

VO-CLOUD – Cloud Based Astronomical Computing Environment

(Running KOREL)

Petr Škoda

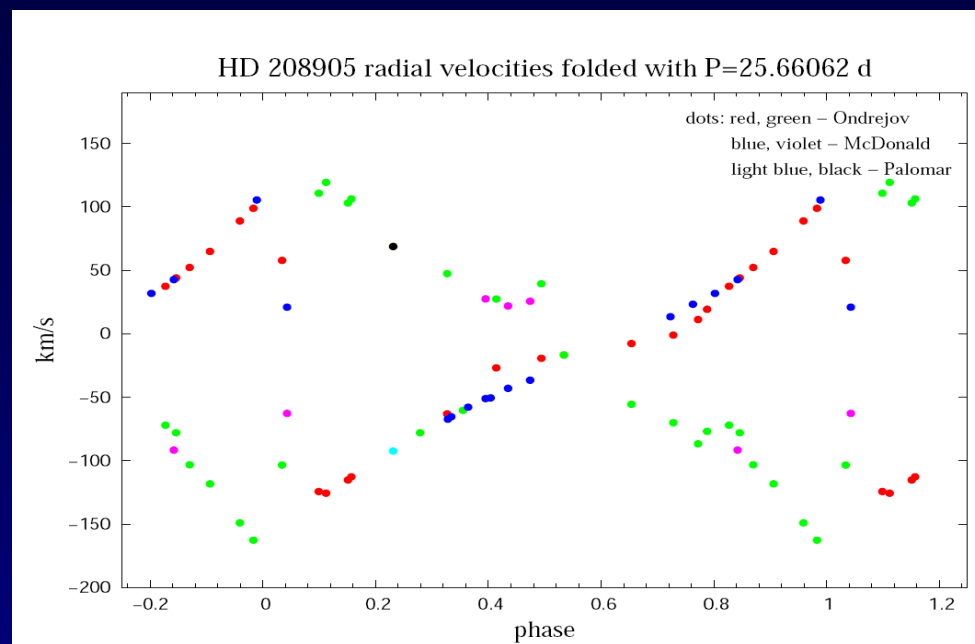
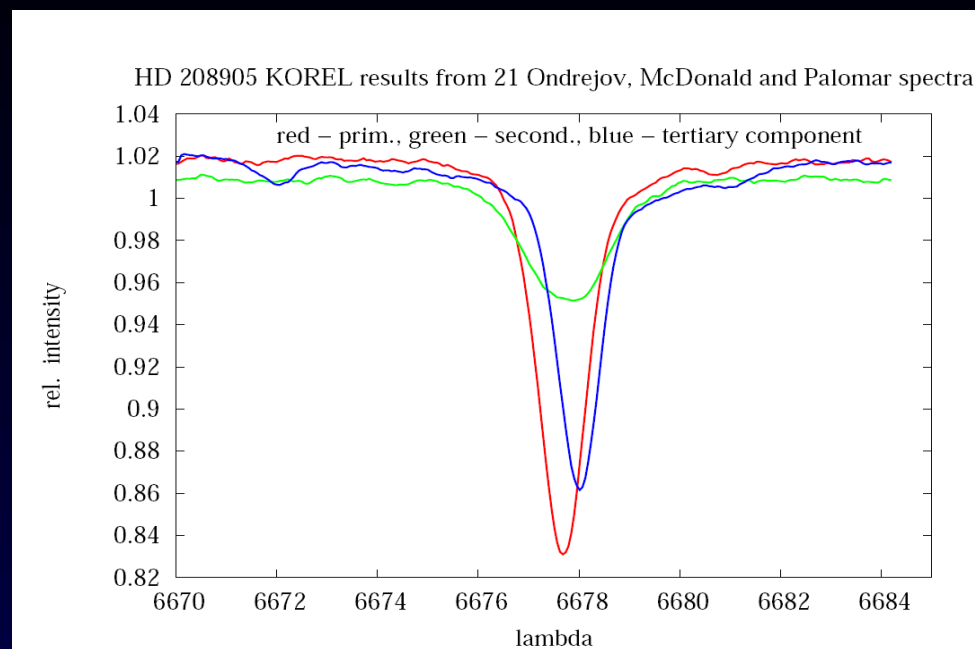
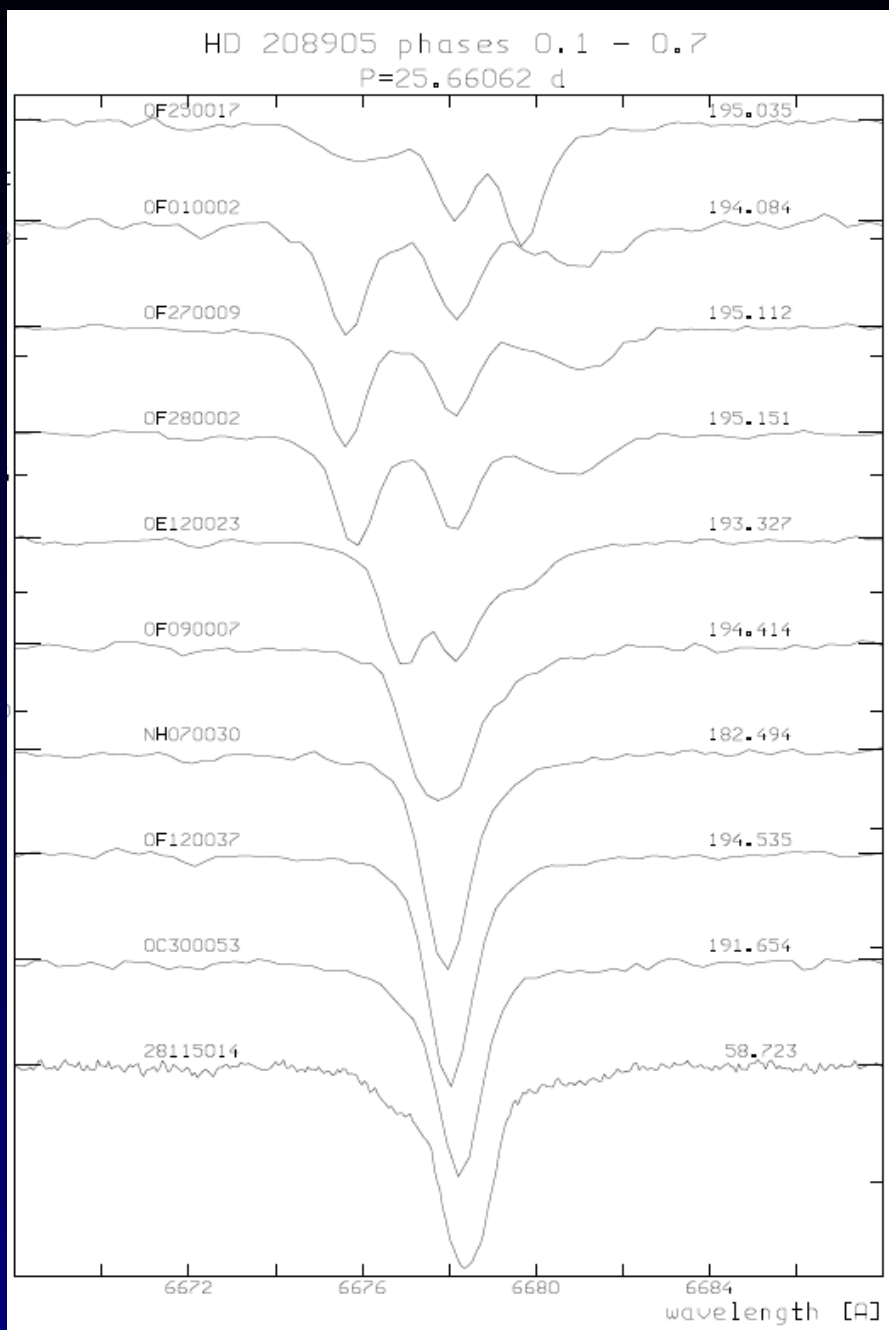
Astronomical Institute Academy of Sciences Ondřejov
Czech Republic

Lumír Mrkva

Faculty of Informatics, Czech Technical University, Prague

IVOA Interoperability meeting GWS session GWS2
NCSA, Champaign-Urbana, USA, 24th May 2012

Spectra Disentangling in Fourier Space - KOREL



KOREL Web Portal

The only one place under full control of author

The disentangling as SERVICE

Provide program and HW (like e.g. firewalls, mail filters)

Official tested version (approved by author)

Optimized for given HW (check of limits time, size)

Scalable – load balance, parallel, Grid

User registration (authorization by author, science case)

Shopping cart model, database - “E-shop” for service..

Integration of graphics, easy forms, links, GUI =Web

KOREL program

FORTRAN77

2 +1 input files (data, params – flags, + opt. template)

Output is variable (depends on input keys IPRINT=1...)

Created graphics +results texts - linked

(text->image by GnuPlot, PS by program)

Parameters needs to play with (many jobs aborted soon)

99% parameters re-usable → restart job

Concept of scientific „CLOUD“

User does all the work outside his desktop SOMEWHERE

He is asking the SERVICE and gets results

Large stable INPUT data + small changing PARAMS

Many runs on SAME data (models, image deconvol, optimiz.)

Graphics visualization from postprocessed output text files

Using WWW browser - supercomputing in PDA/mobil

CZECH TECHNICAL UNIVERSITY IN PRAGUE
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF SOFTWARE ENGINEERING



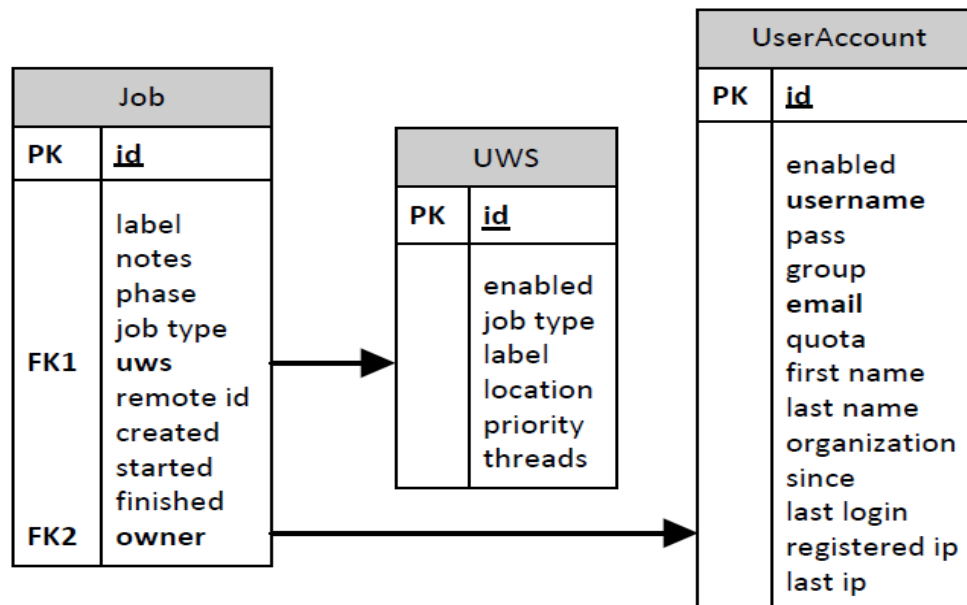
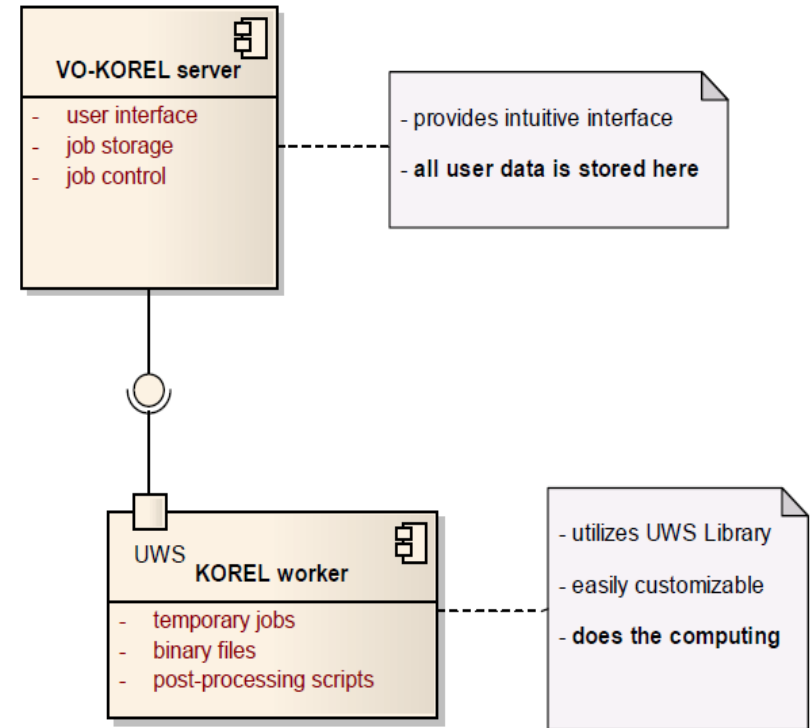
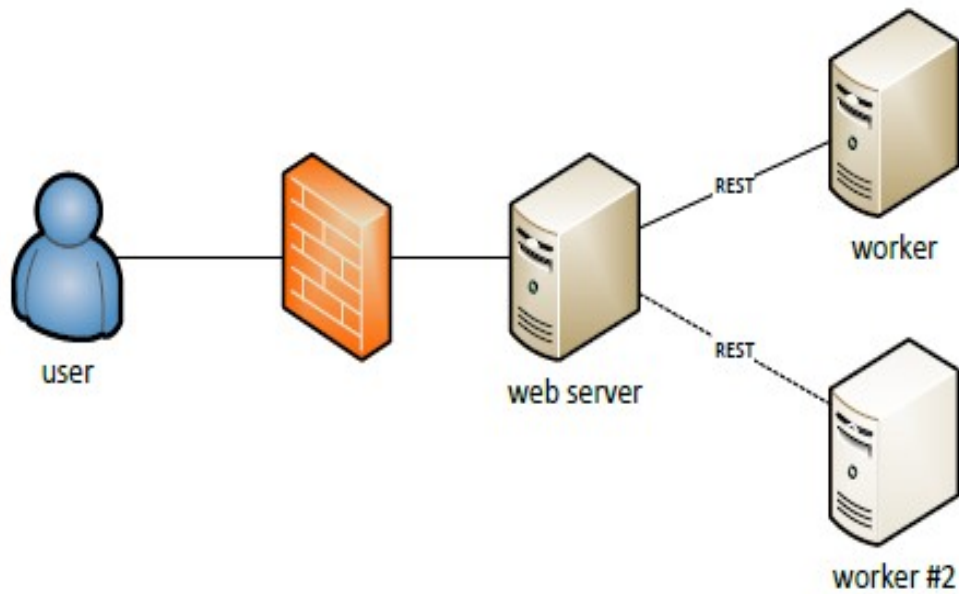
Bachelor's thesis

**VO-KOREL, server for astronomical cloud
computing**

Lumír Mrkva

Supervisor: RNDr. Petr Škoda, CSc.

18th May 2012



CREATE job

server selects the less utilized worker of same type (KOREL)

worker must contain binary of computing engine (+visualization)

server creates job on worker – gives link to *params.zip* stored on server (here are all user inputs = data +params + control info)

worker stores the link (knows how to download)

server stores the jobid obtained from worker

job is in PHASE=PENDING

RUN job

User wants to run the job (Save and run or Run again)

Server send PHASE=RUN to the worker

Worker downolads *params.zip* using the stored link

Server starts polling the PHASE of job on worker

If PHASE=COMPLETED, ABORTED, ERROR

Server downloads the results from worker

Server sends ACTION=DELETE to the worker – destroys job

ABORT job

User decides to ABORT the job

Server send PHASE=ABORT to the worker

Worker stops the computation (signal to KOREL binary)

Server downloads the results from worker

Server sends ACTION=DELETE to the worker – destroys job

SW Requirements

SERVER

PostgreSQL (user database)

Storage space for user experiments (quota)

Glassfish 3.1 or Jboss (needs application server for EJB)

WORKER

Binary of computing engine of given type (KOREL, Synspec)

Visualization scripts (convert to png, test presence of data)

JAVA7 + Servlet engine (Tomcat7, Jboss, GlassFish)

Advantages of this solution

Standard UWS library – no modifications

ABORT returns results !

Users isolation (user sees only his UWS jobs on server)

Security (hash id – direct connection to worker knows only server, encryption of transferred data possible)

Scalability (more workers added dynamically – admin UI)

Workers for different computations may be prepared (load)

User friendly GUI on server – browser on phones (AJAX)

Can be interfaced to Grid or cluster queuing systems

worker sends the source of computing engine + data

does rendering of graphics

DEMO

<http://vocloud.asu.cas.cz>