Data processing with XNAT and OpenStack

Christoph Jansen - HTW Berlin

Christoph Jansen

Computer Science Student:
International Media and Computing (Master) at HTW Berlin

Student Research Assistant:
QMROCT - Project

Internship at AMC:
for 1 Month

Content

The SOMNO.Netz Project
The QM-ROCT Project

Overview: server infrastructure
XNAT: data archiving and job pipelines
OpenStack: processing jobs in virtual machines

Todo: enabling new use-cases with Workflow-Management

SOMNO.Netz Project

Supporting Medical Association: DGSM (German Sleep Society)

- More than 2000 members, 300 sleep laboratories
- Research collaboration in sleep medicine
- Defines quality standards for sleep laboratories
- Organizing multicentric clinical trials

SOMNO.Netz Project

Proceeding to verify the quality of laboratories

- Website and database for reviewers and labs

Patient data and studies (e.g. Polysomnography)

- Collecting patient data for clinical studies
- Set of diagnostic tests
- Stored in XNAT
- Can be processed

QM-ROCT Project

Quality Management for Retinal Optical Coherence Tomography

- Charité Berlin, Beuth University of A.Sc. Berlin, HTW Berlin

Retinal OCT

- method for 3D and 2D scans of the human eye

Problem

- researchers depend on good quality scans
- how to extract good scans from huge databases

Quality Measures

Charité Berlin

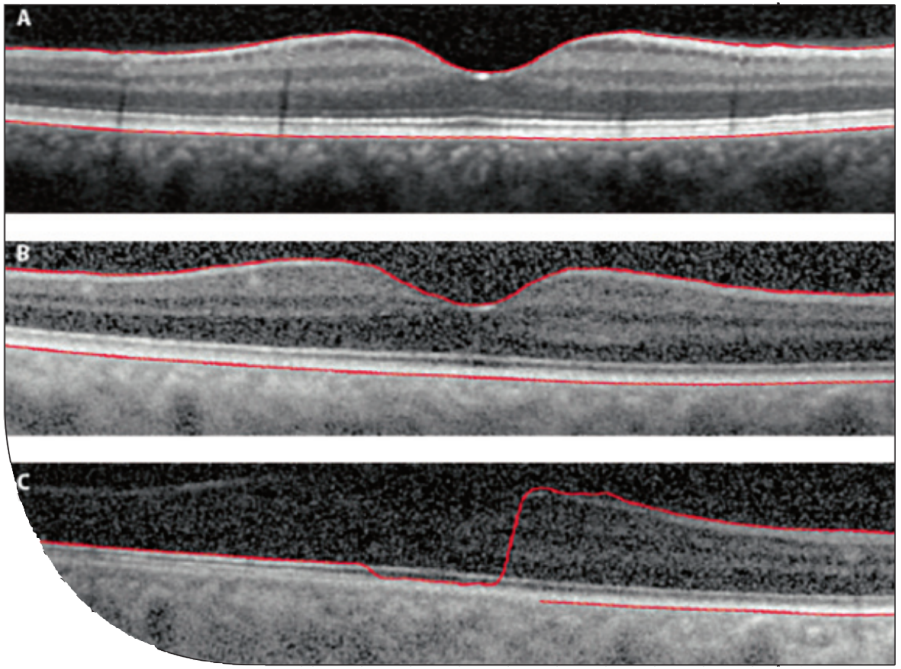
- Medical department

Tasks

- Medical know-how
- Defining quality indicators

e.g.

- Is a scan good enough for a certain study?



Algorithms

Beuth University

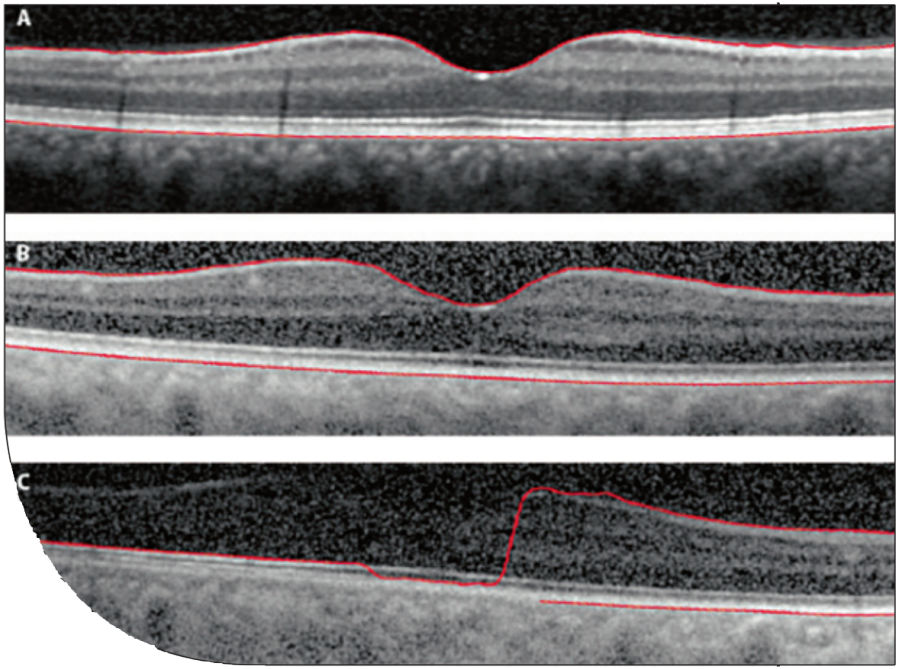
- Physics department

Tasks

- Matlab: image processing
- Developing algorithms

e.g.

- Segmentation lines
- Amount of noise



Infrastructure

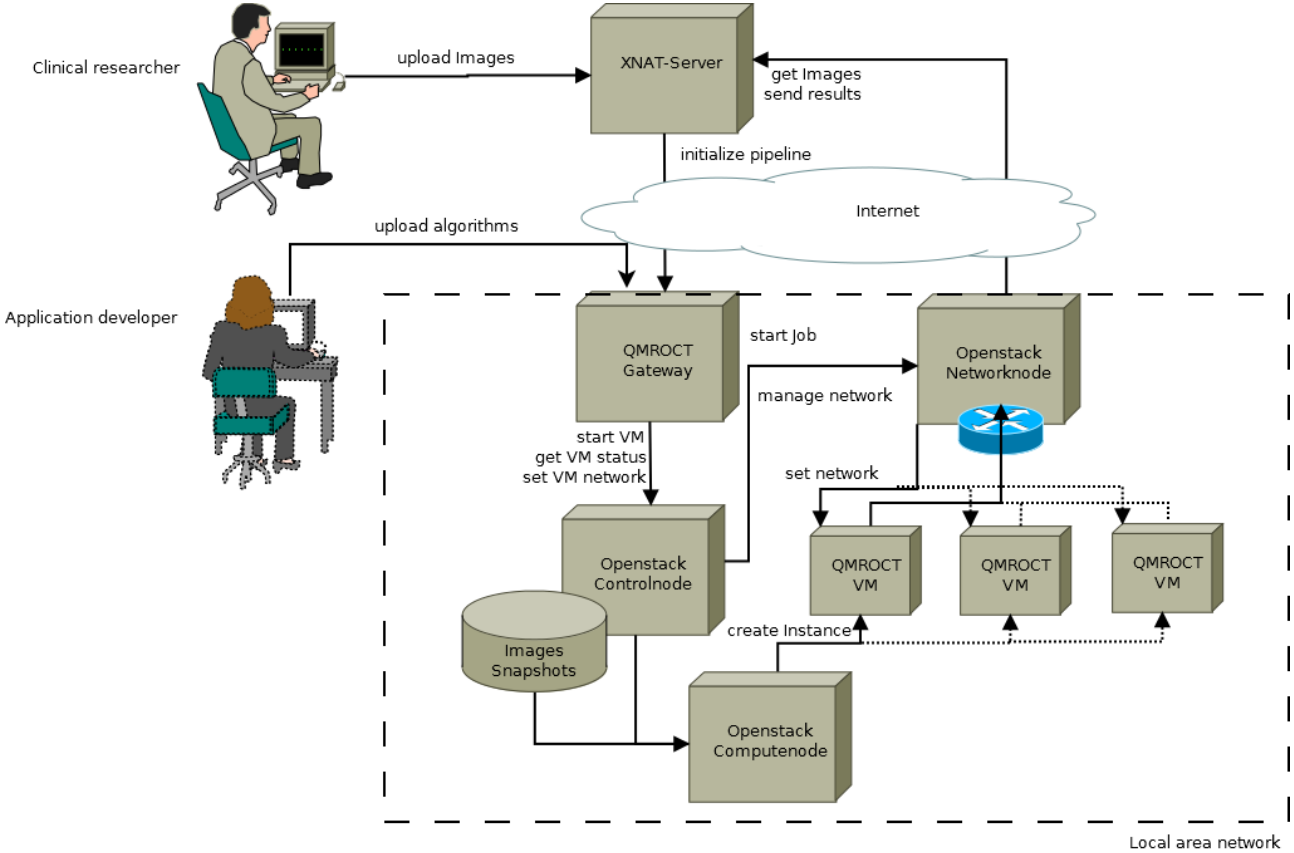
HTW Berlin

- Computer Science department

Tasks

- Developing a server infrastructure
  (archive data and run algorithms)
- Web interface for medical research (XNAT)
- Web interface for algorithm developers (Custom)

Server Infrastructure



XNAT

- Start Pipeline

Gateway

- Receive job
- Forward to

OpenStack

- VMs
- Send result

Communication

- REST APIs

XNAT

Filesystem

- Stores DICOM files
- Any other file

Database

- Stores metadata
- Extendable data types
- Stores results - e.g. Key Quality Indicators (KQI)

XNAT - Pipelines

Context of scan files

- XNAT → Projects → Subjects → Sessions → Scans

Parameters are parsed to remote application

- Project, Subject, Session
- Hostname, User (Token), Password (Token)

Application (e.g. shell script)

- Access REST API via CURL using parameters

XNAT - Researcher Interface

Scans

Scan	Type	Series Desc	Usability	Files	Note
⊞ 302000	Volume IR	Volume IR	usable	Show Counts	
⊞ 302001	Volume IR	Volume IR	usable	Show Counts	
				Total Counts	

History

Action	Launch Time	Status	Note
Uploaded File	2014-01-23 12:49:36.371	Complete	
qmroct-pipeline	2014-01-23 12:49:05.0	Complete	100.0
AutoRun	2014-01-17 13:48:13.0	Failed	100.0
Transferred	2014-01-17 13:48:12.596	Complete	
Created	2014-01-17 13:48:12.596	Complete	By: admin

Assessments

Experiment	ID	Project	Date
KQIF	qmroctxnat_E00015	Test	

Scan Files

- Thumbnails

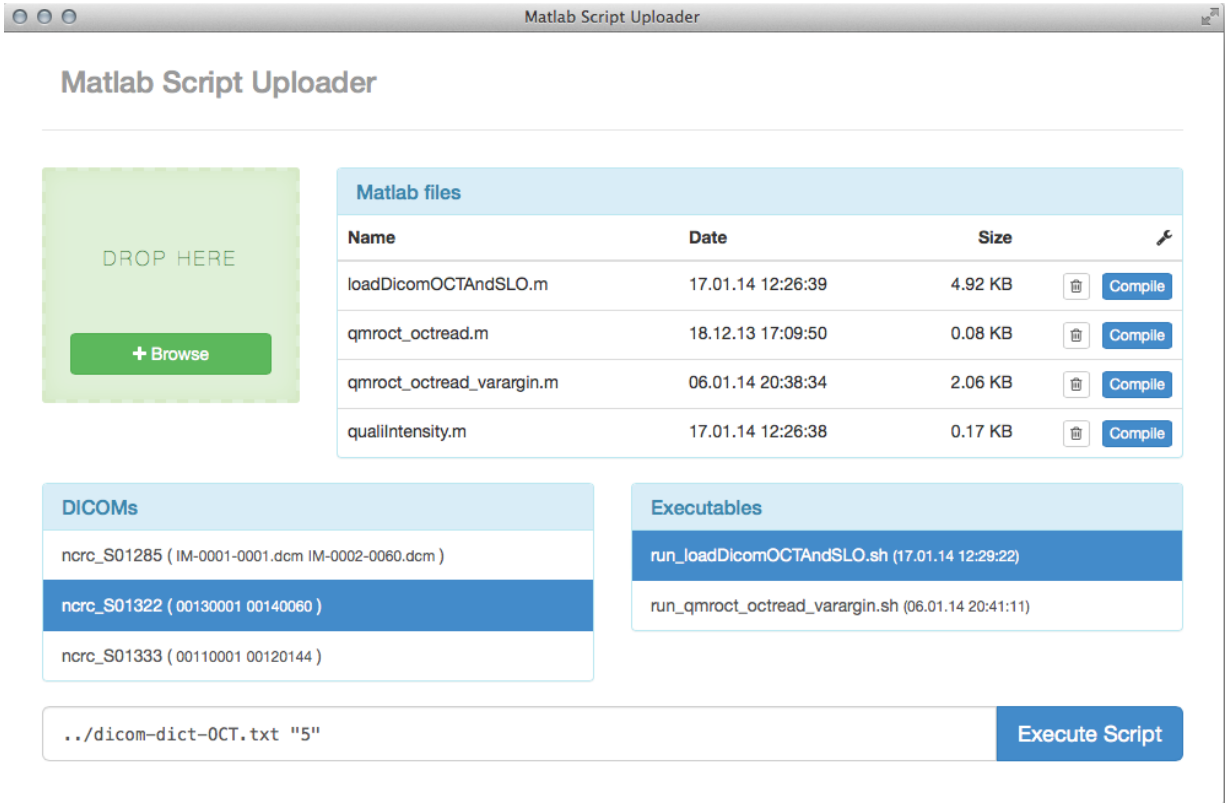
Job History

- Pipelines

Results

- KQI

Gateway - Developer Interface



Upload Matlab Files
Compile

Define CLI params
Run with test data

Deploy (Update VM)
Git - Version Control

OpenStack

Can start and stop VMs

VMs are based on snapshot which contains

- matlab algorithms + runtime environment
- test data
- execution scripts

Update Snapshot

- when new algorithm is deployed

OpenStack - Problems

Start a new VM for every job

- Resources are allocated dynamically
- Security: VM with scans deleted after use

Problems (low hardware resources)

- Starting and stopping VM takes 4 minutes each
  (Processing data only takes 30 seconds)
  → Need to re-use VMs
- No error handling yet

Current Use Case

When a new scan session is uploaded

- start pipeline manually (could be automated with XNAT)
- VM processes all available algorithms for these scans (using a shell script)
- VM sends results for every algorithm to XNAT

Todo at AMC

Enabling new use cases with **workflow management**

- execute one algorithm per scan session as one job
- execute new algorithm with all available sessions

Tools

- **WS-PGRADE** with **gUSE**

Hopefully better control of OpenStack resources

c.jansen@student.htw-berlin.de