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**

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**

Source: Berkeley, 2003
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

Scene classification
- Indoor, beach, sunset, nighttime, autumn

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

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

Semantic descriptor extraction
- Combine all information in index

Ontological Description Process
- Edges, pixel, colour, textures, shapes, shading etc (sun above mountain)

Statistical Class Matching
- Sunset 98%
- Clouds 98%
- Water 96%
- Mountain 90%

Index

Dr Chris Town
Imense approach: Image Retrieval

Text Search
"Sea with cloudy red sunset over mountain without people"

Language Processor
Uses synonyms and cloud, clouds, cloudy, sky, sun, sunset

Semantic Image Query Language
Understands semantics of queries (sunset is over mountain)

Index
Sunset 98%
Clouds 98%
Water 96%
Mountain 90%
People 0%
• There are more than $15 \times 10^9$ 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; -270°C
- 5000 SC magnets
- 700,000L liquid He
- 12,000,000L liquid N₂

Dr Chris Town
Data processing in particle physics

- 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

Scope and complexity of particle-physics experiments has increased in parallel with increases in computing power.
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
STFC funded Collaboration
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
Grid data management

LFC (Logical File Catalogue)
Database that maps Logical File Names to Physical File Names

LFN (Logical File Name)
Alias for any of one or more files (replicas) with identical content

PFN (Physical File Name)
Path to a file at a specific site

SE (Storage Element)
Mass-storage system at a Grid site

Dr Chris Town

Copyright 2009
Grid Computing – software infrastructure

- Component architecture allows customisation for other user groups

Applications
- ATLAS applications
- LHCb applications
- Other applications

User interface for job definition and management

Data storage and retrieval
- Metadata catalogues
- File catalogues
- Tools for data management

Remote repository
Local repository
Ganga monitoring loop
Ganga job archives

Experiment-specific workload-management systems
Local batch systems
Distributed (Grid) systems
Processing systems (backends)

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

- **Application**
  - What to run
- **Backend**
  - Where to run
- **Input Dataset**
  - Data read by application
- **Output Dataset**
  - Data written by application
- **Splitter**
  - Rule for dividing into subjobs
- **Merger**
  - Rule for combining outputs

Dr Chris Town
Image-analysis jobs in Ganga

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

```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

![Graph showing 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