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 reproducible 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

1.3 The single-user server versus the cloud server

2 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 then be published on the OpenCPU cloud server.

2.1 Loading apps in the single-user server

OpenCPU apps are simply R packages, and can be installed as such. Several apps are available from the OpenCPU github organization at https://github.com/opencpu. For example, to install the gitstats app, the user would run:

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

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

3 Cloud Server Installation

3.1 install and uninstall

3.2 deb package structure

3.3 support for other systems

4 Cloud Server Management

5 The API