

Summer Student Programme Report

Evaluation of HPC simulation tools for efficient and cost-effective resource provisioning

MIGUEL PÉREZ DE LA RUBIA
migperezdlrubia@gmail.com

SUPERVISOR:

PABLO LLOPIS SANMILLAN

COLLABORATOR:

ALBERTO NÚÑEZ COVARRUBIAS

Start: 21-06-2021, End: 20-08-2021

Table of context

Introduction

Development

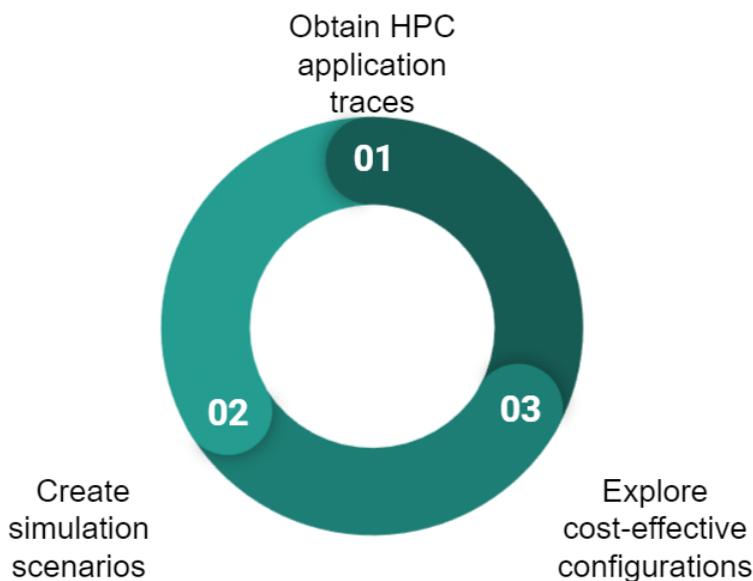
Future Work

Special Thanks

1. Introduction

At the beginning of my internship, I was introduced to a few tools used at CERN. These tools include the HPCBatch cluster, where I was going to develop the project.

The objective of the project is to study a new methodology of evaluating HPC clusters through simulation tools. Using simulation tools enable the possibility of knowing the cost-effectiveness and efficiency with precise results in advance. This methodology has the workflow shown below:



1. **Obtain HPC application traces:** by instrumenting applications using standard tools such as Score-P or Tau, we can generate detailed traces for each application. This way we can create a repository of HPC application traces. These can be generated by experts in the field.
2. **Create simulation scenarios:** by changing the component parameters, one can model faster CPUs or faster memory or storage according to the characteristics offered by vendors.
3. **Explore cost-effective configurations:** leveraging the application traces, we can use them as inputs for the simulator to evaluate different hypothetical scenarios. For instance, one where we invest more in a better network or a situation where we invest more in faster memory, or CPUs. Since the simulator has a graphical interface, this step (and the previous step) can be carried out by service managers, but also anybody in charge of resource procurement.

Due to their expensiveness, knowing where the bottlenecks can be will save a lot of effort trying to optimize applications. Also, it can be useful by testing which applications are better for the existing HPC clusters, with the advantage of not having to saturate the cluster, delaying other possible results executing at that moment. This is applied either way to cloud environments and bare-metal.

2. Development

For the development of the project we have followed the next goals:

- *Evaluate simulation tools*: we evaluate simulation tools that could fit this project's requirements. We found different simulators each of them with its own specifications:
 - COTSon¹, a simulator framework jointly developed by HP Labs and AMD that provides accurate evaluations of current and future computing.
 - M5², which supports the execution of the entire system, including operating system code and models of network and disk devices.
 - Simics³, another full-system simulator that was one of the first academic projects in this area and the first commercial full-system simulator.
 - SIMCAN⁴ is a simulation platform for modeling and simulating both distributed systems and applications.

We agreed to use SIMCAN, as it provides high level of flexibility and scalability. This is given by the simulation core of SIMCAN relies on its repository, which contains a compilation of models that represents the most relevant components of a distributed system, such as CPUs, disks and communication networks. Thanks to this collection of models the same component can be represented by different models, e.g. a CPU processor can be modeled as a Single-Core CPU model and as a Quad-Core CPU model. In addition of this, we also had the opportunity to collaborate with Alberto Nuñez, developer of the simulator, who will help with the installation in the cluster.

- *Choose target HPC applications*: we choose a sample of HPC applications that we want to target for our evaluation and methodology. We chose applications from the Accelerator and Technology Sector and Lattice QCD. These applications are run on CERN's HPC infrastructure.
- *Setup simulation environment at CERN's infrastructure*: this phase includes building the simulation tools, and testing to find a version that works and is suitable for the project. This is a time consuming step due to the complexity of the software stack.
- *Tracing: Instrument HPC applications*: this step involved researching tools such as P-Score, OTF2 and Tau to obtain detailed traces from running HPC applications.
- *Develop trace reader support*: we have to develop code that adds support for using our traces as inputs for the simulator. Our chosen simulator had limited support for these application traces. This is also a time-consuming process due to the complexities of the trace file formats.

Due to external circumstances, the project's time was cut in half, and we could not finish the goal: "Develop trace reader support". Therefore this project is still a work in progress, and we have not arrived at a final conclusion yet.

¹ <http://cotson.sourceforge.net/>

² <https://ieeexplore.ieee.org/document/1677503>

³ <https://www.windriver.com/products/simics>

⁴ <https://www.sciencedirect.com/science/article/pii/S1569190X11001432?pes=vor>

2.1. Installing simulation tools

For the installation of SIMCAN, a step-by-step guide has been developed. SIMCAN is based on OMNeT++⁵, an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. There is also another dependence with INET++⁶, an open-source OMNeT++ model suite for wired, wireless and mobile networks. INET evolves via feedback and contributions from the user community.

2.1.1. OMNeT++ installation

1. Download omnet and untar

```
wget
https://github.com/omnetpp/omnetpp/releases/download/omnetpp-6.0pre11/omnetpp-6.0pre11-src-linux.tgz
$ tar xvfz omnetpp-6.0pre11-src-linux.tgz
```

2. Install python dependencies

```
python3 -m pip install --user --upgrade numpy pandas matplotlib
scipy seaborn posix_ipc
```

3. Add omnet to the PATH

3.1. Either permanently:

```
$ nano ~/.bashrc
$ export PATH=$HOME/omnetpp-6.0pre11/bin:$PATH
```

3.2. Or just with

```
$ source setenv
```

4. Omnet6 needs to be able to compile c++ >= 14, which means gcc >= 9

```
scl --list # show available "software collections"
scl enable devtoolset-9 bash
```

5. Load MPI module

```
module load mpi/openmpi/3.0.0
```

6. Set configure flags WITH_QTENV, WITH_OSG, and WITH_OSGEARTH by editing the file 'configure.user' in the omnet directory.

For the cluster, use these settings:

```
WITH_QTENV=no
WITH_OSG=no
```

⁵ <https://omnetpp.org/>

⁶ <https://inet.omnetpp.org/>

```
WITH_OSGEARTH=no
```

For building with a graphical interface (Eclipse), use:

```
WITH_QTENV=yes  
WITH_OSG=no  
WITH_OSGEARTH=no
```

7. Now just run configure

```
./configure  
make
```

2.1.2. Inet installation

Download version 4.3.2:

```
wget  
https://github.com/inet-framework/inet/releases/download/v4.3.2/i  
net-4.3.2-src.tgz  
tar xf inet-4.3.2-src.tgz  
cd inet4.3
```

Build Inet:

```
which omnetpp || echo "Setup Omnet++ environment first" # make  
sure omnetpp is in the path  
source setenv  
make makefiles  
make -j $(nproc) MODE=debug
```

2.1.3. Simcan installation

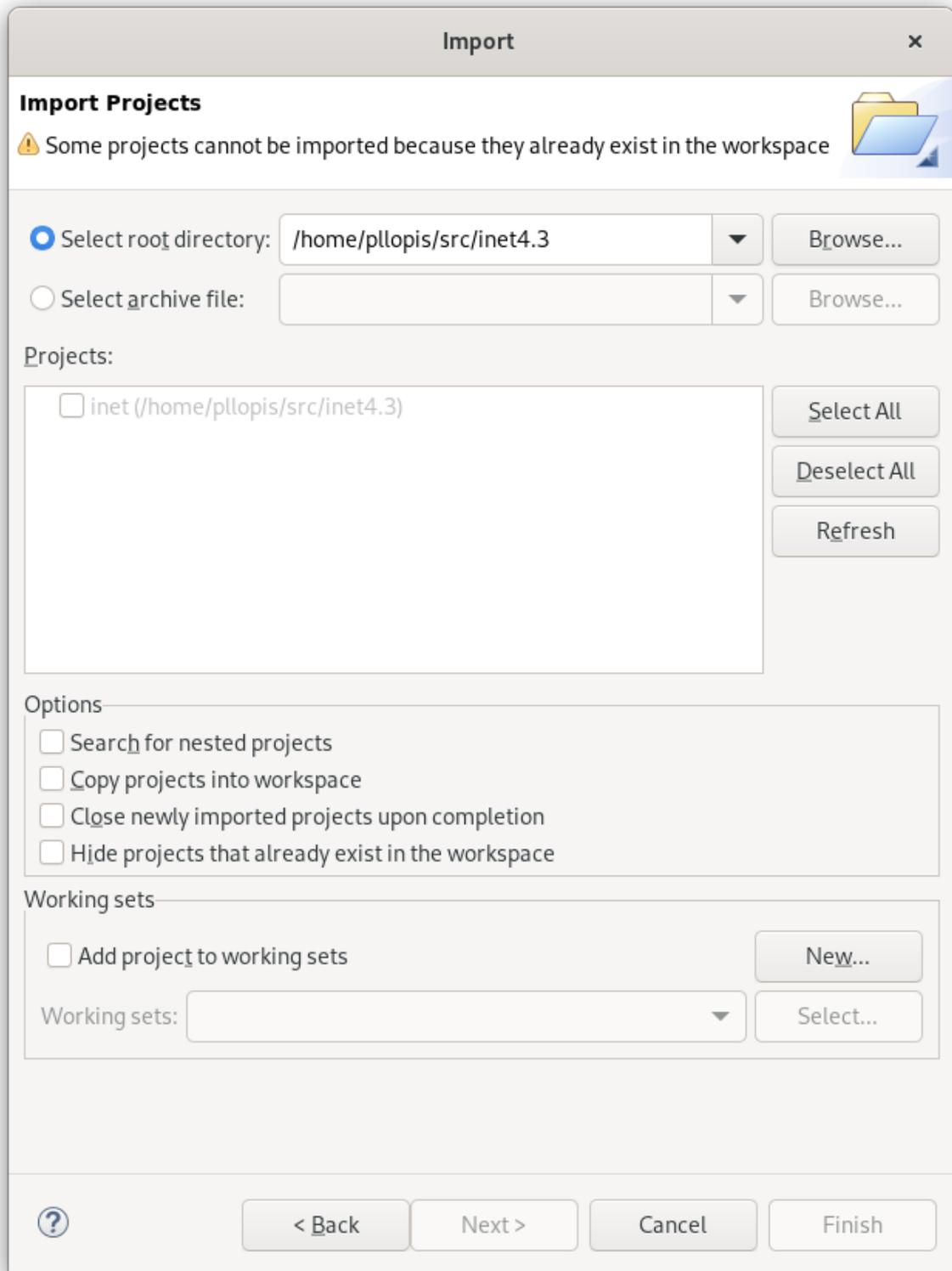
SIMCAN is easily installed in two steps: first, through a graphical environment; second, in the cluster.

Installing through the graphical OMNeT++ Eclipse interface makes sure that the makefiles are automatically generated. Otherwise, the makefiles would have to be built manually (involves setting lots of flags and arguments).

Then, the project can be copied over into the cluster, where a simple ``make clean && make`` will recompile it using the already generated makefiles.

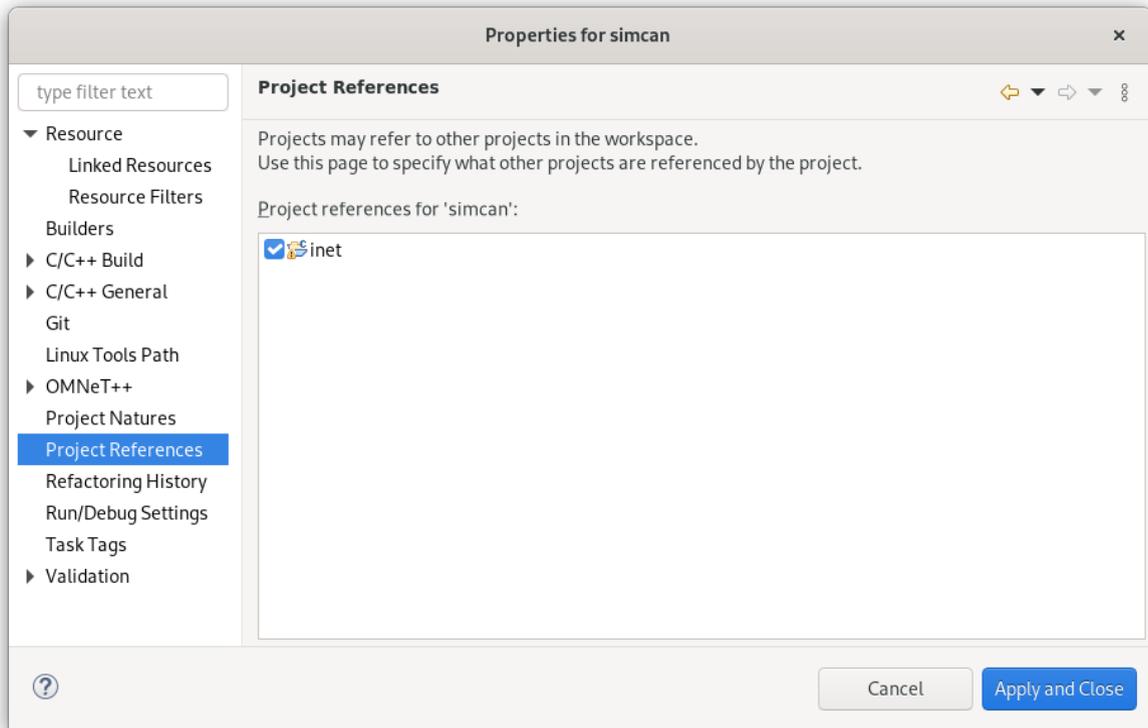
2.1.3.1. Graphical installation

With the OMNeT environment loaded (source setenv), load OMNeT++(run omnetpp).
Make sure to import INET (File | Import | Existing Projects)



Repeat the same process for the SIMCAN directory.

Now make sure that INET is referenced from the SIMCAN project: From the Project Explorer window, right-click on SIMCAN: Properties | Project References and make sure that INET is checked.



Use Project | Build Project to build Simcan.

2.1.3.2. Cluster installation

After building SIMCAN, copy the SIMCAN directory over to the cluster, then run:

```
make cleanall
```

Add the path to the INET in the variable INET_PROJ in the Makefile:

```
nano Makefile
INET_PROJ=~/inet4.3/
make -j $(nproc) MODE=debug
```

If something fails, check:

```
which inet
which omnetpp
scl enable devtools-9 bash
```

2.1.4. Running Simcan

Add to .bashrc the variable INET_HOME with the INET path.

```
$nano ~/.bashrc
$export INET_HOME=~/inet4.3/
```

Run the scenario from its folder with:

```
./run_debug -u Cmdenv
```

2.2. Installing OTF2

For the tracing tools, the HPC batch cluster had also installed TAU so we decided to install OTF2 and decide which one would work better. For the OTF2 installation the following guide has been created:

- First of all download the tar.gz file from the website

```
wget https://zenodo.org/record/4682684/files/otf2-2.3.tar.gz
```

- The next step will be untar the downloaded file

```
tar -xvfz otf2-2.3.tar.gz
```

- When the file is untar, access the directory created as otf2-2.3

```
cd otf2-2.3
```

- From the directory we have to configure the system, so we will execute the next commands:

```
./configure --prefix=$HOME/otf2-2.3/  
export PATH=$PATH:$HOME/otf2-2.3/  
make  
make install
```

- For check the configuration, from \$HOME/otf2-2.3/:

```
cd doc/examples/  
module load <mpi_module>  
make
```

3. Future Work

Due to the external circumstances mentioned above, this project is on its halfway to be finished. The next step will be to continue with the trace reader support, to study the most accurate trace reader that will work in common with SIMCAN. After this, the next step will be put under test all the tools installed, looking for bottlenecks, inconsistencies and potential problems when executing simulations in the cluster.

Once this work is done, the methodology will be finished and ready for being used at CERN to explore the efficiency of actual and future HPC clusters.

4. Special Thanks

I want to thank this opportunity of working at CERN to Pablo and Alberto, for guiding me through this experience, showing a little part of the research world.

Also, I want to say special thanks to CERN representatives, for helping all the Summer Students making the “COVID experience” as if we were working side by side with our partners and making this internship so enjoyable.