Running GIFT image recognition software in XWCH

INTERNAL REPORT, THE VIRTUAL EZ GRID PROJECT

Marko Niinimaki, Mohamed Ben Belgacem, Nabil Abdennadher
hepia
January 2010

ABSTRACT: This report describes running a feature extraction task for a collection of images in the XWCH (XtremWeb-CH) platform. The image collection used in the task is "ImageCLEFMed", and the feature extraction software is GNU GIFT. Design, implementation, and performance measurements are reported.

Background

Content-based image retrieval is increasingly being used as a diagnostic aid in hospitals [MMB2004]. However, hospitals produce large amounts of images -- for instance the University Hospitals of Geneva radiology department produced about 70 000 images per day in 2007 [NZD2008]. Preparing these images in such a way that they can be used in diagnosis is a challenging task due to their volume. Content-based image retrieval systems typically use image features like properties of textures and colours [SMM1998]; here, we call the of extracting features from images *indexing*.

The well-known GIFT, or Gnu Image Finding Tool, software is a content-based image indexing and retrieval package was developed at University of Geneva in the late 1990's. GIFT utilizes techniques common from textual information retrieval and uses a very large collection of binary-valued features (global and local colour and texture features) [SMM1998]. GIFT extract these features and stores them in an inverted file. In a typical desktop PC, the speed of indexing is about 1 or 2 images per second.

The history of the ImageCLEFMed image collection can be summarized as follows: ImageCLEF started within CLEF (Cross Language Evaluation Forum) in 2003. A medical image retrieval task was added in 2004 to explore domain-specific multilingual visual information retrieval [MKK2009]. The ImageCLEFMed2007 used in this report consists of about 50 000 images, originally from radiological journals Radiology and Radiographics. The images are originals used in published articles.

XWCH [AB2007] is a cluster computing software developed at the hepia (la Haute Ecole du paysage, d'ingénierie et d'architecture) in Geneva. It consists of a coordinator node, worker nodes and warehouse nodes used for data storage. For this report, a command line interface was added in the software. This allows one to submit text-based job descriptions, query the status of jobs and recover the results of jobs, using a Java command line program.

In related research, parellelization of the indexing task has been studied in [NZD2008], where the ARC Grid middleware was used as the platform. The principle of parallelization there was the same as in this report (see the next section).

Design and implementation of GIFT indexing in XWCH

Indexing a set of images can be seen as an "embarrassingly parallel" problem, defined as follows: "a problem in which little or no effort is required to separate the problem into a number of parallel tasks. This is often the case where there exists no dependency (or communication) between those parallel tasks." [F1995] Therefore indexing a large set (S) of images can be done by dividing S into small subsets, sending the subsets together with processing instructions into processing nodes, and combining the output of the processing nodes. The workflow is illustrated in Fig 1:

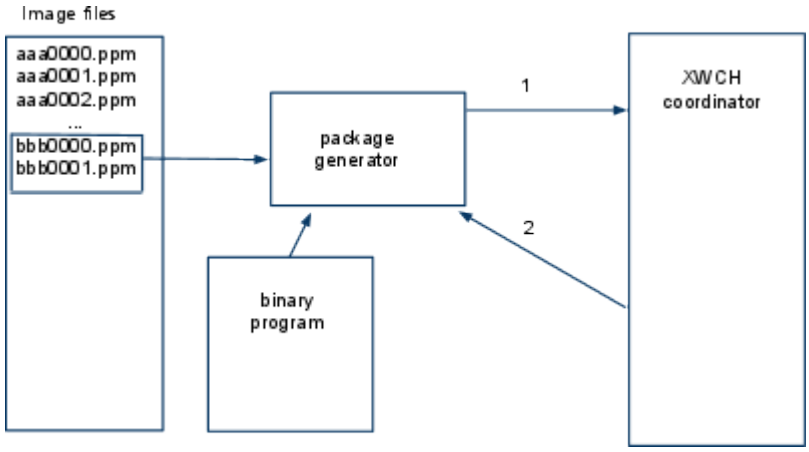

Fig 1: The workflow

In phase (1) the package generator creates "subsets" (zip archives) of the original set, adds a binary program, creates a corresponding job description, and submits them to the XWCH coordinator by using the command line interface. After the packages have been submitted, it (2) scans the jobs and for any finished job downloads the results.

The Package Generator

The package generator is a Perl script with the following functionality

- create a script called **local.sh** that will execute the indexing in each of the nodes (shown below)
- read the names of all image files in the given directory, add them to array **images**
- iterate all items in images. For every 1000'th image, and the last item, do as follows:
    - generate a tar package containing images
    - generate a zip package containing the tar package and **local.sh**
    - generate a xwch job description with the zip package as input (shown in below)
    - submit the job description, add the job name in array **jobs**
- repeat until **jobs** is empty
    - take the first entry **j** of **jobs**
    - check the status of **j**
    - if the status is COMPLETE, download the output files. Otherwise append **j** to **jobs**.

Local.sh

local.sh is a script executed locally in the worker nodes

```
/bin/tar xf images*.tar
/bin/rm images*tar
chmod +x geneva_gift
for i in *ppm
do
   ./geneva_gift $i
done
#/bin/tar cf features.tar *.fts
/bin/rm *ppm
```

Xwch job description

The xwch job description file lists the requirements, executable, input and output for an application. It is submitted by the xwch command line tool.

```
client_ID =xxxx-xxxx-xxxx..
applicationname =images1000
modulename =images1000
jobname =images1000
datainputfilename =images1000.zip
binaryinputfilename =local.zip
type=XWCHJob
os =LINUX
workercommandline= local.sh images1000
outputfilename=images1000.out
downloadresult=0
```

Results

The process of executing the package generator (=the entire packaging/submission/ result retrieving process) took 4 hours 53 minutes 58 seconds (=17638 seconds). This figure is comparable with those achieved by the ARC Grid middleware in [NZD2008]. Individual execution times of the **local.sh** script are shown in Figure 2. The short execution time of the last package is because it contained only 25 images. The average of the execution times was 1006 seconds (=16 min 36 seconds) and the sum of the execution times 51316 seconds. The figure of 51316 seconds (ca 14 hours) would thus roughly correspond with executing the whole task on a single CPU.

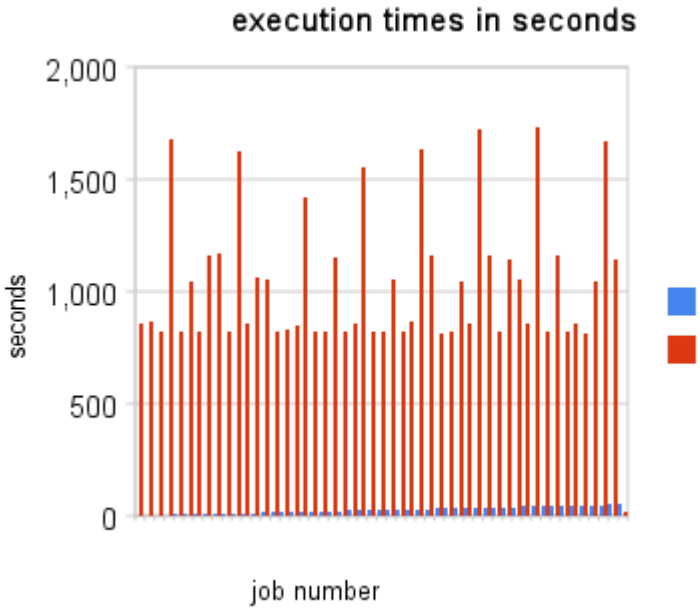

Fig 2: execution times of 1000 image package in xwch nodes.

Acknowledgements

We gratefully acknowledge that the right to use the ImageCLEFMed2007 was granted by Henning Muller on the basis that Marko Niinimaki was still a member of UNIGE's MedGIFT team in 2009.

References

[AB2007] N. Abdennadher and R. Boesch: A Scheduling Algorithm for High Performance Peer-to-Peer Platform, in Euro-Par 2006: Parallel Processing LNCS Volume 4375/2007, 2007.

[F1995] Designing and Building Parallel Programs, by Ian Foster. Addison-Wesley, 1995.

[MKK2009] H. Müller, J. Kalpathy-Cramer, C. E. Kahn Jr., W. Hatt, S. Bedrick and W. Hersh: Overview of the ImageCLEFmed 2008 Medical Image Retrieval Task, Evaluating Systems for Multilingual and Multimodal Information Access, LNCS Volume 5706/2009, 2009.

[MMB2004] H. Mueller, N. Michoux, D. Bandon, and A. Geissbuhler. A review of content–based image retrieval systems in medicine – clinical benefits and future directions. International Journal of Medical Informatics, 73:1–23, 2004.

[NZD2008] M. Niinimaki, X. Zhou, A. Depeursinge, A. Geissbuhler and H. Mueller: Building a Community Grid for Medical Image Analysis inside a Hospital, a Case Study, MICCAI Grid Workshop, New York University, September 2008.

[SMM1998] D. M. Squire, W. Mueller, H. Mueller, T. Pun: Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback. Technical Report 98.04, Computer Vision Group, Computing Centre, University of Geneva, 1998.