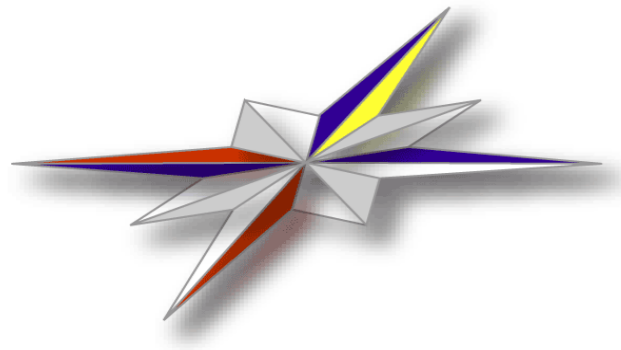# Monitoring ARC services with GangliARC

D. Cameron and **D. Karpenko** (University of Oslo, Norway)

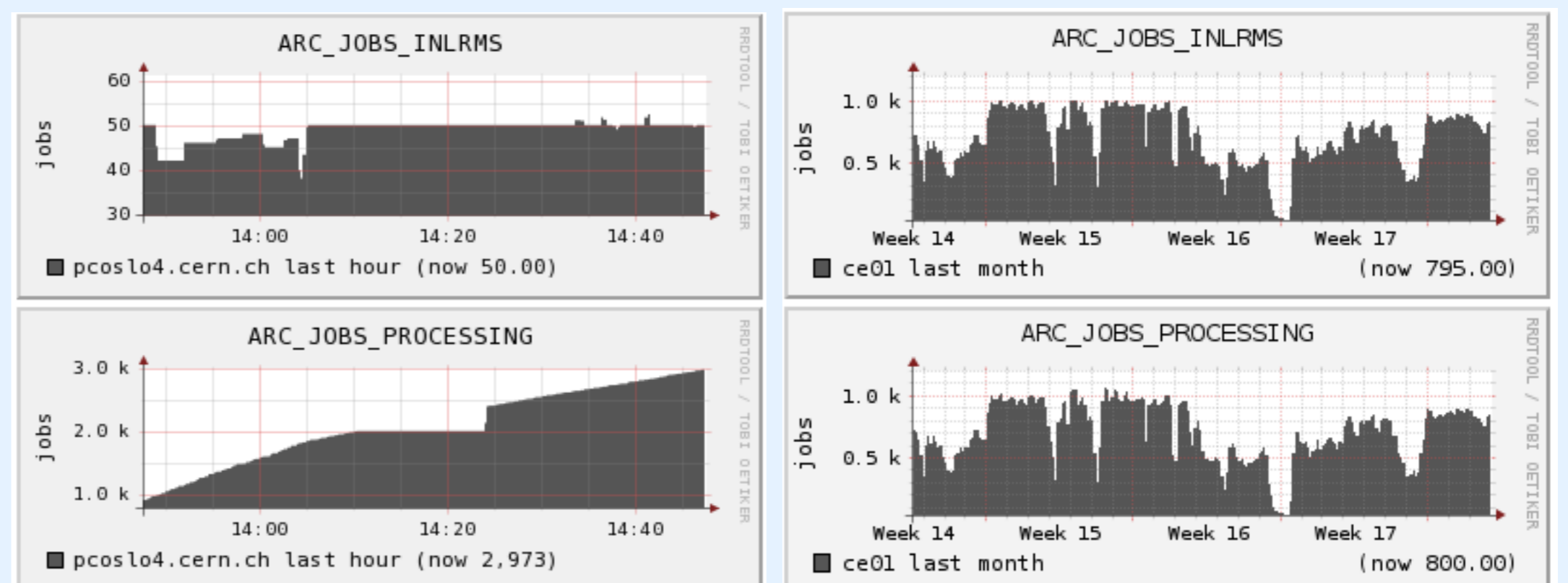## How do you know what your ARC CE is doing?

GangliARC is a framework for adding ARC-related metrics from ARC's Computing Element (A-REX) to Ganglia. It allows at-a-glance monitoring of the state of the CE and easy diagnosis of common problems. It consists of a daemon running on the A-REX host which regularly collects information and adds it to Ganglia's data feed using the gmetric tool.
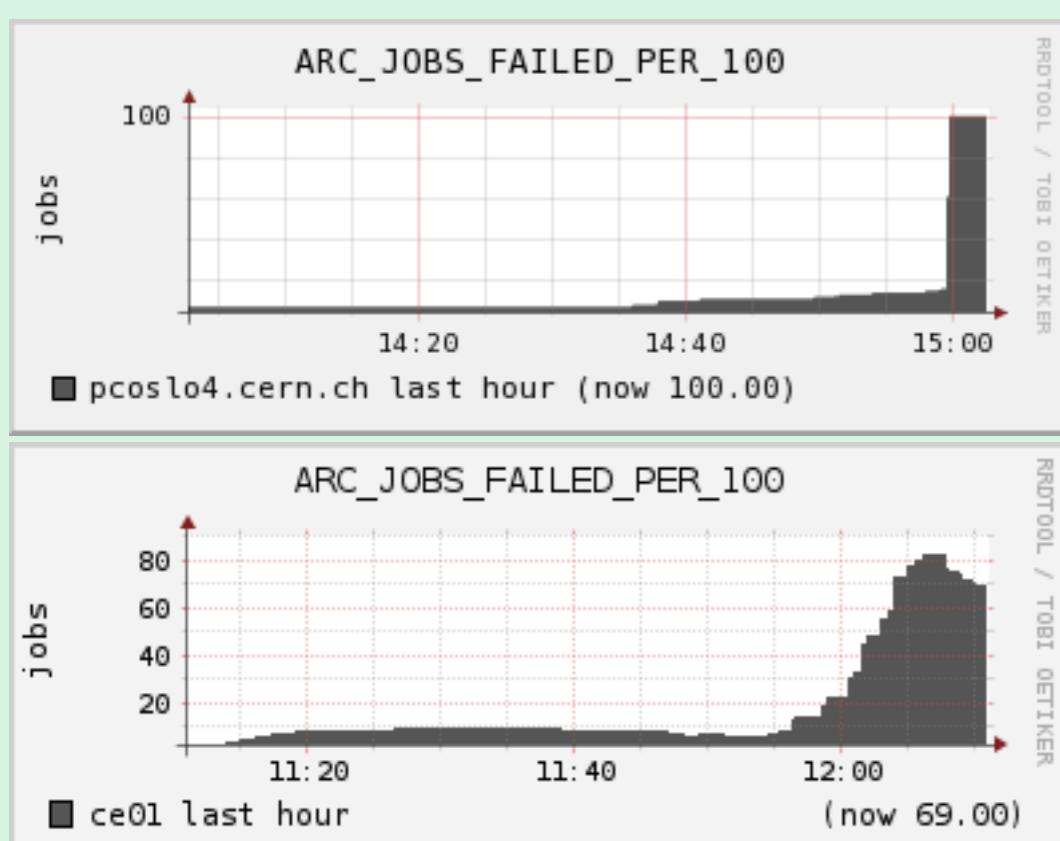
Is A-REX running?
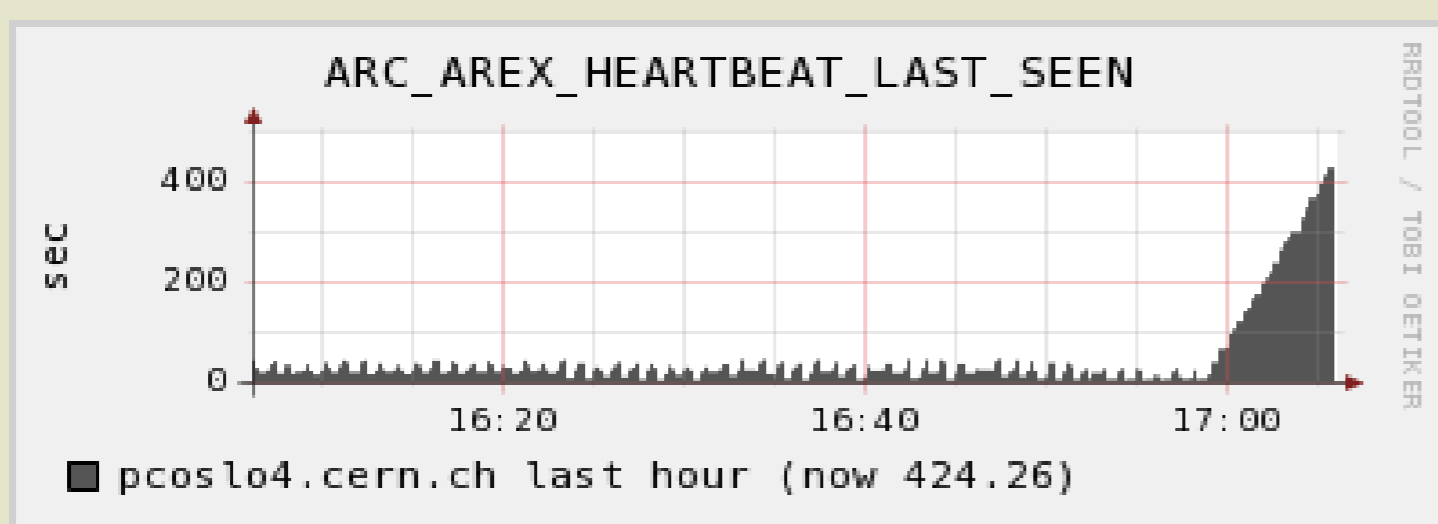
A-REX regularly issues a heartbeat to show it is alive and this graph shows the time since the last heartbeat. It starts to increase after 17:00 which means the heartbeat has stopped, due to the process dying unexpectedly or getting stuck. Some manual intervention is required to investigate more.

Are jobs being processed?

The number of failed jobs out of the last 100 is shown in this graph. In the top graph the number shoots up to 100 and stays there, indicating a possible site-wide problem such as a filesystem failure. The bottom graph shows a gradual increase and then decrease, which probably means the user submitted a large batch of bad jobs.

How is the cache space evolving?

ARC keeps a local cache to avoid repeatedly downloading the same files. This graph shows the cache free space over one week. It can be clearly seen when the oldest files in the cache are cleaned, by the sharp increases from 1TB to 2TB free space.

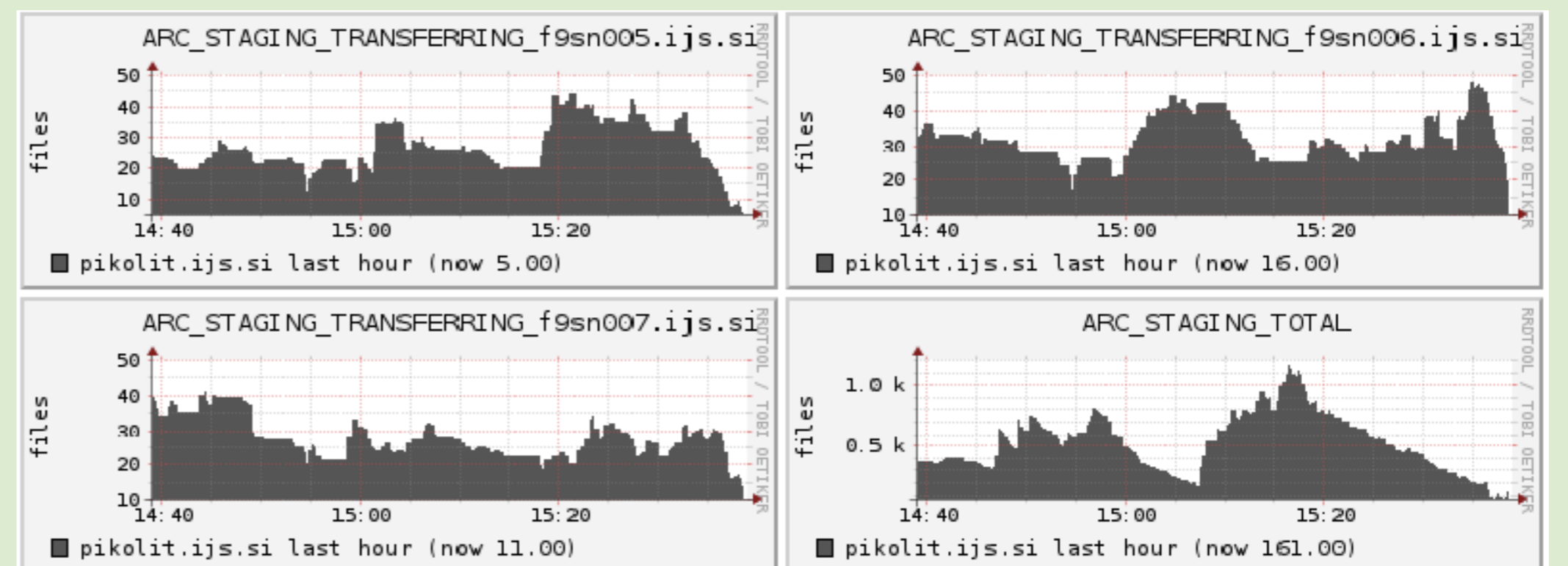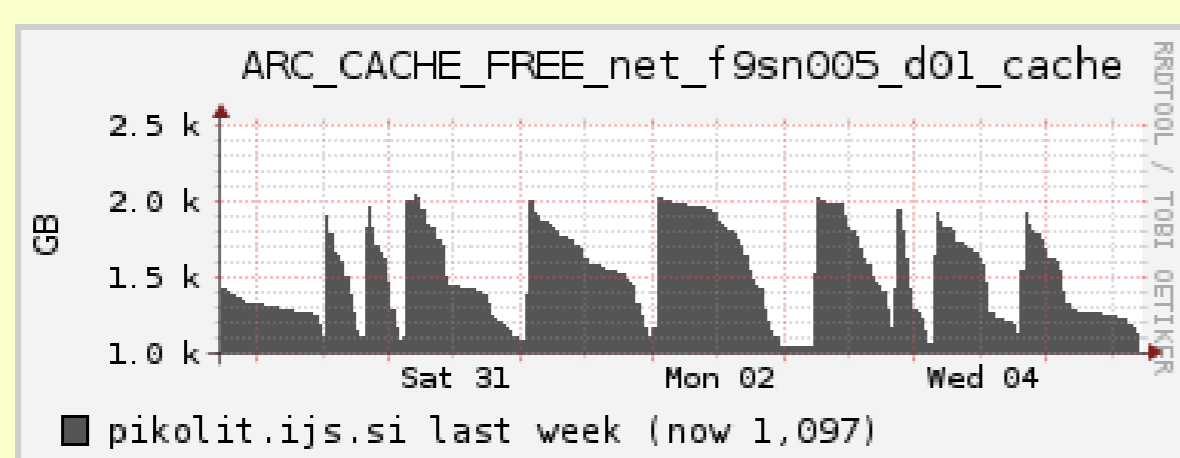## Is the site overloaded with jobs?

The top-left graph shows the number of jobs currently queued or running in the batch system for one CE and it can be seen that ARC has limited this number to 50. The bottom-left graph has the total number of jobs being processed by A-REX and in this case it is steadily rising, indicating that jobs are being submitted much faster than they can be processed and perhaps the user's brokering logic is not optimal. The two right-hand graphs show the same metrics for another CE over the last month but here the situation looks healthier.
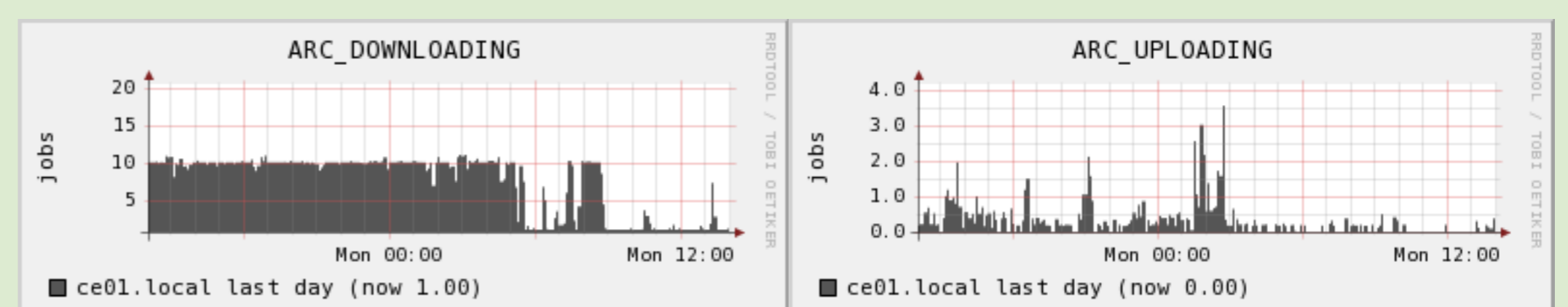
How about the data staging queue?

ARC's new data staging system allows data transfer done by the CE to be spread over multiple nodes to increase network throughput. The first three graphs here show how the 100 available transfer slots are distributed over 3 nodes, and the bottom-right graph gives the evolution of the staging queue.

Alternatively, with the old staging system the number of downloaders and uploaders can be seen. In this case the site was saturated with jobs, requiring the maximum 10 downloaders running continuously for more than 12 hours. It can be seen that for these jobs the size of output data was small compared to input data.

What about integration with Nagios?

Host ↑↓	Service ↑↓	Status ↑↓	Last Check ↑↓	Duration ↑↓	Attempt ↑↓	Status Information
pcoslo4.cern.ch	ARC Failed Jobs	OK	04-05-2012 17:11:21	14d 1h 19m 38s	1/4	CHECKGANGLIA OK: ARC_JOBS_FAILED_PER_100 is 9.00
	ARC Heartbeat	CRITICAL	04-05-2012 17:10:55	0d 0h 11m 4s	4/4	CHECKGANGLIA CRITICAL: ARC_AREX_HEARTBEAT_LAST_SEEN is 705.10
	Current Load	OK	04-05-2012 17:15:28	11d 12h 45m 31s	1/4	OK - load average: 1.07, 1.12, 1.22
	Current Users	OK	04-05-2012 17:12:01	196d 0h 18m 38s	1/4	USERS OK - 8 users currently logged in

Integrating Ganglia and Nagios is a natural way to provide a complete monitoring infrastructure, and GangliARC metrics can easily be added to regular Nagios checks. This screenshot shows the status of checks for the number of failed jobs and the A-REX heartbeat. The heartbeat check is currently in a critical state because the heartbeat has not been seen for 705 seconds.

Questions? Email d.g.cameron@fys.uio.no or visit