Oropo Executor

User Manual

COLLABORATORS			
	TITLE : Oropo Executor		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Maciej Smoleński	June 15, 2010	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
1	2010-04-19	First release	M. Smolenski
2	2010-04-20	Second release	M. Smolenski
3	2010-04-23	Link to picture processing script added to oropo solution section	M. Smolenski
4	2010-04-24	Installation and configuration appendix added	M. Smolenski

Contents

1 Introduction

Have you ever had a lot of data to process ? In such a moment after a while of processing we realize that it will take ages to complete. It would be faster if we could use two or three or even more computers. Lets use some computers - you think it is a lot of configuration ? You are wrong. With Oropo it's easy. Let's see.

It's difficult to talk about processing without an example. Let's discuss a problem of processing large number of pictures. First approach for solving this problem is to process pictures sequentially on one computer. Second approach is to process pictures parallelly on many computers.

2 Problem description

The problem is to process 10000 pictures. Each picture is in hight quality, the goal is to create a smaller version of each one. There is a library libjpeg that provides suitable programs.

USEFUL PROGRAMS FROM LIBJPEG:

djpeg decompress a JPEG file to an image file

cjpeg compress an image file to a JPEG file

Script signature for processing single picture:

- argument: path to the picture
- result: smaller version of the picture

Sample script in bash:

Example 2.1 Script make_smaller.sh

```
#!/bin/bash

QUALITY=30

if [ $# -ne 1 ]; then
  echo "arguments" 1>&2
  exit 1;
fi

FILE_PATH=$1

djpeg $FILE_PATH | cjpeg -quality $QUALITY
```

3 Processing sequentially

All pictures can be processed by invoking script make_smaller.sh for each picture.

Example 3.1 Processing sequentially

```
#!/bin/bash

MAKE_SMALLER=$PWD/make_smaller.sh
IMGS_DIR=$PWD/imgs
TARGET_DIR=$PWD/imgs_smaller

for file in $IMGS_DIR/*; do
  bash $MAKE_SMALLER $file > $TARGET_DIR/${file##*/}
done
```

4 Processing parallelly

We can process all picture using Oropo Executor system. Tasks for processing pictures will be added to a queue and processed parallelly on many computers. Each picture will be processed with script make_smaller.sh.

Example 4.1 Processing parallelly

```
#!/bin/bash

MAKE_SMALLER=$PWD/make_smaller.sh
IMGS_DIR=$PWD/imgs

for file in $IMGS_DIR/*; do
  oropo-system-pusher -p "string:bash" -p "path:$MAKE_SMALLER" -p "path:$file"
done
```

Processing results can be found in /var/lib/oropo/response/*/0 files.

5 Summary

In previous sections two approaches for processing pictures were presented. First approach uses single computer for processing. Second approach uses many computers for processing. Complexity of both solutions deployment is almost the same. With second approach processing will be completed faster.

A Oropo Project

A.1 General

Oropo Project home page: http://www.oropo.org.

A.2 Installation

To install Oropo you need to install Oropo System on central node and Oropo Executor on each node that will be used for processing (it may be central node also).

Oropo packages are located in oropo repository, you need to do these steps to be able to install packages.

A.2.1 Configuration on each node:

Example A.1 Add this entry to /etc/apt/sources.list file:

```
deb http://students.mimuw.edu.pl/~ms209495/oropo/debian sid main
```

Example A.2 Execute command:

apt-get update

A.2.2 Installing Oropo System on central node:

Example A.3 Execute command:

apt-get install oropo-system

A.2.3 Installing Oropo Executor on processing nodes:

Example A.4 Execute command:

apt-get install oropo-executor

A.3 Configuration

A.3.1 Configuration on central node:

Example A.5 Add yourself to oropo group to have sufficient permisssions:

adduser `whoami` oropo

Example A.6 Add processing nodes addresses to the Oropo System:

oropo-monitor-ctl --id_prefix oropomonitor --add node1_ip_address
oropo-monitor-ctl --id_prefix oropomonitor --add node2_ip_address
oropo-monitor-ctl --id_prefix oropomonitor --add nodeN_ip_address