

Exacloud: An Overview

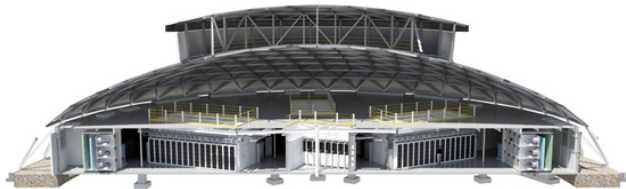
Stephan Lindner

6/7/2016

1. What is Exacloud? And why is it on a Linux server?

What is Exacloud?

- Exacloud a server run by OHSU to support large-scale, computational and data intense workflows.
- Currently more than 35 Terabytes of memory and more than 750 Terabytes of usable storage.
- Housed at the Data Center at OHSU's West Campus.



Exacloud and Linux

- Exacloud uses Linux as operating system.
- By contrast, the CHSE server (and our computers) use Windows as operating systems.
- An operating systems is the “habitat of your programs” — the software that manages a computer’s basic functioning.
- Linux and Windows get along OK, but they do not particularly like each other.
- Most programs (such as R, stata) are developed for both OS (and Mac’s OS).

Why does Exacloud uses Linux?

Linux is ...

- Very stable.
- Slim and scalable and therefore has less hardware requirements.
- Designed as a multi-user system.
- More secure than Windows.
- FOSS (Free and Open Source Software).

What does this mean for us:

- Most programs we use for our analysis are open-source and are developed on Linux: R, git, markdown.
- Stata is more geared toward Windows but has some Linux support.
- Interaction between local Windows machines and a Linux server are not perfect but fine.

2. Accessing and navigating Exacloud

Accessing Exacloud via ssh

- Remote access of CHSE server: through Windows desktop.
- Remote access of Exacloud: through ssh (secure shell).
- Shell is a command prompt that you can use to interact with the computer (e.g., run programs).
- Bare-bone, 1970 technology that requires very little memory.

MobaXterm: ssh for Windows

- Install MobaXterm on desktop.
- Initiate ssh session with
 - ▶ Remote host: `exac1oud.ohsu.edu`
 - ▶ User name: your OHSU user name.
- Prompts for password and then connects to server.

Switch to MobaXterm

Navigating Exacloud

A couple of useful commands:

- Printing the working directory: `pwd`
- Listing files in current directory: `ls (-lh / -a)`
- Start R: `R`
- Start stata: `stata`
- Check git status: `git status`
- Work with hcondor (?): `condor_submit`, `condor_q`

3. Interactive and non-interactive use of Exacloud

Interactive versus non-interactive R / Stata session:

1. Interactive:

- ▶ Workflow: Work on code in script file → Evaluate code in R / Stata → Revise / Debug code → ...
- ▶ Setup: Umbrella program that integrates editor with statistical program: RStudio, standard Stata GUI, etc.
- ▶ Requirement: Umbrella program needs to be able to transfer code chunks to Stata / R and display results.

2. Non-interactive mode:

- ▶ Workflow: Write full script file → run full script in R / Stata → Revise / Debug → ...
- ▶ Setup: Call script file through umbrella program / shell.
- ▶ Requirements: some way to call R / Stata.

Interactive mode on servers:

Two different options:

1. Run umbrella program and R / Stata on server:
 - ▶ Requires a lot of data traffic between remote server and local computer.
 - ▶ This is how we use CHSE server.
 - ▶ Not possible for Exacloud server because it does not have a desktop environment.

Interactive mode on servers:

Two different options:

1. Run umbrella program locally, R / Stata on server:

- ▶ Requires little data traffic between remote server and local computer.
- ▶ Umbrella program needs to be able to transfer code / results back and forth between local computer and server.
- ▶ Possible for Exacloud depending on umbrella program:
 - ▶ Rstudio: No
 - ▶ Stata: ?
 - ▶ Emacs: Yes

Non-interactive mode: a simple example

1. Script in R example1.R

```
x <- 1:1000  
summary(x)
```

2. Run script using Rscript:

```
Rscript example1.R
```

Non-interactive mode: a slightly more complicated example

1. R markdown script in R: example1.Rmd

Example markdown file

```
```{r}
x <- 1:1000
summary(x)
```
```

2. Master script to knit R markdown script in R:

master-knitr.R

```
library(knitr)
library(rmarkdown)

knit(commandArgs(TRUE)[1])
```

- ▶ Supply R markdown file as argument for Rscript.
- ▶ Function knit then knits that R markdown file and outputs .md file.

3. Run script using Rscript:

```
Rscript master-knitr.R example1.Rmd
```


Non-interactive mode using HTCondor

- Purpose: Efficiently allocate resources to processes that run on decentralized computing system such as Exacloud.
- Basic usage is pretty simple:
 1. Write a submit file that tells HTCondor which program to run.
 2. Submit the request to HTCondor.
- There is **a lot** we can do with HTCondor:
 - ▶ Request memory for job.
 - ▶ Run script files in different directories.
 - ▶ Use macros, conditionals, ...

Non-interactive mode using HTCondor

1. HTCondor submit file: `examples.htc`

```
Executable      = /usr/bin/Rscript  
Arguments       = "master-knitr.R example1.Rmd"
```

2. Submit request:

```
condor_submit examples.htc
```