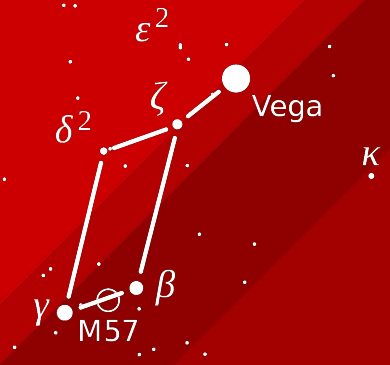
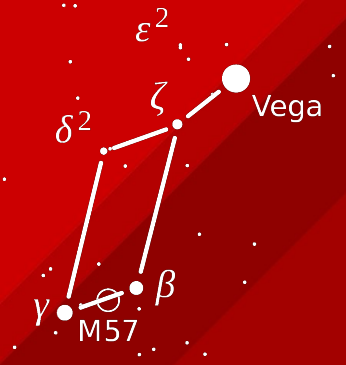


OpenShift-based High-Performance Computing for Research in Astrophysics

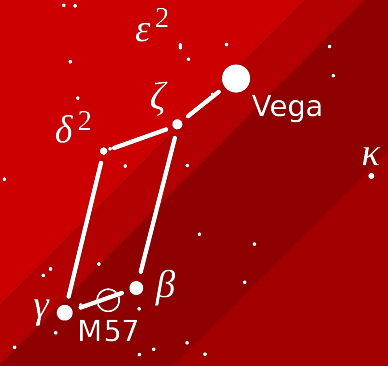
Zdeněk Švécar - Senior Project Manager
Gabriel Szász - Ph.D. Student
Filip Hubík - OpenStack Quality Engineer
Nikolaos Moraitis - OpenShift Software Engineer

January 23rd, 2020

VEGA PROJECT



VEGA PROJECT

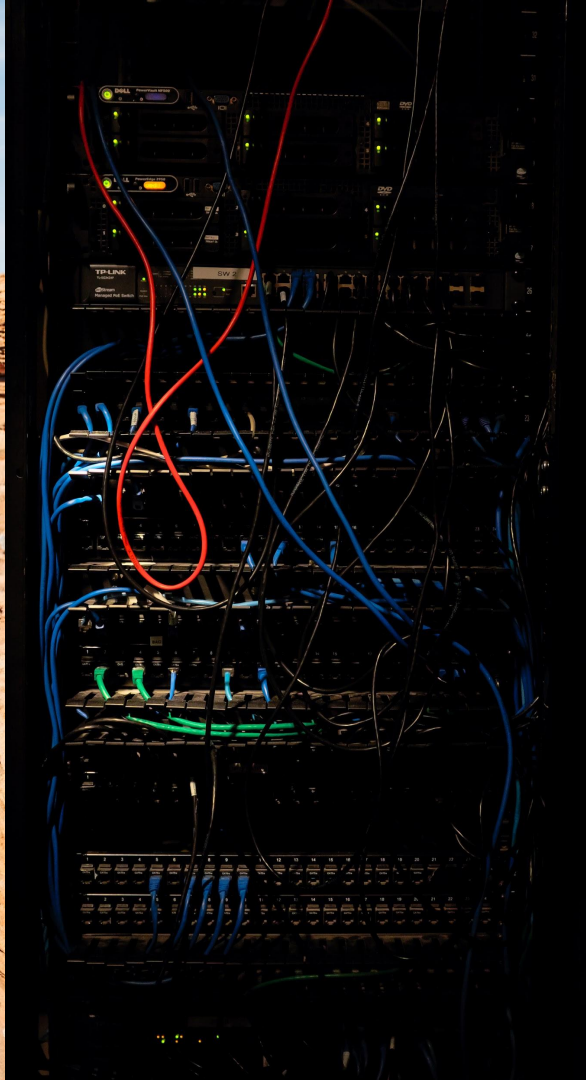


Inconvenience of quantitative spectroscopy
Software codes not ready for cloud-native environment

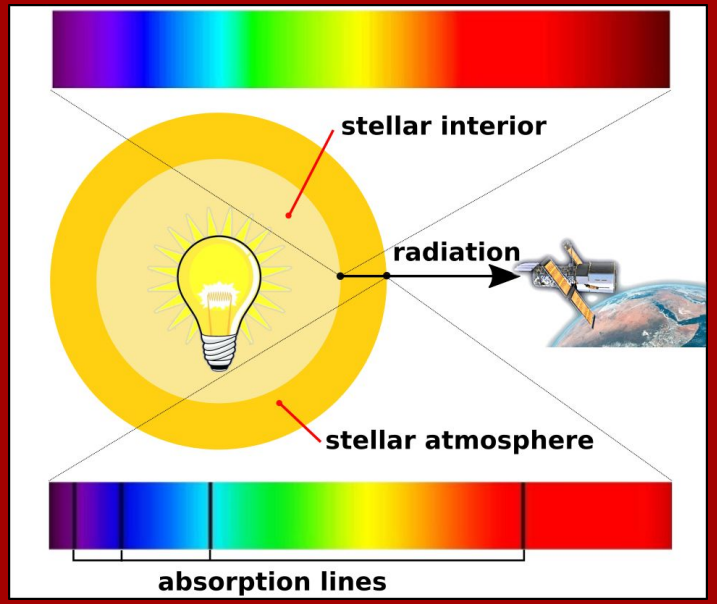
Our path to OpenShift
Compiling what is not compilable
Parallelizing what is not parallelizable

Why we use OpenShift
Description of our pipeline
Benefits of our approach

VEGA
PROJECT



Quantitative spectroscopy



Set initial parameters

Calculate model atmosphere

Calculate synthetic spectrum

Synthetic spectrum

Observed spectrum

Compare

Does it fit?

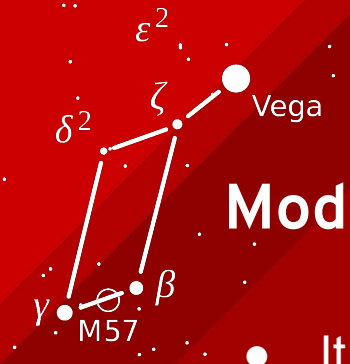
No

Tweak parameters

Yes

You've determined stellar parameters

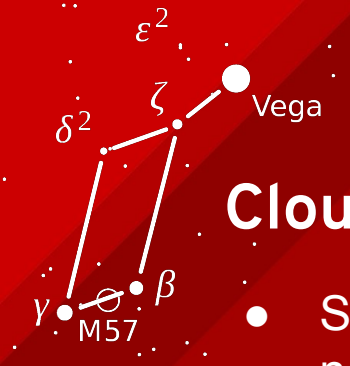
VEGA PROJECT



Model Atmosphere Grids

- It might take days to get a proper model atmosphere
- It is more convenient to **precalculate** model atmospheres for discrete grid in parameter space.
  - Quantitative spectroscopy can be then done by reusing existing grid, and interpolating/extrapolating parameters, if necessary.
- There are several dozens of existing model atmosphere grids, but none of them offers possibility of recalculation, or tweaking model parameters on-line.

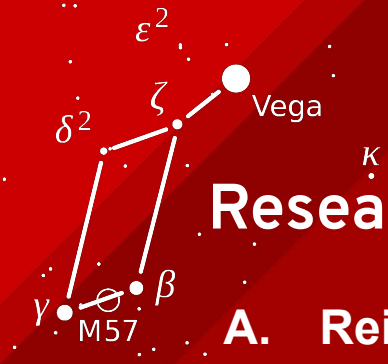
VEGA PROJECT



Cloud Native Science?

- Scientific codes are mostly more than 20-50 years old, written by non-programmers using outdated programming languages.
  - It means that:
    - They are often not parallelized.
    - They are not scalable.
    - They are tight to specific platform.
    - They offer suboptimal user experience.
    - They are poorly documented.
- Running these applications in a cloud-native environment would significantly increase efficiency of scientific work
- This would not only change the way how we are doing quantitative spectroscopy, but it would change entire fundamental science.

VEGA PROJECT



Research: how to compute a single model?

A. Reimplement from scratch the distributed way?

i. NO! tried, even early design failed
OpenWhisk, OpenMP(I), OpenStack, Chapel, ...

ii. Wouldn't be accepted by scientists anyway

B. Use old black-box tools doing who-knows-what?

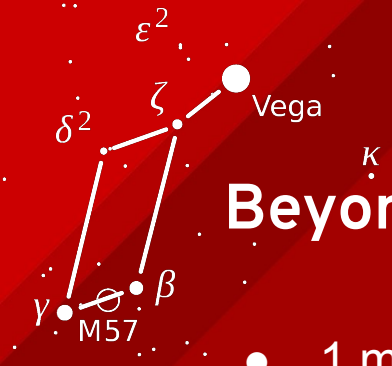
WELCOME TO HELL!

i. Atlas - Ada(paralel)/Fortran, Tlusty, Synspec - Fortran (ifort), Synthe - Fortran - ???

ii. Compiling the uncompilable, running the unrunnable, parallelizing what is not designed to run in parallel

iii. Containerization of diverse environments is a must

VEGA PROJECT



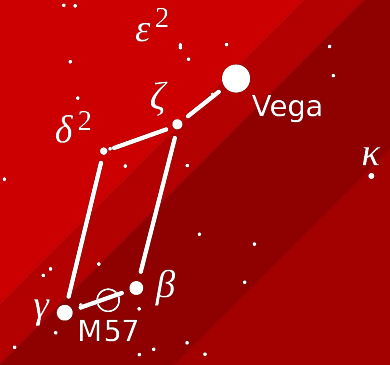
Beyond research: How to compute the matrix?

- 1 mid-grade server node vs. cloud with unlimited parallelism
  - 260 000 years vs. 1 billion USD
- Looking for a “**Middle ground**”, e.g. how to easily
  - engage public/private groups/clouds with resources and interests
  - scale the unlimited way if possible
  - be easy on scientists not being programmers but still use it

Decision: Why not just use plain OpenShift?!

- black boxes as containers, reproducibility, embedded CI, computation templates, scaling, cooperation of diverse clouds

VEGA PROJECT



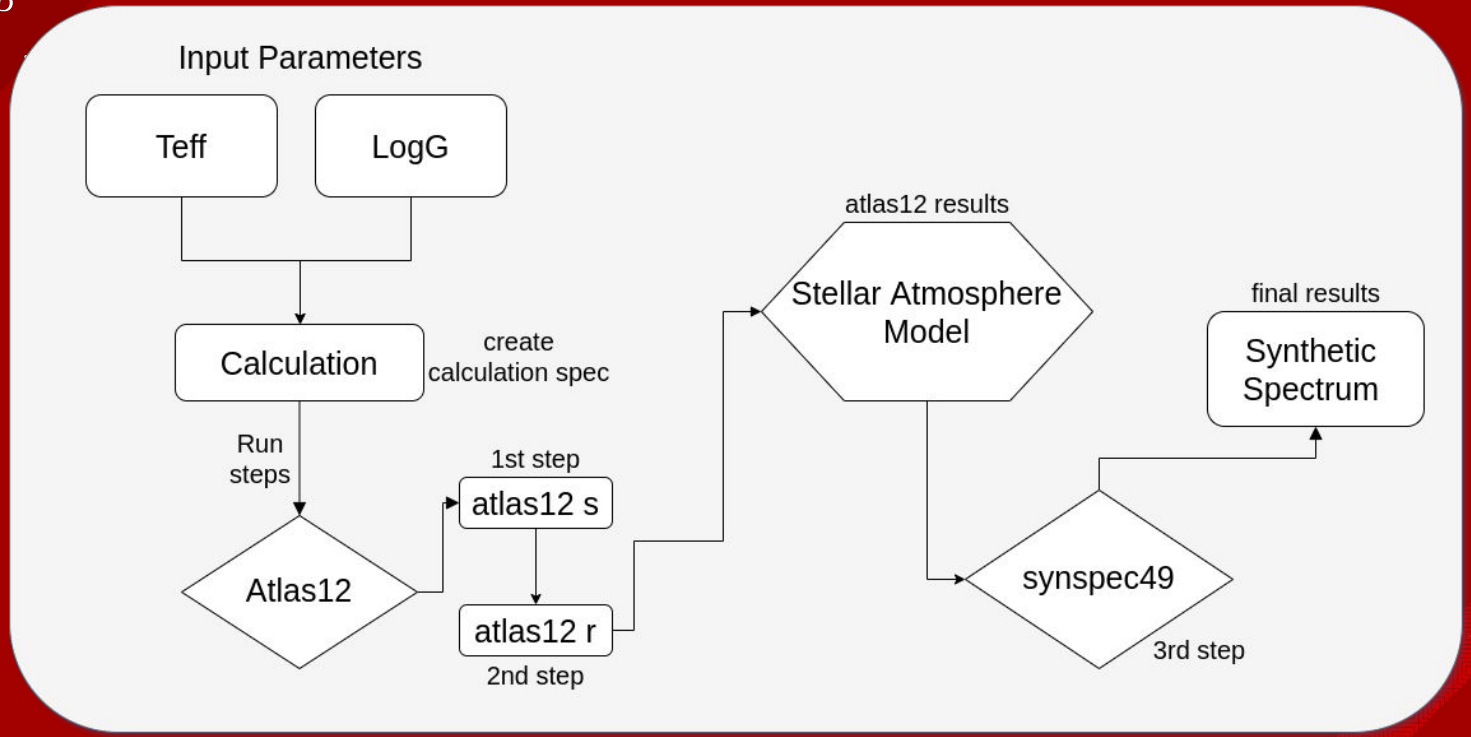
Why OpenShift?

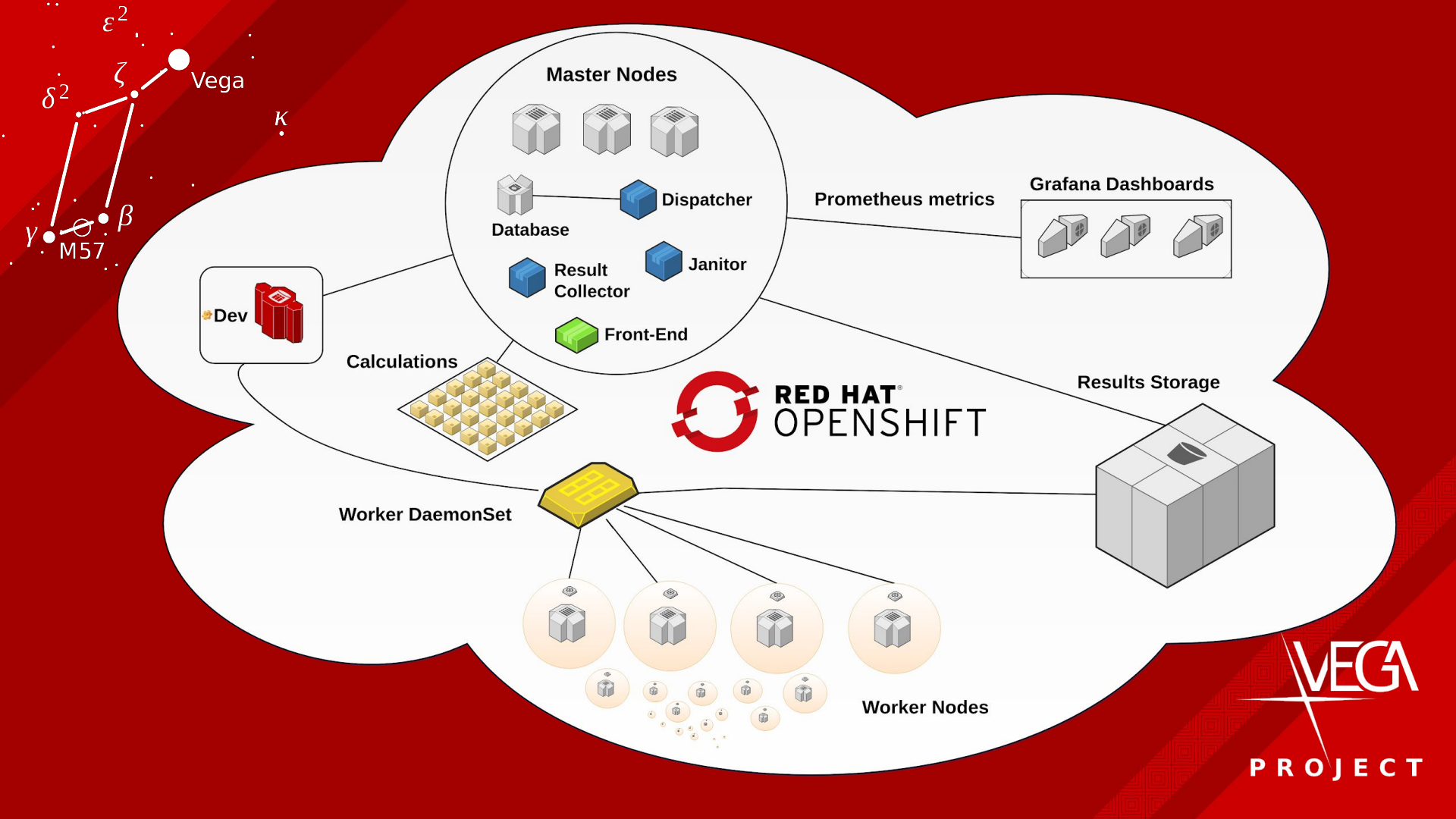
Red Hat OpenShift Container Platform

- CI/CD
- Easy developing
- Hybrid Cloud
- Easy deployment
- Built-in monitoring

VEGA PROJECT

Calculation Pipeline

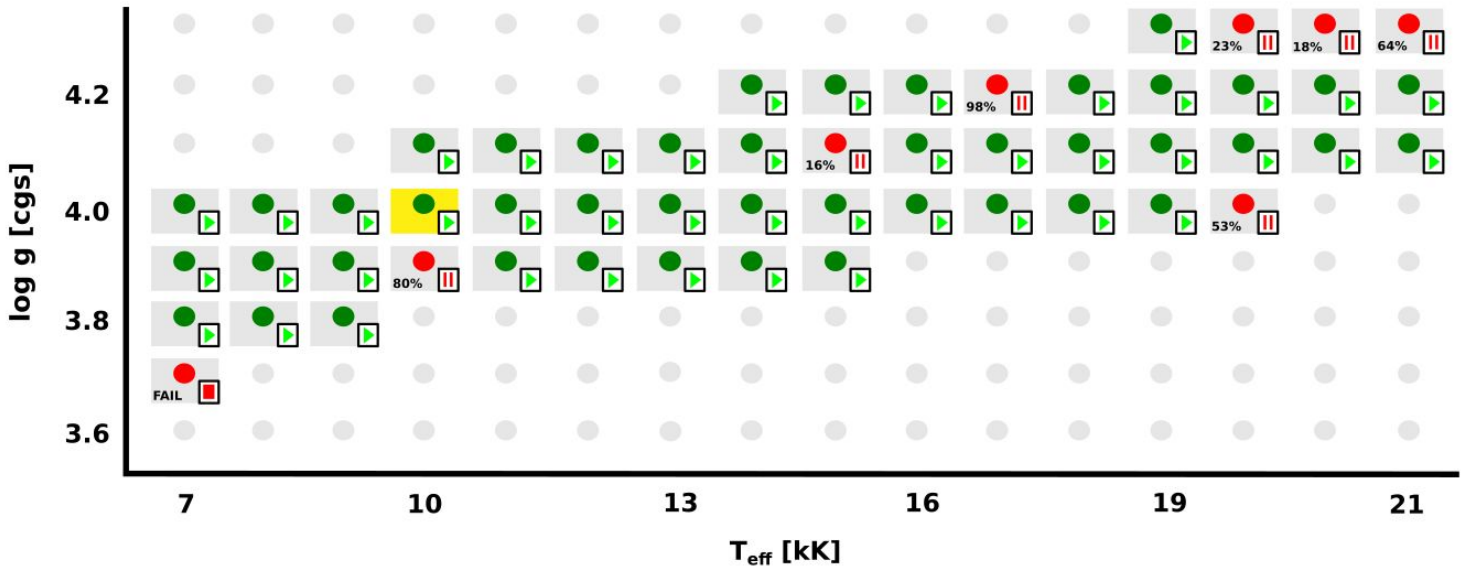



Master Nodes
Dispatcher
Database
Result Collector
Janitor
Front-End
Prometheus metrics
Grafana Dashboards
Dev
Calculations
RED HAT OPENSHIFT
Results Storage
Worker DaemonSet
Worker Nodes
Vega
M57
PROJECT

PROTMAG

Pseudo-ROTational Model Atmosphere Grid

T_{eff} / log g	T_{eff} / v_{turb}	Ω / i



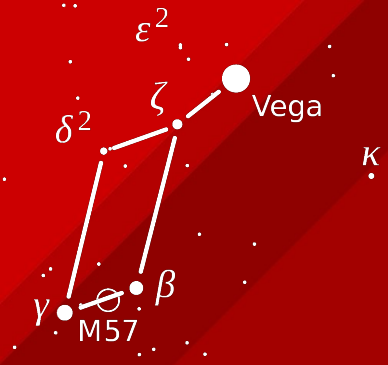
Model Information

Teff = 10000
log g = 4.00

Recalculate

Download

VEGA PROJECT



https://gitlab.physics.muni.cz/vega-project

https://research.redhat.com/projects/vega-project-openshift-based-hpc-for-fundamental-science/

VEGA PROJECT