The new cyclone

A major update on the GFI computing infrastructure

Overview

Introduction and motivation

Who are the users?

Description of the system:

1. compute server
2. storage
3. files/software

Using GPUs

What do you need to do (transition)?

Reanalysis data

2006

ERA-Interim reanalysis
60 levels
0.75 x 0.75 degree grid
3-hourly files

2016

ERA5 reanalysis
137 levels
0.28x0.28 degree grid
1-hourly files

Storage demand increased by a factor of 2 x 6 x 3 = 36

This has consequences for post-processing !

Intended users

BSc students (coursework)
MSc students (coursework, thesis)
PhD students
researchers

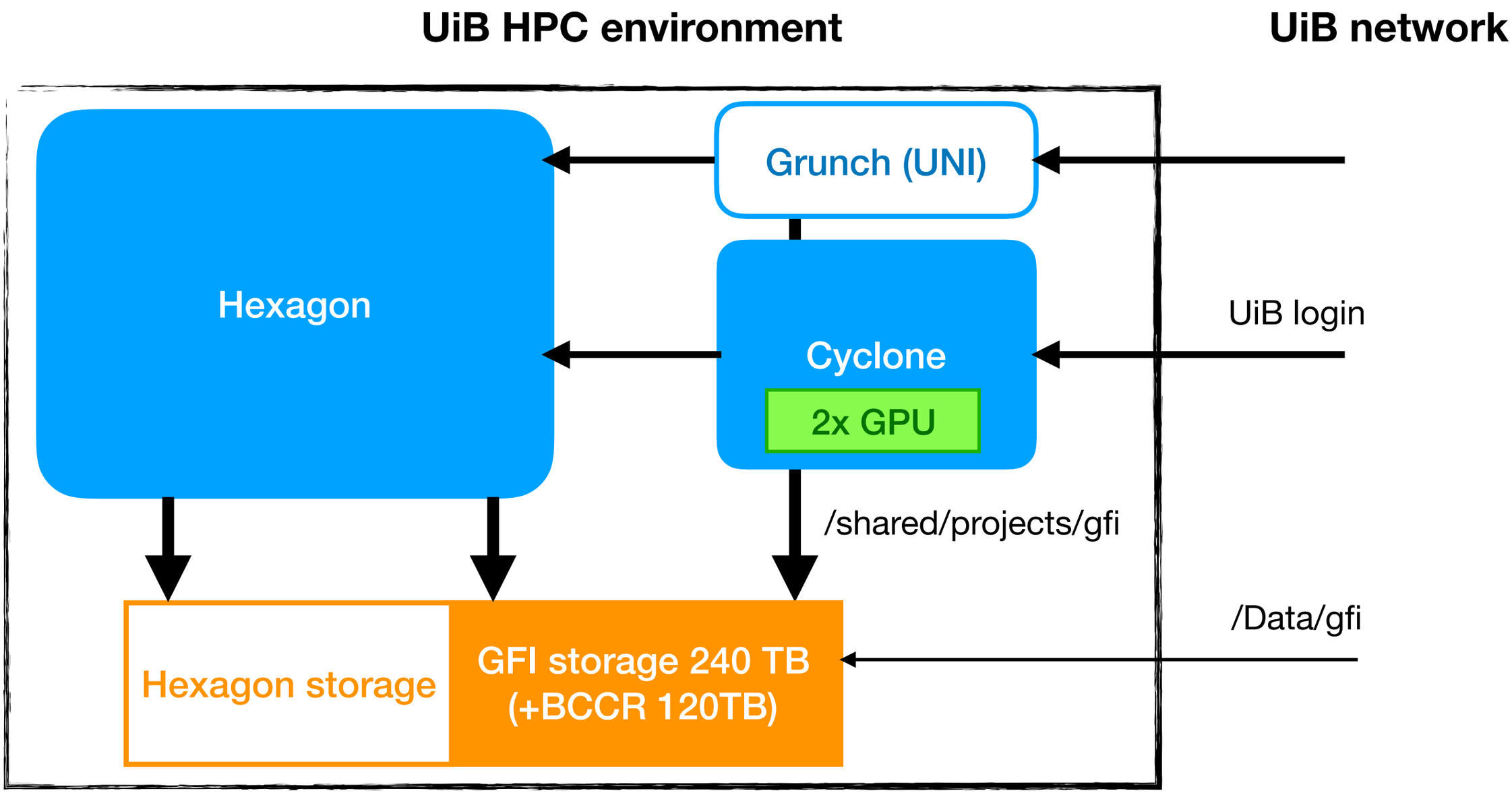
interactive use
batch use

Routine data processing (observations, forecasts)

Requirements:

low threshold
interactive use
safe (against unintentional overuse)
optimized for serial I/O (post-processing)
low parallelisation jobs (single node)
lifetime of 5-7 years

Structure of High Performance Computing environment at UiB



Cyclone

Computing overview

Cyclone is a powerful compute node:

Dell PowerEdge R740 Server configuration with
Intel Xeon Gold 6140M, 18 cores, 36 threads, 2.3GHz, 25MB L3 cache
1.5 TB DDR4-2666 memory
2 NVIDIA Tesla GPU, 12 GB

You need to recompile your code for the new operating system:

CentOS linux operating system

It is easy, flexible and safe to use:

There is no queue system on cyclone
Users can submit to the hexagon queue from cyclone (acts as *login node*)
There is a 30-50% resource limitation per user
Activate software packages based on *module* command
Access outside via current paths, /Data/gfi
On cyclone mounted at /share/projects/gfi
Same/additional software packages available (your suggestions?)
Maintained by the HPC group (!)

Cyclone

Graphics Processing Unit (GPU)

GPUs are massively parallel computers

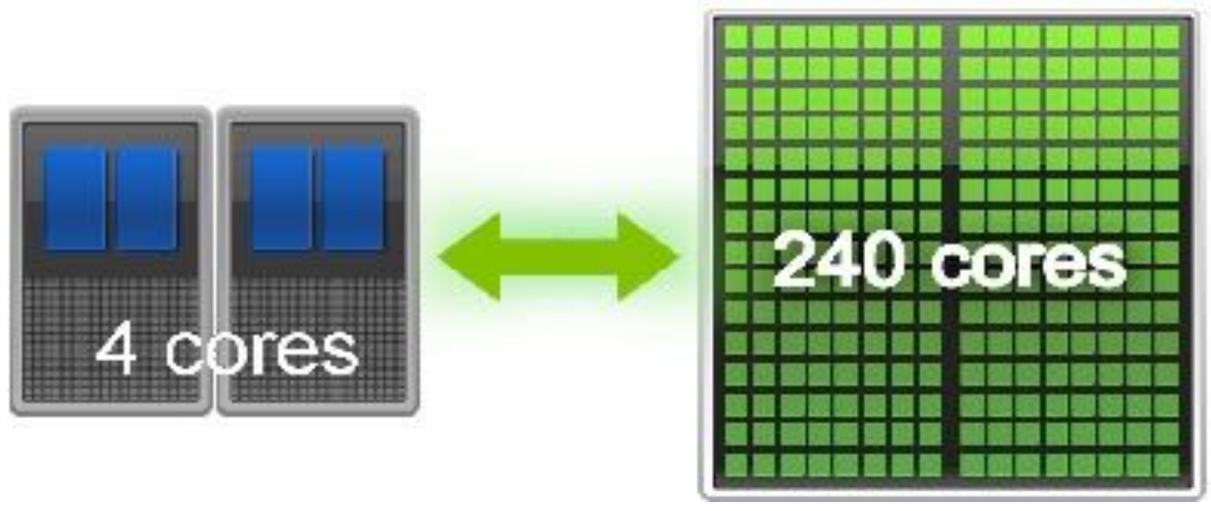
each Tesla P100 has 3584 cores
programming using CUDA, DirectCompute, OpenCL, OpenACC
some libraries are already optimized for GPU usage
first optimisation step are OpenACC compiler statements
bottleneck is to get data on/off the GPU
memory of 12 GB per GPU
both GPUs can also be used in combination

Theoretical performance (per GPU)

Double-Precision Performance	4.7 teraFLOPS
Single-Precision Performance	9.3 teraFLOPS
Half-Precision Performance	18.7 teraFLOPS

For comparison: Xeon Gold 6140: 2.5 teraFLOPS

GPU programming



- GPU – Graphics Processing Unit
  - Development driven by huge graphical demands for computer games
  - High performance, low power consumption
  - Challenging to program because of separate GPU/CPU memory
  - GPU technology to be more easily useable on modern HPCs
- Suitable for massively parallel applications
- Shared memory improves performance for bandwidth-limited applications
- support built into Matlab etc.
- CUDA support in python
- See http://gpgpu.org/developer for more information

```
46
47  static const char* kernelSourceVector =
48    "__kernel void saxpy_parallel\n"                                          \
49    "  (const int n, const float alpha, __global float4 *x, __global float4 *y)\n" \
50    "{\n"                                                                     \
51    "  // get the global thread id\n"                                         \
52    "  uint i = get_global_id(0);\n"                                          \
53    "  // except for special cases, the total number of threads\n"            \
54    "  // adds up to more than the vector length n, so this conditional is\n" \
55    "  // EXTREMELY important to avoid writing past the allocated memory for\n" \
56    "  // the vector y.\n"                                                     \
57    "  if (i<n)\n"                                                            \
58    "    y[i] = alpha*x[i] + y[i];\n"                                         \
59    "}\n";
60
61
62  ////////////////////////////////////
63  // main routine
64  ////////////////////////////////////
65  int main (int argc, char **argv)
66  {
67    ////////////////////////////////////
68    // (0) Preprocessing
69    ////////////////////////////////////
70
71    // Get list of all platforms and their devices from our utility function.
72    oclinfo        *oi = getAllDevices();
73    cl_platform_id myPlatform = NULL;
74    cl_device_id   myDevice = NULL;
75
76    // Pick hard-coded target platform and device (very brute-force)
77    //myPlatform = oi->allPlatforms[1];
78    //myDevice   = oi->allDevices[1][0];
79
80    // Alternatively, we could use the first device matching the target hardware
81    // we aim at. OpenCL provides a set of identifiers for certain device types:
82    // CL_DEVICE_TYPE_DEFAULT
83    // CL_DEVICE_TYPE_CPU
84    // CL_DEVICE_TYPE_GPU
```

GPU "Kernel" in OpenCL

GFI storage 240TB
(+BCCR 120TB)

Storage for GFI

Available storage

240 TB accessible from Hexagon and cyclone (very fast)
120 TB of those accessible from outside (slow connection, NFS)
BCCR has an additional 120TB unit
Storage can be expanded in 120 TB units for 150 kNOK each (by projects etc.)

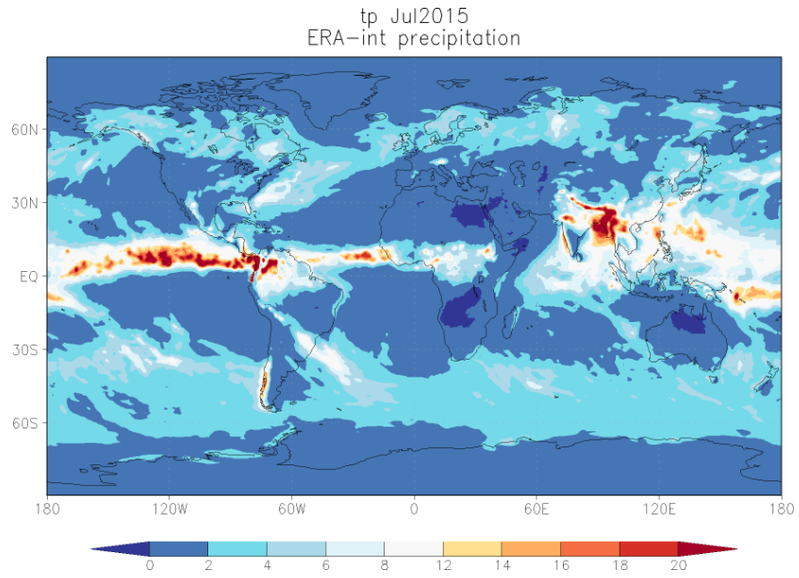
Allocation to datasets, projects, users

We will host large datasets on the disks and provide workspace to projects and users
The ERA5 reanalysis dataset will be archived locally (format currently under investigation)
Allocation will be based on actual demand, (re-)negociateable

Common scratch space /work with 21 day deletion time
Fast I/O from cyclone and Hexagon (for serial I/O jobs)

Hexagon users already have 10 TB

NIRD is working on making similar solutions available



Transition to the new cyclone

- The new cyclone will be online from tomorrow (29.6.2018)
- skd-cyclone will continue working into August
- data will be locked (write protected) when copied, then deleted
- plans for switchover 6th August
- old backups will be deleted after a first new backup has been made
- normal UiB username/password

- you need to recompile all model code (because of libraries etc.)

- subscribe to linux@gfi.uib.no to receive maintenance notes
- learn about available modules using the command 'module list'
- learn how to accelerate code with GPU
- learn how to submit code to hexagon queue

You can reach the new machine at cyclone.hpc.uib.no