Meetings

Notes on meetings. Each meeting is documented under a section title of the form: `<month>-<day> <duration_in_minutes>m <event title>`

10-07 60m First skype with PO (teamPy)

Participants: team + Pyry

Agenda:

- introduction to project
- general discussion

10-09 120m Meeting Jelena

Participants: team + Jelena

Agenda:

- introduction to research group
- learning about research practices
- discussion about project

10-14 120m Drafts meeting

Participants: team - TuIiJu

Agenda:

- initial plans
- current drafts
- distribution of work

10-22 60m Quick SM-PO chat

Participants: Samuel + Pyry

Agenda:

- status overview
- schedule
- practical issues
- technical research methods

10-23 210m Artifacts workskype

Participants: Samuel, Joona, Matias

Agenda:

- Process Overview – Matias
- Product Vision – Joona
- Definition of Done – Joona

10-24 120m Artifacts workskype

Participants: Samuel, Teemu, Tuomo

Agenda:

- Product Backlog
- Technical Overview

10-25 90m Tech workskype

Participants: Samuel, Matias, Teemu, Tuomo, Iiro

Agenda:

- Discussed and studied the state of the art

10-25 120m Triton workskype

Participants: Samuel, Tuomo, Iiro, (Matias)

Agenda:

- Studied Triton and SLURM

10-28 180m Wedshop

Participants: team (team - Juho)

Agenda:

- general status check
- discussing artifacts
- researching about the state of the art

10-30 300m Frifury

Participants: team (team - Juho)

Agenda:

- Agilefant (time tracking)
- Technical
- Vision
- Other artifacts

10-30 120m First meeting with PO

Notes by Joona & Teemu

Questions we discussed

- importance of queue management functions
- technical details

Things we clarified

- requirements by the PO
- the current workflow

Decisions made

- Command prompt applications, no web UI
- Two different applications: Server and client
- Client application contacts server. The client application doesn't have to be running all the time.
- Server application contacts the clusters to run the experiments, manages the experiment queues and collects information on running experiments. The server also saves the information and notifies the client if the experiments go wrong. In an ideal situation the server application is always running.
- Queue management can use either jobman or slurm
- Server makes a .csv document about the past experiments and their outputs and the client can download it.
- Server can tell the information on the available GPU:s etc.
- Config:
  - Defines the ID of the experiment (author, subject, name, group name, git commit ID)
  - Define the variables that must be extracted and sent to the server

- – The Preconditions (minimum available disk space, expected max time, minimum RAM)
- – The files that must be sent to the cluster

Things yet to research/decide

- Simo
- Possibly in the future a web interface and user login

Roadmap

- TBD

11-03 90m Lunch with Simo

Participants: Samuel + Simo Tuomisto

Agenda:

- introduction
- general design approach
- practicalities

Hi,

I had a small chat with Samuel during lunch, but I'll gladly accompany you for an another talk.

I briefly outlined that my recommendation would be to utilize a master-slave structure where the client software would ssh to the cluster (triton, gpu machine etc), start a master deamon process and run desired number of slave jobs using applicable shell wrapper (sbatch, bash).

This master could then listen runtime communication from slaves and stash them for the client to collect. Client could then collect them via ssh.

The main reason for this recommendation is that piping ssh through multiple layers of gateways is inefficient and troublesome to configure. Typical network structure looks like this:

```
user <-> user network gateway / shell server <-> internet <-> cluster
network gateway / frontend <-> cluster node
```

Now running job in cluster node requires ssh to frontend, but if a response would be initiated by the cluster node, it would require ssh to the user shell server and from there to the user machine. As other ports besides ssh ports are not visible to the internet (for security reasons) this makes two-way communication between node and user at the least complicated.

Thus a work flow where run information gathering is initiated by the user is much more straightforward and it requires only one ssh connection. This also means that passphrase protected private ssh keys can be stashed to the user machine instead of moving them all the way to the nodes.

This scheme can be run as is on other machines. There instead of sbatch, some other shell would be used and communication would be done to localhost instead of the login node.

I also recommend that the communication between the master and slaves is done using some simple network protocol instead of by files. This would lessen the load on shared file systems.

I'm checking on giving you access to Triton some way or another.

Regards, Simo Tuomisto CS/Triton Admin

11-04 180m Wedshop

Participants: team - Juho

Agenda:

- Aalto-student contract
- Time tracking & Agilefant
- Evaluation criteria
- Technical Overview (Simo's input)
- EES participation

11-04 120m EES 01: Adapting scrum

Participants: Samuel

PO and requirements:

- Interviews for requirements gathering and/or researching feasibility of potential technical solutions
- How to modify customer requirements into user stories into the backlog
- How to specify requirements if different persons from client company have different opinions about a topic

- What is the customers role as a product owner when the team has done the service design

Teamwork:

- Scrum, scrum master and team leadership. Should a Scrum Master take the role of a Team Leader?
- Leadership in self-organizing teams. Eliciting intrinsic motivation and self-direction and initiative in team members.
- Teamwork power structure, work organization, effort distribution

Software quality:

- How to make your sprint board reflect your DoD
- Quality assurance and automated deployment (continuous integration) with Github, CircleCI and Heroku

11-06 90m Expert meeting (Simo Tuomisto)

Notes:

- Neroman could run on a web browser or be a QT application
- Excel or application shows results (multiple options)
- parameter combinations can be numerous
- SSH can be slow + file system unreliable + can crash unexpectedly
- http could be a better protocol
- SSH-Stdout-payload-scp
- Refresh/real time data
- Neroman could be http server application
- Possible implementing order nerokid-neromum-neroman
- Job scheduling
- hdf5 could be used by neromum to save the experiment output data

11-06 210m Frifury

Participants: team

Agenda:

- 6x15m Scrum
- 6x20m General discussion
- 6x40m Agilefant

- 6x20m Schedule
- 3x100m Market research (matute)
- 2x100m Prototype design (iijo)

Scrum:

Tuomo:

- Has studied Jobman
- Thought it is important to show the technical overview to the PO

Iiro:

- Has studied iPython
- We should continue researching user requirements

Matias:

- Has thought about teamwork practices and Agilefant
- We should communicate more with the end users and then improve the accuracy of the software design

Teemu:

- Thinks it is very important to have good plan before starting to code
- Feels the current plan is pretty reasonable

Joona:

- Has thought about teamwork practices and Agilefant
- Thinks we need to define classes and make class diagrams
- Need input how to specify experiments

Samuel:

- Has been doing techincal overview to prepare to our interviews
- Has been interviewing simo
- Redone agilefant
- Participated in EES

Notes on existing tools

Blocks:

- saving and resuming training
- monitoring, analyzing values
- Theano operations
- algorithms to optimize a model

Ipython:

- manages job distibution
- manages parallelism
- seemingly quite similar to Slurm

Ask the custom whether they have used Blocks/Ipython/Sacred etc

11-06 180m Satscrum in Hangout

Participants: team

Agenda:

- 6x10m Scrum
- 6x30m Discussing results
- 4x120m Prototype development (iitutejo)
- 1x80m Prototype development (ma)
- 6x20m Discussing results

Tasks:

- iitu by 17:00: come up with a design in how the Python package and version management should be dealt in our project. Google and check existing projects.
- jote by 17:00: start sketching the prototype.

Homework:

- Everyone installs and tests a Python 3.5 environment

Homework

On Ubuntu 14.04:

```
sudo add-apt-repository ppa:fkrull/deadsnakes
sudo apt-get update
sudo apt-get install python3.5

python3.5 -m venv projects/neronet/.venv
```

Resources:

- [PEP-0405](#)
- [venv](#)

11-09 90m Review of product vision

Participants: team + PO

Agenda:

- 20:00: 7x10m Discussing agenda
- 20:10: 7x10m Summary of past weeks
- 20:20: 7x20m Discussing existing tools
- 20:40: 7x20m Updating the product vision
- 21:00: 7x20m Discussing backlog management
- 21:20: 7x10m Discussing contacting other researchers

Links:

- https://floobits.com/smarisa/neronet/file/doc/notes_on_meetings.md:362
- https://floobits.com/smarisa/neronet/file/doc/project_information.md:35
- https://floobits.com/smarisa/neronet/file/doc/project_information.md:393
- https://github.com/smarisa/neronet/blob/sprint/doc/product_vision.pdf
- https://floobits.com/smarisa/neronet/file/doc/notes_on_tools.md:115

Notes:

- Keep studying LadderNet's similar functionalities and sacred. Could these be utilized with our project? Possibly also check Blocks.
- Query requirements and other ideas after prototype is finished

- Targets: Simo, Blocks lead developer at GitHub and Torch developers, Google group, Reddit
- Example: Hey, we've been thinking a tool like ... would be great. What do you think?

- Later when the product has taken some shape and we've had some user testing, possibly start marketing more clearly

Backlog management

See notes on tools.

Agilefant

- As a user, I want to specify experiments so that they are easy to manage and run.

  - As a user, I want to alter the experiments data programmatically so that I can manage it more effectively.
  - As a user, I want to batch import experiments data so that it is easy to migrate.
  - As a user, I want to specify experiments by name, files and parameters to distinguish them.
  - As a user, I want to specify experiment collections to help manage them.
  - As a user, I want to edit existing experiments and collections so that I can update them.

- As a user, I want configurable views to my experiments and collections so that I can review them effectively.

  - As a user, I want a compact CLI summary view so that I can review them remotely.
  - As a user, I want a compact CLI collections view so that I can review them remotely.
  - As a user, I want a compact CLI experiment view so that I can review experiment details remotely.
  - As a user, I want an interactive and configurable GUI view to review my experiments.

- As a user, I want to manage the running of experiments so that my resources are best used.

  - As a user, I want to configure clusters to best utilize my resources.
    - As a user, I want to configure clusters by address and type to specify my computing resources.
      - As a user, I want to configure unmanaged nodes to utilize simple clusters.

- · As a user, I want to configure Slurm cluster gateways to utilize Slurm clusters.
  - – As a user, I want to submit individual experiments to be run so that I can later analyse them.
  - – As a user, I want to submit batches to be run so that I can later analyse them.
  - – As a user, I want to specify experiment batches and queues to best utilize my resources.
- As a user, I want to monitor ongoing experiments so that I know what to do next.
  - – As a user, I want to monitor batch and experiment run status so that I know what is going on.
- As a user, I want to access the data of my past experiments so that I can analyse them.
  - – As a user, I want to backup and extract the experiments data files manually so that I remain in full control.
  - – As a user, I want to batch export experiments data so that I remain in full control.

11-11 360m Wedshop

Participants: team

Agenda:

- 13:00: 6x15m Scrum (Python 3.5 environment setup homework)
- 13:15: 6x45m Team spirit recap
- 14:00: 2x15m Updating process & vision artifacts (jote)
- 14:00: 3x15m Updating technical & dod artifacts (tuma)
- 14:35: 6x110m Backlog item planning poker
- 17:20: 2x40m Neroman development (iite)
- 17:20: 3x40m Nerokid and Neromum development (tuma)
- 18:00: 6x15m Discussing results and making decisions
- 18:15: 5x35m Preparing prototype demo
- 18:50: 6x10m Updating agilefant
- 19:00: Going home!
- 16:20: 3x60m Tests developement (iimatu)

Team spirit recap

- Mission: Why we exist

  - Create useful software for Pyry (and others)
  - We are doing this project to learn: software development, re, architecture, project management, quality assurance, Scrum, communication with client
  - We want grade five, quality award

- Values: What we believe in and how we will behave

  - Superior quality
  - Self-development
  - Respect
  - TBD, Achievement? – We work together to deliver superior results and achieve understanding.

- Vision: What we want to be

  - We want to see ourselves as the best of the course teams
  - We want to win the Quality award!
  - We want to get grade 5+.
  - We want to get an awesome reference (GitHub repo) that we can market on our future job applications.
  - We want our tool to serve people in such a way that a community of users develops around it and continues it's development. We want to launch a successful open source project, which we can speak proudly of even years from now.

- Strategy: What our competitive game plan will be

  - Objective: Ace the course and develop a very useful and popular tool
  - Scope: See product vision
  - Advantage: We have high motivation, we meet in person every week, active and responsible Scrum Master

- Balanced scorecard

Neroman development

- Database daemon + CLI client + GUI client

11-13 60m Demo exercise

Participants: team

Agenda:

- 13:00 6x10m Agree demo presentation tasks
- 13:10 6x40m Practice presentation
- 13:55 6x5m Pause

11-13 60m Sprint retrospective

Participants: team

Agenda:

- 14:00 6x10m Sketch retrospective structure
- 14:10 6x30m Discussion
- 14:40 6x10m Review
- 14:50 6x5m Get everyone's aalto account id (username), send them to Simo
- 14:50 6x10m Pause

Evaluate and rank teamwork practices

Sprint planning:

- backlog items must be clear and simple -teemu
- backlog items have been unclear, but the user guide probably helps -joona
- it would have been better if the PO had created the stories from scratch -matias, tuomo
- the PO should give input when developing the user guide -joona
- we should do it on monday -tuomo, joona, teemu
- we should make sure we reserve enough time for the actual story selection on Monday -matias

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.

Teamwork sessions:

- sessions are too long and sometimes people get hungry.
- generally someone has to leave early or comes late
- we balanced the session lengths(wed 6h fri 5h).

Tools:

- flowdock is good x6
- for remote work we have been using google hangout and skype. Skype has proven to be the most stable.
- for faster communication we are using whatsapp.
- agilefant has a steep learning curve. -Iiro
- people tend to forget to log their time at agilefant.
- hope to use more github during sprints
- floobits ain't very good. Doesn't seem to work in its intended purpose.

Top 3 tools

1. GitHub
2. Flowdock
3. Agilefant

Worst 3 tools

1. Floobits
2. Six tactics
3. Agilefant

How teamwork could be improved

- People should be more on time.
- hard to think on improvements on sprint 0

Usernames: smarisa, perat1, marjakj1, pitkanl5, blomqvt1, tahkai1

11-13 120m Sprint 0 demo

Participants: team + coach

Agenda:

- 15:00 7x10m Discussing agenda -Samuel
- 15:05 7x20m Process overview -Joona
- 15:25 7x20m Retrospective results -Teemu
- 15:45 7x10m Product vision -Matias
- 15:55 7x15m Technical overview -Tuomo

- 16:10 7x10m Definition of done -Iiro
- 16:20 7x10m Backlogs -Iiro
- 16:30 7x15m Prototype demo -Samuel
- 16:45 7x15m Post review discussion

11-16 120m Sprint 1 planning

Participants: team + PO

Agenda:

- 11:00: Agenda
- 11:10: Review of sprint 0 results
- 11:20: Sprint team leader's word
- 11:30: Discussing the budget
- 11:40: Discussing the product backlog
- 11:50: Definition of the sprint goal
- 12:10: Selection of backlog items (BIs)
- 12:30: Clarification of BIs (user guide)

What is sprint planning: *A time-boxed event of 1 day, or less, to start a Sprint. It serves for the Scrum Team to inspect the work from the Product Backlog that's most valuable to be done next and design that work into Sprint backlog.*

Sprint team leader

- S0: Samuel
- S1: Joona
- S2: Iiro
- S3: Tuomo
- S4: Matias
- S5: Teemu
- S6: Samuel

Tasks:

- Scheduling & communication with all parties
- Planning and preparing meeting agendas
- Keeping track of progress and ensuring achievement of sprint goals
- Making sure the *big picture* progresses optimally
- Making sure the sprint review & demo is excellent
- Maintaining and boosting team spirit and motivation

Feedback is given to team leader at the end of sprint.

Budget

Fact: We have at least 10h of storyless work per developer, thus only 5x25h of developer time budget left for story fullfilment work.

Fact: Total story points in product backlog: 26

Assumption: We have two two person teams doing pair programming.

Result: The budget is 50h of pair time

Assumption: One story point equals 3h of pair time.

Result: We can do 16 story points this sprint. Which?

Sprint goal

References:

- Scrum.org: Scrum Glossary
- ScaledAgileFrameworks.com: Sprint goals
- RomanPichler.com: Effective sprint goals
- Luxoft.com: 7 sprint goal patterns for building great teams

Quotes:

- *Sprint Goals are a high level summary of the business and technical goals that the team and Product Owner agree to accomplish in a sprint.*
- *Sprint Goal: a short expression of the purpose of a Sprint, often a business problem that is addressed. Functionality might be adjusted during the Sprint in order to achieve the Sprint Goal.*
- *The Sprint Goal, or mission objective, is not repeated over the course of many sprints. Each sprint should have its own unique mission.*
- *Setting the Sprint Goal is the Product Owner's responsibility, but crafting it is a shared responsibility of the Scrum Team, including PO.*
- *The Sprint Goal encourages initiative in multiple areas – teamwork, technology, quality, and mindset. However, while improving those areas, make sure that you are creating a potentially releasable increment by the end of the Sprint.*

Benefits:

- Align team members to a common purpose
- Ensures that everyone moves in the same direction
- Facilitates prioritisation

- Facilitates teamwork
- Facilitates giving and analysing feedback
- Supports communication

Candidates:

- Develop a prototype that offers the most basic functionality via a CLI
- (any alternatives?)

Sprint backlog

Sprint 1 in Agilefant