The OpenCPU Server Manual

Version 0.8-0

Jeroen Ooms

August 4, 2013

Preface

Internet access, public cloud computing, live and open data and scientific super computers are transforming the landscape of data analysis. Researchers increasingly collaborate online by sharing data, code and results. This is a powerful way to learn and teach statistical analysis and at the same time it improves transparency and accessibility of the underlying methods. Moreover, by centralizing computation we can address some scalability challenges and facilitate direct integration of analyses in systems and applications. We expect that providing open and reprodicuble materials on social platforms, in addition to a written report or article, will soon become an integral part of the scientific publication proces. The OpenCPU framework is a first attempt at building a complete system to support such applications.

The latest version of this document is available at https://github.com/jeroenooms/opencpu-manual. To report typos, feedback, bugs, comments, suggestions, etc about this manual, post them in the issues page for the repository.

Contents

1 What is OpenCPU

OpenCPU is a framework for embedded scientific computing and reproducible research, based on R, Latex and Pandoc. The OpenCPU server exposes an HTTP API to share and execute scripts, functions and reproducible documents. The system addresses many of the domain specific problems inherent to scientific computing, and abstracts away technicalities behind a well defined intuitive HTTP interface. This provides a foundation for scalable applications with embedded statistical analysis, vizualization and reporting.

1.1 Design Philosophy

In OpenCPU, the analysis/vizualization is completely seperated from other parts of the system or application. OpenCPU has been designed to run on a seperate server, interfaced only through the HTTP API. The client needs no knowledge of R or Latex; OpenCPU defines a mapping between HTTP requests and R function calls which results in a natural RPC-like protocol.

```
$ curl http://localhost/ocpu/library/stats/R/rnorm/json --data n=3
[
  3.05644,
  0.38511,
  1.11983
]
```

Our philsophy is that R should be used only to do what it is good at: analysis and graphics. The OpenCPU API exposes this functionality though a clean and robust API. Any software program that speaks HTTP call an R function, without the need to generate or parse R code. From there, it is completely up to the client on how to process or present the output. OpenCPU deliberately does not include, suggest or enforce the use of any specific web development language, GUI, etc. This is quite different from some other R/web systems, which ship with built-in templates to generate parameratized out-of-the-box widgets from R code.

When using OpenCPU, the roles of analyst and web developer are seperated. The analyst implements and documents an R function or script, just as he is used to. The web developer can use his favorite language, tools and frameworks to call this function over HTTP. There is no need for the web developer to learn R, nor does the analyst have to worry about GUIs or web related technicalities. Furthermore, we are not restricted to a limited set of available widgets or panels. OpenCPU leaves it completely up to the web developer(s) and their imagination how and which to design their application. This gives a lot of freedom (and work) in the types of applications that we can build.

1.2 OpenCPU Apps

OpenCPU defines a standard way to build and share "apps". An OpenCPU app is an R package which, in addition to the regular contents, ships with some web page(s). These pages interact with the R functions in this package through the OpenCPU API. By convention, these web pages (html/css/js/etc files) are included in the `/inst/www/` directory of the R source package. This way, OpenCPU apps provide a standardized way to ship standalone R web applications.

Because OpenCPU apps are simply R packages, they are developed, distributed and installed the same way as any other R package. Several example apps are available from the OpenCPU github organization at https://github.com/opencpu. For example to install the gitstats app, we can use the `devtools` package:

```
library(devtools)
install_github("gitstats", "opencpu")
```

The application can then be opened in a browser. For example, if the application was installed on an OpenCPU cloud server, it would be available at http://localhost/ocpu/library/gitstats/www.

To make developing OpenCPU apps easier, a simple Javascript client library is available called opencpu.js. This library depends on jQuery and uses `$.ajax` to provide javascript wrappers to the OpenCPU API. It is not mandatory to use this javascript library, but it provides a convenient basis for building OpenCPU apps.

1.3 The OpenCPU single-user server

Two versions of OpenCPU are available: a single-user server that builds runs inside an interactive R session, and a cloud server that builds on Apache and Nginx. The single-user server is intended for development and local use only. The latest version can easily be installed from the Github repository:

```
library(devtools)
install_github("opencpu", "jeroenooms")
```

When the opencpu package is loaded, the server is automatically started:

```
library(opencpu)
```

Because R is single-threaded, the single-user server does not support concurrent requests (but httpuv does a good job in queueing them). Also it does not enforce any security. The single-user server is great for developing apps, that can later be published on the OpenCPU cloud server. When using the single-user server, we can easily load the apps for local use:

```
install_github("gitstats", "opencpu")
opencpu$browse("gitstats")
```

The `opencpu$browse` function will automatically open the app in the default web browser.

2 The OpenCPU cloud server

The OpenCPU cloud server runs on Ubuntu 12.04 (precise) or higher. The OpenCPU system consists of a number of standard Ubuntu installation packages. These are:

- `opencpu` – Meta package which installs both `opencpu-server` and `opencpu-cache`.
- `opencpu-server` – The main OpenCPU API server. Depends on `R` and `apache2`.
- `opencpu-cache` – OpenCPU caching server. Depends on `nginx`.
- `opencpu-full` – Installs `opencpu` plus many `texlive` and `pandoc` packages.

2.1 Getting an Ubuntu server

OpenCPU requires Ubuntu version 12.04 or higher. Any version of Ubuntu will do, e.g. Ubuntu Desktop, Ubuntu Server, Kubuntu, Edubuntu, etc. The preferred way of running OpenCPU is on a clean Ubuntu Server edition. A copy of the Ubuntu Server installation disc ISO can be obtained from the Ubuntu download pages:

http://www.ubuntu.com/download/server

To easiest way to run Ubuntu Server on Amazon EC2 is by using one of the official AMI's as provided by the ubuntu team:

http://cloud-images.ubuntu.com/raring/current/

Another possibility is to install a OpenCPU on a virtual Ubuntu server inside another OS. For example, the free VMware Player is available for Windows and Linux, and on OSX one can use parallels to run an Ubuntu server. This way you can install Ubuntu and OpenCPU safely on top of an existing system.

2.2 Basic OpenCPU installation

Before installing OpenCPU, make sure the system is up to date:

```
sudo apt-get update
sudo apt-get upgrade
```

To install OpenCPU, first add the OpenCPU repository to the system:

```
sudo add-apt-repository ppa:opencpu/opencpu-0.8 -y
sudo apt-get update
```

We can now go ahead and install the server:

```
sudo apt-get install opencpu
```

Installation on a clean server might take a while because R and Latex both have many dependencies. After installation is done, we should be able to open a browser and point it to the `/ocpu` path at server address e.g: . If you see the welcome page, the installation has succeeded.

2.3 support for other systems

3 Cloud Server Management

4 The API