9-Communicating applications

Advanced Computing Tools for Applied Research
(*Herramientas Computacionales Avanzadas para la Investigación Aplicada*)

Rafael Palacios, Jaime Boal

Advanced Computing Tools for Applied Research
Contents

Communicating applications

1

Introduction

Introduction

- Why communicating applications?
  - We can use the best application for each task
  - We can solve the same problem using a different approach
  - We can move the data into the environment where we feel more confident (ex. Matlab, ex. Excel)

- In the case of machine-to-machine communication, we can access data remotely

2

Unix command line

Unix command line

- This is useful for automating data pre-processing
- Only valid for text-based applications
- It is available in several systems:
  - Linux
  - Mac OS X
  - Other Unix: Solaris, HP-UX…
  - DOS with many limitations

Unix command line

- Unix can redirect the standard input and the standard output of any program

```
#prog

#prog >out.txt

#prog <in.txt >out.txt
```

Unix command line

- Unix can redirect the standard input and the standard output of any program

```
#prog

#prog >out.txt

#prog <in.txt >out.txt
```

input from keyboard

output to screen

Unix command line

- Unix can redirect the standard input and the standard output of any program

```
#prog

#prog >out.txt

#prog <in.txt >out.txt
```

input from keyboard

output to file **out.txt**

Unix command line

- Unix can redirect the standard input and the standard output of any program

```
#prog

#prog >out.txt

#prog <in.txt >out.txt
```

input from file **in.txt**

output to file **out.txt**

Unix command line (pipe)

- Unix can convert the standard output of one program into the standard input of another

```
#prog1 | prog2 > out2.txt
```

- Example. Command du to compute storage requirements of each folder, and the command sort to display ordered by size.

```
#du –ks * | sort -n > big_folders.txt
```

Unix command line

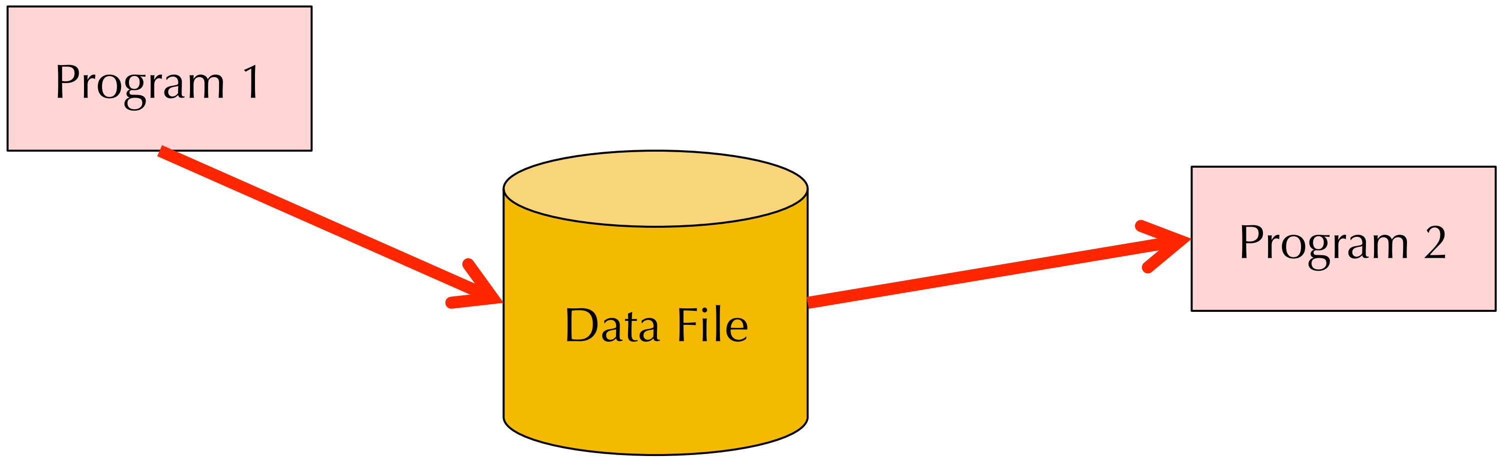
- There are many command to manipulate text
  - `head` ← Extracts lines from the beginning
  - `tail` ← Extracts lines from the end
  - `grep` ← Extracts lines according to a given pattern
  - `cut` ← Extracts columns
  - `tr` ← Replaces one character for another (example comma by TAB)
  - `sed` ← Useful for replacing strings (example "OFF" by "0")
  - `sort` ← Sorts lines
  - `curl` ← Downloads data from websites
- This is very useful for pre-processing data
  - Very efficient
  - Completely automatic
  - Commands could be stored in scripts

3

File-based data exchange

Files

- The most common way to transfer data from one program to another is to store such data into a temporary file.
- Sometimes menu options for export and import are available.



Files

- Some typical formats:
  - CSV (comma separated values)
  - TAB separated values
  - XML (Extensible Markup Language)
    - Most robust method
    - Very flexible
    - Very inefficient
  - Binary formats. Only for compatible applications
  - Other common specific formats:
    - XLS Excel
    - SHP (ESRI Shape Format, ARCview)
    - KML/KML (Google Earth)

Files – some comments on performance

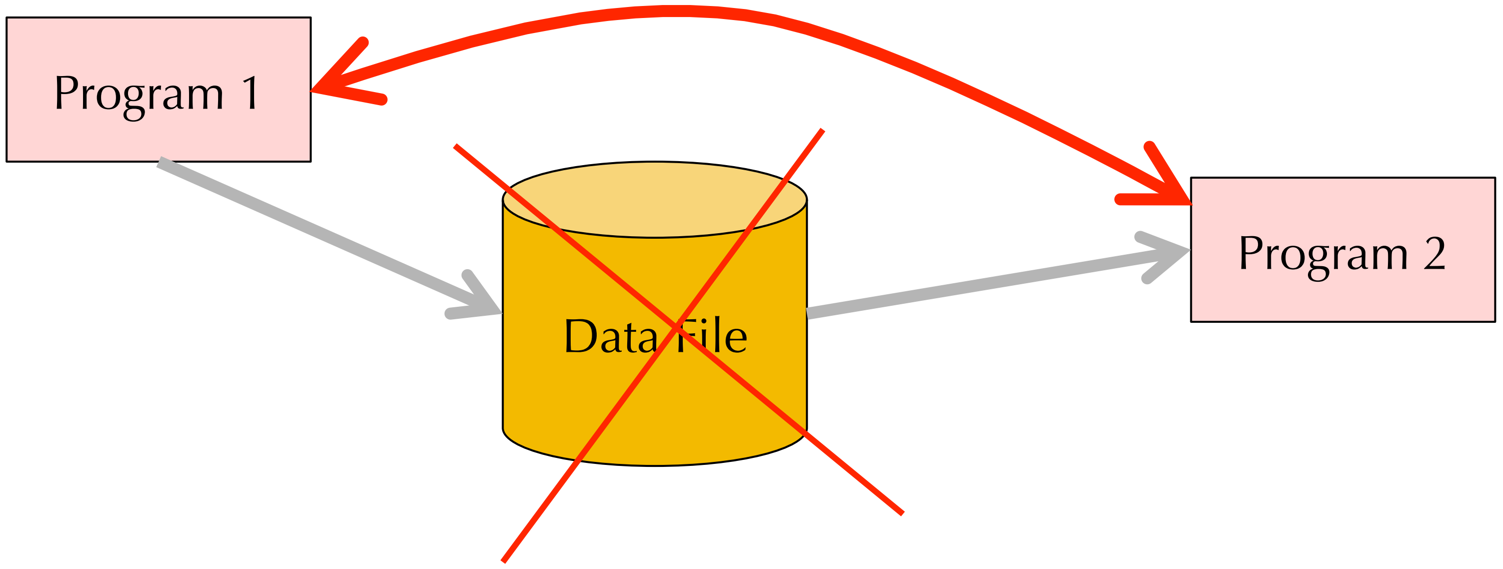
- Reading and writing text files (or XML files) is not efficient.
- Binary files are very efficient. Sometimes they are just memory dumps.
- Hard drive access time is 100,000 times slower that RAM and data transfer is 1000 times slower that RAM
- However:
  - In some Unix systems (Ex. Solaris) the folder /tmp is a RAM disk as long as enough RAM is available.
  - Solid State HDs are made of RAM memory

4

IPC: Inter-process communications

IPC - introduction

- IPC is a mechanism to exchange data directly between processes.
- Avoids temporary files
- Requires specific programming



IPC – Different types

- Inside the same computer
  - Multitasking computer
  - Ex. a program for data acquisition and another for displaying

- Machine-to-Machine communications
  - A network connection is required
  - Concerns about security
  - Ex. Web services, web-based applications, client-server model

IPC techniques (single computer)

- Pipes
  - Unix stdin/stdout redirection
  - popen function

- Shared memory
  - Usually requires using semaphores to control concurrency
  - Several processes can share data simultaneously
  - Message queues is a specific type of SHM

- Signals
  - Just for notification, no data can be sent with signals

IPC techniques (machine-to-machine)

- Sockets (Berkeley sockets)
  - Originally implemented in BSD Unix
  - Used by most (all) TCP applications

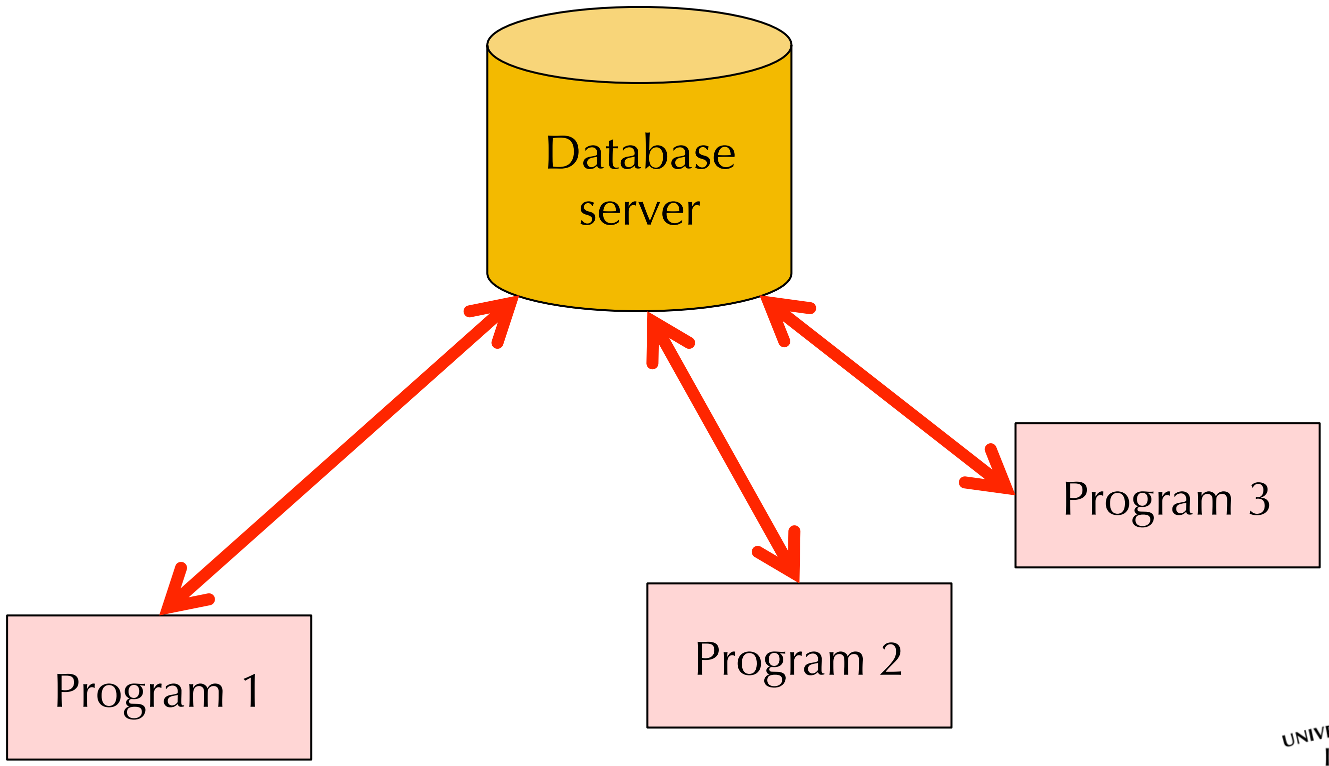
- Message passing
  - Used by MPI (parallel computing)
  - Java RMI (Java Remote Method Invocation)

5

Database connection

Database connection

- A database may be used to exchange information in a similar way as the file exchange approach
- Several programs can access the data at the same time



Databases: Matlab example

```
%This example requiere Matlab's Database toolbox

conn=database('MyConn', '', '');  %'MyConn' is the ODBC name (DSN name)
if ~isempty(conn.Message)
   error(conn.Message);
end
query=['select * from usuarios'];

cur=exec(conn,query);
if (~isempty(cur.Message))
    fprintf(['Error menssage: 
   ',cur.Message,'\nQuery :',cur.SQLQuery,'\n']);
    error('ConsultaTable');
end
curs=fetch(cur);

dat=curs.Data;

close(conn);
```

Databases: PHP example

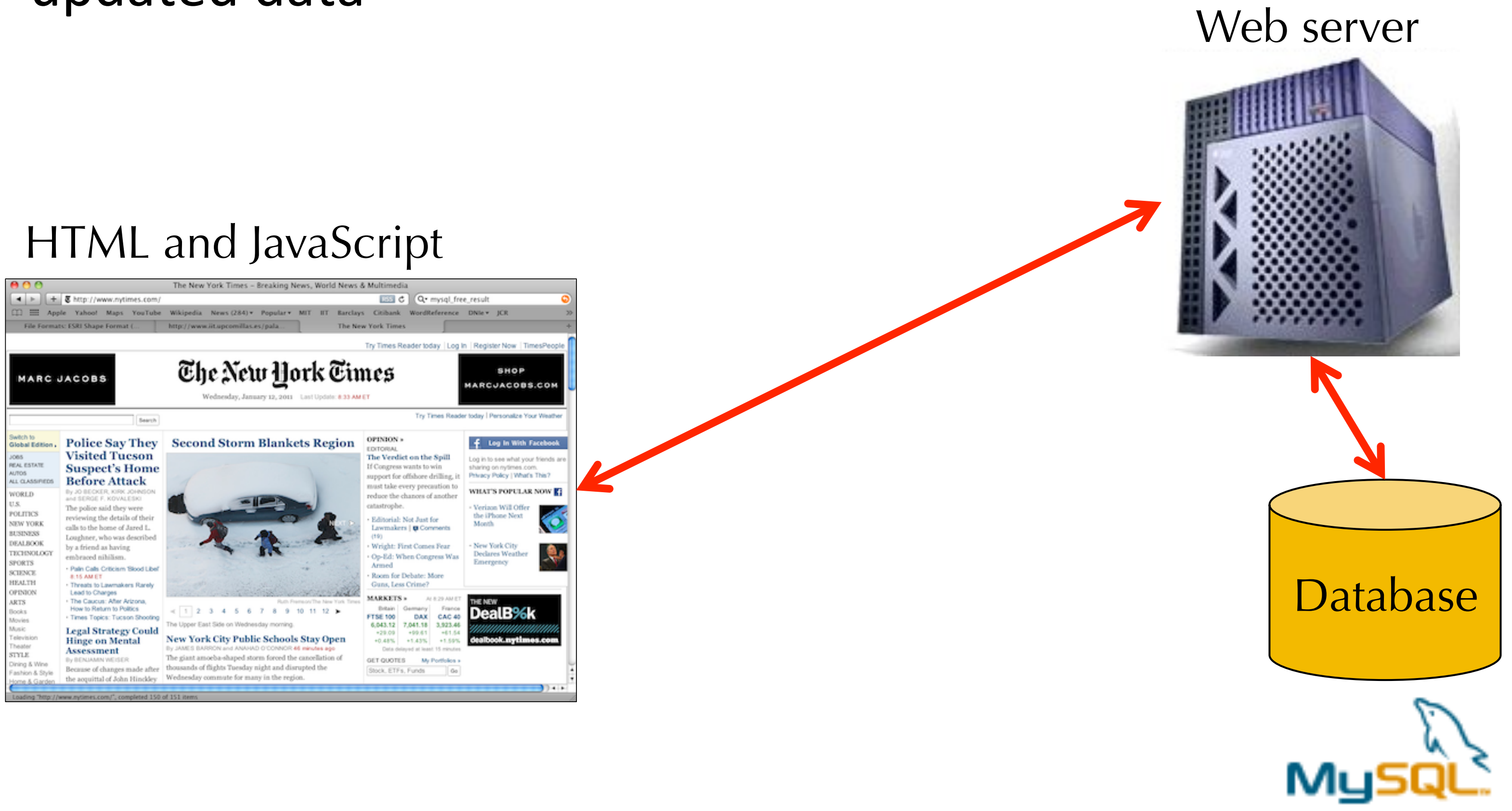
```
$conn=mysql_connect(localhost, "$username", "$password");
mysql_select_db("personel",$conn);

$query = "SELECT * from users";
$result = mysql_query($query,$conn);
if ($row=mysql_fetch_array($result))  {
   while ($row)  {
       print "Name: {$row['name']} {$row['lastname']} <br>\n";
       $row=mysql_fetch_row($result);
   }
}
mysql_free_result($result);
mysql_close($conn);
```

6

AJAX approach

AJAX

- JavaScript applications communicate with web server to get updated data



AJAX example 2

```
//HTML objects

<select name="areas" onchange="MyAJAX(this.value);">
   <option>--Select Area--</option>
   <option value="13">ASI</option>
   <option value="11">ADI</option>
   <option value="14">GEA</option>
   <option value="15">ASF</option>
   <option value="1">REDES</option>
   <option value="2">MAC</option>
   <option value="3">SADSE</option>
   <option value="4">RYE</option>
</select>

<br>

<div id="list">
</div>
```

JavaScript function to be called

Empty container

AJAX example 2

```
//JavaScript function
function MyAJAX(value)
{
   var destination=document.getElementById("list");
   var doc = null;

   doc = new XMLHttpRequest();
   if (doc){
      var my_url="http://www.iit.upcomillas.es/palacios/act/consulta_area.php?area="+value;
      doc.open("GET", my_url, false);
      doc.send(null);

      //store result in list object
      destination.innerHTML = doc.responseText;
   }else{
      destination.innerHTML = 'Browser unable to create XMLHttp Object';
   }
}
```

Instituto de Investigación Tecnológica
C/ Santa Cruz de Marcenado, nº 26
28015 Madrid
Tel +34 91 542 28 00
Fax + 34 91 542 31 76
info@iit.upcomillas.es

www.iit.upcomillas.es