

Ejecutar planificador

Windows

Máster en Inteligencia Artificial

Razonamiento y Planificación Automática

Dr. Cesar Augusto Guzman Alvarez

David Alejandro Espinosa Guaita

Singularity

Nos basamos en los pasos que se mencionan en la siguiente URL:

<https://sylabs.io/guides/3.5/admin-guide/installation.html#windows>

Instalamos los siguientes programas:

- [Git for Windows](#)
- [VirtualBox for Windows](#)
- [Vagrant for Windows](#)

Ejecutamos Git Bash (Windows) y creamos un directorio donde vamos a ejecutar vargrant:

```
$ mkdir vm-singularity && cd vm-singularity
```

Ejecutamos los siguientes comandos para ejecutar vagrant con una máquina de ubuntu.
(Substitute a different value for the \$VM variable if you like)

```
$ export VM=sylabs/singularity-3.5-ubuntu-bionic64 && vagrant init $VM && vagrant up && vagrant ssh
```

```

$ export VM=sylabs/singularity-3.5-ubuntu-bionic64 && vagrant init $VM && vagrant up && vagrant ssh
A 'Vagrantfile' has been placed in this directory. You are now
ready to 'vagrant up' your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
'vagrantup.com' for more information on using Vagrant.
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'sylabs/singularity-3.5-ubuntu-bionic64' could not be found. Attempting to find and install...
    default: Box Provider: virtualbox
    default: Box Version: >= 0
==> default: Loading metadata for box 'sylabs/singularity-3.5-ubuntu-bionic64'
    default: URL: https://vagrantcloud.com/sylabs/singularity-3.5-ubuntu-bionic64
==> default: Adding box 'sylabs/singularity-3.5-ubuntu-bionic64' (v20191206.0.0) for provider: virtualbox
    default: Downloading: https://vagrantcloud.com/sylabs/boxes/singularity-3.5-ubuntu-bionic64/versions/20191206.0.0/providers/virtualbox.box

```

Hasta aquí lo que tenemos es una máquina virtual con Ubuntu en la cual está configurado el Singularity:

```

vagrant@vagrant:~$ s1
singularity size
vagrant@vagrant:~$ singularity --version
singularity version 3.5.1
vagrant@vagrant:~$ |

```

Compilar Planificador

Vamos a probar el planificador “DecStar”. El código se encuentra en el siguiente enlace:

Code: <https://bitbucket.org/ipc2018-classical/team2/src/ipc-2018-seq-opt/>

Descargamos el fichero Singularity

\$ wget <https://bitbucket.org/ipc2018-classical/team2/raw/ipc-2018-seq-opt/Singularity>

Nota: En el caso de probar otro planificador se debe reemplazar del enlace lo rojo y lo azul, respectivamente.

Creamos la imagen del planificador (planner.img):

\$ sudo singularity build planner.img Singularity

Tened paciencia que esto tarda lo suyo.

Creamos el directorio rundir:

\$ mkdir rundir

Descargamos los ficheros PDDL de ejemplo. Serán dos ficheros, un dominio (donde se encuentra toda la definición de acciones que puede ejecutar el agente) y problema (donde se encuentra el estado inicial y final del problema).

\$ wget <https://raw.githubusercontent.com/cguz/RPA/main/pddl/MARs/domain.pddl>

```
$ wget https://raw.githubusercontent.com/cguz/RPA/main/pddl/MARs/problem.pddl
```

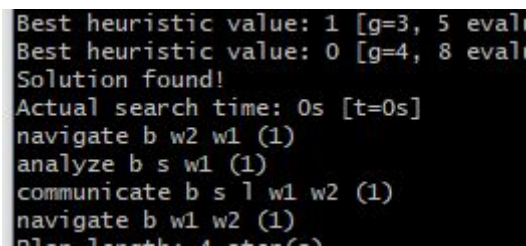
Movemos los ficheros domain.pddl y problem.pddl al directorio rundir:

```
$ mv domain.pddl rundir  
$ mv problem.pddl rundir
```

Ejecutamos el planificador (planner.img) con singularity:

```
$ RUNDIR="$(pwd)/rundir"  
$ DOMAIN="$RUNDIR/domain.pddl"  
$ PROBLEM="$RUNDIR/problem.pddl"  
$ PLANFILE="$RUNDIR/sas_plan"  
$ ulimit -t 1800  
$ ulimit -v 8388608  
$ singularity run -C -H $RUNDIR planner.img $DOMAIN $PROBLEM $PLANFILE
```

El planificador se ejecuta correctamente encontrando un plan solución:



```
Best heuristic value: 1 [g=3, 5 evals]  
Best heuristic value: 0 [g=4, 8 evals]  
Solution found!  
Actual search time: 0s [t=0s]  
navigate b w2 w1 (1)  
analyze b s w1 (1)  
communicate b s l w1 w2 (1)  
navigate b w1 w2 (1)  
plan length 4 steps (1)
```

En cualquier momento se puede salir de ubuntu:

```
$ exit
```

Y entrar nuevamente a través de vagrant con el comando:

```
$ vagrant ssh
```