

Progress report

Neronet

*Toolbox for managing the training
neural networks*

CSE-C2610
Software Project

Aalto University

April 13, 2016

Outline

Introduction

Results

Demo

Quality

Effort

Retros

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Introduction

Goals

Our goal is to develop a tool for computational researchers to enable easy

- ▶ specification and management of experiment queues
- ▶ batch submission of experiment jobs to computing clusters
- ▶ monitoring of ongoing experiments' logs and parameter values
- ▶ access to experiment information during and after the run
- ▶ configurable notifications on experiment state and progress
- ▶ configurable criteria for experiment autotermination
- ▶ logging of experiment history

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Introduction

What

In essence the product is a Python-based tool that enables computational researchers to conduct their research more effectively.

- ▶ It utilizes SSH and TCP sockets to distribute the computational workload into computer clusters. It supports the Slurm job and resource manager but can function without it as well.
- ▶ It is framework agnostic in that it permits the use of a very wide variety of tools to actually conduct the computing.

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Results

Sprint 0

Goal: *Team building and preparing for sprint 1* **Done**

Product Backlog Items: *None*

Sprint 0 took a lot of effort from us since the project topic was very challenging to dive into. Also none of us had done this course before. Interviews with Jelena & Simo helped us to understand the project.

We were proud of our efforts in the sprint.

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Results

Sprint 1

Goal: *Develop a prototype that offers the most basic functionality via a CLI*

Product Backlog Items:

- ▶ US1: As a user, I want a basic user guide that would cover the installation of Neronet and its use via CLI. **Done**
- ▶ US2: As a user, I want to specify clusters by address and type to specify my computing resources. **Done**
- ▶ US3: As a user, I want to specify experiments by name, files and parameters. **Done**
- ▶ US4: As a user, I want to submit experiments to unmanaged nodes. **Done**
- ▶ US5: As a user, I want an experiment status report so that I can review experiment status details. **Done**

Just a prototype, not ready for release to users.

Results

Sprint 2

Goal: *Develop a stable version for end user testing*

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Results

Sprint 2

Goal: *Develop a stable version for end user testing*

Product Backlog Items:

- ▶ US6: as a user, I want to set preferences (name, email, default cluster) **Done**
- ▶ US7: As a user, I want my experiment config attributes to support generation of value combinations **Done**
- ▶ US8: As a user, I want my experiment specifications to be able to inherit properties **Done**
- ▶ US9: As a user, I want to see a basic status report in the gui **Done**
- ▶ US10: As a user, I want the program to enable easy setup. **Done**

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Goal: *Develop a stable version for end user testing*

Product Backlog Items:

- ▶ US6: as a user, I want to set preferences (name, email, default cluster) **Done**
- ▶ US7: As a user, I want my experiment config attributes to support generation of value combinations **Done**
- ▶ US8: As a user, I want my experiment specifications to be able to inherit properties **Done**
- ▶ US9: As a user, I want to see a basic status report in the gui **Done**
- ▶ US10: Aas a user, I want the program to enable easy setup. **Done**

Stable but very limited functionality.

Results

Sprint 3

Goal: *Finish asynchronous system functionality and create a GUI mockup*

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Results

Sprint 3

Goal: *Finish asynchronous system functionality and create a GUI mockup*

Product Backlog Items:

- ▶ US11: As a user, I want to save important information about my clusters **Done**
- ▶ US12: As a user, I want to group my clusters **Done**
- ▶ US13: As a user, I want to delete obsolete versions of my experiments **Done**
- ▶ US14: As a user, I want configurable criteria for experiment warnings and autotermination **Done**
- ▶ US15: As a user, I want a status report so that I can get an overview **Done**

Results

Sprint 3

Goal: *Finish asynchronous system functionality and create a GUI mockup*

Product Backlog Items:

- ▶ US11: As a user, I want to save important information about my clusters **Done**
- ▶ US12: As a user, I want to group my clusters **Done**
- ▶ US13: As a user, I want to delete obsolete versions of my experiments **Done**
- ▶ US14: As a user, I want configurable criteria for experiment warnings and autotermination **Done**
- ▶ US15: As a user, I want a status report so that I can get an overview **Done**

Rewriting, refactoring and development under the hood.

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Results

Sprint 4

Goal: *Publish Neronet as an open source project*

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Results

Sprint 4

Goal: *Publish Neronet as an open source project*

Product Backlog Items:

- ▶ US16: As a user, I want to have example use cases that I can study **Done**
- ▶ US17: As a user, I want to use basic functionalities via gui **Done**
- ▶ US18: As a user, I want to be able to easily visualize variable changes over time. **Done**
- ▶ US19: As a maintainer, I want a clear maintenance guide **Done**
- ▶ US20:: As a user, I want to have a helpful web community to help me out **Done**

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Results

Sprint 4

Goal: *Publish Neronet as an open source project*

Product Backlog Items:

- ▶ US16: As a user, I want to have example use cases that I can study **Done**
- ▶ US17: As a user, I want to use basic functionalities via gui **Done**
- ▶ US18: As a user, I want to be able to easily visualize variable changes over time. **Done**
- ▶ US19: As a maintainer, I want a clear maintenance guide **Done**
- ▶ US20:: As a user, I want to have a helpful web community to help me out **Done**

Improving user experience; set up a continuous integration system

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Results

User testing

We reached out to Triton admin and he gave us great feedback.

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Results

User testing

We reached out to Triton admin and he gave us great feedback.

- ▶ Synchronizing files between clusters and client computers:
 - ▶ Our system so far has used rsync and it has exceeding amounts of I/O operations when used with large experiments.
 - ▶ We should use scp as to reduce disk operations.
 - ▶ Home directory cannot be used in some clusters, we need to have an option to choose working directory.

Results

User testing

We reached out to Triton admin and he gave us great feedback.

- ▶ Synchronizing files between clusters and client computers:
 - ▶ Our system so far has used rsync and it has exceeding amounts of I/O operations when used with large experiments.
 - ▶ We should use scp as to reduce disk operations.
 - ▶ Home directory cannot be used in some clusters, we need to have an option to choose working directory.
- ▶ In general he liked our usability.

Results

User testing

We reached out to Triton admin and he gave us great feedback.

- ▶ Synchronizing files between clusters and client computers:
 - ▶ Our system so far has used rsync and it has exceeding amounts of I/O operations when used with large experiments.
 - ▶ We should use scp as to reduce disk operations.
 - ▶ Home directory cannot be used in some clusters, we need to have an option to choose working directory.
- ▶ In general he liked our usability.

Our PO gave us some input regarding usability after his initial testing.

Results

User testing

We reached out to Triton admin and he gave us great feedback.

- ▶ Synchronizing files between clusters and client computers:
 - ▶ Our system so far has used rsync and it has exceeding amounts of I/O operations when used with large experiments.
 - ▶ We should use scp as to reduce disk operations.
 - ▶ Home directory cannot be used in some clusters, we need to have an option to choose working directory.
- ▶ In general he liked our usability.

Our PO gave us some input regarding usability after his initial testing.

- ▶ Test connection when specifying clusters to assure the ssh address is valid. FIXED
- ▶ Error and success messages should be more specific.
- ▶ Some bugs were also found and fixed.

(Live demonstration of the current release.)

1. Installation
2. Cluster & experiment configuration
3. Status review
4. Experiment submission
5. Viewing updates
6. Viewing plots
7. GUI

Quality

Definition of done:

- ▶ We defined **Done** in three levels: BI, sprint and project
- ▶ BI level: Unit tests are written where applicable, Code conforms to guidelines (PEP8), Code is commented and documented, Independent peer review has been completed
- ▶ Sprint level: All BIs in the sprint backlog are *Done*, Reasonable automated tests written, Comprehensive system tests are conducted, reported and reviewed, Sprint goal is achieved

Quality

Definition of done:

- ▶ We defined **Done** in three levels: BI, sprint and project
- ▶ BI level: Unit tests are written where applicable, Code conforms to guidelines (PEP8), Code is commented and documented, Independent peer review has been completed
- ▶ Sprint level: All BIs in the sprint backlog are *Done*, Reasonable automated tests written, Comprehensive system tests are conducted, reported and reviewed, Sprint goal is achieved
- ▶ Updates to DoD:
 - ▶ All updates to DoD at the end of each sprint are written in the retrospectives section
- ▶ We have mainly followed our DoD as planned, but in sprint 3 we didn't yet have automated tests so we replaced that with manual user testing. Documentation of system testing could also be better.

Used QA practices and tools:

- ▶ Commenting & documentation – forces to rethink from another perspective, facilitates peer review
- ▶ Python standard unittest framework – white-box test automation
- ▶ Travis CI runs unit tests automatically after every push
- ▶ Functional testing – manual black-box testing based on specs
- ▶ Peer review – quality assurance

Quality

Relevant quality attributes:

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Relevant quality attributes:

- ▶ **Usability** – We have put a lot of effort in creating a helpful user manual, start guide and example use cases to help newbies better understand our product. Some error and success messages could be more helpful though, but we will make them better right at the start of the next sprint.

Relevant quality attributes:

- ▶ **Usability** – We have put a lot of effort in creating a helpful user manual, start guide and example use cases to help newbies better understand our product. Some error and success messages could be more helpful though, but we will make them better right at the start of the next sprint.
- ▶ **Reliability** – Our software's reliability should be good at this point. Tests haven't shown many unexpected crashes.

Quality

Relevant quality attributes:

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Relevant quality attributes:

- ▶ **Extendability** – We have paid attention to making our software easily extendable from the start, and in the last sprint we put effort in creating a maintenance guide so it's easier to extend our software even without having been in the development team.

Relevant quality attributes:

- ▶ **Extendability** – We have paid attention to making our software easily extendable from the start, and in the last sprint we put effort in creating a maintenance guide so it's easier to extend our software even without having been in the development team.
- ▶ **Performance** – We found out at the end of this sprint that our software using rsync might cause too much I/O operations in some cases. We will replace that with scp and after that our software's performance should be pretty good.

Effort

Spent and budgeted effort in hours by team member and sprint:

S	Samuel	Teemu	Tuomo	Joona	Iiro	Matias
0	140/50	36/35	45/35	40/35	36/35	43/35
1	52/30	37/33	42/33	46/33	32/33	37/33
2	42/30	27/33	41/33	25/33	27/33	30/33
3	24/15	28/33	14/33	23/33	20/33	27/33
4	33/15	33/33	22/33	26/33	23/33	31/33
5	0/15	0/33	0/33	0/33	0/33	0/33
6	0/20	0/25	0/25	0/25	0/25	0/25
	291/175	161/225	164/225	160/225	138/225	168/225

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Effort

Spent and budgeted effort in hours by team member and sprint:

S	Samuel	Teemu	Tuomo	Joona	Iiro	Matias
0	140/50	36/35	45/35	40/35	36/35	43/35
1	52/30	37/33	42/33	46/33	32/33	37/33
2	42/30	27/33	41/33	25/33	27/33	30/33
3	24/15	28/33	14/33	23/33	20/33	27/33
4	33/15	33/33	22/33	26/33	23/33	31/33
5	0/15	0/33	0/33	0/33	0/33	0/33
6	0/20	0/25	0/25	0/25	0/25	0/25
	291/175	161/225	164/225	160/225	138/225	168/225

- ▶ Our team had comparatively little experience
- ▶ The project's problem domain was challenging to understand
- ▶ Studying existing products and technologies took a lot of time
- ▶ At the beginning scrum master served also as team leader and lead developer

Retros: Sprint 0

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Sprint planning:

- ▶ BI clarity and simplicity (user guide helps)
- ▶ It might have been better if the PO had created the stories from scratch -Matias, Tuomo
- ▶ We should actively seek more input from PO when developing the user guide
- ▶ we should make sure we reserve enough time for the actual story selection on Monday -Matias

Retros: Sprint 0

Daily scrums:

- ▶ We have mostly been doing teamwork, so there has been little new info in the scrums -Matias -Joona -Teemu
- ▶ They have been overly long and they have extended due to inexperience.
- ▶ People are late.

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Retros: Sprint 0

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Teamwork sessions:

- ▶ Sessions are too long and sometimes people get hungry
- ▶ Generally someone has to leave early or comes late

Retros: Sprint 0

Tools:

- ▶ Flowdock is good x6
- ▶ Agilefant has a steep learning curve. -liro
- ▶ People tend to forget to log their time at agilefant
- ▶ Github hasn't been used much. Hope to use it more during future sprints
- ▶ Floobits is very buggy.
- ▶ Top 3 tools: 1) GitHub 2) Flowdock 3) Agilefant
- ▶ Worst 3 tools: 1) Floobits 2) Six tactics 3) Agilefant

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Retros: Sprint 0

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

How teamwork could be improved:

- ▶ People should be more on time
- ▶ Scrum Master shouldn't have to work as a team leader too.
- ▶ Hard to think of improvements since we haven't really started coding yet

Retros: Sprint 1

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Improvements since sprint 0:

- ▶ Replaced six tactics with Team spirit recap
- ▶ Balanced our team's power structure by selecting a team leader for each sprint.
- ▶ Punctuality: Many have improved, Iiro hasn't. Samuel has also been more absent. Everybody should try to be more punctual or at least inform early about being late.
- ▶ Ambiguity in user stories got less of an issue due to the user guide

Retros: Sprint 1

Evaluating practises:

- ▶ Team leader per sprint:
 - ▶ Matias: Great not to give Samuel all responsibility
 - ▶ Iiro: Might be a bit confusing for the PO and I'm not sure how I'll manage it in the next sprint
 - ▶ Tuomo, Teemu, Joonas and Samuel like the idea
- ▶ Pair programming
 - ▶ Matias: Works fine, a lot of time used for coordination within the pair
 - ▶ Iiro: I feel that our pair work might have been a bit inefficient in development and testing, but it helped a lot in planning and documentation
 - ▶ Teemu: Difficult to share work. Otherwise works well.
 - ▶ Samuel: In sprint 2 I suggest we program in pairs but do not employ pair programming, work together as they best see fit.

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Retros: Sprint 1

Evaluating practises:

- ▶ Developing user guide first:
 - ▶ Matias: We shall see
 - ▶ Iiro: I feel we have strayed from the reason we began to implement a user guide. Now it feels to be restrictive rather than descriptive.
 - ▶ Joona: I feel that it is very good as it reflects the requirements of the PO.
 - ▶ Samuel: It should be considered as a sort of prototype. It is not useful to hang ourselves to it.
- ▶ Sprint planning
 - ▶ Matias: Difficult to break user stories into useful and small tasks
 - ▶ Iiro: Our stories were already small, they were difficult to break even smaller.
 - ▶ There was slight contradictions in understanding user stories
 - ▶ Integration work took a lot more time than expected

Retros: Sprint 1

Evaluating practises:

- ▶ Daily Scrums:
 - ▶ Matias: Worked fine this time, not of much use
 - ▶ Iiro: Need to attend more of them...
 - ▶ Teemu: They could have been used more in integration and inter pair coordination
- ▶ Teamwork sessions:
 - ▶ Matias: times fine, people are away too often, peer reviews should be distributed evenly throughout the sprint
 - ▶ Iiro: Sometimes too much commotion
- ▶ Peer review:
 - ▶ Done in a big hurry, we should reserve more time for it
 - ▶ We should study more about how it should be done

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Retros: Sprint 1

Evaluating tools:

- ▶ flowdock:
 - ▶ Matias: Hasn't used much
 - ▶ liro: doesn't like the idea of PO reading the messages
- ▶ Whatsapp:
 - ▶ Matias: reads often
 - ▶ liro: I don't keep my phones internet always on so it hasn't been optimal
- ▶ Agilefant:
 - ▶ Matias: No complaints
 - ▶ liro: Still feels bit rigid

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Retros: Sprint 1

Evaluating tools:

- ▶ Google calendar:
 - ▶ Matias: works fine
 - ▶ Iiro: Haven't used it much after we scheduled regular times
- ▶ Github:
 - ▶ Matias: works fine
 - ▶ Iiro: The usage still needs some working. There are some files that shouldn't have been pushed to the remote
- ▶ Floobits:
 - ▶ Matias: Hasn't used much
 - ▶ Iiro: Good for meetings

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Retros: Sprint 1

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Improvement to teamwork:

- ▶ people are away too often -Matias
- ▶ peer reviews should be distributed evenly throughout the sprint -Matias
- ▶ The pace is not always the best -Matias
- ▶ Pair programming needs to be streamlined -Iiro
- ▶ Let's have a team review of the whole project at the beginning of next sprint.
- ▶ Let's design all interfaces at the start of the sprint.

Retros: Sprint 1

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Implementation of improvements:

- ▶ Team review in the beginning of January
- ▶ More whole team reviews of matters and changes
- ▶ Samuel tries to be less of a lead developer

Retros: Sprint 2

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Visiting and updating the team's DoD:

- ▶ Review and update DOD more often
- ▶ Moved system tests from BI level to sprint level
- ▶ Removed the percentage from system test coverage
- ▶ Implement automated functional tests
- ▶ Added specifics to guidelines (PEP8)

Visiting and reviewing the commitments done in the last sprint retrospective

- ▶ We managed to do some more team reviews of tricky stuff
- ▶ There was slight improvement in punctuality
- ▶ Not all planned improvements were deemed important

Retros: Sprint 2

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Identifying things the team should start doing

- ▶ Consider employing CI with Travis
- ▶ System test automation
- ▶ Iiro starts being early (and being a good team leader)

Identifying things the team should stop doing

- ▶ Let's try not to mix too many topics everywhere all the time

Identifying things the team should continue doing

- ▶ Periodic team reviews

Listing actionable commitments

- ▶ (Actionable = has clear steps to completion and acceptance criteria. f.ex "Check in code at least twice per day: before lunch and before going home")
- ▶ Joona sets up and teaches CI with Travis
- ▶ Write automated system test and run them daily
- ▶ Iiro starts being early (and being a good team leader)
- ▶ More experiments and trials, less debating on whether to do what
- ▶ List items in sprint planning that should be worked on when time permits

Retros: Sprint 3

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Visiting and updating the team's DoD:

- ▶ No updates, after the last update the result was good

Visiting and reviewing the commitments done in the last sprint retrospective

- ▶ Unfortunately we were busy with other stuff so CI and automated tests weren't set up
- ▶ More experiments and trials, less debating on whether to do what. Somewhat successful
- ▶ List items in sprint planning that should be worked on when time permits. Successful

Retros: Sprint 3

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Identifying things the team should start doing

- ▶ CI and automated tests -Joona
- ▶ Following the hours spent by all the developers and properly adjusting them

Identifying things the team should continue doing

- ▶ Hold daily scrums only when some of the developers feel the need for them (At some point during this sprint we started holding daily scrums less often and found out they were often not necessary)

Listing actionable commitments

- ▶ (Actionable = has clear steps to completion and acceptance criteria. f.ex "Check in code at least twice per day: before lunch and before going home")
- ▶ Joonas sets up and teaches CI with Travis
- ▶ Everybody checks their spent hours and properly adjusts their participation so that the required 225 hours will be filled

Retros: Sprint 4

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Visiting and updating the team's DoD:

- ▶ No updates, after the last update the result was good

Visiting and reviewing the commitments done in the last sprint retrospective

- ▶ Employed CI with Travis
- ▶ Continued holding daily scrums when necessary

Retros: Sprint 4

Progress report

Neronet

Introduction

Results

Demo

Quality

Effort

Retros

Identifying things the team should start doing

- ▶ Improving unit tests
- ▶ Report hours done to agilefant after each session
- ▶ Work more in the same room to improve communication

Identifying things the team should continue doing

- ▶ Continue improving testing and continuous integration

Listing actionable commitments

- ▶ (Actionable = has clear steps to completion and acceptance criteria. f.ex "Check in code at least twice per day: before lunch and before going home")
- ▶ Everybody starts marking the hours spent after each session
- ▶ Joona continues writing tests to be used with travis CI