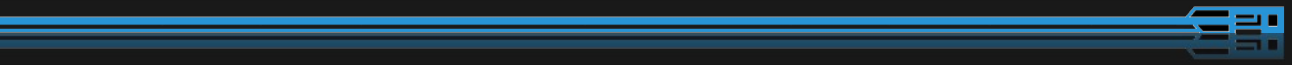
OpenFlightHPC Bootcamp

Liverpool HPC MSc Program

Next Generation HPC

February, 2020

Agenda



openflightHPC
Next Generation HPC on Any Platform

12:00 Introductions

What is Next-Generation High Performance Computing?

The HPC Market

Cloud HPC: A (sort of short) History

13:00 OpenFlightHPC project overview

13:30 Exercises in OpenFlightHPC

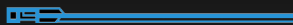
HPC cluster applications and workloads

Next-generation workloads

Review, wrap-up and questions

16:00 Close

alcesflight



Next Generation HPC

Alces Flight

February, 2020

Alces group – what is it?

openflightHPC
Next Generation HPC on Any Platform

- UK small/medium sized enterprise (SME)
- Private self-funded company
- Specialist in High Performance Computing
- Formed in 2008
  - >100% YOY growth for FY1-5
  - No external funding or shareholders
  - Run of 42 profitable quarters
  - Parent + trading divisions

ALCES SOFTWARE

alcesflight

alcesflight

Alces group – what does it do?

openflightHPC
Next Generation HPC on Any Platform

- Builds Research Computing environments for customers
- Acts as a service provider
    - Logistics team physically delivers and installs equipment
    - Engineering team manages and monitors customer systems
    - Software teams responsible for:
        - Building the tools and products used to deliver our services
        - Creating new products to help solve customer problems
        - Supporting our consultants working with 3rd-party software

alcesflight

Alces group – what does it do?

openflightHPC
Next Generation HPC on Any Platform

- Project examples:
  - Configuring a bare-metal HPC cluster for a University
    - e.g. Barkla HPC cluster at Liverpool
  - Extending an existing bare-metal facility with new technologies
  - Reconfiguring an existing facility to do a new thing
  - Advising customers on creating a new HPC facility using hardware, public or private cloud resources
  - Enabling on-demand research computing environments for dynamic end-user requirements

alcesflight

Alces group – what does it do?

openflightHPC
Next Generation HPC on Any Platform

- Technology examples:
  - Hardware
    - IBM, HPE, Dell, Cray, Supermicro, Intel, Mellanox, Cisco….
  - Cloud
    - Amazon Web Services, Microsoft Azure, Google Compute cloud
  - HPC software
    - Linux, job-scheduler, Docker, CUDA, MPI, compilers, languages
  - Application software
    - NAMD, Gromacs, Tensorflow, Lightwave, WRF, Code Saturne, MapReduce, Jupyter Notebook, Matlab

alcesflight

Team members

- Technical Sales Consultant
  - Wil Mayers
- Change Management lead
  - Cristin Merritt
- Technical Product Manager
  - Stu Franks

Traditional High Performance Computing

openflightHPC
Next Generation HPC on Any Platform

- Serial job
  - One CPU core / node / task at a time
  - If dataset is infinitely divisible these are embarrassingly parallel
  - Often run in workflows or batches (e.g. batch-cluster)
- Multi-threaded jobs
  - Like serial jobs, but use multiple resources on the same node
- Parallel jobs
  - Use resources (CPU/memory) from multiple nodes
  - Commonly use an API to simplify programming
  - Message passing interconnect (MPI) handles communication
  - Scalability depends on both application and system

alcesflight

Speciality HPC workloads

openflightHPC
Next Generation HPC on Any Platform

- Distributed / big-data jobs
  - Lots (and lots and lots) of data
  - You want to do lots of work on the same dataset
  - Compute is more efficient to run where the data is already located
  - Designed for large scale workloads
    - Fault-tolerance built in to the software framework
    - Difficult to get smaller workloads to run efficiently
- Artificial intelligence / machine-learning
  - Training a computer model to make statistical predictions
  - Outputs typically include a “confidence” rating
  - Works well on many-core processors (e.g. GPUs)
  - An example of an “overlay” technology
    - Can help improve/optimise other workflows

alcesflight

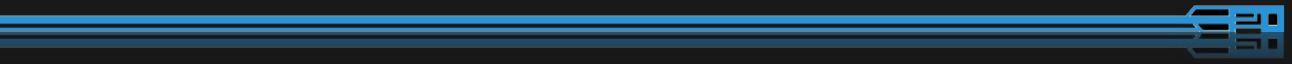
High Performance Workloads

openflightHPC
Next Generation HPC on Any Platform

- Sometimes just one application
  - Matlab, Python code, R script, Java script
  - Gene sequence searching, mapping, detection
  - NAMD, Gromacs, Quantum Expresso
- Sometimes a workflow of applications
  - Prepare data -> process data -> recombine outputs -> present
  - Sieve data -> compare found results -> present
  - Use Hadoop to find pictures of animals on the web -> Use AI to detect which animals are cats -> Use python array to organise by geographic region -> How many Liverpool residents have cats?

alcesflight

Where do these applications come from?



- Home-grown
  - Written by you, your lab partner, your tutor
- Open-source
  - Written and published by a group of developers
  - Mixture of small, single-source and massive group efforts
  - Myriad of software licenses, EULAs
- Commercial
  - Can be closed code-base, proprietary
  - Could be optimised/supported/tested version of open-source app
  - You might need to pay to use it

Distribution of HPC software apps

- Compile it from source-code
  - Pretty standard for open-source HPC cluster applications
- Download binary packages
  - Download, install and run a Windows app
  - RPM, DEB, DMG packages for Linux/Mac
- Get it in a container
  - Docker, Singularity, VMware, Kubernates, Virtualbox VM
  - AWS AMI, Azure or GCP disk image
  - Often includes app, libraries and supporting bits of operating system
- Launch or use an existing online service; e.g.
  - Github / Gitlab
  - Jupyter Notebook
  - Tensorflow.org

Workload maturation

- Most work begins in “discovery” mode
  - Try something, see if it works, try something different
  - Rapid prototyping, agile software development
  - Fast, innovative, quickly changing
- Compute resources tend to be
  - Low capacity, with limited scalability testing
  - Leveraging third-party software, often from multiple sources
  - Very variable during the development process
  - Unoptimised, inefficient, difficult to scale

Workload maturation

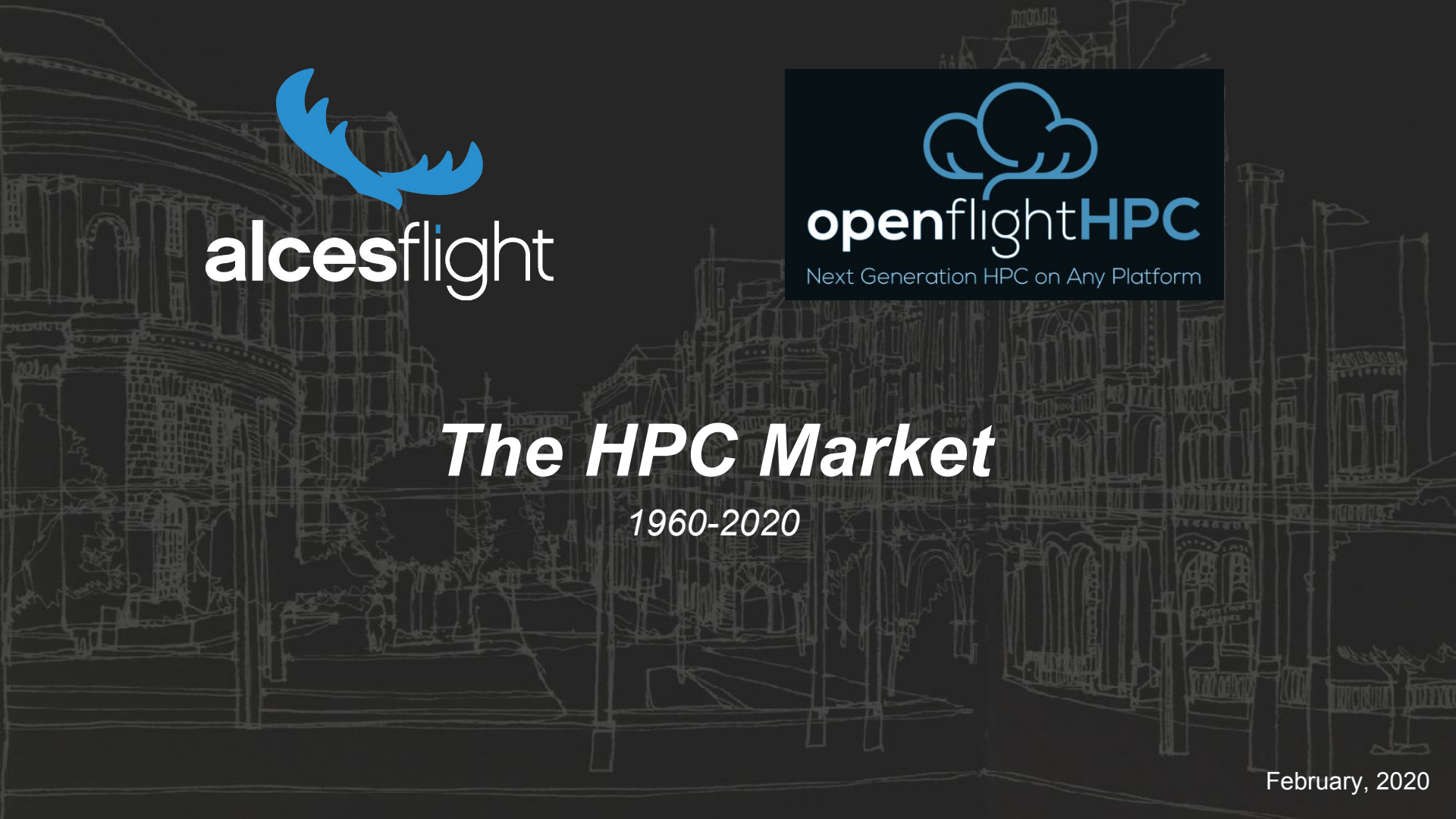
- As workload matures
  - We have better idea of what compute we need
  - Centralise on a specific software environment
  - We can begin to optimise for scale, performance, cost
- Compute resources tend to be
  - Less generic and more specific
  - Capable of running unattended
  - Scaled to handle bursts of work
  - Cost optimised

Workload lifecycle

- High Performance Computing is all of these things
  - Unoptimised, adhoc, rapidly changing
    - Launching an AWS AMI created by a vendor to test a theory
    - Using a vendor-supplied service to test something
    - Trying your MapReduce code on a public Hadoop service
    - Testing your gene sequence for HG19 matches on Galaxy.org
  - Tightly-coupled, highly optimised, reproducible
    - Running 100 x NAMD chemistry simulations on Barkla HPC cluster
    - Submitting 2,000 tensorflow simulations to a batch cluster
    - Connecting to the pay-per-use Azure MapReduce service to process data transmitted from self-driving cars

What is Next-generation HPC?

- All those applications…
  - …run as any of those workflows
  - …using software from any of those sources
  - …deployed using any of those mechanisms
  - …on any of those platforms
  - …by end-users of mixed abilities
  - …over the workload's entire lifecycle.
- Plus the things that were invented while we just read that.


alcesflight
openflightHPC
Next Generation HPC on Any Platform
The HPC Market
1960-2020
February, 2020

Traditional HPC cluster market

openflightHPC
Next Generation HPC on Any Platform

- Most traditional HPC users run on HPC clusters
  - Central service shared between many users
  - Lots of compute nodes, each running separate OS
  - Shared storage
  - Fixed software environment with job-scheduler for time-sharing
  - One cluster typically lasts 3-5 years then is replaced
- UK has lots of these
  - National facility is called “Archer”, hosted at EPCC
  - Liverpool University has “Barkla” HPC cluster
    - 50-100 similar machines across the UK

alcesflight

On-prem HPC market limitations

openflightHPC
Next Generation HPC on Any Platform

- Only so much HPC work to do
  - Large capital investment required = high barrier to entry
  - HPC applications somewhat specialised to setup + use
- Logistical requirements cause limitations
  - Tricky to deliver and install new clusters outside local area
  - HPC cluster installations are often loss-making for vendors
- Lack of standardisation and repeatability
  - Every installation is it's own special snowflake
  - Irreproducibility reduces value of research outputs

alcesflight

Addressing a changing market

openflightHPC
Next Generation HPC on Any Platform

- *Only so much HPC work to do*
  - Make HPC more approachable
  - Reduce barrier to entry
- *Logistical requirements cause limitations*
  - Stop relying on hardware
  - Build a fairer, sustainable market
- *Lack of standardisation and repeatability*
  - Standardise
  - Educate

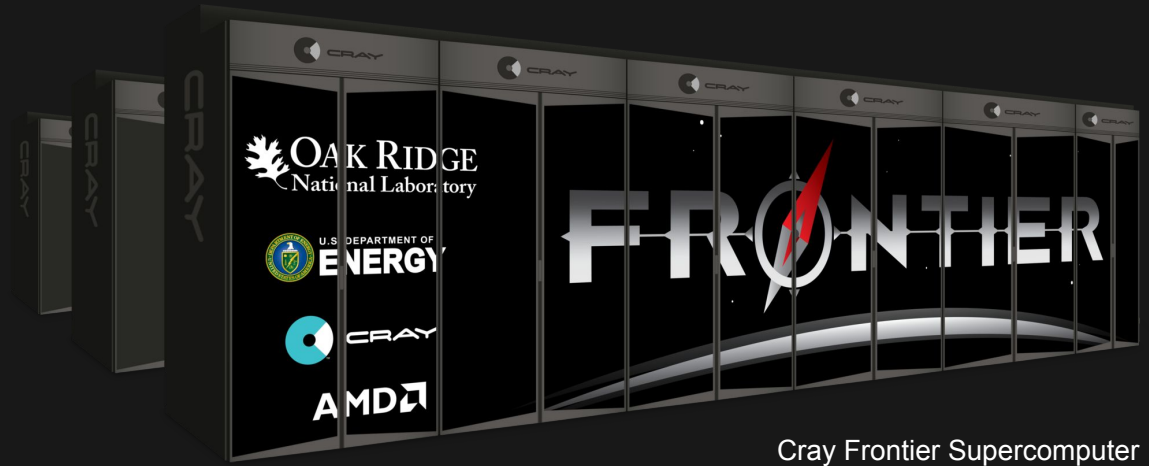
alcesflight

Where HPC fits into the IT Industry

Ferrari 812 Superfast

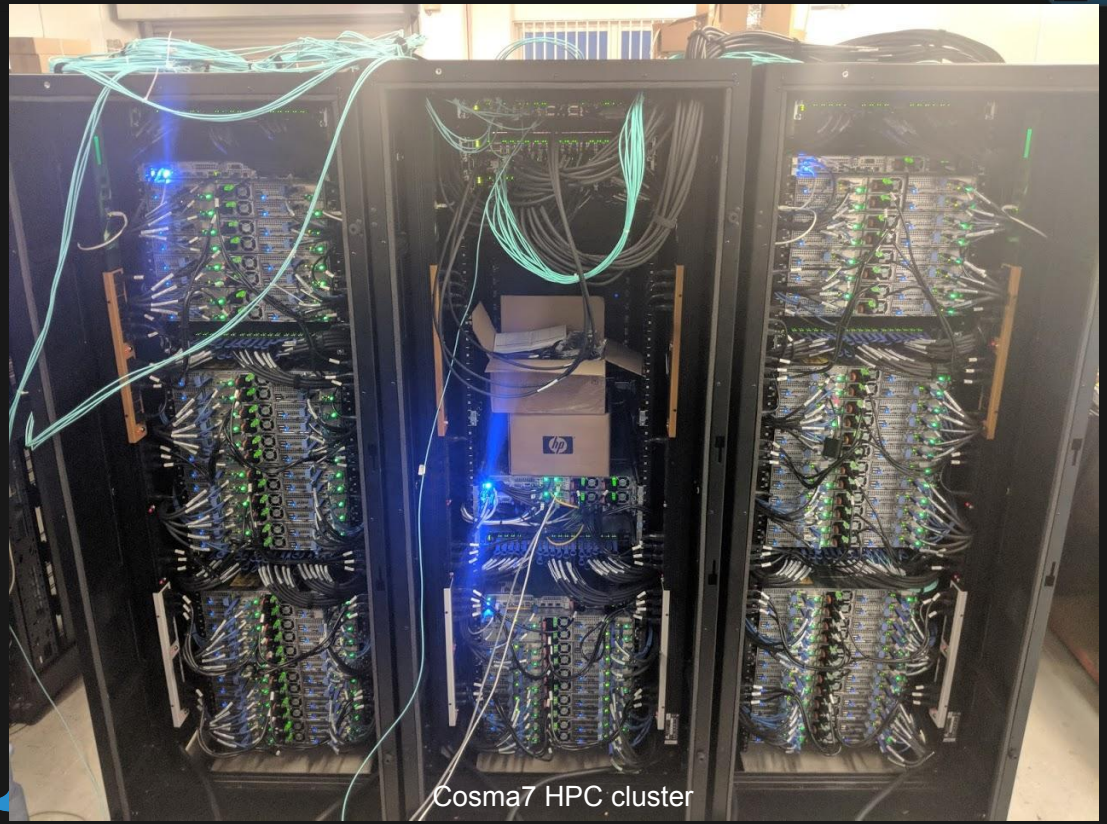
Where HPC fits into the IT Industry

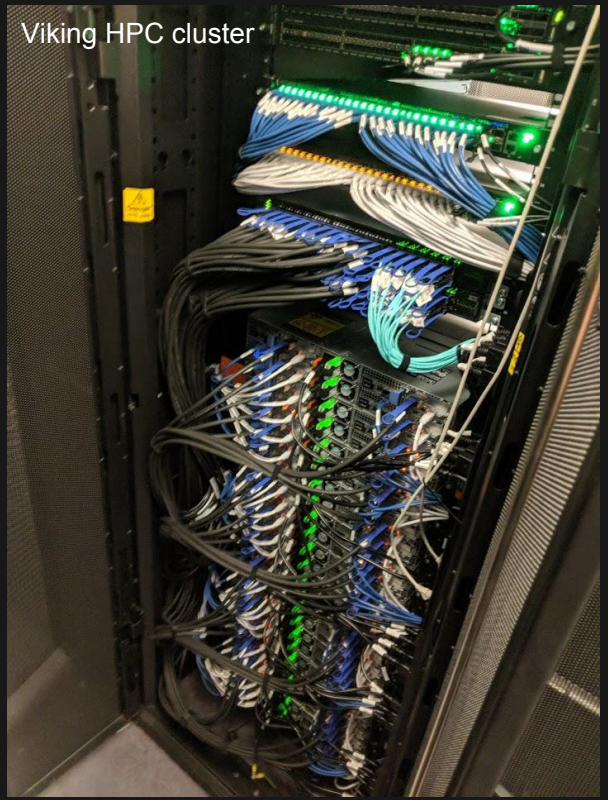
Cray T90 Supercomputer



Cray Frontier Supercomputer

Where HPC fits into the IT Industry


Cosma7 HPC cluster


Viking HPC cluster

Where HPC fits into the IT Industry

openflightHPC
Next Generation HPC on Any Platform

Ferrari 812 Superfast

Vauxhall Corsa 1.2

Landrover Defender TD5

Trex Fly 5 bicycle

30 seater school bus

alcesflight

Cost, not technology, is now the key factor

openflightHPC
Next Generation HPC on Any Platform

- Commodity computers and storage
  - Allow low-cost HPC clusters to be created
  - Very good for well-developed workloads
  - Inflexible for discovery workloads
- The rise of cloud computing
  - Much easier to perform discovery work
  - On-demand launch, try it out, throw it away, try again
  - Much lower barrier to entry with access to latest technology
  - Currently more expensive for static workloads….

alcesflight

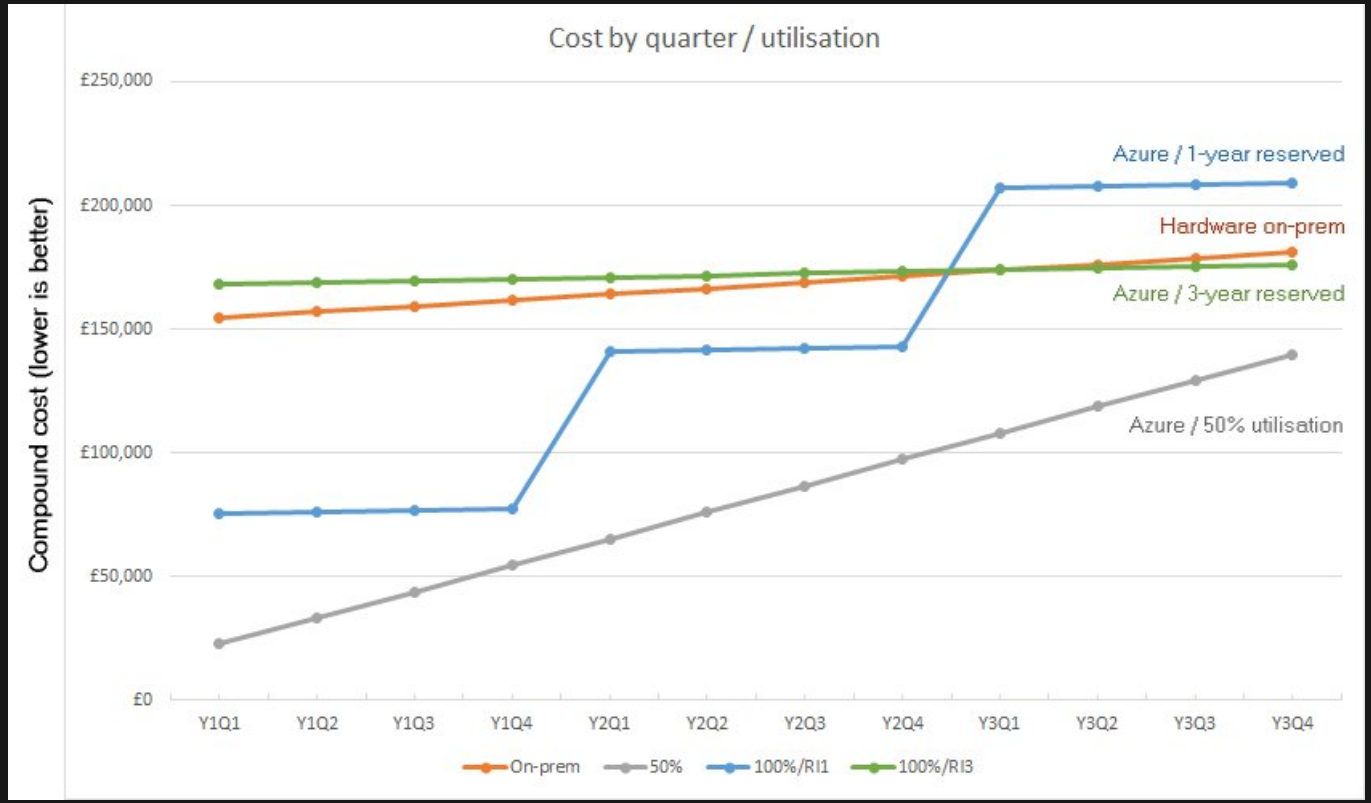
Is Cloud more expensive than on-prem?

openflightHPC
Next Generation HPC on Any Platform

- Buying and running your own hardware
  - Simple to manage your budget
  - Difficult to quantify + measure all the real costs
  - Multi-year investments in buildings, people, process, etc.
  - Measure cost-effectiveness in arrears
- Cloud computing
  - On-demand availability vs. committed spend to get discounts
  - Few demands on local infrastructure; great for start-ups
  - Sometimes hard to predict the final bill
  - Measure cost-effectiveness day-by-day

alcesflight

On-prem vs Cloud examples



On-prem vs cloud examples

openflightHPC
Next Generation HPC on Any Platform

- NHS HPC cluster
  - Gene therapy unit, Manchester Children's Hospital
  - Cluster used to diagnose and treat patients
  - Samples taken, rapidly sequenced and processed
  - Patients arrive at 9am and leave by 5pm with results
- On-prem HPC cluster can process one child per day
  - 22% utilisation (9-5, Mon-Fri, 52 weeks p/a)
- Cloud HPC cluster grows/shrinks on-demand
  - 3x cheaper to run (or 3 children per day for same cost)

alcesflight

HPC market futures

- IDC predicting migration to cloud
  - Some challenges for very specialist HPC machines
  - Cloud technology already good enough for most small-medium HPC clusters
- As cloud takes over IT, hardware becomes more expensive
  - Owning your own data-centre becomes rarer
  - Hardware vendors downsize as economy of scale is reversed
  - Building an on-prem HPC cluster becomes more expensive
- Skills-gap in building, deploying and managing workloads

Cloud HPC: A (sort of short) History

Cristin Merritt

Alces Flight Cloud Adoption Project

February, 2020

Alces Flight at Liverpool University

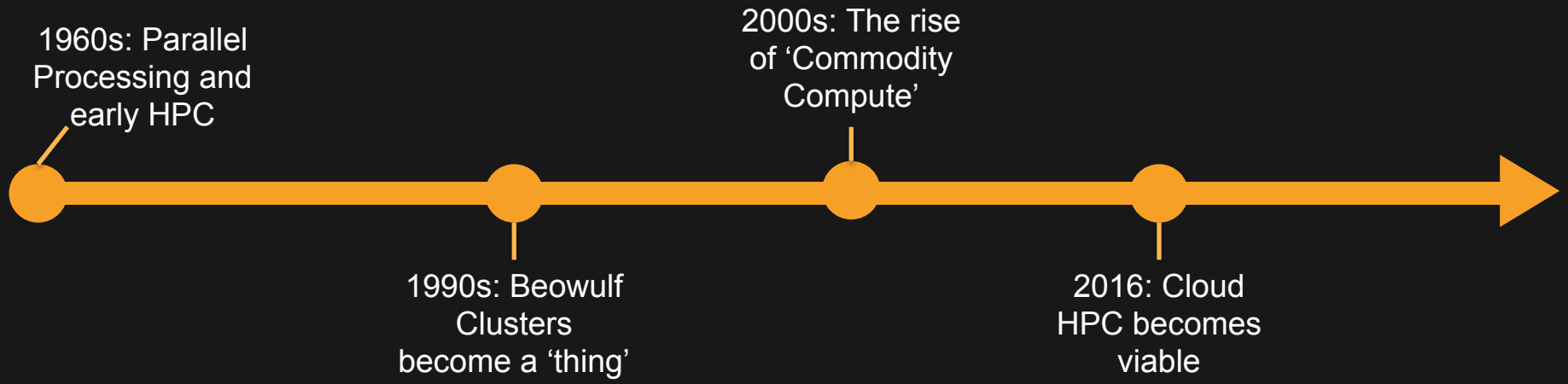
openflightHPC
Next Generation HPC on Any Platform

- Integrators for Liverpool since 2017 (Barkla Cluster)
- Provide Managed Services for Liverpool
    - Look after the daily checks, general maintenance, and manage/perform change requests
    - Build cloud HPC projects (AWS and Azure for Liverpool)
    - Allow on-site team at Advanced Computing Facilities to build user expansion
- Launched OpenFlightHPC Project in November, 2019…

But a bunch of stuff happened before any of this…

alcesflight

A poorly drawn HPC Timeline



2016: Cloud HPC (versus Hardware)

openflightHPC
Next Generation HPC on Any Platform

- Launch of Alces Flight Compute cluster product
  - A traditional HPC compute cluster running on public cloud
- Launch of Cloud Adoption Project for HPC
  - Tracked subscriptions on AWS and Azure (total of around 275 sites… 222 of those are still considered active)
  - Five sites agreed to complete profile of use

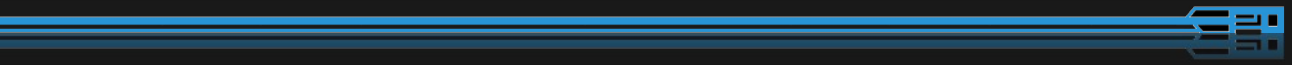
We learned a lot of things…

alcesflight

Cloud HPC disrupts HPC culture

- You cannot see/hold the hardware
- Technically anyone can do HPC on the cloud (purchasing barrier quite low)
- Practically no barriers to having the ‘newest kit’ running your workload
- Cloud opens the door for a lot of new fields with ideas on how to leverage HPC (skills shortage and gap emergence)
- Cloud allows projects to run as fast or as slow as you desire (time v. cost / workload/job queuing)

Good things about cloud for HPC

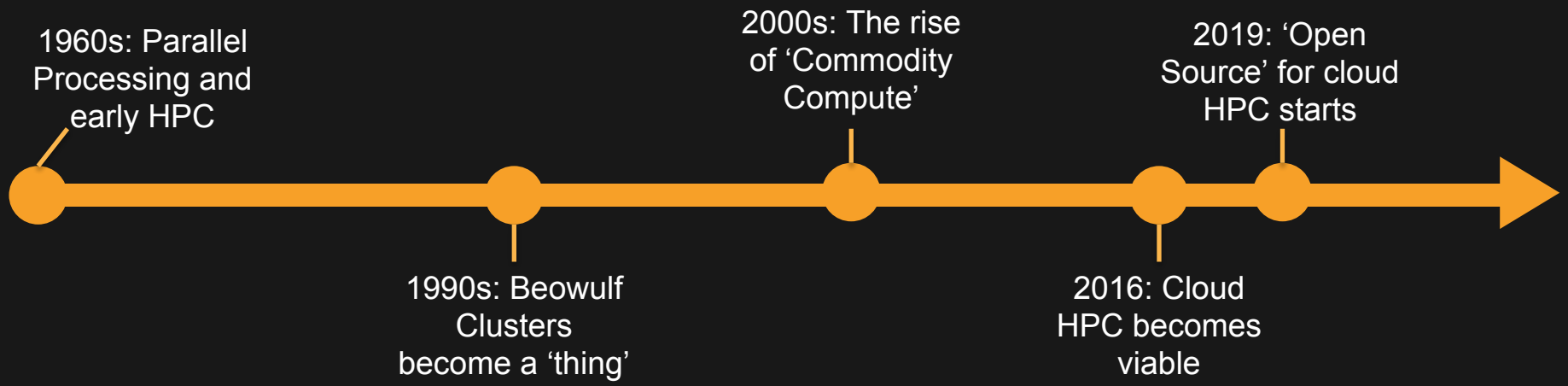


- It does really well when it's project-based
  - Much easier to collaborate with others
- It loves open-source
  - Use whatever software you can find
- It does very well when work is embarrassingly parallel
- It is a great tool to make longer-term decisions with
  - Types of hardware to purchase
  - Testing new theories and ideas
  - Taking ideas into production

Bad things about cloud for HPC

- Despite what the cloud vendors tell you, cloud HPC can get very expensive
    - Budget is not fixed, so you need to keep control
- Commercial vendors still don’t quite know what to do about cloud
- Over abundance of value-add ‘services’
    - Complex environment to navigate for a beginner
- Many parts means reproducibility can be an issue
    - How did I get here?

A poorly drawn HPC Timeline



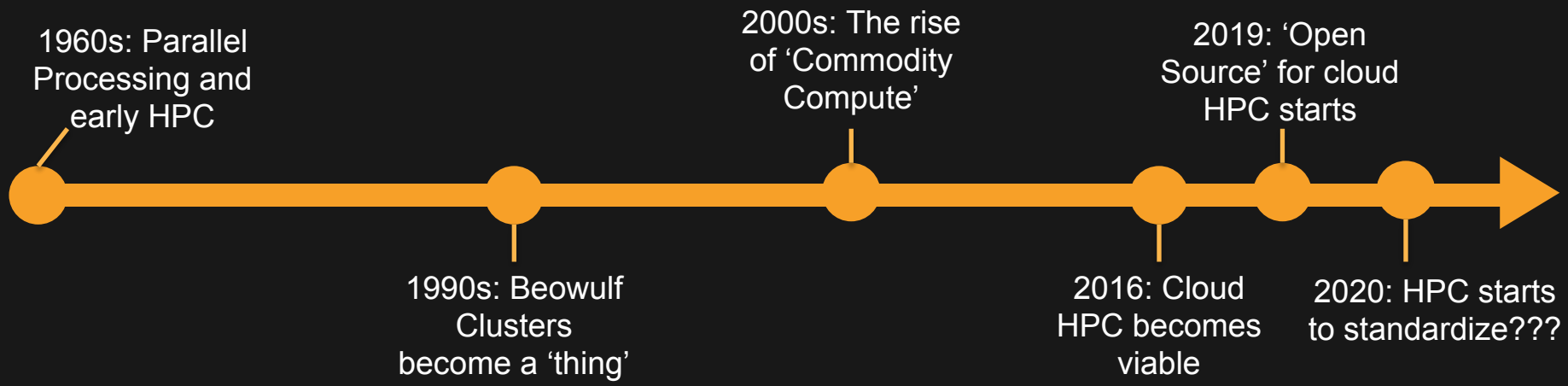
2019: Open-source for cloud HPC

openflightHPC
Next Generation HPC on Any Platform

- Subscription services to cloud HPC deemed too complex to maintain.
- Support for cloud HPC is difficult for cloud vendors to provide – especially as some level of customization work seems to happen in HPC projects.
- Open-source far more viable as you can optimize 75-90% across ANY platform.

alcesflight

A poorly drawn HPC Timeline



2020: Real Skills

openflightHPC
Next Generation HPC on Any Platform

- People skills are ‘in’
    - Customer/Client/User-facing roles
    - Working in partnership or collaboration
- Niche skills are in – especially in Artificial Intelligence or Machine Learning optimization
- Services creation is in – see ‘people skills’ above.
- Standards are in… it’s not the fastest hardware it’s getting things done consistently and are easy to replicate.

alcesflight

2020: Where the jobs are…

openflightHPC
Next Generation HPC on Any Platform

- Research Software Engineers (RSEs): Client-facing roles dedicated to getting workloads to run well. (Field still new – launched approx. 2014)
- Software/Solution Architects (SAs): Commercial version of RSEs, often working with specific clients to build/optimize workloads.
- Computational Data Scientist: Focus on optimizing the access and use of large data sets – often deals in storage. (Might see ‘Storage Engineer’ in prospective job titles)
- CFD / Visualization Engineers: Strong on cloud development and proven to speed along projects which were once workstation based.
- Software Engineering: Generally focused in specific fields – demands in earth sciences (oil and gas) government (military, space exploration, weather) and open-source (ex. Redhat, SuSE, AWS, Azure…)

alcesflight

2020: Where Alces Flight is…

openflightHPC
Next Generation HPC on Any Platform

- Focused on standardization, automation and process of HPC services.
- Focused on utilizing open-source as a means of allowing our clients to work across platforms.

Launched OpenFlightHPC as a centralized project to help users to access, develop and run HPC workloads across any compute platform.

alcesflight

Part One: Quick Introduction

February, 2020

Bootcamp agenda

openflightHPC
Next Generation HPC on Any Platform

- Part One: What is OpenFlightHPC?
  - The project goals
  - How you can get involved
- Part Two: Familiarisation exercises
  - Running applications on a cloud-based HPC cluster
  - Running a big-data workload
  - Running a machine-learning workload

alcesflight

What is OpenFlightHPC?

- *A central resource to help users run HPC workloads on any compute platform*
- HPC workloads
  - Traditional batch and parallel HPC cluster jobs
  - Containerised / packaged applications
  - Client / server web services
- Multi-platform
  - Bare-metal
  - Private-cloud, public-cloud, online services

What can OpenFlightHPC do for you?

- Help to provide you with what you need to run your HPC workload
- Connect you to other users running similar jobs
- Provide guidance for improving and optimising your work
    - Getting started with your discovery process
    - Performance optimisation
    - Improved accessibility
    - Process repeatability and sustainability
    - Budget control and cost reduction

Example – image processing workload

- Initial discovery work
  - Performed using individual AWS instances
  - Iterative process improving successive images over time
- Scalability improvements
  - Host your existing image on larger / more-capable instances
  - Build a workflow that allows your job to run unattended
- Move to production
  - Use an autoscaling batch HPC cluster to process data on arrival
  - Run on the existing Barkla HPC cluster using spare cycles

Example – MapReduce workload

- Initial discovery work
  - Run Apache Hadoop on a single machine
  - Perform test runs with different data to ensure your job works
- Scalability improvements
  - Upload your data to Amazon EMR service
  - Run larger queries to ensure your code works at scale
  - Consider an AI decomposition to narrow search parameters
- Move to production
  - Ensure your workflow is portable across a range of services
  - Review cost of using EMR vs. building your own Hadoop service

How can you get involved?

- Use it
  - Bootcamp provides some example of workflows
  - Plan your future workloads, looking for the applications you want to use
  - Think about how you want to run your workload
    - Where will your software come from?
    - How will that software be distributed and deployed?
    - How can you run at scale?
    - Once things are running, how will you optimise your jobs?
- Share your experiences
  - Provide information for your users, stakeholders and researchers
  - Help other users on the OpenFlightHPC community site
  - Publish your own use-cases, success stories and research outcomes

Part Two: Introductory Bootcamp

February, 2020

OpenFlightHPC familiarization

openflightHPC
Next Generation HPC on Any Platform

- You will need:
  - A laptop with a web-browser
  - Wireless internet access
- OpenFlightHPC web access:
  http://bootcamp.openflighthpc.org/

- Alternatively:
  - An SSH client and a VNC client
    - Windows: http://tiny.cc/turbovnc-win64

alcesflight

OpenFlightHPC familiarization

- What are you logging in to?
  - A compute cluster running in public cloud
    - One login node (*gateway1*)
    - Two small compute nodes (*node01* and *node02*)
    - Virtual private internal cluster network
    - Small, shared filesystem
    - Slurm job-scheduler
- What are we going to do with it?
  - Navigate the cluster, create and run some batch jobs
  - Install and configure some serial and parallel applications
  - Setup Apache Hadoop and run a MapReduce example
  - Run a job using an AI application container

Logging in and cluster navigation

```
# id
# pwd
# df -h .
# sudo pwd
# sudo yum install dos2unix
# ping node01
```

Press “y” to confirm

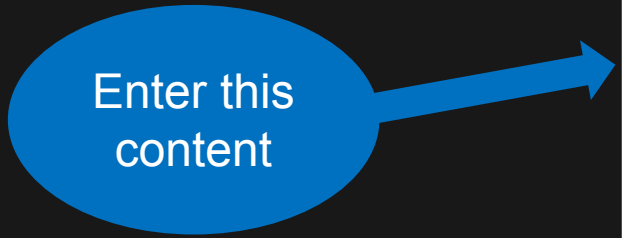
```
# flight start
# ssh node01
# df -h .
# exit
# sinfo -Nl
# flight info
# flight env avail
# flight set always on
```

Linux HPC cluster application management

- Complexity of application varies
  - Some are simple; many are not
  - Most use a variety of different libraries
  - Some also use APIs (e.g. message-passing)
  - Version control is important for reproducibility
  - Dependencies must be satisfied across all nodes
- You can choose how to manage this
  - Configure things manually
  - Get help

Compiling and running an app manually

```
# nano hello.c
```




```
#include <stdio.h>
int main() {
   printf("Hello, World!\n");
   return 0;
}
```

Press CTRL+X to exit and enter “y” to save

```
# gcc -o hello hello.c
# ls
# ./hello
Hello, World!
```

Running a batch job on your cluster

openflightHPC
Next Generation HPC on Any Platform

```
# sinfo -Nl
# nano myjob.sh
```

Enter this content

```
#!/bin/bash
#SBATCH -N1
echo “Hello from $HOSTNAME”
date
sleep 5
./hello
```

Press CTRL+X to exit and enter “y” to save

```
# sbatch myjob.sh
Submitted batch job 2
# cat slurm-2.out
```

Use this number here

Command-line vs graphical desktop

- Depending on your workflow, you might use
  - A command-line interface
  - A graphical desktop interface
  - A web-browser (locally or on the cluster)
- Graphical interaction is great for
  - Discovery work
  - Debugging an issue
  - Anything visual or interactive
- Command-line interfaces are good for
  - Unattended jobs
  - Anything that outputs a lot of data

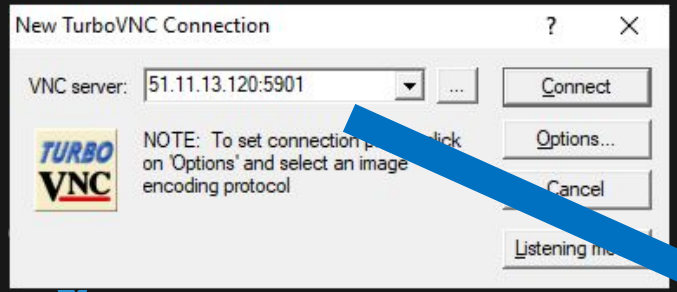
Starting a graphical desktop on a cluster

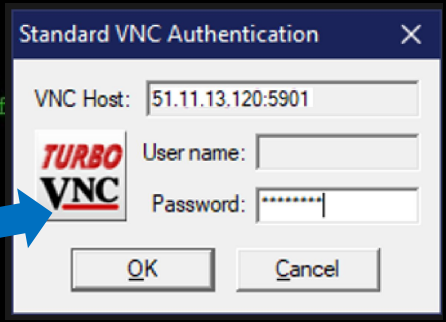
openflightHPC
Next Generation HPC on Any Platform

 # flight desktop start gnome

Start your local VNC client

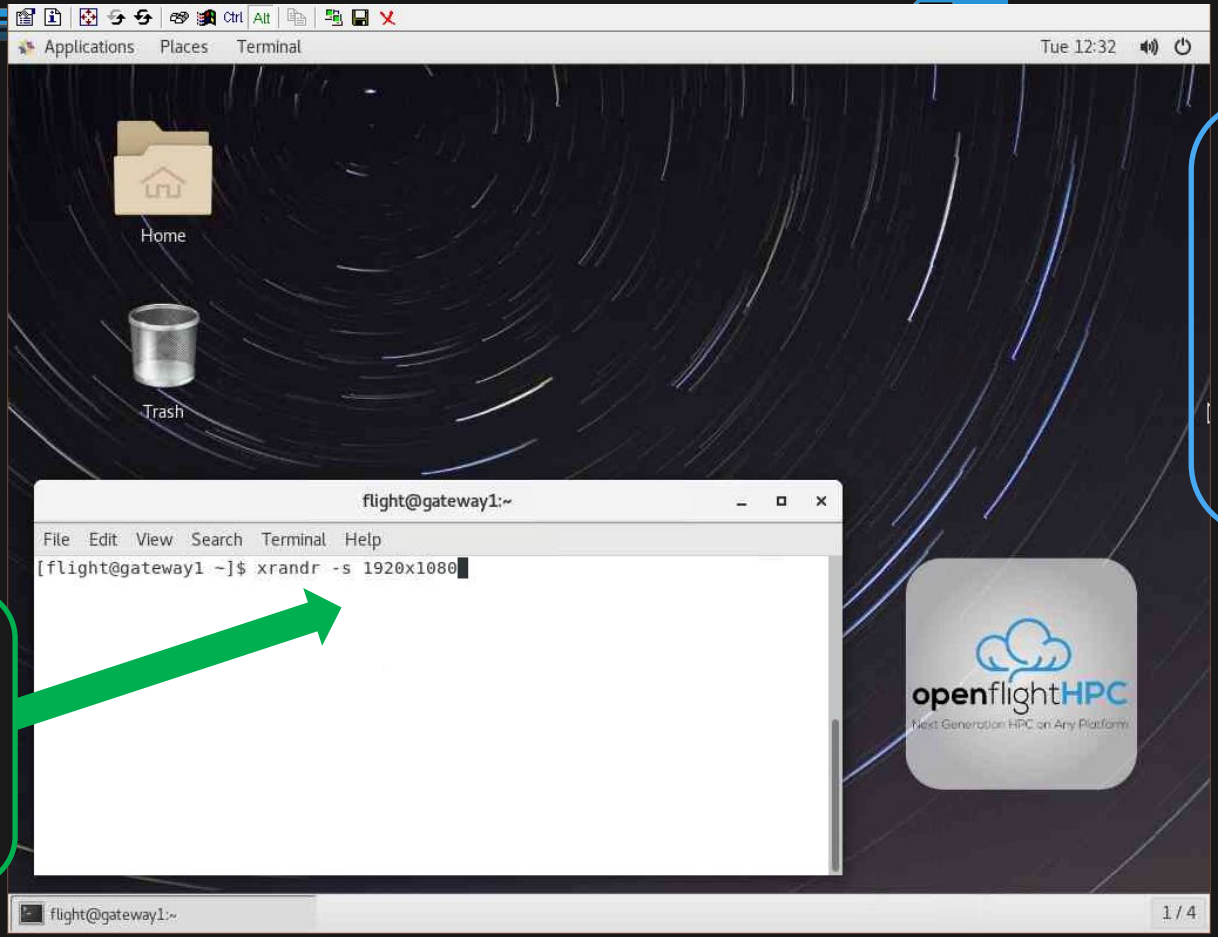
```
t@gateway1 (wildemo7) ~]$ flight desktop start gnome
Starting a 'gnome' desktop session:

   > ☑ Starting session

A 'gnome' desktop session has been started.

== Session details ==

  Identity: df0181a3-a800-473c-81ae-809442f
      Type: gnome
   Host IP: 51.11.13.120
  Hostname: gateway1
      Port: 5901
   Display: :1
  Password: Kk7qLu6s

This desktop session is directly accessible from the public internet.
Depending on your client and network configuration you may be able to
directly connect to the session using:

  vnc://flight:Kk7qLu6s@51.11.13.120:5901
  51.11.13.120:5901
  51.11.13.120:1
```





alcesflight

Starting a graphical desktop



Hardware accelerated GPU rendering also available with compatible platform

Select a higher resolution if desired

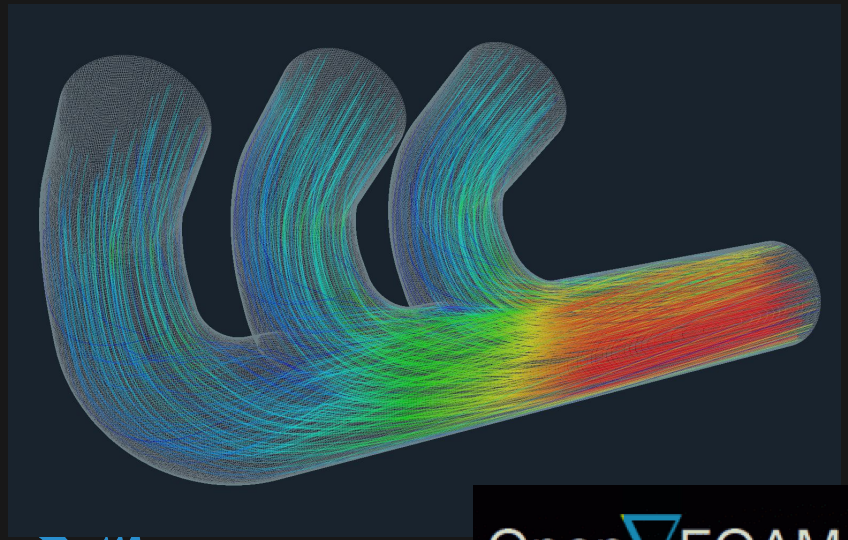
Getting help to install an application

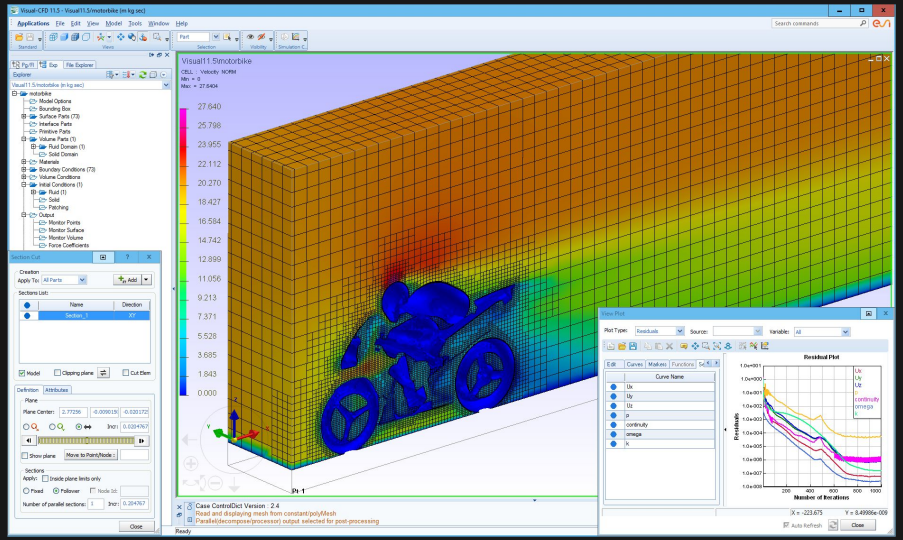
- Ask your cluster administrator
  - Typically for a hardware-based service (e.g. Barkla)
- For your personal HPC cluster
  - Use an application management environment
    - Conda
    - Easybuild
    - Gridware
    - Spack
  - Use a container service
    - Singularity (imports Docker modules)

Computational fluid dynamics with OpenFoam

openflightHPC
Next Generation HPC on Any Platform

- Popular engineering application
- Simulates fluid flow around objects

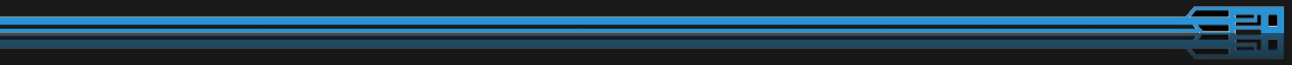




OpenFOAM

alcesflight

OpenFoam



- Distributed as source code
- Add-in toolboxes to support different functions
- Uses MPI for parallelism
- Most workflows are a multi-stage process
    - Interactive design stage to setup the problem
    - Decomposition of the problem into manageable pieces
    - Parallel processing of the decomposed mesh
    - Recombination of the parts into a single result
    - Visualisation of the output
- Relatively complex to install

Installing applications with Gridware

```
# flight env create gridware
# flight env activate gridware
# gridware search openfoam
# gridware install apps/openfoam/4.1
```

Search limited to binary install versions by default

Press “y” to install deps

- *Flight will automatically detect and install software dependencies using binary downloads where possible*
- *Software is installed centrally and available on all nodes*
- *Compilation on demand is also possible if required*
- *Wait a few minutes for this stage to complete…*

Why?

Running an OpenFoam job

openflightHPC
Next Generation HPC on Any Platform

```
# module load apps/openfoam
# icoFoam –help
# nano myfoamjob.sh
```

Enter this content

```
#!/bin/bash
#SBATCH -N 1
flight env activate gridware
module load apps/openfoam
cp -r $FOAM_TUTORIALS/incompressible/icoFoam/cavity/cavity $HOME/.
cd $HOME/cavity
blockMesh
checkMesh
icoFoam
```

Press CTRL+X to exit and enter “y” to save

Or – if that’s too much typing:

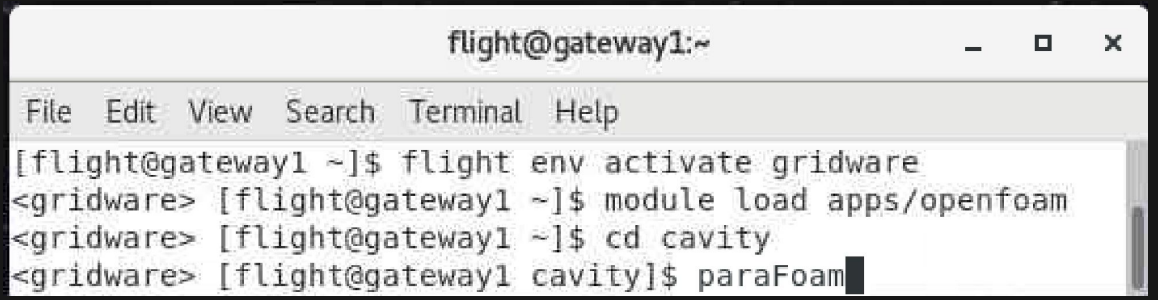
```
# curl -L http://tiny.cc/foamjob > myfoamjob.sh
```

Submitting the OpenFoam job to run

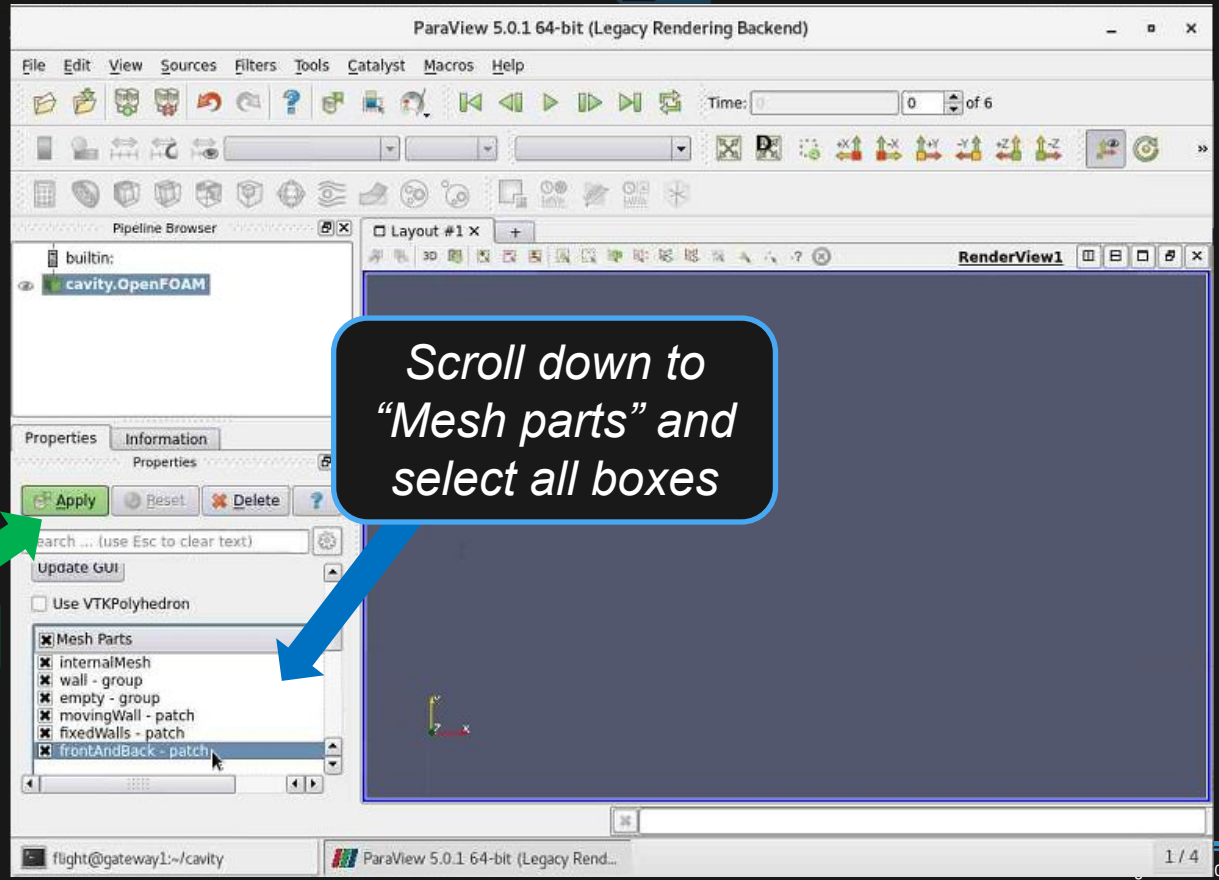
```
# sbatch myfoamjob.sh
# squeue
```

Job should execute in 1-2 minutes on cluster compute nodes

Start a new desktop terminal and launch GUI



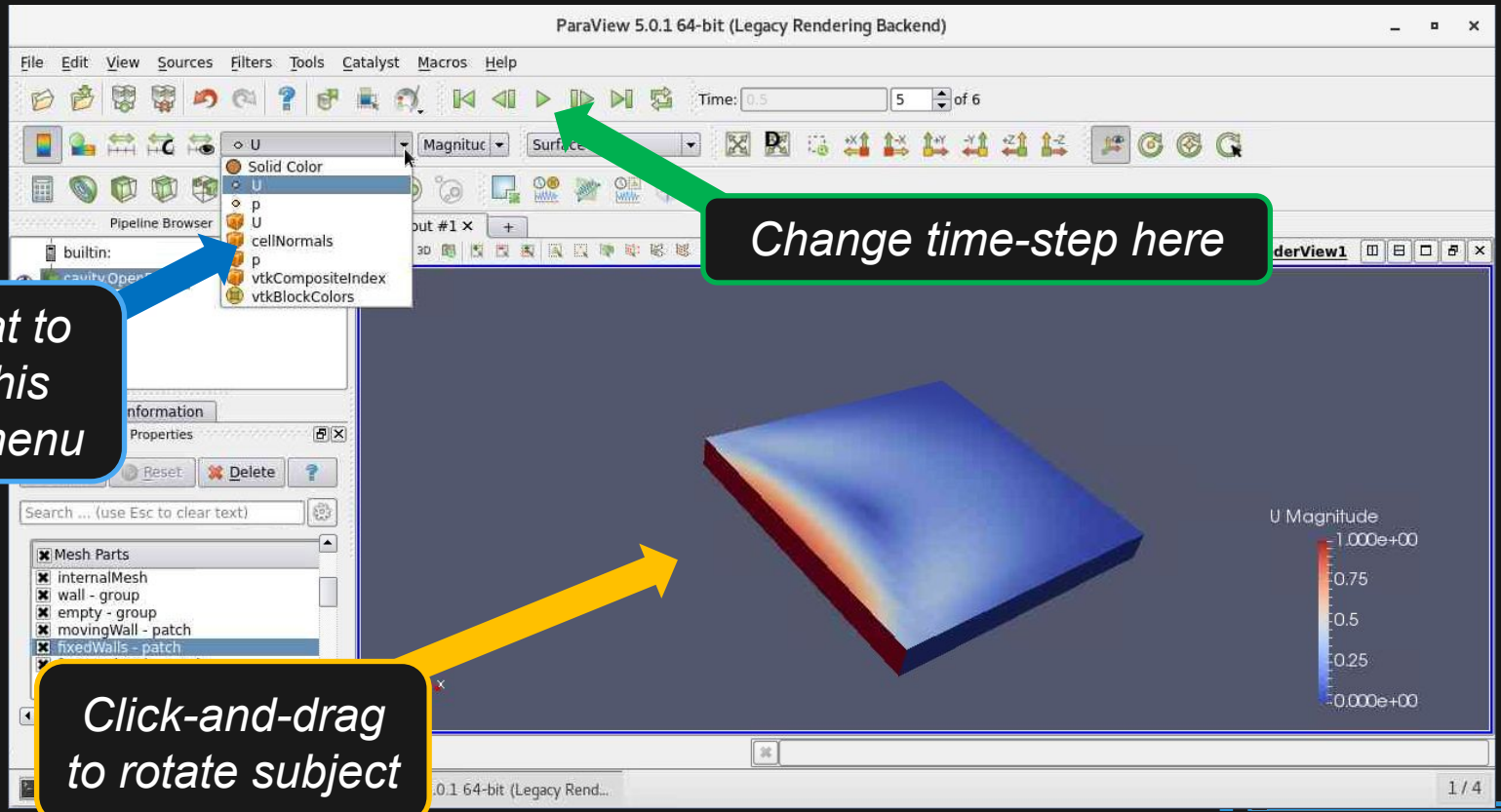
Navigating the GUI



Scroll down to "Mesh parts" and select all boxes

Click Apply when done

CFD visualization

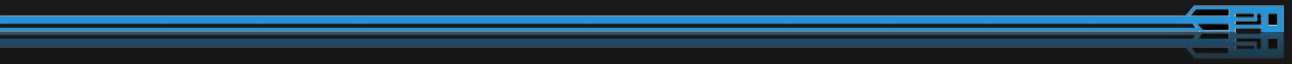


Change time-step here

Choose what to display in this drop-down menu

Click-and-drag to rotate subject

Parallel computing jobs on an HPC cluster



- A parallel job uses resources from multiple nodes
  - CPU cores
  - Shared or distributed memory
  - Special devices (e.g. GPU, FPGA, etc.)
- Most parallel jobs use an API
  - Message-passing interface (MPI)
  - Provides simple routines for running code in parallel
- Compile your application with relevant libraries
  - MPIs: OpenMPI, Intel MPI, MPICH, MVAPICH
  - Threading: OpenMP, Intel TBB, SHMEM

Writing some parallel C code

openflightHPC
Next Generation HPC on Any Platform

```
#include <stdio.h>
#include <mpi.h>
#include <time.h>
#include <string.h>
    int main(int argc, char **argv) {
        char    name[MPI_MAX_PROCESSOR_NAME];
        int     nprocs, procno, len;
        MPI_Init( &argc, &argv );
        MPI_Comm_size( MPI_COMM_WORLD, &nprocs );
        MPI_Comm_rank( MPI_COMM_WORLD, &procno );
        MPI_Get_processor_name( name, &len );
        name[len] = '\0';
        time_t lt;
        lt = time(NULL);
        printf( "Hello !! from %s@%d/%d on %s\n", name, procno, nprocs, ctime(&lt));
        MPI_Barrier( MPI_COMM_WORLD );
        MPI_Finalize();
        return( 0 );
    }
```

```
# module list
# which mpirun
# nano mympicode.c
```

Enter this content

Or – if that's too much typing:

```
# curl -L http://tiny.cc/mpiexample > mympicode.c
```

Compile your parallel code and submit

```
# module list
# which mpicc
# mpicc -o mympiapp -O3 mympicode.c
# ldd mympiapp
# mpirun -np 4 ./mympiapp
# nano mympijob.sh
```

Note: this is a capital "o" for optimisation (not zero)

Enter this content

```
#!/bin/bash
#SBATCH -N 2
flight env activate gridware
module load
mpi/openmpi/1.10.2
mpirun ./mympiapp
```

Press CTRL+X to exit and enter "y" to save

```
# sbatch mympijob.sh
Submitted batch job 4
# cat slurm-4.out
```

Use this number here

More application environments

```
# flight env deactivate
# flight env create conda
# flight env activate conda
# conda install perl
# which perl
# nano myperlscript.pl
```

Press “y” to confirm

Enter this content

```
use strict;
use warnings;
print "hi NAME\n";
```

Press CTRL+X to exit and enter “y” to save

Conda application environment

```
# perl myperlscript.pl
# nano myperljob.sh
```

```
#!/bin/bash -l
#SBATCH -N 1
echo Hello from $HOSTNAME
flight env activate conda
which perl
perl myperlscript.pl
```

Enter this content

Press CTRL+X to exit and enter "y" to save

```
# sbatch myperljob.sh
Submitted batch job 5
# cat slurm-5.out
```

Use this number here

Installing and running a web-service

openflightHPC
Next Generation HPC on Any Platform

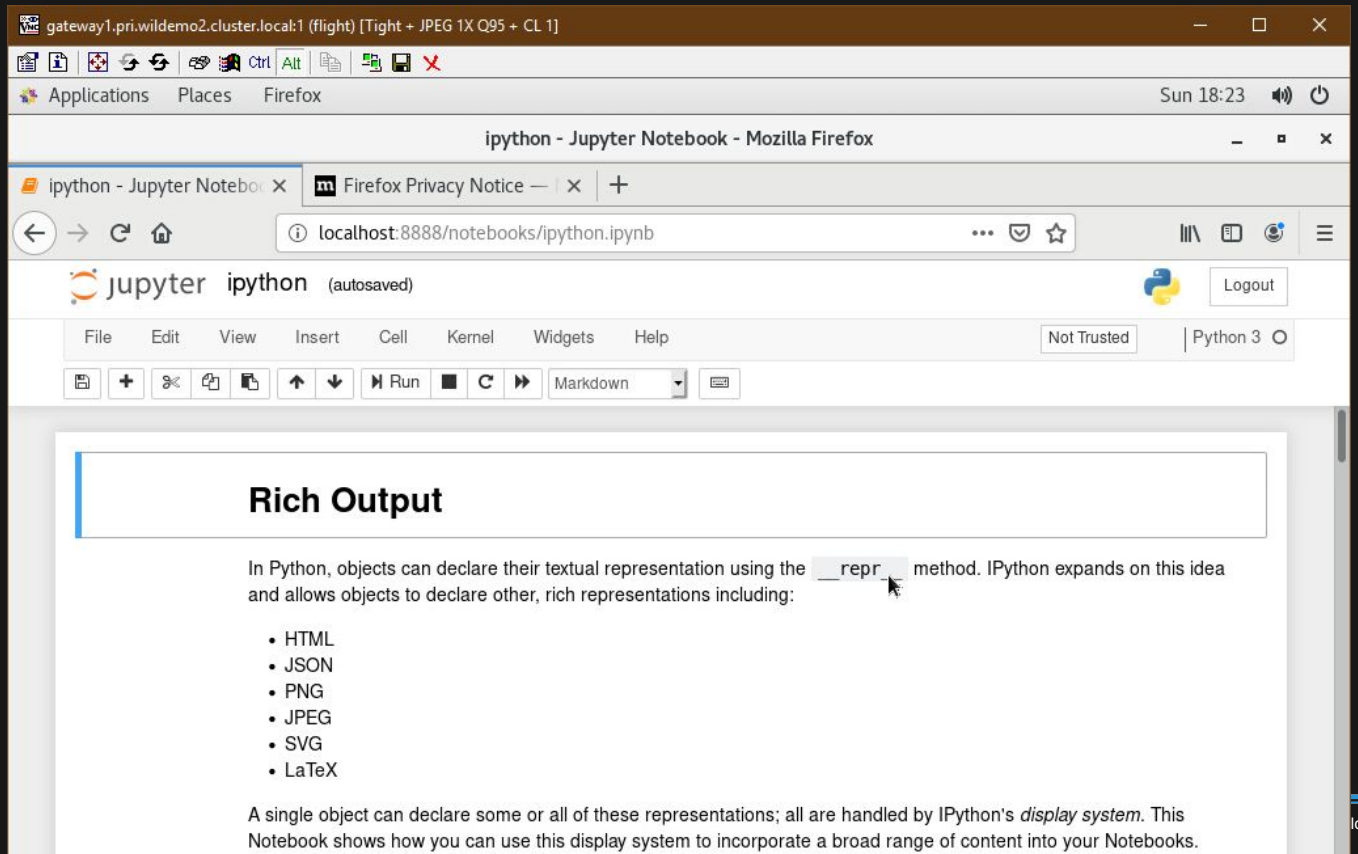
Start a new desktop terminal and enter:

flight@gateway1:~

File Edit View Search Terminal Help

```
[flight@gateway1 ~]$ flight env deactivate
[flight@gateway1 ~]$ flight env activate conda
(base) <conda> [flight@gateway1 ~]$ conda install jupyter
(base) <conda> [flight@gateway1 ~]$ jupyter --version
jupyter core     : 4.6.1
jupyter-notebook : 6.0.3
qtconsole        : 4.6.0
ipython          : 7.12.0
ipykernel        : 5.1.4
jupyter client   : 5.3.4
jupyter lab      : not installed
nbconvert        : 5.6.1
ipywidgets       : 7.5.1
nbformat         : 5.0.4
traitlets        : 4.3.3
(base) <conda> [flight@gateway1 ~]$ curl -L http://tiny.cc/jupyterexample > ipython.ipynb
(base) <conda> [flight@gateway1 ~]$ jupyter notebook ipython.ipynb
```

Jupyter Notebook browser session

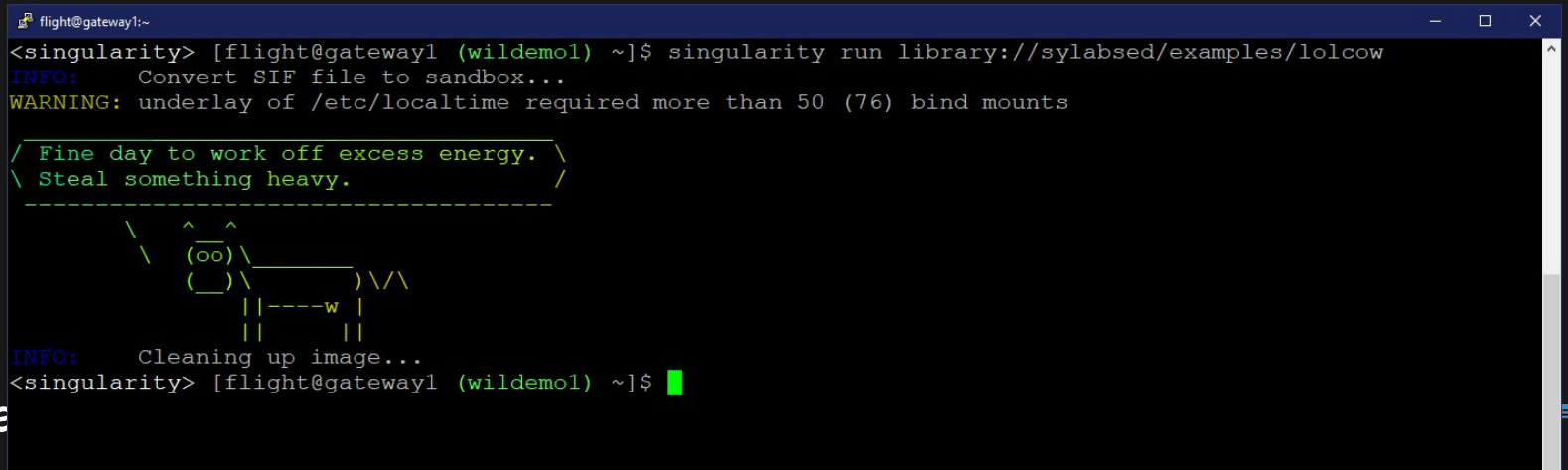


Singularity for Docker containers

openflightHPC
Next Generation HPC on Any Platform

```
# flight env deactivate
# flight env create singularity
# flight env activate singularity
# singularity run library://sylabsed/examples/lolcow
```

Close your *paraFoam* app before running



Running singularity jobs

```
# singularity pull shub://singularityhub/hello-world
# nano mydockerjob.sh
```

Enter this content

```
#!/bin/bash -l
#SBATCH -N 1 --mem=0
flight start
flight env activate singularity
singularity run hello-world_latest.sif
```

Press CTRL+X to exit and enter "y" to save

```
# sbatch mydockerjob.sh
Submitted batch job 6
# cat slurm-6.out
```

Use this number here

Spack application management

openflightHPC
Next Generation HPC on Any Platform

```
# flight env deactivate
# flight env create spack
# flight env activate spack
# spack list
# spack install bowtie
# nano mybowtiejob.sh
```

Enter this content

```
#!/bin/bash -l
#SBATCH -N 1
flight env activate spack
spack load bowtie
wget -O ecoli.fq http://tiny.cc/ecoli
bowtie-build ecoli.fq e_coli
```

Note: this is a capital "o" for output (not zero)

Press CTRL+X to exit and enter "y" to save

```
# sbatch mybowtiejob.sh
Submitted batch job 7
# cat slurm-7.out
```

Use this number here

Easybuild application management

openflightHPC
Next Generation HPC on Any Platform

```
# flight env deactivate
# flight env create easybuild
# flight env activate easybuild
# module load EasyBuild
# eb -S R
# eb R-3.5.1-foss-2018b-Python-2.7.15.eb -r
```

Robot mode – automatically resolve and install dependencies

Easybuild may take some time to complete the build, as it will pull in any dependencies needed

Once completed, applications will be available on all compute nodes of the cluster

alcesflight

Review

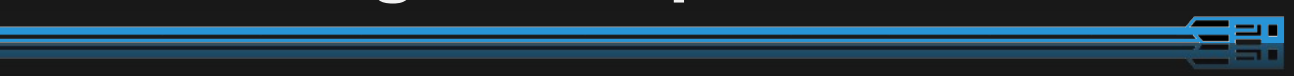
So far we have:

- Created our own applications from source
- Used various application environments to install apps
- Run jobs and visualized data using a compute cluster

What if we want to run:

- A MapReduce workflow?
- A machine-learning workflow?

Installing Hadoop



openflightHPC
Next Generation HPC on Any Platform

Note: this is a capital "o" for output (not zero)

Press "y" to confirm

```
# flight env deactivate
# sudo yum install java-1.8.0-openjdk.x86_64
# sudo yum install java-1.8.0-openjdk-devel.x86_64
# wget -O /tmp/hadoop.tgz http://tiny.cc/hadoop321
# cd /opt/apps
# tar xzf /tmp/hadoop.tgz
# cd hadoop-3.2.1
```

alcesflight

Configuring Hadoop

openflightHPC
Next Generation HPC on Any Platform

```
# nano etc/hadoop/hadoop-env.sh
```

Press CTRL+_ (underscore) and enter 54 to jump to this line

Edit line 54 so it reads:

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0.el7_7.x86_64/jre
```

Press CTRL+X to exit and enter "y" to save

Or – if that's too much typing:

```
# curl -L https://tinyurl.com/hdconfig | bash
```

alcesflight

Preparing our work

openflightHPC
Next Generation HPC on Any Platform

Note: this is a capital "o" for output (not zero)

```
# cd
# wget https://tinyurl.com/hadoopenv
# source hadoopenv
# mkdir MapReduceTutorial
# chmod 777 MapReduceTutorial
# cd MapReduceTutorial
# wget -O hdfiles.zip https://tinyurl.com/hdinput1
# unzip -j hdfiles.zip
# ls
```

This archive includes a spreadsheet of data which we will sort into sales units per region

alcesflight

Compiling java + setting manifest

```
# pwd
/home/flight/MapReduceTutorial
# javac -d . SalesMapper.java SalesCountryReducer.java
SalesCountryDriver.java
# nano Manifest.txt
```

This should be a single long line

Enter this content

Main-Class: SalesCountry.SalesCountryDriver

Press CTRL+X to exit and enter "y" to save

Create the jar file

```
# jar cfm ProductSalePerCountry.jar Manifest.txt
SalesCountry/*.class
# ls *.jar
ProductSalePerCountry.jar
```

This should be a single long line

This is our java archive file containing the code ready to run

Start Hadoop and copy in data-files

```
# $HADOOP_HOME/sbin/start-dfs.sh
# $HADOOP_HOME/sbin/start-yarn.sh
```

Distributed file system service

Resource manager, node manager, app manager

Copy in your data-file to the DFS

```
# mkdir ~/inputMapReduce
# cp SalesJan2009.csv ~/inputMapReduce/.
# $HADOOP_HOME/bin/hdfs dfs -ls ~/inputMapReduce
```

Run your JAR through Hadoop

openflightHPC
Next Generation HPC on Any Platform

Execute your MapReduce job

```
# $HADOOP_HOME/bin/hadoop jar
ProductSalePerCountry.jar ~/inputMapReduce
~/mapreduce_output_sales
```

This should be a single long line

Check that output was produced

```
# $HADOOP_HOME/bin/hdfs dfs -cat
~/mapreduce_output_sales/part-00000 | more
```

This should be a single long line

alcesflight

Running an AI workload in Tensorflow

openflightHPC
Next Generation HPC on Any Platform

This should be a single long line

```
# flight env deactivate
# flight env activate singularity
# git clone https://github.com/tensorflow/models.git
# singularity exec docker://tensorflow/tensorflow:1.15.0
python ./models/tutorials/image/mnist/convolutional.py
```

Version 1.15.0 of Tensorflow runs on CPU by default

This example model will take 5-10 minutes to prepare and run for 20-30 minutes

alcesflight

Alternatively, build in Conda

```
# flight env deactivate
# flight env activate conda
# conda create -n tensorflow python=3.6
# source activate tensorflow
# pip install tensorflow==1.15
```

Version 1.15.0 of Tensorflow runs on CPU by default

```
# git clone https://github.com/tensorflow/models.git
# python ./models/tutorials/image/mnist/convolutional.py
```

Summary

The Hybrid HPC Experience:

- Worked in a fixed cluster environment.
  - Template designed for training.
  - Easy to replicate, easy to mimic HPC clusters such as Barkla.
- Engaged with a special HPC use case using Hadoop.
  - Aimed at project-based HPC work.
  - Designed to deal with unusual or periodic requests.

openflightHPC
Next Generation HPC on Any Platform

Hybrid HPC leverages the right resources at the right time.

alcesflight

What next?

- Your cluster is active until:
- Have a project? Idea? Concept? Talk to us!

Cristin Merritt: Cristin.Merritt@alces-flight.com
Wil Mayers: Wil.Mayers@alces-flight.com

alcesflight