Large-scale Grid Computing for Content-based Image Retrieval

Dr Chris Town

Dr Karl Harrison

UNIVERSITY OF CAMBRIDGE

UNIVERSITY OF BIRMINGHAM

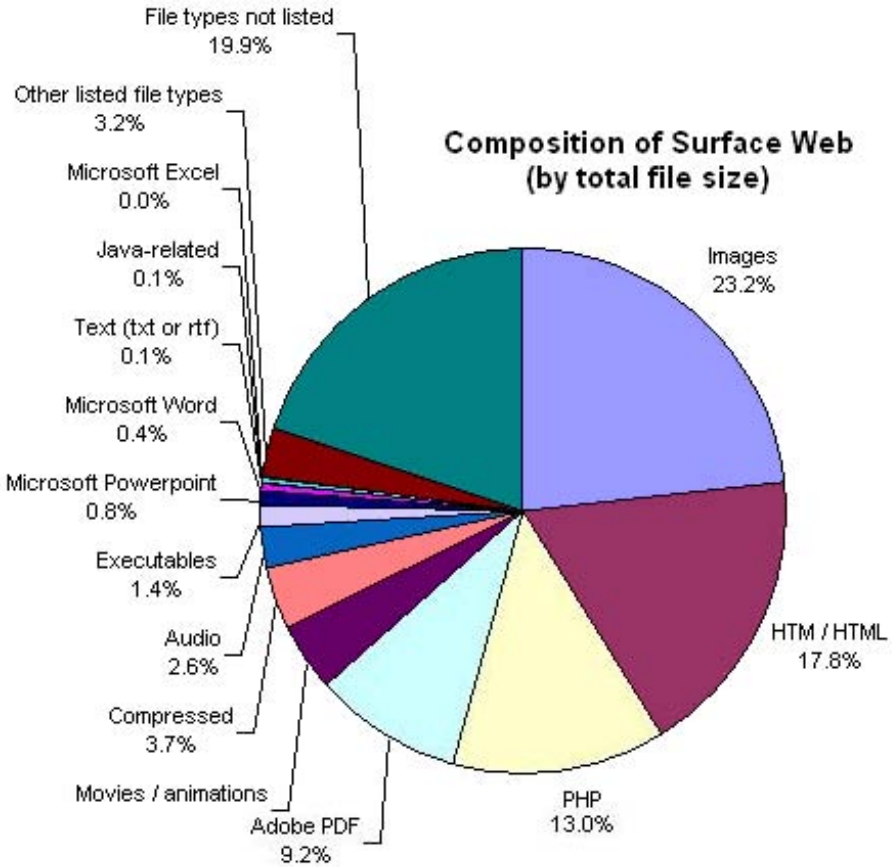
imense

The future of image search

Managing image overload

The Web: Google, Yahoo, MSN etc. only index ***text***

But: Only <25% of internet is text, can't search media ***content***



Source: Berkeley, 2003

The Home:

150 million digital cameras and over 500 million camera phones are sold each year

→ over 1 trillion digital consumer pictures

→ often called "DSC00xxx"…

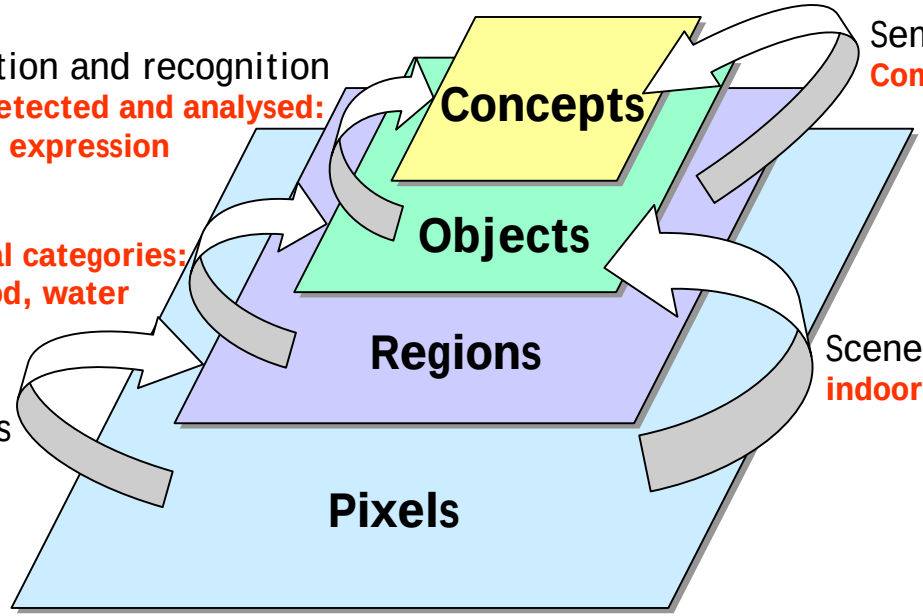
→ no way of searching, organising, or browsing by ***content***

Imense approach: Image Analysis

Object detection and recognition
Human faces detected and analysed: sex, age, facial expression

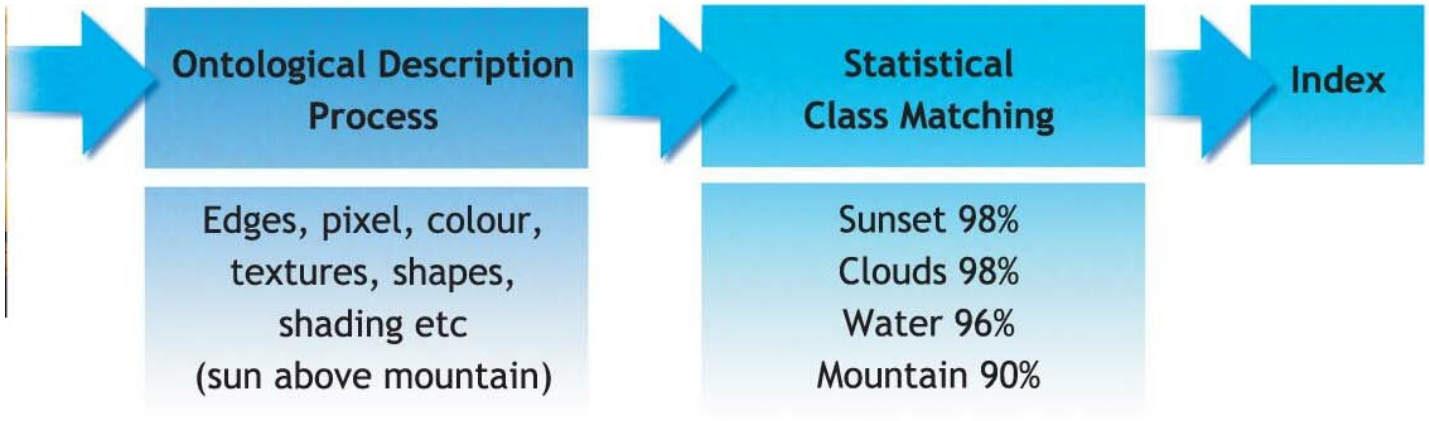
Region classification
Material and environmental categories: skin, cloth, grass, sky, wood, water

Segmentation into regions
Computation of properties: size, colour, shape, texture

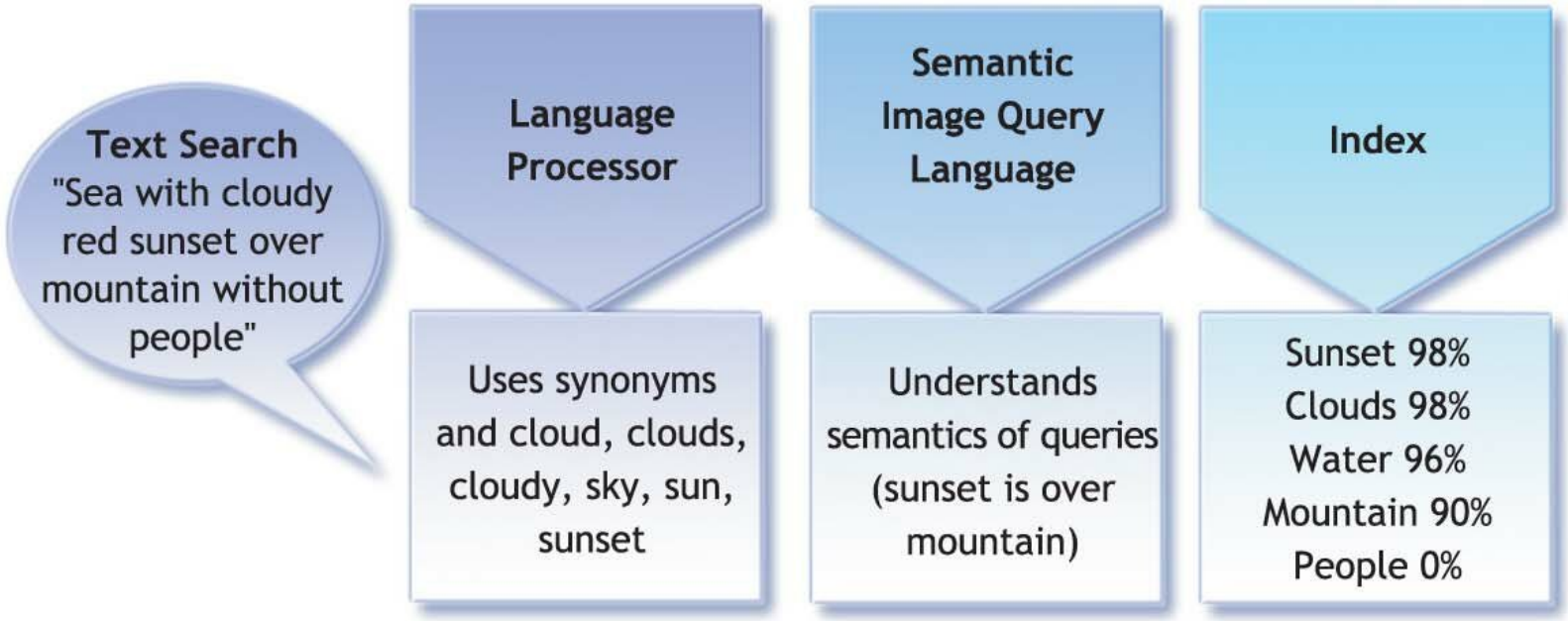


Semantic descriptor extraction
Combine all information in index

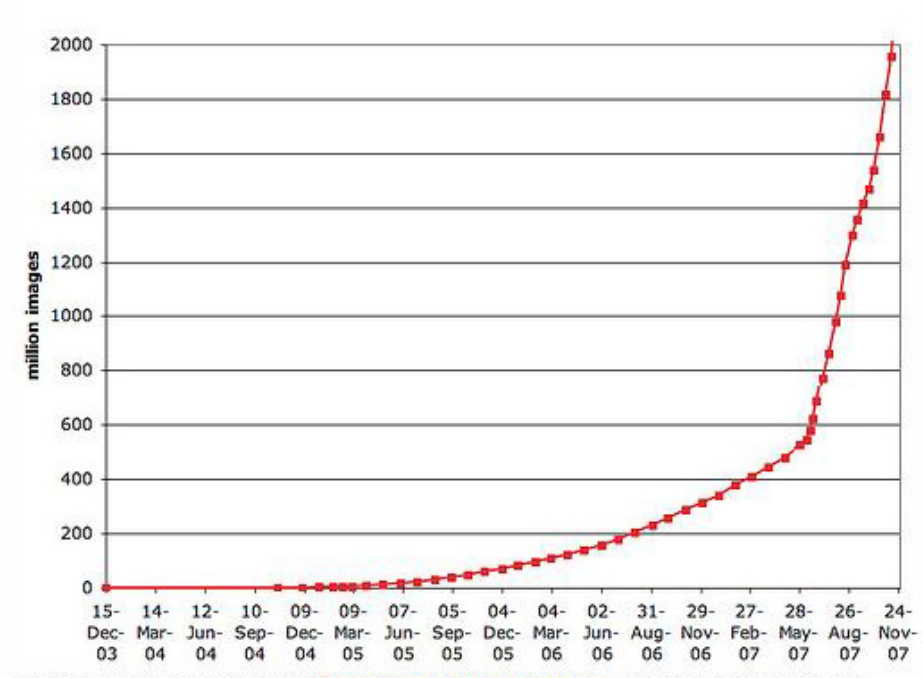
Scene classification
indoor, beach, sunset, nighttime, autumn



Imense approach: Image Retrieval



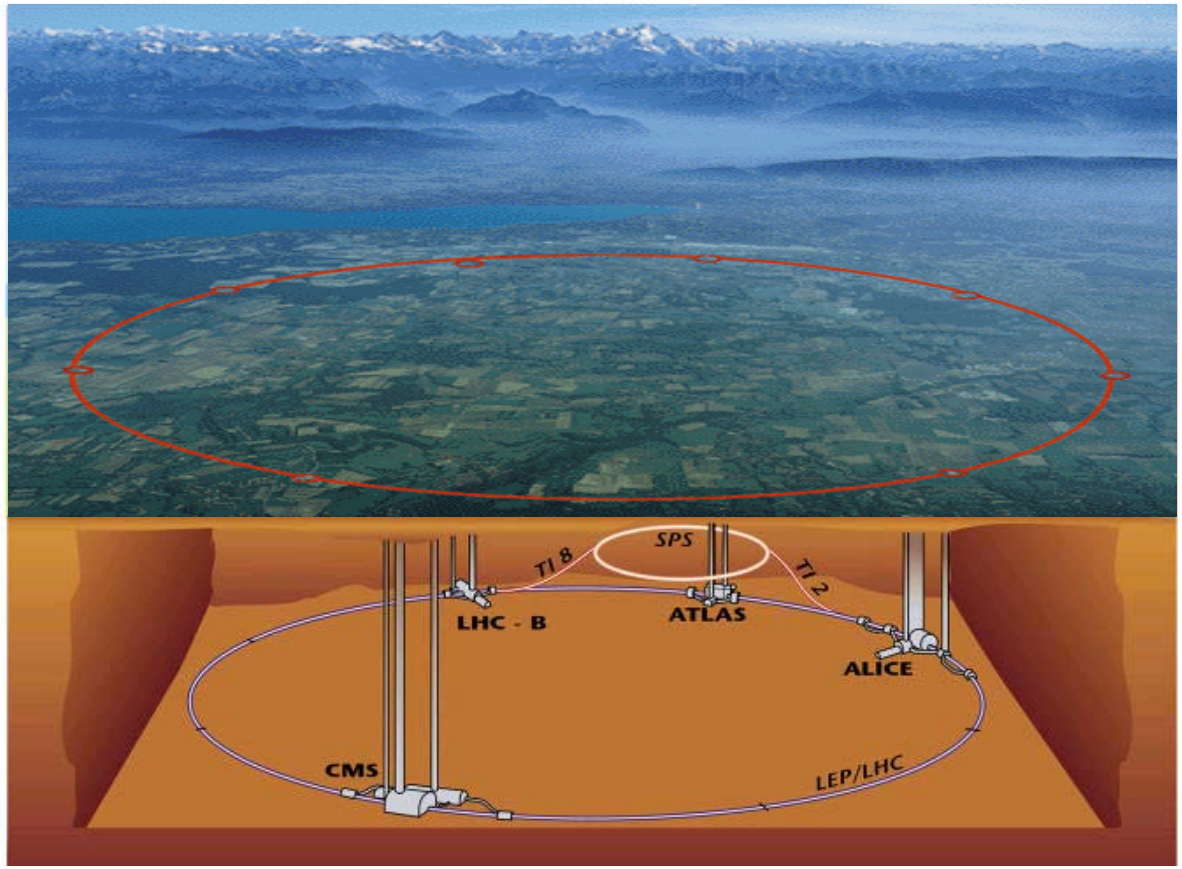
Internet scale Content-based Image Retrieval



2,000,000,000 images uploaded to flickr by Jan 2008

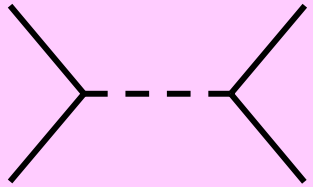
- There are more then 15x10^9 images hosted on the Internet!
  - 4000 CPU-years to index.
    - Or about 14 days on a 100,000 CPU cluster!
  - At 250kBytes/ image: 3750TB
    - $0.10/Gig $375,000 just for bandwidth to move the images about.

CERN

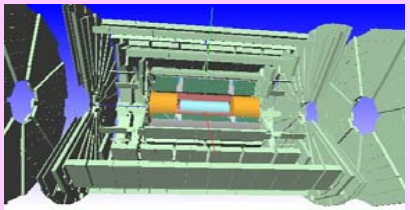


- 14 TeV Collisions
- 27 km circumference
- 1200 14m SC dipoles
- 8.36 Tesla; –270c
- 5000 SC magnets
- 700,000L liquid He
- 12,000,000L liquid N_2

Data processing in particle physics



Interaction kinematics



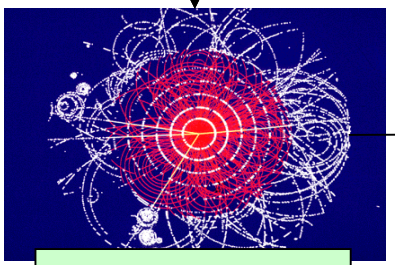
Detector response

```
1000111101011101
0110010111011010
0011101110100011
1011000101010011
11001010100
```

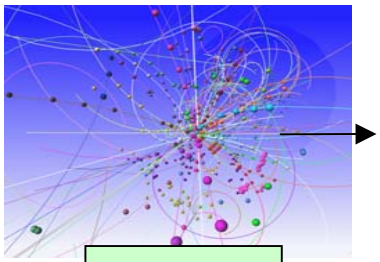
Digitisation

Simulation

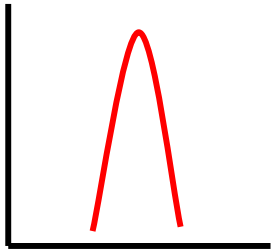
Online selection
and detector readout



Reconstruction



Filtering



Physics analysis

Scope and complexity of particle-physics experiments has increased in parallel with increases in computing power

- Four main experiments at Large Electron-Positron (LEP) accelerator at CERN

⇒ Operated 1989-2000: collected total of 2.7-3.5 TByte of data per experiment

- Four main experiments at Large Hadron Collider (LHC), starting late 2009, will each collect around 5 TByte of data per day

Dr Chris Town

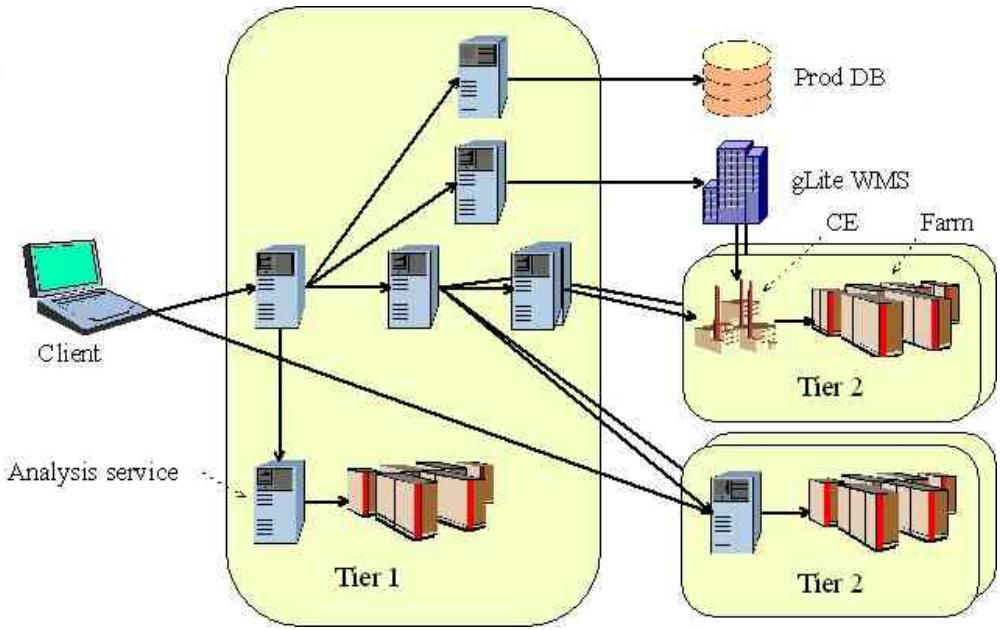
Grid Computing – CERN data centre

STFC funded Collaboration

Collaboration with the Cambridge eScience Centre and the HEP group at the Cavendish (Prof Andy Parker), funded by STFC

Have access to GridPP (UK) and EGEE (Europe) grids

About 120,000 CPU, 100 PB of storage



Science & Technology Facilities Council

Knowledge Transfer

STFC funded Collaboration

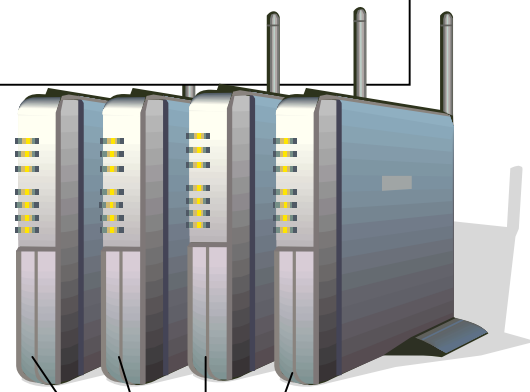
GridPP
UK Computing for Particle Physics

eGee
Enabling Grids
for E-sciencE

Grid CPU access

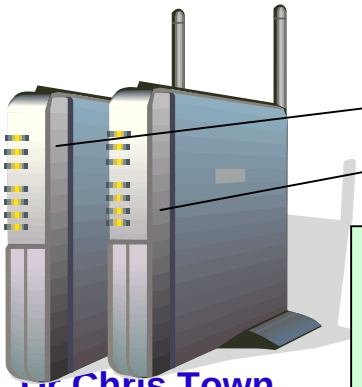
RB (Resource Broker) or WMS (Workload Management System)

Machine that decides where jobs should run



UI (User Interface)

Machine from which user submits processing requests (jobs) specified in Job Definition Language (JDL)



CE (Compute Element)

Machine that manages batch system at Grid site

WN (Worker Node)

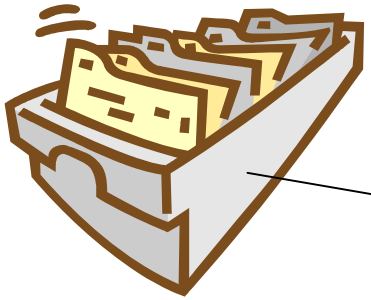
Machine that runs user jobs

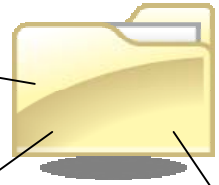
Dr Chris Town

Grid data management

LFN (Logical File Name)

Alias for any of one or more files (replicas) with identical content



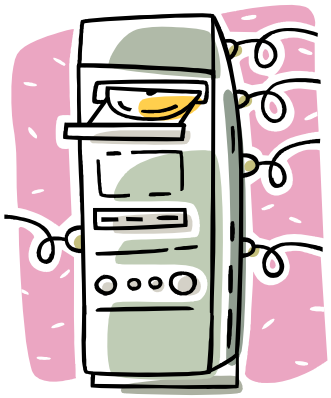


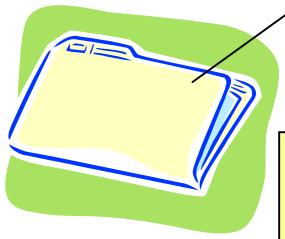
LFC (Logical File Catalogue)

Database that maps Logical File Names to Physical File Names

SE (Storage Element)

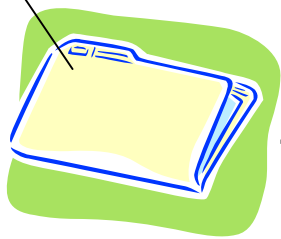
Mass-storage system at a Grid site



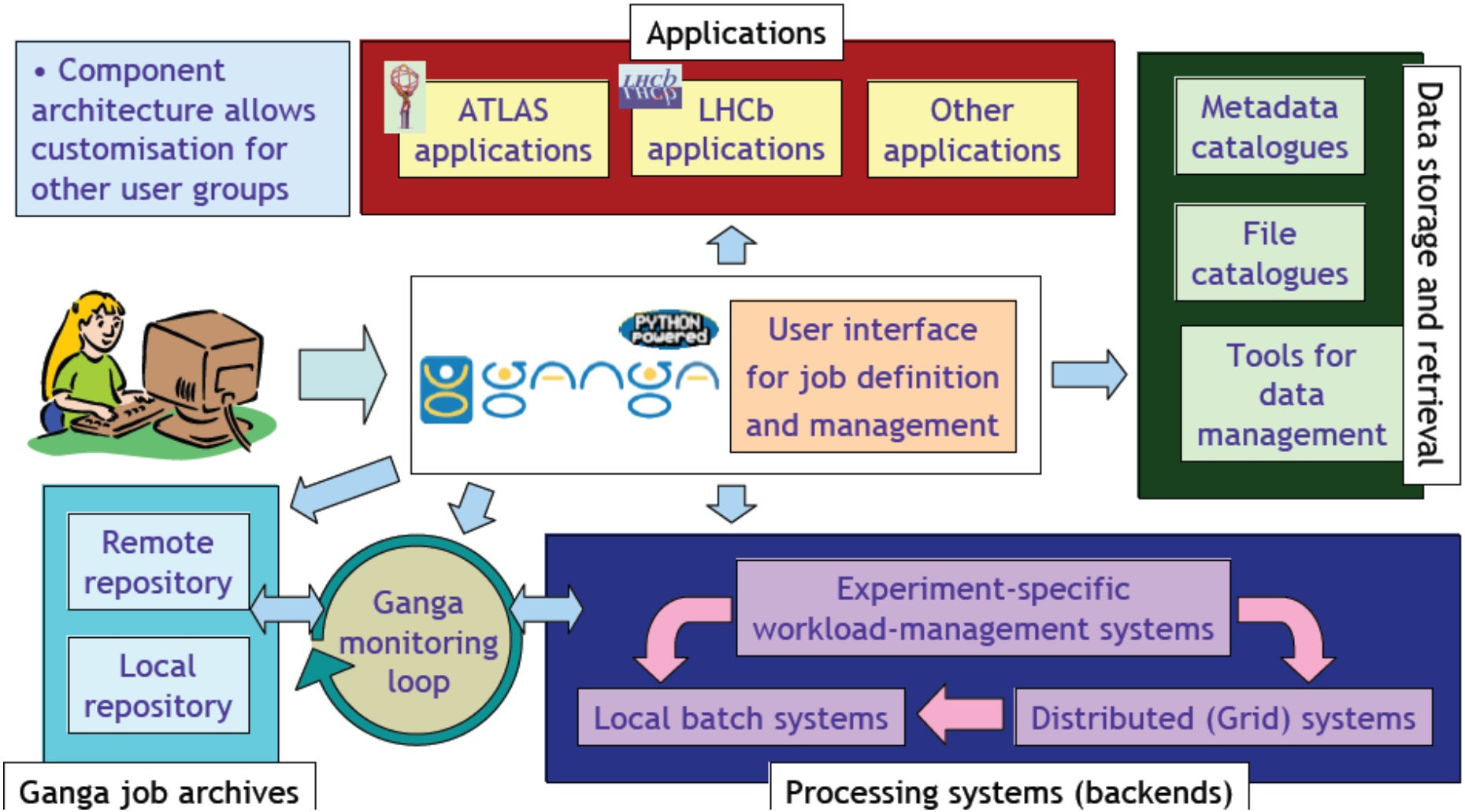


PFN (Physical File Name)

Path to a file at a specific site

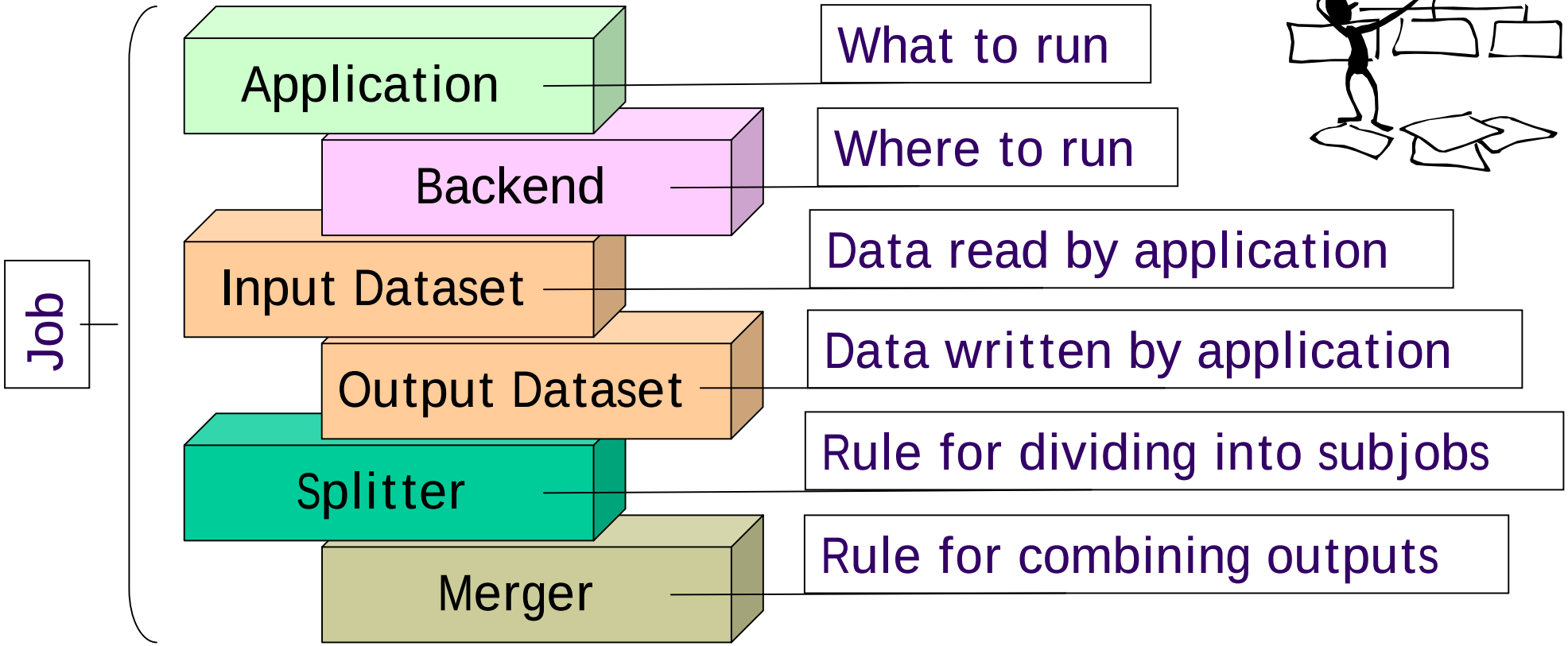


Grid Computing – software infrastructure



Ganga job abstraction

A job in Ganga is constructed from a set of building blocks, not all needed for every job



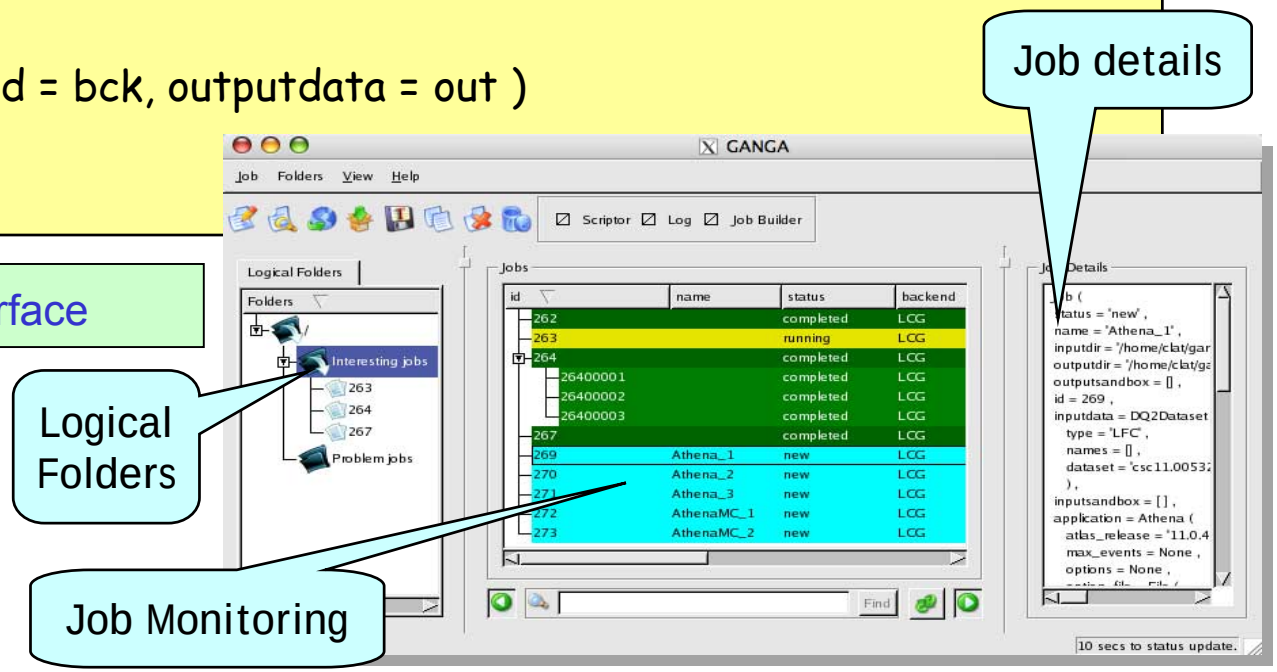
Dr Chris Town

Image-analysis jobs in Ganga

Ganga provides a command-line interface and scripting language, built on Python

```
# Define application to perform image analysis, specifying input
app = Classify( version = "2.0.1", imageList = "imageURLs.txt" )
# Define processing system where job will run
bck = LCG( middleware = "GLITE" )
# Define type of output data to be produced
out = CamontDataset()
# Create job
j = Job( app = application, backend = bck, outputdata = out )
# Submit job
j.submit()
```

Ganga also provides a graphical interface



In practice, use Ganga script:
automated job-submission
and checking 24 hours a day

Dr Chris Town

Initial Results

- 5 million images processed
- 6000 jobs completed
- Ported middleware to non-SL Linux
- Improvements to Ganga (features + usability)
- Job failure rates at 2% level, with two main causes
  - –Proxy credential of submitting user expires before job starts
  - –Network failures, preventing upload of results to storage element

Site monitoring

Example site monitoring: running jobs at Lancaster for 8-day period (July 2008)

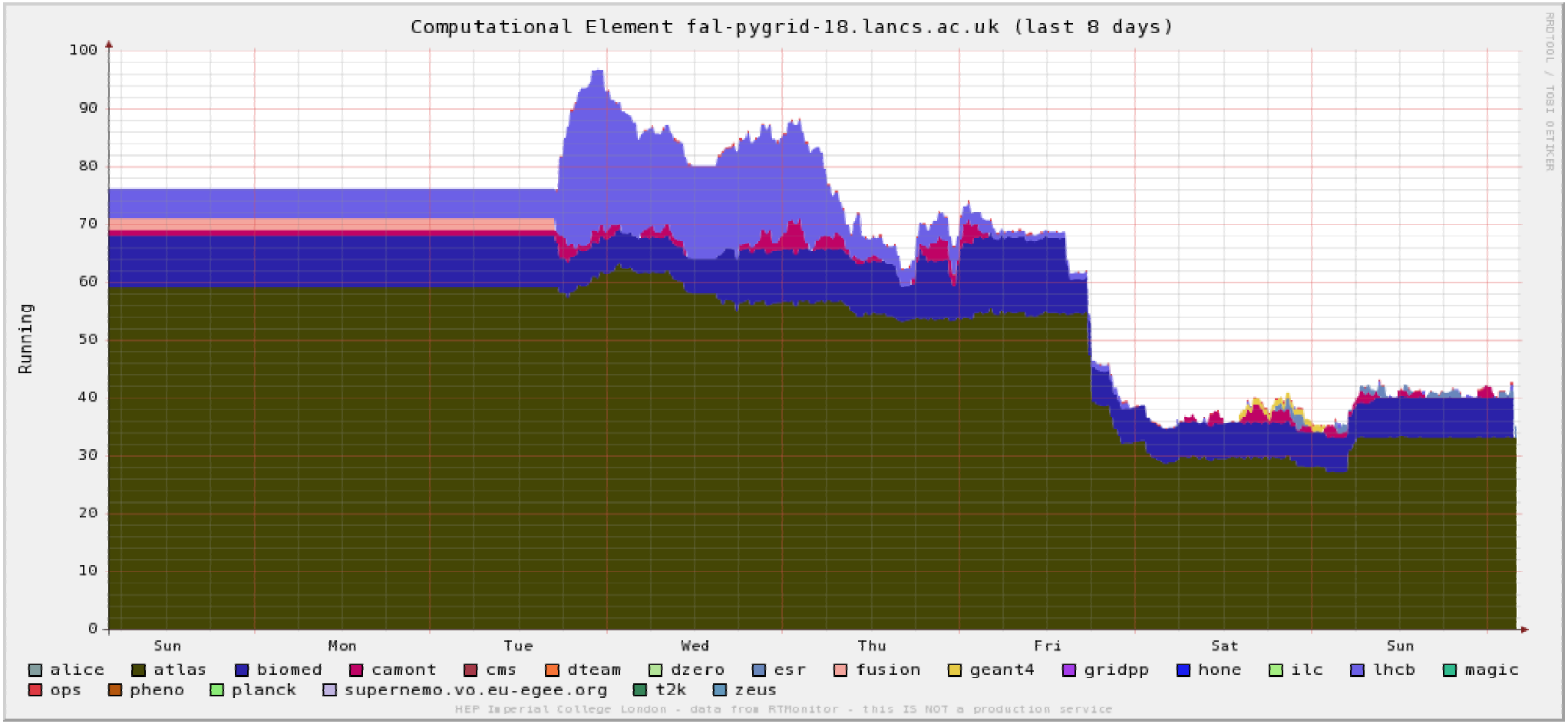


Image-processing jobs (camont) are small fraction of total

Dr Chris Town

Scaling to 18 million images

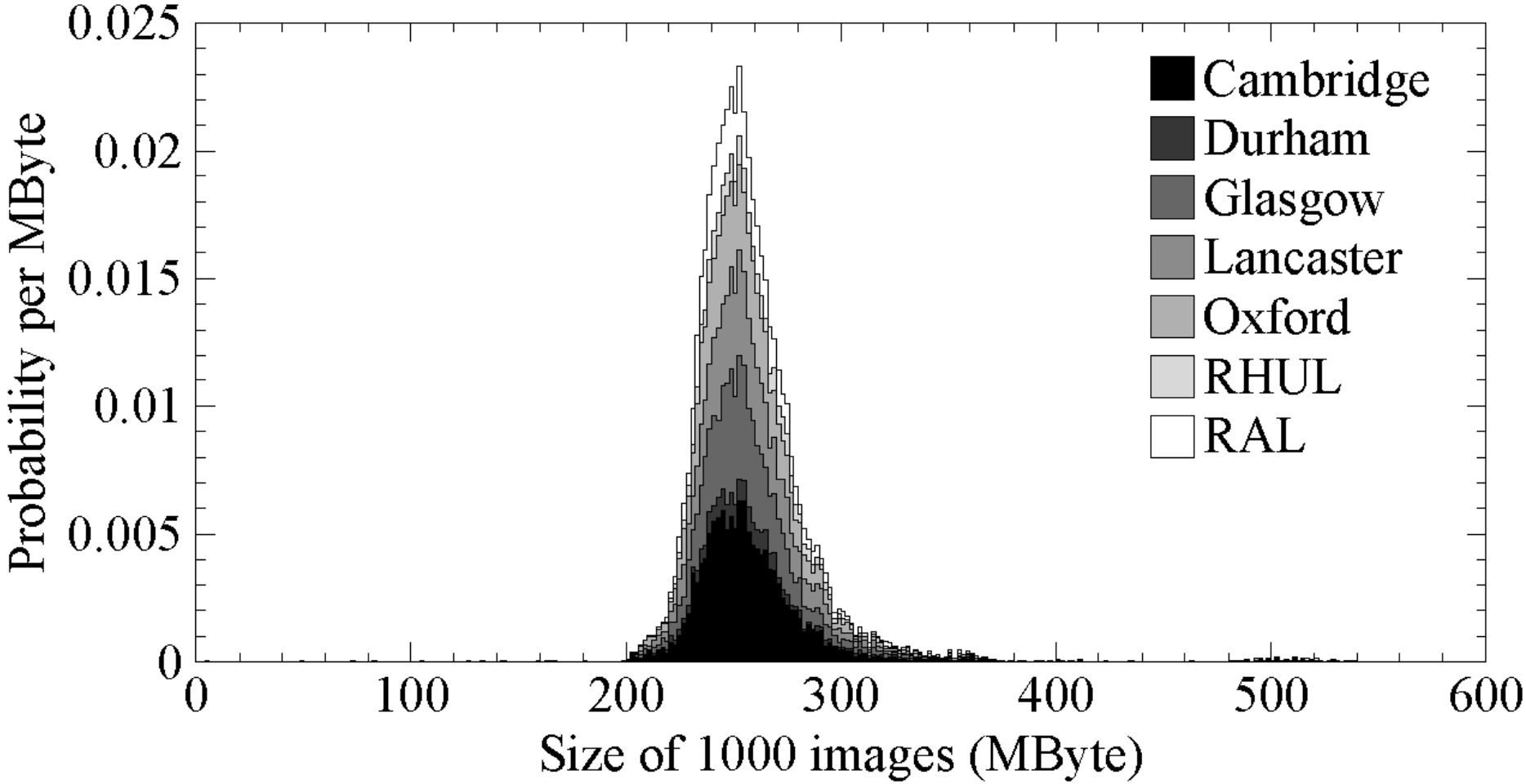
- Nine sites available to camont Virtual Organisation

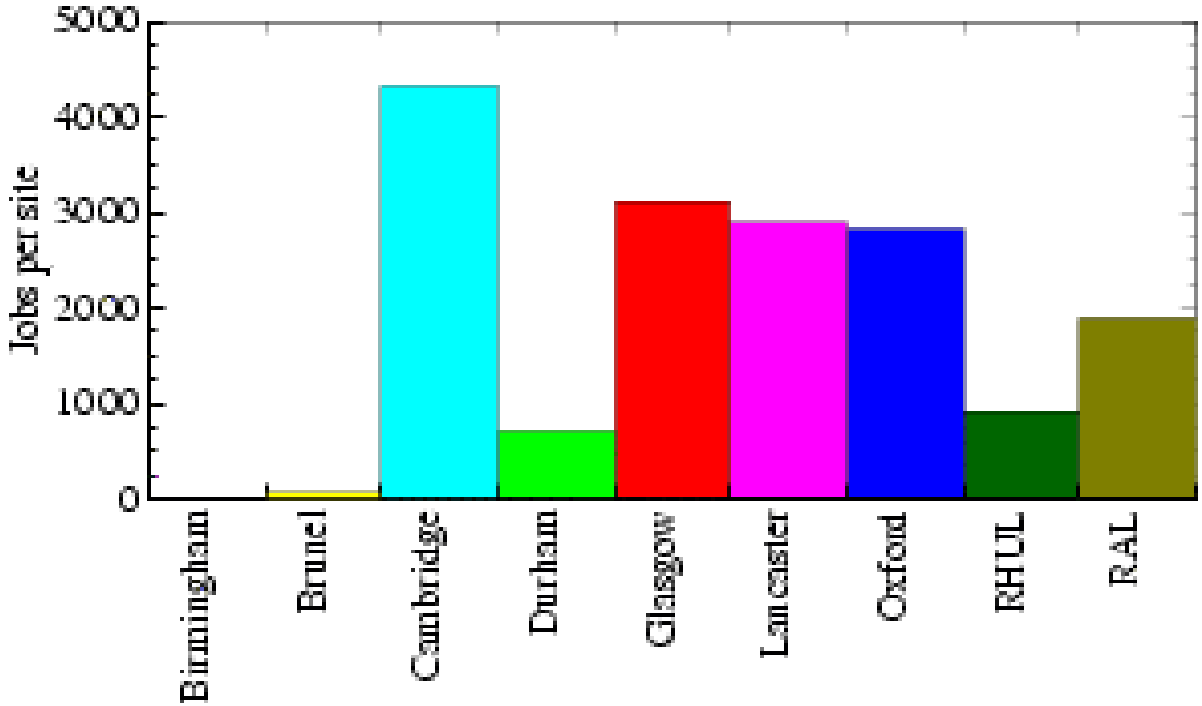
  Birmingham, Brunel, Cambridge, Durham, Glasgow, Lancaster, Oxford, Royal Holloway (RHUL), Rutherford Appleton Laboratory (RAL)
- One Ganga instance run on each image host
- Status of jobs at each site checked every 10 minutes, and new jobs submitted, all going via gLite workload-management system

- Conditions for submitting new job (1000 images):
  - Queued or running jobs at site < 100
  - Queued jobs at site < 30
  - Queued or running jobs at all sites < 400
  - Total failed jobs < 100

Performance Analysis

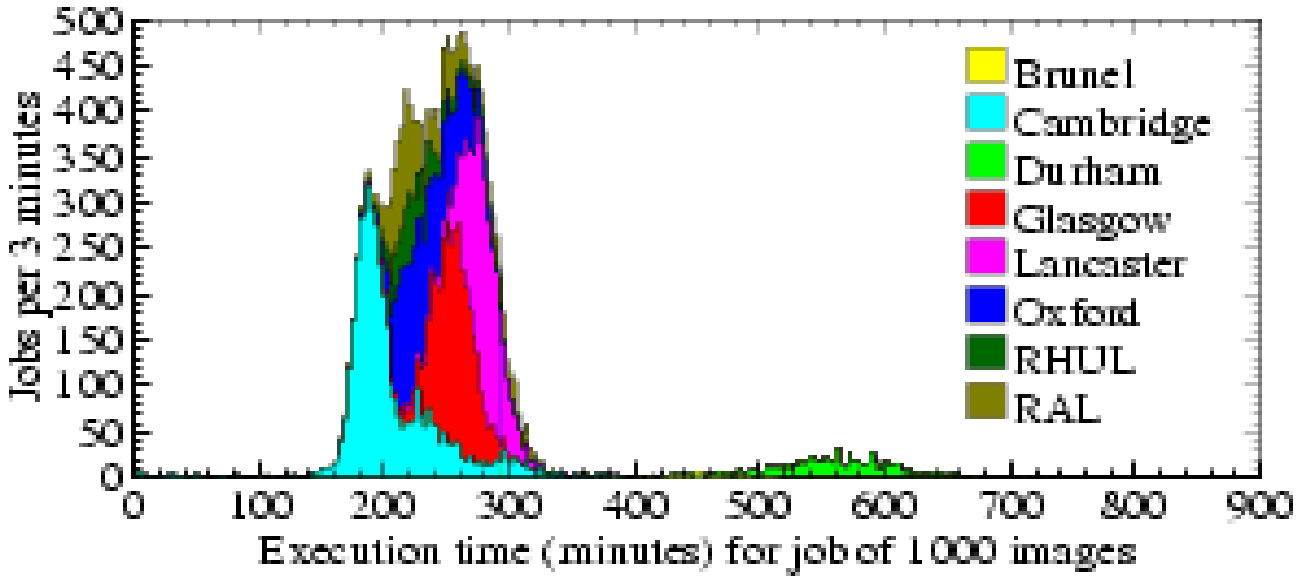
Distribution of image file sizes in each batch of 1000 images at each of the 7 primary sites.



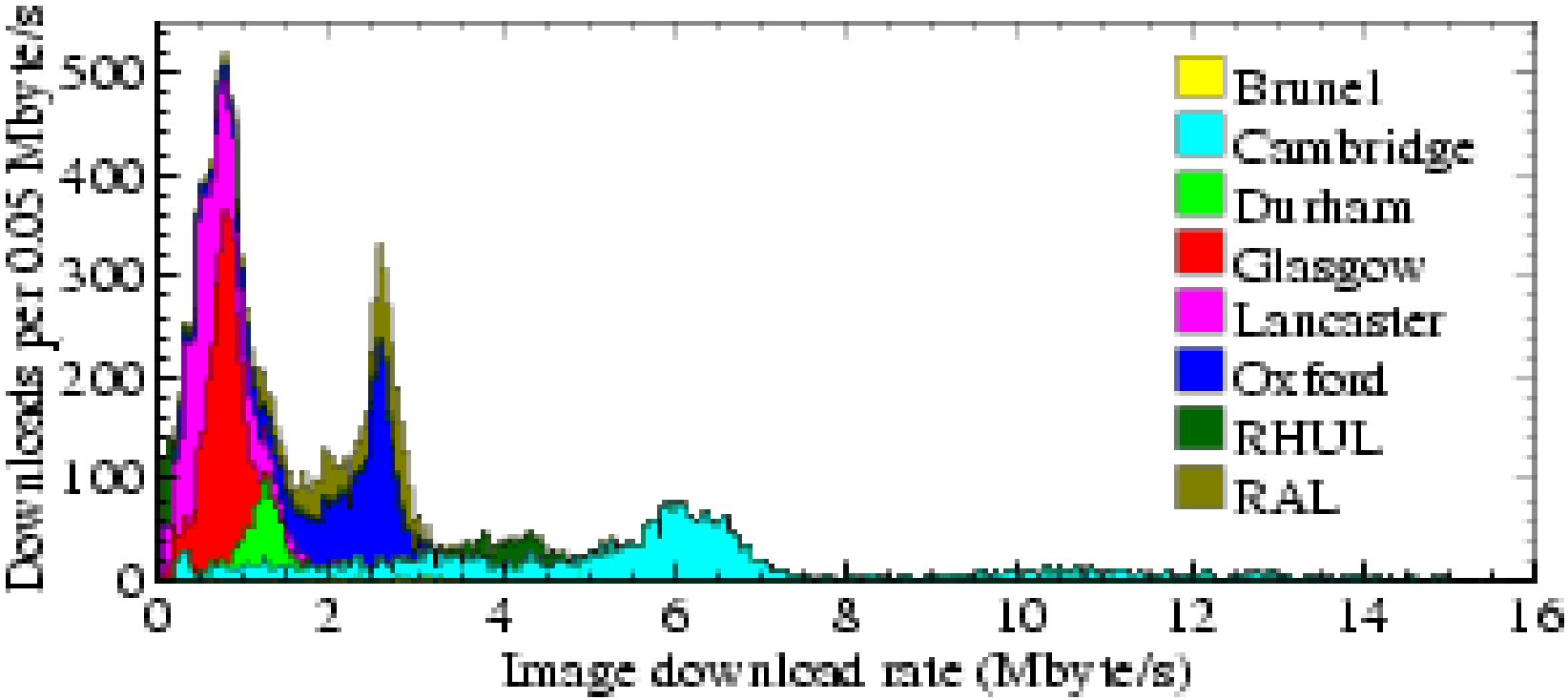


Job destinations and execution times

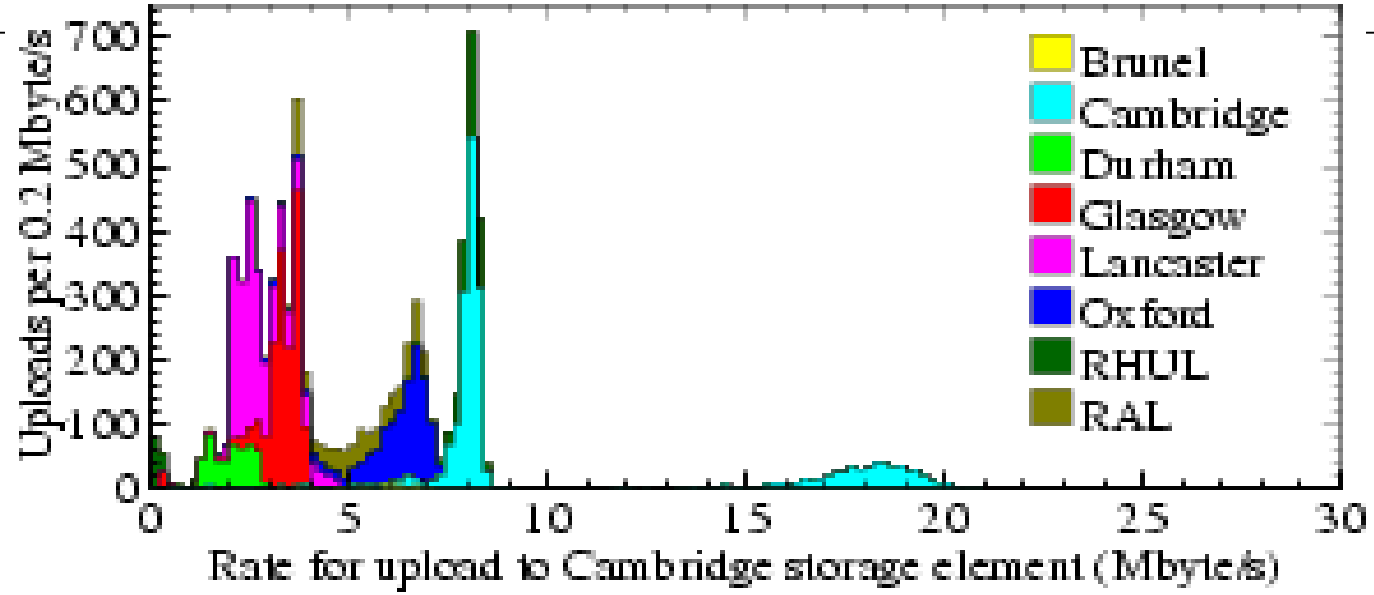
16733 jobs

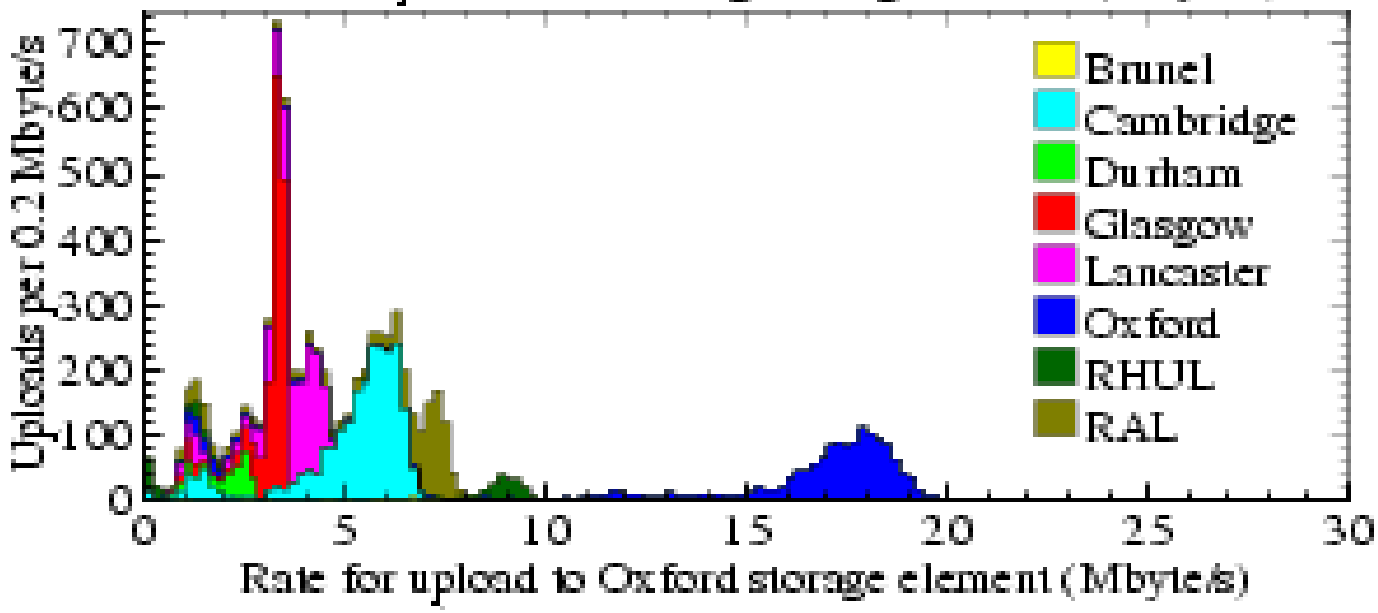


Data transfers to worker nodes

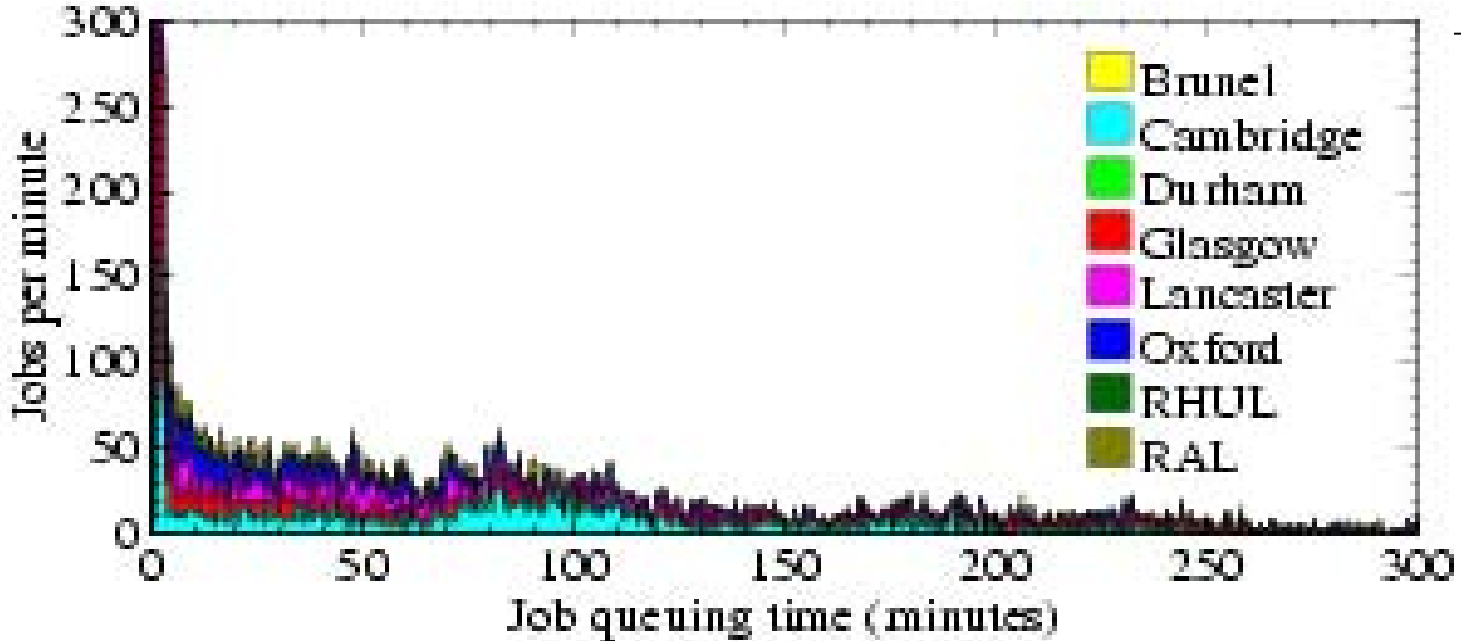


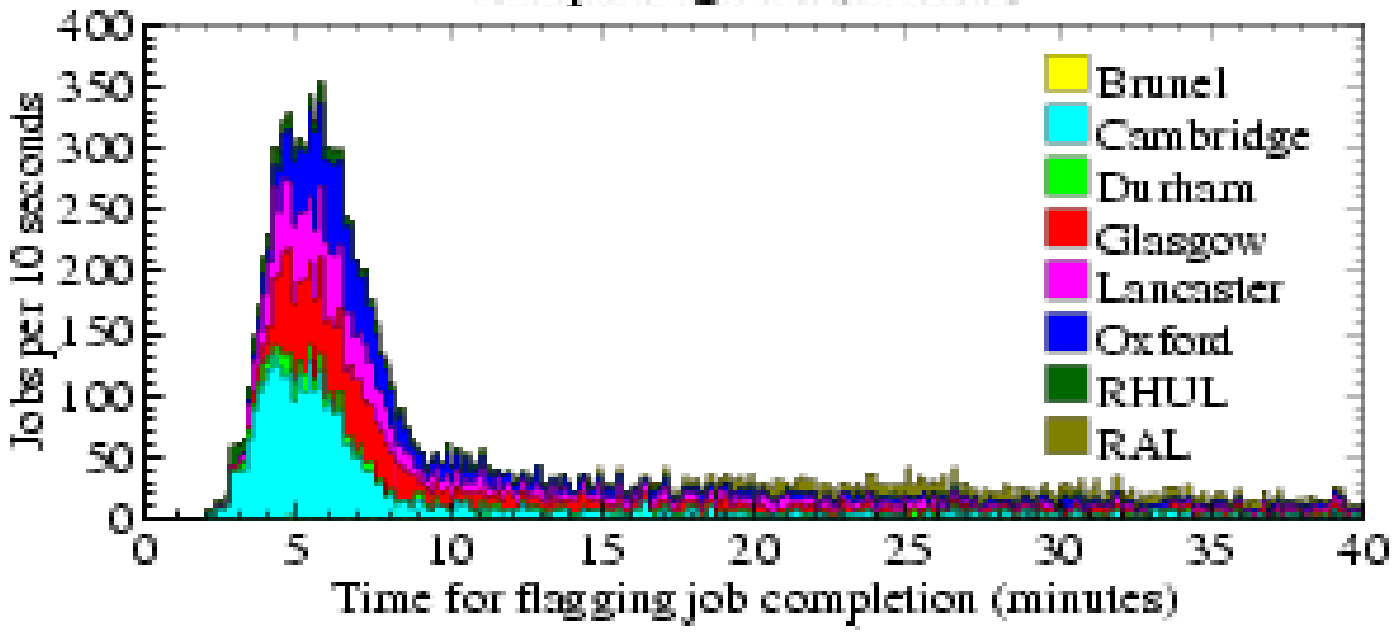
Data transfers from worker nodes to storage elements



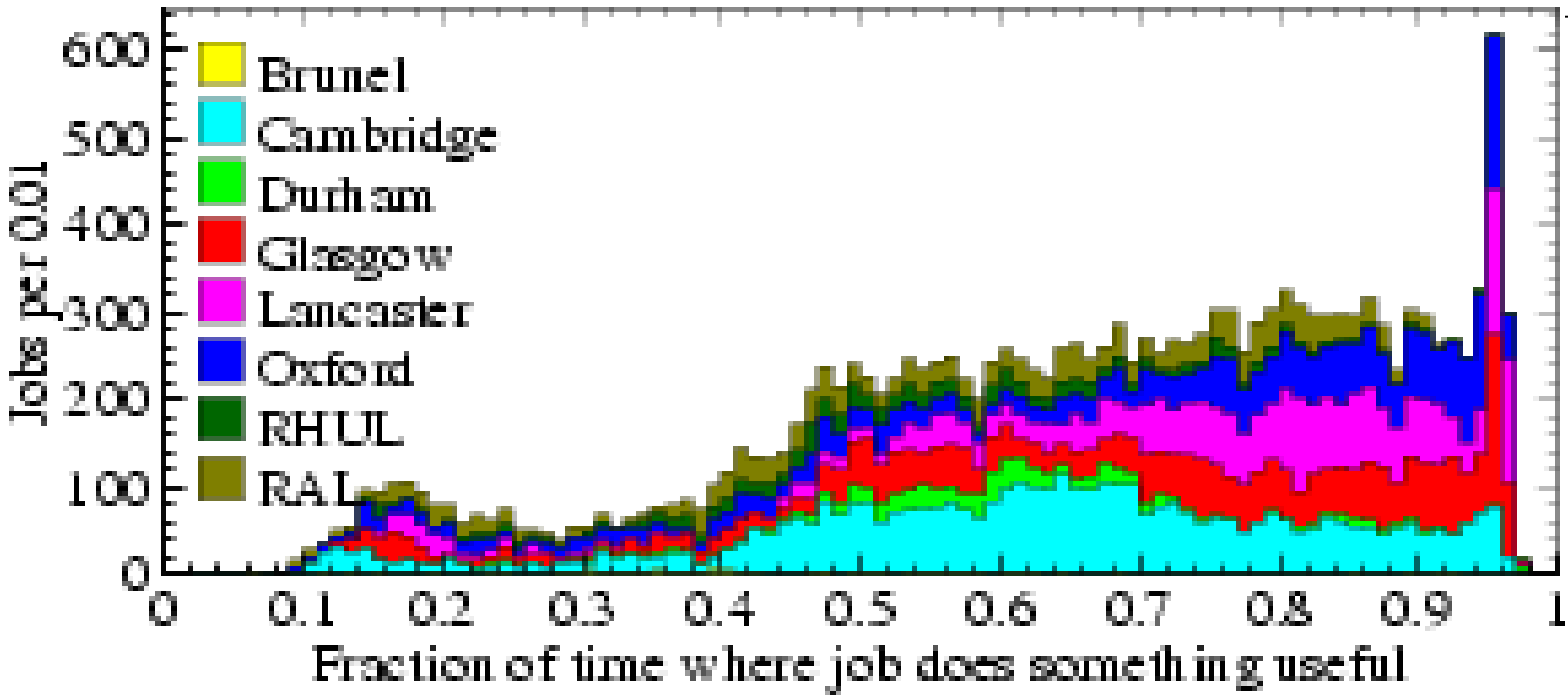


Waiting times





Efficiency



-Useful time calculated as time when job is downloading and processing images

- Grid overheads come from: startup time, system time for logging job completion, result upload and retrieval