

# Technical Overview

## Neronet

### Project 2

Toolbox for managing the training  
neural networks (Pyy Takala)

CSE-C2610 Software Project  
Aalto University

November 2, 2015

# Outline

Overview

Components

Remarks

Technical  
Overview

Neronet

Overview

Components

Remarks

# Outline

## Overview

## Components

## Remarks

Technical  
Overview

Neronet

Overview

Components

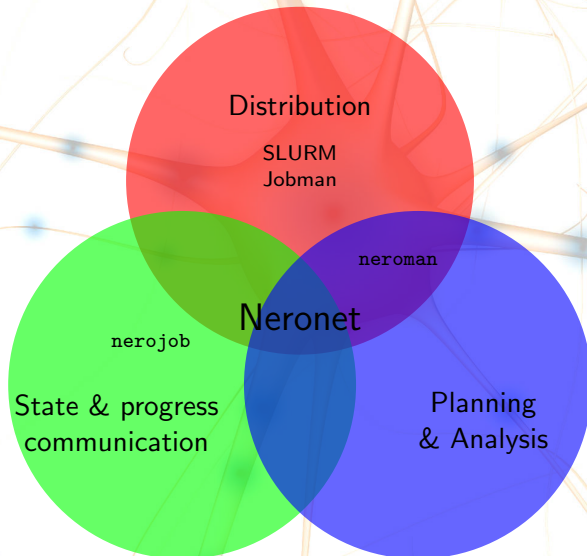
Remarks

Neronet is a framework designed to facilitate the specification, submission, monitoring, control, analysis and management of many different computational experiments. It has three components:

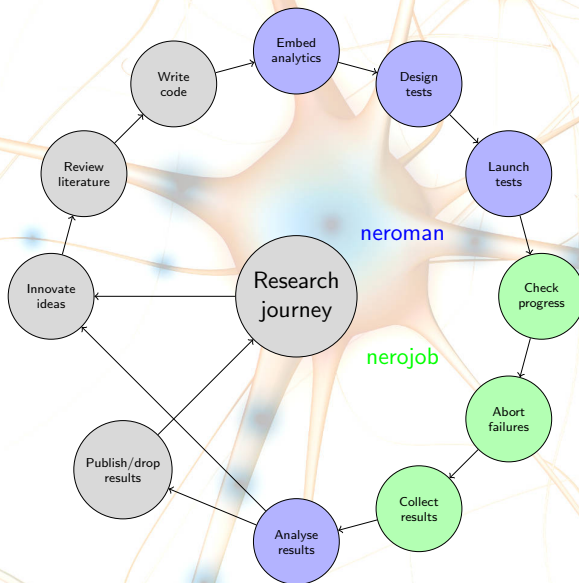
1. A job scheduler for Linux clusters (either SLURM or Jobman)
2. `nerojob` – a program designed to be launched by a job scheduler in cluster nodes to monitor, analyse and control single experiment jobs and their environment in tandem with `neroman`
3. `neroman` – a user run daemon that acts as the Neronet information server and user interface to provide Neronet's features

The following three slides provide a good introduction:

1. *Job management* – the associated problem domain spheres and how the system components are related to them
2. *Journey map* – the journey or work flow of a typical system user (a computational researcher) and the steps in which the Neronet tools are designed to be used
3. *Basic use case* – A sample basic system use case description



# Journey map



- ▶ **User:** A computational researcher
- ▶ **Goal:** To test how well a new design works with several different configuration options and parameter values
- ▶ **Preconditions:** SLURM cluster and Neronet setup, code and analytics developed and test inputs setup in Neronet compatible manner
- ▶ **Basic flow:**
  1. Specify a batch of *nerojobs* in the *experiments Excel* with parameters, inputs and other configurations
  2. Run `neroman --launch 20-28` to dispatch the jobs to the SLURM setup with autogenerated sbatch scripts and arguments
  3. Receive and check progress notifications from email
  4. See near realtime updates in the *results Excel* of analytics variable updates
  5. Receive final results data and updates to the job excel as configured
  6. Analyse, reiterate and/or publish results
- ▶ **Post conditions:** Computational experiments have been conducted in a very straightforward, effective and researcher friendly fashion



# Outline

Overview

Components

Remarks

Technical  
Overview

Neronet

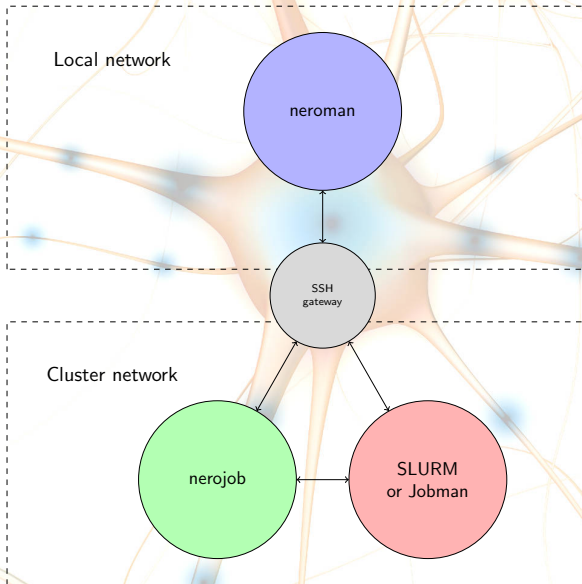
Overview

Components

Remarks

- ▶ Both `neroman` and `nerojob` are Python programs run with just the researcher's privileges on any modern \*nix
- ▶ SSH is used for intercomponent communication (user's existing ssh keys, ssh configs and privileges are used)
  - ▶ `neroman` and `nerojob` exchange data and files via their CLI and Python interfaces
  - ▶ `neroman` and `nerojob` communicate with SLURM and/or Jobman using their APIs and/or CLIs
- ▶ The system is ment to be very easy to setup, lightweight and the usability good for several types of uses.

# Communication



- ▶ Both are existing standard job scheduling systems for Linux clusters.
- ▶ They have both CLI and API interfaces and are expected to be presetup by research systems administrators.
- ▶ SLURM is currently used by Triton and CSC. Jobman could be easily setup for the gpu cluster used by deep learning researchers at Aalto.

- ▶ A daemon administered and configured by the researcher herself
- ▶ Should be run on a system with two way SSH access to the cluster nodes
- ▶ Key functionality
  - ▶ Facilitate and standardize experiment specification
  - ▶ Batch submit experiment jobs to SLURM/Jobman with autogenerated job scripts (sbatch)
  - ▶ Send email notifications with progress data
  - ▶ Facilitate monitoring and control of running jobs
  - ▶ Autocollect key job results into a researcher specifiable format (f. ex. Excel)
  - ▶ Facilitate experiment analysis and history management
  - ▶ Lightweight and extendable with custom functionality
  - ▶ Configurable via YAML files

- ▶ A program ment to be started by SLURM/Jobman on any modern \*nix (cluster nodes) on job launch
- ▶ Key functionality
  - ▶ Fetch and send computing environment information to `neroman`
  - ▶ Monitor experiment job progress (parse output logs and data files (eg. CSV, JSON))
  - ▶ Autocollect and send information and data to `neroman`
  - ▶ Interact with SLURM/Jobman as specified (eg. autotermination based on poor experiment progress)
  - ▶ Lightweight and extendable with custom functionality
  - ▶ Configurable via YAML files

# Outline

Overview

Components

Remarks

Technical  
Overview

Neronet

Overview

Components

Remarks

- ▶ A server (neroman daemon) per user approach is chosen because
  - ▶ easy minimalist setup (easy to try)
  - ▶ no need for special privileges
- ▶ SSH is used because
  - ▶ it is an existing standard in most modern research systems and clusters
  - ▶ user's existing ssh keys and ssh configs provide an easy and effective way to provide safe networking
  - ▶ no need for network, port routing or privileges adjustments
- ▶ Python 3 is used because
  - ▶ available in most modern \*nix systems
  - ▶ has good support for the system's needs
  - ▶ it is easy to learn and familiar to most computational researchers
  - ▶ Many research libraries use it (Scipy, Numpy, Theano, Lasagne, Pylearn2)