

Embedding R within the Apache Web Server: What's the Use?

Jeffrey Horner

jeff.horner@vanderbilt.edu

<http://biostat.mc.vanderbilt.edu/twiki/bin/view/Main/ApacheRproject/>

Web App. Development with R

Interfaces to R exist:

- Network: Rserve, RSOAP
- Languages: Perl, Python, PHP, Ruby

R as Web Language

- Rweb, Rcgi, Rpad, Rho, CgiwithR, others...

Motivation for Embedding R in Apache

Allow statistician to use a familiar language

Eliminate overhead of interface language run-time

Hides cost of R startup and initialization

Useful code bases for other projects:

- Apache Portable Runtime Libraries
- Multi-Processing Modules
- Custom Protocol support

A gentle introduction to R

Hello World

```
-----http.conf-----+-----test.R-----  
|  
LoadModule R_module mod_R.so      | handler <- function(r){  
|                                   |   apache.write(r,"<h1>Hello World!</h1>")  
<Location /test/hello>           |   |   OK  
|                                   |   }  
  SetHandler r-handler           |  
  Rsource /var/www/html/test.R    |  
  RreqHandler handler             |  
</Location>                       |  
|  
-----+-----
```



Examples: Test Handler

```
handler <- function(r)
{
  # Grab all incoming HTTP request data
  args <- apache.get_args(r)
  post <- apache.get_post(r)
  cookies <- apache.get_cookies(r)
  uploads <- apache.get_uploads(r)

  # Test if the GET variable called was set
  if(is.null(args$called))
    called <- 1
  else
    called <- as.integer(args$called) + 1

  # Set a cookie with the incremented called value.
  # expires in 100 seconds.
  apache.add_cookie(r,"called",called,expires=Sys.time()+100)

  apache.write(r,"<HTML><BODY><H1>Hello from mod_R</H1>")

  # Write out the form for capturing GET,POST, and file uploads
  apache.write(r,
    '<form enctype=multipart/form-data method=POST action="/test/R?called=',
    called,'">',
    sep="")
  apache.write(r,
    'Enter a string: <input type=text name=name value="',
    post$name,'"><br>',
    sep="")
  apache.write(r,
    'Enter another string: <input type=text name=name value="',
```

```

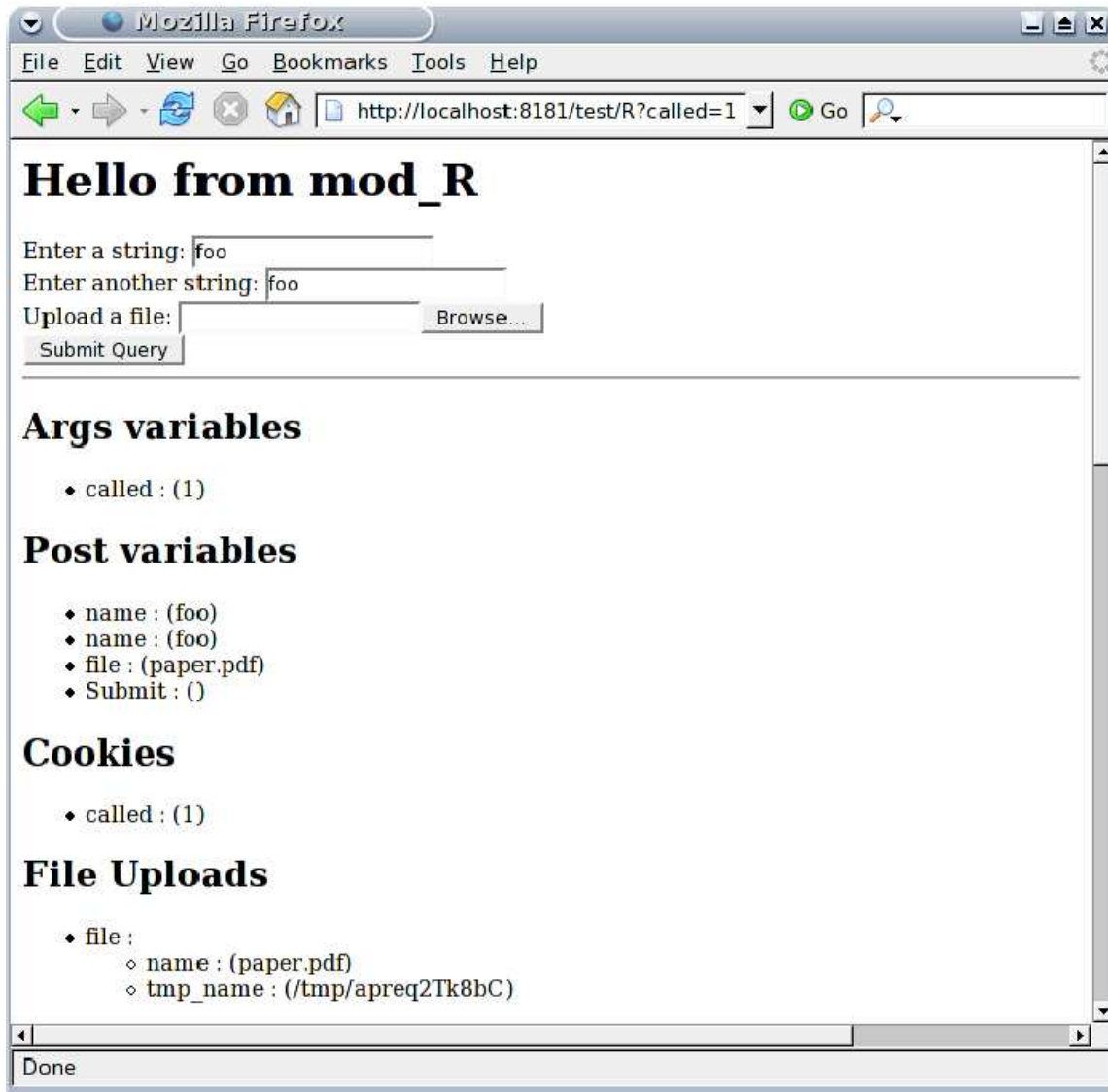
        post$name, '><br>',
        sep="")
    apache.write(r,
        'Upload a file: <input type=file name=file><br>')
    apache.write(r, "<input type=submit name=Submit>")

# Now write out everything we saw
    apache.write(r, '<hr>')
    apache.write(r, "<h2>Args variables</h2>")
    apache.write(r, as.html(args));
    apache.write(r, "<h2>Post variables</h2>")
    apache.write(r, as.html(post))
    apache.write(r, "<h2>Cookies</h2>")
    apache.write(r, as.html(cookies))
    apache.write(r, "<h2>File Uploads</h2>")
    apache.write(r, as.html(uploads))
    apache.write(r, "<h2>Request Record</h2>")
    apache.write(r, as.html(r))
    apache.write(r, "</BODY></HTML>")

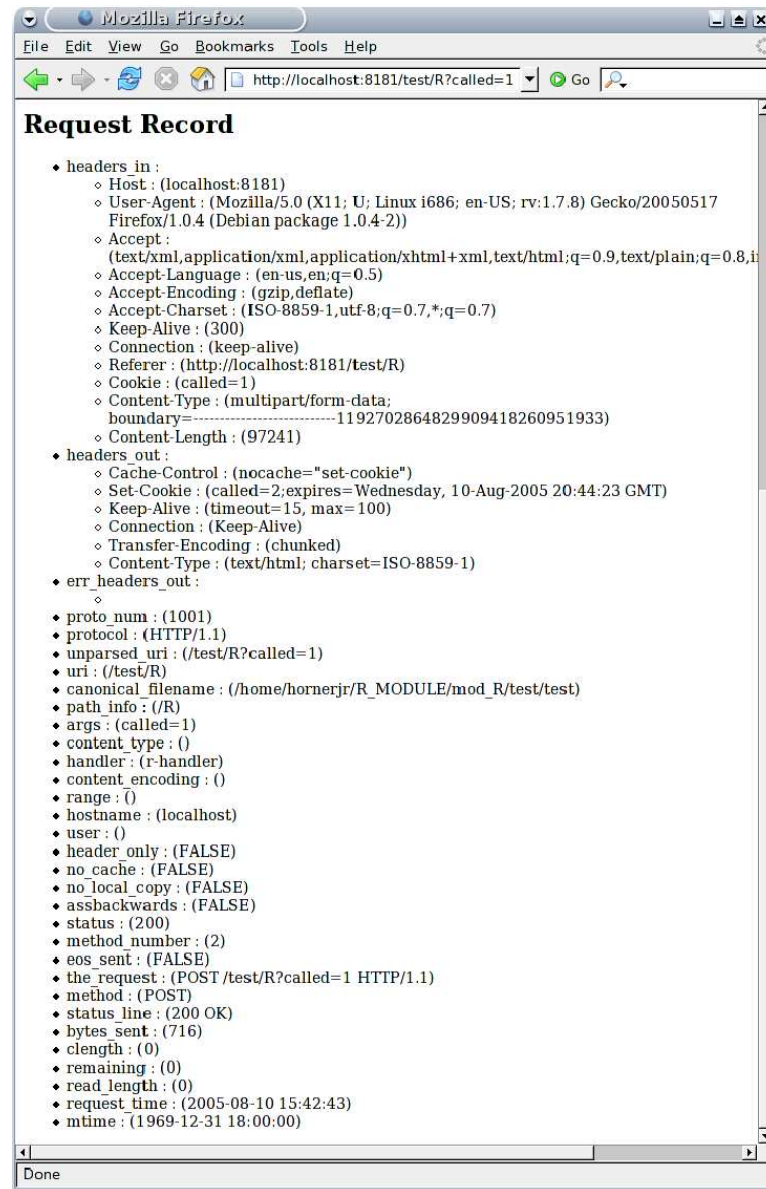
    OK
}

```

Examples: Test Handler Output 1



Examples: Test Handler Output 2



Examples: GD/NRart Handler

```
library('GDD')
library('NRart')
r2 <- function(r)
{
  step <- 2
  args <- apache.get_args(r)

  if(!is.null(args$t)){
    pstep <- as.integer(args$t)
    if (pstep > step && pstep < 121)
      step <- pstep;
  }

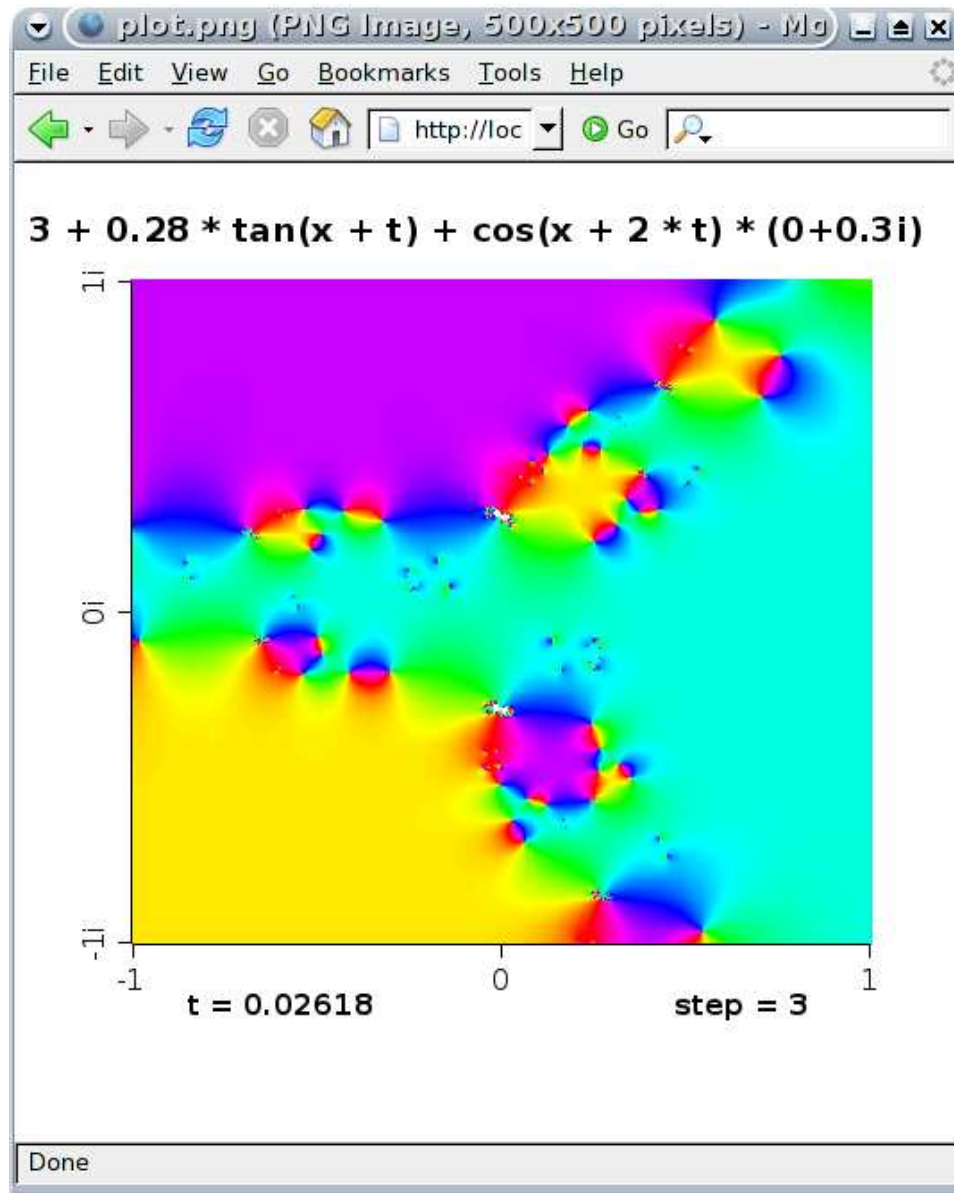
  apache.set_content_type(r,"image/png")

  GDD(ctx=apache.gdlib_ioctx(r),w=500,h=500,type="png")

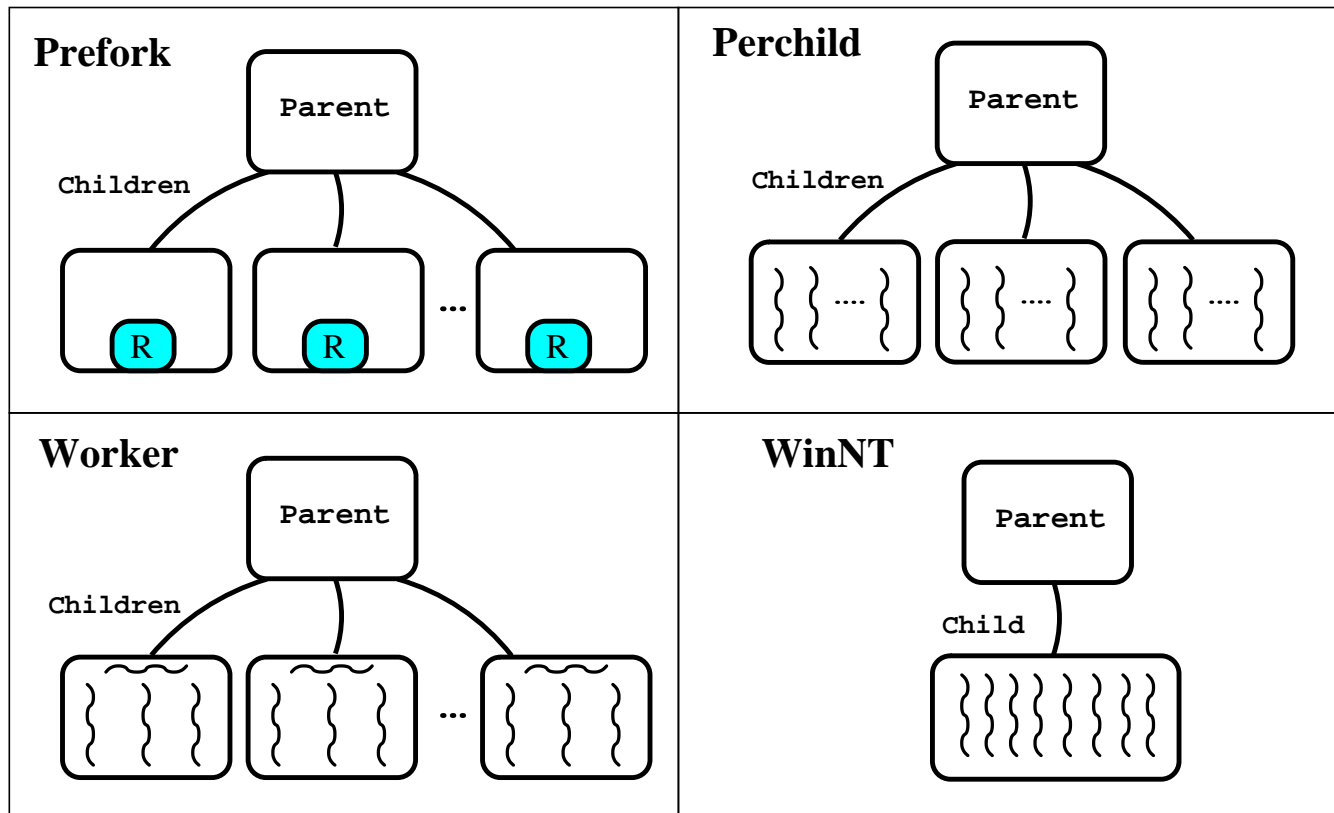
  nr.movie(x^3 + .28 * tan(x + t) + cos(x + 2*t)*.3i - 0.7,
           't', seq(0, pi, length=121)[step],
           extent=1, steps=3, points=400,
           col=rainbow(256), zlim=c(-pi, pi))

  dev.off()
  OK
}
```

Examples: GDD/NRart Handler Output.



Biggest Challenge: Maintaining State



Apache 2.0 Multi-Processing Modules.

Future Work

Find solutions to State Problem

- **memcached**
- database connections
- file connections with file locking

Custom Apache Multi-Processing module for R

- Implement sessions allowing web request to talk to same R interpreter
- Rbatch: Allow interactive R session to communicate with R slave sessions

Work with other R projects

- Rpad: email discussion already under way
- Rserve, RSOAP ?

Extensions to R

- Package level Connection interface
- **attach()** external data stores to search path