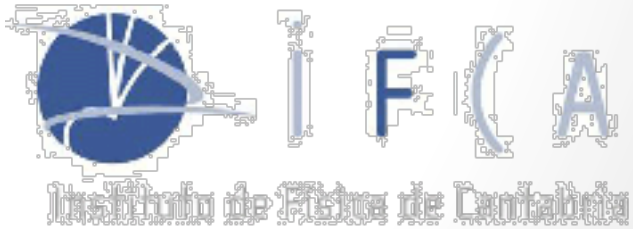
Phenomenology Tools on Cloud Infrastructures using OpenStack

I. Campos, E. Fernández, S. Heinemeyer, A. López, F.v.d. Pahlen

Instituto de Física de Cantabria (CSIC-UC), Spain



Background

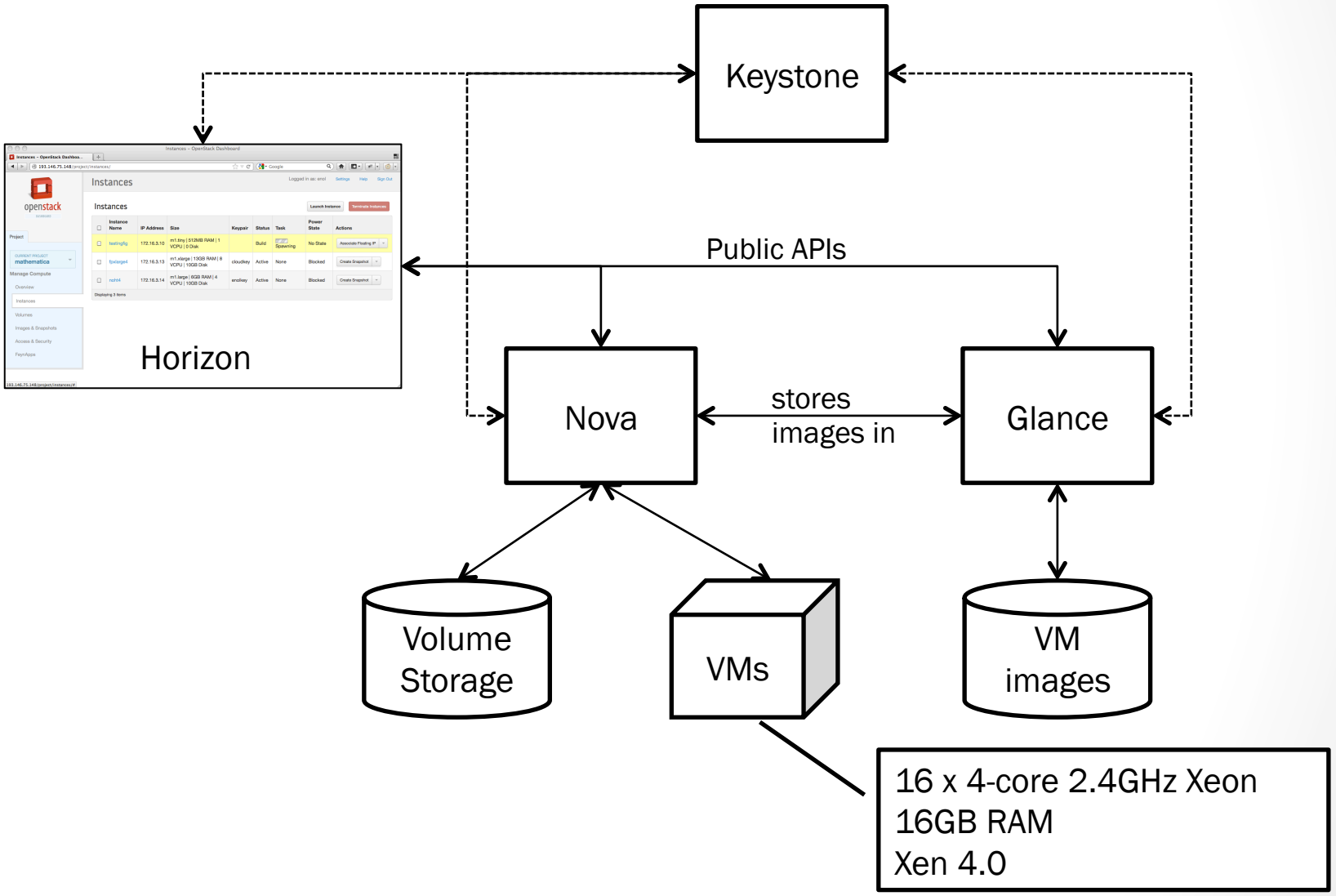
- Particle Physics Phenomenology community develop their own software packages
  - Fortran, C/C++, Mathematica, …
  - Both Legacy and new code executed in a workflow
  - Usually distributed as source

- Users need proper computing environment
  - Deal with compiler/libraries dependencies
  - Each simulation may require different software versions
  - Not always easy to install at shared facilities (e.g. grid)
  - Usually deployed at ad-hoc clusters or desktops

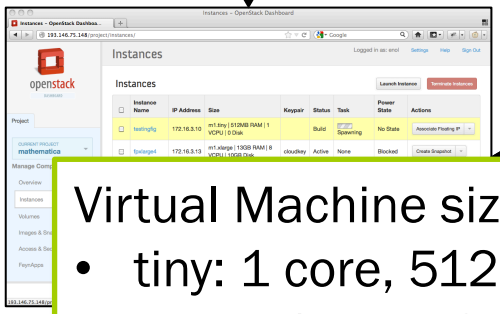
IaaS Cloud for Particle Physics Phenomenology

- Isolated and powerful computing environment
  - VMs size enough to run the software without hitches
  - No other users / processes running
  - Test and execute various configurations independently
  - Reliable (run by the sys admins of the site)
- Users free to tune their VMs to fit their software needs
  - Choose compiler/libraries
  - Licensed software (Mathematica)
- Contextualization:
  - Provide users with ready to use VMs for running their simulations

OpenStack Testbed @ IFCA

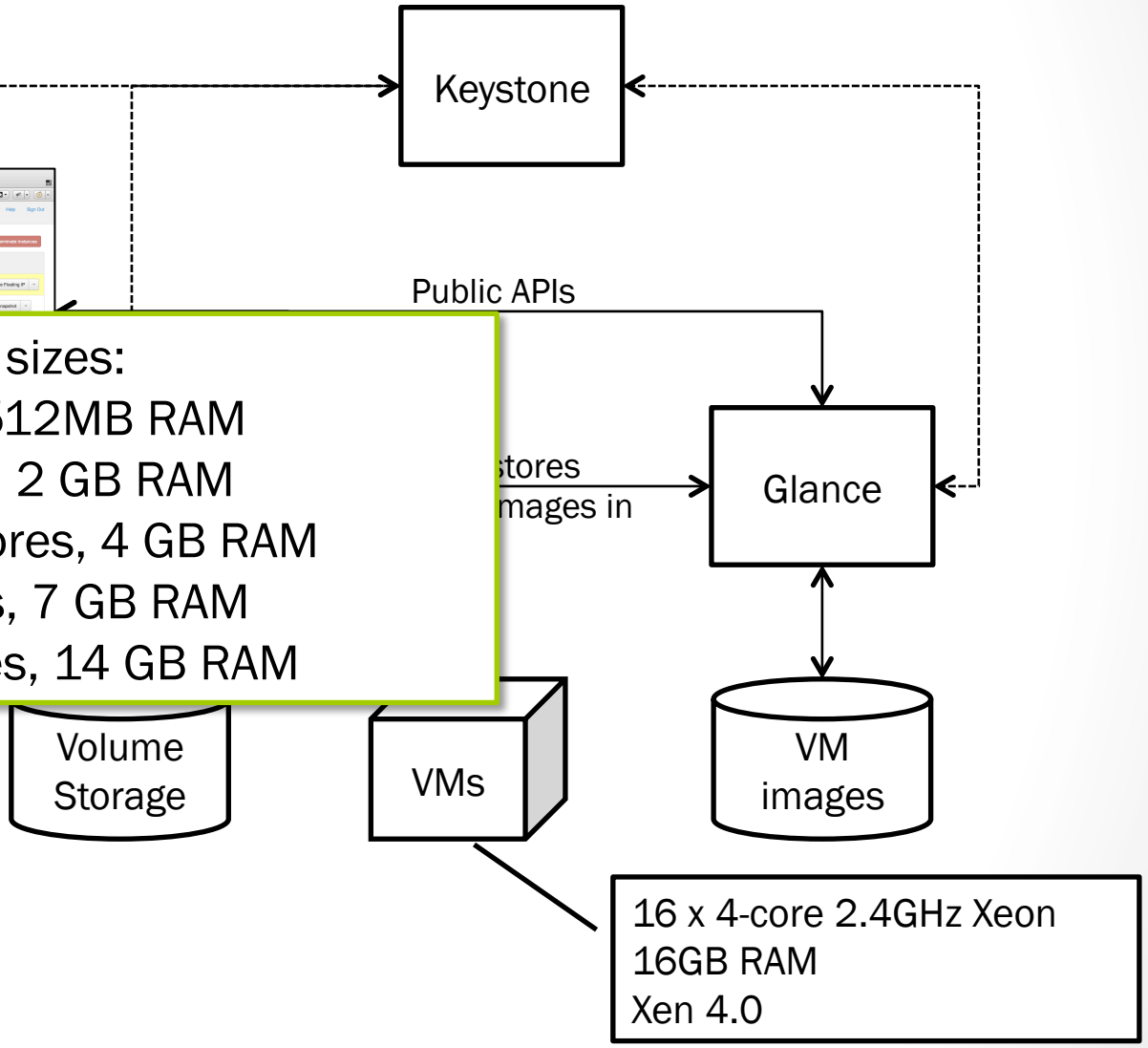


OpenStack Testbed @ IFCA



Virtual Machine sizes:

- tiny: 1 core, 512MB RAM
- small: 1 core, 2 GB RAM
- medium: 2 cores, 4 GB RAM
- large: 4 cores, 7 GB RAM
- xlarge: 8 cores, 14 GB RAM



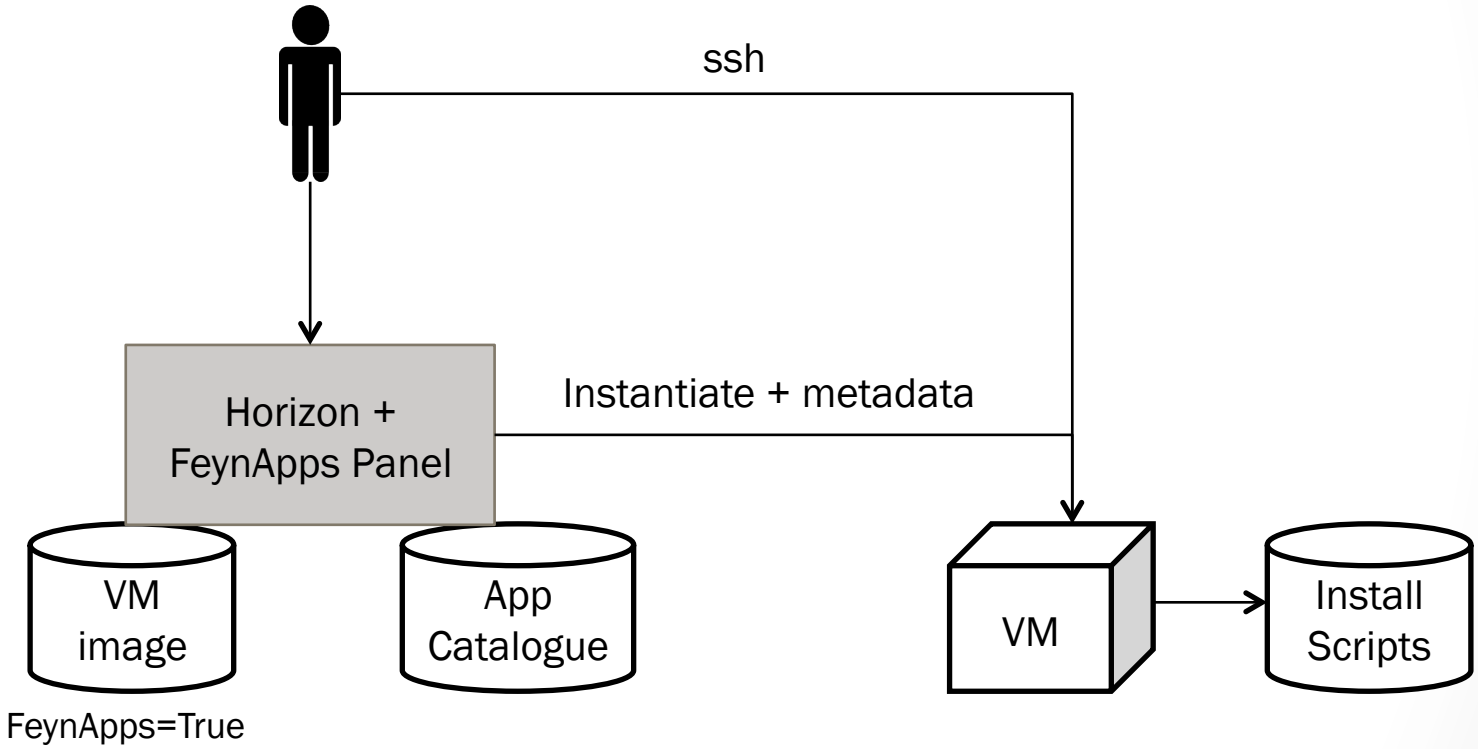
Keystone @ IFCA

- VOMS authz
  - AuthN done by Apache
  - WSGI filter maps VOMS attributes to OpenStack tenants (i.e. groups)
  - see *VOMS-aware identity service for OpenStack* presentation today at 14:00 on Cloud Platforms track
- LDAP authz
  - AuthN done by Apache
  - WSGI filter maps LDAP dns to tenants
  - Still in beta stage (very simple mapping)

Contextualization (I)

- On start-up execute a set of scripts that prepare the machine for the users
  - install software
  - sets environment for its usage
- Why not just using CERNVMFS?
  - Applications change too frequently or under development
  - Some of the apps need to be recompiled for every use-case
  - Contextualization is not limited to software installation
    - start services
    - create user accounts
    - mount filesystems
    - …

Contextualization (II)



Contextualization (III)

- Application catalogue
  - Implemented as a JSON dictionary in a github repo
  - Each application has an entry in the dict
    - Different versions available for each app
    - Dependencies between apps
- Installers
  - Also in a github repo
  - VM pulls changes from the repo on every instantiation → latest versions always available without image rebuild
- Metadata passed to the VM:
  - JSON dict with list of app-version to install

Application description

```
“<application name>” :{
      "app_name": "<application name>",
      "dependencies": [ <list of dependencies> ],
      "installer": "<installer script>",
      "base_url": ”<url for downloading app>"
      "versions": {
          "<version name>": {
              "app_version": "<version name>" ,
              …
          },
          …
       }
}
```

Sample app:

```
“FormCalc” :{
    “app_name": ”FormCalc",
    "dependencies": [ “FeynHiggs” ],
    "installer": "feyntools.sh",
    "base_url": "http://www.feynarts.de/formcalc/",
    "versions": {
       "7.0.2": {
         "base_url": "https://devel.ifca.es/~enol/
feynapps/",
         "app_version": "7.0.2"
       },
       "7.4": {
         "app_version": "7.4"
       }
    }
}
```

Metadata:

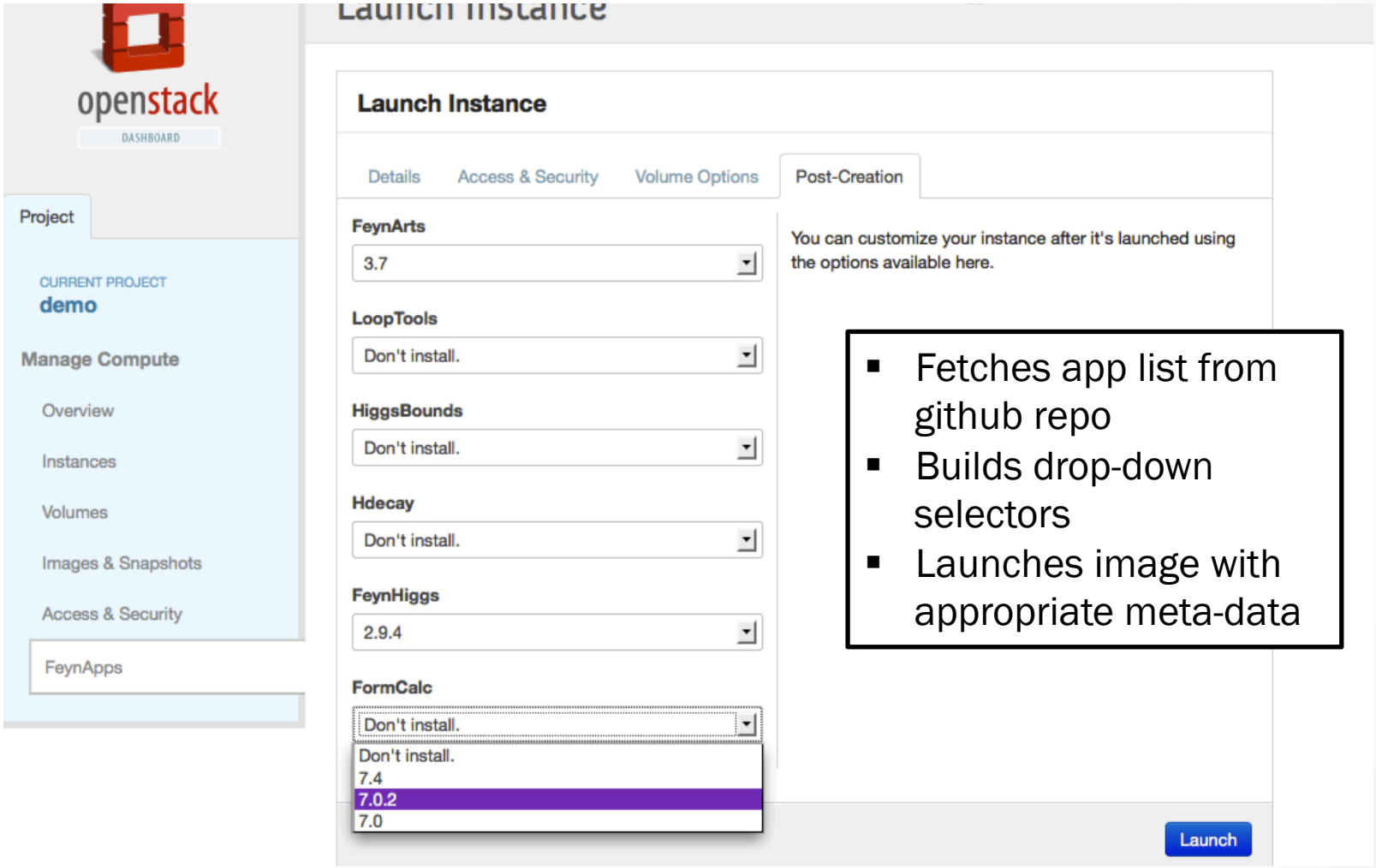
Expected Format:

```
{
  "apps": {
    "<name of app 1>": "<version of app 1>",
    "<name of app 2>": "<version of app 2>",
     ...
  }
}
```

Sample:

```
{
  "apps": {
    "FeynHiggs": "2.9.4",
    "FormCalc": "7.4",
    "FeynArts": "3.7",
  }
}
```

FeynApps Horizon Panel



- Fetches app list from github repo
- Builds drop-down selectors
- Launches image with appropriate meta-data

Use case 1: Physics Problem

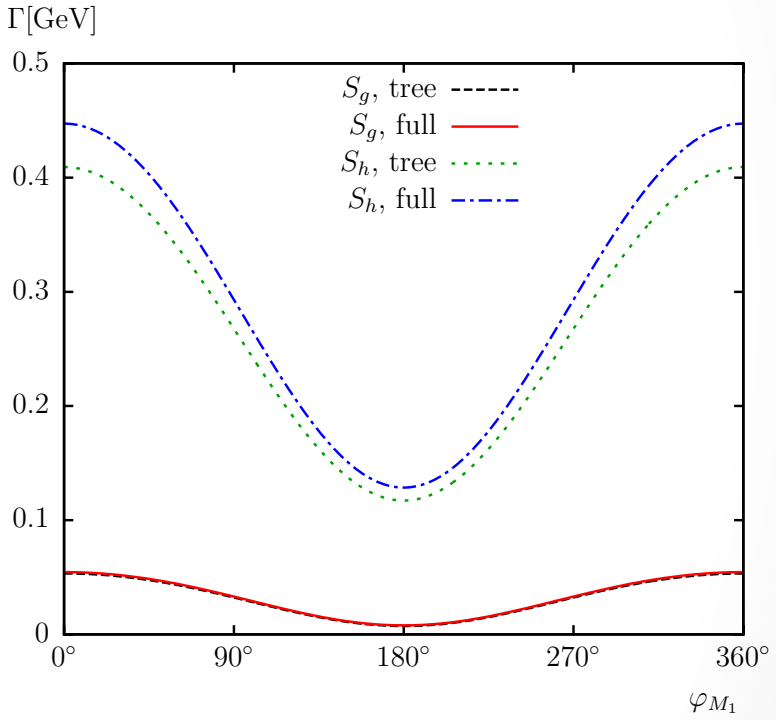
- Minimal Supersymetric Standard Model (MSSM) simulation of *heaviest neutralino* disintegration modes:

$$\tilde{\chi}_4^0 \to \tilde{\chi}_1^0 h_1 \ ,$$
$$\tilde{\chi}_4^0 \to \tilde{\chi}_1^+ W^- \ .$$

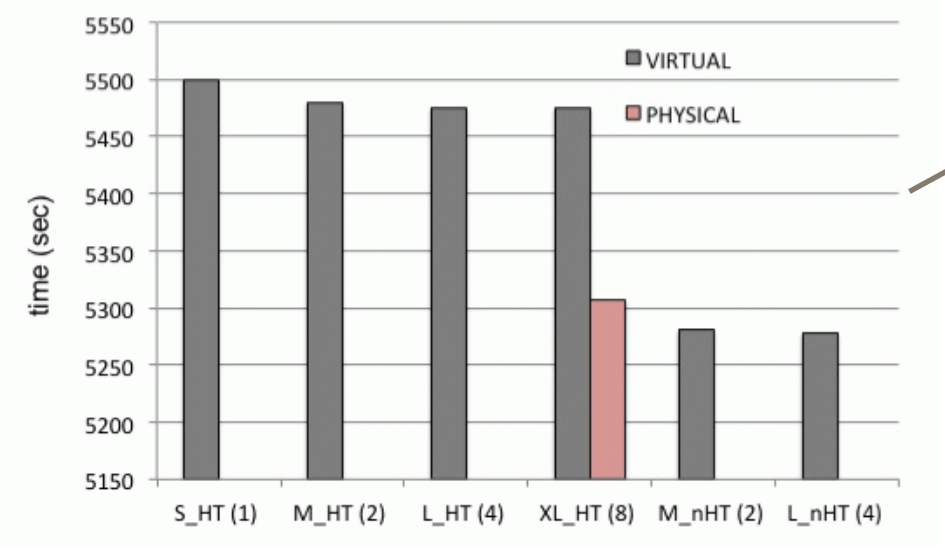
- Software workflow
  1. FeynArts: construct the Feynman diagrams that describe the process
  2. FormCalc: from FeynArts output produce Fortran code that numerical resolve the analytical formulas
  3. Free parameter evaluation of the model with the Fortran code linked with LoopTools (4-dim integrals) and FeynHiggs (Higgs particles predictions)

Use case 1: Physics Problem

- Variation of φ_{M1} from 0 to 360
- 361 steps, 2 parameters → 722 decay evaluations
- 2 calculation methods:
  - tree (approximate)
  - full (better precision)

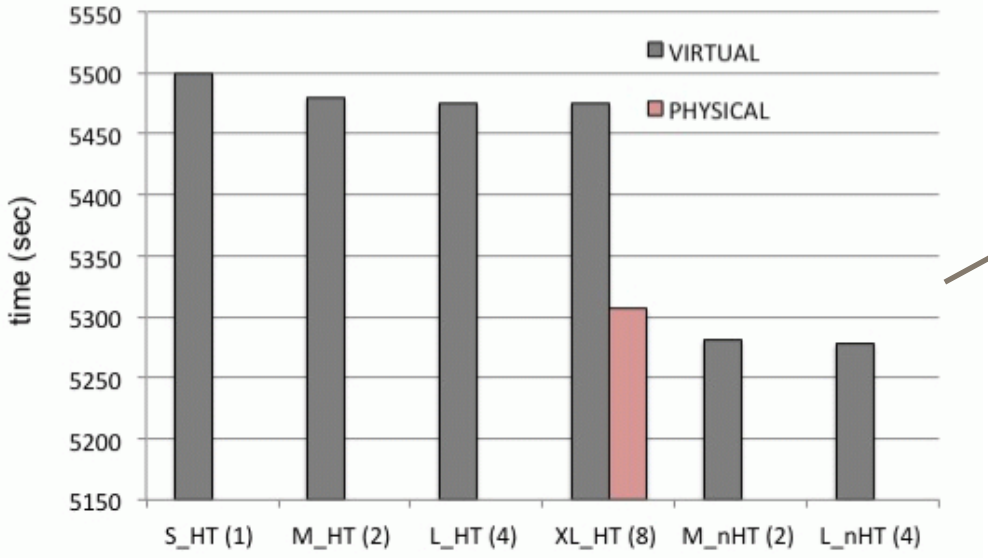


Use case 1: Single process



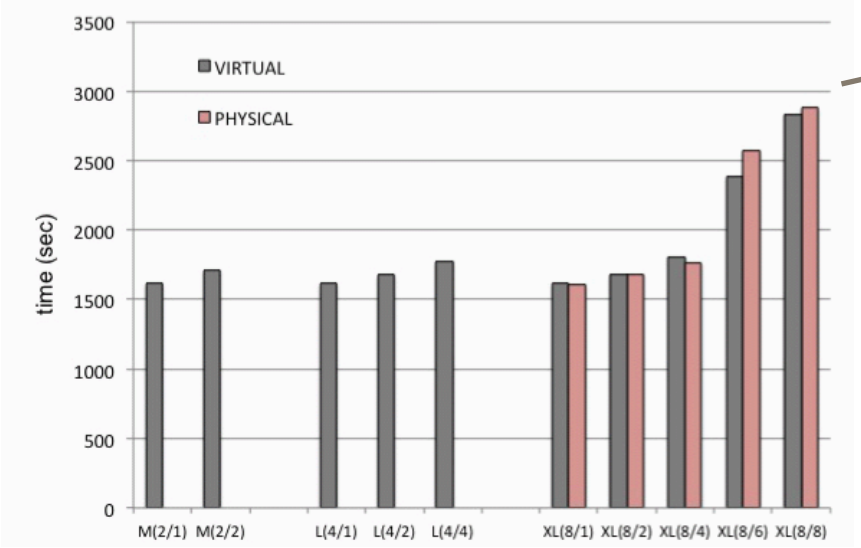
Mathematica Part (FeynArt, FormCalc)

```
S    → 1 core
M    → 2 cores
L    → 4 cores
XL   → 8 cores
HT   → enabled hyperthreading
nHT  → disabled hyperthreading
```



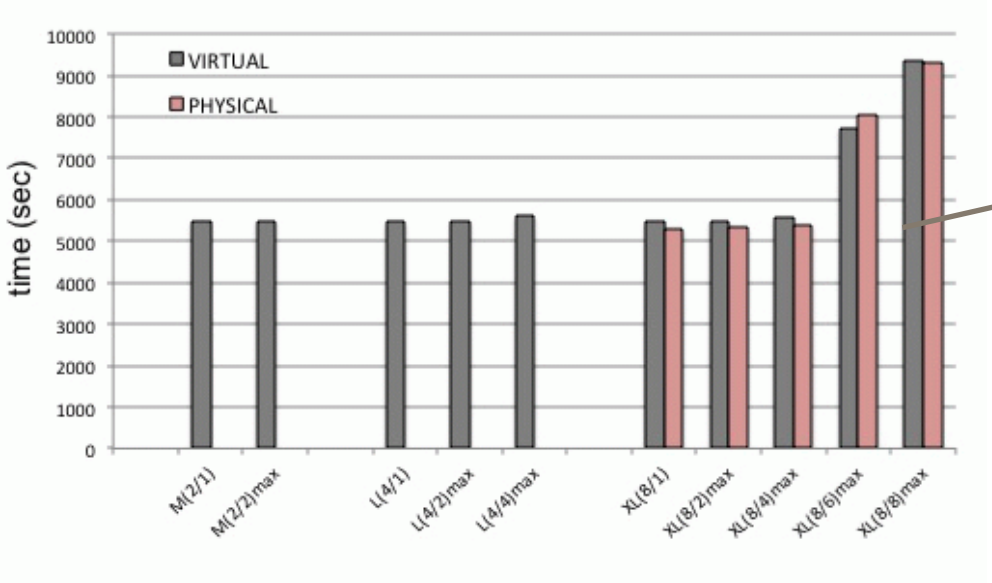
Fortran Part (FeynHiggs, LoopTools)

Use case 1: Multiple processes



Mathematica Part
691.1MB per process

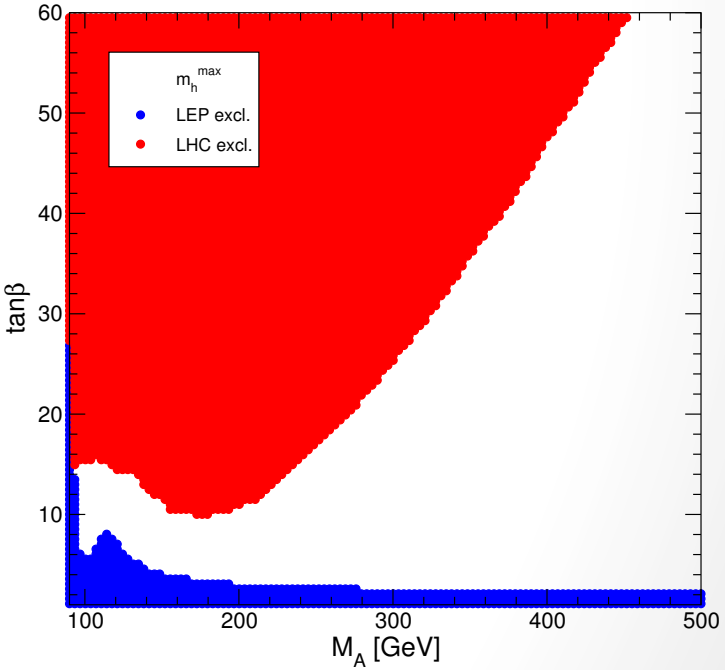
```
S   → 1 core
M   → 2 cores
L   → 4 cores
XL  → 8 cores
```



Fortran Part
189.1MB compilation
36.9 MB execution

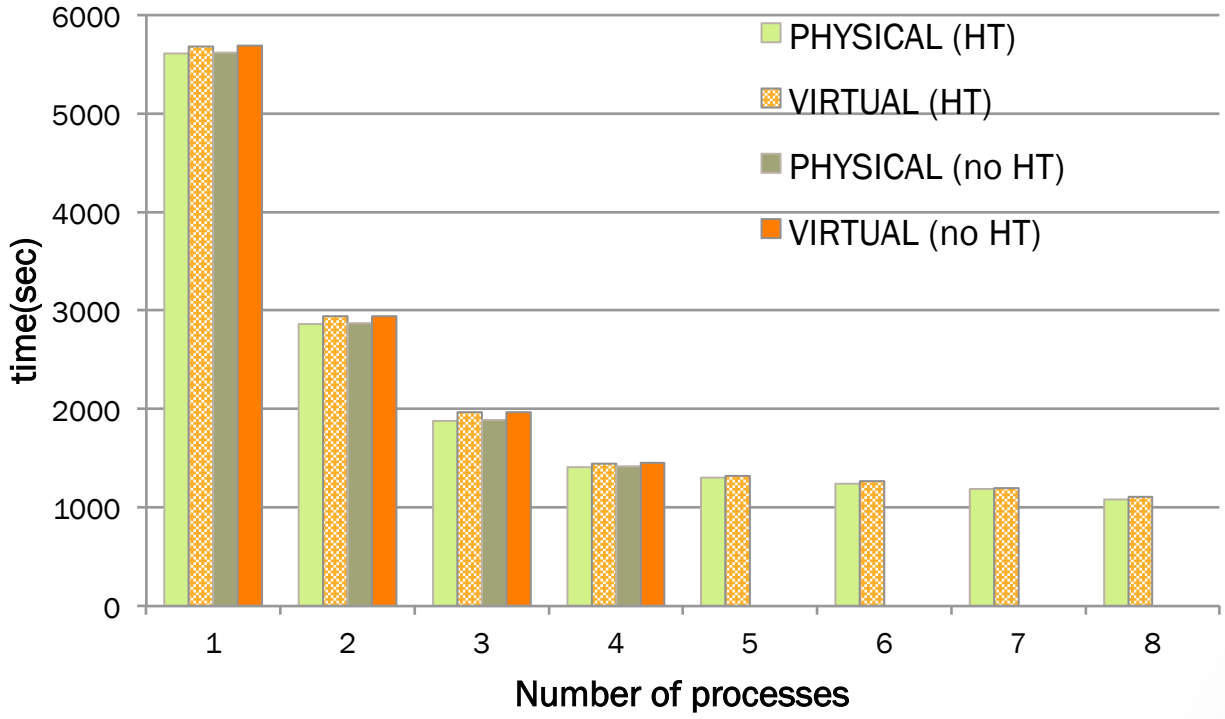
Use case 2: Physics Problem

- Parameter scans of free parameters of the MSSM
  - M_A → mass of Higgs particle in MSSM
  - tan_β → *mixing angle*
- Which values are allowed/excluded according to experimental measurements?
- Steering code calls:
  - FeynHiggs to determine Higgs properties
  - HiggsBounds to determine allowed/excluded values
- Parameter scan (120 steps per p)
  - M_A 90 →500
  - tan_β 1.1 → 60



Use case 2: Parallelization

- OpenMP discarded due to non thread-safe libraries
- Simple MPI parallelization:
  - parameter space is equally divided among all processes
- Lineal Speed-up



Conclusions

- Particle Physics Phenomenology group @ IFCA has a new computing environment based on IaaS cloud
  - flexible infrastructure to suit their needs
  - contextualization provides VMs with all needed software ready to use
  - snapshots provide a way to review results from previous simulations easily
- Good performance on VM
  - less than 5% performance loss
  - HT does not double performance
- Future
  - Extend the contextualization to other areas/cloud stacks

Thanks!

- Accepted for publication in EPJ-C
  - pre-print at http://arxiv.org/abs/1212.4784
- More on VOMS-Keystone:
  - *VOMS-aware identity service for OpenStack* presentation today at 14:00 on Cloud Platforms track
- Repos on github
  - http://github.com/IFCA/feynapps
  - https://github.com/IFCA/feynapps_panel
- See it in action!
  - Today at IBERCLOUD demo