function (to time functions)
 - ▶ The aligned allocation/deallocation functions (needed for optimisation)
 - ▶ Table
 - ▶ Matrix
 - ▶ Some **CMake** macros to run and plot all the results automatically
 - ▶ **runExample(target)** and **runPythonExample(target dependency)** :
To run executables with **make run_all**
 - ▶ **plotPerf("plotName" target1 target2 ...)** :
To plot and compare results from different targets with **make plot_all**
 - ▶ The results are created in **build/Examples/Performances**
- ▶ C++ library : **asterics_hpc**
- ▶ Python module : **astericshpc**

This will simplify all the following examples.

Ask the CPU the number of cycles since the program's beginning

64 bits version :

```
extern long unsigned int rdtsc(void) {  
    >>     long unsigned int a, d;  
    >>     __asm__ volatile ("rdtsc" : "=a" (a), "=d" (d));  
    >>     return (d<<32) | a;  
}
```