UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

User Guide

PG CSI:PC^2

Supervising Tutors:

Prof. Dr. Marco Platzner
Dr. Jens Simon
Dipl. Inf. Tobias Schumacher

Members:

Samir Brulic
Samira Brulic
Pascal Deppe
Dominic Eschweiler
Rosbeh Etemadi
Klaus Herbold
Christoph Konersmann
Robert Meiche
Christoph Scholz
Nils Timm
Andre Ückermann
Elmar Weber
René C. Zorn

PADERBORN CENTER FOR PARALLEL COMPUTING

Paderborn Center for Parallel Computing (PC^2)
An institution of the University of Paderborn

Paderborn on May 20, 2008

Project Group CSI:PC²
University of Paderborn

AUTHORS:	Samir Brulic	samir.brulic@csipc2.de
	Samira Brulic	samira.brulic@csipc2.de
	Pascal Deppe	pascal.deppe@csipc2.de
	Dominic Eschweiler	dominic.eschweiler@csipc2.de
	Rosbeh Etemadi	rosbeh.etemadi@csipc2.de
	Klaus Herbold	klaus.herbold@csipc2.de
	Christoph Konersmann	christoph.konersmann@csipc2.de
	Robert Meiche	robert.meiche@csipc2.de
	Christoph Scholz	christoph.scholz@csipc2.de
	Nils Timm	nils.timm@csipc2.de
	Andre Ückermann	andre.ueckermann@csipc2.de
	Elmar Weber	elmar.weber@csipc2.de
	René C. Zorn	rene.zorn@csipc2.de

P1.7.01.6, University of Paderborn
Warburger Str. 100
33098 Paderborn, Germany
http://www.cs.uni-paderborn.de/fachgebiete/computer-engineering-group.html

Published by the Project Group CSI:PC²
http://csipc2.sourceforge.net

Contents

1 Download

To do biometrical identification with the software of the CSI:PC^2 project group you can get the sourcecode in several ways:

- Get a tarball or zipfile of a release version. Go to the download section of `http://www.csipc2.de` and get the latest software release package.
- Get the cutting edge version via subversion. Checkout the latest sourcecode with `svn co https://csipc2.svn.sourceforge.net/svnroot/csipc2/trunk csipc2`. Anonymous logins just have read access to this subversion repository.

At continued reading we will define `/path_before_csipc2/csipc2` as the directory to the sourcecode.

If you are interested in supporting this project by submitting bugreports or bugfixes in form of patches, please ask at our mailinglist csipc2-devel@lists.sourceforge.net and you might get write access to the repository.

After getting the sourcecode please continue with one of the following sections concerning the installation of the Cluster Control Daemon (CCD) (section 2) and/or the Frontend (section 4).

2 Cluster Control Daemon (CCD)

2.1 Prerequisites

The Cluster Control Daemon (CCD) is part of CSI:PC^2 task force. Before you start compiling the CCD you should consider the following prerequisites.

Cause the CCD is designed for performant parallel matching of large databases with millions of biometrical information (by now fingerprints and/or iriscodes) we require high performance state of the art servers as clusters or supercomputers. Considering a database with two million templates of each supported matching method, the minimal hardware which would be required is shown in table 2.1.

Component	Required
Architecture	Intel® compatible (64bit preferred) architecture
Cluster nodes	$4 - 8$
CPU cores per node	minimum of 2 with 2.0 GHz
RAM per node	$>$ 2 GBytes
Network	Gbit-Ethernet, Infiniband/SCI (for MPI)
Storage	4Gbyte for database (e.g. NAS, SAN, GPFS)

Table 2.1: CCD hardware requirements

The table 2.2 shows what software packages are required for successfull compiling and running.

Component	Required
Operating system	Unix/Linux (2.6.x)[1]
Compiler	GCC (>3.4), Intel® Compiler (9/10.x)
MPI interconnect	MPICH, MVAPICH or SCAMPI
FPGA	installed libs and headers, e.g. admxrc2 for Xilinx
image processing	ImageMagick (Version 6.3.3)[2]

Table 2.2: CCD software requirements

2.2 Compilation

For compiling the sourcecode the following directory structure will be used in our software:

```
/path_before_csipc2 - path to where you unpacked the source
  + ./csipc2        - sourcecode directory
  + ./build         - compilation directory
  + ./dist          - prefix/install directory
  + ./dist/usr/doc  - documentation directory
```

In ./csipc2 exists a configuration script config.sh which does the following tasks automatically for you:

- create build and install directories if necessary
- generate configure and Makefile scripts with autotools
- execute ./csipc2/configure [options]

You may pass configure options to config.sh. Specific configure options are listed in table 2.3.

Option	Description
--with-wand-config=FILE	FILE points to Wand-config of the ImageMagick library
--with-mpicc=FILE	FILE points to the compiler of the specific MPI interconnect you want to use
--with-admxrc2-header=PATH	PATH to Xilinx headers
--with-admxrc2-lib=PATH	PATH to Xilinx libs
--enable-fpga	enable support for FPGA's
--enable-debug	enable debugging mode (definitely more verbose)
--enable-gtk2	build utilities using the gtk2 library
--enable-sse	enable SSE optimizations
--enable-sse2	enable SSE2 optimizations
--enable-sse3	enable SSE3 optimizations

Table 2.3: Configure options

We assume in the following examples that ImageMagick is installed in /opt/ImageMagick and MPICH using GCC is installed in /opt/mpich-gcc and further more that those will be used for compilation. If you want debugging enabled and want to have it installed in /opt/csi a sample compilation could be:

```
# cd /path_before_csipc2
# ./csipc2/config.sh --with-wand-config=/opt/ImageMagick/bin/Wand-config \
                     --with-mpicc=/opt/mpich-gcc/bin/mpicc \
                     --enable-debug --enable-sse --enable-sse2 \
                     --enable-gtk2 --prefix=/opt/csi
# cd build
# make
```

To build the source within an SRPM–Package (not available yet) just type:

```
# rpmbuild --rebuild csipc2_ccd.srpm
```

2.3 Installation

After compiling the sourcecode just type the following to install the CCD:

```
# cd /path_before_csipc2/build
# make install
```

If available you also could install binaries from RPM–Packages with:

```
rpm -ivh csipc2_ccd.x86_64.rpm
```

2.4 Execution

To execute the CCD we assume that the software has been installed on every node in the same directory, ideally it is installed on a shared directory (we choose `/shared` here). The CCD should be started with `mpirun` of the corresponding MPI interconnect (`/opt/mpich-gcc/bin/mpirun` for example). The supported CCD options are listed in table 2.4.

Option	Description
`-c FILE`	points to the configuration FILE (default: /etc/csipc2/ccd.cfg)
`-l FILE`	points to the logging FILE (default: /dev/null)
`-d PATH`	PATH where the database is stored (default: ./db)
`-b PATH`	PATH to the FPGA bytecode images
`-H #`	number of headnodes to use
`-B`	enable benchmark mode
`-h`	print help

Table 2.4: CCD options

CLI options have higher priority than configfile options. Use absolute pathnames only!

Sample execution:

```
# /opt/mpich-gcc/bin/mpicc -np 8 -- /shared/csipc2/bin/ccd \
  -c /shared/csipc2/etc/ccd.cfg \
  -d /shared/db \
  -l /tmp/ccd.log \
  -H 4 \
  -B
```

In the given example ccd is executed on eight nodes. The configfile is `/shared/csipc2/etc/ccd.-cfg`, logging will go to `/tmp/ccd.log` and benchmarking will be enabled. The database ist stored at `/shared/db` and number of four (`-H 4`) headnodes will work on it.

2.5 Known Problems

The following problems are known in Version 1.0:

CCD breaks using ScaliMPI or MVAPICH

After doing some MPI communication some nodes stop working and receiving messages. We are currently working on a solution. As a workaround you might use MPICH or send us a patch solving this issue.

Missing Wand-Magick.h

You got the wrong ImageMagick version. Please use version 6.3.3 or above.

CCD uses 100% CPU time

This problem relates to the MPI interconnect waiting for MPI messages. By now there is no workaround planed.

3 Cluster Tools

3.1 Csi_dbtool

The first task, that should be done with this tool, is to convert an existing database to an equivalent database with a different size of headnodes. Headnodes are the nodes which are fetching the Data from the cluster-filesystem and distribute them over the MPI-Interface. The number of headnodes should be equivalent to the number of ethernet-links (or whatever) the filesystem-cluster has.

The second task, which could be done with this tool, is to extract very large collections of fingerprint-pictures to the template-database. To absolve this the tool knows two different modes which would be detected automatically. The first mode applies if the tool is started in a non MPI-Environment or if the number of ranks is smaller then two. In this case the tool acts in a non parallel manner because the extraction-process would not profit from a parallel processing. If the tool is started in a MPI-Environment with a rank-size greater than two, the tool would act in parallel where rank zero is the node for task management and template storing. All other ranks are used for computing the extraction. If you wish to compute on n ranks for example, you should start the tool with $n+1$ ranks.

Please make sure that all paths can be accessed on all nodes through the same path (on a network filesystem for example).

Overview over the commandline-arguments :

```
-i input dir                        (Database or picture{}-collection
input directory)
-o output dir                       (Database output directory)
-c <CREATE> <CONVERT> <EXT\_FINGER>  (The tool can * a database)
-n headnodes                        (Number of headnodes)
-t <ir> <fp> <db>                   (Database mode for creating,
ir = Iris, fp = fingerprint, db = triple\_dp [combined database for iris
and fingerprints] )

--help                              (printout a help message)
```

* Operation-Modes :

CREATE : Create a new and empty database. `-o -n` and `-t` option must be given :

```
csi_dbtool -c CREATE -o /mnt/test -n 5 -t db
```

CONVERT : Converts an existing database to an equivalent database with a diverent number of headnodes. `-i, -o` and `-n` option must be given :

```
csi_dbtool -c CONVERT -i /mnt/test -o /mnt/test2 -n 3
```

EXT_FINGER : Extracts an picture-collection from the input directory to an database. If there is no database in the target directory, it creates one :

```
mpiexec -n 5 csi_dbtool -c EXT\_FINGER -i /mnt/picture -o /mnt/picture/db
```

3.2 Clustering

Before the clustering matching can be used, the database has to be prepared, the so called clustering. Therefore, two tools exist. The *clustering* is the sequential programme the other is the parallel version called *clustering_mpi*. They are used the following way:

Clustering
To start a sequential clustering the following informations have to be given to the programme:

1. Database directory
2. Nr. of database-files in the directory
3. Nr. of clusters
4. Nr. of bins

If for example the database which is located in */fingerprintdata/fp/cluster* (and includes to db-files) has to be divided into 45 clusters with each 10 bins, following command has to be used:

```
./clustering /fingerprintdata/fp/cluster 2 45 10
```

The number of clusters and bins can be freely chosen but for efficency the number of clusters should be the square root of the number of templates in the whole database. The number of bins should be between 8 to 10.

Parallel clustering
Because clustering can be very time intensive, a version based on MPI exists. Following arguments have to be set:

```
-c CONFIG-FILE      The path to the database and the number of headnodes
                    are the used options from that file (can be the same
                    as used for the CCD)

-nc NR-OF-CLUSTERS  The number of clusters
-na NR-OF-ARD-BINS  The number of bins
```

After a clustering process is successfully done, a new file (clustering_metadata.dat) exists at the database path. There all important information for the matching is saved.
Using the data from the above example, the parallel clustering would be started with:

```
[MPI-EXEC-COMMANDS] ./clustering_mpi  -c parameters.cfg -nc 45 -na 10
```

The MPI-EXEC-COMMANDS depend on the used platform.

Reading the metadata
For getting the information, which is created during the clustering process (centroids etc.), the tool *read_clusteringmetadata* can be used.
Usage:

```
./read_clusteringmetadata -f PATHTO/clustering_metadata.dat  [-a] [-c] [-cc]
```

Arguments:

```
-a    shows the center of the bins of each cluster
-c    shows the count-array (count[i] = nr of elements in cluster i)
-cc   shows the cluster centroids
```

4 Frontend

4.1 Installation

The frontend contains the graphic user interface (GUI), the scanner modules for iris and fingerprint and the feature extraction modules. With this setup the GUI, which controls all the modules on frontend side and manages the connection to the cluster control deamon (CCD), will be installed.

To use the sensors, the drivers for the iris and the fingerprint scanner must be installed. The drivers are provided by IrisGuard and TST. We are not allowed to make them available on our product.

With the installation of the TST driver a service is started. To use the fingerprint sensor this service must be disabled. To do this, call `systemcontrol` → `administration` → `services`. A list of services will be displayed. You have to search for `TST CoreService` and `TST DeviceManager`. Select one of it and click on the right mouse-button. Select `properties`. A window will be opened. Select starttype `disabled` and close the window with `ok`. Then click on the right button again and choose `stop`. Repeat the same for the other service.

To install the GUI just click on the `install_csi.exe`. This self extracting file will install all necessary files for the gui. After that copy the `IFA_APIX.ocx` and `IFA_API.exe` to the Windows/system32 folder and run the command `regsvr32 IFA_APIX.ocx`. The fingerprint scanner and the feature extractions for iris and fingerprint can be called independently.

4.2 Configuration

Usually you do not need to configure the frontend system. If it is necessary to reconfigure make your changes in the `config.ini`.

In this section the single parameters of the ini-file will be explained.

The first block **`server`** contains the parameters `name` and `port`. These two parameters represent the server name and the connection port. You should use a secure connection. In this case the servername is the localhost (127.0.0.1). To build a tunnel for a secure connection use a ssh client (available for free on www.ssh.com).

The second block **`extraction`** determines if the extraction runs on the frontend or on the cluster. To run the extraction on the frontend set `on_frontend=1`, otherwise `on_frontend=0`.

The block **`fef`** stands for "feature extraction finger". The parameter `path` contains the path to the finger feature extraction module. The standard module is installed with the GUI.

The block **fei** stands for "feature extraction iris". The parameters `path` and `path2` contains the path to the finger feature extraction module. The standard module is installed with the GUI. Additionaly, the parameters of the feature extraction are saved in this string. For that reason there are two strings. The iris feature extraction generates a preview image first. With a second call, the extraction is done.

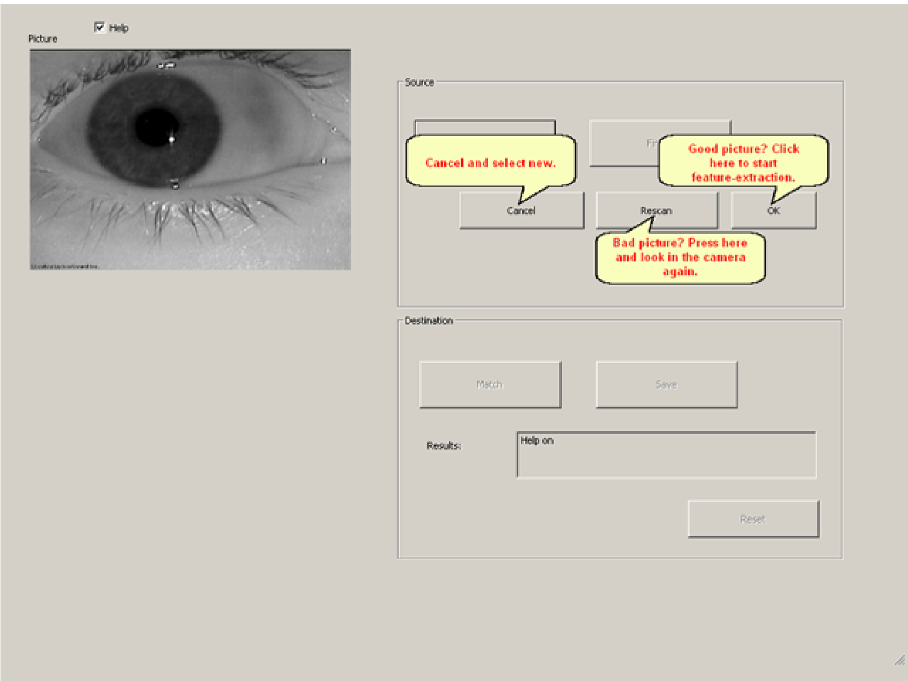
The block **mf** stands for "maching finger". There are 4 different matching modes for the fingerprint matching (minutiae, cluster, filterbank and mixed). If `mode` contains a 0, the default is selected. This is setup on cluster side. Use the 1 for minutiae, the 2 for cluster, the 3 for filterbank and the 4 for the mix of minutiae- and clustering-matching.

The block **irissensor** determines which type of iriscamera is used. Set a 0 to use the Irisguard camera and the 1 for all other kinds.

The block **fingersensor** contains the path to the scanner module for fingerprints. The TST module for the bird3 sensor is installed with the GUI. To use an other scanner, replace this path.

4.3 Execution

To start the GUI execute the `gui.exe` file. The application starts in a window. The window is just for testing. By clicking the maximization button of the window, the window becomes a full screen application. The GUI is optimized for public use with a touchscreen. The full screen application can´t be closed by the user. To close the application, disable the help and press `ESC`.



The checkbox `Help` disables or enables the help function of the GUI. If the box is checked, tooltips will appear on the active buttons. These tooltips tell what to do. If the activation of a button makes no sense in the actual state, the buttons are disabled. So, only the clickable buttons have tooltips. No more than 3 buttons are active at the same time. The usability of the application is given.

In first case you have to decide if you want to scan an iris or a fingerprint. Click the button `iris` to scan an iris or the button `finger` to scan a finger. These buttons start the corresponding sensor. After clicking `iris` look in the camera. One eye should be visible in the mirror. If you are too close or too far away, the camera will tell you. If you do not use the IrisGuard camera the photo is not shot automatically. You have to click in the preview window to shoot the picture. After clicking `finger` press your finger on the sensor. The sensor starts blinking. After scanning the scanned picture will appear in the GUI.

In the second case you can decide between 3 buttons. Click `cancel` to return to the first state. After clicking `cancel` you can select again if you want to scan a finger or an iris. If the quality of the picture is bad, click on `rescan`. After clicking `rescan`, use the sensor again. If the picture is good, press `OK`. After clicking `OK` the feature extraction will start, if extraction on frontend is set up. Otherwise the click is just a confirmation of the picture. If you extract an iris on frontend side, you have to click the `OK` button twice. After the first click a preview image is shown. The boundaries of the eyelids, the white area of the eye (sclera) and the black area of the eye (pupil) should be correct. If they are not, rescan the eye. To close the preview, click on the screen. The second click will extract the rest of the eye.

Now you can save or match the template, which the GUI has created in the last step. Click on `save` to send the template to the database. A message with the template number will be returned. To match the template against all other templates in the database click on `match`. The return message will contain the result of the matching. Click on `reset` to clear the textfield and start a new job.

4.4 Known Problems

The GUI does not start: Copy the `IFA_API.exe` and the `IFA_APIX.ocx` to the Windows/system32 folder and run the command `regsvr32 IFA_APIX.ocx` to solve the problem.

Copyright