Create simple examples of code to launch several jobs that will run in parallel

Measure computing power

Using Matlab compute a matrix multiply and measure execution time.

You me use the following code:

```
sz=1000; %8 MB per matrix
v1=rand(sz,sz);
v2=rand(sz,sz);

fprintf('Begin loop...');
tic
for i=1:100
    res=v1*v2;
end
t=toc;

%Computing times
ta=t/100; %time per vector add
fprintf('Vector size: %f\n',sz);
fprintf('Time per vector add: %f s\n',ta);
fprintf('Performance: %f MFLOPS\n',2*sz*sz*sz/ta/1e6);
```

If this code is executed in a multi-core system check if Matlab is using all your cores. Other functions are also already implemented to use all cores (FFT, fmins with option UseParallel, sort, etc.)

Starting Matlab with the following command disables multithreading:

```
matlab -singleCompThread
```

C with MPI

Download and install MPI from: http://www.mcs.anl.gov/research/projects/mpich2/

There is a demo called cpi.exe to test running C programs in parallel with MPI

- Execute bin/mpiregister and provide login and password
- Execute bin/mpiexec
  - Set number of processes
  - Load examples/cpi.exe

Use Task Manager to monitor the use of CPU cores, and the number of tasks.

Advanced users:

To create a new applications with Visual Studio 2008

- Create a project type: C++, Win32, console
- Configure libraries in Project/properties/
  - Configuration: c/c++, general:
  - add folder C:\Archivos de programa\MPICH2\include
  - Configuration: link, general:
  - add folder C:\Archivos de programa\MPICH2\lib
  - Configuration: link, inputs:
  - add library: mpi.lib

Matlab with pMatlab

Download pMatlab from Lincoln Lab

unzip pMatlab

cd pMatlab

startup.m (or at least execute addpath .\MatMPI)

cd examples

cd mandelbrot

run pMandelbrot:

```
eval(pRun('pMandelbrot',1{}));  %1 core
eval(pRun('pMandelbrot',4{}));  %4 cores
```

The outputs of all children processes are called MatMPI/*.out

Matlab with Parallel Computing Toolbox

(This toolbox is not available in the lab)

```
matlabpool(4) %starts 4 cores (5 Matlab processes can be seen)

sz=2000000; %16MB vector
v=rand(sz,2);
res=zeros(sz,2); %for memory allocation

tic
parfor i=1:2
      res(:,i)=sort(v(:,i));  %each iteration independent of the previous
end
toc

matlabpool close  %back to normal (when you are done)
```

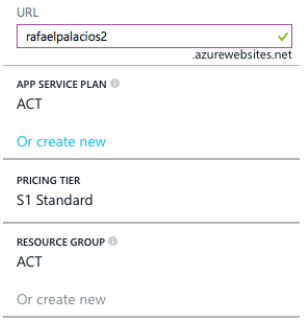
Using Microsoft Azure

1-Sign up in www.microsoftazurepass.com using your access codes. You get an Azure pass valid for 180 days.

- For this step you need your comillas.edu account linked to live.com or you will need to create a new account.
- Don't ask me why Microsoft is not able to link your Azure pass code to your comillas.edu account. It may work for some people.

2-Access to `portal.azure.com` and create a Web App.

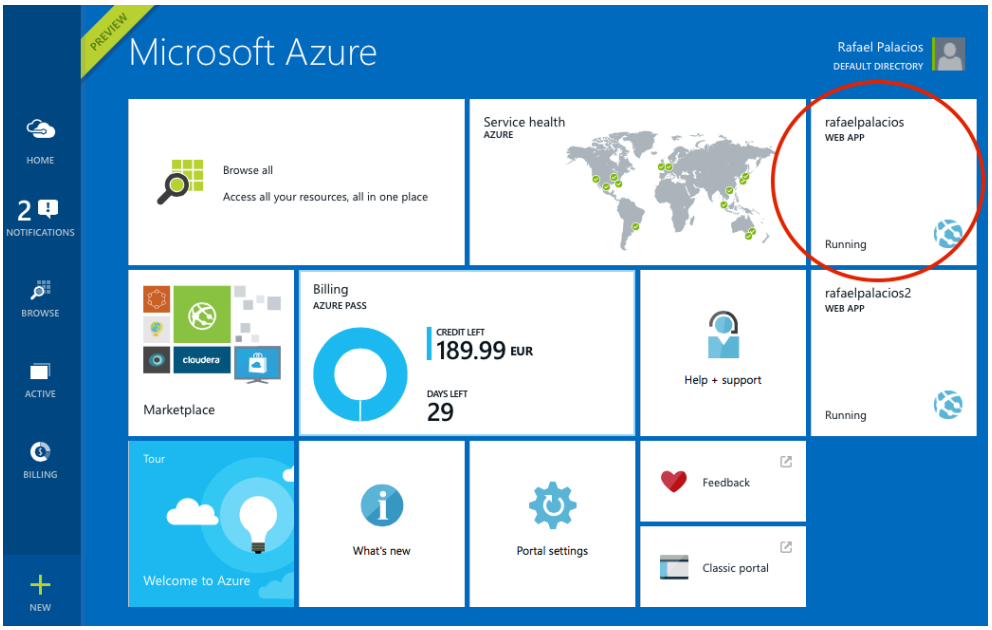
- When you access portal.azure.com make sure that it is using the account where the azure pass code was linked.
- You may need to logout from your account (azure tends to log you in using your comillas.edu account) then log in again with the right account.
- Hit the + icon to create a new application. Select Web+Mobile → Web App
- You will be asked for the URL, you may use your name. Than will be the URL of you website.



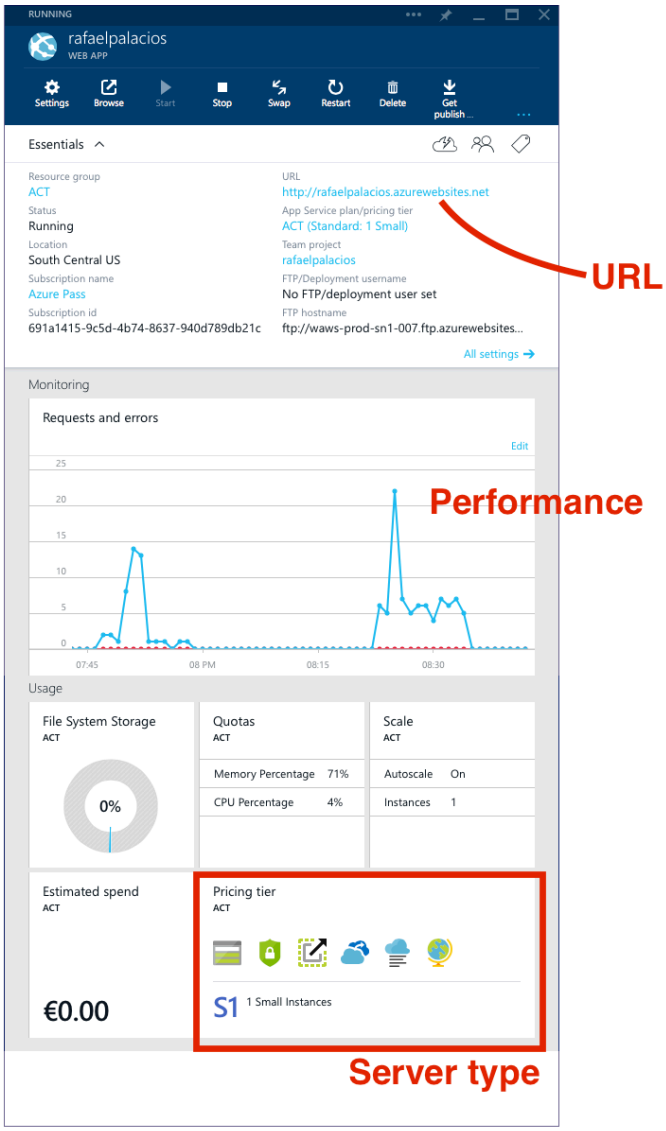
- If you are asked about the payment mode, you must select **Azure PASS.** If such options doesn't appear is because you are using an account not linked to you Azure Code.

2-Test your application

- Now that your applications has been created, it will appear on the mail screen of portal.azure.com

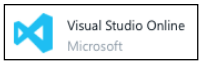


- When you click on the Application you get all the information related to it:



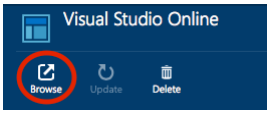
3-Edit your application

- The new Azure interface portal.azure.com doesn't include Monaco web-based editor by default.
- To install Monaco Visual Studio Online, select the application, then settings 

→ Extensions
- Hit Add, then select Visual Studio Online: 

- Now that the extension has been installed, it is easy to add and edit files in the server.
- Go to you App, then settings 

→ Extensions → Visual Studio Online →Browse



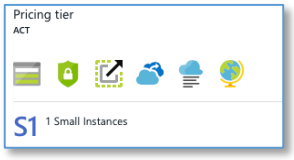
- You get a new browser tab where you can create new files or folders, and where you can edit any file.
- Reload your App URL to test the changes.

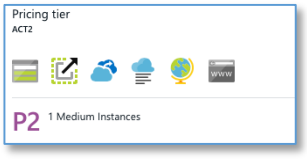
4-Measure performace

- Create a php file, for example program.php, with the following code:

```
<?php
print("begining of program:");
print("<br>");
$t=time();
for ($i=0; $i<10; $i++) {
        for ($j=0; $j<1000000; $j++) {
                $x=pow(234.55,1.0123456);
        }
}
print("end of program: ");
print("<br>");
$elapsed=time()-$t;
print("Time: ".$elapsed." s<br>");
?>
```

- To execute the program use the web browser to access your URL followed by /program.php
- After a few seconds you will get on the screen a message with the elapsed time
- In theory it will be possible to change the server architecture, for example from the default S1 to a more advanced P2:





- As of April/2015 we are getting error while trying to change architecture, however it is possible to create a new Web App defining a more advanced architecture. That way you can have two Web Apps running at the same time and you can compare results
- Using a 2 core architecture, execution time does not change if reloading two browser tabs at the same time, since each process will be executed in a different core. On the other hand, in a single core server the execution time is duplicated if we reload the same page from two browser tabs at the same time.