YAWS

Yet Another Web Server

Bartosz Rakoczy
Mateusz Ziebura

YAWS

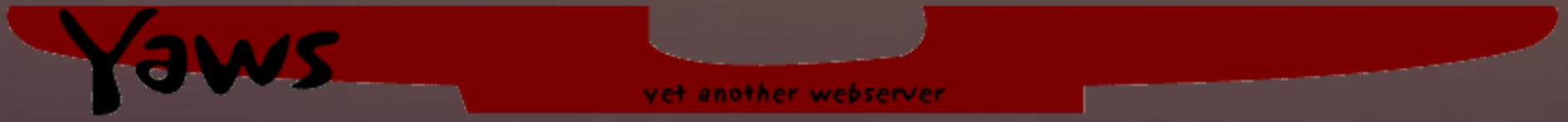
Yet Another Web Server

Bartosz Rakoczy
Mateusz Ziebura

Czym jest YAWS?

"Yaws is a com fectiou curs mainly in warm humid regi Africa, Asia, South and Central A Islands..."

Czym naprawdę jest YAWS?



"Yaws is a HTTP high perfomance 1.1 webserver particularly well suited for dynamic-content web applications."

"The main advantages of Yaws compared to other Web technologies are performance and elegance."

Licencja

BSD license

"BSD licenses are a family of permissive free software licenses, imposing minimal restrictions on the redistribution of covered software."

https://github.com/klacke/yaws

Historia

Wersja 1.0 została opublikowana przez Claes'a "Klacke" Wikstrom'a w 2003 roku.

Contributors:

Tomas Abrahamsson, Manuel DurÃĄn Aguete, Fabian Alenius, Jason Andersson, Per Andersson, Joe Armstrong, Michael Arnoldus, Oleg Avdeev, Tuncer Ayaz, Stu Bailey, Kuzma Bartosz, Eric Baur, Johan Bevemyr, Martin BjÃśrklund, Dominique Boucher, Adam Bozanich, Richard Bucker, Yinso Chen, Gaspar Chilingarov, Mats Cronqvist Anders Dahlin, Robert David, Willem de Jong, FranÃ§ois de Metz, doccarcass, dnz@bk.ru, Colm Dougan, Hans-Christian Esperer, Daniel Fabian, Christopher Faulet John Fessenden, Michael Fig, Bruce Fitzsimmons, Magnus

Tomas Abrahamsson, Manuel DurÃĄn Aguete, Fabian Alenius, Jason Andersson, Per Andersson, Joe Armstrong, Michael Arnoldus, Oleg Avdeev, Tuncer Ayaz, Stu Bailey, Kuzma Bartosz, Eric Baur, Johan Bevemyr, Martin BjÃśrklund, Dominique Boucher, Adam Bozanich, Richard Bucker, Yinso Chen, Gaspar Chilingarov, Mats Cronqvist Anders Dahlin, Robert David, Willem de Jong, FranÃ§ois de Metz, doccarcass, dnz@bk.ru, Colm Dougan, Hans-Christian Esperer, Daniel Fabian, Christopher Faulet John Fessenden, Michael Fig, Bruce Fitzsimmons, Magnus FrÃśberg, Olivier Girondel, Sergei Golovan, Luke Gorrie, Igor Goryachev, Joakim GrebenÃś, Paul Hampson, Per Hedeland, Christian Hennig, Sean Hinde, hun@n-dimensional.de, Wes James, jcortner@cvol.net, joe_e_e, Phanikar.K, Dimitriy Kargapolov, Mikael Karlsson, Bengt Kleberg, Petter Larsson, James Lee, Ahti Legonkov, Peter Lemenkov, Michael Leonhard, Eric Liang, Fredrik Linder, Fabian Linzberger, Daniel Luna, Paul Mahon, Davide MarquÃŞs, Brady McCary, Tom McNulty, Chandru Mullaparthi, Hans Ulrich Niedermann, Chris Newcombe, Julian Noble, Joseph Wayne Norton, Anders Nygren, Thomas O'Dowd, Jimmy Olgeni, Lennart Ostman, Karel Ostrovsky, Erik Pearson, Jean-SÃŠbastien PÃŠdron, Yurii Rashkovskii, Praveen Ray, Matthew Reilly, Mickael Remond, Bruno Rijsman, Bill Robertsson, Jouni Ryno, Yariv Sadan, Kostis Sagonas, Rob Schmersel, Carsten Schultz, Yariv Sedan, Vance Shipley, Alexander Simonov Michael Slaski, Leon Smith, Hal Snyder, Andrei Soroker Matt Stancliff, Sebastian Stroll, Taavi Talvik, Brian Templeton, Nicolas Thauvin, Fredrik Thulin, TorbjÃśrn TÃśrnqvist, Tjeerd van der Laan, Pablo Vieytes, Steve Vinoski, Lev Walkin, wde, John Webb, David Welton, Claes WikstrÃśm, Dan Willemsen, Haobu Yu, Liu Yubao, Tomas Selander, Ulf Wiger, Garret Smith, Nicolas Adiba, Kalle Zetterlund, Tjeerd van der Laan, Kaloyan Dimitrov, Karolis Petrauskas

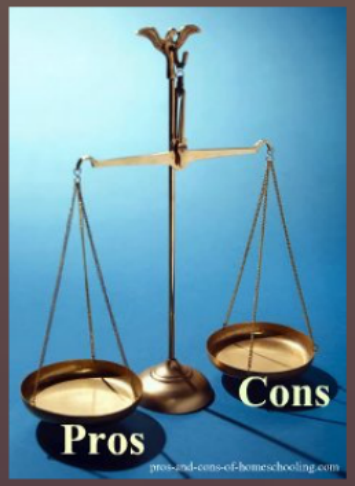
Najważniejsze cechy

Supports:
- HTTP 1.0 and HTTP 1.1
- Static content page delivery
- Dynamic content generation using embedded ERLANG code in the HTML pages
- Common Log Format traffic logs
- Virtual hosting with several servers on the same IP address
- Multiple servers on multiple IP addresses
- HTTP tracing for debugging
- An interactive interpreter environment in the Web server for use while developing and debugging a web site

Zalety i wady

pros:

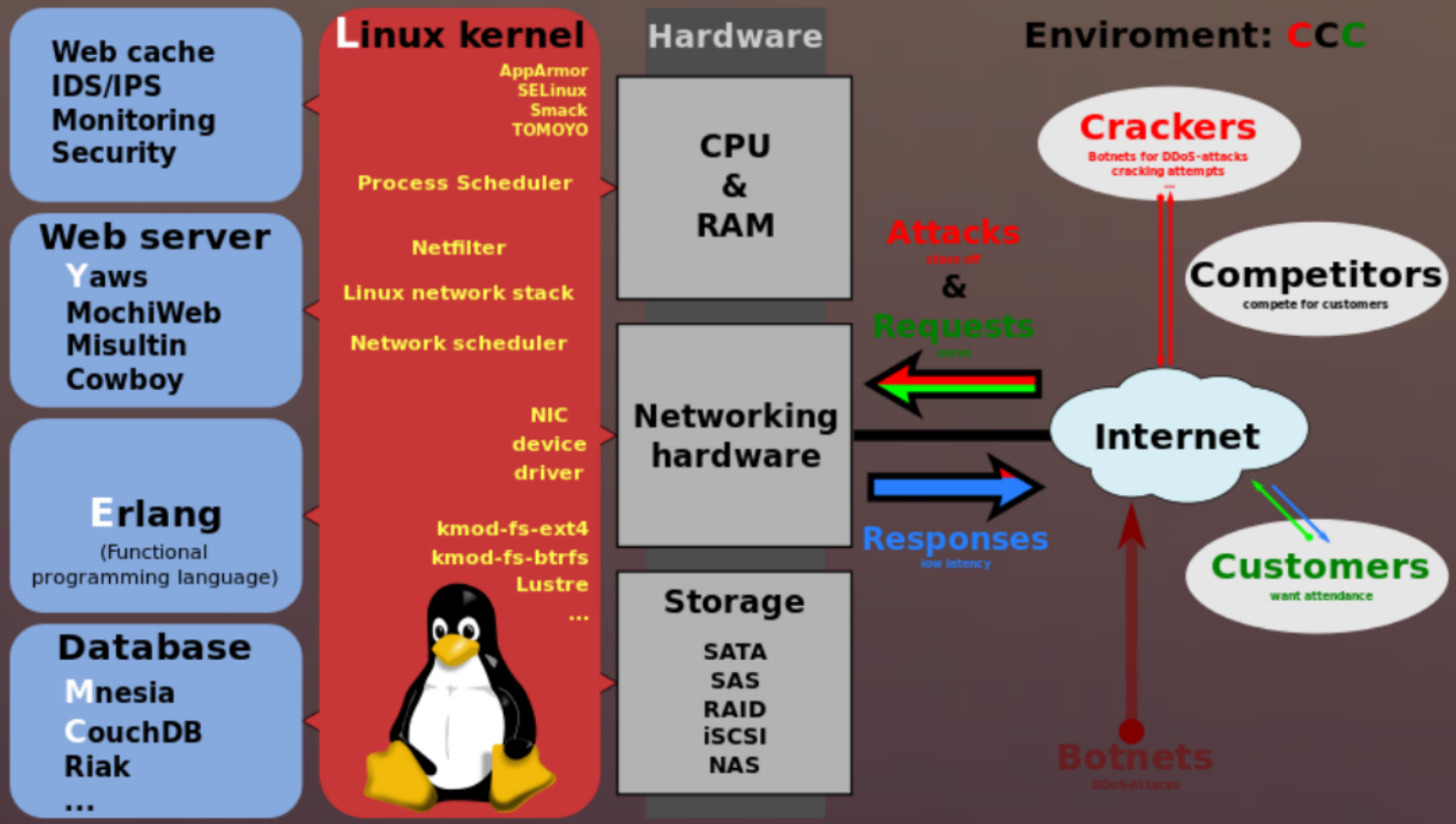
Reliability,
stability,
scalability,
speed,
dynamic content,
beauty - for some.



cons:

Erlang - for some,
popularity,
manuals.

LYME


Web cache
IDS/IPS
Monitoring
Security
Web server
Yaws
MochiWeb
Misultin
Cowboy
Erlang
(Functional programming language)
Database
Mnesia
CouchDB
Riak
...
Linux kernel
AppArmor
SELinux
Smack
TOMOYO
Process Scheduler
Netfilter
Linux network stack
Network scheduler
NIC device driver
kmod-fs-ext4
kmod-fs-btrfs
Lustre
...
Hardware
CPU & RAM
Networking hardware
Storage
SATA
SAS
RAID
iSCSI
NAS
Enviroment: CCC
Crackers
Botnets for DDoS-attacks
cracking attempts
Competitors
compete for customers
Attacks
&
Requests
Internet
Responses
low latency
Customers
want attendance
Botnets
DDoS-Attacks

Uruchamianie

Na początku konfigurujemy nasze ustawienia:

/etc/yaws/yaws.conf

```
ebin_dir =      #Gdzie są nasze pliki .beam
logdir =        #Gdzie są zapisywane logi
include_dir =   #Gdzie są pliki nagłówkowe .hrl
subconfigdir =  #Gdzie są nasze pliki konfiguracyjne
```

Plik localhost.conf w folderze subconfigdir:

```
<server localhost>
    port = 8080
    listen = 0.0.0.0
    docroot = #Gdzie są nasze pliki .yaws i .html
</server>
```

Dynamic content

Możemy dynamicznie zmieniać zawartość naszej strony używając zamiast plików .html pliki .yaws:

```
<html>
<h1>Pierwsza strona w yaws</h1>

<erl>
out(A) ->
    {html, "<p>Hello, yaws!</p>"}.
</erl>

<erl>
out(A) ->
    {ehtml,
        [ {p, [ ], "Hello, yaws!"}]}.
</erl

</html>
```

Example 1

Back to index

HEADERS:

"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.48 Safari/537.36"

GET:

[{"name","Mateusz"},{"points","6"}]

POST:

[]

REQUEST:

- method: GET
- path: {abs_path,"/example1.yaws/?name=Mateusz&points=6"}
- version: {1,1}

Moduły Erlanga

Nasz pfactorial:

```
<html>
<erl>
out(A) ->
   {ehtml, [ {ul, [], listOfFactorials(pfactorial:start(100, 4)) } ] };


listOfFactorials([]) -> [];
listOfFactorials([H | T]) ->
      [{li, [], io_lib:format("~p", [H])} | listOfFactorials(T)].
</erl>
</html>
```

Dostęp do POST, GET

POST:

```
yaws_api:pars_post(A)
```

GET:

```
yaws_api:parse_query(A)
```

W wyniku dostajemy listę krotek:

```
[ {"name", "Mateusz"}, {"points", "6"} ]
```

Praktyczne użycie POSTa

```
<html>

<h1> Example 4 </h1>
<a href="index.yaws">Back to index</a>

<form action="example4.yaws" method="post">
 <p>Number:</p>
 <input name="number" type="text">
 <input type="submit">
</form>
```

```
</form>
```

yaws_

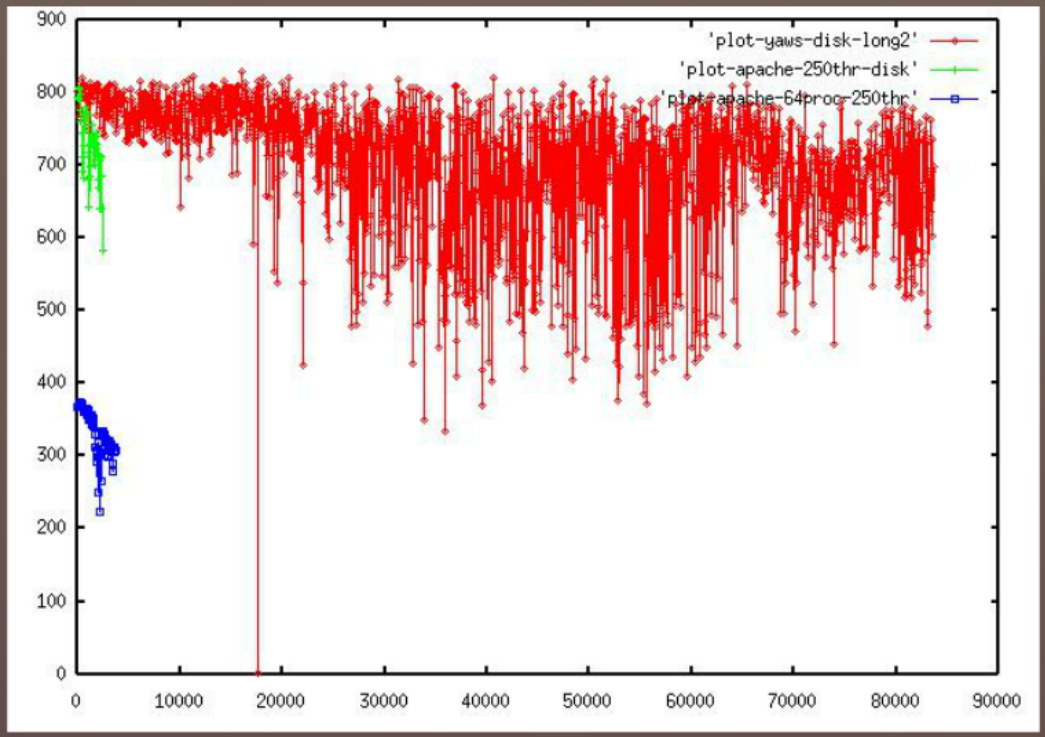
```
<erl>
out(A) ->
 case yaws_api:parse_post(A) of
  [{"number", Number} | _ ] ->
   {Num, _} = string:to_integer(Number),
   {ehtml, [
    {ul, [], listOfFactorials(pfactorial:start(Num, 4))}
   ]};
  _ ->
    {ehtml, [
       {p, [], "Number not posted"}
     ]}
 end.

listOfFactorials([]) -> [];
listOfFactorials([H | T]) ->
 [{li, [], io_lib:format("~p", [H])} | listOfFactorials(T)].
</erl>
```

Wygoda - czasem lepiej HTML

```
<form action="example1.yaws" method="post">
    <p>Your name:</p>
        <input name="name" type="text">
        <input type="submit">
</form>
```

```
<erl>
out(A) ->
    {ehtml,
     [{form, [{action, "example1.yaws"}, {method, "post"}],
      [{p, [], "Your name:"},
       {input, [{name, "name"}, {type, "text"}], [ ]},
       {input, [{type, "submit"}], [ ]}
      ]}
     ]
    }.
</erl>
```

Wydajność



"A load test conducted in 2002 comparing Yaws and Apache found that with the hardware tested, Apache 2.0.39 with the worker MPM failed at 4,000 concurrent connections, while Yaws continued functioning with over 80,000 concurrent connections."

Przydatne linki

Strona główna: http://hyber.org/

Dokumentacja: http://hyber.org/yaws.pdf

Yaws wiki: https://github.com/klacke/yaws/wiki

Nitrogen framework: http://nitrogenproject.com/

https://github.com/MrLynx93/Erlang

Wydajność - nasz mały test

Yaws

```
mateusz-laptop mateusz # ab -n 20000 -c 20000 http://localhost:8080/index.yaws
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 2000 requests
Completed 4000 requests
Completed 6000 requests
Completed 8000 requests
Completed 10000 requests
Completed 12000 requests
Completed 14000 requests
Completed 16000 requests
Completed 18000 requests
apr_socket_recv: Connection reset by peer (104)
Total of 19989 requests completed
```

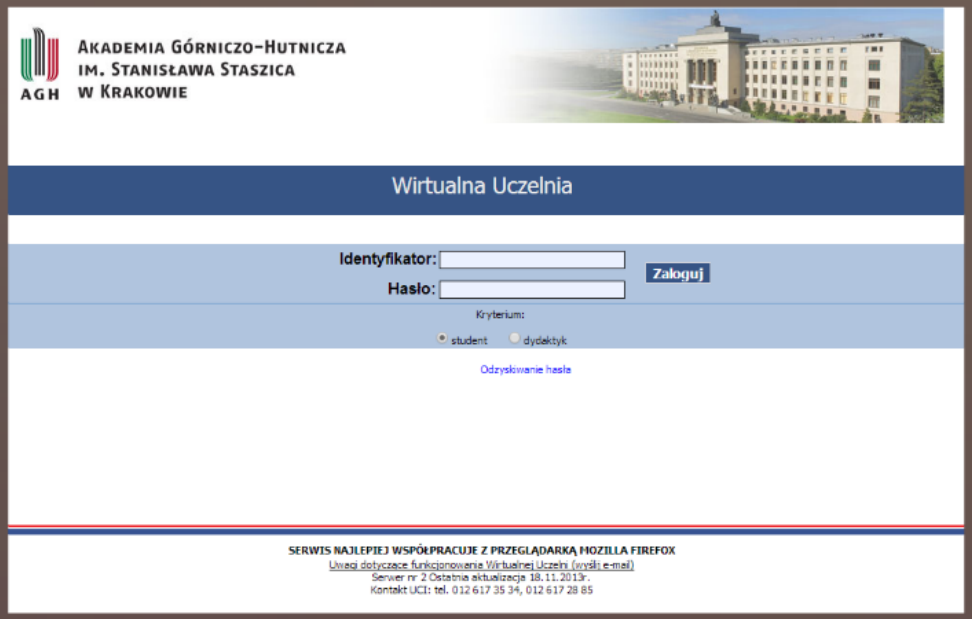
Apache

```
mateusz-laptop mateusz # ab -n 20000 -c 20000 http://localhost:8000/
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 2000 requests
Completed 4000 requests
Completed 6000 requests
Completed 8000 requests
Completed 10000 requests
Completed 12000 requests
apr_socket_recv: Connection reset by peer (104)
Total of 13796 requests completed
```

Przy 20000 requestów yaws obsłużył 19989, a Apache 13796. Wynik nie jest tak dobry, jak w testach z 2002 roku - yaws miał działać przy 80000 requestów, a Apache przy 4000.

Wniosek?

YAWS - obsługuje 20 000 requestów.



Wirtualna Uczelnia obsługuje 2...

YAWS

Yet Another Web Server

Bartosz Rakoczy
Mateusz Ziebura

Dziękujemy za uwagę

Zadowolony kot:

Źródła:

http://hyber.org/

http://wikipedia.com

https://github.com/klacke/yaws/wiki/

http://www.sics.se/~joe/apachevsyaws.html