

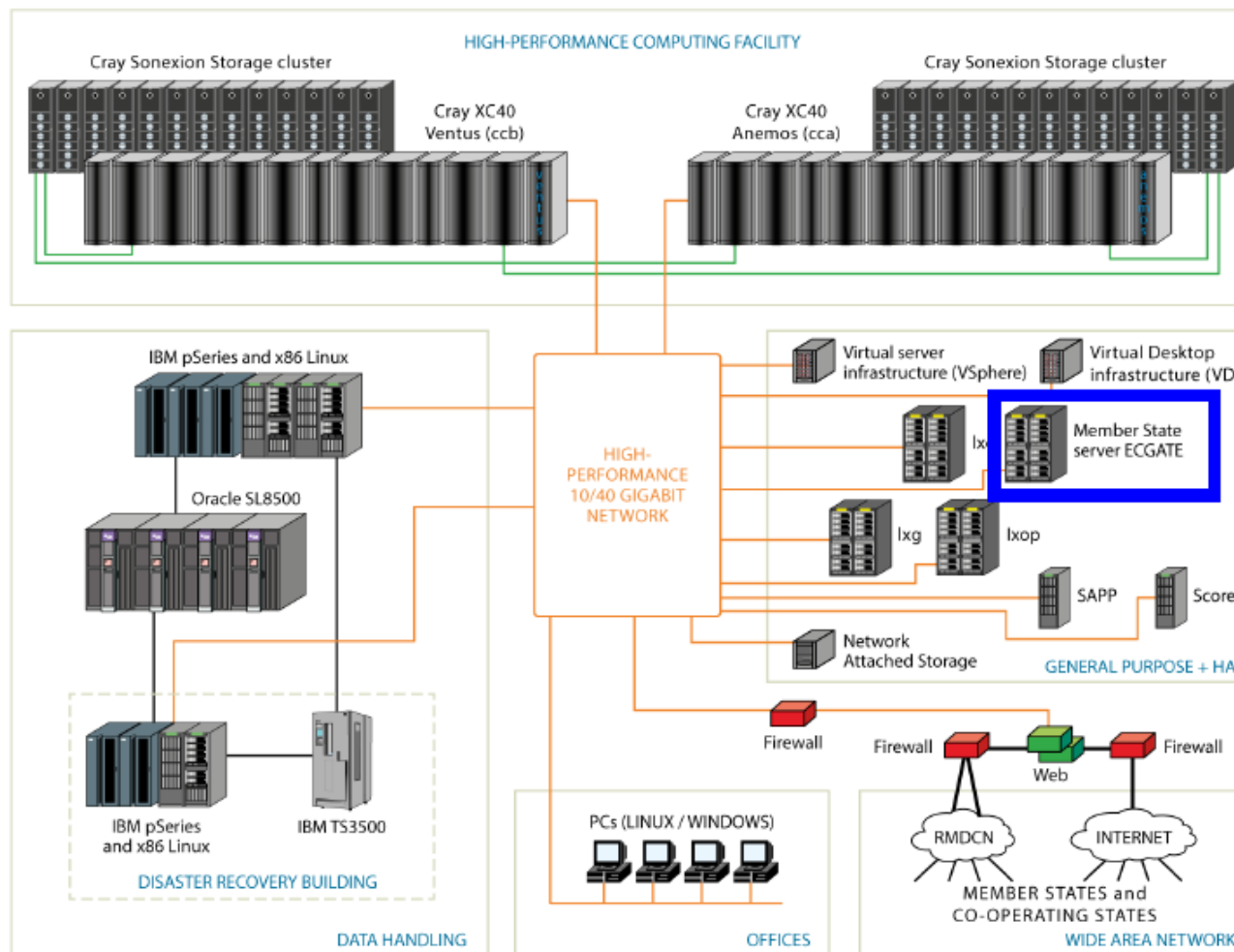
# ECMWF Facilities

**ALADIN/HIRLAM common data assimilation training week**  
The Hungarian Meteorological Service (OMSZ)  
Budapest, Hungary

10-15 February 2019

# Linux cluster –ecgate

- Web documentation: [www.ecmwf.int/en/computing/our-facilities/ecgate](http://www.ecmwf.int/en/computing/our-facilities/ecgate)



## ecgate –configuration

- 12 compute nodes each with
  - 2 Intel Xeonprocessors(Sandy Bridge-EP): 16 core at 2.7 GHz
  - Hyper threading provides32 virtual CPUs per node
  - 128 GB memory
  - 2 x 900 GB SAS HDD
- One (+one as backup) node used as a "login" node
- RedHatEnterprise Linux Server 6.8
- 6I/O server nodes
  - Provides ~275 TB raw disk space (~200 TB of usable space)
  - All file systems are GPFS (General Parallel File Systems)
  - File systems use RAID 5 for speed and resilience
- Available to ~3000 users at more than 350 institutions





## Linux cluster Login – ecgate

- ssh -Y [uid@ecaccess.ecmwf.int](mailto:uid@ecaccess.ecmwf.int)

NX

- ssh -YC sbu@ecaccess.ecmwf.int -oKexAlgorithms=+diffie-hellman-group1-sha1 -oHostKeyAlgorithms=+ssh-dss

## ecgate – purpose

### Time-critical applications

- Option 1
- Option 2

### Batch submission

- SLURM
- ECaccess Tools

### Program development

### Visualisation

- Metview
- Magics

### Data transfer

- ftp / sftp
- ectrans

### Access to archives

- MARS
- ECFS



## Interactive vs Batch

- When you login, the default shell on **ecgate** is either Bash, Korn-shell (ksh), or the C-shell (csh).
- To run a script or a program **interactively**, enter the executable name and any necessary arguments at the system prompt.
- You can also run your job in **background** so that other commands can be executed at the same time...

```
$> ./your-program arg1 arg2  
$> ./your-program arg1 arg2 &
```

## Interactive vs Batch

- But...

**Background is **not** batch**

- The program is still running interactively on the login node
  - You share the node with the rest of the users
- The limits for interactive sessions still apply:
  - CPU time limit of 30 min per process

```
$> ulimit -a
```

- **Interactive sessions should be limited to development tasks, editing files, compilation or very small tests**



## Interactive vs Batch

- But...

**Background is **not** batch**

- The program is still running interactively on the login node
  - You share the node with the rest of the users
- The limits for interactive sessions still apply:
  - CPU time limit of 30 min per process

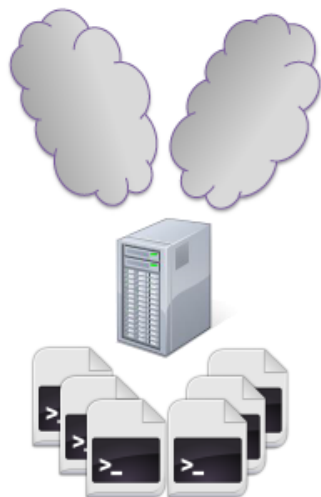
```
$> ulimit -a
```

- **Interactive sessions should be limited to development tasks, editing files, compilation or very small tests**

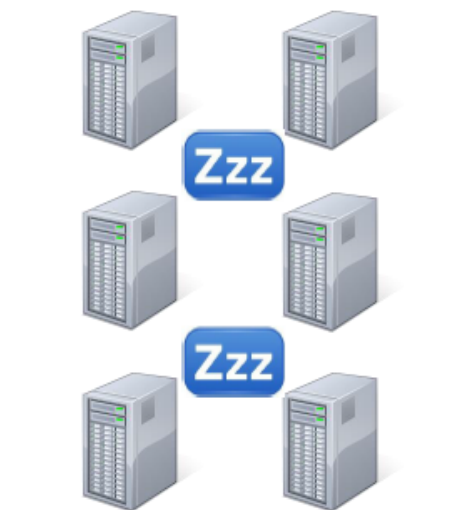
## Interactive

vs

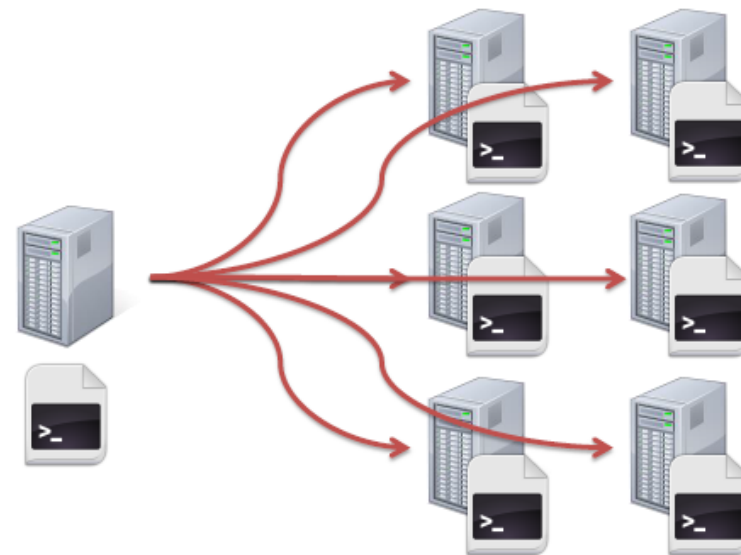
## Batch



Login node



Computing (batch) nodes

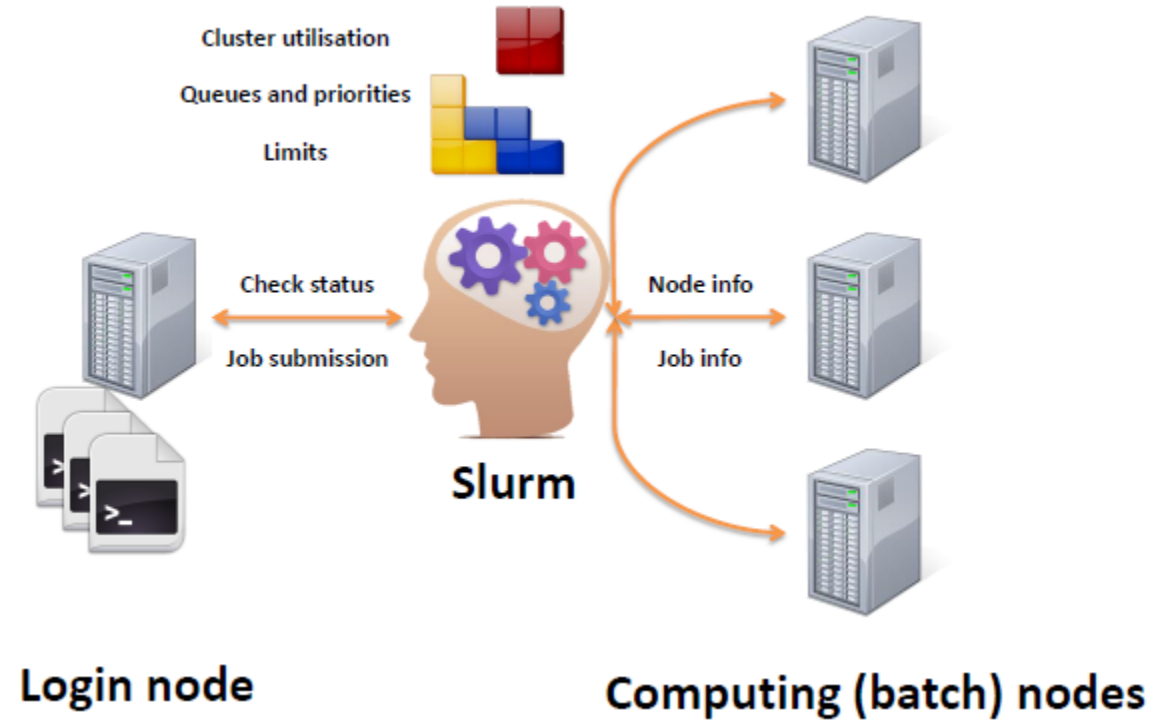


Login node

Computing (batch) nodes

## Batch on ecgate

- **Slurm:** Cluster workload manager:
  - Framework to execute and monitor batch work
  - Resource allocation (where?)
  - Scheduling (when?)
- **Batch job:** shell script that will run unattended, with some special directives describing the job itself



## Quality of service (queues)

- In Slurm, QoS(Quality of Service) = **queue**
- The queues have an associated priority and have certain limits
- Standard queues available to all users

QoS	Description	Priority	Wall Time Limit	Total Jobs	User Jobs
express	Suitable for short jobs	400	3 hours	256	32
normal	Suitable for most of the work. This is the default	300	1 day	256	32
long	Suitable for long jobs	200	7 days	32	4

## Job directives

Directive	Description	Default
<b>--job-name=...</b>	A descriptive name for the job	Script name
<b>--output=...</b>	Path to the file where standard output is redirected. Special placeholders for job id ( %j ) and the execution node ( %N )	slurm-%j.out
<b>--error=...</b>	Path to the file where standard error is redirected. Special placeholders for job id ( %j ) and the execution node ( %N )	output value
<b>--workdir=...</b>	Working directory of the job. The output and error files can be defined relative to this directory.	submitting dir
<b>--qos=...</b>	Quality of service (queue) where the job is to be submitted	normal*
<b>--time=...</b>	Wall clock limit of the job (not cpu time limit!) Format: m, m:s, h:m:s, d-h, d-h:m or d-h:m:s	qos default
<b>--mail-type=...</b>	Notify user by email when certain event types occur. Valid type values are BEGIN, END, FAIL, REQUEUE, and ALL	disabled
<b>--mail-user=...</b>	Email address to send the email	submit user
<b>--hold</b>	Submit the job in held state. It won't run until released with scontrol release <jobid>	not used

## Submitting jobs: sbatch

- sbatch: Submits a job to the system. Job is configured:
  - including the directives in the job script
  - using the same directives as command line options
- The job to be submitted can be specified:
  - As an argument of sbatch
  - If no script is passed as an argument, sbatch will read the job from standard input

```
$> sbatch hello.sh
Submitted batch job 1250968
$> cat hello-1250968.out
Hello world!
$>
```

- The corresponding job id will be returned if successful, or an error if the job could not be submitted

## Checking the queue: squeue

- **squeue**: displays some information about the jobs currently running or waiting
- By default it shows all jobs from all users, but some filtering options are possible:
  - u <comma separated list of users>
  - q <comma separated list of QoSs>
  - n <comma separated list of job names>
  - j <comma separated list of job ids>
  - t <comma separated list of job states>

```
$> squeue -u $USER
JOBID      NAME      USER      QOS      STATE      TIME  TIMELIMIT  NODELIST (REASON)
1250968  helloworld  usxa      express  RUNNING    0:08      5:00      ecgb07
```

## Canceling a job: scanceloptions

- The most common usage of **scancel** is:

```
$> scancel <jobid1> <jobid2> <jobid3>
```

Option	Description
<b>-n &lt;jobname&gt;</b>	Cancel all the jobs with the specified job name
<b>-t &lt;state&gt;</b>	Cancel all the jobs that are in the specified state (PENDING/RUNNING)
<b>-q &lt;qos&gt;</b>	Cancel only jobs on the specified QoS
<b>-u \$USER</b>	Cancel ALL the jobs of the current user. Use carefully!
<b>-i</b>	Interactive option: ask for confirmation before cancelling jobs
<b>-v</b>	Verbose option. It will show what is being done

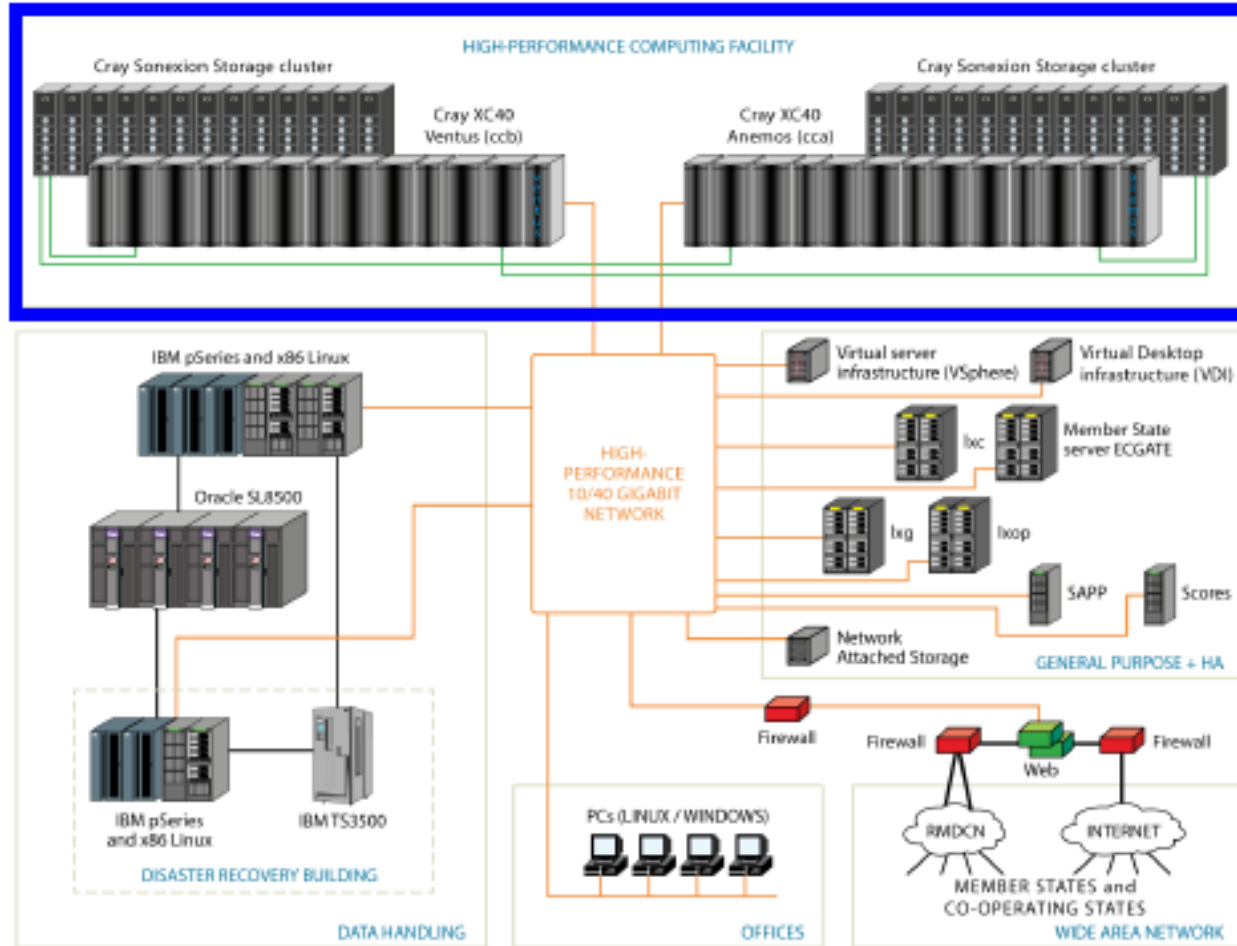
**Note: An ordinary user can only cancel their own jobs**



# HPCF

- Web documentation:

[www.ecmwf.int/en/computing/our-facilities/supercomputer](http://www.ecmwf.int/en/computing/our-facilities/supercomputer)



## HPCF – Cray XC40



- Operational at ECMWF since 19 September 2014
- Phase 2: ~3600 nodes / 130,000 compute cores per cluster
- ~8.5 petaflops peak and ~320 teraflops sustained performance
- Numbers 23 and 24 in the November 2016 Top 500 Supercomputers list
- Contract with Cray extended to 30 September 2020

## HPFC – purpose

### Batch submission

- PBSpro
- ECaccess Tools

### Time-critical applications

- Option 1
- Option 2
- Option 3

### Access to archives

- MARS
- ECFS

### Data transfer

- ftp / sftp
- ectrans



### Running meteorological models

- Member State models
- ECMWF's IFS

## Access to Cray HPCF

- **Interactive:** The Cray HPCF systems are called cca (and ccb).  
Access to the Crays from outside ECMWF is through ECaccess (ssh or NX).  
Within ECMWF, access is through ssh.  
ssh host based authentication is in place. You should not set up your own ssh keys.
- **Batch access:** Directly on Cray HPCF.  
From ECaccess web interface and ECaccess WebToolkit or ECtools. No new version required.  
'Interactive batch' access available with 'qsub -I'. Useful for testing/debugging:

**qsub -I -q ns**

qsub: waiting for job 6684896.sdb to start

## File systems on Cray HPCs

Filesystem	Quota	File quota (k)	Snapshots	Backup	Select/Delete	Type
\$HOME	480MB	20	yes	yes	no	NAS (NFS)
\$PERM	26GB	100	yes	no	no	NAS (NFS)
\$SCRATCH	26TB	1000	no	no	yes	Lustre

- **Check quotas:** **ecquota**
- **Cross mounts:**
  - On ecgate: /hpc\$HOME and /hpc\$PERM.
  - On cca (interactive node only): /ws\$HOME and /ws\$SCRATCH.
- You can configure the stripe settings in the Lustre file system.
- Specialised file systems have been set up for some projects and for Time Critical option 2 activities.

## The Cray batch service: PBSpro

- Directives in batch jobs start with **#PBS**.
- Main User commands:

User Commands	PBSpro
Job submission	qsub <script>
Job cancel	qdel <job_id>
Job status	qstat [job_id]
Queue list	qstat -Q [-f] [queue]
Job status	qscan
See job output file	qcat

- 3 main types of nodes on cca: Pre/post-Processing nodes (PPN), MOM nodes and Computational nodes (CN).

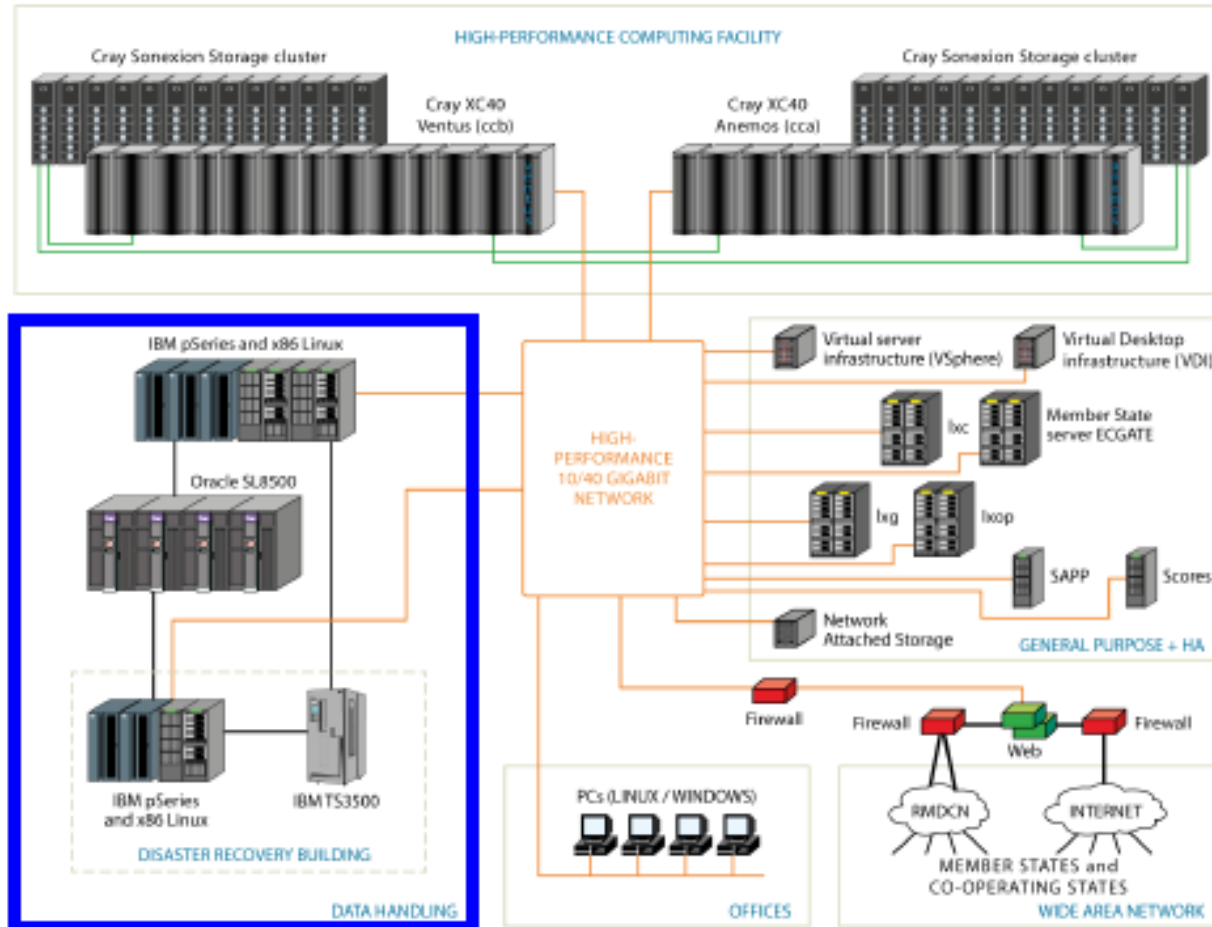
## PBSpro: Batch queues on ECMWF Cray HPCs

- Current User Limits:
  - 20 jobs per queue
  - 1GB per process.
  - 24 hours walltime (48 hours if no limit given)
  
- ‘qstat -Q -f <queue\_name>’ gives full details on specified queue.

User Queue Name	Suitable for	Target nodes	Number of processes (min/ max)	Shared/not Shared	Processes per node available for user jobs
ns	serial	PPN	1/1	shared	48
nf	fractional	PPN	2/24	shared	48
np	parallel	MOM+CN	1/48	CN not shared	48

# Data Handling System (DHS)

- Web documentation: [www.ecmwf.int/en/computing/our-facilities/data-handling-system](http://www.ecmwf.int/en/computing/our-facilities/data-handling-system)





## DHS - configuration

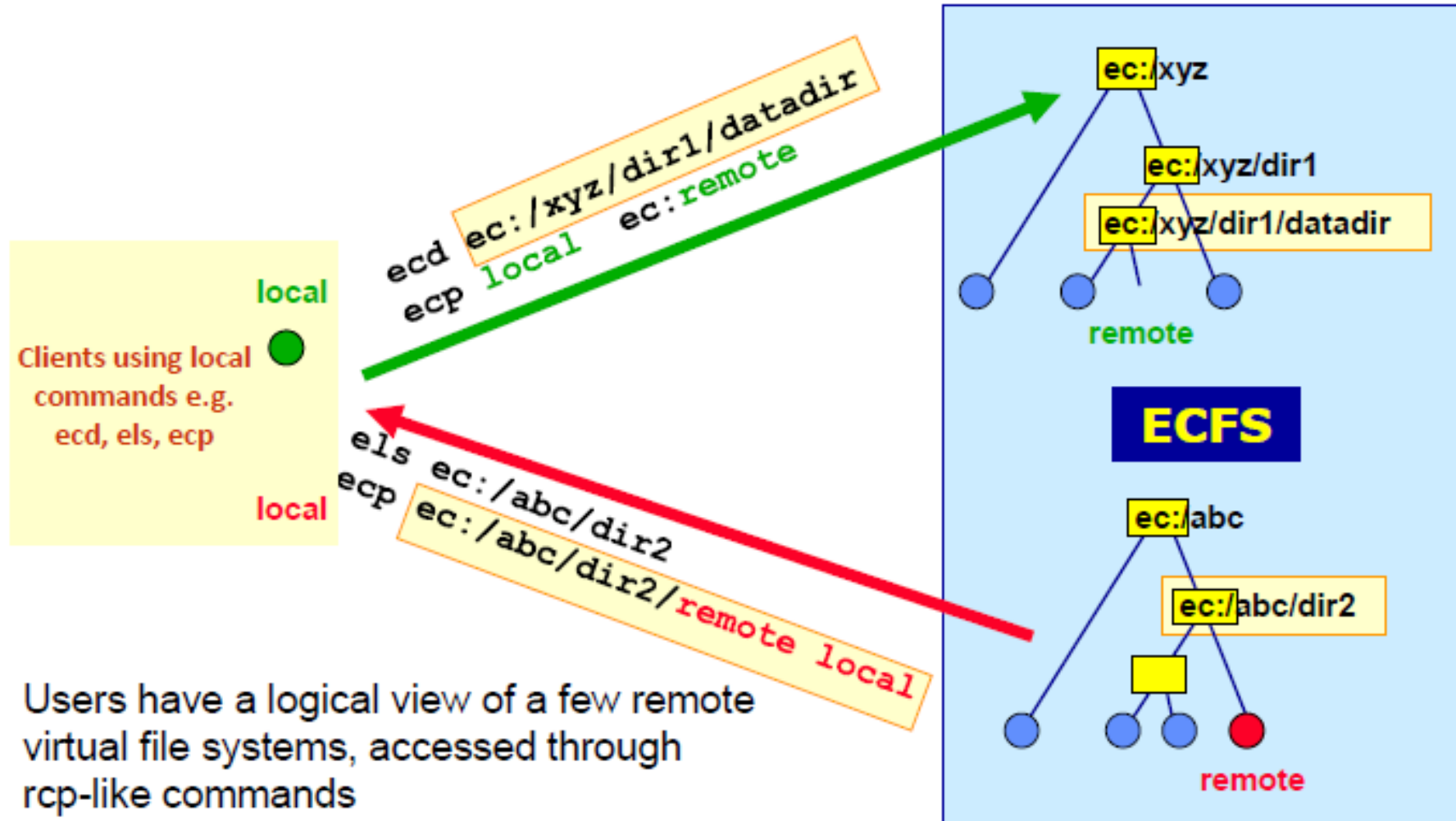
- DHS Hardware
  - Intel-based Linux servers
  - Some IBM p575/p620 servers
  - 4 Oracle SL8500 automated tape libraries
- DHS Software
  - Based on HPSS (High-Performance Storage System)
- Comprises two archives
  - **MARS** – Meteorological archive
  - **ECFS** – User archive



## DHS Services

- MARS – Meteorological Archive and Retrieval System
  - Data is accessed via a meteorological meta-language interface
  - Bulk of the data, few files (but holding billions of fields in total)
  - Relies upon excellent tape drive performance when retrieving lots of small parcels of data from tape
- ECFS – ECMWF File System
  - HSM-like (Hierarchical Storage Management) service for “ad-hoc” files that are not suitable for storing in MARS
  - Data is accessed via an rcp-like interface
  - Millions of files, many very small
- HPSS
  - Both MARS and ECFS rely on HPSS as the underlying data management system that is used to store the data
  - Users do not have direct access to HPSS, only via MARS and ECFS

# ECFS –the user’s view



## ECFS commands

- The Unix style of file interface has been adopted by ECFS:
  - files are mapped to a Unix-compatible directory tree
  - either absolute and relative pathnames can be used
  - there is a current ECFS working directory, analogous to the Unix current working directory \$PWD
  - limited wildcard support is provided.
- In ECFS the well-known Unix file management commands simply become:
  - els, erm, ermdir, emkdir, ecd, epwd, echmod, echgrp, ecp, emv (and emove), ecat, etest, etouch and eumask.
- In addition the following ECFS commands are available:
  - ecfsdir, ecfs\_status

- Use of Member State IDs
- Interactive access via Internet link  
`ssh -X -I <UID> ecaccess.ecmwf.int`  
or withNX from No Machine(the desktop Virtualization Company)



- Training IDs (tr ?, passwords )
- modules
  - module list
  - module avail
  - module load

See <https://software.ecmwf.int/wiki/display/UDOC/Modules>

## Compiling on the Crays

- 3 compiler suites: Cray, Intel and Gnu compilers
- All 3 are supported by Cray. Default compiler suite is **Cray**.
- Users can change compiler suite with 'module
- ECMWF (and Cray) recommend the **use of the compiler wrappers ftn, cc and CC for the 3 compiler suites.**
- For more information on a compiler, you'll need to read the man page of the specific compiler

# VIEW ECFLOW SERVER

ecFlowUI (4.7.1) - (menu: User)

File Panels Refresh Servers Tools Help

Manage servers - ecFlowUI (4.7.1)

... Favourites and loaded

L	Name	Host
<input type="checkbox"/>	1_od	vsms1
<input type="checkbox"/>	1_sappa	sappa
<input type="checkbox"/>	1_sappb	sappb
<input type="checkbox"/>	2_biceps_mumo_tc	ecgate
<input type="checkbox"/>	2_greece_tc	ecgate
<input type="checkbox"/>	2_harmonie_tc	ecgate
<input type="checkbox"/>	2_ireland_tc	ecgate
<input type="checkbox"/>	3_dataset_backup	vali
<input type="checkbox"/>	3_dataset_public	vsms2
<input type="checkbox"/>	3_od2	vsms2
<input type="checkbox"/>	3_od4	vsms2
<input type="checkbox"/>	3_od5	vsms2
<input type="checkbox"/>	3_tigge_lam	vsms2
<input type="checkbox"/>	5_od3	vsms2
<input type="checkbox"/>	5_webapps	ecflow-minos
<input type="checkbox"/>	9_ode	vsms3
<input type="checkbox"/>	consolidation	consolidation
<input type="checkbox"/>	deploy_master	ecflow-metab
<input type="checkbox"/>	eod2	vsms2
<input type="checkbox"/>	eod3	vsms2
<input type="checkbox"/>	eod4	vsms2
<input type="checkbox"/>	eod5	vsms2
<input type="checkbox"/>	eode	vsms3
<input type="checkbox"/>	lara	ecgb11
<input type="checkbox"/>	marsflow	marsflow
<input type="checkbox"/>	nan	charvbdis

Edit server

+ Add server

Duplicate server

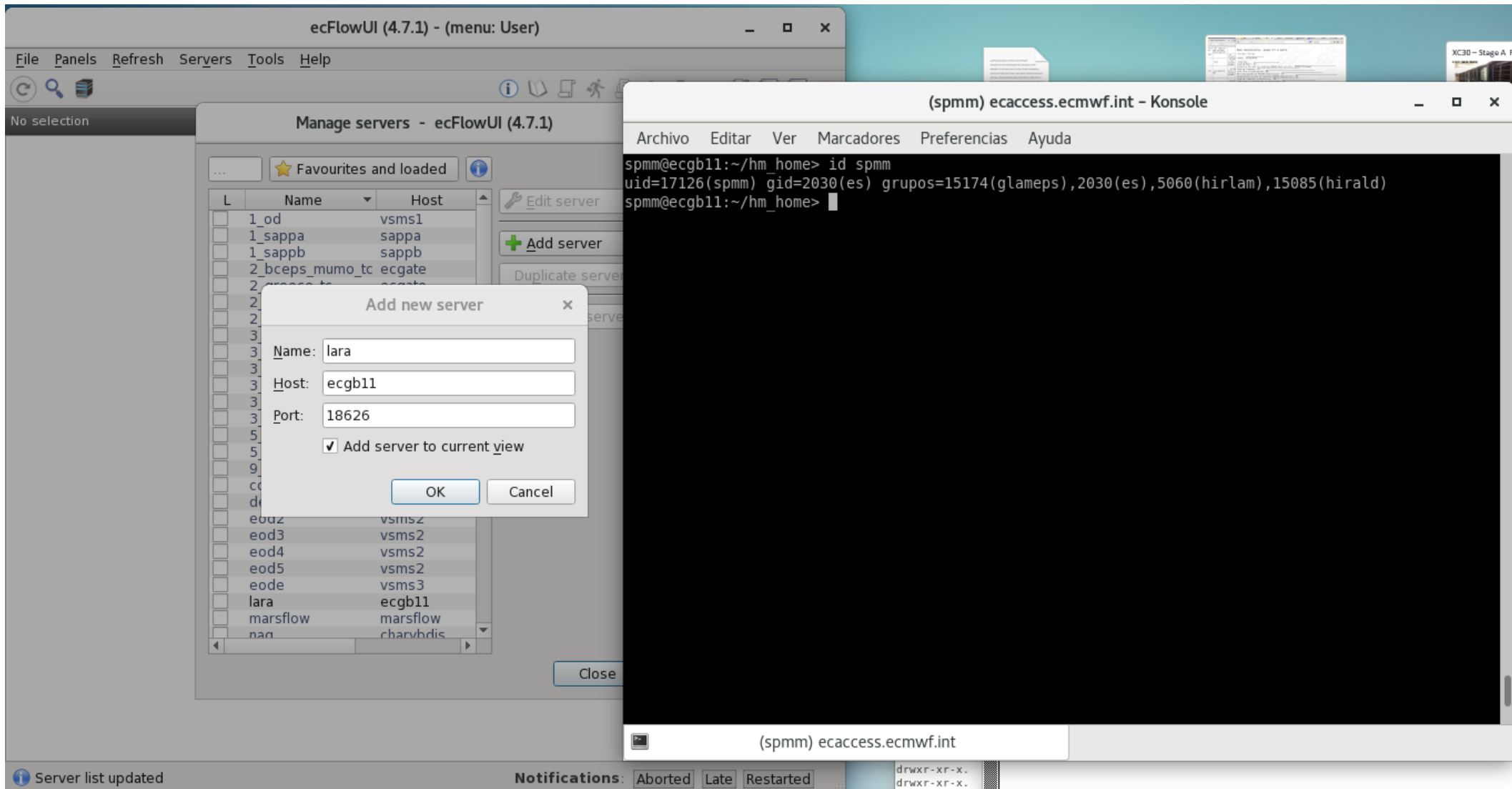
Remove server

Close

Server list updated

Notifications: Aborted Late Restarted

# VIEW ECFLOW SERVER



The screenshot displays the ecFlowUI (4.7.1) interface. The main window shows a 'Manage servers - ecFlowUI (4.7.1)' dialog with a table of servers. An 'Add new server' dialog is open, showing the following details:

Name	Host	Port
lara	ecgb11	18626

The 'Add server to current view' checkbox is checked. The terminal window shows the following output:

```
spmm@ecgb11:~/hm_home> id spmm
uid=17126(spmm) gid=2030(es) grupos=15174(glameps),2030(es),5060(hirlam),15085(hirald)
spmm@ecgb11:~/hm_home>
```

At the bottom of the screen, a notification bar shows 'Server list updated' and 'Notifications: Aborted Late Restarted'.

## More info in:

- Introduction:

<https://confluence.ecmwf.int/x/dl48Aw>

- Cray Computers

<https://confluence.ecmwf.int/x/aZ8wAw>

- Generic docs

<https://confluence.ecmwf.int/display/UDOC/User+Documentation>

