plan4res installation

version 2.0 Feb 14, 2025

It is recommended to install plan4res with the full environment (p4r-env), which is embedded in a singularity container. This requires at least 3Gb. In case you do not have this available, you may install plan4res without the environment, but it requires installing all the dependencies first, which may sometimes be quite tricky.

Plan4res requires use of external solver, which can be CPLEX, GUROBI, SCIP or HiGHS. If you do have a CPLEX or GUROBI license, it is recommended to use one of these software. If not, it is recommended to use HiGHS.

If you wish to use CPLEX, you must have a CPLEX linux installer available, such as cplex_studio2211.linux_x86_64.bin, referenced as *cplex_studioXXXXXX.bin* below.

If you wish to use GUROBI, you must have a GUROBI linux installer, sur as gurobi11.0.1_linux64.tar.gz and a gurobi license file: gurobi.lic.

Before starting the install, create an empty directory (for example *P4R_DIR)* where you want to install plan4res. It should not contain special characters or whitespaces!

1 Installation of the full plan4res environment, for all operating systems

Requirements: at least 3G available.

1.1 Installing on Linux:

Move to your install directory: cd P4R_DIR, and type the following commands:

```
cd P4R_DIR

git clone https://github.com/plan4res/install

mv ./install/* . && rm -rf install

chmod a+x *.sh

./plan4res_install.sh [-S <SOLVER>] [-I <installer>] [-L <license>] [-v
<version>] [-M <mpi>]
```

Where:

- SOLVER is the chosen solver among CPLEX, GUROBI, SCIP, HiGHS. If this option is not provided, HiGHS is chosen
- Installer is the installer file for CPLEX of Gurobi, the -I option must only be provided when CPLEX or GUROBI are chosen
- License is the license file, only for GUROBI
- Version is the solver version, only for SCIP
- Mpi is the chosen MPI version among MPICH and OpenMPI. If this option is not provided, OpenMPI is chosen.

Examples:

./plan4res_install.sh

- Installs plan4res with HiGHS

./plan4res_install.sh -S CPLEX -I cplex_studio2211.linux_x86_64.bin

- Installs plan4res with CPLEX, using ths installer cplex_studio2211.linux_x86_64.bin available in P4R_DIR

./plan4res_install.sh -S GUROBI -I gurobi11.0.1_linux64.tar.gz -L gurobi.lic

- Installs plan4res with GUROBI using the gurobi installer and licence available in P4R_DIR

./plan4res_install.sh -S SCIP -v 9.1.1

- Installs plan4res with SCIP, choses the version 9.1.1 from SCIP

It is possible to configure a specific location where the user wishes to run plan4res. This also allows to install the software once on e.g. a shared directory in an organization, and run it on different user's sessions.
For doing so, copy the script user_init_plan4res.sh which is present in P4R_DIR to the location where the user wants to run plan4res, e.g. P4R_DIR_LOCAL and run the command:

```
./user_init_plan4res -D P4R_DIR -S SOLVER
```

Where:

- P4R_DIR is the directory where plan4res has been previously installed
- SOLVER is the chosen solver among CPLEX, GUROBI, SCIP, HiGHS. If this option is not provided, HiGHS is chosen (this allows to configure the settings files in the example of dataset according to the choice of the solver)

1.2 Installing on Windows, using WLS

Follow the procedure 'Installing on Linux'

1.3 Installing on Windows, using Vagrant

The procedure is available at **https://gitlab.com/cerl/plan4res/p4r-env#windows**. It is reproduced below with some more information.

Installation requires Windows 7 Pro 64bit SP1 or higher and PowerShell 3.0 or higher. Furthermore, the CPU must support hardware virtualization. On many systems, the hardware virtualization features first need to be enabled in the BIOS.

1.3.1 Required packages Installation (execute once)

- Install Git for Windows (use default settings) https://git-for-windows.github.io/
- Install VirtualBox and Extension Pack https://www.virtualbox.org/wiki/Downloads
- Install Vagrant https://www.vagrantup.com/downloads.html
- (Optional) Install Vagrant Manager http://vagrantmanager.com/downloads/

The goal of Vagrant and VirtualBox is to emulate a UNIX system on the Windows computer. Vagrant Manager provides a GUI to facilitate the management of the VM.

1.3.2 Installation

Run Git Bash.

If working behind a proxy, define the following environment variables:

```
export http_proxy = <proxy address>:<port>

export https_proxy = ${http_proxy}
```

Move to your installation directory:

```
cd P4R_DIR
```

Then enter the following commands:

```
git clone https://github.com/plan4res/install

mv ./install/* .

chmod a+x *.sh

./plan4res_install.sh [-S <SOLVER>] [-I <installer>] [-L <license>] [-v
<version>] -M MPICH -V <memory>
```

Where:

- SOLVER is the chosen solver among CPLEX, GUROBI, SCIP, HiGHS. If this option is not provided, HiGHS is chosen
- Installer is the installer file for CPLEX of Gurobi, the -I option must only be provided when CPLEX or GUROBI are chosen
- License is the license file, only for GUROBI
- Version is the solver version, only for SCIP

Examples:

./plan4res_install.sh -M MPICH -V 8192

- Installs plan4res with HiGHS

./plan4res_install.sh -M MPICH -V 8192 -S CPLEX -I cplex_studio2211.linux_x86_64.bin

- Installs plan4res with CPLEX, using ths installer cplex_studio2211.linux_x86_64.bin available in P4R_DIR

./plan4res_install.sh -M MPICH -V 8192 -S GUROBI -I gurobi11.0.1_linux64.tar.gz -L gurobi.lic

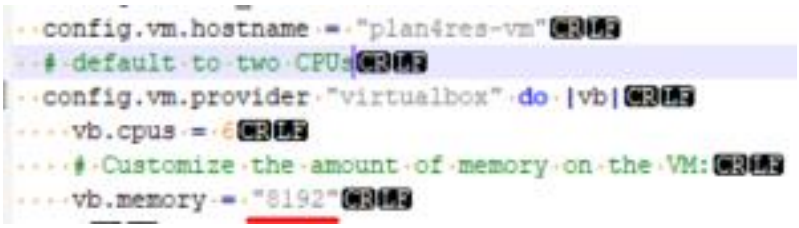
- Installs plan4res with GUROBI using the gurobi installer and licence available in P4R_DIR

./plan4res_install.sh -M MPICH -V 8192 -S SCIP -v 9.1.1

- Installs plan4res with SCIP, choses the version 9.1.1 from SCIP

1.3.3 Vagrant settings

To avoid issues with memory fragmentation while using Vagrant, we advise to allocate at least 8 Gb of RAM if the computer can afford it. This can be done via the -V option of the installation command (see above), but you can also edit the file *Vagrantfile* located in *P4R_DIR/p4r-env*, edit parameter `vb.memory` (in Mb):



You may also increase the number of CPUs allocated to the VM if possible, on your machine. In above example, it is set to 6.

1.4 Updating

1.4.1 Update p4r-env

```
./plan4res_install.sh -U p4r-env
```

1.4.2 Update StOpt

```
./plan4res_install.sh -U stopt
```

1.4.3 Update SMS++

```
./plan4res_install.sh -U sms++
```

1.4.4 Change the solver

```
./plan4res_install.sh -S New_SOLVER [ -I installer -L licence -v version]
```
(installer to be provided in New_SOLVER is CPLEX or GUROBI, licence to be provided if New_SOLVER is GUROBI, version to be provided in New_SOLVER is SCIP)

1.4.5 Update the solver

```
./plan4res_install.sh -S SOLVER -U SOLVER [ -I installer -L licence -v version]
```
(installer to be provided in New_SOLVER is CPLEX or GUROBI, licence to be provided if New_SOLVER is GUROBI, version to be provided in New_SOLVER is SCIP)

1.4.6 Update plan4res scripts and documentation

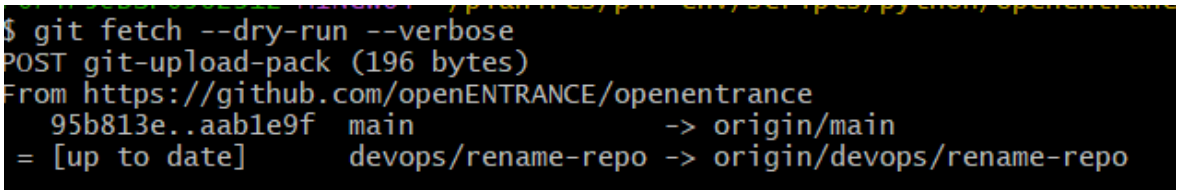
The following directories can be updated:

- `P4R_DIR/documentation`
- `P4R_DIR/p4r-env/scripts/python/openentrance`: to update the openENTRANCE nomenclature definition.
- `P4R_DIR/p4r-env/scripts/python/plan4res-scripts`: to update the data processing and visualization scripts
- `P4R_DIR/p4r-env/scripts/include`: to update the launching scripts

To check if new versions are available, run the following commands:

1. `cd P4R_DIR/p4r-env/scripts/python/openentrance`
2. `git fetch --dry-run --verbose`

If the following output is displayed, it means your installation is up to date:



Otherwise, you can perform the update using:

3. `git pull`

Do the same with the repositories `p4r-env/scripts/python/plan4res-scripts` and `p4r-env/scripts/include` if necessary.

1.4.7 Update example of dataset

When installing plan4res, an example of dataset is created in P4R_DIR/data/toyDataset

As plan4res may have been ran in this dataset, it is not recommended to update it but to download the last version of this toyDataset:

- From P4R_DIR/p4r-env/data, change the name of the dataset (mv toyDataset toyDataset_save)
- Download the new version of the dataset:

```
git clone https://github.com/plan4res/toyDataset
```

2 Installation without p4r-env

It is possible to install plan4res without the p4r-env environment. This requires installing all the dependencies first. It is possible to install on Linux (Ubuntu or Debian), MacOS or Windows

2.1 Install the dependencies

2.1.1 Python:

Python3 must be the default version of python

The following python packages must be installed:

setuptools, openpyxl, scipy, pandas, pyyaml, matplotlib, geopandas, fiona, descartes, shapely, pillow, mapclassify, numpy, requests, requests-toolbelt, rtree, pygeos, wheel, netCDF4, pyam-iamc

2.1.2 SMS++ dependencies

The requirements for installing SMS++ are described in the SMS++ gitlab repo[1]. Their installation may be included in the installation script of SMS++, but, in particular in the case of a Linux installation, the version which would be installed depend on which linux OS is installed (and which version), and thus may be too old. (This in particular happens on linux Debian::bullseye). In that case you should install manually a more recent version.

For each SOFTWARE install, on linux, PATH and LD_LIBRARY_PATH must be updated to include $SOFTWARE_PATH/mpi and $SOFTWARE_PATH/mpi/lib. For BOOST, the variable $BOOST_PATH must also be set.

- **Boost[2] (minimum version 1.87).**
- **OpenMPI[3] (minimum version 3.1.4) OR MPICH[4] (minimum version 3.3.2)**
- **NETCDF[5] (minimum version 4.9.2) and NETCDFCXX (minimum version: 4.3.1).**
- **Eigen[6] (minimum version 3.3.7)**
- **Cmake[7] (minimum version 3.28.1)**

2.2 Install plan4res

```
cd P4R_DIR

git clone https://github.com/plan4res/install
```

[1] https://gitlab.com/smspp/smspp-project/-/wikis/Installing-SMS++#requirements
[2] https://www.boost.org/more/getting_started/unix-variants.html
[3] Open MPI: Open Source High Performance Computing
[4] MPICH | High-Performance Portable MPI
[5] netCDF Downloads
[6] libeigen / eigen · GitLab
[7] Kitware/CMake: Mirror of CMake upstream repository

```
chmod a+x *.sh
```

```
./plan4res_install.sh -X -S <SOLVER> [-D <solverpath>] [-I <installer>]
[-L <license>] [-v <version>] [-M <mpi>]
```

Where:

- SOLVER is the chosen solver among CPLEX, GUROBI, SCIP, HiGHS. If this option is not provided, HiGHS is chosen
- `solverpath` is to be provided if the solver is already installed in your system. In that case you need to provide the path where it is installed
    - for CPLEX: solverpath is where you find e.g. cplex/bin and cplex/lib
    - for GUROBI: solverpath is where you find e.g. gurobi1101/linux64 and gurobi1101/gurobi.lic is version 11.01 is installed
    - for HiGHS: solverpath is where you find bin, include and lib
- Installer is the installer file for CPLEX of Gurobi, the -I option must only be provided when CPLEX or GUROBI are chosen
- License is the license file, only for GUROBI
- Version is the solver version, only for SCIP
- mpi is the chosen MPI version among MPICH and OpenMPI. If this option is not provided, OpenMPI is chosen.=