

# CS290i - Lecture 2

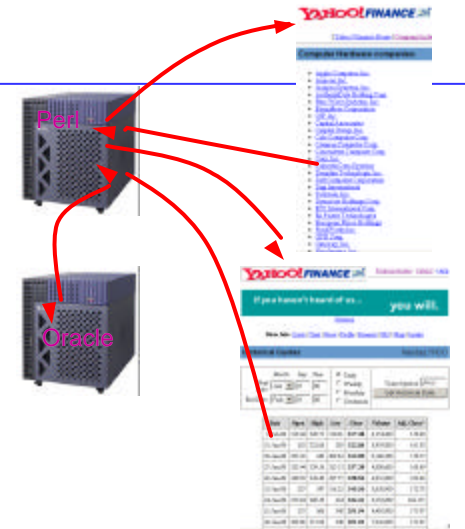
## Lift-off...

Scalable Internet Services and Systems, Winter 2002

Thorsten von Eicken  
Department of Computer Science  
University of California at Santa Barbara

## Outline

- **Crawling web sites**
  - Mechanics
  - Robot rules
- **Perl**
  - Basics
  - Tips & tricks
- **Yahoo Auctions**
  - Auction data
  - Table structure
- **SQL**
  - Basics



## Robots & Crawlers

### ■ Definition attempt:

- Crawler: automated program that fetches and processes (a significant portion) of a web site's document hierarchy
- Robot: automated program that fetches and processes specific pages of a web site

### ■ Uses:

- Search engines
- Comparison shopping
- Performance measurements
- Automated testing
- Server monitoring
- Others?

### ■ Project 1:

- Fetch auctions from Yahoo

## Manual fetch: telnet

```
■ [bugatti ~] telnet www.yahoo.com 80
Trying 204.71.200.67...
Connected to www.yahoo.akadns.net.
Escape character is '^]'.
GET / http/1.0
HTTP/1.0 200 OK
Content-Length: 18865
Content-Type: text/html
<html><head><title>Yahoo!</title><base
href=http://www.yahoo.com/><meta http-equiv="PICS-Label"
content="(PICS-1.1 "http://www.rsac.org/ratingsv01.html" 1
gen true for "http://www.yahoo.com" r (n 0 s 0 v 0 1
0))"></head><body onLoad="document.f.p.focus();"
<script><!--
...
<a href=r/ao>Advertising</a><p>Copyright &copy; 2001 Yahoo!
Inc. All rights reserved.</small><br><a href=r/pv>Privacy
Policy</a></form></center></body></html>
Connection closed by foreign host.
[bugatti ~]
```

## Manual fetch: wget

```
[bugatti ~] wget -O- http://www.yahoo.com/
--20:14:46-- http://www.yahoo.com:80/
=> '-'
Connecting to www.yahoo.com:80... connected!
HTTP request sent, awaiting response... 200 OK
Length: 16,897 [text/html]

OK -><html><head><title>Yahoo!</title><base
href=http://www.yahoo.com/><meta http-equiv="PICS-Label"
content='(PICS-1.1 "http://www.rsac.org/ratingsv01.html" 1
gen true for "http://www.yahoo.com" r (n 0 s 0 v 0 l
0))'></head><body onLoad="document.f.p.focus();">
<script><!--
...
<a href=r/hr>Jobs</a> -
<a href=r/ao>Advertising</a><p>Copyright &copy; 2001 Yahoo!
Inc. All rights reserved.</small><br><a href=r/pv>Privacy
Policy</a></form></center></body></html>
..... [100%]

20:14:46 (211.55 KB/s) - '-' saved [16897/16897]
[bugatti ~] wget -O- http://www.yahoo.com/ |& less
```

## Robot rules

■ See <<http://www.robotstxt.org/wc/guidelines.html>>

■ “Guidelines for Robot Writers”, M. Koster, 1993

- 1. Reconsider (Are you sure you really need a robot?)
- 2. Be Accountable
  - ◆ Identify your Web Wanderer (HTTP User-agent field)
  - ◆ Identify yourself (HTTP From field with your email address)
  - ◆ Announce it to the target (tell the webmaster)
  - ◆ Be informative (use the HTTP referer field)
  - ◆ Be there
  - ◆ Notify your authorities (sysadmin, network admin)
- 3. Test Locally
- 4. Walk, don't run

## Robot rules (cont.)

- 5. Use If-modified-since or HEAD where possible
- 6. Ask for what you want (HTTP Accept field)
- 7. Ask only for what you want (check suffixes of links)
- 8. Check the results
- 9. Don't Loop or Repeat (keep list of visited URLs, handle DNS aliases)
- 10. Run at opportune times (e.g. their night)
- 11. Don't run it often
- 12. Don't try queries
- 13. Log
- 14. Be interactive (start/pause/checkpoint/stop robot)

## Robots.txt

■ A Standard for Robot Exclusion

- Text file, URI: /robots.txt
- Lines of the form:  
<field>:<optionalspace><value><optionalspace>
- Comments start with #
- Disallow record: URI prefix that should not be fetched, empty=>allow all
- User-Agent field: following Disallow records apply to specific robot
  - ◆ Example:

```
# robots.txt for http://www.example.com/
User-agent: *
Disallow: /cyberworld/map/
User-agent: cybermapper
Disallow:
Allows all to cybermapper, and all but /cyberworld/map to others
```

## Robot/crawler limitations

- Unlinked pages (plain text links)
- Authentication, cookies
- Parameters/dynamic pages
- Browser/host type
- Document types
- Others?
  - ◆ Q: how about your favorite search engine?

9

## Robot architecture

- What does a low-end robot need to do?
- How about a high-end robot?

10

## Perl

- Interpreted high-level programming language
- Developed by Larry Wall
- According to Larry, he included in Perl all the cool features found in other languages and left out those features that weren't so cool

### ■ The name:

- Practical Extraction and Report Language
- Pathologically Eclectic Rubbish Lister

### ■ On bugatti.cs.ucsb.edu

- Use `/usr/local/bin/perl` !!!  
(or else you won't have the modules you need)

11

## Perl motto

- The Perl motto is ``There's more than one way to do it.''  
Divining how many more is left as an exercise...
- Perl's learning curve is therefore shallow (easy to learn) and long (there's a whole lot you can do if you really want)
- The first of the four Perl Paradoxes:  
Perl programs are easy to write but not always easy to read
- For example, the following lines are equivalent!
  - ◆ `if ($x == 0) {$y = 10;} else {$y = 20;}`
  - ◆ `$y = $x==0 ? 10 : 20;`
  - ◆ `$y = 20; $y = 10 if $x==0;`
  - ◆ `unless ($x == 0) {$y=20} else {$y=10}`
  - ◆ `if ($x) {$y=20} else {$y=10}`
  - ◆ `$y = (10,20)[$x != 0];`

12

## Perl documentation

### ■ Man

- Man perl => list of man pages
- Perldoc => same as man, without other stuff

### ■ Web pages

- www.perl.com (O'Reilly)
- www.cpan.org (Comprehensive Perl Archive Network)

### ■ Modules

- HTTP::Request
- HTTP::Response
- LWP::UserAgent
- And more...

13

## Example: extract HREFs

### ■ Extract all HREFs out of www.yahoo.com

```
[bugatti ~] cat test.pl
#!/usr/local/bin/perl -w
use strict;
require LWP::UserAgent;
require HTTP::Request;

my $ua = LWP::UserAgent->new;
my $request = HTTP::Request->new(GET =>
'http://www.yahoo.com');
my $response = $ua->request($request);
print "Response code: ", $response->code, "\n";
my @response = split /\r?\n/ , $response->as_string;
for(my $i=0; $i<5; $i++) {
    print $response[$i], "\n";
}
print "----\n";
my @hrefs = $response->as_string =~ /href="?([">]+)/g;
foreach my $h (@hrefs) {
    print "$h\n";
}
[bugatti ~]
```

14

## Extract HREFs (cont.)

```
[bugatti ~] ./test.pl |& head -20
Response code: 200
HTTP/1.0 200 OK
Content-Base: http://www.yahoo.com/
Content-Length: 16897
Content-Type: text/html
Client-Date: Thu, 05 Apr 2001 06:33:05 GMT
----
http://www.yahoo.com/
r/al
r/pl
r/ml
r/wn
r/il
r/hw
/homet/?http://taxes.yahoo.com
http://messenger.yahoo.com
/homet/?http://baseball.fantasysports.yahoo.com/baseball/
r/so
/homet/?http://travel.yahoo.com
r/a2
[bugatti ~]
```

15

## Perl Modules

### ■ Find them on CPAN

- Installation:
  - ♦ wget www.cpan.org/.../foo.tar.gz
  - ♦ gunzip <foo.tar.gz | tar xfv -; cd foo
  - ♦ perl Makefile.PL PREFIX=/home/tve/perl
  - ♦ make;make test; make install
  - ♦ use lib "/home/tve/perl/lib/site\_perl";
- Database module
  - ♦ use DBI;
- Request object manipulation, perform remote access
  - ♦ use HTTP::Request;
  - ♦ se HTTP::Request::Common;
  - ♦ use LWP::UserAgent;
- URI manipulation and HTML conversions
  - ♦ use URI::URL;
  - ♦ use HTML::Entities;

16

## What is Perl good for?

- Good applications/settings for Perl (or many other interpreted languages)
- Poor applications/settings for Perl
- How is Perl "interpreted"?

17

## Yahoo! Auctions

- Project 1
  - Fetch list of "game" auctions
  - Fetch information about each item
  - Store data about items:
    - ◆ About the item
    - ◆ About the seller
    - ◆ About the bids
- Project 2
  - Display the items you have stored



18

## SQL information

- SQL: Structured Query Language
  - pronounced "ess-cue-el" or "sequel"
- Tutorials
  - Philip Greenspun's "SQL for Web Nerds" (<http://www.arsdigita.com/books/sql/>)
  - Mike Chapple's Introduction to SQL (<http://databases.about.com/compute/databases/library/weekly/aa020401a.htm>)
  - James Hoffman's Introduction to SQL (<http://www.geocities.com/SiliconValley/Vista/2207/sql1.html>)
- Oracle: ready to take over the world
  - [technet.oracle.com](http://technet.oracle.com) - requires free registration
  - See links on cs290i web page

19

## SQL example

- [bugatti ~] sqlplus tve  
SQL\*Plus: Release 8.1.7.0.0 - Production on Sat Jan 12 18:20:14 2002  
(c) Copyright 2000 Oracle Corporation. All rights reserved.  
Enter password:  
  
Connected to:  
Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production  
JServer Release 8.1.7.0.0 - Production
- SQL> create table bid (price decimal(8,2), userid varchar(16));  
Table created.
- SQL> desc bid  
Name Null? Type  
-----  
PRICE NUMBER(8,2)  
USERID VARCHAR2(16)

20

## SQL example (cont.)

- SQL> insert into bid (userid, price) values ('tve', 100.00);  
1 row created.
- SQL> insert into bid (userid, price) values ('josep', 105.00);  
1 row created.
- SQL> select \* from bid;  
PRICE USERID  
-----  
100 tve  
105 josep
- SQL> select \* from bid where userid = 'tve';  
PRICE USERID  
-----  
100 tve

21

## SQL example (cont.)

- SQL> select max(price) from bid;  
  
MAX(PRICE)  
-----  
105
- SQL> drop table bid;  
  
Table dropped.
- SQL> exit  
Disconnected from Oracle8i Enterprise Edition  
Release 8.1.7.0.0 - Production  
JServer Release 8.1.7.0.0 - Production

22

## Perl DBI example

- Connect to the database  
my \$dsn = "DBI:Oracle:cs290i";  
my \$dbh = DBI->connect(\$dsn, \$user, \$password)  
or return undef; # Failure while connecting to db
- Fill the books table  
my \$command = "INSERT INTO bid (price, userid) VALUES (?,?)";  
\$dbh->do(\$command, undef, 100, "tve") or die \$dbh->errstr;
- Disconnect from the oracle server  
\$self->{dbh}->disconnect if defined \$dbh;
- Test:  
[bugatti ~] perl -e 'use DBI; my \$dsn = "DBI:Oracle:cs290i";  
my \$dbh = DBI->connect(\$dsn, "tve", "\*\*\*\*"); my \$command =  
"INSERT INTO bid (price, userid) VALUES (?,?)"; \$dbh->  
>do(\$command, undef, 100, "tve") or die \$dbh->errstr;'  
[bugatti ~] sqlplus tve  
SQL> select \* from bid;  
PRICE USERID  
-----  
100 tve  
SQL>

23

## DBI prepare

- Scheme: prepare, execute, fetch, fetch, ..., execute, fetch, fetch, ...
- my \$sth = \$dbh->prepare(q{  
INSERT INTO sales (product\_code, qty, price) VALUES (?, ?, ?)  
}) or die \$dbh->errstr;
- while (<>) {  
chomp;  
my (\$product\_code, \$qty, \$price) = split /, /;  
\$sth->execute(\$product\_code, \$qty, \$price) or die \$dbh->errstr;  
}
- \$dbh->commit or die \$dbh->errstr;
- Notes
  - Use of q{...} to avoid clashes with quotes in SQL statements
  - Use qq{...} if you want to interpolate vars (see perl op man page)

24

## **FYI: Speed of fiber**

- Does speed of light matter? And how fast is light in fiber?
- Refractive index = speed of light in space / speed of light in medium
- Refractive index of glass in fiber = 1.5 (approx)
- Light in fiber = 2/3 Speed of light in space
- Europe->US (LON-NYC) = 3500mi = 5600km
- Time:  $5600 * 1.5 / 300000 = 28\text{ms}$  -> 56ms r/t
- Measurements: LON~20ms, NYC~100ms