

How to stalk the users of your cluster using OGRT

1st EasyBuild User Meeting

29.01.2016

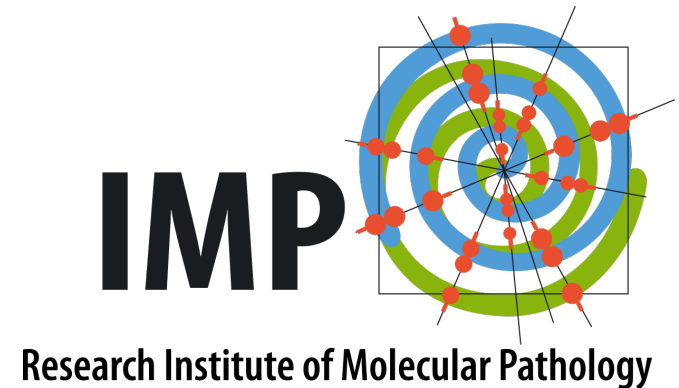
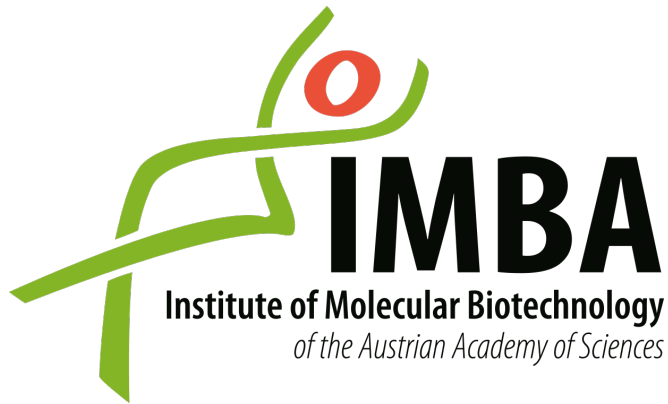
Ghent, Belgium

<http://goo.gl/zbvChr>

Hello. Who am I?

Georg Rath

Systems Engineer at IMP/IMBA



History

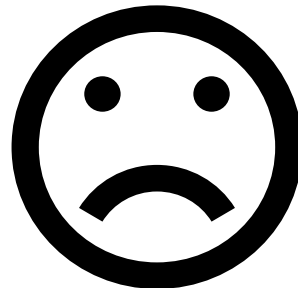
we got this cluster handed to us...

- diverse workload (we have been told)
 - next generation sequencing
 - electron microscopy reconstruction
 - video compression
 - and who knows what else...

Who knows?

Let's ask the users!

1. "We use this pipeline: 'mnseq_4_custom_3.Copy 2.sh'."
2. Go through the shell script, check the programs, module loads without versions, hardcoded paths, everything you could and could not imagine.
3. Rinse, repeat



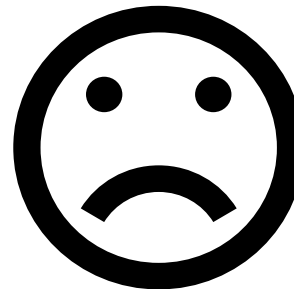
Who knows?

Let's hook module loads!

A sample ~/.profile:

```
module load cd-hit  
module load emboss  
module load hmmer  
module load ncbi-blast  
module load ncbi-blast+  
module load mafft  
module load muscle
```

load \neq *use*



Who knows?

"Did you check out XALT?"

- Somebody had the same problem!
- Compilation tracking?
- 'Watermarking' of binaries and shared objects?
- Amazing!

Unfortunately...

- Needed a launcher (was true in 2014, parts of 2015)
- Not designed to track everything
- Tailored for HPC (TACC) needs

What do we want?

- Track execution of **all** programs in a job
- Track every shared object a program loads
- Be able to embed a signature into programs and shared objects
- Without a launcher
- Be as lightweight as possible
- Be as transparent as possible
- Process data in near real-time



An old friend appears...

LD_PRELOAD

The loader "preloads" a shared object when loading a dynamic executable.

...combine with a GCC 'constructor':

No Launcher 😊

(for dynamically linked executables using the GNU libc)

And the shared objects?

"The **dl_iterate_phdr()** function allows an application to inquire **at run time** to find out which **shared objects** it has **loaded**."

With a signature?

- **dl_iterate_phdr()** provides ELF program headers
- can we get our signature into a program header?
 - link section into target program and mark it allocatable

"The `dl_iterate_phdr()` function is Linux-specific and should be avoided in portable applications."

Signature

- Similar to XALT
- Link in an object file
- Creates a note section in ELF (GCC does this too)
 - gets loaded into memory on execution
- Embeds an UUID

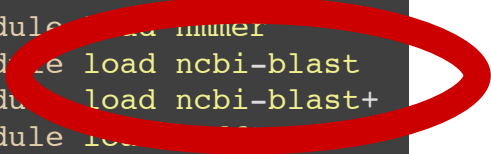
Why the signature?

- same path - different executable
- discern user generated programs

For example:

A sample ~/.profile:

```
module load cd-hit
module load emboss
module load mmmer
module load ncbi-blast
module load ncbi-blast+
module load ...
module load muscle
```



"shadowing" of programs

Are we lightweight?

We are doing everything in memory.*

Are we transparent?

Well, our users did not notice...

*`realpath()` *could* walk the filesystem

Recap

- ~~Track execution of all programs in a job~~
- ~~Track every shared object a program loads (with a watermark)~~
- ~~Without a launcher~~
- ~~Be as lightweight as possible~~
- ~~Be as transparent as possible~~
- Process data in near real-time

The Transport



*for debugging only

Demo

Outlook

- Testing
- Documentation
- DB Level XALT compatibility
- Symbol level tracking (has the function x() been used)

Fin

<https://github.com/IMPIMBA/ogrt>