Extending desktop applications with cloud computing

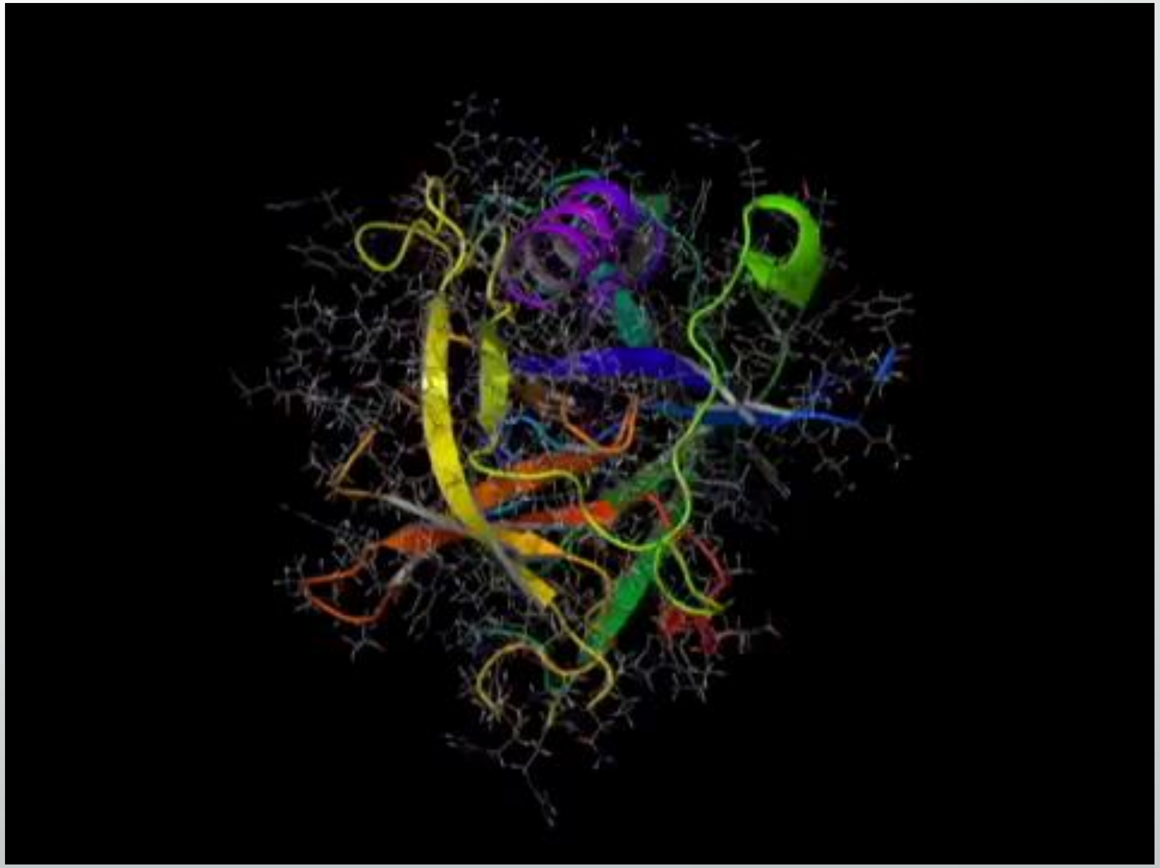
Damjan Temelkovski

damjan.temelkovski@my.westminster.ac.uk

9th International Workshop on Science Gateways
20 June 2017

Background

- Molecular docking is a computer simulation that predicts the interactions between 2 molecules, a *receptor* and a *ligand*
- Docking a large number of ligands to 1 receptor is known as *virtual screening*, a computationally intensive method used in drug discovery to find drug candidates



Video by Noeris Salam

Experiences of VS at UoW

- TV (*Trichomonas Vaginalis*) is a protozoan parasite that causes *trichomoniasis* – a sexually transmitted infection affecting an estimated 160 million people annually

- Currently, 1 treatment for *trichomoniasis* exists - if TV becomes resistant we would not be equipped to fight it

- Biomedical scientists at UoW are looking for a drug candidate by docking hundreds of thousands small molecules (ligands) to a protein of TV (receptor)

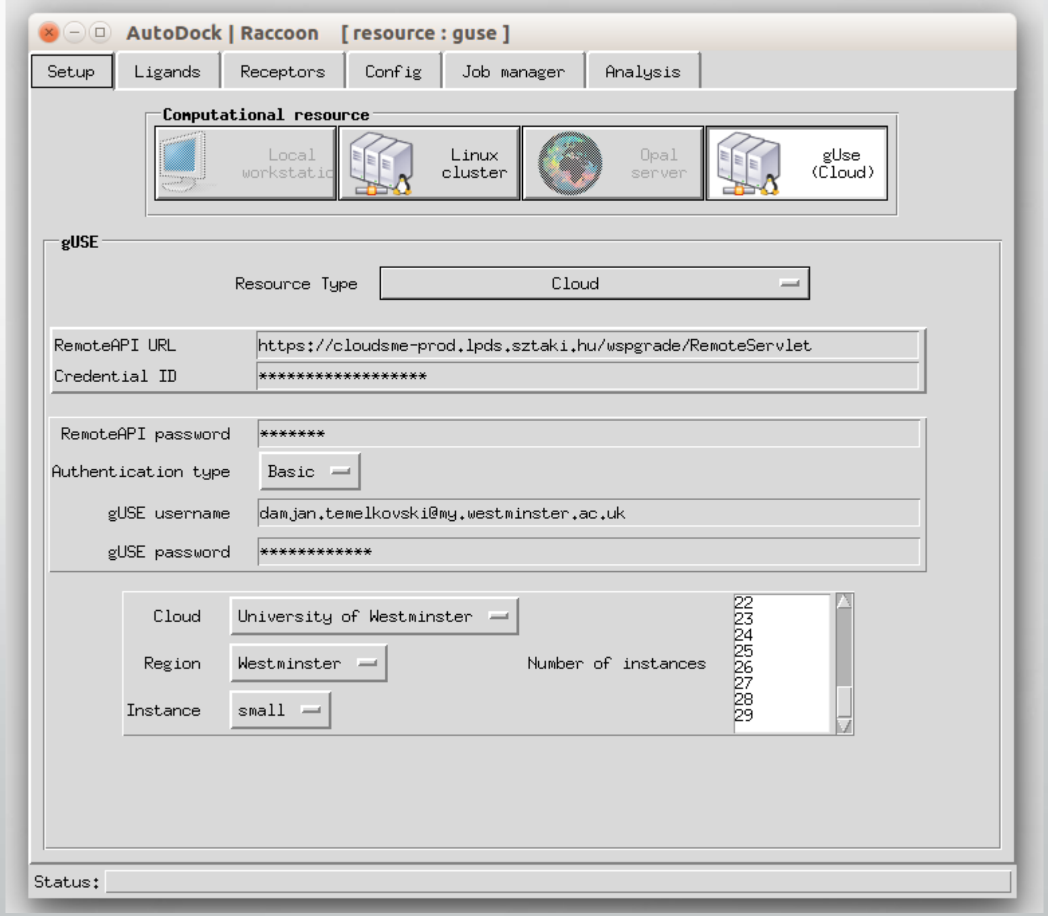
Limitations of existing tools

- WS-PGRADE Portal connected to UoW Desktop Grid
    1. Not intuitive for biomedical scientists
    2. Too restricted (you have to know the exact input files and attach them)

- Biomedical scientist used the desktop application “Raccoon” to run tiny VS simulations on their own computers

Raccoon and Raccoon2

- Raccoon2 is a graphical interface for preparing, executing and analysing AutoDock Vina virtual screenings on a PBS/SGE cluster.
- http://autodock.scripps.edu/resources/raccoon2



Resource Type: Cloud

RemoteAPI URL: https://cloudsme-prod.lpds.sztaki.hu/wspgrade/RemoteServlet

Credential ID: ******************

RemoteAPI password: *******

Authentication type: Basic

gUSE username: damjan.temelkovski@my.westminster.ac.uk

gUSE password: ************

Cloud: University of Westminster

Region: Westminster

Instance: small

Number of instances: 22 23 24 25 26 27 28 29

Research gaps

- Virtual screening requires distributed computing infrastructures (DCIs)
- Virtual screening simulations rarely use cloud computing

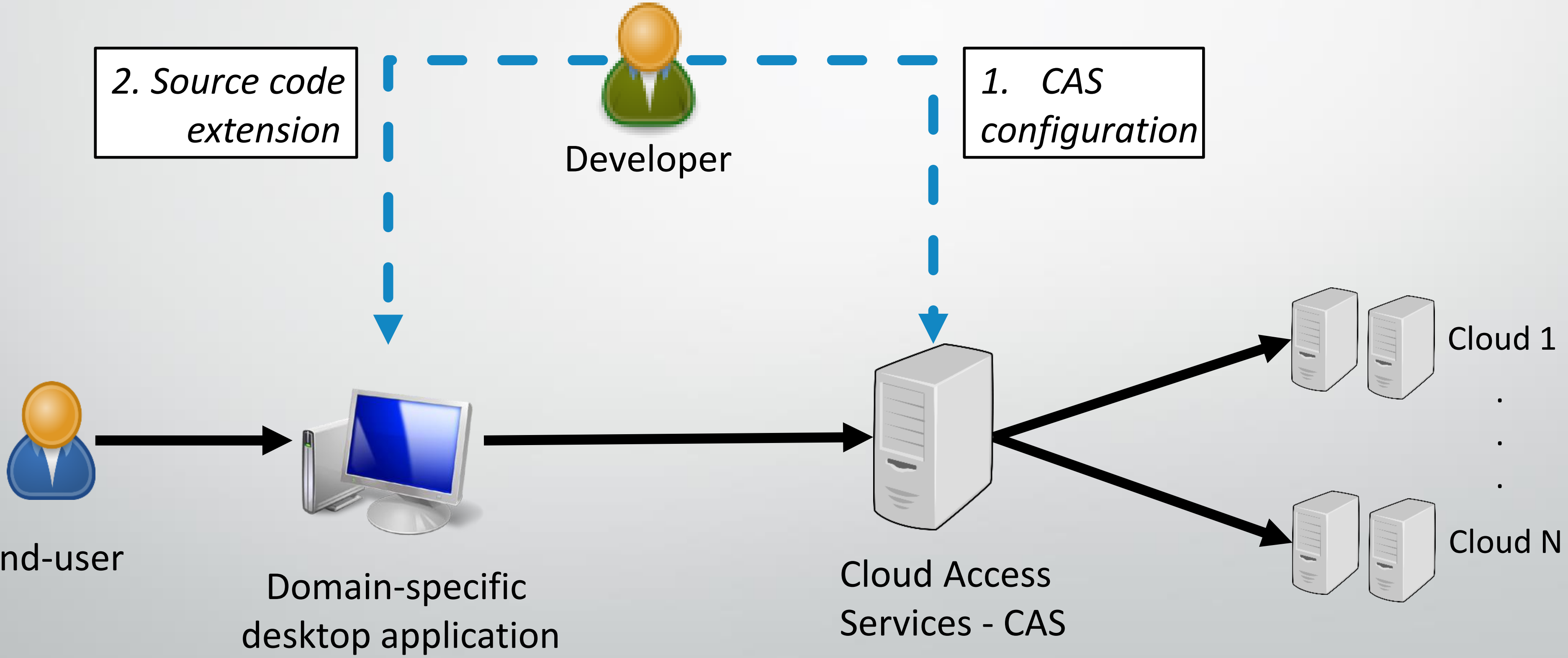
- Domain scientists still run simulations on user-friendly desktop applications
- These desktop applications usually don’t use cloud computing

Our approach

1. Configure Cloud Access Services (CAS) to run the simulation on clouds
2. Alter the source code of the desktop application
    - Insert a code segment that communicates with CAS

Retain the same familiar GUI which domain scientists are used to

Generic concept



2. Source code extension

Developer

1. CAS configuration

End-user

Domain-specific desktop application

Cloud Access Services - CAS

Cloud 1

.
.
.

Cloud N

Our implementation

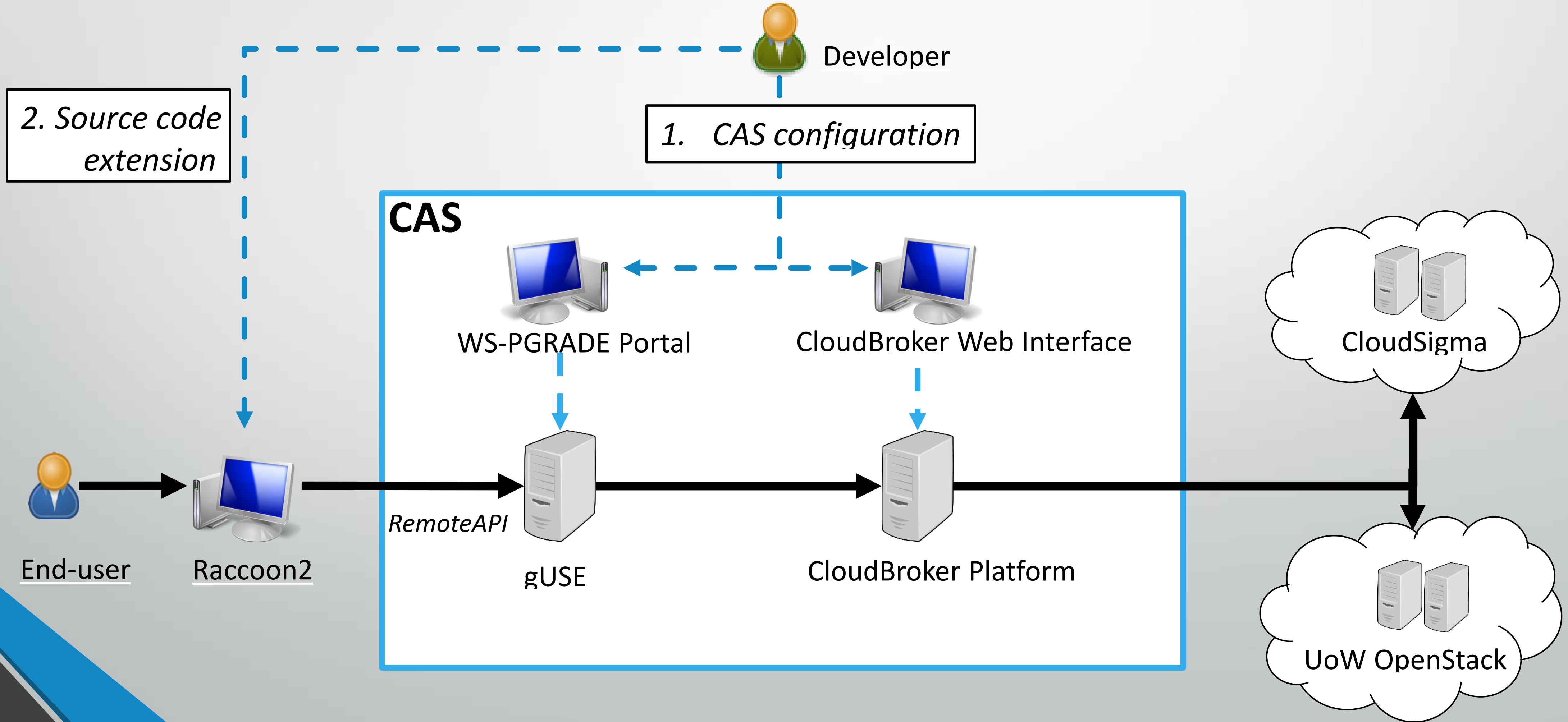
1. CAS configuration

- The CAS consist of
  - WS-PGRADE/gUSE science gateway with the RemoteAPI
  - CloudBroker Platform

2. Source code extension

- Raccoon2 has been written in Python
  - 1 new class that communicates with the CAS via http to the gUSE RemoteAPI

Diagram of our implementation



Details of the CAS configuration

1. Configure the gUSE (create the WS-PGRADE workflow)
    - Create the workflow in a WS-PGRADE portal, test it with test input data, and export it
    - Configure the exported workflow in code and attach it to the RemoteAPI ‘submit’ call
2. Configure the CloudBroker platform
    - Deploy the executable files that are needed to run the workflow on a cloud

Details of the source code extension

- Submit workflow
  - The GUI asks users to specify cloud configuration information and saves them to “workflow.xml”
- Check status
  - Provide status report every 20 s
- Download results
  - The results can be used by the analysis tab of the original Raccoon2 GUI

Results

Virtual screening using real-life input data obtained from biomedical scientists: the protein *ribokinase* of TV, and 130 216 drug-like small molecules

1. Proof-of-concept
    - UoW Cloud (64-bit)
    - CloudSigma Cloud (32-bit)
    - CloudSigma Cloud (64-bit)

2. Scalability tests on the UoW Cloud
    - 7 small instances, 7 medium instances, 7 large instances
    - 7 small instances, 14 small instances, 28 small instances

Diagram of results of proof of concept

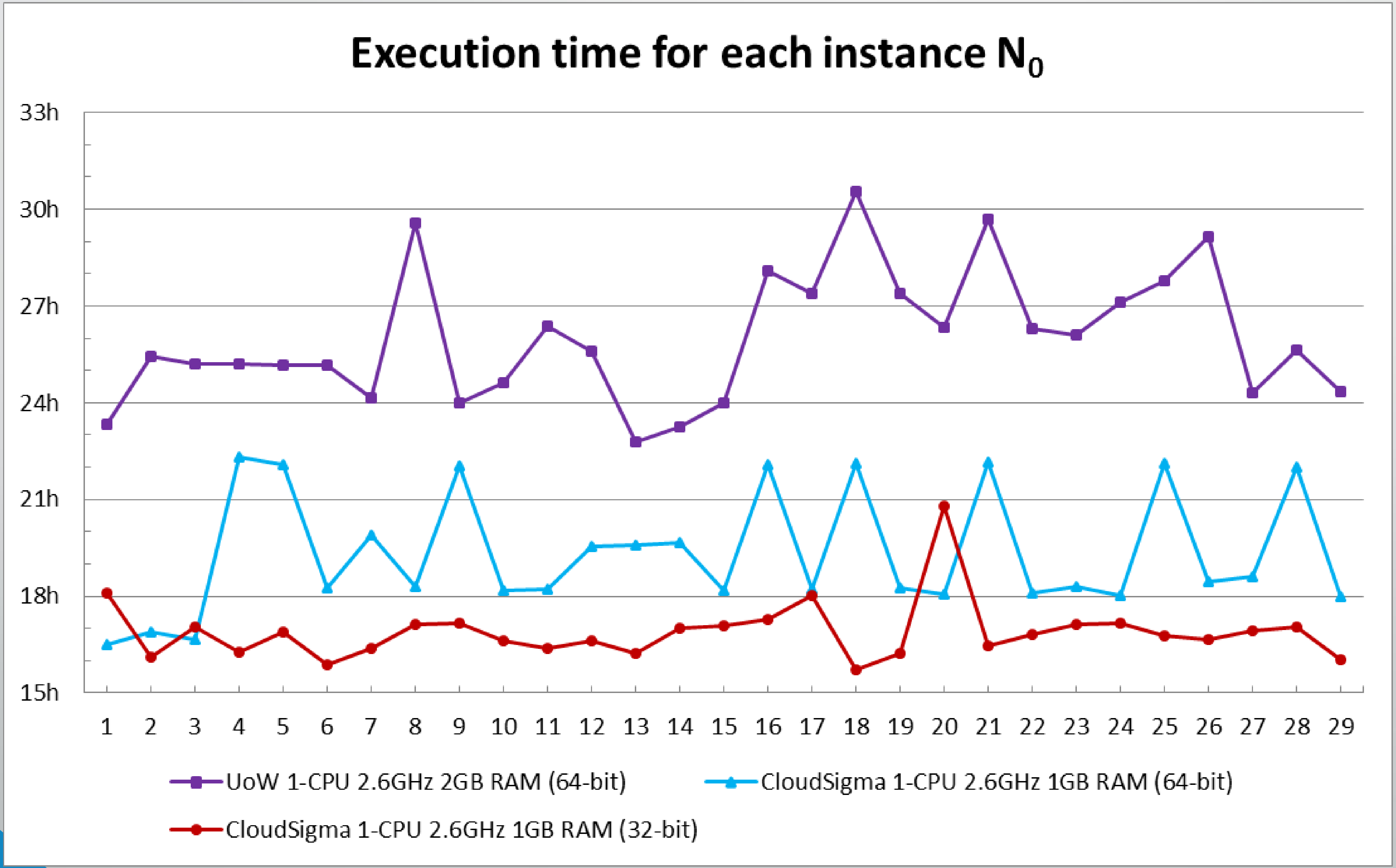
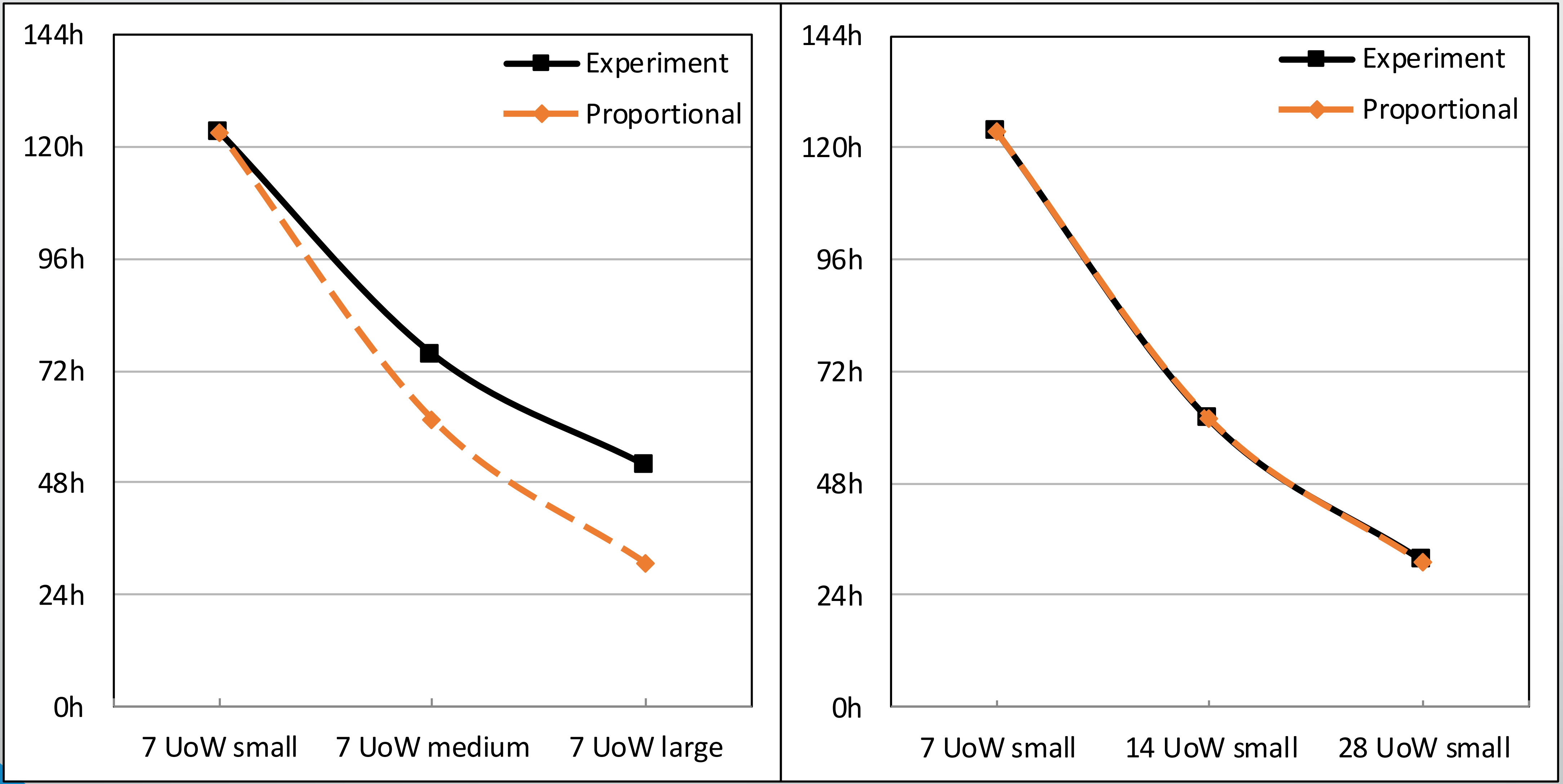


Diagram of results of scalability tests



Conclusion and future work

- Performance test results for virtual screening with AutoDock Vina
  - Using 32-bit virtual machines is faster than 64-bit
  - Using many small virtual machines is faster than using fewer large
- Biomedical scientists no longer need access to a cluster - virtual screening is more accessible for biomedical scientists around the world
- In general, domain scientists can use this approach to make desktop applications cloud-enabled

- Future work: ways for biomedical scientists to store docking results in a repository to share and analyse each other's results

Acknowledgements

European Commission Projects

- **CloudSME (Cloud-Based Simulation platform for Manufacturing and Engineering) Project No. 608886**
- **COLA (Cloud Orchestration at the level of Applications) Project No. 731574**

- UoW Centre for Parallel Computing (CPC)
    - Tamas Kiss
    - Gabor Terstyanszky
    - Pamela Greenwell
    - Hans Heindl
    - Gregoire Gesmier
    - Juha Hemminki
    - Noam Weingarten
    - Hannu Visti
    - Junaid Arshad
- SZTAKI
    - Zoltan Farkas
    - Peter Kacsuk
- CloudBroker
    - Andrey Sereda