

## 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

### **10-22 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 worksype**

Participants: Samuel, Teemu, Tuomo Agenda:

- Product Backlog
- Technical Overview

### **10-25 90m Tech worksype**

Participants: Samuel, Matias, Teemu, Tuomo, Iiro Agenda:

- Discussed and studied the state of the art

### **10-25 120m Triton worksype**

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)
- Market research
- Prototype development
- 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 300m Frifury**

Participants: team Agenda:: - 7x15m Scrum - 3x120m Market research (matute)  
- 3x120m Prototype design (iijoju) - 7x40m Discussing results - 2x120m Market research (iijo) - 4x120m Prototype development (matuteju)