# Simone Mosciatti

*Curriculum*

*Italy*
✉ *simone@mweb.biz*
*siscia.github.io*

## Education

2013–Present **Bachelor in Engineering of Computer Systems**, *Politecnico di Milano*, Milan, Italy.

Currently, for my thesis I am working on **Estimating the time to completion of MapReduce jobs**. This work leverages MARC, a framework for the estimation of resource consumption, as a mean to provide accurate models for the completion time of MapReduce jobs executed on a Hadoop cluster. I take into consideration the percentage of job completion as a resource that can be consumed; a piece-wise linear model is then generated subdividing the job execution into different phases, according to configuration parameters and runtime metrics. The proposed model allows the user to have both a better estimate of the time required by an already submitted job to complete and the ability to perform more accurate resource reservations, towards a more efficient scheduling and a higher cluster utilization.

2014–2015 **Bachelor in Engineering of Computer Systems**, *Tongji University*, Shanghai.

Within the Politong Double Degree Program

## Experience

2013–2014 **Software Engineer, Freelance**, *workinvoice.it*, Milan.

The platform is a marketplace to buy and sell invoices.
I re-engineered the backend, written in Clojure, to improve code maintainability and system scalability. I also worked on the frontend, which uses standard HTML5, CSS and Javascript technologies, to enhance the user-experience and the interface usability.
Overall, the re-engineering process focused on:

- The authentication system for different kinds of users
- Designing and implementing the backend APIs
- Re-designing the SQL schema

2012–Present **OpenSource Contribution**.

Contribution to the following libraries:

- https://github.com/clj-time/clj-time
  The standard way to handle time in Clojure.
  • I implemented periodic sequences with upperbound.
- https://github.com/budu/lobos
  SQL schema manipulation and migration library for Clojure.
  • I made possible to use lobos with the last version of the java drivers, it was particulary needed because all the libraries that talks with databases in clojure didn't support anymore the previous version on the driver, making the whole project lobos useless.

2012–Present **OpenSource Author**.
I am the author of the following projects:

- https://github.com/siscia/effe
  Open Source Software to write serverless microservices, it expects four handler (start-up, create a context, run computation, destroy context) to be implemented by the developer and plug those handlers inside a request loop.
- https://github.com/siscia/effe-tool
  Makes it simple to work with effe creating the templates for the handlers, it helps in compiling the software and it also makes extremely small docker containers.
- Introduction to Highly Scalable, Fault Tolerant, Distributed System
  Tiny booklet about the basic ideas behind modern distributed systems and how to make those systems scalable and fault tolerant.
- https://github.com/siscia/numerino
  Priority Queue in Elixir/Erlang/OTP, it is a web server that provides two endpoints, one to push into the queue, the other to pull out of the queue. Its focus is on performace covering more than 9k req/sec on slow ARM hardware. The next logical step would be to make the software distributed in order to achieve horizontal scalability.
- https://github.com/siscia/postgres-type
  Makes possible the use the JSON datatype in Postgres database with Clojure.
- https://github.com/siscia/css-parser
  It parse css into a AST, it is fairly more complete than other css parser since it is extremely precise on how the attributes values are expressed.

## Languages

Italian **Mother tongue**

English **Proficient** *I was an AFS exchange student in 2011-2012 at Carthage, MO, USA high school.*

Chinese **Basic**

## Programming Skills

Proficient **Clojure**, *compojure, monger, korma, core.async*.
Proficient **Python**, *Flask, Django*.
Proficient **Elixir**, *Erlang/OTP, BEAM VM*.
Very Good **SQL**, *PostgreSQL, SQLite3*.
Very Good **NoSQL**, *MongoDB*.
Very Good **Java**.
Very Good **Go(lang)**.
Good **C/C++**.
Good **JavaScript**, *mongoose, node.js*.
Basic **Haskel**.
Basic **Rust**.