UPPSALA
UNIVERSITET

Introduction to parallel computing and UPPMAX

Intro part of course in Parallel Image Analysis

Elias Rudberg

elias.rudberg@it.uu.se

March 22, 2011

Parallel computing

Parallel computing is becoming increasingly important nowadays.

- Physics
- Chemistry
- Biology
- Weather forecasts
- Image analysis!
- ...

Supercomputers and computing centers

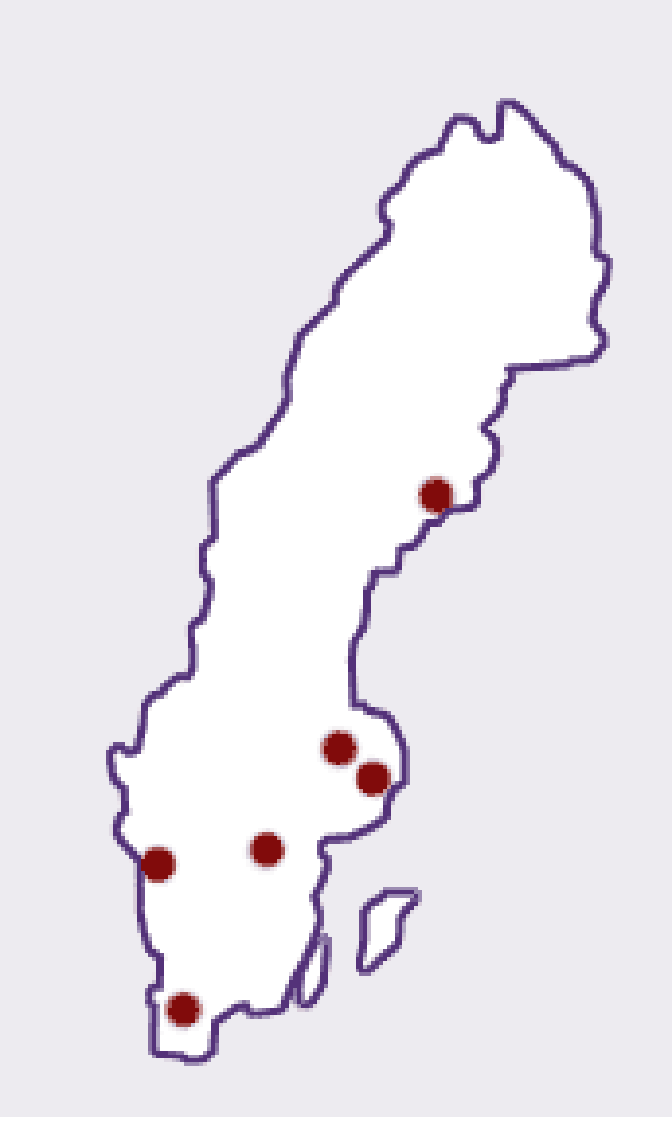
Many computing centers / supercomputers exist in the world today.
A few of the largest ones:

- National Supercomputing Center of Tianjin (China)
- Oak Ridge National Laboratory (USA)
- GSIC Center, Tokyo Institute of Technology (Japan)
- Commissariat à l'énergie atomique (CEA) (France)
- Forschungszentrum Jülich (Germany)

In Sweden, there are six computing centers organized through the Swedish National Infrastructure for Computing (SNIC).

UPPSALA
UNIVERSITET

Six computing centers in Sweden



SNIC centers:

- HPC2N in Umeå.
- UPPMAX in Uppsala. Here!
- PDC in Stockholm.
- NSC in Linköping
- C3SE in Göteborg
- LUNARC in Lund.

UPPSALA
UNIVERSITET

UPPMAX

Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX).

UPPMAX is Uppsala University's resource of high-performance computers, large-scale storage, and know-how of high-performance computing (HPC).

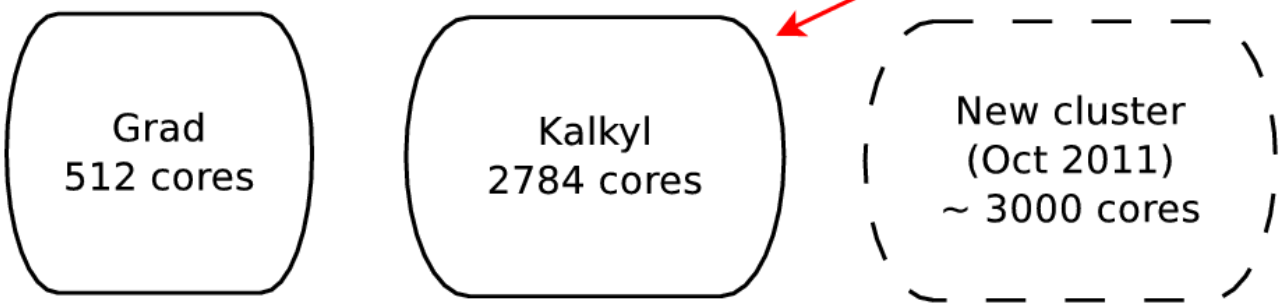
UPPMAX

UPPMAX

Staff:

- Director: Ingela Nyström
- Research administrator: Lena Wadelius
- System experts: Jukka Komminaho, Tore Sundqvist, Martin Agback, Jonas Hagberg, Lennart Karlsson, Samuel Lampa
- Application experts: (~ 7 persons at 50%)

Resources (compute clusters):



UPPSALA
UNIVERSITET

Kalkyl

The compute cluster Kalkyl was delivered in November 2009 and named after Professeur Tryphon Tournesol, (Professor Karl Kalkyl in Swedish, Professor Cuthbert Calculus in English) from the cartoons by Hergé.

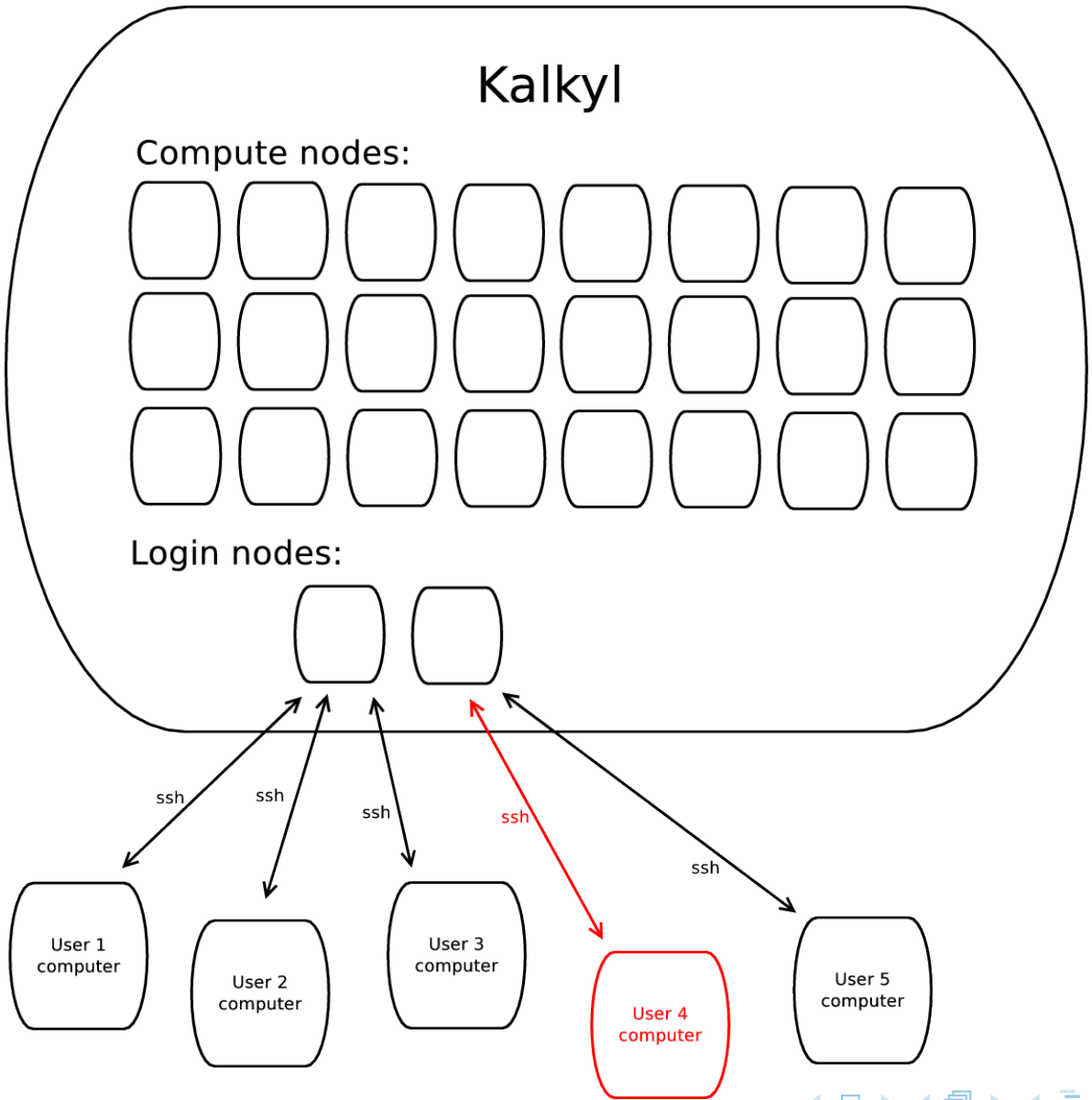
UPPSALA
UNIVERSITET

How does a compute cluster like Kalkyl work?

- Many compute nodes connected via fast interconnect (Infiniband).
- A few nodes act as *login nodes*.
- Each node is a multicore computer running Linux.
- Users log in to the login nodes using ssh (Secure Shell protocol).
- Users run jobs on compute nodes through a queueing system.

UPPSALA UNIVERSITET

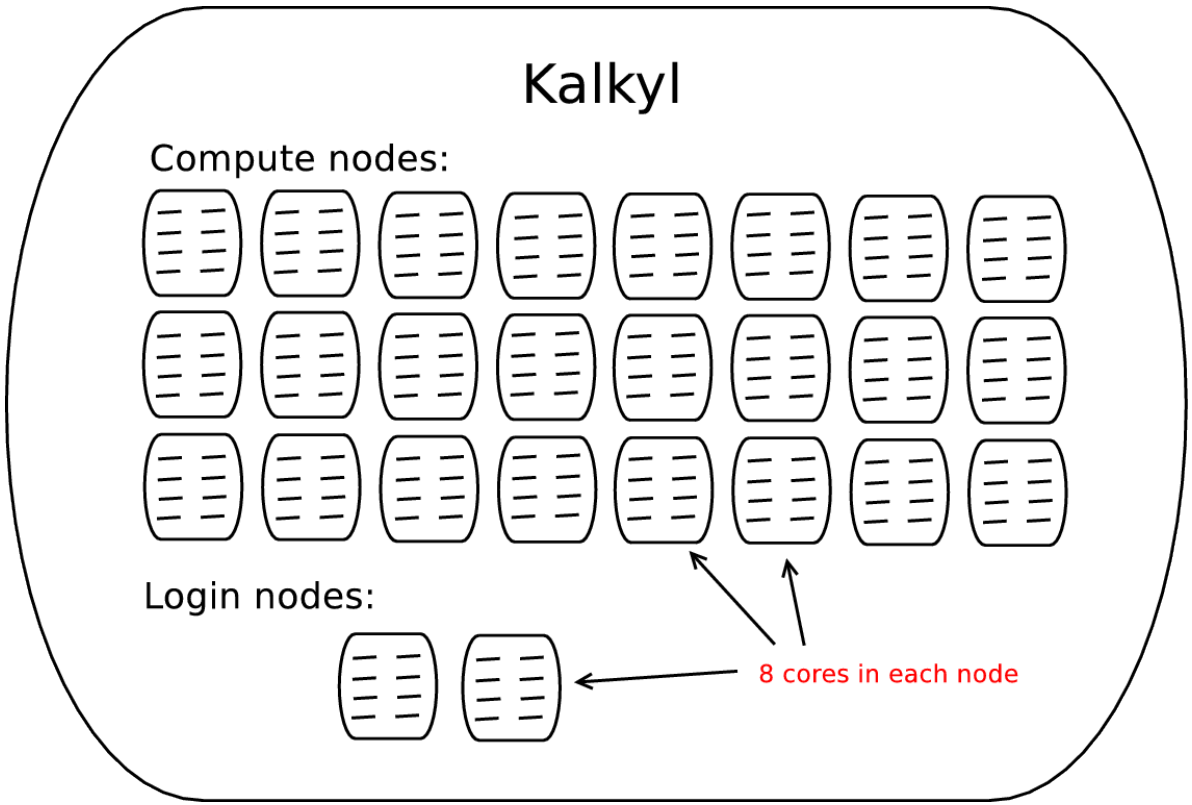
How does a compute cluster like Kalkyl work?



UPPSALA
UNIVERSITET

How does each compute node work?

Each compute node is a multi-core computer. On Kalkyl, each node has 8 cores.



Linux

Each node on Kalkyl in running the Linux operating system. The Linux version installed now is Scientific Linux SL release 5.5.

You can use Kalkyl even if your local computer is not using Linux.

- If using Mac, you can use ssh directly at the command prompt.
- If using Windows, you can install and use some ssh program, for example “putty”.
- In other operating systems there is probably also some way of using ssh.

Ways to use a compute cluster like Kalkyl

There are several different ways of using a compute cluster:

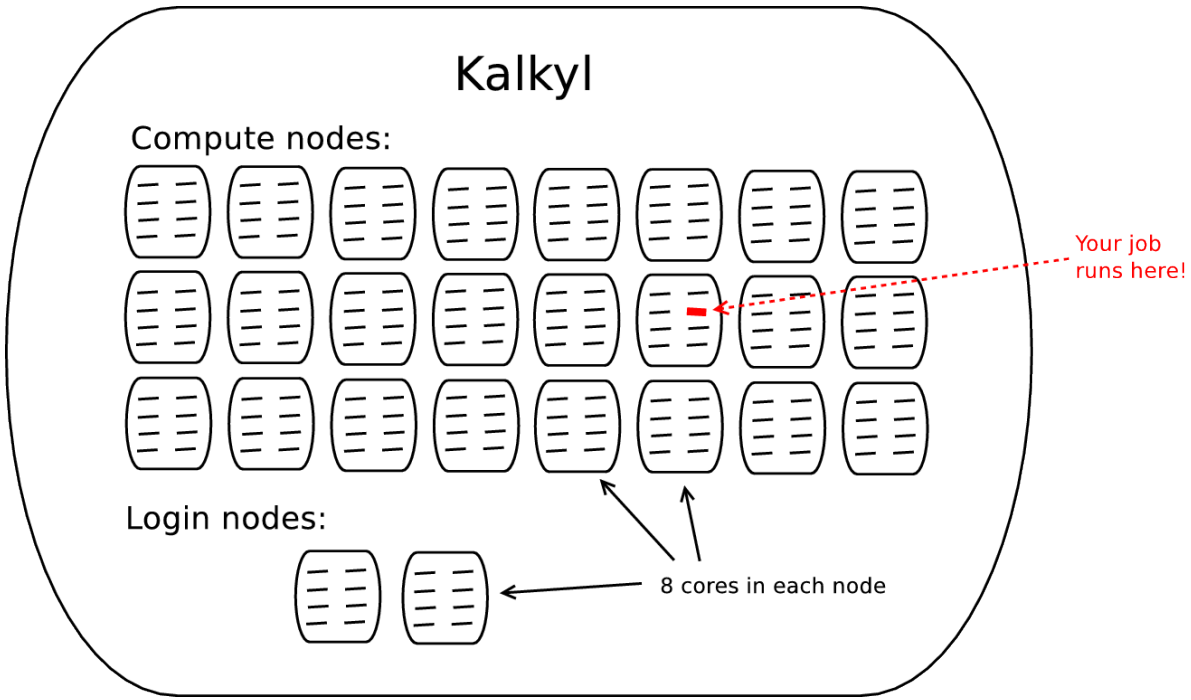
- Run serial program as a single-core job. Possibly run many such independent jobs in parallel. No parallel programming needed.
- Run threaded program as a single-node job. The same program then uses all 8 cores in one compute node. Possibly 8 times faster compared to single-core run.
- Run program using distributed memory parallelization using the message passing interface (MPI). The same program uses N compute nodes. Possibly $N \times 8$ times faster compared to single-core run.

UPPSALA UNIVERSITET

Ways to use a compute cluster like Kalkyl

Single-core job

A serial program (a program not using threads) may run as a single-core job.



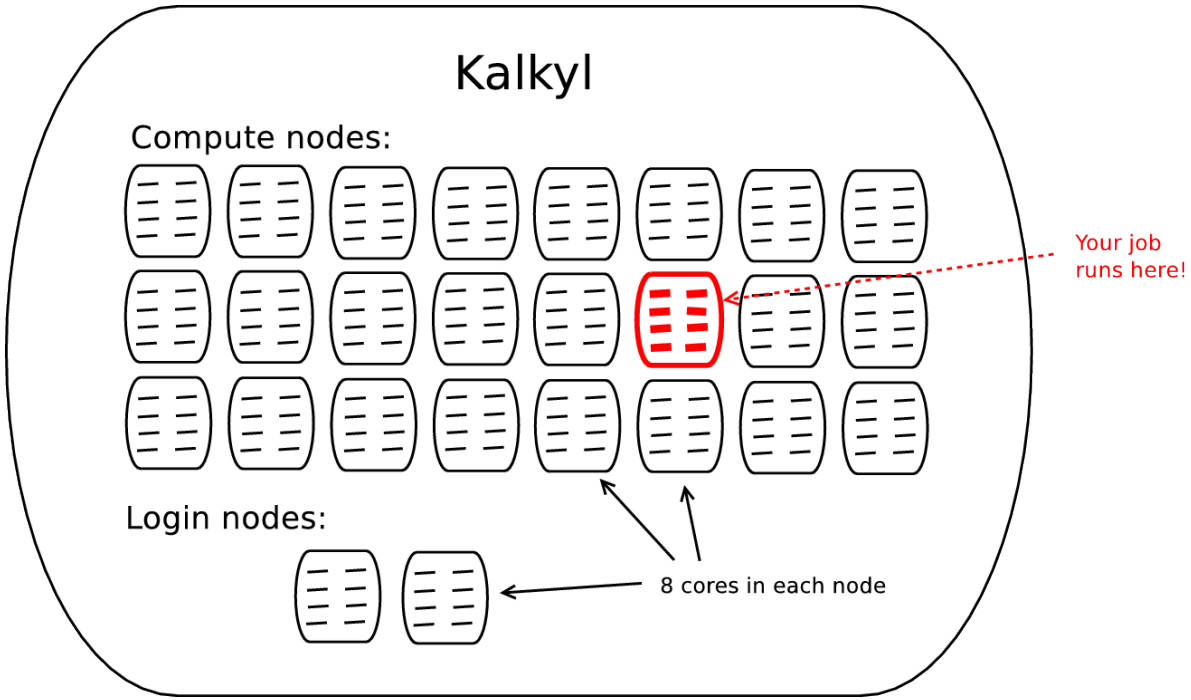
Other users may have jobs running on other cores on the same compute node wher your job runs.

UPPSALA UNIVERSITET

Ways to use a compute cluster like Kalkyl

Single-node job

A threaded program may run as a single-node job.

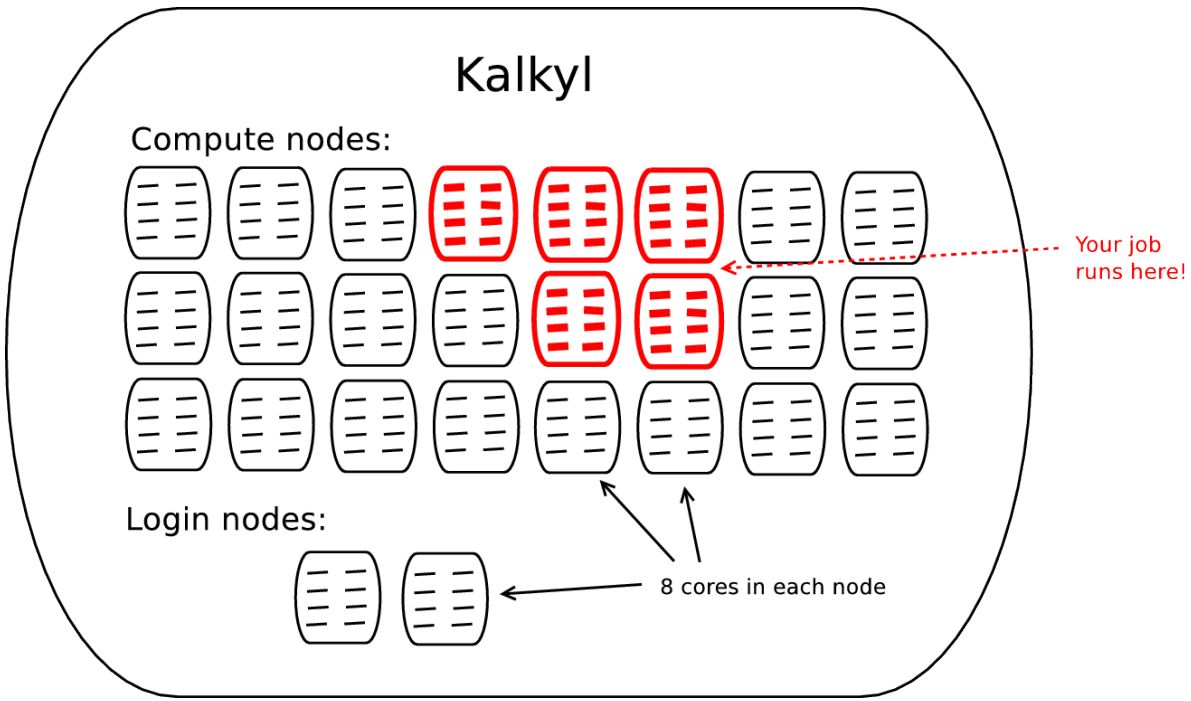


UPPSALA UNIVERSITET

Ways to use a compute cluster like Kalkyl

Multi-node job

A distributed memory parallelized (typically MPI) program may run on several compute nodes.



Queueing systems

As a user, you typically run your program on compute node(s) by submitting a job script through a *queueing system*. The queueing system assigns different priorities to different jobs depending on which project they belong to and how much of each project's time has been used.

There are several different queueing systems, for example:

- SLURM Used on Kalkyl at UPPMAX!
- Grid Engine (used on Grad)
- PBS/Torque/Maui (used at HPC2N)
- EASY scheduler (used at PDC)

We focus on SLURM.

UPPSALA UNIVERSITET

Using the SLURM queueing system on Kalkyl

The Simple Linux Utility for Resource Management (SLURM) is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters.

Useful commands available to control SLURM on Kalkyl:

```
sbatch     (submit a job)
squeue     (get info about previously submitted jobs)
scancel    (cancel a job)
sinfo      (get info about available resources)
projinfo   (get info about project, hours used etc)
```

Example of job script

Example of job script (filename jobscript.sh):

```
#!/bin/bash
#SBATCH -p node
#SBATCH -N 1
#SBATCH -t 00:30:00
#SBATCH -J eliasjob1
#SBATCH -A g2011040
./ergo -m 0002.xyz < egonparams.ego
```

To submit the job:

```
$ sbatch jobscript.sh
Submitted batch job 417772
```

UPPSALA UNIVERSITET

Using the SLURM queueing system on Kalkyl

After having submitted job(s), use the squeue command to check the status of your jobs:

```
$ squeue -u eliasr
  JOBID PARTITION     NAME     USER  ST       TIME  NODES NODELIST(REASON)
 417778      node eliasjob   eliasr  PD       0:00      1 (Priority)
```

If you have more than one job, squeue gives a list of jobs:

```
$ squeue -u eliasr
  JOBID PARTITION     NAME     USER  ST       TIME  NODES NODELIST(REASON)
 417781     devel eliasjob   eliasr   R       0:18      1 q33
 417778      node eliasjob   eliasr   R       0:28      1 q144
```

UPPSALA
UNIVERSITET

Kalkyl User Guide

For more information about how to run jobs, see the Kalkyl User Guide available at the UPPMAX web page:

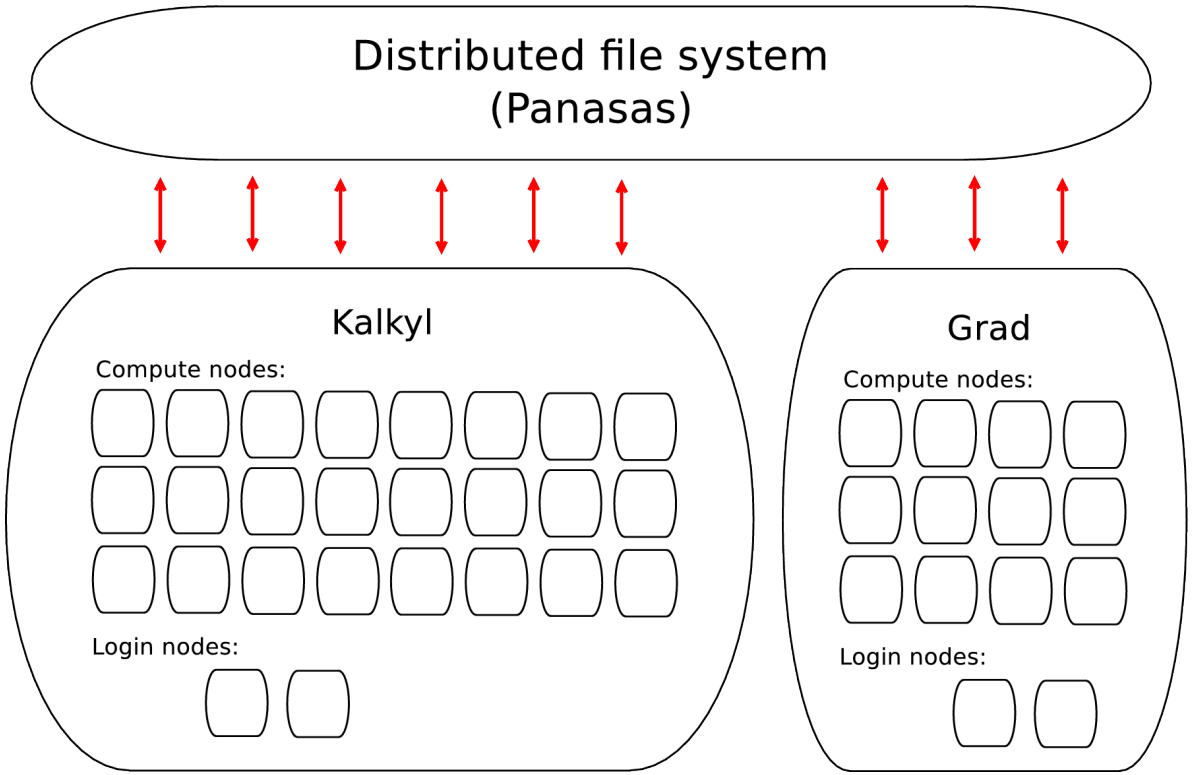
`http://www.uppmax.uu.se/`

`http://www.uppmax.uu.se/support/user-guides/kalkyl-user-guide`

UPPSALA UNIVERSITET

Distributed file systems

- Same file system accessible from all compute nodes.
- Same file system also accessible from all UPPMAX clusters (currently Grad and Kalkyl).
- Local /scratch directory on each node can provide faster file operations, useful for temporary files.



UPPSALA UNIVERSITET

UPPMAX accounts and projects

Each person using UPPMAX needs a personal *user account*.

The type and amount of resources accessible to each user depends on which *project(s)* the user belongs to. When submitting a job to the queueing system, you must specify which project should be charged with the run time for the job.

When logged in to Kalkyl, you can check which projects you are a member of using the projinfo command:

```
$ projinfo g2011040
(Counting the number of core hours used since 2011-03-01/00:00:00 until now.)

Project             Used[h]   Current allocation [h/month]
   User
-------------------------------------------------------
g2011040              11.49                2000
   cris                1.32
   eliasr             10.17
```

What will happen during this course?

- This week: start using queueing system on Kalkyl to run single-core jobs.
- Week 2: Shared-memory parallelization for multi-core computers. OpenMP.
- Week 3: parallelization on GPU:s using OpenCL.
- Week 4: Distributed-memory parallelization. MPI.
- Week 5: Your own projects.