HPCaaS with DEMO

Ákos Kovács

Széchenyi István University (SZE)

Joint work with Z. Horváth (Head), B. Liszkai, Á. Kovács, T. Budai, Cs. Tóth (SZE)

ENCCS Workshop

Agenda

- Two Main part
- 1. Simulating
- 2. Explanation

First let’s do some simulation ☺

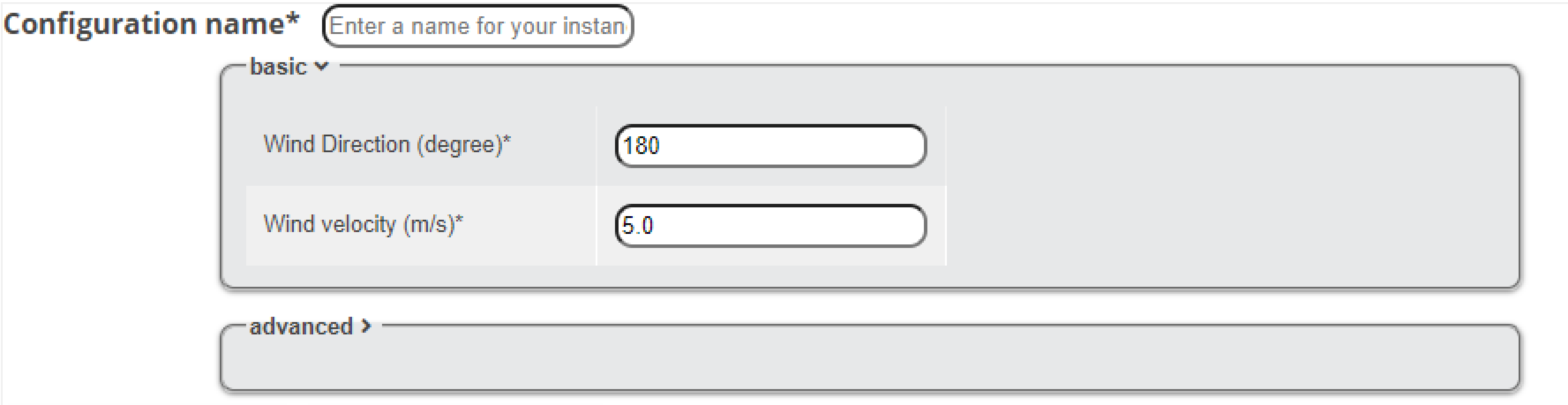
- Download the blueprint
- **http://sophora-192.man.poznan.pl/blueprints/**
- **Download uap-v2.1-enccs.yaml -> save as**
- Or just copy the content of it

Uploading the Simulation

- Go to **https://webint.hidalgo-project.eu**
- Login with your user/pass (previously sent)
- Go to Settings/Manage Application
- Add a name to your application
- Select the file, and hit upload!

Configuring the Simulation

- Go to Experiments/Manage Instances
- Select your Application (uploaded)
- Add a name and save



Configuration name* Enter a name for your instan

basic ⌄

Wind Direction (degree)*	180
Wind velocity (m/s)*	5.0

Starting the simulation

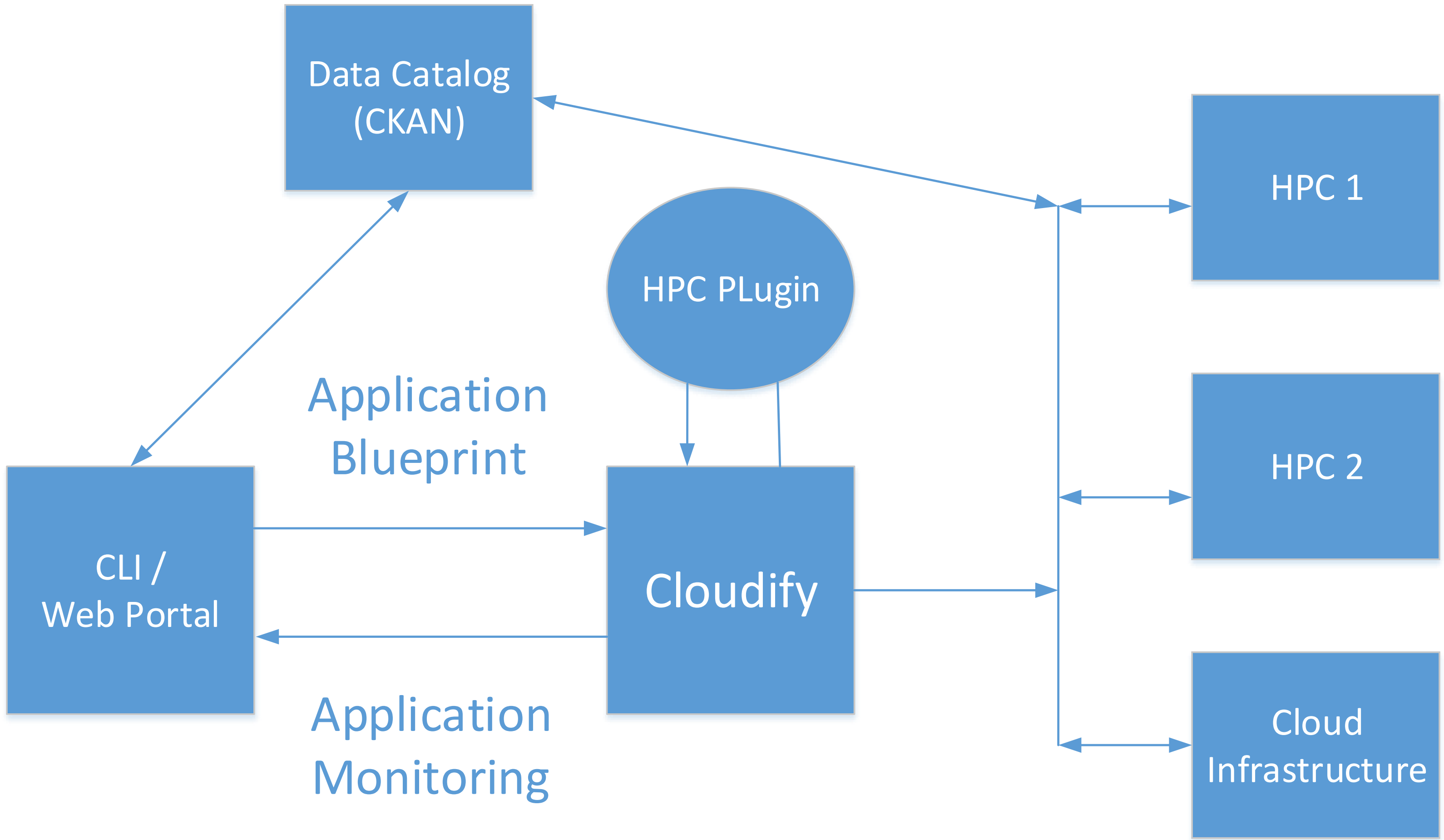
- Go to Experiments/Execute
- Choose **your** application
- Choose **your** configuration
- Choose **sophora-192.man.poznan.pl** as HPC

Run Instance!

HPCaaS

- Goal is to reach more people with HPC application
- HPCaaS trough API, means it is possible to create web portal or CLI tools to manage HPC jobs/applications
- Cloudify with HPC plugin

HPCaaS High level Architeture



TOSCA file

- Topology and Orchestration Specification for Cloud Applications (TOSCA)
- „The idea behind the TOSCA standard is to render improvements in the deployment, termination, and any other management function of cloud applications."
- Describing workflows and input data, and a relation between them.

Application Blueprint

- YAML format (Human readable, less writable ☺)
  - Strict whitespace rules
- Defining application inputs, files, datasets
  - Files to download
  - Online File edit
  - Lists
  - Strings
  - Checkbox (Boolean)
  - Number (int,float)

Application Blueprint II.

- Defining workflow
  - Add executabe commands, with input files/string defined
  - Add possibility to use Module system for HPC (Loading modules)
  - Define simulation steps and add dependencies between them, so all steps can be serialized or parallelized

HPC Applications – Singularity Container

- Each HPC has it’s own ecosystem
- Different modules
- Different Operating system
- Different software versions:
  - Our code requires python 3.8, but the HPC has only 3.7 module
- You have to install your own dependecies -> time consuming

HPC Applications – Singularity Container

- Solution -> Containerize your application(s)
- Our singularity container contains:
  - OpenFOAM v2006
  - SUMO v1.6
  - Own tools based on JAVA
  - Own tools based on python
    - With bunch of python modules installed (ckanapi, numpy, scipy, etc.)
  - ECMWF polytope client

HPC Applications – Singularity Container

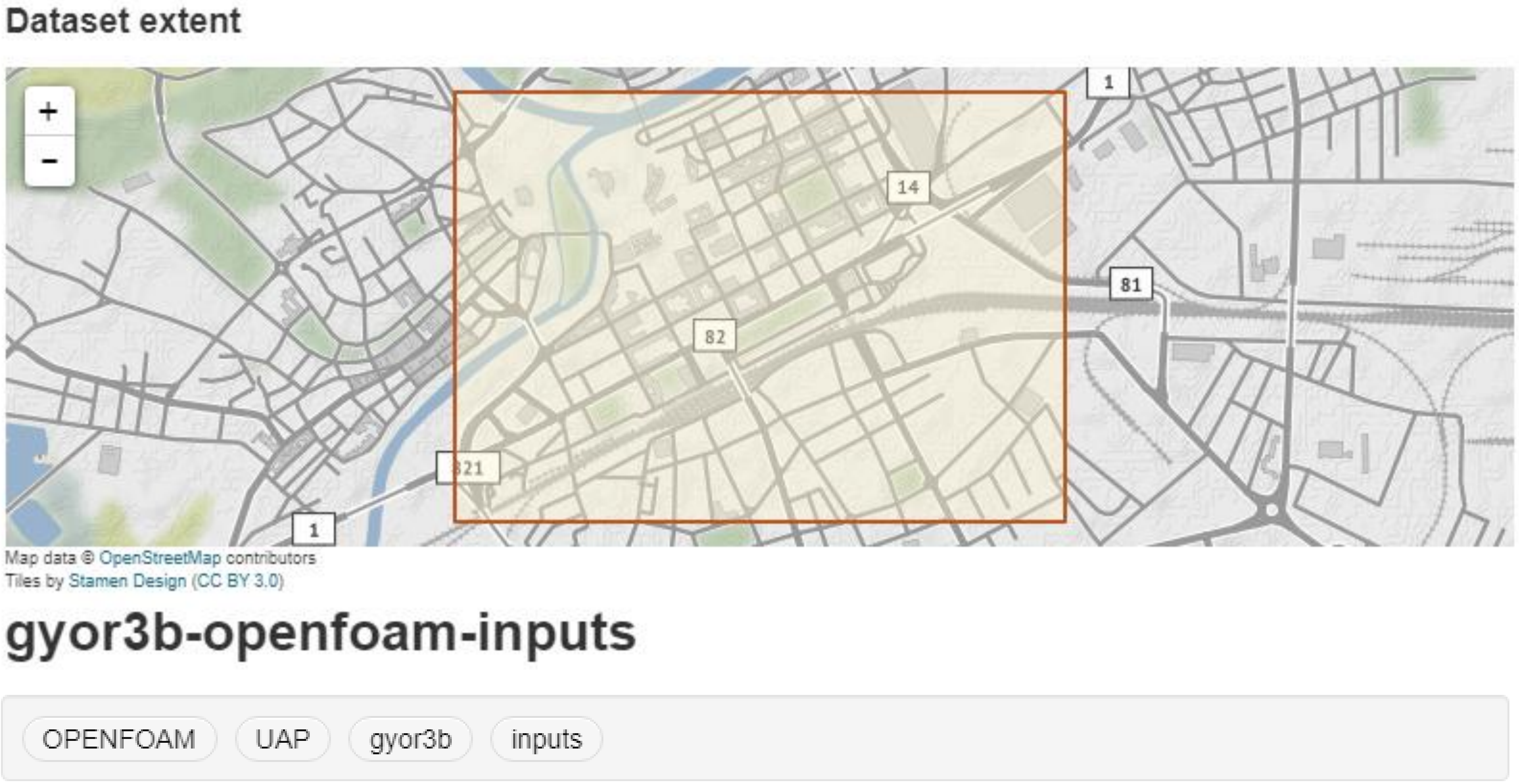
- Install all software into one ecosystem
- Only Singularity framework is needed to install system wide
- Singularity vs Docker
  - 1 image file vs fs layers
  - Host mode networking (performance)
  - Special interconnects works out-of-the-box
  - mobility

Data catalog

- Ckan data catalog
- Can be used trough API
- Store files or just the URL of the file
- Defining datasets
- Datasets, and Files can be tagged, for easy search
- Authentication with secret API key

Data catalog

- Extendable with plugins like Spatial information, and tags



Dataset extent

Map data © OpenStreetMap contributors
Tiles by Stamen Design (CC BY 3.0)

gyor3b-openfoam-inputs

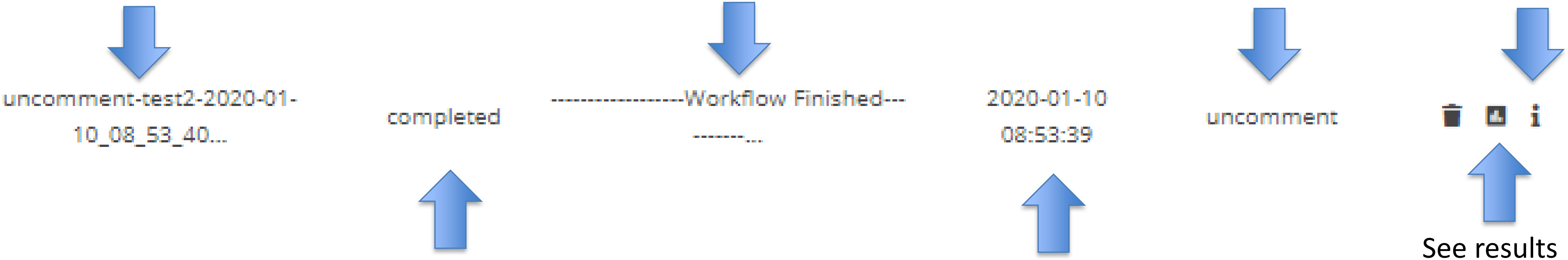
OPENFOAM UAP gyor3b inputs

Data catalog

- Full API support, you can manage CKAN trough your application
  - Dataset generation
  - User modifications
  - Upload/Download files
  - Manage permissions
- CLI client (ckanapi) or python library

Check our simulation

- Go to Experiments/Dasboard



Your simulation name

Last log message

App name

See logs

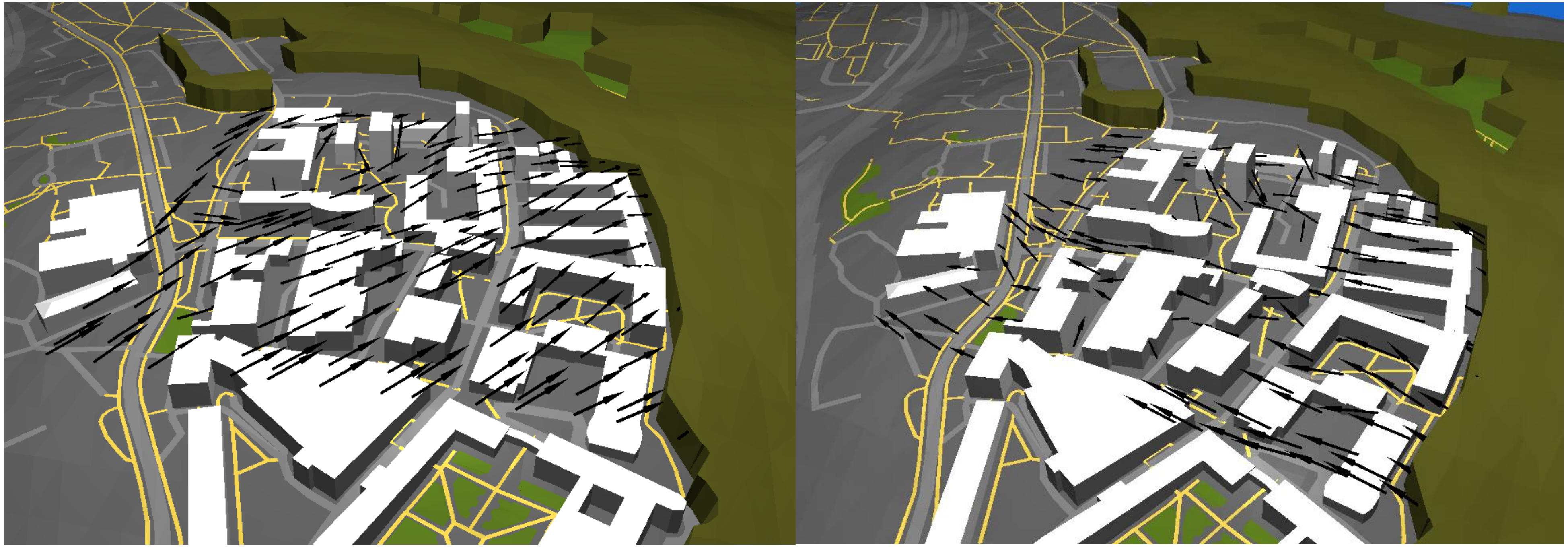
uncomment-test2-2020-01-10_08_53_40...	completed	------------------Workflow Finished----------...	2020-01-10 08:53:39	uncomment	🗑 ▣ i

Status

Starting time

See results (3D viewer)

Results with two different wind profile



THANK YOU !

QUESTIONS ?

Ákos Kovács
Széchenyi István University
Egyetem tér 1.
9026 Győr, Hungary
Email: kovacs.akos@sze.hu